



## 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, P.R.C. 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. 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 Edifício 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 20, 5652 AG Eindhoven, The Netherlands  
TEL: 31-40-2592850 / FAX: 31-40-2592851

DVP-4949620-07

2020-03-12

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

# ISPSOft User Manual



## ISPSOft User Manual

[www.deltaww.com](http://www.deltaww.com)



# ISPSOft User Manual

## Revision History

Version	Revision	Date
1 <sup>st</sup>	The first version was published.	2013/01/18
2 <sup>nd</sup>	Appendix D and Appendix E are added.	2015/03/18
3 <sup>rd</sup>	<ol style="list-style-type: none"> <li>1. New contents concerning AHxxEMC and AS300 series are added in Chapter 3.</li> <li>2. New contents concerning Data Unit Type are added in Chapter 8.</li> <li>3. New contents concerning Axis are added in Chapter 9.</li> <li>4. New contents concerning Continuous Function Charts are added in Chapter 15.</li> <li>5. New contents concerning G-Code Editor and E-CAM Editor are added in Chapter 21.</li> <li>6. New contents concerning Wizard Tool are added in Chapter 22.</li> <li>7. New contents concerning AHxxEMC and AS300 series are added in Appendix B.</li> <li>8. Appendix D and Appendix E are removed.</li> <li>9. Update all software images.</li> </ol>	2016/05/13
4 <sup>th</sup>	<ol style="list-style-type: none"> <li>10. New contents concerning setting communication parameters for AS300 simulator are added in Chapter 2.</li> <li>11. Update information concerning AHxxEMC and AS300 in Chapter 3.</li> <li>12. New contents concerning axis are updated in Chapter 9.</li> <li>13. Update new folding function in the action list for SFC.</li> <li>14. New contents concerning change DVP series modules to AS series modules are updated in Chapter 16.</li> <li>15. Update information concerning online monitoring in Chapter 17.</li> <li>16. New contents concerning ECAM editor and polynomial are added in Chapter 21.</li> <li>17. New contents concerning planning table, lists, data tracer, data logger and double vertical measurement lines are added in Chapter 22.</li> <li>18. New contents concerning installing the USB driver in Windows XP with SP3, Windows 7, Window 8 and Window 10 are added in Appendix A.</li> <li>19. The ranges of the S device for AS300 series are revised in Appendix B.</li> </ol>	2017/06/15
5 <sup>th</sup>	<ol style="list-style-type: none"> <li>1. Add new contents regarding DVP15MC and information on ISPSOft for installing/removing in Chapter 1.</li> <li>2. Update backup file number of auto save project and limitation on the number of AH/DV{/AS simulators; new contents on DVP15MC and edited routing mode information are added in Chapter 2.</li> <li>3. Add new information containing HWCONFIG for AH500 redundancy system series and DVP15MC, redundant extension rack, new tab-IO input filter</li> </ol>	2018/06/12

Version	Revision	Date
	<p>content in HWCONFIG basic CPU parameter setting of AHxxEM series. Edit basic CPU parameter-Name and Comment boxes, parameters of AS series CPU modules in Chapter 3.</p> <ol style="list-style-type: none"> <li>4. Update contents regarding task management for AH500 redundancy system and DVP15MC in Chapter 5.</li> <li>5. Edit AS series data types and add information like S device supports by AH series CPU modules and data types of DVP15MC as well as symbol classes in Chapter 6.</li> <li>6. Add data types not supported by the function blocks in AS series CPU modules, assigning memory blocks to function blocks in AH redundancy system series, edit basic specifications for function blocks and new contents regarding function blocks I/O and library for DVP15MC are included in Chapter 7.</li> <li>7. Add contents regarding structure DUT and limitation on number of enumerations, change structure to enumeration and from enumeration to structure as well as information for DVP15MC in Chapter 8.</li> <li>8. Updates on cycle time for axis in the project management section, introduce 'Synchronize with ECAT Builder' option and add information on axis parameter setting including node address, monitoring and test run functions for DVP15MC and AHxxEMC series in Chapter 9.</li> <li>9. Add new contents including MPS instruction and information on DVP15MC; edit ladder diagram regarding activating or inactivating a network in Chapter 10.</li> <li>10. Edit contents of putting labels in Chapter 11</li> <li>11. Add ST instructions not supported by the AH/AS series CPU modules and ST instructions supported by the AS series CPU modules but with limitations in Chapter 13.</li> <li>12. Edit contents of S device occupied by the STEP data type in the PLC and add AH500 redundancy system series in running the 'Final Scan' as well as SFC system structure contents in Chapter 14.</li> <li>13. Correct the wording used in warning the operation that causes loss of data or task; add new contents of DVP15MC; edit device comment lists of AH/AS/DVP series and contents in used device report in Chapter 16.</li> <li>14. Add new contents including DVP15MC, online editing and update function, DVP simulator information; edit debugging mode for DVP series, and system log functions for AH500 redundancy system series in Chapter 17.</li> <li>15. Expand contents regarding DVP15MC in Chapter 18.</li> <li>16. Edit contents of NWCONFIG in Chapter 19.</li> <li>17. Add new information regarding CARD Utility for AH500 redundancy system series and edit backup source in card utility for AS series added in</li> </ol>	

Version	Revision	Date
	<p>Chapter 20.</p> <ol style="list-style-type: none"> <li>18. Correct Chapter 21 of G-code editor to 'not supported by the current ISPSOft version' and edit E-CAM functions.</li> <li>19. Edit contents regarding position planning table, data tracer, data logger and high speed counter in Chapter 22.</li> <li>20. Add AH560EN2 module to the installation path in Appendix A</li> <li>21. Add new contents regarding DVP15MC, AHCPU501-EN, AHCPU501-RS2, AHCPU521-RS2, AHCPU531-RS2, AS200 series; update data exchange setups and NWCONFIG; edit the address range for AHCPU511-EN/ AHCPU511-RS2 and AHCPU531-EN modules and program compilation &amp; time for uploading/downloading in Appendix B.</li> </ol>	
6 <sup>th</sup>	<ol style="list-style-type: none"> <li>1. Modify chapter 1: add DVPxxMC description; information on installing ISPSOft and remove modified ISPSOft content; installing COMMGR and remove modified COMMGR content; add language supported by PLC types in ISPSOft and function support list.</li> <li>2. Modify chapter 2: ISPSOft first step, window title; content, function toolbar and icon toolbar; project, message and edit work content section; project structure in ISPSOft, single project and group project; integrate motion module project content; system and environment setting; enable and close communication COMMGR as well as drive management; create connection channel, create ISPSOft and COMMGR connection as well as connection with host and communication port.</li> <li>3. Modify chapter 3: DVPxxMC PLC host parameter setting; AH560 redundant system parameter setting and related content; HWCONFIG tools.</li> <li>4. Modify chapter 4: add new project and hardware configuration and module allocation; host and module parameter setting and create program content; basic edit- select ladder diagram section and increase operation function; basic edit- insert API instructions, program check and editing; create connection between testing and debugging, download program and configuration parameter, connection testing examples.</li> <li>5. Modify chapter 5: the POU structure and task management in ISPSOft; task in project section and POU execution order content in task; create POU and property setting as well as POU active state, delete and copy POU, POU password setting and canceling, export and import POU, export and import function block POU; task management operation and task property and condition for interrupt, POU allocation, POU order and content; programming examples of POU and task, interrupt program examples of DVP series, interrupt program examples of AH500 series.</li> <li>6. Modify chapter 6: symbol variable category and</li> </ol>	2019/06/14

Version	Revision	Date
	<p>data types as well as position allocation and initial value content; symbol variable table and add symbol variable as well as array and string symbols, modification on symbol variable and edit symbol table, download initial value of symbol variables, device resource allocation setting.</p> <p>7. Modify chapter 7: symbol variable of function block, input and output of function block; the function block call or 'nesting FBs' and function block memory allocation and basic specification; create program example; add and refer to user library as well as Delta library.</p> <p>8. Modify chapter 8: create Structure content</p> <p>9. Modify chapter 9: create axis content in AHxxEMC; create axis and basic settings in DVPxxMC; the main symbol and axis symbol in global symbol table; AHxxEMC and DVPxxMC axis parameter setting; AHxxEMC and DVPxxMC monitoring and testing</p> <p>10. Modify chapter 10: LD edit environment in ISPSOft; use device symbol variable and contain in LD, insert API instruction and function block, edit section logic command, switching between symbol mode and position mode content.</p> <p>11. Modify chapter 11: FBD edit environment in ISPSOft, FBD device symbol variable and constant; contacts of API comparison and function block content; switching between symbol mode and position mode content.</p> <p>12. Modify chapter 12: edit IL command, insert API and function block content.</p> <p>13. Modify chapter 13: ST edit environment in ISPSOft, edit ST description, insert API and function block, create program example.</p> <p>14. Modify chapter 14: Step and Action, Transition, Jump, condition for Action modifiers, condition defined, initial Step; internal attribute, Step attribute parameter, Action and Transition attribute parameter; SFC edit environment in ISPSOft, create and manage Action &amp; Transition, assign Step and Transition symbol, allocate Action/ folded Action list; create program example.</p> <p>15. Modify chapter 15: instruction and function block content.</p> <p>16. Modify chapter 16: add modified PLC types list, add project download and upload list, device note table and device use status for DVP series, content for program alignment.</p> <p>17. Modify chapter 17: debugging mode in DVP series, AH500 simulator content; PLC system information, PLCY system log content.</p> <p>18. Modify chapter 18: ISPSOft password structure, POU password, subroutine password, and other password protection function.</p> <p>19. Modify chapter 19: NWCONFIG brief intro, basic introduction, communication parameters setting, task procedure modification; deploy node, connect internet, node and internet property setting,</p>	

Version	Revision	Date
	<p>device or hide/show internet, legal internet structure, download routing table, routing table testing content; choose host device, communication parameter setting, data exchange table, section setting, synchronized data exchange for PLCs, management for data exchange table, PLC Link monitoring function, setup PLC Link notes; Ether Link brief intro, open Ether Link setting window, create and manage data exchange table, device list and icon operation, Ether Link operating mode, download Ether Link device, upload Ether Link device, delete non synchronized device, online start/stop Ether Link, system download, routing application in ISPSOft.</p> <p>20. Modify chapter 20: information backup memory card operation, PLC permanent backup setting, memory card utility software intro and backup operation, restore operation content.</p> <p>21. Modify chapter 21: delete G-Code editor; E-CAM editor, about E-CAM and its details as well as usage.</p> <p>22. Modify chapter 22: setup position planning, position planning simulation, upload and download position planning table ; axis control mode- point to point single axis mode, single axis multiple section output mode, 2-axis linear interpolation mode, 2-axis arc interpolation mode; about and open data tracer, parameter sample setting, sample mode, measurement content; About and open data logger, setting sample parameters, monitoring and log mode, measurement content; About high-speed timer wizard.</p> <p>23. Modify Appendix A: install USB driver in Windows 10.</p> <p>24. Modify Appendix B: DVPxxMC device types and device address format as well as retain latched device.</p>	
7 <sup>th</sup>	<ol style="list-style-type: none"> <li>1. Add a new Chapter 16; the chapter number of the chapters after chapter 16 have been modified accordingly.</li> <li>2. Update List of Programming Languages supported by ISPSOft in Chapter 1.</li> <li>3. Update the project display and add new initialization options for PLC in Chapter 2.</li> <li>4. Add new introductions on HWCONFIG 4.0 in Chapter 3.</li> <li>5. Update execution images of the EI instruction in section 5.3.2.</li> <li>6. Update descriptions on Global Symbols which may have more sub-options for some series and update descriptions on the initial values in Chapter 6.</li> <li>7. Update descriptions on bookmarks which can be inserted in a program in Chapter 10.</li> <li>8. Add descriptions on the usage of multiple comments (using words between (* and *), (/ * and */) and single line comment (using (/ /)) in a</li> </ol>	2020/03/12

Version	Revision	Date
	<p>structured text in Chapter 13. When a program is compiled, the system automatically skips the words between (* and *) , (/ * and */) and (/ /) in the program.</p> <p>9. Update descriptions on Search for Device and Search for Symbols in Chapter 17.</p> <p>10. Add descriptions of the function of Temporarily Store Password in Online Editing Function in Chapter 18.</p> <p>11. Update descriptions of G-Code functions in Chapter 22.</p>	

# ISPSoft User Manual

## Table of Contents

<b>Preface .....</b>	<b>I</b>
 <b>Chapter 1 Introducing ISPSoft</b>	
<b>1.1 Introduction of ISPSoft and System Requirements.....</b>	<b>1-2</b>
1.1.1 Characteristics.....	1-2
1.1.2 System Requirements.....	1-3
1.1.3 Installing ISPSoft .....	1-4
1.1.4 Uninstalling ISPSoft.....	1-8
<b>1.2 Introduction of COMMGR .....</b>	<b>1-10</b>
1.2.1 Operating Mode of COMMGR.....	1-10
1.2.2 Installing COMMGR .....	1-11
1.2.3 Uninstalling COMMGR .....	1-13
<b>1.3 ISPSoft for PLC Devices .....</b>	<b>1-14</b>
1.3.1 ISPSoft - List of Programming Languages.....	1-14
1.3.2 ISPSoft - List of Functions .....	1-15
 <b>Chapter 2 Starting and Setting ISPSoft</b>	
<b>2.1 Guidelines and Environment .....</b>	<b>2-2</b>
2.1.1 Getting Started.....	2-2
2.1.2 Window Title and Status Bar .....	2-6
2.1.3 Toolbar - Functions.....	2-7
2.1.4 Toolbar - Icons .....	2-10
2.1.5 Project and Compile Message .....	2-11
2.1.6 Workspace .....	2-13
<b>2.2 Project Framework.....</b>	<b>2-15</b>
2.2.1 Single Project and Group Project .....	2-15
2.2.2 Integration of Motion Control Modules .....	2-16
2.2.3 Managing a Single Project .....	2-17



2.2.4	Managing Group Project.....	2-22
2.2.5	Managing Motion Control Modules .....	2-28
<b>2.3</b>	<b>Basic Configuration .....</b>	<b>2-32</b>
2.3.1	System and Environment .....	2-32
2.3.2	Importing and Exporting User Preference .....	2-38
<b>2.4</b>	<b>Communication Settings .....</b>	<b>2-39</b>
2.4.1	Start/Close COMMGR.....	2-39
2.4.2	Driver Management for COMMGR .....	2-41
2.4.3	Creating Connection Channel - Add Driver.....	2-41
2.4.3.1	Communication Parameter Setting - RS232/422/485.....	2-43
2.4.3.2	Communication Parameter Setting - USB (Virtual COM) .....	2-44
2.4.3.3	Communication Parameter Setting - DirectLink (USB) .....	2-45
2.4.3.4	Communication Parameter Setting - Ethernet.....	2-45
2.4.3.5	Communication Parameter Setting - DirectLink (Ethernet).....	2-46
2.4.3.6	Communication Parameter Setting - DVP Simulator.....	2-47
2.4.3.7	Communication Parameter Setting - AH5x0 & AH5x1 Simulator.....	2-48
2.4.3.8	Communication Parameter Setting - AS Simulator .....	2-48
2.4.4	Creating Connection Channel - Start/Stop a Driver.....	2-49
2.4.5	Creating Connection Channel - Configure/Delete a Driver.....	2-50
2.4.6	Creating Connection Channel - Simulator with Operating Panels .....	2-51
2.4.7	Creating Connection between ISPSOft and COMMGR.....	2-54
2.4.8	Connecting PLC (Host) and Communication Port .....	2-57
2.4.9	Practical Connection Test .....	2-59

## Chapter 3 PLC System Configuration & Settings

<b>3.1</b>	<b>HWCONFIG 4.0 – Hardware Configuration Tool .....</b>	<b>3-3</b>
3.1.1	Open Your HWCONFIG 4.0 .....	3-4
3.1.2	HWCONFIG 4.0 Environment .....	3-6
3.1.3	Module Configurations .....	3-15
3.1.4	Parameter Settings for PLC CPU.....	3-33
<b>3.2</b>	<b>Hardware Configuration Tool – HWCONFIG 3.0 .....</b>	<b>3-66</b>
3.2.1	HWCONFIG Environment.....	3-66
3.2.2	Module Configuration.....	3-69
3.2.3	Managing the Version of a Module .....	3-107

<b>3.3</b>	<b>Parameter Settings for AH500 Series PLC .....</b>	<b>3-110</b>
3.3.1	Open the PLC CPU Parameter Setting Page .....	3-110
3.3.2	Setting CPU Parameters .....	3-111
3.3.3	Setting COM Port Parameters .....	3-120
3.3.4	Setting Ethernet Parameters.....	3-121
3.3.5	Setting Advanced Ethernet Parameters .....	3-122
3.3.6	Data Exchange Setup - COM & Ethernet.....	3-140
<b>3.4</b>	<b>Parameter Settings for AHxxEMC Series PLC .....</b>	<b>3-144</b>
3.4.1	Open the PLC CPU Parameter Setting Page.....	3-144
3.4.2	Options.....	3-146
3.4.3	Data Exchange - COM2 &Ethernet .....	3-157
<b>3.5</b>	<b>Parameter Setting for DVP Series PLC.....</b>	<b>3-162</b>
3.5.1	System Management Tools for DVP Series PLC .....	3-162
3.5.2	Retentive Range .....	3-163
3.5.3	Connected Information .....	3-167
3.5.4	Parameter Setting for DVP-ES3 Series PLC .....	3-168
<b>3.6</b>	<b>Parameter Settings for DVPxxMC Series PLC.....</b>	<b>3-170</b>
3.6.1	Opening the PLC Parameter Setting Window.....	3-170
3.6.2	Options.....	3-171
<b>3.7</b>	<b>Parameter Settings for Modules .....</b>	<b>3-182</b>
3.7.1	Setting Internal Parameters for AH5x0 and AH5x1 Modules.....	3-182
3.7.2	Parameter Settings for AHxxEMC Modules .....	3-187
3.7.3	Exporting and Importing Parameters for Modules .....	3-189
3.7.4	Updating Parameters for Modules.....	3-190
3.7.5	Parameter Settings for Intelligent Modules .....	3-190
<b>3.8</b>	<b>HWCONFIG Parameter Management and Online Diagnosis</b>	<b>3-191</b>
3.8.1	Save and Print function.....	3-191
3.8.2	Purchase Order .....	3-192
3.8.3	Rack Information List for AH Series .....	3-193
3.8.4	Download/Upload System Parameters.....	3-194
3.8.5	I/O Scan .....	3-196
3.8.6	AH Series Online Diagnosis.....	3-199
<b>3.9</b>	<b>Setting a RTC.....</b>	<b>3-206</b>
<b>3.10</b>	<b>Setting the Memory in a PLC .....</b>	<b>3-207</b>

## Chapter 4 Quick Start

<b>4.1 Quick Start .....</b>	<b>4-2</b>
4.1.1 Example .....	4-2
4.1.2 Hardware.....	4-3
4.1.3 Program .....	4-3
<b>4.2 Procedure for Creating a Project in ISPSoft.....</b>	<b>4-4</b>
<b>4.3 Creating a Project.....</b>	<b>4-5</b>
<b>4.4 Hardware Configuration.....</b>	<b>4-6</b>
4.4.1 Configuring a Module .....	4-6
4.4.2 Parameter Setting for PLCs and Modules.....	4-8
<b>4.5 Creating a Program .....</b>	<b>4-10</b>
4.5.1 Adding a Ladder Diagram .....	4-10
4.5.2 Basic Editing - Creating a Contact and a Coil.....	4-12
4.5.3 Basic Editing - Inserting a Network and Typing an Instruction .....	4-15
4.5.4 Basic Editing - Selection of a Network and Operation.....	4-17
4.5.5 Basic Editing - Connecting a Contact in Parallel .....	4-20
4.5.6 Basic Editing - Editing a Comment .....	4-21
4.5.7 Basic Editing - Inserting an Applied Instruction.....	4-22
4.5.8 Basic Editing - Creating a Comparison Contact and Typing a Constant ....	4-24
4.5.9 Writing a Program .....	4-25
4.5.10 Checking and Compiling a Program .....	4-26
<b>4.6 Testing and Debugging a Program.....</b>	<b>4-27</b>
4.6.1 Creating a Connection .....	4-27
4.6.2 Downloading a Program and Parameters.....	4-30
4.6.3 Connection Test.....	4-33

## Chapter 5 POU and Task

<b>5.1 Program Organization Units and Tasks.....</b>	<b>5-2</b>
<b>5.2 Program Organization Units .....</b>	<b>5-3</b>
5.2.1 Program Architecture and Types .....	5-3
5.2.2 POUs in ISPSoft .....	5-4

<b>5.3</b>	<b>Tasks</b> .....	<b>5-6</b>
5.3.1	Managing the Tasks in ISPSOft.....	5-6
5.3.2	Tasks in the Project Management Area .....	5-11
5.3.3	Executing the POU's Assigned to a Task.....	5-13
<b>5.4</b>	<b>Managing a POU</b> .....	<b>5-16</b>
5.4.1	Creating and Setting a POU.....	5-16
5.4.2	POU Active State .....	5-19
5.4.3	Delete and Copy POU .....	5-20
5.4.4	Set and Remove POU Password .....	5-22
5.4.5	Exporting POU Program .....	5-23
5.4.6	Importing POU Program.....	5-24
5.4.7	Exporting Function Block POU.....	5-25
5.4.8	Importing Function Block POU .....	5-26
<b>5.5</b>	<b>Task Management</b> .....	<b>5-28</b>
5.5.1	Setting Task Description and Condition for Interruption .....	5-30
5.5.2	Configuration of POU .....	5-34
5.5.3	POU Order .....	5-36
<b>5.6</b>	<b>Example</b> .....	<b>5-37</b>
5.6.1	Programming TASK and POU .....	5-37
5.6.2	Example of an Interrupt Subroutine in a DVP Series PLC.....	5-41
5.6.3	Examples of AH5x0 Series Interrupt Programs.....	5-47

## Chapter 6 Symbol Variables

<b>6.1</b>	<b>Introduction of Symbols</b> .....	<b>6-2</b>
6.1.1	Application of Symbols and Creation of Identifiers.....	6-2
6.1.2	Symbol Variable Classification.....	6-3
6.1.3	Data Types.....	6-5
6.1.4	Symbol Variable Address Allocation and Initial Value.....	6-11
6.1.5	Indirect Assignment and Modification for Register Symbol Variable.....	6-14
6.1.6	Bit Operation of Symbol Variable (Only AH/AS Series ) .....	6-16
<b>6.2</b>	<b>Symbol Variable Management in ISPSOft</b> .....	<b>6-18</b>
6.2.1	Symbol Variable Table .....	6-18

6.2.2	Adding Symbol Variable .....	6-20
6.2.3	Principles of ARRAY or STRING Symbol .....	6-25
6.2.4	Modify Symbol Variables and Edit Symbol Table .....	6-29
6.2.5	Remove Symbol Variable Allocated Address .....	6-30
6.2.6	Downloading the Initial Values of the Symbols.....	6-31
6.2.7	Export Symbol Table .....	6-32
6.2.8	Importing a Symbol Table.....	6-33
6.2.9	Arranging the symbols.....	6-36
6.2.10	Filtering the Symbols.....	6-37
6.2.11	Device Resource Allocation Setting.....	6-38
<b>6.3</b>	<b>Example .....</b>	<b>6-39</b>
6.3.1	Planning a Symbol Table.....	6-39
6.3.2	Writing a Program.....	6-41

## Chapter 7 Function Block and Library

<b>7.1</b>	<b>Knowing Function Blocks .....</b>	<b>7-2</b>
7.1.1	Introduction of Function Blocks.....	7-2
7.1.2	Characteristics and Advantages of Function Blocks .....	7-3
<b>7.2</b>	<b>Structure of the Function Blocks in ISPSoft.....</b>	<b>7-5</b>
7.2.1	En Pin of Function Block.....	7-5
7.2.2	Symbol Variables in Function Block .....	7-6
7.2.3	Input/Output Pins of a Function Block.....	7-8
7.2.4	Index Type Symbol Variables.....	7-14
7.2.5	Function Block Definition and Function Block Instance .....	7-19
7.2.6	Calling Relation Between Function Blocks.....	7-22
7.2.7	Memory Configuration of Function Block .....	7-25
<b>7.3</b>	<b>Using a Function Block .....</b>	<b>7-33</b>
7.3.1	Basic Specifications for Function Blocks .....	7-33
7.3.2	Pulse Instructions for Function Blocks (AH/AS Series ONLY) .....	7-35
7.3.3	Monitoring the Program in a Function Block.....	7-36
7.3.4	Modifying the Program in a Function Block .....	7-37

<b>7.4</b>	<b>Instance</b> .....	<b>7-38</b>
7.4.1	Planning a Program .....	7-38
7.4.2	Creating the Program .....	7-39
<b>7.5</b>	<b>Knowing the Library</b> .....	<b>7-49</b>
7.5.1	Creating the User-defined Library .....	7-49
7.5.2	Including the Function Blocks in the User Defined Library.....	7-51
7.5.3	Using Delta Library.....	7-53

## Chapter 8 Data Unit Type

<b>8.1</b>	<b>User-defined Data Type/Data Unit Type</b> .....	<b>8-2</b>
<b>8.2</b>	<b>Structure</b> .....	<b>8-2</b>
8.2.1	Definition of a Structure.....	8-2
8.2.2	Creating a Structure .....	8-2
8.2.3	Using a Structure Variable.....	8-5
8.2.4	Applications of Structures.....	8-6
8.2.5	Change Structure to Enumeration .....	8-8
<b>8.3</b>	<b>Enumeration</b> .....	<b>8-9</b>
8.3.1	Definition of Enumeration.....	8-9
8.3.2	Creating an Enumeration.....	8-9
8.3.3	Using an Enumeration Variable .....	8-11
8.3.4	Applications of Enumerations .....	8-13
8.3.5	Change Enumeration to Structure .....	8-14

## Chapter 9 Axis

<b>9.1</b>	<b>Axis</b> .....	<b>9-2</b>
9.1.1	The Meaning of Axis.....	9-2
9.1.2	Creating a New Axis.....	9-3
9.1.3	Main Table and Axis Table under Global Symbols.....	9-10
<b>9.2</b>	<b>Axis Parameter and Monitor &amp; Test Run</b> .....	<b>9-11</b>
9.2.1	Axis Parameter .....	9-11
9.2.2	Monitor & Test Run .....	9-21

## Chapter 10 Ladder Diagram

<b>10.1</b>	<b>Introduction of a Ladder Diagram .....</b>	<b>10-2</b>
10.1.1	Editing Environment .....	10-2
10.1.2	Networks in a Ladder Diagram .....	10-4
10.1.3	Selecting Objects.....	10-5
<b>10.2</b>	<b>Creating a Ladder Diagram in ISPSoft.....</b>	<b>10-7</b>
10.2.1	Creating a Contact, Coil & MPS.....	10-7
10.2.1.1	Inserting a Contact and Changing a Contact Type .....	10-7
10.2.1.2	Inserting a Coil and Changing a Coil Type .....	10-9
10.2.1.3	Add MPS .....	10-10
10.2.2	Use Device, Symbols and Constants in LD .....	10-11
10.2.3	Typing Instructions.....	10-13
10.2.4	Inserting Applied Instructions and Function Blocks.....	10-14
10.2.5	Creating a Comparison Contact .....	10-17
10.2.6	Inserting a Block Logic Instruction.....	10-19
10.2.7	Creating Multiple Outputs .....	10-22
10.2.8	Putting a Label .....	10-23
10.2.9	Editing a Comment .....	10-24
10.2.10	Symbol Mode and Address Mode .....	10-26
10.2.11	Bookmark.....	10-27
10.2.12	Activating/Inactivating a Network.....	10-28

## Chapter 11 Function Block Diagram

<b>11.1</b>	<b>Introduction of Function Block Diagrams .....</b>	<b>11-2</b>
11.1.1	Knowing Function Block Diagrams .....	11-2
11.1.2	Editing Environment .....	11-2
11.1.3	Networks in a Function Block Diagram.....	11-4
11.1.4	Selecting Objects.....	11-5
<b>11.2</b>	<b>Creating a Function Block Diagram in ISPSoft.....</b>	<b>11-6</b>
11.2.1	Input Nodes and Output Nodes .....	11-7
11.2.2	Use Devices, Symbols Constants in FB Diagram .....	11-11
11.2.3	And and OR Block.....	11-12

11.2.4	Inverse Logic .....	11-15
11.2.5	Rising and Falling edge-triggered Input .....	11-17
11.2.6	Setting an Output and Resetting an Output.....	11-19
11.2.7	API, Comparison Contact and Function Block.....	11-21
11.2.8	Setting Label.....	11-26
11.2.9	Comments and Hints.....	11-27
11.2.10	Symbol Mode and Address Mode.....	11-29
11.2.11	Bookmark.....	11-30
11.2.12	Activating/Inactivating a Network .....	11-31

## Chapter 12 Instruction List

<b>12.1</b>	<b>Introduction of Instruction Lists .....</b>	<b>12-2</b>
12.1.1	Structure of an Instruction List.....	12-2
12.1.2	Calling a Function Block.....	12-4
12.1.3	Important Points About Instruction Lists .....	12-5
<b>12.2</b>	<b>Create Instruction List in ISPSOft.....</b>	<b>12-7</b>
12.2.1	Editing Environment.....	12-7
12.2.2	Edit IL instruction .....	12-8
12.2.3	Insert API and Function Blocks .....	12-9
12.2.4	Bookmark .....	12-11
12.2.4.1	Adding Bookmarks .....	12-11
12.2.4.2	Removing Bookmarks .....	12-12
12.2.4.3	Going to a Bookmark.....	12-13

## Chapter 13 Structured Texts

<b>13.1</b>	<b>Introducing Structured Texts .....</b>	<b>13-2</b>
13.1.1	Basic Structure of a Structured Text.....	13-2
13.1.2	Statement .....	13-3
13.1.3	Expression.....	13-4
13.1.4	Operand and Operator.....	13-5
13.1.5	Keyword and Comment .....	13-6
13.1.6	Using Array Symbols in ST.....	13-8



13.1.7	Notes on ST Programming .....	13-9
<b>13.2</b>	<b>Structure of a Statement .....</b>	<b>13-11</b>
13.2.1	Assignment Structure—:= .....	13-11
13.2.2	Conditional Structure - IF .....	13-13
13.2.3	Conditional Structure—CASE.....	13-15
13.2.4	Loop Structure - REPEAT .....	13-18
13.2.5	Loop Structure - WHILE.....	13-19
13.2.6	Loop Structure - FOR .....	13-20
13.2.7	Applied Instruction Structure .....	13-22
13.2.8	Function Block Structure .....	13-23
13.2.9	Blank Statement .....	13-25
13.2.10	RETURN Statement.....	13-25
13.2.11	EXIT Statement.....	13-26
<b>13.3</b>	<b>Create a Structured Text in ISPSoft .....</b>	<b>13-27</b>
13.3.1	ST Editing Environment .....	13-27
13.3.2	Edit Structured Text .....	13-28
13.3.3	Insert API and Function Blocks .....	13-29
13.3.4	Bookmark.....	13-30
<b>13.4</b>	<b>Example of a Structured Text .....</b>	<b>13-32</b>
13.4.1	Explanation .....	13-32
13.4.2	Planning Hardware.....	13-32
13.4.3	Planning a Program.....	13-32
13.4.4	Creating a Program.....	13-33

## Chapter 14 Sequential Function Charts

<b>14.1</b>	<b>Knowing Sequential Function Charts.....</b>	<b>14-2</b>
14.1.1	Structure of a Sequential Function Chart .....	14-2
14.1.2	Principle of a Sequential Function Chart.....	14-3
<b>14.2</b>	<b>Sequential Function Chart in ISPSoft .....</b>	<b>14-4</b>
14.2.1	Steps and Actions.....	14-4
14.2.2	Transitions.....	14-7

14.2.3	Simultaneous Divergence and Divergence of Sequence Selection...	14-10
14.2.4	Simultaneous Convergence & Select Convergence .....	14-11
14.2.5	Jump .....	14-12
14.2.6	Qualifier of an Action.....	14-15
14.2.6.1	Qualifier Types.....	14-15
14.2.6.2	Important Points About Qualifying an Action.....	14-20
14.2.7	Initial Step.....	14-21
14.2.8	Internal Property .....	14-22
14.2.8.1	Step Property .....	14-22
14.2.8.2	Action and Transition Property .....	14-26
<b>14.3</b>	<b>Create SFC in ISPSoft .....</b>	<b>14-32</b>
14.3.1	Editing Environment.....	14-32
14.3.2	Creating and Managing Actions and Transitions .....	14-34
14.3.3	Adding a Step .....	14-36
14.3.4	Connecting a Transition in Parallel .....	14-37
14.3.5	Connecting a Step in Parallel.....	14-38
14.3.6	Step Structure - Simultaneous Divergence & Select Convergence ..	14-40
14.3.7	Step Structure – Select Divergence & Simultaneous Convergence..	14-41
14.3.8	Inserting a Jump Point .....	14-42
14.3.9	Assign Steps and the Transitions .....	14-43
14.3.10	Specifying an Initial Step.....	14-44
14.3.11	Assigning Actions / Fold the Action Table .....	14-44
<b>14.4</b>	<b>Sequential Function Chart Examples .....</b>	<b>14-48</b>
14.4.1	Example Descriptions.....	14-48
14.4.2	Planning Hardware.....	14-50
14.4.3	Planning a Program.....	14-50
14.4.4	Creating a Program.....	14-51
<b>Chapter 15 Continuous Function Charts</b>		
<b>15.1</b>	<b>Continuous Function Charts (CFC) .....</b>	<b>15-2</b>
15.1.1	About Continuous Function Charts .....	15-2
15.1.2	Things to Note When Using CFC .....	15-2
15.1.3	CFC Programming in ISPSoft.....	15-2

15.1.4	CFC Editing Toolbar .....	15-3
15.1.5	Shortcuts for CFC Programming .....	15-4
<b>15.2</b>	<b>Creating a CFC in ISPSoft .....</b>	<b>15-5</b>
15.2.1	Selecting Objects .....	15-5
15.2.2	Input/Output Nodes and Logic Gates .....	15-5
15.2.3	Changing a Pin Type .....	15-8
15.2.4	Connecting objects and Canceling Connections .....	15-9
15.2.5	Instructions and Function Blocks .....	15-9
15.2.6	Deleting objects .....	15-13
15.2.7	Editing Devices or Symbols.....	15-15
15.2.8	Activating/Inactivating an Object .....	15-16
15.2.9	Inserting a Comment .....	15-16
15.2.10	Changing the Order in Which Objects are Executed .....	15-18
15.2.11	Displaying/Hiding Information.....	15-19

## Chapter 16 C The Programming Language

<b>16.1</b>	<b>C The Programming Language (C) .....</b>	<b>16-2</b>
16.1.1	C Structures .....	16-2
16.1.2	C Environment .....	16-3
16.1.3	C Compiling and Debugging .....	16-9
16.1.4	Internal Function Library Instructions.....	16-10
16.1.5	Bookmark and Comment .....	16-11
16.1.6	Operation Example .....	16-13

## Chapter 17 Auxiliary Editing Tools

<b>17.1</b>	<b>ISPSOft Editing Tools and Auxiliary Functions.....</b>	<b>17-2</b>
17.1.1	Modify PLC Types .....	17-2
17.1.2	Change DVP Series Modules to AS Series Modules .....	17-3
17.1.3	Download/Upload a Project .....	17-14
17.1.4	Find/Replace in LD/FBD/ CFC .....	17-17
17.1.5	Find/Replace in IL/ST .....	17-20
17.1.6	Find/Replace in SFC .....	17-23
17.1.7	Find Function in Symbol Table.....	17-25

17.1.8	Find/Replace in Project .....	17-26
17.1.9	Print Function.....	17-27
<b>17.2</b>	<b>Devices and Registers .....</b>	<b>17-30</b>
17.2.1	Device Comment List for a DVP Series PLC .....	17-30
17.2.2	AH/AS Series Device Comment List.....	17-34
17.2.3	Using Device Report.....	17-40
17.2.4	Edit Register Memory .....	17-44
17.2.5	Edit Bit Memory.....	17-51
17.2.6	Edit DVP Series File Register .....	17-56
<b>17.3</b>	<b>Step Positioning .....</b>	<b>17-63</b>
17.3.1	Using Step Positioning.....	17-63
<b>17.4</b>	<b>Program Comparison.....</b>	<b>17-64</b>
17.4.1	Introduction to Program Comparison.....	17-64
17.4.2	Compare with File.....	17-65
17.4.3	Compare with PLC .....	17-67
 <b>Chapter 18 Testing and Debugging Tools</b>		
<b>18.1</b>	<b>Online Monitoring Function .....</b>	<b>18-2</b>
18.1.1	Changing the Operating Status of a PLC .....	18-2
18.1.2	Online Monitoring Function and Environment.....	18-2
18.1.3	Changing the States of the X Devices in the Online Mode.....	18-7
18.1.4	Monitoring a Program Online.....	18-8
18.1.5	Device Monitoring Table .....	18-15
18.1.6	Online Editing Function and Online Update Function .....	18-21
<b>18.2</b>	<b>Debugging Mode for DVP Series .....</b>	<b>18-24</b>
18.2.1	Enabling the Debugging Mode for DVP Series .....	18-24
18.2.2	Adding and Clearing Breakpoints .....	18-25
18.2.3	Execution of the Program in the Debugging Mode .....	18-25
<b>18.3</b>	<b>Debugging Mode for AH/AS Series .....</b>	<b>18-27</b>
18.3.1	Enter Debugging Mode .....	18-27
18.3.2	Adding and Clearing Breakpoints .....	18-27

18.3.3	Continuous Execution.....	18-29
18.3.4	Single-Step Execution .....	18-29
<b>18.4</b>	<b>Checking the Status of a PLC.....</b>	<b>18-31</b>
18.4.1	System Information.....	18-31
18.4.2	PLC System Log .....	18-33
<b>18.5</b>	<b>3D Chart.....</b>	<b>18-40</b>
18.5.1	Features.....	18-40
18.5.2	Creating a 3D Chart.....	18-40
18.5.3	Creating a Curve .....	18-41
18.5.4	Display a 3D Chart .....	18-43
 <b>Chapter 19 Password Management and Data Protection</b>		
<b>19.1</b>	<b>Password Protection Mechanisms Provided by ISPSoft.....</b>	<b>19-2</b>
<b>19.2</b>	<b>Program ID and PLC ID.....</b>	<b>19-3</b>
19.2.1	Setting and Unlocking a Program ID .....	19-3
19.2.2	Setting and Unlocking a PLC ID .....	19-4
<b>19.3</b>	<b>Project Password and PLC Password.....</b>	<b>19-5</b>
19.3.1	Setting and Unlocking a Project Password.....	19-6
19.3.2	Setting and Unlocking a PLC Password .....	19-8
19.3.3	Synchronize Project and PLC Password.....	19-9
<b>19.4</b>	<b>POU Password.....</b>	<b>19-10</b>
<b>19.5</b>	<b>Subroutine Password .....</b>	<b>19-11</b>
19.5.1	Introduction of a Subroutine Passwords.....	19-11
19.5.2	Setting and Unlocking a Subroutine Password .....	19-12
<b>19.6</b>	<b>Other Password and Data Protection Functions .....</b>	<b>19-13</b>
19.6.1	Disable Program Upload .....	19-13
19.6.2	Setting Read-only Area .....	19-14
19.6.3	Setting TC-01 Password Key .....	19-14

## Chapter 20 Network Configuration and Data Exchange

<b>20.1</b>	<b>Network Configuration Tool - NWCONFIG .....</b>	<b>20-3</b>
20.1.1	Introduction of NWCONFIG .....	20-3
20.1.2	Basic Knowledge.....	20-4
20.1.3	Communication Setting in NWCONFIG.....	20-6
20.1.3.1	Connection Mechanism in NWCONFIG.....	20-6
20.1.3.2	Setting Communication Parameters.....	20-8
20.1.4	Workflow .....	20-9
<b>20.2</b>	<b>Creating a Network Architecture .....</b>	<b>20-14</b>
20.2.1	Deploying Nodes.....	20-14
20.2.2	Connecting to a Network .....	20-18
20.2.3	Adjusting or Deleting Devices or Networks.....	20-23
20.2.4	Setting the Attributes of a Node/Network.....	20-26
20.2.5	Hiding/Displaying Devices or Networks .....	20-31
20.2.6	Correct Network Architecture .....	20-34
20.2.7	Downloading Routing Tables.....	20-37
20.2.8	Testing Routing .....	20-39
<b>20.3</b>	<b>Constructing a PLC Link.....</b>	<b>20-41</b>
20.3.1	Opening the PLC Link Table Editor Window.....	20-41
20.3.2	Select Master Station Device (Step 1) .....	20-43
20.3.3	Communication Parameter Settings (Step 2) .....	20-44
20.3.4	Create Data Exchange Table (Step 3) .....	20-46
20.3.4.1	Introduction of a Data Exchange Table.....	20-46
20.3.4.2	Setting a Data Exchange Group.....	20-48
20.3.4.3	Device Synchronization in Data Exchange Table .....	20-50
20.3.4.4	Managing Data Exchange Table.....	20-52
20.3.5	Monitoring a PLC Link.....	20-53
20.3.6	Notifications on PLC Link.....	20-58
<b>20.4</b>	<b>Constructing an Ether Link .....</b>	<b>20-59</b>
20.4.1	Introduction of an Ether Link.....	20-59
20.4.2	Open Ether Link Configuration.....	20-60

20.4.3	Create and Manage Data Exchange Table.....	20-62
20.4.4	Node List and Display Area .....	20-66
20.4.5	Start Mode of an Ether Link .....	20-68
20.4.6	Download Ether Link Configuration .....	20-70
20.4.7	Upload Ether Link Configuration .....	20-72
20.4.8	Deleting Asynchronous Device .....	20-74
20.4.9	Enable/Disable Online Monitoring Function.....	20-76
20.4.9.1	Enabling a Monitoring Function.....	20-77
20.4.9.2	Monitoring Status .....	20-79
20.4.9.3	Disabling a Monitoring Function.....	20-80
20.4.10	Online Start/Stop Ether Link (SM Flag).....	20-81
20.4.10.1	Starting the Execution of an Ether Link.....	20-81
20.4.10.2	Stopping the Execution of an Ether Link .....	20-83
20.4.11	Monitoring Table and Error Log .....	20-85
<b>20.5</b>	<b>NWCONFIG Management and Application .....</b>	<b>20-86</b>
20.5.1	Save and Print .....	20-86
20.5.2	Downloading.....	20-87
20.5.2.1	Downloading Parameters .....	20-87
20.5.2.2	Description of Downloading.....	20-87
20.5.3	ISPSOFT Routing Application .....	20-89
<b>Chapter 21 Data Backup and Data Restoration</b>		
<b>21.1</b>	<b>Data Backup Memory Cards.....</b>	<b>21-2</b>
21.1.1	Introduction of Data Backup Memory Cards .....	21-2
21.1.2	Operating Data Backup by Memory Card .....	21-3
<b>21.2</b>	<b>Permanent Data Backup.....</b>	<b>21-4</b>
<b>21.3</b>	<b>CARD Utility .....</b>	<b>21-5</b>
21.3.1	Introduction of CARD Utility .....	21-5
21.3.2	Backup.....	21-7
21.3.3	Restoration .....	21-12

## Chapter 22 G-Code Editor and E-CAM Editor

<b>22.1</b>	<b>G-Code Editor .....</b>	<b>22-2</b>
22.1.1	About G-Code.....	22-2
22.1.2	Using G-Code Editor .....	22-2
22.1.3	G-Code Command Structure.....	22-7
22.1.4	G-Code Functions.....	22-10
22.1.5	G-Code Example of Drawing Delta Logo through Three-Axis Motion .....	22-31
<b>22.2</b>	<b>E-CAM Editor .....</b>	<b>22-33</b>
22.2.1	About E-CAM (Electronic CAM) .....	22-33
22.2.2	The Significance and Description of E-CAM.....	22-33
22.2.3	Using E-CAM Editor .....	22-35

## Chapter 23 Wizard Tool

<b>23.1</b>	<b>Position Planning Table .....</b>	<b>23-2</b>
23.1.1	About the Position Planning Table .....	23-2
23.1.2	Setting the Position Planning Table.....	23-2
23.1.3	Positioning Table Simulation .....	23-6
23.1.4	Upload and Download Position Planning Table.....	23-8
23.1.5	Control Mode – Single axis point-to-point motion.....	23-9
23.1.6	Control Mode – Single axis multi-segment .....	23-10
23.1.7	Control Mode - 2-axis linear interpolation motion.....	23-11
23.1.8	Control Mode - 2-axis Arc Interpolation Motion .....	23-12
<b>23.2</b>	<b>Data Tracer .....</b>	<b>23-14</b>
23.2.1	About Data Tracer.....	23-14
23.2.2	Opening the Data Tracer Window .....	23-14
23.2.3	Sample Parameter Settings.....	23-16
23.2.4	Sampling Modes .....	23-19
23.2.5	Measurement .....	23-21
<b>23.3</b>	<b>Data Logger.....</b>	<b>23-25</b>
23.3.1	About Data Logger .....	23-25



23.3.2	Opening the Data Logger Window.....	23-25
23.3.3	Sample Parameter Settings.....	23-27
23.3.4	Watch and Record.....	23-29
23.3.5	Measurement.....	23-32
<b>23.4</b>	<b>High Speed Counter .....</b>	<b>23-36</b>
23.4.1	About High Speed Counter.....	23-36
23.4.2	Using High Speed Counter .....	23-36

## Appendix A USB Connection

<b>A.1</b>	<b>Installing the USB Driver for an AS Series CPU module.....</b>	<b>A-2</b>
A.1.1	Installing the USB Driver in Windows XP with SP3 .....	A-2
A.1.2	Installing the USB Driver in Windows 7 .....	A-6
A.1.3	Installing the USB Driver in Windows 8.1.....	A-11
A.1.4	Installing the USB Driver in Windows 10.....	A-14
<b>A.2</b>	<b>Create USB Driver in COMMGR .....</b>	<b>A-18</b>
<b>A.3</b>	<b>Setting the USB Port on a DVP-SX2 Series PLC.....</b>	<b>A-20</b>

## Appendix B Notification for PLC Types

<b>B.1</b>	<b>Device Addresses for PLC Types .....</b>	<b>B-2</b>
B.1.1	AH/AS Series Device Types .....	B-2
B.1.2	The X/Y/D/L Device Address Format in AH/AS Series.....	B-4
B.1.3	Real-time Access of X/Y Contact in AH/AS Series .....	B-6
B.1.4	DVPxxMC Device Types .....	B-7
B.1.5	DVPxxMC Series Device Address Format .....	B-8
B.1.6	DVPxxMC Retentive Devices .....	B-8
<b>B.2</b>	<b>Device Resources for PLC Types .....</b>	<b>B-9</b>
B.2.1	AHCPU500-EN/AHCPU500-RS2 .....	B-9
B.2.2	AHCPU510-EN/AHCPU510-RS2 .....	B-10
B.2.3	AHCPU520-EN/AHCPU520-RS2 .....	B-11
B.2.4	AHCPU530-EN/AHCPU530-RS2 .....	B-12
B.2.5	AHCPU501-EN/AHCPU501-RS2 .....	B-13

B.2.6	AHCPU511-EN/AHCPU511-RS2 .....	B-14
B.2.7	AHCPU521-EN/AHCPU521-RS2 .....	B-15
B.2.8	AHCPU531-EN/AHCPU531-RS2 .....	B-16
B.2.9	AHCPU560-EN2 .....	B-17
B.2.10	AHxxEMC .....	B-18
B.2.11	AS200/AS300 .....	B-19
B.2.12	DVPxxMC .....	B-20
<b>B.3</b>	<b>Compile &amp; Uploading/Downloading Time .....</b>	<b>B-20</b>
<b>Appendix C Print Management Tool</b>		
<b>C.1</b>	<b>Introduction of the Environment .....</b>	<b>C-2</b>
<b>C.2</b>	<b>Introduction of the Setting Area .....</b>	<b>C-4</b>

## 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 seven programming languages for programmable control systems: instruction lists (ILs), structured texts (STs), ladder diagrams (LDs), sequential function charts (SFCs), and function block diagrams (FBDs), C the programming language and continuous function charts (CFCs).
- 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 by 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 8 introduce the concepts related to IEC61131-3, and the ways in which the concepts are applied to ISPSOft.
- Chapter 9 introduces the relative concepts of motion control and axis and the ways in which the concepts are applied to ISPSOft.
- Chapter 10~chapter 16 introduce ladder diagrams (LDs), function block diagrams (FBDs), instruction lists (ILs), structured texts (STs), sequential function charts (SFCs), and continuous function charts (CFCs), the C programming language and the ways in which the concepts are applied to ISPSOft.
- Chapter 17 introduces the auxiliary tools provided by ISPSOft.
- Chapter 18 is about monitoring a program, testing a program, and debugging a program through ISPSOft.
- Chapter 19 introduces the password setting, and the data protection mechanism in ISPSOft.
- Chapter 20 is about configuring a network through ISPSOft, and creating a data exchange mechanism.
- Chapter 21 is about backing up data and restoring data through ISPSOft.
- Chapter 22 introduces the tools used commonly in motion control through ISPSOft.
- Chapter 23 introduces auxiliary tools and wizards provided by ISPSOft for easier programming and better operation.

### **Graphic representations**

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

<b>Graphic representation</b>	<b>Description</b>
	Clicking the left mouse button
	Clicking the right mouse button
	Clicking the left mouse button two times in quick succession
	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 to describe a figure
	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 Introducing ISPSoft

## Table of Contents

<b>1.1</b>	<b>Introduction of ISPSoft and System Requirements.....</b>	<b>1-2</b>
1.1.1	Characteristics.....	1-2
1.1.2	System Requirements.....	1-3
1.1.3	Installing ISPSoft .....	1-4
1.1.4	Uninstalling ISPSoft.....	1-8
<b>1.2</b>	<b>Introduction of COMMGR .....</b>	<b>1-10</b>
1.2.1	Operating Mode of COMMGR.....	1-10
1.2.2	Installing COMMGR .....	1-11
1.2.3	Uninstalling COMMGR .....	1-13
<b>1.3</b>	<b>ISPSOft for PLC Devices .....</b>	<b>1-14</b>
1.3.1	ISPSOft - List of Programming Languages.....	1-14
1.3.2	ISPSOft - List of Functions .....	1-15

# 1

## 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 seven programming languages and adopts a large number of applied instructions. 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 seven programming languages. They are ladder diagrams (LD), sequential function charts (SFC), function block diagrams (FBD), instruction lists (IL), and structured texts (ST), continuous function chart (CFC) and C language (C). Users can use more than 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 online operation such as monitoring programs online, editing programs online, monitoring devices online, debugging programs online, operating/setting a PLC online, 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 mechanisms and data protection mechanisms.
- It supports COMMGR, a new generation communication manager.
- There are three built-in configurations.
  - HWCONFIG: It is used to configure hardware for a system, and manage parameters.
  - NWCONFIG: It is used to configure networks for a PLC system, and manage data exchanges.

- CARD Utility: Users can backup and restore a system through a management wizard and a memory card.
- It provides various solutions for motion control including PLCopen MC function block, G-code editor, E-CAM editor, positioning planning chart tool and many more.
- The software can be applied to DVPxxMC series including DVP15MC, DVP15MC-06, DVP50MC and DVP50MC-06.

### 1.1.2 System Requirements

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

Item	System requirement	
<b>Operating system</b>	Windows XP / 7 / 8 / 10	
<b>CPU</b>	Pentium 1.5 G or above	
<b>Memory</b>	256 MB or above (A memory having a capacity of 512 MB or above is	
<b>Hard disk drive</b>	Capacity : 1000 MB or above	
<b>CD-ROM drive</b>	For installing ISPSOft It is optionally required.	
<b>Monitor</b>	Resolution: 800x600 pixels or above (Recommended setting: 1024x768/96 dpi )	
<b>Keyboard/Mouse</b>	General keyboard/mouse, or device compatible with Windows	
<b>Printer</b>	Printer with a driver for Windows (It is used to print projects, and is optionally required.)	
<b>RS-232 port</b>	For connecting to a PLC	Users have to select one of them according to the communication interfaces provided by the PLC or the module used. (*1)
<b>USB port</b>	For connecting to a PLC	
<b>Ethernet port</b>	For connecting to a PLC	
<b>Communication software</b>	COMMGR, a communication manager, must be installed on a computer. (*2)	
<b>Models which are supported</b>	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 series AC motor drives /VFD-E series AC motor drives	



1

Item	System requirement
	(*3) PLC: all AH series, AS series and DVP series (except DVP-PM series) AC motor drive: VFD series with PLC built-in Text panel: TP series with PLC built-in

\*1. ISPSOft supports several ways in which a computer is connected to a PLC. Users have to make sure of the ports and the modes supported by a PLC before a computer is connected to the PLC.

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

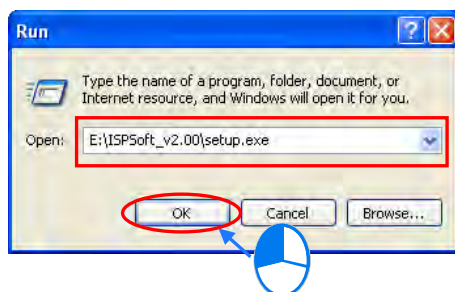
\*3. In addition to ISPSOft, users must use PMSOft version 2.05 or above to develop AH10PM-5A and AH20MC-5A.

\*4. The functions and specifications mentioned above are only applicable to ISPSOft version 2.00 or above. The older versions may not be equipped with complete functions.

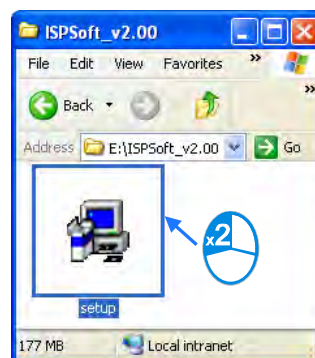
### 1.1.3 Installing ISPSOft

When the previous version of ISPSOft is detected in a computer, that version is advised to be uninstalled first before the latest ISPSOft can be installed.

- (1) Start a computer and enter the operating system. Users have to log on to the system as a system administrator before installing ISPSOft.
- (2) Put an ISPSOft CD in the CD-ROM drive, or download the installation program from the official Delta website <http://www.deltaww.com/> to download ISPSOft. (The installation programs need to be decompressed if downloaded from the internet.)
- (3) Click **Start**, and **Run...** to open the **Run** window. Specify the path denoting the executable file which is used to install COMMGR in the **Open** box, and then click **OK**. Alternatively, users can double-click the icon which is used to install ISPSOft to execute the installation program.

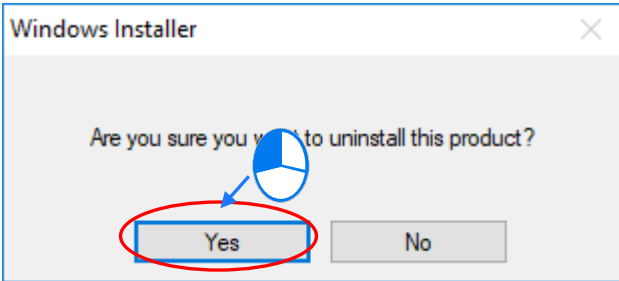
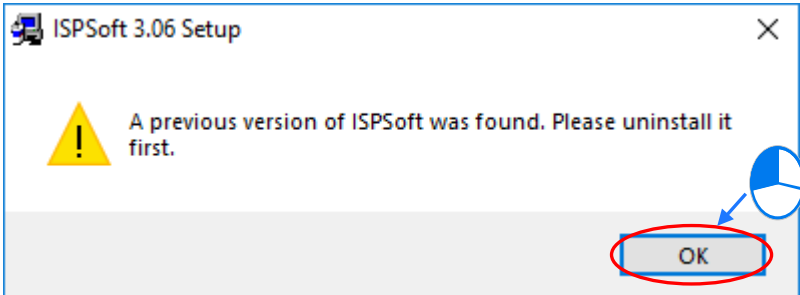


OR

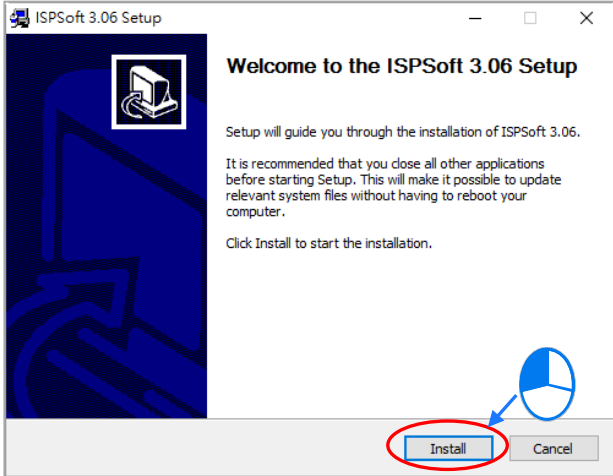


- (4) When a previous version of the ISPSOft is found, click **OK** then **Yes** to uninstall that version shown in the pop-up windows (see below).

1

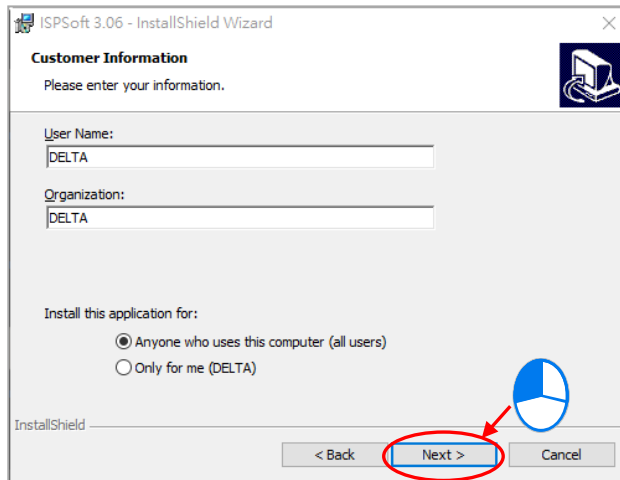


- (5) Click **Install** once Shield Wizard window appears.

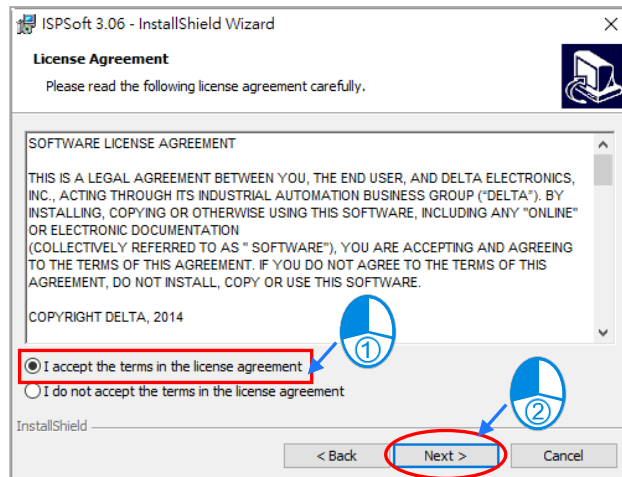


1

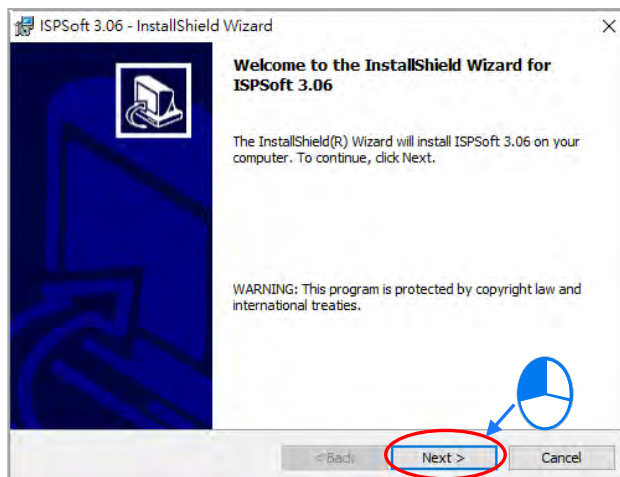
- (6) Type related information in the **User Name** box and the **Organization** box, and then click **Next**.



- (7) Choose **I accept the terms in the license agreement**. Click **Next** to proceed to the next step.

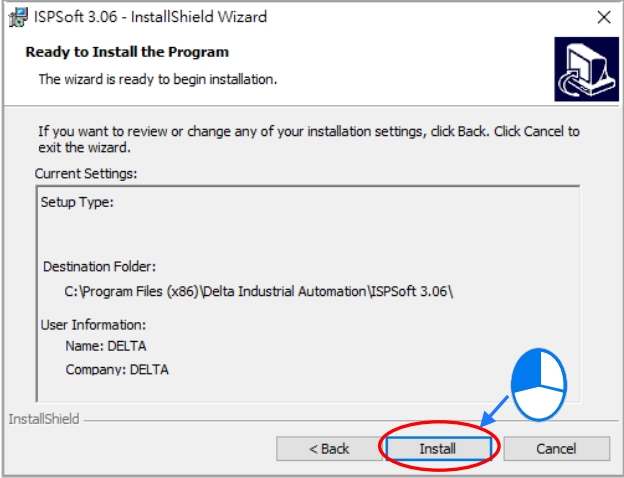


- (8) Then, click **Next** for the next step.

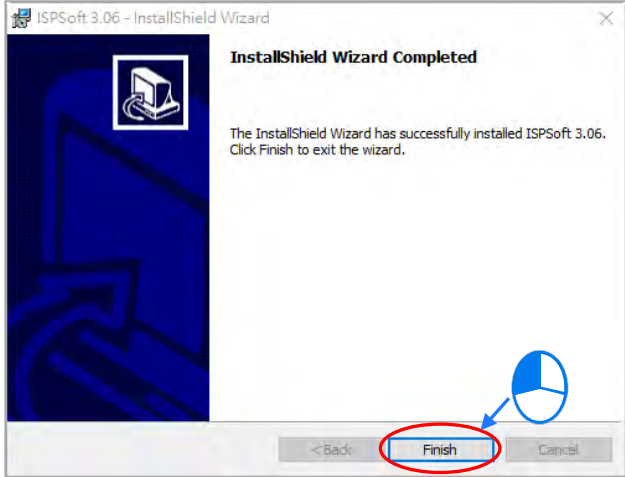
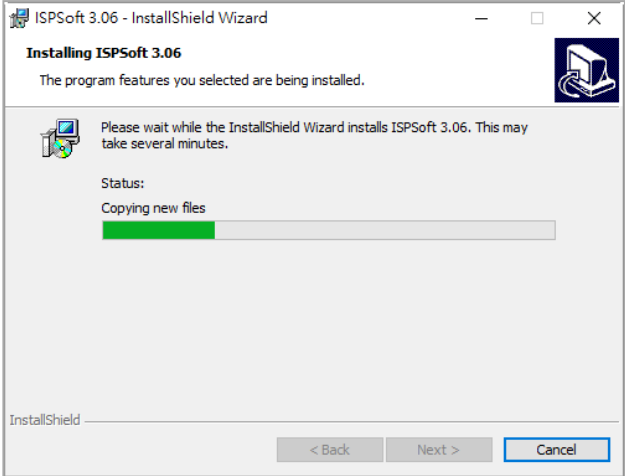


(9) Check the installation information, and then click **Install**.

1

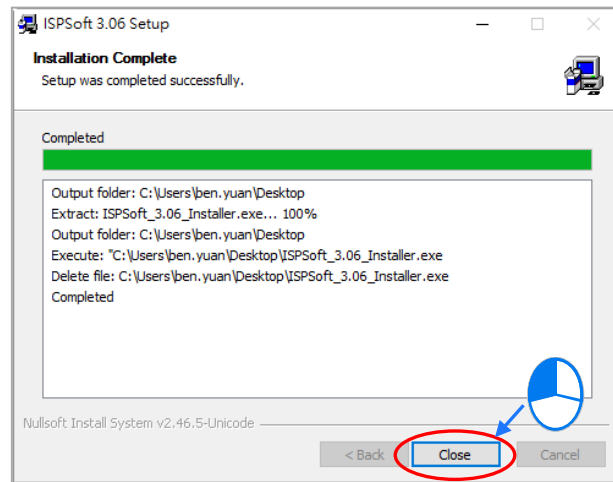


(10) After installation is complete, click **Finish** to continue the next step.



1

(11) When installation is complete, shortcuts to the software is created on the desktop and Start menu. Click **Close** to exit the setup.

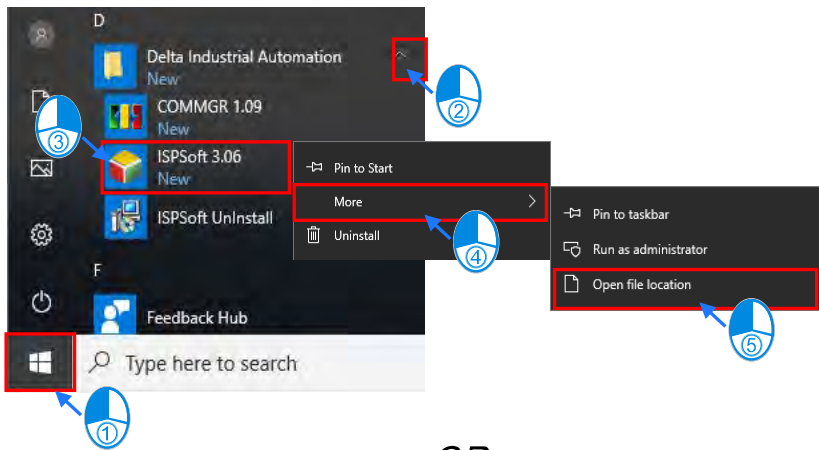


### 1.1.4 Uninstalling ISPSOft

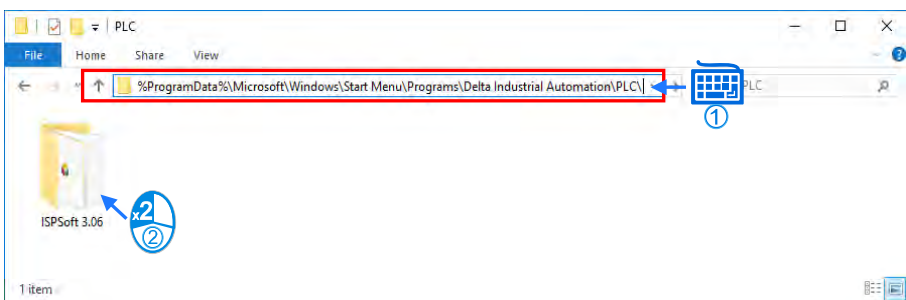
(1) Generally, users can click **ISPSOft Uninstall** or choose **Programs** under **Control Panel** to remove the ISPSOft; when **ISPSOft Uninstall** is not found, there are two methods to uninstall the software:

- Method 1: Choose **ISPSOft x.xx** from the Windows list, click **More** then select **Open file location**.
- Method 2: Place **%ProgramData%\Microsoft\Windows\Start Menu\Programs\Delta Industrial Automation\PLC\** in the address box and press Enter. Then, double click ISPSOft x.xx file.

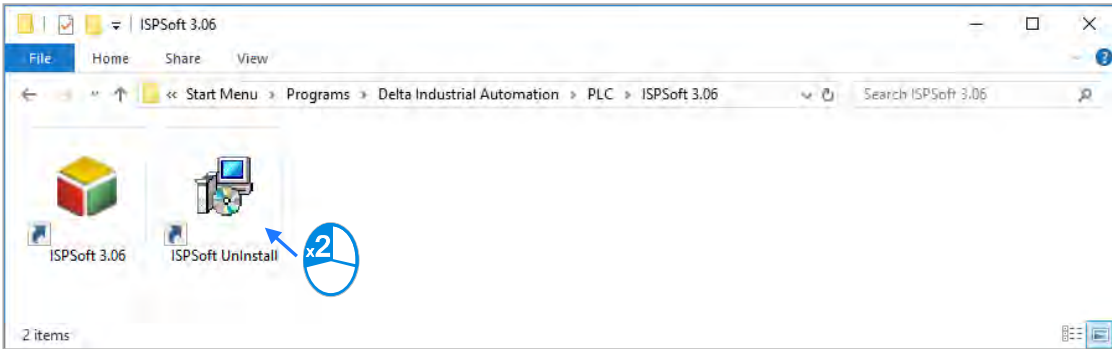
1



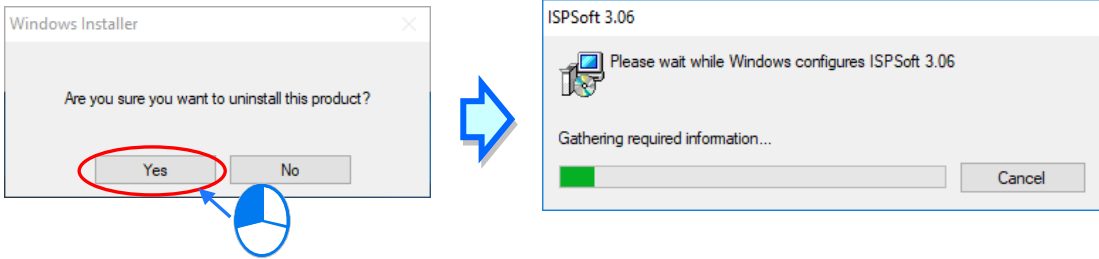
OR



(2) Remove the software by double-click the **ISPSOft UnInstall**.



(3) To uninstall ISPSOft, click **Yes** shown in the pop-up window. The window will automatically close once the software is removed.



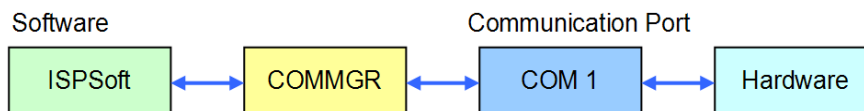
## 1

## 1.2 Introduction of COMMGR

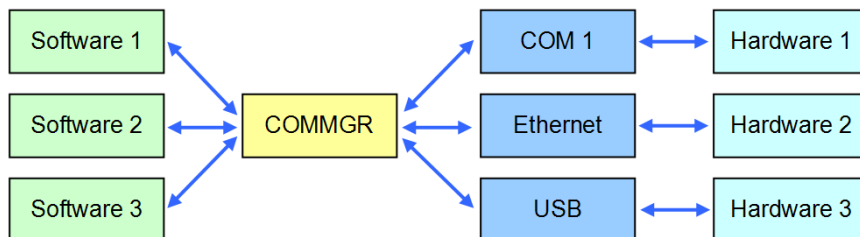
COMMGR is a new generation communication management tool developed by Delta Electronics, Inc. in 2011. It functions as a communication bridge between Delta software and hardware. Communication becomes more convenient and more efficient through the management of COMMGR.

### 1.2.1 Operating Mode of COMMGR

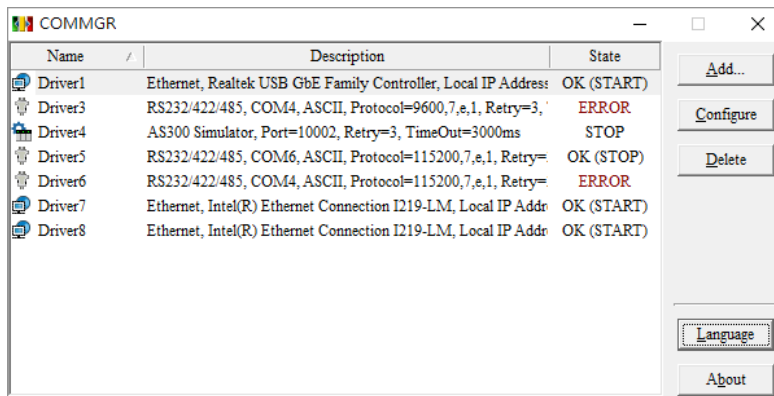
Users can create communication parameters which must be set on the management list in COMMGR in advance. The communication parameters which have been created in advance are called drivers. Users can start or stop a driver in COMMGR. If a driver is started, a connection will be created automatically.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



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



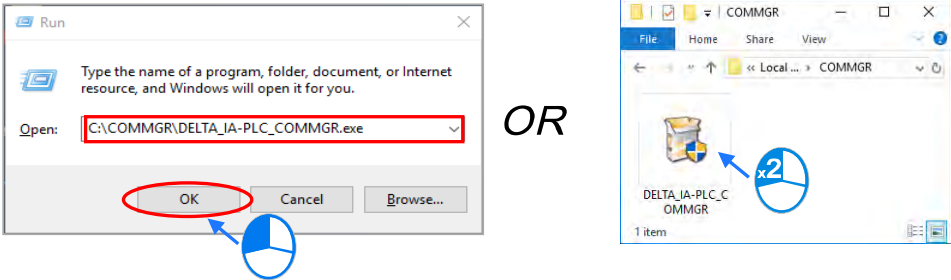
The **COMMGR** window and the management list in the **COMMGR** window are shown below. The drivers which are named by users are displayed in the **Name** column, parameters related to the drivers are displayed in the **Description** column, and the statuses of the drivers are displayed in the **Status**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



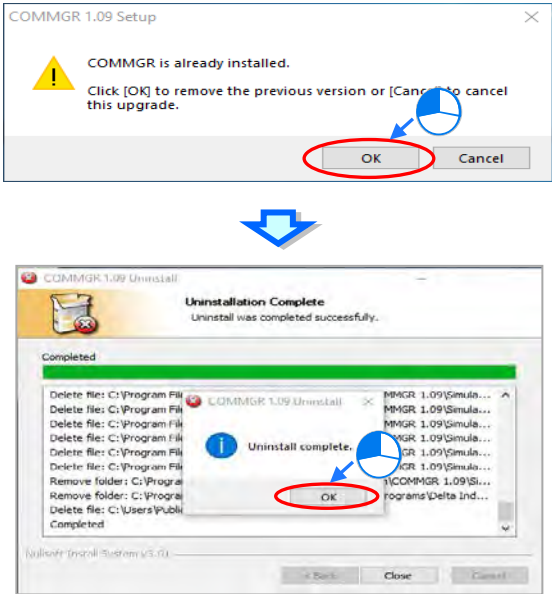
### 1.2.2 Installing COMMGR

COMMGR is a software independent of ISPSOft. It must be installed separately. When the previous version of COMMGR is detected in a computer, that version is advised to be uninstalled first before the latest COMMGR can be installed.

- (1) Start a computer and enter the operating system. Users have to log on to 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 the official Delta website <http://www.deltaww.com/>. (The installation programs need to be decompressed if downloaded from the internet.)
- (3) Click **Start**, and then click **Run...** to open the **Run** window. Specify the path denoting the executable file which is used to install COMMGR in the **Open** box, and then click **OK**. Alternatively, users can double-click the icon which is used to install COMMGR to execute the installation program.



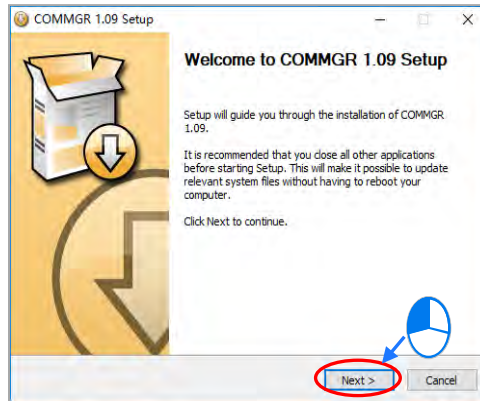
- (4) When the previous version of COMMGR is installed, click **OK** to remove that version shown in the pop-up window (see below) and when uninstall is complete, click **OK** again.



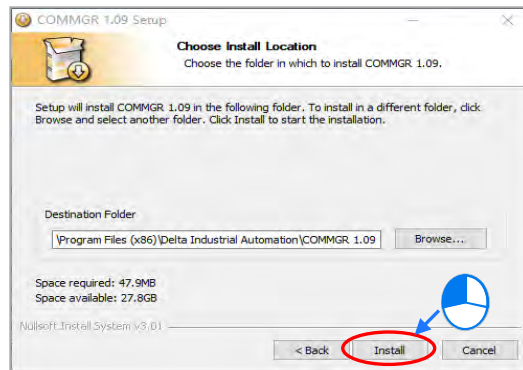


1

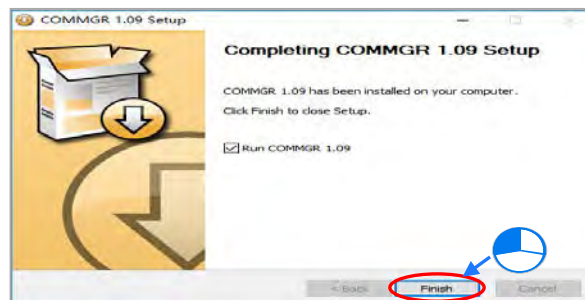
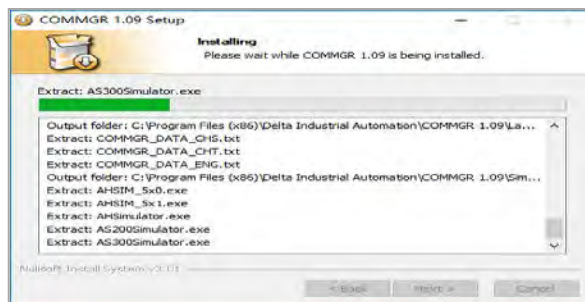
(5) Click **Next** after the Setup window appears.



(6) Use default setup in the destination folder. Click **Install** to start the installation.

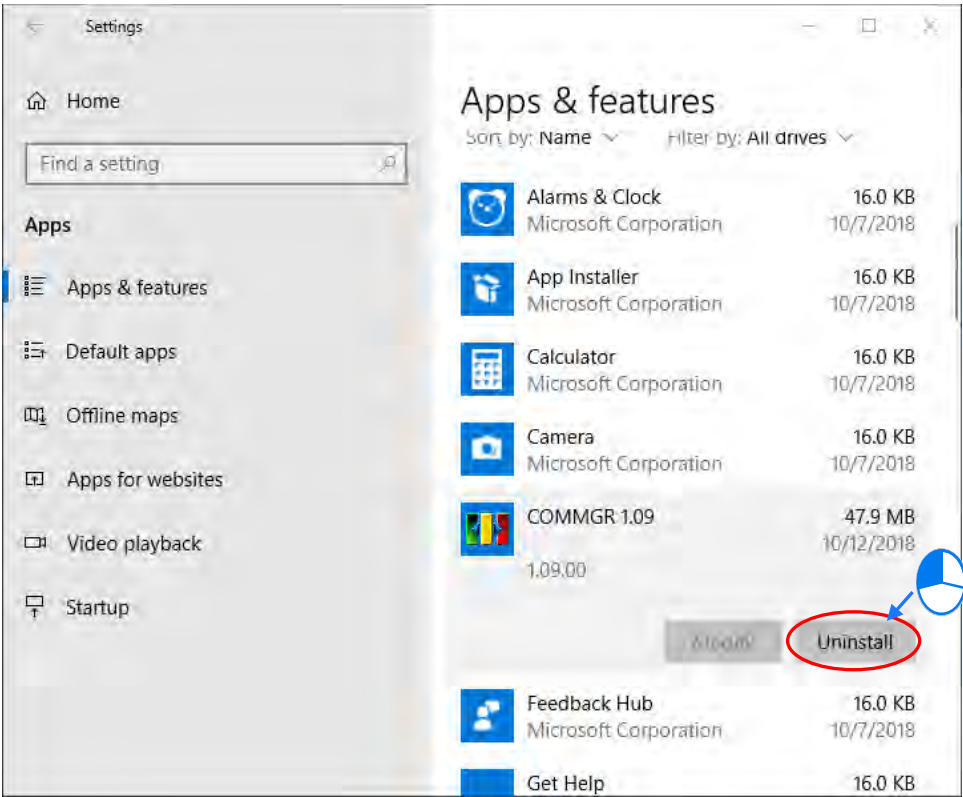


(7) When the installation is complete, the shortcut for COMMGR is created on the Start menu, click **Finish** to close the setup.

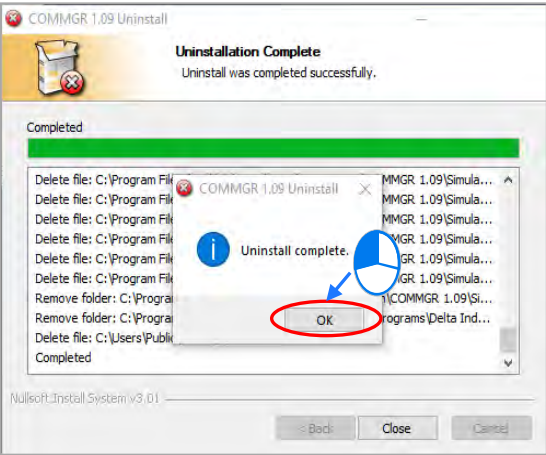
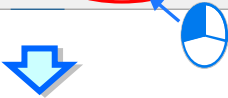
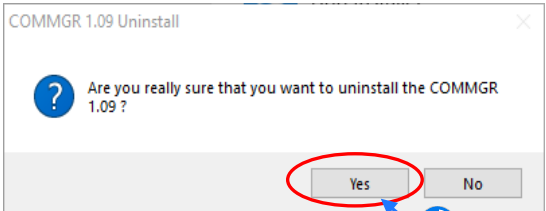


### 1.2.3 Uninstalling COMMGR

(1) Enter the settings of **Apps & features** in Windows, select **COMMGR x.xx** and click **Uninstall**.



(2) Click **Yes** then **OK** to complete COMMGR uninstallation.



## 1

## 1.3 ISPSOft for PLC Devices

The ISPSOft supports a variety of Delta PLC devices with many functions included. This chapter provides information regarding the ISPSOft that is applied in 8 PLC series: AS200, AS300, AH5x0, AH5x1, AH560, AHxxEMC, DVP, DVPxxMC series.

### 1.3.1 ISPSOft - List of Programming Languages

It supports seven programming languages. They are ladder diagrams (LD), sequential function charts (SFC), function block diagrams (FBD), instruction lists (IL), and structured texts (ST), continuous function chart (CFC) and C language (C). Users can use more than one programming language in one project.

	AS200	AS300	AH5x0	AH5x1	AH560	AHxxEMC	DVP	DVPxxMC
Ladder Diagram (LD)	V	V	V	V	V	V	V	V
Function Block Diagram (FBD)			V	V	V			
Instruction List (IL)			V	V	V		Do Not Support ES3	
Structured Text (ST)	V	V	V	V	V	V	Only Supports ES3	V
Sequential Function Charts (SFC)	V	V	V	V	V	V	V	
Continuous Function Chart (CFC)	V	V				V	Only Supports ES3	
C Language (C)	V	V						

### 1.3.2 ISPSOft - List of Functions

The following table shows major functions of ISPSOft.

	AS200	AS300	AH5x0	AH5x1	AH560	AHxxEMC	DVP	DVPxxMC
Motion Module			V	V	V	V		
HWCONFIG	V	V	V	V	V	V	Supports ES3 ONLY	V
HWCONFIG Printing Function	V	V	V	V	V	V		
Ordered List	V	V	V	V	V	V		
Custom Library	V	V	V	V	V	V	V	
Delta Library	V	V	V	V	V	V	V	V
User-defined	V	V	V	V	V	V	V	V
Axis						V		V
PLC Modification	V	V	V	V	V	V	V	V
Device Comment List	V	V	V	V	V	V	V	
Status Bar	V	V	V	V	V	V	V	
Edit Register	V	V	V	V	V	V	V	
Status Editing	V	V	V	V	V	V	V	
File Registers	V	V					Supports SA/SX/SC/E H/EH2/SV/E H2-L/ES2/ES 2-E/EX2/SX2 /SA2/EH3/SV 2/EH3-L/ES3	
Position-Step	V	V	V	V	V	V	V	
Alignment	V	V	V	V	V	V	V	
Online Editing	V	V	V	V	V	V	Supports EH3/EH3-L/S V2/ES2/ES2- E/EX2/SA2/S X2/MC/SS2	
Troubleshooting	V	V	V	V	V	V	v	
System Log	V	V	V	V	V	V	Supports ES3 ONLY	V
3D Coordinate						V		



1

	AS200	AS300	AH5x0	AH5x1	AH560	AHxxEMC	DVP	DVPxxMC
Identifier	V	V	V	V	V	V	v	V
Password (case-sensitive)	V	V	V	V	V	V	v	V
POU Password	V	V	V	V	V	V	v	V
Subroutine Password							Do Not Support ES3	
NWCONFIG			v				Do Not Support ES3	
Data Backup Memory Card (Supplementary)							Supports EH3/EH2	
PLC Permanent Backup Setting							Supports EH2/EH2-L/EH3/EH3-L/SV/SV2	
CARD Utility	V	V	V	V	V	V		
G-Code Editor								V
E-CAM Editor	V	V				V		V
Position Planning Table	V	V					Supports ES3 ONLY	
Oscilloscope	V	V		V		V	Supports ES3 ONLY	
Data Logger	V	V		V			Supports ES3 ONLY	
High-speed Counter Setting Wizard	V	V					Supports ES3 ONLY	
AIO Wizard Setting							Do Not Support ES3	
Temperature Control							Do Not Support ES3	
Weighing Module							Do Not Support ES3	
Extension Module Wizard							Do Not Support ES3	
Extension Module-Monitoring Wizard							Do Not Support ES3	

# Chapter 2 Starting and Setting I SPSoft



## Table of Contents

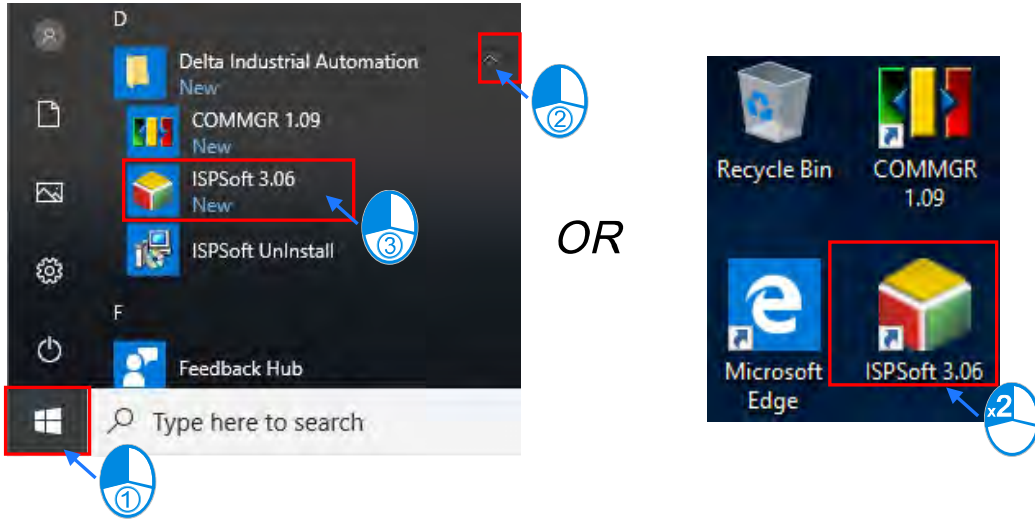
<b>2.1 Guidelines and Environment .....</b>	<b>2-2</b>
2.1.1 Getting Started.....	2-2
2.1.2 Window Title and Status Bar .....	2-6
2.1.3 Toolbar - Functions.....	2-7
2.1.4 Toolbar - Icons .....	2-10
2.1.5 Project and Compile Message .....	2-11
2.1.6 Workspace .....	2-13
<b>2.2 Project Framework.....</b>	<b>2-15</b>
2.2.1 Single Project and Group Project .....	2-15
2.2.2 Integration of Motion Control Modules .....	2-16
2.2.3 Managing a Single Project .....	2-17
2.2.4 Managing Group Project.....	2-22
2.2.5 Managing Motion Control Modules .....	2-28
<b>2.3 Basic Configuration .....</b>	<b>2-32</b>
2.3.1 System and Environment .....	2-32
2.3.2 Importing and Exporting User Preference .....	2-38
<b>2.4 Communication Settings .....</b>	<b>2-39</b>
2.4.1 Start/Close COMMGR .....	2-39
2.4.2 Driver Management for COMMGR .....	2-41
2.4.3 Creating Connection Channel - Add Driver.....	2-41
2.4.3.1 Communication Parameter Setting - RS232/422/485.....	2-43
2.4.3.2 Communication Parameter Setting - USB (Virtual COM) .....	2-44
2.4.3.3 Communication Parameter Setting - DirectLink (USB) .....	2-45
2.4.3.4 Communication Parameter Setting - Ethernet .....	2-45
2.4.3.5 Communication Parameter Setting - DirectLink (Ethernet) .....	2-46
2.4.3.6 Communication Parameter Setting - DVP Simulator.....	2-47
2.4.3.7 Communication Parameter Setting - AH5x0 & AH5x1 Simulator ..	2-48
2.4.3.8 Communication Parameter Setting - AS Simulator.....	2-48
2.4.4 Creating Connection Channel - Start/Stop a Driver .....	2-49
2.4.5 Creating Connection Channel - Configure/Delete a Driver.....	2-50
2.4.6 Creating Connection Channel - Simulator with Operating Panels....	2-51
2.4.7 Creating Connection between I SPSoft and COMMGR .....	2-54
2.4.8 Connecting PLC (Host) and Communication Port .....	2-57
2.4.9 Practical Connection Test.....	2-59

## 2.1 Guidelines and Environment

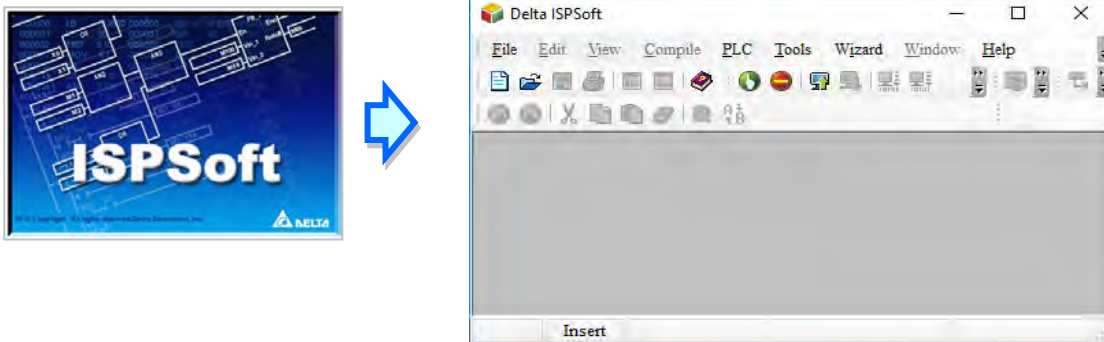
### 2.1.1 Getting Started


After installing ISPSoft, shortcuts is created on your desktop and **Start** menu for quick access of the software. In addition, users can install ISPSoft by taking the following steps below .

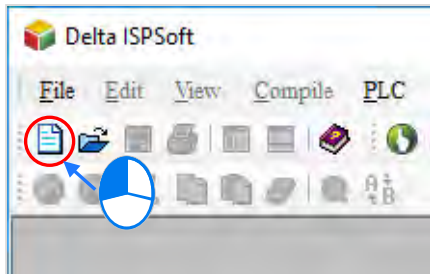
2



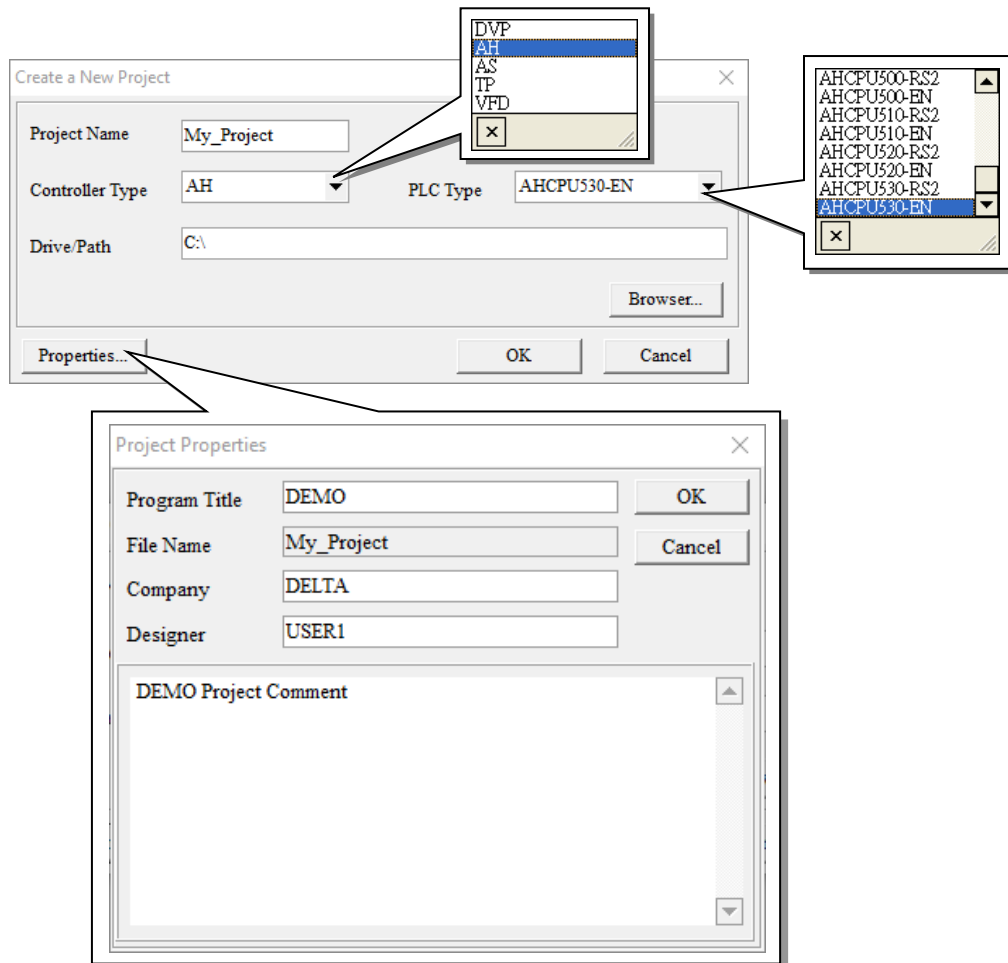
After the welcome screen image, the **Delta ISPSoft** window appears with basic functions provided.





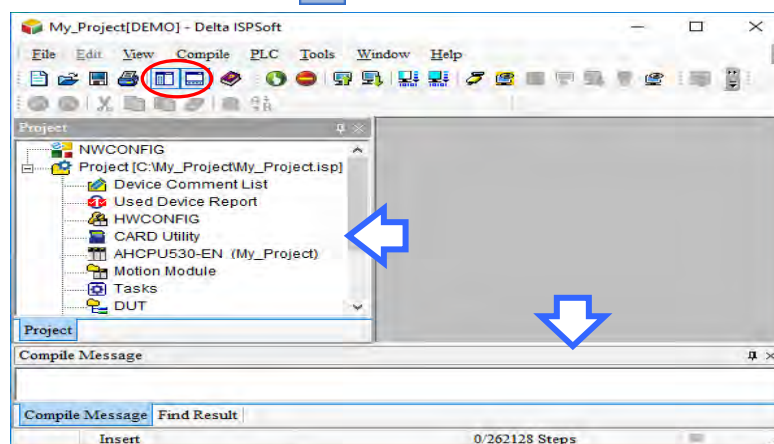
Click  on the toolbar to create a new project.



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



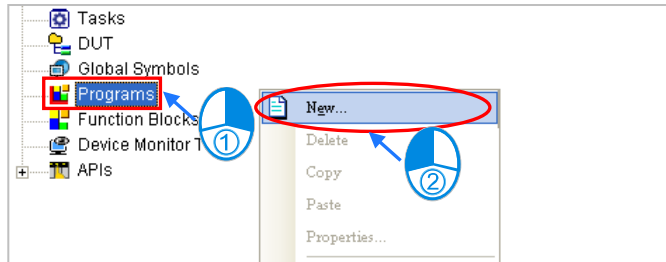
After the project is created successfully, a Project section appears on the left side of the dialog box with items listed in a hierarchical tree structure. If the section did not appear, users can click **View** on the toolbar, then choose **Workspace** or click  from the icon bar. To view Compile Message section, select **View > Output Window (M)** or click .



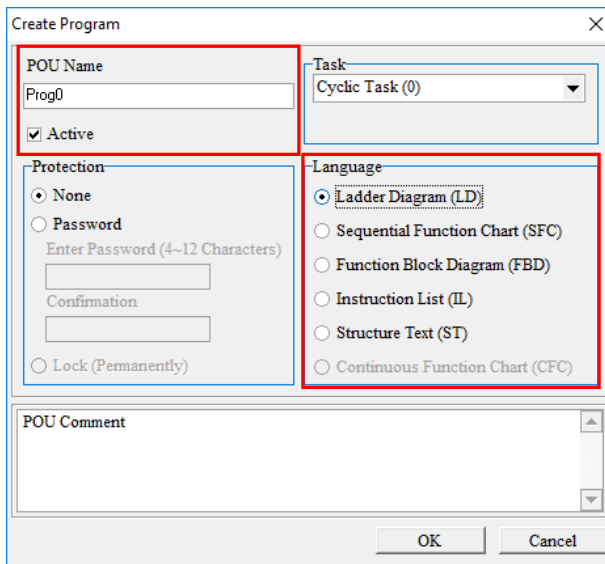


When a project is created, the main working area is blank with no editing. To start PLC programming, users can right-click **Programs** in the Project section and click **New...**

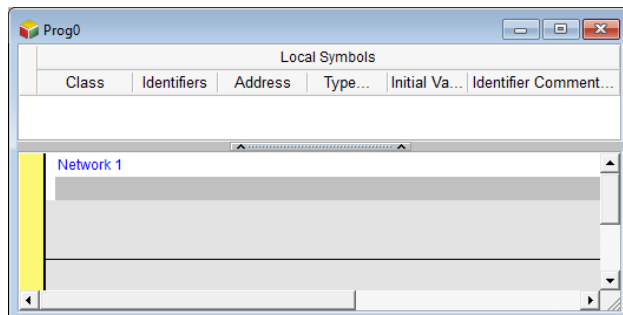
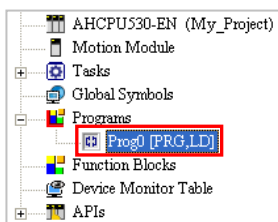
2



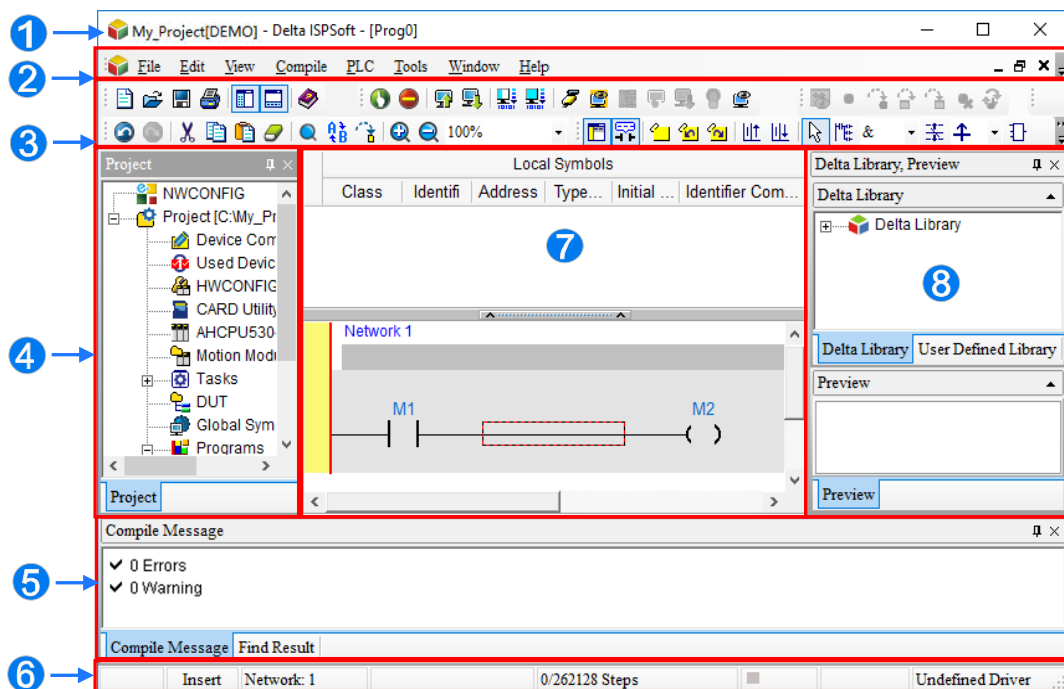
In the **Create Program** window, users can type a program name in the **POU Name**, and select a programming language in the **Language** section. Other parameters in the window use defaults and the description is provided in section 5.4.



When a program is added under **Programs**, an editing area is formed in the new program window.

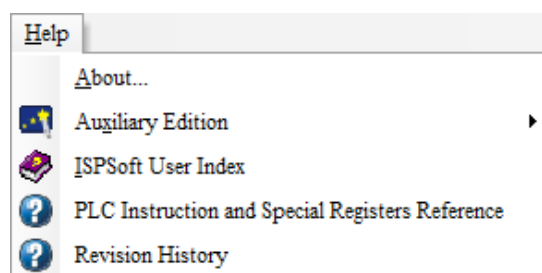


The ISPSOft main screen is shown below.




- ① Window title: It displays a project name and a program title.
- ② Menu bar: There are eight menus.
- ③ Toolbar: There are five toolbars.
- ④ Project: Uses hierarchical tree structure to manage the section.
- ⑤ Compile Message: Shows compiling result and project search.
- ⑥ Status bar: Displays current edit or network status.
- ⑦ Work Edit: Includes program edit section, local symbols, monitoring chart a device table, and etc. are displayed in this area.
- ⑧ Delta Library: Lists out contents regarding Delta library and user-defined library.

When using ISPSOft, users can click **Help** from the toolbar to get help.

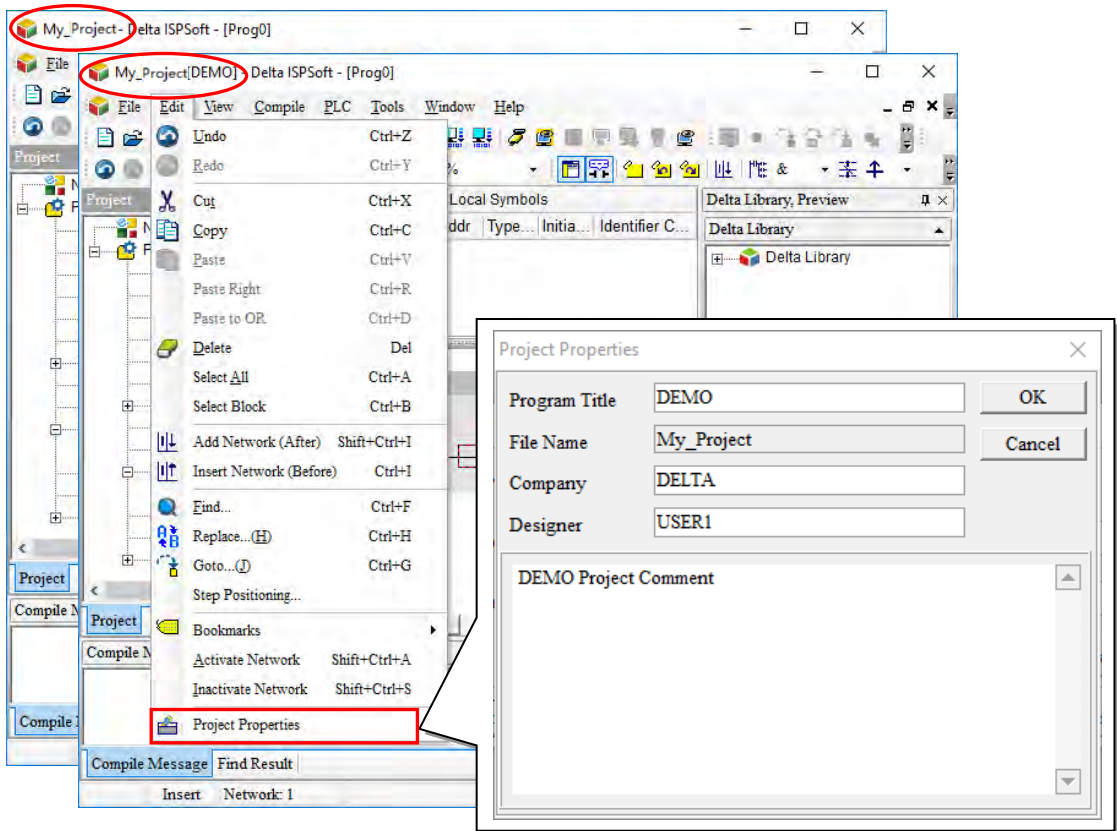


2

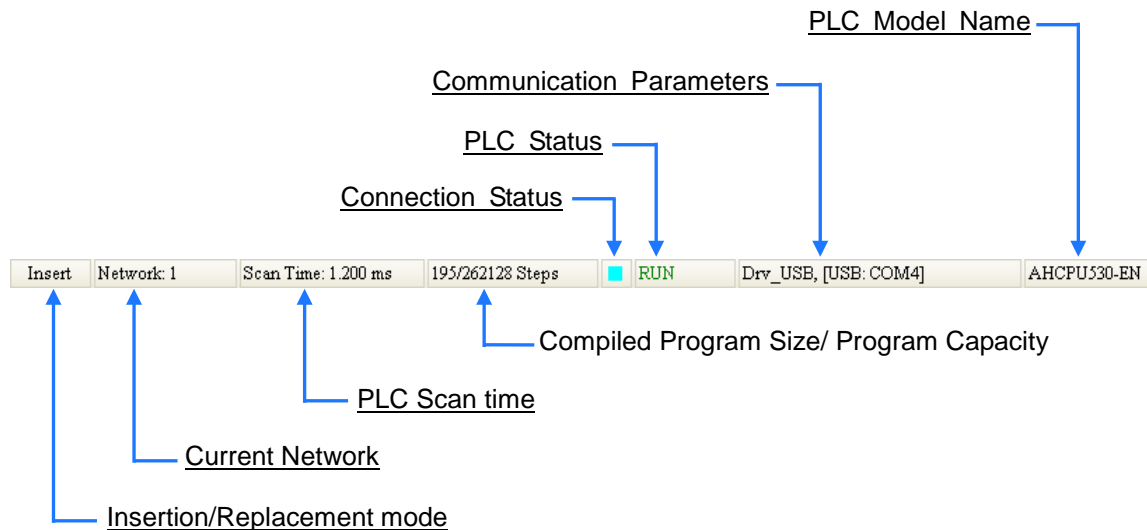
Item	Description
About...	Information about software version and date of release.
Auxiliary Edition	Auxiliary edit tools, e.g. communication format < = > parameter code.
ISPSOft User Index	Contains software usage information. (Alternatively, users can click  on the toolbar to view the information.)
PLC Instruction and Special Registers Reference	Information on instructions and registers. Users can click any instruction in the program edit section and press <b>F1</b> on the keyboard for explanations.
Revision History	Lists out the software version updates or modifications.

### 2.1.2 Window Title and Status Bar

After a project is created, the window title displays in the format of **project name [program title]**. If there is no program title, the project name is shown as the window title. Users can also click **Edit (E)** and select **Project Properties** to view the program title.



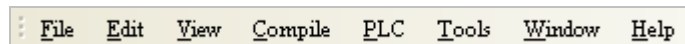
The status bar at the bottom of a window displays all kinds of working status. For example, edit mode (insertion/replacement), current network, PLC scan time, compiled program size or program capacity, connection status, PLC status, communication parameters, and PLC model name.



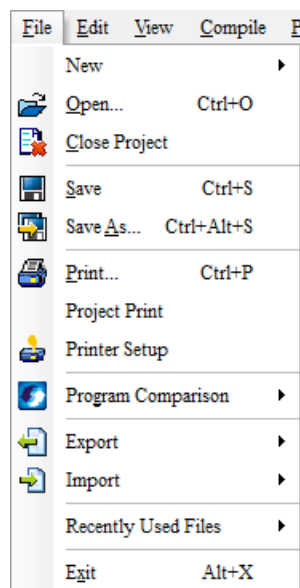
2

### 2.1.3 Toolbar - Functions

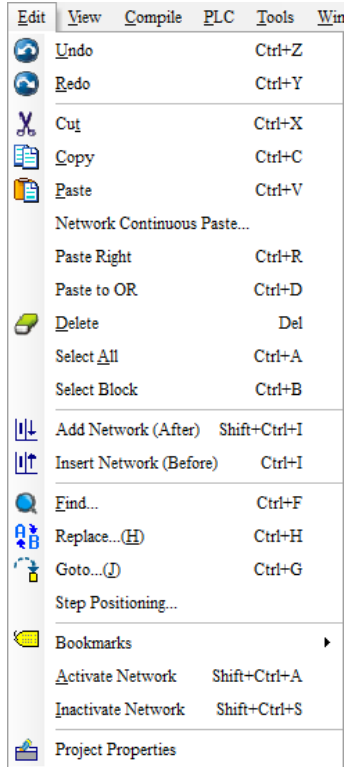
The ISPSOft toolbar contains eight functions. The contents of the functions may vary based on the editing and model selected. This chapter presents a brief introduction of the toolbar. For more explanation, please view the following chapters concerning the functions.



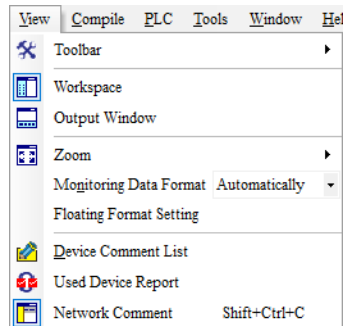
- **File:** To save a project as its main function.



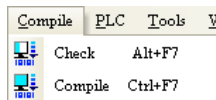
- **Edit:** Lists out the operating tools used for editing.



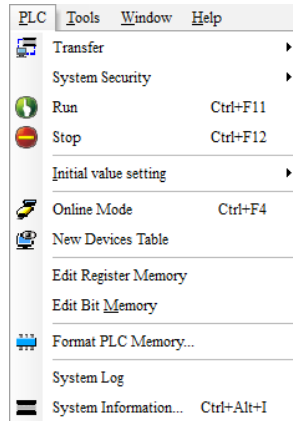
- **View:** Offers information regarding the project and workspace.



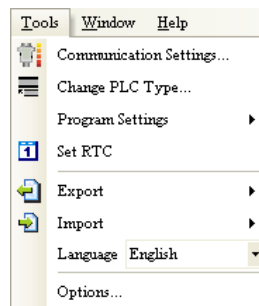
- **Compile:** Check programming syntax or compile programs into executable code.



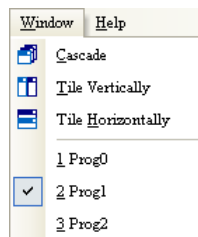
- **PLC:** Sets up a PLC connection and functions through ISPSOft.



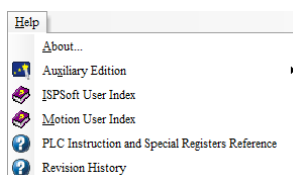
- **Tools:** Provides convenient functions to execute ISPSOft.



- **Window:** Manages the windows regarding the Edit section in workspace.



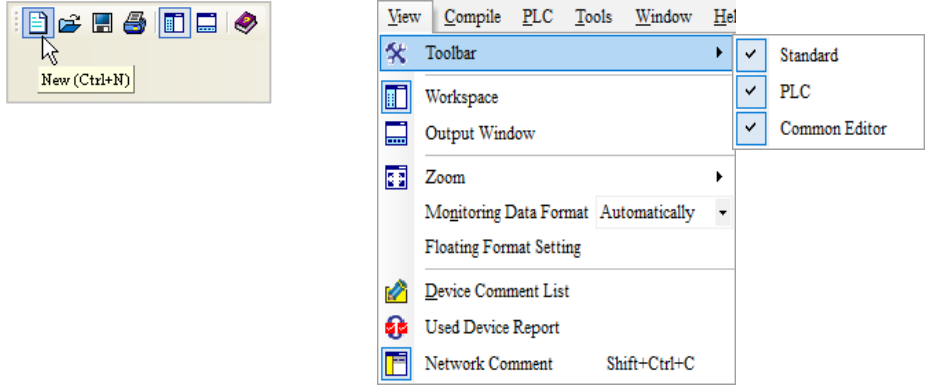
- **Help:** Provides auxiliary functions for ISPSOft.



### 2.1.4 Toolbar- Icons

There are five icon categories on the toolbar. The icons may also vary based on the editing and model selected. When a mouse cursor points at an icon for a short while, its function and keyboard shortcuts is shown; users can select their icons to show or hide under **Toolbar** by clicking **View**.

2



- **File toolbar:** Provides functions related to managing a project.



- **Editing toolbar:** Provides functions related to editing work.



- **Quick PLC toolbar:** Provides functions related to PLC operation.



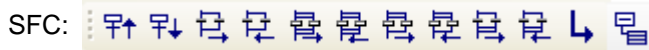
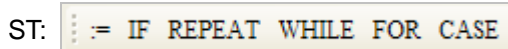
- **Troubleshooting toolbar:** Provides functions related to troubleshooting. The icons in the toolbar may vary based on the model selected.

\*. The DVPxxMC currently does not support this function.



- **Programming toolbar:** Provides functions related to programming. Icons may vary based on the programming language used.





2

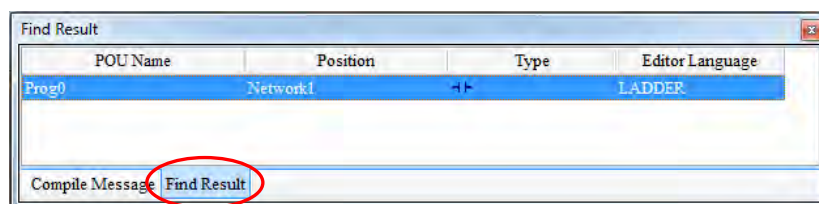
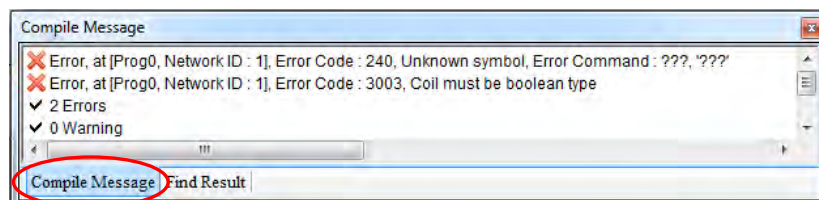
### 2.1.5 Project and Compile Message

The Project section contains contents related to project development and uses hierarchical tree structure to manage its interface. The section includes tools for configuration, device information and instruction application, user-defined variables, program items and monitoring tables. In addition, the lists in the section may vary based on the model selected.



On the left of the ISPSOft screen is the default Project section and click on the upper right corner to close the section. Users can click or select **Workspace** under **View** from the toolbar to open or close the Project window. When users right-click an item in the Project section, a corresponding quick menu offers items for selection.

In **Compile Message** section, users can click the tabs referring to the **Compile Message** and the **Find Result**. The **Compile Message** page shows the results of compiled programs or messages when checking on the syntax. While the **Find Result** page lists out the results on the searched projects.

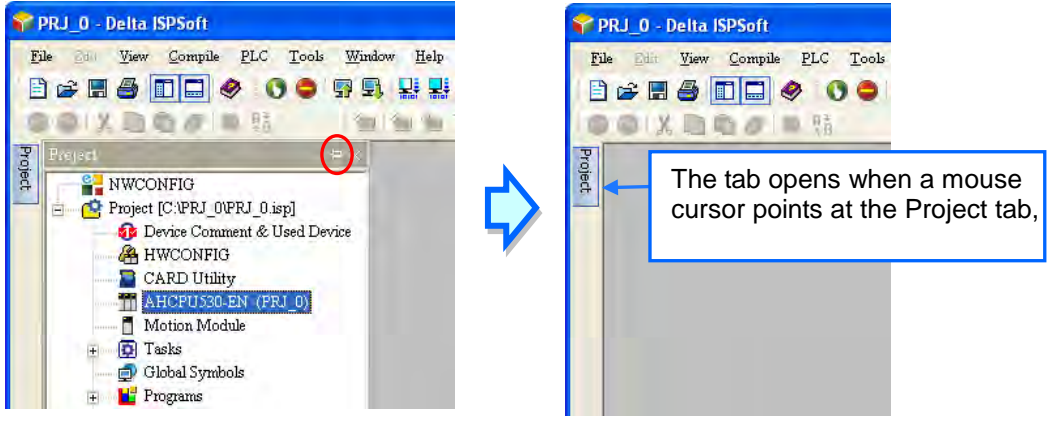
The default Compile Message section designed at the bottom of the ISPSOft main page, users can click on the toolbar or select **Output Window (M)** under **View** from the toolbar to open or close the section. Click on the upper right corner of the window can also close the window.



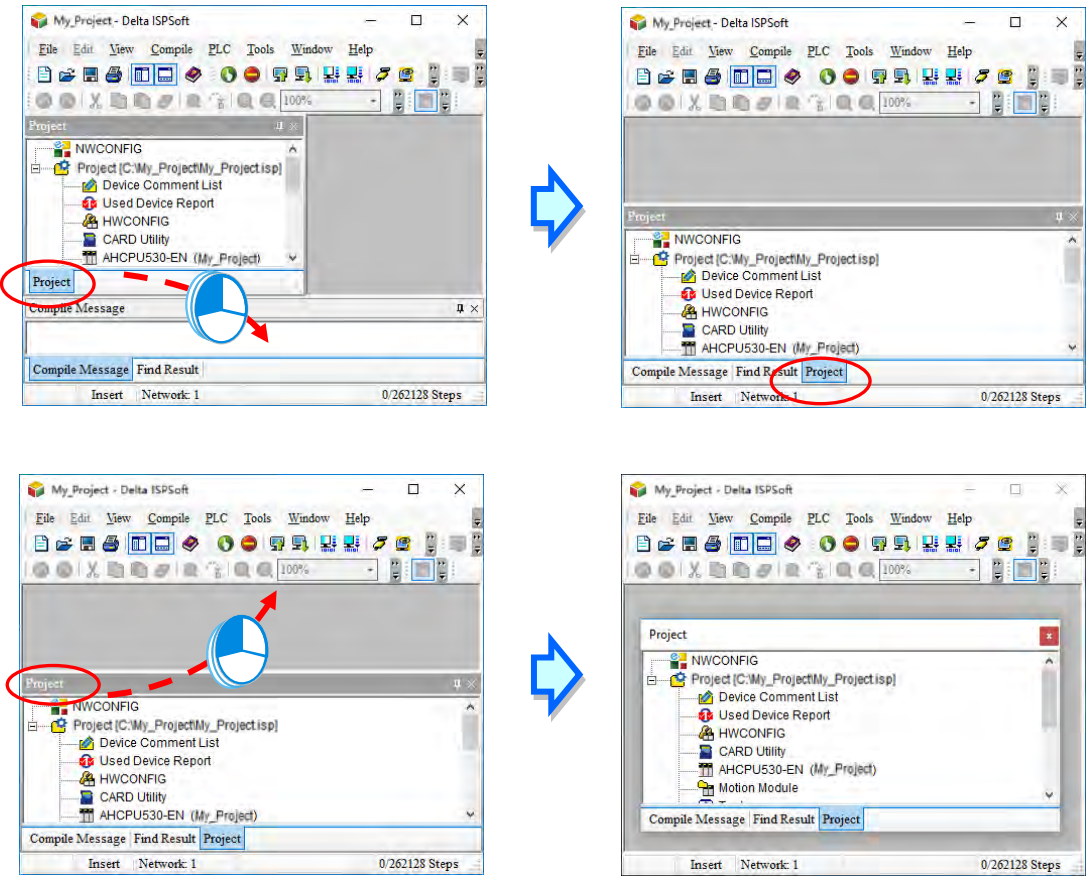


Users can click  on the upper right corner of the window to change the display of the Project or Compile Message section. When the icon changes to , the section is hidden. When the mouse cursor moves away from the section for a while, the window of the section automatically turns into a tab on the page, but when the mouse cursor moves to the tab, the tab is opened.

2



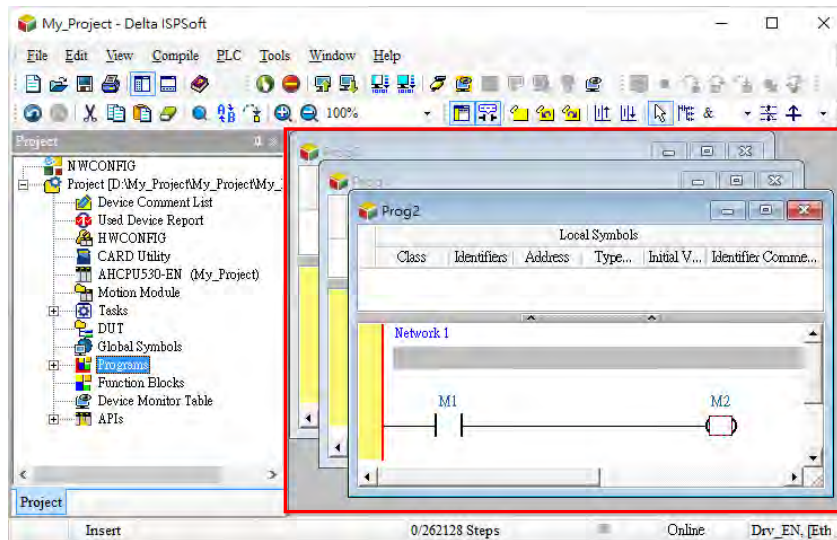
When the section is not hidden, users can drag the tab with a left click mouse button to any position or combine the tab to a different section. If users want to drag the whole section, they have to drag the title bar.



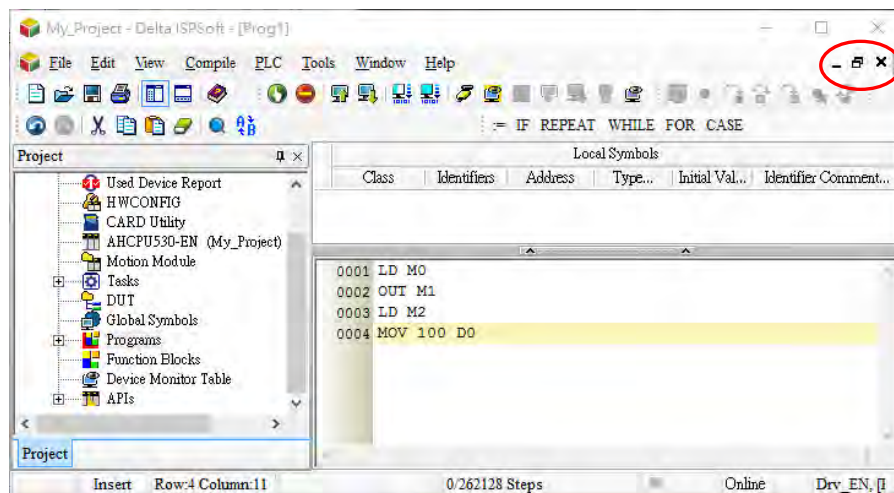
## 2.1.6 Workspace

Users can view maximized or minimized windows in this section. To maximize a window, the status buttons for the window appears on the right side of the toolbar.

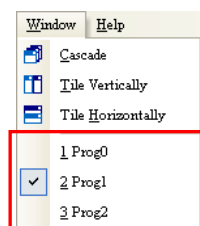
### ● Display windows



### ● Maximize the window

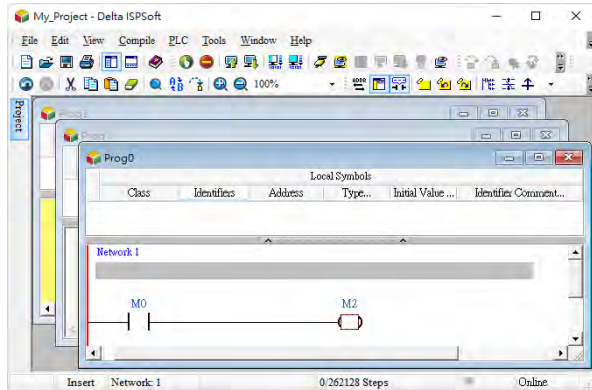
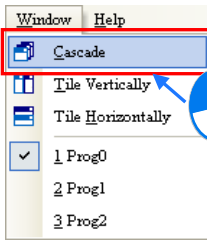


Select **Window** on the menu to change the Prog. windows.

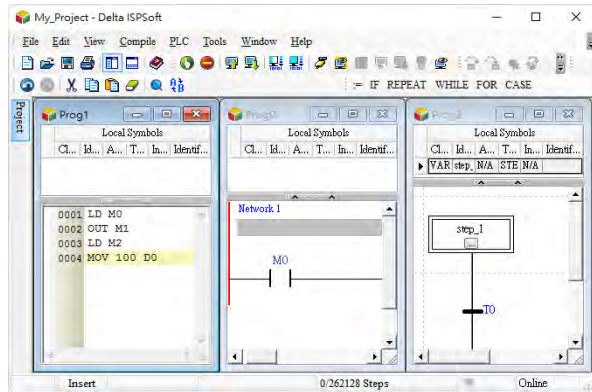
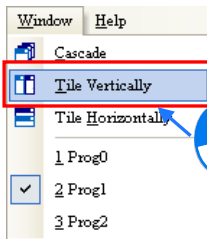


Click to change for the current Prog. Window.

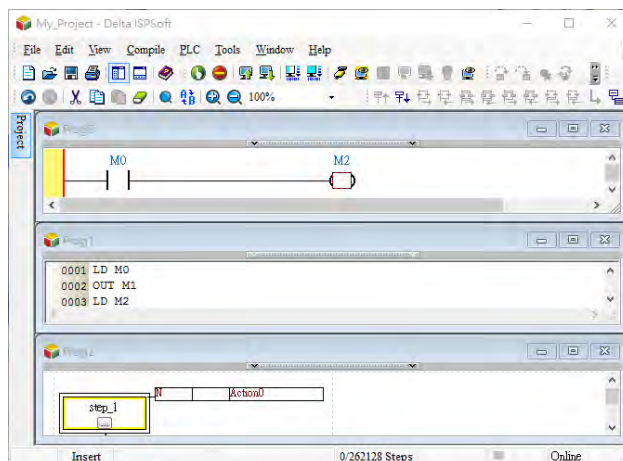
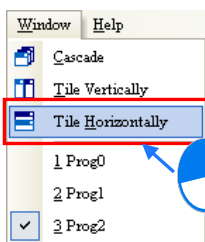
- **Cascade:** When selected, all windows will stack on top of each other, and the current window is the first one displayed.



- **Tile Vertically:** The windows are displayed like tiles in vertical alignment; the current window is at the leftmost side.



- **Tile Horizontally:** The windows are displayed like tiles in horizontal alignment; the current window is at the topmost.



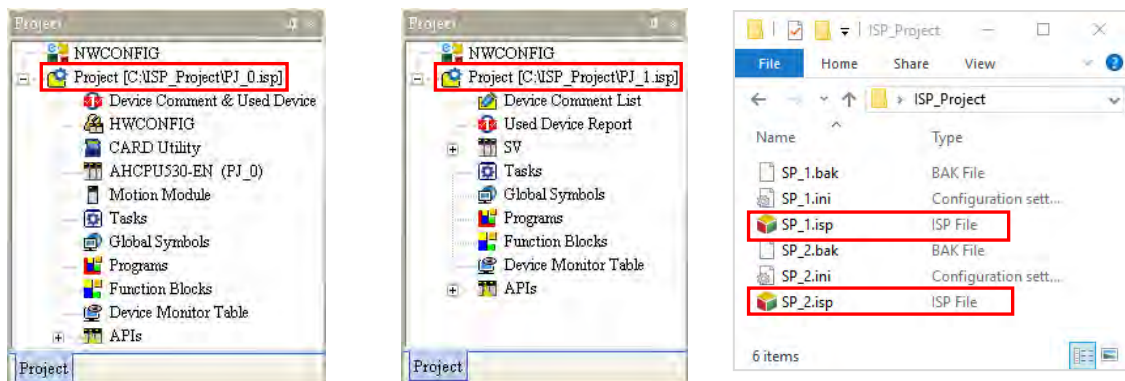
## 2.2 Project Framework

### 2.2.1 Single Project and Group Project

Two types of project frameworks are described below.

#### ● Single project

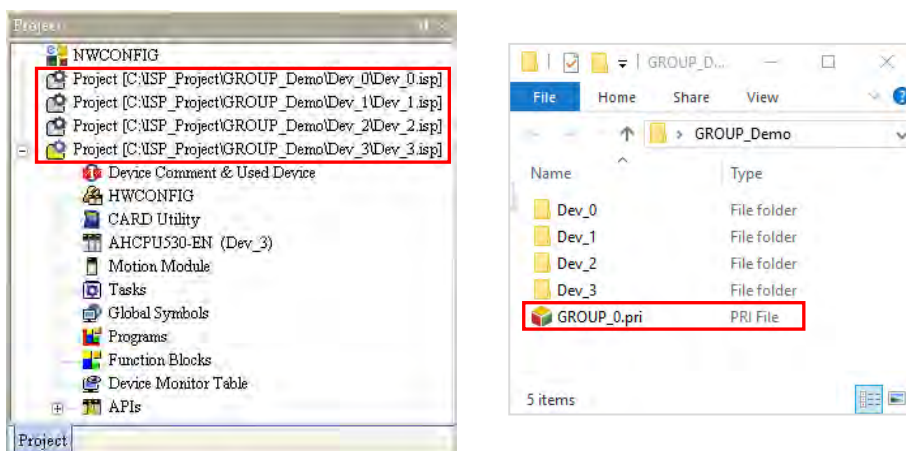
This is the basic project type. Each project can only execute one model. The file type of a single project is \*.isp.



#### ● Group project

If several devices are connected on a network, users can create a group of single projects for the devices without restriction. 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. Users can import a single project which was created previously to a group.

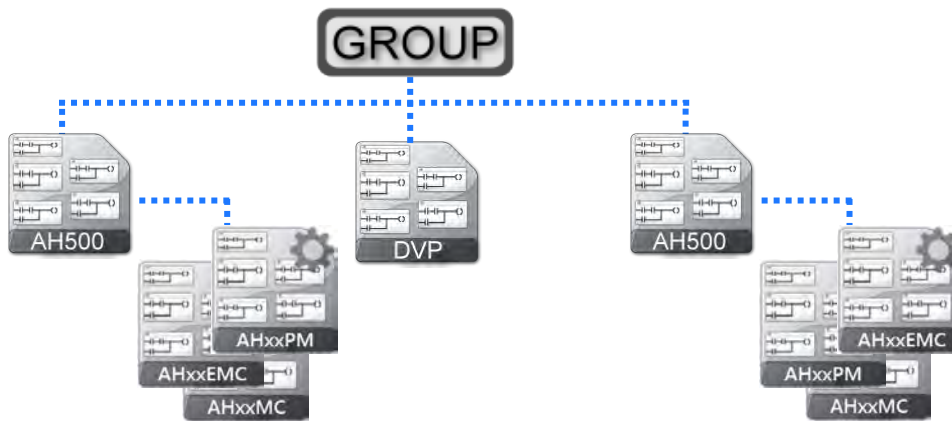


\*. Please refer to chapter 20 for more information about NWCONFIG.

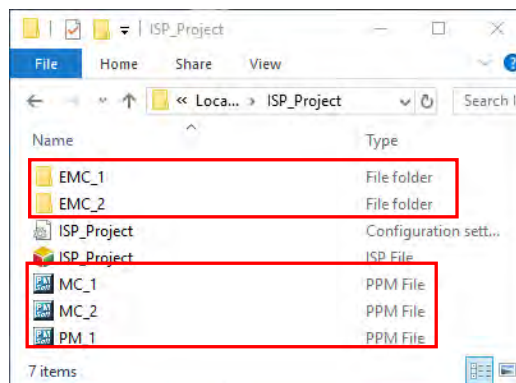
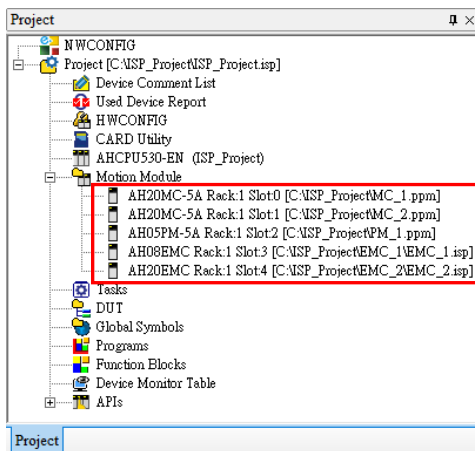
## 2.2.2 Integration of Motion Control Modules

The AH series include motion control modules such as AH05PM-5A, AH10PM-5A, AH15PM-5A, AH20MC-5A, AH08EMC, AH10EMC, AH20EMC. Users can write a program regarding AHxxPM and AHxxMC motion control modules by means of PMSOft, a software developed by Delta Electronics, Inc. Please refer to PMSOft User Manual for more information about usage. The AHxxEMC can be used as host or motion control module in ISPSOft.

Users can create projects concerning motion control modules in ISPSOft and configure hardware as well as parameter settings through **HWCONFIG**. Each AH series can create more than one motion control module project whether the project framework is single or in a group, the corresponding motion control modules also need to be added for hardware configuration in HWCONFIG. Please refer to section 2.2.5 for more information.




The PMSOft projects include AHxxPM and AHxxMC modules, with filename as .ppm. While ISPSOft project contains AHxxEMC modules, the file name is .isp and is placed in the same path as of single projects for AH series. Regarding previously created PMSOft or ISPSOft projects, users can import those to an ISPSOft project.

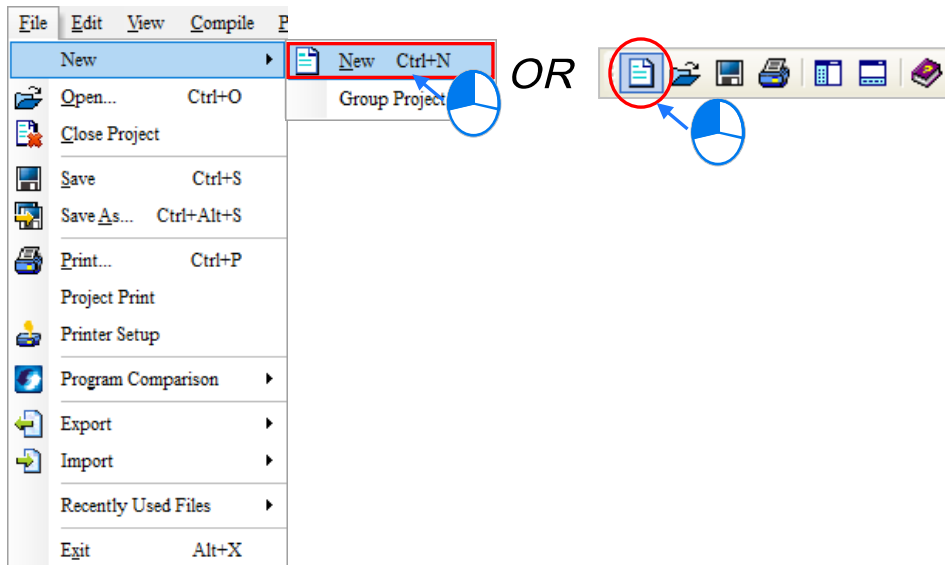


\*1. An imported PM project must use file format that is available for PMSOft v2.05 or above, and do not support DVP-PM series projects.

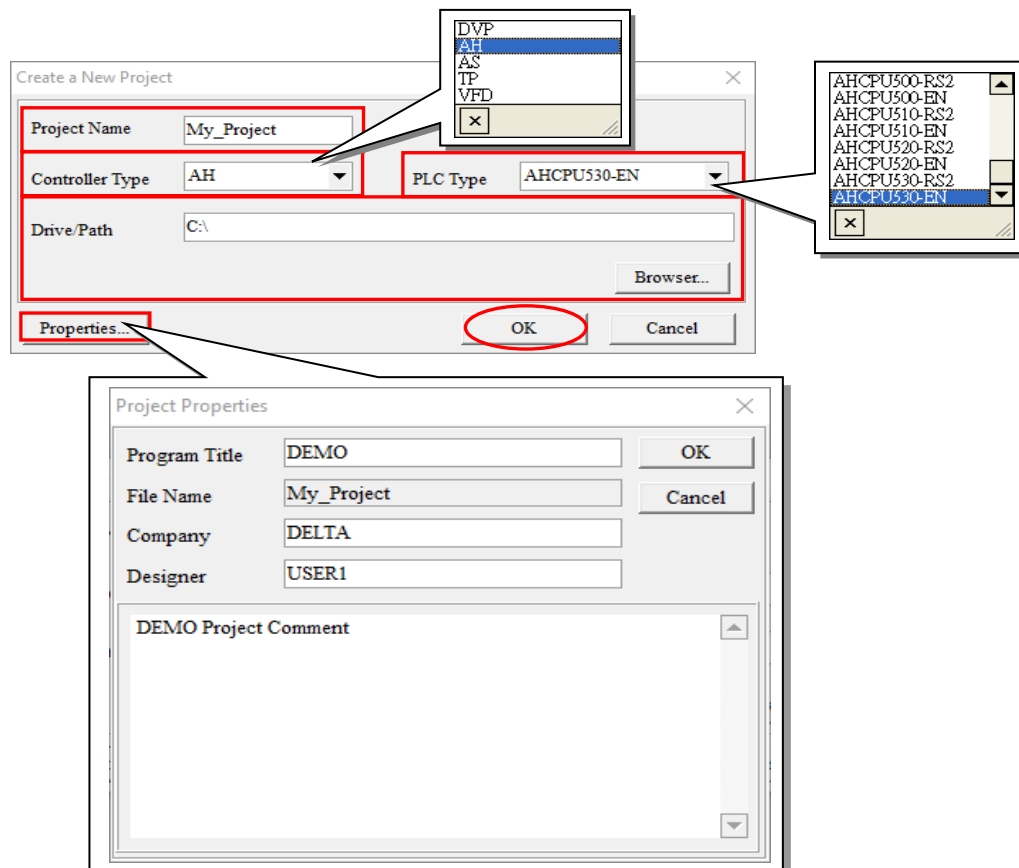
## 2.2.3 Managing a Single Project

### ● 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.

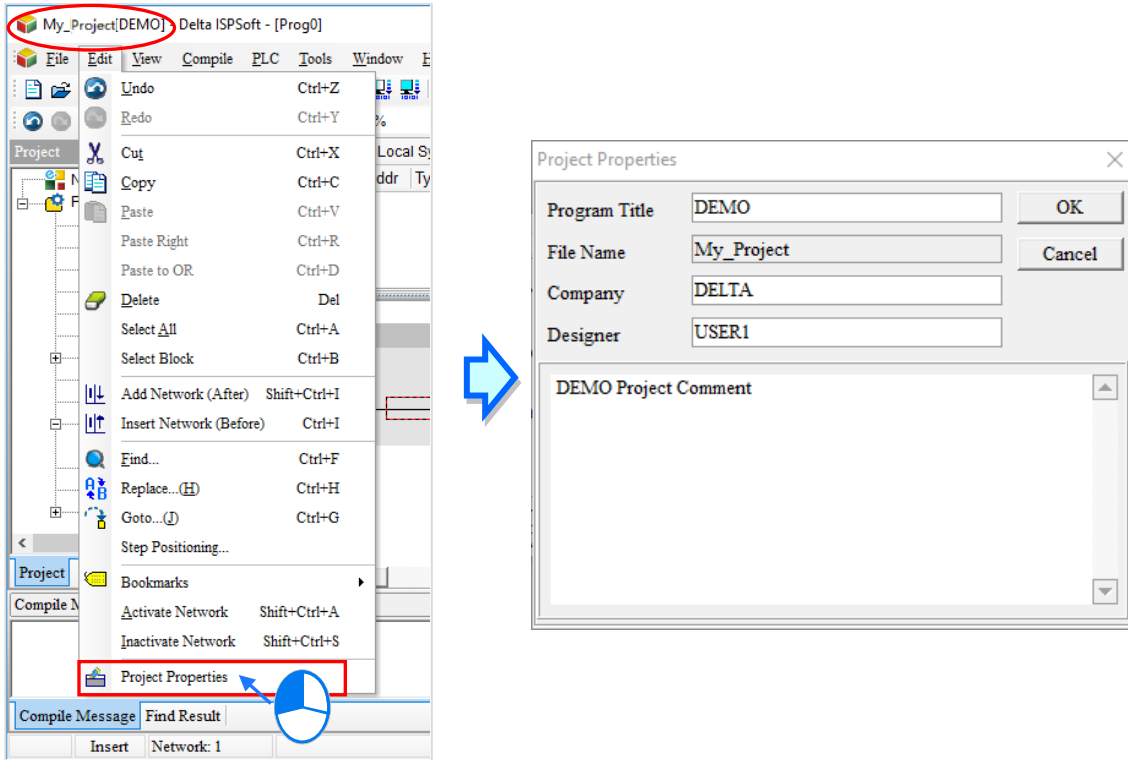


In the **Create a New Project** window, type a project name in the **Project Name** box and a path in the **Drive/Path** box, and select a **Controller Type** and **PLC Type** from the drop-down lists. After users click **Properties...**, users can add description of this project. Finally, click **OK**.

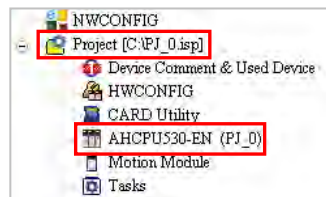


After a project is successfully created, the ISPSOft window title displays the project name, under the format of **project name [Program Title]**. When no program title is setup, the window will display only the **project name**. When users want to view or modify the program title, users can click **Edit** from the toolbar and select **Project Properties** to type in the information.

2




Under Project, users can view the path of the project file and selected model. For AH/AS series projects, the PLC label name is shown in parentheses next to the selected model, and the project name is set as default. However, the project name can be reset in **HWCONFIG**. Users can refer to section 3.3.2.1 for more information.

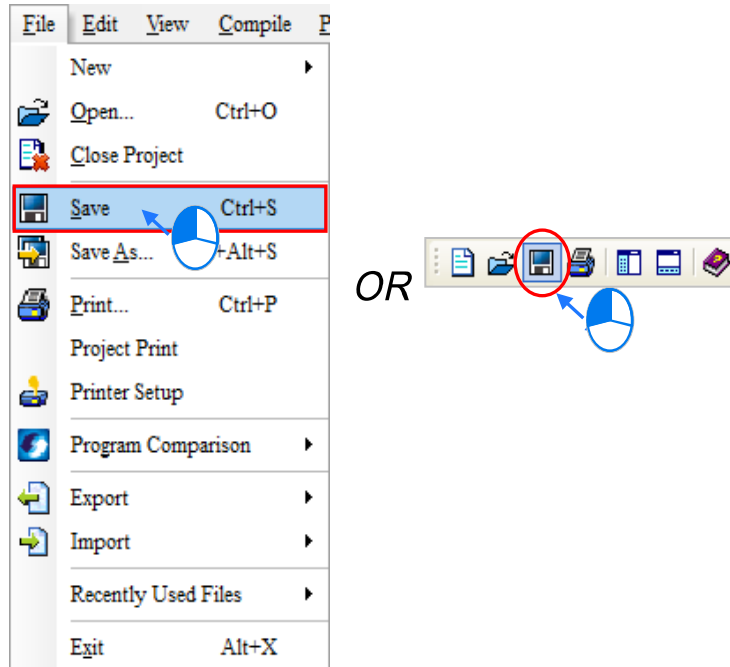


\*. Only AH/AS and DVPxxMC series have PLC label names, but DVP series do not. The main function of the label name is to perform device identification, especially for network applications or other online operations. The gathered information can be used to check whether the object is operating as we expected.

● **Save project**

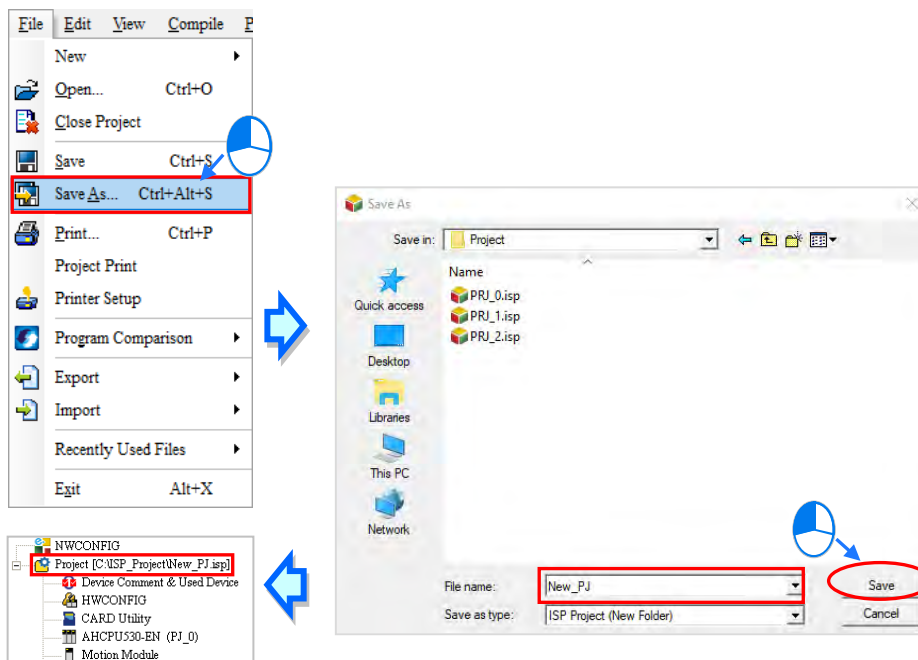
When a project is created, the file denoted by a path is not established immediately. Users need to click 'Save' first for the file to exist. To 'Save' a project, please refer to the following description.

Select **File** from the toolbar and click **Save**. Or click  on the icon bar.



2


Select **Save As** from the **File** menu to save the file using a different file name or save the file denoted by a path in another location on your computer. After setting a new path and a new file name, click **Save**.



\*. To 'Save As' a file will not change the PLC label name, the name adopts the created project name by default but can be modified in HWCONFIG.

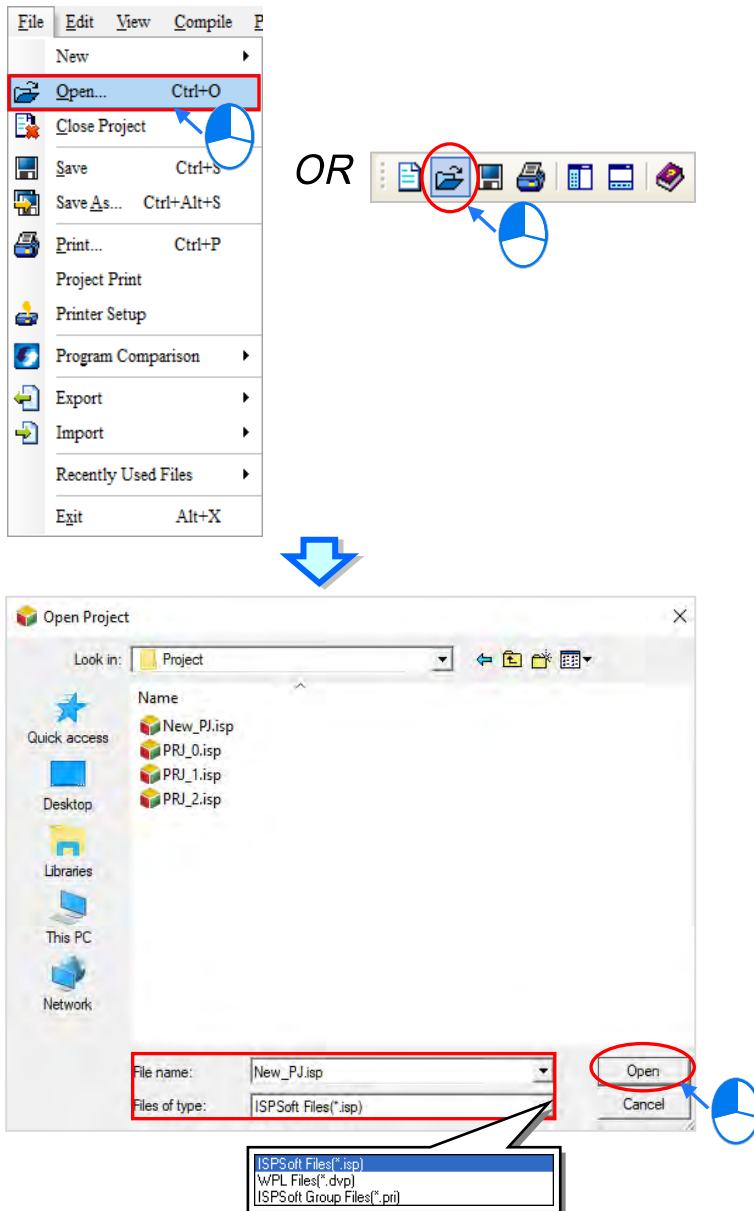


● **Open previous project**

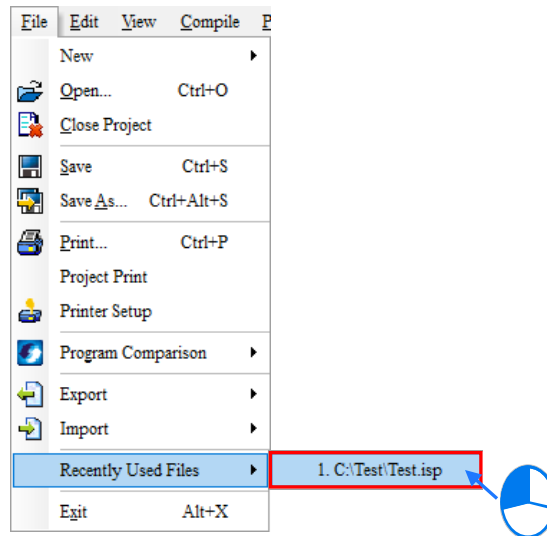
If users want to open a previous project, they can click **Open** on the **File** menu, or click  on the toolbar. Users can select from the **Files of type** for the previous file and click **Open**. The following table shows the file types that can be opened.

File type	Description
ISPSoft Files (*.isp)	This is a single project file. If old file format is opened, the ISPSoft will automatically change the file into the new format.
WPL Files (*.dvp)	This file is created be WPLSoft. When the file is opened and saved, it is automatically changed into an ISPSoft file (*.isp).
ISPSoft Group Files (*.pri)	It is a group of files in ISPSoft.

\*. WPLSoft is a PLC programming software produced by Delta Electronics, Inc.. Please refer to WPLSoft User Manual for more information.



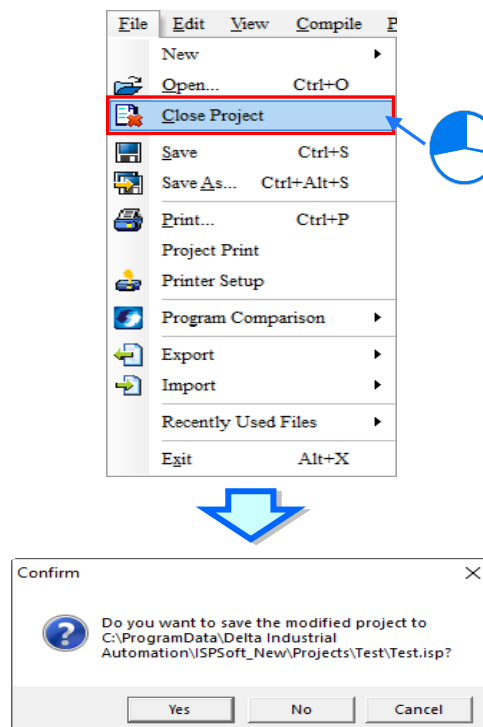
Users can select **File** from the menu to open Recently Used Files.



2

### ● Close project

Click **Close Project** from the **File** menu once the editing is complete. When the modified project is not saved, a pop-up window appears to confirm whether the project needs saving or not.



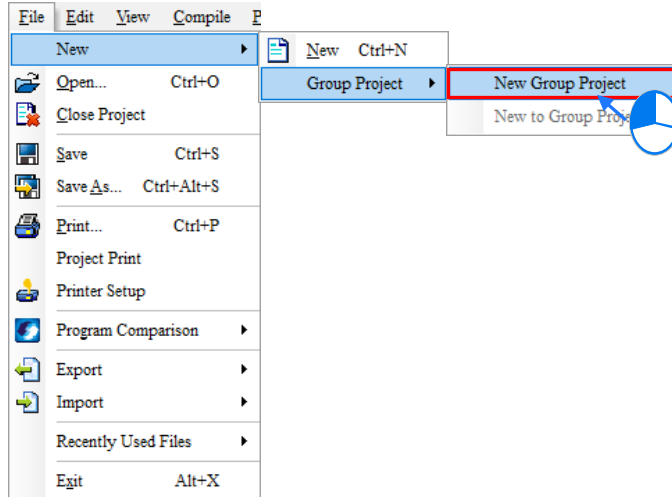
### **Additional remark**

If users do not close the project before leaving ISPSOft and restart the software again next time, the system will automatically open the file edited from last time by default. (Users can set the function to enable or disable for opening previous projects. For more details, please refer to section 2.3.1.)

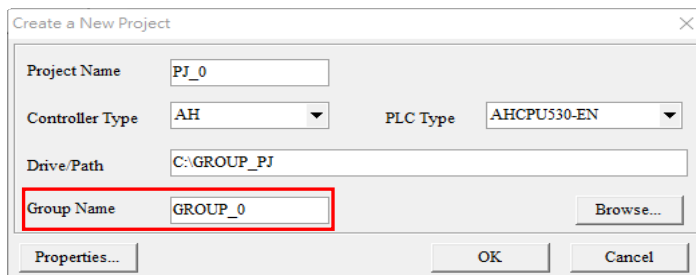
## 2.2.4 Managing Group Project

### ● Creating group project

Click the **File** menu, choose **New** then move to **Group Project** and click on **New Group Project**.



When creating group project, users also need to create a single project from the group. In the **Create a New Project** window, type the **Project Name** and **Drive/Path**, select the **PLC Type** and **Controller Type** from the drop-down lists. Users can add information and description related to the project by clicking **Properties**; Nevertheless, type **Group Name** in the box then click **OK**.

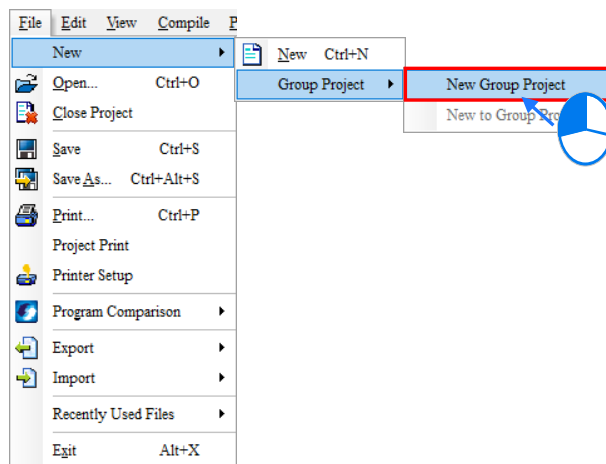


After a group project is created, information related to the first single project will be displayed in the Project section. Owing to the fact that there is only one single project in the group, so it is displayed and operated in the same way as creating a single project. The only difference is under the group project framework, users cannot execute the **Save As** function.

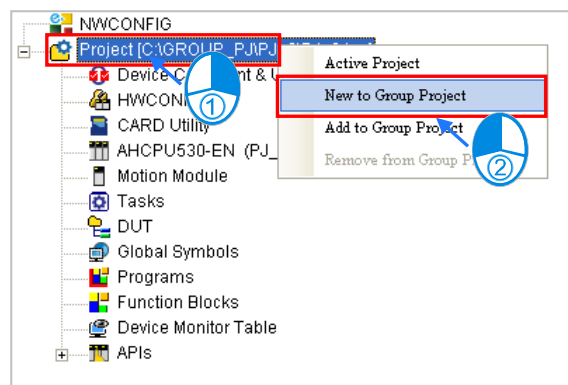


### ● New to group project

Click **File** from the menu, choose **New** move to **Group Project** and click **New to Group Project** to add new single project to the group. Or, you can also right-click a **Project** in the section and select **New to Group Project**.

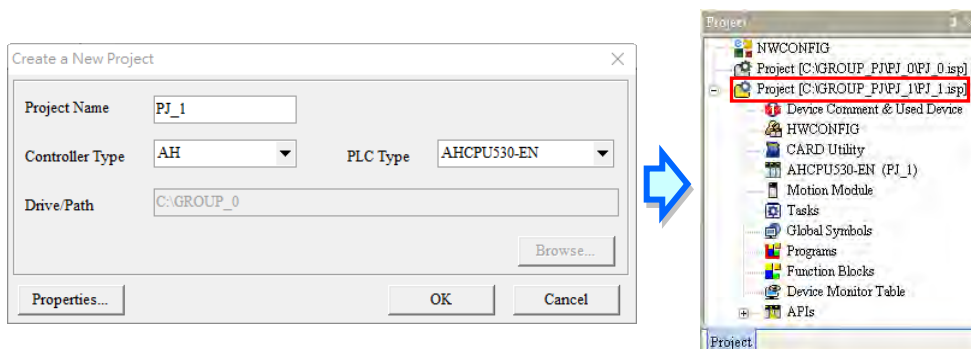


OR



In Group Project, only one project is edited at a time. Therefore, when a new project is added to the group, the editing project needs to be closed. A window will appear to confirm if this editing project needs saving or not.

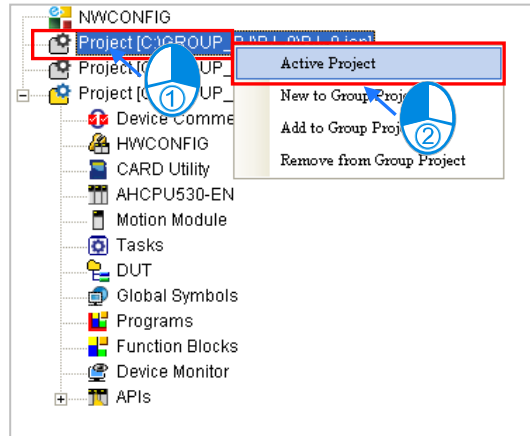
In the **Create a New Project** window, users can type a project name in the **Project Name** box, and select a PLC in the **PLC Type** drop-down list box. Besides, after users click **Properties...**, they can give a description of the project. However, users can not specify a path.



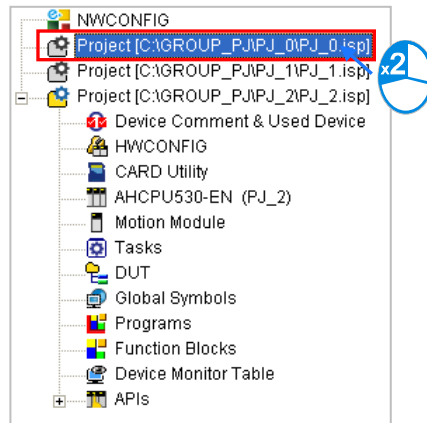
● **Active project**

In Group Project, only one project is edited at a time. To edit another project, right-click on the selected project and choose **Active Project**. Or, double click to activate on the selected project.

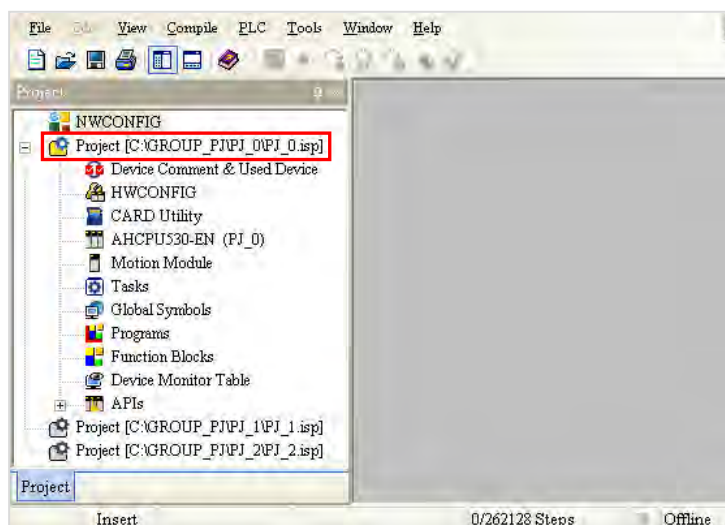
2



OR

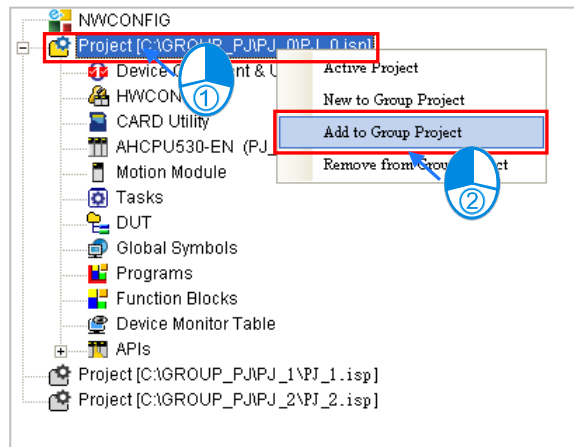


When switching to another project, the selected file can be edited in the Project section, while other files not activated are in gray colors.

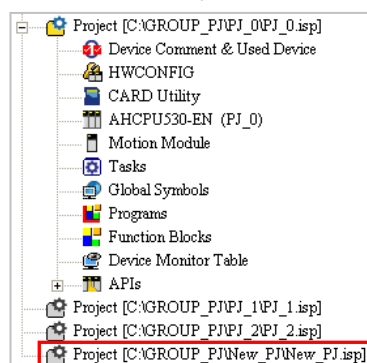
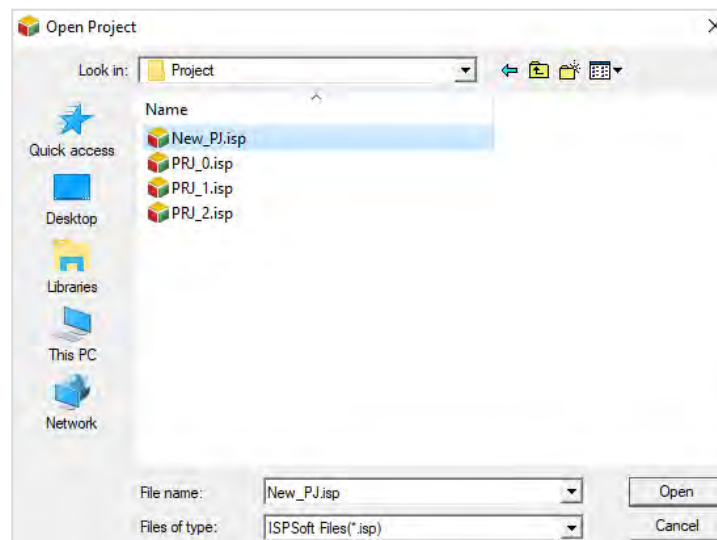


### ● Importing single project to a group

When a single project (\*.isp) file is not created under a group, users can import it into the group. To import a single project, you can right-click a project in the Project section, then select **Add to Group Project** from the quick menu to select the import file.



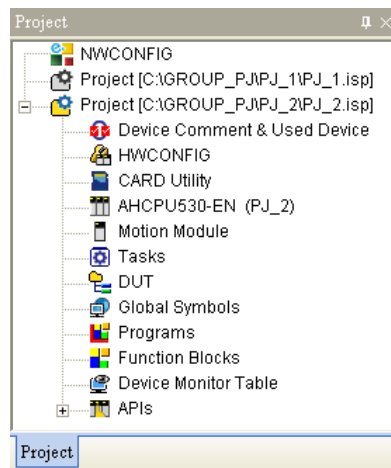
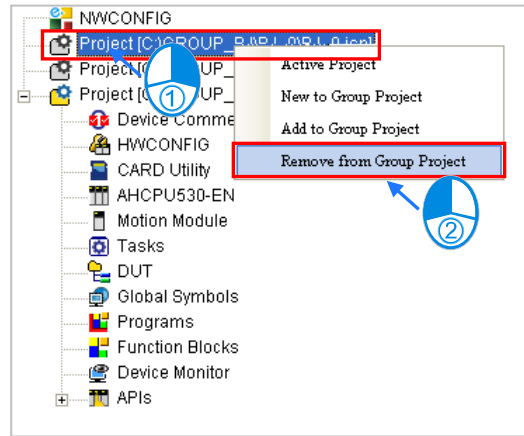
Only single project \*.isp files are allowed to be imported. When importing a single project, the project will be copied to the group project file, and the single project is still saved in the original path of your computer.



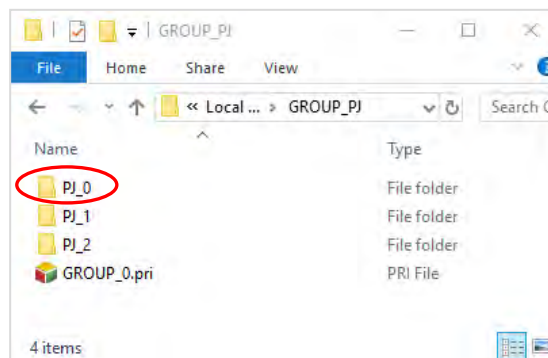
● **Remove from group project**

When removing a single project from a group, they can right-click the selected file in the Project section and click **Remove from Group Project** from the quick menu. However, if there is only one project in the group, that project cannot be removed.


2

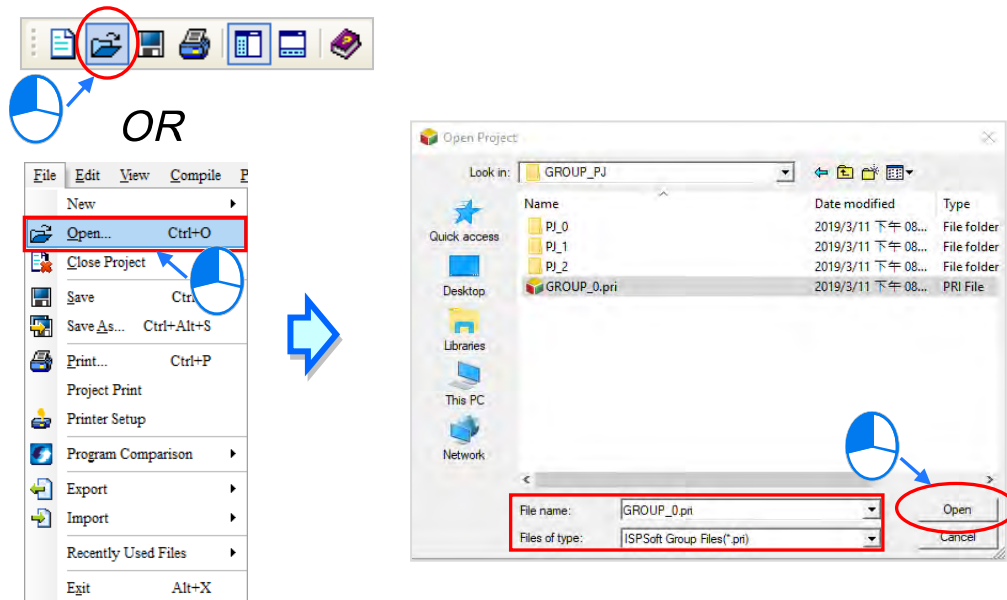


When removing, the file is removed only from the list of projects in the group project file, and not from its former path. Users can re-add the project into the group through importing the file.

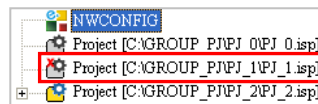


## ● Open a group project

To open previously created group projects, click **Open** from the **File** menu, or  in the toolbar, then choose **Files of type** to select **ISPSOft Group Files (\*.pri)** in the drop-down list box and click **Open**.

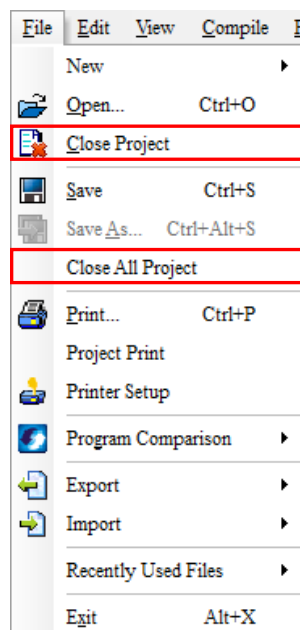


When the group file to be opened has a red cross icon on it, this means the single project saved in the path used to denote group project appears abnormal and may be either removed, saved in a new path or using a new file name.



## ● Close project

Click **Close Project** from the **File** menu to close editing projects, but other inactivated single projects can still be activated. Users can click **Close All Project** to close the entire project.



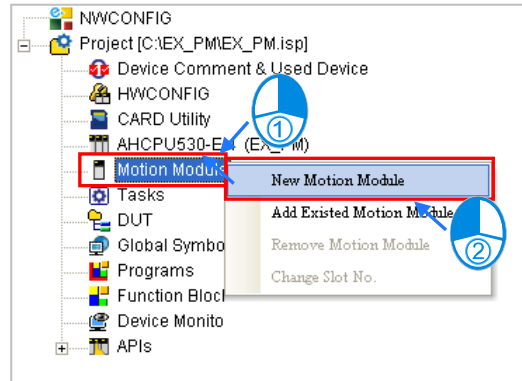


## 2.2.5 Managing Motion Control Modules

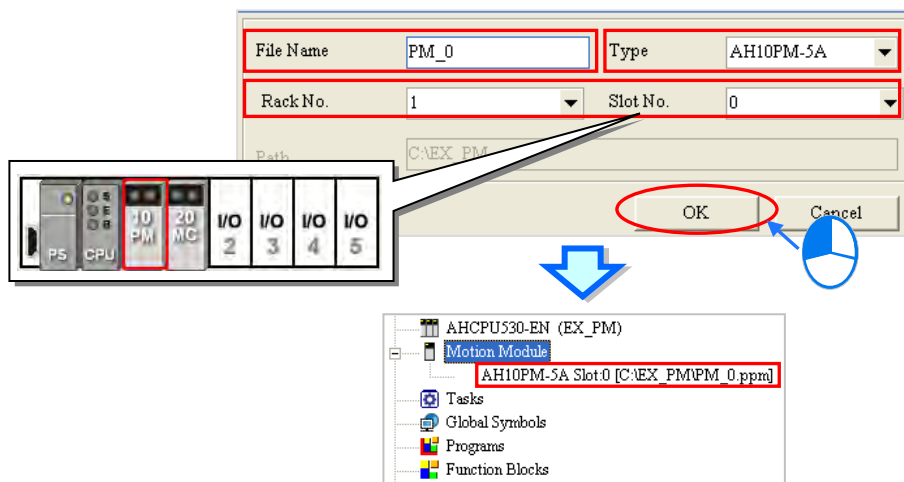
In the Project section of ISPSOft, the **Motion Module** option is added for AH series. To add AHxxPM and AHxxMC motion modules, users must install PMSOft version 2.05 or above.

### 2 • New Motion Module - AHxxPM and AHxxMC

If users want to create a PMSOft project in ISPSOft, click **Motion Module** in the **Project section** and choose **New Motion Module** from the quick menu.

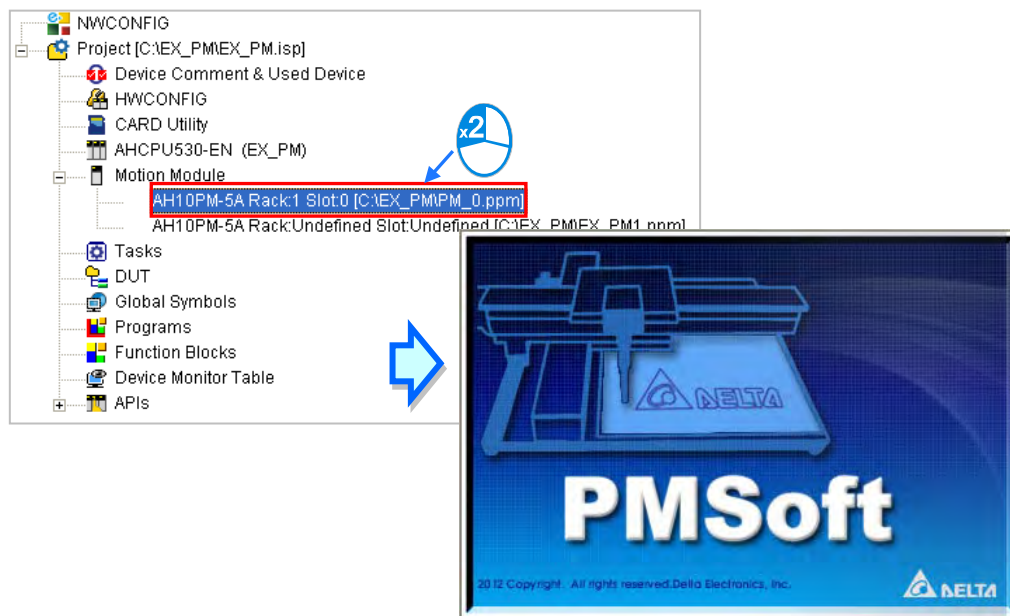


Then, type the **File Name** of the motion module project and select a model **Type** from the drop-down list. In addition, choose the **Rack No.** and **Slot No.** from the drop-down lists based on **HWCONFIG** configuration. Each slot also corresponds to a motion module. If users are uncertain of the mounting position or may rearrange the slots later, users can select **Undefined** from the **Slot No.** drop-down list. Click **OK** once the setting is complete.



### ● Open AHxxPM and AHxxMC Motion Module Projects

To edit AHxxPM and AHxxMC motion modules, please double-click the project file and PMSOft will automatically be opened for users to implement software development and programming on motion modules in PMSOft.

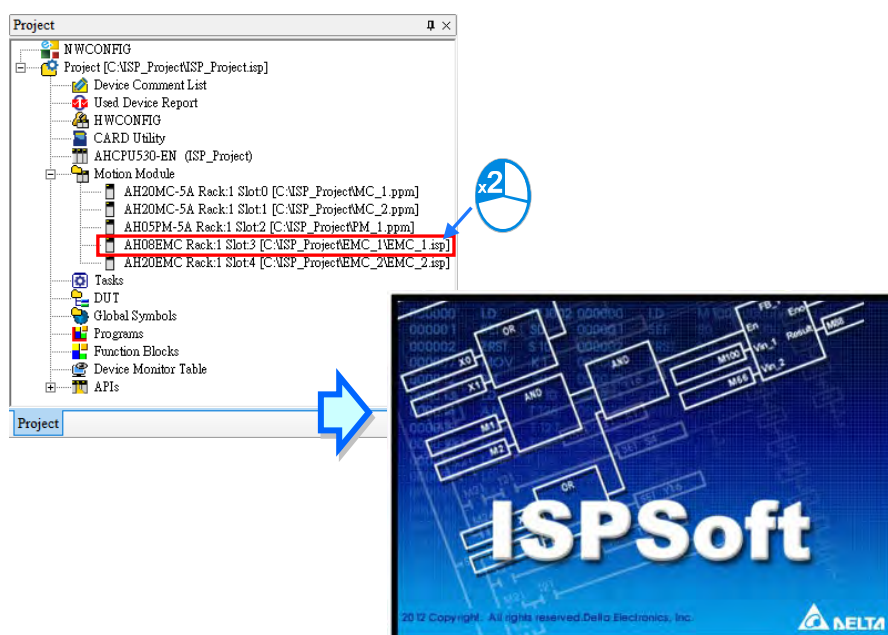


2

\*. Please refer to PMSOft User Manual for more information on the usage. If PMSOft is opened through ISPSOft, users can only edit motion modules, but must save the edited content in PMSOft to complete the execution.

### ● Open AHxxEMC Motion Module Projects

To edit an existed project concerning AHxxEMC motion module, please double-click on the file and ISPSOft will automatically be opened for users to implement software development and programming on motion modules in ISPSOft.

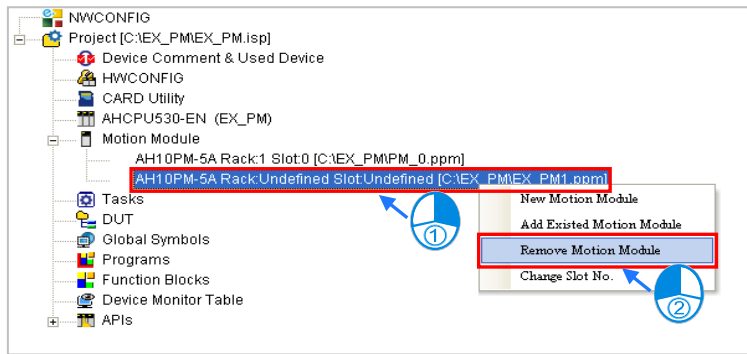


\*. When ISPSOft is opened through the ISPSOft, users can only edit motion modules, but must save the edited content in ISPSOft to complete the execution.

● **Remove Motion Module**

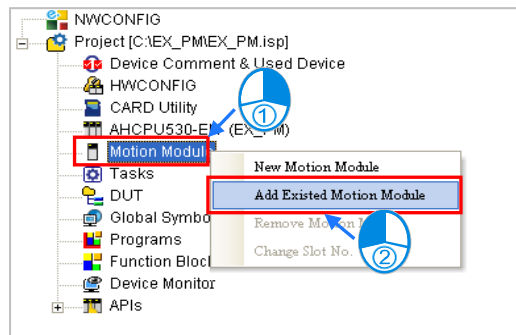
To delete an existed motion module project, users can right-click the selected project and choose **Remove Motion Module** from the quick menu; when removing, the module project files (\*.ppm) or (\*.isp) is removed only from the list under the Project section, but still saved in its original path.

2

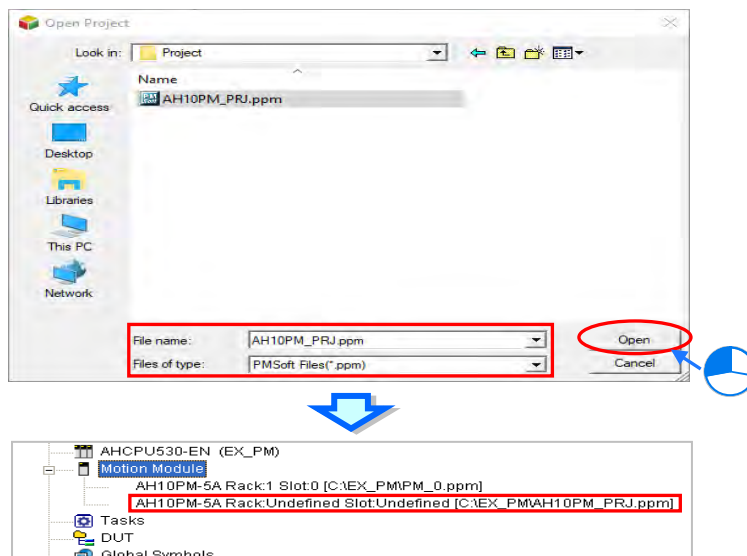


● **Add Existed Motion Module**

To add an existed motion module project file (\*.ppm) or (\*.isp), right-click **Motion Module** in the Project section and select **Add Existed Motion Module** from the quick menu.



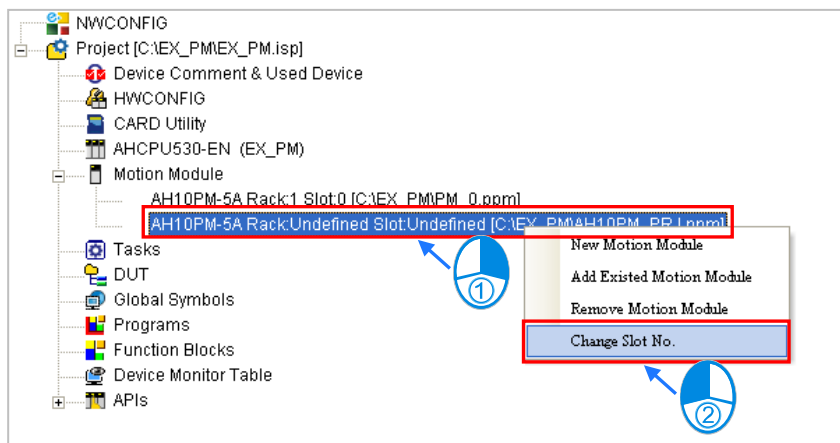
Choose the added motion module project based on file name and type in the Open Project window, then click **Open**. The slot number is set as undefined.



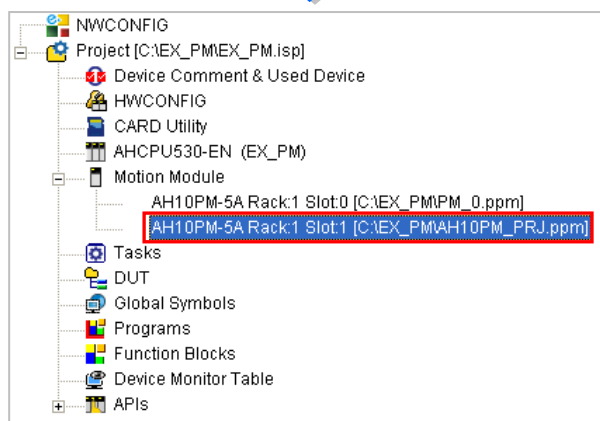
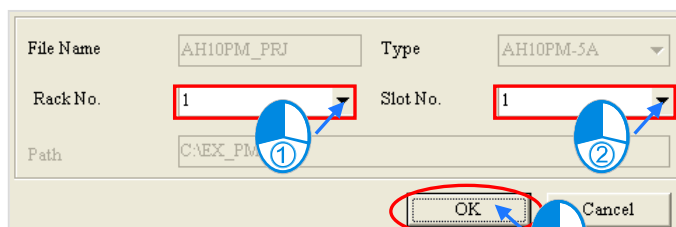
When adding a PMSOft project, the file is copied to the path concerning ISPSOft projects, while the original PMSOft project is saved in the former path. The file format used in PMSOft version 2.05 or above is applied to the added PMSOft project, but does not apply to DVP-PM projects; when adding an ISP project file, a file is created and copied to the path concerning ISPSOft projects.

## ● Change Slot Number

If users want to change the slot and rack number in a motion module project, right-click the selected project and choose **Change Slot No.** from the quick menu.



Select the **Rack No.** and **Slot No.** from the drop-down list. Please do not select a slot number which is already occupied, then click **OK**.

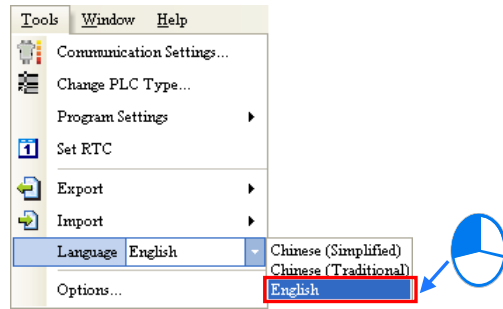


## 2.3 Basic Configuration

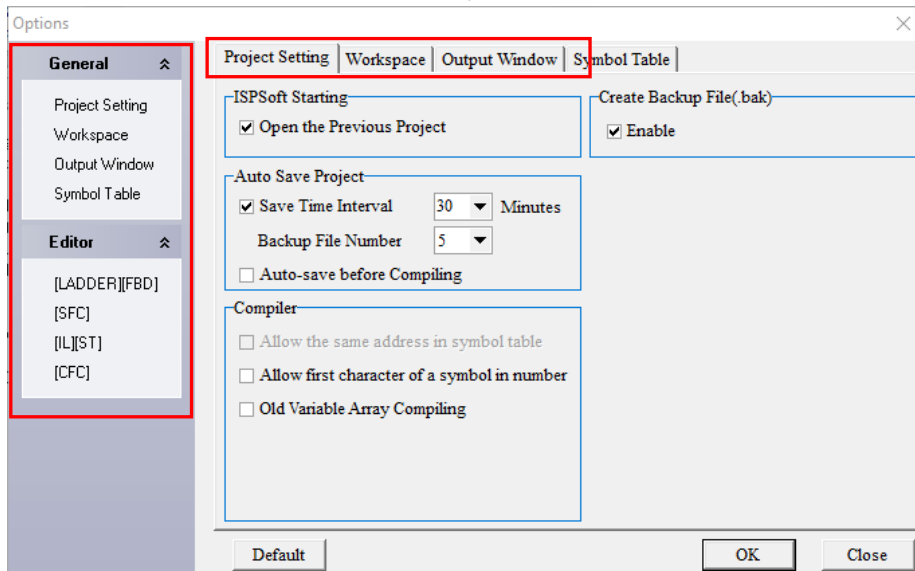
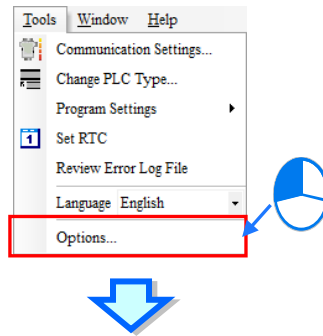
### 2.3.1 System and Environment

Select **Tools** and choose **Language** from the drop-down list.

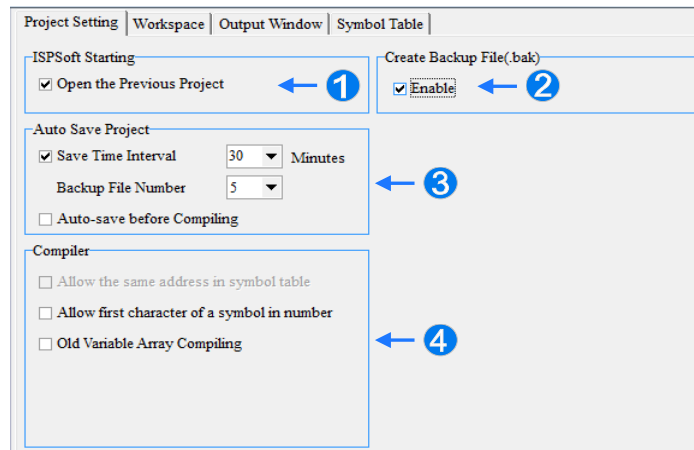
2



Users can click **Options** in the **Tools** menu and choose desired settings from corresponding tabs at the top or listed on the left side of the window.



## ● System Configuration - Project Setting

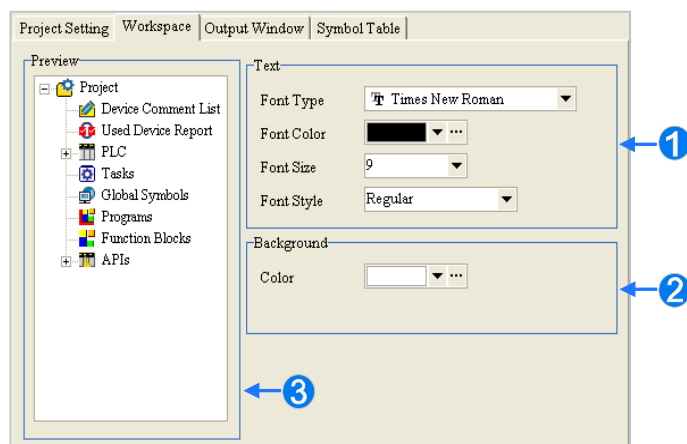


- 1 **Open the Previous Project:** When starting ISPSOft, the system will automatically open the previous project that was last edited and not closed.
- 2 **Create Backup File & Enable:** Creates a backup file when saving.
- 3 **Save Time Interval:** The system automatically saves the project files according to the selected **Minutes** from the drop-down list in the Auto Save Project section. Next, users can choose the maximum **Backup File Number**. If the auto-saved files exceeds the maximum backup number, the earliest saved files will be replaced by the backup files accordingly; if ISPSOft is not closed properly and a project file is opened, users will be asked if the last backup file of that project need to be loaded in or not.

**Auto-save before Compiling:** Auto-save the edited project file before compiling.

- 4 **Allow the same address in symbol table:** Allow different symbol variables to assign the same address.
  - Allow first character of a symbol in number:** Allow the first character of a symbol variable to be a number.
  - Old Variable Array Compiling:** Supports variable array compiling of previous version.
- \*Functions may vary based on different PLC models.

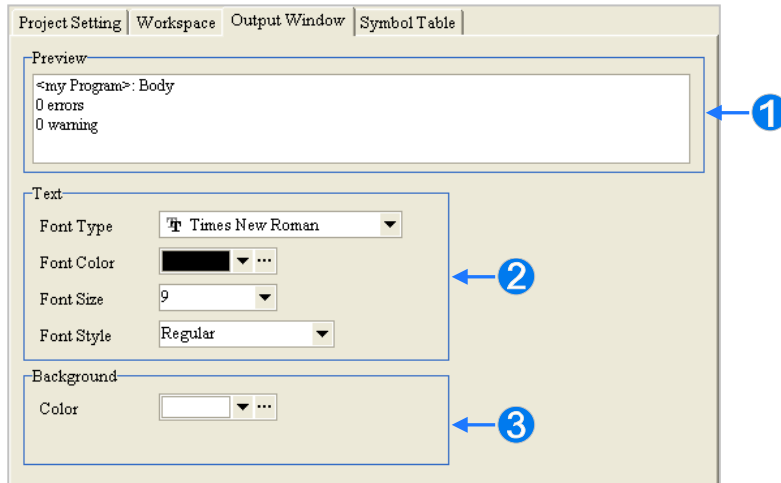
## ● System Configuration - Workspace



- 1 Setup Text related contents i.e. font type, color, size and style.
- 2 Setup a background color for Workspace.
- 3 A preview of the modification.

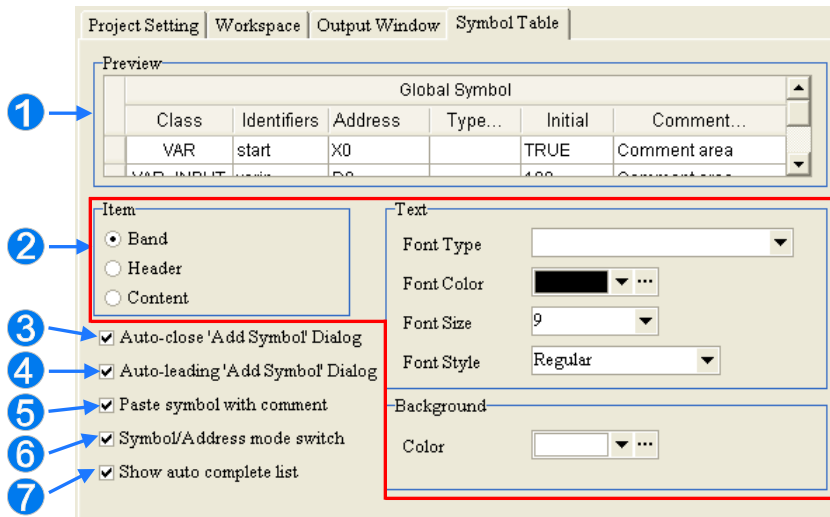
## 2

### ● System Configuration - Output Window




- 1 A preview of the modification.
- 2 Setup output text.
- 3 Setup a background color for Output.

### ● System Configuration - Symbol Table

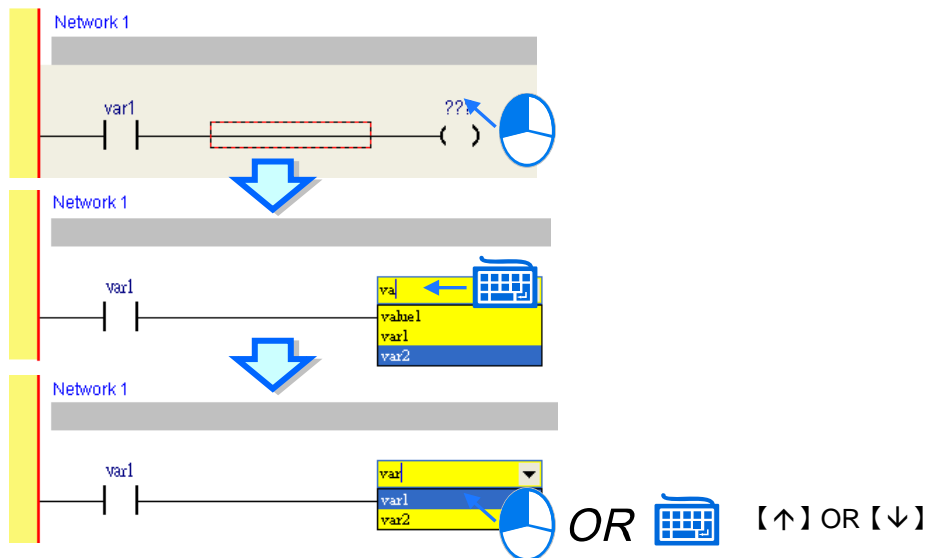


- 1 A preview of the modification.
- 2 Select an item in the left **Item** box, then choose the style of display in the right **Text** box.
- 3 The dialog **Add Symbol** configuration will close automatically once it is declared.
- 4 Input undefined symbol variables in ladder diagram (LD), function block diagram (FBD), or a

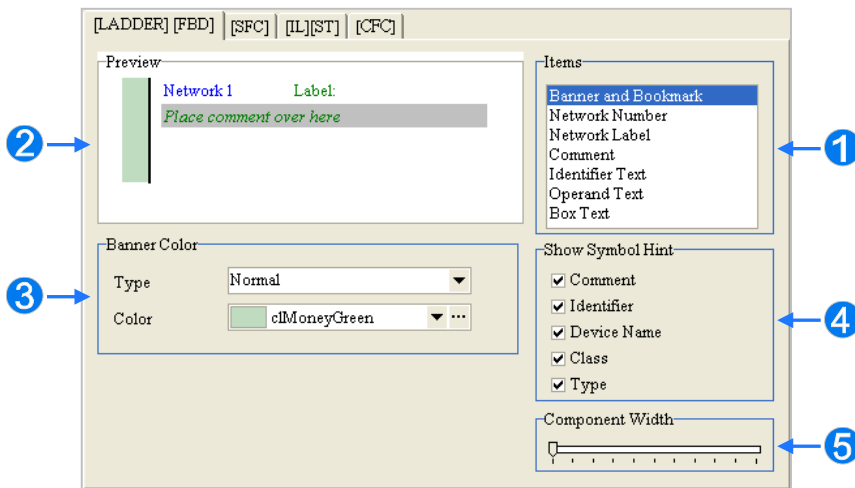
sequential function chart (SFC) environment, click [Enter] when complete and the dialog **Add Symbol** configuration appears.

- 5 When copy or paste from symbol tables, select to include comment.
- 6 Select  from the icon toolbar to switch between Symbol or Address mode.
- 7 Input symbol variables in the program and a list of created variables are shown. To select, users can use the mouse or the up and down arrow keys on the keyboard. (See below)

2



● Configuration Editor - LADDER and FBD



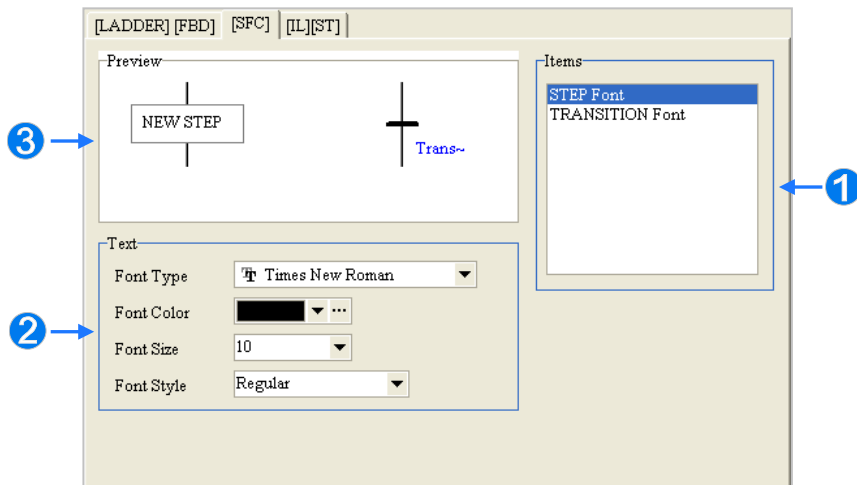
- 1 Select an item for setup.
- 2 A preview of the modification.
- 3 Select a display type and color.
- 4 In Ladder and FBD programming editor section, users can choose from the 'Show Symbol Hint' box to display selected hints of the device or symbol variable pointed by the mouse cursor.



- 5 Set the Ladder and FBD component width in the box. The setup will affect the word length in FBD programming.

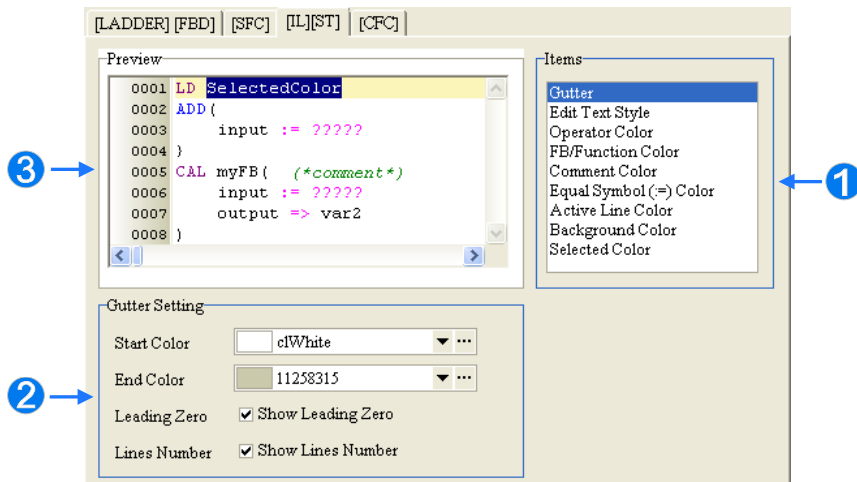
● Configuration Editor - SFC

2



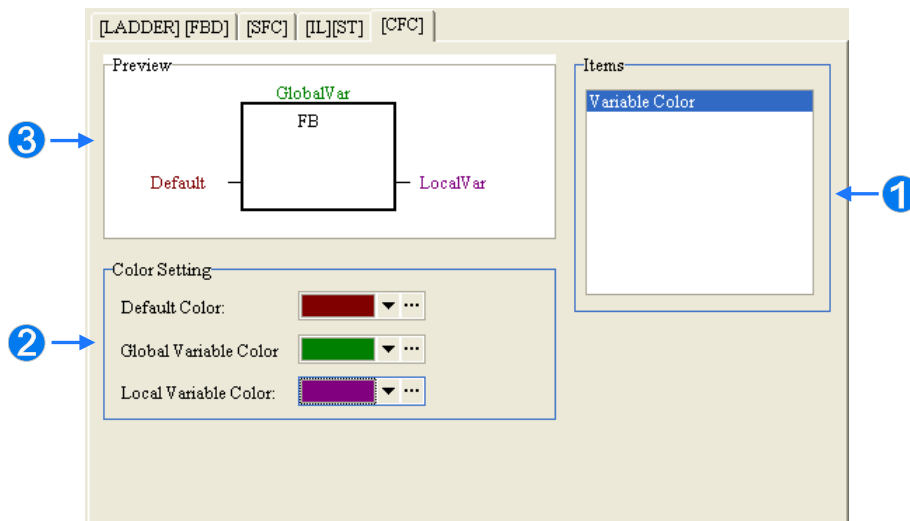
- 1 Select an item for setup.
- 2 Based on the selected item, select the Text box for display.
- 3 A preview of the modification.

● Configuration Editor - IL and ST



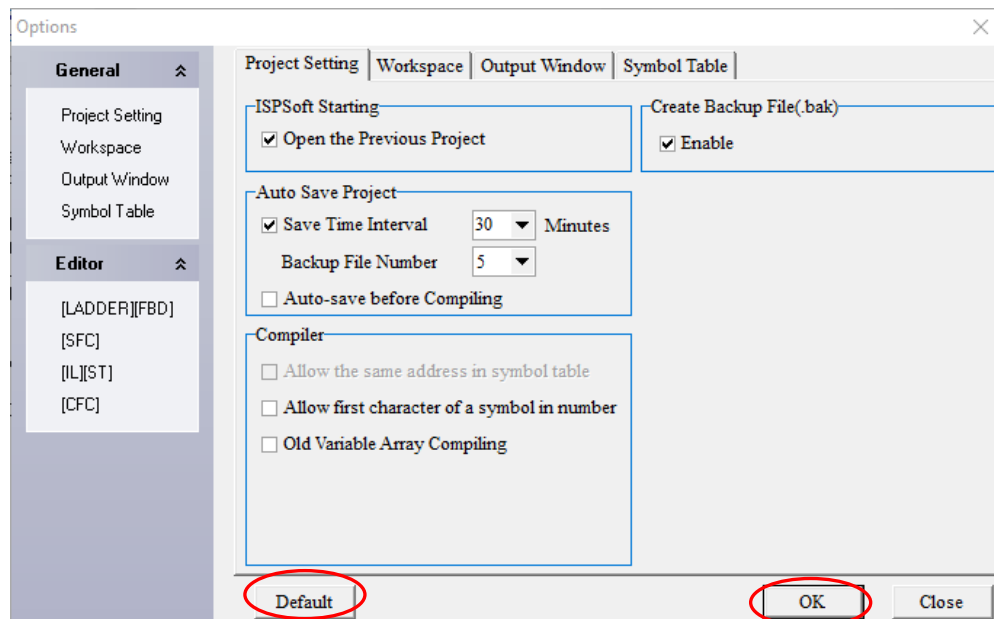
- 1 Select an item for setup.
- 2 Based on the selected item, select the Text box for display.
- 3 A preview of the modification.

### ● Configuration Editor - CFC



- ① Select an item for setup.
- ② Based on the selected item, select the Color Setting for display.
- ③ A preview of the modification.

By clicking **Default**, all settings are restored to defaults. When modification for all setups are complete, click **OK**.



\*. When several windows are opened in ISPSOft, each window provides a different setting. However, the software only remembers the environment setting of the last window closed, so it will automatically apply that setup once re-started.

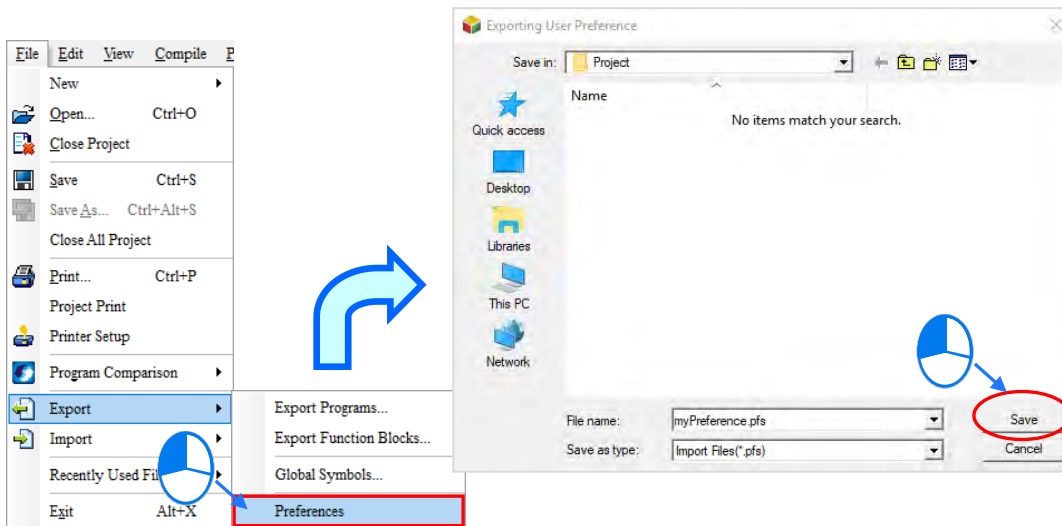
### 2.3.2 Importing and Exporting User Preference

Users can export and import their preference concerning environment settings through ISPSOft.

**● Export User Preference**

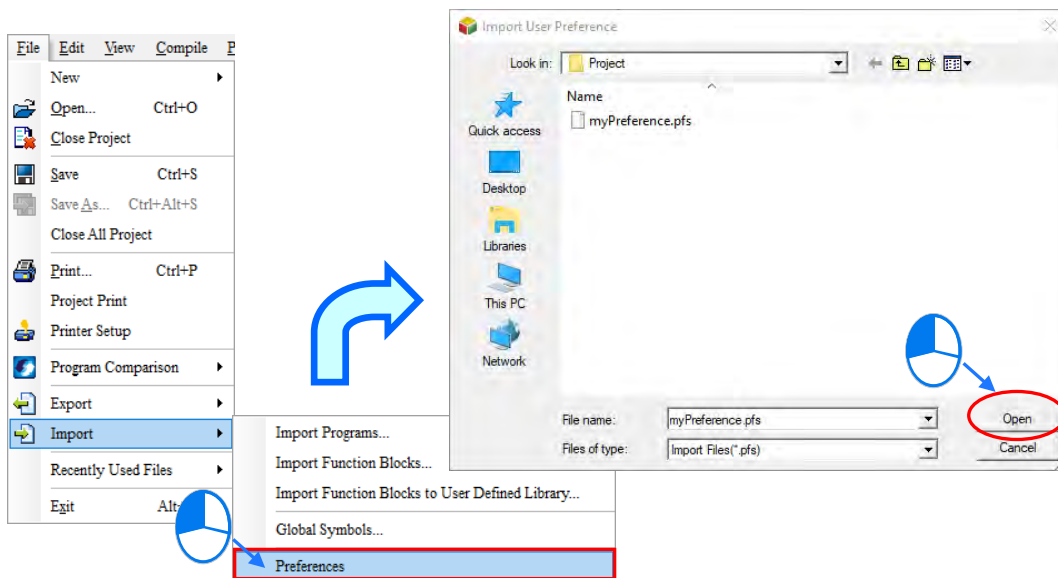
2

Choose **Export** from the **File** menu and select **Preferences**. Input file name to export from the pop-up window and click **Save**.



**● Import User Preference**

Choose **Import** from the **File** menu and click **Preferences**. Select a file for import from the pop-up window and click **Open**.

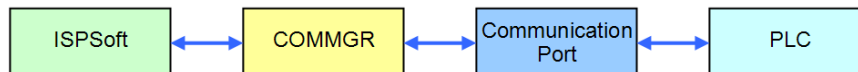


**Additional remark**

Since environment parameters may vary based on different ISPSOft versions, not all import preferences may be applicable. Users need to check the imported **Preference** file (\*.pfs) to make sure the contents is desired if unsure which ISPSOft version it is exported from.

## 2.4 Communication Settings

The following picture shows a communication structure between ISPSOft and a Delta PLC. Unlike previous ways of connection, the ISPSOft uses Communication Manager- **COMMGR** as the communication interface. The section explains how to build communication between ISPSOft and PLCs with basic tests completed.



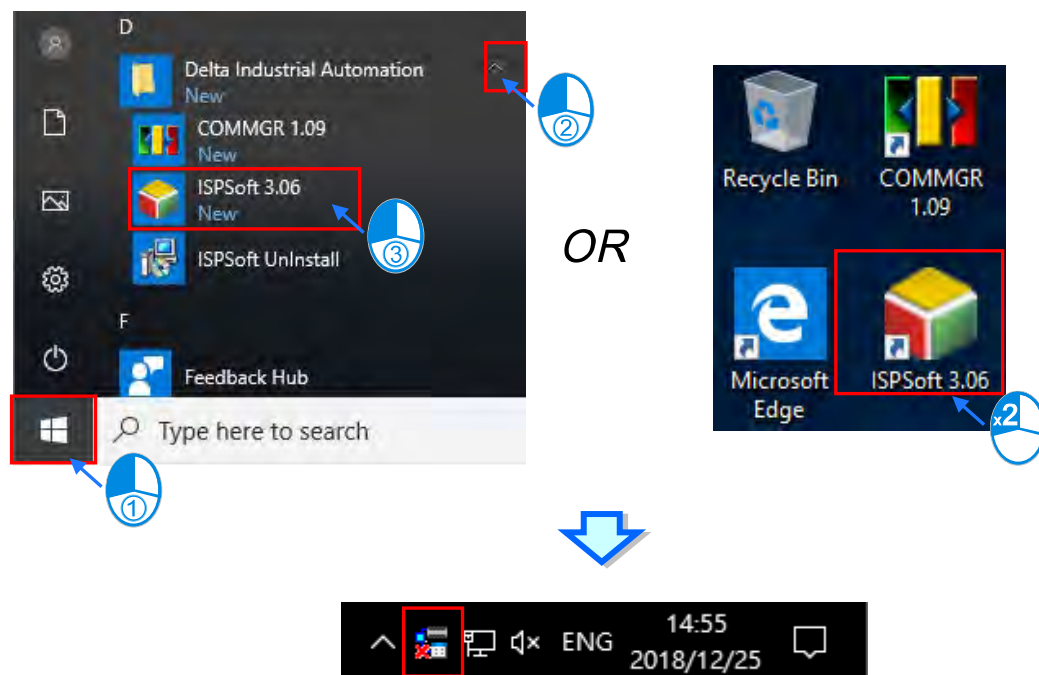
\*1. Please refer to section 1.2 for information about **COMMGR** and installation guide.

\*2. For ISPSOft version 2.0 or above, **COMMGR** is used in the communication structure; while earlier versions adopts the traditional ways of connection.

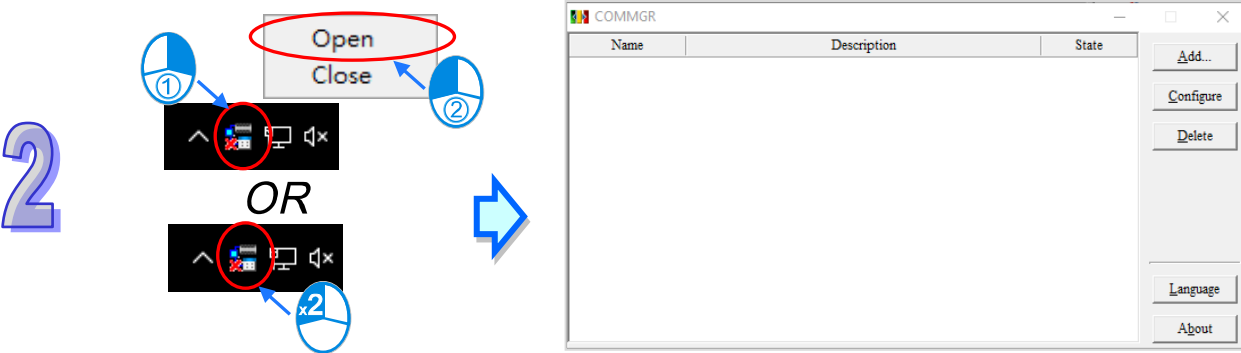
### 2.4.1 Start/Close COMMGR



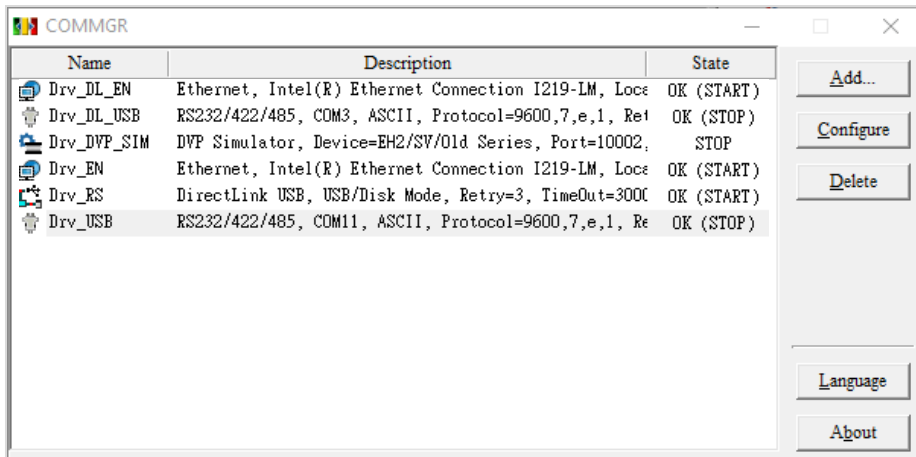
When **COMMGR** is installed, its shortcut can be found in the Programs of a PC **Start** menu. Users need to click the shortcut to start **COMMGR** only for first-time installation. Afterwards, when starting a computer, the **COMMGR** is automatically started and remains in the Windows system; however, when its icon is not shown, please click the shortcut of **COMMGR** listed in the Programs.



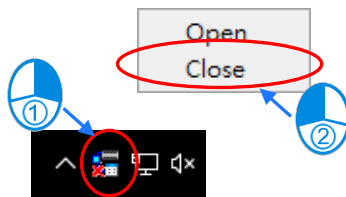
To start **COMMGR**, select its icon and double-click **Open** or right-click the icon to open the program on Windows PC.



The following image is a display of the **COMMGR** status window. In the middle of the window contains description of drivers established for communication. The ISPSoft assigns a driver from the description box to form connection with PLCs; on the right are buttons for managing these drivers. For more information, please read the next section.



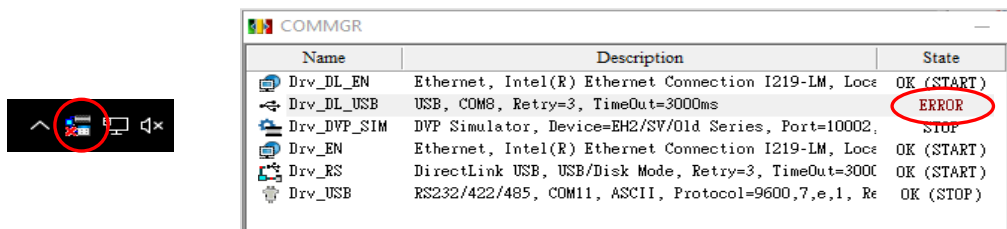
Users can close the **COMMGR** window by clicking **×** or **-** in the upper right corner, but the program still exists in Windows; to close **COMMGR** completely, right-click the **COMMGR** icon displayed in the system and choose **Close**.



## 2.4.2 Driver Management for COMMGR

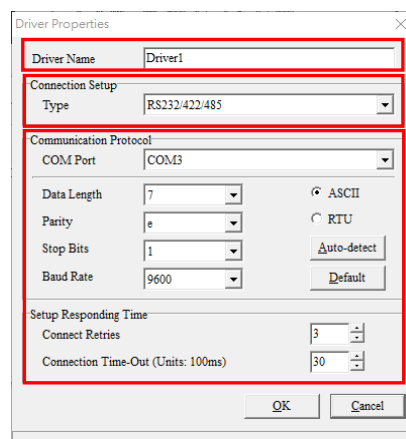
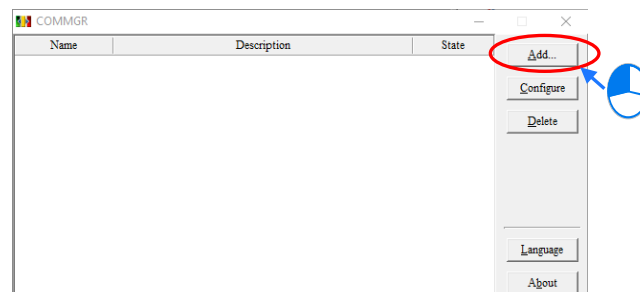


The driver in **COMMGR** act as a channel that connects the program and communication port. When users add a new driver and has setup the communication parameters, the COMMGR forms a channel connecting the assigned communication port in the driver and when the computer reboots, the **COMMGR** will automatically start the driver. However, when the channel for connecting the driver is not properly functioning, for instance, the existed network card or USB cable is removed, the **COMMGR** will automatically stop the driver and **ERROR** status is displayed. Meanwhile, a small red 'X' appears on the COMMGR icon of the Windows system tray. When connection restores, the status for the driver also returns to **OK**.



## 2.4.3 Creating Connection Channel - Add Driver

Click **Add** on the right side of the **COMMGR** window to start settings concerning communication parameters.



Quick steps to set up a driver:

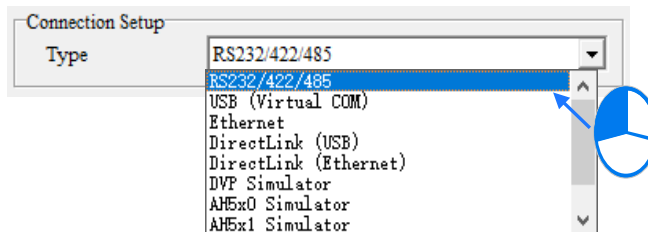
### (1) Driver Name Setup

Type the **Driver Name** in the box. Users can input maximum of 31 characters, and special marks such as “\_” but not \*, #, ?, \, %, @ in the name box.



### (2) Connection Setup

Select a desired connection **Type** from the drop-down list. The following is a list of the connection types supported by COMMGR.



#### ➤ **RS232/422/485**

Uses COM Port to communicate with PLC hosts.

#### ➤ **USB (Virtual COM)**

Some PLC hosts provides USB port, so a PC and PLC host can directly connect through USB. However, before using this connection method, please make sure that the software for the USB driver is installed in the computer. For more information on methods of installation, please refer to appendix A or other PLC user manuals.

#### ➤ **Ethernet**

Uses Ethernet to communicate with PLC hosts.

#### ➤ **DirectLink (USB) & DirectLink (Ethernet)**

Used by Delta human-machine interfaces (HMI) for connection. Under normal connection of PLC and HMI, a computer uses an USB or Ethernet to connect with the HMI, and indirectly builds communication with PLC host. Please refer to Delta HMI user manuals for more information on connection and notifications.

#### ➤ **DVP Simulator & AH5x0 Simulator & AH5x1 Simulator & AS300 Simulator & AS200 Simulator**

Serves as virtual channels for simulators regarding all PLC types, but must use corresponding devices for connection in the ISPSOft.

### (3) Communication Protocol Setup

Setup the communication protocol based on the selected connection type. Each type of connection has its own communication protocol setting. The following section will introduce the different types of communication parameter settings.

Communication Protocol

COM Port: COM3

Data Length: 7

Parity: e

Stop Bits: 1

Baud Rate: 9600

Radio buttons:  ASCII,  RTU

Buttons: Auto-detect, Default

Setup Responding Time

Connect Retries: 3

Connection Time-Out (Units: 100ms): 30

### 2.4.3.1 Communication Parameter Setting - RS232/422/485

Before starting the setup of RS232/422/485 driver, make sure:

- (a) The software for driver installation can be found in the computer.
- (b) The PC and PLC hosts has now established connection and is functioning.

① Driver Name: Driver1

② Connection Setup Type: RS232/422/485

③ Communication Protocol COM Port: COM3

④ Data Length: 7, Parity: e, Stop Bits: 1, Baud Rate: 9600. Radio buttons:  ASCII,  RTU

⑤ Setup Responding Time: Connect Retries: 3, Connection Time-Out (Units: 100ms): 30

- (1) Type the **Driver Name**. To add special marks, only '\_' can be used .
- (2) For **Connection Setup**, select **RS232/422/485** from the drop-down list for **Type**.
- (3) For **COM Port**, select the desired port for communication from the drop-down list. The list contains complete information regarding COM Port number and device name, which is the same as in **Windows Device Manager**. If (a) and (b) mentioned above is confirmed, users will be able to select the COM Port.

COM Port: COM5

Intel(R) Active Management Technology - SOL (COM3)

ELTIMA Virtual Serial Port (COM11)

Prolific USB-to-Serial Comm Port (COM5)

- (4) The data format for communication is **ASCII** or **RTU**.
- (5) The communication protocol setting of the COM Port is the same as its connected device. When **Default** is clicked, the communication protocol parameters returns to defaults.

If not sure about the communication protocol of the connected device, click **Auto-detect** to find out the parameter. When detection is successful, users can connect the device to a RS232



communication port selected with a RS232 cable, and click to automatically detect the communication protocol. If the communication protocol is successfully detected, its parameters will be automatically filled in the corresponding boxes in the section; however, the system will not auto-detect parameters for **COM Port** and **ASCII / RTU**, therefore, please confirm these two parameters before clicking auto-detect.

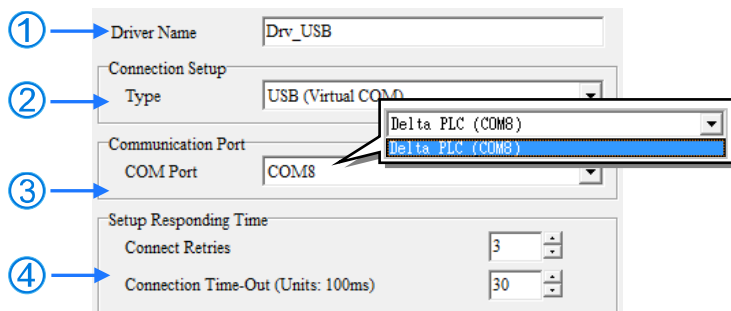
(6) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

\*. When changing a hardware device connected to a PC, users need to close the setting page concerning communication driver properties and re-start in order to obtain the new hardware communication device.

### 2.4.3.2 Communication Parameter Setting - USB (Virtual COM)

Before starting the setup of USB type drivers, make sure:

- (a) The software for driver installation can be found in the computer.
- (b) The PC and PLC hosts has now established connection and is functioning.



- (1) Type the **Driver Name**. To add special marks, only ‘\_’ can be used.
- (2) For **Connection Setup**, select **USB (Virtual COM)** from the drop-down list for **Type**.
- (3) For **COM Port**, select the desired port for communication from the drop-down list. If (a) and (b) mentioned above is confirmed, the list will display the host device name and COM Port number.
- (4) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

\*. For more information regarding USB drivers in PLC hosts, please refer to appendix A .

### 2.4.3.3 Communication Parameter Setting - DirectLink (USB)

The screenshot shows a configuration window for DirectLink (USB). It has four numbered callouts: 1 points to the Driver Name field containing 'Drv\_DL\_USB'; 2 points to the Connection Setup Type dropdown menu showing 'DirectLink (USB)'; 3 points to the HMI USB Mode section where 'USB / Disk Mode' is selected; 4 points to the Setup Responding Time section, specifically the 'Connect Retries' spinner set to 3 and the 'Connection Time-Out (Units: 100ms)' spinner set to 30.

- (1) Type the **Driver Name**. To add special marks, only '\_' can be used.
- (2) For **Connection Setup**, select **DirectLink (USB)** from the drop-down list for **Type**.
- (3) Choose from the **HMI USB Mode**. When selecting **CDC Mode**, choose the desired COM Port from the drop-down list. For more methods regarding connection, please refer to Delta HMI user manuals.
- (4) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

### 2.4.3.4 Communication Parameter Setting - Ethernet

The screenshot shows a configuration window for Ethernet. It has five numbered callouts: 1 points to the Driver Name field containing 'Drv\_EN'; 2 points to the Connection Setup Type dropdown menu showing 'Ethernet'; 3 points to the Ethernet Card Description dropdown menu showing 'Intel(R) Ethernet Connection I219-LM'; 4 points to the IP Address Setting table; 5 points to the Setup Responding Time section, specifically the 'Connect Retries' spinner set to 3 and the 'Connection Time-Out (Units: 100ms)' spinner set to 30.

IP Address	Port	Label	Type
192.168.2.56	502	Main_Controller	AHCPU530-EN
192.168.2.40	502	CMC-EIP01	CMC-EIP01
192.168.2.211	502	DVP32ES2-E	ES2-E
192.168.2.210	502	DVP12SE	SE

- (1) Type the **Driver Name**. To add special marks, only '\_' can be used.
- (2) For **Connection Setup**, select **Ethernet** from the drop-down list for **Type**.
- (3) For **Ethernet Card**, select the desired network card from the drop-down list. The current IP address is shown below the description.

(4) In the **IP Address Setting**, setup the communication device IP address and name.

- Click **Search** and the result for devices in the same network are shown in the box.
- Click **Add** for a new IP address and input information regarding each cell.

- 1 IP address: The device IP address.
- 2 Port: Used during communication, default setting is 502.
- 3 Label: The name of the device.
- 4 Type: The device model type that can be found through clicking **Search**.

- Select an IP address and click **Del** or press Delete on the keyboard to remove it from the list.

(5) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

\*. When changing an internet device connected to a PC, users need to close the setting page concerning communication driver properties and re-start in order to obtain the new internet device.

### 2.4.3.5 Communication Parameter Setting - DirectLink (Ethernet)

1 Driver Name: Drv\_DL\_EN

2 Connection Setup Type: DirectLink (Ethernet)

3 Ethernet Card Description: Intel(R) Ethernet Connection I219-LM  
192.168.2.95

4 IP Address Setting

IP Address	Port	Label	Type
192.168.2.220	502	HMI_1	

5 Setup Responding Time

Connect Retries: 3

Connection Time-Out (Units: 100ms): 30

(1) Type the **Driver Name**. To add special marks, only ‘\_’ can be used.

(2) For **Connection Setup**, select **DirectLink (Ethernet)** from the drop-down list for **Type**.

(3) For **Ethernet Card**, select the desired network card from the drop-down list. The current IP address is shown below the description.

(4) In the **IP Address Setting**, setup the communication device IP address and name.

- Click **Search** and the result for devices in the same network are shown in the box.

- Click **Add** for a new IP address and input information regarding each cell.
    - ① IP address: The device IP address.
    - ② Port: Used during communication, default setting is 502.
    - ③ Label: The name of the device.
    - ④ Type: The device model type that can be found through clicking **Search**.
  - Select an IP address and click Del or press Delete on the keyboard to remove it from the list.
- (5) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.
- \*. When changing an internet device connected to a PC, users need to close the setting page concerning communication driver properties and re-start in order to obtain the new internet device.

#### 2.4.3.6 Communication Parameter Setting - DVP Simulator

The screenshot shows a configuration window for a DVP Simulator. It contains the following fields and settings:

- Driver Name:** Drv\_DVP\_SIM
- Connection Setup:** Type is set to DVP Simulator.
- Simulator:** Port Number is set to 10004.
- Device Setting:** Device is set to EH2/SV/Old Series.
- Setup Responding Time:**
  - Connect Retries: 3
  - Connection Time-Out (Units: 100ms): 30

- (1) Type the **Driver Name**. To add special marks, only ‘\_’ can be used.
- (2) For **Connection Setup**, select **DVP Simulator** from the drop-down list for **Type**.
- (3) Input assigned **Port Number**.
- (4) For **Device Setting**, DVP simulators include **EH2/SV/Old Series**, **EH3/ EH3-L/SV2**, **ES2/ EX2/SA2/SX2/MC**, **SE** and **SS2**.
- (5) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

- \*. When ES, SS, EX, VFD, EC devices download the DVP simulator, these devices will change to EH device once uploaded.
- \*. Regarding TP devices, please select SS2 series as simulator.

### 2.4.3.7 Communication Parameter Setting - AH5x0 & AH5x1 Simulator

- (1) Type the **Driver Name**. To add special marks, only '\_' can be used.
- (2) For **Connection Setup**, select **AH5x0 Simulator** or **AH5x1 Simulator** from the drop-down list for **Type**.
- (3) Input assigned **Port Number**.
- (4) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

### 2.4.3.8 Communication Parameter Setting - AS Simulator

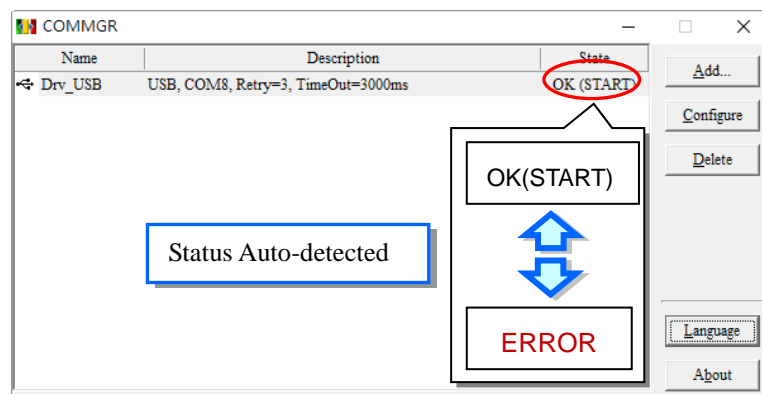
- (1) Type the **Driver Name**. To add special marks, only '\_' can be used.
- (2) For **Connection Setup**, select **AS200 Simulator** or **AS300 Simulator** from the drop-down list for **Type**.
- (3) Input assigned **Port Number**.
- (4) To setup parameters concerning responding time. For **Connect Retries**, users need to setup the number of retry once connection error occurs, while for **Connection Time-Out** setting, the parameter concerns the time interval between retries.

**Additional remark**

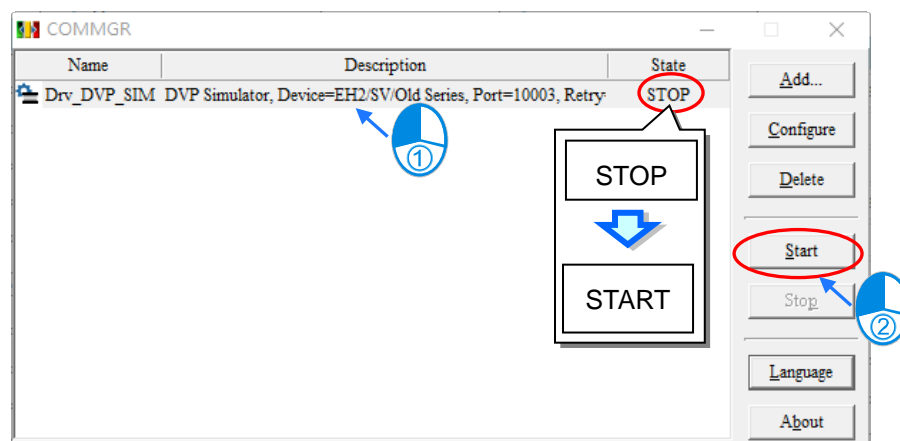
- (a) To use simulator functions, we suggest using ISPSOft V3.05 or later versions.
- (b) When using simulators, users need to be aware that they do not support all functions and commands.

**2.4.4 Creating Connection Channel - Start/Stop a Driver**

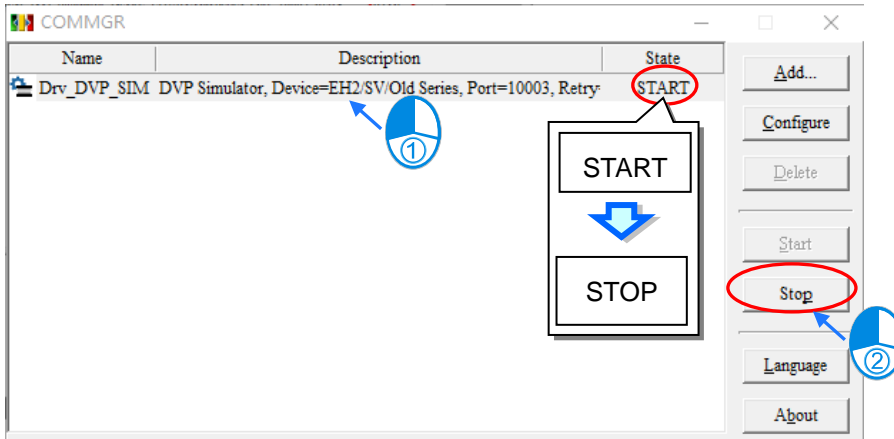
For USB driver, users do not need to operate because COMMGR will auto-detect for the connection when a device is properly connected to a PC.



Users must enable a simulator first to start execution. By choosing the desired simulator driver from the list in the COMMGR window and click **Start** button on the right, you will see **START** shown in the State column if the simulator driver is successfully enabled.

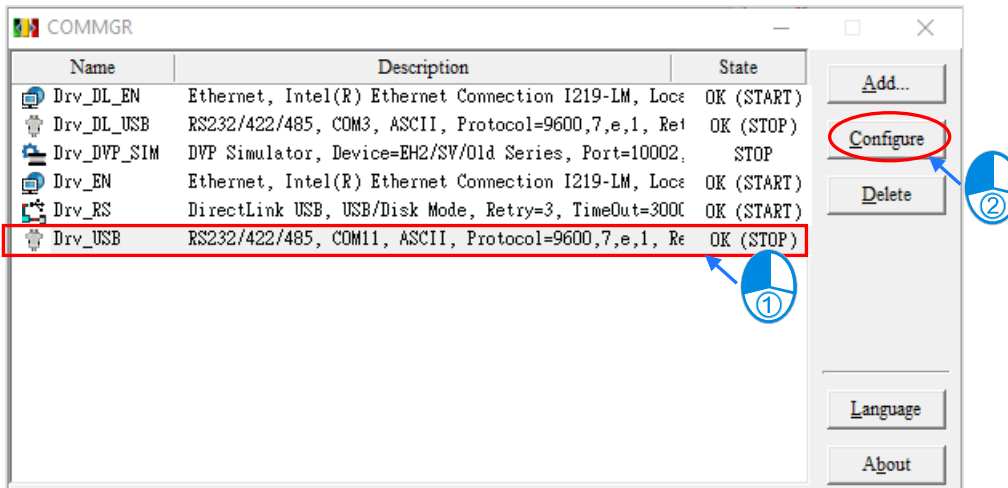


To stop a driver, select the desired driver on the list in **COMMGR** window and click **Stop** button on the right, you will see **STOP** shown in the State column if the simulator drive is stopped.

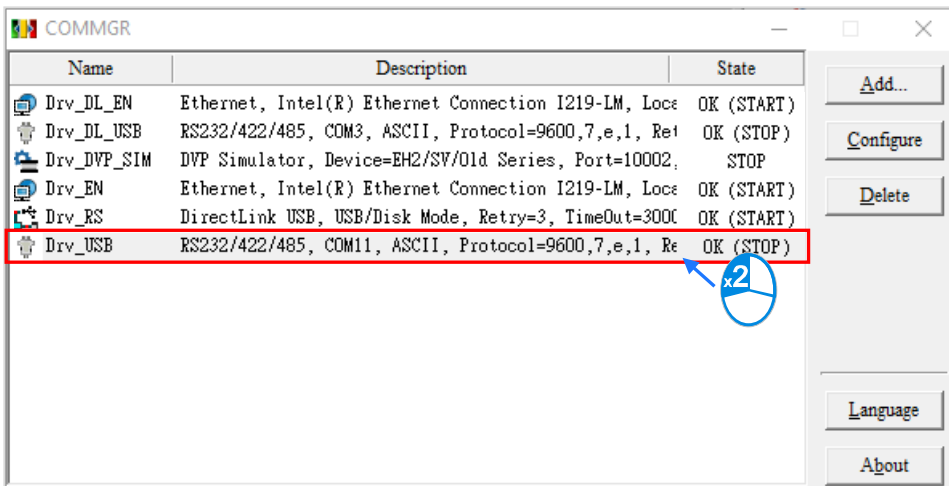


### 2.4.5 Creating Connection Channel - Configure/Delete a Driver

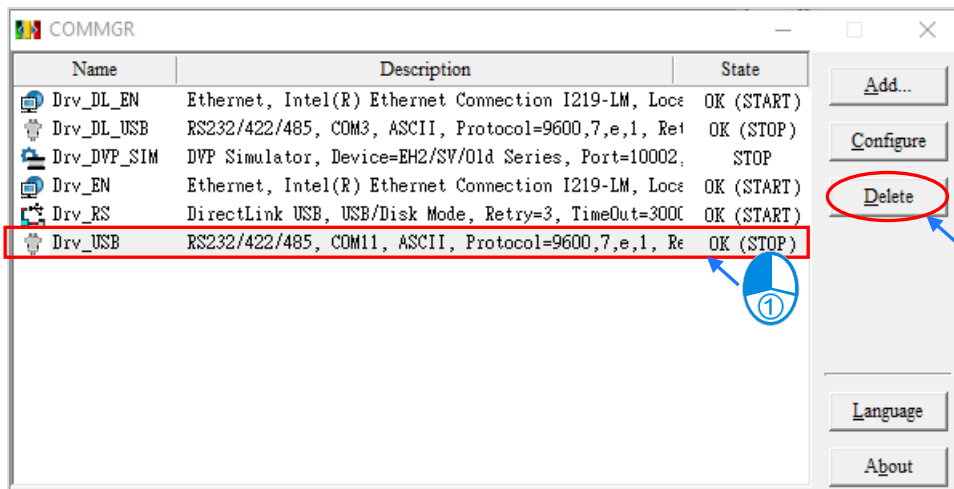
To modify the parameter of a particular driver, please click **Configure** button on the right or double-click the selected driver to open a setup window for modification.



OR



To delete a driver, select the desired driver and click **Delete** button on the right or press DEL on the keyboard for removal.



2

### 2.4.6 Creating Connection Channel - Simulator with Operating Panels

When the state of a simulator shows **START**, an operating panel (see below) pops-up to demonstrate the current states of AH5x0 and AH5x1 simulators and provide STOP or RUN mode for users to control.



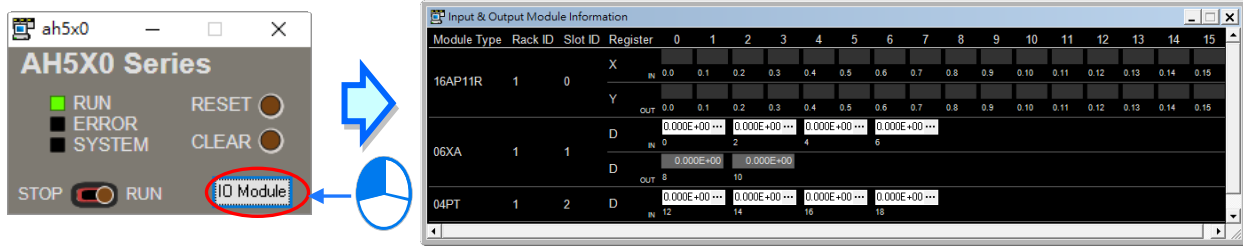
The **IO Module** button of a simulator can open an IO module panel. However, modules must be configured in HWCONFIG and please refer to section 3.8.4 for parameter download to hosts.



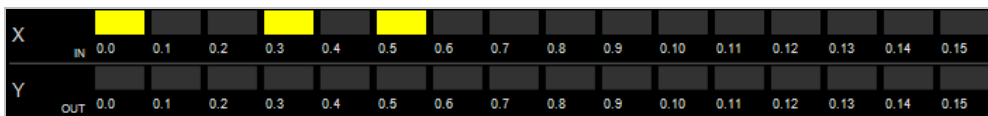
Information: Rack 1						
Slot No.	Label	MDS Version	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	01.00.00	Basic CPU module building with Ethernet, RS4	None	None	
0	AH16AP11P-5A	-	8 x DI VDC, 8 x DO PNP VDC	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1	AH06XA-5A	01.00.00	4 x 16 bit AI, 2 x 16bit AO	D0 ~ D7	D8 ~ D11	
2	AH04PT-5A	01.00.00	4 x 3/4 wires RTD input 0.1 degree Celsius/0.1	D12 ~ D19		



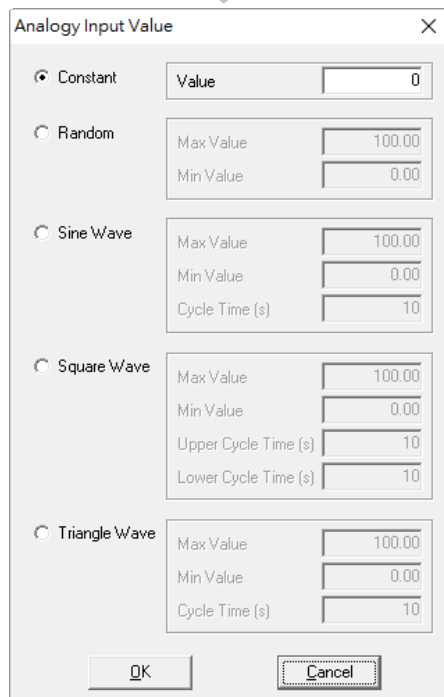
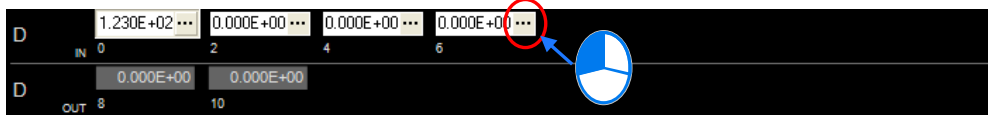
Users can switch to RUN mode and view the same HWCONFIG module configuration and state on IO module panel.



The following chart demonstrates the digital input and output registers area. The Y device offers only output status but cannot perform any operation; while X device offers users to click and change the ON/OFF status of each contact for external input simulation.

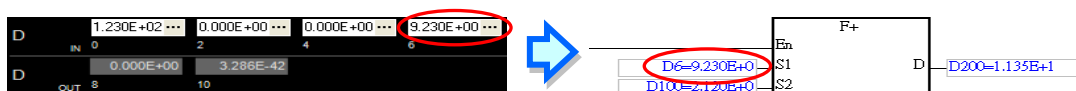


The following chart shows analog input and output registers area. Each column requires 32-bit floating-point format and corresponds to every channel of the actual module; for D8 and D10 (see below) corresponds to analog output channel - D device, the output are for viewing only; while D0, D2, D4 and D6 corresponds to analog input channel - D device. Users can click to open the setup window (see below) for analog input value.

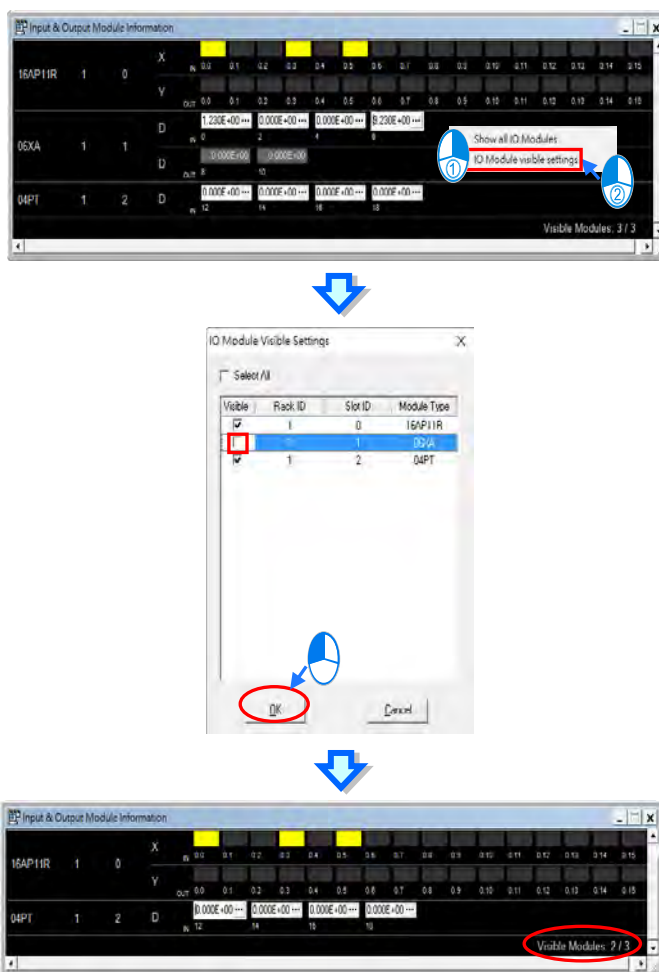


Signal Type	Description
Constant	A constant signal with fixed value.
Random	To assign the maximum and minimum value of random signal.
Sine Wave	To assign the maximum or minimum value and signal cycle of sine wave.
Square Wave	To assign the maximum or minimum value, ON and OFF times of cyclic square wave.
Triangle Wave	To assign the maximum or minimum value and signal cycle of triangle wave.

When setting is complete, the column (channel) will continue to send data to COMMGR based on the signal mode setup. While ISPSOft can conduct related tests when capturing simulator signals through COMMGR.

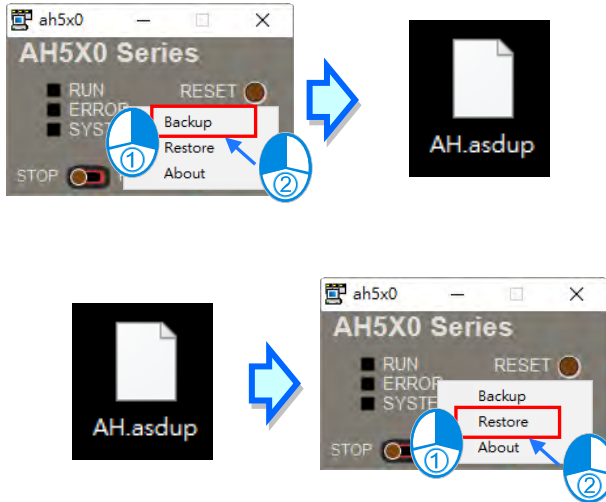


Users can right-click the IO module panel and select **Show All IO Module** to display all modules. To hide the modules, click **IO Module Visible Settings** and choose the module to hide from the list of module types then click **OK**. You will be able to view the current state on the right bottom corner of the window. (See below)

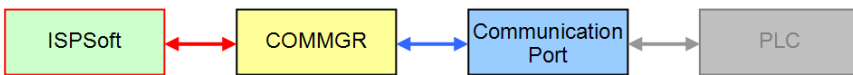


The AH5x0 and AH5x1 simulators also provides backup function which combines parameters and program backup into an \*.asdup file then use this file and restore in other identical simulators. (See below)

2

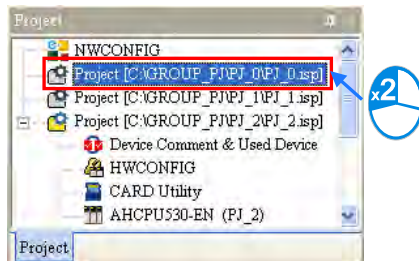


### 2.4.7 Creating Connection between ISPSOft and COMMGR

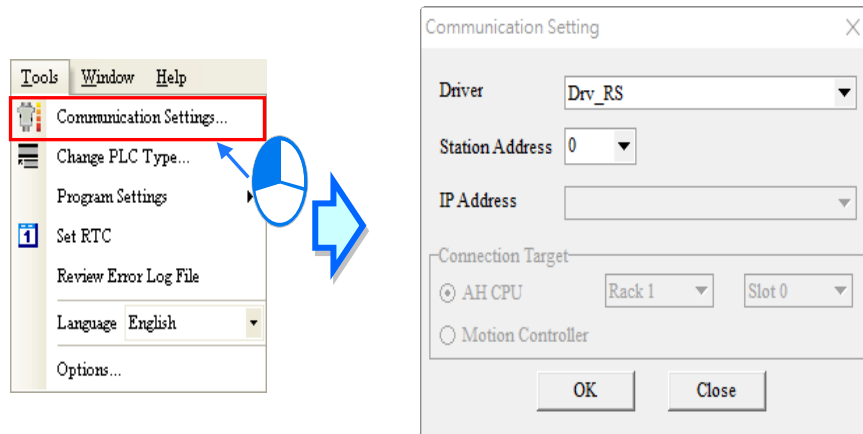


When a driver is created and also enabled in COMMGR, users can assign the driver and use it in ISPSOft. After the setting is complete, a connection between ISPSOft and COMMGR is formed. For single project setup, each requires an assigned communication, thus, under the structure of **group project**, users must open each single project and complete the setups.

- (1) First, please enable the desired project for communication setup, but if under a single project structure, you can skip this step.

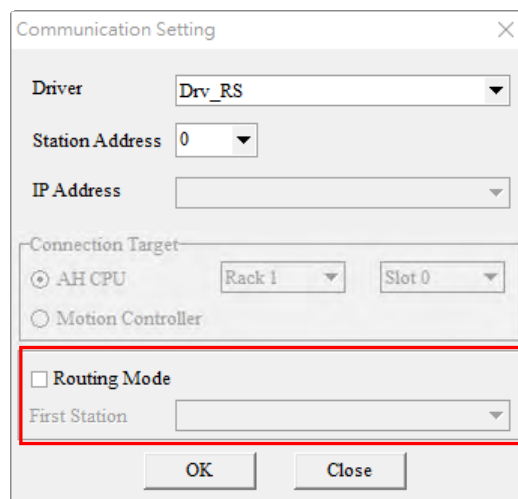


- (2) Choose **Communication Settings** from **Tools** menu in ISPSOft and its setting window will appear.

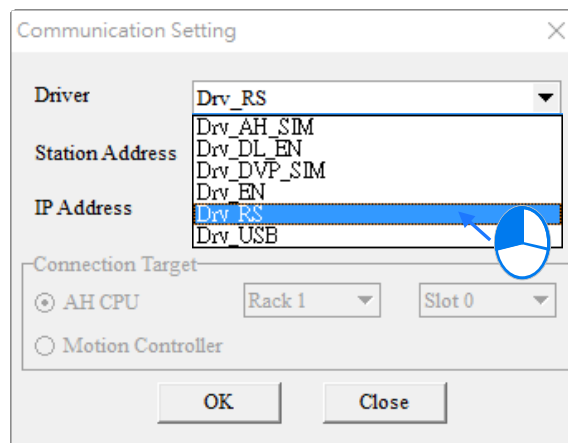


2

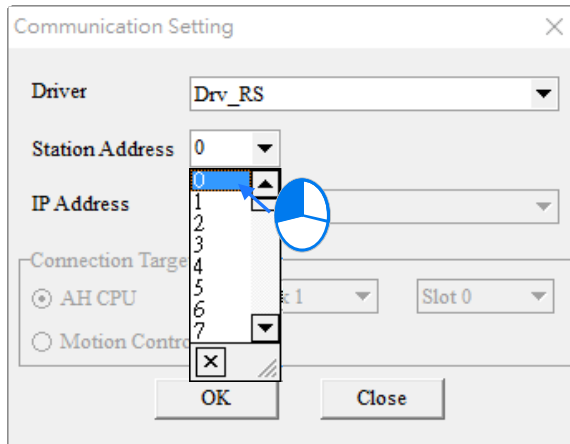
- (3) When AH5x0 series is selected as the connection target for **group project**, the option **Routing Mode** is added and used together with **NWCONFIG**. Please refer to chapter 20 for more information.



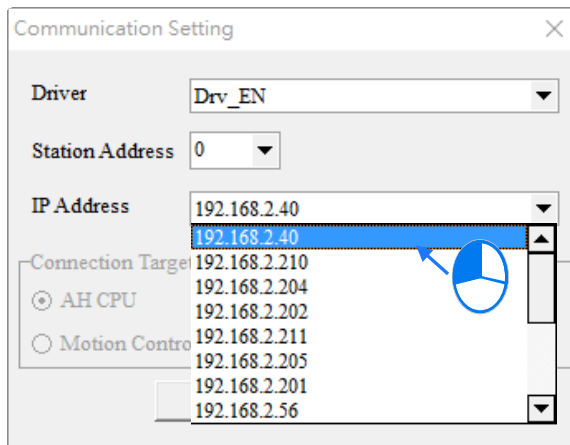
- (4) Select the desired **Driver** from the drop-down list in **Communication Setting** window.



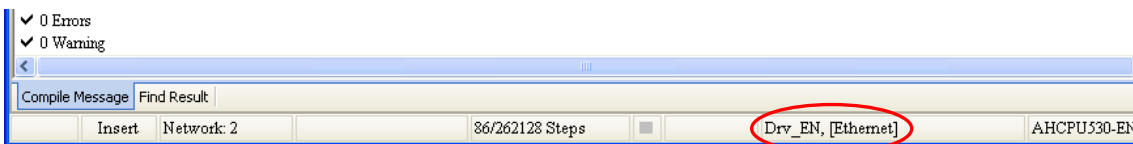
- (5) Select the PLC station address from the drop-down list that is to connect with the PC or choose 0 for station address (broadcast).



- (6) When driver type is **Ethernet** or **DirectLink (Ethernet)**, users can select the desired **IP Address** to connect with PLC hosts from the drop-down list.



- (7) Click **OK** once all settings are complete. Meanwhile, the current connected driver information will appear in the ISPSOft status bar.



**Additional remark**

If COMMGR setting is modified, then communication settings in ISPSOft must be re-selected for proper connection.

### 2.4.8 Connecting PLC (Host) and Communication Port

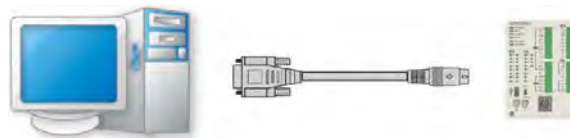


Users can use communication cables to connect PLC hosts and assigned communication ports. The following introduces most commonly used connections and useful notes. For more details, please refer to any types of PLC user manuals.

#### 1. DVP series PLC (RS232)

Adopts **Delta communication cable** to connect a computer with a PLC host. Select RS232 for COMMGR driver type.

**Note:** Before connecting, please confirm that the setups in COMMGR drive concerning RS232 parameters is the same as in PLC hosts.



#### 2. DVP-SX2 series PLC (USB)

A DVP-SX2 series PLC provides a type B mini USB interface. Users can use the USB cable to connect a PLC host with a computer. However, since the USB framework in SX2 series contains USB converting to RS232 (circuit built-in), therefore, the internal function adopts RS232 mode and is also selected as the COMMGR driver type.

**Note:**

- (a) Please confirm that the USB driver is properly installed in SX2 series. For more details, please refer to Appendix A.
- (b) Before connecting, please confirm that the setups in COMMGR drive concerning RS232 parameters is the same as in PLC hosts.



#### 3. AH/AS series PLC (USB)

Adopts USB cable to connect a computer with a PLC host. The USB port of the host is a type B mini USB. Select USB (Virtual COM) for COMMGR driver type.

**Note:** Please confirm that the USB driver is properly installed in AH/AS series. For more information, please refer to Appendix A.

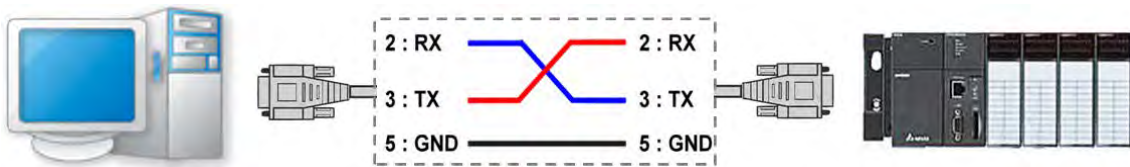


4. AH series PLC (RS232/422/485)

The COM port for AH series PLC host is a standard type, therefore, please use RS232 communication of RX and TX lines or an adapter that has jumper function to proceed the connection with a computer. Please select RS232/422/485 for COMMGR driver type.

**Note:**

- (a) Before connecting, please confirm that the setups in COMMGR drive concerning RS232 parameters is the same as in PLC hosts.
- (b) The COM interface of AH series provides many modes including RS232/RS485/RS422. Before connecting, please confirm that current desired mode is correct. (Please refer to follow-up comments.)

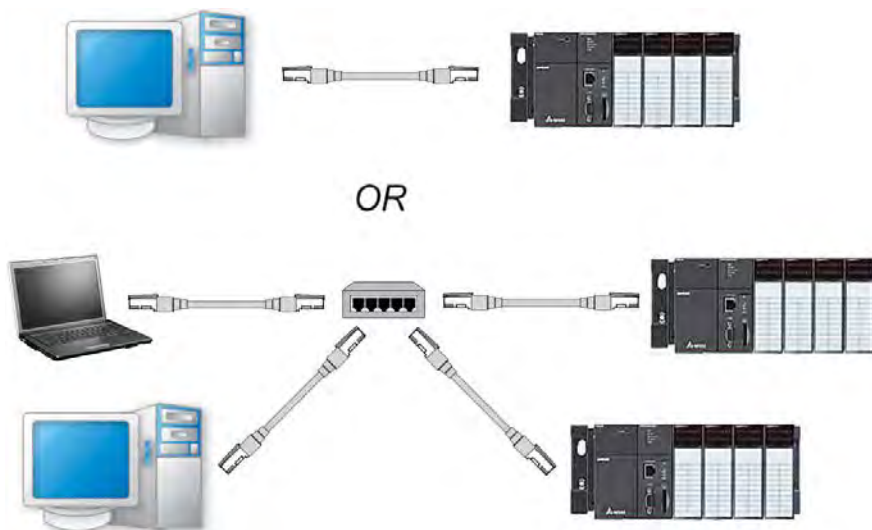


5. AH/AS series PLC, DVPxxMC and DVP-SE (Ethernet)

The AHCPU5xx-EN, AHxxEMC, AS series, DVPxxMC and DVP-SE series PLC hosts contain built-in RJ-45 port (Ethernet) and can connect with a PC through Ethernet. A PC can also connect with a PLC host to the same domain through the hub or uses direct network cable for connection. Please choose Ethernet for COMMGR driver type.

**Note:**

- (a) Before connecting, please check the structure of the internet is properly functioning.
- (b) Check the accuracy of the Ethernet parameter in host CPU. Please refer to follow-up comments

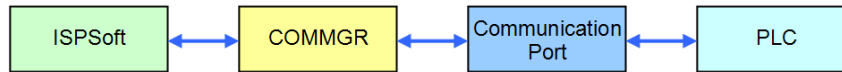


**Additional remark**

For communication parameter defaults regarding all PLC types, please refer to related user manuals.

### 2.4.9 Practical Connection Test

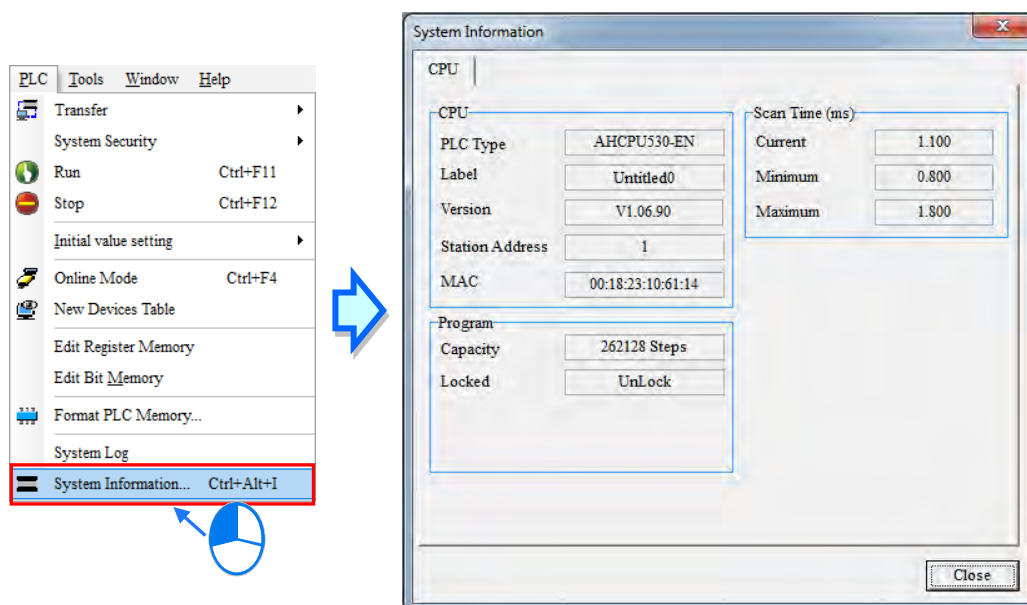
When all the steps mentioned above are completed, users can conduct a simple test to check on the connection with a PLC host is properly functioning.



Before connecting, please check on the following:

- A PLC host and PC is properly connected through communication cable, while the host power is turned on and well-functioning.
- Provides communication device including PC network cards, internet hub and serial port is properly functioning.
- To use the desired COMMGR driver, please check related parameters to be correct and the driver state is **OK**.
- Please check the assigned driver, station number and IP address in **Communication Setting** for ISPSOft is correct.

After confirming the list above, please select **System Information (I)** from **PLC** menu and a pop-up window will appear. However, if the pop-up window concerning system information did not appear, this means connection failure, users will need to check the above list (a) to (d) again.





**MEMO**

2

# Chapter 3 PLC System Configuration & Settings

## Table of Contents

<b>3.1</b>	<b>HWCONFIG 4.0 – Hardware Configuration Tool .....</b>	<b>3-3</b>
3.1.1	Open Your HWCONFIG 4.0.....	3-4
3.1.2	HWCONFIG 4.0 Environment .....	3-6
3.1.3	Module Configurations.....	3-15
3.1.4	Parameter Settings for PLC CPU .....	3-33
<b>3.2</b>	<b>Hardware Configuration Tool – HWCONFIG 3.0 .....</b>	<b>3-66</b>
3.2.1	HWCONFIG Environment .....	3-66
3.2.2	Module Configuration .....	3-69
3.2.3	Managing the Version of a Module .....	3-107
<b>3.3</b>	<b>Parameter Settings for AH500 Series PLC .....</b>	<b>3-110</b>
3.3.1	Open the PLC CPU Parameter Setting Page .....	3-110
3.3.2	Setting CPU Parameters .....	3-111
3.3.3	Setting COM Port Parameters.....	3-120
3.3.4	Setting Ethernet Parameters.....	3-121
3.3.5	Setting Advanced Ethernet Parameters .....	3-122
3.3.6	Data Exchange Setup - COM & Ethernet .....	3-140
<b>3.4</b>	<b>Parameter Settings for AHxxEMC Series PLC .....</b>	<b>3-144</b>
3.4.1	Open the PLC CPU Parameter Setting Page .....	3-144
3.4.2	Options .....	3-146
3.4.3	Data Exchange - COM2 & Ethernet .....	3-157
<b>3.5</b>	<b>Parameter Setting for DVP Series PLC .....</b>	<b>3-162</b>
3.5.1	System Management Tools for DVP Series PLC.....	3-162
3.5.2	Retentive Range.....	3-163
3.5.3	Connected Information.....	3-167
3.5.4	Parameter Setting for DVP-ES3 Series PLC .....	3-168
<b>3.6</b>	<b>Parameter Settings for DVPxxMC Series PLC .....</b>	<b>3-170</b>
3.6.1	Opening the PLC Parameter Setting Window .....	3-170
3.6.2	Options .....	3-171
<b>3.7</b>	<b>Parameter Settings for Modules .....</b>	<b>3-182</b>
3.7.1	Setting Internal Parameters for AH5x0 and AH5x1 Modules .....	3-182
3.7.2	Parameter Settings for AHxxEMC Modules .....	3-187

3

- 3.7.3 Exporting and Importing Parameters for Modules ..... 3-189
- 3.7.4 Updating Parameters for Modules ..... 3-190
- 3.7.5 Parameter Settings for Intelligent Modules ..... 3-190
- 3.8 HWCONFIG Parameter Management and Online Diagnosis 3-191**
  - 3.8.1 Save and Print function ..... 3-191
  - 3.8.2 Purchase Order ..... 3-192
  - 3.8.3 Rack Information List for AH Series ..... 3-193
  - 3.8.4 Download/Upload System Parameters ..... 3-194
  - 3.8.5 I/O Scan ..... 3-196
  - 3.8.6 AH Series Online Diagnosis ..... 3-199
- 3.9 Setting a RTC ..... 3-206**
- 3.10 Setting the Memory in a PLC ..... 3-207**

You can use ISPSOft features and tools to set up configurations and parameters for DVP Series PLC and its entire system. For other series, you need to use HWCONFIG in ISPSOft to accomplish the same operation. The old HWCONFIGE does NOT have a unified interface for all the various PLC types but fortunately things are about to change. We have integrated the assorted interfaces and come up with a whole new version, HWCONFIG 4.0. Now the new HWCONFIG 4.0 is available for AS Series. We will continue to include all the other series in the near future. This chapter will introduce the steps and setups concerning configuration and parameter settings.

Series Version	AS200	AS300	AH5x0	AH5x1	AH560	AHxx EMC	DVP	DVPxx MC
HWCONFIG 4.0	V	V						
HWCONFIG 3.0			V	V	V	V	Only ES3	V

3

### 3.1 HWCONFIG 4.0 – Hardware Configuration Tool

HWCONFIG is one of the tools in ISPSOft for hardware configuration. Its functions include configuration of module racks, parameter settings for modules, download/upload hardware parameters and simple on-line detection and diagnosis function. HWCONFIG 4.0 provides you with different display theme styles, light and dark. Here we use light theme for display examples. Refer to section 3.1.3.20 for more details on theme style settings.

**⚠ The settings will only be effective when the parameters are downloaded to the PLC CPUs. Make sure you download the configuration parameters to the PLC CPU after the setups are complete.**

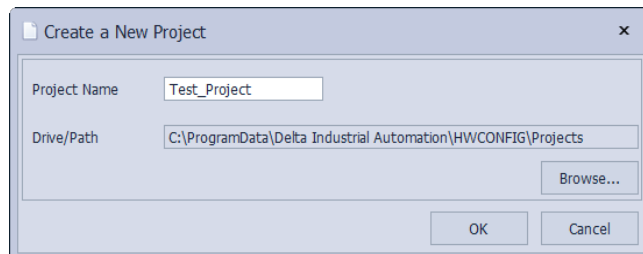
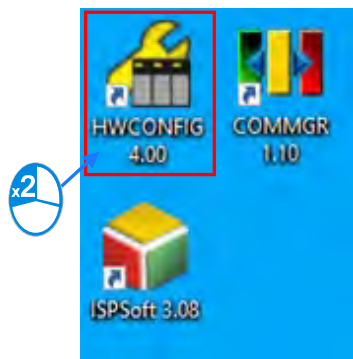
### 3.1.1 Open Your HWCONFIG 4.0

You can open HWCONFIG 4.0 either by clicking its independent shortcut icon as the image shown below or by clicking the setup option below Project in ISPSoft V3.10.

- **Open HWCONFIG 4.0 through its shortcut icon:**

1. Double-click **HWCONFIG** icon to open HWCONFIG software.
2. In **Create a New Project** window, input a **Project Name** and choose a file saving path for the project.
3. Click **OK**.

3



- **Open HWCONFIG 4.0 through ISPSOFT V3.10:**

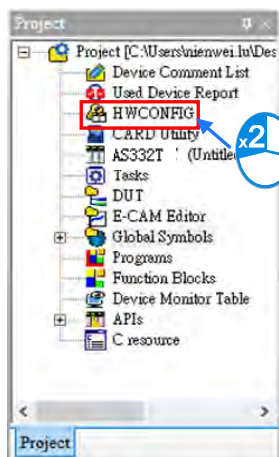
1. Double-click **ISPSOFT** icon to open ISPSOFT software.

2. Open an existing project or create a new project in ISPSOFT. Click *File->New* or click the icon

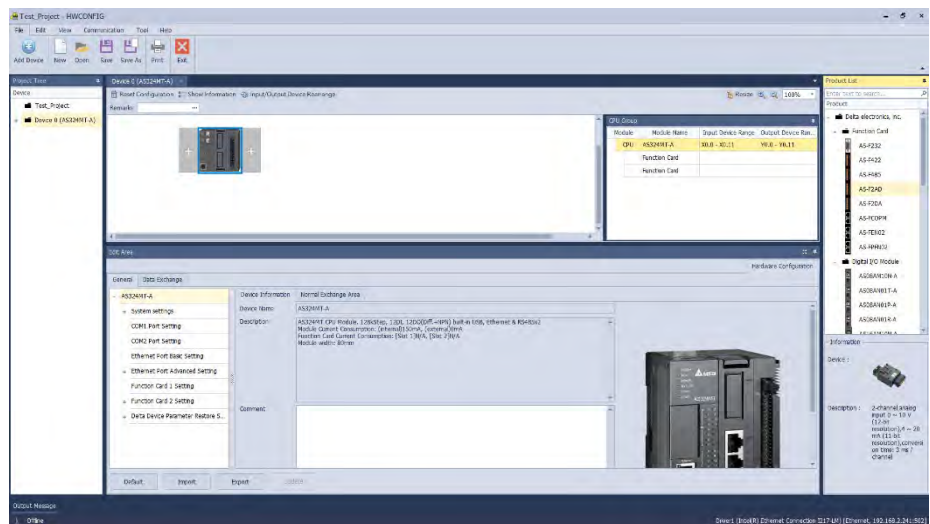


to open the **Create a New Project** window. Input a **Project Name**, select a **Controller Type** and **PLC Type** and then choose a file saving path for the project. Click **OK**.

3. Double-click **HWCONFIG** in the Project section to open HWCONFIG 4.0.

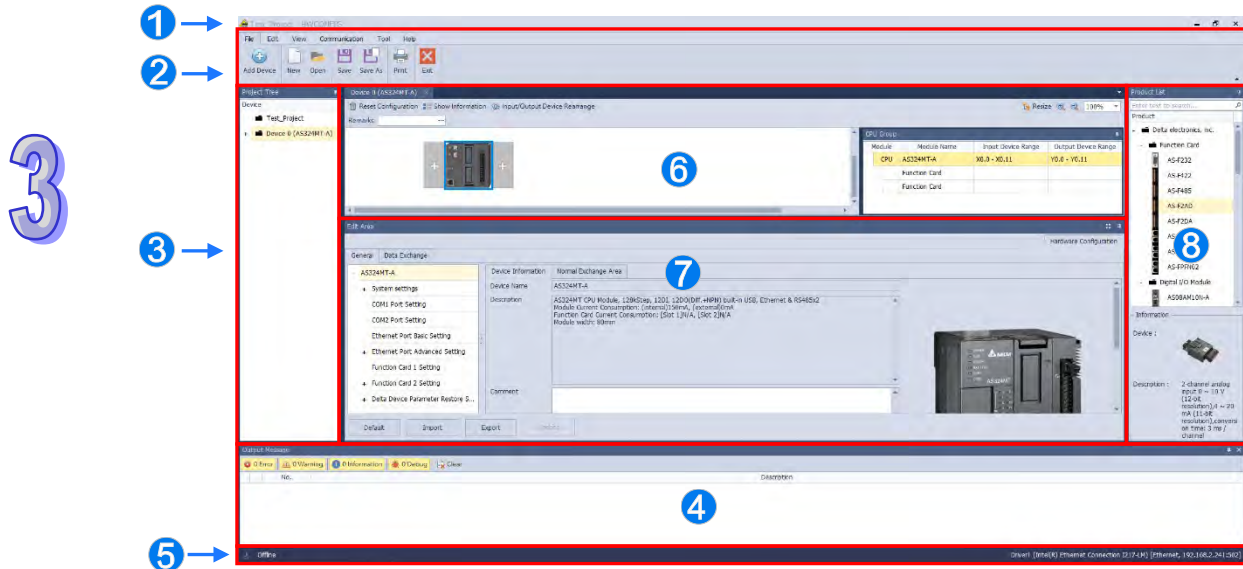


3



### 3.1.2 HWCONFIG 4.0 Environment

HWCONFIG 4.0 Environment is consist of Window Title, Tool Bar, Project Tree, Output Message Section, Status Bar, Hardware Configuration Area, Edit Area and Product List. The options under them may vary according to the device selected.

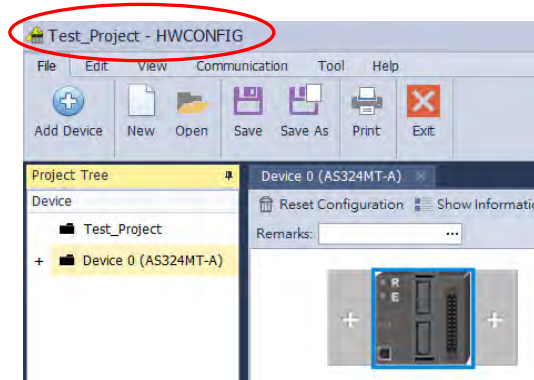


- ❶ **Window Title:** Here shows the current project name.
- ❷ **Tool Bar:** There are 6 tabs on the tool bar.
- ❸ **Project Tree:** Tree-structured projects for easier management
- ❹ **Output Message Section:** Here shows project related information.
- ❺ **Status Bar:** Here shows the connection status and the communication parameters.
- ❻ **Hardware Configuration Area:** You can set up hardware configuration here.
- ❼ **Edit Area:** You can set up parameters for PLC CPU and modules.
- ❽ **Product List:** You can find available hardware here.

### 3.1.2.1 Window Title and Status Bar

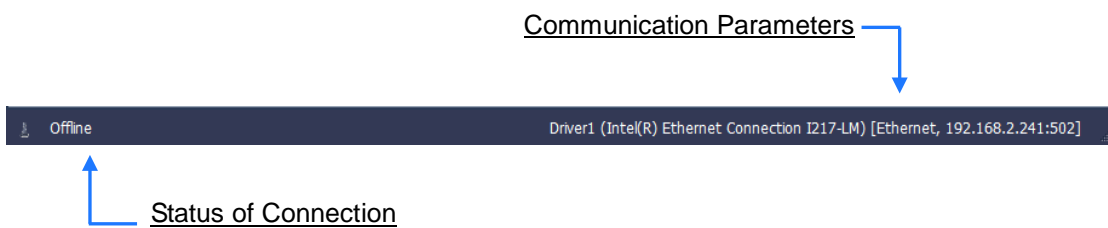
- **Window Title**

After you have created your project, the name of your created project shows in here.



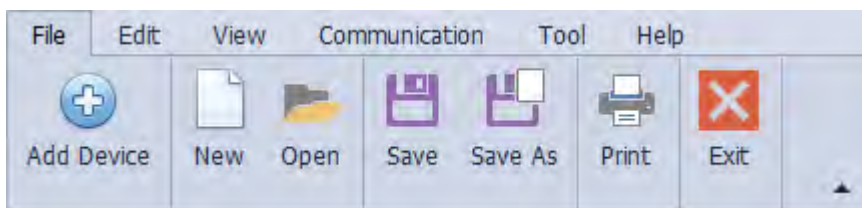
- **Status Bar**

From the Status Bar, you can find the connection status and the communication parameters.



### 3.1.2.2 File Tab

File Tab: Use options under this tab to create, save or print a project.



Item	Description
<b>Add Device</b>	Add a new PLC CPU to the current project. (only available for independent HWCONFIG software)
<b>New</b>	Create a new project. (only available for independent HWCONFIG software)
<b>Open</b>	Open an existing project. (only available for independent HWCONFIG software)
<b>Save</b>	Save a project.
<b>Save As</b>	Save the project to another location or another project name.
<b>Print</b>	Print the hardware configuration.
<b>Exit</b>	Exit HWCONFIG 4.0

3



### 3.1.2.3 Edit Tab

Edit Tab: Use options under this tab to paste, cut, copy or delete to add or remove devices from the hardware configuration area.

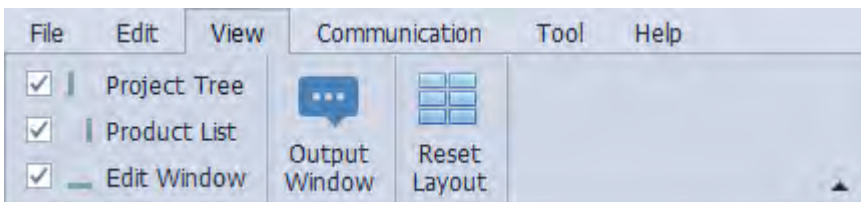


3

Item	Description
<b>Paste</b>	After selecting an added device in the hardware configuration area, click <b>CUT</b> or <b>COPY</b> , and then use <b>Paste</b> to move or add the selected device to a new position.
<b>Cut</b>	After selecting an added device in the hardware configuration area, click <b>CUT</b> to remove the selected device.
<b>Copy</b>	After selecting an added device in the hardware configuration area, click <b>Copy</b> to add the selected device in the hardware configuration area.
<b>Delete</b>	After selecting an added device in the hardware configuration area, click <b>Delete</b> to remove the selected device.

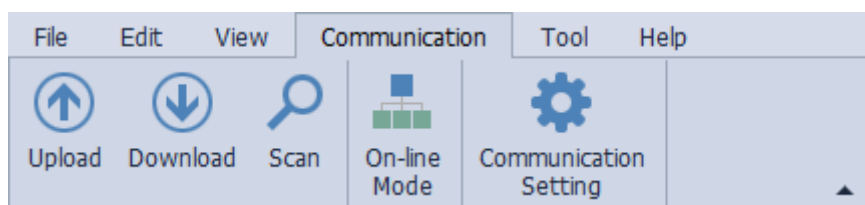
### 3.1.2.4 View Tab

View Tab: Use options under this tab to organize the HWCONFIG display. You can open or close the **Project Tree**, **Product List**, **Edit Window**, **Output Window** or use **Reset Layout** to manage the HWCONFIG layout.



### 3.1.2.5 Communication Tab

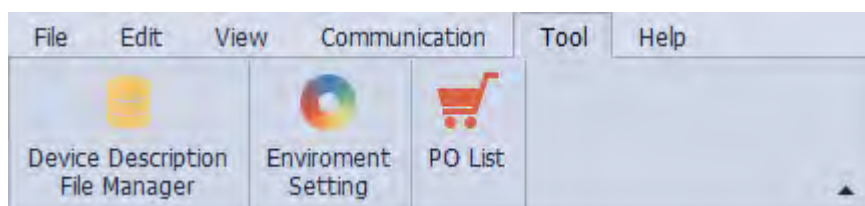
Communication Tab: Use options under this tab to **Upload / Download** parameters, use **Scan** to scan the current connected module configurations and upload to HWCONFIG for further configuration, use **On-line Mode** to monitor the module status as well as its information, and use **Communication** to set up communication.



Item	Description
<b>Upload</b>	Upload PLC parameters and module configurations to HWCONFIG for further configuration.
<b>Download</b>	Download parameters and configuration in HWCONFIG to PLC.
<b>Scan</b>	Scan the current connected module configurations and upload to HWCONFIG for further configuration.
<b>On-Line Mode</b>	Monitor the module status and its information.
<b>Communication Setting</b>	Set up the communication for the PLC.

### 3.1.2.6 Tool Tab

Tool Tab: Use options under this tab to open **Device Description File Manager**, **Environment Setting** and **PO List** for further configuration.



Item	Description
<b>Device Description File Manager</b>	Use the wizard to install / uninstall description files (IODD, DDF).
<b>Environment Setting</b>	General: Enable prompt confirmation, retain last layout setting and manage theme style. Language: Selections are English, Traditional Chinese, Simplified Chinese, and Russian.

<b>PO List</b>	List all the compatible devices that the created project may need for your reference. You can also export the list and
----------------	--

### 3.1.2.7 Help Tab

Tool Tab: Use options under this tab to open **Device Description File Manager**, **Environment Setting** and **PO List** for further configuration.

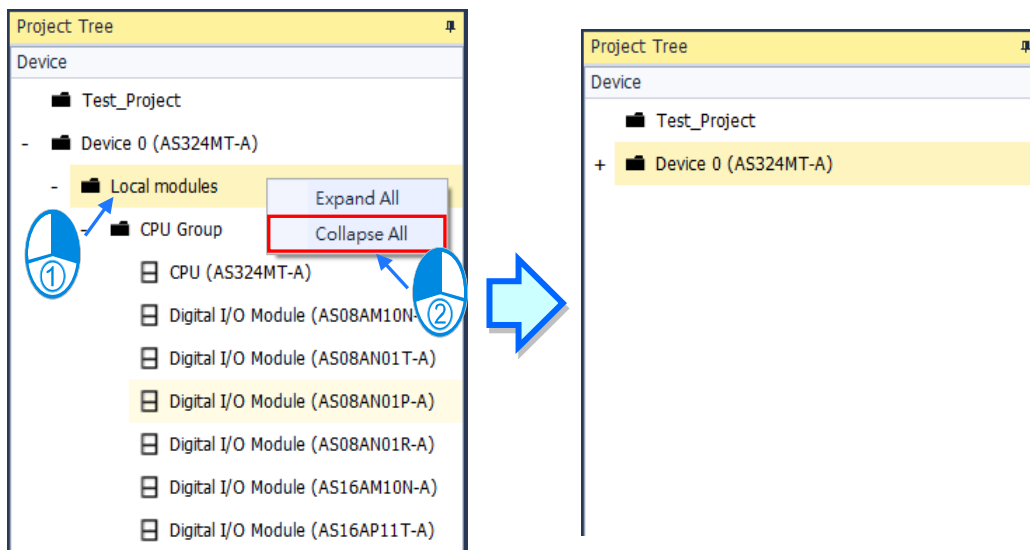
3

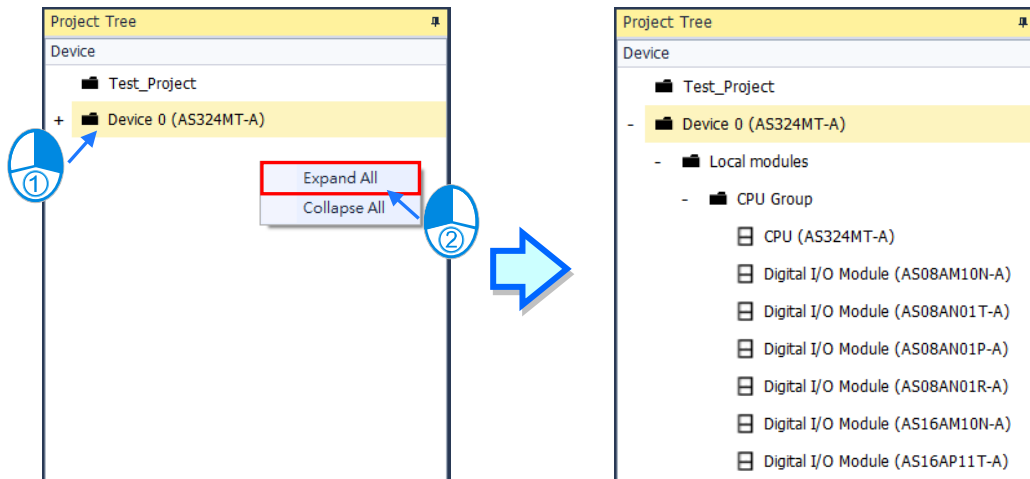


Item	Description
<b>About</b>	Click this button to see the version and the release date of this HWCONFIG.
<b>Help</b>	Click this button to see the user manual.
<b>Revision History</b>	Click this button to see the change history of HWCONFIG.

### 3.1.2.8 Project Tree

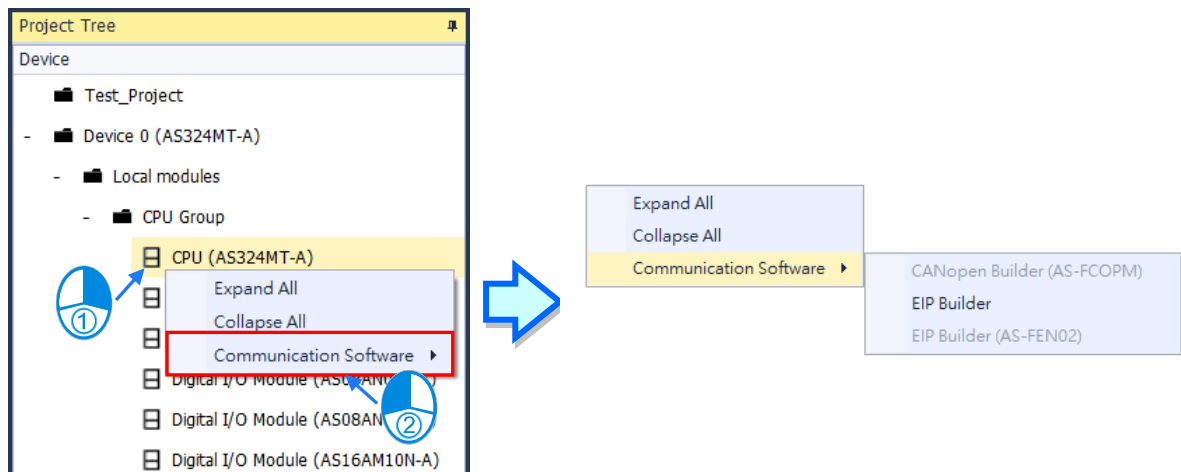
List the added devices of the project in a tree structure for easier management. The **Project Tree** is on the left side of the HWCONFIG by default. You can rearrange the layout; refer to section 3.1.2.4 for more details. You can right-click the item on the Project Tree to see options of **Expand All** or **Collapse All**, use these options to fold or unfold the devices under the item.





3

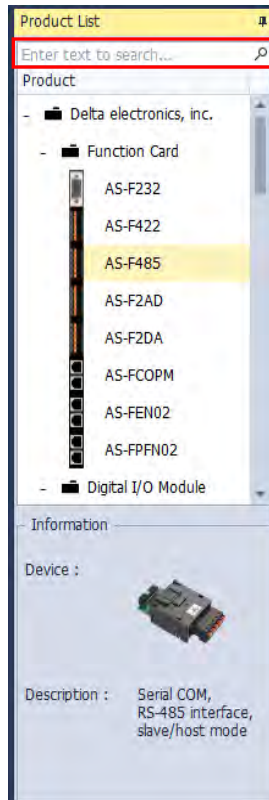
Right-click the **PLC CPU** on the **Project Tree**, you can see **Communication Software** on the context menu. Click **Communication Software** to see which software is available for this project. If the software option is grayed out, you may need applicable function cards to work along with the project.



### 3.1.2.9 Product List

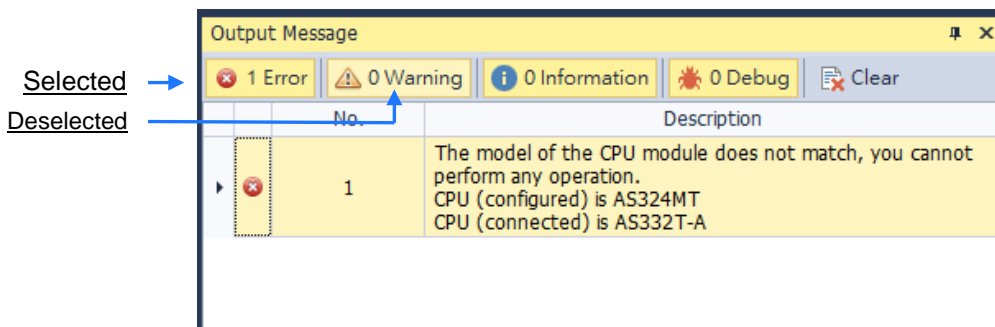
Applicable devices, including modules and function cards for the project, are listed in a tree structure for easier management. The **Project Tree** is on the right side of the HWCONFIG by default. You can rearrange the layout; refer to section 3.1.2.4 for more details. Click to select the device and you can see its information below. You can use the search function by entering a key work in the blank field to search for the device you need.

3



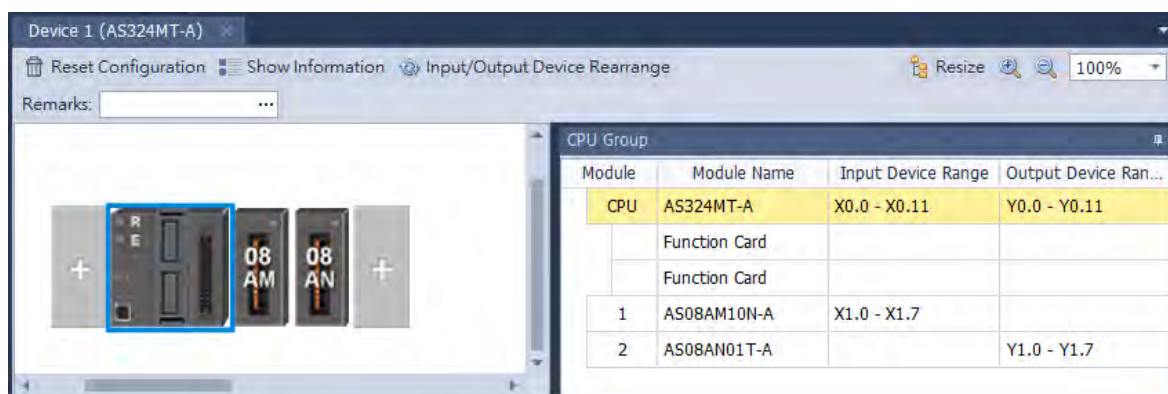
### 3.1.2.10 Output Message

The **Output Message** is on the bottom of the HWCONFIG by default. You can rearrange the layout; refer to section 3.1.2.4 for more details. You can see four types of messages, including **Error**, **Warning**, **Information**, **Debug** and one **Clear** button, in the **Output Message** section. Click to select the message type and you can see its information below. Use **Clear** button to clear all messages.



### 3.1.2.11 Hardware Configuration Area

The **Hardware Configuration Area** is on the right side of the HWCONFIG by default. You can rearrange the layout; refer to section 3.1.2.4 for more details. You can see the configurations of the modules and function card here. You can also use the functional tabs including **Reset Configuration** to set the configurations back to default values, **Show Information** to see the hardware configurations and **Input / Output Device Rearrange** to rearrange the device ranges.



Item	Description
<b>Reset Configuration</b>	Use this button to set the configurations back to default values
<b>Show Information</b>	Use this button to see the hardware configurations
<b>Input / Output Device Rearrange</b>	Use this button to rearrange the device ranges.
<b>Resize</b>	Use this button to set the display of the configuration area back to its default values (shown at 100% and in the center).
<b>+</b>	Use this button to enlarge the display of the configuration area
<b>-</b>	Use this button to shrink the display of the configuration area
<b>%</b>	Enter a precise number for the display of the configuration area
<b>Remark</b>	Use this field to enter notes

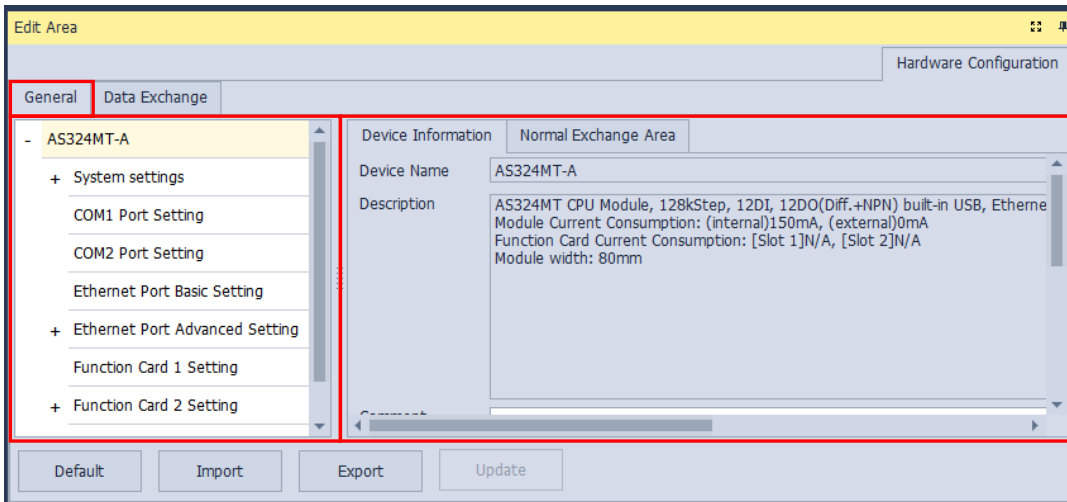
3

### 3.1.2.12 Edit Area

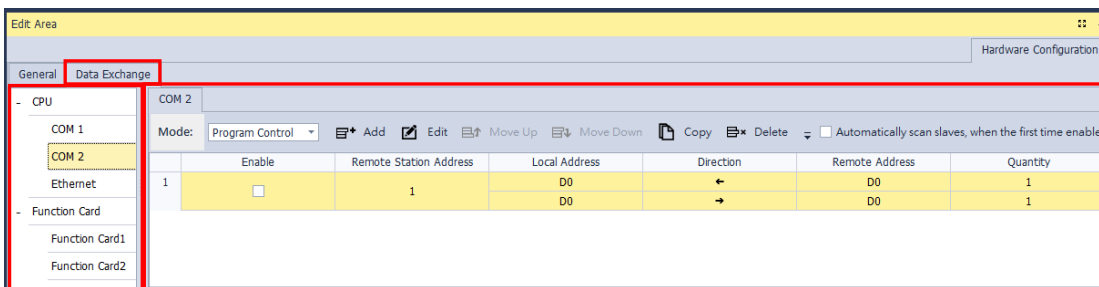
The **Edit Area** is on the bottom of the Hardware Configuration Area by default. You can rearrange the layout; refer to section 3.1.2.4 for more details. You can set the configurations of the PLC CPU, modules, communication port settings, Ethernet settings, function card and many more. Refer to module operation manual for more details on module configurations.

When the image of PLC CPU is selected on the **Hardware Configuration Area**, you can set the following parameters for the PLC CPU, including system settings, communication port settings, Ethernet port settings, built-in AD/DA settings, built-in CAN communication settings on the **General tab**.

3



You can set the data exchange related settings, including communication mode and the device settings on the **Data Exchange tab**.



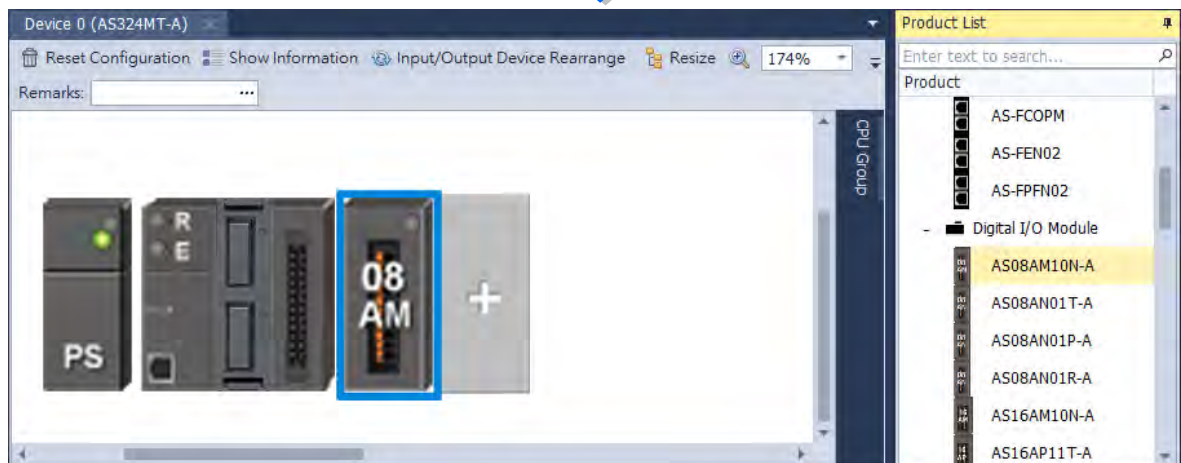
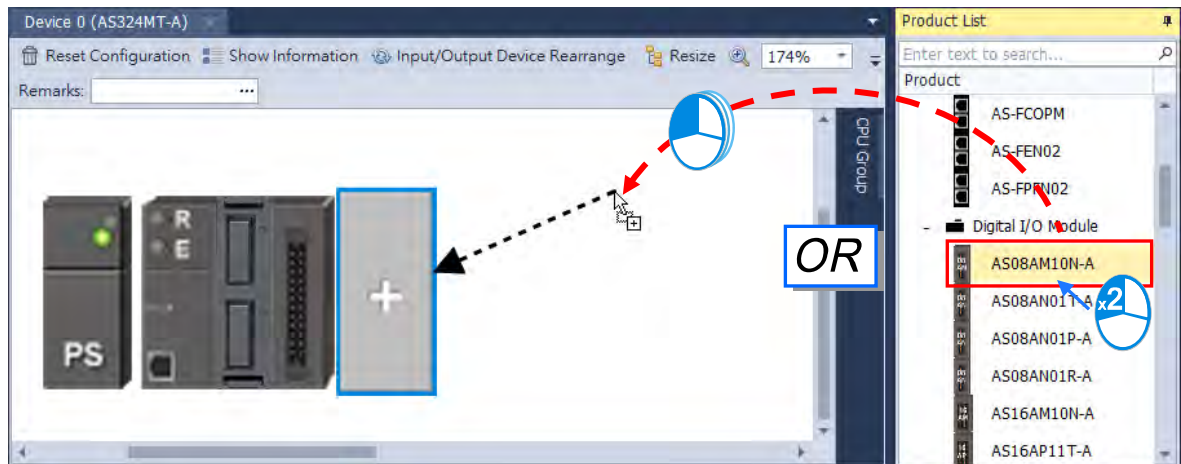
### 3.1.3 Module Configurations

#### 3.1.3.1 Add a Module / Add More Modules At the Same Time

##### Add a Module

- Method 1

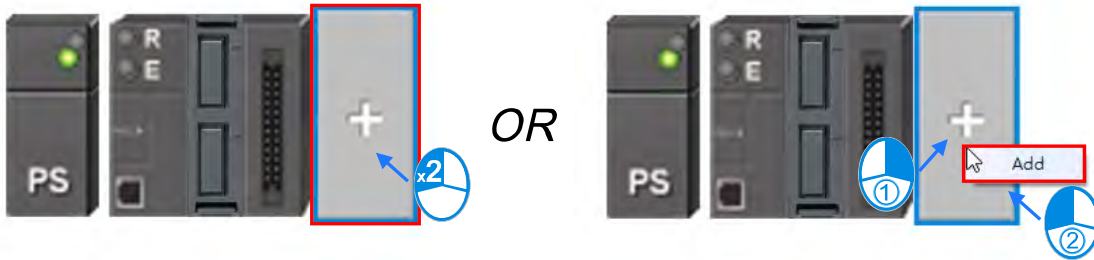
Double-click the module you want to add in the project or select it from the **Product List** and drag that module to the desired position. There will be a suggestive dotted line to indicate the legitimate position for the selected module.





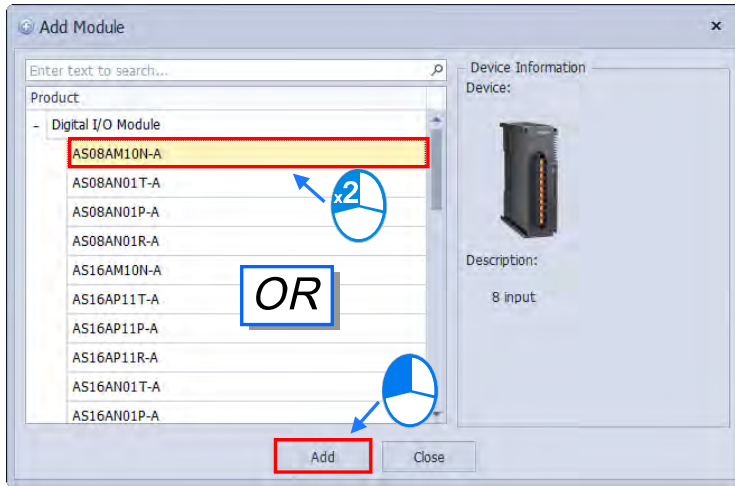
● Method 2

- (1) Double-click a vacant slot or right-click the slot and choose **Add**.



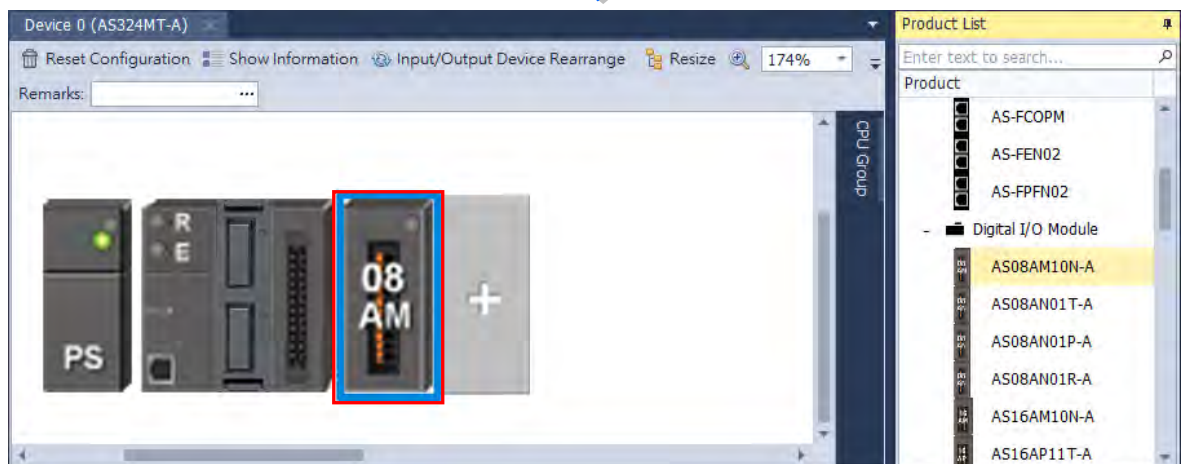
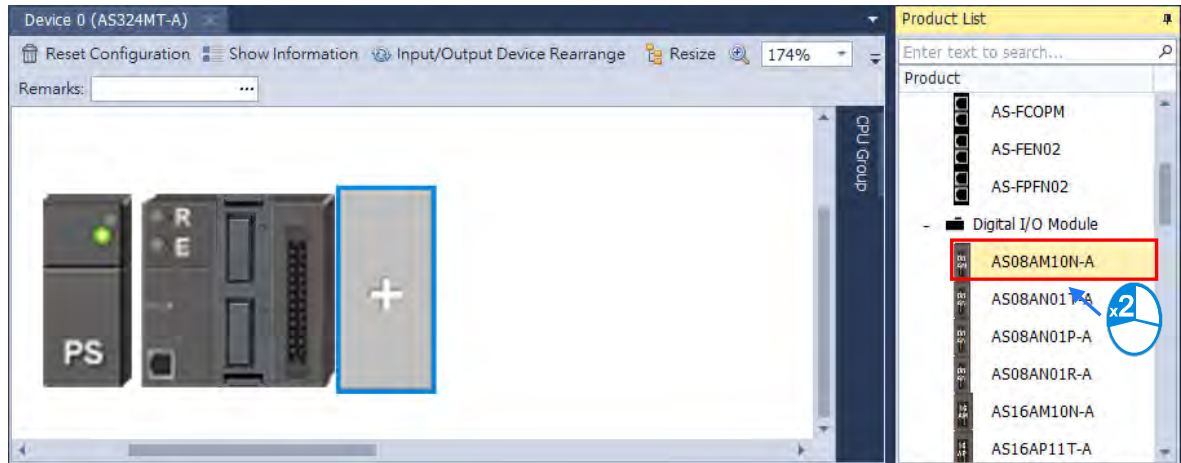
3

- (2) Double-click the selected module to add. You can repeat these two steps to add more modules in.



● **Method 3**

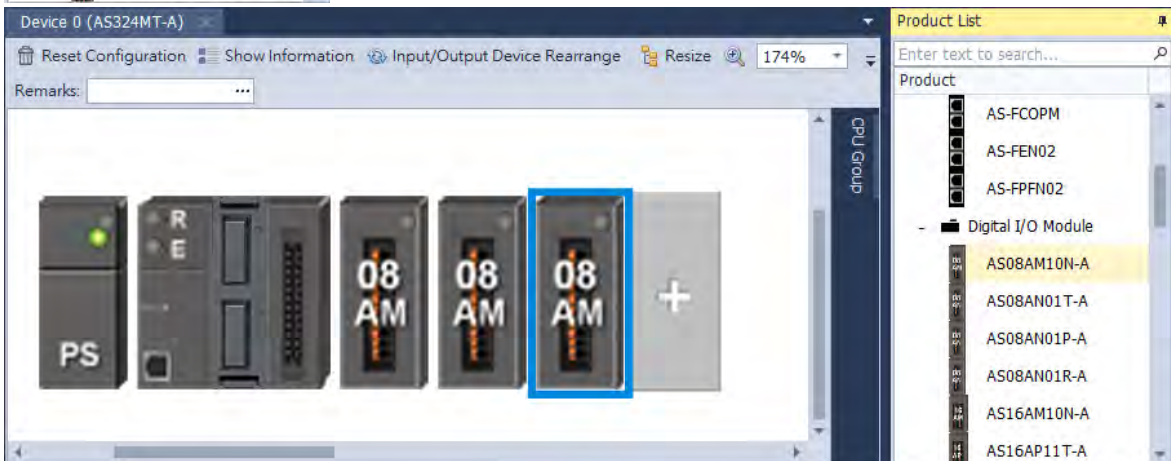
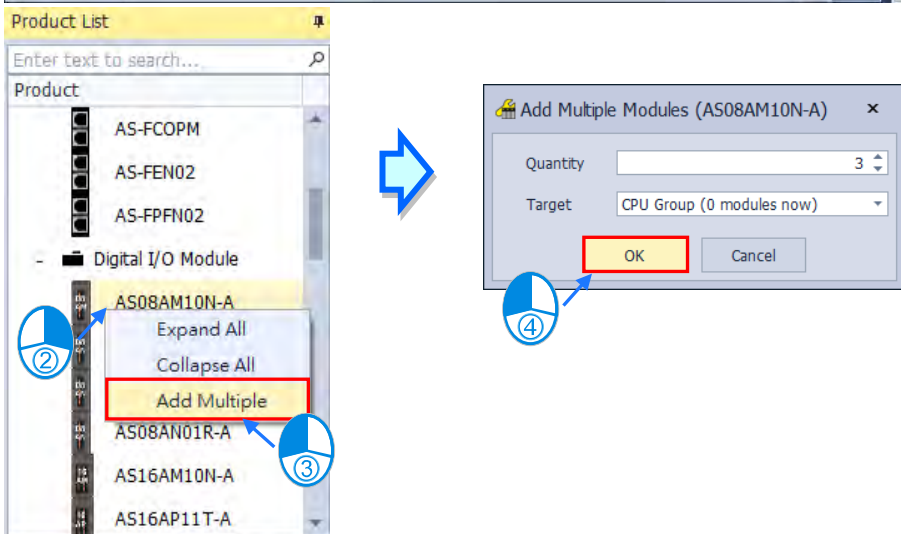
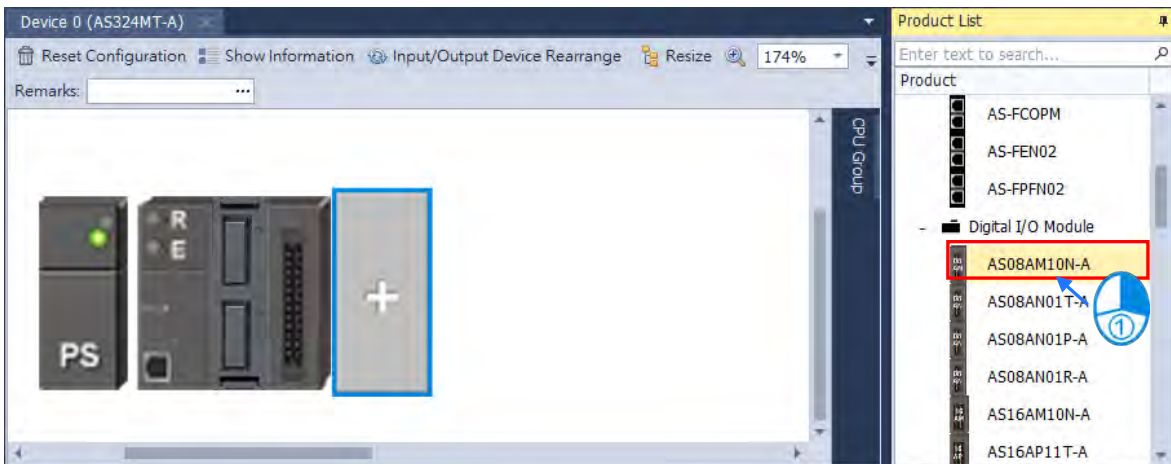
In the **Product List** section, double-click the selected module or select the desired module and press **Enter** on the keyboard to add the selected module in the hardware configuration area.



### Add More Modules at the Same Time

In the **Product List** section, right-click the selected module that you need for more than one, and you will see the context menu showing three options **Expand All**, **Collapse All**, and **Add Multiple**. Click **Add Multiple**, and then an Add Multiple Modules (your desired module name) window appears, use the up and down arrows to increase or decrease the quantity number that you need and then click **OK** to confirm the setting.

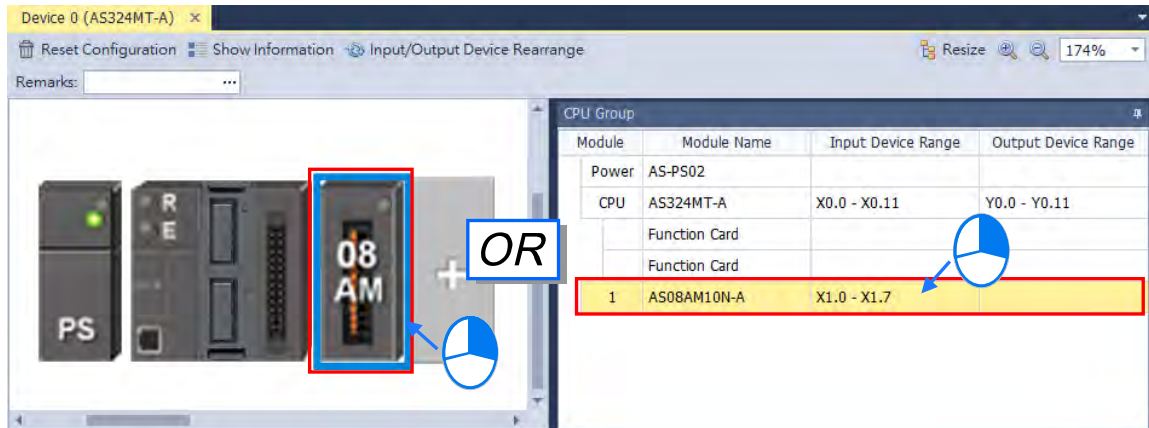
3



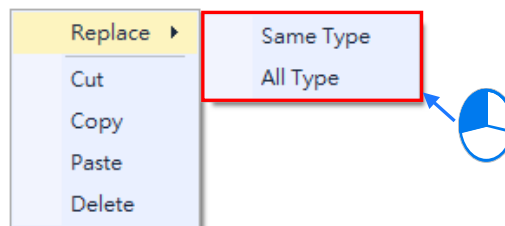
### 3.1.3.2 Replace the Module

The following steps show you how to replace the module.

- (1) Select and right-click the module for replacement in **Hardware Configuration** area or from **CPU Group Information** section.



- (2) Choose **Replace** on the context menu. After that you can see two different types of replacements for selection, **Same Type** and **All Type**.



#### ➤ Same Type

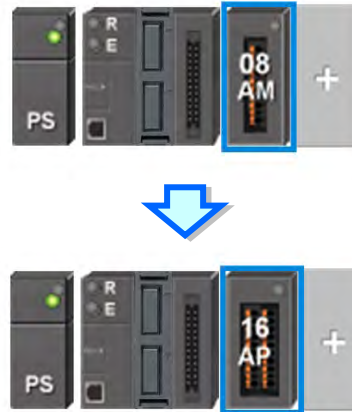
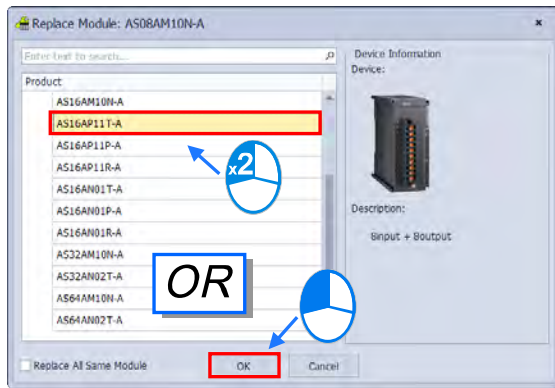
Replace only the selected module with the same type module. The new module **Input/Output Device Range** will be the same, while other parameters may return to system defaults if they cannot be matched.

#### ➤ All

Replace selected module to be any type of module. The result is similar to deleting the original module by adding a new one, so the new module **Input/Output Device Range** will be re-configured and other parameter settings will also return to system defaults.

- (3) Once the replace type is selected, the Module Selection window appears with modules available for the selected replace type. Double click or select the module you want to replace with and click **OK**.

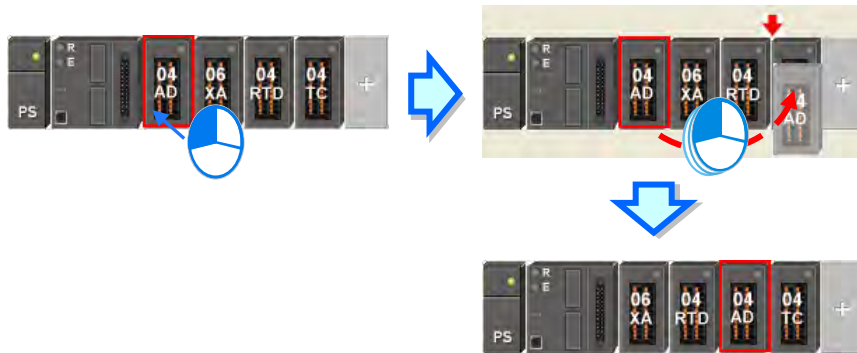
3



### 3.1.3.3 Rearrange Module Position by Drag and Drop

Except CPU modules, you can drag and drop all module graphics in **Hardware Configuration** area to rearrange their positions.

AS Series PLC is non-backplane designed. When the module is dragged to a position between two modules, a red arrow mark appears indicating the position where the module will be after dropping.

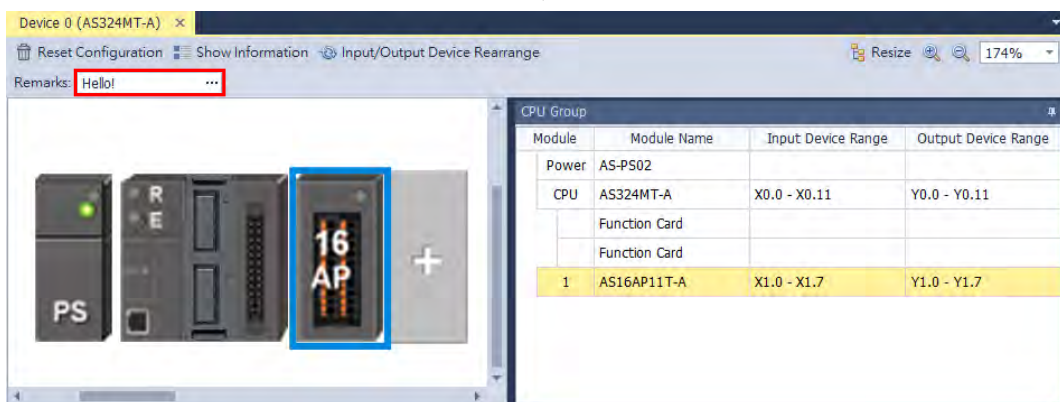
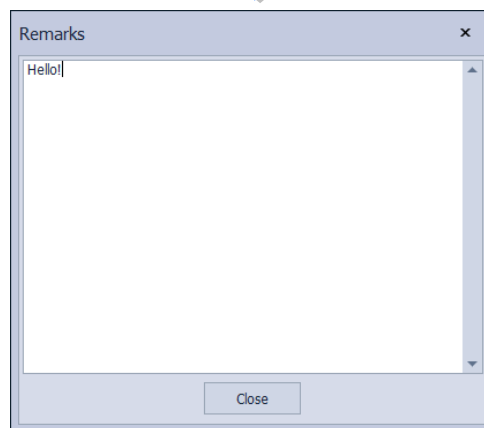
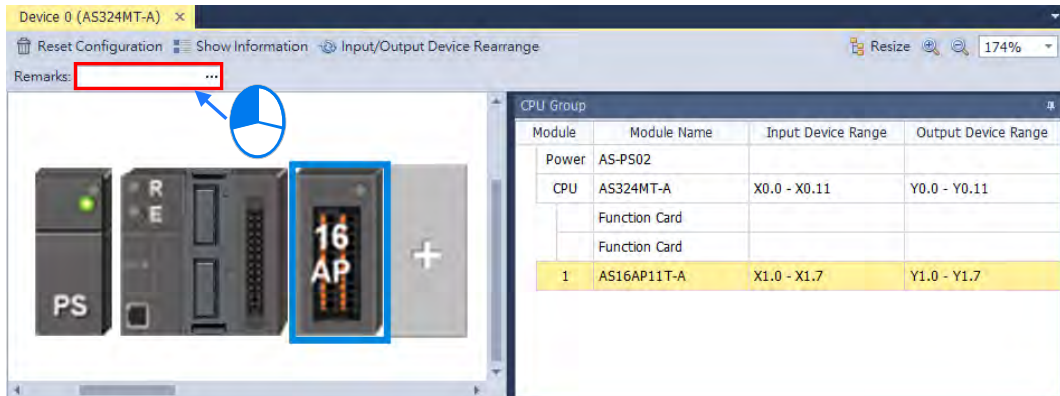


\* When rearranging the module position, the input/output device range, comment, internal parameters, corresponding device D and advanced parameters for intelligent modules are also rearranged with the module.

### 3.1.3.4 Remarks and Comments

- **Remarks for the project**

Click **Remarks** field on the top of the **Hardware Configuration** area and a blank box appears for you to leave remarks for the project. After typing the remarks, press **Enter** on the keyboard or click **Close** to save the remarks.

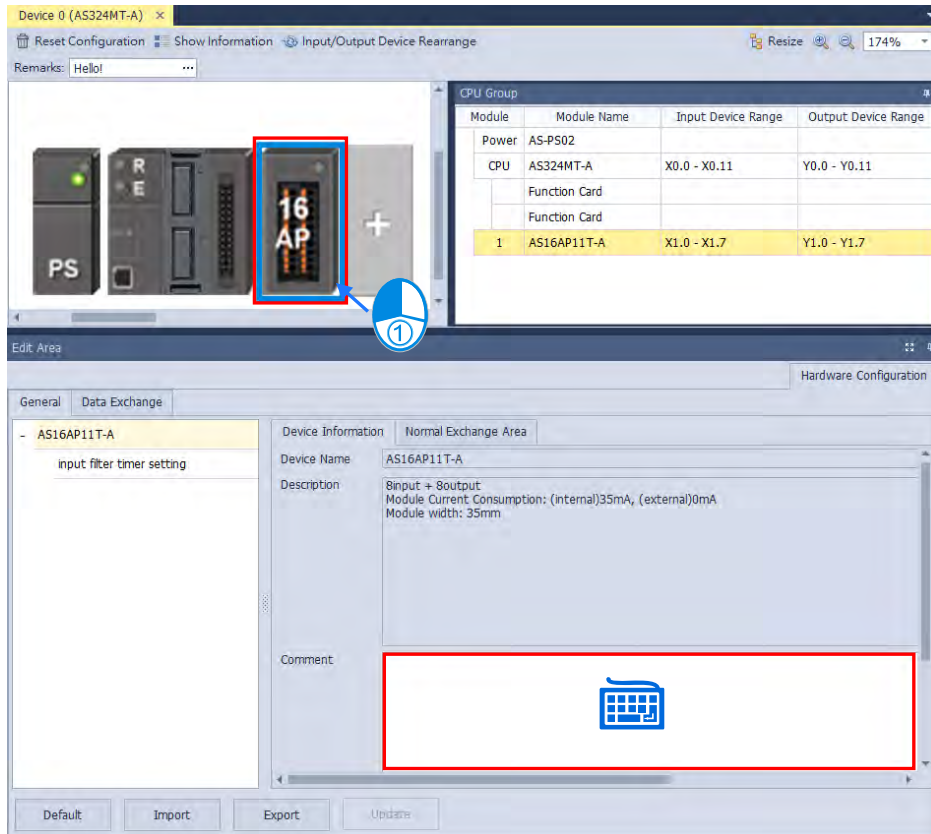


3

- **Comments for PLC CPU and Modules**

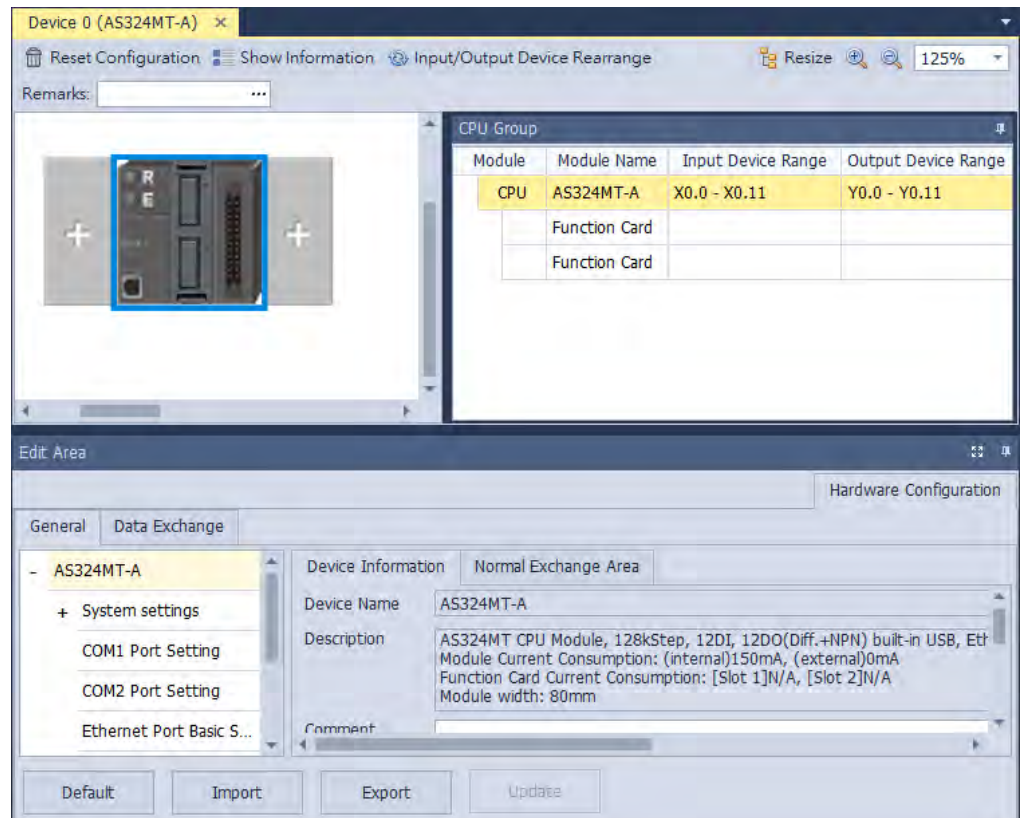
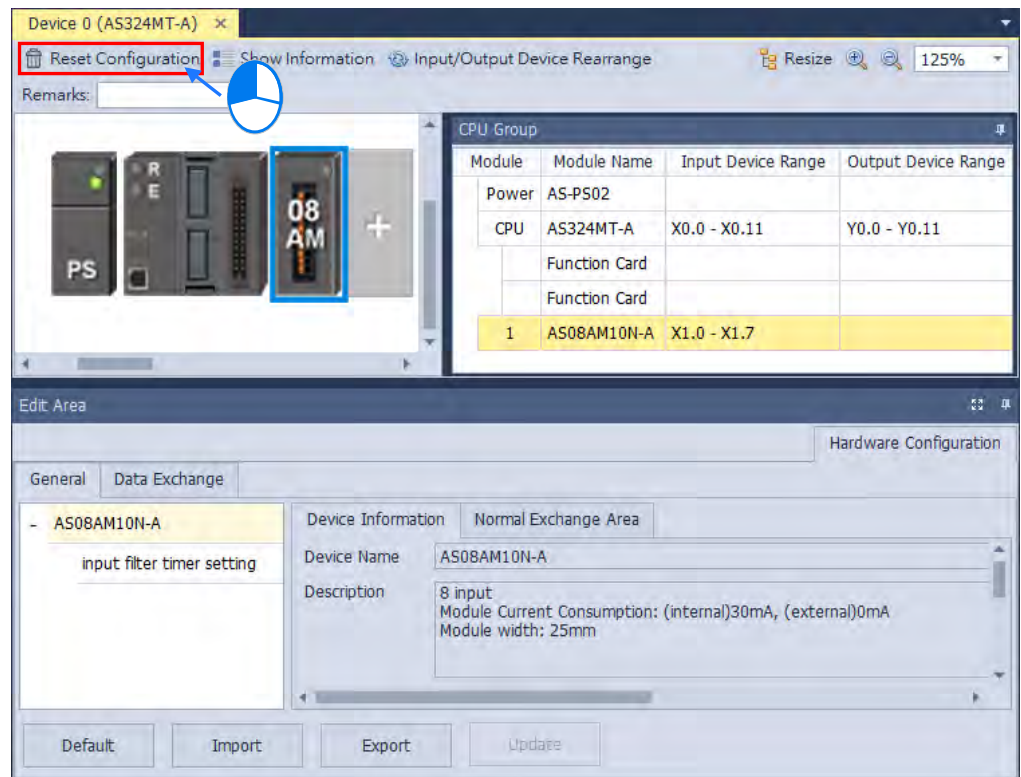
Click the module graphic and you can see its details in the Edit Area. You can leave comments for the PLC CPU and Modules. After typing the comments, it saves the comments automatically.

3



### 3.1.3.5 Hardware Configuration Area - Reset Configuration

Use the functional button **Reset Configuration** to set the PLC configurations back to default values

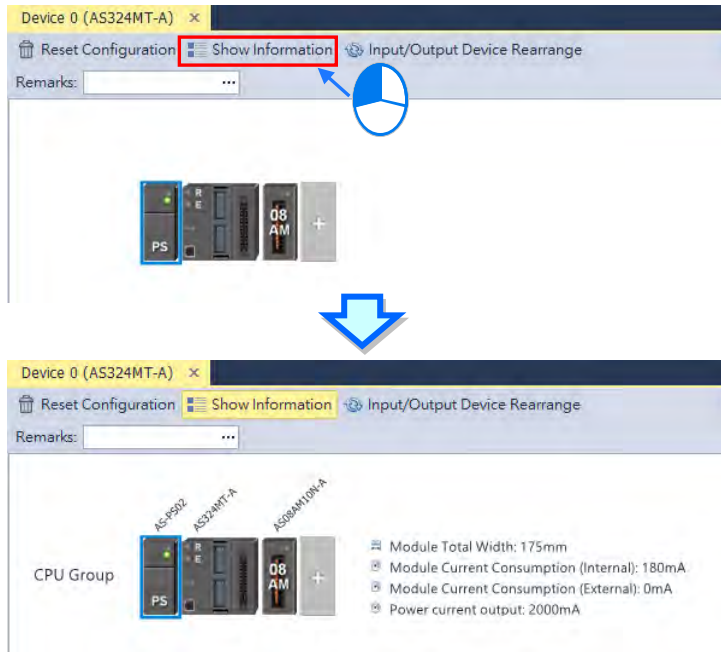


3



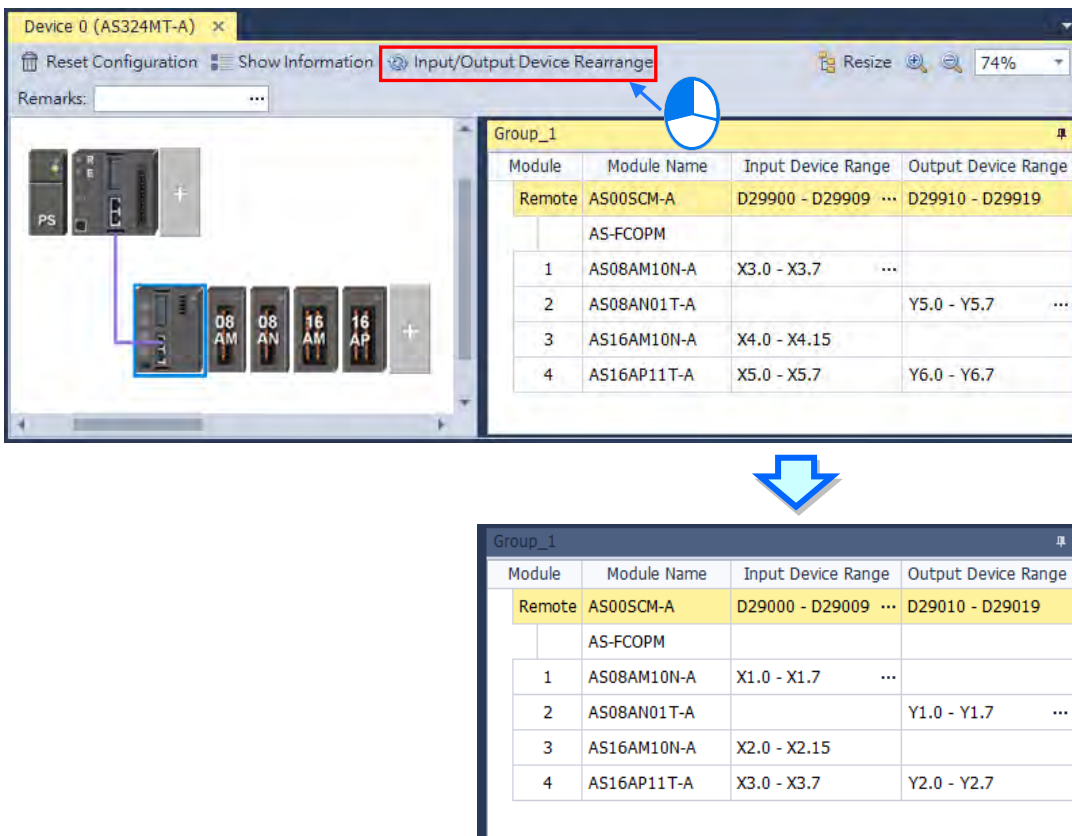
### 3.1.3.6 Hardware Configuration Area - Show Information

Use the functional button **Show Information** to see the hardware configurations



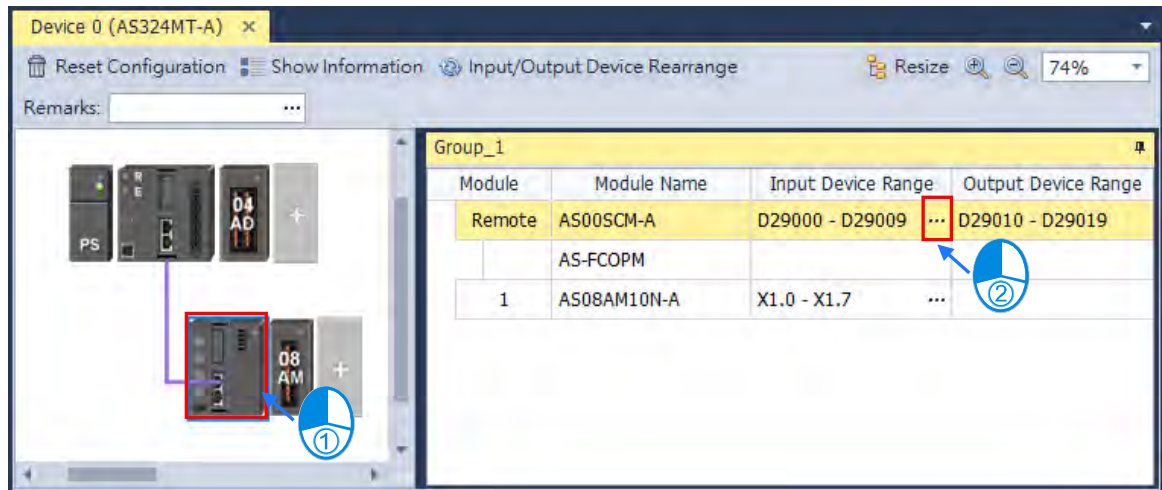
### 3.1.3.7 Hardware Configuration Area - Input / Output Device Rearrange

Use the functional button and **Input / Output Device Rearrange** to rearrange the device ranges.

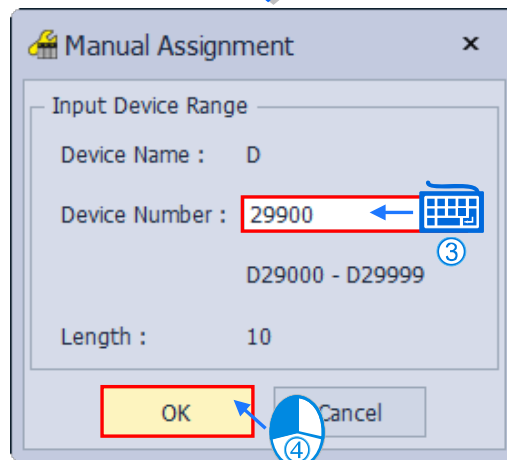


### 3.1.3.8 Hardware Configuration Area - Change Input Device Range of the Remote Modules

Click the remote module and then you can see ... in the column of **Input Device Range**. Double-click it to see the **Manual Assignment** window and start editing your desired input device range.



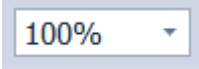



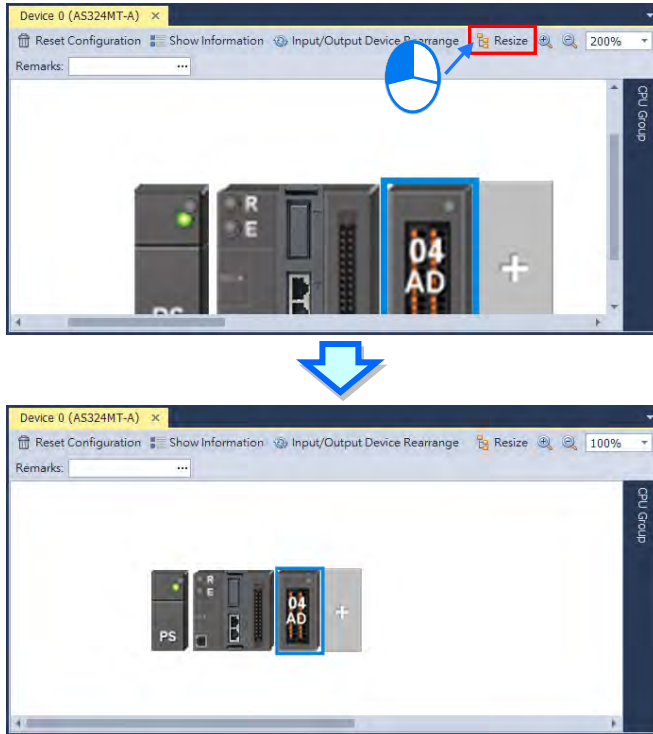
3



Module	Module Name	Input Device Range	Output Device Range
Remote	AS00SCM-A	D29900 - D29909 ...	D29910 - D29919

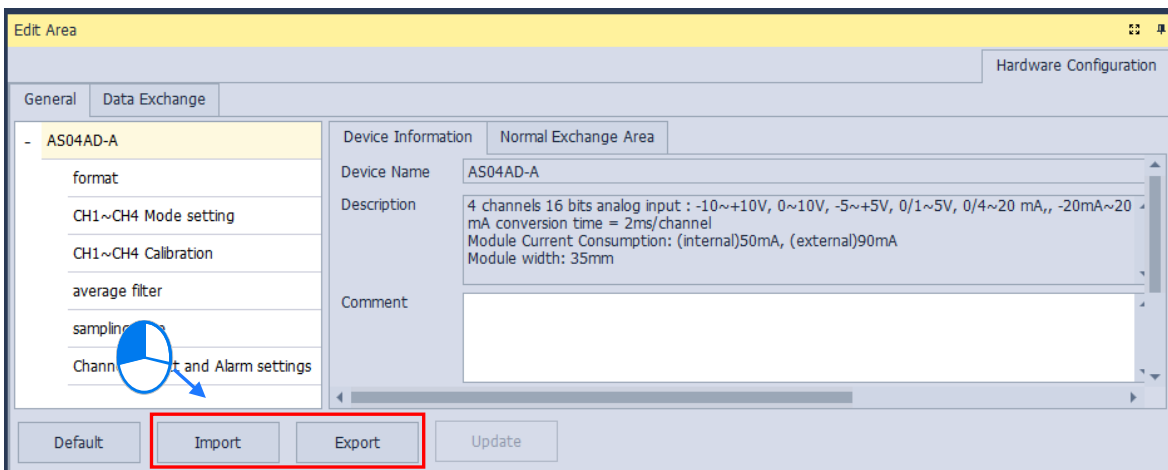
### 3.1.3.9 Hardware Configuration Area - Resize

Use the functional buttons  ,  or  to rearrange the size of the device images in Hardware Configuration Area. Use  to set the display of the configuration area back to its default values (shown at 100% and in the center).



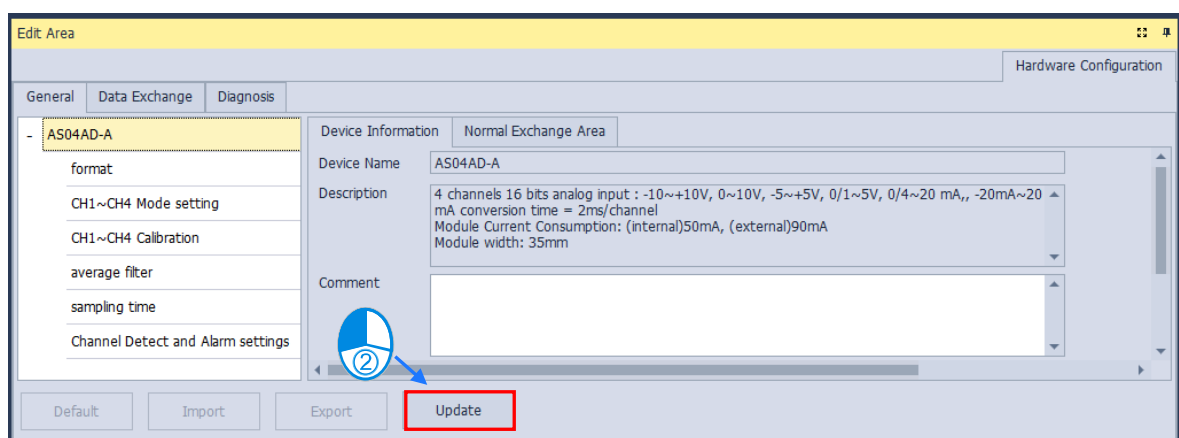
### 3.1.3.10 Edit Area – Import and Export

You can import/export the module parameters in .dep format. Click the **Import** button and then choose a file to import. Click the **Export** button and then choose a path and enter a file name for the exported file. When importing, the system checks if the file format and the module name are matched. If not, an error message shows up.






### 3.1.3.11 Edit Area – Update

To update the module parameters, first you need to be in the online mode. Select the module that you want to update its parameters and then click the **Update** button. This functionality is only available for digital IO modules, analog IO modules and network modules.

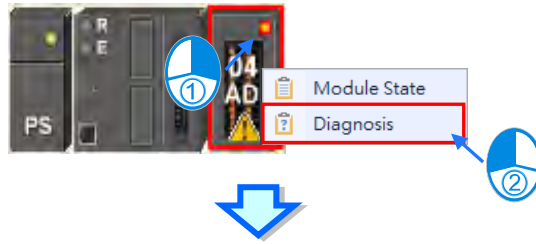


### 3.1.3.12 Show or Hide the Display

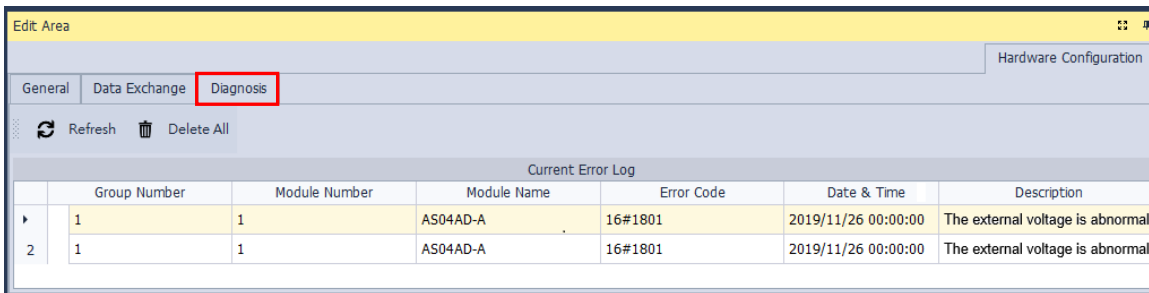
Click  or  to hide the display area and after that only its tab remains shown. Move your cursor to the tab to have the hidden display area shown. Click  to pin and lock the display area to keep it shown.



### 3.1.3.13 Module State and Diagnosis

To check the module state and diagnosis, first you need to be in the online mode. Right-click the module that you want to check its module state and diagnosis and then a context menu appears.




3



After you click Diagnosis, you can see a table with three tabs. On the Diagnosis tab, you can see the Current Error Log table. When the error is cleared, you can use  to clear the error log stored in the module and the module state can be restored to normal. Use  to update the module state.

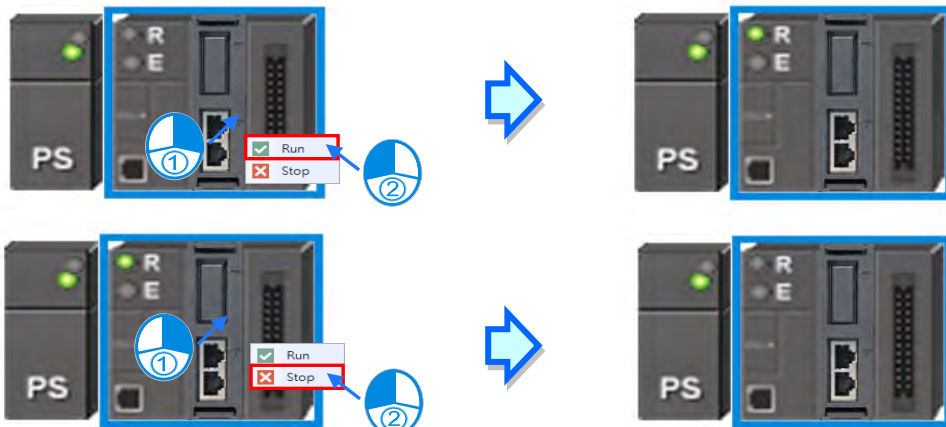
### 3.1.3.14 Change Module State in Online Mode

You can change the module state in online mode.

 Before changing the module state, make sure no personnel or system will be affected.

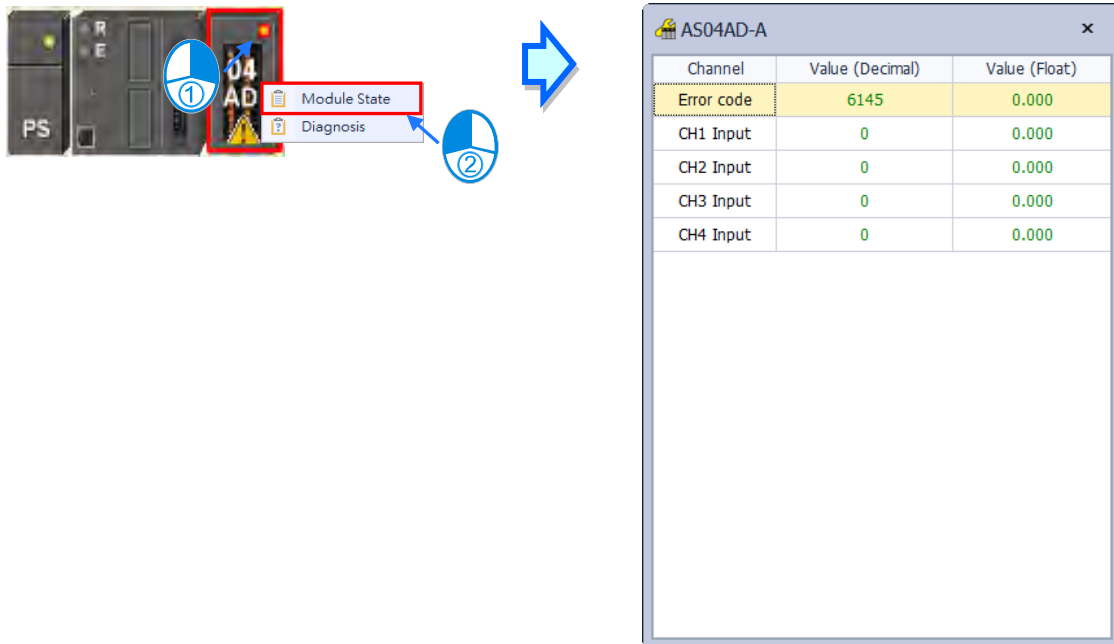
#### Change the module operation state

- Right-click the PLC CPU to see the context menu and click **Run** for the PLC CPU and module to start running (RUN LED ON) or click **Stop** for the PLC CPU and module to stop running.



**Change the module I/O state**

- This functionality is only available for digital IO modules, analog IO modules and temperature modules. Right-click the module to see the context menu and click **Module State** and then you can see an IO state table shows up. For digital IO modules, you can right-click to set the input/output channel to ON or OFF, when the PLC CPU and the module are on the RUN state.

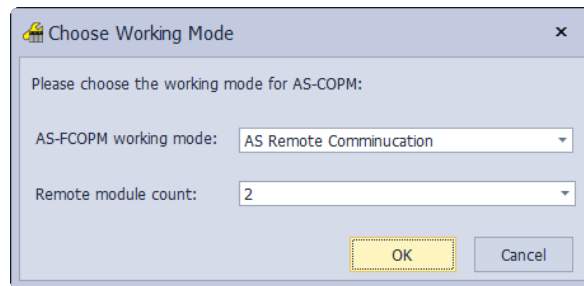
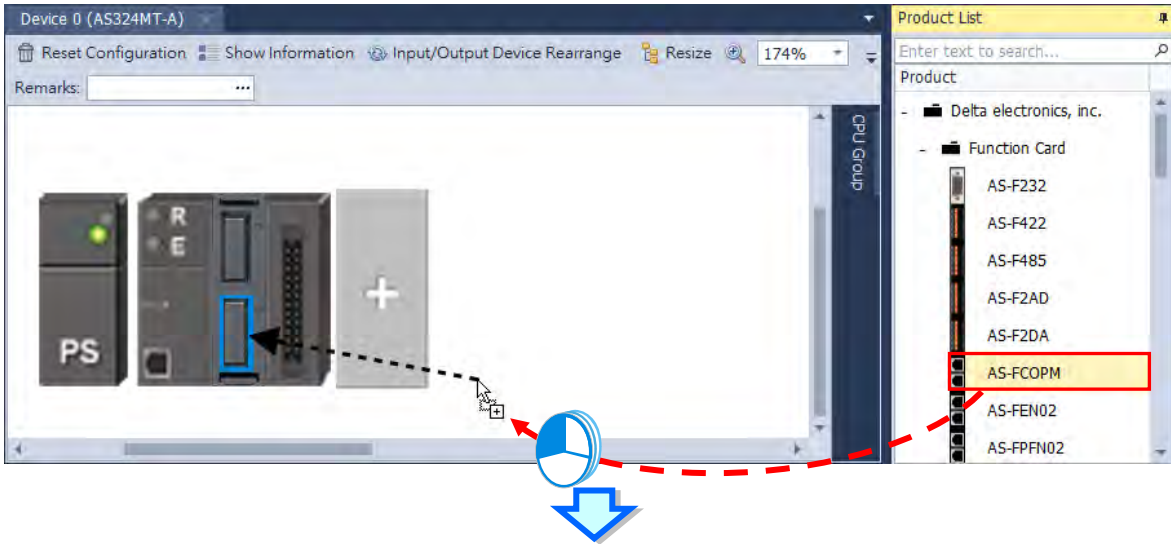


- **Force to ON** : Force to set the channel state to ON
- **Force to OFF** : Force to set the channel state to OFF
- **Release** : Release the selected channel from the force
- **Release all** : Release all channels from the force

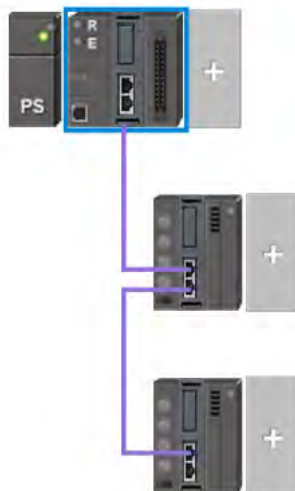
### 3.1.3.15 Configure a Remote Module

Drag AS-FCOPM from the Product List to the Function Card Slot 2 of the PLC CPU. After that you can set the Working Mode for the AS-FCOPM as AS Remote Communication and then you can set the quantity of the Remote Modules that you want to connect.

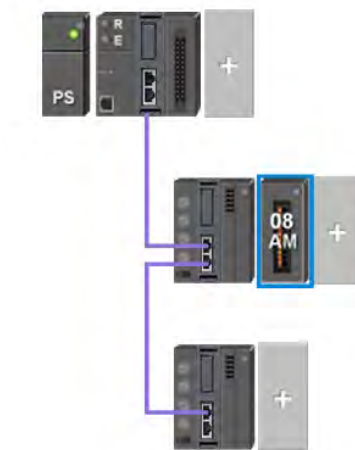
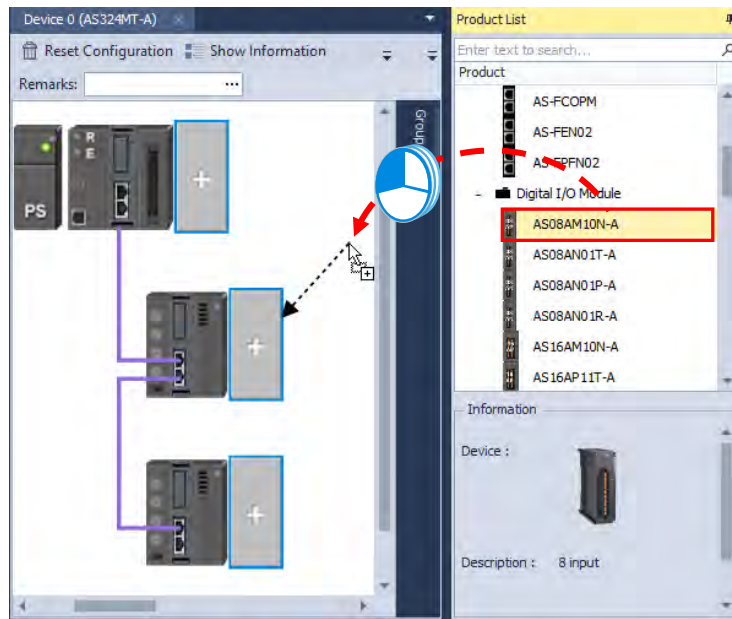
3



If you set the Remote module count to 2 and click OK, you can see two remote modules connected to the PLC CPU.



You can add more digital IO modules or analog modules to the remote module.

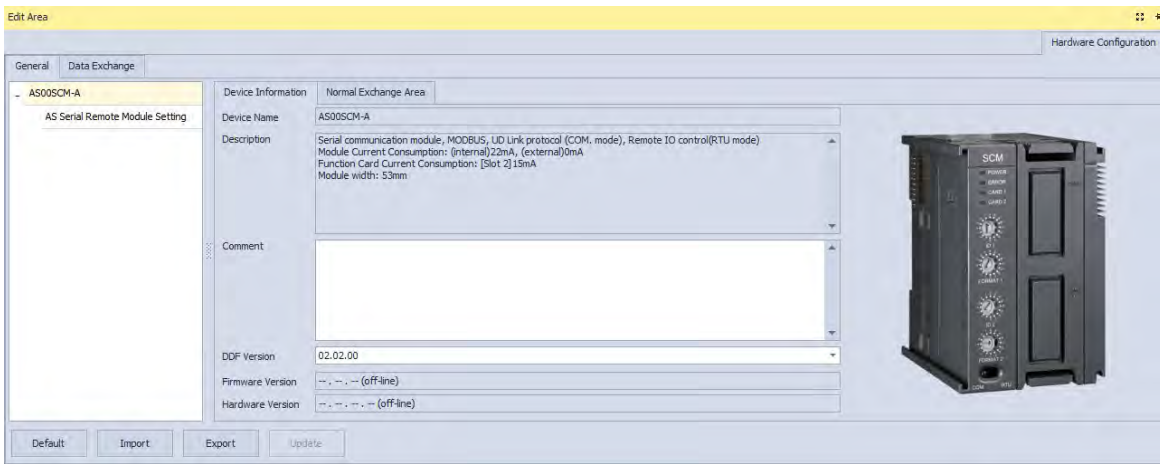
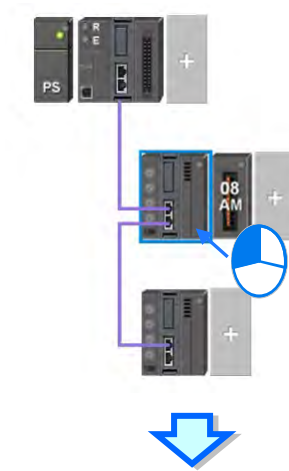


3



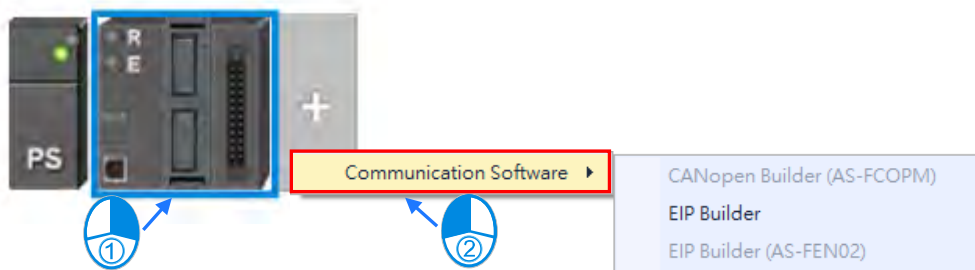
And then you can click the module to open its configuration page to configure.

3



### 3.1.3.16 Open Communication Software from HWCONFIG

Right-click the PLC CPU and click **Communication Software** to see which software is available for this PLC CPU. If the software option is grayed out, you may need applicable function cards to work along with the project.

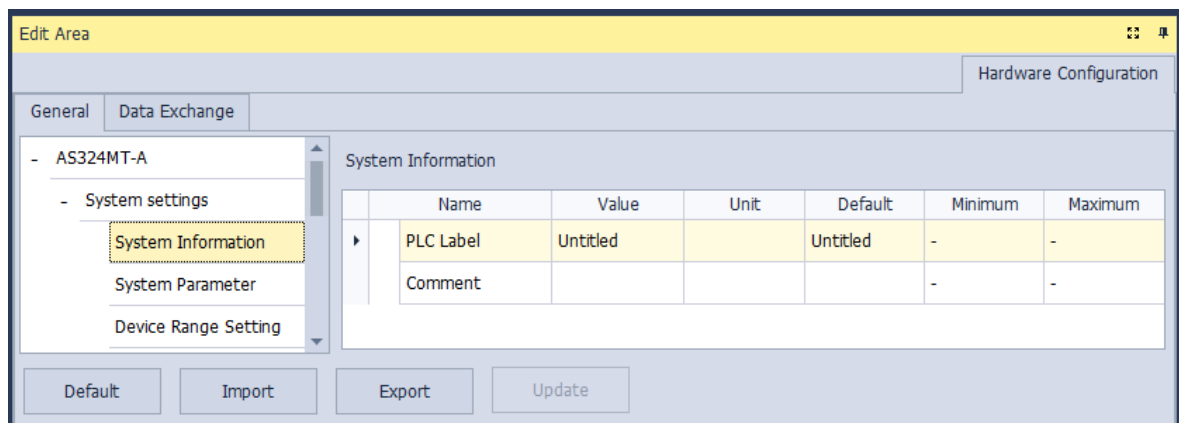


### 3.1.4 Parameter Settings for PLC CPU

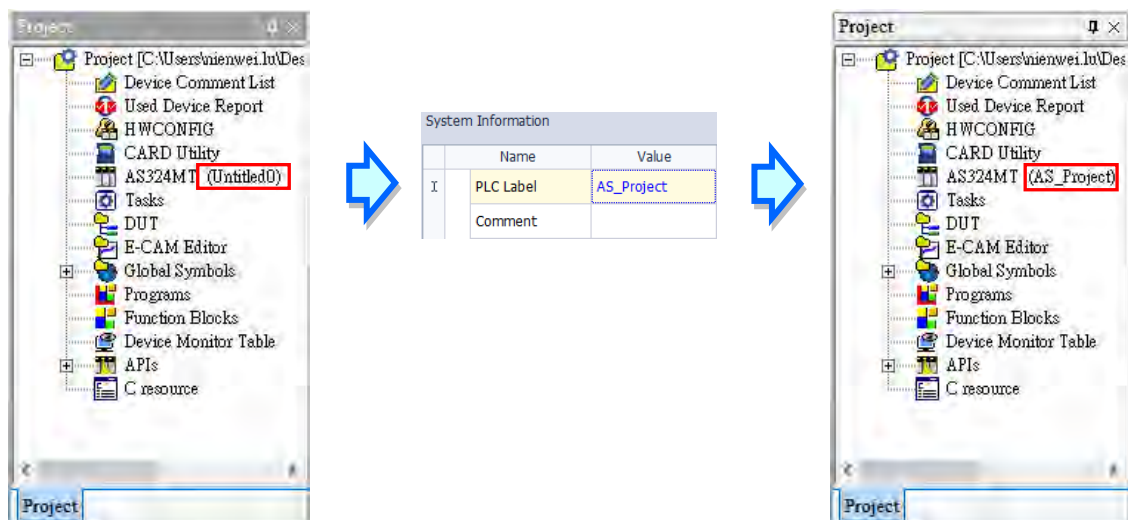
The functions and parameters of a module vary with the different type of PLC CPU. This section gives you a roughly instruction on PCL CPU parameter settings; for more details, refer to the individual operation manual of each PLC CPU.

#### 3.1.4.1 System Settings – System Information

In the Edit Area, select the option **System Settings** and you can find two items, **PLC Label** and **Comment** under this option. You can input up to 16 characters in the value field of **PLC Label** and 32 characters in the value field of **Comment**. You can use all characters in the fields, including special characters and spaces. Note: One Chinese character occupies two character.



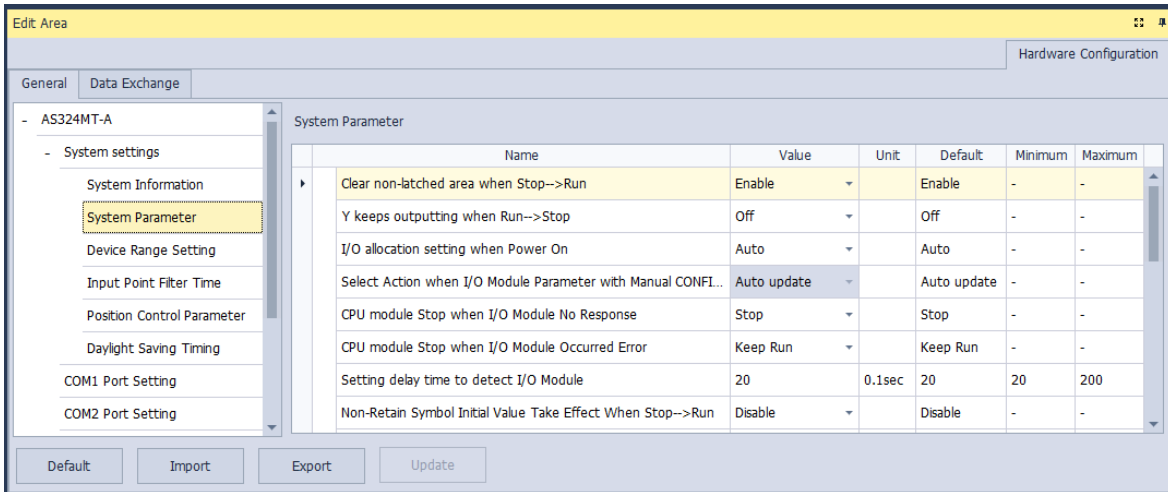
After the project is created, you can find the PLC label behind the product name in the project management area. You can change the PLC label in HWCONFIG as it is mentioned above. The PLC label is very useful when you have more than one PLC in the project. This label can be seen as the PLC identity. So that you will not change other PLC parameters by mistake. To prevent errors, when you download/upload the program, the system will remind you to check the name of the PLC CPU and the PLC label.



### 3.1.4.2 System Settings – System Parameter

The parameters on **System Parameter** table are shown in the following window. You can set appropriate values via a drop-down list or type the values in the box.

3



- **Clear non-latched area when STOP --> RUN**

If the state of the CPU module changes from **Stop** to **Run**, the states of the non-latched devices and the values in the non-latched devices are cleared.

- Enable: The current states and values of devices are retained.
- Disable: The states and values of devices are cleared to default value.

- **Y Keeps outputting when RUN --> STOP**

Set the state of Y device when the state of CPU is switched between RUN and STOP.

- Off : All Y devices are set to OFF.
- Retain Present State: The states of the Y devices are retained.

- **I/O allocation setting when Power On**

The actual I/O module configuration is read automatically when the CPU module is powered ON.

Or the CPU module operates based on the configuration set manually in HWCONFIG.

- Auto: Operates based on actual connected modules without any module and parameters set.
- Manual: Operates based on the configuration set in HWCONFIG.

➤ **Manual + Flag:** Select this option, you need to use it with flags SM230-SM261. For less module applications, you can simply use special flags to mark which I/O module will NOT be used to meet the actual I/O module placement. In this mode, you can use the same PLC program and download the I/O allocation table once for various I/O applications.

- **Select Action when I/O Module parameter with Manual CONFIG Different**

The parameter cannot be set unless the **Manual** option is selected in the above parameter **I/O module CONFIG by Auto/Manual when Power On**.

➤ **Auto update:** The CPU module will update its configurations based on the actual module configurations.

➤ **CPU alarm:** The CPU module shows a warning when an error occurs.

- **CPU module Stop when I/O Module No Response**

The parameter sets whether the CPU and other normal modules can operate constantly when there is a expansion module, which does not response when offline.

➤ **Stop:** The CPU module stops running and gives errors.

➤ **Keep Run:** The CPU module and other normal modules keep running.

- **CPU module Stop when I/O Module Occurred Error**

The parameter sets the method to deal with a tiny error in the expansion modules.

➤ **Stop:** The CPU stops running and sends an error.

➤ **Keep Run:** The CPU keeps running but records the warning message.

- **Setting delay time to detect I/O Module**

The parameter sets how long after the CPU is powered on, the I/O module and remote I/O module begin to be scanned.

- **Assign X Input Point Control Run/Stop**

The parameter sets whether one X input point controls the RUN or STOP state of the CPU.

- Disable: The function is closed. The state of the CPU can be changed with its DIP switch.
- Enable: Use the specified X input point to control and the DIP switch of the CPU can also be used.

### 3

- **Select X Input Point**

Select one X input point on the drop-down list to switch Run/Stop if the above parameter **Assign X Input Point Control Run/Stop** is enabled.

- **Constant Scan Cycle Time**

The parameter sets the shortest constant time to complete a cycle of scan.

- Disable: The function is closed.
- Enable: The CPU starts the next scan after the waiting time reaches the setting value if the actual scan time is less than the setting value. However, the CPU will operate based on actual scan time disregarding the setting value if the actual scan time is greater than the setting value.

- **Input Constant Time**

Type the cycle time value in the parameter box if the parameter **Constant Scan Cycle Time** is enabled. Please note if the setting value of scan time is greater than the setting value of **Setting Watchdog Time**, there will be a timeout error once the CPU operates.

- **Setting Watchdog Time**

The parameter sets a timeout during which the program is scanned. The CPU module will send an error if the program execution exceeds the time.

- **Show Battery Low Voltage Error**

The parameter sets whether the alarm is shown when the lithium battery for the real-time clock is of low voltage or is not installed in AS series.

- Disable: The function is closed.
- Enable: There is an alarm when the lithium battery is of low voltage or not installed.

- **Save Error history into**

The parameter sets the location to store error logs.

- PLC: The error logs are stored in PLC. Twenty error logs at most can be stored. If there are more than twenty error logs, the oldest error log will be overwritten by the latest error log.
- PLC & SD Card: The error logs are stored in PLC and SD card. There are more than twenty error logs, the oldest error log will be moved to the SD card before the oldest error log is overwritten.。

- **COM Communication Error Record**

The parameter sets whether to enable the **error record** when there is an error at COM port.

- Disable: The function is closed.
- Enable: The error record function is enabled.

- **Select Action When 24Vdc Input Unstable**

What to do when the 24Vdc power is unstable

- Continue Running when power stable: waits till the power is stable and then PLC begins to run.
- Into Error Status: Stops and go to the error mode °

- **Communication Module Refreshed Priority**

- Scan time first: only refresh the data that is different from the original ones to save scan time
- Synchronous Data first: scan and refresh all the data to ensure all data are synchronized.

- **AS remote module updated method**

- PLC Scan cycle first: update according to PLC program scan cycle
- Update one remote module by one scan time: update one remote module at one scan time
- Update all remote modules at one scan time: update all remote modules at one scan time

- **1601 Timer interrupt Setting Time Base**

3 Users can set the interval of triggering 1601 timer interrupt. The function is used with Timer Interrupt 0 in the work management together. Please refer to chapter 5 for more information.

- **1602 Timer interrupt Setting Time Base**

Users can set the interval of triggering 1602 timer interrupt. The function is used with Timer Interrupt 1 in the work management together. Please refer to chapter 5 for more information.

- **1603 Timer interrupt Setting Time Base**

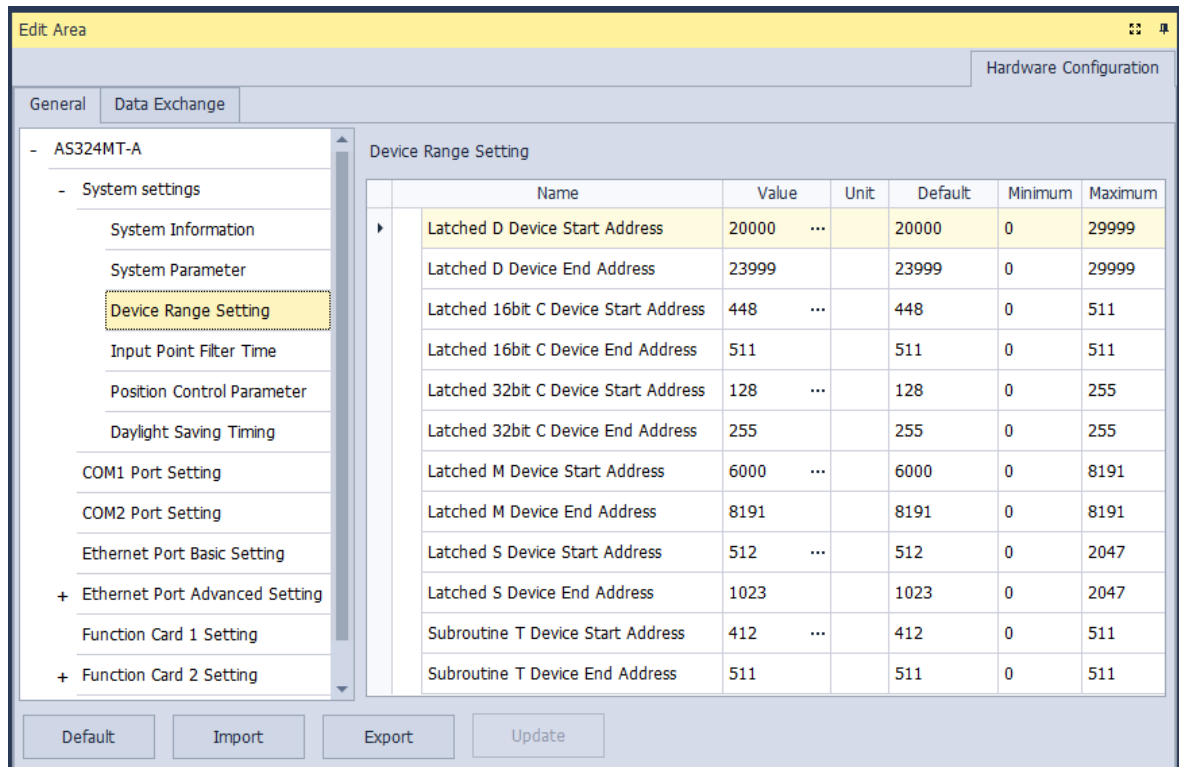
Users can set the interval of triggering 1603 timer interrupt. The function is used with Timer Interrupt 2 in the work management together. Please refer to chapter 5 for more information.

- **1604 Timer interrupt Setting Time Base**

Users can set the interval of triggering 1604 timer interrupt. The function is used with Timer Interrupt 3 in the work management together. Please refer to chapter 5 for more information.

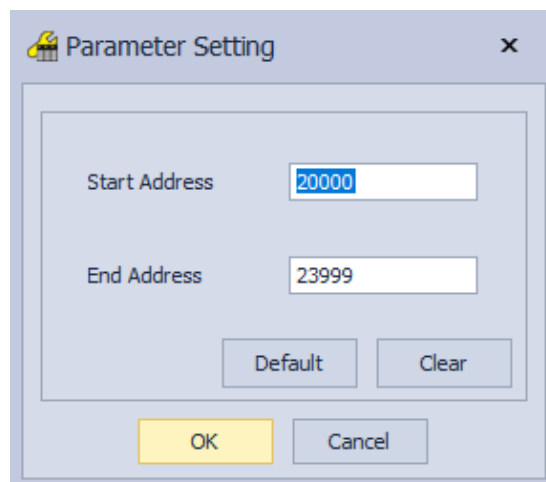
### 3.1.4.3 System Settings – Device Range Setting

The parameters on **Device Range Setting** table are shown in the following window.



3

Set the start address and end address of this kind of devices by clicking **...** button. Click **Default** button to restore to the default value. Click **Clear** button to clear the box content to 0. Finally, click **OK** to save the settings or **Cancel** to give up the setting and leave the dialogue box.

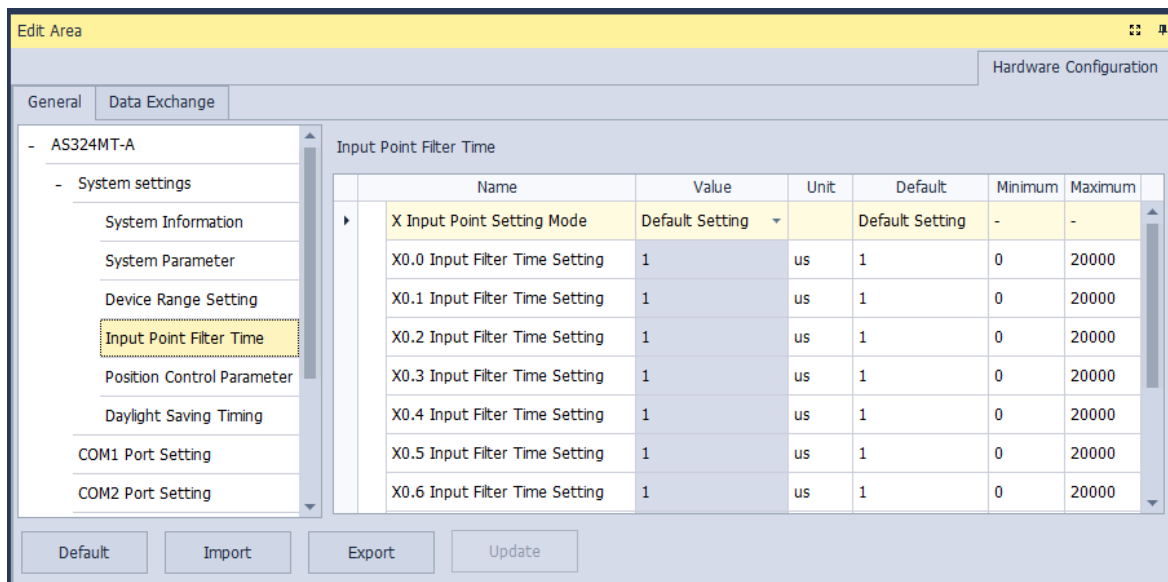




### 3.1.4.4 System Settings – Input Point Filter Time

The filter time of every input point is set on the **Input Point Filter Time** table. The signal will be filtered as a noise if the period of time during which an input point receives a signal less than the filter time. And users could set an appropriate filter time according to the interference state of the field environment.

3



- **X Input Point Setting Mode**

The parameter sets whether to set the filter time of input points or to use default settings.

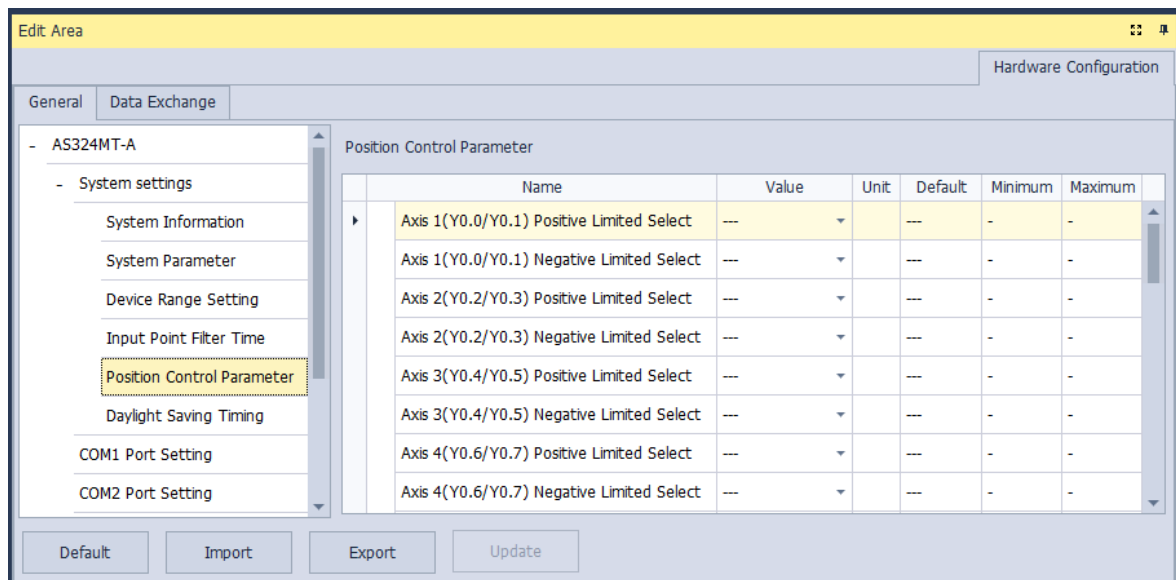
- Default Setting: Filter time is set to default value.
- Custom Setting: Users set the filter time of every X input point themselves.

- **X0.0~X0.15 Input Filter Time Setting**

Users could type the respective filter time of X0.0~X0.15 if Custom Setting is selected as **X Input Point Setting Mode**.

### 3.1.4.5 System Settings – Position Control Parameter

The parameters on **Position Control Parameter** table set to specify input points as the positive and negative limits of axis 1~ axis 6 channels. 12 limit points can be set at most. Axis No. 1~6 can also be set as Z phase triggers, home function finish points and clear output selects or positive/negative limited position.



- **Axis1 (Y0.0/Y0.1) Positive/negative Limited Select ~ Axis 6 (Y0.10/Y0.11) Positive/negative Limited Select.**

Select the rising or falling edge trigger and X input point on the drop-down list.

- **Axis1 (Y0.0/Y0.1) Z Phase Trigger ~ Axis6 (Y0.10/Y0.11) Z Phase Trigger**

Select the rising or falling edge trigger and X input point on the drop-down list.

- **Axis1 (Y0.0/Y0.1) Home Function Finish and Clear Output Select ~ Axis6 (Y0.10/Y0.11) Home Function Finish and Clear Output Select**

Select the rising or falling edge trigger and X input point on the drop-down list.

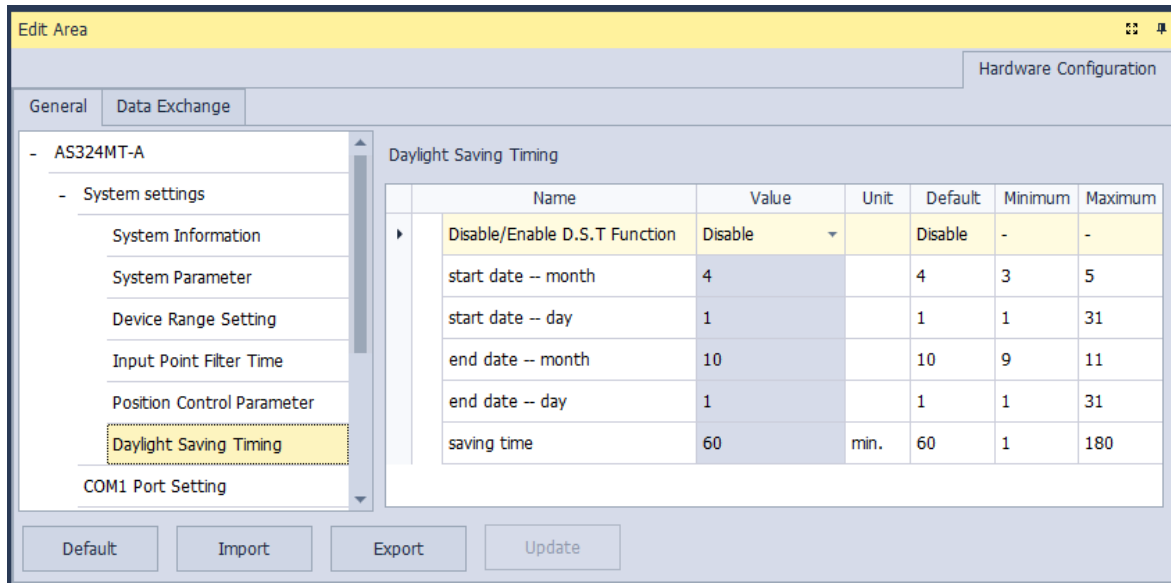
- **Axis1 (Y0.0/Y0.1) Positive/negative Limited Position ~ Axis6 (Y0.10/Y0.11) Positive/negative Limited Position**

Set up the number of pulses as the positive or negative limited position in axis 1~6; setting range is -2147483647 ~ 2147483647.

3

### 3.1.4.6 System Settings – Daylight Saving Timing

The parameters on **Daylight Saving Timing** table are used to enable or disable the function of daylight saving time and set the date when the daylight saving is conducted. The clock will automatically set the daylight saving time in advance based on the period of time set after the function is enabled.



The screenshot shows the 'Edit Area' window with the 'Daylight Saving Timing' configuration table. The table has the following data:

Name	Value	Unit	Default	Minimum	Maximum
Disable/Enable D.S.T Function	Disable		Disable	-	-
start date -- month	4		4	3	5
start date -- day	1		1	1	31
end date -- month	10		10	9	11
end date -- day	1		1	1	31
saving time	60	min.	60	1	180

- **Disable/Enable D.S.T Function**

The parameter sets to enable or disable daylight saving time.

- **Start date- month**

The parameter sets the month from which the daylight saving time starts.

- **Start date- day**

The parameter sets the date on which the daylight saving time starts.

- **End date- month**

The parameter sets the month in which the daylight saving time ends.

- **End date- day**

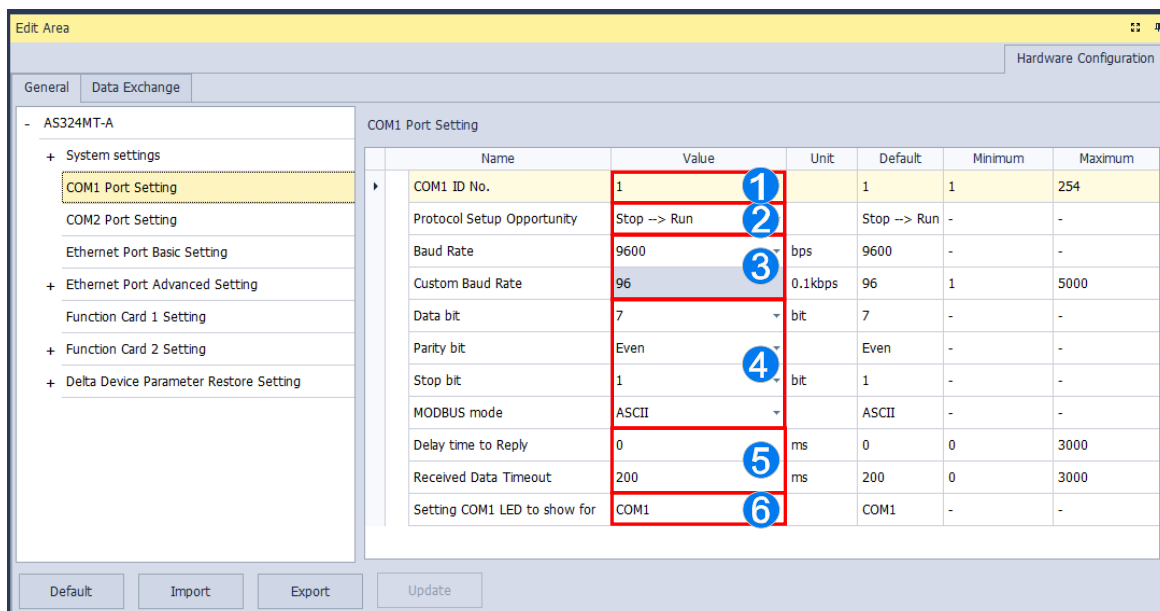
The parameter sets the date on which the daylight saving time ends.

- **Saving time**

The parameter is used to adjust to the earlier daylight saving time with the unit of minute.

### 3.1.4.7 Options - COM1 Port Setting & COM2 Port Setting

PLC modules are equipped with one or two COM ports based on different models. COM1 and COM2 parameters are set by clicking **COM1 Port Setting** and **COM2 Port Setting** in the list on the left-side of the following window. And the specific setting items are same for the two COM ports.



- ❶ 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.
- ❷ Users can set the timing of getting effective for the communication port. The communication port will take effect when the CPU module changes from STOP to RUN if users select **Stop --> Run**. The communication port will take effect when the CPU module is powered on if users select **Power on**.
- ❸ Users set the baud rate for the COM port. Users can select one of the preset baud rates on the drop-down menu or select **custom** to type a custom baud rate.
- ❹ Users set the communication protocol for the COM port.
- ❺ Users set a value for **Delay time to Reply**. The AS CPU module waits for a period of time set and then gives a reply after receiving a communication command so as to match the time when the remote device switches the mode. Users set a value for **Received Data Timeout**. The AS CPU module sends a communication command as a master. But it cannot receive a reply within a period of time set and that will be regarded as a timeout.
- ❻ Setting COM1 to show for: when COM1 is selected from the drop-down list, it will show the RS485 communication status of the COM1; when Card1 is selected from the drop-down list, it

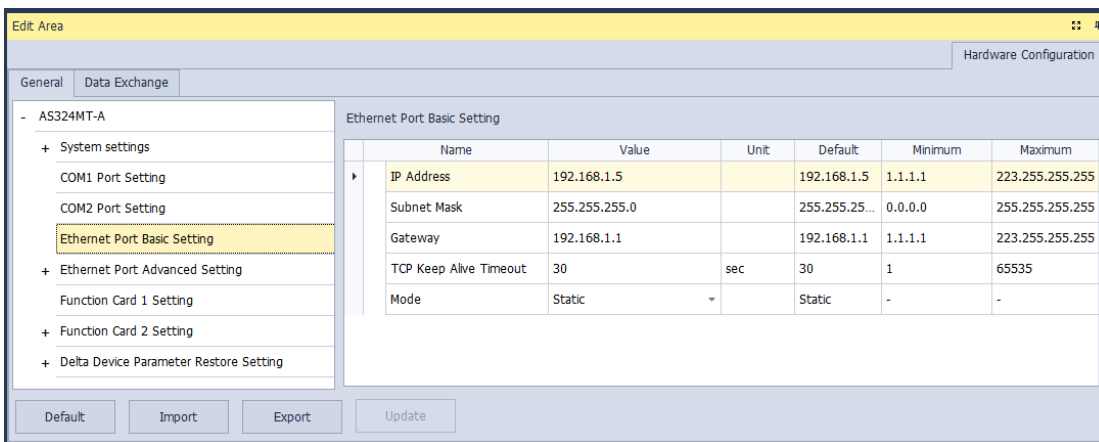
will show the Card1 communication status.

Setting COM2 to show for: when COM2 is selected from the drop-down list, it will show the RS485 communication status of the COM2; when Card2 is selected from the drop-down list, it will show the Card2 communication status.

### 3.1.4.8 Options - Ethernet Port Basic Setting

The communication parameters can be set for the Ethernet port in the CPU module in this page. Refer to other related documents or manuals for more information about Ethernet.

3



**Mode** sets the addressing mode of the IP of the CPU module.

If users select **Static**, they can specify an IP address directly. If **DHCP** or **BOOTP** is selected, the IP address is assigned dynamically by a DHCP/BOOTP server.

### 3.1.4.9 Ethernet Port Advance Setting – IP Filter

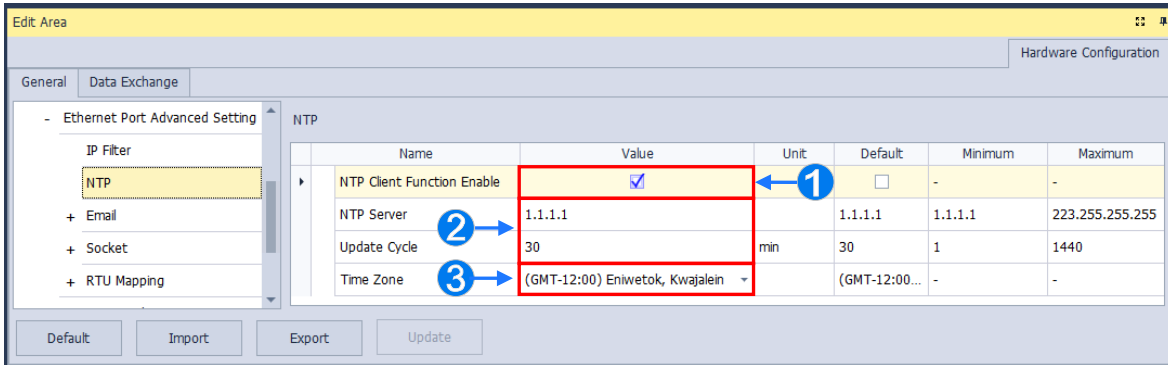
**IP Filter** sets the filter function of network devices. It is another acknowledgement mechanism to ensure the communication object is a device which is allowed by us. The function can limit the communication object. Only the devices in the IP address list are allowed to communicate with the CPU module after the IP filter function is enabled. The data packets sent from devices of other IP addresses will be discarded by the CPU module. The setting steps are shown below. Eighteen groups of IP address at most can be listed in the table.

Name	Value	Unit	Default	Minimum	Maximum
IP Filter Function Enable	<input type="checkbox"/>		<input type="checkbox"/>	-	-
1st Beginning IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
1st Ending IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
2nd Beginning IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
2nd Ending IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
3rd Beginning IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
3rd Ending IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
4th Beginning IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
4th Ending IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
5th Beginning IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
5th Ending IP Address	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255

Select **IP Filter Function Enable** checkbox to enable the IP filter function and type the start address and end address of group 1~8 IP.

### 3.1.4.10 Ethernet Port Advance Setting – NTP

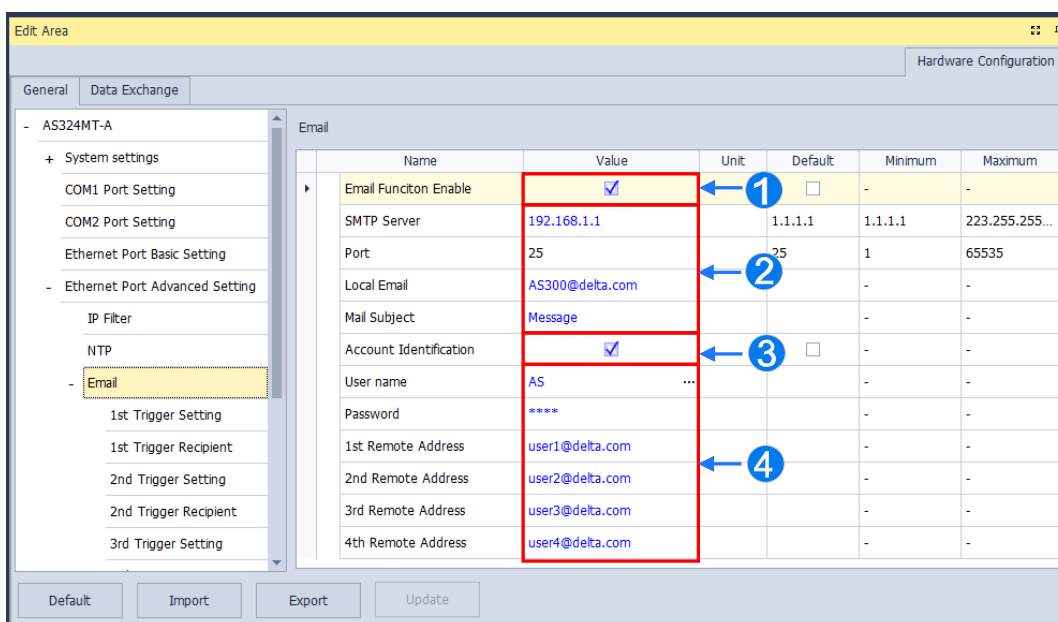
Users can use **NTP** table to enable the CPU module and adjust time via NTP server. Relevant parameters are set here. For more information on NTP, refer to related literature and manuals.



- ❶ Select **NTP Client Function Enable** checkbox to enable the function.
- ❷ Set the address of NTP server and cycle of constant update. Take the figure above as an example. The CPU module connects to the NTP server every 30 minutes.
- ❸ Set up the time zone for the daylight saving time area.

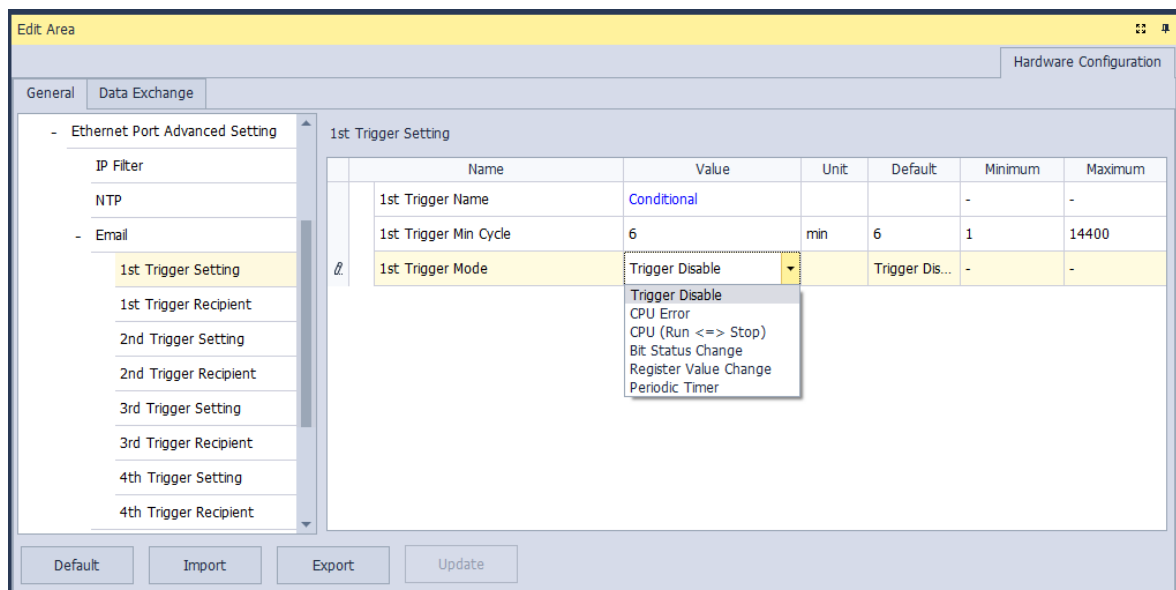
### 3.1.4.11 Ethernet Port Advance Setting – Email

**Email** table is for users to set the email-related functions. The email will be sent to the set email address after the email function is enabled. Total four groups of email sending conditions and four groups of email addresses can be set.



- ❶ Select **Email Function Enable** to enable the function and then start setting the following parameters.
- ❷ Set an IP address of SMTP server. Set the COM port of SMTP server at the COM port and set the sender's email box at local email address. Type a mail subject as the start of the subject of every email.
- ❸ Select **Account identification** checkbox to enable the function. Users can set to authenticate themselves with a user name and a password before logging in to an SMTP server.
- ❹ Type the target email address of a receiver.

Type a trigger name in **Trigger Name box** and a minimum interval in **Trigger Min Cycle** in the **Trigger Setting** table. And then select a trigger condition on the drop-down list. When the sending condition is met, the system will send an email every a period of time. But the same email will not be sent again within the set interval.



Trigger modes can be set as follows.

- **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 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 and an email will be sent.




b) **All Errors:** If an error occurs in the CPU module, the condition of triggering the sending of an email is met and an email will be sent.

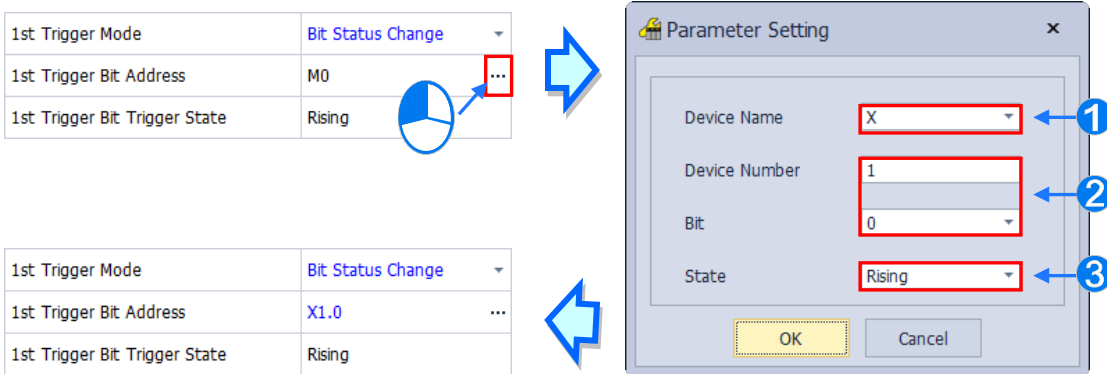
- **CPU (RUN<=>STOP)**

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 and an email will be sent.

### 3

- **Bit Status Change**

If the state of a bit device specified meets a condition set, the sending of an email will be triggered and an email will be sent. 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 click  button in the following window.



1st Trigger Mode	Bit Status Change
1st Trigger Bit Address	M0
1st Trigger Bit Trigger State	Rising

1st Trigger Mode	Bit Status Change
1st Trigger Bit Address	X1.0
1st Trigger Bit Trigger State	Rising

**Parameter Setting**

Device Name: X

Device Number: 1

Bit: 0

State: Rising


OK Cancel

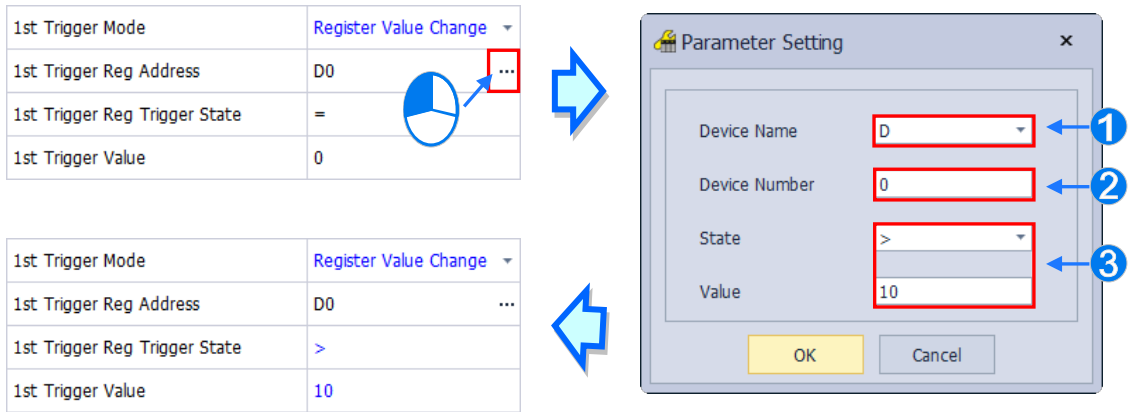
① **Device Name:** Users can select a device type in the **Device Name** drop-down list box.

② **Device Number & Bit:** Users can type a device address in the **Device Number** box. If the device type selected is X/Y, the users have to specify a bit number.

③ **State:** Users can select **Rising** or **Falling** in the **State** drop-down list.

- **Register Value Change**

If the value in a device specified meets a set condition, the sending of an email will be triggered and an email will be sent. For example, if the value in D0 is larger than 10, the condition of triggering the sending of an email will be met and an email will be sent. If users want to set a condition, they can they can click  button in the following window.



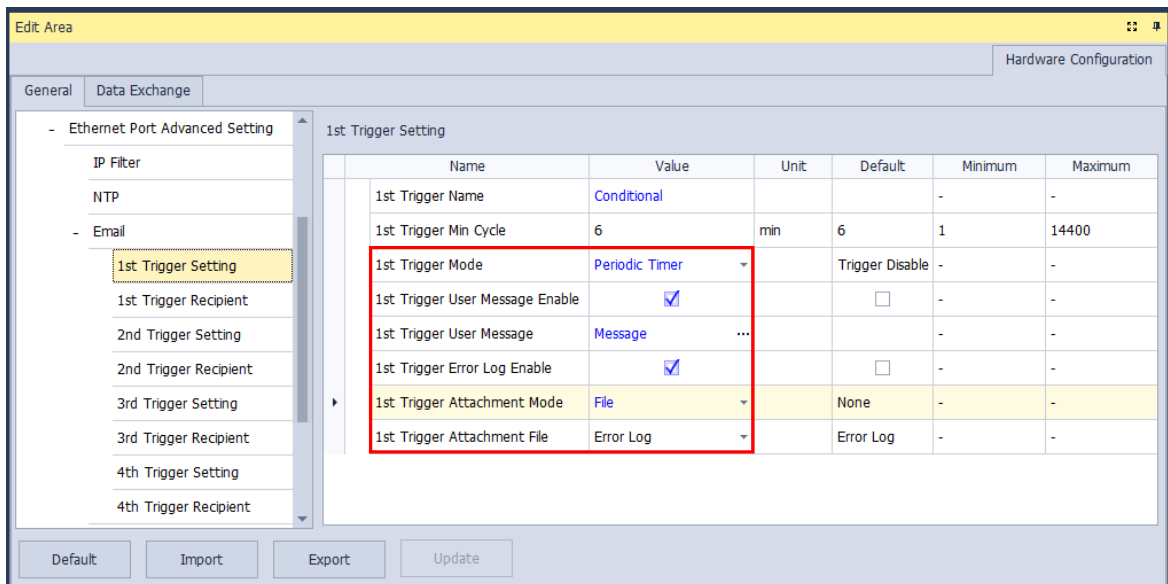
- 1 **Device Name:** Users can select a device type in the **Device Name** drop-down list.
- 2 **Device Number:** Users can type a device address in the **Device Number** field.
- 3 **State & Value:** Users can set s condition of triggering the sending of an email here.


3

● **Periodic Timer**

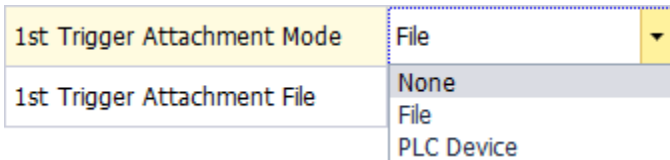
An email is sent periodically. How often an email is sent depends on the interval typed in the **Trigger Min Cycle** box in the **Trigger Setting** section.

Where any trigger mode is set, the user message and error log related parameters will show up.



Select the **Trigger User Message Enable** box and then click  button to the right side of **Trigger User Message**. Type some content as the email text in the pop-up window.

If users select the **Error Log Enable** checkbox in the **Trigger Setting** table, the error log will be added to the email content automatically.



### 3

The options on the drop-down menu of **Trigger Attachment Mode** decide whether to add an attachment to the email. Please make sure the maximum size of the email file allowed before setting an attachment. For more information, refer to relevant operation manuals.


- **None**

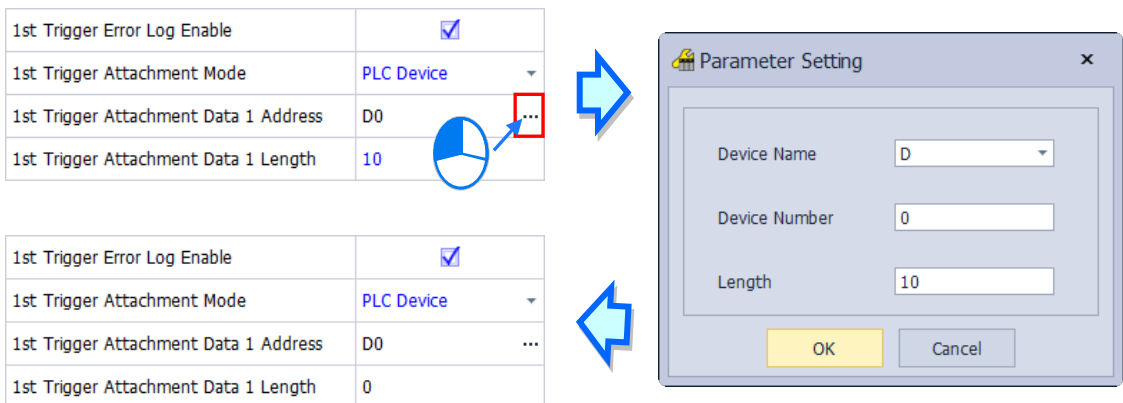
If this option button is selected, no attachment will be inserted.

- **File**

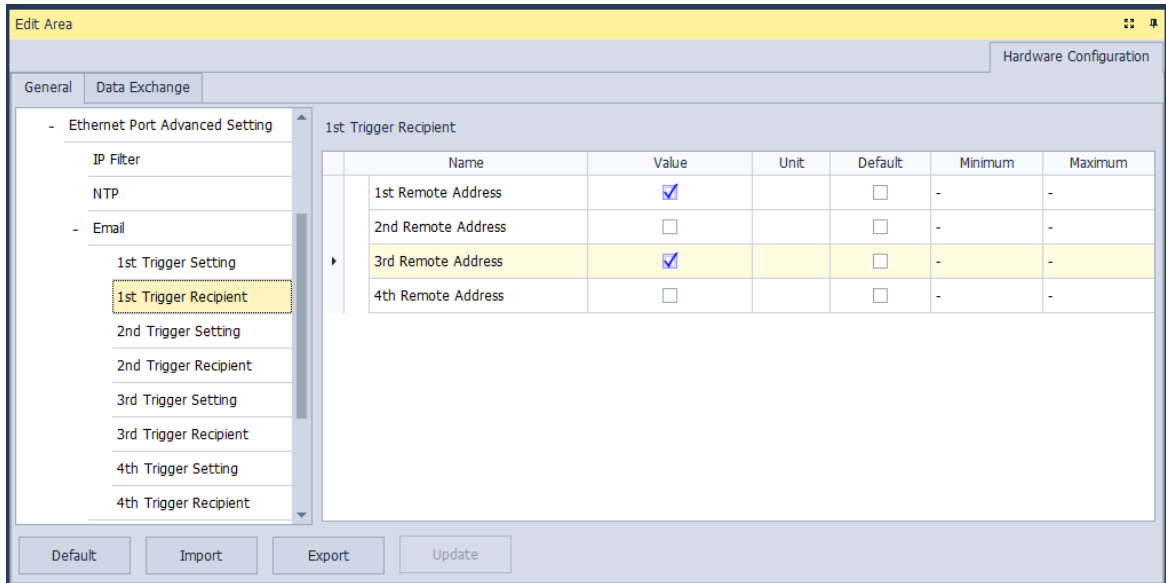
Users can select an error log in the memory card, or the system backup file in the memory card as the attachment of the email.

- **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 click  button in the following window 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.



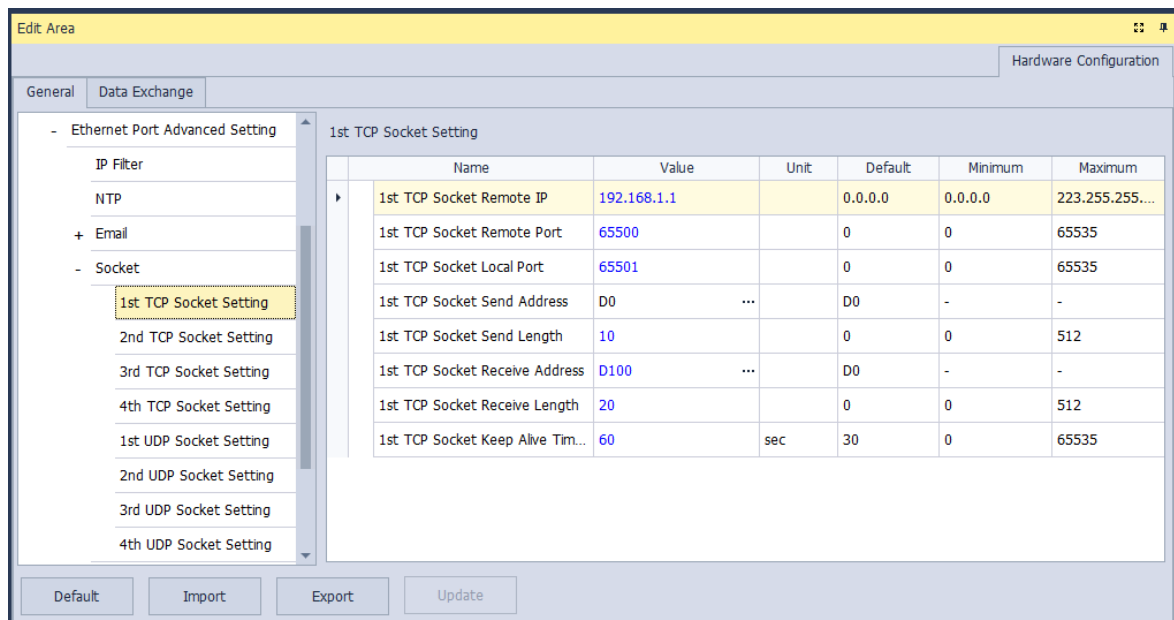
Select the target email address of a receiver when the condition of sending an email is met in **Trigger Receiving** table. And the specific email box is set in the **Email** section.

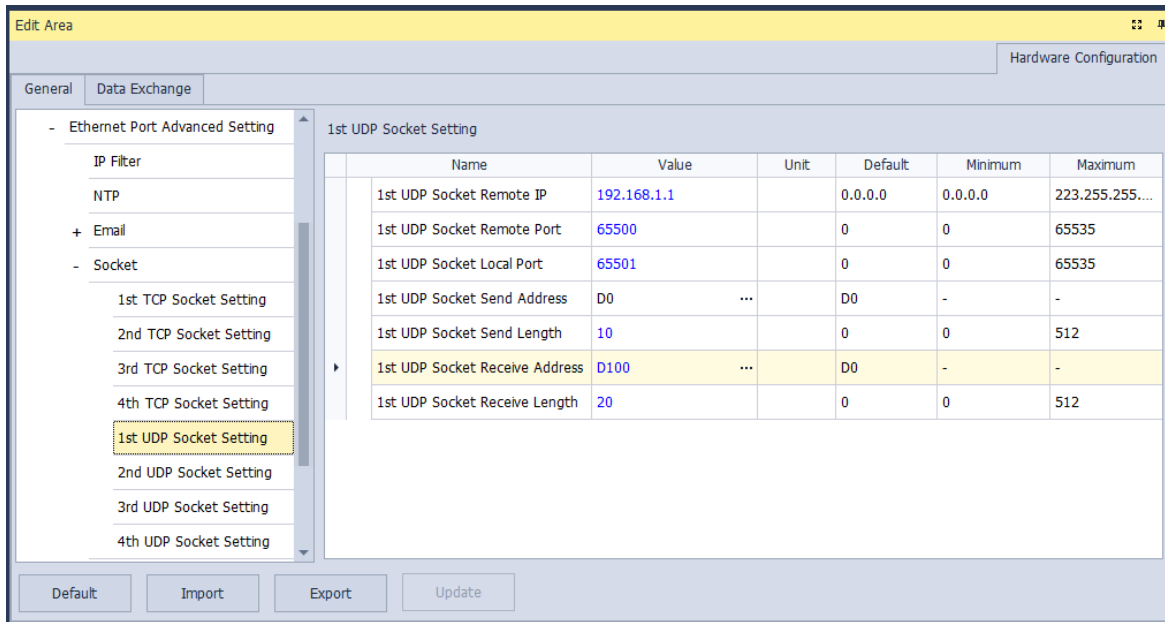


3

### 3.1.4.12 Ethernet Port Advance Setting – Socket

In the **Socket** table, users can set COM port parameters for data transmission via Ethernet. However, the function should be applied with specific API instructions. The system supports the data transmission between the CPU module and other CPU module or device via a socket as well as TCP and UDP protocols in each of which four groups of connections can be set.





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

- **Remote IP:** Users can set a remote IP address.
- **Remote Port:** 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.
- **Local Port:** 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.
- **Send Address:** 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 512 bytes.
- **Send Length:** Users can set an initial device in the CPU module where data which will be sent is stored.
- **Receive Address:** 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 512 bytes.
- **Receive Length:** Users can set an initial device in the CPU module where data which will be received is stored.
- **Keep Alive Timer:** Users can set a maximum keep alive time for the connection. If no data is transmitted, and the keep alive 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 cannot be the same, and the devices where data which will be sent is stored cannot 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.

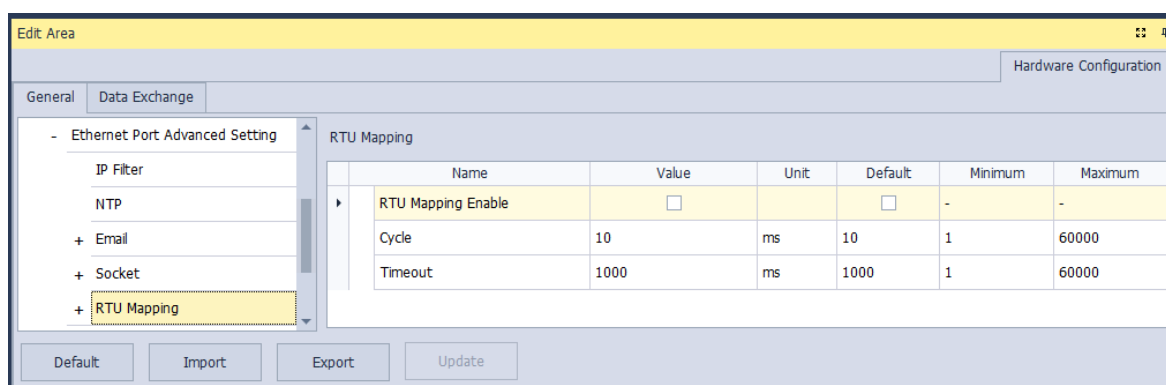
If the local CPU module wants to send 10-word data to the remote device, the data will be stored in D0~D9 before the data is sent. If the local CPU module receives 20-word data from the remote device, the data will be stored in D100~D119.

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.

### 3.1.4.13 Ethernet Port Advance Setting – RTU Mapping

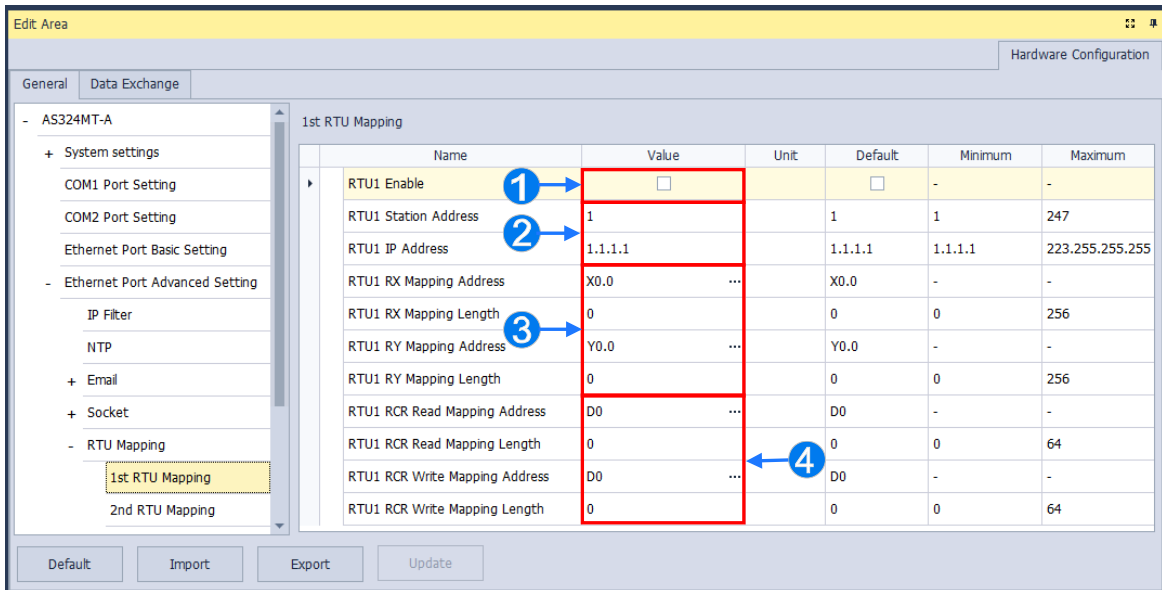
In the **RTU Mapping** table, users can set Delta RTU-EN01 slave in connection with PLC CPU. The remote device can be controlled via Ethernet. For the setting and operation of RTU-EN01, refer to the operation manual.



Select RTU Mapping Enable checkbox to enable the function of connection between RTU module and PLC CPU. Set the update cycle in **Cycle** box and a timeout in **Timeout** box. It means a timeout when RTU does not give a reply within the timeout set.

The data mapping between each group of RTU-EN01 and PLC CPU is set in the **RTU Mapping** section. RTU-EN01 and I/O module connected to it are set via DCISoft. For more information on DCISoft, refer to the operation manual.

3



- ❶ Users can select RTU1 Enable to enable this RTU function. Each RTU can be set individually.
- ❷ Users can set a station address and IP address.
- ❸ Users can set RX mapping address and RX mapping length to make the digital input points of DI module connected to RTU mapped to X/M devices and length of PLC CPU. While, the RY mapping address and length set is used for making the digital output points of DO module mapped to Y/M devices and length of PLC CPU.
- ❹ Users can set RCR Read mapping address and RCR Read mapping length to make the analog input points of AI module connected to RTU mapped to D/SR devices and length of PLC CPU. While, the RCR Write mapping address and length set is used for making the analog output points of AO module mapped to D/SR devices and length of PLC CPU.

### 3.1.4.14 Function Card 1 Setting

The **Function Card 1 Setting** table is used for setting parameters of function card 1 in AS300 series CPU module, which is installed by inserting to the slot.

Name	Value	Unit	Default	Minimum	Maximum
Card 1 Detect mode	Auto Detect		Auto Detect	-	-
Manual Select Card	None		None	-	-
Card 1 ID No.	1		1	1	254
Protocol Setup Opportunity	Stop --> Run		Stop --> Run	-	-
Baud Rate	9600	bps	9600	-	-
Data bit	7	bit	7	-	-
Parity bit	Even		Even	-	-
Stop bit	1	bit	1	-	-
MODBUS mode	ASCII		ASCII	-	-
Delay time to Reply	0	ms	0	0	3000
Received Data Timeout	200	ms	200	0	3000
F2AD Analog Input mode	0~10V		0~10V	-	-
F2DA Analog Output mode	0~10V		0~10V	-	-
F2AD Sampling Time	3	ms	3	3	15
F2AD Average Times	10		10	1	15

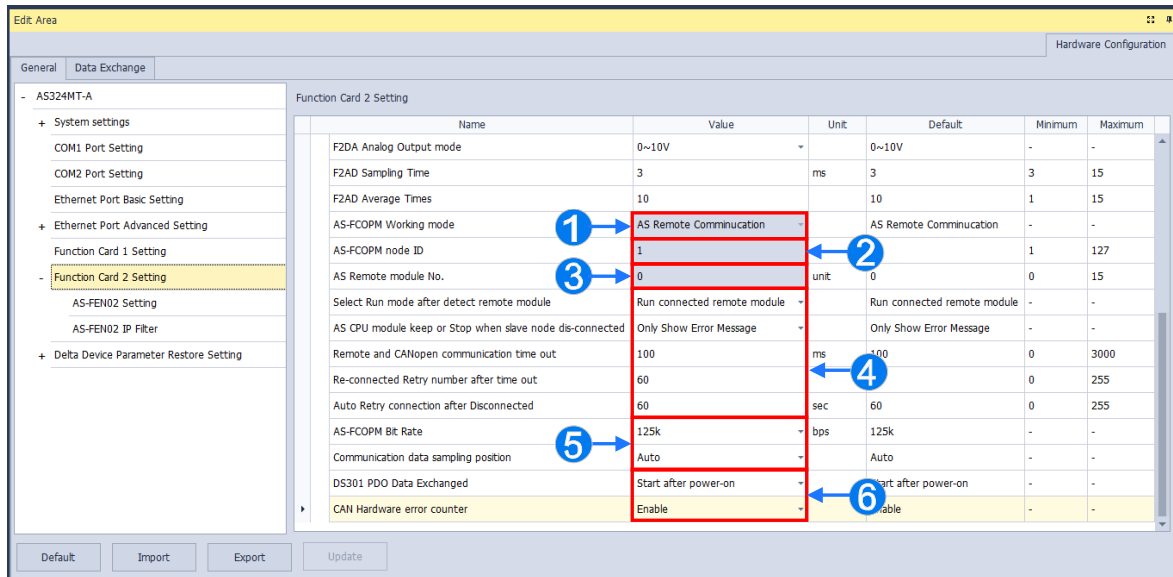
- ❶ Users can select **Auto** in **Card1 Detect mode** box to detect the actual card model and setting or **Manual** to select the AS--F232/ F422/ F485/F2AD/F2DA function card which need be configured in the **Manual Select Card** box.
- ❷ For the setting of parameters in AS--F232/ F422/ F485.
- ❸ Users can set **F2AD** to receive the signal of 0~10V or 4~20mA in **F2AD Analog Input Mode** box and set F2AD to send the signal of 0~10V or 4~20mA in **F2DA Analog Output Mode** box. The average times are set in **F2AD Sampling Time** and **Average Times** boxes respectively.



### 3.1.4.15 Function Card 2 Setting

The Function Card 2 Setting table is used for setting parameters of function card 2 in AS300 series CPU module, which is installed by inserting to the slot. Besides function card 2 settings, the option also include settings for AS-FEN02 and IP filter. For function card 2, AS-FCOPM and AS-FEN02 in **Manual Select Card** box are additional options that does not exist for function card 1. If users select AS-FCOPM card, the parameters can be set in the following window.

3



- ① Users can select AS Remote Control/Delta Servo Drives and AS Remote Module/CANopen DS301 from the **AS-FCOPM working mode** box. The AS Remote Module and Delta Servo Drives adopts communication protocols exclusive to Delta servo products and AC motor drives. In addition, the AS Remote Module and Delta Servo Drives are also applied in connecting remote I/O AS series as well as CANopen DS301 for the application of DS301 standard protocol.
- ② Users can set the address of the station when CANopen DS301 is selected from the **AS-FCOPM working mode** box.
- ③ The remote I/O parameters can be set if AS Remote Mode is selected from the AS-FCOPM working mode box. The **AS Remote Module No.** sets the number of remote I/O modules and each one can connect an expansion module.
- ④ If AS Remote Mode/Delta servo drives and AS Remote Mode is selected from the AS-FCOPM working mode box, users can **Select run mode after detect remote module** to run the number of actual connected modules inconsistent with the set number of modules, then the actual connected remote module operating or all remote modules will stop running; **AS CPU keep or Stop when slave node dis-connected** only shows errors or stops running and sends an alarm when the remote I/O module is disconnected during operation; **AS**

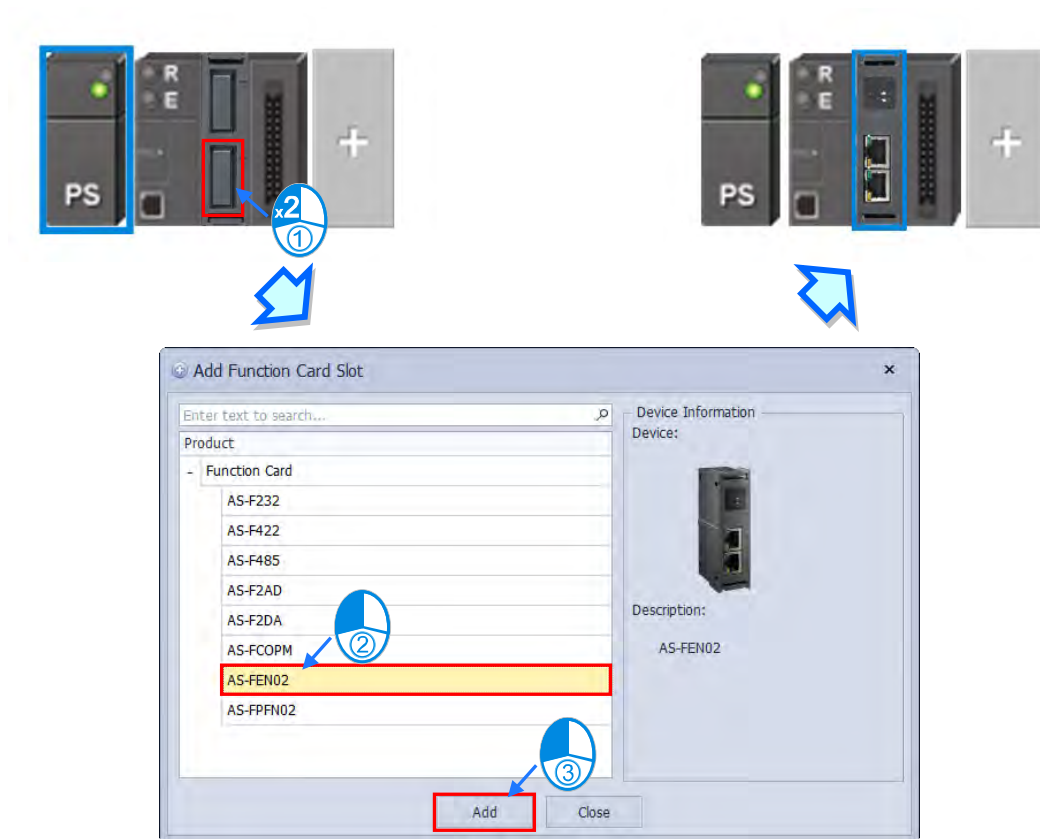
**Remote and CANopen Communication Timeout** sets the timeout during which the remote I/O module does not give a reply; **Auto Retry connection after Disconnected** sets the times of retrying to make a connection after a timeout occurs.

- ⑤ All three AS-FCOPM working modes can set the **AS-FCOPM Bit Rate**.
- ⑥ The **DS301PDO Data Exchanged** parameter contains **Start after power** and **PLC RUN**

Will work when CANopen DS301 protocol is selected from the AS-FCOPM working mode box.

Insert AS-FEN02 by clicking the function card slot 2 and selecting AS-FEN02, as the image shown below.

3



You can also set AS-FEN02 card in the Edit Area. Set AS-FEN02 as **Manual Select Card** (Note: If AS-FEN02 card is selected from **Manual Select Card**, then Function Card 1 cannot be set.) After that you can set up Ethernet communication parameters.

3

The screenshot shows the 'Edit Area' window with the 'Hardware Configuration' tab selected. The left sidebar shows a tree view under 'AS324MT-A' with 'Function Card 2 Setting' selected. The main area displays a table for 'Function Card 2 Setting' with the following data:

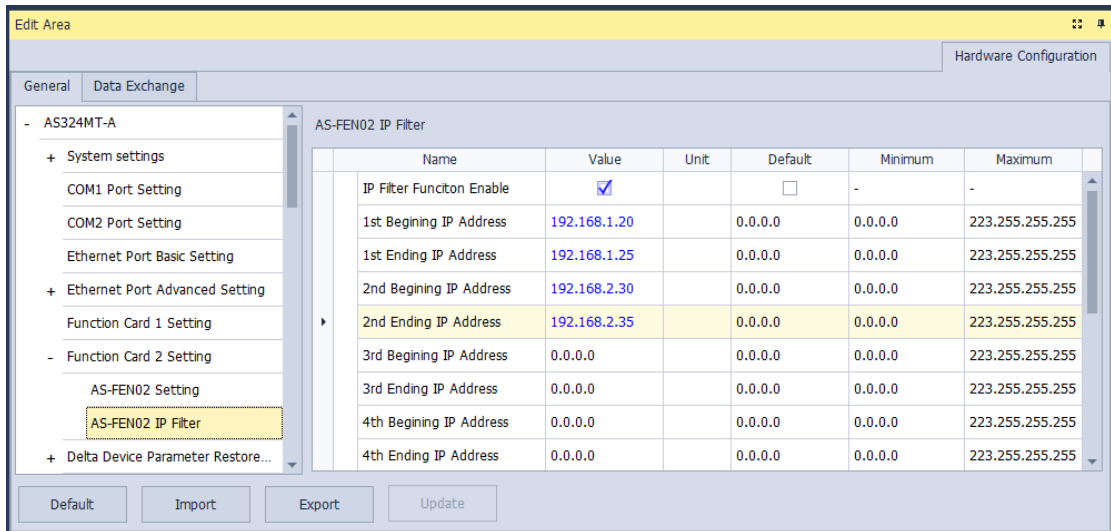
Name	Value	Unit	Default	Minimum	Maximum
Card 2 Detect mode	Manual		Auto Detect	-	-
Manual Select Card	AS-FEN02 Card		None	-	-
Card 2 ID No.	1		1	1	254
Protocol Setup Opportunity	Stop -> Run		Stop -> Run	-	-
Baud Rate	9600	bps	9600	-	-
Data bit	7	bit	7	-	-
Parity bit	Even		Even	-	-
Stop bit	1	bit	1	-	-
MODBUS mode	ASCII		ASCII	-	-



The screenshot shows the 'Edit Area' window with the 'Hardware Configuration' tab selected. The left sidebar shows a tree view under 'AS324MT-A' with 'AS-FEN02 Setting' selected. The main area displays a table for 'AS-FEN02 Setting' with the following data:

Name	Value	Unit	Default	Minimum	Maximum
IP Address	192.168.1.5		192.168.1.5	1.1.1.1	223.255.255.255
Subnet Mask	255.255.255.0		255.255.255.0	0.0.0.0	255.255.255.255
Gateway	192.168.1.1		192.168.1.1	1.1.1.1	223.255.255.255
TCP Keep Alive Timeout	30	sec	30	1	65535
Mode	Static		Static	-	-

The IP filter adds another layer of confirmation to ensure the devices for communication are permitted. This function can also limit communication objects. When IP filter function enables, only the devices within the listed IP address from the IP filter table are allowed to communicate with AS-FEN02. By selecting **IP Filter Function Enable**, users can input the beginning and ending of 1-8 set of IP address.



3

### 3.1.4.16 Delta Device Parameters Backup and Restore

ISPSOft V3.10 supports Delta device parameters backup and restore. The firmware of your AS PLC CPU should be V1.08.10 or higher and you need to use EIP Builder V1.70 or later to backup / restore Delta device parameters. Now ISPSOft V3.10 only supports Delta C2000 inverter for parameters backup and restoration.

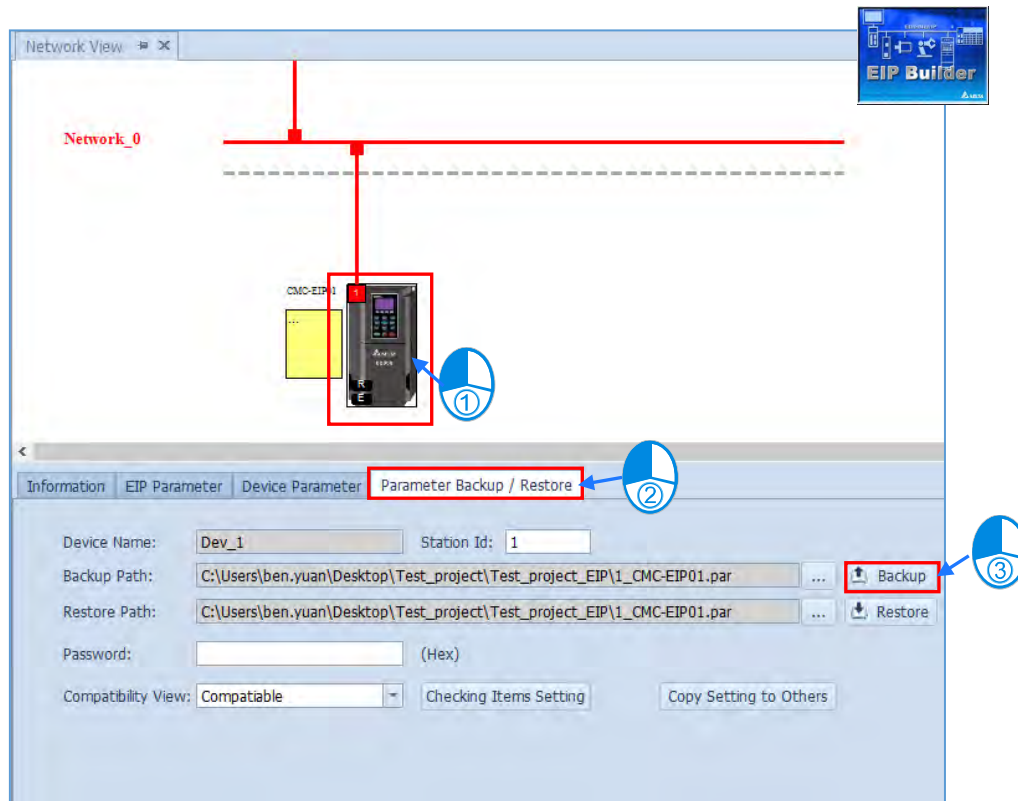
After you back up the parameters, with ISPSOft and EIP Builder, you can restore the parameters to your devices. Up to 32 Delta devices are supported. Refer to AS Series Operation Manual for further information.

## 3

#### Steps to back up Delta device parameters to your PC

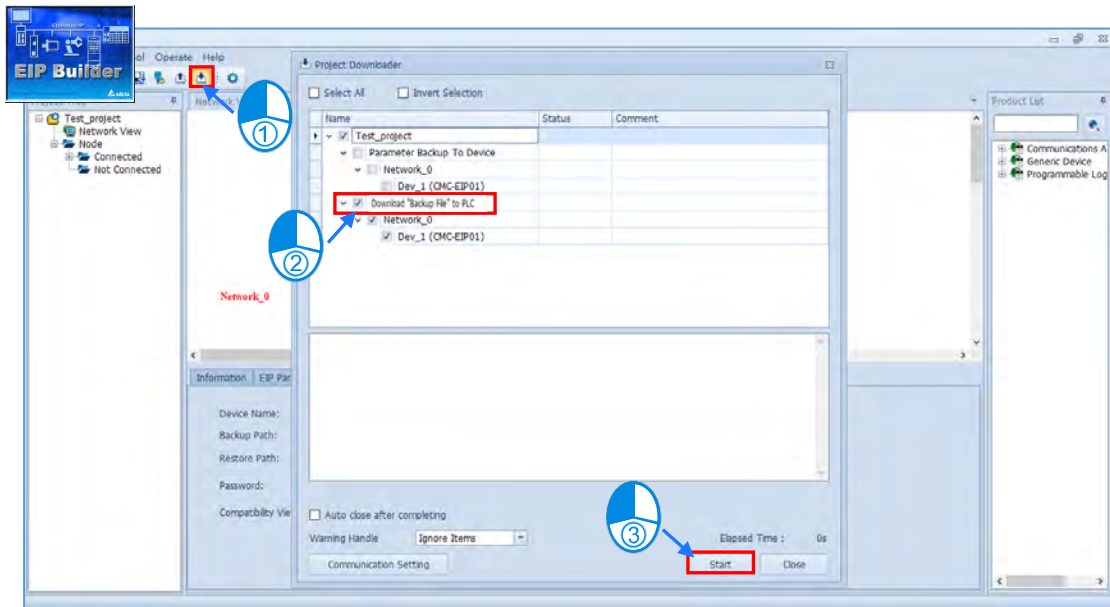
1. Open EIP Builder (refer to section 3.1.3.16) and complete the network planning.
2. Click CMC-EIP01 (Delta C2000) and click **Parameter Backup / Restore** tab.
3. Select a backup path and click **Backup** button to back up Delta C2000 parameters.

The backup file is stored as the .par file. For multiple devices, the file name varies according to different station number for easier identification.



**Steps to store Delta device parameters from your PC to the SD card on the PLC**

1. Make sure the SD card is installed on the PLC.
2. Make sure that AS Series PLC CPU is at the state of STOP.
3. Click download on the tool bar and select the option **Download “Backup File” to PLC**.
3. Click **Start** button to store the Delta C2000 parameter backup file to the SD card.



**Remarks:**

- Default backup file path
  - AS300: \SDCard\PLC CARD\AS300\DevPara
  - AS200: \SDCard\PLC CARD\AS200\DevPara
- File naming: mmmm\_nn\_tt\_pppppppppppppppp.par
  - mmmm: device or module code (e.g. PLC CPU is 0000.)
  - nn: communication port number (e.g. PLC Ethernet communication port is 03.)
  - tt: communication protocol type (e.g. Ethernet communication is fixed to 01.)
  - pppppppppppppppp: device information (IP + ID + 000000 in the hexadecimal format)

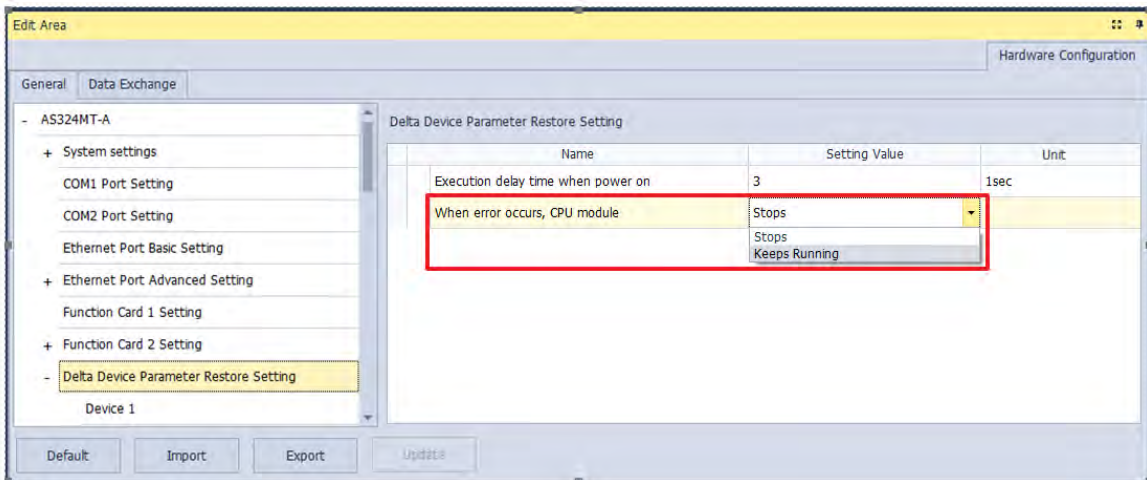
For example: If the device IP is 192.168.1.9 and its ID is 9, the device information is C0A80109 (IP) + 09 (ID) + 000000 = C0A8010909000000

### Delta Device Parameter Restore Setting

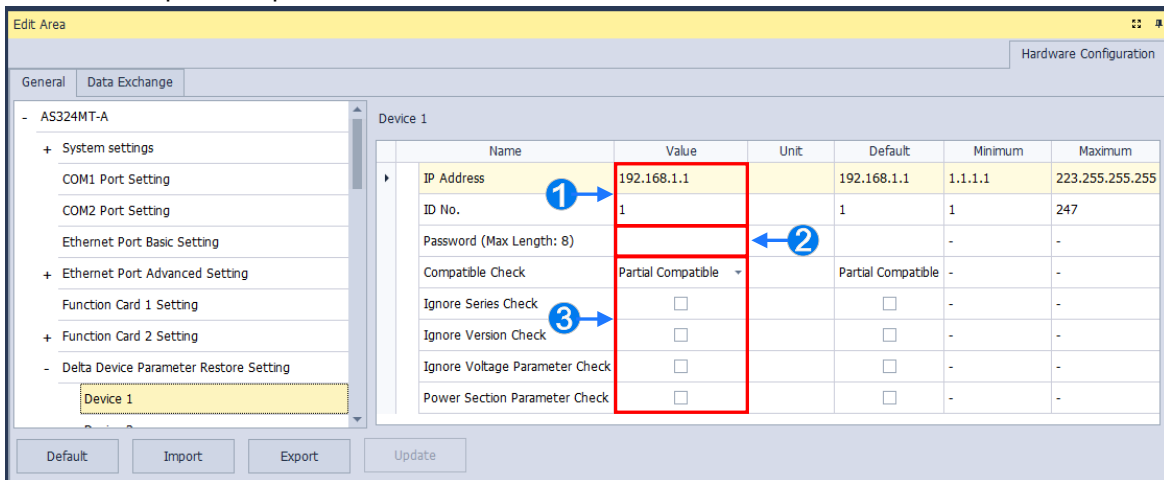
You can set up the handling method when an error occurs while restoring.

- Execution delay time when power on
- When error occurs, CPU module stops or keeps running.

3



You can set up further parameters for Delta devices 1 ~ 32.



- 1 Set up the IP address and ID number for the device.
- 2 Set up the password for the device; up to 8 characters can be used.
- 3 Set up the compatible check. 3 different checks, **Partial Compatible**, **Full Compatible**, and **Force Write** can be set.
  - \* **Partial Compatible:** you can set up which of the four following conditions can be ignored.
    - **Ignore Series Check if Possible:** Even if the series in the backup file is different from the one on the list, the system still tries to restore the parameters.

- **Ignore Version Check if Possible:** Even if the version in the backup file is different from the one on the list, the system still tries to restore the parameters.

- **Ignore Voltage Parameter Check if Possible:** Even if the voltage parameter in the backup file is different from the one on the list, the system still tries to restore the parameters.

- **Ignore Power Section Parameter Check if Possible:** Even if the power parameter in the backup file is different from the one on the list, the system still tries to restore the parameters.

Note: If the target device does NOT support the partial compatibility, the restoration may not begin. Before restoration begins, you need to check which condition is supported by the target device by referring to the target device manual.

\* **Full Compatible:** All the conditions should be met before the restoration begin.

\* **Force Write:** Ignore any difference between the backup file and the device on the list and force the system to restore the parameters.

Besides using ISPsoft and EIP Builder, you can also use other PLC editing software, HMI or SCADA to work with SR devices to restore the parameters. Refer to the following SR device for more information. After you have set the configurations, the PLC executes the restorations according to the settings when the PLC is Power-On.

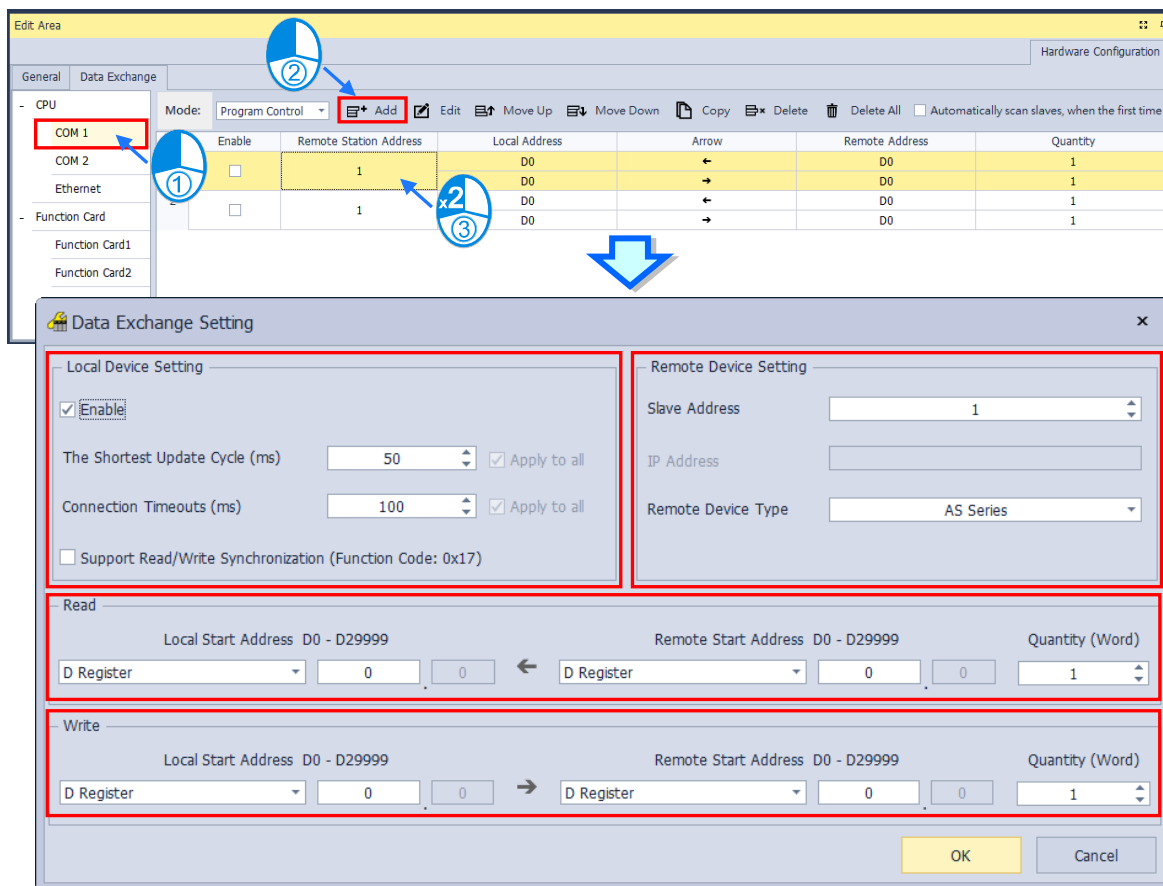
SR	Function	Description	Retain ability
SR1600	Parameter restoration code	0 : Disable (default) 1 : When PLC is Power-ON or Stop→Run, execute the restoration. Once the restoration is complete, clear the value to 0.	Yes
SR1601.0~SR1601.15 ⋮ SR1602.0~SR1602.15	Target device number	SR1601.0: refer to target device #1 SR1601.1: refer to target device #1 ⋮ SR1602.15: refer to target device #32  Off : Disable (default) On : Execute the restoration on the appointed target device	Yes



### 3.1.4.17 Data Exchange

On the Data Exchange Tab, you can set up data exchange for COM communication, Ethernet communication and Function Card. Click **Add** and then **Edit** or double-click the data exchange table to open the editing page.

3



- **Local Device Setting**

Tick the option **Enable** for data exchange to start. You can set the shortest update cycle and connection timeout time in ms. If for a period of time that you have set, there is no response from the target device, this is considered as a timeout. If you tick the option **Support Read/Write Synchronization (Function Code: 0x17)**, the master PLC CPU can use MODBUS function code to complete read and write synchronization at one operation. However you need to make sure all the devices support MODBUS function codes; otherwise, the slaves devices may NOT recognize the function code and fail to complete read/write synchronization.

- **Remote Device Setting**

**Slave Address:** station number of the slave device for data exchange

**IP Address:** IP address of the slave device (only available for EtherNet)

**Remote Device Type:** slave device model type, such as Delta PLC or standard MODBUS devices

- **Read:** When PLC CPU reads data from the remote device: PLC CPU defines a device range to store data including the device type, starting address and quantity which are read from the remote device. Define the device type, starting address and quantity in the remote device that will be read by PLC CPU.

**Local Start Address:** Device type and start address of devices where PLC CPU store data

**Remote Start Address:** Device type and start address of the remote device to be read

**Quantity:** Data length of input

- **Write:** PLC CPU writes data to a remote device; PLC CPU defines a device range for the remote device to read the following data, the device type, starting address and quantity. Define the device type, starting address and quantity in the remote device that will be written by the PLC CPU.

**Local Start Address:** Device type and start address of the source data of PLC CPU

**Remote Start Address:** Device type and start address of the remote device where data are to be written

**Quantity:** Data length of output



## 3.2 Hardware Configuration Tool – HWCONFIG 3.0

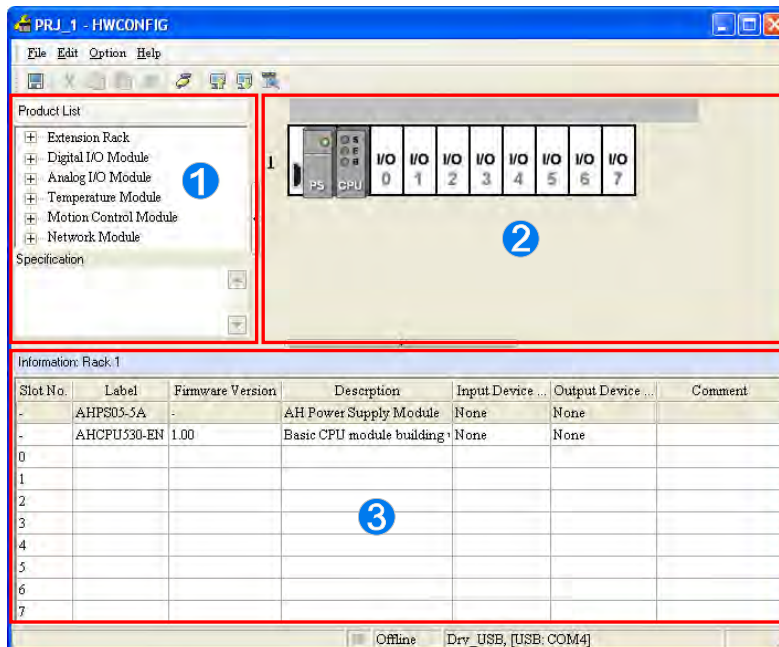
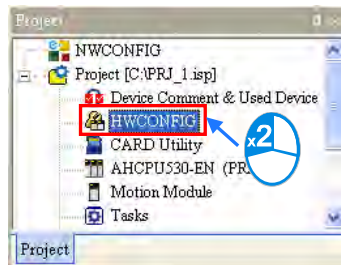
HWCONFIG is one of the tools in ISPSOft for hardware configuration. Its functions include configuration of module racks, parameter settings for modules, download/upload hardware parameters and simple on-line detection and diagnosis function. This section will introduce the functions mentioned above and point out some model types and its unique functions.

**Attention! When parameter settings regarding hardware configuration are complete, the settings are effective once downloaded into PLC CPUs.**

### 3

#### 3.2.1 HWCONFIG Environment

Users can double-click **HWCONFIG** from the Project section to open the setting window.

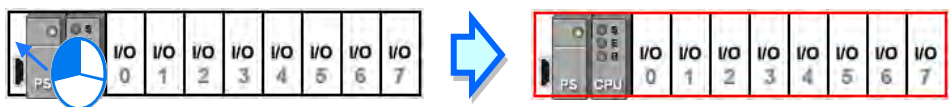


- ① **Product list:** Available hardware are listed in the box.
- ② **System configuration area:** This is the main work area for system configuration and settings.
- ③ **Information:** Manages the present system configuration through a list.

In **System Configuration**, the area uses graphics to demonstrate the current configuration. For AH series PLC, each module is represented by a graphic and the backplane number of a module is shown on the left of the graphic; when operating, use the mouse to click on the selected module in the graphic.

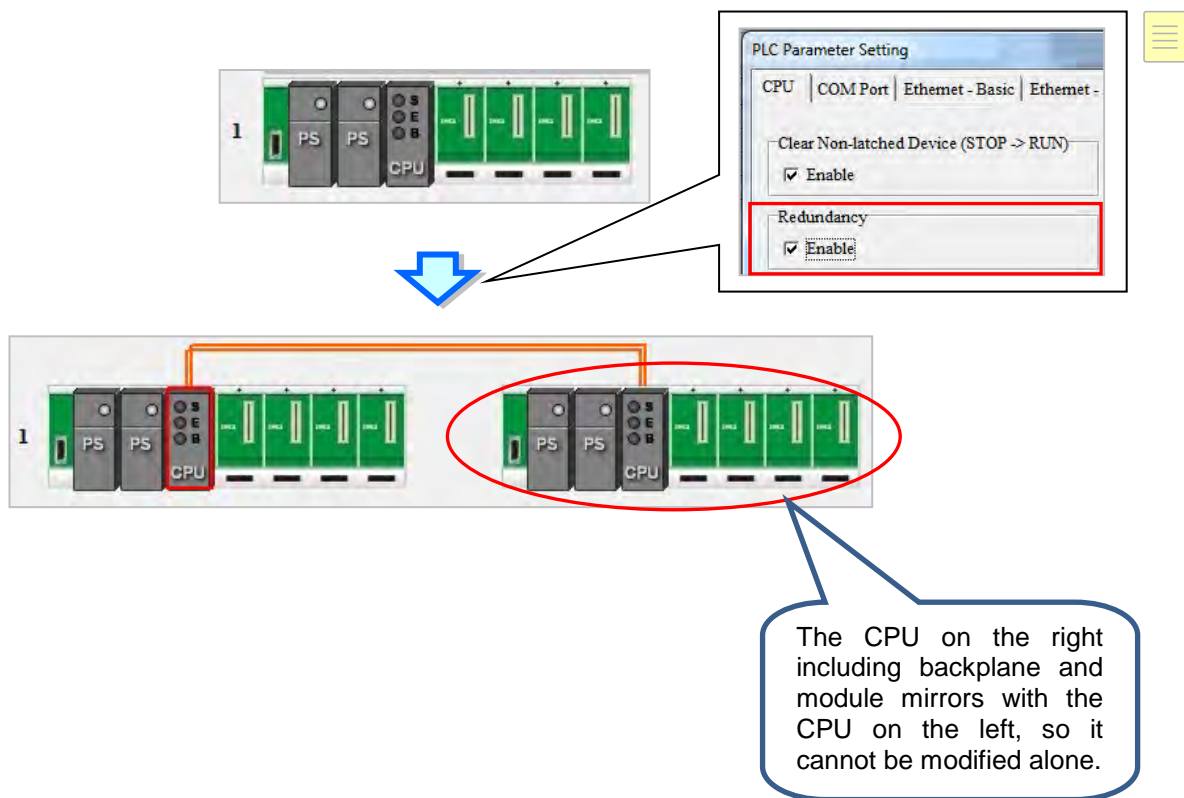


When selecting a backplane, click on the connected port on the left of the graphic.

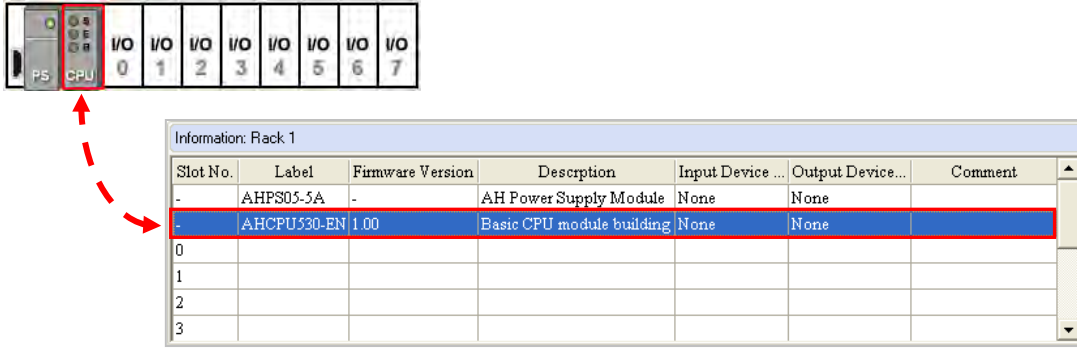


3

When using AH560 redundant PLC series, double-click on the CPU and select **Enable** from the **PLC Parameter Setting** window where users will find a redundant CPU appear on the right with settings that mirror the CPU on the left, but cannot be modified alone.



The system configuration area and Information section will update together in real-time. So, when a module is chosen from the system configuration area and Information section, the module is displayed as selected in both places.



When using DVPxxMC series, a HWCONFIG window is shown below.

3



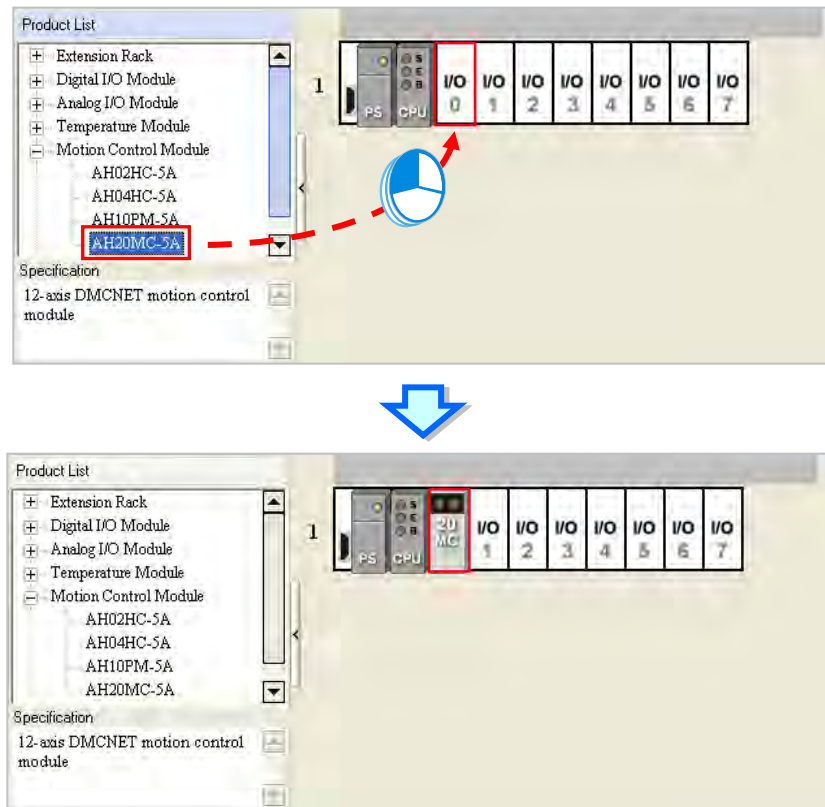
- 1 PLC CPU image.
- 2 **Description:** Describes the PLC CPU functions.
- 3 **Comment:** Comments concerning the PLC CPU.
- 4 **Version:** DDF version.
- 5 **Parameter:** Click **Settings** to enter PLC Parameter Settings
- 6 **CANopen:** Click **CANopen Builder** to open the software.

### 3.2.2 Module Configuration

#### 3.2.2.1 Add Module

- Method 1

Select the desired module from the left Product List regarding AH series and drag that module to the desired vacant slot on the right.

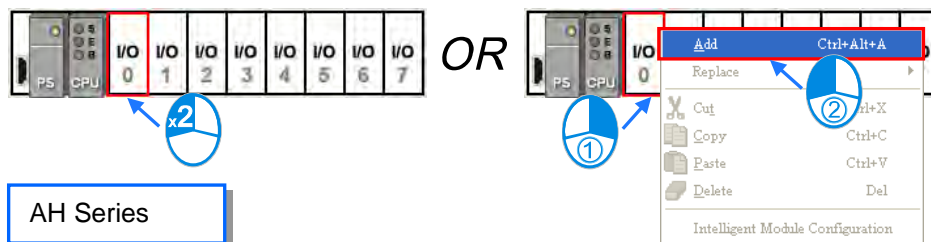


3

\*. When a selected module is dragged to the system configuration area on the right, the module is placed in a vacant slot. The system does not allow the selected module be placed in an existed module slot.

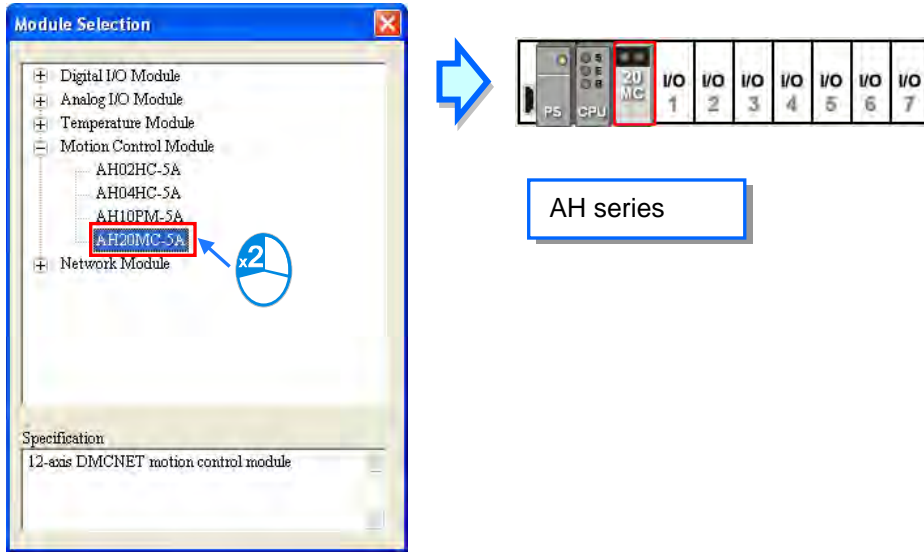
- Method 2

For AH series PLCs, double-click a vacant slot or right-click the slot and choose **Add**. For AS series PLCs, double-click on the + position or right-click the position and choose **Open**.



(2) Double-click on the selected module to add.

3



● Method 3

(1) In **Information**, double-click on the blank row of the corresponding slot or right-click and choose Add to see the Module Selection window.

Information: Rack 1

Slot No.	Label	Firmware ...	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU330-EN	1.00	Basic CPU module building	None	None	
0						
1						
2						
3						
4						
5						
6						
7						

OR

Information: Rack 1

Slot No.	Label	Firmware ...	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU330-EN	1.00	Basic CPU module building	None	None	
0						
1						
2						
3						
4						
5						
6						
7						

①

②

Add      Ctrl+Alt+A

Replace

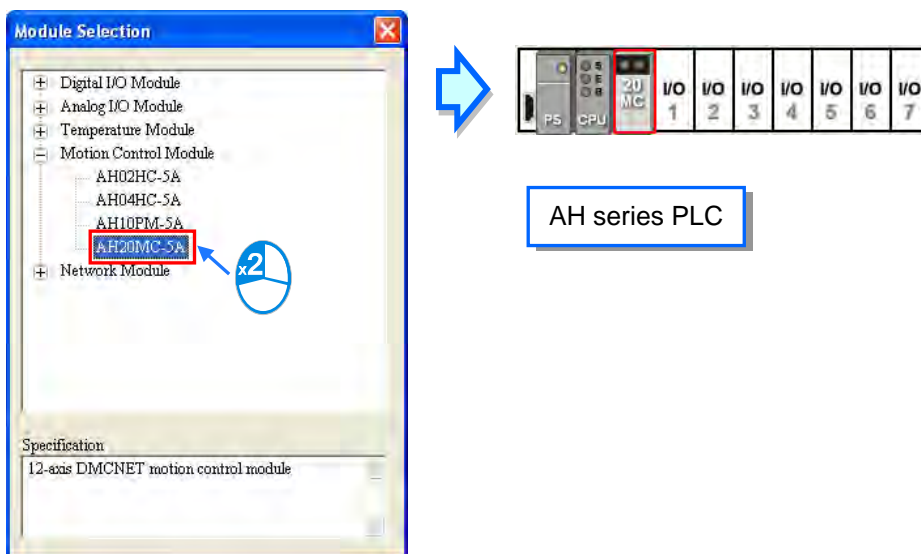
Cut      Ctrl+C

Copy      Ctrl+V

Paste      Del

Intelligent Module Configuration

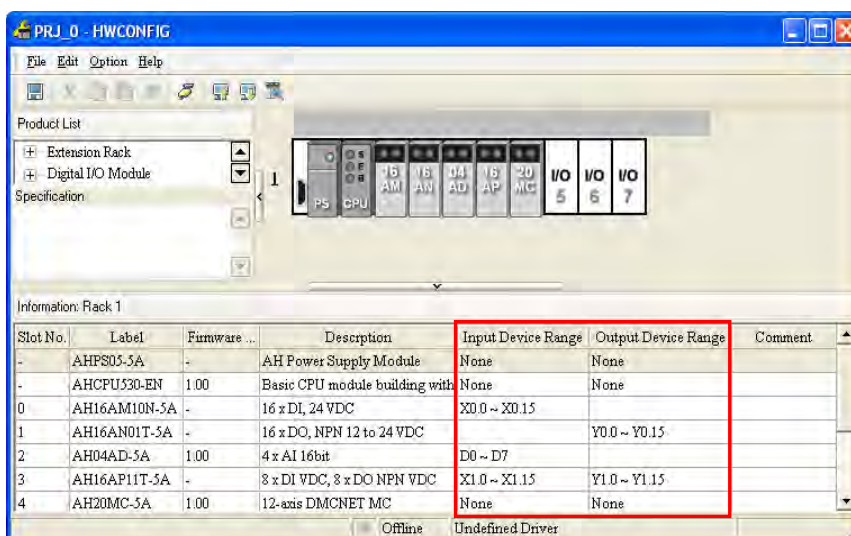
- (2) By clicking on a selected module from the list, users can view the **Specification** containing the module information and double-click to add the module.



3

### 3.2.2.2 Address Assignment of Modules

When data updates are constantly required for AH series PLCs, for example, analog input modules constantly receives analog signals, the system will automatically assign a device address to save data. The assigned device address is shown in **Information** containing **Input Device Range** and **Output Device Range** (see below). For AS Series, the group PLC do not support manual settings for address assignment but provides auto configuration on fixed IP address according to the installation position of the modules. For RTU, users can only set up the starting device IP of the first SCM module then the following modules are assigned accordingly. Please refer to section 3.5.3 for AS series remote module configuration.





● **AH16AM10N-5A Module (slot 0)**

This is a 16-point digital input module. The system assigns X0.0~X0.15 that is 1 WORD device in total to store and input module status regarding all points.

● **AH16AN01T-5A Module (slot 1)**

This is a 16-point digital output module. The system assigns Y0.0~Y0.15 that is 1 WORD device in total to store and output module status regarding all points.

3

● **AH04AD-5A Module (slot 2)**

This is a 4-channel analog input module. The system assigns D0~D7 which is 8 WORD in total and the conversion of external analog input signal is stored in D0~D7.

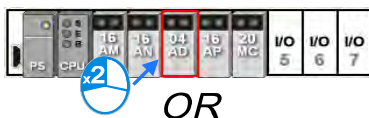
● **AH16AP11T-5A Module (slot 3)**

This is a 8-point digital input/output module. Since the assigned device requires at least 1 word unit, the system assigns X1.0~X1.15 (actual range of use: X1.0 ~ X1.7) device to store and input module status regarding all points; in addition, Y1.0 ~ Y1.15 (actual range of use: Y1.0 ~ Y1.7) is assigned to store and output module status regarding all points.

● **AH20MC-5A Module (slot 4)**

This is a motion control module, but since the module does not require constant data updates, therefore, the system does not assign device address. To obtain the corresponding functions of the assigned device address, double-click on the graphic module or the row on Information to open the Parameter Setting window for the module.

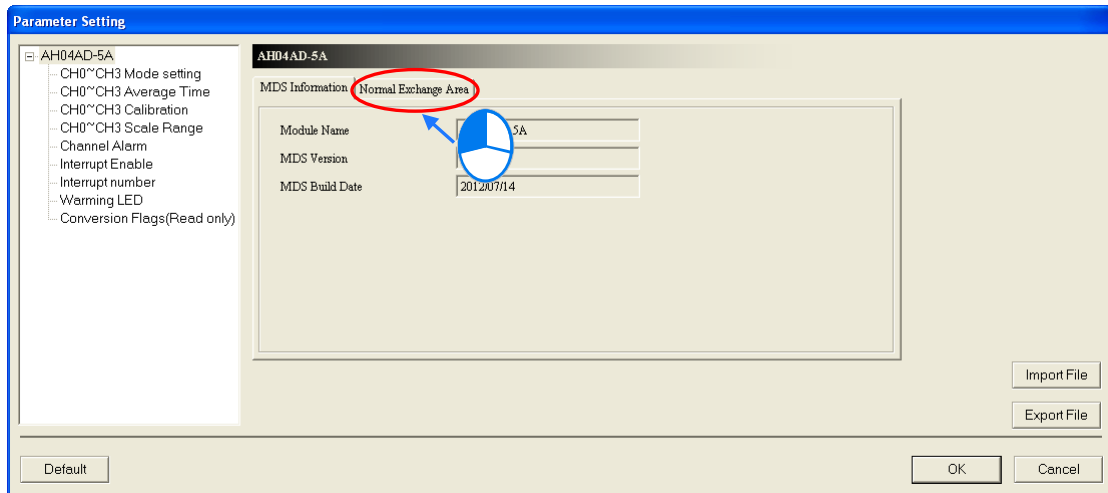
If users want to know functions to which devices assigned correspond, they can double-click the module or the information about the module on the information list to open the **Parameter Setting** window.



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	AH16AM10N-5A	-	16 x DI, 24 VDC	X0.0 ~ X0.15		
1	AH16AN01T-5A	-	16 x DO, NPN 12 to 24 VDC		Y0.0 ~ Y0.15	
2	AH04AD-5A	1.00	4 x AI 16bit	D0 ~ D7		
3	AH16AP11T-5A	-	8 x DI VDC, 8 x DO NPN VDC	X1.0 ~ X1.15	Y1.0 ~ Y1.15	
4	AH20MC-5A	-	12-axis DMCNET MC	None	None	
5						
6						
7						

\*. When users double-clicks the row on Information box, please click on either the Slot No., Label, or Description cell to avoid columns for edit.

Click the **Normal Exchange Area** tab in the **Parameter Setting** window to see a list regarding device address and corresponding descriptions.



3

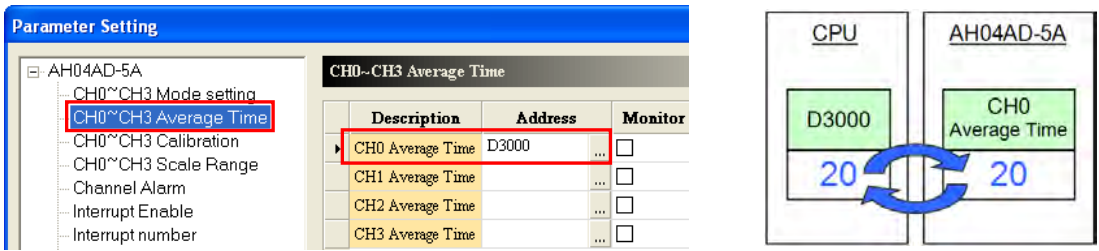


MDS Information		Normal Exchange Area
	Description	Address
▶	CH0 Input value	D0 - D1
	CH1 Input value	D2 - D3
	CH2 Input value	D4 - D5
	CH3 Input value	D6 - D7

In order to update data constantly, the system uses auto-configuration for the device in storing data. As for internal parameters not updated frequently in modules, users can define the corresponding device D and when the system starts operating, the corresponding device D in the PLC CPU will align with internal parameters of the assigned module. As a result, users can indirectly save internal data of the module through corresponding device D. This is more effective than using the traditional **FROM / TO** command to save internal data of the module.

For example, the **AH04AD-5A** module parameter setting window has D3000 defined as the module parameter – **CH0 Average Time** of **corresponding device D**. Thus, after HWCONFIG parameters are downloaded to the PLC CPUs, users can directly modify D3000 content value in the PLC CPU and indirectly modify **CH0 Average Time** of **AH04AD-5A** module.

In addition, if modules parameters written in corresponding device D are not permitted, the system will restore the parameters to its original values.

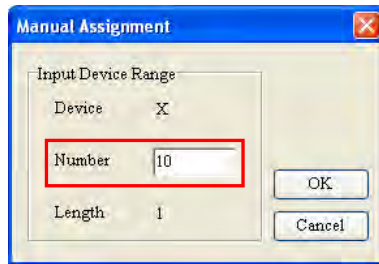


\*. Please refer to section 3.7 for more information about internal parameter settings of modules; for FROM/TO command, please see the instruction manuals for any PLC types.

3

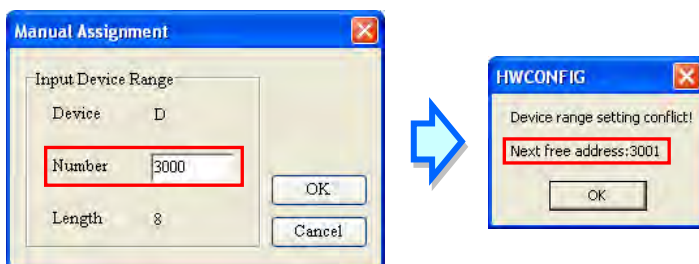
The information in **Input and Output Device Range** is auto-configured by the system, but also user-defined. Users can choose the selected row listed in Information box and then click **...**, or type the starting address to open the setting window and click **OK** once finish input the address.

Slot No.	Label	Firmware ...	Description	Device Range	Output Device Range	Comment
-	AHPS05-5A	-	AH Power Supply Module		None	
-	AHCPU530-EN	1.00	Basic CPU module building with		None	
0	AH16AM10N-5A	-	16 x DI, 24 VDC	X10.0 ~ X10.15		
1	AH16AN01T-5A	-	16 x DO, NPN 12 to 24 VDC		Y0.0 ~	



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	AH16AM10N-5A	-	16 x DI, 24 VDC	X10.0 ~ X10.15		
1	AH16AN01T-5A	-	16 x DO, NPN 12 to 24 VDC		Y0.0 ~ Y0.15	

Whether in **Input and Output Device Range** from **Information** or **corresponding device D** for defining the internal parameters of the module, the input device range cannot overlap. When the setting address is in conflict, the system will auto-correct and suggest another available address based on the current input; for instance, when D3000 device is occupied by another module but is being assigned again, the system will automatically search for another available address from D3000.



When **Input and Output Device Range** is modified and modules need to be added, the default device address will be based on the last modified device address and skip the used ones for assignment; even though the number in the last modified address is not the biggest, the new module address assignment is still based on the last modification.

The following example shows **Input Device Range** is modified to D50~D57 for Slot No. 2.

Slot No.	Label	Firmware ...	Description	Input Device Range	Output Device Range	Comment
-			Module	None	None	
-			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						

3

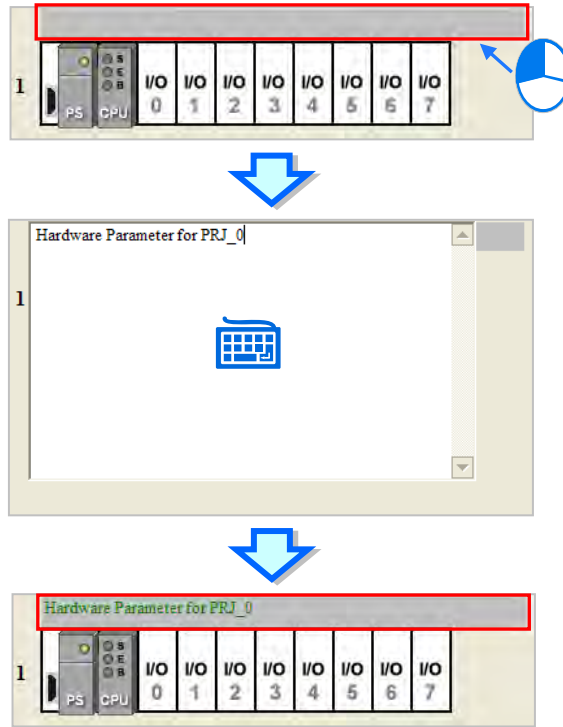
When a new module is added, the **Input Device Range** is automatically assigned from the last modified D50~D57 and not based from the system's biggest D100~D107. As a result, the **Input Device Range** of the module is assigned as D58~D65.

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						

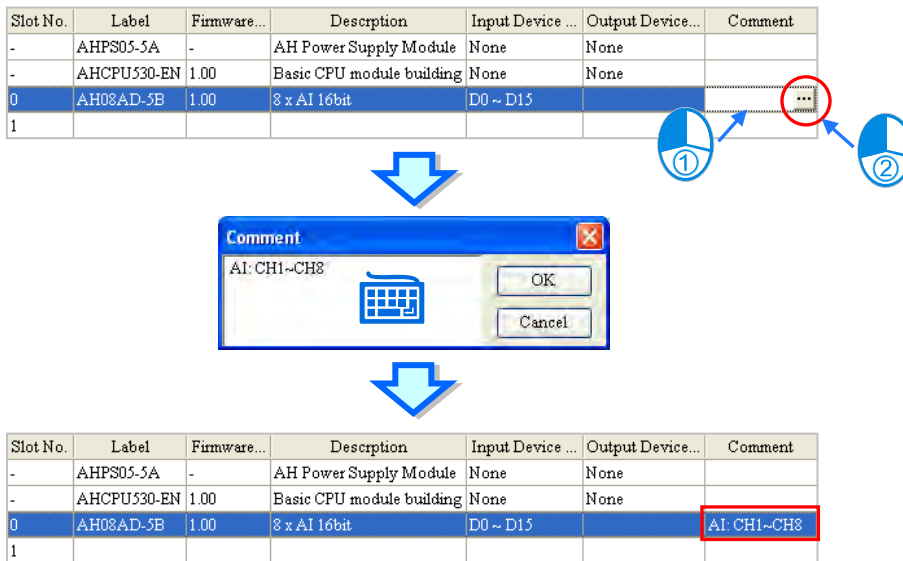
### 3.2.2.3 Edit Comment

Click the gray area on the top of the **System Configuration** area and type comments for the hardware configuration regarding the Project. When typing the comments, press Shift + Enter to start a new line and when complete, click Enter.

3



For AH5x0, AH5x1 series and AH560 redundant series, users can select the **Comment** column and click on the keyboard icon in the Comment window or click to view a pop-up window and input the comment concerning the module.

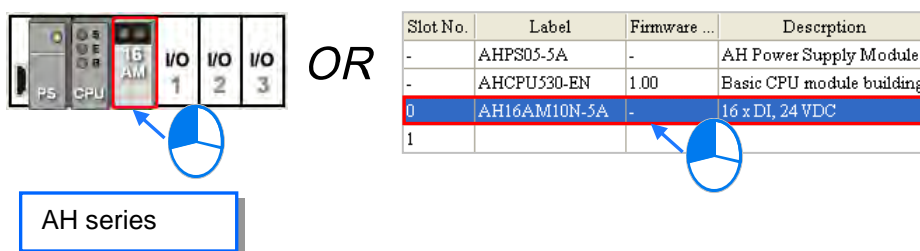


### 3.2.2.4 Delete Module

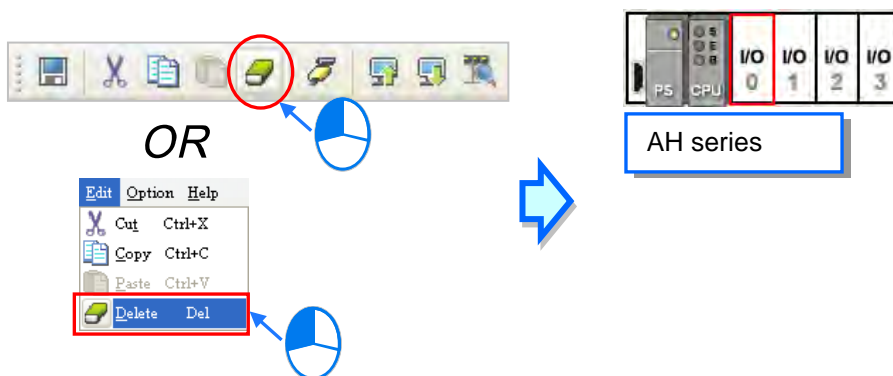
There are two ways to delete a module which has been configured. (The CPU module and AH-series power supply module cannot be deleted.)

● **Method 1**

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

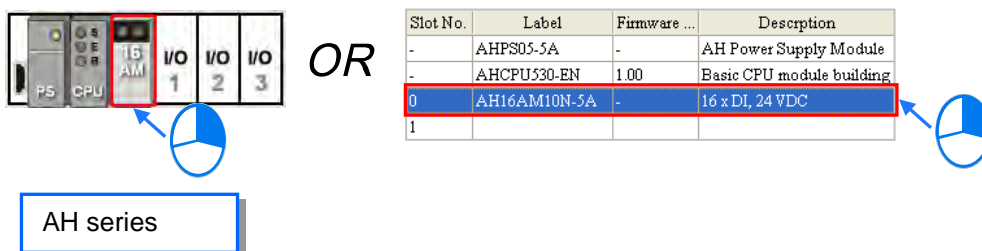


- (2) Click **Delete** on the **Edit** menu, click  on the toolbar, or press Delete on the keyboard.

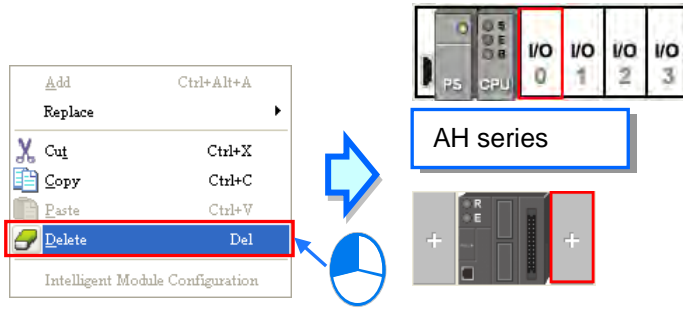


● **Method 2**

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



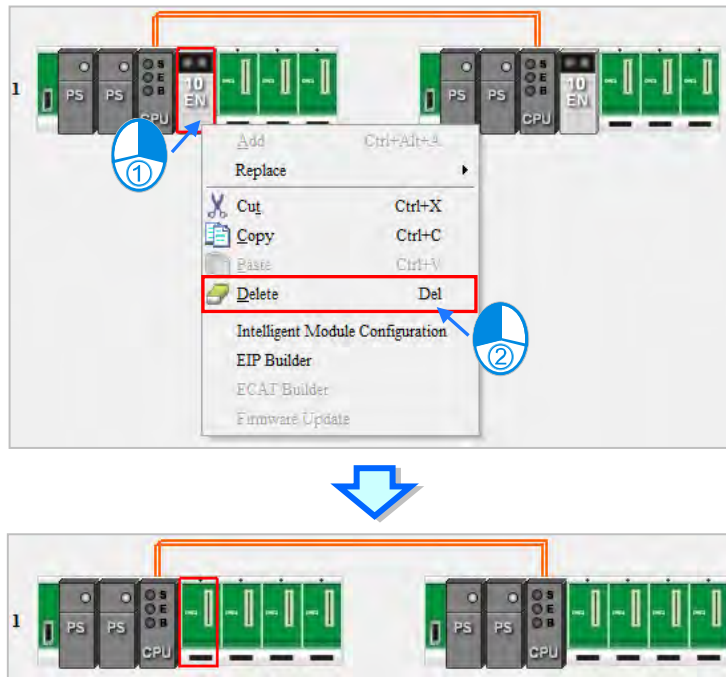
(2) Click **Delete** on the context menu.



**Additional remark**

3

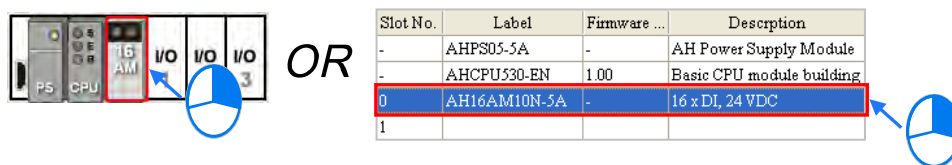
When users select AH560 redundant series and enables **Redundancy** function, the redundant backplane module can be deleted only from the left redundant backplane shown in the **System Configuration** area, while the redundant backplane configuration on the right mirrors the left so it cannot be modified.



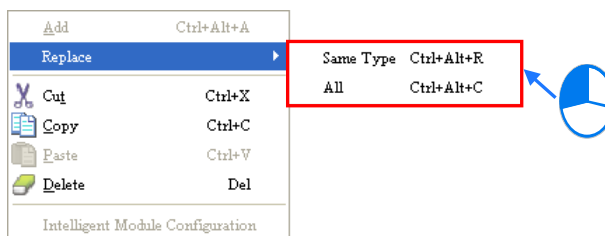
### 3.2.2.5 Replace Module

Users can take the following steps to modify AH series modules in System Configuration area:

- (1) Select and right-click an AH series module for modification in **System Configuration** area or from **Information**. (CPU and power module cannot be modified.)



- (2) Choose **Replace** on the quick menu and two ways are shown in the following.



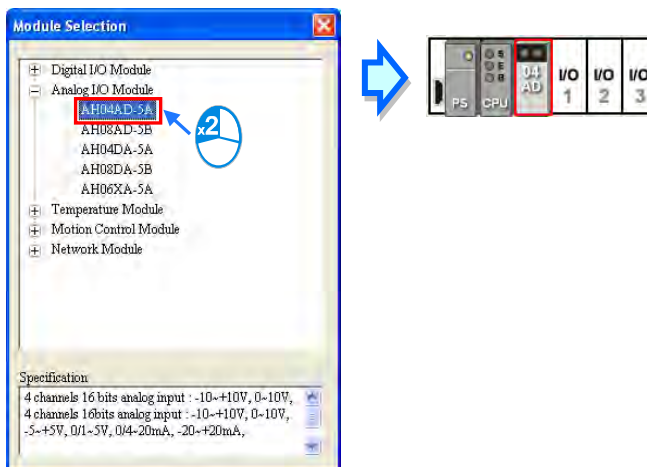
➤ **Same Type**

Replace only the selected module to be the same type module. The new module **Input/Output Device Range** will be the same, while other parameters may return to system defaults if it is not corresponding.

➤ **All**

Replace selected module to be any type of module. The result is similar to deleting the original module by adding a new one, so the new module **Input/Output Device Range** will be re-configured and other parameter settings will also return to system defaults.

- (3) When selecting on methods to Replace, the Module Selection window appears with modules available for the selected Replace method; when the replaced module is decided, you can double-click on that module.

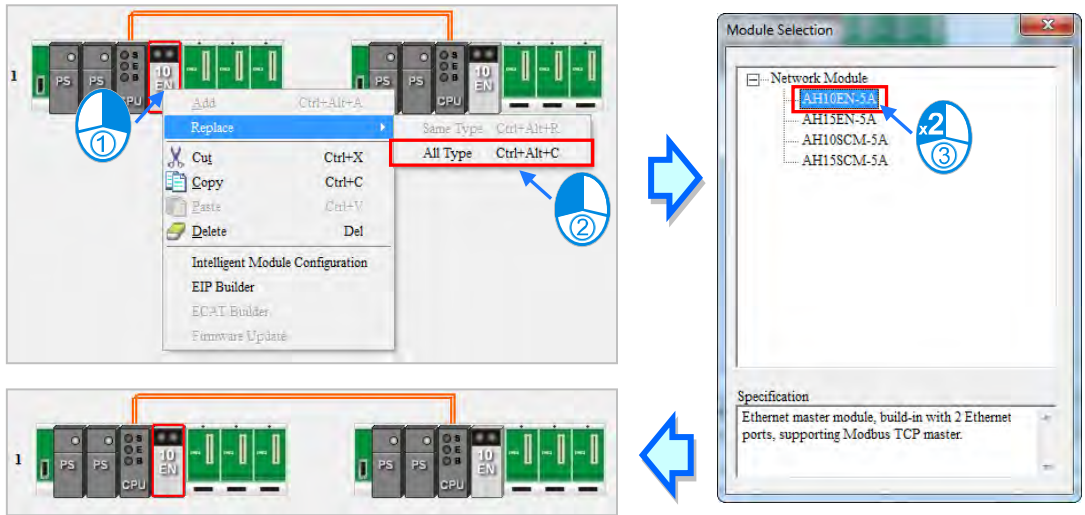




**Additional remark**

When users select AH560 redundant series and enables **Redundancy** function, the redundant backplane module can be deleted only from the left redundant backplane shown in the **System Configuration** area, while the redundant backplane configuration on the right mirrors the left so it cannot be modified.

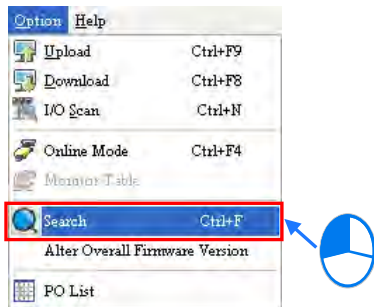
3



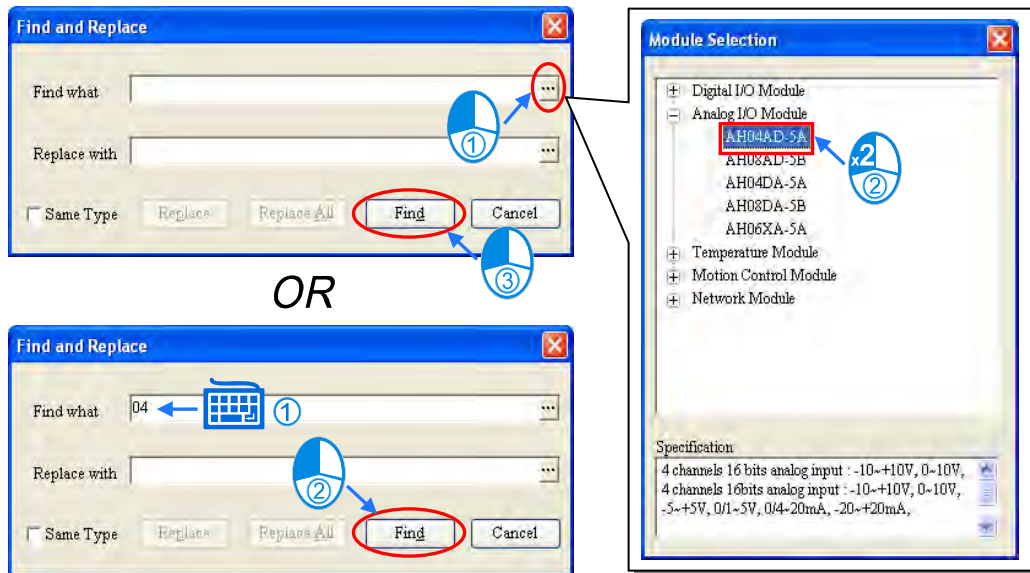
**3.2.2.6 Search/Find and Replace**

Users can take the following steps to search or find and replace AH series modules in System Configuration area:

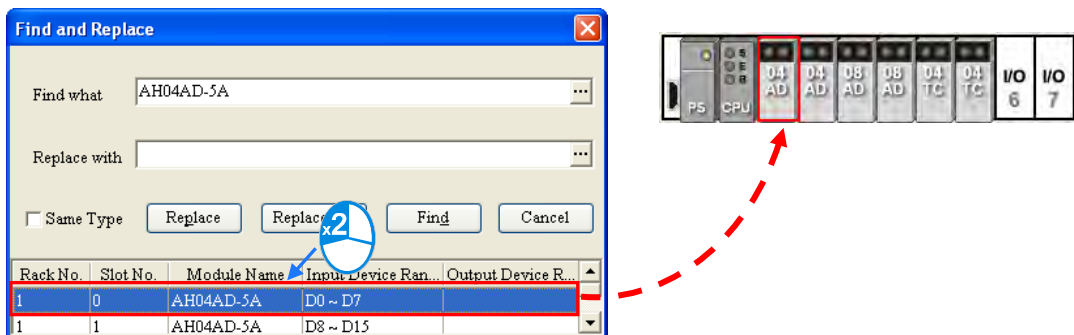
Click **Search** from the **Option** menu in the HWCONFIG toolbar and an operating window appears.



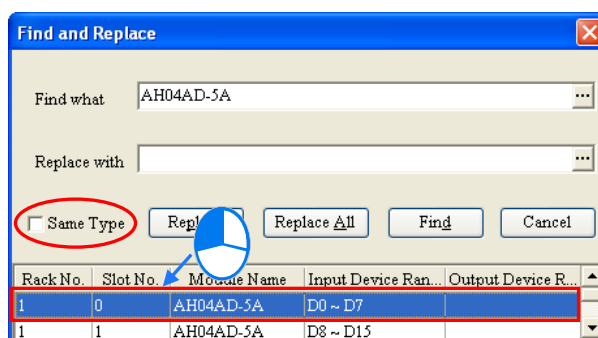
- (1) Click **...** in the **Find and Replace** box to search the selected module type from the **Module Selection** window and double-click on the module. Or input keywords regarding the module type in the Find What box and click **Find**.



- (2) When finding is complete, a list of modules that meet the search condition is shown below in the Find and Replace window; when double-click a selected item from the list, that module is also selected in the System Configuration area.



- (3) To replace a module, please first select the item to be replaced from the list in the Find and Replace window (see below) and select the checkbox if to replace as the Same Type.




➤ **Same Type Selected**

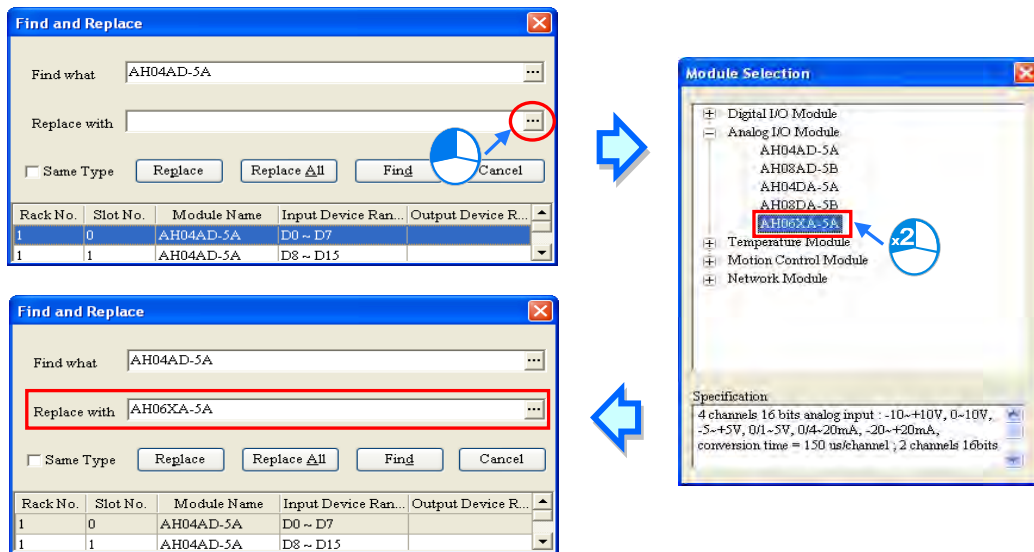
A module selected is replaced by a same type of module. After the module is replaced, the input/output devices assigned to the new module are the same as the input/output devices assigned to the module replaced. Besides, if the parameters in the new module are not the same as the parameters in the module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

➤ **Same Type 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.

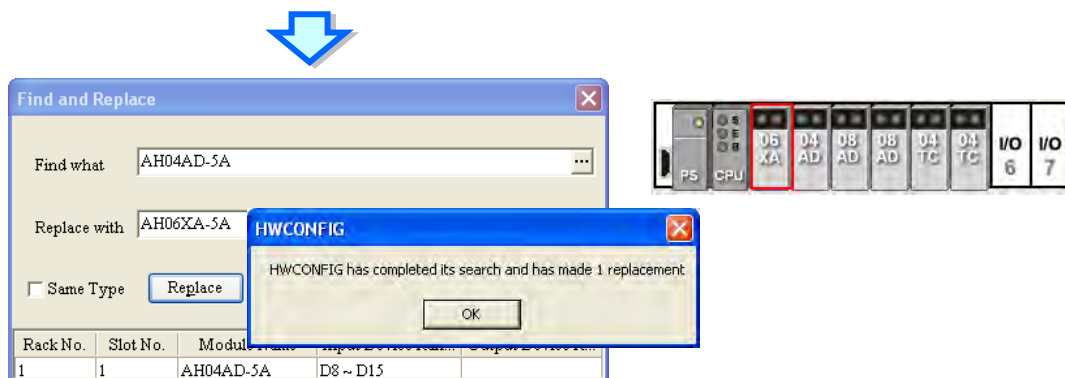
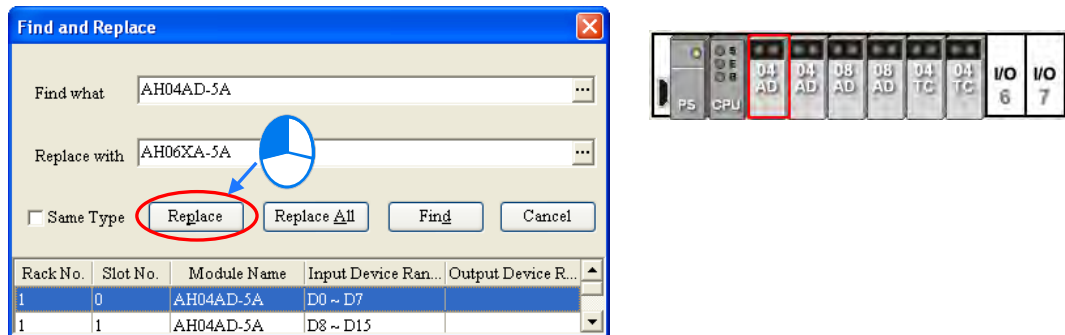
3

- (4) Click  in the **Replace with** box, select a module in the **Module Selection** window, and double-click the module. Owing 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.

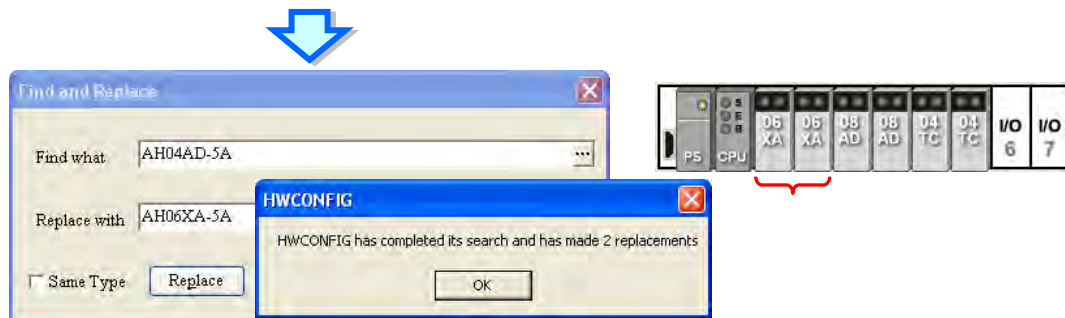
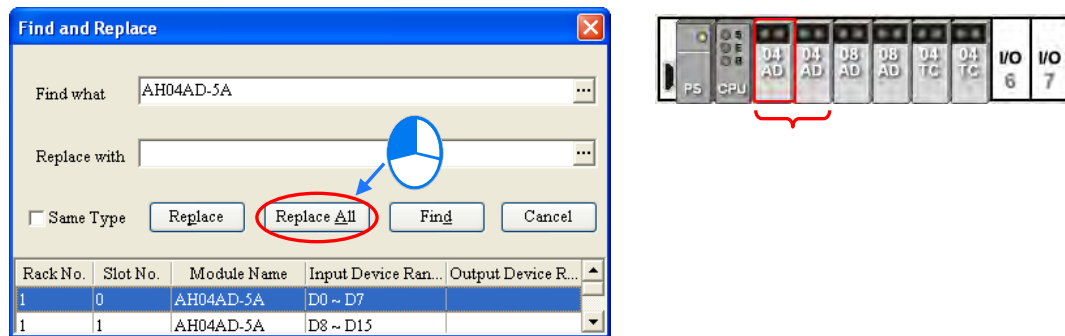


\*. If the **Replace With** box is blank, the selected module is deleted once **Replace** function is executed.

- (5) When Find and Replace setting is complete, users can click Replace to replace the module selected with the new module. The search list is updated once Replace is complete and will automatically select the next item.



- (6) After filling in the setups in the Find and Replace window, users can click **Replace All** and new modules based on the setting will replace the listed items shown in the window.



\* When using AH560 redundant system with Redundancy function enabled, only the main redundant backplane on the left in System Configuration area can perform and replace modules, while the main redundant backplane on the right is configured to mirror the left and cannot be modified. To execute Find and Replace function, the configuration of the main redundant backplane on the right is not on the search list.

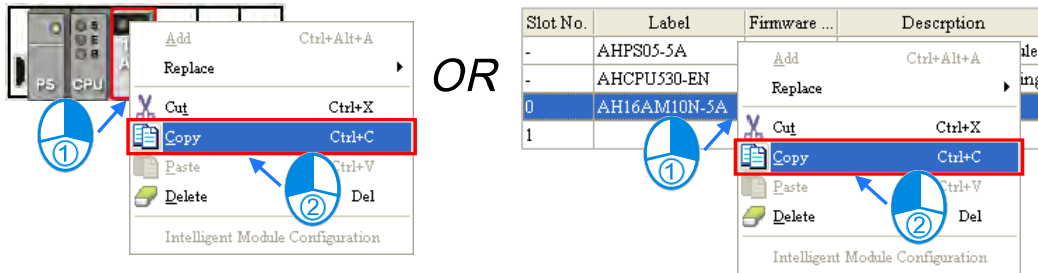
### 3.2.2.7 Copy/Paste

There are two ways to copy modules (CPU and power modules are not to copy/paste):

● **Method 1**

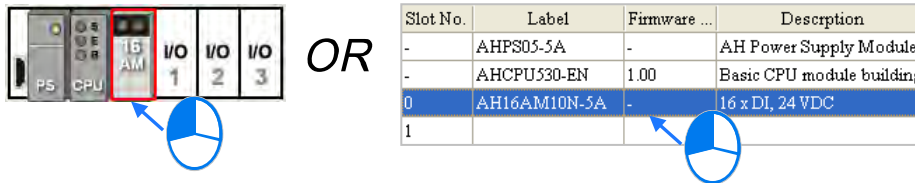
Right-click a selected module to copy in the **system configuration area** or **Information**, and choose **Copy** from the context menu.


3

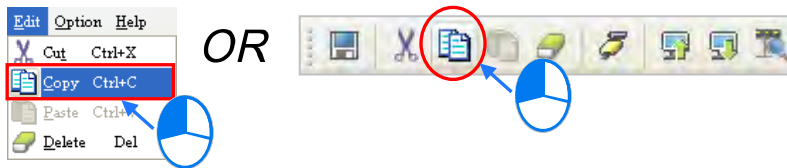


● **Method 2**

(1) Select a module to copy from **System Configuration** area or **Information**.



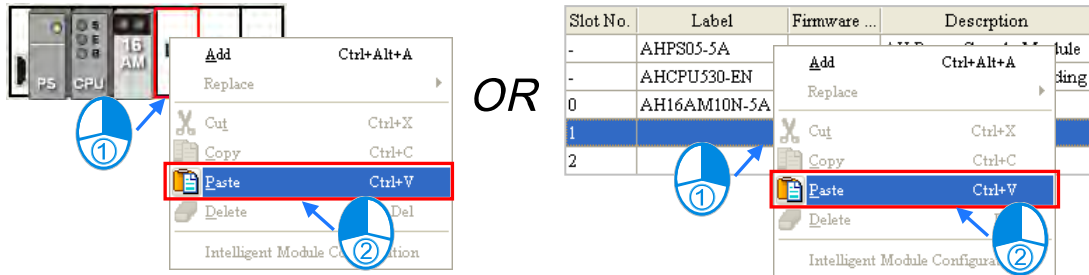
(2) Click **Copy** from the **Edit** menu, or  on the toolbar.



There are two ways to paste modules:

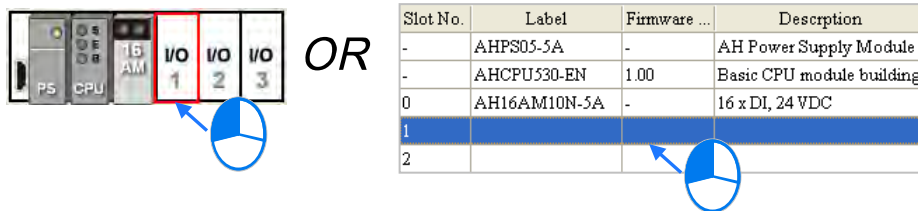
● **Method 1**

Right-click a slot to paste from **System Configuration** area or **Information** and choose **Paste** from the quick menu.



● **Method 2**

(1) Select a slot to paste from the **System Configuration** area or **Information**.



(2) Choose **Paste** from **Edit** in the HWCONFIG toolbar, or click  on the toolbar.



**Additional remark**

To copy/paste a module, users can view the following descriptions on module parameters.

- **Input/Output device range:** The paste module automatically skips to the available device range number.
- **Comment:** Comments are copied to the paste module.
- **Internal parameters:** Parameters are copied the paste module.


- **Corresponding device D:** An assigned column of the **Corresponding device D** for internal PLC paste module, which automatically skips to the available device range number.
- **Advanced parameters for intelligent module:** The parameters cannot be copied and is referred to as system defaults. Users will need to reset the parameters.

The following example shows that slot 0 is the copy module, while slot 1 is the paste module.

3

Description	Address	Monitor	Initial
CH0 Average Time	D3000	<input checked="" type="checkbox"/>	30
CH1 Average Time		<input type="checkbox"/>	10

Description	Address	Monitor	Initial
CH0 Average Time	D3001	<input checked="" type="checkbox"/>	30
CH1 Average Time		<input type="checkbox"/>	10



Slot No.	Label	Firmware ...	Description	Input Device ...	Output Device ...	Comment
-	AHPS03-5A	-	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						

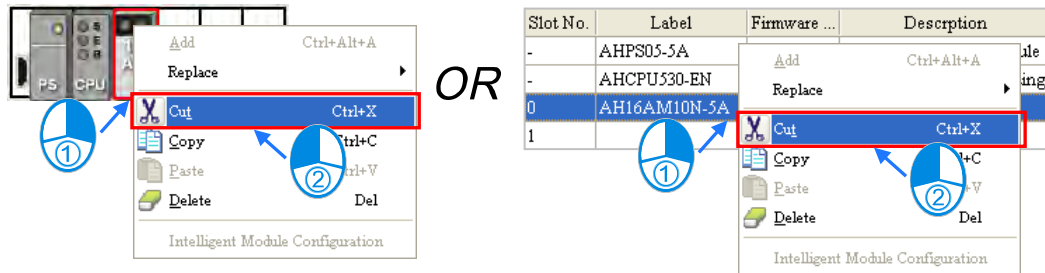
When using AH560 redundant system with Redundancy function enabled, only the main redundant backplane on the left in System Configuration area can copy/paste modules, while the main redundant backplane on the right is configured to mirror the left and cannot be modified.

### 3.2.2.8 Cut/Paste

There are two ways to cut modules (CPU and power modules are not to cut or paste):

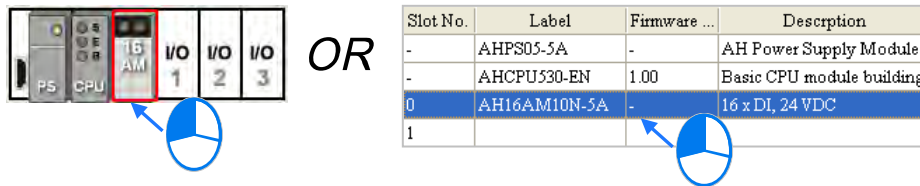
● **Method 1**


Right-click a selected module to cut in the **system configuration area** or **Information**, and choose **Cut** from the context menu.

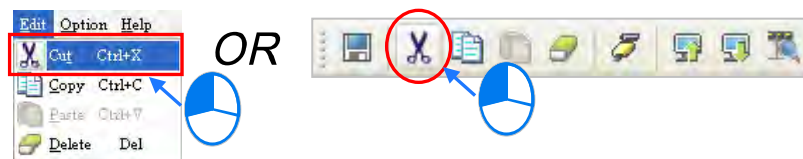


● **Method 2**

(1) Select a module to cut from **System Configuration area** or **Information**.



(2) Choose **Cut** from the **Edit** menu, or click  on the toolbar.



3

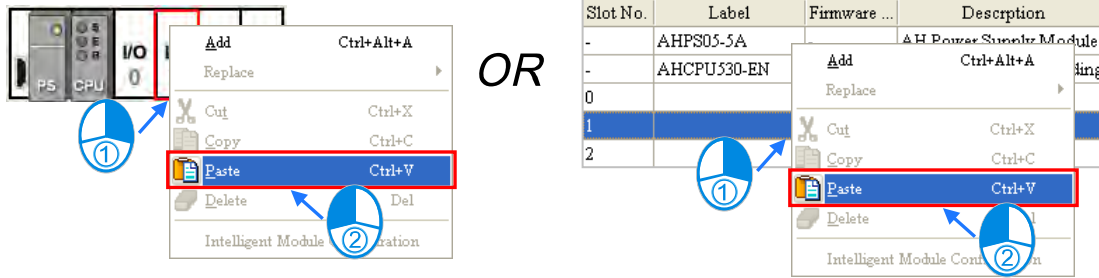


There are two ways to paste modules:

● **Method 1**

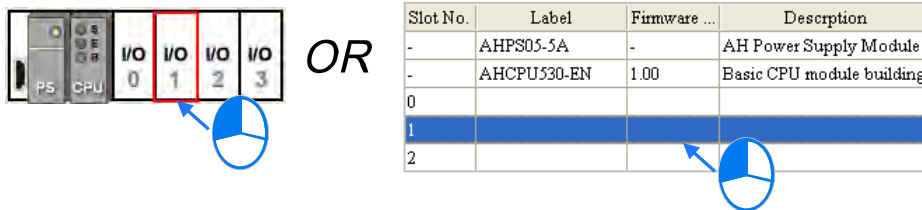
Right-click on the selected slot to paste from **System Configuration** area or **Information**, and choose **Paste** in the context menu.


3

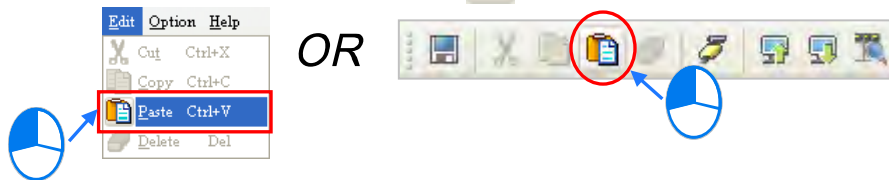


● **Method 2**

(1) Select a slot to paste from **System Configuration** area or **Information**.



(2) Choose **Paste** from the **Edit** menu, or click  on the toolbar.



**Additional remark**

When executing **Cut/Paste** function, modules that are cut can only be paste once. However, the **input/output device range, comment, internal parameters, corresponding device D and advanced parameters for intelligent modules** from modules that are cut is altogether transferred to the paste module.

When using AH560 redundant system with Redundancy function enabled, only the main redundant backplane on the left in System Configuration area can cut/paste modules, while the main redundant backplane on the right is configured to mirror the left and cannot be modified.

### 3.2.2.9 Drag and Drop

Besides CPU and power modules, all module graphics in System Configuration area can be dragged by the mouse .

For AH series, there are two dragging and dropping conditions:

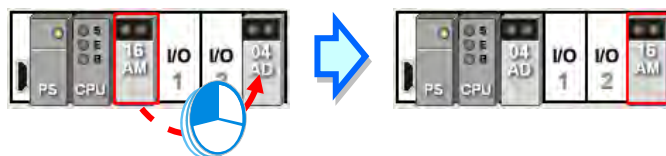
- **Condition 1**

Select a module to drag and drop onto a vacant slot, then the will be moved to a new position.



- **Condition 2**

Drag a selected module to a position occupied by another module, the position of the two modules are changed.



#### **Additional remark**

When using AH560 redundant system with Redundancy function enabled, only the main redundant backplane on the left in System Configuration area can drag and drop modules, while the main redundant backplane on the right is configured to mirror the left and cannot be modified.

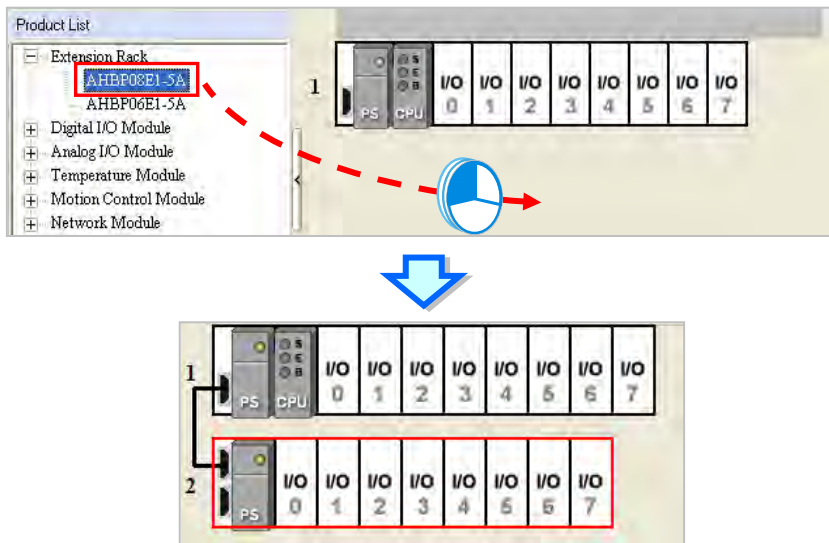
### 3.2.2.10 Add Expansion Backplane and Redundant Expansion Backplane

For AH series, there are two ways to add expansion backplanes:

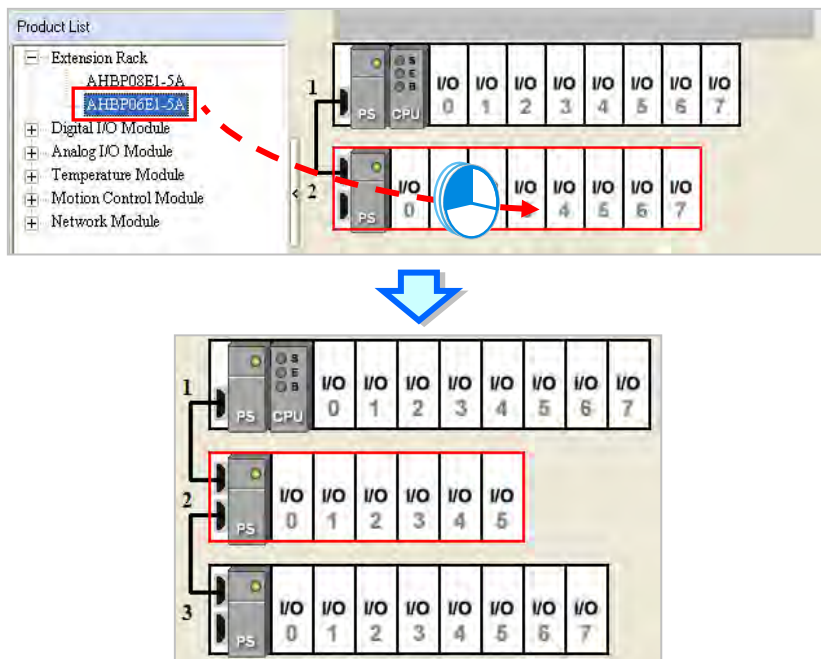
- **Method 1**

Choose a backplane model from the Product List and drag to **System Configuration** area; when the backplane is dropped onto the blank workspace area, the new backplane is placed underneath the present one.

3

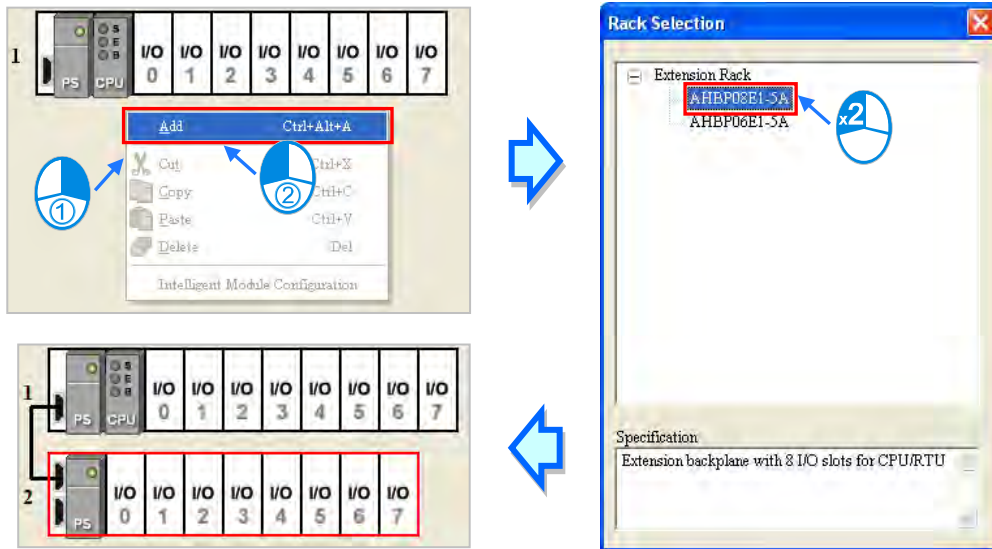


When the backplane is dropped onto an existed backplane, the new backplane is placed on top of the present one.

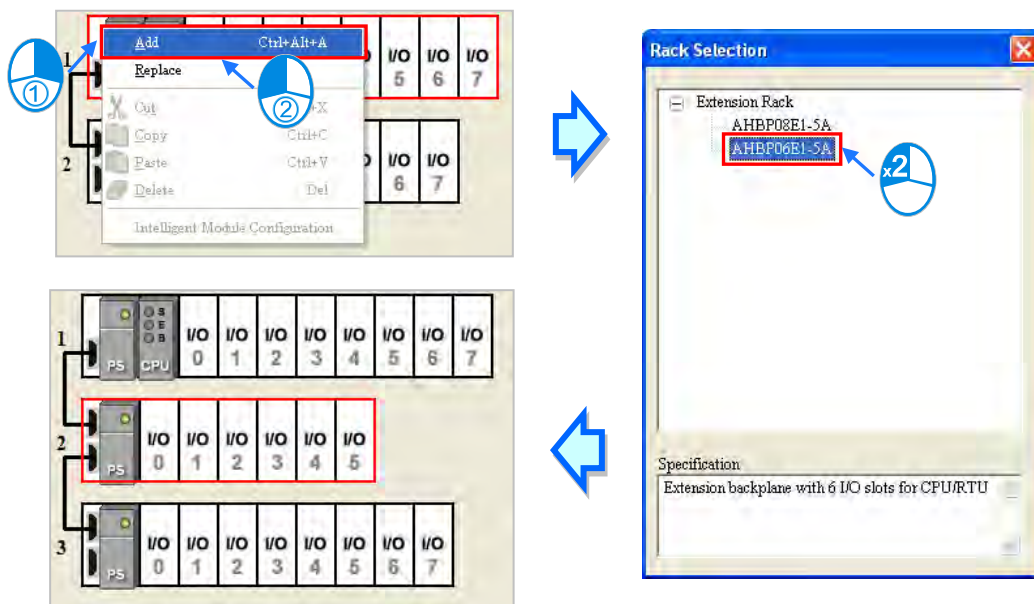


● **Method 2**

Right-click the blank in the system configuration area and click **Add** from the context menu, then double-click on the selected backplane. The new backplane is placed underneath the selected backplane.



Right-click the left side of the selected backplane graphic, choose **Add** from the context menu and double-click on the backplane to add. The new backplane is placed underneath the selected backplane.

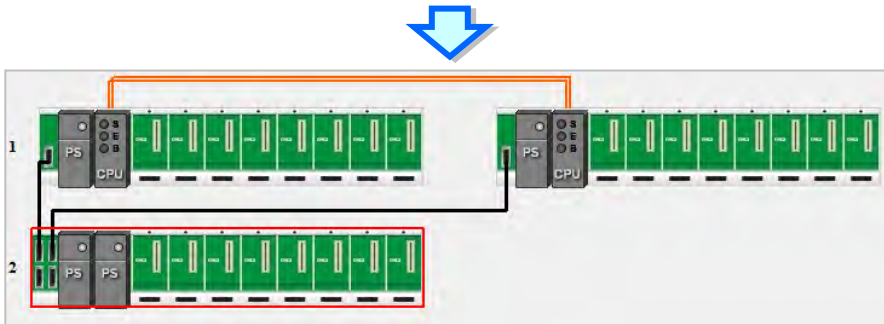
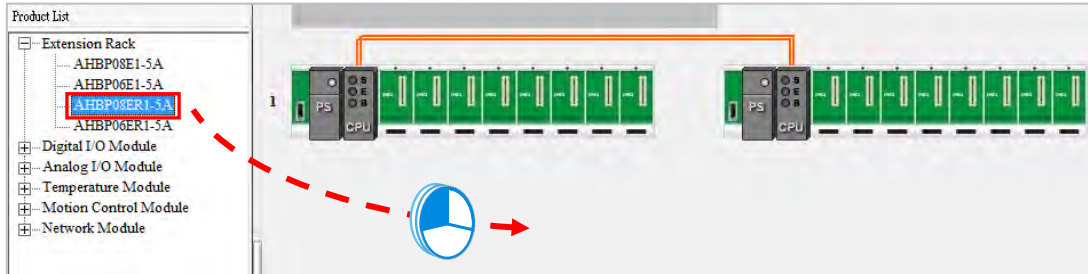


To add redundant expansion backplane for AH5x0 and AH5x1 series, redundant expansion backplane and expansion backplane cannot exist at the same time. For AH560 redundant system with Redundancy function enabled, please refer to related product manuals on adding redundant expansion backplane. Here are two ways to add expansion backplanes:

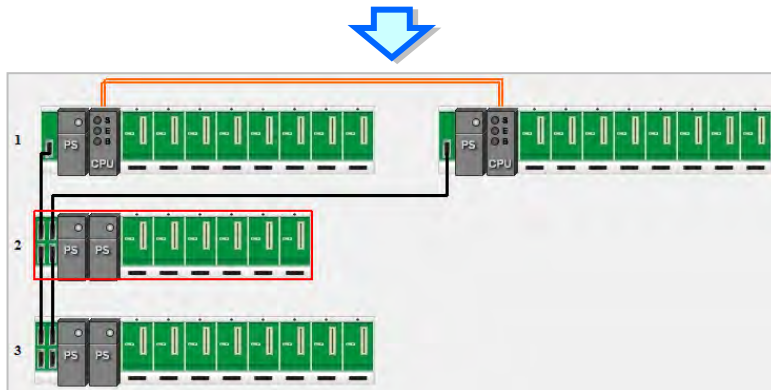
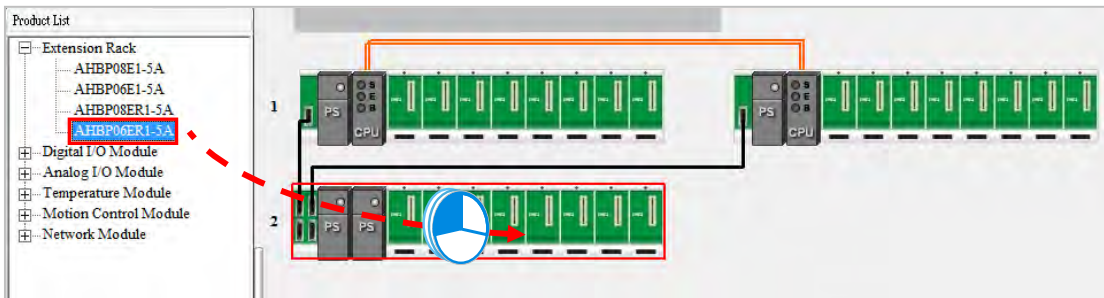
● Method 1

Choose a redundant expansion backplane model from the Product List and drag to **System Configuration** area; when the redundant expansion backplane is dropped onto the blank workspace area, the new backplane is placed underneath the present one.

3



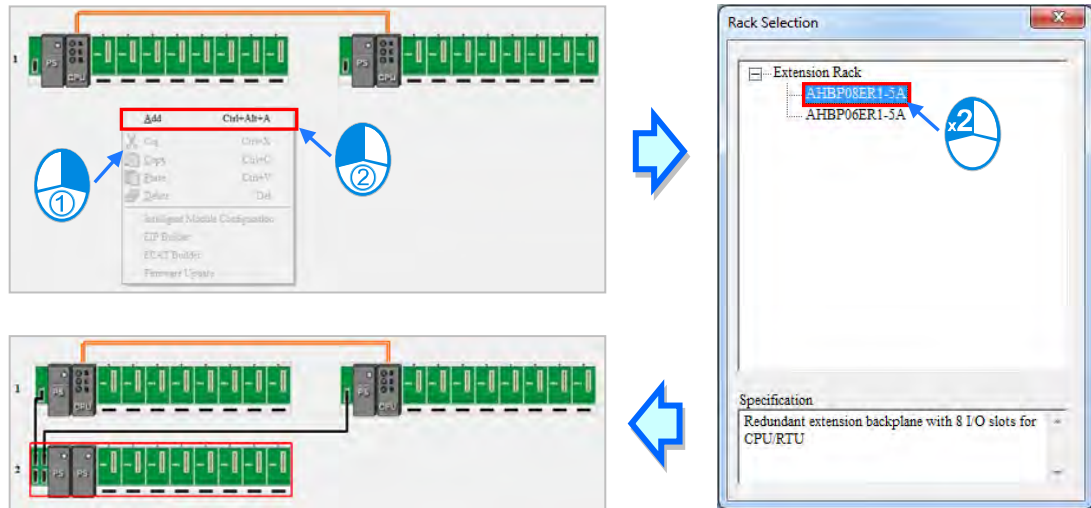
When the redundant expansion backplane is dragged to an existed redundant expansion backplane, the new redundant expansion backplane is placed on top of the existed one.



\* A redundant expansion backplane cannot be dropped on top of the main backplane and redundant backplane nor exist along with expansion backplanes.

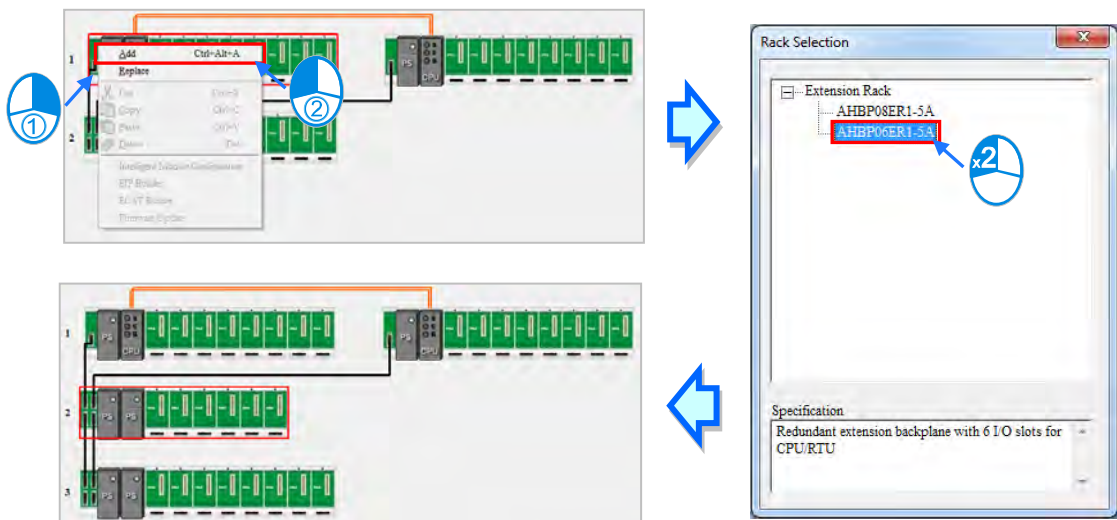
● **Method 2**

Right-click the blank in the system configuration area and click **Add** from the context menu, then double-click on the selected redundant expansion backplane. The new redundant expansion backplane is placed underneath the selected backplane.



3

Right-click the left side of the selected backplane graphic, choose **Add** from the context menu and double-click on the redundant expansion backplane (rack) to add. The new backplane is placed underneath the selected backplane.



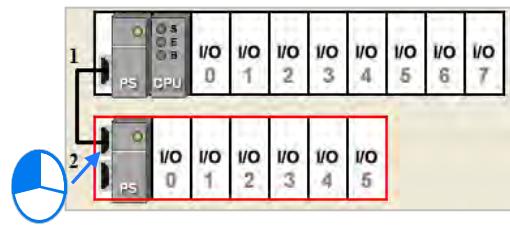
### 3.2.2.11 Delete Expansion Backplane and Redundant Expansion Backplane

There are two ways to delete expansion backplanes (the main backplane cannot be deleted):

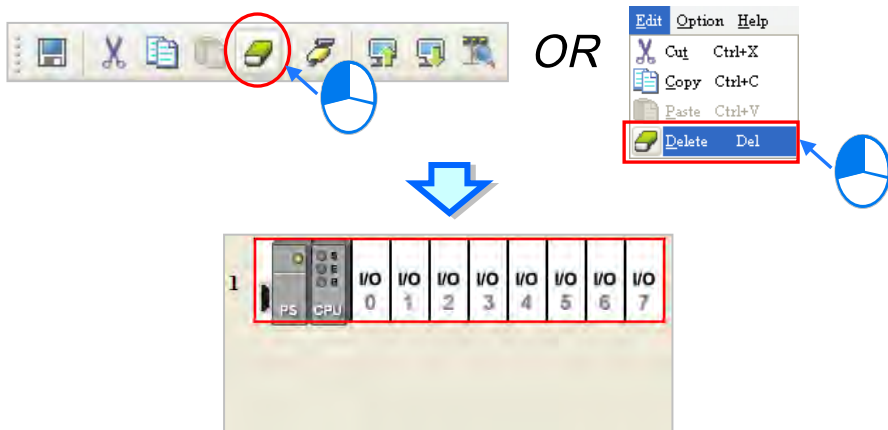
● **Method 1**

(1) Click on the left side of the selected backplane graphic to be deleted.

3

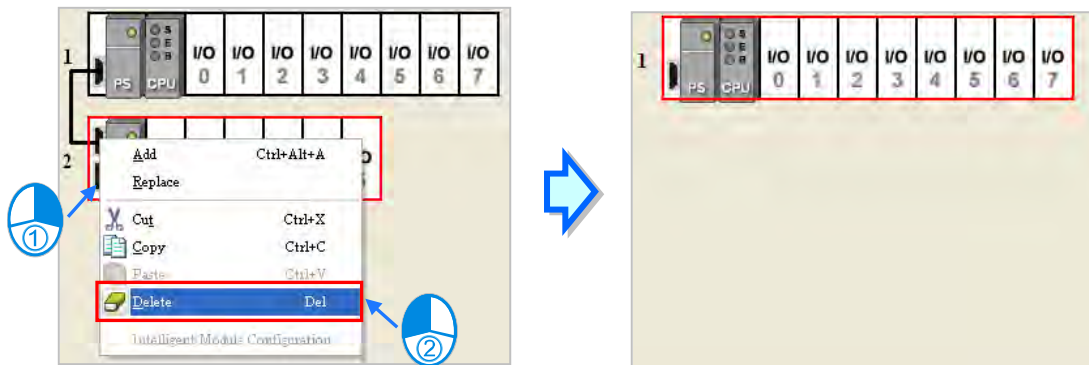


(2) Select **Delete** from the **Edit** menu or click  on the toolbar. Users can press Delete on the keyboard too.



● **Method 2**

Right-click the left side of a backplane and choose **Delete** from the context menu.

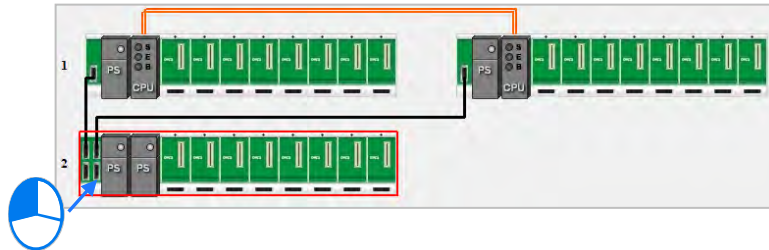



\*The AHxxEMC series does NOT support expansion backplane function.

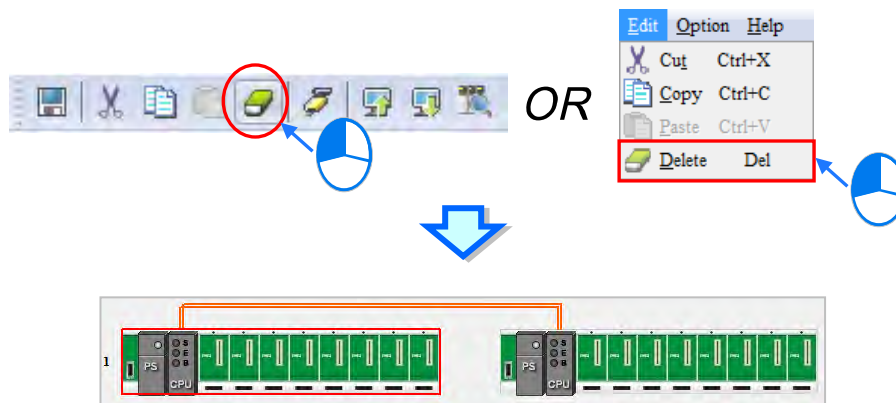
There are two ways to delete redundant expansion backplanes (redundant main backplane cannot be deleted):

● **Method 1**

(1) Click on the left side of the selected redundant expansion backplane graphic to be deleted.



(2) Choose **Delete** from the **Edit** menu in HWCONFIG functions, or click  from the toolbar. Users can press Delete on the keyboard too.



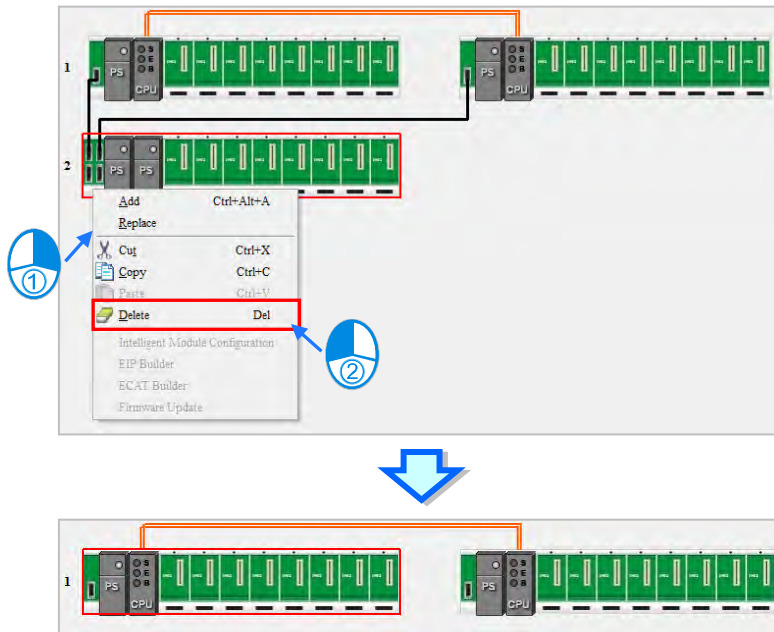
3



● **Method 2**

Right-click on the left side of a redundant expansion backplane to be deleted and choose **Delete** from the context menu.

3

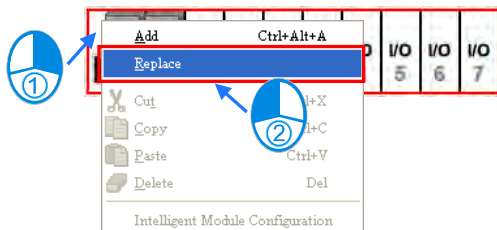


\*The AHxxEMC series does NOT support expansion backplane function.

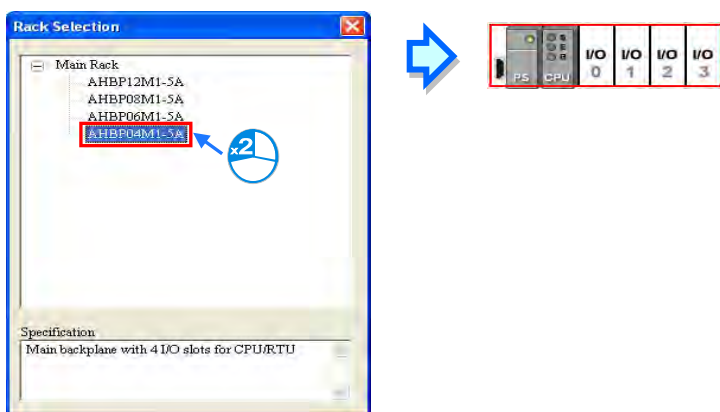
### 3.2.2.12 Replace Backplanes

To replace backplanes, users can follow these steps:

- (1) Right-click on the left side of a backplane to be replaced and choose **Replace** from the context menu.

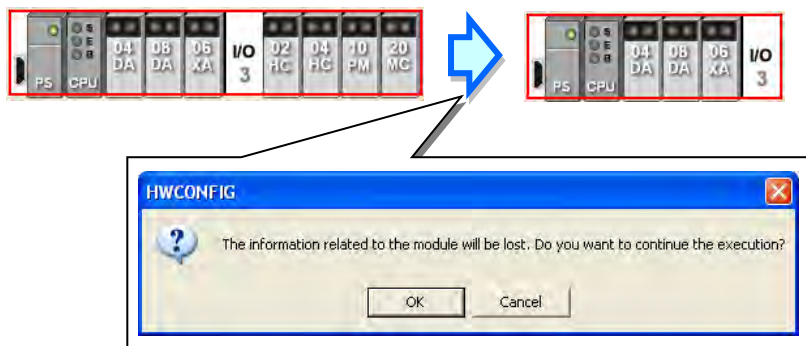


- (2) Double-click on the selected rack (backplane model) to be replaced in the window.



#### Additional remark

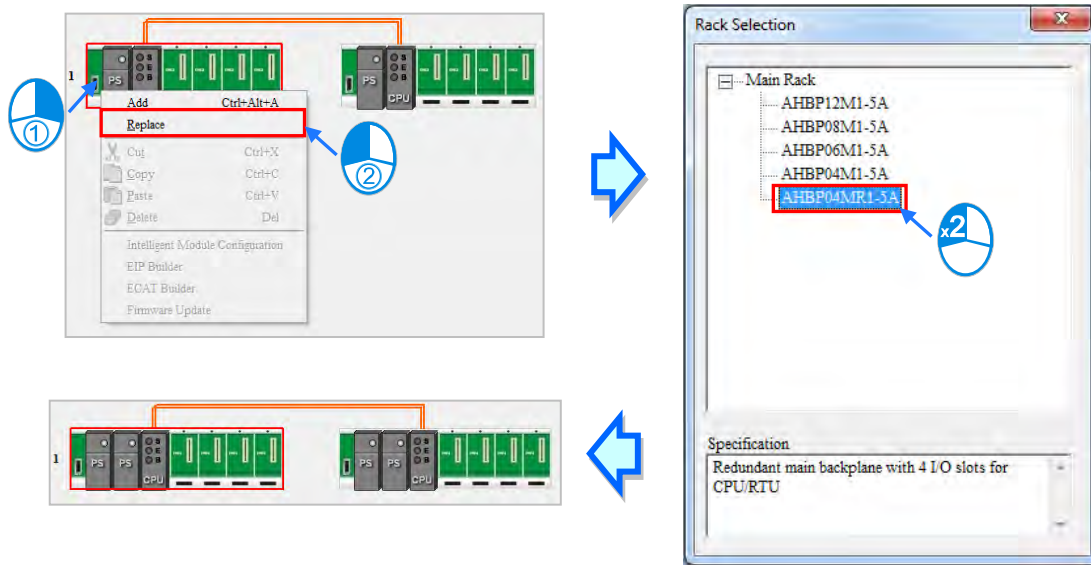
When the slot number of the new backplane is smaller than the original backplane, please be take note that once replace is executed, modules in the original backplane with slot number exceeding the new backplane is automatically deleted.



3

When using AH560 redundant system with Redundancy function enabled, only the main redundant backplane on the left can be replaced, while the main redundant backplane on the right is configured to mirror the left and cannot be modified.

3

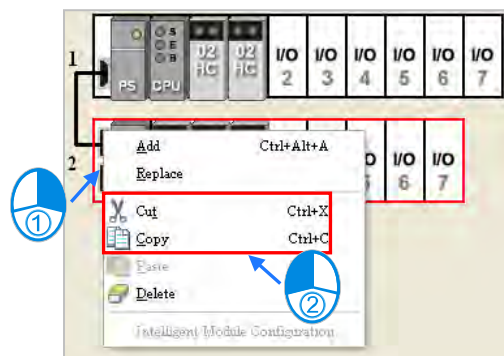


### 3.2.2.13 Cut/Copy/Paste an Expansion Backplane and Redundant Expansion Rack

There are two ways to copy or cut an expansion backplane (the main backplane cannot copy/cut/paste):

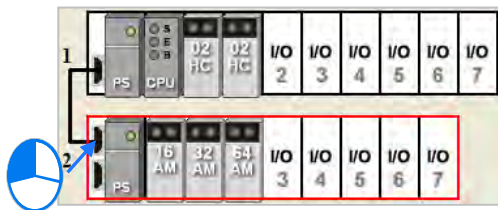
- **Method 1**

Right-click the left side of a backplane to copy or cut, and then click **Copy** or **Cut** on the context menu.

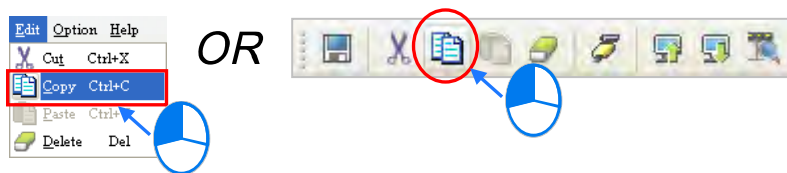


● **Method 2**

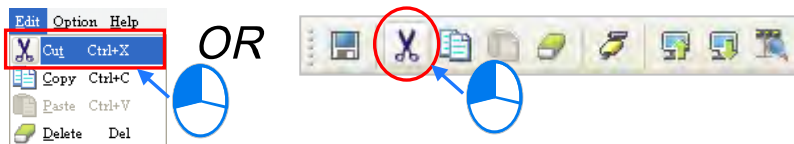
Click the left side of the selected backplane to cut or copy.



To copy, users can click **Copy** from the **Edit** menu, or click  from the toolbar.



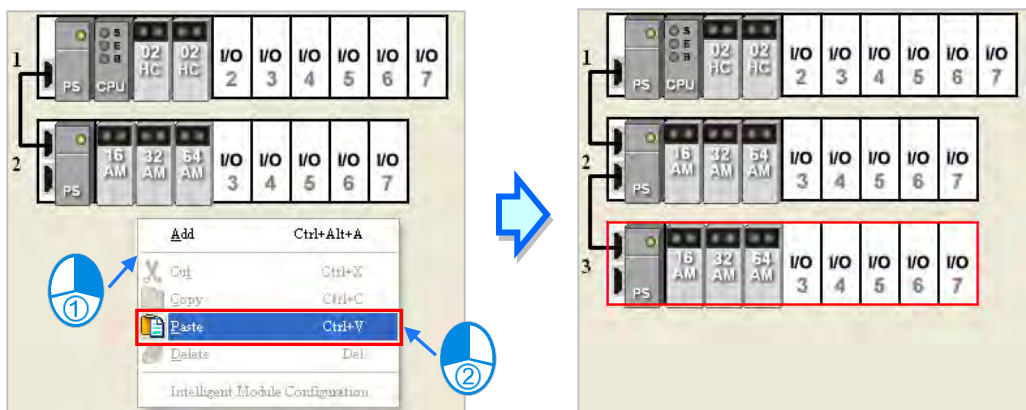
To cut, users can click **Cut** from the **Edit** menu, or click  from the toolbar.



There are two ways to paste an expansion backplane. If an expansion backplane is pasted on the blank in the system configuration area, it will be under the present racks. If an expansion backplane is pasted on a present expansion backplane, it will be put above the present expansion backplane.

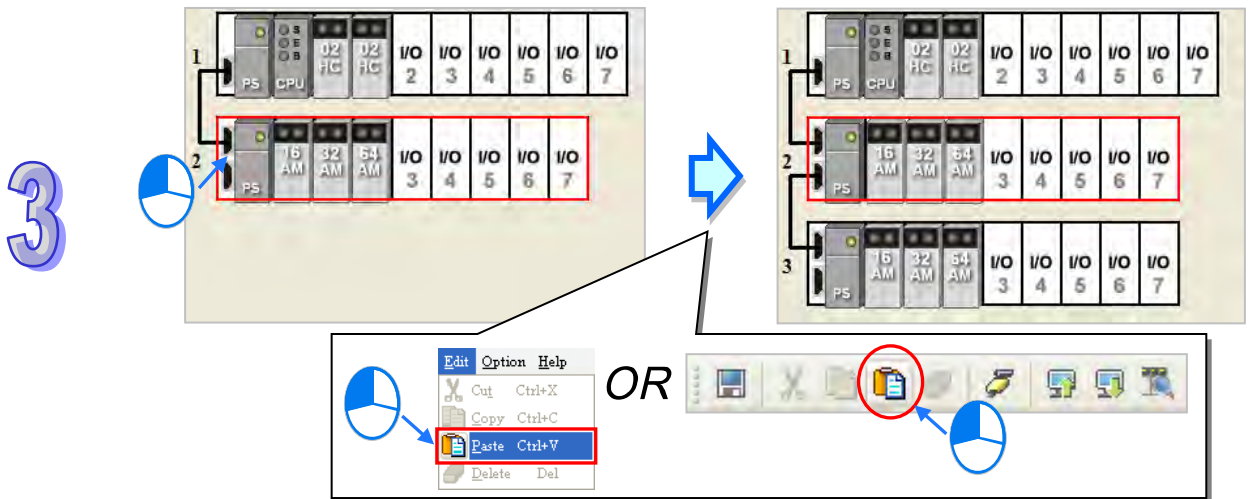
● **Method 1**

Right-click on a selected position to paste in **System Configuration** area and then select **Paste** on the context menu.



● **Method 2**

Click the selected position in **System Configuration** area, and click **Paste** on the **Edit** menu, or  on the toolbar.

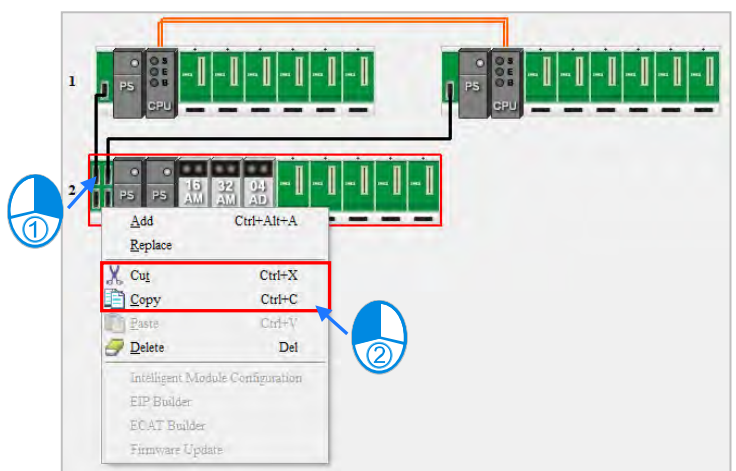


\*The AHxxEMC series does NOT support expansion backplane function.

There are two ways to **Copy** or **Cut** a redundant expansion backplane (the main backplane cannot copy/cut/paste)

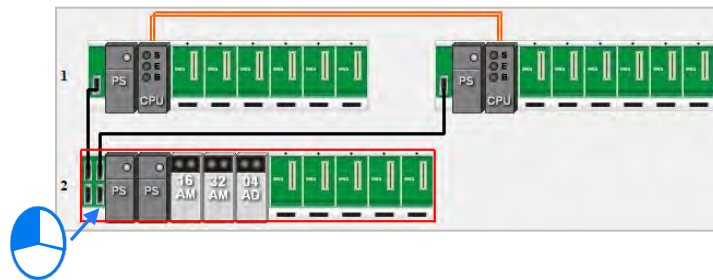
● **Method 1**

Right-click the left side of a redundant expansion backplane to copy or cut and then click **Copy** or **Cut** from the context menu.



● **Method 2**

Click on the left side of the selected backplane to copy or cut.



To copy, users can click **Copy** on the **Edit** menu, or click  on the toolbar.



To cut, users can click **Cut** on the **Edit** menu, or click  on the toolbar.



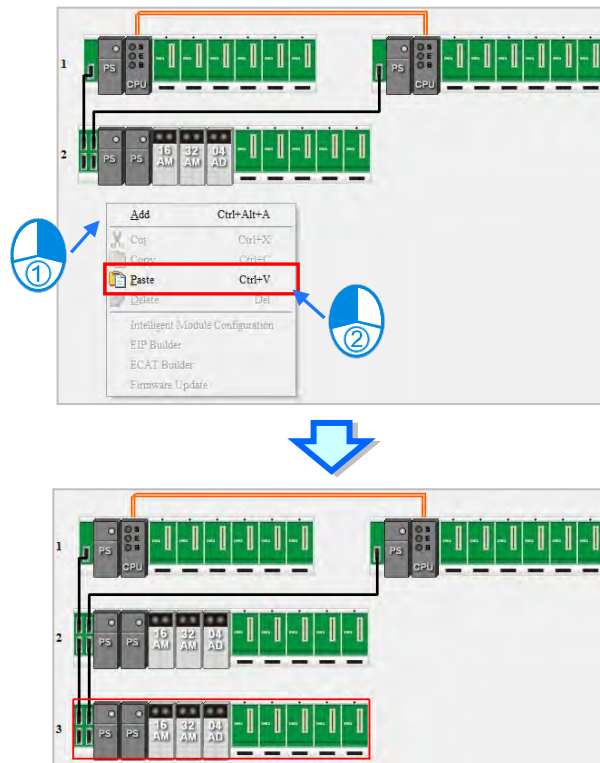
3

There are two ways to paste the redundant expansion backplane. If the redundant expansion backplane is pasted on the selected blank space in **System Configuration** area, it is placed under the present backplane. If the redundant expansion backplane is to paste onto an existed redundant expansion backplane, it is placed on top of the present redundant expansion backplane.

● **Method 1**

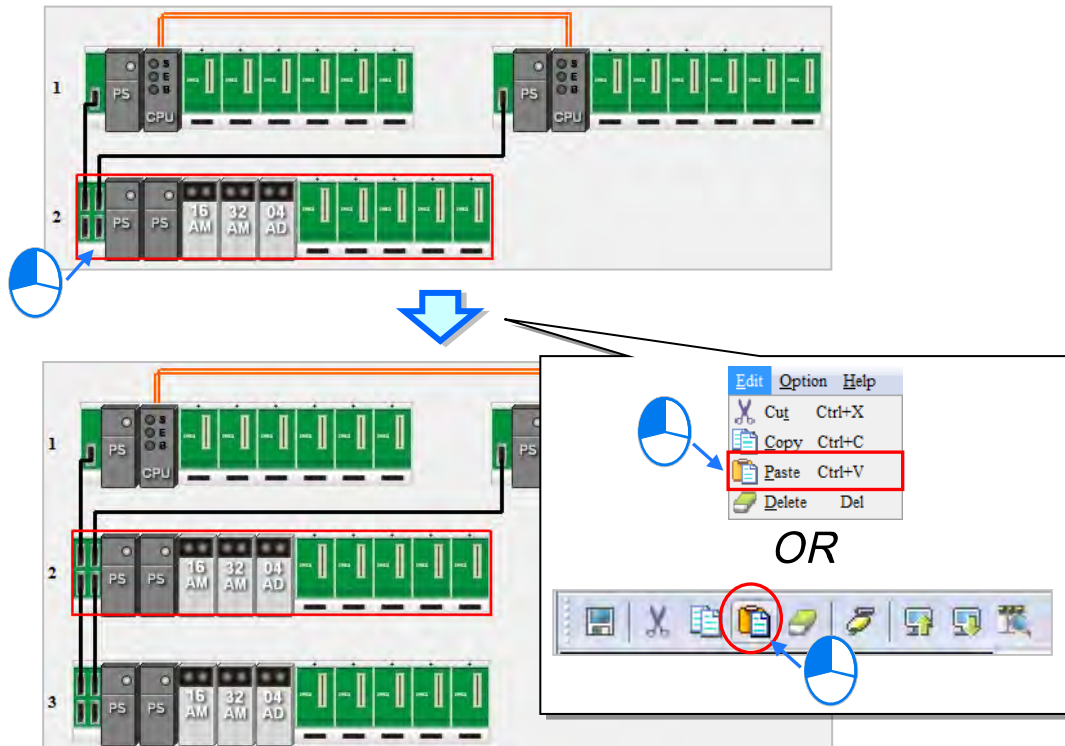
Right-click to select a place to paste in **System Configuration** area, and then click **Paste** on the context menu.

3



● **Method 2**

Click to select the place to paste and then click **Paste** from the **Edit** menu, or  on the toolbar.



\*The AHxxEMC series does NOT support redundant expansion backplane function.

**Additional remark**

When an extension or redundant type expansion backplane is cut/pasted, the parameters of the modules are processed as follows.

- **Input/output device range:** The input/output devices assigned to the modules on the extension or redundant type expansion backplane cut are automatically assigned to the modules on the or redundant type expansion backplane pasted.
- **Comments about the modules:** The comments about the modules on the extension or redundant type expansion backplane cut are copied into the modules on the extension or redundant type expansion backplane pasted.
- **Parameters in the modules:** The parameters in the modules on the extension or redundant type expansion backplane cut are copied into the modules on the extension or redundant type expansion backplane pasted.



- **Data registers:** The data registers assigned to the modules on the extension or redundant type expansion backplane cut are automatically assigned to the modules on the extension or redundant type expansion backplane pasted.
- **Parameters in the intelligent modules:** The setting of the parameters in the intelligent modules is pasted.

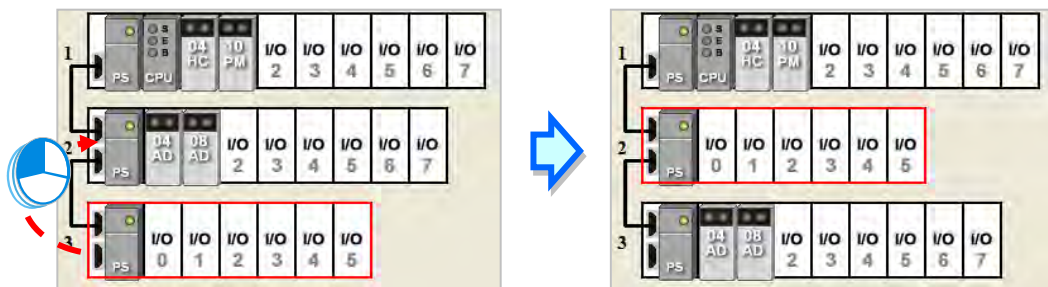
When an extension or redundant type expansion backplane is copied/pasted, the parameters in the module are processed as follows.

3

- **Input/Output device range:** The input/output devices assigned to the module on the extension or redundant type expansion backplane copied will automatically switch number to the module on the extension or redundant type copied.
- **Comment about the module:** The comment about the module 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 will automatically switch number to the module copied.
- **Parameters in the intelligent module:** The setting of the parameters in the intelligent module is not copied, and is set to default value.

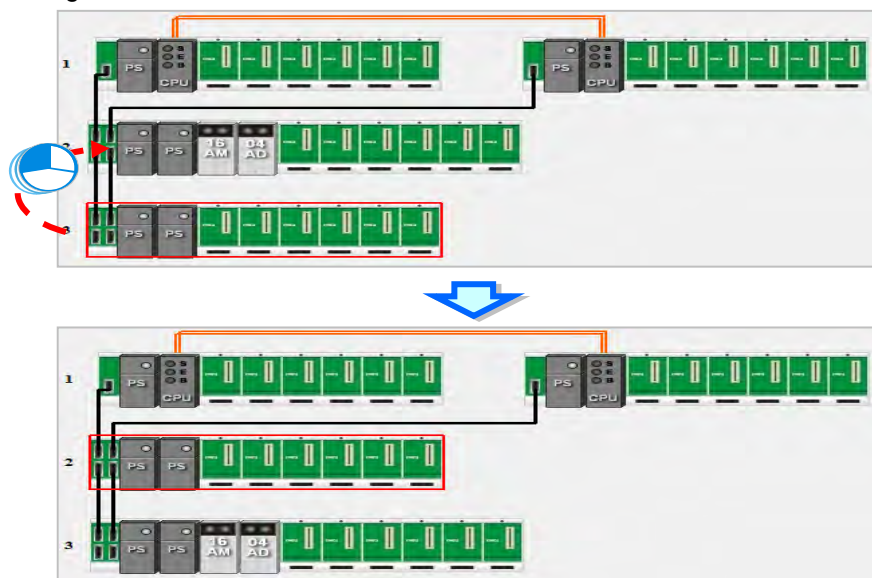
### 3.2.2.14 Drag an Expansion or Redundant Expansion Backplane

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



\*The AHxxEMC series does NOT support expansion backplane function.

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



\*The AHxxEMC series does NOT support expansion backplane function.

3

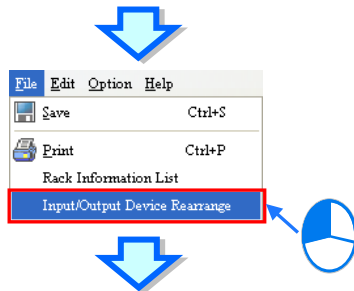
### 3.2.2.15 Rearrange Input/Output Device Range

For AH series, since the process of module configuration involves many executions to drag and drop or copy/cut/paste, the input/output device range for each module may be automatically assigned by the system and not according to a continuous sequence. Thus, users can click **Input/Output Device Rearrange** from the **File** menu in HWCONFIG toolbar and all the **Input/Output Device Range** of the module will rearrange base on the order of the slot and backplane number.

3

The execution does not change the corresponding device D setting of an internal module; when rearranging, if **Input/Output Device Range and Corresponding Device D** are not in conflict, the system will automatically skip the address for corresponding device D to program the next available address.

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	

### 3.2.3 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 will be also installed.

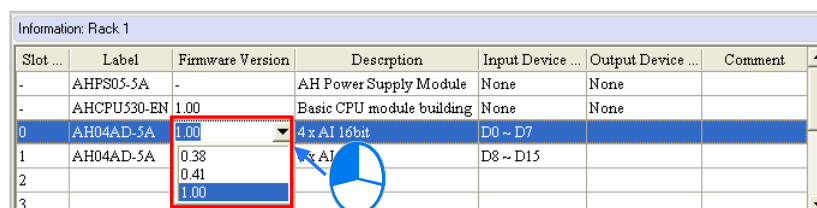
An MDS document provides the information about the firmware versions of the modules. If users want to configure modules in HWCONFIG, they can select a suitable configuration according to the firmware versions of the actual modules. Besides, the parameters which can be set in the modules vary with the 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 with the default values.
The firmware version set is newer than the actual firmware version.	The module cannot 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.

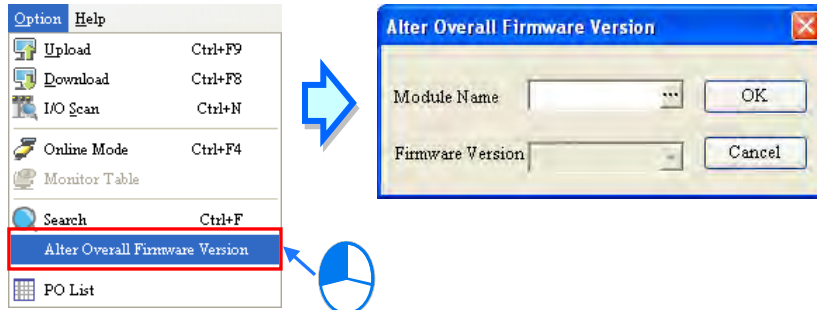
\*. The equipment description file can be in DDF or MDS format based on different models.

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.



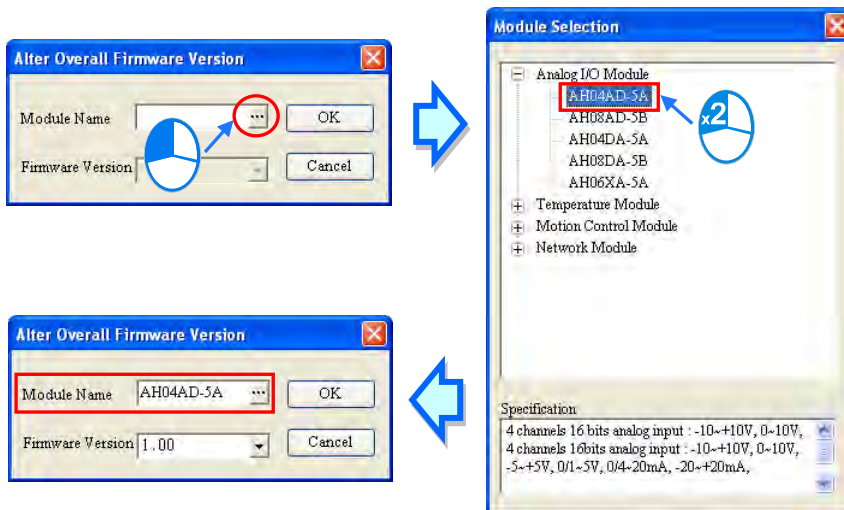
Slot ...	Label	Firmware Version	Description	Input Device ...	Output Device ...	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	1.00	Basic CPU module building	None	None	
0	AH04AD-5A	1.00	4 x AI 16bit	D0 ~ D7		
1	AH04AD-5A	0.38	4 x AI	D8 ~ D15		
2		0.41				
3		1.00				

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.

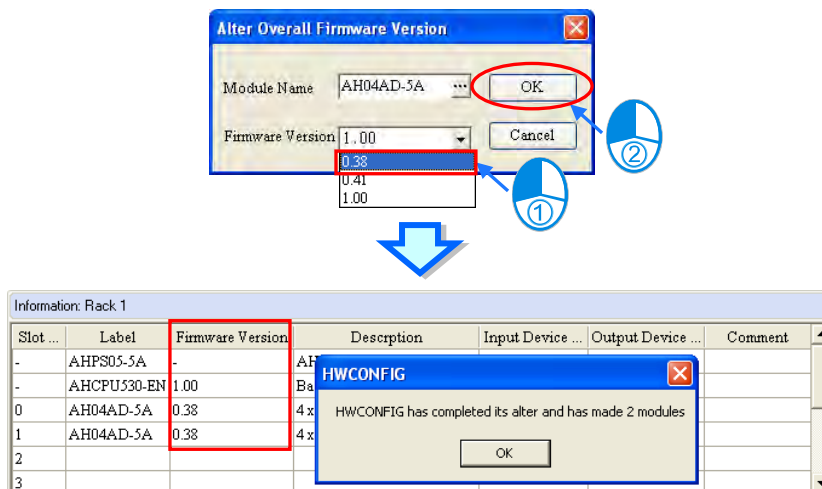


3

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. 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 ISPSOft 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 clicking **I/O Scan** on the **Option** menu, or viewing 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 **3.8.5** and section **3.8.6** for more information.



### 3.3 Parameter Settings for AH500 Series PLC

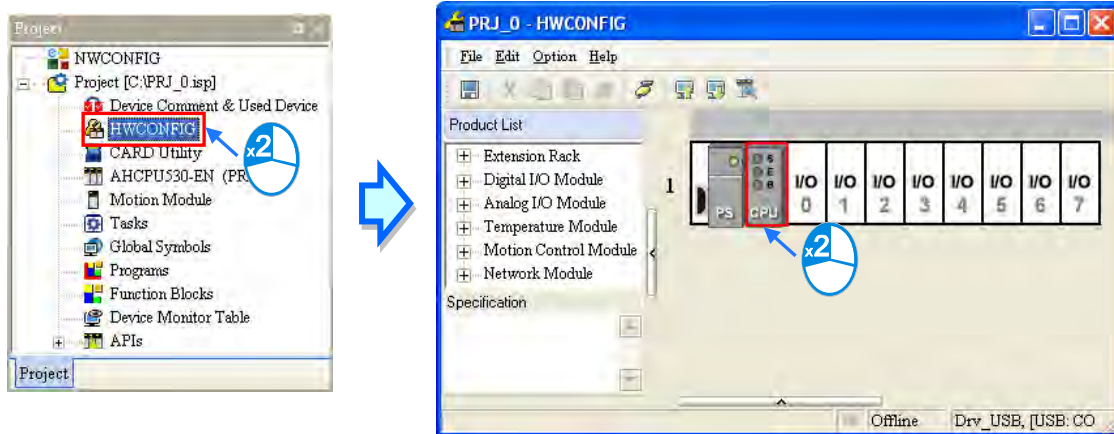
#### 3.3.1 Open the PLC CPU Parameter Setting Page

When HWCONFIG is enabled, double-click the CPU graphic in **System Configuration** area to open the **PLC CPU Parameter Setting** page. In addition, the setting paged for parameters differ based on different PLC model types.

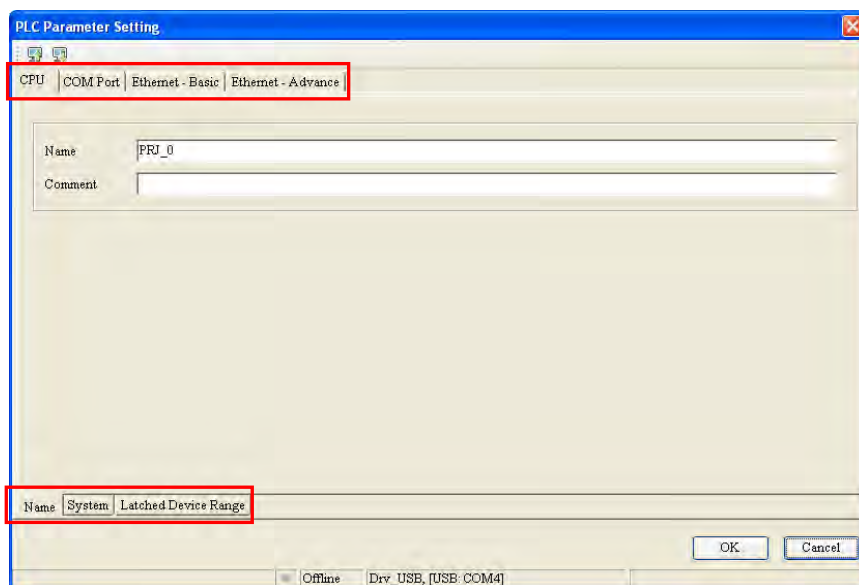
**⚠ Before setting the PLC CPU parameters, please check on user manuals for all PLC types and the effects of parameter settings to the PLC CPU as well as the whole system, in order to avoid any system or personnel risks.**

3

\*. Download the parameter once settings are complete, please refer to section 3.8.4 for ways to download.



The setting window contains many tabs based on different parameters. Users can use the main category tabs near the top of the window, while each sub-category tab refers to a parameter setting is shown near the bottom of the main category page.



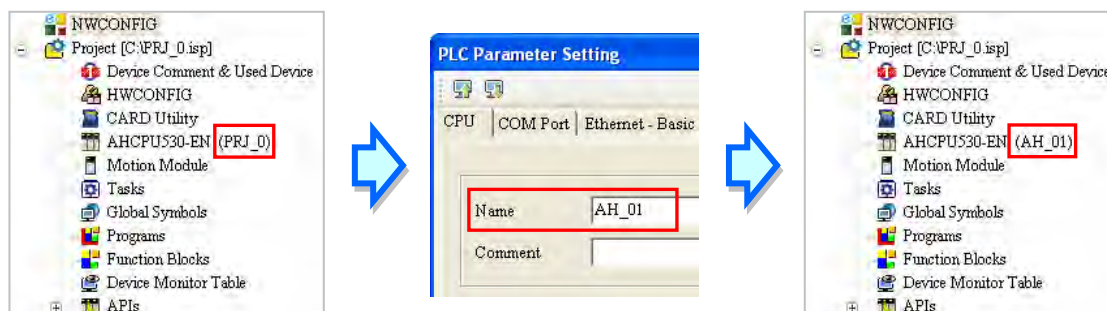
### 3.3.2 Setting CPU Parameters

#### 3.3.2.1 CPU Basic Parameter: Name

After users click the **CPU** tab on the top of the window, and the **Name** tab at the bottom of the window, the AH5x0 series can type 32 characters at most in the **Name** box, and 60 characters at most in the **Comment** box. While the AH5x1 series or AH560 redundant system, users can type 31 characters at most in the **Name** box, and 63 characters at most in the **Comment** box, there is no restriction on characters typed in the **Comment** box. Characters typed in the boxes can include special marks and spaces. However, please take note that a Chinese character occupies two characters.



When adding an ISPSOft project, the system uses the project name as the PLC CPU name by default and appears after the model type in the Project section which can be redefined from here.



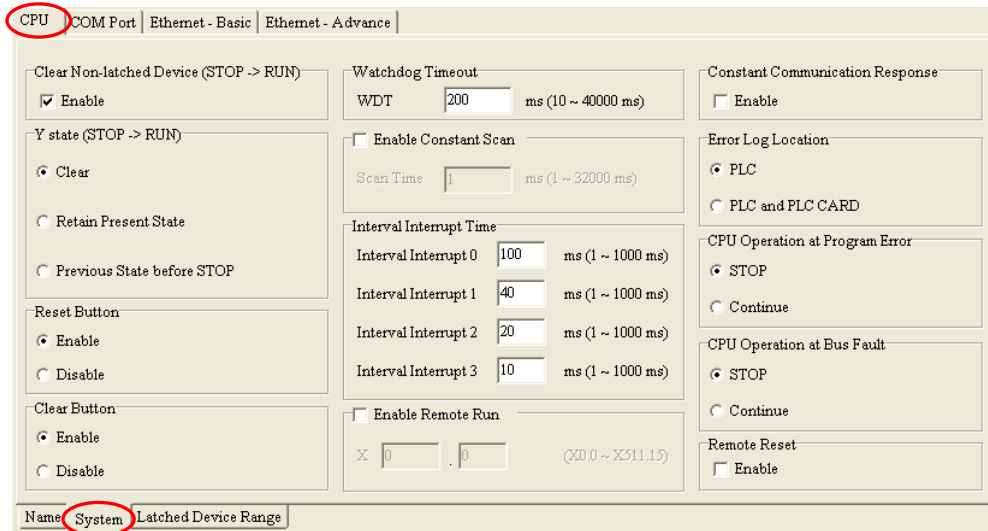
By using PLC label name for device identification, especially during networking applications or connecting, users can avoid operating mistakes through checking the PLC label name; several devices are connected to one network, users can check whether a device connected to the computer is the device they expect by checking the device name. To prevent errors, if users want to download/upload the program, 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, when the name of the CPU module is different from the name attached to the model in the project management area.



### 3.3.2.2 CPU Basic Parameter: System

In **System** sub-category page of **CPU** main parameter category, the page provides CPU parameter settings.

3



#### ● Clear Non-latched Device (STOP→RUN)

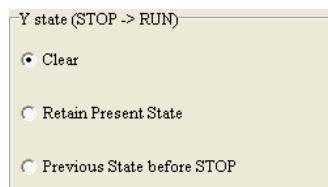
When **Enable** is selected and PLC CPU switches from STOP to RUN, the state of non-latched devices and values are cleared.



#### ● Y State (STOP→RUN)

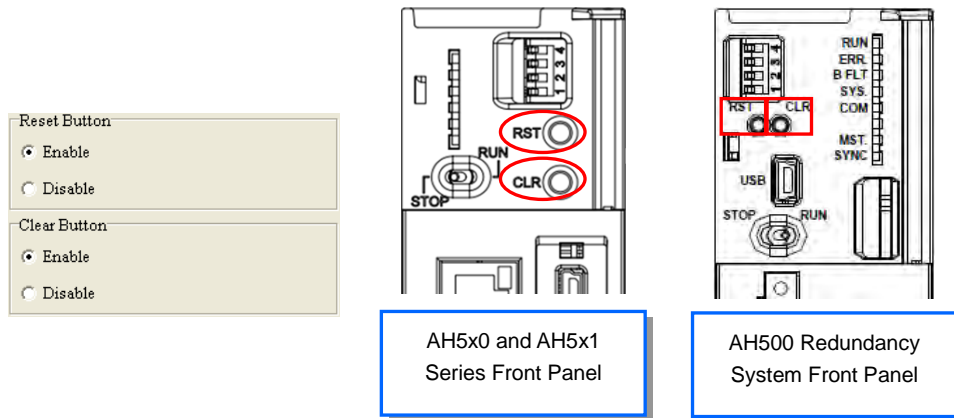
To set the PLC CPU state from STOP to RUN of Y state.

- **Clear:** Set all Y device output to OFF.
- **Retain Present State:** Output retained to present state.
- **Previous State before STOP:** Restore to previous states before STOP.



● **Reset Button & Clear Button**

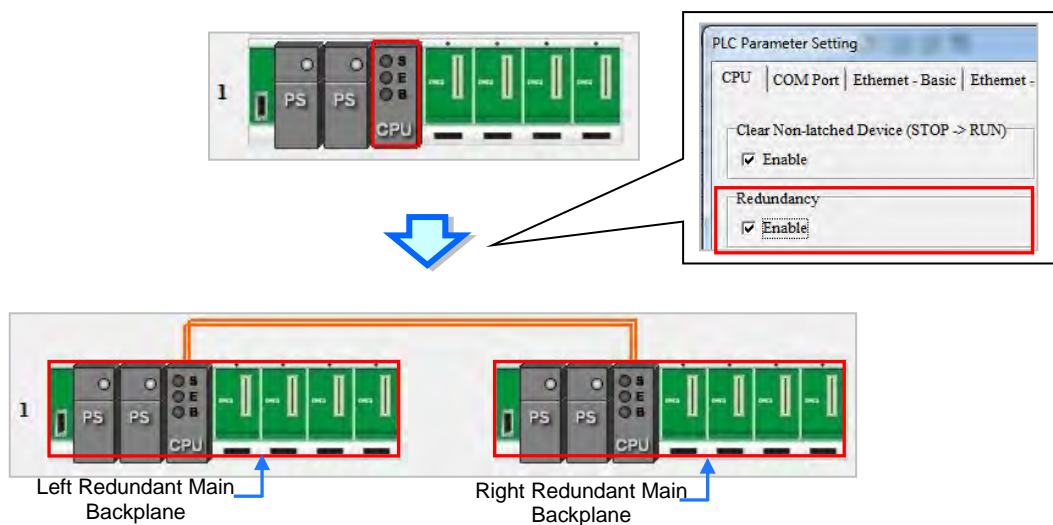
The parameters are used to setup and enable the function concerning Reset (RST) and Clear (CLR) button on CPU panels. For descriptions regarding button function and operation, please refer to user manuals for related PLC types.



3

● **Enable Redundancy**

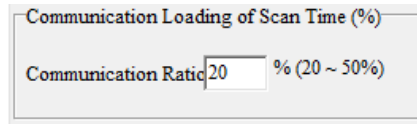
When enabled, two redundant main backplanes will appear in System Configuration area (see below) containing a left redundant main backplane and a right redundant main backplane which mirrors the left and cannot execute functions including add, delete or replace modules. To perform these functions, users need to use the left redundant main backplane. When Redundancy function is enabled, the expansion backplane in System Configuration area will automatically change to redundant expansion backplane; For redundant main backplane configuration module and functions, please refer to **AH560 redundant system related manuals**. **(Currently, only AH560 redundant system series support Redundancy Enable function.)**



### ● Communication Loading of Scan Time

When enabled, users can adjust communication loading of scan time.

(ONLY AH5x1 and AH560 redundant system series supports this function.)



Communication Loading of Scan Time (%)  
Communication Ratio: 20 % (20 ~ 50%)

## 3

### ● Watchdog Timeout

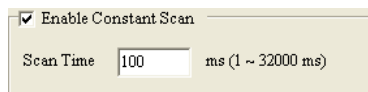
Users can set a timeout during which the program is scanned. Generally speaking, the watchdog timeout is reset whenever the scan of the program is complete. If the scan procedure cannot be completed within the time set for some reason, the watchdog timeout occurs, and the CPU module will be in an erroneous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



Watchdog Timeout  
WDT: 200 ms (10 ~ 40000 ms)

### ● 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.



Enable Constant Scan  
Scan Time: 100 ms (1 ~ 32000 ms)

### ● 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.

Interval Interrupt Time		
Interval Interrupt 0	100	ms (1 ~ 1000 ms)
Interval Interrupt 1	40	ms (1 ~ 1000 ms)
Interval Interrupt 2	20	ms (1 ~ 1000 ms)
Interval Interrupt 3	10	ms (1 ~ 1000 ms)

Task Type

- External Interrupt (236)
- External Interrupt (237)
- External Interrupt (238)
- External Interrupt (239)
- External Interrupt (240)
- External Interrupt (241)
- External Interrupt (242)
- External Interrupt (243)
- External Interrupt (244)
- External Interrupt (245)
- External Interrupt (246)
- External Interrupt (247)
- External Interrupt (248)
- External Interrupt (249)
- External Interrupt (250)
- External Interrupt (251)
- Timer Interrupt 0 (252)
- Timer Interrupt 1 (253)
- Timer Interrupt 2 (254)
- Timer Interrupt 3 (255)

Task Description

Interrupt Label: Timer Interrupt 0 (252)

CPU Timer Interruption  
Please set the parameter at  
HWCONFIG

Unassigned POU's

Assigned POU's

3

● **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.

Enable Remote Run

X   (X0.0 ~ X511.15)

● **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.

Constant Communication Response

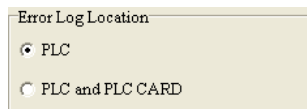
Enable

● **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.

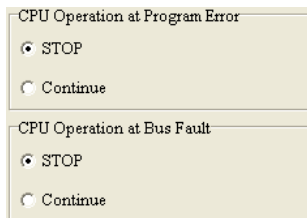
Twenty error logs at most can be stored in an AH500 series CPU module. If there are more than twenty error logs, the oldest error log will be overwritten by the latest error log. However, if the **PLC and PLC CARD** option button is selected, and there are more than twenty error logs, the oldest error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

3



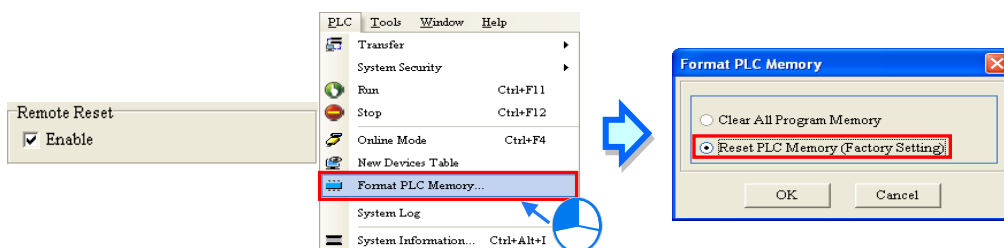
● **CPU Operation at Program Error & CPU Operation at Bus Fault**

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.



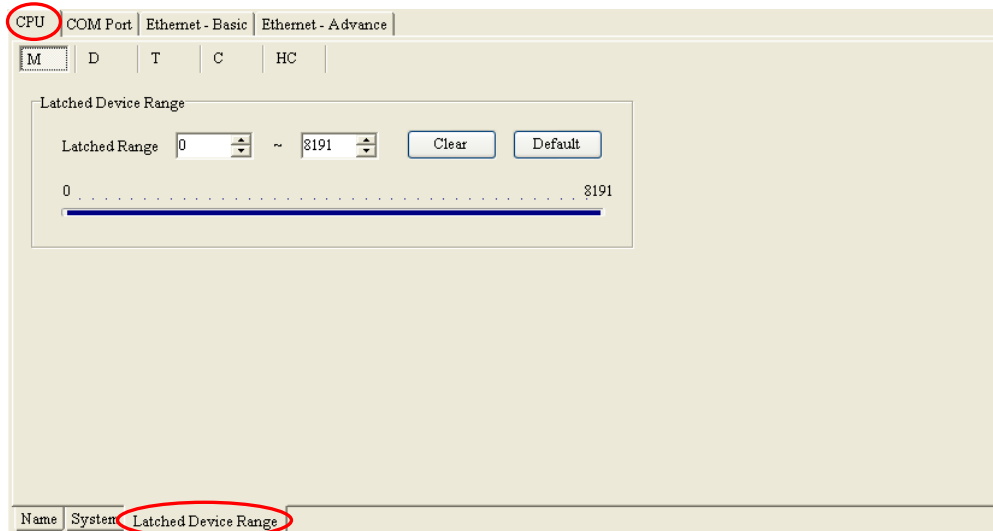
● **Remote Reset**

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.

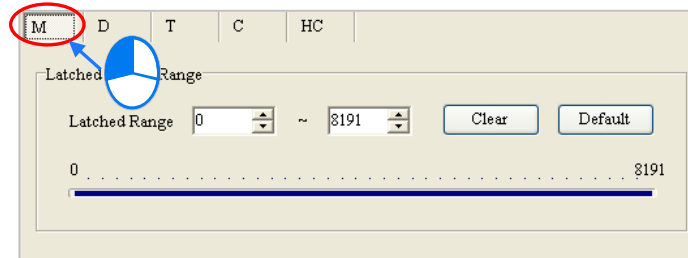


### 3.3.2.3 CPU Basic Parameter: Latched Device Range

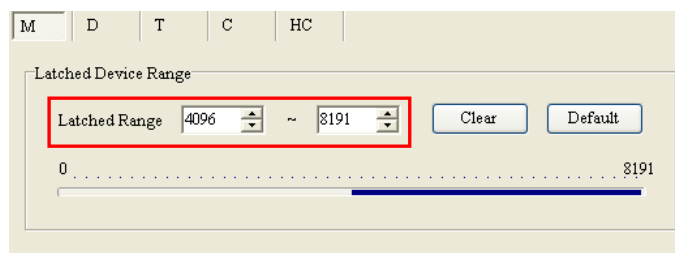
After users click the **CPU** tab on the top of the window, and the **Latched Device Range** tab at the bottom of the window, they can set a range of latched devices.



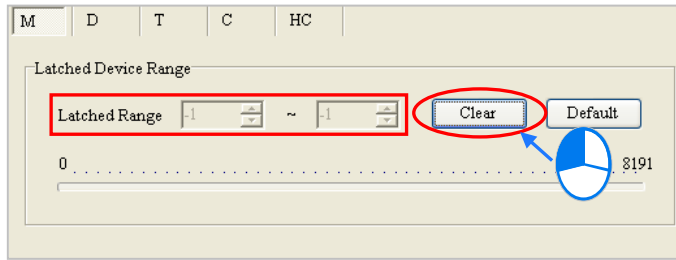
- (1) After users click a device type, they can be set a range of latched devices. There is a rectangular bar in the setting area. The values at the two ends of the bar represent a maximal value and a minimal value. The blue part represents a range of latched devices.



- (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 cannot be less than the minimal value at the left end of the rectangular bar, and the value in the right box cannot be larger than the maximal value at the right end of the rectangular bar.

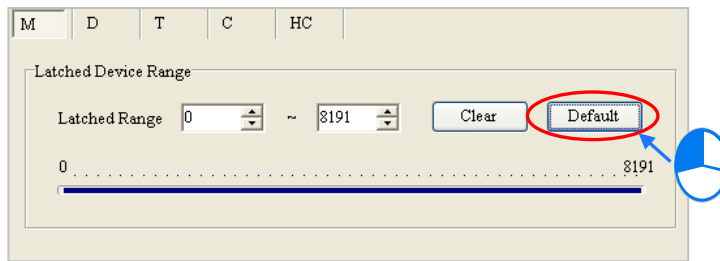


- (3) If all the devices in the block are not latched devices, the users can click **Clear**. After the users click **Clear**, the boxes become gray, and the range of latched devices will be -1~-1.



3

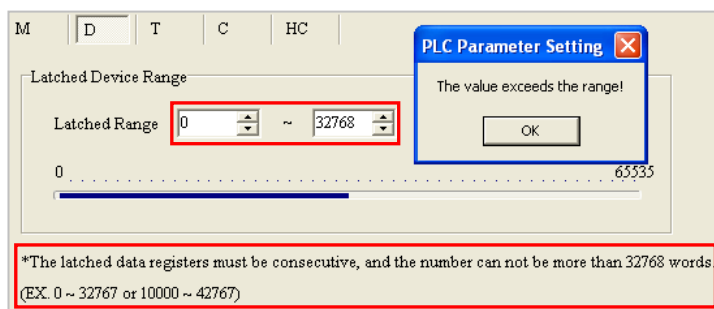
- (4) If the users want to restore the setting to the default values, they can click **Default**. After the users click **Default**, the setting will be restored to the default values.



**Additional remark**

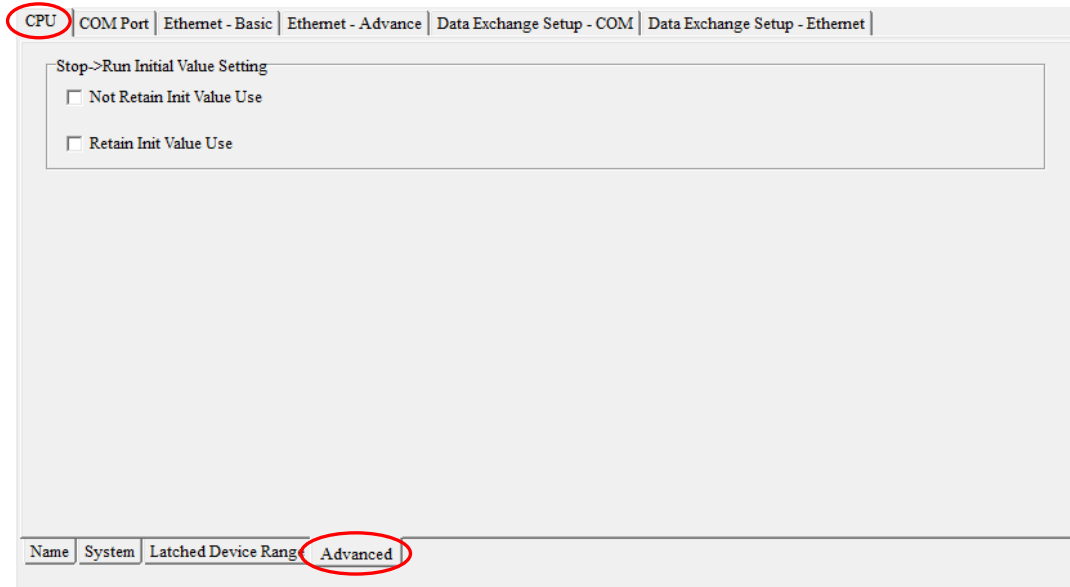
Some latched device range may not cover the entire area. When there is a maximum value restriction, a hint will appear under the setting area. The following image is the AHCPU530-EN model setting page for latched range of D device; since the permitted latched range total is 32768 words, this means users can setup the latched range to be smaller or equal to 32768 from the entire section setting range from 0~65535. For example, 0 ~ 32767 or 10000 ~ 42767.

Take the diagram below for instance, the image show input device range as 0~32768, yet the length has reached 32769 words, thus, when leaving the page, the system shows a warning message regarding PLC Parameter Setting.



### 3.3.2.4 CPU Advanced Parameter

Click the **CPU** tab on the top of the page and click the **Advanced** tab at the bottom of the page, you can set the initialization related settings.



3

When PLC CPU switches from stop to run, the system initializes the retentive values or the non-retentive value.

- **Not Retain Init Value Use:** When PLC CPU switches from stop to run, the system initializes the non-retentive value.
- **Retain Init Value Use:** When PLC CPU switches from stop to run, the system initializes the retentive value.



### 3.3.3 Setting COM Port Parameters

When users click **COM Port** from the main category label and enters the setting page and two COM port is setup for the model type, e.g. AHCPU5xx-RS2. The setting page will have two identical COM port setting section but both are independent from each another. For notes on connecting PLC CPUs through COM ports, please refer to section 2.4.8 for more information.

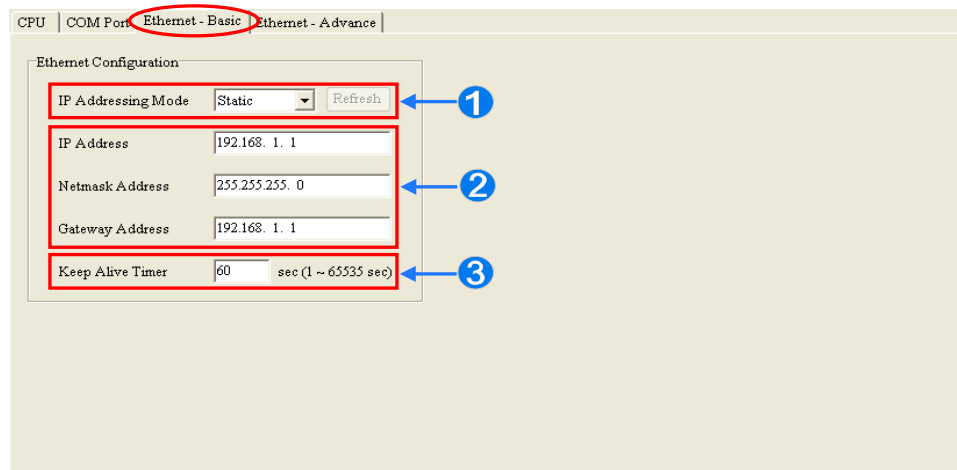
- 1 Select types of communication with options including **RS232**, **RS485**, or **RS422**.
- 2 Select the **Baud Rate** for the communication speed. When choosing RS232, the baud rate options include 9600 bps, 19200 bps, 38400 bps, 57600 bps and 115200 bps; while for RS485 or RS422, the baud rate options include 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps, 230400 bps, 460800 bps, or 921600 bps.
- 3 Setup other communication protocol parameters. When **RTU** is selected for **Transfer Mode**, the **Data Length** is limited to choosing **8 bit**.
- 4 Setup COM Port station address for device identification during connection, while station address on the same network is not allowed to be repeated, the setting range is 0 ~ 247.

When the port is referred to as a slave and there are other slaves connected, the port station address cannot be set as 0 because it is defined as broadcast communication under the communication protocol. However, when the Master assigns data transfer to Slave station address 0, this means the data packets are transferred to all slave stations regardless of the station address.

- 5 When communication fails, the PLC CPU will retry to connect. Users can setup the times of auto-retry within the range of 0~20.
- 6 Setup the time interval for auto-retry once communication fails. When PLC CPU retries to connect due to communication failure, users need to setup the interval time to start the next connection. The setting range is 100~65535 ms.

### 3.3.4 Setting Ethernet Parameters

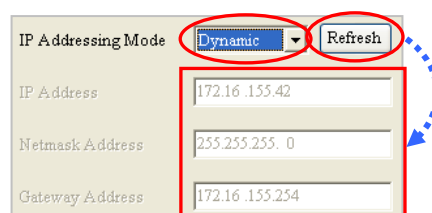
Users need to select **Ethernet - Basic** from the main category label for parameter settings, e.g. AHCPU5xx-RS2. The page provides parameter settings for Ethernet port. For information regarding PLC CPU connection through Ethernet, please refer to section 2.4.8. As for descriptions about the Ethernet, please view other related documents or references.



- ① Setup the IP address mode. When choosing **Static**, this means users can directly assign the IP address parameter; but when selecting **DHCP** or **BOOTP**, this means the IP address is managed by the dynamic allocation of **DHCP** or **BOOTP**.
- ② When IP address mode is **Static**, users can setup the **IP Address**, **Netmask Address**, and **Gateway Address**.
- ③ When the PLC CPU is not transmitting any data on the internet and already exceeds the Keep Alive Timer, it will automatically disconnect.

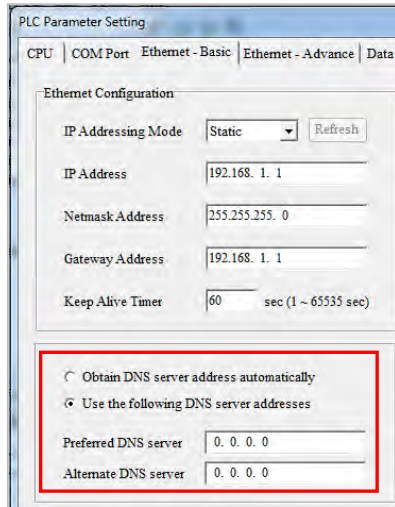
If **Dynamic** is selected in the **IP Addressing Mode** drop-down list box, the IP address, the subnet mask, and the gateway can be read from the CPU module by means of the following steps.

- (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.



For AH5x1 or AH560 redundant system series, users can use the DNS server to auto-retrieve the DNS server address or adopt user-defined one or other server addresses.

3

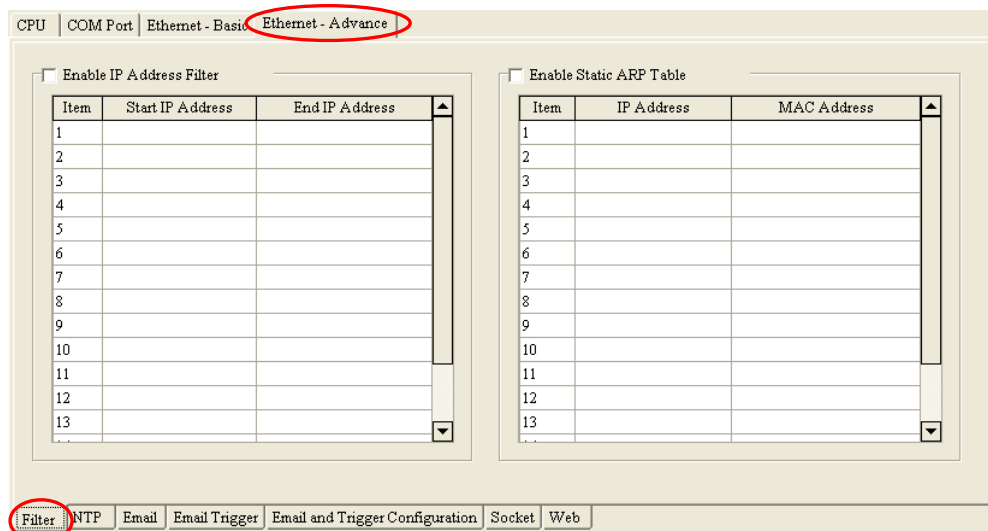


### 3.3.5 Setting Advanced Ethernet Parameters

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 on the top of the window, they can set devices on a network.

#### 3.3.5.1 Ethernet - Advance: Filter

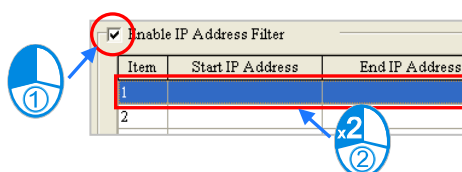
After users click the **Ethernet - Advance** tab on the top of the window, and the **Filter** tab at the bottom of the window, they can set a filter. Devices on a network are filtered. The setting here ensures that objects communicating with the CPU module are devices which are allowed.



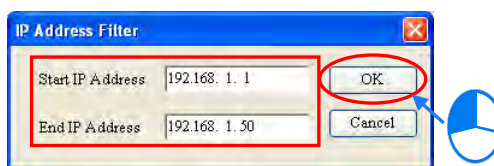
### ● Enable IP Address Filter

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.



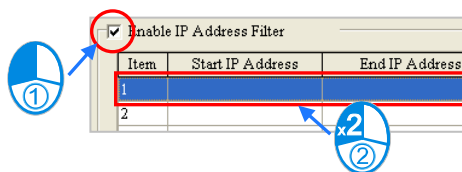
- (2) Type a start IP address and an end address in the **IP address Filter** window. The start IP address must be less than the end IP address. Finally, click **OK**.



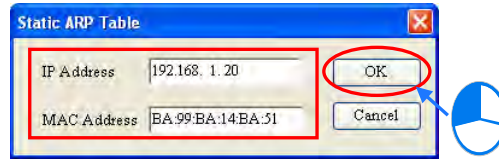
### ● Enable Static ARP Table

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.

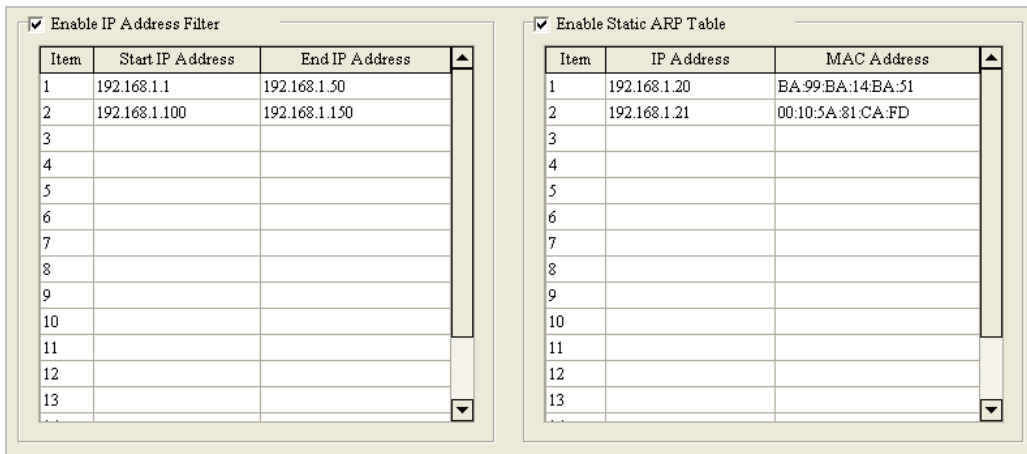


(2) Type an IP address and a MAC address in the **Static ARP Table** window. Finally, click **OK**.



3

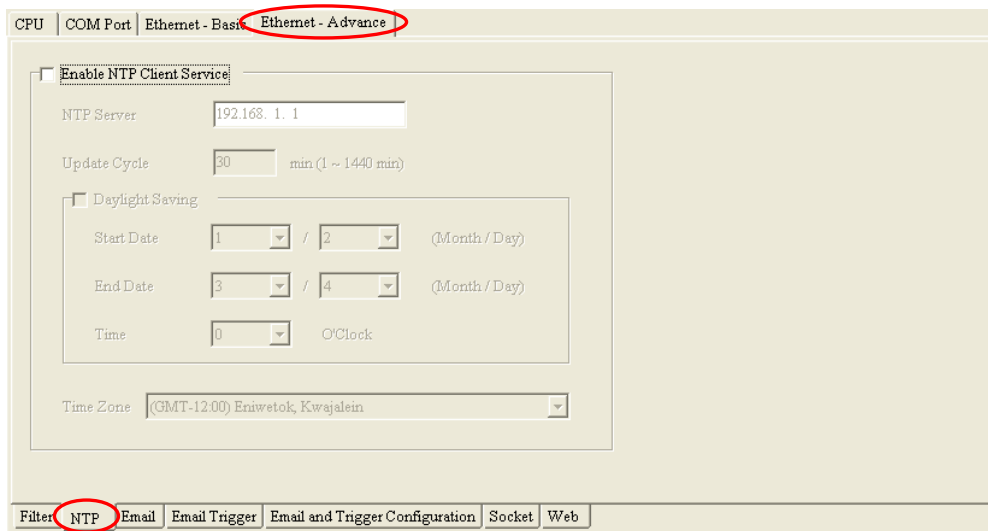
Take the tables below for example. After the setting in the tables is downloaded to the CPU module, devices whose IP addresses are within the range between 192.168.1.1 and 192.168.1.50, and devices whose IP addresses are within the range between 192.168.1.100 and 192.168.1.150 are allowed to communicate 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.



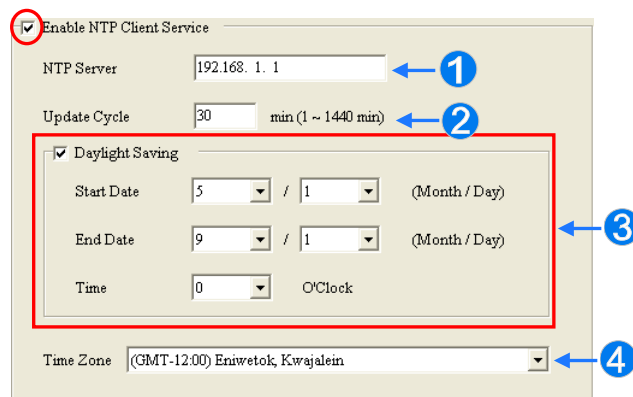
\*1. If users want to know the MAC address of a CPU module, they can click Online Mode in HWCONFIG to view the information about the CPU module (section 3.8.6), or click System Information in ISPSOft to get the information about the CPU module connected and the status of the CPU module (section 18.6.1).

### 3.3.5.2 Ethernet - Advance: NTP

After users click the **Ethernet—Advance** tab on 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.

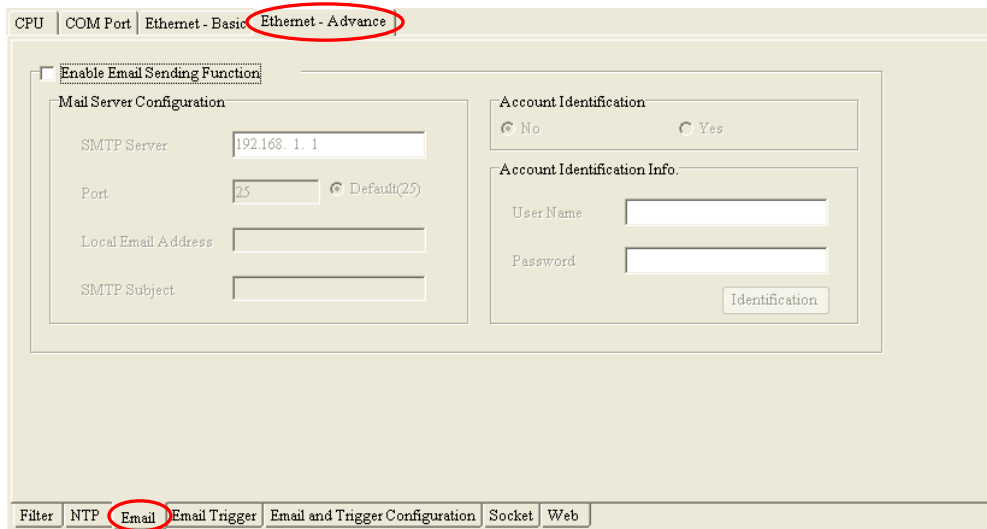


- ① Users can set the IP address of an NTP server. The CPU module corrects the time inside itself by connecting to the server periodically.
- ② Users can set an interval of correcting the time in the CPU module. If the interval is thirty minutes, the CPU module will connect to the NTP server every thirty 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.

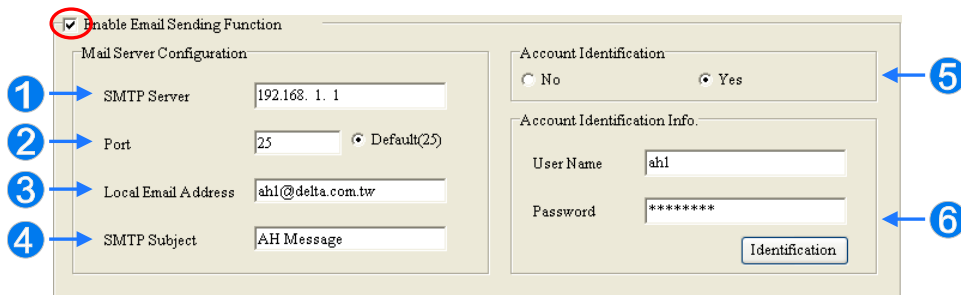
### 3.3.5.3 Ethernet - Advance: Email

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

3



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



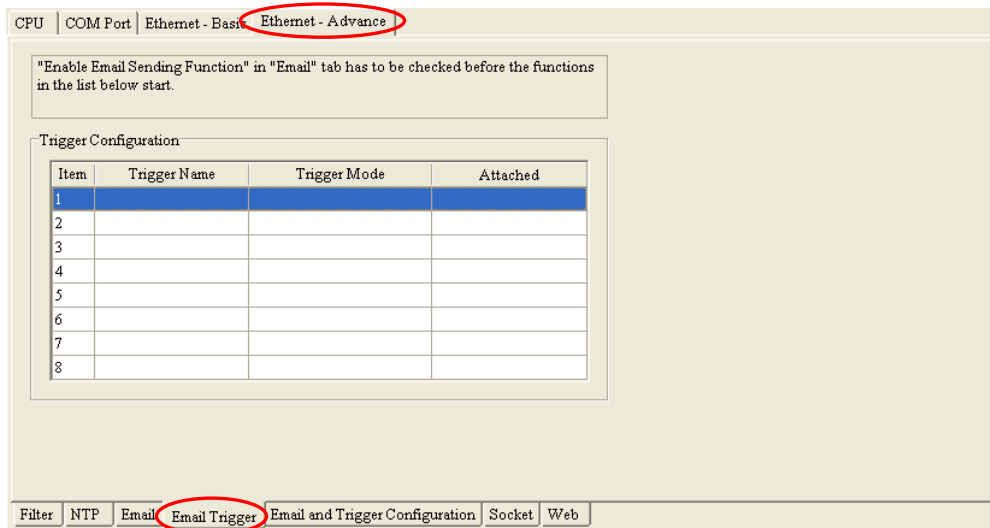
- 1 **SMTP Server:** Users can set the IP address of an SMTP server.
- 2 **Port:** Users can set a port for an SMTP server, or select the **Default (25)** option button.
- 3 **Local Email Address:** Users can set the address of an actual sender. A local email address is composed of 64 characters at most.
- 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.

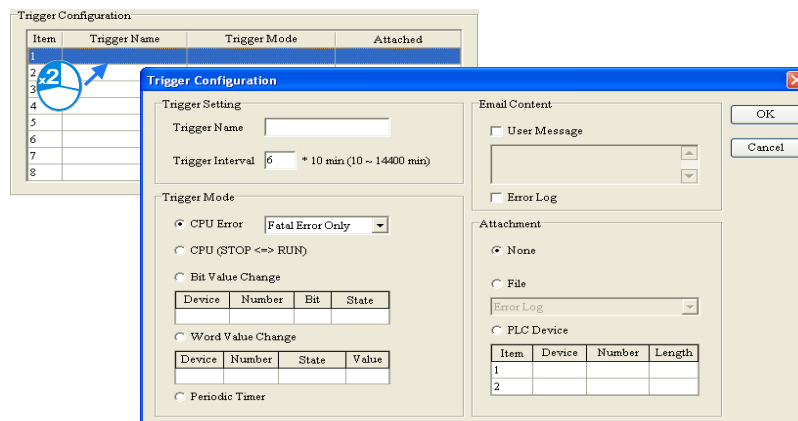
### 3.3.5.4 Ethernet - Advance: Email Trigger

After users click the **Ethernet - Advance** tab on 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.

3



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 be sent within the intervals.

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

Device	Number	Bit	State

Device	Number	State	Value

### ● 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.

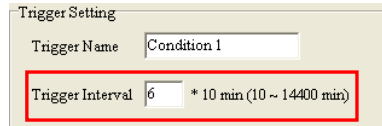
(b) **All Errors**: If an error occurs, the condition of triggering the sending of an email is met.

### ● 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**

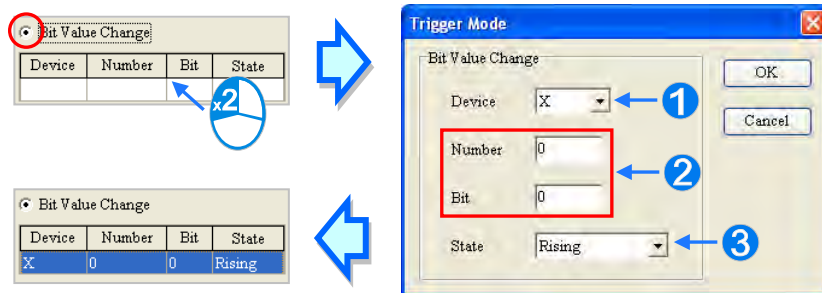
An email is sent periodically. How often an email is sent depends on the interval typed in the **Trigger Interval** box in the **Trigger Setting** section.



● **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.

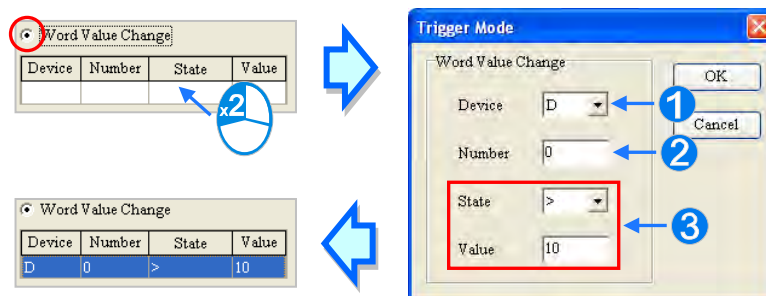
3



- ① **Device:** Users can select a device type in the **Device** drop-down list box.
- ② **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.
- ③ **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.

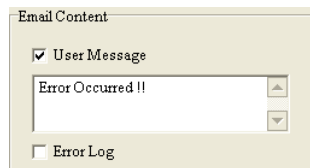


① **Device:** Users can select a device type in the **Device** drop-down list box.

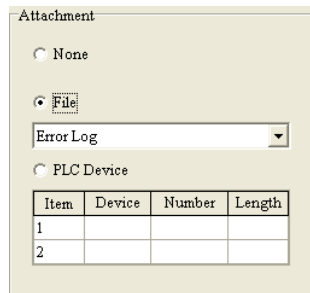
② **Number:** Users can type a device address in the **Number** box.

③ **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**

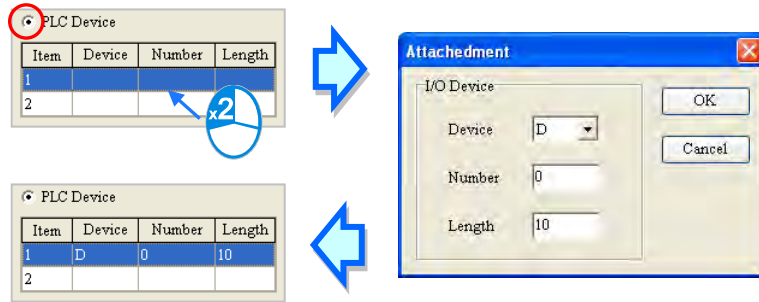
If this option button is selected, no attachment will be inserted.

● **File**

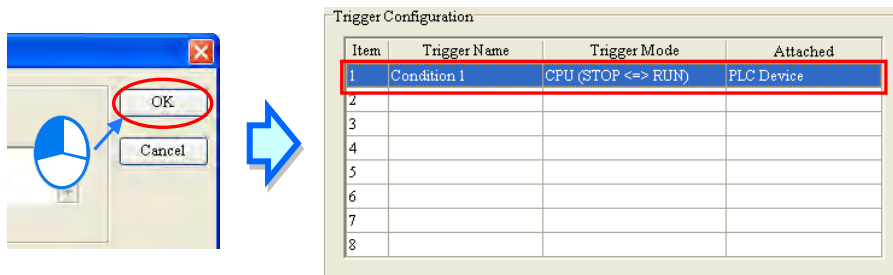
Users can select an error log in the memory card, or the system backup file in the memory card as the attachment of the email.

● **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.

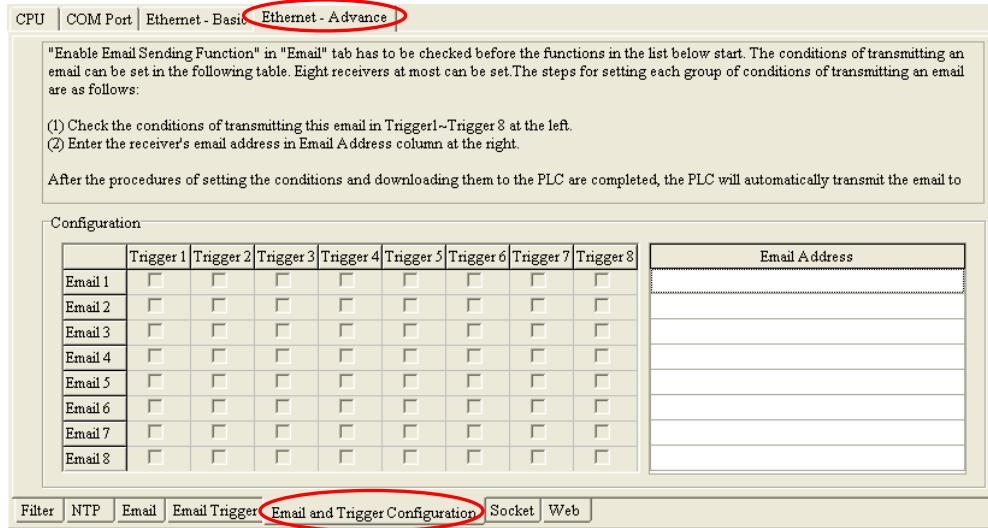


3

### 3.3.5.5 Ethernet - Advance: Email and Trigger Configuration

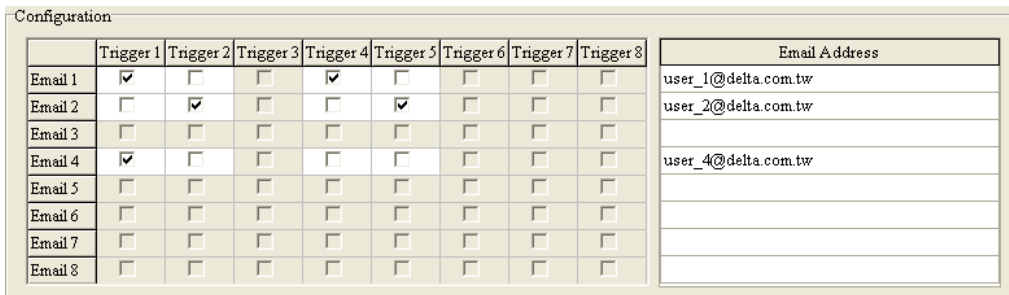
After users click the **Ethernet–Advance** tab on the top of the window, and the **Email and Trigger Configuration** tab at the bottom of the window, they can set the relation between triggers for the sending of emails and receivers. Before setting the function here, the users have to click the **Email** 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.

3



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.

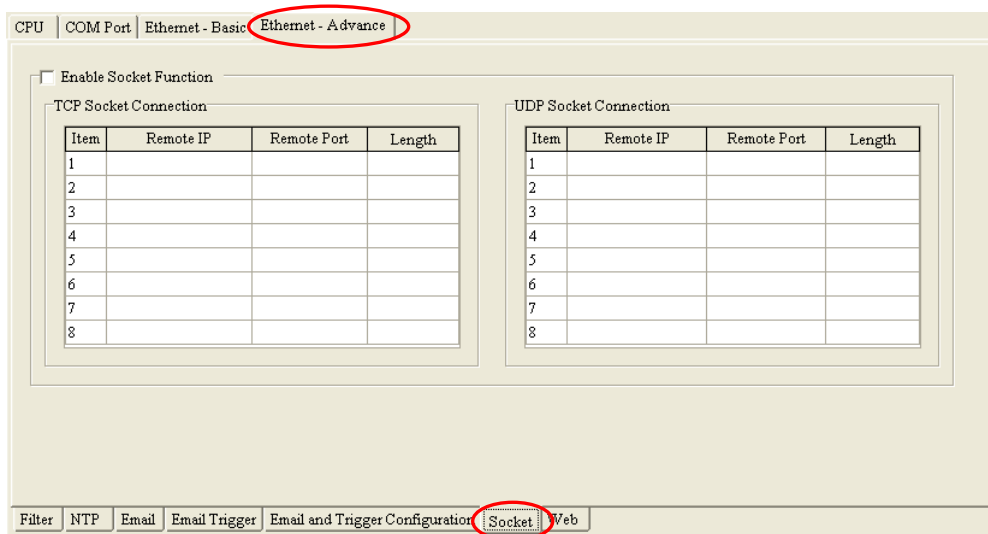
For example, if the condition represented by trigger 1 or the condition represented by trigger 4 is met, the system will automatically send the emails to **user\_1@delta.com.tw**. If the condition represented by trigger 2 or the condition represented by trigger 5 is met, the system will automatically 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**.



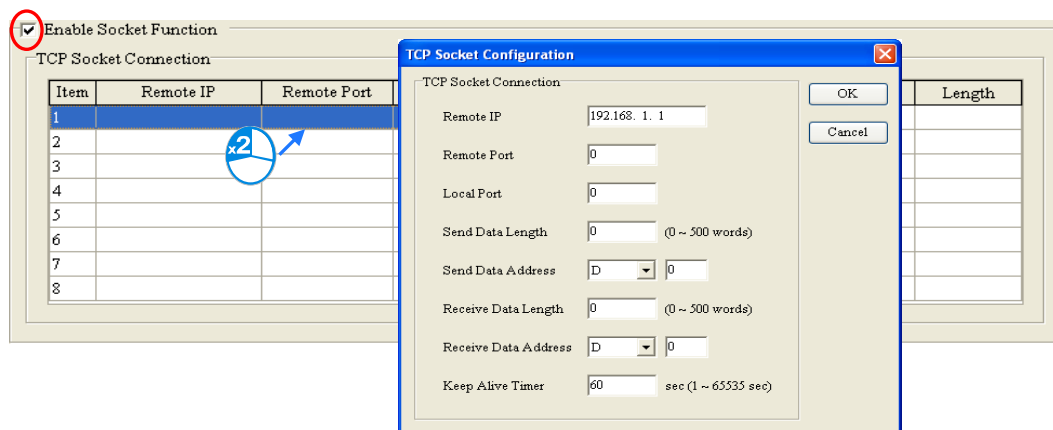
### 3.3.5.6 Ethernet - Advance: Socket

After users click the **Ethernet - Advance** tab on 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.

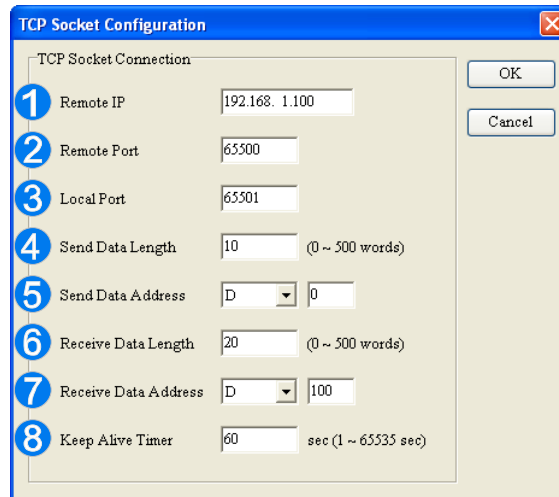
Sockets will not be introduced here. Please refer to related documents or manuals for more information about sockets.



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 keep alive time for the connection. If no data is transmitted, and the keep alive 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 cannot 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.

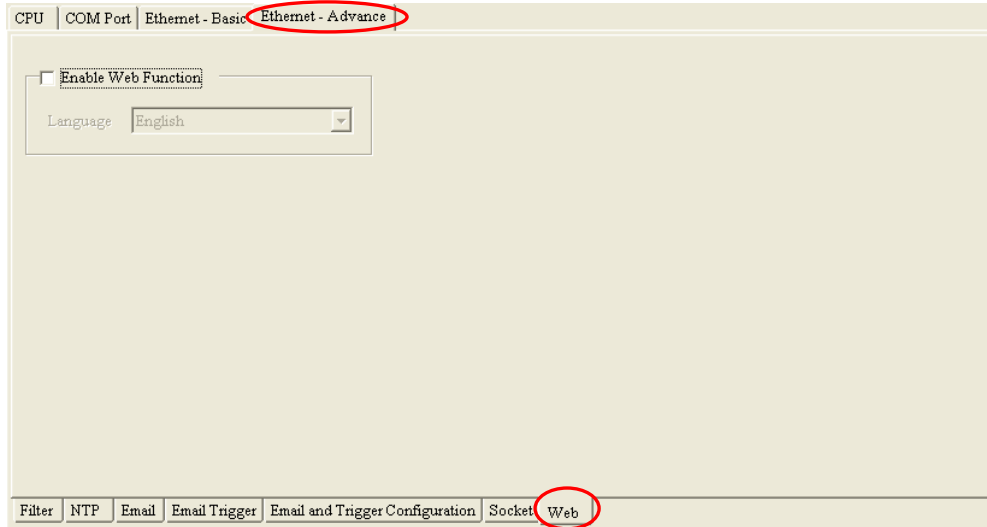




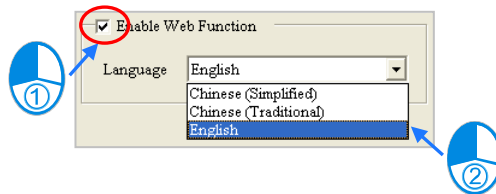
### 3.3.5.7 Ethernet - Advance: Web

After users click the **Ethernet - Advance** tab on 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.

3



Select the **Enable Web Function** checkbox, and 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.



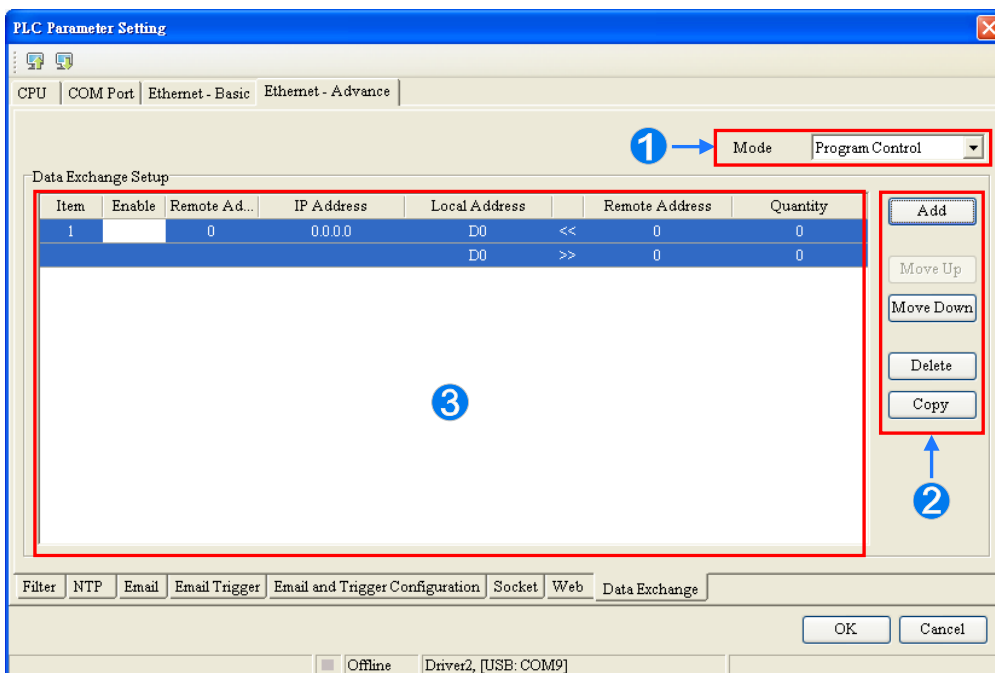
\*. Please refer to operation manuals for more information about the web monitoring function.

### 3.3.5.8 Ethernet - Advance: Data Exchange

The CPU module of AH5x0-EN series contains the data exchange function based on MODBUS TCP. The built-in function is enabled by clicking the **Ethernet - Advance** tab on the top of the window, and the **Data Exchange** tab at the bottom of the window as below.



If users open this page of the PLC Parameter Setting window for the first time, the table is blank due to no data set. A new row of data can be inserted by clicking **Add**. Double-click the data so that the data setting window is opened.

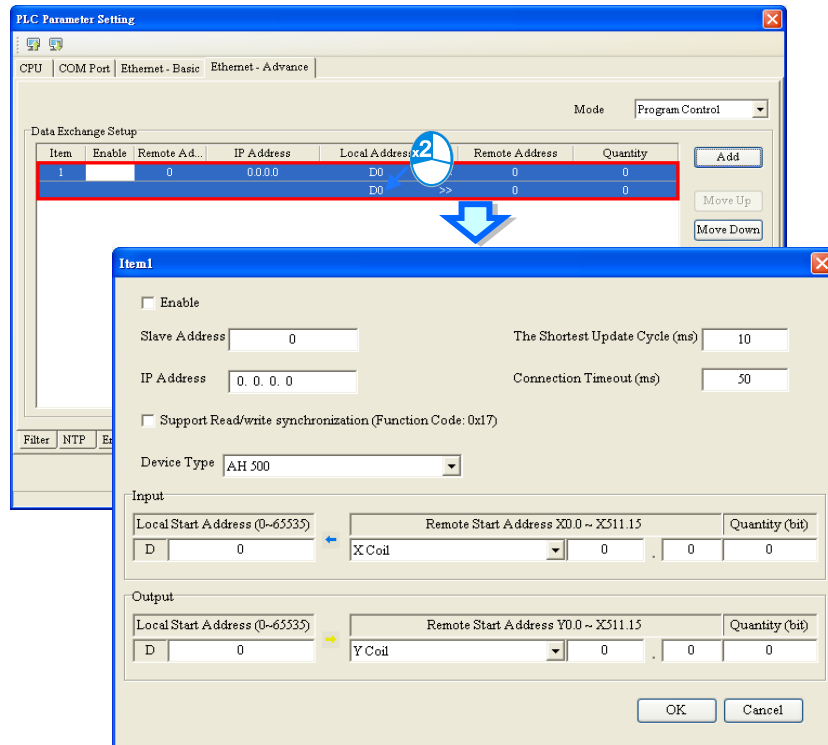


- ❶ Set the starting mode to **Program Control**; PLC decides whether the set data exchange is performed. **PLC Run** means that the set data exchange will be performed automatically when PLC is in RUN state. **Always Enable** means that the data exchange will be performed constantly when PLC is powered on.
- ❷ A new data exchange table will be added in area ❸ by clicking **Add**. The present data exchange table selected will be moved up or down by clicking **Move Up** and **Move Down**. The present data exchange table selected will be deleted by clicking **Delete**. The present data exchange table selected will be copied by clicking **Copy** and then can be pasted to the last row.
- ❸ The data exchange table is listed and explained as below.

3

Name	Description
Item	The number of the block for data exchange
Enable	To select whether to enable the data exchange table when the data exchange is performed.
Remote Station Address	The target slave station address of the data exchange table
IP Address	The target IP address of the data exchange table. One IP address can be set for multiple data exchange tables.
Local Address	The device address range used by the master in the data exchange table
<< / >>	[ << ] : Input; the data block where the master read a slave [ >> ] : Output; the data block where the master write to a slave
Remote Address	The device address range used by a slave in the data exchange table The device range will be in hexadecimal display, if the slave in the data exchange table is a user-defined MODBUS Device.
Quantity	The size of the data exchange table, which is basically consistent with the result calculated from the device range.

Double click a data exchange table to be set so as to open the Item window as below. Select the checkbox of **Enable** to enable the data exchange table in the mode mentioned above. **Remote Address** is the address of the target slave for data exchange. **The Shortest Update Cycle** is the period for data exchange of the data exchange table. If the target device does not make any response within the time specified by **Connection Timeout**, that is timeout. The master can complete the read and write in a command to improve the efficiency of data exchange by selecting **Support Read/write synchronization (Function code 0x17)** in which the specific MODBUS function code is used. Please make sure that all devices in data exchange support MODBUS function code for read and write synchronization. Otherwise, the read and write will fail on account of the failure to identify the function codes after the command from the master is received. **Device Type** is the model of a slave including Delta PLC and standard MODBUS devices.



3

- **Input**

AH5x0-EN series read data from a remote device.

**Local Start Address:** The start address of D data registers where AH5x0-EN series store data

**Remote Start Address:** Device type and start address of the remote device to be read

**Quantity:** Data length of input

- **Output**

AH5x0-EN series write data to a remote device.

**Local Start Address:** The start address of D device type of the source data of AH5x0-EN series

**Remote Start Address:** Device type and start address of the remote device to be written

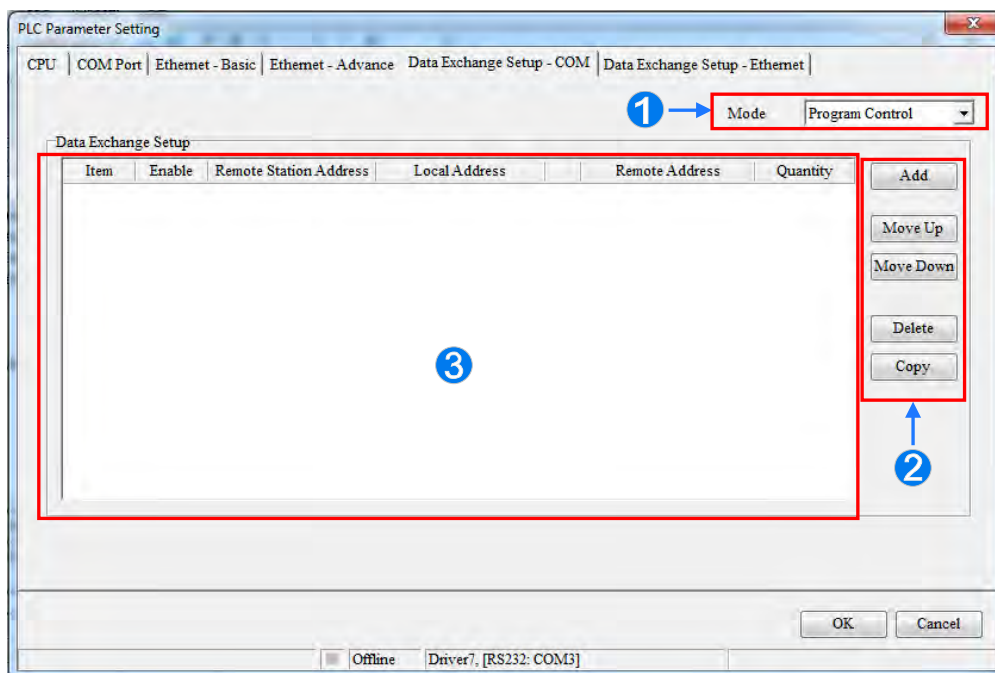
**Quantity:** Data length of output

While the data exchange setup for AH5x0-EN series is conducted, the local device can only be the data register D with Word as the unit. And remote address can take Bit as the unit which may not start from 0, such as D0.3 as a remote start address. If the set data size is not a multiple of Word, the local devices will still make the data exchange via corresponding bits and the bits which are not in data exchange will keep original state. For example, if the data in remote M0~M19 is read to local data register D0~D1, the data will be stored in D0.0~D1.3 in order and D1.4~D1.15 will keep their original states.

### 3.3.6 Data Exchange Setup - COM & Ethernet

For AH5x1 and AH560 redundant system series, there are additional tabs for data exchange setup in PLC Parameter Setting window, such as two COM ports or one COM port and one Ethernet port according to different models. The parameters are set on the tabs of **Data Exchange Setup-COM1** and **Data Exchange Setup-COM2**, or **Data Exchange Setup-COM** and **Data Exchange Setup-Ethernet** for making the data exchange table of the communication port built in AH5x1 series. AH5x1 and AH560 redundant system series working as the master exchanges data with the remote devices working as slaves via COM1 and COM2 based on MODBUS protocol or via Ethernet port based MODBUS TCP protocol.

3



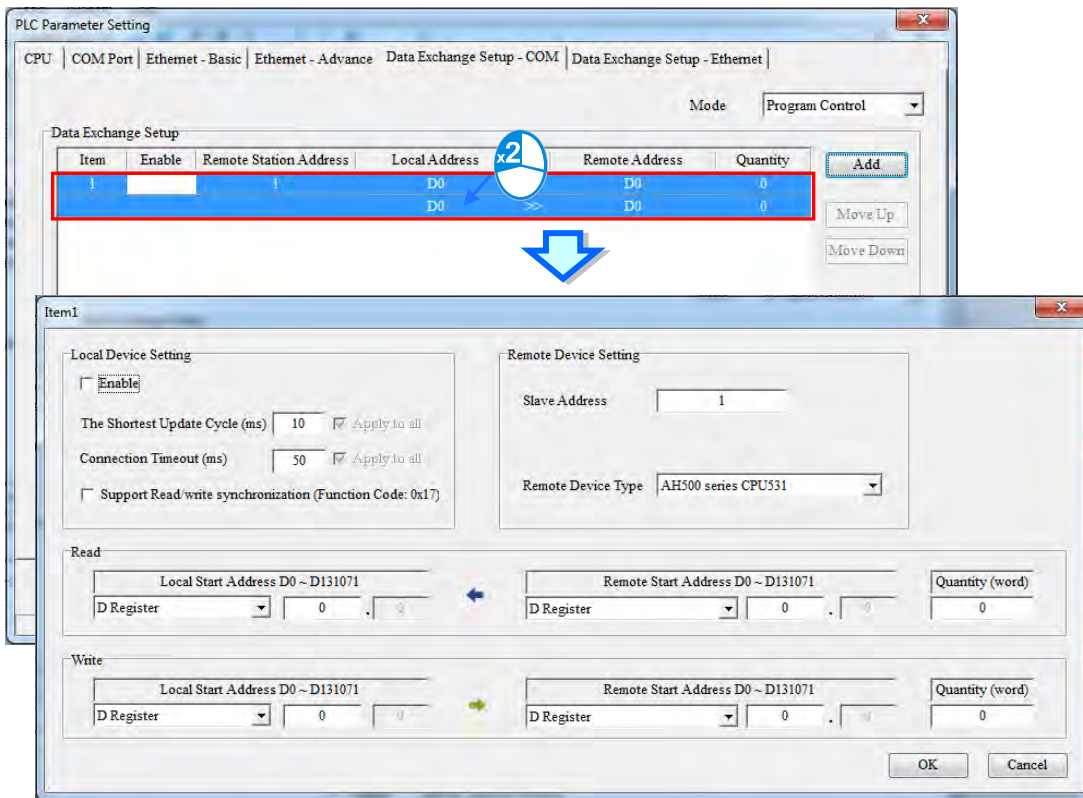
- ❶ Set the starting mode to **Program Control**; PLC decides whether the set data exchange is performed. **PLC Run** means that the set data exchange will be performed automatically when PLC is in RUN state. **Always Enable** means that the data exchange will be performed constantly when PLC is powered on.
- ❷ A new data exchange table will be added in area ❸ by clicking **Add**. The present data exchange table selected will be moved up or down by clicking **Move Up** and **Move Down**. The present data exchange table selected will be deleted by clicking **Delete**. The present data exchange table selected will be copied by clicking **Copy** and then can be pasted to the last row.
- ❸ The data exchange table is listed and explained as below.

Name	Description
Item	The number of the block for data exchange
Enable	To select whether to enable the data exchange table when the data exchange is performed.
Remote Station Address	The slave station address of the data exchange table, One address can be set for multiple data exchange tables. It is a station address for COM1 and COM2 tabs and IP address for Ethernet tab.
Local Address	The device address range used by the master in the data exchange table
<< / >>	「<<」 : Input; the data block where the master read a slave 「>>」 : Output; the data block where the master write to a slave
Remote Address	The device address range used by a slave in the data exchange table The device range will be in hexadecimal display, if the slave in the data exchange table is a user-defined MODBUS Device.
Quantity	The size of the data exchange table, which is basically consistent with the result calculated from the device range.

### 3.3.6.1 Data Exchange Setup - COM1/COM2 /COM

On the page of tab **Data Exchange Setup - COM1**, **Data Exchange Setup - COM2**, or **Data Exchange Setup - COM**, double click a data exchange table which is to be set to open Item window as below. Select the checkbox of Enable to enable the data exchange table in the mode mentioned above. **Remote Address** is the address of the target slave for data exchange. **The Shortest Update Cycle** is the period for data exchange of the data exchange table. If the target device does not make any response within the time specified by **Connection Timeout**, that is timeout. The master can complete the read and write in a command to improve the efficiency of data exchange by selecting **Support Read/write synchronization (Function code 0x17)** in which the specific MODBUS function code is used. Please make sure that all devices in data exchange support MODBUS function code for read and write synchronization. Otherwise, the read and write will fail on account of the failure to identify the function codes after the command from the master is received.

**Remote Device Type** is the model of the target slave including Delta PLC and standard MODBUS devices.



- **Input**

AH5x1-EN series read data from a remote device.

**Local Start Address:** Device type and start address of devices where AH5x1-EN series store data;

**Remote Start Address:** Device type and start address of the remote device to be read;

**Quantity:** Data length of input

- **Output**

AH5x1-EN series write data to a remote device.

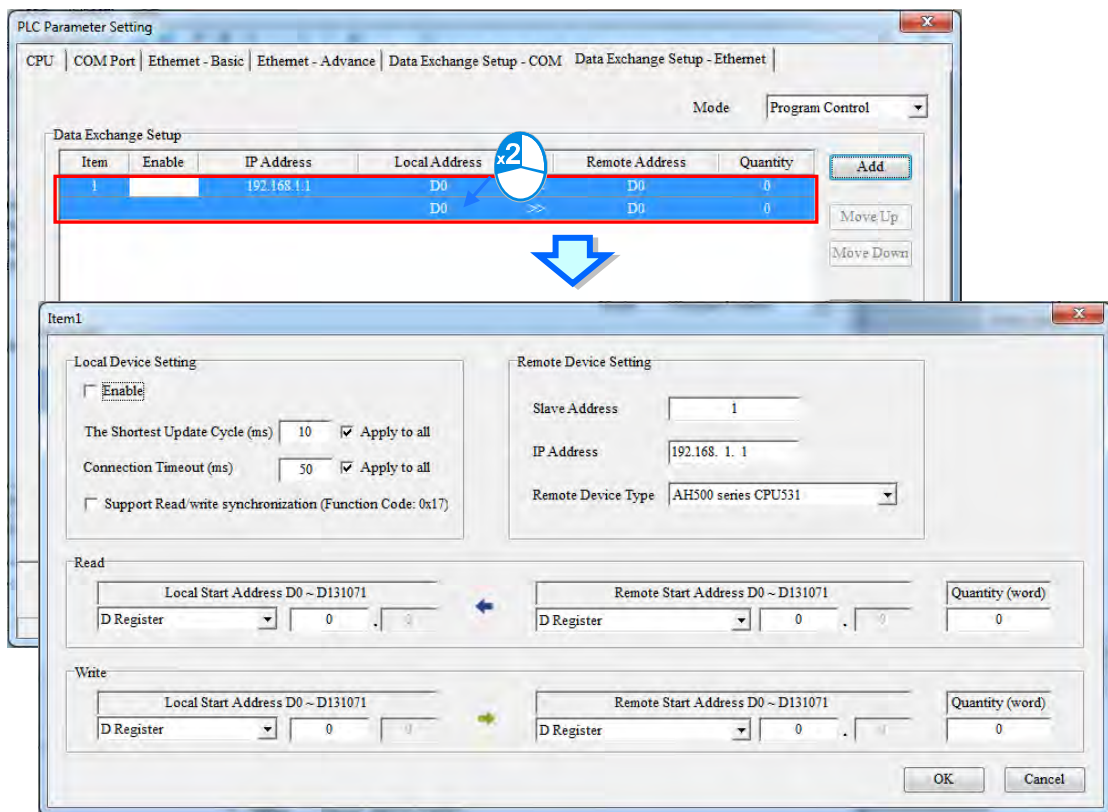
**Local Start Address:** Device type and start address of the source data of AH5x1-EN series;

**Remote Start Address:** Device type and start address of the remote device where data are to be written;

**Quantity:** Data length of output

### 3.3.6.2 Data Exchange Setup - Ethernet

On the page of tab **Data Exchange Setup - Ethernet**, double click the data exchange table to be set so as to open the Item window as below. The settings are almost same as that for **Data Exchange Setup - COM** except the target slave model option in Device Type which contains Delta PLC and standard MODBUS TCP equipment as well as IP address for typing the IP of remote slaves. For other settings, refer to the content of **Data Exchange Setup - COM**.



Select the check box of **Apply to all** to the right side of **The Shortest Update Cycle** and **Connection Timeout** and then click OK. The settings for **The Shortest Update Cycle** and **Connection Timeout** will be written to other data tables on the tab **Data Exchange Setup - Ethernet**. Users could set a data exchange table separately without selecting the check box of **Apply to all**.



### 3.4 Parameter Settings for AHxxEMC Series PLC

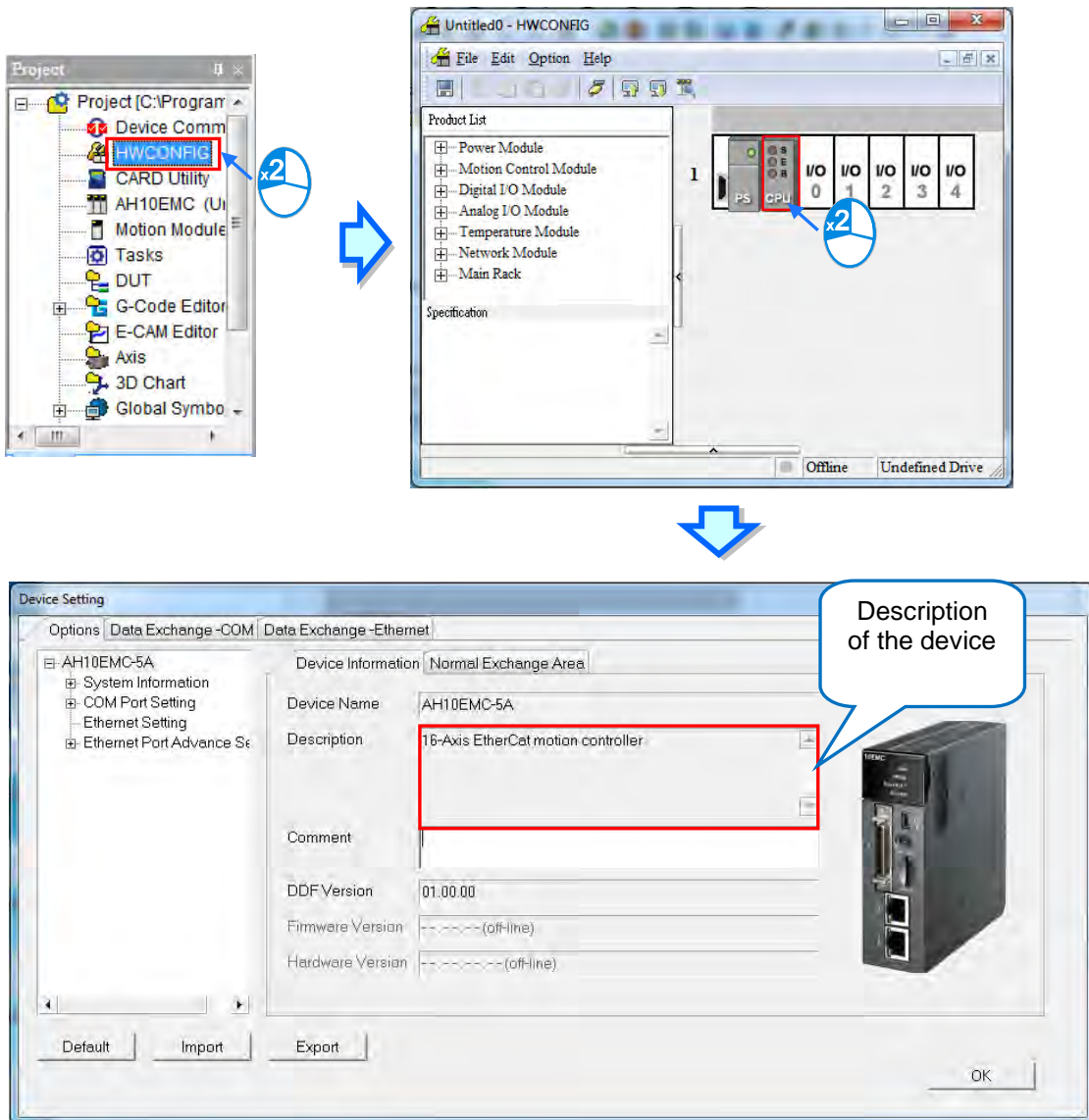
#### 3.4.1 Open the PLC CPU Parameter Setting Page

Double click the CPU icon on the system configuration area after HWCONFIG is started and the **PLC Parameter Setting** window will be opened automatically. The parameters to be set and window display may look different for different CPU models.

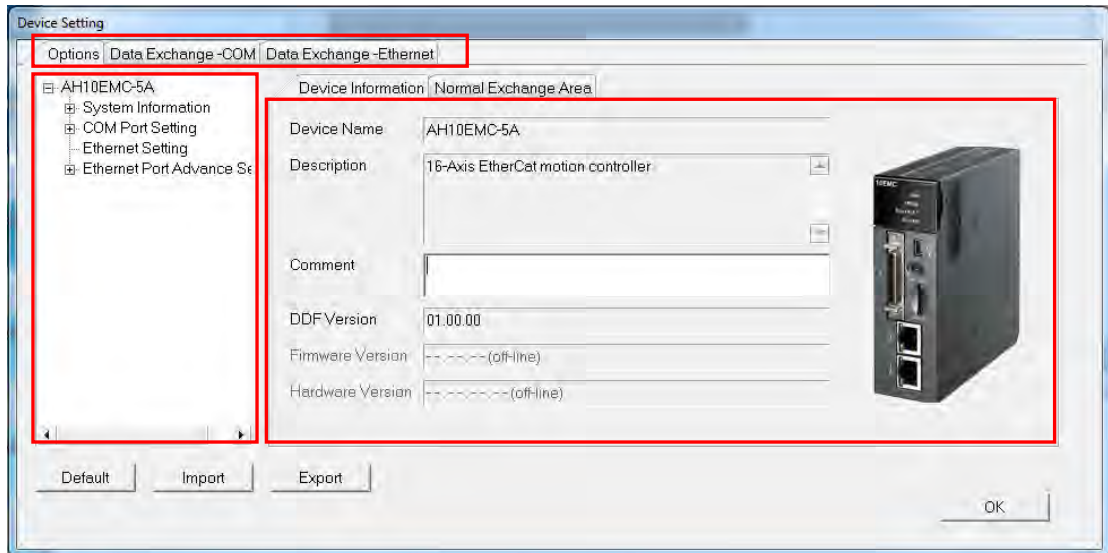
**⚠ Please refer to the operation manual for the CPU module before the parameters in a CPU module are set and learn about the impact of the set parameters on the CPU module and the entire system so as to prevent from the system and personnel damages.**

3

\*. Download the parameter settings to the PLC CPU. For ways to download, please refer to section 3.8.4.



The parameters are classified into several types. The users can click the primary tabs on the top of the window. Users should switch to different setting pages according to main types of parameters and their secondary tables. **Parameter name**, **Value**, **Unit**, **Default**, **Minimum** and **Maximum** are listed in the title bar in the right of the window.



3

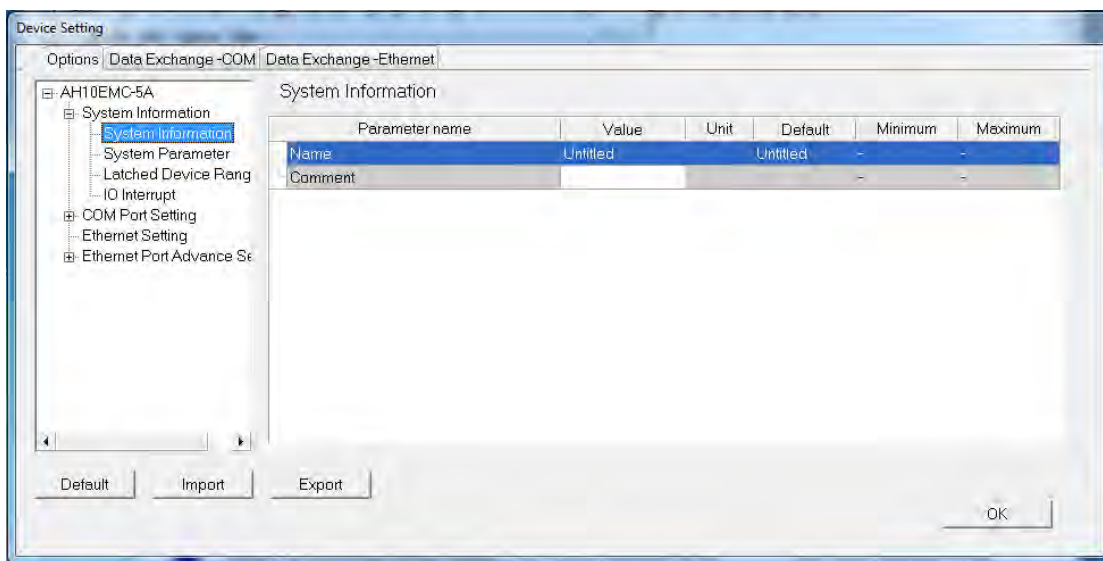
### 3.4.2 Options

#### 3.4.2.1 Options - System Information

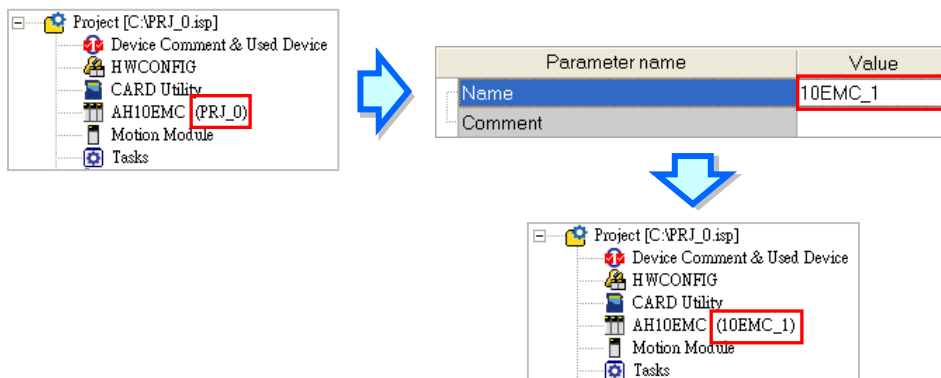
**Device Info, System Parameter, Latched Device Range and IO Interrupt** under **Device Setting**, click the **Options** tab to see these options.

There are **Name** box where maximum 31 characters can be input and **Comment** box where maximum 63 characters can be input on the **Device Setting** page. There is no special restriction on the input characters; special marks and spaces are allowed to use here. However, the users have to note that one Chinese character occupies two characters.

3



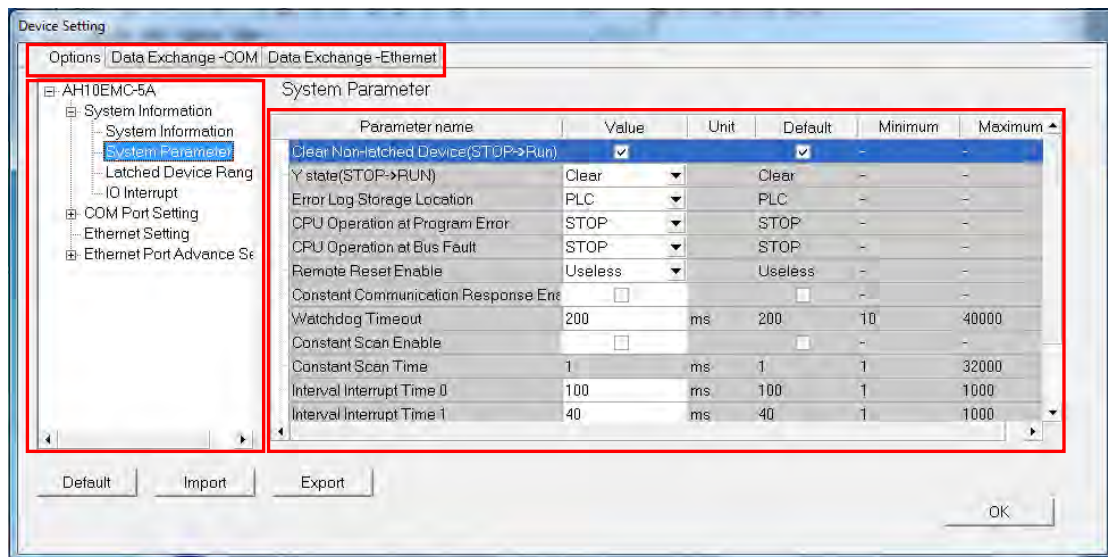
When adding an ISPSOft project, the system will use the project name as the PLC CPU name by default and appear next to the model name in Project section to be re-defined.



Users can identify a device by means of the name of the device. When several devices are connected to one network, users can check whether a device connected to the computer is the device they expect by checking the device name. To prevent errors, when users are downloading or uploading data or operating online, the system will check whether the label name of the PLC CPU

and the label name in the Project section is the same or not, when the name of the CPU module is different from the name attached to the model in the project management area.

The following image is the **System Parameter** setting page that provides drop-down lists to choose or box to input appropriate parameter values.



3

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

If the **Clear Non-latched Devices** checkbox is selected, the states of the non-latched devices and the values in the non-latched devices will be 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:** Set all Y device output to OFF.
- **Retain Present State:** Output retain to present state.
- **Previous State before STOP:** Restore Y device output to previous state before STOP.

- **Error Log Storage Location**

Users set the place where the error log is stored.

- **PLC:** If the **PLC** option button is selected, twenty error logs at most can be stored in an AHxxEMC series CPU module. If there are more than twenty error logs, the oldest error log will be overwritten

by the latest error log.

➤ **PLC and PLC CARD:** If the **PLC and PLC CARD** option button is selected, and there are more than twenty error logs, the oldest error log will be moved to the memory card before the overwritten.

- **CPU Operation during an occurrence of a Program Error**

If an error occurs in the program, the status of the CPU may change.

3

- **Stop:** CPU switches to the Stop state.
- **Continue:** CPU retains the state before any error occurs.

- **CPU Operation during an occurrence of a Bus Fault**

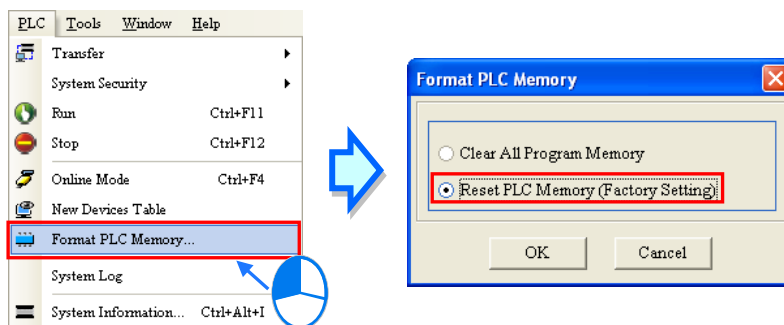
If an error occurs on the bus, the status of the CPU may change.

- **Stop:** CPU switches to the Stop state.
- **Continue:** CPU retains the state before any error occurs.

- **Remote Reset Enable**

The function is for setting whether to allow the remote system to reset the CPU. After users select the function 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 in ISPSOft.

- **Useless:** The function is not used.
- **Use:** The function is used.



- **Constant Communication Response Enable**

If this function is not enabled, commands received through the communication ports will not be processed until the scan cycle is complete. If this function is enabled, 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 this function is enabled. Please make sure that the operation of the system is not affected when this function is used.

- **Watchdog Timeout**

Users can set a timeout during which the program is scanned. If the scan procedure cannot be completed within the time set for some reason, the watchdog timeout occurs, and the CPU module will be in an erroneous condition.

- **Constant Scan Enable**

Users can select the **Constant Scan Enable** checkbox, and set a scan time. If the actual scan time is less than the 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.

- **Scan Time**

Users type the period of scan time in the box if the preceding constant scan is enabled. User should notice that the timeout error will occur once the CPU starts to run if the set scan time is larger than Watchdog Timeout value.

- **Interval Interrupt Time 0**

Users can set interval timers to trigger the interrupts. The function is used with Timer Interrupt 0 in the work management together. Please refer to chapter 5 for more information.

- **Interval Interrupt Time 1**

Users can set interval timers to trigger the interrupts. The function is used with Timer Interrupt 1 in the work management together. Please refer to chapter 5 for more information.

- **Interval Interrupt Time 2**

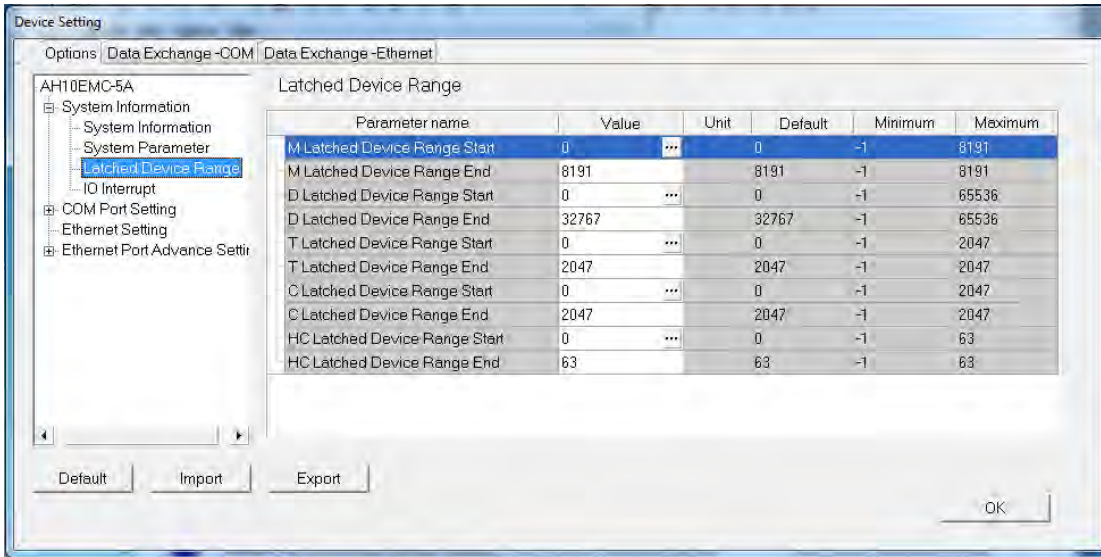
Users can set interval timers to trigger the interrupts. The function is used with Timer Interrupt 2 in the work management together. Please refer to chapter 5 for more information.

● **High-speed Scan**

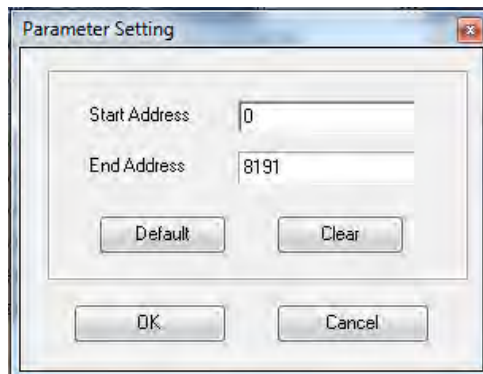
Users can set interval timers to trigger the interrupts. The function is used with Timer Interrupt 3 in the work management together. Please refer to chapter 5 for more information.

Users set a range of latched devices on tab **Latched Device Range**.

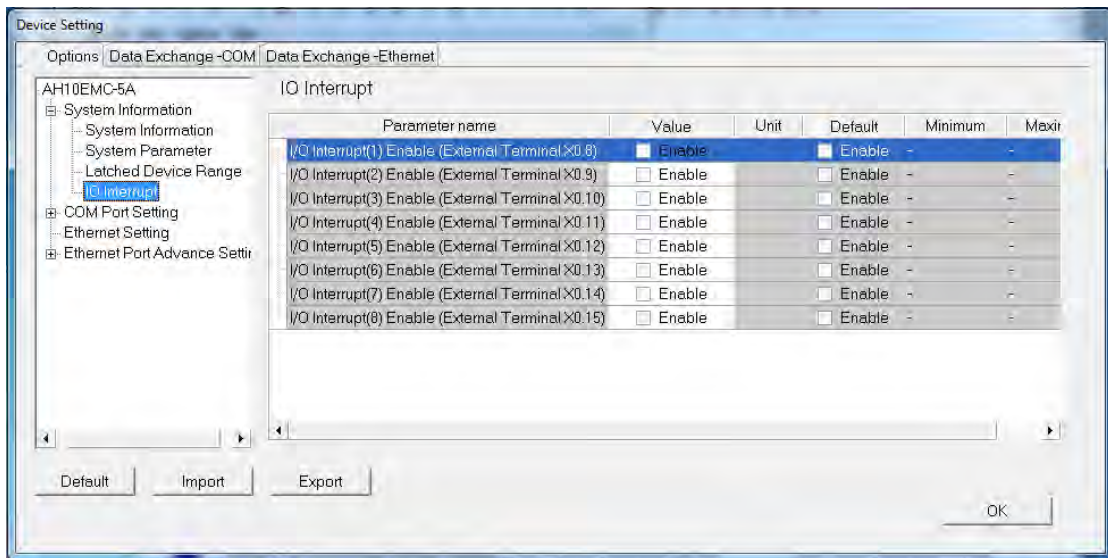
3



Click button **...** to open the setting window where users can set the start and end addresses of the device. If users need to use the function of non-latched area, users need to set up the start and end addresses of the device in the range of zero and the maximum. After the users click **Default**, the setting will be restored to the default values. After the users click **Clear**, the value in the box will be cleared to -1 and the function of the non-latched area will be disabled. Exit from the window by clicking OK to save the settings and Cancel to give up the settings.



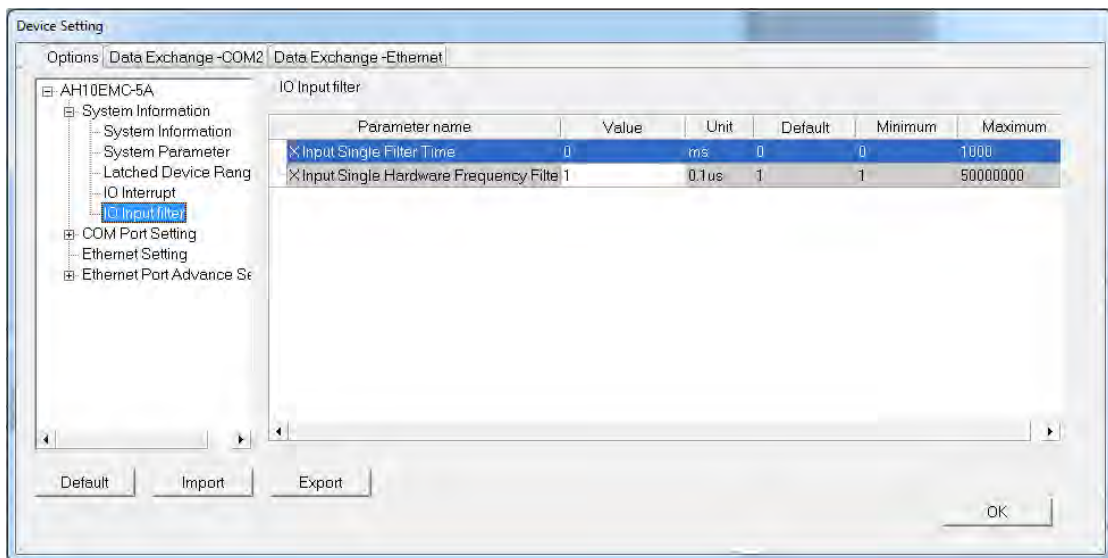
Tab **IO Interrupt** is for users to select X input points and enable the IO interrupt input function.



3

X0.8~X0.15 will match IO interrupt 1~ IO interrupt 8 separately. Please refer to chapter 5 for more information.

Tab **IO Input Filter** is for users to setup IO input filter to reject unwanted signals.





● **X Input Single Filter Time**

Users can set up the response time for pulse input, the setting range is 0~1000, unit is ms. Any response time below the set-up value will be filtered.

● **X Input Single Hardware Frequency Filter Time**

Users can set up the value to be lower than the time of X input Duty On, the setting range is 1~50000000, unit is 0.1us. Base on the time (t) and frequency filter equation (f) \* =

3

$\frac{1}{2 \text{ Filter Time (t)}}$  (Hz), if the frequency input signal is higher than the (f) \* value, the input signal will be

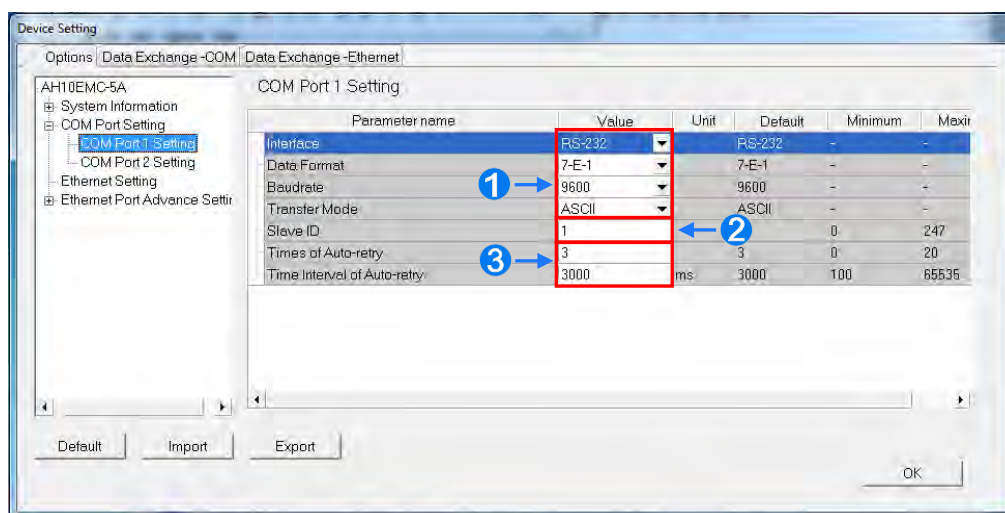
filtered. (For more information on the time of X input Duty On, please refer to **AH Motion Controller** operation manual.)

\*. About IO Input Filter, please refer to AH Motion Controller operation manual for more information.

### 3.4.2.2 Option - COM Port Setting

AHxxEMC is equipped with 2 COM ports. Users could set COM1 and COM2 parameters individually by clicking **COM Port 1 Setting** or **COM Port 2 Setting** under COM Port Setting. COM1 is for uploading and downloading; COM2 is for Data Exchange.

The setting items are same for the two COM ports.



- ❶ Set the communication protocol of the COM port.
- ❷ Users can set a station address at the communication port. 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 cannot be the same as the station address of another device on the same network.

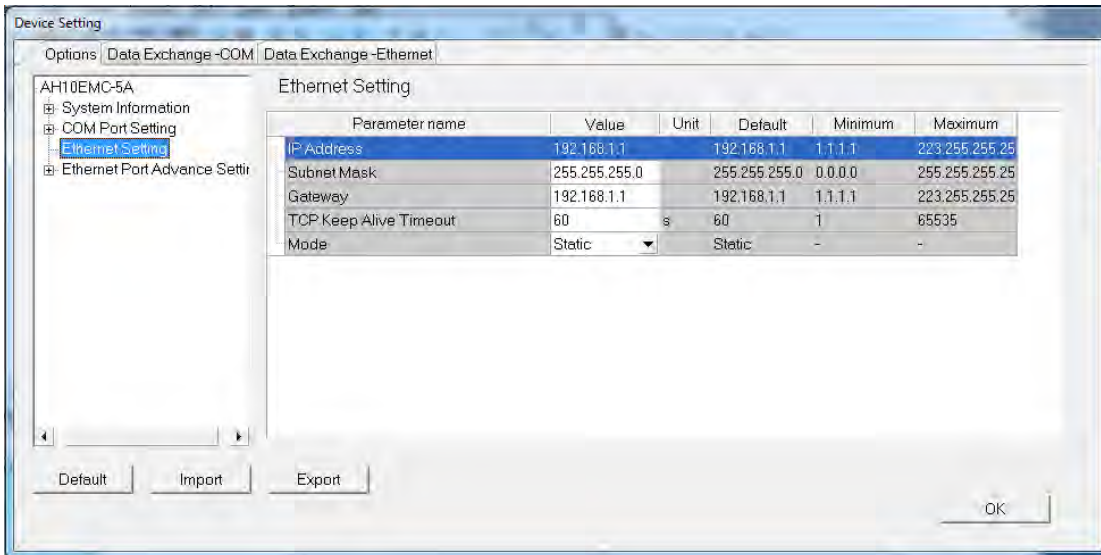
If the communication port functions as a slave, and there are other slaves, the station address of the communication port cannot be 0. Station address 0 has the meaning of broadcasting to all slaves in a communication protocol. If a master specifies in a data packet that data must be sent to station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

- ❸ If the sending of a command fails, the CPU module will retry the sending of the command. Users can set the number of times for sending a command in the box of **Times of Auto-retry** and set the time interval to retry the sending of a command in the box of **Time Interval of Auto-retry** when the communication fails. PLC module will make a next connection with the network every the set interval time when the communication fails.

### 3.4.2.3 Option- Ethernet Setting

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.

3



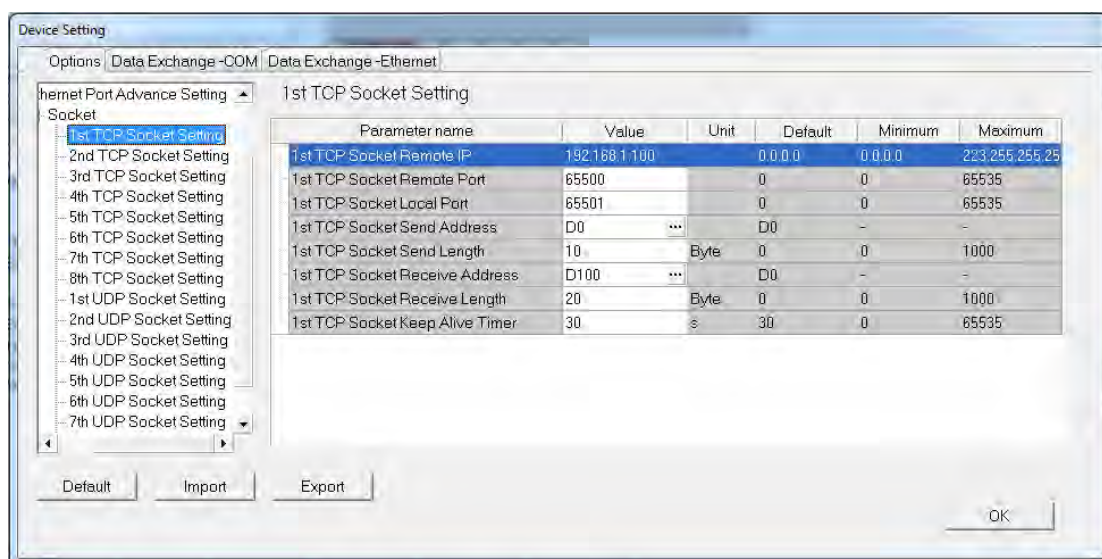
Users can set an IP addressing mode for the CPU module. If users select **Static** in the **IP Addressing Mode** drop-down list box, they can specify an IP address. If **Dynamic** is selected in the **IP Addressing Mode** drop-down list box, an IP address is assigned dynamically.

### 3.4.2.4 Option- Ethernet Setting (advanced)

On the Ethernet Port Advance Setting window, users can set the parameters for sockets through which data is transmitted. In the **Socket** table, users can set COM port parameters for data transmission via Ethernet. However, the function should be applied with specific API instructions. Please refer to AHxxEMC Series Instruction Manuals for more information. Sockets will not be introduced here. Please refer to related documents or manuals for more information about sockets.

The system for AHXXEMC series modules supports the data transmission between the CPU module and other CPU module or device via a socket as well as TCP and UDP protocols in each of which 8 groups of connections can be set. Users should select **Socket Function Enable** checkbox before any setup and then do other setting in TCP and UDP setting pages respectively.

3



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

- **Remote IP:** Users can set a remote IP address.
- **Remote Port:** 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.
- **Local Port:** 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.
- **Send Address:** Users can set the length of data which will be sent by the local CPU module.
- **Send Length:** Users can set an initial device in the CPU module where data which will be sent is stored. The length must be within the range between 0 and 1000 bytes.

- **Receive Address:** Users can set the length of data which will be received by the local CPU module.
- **Receive Length:** Users can set an initial device in the CPU module where data which will be received is stored. The length must be within the range between 0 and 1000 bytes.
- **Keep Alive Timer:** Users can set a maximum keep alive time for the connection. If no data is transmitted, and the keep alive period has elapsed, the CPU module will terminate the connection automatically.

### 3

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

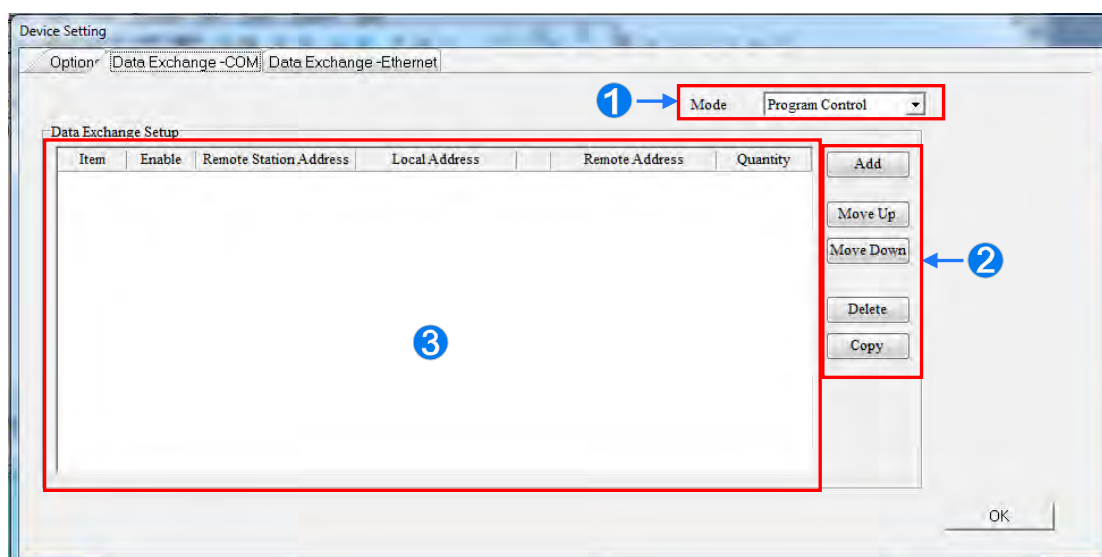
If the local CPU module wants to send 10-word data to the remote device, the data will be stored in D0~D9 before the data is sent. If the local CPU module receives 20-word data from the remote device, the data will be stored in D100~D119.

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 30 seconds have elapsed, the CPU module will close the socket, and terminate the TCP connection.

### 3.4.3 Data Exchange - COM2 & Ethernet

After users click the tab Data Exchange - COM and Data Exchange – Ethernet, the parameters for making the data exchange table of the communication port built in AHxxEMC series can be set.

AHxxEMC series as a master exchanges data with remote devices working as slaves via COM2 based on MODBUS protocol or via Ethernet port based on MODBUS TCP protocol.



- ❶ As one start mode, **Program Control** decides whether the set data exchange is performed. **PLC Run** means that the set data exchange will be performed automatically when PLC is in RUN state. **Always Enable** means that the data exchange will be performed constantly when PLC is powered on.
- ❷ A new data exchange table will be added in area ❸ by clicking **Add**. The present data exchange table selected will be moved up or down by clicking **Move Up** and **Move Down**. The present data exchange table selected will be deleted by clicking **Delete**. The present data exchange table selected will be copied by clicking **Copy** and then can be pasted to the last row.
- ❸ The data exchange table is explained as below.

Name	Description
Item	The number of the block for data exchange
Enable	To select whether to enable the data exchange table when the data exchange is performed.
Remote Station Address	The slave station address of the data exchange table, One address can be set for multiple data exchange tables. It is a station address for COM2 tab and IP for Ethernet tab.
Local Address	The device address range used by the master in the data exchange table

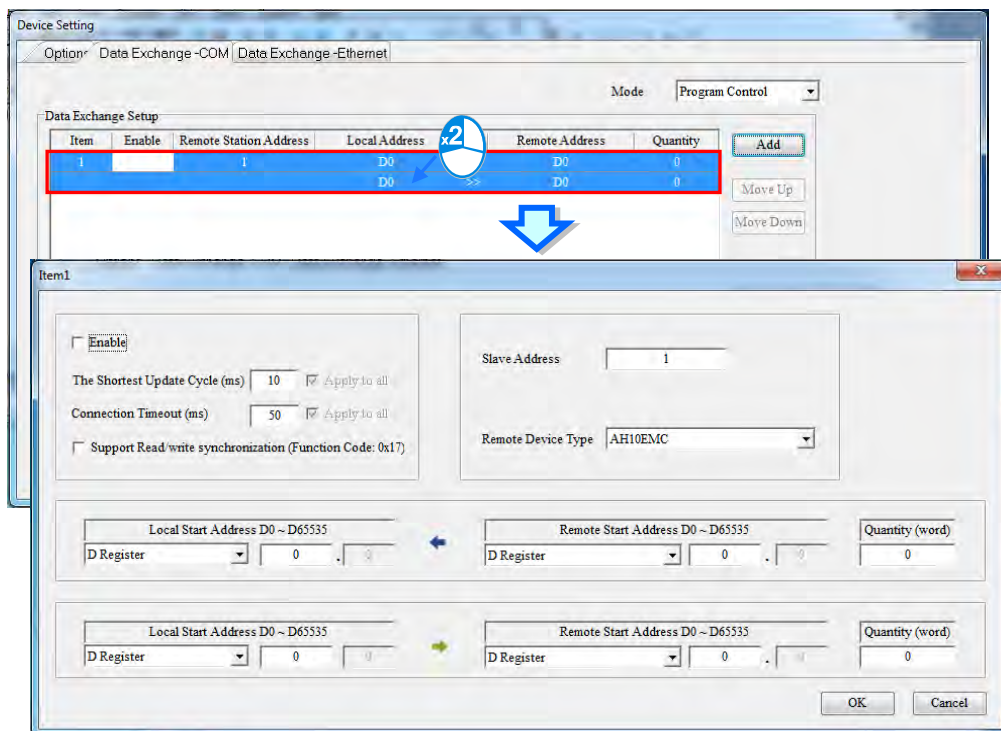
<< / >>	[<<] : Input; the data block where the master read a slave [>>] : Output; the data block where the master write to a slave
Remote Address	The device address range used by a slave in the data exchange table The device range will be in hexadecimal display, if the slave in the data exchange table is a user-defined MODBUS Device.
Quantity	The size of the data exchange table, which is basically consistent with the result calculated from the device range.

# 3

### 3.4.3.1 Data Exchange - COM2

On the page of tab **Data Exchange - COM2**, double click a data exchange table which is to be set to open **Item** window as below. Select the checkbox of **Enable** to enable the data exchange table in the mode mentioned above. **Remote Address** is the address of the target slave for data exchange. **The Shortest Update Cycle** is the period for data exchange of the data exchange table. If the target device does not make any response within the time specified by **Connection Timeout**, that is timeout. The master can complete the read and write in a command to improve the efficiency of data exchange by selecting **Support Read/write synchronization (Function code 0x17)** in which the specific MODBUS function code is used. Please make sure that all devices in data exchange support MODBUS function code for read and write synchronization. Otherwise, the read and write will fail on account of the failure to identify the function codes after the command from the master is received. **Device Type** is the model of the target slave including Delta PLC and standard MODBUS devices.

3



- **Read**

AHxxEMC series read data from a remote device.

**Local Start Address:** Device type and start address of devices where AHxxEMC series store data;

**Remote Start Address:** Device type and start address of the remote device to be read;

**Quantity:** Data length of input



- **Write**

AHxxEMC series write data to a remote device.

**Local Start Address:** Device type and start address of the source data of AHxxEMC series;

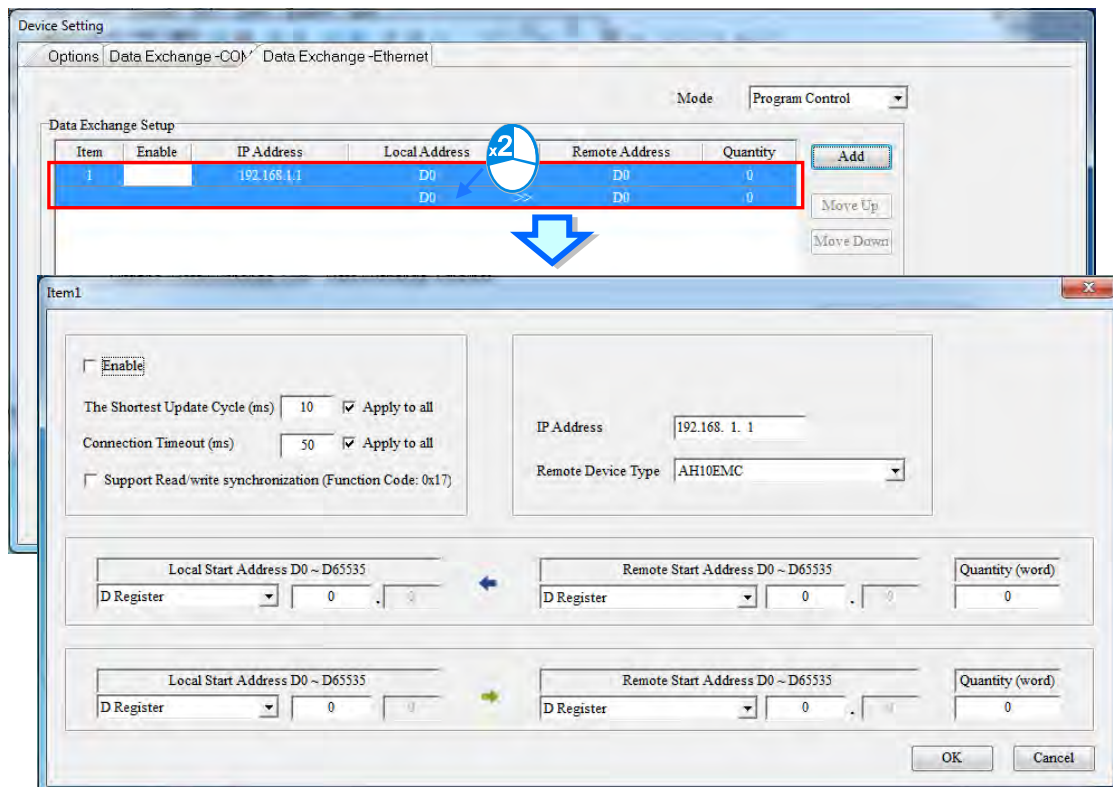
**Remote Start Address:** Device type and start address of the remote device where data are to be written;

**Quantity:** Data length of output



### 3.4.3.2 Data Exchange - Ethernet

On the page of tab **Data Exchange - Ethernet**, double click the data exchange table to be set so as to open the **Item** window as below. The settings are almost same as that for **Data Exchange - COM2** except the target slave model option in Device Type which contains Delta PLC and standard MODBUS TCP equipment as well as IP address for typing the IP of remote slaves. For other settings, refer to the content of **Data Exchange - COM2**.



Select the check box of **Apply to all** to the right side of **The Shortest Update Cycle** and **Connection Timeout** and then click **OK**. The settings for **The Shortest Update Cycle** and **Connection Timeout** will be written to other data tables on the tab Data Exchange- Ethernet. Users could set data exchange tables separately without selecting the check box of **Apply to all**.

## 3.5 Parameter Setting for DVP Series PLC

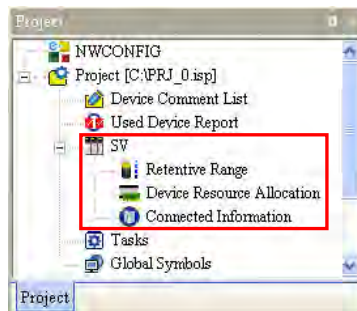
\*. The old DVP series do not support HWCONFIG; however, the new DVP-ES3 supports HWCONFIG. Regarding parameter setting for DVP-ES3, please refer to section 3.5.4.

### 3.5.1 System Management Tools for DVP Series PLC

In the project for DVP series, users can find system management tools under the category of the PLC hosts name where some of the functions need to connect with the host for operation. The functions are described in the following:

3

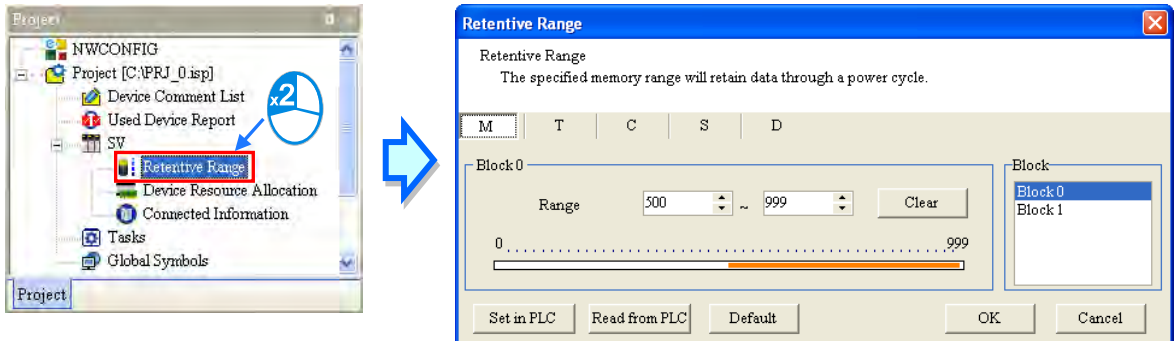
- **Retentive Range:** Set the PLC retentive device range.
- **Device Resource Allocation:** Set the device range for auto-configuring symbol variables.
- **Connected Information:** View the current PLC host system setting and parameters.



Apart from **Device Resource Allocation** function to be introduced in section 6.2.11, the **Retentive Range** and **Connected Information** function will be explained in the following sections.

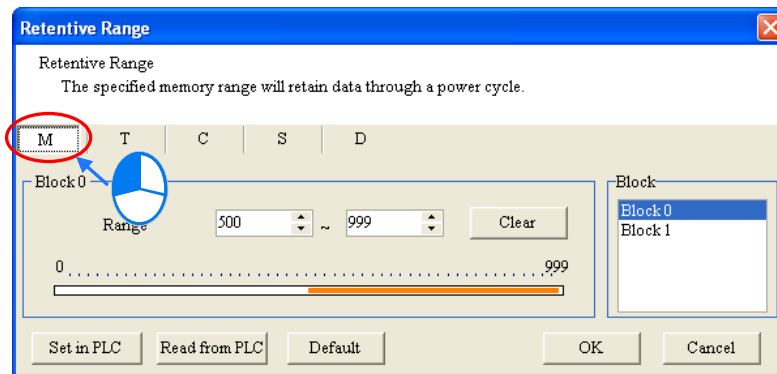
### 3.5.2 Retentive Range

- (1) Double-click the **Retentive Range** and its setting window will appear.

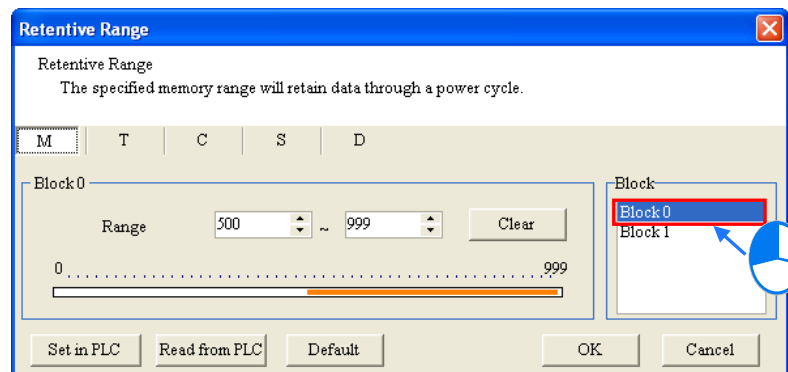


3

- (2) Click on a desired device type for setup.

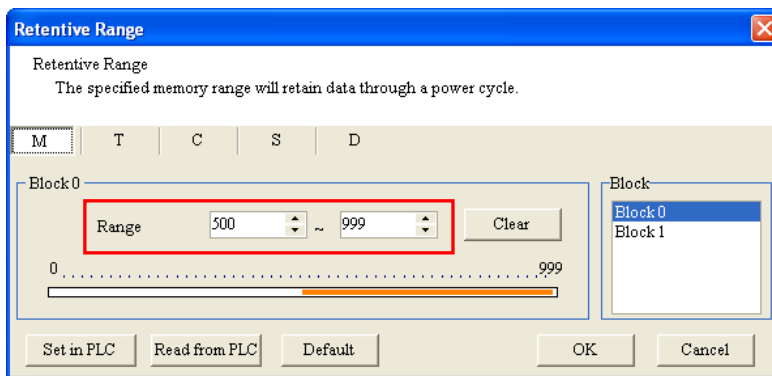


- (3) Since the range for PLC retentive block may be discontinuous, therefore, please first select the desired setting section.

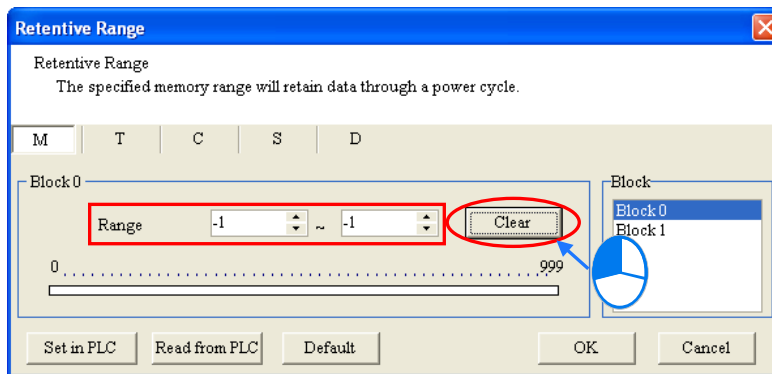


- (4) Users will see a strip graph in the block section, with two values on both ends representing the setting range limit. The orange color in the strip graph refers to the existed retentive range; however, when the Block 0 section shows grey content, this means the current section cannot setup retentive range for the selected PLC model.

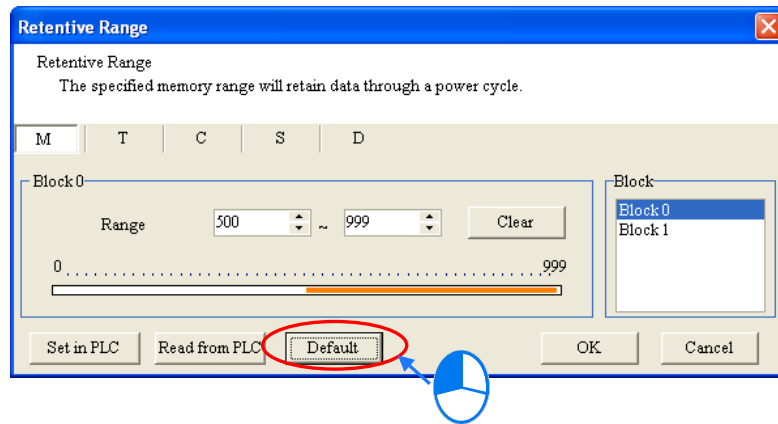
For the setup, users need to input values of the **Retentive Range**, where the beginning point is the value on the left box and the end point is the value on the right box. When input values, the beginning point on the left must be smaller than the end point on the right and cannot exceed the strip graph appearing as the range limit.



- (5) Click **Clear** to remove the retentive function for the devices in that block and the range values will be from -1 to -1.

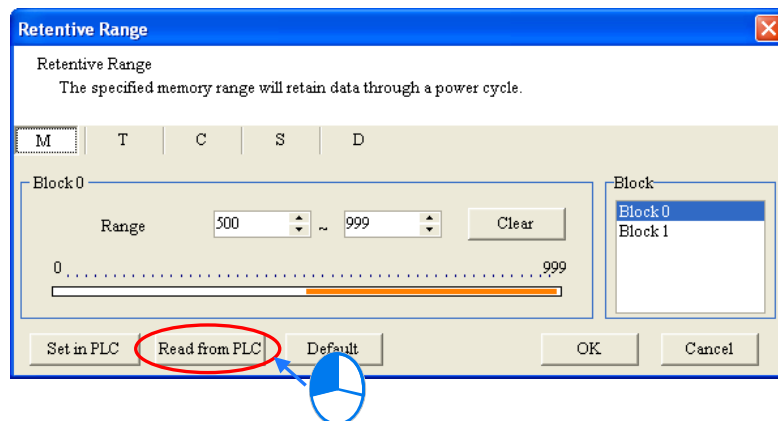


- (6) Click **Default** for settings to return to default values and the retentive function for all devices will return to defaults.

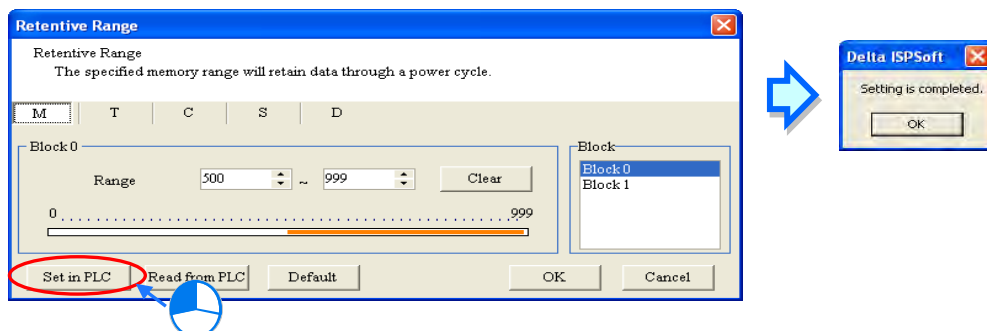


3

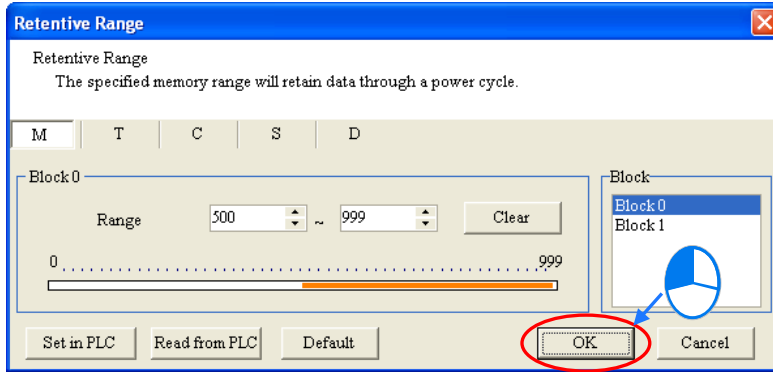
- (7) When ISPSoft is connected to a PLC host, click **Read from PLC** to read the retentive setups on all devices from the PLC host.



- (8) When all setting is complete and ISPSoft is in connection with a PLC host, click **Set in PLC** to write the retentive settings for all devices into PLC hosts. Meanwhile, the system will automatically close the setting window and the setups are stored, but it will be saved once an ISPSoft project is saved.

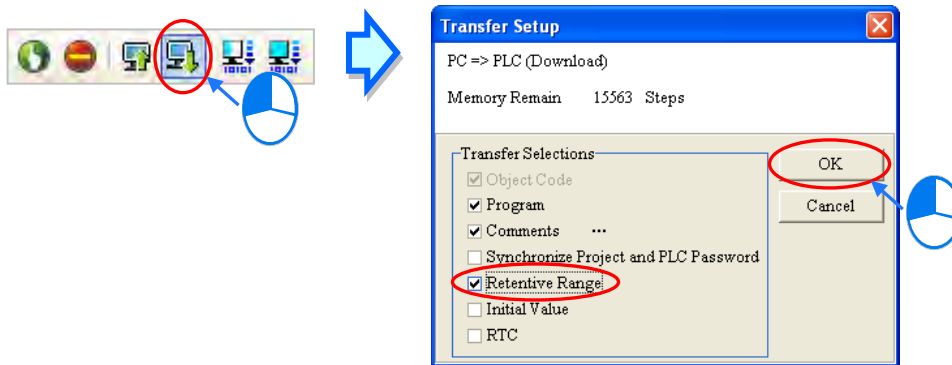


- (9) When setting values are not required temporarily, click **OK** to apply the setting. The setup is saved once an ISPSOft project is saved.



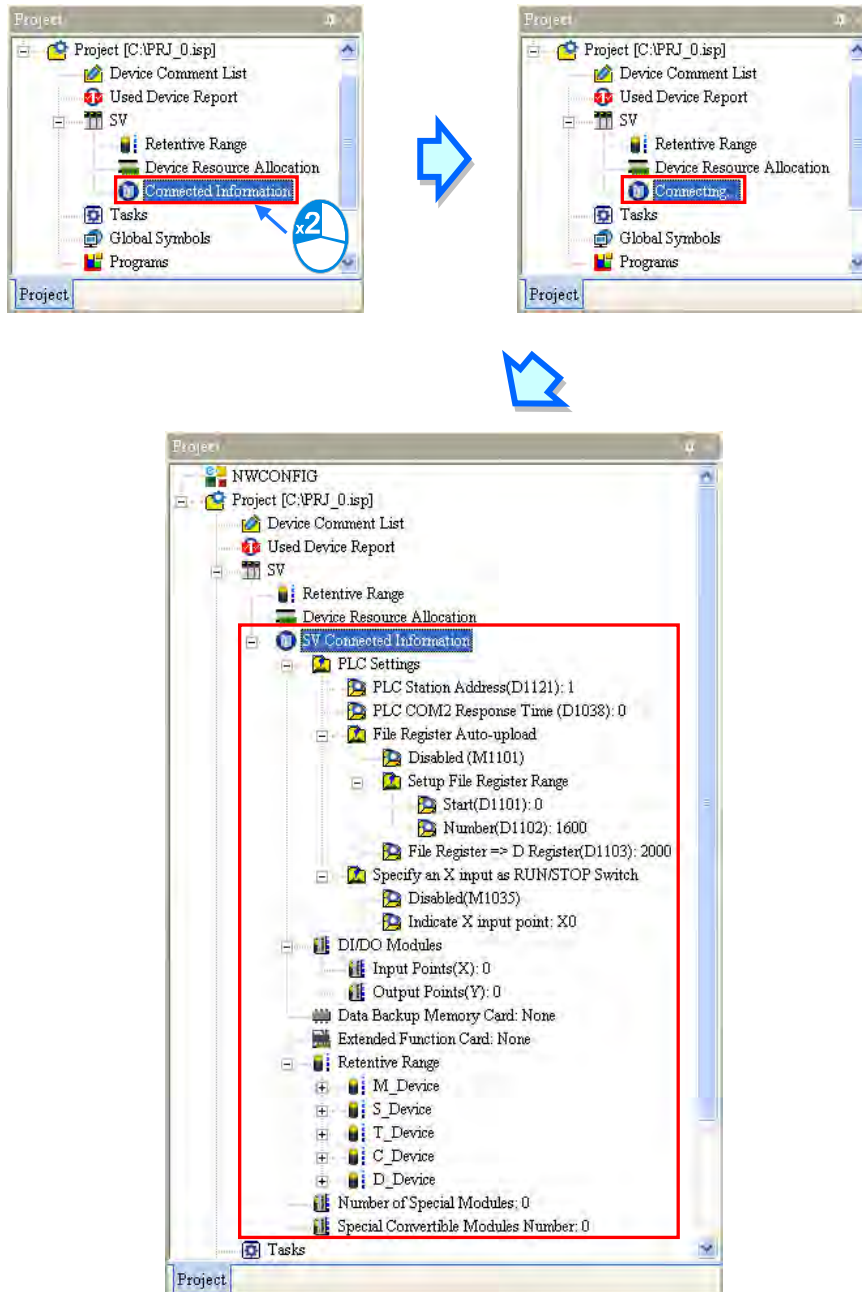
3

- (10) Besides using the setting window to write the setting values of functions into PLC hosts, the retentive range in a project setting can also be written into PLC hosts when project program is downloaded. Users only need to select **Retentive Range from the Transfer Setup** window and click **OK** to download data to PLC hosts. For more details on download, please refer to section 17.1.3.



### 3.5.3 Connected Information

When ISPSOft is in connection with a PLC host, users can double-click **Connected Information** from the Project section, with PLC system parameters been captured and placed under Connected Information file for viewing only.



3

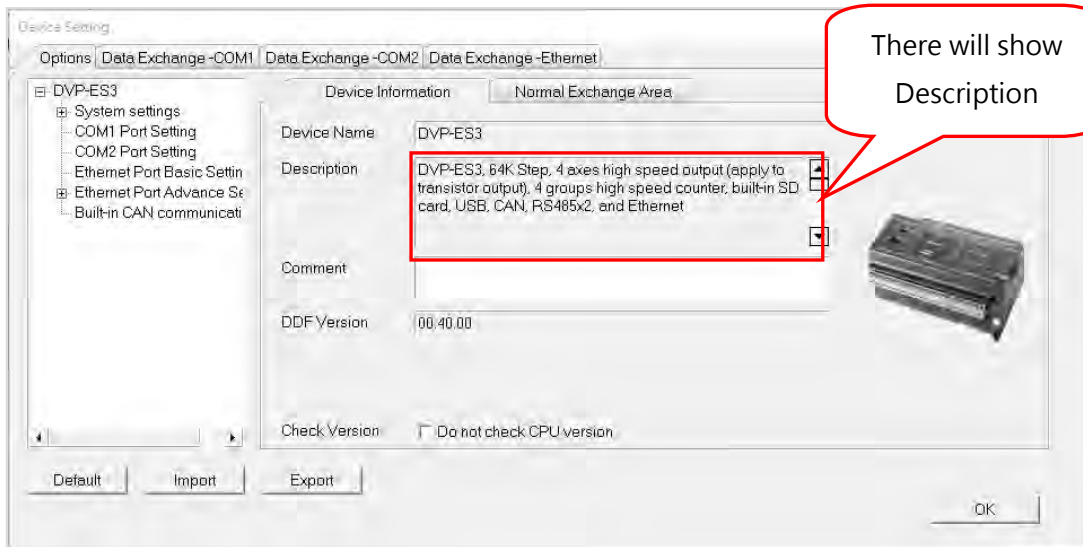


### 3.5.4 Parameter Setting for DVP-ES3 Series PLC

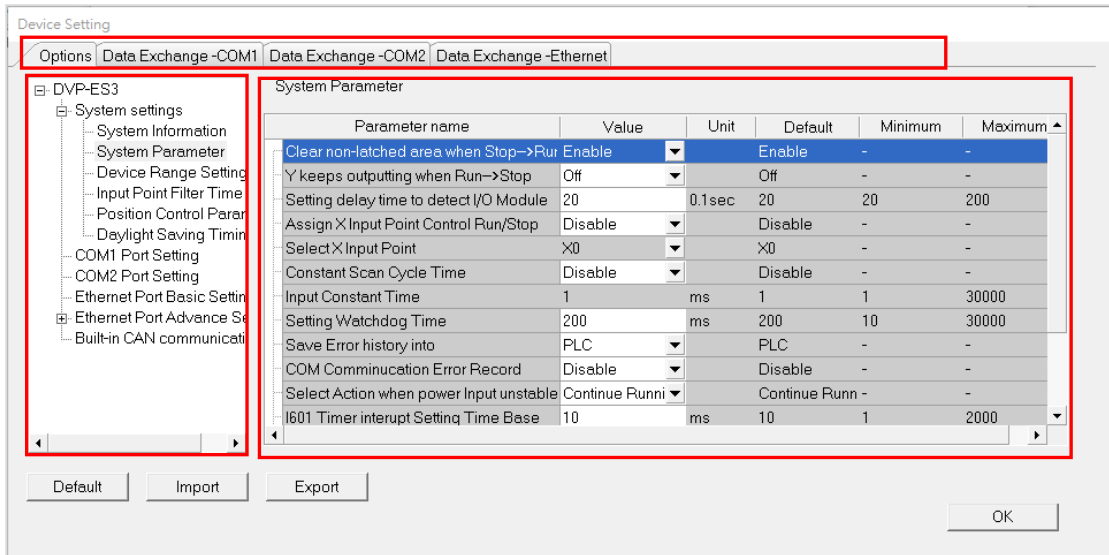
Double-click **HWCONFIG** in the Project section and choose **Parameter Setting** to open the setting page of the host CPU.

**!** Before setting PLC parameters, please check the user manuals for all types of PLCs to avoid any personal loss or system damages due to the effects of the parameter setting for PLCs and the whole system.

3

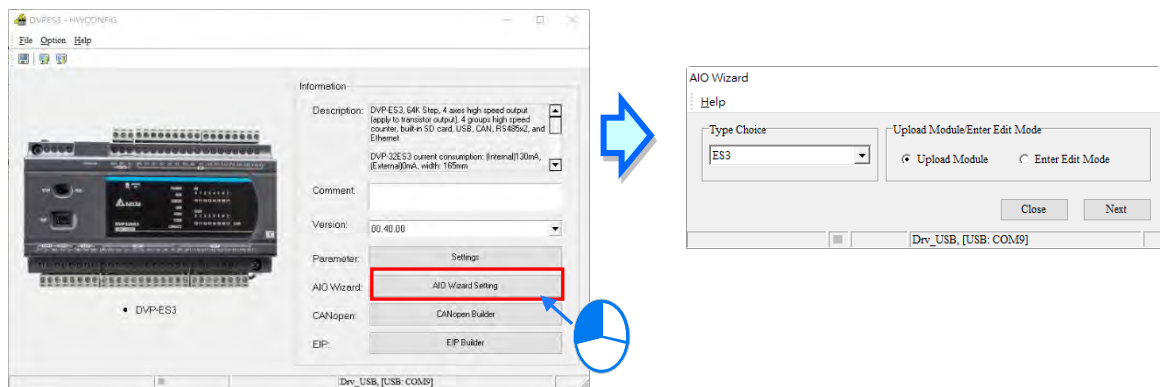


In the **Device Setting** window, parameters are categorized into many pages and users can use the **main category labels** on the top of the window or select from the list of items in hierarchy on the left. On the right, the titles on the page (from left to right) are **Parameter Name**, **Value** for user input, **Unit** of the setting value as well as **Default**, **Minimum** and **Maximum** value. For more details, please refer to section 3.5.2.



3

DVP-ES3 series requires AIO wizard setting for configuring modules on the right. These parameters also determines the actual function and characteristics of each modules. For module parameter settings, please click **AIO Wizard Setting** to open its setup window and select **Upload Module** to upload back the configuration through PLC or select **Enter Edit Mode** to create modules manually as shown below.



## 3.6 Parameter Settings for DVPxxMC Series PLC

### 3.6.1 Opening the PLC Parameter Setting Window

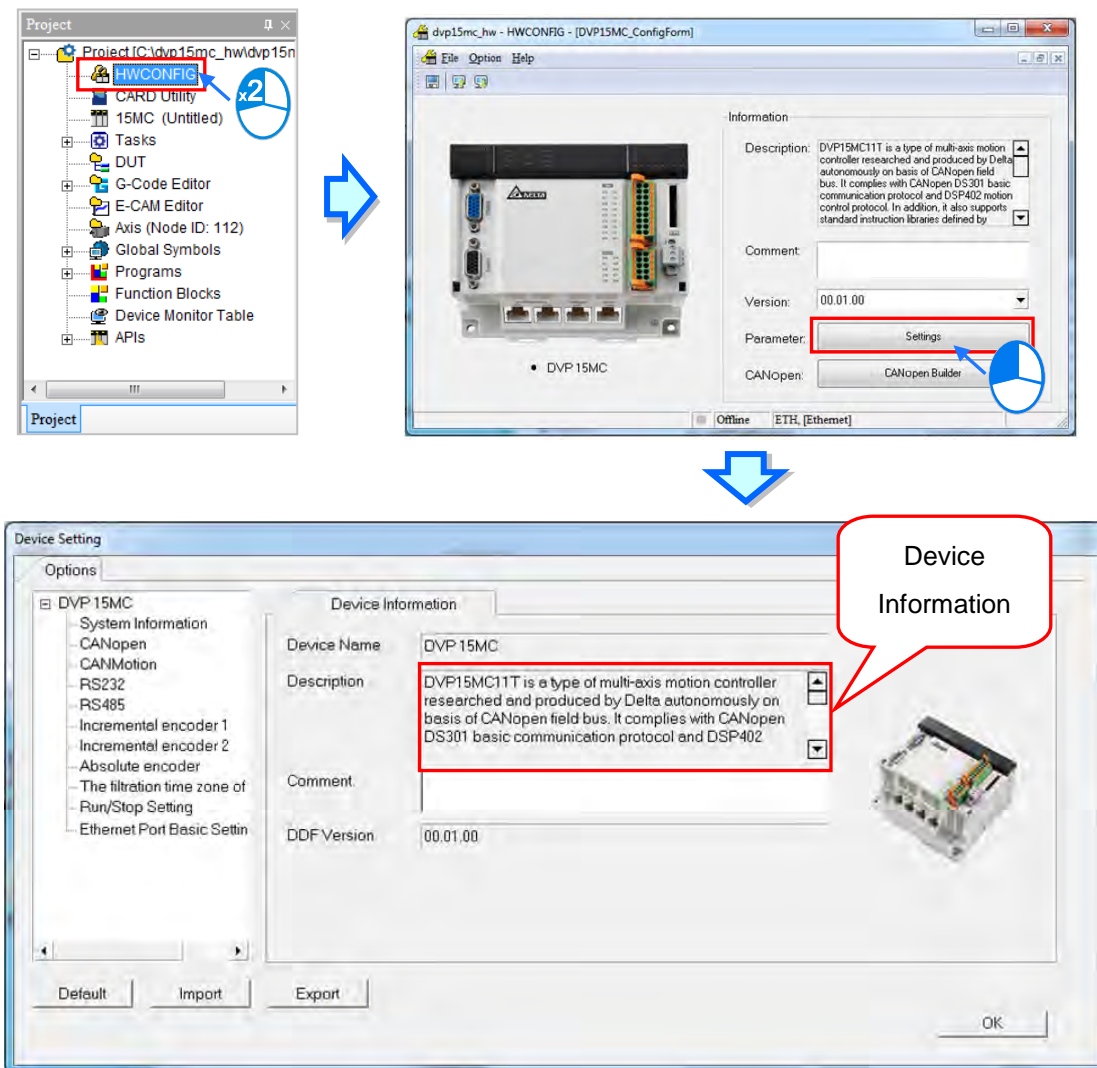
After double-clicking HWCONFIG in the system configuration area, users can select Setting from the Parameter field and the **Device Setting** window will appear. The parameters which can be set vary with the models or hardware of the CPU modules.

**⚠ Before setting the parameters in a CPU module, users have to refer to the operation**

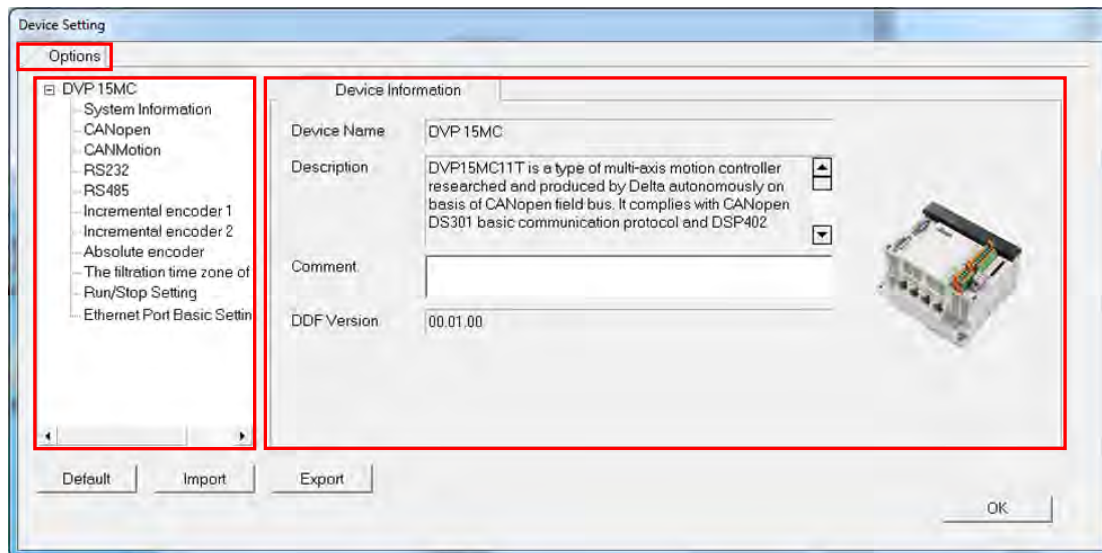
3

**manual for the CPU module and to make sure the effect of the setting parameters have on the CPU module and the whole system is safe, so as to avoid system damage or staff injuries.**

\*. The parameter settings are not effective until they are downloaded to the CPU module. Refer to section 3.8.4 for the download method.



The parameters are classified into several tabs on the setting page. Users can click the primary tabs on the top of the window and can switch to other setting pages base on different levels of settings. On the right, a table regarding parameter name as well as corresponding value is provided for users to set its unit, default, maximum and minimum.



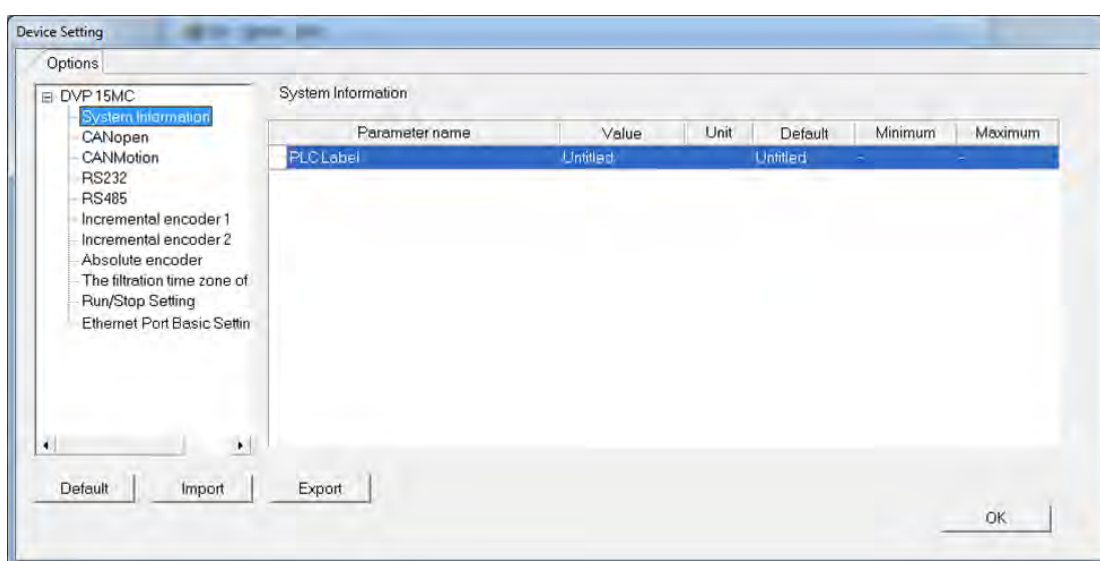
3

### 3.6.2 Options

#### 3.6.2.1 Options - System Settings

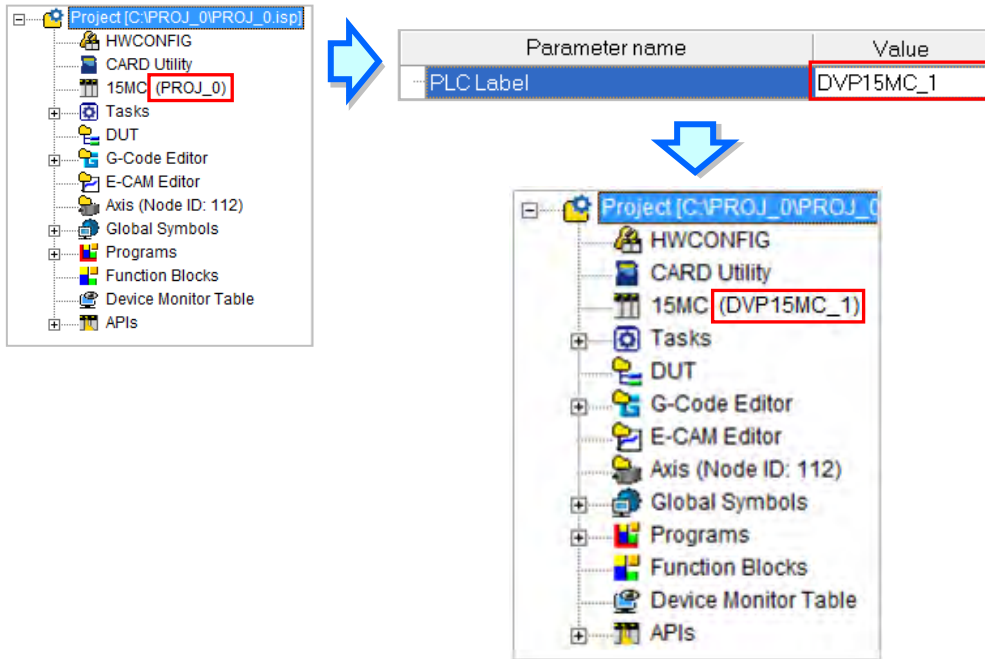
The **System Information** page under **Device Settings** appear after users click the **Options** tab.

There are only **PLC Label** box where maximum 15 characters can be input on the page. There is no special restriction on the input characters; special marks and spaces are all allowed to use here. However, the users have to note 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 **Name** box later.

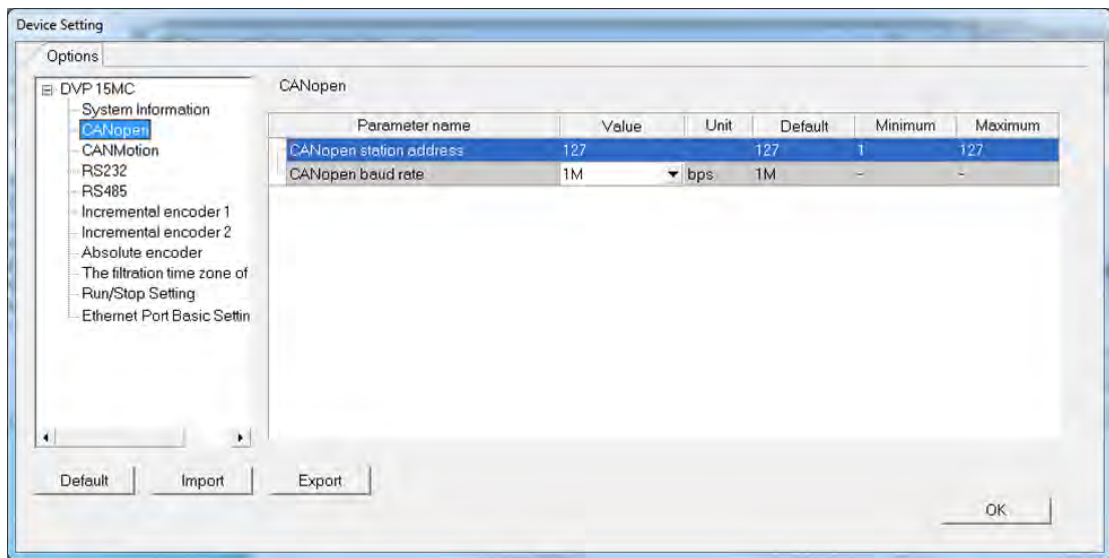
3



Users can identify a device by means of the name of the device. When several devices are connected to one network, users can check whether a device connected to the computer is the device they expect by checking the device name. To prevent errors, if users want to download/upload the program, 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, when the name of the CPU module is different from the name attached to the model in the project management area.

### 3.6.2.2 Options - CANopen

The parameters for CANopen are listed in the setting window below and users can set these values in the input box or via the drop-down menu.



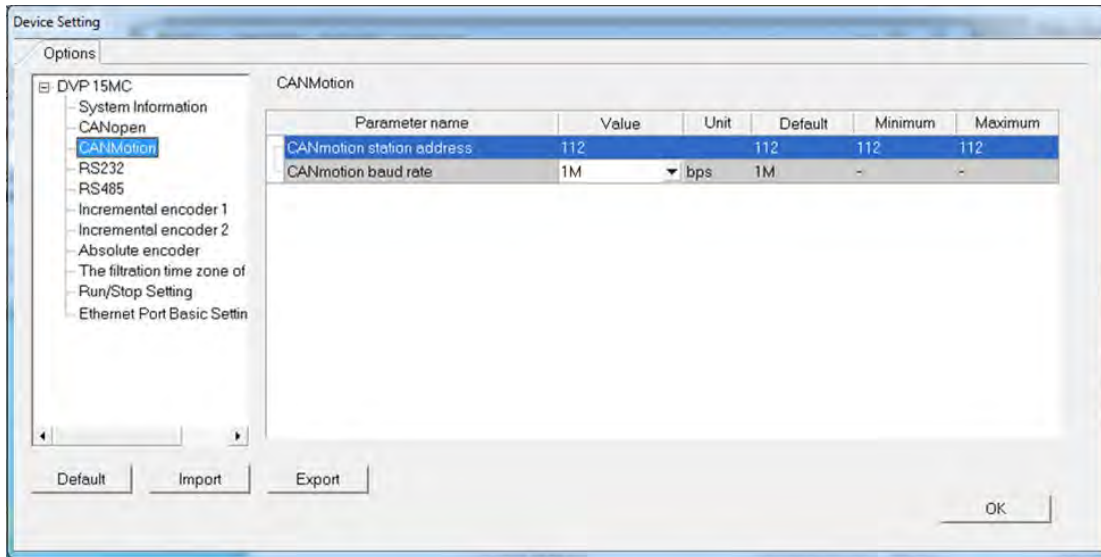
3

Parameters for setting include CANopen station address and CANopen baud rate. Users can select CANopen baud rate value such as 20k bps, 50k bps, 125k bps, 250k bps, 500k bps, 800k bps and 1Mbps.

### 3.6.2.3 Options - CANmotion

The parameters for CANmotion are listed in the setting window below and users can set these values in the input box or via the drop-down menu.

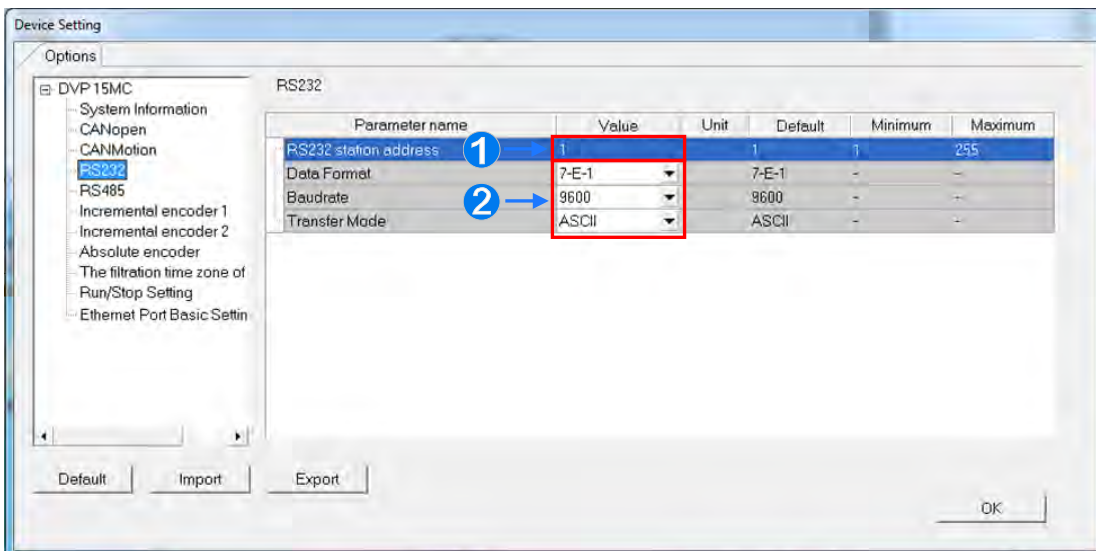
3



Parameters like CANmotion station address is set up with a default value of 112 and cannot be changed; users can select either 1Mbps or 500kbps for CANmotion baud rate.

### 3.6.2.4 Options - RS232

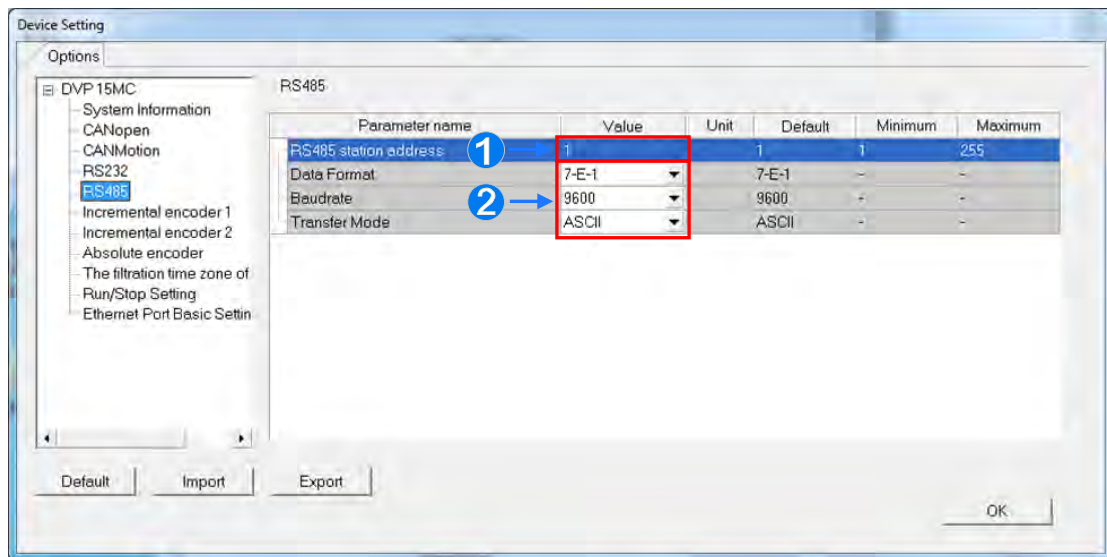
The settings of RS232 station are listed in the window below.



- ❶ Users can set up RS232 station address
- ❷ Select values of RS232 data format, baud rate and transfer mode. Users can select baud rate including 9600, 19200, 38400, 57600 and 115200.

### 3.6.2.5 Options - RS485

The settings of RS485 station are listed in the window below.



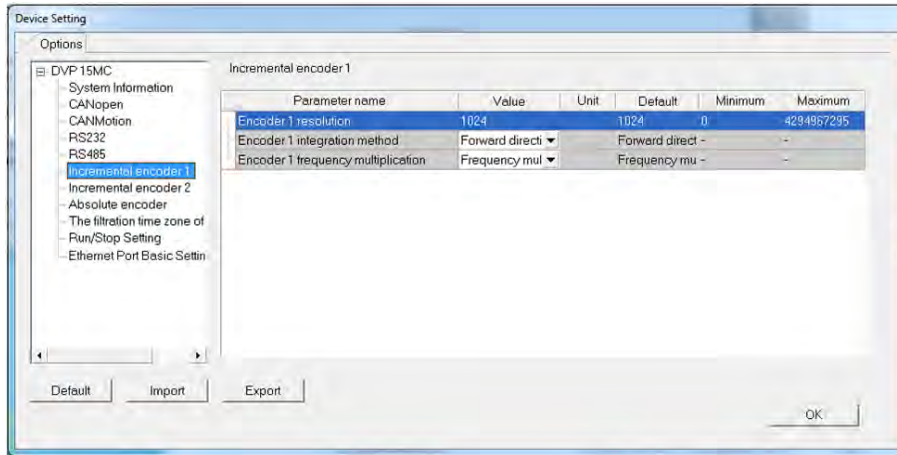
- ① Users can set up RS485 station address
- ② Select values of RS485 data format, baud rate and transfer mode. Users can select baud rate including 9600, 19200, 38400, 57600 and 115200.



### 3.6.2.6 Options - Incremental Encoder 1

The page below is for the first set of device settings.

3



Parameters settings include resolution, integration method and frequency multiplication.

- **Incremental Encoder 1 Resolution**

The default value of incremental encoder 1 resolution is 1024.

- **Incremental Encoder 1 Integration Method**

The settings include forward and reverse direction.

- **Incremental Encoder 1 Frequency Multiplication**

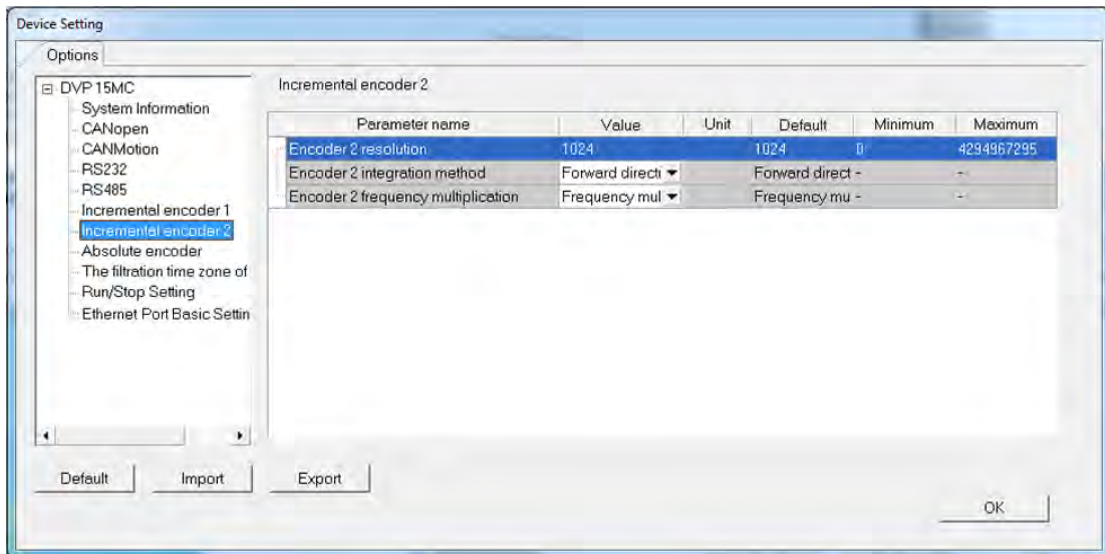
The settings include one/ two/ four times frequency multiplication.

The integration method and frequency multiplication is explained in the following table.

Integration Method	Meaning	Counting Method
One time frequency multiplication in forward direction	The counting starts when phase B rising edge is detected.	Increase the count
Two times frequency multiplication in forward direction	The counting starts when phase B rising and falling edge is detected.	Increase the count
Four times frequency multiplication in forward direction	The counting starts when the rising and falling edge of phase A and B are both detected.	Increase the count
One time frequency multiplication in reverse direction	The counting starts when phase B rising edge is detected.	Decrease the count
Two times frequency multiplication in reverse direction	The counting starts when phase B rising and falling edge is detected.	Decrease the count
Four times frequency multiplication in reverse direction	The counting starts when the rising and falling edge of phase A and B are both detected.	Decrease the count

### 3.6.2.7 Options - Incremental Encoder 2

The page below is for the second set of device settings.



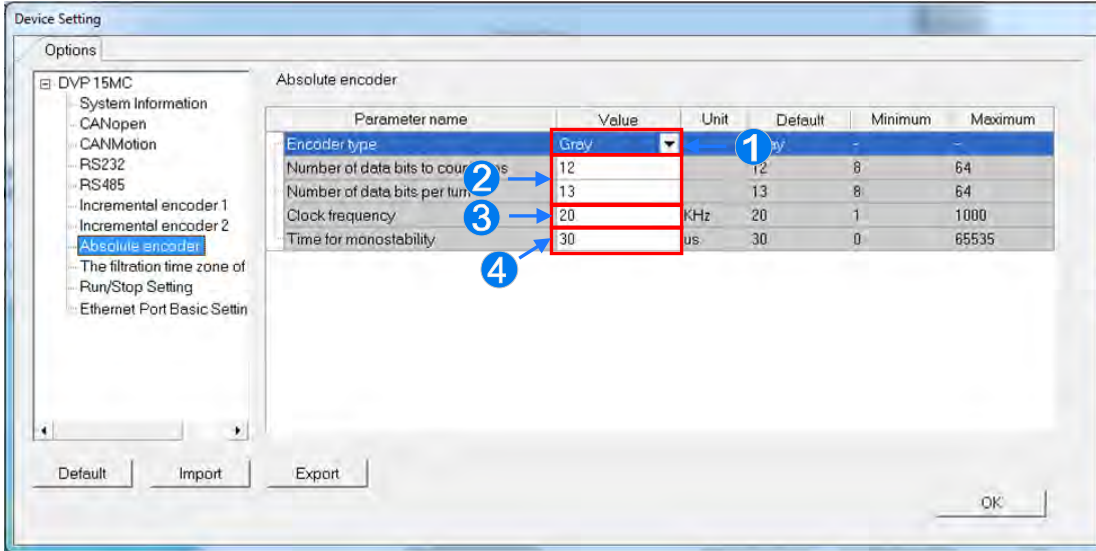
Parameters settings including resolution, integration method and frequency multiplication is the same as explained in incremental encoder 1 page. Please refer to section 3.6.2.6 for more information.

3

### 3.6.2.8 Options - Absolute Encoder

The page below is for setting the absolute encoder.

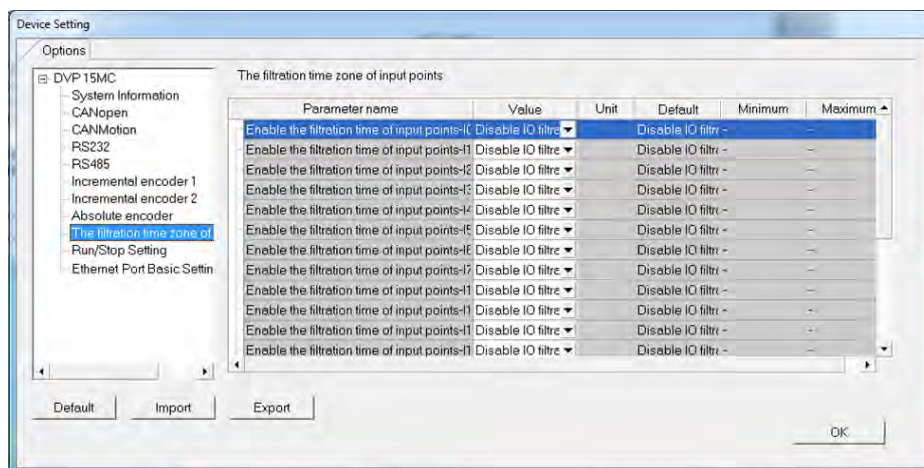
3



- ❶ The output value for absolute encoder include Gray or Binary. This parameter is set according to the encoder type.
- ❷ The parameter **Number of data bits to count turns** item sets resolution for multi-turn. A recount will take place if the number of pulses exceeds the valid maximum turn count. The number of turns (N) equals to  $2^N$ . This parameter is set according to the encoder type; the **Number of data bits per turn** item sets resolution for a single turn. The number of turns (N) equals to  $2^N$ . An encoder turns to create the number of pulses. This parameter is set according to the encoder type.
- ❸ The item sets data transfer speed.
- ❹ The item determines the minimum transfer speed that counts from the falling edge of the last pulse in clockwise direction.

### 3.6.2.9 Options - Filtration Time Zone of Input Points

The setting page for DVPxxMC Filtration Time Zone of Input Points and updates IO along.



3

- **Enable the filtration time of input points -I0~I7**

The input points to filter the short pulse signals via filtration is set between -I0~I7. This can increase filtration rate and reduce signal interference, input signal tremble or external interference. Users can select enable or disable IO filter via the drop-down menu.

- **Enable the filtration time of input points -I10~I17**

The input points to filter the short pulse signals via filtration is set between-I10~I17. This can increase filtration rate and reduce signal interference, input signal tremble or external interference. Users can select enable or disable IO filter via the drop-down menu.

- **Filtration Time Factor of I0~I7**

The I0~I7 time factor of filtration is set between 0~255. The filtration time:  $t=31\mu s * (\text{time factor of filtration})$ , the filtration time is a multiple of 31us and the default value of time factor is set to 0.

- **Filtration Time Factor I10~I13**

The I10~I13 time factor of filtration is set between 0~255. The filtration time:  $t=31\mu s * (\text{time factor of filtration})$ , the filtration time is a multiple of 31us and the default value of time factor is set to 0.

- **Filtration Time Factor I14~I17**

The I14~I17 time factor of filtration is set between 0~255. The filtration time:  $t=31\mu s * (\text{time factor of filtration})$ , the filtration time is a multiple of 31us and the default value of time factor is set to 0.

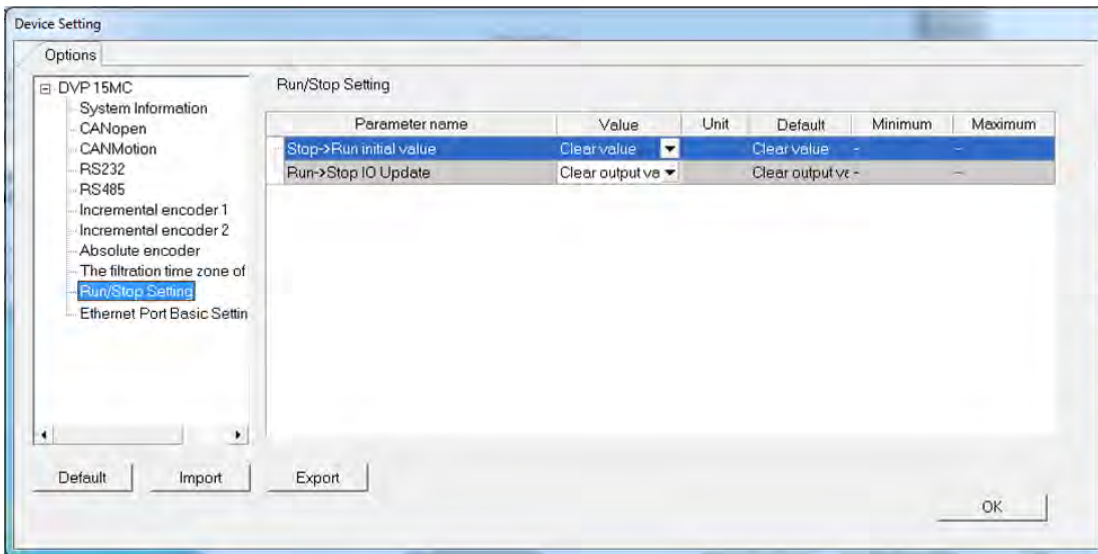
- **IO Update with Task**

The I/O includes DVPxxMC I/O points, left and right expansion modules and CANopen data. Select **FF** and IO data is updated in the system; select other assigned Task and the IO data will be updated before executing the assigned Task.

### 3.6.2.10 Options - Run/Stop Setting

The page below is set to refresh IO and initial value of parameters.

3



- **Stop -> Run Initial Value**

Users can select Clear value or Keep value

(1) When Clear Value is selected and the device operates from Stop to Run, non-retentive variables and devices will return to initial value or default value, while the retentive variables (VAR\_RETAIN type) and devices will not return to initial value.

When the non-retentive variables are bound to retentive devices, the variable will return to initial value if operates from Stop to Run or powering up; If the initial value is not set, the variable will be set as the value stored in the retentive device operates from Stop to Run or powering up.

(2) When Keep Value is selected and the device operates from Stop to Run, the variables and devices will keep the value before Stop. When the software starts download again and operates from Stop to Run, the, non-retentive variables and devices will return to initial value or default value; while the retentive variables (VAR\_RETAIN type) and devices will not return to initial value.

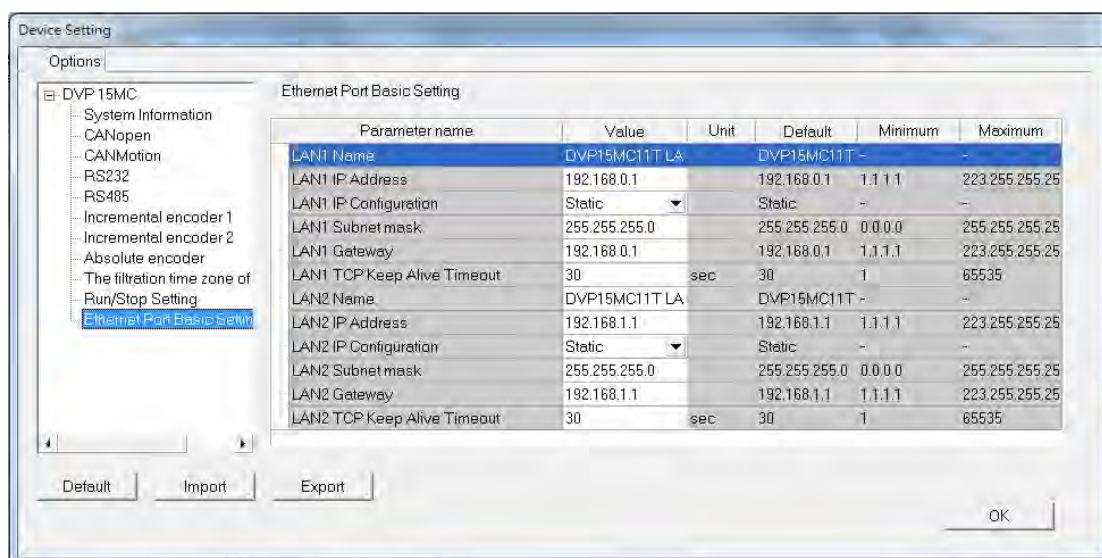
- **Run -> Stop IO Update**

Users can select Clear Output Value or Keep Output Value for IO refresh setting when the device operates from Run to Stop.

- (1) When Clear Output Value is selected and the device is under Stop, the current IO data is deleted not updated.
- (2) When Keep Output Value is selected and the device is under Stop, the current IO data is not deleted and will continue to update while under Stop.

### 3.6.2.11 Options - Ethernet Port Basic Setting

Users can select **Ethernet Port Basic Setting** under the Option menu to setup the two Ethernet port communication parameters.

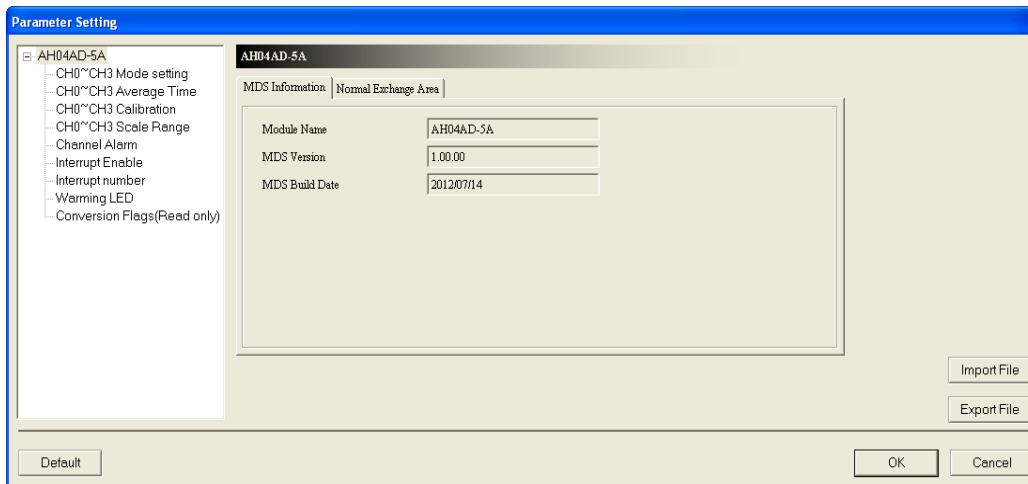


## 3.7 Parameter Settings for Modules

### 3.7.1 Setting Internal Parameters for AH5x0 and AH5x1 Modules

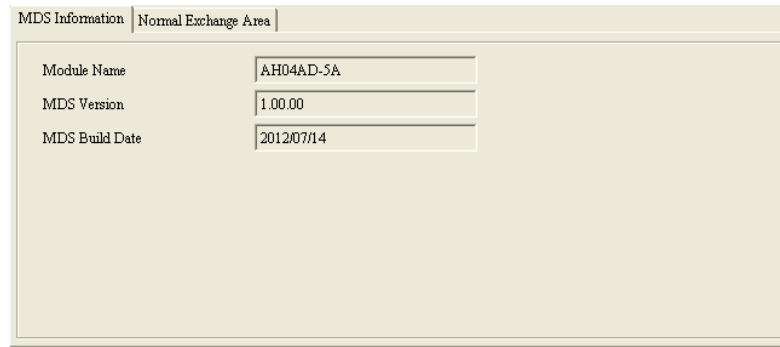
Users can set the parameters in every module configured in HWCONFIG. The functions and characters of every module depend on the setting of the parameters. (The parameter setting of AH560 redundant system series is based on AH5x1 series.) If users want to set the parameters in a module, they can double-click the module in the system configuration area. The system will automatically open the **Parameter Setting** window. The setting windows of various models are a bit different.

3



**! 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 setting window of AH5x0 and AH5x1 Modules. 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 will ask 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.

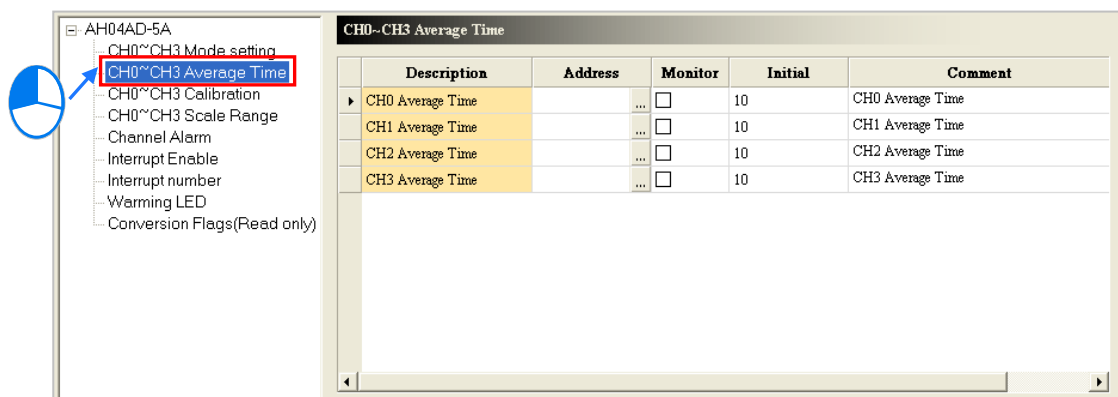
Information: Rack 1

Slot ...	Label	Firmware Version	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						

MDS Information Normal Exchange Area

Description	Address
CH0 Input value	D0 ~ D1
CH1 Input value	D2 ~ D3
CH2 Input value	D4 ~ D5
CH3 Input value	D6 ~ D7

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 grayed out, the cell cannot be edited.

3



● **Description**

The parameter names or descriptions are shown in this column.

Description	Address	Monitor	Initial	Comment
CH0 Average Time	...	<input type="checkbox"/>	10	CH0 Average Time
CH1 Average Time	...	<input type="checkbox"/>	10	CH1 Average Time
CH2 Average Time	...	<input type="checkbox"/>	10	CH2 Average Time
CH3 Average Time	...	<input type="checkbox"/>	10	CH3 Average Time

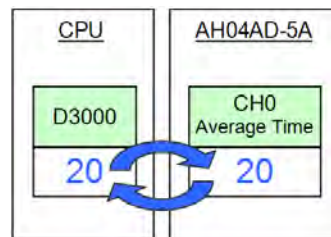
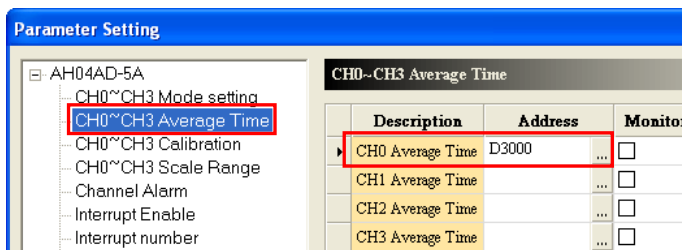
● **Address**

3

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.

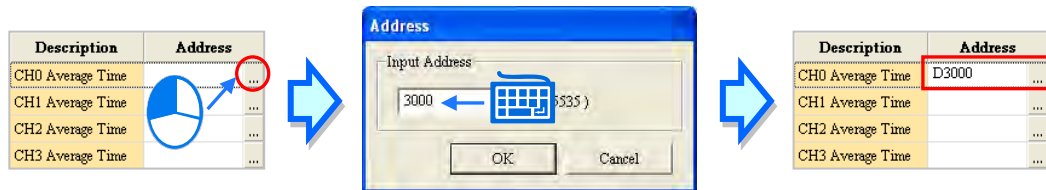
As the example below shows, D3000 corresponds to the parameter **CH0 Average Time** in the **Parameter Setting** window for AH04AD-5A. After the parameters in HWCONFIG are downloaded to the CPU module, users can change the value of the parameter **CH0 Average Time** by changing the value in D3000 in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

But 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 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 note that a data register address which has been assigned to the module cannot be used again. 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 grayed out, 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	...	<input type="checkbox"/>	<input type="checkbox"/> flashing	Warn of CH0 Input v
Warn of CH1 Input value Out of Physical Range		<input type="checkbox"/>	<input type="checkbox"/> flashing	Warn of CH1 Input v
Warn of CH2 Input value Out of Physical Range		<input type="checkbox"/>	<input type="checkbox"/> flashing	Warn of CH2 Input v
Warn of CH3 Input value Out of Physical Range		<input type="checkbox"/>	<input type="checkbox"/> 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 cannot 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 3.8.6 for more information about **Monitor Table** in HWCONFIG.

Description	Address	Monitor	Initial	Comment
CH0 Average Time	D3000	<input checked="" type="checkbox"/>	20	CH0 Average Time
CH1 Average Time	D3001	<input checked="" type="checkbox"/>	10	CH1 Average Time
CH2 Average Time	D3002	<input type="checkbox"/>	10	CH2 Average Time
CH3 Average Time	D3003	<input type="checkbox"/>	10	CH3 Average Time

3

● **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.

3

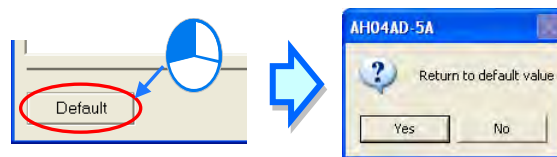
Description	Address	Monitor	Initial	Comment
CH0 Average Time	D3000 ...	<input checked="" type="checkbox"/>	20	CH0 Average Time
▶ CH1 Average Time	D3001 ...	<input checked="" type="checkbox"/>	10	CH1 Average Time
CH2 Average Time	D3002 ...	<input type="checkbox"/>	10	CH2 Average Time
CH3 Average Time	D3003 ...	<input type="checkbox"/>	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.

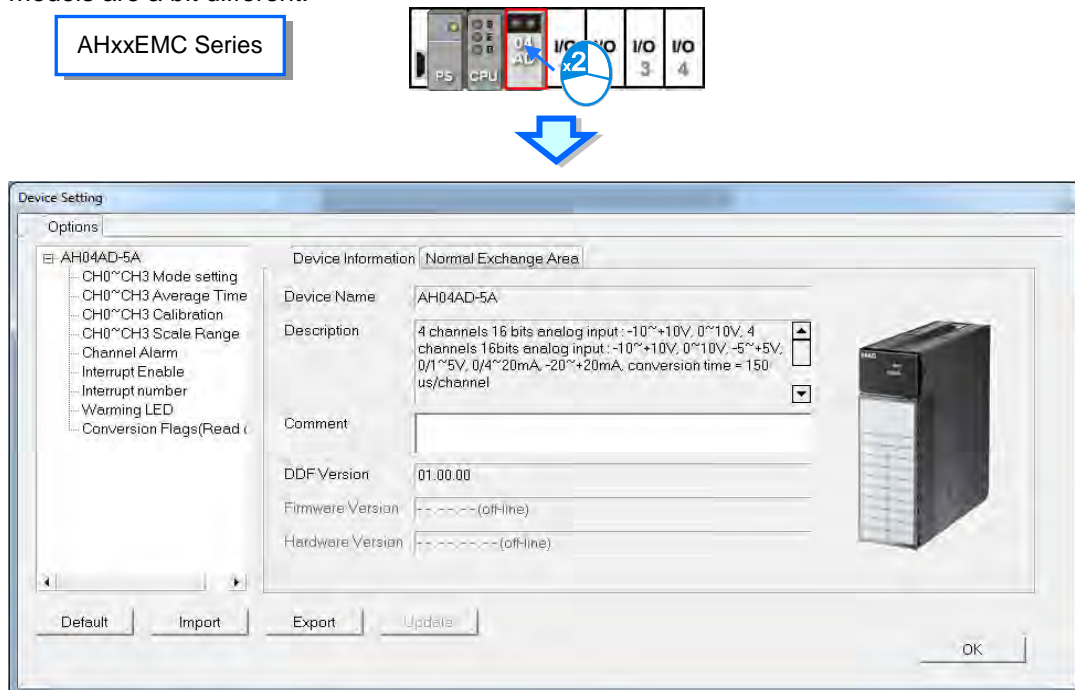
Description	Address	Monitor	Initial	Comment
▶ CH0 Average Time	D3000 ...	<input checked="" type="checkbox"/>	20	Comment...
CH1 Average Time	D3001 ...	<input checked="" type="checkbox"/>	10	CH1 Average Time
CH2 Average Time	D3002 ...	<input type="checkbox"/>	10	CH2 Average Time
CH3 Average Time	D3003 ...	<input type="checkbox"/>	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.



### 3.7.2 Parameter Settings for AHxxEMC Modules

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 setting windows of various models are a bit different.



**⚠ The parameters are set according to the functions supported by the module. Before users set the parameters, they have to refer to Operation 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 setting window of AHxxEMC series modules. They are the **Device Information** tab and the **Normal Exchange Area** tab. If users click the **Device Information** tab, the module information, appearance and present DDF document version 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 DDF document. Besides, the firmware version and hardware version of the CPU module will be displayed as well when all modules are connected online.

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.

Parameter name	Value	Unit	Default	Minimum	Maximum
CH1 average times	10		10	1	100
CH2 average times	10		10	1	100
CH3 average times	10		10	1	100
CH4 average times	10		10	1	100
CH1 filter Proportion	10%		10%	-	-
CH2 filter Proportion	10%		10%	-	-
CH3 filter Proportion	10%		10%	-	-
CH4 filter Proportion	10%		10%	-	-

**3** The descriptions of the columns in the table are as follows. If the color of a table cell is grayed out, the cell cannot be edited.

**Parameter name**

The parameter names or descriptions are shown in this column.

Parameter name	Value	Unit	Default	Minimum	Maximum
CH1 average times	10		10	1	100
CH2 average times	10		10	1	100
CH3 average times	10		10	1	100
CH4 average times	10		10	1	100

**Value**

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 type initial values for some parameter types, or select initial values on a drop-down menu for other parameter types in the **Value** column. Please refer to the instructions of the module for more information about the range of values which can be set, and the processing mechanism after the initial values are downloaded.

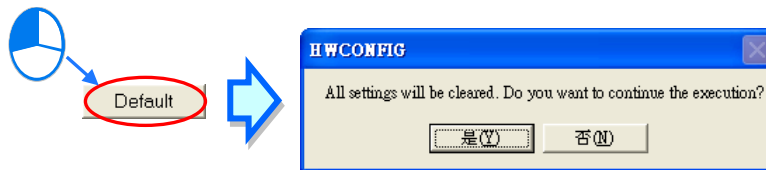
Parameter name	Value	Unit	Default	Minimum	Maximum
CH1 average times	10		10	1	100
CH2 average times	10		10	1	100
CH3 average times	10		10	1	100
CH4 average times	10		10	1	100

**Unit, Default, Minimum and Maximum**

Users can refer to the unit, default, the minimum and maximum of the parameters in the following window when setting them.

Parameter name	Value	Unit	Default	Minimum	Maximum
CH1 average times	10		10	1	100
CH2 average times	10		10	1	100
CH3 average times	10		10	1	100
CH4 average times	10		10	1	100

After **Default** button 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.



### 3.7.3 Exporting and Importing Parameters for Modules

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.

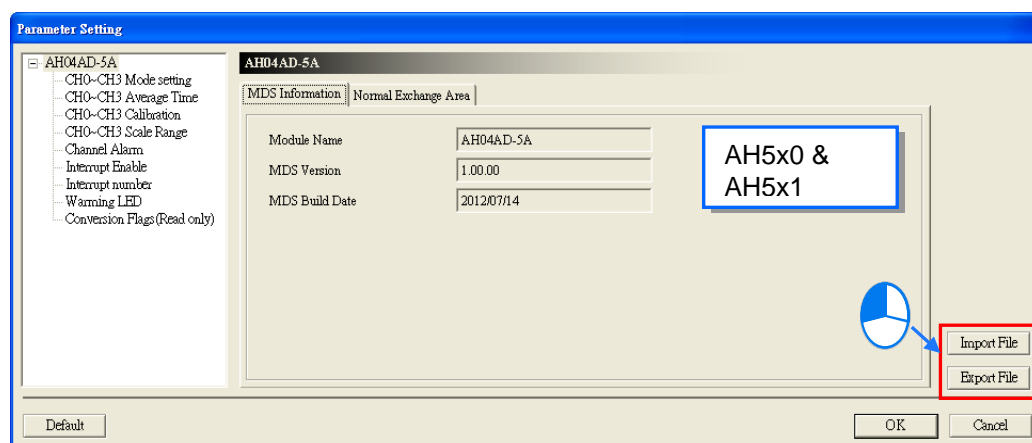
#### ● Exporting the parameters in a module

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**. The windows for importing and exporting parameters of various modules are a bit different as follows.

(The exporting and importing on the parameter setting of AH560 redundant system is based on AH5x1 series.)



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 stop 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.

### 3.7.4 Updating Parameters for Modules

For the parameters update in a module, the system needs to be in the online mode. For the operation of the online mode, please refer to section 3.8 for details. After the system is in the on-line mode, users can select the module that needs to update the parameters and then the device setting window will show up. From the device setting window, the Update option will appear. Click Update button to update the parameters in a module. This function is only available for digital I/O modules, analog I/O modules and network modules.

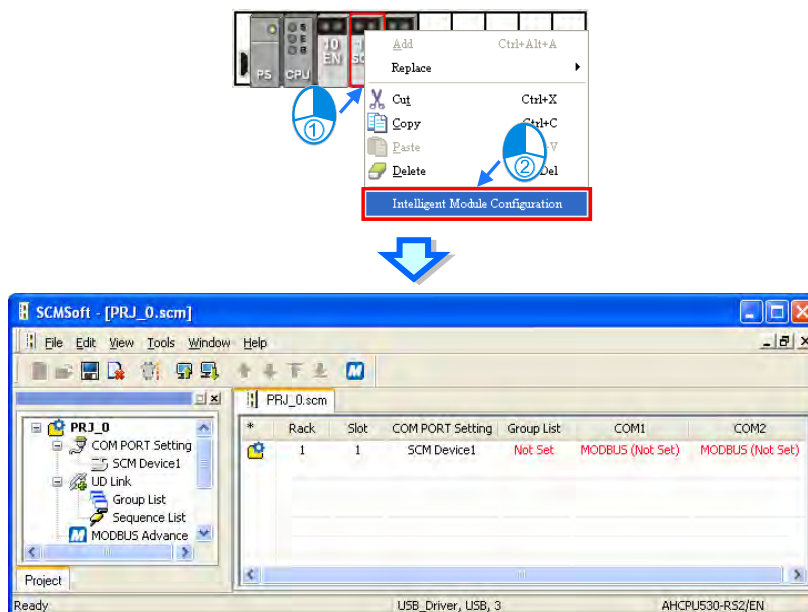
## 3

### 3.7.5 Parameter Settings for Intelligent Modules

Delta Electronics, Inc. provides the exclusive configuration tools for some modules. Through the software, the modules can be further configured and the parameters in the modules can be further set, such as CANopen network planning and etc.

Take the AH5x0 as an example (shown below), 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 starting the corresponding software. Once the corresponding software is started, users can not perform any editing tasks in HWCONFIG before the software is closed. Some software need be installed additionally and there will be a message to remind users of installing the configuration software if the software is not installed.

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




## 3.8 HWCONFIG Parameter Management and Online Diagnosis

	AS200	AS300	AH5x0	AH5x1	AH560	AHxxEMC	DVP	DVPxxMC
HWCONFIG Print Function	V	V	V	V	V	V		
Purchase Order	V	V	V	V	V	V		

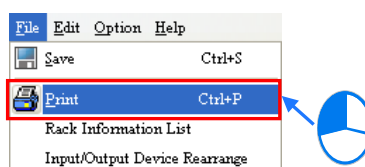
### 3.8.1 Save and Print function

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.

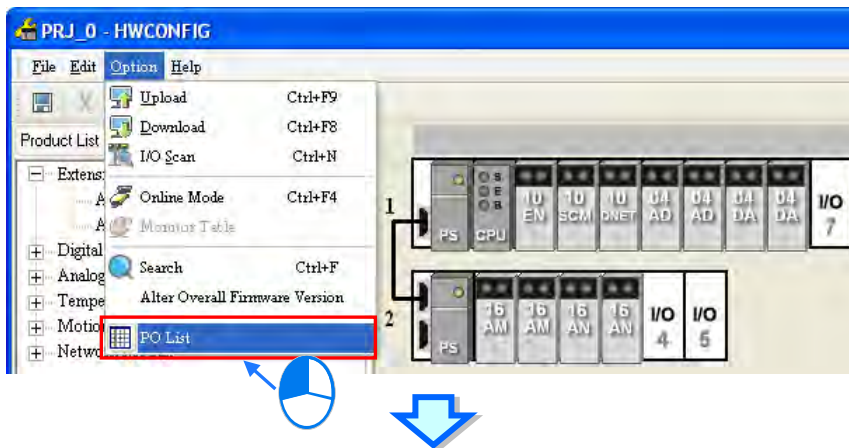


The problematic modules are marked with triangular error signs in HWCONFIG. These modules are in an abnormal condition. The **Download** function and the **Online Mode** function cannot be enabled. After the incorrect device addresses are modified, users can proceed with the operation. Besides, the triangular error signs will disappear after the incorrect devices addresses are modified.



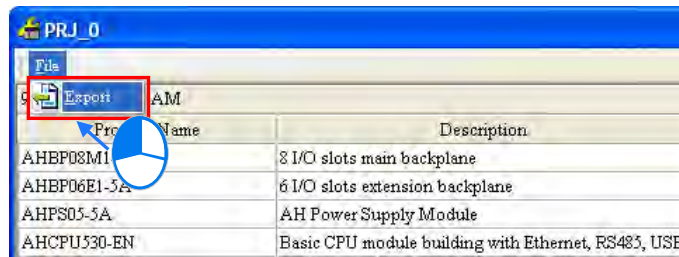
### 3.8.2 Purchase Order

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 show it in a list.



Product Name	Description	Count
AHBP08M1-5A	8 I/O slots main backplane	1
AHBP06E1-5A	6 I/O slots extension backplane	1
AHPS05-5A	AH Power Supply Module	2
AHCPU530-EN	Basic CPU module building with Ethernet, RS485, USB	1
AH10EN-5A	Ethernet master module	1
AH10SCM-5A	Serial communication module	1
AH10DNET-5A	DeviceNet scanner	1
AH04AD-5A	4 x AI 16bit	2
AH04DA-5A	4 x AO 16bit	2
AH16AM10N-5A	16 x DI, 24 VDC	2
AH16AN01R-5A	16 x DO, Relay 240 VAC/24 VDC	2
Extension Communication Cable	I/O Extension Cable	1

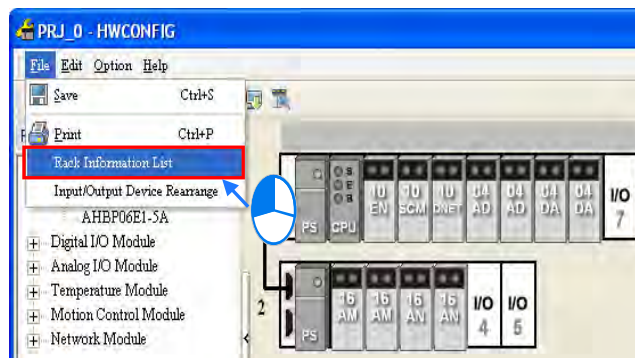
After **Export** on the **File** menu in the upper left corner of the window is clicked, the list will be saved as a CSV file. The data which is exported can be edited or managed through Microsoft Excel.



### 3.8.3 Rack Information List for AH Series

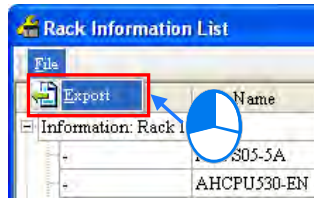
3

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



Slot No.	Name	Description	Input Device Range	Output Device Range	Comment
<b>Information: Rack 1</b>					
-	AHPS05-5A	AH Power Supply Module	None	None	
-	AHCPU530-EN	Basic CPU module building with	None	None	
0	AH10EN-5A	Ethernet master module	D0 ~ D19	D20 ~ D39	
1	AH10SCM-5A	Serial communication module	D40 ~ D57		
2	AH10DNET-5A	DeviceNet scanner	None	None	
3	AH04AD-5A	4 x AI 16bit	D58 ~ D65		
4	AH04AD-5A	4 x AI 16bit	D66 ~ D73		
5	AH04DA-5A	4 x AO 16bit		D74 ~ D81	
6	AH04DA-5A	4 x AO 16bit		D82 ~ D89	
7					
<b>Information: Rack 2</b>					
-	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.




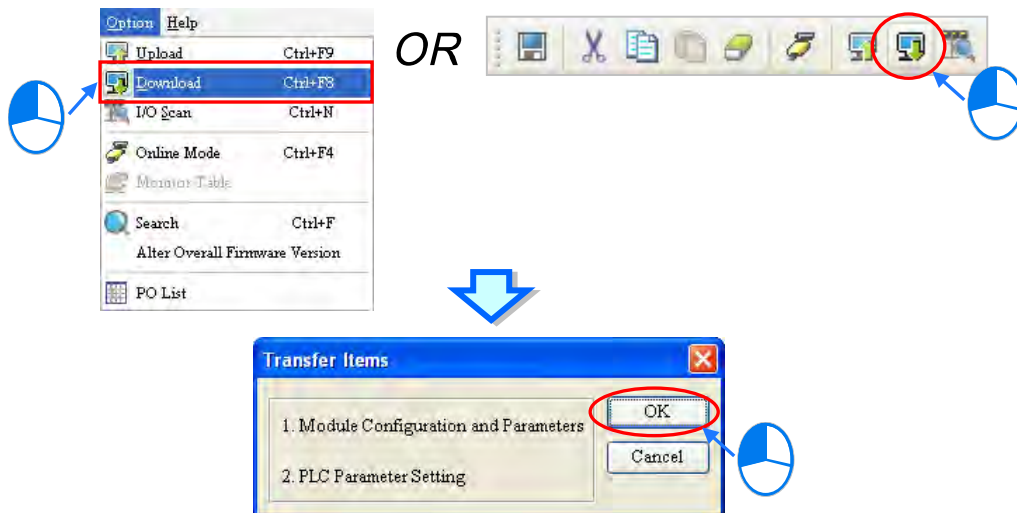
### 3

### 3.8.4 Download/Upload 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. 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.


#### ● Downloading the system parameters

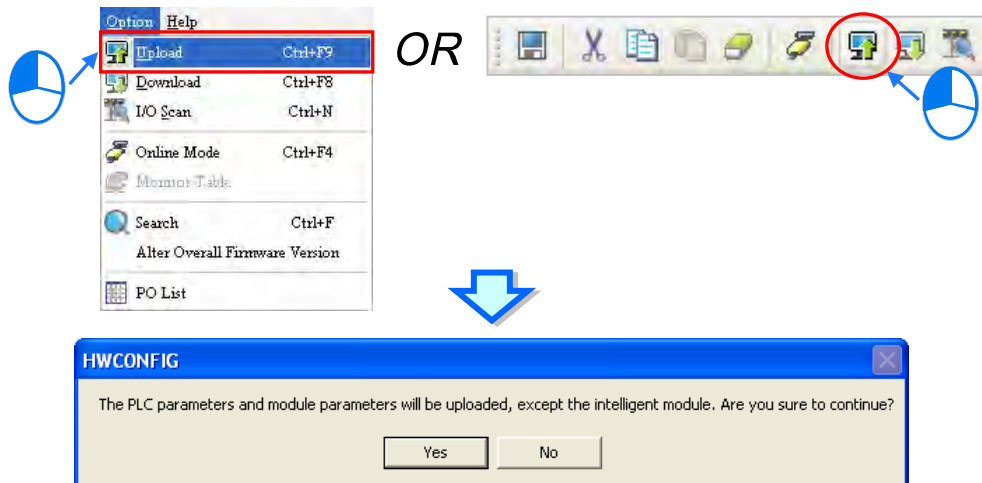
After users click **Download** on the **Option** menu, or  on the toolbar, the **Transfer Items** window will appear. The items which will be downloaded can be selected as needed, but the items do not include the parameters in the intelligent modules. After the users click **OK**, the parameters will be downloaded to the CPU module.





\*. For AH5x0 series, download items are not selectable. But AH560 redundant system and AH5x1 series provides download items.

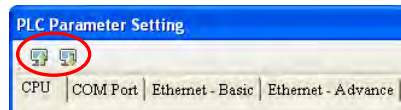
● **Uploading the system parameters**

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.



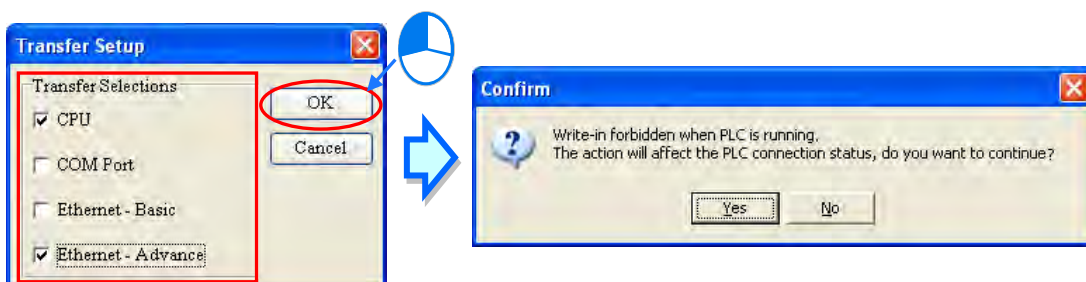
3

For AH5x0 series, the PLC parameter setting window provides the function of uploading and downloading, but only for the CPU parameters. Click  to download PLC parameters and  to upload PLC parameters in the upper left of the PLC parameter setting window.



All CPU parameters will be uploaded during uploading. During downloading data, users can select the parameters to be downloaded in the following window rather than downloading all settings to the CPU module.

The CPU module must be in STOP state if **CPU** or **Ethernet –Advance** checkboxes are selected in the following window. And so the system will show a reminder message before the downloading starts if the CPU module is in RUN state.




\*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.

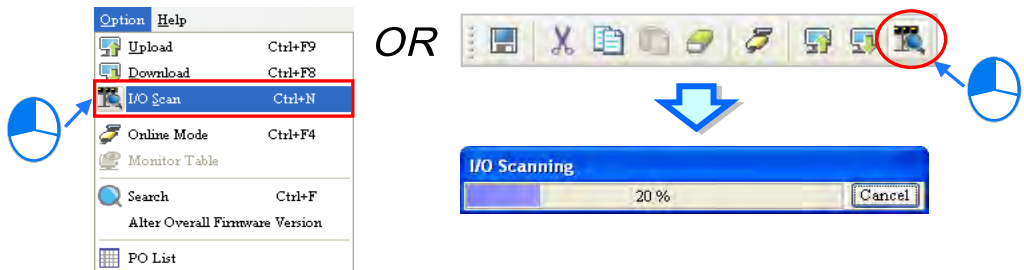
\*2. HWCONFIG adopts the communication setting in ISPSOft. Please make sure that ISPSOft is connected to the CPU module normally before the parameters are downloaded/uploaded. Please refer to section 2.4 for more information.

### 3.8.5 I/O Scan

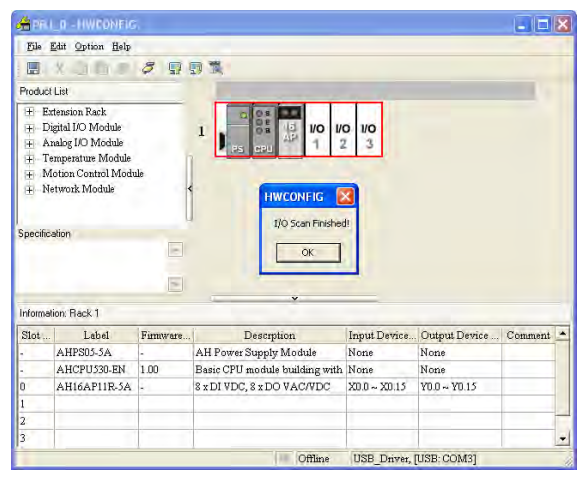
3

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.

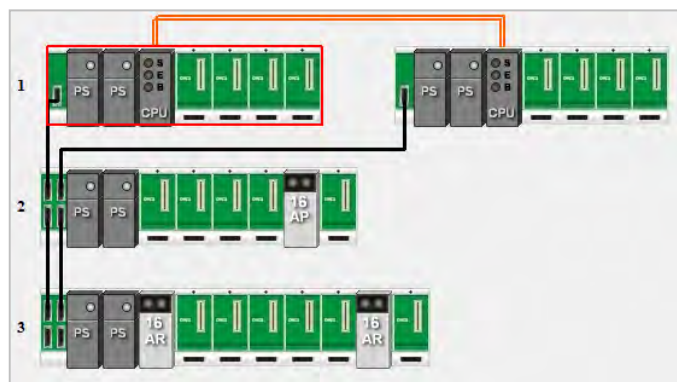
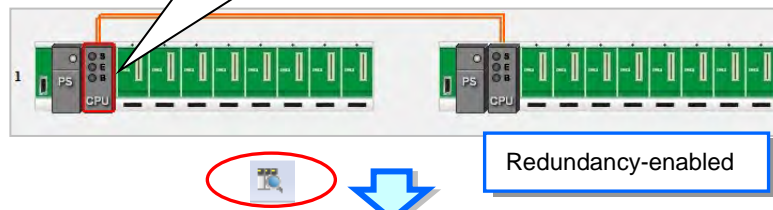
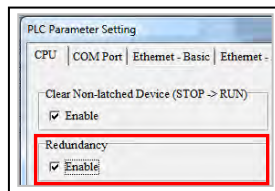
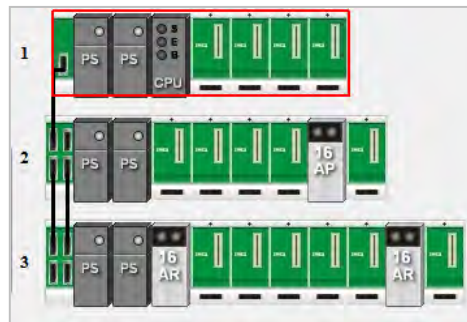
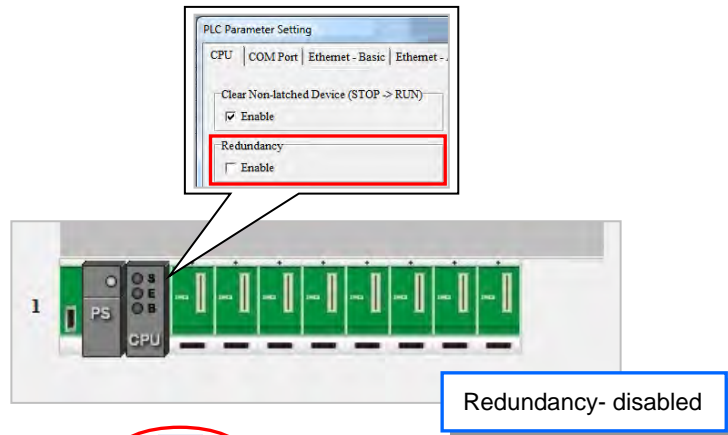


After the scan is complete, the actual hardware configuration will be displayed in the system configuration area, the system will assign input/output devices to the modules, the parameters in the modules are assigned the default values.



For AH560 redundant system, the I/O Scan result is different when the redundancy function is enabled (please refer to section 3.3.2.2 for more information) or disabled. Two examples are given below. (The actual scan result may differ base on real connection status.)

3



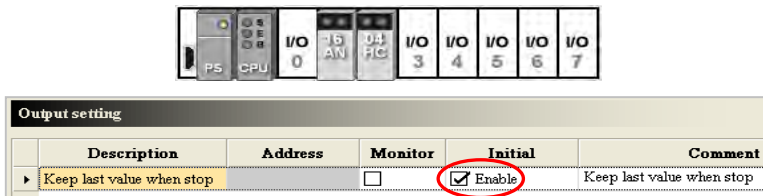
The firmware versions of AH5x0,AH5x1 and AH560 redundant system series modules will be based on the actual firmware versions of the modules scanned. If the firmware version of a module which is scanned is higher than the firmware version of the module in the MDS document, the highest version which can be selected will be displayed in the **Firmware Version** drop-down list for the module, and the actual firmware version of the module will be displayed in the **Comment** cell for the module. For AHxxEMC series, both module firmware adopts the scanned firmware version. However, if the firmware version of the scanned module is not compatible with the system firmware (DDF version), users can select the highest version number for Firmware Version from the drop-down list, while the actual firmware version is shown in the Comment cell.

3

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

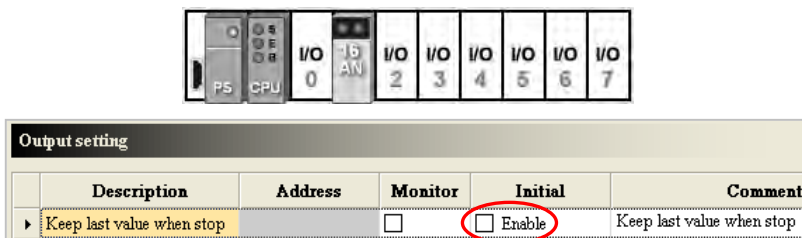
The difference between **I/O Scan** and **Upload** can be illustrated with the following example. If **I/O Scan** is clicked, the hardware configuration will be scanned, and the parameters in the modules will not be scanned. If **Upload** is clicked, the hardware configuration in the CPU module will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

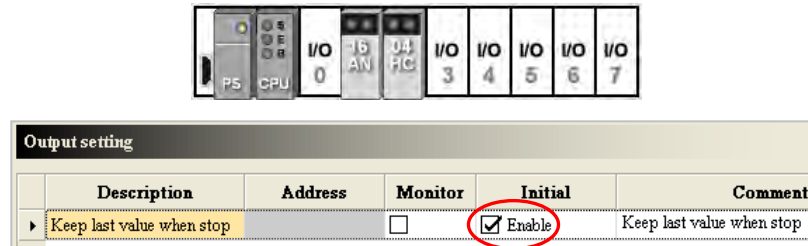


Then, AH04HC-5A installed in slot 2 is removed from the actual backplane. The BUS FAULT LED indicator on the CPU module will be ON, that is to say, the hardware configuration in the CPU is different from the actual hardware configuration.

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.





3

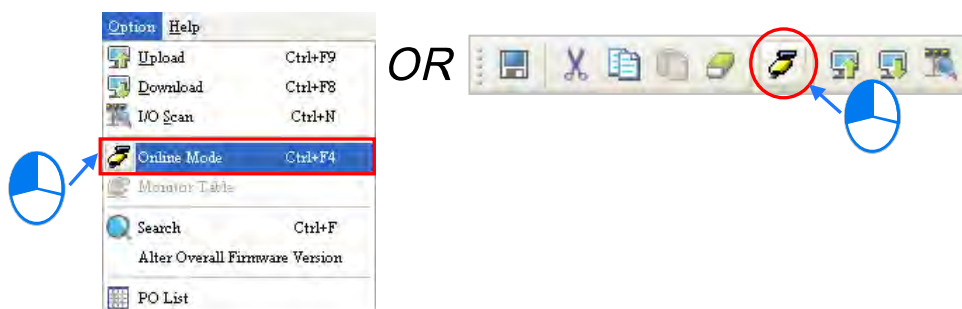
Next, **I/O Scan** is clicked, and the scan result is downloaded to the CPU module. The BUS FAULT LED indicator on the CPU module will be OFF, that is to say, the hardware configuration in the CPU module is the same as the actual hardware configuration.

### 3.8.6 AH Series 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. 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.

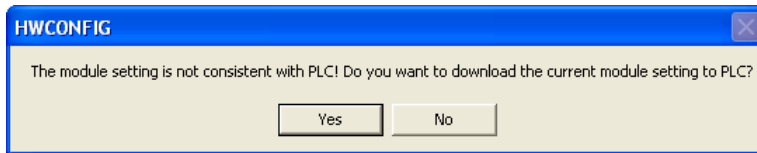
#### 3.8.6.1 Online Mode

- (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.



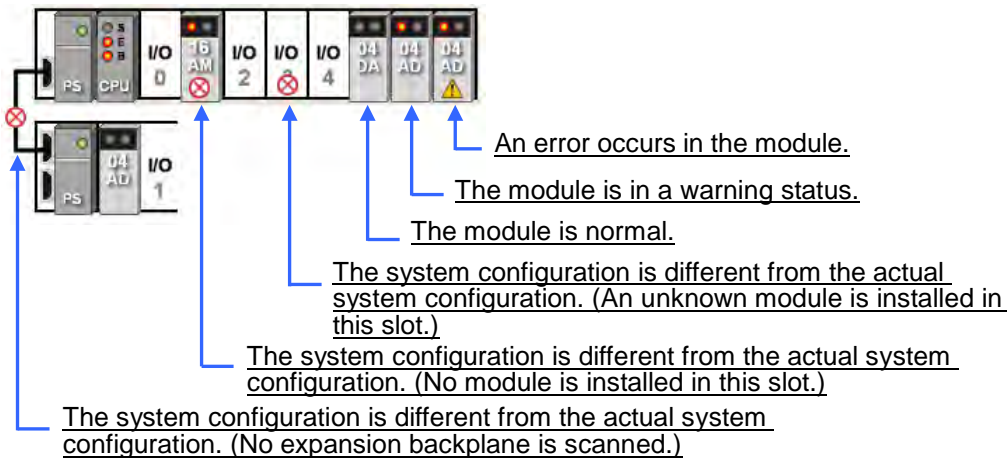


- (2) Before the hardware configuration enters the online mode, the system checks whether the hardware configuration stored in the CPU module is the same as the hardware configuration in the system configuration area. If the hardware configuration stored in the CPU module is different from the hardware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



3

- (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.

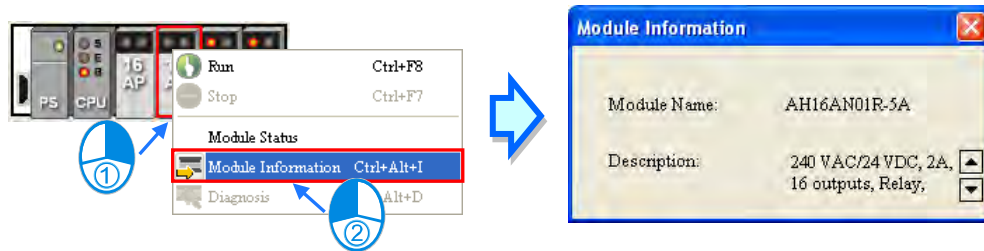


For AH560 redundant system, if the **Redundancy** function is enabled and enters **online mode**, the status of Master and Standby controllers will be displayed in the system configuration area.

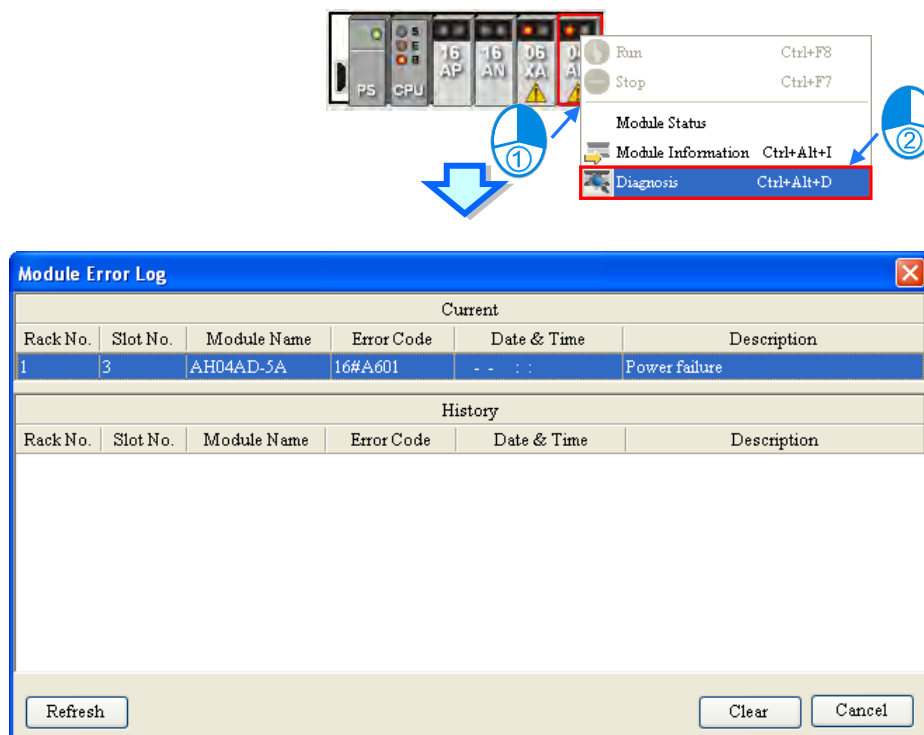


### 3.8.6.2 Module Information and Diagnosis

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.

### 3.8.6.3 Changing the Status of a Module Online

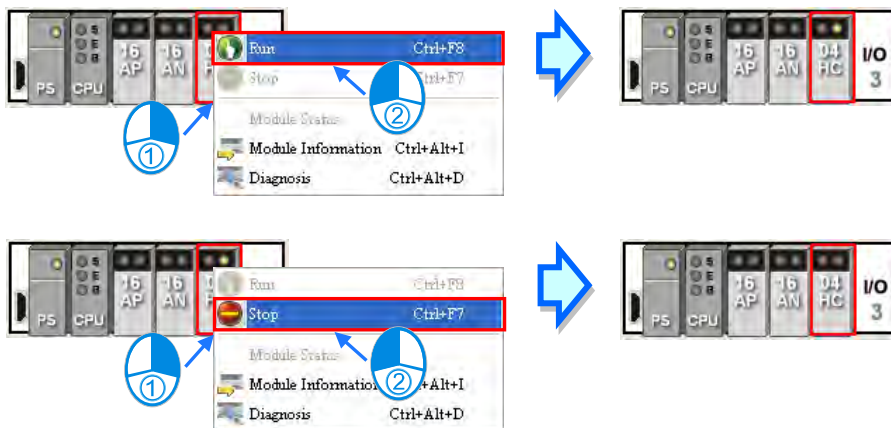
If the hardware configuration is in the online mode, users can change the status of a module, carry out a simple test.

3

**⚠ 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.**

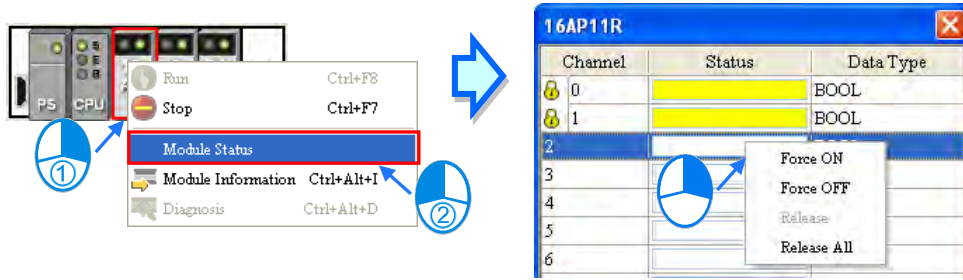
#### ● Changing the operating state of a module

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.



#### ● 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.



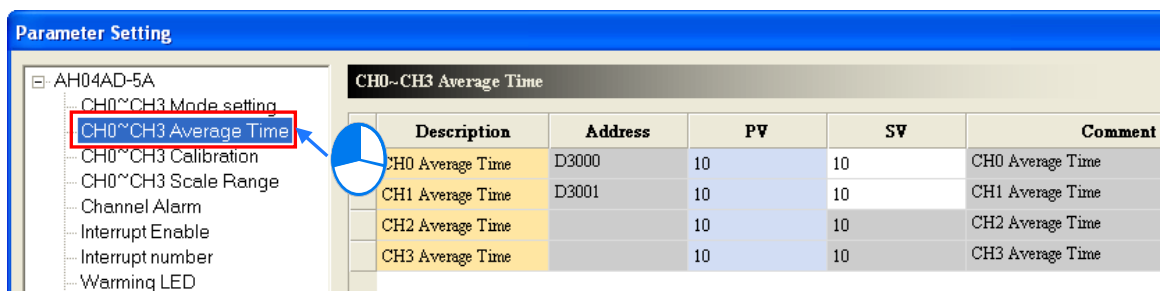
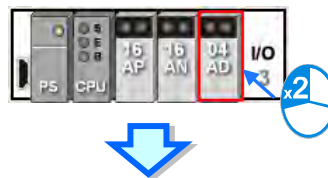
- **Force ON:** The channel selected is forced ON.
- **Force OFF:** The channel selected is forced OFF.
- **Release:** The channel selected is not forced ON/OFF.
- **Release all:** All the channels are not forced ON/OFF.

3

● **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  in the window, the system will retrieve the present values again, and the values retrieved will be displayed in the **PV** column.

CH0~CH3 Average Time					
	Description	Address	PV	SV	Comment
▶	CH0 Average Time	D3000	10	10	CH0 Average Time
	CH1 Average Time	D3001	10	10	CH1 Average Time
	CH2 Average Time		10	10	CH2 Average Time
	CH3 Average Time		10	10	CH3 Average Time

3

- (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  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.

CH0~CH3 Average Time					
	Description	Address	PV	SV	Comment
	CH0 Average Time	D3000	10	20	CH0 Average Time
▶	CH1 Average Time	D3001	10	30	CH1 Average Time
	CH2 Average Time		10	10	CH2 Average Time
	CH3 Average Time		10	10	CH3 Average Time

\*. 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.

### 3.8.6.4 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 3.7.2 for more information.

Description	Address	Monitor	Initial
CH0 Cal. Offset (V/mA)	D3008-D3009	<input checked="" type="checkbox"/>	0.000000
CH1 Cal. Offset (V/mA)	D3010-D3011	<input checked="" type="checkbox"/>	0.000000
CH2 Cal. Offset (V/mA)	D3012-D3013	<input checked="" type="checkbox"/>	0.000000
CH3 Cal. Offset (V/mA)	D3014-D3015	<input checked="" type="checkbox"/>	0.000000
CH0 Cal. Gain	D3016-D3017	<input type="checkbox"/>	1.000000
CH1 Cal. Gain	D3018-D3019	<input type="checkbox"/>	1.000000
CH2 Cal. Gain		<input type="checkbox"/>	1.000000
CH3 Cal. Gain		<input type="checkbox"/>	1.000000

Rack No.	Slot No.	Module Name	Device Name	PV	Radix	Comment
1	2	AH04AD-5A	D3008	0	Float	CH0 Cal. Offset (V/mA)
1	2	AH04AD-5A	D3010	0	Float	CH1 Cal. Offset (V/mA)
1	2	AH04AD-5A	D3012	0	Float	CH2 Cal. Offset (V/mA)
1	2	AH04AD-5A	D3014	0	Float	CH3 Cal. Offset (V/mA)

3

\*. 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 3.7.1. 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.

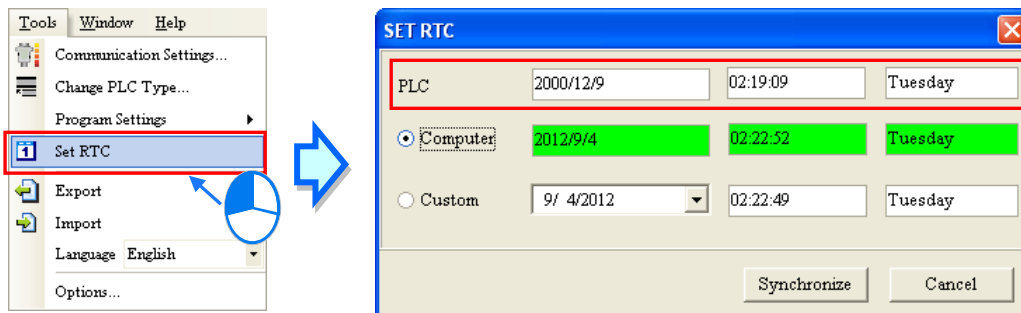
Rack No.	Slot No.	Module Name	Device Name	PV	Radix	Comment
1	2	AH04AD-5A	D3008	0	Float	CH0 Cal. Offset (V/mA)
1	2	AH04AD-5A	D3010	0	Float	CH1 Cal. Offset (V/mA)
1	2	AH04AD-5A	D3012	0	Float	CH2 Cal. Offset (V/mA)
1	2	AH04AD-5A	D3014	0	Float	CH3 Cal. Offset (V/mA)

### 3.9 Setting a RTC

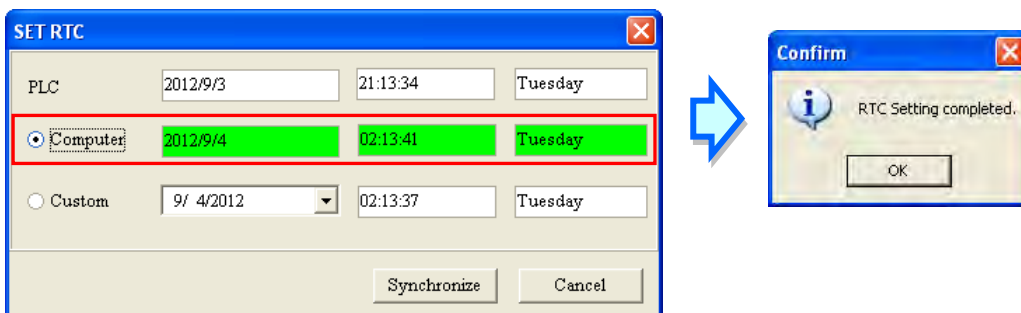
The RTC in a PLC can be set by a tool provided by ISPSOft. Before the real-time clock in a PLC is set, users have to make sure that ISPSOft is connected to the PLC normally. Please refer to section 2.4 for more information about communication setting.


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

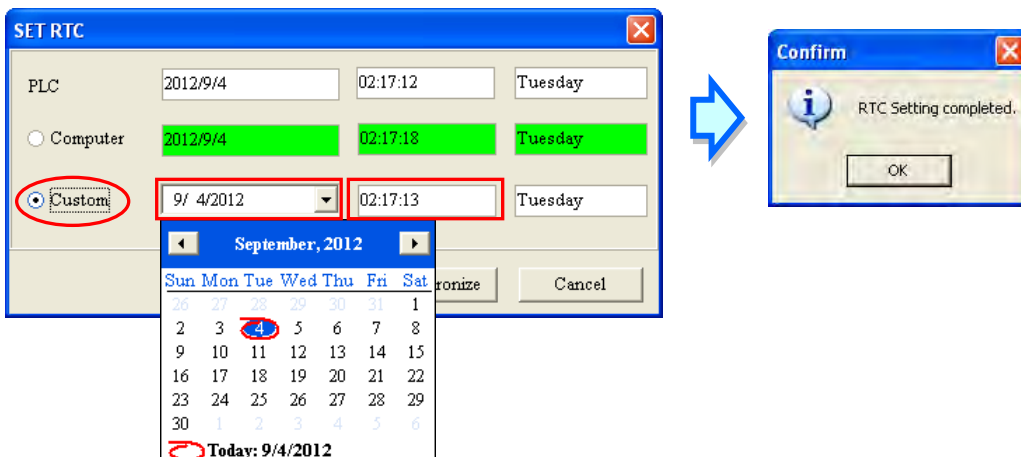
3



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**.



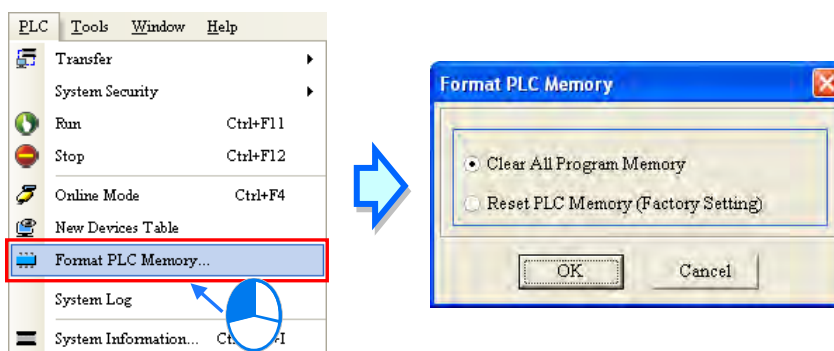
If users want to set the real-time clock by themselves, they can select the **Custom** option button, and type a date and a time in the boxes. If users click  in the box, an auxiliary tool will appear. Click **OK** after the setting of the real-time clock is complete



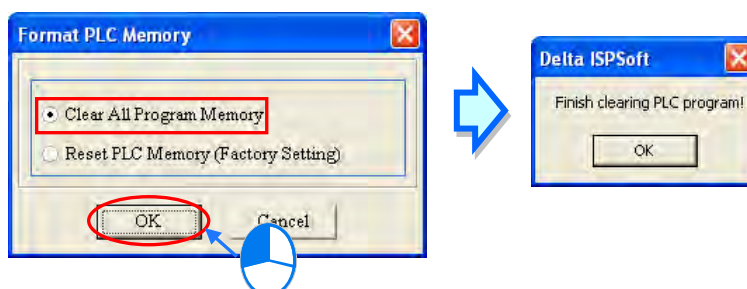
### 3.10 Setting the Memory in a PLC

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.

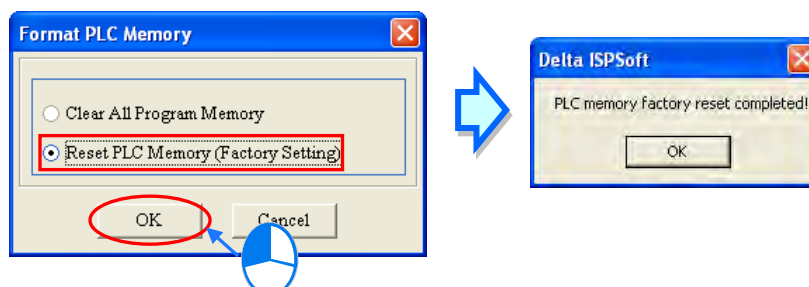
After users click **Format PLC Memory...** on the **PLC** menu, the **Format PLC Memory** will appear.



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



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 AH 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 and 3.4 for more information.



# 4

## Chapter 4 Quick Start

### Table of Contents

<b>4.1 Quick Start .....</b>	<b>4-2</b>
4.1.1 Example .....	4-2
4.1.2 Hardware.....	4-3
4.1.3 Program .....	4-3
<b>4.2 Procedure for Creating a Project in ISPSOft .....</b>	<b>4-4</b>
<b>4.3 Creating a Project .....</b>	<b>4-5</b>
<b>4.4 Hardware Configuration .....</b>	<b>4-6</b>
4.4.1 Configuring a Module.....	4-6
4.4.2 Parameter Setting for PLCs and Modules.....	4-8
<b>4.5 Creating a Program.....</b>	<b>4-10</b>
4.5.1 Adding a Ladder Diagram.....	4-10
4.5.2 Basic Editing - Creating a Contact and a Coil .....	4-12
4.5.3 Basic Editing - Inserting a Network and Typing an Instruction .....	4-15
4.5.4 Basic Editing - Selection of a Network and Operation.....	4-17
4.5.5 Basic Editing - Connecting a Contact in Parallel.....	4-20
4.5.6 Basic Editing - Editing a Comment .....	4-21
4.5.7 Basic Editing - Inserting an Applied Instruction.....	4-22
4.5.8 Basic Editing - Creating a Comparison Contact and Typing a Constant ...	4-24
4.5.9 Writing a Program.....	4-25
4.5.10 Checking and Compiling a Program .....	4-26
<b>4.6 Testing and Debugging a Program.....</b>	<b>4-27</b>
4.6.1 Creating a Connection .....	4-27
4.6.2 Downloading a Program and Parameters.....	4-30
4.6.3 Connection Test.....	4-33

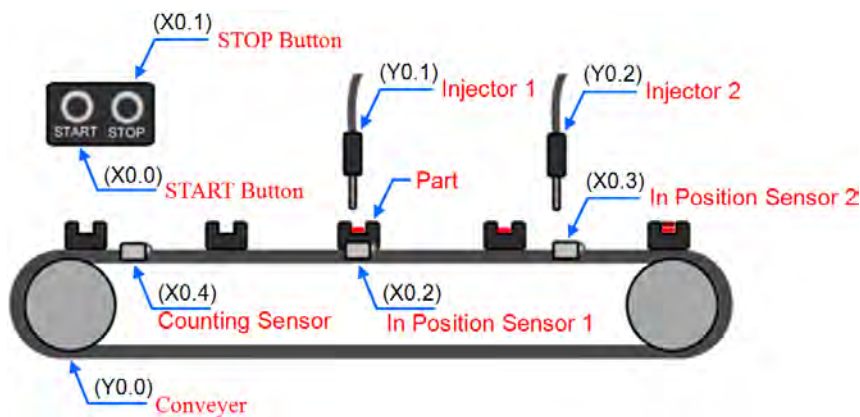
## 4.1 Quick Start

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, POU, 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, we use AH5x0 series host CPU- **AHCPU530-EN**, the **AH16AP11R-5A** with 8-point digital I/O module, and **AHBP04M1-5A** with 4 slots as the main backplane. The following is an example of 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

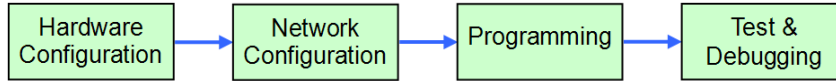


### 4.1.3 Program

- (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.

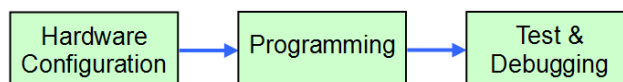
### ● Programming

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

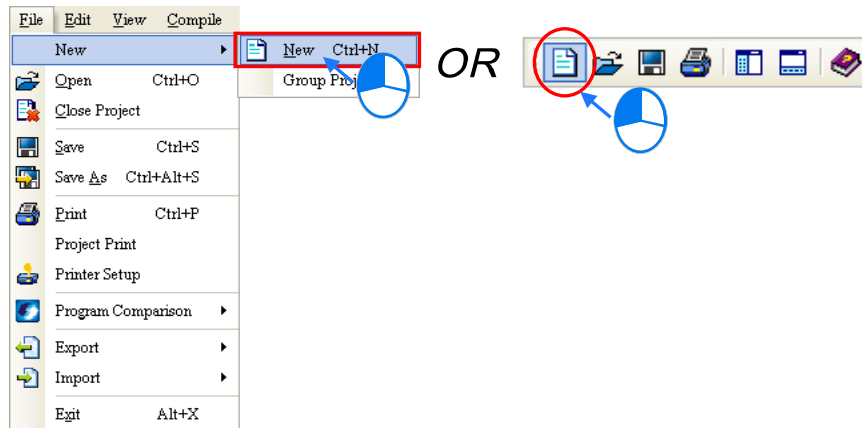
Users can download a program which is compiled, a hardware configuration, and a network configuration to a PLC. Besides, the users can test and debug the program online by means of the functions provided by ISPSOft.

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.

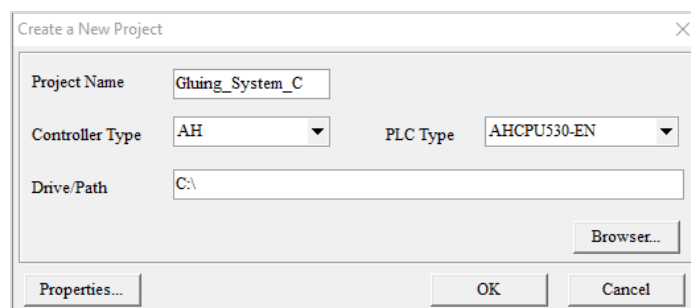



### 4.3 Creating a Project

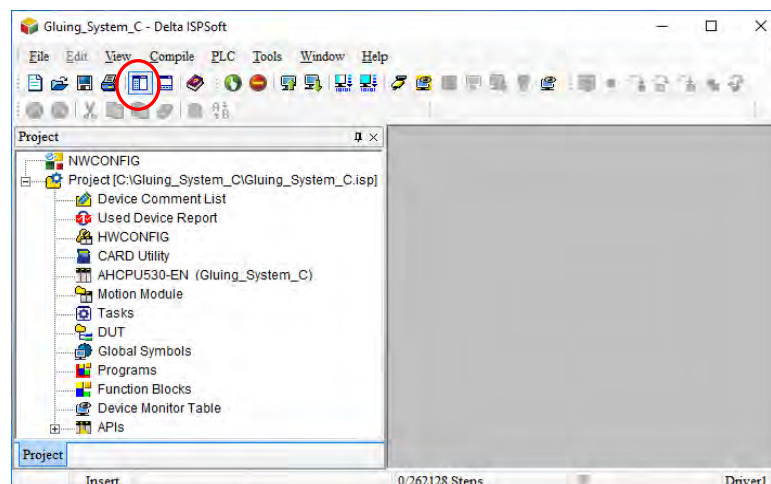
After ISPSOft is started, users can click the **File** menu, point to **New**, and click **New** to create a new project. They can also create a new project by clicking  on the toolbar after ISPSOft is started.



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 drop-down list boxes of **Controller Type** and **PLC Type**, and click **OK**. (The PLC used in this example is AHCPU530-EN.)



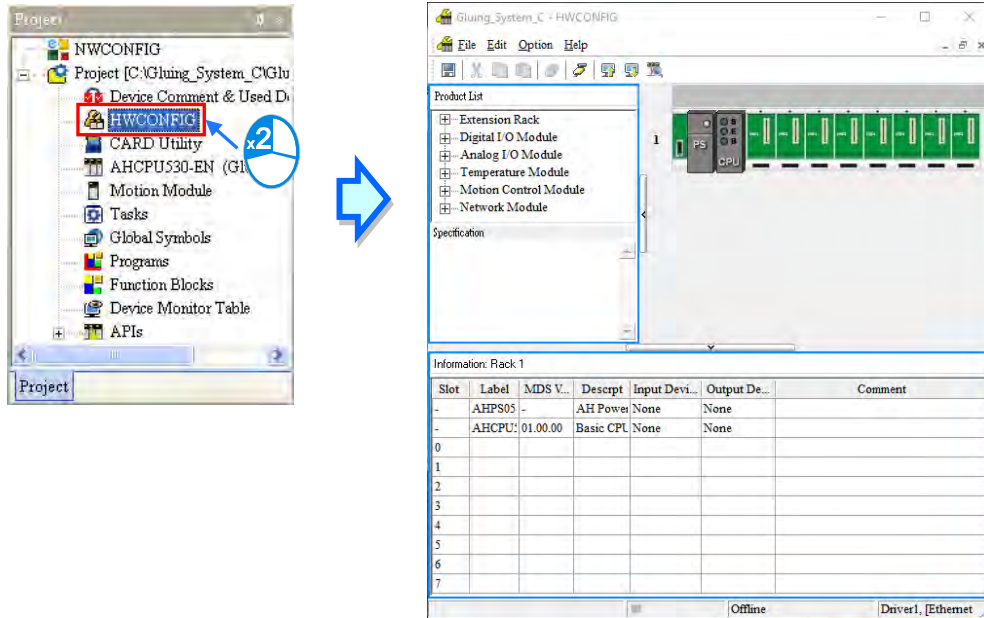
After the project is created successfully, a project management area will appear at the left side of the main screen. The relation between the items listed in the project management area is represented by a hierarchical tree structure. If the project management area does not appear, the users can click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



4

## 4.4 Hardware Configuration

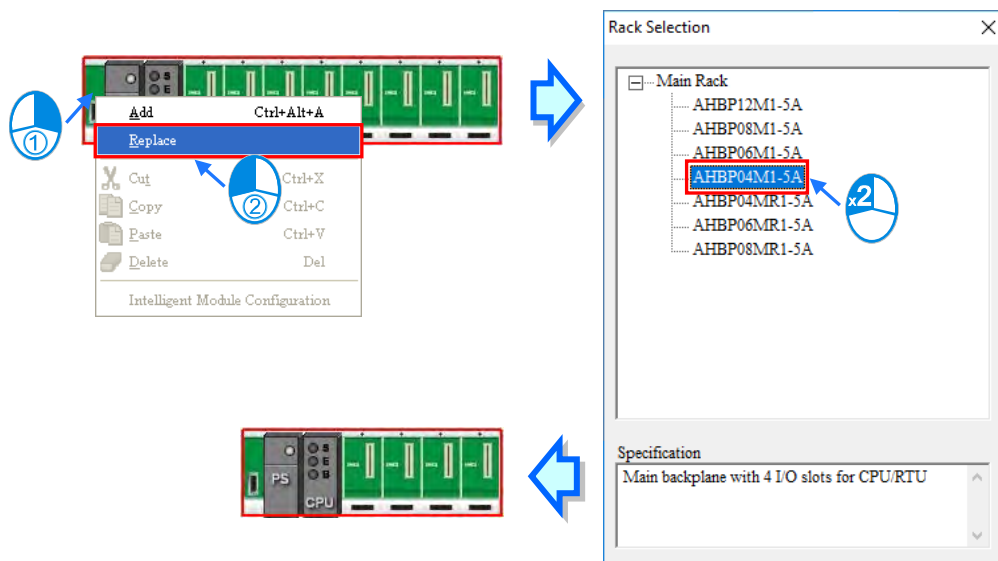
After users double-click **HWCONFIG** in the project management area, the **HWCONFIG** window will appear.



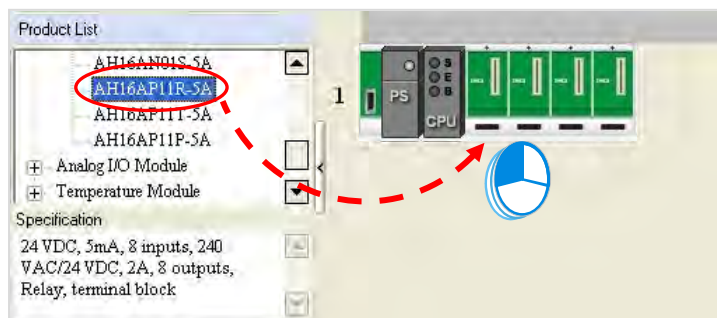
### 4.4.1 Configuring a Module

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	MDS Version	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	01.00.00	Basic CPU module building with Ethernet, RS485,	None	None	
0	AH16AP11R-5A	-	8 x DI VDC, 8 x DO VAC/VDC	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1						
2						
3						

The system automatically assigns devices to a module which is added. If the devices assigned to a module do not 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.

Information: Rack 1

Slot No.	Label	MDS Version	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	-	AH Power Supply Module	None	None	
-	AHCPU530-EN	01.00.00	Basic CPU module building with Ethernet, RS485,	None	None	
0	AH16AP11R-5A	-	8 x DI VDC, 8 x DO VAC/VDC	X0.0 ~ X0.15 <b>...</b>	Y0.0 ~ Y0.15	
1						
2						
3						

Blue circles with numbers 1 and 2 are placed over the '...' button and the 'Input Device Range' cell respectively, with arrows pointing to the Manual Assignment window below.

Manual Assignment

Input Device Range

Device X

Number 0  
(X0 ~ X511)

Length 1

OK Cancel

4

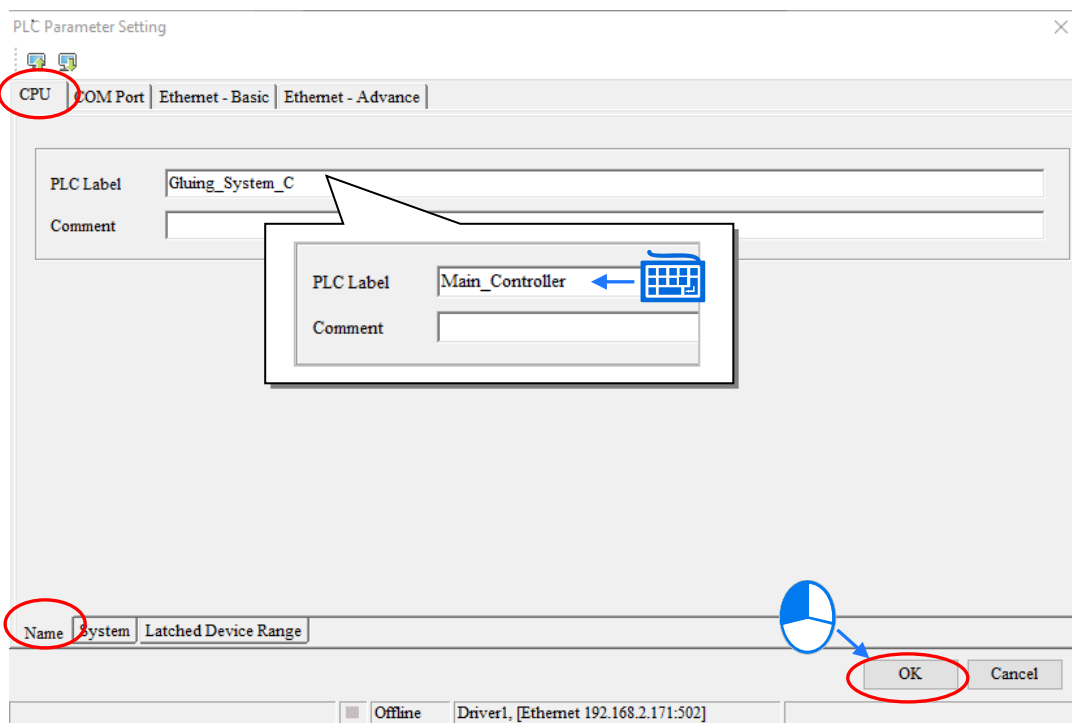
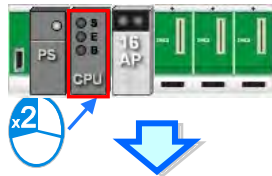
### 4.4.2 Parameter Setting for PLCs and Modules

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.

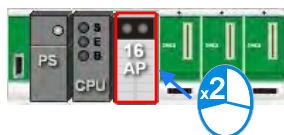
After the CPU module is double-clicked, the **PLC Parameter Setting** window will appear. The users can click the primary tabs at the top of the window, and the secondary tables at the bottom of the window to set the parameters. In this example, the users only need to define the name of the CPU module.

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**.

4



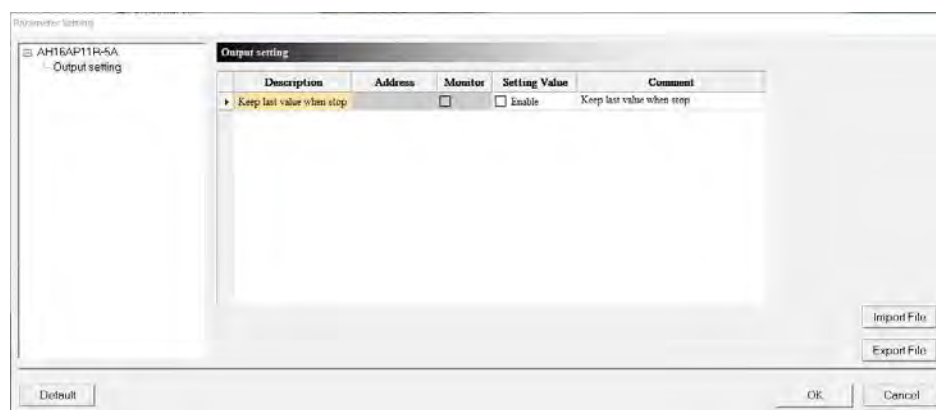
After the users double-click AH16AP11R-5A, the **Parameter Setting** window will appear.






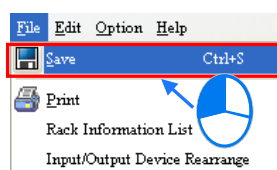
After the **Parameter Setting** window is opened, the users can view the information related to the module. The users can select the parameter type at the left side of the window, and then set the parameter in the table at the right side of the window.

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



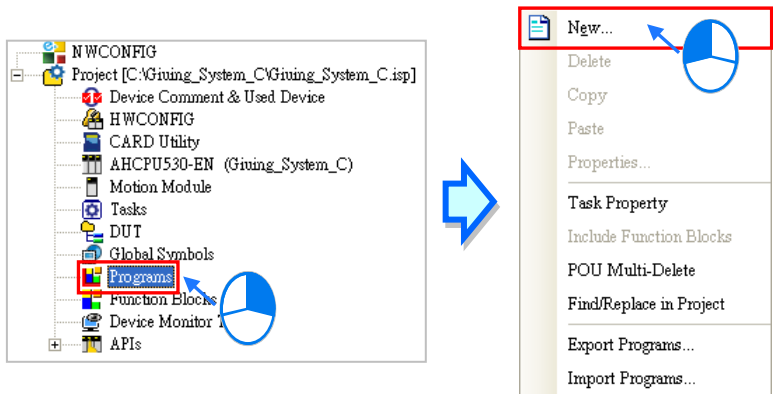
\*. Please refer to chapter 3 for more information about HWCONFIG.

## 4.5 Creating a Program

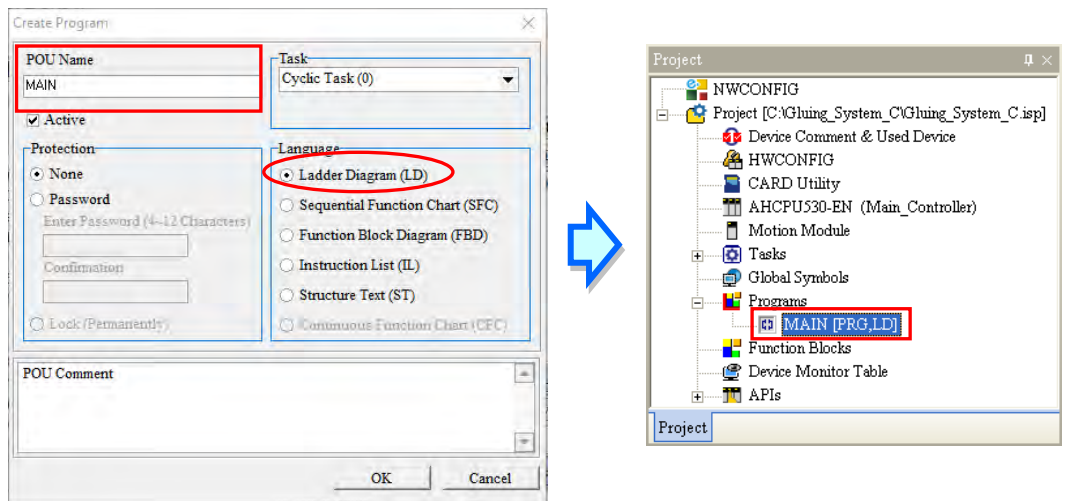
The following sections will lead users to create a traditional ladder diagram in ISPSOft. The contents of the following sections include creating a POU, editing a traditional diagram, and compiling a program. The users are expected to equip themselves with the basic abilities to create a traditional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

### 4.5.1 Adding a Ladder Diagram

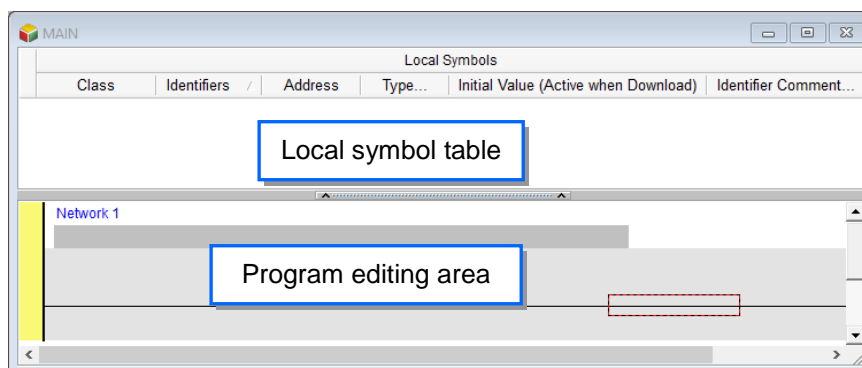
- (1) Right-click **Programs** in the project management area and click **New....**



- (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
	None	Switch to the Address mode
	Shift+Ctrl+C	Displaying/Hiding the comments on the networks
	None	Displaying/Hiding the commands on the devices
	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

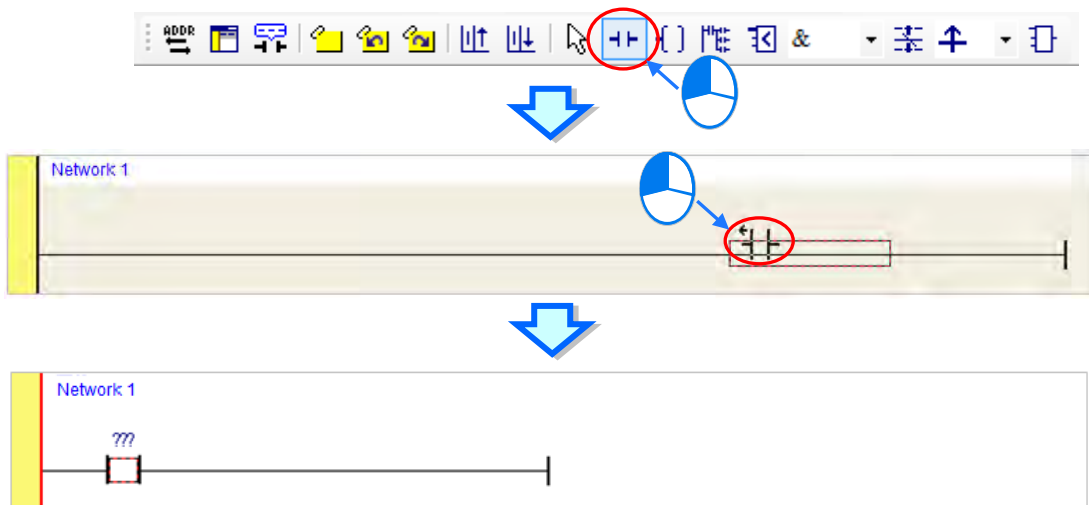
\*. Please refer to section 4.5.3 for more information about typing an instruction.


### 4.5.2 Basic Editing - Creating a Contact and a Coil

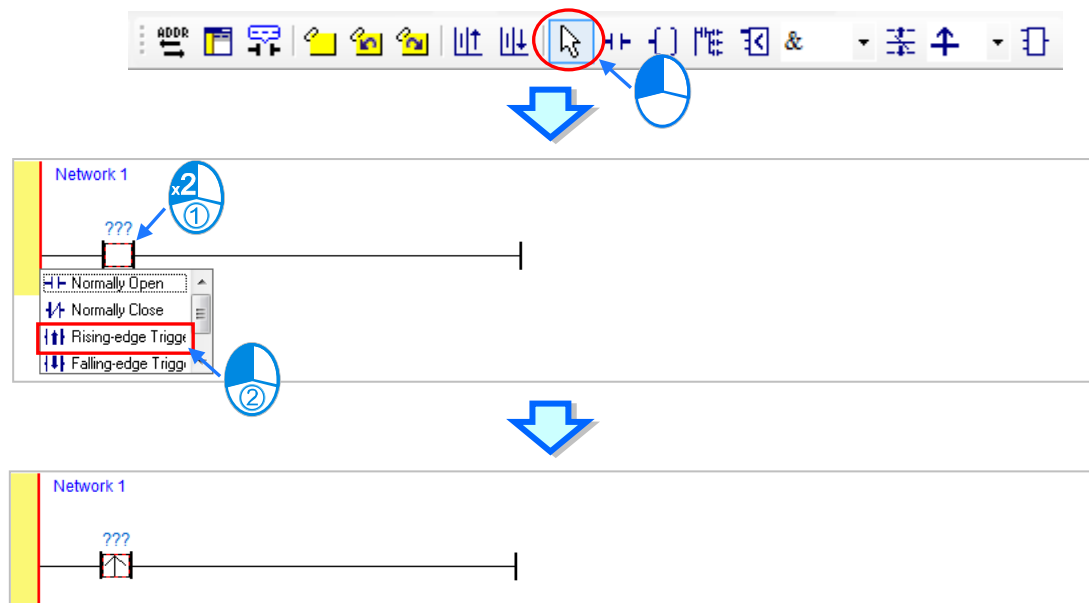
- Click  on the toolbar, and then move the mouse cursor to the red frame in network 1. The mouse cursor appears as a contact when the mouse cursor is moved to the left side of the red frame, the right side of the red frame, or the bottom of the red frame. Users can decide where to insert a contact. If 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.

4

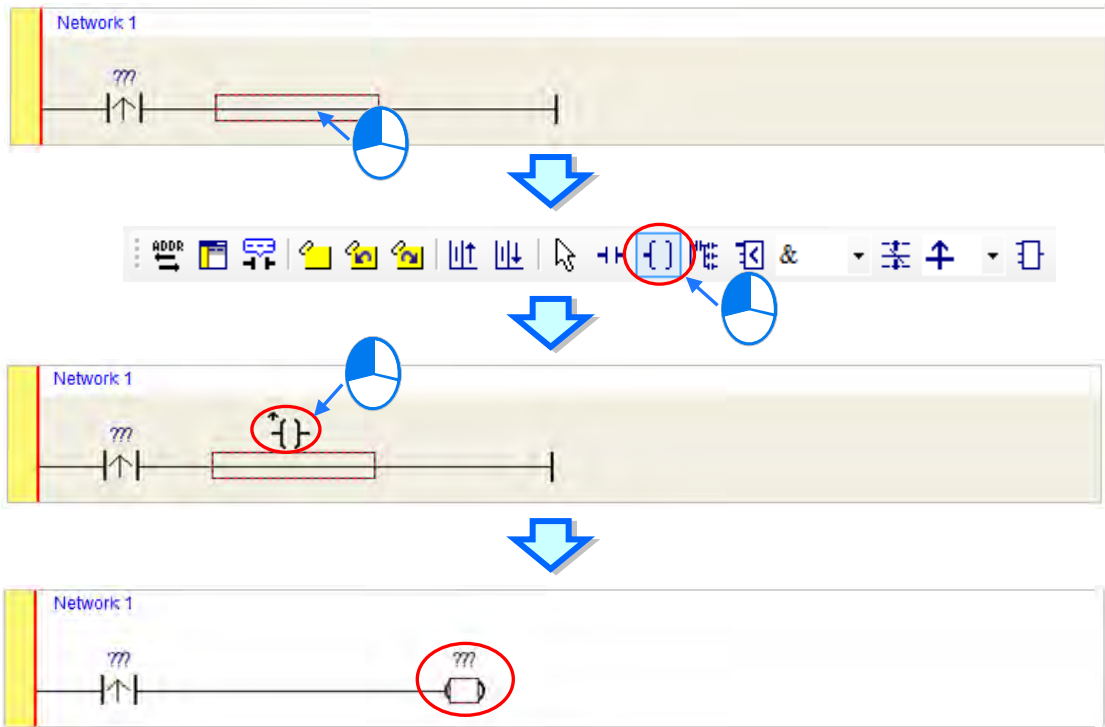



- Click  on the toolbar, or press Esc on the keyboard. After the contact is double-clicked, a drop-down list will appear. The items on the drop-down list are **Normally Open**, **Normally Close**, **Rising-edge Trigger**, and **Falling-edge Trigger**. In this example, **Rising-edge Trigger** is selected.

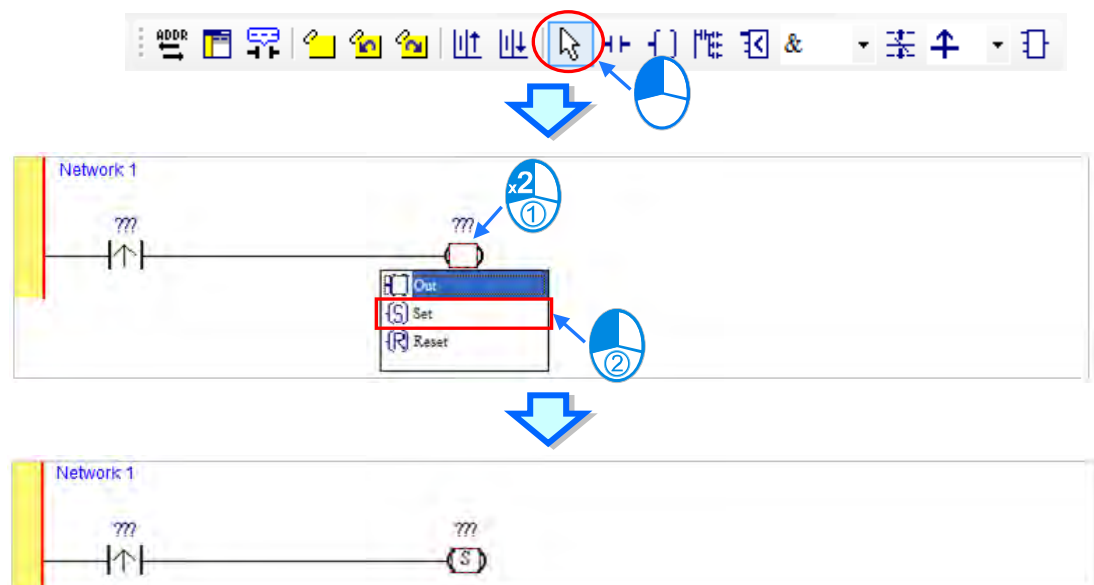


- (3) Click the line at the right side of the contact, click  on the toolbar, and move the mouse cursor to the red frame. Likewise, the mouse cursor appears as a coil when the mouse cursor is above or under the red frame. Users can decide where to insert the coil.

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.

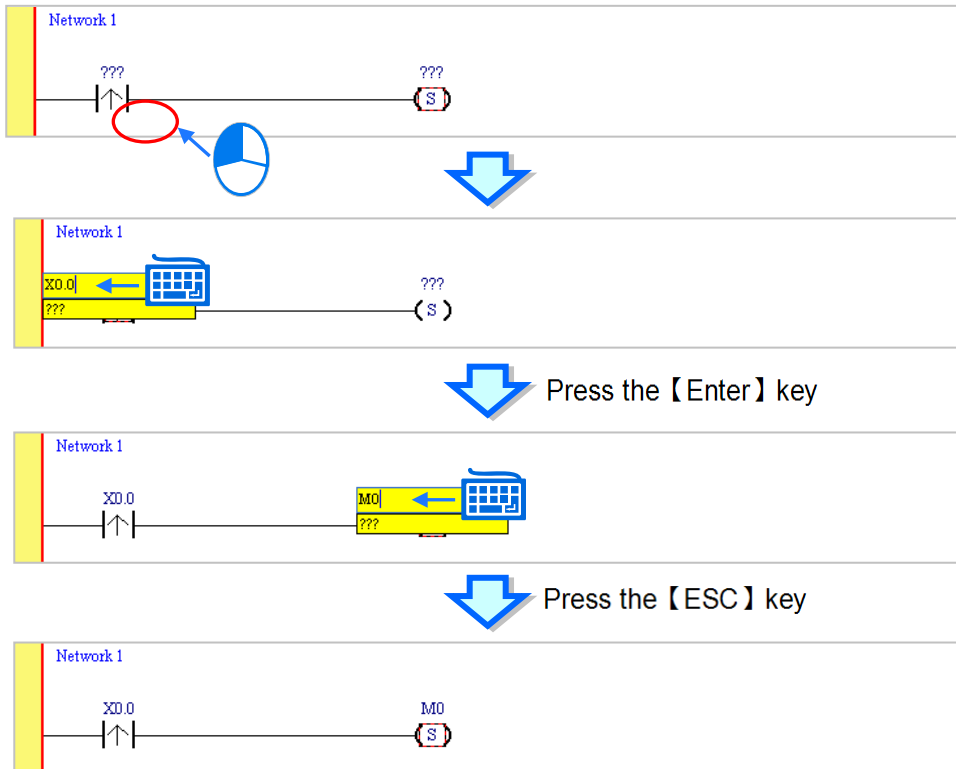


- (4) Click  on the toolbar, or press Esc on the keyboard. After the coil is double-clicked, a drop-down list will appear. The items on the drop-down list are **Out**, **Set**, and **Reset**. In this example, **Set** is selected.



- (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.

4

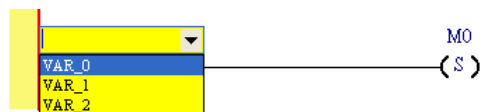


**Additional remark**



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.

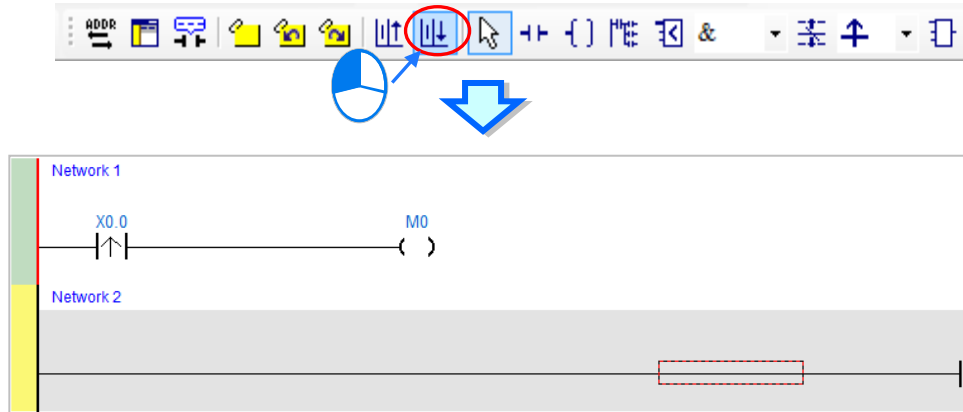
If the users have declared symbols, users can type the symbols or clear the field and use the drop-down list to select the symbols which can be assigned to the object.



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.



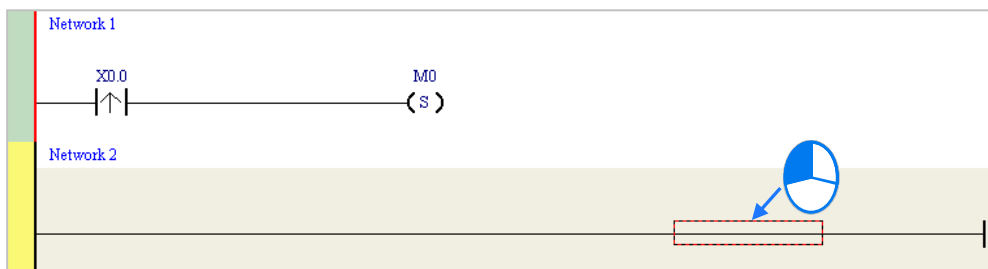
### 4.5.3 Basic Editing - Inserting a Network and Typing an Instruction

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.

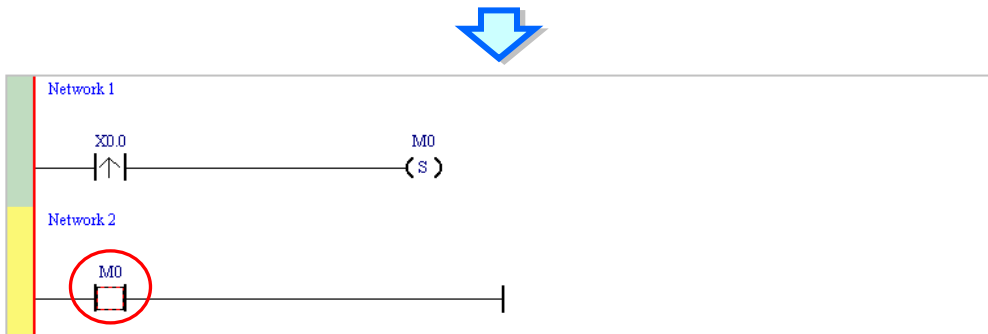
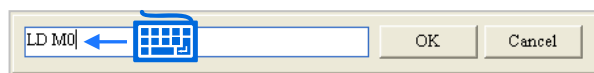


A contact and a coil can be created not only by clicking  and  on the toolbar, but also by typing instructions.

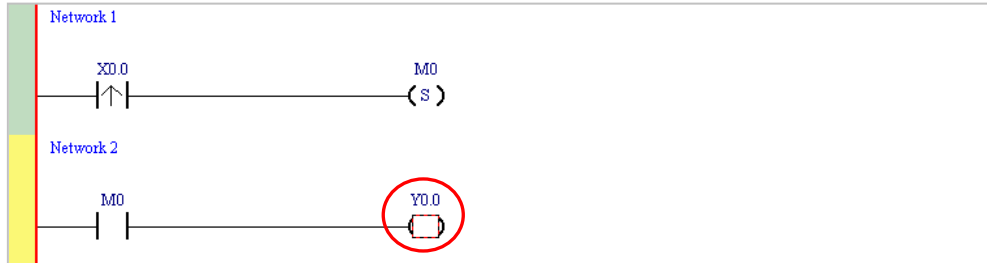
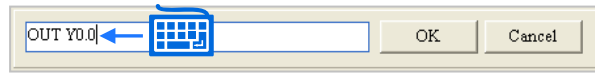
- (1) Click the line in network 2.



- (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.

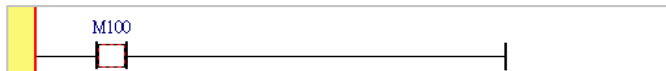
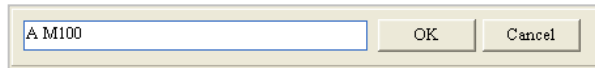


**Additional remark**

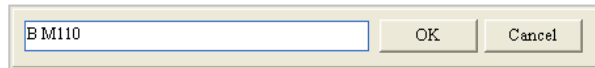
4

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

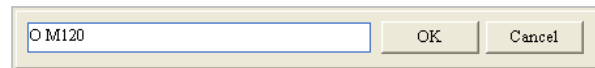
- Inserting a normally-open contact (contact A): “A Device address”



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

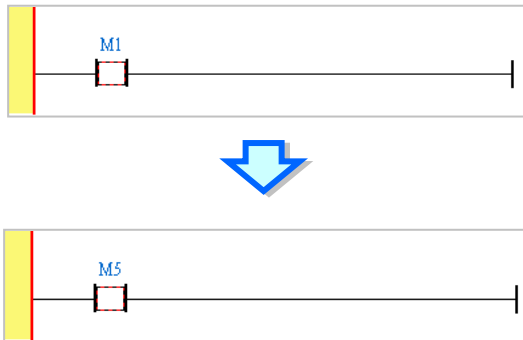


- Inserting an output coil (OUT): “O Device address”




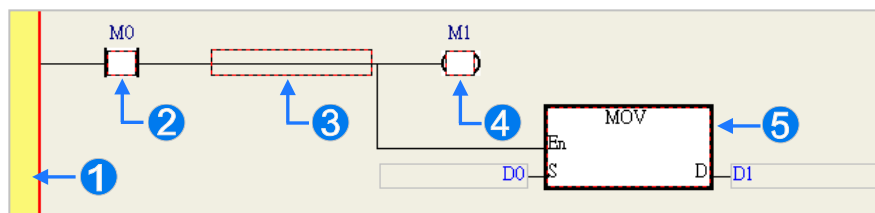


When adding devices, press [Insert] on the keyboard to choose **Insert** or **Replace** mode. Use [Insert] on the keyboard to input LD M5 on M1 position and M1 is replaced by M5 (see below).



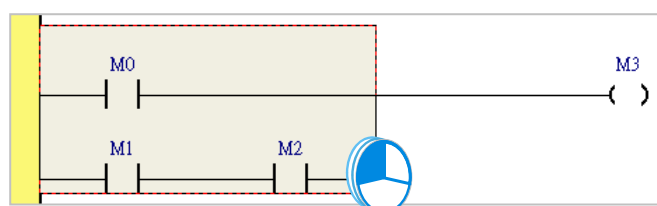
#### 4.5.4 Basic Editing - Selection of a Network and Operation

When selecting in the ladder diagram, users can press [Esc] on the keyboard, or click  from the toolbar. After the mouse cursor appears in select mode, click the selected item. Basic operating steps are shown below.

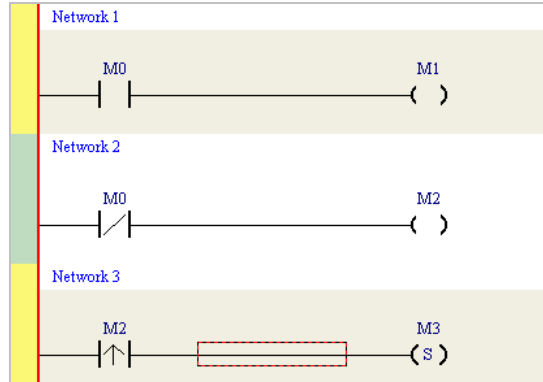


- ❶ Select the entire network
- ❷ Select an input
- ❸ Select the section
- ❹ Select the output coil
- ❺ Select the block

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 cannot be in the same frame.



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.



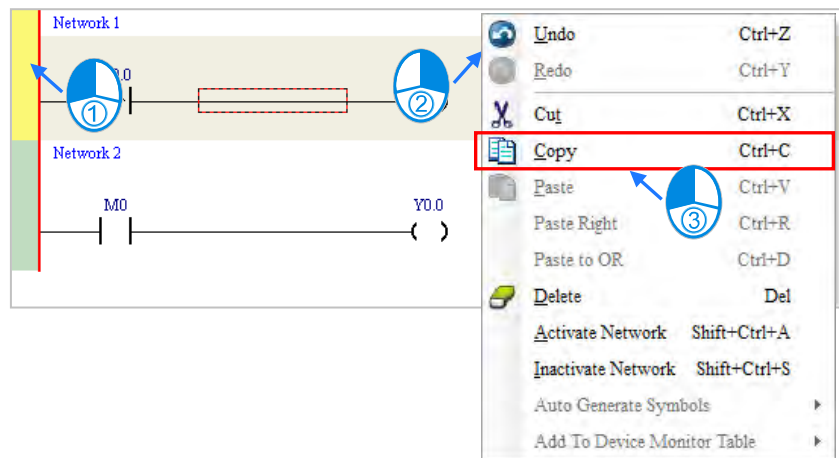
# 4

Right-click the selected item and execute basic operations from the quick start menu.

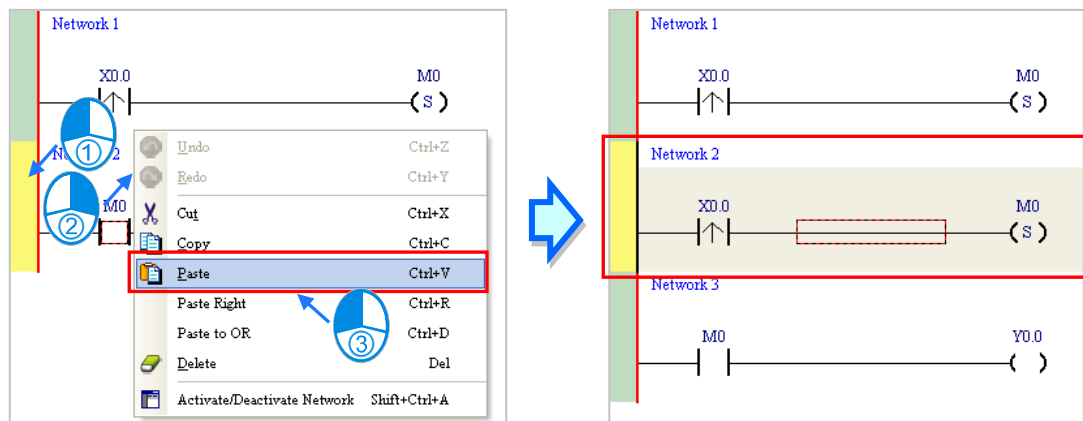
Item	Function
<b>Undo</b>	Undoing the last action (The number of previous actions that can be undone is 20.)
<b>Redo</b>	Redoing an action which has been undone
<b>Cut</b>	Cutting a device, a block, or a network
<b>Copy</b>	Copying a device, a block, or a network
<b>Paste</b>	Paste an object which has been copied or cut on the present position
<b>Paste right</b>	Pasting an object at the right side of the position selected (The object will be connected to the position selected in series.)
<b>Paste under</b>	Pasting an object under the position selected (The object will be connected to the position selected in parallel.)
<b>Delete</b>	Deleting a device, a block, or a network
<b>Activate Network</b>	Set the selected network to activate
<b>Inactivate Network</b>	Set the selected network to Inactivate
<b>Variable Generated</b>	To generate variable automatically and use on pins for function blocks
<b>Add to Device Monitoring Table</b>	Use on the selected pins to quickly add the device in the monitoring table

Users can proceed with the operation in the example.



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



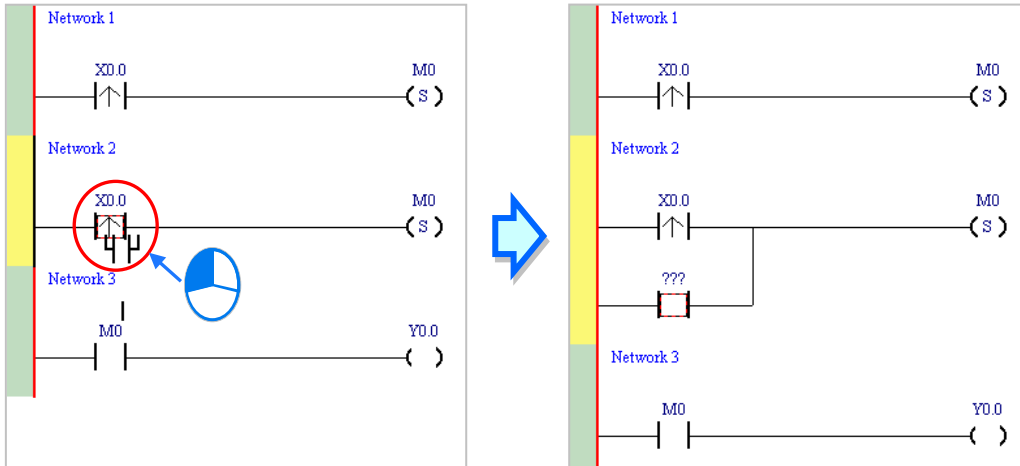
- (2) Select network 2, right-click network 2, and click **Paste** on the context menu. A copy of network 1 will be put above network 2, and network 2 will become network 3.



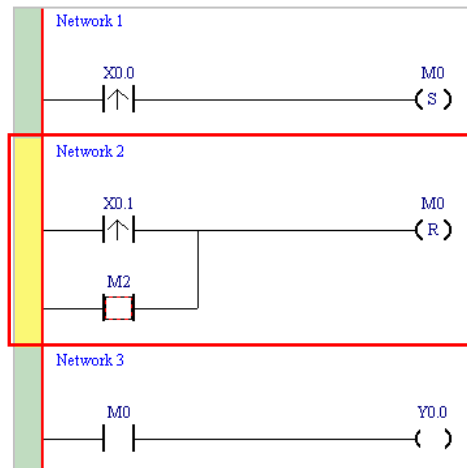
### 4.5.5 Basic Editing - Connecting a Contact in Parallel

- 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.

4

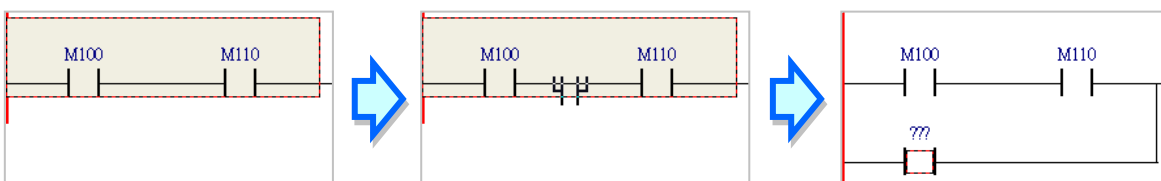


- Write the program in network 2 shown below in the way described above.




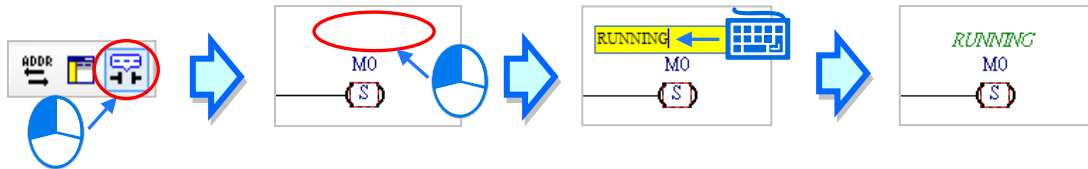
**Additional remark**


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

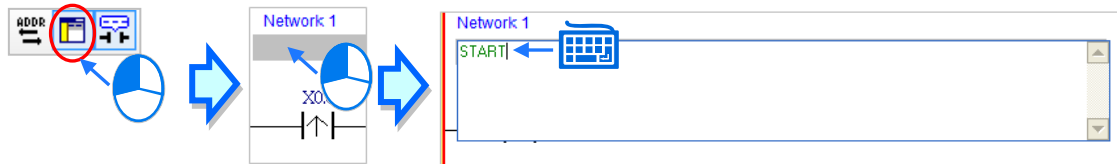


### 4.5.6 Basic Editing - Editing a Comment

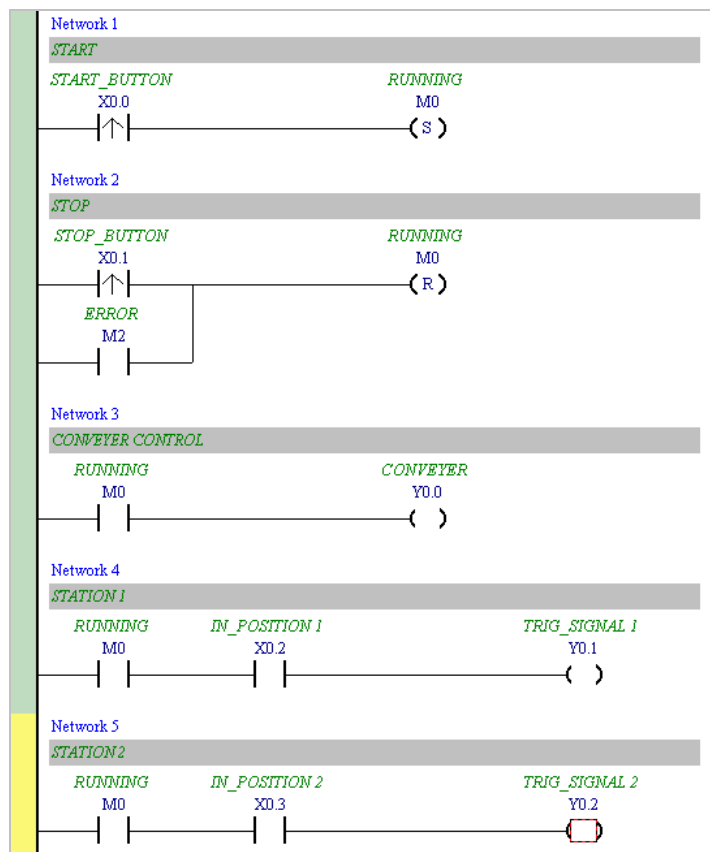
- (1) Make sure that  on the toolbar is pressed. Click the position above a device name, type a comment in the box, and press Enter on the keyboard.



- (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.



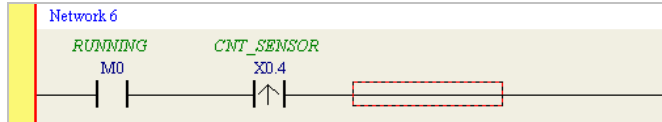
- (3) Write the program shown below in the way described above.



4

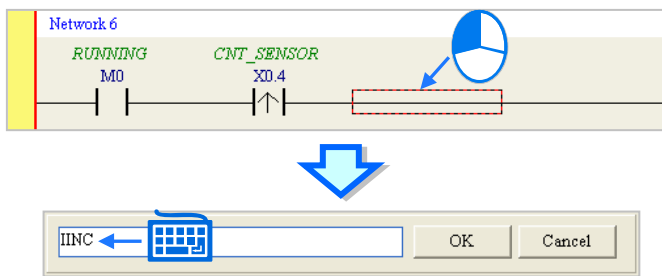
### 4.5.7 Basic Editing - Inserting an Applied Instruction

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.



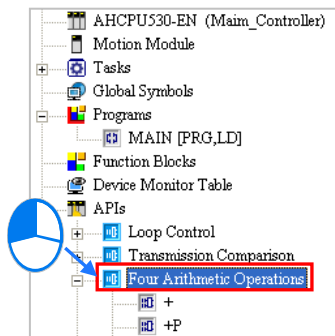
● Method 1

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

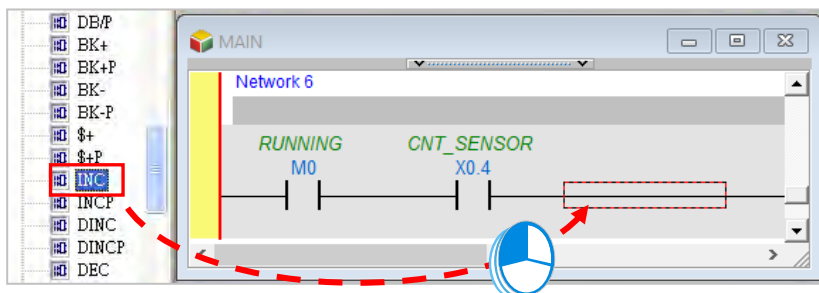


● Method 2


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

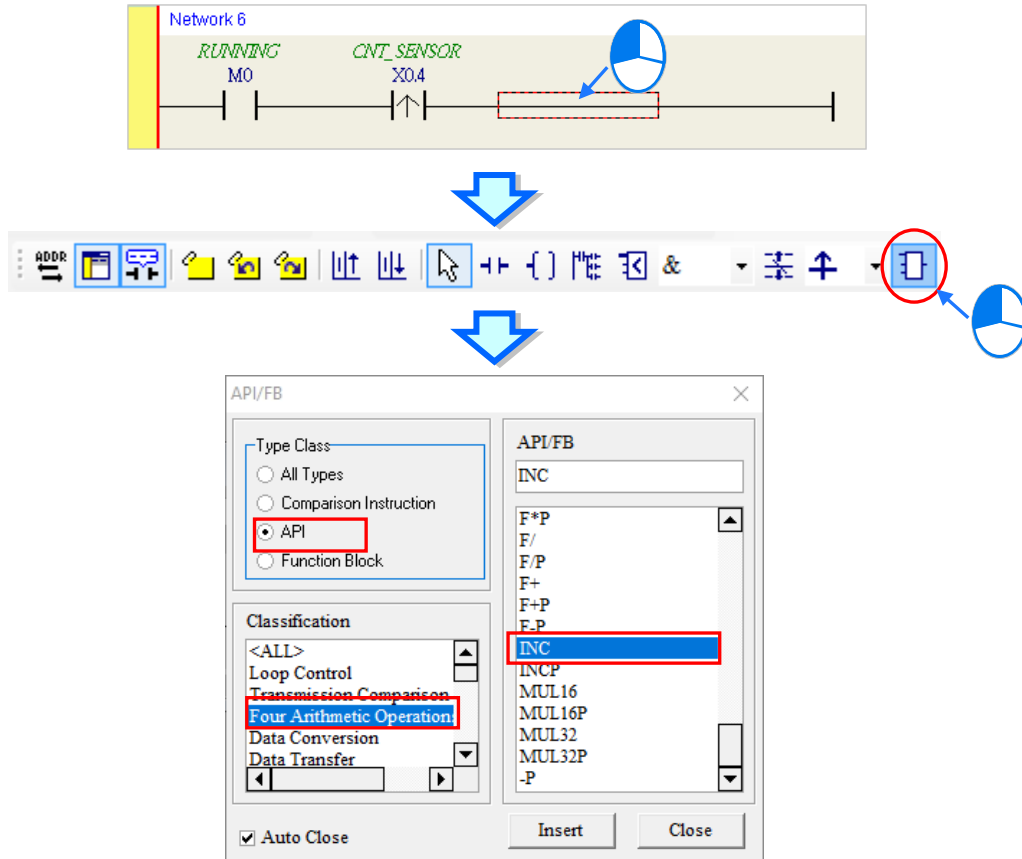


Select the instruction (INC in this example) which will be inserted, and then drag it to the position where it will be inserted.



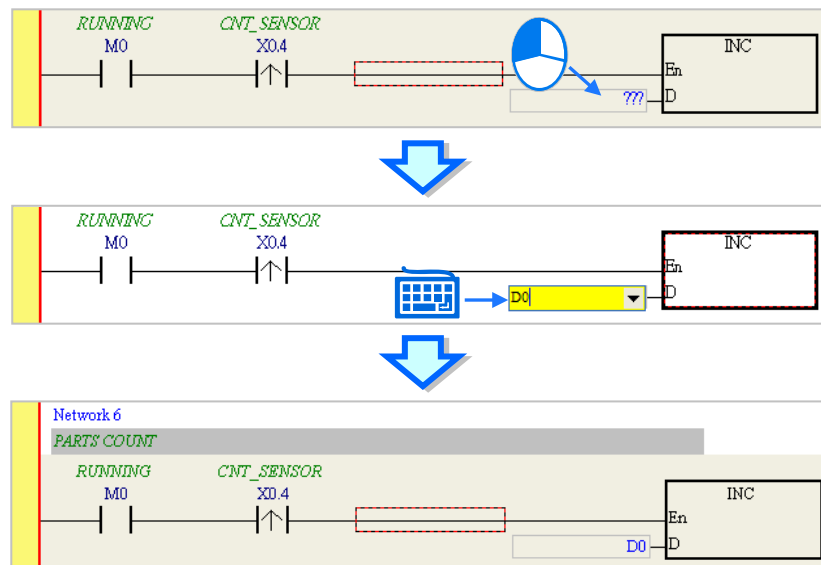
● Method 3

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**.



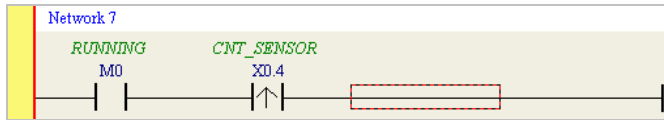
4


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

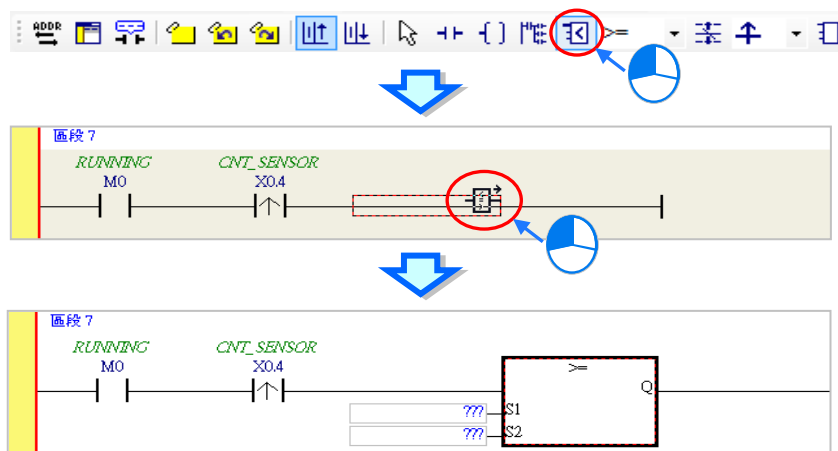
A comparison contact can be inserted not only in one of the three ways described in section 4.5.7, but also by means of the following steps. Users need to put network 7 under network 6, and write the program shown below.



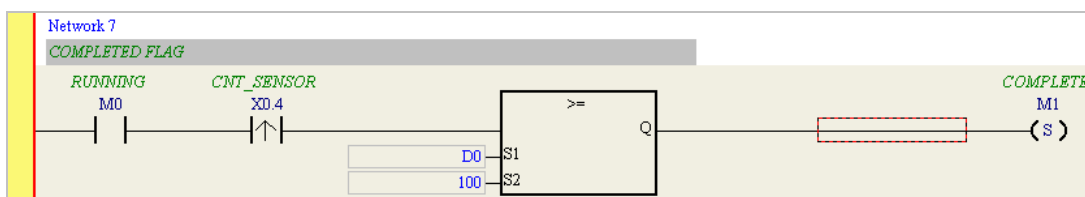
- (1) Click  on the toolbar, and then select a type (>= in this example).



- (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 mouse 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.



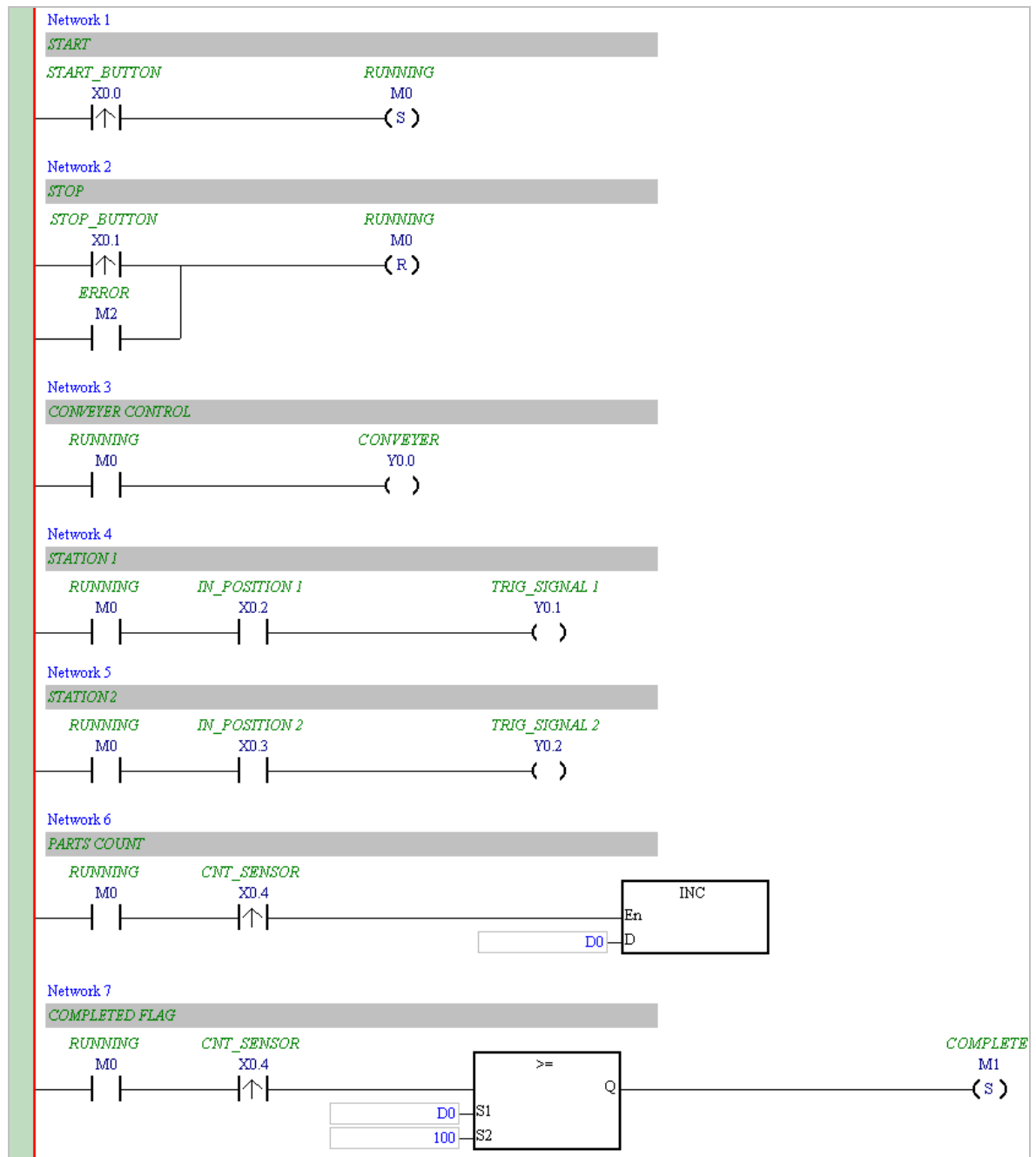
Write the program shown below in the way described above. In WPLSoft, a decimal value is preceded by K, and a hexadecimal value is preceded by H. If users want to type a decimal value in ISPSOft, they can type it directly. If users want to type a hexadecimal value in ISPSOft, they have to type "16#" 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. Owing 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 10 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.

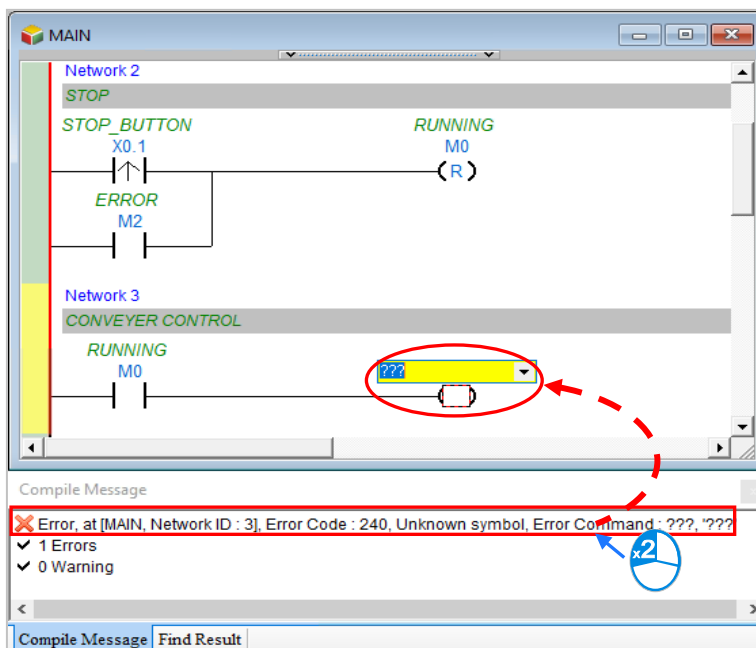


● **Compile**

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.



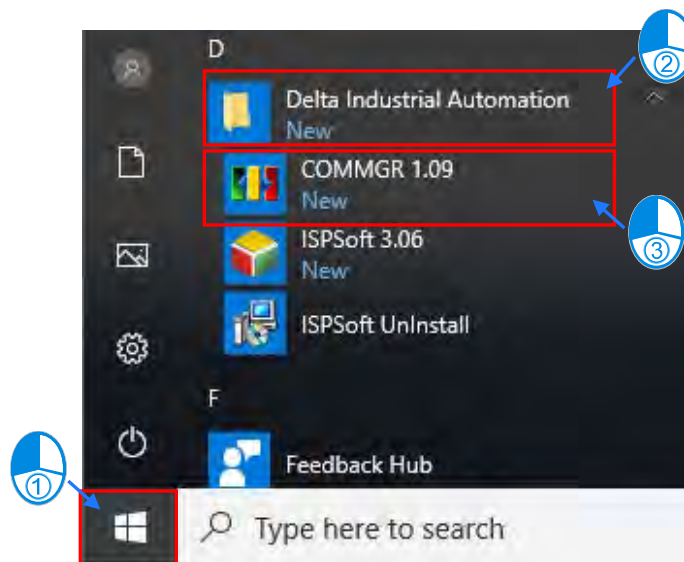
## 4.6 Testing and Debugging a Program

### 4.6.1 Creating a Connection

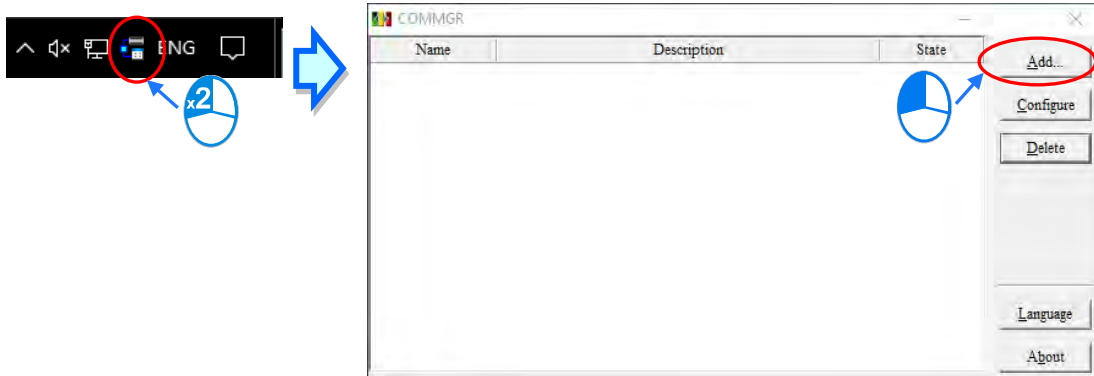
Before a program and parameters are downloaded to a PLC or monitored online, ISPSoft must be connected to the PLC. In this example, ISPSoft is connected to AHCPU530-EN host CPU through a USB cable.

For those already connected to the host, please skip to section 4.6.2 on details regarding download program and configuration parameters.

- (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**).

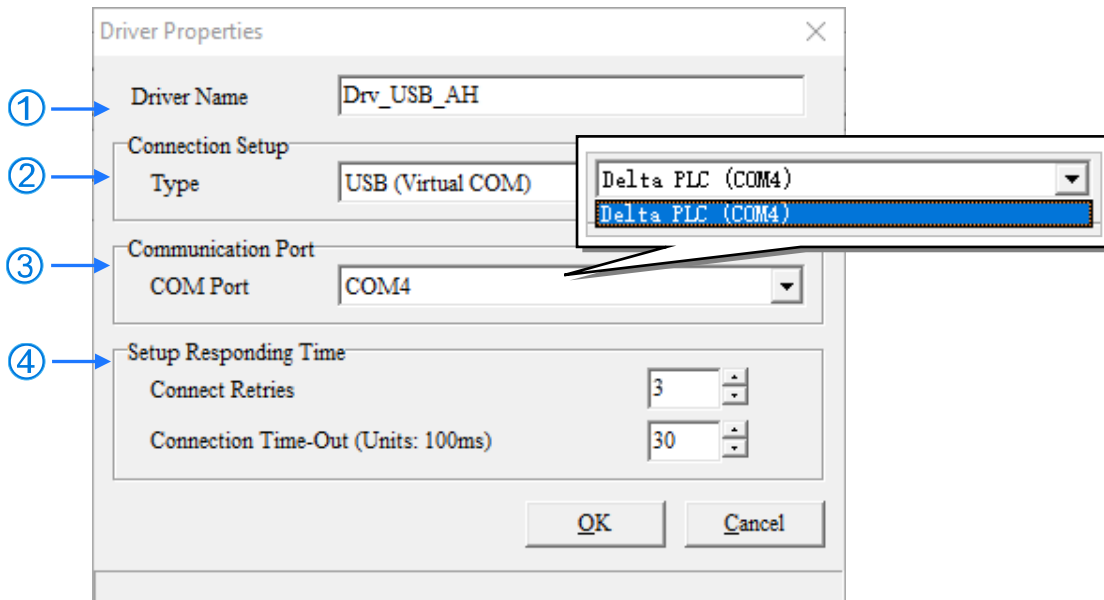


- (4) Double-click the icon representing COMMGR on the system tray to open the **COMMGR** window. Click **Add** in the **COMMGR** window to create a driver.



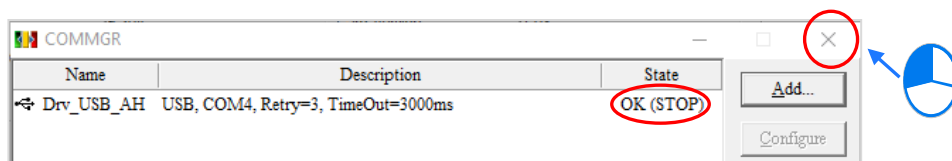
- (5) Set the parameters in the **Driver Properties** window, and then click **OK**.

4

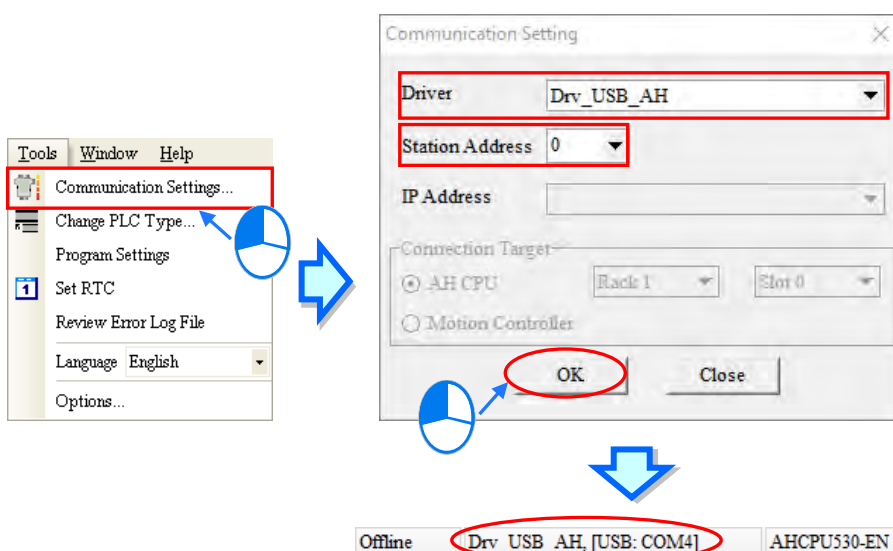


- ① Type a driver name in the **Driver Name** box.
- ② Select **USB (Virtual COM)** in the **Type** drop-down list box in the **Connection Setup** section.
- ③ Select a communication port in the **COM Port** drop-down list box. If the first two steps are complete, the PLC which is connected and its communication 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.

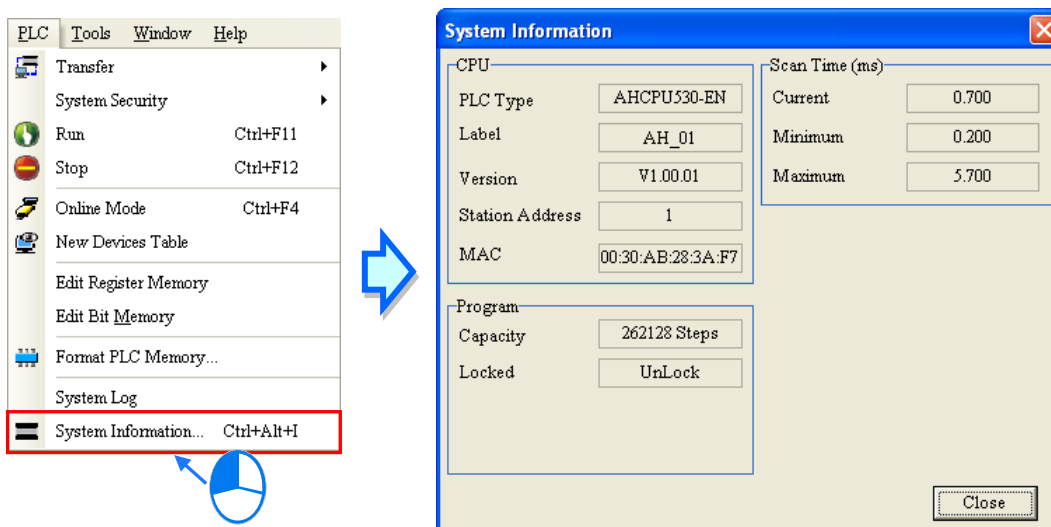
- (6) After the status of the driver displayed in the window becomes **OK**, the window can be closed. The icon representing COMMGR will still be displayed on the system tray.



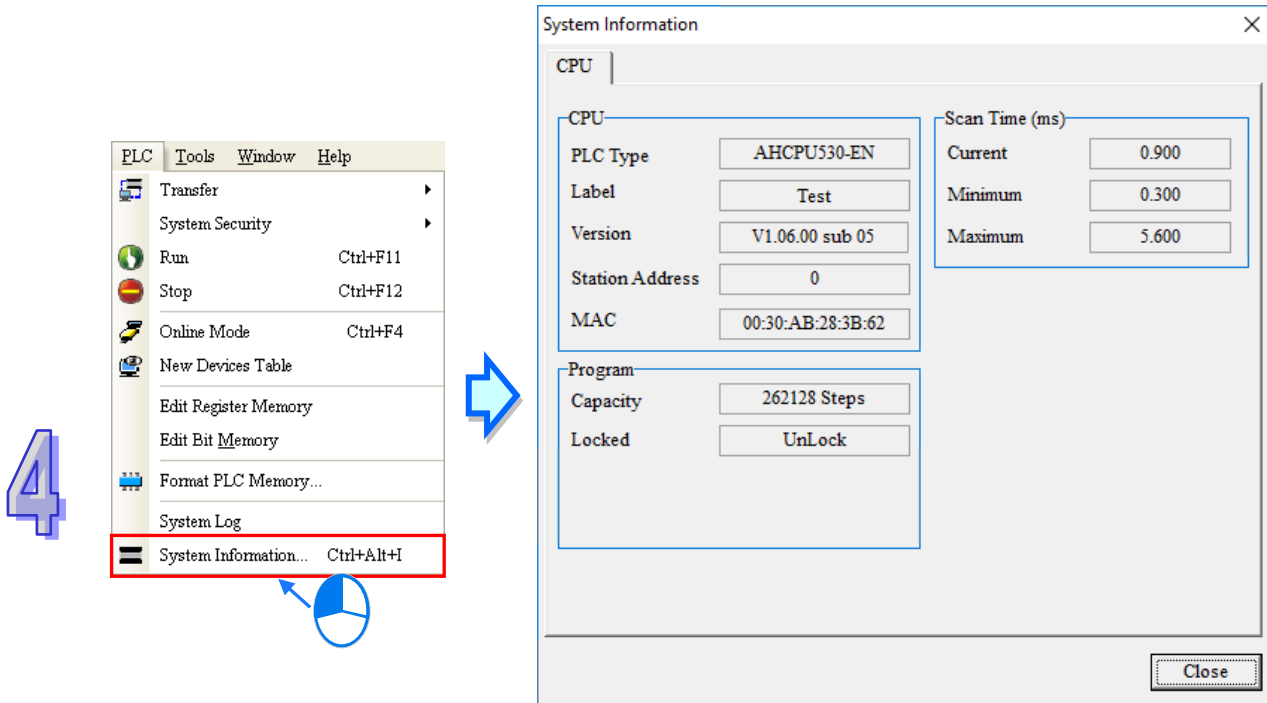
Start ISPSOft, and then click **Communication Settings...** on the **Tools** menu. In the **Communication Setting** window, select the driver which has been created in the **Driver** drop-down list box, appear, and select 0 in the **Station Address** drop-down list box, and click **OK**. The information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



4



- (7) 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.

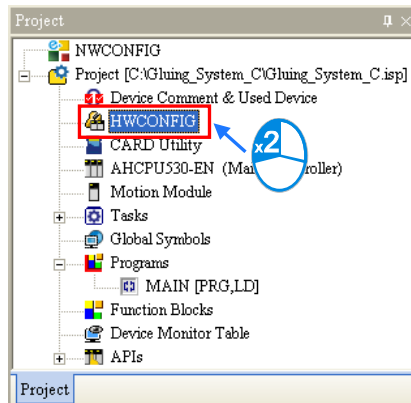


#### 4.6.2 Downloading a Program and Parameters

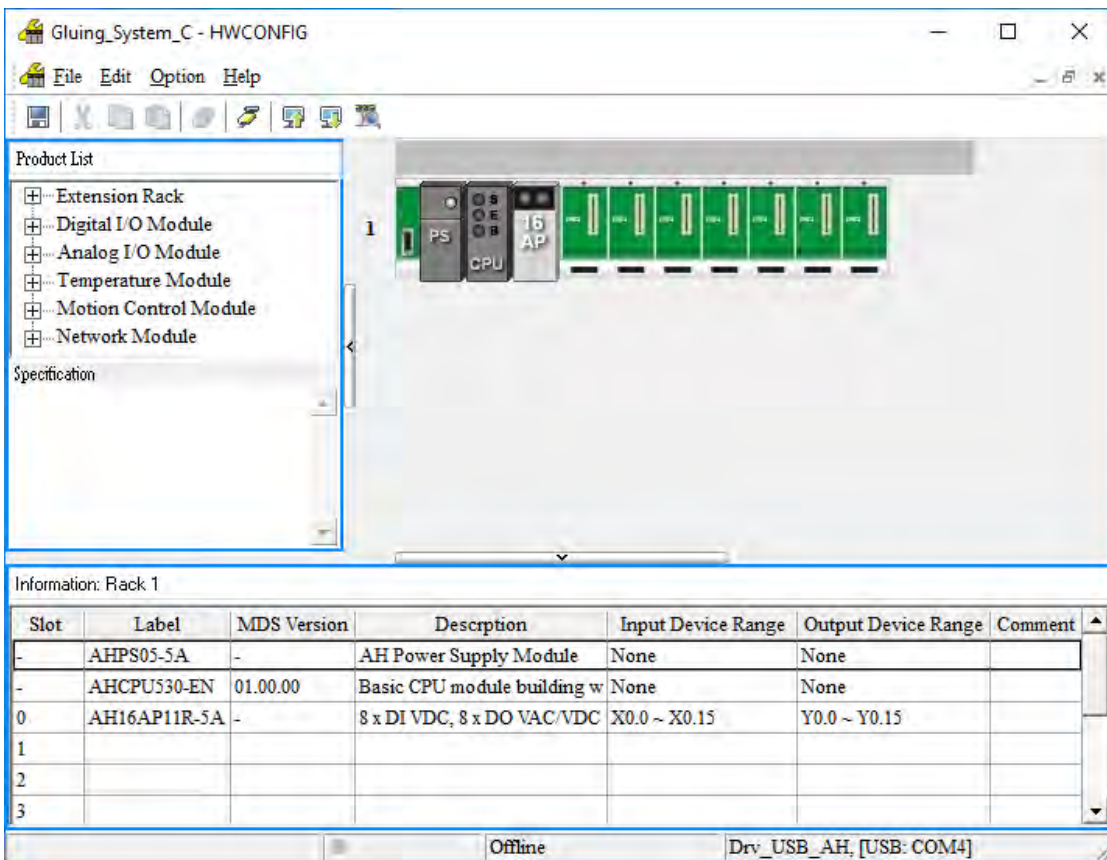
If ISPSOft is connected to a PLC normally, the parameters and the program in the project can be downloaded to the PLC. First, start ISPSOft and open the project created in the previous sections. In this example, two types of parameters are downloaded to the CPU module. They are the hardware configuration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

- **Downloading the hardware configuration**


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

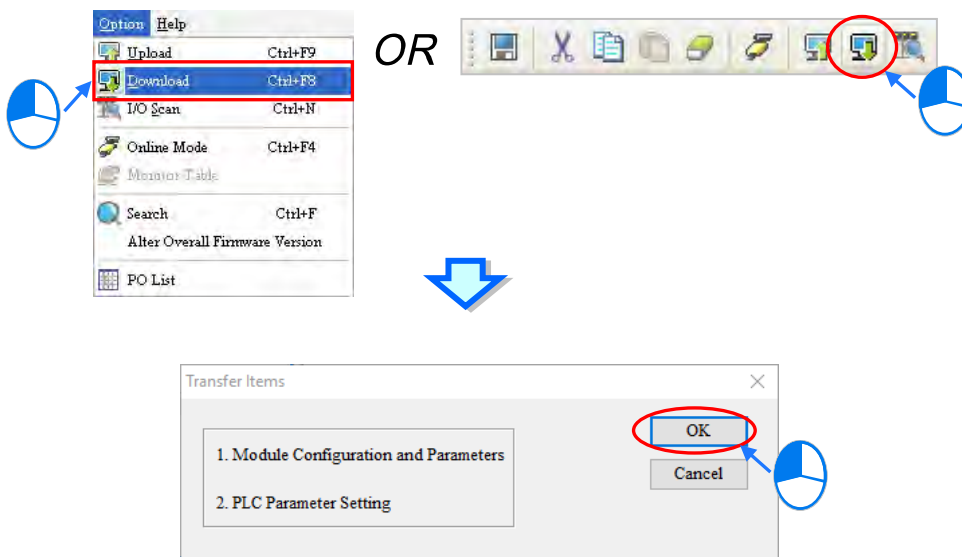


- (2) The hardware configuration is displayed in the window. Before the hardware configuration is downloaded to the CPU module, users have to make sure that the actual hardware configuration is the same as the hardware configuration in the window.




4

- (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.

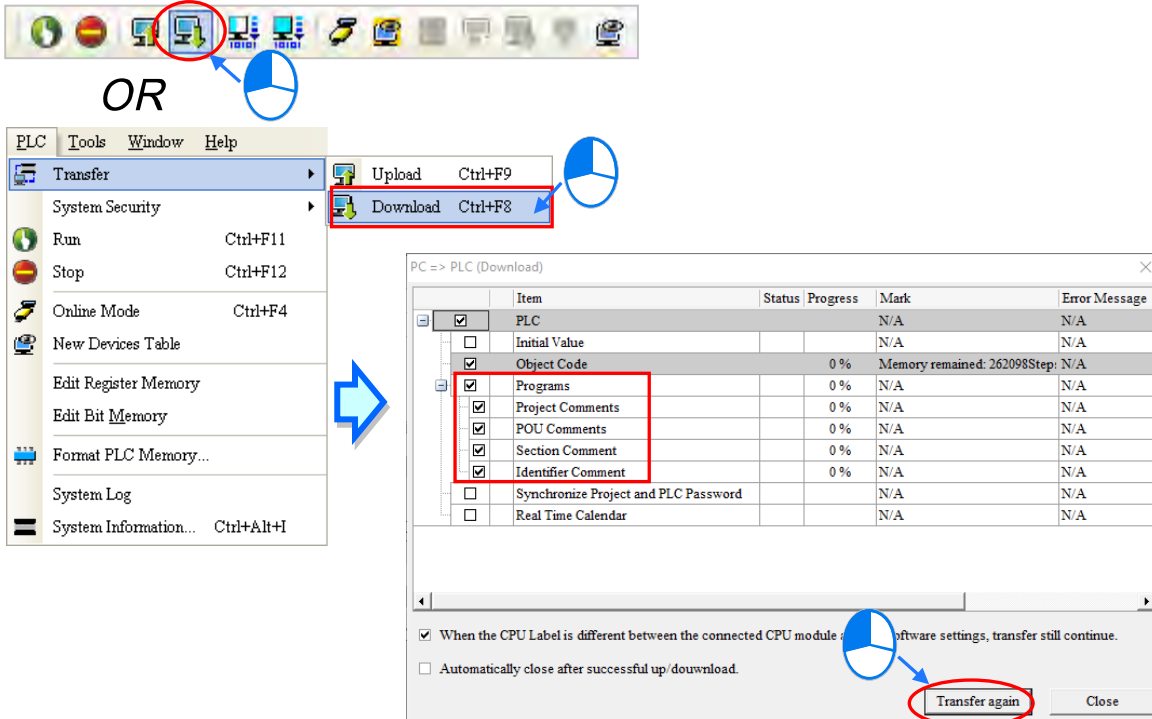


- (4) After the hardware configuration is downloaded to the CPU module successfully, the BUS FAULT LED indicator on the CPU module will be OFF. The users can close the **HWCONFIG** window. If the BUS FAULT LED indicator on the CPU module is still ON or blinking, the CPU module is in an abnormal state. 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** and related **Comments** checkboxes in the **Transfer Setup** window so that the program in the CPU module can be uploaded later, and then click **OK**.

4



OR

Item	Status	Progress	Mark	Error Message
<input checked="" type="checkbox"/> PLC			N/A	N/A
<input type="checkbox"/> Initial Value			N/A	N/A
<input checked="" type="checkbox"/> Object Code		0 %	Memory remained: 262098Step: N/A	N/A
<input checked="" type="checkbox"/> Programs		0 %	N/A	N/A
<input checked="" type="checkbox"/> Project Comments		0 %	N/A	N/A
<input checked="" type="checkbox"/> POU Comments		0 %	N/A	N/A
<input checked="" type="checkbox"/> Section Comment		0 %	N/A	N/A
<input checked="" type="checkbox"/> Identifier Comment		0 %	N/A	N/A
<input type="checkbox"/> Synchronize Project and PLC Password			N/A	N/A
<input type="checkbox"/> Real Time Calendar			N/A	N/A



When the CPU Label is different between the connected CPU module and software settings, transfer still continue.  
 Automatically close after successful up/download.

**Transfer again**    Close




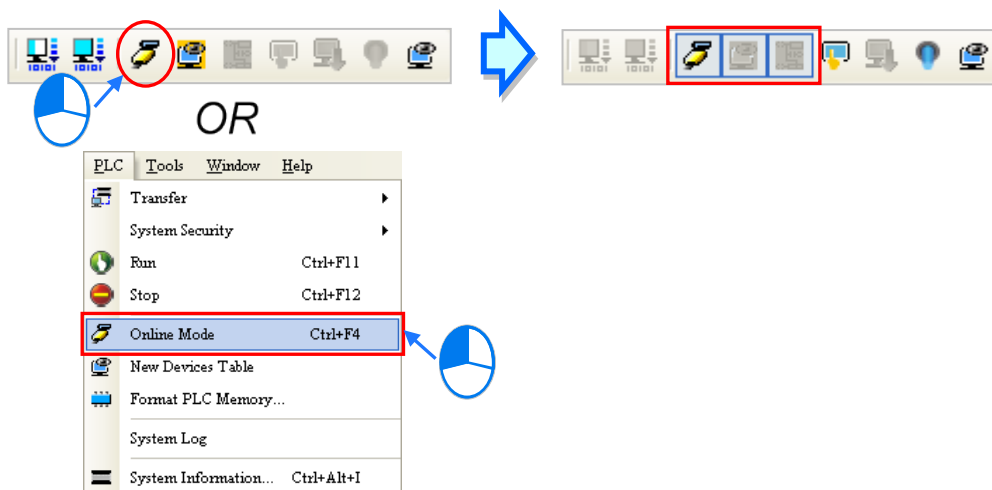
### 4.6.3 Connection Test

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.

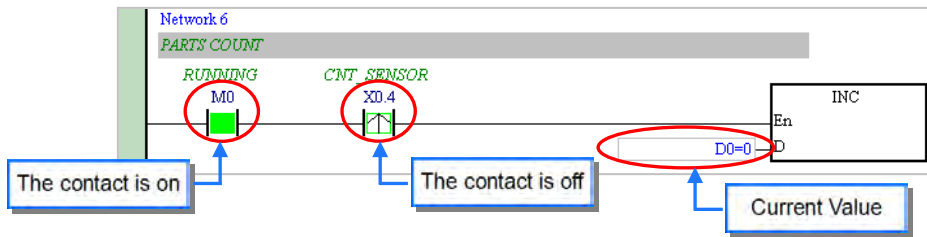
After users click **Online Mode** on the **PLC** menu, or  on the toolbar, the online monitoring function will be enabled. The system will also enable the device monitoring mode and the program monitoring mode.





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.

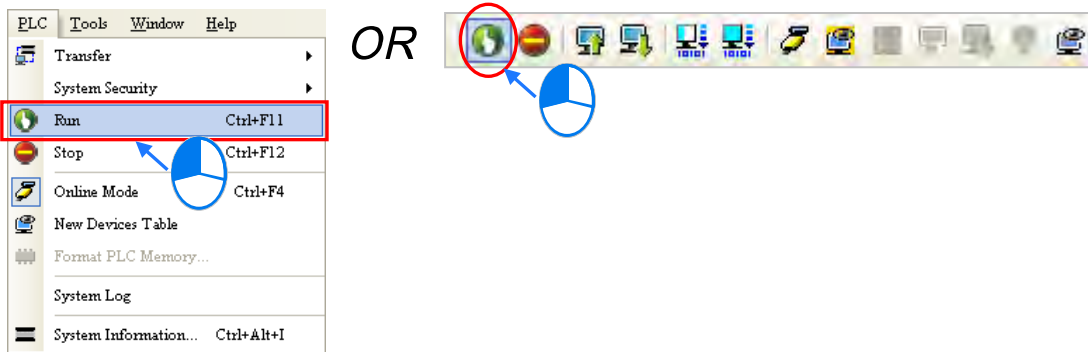


Besides, the present statuses of the devices will be displayed in the original program editing window after the program monitoring function is enabled.



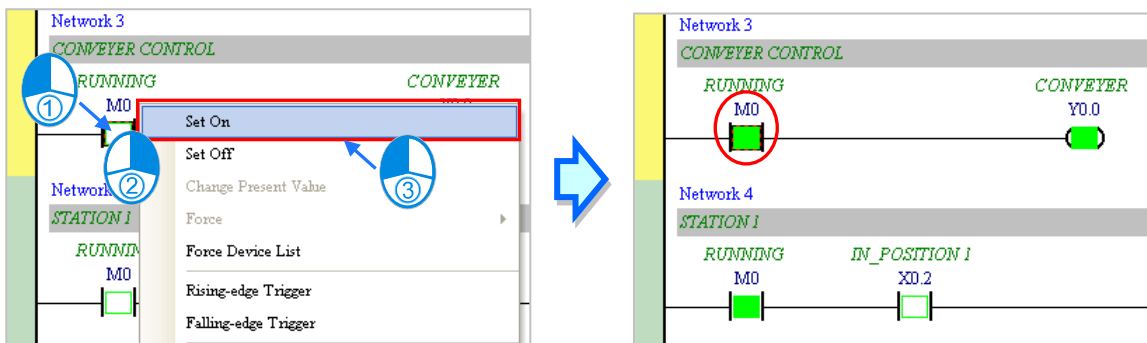
Users can change the operating status of a PLC by the RUN/STOP switch on the PLC. They can also change the operating status of the PLC through the functions provided by ISPSoft. After users click **Run** on the **PLC** menu or  on the toolbar, the PLC will begin to run. The PLC will stop running after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

4



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.**

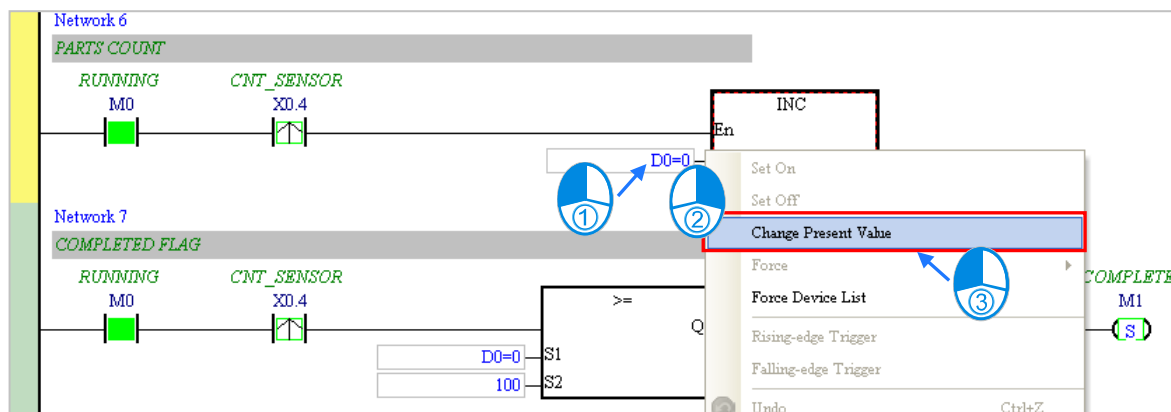


The items on the context menu are described below. **Force** on the context menu only applies to input contacts and output contacts.

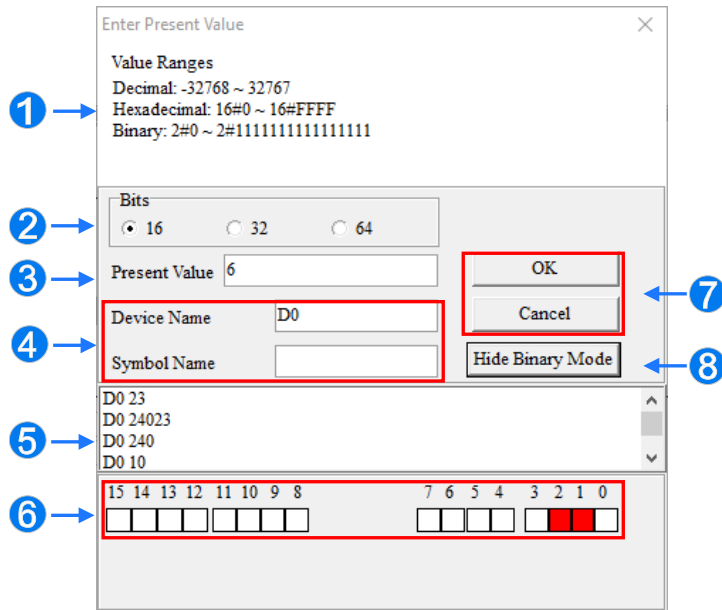
Item	Description
Set On	Setting the contact selected to ON
Set Off	Setting the contact selected to OFF
Rising-edge Trigger	No matter what the state of the contact selected is, the system set the contact to OFF, and then set it to ON.
Falling-edge Trigger	No matter what the state of the contact selected 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

4

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.

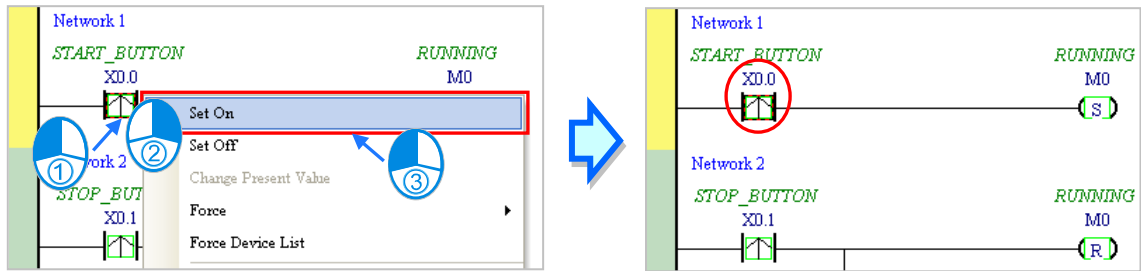


The **Enter Present Value** window is described below.

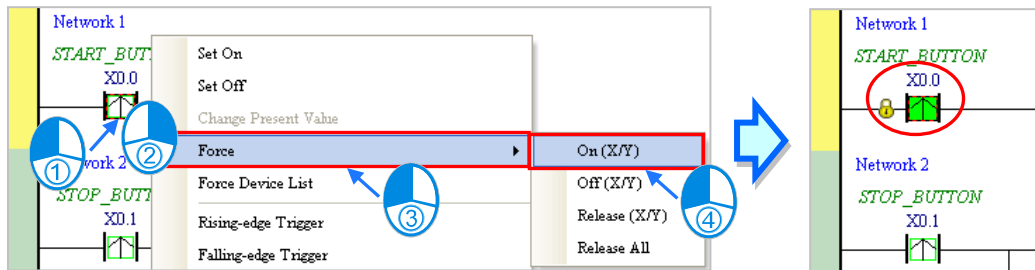


- 1 Message text.
- 2 Select input number format: 16 bits/32 bits/64 bits.
- 3 Input present value for modification.
- 4 To modify present device name or symbol name.
- 5 Log for modified number (Format shown: Device name+ modified value)
- 6 When binary mode is enabled, use the mouse to set ON/OFF state for each bit.
- 7 Click **OK** to apply the setting value, and click **Cancel** to close the setting window.
- 8 Click to open or hide 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.



4

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.

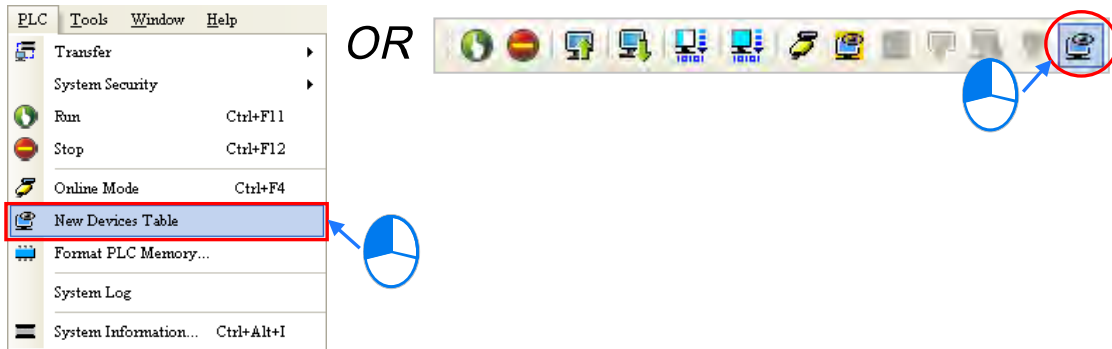


\*. If the online monitoring function is disabled, the contacts will not be automatically released from the locked states. As a result, users have to check whether the contacts need to be released from the locked states after the test is complete.

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

● **Method 1**

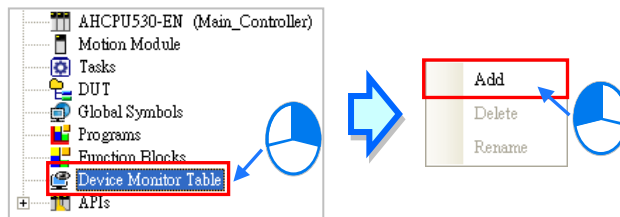
Click **New Devices Table** on the **PLC** menu, or  on the toolbar.



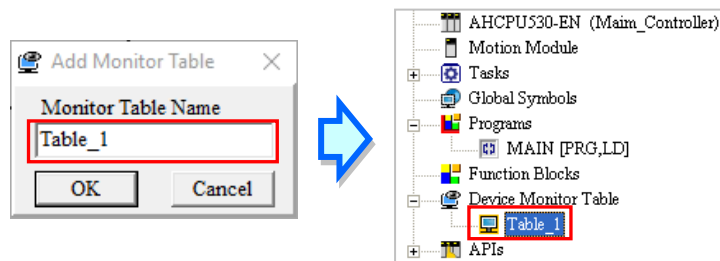
4

● **Method 2**

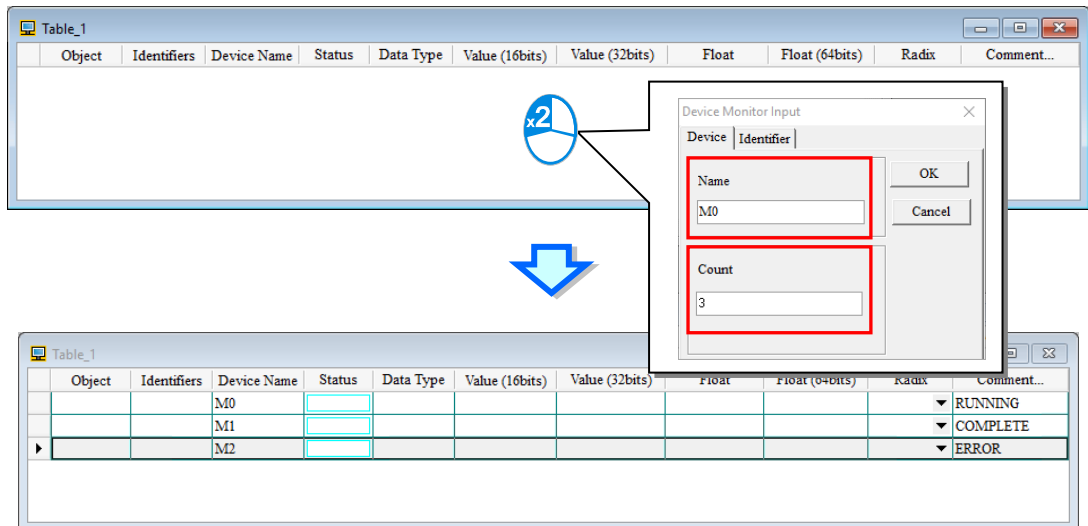
Right-click **Device Monitoring Table** in the project management area and click **Add**.



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. 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.



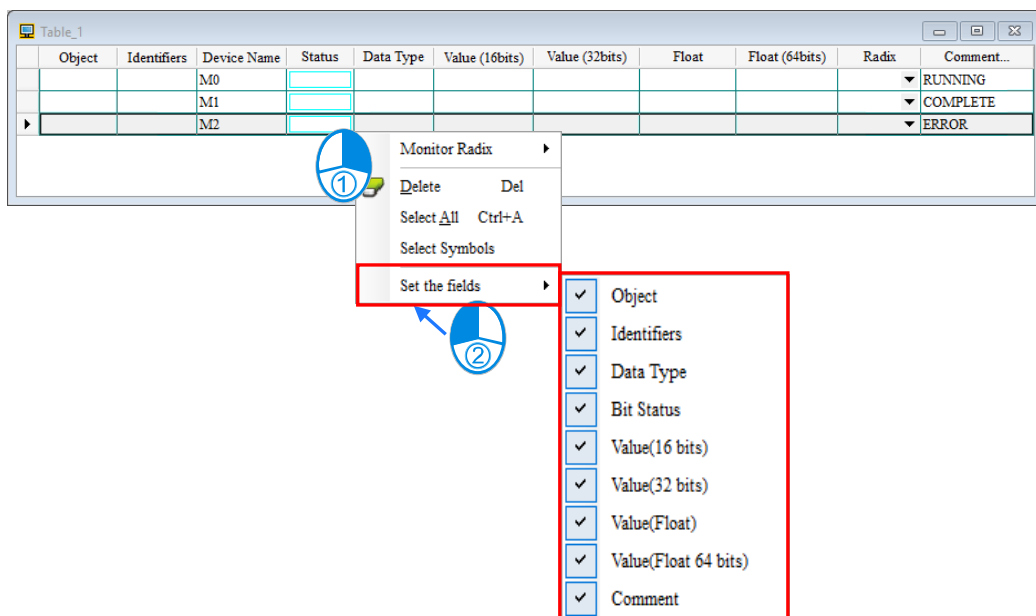
When adding a device item, press [Insert] on the keyboard to select **Insert** or **Replace** mode; users can find the current selected state in the ISPSoft status column.

4

When choosing **Insertion** mode, the item added will be on top of the selected item in the monitoring table; when choosing **Replace** mode, the item added will overwrite the selected item 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.

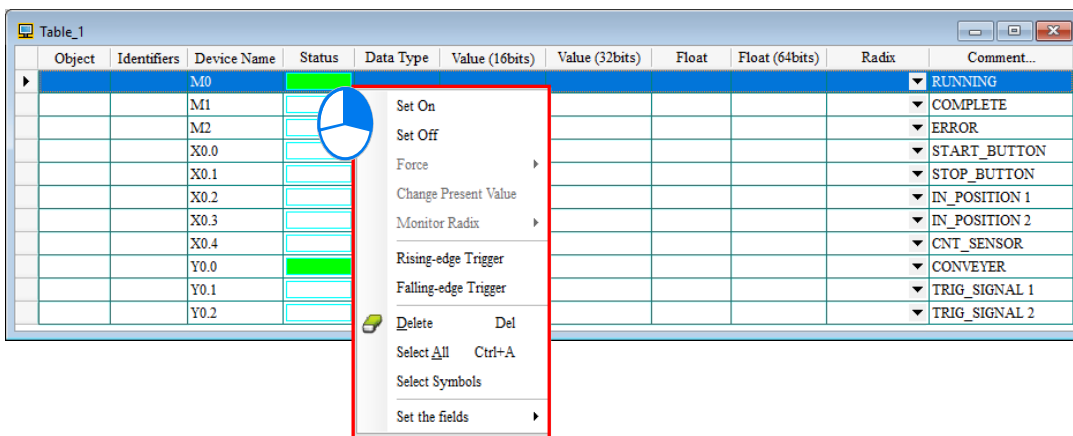


Description on the columns of the monitoring table.

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 (32-bit floating-point )	In the online mode, a 32-bit floating-point number is displayed.
Float(64-bit floating-point )	In the online mode, a 64-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.

4

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 18 for more information about testing and debugging a program.



# 5

## Chapter 5 POU and Task

### Table of Contents

<b>5.1</b>	<b>Program Organization Units and Tasks</b> .....	<b>5-2</b>
<b>5.2</b>	<b>Program Organization Units</b> .....	<b>5-3</b>
5.2.1	Program Architecture and Types .....	5-3
5.2.2	POUs in ISPSOft .....	5-4
<b>5.3</b>	<b>Tasks</b> .....	<b>5-6</b>
5.3.1	Managing the Tasks in ISPSOft.....	5-6
5.3.2	Tasks in the Project Management Area .....	5-11
5.3.3	Executing the POUs Assigned to a Task.....	5-13
<b>5.4</b>	<b>Managing a POU</b> .....	<b>5-16</b>
5.4.1	Creating and Setting a POU .....	5-16
5.4.2	POU Active State .....	5-19
5.4.3	Delete and Copy POU .....	5-20
5.4.4	Set and Remove POU Password .....	5-22
5.4.5	Exporting POU Program .....	5-23
5.4.6	Importing POU Program.....	5-24
5.4.7	Exporting Function Block POU .....	5-25
5.4.8	Importing Function Block POU .....	5-26
<b>5.5</b>	<b>Task Management</b> .....	<b>5-28</b>
5.5.1	Setting Task Description and Condition for Interruption .....	5-30
5.5.2	Configuration of POU .....	5-34
5.5.3	POU Order .....	5-36
<b>5.6</b>	<b>Example</b> .....	<b>5-37</b>
5.6.1	Programming TASK and POU .....	5-37
5.6.2	Example of an Interrupt Subroutine in a DVP Series PLC .....	5-41
5.6.3	Examples of AH5x0 Series Interrupt Programs.....	5-47

---

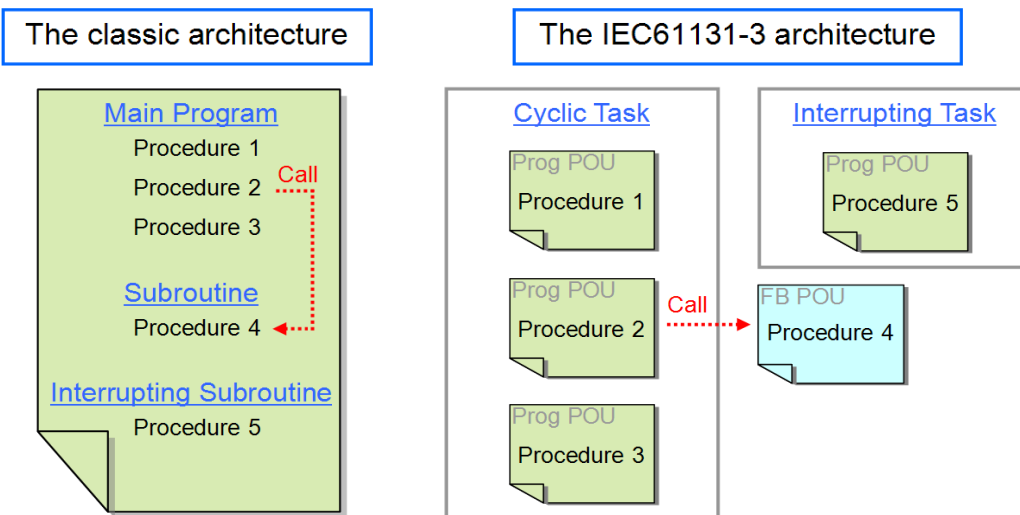
## 5.1 Program Organization Units and Tasks

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	<ol style="list-style-type: none"> <li>1. A single program is divided into several independent program organization units.</li> <li>2. A program organization unit which has been created is assigned to a cyclic task.</li> </ol>
General Subroutine	<ol style="list-style-type: none"> <li>1. A subroutine is a function block.</li> <li>2. A function block which has been created can be called by a program organization unit.</li> </ol>
Interrupt subroutine	<ol style="list-style-type: none"> <li>1. Different interrupt subroutines are independent program organization units.</li> <li>2. A program organization unit which has been created is</li> </ol>

5

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 POU in ISPSOft. They are programs (PROGs) and function blocks (FBs).

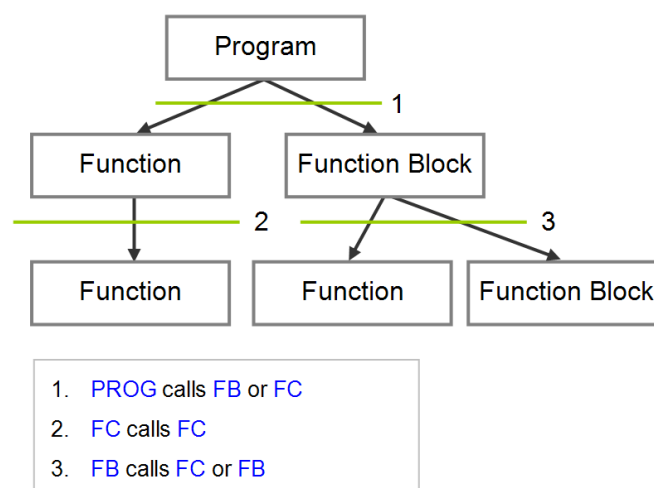
#### • Program (PROG)

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 POU is shown below. A function (FC) is a type of POU. Owing to the fact that ISPSOft currently does not support Function POU, this part will not be covered in this chapter.

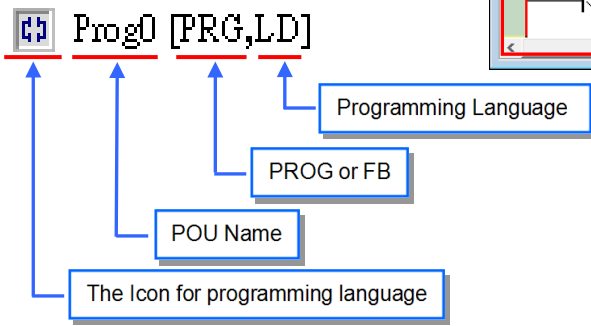
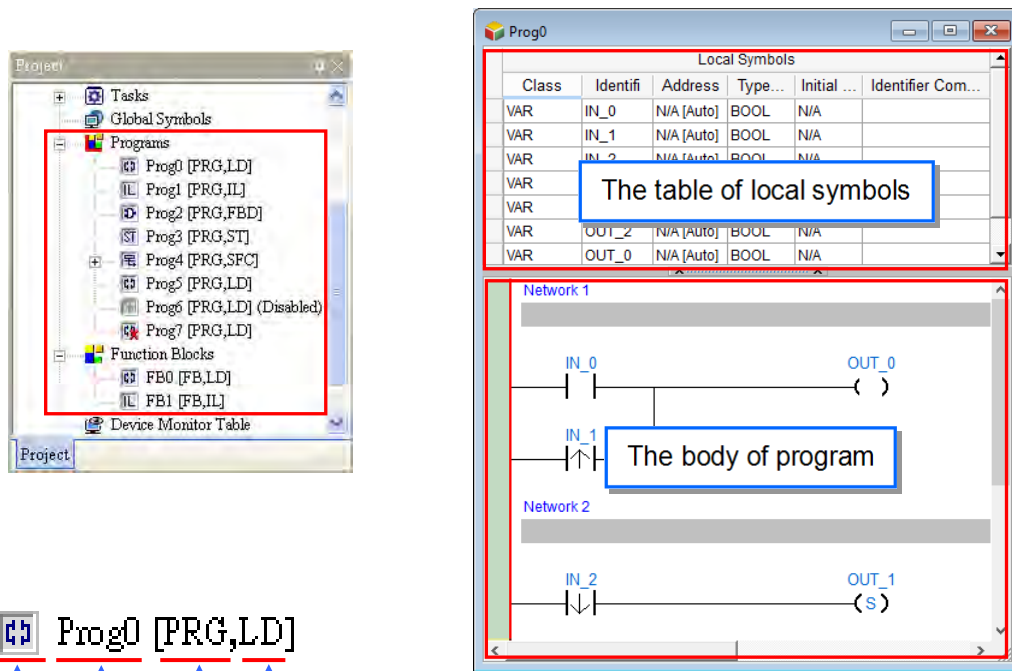


### 5.2.2 POU in ISPSOft









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.

5



The icons representing POUs are described below.

Icon	Description
	The POU is a ladder diagram (LD).
	The POU is an instruction list (IL).
	The POU is a function block diagram (FBD).
	The POU is a structured text (ST).
	The POU is a sequential function chart (SFC).
	The POU is a continuous function chart (CFC).
	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

\*. 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. The Task is like a mission. 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. When a POU is not assigned to a task, the POU is refer to as a source code saved in a project, but not to be translated into an execution code. 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 independent POUs. All POUs are rearranged to scan as execution code for compiling and the combination is based on the TASK allocation status.

In ISPSOft, a POU is assigned to one TASK, however, DVPxxMC series can assign a POU to multiple tasks, but the same task can allocate to more than one POU and the allocated POU can also assign the execute order in task; when users add a POU, a task has to be assigned by default.

Applicable	AS/AH/DVP	DVPxxMC
Work Management	A POU is assigned to one task	A POU is assigned to multiple tasks
Operation	Periodic, Timer Interrupt and Condition for Interrupt	Freewheeling ,Cyclic and Triggered by Event

For AS, AH and DVP series, the TASK operation can be categorized into three types: **Periodic**, **Timer Interrupt** and **Condition for Interrupt**.

#### (1) Periodic

For POU assigned to periodic task, the execution is simply to scan back and forth. Since there is only one periodic task in DVP series, therefore, all POUs been periodically scanned are assigned to this task; while AH series provides 32 periodic tasks for selection with number from 0 ~31. The scanning starts with small number as number one priority and AH series supports API command (TKON and TKOFF) to control task. Users can adopt the command to activate or stop a task during PLC operation.

The POU is created in SFC language and can only be allocated in periodic task and not task with interrupts.

**(2) Timer Interrupt**

For POU assigned to a task with timer interrupt, the function is similar to the subroutine of timer interrupt. When the timer interrupt has reached its time, all the POUs assigned to this task is executed according to the combination order. The task number with timer interrupt depends on the type of PLC used. In addition, the selected PLC type also determines the number of task group with timer interrupt based on the support of interrupt source groups.

**(3) Condition for Interrupt**

There are several types of condition for interrupt. For example, external interrupt, I/O interrupts, count interrupt and etc. Since different types of PLCs provides different interrupt service and amount. Thus, users have to make sure which interrupt modes are supported by the selected PLC type. When POU is allocated to task with condition for interrupt, the function is similar to subroutine interrupt. When the condition for interrupt is formed, for example the count has reached setting value, then all the POU assigned to that task will execute in order.

For DVPxxMC series, a POU can assign to multiple tasks and the same POU can repeatedly assign to the same task. While one task can assign to more than one POU and the allocated POU can also assign the execution order in the task. Three ways regarding DVPxxMC series task operation include **freewheeling**, **cyclic** and **triggered by event**. The following introduces ways of task operation and execution according to priority.

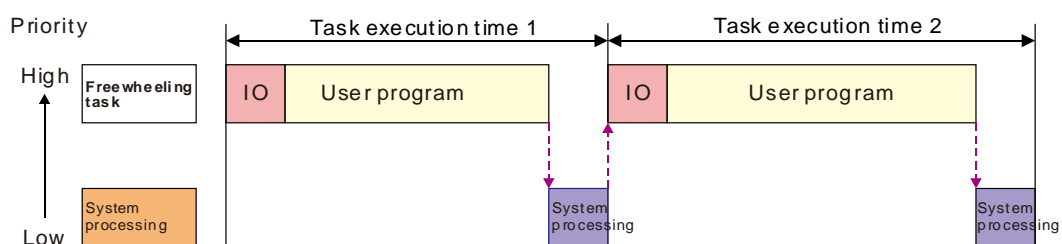
5

- **Task Operation**

**(1) Freewheeling**

The task is handled as soon as the program running starts. The task will be restarted automatically in the next cycle after one execution cycle ends.

➤ Executing freewheeling task:

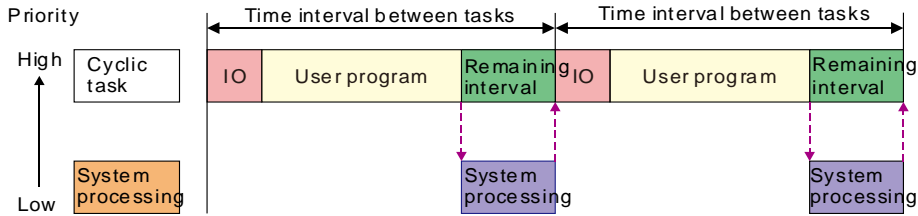


**Note:** There is no fixed execution time for the freewheeling task. So the values of task execution time 1 and task execution time 2 may not be equal in the above figure

(2) Cyclic

The cyclic task executes base on the setting of time interval in TASK Manager.

➤ Executing cyclic task:



**IO:** IO means I/O refresh. I/O includes local I/O points and left-side and right-side extension module data and CANopen data. The data can be specified to refresh before the set task is executed. If not specified, the data will be refreshed during the system processing.

**User Program:** User Program stands for user program execution which is based on the execution sequences of programs assigned in a task.

**Remaining interval:**

When the controller is to perform system processing, the low-priority task is executed first if any and then the system processing is performed.

5

**System processing:**

The controller will perform the system processing which includes Ethernet, RS232 and RS485 communication processing after all task requests are completed.

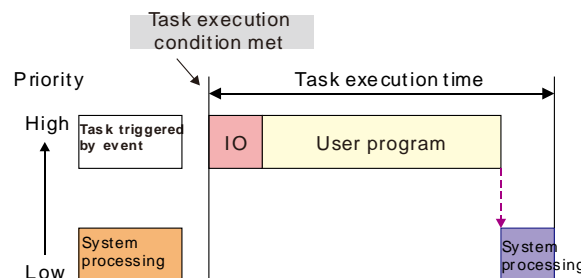
The four terms mentioned above have the same meanings as those in the following sections.

**Note:** If the cycle defined in the cyclic task is too short, the task will immediately repeat execution once finishing the user program but will not execute other low priority task or work on the system, and this may affect the execution on all tasks. If the task has setup watchdog timer for program scan, this will cause response timeout by the timer and the controller will be in Error state and stops executing the user program; if the task did not setup watchdog timer for program scan, the controller will not be able to execute the system, causing problems like communication timeout.

(3) Task triggered by event

An event task is executed once just when the specified event happens. The timing for execution of an event task depends on the timing for occurring of the event and the priority level of the event task.

➤ Executing an event task:





➤ **Events that are triggered in the following ways:**

- Motion event (Motion control task)
- Rising edge or falling edge of local input points (I0~I7 and I10~I17)
- CANopen SYNC signal
- Z pulse rising edge of incremental encoder 1 or encoder 2

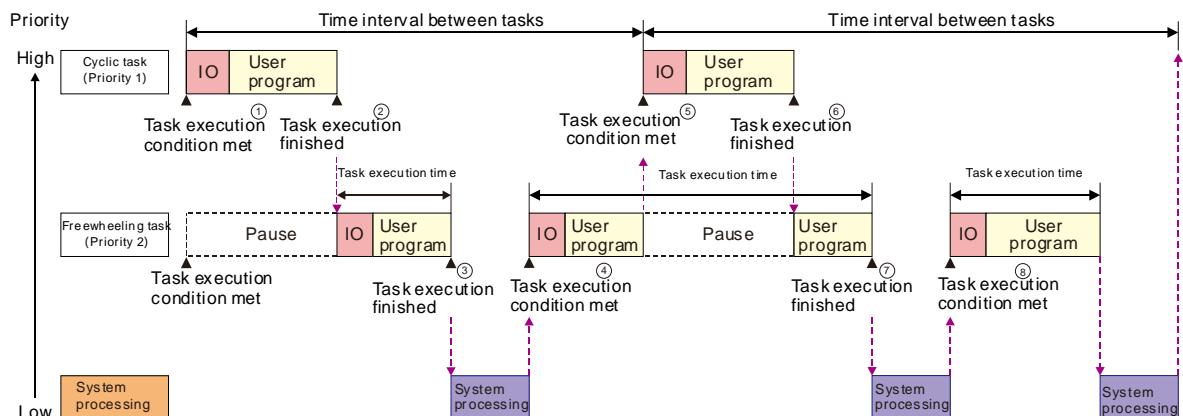
The condition for the second-time execution is ignored when the condition required for execution of the event task is met again before the event task is completed. The period before an event task is completed is the course while the event task is being executed or is waiting to be executed.

● **Executing Task According to Priority**

The controller can not perform multiple tasks simultaneously. Every task must be given a priority level and they are executed according to preset priorities. Priority level can be set within the range of 1 to 24. (1 is the highest priority and 24 is the lowest priority.) The priority level of each task must be unique. The task with higher priority takes priority to perform. The high-priority task can interrupt the low-priority task.

We recommend that the task which has a high requirement of real time should be given a high priority and the task which has a low requirement of real time should be given a low priority. The priority of the default motion control task built in ISPSOft is 1 by default.

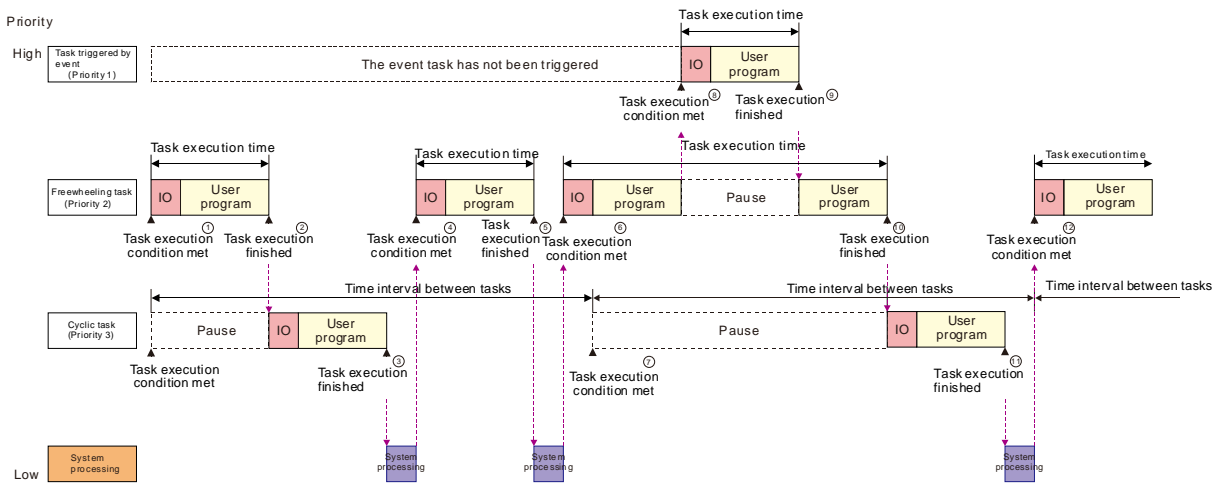
➤ **Executing two tasks at the same time (Cyclic and freewheeling)**



- ① The execution conditions for the cyclic task and freewheeling task are met at the same time. The cyclic task is executed first because of its higher priority.
- ② When the cyclic task execution is finished, the freewheeling task execution starts.
- ③ The controller will execute the system processing if there is no other task after the execution of the freewheeling task is completed.
- ④ The execution of the freewheeling task continues since the high-priority cyclic task request has not arrived.

- ⑤ The cyclic task interrupts the freewheeling task execution and the controller executes the cyclic task because of the arrival of the high-priority cyclic task request during the execution of the freewheeling task.
- ⑥ The controller continues to execute the part of the low-priority freewheeling task, which has not been executed yet when the execution of the cyclic task is completed.
- ⑦ When the execution of the freewheeling task is completed, the controller executes the system processing due to no other task request.
- ⑧ When the system processing is completed, the execution of the freewheeling task continues due to no high-priority cyclic task request.

➤ **Executing mixture of three tasks (Event, Cyclic and Freewheeling task)**



5

- ① When the conditions for execution of the freewheeling task and cyclic task are both met, the freewheeling task is executed first because the priority of the freewheeling task is higher.
- ② The cyclic task execution starts when the freewheeling task execution is completed.
- ③ When the cyclic task execution is completed, the controller executes the system processing due to no other task request.
- ④ The freewheeling task is executed when the system processing is completed.
- ⑤ When the freewheeling task execution is completed, the controller executes the system processing due to no other task request.
- ⑥ The freewheeling task is executed when the system processing is completed.

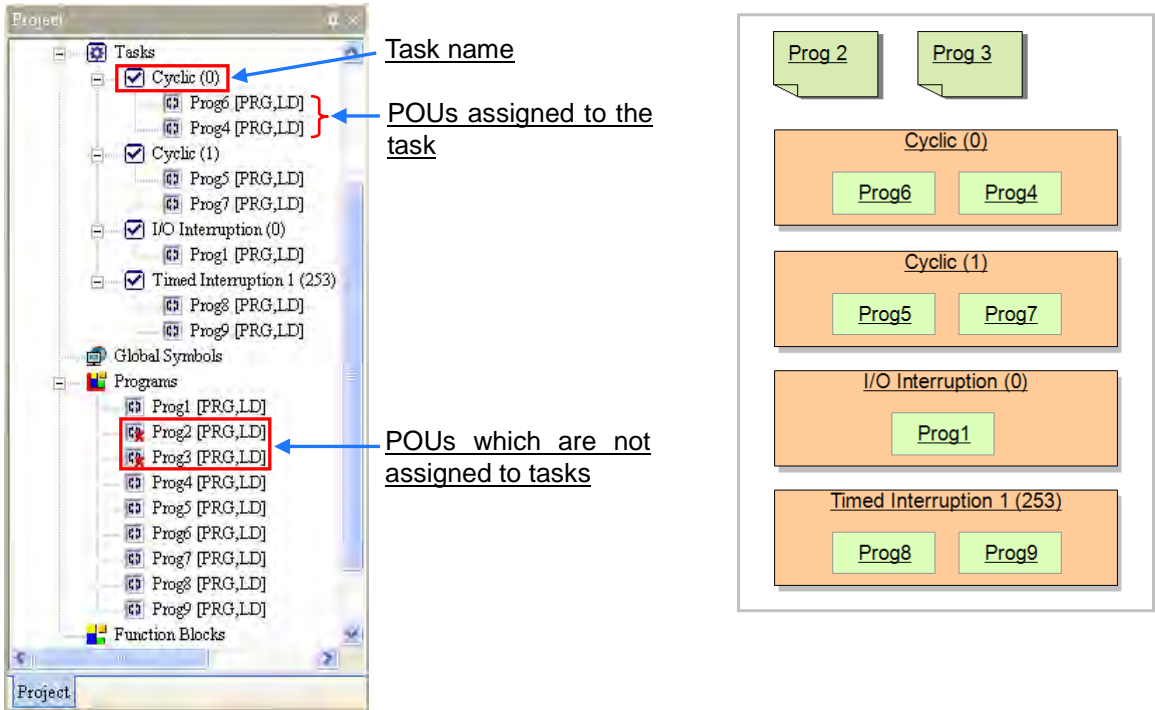
- ⑦ The freewheeling task execution continues because the freewheeling task has a higher priority than the cyclic task although the execution condition for the cyclic task is met. And the cyclic task waits to execute.
- ⑧ The event task interrupts the freewheeling task execution because the event task has the highest priority and the execution condition for the event task is met.
- ⑨ The controller continues to execute the part of the low-priority freewheeling task, which has not been executed yet when the event task execution is completed.
- ⑩ The freewheeling task execution is completed. The controller executes the cyclic task since the cyclic task request in ⑦ is not responded yet.
- ⑪ The cyclic task execution is completed. The controller executes the system processing due to no other task request.

### 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	Owing 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.

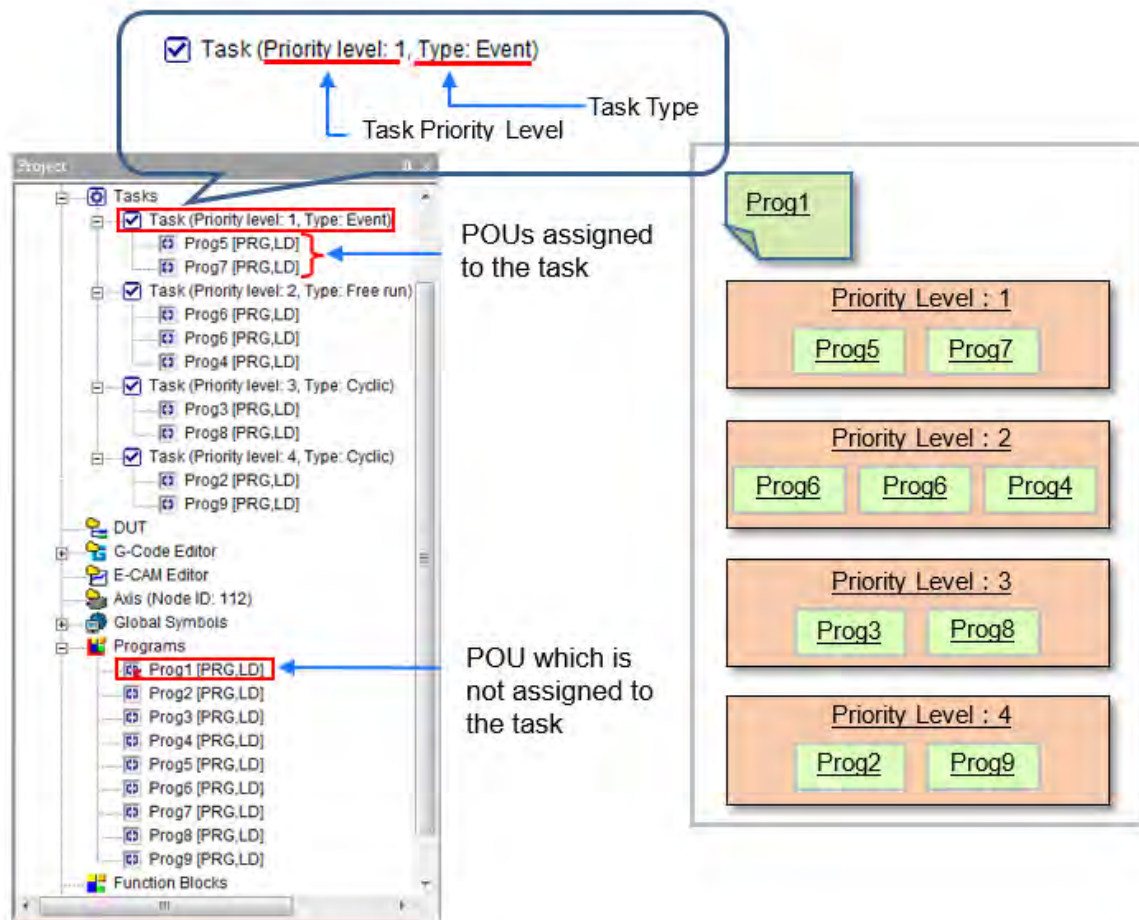


5

For DVPxxMC series, the task name appears to be a bit different with priority and type added. The following projects has established 9 POUs in total and the task allocation are described below.

The priority order and task type determine the task execution order.

TASK Priority Level	TASK Type	POU	Execution
Not allocated		Prog1	The POU is not executed, because the Task is not assigned.
Priority: 1	Triggered by Event	Prog5, Prog7	When executing, the program scans from Prog5 → Prog7 (back and forth)
Priority: 2	Freewheeling	Prog6, Prog6, Prog4	When executing, the program scans from Prog6 → Prog6 → Prog4 (back and forth)
Priority: 3	Cyclic	Prog3, Prog8	When executing, the program scans from Prog3 → Prog8 (back and forth)
Priority: 4	Cyclic	Prog2, Prog9	When executing, the program scans from Prog2 → Prog9 (back and forth)



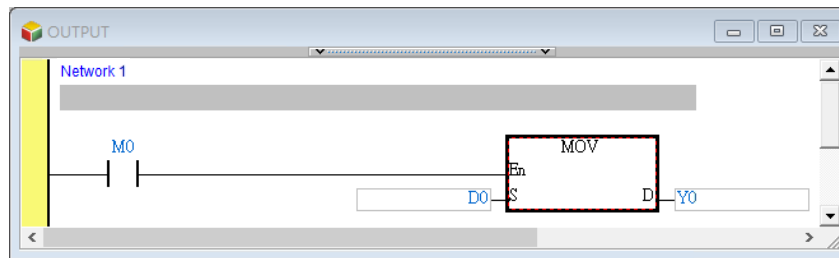
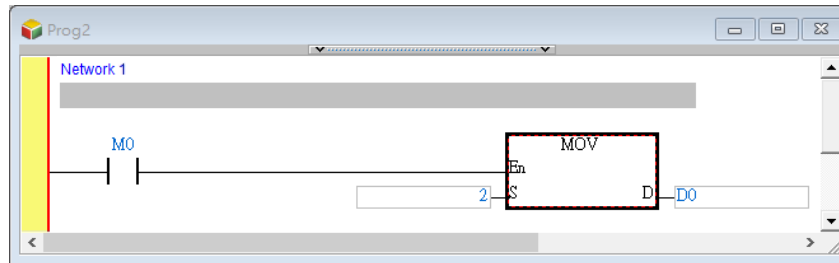
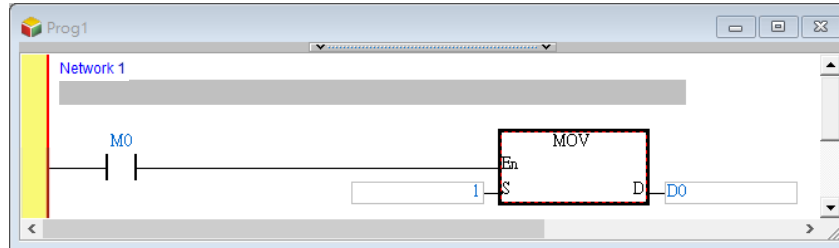
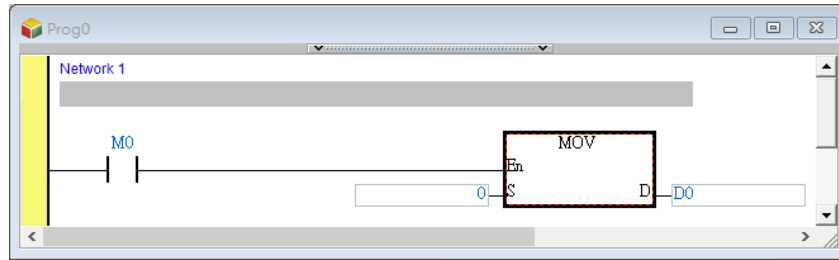
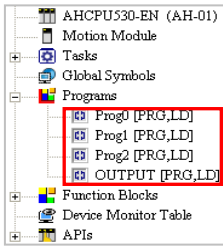
5

### 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 POU are as follows.

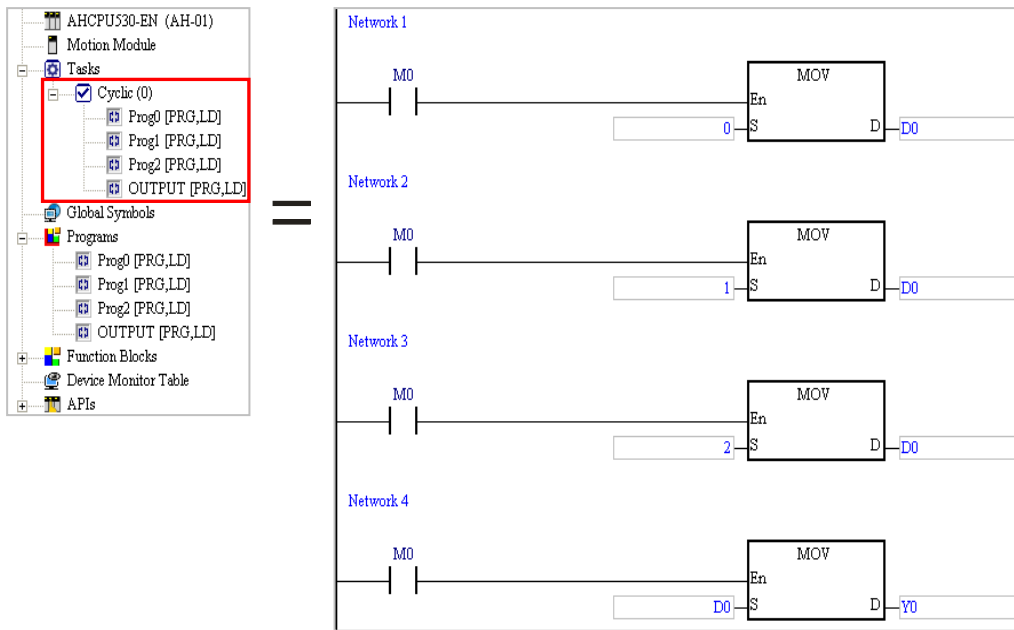


5

Suppose the four POU of the program type are assigned to cyclic task (0). The arrangement of the POU is shown below. The order in which these POU are executed is

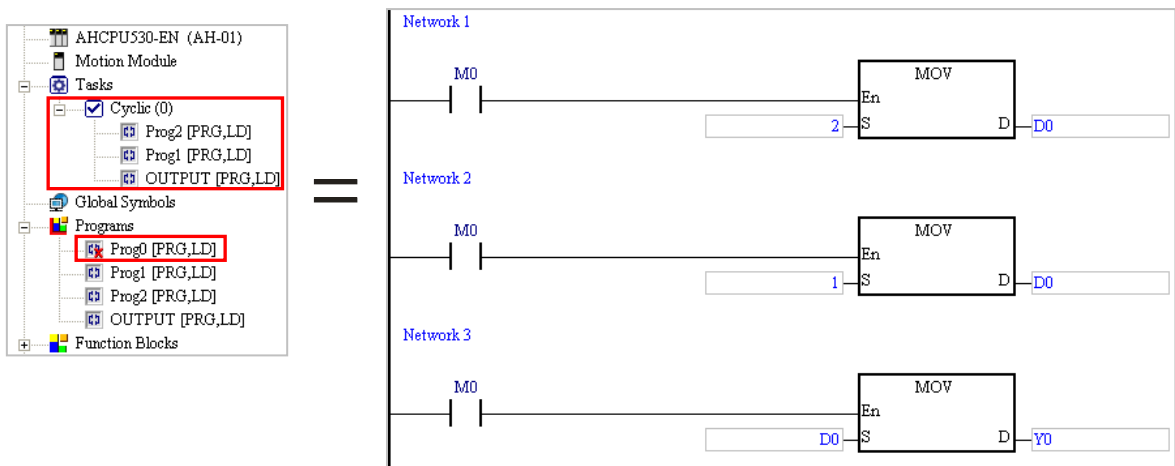
Prog0→Prog1→Prog2→OUTPUT. The value in D0 in OUTPUT depends on the program in Prog2.

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 POU 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

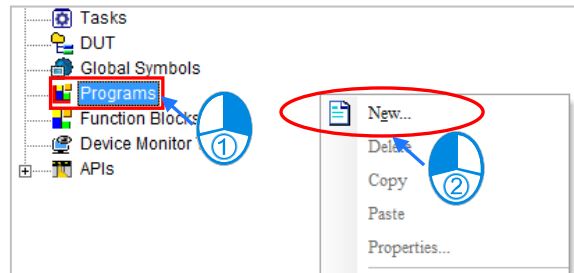


## 5.4 Managing a POU

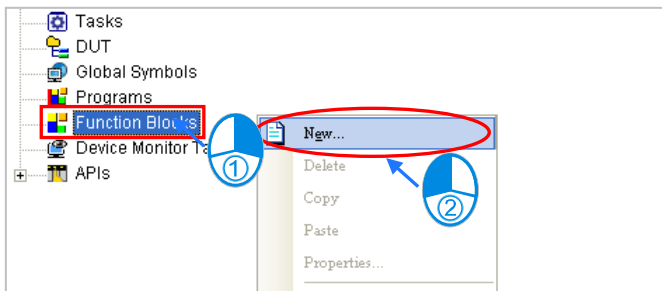
### 5.4.1 Creating and Setting a POU

- (1) If users want to create a POU of the program type, they have to right-click **Programs** in the project management area 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 and click **New...**

- Adding a POU of the program type



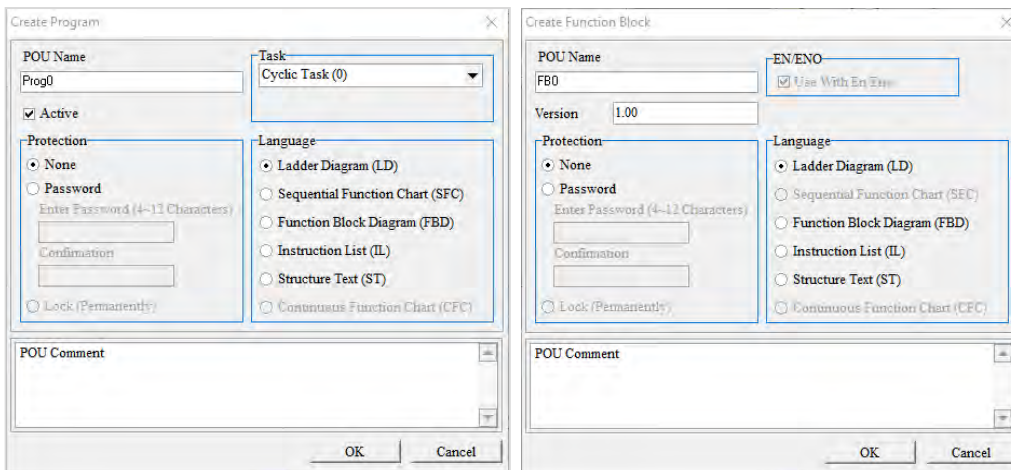
- Adding a POU of the function block type



- (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. Users can choose the task types from the drop-down menu base on the selected PLC series.

- Program (PROG)

- Function Block (FB)

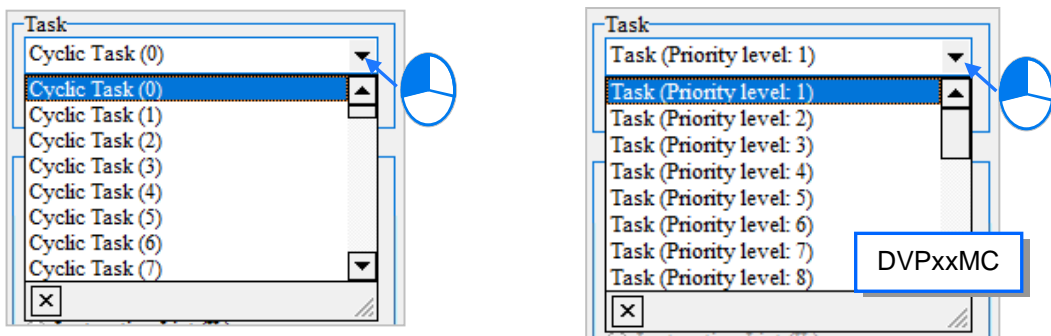




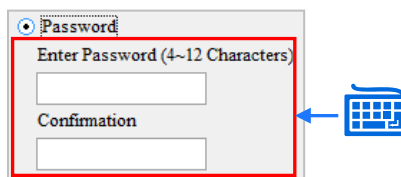
- **POU Name:** Users can type a POU name in this box.
  - (a) Special marks and spaces can not be used. For example, \*, #, ?, \, %, @, and etc. can not be used.
  - (b) A POU name is composed of thirty characters at most. Users have to notice that a Chinese character occupies two characters.
  - (c) 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.



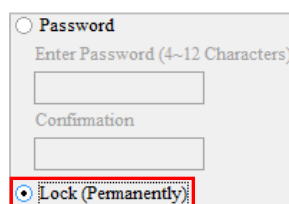
- **Task:** POU is created and can assign a task type. For DVPxxMC series, the content appears to be different (see below).



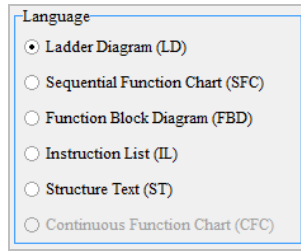
- **Password:** If users want to set a password, they have to type the same passwords in the two boxes. English letters, numerical digits, and special marks can be typed in the boxes.



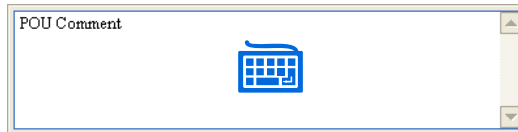
- **Lock (Permanently):** This function can lock the function block permanently. A copy of the project is required if Lock (Permanently) is selected. Users can right-click Attributes on the POU item after adding a function block (FB POU).



- **Language:** Users have to select a programming language in which the POU is written. (The language may vary according to the selected PLC series)



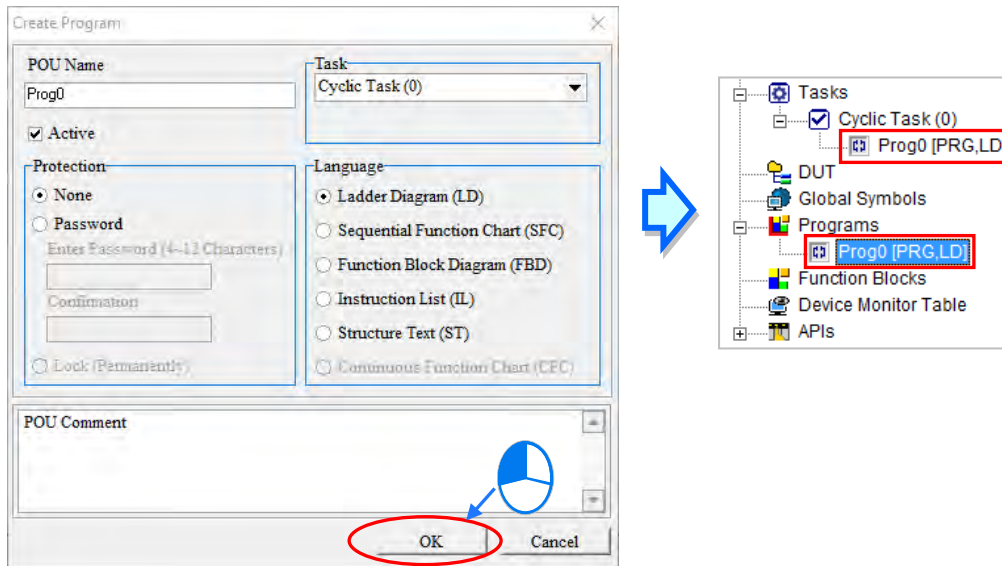
- **Comment:** Users can type a comment on the POU in this box.



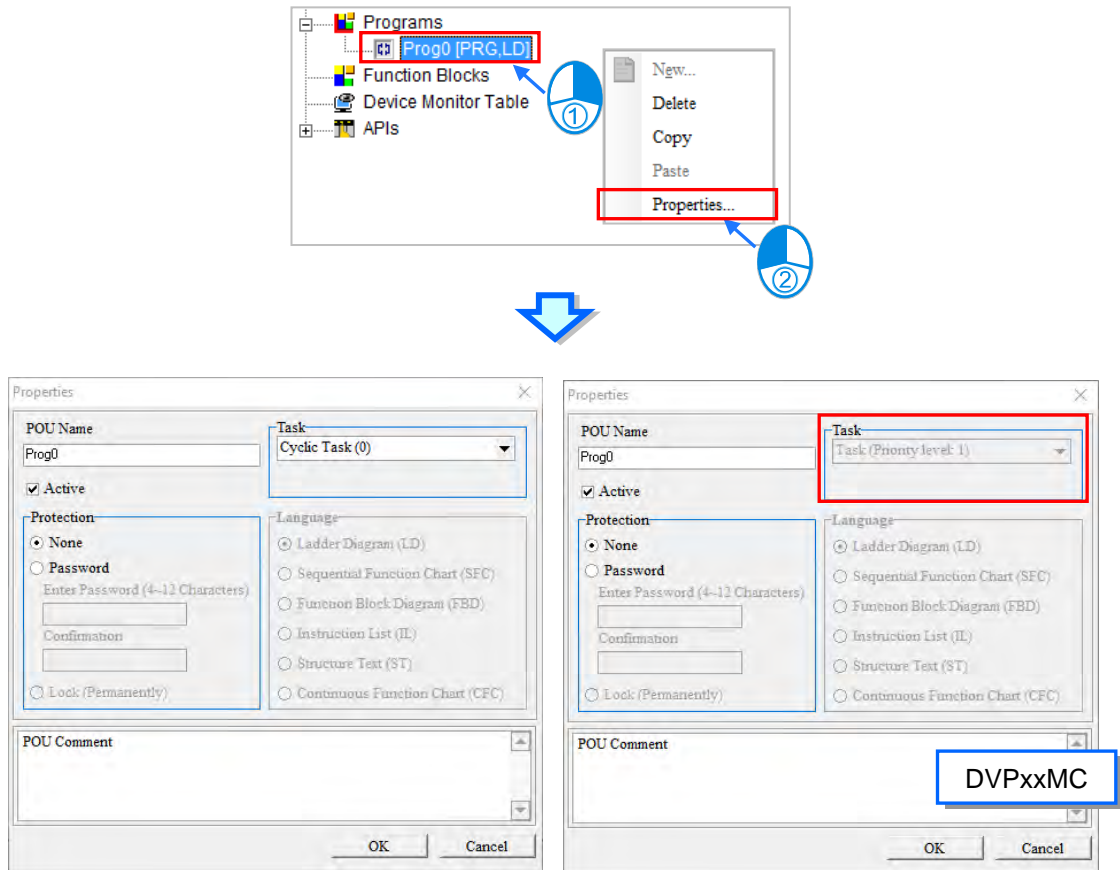
\*. If the POU created is a POU of the program type, the Active checkbox will be in the window. The function of the Active checkbox will be introduced in section 5.4.2.

- (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.

5



- (4) If users want to modify the properties of a created POU, right-click **Properties** from the POU item and refer to previous description for setup. The programming language cannot be changed when using edit on Properties. For DVPxxMC series, editing Properties cannot change the task setting.



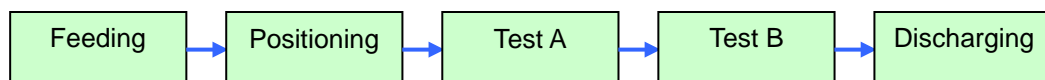
5

### 5.4.2 POU Active State

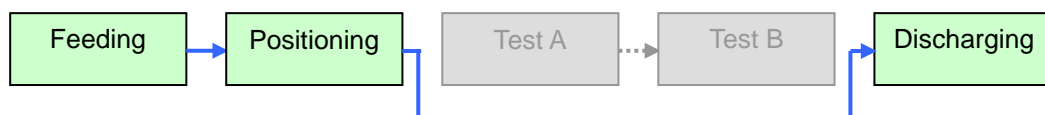
ISPSOft provides the function to temporarily close POU. When POU is closed, the POU will be skipped for compiling and the internal process will also not be executed. However, this function only supports in program type POU.

This function is used for testing or debugging. To skip some process beforehand, users can set POU as ineffective and restore to active testing is complete.

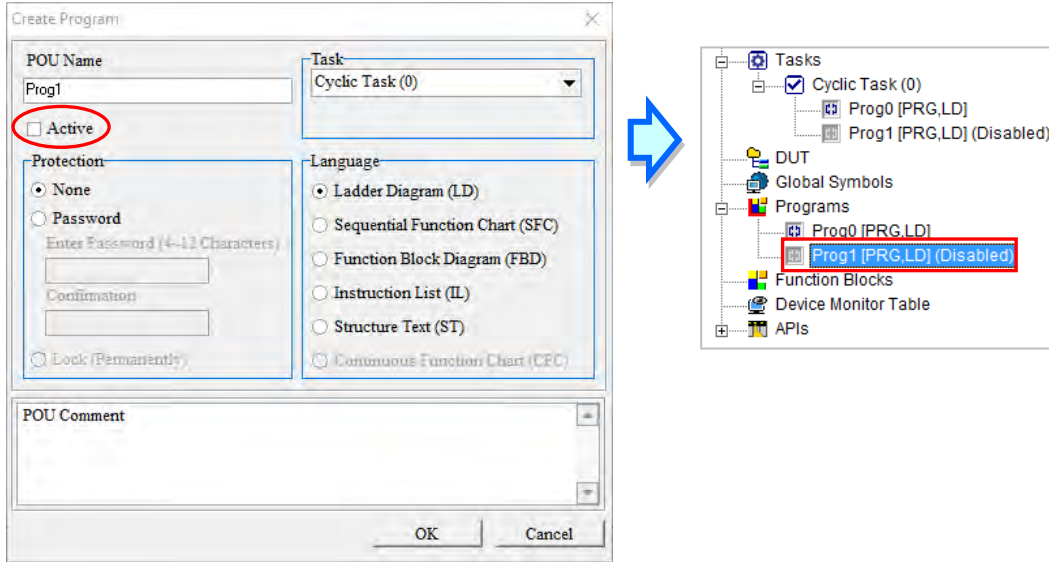
The following example shows the equipment has five workstations and the task process is categorized into 5 POU for development.



When equipment assembly is complete, but test instrument are not, users can disable both POU- test A and test B. Then, users can skip the two station to focus on other parts for adjusting and testing.



To set POU in active state, click **Active** in the Create Program window. If Active is not selected, the POU function is closed. A different icon is shown in the Project section to represent POU function closed.

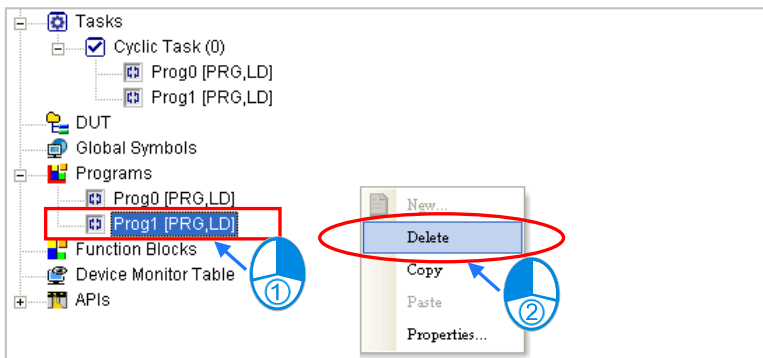


5

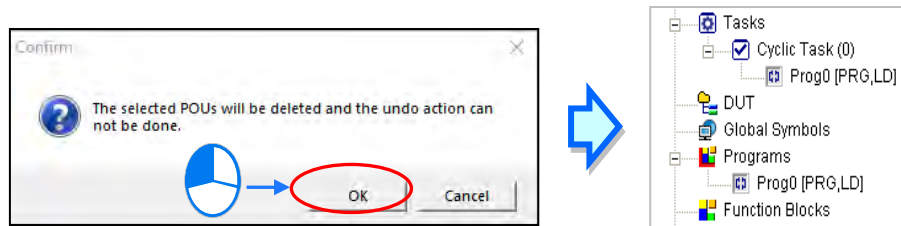
- \*1. Users can disable a POU by removing it from the original task section. However, if the users want to execute the POU later, they have to assign the POU to the original task again, and rearrange the POUs in the task section. As a result, it is more convenient for the users to disable the POU temporarily by unselecting the Active box in the window.
- \*2. A POU can be enabled not only by selecting the Active box, but also from Task Manager. Please refer to section 5.5.3 for more information.

### 5.4.3 Delete and Copy POU

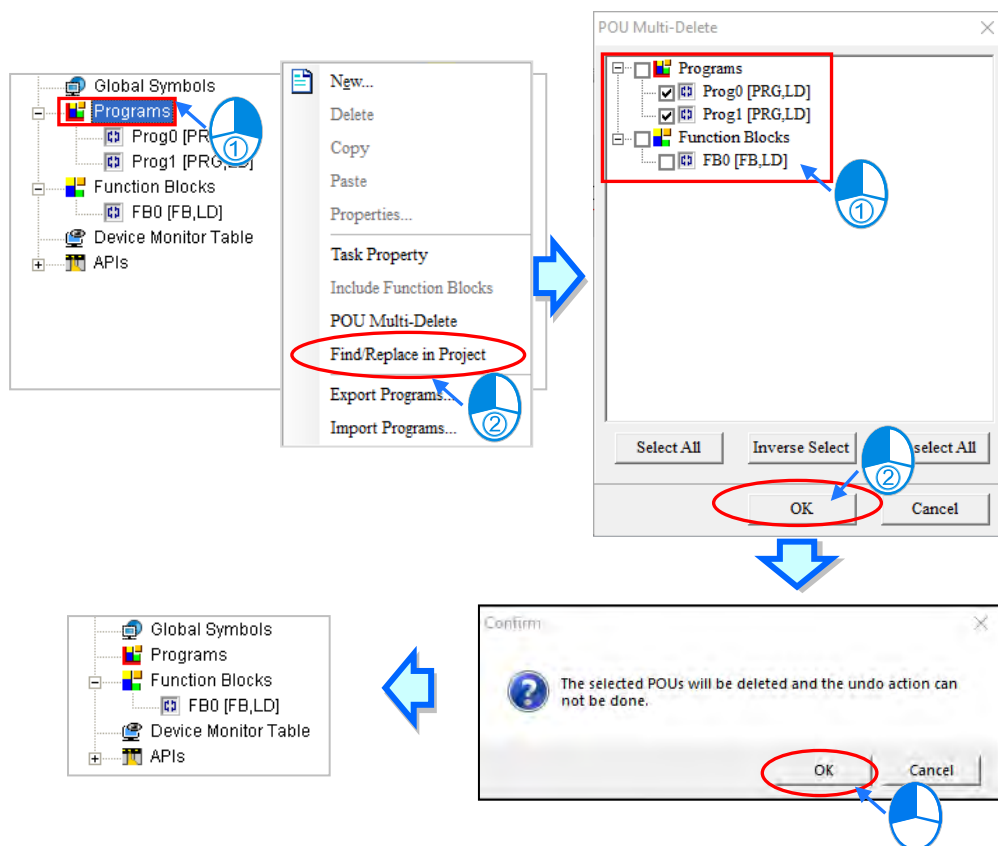
To delete a POU, right click on the selected item and click **Delete**.



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.

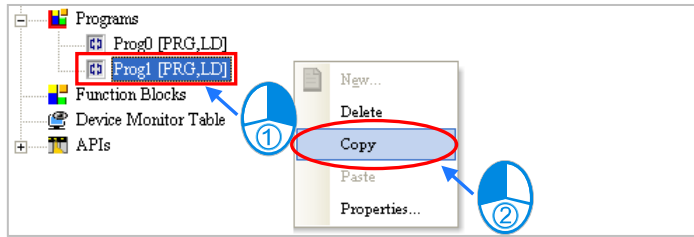


POU Multi-deletions are also supported. Right-click "Programs" or "Function Blocks" to see and select the option "POU Multi-Delete". And then users will see the POU Multi\_Delete window, use the "Select All", "Inverse Select" and "Deselect All" buttons to select and delete more than 1 POU at the same time.

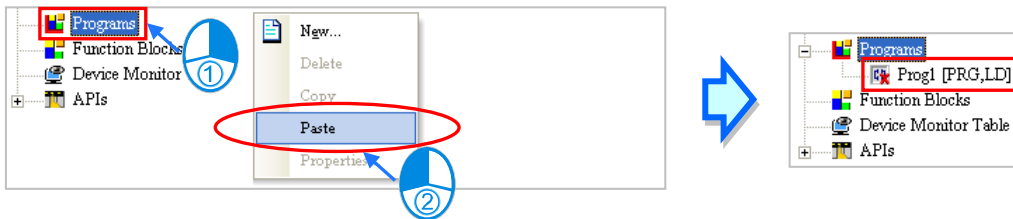


5

To copy a POU, right-click on the selected item 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** 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.

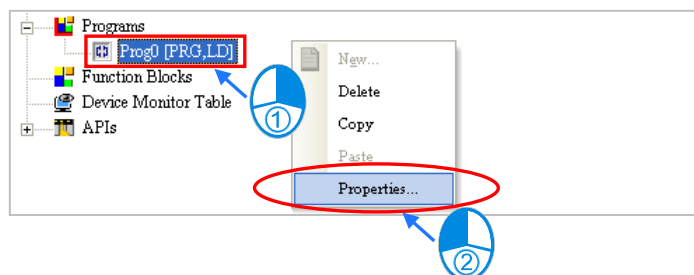


\*. If a POU is pasted on a position where the same POU name exists, the system will put CopyOf\_n at the end of the POU which is pasted. (n in CopyOf\_n is a serial number.)

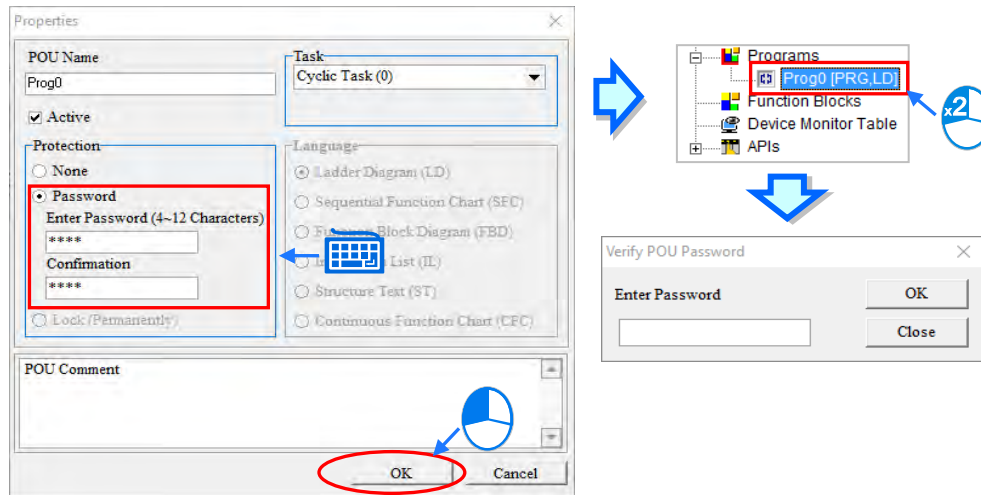
#### 5.4.4 Set and Remove POU Password

Users can set a password for a POU in ISPSOft. The setting of a password, and the unlocking of the password are described below.

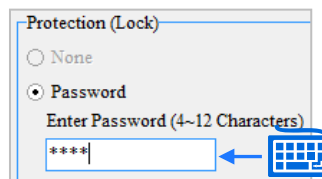
- (1) When users create a POU, they can set a password for the POU. If users do not set a password a POU when the POU is created, but want to protect the POU with a password later, they have to right-click the POU in the project management area and click **Properties...** to open the **Properties** window.



- (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**.

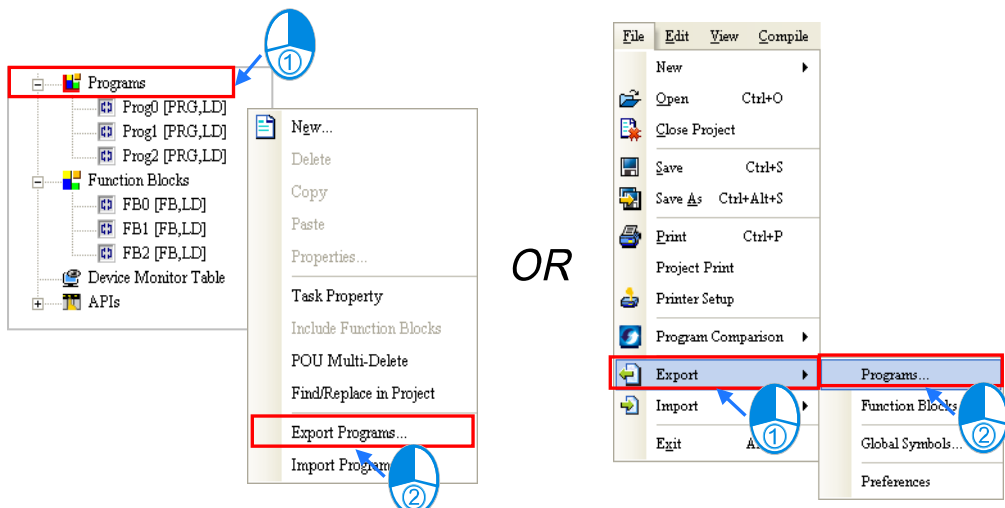


5

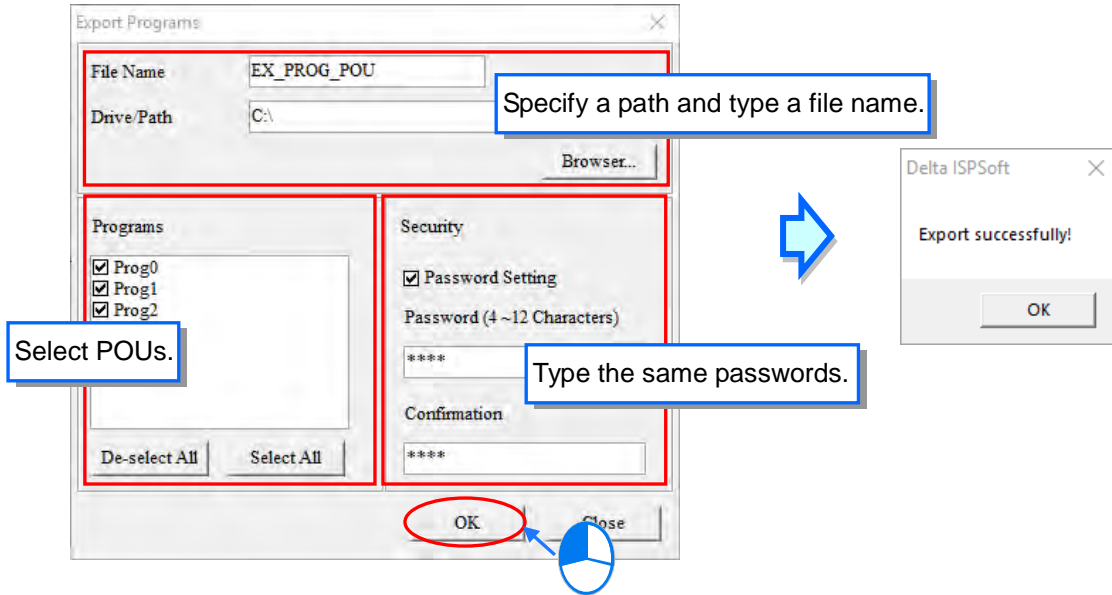
### 5.4.5 Exporting POU Program

ISPSOft supports exporting and importing POU. But, under group project structure, users can only operate on the current editing project.

Choose **Programs** from **Export** in the **File** menu to directly operate on the current editing project for export; or, right-click **Programs** to select **Export Programs** from the quick menu.



In the **Export Programs** window, specify a path in the **Drive/Path** box, type a file name in the **File Name** box, and select POU's 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**.

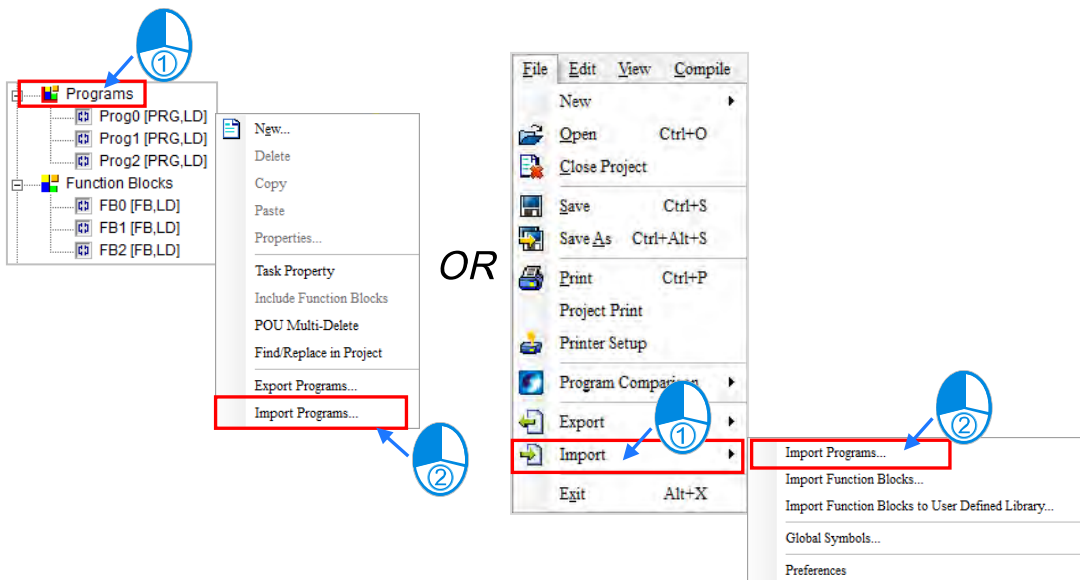


5

### 5.4.6 Importing POU Program

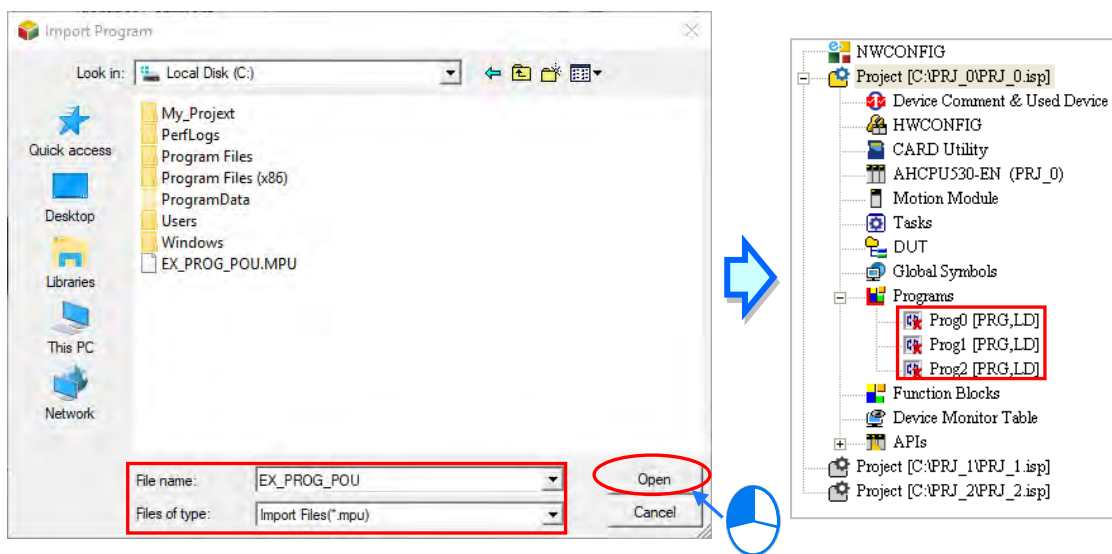
Similar to **Exporting**, users can only operate on the current editing project for **Import**. When importing, exported function block POU cannot import in **Programs**.

Choose **Import Programs** from **Import** in the **File** menu to directly operate on the current editing project for import; or, right-click **Programs** to select **Import Programs** from the quick menu.





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 POU 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 POU will be imported normally.

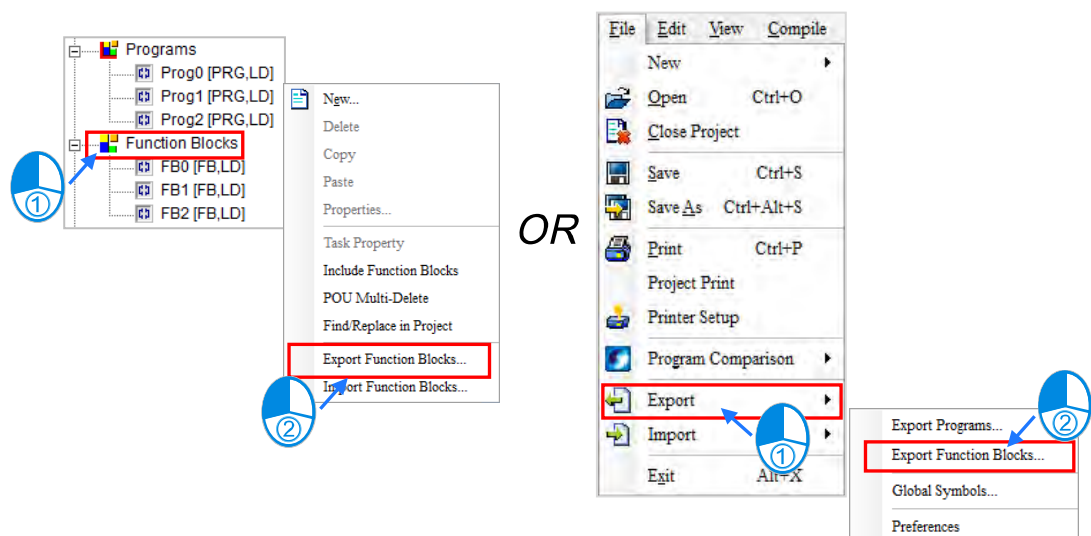


5

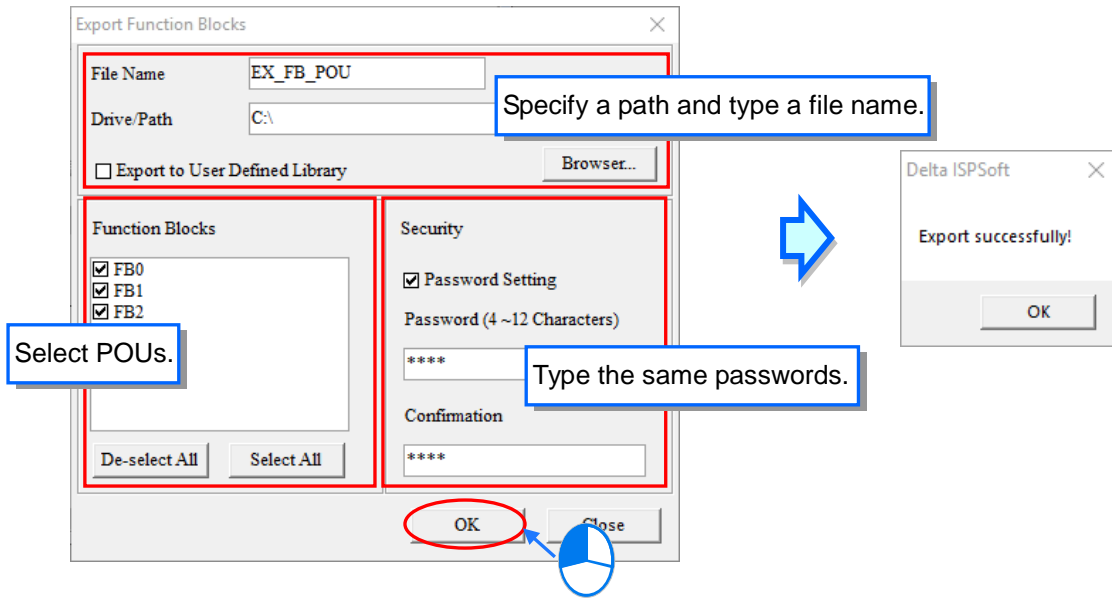
### 5.4.7 Exporting Function Block POU

ISPSOft supports exporting and importing POU. But, under group project structure, users can only operate on the current editing project.

Choose **Export Function Blocks** from **Export** in the **File** menu to directly operate on the current editing project for export; or, click **Function Blocks** to select **Export Function Blocks** from the quick menu.



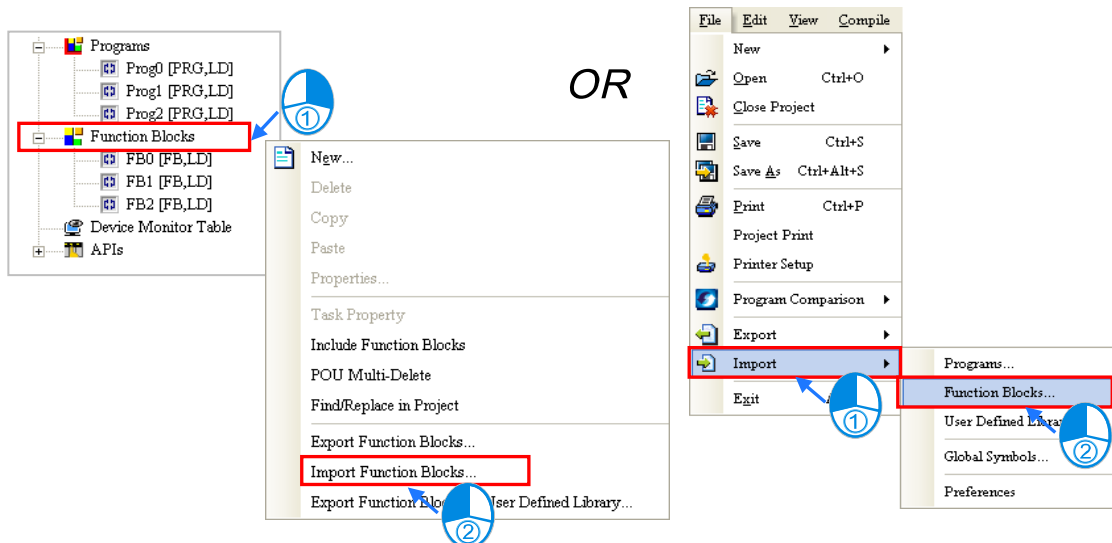
In the **Export Function blocks** window, specify a path in the **Drive/Path** box, type a file name in the **File Name** box, and select POU's of the function block type which will be exported in the **Function Blocks** section. If the users want to protect the file which will be exported with a password, 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**.



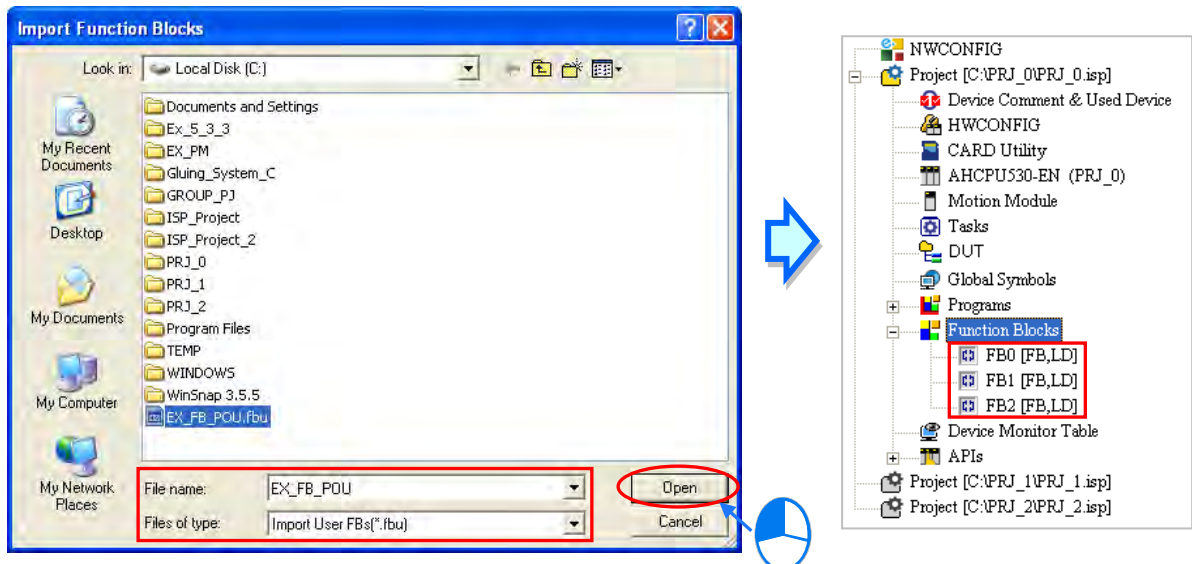
### 5.4.8 Importing Function Block POU

Similar to **Exporting**, users can only operate on the current editing project for **Import**. When importing, exported POU cannot import back under **Function Blocks**.

Choose **Function Blocks** from **Import** in the **File** menu to directly operate on the current editing project for import; or, click **Function Blocks** to select **Import Function Blocks** from the quick menu.

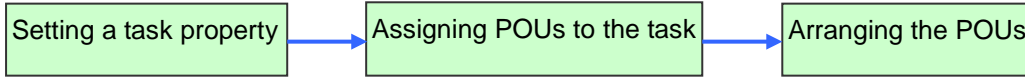


Select a file which will be imported, and then click **Open**. If the file is protected with a password, the users have to type the password in the **Password for Decrypting** window, and click **OK**. Besides, if the name of a POU which will be imported is the same as the name of a POU in the project, the system will not allow the POU to be imported, but the other POUs will be imported normally.



## 5.5 Task Management

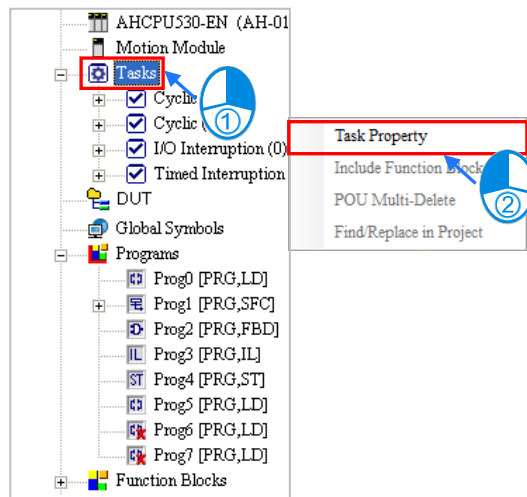
The steps of managing the tasks in ISPSOft are shown below. Please refer to the following sections for more information about the management of the tasks in ISPSOft.



Before managing the tasks in ISPSOft, users have to start the Task Manager. There are three ways to start the tasks management tool.

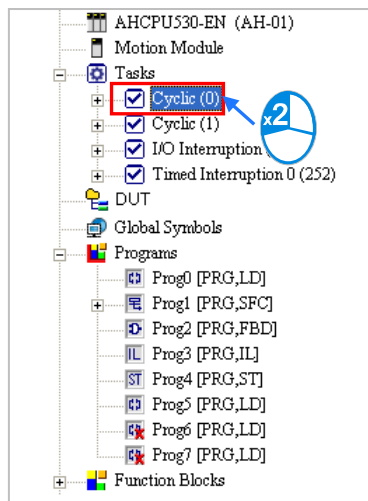
### ● Method 1

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

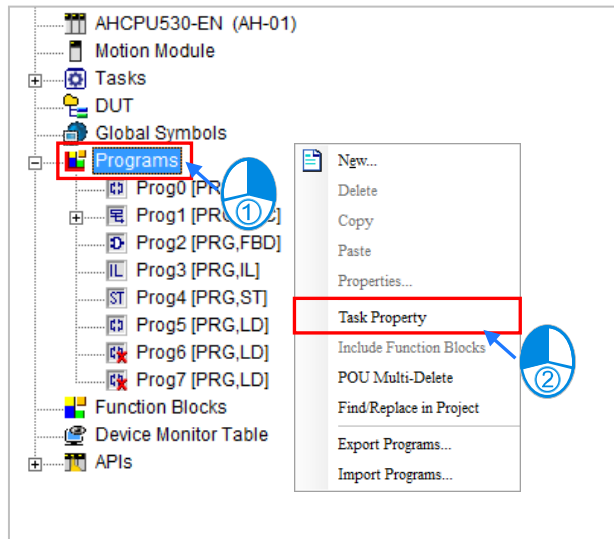
Unfold the **Tasks** section in the project management area, and then double-click a task which will be set to open the **Task Manager** window.



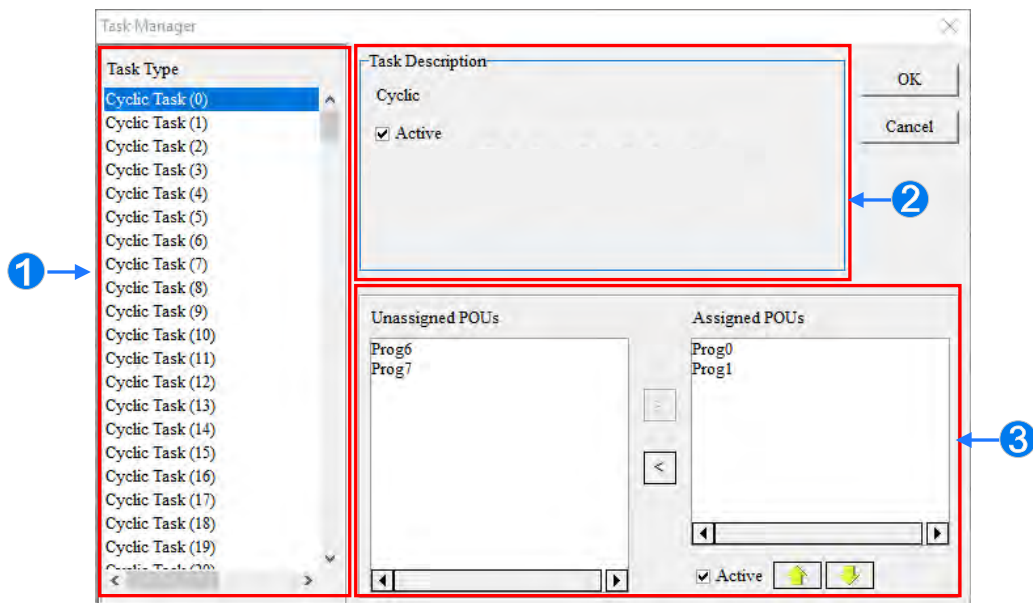
5

### ● Method 3

Right-click the **Programs** section in the project management area, and then click **Task Property** on the context menu to open the **Task Manager** window.



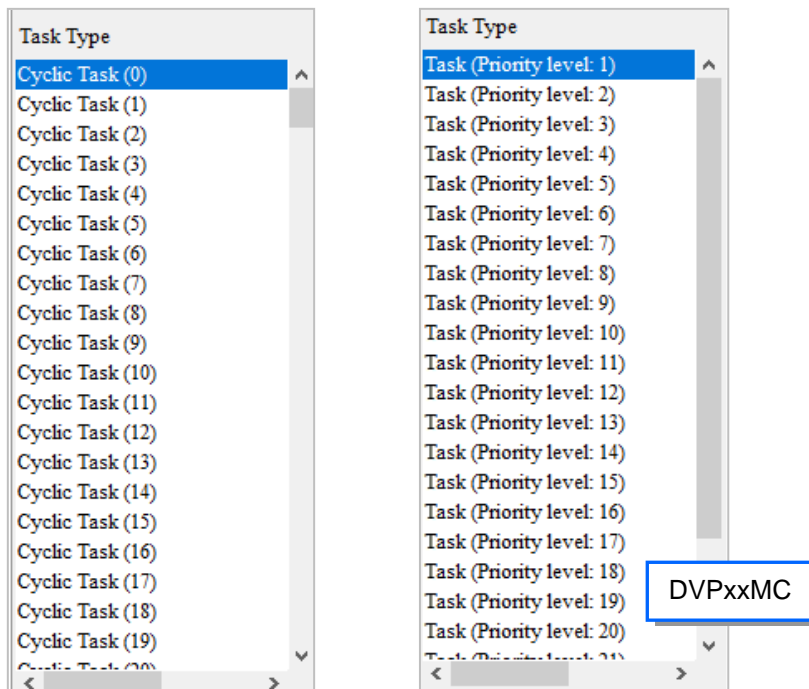
Below is the **Task Manager** setting page.



- 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.

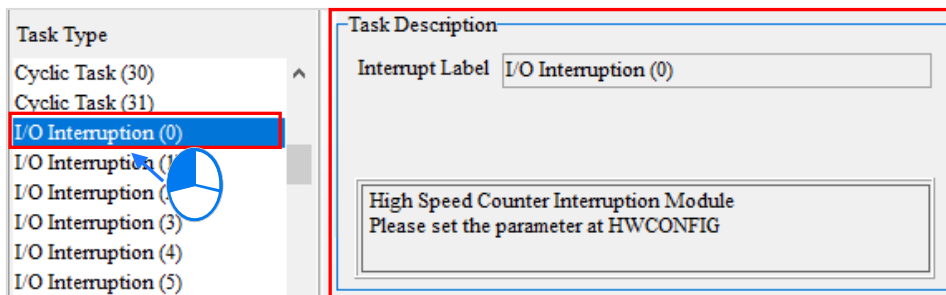
### 5.5.1 Setting Task Description and Condition for Interruption

The following TASK listed in **Task Type** differs according to the different types of PLC.

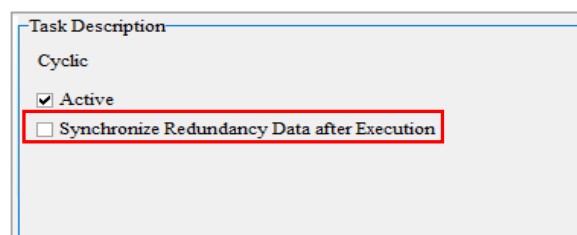


5

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. The tasks provided by projects are related to the functions of selected PLCs. Thus, each task property content is different, but basic settings are the same.



The content in Task Description also differs depending on the types of PLC. For instance, when using AH560 redundant system series, specific option like **Synchronize Redundancy Data after Execution** appears. For more information regarding the function, please refer to related product user manual.

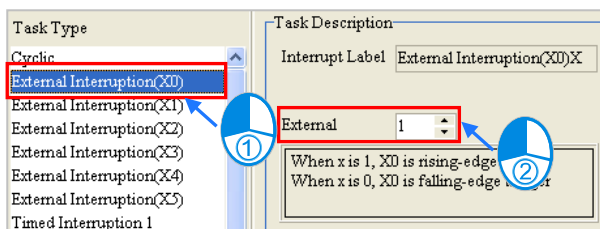


The setting of the properties of tasks can be illustrated with some simple examples. Users can refer to the examples and the operation manual for a PLC for more information about setting the properties of tasks.

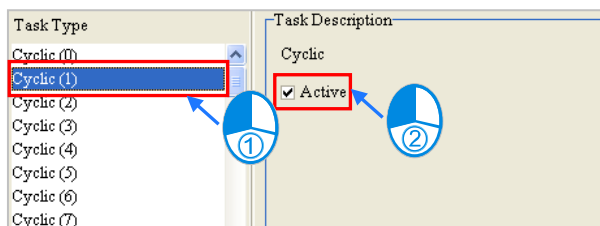
Example 1: To set the interrupt task property of SV series, please check on the interrupt indicators provided by SV series. The following table demonstrates interrupt indicators in SV series from the DVP-PLC Application Manual (Programming).

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

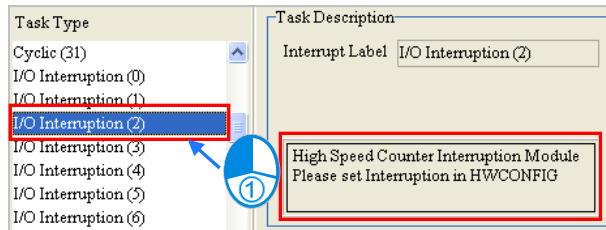
The **Task Type** section for a DVP-SV series PLC is shown below. If users want to set a task corresponding to the interrupt I001, they have to select **External Interruption (X0)** in the **Task Type** section first. (I001 is triggered when the signal passing through X0 goes from low to high.) The description 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.



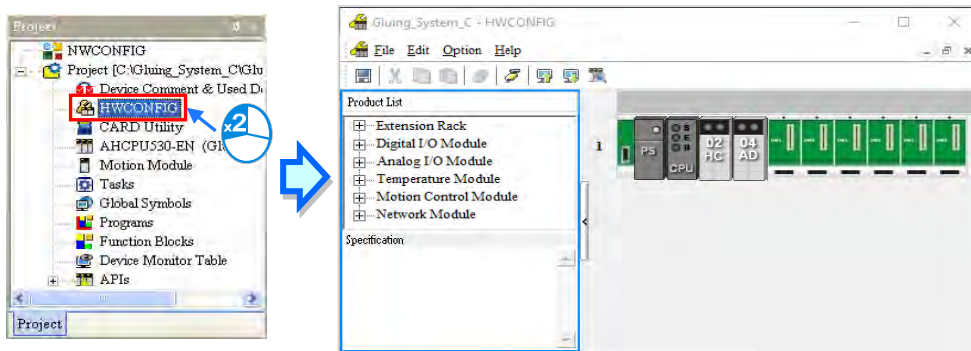
Example 3: The figure below is the **Task Manager** window for AHCPU530-EN. The task selected is **I/O Interruption (2)**. Owing to the fact that users can set the condition of an I/O interrupt by themselves, the message in the **Task Description** section indicates that the condition of **I/O Interruption (2)** can be set in HWCONFIG. The setting of the condition of **I/O Interruption (2)** in HWCONFIG is described below.



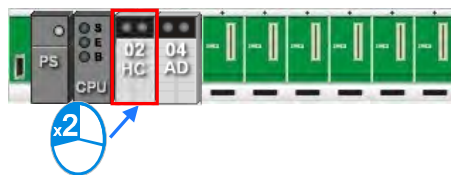
Suppose **I/O Interruption (2)** will be triggered after channel 1 on AH02HC-5A receives a certain number of pulses. The setting of **I/O Interruption (2)** in HWCONFIG is described below. Users can refer to chapter 3 for more information about setting the parameters in a module, and refer to AH500 Operation Manual for more information about the interrupts in an AH500 series CPU module.

- (1) Double-click **HWCONFIG** in ISPSOft.

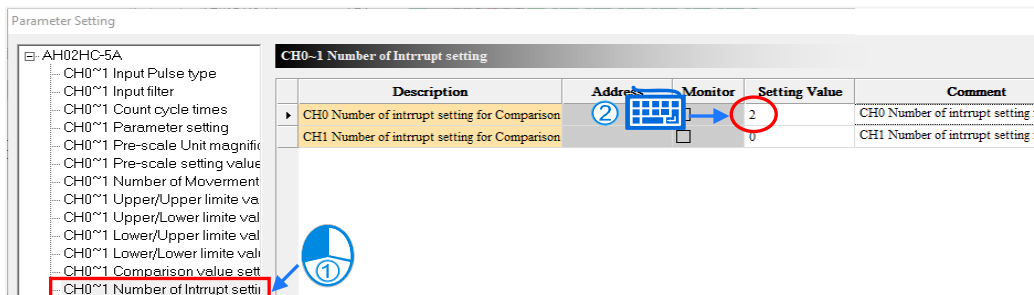
5



- (2) Double-click AH02HC-5A to open the **Parameter Setting** window.



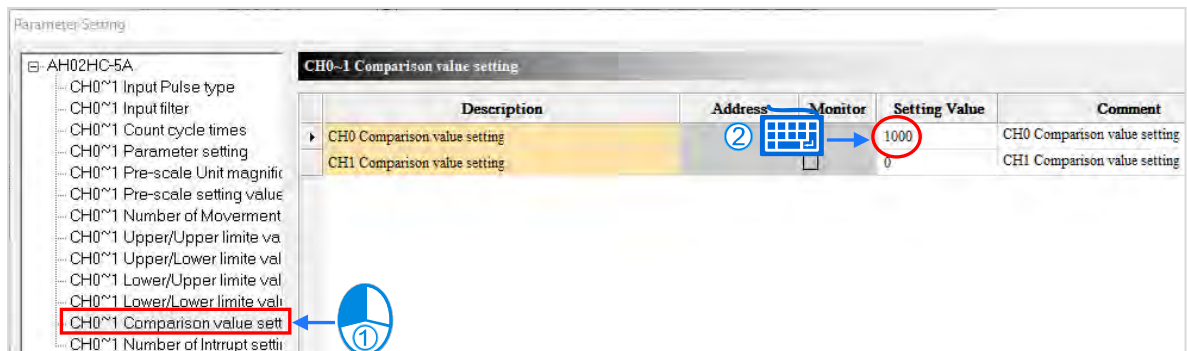
- (3) Select **CH1~2 Number of Interrupt Setting**, and type 2 in the **Initial** cell for **CH1 Number of Interrupt Setting for Comparison**.



\*. The number of interrupt setting for channel 0 and 1 can be the same. This setting means when the condition of any channel is set, it will trigger the same I/O interrupt.



- (4) Switch to comparison value setting page and input the Setting Value for triggering interrupt in **CH0 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.



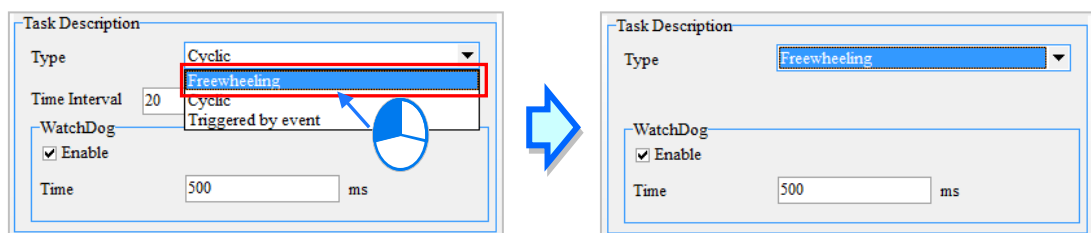
When using DVPxxMC series, the work list has options containing priority level 1~24 (1 refer to the highest priority level, 24 is the lowest priority level). After choosing the priority level, users can select the appropriate **Type** and setup **Watchdog** for program scan from Task Description. The following section introduces **Task Type** and **Watchdog** function.

5

● **Type**

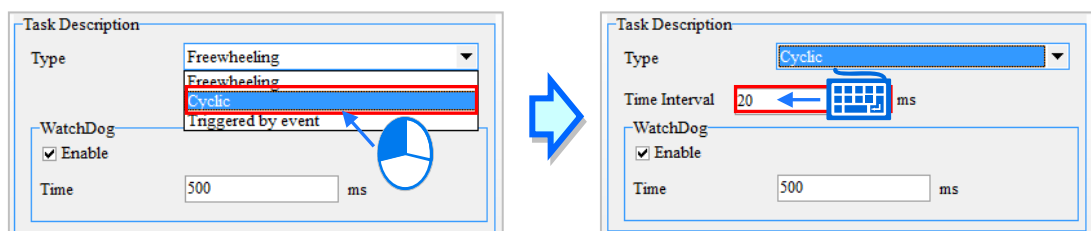
The three task types are **freewheeling**, **cyclic** and **triggered by event**.

- (1) Freewheeling



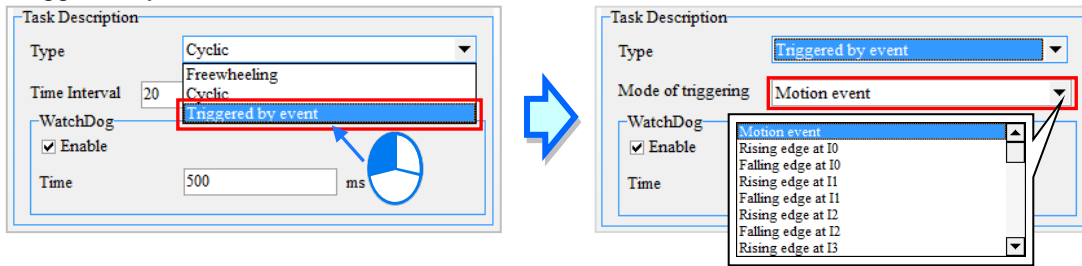
Choose **Freewheeling** from the **Type** drop-down list. This is the first task to choose when start operating. After operating for a cycle, the task is activated in the next cycle.

- (2) Cyclic



Choose **Cyclic** from the **Type** drop-down list and setup the **Time Interval**; The cyclic task is operated according to the time interval setting from Task Manager.

(3) Triggered by Event



Choose **Triggered by Event** from the **Type** drop-down list and select from **Mode of Triggering** drop-down list.

➤ The mode of triggering contains the following events:

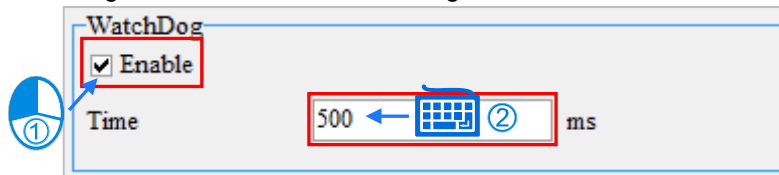
- Motion event (Motion control task)
- Rising edge or falling edge of local input points (I0~I7 and I10~I17)
- CANopen SYNC signal
- Z pulse rising edge of incremental encoder 1 or encoder 2

An event task is only executed once to the assigned event. The executing time for event task is determined by the time of event and event task according to priority.

5

● **Watchdog**

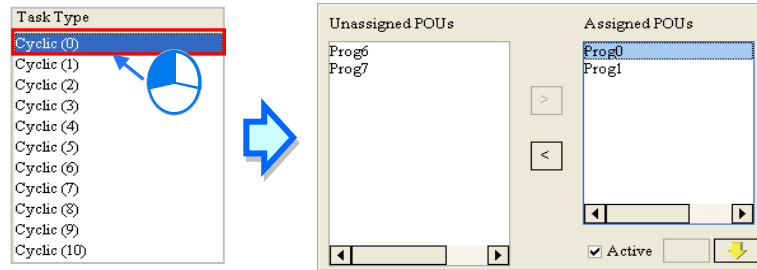
The Watchdog is used to monitor every task. When the task execution time exceeds the set watchdog time, the controller will enter Error state and the user program execution will stop. Every task can set its exceeding time and click the Watchdog function if needed.



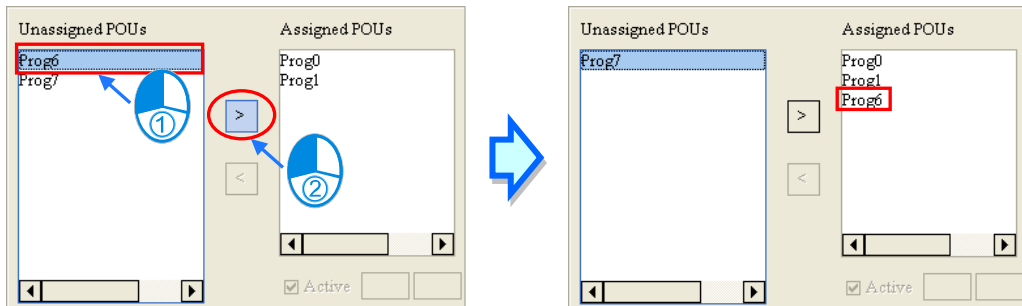
**5.5.2 Configuration of POU**

When Task Description setting is complete, the POU's will be configured in the TASK list. Please pay attention that one POU is assigned to one task, while POU that uses SFC for editing can only be configured to cyclic task.

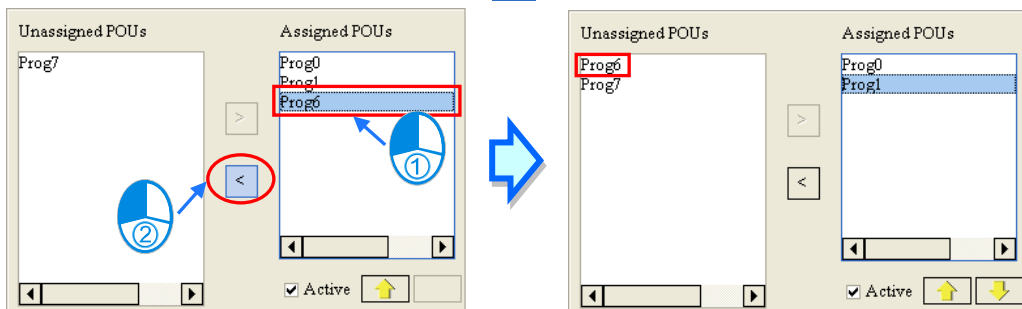
- (1) After a task in the **Task Type** section is selected, the assignment of the POU's will be displayed in the task management area. The POU's which have not been assigned to the task are in the **Unassigned POU's** section, and the POU's which have been assigned to the task are in the **Assigned POU's** section.



- (2) If users want to assign a POU in the **Unassigned POU** section to the task, they can select the POU in the **Unassigned POU** section, and click **>**.

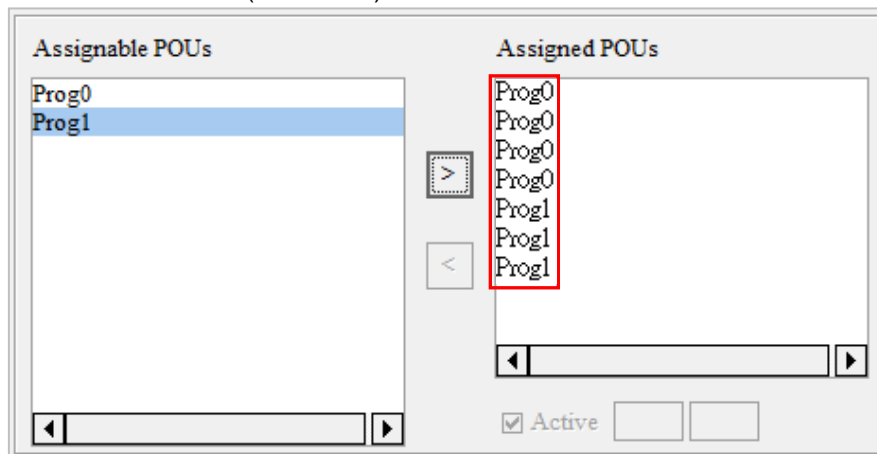


- (3) If the users want to remove a POU from the **Assigned POU** section, they can select the POU in the **Assigned POU** section, and click **<**.




5

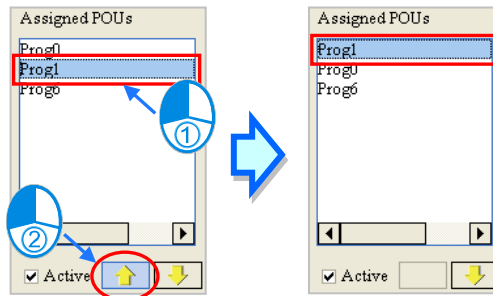
When using DVPxxMC series in project, the same POU can be reassigned to multiple tasks and be reassigned in the same task too (see below).



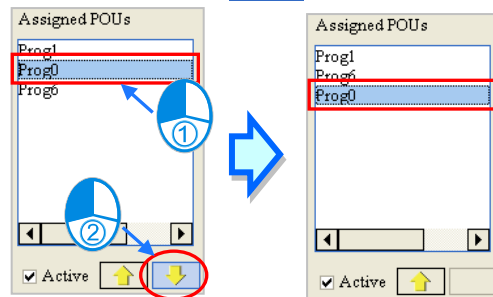
### 5.5.3 POU Order

When POU configuration is complete under the Task list, these POUs need to be arranged in execution order.

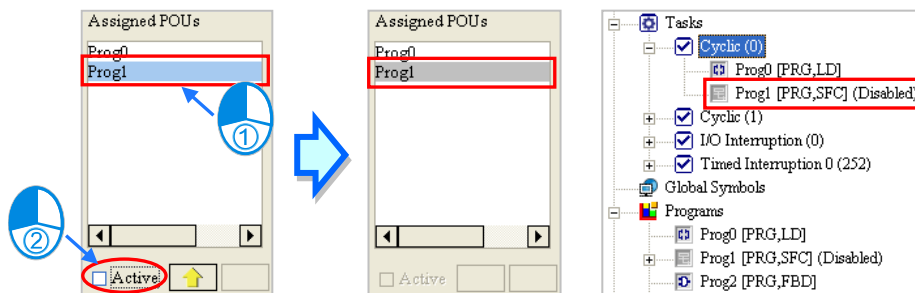
- (1) If users want to execute a POU in the **Assigned POUs** section earlier, they can select the POU in the **Assigned POUs** section, and click  to move the POU upward.



- (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.

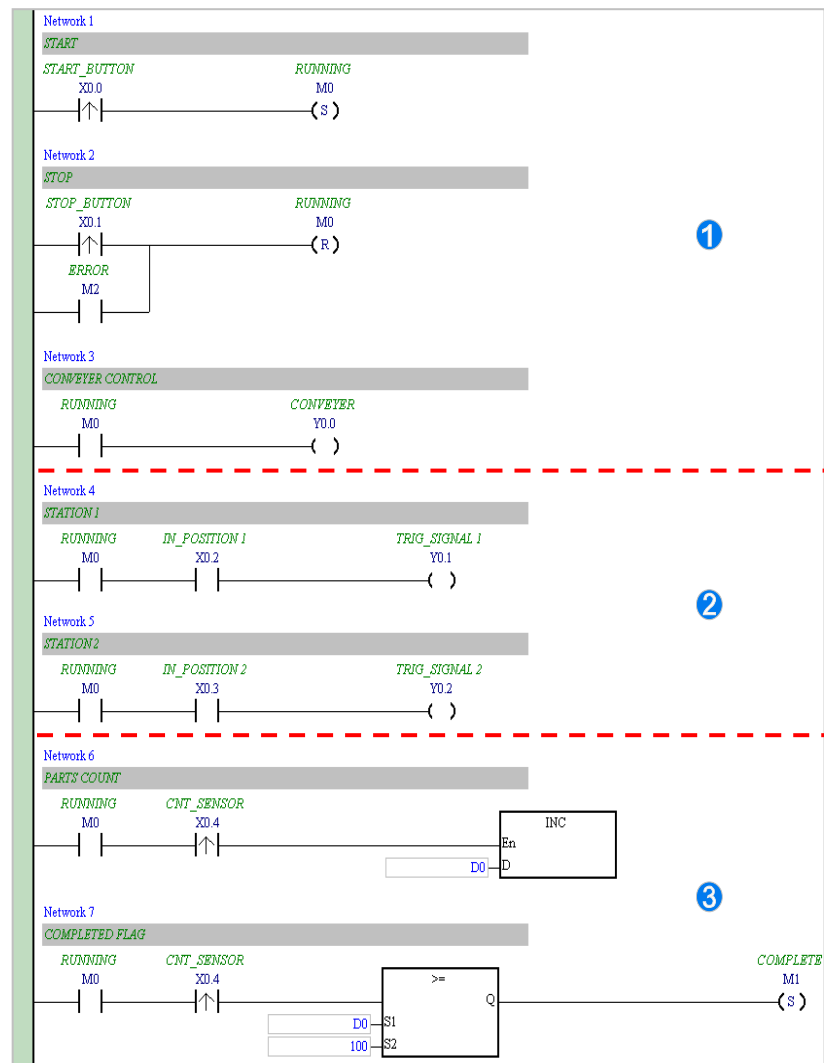
- (4) Click **OK** in the **Task Manager** window after the setting of the task is complete.

5

## 5.6 Example

### 5.6.1 Programming TASK and POU

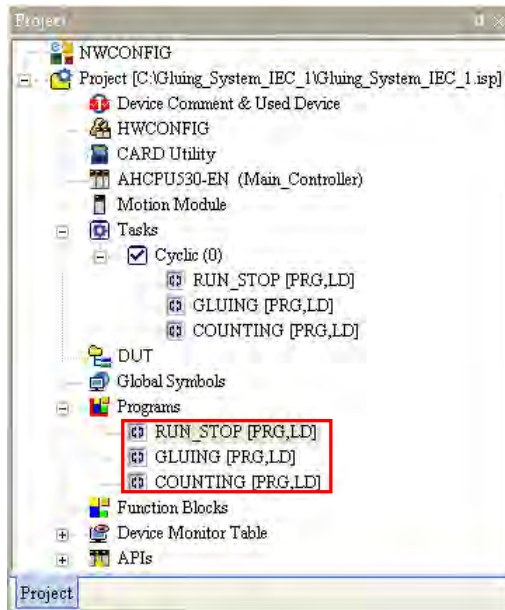
The examples created in chapter 4 will be rewritten in TASK and POU program. The program created in chapter 4 is shown below. It will be divided into three parts according to the functions listed below.



- 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.

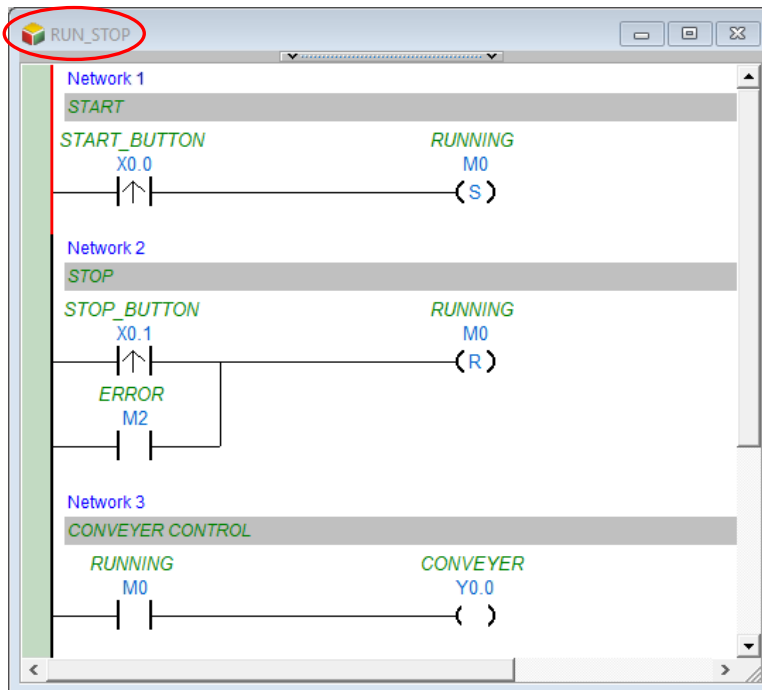
First, create a new project. Then, refer to chapter 4, and complete the hardware configuration. Finally, refer to section 5.4.1, and create three POUs in the **Programs** section in accordance with the three points listed above. These three POUs are assigned to **Cyclic (0)**.



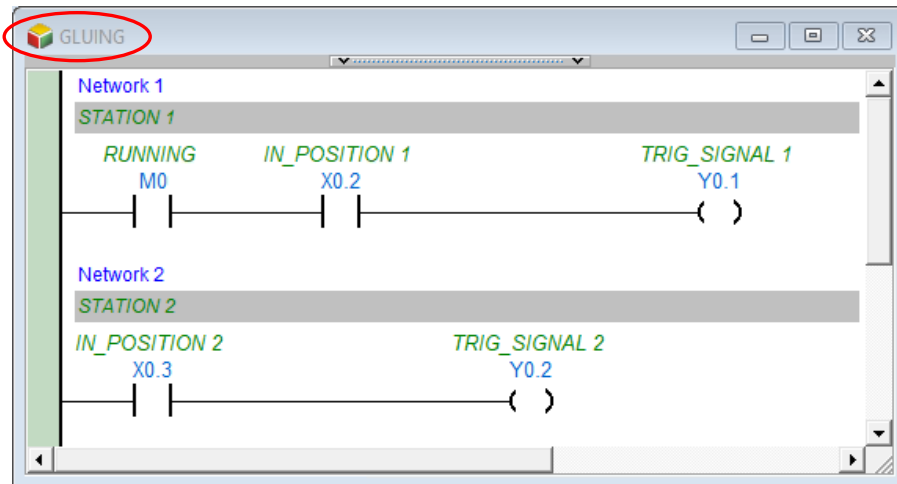
# 5

Please refer to chapter 4, and write the programs in the POUs shown below.

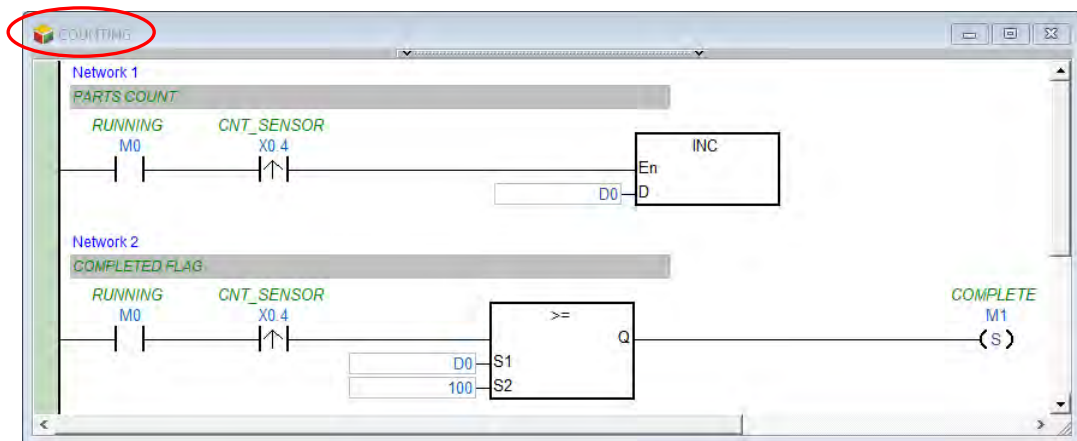
The program in **RUN\_STOP** functions to start or stop the system.



POU- **GLUING** to control gluing.

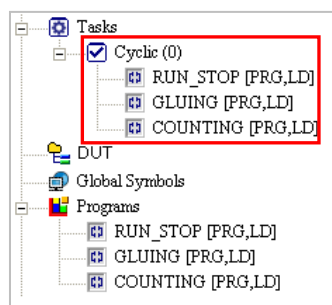


POU- **COUNTING** to finish the final counting.

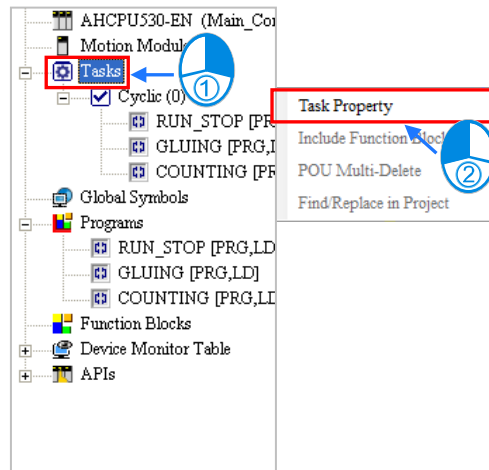


5

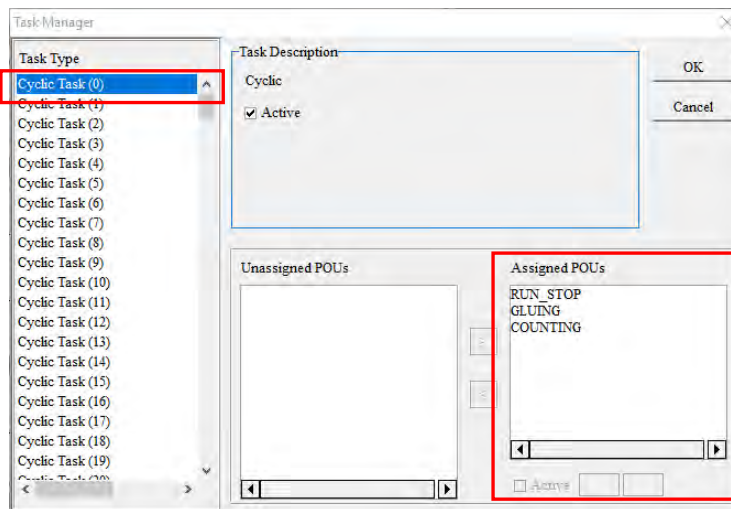
When programming is complete, please check the POU order under Tasks is the same as the diagram shown below.



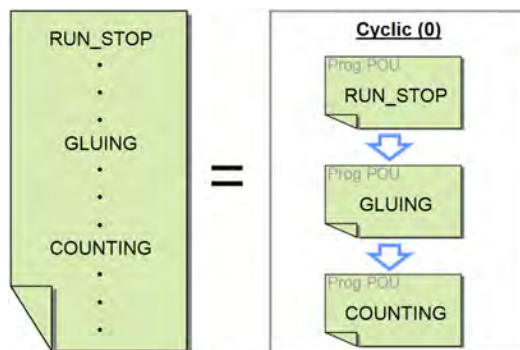
When re-planning the task configuration for each POU, right-click **Tasks** in Project section and choose **Task Property** from the quick menu to open **Task Manager** setting page and add POU to the assigned POUs in Cyclic (0).



5



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.

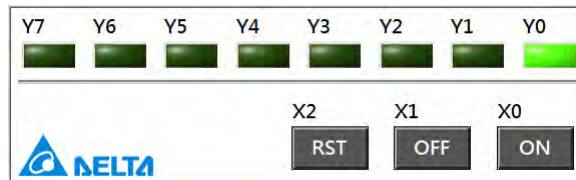




## 5.6.2 Example of an Interrupt Subroutine in a DVP Series PLC

### • 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

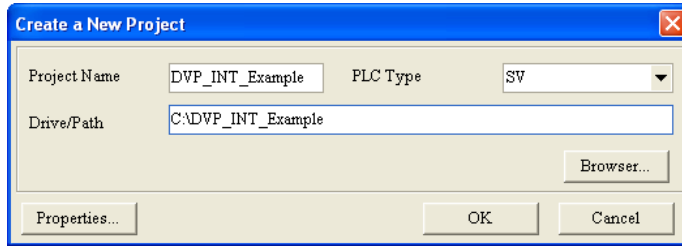
The model used in this example is a DVP-SV PLC, and four POU are created. The functions of the POU are described below.

POU name	Function
Run_Stop	<ol style="list-style-type: none"> <li>1. Reset the operation flag M0, transfer the value 1 to the output data register D0, and enable the interrupt.</li> <li>2. When X0 is turned from OFF to ON, the operation flag M0 is set to ON.</li> <li>3. When X1 is turned from OFF to ON, the operation flag M0 is set to OFF.</li> </ol>
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.)

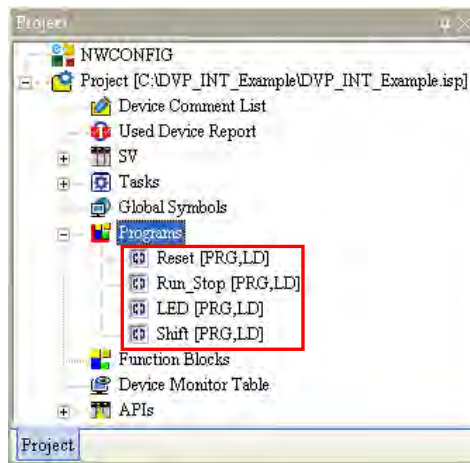
\*. Please refer to DVP-PLC Application Manual—Programming for more information about the interrupts in a DVP-SV series PLC and the related applied instructions.

• **Creating A New Project**

First, create a SV series project.

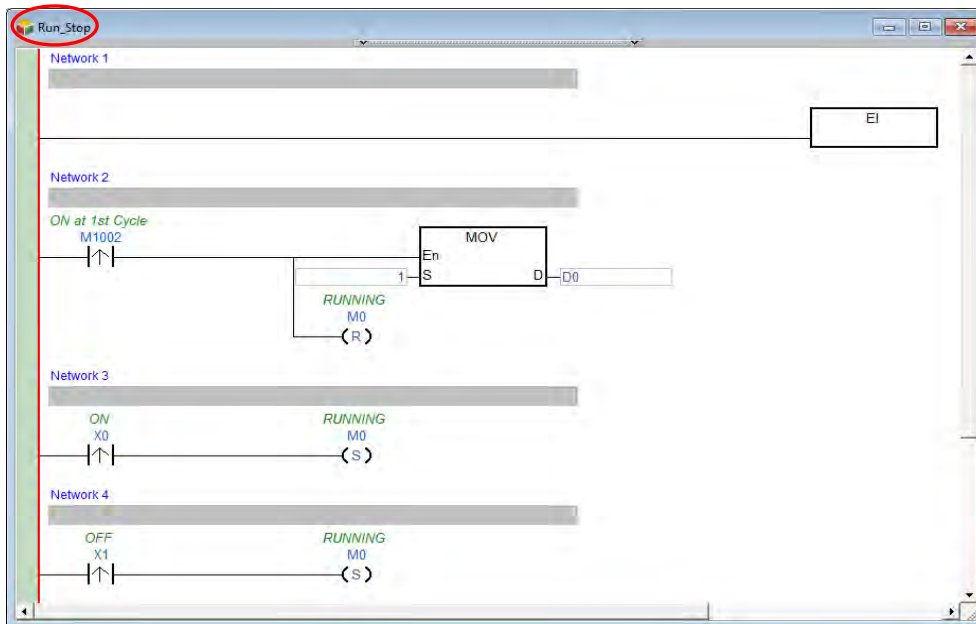


Under **Programs** in Project section, create for instance 4 POU's and **Task** column in POU Property can remain as **Cyclic** by default.

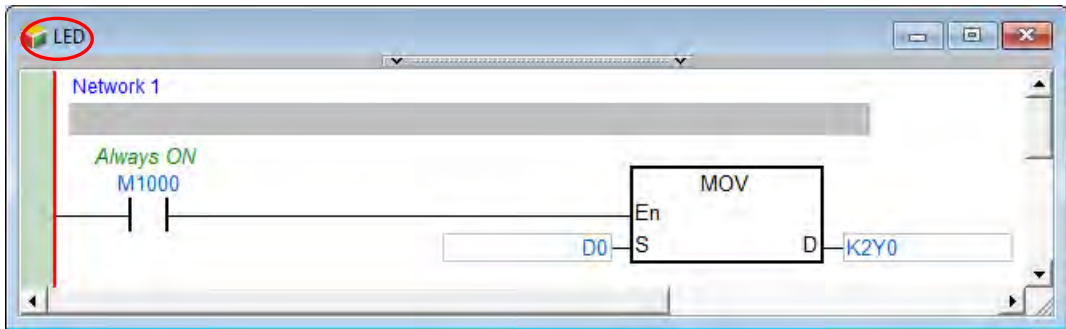


5

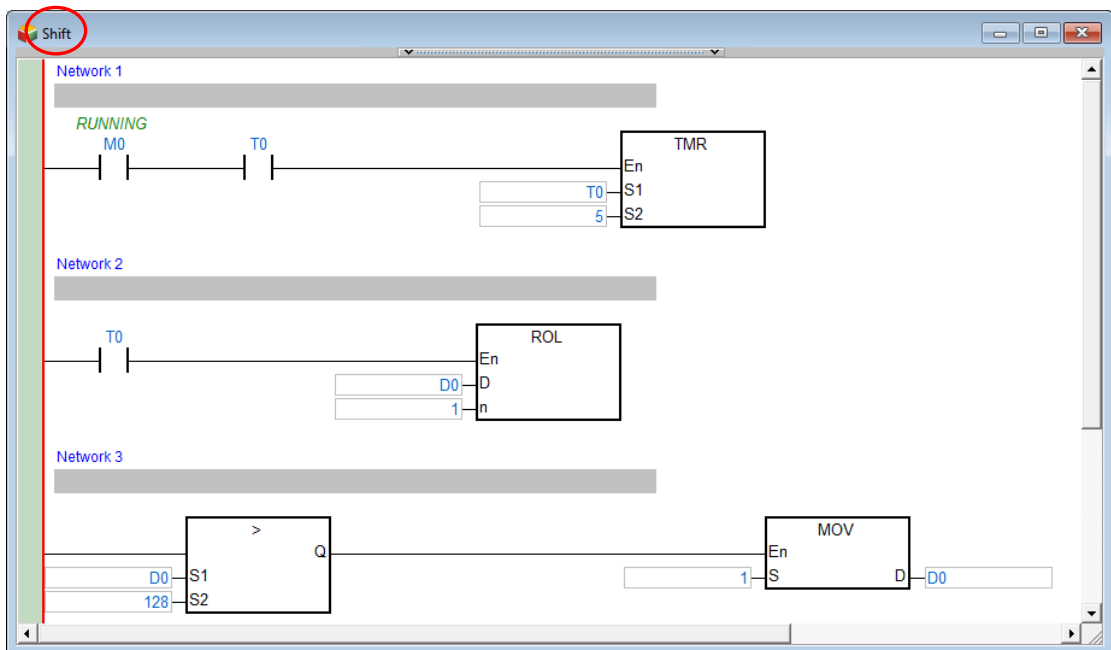
Write the programs in the POU's shown below.



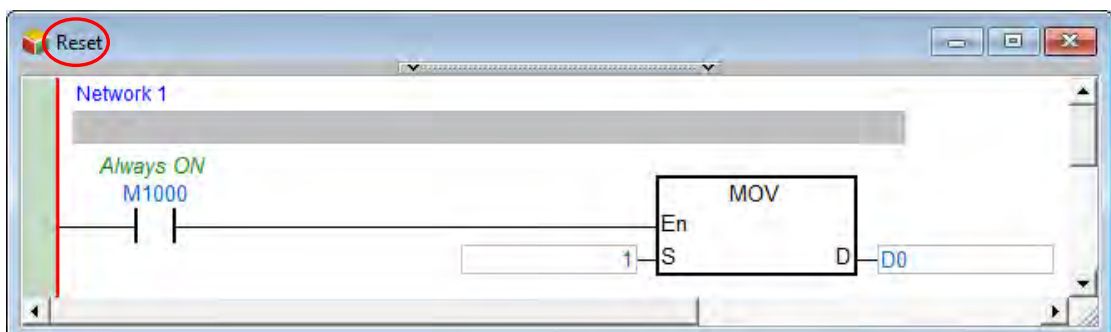
\*. The applied instruction MOV is in the Transmission Comparison section, and the applied instruction EI is in the Loop Control section.



\*. The applied instruction MOV is in the Transmission Comparison section.



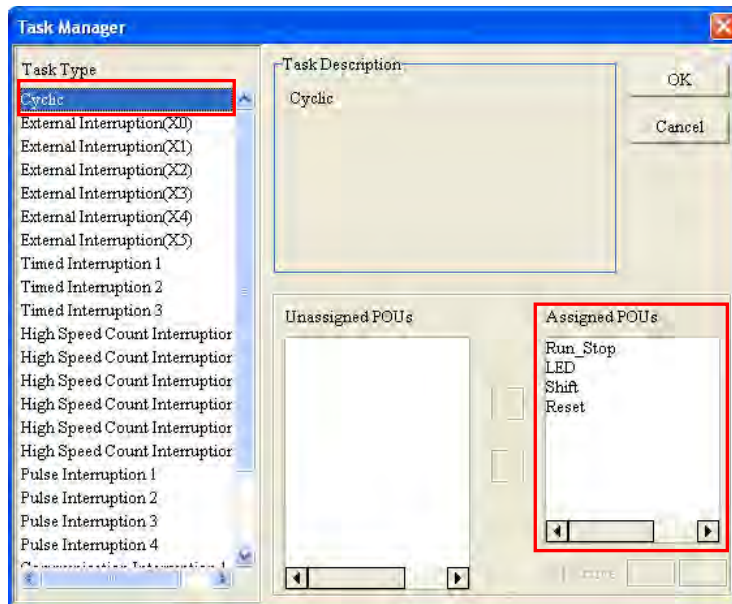
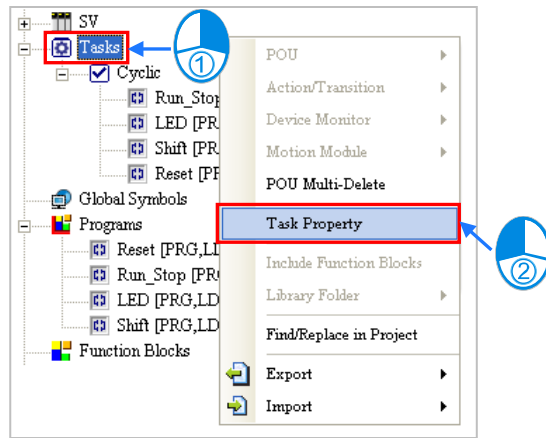
\*. The applied instruction TMR is in the Basic Instructions section, the applied instruction ROL is in the Rotation and Displacement section, the applied instruction MOV is in the Transmission Comparison section, and the comparison contacts are in the Contact Type Logic Operation section.



\*. The applied instruction MOV is in the Transmission Comparison section.

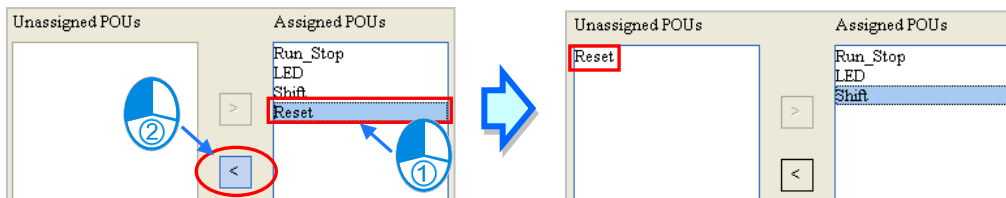
5

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**.

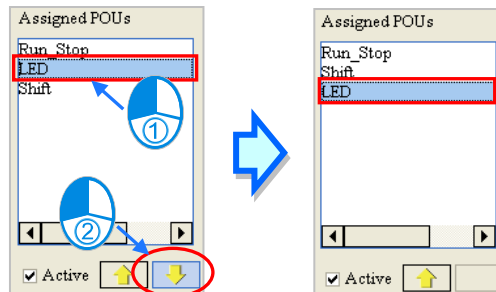


5

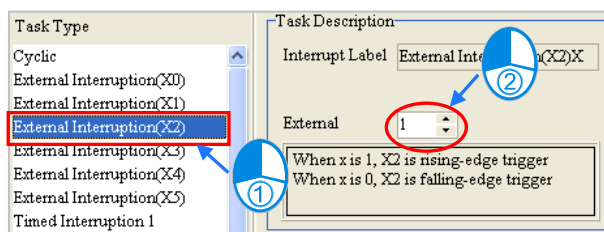
However, since **Reset** function is triggered by external interrupt, therefore, please remove **Reset** from the TASK list.



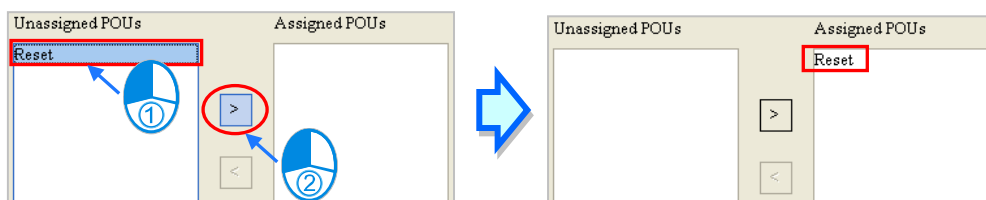
After the assignment of the POU to **Cyclic** is complete, users can arrange the POU in the **Assigned POU** section. Owing to the fact that **Run\_Stop** and **Shift** must be executed earlier than **LED**, the order in which **Run\_Stop**, **Shift**, and **LED** are listed in the **Assigned POU** section is **Run\_Stop**→**Shift**→**LED**.



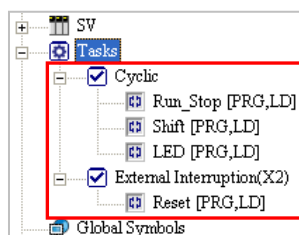
Select **External Interruption (X2)** in the **Task Type** section, and select 1 in the **External** box in the **Task Description** section. **External Interruption (X2)** is triggered when the signal passing through X2 goes from low to high.



Assign **Reset** to **External Interruption (X2)**. Owing to the fact that there is only one POU in the **Assigned POU** section, the users do not need to arrange the POU.



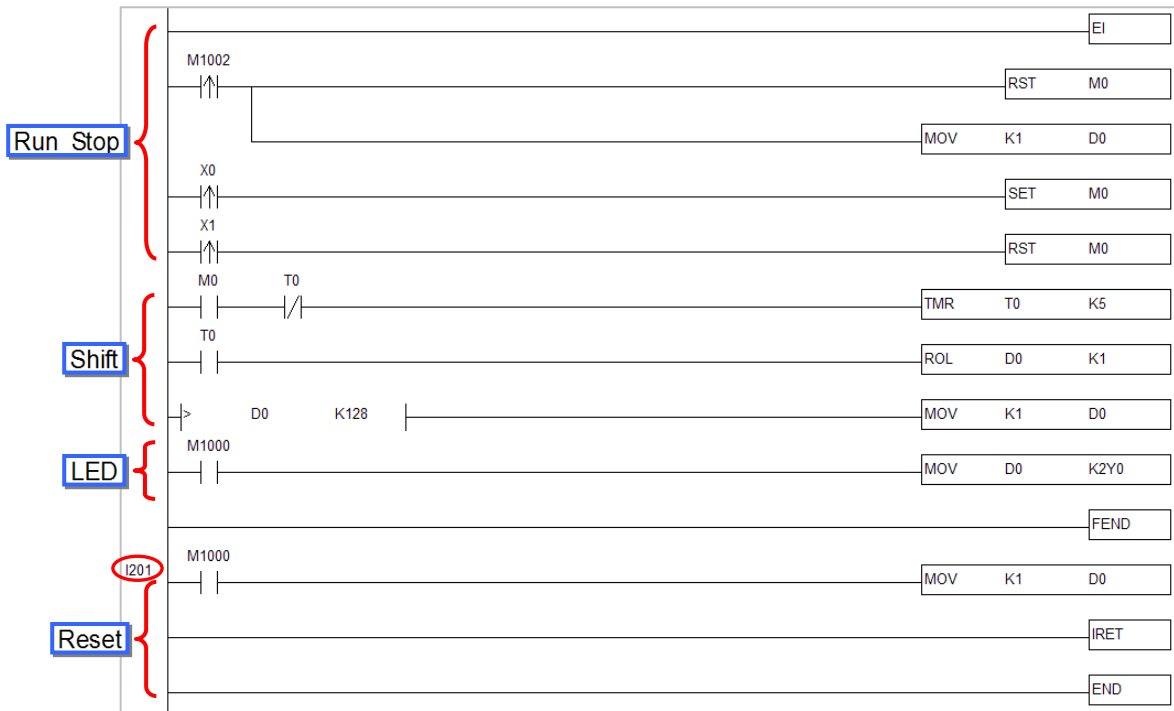
After **OK** is clicked, the assignment and the setting in the **Task Manager** window will be applied. The assignment is shown below. The writing of the program has been completed so far. The users can try to compile the program, download the program to the PLC, and test the program by themselves.



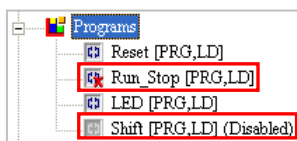
**Additional remark**

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 POU's which are assigned to the cyclic task. The order in which the the POU's 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 POU's **Run\_Stop** and **Shift**.

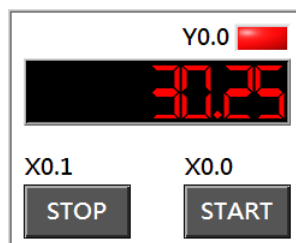


\*. WPLSoft is programming software for Delta PLCs. The programming architecture adopted by WPLSoft is traditional programming architecture. Please refer to WPLSoft User Manual for more information.

### 5.6.3 Examples of AH5x0 Series Interrupt Programs

- **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).



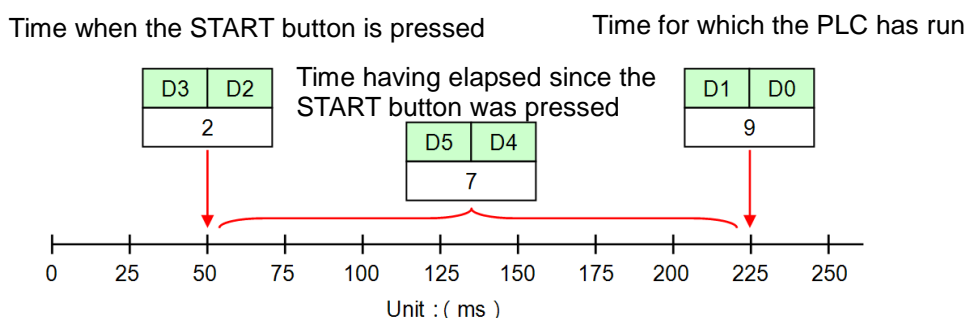
- **Hardware configuration**

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).



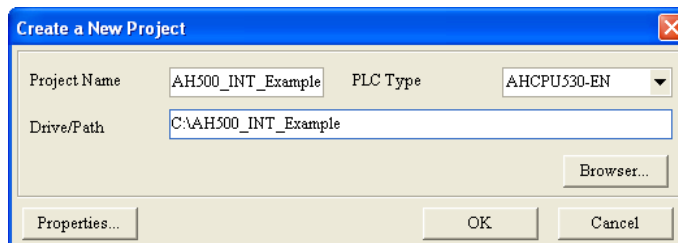
Examples in this section include timed interrupt in an AH5x0 series timed interrupt and the two control instructions concerning TKON and TKOFF. The following chart briefly describes the 5 POU's of a program.

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 <b>Timed Interruption (0)</b> .)
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, <b>Cyclic (0)</b> to which the POU's are assigned is enabled, and <b>Cyclic (1)</b> to which <b>Initialize</b> is assigned is disabled. As a result, <b>Initialize</b> is only executed during the first cycle.
Control	When X0.0 is turned from 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.

\*. For more information on AH5xo series interrupt indicator and related API instructions, please refer to AH500 operation manual and AH500 programming manual.

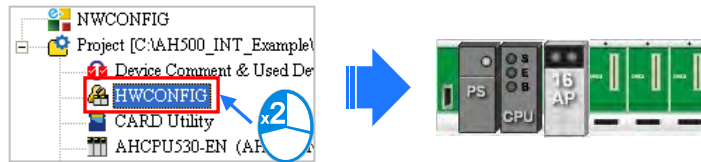
### • Creating New project

First, create an AHCPU530-EN series project.

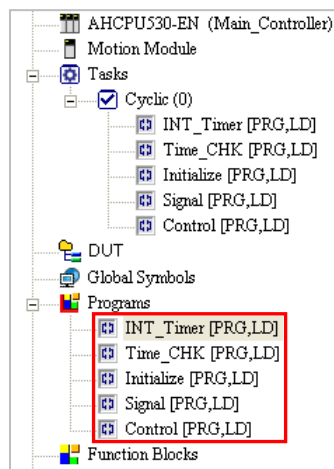




Click HWCONFIG to add a 16AP module and download hardware configuration and the parameter setting into the hosts.

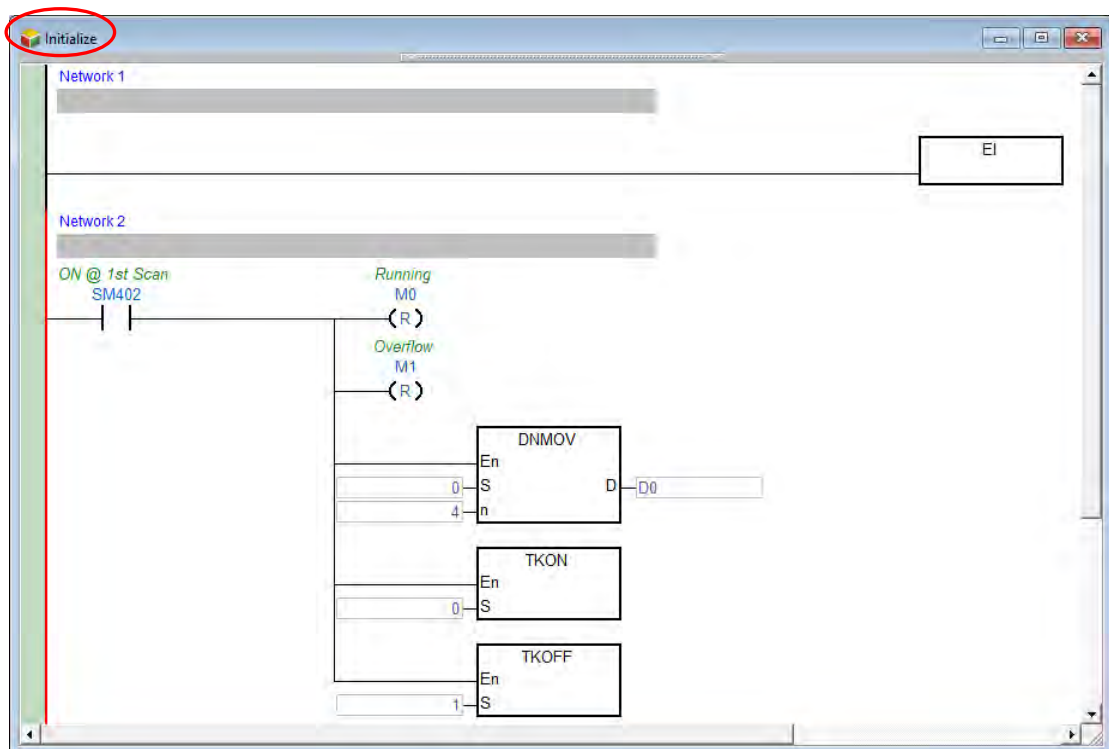


Create the 5 POUs in the **Programs** section in the project management area shown below. When these POU are created, they are assigned to **Cyclic (0)**.

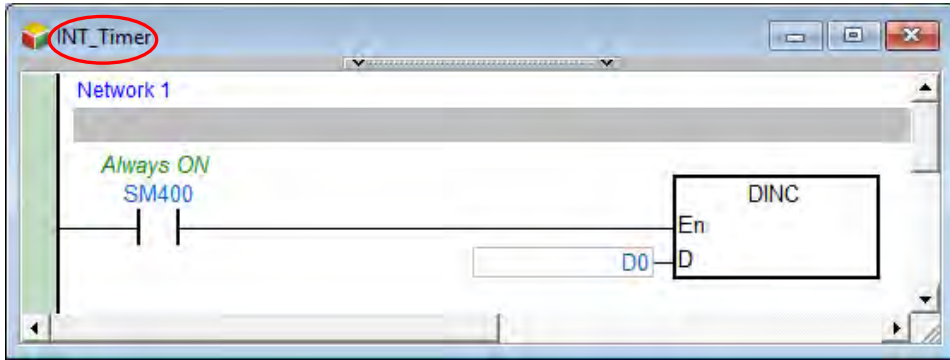


5

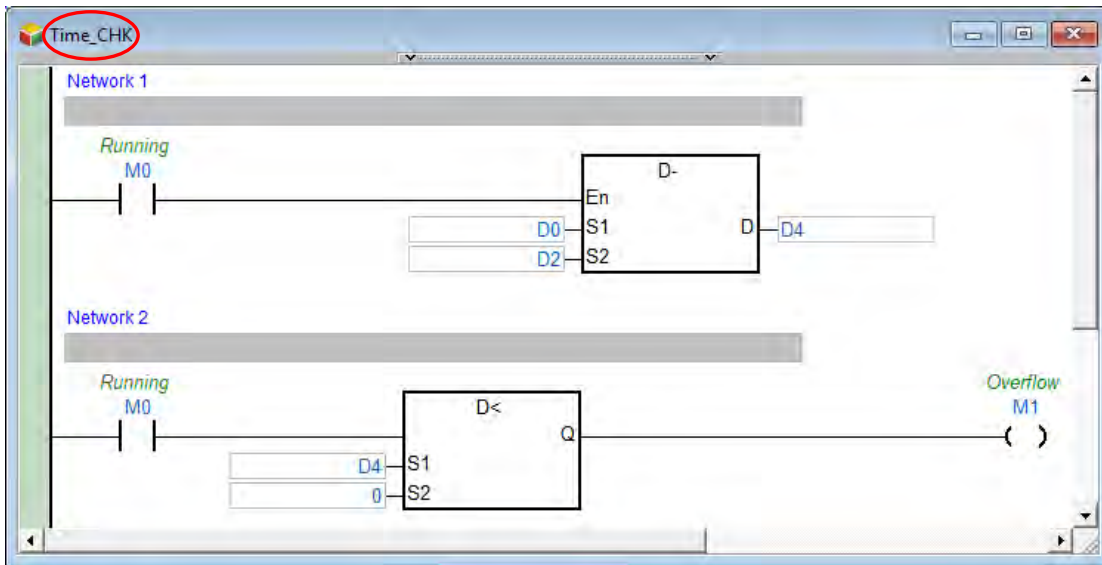
Please complete the following programming in each POU window.



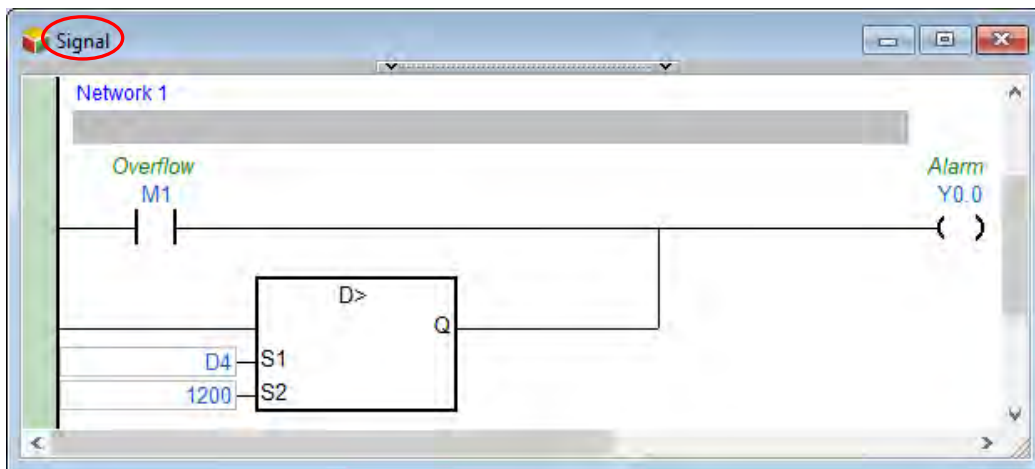
\*. The applied instruction EI is in the Loop Control section, the applied instruction DNMOV is in the Data Transfer section, and the applied instructions TKON and TKOFF are in the Task Control section.



\* The applied instruction EI is in the Four Arithmetic Operations section.

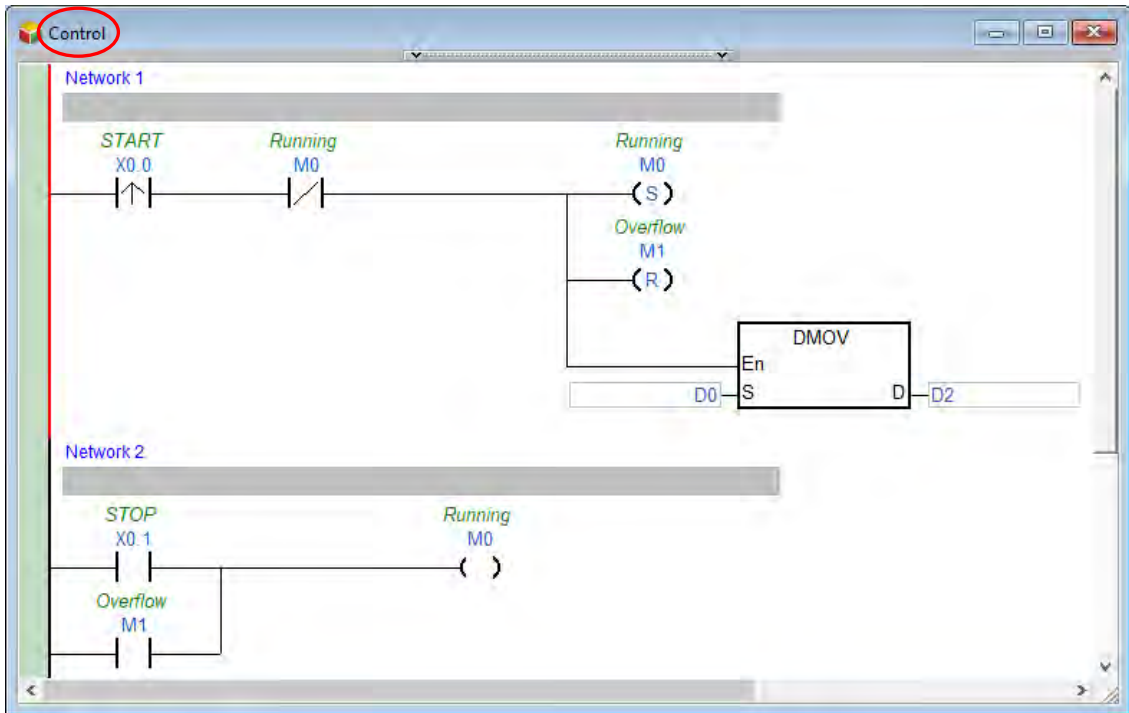


\* The applied instruction D- is in the Four Arithmetic Operations section, and the comparison contact D< is in the Contact Type Logic Operation section.



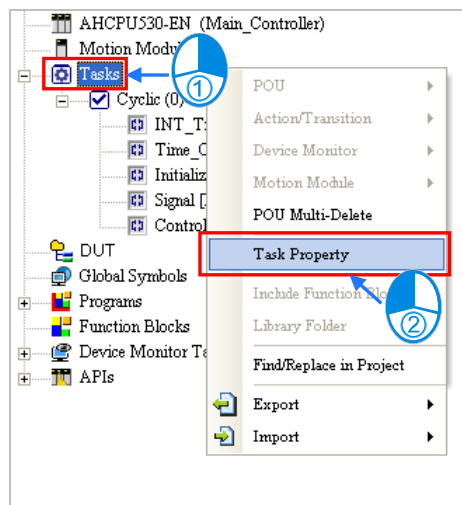
\* The comparison contact D> is in the Contact Type Logic Operation section.

5

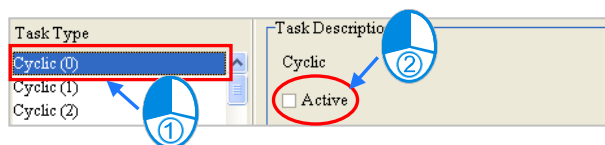


\*. The applied instruction DMOV is in the Data Transfer section.

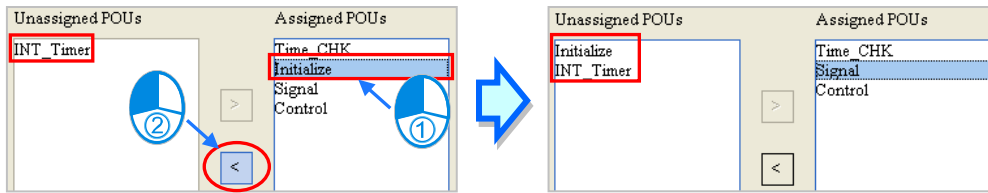
When programming is complete, double-click **Task Property**.



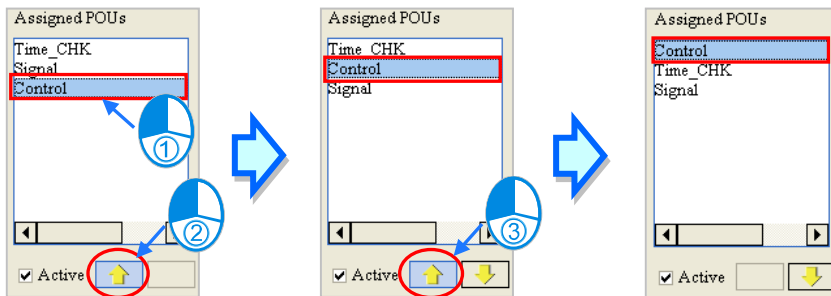
Set the property of **Cyclic (0)** first. Owing to the fact that **Cyclic (0)** is enabled by **Initialize**, the **Active** checkbox is unselected.



**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 POU** section for **Cyclic (0)**.

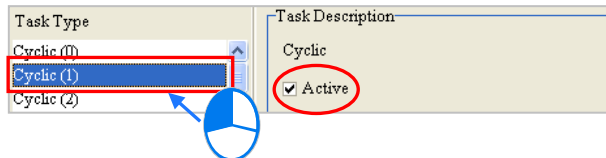


After the assignment of the POU to **Cyclic (0)** is complete, users can arrange the POU in the **Assigned POU** section. Owing to the fact that **Control** and **Time\_CHK** must be executed earlier than **Signal**, the order in which **Control**, **Time\_CHK**, and **Signal** are listed in the **Assigned POU** section is **Control**→**Time\_CHK**→**Signal**.

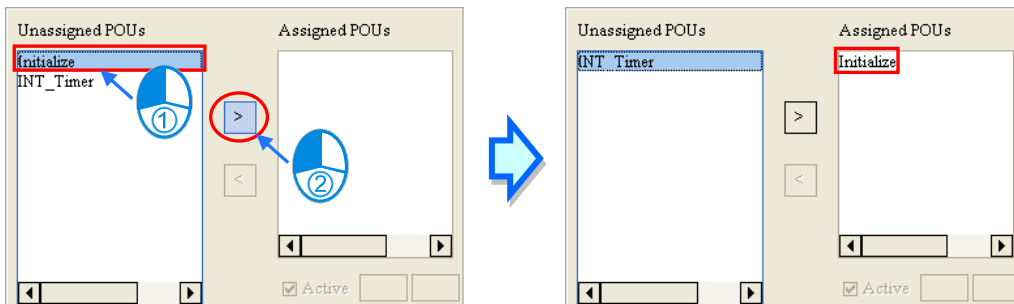


5

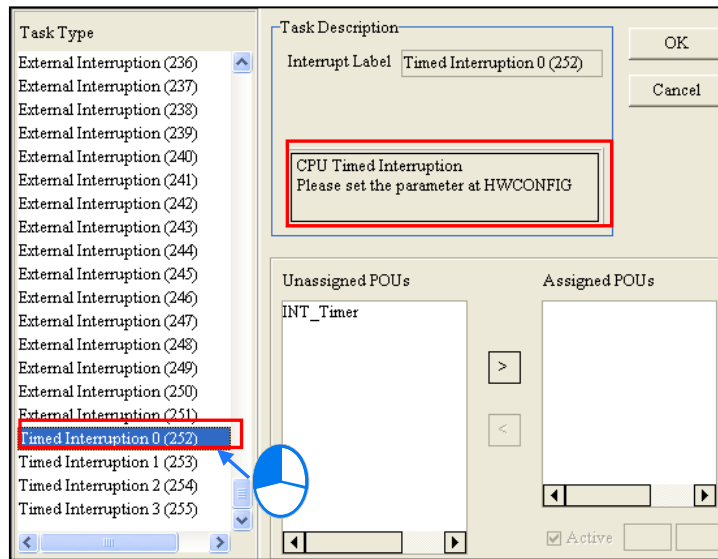
Select **Cyclic (1)** in the **Task Type** section. Owing to the fact that **Cyclic (1)** will be disabled after the first scan cycle is complete, the **Active** checkbox is selected.



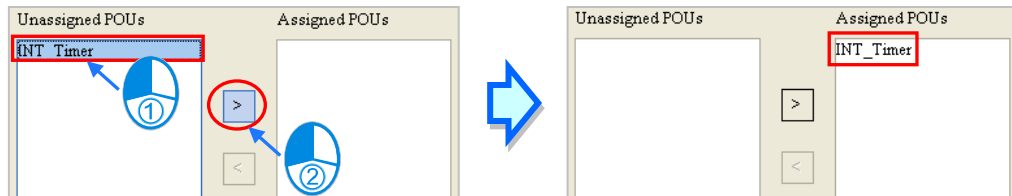
Assign **Initialize** to **Cyclic (1)**. Owing to the fact that there is only one POU in the **Assigned POU** section, the users do not need to arrange the POU.



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.

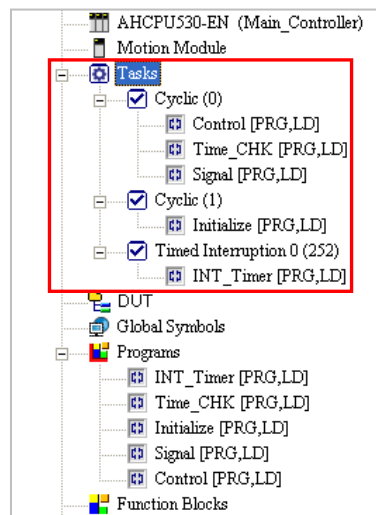


Assign **INT\_Timer** to **Timed Interruption 0 (252)**.

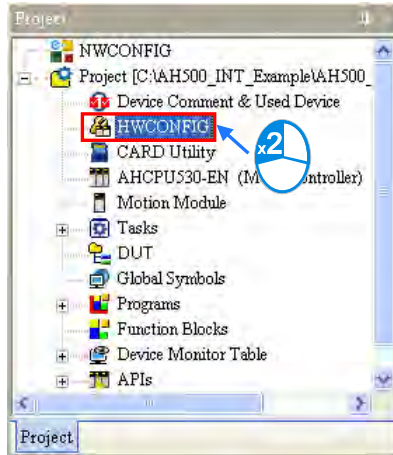


5

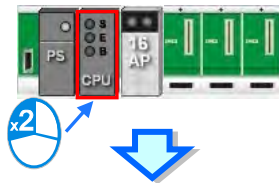
After **OK** is clicked, the assignment and the setting in the **Task Manager** window will be applied. The assignment is shown below.



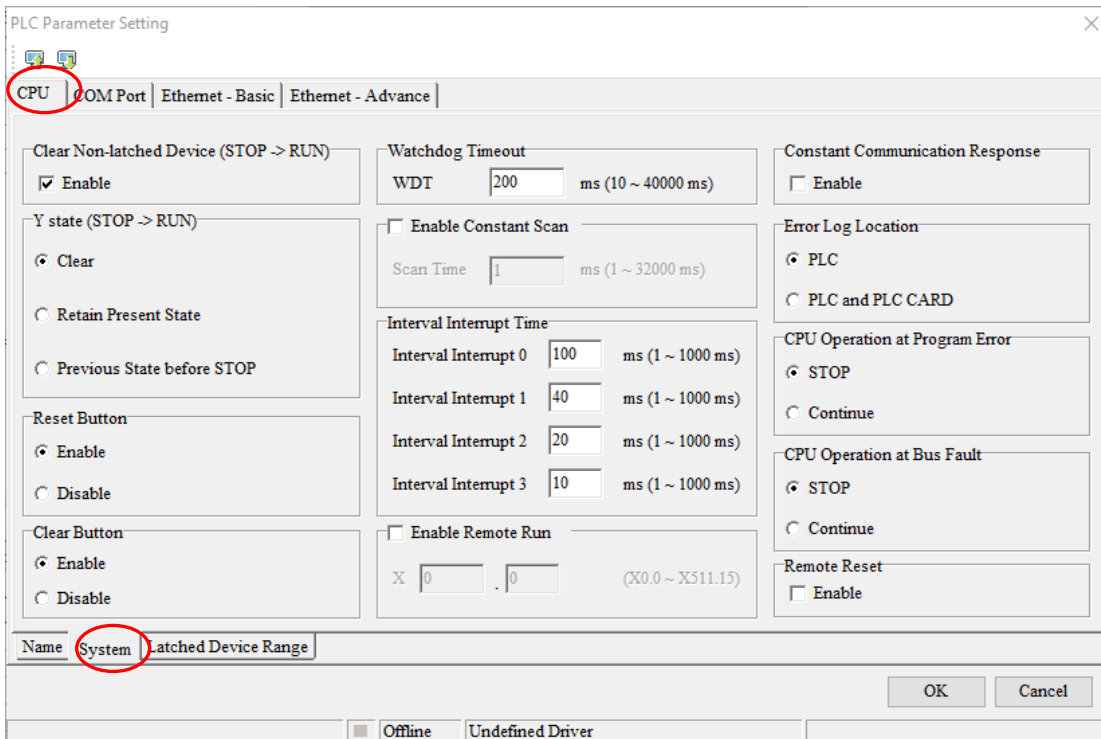
In order to set the property of **Timed Interruption 0 (252)**, HWCONFIG is started.



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.



5



Find the **Interval Interrupt Time** section. Type 25 ms in the **Interval Interrupt 0** box, and then click **OK**.

Interval Interrupt Time		
Interval Interrupt 0	25	ms (1 ~ 1000 ms)
Interval Interrupt 1	40	ms (1 ~ 1000 ms)
Interval Interrupt 2	20	ms (1 ~ 1000 ms)
Interval Interrupt 3	10	ms (1 ~ 1000 ms)

When complete, save the settings in HWCONFIG and download the setting value to the hosts to come into effect. Then, please compile the program and download to the host for testing.

**MEMO**

5





# Chapter 6 Symbol Variables

..

## Table of Contents

<b>6.1</b>	<b>Introduction of Symbols .....</b>	<b>6-2</b>
6.1.1	Application of Symbols and Creation of Identifiers .....	6-2
6.1.2	Symbol Variable Classification.....	6-3
6.1.3	Data Types.....	6-5
6.1.4	Symbol Variable Address Allocation and Initial Value.....	6-11
6.1.5	Indirect Assignment and Modification for Register Symbol Variable.....	6-14
6.1.6	Bit Operation of Symbol Variable (Only AH/AS Series ) .....	6-16
<b>6.2</b>	<b>Symbol Variable Management in ISPSoft.....</b>	<b>6-18</b>
6.2.1	Symbol Variable Table .....	6-18
6.2.2	Adding Symbol Variable .....	6-20
6.2.3	Principles of ARRAY or STRING Symbol .....	6-25
6.2.4	Modify Symbol Variables and Edit Symbol Table .....	6-29
6.2.5	Remove Symbol Variable Allocated Address .....	6-30
6.2.6	Downloading the Initial Values of the Symbols.....	6-31
6.2.7	Export Symbol Table .....	6-32
6.2.8	Importing a Symbol Table.....	6-33
6.2.9	Arranging the symbols.....	6-36
6.2.10	Filtering the Symbols.....	6-37
6.2.11	Device Resource Allocation Setting.....	6-38
<b>6.3</b>	<b>Example.....</b>	<b>6-39</b>
6.3.1	Planning a Symbol Table .....	6-39
6.3.2	Writing a Program .....	6-41

## 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.

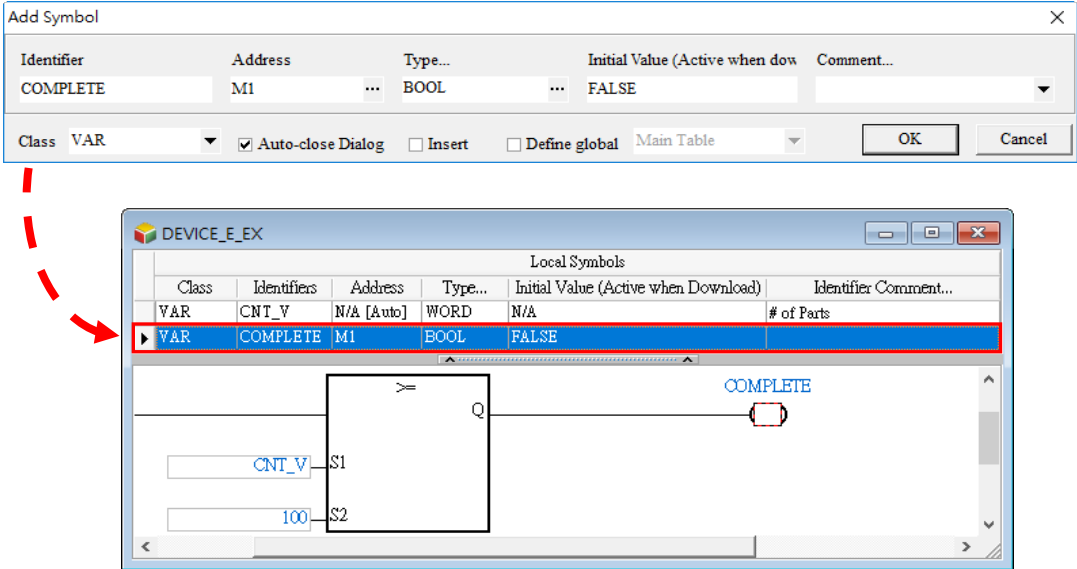
**\*. Variables in ISPSOft are called symbols. As a result, variables are the same as symbols in terms of meaning in this manual.**

### 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 POU 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 cannot 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.
- The name of a symbol and the name of an enumeration cannot be the same. Refer to chapter 8 for the enumeration usage.
- Spaces cannot be used. For example, “INPUT CH0” is an illegal name.
- Underlines can be used, but they cannot 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 cannot be used. For example, \*, #, ?, \, %, @, and etc. cannot be used.



\* When naming for the variable symbols, do not use digits or DFB\_(initials of the function keys by default). To achieve optimization, it is recommended to use what the system assigns for the variable address.

### 6.1.2 Symbol Variable Classification

Symbol variables can be categorized into five types based on functions. The symbol category supported by global variable and program (local variable) are the same. The following table depicts the characteristics and details of the program (local variable) and function block (local variable).

Symbol Variable Classification	Program (Local Variable)			Function Block (Local Variable)		
	AH/AS	DVP	DVPxxMC	AH/AS	DVP	DVPxxMC
VAR	✓	✓	✓	Position cannot be Assigned	Supports ONLY ES3 & Position	Position cannot be Assigned
VAR_RETAIN	Position cannot be Assigned (*1)	Supports ONLY ES3 & Position cannot be	✓			Position cannot be Assigned
VAR_INPUT				Position cannot be	Position cannot be	Position cannot be
VAR_OUTPUT				Position cannot be	Position cannot be	Position cannot be
VAR_IN_OUT				Position cannot be	Position cannot be	

\*1. Data type cannot be TIMER, COUNTER or STEP.

● **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.

● **VAR\_RETAIN - Latched symbol**

When symbol variable is declared as VAR\_RETAIN, it will automatically be configured in retain latched area. Therefore, when PLC host is latched, the symbol variables retains its value.

● **VAR\_INPUT - Symbol variable for function block input terminal point**

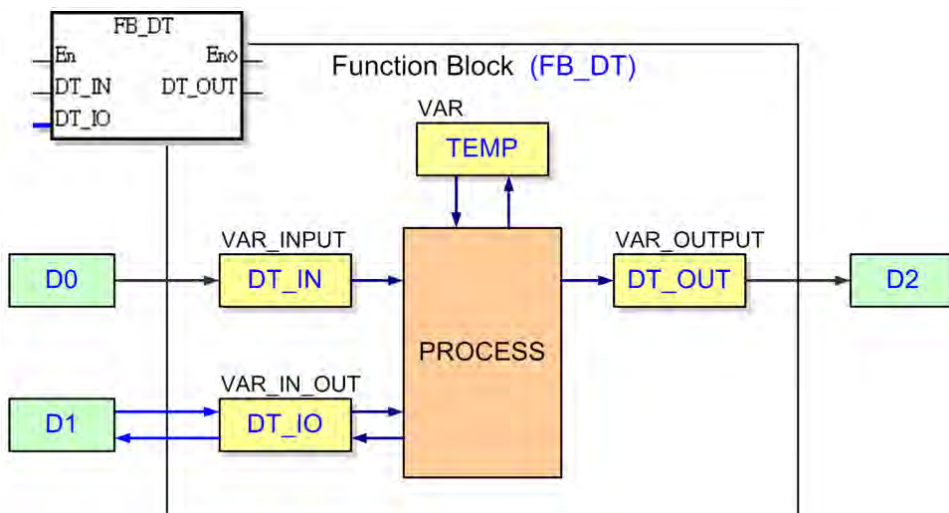
A symbol of this class is used as an input terminal point in a function block. If a function block is called, the symbol of this class can receive the input value from the caller. In ladder or function block editing environment, the VAR\_INPUT is arranged on the left side of the function block diagram and will configure one terminal point to receive the input value of the caller.

● **VAR\_OUTPUT - Symbol variable for function block output terminal point**

A symbol of this class is used as an output terminal point in a 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. In ladder or function block editing environment, the VAR\_OUTPUT is arranged on the right side of the function block diagram and will configure one terminal point to receive the input value of the caller.

● **VAR\_IN\_OUT - Symbol variable for feedback terminal point**

A symbol of this class is used as a feedback terminal point in a 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. In ladder or function block editing environment, the VAR\_IN\_OUT is arranged on the left side of the function block diagram and the terminal point is in blue bold lines used to connect with the caller.



6

### 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\_2 represents a 16-bit device which can be involved in arithmetic or data transfer.

The data types supported by ISPSOft are listed below. Different PLC types supports different data types, however, global variable and program (local variable) are mainly the same, but symbol variables declared by function block (local variable) are different (see below).

Data type	Description	AH/AS		DVP		DVPxxMC	
		Program	FB	Program	FB	Program	FB
<b>BOOL</b>	Boolean data type, use 0, 1, or TRUE or FALSE to represents a pin device status; DVPxxMC only supports TRUE or FALSE, default setting is FALSE.	✓	✓	✓	✓	✓	✓
<b>BYTE</b>	8-bit value, the effective range is 16#00~FF and can store 8-bit data size, default setting is 16#00.					✓	✓
<b>WORD</b>	16-bit value, the effective range is 16#00~FFFF and can store 16-bit data size.	✓	✓	✓	✓	✓	✓
<b>DWORD</b>	32-bit value, the effective range is 16#00000000~FFFFFFFF and can store 32-bit data size, default	✓	✓	✓	✓	✓	✓
<b>LWORD</b>	64-bit value, the effective range is 16#0000000000000000 ~ FFFFFFFFFFFFFFFFFF can store 64-bit data size,	Only support AH	Only support AH			✓	✓
<b>SINT</b>	8-bit signed integer, effective range is -128 ~ 127 and using positive/negative sign to					✓	✓

Data type	Description	AH/AS		DVP		DVPxxMC	
		Program	FB	Program	FB	Program	FB
<b>INT</b>	16-bit signed integer, effective range is -32768 ~ 32767 and using positive/negative sign to represent the highest bit. Default value is 0.	✓	✓	Only support AH	Only support AH	✓	✓
<b>DINT</b>	32-bit signed integer, effective range is -2147483648 ~ 2147483647 and using positive/negative sign to represent the highest bit. Default value is 0.	✓	✓	Only support AH	Only support AH	✓	✓
<b>LINT</b>	64-bit signed integer, effective range is -9223372036854775808 ~ 9223372036854775807 and using positive/negative sign to represent the highest bit. Default value is 0.	Only support AH	Only support AH			✓	✓
<b>USINT</b>	8-bit unsigned integer, effective range is 0 ~ 255, default value is 0.					✓	✓
<b>UINT</b>	16-bit unsigned integer, effective range is 0 ~ 65535, default value is 0.					✓	✓
<b>UDINT</b>	32-bit unsigned integer, effective range is 0 ~ 4294967295, default value is 0.					✓	✓
<b>ULINT</b>	64-bit unsigned integer, effective range is 0 ~ 18446744073709551615,					✓	✓

Data type	Description	AH/AS		DVP		DVPxxMC	
		Program	FB	Program	FB	Program	FB
	default value is 0.						
<b>REAL</b>	32-bit floating-point value, effective range is -3.402823e+38 ~ -1.175495e-38, 0 and 1.175495e-38 ~ 3.402823e+38, contains 32-bit to represent a decimal value. Default value is 0.	✓	✓	✓	✓	✓	✓
<b>LREAL</b>	64-bit floating-point value, effective range is -1.79769313486231e+308 ~ -2.22507385850721e-308, 0 and 2.22507385850721e-308 ~ 1.79769313486231e+308, contains 64-bit to represent a decimal value. Default value is 0.	Only support AH	Only support AH			✓	✓
<b>ARRAY</b>	Array data type When a symbol is declared, the length of an array and the data type which to be stored are specified. (The max. length of array is 512.)	✓	✓	✓	✓	✓	✓
<b>STRING</b>	String data type, use 8-bit to represent one ASCII character. AH/AS series has to declare assigned string length, maximum 128 characters; the string format of DVPxxMC series uses apostrophe, e.g. 'abcd' , the initial biggest length is	✓	✓	Only support ES3	Only support ES3	✓	✓

Data type	Description	AH/AS		DVP		DVPxxMC	
		Program	FB	Program	FB	Program	FB
	32000 characters; the initial character length by default is 80 characters.						
<b>STEP</b>	It is used as a recognition flag for step. (*1) (*2)	✓		✓	Do NOT support ES3		
<b>Function Block</b>	Represents a named function block. (*2)	✓	✓	✓	Only support ES3	✓	✓
<b>COUNTER</b>	Timer format use to represent timer device.	✓		✓	Do NOT support ES3		
<b>TIMER</b>	Timer format use to represent timer device.	✓		✓	Do NOT support ES3		
<b>POINTER</b>	Select VAR_IN_OUT in using WORD device as index indication. (*2)		✓		Only support ES3		
<b>T_POINTER</b>	Select VAR_IN_OUT in using timer as index indication. (*2)		✓		Only support ES3		
<b>C_POINTER</b>	Select VAR_IN_OUT in using timer variables as index indicator. (*2)		✓		Only support ES3		
<b>HC_POINTER</b>	Select VAR_IN_OUT in using high-speed timer variables as index indicator. (*2)		✓		Only support ES3		
<b>TIME</b>	Time format, using TIME#XXXXXXdXXhXXmXXsXXX.XXXms or					✓	✓

6



Data type	Description	AH/AS		DVP		DVPxxMC	
		Program	FB	Program	FB	Program	FB
	<p>T#XXXXXXdXXhXXmXXsX                      XX.XXXms for input, unit is ms. Effective range:                      T#0ms~T#213503d23h34m33s709.551ms, the value can be decimal (number) but not negative number; the number can exceed the effective range of TIME, e.g. T#25h and T#1d1h ; d, h, m, s, ms unit can be dropped, but one unit should remain and cannot arrange randomly; numbers must be decimal, default value is T#0ms.</p>						
<b>DATE</b>	<p>Time format, using DATE# Y-M-D or D#Y-M-D.                      Effective range : Display range:D#1970-01-01~D#2106-02-07. Unit is day. The Y, M,D cannot be dropped nor exceed the effective range; additional number 0 can be added before the number, the number must be decimal, default value is D#1970-01-01.</p>					✓	✓
<b>TOD</b>	<p>Time format. Using TIME_OF_DAY# hr: min: sec. ms or TOD# hr: min: sec. ms. Effective range: TOD#00:00:00~23:59:59.99. Unit is ms. When value is 0, TOD#00:00:00 is</p>					✓	✓

Data type	Description	AH/AS		DVP		DVPxxMC	
		Program	FB	Program	FB	Program	FB
	displayed; when value is 1, TOD#00:00:00.001 is displayed; when value is 86399999, TOD#23:59:59.999 is displayed; when value is 86400000, TOD#00:00:00 is displayed; when value is 4294967295, TOD#17:2:47.295 is displayed. Default value is TOD#00:00:00.						
DT	Time format. Using DT#Y-M-D-hr-min-sec for input. Effective range: DT#1970-01-01-0:0:0~DT#2106-02-07-6:28:15. Unit: s. Default value is DT#1970-01-01-0:0:0.					✓	✓

6

\*1. For more information on STEP, please refer to chapter 14.

\*2. The symbol variables for function block types and index types has special meanings. For more details, please refer to chapter 7- Function Block (FB) and library.

### 6.1.4 Symbol Variable Address Allocation and Initial Value

Every symbol variable is allocated to a corresponding device address based on data types and can setup an **initial value**. When downloading the project, users can select to write initial value in the corresponding device address of the symbol variable; when downloading DVPxxMC series project, users need to setup variables and device initial values in HWCONFIG.

The following chart provides a description on the symbol variable address allocation principle. The allocation principle differs based on different PLC types.

Model	Allocation Principle
AH/AS	<ul style="list-style-type: none"> <li>● Users can assign devices to the global symbols and the local symbols declared in the POU's of the program type. The system can also automatically assign devices to the global symbols and the local symbols declared in the POU's of the program type.</li> <li>● The system automatically allocates the local symbols of function blocks, but does not provide allocation by users.</li> <li>● The auto-allocated address is configured to the reserved memory section in the hosts, and do not occupy D or M device; for STEP, TIMER, or COUNTER types, the system still automatically allocate S, T and C device.</li> </ul>
DVP	<ul style="list-style-type: none"> <li>● Users can allocate devices to all the global symbols and the local symbols. The system can also automatically allocate devices to all the global symbols and the local symbols.</li> <li>● The devices allocated by the system are usable devices. (Users can set a range of devices which can be allocated automatically.)</li> </ul>
DVPxxMC	<ul style="list-style-type: none"> <li>● Users can assign devices to the global symbols and the local symbols declared in the POU's of the program type. The system can also automatically assign devices to the global symbols and the local symbols declared in the POU's of the program type.</li> <li>● The system automatically allocates the local symbols of function blocks, but does not provide allocation by users.</li> <li>● The auto-allocated address is configured to the reserved memory section in the hosts, and do not occupy I/Q/ M device.</li> </ul>

The relation between the data types and the device types which can be allocate is described below.

Data type	AH/AS		DVP	
	Device allocate by users	Device allocate by the system	Device allocate by users	Device allocate 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 (Only support AH)	Internal memory	None	
INT	D/L/X/Y	Internal memory	None	
DINT	D/L/X/Y	Internal memory	None	
LINT	D/L/X/Y (Only support AH)	Internal memory	None	
REAL	D/L/X/Y	Internal memory	D	D
LREAL	D/L/X/Y (Only support AH)	Internal memory	None	
STRING	D/L/X/Y	Internal memory	None	
STEP	S (Only support AH5x0)	S	S	S
COUNTER	C/HC	C	C	C
TIMER	T	T	T	T
ARRAY	<p>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.</p> <p>The devices assigned to a symbol whose data type is ARRAY cannot be SR/SM/E/F devices.</p>			

6

Data type	DVPxxMC	
	Device assigned by users	Device assigned by the system
BOOL	I/Q/M of B and X	Internal memory
BYTE	I/Q/M of B device	Internal memory
WORD	I/Q/M of W device	Internal memory
DWORD	I/Q/M of D device	Internal memory
LWORD	I/Q/M of L device	Internal memory
USINT	I/Q/M of B device	Internal memory
SINT	I/Q/M of B device	Internal memory
UINT	I/Q/M of W device	Internal memory
INT	I/Q/M of W device	Internal memory
UDINT	I/Q/M of D device	Internal memory
DINT	I/Q/M of D device	Internal memory
LINT	I/Q/M of L device	Internal memory
ULINT	I/Q/M of L device	Internal memory
REAL	I/Q/M of D device	Internal memory
LREAL	I/Q/M of L device	Internal memory
STRING	I/Q/M of B device	Internal memory
TIME	I/Q/M of L device	Internal memory
DATE	I/Q/M of D device	Internal memory
TOD	I/Q/M of D device	Internal memory
DT	I/Q/M of D device	Internal memory
ARRAY	I/Q/M device and base on types to determine devices (B,W,D,L), WORD Array does not support the assigned address writing for WORD[ArrayIndex].Bit	Internal memory

\*1. If the device assigned to a symbol is an internal memory in an AH/AS series CPU module, users cannot know the actual device address.

\*2. Please refer to section 6.2.3 for more information about ARRAY and STRING.

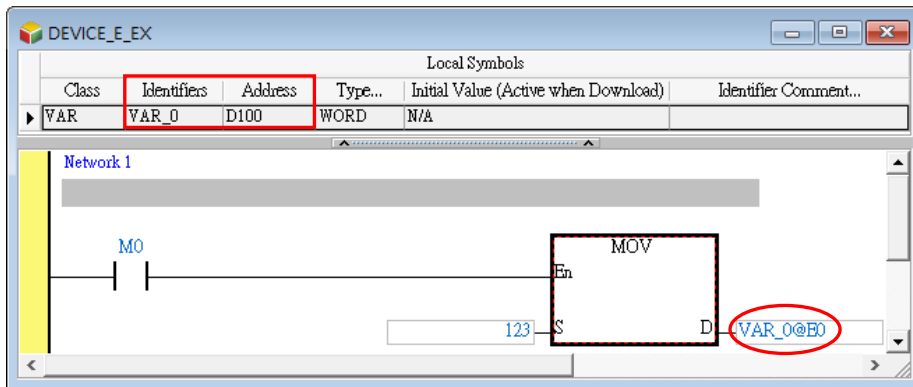
\*3. Please refer to section 6.2.11 for more information about setting a range of devices which can be assigned automatically in a DVP series PLC.

\*4. A symbol in a function block has a special significance. Please refer to chapter 7- Function Block and Library 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.

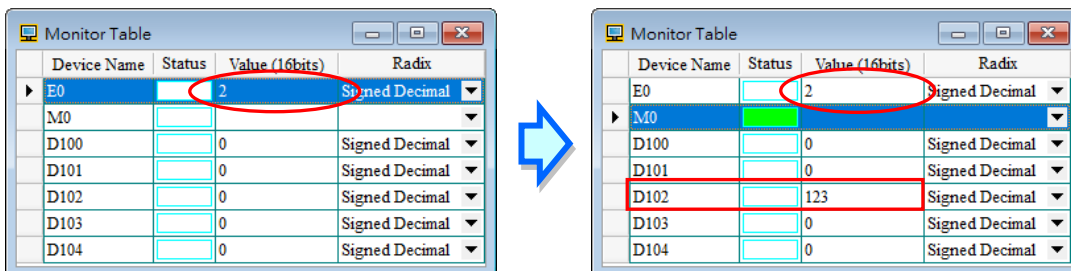
### 6.1.5 Indirect Assignment and Modification for Register Symbol Variable

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.



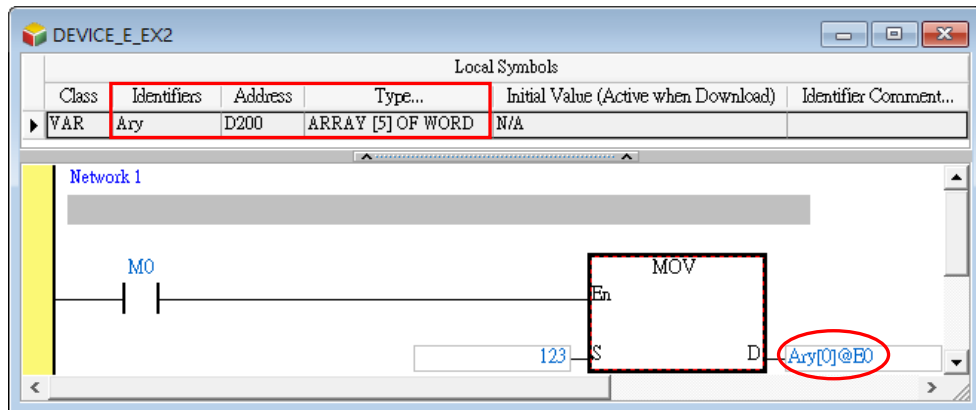
The data stored in an index register indicates the offset for an object modified. Take the figure above for example. If the value in the index register E0 is 2, VAR\_0@E0 indicates that 2 is added to the device address (D100) assigned to VAR\_0, that is, VAR\_0@E0 represents D102. When M0 is ON, the value 123 is transferred to the data register D102.

6



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 cannot 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	AH/AS	DVP	DVPxxMC
Modifying operands	MOV 100 VWD@E0	According to instruction	According to instruction	Do NOT Support
Modifying contact or coil	LD VBL_0@E0 OUT VBL_1@E1	Support	According to model types	Do NOT Support
Modifying symbol variable of ARRAY	ARY[0]@E0	Support	Support	Do NOT Support

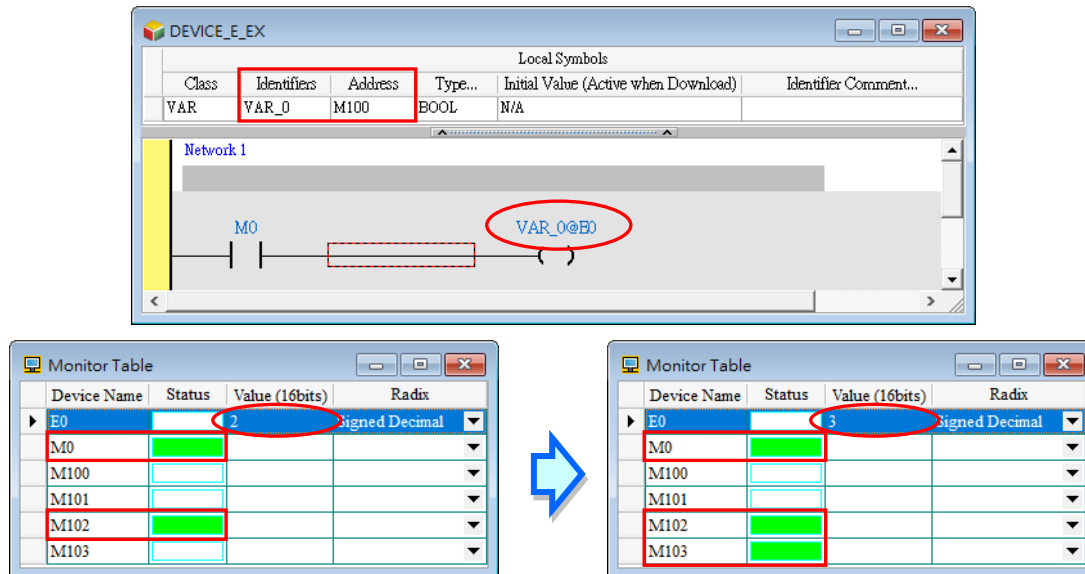
\*1. In an AH/AS series CPU module, only the E devices are index registers. In a DVP series PLC, the E devices and the F devices are index registers. Please refer to the operation manual for a model for more information.

\*2. An index register cannot 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 Operation of Symbol Variable (Only AH/AS Series )

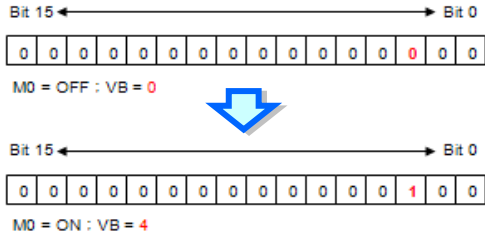
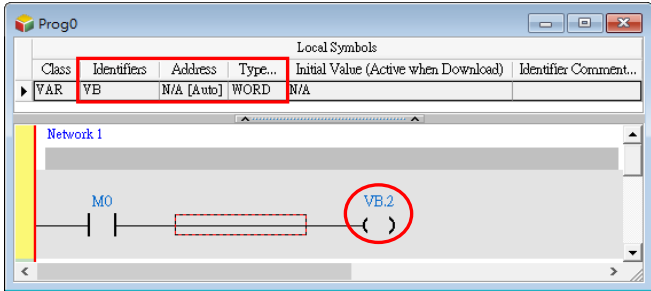
An AH/AS series CPU module supports the manipulation of the bits in a word device. The format is **Word device.Bit number**. For example, D0.2 indicates that bit 2 in D0 is manipulated. The same rule applies to the symbols. The format is **Identifier.Bit number**, e.g. VB.2. The rule also applies to 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 cannot 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.

6





In an AH/AS 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.

The following modifying ways are also available for AH Series CPU modules.

Example 1: 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 2: 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.

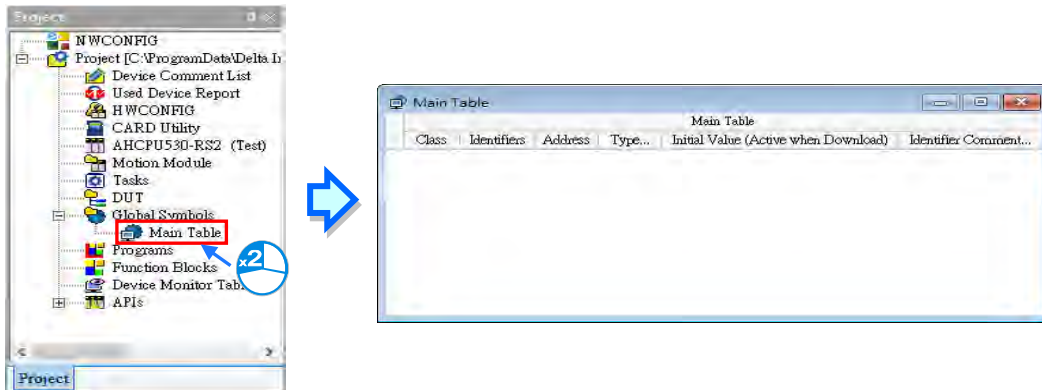
- \*1. If a bit in a device is manipulated, the system will faithfully get the state of the bit. Users have to judge whether the state of the bit is significant by themselves.
- \*2. In the present version of ISPSOft, a bit number must be within the range between 0 and 15. However, in the future version 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.

## 6.2 Symbol Variable Management in ISPSOft

### 6.2.1 Symbol Variable Table

- **Global symbol table**

Double-click the option **Main Table** under the **Global Symbols** in the project management area to open the **Main Table** window.

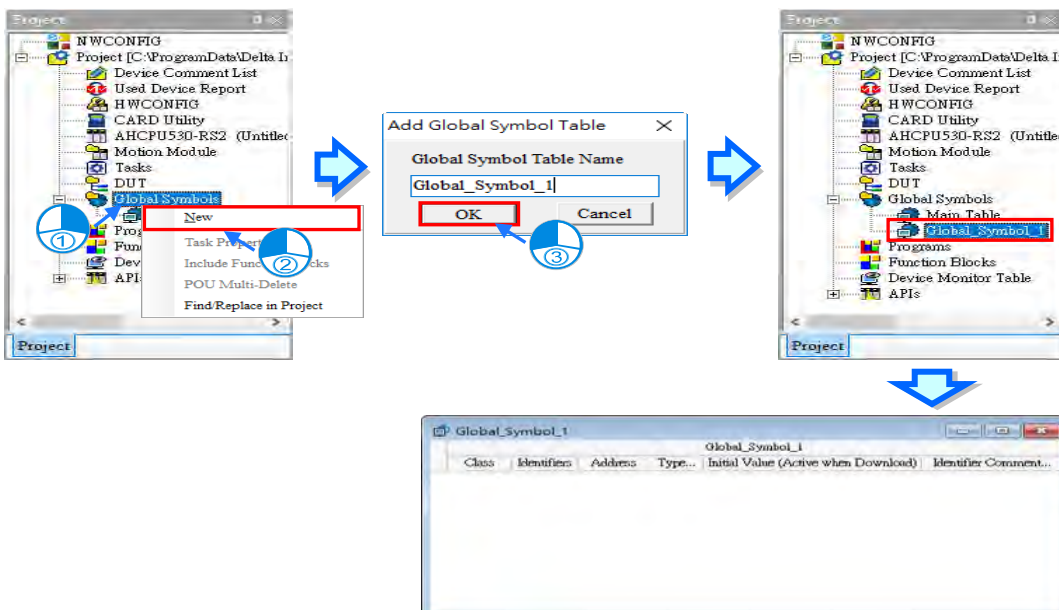


The option Global Symbols may have more sub-options for some series. For example the AHxxEMC series has more sub-options, including Main Table, Axis Table, and EtherCAT Table. Please refer to the related chapters for more information on how to use these symbols.



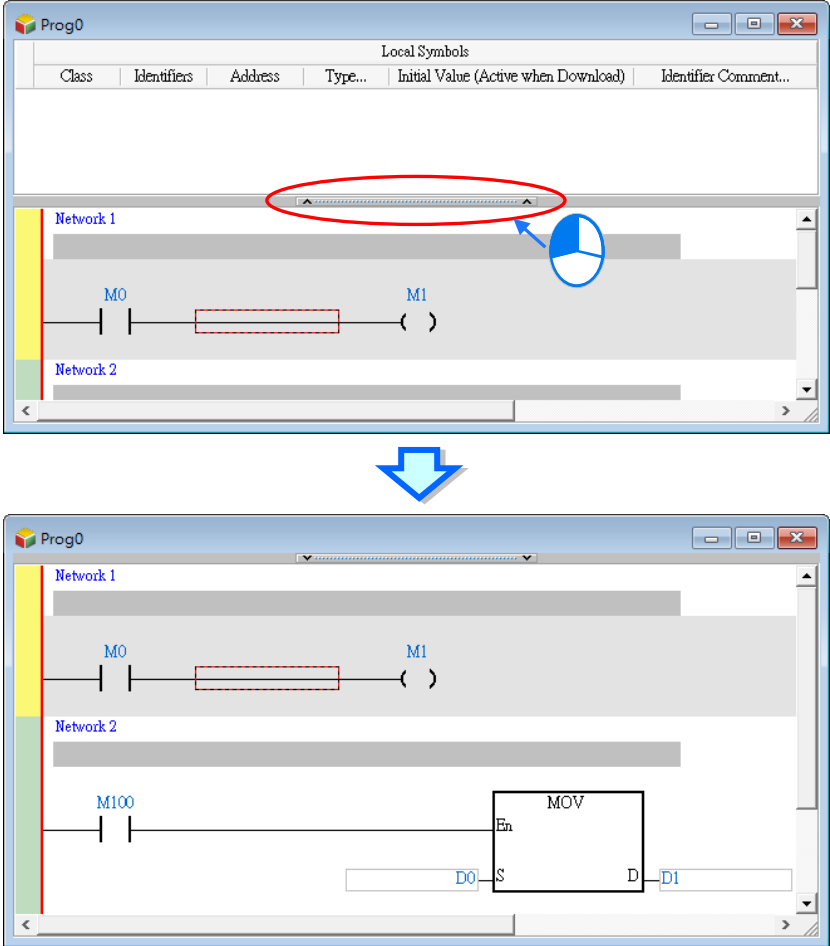
# 6

Right-click **Global Symbols** in the project management area and click **NEW** to open Add Global Symbol Table and click **OK** to create a Global Symbol node. Click the **Global Symbol node** to open the Global Symbol Table. To delete the newly created node, you can right-click the newly created node and click **Delete**, the node will be deleted.



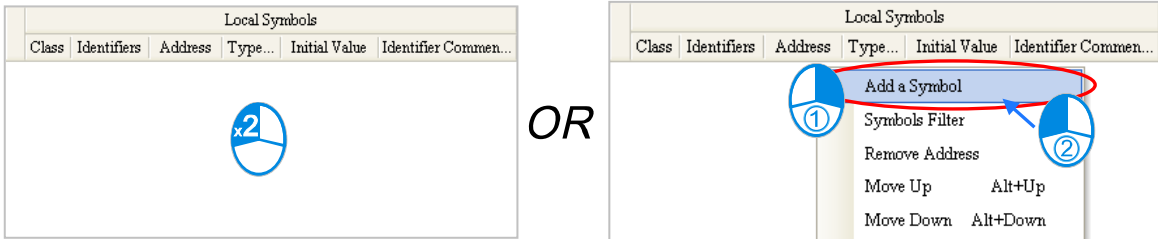
● Local symbol table

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.

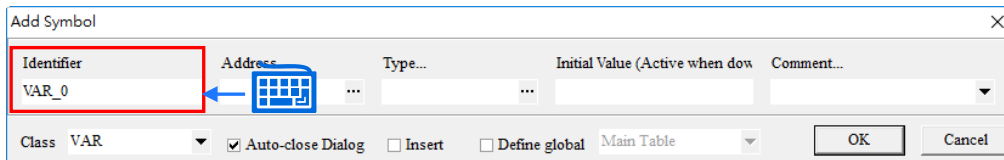


### 6.2.2 Adding Symbol Variable

- (1) After users double-click or right-click the blank in a symbol table, they can click **Add a Symbol** on the context menu.



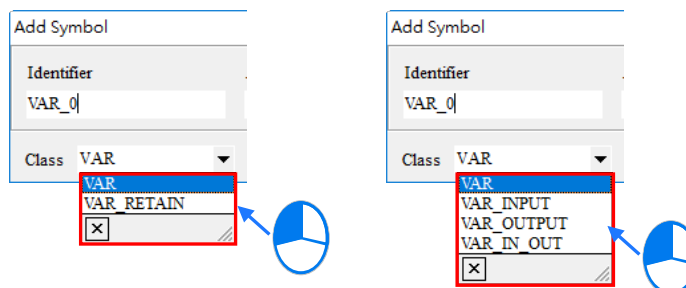
- (2) After **Add a Symbol** is clicked, the **Add Symbol** window will appear. Type an identifier in the **Identifier** box. Please refer to section 6.1.1 for more information about the regulations of creating the identifier of a symbol.



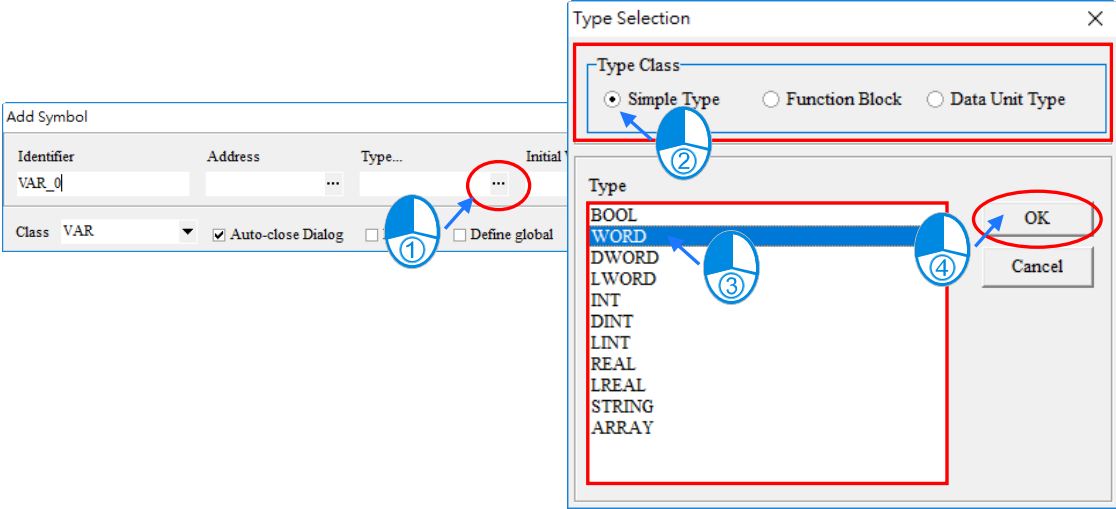
- \*1. The window which appears varies with the model selected, and the symbol table created (the global/local symbol table). However, the setting does not vary.
- \*2. There is a way to open a declaration window in every programming environment. Please refer to chapter 8~chapter 12 for more information.

6

- (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.



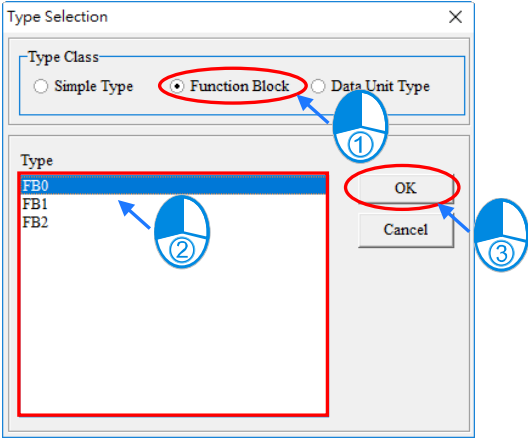
- (4) Click the button at the right side of the **Type...** box, set the data type of the symbol in the **Type Selection** window, and click **OK**. Please refer to section 6.1.3 for more information about the data types.



The setting of **Function Block**, **Data Unit Type**, **STRING**, and **ARRAY** are described below.

● **Function Block**

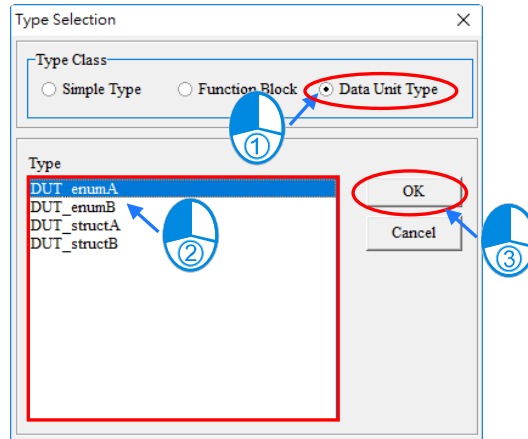
Select the **Function Block** option button in the **Type Class** section, and then select a function block definition in the **Type** box. If the data type of a symbol is a function block, a function block instance is declared. Please refer to chapter 7 for more information.







6

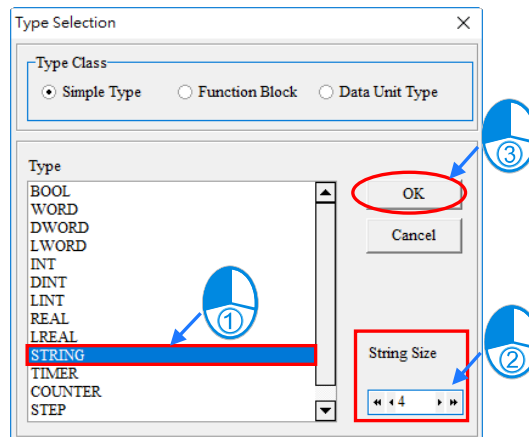
## ● Data Unit Type

Select the **Data Unit Type** option button in the **Type Class** section, and then select a DUT\_enumA in the **Type** box, indicating the symbol is declared here. Please refer to chapter 8 for more information.







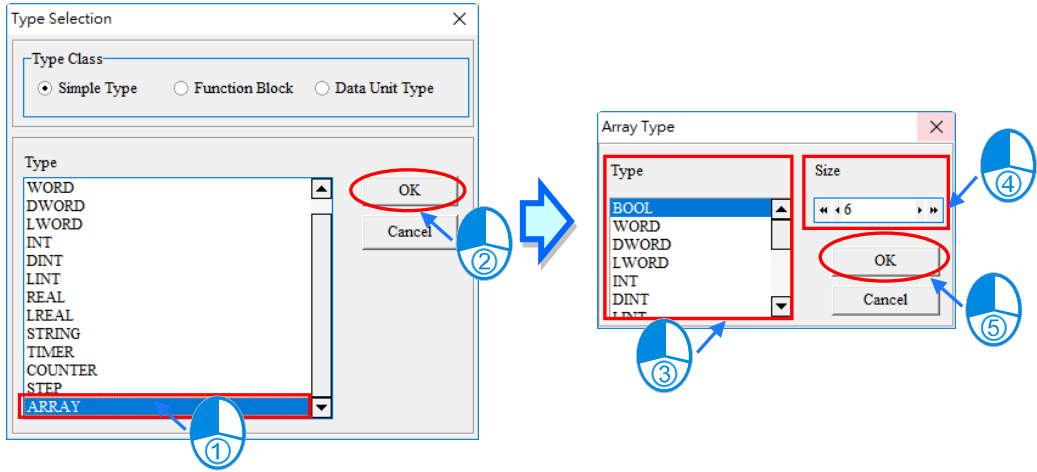
## ● STRING

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.

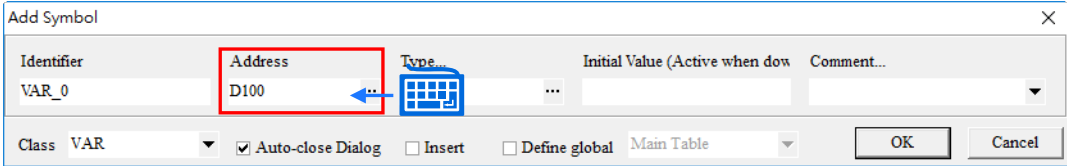


## ● ARRAY

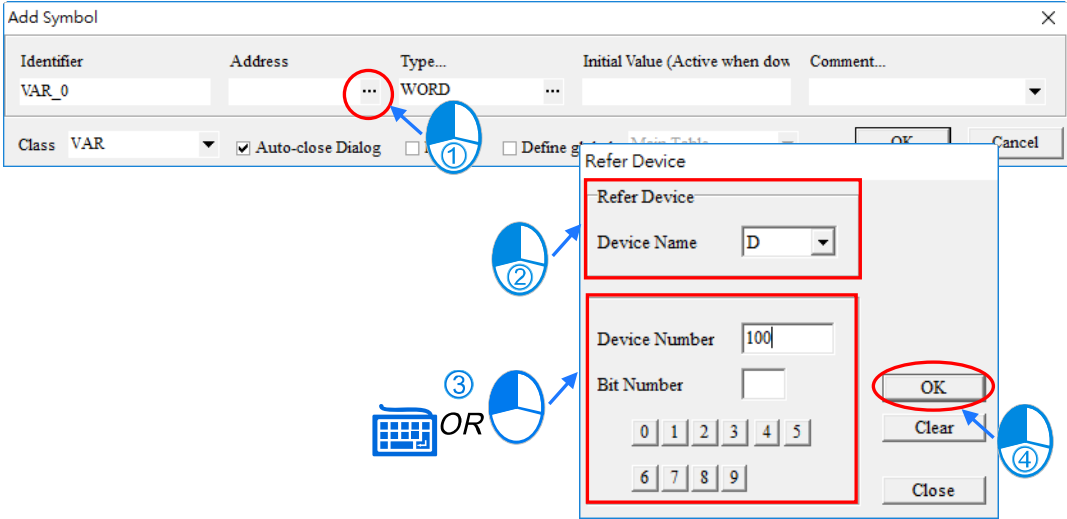
If the data type of a symbol is ARRAY, the **Array Type** window will appear after **OK** in the **Type Selection** window is clicked. Users have to select an array type and set the size of the array in the **Array Type** window. The size of an array is within the range between 1 element and 512 elements. 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.



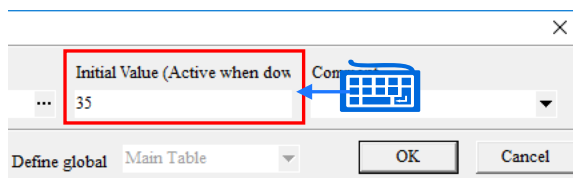
OR



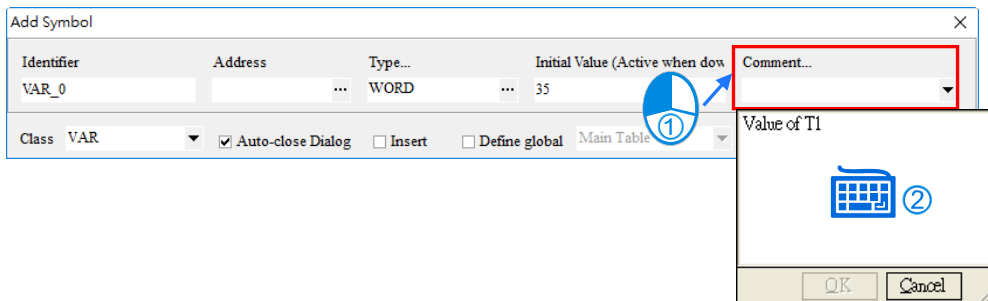
- \*1. When function block is selected for Type, the address column cannot be setup.
- \*2. For bit operation supported model types, the device setting window has Bit Number column to input e.g. X0.0, D0.1.

- (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.

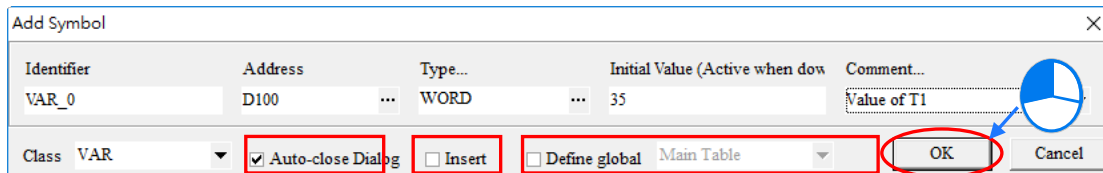
If the data type of the symbol is BOOL, the initial value is either TRUE (abbreviated to T) or FALSE (abbreviated to F). If users type 1 or 0 in the **Initial** box, the system will automatically convert it into TRUE or FALSE. If the data type of the symbol is STRING, the number of characters in the **Initial** box cannot be larger than the maximum number of characters in the string which has been set. The characters cannot be put in double quotes. Otherwise, the double quotes will be regarded as a part of the string. For DVPxxMC, users can add apostrophe to the first and last character in a string. When the data type of the symbol is function block or user-defined, the **Initial** column cannot be setup.



- (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 three 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.



- **Insert**

When selected, the symbol variable declared here will be added on top of the symbol variables for selection in the symbol table; when not selected, it is added to the last row in the symbol table.

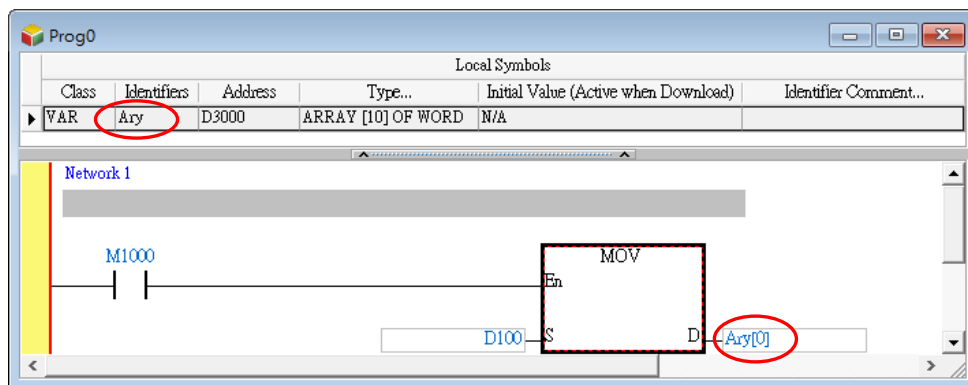
- **Declared in Global Symbol Table**

If users select the **Define Global** checkbox, you can click the drop-down list to find the Main Table and self-defined Global Symbol Table with the declared symbols in them. However, if you are on the Main Table and self-defined Global Symbol Table, this option will not appear.

### 6.2.3 Principles of ARRAY or STRING Symbol

- **Symbol Variable - ARRAY**

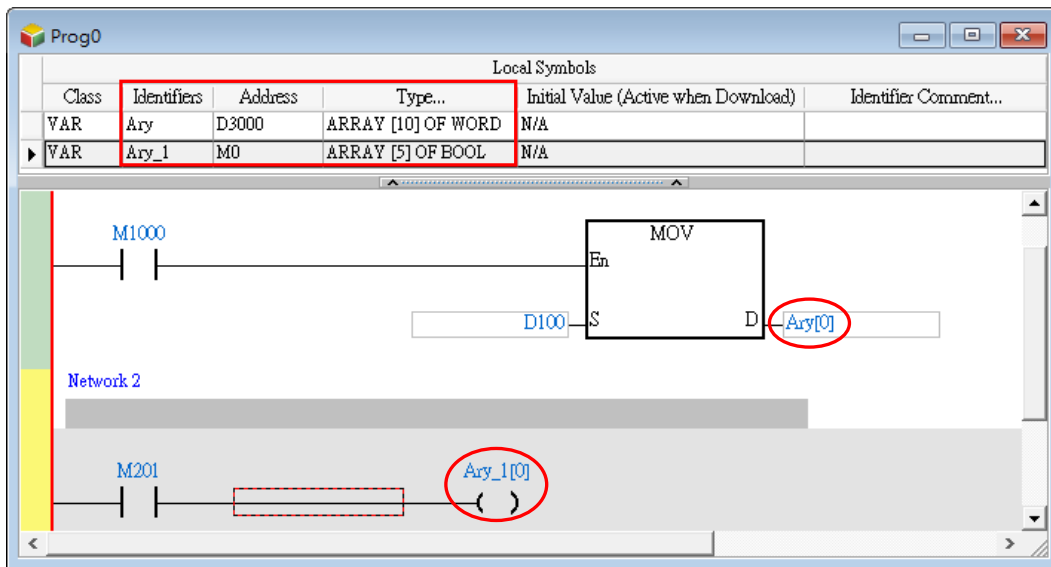
If users want to use a symbol whose data type is ARRAY in a program, the expression format is **Identifier[Index]**. Generally speaking, the index is a decimal constant. If the programming language in a project for an AH/AS series CPU module is a structured text, the index can be a symbol. The minimum index value must be 0, whether the index is a constant or a symbol. The maximum index value cannot 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 cannot 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 512. 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 cannot 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. For DVPxxMC, if **ARRAY[10] OF BOOL** in the Type, the assigned address is %IB0, this means the array will occupy %IB0 ~ %IB9 device, a total of 10 IB devices.



6

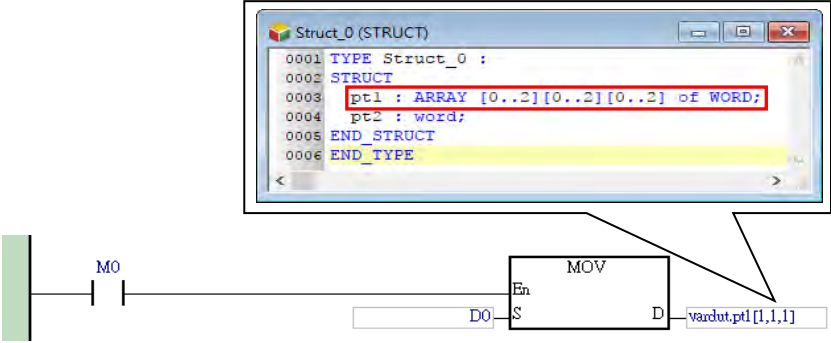
When users declare a symbol whose data type is ARRAY, they can set the initial value of the symbol. Please refer to the following example.

**[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

When the element is with a two-dimensional array or three-dimensional array in the data unit type, the symbol can use [n,n] or [n,n,n]. As the image shown below, the element pt1 is with a three-dimensional array, the data unit type of the variable can be vardut.pt1[1,1,1] to assign the element [1,1,1] from the array.



● **Symbol Variable - STRING**

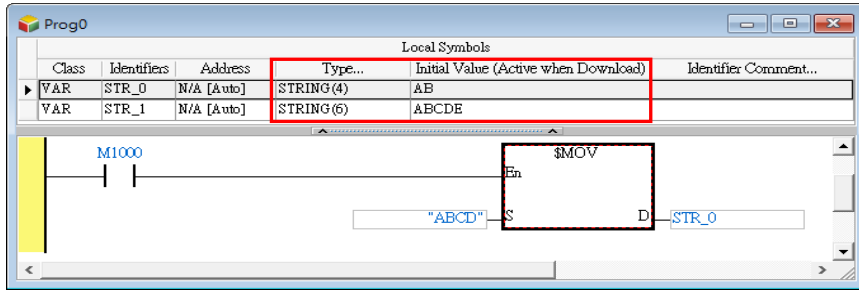
In ISPSOft, a string is composed of ASCII codes, and the ASCII codes are surrounded by double apostrophe, e.g. "ABCD", while for DVPxxMC series, the single apostrophe is used, e.g. 'ABCD'. A symbol whose data type is STRING is generally used with an applied instruction. Please refer to the programming manual for a model for more information.

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.

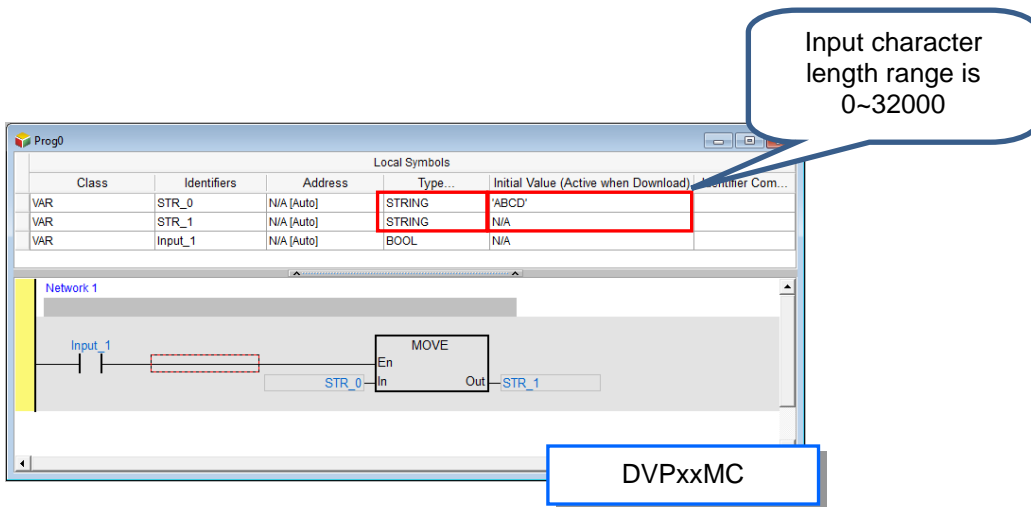
For example, two word devices (four bytes) will be assigned to a string which is composed of three characters, and three word devices (six bytes) will be assigned to a string which is composed of four or five characters.

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. The characters in the **Initial Value** cell for a symbol whose data type is STRING cannot 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.





When using DVPxxMC, users do not need to adjust because the string length declared is allocated by software. For initial value assigned, the character length of an initial value is 0~32000 characters. After declaring, the maximum length of characters is not shown in the brackets beside STRING type.



6

### 6.2.4 Modify Symbol Variables and Edit 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.

Local Symbols					
Class	Identifiers	Address	Type...	Initial Value (Active when Download)	Identifier Comment...
VAR	VAR_0	N/A [Auto]	BOOL	TRUE	
VAR	VAR_1	N/A [Auto]	WORD	0	



Modify Symbol

Identifier	Address	Type...	Initial Value (Active when dow	Comment...
VAR_0	...	BOOL	TRUE	

Class: VAR  Auto-close Dialog  Define global Main Table

OK Cancel

When editing an item in the symbol table, users can click **Edit** in the toolbar, or right-click on the symbol table to operate from the quick menu. Please refer to the following table for basic operations.

Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	VAR_0	N/A [Auto]			
VAR	VAR_1	N/A [Auto]			

OR

Edit	View	Compile	PLC
Undo		Ctrl+Z	
Redo		Ctrl+Y	
Cut		Ctrl+X	
Copy		Ctrl+C	
Paste		Ctrl+V	
Delete		Del	
Select All		Ctrl+A	
Find...		Ctrl+F	
Replace...(H)		Ctrl+H	
Bookmarks			

Item	Function
<b>Undo</b>	Undoing the last action (*The number of previous actions that can be undone is 20.)
<b>Redo</b>	Redoing an action which has been undone
<b>Cut</b>	Cutting the symbol selected
<b>Copy</b>	Copying the symbol selected
<b>Paste</b>	Pasting an object which has been copied or cut on the present symbol table
<b>Delete</b>	Deleting the symbol selected
<b>Select All</b>	Selecting all the symbols in the symbol table

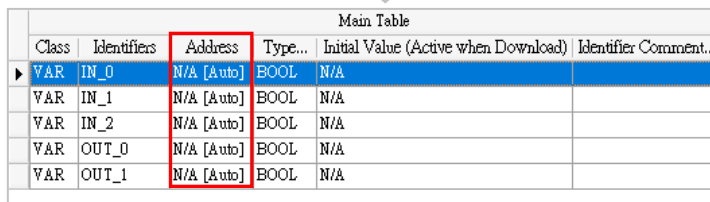
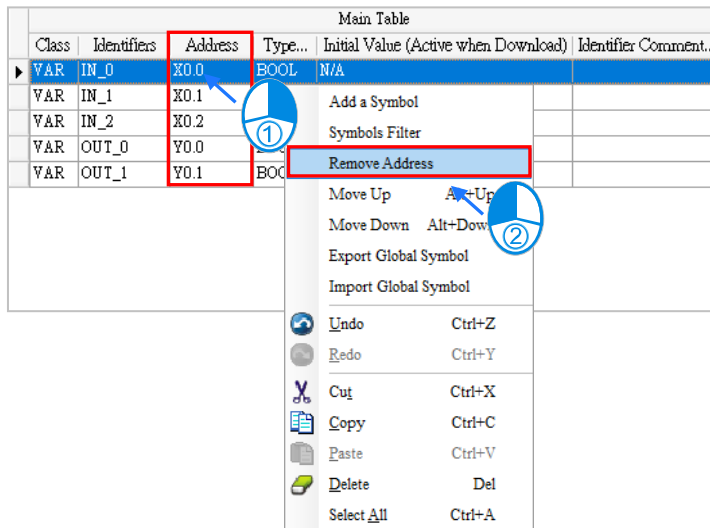
\*1. To copy/cut/paste symbol variables in different PLC types or POU's, please check the pasted symbol variable attributes are correct.

- \*2. The AH/AS series, DVP series and DVPxxMC series, users cannot copy/ paste/ cut the symbol variables amongst the three types .
- \*3. When the pasted symbol table contains the same symbol name, the system will automatically add CopyOf\_n (n is a serial number) after the pasted symbol.

### 6.2.5 Remove Symbol Variable Allocated Address

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.

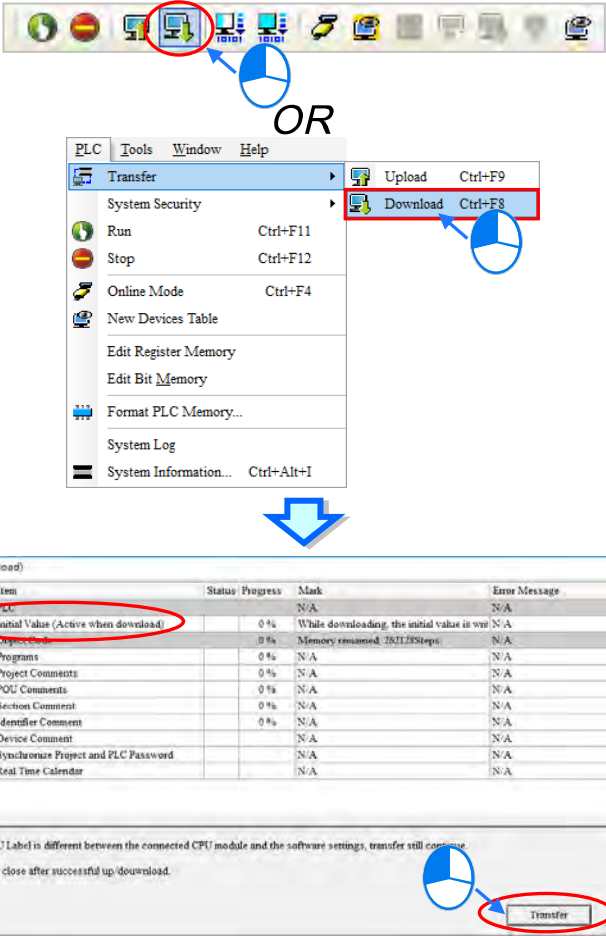


6

### 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. When executing online updates, the system does not download the initial values of the symbol variables; DVPxxMC does not have this option, therefore, when downloading the project, users need to setup parameters and device initial value in HWCONFIG.

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.

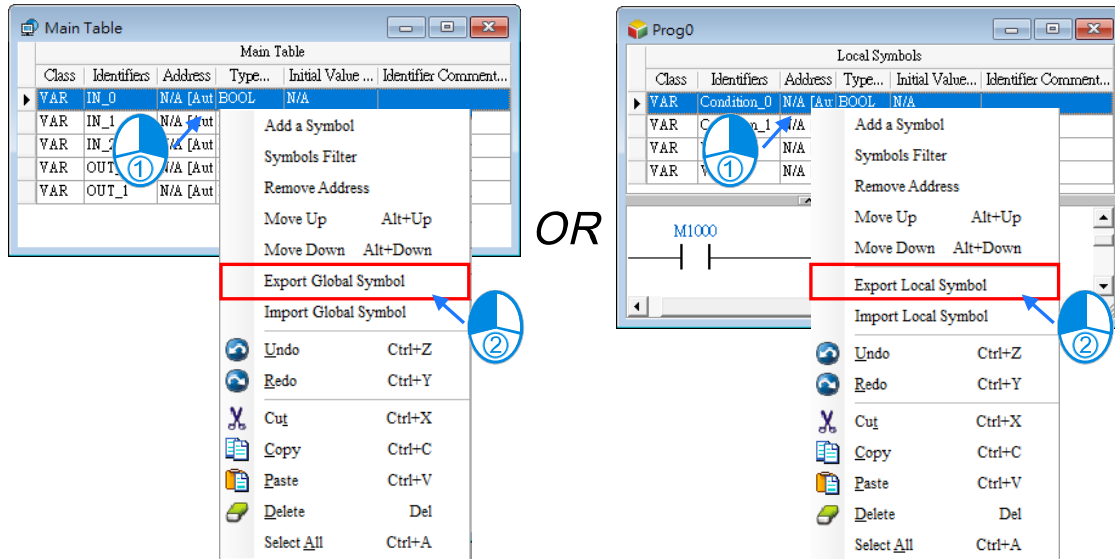


\*. Before the initial values are downloaded, users have to make sure that ISPSOft is connected to the PLC normally.

## 6.2.7 Export 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.



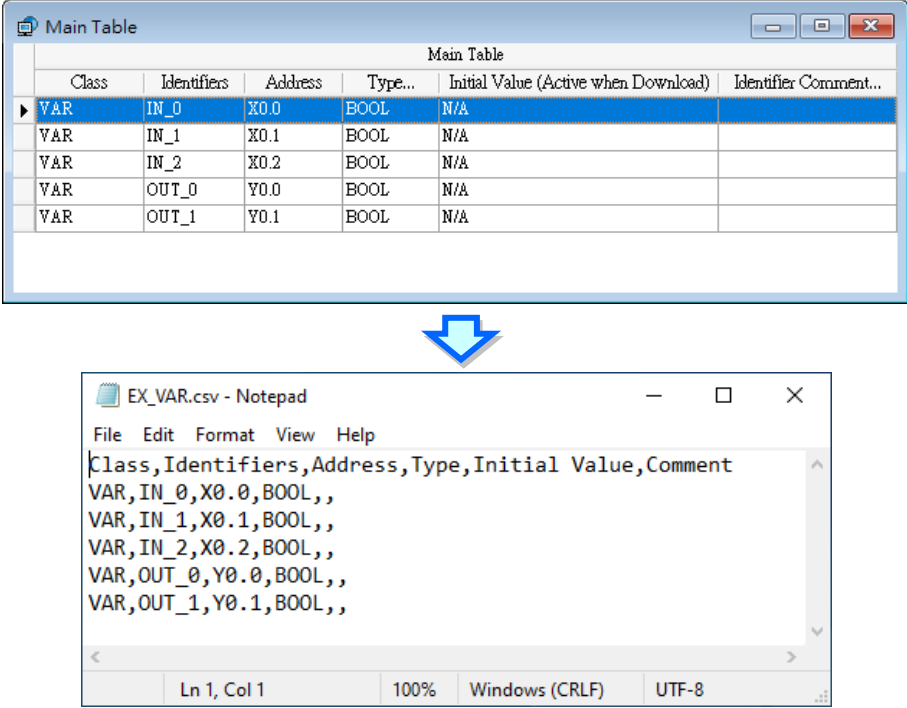
The users can also export the global symbol table in a project through the toolbar **File -> Export -> Global Symbols** or right-click the global symbol table and select **Export Global Symbol** from the context menu to export the global symbols which is being edited presently. If there is a group of projects in ISPSOft, the users can only export the global symbol table in the project which is being edited presently.

6



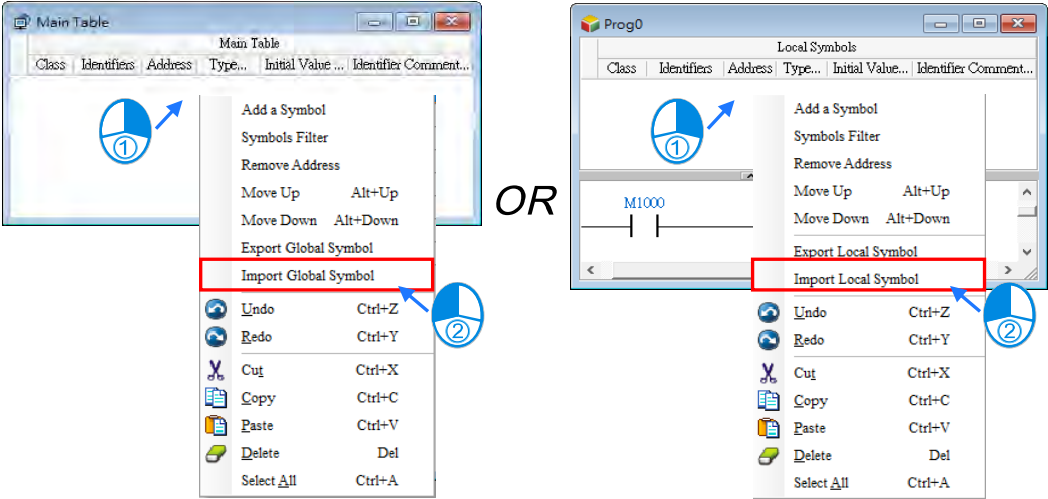
### 6.2.8 Importing a Symbol Table

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.



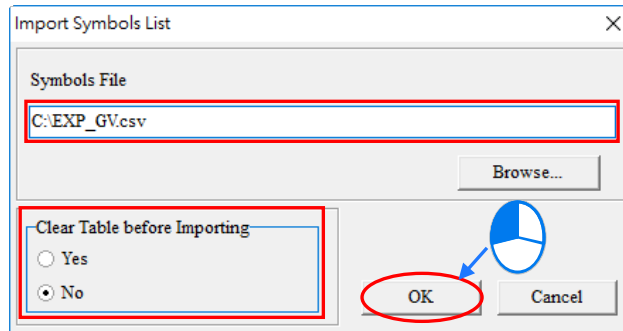
If users want to import a symbol table, they can right-click a symbol table, and click **Import Global Symbol** or **Import Local Symbol** on the context menu.

6



The users can also import the global symbol table in a project through the toolbar **File -> Export -> Global Symbols** or right-click the global symbol table and select **Import Global Symbol** from the context menu to import the global symbols which is being edited presently. 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.

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**.



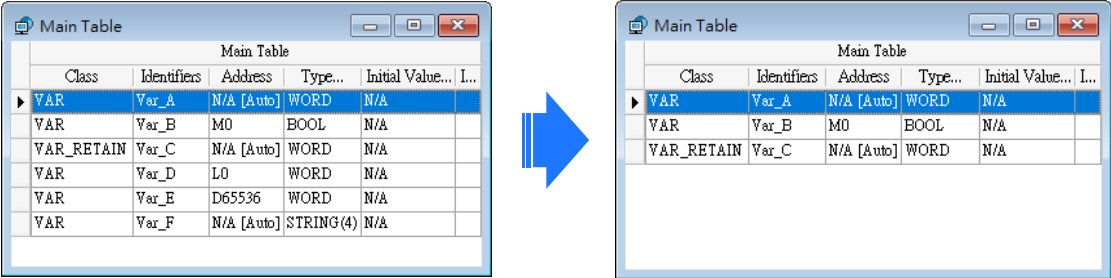
### **Additional remark**

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. Currently, the DVPxxMC series do not support export and import symbol variables with other model types. 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 AH/AS 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.

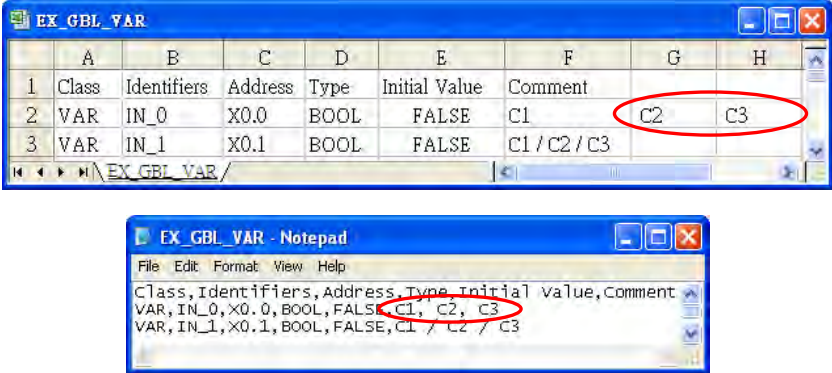
An example of importing the global symbol in a project for AHCPU530-EN into a project for a DVP-SV series PLC is shown below. Var\_A, Var\_B, and Var\_C in the project for AHCPU530-EN can be imported into the project for a DVP-SV series PLC. Var\_C in the project for AHCPU530-EN is a symbol of the VAR\_RETAIN class. After Var\_C in the project for AHCPU530-EN is imported into the project for the DVP-SV series PLC, Var\_C becomes a symbol of the VAR class. Owing to the fact L0

is not supported by the DVP-SV series PLC, Var\_D is eliminated. D65535 is not within the range of devices in the DVP-SV series PLC, and therefore Var\_E is eliminated. Var\_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.



6

● Before the symbol table is exported

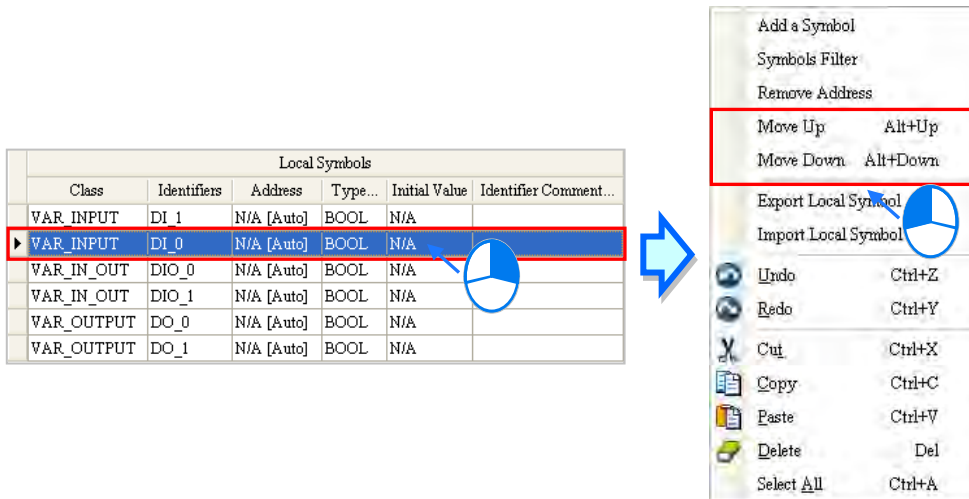
Global Symbols						
Class	Identifiers	Addr...	Type...	Initial Value	Identifier Comment...	
VAR	IN_0	X0.0	BOOL	FALSE	C1, C2, C3	
VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3	

● After the symbol table is imported

Global Symbols						
Class	Identifiers	Addr...	Type...	Initial Value	Identifier Comment...	
VAR	IN_0	X0.0	BOOL	FALSE	C1	
VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3	

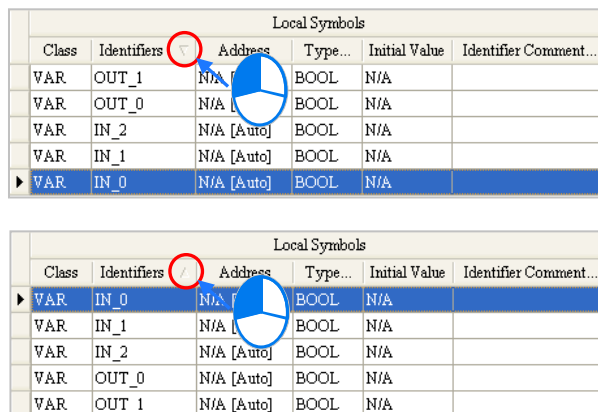
### 6.2.9 Arranging the symbols

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 right-click a symbol which will be moved upward or moved downward, and press Alt + ↑ or Alt + ↓ on the keyboard.



In global symbol table and local symbol table of POU, users can click the **Identifier** or **Address** 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 cannot be sorted in this way.

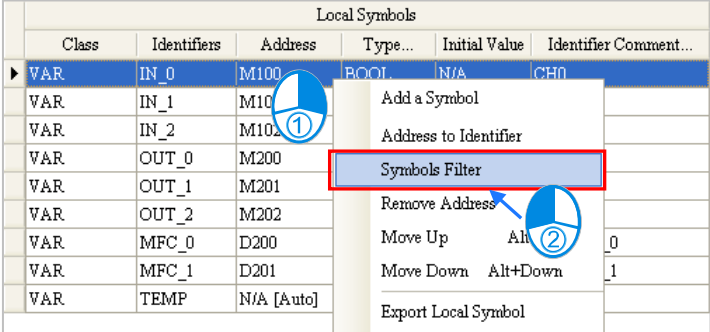
6



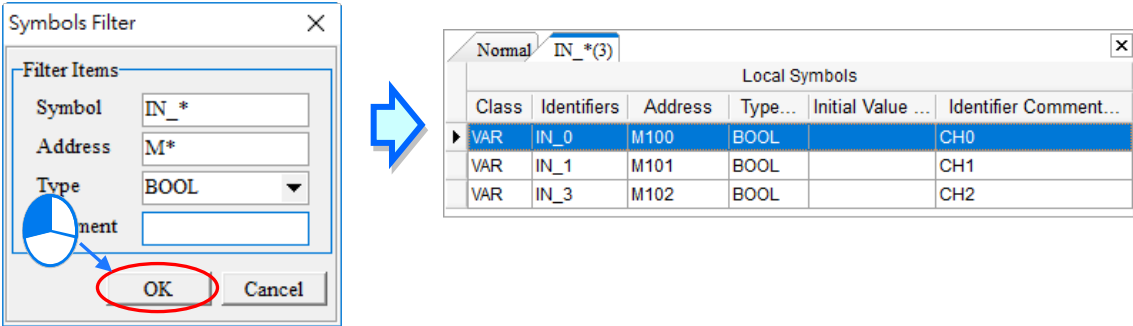
### 6.2.10 Filtering the Symbols

Users can view the symbols which meet the same conditions.

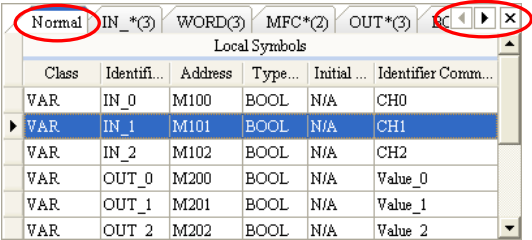
- (1) Right-click a symbol table, and click **Symbols Filter** on the context menu.



- (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.



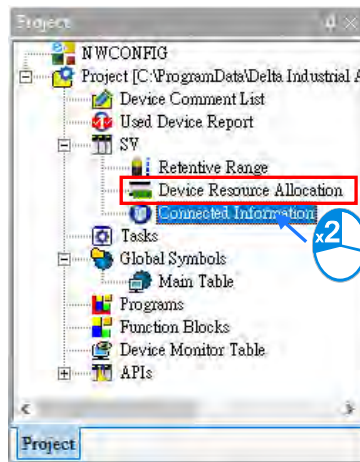
- (3) The symbols which are sieved out are displayed in the additional page. The users can switch between the pages by clicking the tabs at the top of the table. If several groups of symbols are sieved out, the users can switch among the tabs by clicking ◀ or ▶ in the upper right corner of the window. 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 cannot edit the tables in the pages. After the users click the **Normal** tab, they can edit the original symbol table.



## 6.2.11 Device Resource Allocation Setting

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.

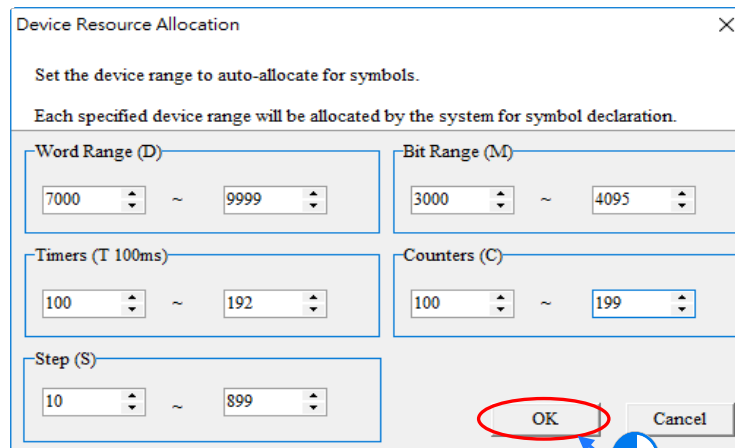
Unfold the PLC section in the project management area in a project for a DVP series PLC, double-click **Device Resource Allocation**.



6

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 reorganizing the structure of the POU and TASK. 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
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.

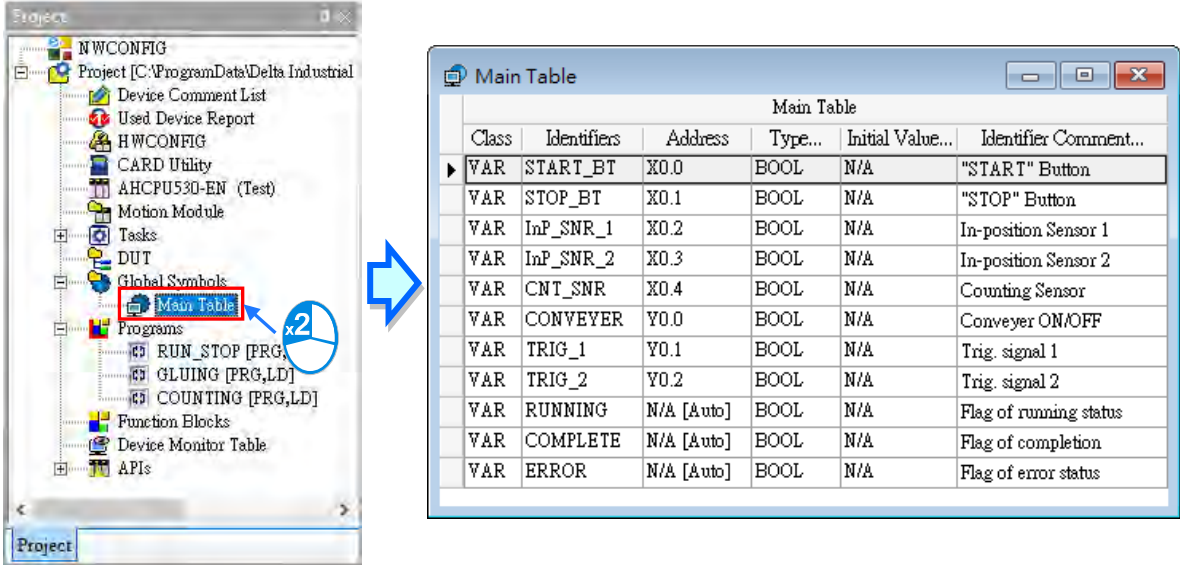
# 6



### 6.3.2 Writing a Program

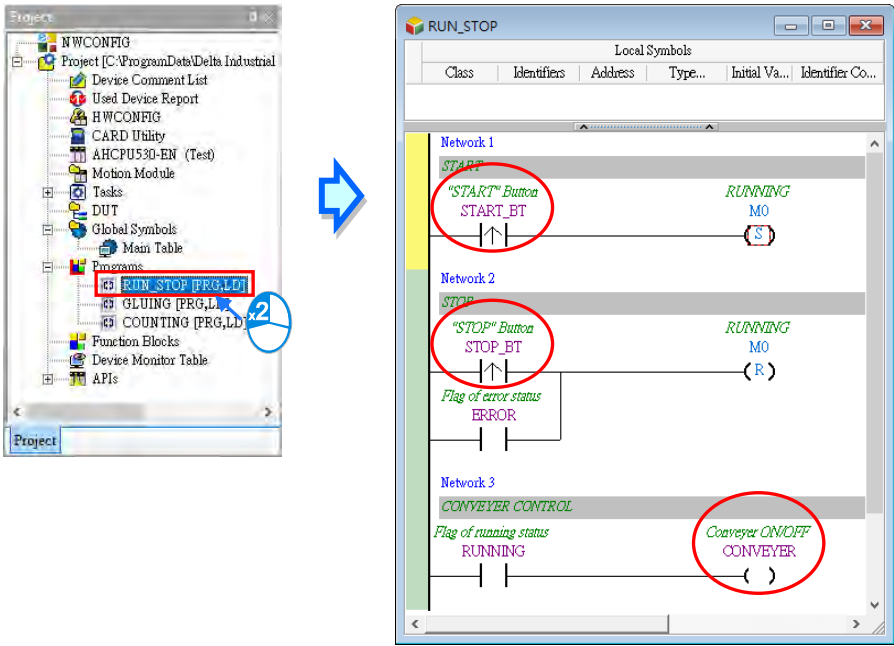
Users can directly modify the program created in section 5.6.1, or write the program shown in section 5.6.1 again. If the users want to write the program again, they have to create a new project, and refer to chapter 4 and section 5.6.1 for more information.



First, open the **Main Table** window. Then, refer to section 6.2, and create the symbols shown below in the global symbol table.

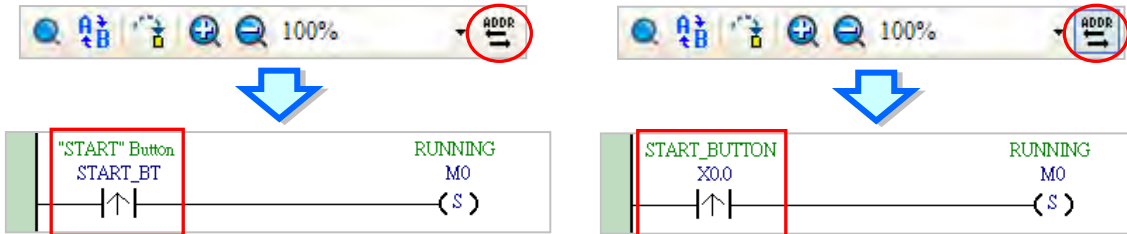




Double-click **RUN\_STOP** in the project management area after the creation of the symbols in the global symbol table is complete. If users directly modify the program created in section 5.6.1, they will find that the symbols to which they assign the device addresses replace the corresponding device 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.


6

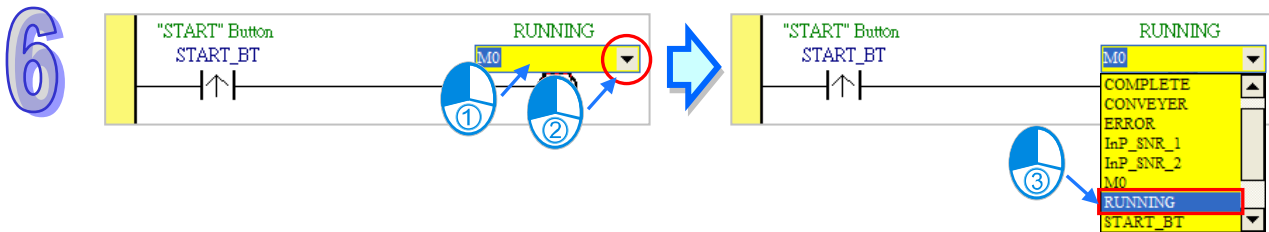


The users can choose whether to display the symbols to which they assign devices in the ladder diagram. If  on the toolbar is pressed, the device names and the comments on the devices will be displayed in the program. If  is not pressed, the identifiers and the comments on the symbols will be displayed in the program.

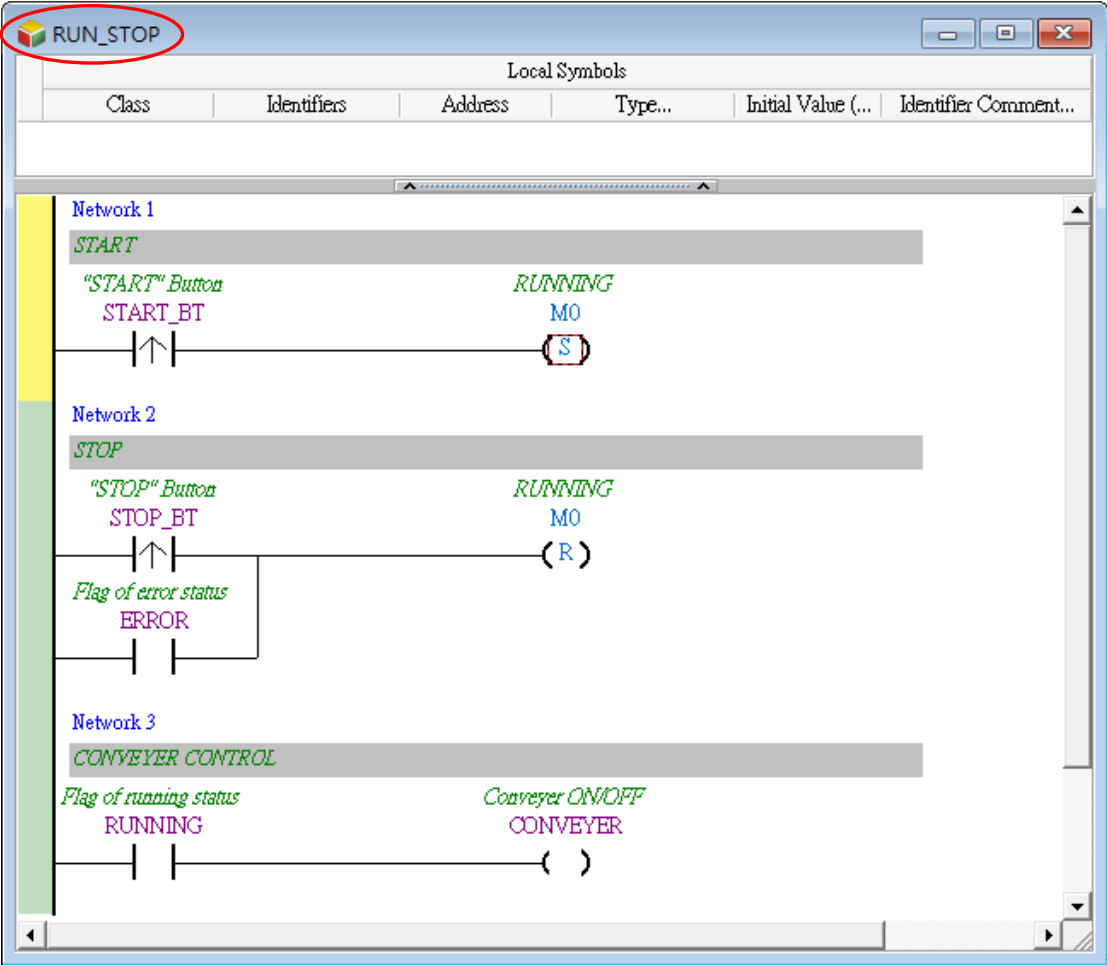


\*. M0 is not assigned to any symbol, and therefore it remains unchanged if  is not pressed. Likewise, if users do not assign a device address to a symbol, or the system automatically assigns an internal memory to a symbol, the symbol will remain unchanged after  is pressed.

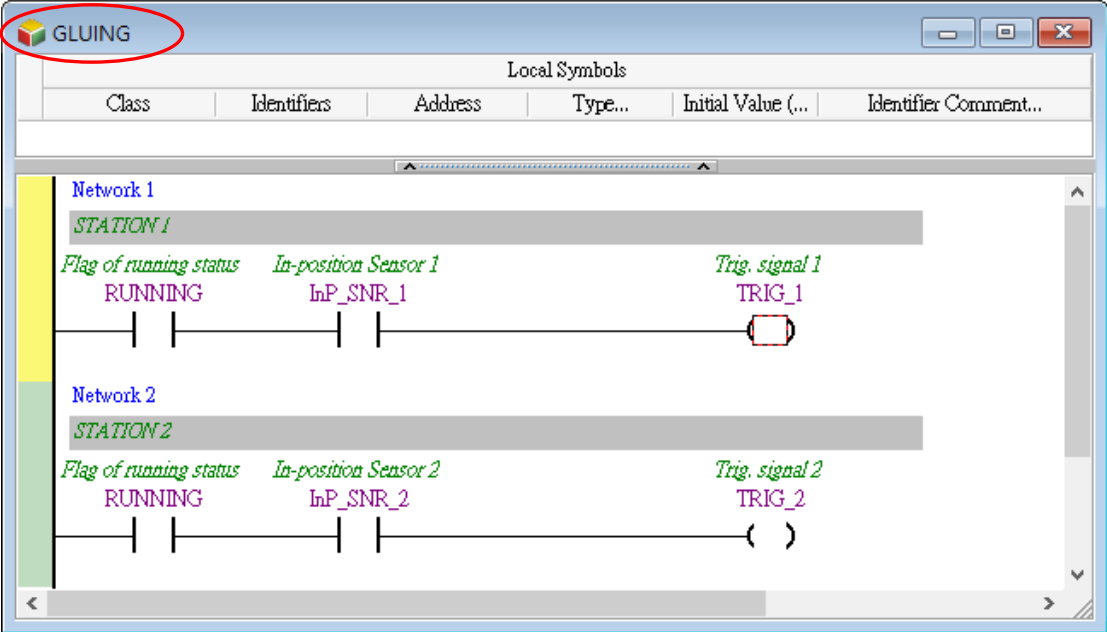
The operation flag M0 can be replaced by the identifier RUNNING. The users can refer to chapter 4, and type RUNNING directly. They can also click M0, click  in the box, and select an identifier on the drop-down list. Besides, the users can press Page Down on the keyboard when they edit the box. The users can select an identifier on the drop-down list by  $\uparrow$  or  $\downarrow$  on the keyboard. Press Enter on the keyboard after the selection is complete.

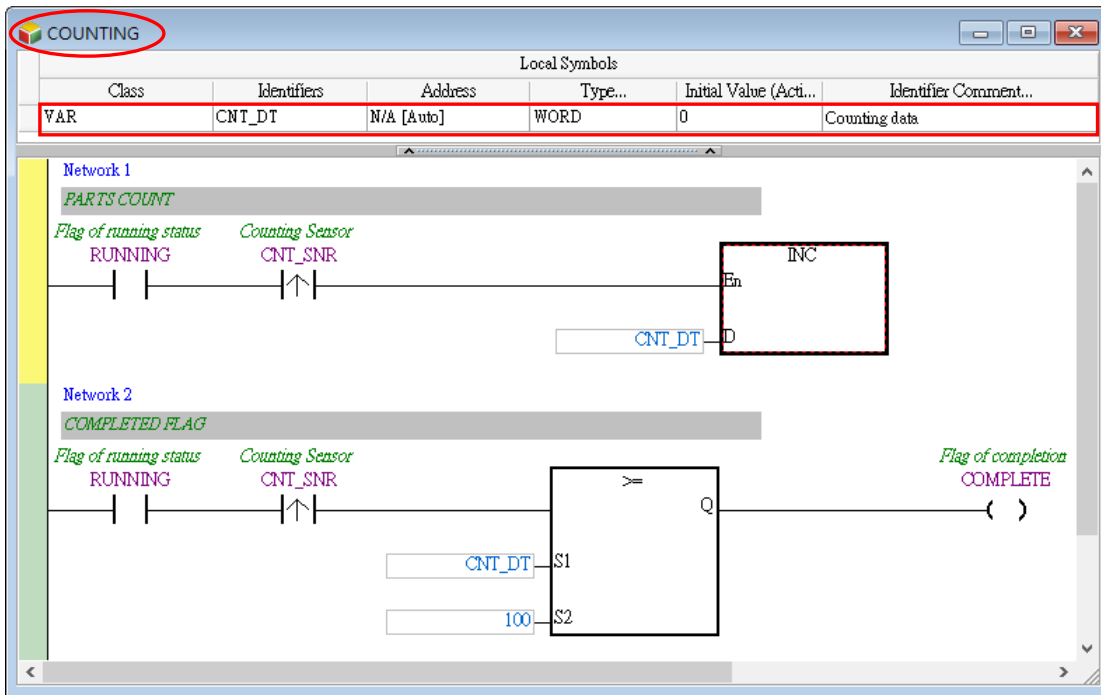


Refer to the description above, and write the program in the POU shown below.



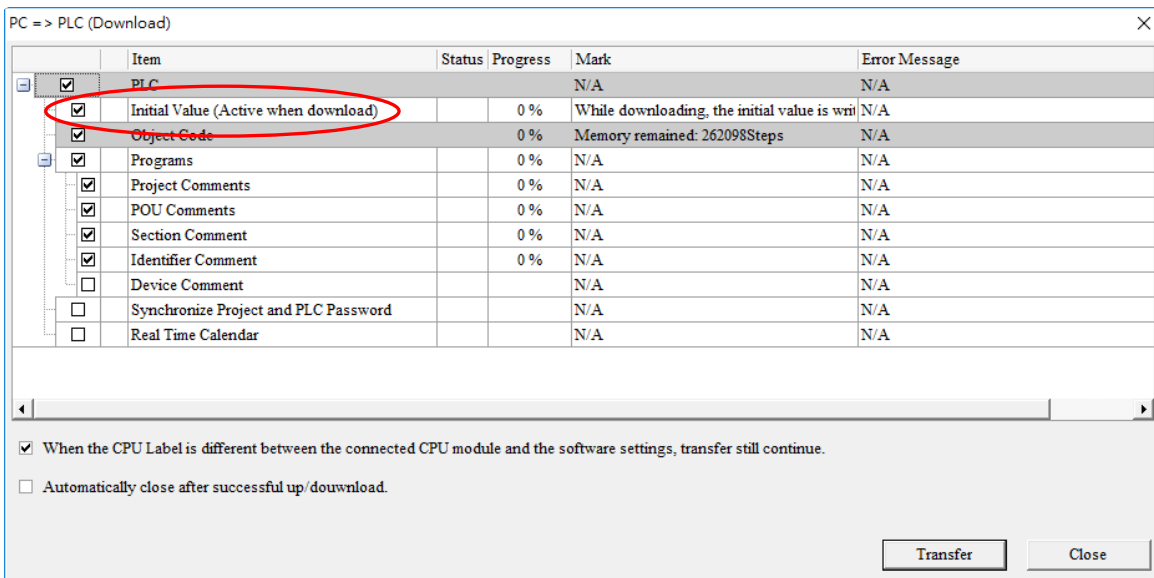
6





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.

# 6



# Chapter 7 Function Block and Library



## Table of Contents

<b>7.1</b>	<b>Knowing Function Blocks.....</b>	<b>7-2</b>
7.1.1	Introduction of Function Blocks.....	7-2
7.1.2	Characteristics and Advantages of Function Blocks .....	7-3
<b>7.2</b>	<b>Structure of the Function Blocks in ISPSoft.....</b>	<b>7-5</b>
7.2.1	En Pin of Function Block.....	7-5
7.2.2	Symbol Variables in Function Block.....	7-6
7.2.3	Input/Output Pins of a Function Block.....	7-8
7.2.4	Index Type Symbol Variables .....	7-14
7.2.5	Function Block Definition and Function Block Instance .....	7-19
7.2.6	Calling Relation Between Function Blocks .....	7-22
7.2.7	Memory Configuration of Function Block .....	7-25
<b>7.3</b>	<b>Using a Function Block .....</b>	<b>7-33</b>
7.3.1	Basic Specifications for Function Blocks .....	7-33
7.3.2	Pulse Instructions for Function Blocks (AH/AS Series ONLY) .....	7-35
7.3.3	Monitoring the Program in a Function Block.....	7-36
7.3.4	Modifying the Program in a Function Block .....	7-37
<b>7.4</b>	<b>Instance .....</b>	<b>7-38</b>
7.4.1	Planning a Program .....	7-38
7.4.2	Creating the Program .....	7-39
<b>7.5</b>	<b>Knowing the Library .....</b>	<b>7-49</b>
7.5.1	Creating the User-defined Library .....	7-49
7.5.2	Including the Function Blocks in the User Defined Library.....	7-51
7.5.3	Using Delta Library.....	7-53

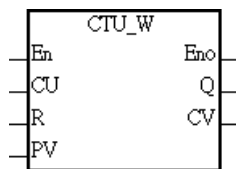
## 7.1 Knowing Function Blocks

Function blocks (FB) and Function (FC) play important roles in the PLC programming. They are also supported by IEC 61131-3. Owing to the characteristics and advantages of function blocks, ISPSOft also supports and offer many functionalities related to function blocks, for instance user-defined FB POU and system-defined FB POU. Though self-defined FC POU is not supported, ISPSOft provides API instructions for users to use as system-defined FC POU.

### 7.1.1 Introduction of Function Blocks

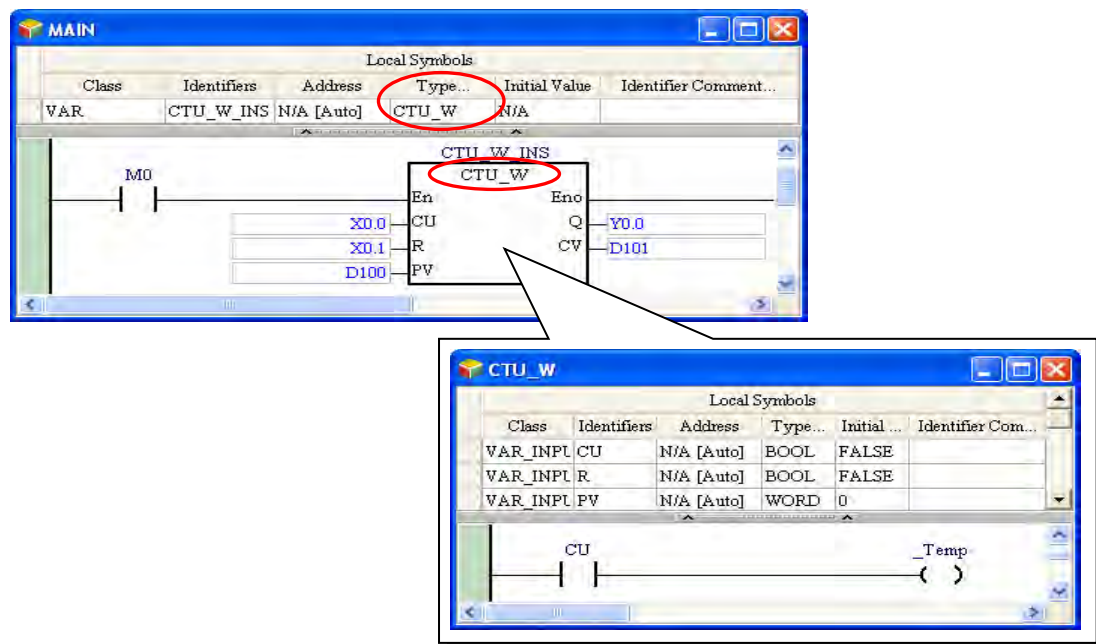
A function block is a component in a program which performs an operation. It is a type of POU, but it can not operate by itself. After a POU of the program type calls a function block, and sends the related parameters, the function of the function block can be executed. After the execution of the function 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.

**7**

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 instance 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 POU's 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**

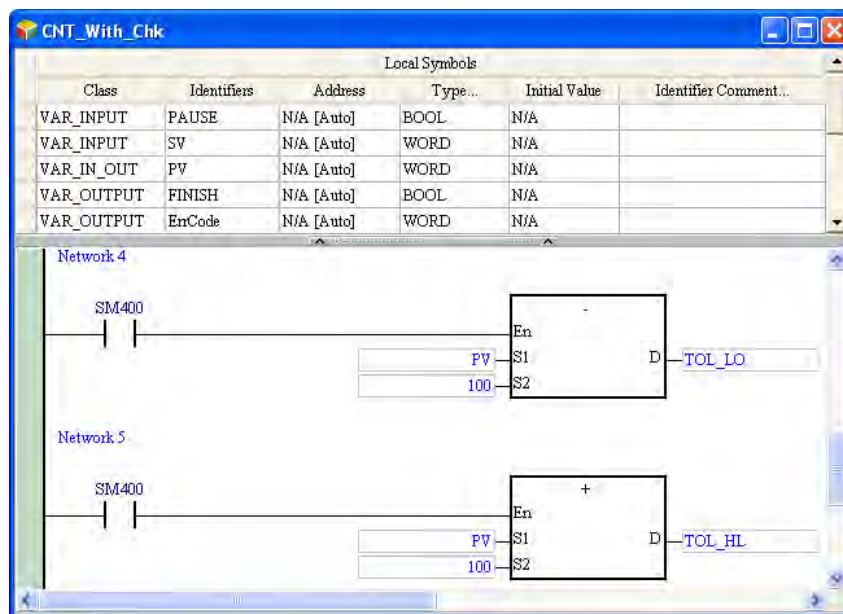
The application of function blocks provides a highly efficient development environment in terms of project development. Owing to the fact that function block are modular, developers and manufacturers can take part in the development of a project together, and the use of human resources are more flexible.





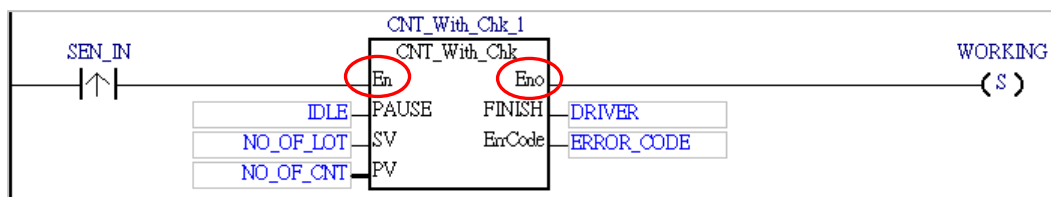
## 7.2 Structure of the Function Blocks in ISPSOft

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 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 Symbol Variables in Function Block

Function blocks are similar to POU's 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 be portable. If the function block is imported into another project, the project may not have the same global symbols.

The following table describes the symbol variables related to function blocks and different categories of symbol variables supported by each PLC type.

- The category of symbol variables for function block:

Category	Description	AH/AS	DVP	DVPxxMC
VAR	Only for function block arithmetic purpose internally. After executing, the symbol variable status is retained.	✓	✓	✓
VAR_INPUT	Used as input point of function block. The variable receives the assigned operand from the upper level POU (caller). When executing the function block, the operand value is copied into the function block.	✓	✓	✓
VAR_OUTPUT	Used as output point of function block. When finish executing the function block, the variable outputs the arithmetic result to the assigned operand by the upper level POU (caller).	✓	✓	✓
VAR_IN_OUT	Feedback points of a function block. When executing function blocks, use the point to send in the assigned operand content by the upper level POU (caller) to the function block. When finish executing the function block, arithmetic result feedback is output to the assigned operand by the upper level POU (caller). (DVPxxMC does NOT support VAR_IN_OUT)	✓	✓	
VAR_RETAIN	The retain variables can retain their value after power failure and the variable can be defined up to 15999 in total.			✓

7

- Data types supported by internal function block local symbols :

Model Types	Description
AH/AS	BOOL, WORD, DWORD, LWORD(*2), INT, DINT, LINT(*2), REAL, LREAL(*2), ARRAY, STRING, POINTER(*1), T_POINTER(*1), C_POINTER(*1), HC_POINTER(*1), function block(FB) types, user-defined types
DVP	BOOL, WORD, DWORD, , REAL, ARRAY, STEP, TIMER, and COUNTER
DVPxxMC	BOOL, BYTE, WORD, DWORD,LWORD, SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL, ARRAY, STRING, TIME, DATE, TOD,DT, function block (FB) types, user-defined types

\*1. Please refer to section 7.2.4 for more information about POINTER, T\_POINTER, C\_POINTER, and HC\_POINTER.

\*2. The LWORD, LINT and LREAL data types are not supported by the function blocks in AS series CPU modules.

- Memory configuration of a function block

Model Types	Description
AH/AS	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.
DVPxxMC	When declaring symbol variables as function block types (FB instance)as well as local symbols created in function blocks, both cannot assign device address by themselves, the system will auto-configure a reserved memory section in the host.

\*. Please refer to sections 7.2.5-7.2.7 for more information about function block instances and the assignment of memories.

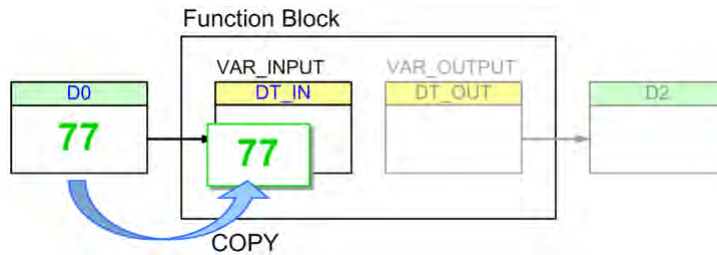
### 7.2.3 Input/Output Pins of a Function Block

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.

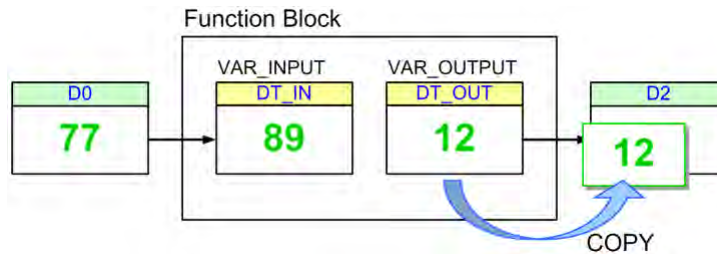
● **Before a function block is executed**

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.)



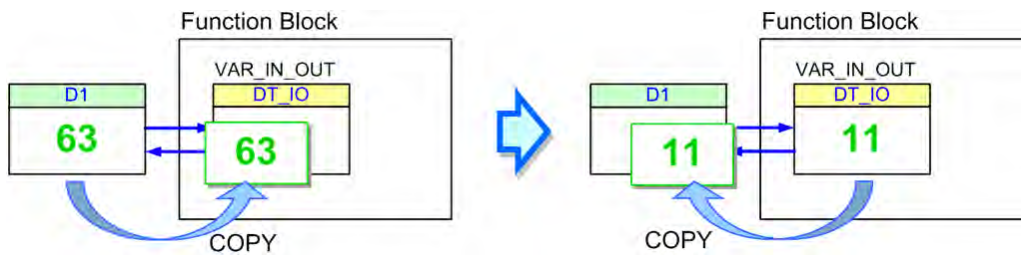
● **After a function block is executed**

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.)



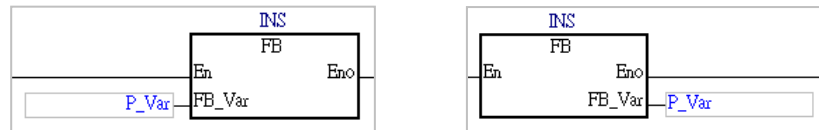
7

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 AH/AS 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. The basic principles below is supported by DVPxxMC.



● **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.

Model Types		AS/AH/DVP								
		FB_Var (function block)								
Data type		WORD	DWORD	LWORD	INT	DINT	LINT	REAL	LREAL	BOOL
P_VAR (caller)	WORD	✓			✓					
	DWORD	✓	✓		✓	✓		✓		
	LWORD	✓	✓	✓	✓	✓	✓	✓	✓	
	INT	✓			✓					
	DINT	✓	✓			✓				
	LINT	✓	✓	✓			✓			
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 AARRAY as a symbol whose data type is ARRAY, and the size of the array for the symbol will be one element.
3. If the array type of P\_Var or the array type of FB\_Var is WORD, INT, REAL or any other types, users can compare both 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 and contains the length of two elements and for a symbol whose array type WORD and has a length of four elements is compatible with a symbol whose array type is LINT. However, a symbol whose array type is INT and has a length of four elements is not compatible with a symbol whose array type is LINT, because both data types do not contain the array type of WORD and do not fit the principle.

The following examples are:

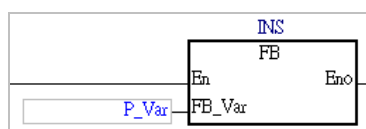
- ✓ Example 1: The data type of P\_Var (in the caller) is INT, and the data type of FB\_Var (in the function block) is INT.  
➔ The basic principle for the general data types is satisfied. As a result, the data types are legal.
- ✓ Example 2: The data type of P\_Var (in the caller) is WORD, and the data 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.

- **Principles of Data Types for DVPxxMC**

- Example 1



1. Generally, the data type of P\_Var (in the caller) and the data type of FB\_Var (in the function block) is the same. But for some situations, the ISPSOft allows the length of the data type of P\_Var (in the caller) to be shorter or equal to the length of FB\_Var (in the function block).

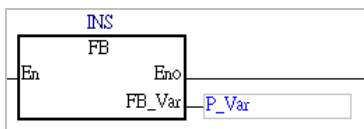
- If the data type of FB\_Var (in the function block) is ARRAY, then the data type of P\_Var (in the caller) has to be ARRAY and the content as well as length has to be the same as FB\_Var (in the function block).

Model Types		DVPxxMC														
Data Type		FB_Var (Function Block)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (Caller)	BOOL	✓														
	BYTE		✓	✓	✓	✓✓	✓	✓	✓	✓✓	✓	✓	✓	✓✓	✓	✓
	WORD			✓	✓	✓✓		✓	✓	✓✓		✓	✓	✓✓	✓	✓
	DWORD				✓	✓✓			✓	✓✓			✓	✓✓	✓	✓
	LWORD					✓			✓				✓		✓	✓
	USINT		✓	✓	✓	✓✓	✓	✓	✓	✓✓	✓	✓	✓	✓✓	✓	✓
	UINT			✓	✓	✓✓		✓	✓	✓✓		✓	✓	✓✓	✓	✓
	UDINT				✓	✓✓			✓	✓✓			✓	✓✓	✓	✓
	ULINT					✓				✓				✓	✓	✓
	SINT		✓	✓	✓	✓✓	✓	✓	✓	✓✓	✓	✓	✓	✓✓	✓	✓
	INT			✓	✓	✓✓		✓	✓	✓✓		✓	✓	✓✓	✓	✓
	DINT				✓	✓✓			✓	✓✓			✓	✓✓	✓	✓
	LINT					✓				✓				✓	✓	✓
	REAL														✓	✓
	LREAL														✓	✓

7

Model Types		DVPxxMC				
Data Type		FB_Var (Function Block)				
		TIME	DATE	TOD	DT	STRING
P_VAR (Caller)	TIME	✓				
	DATE		✓			
	TOD			✓		
	DT				✓	
	STRING					✓

Example below:





1. Generally, the data type of P\_Var (in the caller) and the data type of FB\_Var (in the function block) is the same. But for some situations, the ISPSOft allows the length of the data type of P\_Var (in the caller) to be longer or equal to the length of FB\_Var (in the function block).
2. If the data type of FB\_Var (in the function block) is ARRAY, then the data type of P\_Var (in the caller) has to be ARRAY and the content as well as length has to be the same as FB\_Var (in the function block).

Model Type		DVPxxMC														
Data Type		FB_Var (Function Block)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (Caller)	BOOL	✓														
	BYTE		✓✓				✓✓				✓✓					
	WORD		✓✓	✓			✓✓	✓			✓✓	✓				
	DWORD		✓✓	✓	✓		✓✓	✓	✓		✓✓	✓	✓			
	LWORD		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
	USINT		✓✓				✓✓				✓✓					
	UINT		✓✓	✓			✓✓	✓			✓✓	✓				
	UDINT		✓✓	✓	✓		✓✓	✓	✓		✓✓	✓	✓			
	ULINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	SINT		✓✓				✓✓				✓✓					
	INT		✓✓	✓			✓✓	✓			✓✓	✓				
	DINT		✓✓	✓	✓		✓✓	✓	✓		✓✓	✓	✓			
	LINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	REAL		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	LREAL		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Model Type		DVPxxMC				
Data Type		FB_Var (Function Block)				
		TIME	DATE	TOD	DT	STRING
P_VAR (Caller)	TIME	✓				
	DATE		✓			
	TOD			✓		
	DT				✓	
	STRING					✓

7

**Additional remark**

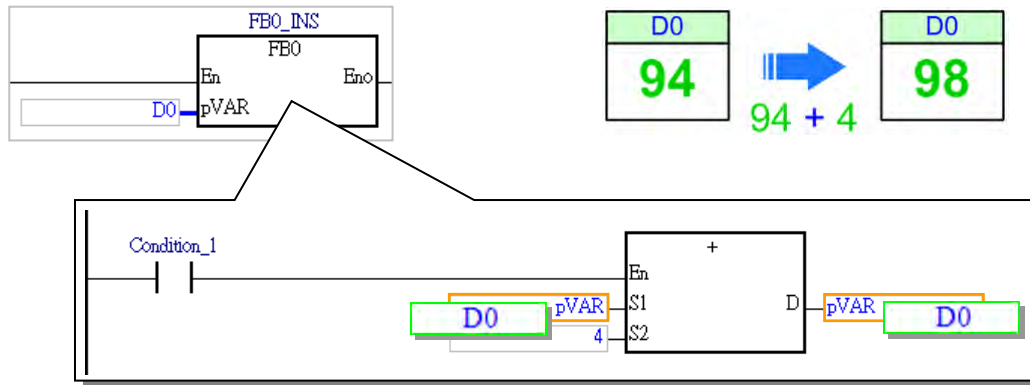
- (a) According to the above principle, the data length of P\_Var (in the caller) is larger than that of FB\_Var (in function block), and the value of P\_Var is sent to FB\_Var, the values of the bits starting from the lowest bit in P\_Var are retrieved, and the number of values retrieved conforms to the data length of FB\_Var. If 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 careful about the correctness of the data.
- (b) The data type of a symbol in a function block in a project for an AH series CPU module can be POINTER. Please refer to section 7.2.4 for more information.
- (c) For DVPxxMC, if the data length of P\_Var (in the caller) is shorter 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 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 highest bits which exceeds the length of FB\_Var is overwritten as 0. Users have to be more careful about the correctness of the data.

## 7.2.4 Index Type Symbol Variables

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 AH/AS 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	8 symbols whose data types are HC_POINTER can be
	Reference	32-bit counter or symbol whose data type is COUNTER and to which a 32-bit counter is assigned

7

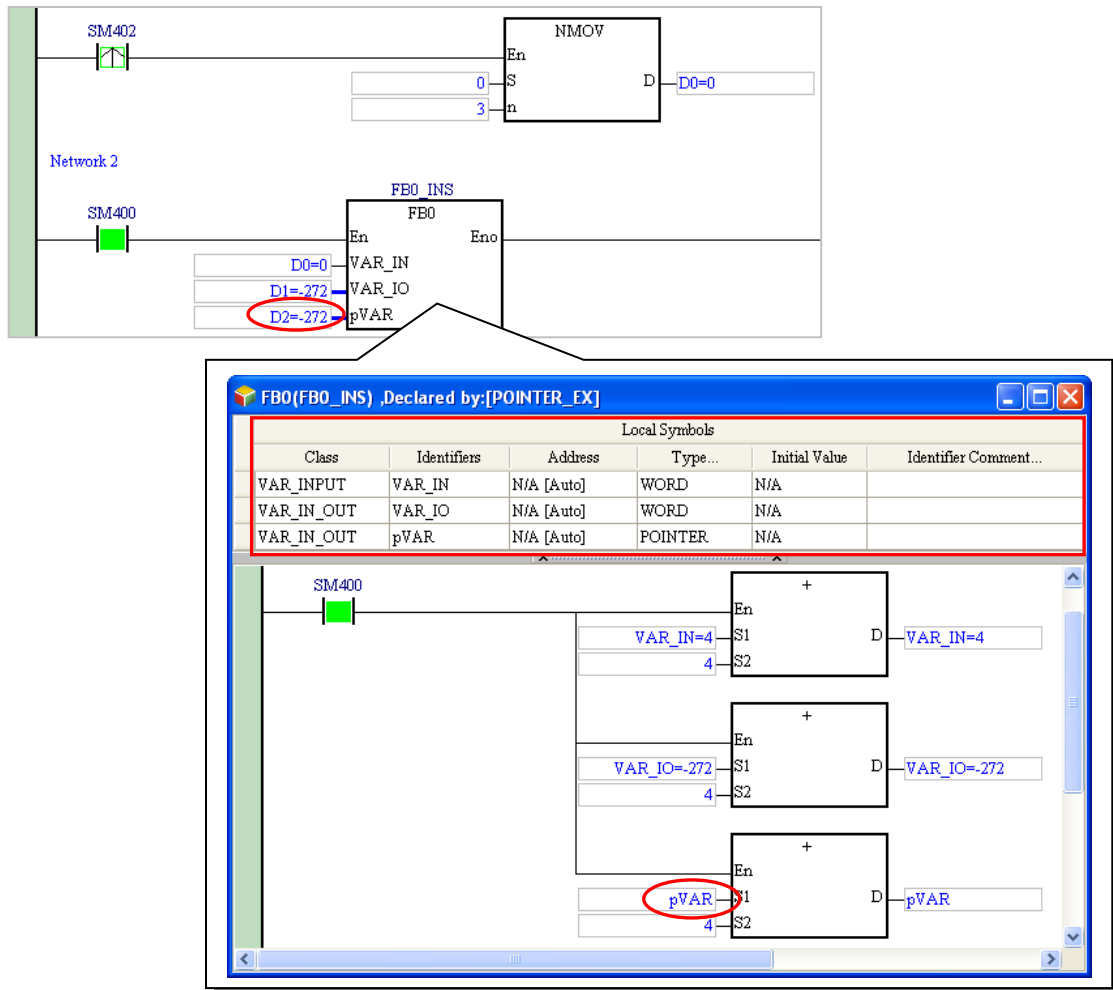
**● Example of using a symbol whose data type is POINTER**

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 VAR\_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 AH/AS series. It is ON when the CPU module is powered. SM402 initializes forward (an instant RUN is ON).

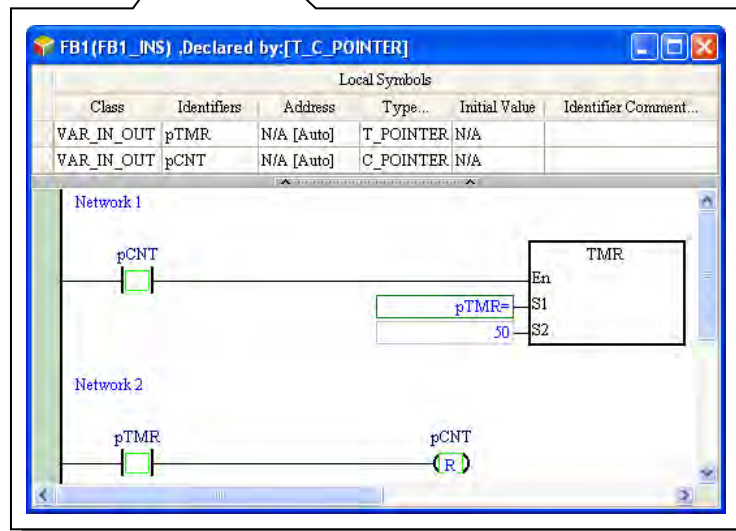
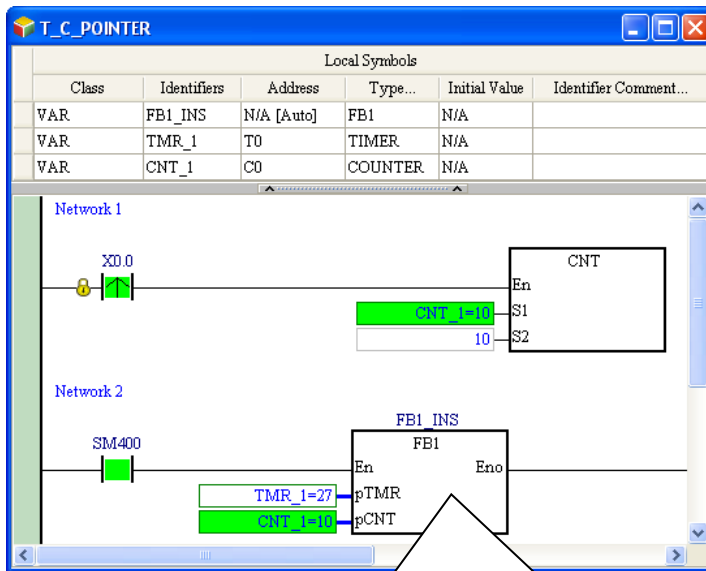
● **Example of using a symbol whose data type is T\_POINTER and a symbol whose data type is C\_POINTER**

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 AH/AS 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.



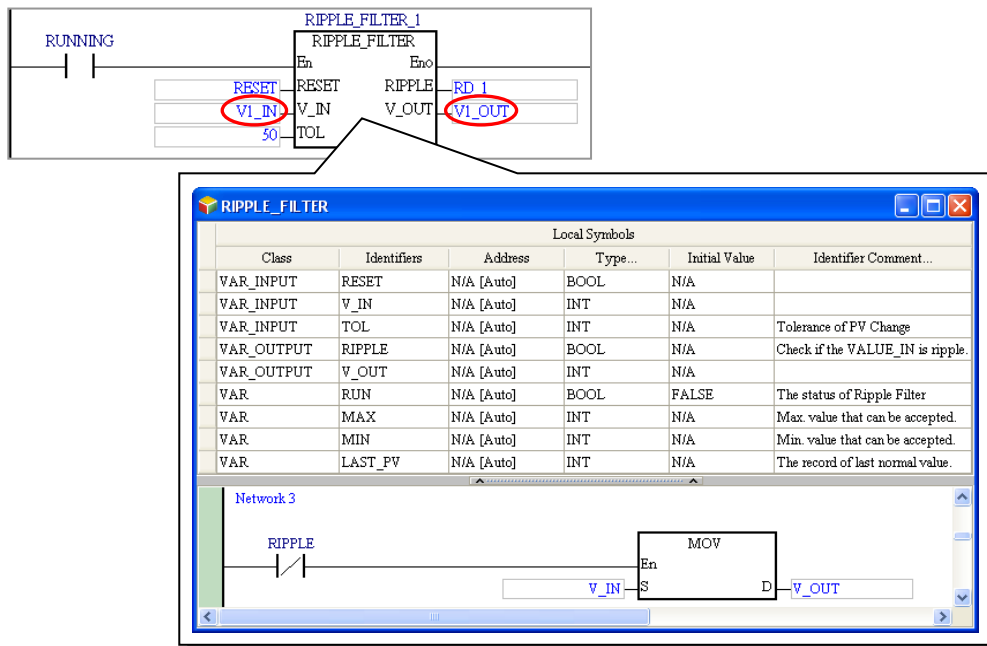
Likewise, the present value of pTMR and that of pCNT are not displayed when the online monitoring function is enabled.



7

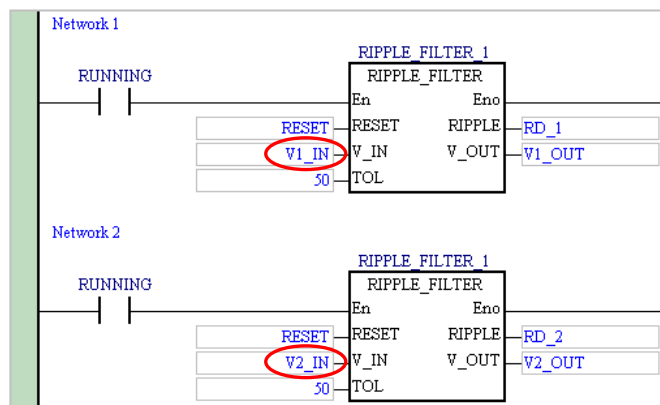
### 7.2.5 Function Block Definition and Function Block Instance

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.

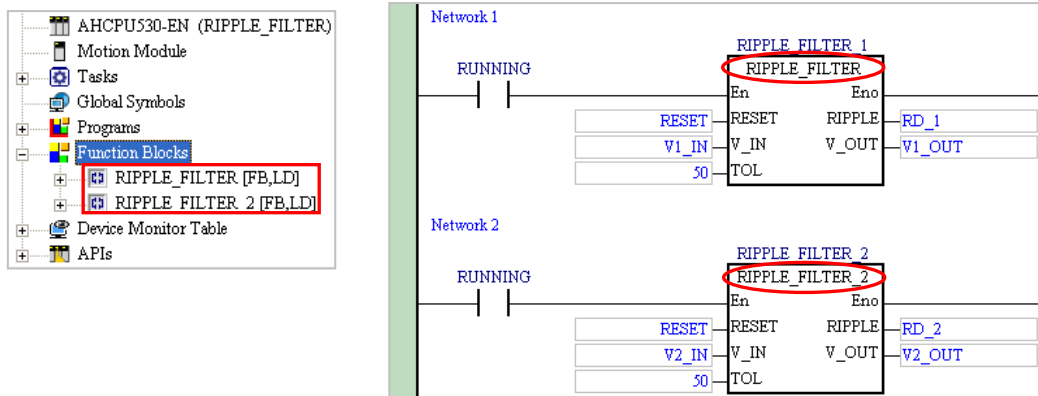


If several pieces of data in the superior POU are processed in the same way, and these pieces of data are independent of one another, the execution result may be incorrect, as shown in the figure below. After the value of V2\_IN is sent to the function block, it will be compared with the value gotten from the operation performed on V1\_IN last time. As a result, V1\_IN and V2\_IN interfere with each other.

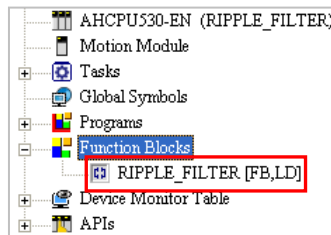
7



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.



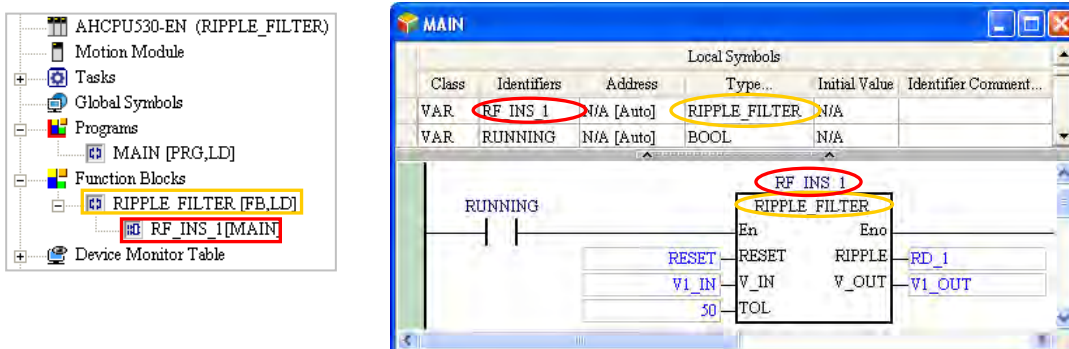
In view of this, the concept related to function block instances is introduced into IEC 61131-3. After users add a POU of the function block type, declare local symbols in the POU, and write a program in the POU, they will get a function block definition. The function block definition is like a document. It does not participate in any operation, and does not appropriate any resources that the PLC uses.



# 7

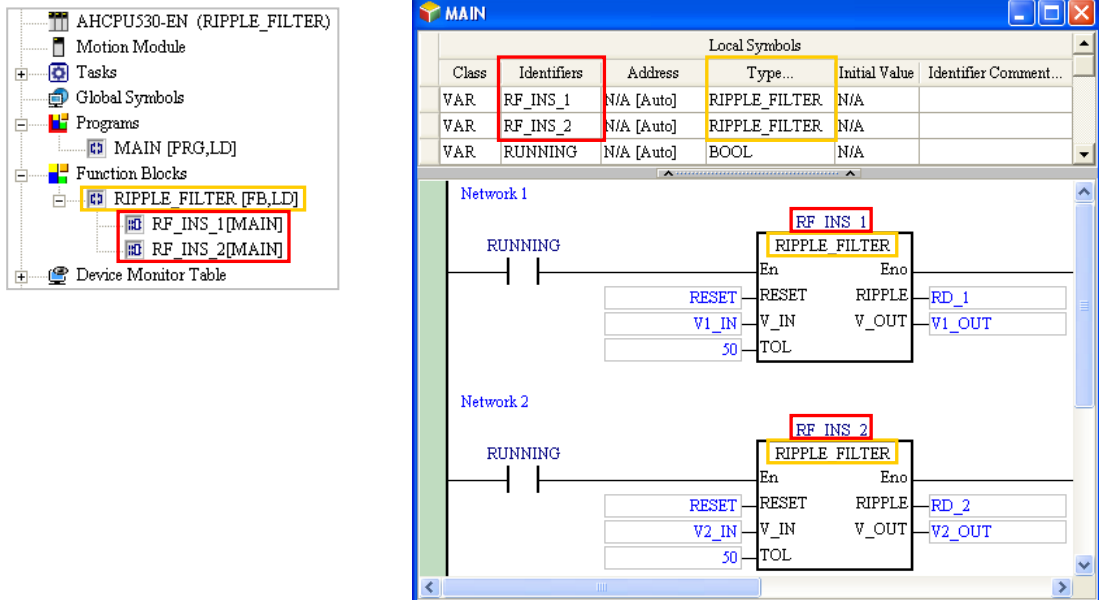
If a POU of the program type wants to call this function block, the users must declare a symbol whose data type is a function block in advance. In other words, the users have to produce an object which takes part in an operation according to the function block definition. The object produced is a function 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.

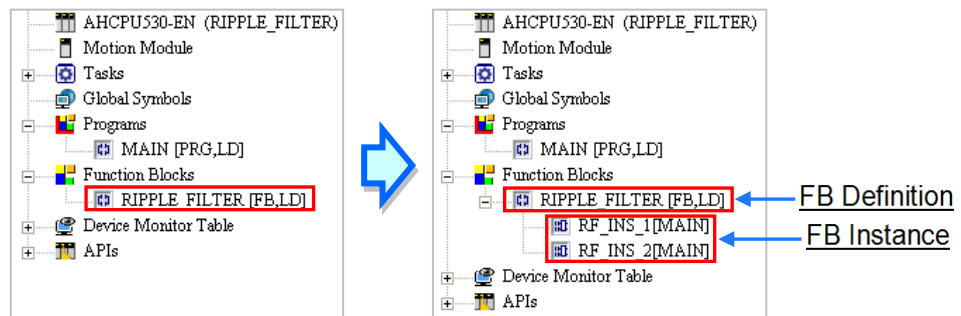




If the same operation is required, and a different memory block is also required, users can make use of the characteristic of a function block instance. After a different symbol is declared for a function block which requires a different memory block, the system will regard the different symbol as a 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]**.



7

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 AH5X0 series CPU module can not be larger than 1024. (The function block definitions which do not produce the instances are not included.)

If a symbol with a general data type is declared, a device will be assigned to the symbol when the program is compiled. Likewise, whether it will be used or not, if a symbol with a function block data type is declared, a corresponding function block instance will be produced, whenever a program is compiled. However for an AH5X0 series CPU module, the number of function block instances as POU in a project cannot exceed 1024. (The function block definitions which do not produce instances are not included.) This limitation only applies to AH5X0 series; for other series, the number of function block definitions can be as large as the size of the PLC internal memory.

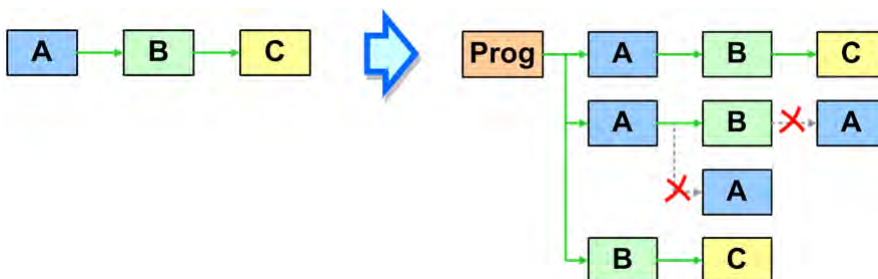
In sum, the operation of a function block involves a definition and an instance. Function block definitions are related to planning, and function block instances are related to execution. Take products for example. A function block definition is the same as the product specifications developed by engineers, and a function block instance is the same as the product produced according to the specifications.

### 7.2.6 Calling Relation Between Function Blocks

In ISPSOft function block specification, a function block can call another function block. For AH series, maximum of 32 layers can be called; AS/DVP series, a maximum of 16 layers can be called; DVPxxMC series, there are no limits on calls. To identify, the function block example called by POU is defined as the first layer.

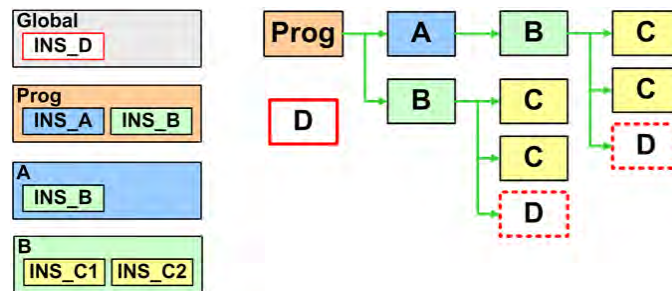


The program in function block A calls function block B, and the program in function block B calls function block C. In other words, function block A is superior to function block B and function block C, and function block B is superior to function block C. In ISPSOft, a function block is not allowed 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.

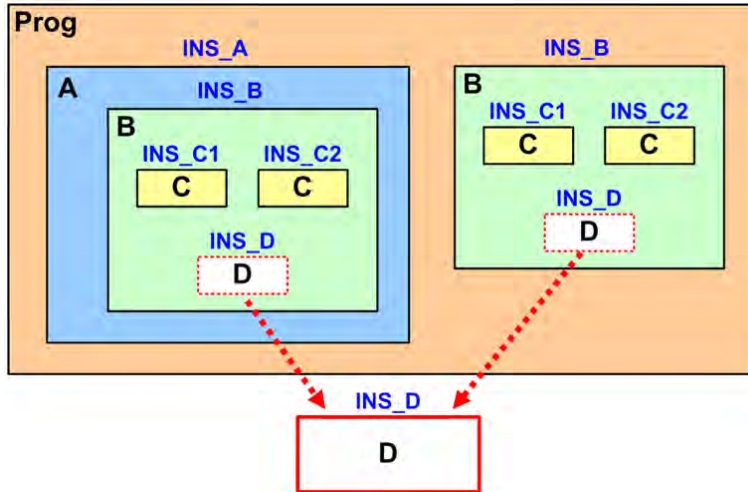
Please refer to the declaration of the function block instances and the relation among the function blocks below. Function block D is declared in the global symbol table while the other function block instances are declared in the local symbol tables in the superior POUs (callers).



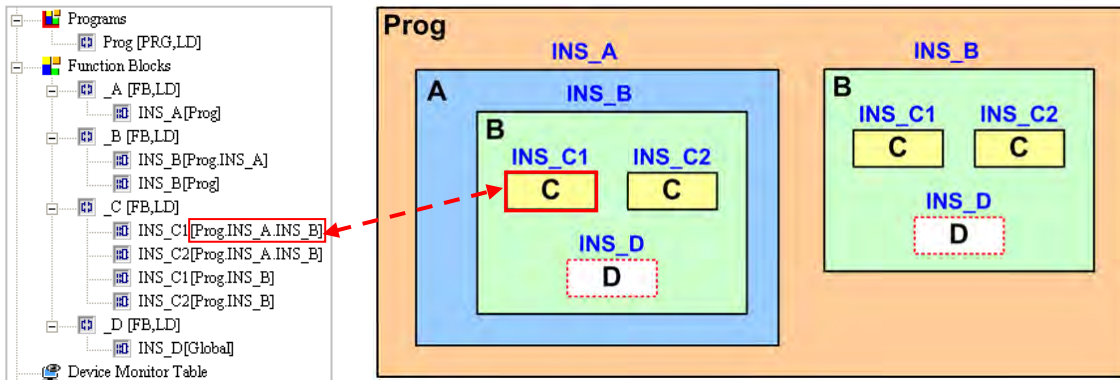
Declaration position		Function block which is declared
Global symbol table		One instance of function block D: INS_D
Local symbol table	POU of the program type	One instance of function block A: INS_A One instance of function block B: INS_B
	Function block A	One instance of function block B: INS_B
	Function 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. Since 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.



# 7

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 AH/AS 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 AH/AS 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.

## 7.2.7 Memory Configuration of Function Block

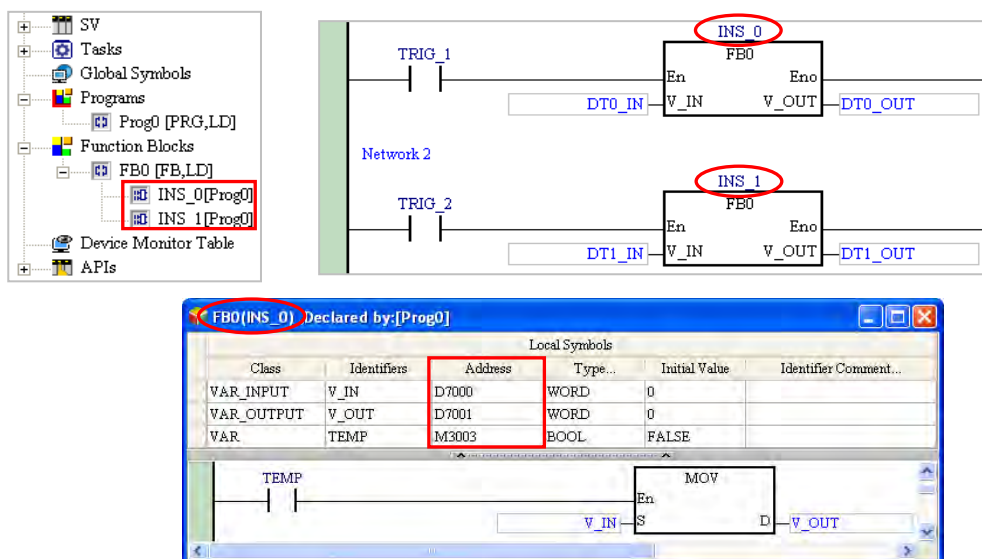
When internal function blocks has local symbols declared to auto-configure address, the system will configure an independent memory section for these local symbols during the process of compiling and creating the instance; if the internal function block declare that the local symbol has assigned device address by themselves, then the symbol variable is referred as an absolute address and a memory section is not configured to the variable during compiling. Even in different function block instances, as long as the process is related to this variable, the system will view as user-defined device content to operate.

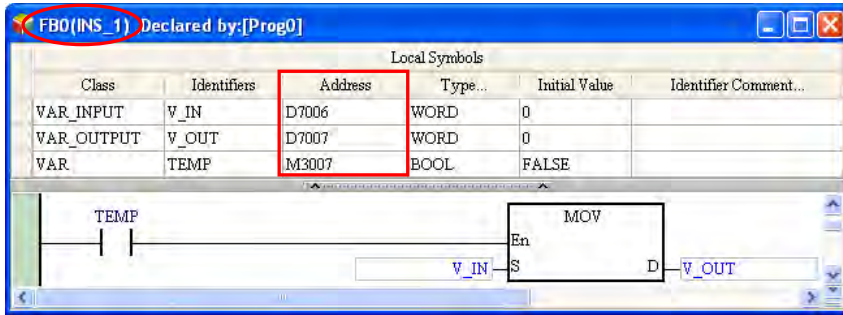
When the system is configuring resources, DVP series, DVPxxMC series, AH5X0 series, AH5X1 series, AH560 redundant system, AHxxEMC and AS series all have different mechanisms. Please refer to the following explanations.

### ● 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.

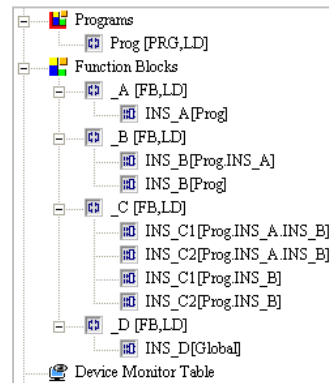
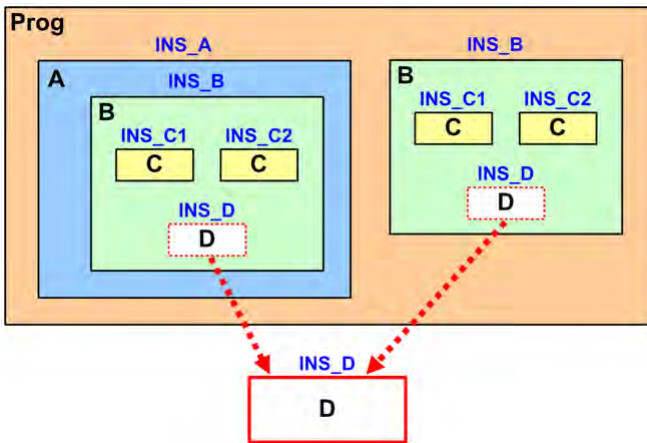
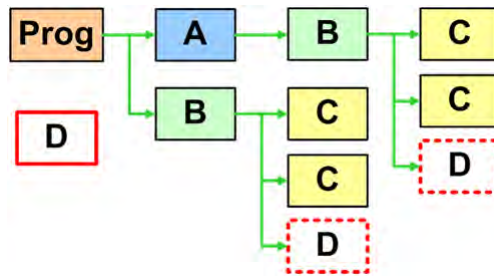




● AH5X0 series CPU module

The system in an AH5X0 series CPU module reserves a fixed number of memory blocks for function block instances. The number of memory blocks varies with the model selected and every instance produced will occupy one P device. 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 AH5X0 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 following diagram is the call structure of function block MOV section 7.2.6 example.

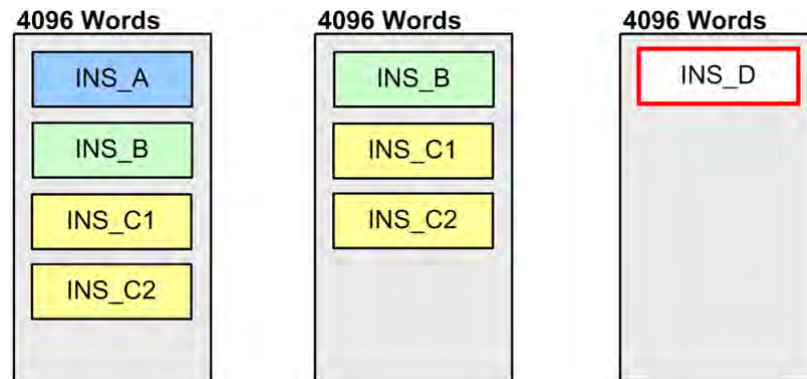


7

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 AH5X0 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 AH5X0 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 AH5X0 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.

4096 Words	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  $2 \times 4=8$  (words).

The size of the memory that the function block instances and the local symbols occupy is  $8+6+5+8=27$  (words).

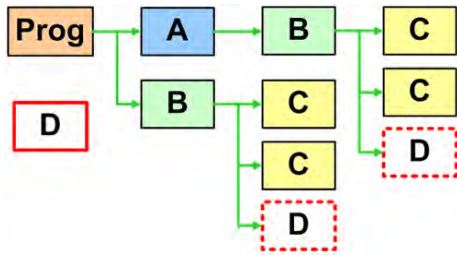


● **AH5X1, AH560 Redundant System, AHxxEMC, and AS Series**

The memory blocks that an AH5X1, **AH560 Redundant System**, AHxxEMC, or ASseries CPU module reserves for function block instances are stored in the internal memory. That means it will not take up the device resources while allocating. The difference is for AH5X1, AHxxEMC, AS series, the number of memory blocks assigned to the function block instances can be as large as the size of the PLC internal memory, but for AH5X0, the limitation is 4096 words. And one function block instance occupies one P device.

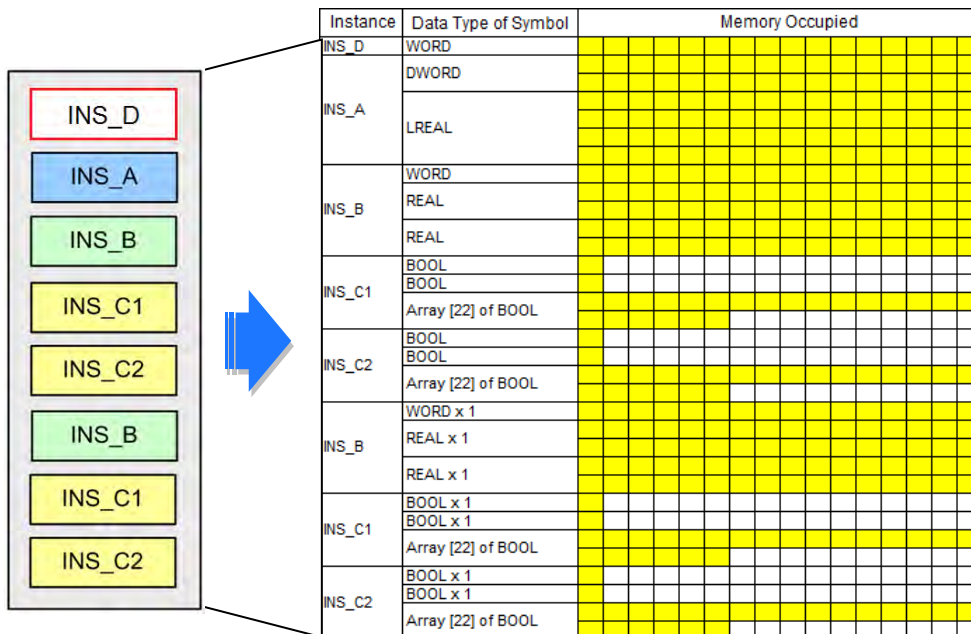


The following diagram adopts section 7.2.6 example, including an estimated amount of local and global symbol data types declared in the function block definition.



Function block definition	Number of local/global symbols
Function block A	One symbol whose data type is 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.
Function block D	One symbol whose data type is WORD.

Following figure illustrates how the internal memory occupies in the instances.



7

The calculation for the example is as follows.

- (1) All the instances will occupy the internal memory by the order of their declarations. The first declared one is the function block instance INS\_D.

➤ **Function block instance INS\_D**

➔ One word (WORD x 1)

The sum is 1 (word).

- (2) The size of the memory that a local/global 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 (WROD 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/global 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  $2 \times 4=8$  (words).

➤ **Function block instance INS\_B / INS\_C1 / INS\_C2**

➔ As the previous example shown INS\_B occupies 5 (Words) ; INS\_C1 / INS\_C2 occupies 8 (Words); and therefore the size of the memory that two function block occupy is  $5 + 8 = 13$  (Words).



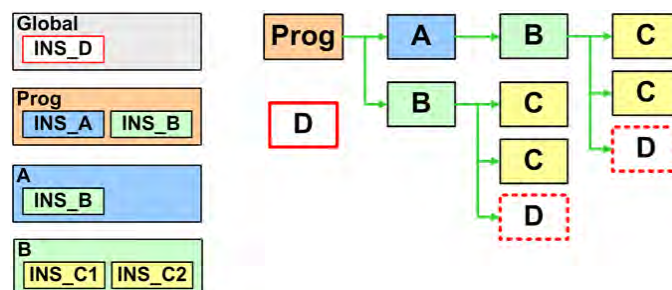
The size of the memory that the function block instances and the local/global symbols occupy is  $1 + 6 + 5 + 8 + 13 = 33$  (words).

● **DVPxxMC series PLC**

When DVPxxMC series is configured in memory section, the instance is assigned to a particular section of the memory and till the memory space reaches  $2 \times 10^{13}$  Byte. Users need to take note that the memory section is still affected by the number of POU, therefore, the more number of FB defined by users, there will be limits in corresponds to the number of instances referred by each function block definition.

For local symbols in each instance, when byte number for each symbol variable based on data size does not reach 1 byte, it is calculated as 1 byte (no device attached); if symbol variable data type is BOOL and has device attached, the device type is calculated in Bit if device type is Bit, if the device type is Byte, then it is calculated in Byte.

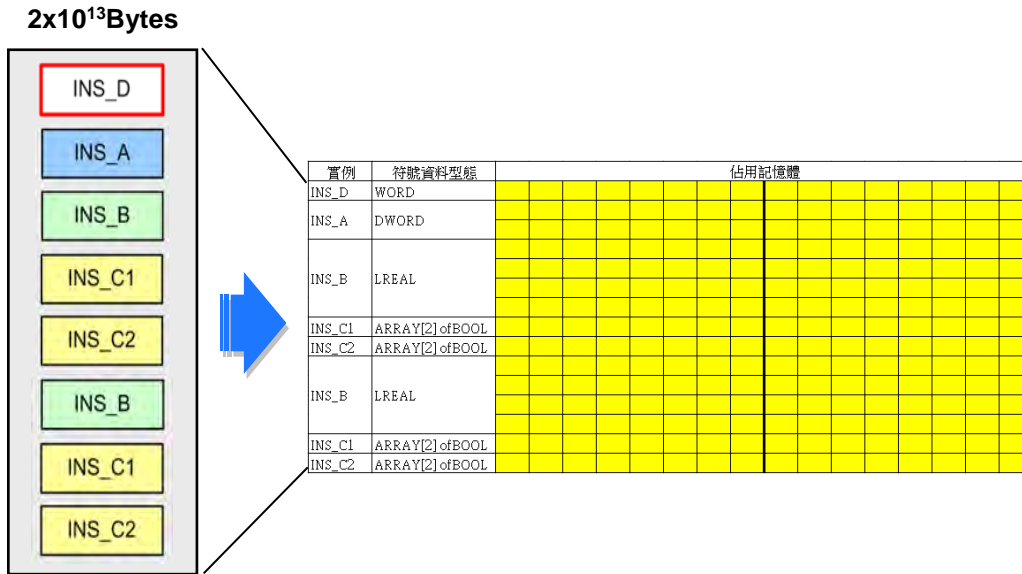
The following diagram is the section 7.2.6 instance regarding call structure of function block, please refer to the section for more details. In addition, the diagram includes an estimated amount of local and global symbol data types declared by each function block definition.



7

Function Block Definition	Amount of Local / Global Symbols
Function Block A	DWORD x 1, lower level function instance, function block instance x 1
Function Block B	LREAL x 1, lower level function instance x 2
Function Block C	BOOL array x 1 (length is 2)
Function Block D	WORD x 1

Following figure illustrates how the internal memory occupies in the instances.



The calculation for the example is as follows.

➤ **Function block instance INS\_A**

➔ 4 Bytes (DWORD x 1), total 5 (Bytes)

➤ **Function block instance INS\_B**

➔ 8 Bytes (LREAL x 1), total 8 (Bytes)

➤ **Function block instance INS\_C1/INS\_C2**

In BOOL array. The BOOL element in each array is fixed to occupy 1 Byte. Calculation are as followed.

➔ 2 Bytes (BOOL array x 1, length is 2), total is 2; function block C produces 2 instances in total. So, the two function block instances total is 2x2=4 (Bytes) i.

In short, the total memory size of the occupied section that corresponds to the function block is 5+8+4=17 (Bytes).



## 7.3 Using a Function Block

### 7.3.1 Basic Specifications for 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

<b>Number of function block definitions</b>	All series: no limitation
<b>Number of function block instances</b>	AH5X1/ AH560 Redundancy System / AHxxEMC/AS: Depends on the controller devices available. AH5X0: The maximum number of memory block instances varies according to the series (Please refer to appendix B).Depends on the controller devices available. DVP: Depends on the controller devices available.
<b>Size of a function block</b>	AH5X1/ AH560 Redundancy System/AS: Without limit, until the internal memory is used up. AH5X0: The size of a memory block instance is 4096 words. DVP: Depends on the devices assigned to the function block.
<b>Number of layers</b>	For AH series, the nest structure of function block calling function block, maximum layer is 32 ; AS/DVP series, maximum layer is 16; DVPxxMC series does not have limit.

#### ● 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, function block diagrams and continuous function charts. 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, function block diagrams and continuous function charts. 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 AH/AS 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 AH/AS series CPU module. LDP/ANDP/ORP, LDF/ANDF/ORF, PLS, PLF, NP, PN, MC, MCR, and GOEND can not be used.
- AH/AS 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 10~chapter 15 for more information about calling function blocks.**



### 7.3.2 Pulse Instructions for Function Blocks (AH/AS Series ONLY)

Pulse instructions such as LDP, LDF, NP, and PN can not be used in the function blocks in a project for an AH/AS series CPU module, and therefore users can consider using the following instructions. Please refer to AH/AS Programming Manual for more information about the instructions.

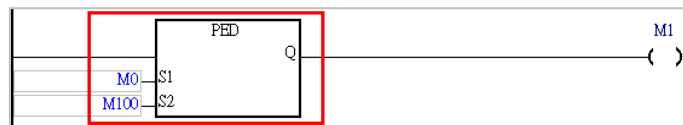
- PED and NED

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.

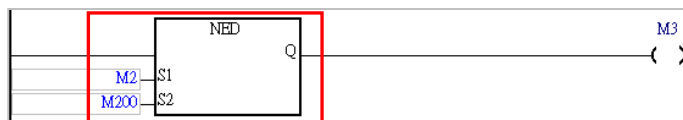
- LDP (rising-edge detection) and LDF (falling-edge detection) can be used in a POU of the program type.



- 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.



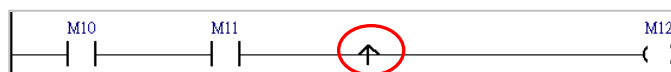
- Users can use NED in place of LDF in a function block, but they have to specify a device, e.g. M200 in the figure below.



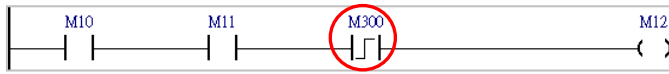
- FB\_NP and FB\_PN

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.

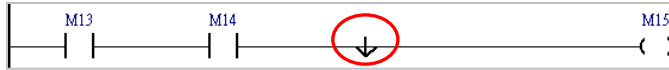
- NP can be used in a POU of the program type.



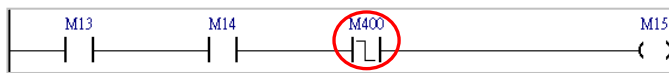
- Users can use FB\_NP in place of NP in a function block, but they have to specify a device, e.g. M300 in the figure below.



- PN can be used in a POU of the program type.

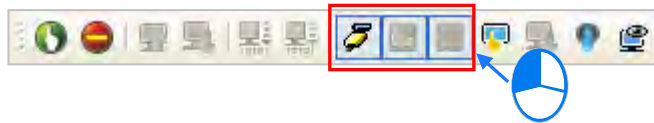


- Users can use FB\_PN in place of PN in a function block, but they have to specify a device, e.g. M400 in the figure below.



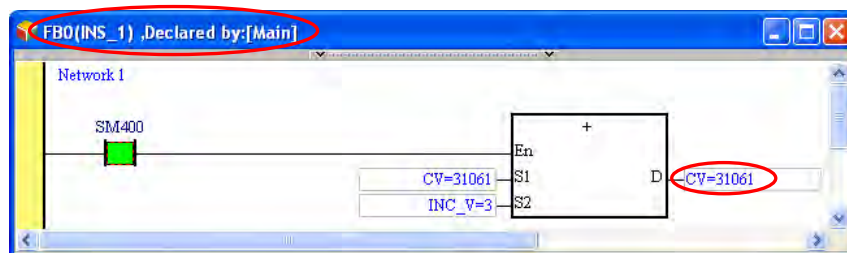
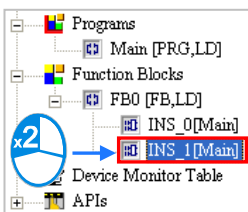
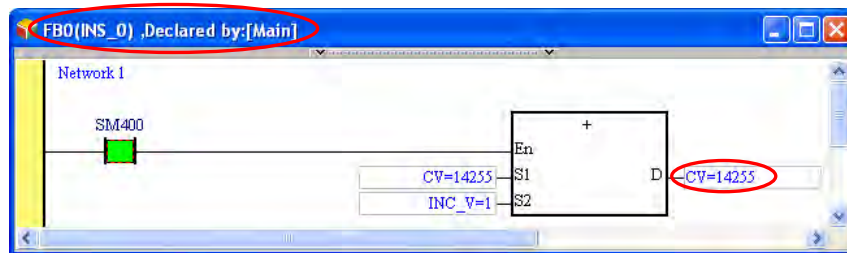
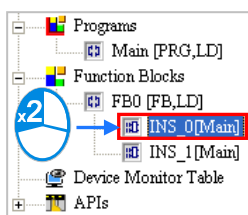
### 7.3.3 Monitoring the Program in a Function Block

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.

7





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

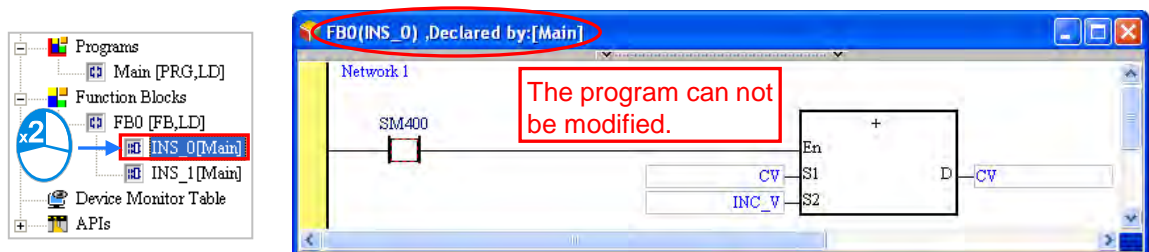
The object produced from a function block definition is a function block instance. The program in a function block instance can not be modified, but the program in a function block definition can be modified. After the program in a function block definition is modified, the modification result will be 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.

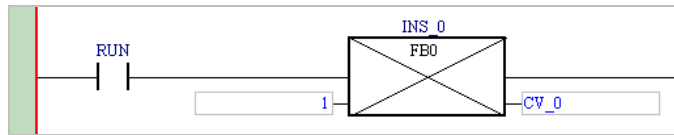


7

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.



## 7.4 Instance

An instance 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 POU and TASK. 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 POU's 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.



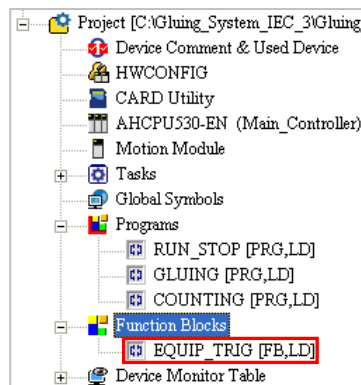
Two function blocks will be created according to the characteristics of the POU's.

- 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.

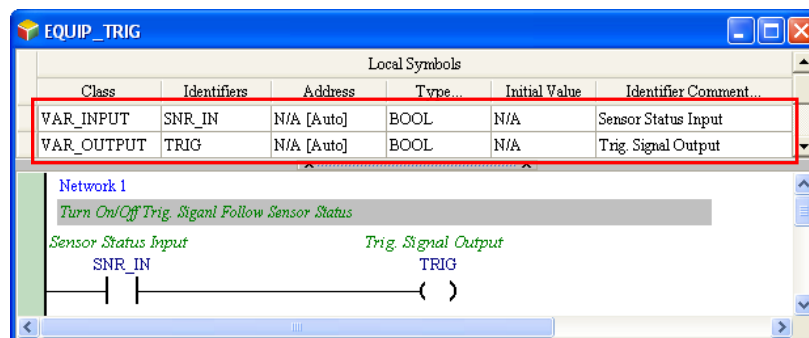
## 7.4.2 Creating the Program

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.

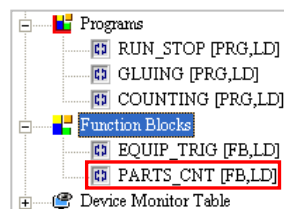


The program in **EQUIP\_TRIG** functions to inject the glue. Please refer to section 6.2.2, and create the two symbols shown below in the local symbol table in the **EQUIP\_TRIG** window. Please refer to chapter 4–chapter 6, and write the program shown below.

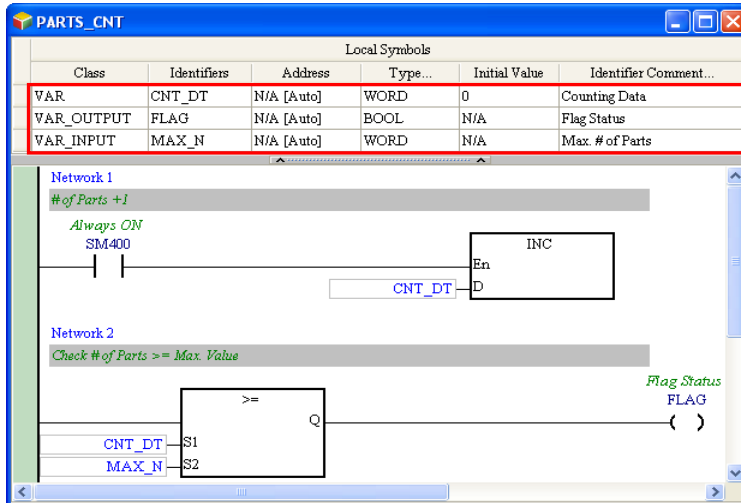


7

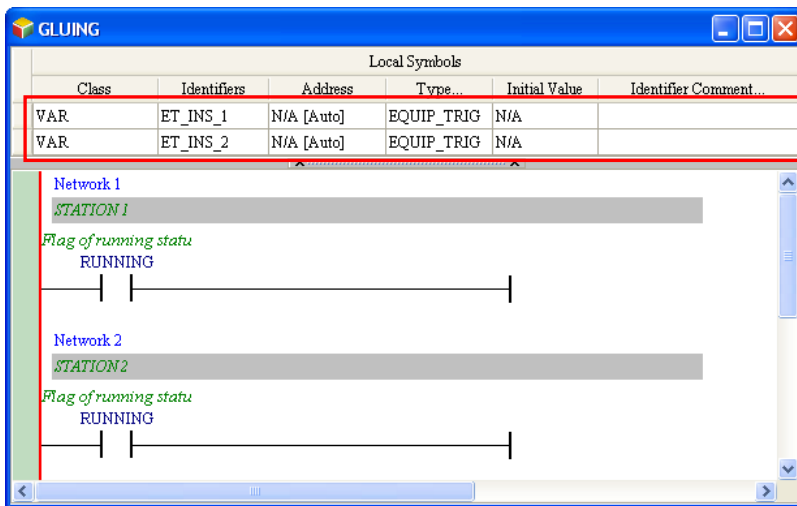
Create **PARTS\_CNT** in the **Function Blocks** section in the project management area.



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.

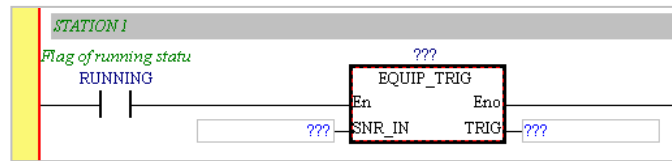
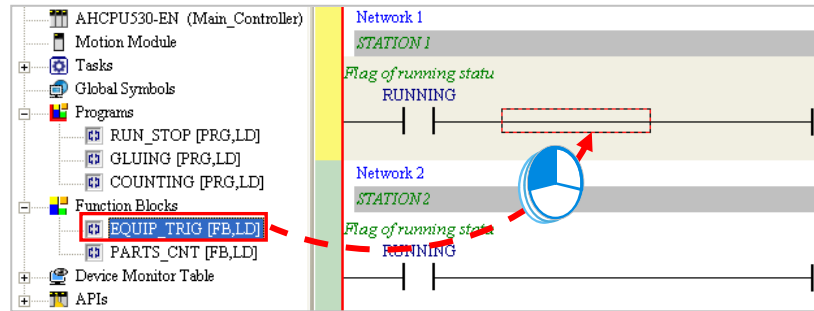


The function block **EQUIP\_TRIG** is called by the POU **GLUING**. Please refer to section 6.2.2, and declare the two function block instances shown below in the local symbol table in the **GLUING** window. Write the program shown below.

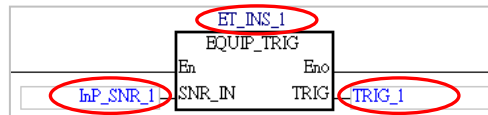


7

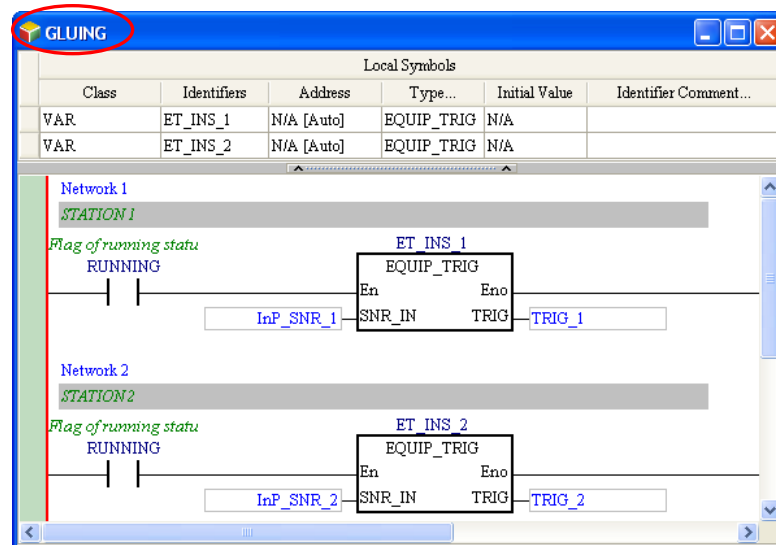
A function block is inserted into a program in a way in which an applied instruction is inserted into a program. Drag the function block **EQUIP\_TRIG** in the project management area to the position where it will be inserted. If there is a function block instance under **EQUIP\_TRIG** in the project 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.

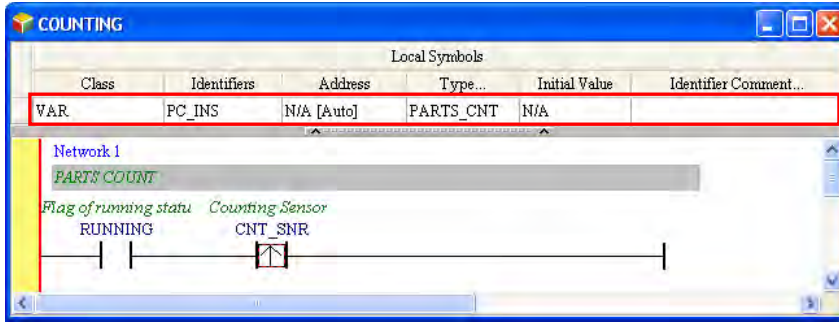


Write the program shown below.

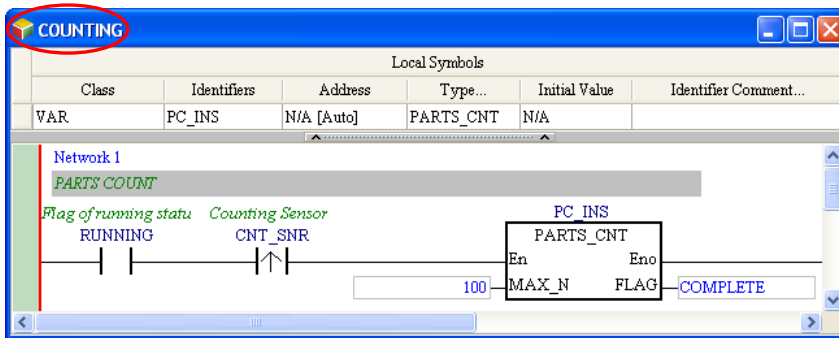


7

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.

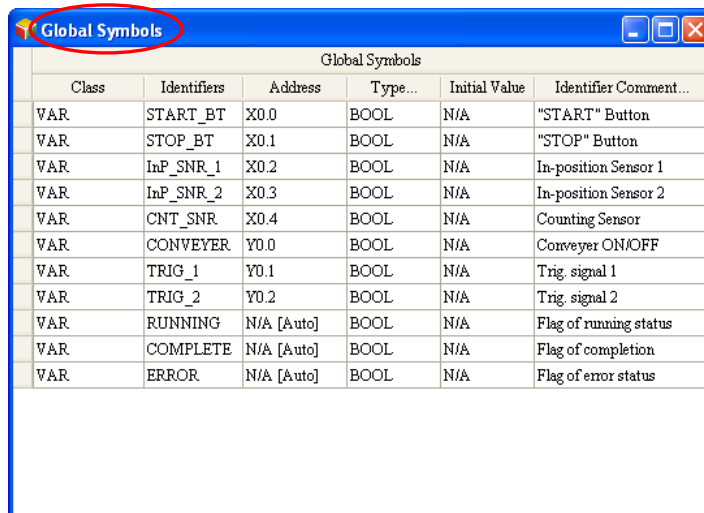
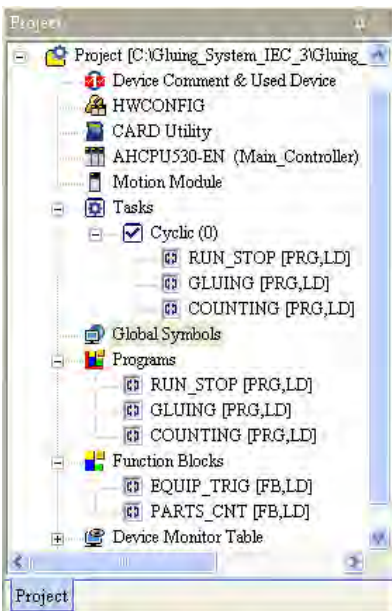


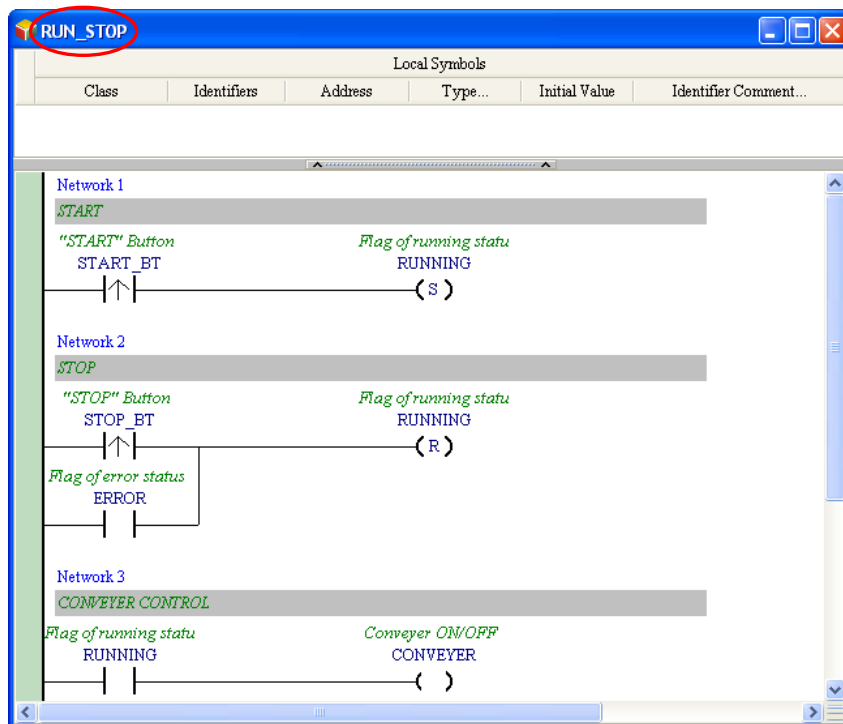
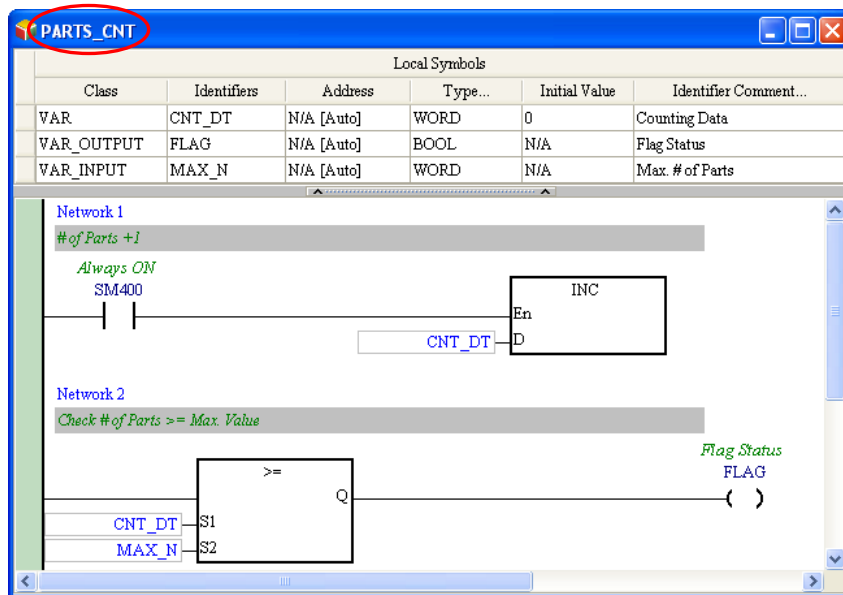
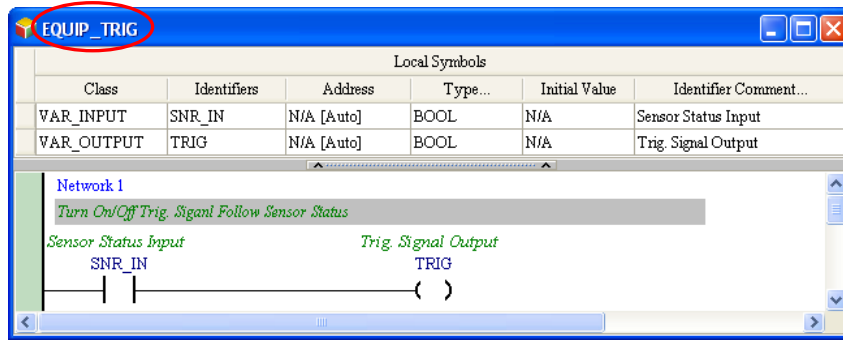
Please refer to the description above, and insert the function block **PARTS\_CNT**. Write the program shown below.



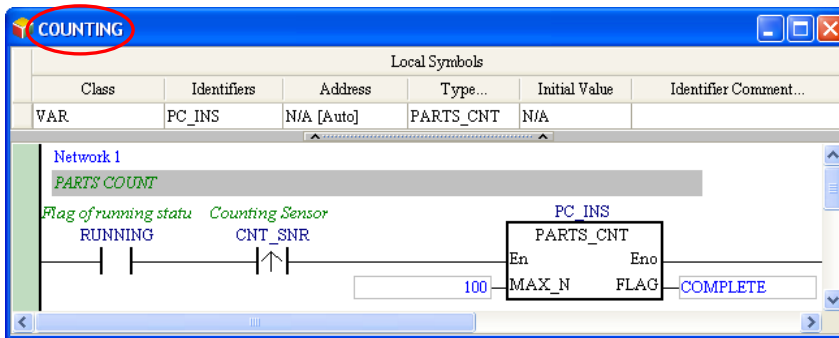
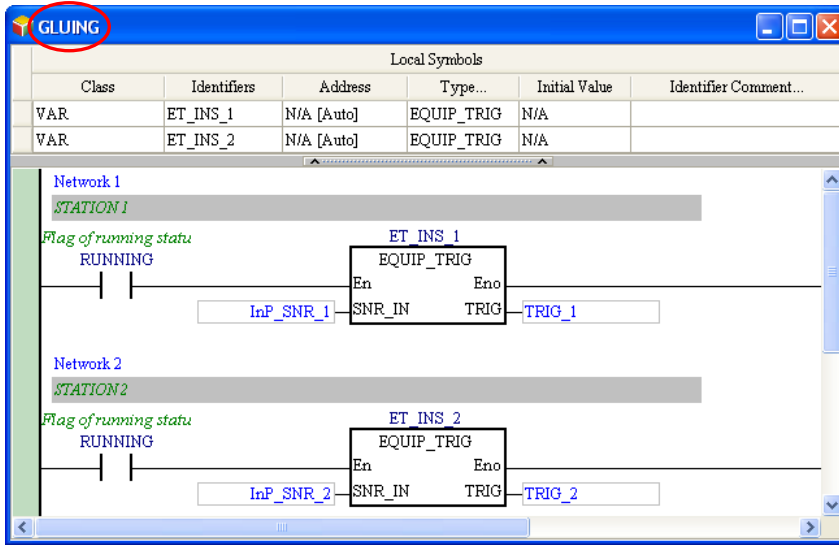
The editing of the program in this section has been completed so far. Users have to make sure that the program they write is the same as the program shown below.

7



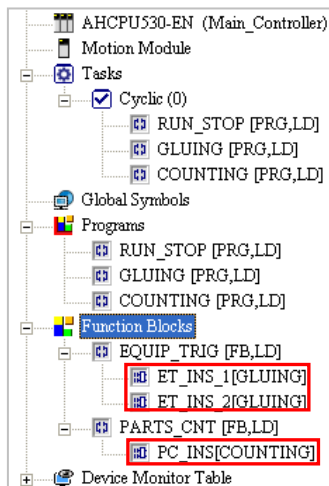


7



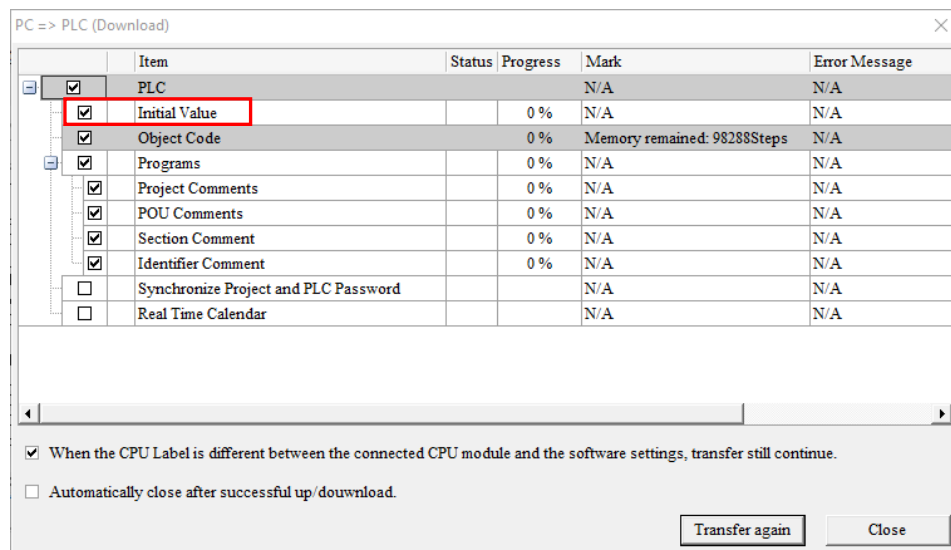
After the users make sure that the program they write is the same as program shown above, they can try to compile the program. After the program is compiled, the function block instances declared will appear in the project management area.



7





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.

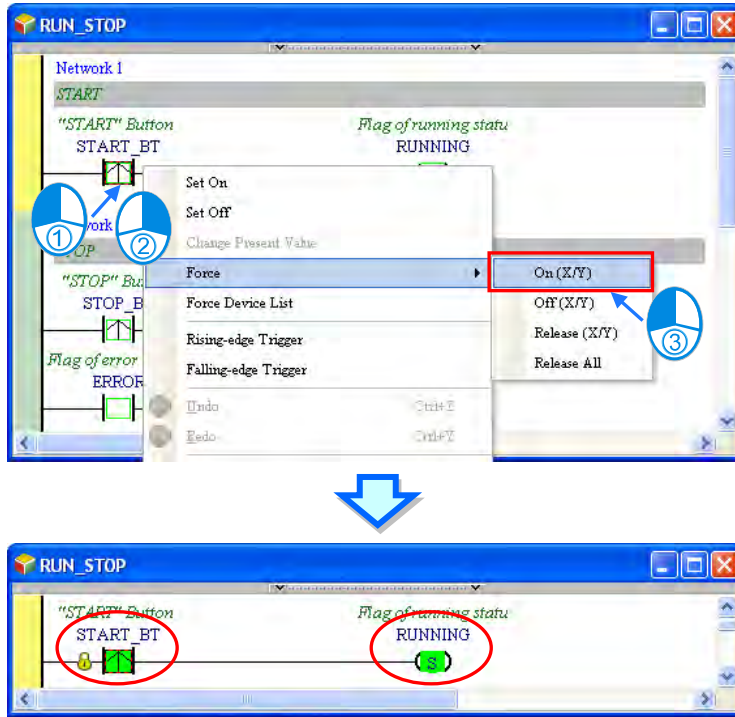


When the program is downloaded, the users can click  and  on the toolbar. Please follow the steps below, and test the characteristics of the function blocks.



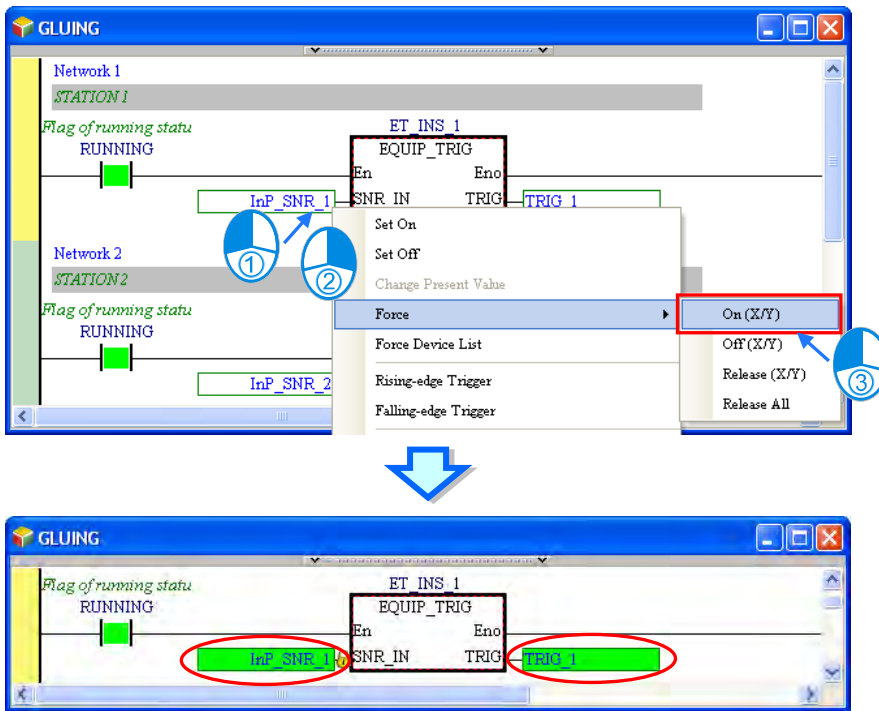
Open the window for **RUN\_STOP**, and force **START\_BT** ON. **RUNNING** is ON. The equipment begins to run.

7

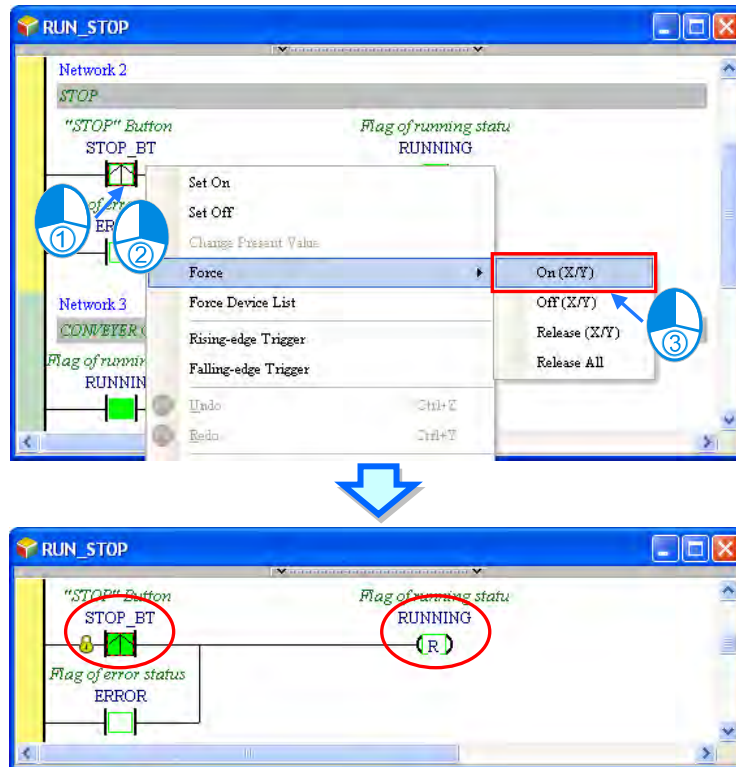


Open the window for **GLUING**, and force **InP\_SNR\_1** ON. **TRIG\_1** is ON. The sensor senses a part, and a trigger signal is sent.

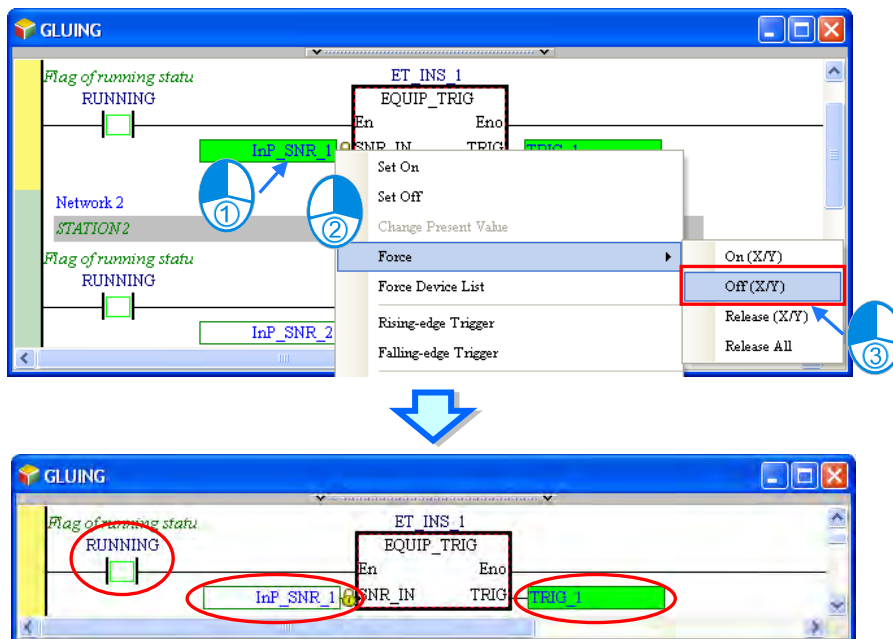
7



Open the window for **RUN\_STOP**, and force **STOP\_BT** ON. **RUNNING** is reset to OFF. The equipment stops running.

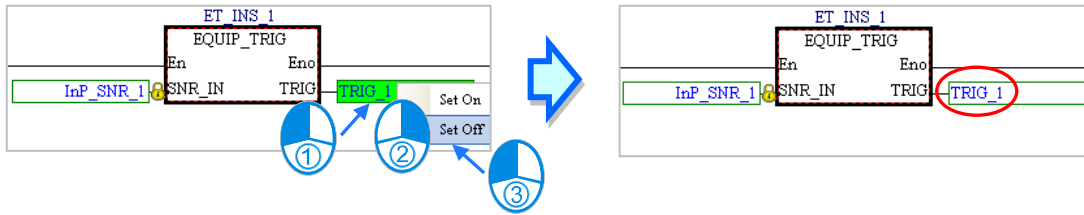


Open the window for **GLUING**, and force **InP\_SNR\_1** OFF. **TRIG\_1** is still ON. The situation is correct. **RUNNING** is a flag connected to the En pin of the function block. **RUNNING** is OFF, and therefore the function block is not executed. The value of **TRIG**, a pin of the function block, is not sent to the symbol **TRIG\_1**, and therefore the state of **TRIG\_1** remains unchanged.

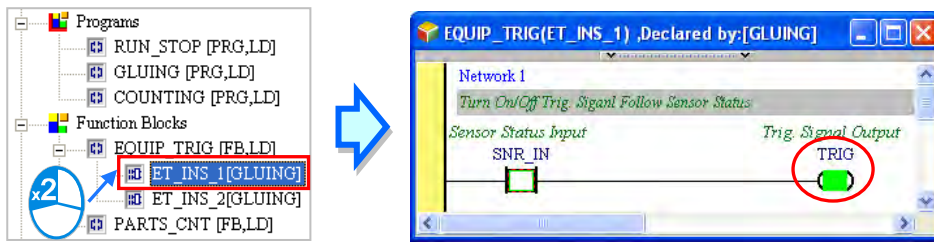


7

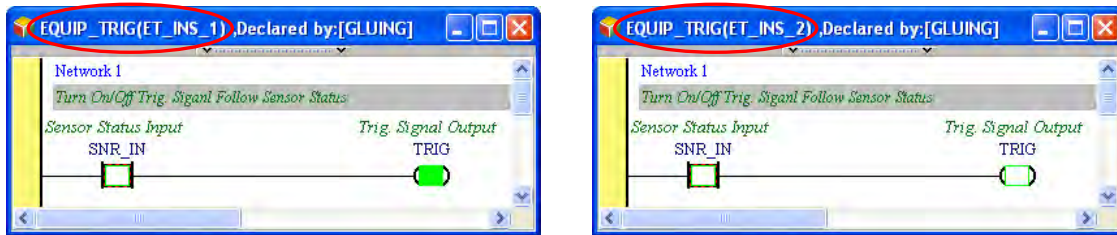
Set **TRIG\_1** to OFF.



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.



# 7

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.

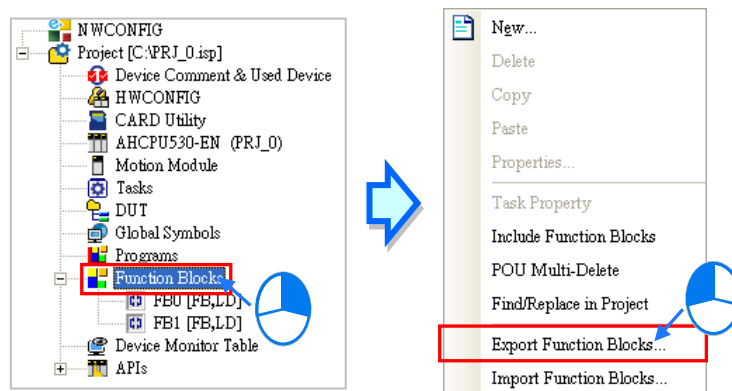
Obviously, the execution of the program in this section is different from the execution of the program in section 6.3. In the program in section 6.3, **TRIG\_1** is OFF when the flag **RUNNING** is OFF. In the program in this section, **TRIG\_1** is ON when the flag **RUNNING** is OFF. This is an expectable 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.

## 7.5 Knowing the Library

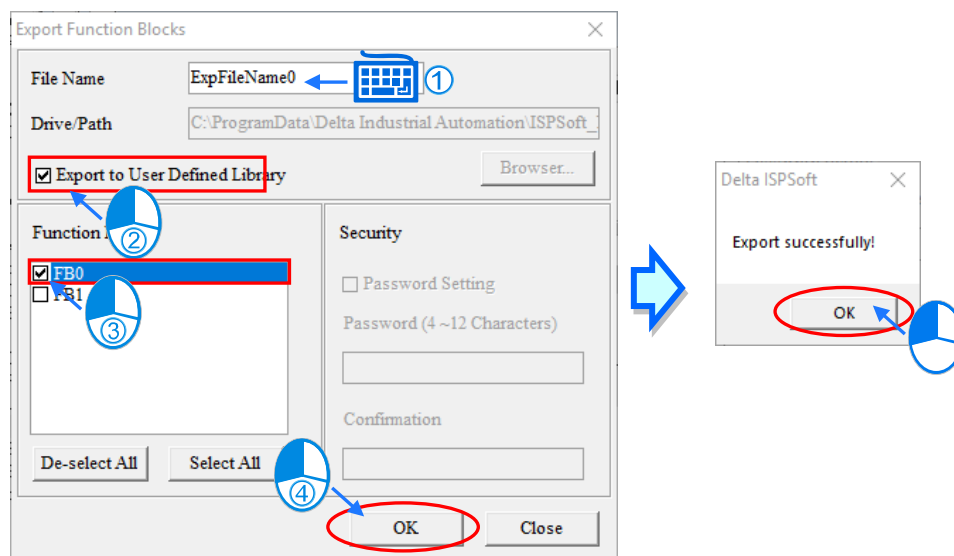
Library contains User Defined Library and Delta Library. A function block can be exported to the User Defined Library and can be used in other projects. There are several function blocks for various applications stored in the Delta Library for users to apply.

### 7.5.1 Creating the User-defined Library

When there is a function block created in a project, the function block can be exported to the user-defined library and by doing so, the function block can be used in other projects. Right click the **Function Blocks** under the Project node and click **Export Function Blocks**.

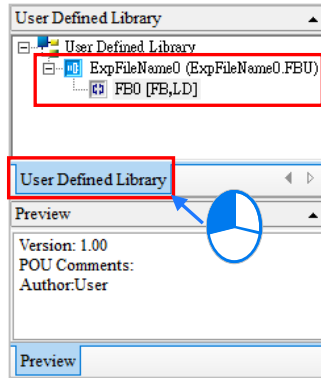


After that you will see the Export Function Blocks window, type the file name, select the option **Export to User Defined Library** and click to select the function blocks you'd like to export. File name is used for categorization; one file can contain more than 1 function block. After the setup is done, click **OK** to confirm the setting. Click **OK** again once the Export Successfully window show up.

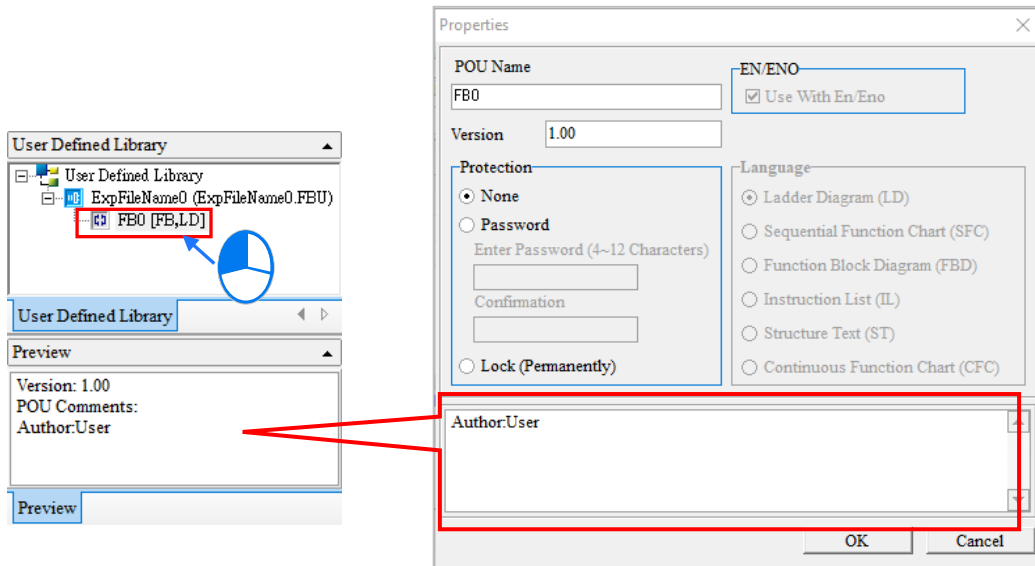


7

Click the User Defined Library tab and you will see the exported function blocks in this section.



Click on the Function Block, its properties will be shown in the Preview section.



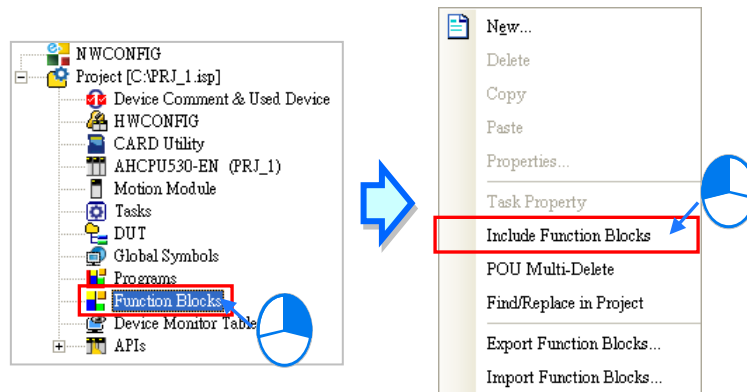
7

## 7.5.2 Including the Function Blocks in the User Defined Library

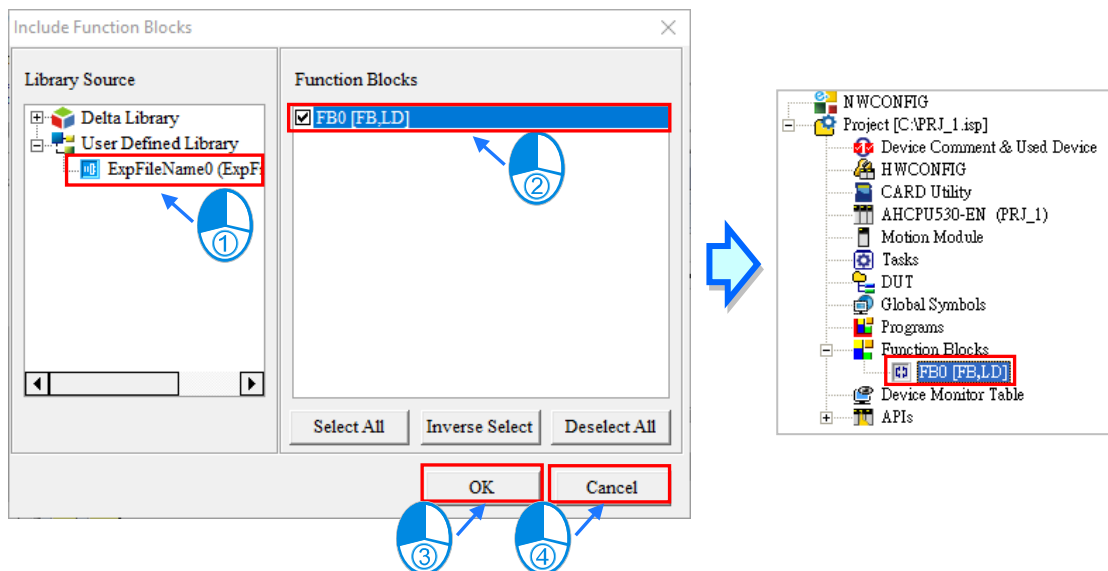
Aforementioned, once a function block is exported to the User Defined Library, users can find the exported function block in the User Defined Library when opening a new project in the same computer. Through the following methods, users can include the function to the function block.

- Method 1:

Right click the **Function Blocks** under the Project node and click **Include Function Blocks**.

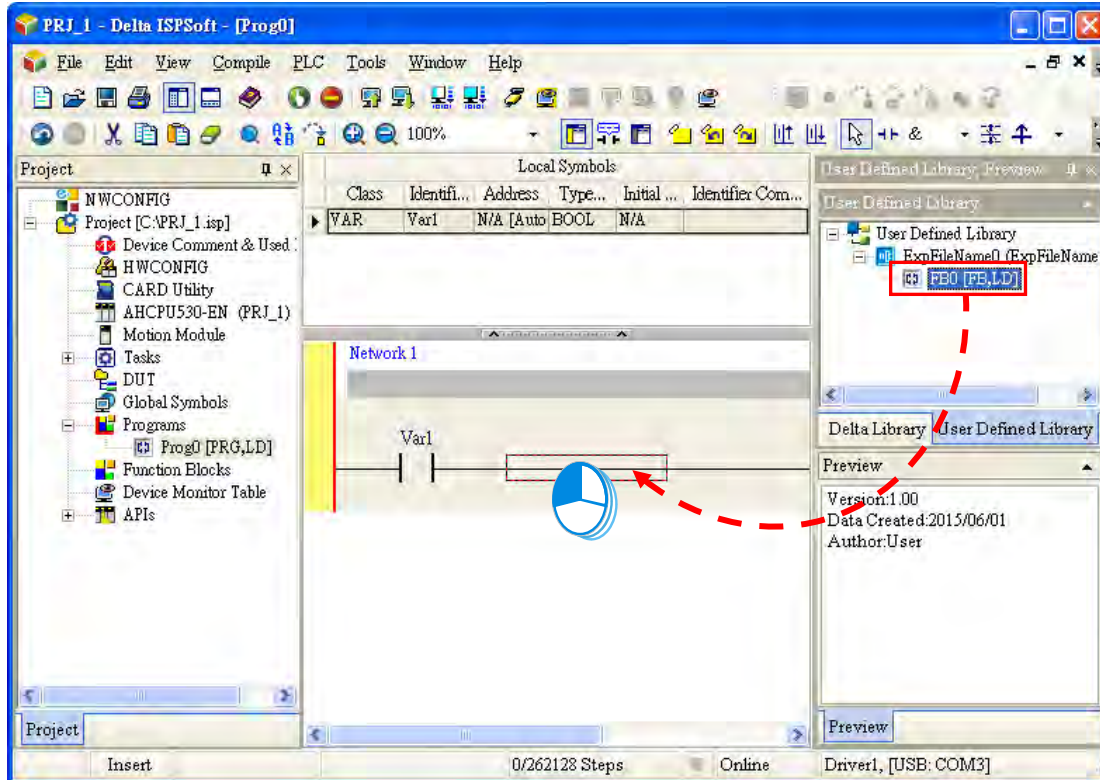


After that you will see the Include Function Blocks window, select the function that you'd like to include and click **OK**. Click **Cancel** to close the window. After that you will see the newly included function under the Function Blocks node.



- Method 2:

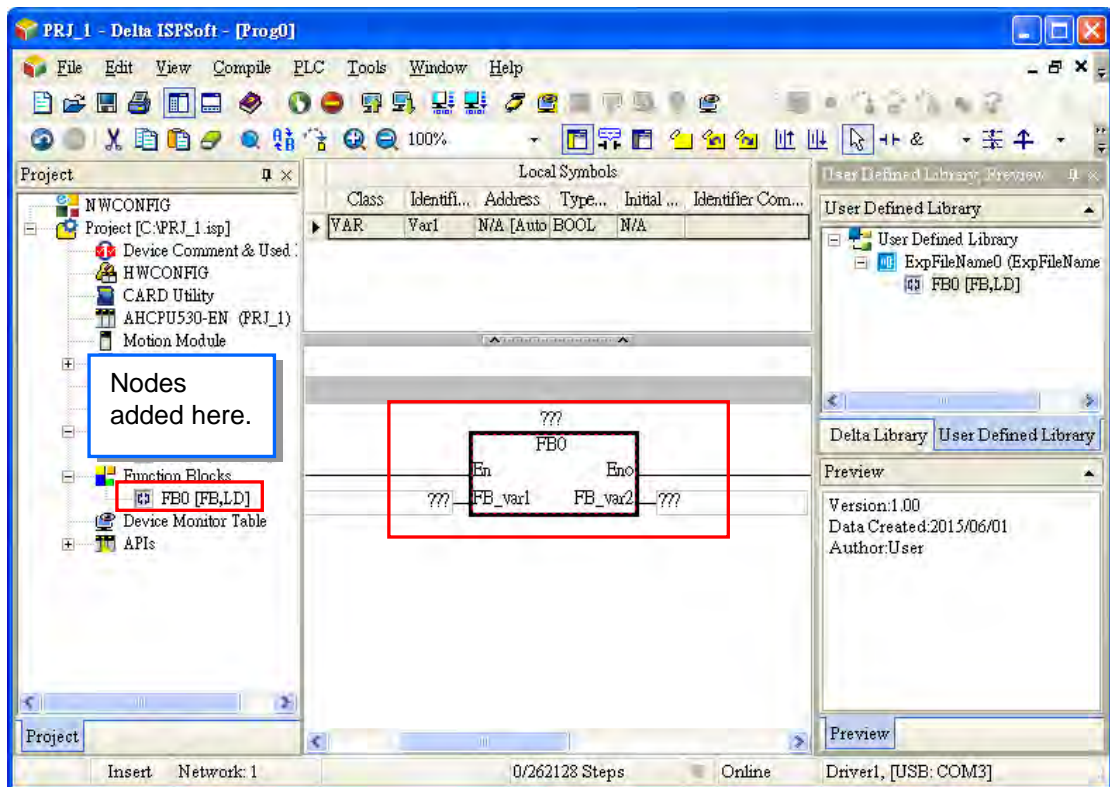
In the User Defined Library click and drag the function you'd like to use to the position where you'd like to add in the POU and it will be added in the POU.



7

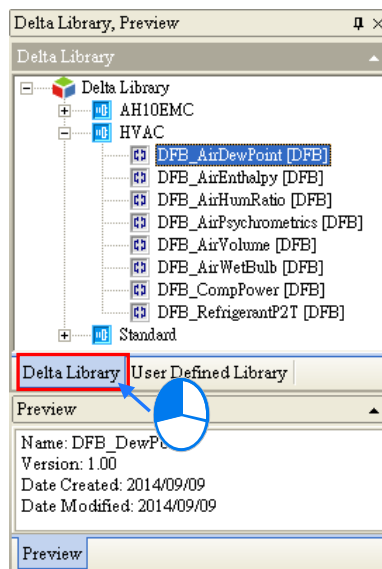


After that you will see the newly added function under the Function Blocks node.



### 7.5.3 Using Delta Library

For various applications, Delta provides several function blocks stored in the Delta Library for users to apply. Click the Delta Library tab to see the provided function blocks.

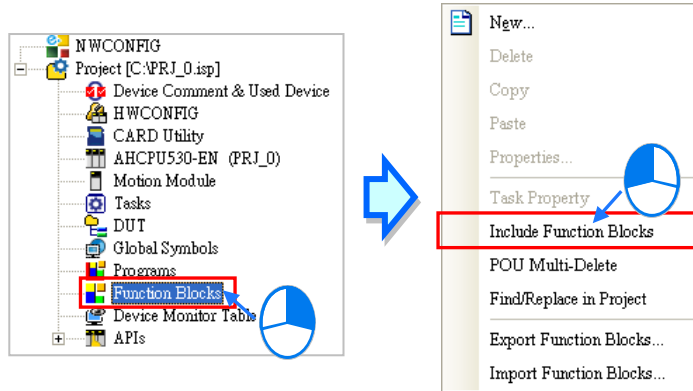


7

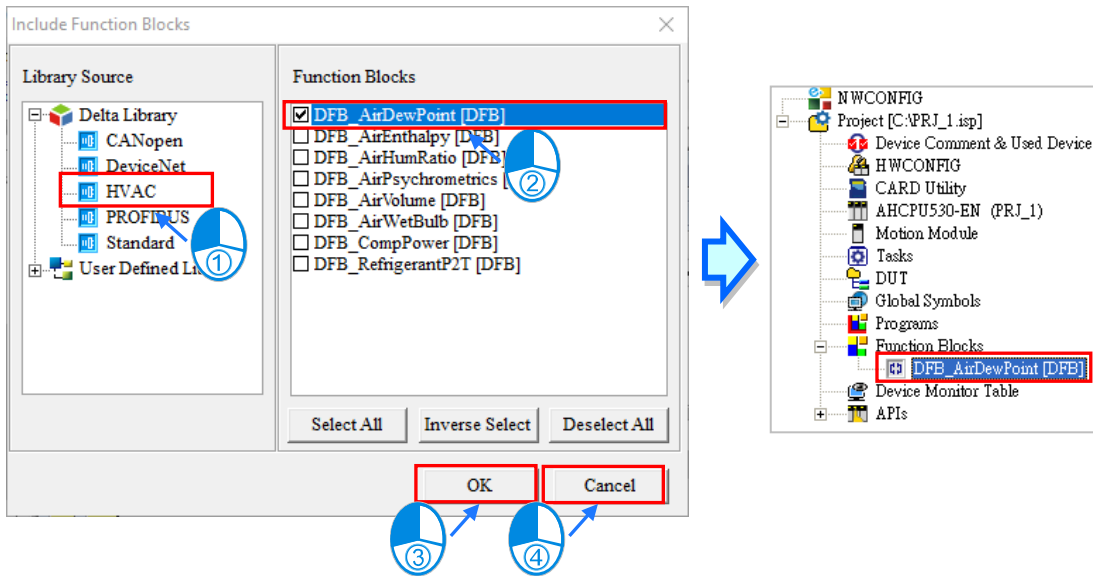
Through the following methods, users can include the function to the function block.

- Method 1:

Right click the **Function Blocks** under the Project node and click **Include Function Blocks**.

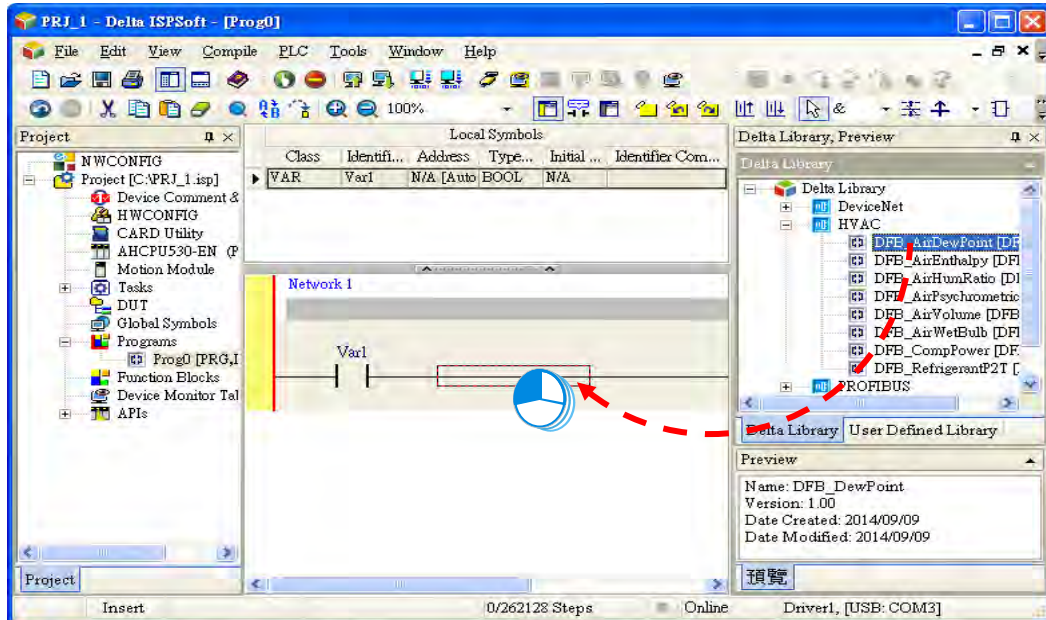


After that you will see the Include Function Blocks window, select the function that you'd like to include and click **OK**. Click **Cancel** to close the window. After that you will see the newly included function under the Function Blocks node.

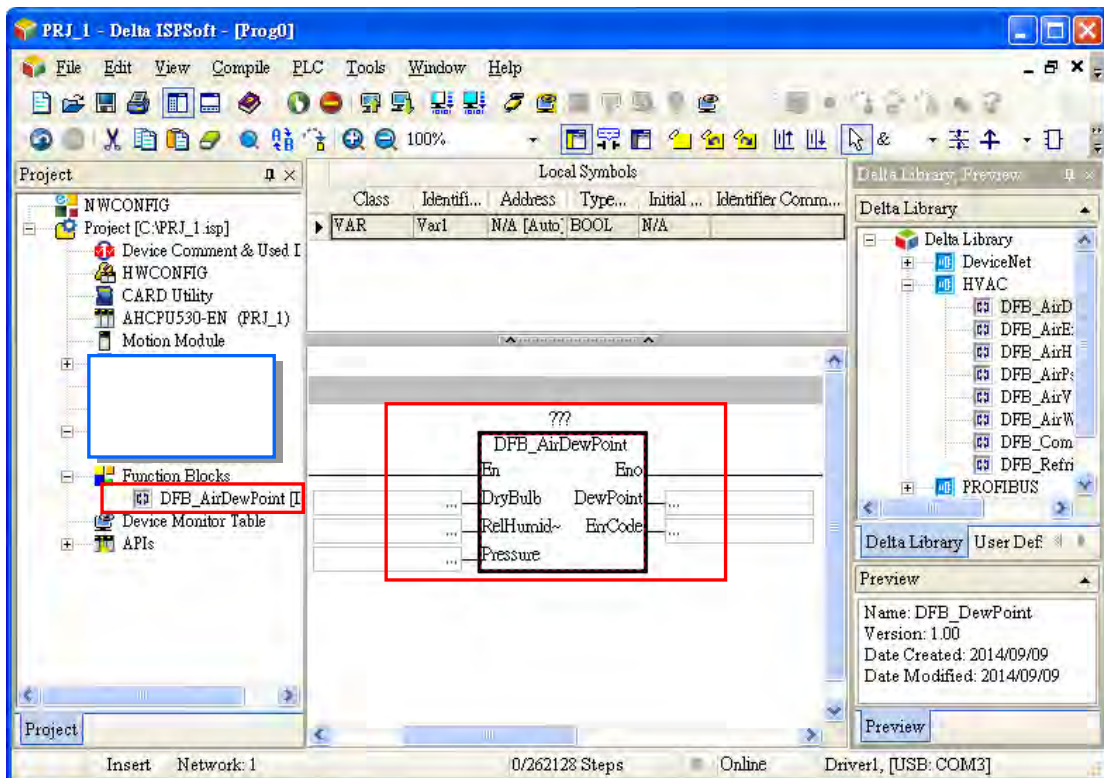


- Method 2:

In the Delta Library click and drag the function you'd like to use to the position where you'd like to add in the POU and it will be added in the POU.



After that you will see the newly added function under the Function Blocks node.



7

\*1. For the function blocks in the Delta Library, the contents are not for users to open to view or edit.

**MEMO**

7



# Chapter 8 Data Unit Type

## Table of Contents

<b>8.1 User-defined Data Type/Data Unit Type</b> .....	<b>8-2</b>
<b>8.2 Structure</b> .....	<b>8-2</b>
8.2.1 Definition of a Structure .....	8-2
8.2.2 Creating a Structure.....	8-2
8.2.3 Using a Structure Variable.....	8-5
8.2.4 Applications of Structures .....	8-6
8.2.5 Change Structure to Enumeration .....	8-8
<b>8.3 Enumeration</b> .....	<b>8-9</b>
8.3.1 Definition of Enumeration .....	8-9
8.3.2 Creating an Enumeration .....	8-9
8.3.3 Using an Enumeration Variable .....	8-11
8.3.4 Applications of Enumerations.....	8-13
8.3.5 Change Enumeration to Structure.....	8-14

## 8.1 User-defined Data Type/Data Unit Type

Sometimes basic data types are not sufficient for users to write their own programs for the PLC. In the IEC 61131-3 standard, a high-level language such as the user-defined data type is adopted to create a new data type as users desired, allowing users to define their own variables and improving the readability and the efficiency of developing a program.

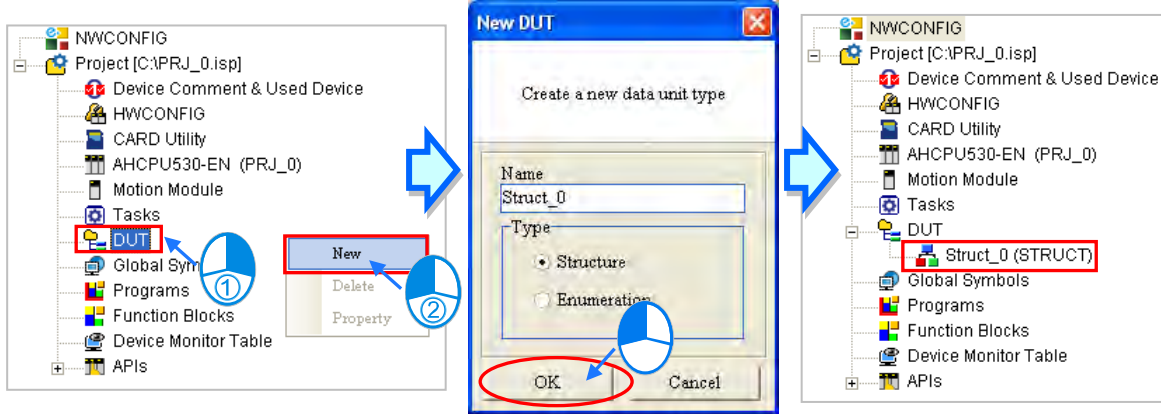
## 8.2 Structure

### 8.2.1 Definition of a Structure

A structure is the collection of data, consisting of same amount various data types. Structured or array elements can be elements for other structures. A structure is used for grouping data and can be seen as a data unit to transfer parameters.

### 8.2.2 Creating a Structure

Right-click **DUT (Data Unit Type)** in the project management area to see the option **New**. Click **New** to bring out a **New DUT** window. Input a structure name, select the type as **Structure** and then click **OK** to confirm the setting. After that, a new DUT with name and type will appear under the DUT option.



8

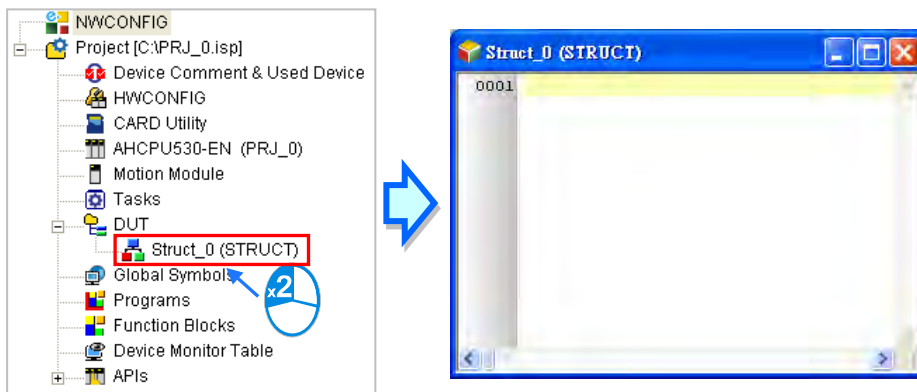
\*. The naming rules of a structure and an element can be found below.

After a structure DUT is created, an editing window will be presented with examples and definitions for reference. Users can edit the examples to make them as their own structures. . .

```

0001 *****
0002 ** Example1:
0003 **
0004 ** TYPE DUT_StructA :
0005 ** STRUCT
0006 **   Element1 : BOOL;
0007 **   Element2 : REAL;
0008 **
0009 **   (Array notation: ARRAY [lowerbound_index..upperbound_index] OF DataType)
0010 **   (Hint: the max dimension of ARRAY is 3)
0011 **   Element3 : ARRAY [0..1][0..3][0..5] OF INT;
0012 ** END_STRUCT
0013 ** END_TYPE
0014 **
0015 ** Example2:
0016 **
0017 ** TYPE DUT_StructB :
0018 ** STRUCT
0019 **   Element1 : DUT_EnumA; (Hint: using Enum in Struct declaration)
0020 **   Element2 : ARRAY [0..7] OF DUT_StructA;
0021 ** END_STRUCT
0022 ** END_TYPE
0023 *****
0024
0025 TYPE Structure_0 :
0026 STRUCT
0027   Element1 : BOOL;
0028 END_STRUCT
0029 END_TYPE
    
```

Users can also find it under the DUT option in the project management area. Double click the new DUT to open it and start to add the contents of the newly added DUT.



The structure format is stated as below: wordings in blue are the system keywords; wordings in black are the user-defined structure names and element names.

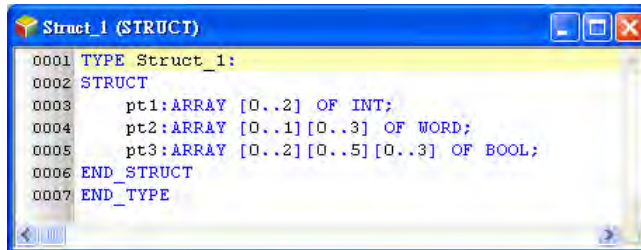
**TYPE** <Structure Name>:  
**STRUCT**  
 <Element 1 Name>: <Data Type>;  
 <Element 2 Name>: <Data Type>;  
 ...  
 <Element N Name>: <Data Type>;  
**END\_STRUCT**  
**END\_TYPE**

```

0001 TYPE Struct_0:
0002 STRUCT
0003   pt1:BOOL;
0004   pt2:WORD;
0005 END_STRUCT
0006 END_TYPE
0007
    
```



- **Structure Name:** Define the structure name. It is recommended to use the same name as the name shown on the project management area.
- **Element Name:** Define the element name.
- **Data Type :** Define the data type for every element, such as BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, REAL, LREAL, ARRAY, or another STRUCTURE or,ENUMERATION. (For DVP15MC, data types include BYTE, SINT, USINT, UINT, UDINT, ULINT, TIME, DATE, TOD, DT, STRING.) As the image shown below, when the element data type is ARRAY, the format will be **ARRAY [0..n] OF <data type>**; the maximum for n value is 65535. (For DVP15MC, the maximum for n value is 2147483647.) If the element data type is Structure, write another structure name. But if the data type is ARRAY, users cannot import ARRAY again. The system supports multi-dimensional array, up to three-dimension. The format of a two-dimensional array is **ARRAY [0..n] [0..n] OF <data type>** and the three-dimensional array is **ARRAY [0..n] [0..n] [0..n] OF < data type>**. As the image shown below:



```
Struct_1 (STRUCT)
0001 TYPE Struct_1:
0002 STRUCT
0003 pt1:ARRAY [0..2] OF INT;
0004 pt2:ARRAY [0..1][0..3] OF WORD;
0005 pt3:ARRAY [0..2][0..5][0..3] OF BOOL;
0006 END_STRUCT
0007 END_TYPE
```

The naming rules of a structure and an element are stated as below.

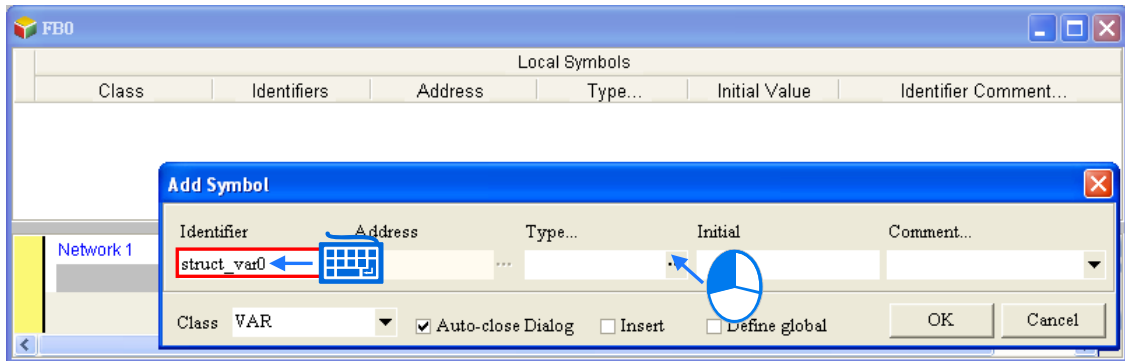
- An identifier cannot be the same as names of the user-defined data types. The repetition of a structure identifier is not allowed.
- The repetition of an element identifier in the same structure is not allowed.
- The identifiers of a structure and an element can be composed of 30 characters at most, and a Chinese character takes two characters.
- It is not allowed to use 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 names of a structure or an element, it is then a legal name. For example, "M0" is an illegal name, but "\_M0" is a legal name.
- The identifiers of a structure and an element cannot contain space. For example, "INPUT CH0" is an illegal name.
- The identifiers of a structure and an element can contain underline but it is not allowed to use 2 underlines in a row or put it in the end. For example, "INPUT\_CH0" is acceptable, but "INPUT\_\_CH0" and "INPUT\_CH0\_" are illegal.
- The identifier of a structure and an element cannot contain special characters, including \*, #, ?, \, %, @ and more.



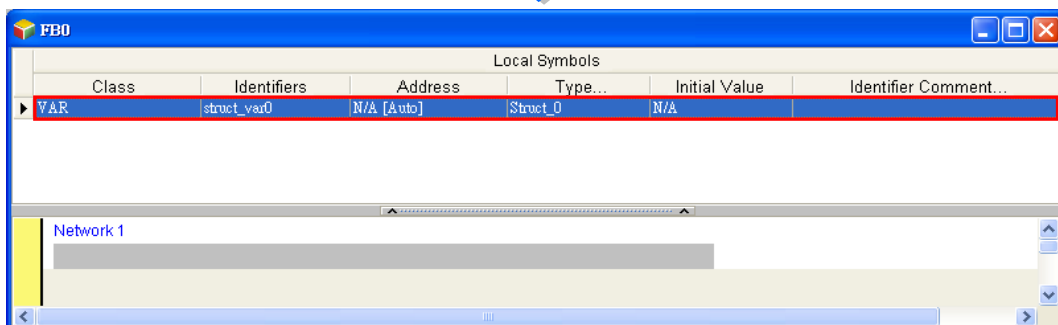
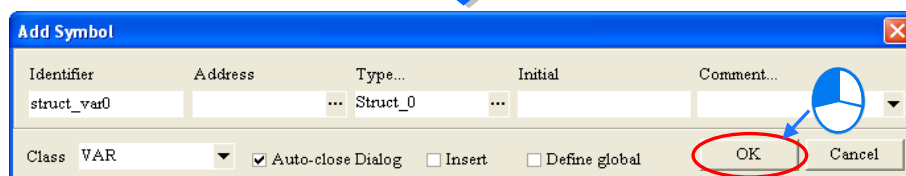
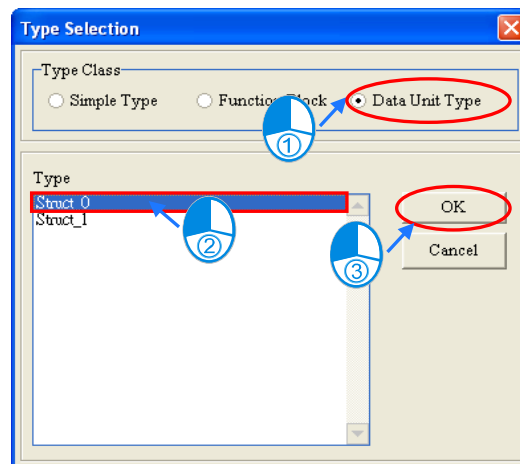


### 8.2.3 Using a Structure Variable

After the creation of the structure type is done, users can declare the variable symbol in the Local Symbols page. Open the Add Symbol window from the Global and local symbol table and input the identifier and then click ... on the right as the image shown below.

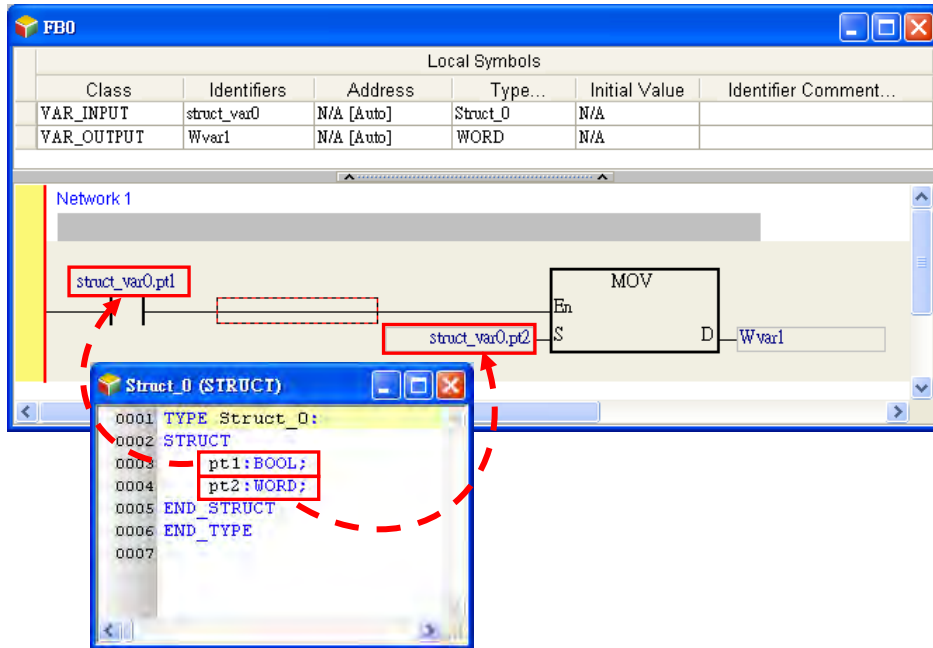


Select the Data Unit Type, and the declared structure data type and then click OK to add a new symbol.

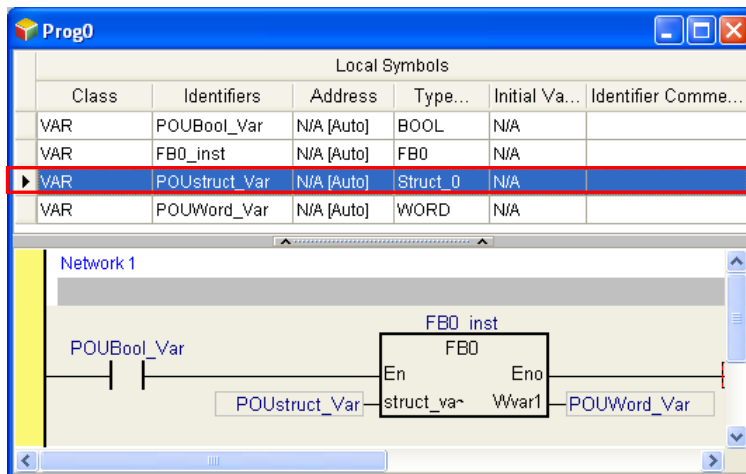


### 8.2.4 Applications of Structures

After the structure and symbol are declared, they can be used in the POU. The data type of the symbol **struct\_var0** is **Struct\_0** in the function block as the image shown below. Users can use “.” to assign the element for the structure symbols, for example **struct\_var0.pt1** means the first element pt1 is assigned as a contact to the data type BOOL operand and **struct\_var0.pt2** means the second element pt2 is assigned as a pin to the data type WORD operand as the image shown below.

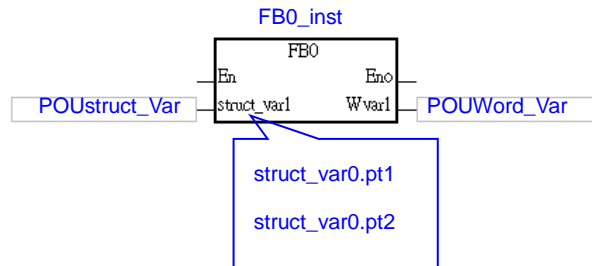


When using the aforementioned function block, Struct\_0 data type symbol should be declared in the POU, so that Struct\_0 data type input pin and output pin can be consistent. The elements with the same structure can use “.” to assign the element for the structure type symbols, for example POUstruct\_var.

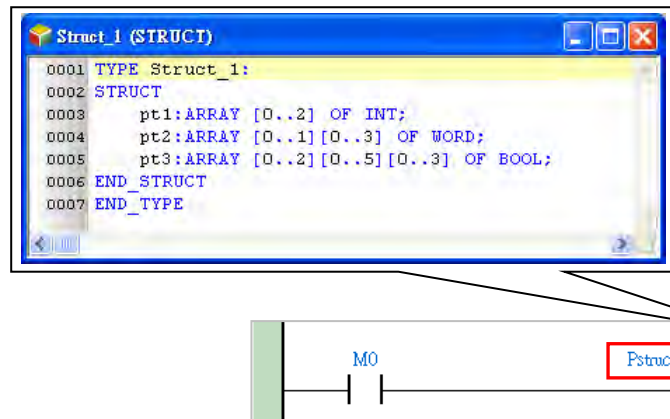


8

From the description above, we know that when the structure type symbol **POUstruct\_Var** is used in the FB0, it also delivers all the elements from the symbol for **struct\_var0.pt1** and **struct\_var0.pt2** to operate. Thus users can define the structure type symbols as the input/output pin for the function block to deliver all the elements from the symbol. So that the pin defining time can be saved and the program can be simpler.

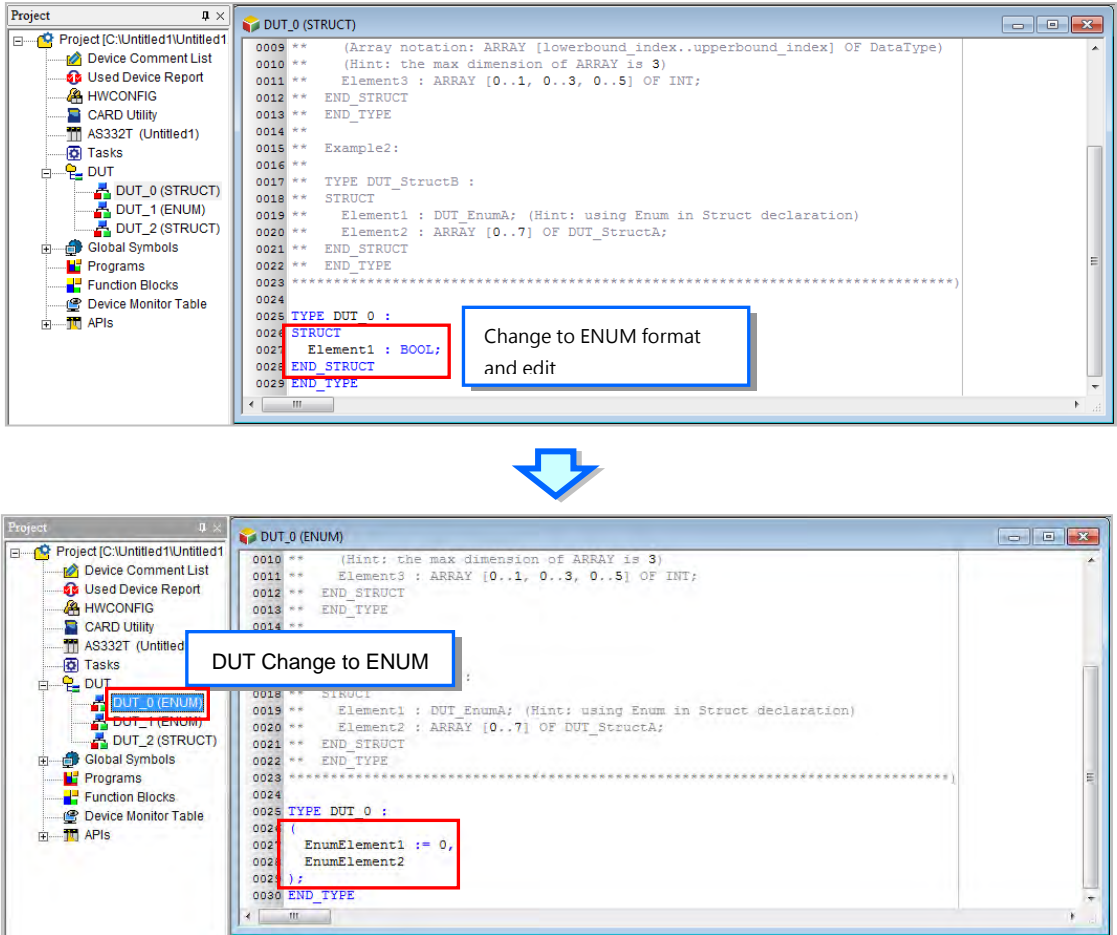


When the structure is constructed with an array element, the variables in the structure can be written as the example shown below: the status of M0 can be written in the [1,1,1] of the appointed element pt3 in the variable Pstruct\_var1.



### 8.2.5 Change Structure to Enumeration

Users can change a DUT from **Structure** to **Enumeration** by editing the structure names of STRUCT and END STRUCT to “ ( ” and “ ) ;” as well as the element names from STRUCT to ENUM format, please refer to section 8.3 for more information.



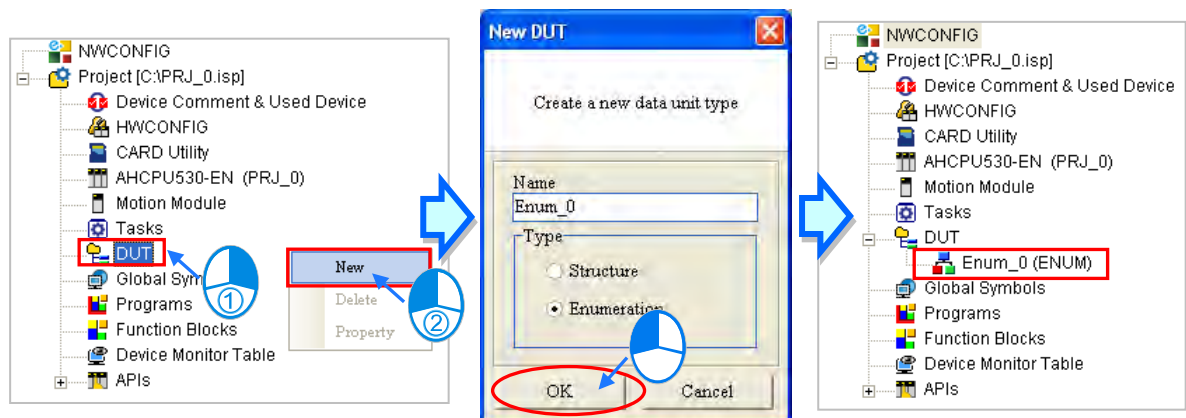
## 8.3 Enumeration

### 8.3.1 Definition of Enumeration

Users can enumerate (obtain one at a time) the elements in a collection of objects, providing an efficient way to define a set of named integral constants that may be assigned to a variable and values that the enumeration represents.

### 8.3.2 Creating an Enumeration

Right-click DUT (Data Unit Type) in the project management area to see the option New. Click New to bring out a New DUT window. Input an enumeration name, select the type as Enumeration and then click OK to confirm the setting. After that, a new DUT with name and type will appear under the DUT option.



\*. The naming rules of an enumeration can be found below.

After an enumeration DUT is created, an editing window will be presented with examples and definitions for reference. Users can edit the examples to make them as their own enumerations.

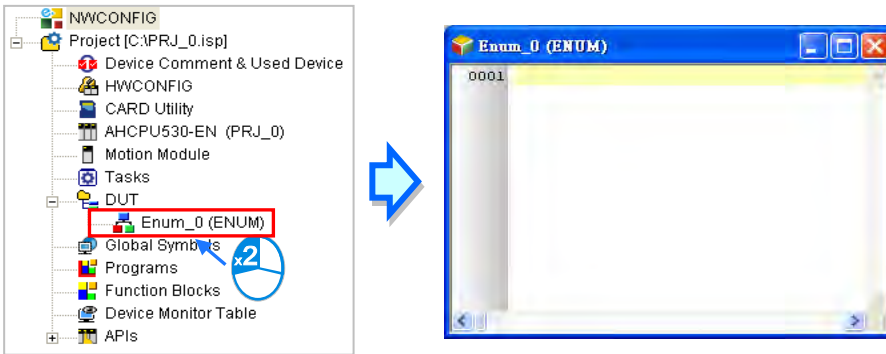
```

Enum_0 (ENUM)
0001 |-----|
0002 ** Example1:
0003 **
0004 ** TYPE DUT_EnumA :
0005 ** (
0006 **   Red := 0,
0007 **   Blue := 2,
0008 **   Green := 3
0009 ** );
0010 ** END_TYPE
0011 **
0012 ** Example2:
0013 **
0014 ** TYPE DUT_EnumB :
0015 ** (
0016 **   Red := 0, (Hint: allowable to set or not to set the Enum value)
0017 **   Blue,
0018 **   Green
0019 ** );
0020 ** END_TYPE
0021 |-----|
0022
0023 TYPE Enum_0 :
0024 (
0025   EnumElement1 := 0,
0026   EnumElement2
0027 );
0028 END_TYPE

```

8

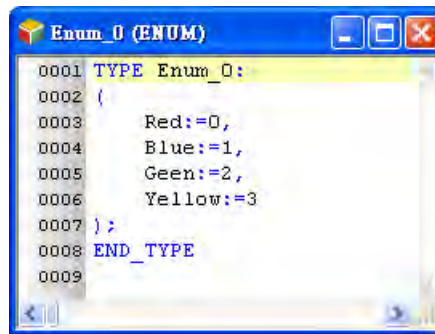
Users can find it under the DUT option in the project management area. Double click the new DUT to open it and start to add the contents of the newly added DUT.



The enumeration format is stated as below: wordings in blue are the system keywords; wordings in black are the user-defined enumeration names, element names and values.

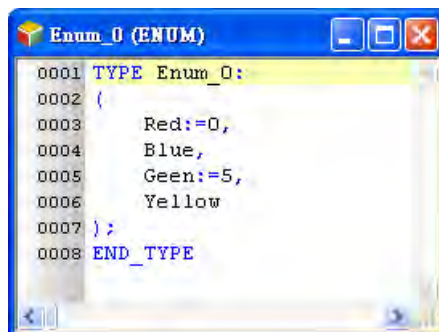
**TYPE <Enumeration Name>:**

```
(
  < Element 1 Name > := <Value 1>,
  < Element 2 Name > := <Value 2>,
  ...
  < Element n Name > := <Value n>
);
```



**END\_TYPE**

- **Enumeration Name:** Define the enumeration name. It is recommended to use the same name as the name shown on the project management area.
- **Element Name:** Define the element name.
- **Value:** Define values for every element, ranging -32768~32767. It is acceptable to leave this field blank. The system will use defaults according to the previous element value and avoid duplications. The system will set the value of the element Blue to 1 and Yellow to 6 as the image shown below.

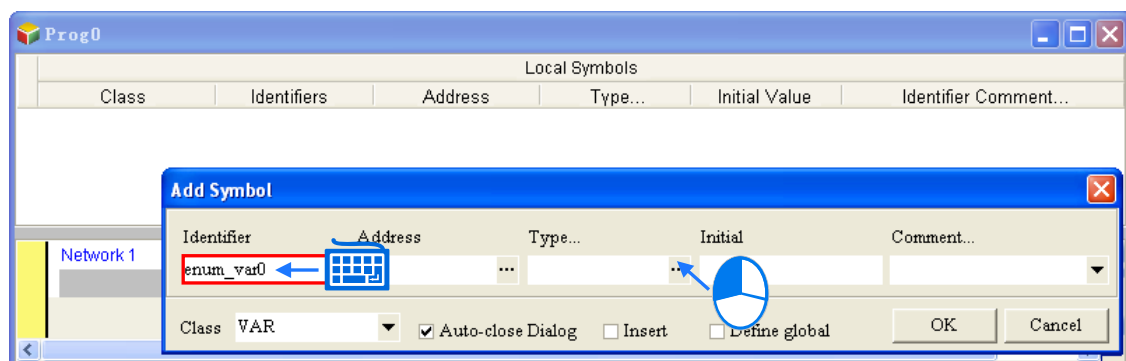


The naming rules of an enumeration and an element are stated as below.

- An identifier cannot be the same as names of the user-defined data types. The repetition of an enumeration identifier is not allowed.
- The repetition of an element identifier in the same enumeration is not allowed.
- The identifiers of an enumeration and an element can be composed of 30 characters at most, and a Chinese character takes two characters.
- It is not allowed to use 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 names of a structure or an element, it is then a legal name. For example, "M0" is an illegal name, but "\_M0" is a legal name.
- The identifiers of an enumeration and an element cannot contain space. For example, "INPUT CH0" is an illegal name.
- The identifiers of an enumeration and an element can contain underline but it is not allowed to use 2 underlines in a row or put it in the end. For example, "INPUT\_CH0" is acceptable, but "INPUT\_\_CH0" and "INPUT\_CH0\_" are illegal.
- The identifier of an enumeration and an element cannot contain special characters, including \*, #, ?, \, %, @ and more.
- The data size of enumeration variables are the same as INT data size.

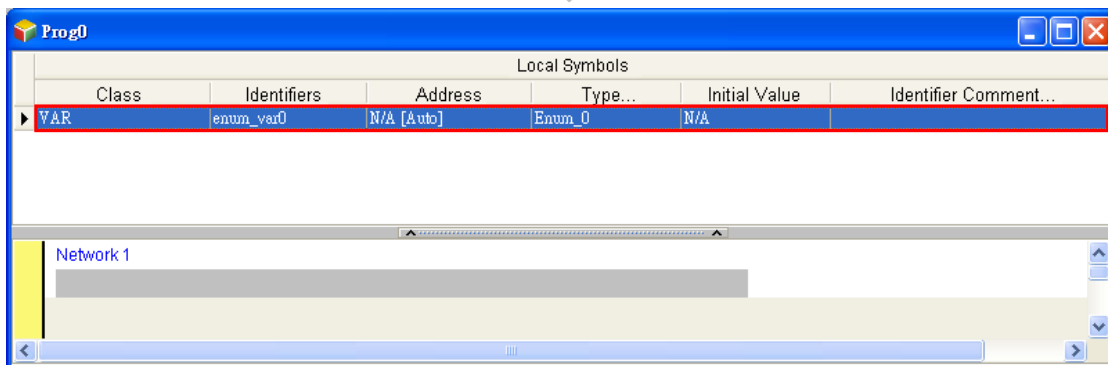
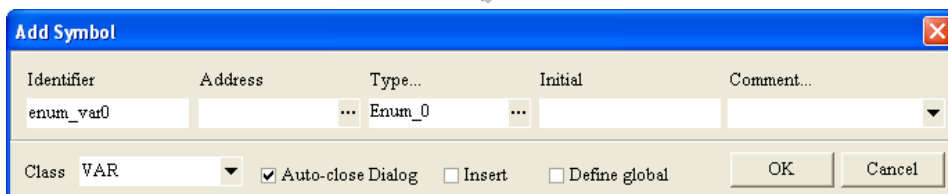
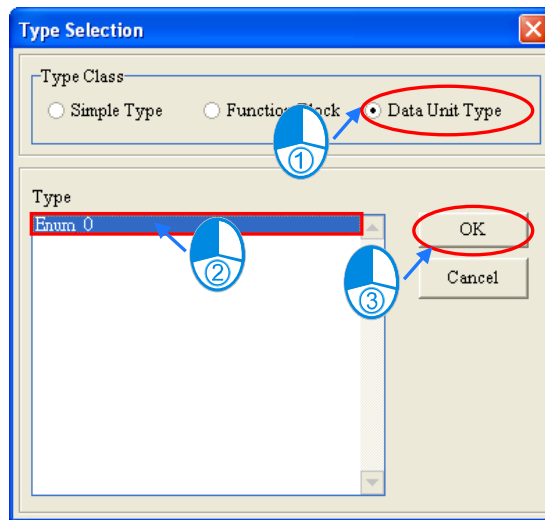
### 8.3.3 Using an Enumeration Variable

After the creation of the enumeration type is done, users can declare the symbol in the Local Symbols page. Open the Add Symbol window from the Global and local symbol table and input the identifier and then click ... on the right as the image shown below.



8

Select the Data Unit Type, and the declared structure data type and then click OK to add a new symbol.



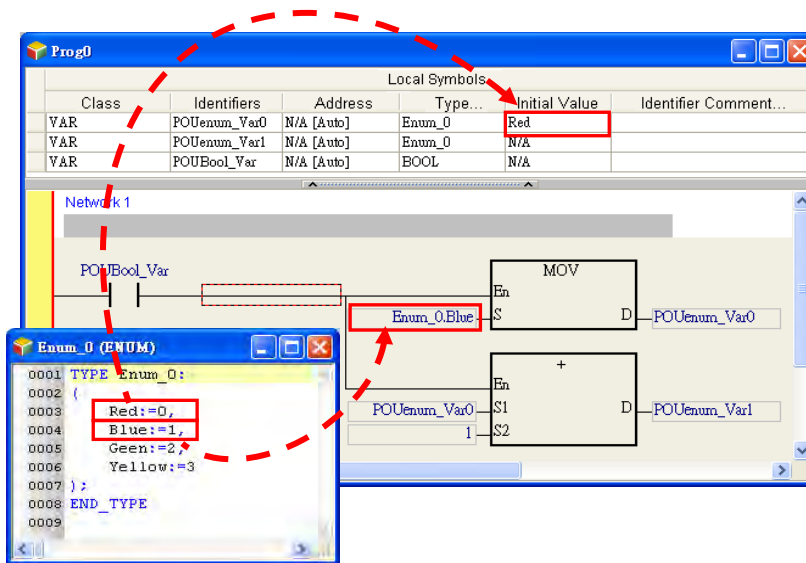
8



### 8.3.4 Applications of Enumerations

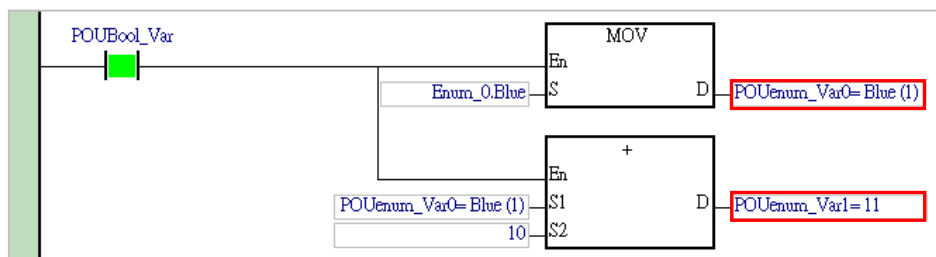
After the enumeration and symbol are declared, they can be used in the POU. The data type of the symbol **POUenum\_Var0** is Enum\_0. The Initial Value can be set by the element name or values. The image below shows the element name, Red.

Users can use “.” to assign the element for the enumeration symbols, for instance, users can assign **Enum\_0.Blue** to **POUenum\_Var0**, the symbol Enum\_0, and execute the instruction MOV; the defined element or value will be the enumeration value and after the execution of MOV, the value of POUenum\_Var0 will be 1.



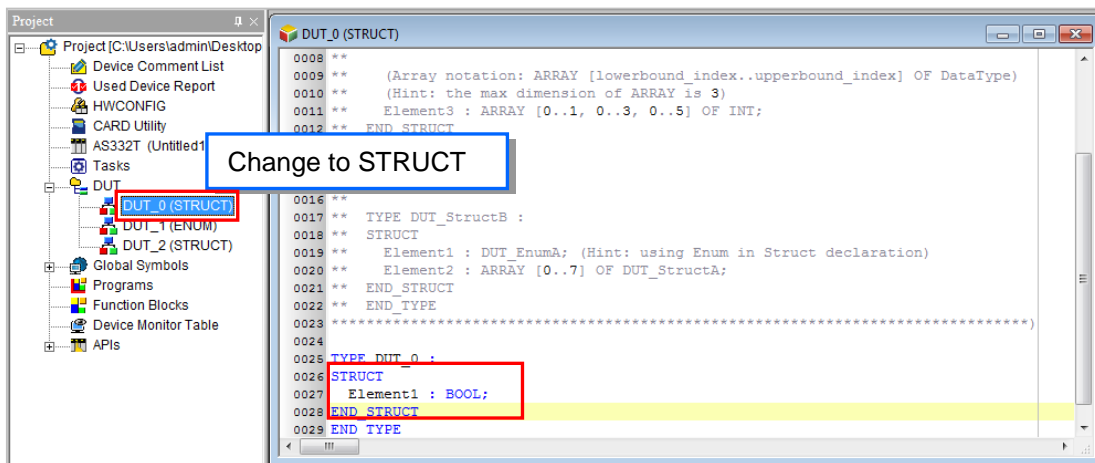
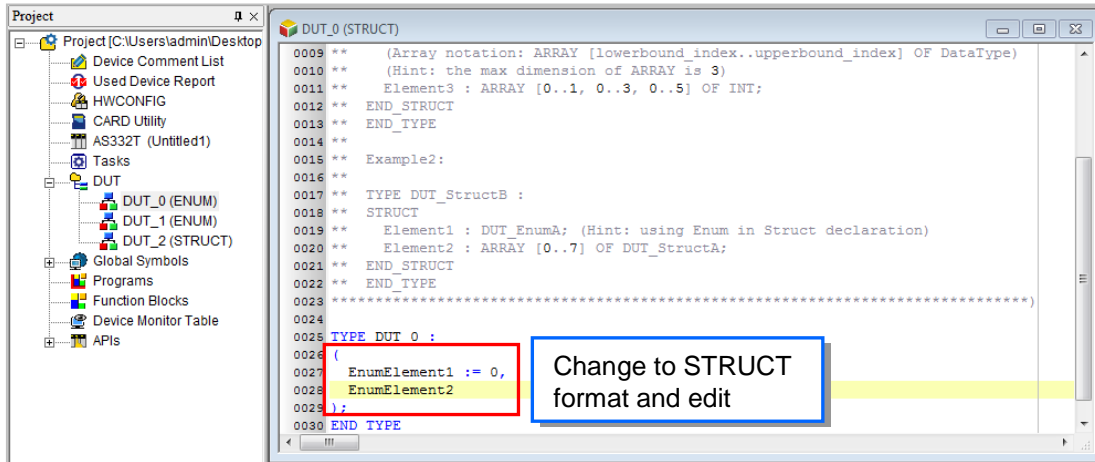
Users can also perform numerical operations; using the instruction + and set the value to 1, POUenum\_Var0 will be incremented by 1 as the example shown above. When the value of POUenum\_Var0 is 1, after executing the instruction +, the value result will be 2 and its element will be Green as defined.

When running and monitoring the program, the enumeration symbols will show their current defined element names. For example when the value is 1, the POUenum\_Var0 will be seen as POUenum\_Var0=Blue(1) as they are defined in the following image. But if the value of the enumeration symbol is not in the defined element range, the value will be shown as it is; as the following example shows when the value is 11 but 11 is not in the defined range, POUenum\_Var1=11 will be stated as the image shown below.



### 8.3.5 Change Enumeration to Structure

Users can change a DUT from **Enumeration** to **Structure** by editing “ ( ” and “ ) ; ” to structure names of STRUCT and END STRUCT as well as the element names from ENUM to STRUCT format, please refer to section 8.2 for more information.





# Chapter 9. Axis

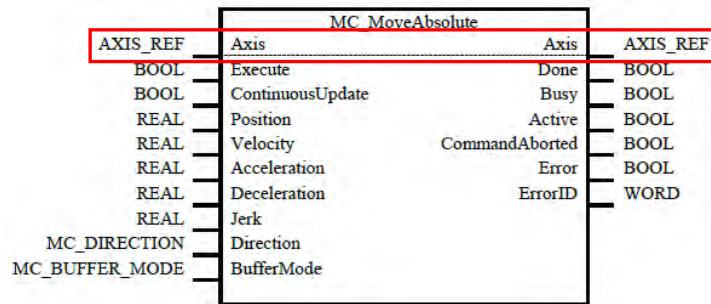
## Table of Content

<b>9.1</b>	<b>Axis</b> .....	<b>9-2</b>
9.1.1	The Meaning of Axis .....	9-2
9.1.2	Creating a New Axis .....	9-3
9.1.3	Main Table and Axis Table under Global Symbols.....	9-10
<b>9.2</b>	<b>Axis Parameter and Monitor &amp; Test Run</b> .....	<b>9-11</b>
9.2.1	Axis Parameter .....	9-11
9.2.2	Monitor & Test Run .....	9-21

## 9.1 Axis

### 9.1.1 The Meaning of Axis

As defined in the PLCopen MC standard, **Axis** on the software is the VAR\_IN\_OUT parameter of the STRUCTURE data type used in motion control function blocks. **Axis** provides a connection between the motion control in the program and the physical output device. Axis stores various motion control information such as position, velocity and so on. Below is an example of one function block defined in the PLCopen MC standard and AXIS\_REF being the axis defined in the standard.



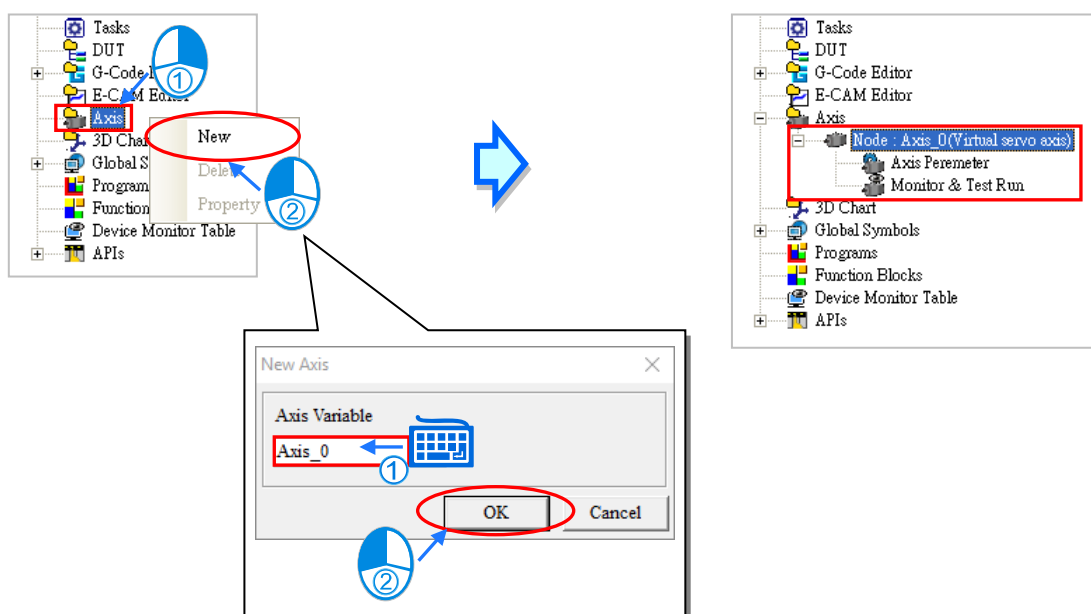
In the actual application, different motion function blocks are used to control an axis and then the target servo motor receives the output to achieve the motion control.

\*. Currently the DVPxxMC does not follow the principle for VAR\_IN\_OUT parameter of the STRUCTURE data type used in the Axis, but uses the VAR\_INPUT of the STRUCTURE data type instead.

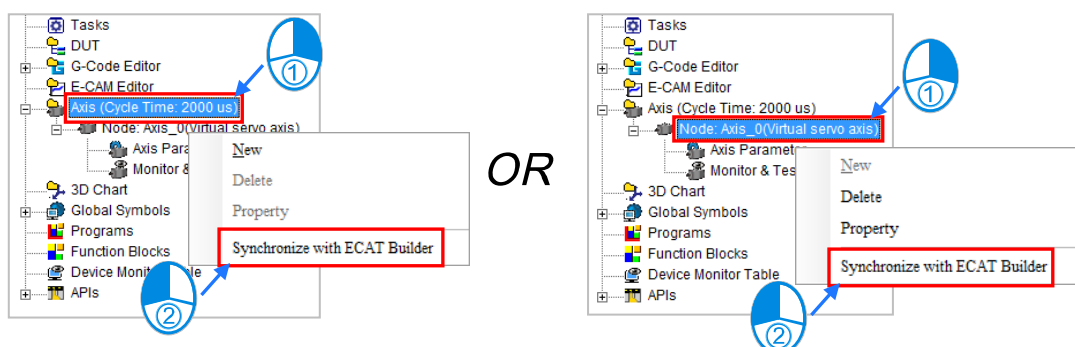
## 9.1.2 Creating a New Axis

### 9.1.2.1 AHxxEMC Series PLC - New Axis Settings

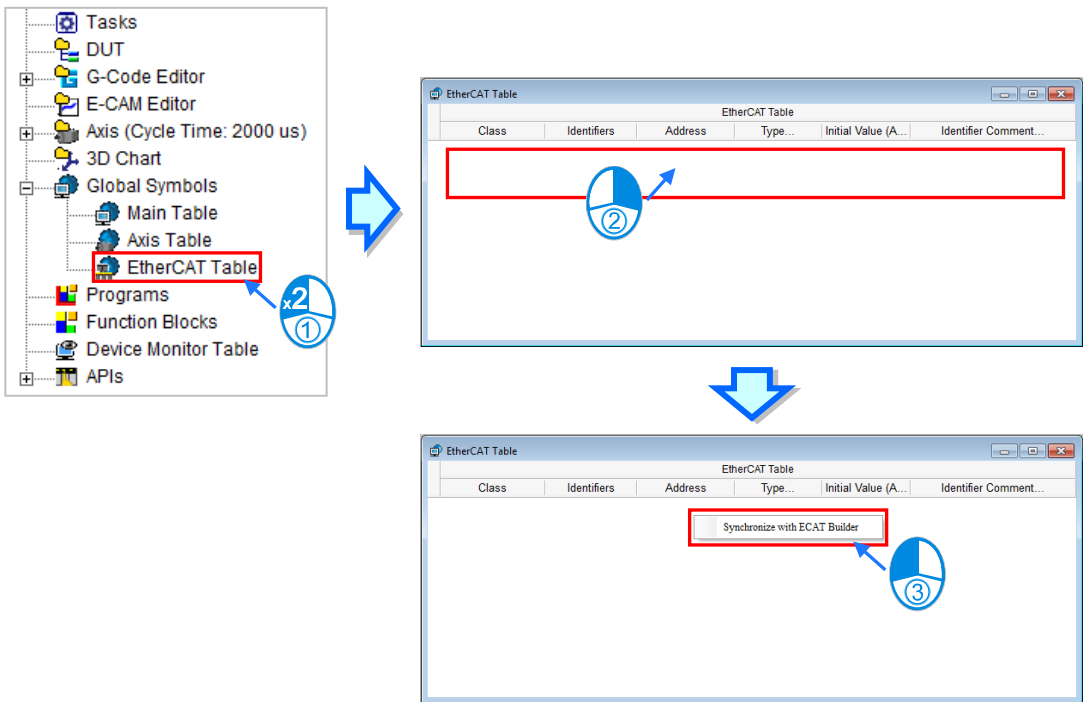
Right-click **New** from the context menu in Project area. The **New Axis** window will appear. Type an axis variable name under **Axis Variable** and click **OK** button in the following window to add the node item under **Axis** in the project management area. Make sure that the axis variable name should not be the same as other global symbol names or local symbol names. The **Cycle Time** shown next to the Axis in the project management is the cycle time of ECAT Builder master station, please refer to section **3.2 Master Setting** of the ECAT Builder User Manual for more information.



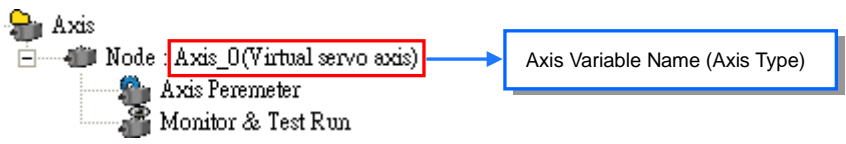
Users can synchronize the **Axis** information (including axis parameter, node information) which is saved in the ECAT Builder, then right-click the Axis in the project management area and select **Synchronize with ECAT Builder** on the context menu to display the synchronized information.



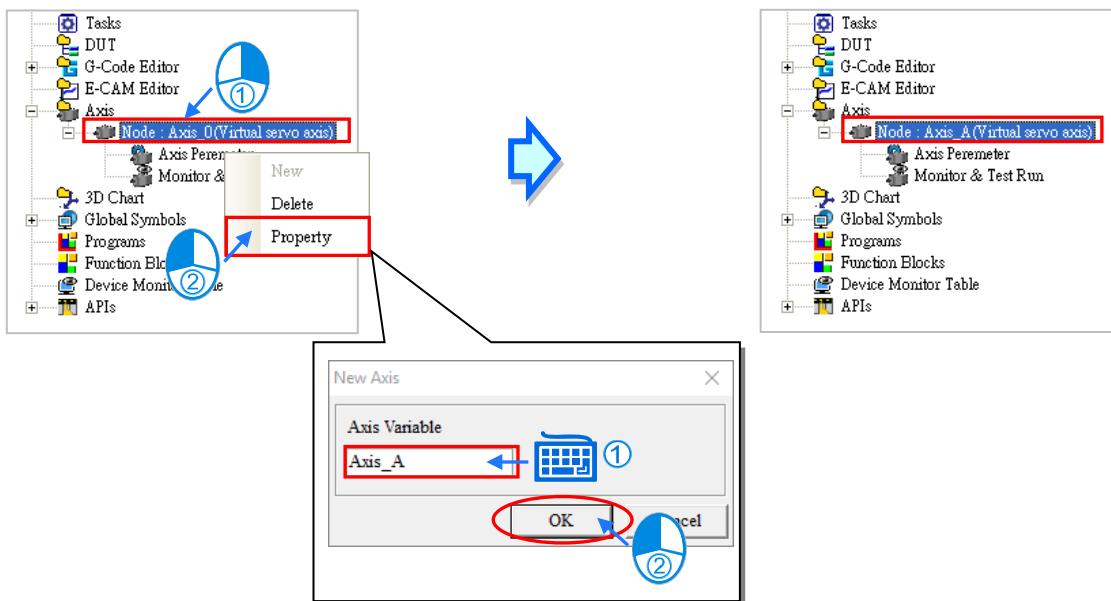
Users can also right-click the EtherCAT Table and select **Synchronize with ECAT Builder** on the context menu to view synchronized information of the ECAT Builder, Axis and EtherCAT Table.



The full name of the node item under **Axis** contains the axis variable name and axis type mentioned above. The axis type here which is the same as that in the parameter setting can be the EtherCAT slave name and the station address of a virtual or real axis. There are two sub-items, **Axis Parameter** and **Monitor & Test Run** under every node. Refer to the following sections for more information.

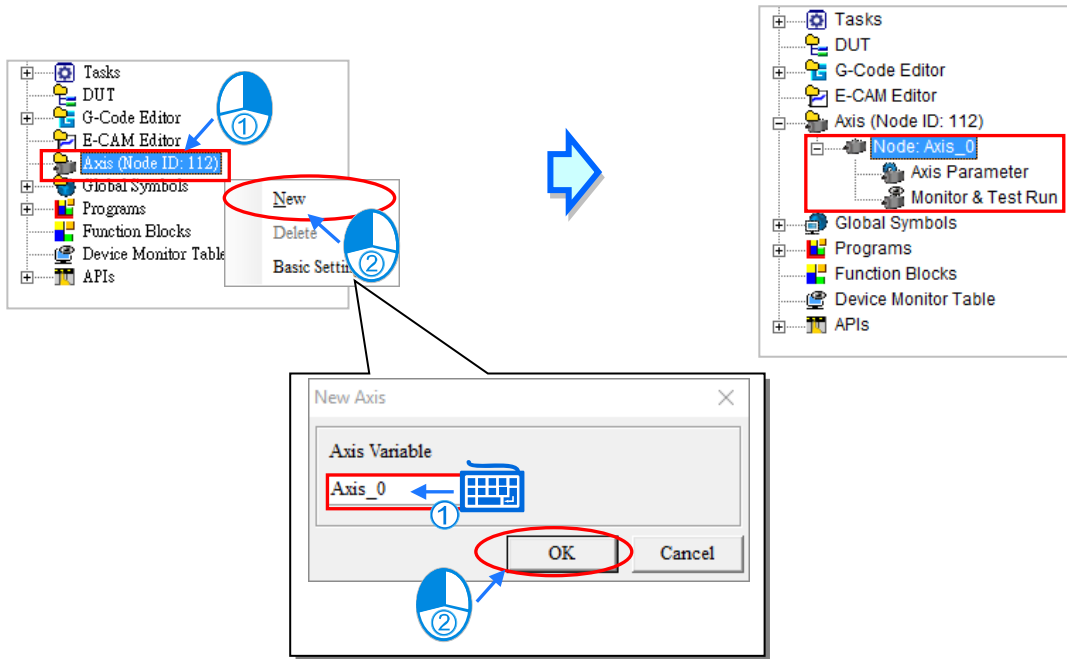


Right-click the axis which has been built and then select **Delete** button on the context menu to delete the axis or select **Property** button to reset the axis variable name.

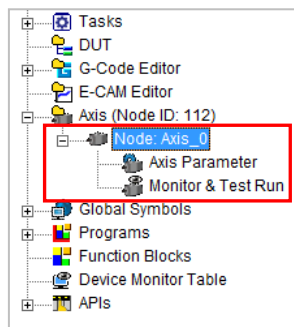


### 9.1.2.2 DVPxxMC – Create Axis & Basic Settings

Right-click **Axis** in the Project area and then click **New** on the context menu. The **New Axis** window will appear. Type an axis variable name under **Axis Variable** and click **OK** button in the following window to add the node item under **Axis** in the project management area. Currently, the DVPxxMC provides maximum number of 32 axes and supports up to 24 real axes. For settings on real axis and virtual axis, please refer to section 9.2.1.2. Make sure that the axis variable name should not be the same as other global symbol names or local symbol names. On the right side of the Axis items is **Node ID:112**, it is the station number of motion controller, default setting is 112 and cannot be changed.

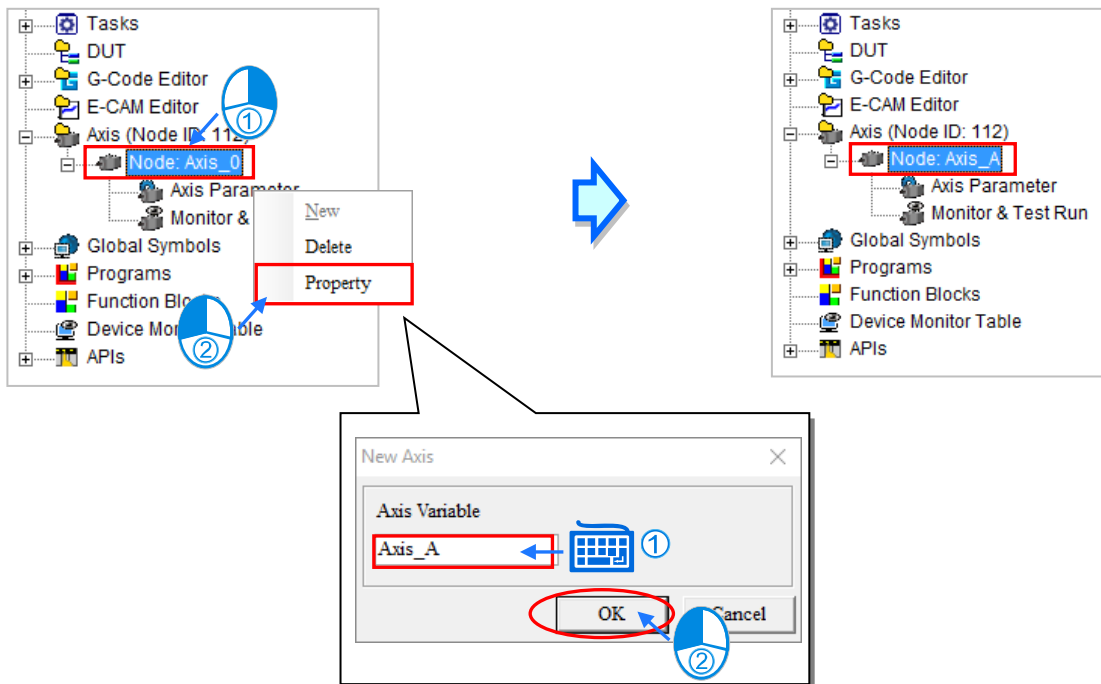


The Axis name is shown next to the Node item and contains two sub-items- **Axis Parameter** and **Monitor & Test Run**. Please view the following chapters for more details.

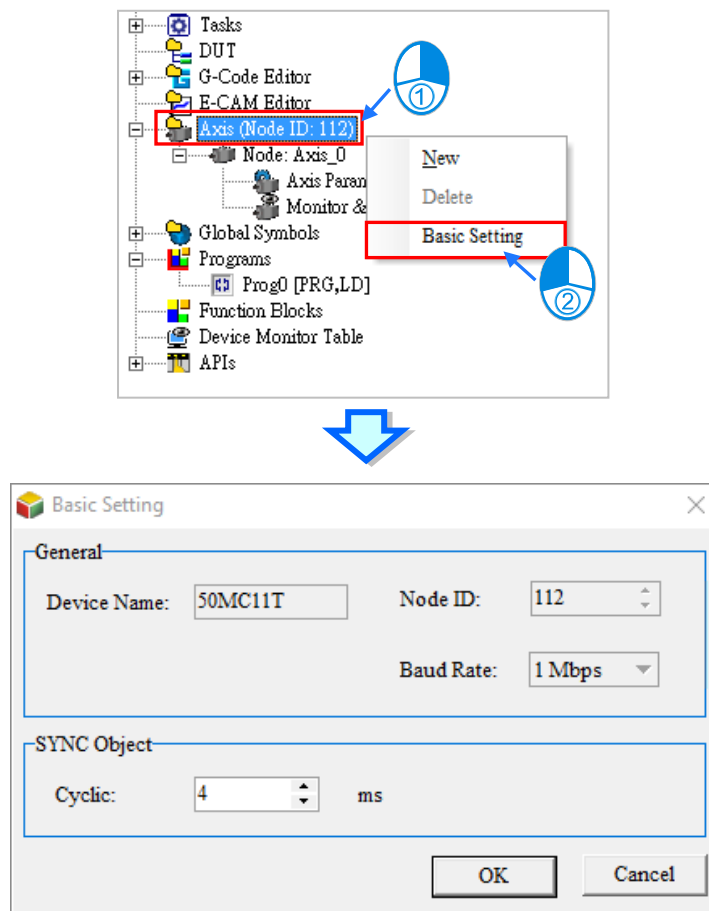




Right-click the axis which has been built and then select **Delete** button on the context menu to delete the axis or select **Property** button to reset the axis variable name.

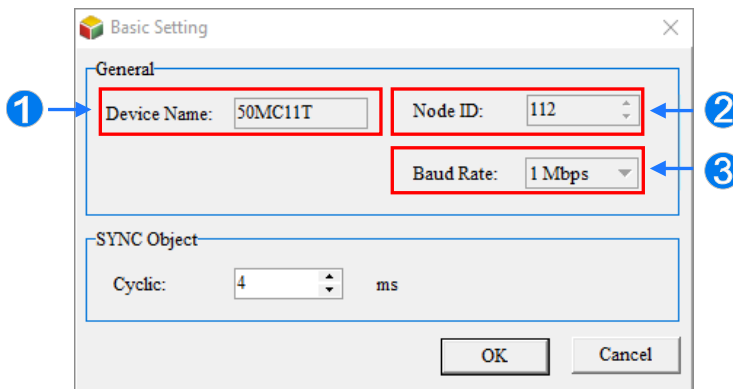


Right-click the Axis to select **Basic Setting** for adjusting axis parameter settings.



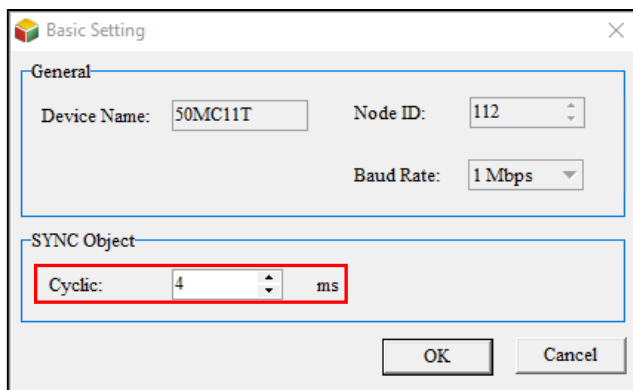
The Basic Setting window contains two items- **General** and **SYNC Object** and descriptions are introduced below.

● **General**



- ❶ Device Name: Device name for project by default and cannot change.
- ❷ Node ID: Default is 112 and cannot change.
- ❸ Baud Rate: The baud rate of CANMotion in HWCONFIG. For more information on CANMotion baud rate setting, please refer to section 3.6.2.

● **SYNC Object**



**Cyclic:** Represents setting the synchronous cycle time for the master to transmit a new datagram. This is an important motion control parameter. If the cycle time is set wrong, it will cause alarm AL303/AL302/ AL301 or errors in servo drives during the network communication cycle.

9

The Cyclic starts by scanning the motion control program and use algorithm to calculate the motion data and send the information to all axes. It can be identified as a total time for motion programming and communication between DVPxxMC and all servo drives.

The maximum execution time for motion programming is set as the execution time for motion programming time (unit: ms). Since 1000 microseconds equals to 1 millisecond. It is suggested to round up the unit (ms) for applications. For example, the maximum execution time for motion programming is 2567 microseconds and equals to 2.5 milliseconds. Thus, the execution time is rounded to 3 milliseconds.

The estimated communication time between DVPxxMC and servo drives is 0.5 milliseconds per axis. It is suggested to round up the unit (ms) for applications. For example, the communication time for five servo drives is  $5 \times 0.5$  milliseconds and equals to 2.5 milliseconds. Thus, the execution time is rounded to 3 milliseconds.

Calculation formula: Cyclic (ms) = round the max. execution time (ms) + communication time between DVPxxMC and servo drives (ms) + 1 (reserved time for program change) (ms).

For program change, the reserved time is set between 1 to 2 milliseconds so that the synchronous time setting is not affected by increased programming time.

Example:

The maximum execution time for five servo drives is 1634 microseconds. The reserved time for program change is 1 millisecond.

Cyclic = 2 milliseconds (round the max. execution time of 1634 microseconds to an integer) + 3 milliseconds (round  $5 \times 0.5$  to an integer) + 1 millisecond (reserved time to program) = 6 milliseconds.

For DVPxxMC series, the following table provides axis related system structure:

Member Variable	Data Type	Function Description
CmdPos	LREAL	Command position (unit: unit)
CmdVel	LREAL	Command velocity (unit: unit/sec)
CmdAcc	LREAL	Command acceleration (unit: unit/sec)
CmdTrq	INT	Command torque (per thousand (1‰) of rated torque)
ActPos	LREAL	Actual position (unit: unit)
ActVel	LREAL	Actual velocity (unit: unit/sec)
ActTrq	INT	Actual torque (per thousand (1‰) of rated torque)
ActCur	INT	Actual current

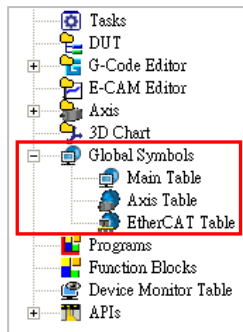
### 9.1.3 Main Table and Axis Table under Global Symbols

For management of the symbols by classification, there are different sub-items including **Main Table**, **Axis Table** and other network symbol table under **Global Symbols**.

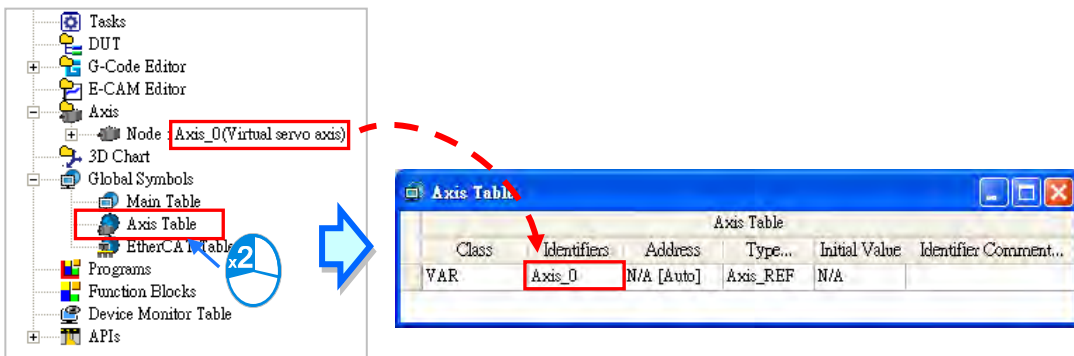
**Main Table**, the global symbols of other models, is used for editing the global symbols.

**Axis Table** lists all axis variable names which have already been created.

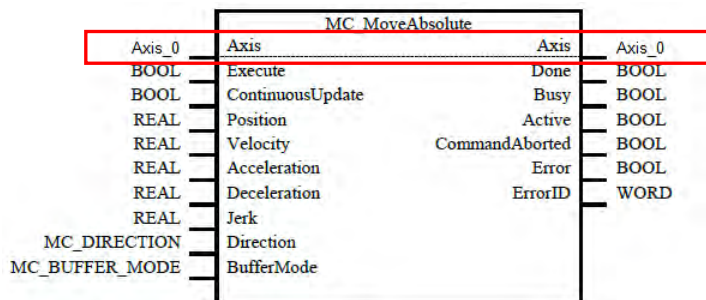
Other network symbol table contains the symbols in other network configuration software such as ECAT Builder supported by the model. For details on other network configuration software, refer to operation manuals of the software.



After the axis is created, the identifier of the axis variable name will appear automatically in **Axis Table** under **Global Symbols** and the axis symbol table cannot be edited. Double-click **Axis Table** to open the **Axis Table** window.



The axis variable symbol can be used for the pins of PLCopen MC function blocks and Delta function blocks as the example shown below.



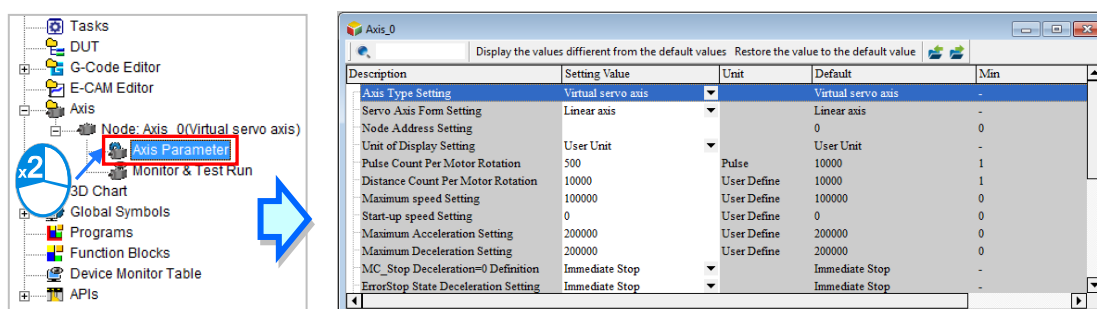
## 9.2 Axis Parameter and Monitor & Test Run

### 9.2.1 Axis Parameter

#### 9.2.1.1 AHxxEMC Series Parameter Settings

The parameter setting pop-up window **AxisParaForm** appears after the node item is created. Users can double-click **Axis Parameter** under the axis node item to open the **AxisParaForm** window once more after the window is closed. All settings on motion control of the axis are listed in the window as below. For details, refer to the user manuals of modules.

The set values can be downloaded to the PLC with the project together after the setting is done.



#### Axis Type Setting

Users can set the real axis which produces actual output to the drive assembly or the virtual servo axis on which the operation is done in the controller.

#### Servo Axis Form Setting

The axis can be set as **Linear axis** or **Rotary axis**. The position value will be constantly increased (or decreased) when the linear axis rotates toward the positive direction (or negative direction). The position value will be calculated again from minimum (maximum) value after the axis which rotates toward the positive direction (or negative direction) reaches the limit value.

#### Node Address Setting

If **Real axis** is selected in the **Axis Type Setting** box, the content in **Node Address Setting** must be chosen by the servo names and station addresses in ECAT Builder. It means that the axis is the set EtherCAT servo drive to which the actual output is delivered. The node address is selected as the EtherCAT slave, ASDA-A2-E CoE Drive [5] (the station address) as below. The drop-down menu of the **Node Address Setting** is updated when the setting information is saved in the ECAT Builder and then close both ECAT Builder and HWCONFIG; Or the updated setting will take place when the **axis** information synchronizes with the ECAT Builder, please refer to **section 9.1.2** in this chapter for more detail.

Axis Type Setting	Servo axis
Servo Axis Form Setting	Linear axis
Node Address Setting	ASDA-A2-E CoE Drive[5]

### **Unit of Display Setting**

Users can select a system unit. The default unit is user unit.

### **Pulse Count Per Motor Rotation**

Users can set how many pulses the controller outputs for a motor to rotate.

### **Distance Count Per Motor Rotation**

Users can set the distance in an actual mechanism move whenever a motor rotates.

### **Maximum speed setting**

Users can set the upper limit of the speed of the axis.

### **Start-up speed setting**

Users can set the start-up speed for the axis to start running.

### **Maximum Acceleration Setting**

Users can set the upper limit of the acceleration of the axis.

### **Maximum Deceleration Setting**

Users can set the upper limit of the deceleration of the axis.

### **MC\_Stop deceleration = 0 Setting**

Users can set the axis stopping method, when an error occurs or an instruction is executed. The option **Immediate Stop** or **Deceleration Stop** can be selected.

### **Error Stop Deceleration Setting**

Users can set the axis stopping method, when the axis runs beyond the software limit set. The option **Immediate Stop** or **Deceleration Stop** can be selected.

## **9**

### **Software Limit Enable Setting**

Users can set whether to enable or disable the software limit control and the limit values depend on the following two parameters.

### **Axis software positive limit Setting**

Users can set the positive software limit value of the axis, which is only applied to the linear axis.

**Axis software negative limit Setting**

Users can set the negative software limit value of the axis, which is only applied to the linear axis.

**Rotary Maximum Position Setting**

Users can set the maximum position of the rotary axis which moves toward the positive direction. The position value will be calculated again from the minimum value after the axis reaches the limit value.

**Rotary Minimum Position Setting**

Users can set the minimum position of the rotary axis which moves toward the negative direction. The position value will be calculated again from the maximum value after the axis reaches the limit value.

**Gear Ratio Numerator**

Users can set the numerator value of the electronic gear ratio.

**Gear Ratio Denominator**

Users can set the denominator value of the electronic gear ratio.

**Homing Acceleration**

The acceleration speed while homing is performed.

**Homing Velocity**

The parameter is the speed while the homing is performed

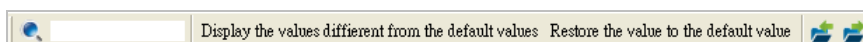
**Homing Closed Velocity**

While the homing is performed, the motor moves at the homing velocity at first. Then the motor starts to decelerate to this velocity and moves at this velocity when the trigger is close to the home signal.

**Permitted Deviation**

The permitted deviation between the actual position and the command position

The axis parameter setting window provides the following tool bar for editing the axis parameter table.



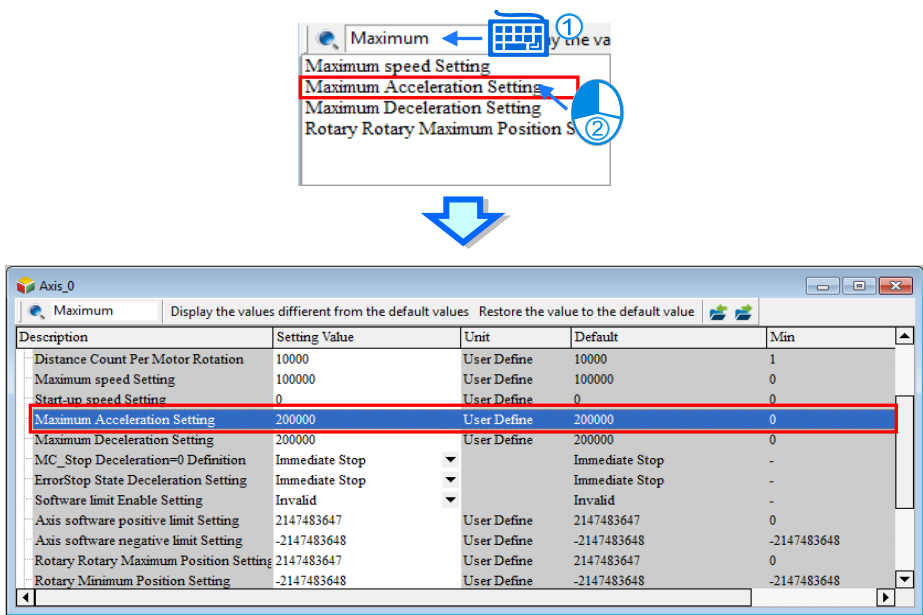
### Single Axis Start Immediately

Users can select **Disable** or **Enable**. When single axis is enabled, it will enable motion immediately and not be affected by the CPU cycle time. Please refer to AHxxEMC Series PLC manuals for more information.

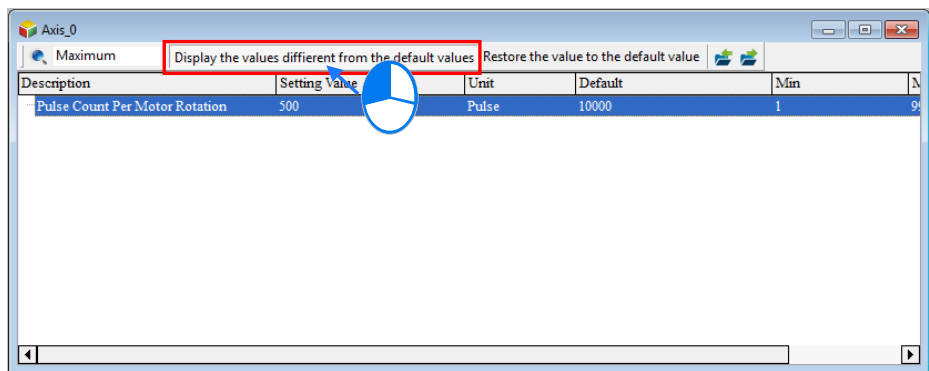
### MC\_CamIn Master Direction

Users can select **Positive Direction** or **No Direction**. Under **Positive Direction**, the main axis moves backwards and this will halt the slave; Under No Direction, the slave is not affected by the main axis moving forward or backwards.

**Search Tool:** After a character string is typed, relevant character strings will be listed automatically. If one item in the list of relevant character strings is selected, the corresponding line in the table will be selected accordingly.




After the button **Display the values different from the default values** is clicked, the parameter values different from the default values will be filtered out.




9



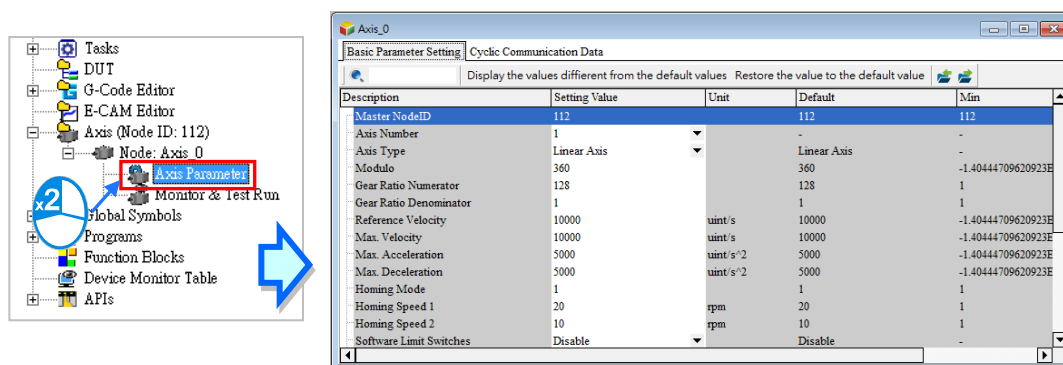
After the button **Restore the value to the default value** is clicked, all setting values will be restored to the default values.

If the button  is clicked, current parameters can be exported to the specified path in the file format of .CSV.

If the button  is clicked, the file at the specified path can be imported.

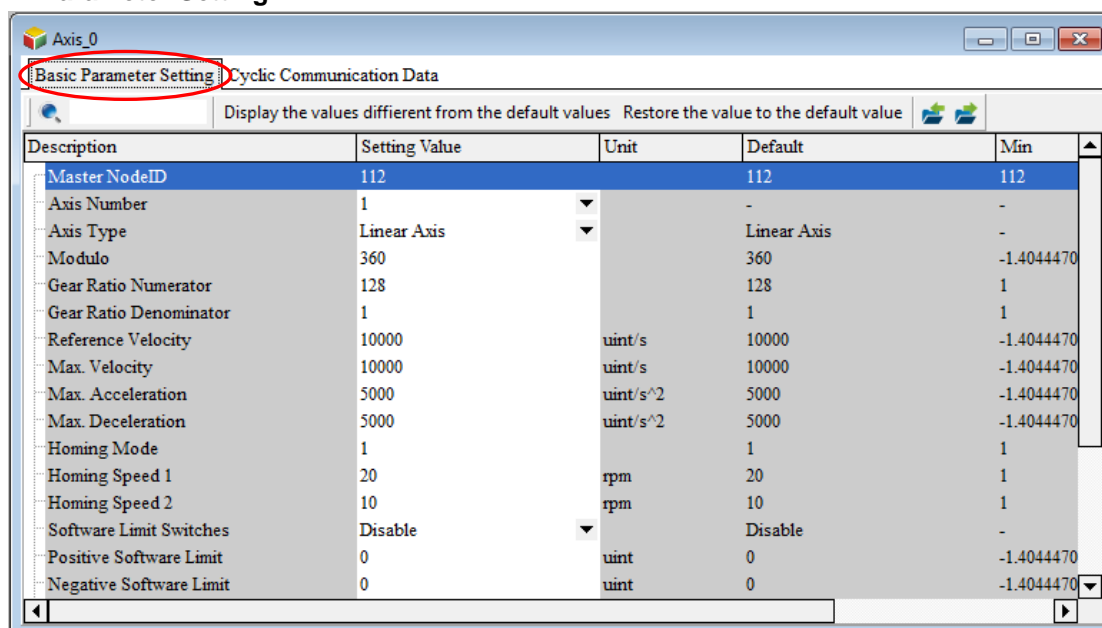
### 9.2.1.2 DVPxxMC Parameter Settings

For DVPxxMC, the **Basic Parameter Setting** pop-up window of the axis appears once the node item is created. Users can double-click **Axis Parameter** under the node item to open the window once more when closed.



The basic parameter setting window contains two items - **Basic Parameter Setting** and **Cyclic Communication Data**, please refer to PLC series manuals for more information. The parameter settings for projects can be downloaded to the CPU.

- **Parameter Setting**



Description of the Parameter Settings:

**Master Node ID**

The master node ID set as 112 is permanent.

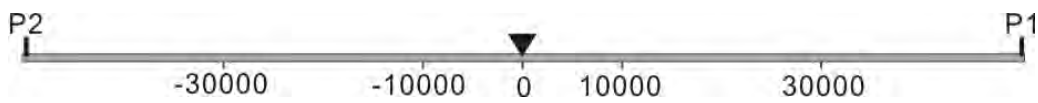
**Axis Number**

The setting value range is from 1~32.

**Axis Type**

The axis type include Linear Axis and Rotary Axis.

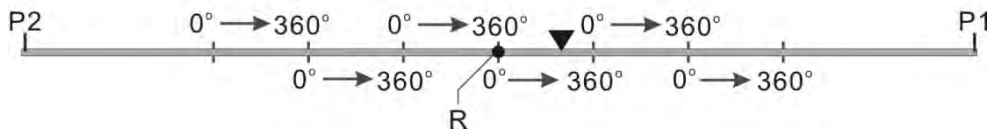
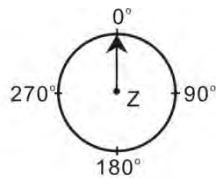
- Linear Axis Model



Note:

P1	Positive Limit
P2	Negative Limit
▼	Servo Position

- Rotary Axis Model ( "modulo" : 360)



Note:

P1	Positive Limit
P2	Negative Limit
▼	Servo Position
R	Homing Position
Z	Servo Motor Shaft

**Modulo**

The cycle is used for equally dividing the actual position of the actuator.

**E-gear Ratio Numerator**

This parameter and gear ratio denominator sets the motor pulse number per round.

**E-gear Ratio Denominator**

This parameter and gear ratio numerator sets the motor pulse number per round.

**Reference Velocity**

A temporary parameter that does not affect the functions of the controller.

**Max. Velocity**

It sets the max. velocity of axis connection to a device

**Max. Acceleration**

A temporary parameter that does not affect the functions of the controller.

**Max. Deceleration**

A temporary parameter that does not affect the functions of the controller.

**Homing Mode**

Users can set the homing mode range from 1~35. Please refer to DVP15MC manuals for more detail.

**Homing Speed 1**

The speed from homing to finding the homing switch

**Homing Speed 2**

The speed after finding the homing switch to machine homing.

**Software Limit Switches**

Users can select **Enable** or **Disable**. If software limit switch is enabled and the axis actuator position reaches positive limit or negative limit setting, motion instruction is wrong and stops

**Positive Software Limit**

It refers to the positive direction of an axis

**Negative Software Limit**

It refers to the negative direction of an axis

**Deviation Switch**

It sets the deviation detected between command position and feedback position. When the deviation switch is enabled, the controller detects the deviation between command position and feedback

position; if deviation switch is disabled, the controller will not detect for deviation between command and feedback positions.

**Permitted Deviation**

It refers to the deviation between the command position and feedback position

**Gearbox Input Rotation**

This parameter and gearbox output rotation decides the mechanical gear ratio.

**Gearbox Output Rotation**

This parameter and gearbox input rotation decides the mechanical gear ratio.

**Units Per Rotation**

The number of units which the actuator moves while gearbox output rotates for one circle.

**Axis Mode**


This parameter sets the **real axis** to read the actual position of servo components, the **virtual axis** to calculate by a controller and the **encoder axis** to perform with an encoder.

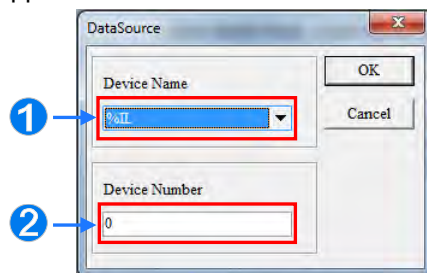
**Encoder Axis Mode**

Users can select **Encoder Axis** from the **Axis mode** and mode options include incremental encoder 1, incremental encoder 2, absolute encoder and data source.

Mode	Function
Incremental Encoder 1	The data source is derived from incremental encoder 1
Incremental Encoder 2	The data source is derived from incremental encoder 2
Absolute Encoder	The data source is derived from absolute encoder
DataSource	The data source is derived from assigned device data

**Encoder Axis Data Source**

Users can select **DataSource** from the **Encoder Axis Mode** and left-click  icon for DataSource setting window to appear as below.



- ❶ The drop-down list of Device Name include %IL, %QL and %ML.
- ❷ This section is for Device Number settings

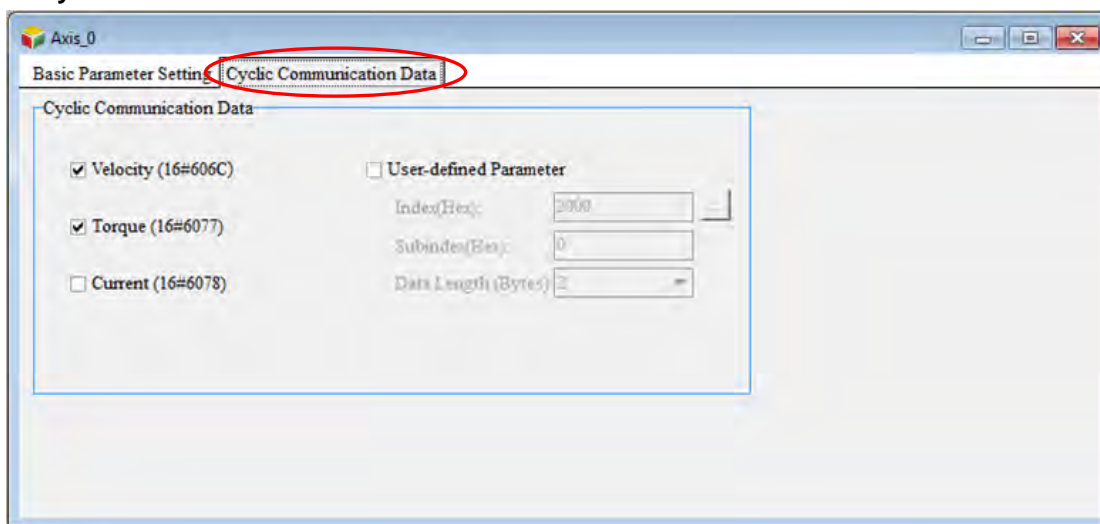
9

Under **DataSource**, the variation amount of the encoder axis position is the same as that of the variation amount for each scanning cycle of the assigned device value.

### EtherCAT Node

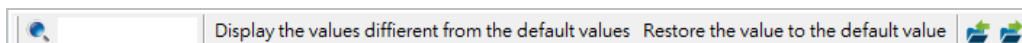
DVP50MC or DVP50MC-06 supports this item. The node number begins from 1001 to 1024.

- **Cyclic Communication Data**

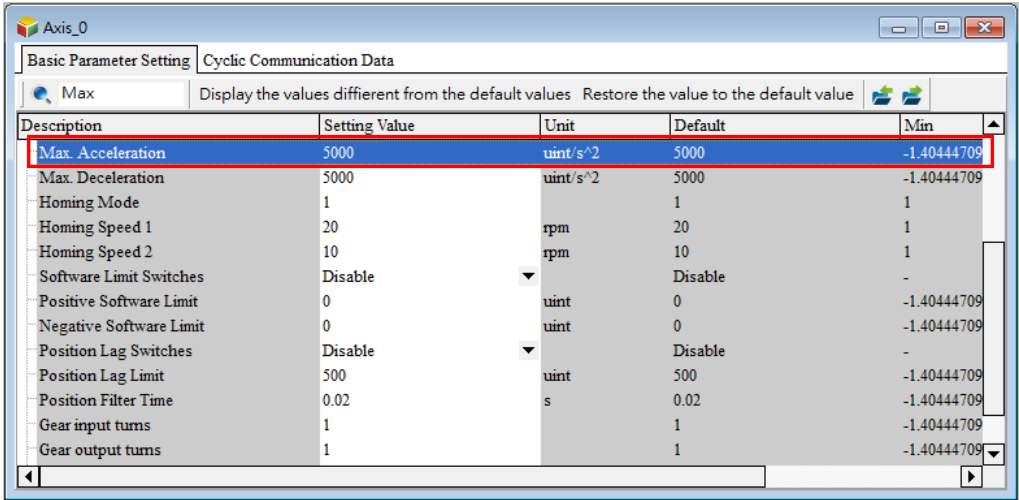
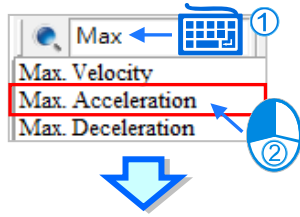


After users select Velocity, Torque, Current or User-defined Parameter, the servo drives operates in cyclic synchronous velocity and can transmit data length of up to 8 bytes regarding information data and cyclic communication data to the bus network. When the corresponding data are selected, users can monitor corresponding parameters from the Axis status.

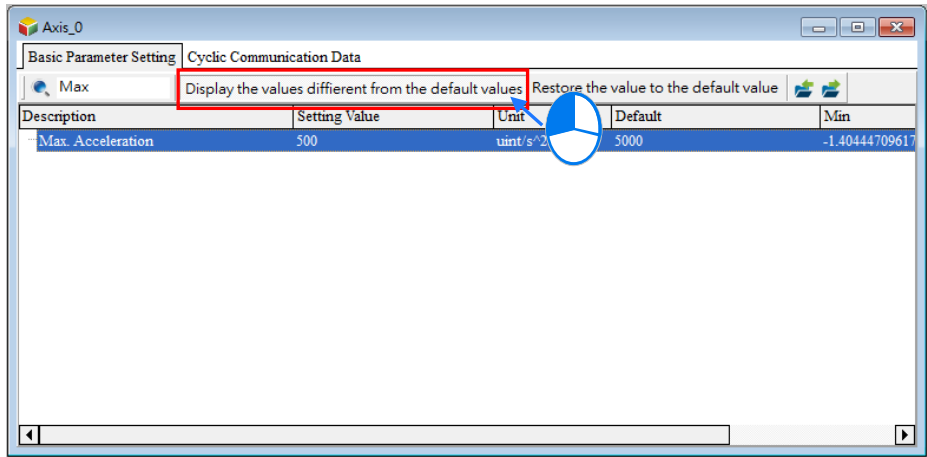
The axis parameter settings window provided the following toolbar for editing the axis parameter table.



**Search Tool:** After a character string is typed, relevant character strings will be listed automatically. If one item in the list of relevant character strings is selected, the corresponding line in the table will be selected accordingly.





After the button **Display the values different from the default values** is clicked, the parameter values different from the default values will be filtered out.



After the button **Restore the value to the default value** is clicked, all setting values will be restored to the default values.

9

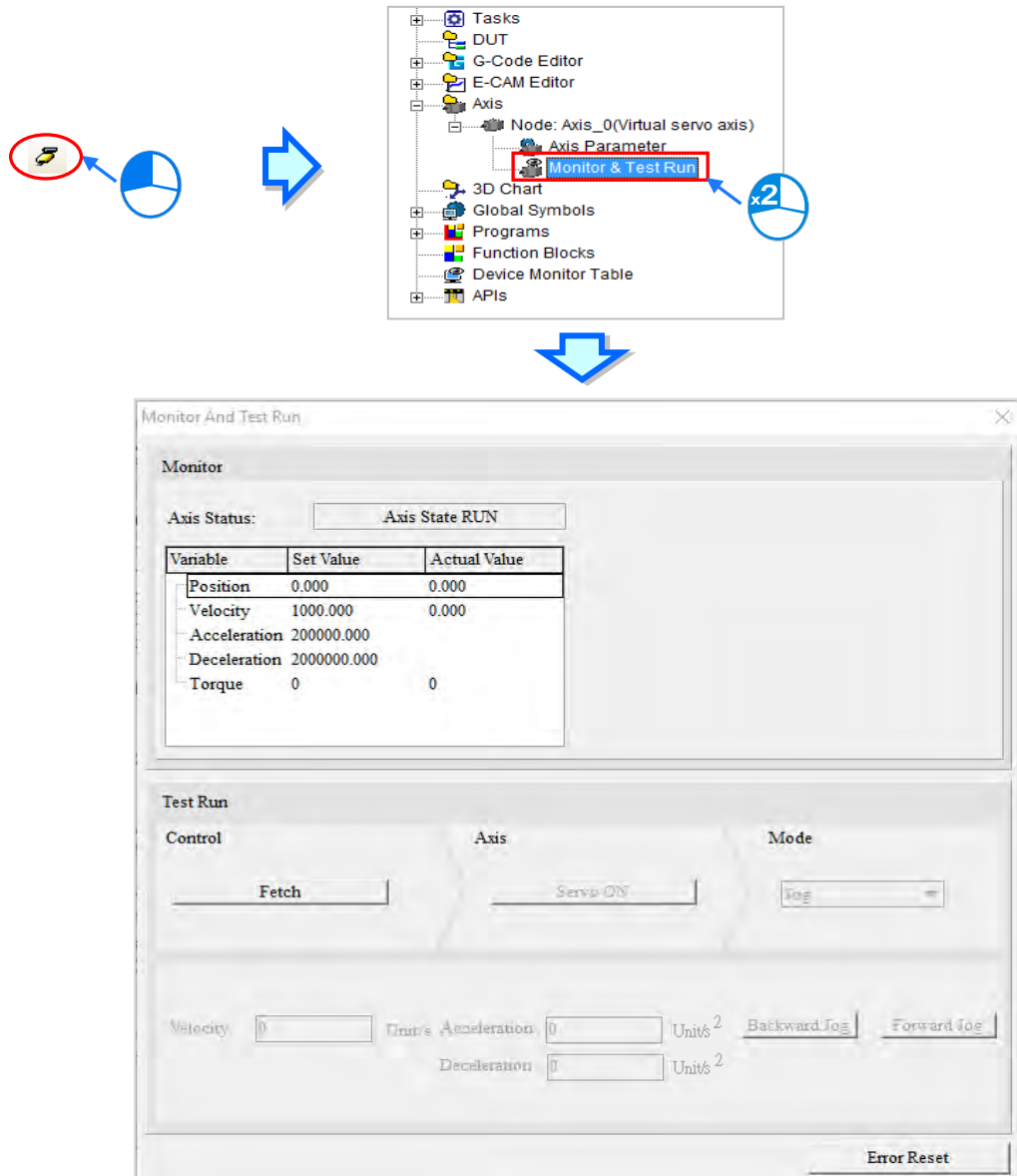
If the button  is clicked, current parameters can be exported to the specified path in the file format of .CSV.

If the button  is clicked, the file at the specified path can be imported.

## 9.2.2 Monitor & Test Run

### 9.2.2.1 AHxxEMC Series Monitoring & Test Run

When ISPSOft is in online mode, double-click **Monitor & Test Run** to open the setting window.



The upper part of the window above is a monitor window for displaying the axis state, the set value and actual value of a variable such as positions and velocities.

Monitor

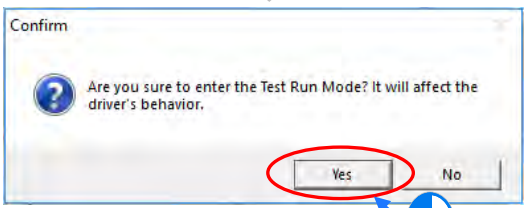
Axis Status:

Variable	Set Value	Actual Value
Position	0	0
Velocity	1000	-902829859
Acceleration	500	
Deceleration	300	
Torque	0	-349253845

The bottom part of the window is a test-run window for giving a test-run command to the axis after hardware connection is made in the period of the test.

After the **Fetch** button is clicked, the **Confirm** window will appear for confirmation of the control over the axis.

Please note that the field configuration and personnel's positions must be confirmed in order to make sure that no injury to personnel and no damage to devices when the motor works before the test run of the axis is started.

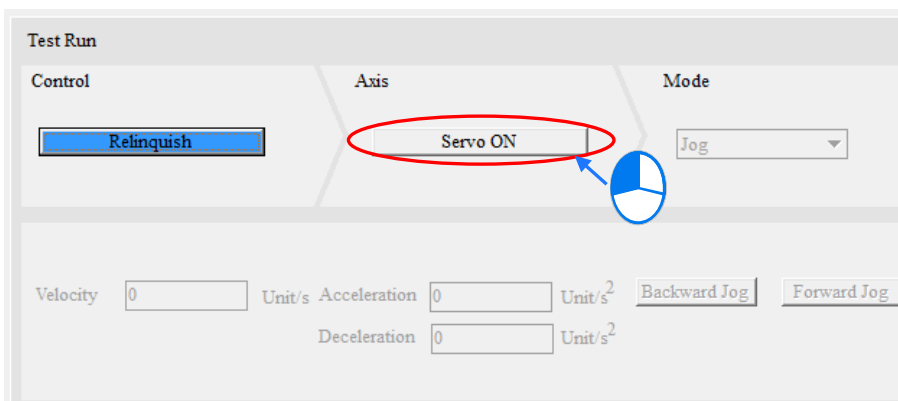


9



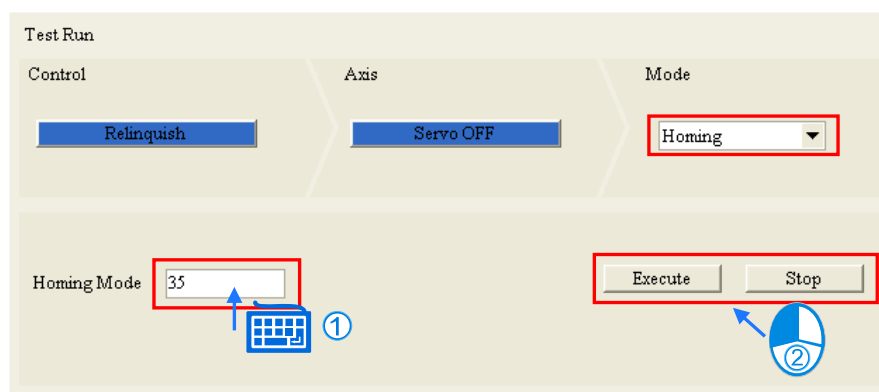
To exit the test-run mode, click the **Relinquish** button to stop having the control over the axis.

Click **Servo ON** button to turn on or **Servo OFF** button to turn off the servo drive after having the control over the axis.

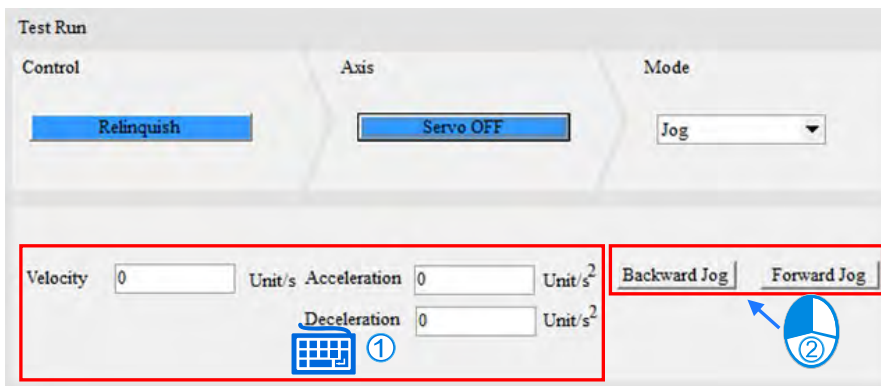


When **Servo ON** is clicked, various test-run modes in the **Mode** field are for option. There are different execution functions for users to type under various modes. The corresponding functions are explained as below.

**Homing Mode:** Perform the homing according to the axis parameter setting and type the homing mode number in the **Homing Mode** field based on the 35 homing modes in CANopen standard. Click the **Execute** button to start the homing. Click **Stop** to abort the homing action.



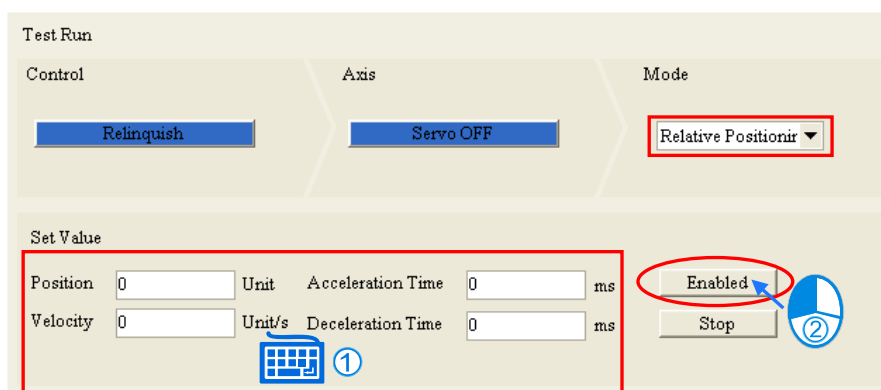
**Jog:** Execute the jog action according to the jog velocity in the axis parameter setting. After typing the jog velocity in the **Velocity** field, the acceleration unit and the deceleration unit, execute the backward jog or forward jog by clicking the **Backward Jog** or **Forward Jog** button.



Velocity: Type the values in the **Velocity**, **Acceleration Time** and **Deceleration Time** fields. If the **Forward** or **Negative** button is clicked, the drive will accelerate till the set velocity is reached according to the set acceleration time and then keep moving in the forward or backward direction. By clicking **Stop**, the axis will decelerate to stop according to the set acceleration time.

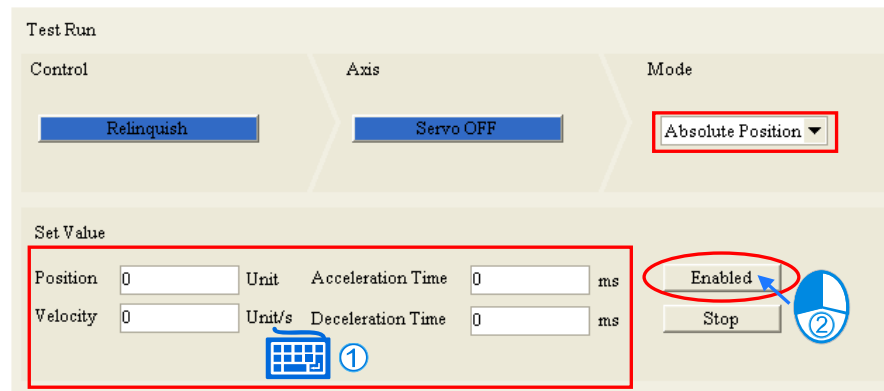


**Relative Positioning:** Type values in the **Position** (adding + or -), **Velocity**, **Acceleration Time** and **Deceleration Time** fields. After the **Enabled** button is clicked, the drive will accelerate till the set velocity is reached and keep moving according to the set acceleration time. And then the drive decelerates to stop and finally reaches the relative position according to the set deceleration time. Clicking **Stop** will abort the action during the motion.

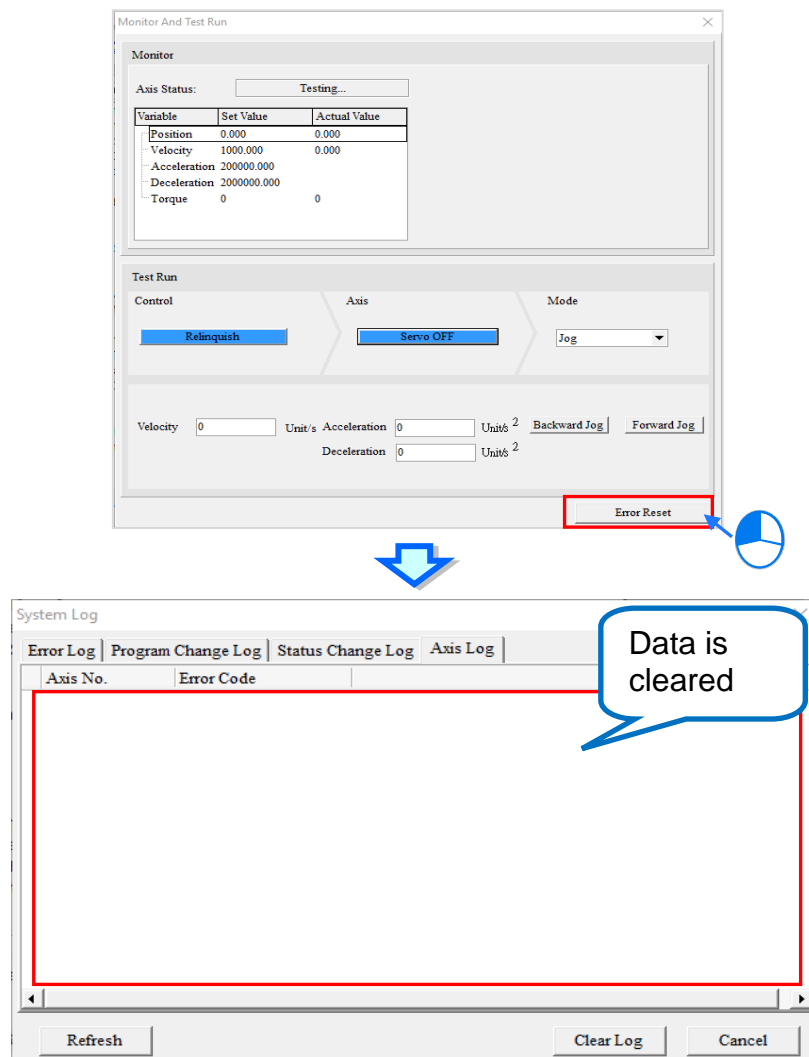


9

**Absolute Positioning:** Type values in the **Position**, **Velocity**, **Acceleration Time** and **Deceleration Time** fields. After the **Enabled** button is clicked, the drive will accelerate till the set velocity is reached and keep moving according to the set acceleration time. And then the drive decelerates to stop and finally reaches the absolute position according to the set deceleration time. Clicking **Stop** will abort the action during the motion.

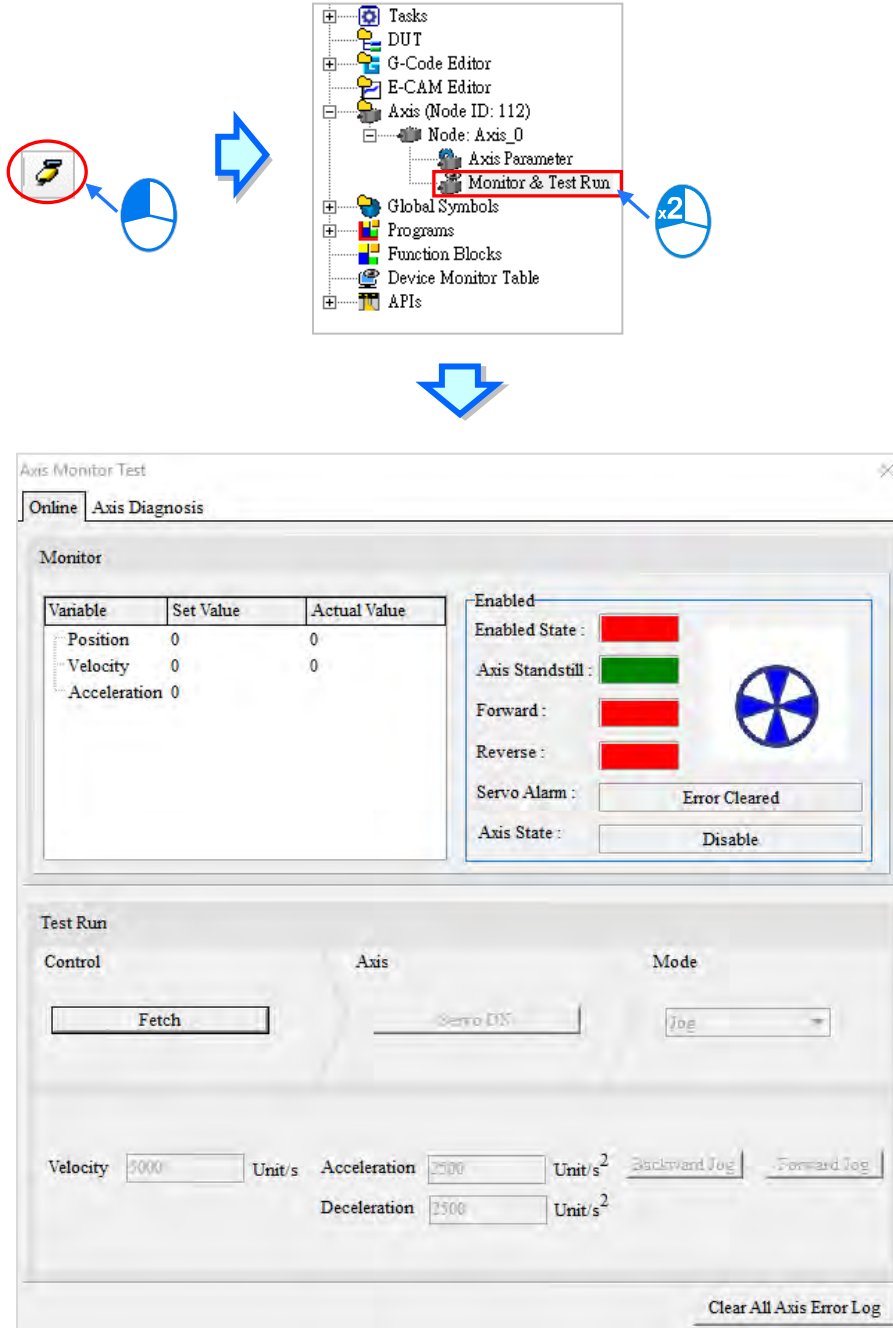


Click Error Reset and all the data in the System Log is cleared.

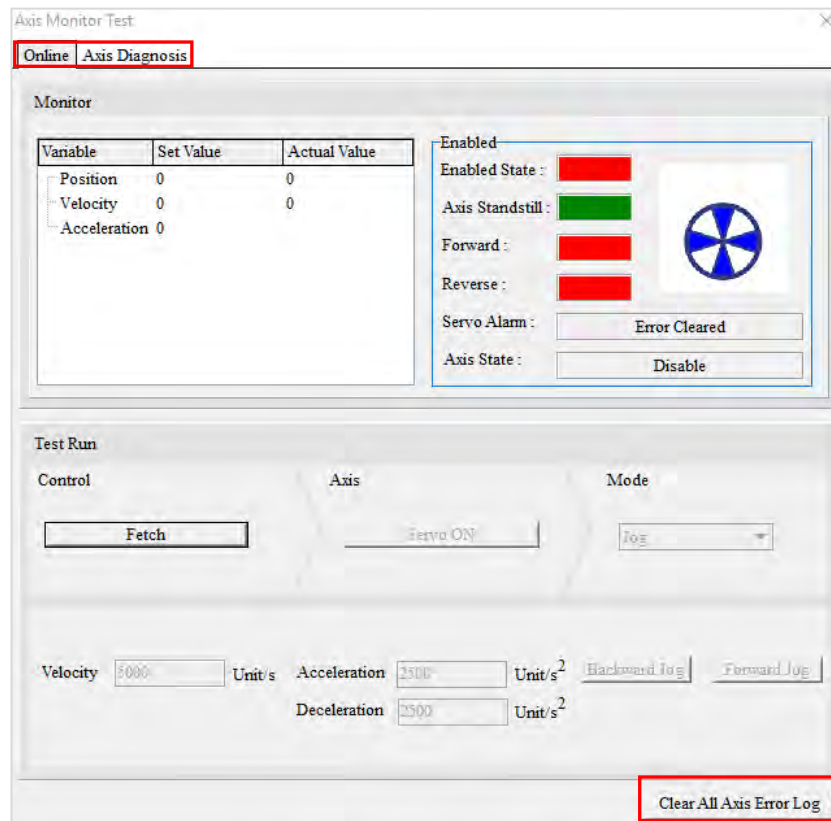


### 9.2.2.2 DVPxxMC Series Monitoring & Test Run

For DVPxxMC, double-click **Monitor & Test Run** under **Axis** to open the Monitor & Test Run window after ISPSOft enters the online mode.

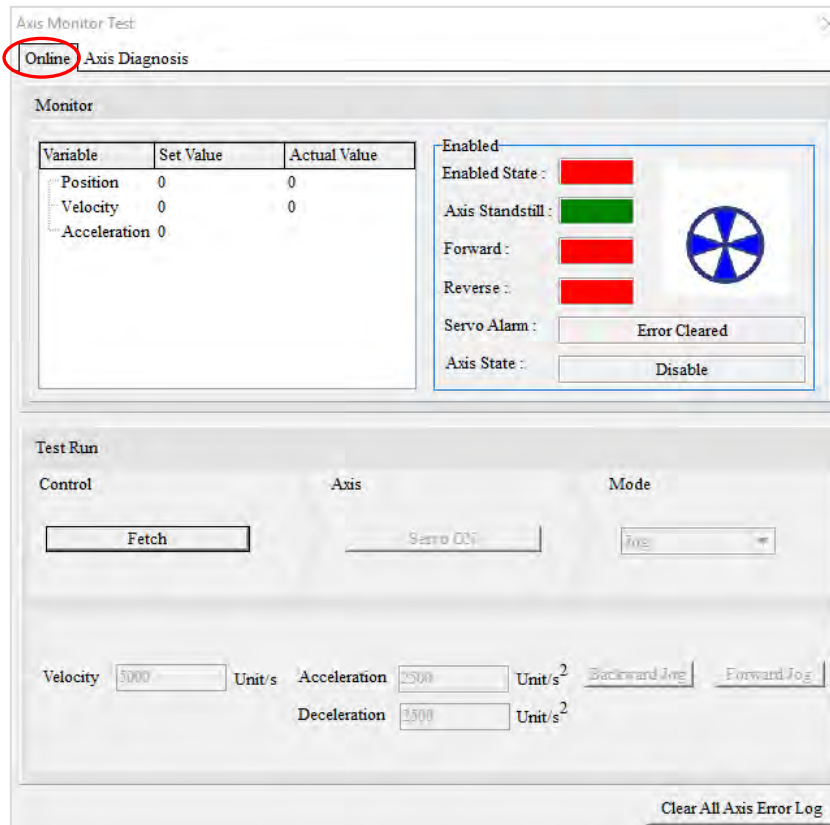


The Monitor & Test Run window contains **Online** and **Axis Diagnosis** tabs. The **Clear All Axis Error Log** button at the bottom of the page can clear all the axis error records from the project management area.

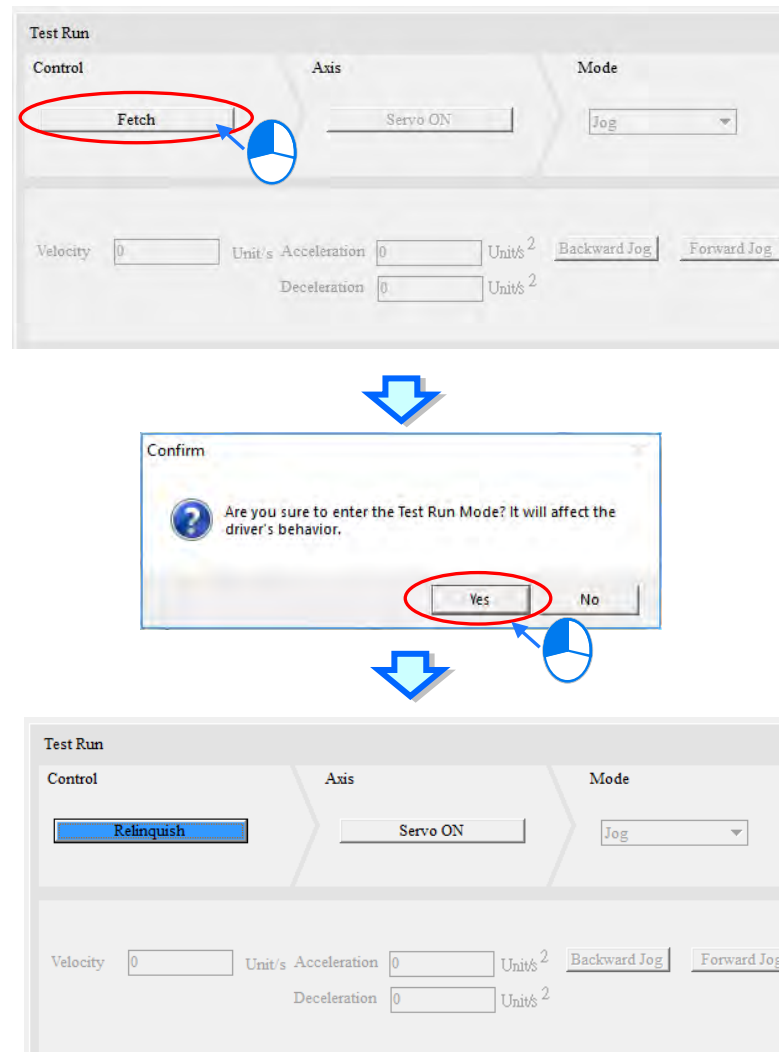


- **Online**

The upper part of **Online** is the **Monitor** section. On the left are items regarding real-time set value as well as actual value of the axis state and the physical quantity with the following variables, e.g. position and velocity. On the right is the **Enabled** section that displays the axis state and servo alarm. This section can be used jointly with the Test Run section at the lower part of the window.



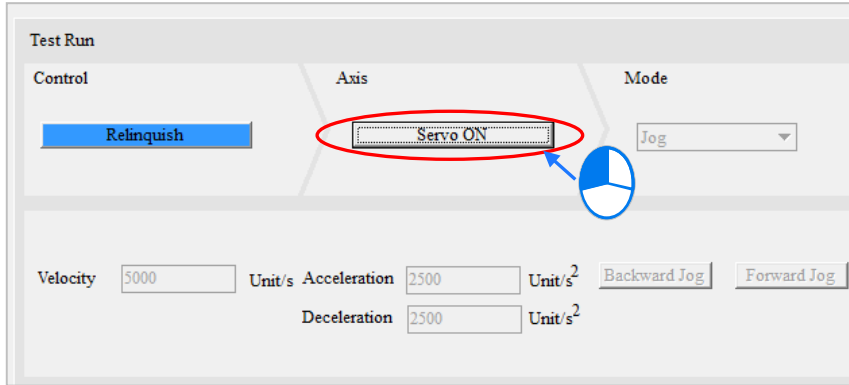
The **Test Run** section is for testing the axis state after connecting to the hardware. Users can click Fetch and a window will appear for confirmation of the control over the axis. When motors are running, please make sure that field configuration and personnel's positions will not cause personnel injury or object damage before doing the Test Run.



To exit the test-run mode, click the **Relinquish** button to stop having the control over the axis.

Click **Servo ON** button to turn on or **Servo OFF** button to turn off the servo drive after having the control over the axis.

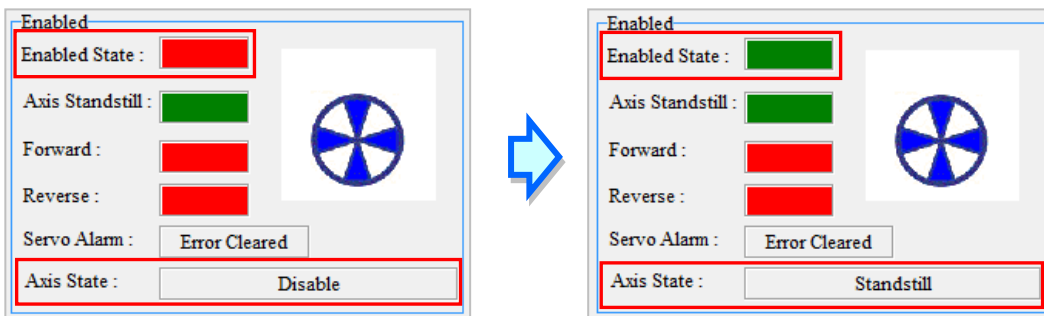
\*. If users select Virtual Axis for mode, the axis will automatically switch to Servo ON, please refer to section 9.2.1.2 for more information on axis mode.



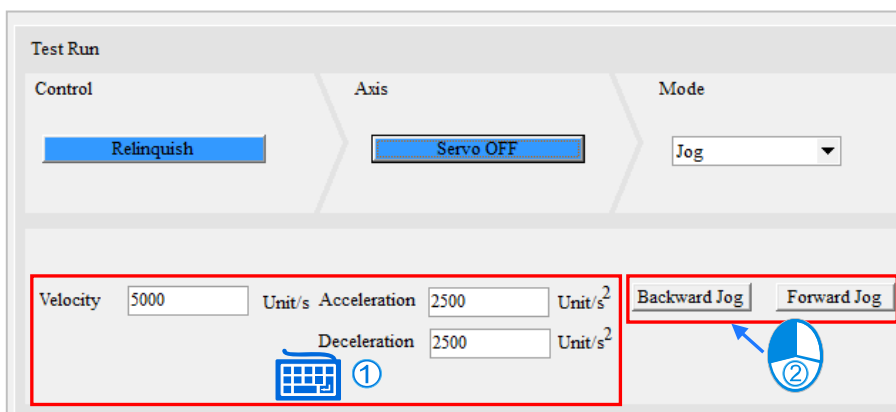
By connecting servo drives and motion controllers via communication ports, users can click **Servo ON** and the system will point out the following remarks in Jog mode.

- (a) Velocity selected under cyclic data exchange.
- (b) Do NOT use MC\_Power instruction for axis programming
- (c) If communication disconnects during jog mode, the servo axis will automatically stop after 1 sec.

When Servo OFF change to ON, the red light will change to green in the **Enabled State** field and the **Axis State** changes from Disable to Standstill under Enabled section.



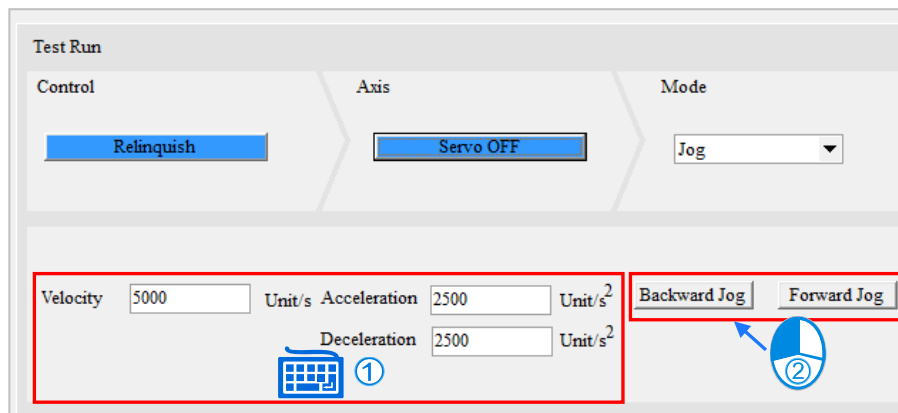
Under Servo is ON, currently only Jog mode can be selected for test run and functions are explained as below.



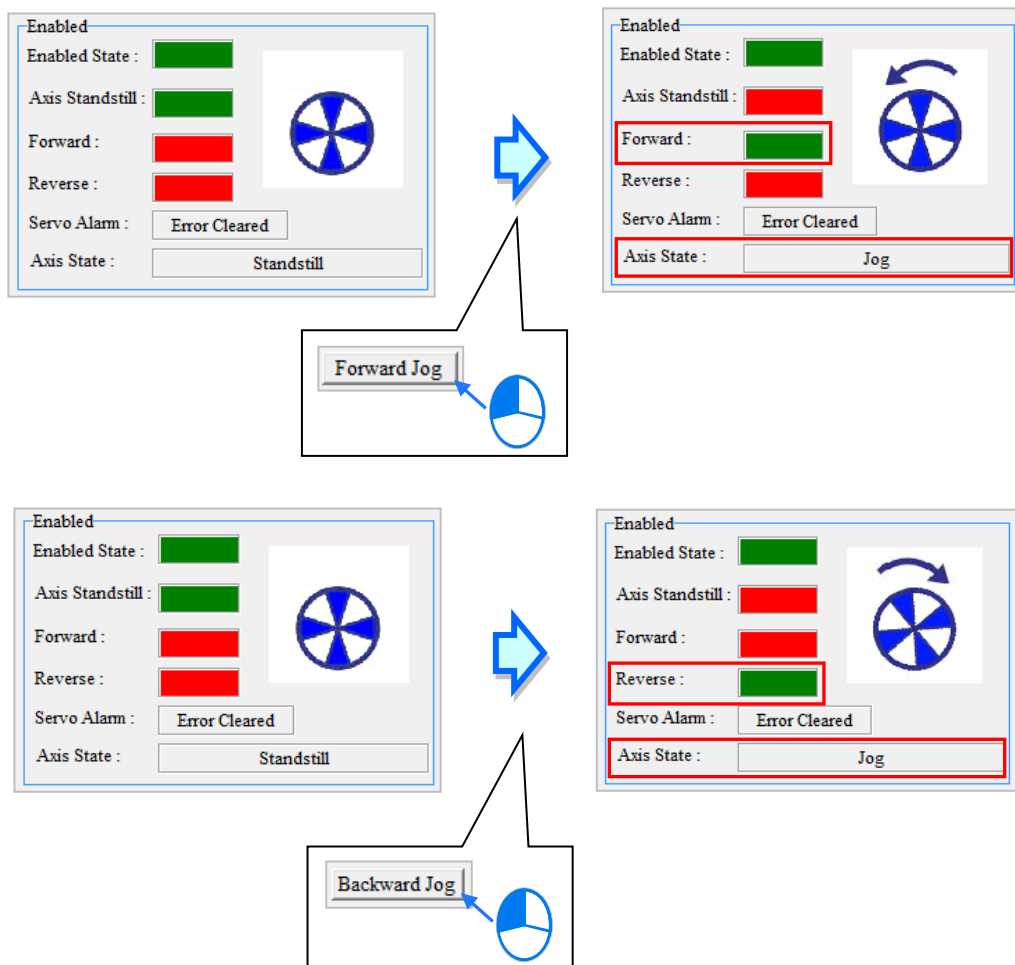
9



**Jog:** Users can input parameter settings including **Velocity**, **Acceleration**, **Deceleration** and press **Forward Jog** or **Backward Jog** for execution.



When **Forward Jog / Backward Jog** button is clicked, the red light will turn to green in Forward /Reverse field and the **Axis State** changes from Standstill to **Jog** under Enabled section. (This function is used when controllers are in operating mode.)

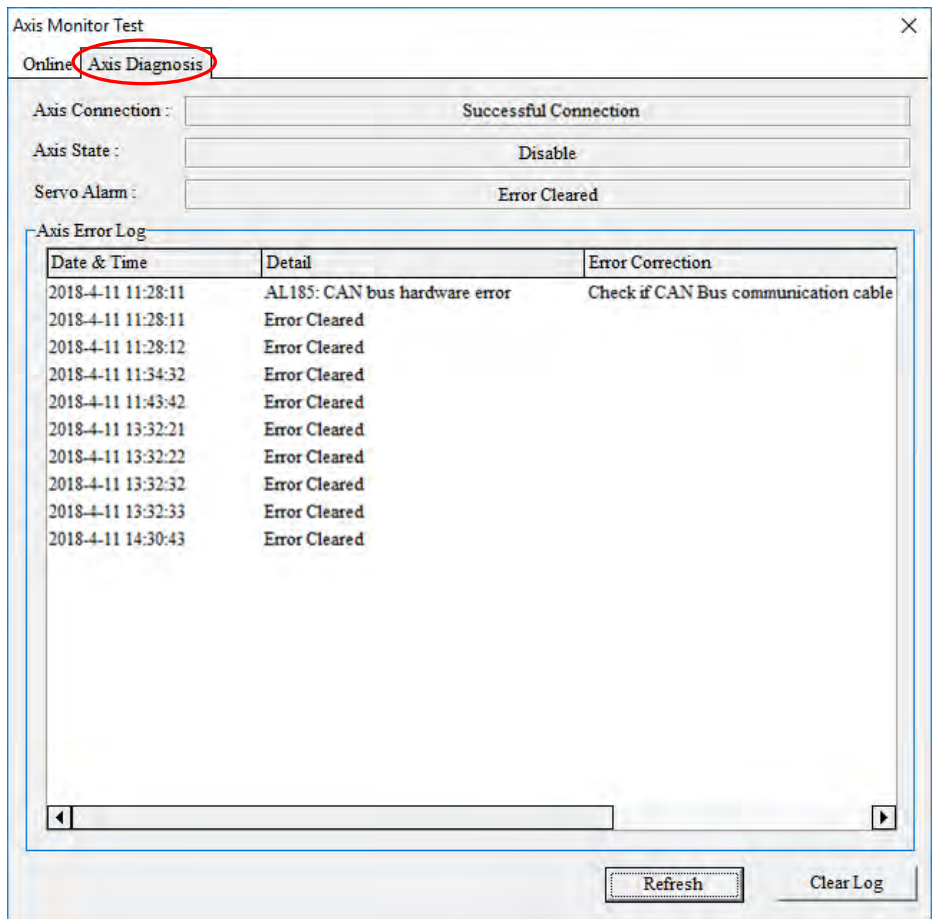


Relevant parameters in Enabled section are described below.

Parameter	Description
Enabled State	Color : Green - Enabled; Red - Disabled
Axis Standstill	Color : Green - Standstill; Red - Jog
Forward	Color : Green – Forward; Red – Reverse/Standstill
Reverse	Color : Green – Backward; Red – Reverse/Standstill
Servo Alarm	Display error message
Axis State	Include Disable, Standstill, ErrorStop, Stopping, Homing, Discrete Motion, Continuous Motion, Synchronized Motion and Jog

- Axis Diagnosis

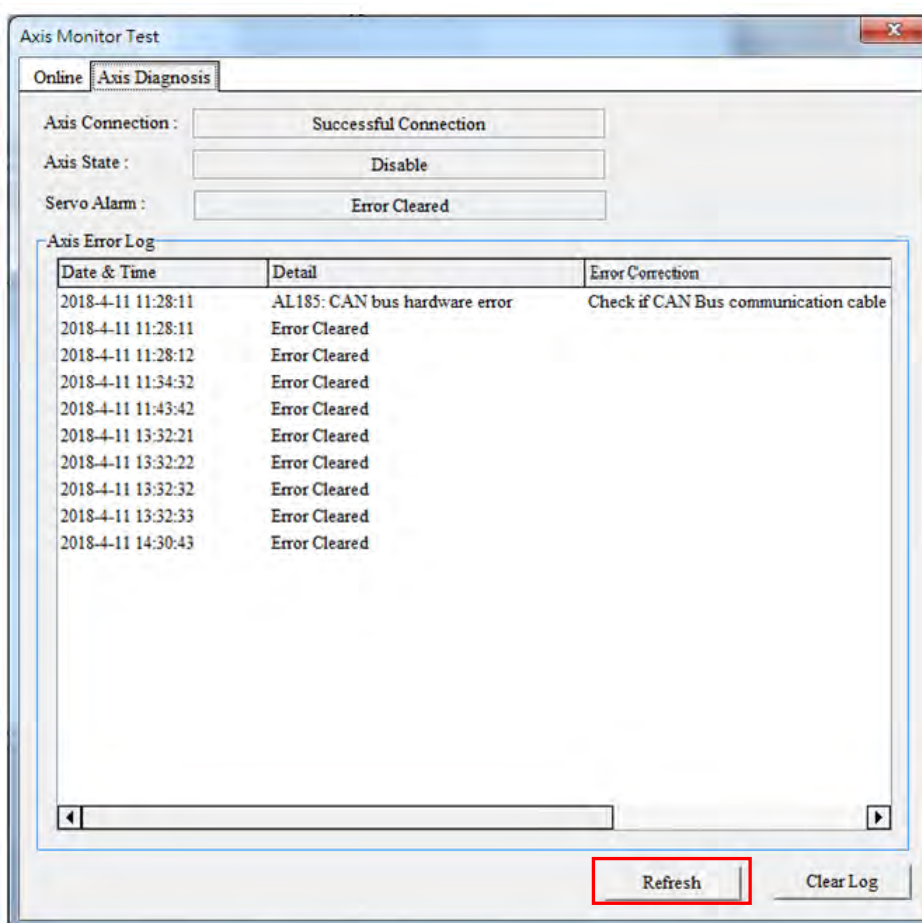
The function provides diagnosis for axis connection, state and servo alarm, while error detail and correction are displayed in the **Axis Error Log** section.



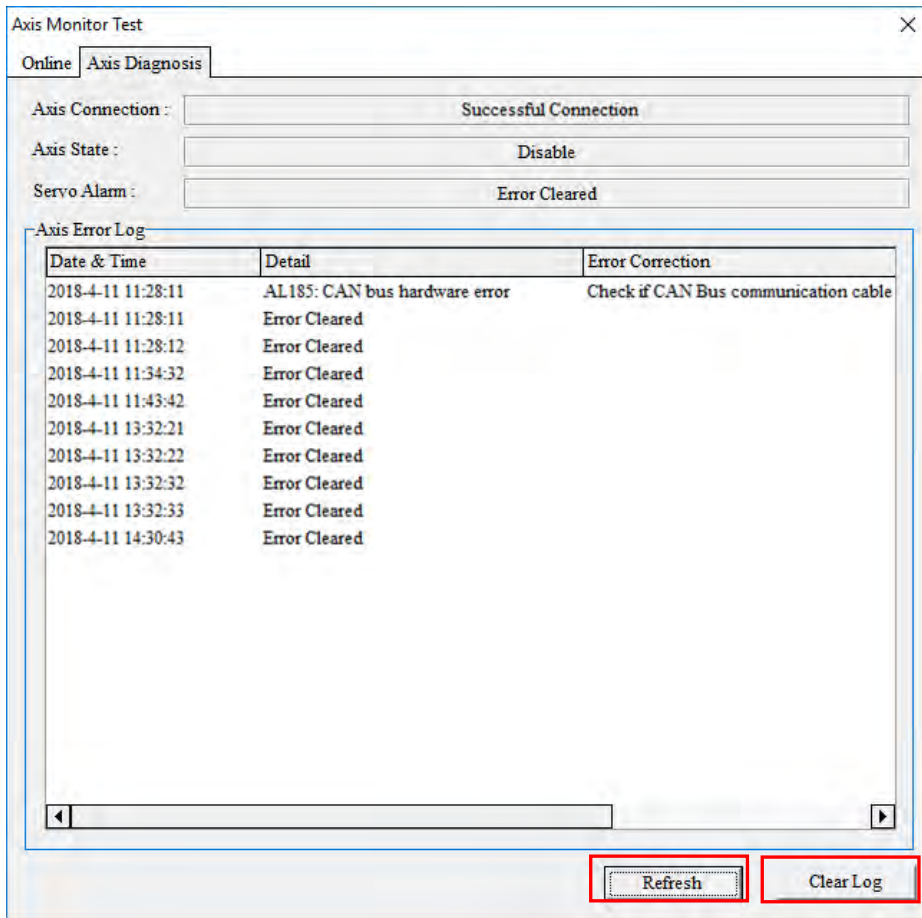
Relevant parameters of Axis Diagnosis are described below.

Parameter	Description
Axis Connection	Successful connection or disconnection between servo drives and controllers
Axis State	Corresponds to the state machine ( Please refer to related product manuals for more details )
Servo Alarm	Display error message
Axis Error Log	Records error data including the date & time, detail and correction.

Click **Refresh** to update the Axis Diagnosis page.



Click **Clear Log** button to delete the current axis error log data.



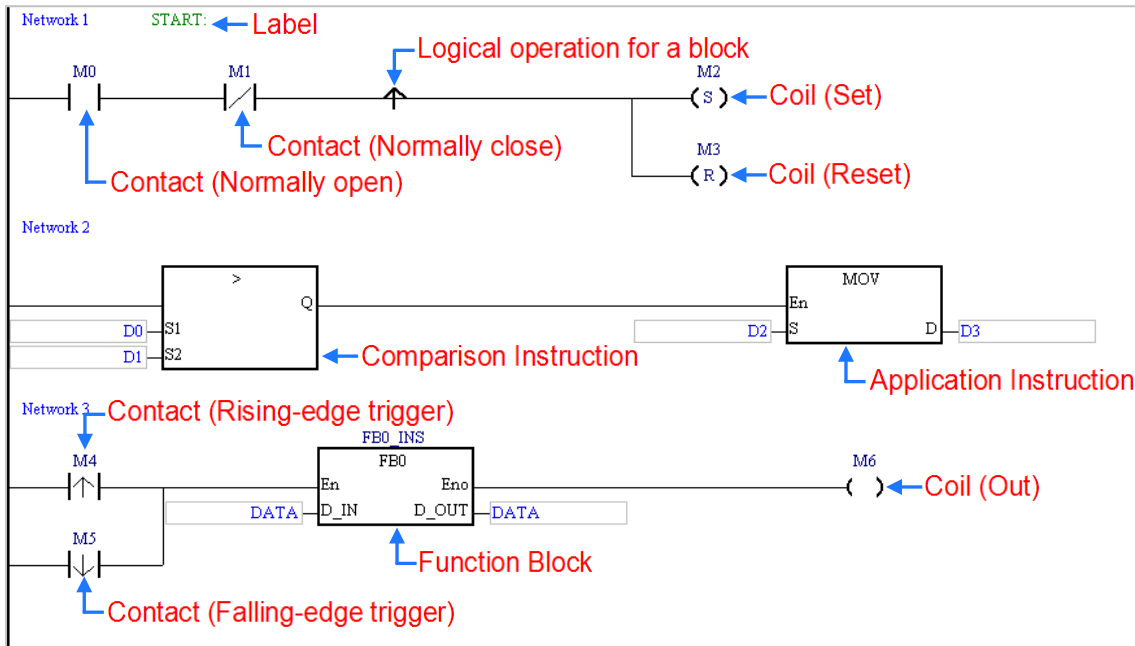
# Chapter 10 Ladder Diagram

## Table of Contents

<b>10.1</b>	<b>Introduction of a Ladder Diagram .....</b>	<b>10-2</b>
10.1.1	Editing Environment.....	10-2
10.1.2	Networks in a Ladder Diagram .....	10-4
10.1.3	Selecting Objects.....	10-5
<b>10.2</b>	<b>Creating a Ladder Diagram in ISPSOft .....</b>	<b>10-7</b>
10.2.1	Creating a Contact, Coil & MPS.....	10-7
10.2.1.1	Inserting a Contact and Changing a Contact Type .....	10-7
10.2.1.2	Inserting a Coil and Changing a Coil Type.....	10-9
10.2.1.3	Add MPS .....	10-10
10.2.2	Use Device, Symbols and Constants in LD.....	10-11
10.2.3	Typing Instructions .....	10-13
10.2.4	Inserting Applied Instructions and Function Blocks.....	10-14
10.2.5	Creating a Comparison Contact .....	10-17
10.2.6	Inserting a Block Logic Instruction .....	10-19
10.2.7	Creating Multiple Outputs .....	10-22
10.2.8	Putting a Label .....	10-23
10.2.9	Editing a Comment .....	10-24
10.2.10	Symbol Mode and Address Mode.....	10-26
10.2.11	Bookmark.....	10-27
10.2.12	Activating/Inactivating a Network .....	10-28

## 10.1 Introduction of a Ladder Diagram

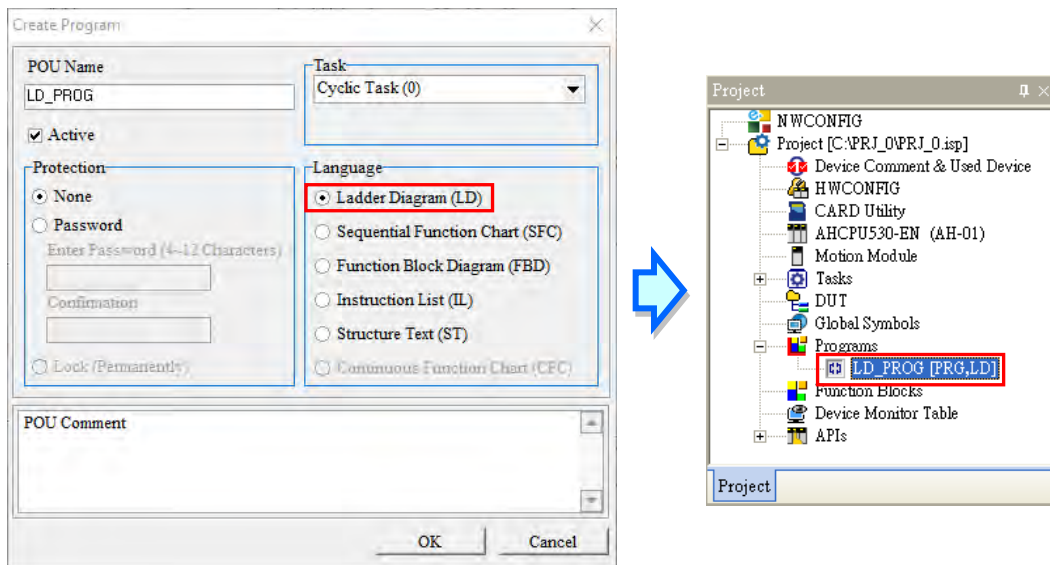
A ladder diagram is one of the programming languages defined by IEC 61131-3, and is widely used to create a PLC program. A ladder diagram is a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. A ladder diagram in ISPSOft is shown below.



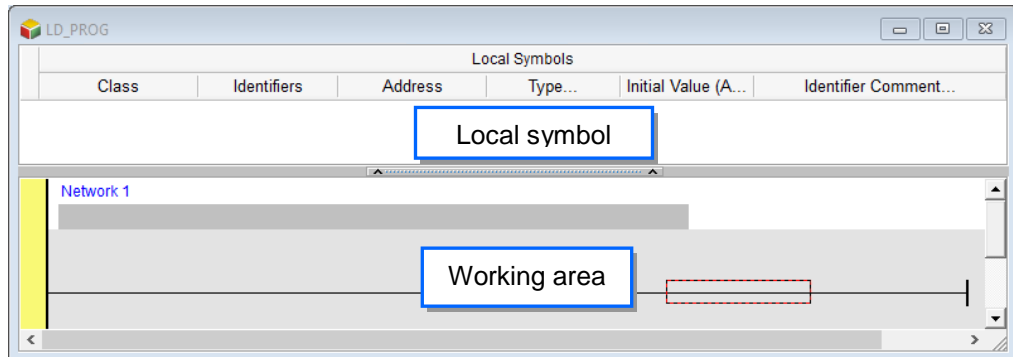
\*. Please refer to DVP-PLC Application Manual—Programming for more information about ladder diagrams.

### 10.1.1 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
		Display/Hiding symbol or address
	Shift+Ctrl+C	Displaying/Hiding the comments on the networks
	None	Displaying/Hiding the commands on the devices
	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	Add MPS
	Typing an instruction	Inserting a comparison contact
	Typing an instruction	Selecting a type of comparison contact

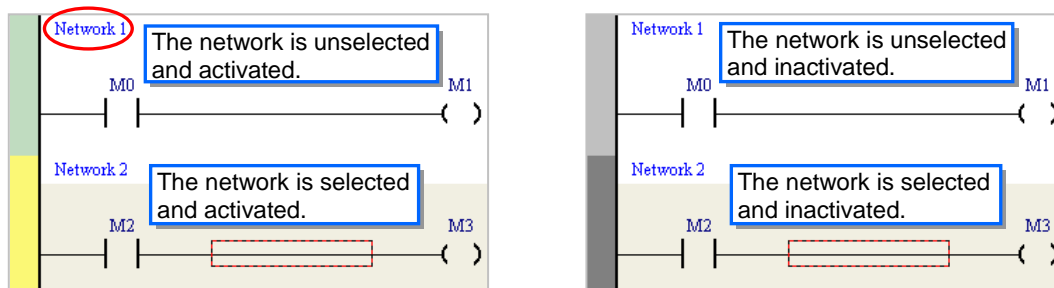
Icon	Keyboard shortcut	Function
	Typing an instruction	Inserting a block logic instruction (NP/PN/INV/FB_NP/FB_PN)
	Typing an instruction	Selecting a type of block logic instruction
	Shift+Ctrl+U	Inserting an instruction or a function block

\*. Please refer to section 8.2.3 for more information about typing an instruction.

### 10.1.2 Networks in a Ladder Diagram

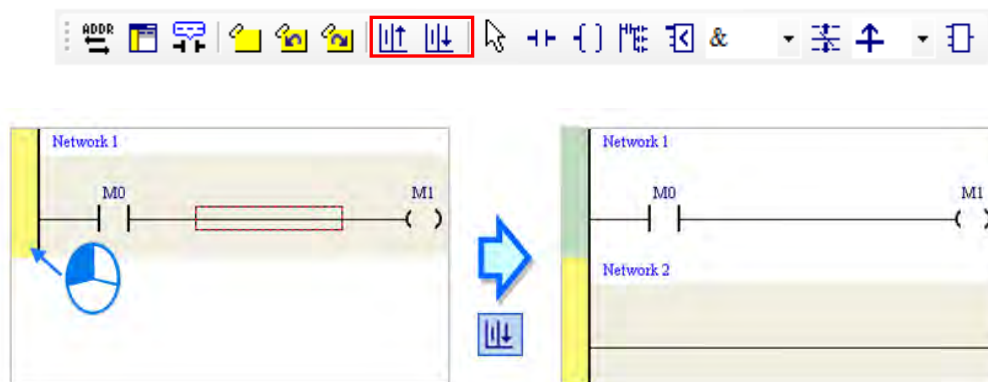
A ladder diagram consists of networks. Every network is an independent program. Besides, there is no limit on the number of objects which can be inserted in a network in ISPSOft, and therefore there is no mark which is used to connect two networks.

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.



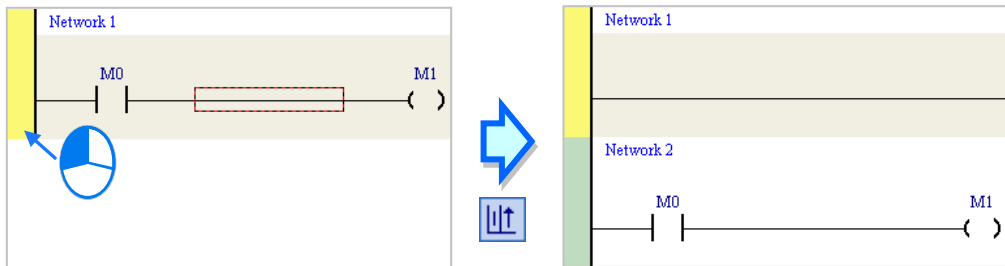
After a new window in which a ladder diagram will be created is opened, the system will insert a blank network in the window. If users want to add a network, they can select a network, and click on the toolbar. The network added will be under the network selected.

10




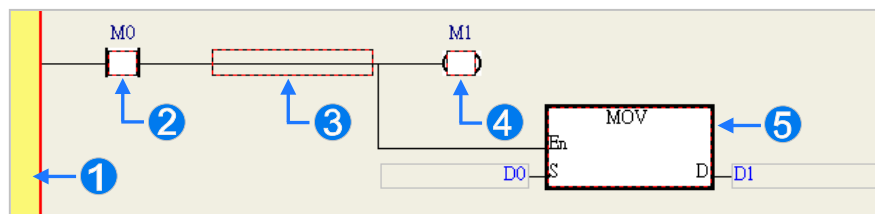


If users click  on the toolbar, a blank network will be put above the network selected.



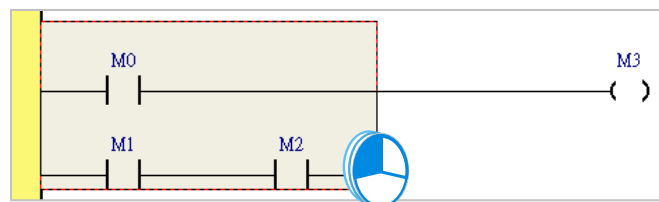
### 10.1.3 Selecting Objects

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.

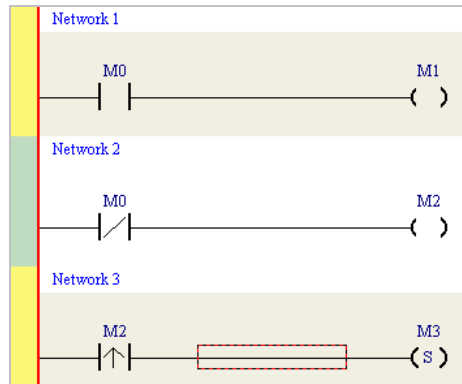


- ❶ Selecting the network
- ❷ Selecting the input contact
- ❸ Selecting the network
- ❹ Selecting the output coil
- ❺ Selecting the block

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.



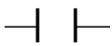


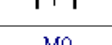



If users want to select several networks, they can hold down Ctrl on the keyboard while they click the networks. The users can also select a range of networks by pressing Shift on the keyboard, clicking the first network within the range, and the last network within the range.



## 10.2 Creating a Ladder Diagram in I SPSoft


### 10.2.1 Creating a Contact, Coil & MPS

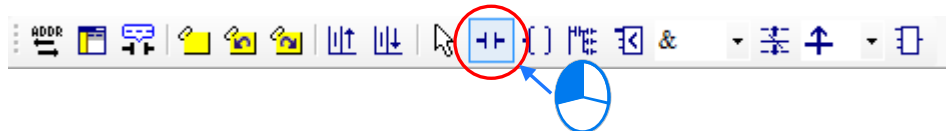
The representations of contacts and the representations of coils in I SPSoft are listed below.

Contact	Normally-open contact	M0 
	Normally-closed contact	M0 
	Rising edge-triggered contact	M0 
	Falling edge-triggered contact	M0 
Coil	OUT	M1 
	SET	M1 
Coil	RESET	M1 

#### 10.2.1.1 Inserting a Contact and Changing a Contact Type

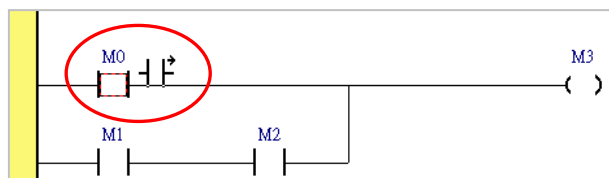
- (1) Select a line, a contact, or a group of devices. Before users connect a contact to a group of devices in parallel or in series, they have to draw a frame round the group of devices.

After users make sure of the position in which a contact will be inserted, they can click  on the toolbar.

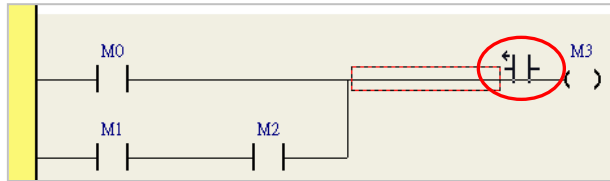


- (2) The mouse cursor appears as a contact when the mouse cursor is near the position in which a contact will be inserted.

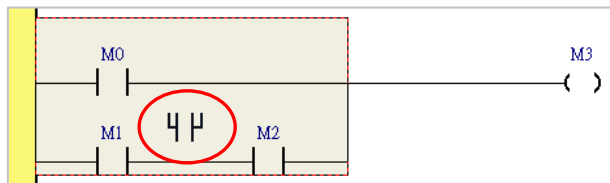
- If the mouse cursor is near a contact, a contact will be connected to the contact in parallel or in series.



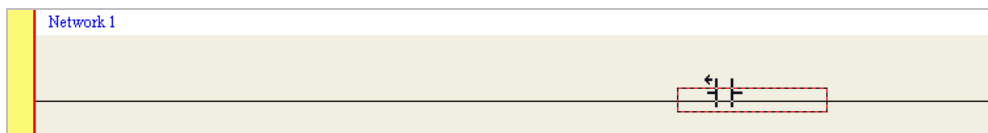
- If the mouse cursor is near a line, a contact will be inserted in the line.



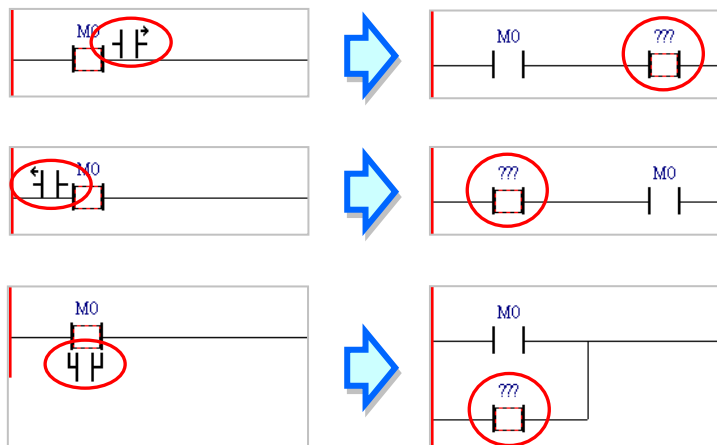
- If the mouse cursor is near a group of devices, a contact will be connected to the group of devices in parallel or in series.




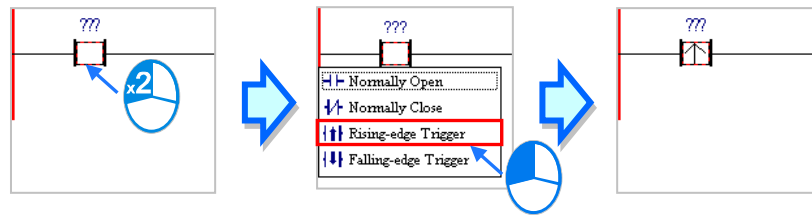
- If users want to insert a contact in a new network, they can move the mouse cursor to the red frame in the new network.




- (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.



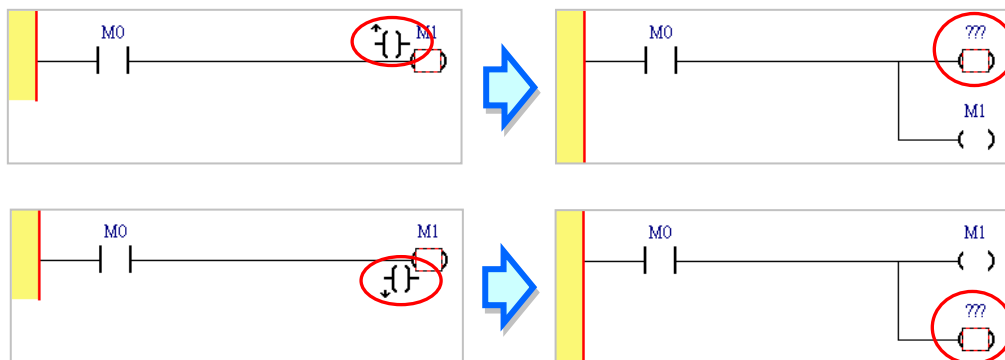
### 10.2.1.2 Inserting a Coil and Changing a Coil Type

- (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.



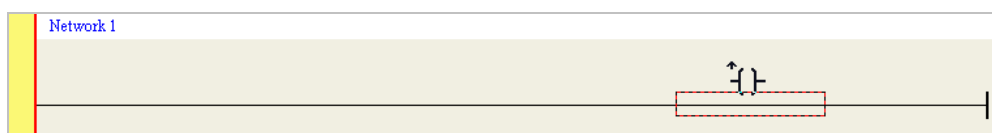
- (2) The mouse cursor appears as a coil when the mouse cursor is near the position in which a coil will be inserted. If the mouse cursor is at the top of a coil or at the top of an applied instruction, a coil will be put above the coil or the applied instruction. If the mouse cursor is at the bottom 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.

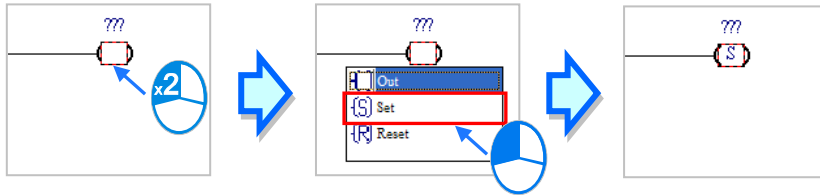


10


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.



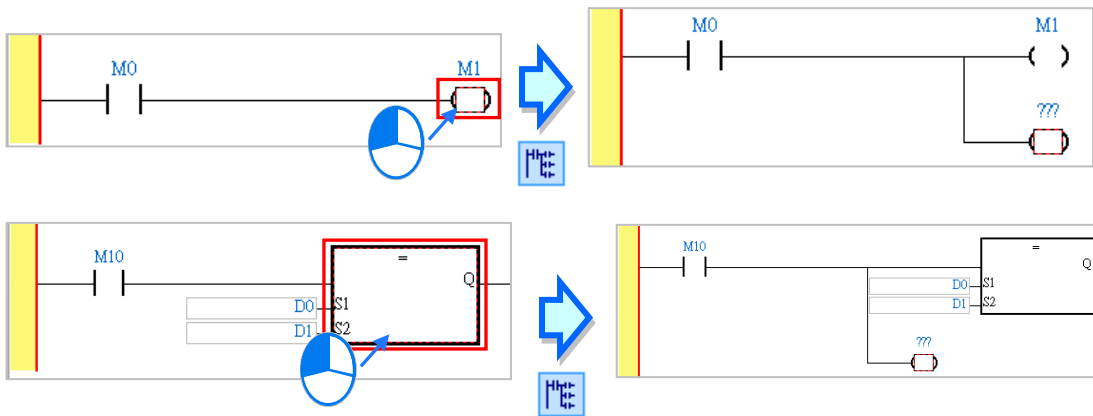
- (3) After a coil is inserted, users can click  on the toolbar or press Esc on the keyboard. After the coil is double-clicked, users can select a coil type on the drop-down list.



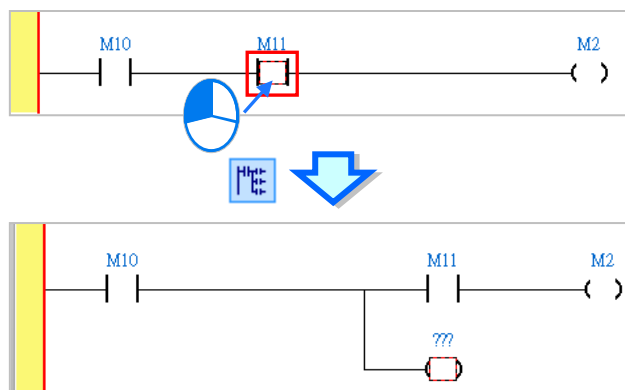
### 10.2.1.3 Add MPS

Choose a line, coil, comparison or API position to confirm and add the MPS by clicking  on the toolbar which will create a bifurcation point.

Users can select the line or comparison instruction, then click the add MPS icon to create a bifurcation point.

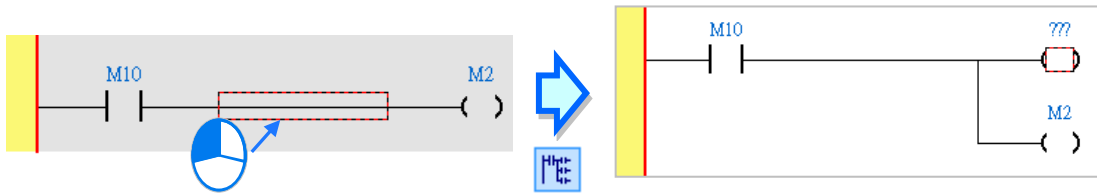


Users can select a contact, then click the add MPS icon to create a bifurcation point.

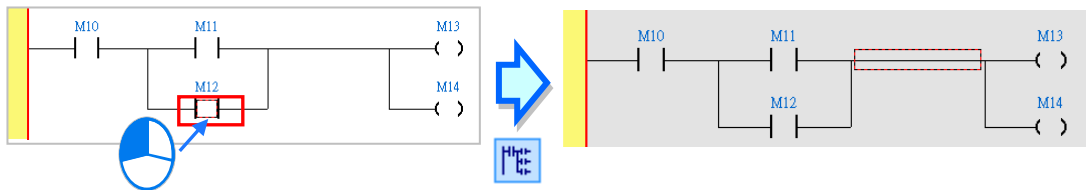


10

Users can select a line, then click the add MPS icon to create a bifurcation point.

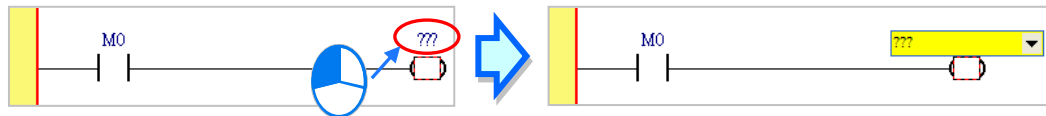



Some positions cannot add MPS, so when users click add MPS icon, the diagram will show the position to insert MPS.

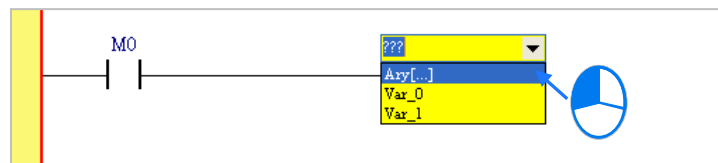


### 10.2.2 Use Device, Symbols and Constants in LD

After users click ??? above an object inserted in a network, they can type an operand.

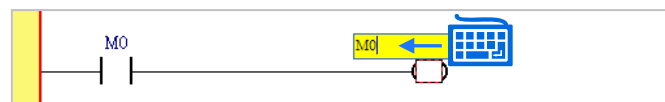


The users can click  in the box, and select a declared symbol on the drop-down list. Select an ARRAY data type symbol from the drop-down list and type an element in the [...]. See below for ARRAY element descriptions.



10

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**

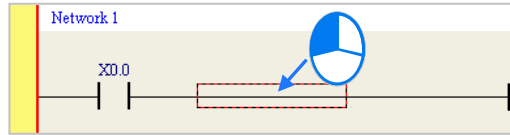
- (a) If users want to use constants in a ladder diagram in ISPSOft, the constants must be represented in the following ways.
- Decimal value: 23456 (A value which is not preceded by any mark will be regarded as a decimal value.)
  - Hexadecimal value: 16#5BA0 (A value is preceded by 16#.)
  - Octal 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 apostrophes); use single apostrophe for DVPxxMC
  - BOOL:
    - AH/AS: SM400 (The flag is always ON when CPU runs), or SM401 (The flag is always OFF when CPU runs) is used.
    - DVP: M1000 or M1001 is used.
- (b) If users want to use a symbol whose data type is ARRAY, the expression format is as the image shown above Ary[...] (Identifier[Index]). An index should be a decimal constant, less than the size of the ARRAY and should start from 0, supporting up to three-dimensional array and separating by a comma, for example Ary[1,2,3], please refer to chapter 8.



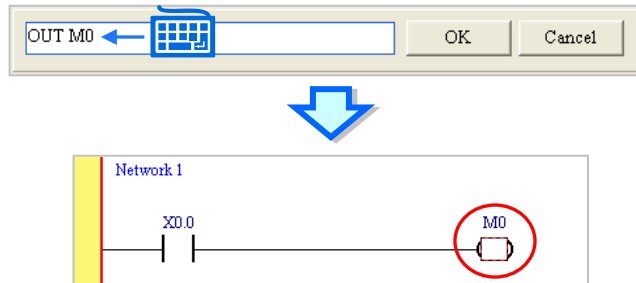
### 10.2.3 Typing Instructions

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"

A dialog box with a text input field containing 'A X0.0' and 'OK' and 'Cancel' buttons.

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

A dialog box with a text input field containing 'B X0.0' and 'OK' and 'Cancel' buttons.

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

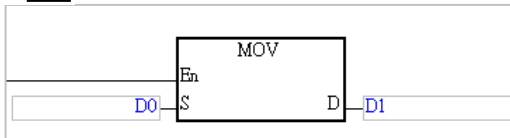
A dialog box with a text input field containing 'O M0' and 'OK' and 'Cancel' buttons.

### 10.2.4 Inserting Applied Instructions and Function Blocks

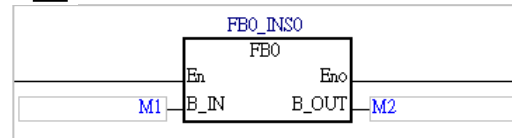
The applied instructions and function blocks in a ladder diagram in ISPSOft are represented by blocks. The pins of a block representing an applied instruction include the pins corresponding to the operands specified in the applied instruction and an En pin. The pins of a block representing a function 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**



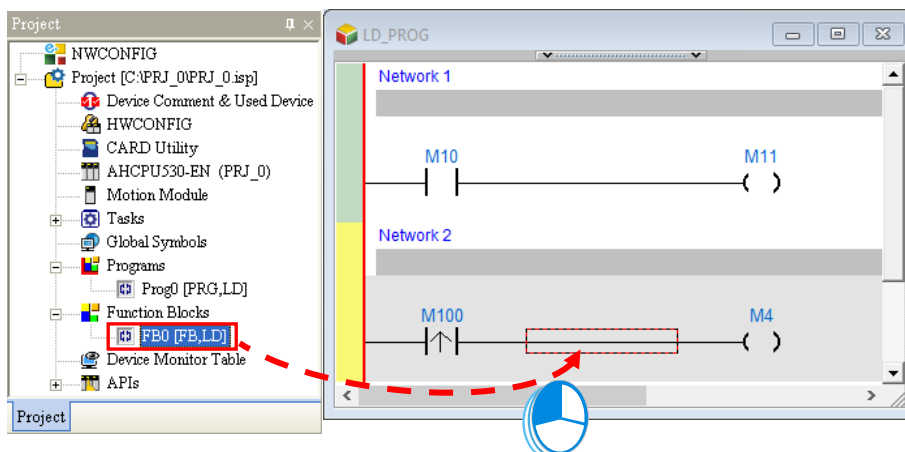
● **FB**



Users can insert an applied instruction or a function block in a ladder diagram in ISPSOft in one of the three ways described below.

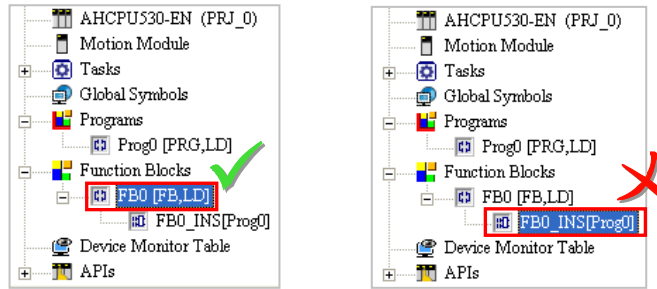
● **Method 1**

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.




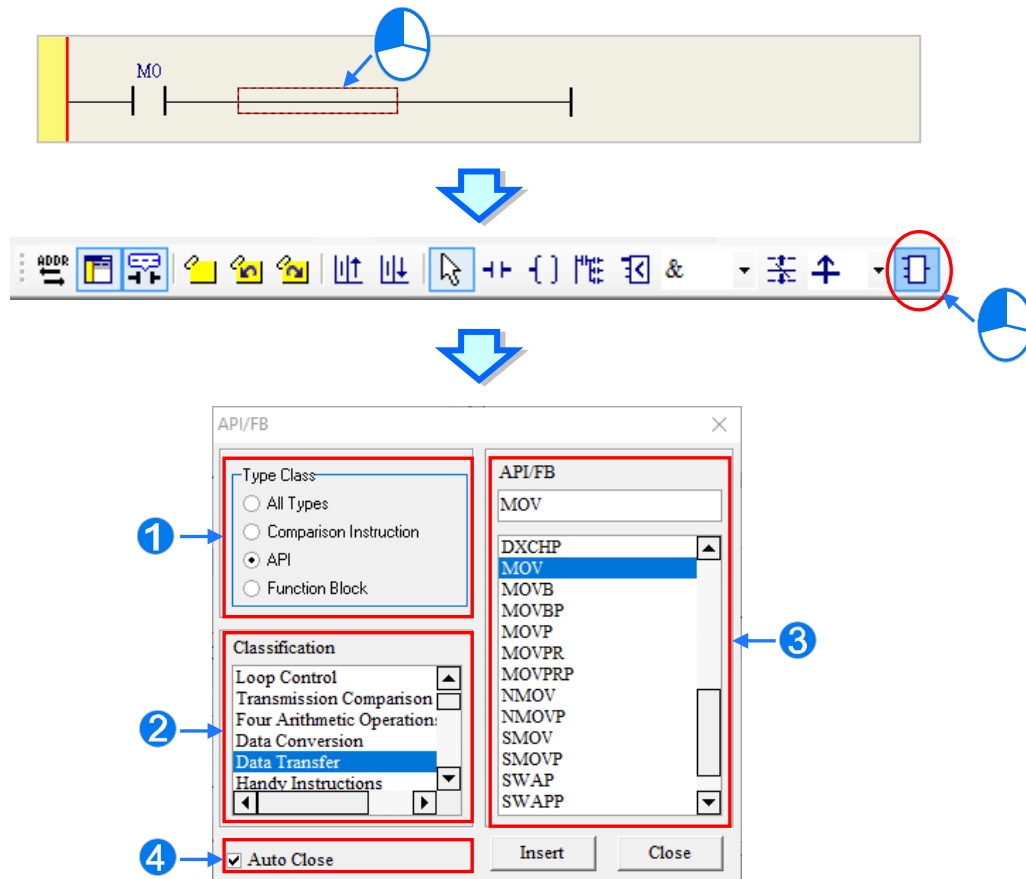
10

Function block definitions can be dragged, but function block instances can not be dragged.



● **Method 2**

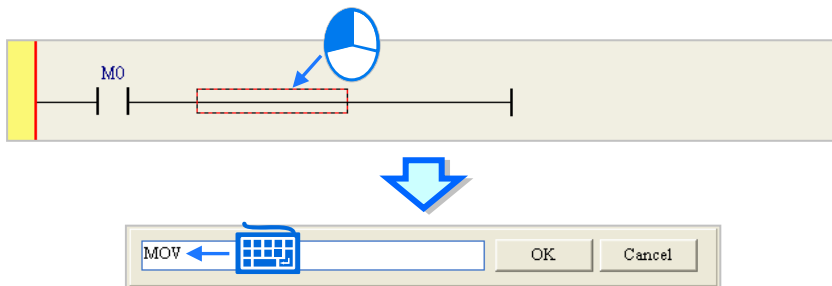
Click the position in which an applied instruction or a function block will be inserted, click  on the toolbar, select the item which will be inserted in the **API/FB** window, and click **Insert**.



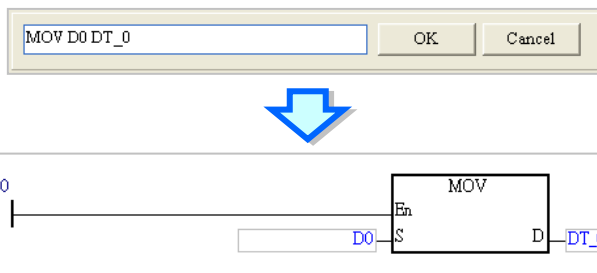
- 1 Users can select the **All Types** option button, the **Comparison Instruction** option button, the **API** option button, or the **Function Block** option button.
- 2 If users select the **API** option button in the **Type Class** section, they have to select an instruction type in the **Classification** 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.)

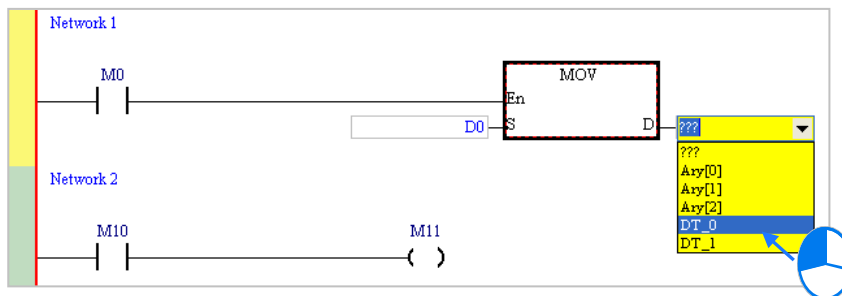


If users want to insert a function block, they have to type the function block definition, and they can not type the operands specified in the function block. If users want to insert an applied instruction, they have to type the applied instruction, and they can type the operands specified in the applied 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.



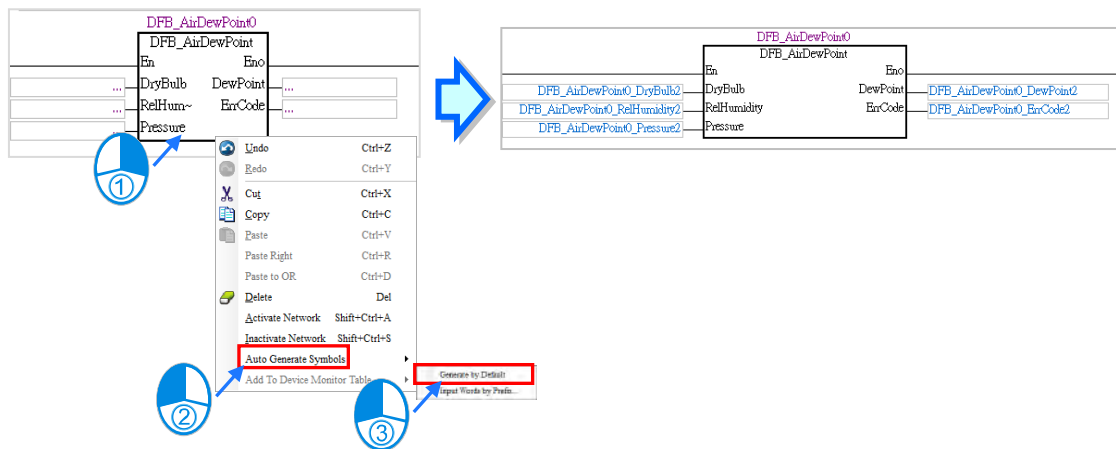
10

After an applied instruction or a function block is inserted successfully, users have to refer to section 10.2.2, and type operands.



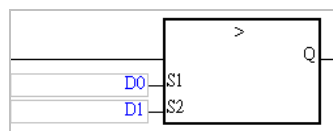
\*. If the Auto-leading 'Add Symbol' Dialog checkbox in the Options window is selected, the Add Symbol window will appear automatically after users type a symbol which is not declared. Please refer to section 2.3.1 for more information.

Right-click on top of the function block, select Auto Generate Symbols and choose to generate by default. The default name usually adopts function block name + pin name and once symbols are generated type in pin position (see below); type in the spelling or words to generate symbols and type in pin position.



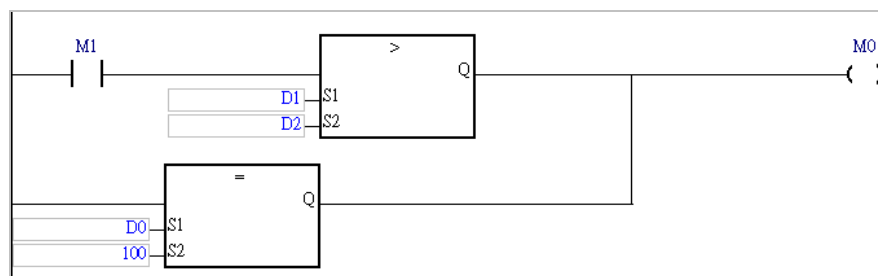
### 10.2.5 Creating a Comparison Contact

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.




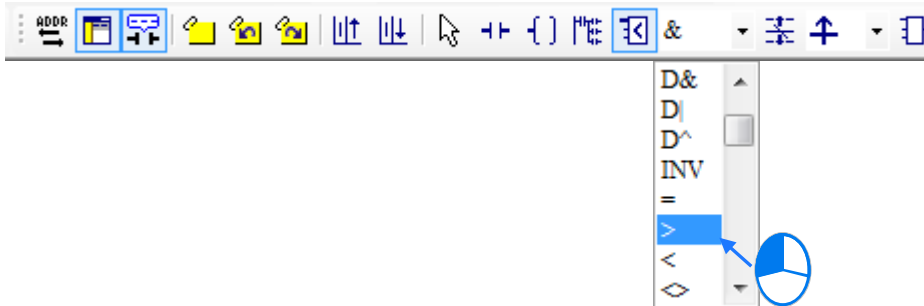
10


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 10.2.4, but also by means of the following steps.

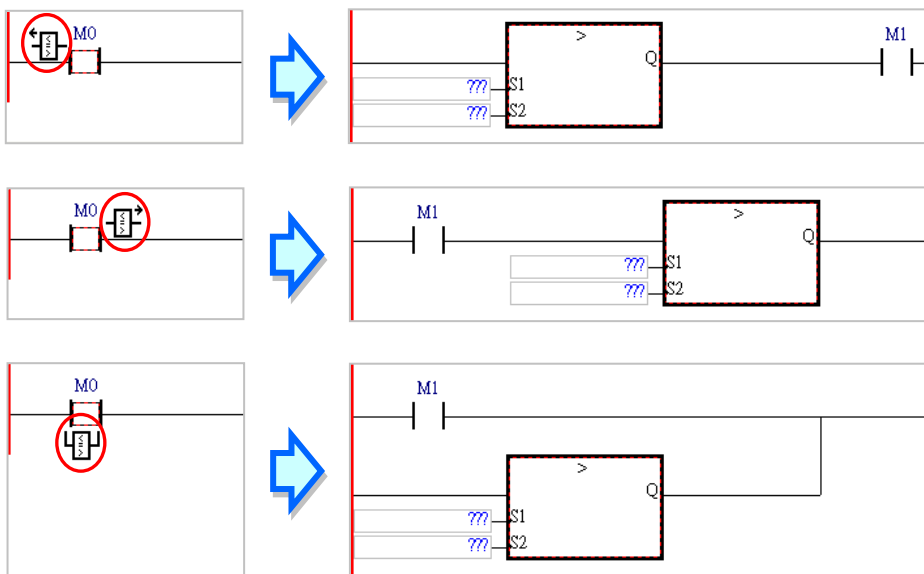
- (1) Click  on the toolbar, and then select a type.



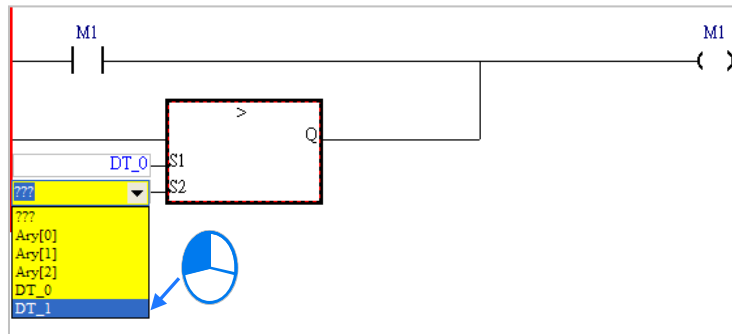
- (2) Select a line, a contact, or a group of devices. Before users connect a comparison contact to a group of devices in parallel or in series, they have to draw a frame round the group of devices. After users make sure of the position in which a comparison contact will be inserted, they can click  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.

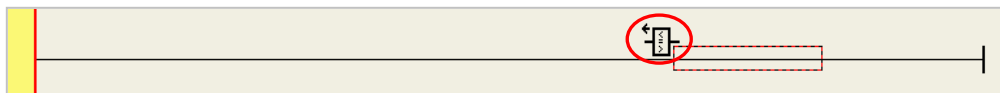


- (4) After an comparison contact is inserted successfully, users have to refer to section 10.2.2, and type operands.



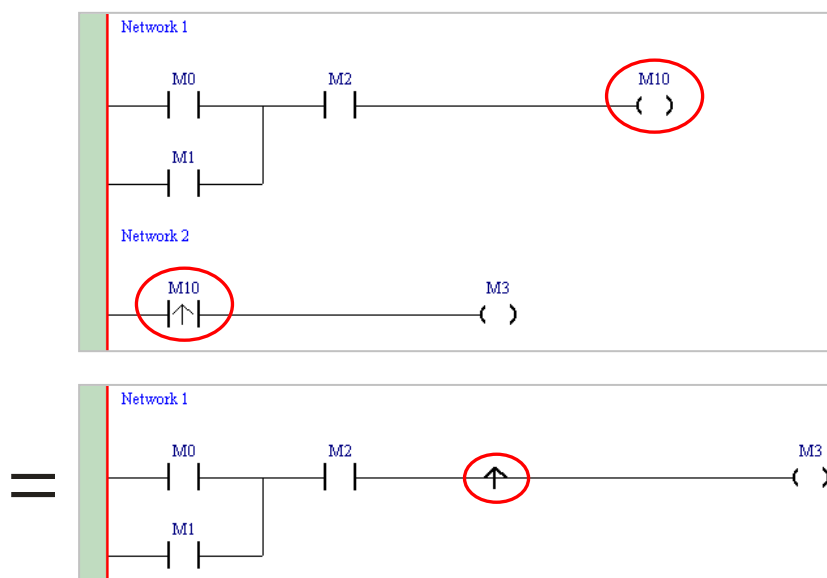
**Additional remark**

If users want to insert a comparison contact in a new network, the mouse cursor must be near the line selected.







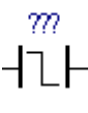
**10.2.6 Inserting a Block Logic Instruction**

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.



10


The following table provides block logic instructions in ISPSOft and the PLC types which can use these instructions and supports regarding POU program or function blocks.

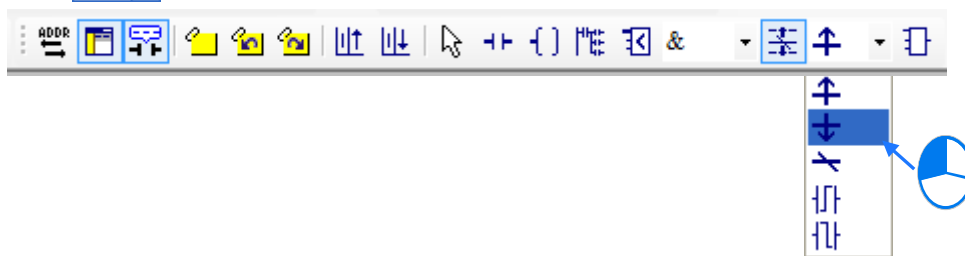
Icon	Instruction	Function	AH/AS		DVP		DVPxxMC	
			POU	FB	PO	FB	POU	FB
	NP	When the block preceding this block logic instruction is tuned from OFF to ON,	✓	✓	✓	✓		
	PN	When the block preceding this block logic instruction is turned form ON to OFF,	✓	✓	✓	✓		
	INV	Invert the preceding block logic status and output.	✓	✓	✓	✓	✓	✓
	FB_NP	The function is similar to NP, users need to specify a register device or		✓		Support ES3		
	FB_PN	The function is similar to PN, users need to specify a register device or		✓		Support ES3		

\* Please refer to AH/AS 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.

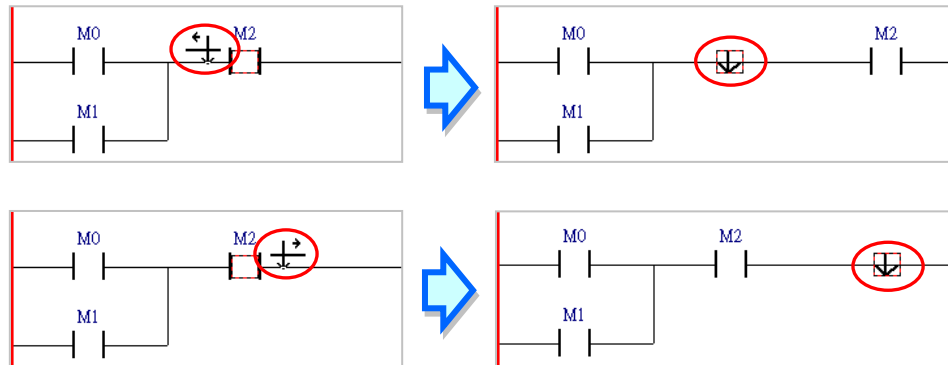


- (2) Click  on the toolbar.



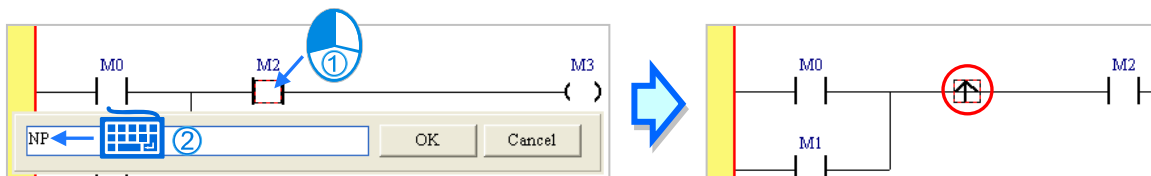


- (3) The mouse cursor appears as the representation of a block logic instruction when the mouse cursor is near the position in which a block logic instruction will be inserted. Please refer to the figures below. The mouse cursor appears as the representation of a block logic instruction when the mouse 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.



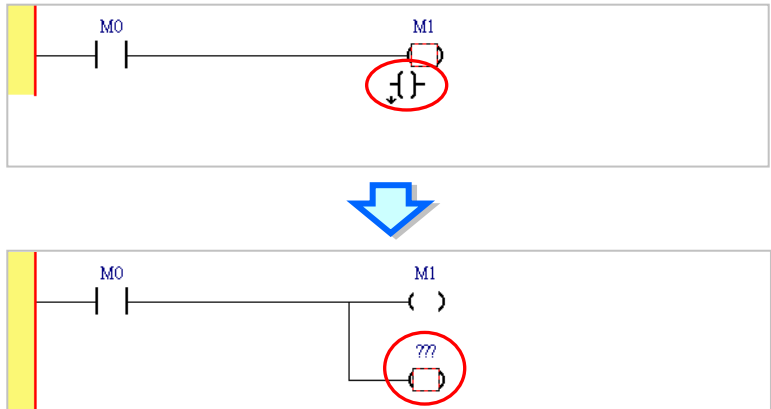
#### ● Method 2



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.)

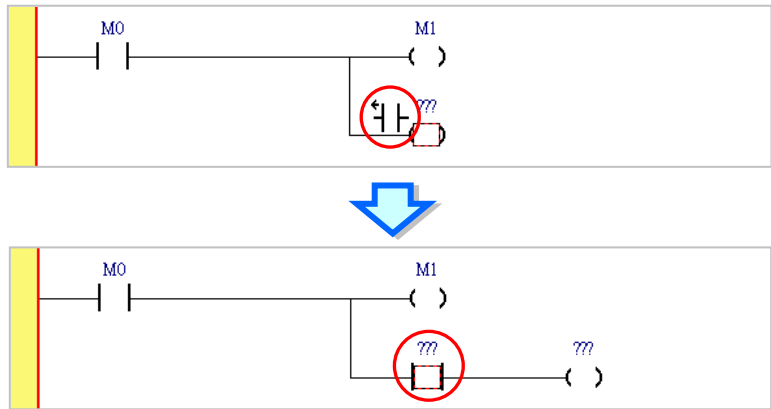


### 10.2.7 Creating Multiple Outputs

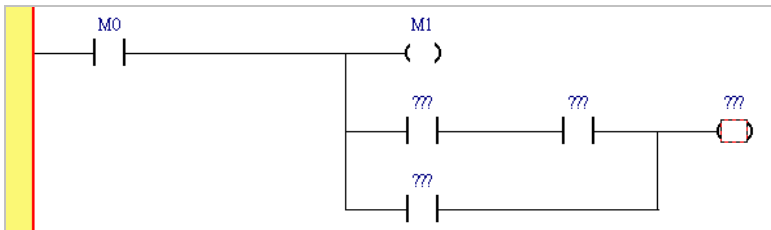
(1) If users want to create multiple outputs, they have to insert a coil or an applied instruction first.



(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.

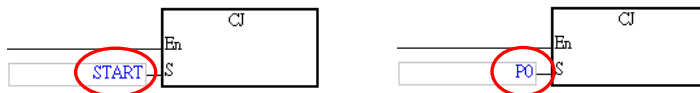


### 10.2.8 Putting a Label

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.

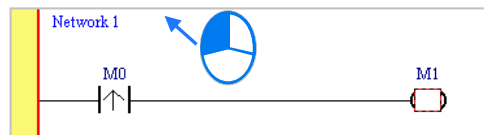


If users want to specify a label in an instruction, they do not have to type a colon in front of the label.

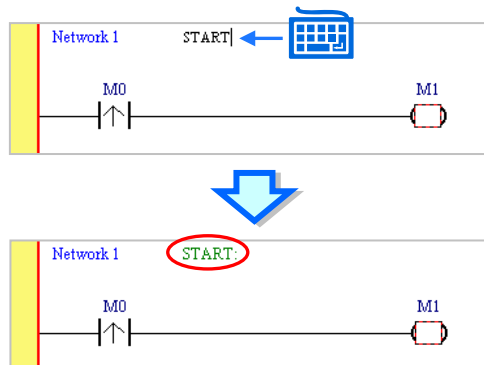


The steps of putting a label are as follows.

- (1) Click the right side of the network number at which a label will be put.

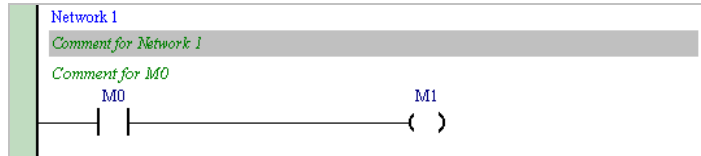


- (2) Users have to type a label symbol. Press Enter on the keyboard after a P device number or a label symbol is typed. The system will automatically put a colon at the right side of the label.





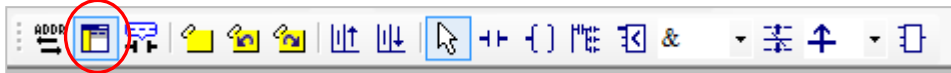
### 10.2.9 Editing a Comment

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.

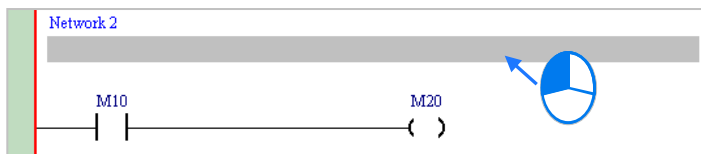


#### ● Editing the comment on a network

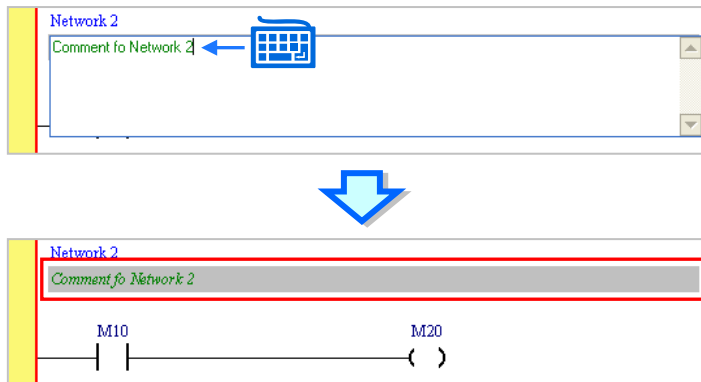
- (1) After users click  on the toolbar, the comments on the networks will be displayed. If the users click  on the toolbar again, the comments on the networks will be hidden.





- (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.

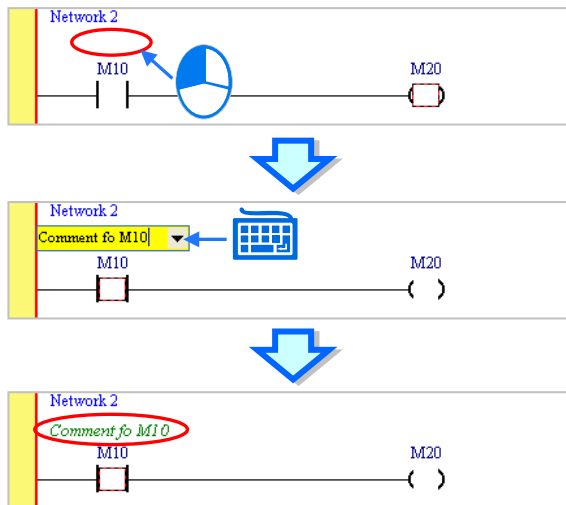


#### ● Editing the comment on a device


- (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.




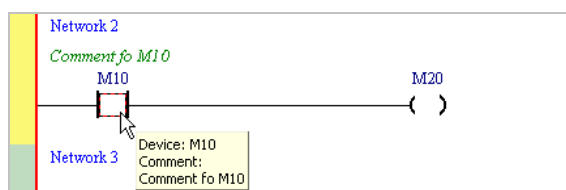
- (2) After the users click the position above a device name, they can type a comment in the box and also edit pin content of API. Press Enter on the keyboard after the editing is complete.





### 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.

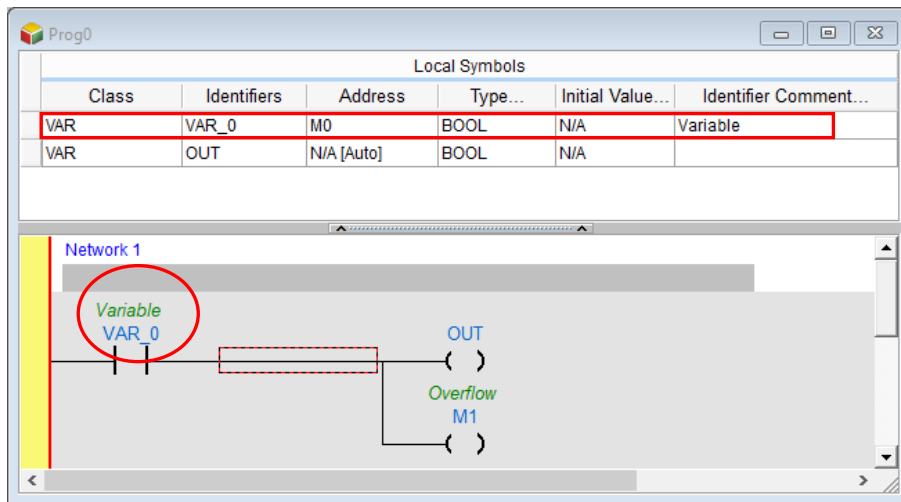
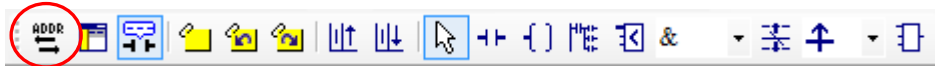
Besides, if  on the toolbar is clicked, the hints related to a device will appear after the mouse cursor stays at the device for a while. Users can select hints in the **Options** window in ISPSOft. Please refer to section 2.3.1 for more information.




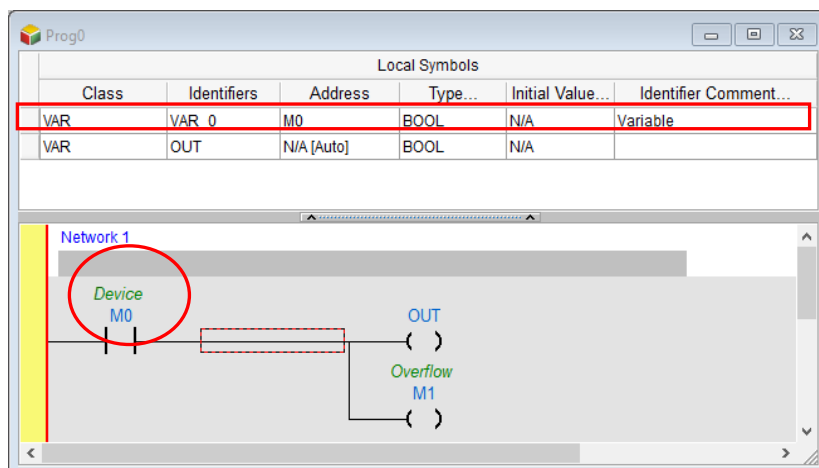
### 10.2.10 Symbol Mode and Address Mode

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.

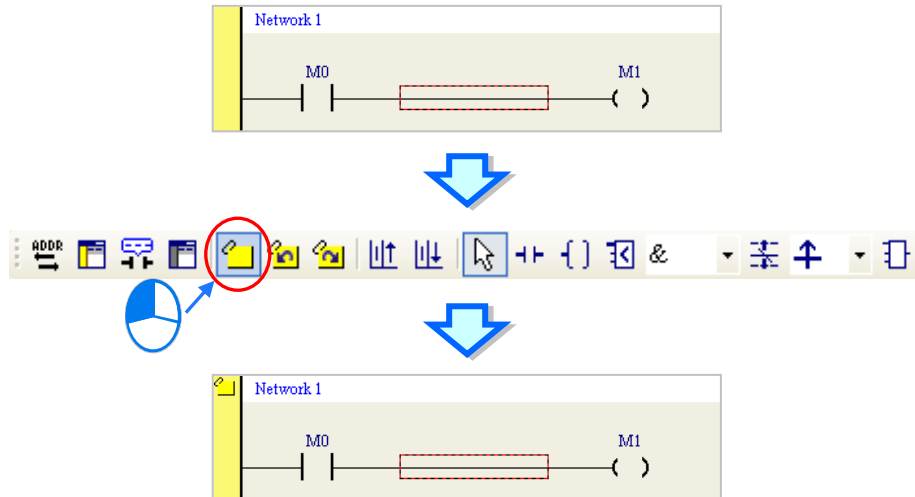




10

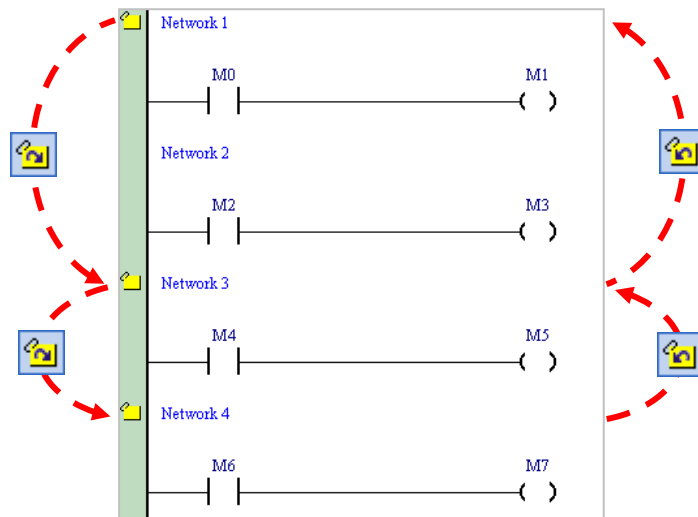
### 10.2.11 Bookmark

After networks are bookmarked, users can find or go to the networks easily.

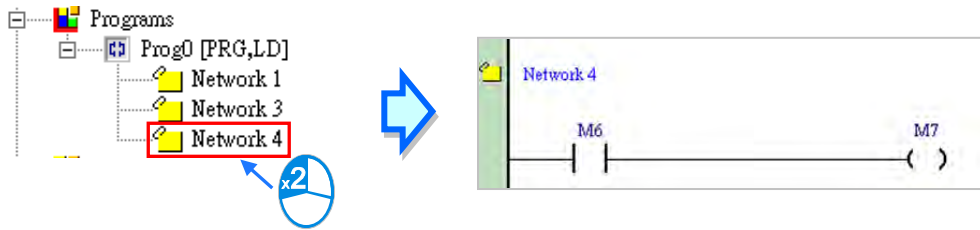
- (1) After users click  on the toolbar, a bookmark will be inserted in the network selected. If the users want to delete the bookmark, they can select the network, and click  on the toolbar again.



- (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.



- (3) You can also go to where the bookmarked is by clicking twice on the bookmarked network in the Project area.



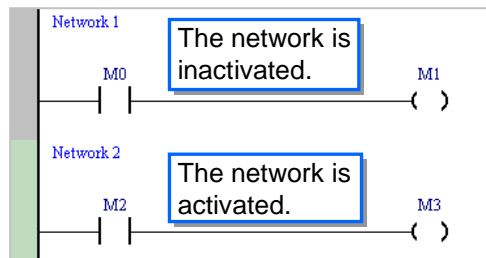
**Additional remark**

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.

**10.2.12 Activating/Inactivating a Network**

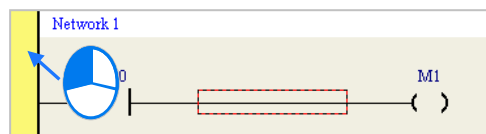
If a network is inactivated, the compiling of the program will skip the network. When users test or debug a program, they can inactivate some part of the program temporarily by means of this function.

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.



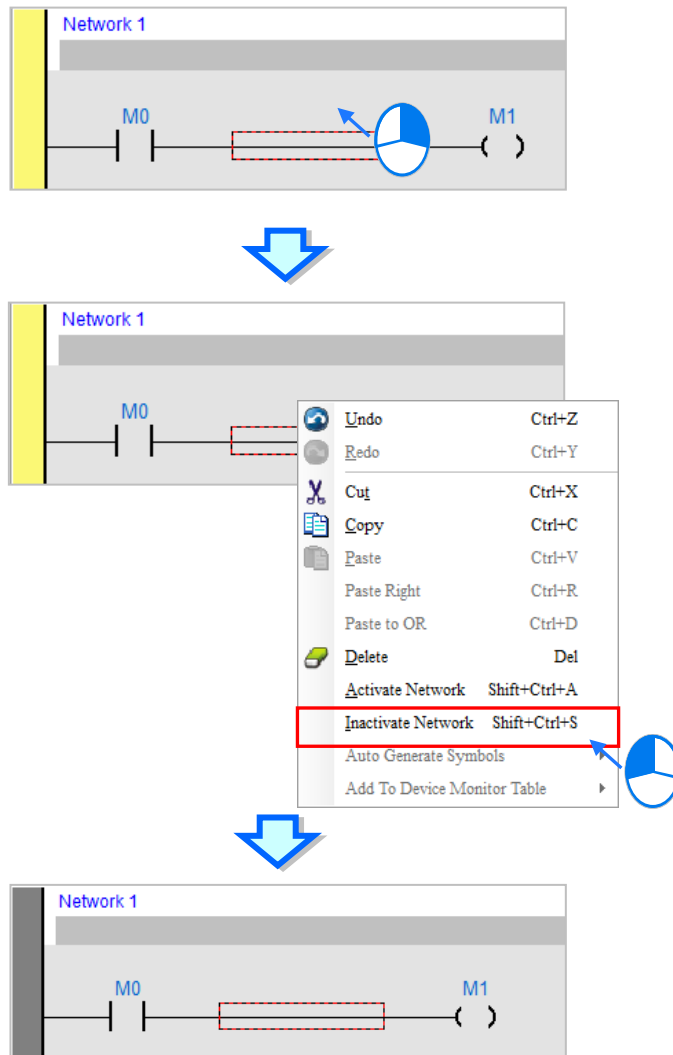
10

If users want to activate or inactivate a network, they have to select the network first. Users can select several networks simultaneously.

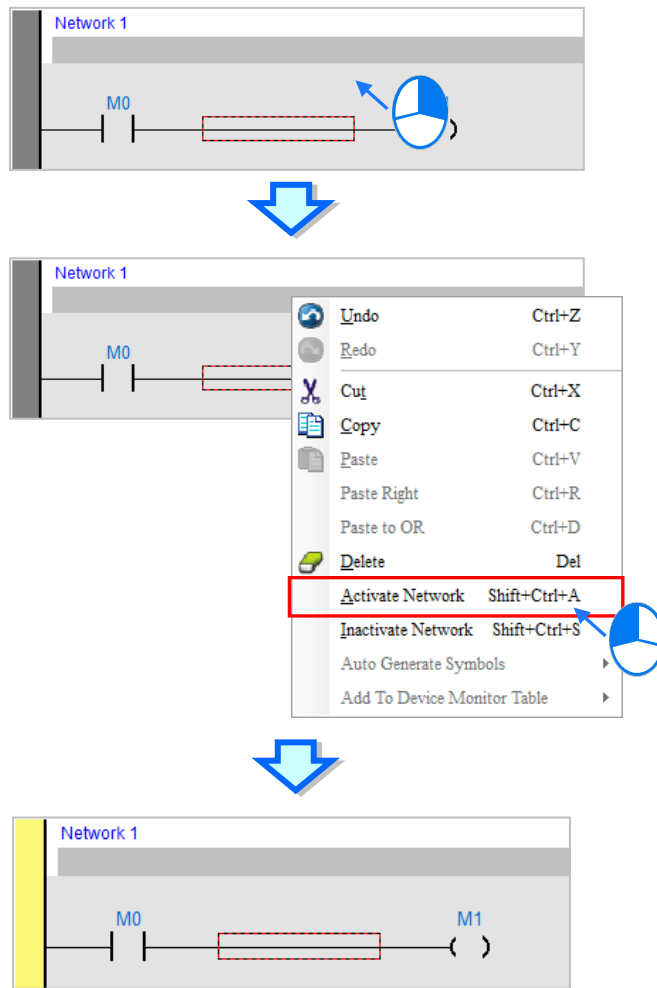


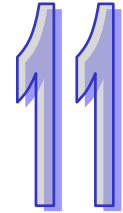


Click **Inactivate Network** on the **Edit** menu or right-click the network selected and click **Inactivate Network** on the context menu. The network selected will be inactivated.



If users want the inactivated network selected to be activated, they can click **Activate Network** on the **Edit** menu or right-click the network selected and click **Activate Network** on the context menu.





# Chapter 11 Function Block Diagram

## Table of Contents

<b>11.1</b>	<b>Introduction of Function Block Diagrams .....</b>	<b>11-2</b>
11.1.1	Knowing Function Block Diagrams.....	11-2
11.1.2	Editing Environment.....	11-2
11.1.3	Networks in a Function Block Diagram.....	11-4
11.1.4	Selecting Objects.....	11-5
<b>11.2</b>	<b>Creating a Function Block Diagram in ISPSoft .....</b>	<b>11-6</b>
11.2.1	Input Nodes and Output Nodes.....	11-7
11.2.2	Use Devices, Symbols Constants in FB Diagram.....	11-11
11.2.3	And and OR Block .....	11-12
11.2.4	Inverse Logic .....	11-15
11.2.5	Rising and Falling edge-triggered Input .....	11-17
11.2.6	Setting an Output and Resetting an Output.....	11-19
11.2.7	API, Comparison Contact and Function Block.....	11-21
11.2.8	Setting Label.....	11-26
11.2.9	Comments and Hints.....	11-27
11.2.10	Symbol Mode and Address Mode.....	11-29
11.2.11	Bookmark.....	11-30
11.2.12	Activating/Inactivating a Network .....	11-31

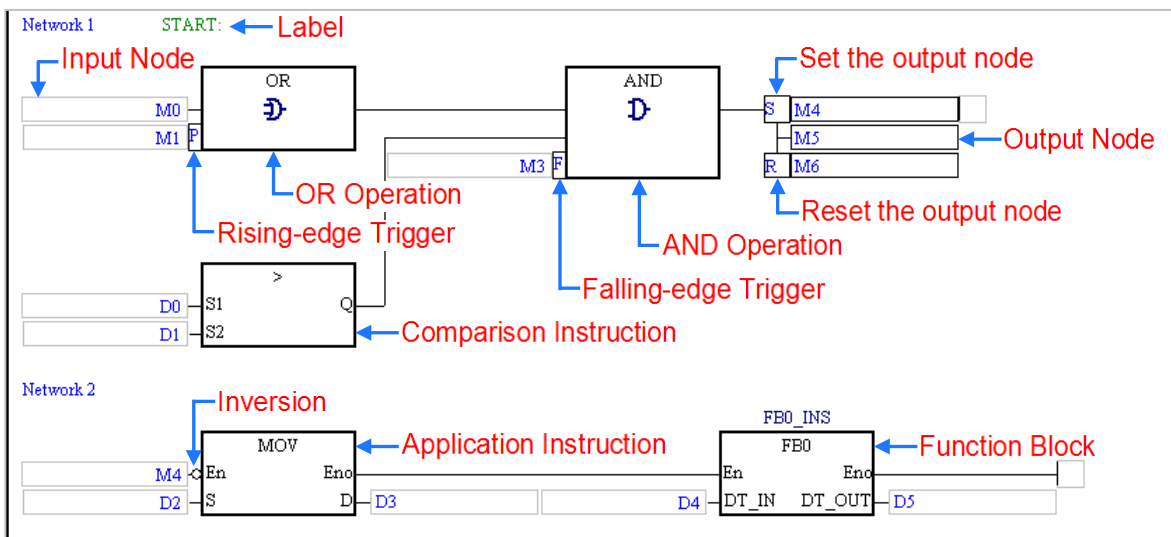
## 11.1 Introduction of Function Block Diagrams

### 11.1.1 Knowing Function Block Diagrams



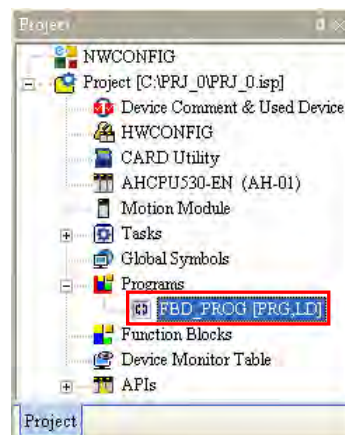
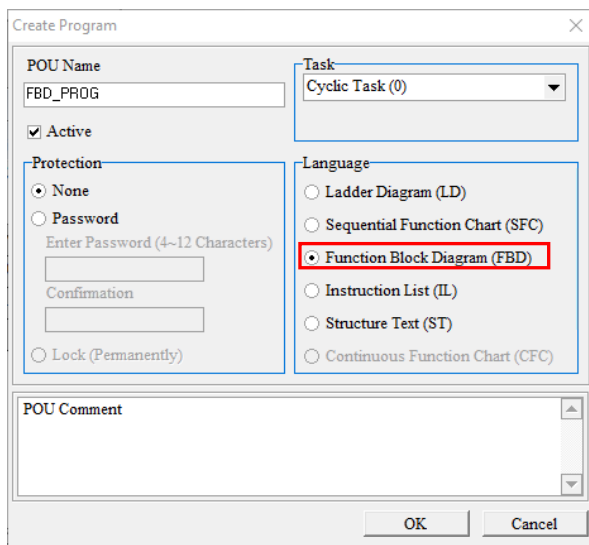
A function block diagram is one of the five languages defined by IEC 61131-3 standard. A function block diagram and a ladder diagram are graphical programming languages. A ladder diagram is a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay 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.

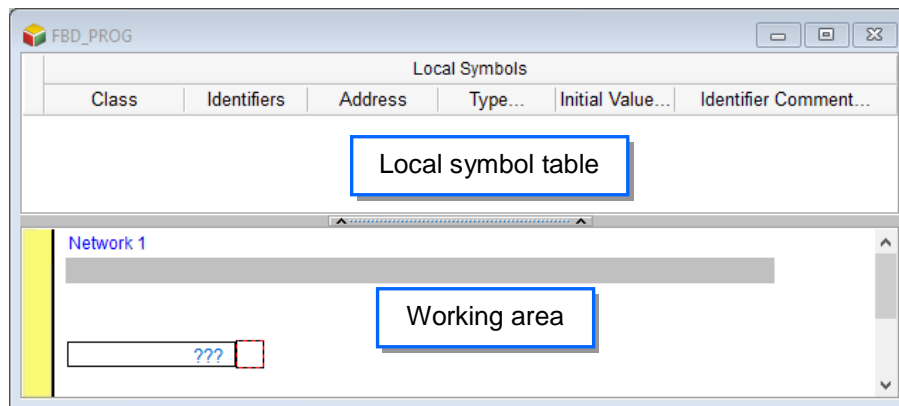


### 11.1.2 Editing Environment

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
	None	Displaying/Hiding the symbol or address
	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



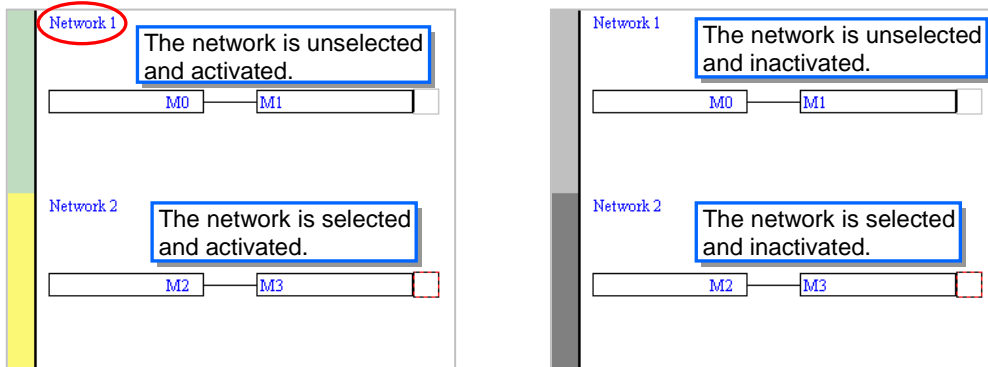
Icon	Keyboard shortcut	Function
	Shift+F7	Setting/Resetting the input
	Shift+F8	The input is rising edge-triggered/falling edge-triggered.
	Shift+Ctrl+U	Inserting an instruction or a function block

\*. Instructions and function blocks can be inserted by means of a keyboard. Please refer to section 11.2.7 for more information.

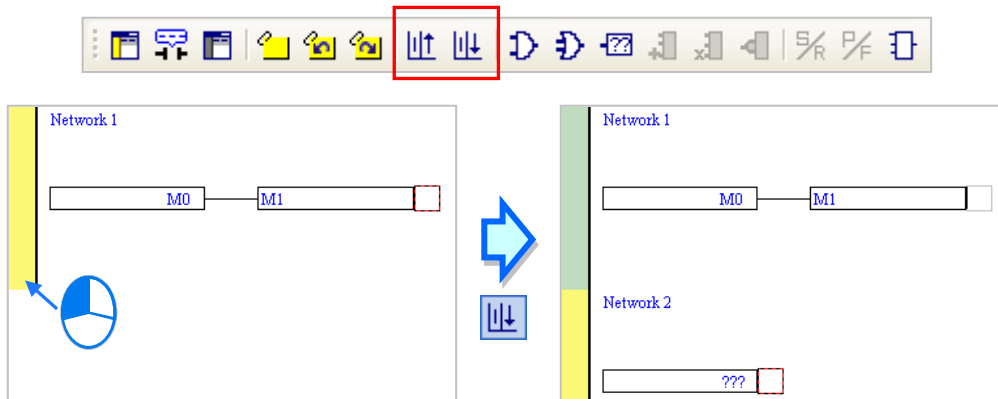
### 11.1.3 Networks in a Function Block Diagram

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.

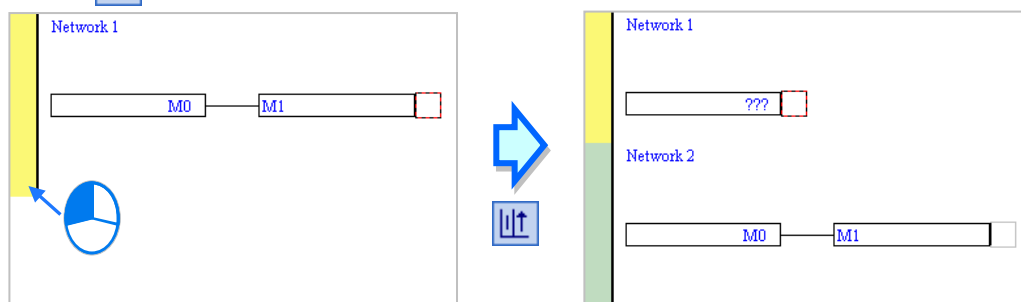
Network numbers are marked on the networks in a ladder diagram. The color at the left side of a network indicates the state of the network. A network can be activated or inactivated, and a network can be selected or unselected. Users can select colors which will be shown at the left sides of the networks 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.

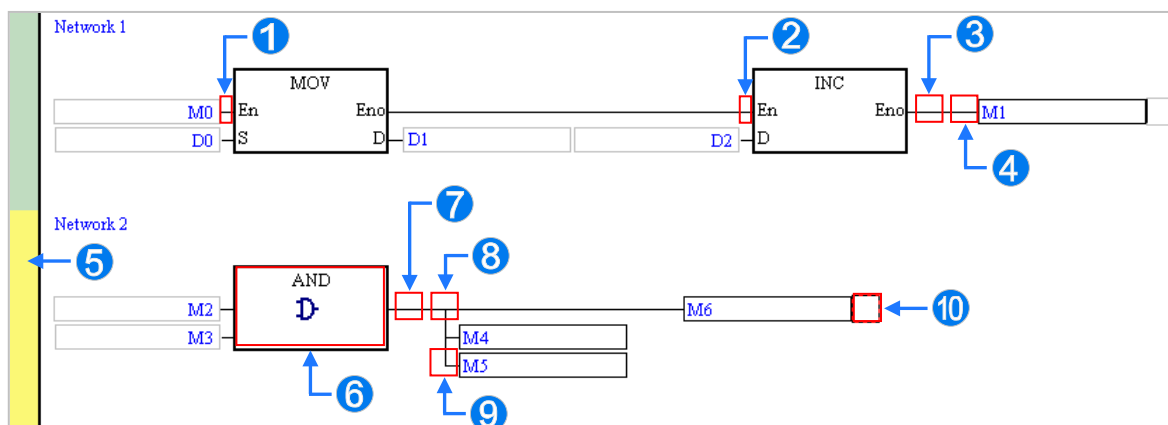


If users click  on the toolbar, a blank network will be put above the network selected.



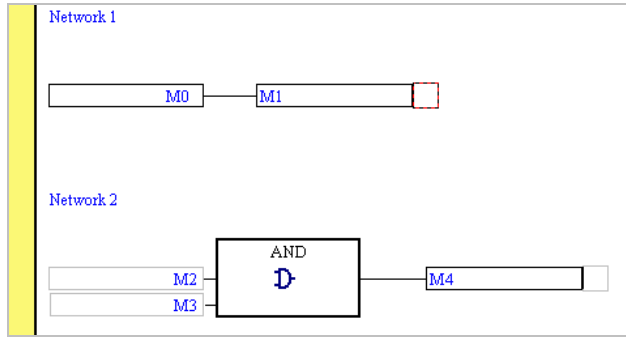
### 11.1.4 Selecting Objects

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 the 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.



- ① The input node at the right side of this position is selected. M0 in the figure is selected.
- ② The block before this position is selected.
- ③ All the blocks before this position are selected.
- ④ The output node at the right side of this position is selected. M1 in the figure is selected.
- ⑤ The whole network is selected.
- ⑥ The block is selected.
- ⑦ The block before this position is selected.
- ⑧ The output nodes under this position are selected.
- ⑨ The output node at the right side of this position is selected.
- ⑩ The block before this position is selected.

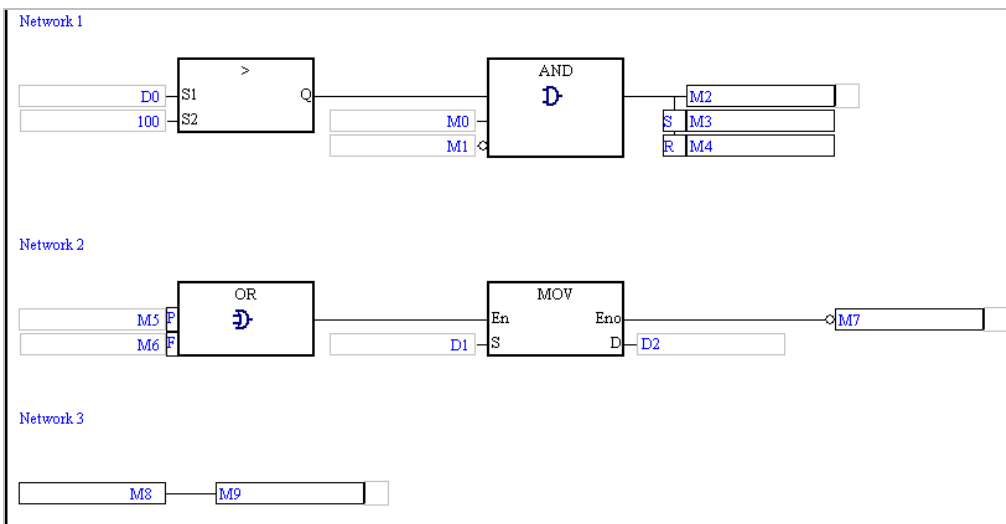
If users want to select several networks, they can hold down Ctrl on the keyboard while they click the networks. The users can also select a range of networks by pressing Shift on the keyboard, clicking the first network within the range, and the last network within the range.



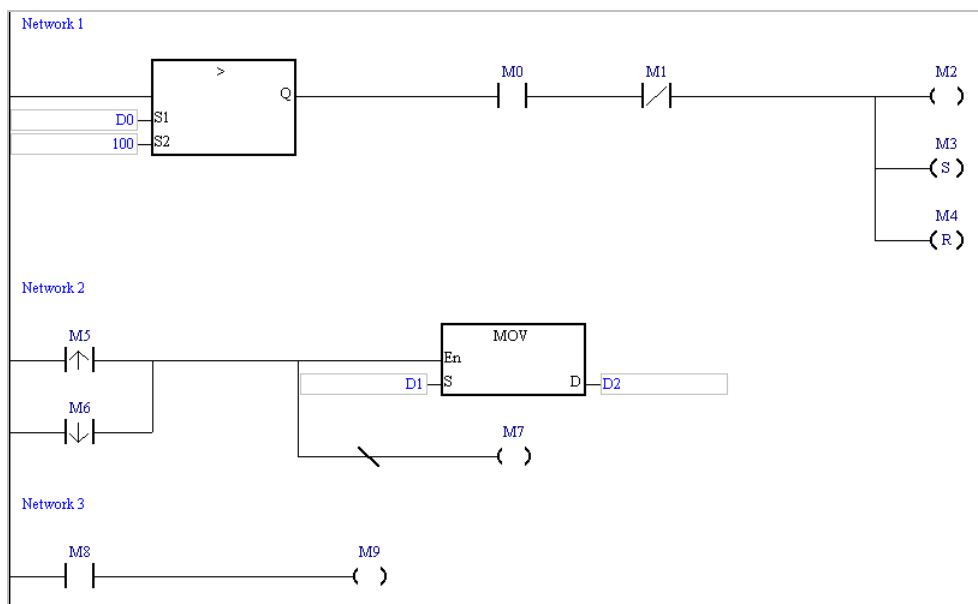
## 11.2 Creating a Function Block Diagram in ISPSoft

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



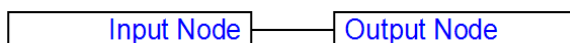


### 11.2.1 Input Nodes and Output Nodes

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

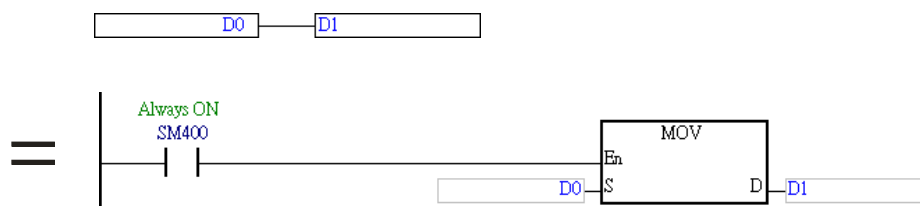


#### ● Example

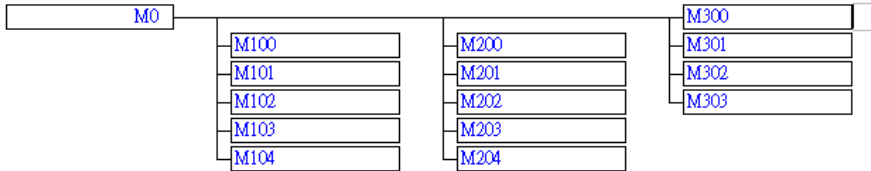
Example 1: The state of the contact M0 is sent to M1.





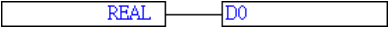

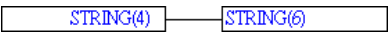


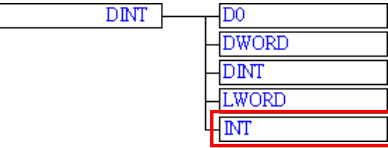
Example 2: The data in D0 is sent to D1.



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.

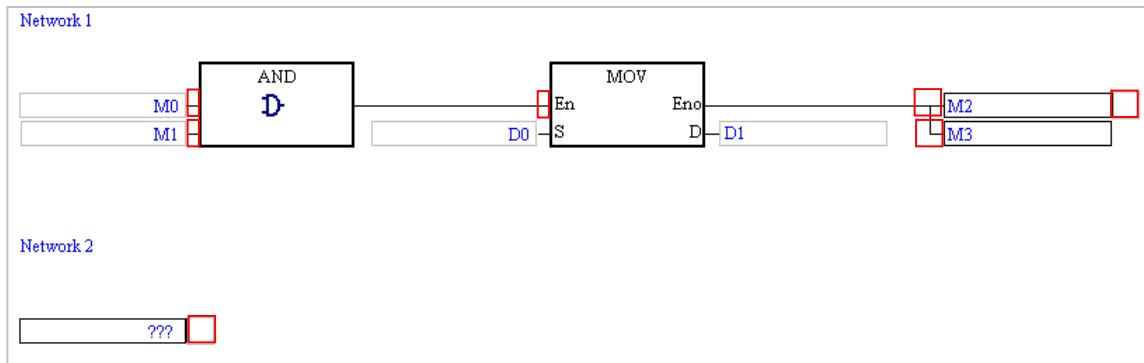




Example 4: Some incorrect examples are listed below.

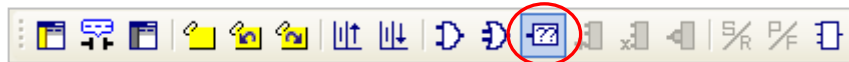
Incorrect example	Description
	The data type of the input node must not be different from the data type of the output node.
	The data length of the output node must not be less than the data length of the input node.
	The value of a symbol whose data type is REAL can not be sent to a D device.
	If the 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.
	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.

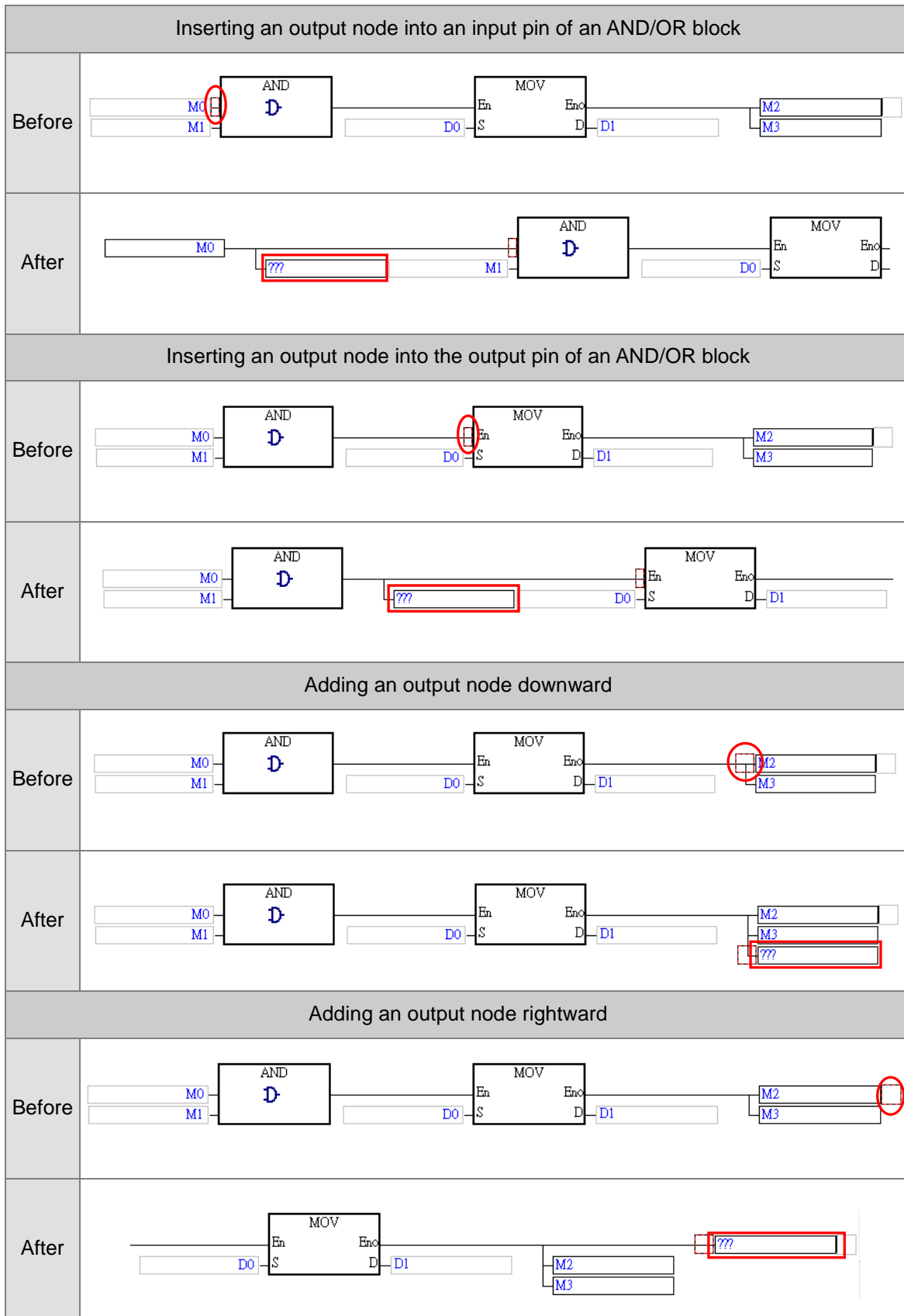


- (2) In addition to the structure in the figure above, users may be allowed to insert an output node into another structure. If users are allowed to insert an output node into the position selected,  on the toolbar can be clicked. After  is clicked, an output node will be inserted.



An output node can be inserted in the following structures.

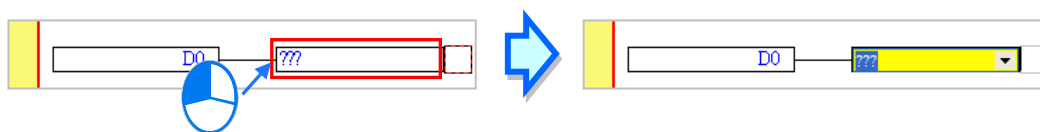
11




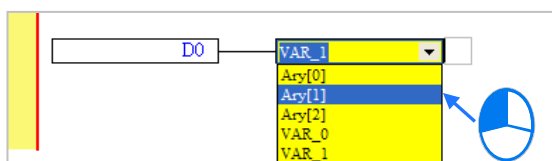
\*. Users are not allowed to insert output nodes into the output pins of a function block/an applied instruction to which operands are assigned.

## 11.2.2 Use Devices, Symbols Constants in FB Diagram

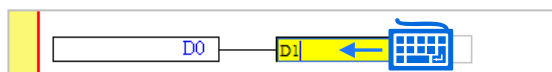
Before users edit a node or an operand of an instruction, they have to click the node or the operand.



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[...] 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.)
  - BOOL: Need to use normally-open or normally-closed contact

AH/AS Series: Use SM400/ SM401 .

DVP Series: M1000/ M1001.



If users want to use a symbol whose data type is ARRAY, the expression format is **Identifier[Index]**, for example Ary [1] from the drop-down list of the above diagram.

### 11.2.3 And and OR Block

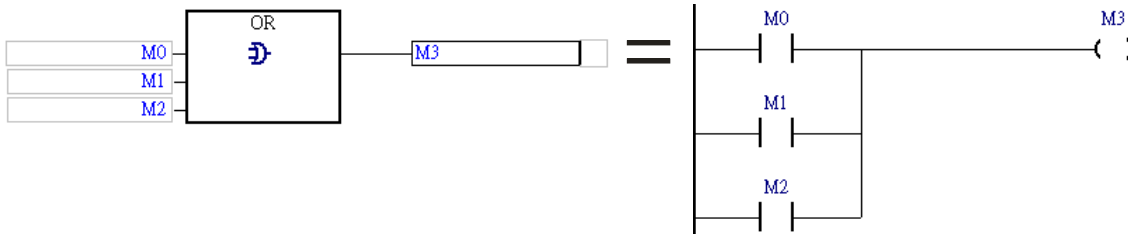
Contacts in a function block diagram can be connected in series/parallel by means of an AND/OR block. The input nodes and the output nodes of an AND/OR block must represent Boolean devices or symbols whose data types are BOOL. An AND/OR block have 32 input nodes and 32 output nodes at most. Besides, an AND/OR block has at least one output node, or is connected to the next object.

- **Graphic representation**

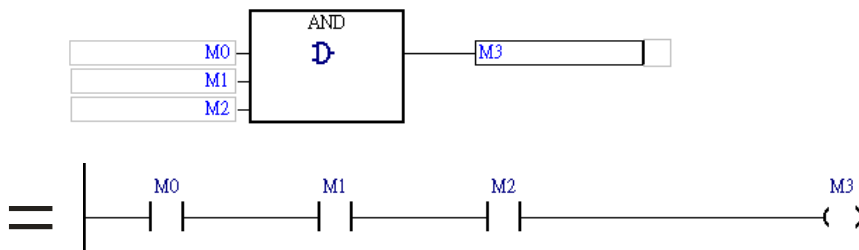


- **Example**

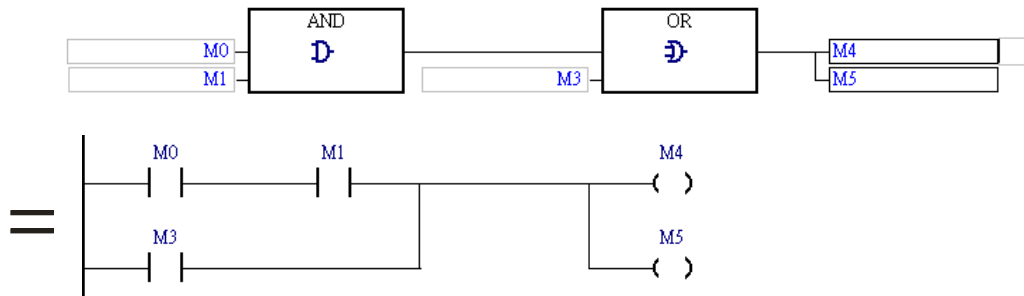
Example 1: The contacts M0, M1, and M2 are assigned to the OR block, and the result of the operation is sent to M3. In other words, after M0, M1, and M2 are connected in parallel, the coil M3 will be driven if M0, M1, or M2 is ON.



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.

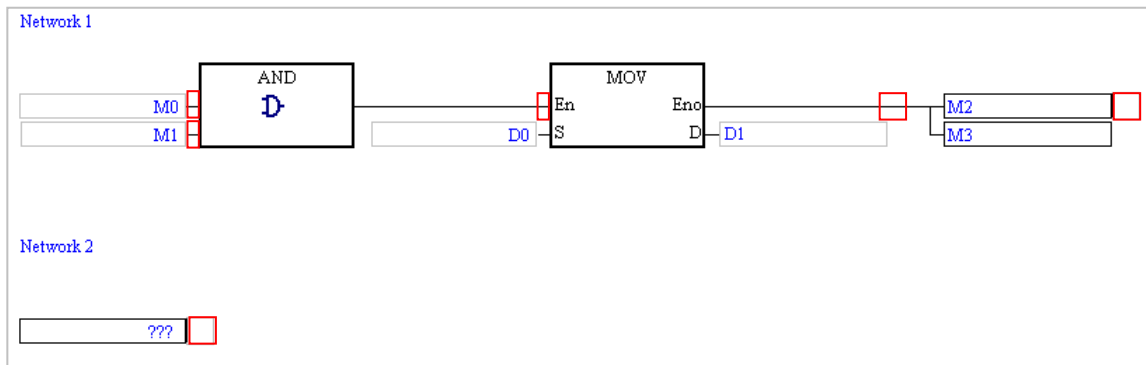




Example 3: The use of an AND block and an OR block in a function block diagram is equivalent to the use of a serial connection and a parallel connection in a ladder diagram.



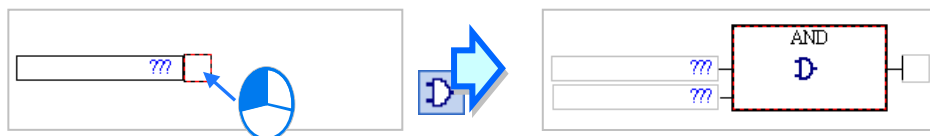
● **Inserting an AND/OR block**

- (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.



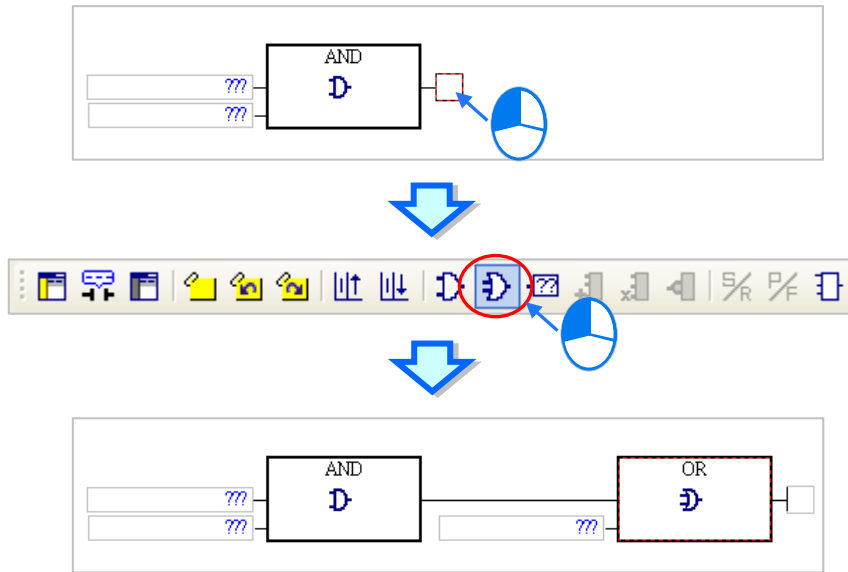
- (2) In addition to the structure in the figure above, users may be allowed to insert an AND/OR block into another structure. If users are allowed to insert an AND/OR block into the position selected,  /  on the toolbar can be clicked.

After  is clicked, an AND block will be inserted.





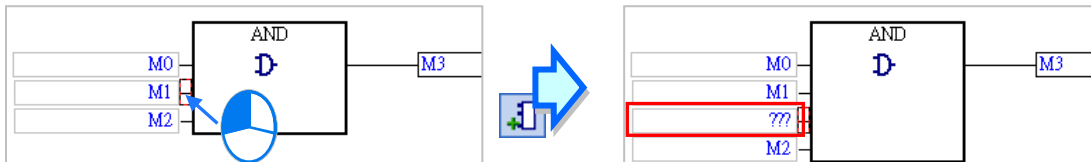
After  is clicked, an OR block will be inserted.


11

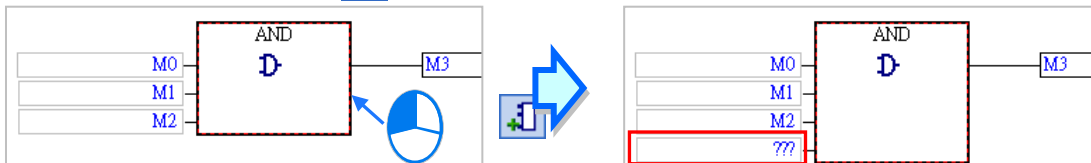


● **Adding an input node to an AND/OR block**



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.

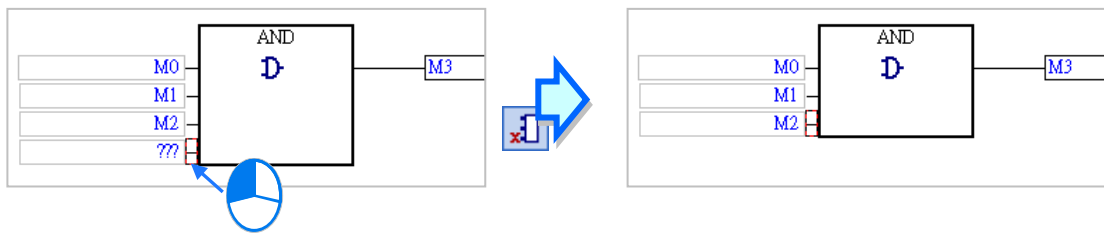


● **Deleting an input node from an AND/OR block**

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.





### 11.2.4 Inverse Logic

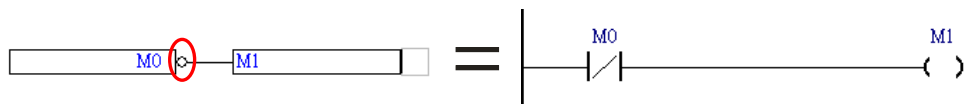
Users can insert inverse logic into a Boolean position in a function block diagram, e.g. an input node which represents a Boolean device or a symbol whose data type is BOOL, an output node which represents a Boolean device or a symbol whose data type is BOOL, the En pin of a function block, or the En 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**

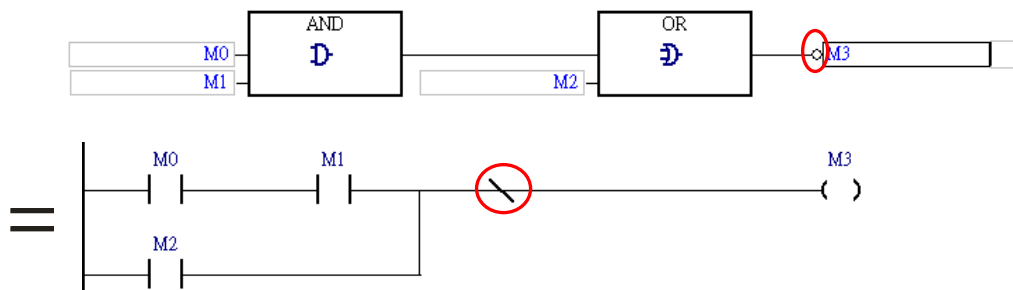


- **Example**

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.



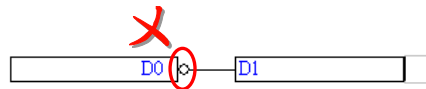
Example 2: The inverse logic inserted into the output node indicates that the result of the operation is inverted. The inversion result is sent to M3.



Example 3: If M0 is ON, the function block BLACK will be executed. If M0 is OFF, the function block WHITE will be executed.

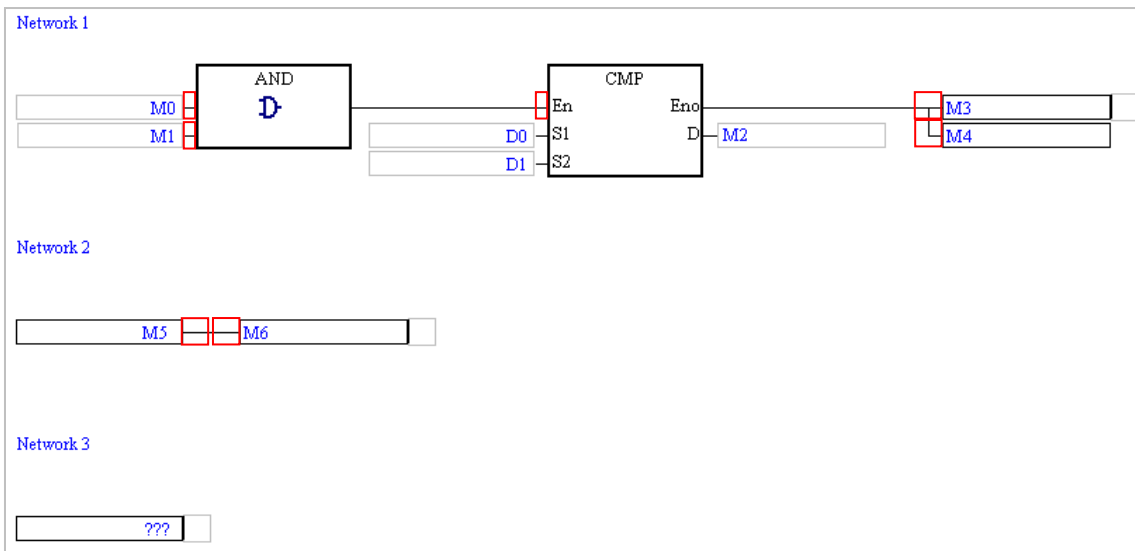




Example 4: The figure below is an incorrect example. Inverse logic can not be inserted into a node which does not represent a Boolean device or a symbol whose data type is BOOL.



● Inserting inverse logic

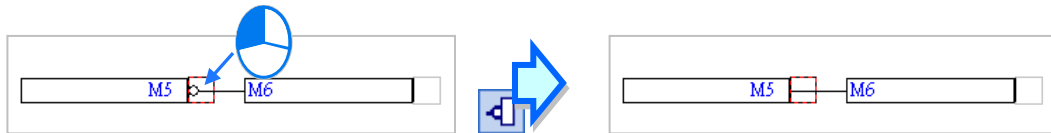
- (1) Select a position into which inverse logic will be inserted in the working area. Users are allowed to insert inverse logic into the positions marked with red frames in the figure below. Although the operand assigned to the output pin of the instruction CMP is a Boolean operand, inverse logic can 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.



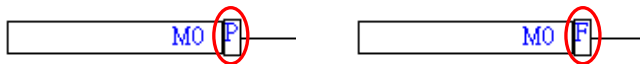
If the users want to cancel the inverse logic, they can click  again.



### 11.2.5 Rising and Falling edge-triggered Input

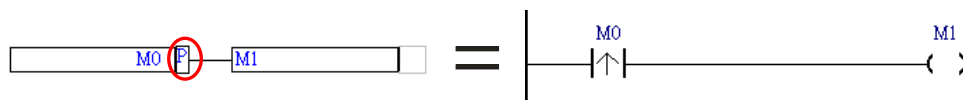
In a function block diagram in ISPSOft, the mark  on an input node indicates that the input node is rising edge-triggered, and the mark  on an input node indicates that the input node is falling edge-triggered. The mark  can only be inserted into an input node which represents a Boolean device or a symbol whose data type is BOOL, or an input node of an AND/OR block. Users are not allowed to insert  into the operands of a function block/an applied instruction, and the En pin of a function block/an applied instruction.


- **Graphical representation**

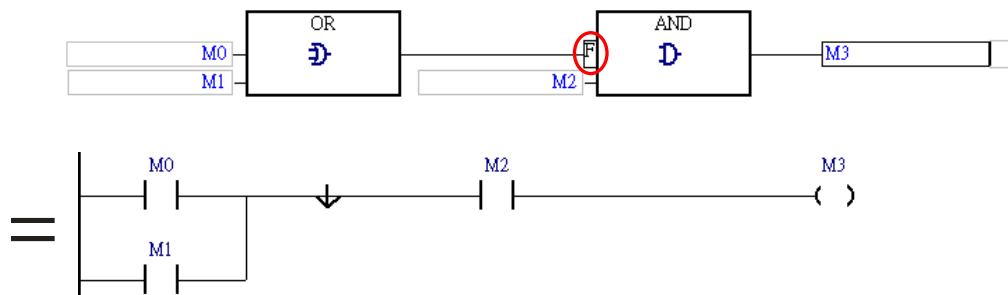


- **Example**

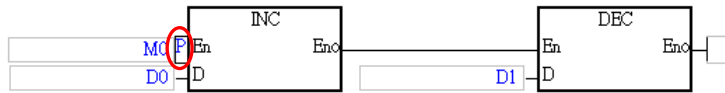
Example 1: When M0 is turned from OFF to ON, M1 is ON. When M0 is turned from ON to OFF, M1 is OFF.



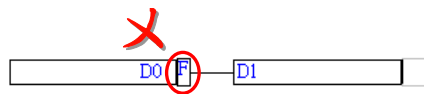
Example 2: The mark  is inserted into an input pin of the AND block.



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 the input node representing M0. However, the mark **P/F** can not be inserted into the En pin of the second applied instruction.

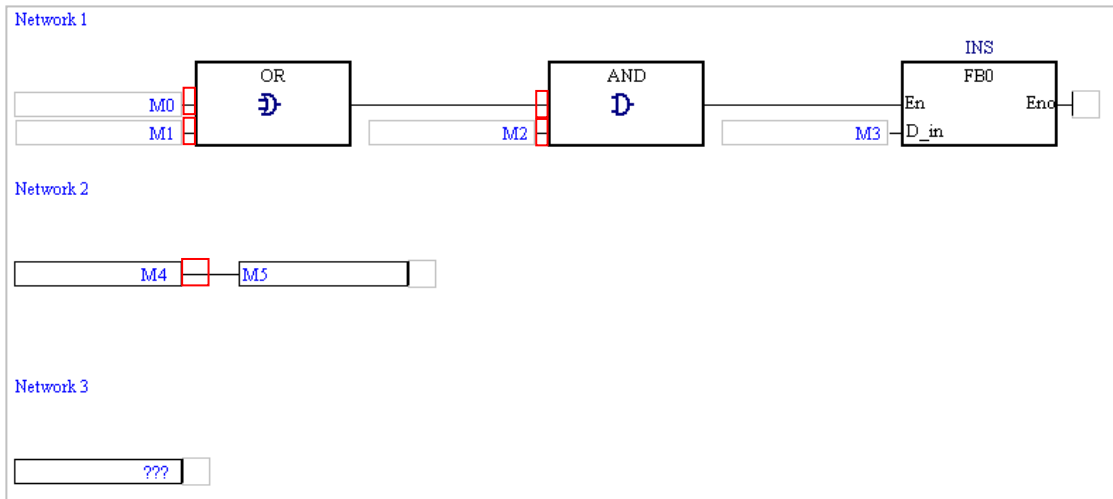


Example 4: The figure below is an incorrect example. The mark **F** can not be inserted into an input node which does not represent a Boolean device or a symbol whose data type is BOOL.

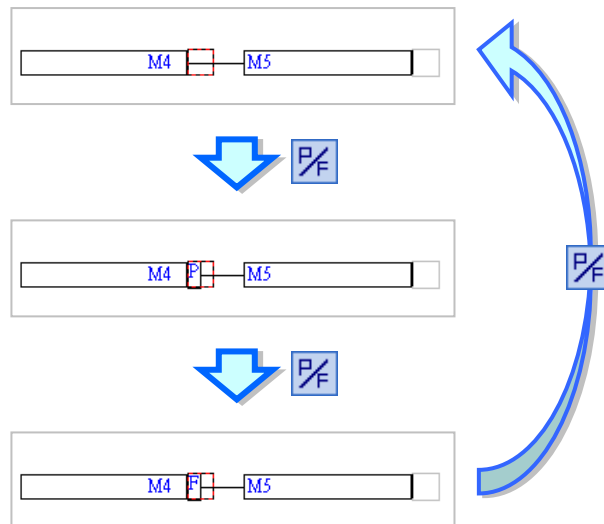
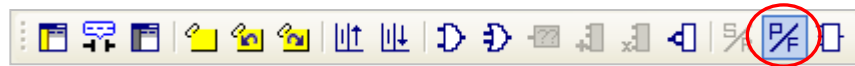


● **Inserting the mark **P/F****

- (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.



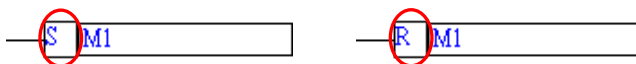
- (2) In addition to the structure in the figure above, users may be allowed to insert the mark  $\overline{P/F}$  into another structure. If users are allowed to insert the mark  $\overline{P/F}$  into the position selected,  $\overline{P/F}$  on the toolbar can be clicked. After users click  $\overline{P/F}$ , the mark  $\overline{P}$  will be inserted. If the users click  $\overline{P/F}$  again, the mark  $\overline{F}$  will be inserted.



### 11.2.6 Setting an Output and Resetting an Output

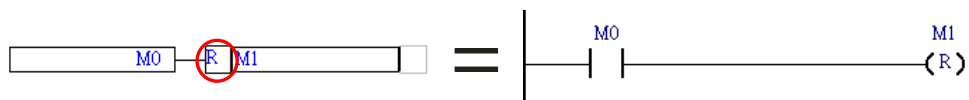
In a function block diagram in ISPSOft, the mark  $\overline{S}$  on an output node indicates that the output node is set to ON, and the mark  $\overline{R}$  on an input node indicates that the output node is reset to OFF. Users are not allowed to insert  $\overline{S/R}$  into the operands of a function block/an applied instruction.

- Graphical representation

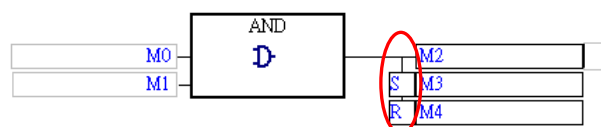


- Example

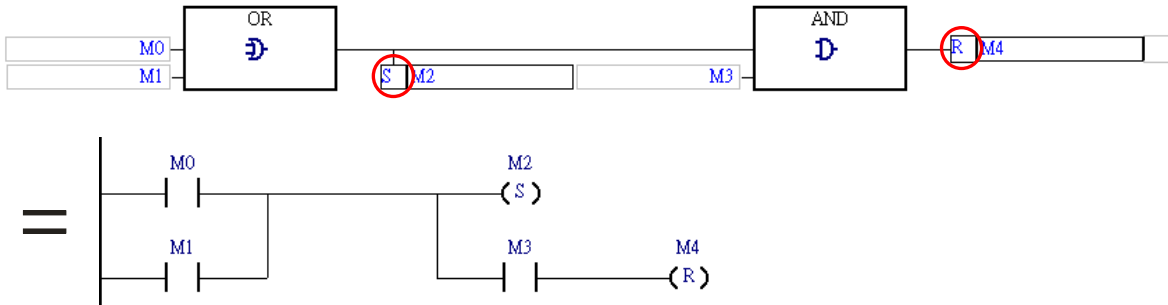
Example 1: If M0 is ON, M1 will be reset to OFF.



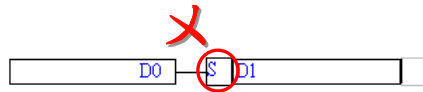
Example 2: In the figure below, the mark  $\overline{S}$  is inserted into the output node representing M3, and the mark  $\overline{R}$  is inserted into the output node representing M4.



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.

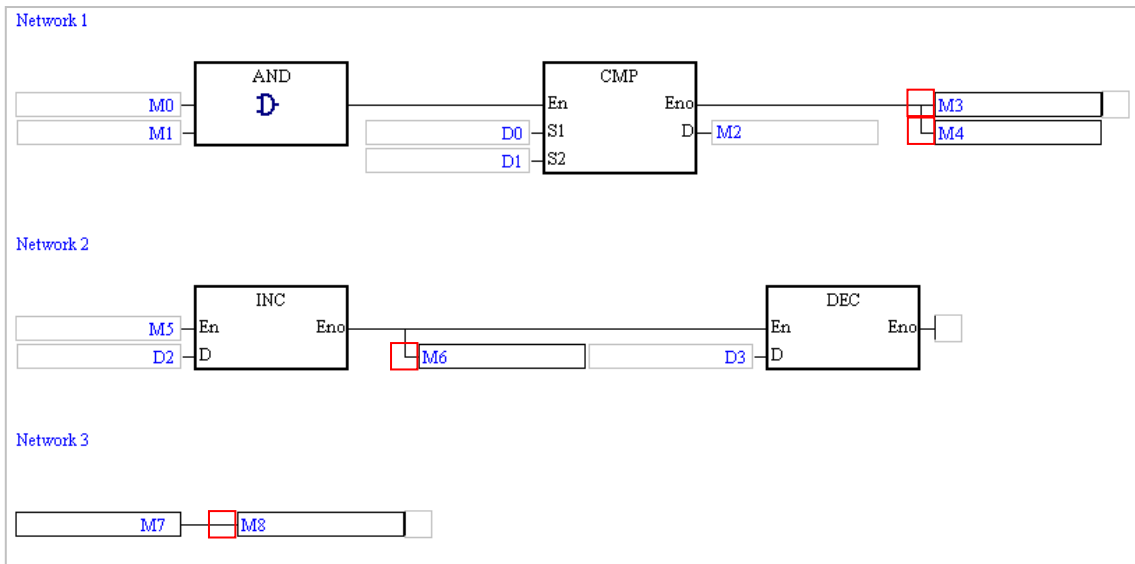


Example 4: The figure below is an incorrect example. The mark  $\boxed{S/R}$  can not be inserted into an output node which does not represent a Boolean device or a symbol whose data type is BOOL.

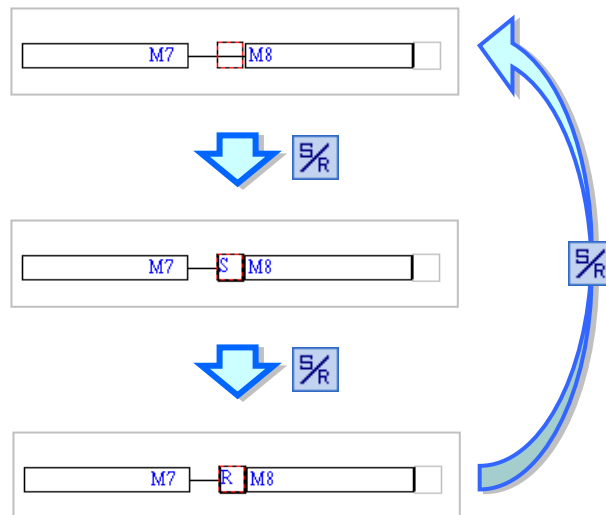


● Inserting the mark  $\boxed{S/R}$

- (1) Select a position into which the mark  $\boxed{S/R}$  will be inserted in the working area. Users are allowed to insert the mark  $\boxed{S/R}$  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  $\boxed{S/R}$  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  $\boxed{S/R}$  into another structure. If users are allowed to insert the mark  $\boxed{S/R}$  into the position selected,  $\boxed{S/R}$  on the toolbar can be clicked. After users click  $\boxed{S/R}$ , the mark  $\boxed{S}$  will be inserted. If the users click  $\boxed{S/R}$  again, the mark  $\boxed{R}$  will be inserted.



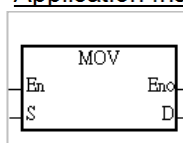
### 11.2.7 API, Comparison Contact and Function Block

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.

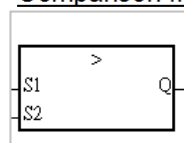
If the logic state connected to the En pin of a block representing an applied instruction or a function block is ON, the applied instruction or the function block will be executed. A function block diagram is flexible in that users can put a logical operation, e.g. inverse logic or an AND/OR block, between two function blocks, between two applied instructions, or between a function block and an applied instruction in a network.

● **Graphic representation**

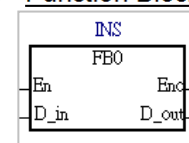
➤ Application Instruction



➤ Comparison Instruction

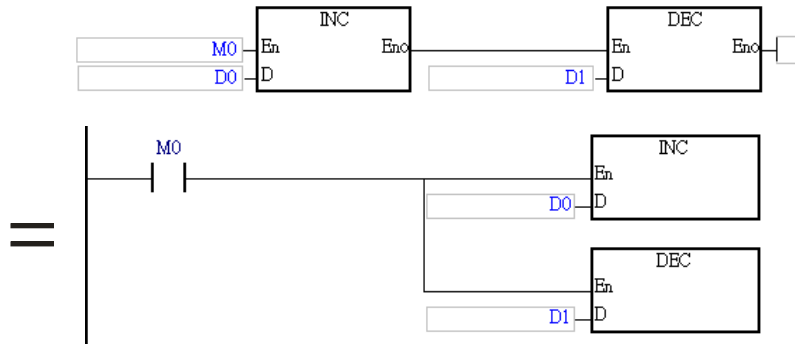


➤ Function Block

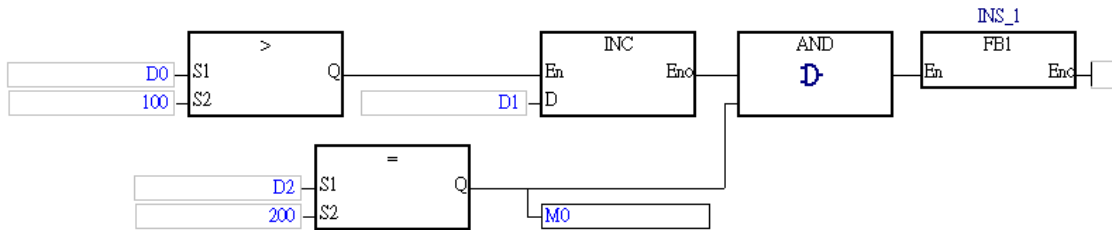


● **Example**

Example 1: The En pin of a block representing an applied instruction in a function block diagram can be connected to another instruction or object.



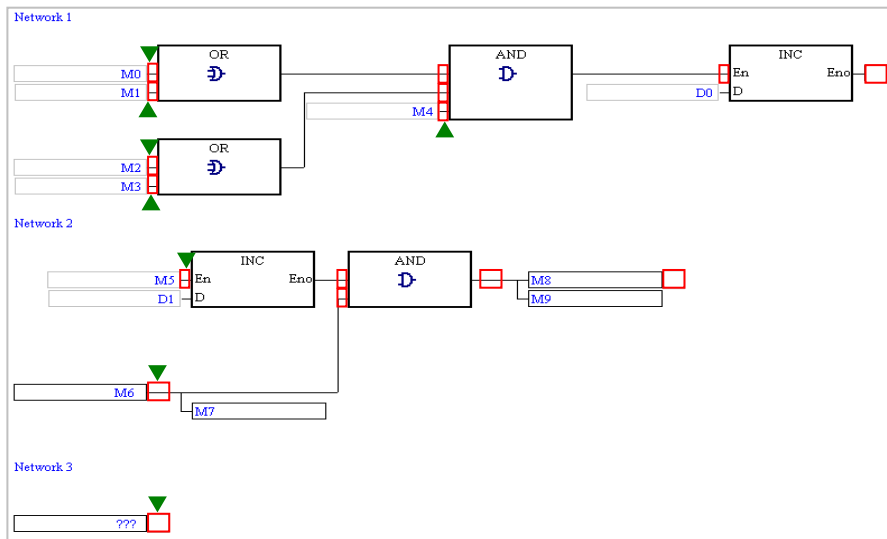
Example 2: In a function block diagram, no object can precede a comparison contact, but the output pin of the comparison contact can connect to an input pin of an AND/OR block, an output node, the En pin of a block representing a function block, or the En pin of a block representing an applied instruction.



\*. 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.

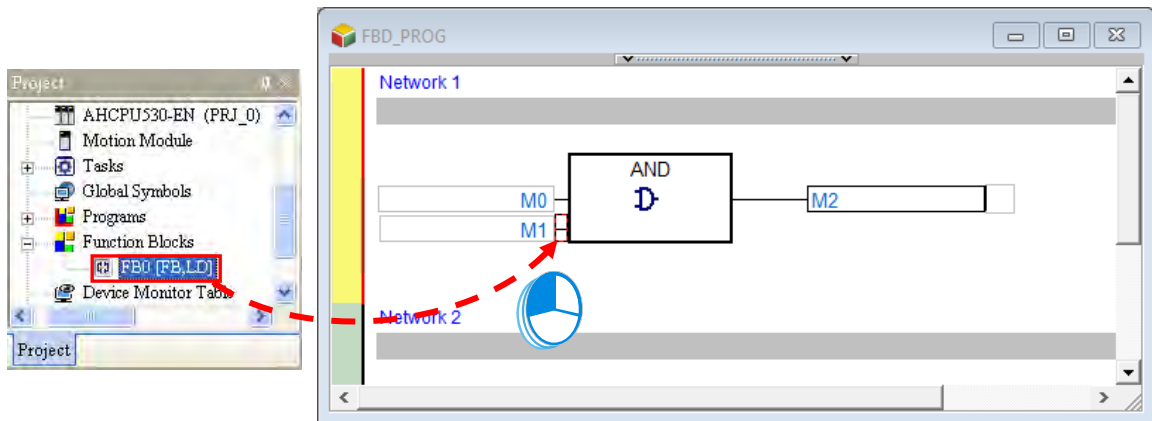




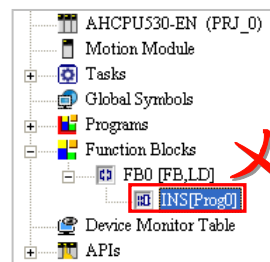
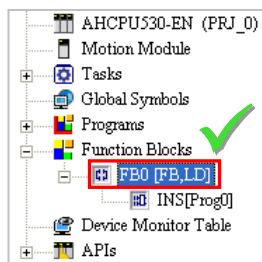
- (2) There are three ways to insert an applied instruction, a comparison contact, or a function block in a function block diagram.

➤ **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.



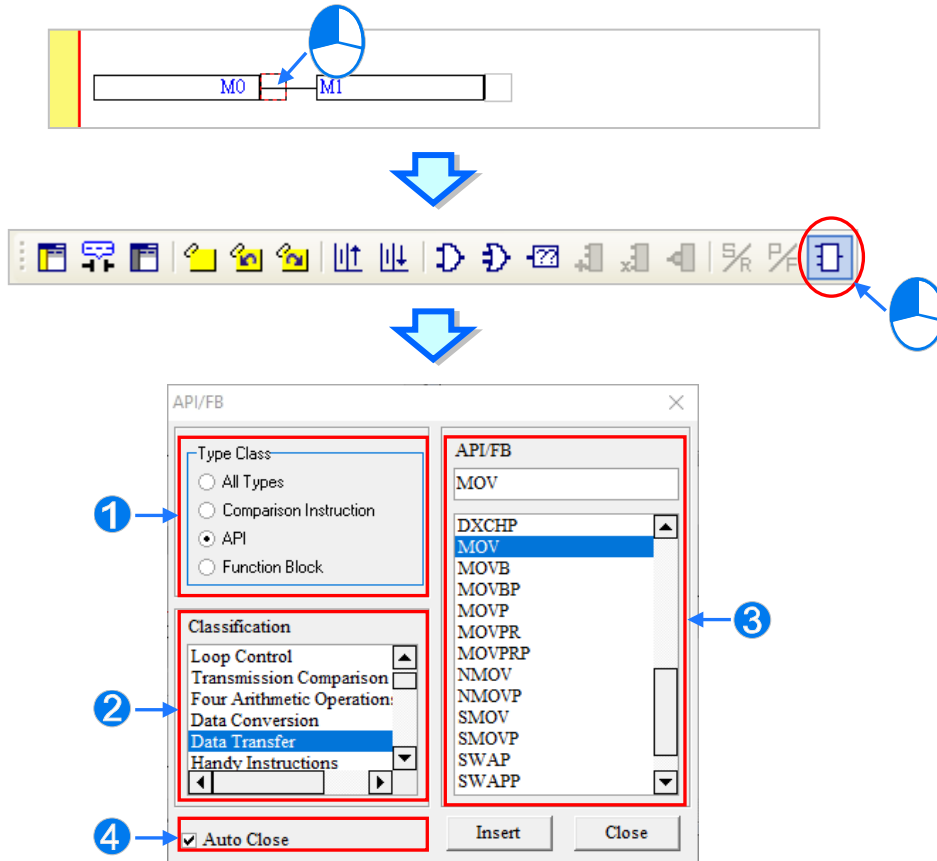
Function block definitions can be dragged, but function block instances can not be dragged.



➤ **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**.

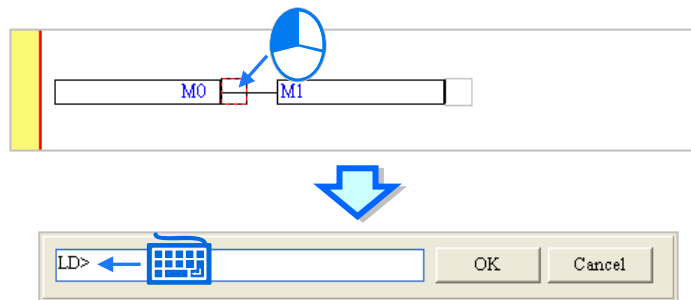
11



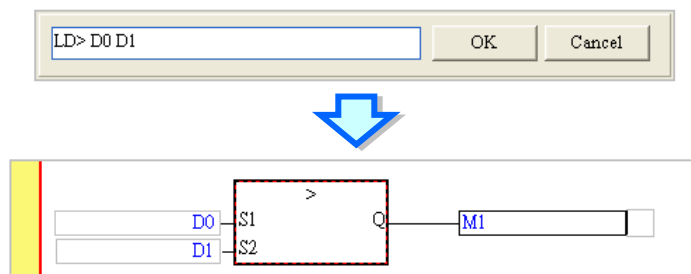
- ➊ Users can select the **All Types** option button, the **Comparison Instruction** option button, the **API** option button, or the **Function Block** option button.
- ➋ If users select the **API** option button in the **Type Class** section, they have to select an instruction type in the **Classification** box.
- ➌ Users can select an applied instruction, a comparison contact, or a function block here.
- ➍ If users unselect the **Auto Close** checkbox, they can insert instructions or function blocks continuously.

### ➤ Method 3

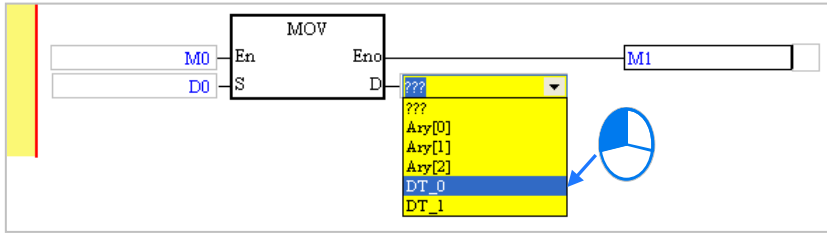
Click the position into which an applied instruction, a comparison contact, or a function block will be inserted, and type the applied instruction, the comparison contact instruction, or the function block definition. As soon as the applied instruction, the comparison contact instruction, or the function 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 specified in the function block. If users want to insert a comparison contact or an applied instruction, they have to type the comparison contact instruction or the applied instruction, and they can type the operands specified in the comparison contact instruction or the applied instruction. The users do not have to type all the operands specified in the comparison contact instruction 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.



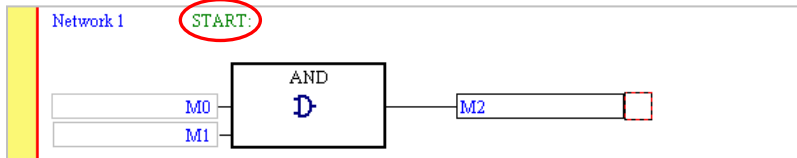
- (3) After an applied instruction, a comparison contact, or a function block is inserted successfully, users have to type operands.



\*. 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.

### 11.2.8 Setting 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. When choosing DVP series or AH/AS series, since the P device is automatically assigned by the system and cannot be directly used by users, input label name must not be repeated and the label symbols do not need to be declared in the symbol table. For more information on using P device and limitations, please refer to user manuals of PLC types.

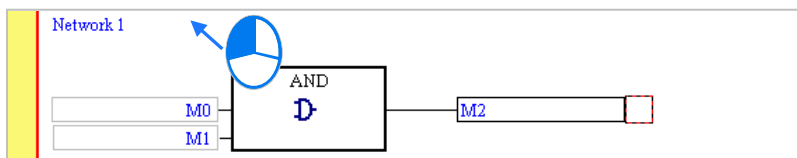


If users want to specify a label in an instruction, they do not have to type a colon in front of the label.

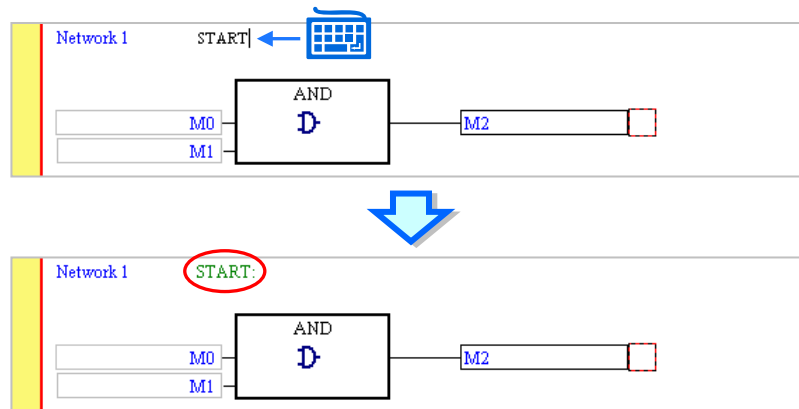


The steps of putting a label are as follows.

- (1) Click the right side of the network number at which a label will be put.

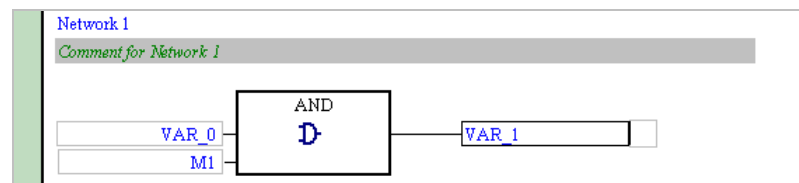


- (2) If the project created is a project for a DVP or AH/AS series CPU module, users have to type a label symbol. Press Enter on the keyboard after a P device number or a label symbol is typed. The system will automatically put a colon at the right side of the label.





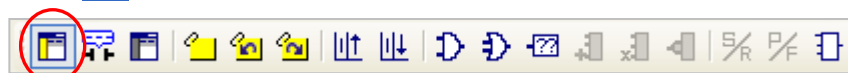
### 11.2.9 Comments and Hints

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.

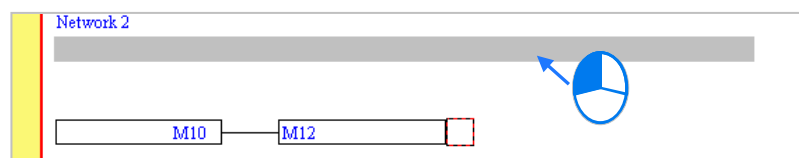


- **Editing the comment on a network**

- (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.

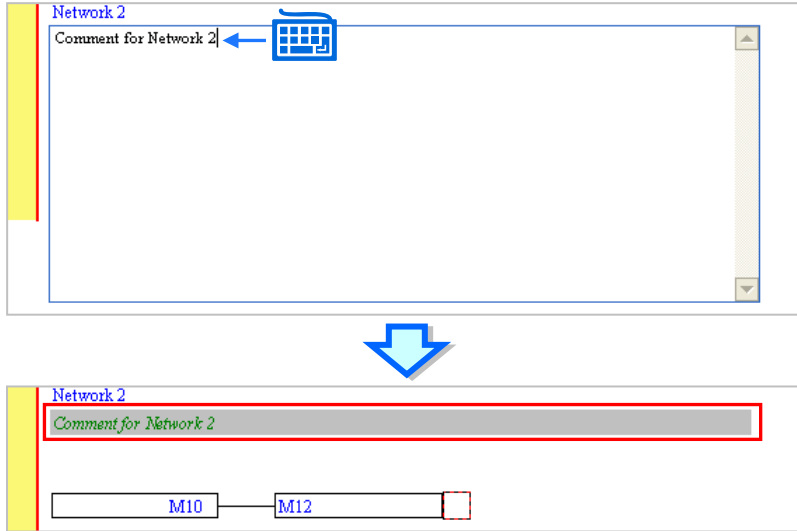


- (2) After the users click the position under a network number, a box will appear.





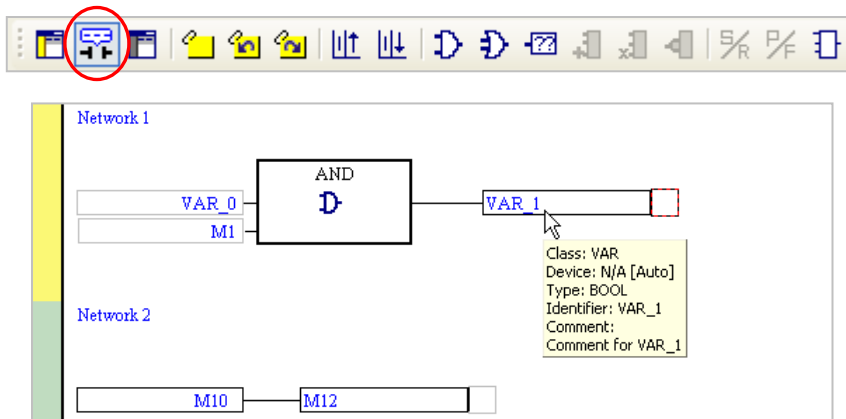
- (3) Type a comment in the box. If the users want to start a new line of text at a specific point, they can press Shift+Enter on the keyboard. Press Enter on the keyboard after the editing is complete.

11

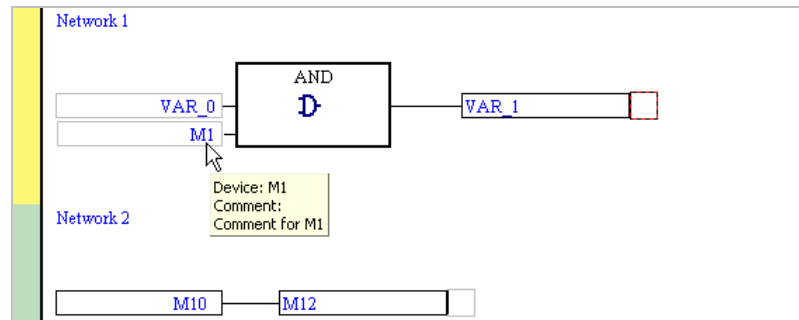


● **Hints related to the devices and the symbols in a function block diagram**

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.




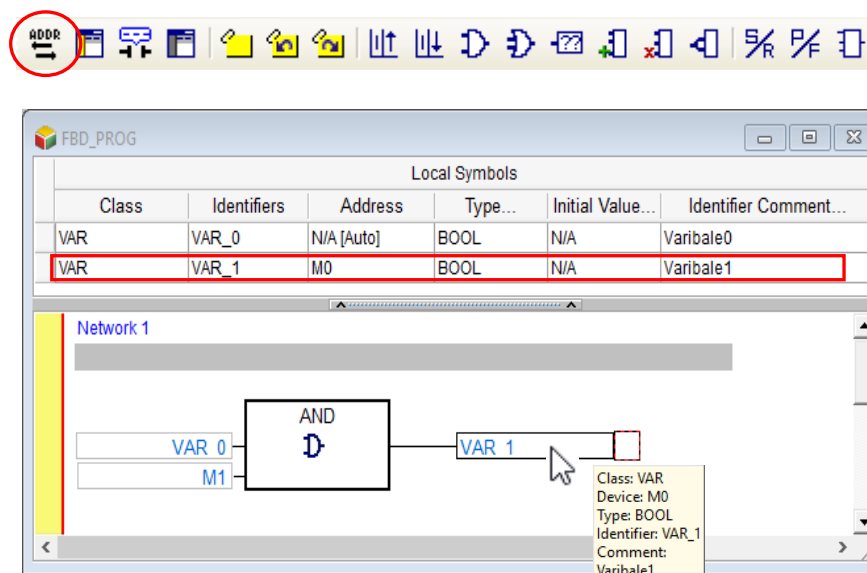
### Additional remark


Users can make a comment on a device by means of the comment management tool provided by ISPSOft. Please refer to section 17.2 for more information.

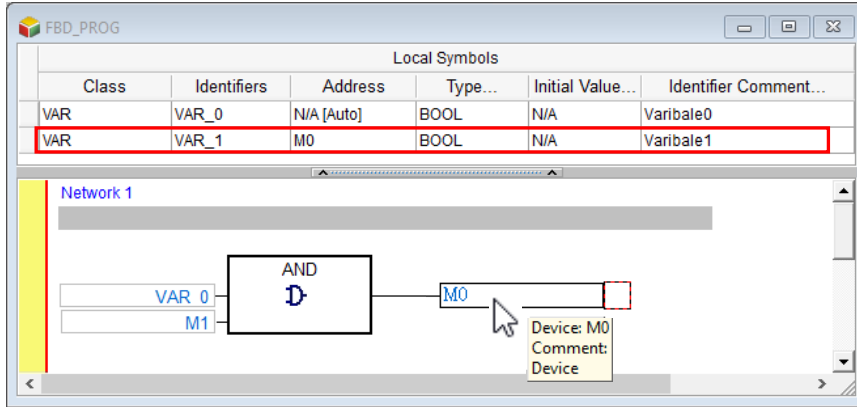
### 11.2.10 Symbol Mode and Address Mode

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.





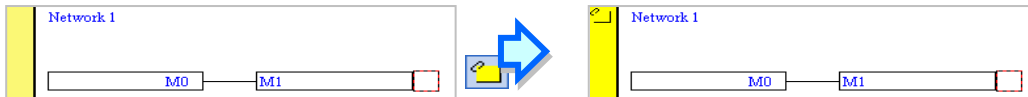
If  on the toolbar is clicked, the devices to which symbols are assigned will be displayed, and the hints which appear will be the hints related to the devices.





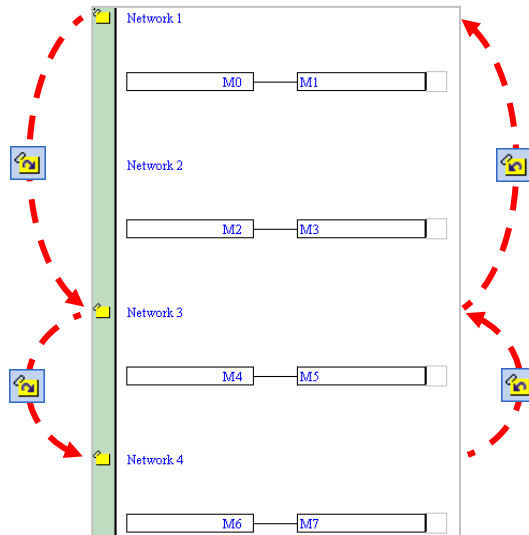
### 11.2.11 Bookmark

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.





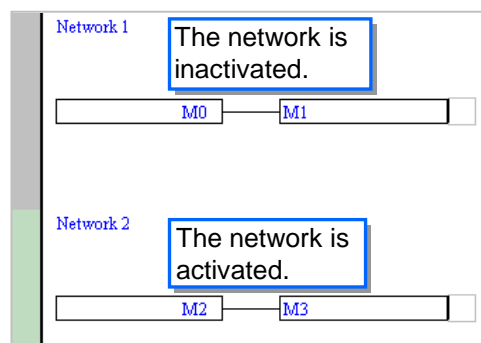
**Additional remark**

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.

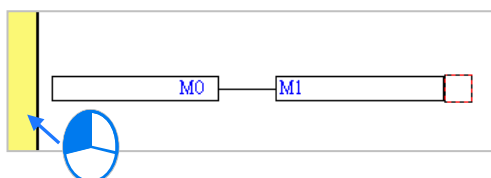
**11.2.12 Activating/Inactivating a Network**


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.

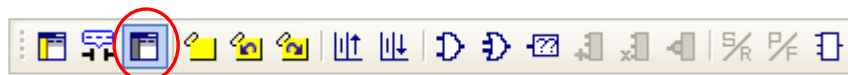
The color at the left side of a network indicates whether the network is activated or inactivated. Users can select colors which will be shown at the left sides of the networks in a ladder diagram in the **Options** window in ISPSOft. Please refer to section 2.3.1 for more information.



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 12 Instruction List

## Table of Contents

<b>12.1</b>	<b>Introduction of Instruction Lists .....</b>	<b>12-2</b>
12.1.1	Structure of an Instruction List .....	12-2
12.1.2	Calling a Function Block.....	12-4
12.1.3	Important Points About Instruction Lists .....	12-5
<b>12.2</b>	<b>Create Instruction List in ISPSoft.....</b>	<b>12-7</b>
12.2.1	Editing Environment.....	12-7
12.2.2	Edit IL instruction .....	12-8
12.2.3	Insert API and Function Blocks .....	12-9
12.2.4	Bookmark.....	12-11
12.2.4.1	Adding Bookmarks .....	12-11
12.2.4.2	Removing Bookmarks .....	12-12
12.2.4.3	Going to a Bookmark.....	12-13

## 12.1 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.

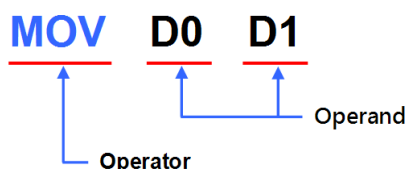
### 12.1.1 Structure of an Instruction List

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	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
OR M1: The logical OR operation is performed on M1.	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
AND M2: The logical AND operation is performed on M2.	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
OUT M3: M3 The result of the operation is sent to M3.	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
(*Note*): It is a comment on the program. When the program is compiled, this line is skipped automatically.	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">No action</div>

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.

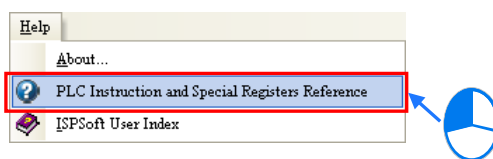


On an instruction list, the words between (\* and \*) are regarded as comments. When a program is compiled, the system automatically skips the words between (\* and \*). Besides, as long as the structure of an instruction list is not destroyed, comments can be put anywhere on the instruction list. However, 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.

The image shows a sequence of actions. On the left, a small instruction list window has 'MOV D0 D1' highlighted in yellow. A blue arrow points from this window to a larger window titled 'PLC Instruction and Special Registers Reference'. A blue arrow labeled **[F1]** points from the instruction list to the larger window, indicating that pressing F1 opens this window.

The 'PLC Instruction and Special Registers Reference' window contains a table with the following data:

API	Instruction code		Operand	Function	
0300	D	MOV	P	S, D	Transferring the data

Device	X	Y	M	S	T	C	H	C	D	L	SM	SR	E	PR	K	16#	\$	F
S	-	-	-	-	-	-	-	-	-	-	-	-	-	TM	-	TM	TM	TM
D	-	-	-	-	-	-	-	-	-	-	-	-	-	TM	-	-	-	-

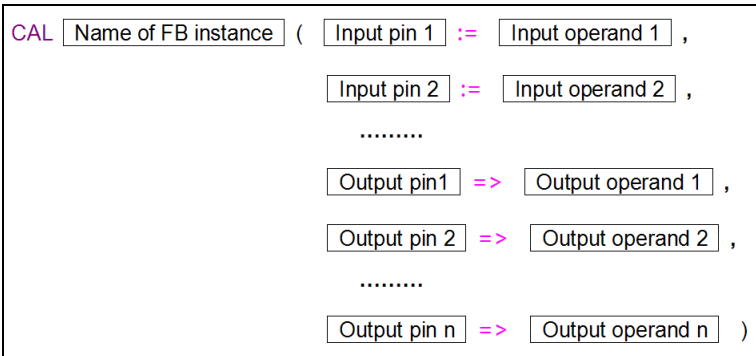
  

Pulse instruction	16-bit instruction (5 steps)	32-bit instruction (5 steps)
AH500	AH500	AH500

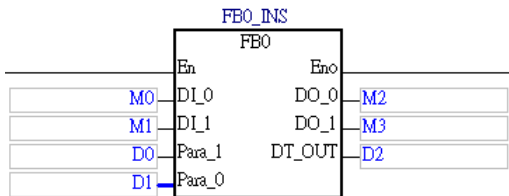
### 12.1.2 Calling a Function Block

The important points about calling a function block on an instruction list are listed below.

12



- 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.



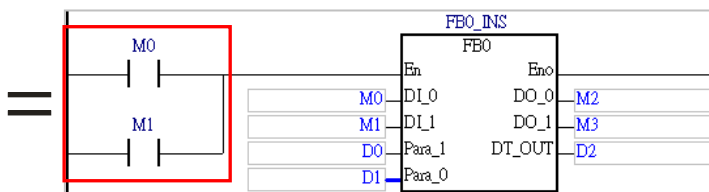
```

0003 CAL FBO_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.

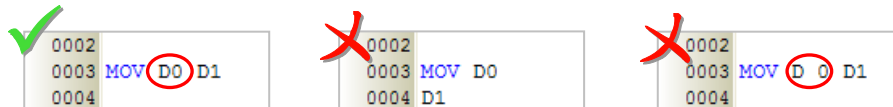
```

0001 LD M0
0002 OR M1
0003
0004 CAL FBO_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
    
```





### 12.1.3 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.)
  - BOOL: Need to use normally-open contact / normally-closed contact
    - AH/AS Series: SM400/ SM401
    - DVP Series: M1000/ M1001

- When using ARRAY, the input format is **Identifier[Index]** and since IL currently does not support index value, therefore, users can use decimal-constant smaller than array, but must begin with 0.
- When using function blocks or APIs, if there is no LD commands or logic calculation area before giving an instruction, an error will occur during compiling.

 <pre> 0001 0002 0003 MOV D0 D1 0004 0005 CAL FBO_INS ( DI_0 := M0 , 0006                 DI_1 := M1 , 0007                 Para_1 := D0 , 0008                 Para_0 := D1 , 0009                 DO_0 =&gt; M2 , 0010                 DO_1 =&gt; M3 , 0011                 DT_OUT =&gt; D2 ) 0012                 </pre>	 <pre> 0001 LD M100 0002 0003 MOV D0 D1 0004 0005 CAL FBO_INS ( DI_0 := M0 , 0006                 DI_1 := M1 , 0007                 Para_1 := D0 , 0008                 Para_0 := D1 , 0009                 DO_0 =&gt; M2 , 0010                 DO_1 =&gt; M3 , 0011                 DT_OUT =&gt; D2 ) 0012                 </pre>
--	--

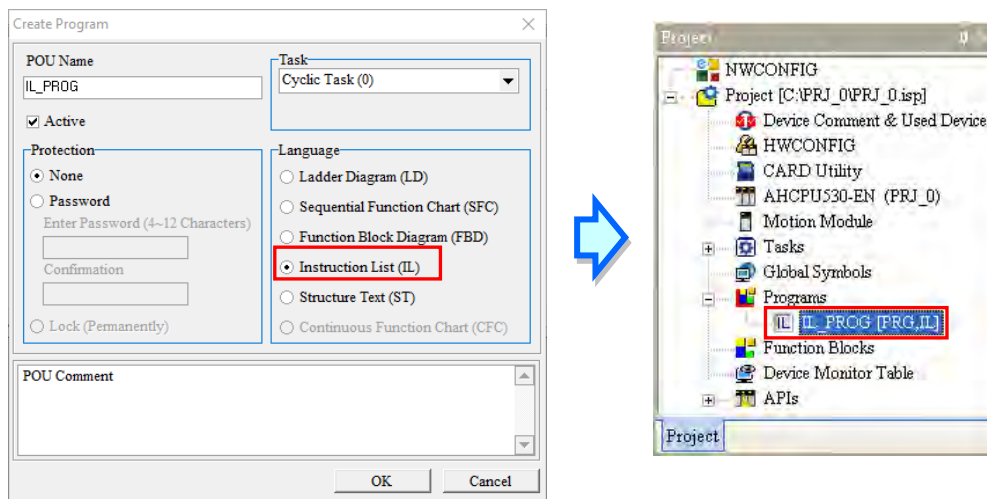
- Although there is no limit on the number of lines, users still have to consider the capacity of the memory in the PLC.
- The sections of an instruction list in ISPSOft can be copied/cut/pasted. Users can copy the text in a file edited with a text editor into an instruction list in ISPSOft.



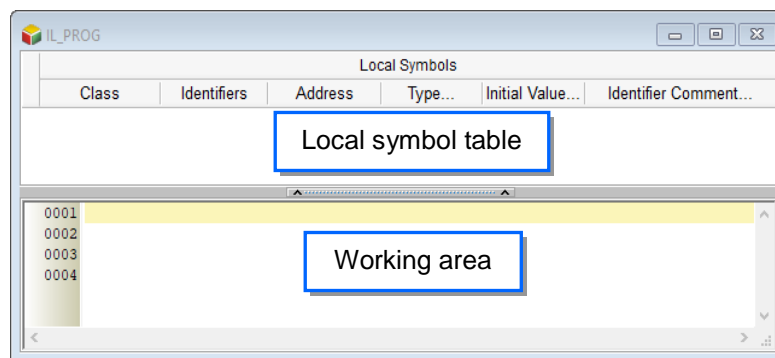
## 12.2 Create Instruction List in ISPSoft





### 12.2.1 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.

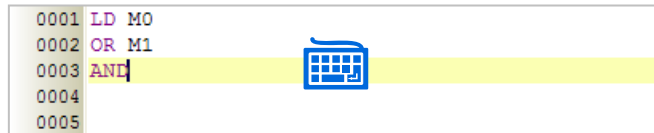


Icon	Keyboard shortcut	Function
	Shift+Ctrl+B	Adding a bookmark to the network selected or deleting the bookmark from the network selected
	Shift+Ctrl+P	Going to the previous bookmarked position
	Shift+Ctrl+N	Going to the next bookmarked position
	Shift+Ctrl+U	Inserting an instruction or a function block

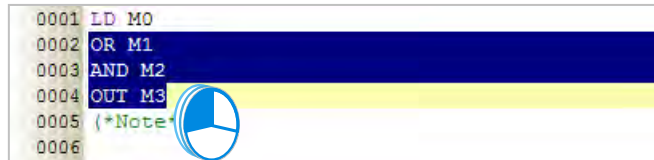
### 12.2.2 Edit IL instruction

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.

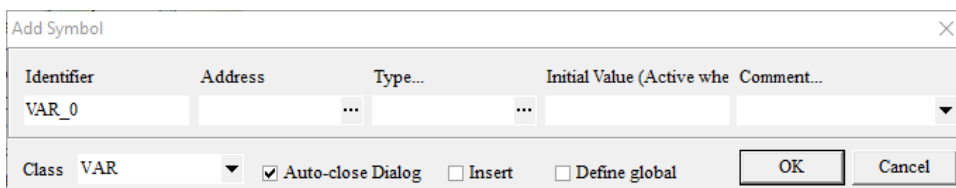
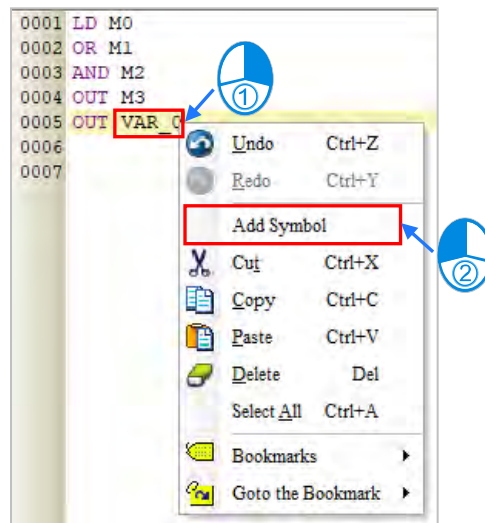
12



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.



### 12.2.3 Insert API 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.

```

0001 LD MO
0002 CAL FBO_INS (
0003     TRIG := MO ,
0004     DT_IN := D0 ,
0005     DT_OUT => D1
0006 )
0007
0008

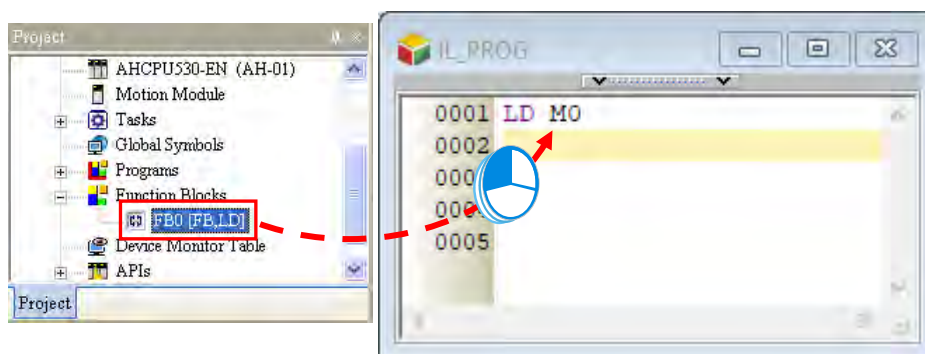
```



12

- **Method 2**

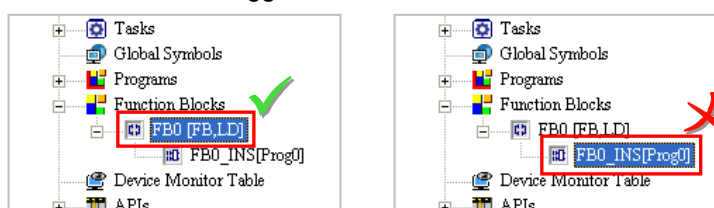
Unfold the **APIs** section or the **Function Blocks** section in the project management area, and find the item which will be inserted. Select the item, and drag it to the position in which it will be inserted.




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.

<pre> 0001 LD MO 0002 CAL ????? ( 0003     TRIG := ?????, 0004     DT_IN := ?????, 0005     DT_OUT =&gt; ????? 0006 ) 0007 </pre>		<pre> 0001 LD MO 0002 CAL FBO_INS ( 0003     TRIG := MO , 0004     DT_IN := D0 , 0005     DT_OUT =&gt; D1 0006 ) 0007 </pre>
---	---	--

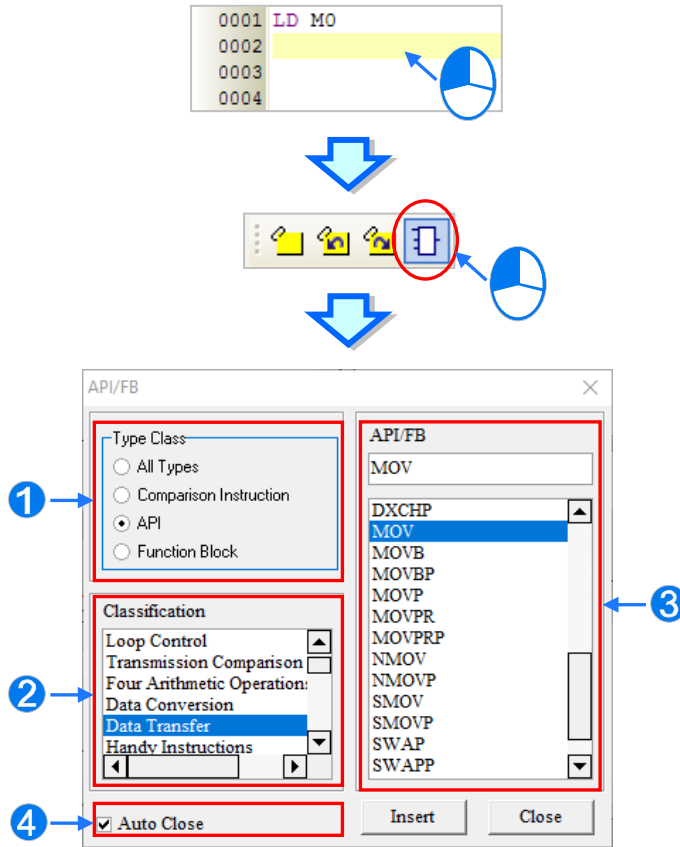
Function block definitions can be dragged, but function block instances can not be dragged.



● **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**.

12



- 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.

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.



## 12.2.4 Bookmark

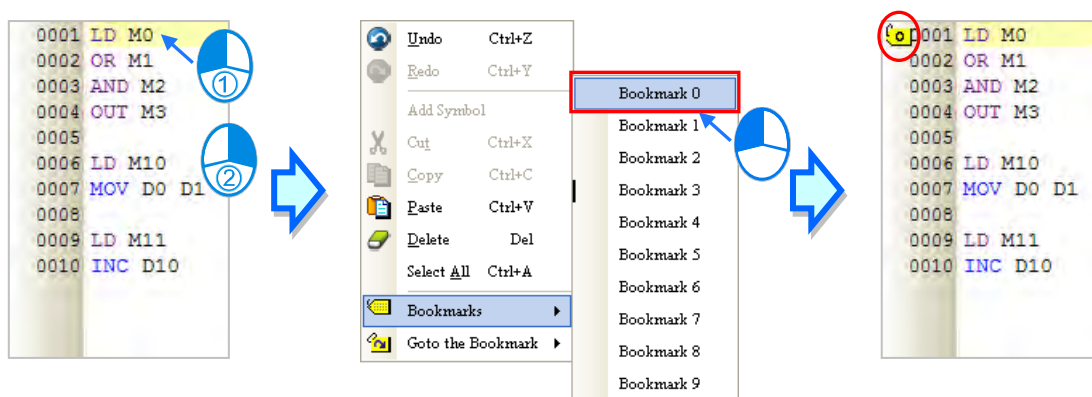
After lines are bookmarked, users can find or go to the lines easily. Users can add ten bookmarks at most to an instruction list.

### 12.2.4.1 Adding Bookmarks


#### ● Method 1

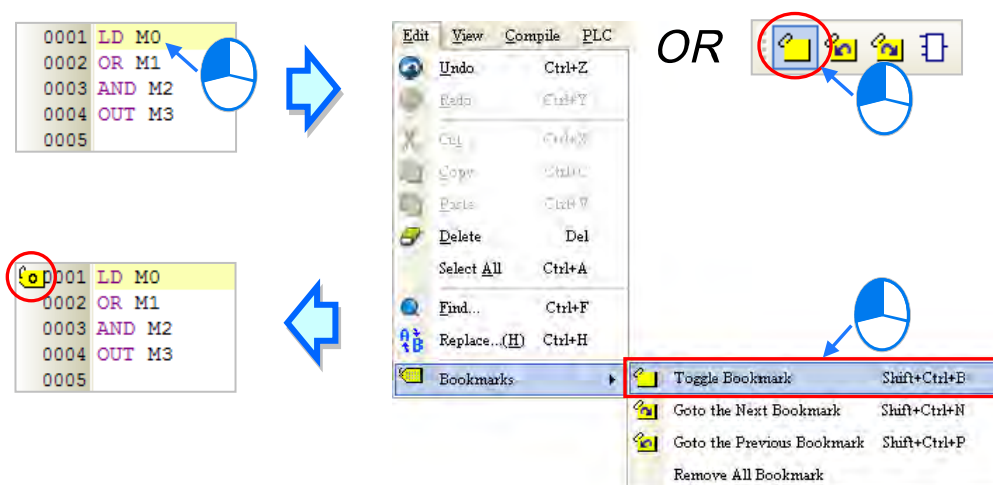
If users want to add a bookmark to a line, they can select the line, click the right mouse button, click **Bookmarks** on the context menu, and select a bookmark number. If the bookmark number selected has been added to another line, the bookmark number will be moved to the line selected presently.

12



#### ● Method 2

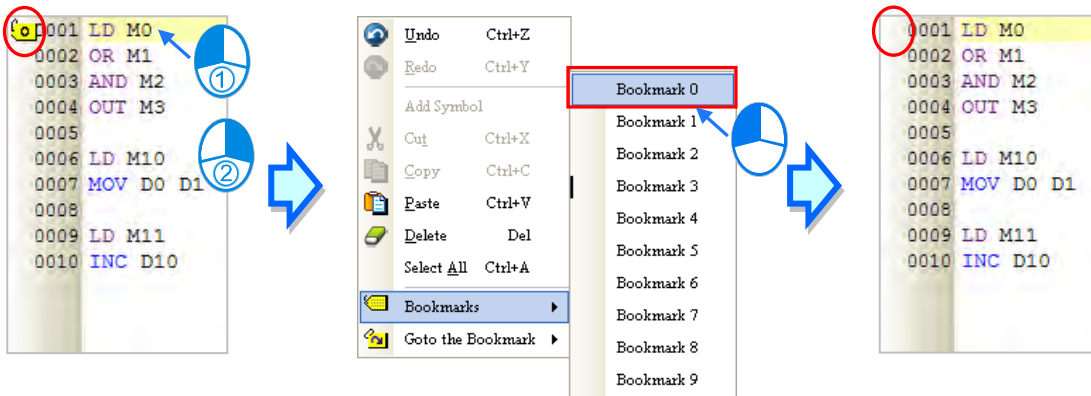
If users want to add a bookmark to a line, they can click the line, and click  on the toolbar. The users can also add a bookmark to a line by clicking the line, clicking the **Edit** menu, pointing to **Bookmarks**, and clicking **Toggle Bookmark**. The bookmarks added in this way are numbered in sequence. The users can not specify bookmark numbers.




### 12.2.4.2 Removing Bookmarks

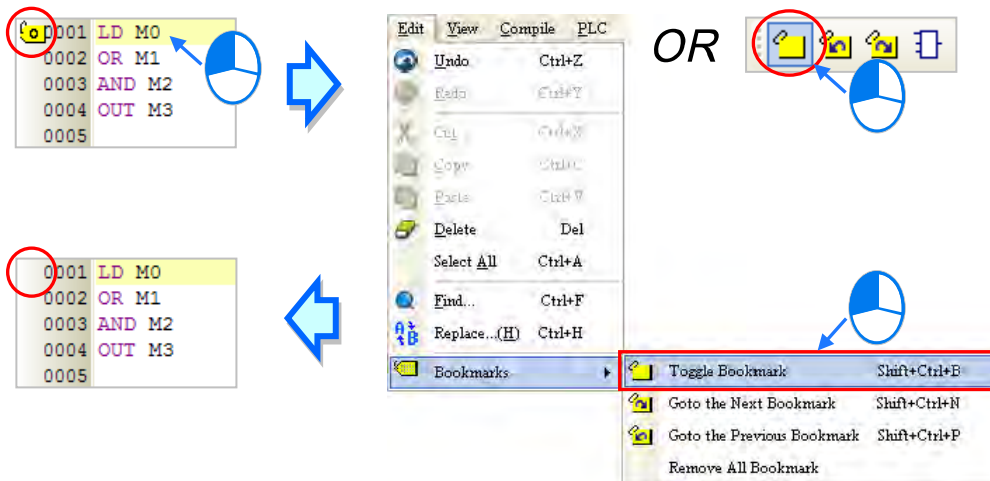
● **Method 1**

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**

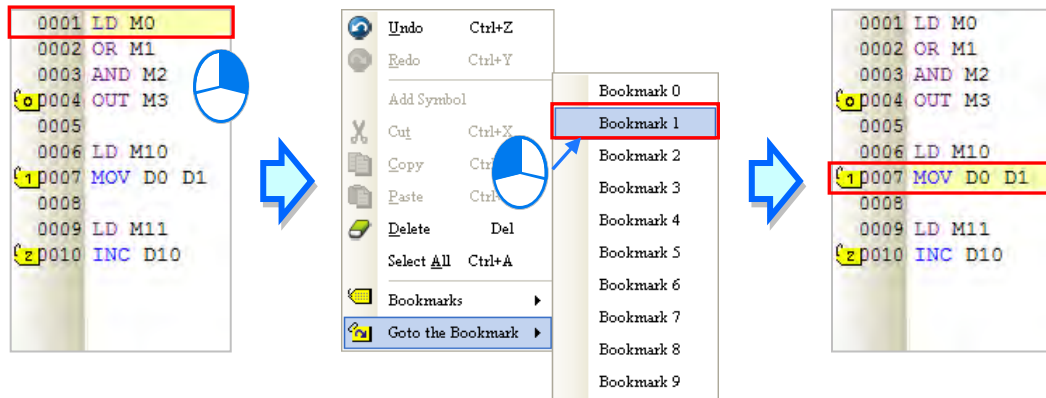
If users want to remove the bookmark from a line, they can click the line, and click  on the toolbar. The users can also remove the bookmark from a line by clicking the line, clicking the **Edit** menu, pointing to **Bookmarks**, and clicking **Toggle Bookmark**. If the users click the **Edit** menu, point to **Bookmarks**, and click **Remove All Bookmarks**, all the bookmarks in the present program editing window will be deleted.





### 12.2.4.3 Going to a Bookmark

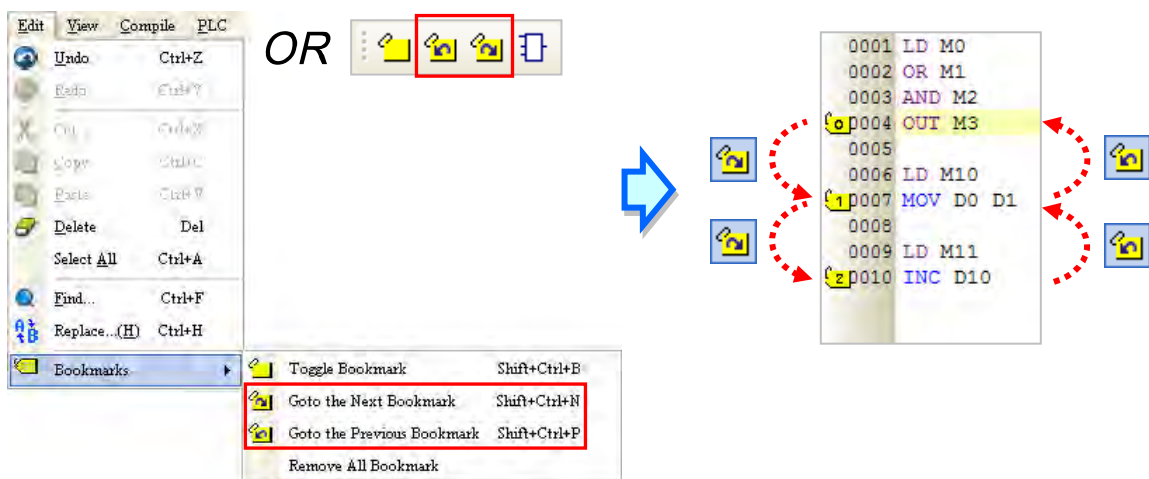
- **Method 1**

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**.



# Chapter 13 Structured Texts

## Table of Contents

<b>13.1</b>	<b>Introducing Structured Texts .....</b>	<b>13-2</b>
13.1.1	Basic Structure of a Structured Text.....	13-2
13.1.2	Statement .....	13-3
13.1.3	Expression .....	13-4
13.1.4	Operand and Operator.....	13-5
13.1.5	Keyword and Comment .....	13-6
13.1.6	Using Array Symbols in ST.....	13-8
13.1.7	Notes on ST Programming .....	13-9
<b>13.2</b>	<b>Structure of a Statement .....</b>	<b>13-11</b>
13.2.1	Assignment Structure—: = .....	13-11
13.2.2	Conditional Structure - IF .....	13-13
13.2.3	Conditional Structure—CASE .....	13-15
13.2.4	Loop Structure - REPEAT .....	13-18
13.2.5	Loop Structure - WHILE.....	13-19
13.2.6	Loop Structure - FOR .....	13-20
13.2.7	Applied Instruction Structure .....	13-22
13.2.8	Function Block Structure.....	13-23
13.2.9	Blank Statement.....	13-25
13.2.10	RETURN Statement.....	13-25
13.2.11	EXIT Statement.....	13-26
<b>13.3</b>	<b>Create a Structured Text in ISPSOft.....</b>	<b>13-27</b>
13.3.1	ST Editing Environment .....	13-27
13.3.2	Edit Structured Text.....	13-28
13.3.3	Insert API and Function Blocks .....	13-29
13.3.4	Bookmark .....	13-30
<b>13.4</b>	<b>Example of a Structured Text .....</b>	<b>13-32</b>
13.4.1	Explanation.....	13-32
13.4.2	Planning Hardware.....	13-32
13.4.3	Planning a Program.....	13-32
13.4.4	Creating a Program.....	13-33



## 13.1 Introducing Structured Texts

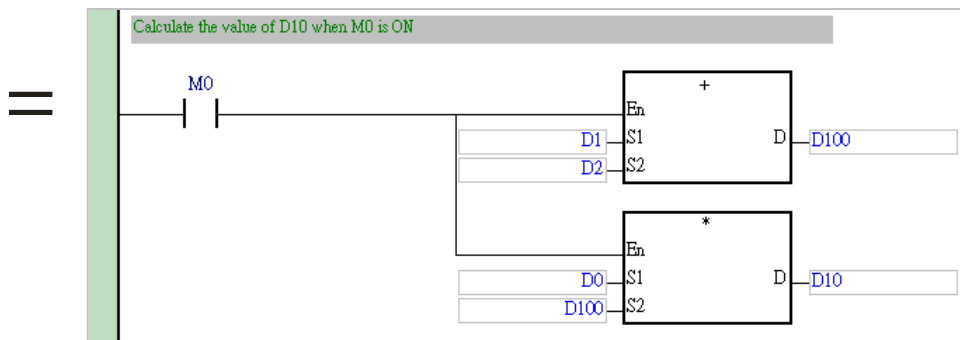
A structured text is a high level language that is syntactically resembles C or Pascal. It is composed of text. Compared with instruction lists, structured texts are more readable. A structured text is very flexible and intuitive for writing control algorithms. People who are trained in programming languages 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.

### 13.1.1 Basic Structure of a Structured Texts

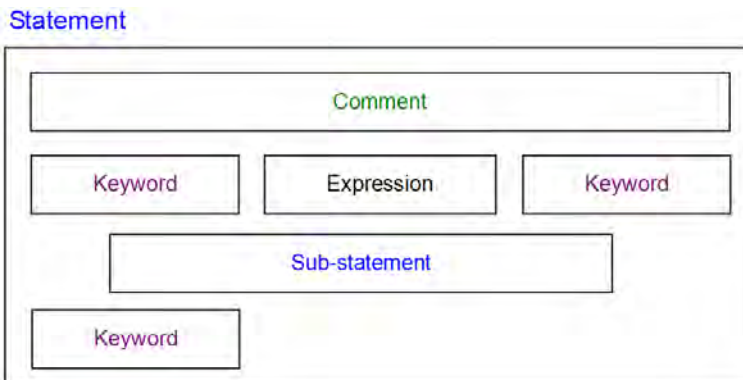
The structured text shown below is a basic structured text. It has the same function as the ladder diagram under it. The structure of the structured text is similar to the syntax of human language. A structured text is composed of statements.

```

0001 (*Calculate the value of D10 when M0 is ON*)
0002 IF M0 THEN
0003     D10 := D0*(D1+D2) ;
0004 END_IF ;
0005
    
```



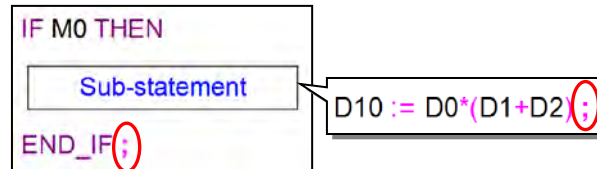
The structure of the structured text above is shown below.



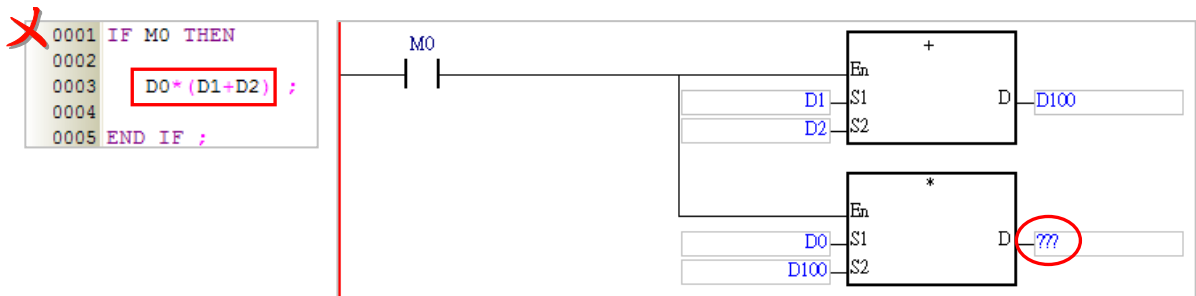
\* In a structured text, “:=” means “to assign”, and “=” means “to be equal to”.

### 13.1.2 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.

```

0001
0002 IF M0 THEN
0003     M2 := TRUE ;
0004 ELSIF M1 THEN
0005     M2 := FALSE ;
0006 END_IF ;
0007
0008 M20 := M10 ;
0009
0010 M21 := M11 ;
0011
0012 IF M2 THEN
0013     ROR(D0,2) ;
0014 END_IF ;
0015

```

\* In a structured text, “:=” means “to assign”, and “=” means “to be equal to”.

### 13.1.3 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 FALSE.

- **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 is 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**.

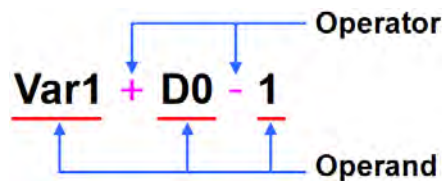
Please refer to the following example. The program is mainly composed of two IF statements. When the first IF statement is executed, the system evaluates the expression **M0**. If **M0** is ON, the value of the arithmetic expression **D21+D22** will be calculated, and the result of the operation will 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;
```

### 13.1.4 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	$D0 \text{ MOD } 3$	2	
+, -	Addition,	Any value	Any value	$D0 + 3$	8	

Symbol	Function	Data format		Example of an expression		Precedence
		Operand	Result of an operation (Value of an expression)	Expression	Value	
	subtraction					
< , >, <=, >=	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.

### 13.1.5 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 cannot 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, you can use multiple comments (using words between *( \* and \* )*, *(/\* and \*/)* and single line comment (using *( / )*). When a program is compiled, the system automatically skips the words between *( \* and \* )*, *(/\* and \*/)* and *( / )* in the program. 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 examples below. The comments in the left example are legal. The comment in the right example divides the keyword “END\_IF” into two parts by using */\*and\*/*, and therefore an error will occur after the program is compiled.

```

0001 (*Note*)
0002
0003 IF (*Note*) M100 = TRUE THEN
0004
0005     D10 (*Note*) := D10 * (D1+D2);
0006
0007 END_IF (*Note*); (*Note*)
0008
0009 (*Note*)
0010
0011 //Note
0012
0013 IF /*Note*/ M100 = TRUE THEN
0014
0015     D10 /*Note*/ := D10 * (D1+D2);
0016
0017 END_IF/*Note*/; /*Note*/
0018
0019 /*Note*/
    
```

```

0001
0002 IF M100 = TRUE THEN
0003
0004     D10 := D10 * (D1+D2);
0005
0006 END (*Note*)_IF;
0007
0008
0009 IF M100 = TRUE THEN
0010
0011     D10 := //D10 * (D1+D2);
0012
0013 END_/*Note*/_IF;
0014
    
```

13

ISPSOft provides a quick way to change the programs to Block Comment/Line comment. Click and drag the ST block that you'd like to edit and then right click the selected area to open the quick menu.

The screenshot illustrates the process of converting a code block into a block comment. It shows a code editor with the following code:

```

0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 P_temp := _POS ;
0005 IF ( (NOT _NEG) AND N_temp ) THEN
0006     N_CNT := N_CNT + 1 ;
0007 END_IF ;
0008 N_temp := _NEG ;
    
```

A blue arrow points to the selected code block. A context menu is open over the selected area, showing the following options:

- Undo (Ctrl+Z)
- Redo (Ctrl+Y)
- Add Symbol
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Del)
- Select All (Ctrl+A)
- Block Comment/Uncomment (Ctrl+Q)
- Line Comment/Uncomment (Ctrl+Alt+Q)
- Bookmarks
- Goto the Bookmark

After selecting the Block Comment/Line Comment, the comments will be added. Click again if you'd like to remove the comments.

```

0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 (*P_temp := POS ;
0005 IF ( (NOT NEG) AND N_temp ) THEN
0006     N_CNT := N_CNT + 1 ;
0007 END_IF ;*)
0008 N_temp := _NEG ;

```

Block Comment

```

0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 (*P_temp := POS ;*)
0005 (*IF ( (NOT NEG) AND N_temp ) THEN*)
0006 (*     N_CNT := N_CNT + 1 ;     *)
0007 (*END_IF ;*)
0008 N_temp := _NEG ;

```

Line Comment

### 13.1.6 Using Array Symbols in ST

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 AH/AS 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 cannot 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 but the structure from the Data Unit Type can support up to three-dimensional arrays. This type of variables can be written as the image shown below. DO value is written in the pt1 [1,1,1] of the vardut. Refer to chapter 8 for more information.

```

0001 IF M0 THEN
0002
0003 vardut.pt1[1,1,1] := DO ;
0004
0005 END_IF ;
0006

```

- 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.

### 13.1.7 Notes on ST Programming

- If the structure of the keywords or the statements in a structured text is not destroyed, starting new lines and leaving blank spaces is allowed.
- In principle, full width characters and half width characters in ISPSoft are regarded as different characters. Please avoid using full width 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 (BOOL): 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, both 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 two operands can take part in the operation as well. If the data lengths of the operands in an expression are different, the operands cannot 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, the system regards it as a signed integer during the operation.
- Although there is no limit to 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 edited text into a structured text in ISPSoft.



- The following instructions are not supported in a structured text:

<b>AH/AS series CPU modules</b>
All the IL instructions are not supported. For example, LD, LDI, OUT, SET, RST, LD&, LDP are not supported.
All the pulse type API instructions are not supported. For example, ROLP, RORP, BCDP, TRDP 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. In addition, instructions including PLS, PLF, MOV, DMOV, DFMOV, \$MOV, MC, MCR, CJ, JMP, GOEND are not supported.
Also, instructions such as CNT, DCNT used in counter values and TMR instruction used by counters in subroutine program are not supported.

- The following instructions are supported in a structured text but with limitation:

<b>AS series CPU module</b>
All the high speed output instructions (API 27xx) for example, DPLSY and DDRVI are supported. Please refer to DDRVI instruction for application.
The use of DCNT instructions for high speed counters is supported. But, statements need to be executed so that values can be updated.
Communication instructions are supported but only after flag instruction setup is completed. The instructions used in the program procedures can be scanned for execution.

## 13.2 Structure of a Statement

### 13.2.1 Assignment Structure—:=

#### ● Format

Device or Symbol	:=	Expression
------------------	----	------------

#### ● Description

The value represented by the expression at the right side of := is assigned (transmitted) to the device or the symbol at the left side of :=.

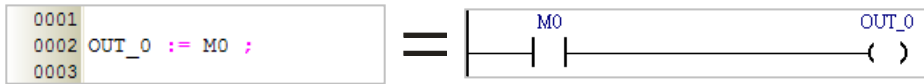
#### ● Rules and limits

- 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 :=.
- For the assignment / transmission, 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 and the other operand is a symbol whose data type is INT.
  - (f) Both operands are symbols whose data types are REAL/STRING. The data lengths of the operands must be the same.
  - (g) One operand is a symbol whose data type is WORD, and the other operand is a symbol whose data type is DWORD. The transmission direction can only be from WORD to DWORD but not the other way around or the data might be lost.
  - (h) One operand is a symbol whose data type is INT, and the other operand is a symbol whose data type is DINT. The transmission direction can only be from INT to DINT but not the other way around or the data might be lost.

13

● Example

Example 1: The value of M0 is assigned to the symbol OUT\_0.



Example 2: M0 is set to ON.



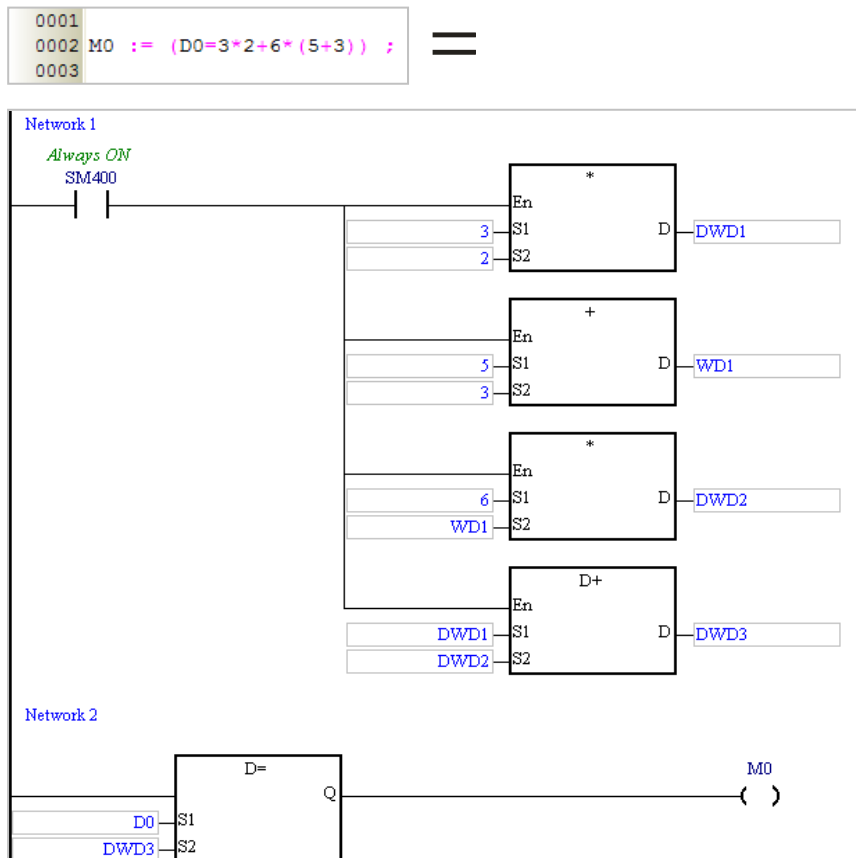
Example 3: The value in D2 is added to the value in D1. The sum is stored in D0.



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.



Example 6: Examples of incorrect use of the assignment structure are as follows.

Statement	Description
<code>D0 := M0 ;</code>	The data type of one operand must not be different from the data type of the other operand.
<code>D0 := V_REAL ;</code>	The value of a symbol whose data type is REAL can not be assigned to a D device.
<code>V_LREAL := V_REAL ;</code>	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.
<code>V_DWORD := V_TIMER ;</code>	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.
<code>V_WORD := HC0 ;</code>	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.

13

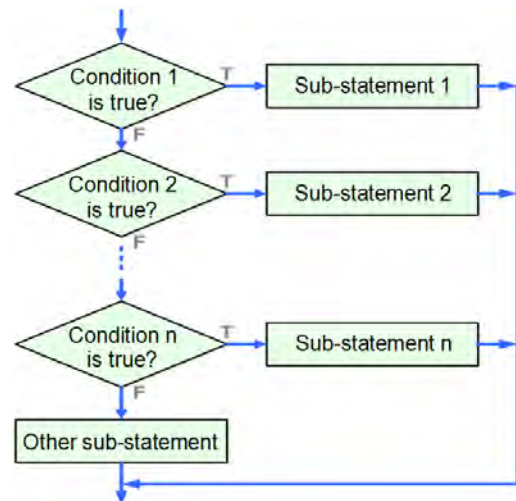
### 13.2.2 Conditional Structure - IF

#### • 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.

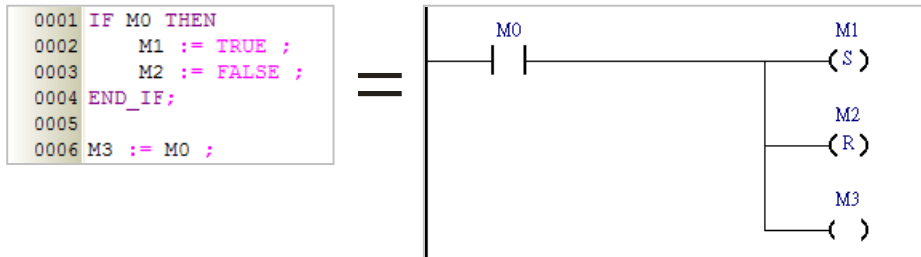
● Rules and limits

- 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.

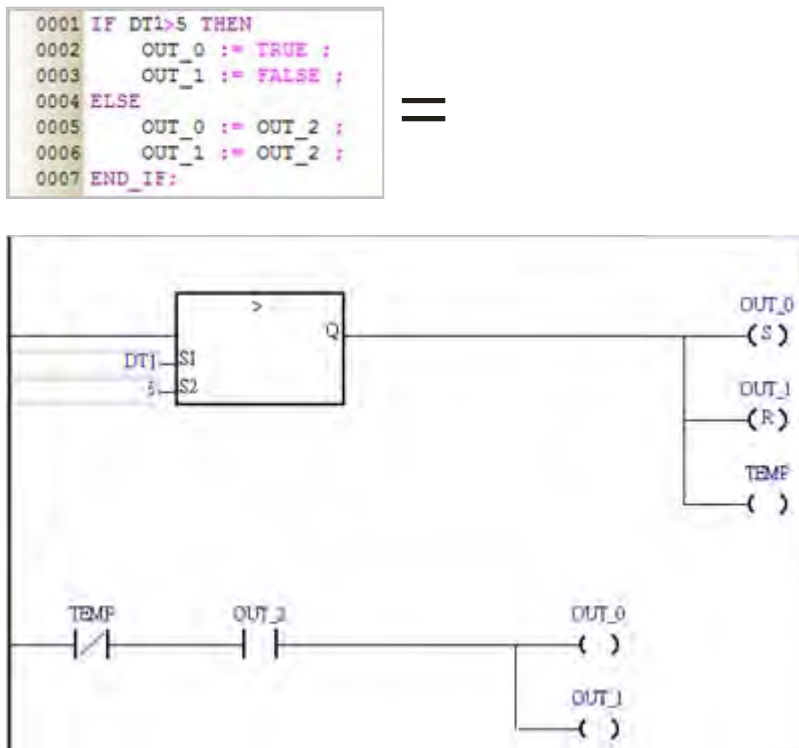
13

● 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.



Example 2: If the value of DT1 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.

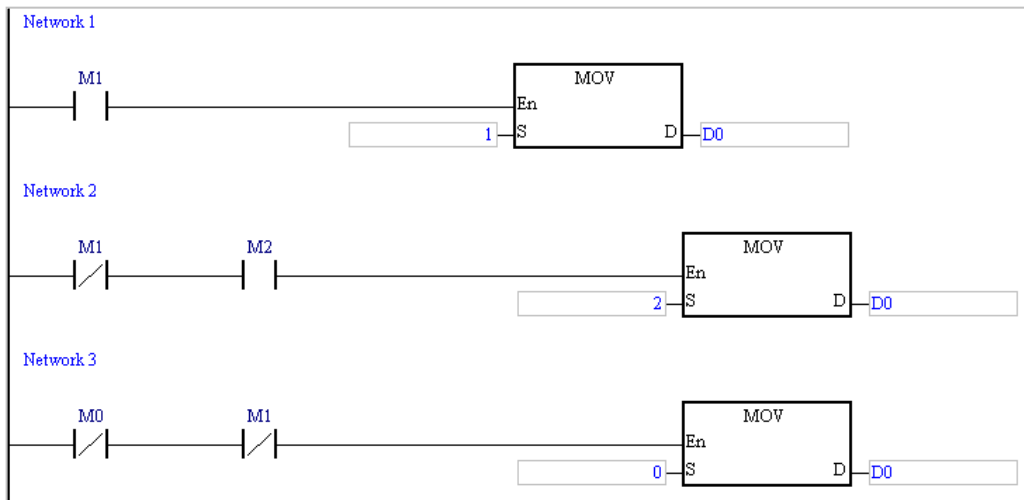


Example 3: If M1 is ON, the value in D0 will be 1. If M1 is not ON, the PLC will evaluate M2. If M2 is ON, the value in D0 is 2. If M2 is not ON, the value in D0 will be 0.

```

0001 IF M1 THEN
0002     D0 := 1 ;
0003 ELSIF M2 THEN
0004     D0 := 2 ;
0005 ELSE
0006     D0 := 0 ;
0007 END_IF;
    
```

=



13

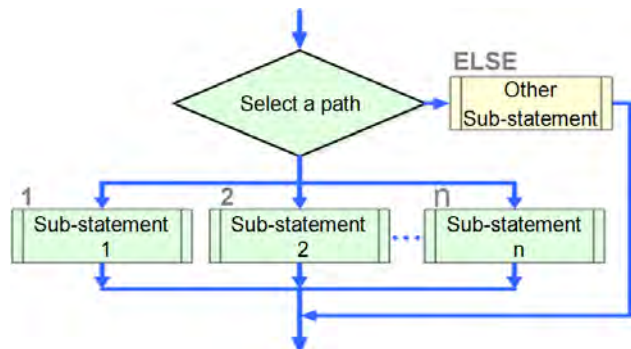
### 13.2.3

### 13.2.3 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**

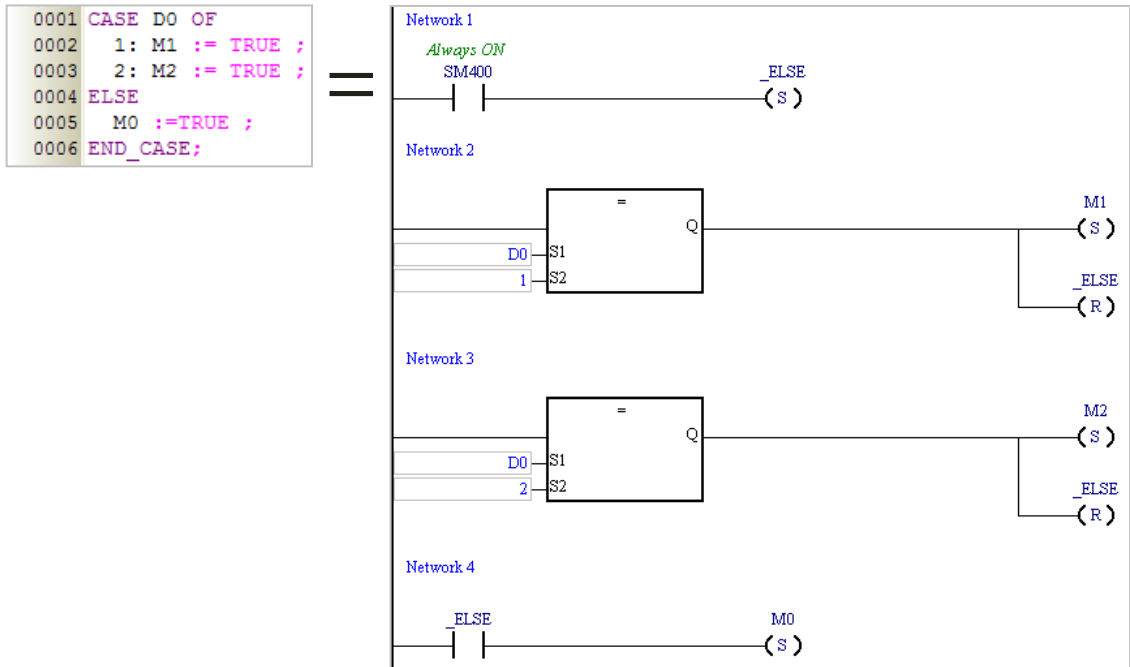
The PLC judges whether the value of **Integer expression** is on the list of condition values. If the value of **Integer expression** is the same as a condition value on the list, the sub-statement following the condition value will be executed. If the value of **Integer expression** is not on the list 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.
- 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.
  - 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.

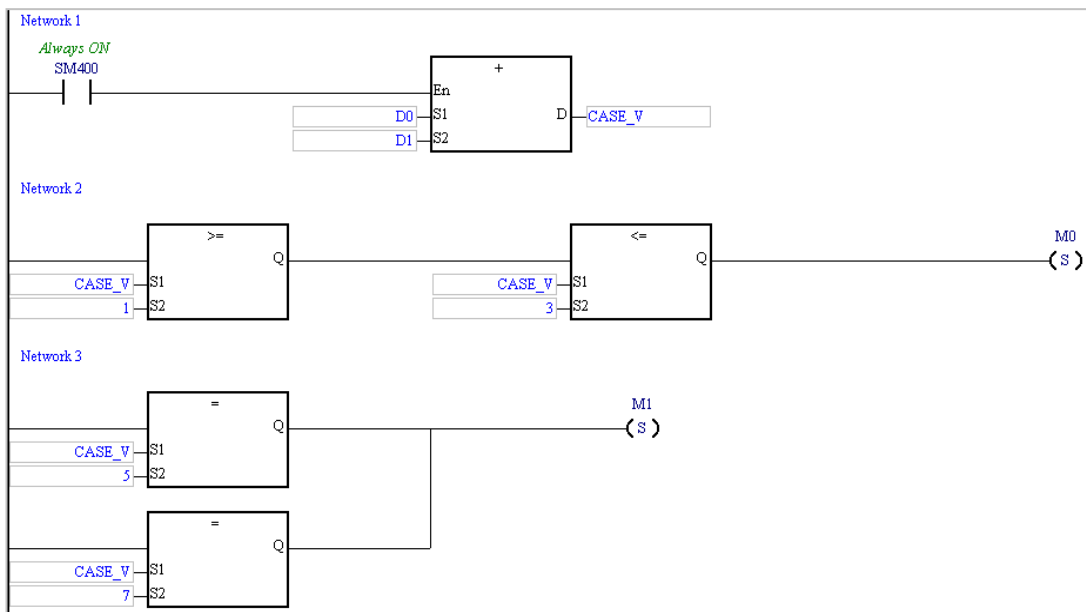


13

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;
    
```





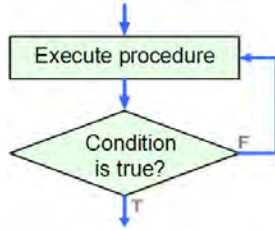
### 13.2.4 Loop Structure - REPEAT

#### ● Format

```

REPEAT
  Sub-statement ;
UNTIL Bool expression
END_REPEAT ;

```



#### ● 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 cannot 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 DT1 is 0. In the loop structure, the value of DT1 increases progressively by 2. The loop will not be terminated until the value of DT1 is larger than 100.

```

0001 DT1 := 0 ;
0002 REPEAT
0003   DT1 := DT1 + 2 ;
0004 UNTIL DT1 > 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  $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$ .

```

0001 D10 := 5 ;
0002
0003 D0 := 1 ;
0004 REPEAT
0005   D0 := D0 * D10 ;
0006   D10 := D10 - 1 ;
0007 UNTIL D10 = 0
0008 END_REPEAT ;

```

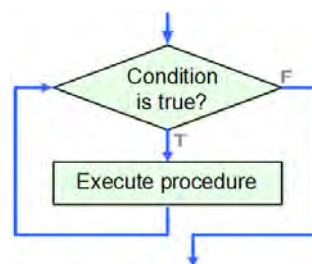
### 13.2.5 Loop Structure - WHILE

#### ● Format

```

WHILE Bool Expression DO
  Sub-statement ;
END_WHILE ;

```



13

#### ● 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** cannot be a device or a symbol which cannot 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 DT1 is 0. In the loop structure, the value of DT1 increases progressively by 2. The loop will not be terminated until the value of DT1 is larger than 100.

```
0001 DT1 := 0 ;
0002
0003 WHILE DT1 <= 100 DO
0004     DT1 := DT1 + 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.

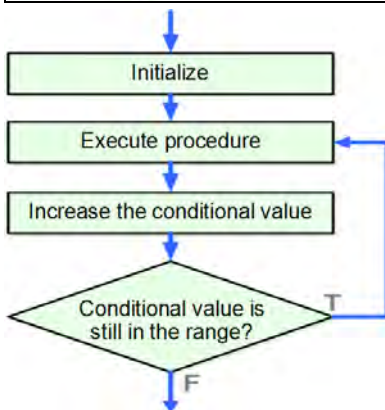
```
0001 M0 := TRUE ;
0002 D0 := 0 ;
0003 D1 := 0 ;
0004
0005 REPEAT
0006     D0 := D0 + 1 ;
0007 UNTIL M0
0008 END_REPEAT ;
0009
0010 WHILE (NOT M0) DO
0011     D1 := D1 + 1 ;
0012 END_WHILE ;
```

13

### 13.2.6 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.

13

### ● 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

```

### 13.2.7 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.

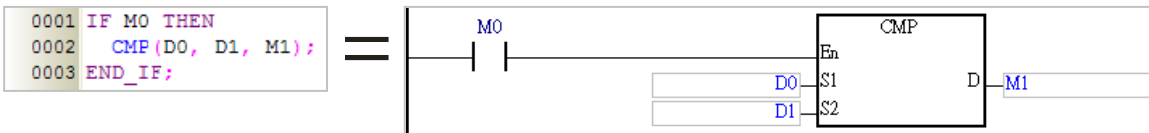
```
0001 CMP (
0002   D0,
0003   D10,
0004   M0
0005 ) ;
```

```
0001 CMP
0002 ( D0, D10, M0 ) ;
0003
0004
0005
```

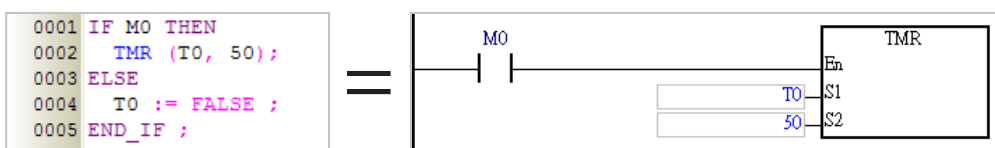
```
0001 CM
0002 P ( D0, D10, M0 ) ;
0003
0004
0005
```

● **Example**

Example 1: The instruction CMP is used with an IF statement.

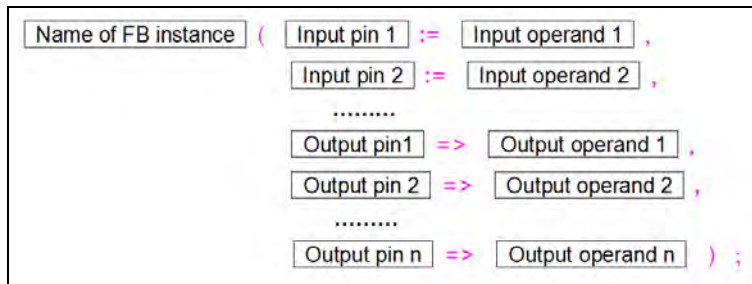


Example 2: The state of the En pin does not need to be specified in an applied instruction. If the instruction TMR is used, a statement about resetting the timer is required. Otherwise, the value of the counter will not become zero. For example, line 0004 in the figure below is a statement about resetting the timer.



## 13.2.8 Function Block Structure

### ● Format

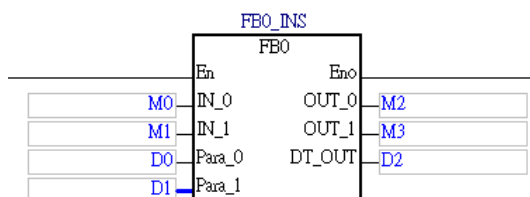


### ● Description

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.



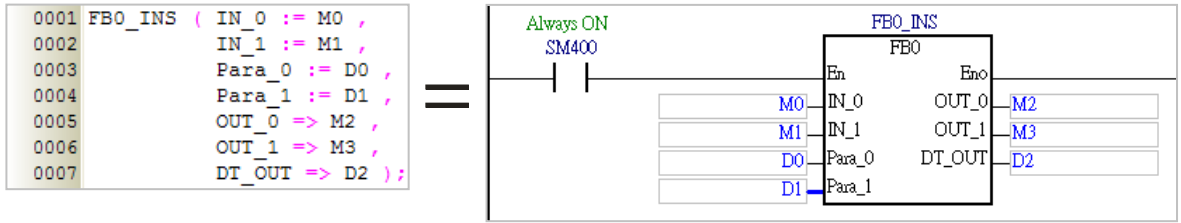
```

0001 FBO_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 ) ;

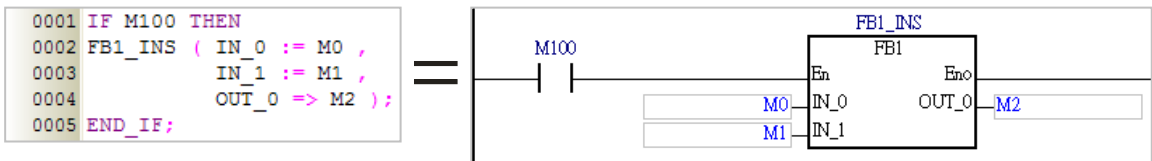
```

● Example

Example 1: When a statement which calls a function block is executed, the function block is executed.



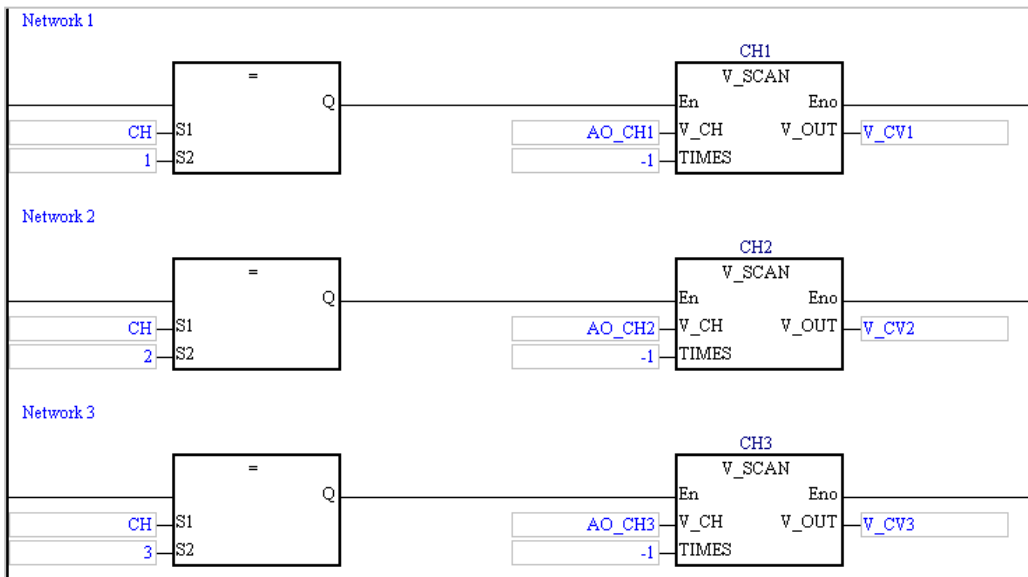
Example 2: A function block is used with an IF statement.



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;
    
```



### 13.2.9 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;
```

13

### 13.2.10 RETURN Statement

- **Format**

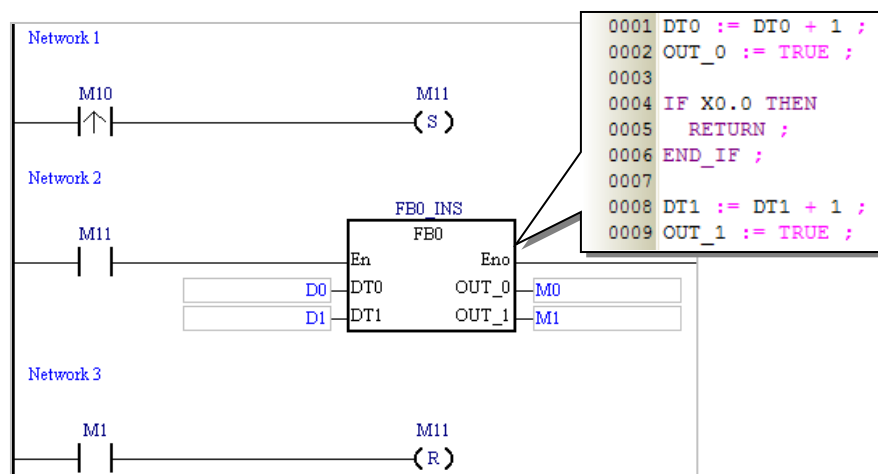
```
RETURN ;
```

- **Description**

A RETURN 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.





### 13.2.11 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 DT1 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 DT1 is 110. After the loop body is executed for the 100<sup>th</sup> time, the value of DT1 will become 10. In other words, the final value of DT1 will be 10. The value of DT1 will not decrease to 0.

```
0001 DT1 := 110 ;
0002 D0 := 0 ;
0003
0004 REPEAT
0005
0006     DT1 := DT1 - 1 ;
0007     D0 := D0 + 1 ;
0008
0009     IF D0=100 THEN
0010         EXIT ;
0011     END_IF ;
0012
0013 UNTIL DT1 = 0
0014 END_REPEAT ;
```

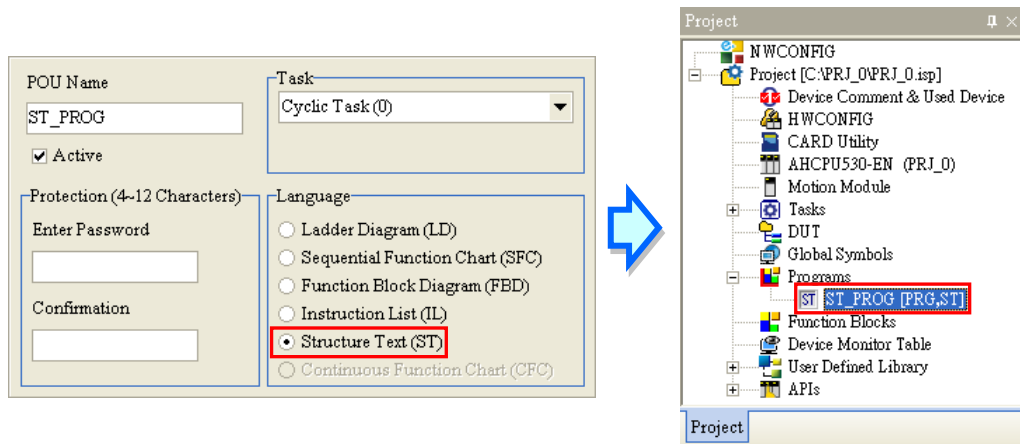
Example 2: The loop below is a FOR loop. Originally, the arithmetic hierarchy is DT1 value (initial value is 10) and the result of the operation is stored in D0. The IF structure in the loop is used with an EXIT statement. The loop will not be terminated until the value in D0 is larger than 100. When the value in D10 increases to 5, the value in D0 is  $1*2*3*4*5=120$ , and the loop is terminated.

```
0001 DT1 := 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 ;
```

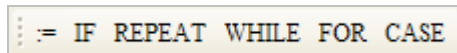
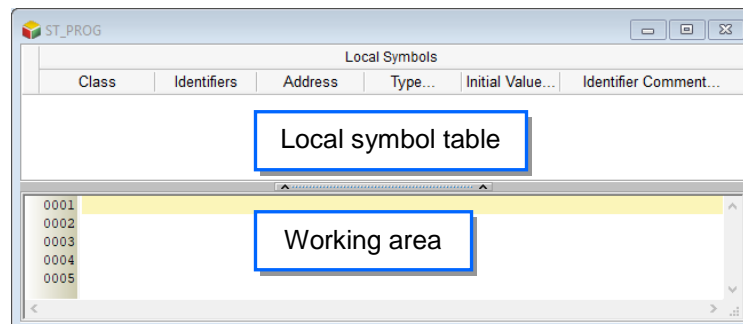
## 13.3 Create a Structured Text in ISPSOft

### 13.3.1 ST 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
	Inserting an example of an if statement
	Inserting an example of a repeat loop
	Inserting an example of a while loop

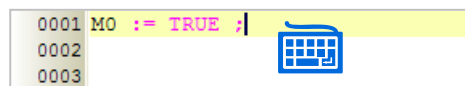
Icon	Function
	Inserting an example of a for loop
	Inserting an example of a case statement

### 13.3.2 Edit Structured Text

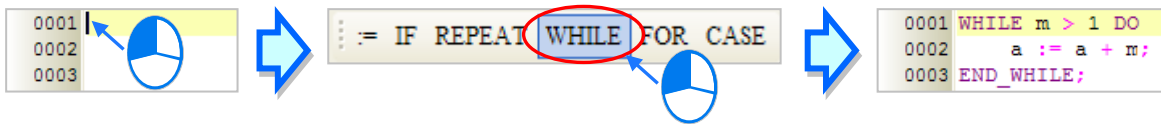
The way in which a structured text file is edited is similar to the way in which a text file is edited with a general text editor. If users want to create a structured text, they can type or modify text in the working area. If the users want to start a new line of text at a specific point, they can press Enter on the keyboard.

13

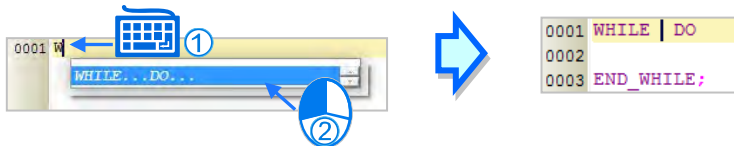
```
0001 MO := TRUE ;
0002
0003
```



Besides, the users can click a line which will be edited, insert an example of a structure, and modify the structure.

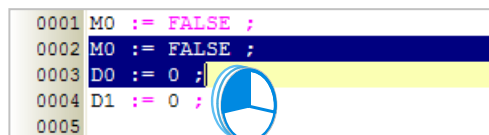


Or you can also type the first letter of the structured text and then the suggested structured text appears. Double-click it and the cursor moves to where the text is and then you can start editing.

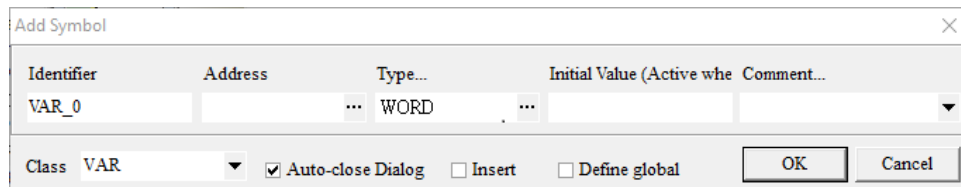
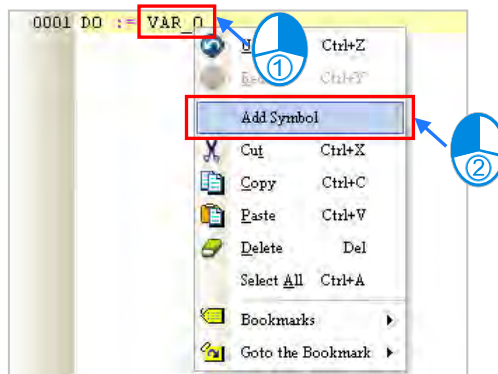


If users want to select any amount of text, they can click where they want to begin the selection, hold down the left mouse button, and drag the pointer over the text that they want to select. The users can also click at the start of the selection, scroll to the end of the selection, and hold down **Shift** on the keyboard while they click where they want the selection to end.

```
0001 MO := FALSE ;
0002 MO := FALSE ;
0003 DO := 0 ;
0004 D1 := 0 ;
0005
```



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.



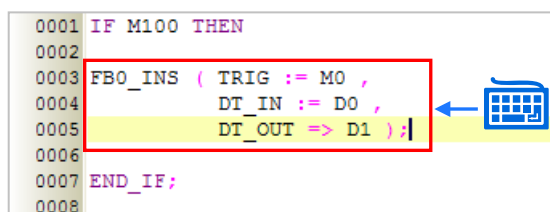
While typing in the symbol, the system will autofill and show a list with variable symbols for users to choose from. Users can use “↑” and “↓” on the keyboard to select or simply click on the variable symbol you'd like to use.

### 13.3.3 Insert API 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.

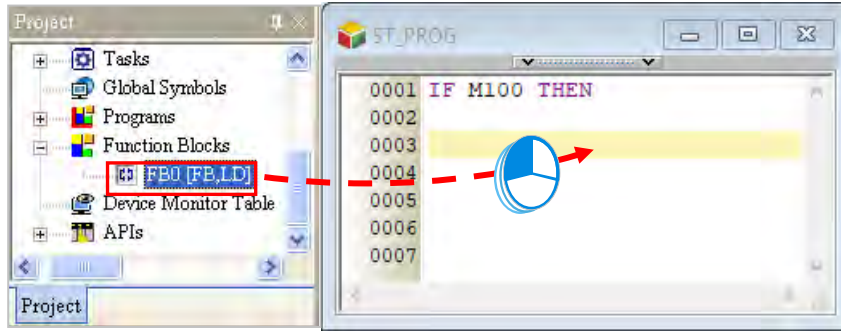
#### ● Method 1

An applied instruction or a function block is inserted according to the format of the applied instruction or the format of the function block.

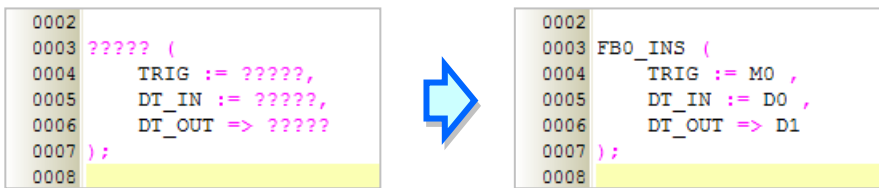


#### ● Method 2

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.



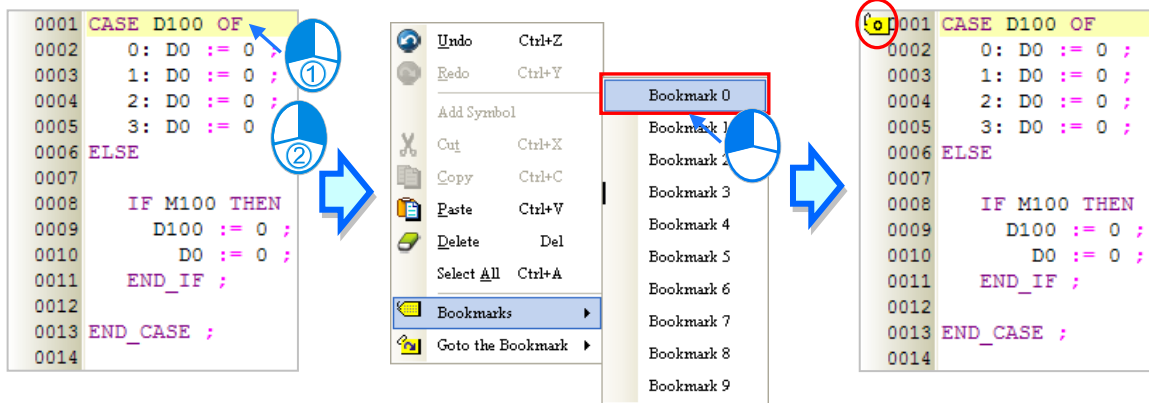
### 13.3.4 Bookmark

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.

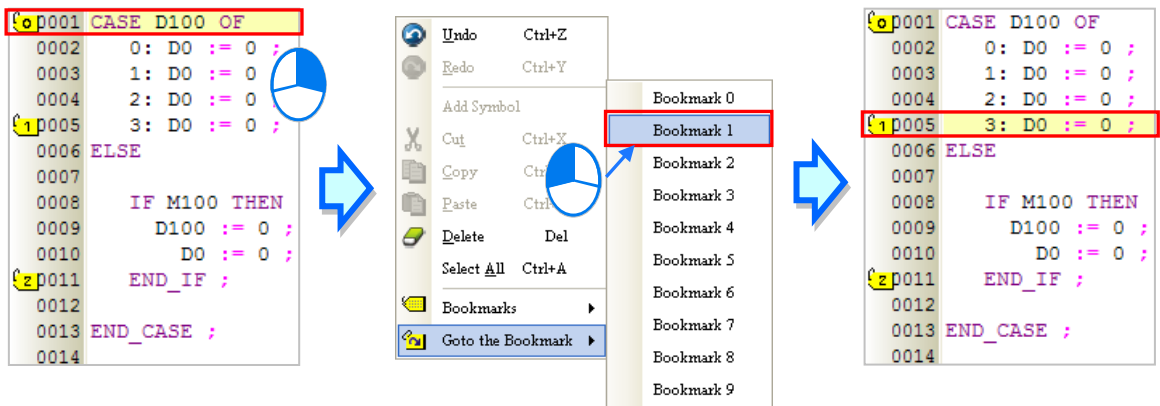
If users click a bookmark number which has been added to another line, the bookmark number will be moved to the line selected presently.



● **Going to a bookmark**

Right-click the working area, point to **Goto the Bookmark** on the context menu, and click a bookmark number.

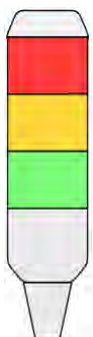
13



## 13.4 Example of a Structured Text

### 13.4.1 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	↑
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

### 13.4.2 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.

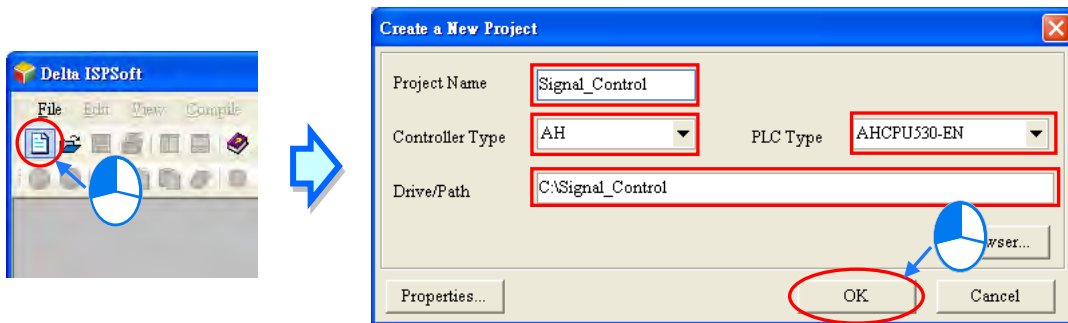
Type	ID	Description
Digital output	Y0.0	Red light
Digital output	Y0.1	Yellow light
Digital output	Y0.2	Green light
Digital output	Y0.3	Buzzer

### 13.4.3 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.

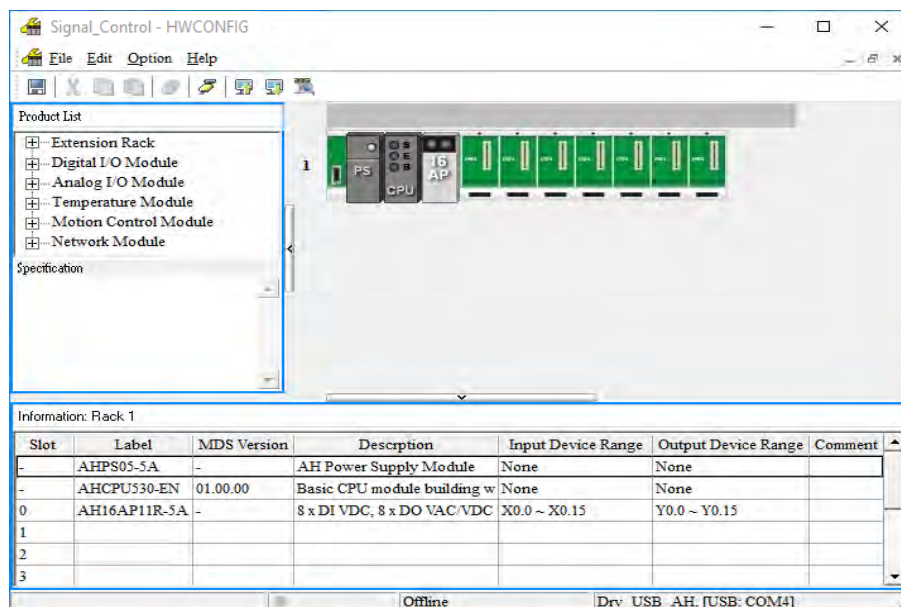
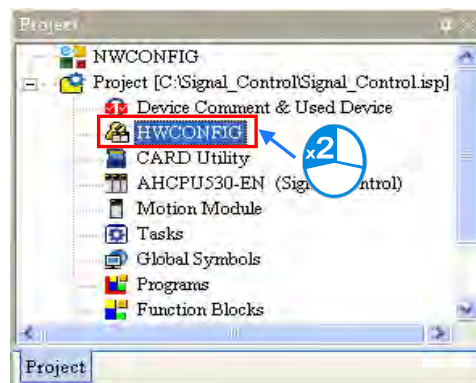
### 13.4.4 Creating a Program

Start ISPSOft, and then create a new project.



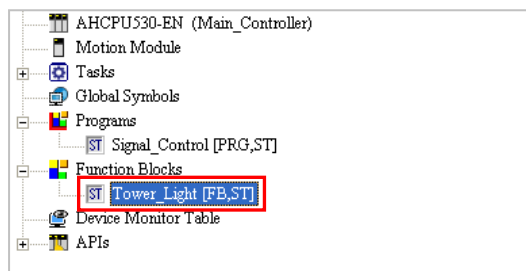
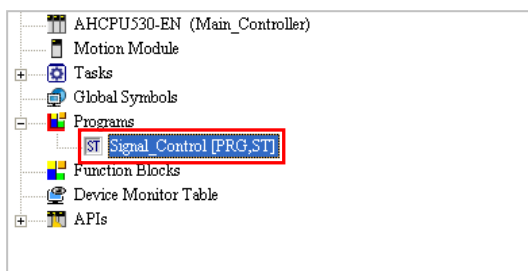
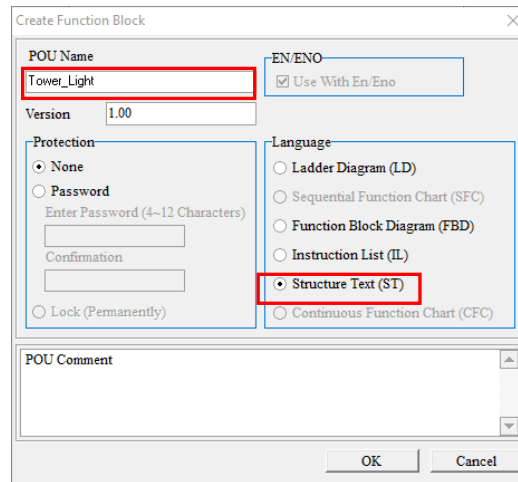
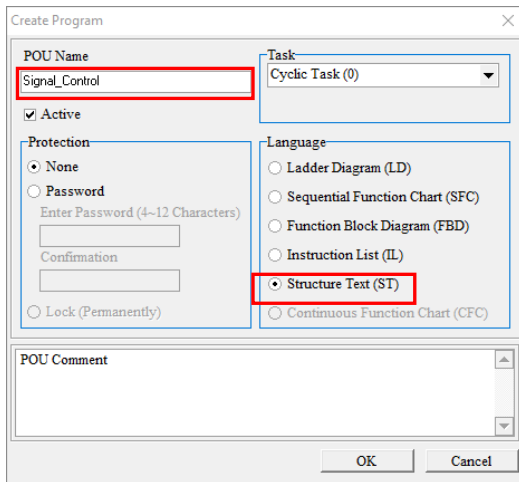
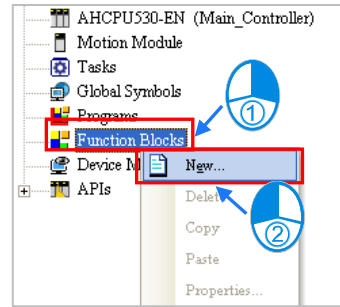
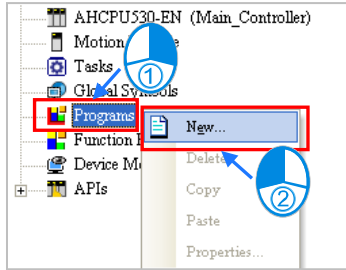
The hardware configuration in this example is the same as the hardware configuration in chapter 4. Please refer to chapter 4, and complete the hardware configuration.

13





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.



13

Create the flags and the digital outputs in the global symbol table.

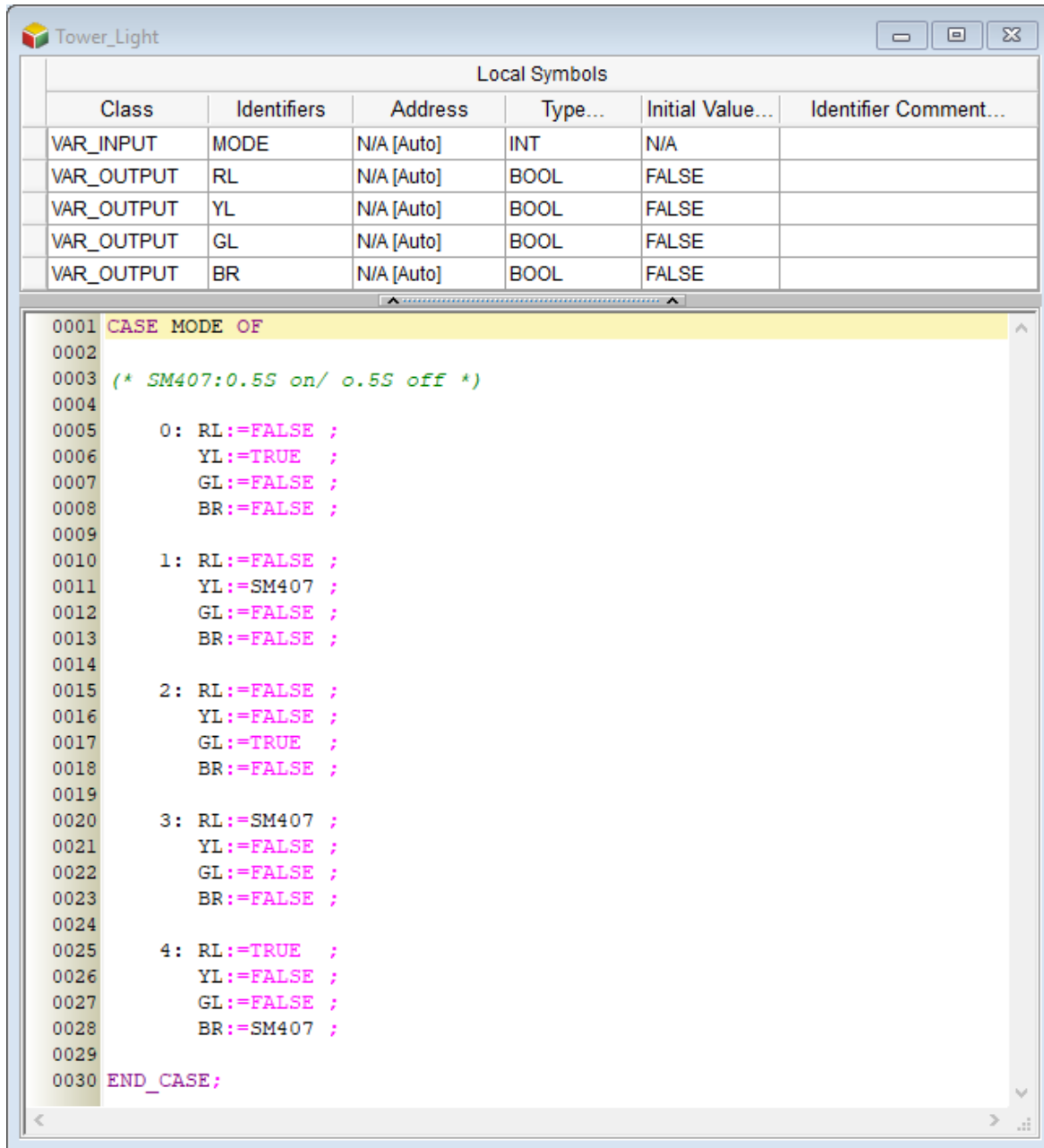
Global Symbols						
	Class	Identifiers	Address	Type...	Initial Valu...	Identifier Comment...
	VAR	ERROR	N/A [Auto]	BOOL	FALSE	
	VAR	RUNNING	N/A [Auto]	BOOL	FALSE	
	VAR	COMPLETE	N/A [Auto]	BOOL	FALSE	
	VAR	BUZZER_OFF	N/A [Auto]	BOOL	FALSE	
	VAR	RED	Y0.0	BOOL	FALSE	
	VAR	YELLOW	Y0.1	BOOL	FALSE	
	VAR	GREEN	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.

13



The screenshot shows the Tower\_Light software interface. At the top, there is a window titled "Tower\_Light" with standard window controls. Below the title bar is a table titled "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	

Below the table is a text editor showing a ladder logic program. The program starts with a CASE statement and includes several assignments for the local symbols RL, YL, GL, and BR.

```

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.

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		
VAR	Tower_Light0	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 (MODE := SC ,
0021         RL => RED ,
0022         YL => YELLOW ,
0023         GL => GREEN ,
0024         BR => BUZZER );
0025
  
```

13

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**

13



# Chapter 14 Sequential Function Charts

## Table of Contents

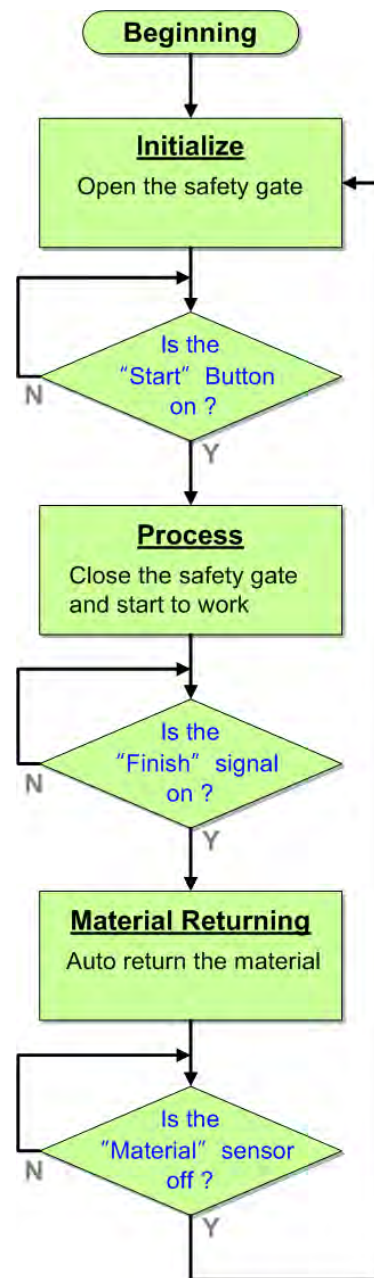
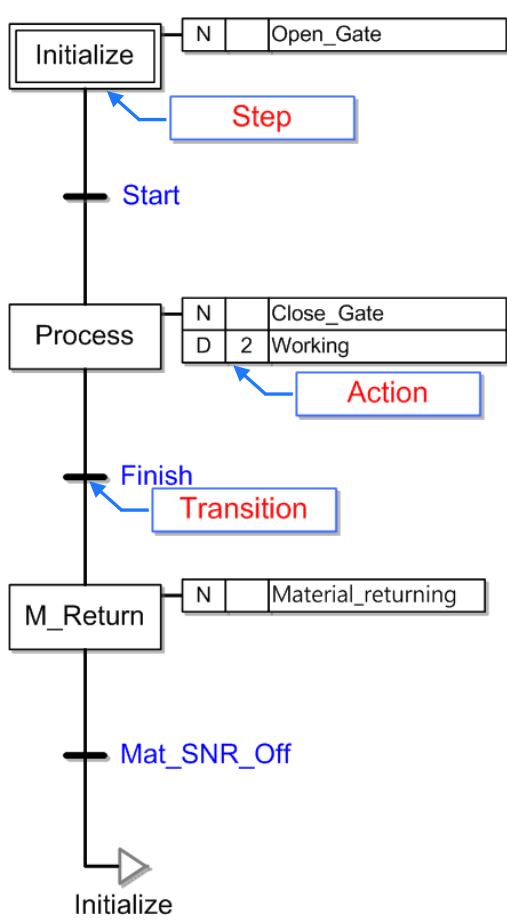
<b>14.1</b>	<b>Knowing Sequential Function Charts .....</b>	<b>14-2</b>
14.1.1	Structure of a Sequential Function Chart .....	14-2
14.1.2	Principle of a Sequential Function Chart.....	14-3
<b>14.2</b>	<b>Sequential Function Chart in ISPSOft .....</b>	<b>14-4</b>
14.2.1	Steps and Actions.....	14-4
14.2.2	Transitions.....	14-7
14.2.3	Simultaneous Divergence and Divergence of Sequence Selection... ..	14-10
14.2.4	Simultaneous Convergence & Select Convergence .....	14-11
14.2.5	Jump .....	14-12
14.2.6	Qualifier of an Action.....	14-15
14.2.6.1	Qualifier Types.....	14-15
14.2.6.2	Important Points About Qualifying an Action.....	14-20
14.2.7	Initial Step.....	14-21
14.2.8	Internal Property .....	14-22
14.2.8.1	Step Property .....	14-22
14.2.8.2	Action and Transition Property .....	14-26
<b>14.3</b>	<b>Create SFC in ISPSOft .....</b>	<b>14-32</b>
14.3.1	Editing Environment.....	14-32
14.3.2	Creating and Managing Actions and Transitions .....	14-34
14.3.3	Adding a Step .....	14-36
14.3.4	Connecting a Transition in Parallel .....	14-37
14.3.5	Connecting a Step in Parallel.....	14-38
14.3.6	Step Structure - Simultaneous Divergence & Select Convergence ..	14-40
14.3.7	Step Structure – Select Divergence & Simultaneous Convergence..	14-41
14.3.8	Inserting a Jump Point .....	14-42
14.3.9	Assign Steps and the Transitions .....	14-43
14.3.10	Specifying an Initial Step.....	14-44
14.3.11	Assigning Actions / Fold the Action Table .....	14-44
<b>14.4</b>	<b>Sequential Function Chart Examples .....</b>	<b>14-48</b>
14.4.1	Example Descriptions.....	14-48
14.4.2	Planning Hardware.....	14-50
14.4.3	Planning a Program.....	14-50
14.4.4	Creating a Program.....	14-51

## 14.1 Knowing Sequential Function Charts

### 14.1.1 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.

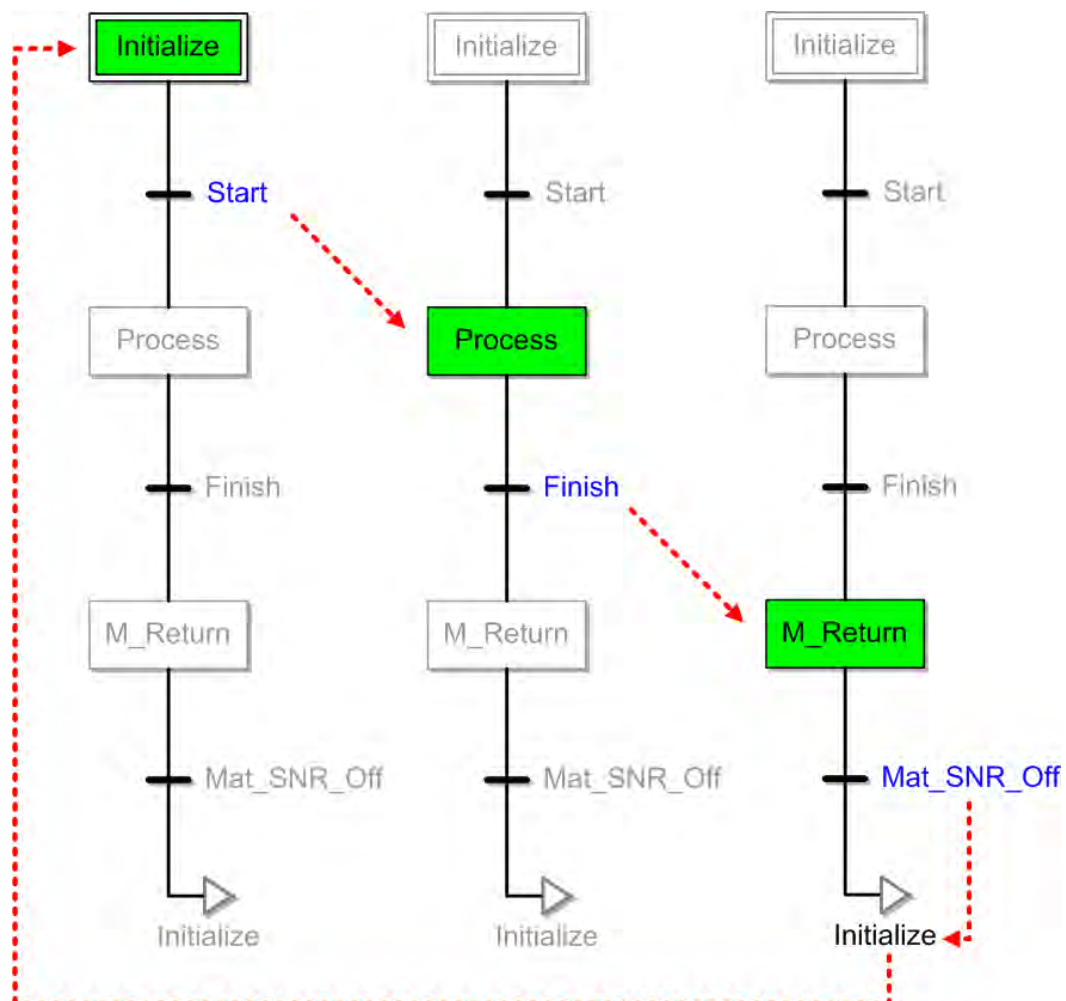


### 14.1.2 Principle of a Sequential Function Chart

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.



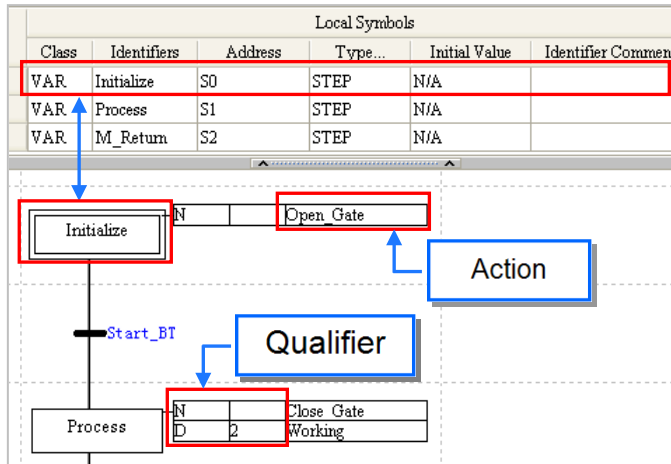
## 14.2 Sequential Function Chart in ISPSOft

If users use sequential function chart (SFC) for programming, the content of this program format may vary according to the PLC series. This chapter contains introduction of the SFC structure.

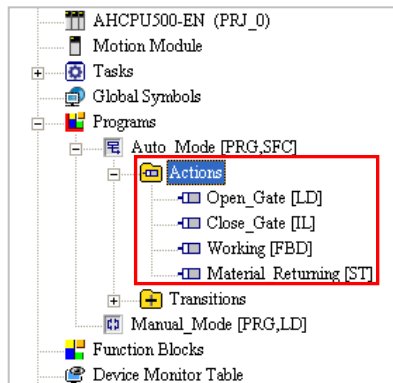
### 14.2.1 Steps and Actions

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 in AH5x0 and DVP series whose data type is STEP occupies an S device in the PLC. Please refer to operation manuals for more information about the S devices.

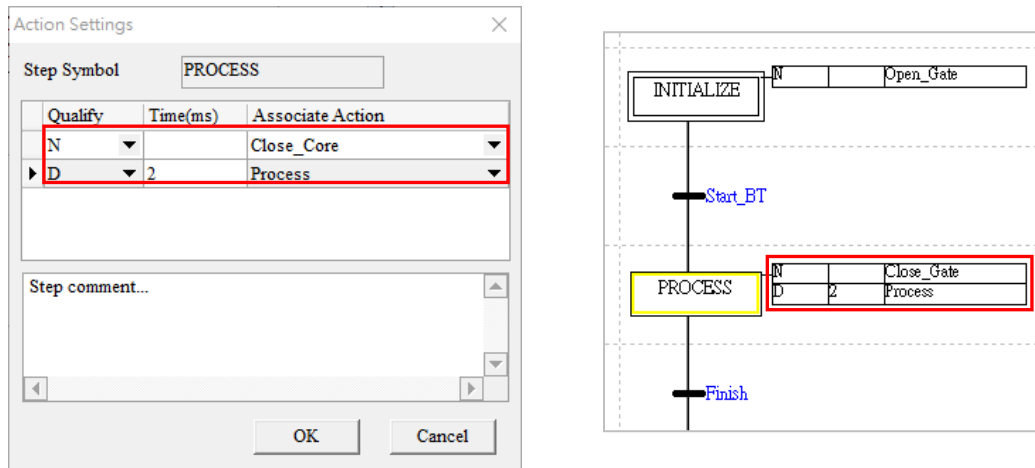
After a step is activated, the actions associated with the step will be executed according to the qualifiers. In ISPSOft, the actions can be modularized, and can be associated with different steps. A step can be associated with more than one action. Besides, every action has a qualifier that determines 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.



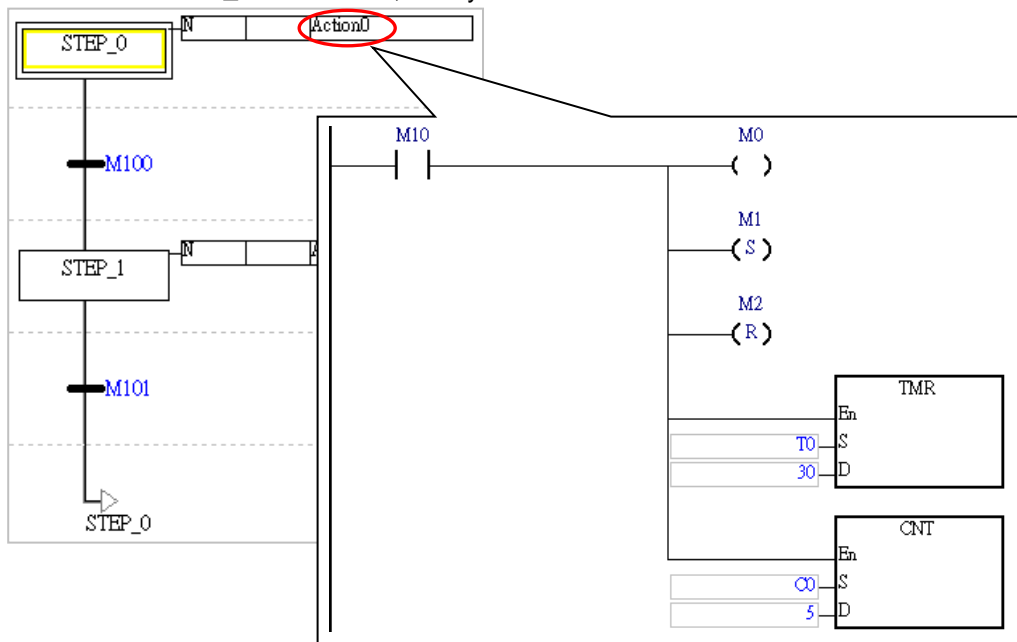
If users want to edit a sequential function chart, they have to create actions, assign the actions to the steps, and assign qualifiers to the actions. The **Action Settings** window and the corresponding action list are shown below.



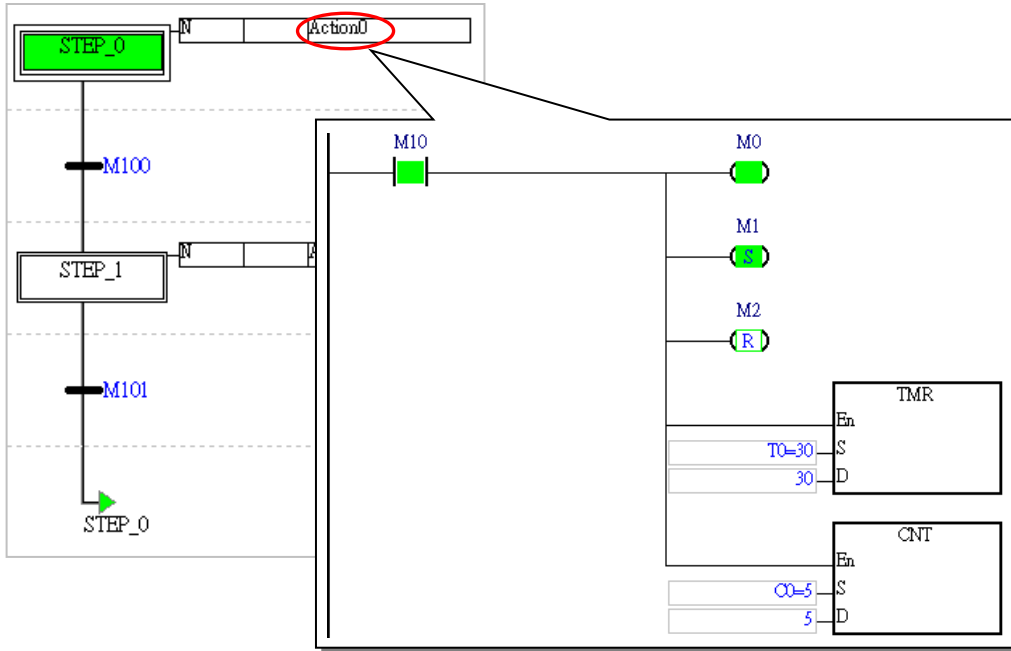
Final scans are important procedures during the execution of a sequential function chart. After the execution of an action stops, the system will automatically execute a procedure to disable all the outputs in the action. If a final scan is executed, the coils driven by the instruction OUT will be OFF, 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.

14

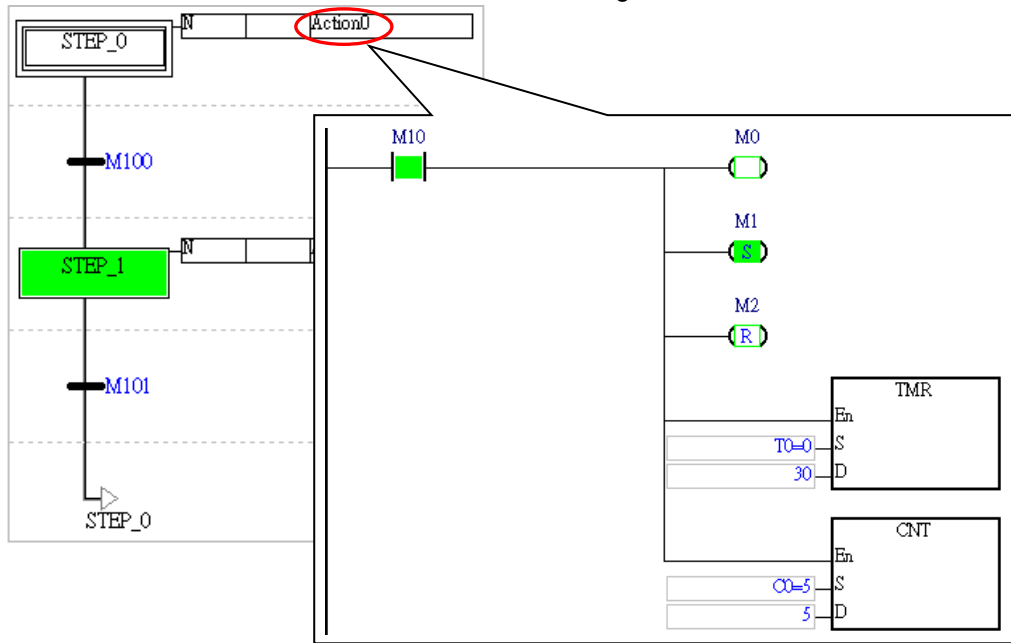
Action0 in the figure below is an action associated with STEP\_0, and is an N action. When STEP\_0 is deactivated and STEP\_1 is activated, the system executes a final scan.



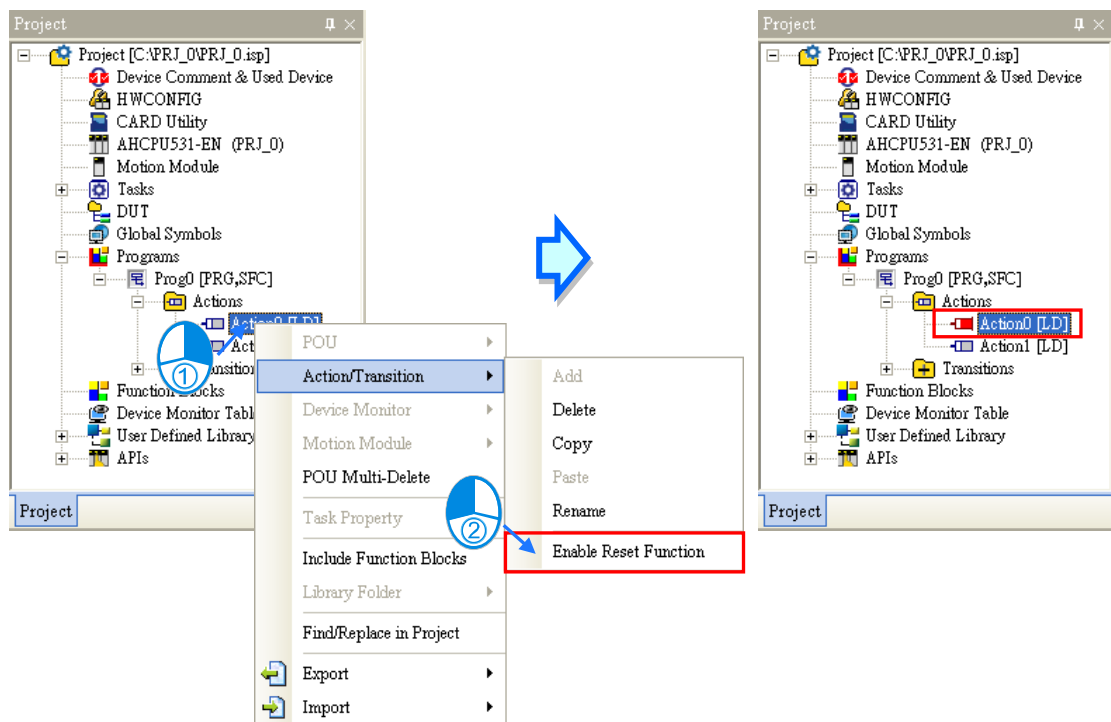
STEP\_0 in the figure below is activated. M10 in Action0 is ON, and therefore M0 is ON, M1 is set to ON, M2 is reset to OFF, the value in T0 is 30, and the value in C0 is 5.



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.



For the AH5x1, AH560 redundant system, AHxxEMC and AS series, users can set the Final Scan to run or not for every Action. The default is to run the Final Scan in the ISPSoft. To disable this function, users can right-click the Action that does not require a Final Scan to see a context menu and then deselect the Action/Transition > Enable Reset Function. After that users will see a red indicator situated in front of the Action that does not require a Final Scan. The state of an Action that does not run a Final Scan will stay the same as the previous state does, while the state of an Action that runs a Final Scan will be cleared.

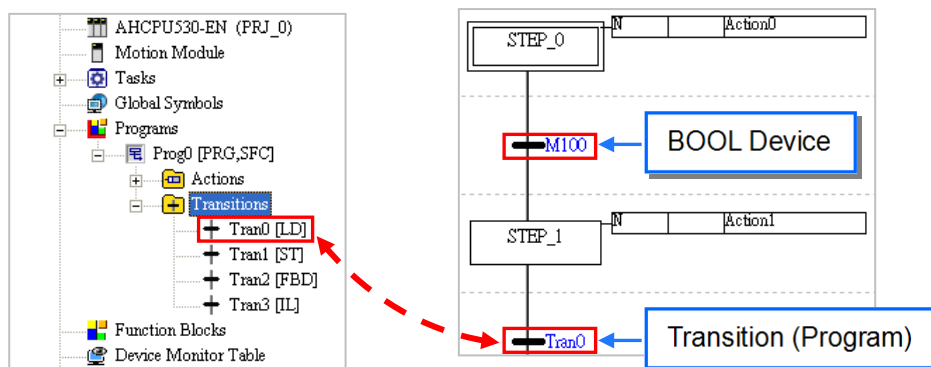


14

### 14.2.2 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.

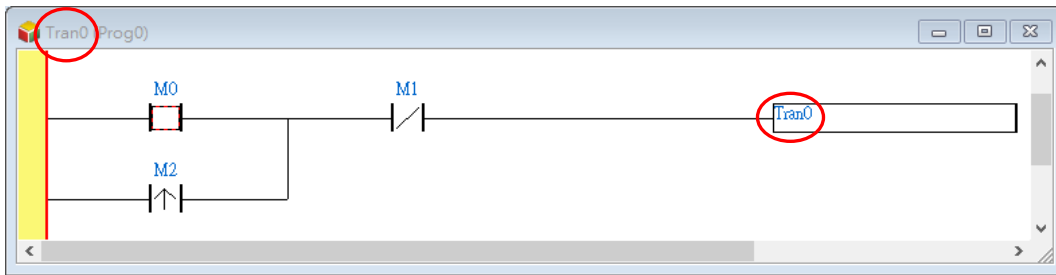
In ISPSOFT, a transition can be a Boolean device or symbol, or a program code. If a transition is a logical operation, users can create a transition program, and assign the program to the transition. The transition programs which are created are listed in the project management area.



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.

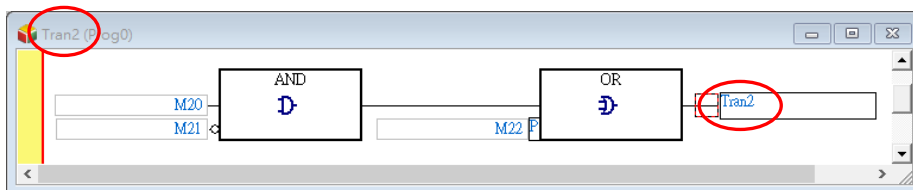


If the programming language used to create a transition program is a structured text, there is no limit on the number of lines of program code. However, a Boolean state in a transition program must be assigned to a symbol name which is the same as the transition program name.

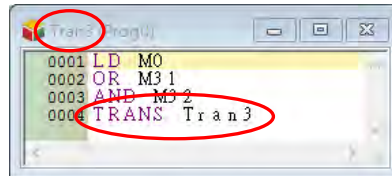
```

Tran1 (Prog1)
0001 IF M10 THEN
0002   Tran1 = TRUE;
0003 ELSE
0004   Tran1 = (M11 AND M12) OR (NOT M13);
0005 END_IF;
    
```

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 structure.

```
0001 IF (M0 AND M1) THEN
0002     Tran4 := TRUE;
0003 END_IF;
```

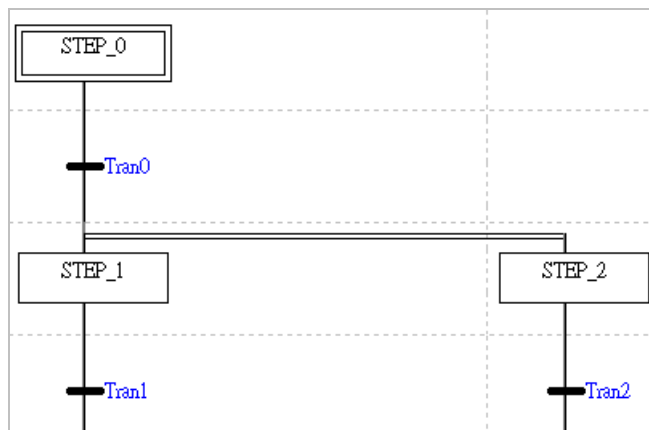
### 14.2.3 Simultaneous Divergence and Divergence of Sequence Selection

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.

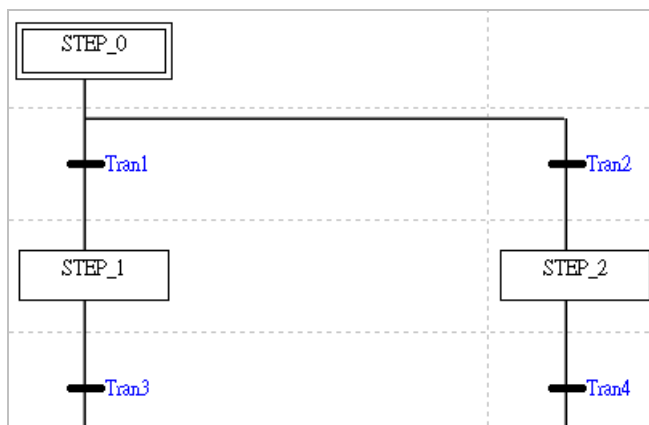
When Tran0 is true, STEP\_0 becomes inactive, and STEP\_1 and STEP\_2 become active.



- **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.



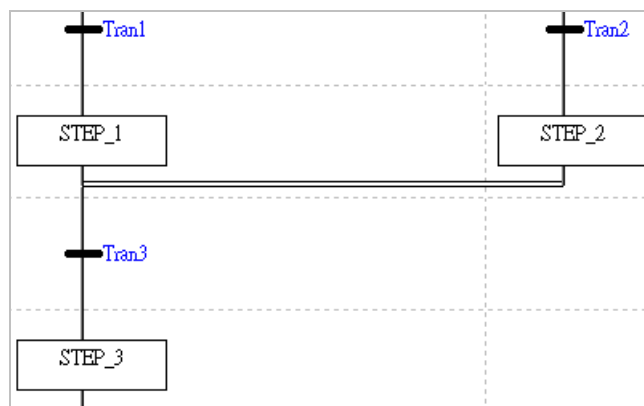
### 14.2.4 Simultaneous Convergence and Convergence of Sequence Selection

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.

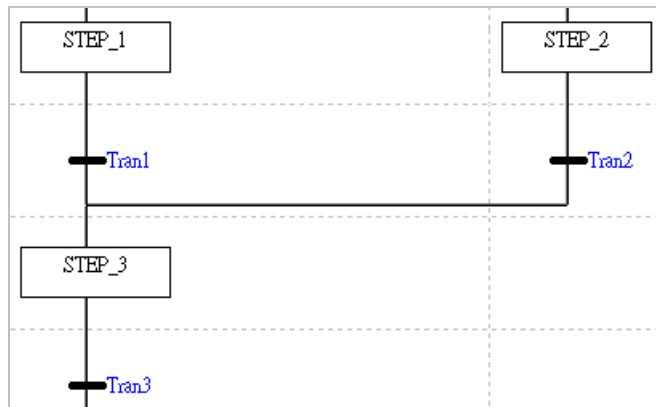


- **Convergence of sequence selection**

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.

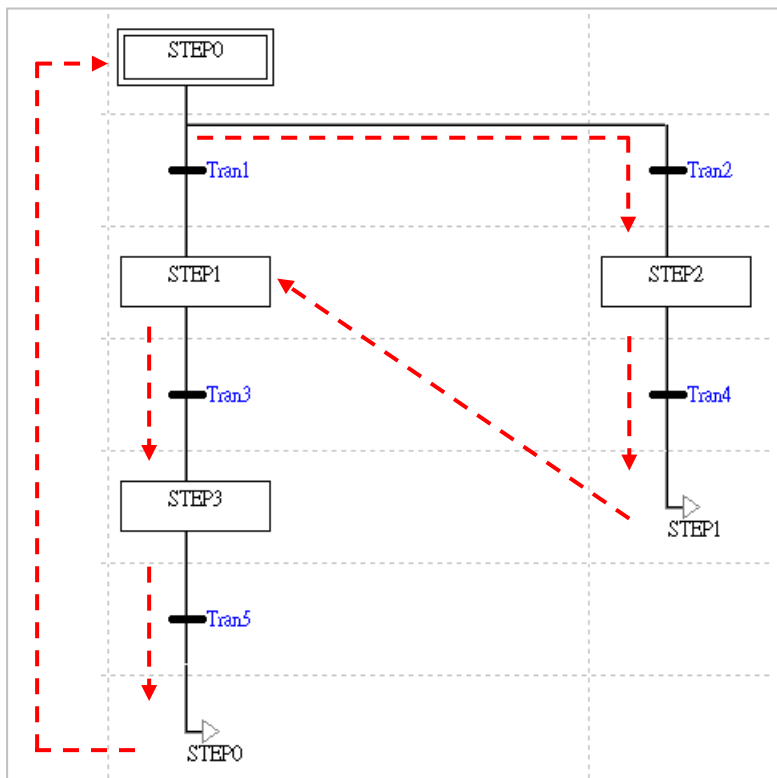




### 14.2.5 Jump

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.

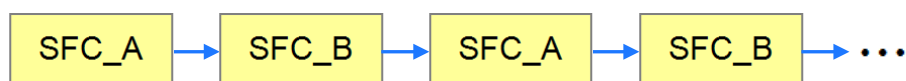
The program below is a sequential function chart where there is a divergence of sequence selection. If Tran2 is true, control will be passed to STEP2. When Tran4 is true, it passes control to STEP1 through a jump. When Tran5 is true, it passes control to STEP0.



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.

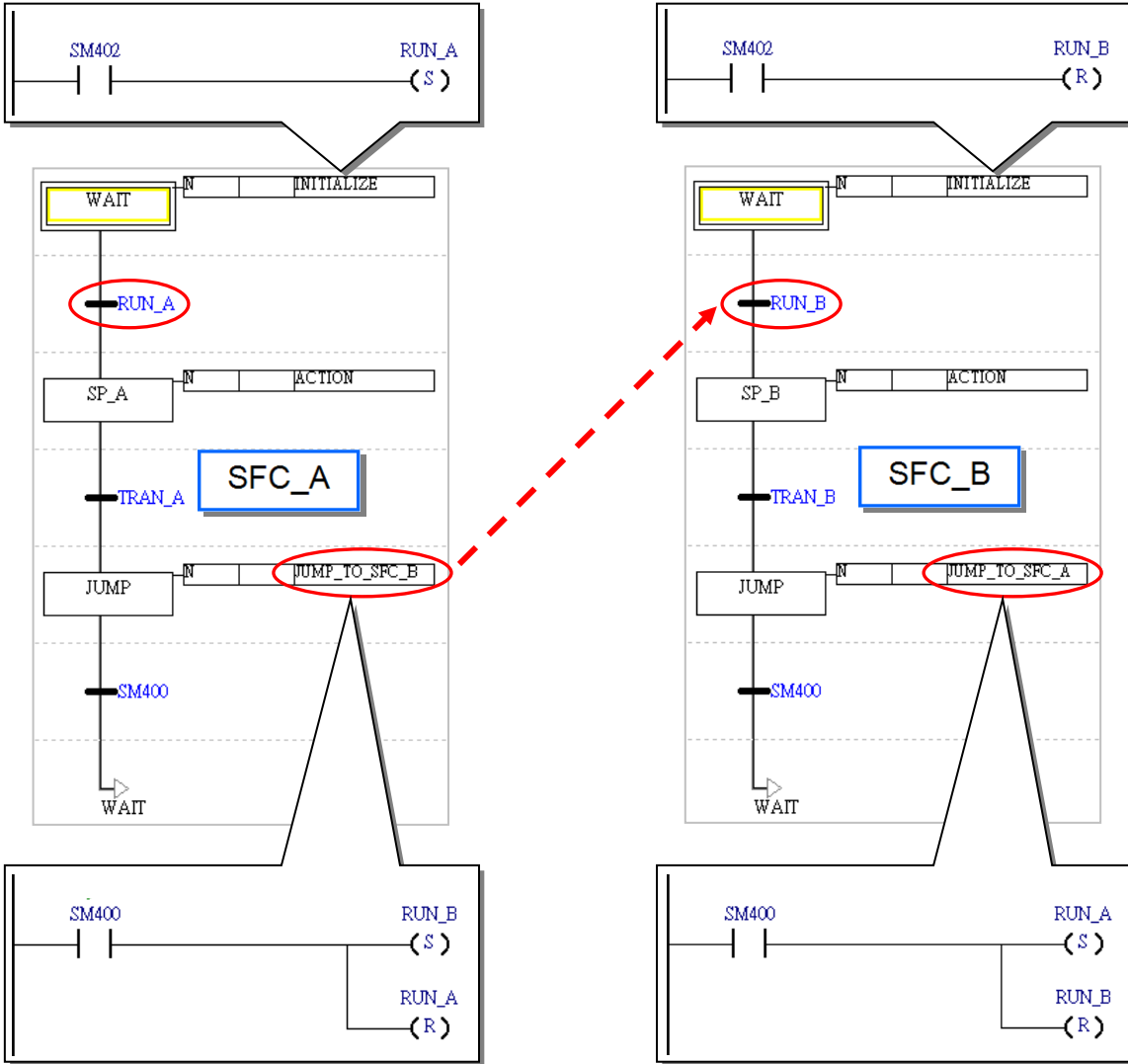


The steps **SP\_A** and **SP\_B** in the example below are used to express the sequential function chart which will be executed. They are preceded and followed by steps. Besides, the transitions **RUN\_A** and **RUN\_B** are declared in the global symbol table.

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.

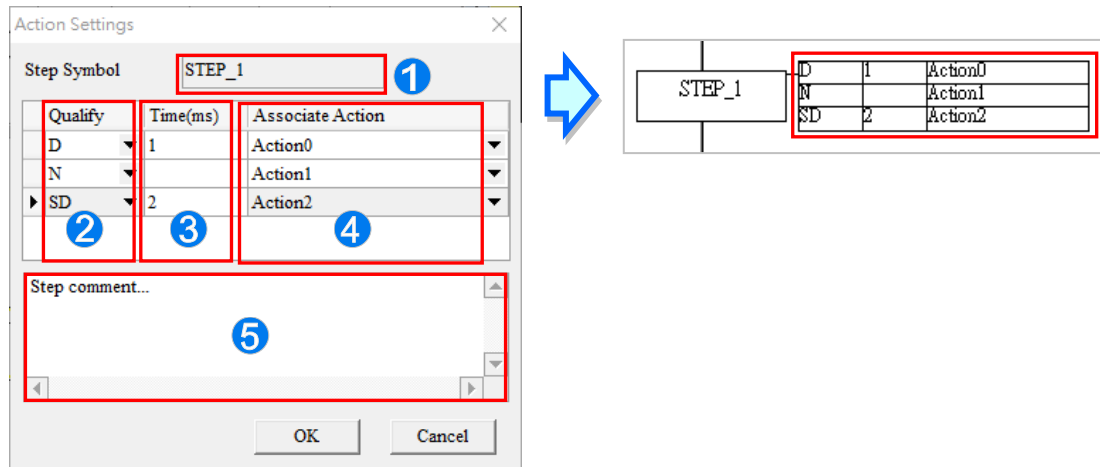
The use of **WAIT** and **JUMP** in SFC\_A serves to set **RUN\_B** to ON, and the use of **WAIT** and **JUMP** in SFC\_B serves to set **RUN\_A** to ON. **WAIT** and **JUMP** in SFC\_A and **WAIT** and **JUMP** in SFC\_B actually have little significance. If the program is applied to an AH/AS series CPU module, users can assign the qualifier **P** or **P1** to the actions associated these steps. Please refer to section 14.2.6 for more information.



14

## 14.2.6 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.

### 14.2.6.1 Qualifier Types

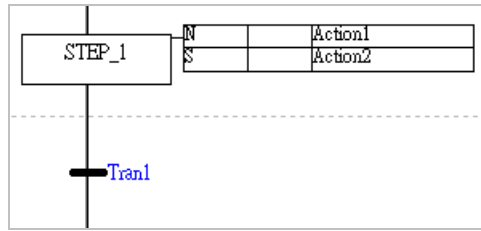
- **N (Normal)**

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.

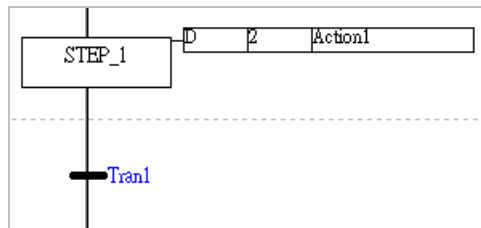
When STEP\_1 in the figure below is activated, Action1 and Action2 are executed. When Tran1 is true, STEP\_1 becomes inactive. Once STEP\_1 is inactivated, the execution of Action1 will stop, the system will execute a final scan, and Action2 will still be executed.



● **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.

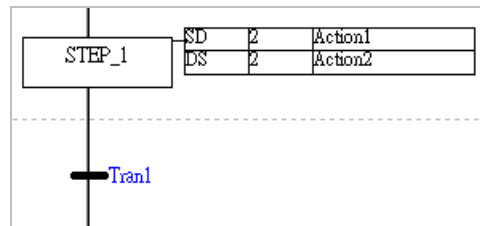
Action1 and Action2 are not executed at the time when STEP\_1 is activated. The execution of Action1 and the execution of Action2 are delayed for two seconds. There are two situations which need to be considered.

(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.

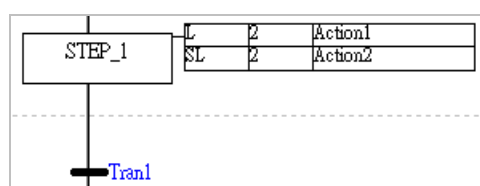
When STEP\_1 is activated, Action1 and Action2 are executed. There are two situations which need to be considered.

(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.

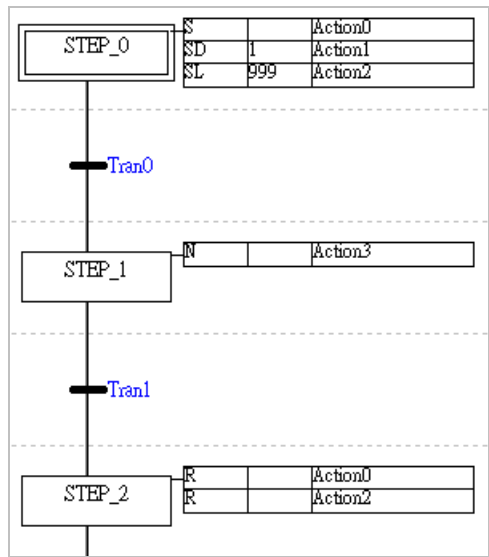
After control is passed to the next step, the execution of Action1 will stop, and the system will execute a final scan. The execution of Action2 will stop after two seconds have elapsed.



● **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.

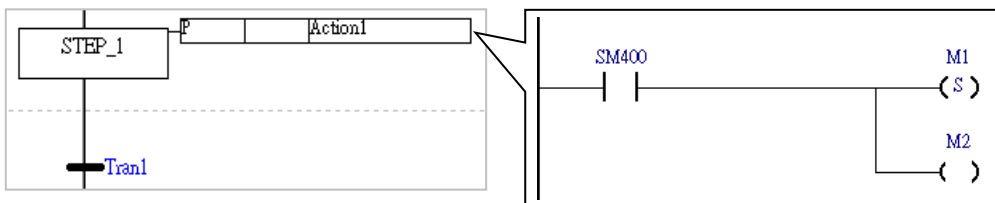
Suppose that Action0, Action1, and Action2 are executed during the activation of STEP\_0, and will be executed until control is passed to STEP\_2. After the control is passed to STEP\_2, the execution of Action0 and the execution of Action2 will stop, and the system will execute final scans. Owing to 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.

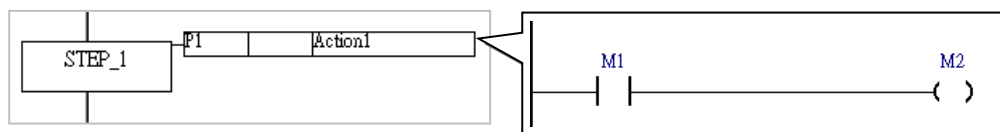
During the first scan cycle, Action1 is executed when STEP\_1 is activated. Owing to the fact that SM400 is ON, M1 and M2 are ON. During the second scan cycle, the system inactivates Action1. Although SM400 is still ON, M2 becomes OFF. M1 is driven by the instruction SET, and therefore it remains ON.



- **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.

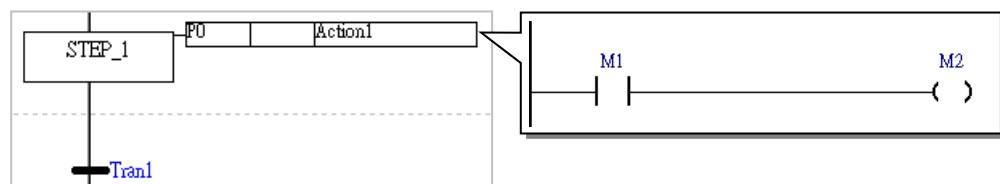
During the first scan cycle, Action1 is executed when STEP\_1 is activated. If M1 is ON, M2 will be ON. Action1 is executed only when STEP\_1 is activated during the first scan cycle. Besides, the system will not execute a final scan after the execution of Action1 stops. Consequently, the state of M2 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.

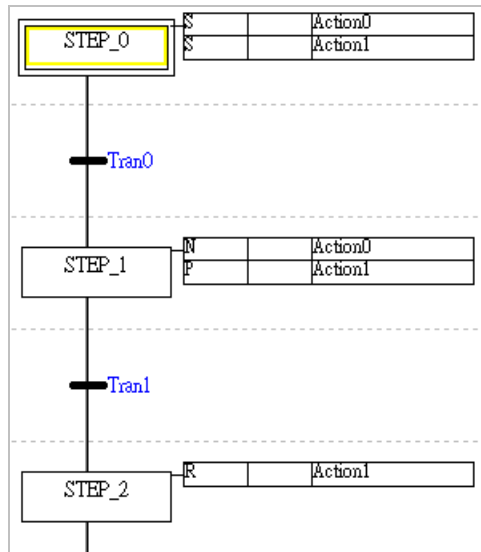




### 14.2.6.2 Important Points About Qualifying an Action

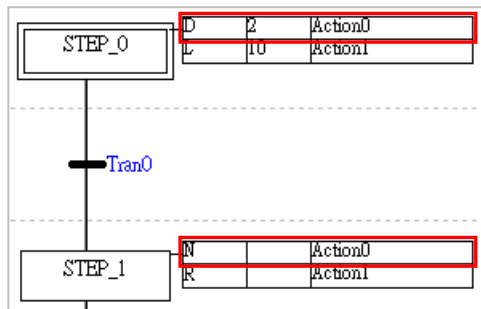
- Once an action qualified by S is executed, it will still be executed even if it is qualified by N or P. The execution of the action stops only when the action is qualified by R. If there is an action which is qualified by S, users have to check whether they have to assign R to the action to stop the 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.



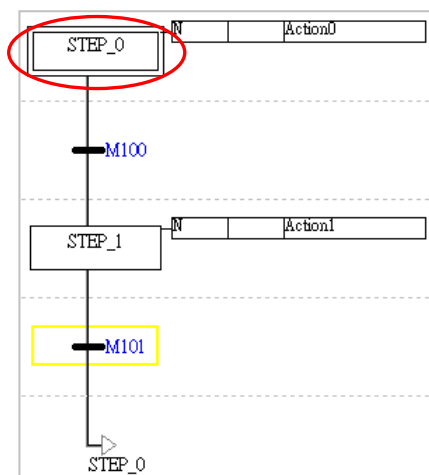
- Except the R qualifier, when assigning a time-related qualifier for an Action, the qualifier can only be used in the SFC instruction once. In other words, if an action is qualified by D, SD, DS, L, or SL, no other qualifiers can be used for the action, except the R qualifier.

From the example shown below, we can see that in the STEP\_0, the Action0 and Action1 are qualified by D and L respectively. And in the STEP\_1, the Action0 and Action1 are qualified by N and R. The usage of Action 0 being assigned with D in the STEP\_0 and N in the STEP\_1 is wrong and will cause error to occur. The usage of Action 1 being assigned with L in the STEP\_0 and R in the STEP\_1 is correct, since R can be added to an Action with more than 1 qualifier.



### 14.2.7 Initial Step

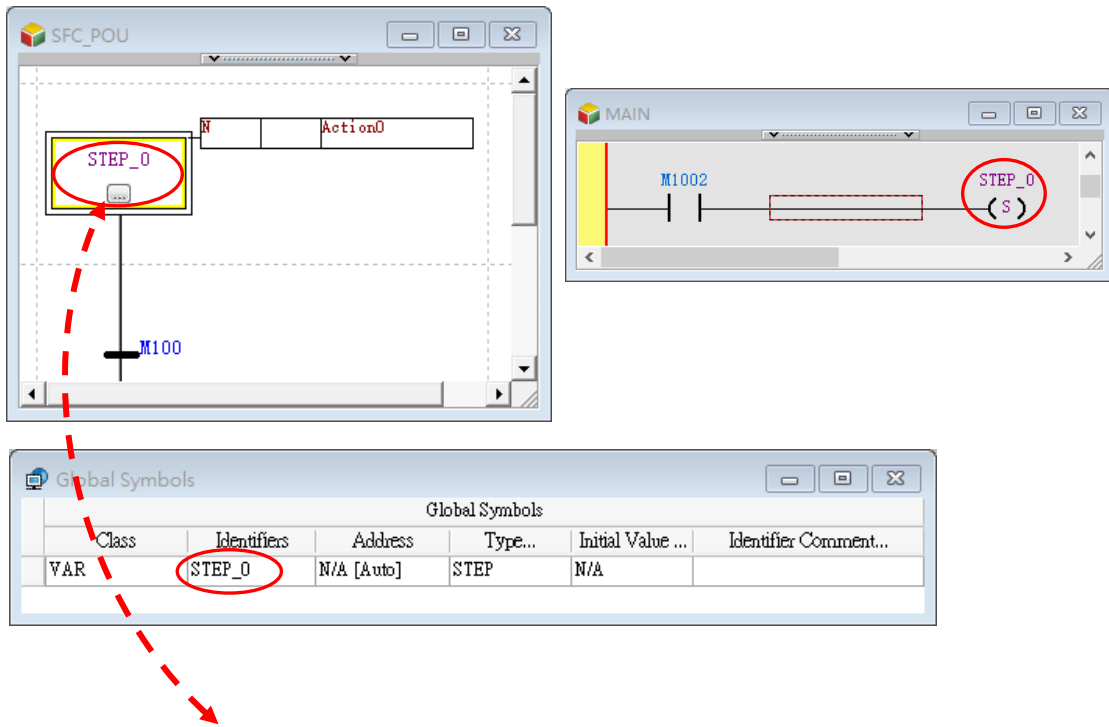
During the execution of a sequential function chart, every step is activated in turn and cyclically. It is necessary to define a step as an initial step which is activated when the program is executed for the first time. In ISPSOft, the initial step in a sequential function chart is drawn as a box with a double line. Besides, there is only one initial step in a sequential function chart.



Any step in a sequential function chart in an AH/AS series CPU module can be defined as the initial step in the sequential function chart. When a sequential function chart in an AH/AS 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.



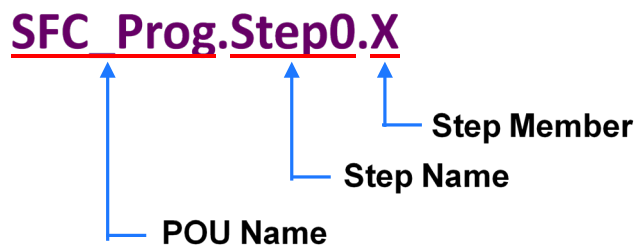
14

### 14.2.8 Internal Property

The internal property provides users to use in POUs and particular parameter properties in SFC programming. The parameter properties are categorized into Step, Action and Transition. The following content introduce the properties accordingly.

#### 14.2.8.1 Step Property

To use SFC in POU, the step property adopts STRUCTURE framework to display. The structure consists of POU Name, Step Name and Step Member. Users can select the appropriate POU and Step name then decide on the step member.

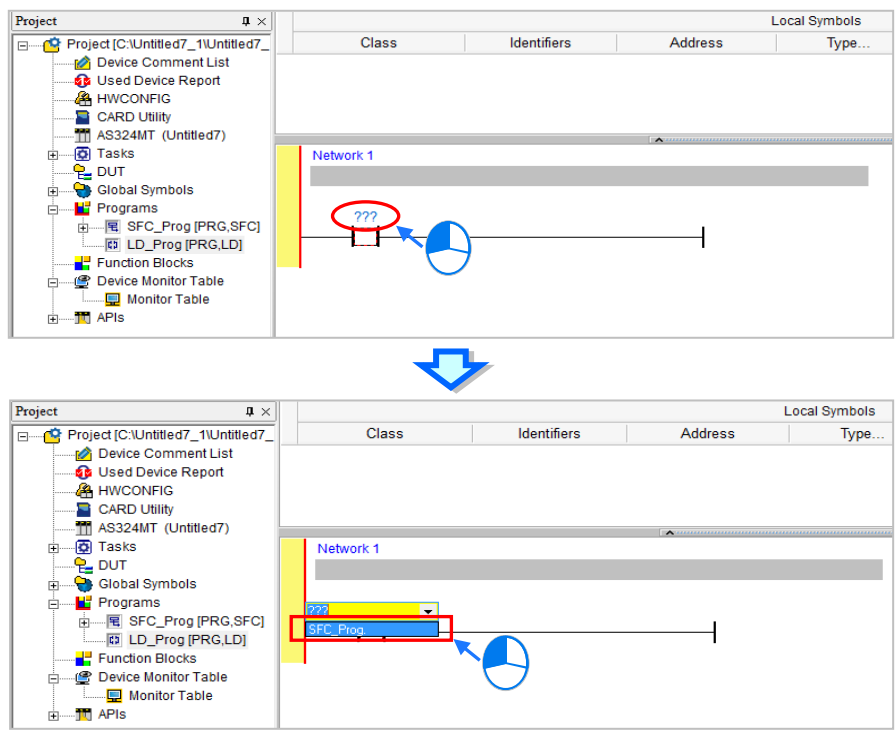


The following table provides Step member categorization.

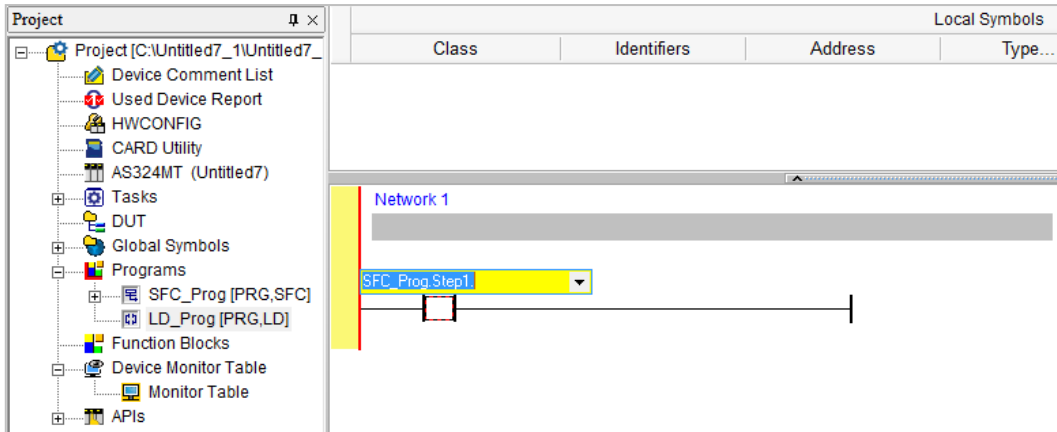
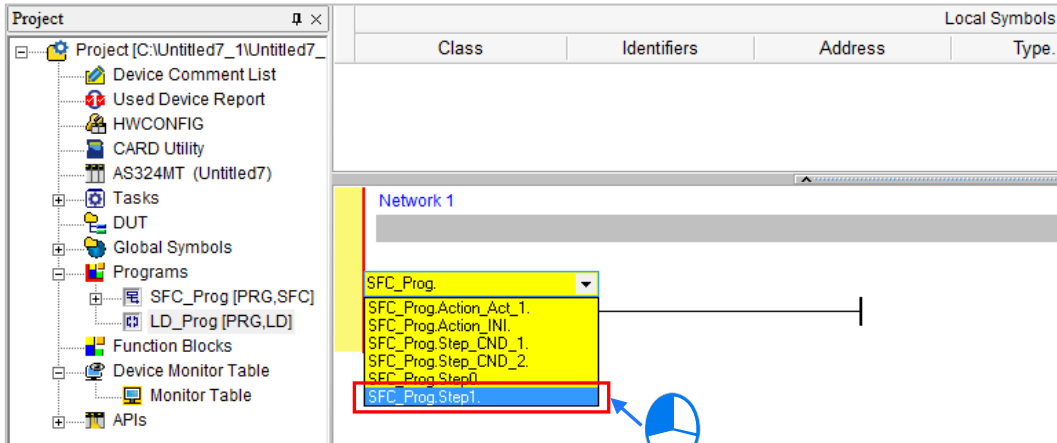
Category	Member	Description	Read	Write
Step	X	When executing Step, the SFC parameter property is ON ; this parameter property can replace previous S device.	V	X
	done	When Step Property <b>T</b> (the <b>current time spent executing Step</b> ) $\geq$ Step property <b>timePreset</b> and the value is ON	V	X
	counter	When Step property <b>X</b> changes from OFF to ON, parameter property count adds 1	V	X
	timePreset	Determines the time for Step property <b>done</b> changes from OFF to ON (unit: <b>ms</b> , data length: <b>DWORD</b> )	V	X
	maxTime	Record the maximum time of Step property <b>T</b> (unit: <b>ms</b> , data length: <b>DWORD</b> )	V	X
	T	The time spent executing Step (unit: <b>ms</b> , data length: <b>DWORD</b> )	V	X

14

Take executing Step member X in SFC for example, when users want to trigger the contact to ON for POU in ladder programming once Step execution starts, the programs are written in the following ways. By using normally-open contact of POU in ladder program, click on top of the normally-open contact, a drop-down list appears and provides POU name options for SFC program then click the name.

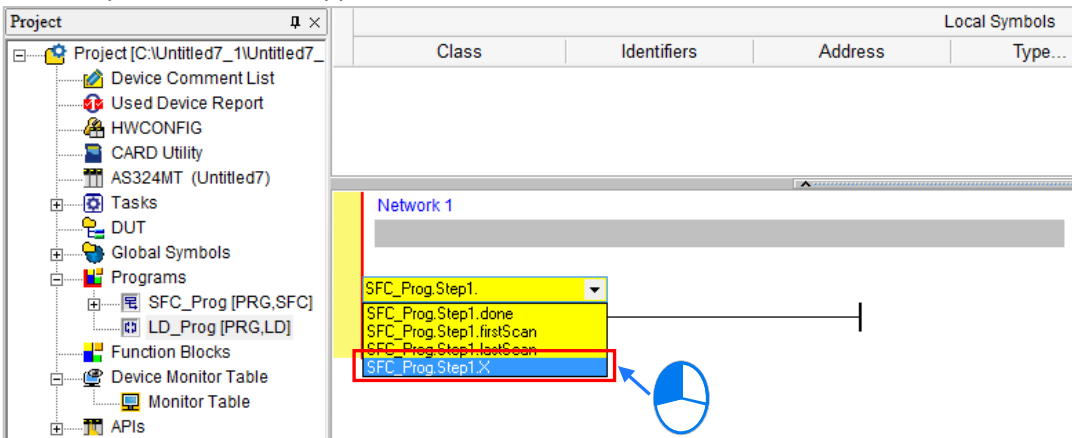


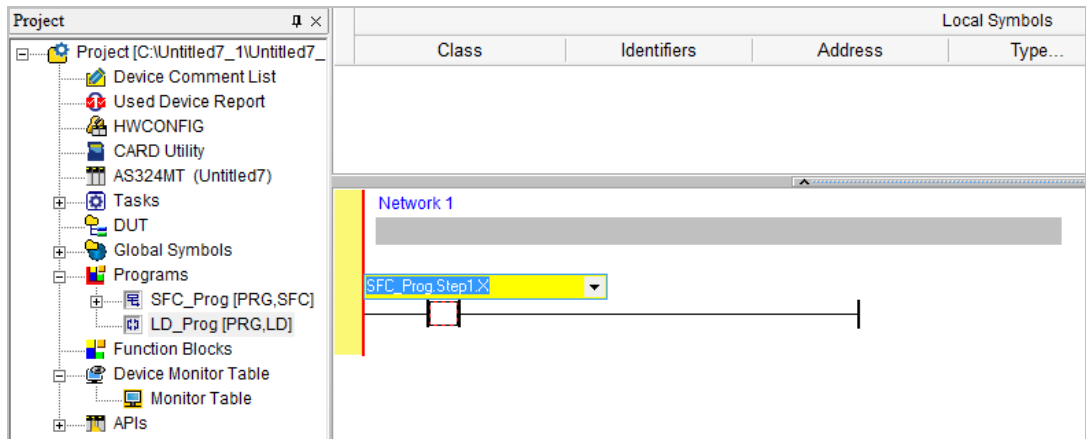
After clicking the POU name for SFC program, right-click the mouse and a drop-down list with POU Step name appears, then select the name.



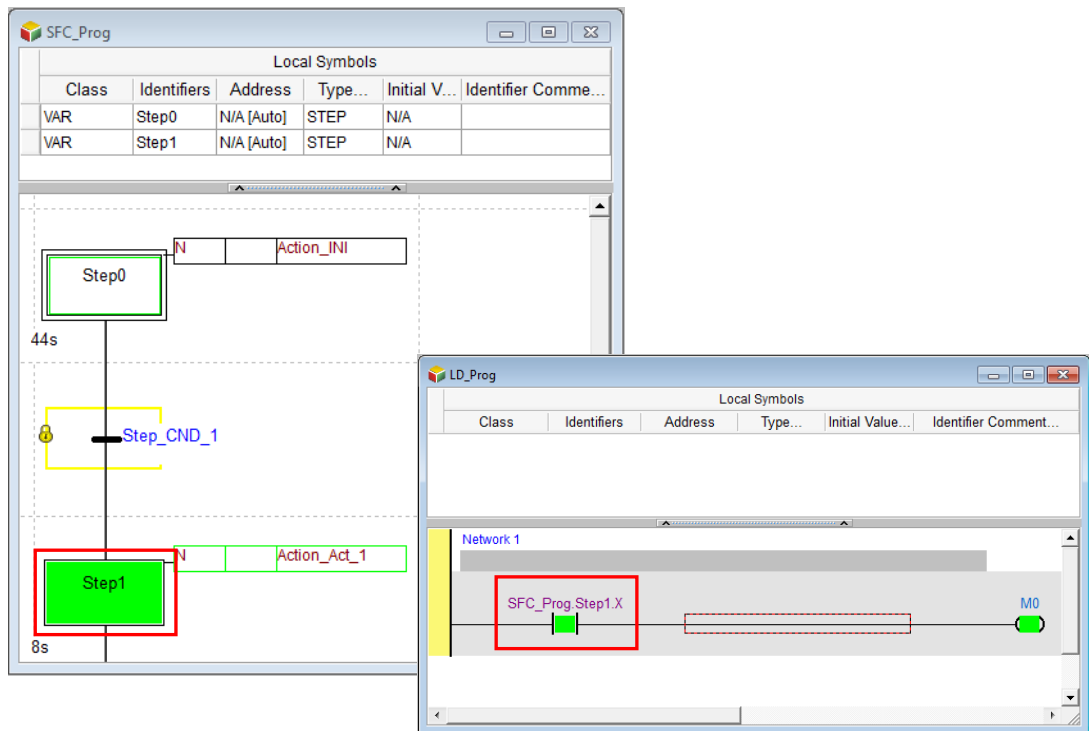
14

After clicking the POU Step name for SFC program, right-click the mouse and a drop-down list with POU Step Member name appears, then select the name.





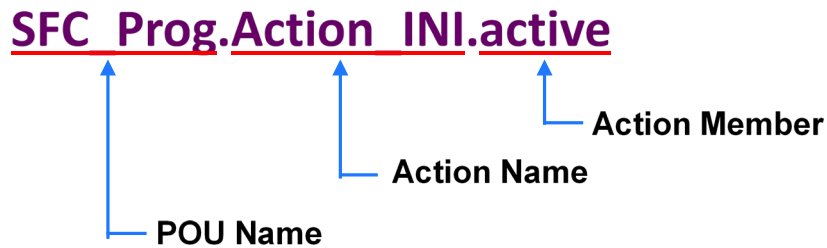
When executing POU Step in SFC program, the normally-open contact of SFC\_Prog.Step1.X is ON.



14

### 14.2.8.2 Action and Transition Property

To use SFC in POU, the Action and Transition property adopts STRUCTURE framework to display. For Action property structure consists of POU Name, Action Name and Action Member. Users can select the appropriate POU and Action name then decide on the Action member.



For Transition property structure consists of POU Name, Transition Name and Transition Member. Users can select the appropriate POU and Transition name then decide on the Transition member.

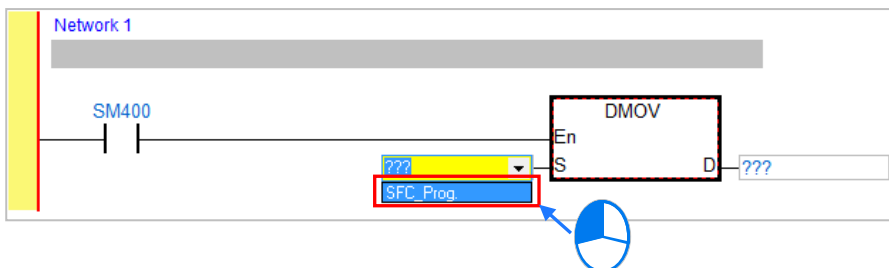
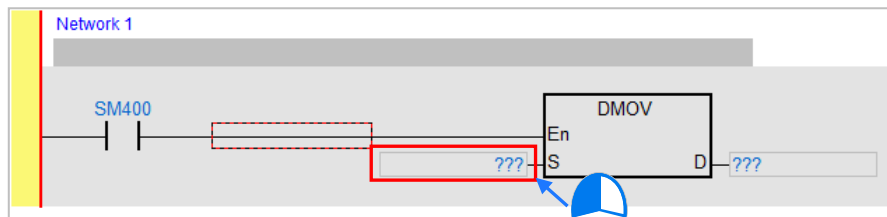


The following table provides Action and Transition member categorization.

Category	Member	Description	Read	Write
Action	enabled	When qualifier condition of Action is set, the SFC parameter property is ON and checks if Action is enabled or not.	V	X
	active	When finish Action program for the first time, the SF parameter property is ON.	V	X
	rstDisable	When Final scan is enabled, its value is OFF; when Final scan is not enabled, its value is ON.	V	X
	counter	When Active property <b>active</b> changes from OFF to ON, parameter property count adds 1.	V	X
	setTimer	Shows time for time related qualifier (unit: <b>ms</b> , data length: <b>DWORD</b> )	V	X
Transition	transOut	When Transition condition set is ON, or Transition is forced to start is ON.	V	X

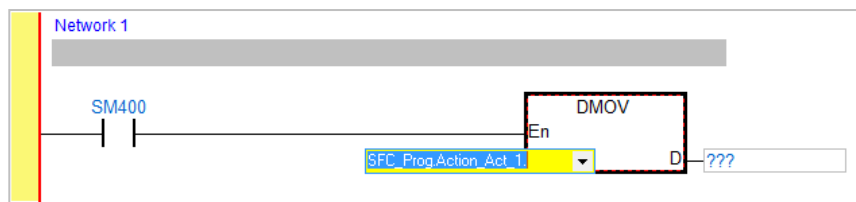
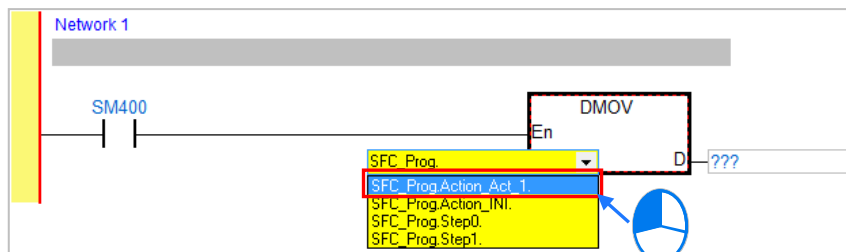
Take Action member counter for example, if users want to monitor Action property active number change from 0 to 1 and transmit the parameter value to D100 register, the programs are written in the following ways.

Use SM400 (The flag is always ON when CPU runs) and DMOV command in ladder program, click the S contact of DMOV command and choose the POU name for SFC program from the drop-down list and click the name.



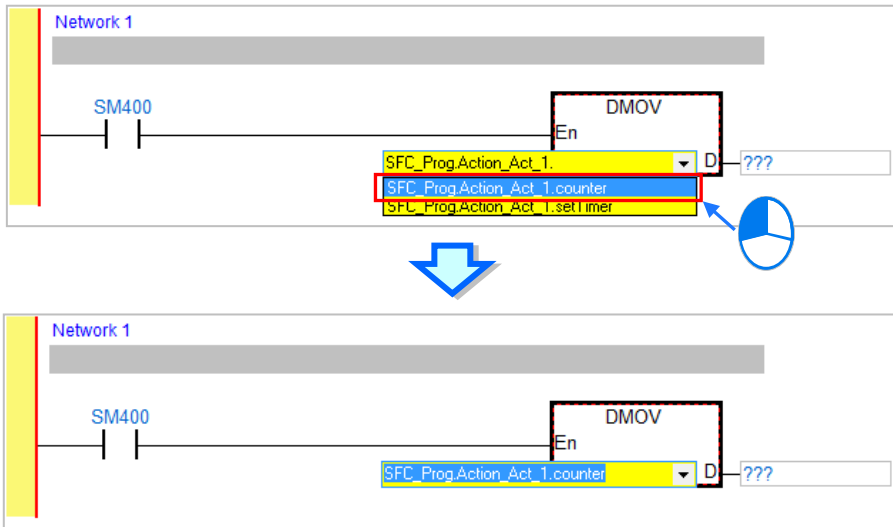
14

After clicking the POU name for SFC program, right-click the mouse and a drop-down list with POU Action name appears, then select the name.



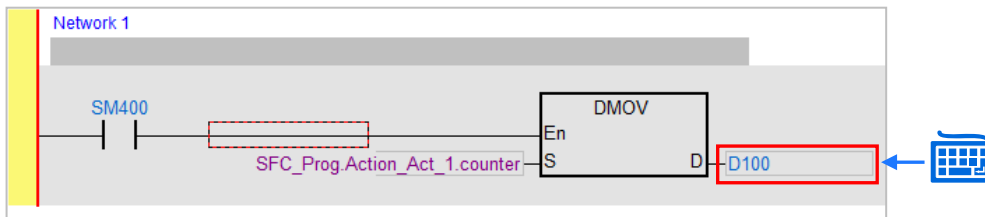


After clicking the POU Action name for SFC program, right-click the mouse and a drop-down list with POU Action Member name appears, then select the name.



For D contact of DMOV command, input D100.

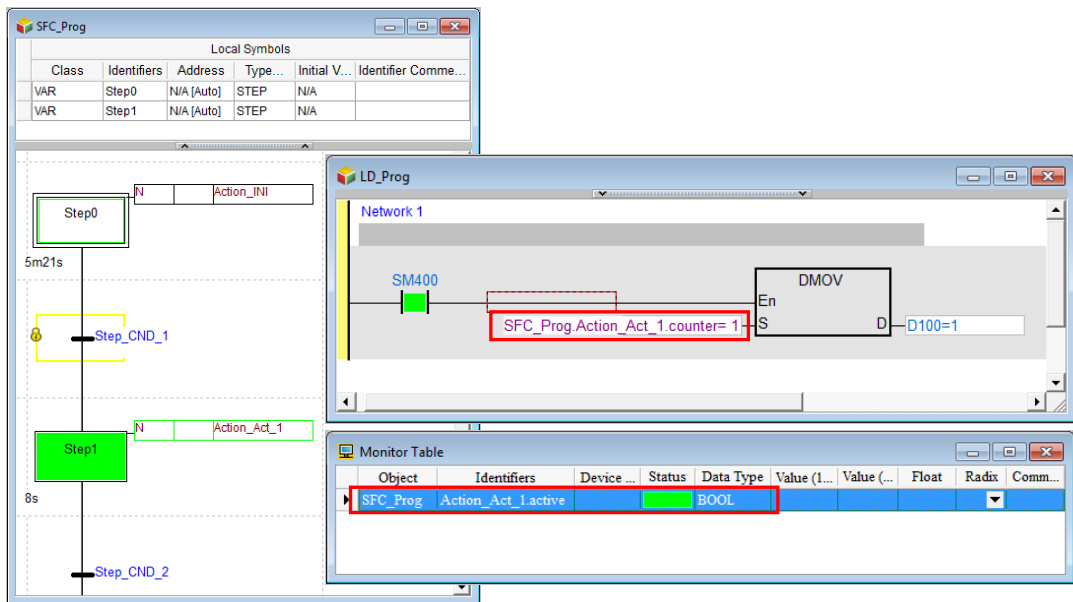
14



When SFC\_Prog.Action.Act\_1.Active changes from 0 to 1, the SFC\_Prog.Action.Act\_1.counter adds 1.

The screenshot displays the software's multi-window interface. On the left, the SFC\_Prog window shows a state transition diagram with Step0 and Step1. In the center, the LD\_Prog window shows the ladder logic network with the DMOV command's 'D' input set to 'D100=0'. At the bottom, the Monitor Table window shows the following data:

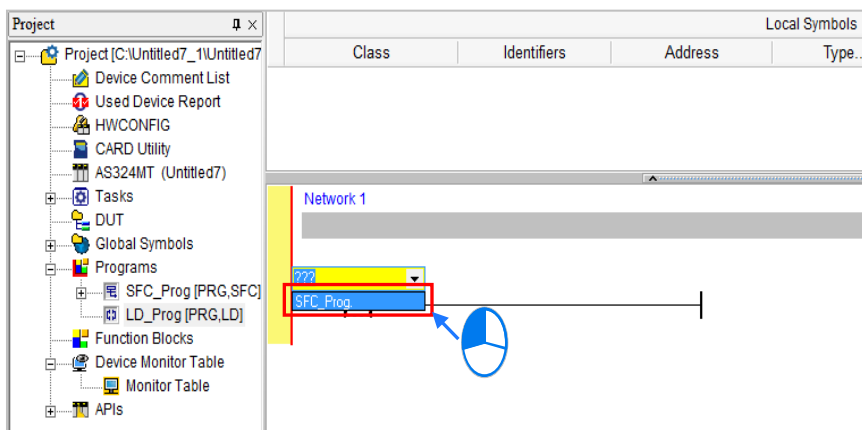
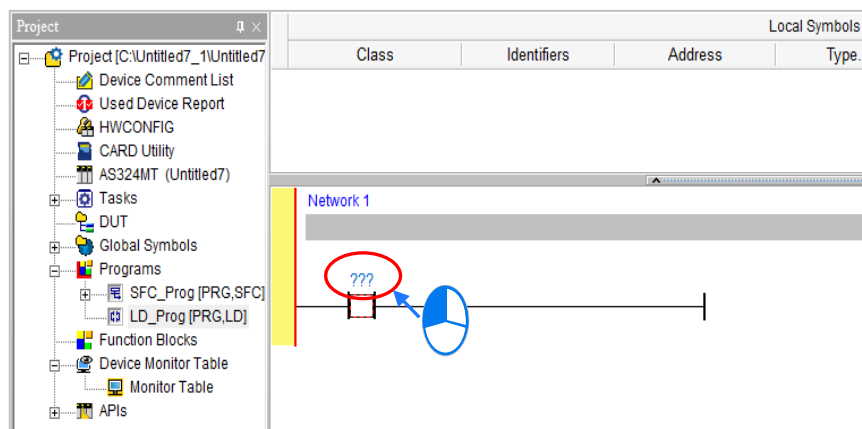
Object	Identifiers	Device	Status	Data Type	Value (1...	Value (...	Float	Radix	Comm...
SFC_Prog	Action_Act_1.active			BOOL					



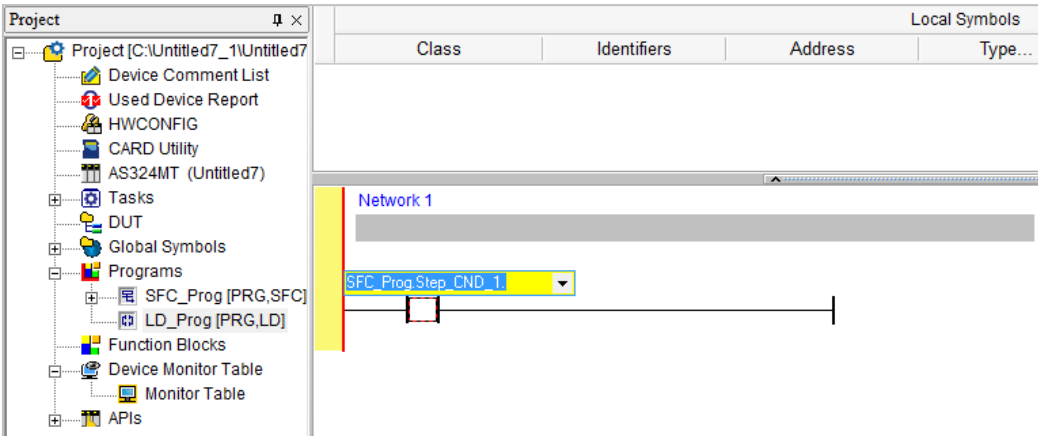
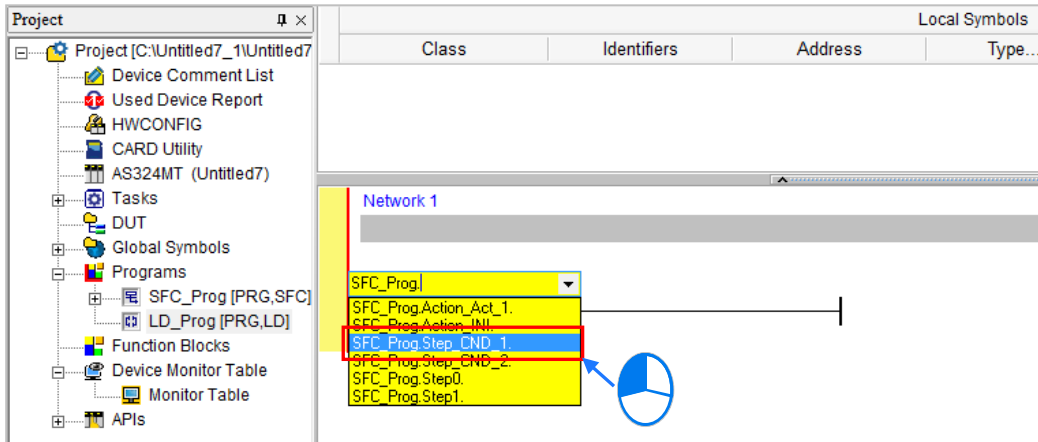
Take Transition member transOut for example, if users want Transition condition set in POU, trigger the normally-open contact in POU for ladder program to ON, the programs are written in the following ways.

Use a normally-open contact and click on top of the contact and choose the POU name for SFC program from the drop-down list and click the name.

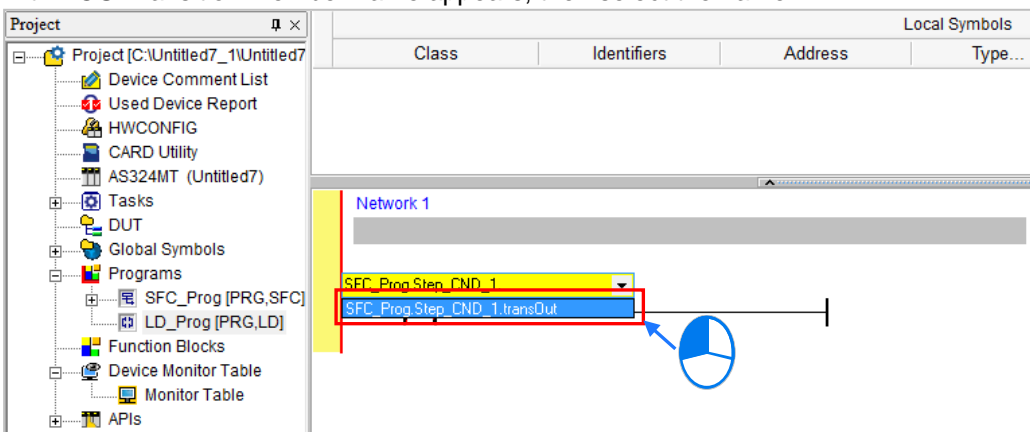
14

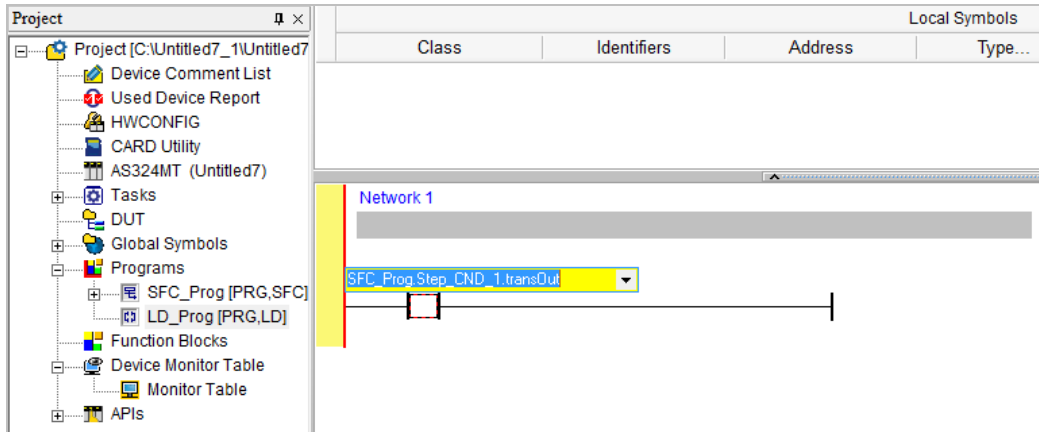


After clicking the POU name for SFC program, right-click the mouse and a drop-down list with POU Transition name appears, then select the name.

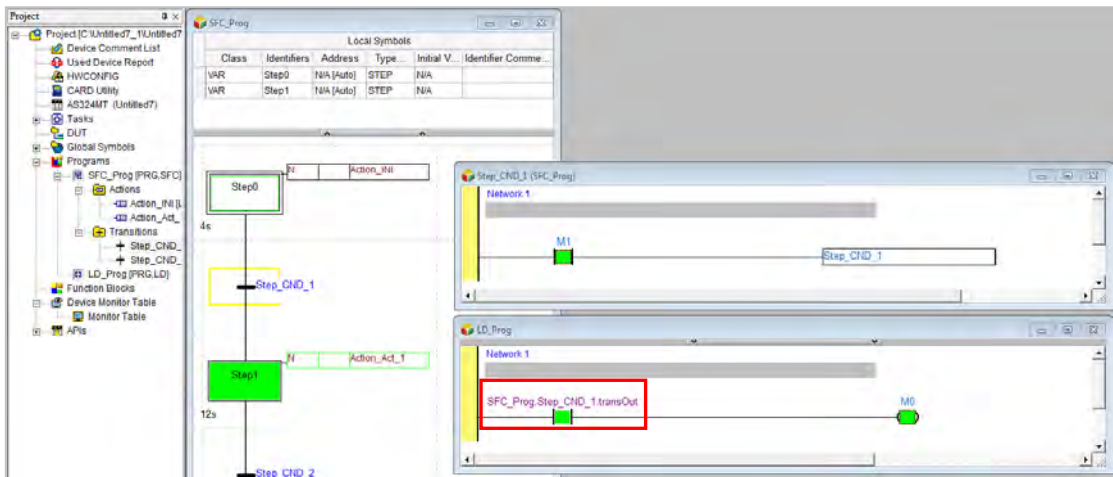


After clicking the POU Transition name for SFC program, right-click the mouse and a drop-down list with POU Transition Member name appears, then select the name.





When Transition condition Step\_CND\_1 in POU for SFC program is set, the normally-open contact of SFC\_Prog. Step\_CND\_1.transOut is ON.



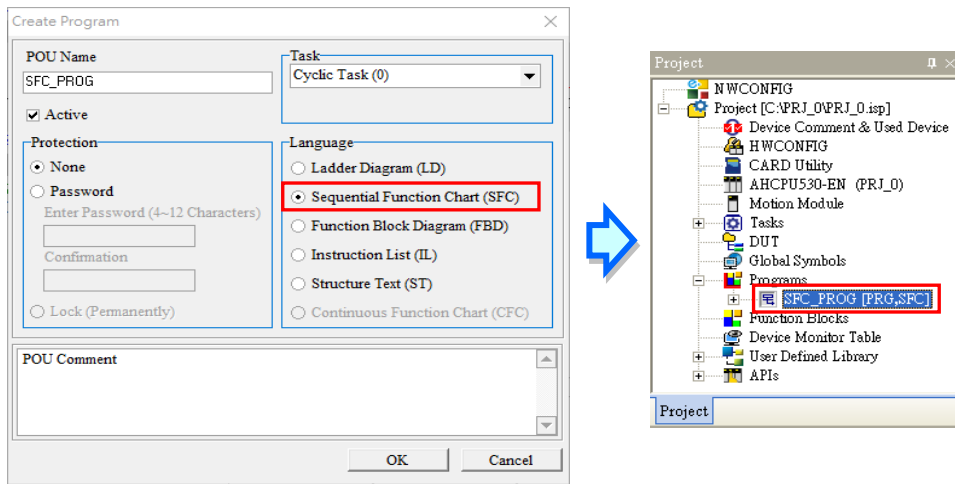
14

## 14.3 Create SFC in ISPSOft

### 14.3.1 Editing Environment

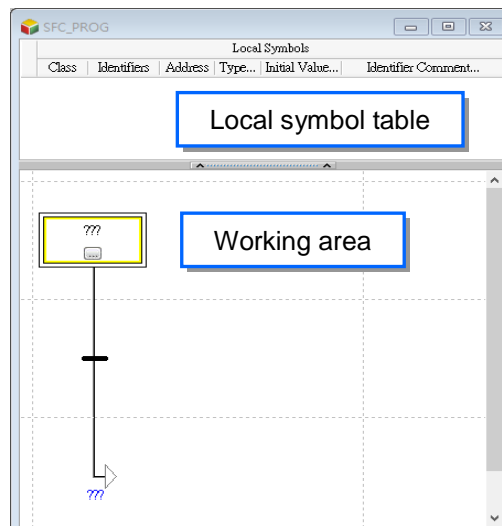
Select the **Sequential Function Chart (SFC)** option button in the **Language** section in the **Create Program** window. Please refer to section 5.4.1 for more information about creating a sequential function chart. In addition, users have to pay attention to the following two points when they create 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.

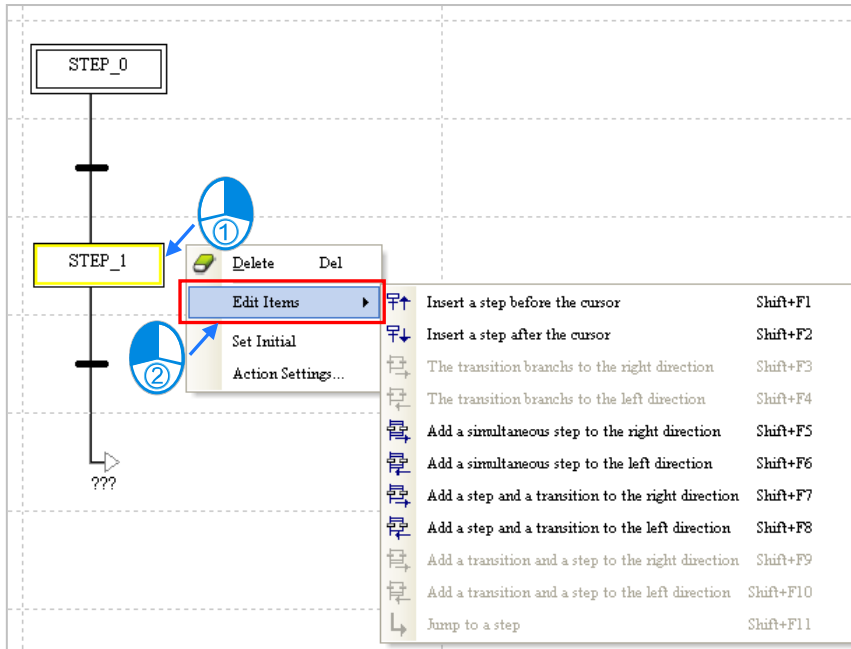


After a program editing window in which a sequential function chart can be created is opened, the corresponding toolbar will appear in the ISPSOft window. The functions are described below.



Icon	Keyboard	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

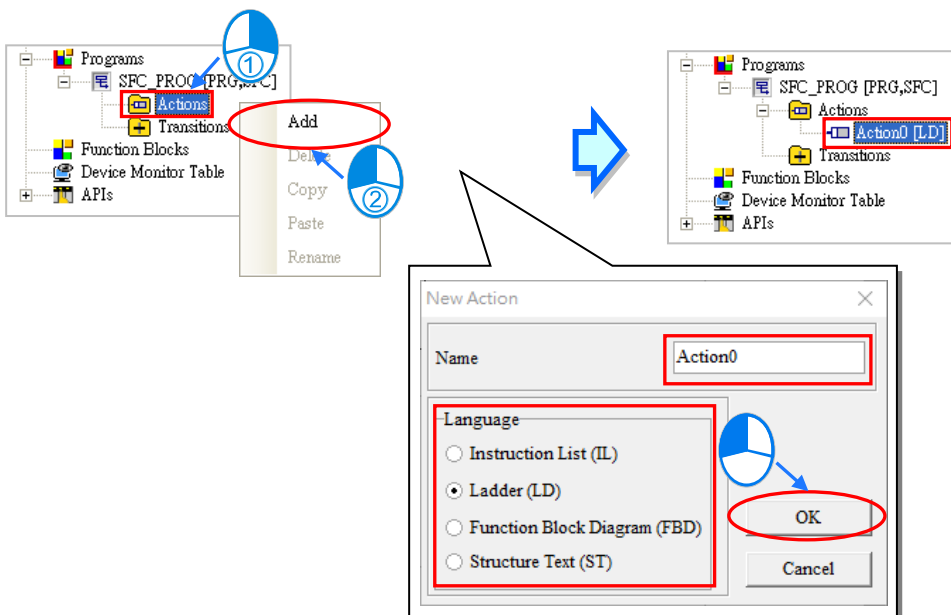
Users can edit the sequential function chart by means of the toolbar. They can also edit the sequential function chart by clicking an object in the working area, and clicking an item on the context menu. To ensure that the sequential function chart is a complete and legal sequential function chart, the editing function which can be used varies with the object selected.



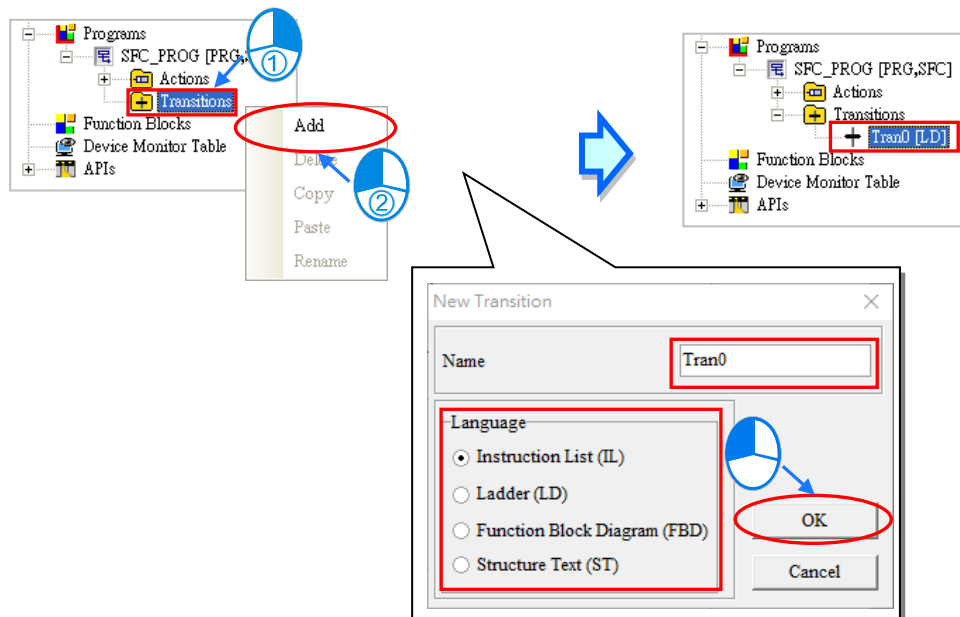
14

### 14.3.2 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**, and then click **Add**. Type an action name in the **New Action** window, and select a programming language in the **New Action** window.



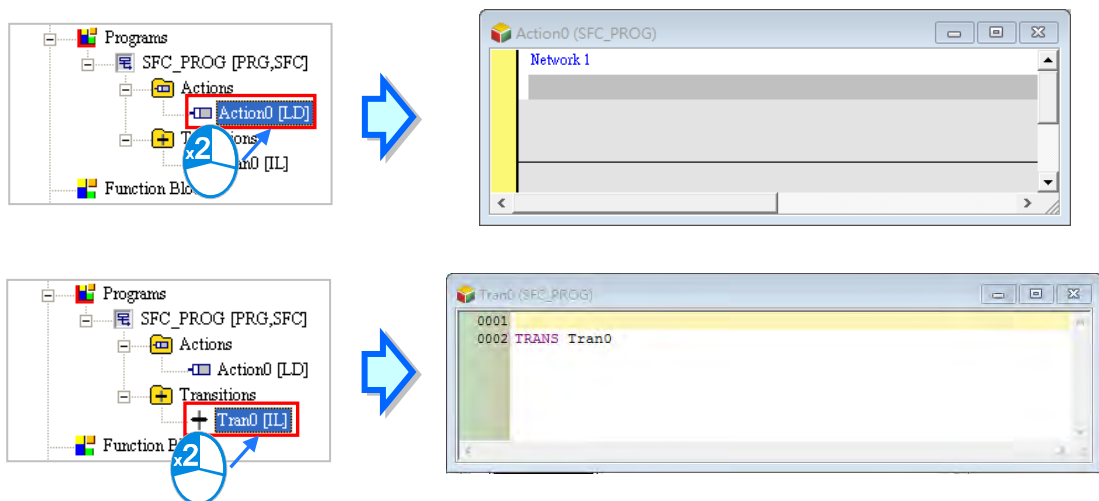
If the users want to create a transition, they have to unfold the sequential function chart section, selection Transitions, right-click Transitions and then click **Add**, type a transition name in the **New Transition** window, and select a programming language in the **New Transition** window.



14

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.

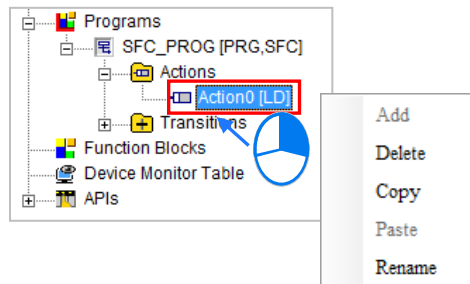
The action and the transition are edited in much the same way as a general POU is. However, there are no local symbol tables in the window for the action and the window for the transition. If users want to declare a local symbol table, they have to declare it in the window for the sequential function chart. Besides, the actions and the transitions in a sequential function chart share the same local symbol table.



\*. 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.

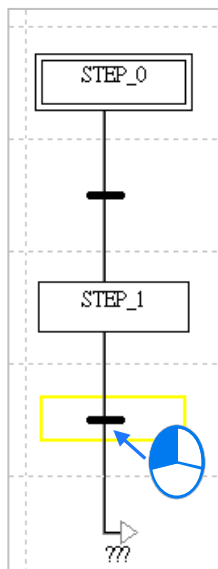




If the users want to manage the action or the transition created in the project management area, they can right-click the action or the transition and then click **Add**. The users have to notice that the action can not be copied into the **Transition** section, and the transition can not be copied into the **Action** section.

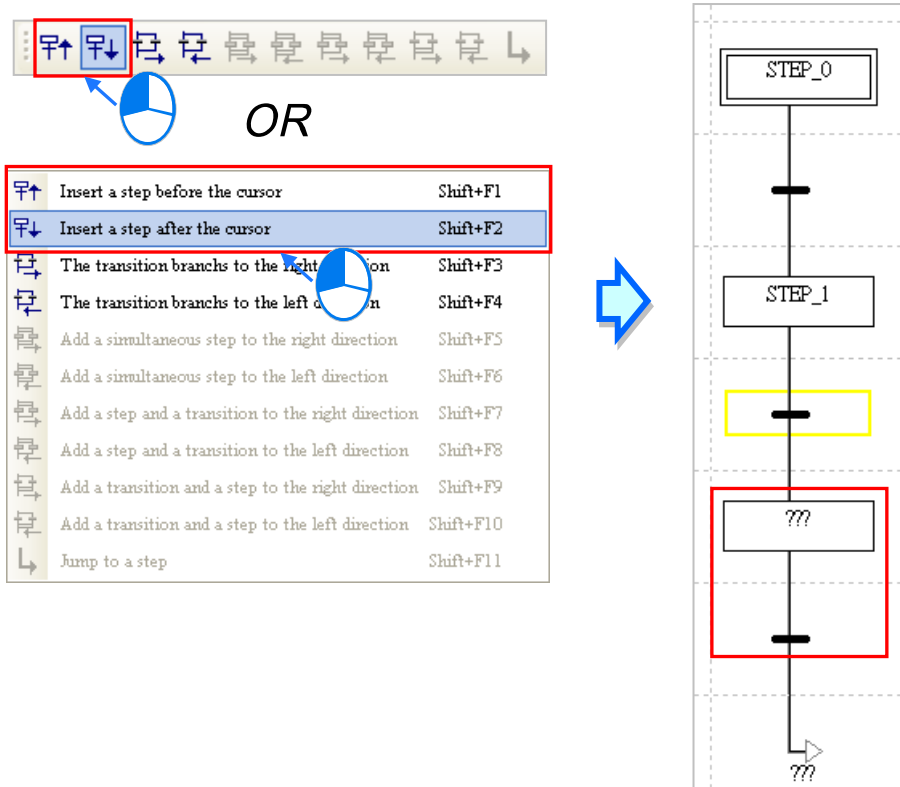


### 14.3.3 Adding a Step

- (1) Select a step in the working area.



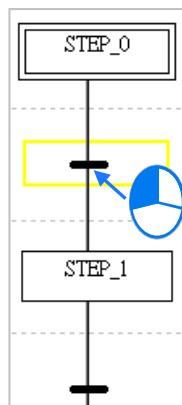
- (2) If users want to put a step to the top or the bottom of the step selected, they can click  for the top or  for the bottom direction on the toolbar. Or users can also right-click the step selected and click the suitable option, for the step to be added to the before or after the cursor.





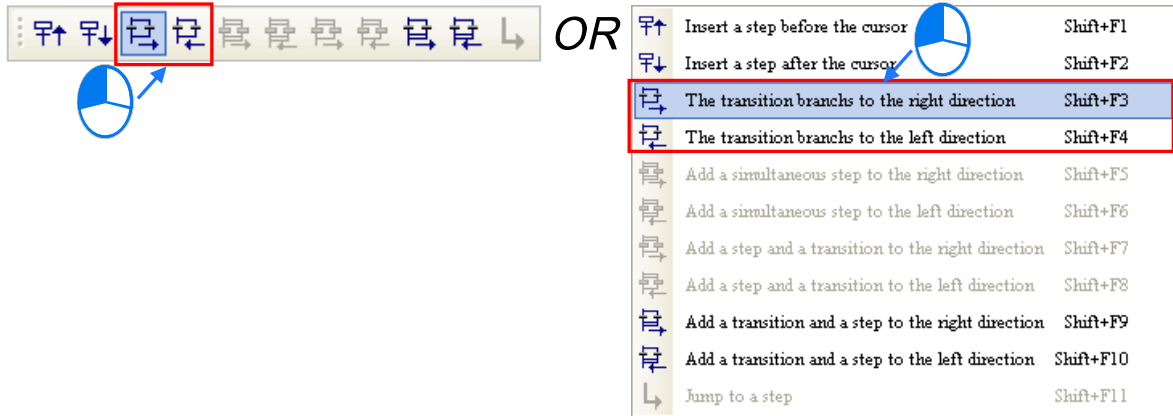
14

### 14.3.4 Connecting a Transition in Parallel












- (1) Select a transition in the working area.

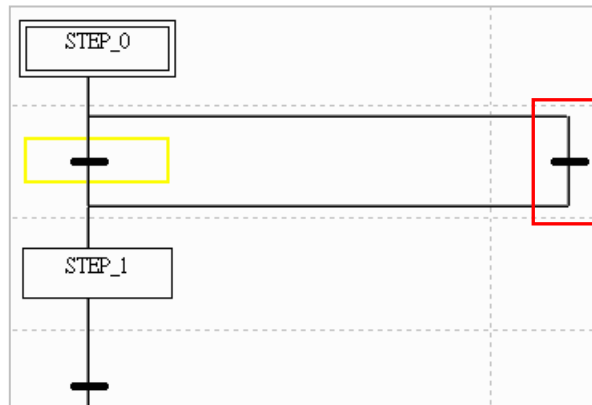


- (2) If users want to put a transition at the right/left side of the transition selected, they can click  on the toolbar for the right direction or  for the left direction. Or users can also right-click the transition selected and click the suitable option, for the transition to be branched to the right or left direction.



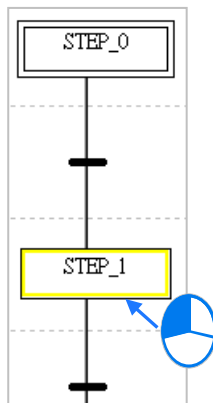
The image shows a toolbar with icons for inserting steps and transitions. A red box highlights the right and left branch icons. To the right, a context menu is shown with the following options:



	Insert a step before the cursor	Shift+F1
	Insert a step after the cursor	Shift+F2
	The transition branches to the right direction	Shift+F3
	The transition branches to the left direction	Shift+F4
	Add a simultaneous step to the right direction	Shift+F5
	Add a simultaneous step to the left direction	Shift+F6
	Add a step and a transition to the right direction	Shift+F7
	Add a step and a transition to the left direction	Shift+F8
	Add a transition and a step to the right direction	Shift+F9
	Add a transition and a step to the left direction	Shift+F10
	Jump to a step	Shift+F11

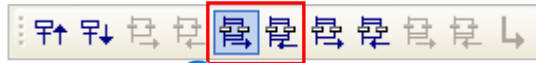


### 14.3.5 Connecting a Step in Parallel












- (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/left side of the step selected, they can click  on the toolbar for the right direction or  for the left direction. Or users can also right-click the simultaneous step selected and click the suitable option, for the simultaneous step to be added to the right or left direction.



OR

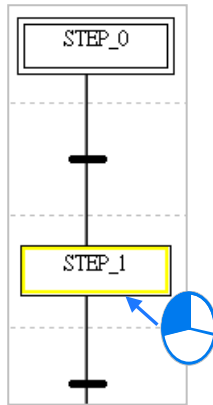
	Insert a step before the cursor	Shift+F1
	Insert a step after the cursor	Shift+F2
	The transition branches to the right direction	Shift+F3
	The transition branches to the left direction	Shift+F4
	Add a simultaneous step to the right direction	Shift+F5
	Add a simultaneous step to the left direction	Shift+F6
	Add a step and a transition to the right direction	Shift+F7
	Add a step and a transition to the left direction	Shift+F8
	Add a transition and a step to the right direction	Shift+F9
	Add a transition and a step to the left direction	Shift+F10
	Jump to a step	Shift+F11





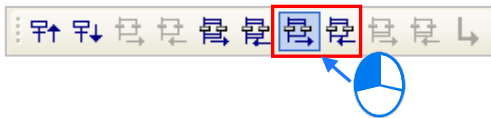
14

### 14.3.6 Step Structure - Simultaneous Divergence & Select Convergence












- (1) Select a step in the working area. The step selected can not be the topmost step in the sequential function chart.

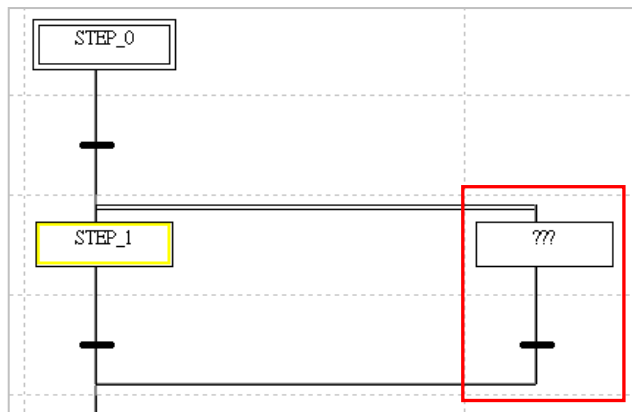


- (2) If users want to create a simultaneous divergence and a convergence of sequence selection at the right/left side of the step selected, click  on the toolbar for the right direction or  for the left direction. Or users can also right-click the step selected and click the suitable option, for the step and a transition to be added to the right or left direction.



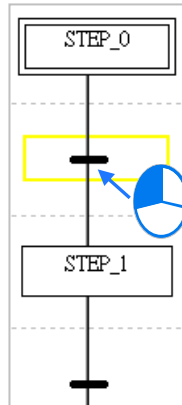
OR




	Insert a step before the cursor	Shift+F1
	Insert a step after the cursor	Shift+F2
	The transition branches to the right direction	Shift+F3
	The transition branches to the left direction	Shift+F4
	Add a simultaneous step to the right direction	Shift+F5
	Add a simultaneous step to the left direction	Shift+F6
	Add a step and a transition to the right direction	Shift+F7
	Add a step and a transition to the left direction	Shift+F8
	Add a transition and a step to the right direction	Shift+F9
	Add a transition and a step to the left direction	Shift+F10
	Jump to a step	Shift+F11














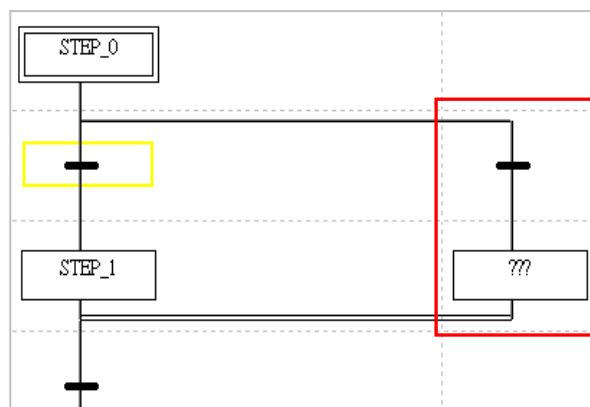
### 14.3.7 Step Structure – Select Divergence & Simultaneous Convergence

- (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/left side of the step selected, click  on the toolbar for the right direction or  for the left direction. Or users can also right-click the step selected and click the suitable option,  for the transition and a step to be added to the right or left direction.


	Insert a step before the cursor	Shift+F1
	Insert a step after the cursor	Shift+F2
	The transition branches to the right direction	Shift+F3
	The transition branches to the left direction	Shift+F4
	Add a simultaneous step to the right direction	Shift+F5
	Add a simultaneous step to the left direction	Shift+F6
	Add a step and a transition to the right direction	Shift+F7
	Add a step and a transition to the left direction	Shift+F8
	Add a transition and a step to the right direction	Shift+F9
	Add a transition and a step to the left direction	Shift+F10
	Jump to a step	Shift+F11



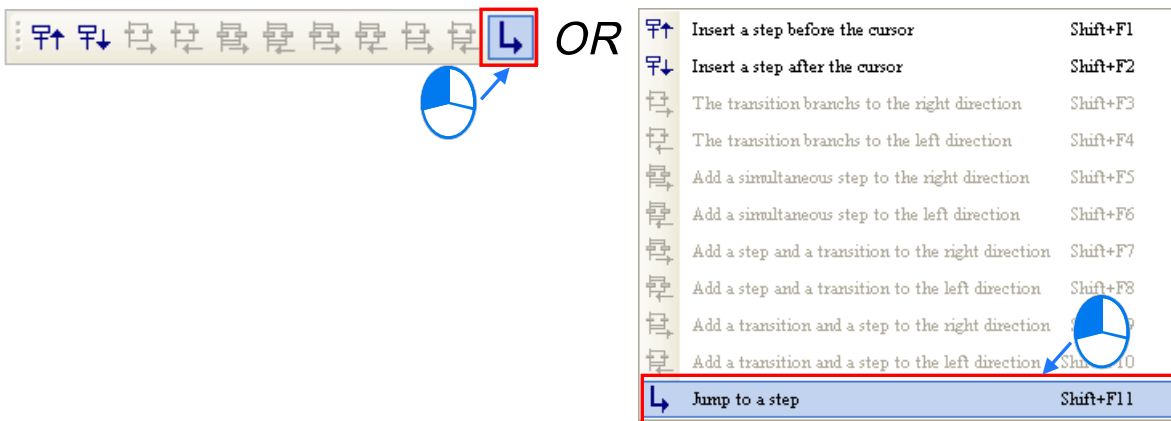
### 14.3.8 Inserting a Jump Point

- (1) Select the bottommost step or the bottommost transition on a divergent path in the working area.

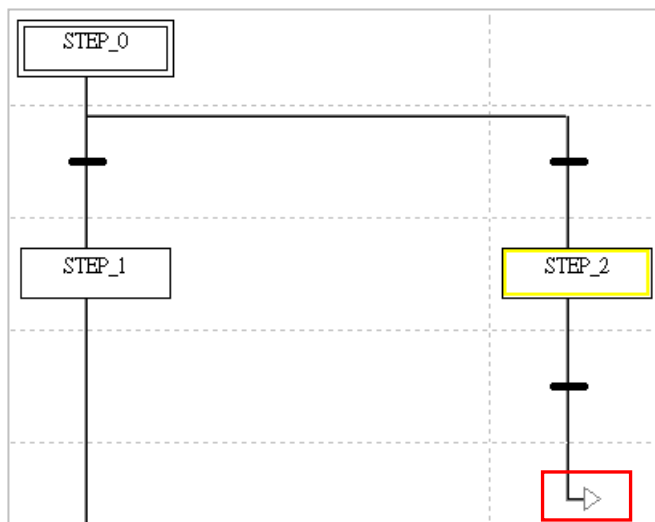


- (2) After users click  on the toolbar, a jump point will be inserted. The users can also insert a jump point by right-clicking the step or the transition selected and click **Jump to a Step**.

14

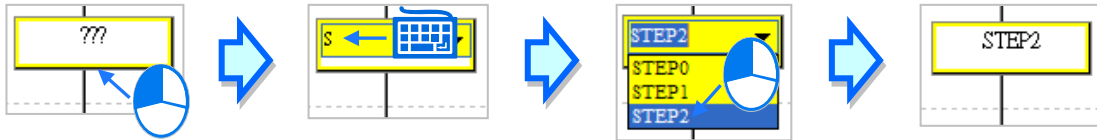


	Insert a step before the cursor	Shift+F1
	Insert a step after the cursor	Shift+F2
	The transition branches to the right direction	Shift+F3
	The transition branches to the left direction	Shift+F4
	Add a simultaneous step to the right direction	Shift+F5
	Add a simultaneous step to the left direction	Shift+F6
	Add a step and a transition to the right direction	Shift+F7
	Add a step and a transition to the left direction	Shift+F8
	Add a transition and a step to the right direction	Shift+F9
	Add a transition and a step to the left direction	Shift+F10
	Jump to a step	Shift+F11

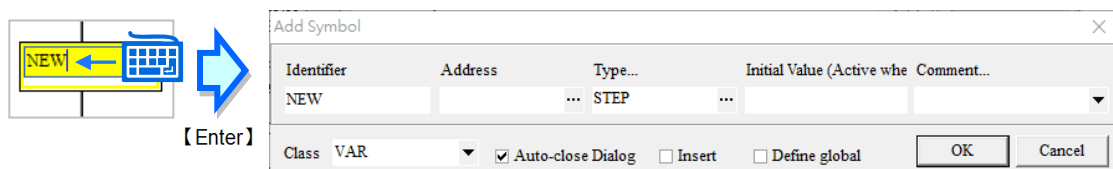


### 14.3.9 Assign Steps and the Transitions

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 can type partial step name to see a dropdown list and then click to select the step or the transition. 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 and 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.



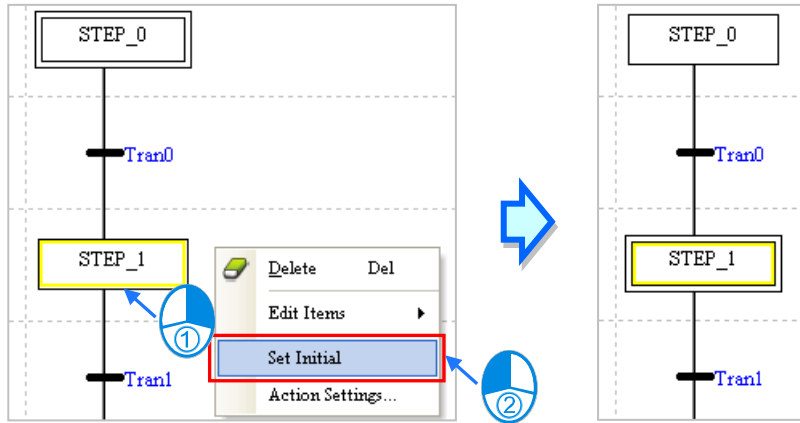
\*.Transitions can be assigned as the element **BOOL** for the structure variable. Please refer to chapter 8, for more information about structure.

\*. 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.



### 14.3.10 Specifying an Initial Step

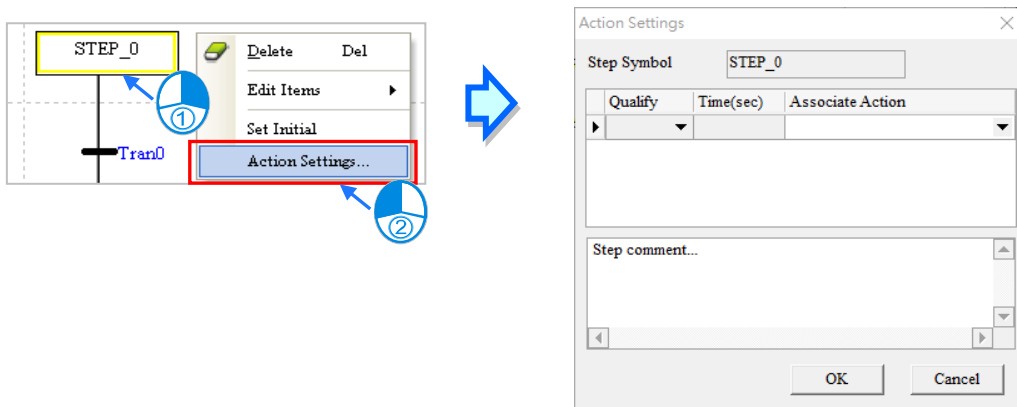
Users can define a step in a sequential function chart in an AH/AS series CPU module as the initial step in the sequential function chart whereas they can not define a step in a sequential function chart in a DVP series PLC as the initial step in the sequential function chart. If users want to define 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.



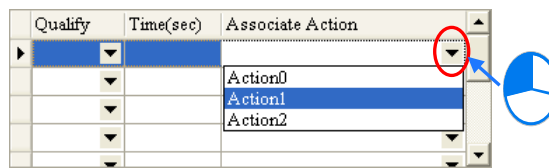
\*. Please refer to section 14.2.7 for more information about activating the steps in a DVP series PLC.

### 14.3.11 Assigning Actions / Fold the Action Table

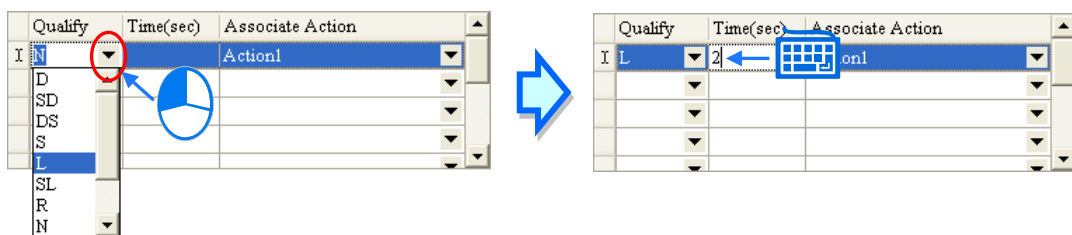
- (1) Select the step to which an action will be assigned, right-click the step, and click **Action Settings...** on the context menu to open the **Action Settings** window. If the step is not declared, the system will not allow the assignment of an action to the step.



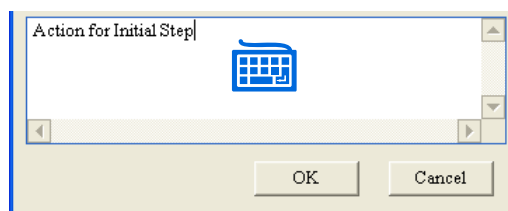
- (2) Select an action in an **Associate Action** drop-down list cell in the **Action Settings** window.



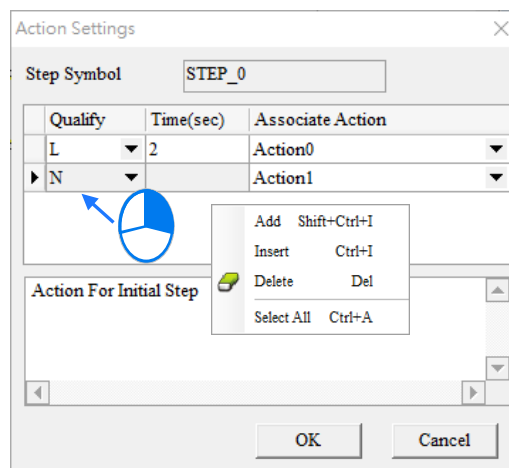
- (3) After an action is selected, users have to select a qualifier in the **Qualify** drop-down list cell for the action. If the qualifier selected is related to time, the users have to type a time interval in the **Time (sec)** cell for the qualifier. A second is a measurement unit of a time interval. 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.

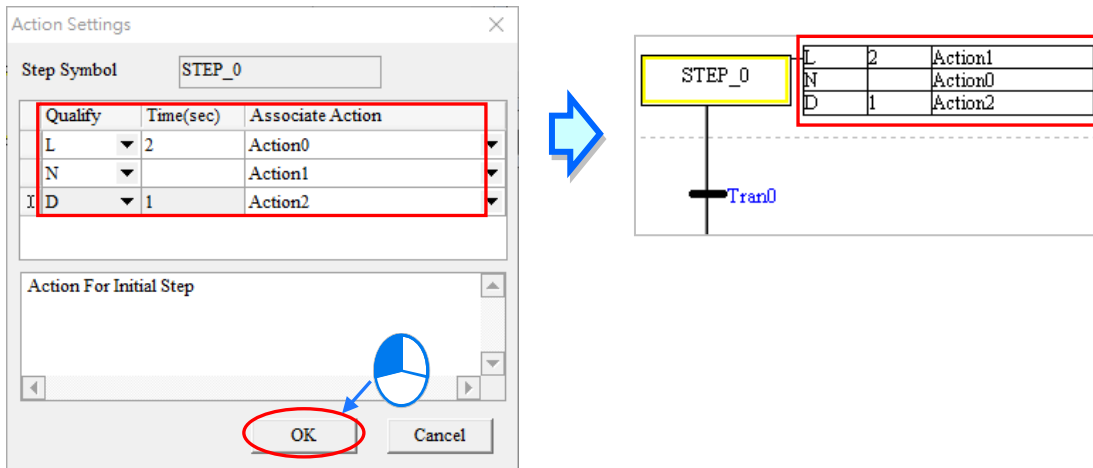


- (5) After the users right-click an item on the list, they can click **Add**, **Insert**, **Delete**, or **Select All** on the context menu. When there are more than one action that needed to be selected, users can hold the Control button on the keyboard and simultaneously use the mouse to click and select the actions. Users can also hold the Shift button on the keyboard and simultaneously use the mouse to click the first action needed and the last action needed to set the selection range.



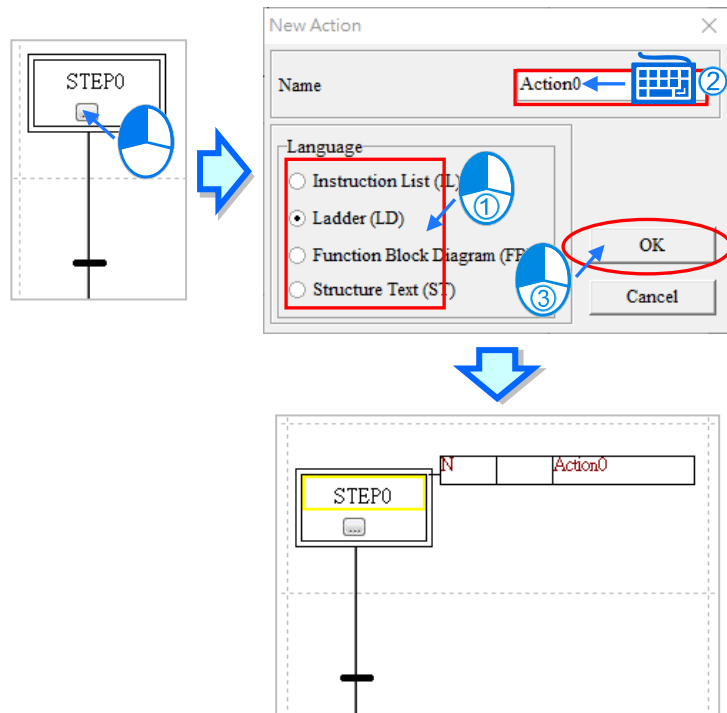
Item	Function
Add	Adding a row to the bottom of the cell selected.
Insert	Inserting a row above the cell selected
Delete	Deleting the row selected
Select All	Selecting all the items

(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.




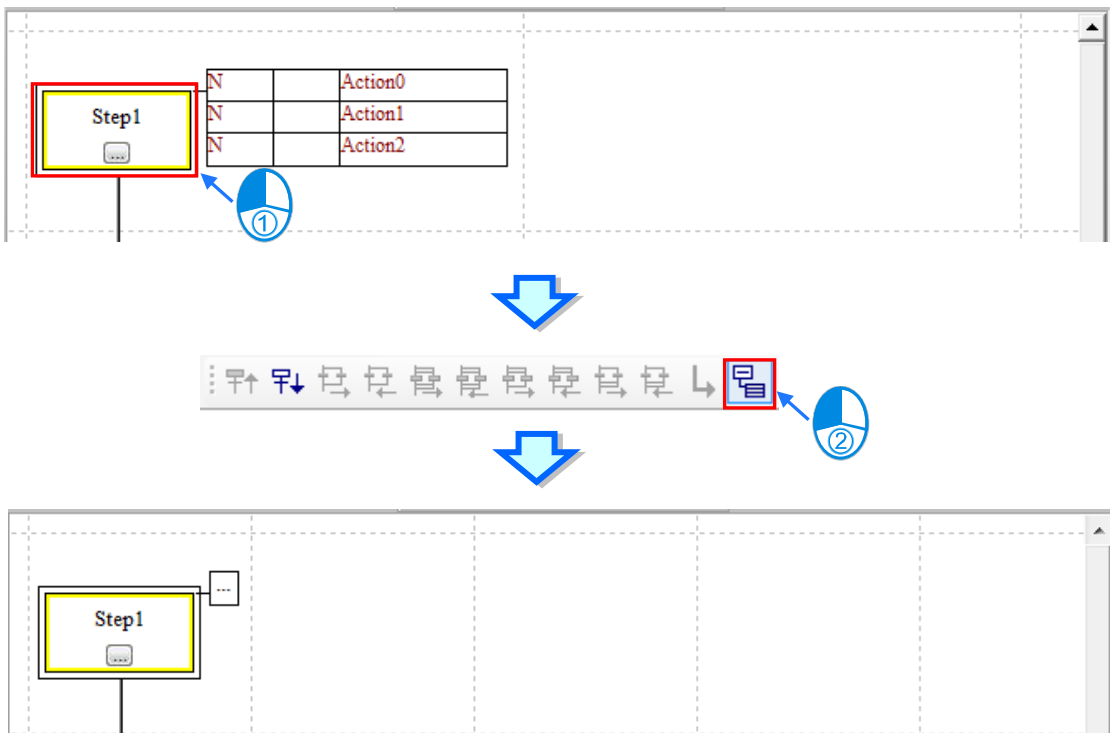
**Additional remark**

ISPSOft has a new button for quickly adding and set up actions. Users can click the button in the STEP to open a New Action window to add and set up actions. Once the creation is done, click the OK button to have the action added to the step you have selected.



14

Users can use the icon  to fold the Action table to save some space.



## 14.4 Sequential Function Chart Examples

### 14.4.1 Example Descriptions

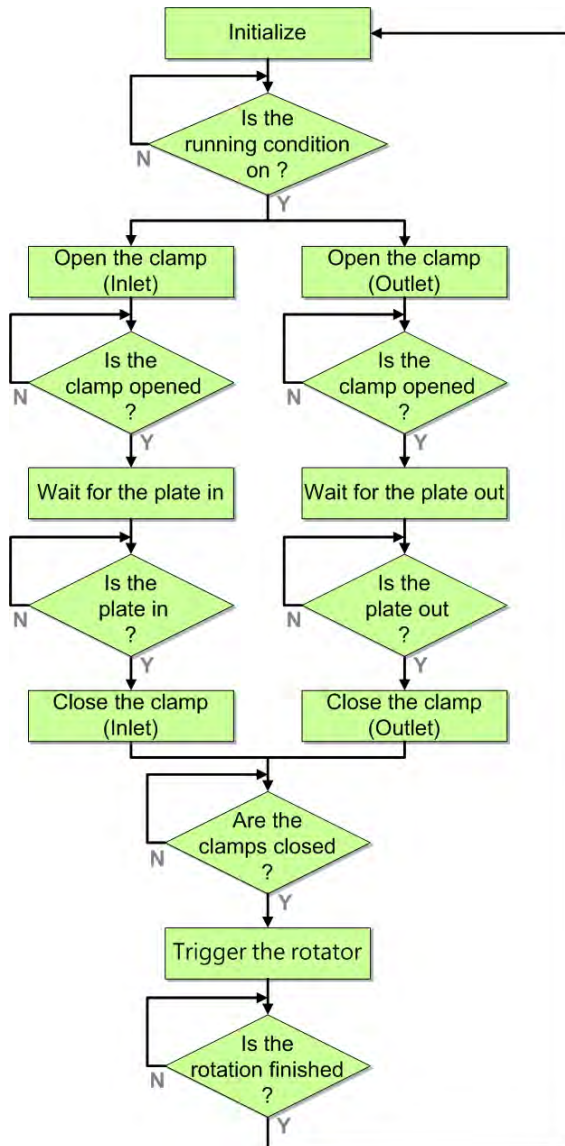
The example is about the rotation of two plates. The figure below is a flow chart. The actual equipment is also diagramed below.

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.

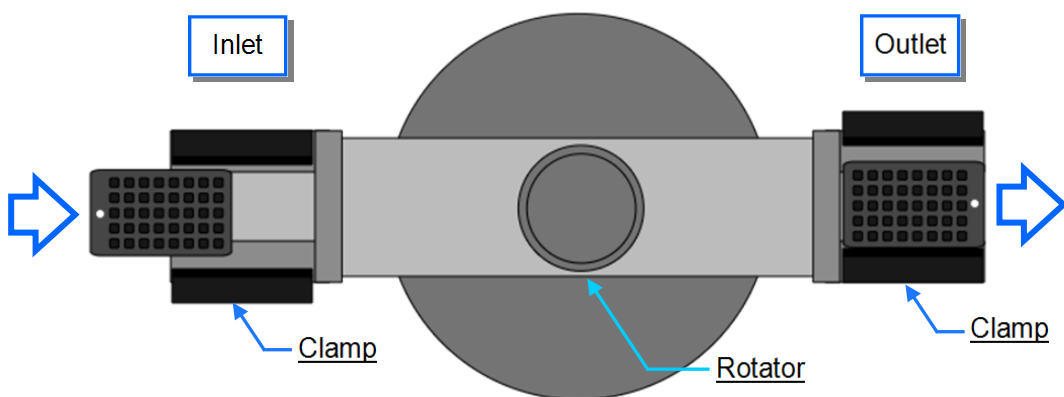
When the system begins to operate, the cylinders under the clamps open the clamps. After the clamps are opened, the system will send readiness signals. After the readiness signals are sent, a plate will enter the inlet, and the plate in the outlet will make its exit from the outlet.

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.





14



### 14.4.2 Planning Hardware

In this example, the AH5x0 series **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.

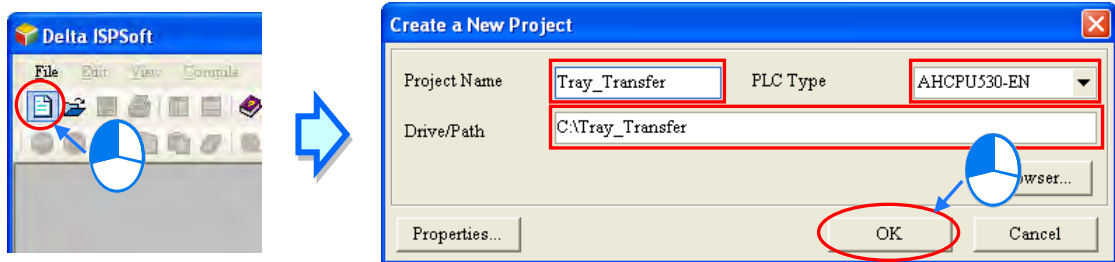
Type	ID	Description
Digital input	X0.0	Completion signal
Digital input	X0.1	Completion signal
Digital input	X0.2	Completion signal
Digital input	X0.3	(Inlet) Cylinder—Opening sensor
Digital input	X0.4	(Inlet) Cylinder—Closing sensor
Digital input	X0.5	(Outlet) Cylinder—Opening Sensor
Digital input	X0.6	(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

### 14.4.3 Planning a Program

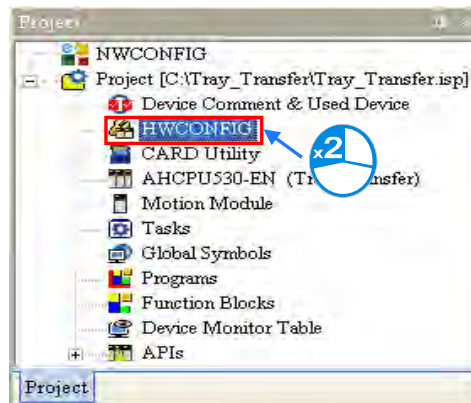
- (1) The procedure for initializing the system is to reset the trigger signal and the readiness signals to OFF until the condition for the operation of the system is met. In this example, when the operation flag is ON, and the error flag is OFF, the condition for the operation of the system is met.
- (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.

### 14.4.4 Creating a Program

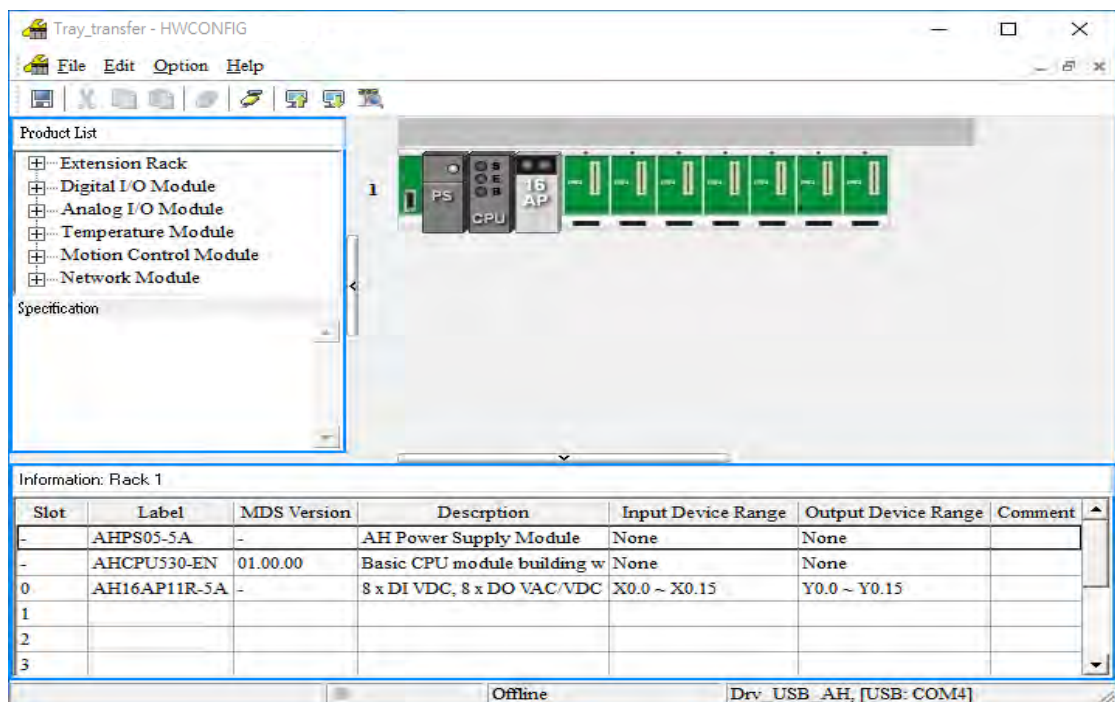
Start ISPSOft, and then create a new project.



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.

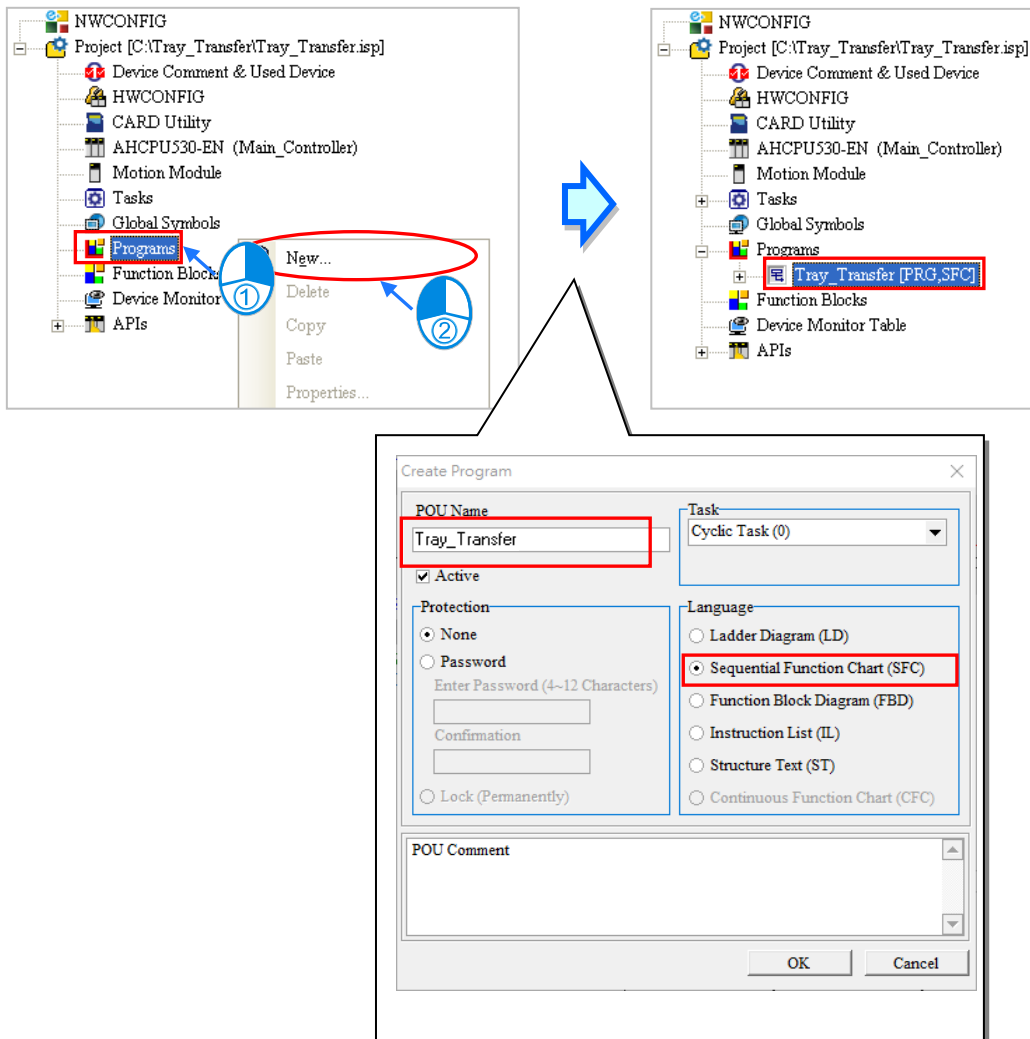


14





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.



14

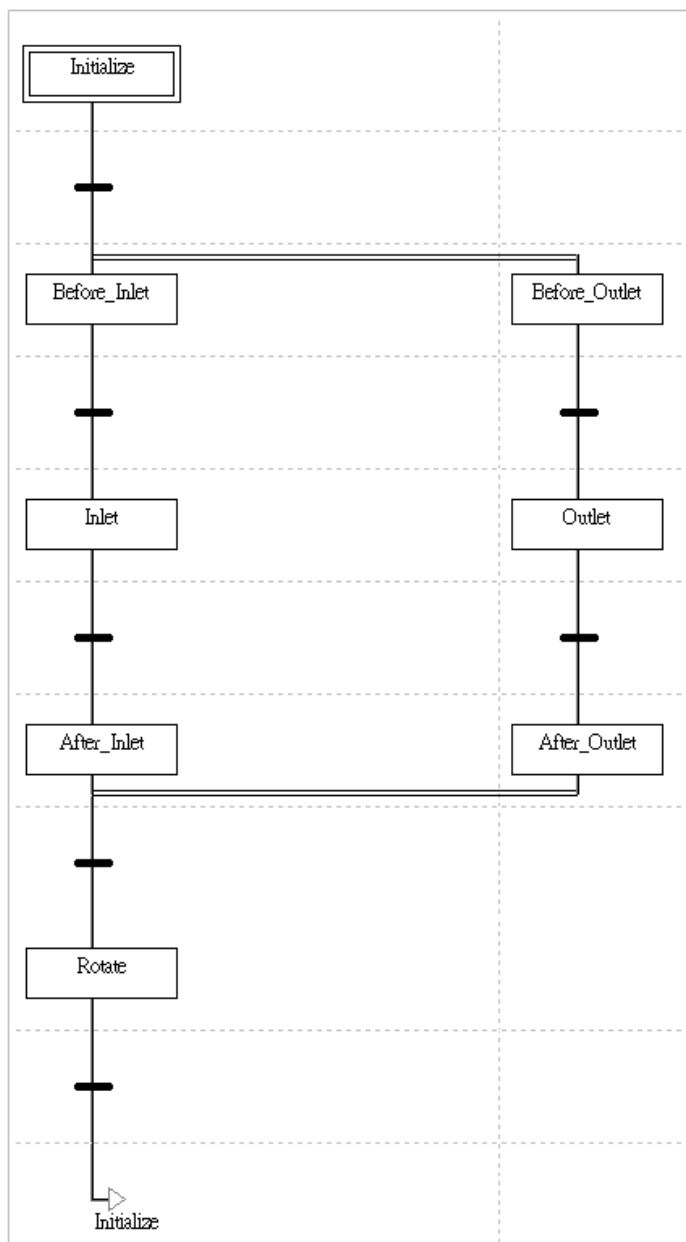
Create the flags, the digital input, and the digital outputs in the global symbol table.

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.

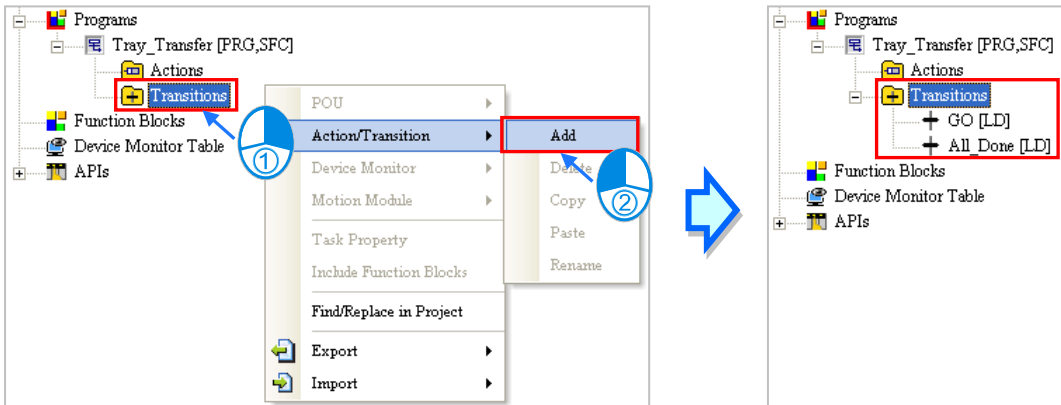
Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	Initialize	N/A [Auto]	STEP	FALSE	
VAR	Before_Inlet	N/A [Auto]	STEP	FALSE	
VAR	After_Inlet	N/A [Auto]	STEP	FALSE	
VAR	Before_Outlet	N/A [Auto]	STEP	FALSE	
VAR	After_Outlet	N/A [Auto]	STEP	FALSE	
VAR	Inlet	N/A [Auto]	STEP	FALSE	
VAR	Outlet	N/A [Auto]	STEP	FALSE	
VAR	Rotate	N/A [Auto]	STEP	FALSE	

Base on the sequential function chart shown below, use the first step by default for initial step.



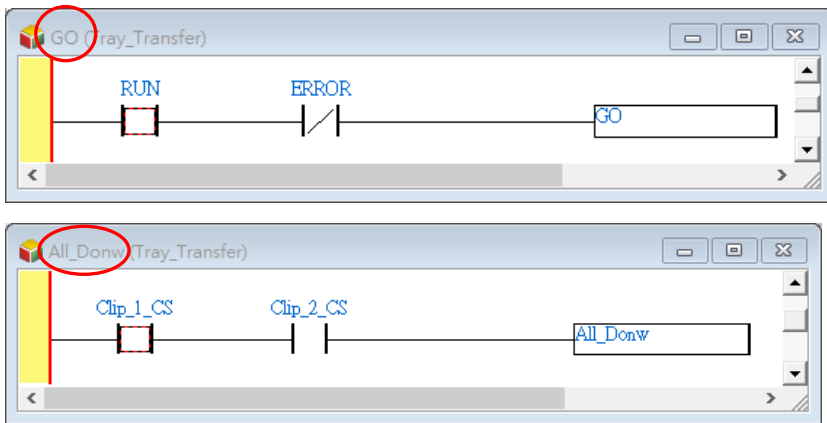
14

Create a transition called **Go** and a transition called **All\_Done** under the **Transitions** section in the project management area.

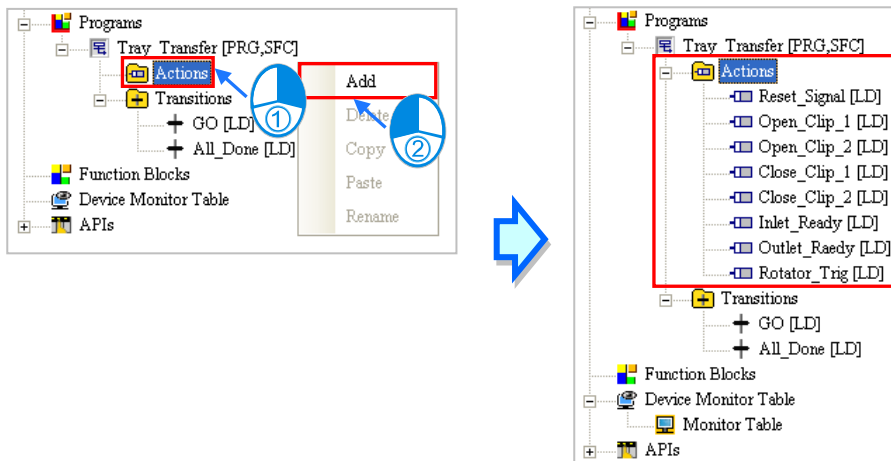


After a transition called **Go** and a transition called **All\_Done** are created under the **Transitions** section in the project management area, users have to write the programs shown below.

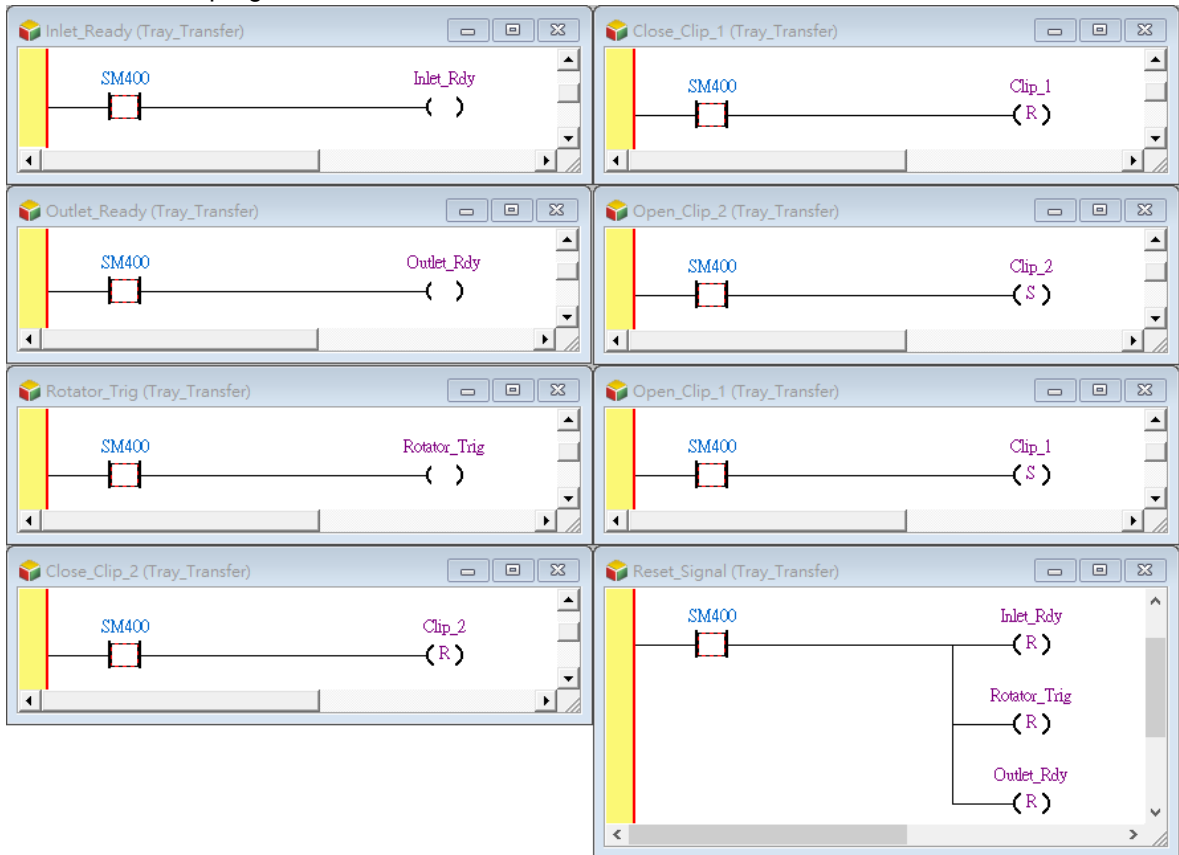
14



Create the actions which are shown below under the **Actions** section in the project management area.

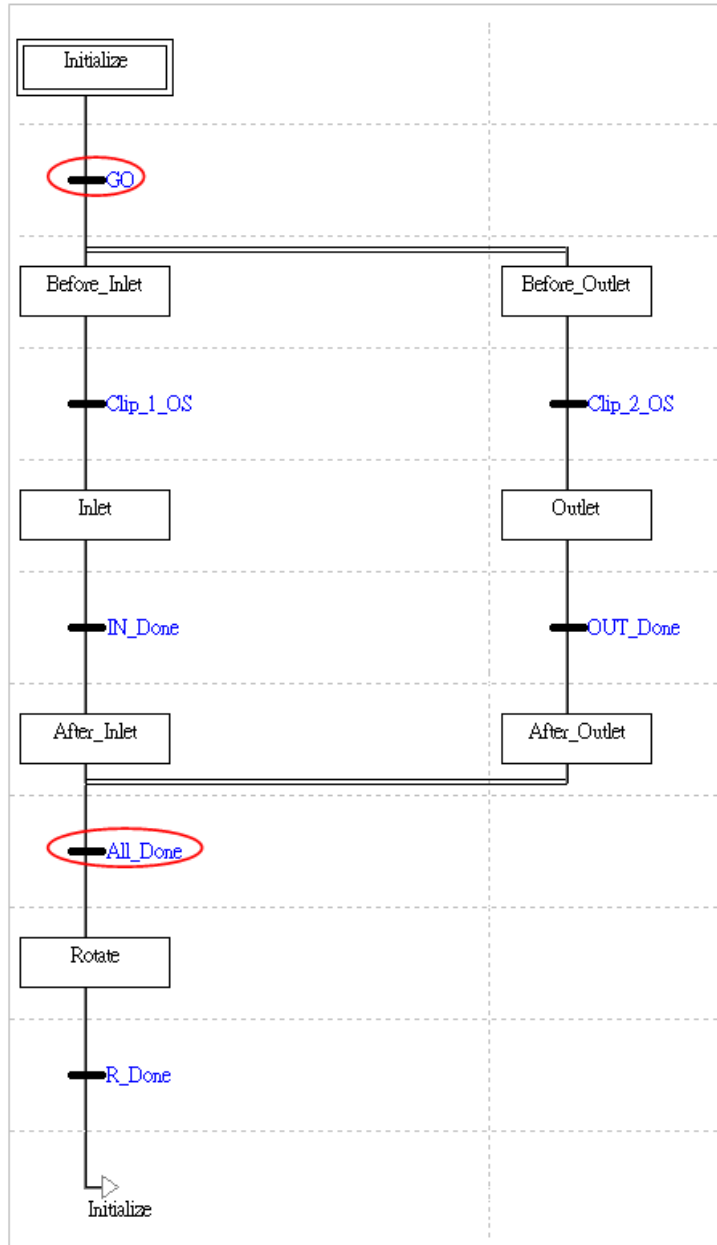


After the actions are created under the **Actions** section in the project management area, the users have to write the programs shown below.



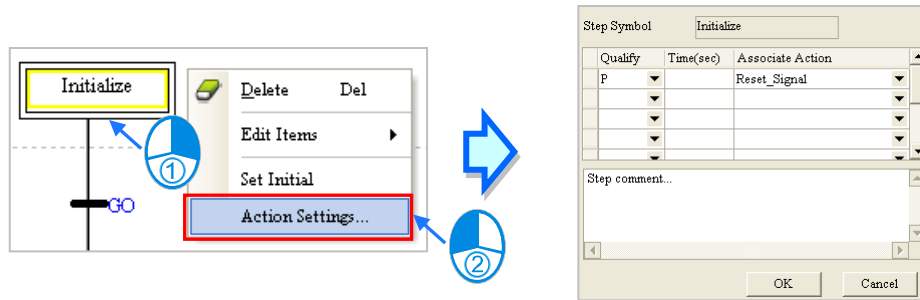
14

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**.

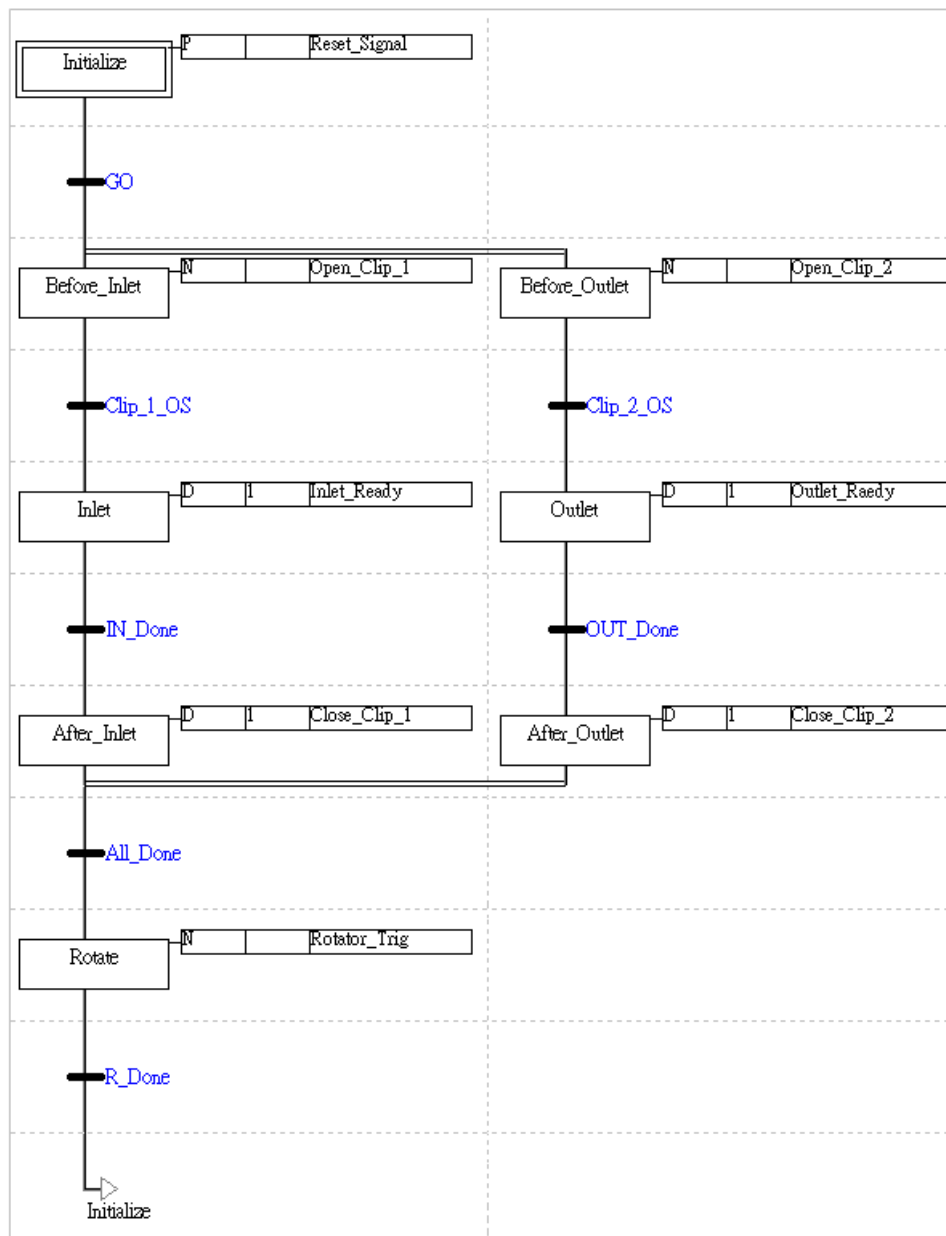


14

The users have to assign the actions to the steps in the sequential function chart, and assign qualifiers to the actions.



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.





**MEMO**





# Chapter 15 Continuous Function Charts

## Table of Contents

<b>15.1 Continuous Function Charts (CFC)</b> .....	<b>15-2</b>
15.1.1 About Continuous Function Charts .....	15-2
15.1.2 Things to Note When Using CFC .....	15-2
15.1.3 CFC Programming in ISPSoft .....	15-3
15.1.4 CFC Editing Toolbar .....	15-4
15.1.5 Shortcuts for CFC Programing .....	15-5
<b>15.2 Creating a CFC in ISPSoft</b> .....	<b>15-6</b>
15.2.1 Selecting Objects .....	15-6
15.2.2 Input/Output Nodes and Logic Gates .....	15-6
15.2.2.1 Inserting Nodes or Gates .....	15-7
15.2.2.2 Add/Delete a Pin .....	15-9
15.2.3 Changing a Pin Type .....	15-10
15.2.4 Connecting objects and Canceling Connections .....	15-11
15.2.5 Instructions and Function Blocks .....	15-11
15.2.5.1 Inserting an Instruction .....	15-13
15.2.5.2 Inserting a Function Block .....	15-15
15.2.6 Deleting objects .....	15-16
15.2.7 Editing Devices or Symbols .....	15-18
15.2.8 Activating/Inactivating an Object .....	15-19
15.2.9 Inserting a Comment .....	15-20
15.2.10 Changing the Order in Which Objects are Executed .....	15-22
15.2.11 Displaying/Hiding Information .....	15-23

## 15.1 Continuous Function Charts (CFC)

### 15.1.1 About Continuous Function Charts

Continuous Function Chart (CFC) is a PLC programming language in extension to the IEC 61131-3 standard. CFC is often used in the area of motion control. The most significant feature is its graphical programming language basing on the function block diagram language. Comparing to other PLC programming languages, CFC can provide a more logical relation between input and output and a more clear execution order of the program. The operation principle of CFC will not be covered in this chapter; this chapter will only focus on the creation and operation of CFC in ISPSOft.

### 15.1.2 Things to Note When Using CFC

- It supports keyboards but does not support the instructions of LD, LDI, LDP, and LDF. Users can type the register or the name of the variable, and the system will add input node in the back.
- Inputted data is not case-sensitive, any uppercase or lowercase character can be entered. For example, OUT, Out and out are all correct. Common positional numeral systems:

- Decimal range: 2345
- Hexadecimal range: 16#5BA0
- Floating point number: 4.123

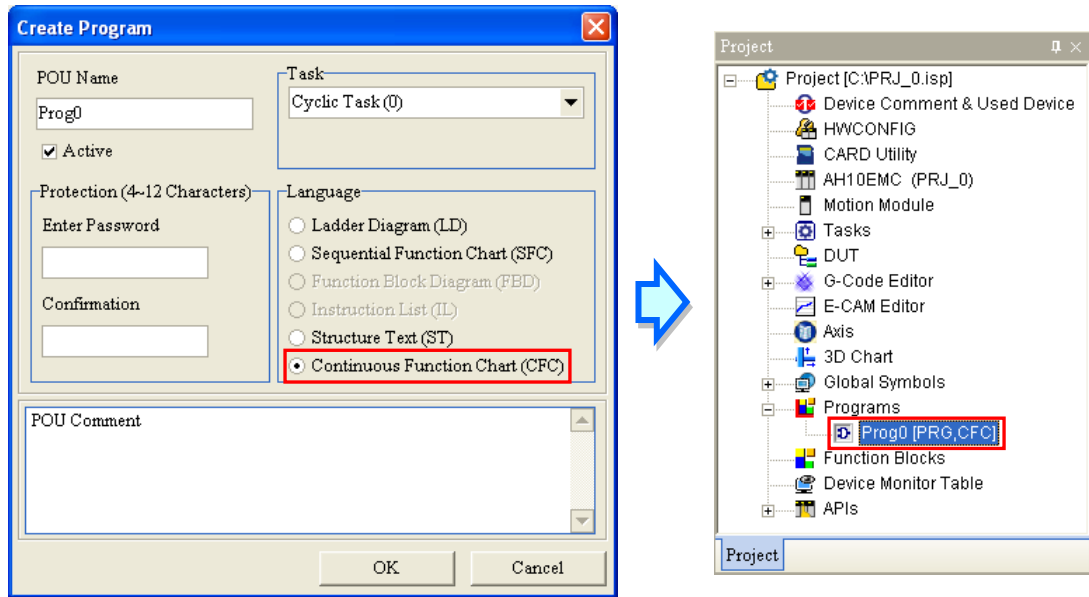
\*. The K and H notations are supported.

- There is no limit on the number of the loops for CFC to edit. But the size of a edited programs should not be larger than the memory size.

The input pin of nodes, logic gates, instructions, and function blocks can only be from one origin. But the targets of the output pin can be multiple.

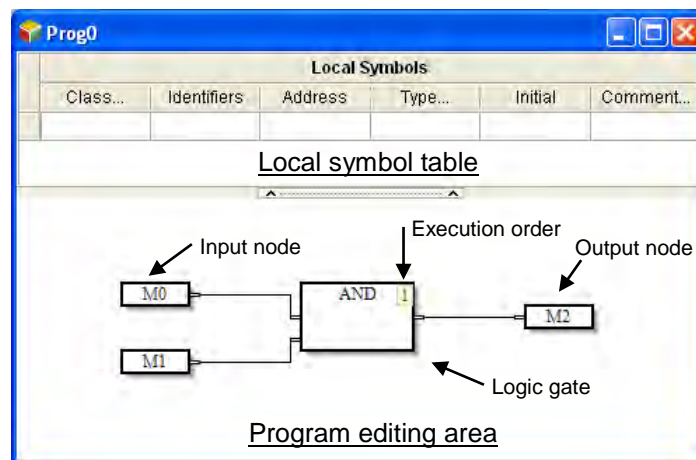
### 15.1.3 CFC Programming in ISPSoft

When creating a POU, select **Continuous Function Chart (CFC)** in the Language selection section and then users can see the created program in the project management area. Refer to section 5.4.1 or more information.



15

Below is the editing area for CFC in the ISPSoft. The upper section is for Local Symbols to be declared. And the lower section is for CFC program editing, including the input/export nodes and logic gate. The number on the upper right corner of the object is the execution order.



### 15.1.4 CFC Editing Toolbar















After a program editing window in which a CFC diagram can be created is opened, the corresponding toolbar will appear in the ISPSOft window. The functions are described below.



Icon	Keyboard Shortcut	Function
	None	Displaying/Hiding the commands on the devices
	Shift + F9	Inserting an input node
	Shift + F3	Inserting an output node
	Shift + F10	Comment
	Shift + F11	OUT gate
	Shift + F1	Inserting an AND block
	Shift + F2	Inserting an OR block
	Shift + F12	Enable/Disable function blocks or the EN pin of
	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
	None	Auto arrange All the objects will be sorted according to the data
	None	Auto resize to fit the screen to see all the objects

### 15.1.5 Shortcuts for CFC Programming

Right-click in the editing area, users will see the following shortcuts. The functions are described below.

	Undo	Ctrl+Z
	Redo	Ctrl+Y
	Cut	Ctrl+X
	Copy	Ctrl+C
	Paste	Ctrl+V
	Delete	Del
	Select All	Ctrl+A
	Comment	Shift+F10
	Execution Order	
	Activate/Inactivate	
	EN	Shift+F12
	Negate	Shift+F6
	Rising/Falling	Shift+F8
	Set/Reset	Shift+F7

Item	Description
Undo	It erases the last editing done to the document reverting it to an older state.
Redo	It reverses the undo.
Cut	Cut the selected object.
Copy	Copy the selected object.
Paste	Paste the lately cut/copied object to the editing position.
Delete	Delete the selected object.
Select All	Select all objects in the editing area.
Comment	Insert a comment at the cursor position.
Execution Order	Set up the execution order of the logic gates, instructions and function blocks. Data flow: all objects will be run according to the original set order. Topology: all the objects will be sorted and executed in an order of top-to-bottom and left-to-right.
Activate/Inactivate	Activate/inactivate the selected object
EN	Enable/Disable function blocks or the EN pin of the instructions
Negate	Inserting inverse logic
Rising/Falling	The input is rising edge-triggered/falling edge-triggered.
Set/Reset	Setting/Resetting the input

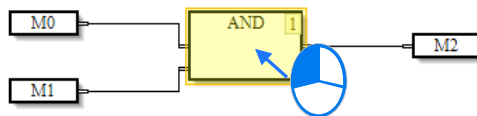
## 15.2 Creating a CFC in ISPSOft

### 15.2.1 Selecting Objects

Objects in the CFC can be selected by clicking the mouse. After selected, users can start to edit, copy, paste and so on.

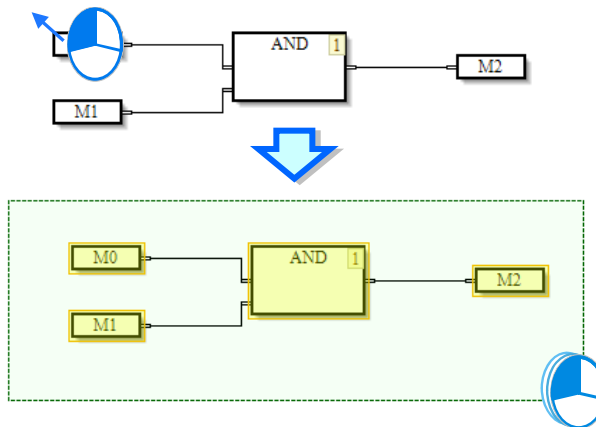
- Select an object

Move the cursor to the object you'd like to select and click the mouse or use the directional buttons on the keyboard to select objects. The selected object will be in orange; as for unselected objects, they are in white as the image shown below.



- Select multiple objects

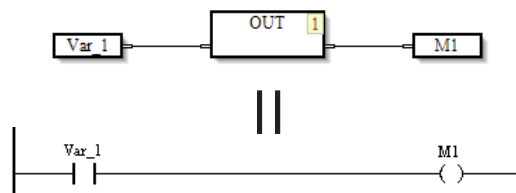
Click and drag the mouse to select the area of objects you'd like to select.



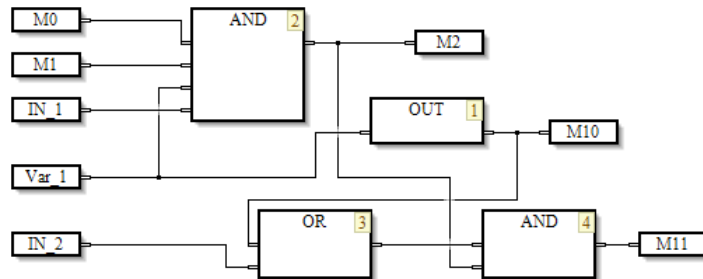
15

### 15.2.2 Input/Output Nodes and Logic Gates

Nodes and logic gates are factors to construct a CFC. Input nodes can be the variable status of the input origin or the status of the register. The output nodes are the results from the operation of logic gates. The following example illustrates the same program in 2 programming languages, the CFC and LD. Input nodes and output nodes cannot be linked directly; that is why logic gates are needed.



Use logic gates such as AND gate and OR gate to link the objects to create various complicated loops, as the following example shown.

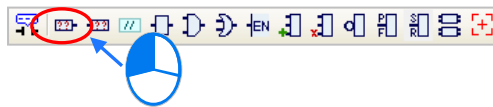


### 15.2.2.1 Inserting Nodes or Gates

There are 2 methods to insert a node, choose one for insertions.

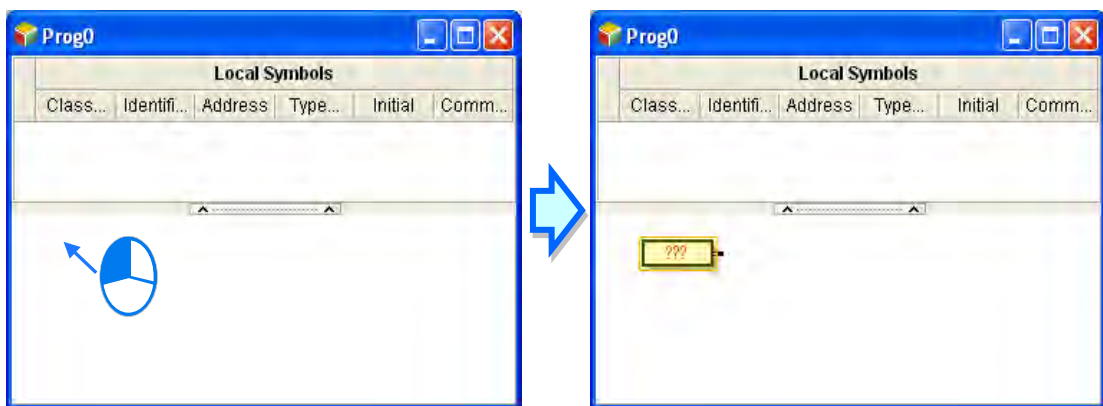
- Method I: use the icons from the toolbar

(1) Click the icon from the toolbar, as the image shown below.



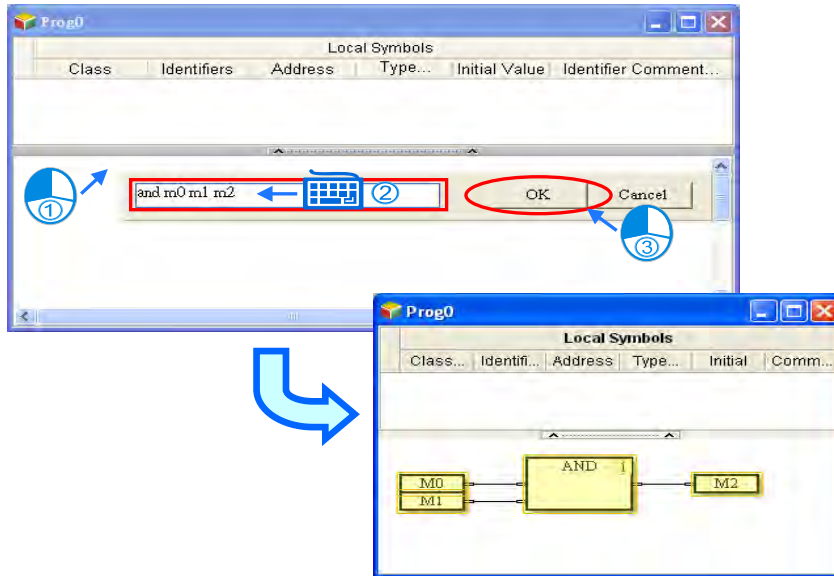
15

(2) Click the mouse on where you'd like to put the nodes/gates in the editing area, as the image shown below.



• Method II: use a keyboard

- (1) Click the mouse in the editing area and type the name of the instruction. After pressing Enter on the keyboard or click the OK button on the screen, the instruction will appear. Click where you'd like to use the instruction and then the instruction creation is done. (The instructions you type are not case-sensitive. But if the instruction name is wrong, an error message will appear.)



15

Instructions can be created via a keyboard:

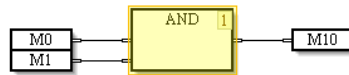
Instruction	Description
OUT	Creating an OUT gate
AND	Creating an AND gate
OR	Creating an OR gate
Name of a register	Creating an input nodes for the register
Name of a instruction	Creating an instruction block
Name of a function block	Creating a function block




### 15.2.2.2 Add/Delete a Pin

AND and Or gates can be used for adding and deleting a pin, suitable for programs with multiple conditions.

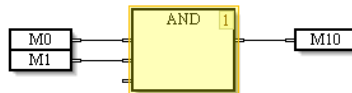
(1) Click to select the logic gate as the image shown below.




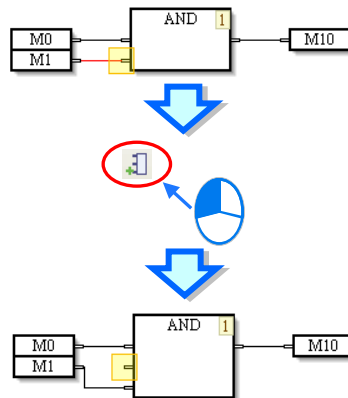
(2) Click the icon  to add an input node to the AND/OR gates as the image shown below.





(3) Then there will be a pin added to the logic gate as the image shown below.

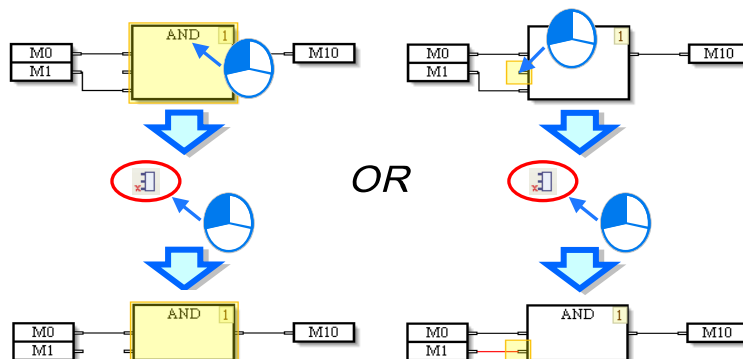


(4) If a pin needs to be added before a specific pin, click to select the pin you'd like to add a new pin before it and click the icon . A newly added pin will appear right before the pin you have selected as the image shown below.





15

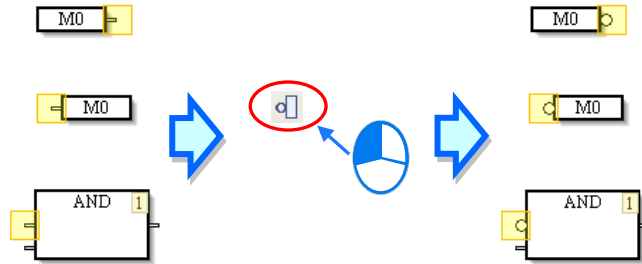
(5) Click the icon  to delete an input node from the AND/OR gates. Users can also click and select a specific pin and click the icon  to delete as the image shown below.






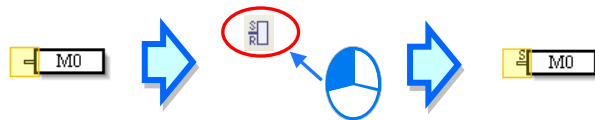
### 15.2.3 Changing a Pin Type




Users can change a pin type in a way described below, and design a complex continuous function chart.

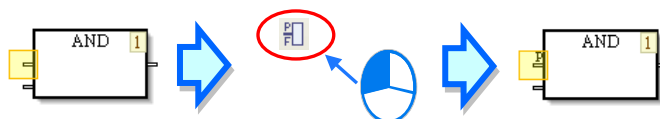
- (1) For negating, users can select the pin of a node, a pin of a logic gate, a pin of a function block, or a pin of an instruction, and then click  on the toolbar or on the context menu which will appear after the users right-click the pin, the state of the pin will be negated. If users select a pin whose state is negated, and then click , the state of the pin will not be negated.



- (2) For setting a pin, users can select the pin of an output node, and then click  on the toolbar or on the context menu which will appear after the users right-click the pin, the pin will be set. For resetting a pin, if users select a pin which is set, and then click , the pin will be reset. If users select a pin which is reset, and then click , the pin will not be reset.



- (3) For rising, users can select a pin of a logic gate, a pin of a function block, or a pin of an instruction, and then click  on the toolbar or on the context menu which will appear after the users right-click the pin, the pin will be rising-edge triggered. For falling, when users select a pin which is rising edge-triggered, and then click , the pin will be falling edge-triggered. If the users select a pin which is falling edge-triggered, and then click , the pin will not be falling edge-triggered.



15

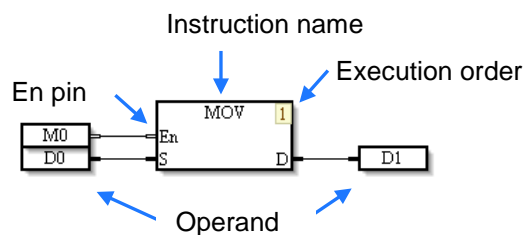
### 15.2.4 Connecting objects and Canceling Connections

When users click a pin of an object, and hold down the left mouse button, the target pins to which the pin can be connected are indicated by gray. When the users drag the mouse cursor to a target pin of an object, an arrow appears. After users release the left mouse button, the two pins will be connected. If users want to change the connection or cancel the connection, they can drag the line to a new target pin or to the blank.



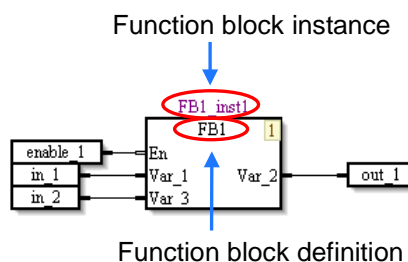
### 15.2.5 Instructions and Function Blocks

An instruction name and operands are shown in a block. The orders in which the instructions in a continuous function chart are executed are indicated by the numbers in the right corners of the blocks representing the instructions. An instruction is executed only when the logic state connected to the En pin of the block representing the instruction is ON.

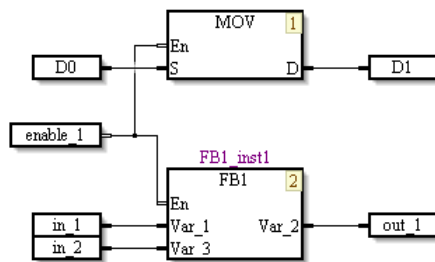


15

A function block in a continuous function chart is shown below. It is executed only when the logic state connected to the En pin is ON. Besides, the text on the top of the function block is the name given to the function block, i.e. a function block instance. Please refer to chapter 7 for more information about function blocks.

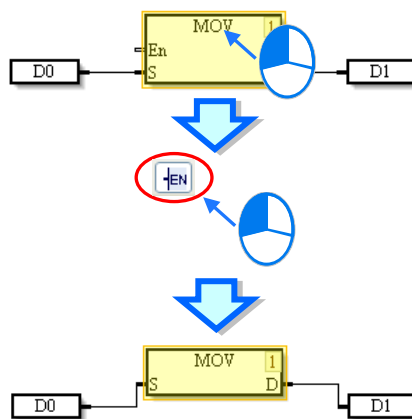


If users need to enable an instruction and a function block by a condition, they can connect the input node of the condition to the En pin of the block representing the instruction and the En pin of the function block. When the input node is ON, the instruction and the function block is executed in order. From the image shown below, enable\_1 controls the instruction and the function block.



If an instruction or a function block does not need to be controlled by a condition, users can select the block representing the instruction or the function block, and then click **EN** on the toolbar or on the context menu which will appear after the users right-click the block representing the instruction or the function block. After the En pin of a block representing the instruction or the En pin of a function block is disabled, the instruction or the function block will be executed if it is the instruction's turn or the function block's turn to be executed. If users select an instruction or a function block whose En pin is disabled, and then click **EN**, the En pin of the instruction or the En pin of the function block will be enabled.

15

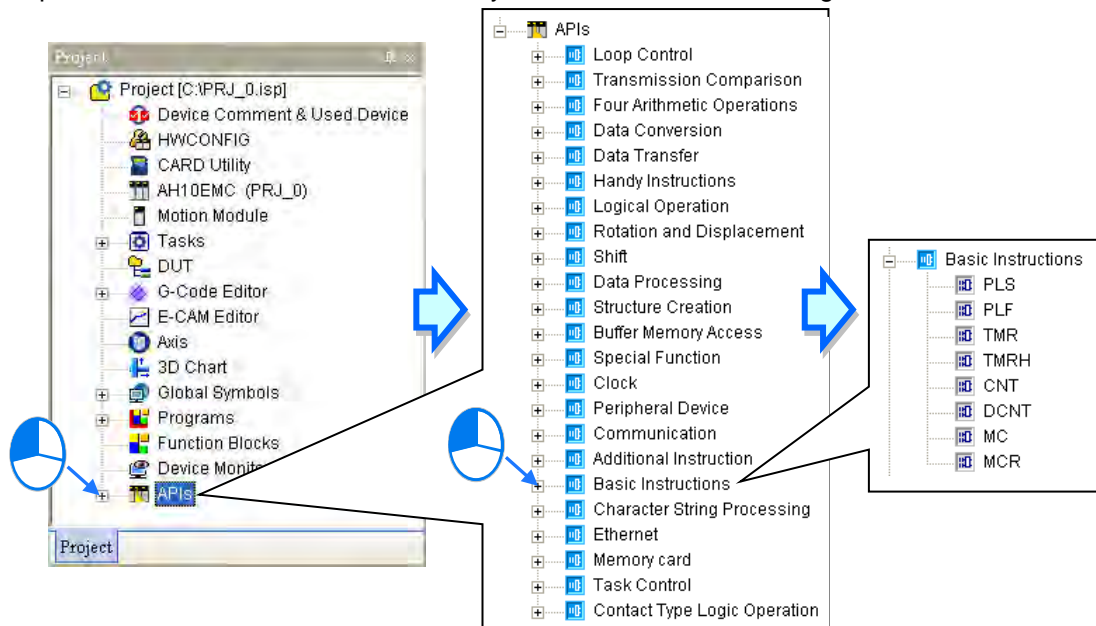


### 15.2.5.1 Inserting an Instruction

There are two methods to insert an instruction.

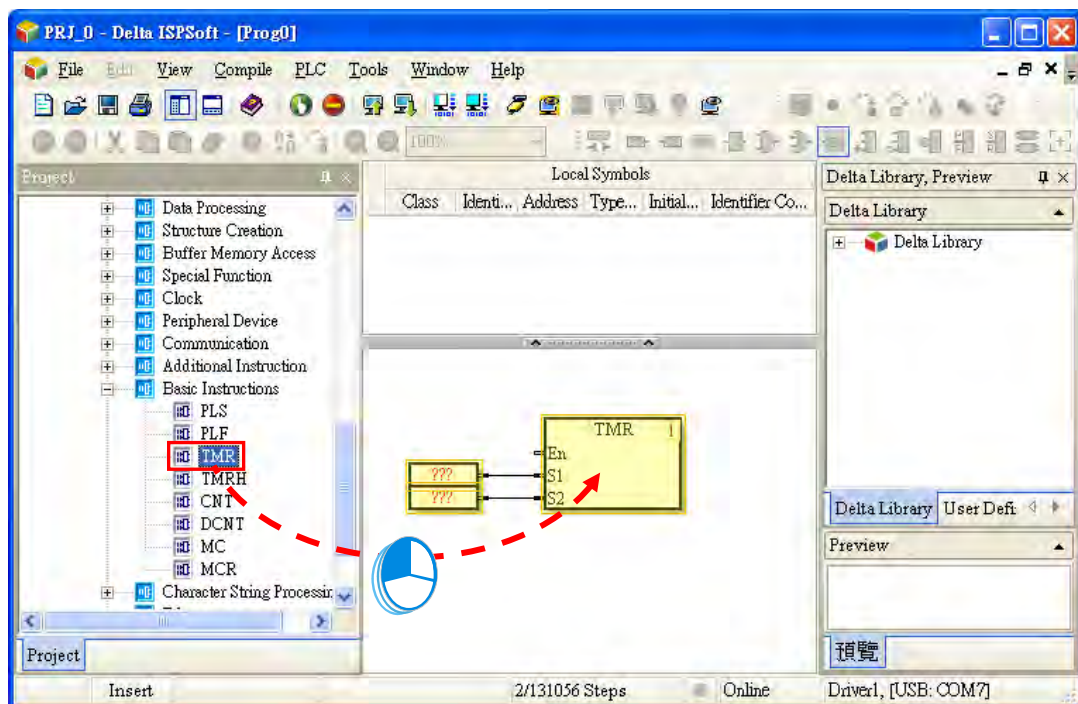
- Method 1: Drag the instruction

(1) Open the API to seek for the instruction you'd like to use, as the image shown below.



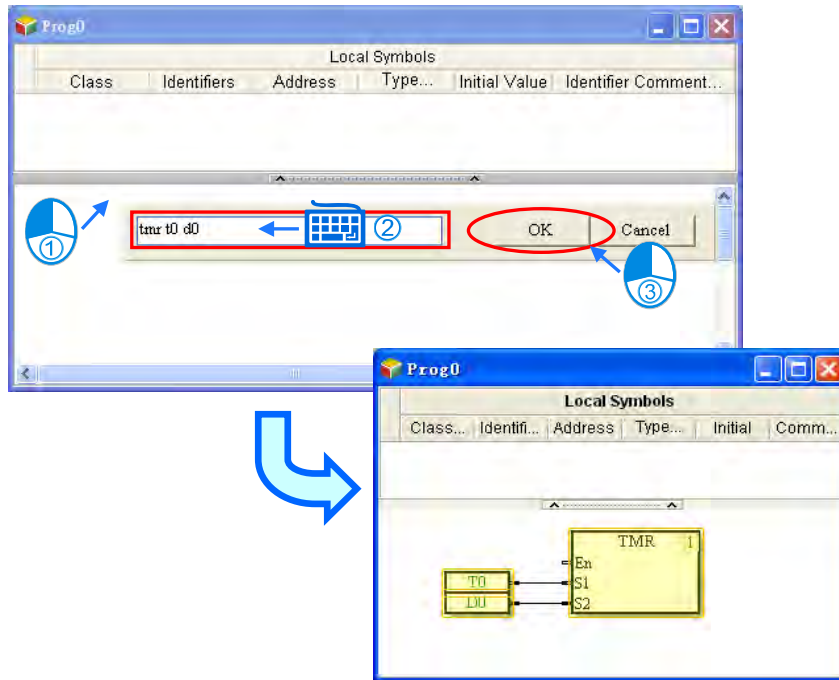
15

(2) Click on the instruction you'd like to add and drag it to where you'd like to add in the editing area. And then the instruction is added.



- Method 2: Keyboard

Type an instruction in the **Instruction** box, and press Enter on the keyboard, or click **OK**. If users move the mouse cursor and click where you'd like to add the instruction in the program editing area, the instruction will be inserted. (The instruction that users type in is not case-insensitive. If the users type an incorrect applied instruction, or an incorrect device name, an error message will appear after they click **OK**.), as the image shown below.

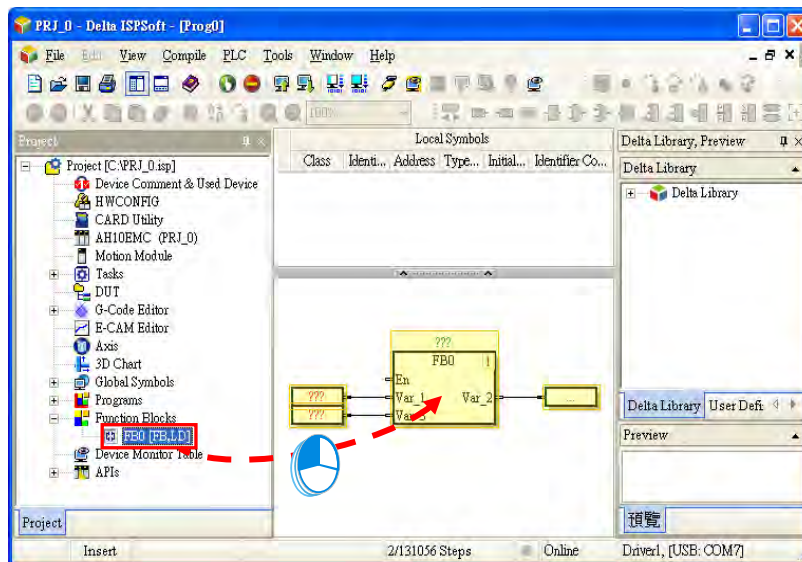


### 15.2.5.2 Inserting a Function Block

There are two methods of inserting a function block.

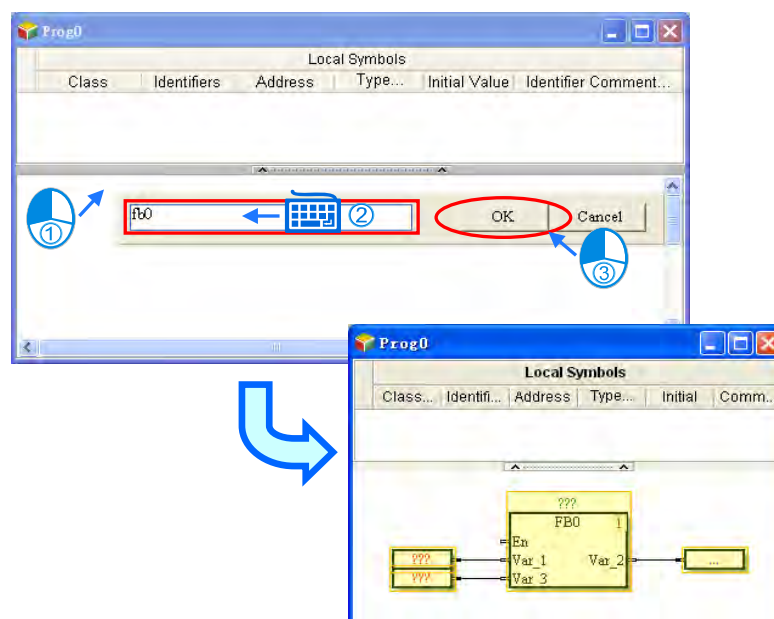
- Method 1: Dragging a function block definition

Click on the function block definition you'd like to add and drag it to where you'd like to add in the editing area. And then the function block is added.



- Method 2: Keyboard

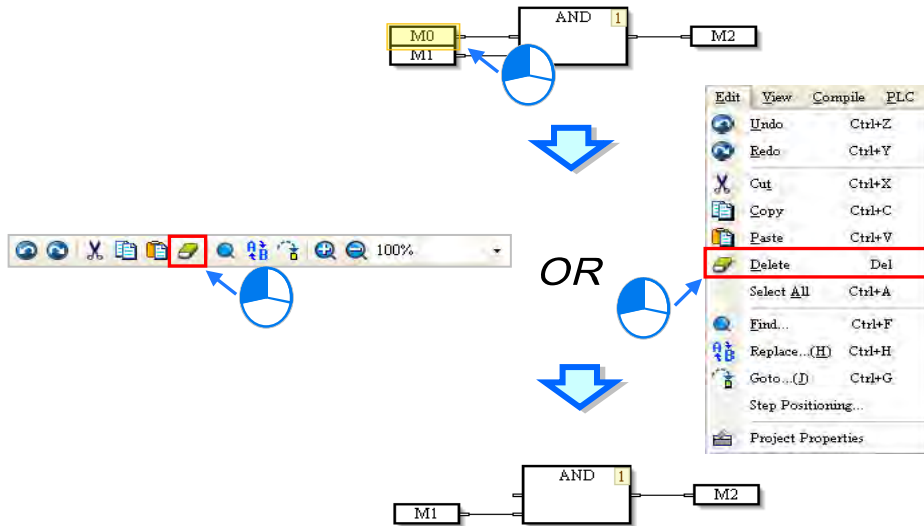
Type an instruction in the **function block definition** box, and press Enter on the keyboard, or click **OK**. If users move the mouse cursor and click where you'd like to add the function block, the function block will be inserted. (The function block that users type in is not case-insensitive. If the users type an incorrect function block, or an incorrect device name, an error message will appear after they click **OK**.), as the image shown below.



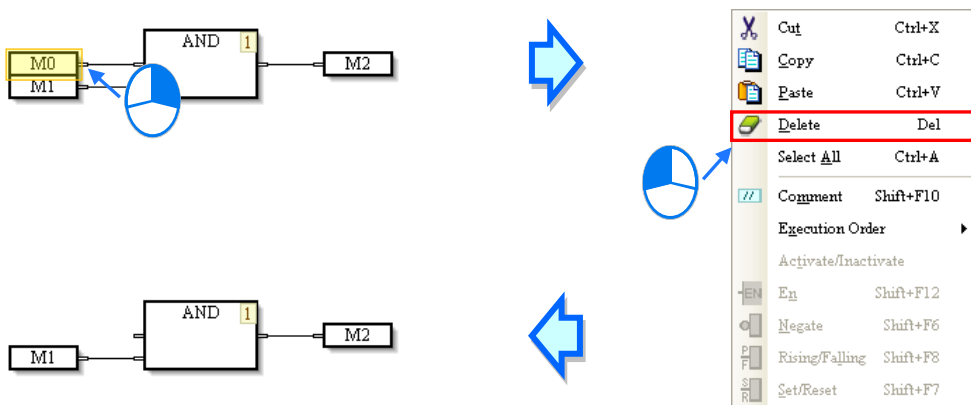
### 15.2.6 Deleting objects

If users want to delete a single object or several objects in a continuous function chart, they can use one of the methods below.

- (1) The users have to click an object, and click **Delete** on the **Edit** menu, or  on the standard toolbar, as the image shown below.

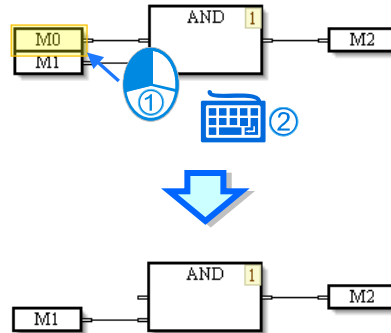


- (2) The users have to click to select the object and right-click on the object, and then click **Delete** on the context menu which appears



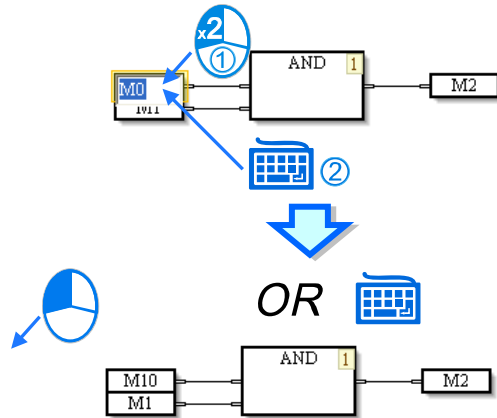


(3) The users have to click an object, and press Delete on the keyboard, as the image shown below.



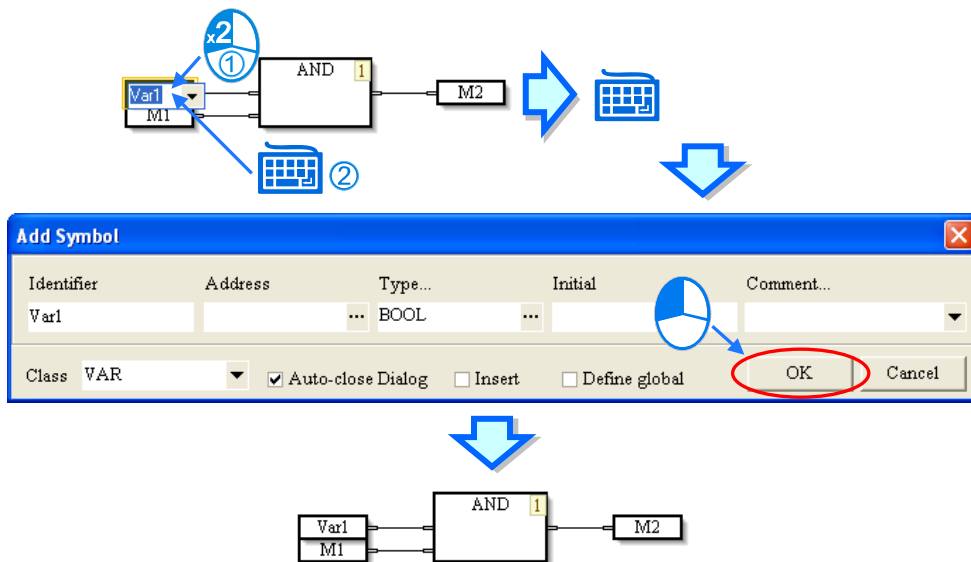
### 15.2.7 Editing Devices or Symbols

If users want to edit a node, they have to double-click the node, type a device name, and click the blank in the program editing area or press Enter on the keyboard.



If the users type a symbol in a node, the **Add Symbol** window will appear after they press Enter on the keyboard. After the users set a symbol and click **OK** in the **Add symbol** window, they will finish declaring the symbol, as the image shown below. Please refer to chapter 6 for more information about symbols.

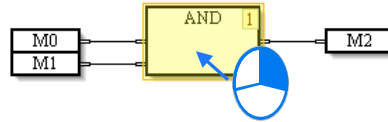
15



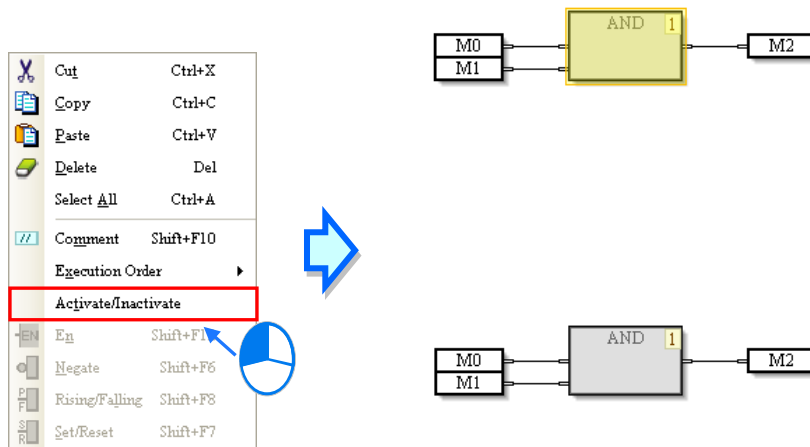
### 15.2.8 Activating/Inactivating an Object

If an object in a continuous function chart is inactivated, the compiling of the continuous function chart will skip the object. Users can temporarily inactivate some parts of a program.

(1) The users have to right-click on an object, as the image shown below.



(2) After the users click **Activate/Inactivate** on the context menu which appears, the object will be in a dark color, as the image shown below.



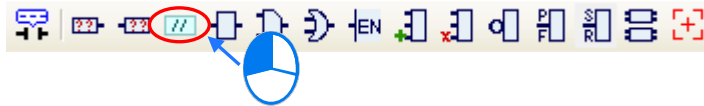
(3) If the users want to activate the object, they have to right-click the object, and click **Activate/Inactivate** on the context menu which appears.

### 15.2.9 Inserting a Comment

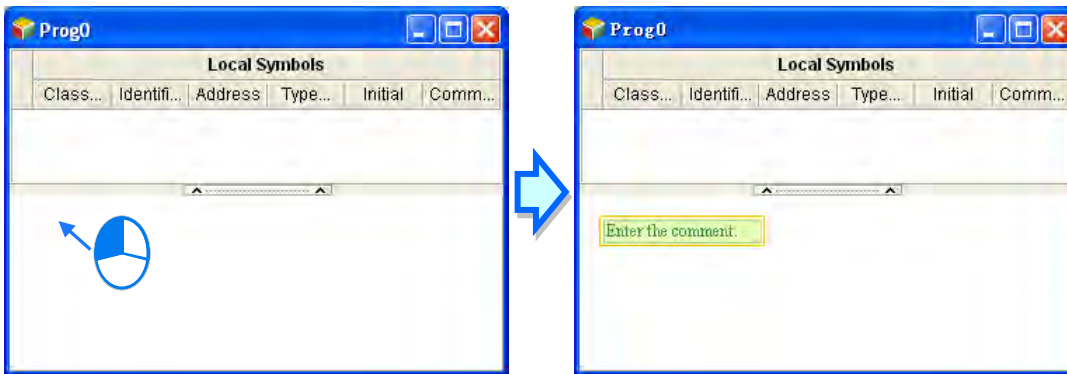
Users can insert comments in a continuous function chart. There are two methods of inserting a comment.

- Method 1: Inserting a comment in any position

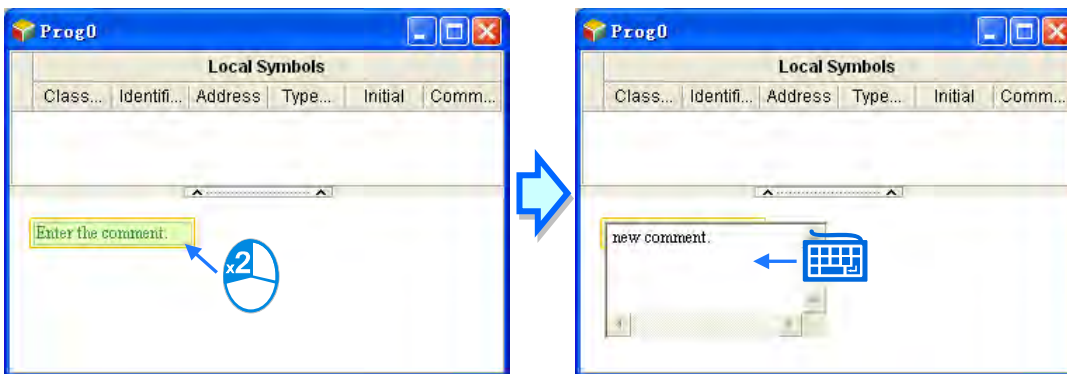
(1) The users have to click **Comment** on the toolbar, as the image shown below.



(2) After the users click a position in the program editing area, an object will be inserted, as the image shown below.

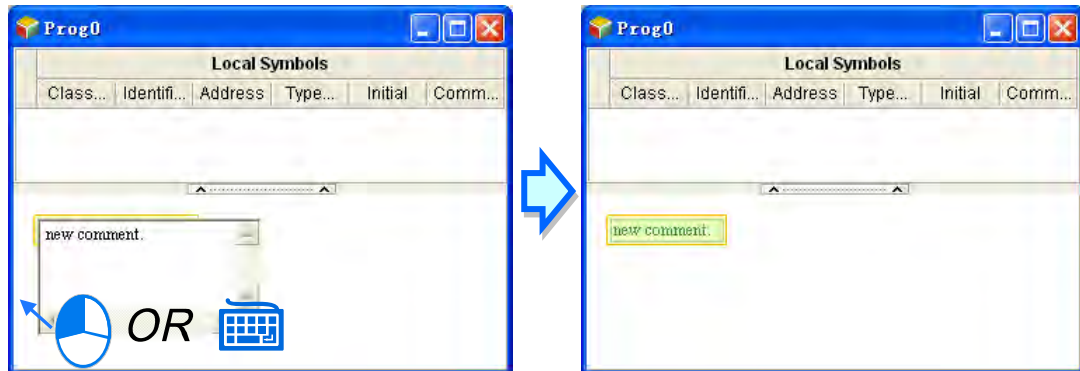


(3) After the users double-click the object inserted, a box will appear. The users can type a comment in the box. If the users want to start a new line of text at a specific point, they can press Shift+Enter on the keyboard.



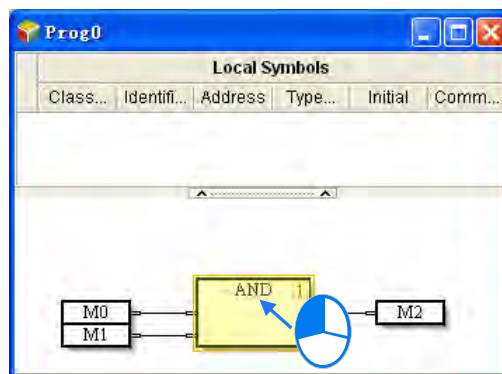
15

- (4) After the users type a comment, they have to click the blank in the program editing window, or press Enter on the keyboard. This type of comment is independent of the other types of objects in the program editing area. It does not vary with the deletion or the movement of another type of object.



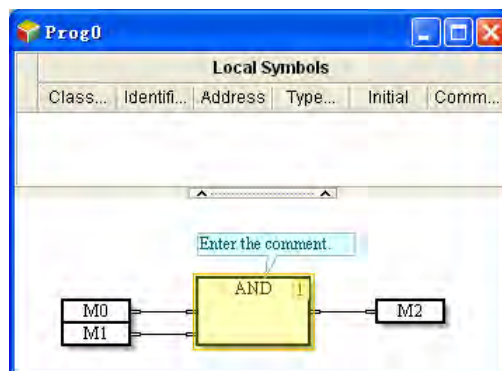
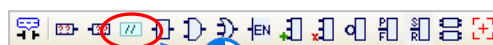
- Method 2: Inserting a comment binding with an object

- (1) The users have to click the object in which a comment will be inserted, as the image shown below.

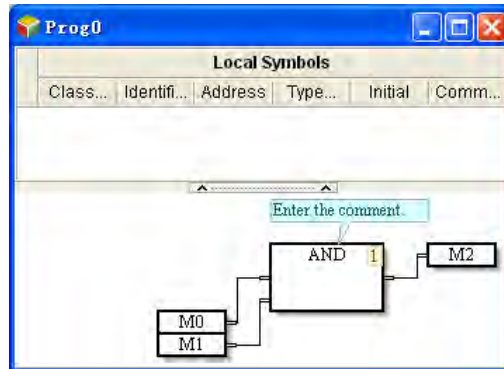


15

- (2) After the users click **Comment** on the toolbar, the comment binding with the object will appear. The users can edit the comment.

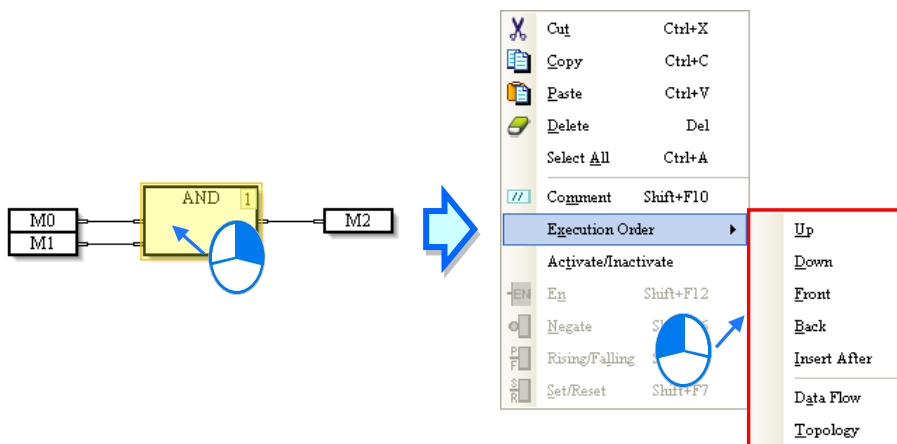


- (3) The position of the comment varies with the movement of the object. When the users move the object, the comment moves, as the image shown below.




### 15.2.10 Changing the Order in Which Objects are Executed

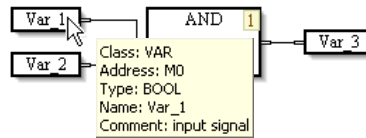
The numbers in the upper right corners of the objects in a continuous function chart indicate the order in which the objects are executed. If users want to change the number in the upper right corner of an object, they can right-click the object, point to **Execution Order** on the context menu which appears, and click **Up**, **Down**, **Front**, **Back**, **Insert After**, **Data Flow**, or **Topology**. If **Data Flow** is clicked, the order in which the objects in a continuous function chart are executed will be determined by the data flow of the objects. If **Topology** is clicked, the order in which the objects in a continuous function chart are executed will be determined by the relative positions of the objects.



15

### 15.2.11 Displaying/Hiding Information

If users move the mouse cursor to a device or a symbol after they click  on the toolbar, the information about the device or the symbol will appear. The information related to a device or a symbol includes a device address and a comment



**MEMO**

15



# Chapter 16 C the Programming Language

## Table of Contents

<b>16.1 C Programming Language (C)</b> .....	<b>16-2</b>
16.1.1 C Structures .....	16-2
16.1.2 C Environment.....	16-3
16.1.3 C Compiling and Debugging .....	16-9
16.1.4 Internal Function Library Instructions .....	16-10
16.1.5 Bookmark and Comment.....	16-11
16.1.6 Operation Example.....	16-13

## 16.1 C Programming Language (C)

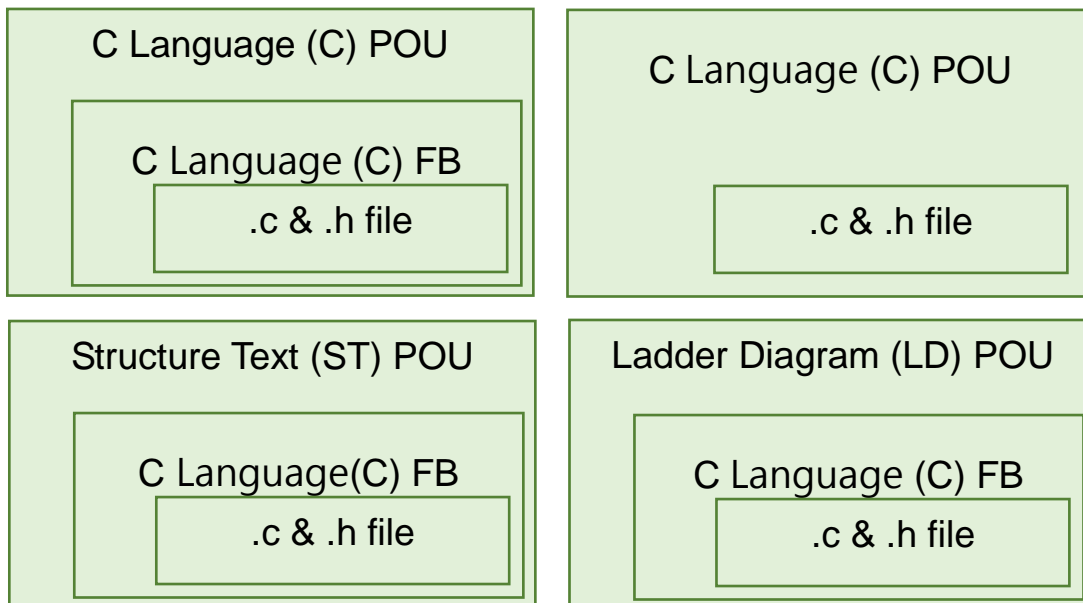
One characteristic of the C language is that the entire C program consists of statements. Compared with instruction lists (IL), C language has a higher readability. Complex expressions can be expressed in the syntax like mathematical equations. Therefore, C is the programming language more suitable for the programming applications where complex operations are required.

\* Before downloading a C program to the PLC, do confirm that the version of the AS series PLC firmware is V1.06.00 or later.

### 16.1.1 C Structures

The programming language C is a fit for the programmers who have already been familiarized with the programming in C language. Please refer to the following programming language structures supported by C language before starting the C programming.

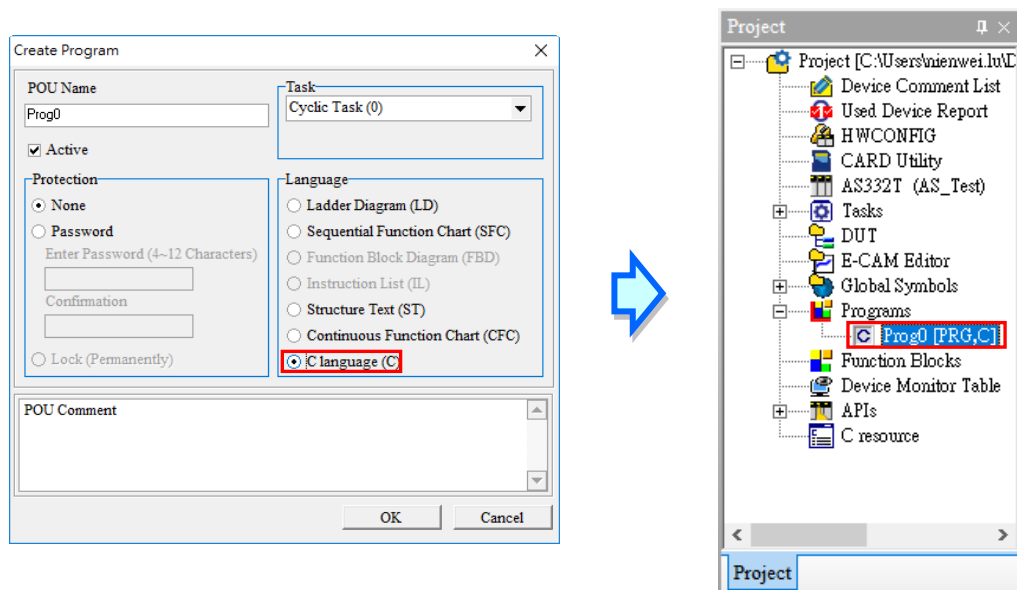
In the first place, use one of the most common editors on the market, such as DEV-C, TurboC and Visual Studio for writing, compiling and debugging. Then import the generated ".c" and ".h" files through the **Import** function of ISPSOft and use them in the C Language (C) FB or (C) POU. (C) FB can be cited in Structure Text (ST) POU and Ladder Diagram (LD) POU as shown below.



### 16.1.2 C Environment

In AS series PLC, the storage space for C execution codes (ROM) is 64K Bytes, and the available space for variable declaration (RAM) is 8K Bytes. The two storage spaces mentioned above and the space for storing the existing PLC programs are independent with each other. We suggest you pay attention to the remaining space in the **Compile Message** area once compiling is completed.

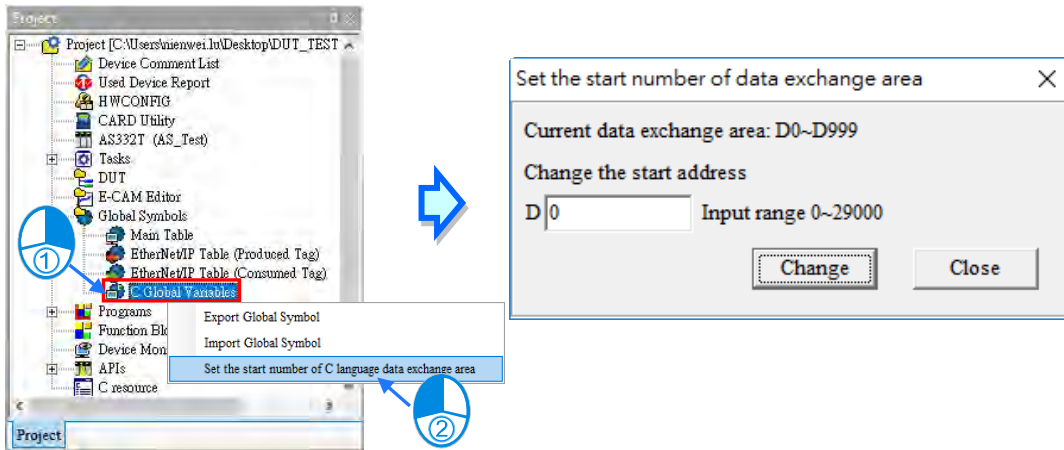
C programs are written in the C editing workplace of ISPSOft. A C program can be built just by choosing the **C language (C)** option in the field of **Language** during the creation of a POU.



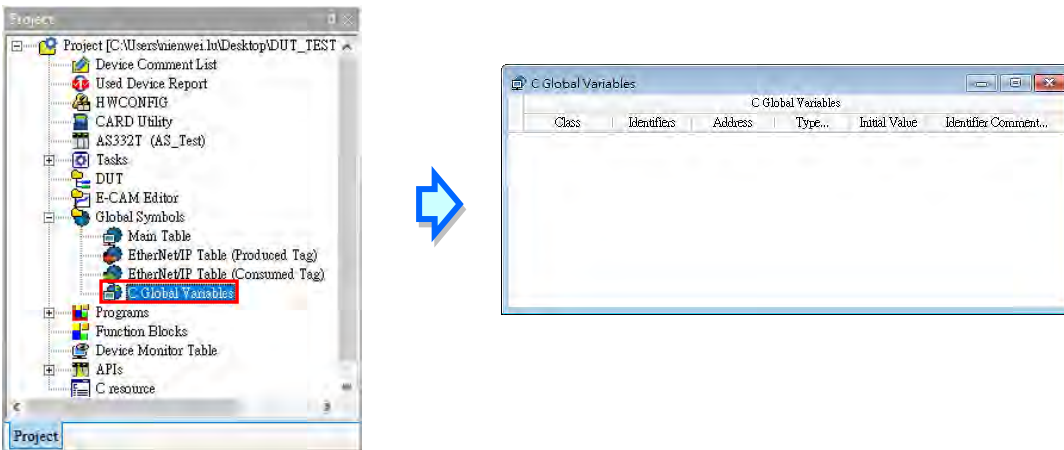
In the C editing workspace, only the standard C writing is supported and other high-level syntax like PLC instructions, C++ or Visual C is not supported. The operation in the C editing workspace is similar to that in a common text editor. When editing a C program, directly type characters or modify them following the format of C syntax in the workspace and switch to the next line when you press **Enter** key of the keyboard.

```
0019 C_Var_0 = 123;
0020
0021
```

Array, Struct and Point and etc. can be used to make a variable declaration. Please note that the memory for variable declarations is limited. So variable declarations should be made within the memory-allowed range in case of the PLC crash. A new variable symbol should be added in the **C Global Variables** table. There are 1000 D devices for the C variables to use. The start address for C variables is set in the **C Global Variables** table. Right-click **C Global Variables** and select the **Set the start number of C language data exchange area** option from the context menu to set the start position of the data exchange area. One of D0~D29000 can be filled as the start address and then the system automatically assigns 1000 consecutive D devices. The device range of the current data exchange area is displayed once the setting is done.



Then double-click **C Global Variables** below. The **C Global Variables** window appears right away.

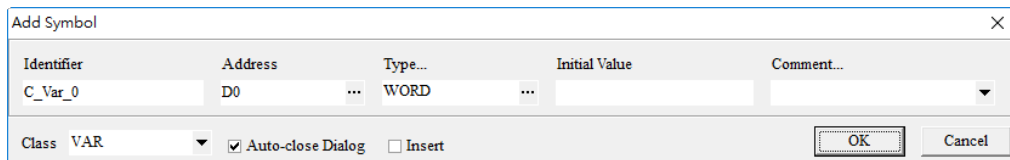
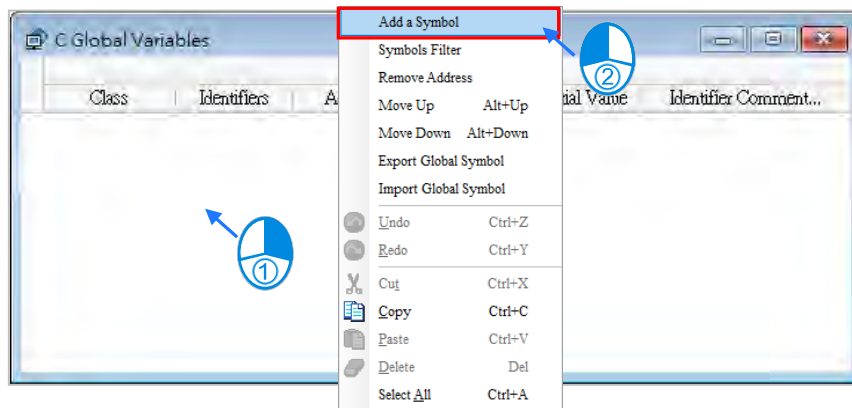


16

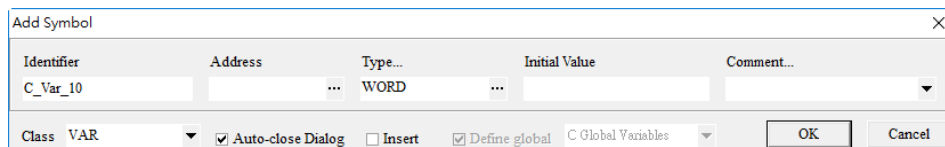
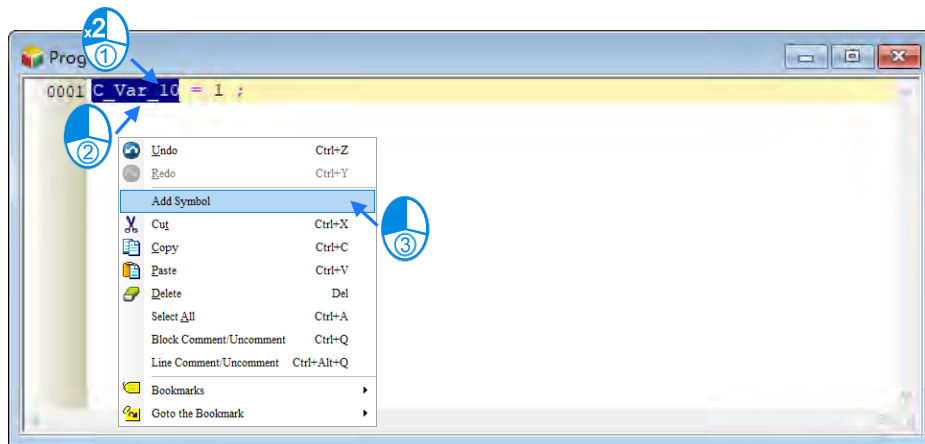
Right-click the blank space of the **C Global Variables** table and select the **Add a Symbol** item from the context menu. Then the variable symbol declaration window appears.

C variable symbols only support the four data types BOOL, WORD, DWORD and REAL, the start address within the range of D0~D999 and the 1000 consecutive D devices.

Then the variable symbol can be used directly in the LD and C programs after the variable declaration is done.



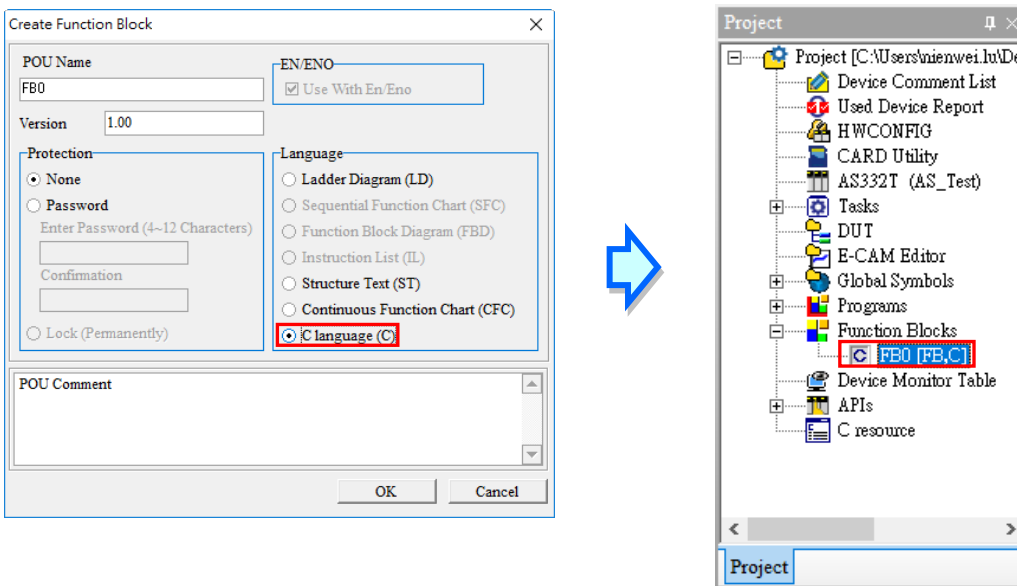
After an undeclared variable symbol is entered during the editing, double-click the text, press the right button of the mouse and select **Add Symbol** from the context menu. And then the variable symbol declaration window pops out.



During the program writing, the software automatically provides suitable variable symbols which have been created for option. Use the up [ ↑ ] and down [ ↓ ] keys of the keyboard or simply select the desired one from the list by pressing the left button of the mouse.

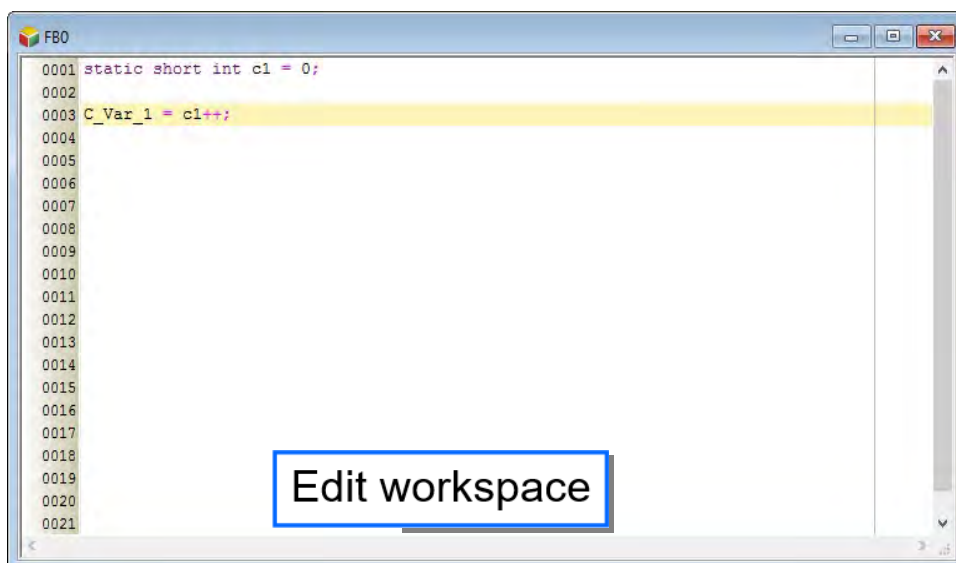


A function block is created by selecting the **C language** option for the **Language** field as below.



# 16

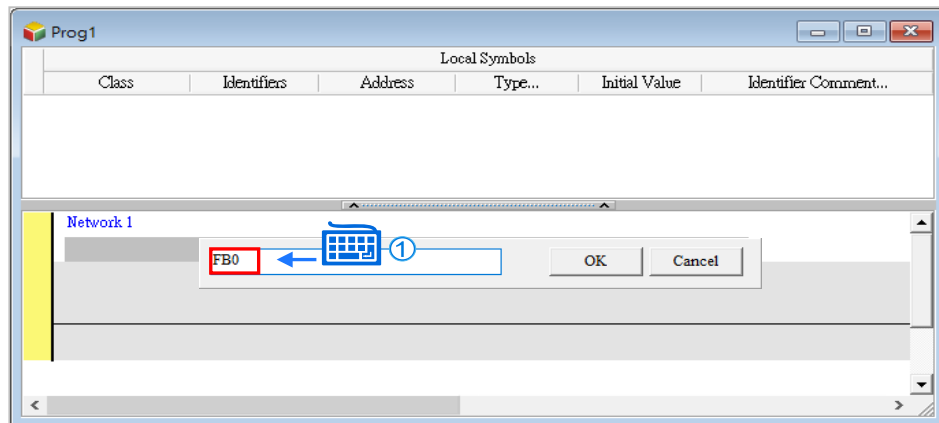
The editing of a C function block is the same as that of a C program. Type or modify the expression following the C syntax format in the following Edit workspace through your keyboard.



There are two methods to insert a C function block to a LD program.

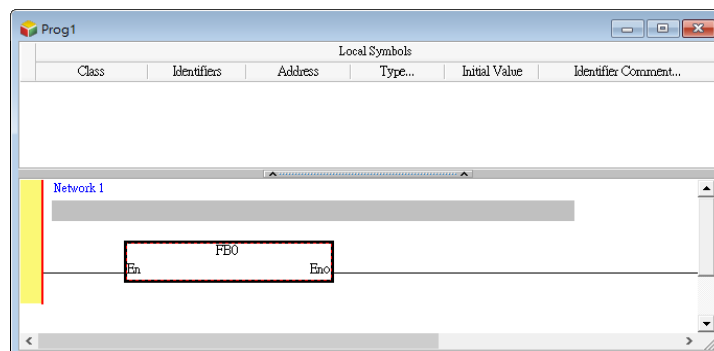
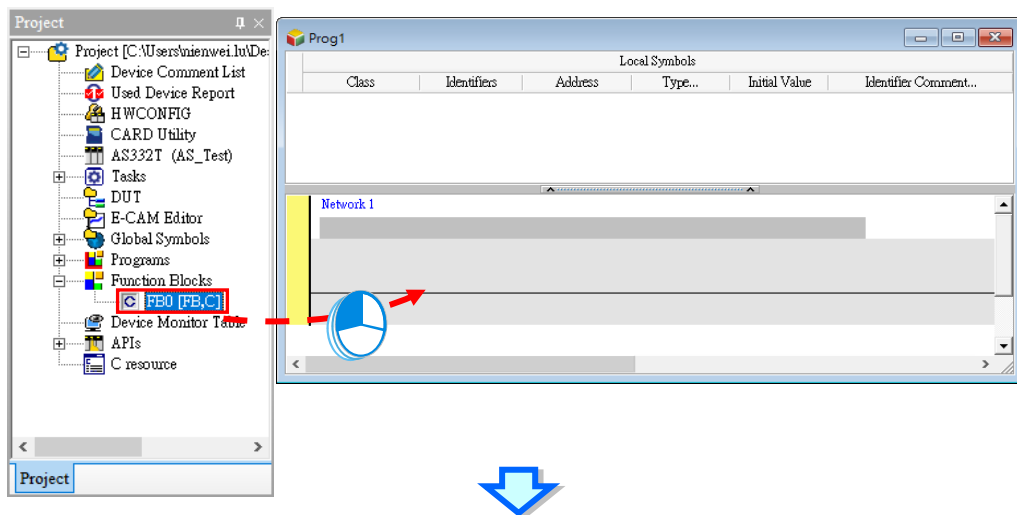
### Method 1

Type the function block name through the keyboard to insert the function block to the LD editing workplace.

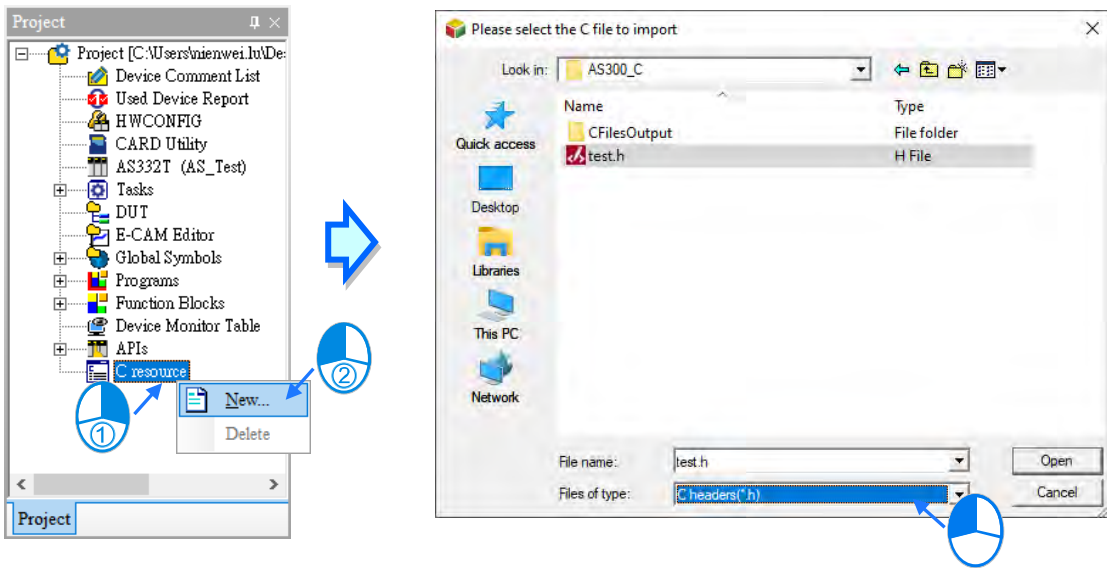
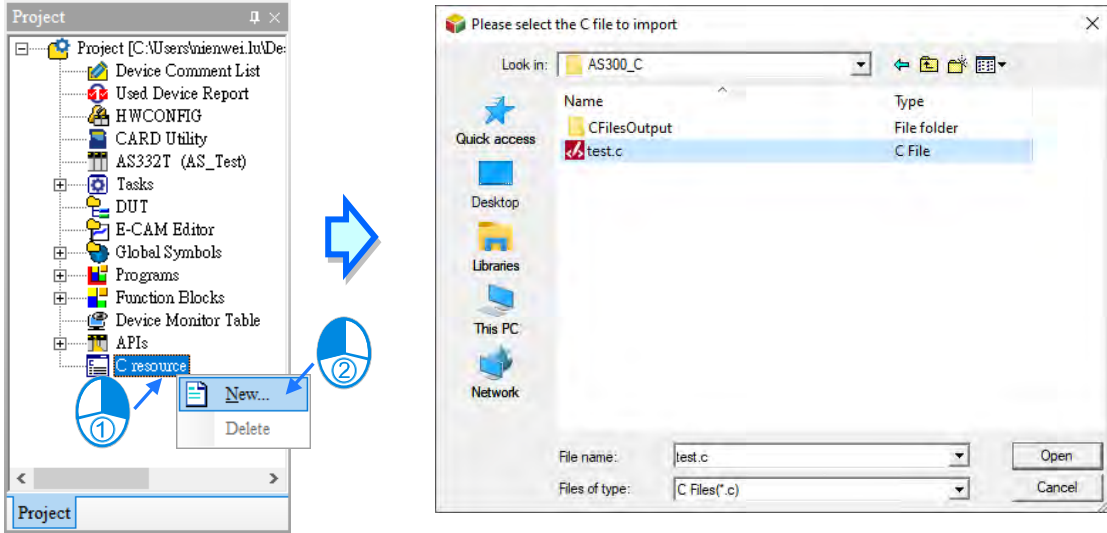


### Method 2

Unfold **Function Blocks** in the **Project** area, find the item to be inserted and drag it to the position where it should be inserted as below.



To import the generated “.c” and “.h” files, right-click **C resource**, select **New** from the context menu and then choose the C file to be imported in the pop-up window as follows. To import the “.h” file, choose C headers (\*.h) for the **Files of Type** field.

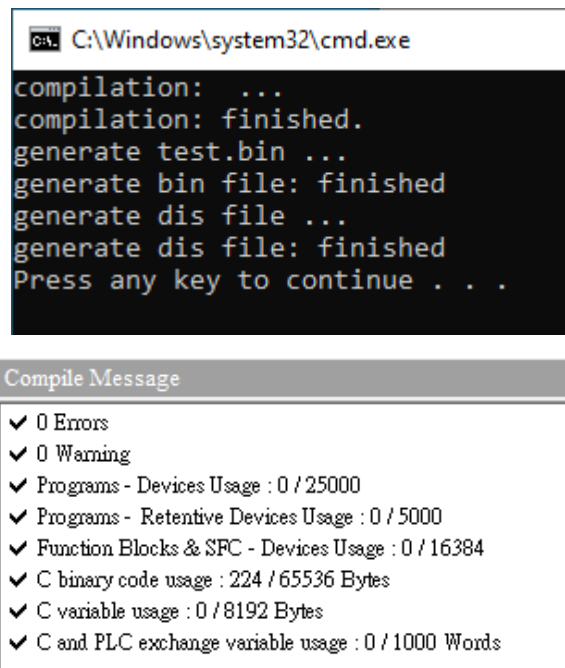


16



### 16.1.3 C Compiling and Debugging

ISPSOft provides the standard GCC compiler as well as triangular function and exponential function libraries which have been built in the PLC. They can be used without being declared. But ISPSOft does not provide other standard C function libraries (e.g. math.h) on the market, in order to save the storage space for C codes in the PLC.



```
C:\Windows\system32\cmd.exe
compilation: ...
compilation: finished.
generate test.bin ...
generate bin file: finished
generate dis file ...
generate dis file: finished
Press any key to continue . . .
```

Compile Message

- ✓ 0 Errors
- ✓ 0 Warning
- ✓ Programs - Devices Usage : 0 / 25000
- ✓ Programs - Retentive Devices Usage : 0 / 5000
- ✓ Function Blocks & SFC - Devices Usage : 0 / 16384
- ✓ C binary code usage : 224 / 65536 Bytes
- ✓ C variable usage : 0 / 8192 Bytes
- ✓ C and PLC exchange variable usage : 0 / 1000 Words

ISPSOft only shows the error messages which emerge after the GCC compiling is done. But it does not support the function of jumping to the location where the error occurs to make a modification by clicking an error statement.

The online monitoring of existing PLC programs and function blocks can be conducted. But ISPSOft does not support the line-by-line monitoring of the C codes in C programs.

### 16.1.4 Internal Function Library Instructions

The common four mathematical operations of integers or floating-point numbers, "addition, subtraction, multiplication and division" have already been built in ISPSOft. So they can be used directly without being declared additionally.

The names of built-in triangular functions:

C function name	ISPSOft-edited name	Format	Description
sinf	DELTA_SINF	Float DELTA_SINF (float radian)	Sin operation
cosf	DELTA_COSF	Float DELTA_COSF F (float radian)	Cos operation
tanf	DELTA_TANF	Float DELTA_TANF (float radian)	Tan operation
asinf	DELTA_ASINF	Float DELTA_ASINF (float arg)	$\text{Sin}^{-1}$ operation
acosf	DELTA_ACOSF	Float DELTA_ACOSF (float arg)	$\text{Cos}^{-1}$ operation
atanf	DELTA_ATANF	Float DELTA_ATANF (float arg)	$\text{Tan}^{-1}$ operation
sinhf	DELTA_SINHF	Float DELTA_SINHF (float arg)	Sinh operation
coshf	DELTA_COSHF	Float DELTA_COSHF (float arg)	Cosh operation
tanhf	DELTA_TANHF	Float DELTA_TANHF (float arg)	Tanh operation

The names of built-in exponential functions:

C function name	ISPSOft-edited name	Format	Description
sqrf	DELTA_SQRF	Float DELTA_SQRF (float value)	Square root operation
logf	DELTA_LOGF	Float DELTA_LOGF (float base, float value)	Log operation
lnf	DELTA_LNF	Float DELTA_LNF (float arg)	Ln operation
powf	DELTA_POWF	Float DELTA_POWF (float base, float power)	Exponential operation

### 16.1.5 Bookmark and Comment

The **Bookmarks** function is a convenience tool used for a search for or moving to a specific editing location.

- **Add/ Remove a bookmark**

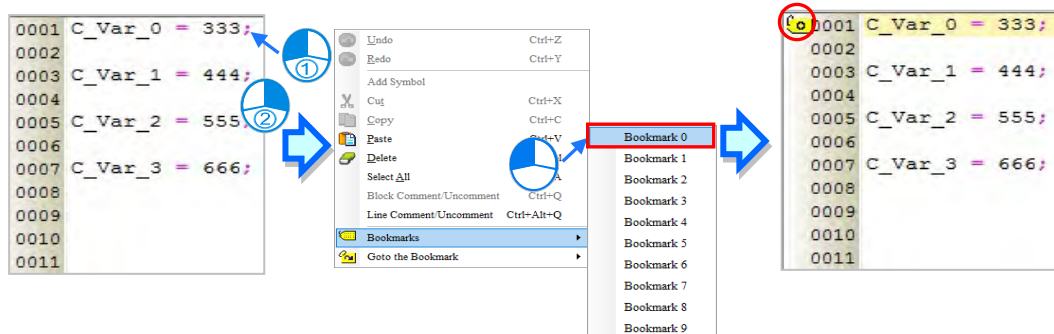
Select the line to which a bookmark is added and right-click the selected line. Then select the **Bookmarks** option from the context menu and pick the desired bookmark number.

To remove a bookmark, in the same way select the line of which a bookmark is to be removed, right-click the selected line and then select the **Bookmarks** option from the context menu as well as pick the bookmark number which is the same as that of the selected line.

However, the original bookmark of the selected line will be replaced by the currently picked one if they are different.

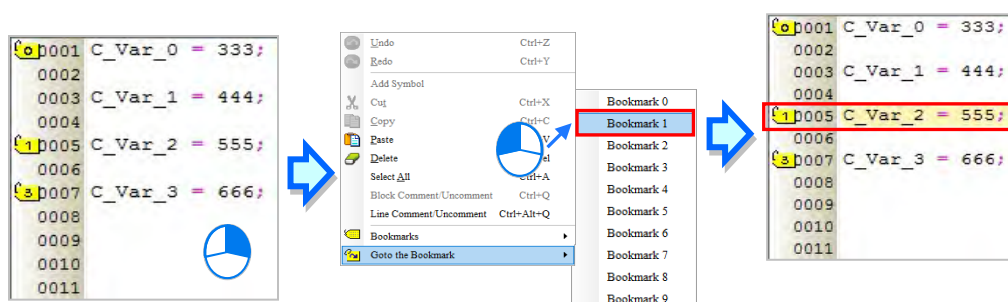
In addition, if you select **Edit menu > Bookmarks > Remove All Bookmark**, all bookmarks will be cancelled in the editing window.

If you select one bookmark number which has been used for other line, the picked bookmark number will be moved to the currently selected line.



- **Go to the bookmark**

Right-click the workplace, select the **Goto the Bookmark** option from the context menu and then pick the bookmark number from the dropdown list to come to the target bookmark location.

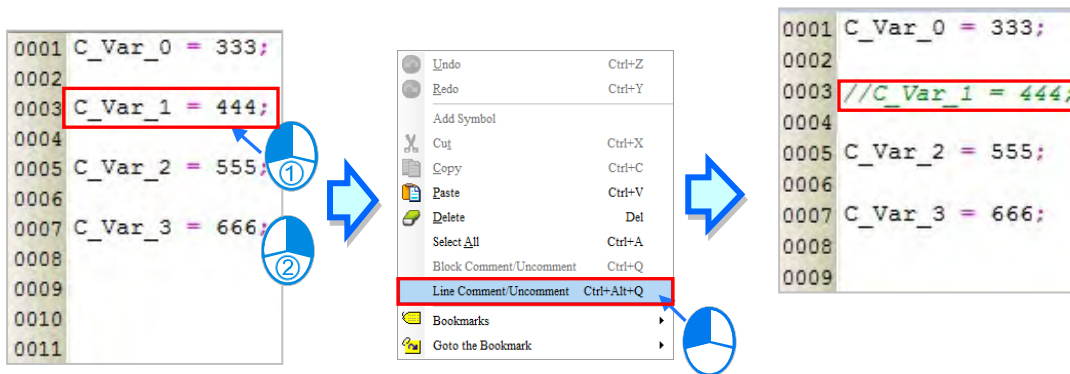


The **Comment** function is a convenience tool used to comment the selected block or lines.

- **Comment/Uncomment a line**

Right-click the line to be commented and select the **Line Comment/Uncomment** option from the context menu. Then the two slashes // are automatically added to the start of the selected line.

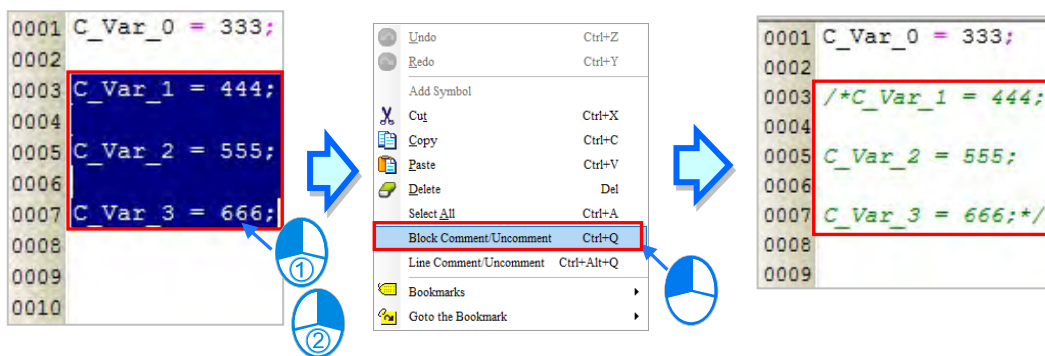
To uncomment a line, in the same way right-click the line to be uncommented and select the **Line Comment/Uncomment** option from the context menu. Then the two slashes // are automatically removed from the start of the selected line.



- **Comment/Uncomment a block**

Right-click the block to be commented and select the **Block Comment/Uncomment** option from the context menu. Then the marks /\* and \*/ are automatically added to the start and end of the selected block respectively.

To uncomment a block, in the same way right-click the block to be uncommented and select the **Block Comment/Uncomment** option from the context menu. Then the marks /\* and \*/ are automatically removed from the start and end of the selected block.

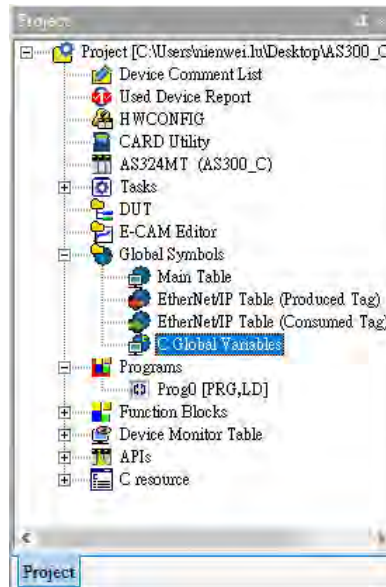


### 16.1.6 Operation Example

#### Example:

One value for the Sin function is output by a change of  $5^\circ$  every 10ms and the output values for the Sin function of  $0\sim 355^\circ$  are recorded via the data tracer function

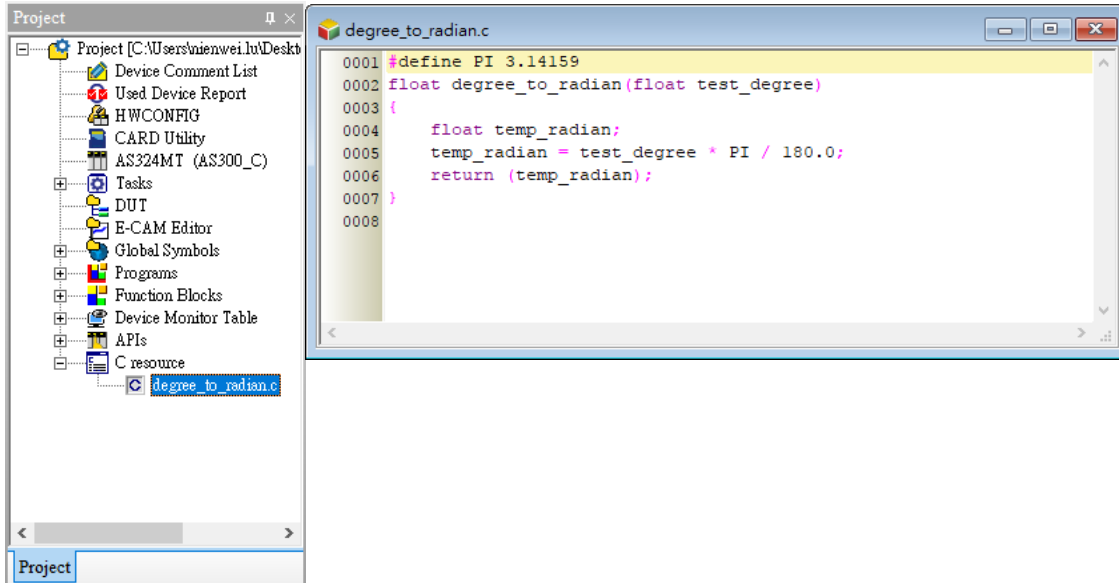
**Step 1:** Create C global variable symbols necessary for PLC and C program. See Section 16.1.2 for details.



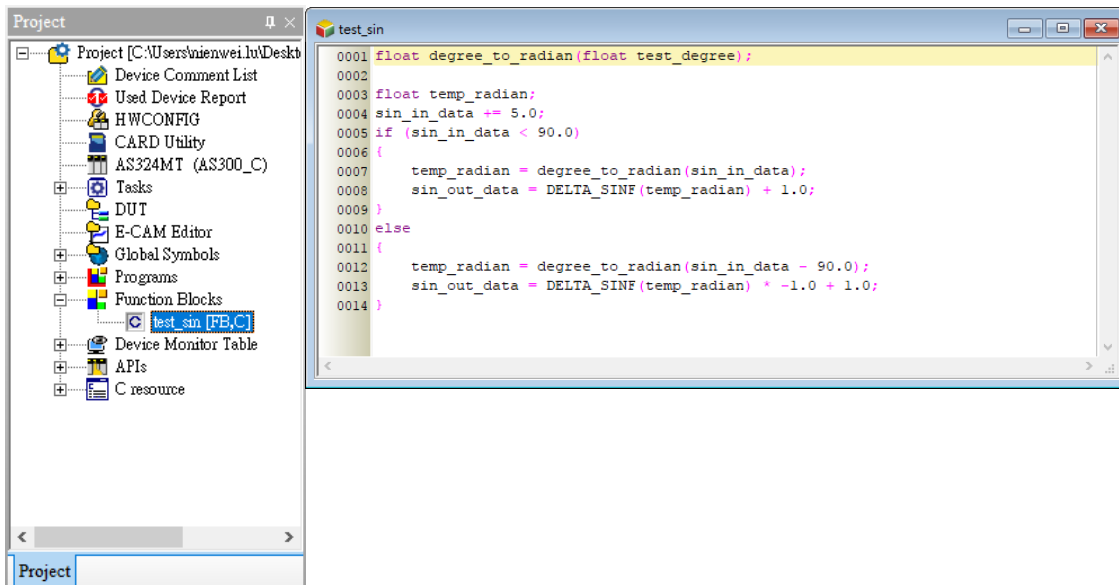
C Global Variables						
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...	
VAR	sin_out_data	D12	REAL	N/A		
VAR	sin_in_data	D10	REAL	N/A		

16

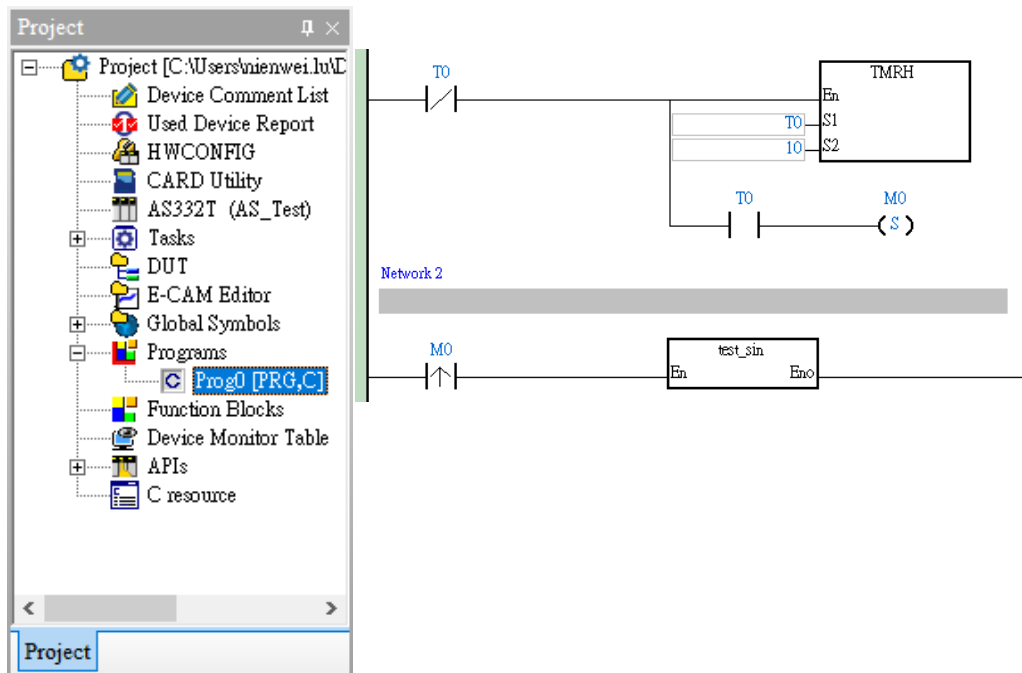
**Step 2:** Import the degree\_to\_radian.c function which has been edited and debugged. See Section 16.1.2 for details. This step can be skipped according to actual need.



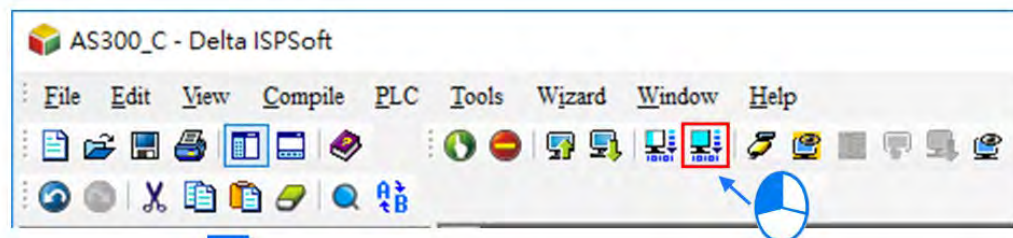
**Step 3:** Edit a new function block **test\_sin** and write the computing process in it where the degree\_to\_radian library function is called.



**Step 4:** Create a POU and write a segment where the **test\_sin** FB is called every 10ms .



**Step 5:** Press the **Compile the Project** button. Suppose no error occurs, the process is seen as below.



16

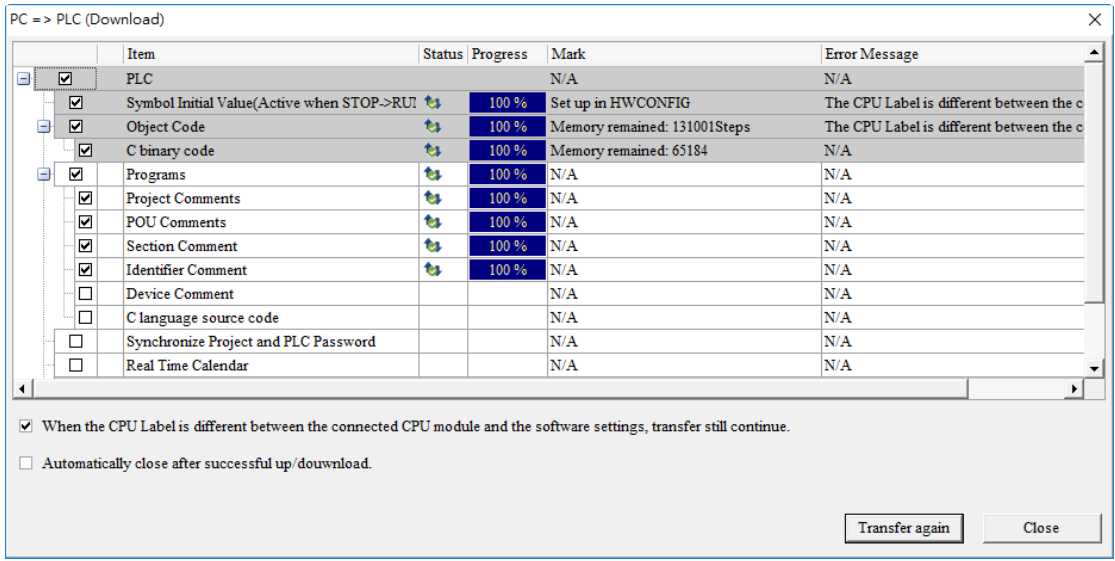
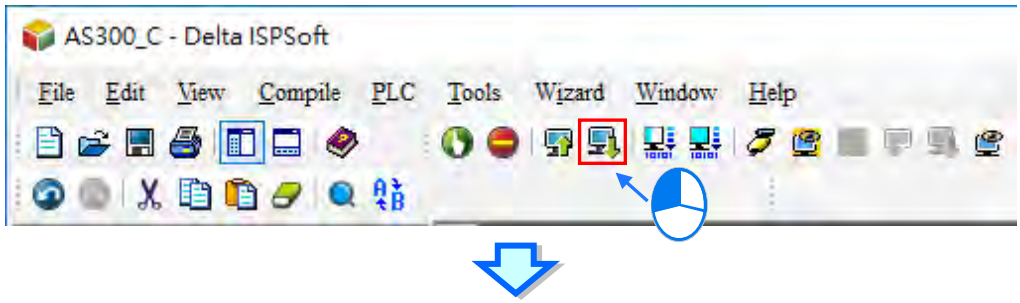
```

C:\Windows\system32\cmd.exe
compilation: ...
compilation: finished.
generate test.bin ...
generate bin file: finished
generate dis file ...
generate dis file: finished
Press any key to continue . . .
    
```

**Compile Message**

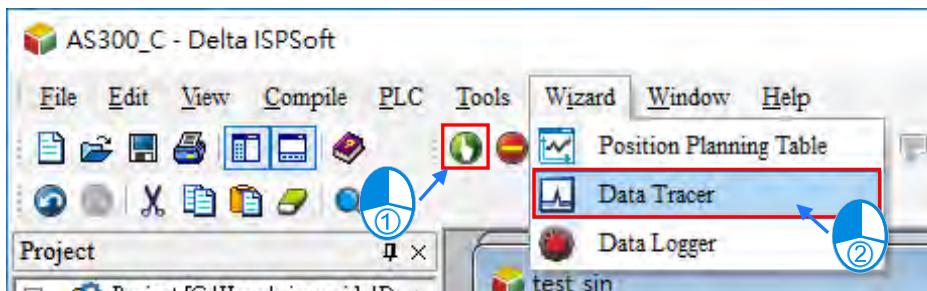
- ✓ 0 Errors
- ✓ 0 Warning
- ✓ Programs - Devices Usage : 0 / 25000
- ✓ Programs - Retentive Devices Usage : 0 / 5000
- ✓ Function Blocks & SFC - Devices Usage : 0 / 16384
- ✓ C binary code usage : 224 / 65536 Bytes
- ✓ C variable usage : 0 / 8192 Bytes
- ✓ C and PLC exchange variable usage : 0 / 1000 Word

**Step 6:** Press **Download to PLC** button to download the program to the PLC after compiling is done.



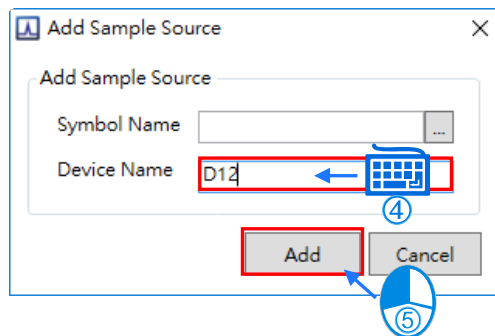
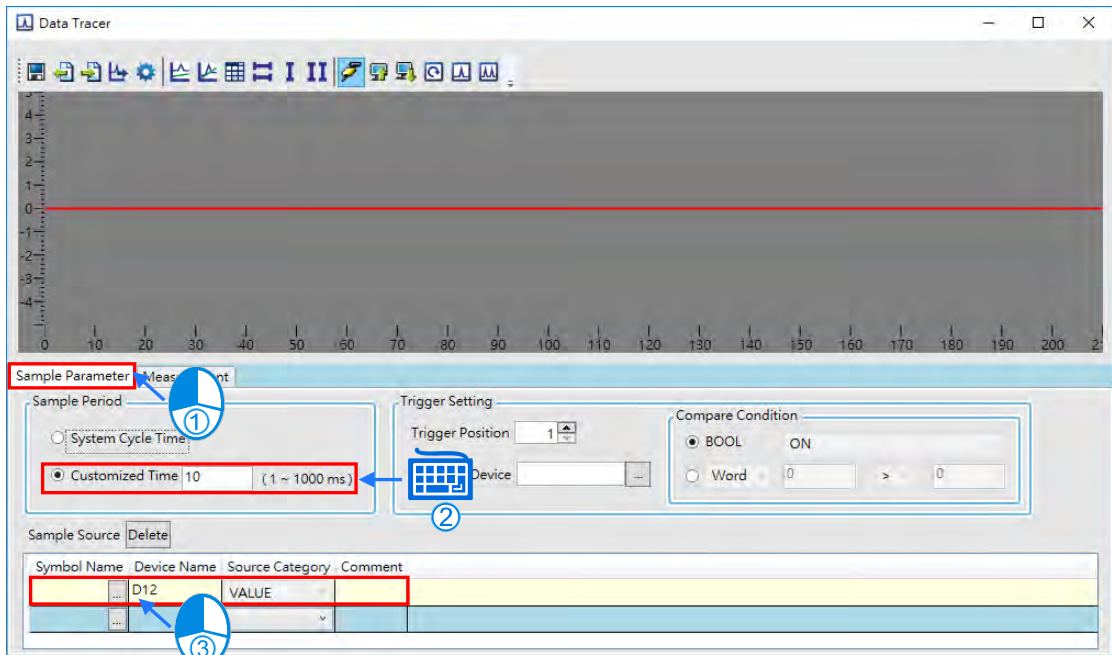
# 16

**Step 7:** Press the **Run** button and then start the **Data Tracer** function.

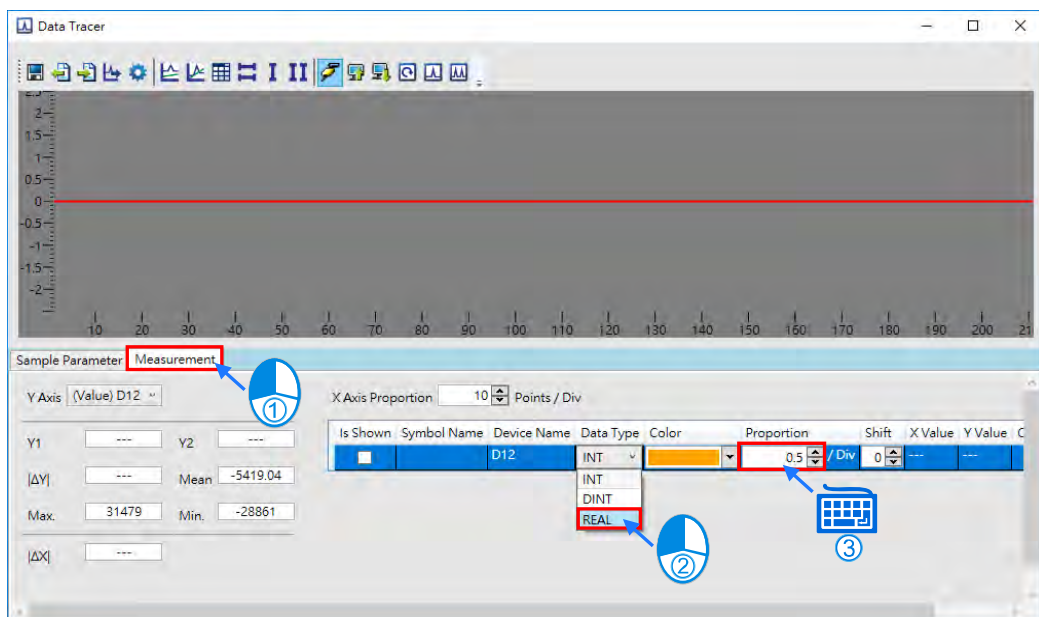




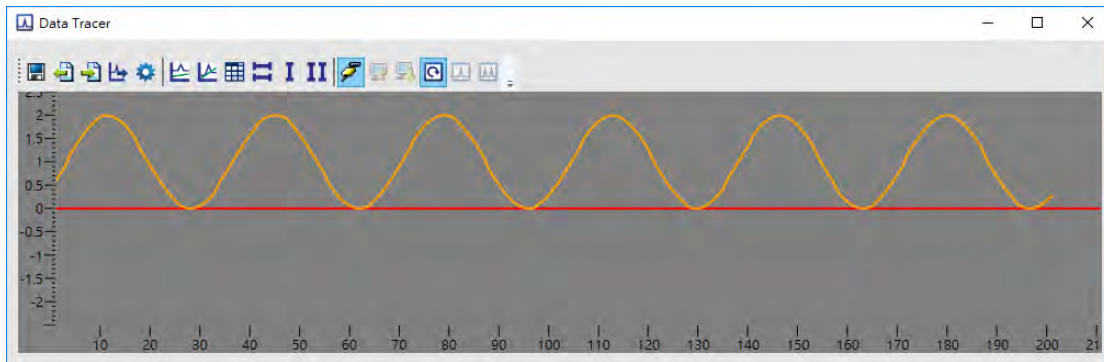
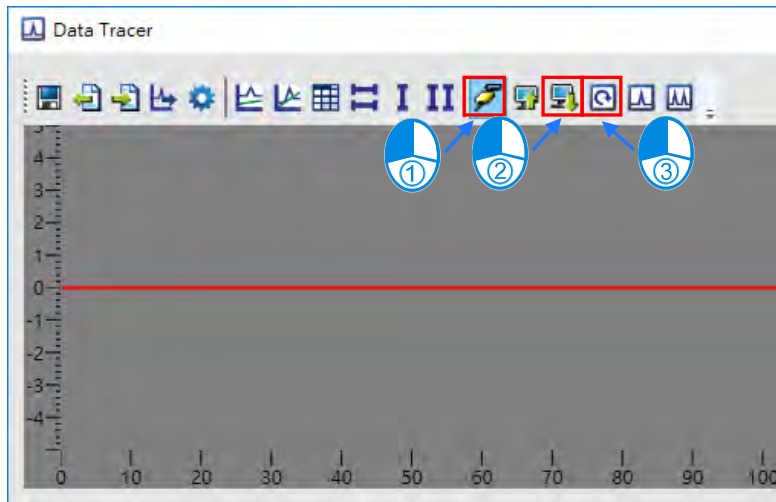
**Step 8:** Set sample parameters and measurement for Data Tracer.



16



**Step 9:** Set the online, download and monitor for Data Tracer as follows.





# Chapter 17 Auxiliary Editing Tools

## Table of Contents

<b>17.1</b>	<b>ISPSOft Editing Tools and Auxiliary Functions .....</b>	<b>17-2</b>
17.1.1	Modify PLC Types.....	17-2
17.1.2	Change DVP Series Modules to AS Series Modules .....	17-3
17.1.3	Download/Upload a Project .....	17-14
17.1.4	Find/Replace in LD/FBD/ CFC .....	17-17
17.1.5	Find/Replace in IL/ST .....	17-20
17.1.6	Find/Replace in SFC .....	17-23
17.1.7	Find Function in Symbol Table.....	17-25
17.1.8	Find/Replace in Project.....	17-26
17.1.9	Print Function.....	17-27
<b>17.2</b>	<b>Devices and Registers .....</b>	<b>17-30</b>
17.2.1	Device Comment List for a DVP Series PLC .....	17-30
17.2.2	AH/AS Series Device Comment List.....	17-34
17.2.3	Using Device Report.....	17-40
17.2.4	Edit Register Memory .....	17-44
17.2.5	Edit Bit Memory.....	17-51
17.2.6	Edit DVP Series File Register .....	17-56
<b>17.3</b>	<b>Step Positioning .....</b>	<b>17-63</b>
17.3.1	Using Step Positioning .....	17-63
<b>17.4</b>	<b>Program Comparison.....</b>	<b>17-64</b>
17.4.1	Introduction to Program Comparison.....	17-64
17.4.2	Compare with File.....	17-65
17.4.3	Compare with PLC .....	17-67

## 17.1 ISPSoft Editing Tools and Auxiliary Functions

### 17.1.1 Modify PLC Types

In ISPSoft, users can change PLC types, most series offers two-way but some series only offer one-way modification. For example, DVP Series can change to AS200/AS300 Series, but not vice versa; AH5xo Series can change to AH560 Series but not vice versa. Please refer to the following table.

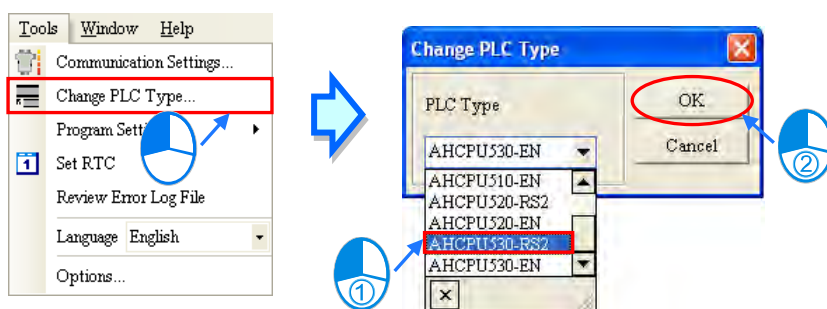
Purpose Source	AS200	AS300	AH5x0	AH5x1	AH560	AHxxEMC	DVP	DVP xxMC
AS200	V	V					Support ES3	
AS300	V	V					Support ES3	
AH5x0			V	V	V	V		
AH5x1				V	V	V		
AH560				V	V	V		
AHxxEMC						V		
DVP	V	V					V (*1)	
DVPxxMC								V

\*1. DVP-ES3 Series cannot change other DVP Series, but other DVP Series can change to DVP-ES3 Series.


# 17

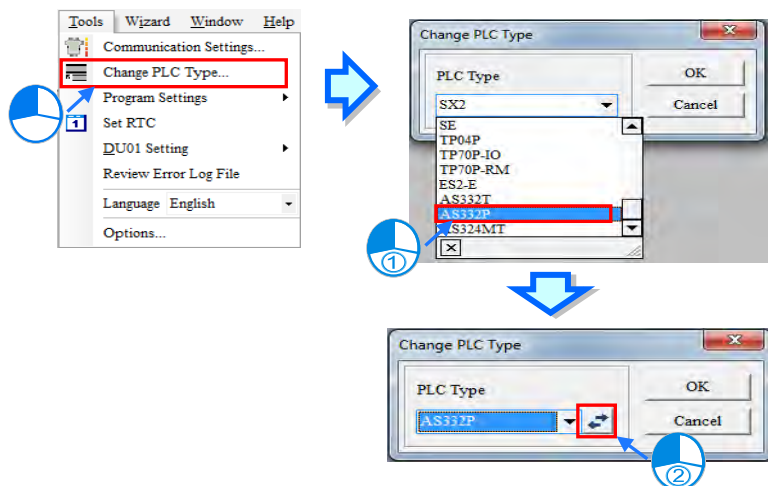
The modified PLC types will change the content of current project. Thus, the project should be saved first before modify the PLC. Since PLC functions, device range or support command can vary before or after the modification, please make sure the original program, parameter setting, hardware and network planning is functioning properly.


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**.

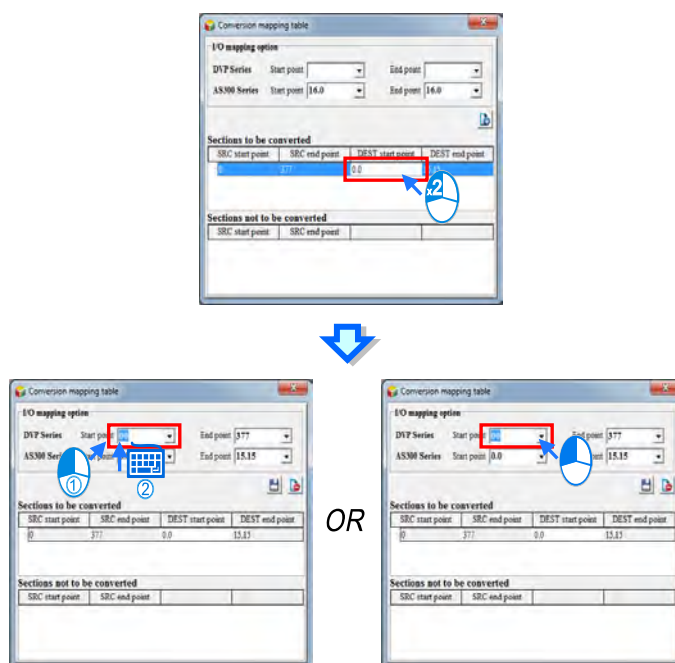


### 17.1.2 Change DVP Series Modules to AS Series Modules

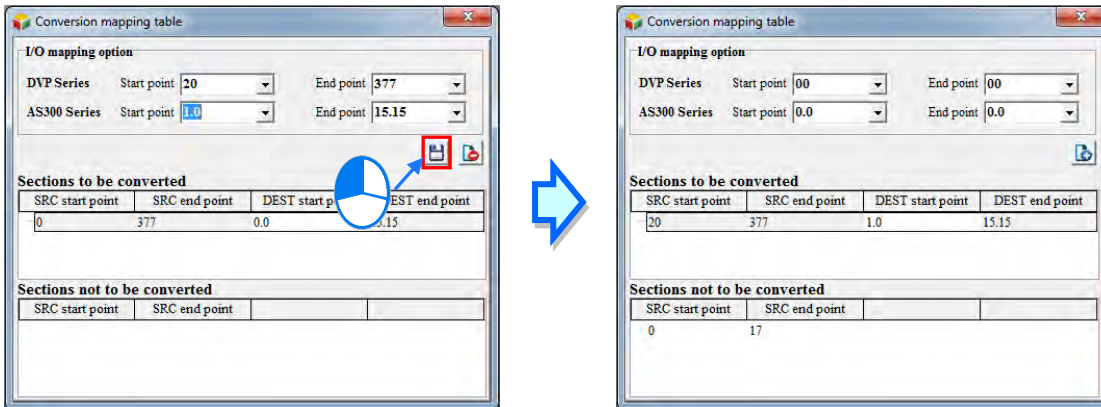
For the change of a DVP series module to an AS series module, user can click Tools tab and then Change PLC Type to see a change PLC Type window. Select the PLC type you'd like to change from the drop-down list and then click the exchange icon  to complete the conversion I/O mapping change.




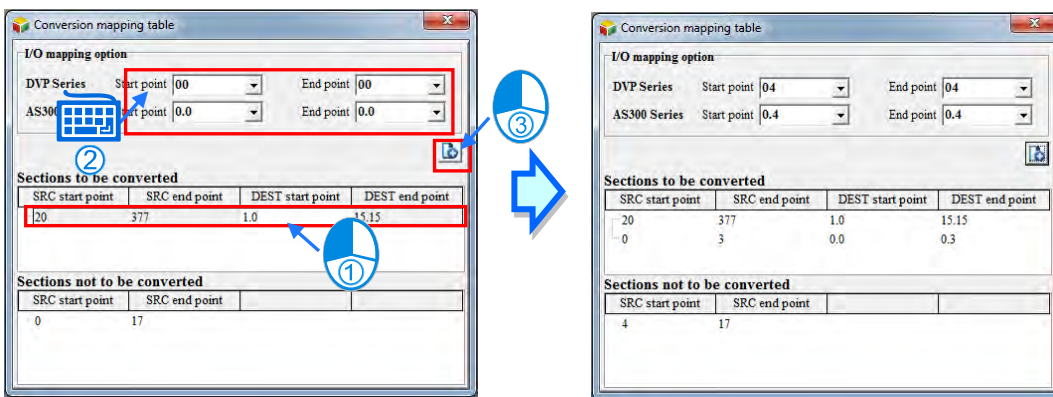
Input the start/end point of the DVP series module to be changed with the AS series module and click save . After that users can see the start/end points of the DVP series and AS series in the list of sections to be converted. The SRC start point and end point correspond to the start and end point of the input points X and Y of a DVP series module. DEST start and end points correspond to the input points X and Y of an AS series module. Please note if the I/O conversion range of a DVP series module and an AS series module cannot be matched, a warning will appear; Double click the mouse on the selected box(See below), the save or delete icon will appear for users to directly change the value in the box or select it from the drop-down menu.




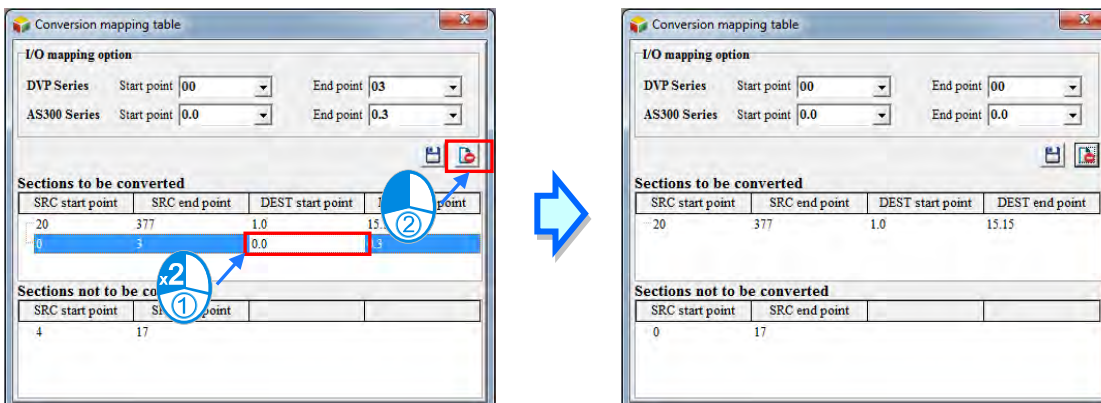
When start/end points are set, click save to complete the setting. Then, **Sections to be converted** and **Sections not to be converted** will appear.



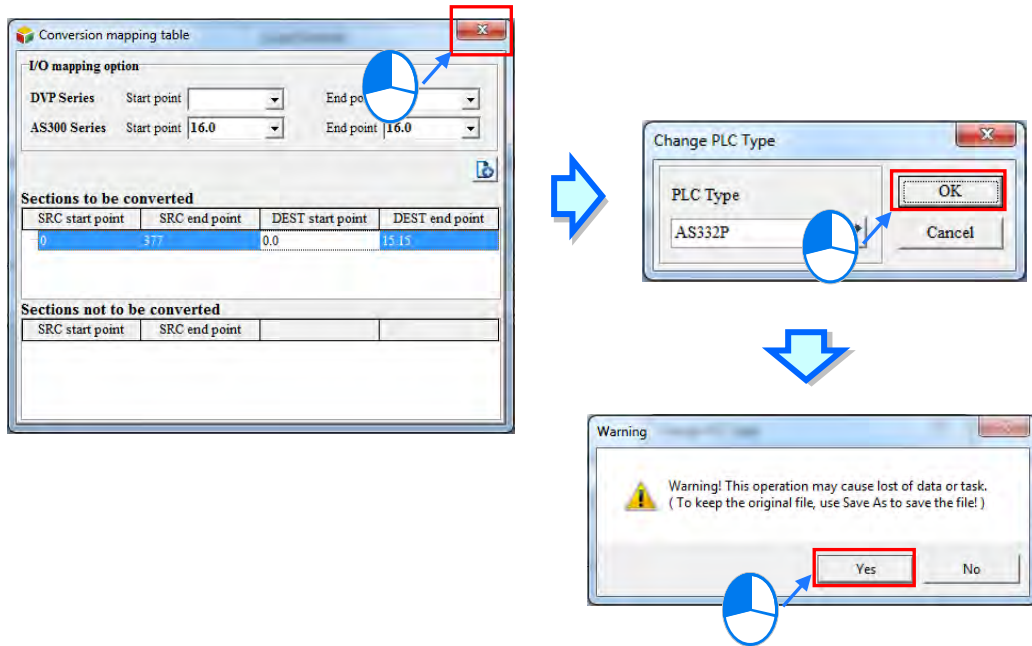
To add a new conversion, users can click on the sections to be converted list and then input new start/end points and then click Plus  after that a new conversion will be added.



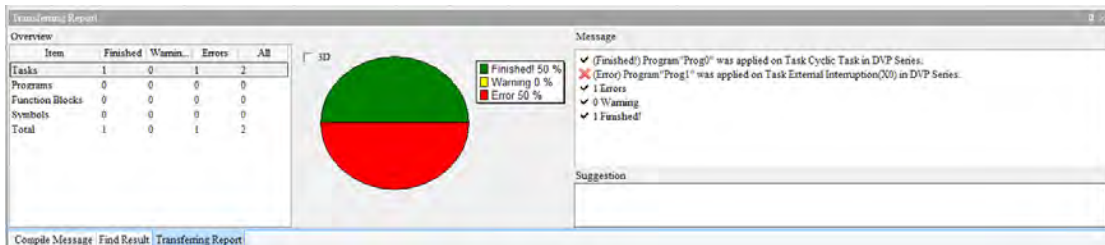
To delete a conversion item from the list, users can double click to select the item to be deleted on the list and then click the delete icon  after that the selected item will be deleted.



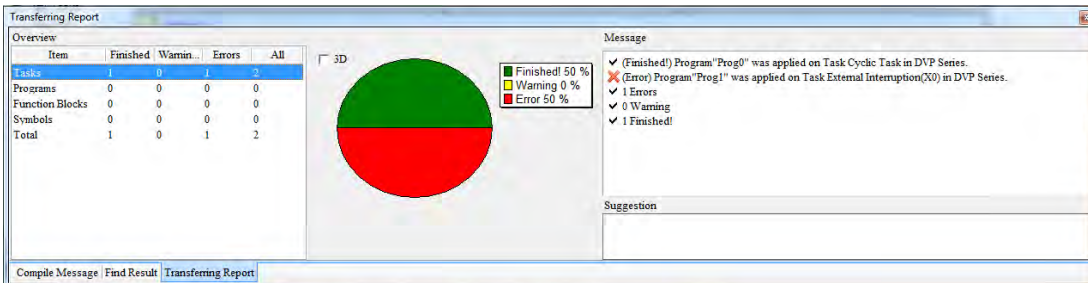
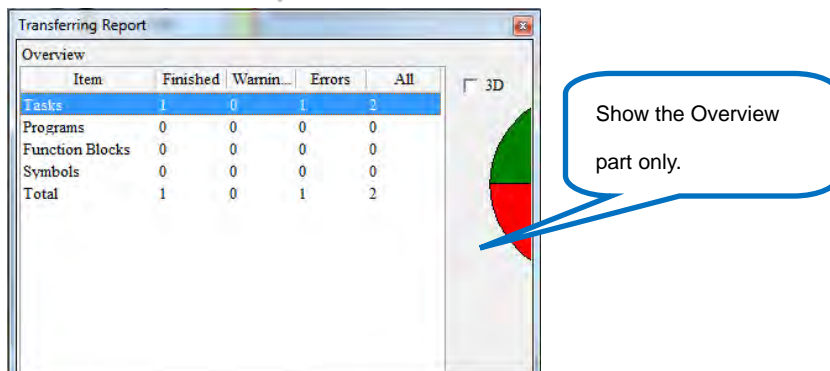
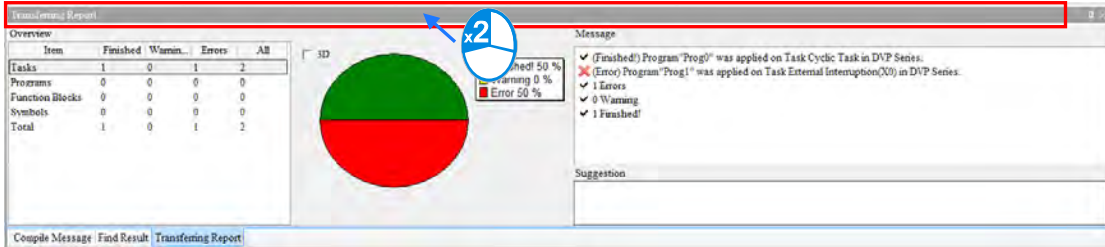
After the conversion mapping table is set, users can close the window. A Change PLC Type window will then appear. Click OK to confirm the change and a warning window stating “Warning! This operation may cause loss of data or task.” will appear. (If users want to keep the original file, click “No” to cancel the conversion, then use Save As to save the file for conversion). Click Yes to double confirm the change for conversion.



After the conversion is done, a Transferring Report will show up.

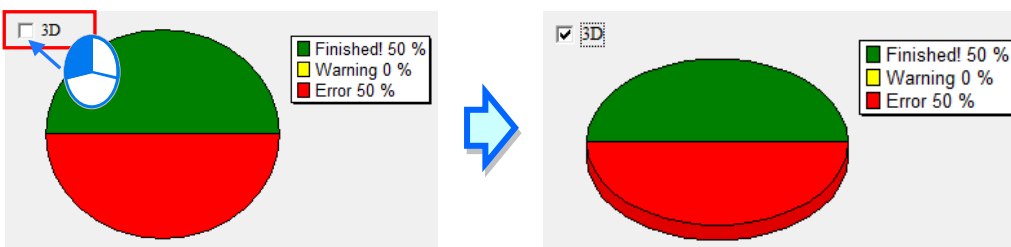


Click and drag the header of the Transferring Report to move the window. Double click the header to have it shown the Overview part only and double click the header again to have a complete window back.



17

On the left side of the Transferring Report, there is an overview of the conversion, including tasks, programs, function blocks, symbols and total. A pie chart will show the corresponding result of the selected item in the middle of the transferring report. Green indicates the percentage of the finished part, yellow shows the percentage of the warning part, and red represents the percentage of the error part. When the 3D option is selected, the pie chart will be shown in 3D.





On the right side of the transferring report, a message section shows the description of the selected item from the overview section. Users can click any number shown in the overview to see the detailed descriptions of the selected item.

Shows the description of selected item (Tasks: Finished)

Item	Finished	Warn...	Errors	All
Tasks	1	0	2	3
Programs	0	0	1	1
Function Blocks	0	0	0	0
Symbols	0	0	0	0
Total	1	1	2	4

Message: (Finished) Program "Prog0" was applied on Task Cyclic Task  
 ✓ 1 Finished!

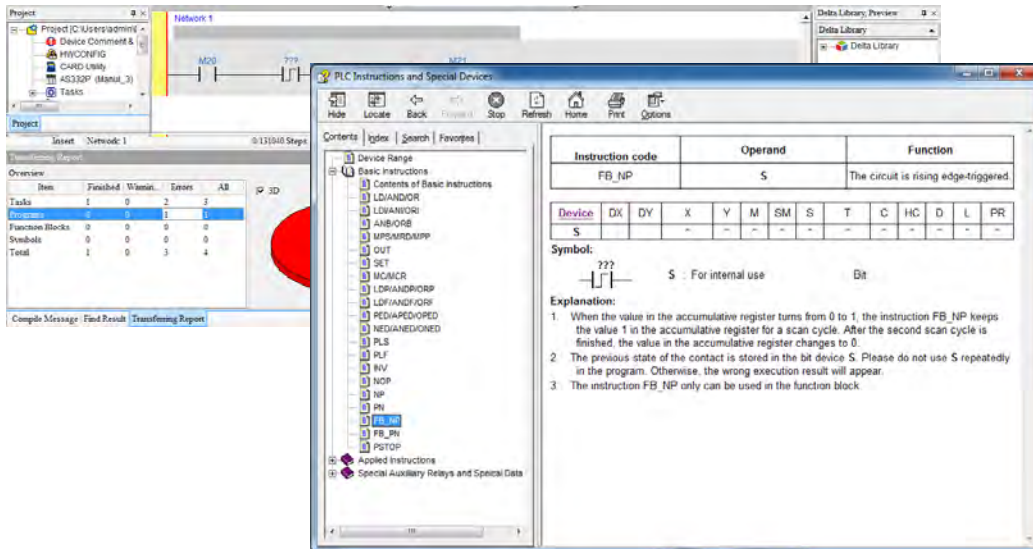
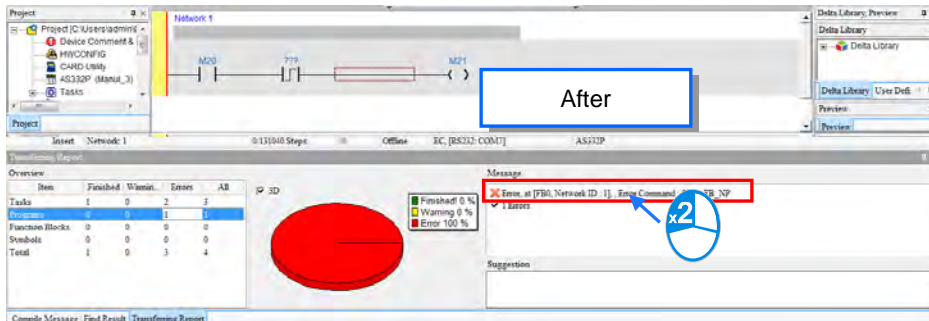
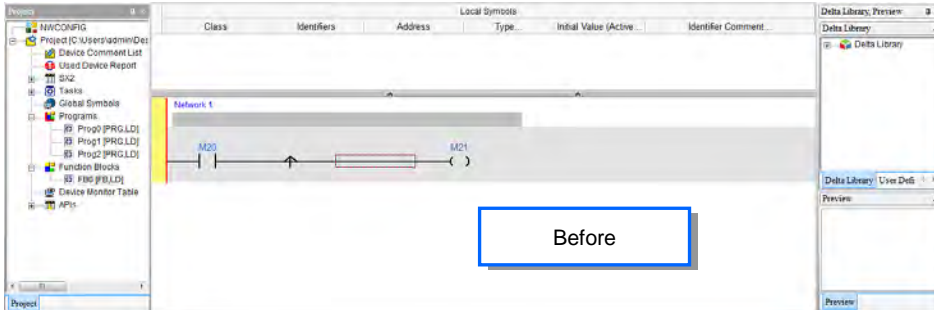


Shows the description of selected item (Programs: Warning)

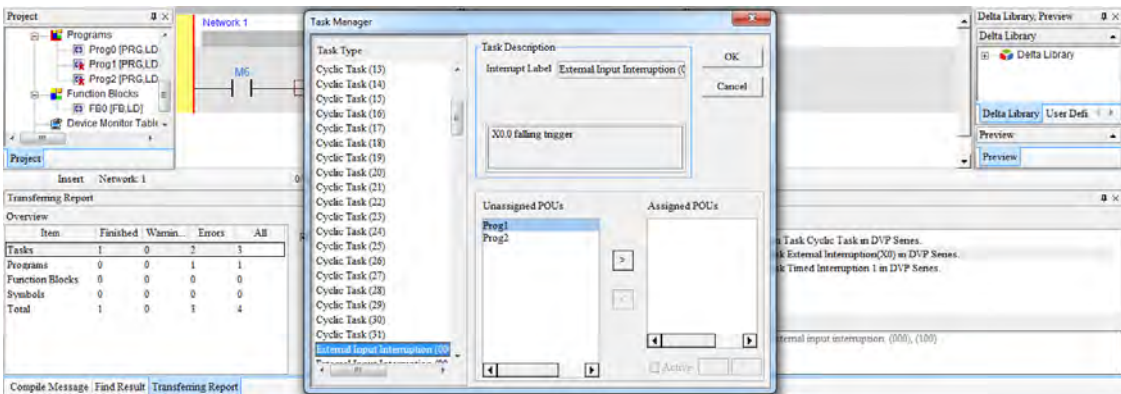
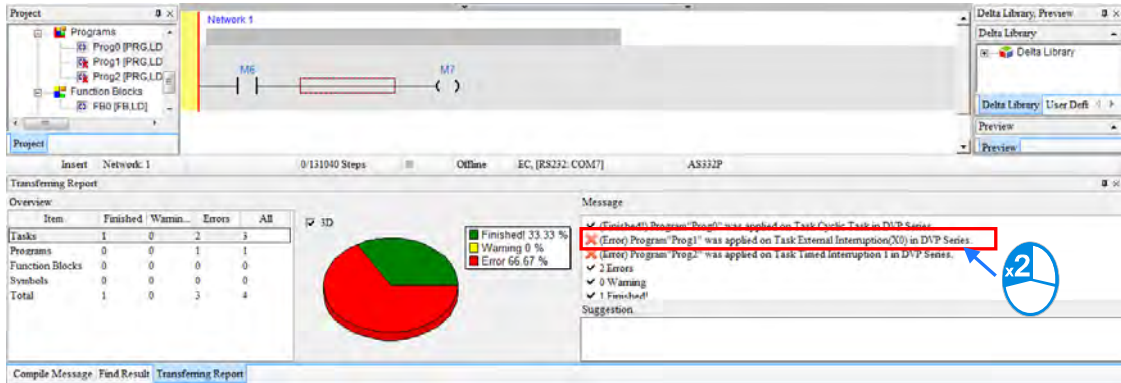
Item	Finished	Warn...	Errors	All
Tasks	1	0	2	3
Programs	0	1	0	1
Function Blocks	0	0	0	0
Symbols	0	0	0	0
Total	1	1	2	4

Message: Warning: Position [Prog0, Network ID: 1], Warning Command: DM...  
 ✓ 1 Warning

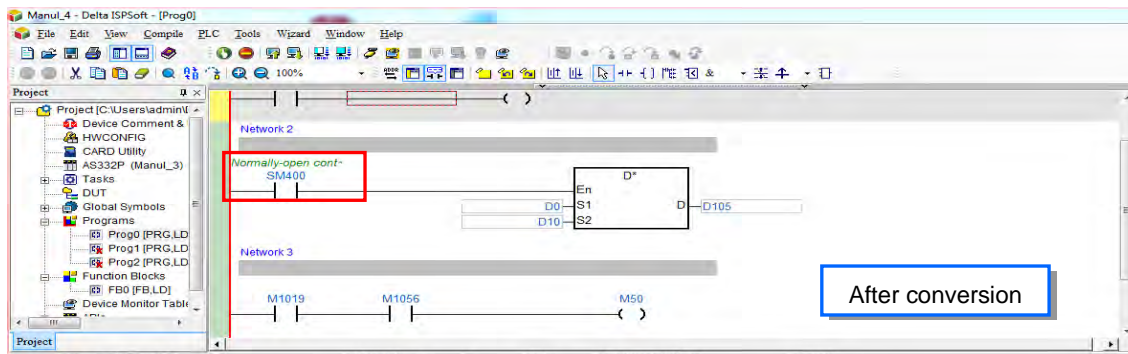
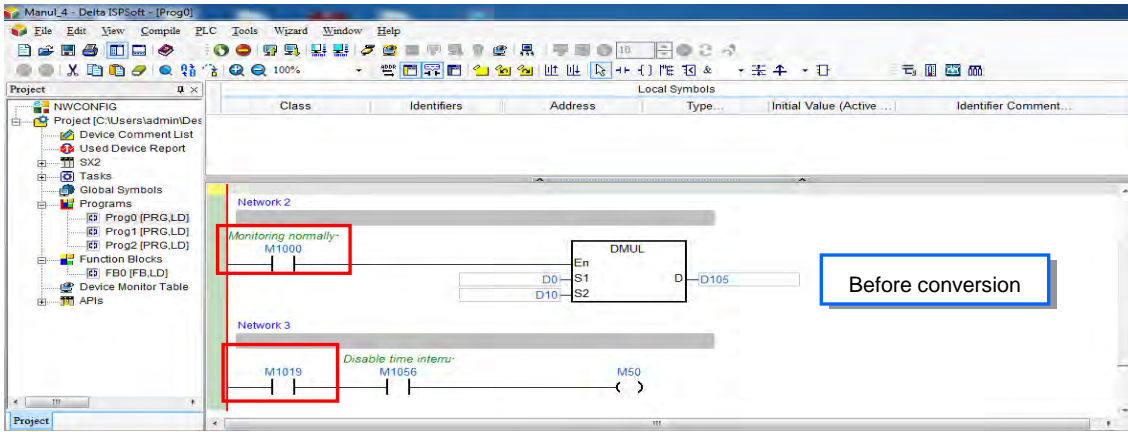
On the right side of the transferring report, a suggestion section targeting on “warning” or “error” messages in the selected item will be shown in the section. After transferring, users can use the corresponding index files to edit the instructions.



After the conversion is done, if there is any change in the Task Manager, suggestions will be shown in the suggestion section. Users can click the detailed message in the message section to open the Task Manager to edit the setting.



If the special M or special D of the DVP series can correspond to the ones in AS series, the system will generate the relays and registers automatically during the conversion. If not, users can find suggestions in the suggestion section after the conversion. Suggestion section is used for providing suggestions for users when warnings or errors occur.



Transferring Report

Item	Finished	Warnin...	Errors	All
Tasks	1	0	2	3
Programs	1	3	5	
Function Blocks	0	0	0	0
Symbols	0	0	0	0
Total	2	1	5	8

Message

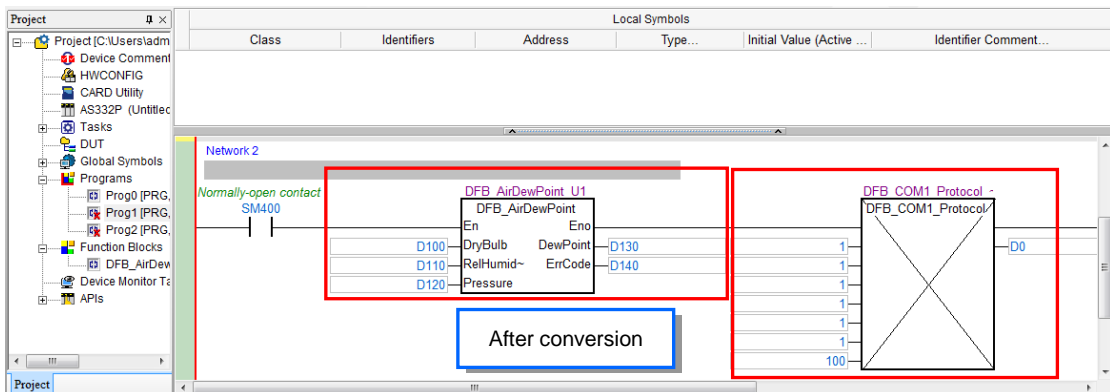
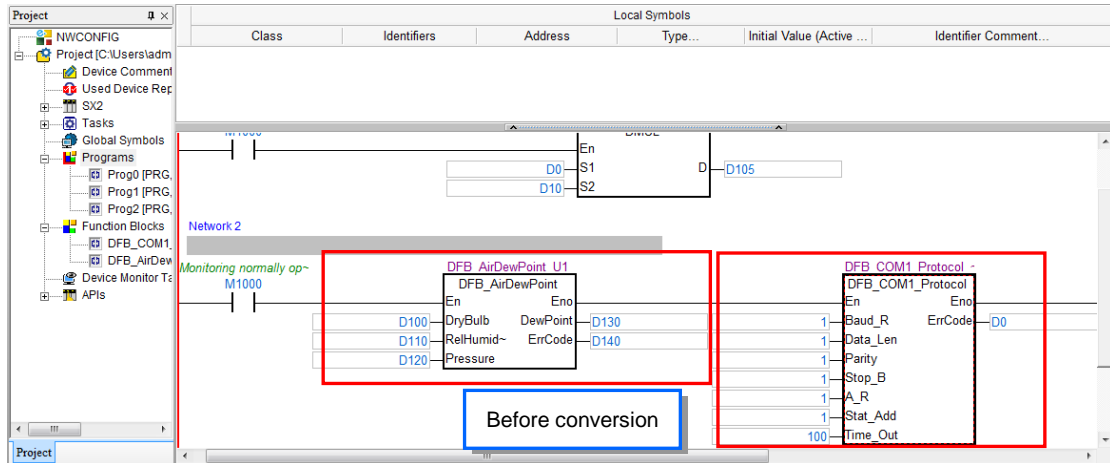
- Error at [Prog0, Network ID : 3], Error Command : M1056
- Error at [Prog0, Network ID : 3], Error Command : M1019
- Error at [FB0, Network ID : 1], Error Command : ???->FB\_NP

Suggestion

- Conversion manually. Suggest that referring API DSPD\_PWB

17

When executing program conversion, if both DVP series and AS series support Delta Function Blocks then it will automatically convert to support once execution is complete; after conversion, however, when AS series does not have any corresponding Delta Function Blocks, an error cross mark appears in the function block (see below).



If the symbols used in the DVP series and the symbols, instructions names, device names used in AS series are contradicted, after the conversion, errors will occur.

Class	Identifiers	Address	Type...	Initial Value (Active...)	Identifier Comment...
VAR	JOG	N/A(Auto)	WORD	N/A	
VAR	w1	N/A(Auto)	BOOL	N/A	



Item	Finished	Warnin...	Errors	All
Tasks	1	0	2	3
Programs	0	1	0	1
Function Blocks	0	0	0	0
Symbols	0	0	2	2
Total	1	1	4	6

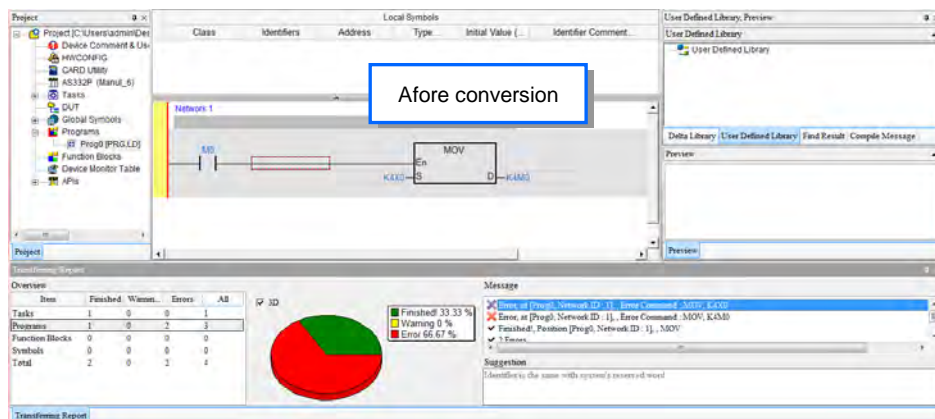
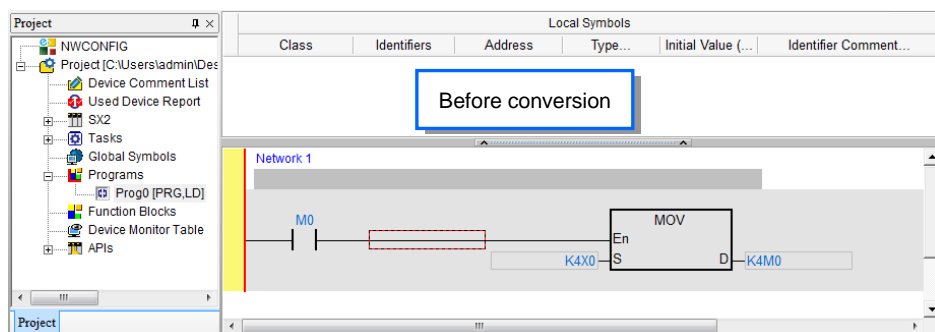
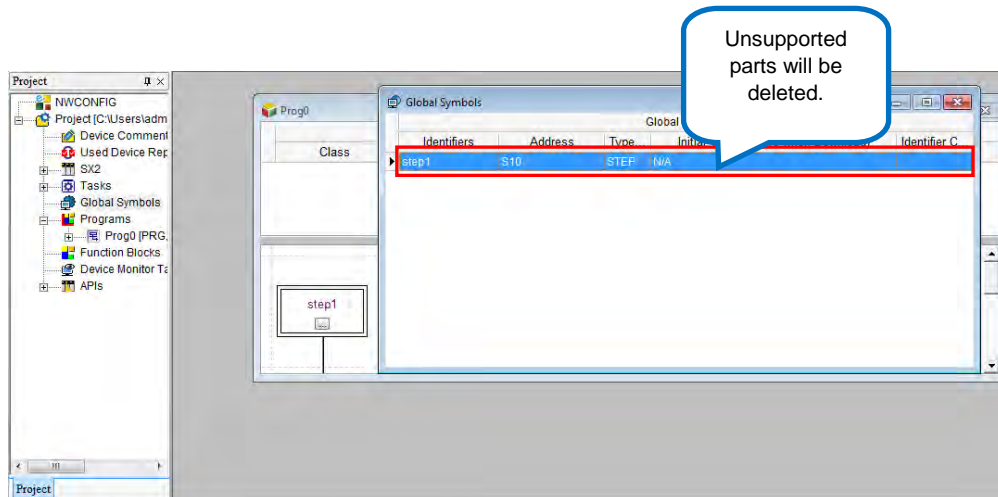
Message

- Error at [Prog0]. Local Symbols. Error at row (1). Identifier is the same with system's reserved word. Error Command : JOG
- Error at [Prog0]. Local Symbols. Error at row (2). Identifier is the same with system's reserved word. Error Command : w1

Suggestion


Identifier is the same with system's reserved word.

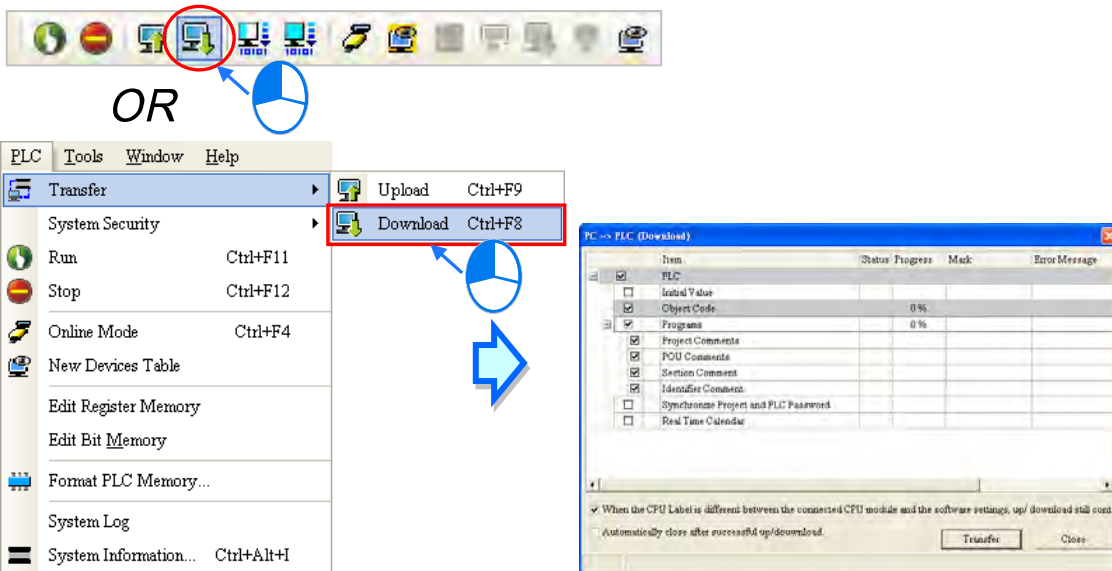
The programming languages such as LD and SFC are supported by the conversion of DVP series and AS series. Thus these two programming languages can be used without problem during conversion. When any unsupported programming language is used, the unsupported part will be deleted, after the conversion is done. For instance, a STEP symbol is declared in DVP series but this symbol will be deleted after the conversion, since AS series does not support this variable. If Kn is used in devices for DVP series, after the conversion, an error will occur, since AS series does not support Kn.




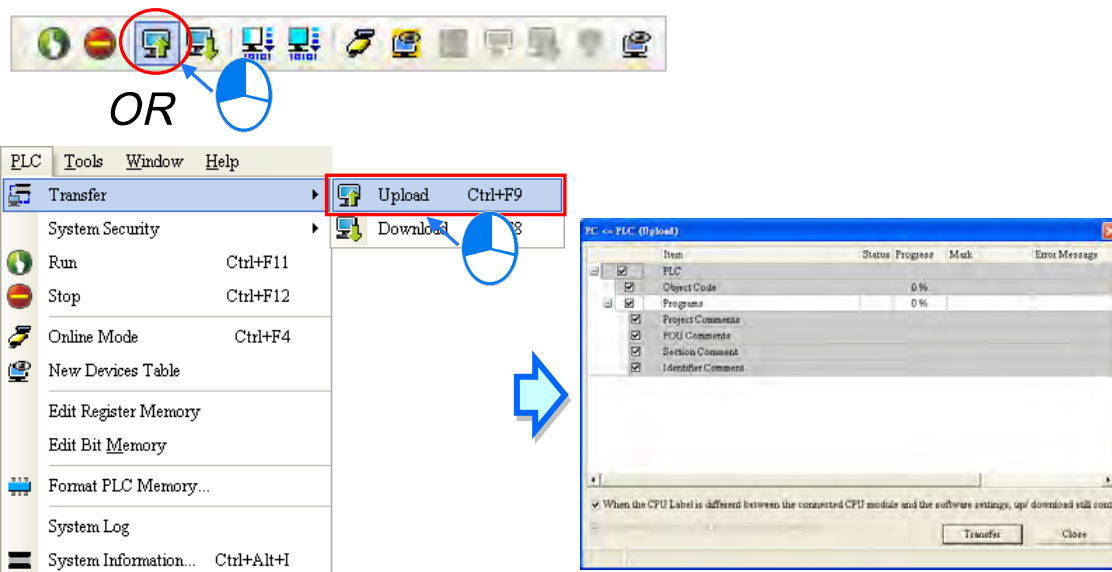
### 17.1.3 Download/Upload a Project

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**. Or click  on the toolbar. After that users will see a PC=>PLC (Download) window, select the items you'd like to download and then click **Transfer**.



If users want to download the project, they can click the **PLC** menu, point to Transfer, and click **Upload**. Or click  on the toolbar. After that users will see a PC=>PLC (Upload) window, select the items you'd like to upload and then click **Transfer**.



17



The items in the upload and download windows vary according to different series. The descriptions of the items in the upload/download window are as follows.

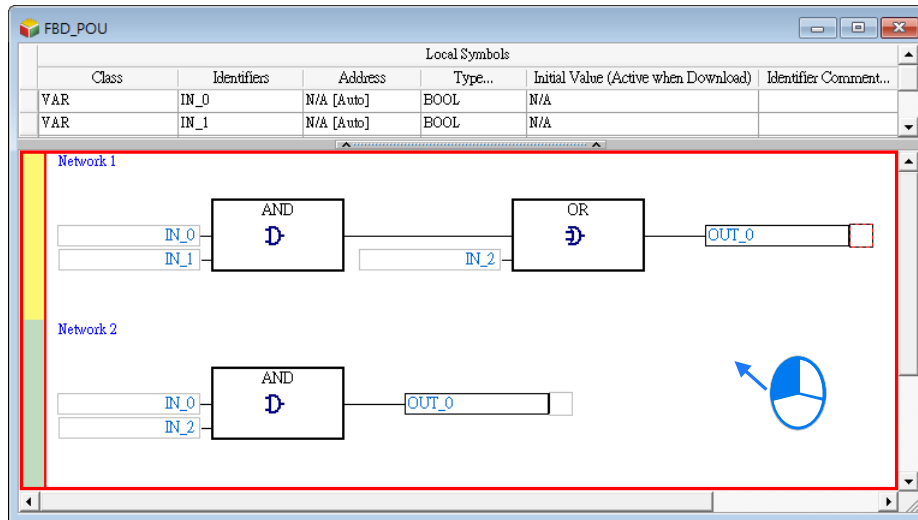
Transmit Item	Description	AH/AS		DVP		DVPxxMC	
		Upload	Download	Upload	Download	Upload	Download
Execution code	Program compiled PLC execution code	V	V	V	V	V	V
Program	Users create project program	V	V	V	V	V	V
Project Comment	Comments for projects	V	V	V	V	V	V
POU Comment	POU and FB comments	V	V	V	V	V	V
Segment Comment	POU and FB program Segment Comment	V	V	V	V	V	V
Symbol Comment	Comments regarding global and local symbols	V	V	V	V	V	V
Synchronize project and PLC passwords	Synchronize project password to PLC host password		V		V		V
Task Parameter	Setting task parameters					V	V
Retention Range (power failure)	Setting value for retention range (power failure)				Do Not Support ES3		
Retention Symbol Table (power failure)	Retention symbol for power failure listed in the table					V	V
CANopen Parameter	Setting					V	V

Transmit Item	Description	AH/AS		DVP		DVPxxMC	
		Upload	Download	Upload	Download	Upload	Download
	CANopen Builder Value						
Symbol Initial Value	Write the symbol initial value in the corresponding hosts memory		<b>V</b>		<b>V</b>		
Calendar	Write in the Date/Time settings in PLC		<b>V</b>		<b>V</b>		<b>V</b>
Position planning Table	Content of position planning table	Support AS	Support AS	Support ES3	Support ES3		
Axis Parameter	Axis parameter content	Support AHxxEMC	Support AHxxEMC			<b>V</b>	<b>V</b>
Ethernet/IP Symbol Table	Ethernet/IP symbol table content		Support AS		Support ES3		
E-CAM	E-CAM content	Support AS &AHxxEMC	Support AS &AHxxEMC	Support ES3		<b>V</b>	<b>V</b>

### 17.1.4 Find/Replace in LD/FBD/ CFC

The **Find/Replace** 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 / function block diagram / continuous function chart can be created.

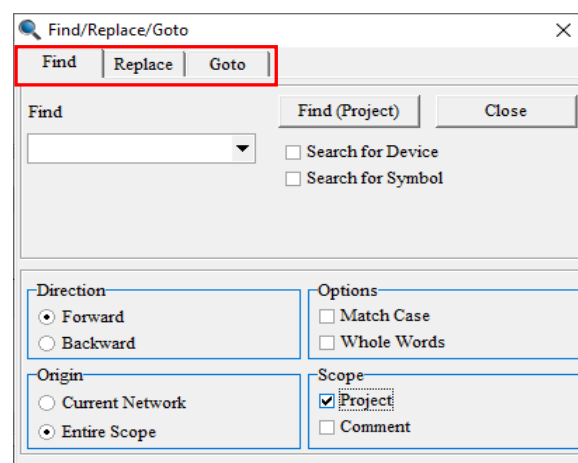
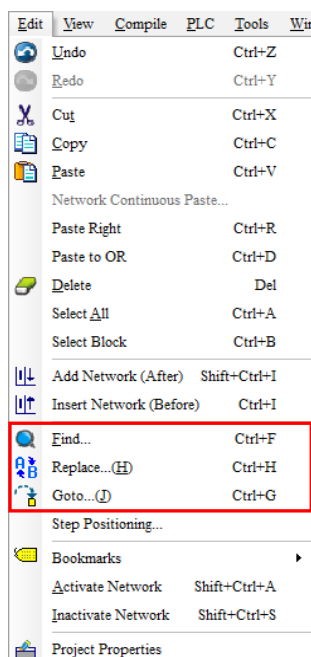
1. Open a program editing window, and click anywhere in the working area.



2. Find **Edit** on the toolbar and after clicking **Edit**, you can see the options **Find**, **Replace** and **Goto**. You can either select **Find**, **Replace** and **Goto**, the system directs you to the same setting page as the image shown below. You can also use the icons on the toolbar for the functions of **Find**, **Replace** and **Goto**.

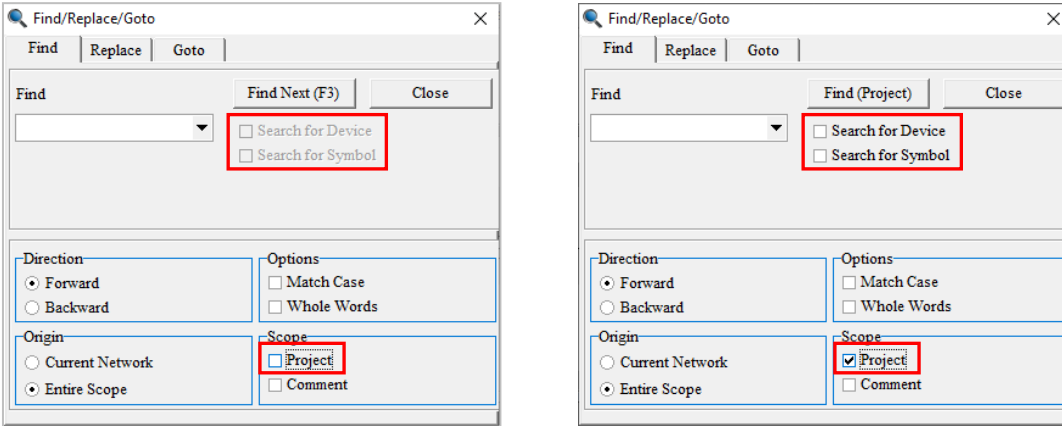



OR



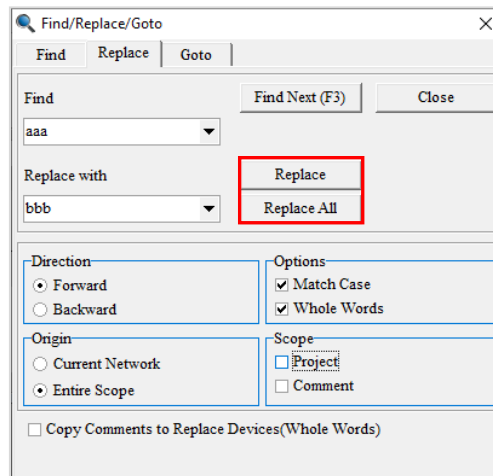
17

Type the object you'd like to search 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 even when the **Find/Replace/Goto** window is closed. When not defining the Scope, the search will only focus on the current editing. When **Project** is selected in the Scope section, you can search for the device or the symbols.

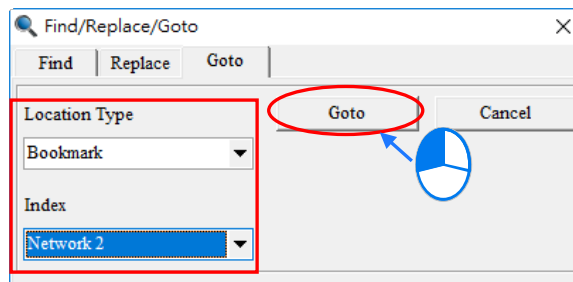


Search condition	Description
<b>Find</b>	<p>Users can type the object which will be searched for, or select an object which has been searched for after they click  in the box.</p> <p><b>Search for device:</b> searching for all devices which are matched</p> <p><b>Search for symbol:</b> searching for all symbols which are matched</p>
<b>Direction</b>	Users can search down or search up.
<b>Options</b>	<p><b>Match Case:</b> ISPSOft searches only for words that match the case of the word that users type in the <b>Find</b> box.</p> <p><b>Whole Words:</b> <b>Whole Words</b> instructs ISPSOft to find complete words only, and not to find words that only contain what users type. If the option is selected, users can further select the “<b>Copy Comments to Replace Devices (Whole Words)</b>” to copy the source device comments to the target device comments.</p>
<b>Origin</b>	<p><b>Current Network:</b> The search starts at the position selected.</p> <p><b>Entire Scope:</b> The search starts at the beginning of the program or the end of the program. This is the only option for CFC.</p>
<b>Scope</b>	<p>Users can define the searching ranges</p> <p>When either <b>Project</b> or <b>Comment</b> is selected, you can search for devices or symbols.</p>

Type the object you'd like to be replaced with in the field of **Find** and type the new object you'd like to use in the field of **Replace With**. And that select **Replace** to replace the next matched object or select **Replace all** to replace all the matched objects.



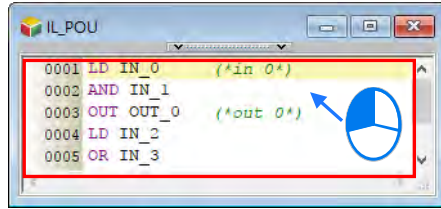
If the users want to go to a specific position in the LD and FBD program, they can click the **Goto** tab in the **Find/Replace/Goto** window, select **Network Label** or **Bookmark** in the **Location Type** drop-down list box, select an item in the **Index** drop-down list box, and click **Goto**.




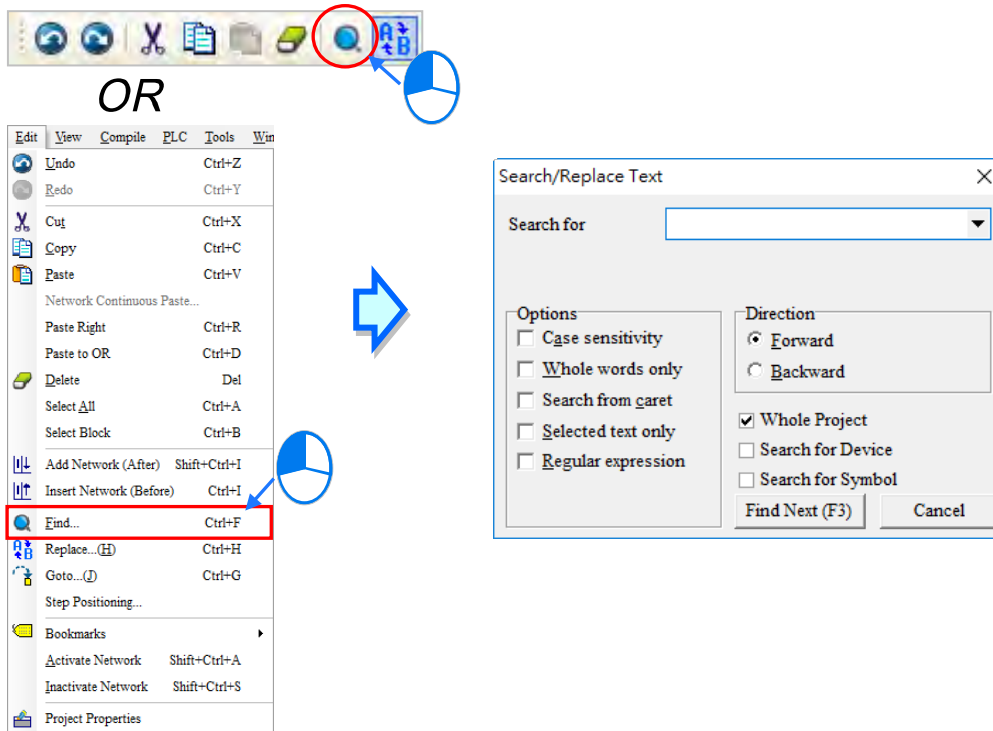
### 17.1.5 Find/Replace in IL/ST


Use the **Find** and the **Replace** functions in IL/ST.

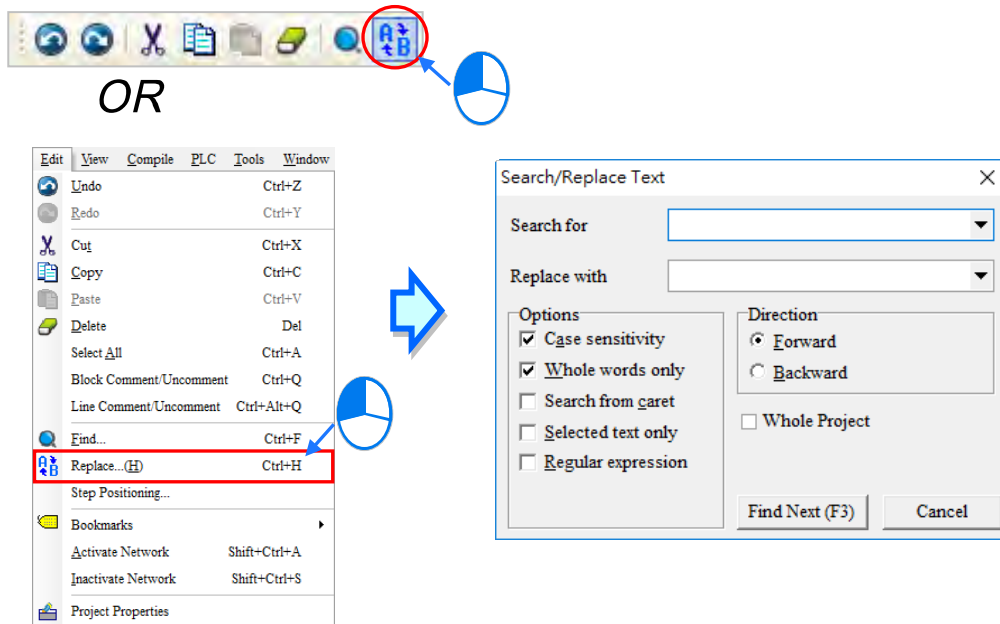
1. Open a program editing window, and click anywhere in the working area.



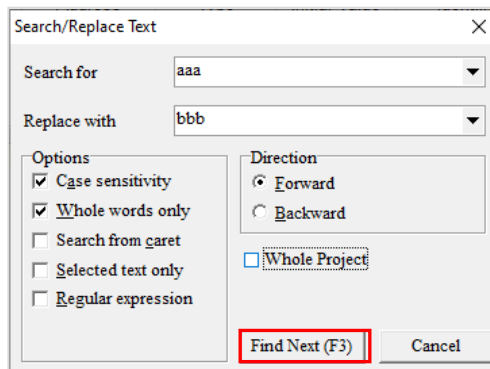
2. You can use the icon  on the toolbar for the functions of **Find**. Or you can click **Edit** on the toolbar to see the option **Find**. And after clicking the mentioned icon or option, the **Search/Replace Text** window appears.




3. After the users click **Replace** on the **Edit** menu, or  on the toolbar, the **Search/Replace Text** window will appear.



4. Type the object you'd like to be replaced with in the field of **Search for** and type the new object you'd like to use in the field of **Replace With**. You can also select options as the searching conditions. And then click **Find Next (F3)** to replace the next matched object.



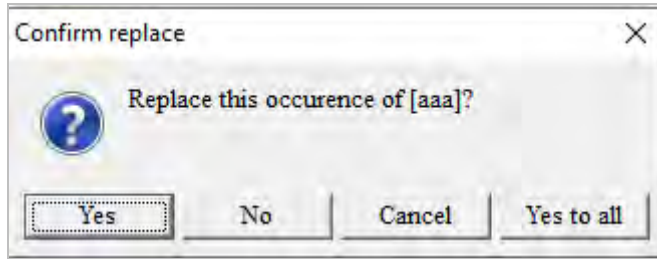
17

Search condition	Description
<b>Search for</b>	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.
<b>Whole project</b>	Once selected, you can search for devices or symbols.
<b>Search for device</b>	Searching for all devices which are matched
<b>Search for symbol</b>	Searching for all symbols which are matched
<b>Case sensitivity</b>	ISPSOft searches only for words that match the case of the word that users type in the <b>Search for</b> box.
<b>Whole words only</b>	<b>Whole words only</b> instructs ISPSOft to find complete words only, and not

	to find words that only contain what users type.
<b>Search from caret</b>	The search starts at the text cursor. (If this checkbox is not selected, the system will search the whole program.)
<b>Selected text only</b>	The search starts at the text selected. (If this checkbox is not selected, the system will search the whole program.)
<b>Regular expression</b>	After this checkbox is selected, regular expressions can be used.
<b>Direction</b>	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 appears after an matched object is found.



Button	Function
<b>Yes</b>	The object found is replaced by the replacement object, and the system searches for the next occurrence of the object typed in the <b>Search for</b> box.
<b>No</b>	The object found is not replaced by the replacement object, and the system searches for the next occurrence of the object typed in the <b>Search for</b> box.
<b>Cancel</b>	The search is canceled, and the <b>Confirm replace</b> window is closed,
<b>Yes to all</b>	All the objects meeting the search conditions selected are replaced by the replacement object.

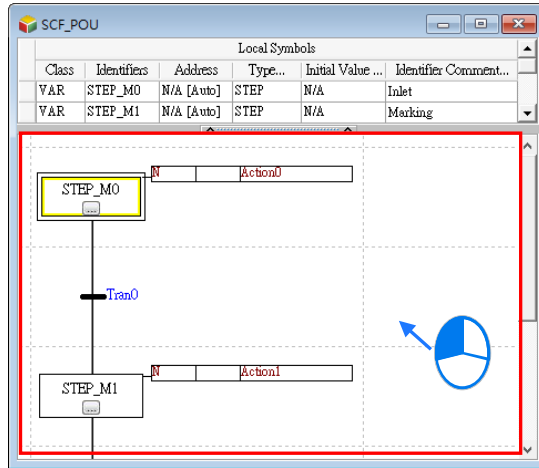
17





### 17.1.6 Find/Replace in SFC

Use the **Find** and the **Replace** functions in a sequential function chart (SFC).

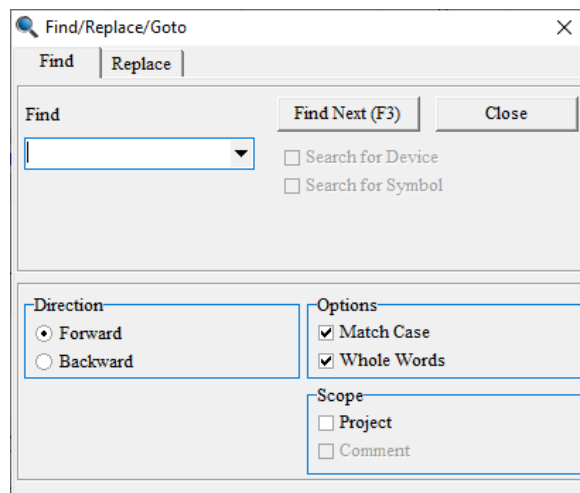
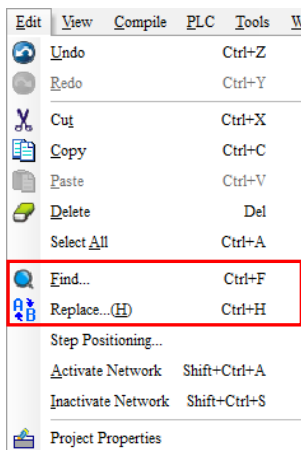
1. Open a program editing window, and click anywhere in the working area.



2. You can use the icons  or  on the toolbar for the functions of **Find** and **Replace**. Or you can click **Edit** on the toolbar to see the options **Find** and **Replace** and after clicking the mentioned icon or option, the **Find/Replace/Goto** window appears.

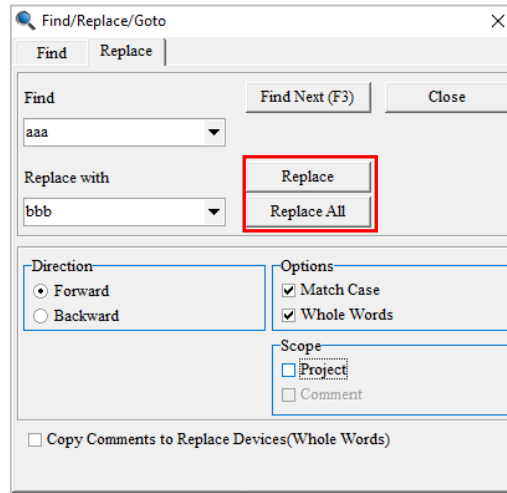



OR



17

3. Type the object you'd like to be replaced with in the field of **Find** and type the new object you'd like to use in the field of **Replace With**. After that select **Replace** to replace the next matched object or select **Replace all** to replace all the matched objects.



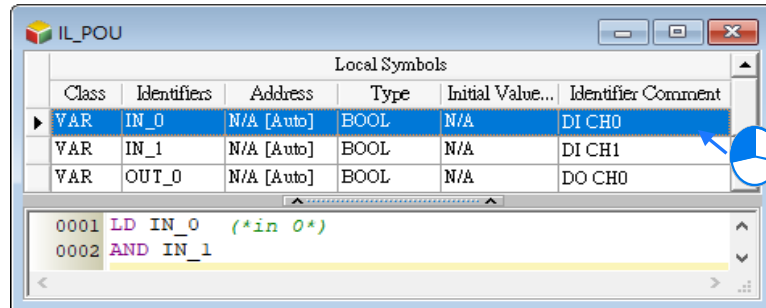
Search condition	Description
<b>Find</b>	Users can type the object which will be searched for, or select an object which has been searched for after they click  in the box. Search for device: searching for all devices which are matched Search for symbol: searching for all symbols which are matched
<b>Direction</b>	Users can search down or search up.
<b>Options</b>	<b>Match Case:</b> ISPSOft searches only for words that match the case of the word that users type in the <b>Find</b> box. <b>Whole Words:</b> <b>Whole Words</b> instructs ISPSOft to find complete words only, and not to find words that only contain what users type. If the option is selected, users can further select the <b>“Copy Comments to Replace Devices (Whole Words)”</b> to copy the source device comments to the target device comments.
<b>Scope</b>	Users can select ranges which will be searched. When either <b>Project</b> or <b>Comment</b> is selected, you can search for devices or symbols.


17

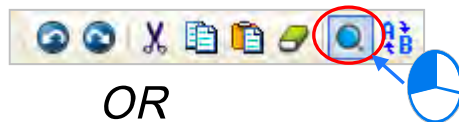
### 17.1.7 Find Function in Symbol Table

Use the **Find** function in a local symbol table/global symbol table.

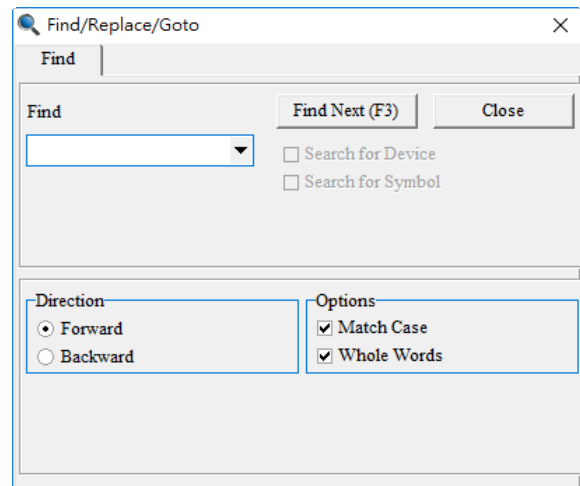
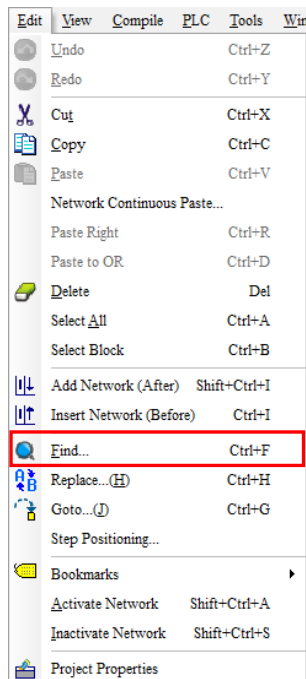
1. Open a program editing window, and click anywhere in the working area.




2. You can use the icons  on the toolbar for the functions of **Find**. Or you can click **Edit** on the toolbar to see the option **Find** and after clicking the mentioned icon or option, the **Find/Replace/Goto** window appears.



OR

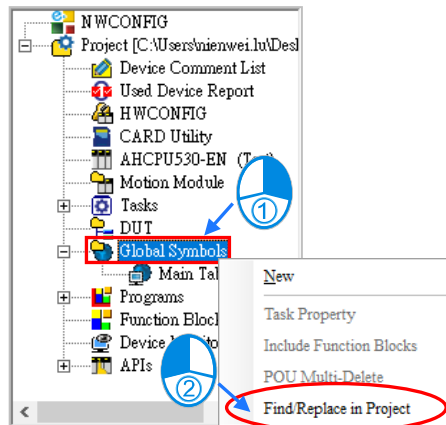


3. Type the object you'd like to search for and select search conditions. The users can do a search on the object typed in the **Find** box by clicking **Find Next (F3)** or pressing F3 on the keyboard. The users can still do a search even when the **Find/Replace/Goto** window is closed.

Search condition	Description
<b>Find</b>	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.
<b>Direction</b>	Users can search down or search up.
<b>Options</b>	<p><b>Match Case:</b> ISPSOft searches only for words that match the case of the word that users type in the <b>Find</b> box.</p> <p><b>Whole Words: Whole Words</b> instructs ISPSOft to find complete words only, and not to find words that only contain what users type.</p>

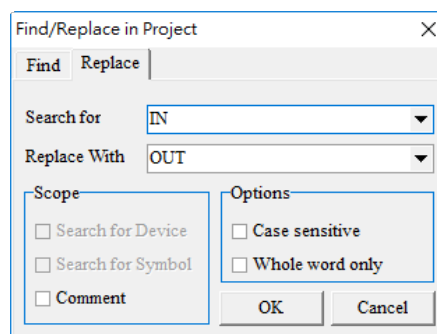
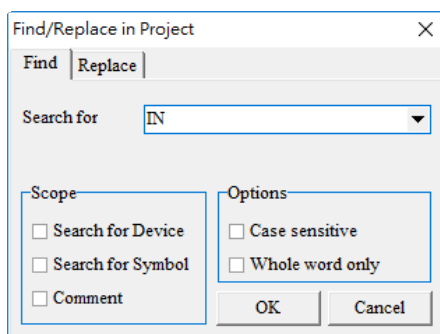
### 17.1.8 Find/Replace in Project

- Right-click the Global Symbols/Programs/Function Blocks in the project management area, and then click **Find/Replace in Project** on the context menu to find device or symbol in a project.



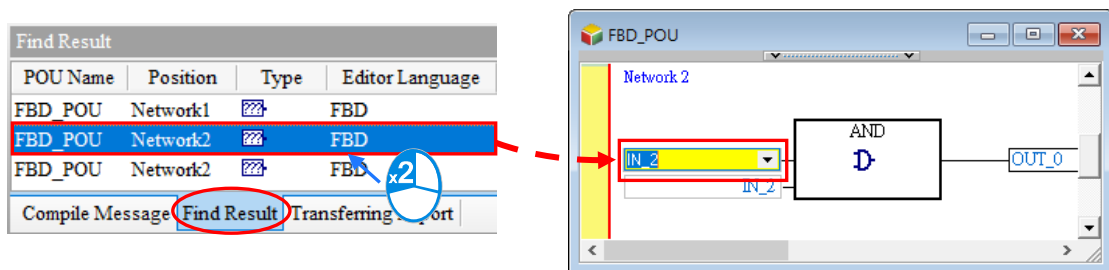
17

- The **Find** page and the **Replace** page are shown below. There are two tabs in the **Find/Replace in Project** window. Type the object you'd like to search for in the field of **Search for** on the Find tab page and select search conditions. Click **OK** to start searching. Type the object you'd like to be replaced with in the field of **Search for** and a new object you'd like to use in the field of **Replace With**. After that click **OK** to replace the next matched object. Note this act (replacement) can NOT be undone.



Search condition	Description
<b>Scope</b>	Search for device: searching for all devices which are matched Search for symbol: searching for all symbols which are matched Comment: searching for all the comments which are matched
<b>Options</b>	<b>Match Case:</b> ISPSOft searches only for words that match the case of the word that users type in the <b>Find</b> box. <b>Whole Words: Whole Words</b> instructs ISPSOft to find complete words only, and not to find words that only contain what users type.

After using 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 direct the users to the position of the object corresponding to the search result.

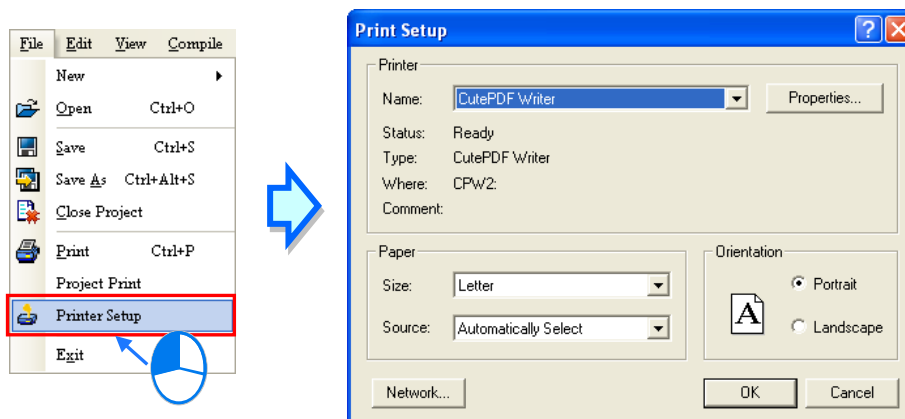


### 17.1.9 Print Function

ISPSOft provides a printing function.

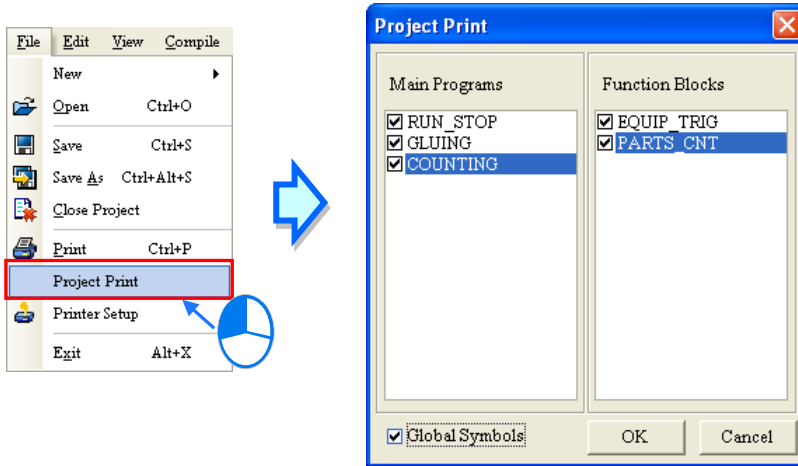
- **Printer Setup**

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.

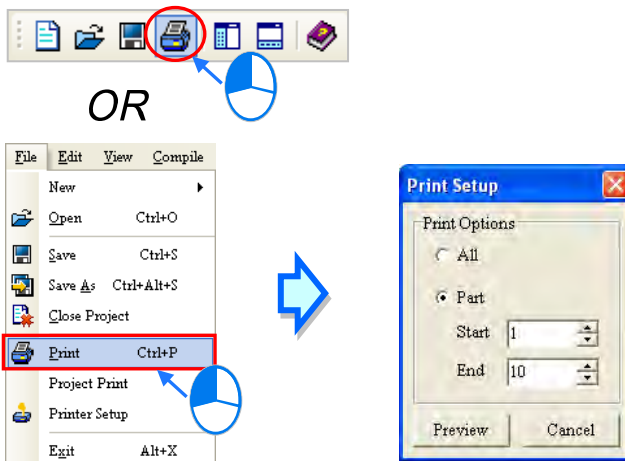
Declaration Type	Identifiers	Address	Type	Initial Value
VAR	ET_INS_1	N/A [Auto]	EQUIP_TRIG	N/A
VAR	ET_INS_2	N/A [Auto]	EQUIP_TRIG	N/A

17

● **Print**

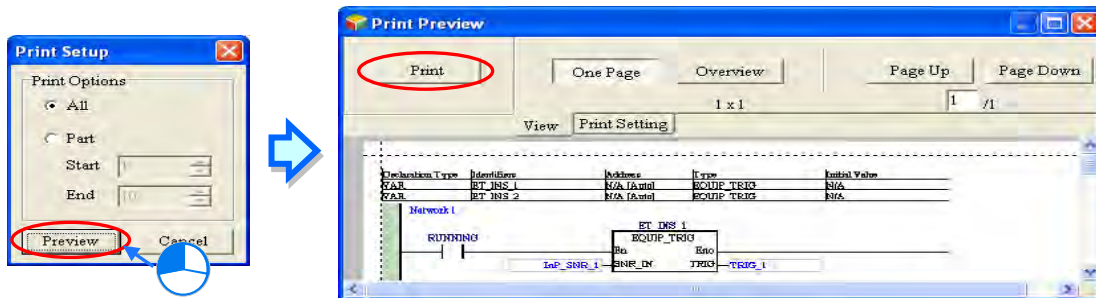
If users want to print the contents of the present window, they can click **Print** on the **File** menu or  on the toolbar. The users can set a print range in the **Print Setup** window. The print result varies with the contents of the present window. Please refer to the table below for more information.



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.
Continuous function chart	The complete program and local symbol table are printed.	Users can not print a part of the continuous function chart.

\*.After users open the window for an action/transition, they can print the action/transition.

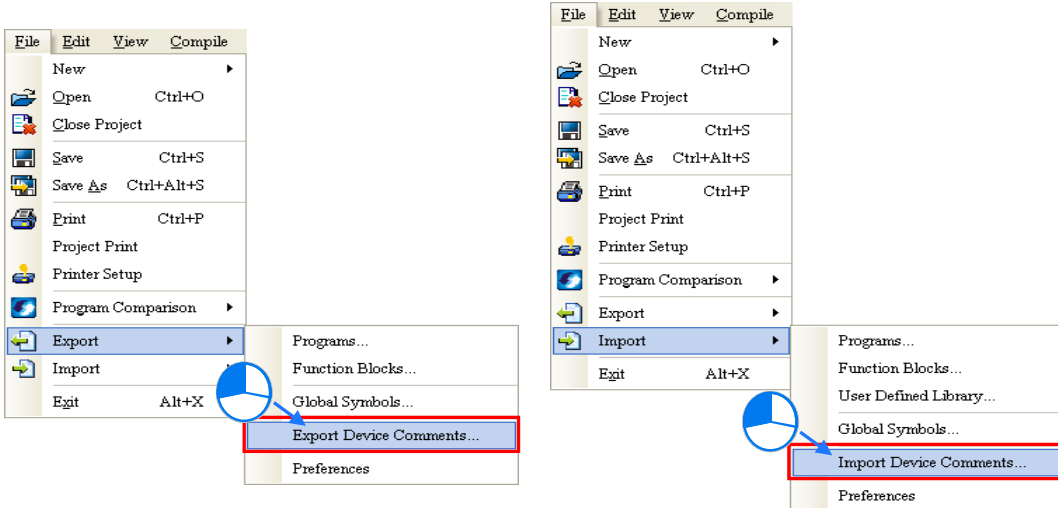
After the users click **Preview** in the **Print Setup** window, the **Print Preview** window will be opened. Please refer to appendix C for more information about the print management tool that ISPSOft provides.



## 17.2 Devices and Registers

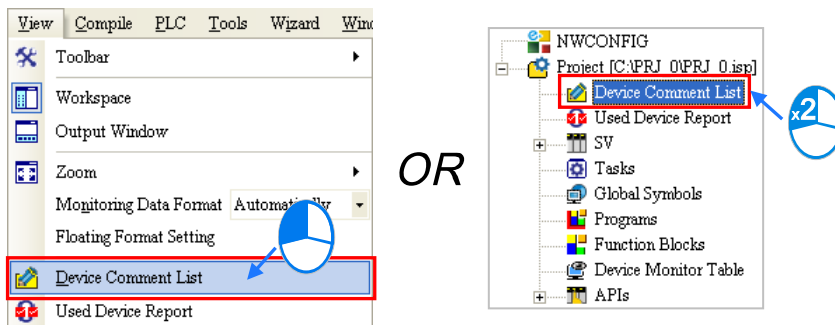
### 17.2.1 Device Comment List for a DVP Series PLC

If users want to export the device comment list from a project for a DVP series PLC, they can click the **File** menu, point to **Export**, and click **Export Device Comments....** If the users want to import the device comment list, they can click the **File** menu, point to **Import**, and click **Import Device Comments....**



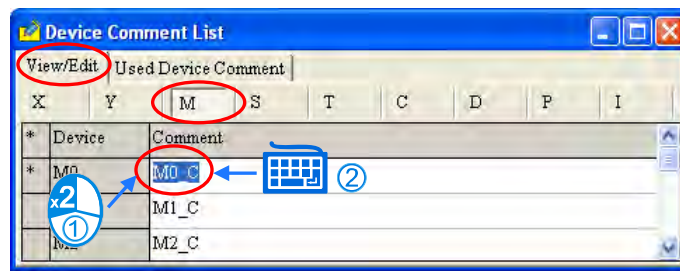
If the users want to manage the comments on devices, they can click **Device Comment List** on the **View** menu, or double-click **Device Comment List** in the project management area. After the users click **Device Comment List** on the **View** menu, or double-click **Device Comment List** in the project management area, the system will open the **Device Comment List** window.

17

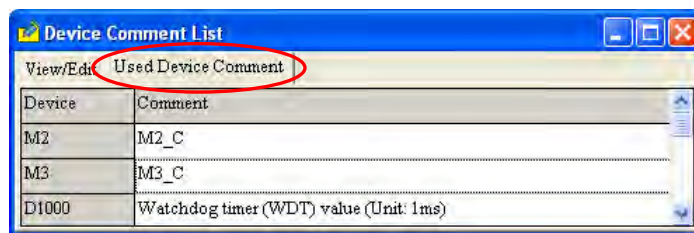


The comments on devices are on a **View/Edit** page. After the users click a tab which indicates a device type in the window, the comments on devices are displayed. The mark “\*” at the left side of a device indicates that the device is used in the program. If the users want to make a comment on a device, they can double-click the **Comment** cell for the device, and type the comment.



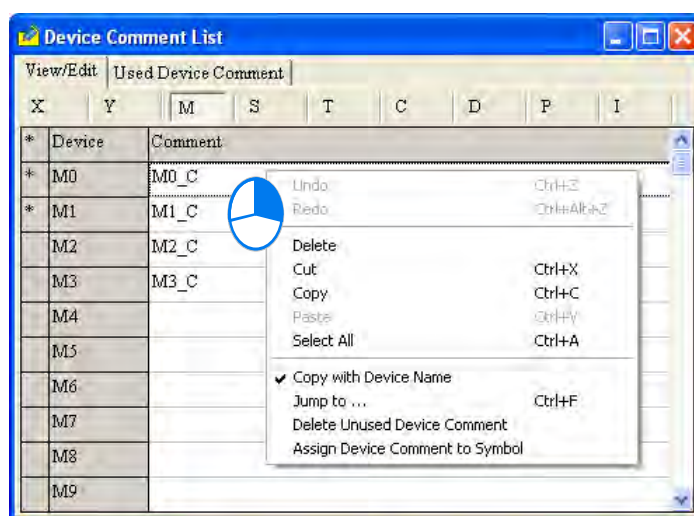


The devices on which comments are made and all the comments on the devices are on the **Used Device Comment** page. After the users double-click the **Comment** cell for a device, they can edit the comment on the device.



- \*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.

After the users 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...**

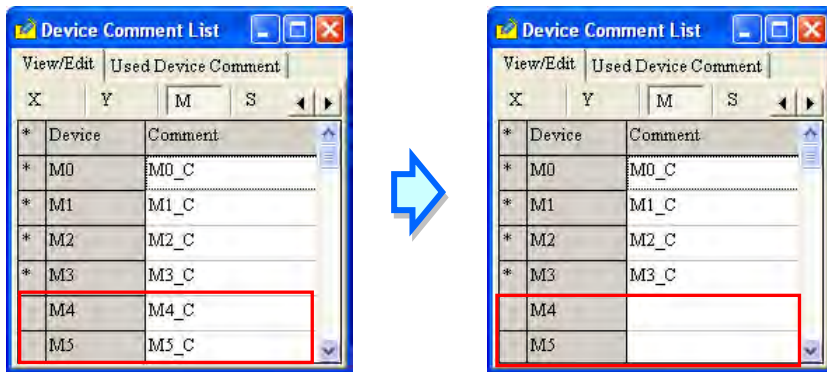
After the users click **Jump to...** on the context menu, the **Jump to...** window will appear. The users can type a device name in the **Device Name** box. After the users click **OK** in the **Jump to...** window, they can easily view the device name and the comment on the device on a **View/Edit** 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**

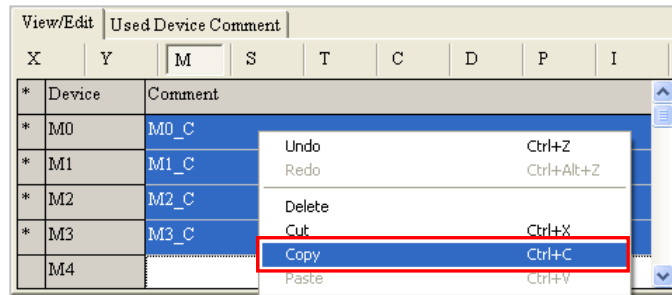
After the users click **Delete Unused Device Comment** on the context menu, the system will delete the comments on the devices which are not used in the program.



17

● **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.



Copy with Device Name

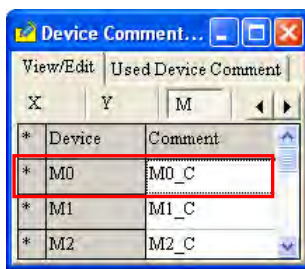
	A	B	C
1	M0	M0_C	
2	M1	M1_C	
3	M2	M2_C	
4	M3	M3_C	
5			

Copy with Device Name

	A	B
1	M0_C	
2	M1_C	
3	M2_C	
4	M3_C	
5		

● **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.



Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	VAR_0	M0	BOOL	FALSE	VAR_0C
VAR	VAR_1	N/A [Auto]	BOOL	FALSE	VAR_1C

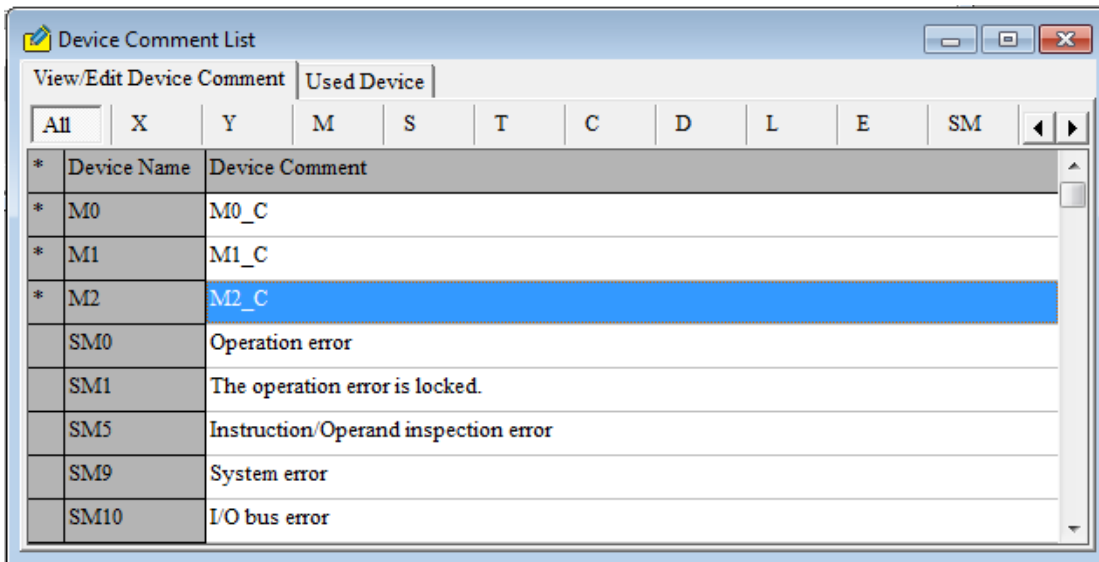
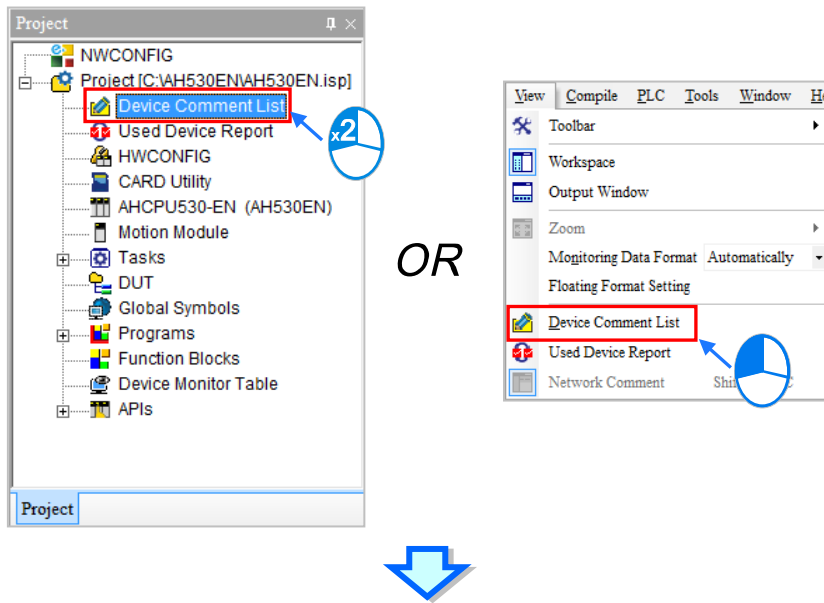


Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	VAR_0	M0	BOOL	FALSE	M0_C
VAR	VAR_1	N/A [Auto]	BOOL	FALSE	VAR_1C

### 17.2.2 AH/AS Series Device Comment List

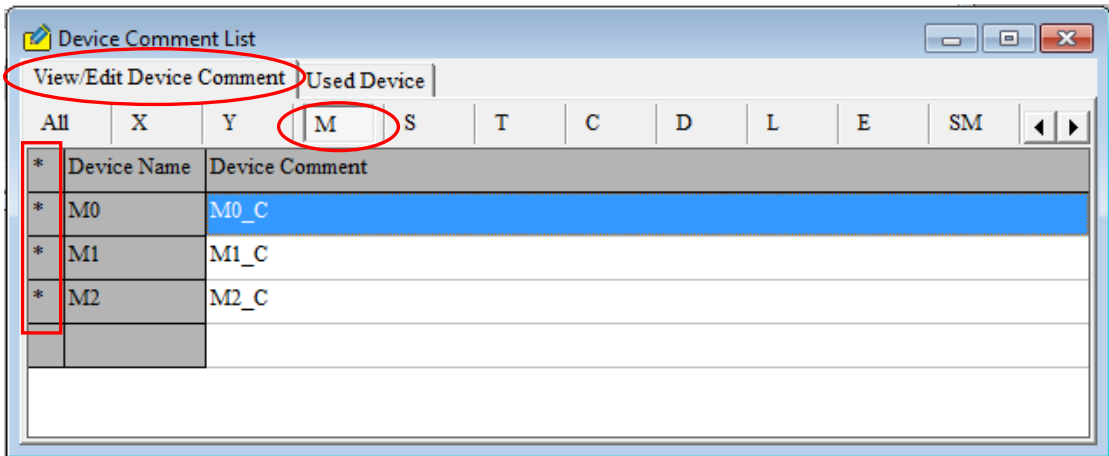
Users can manage the devices in an AH/AS series CPU module. After users click **Device Comment List** on the **View** menu, or double-click **Device Comment List** in the project management area, the **Device Comment List** window will be opened.

There are two tabs at the top of the **Device Comment & List** 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.

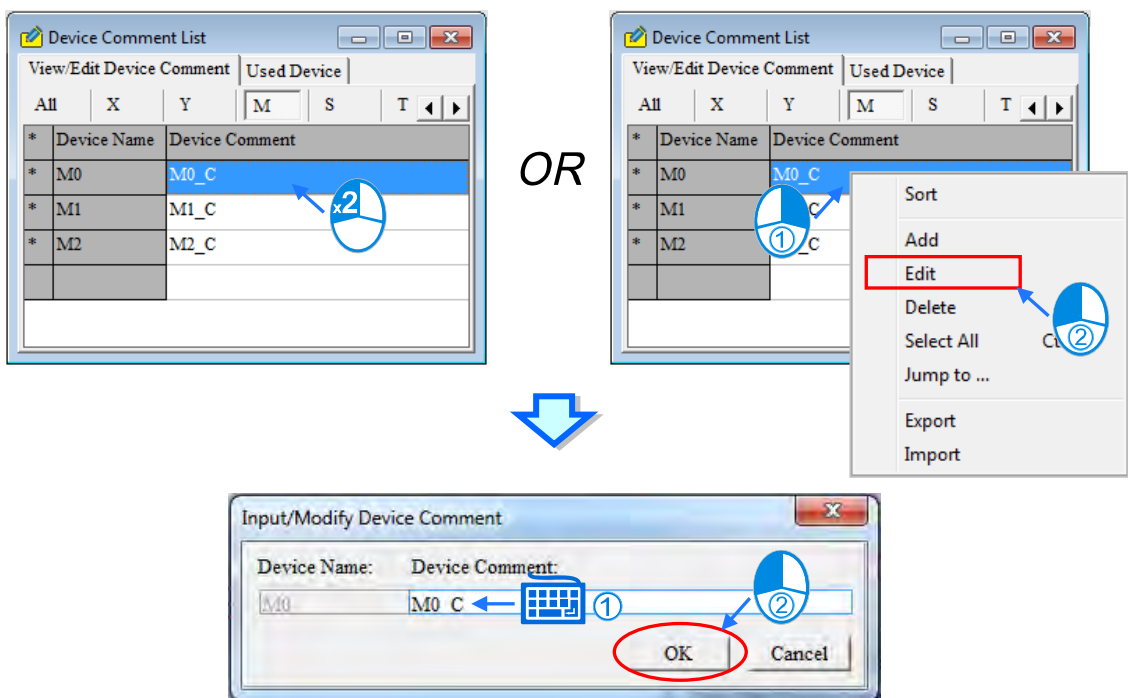


17

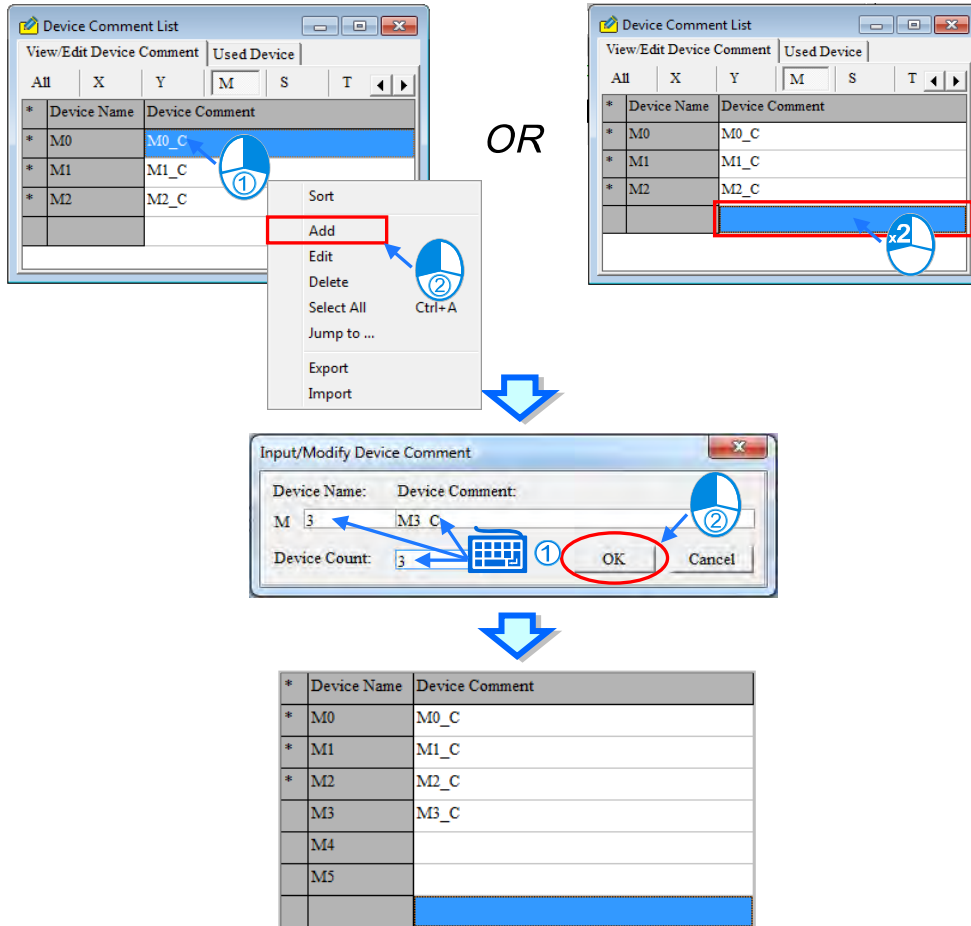
The **View/Edit Device Comment** page displays the device names with corresponding comments on the list. The tabs below the Comment indicate the device types. After the users click **All**, the devices on which comments are made are displayed. Besides, the mark “\*” at the left side of a device indicates that the device is used in the program.



The users can modify the comments on 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 or 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.

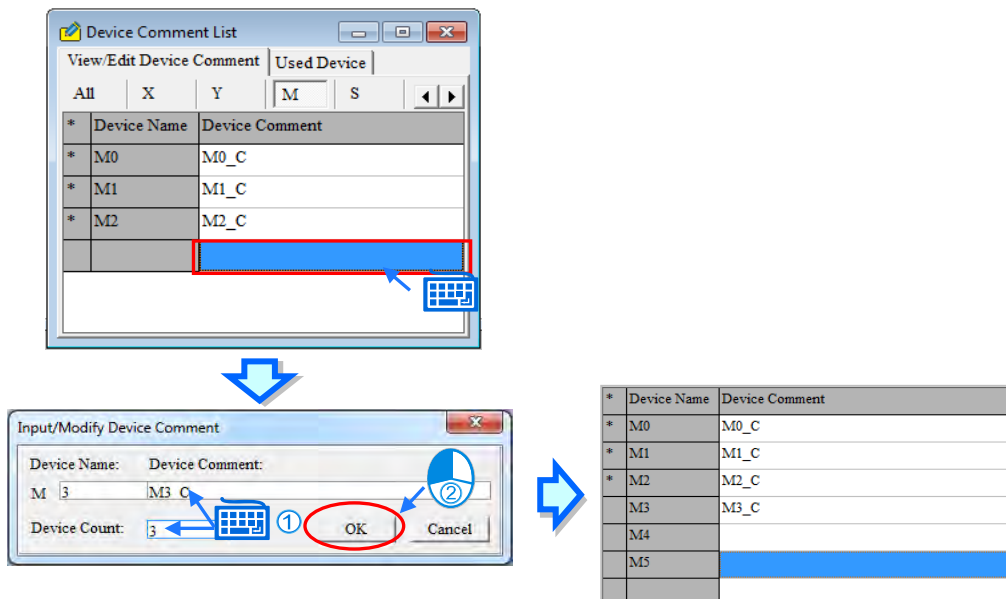


Right-click **Add** on the selected Device Comment from the Device Comment List window or double left-click the list and users can modify the comments shown in the Input/Modify Device Comment window. Users can input a value in the Device Count box to indicate the number of devices and corresponding comments as the image shown below.

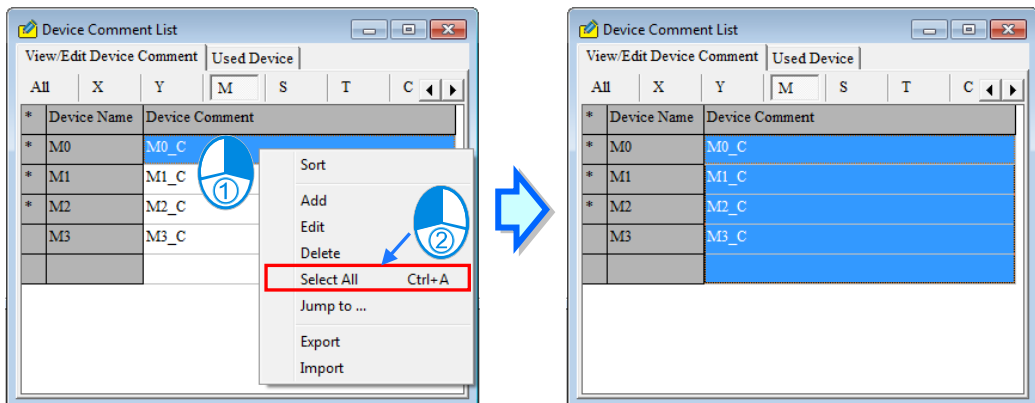


# 17

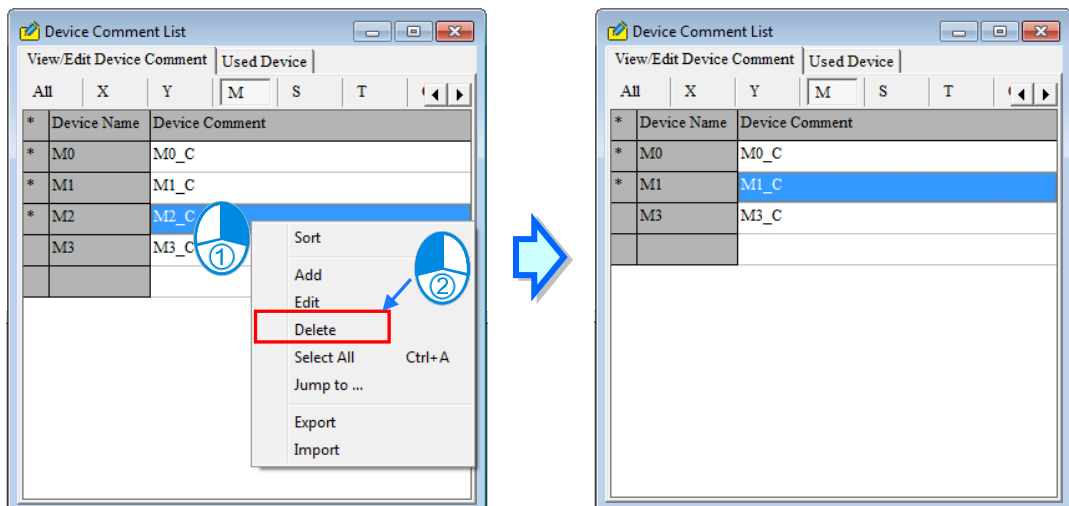
Users can also input the device location and then the Input/Modify Device Comment window will appear. Input the comment and the Device Count to complete the setting.



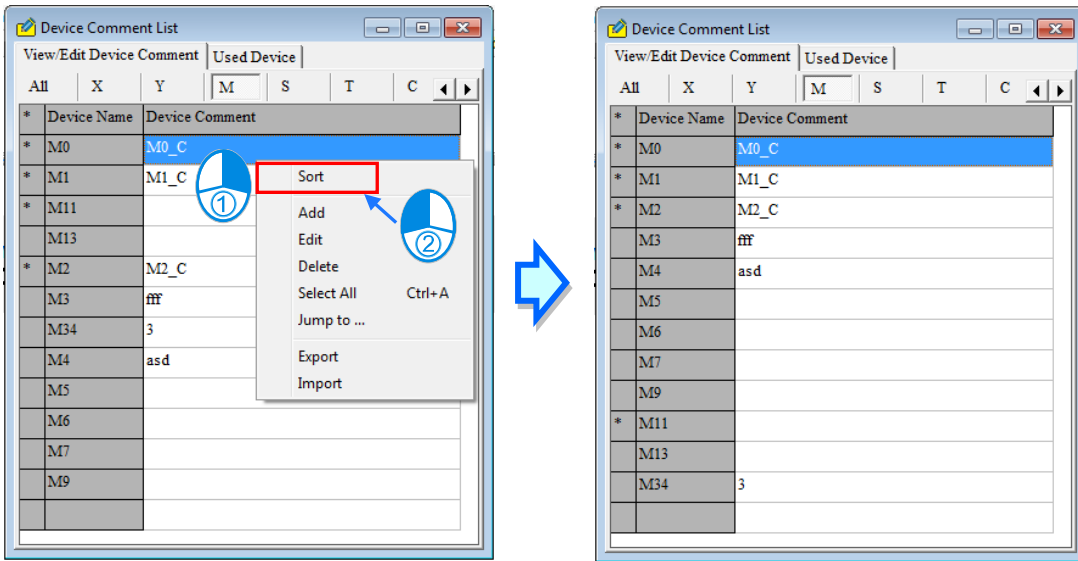
If the users want to delete the comment on a device on a **View/Edit Device Comment** page, they can select the comment, right-click the comment, and click **Delete** on the context menu. If the users want to delete a range of comments on a **View/Edit Device Comment** page, they can select the first 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.



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 is recovered to default value.



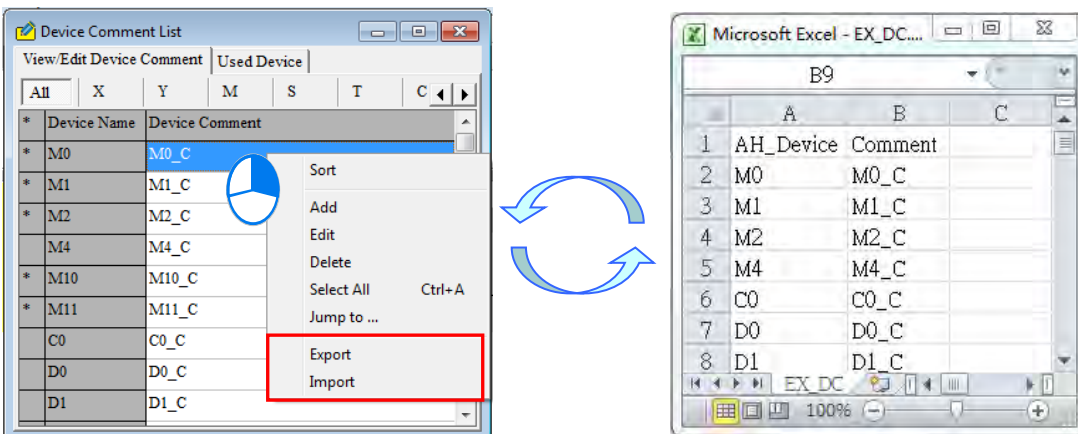
If the users want to rearrange the devices on a **View/Edit Device Comment** page, they can right-click the page, and click **Sort** on the context menu.



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 these 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.

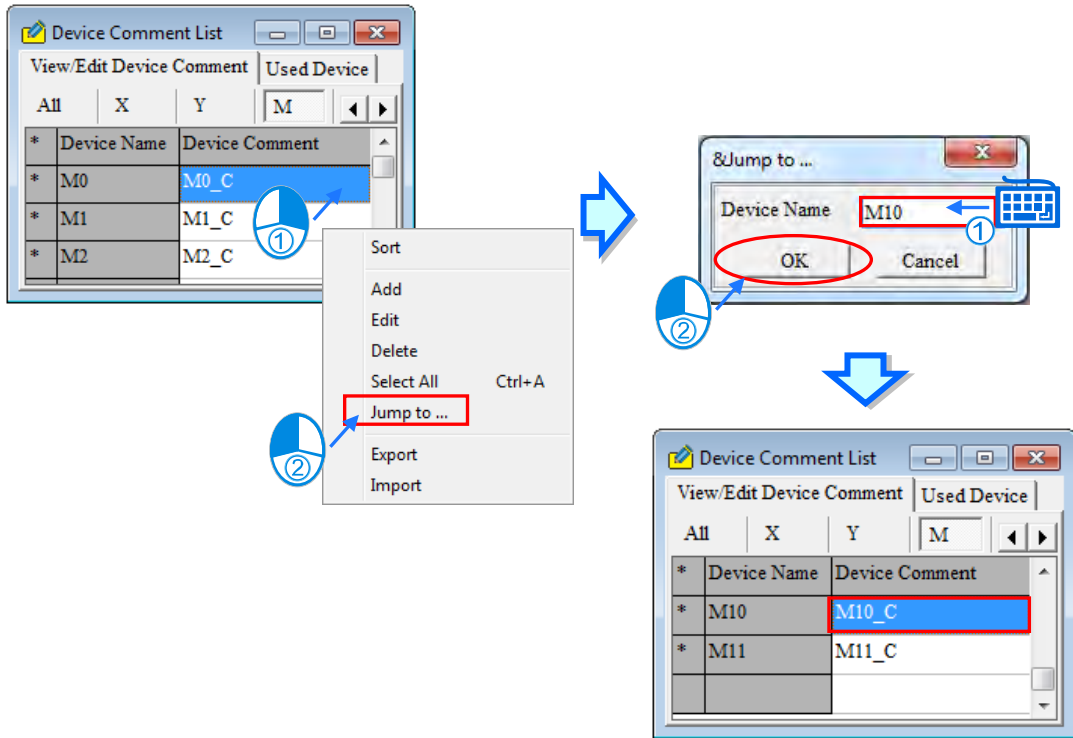
17

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**.

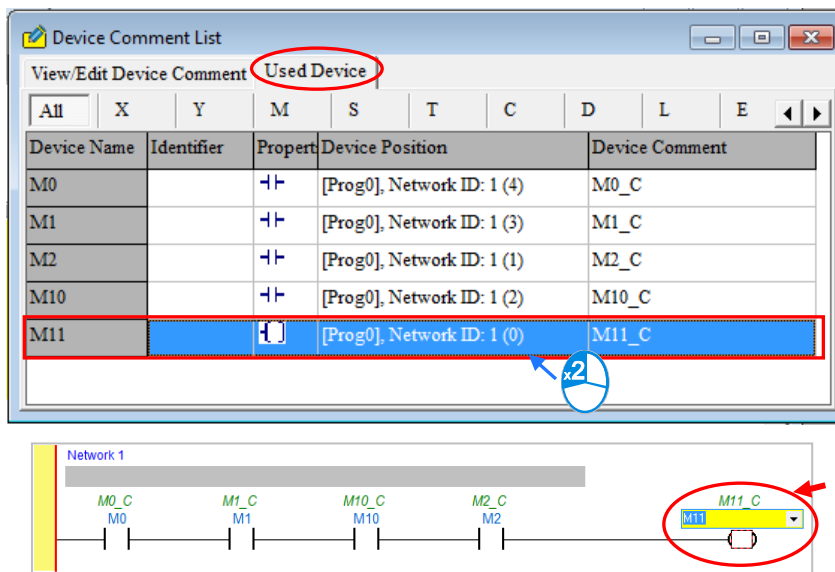




For a quick way to go to the specific device range, right-click on the Device Comment area to see the context menu and select “Jump to”. After that a Jump to window will show up. Type the device name and press OK. The table will display the required device range. Before typing the device name, you will need to make sure the device name you are going to type exists in the table. You can also use Jump to function in the Used Device tab.



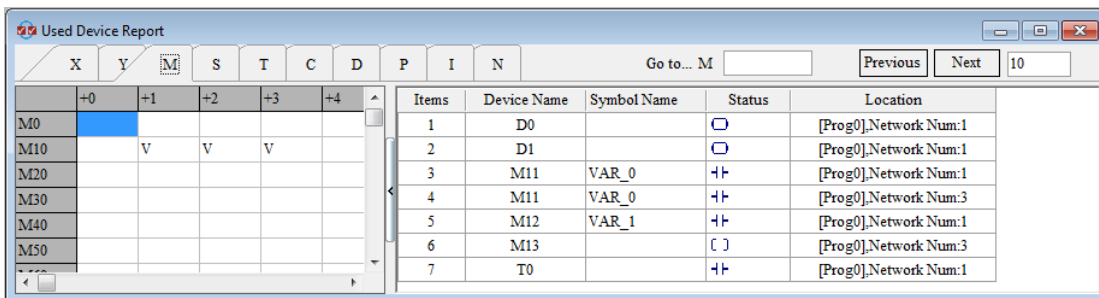
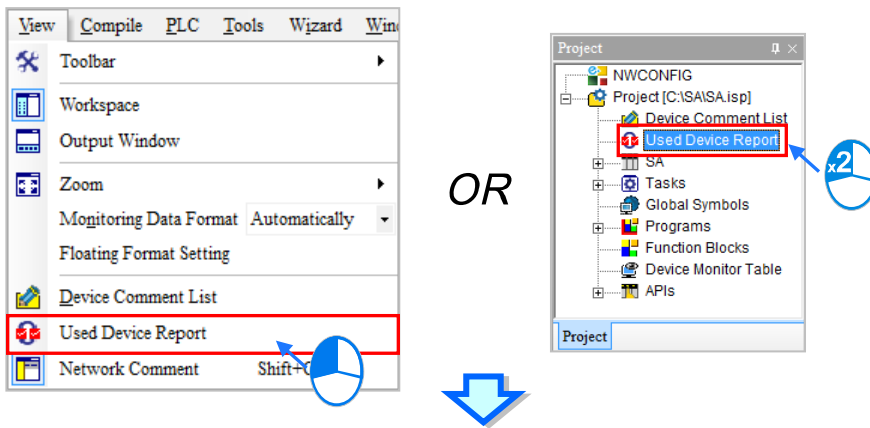
The devices used in the program and the information about the devices are listed in the **Used Device** page. The tabs below this page indicate device types. Besides, after the users double-click a row on the **Used Device** page, the system will lead the users to the corresponding device position of that row.



### 17.2.3 Using Device Report

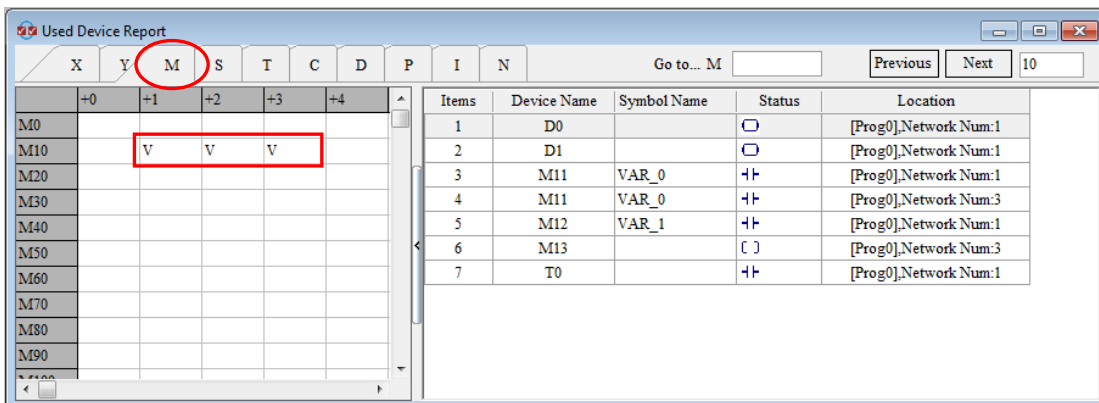
When users edit a project, they can refer to the report on the devices used in the project. However, the users have to compile the program in the project before they refer to the report on the devices used in the project for a DVP series PLC. Besides, if the users modify the program after the window 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.

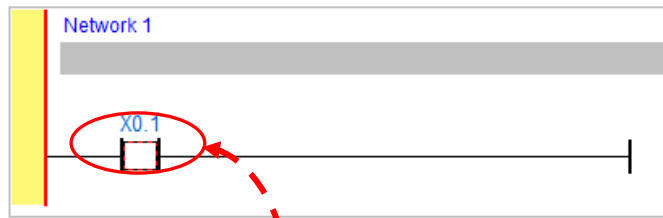


17

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 of devices indicate that the devices are used in the program. For example, there are checks in the cells for M11 M12 and M13.



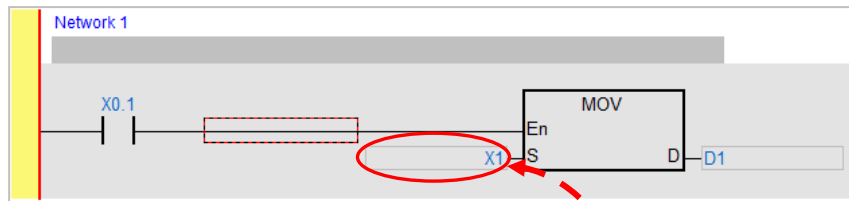
If users select AH/AS Series CPU modules and use any bits of the device or the device itself, that device box will be checked.



Used Device Report

	X	Y	M	S	T	C	D	HC	E	SM	SR	FR
				+0	+1	+2	+3					
X0	V											
X10												
X20												
X30												
X40												
X50												
X60												

Items	Device Name	Symbol Name
1	X0.1	

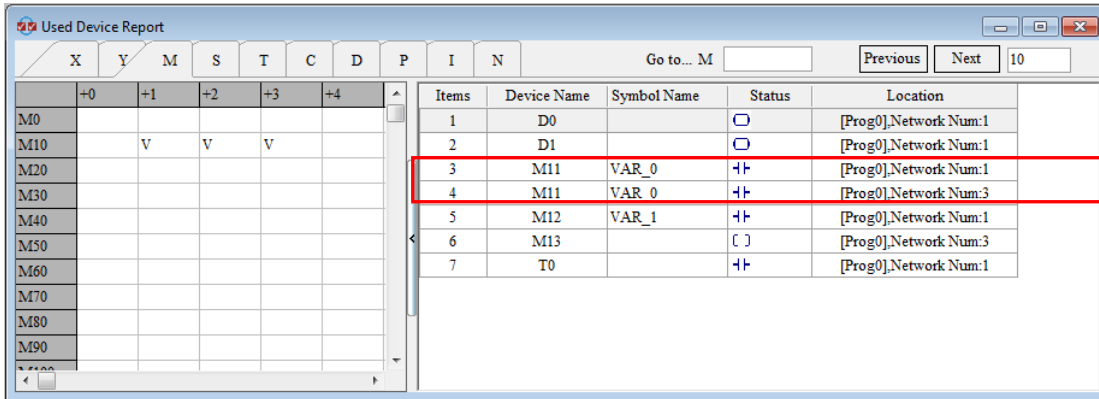


Used Device Report

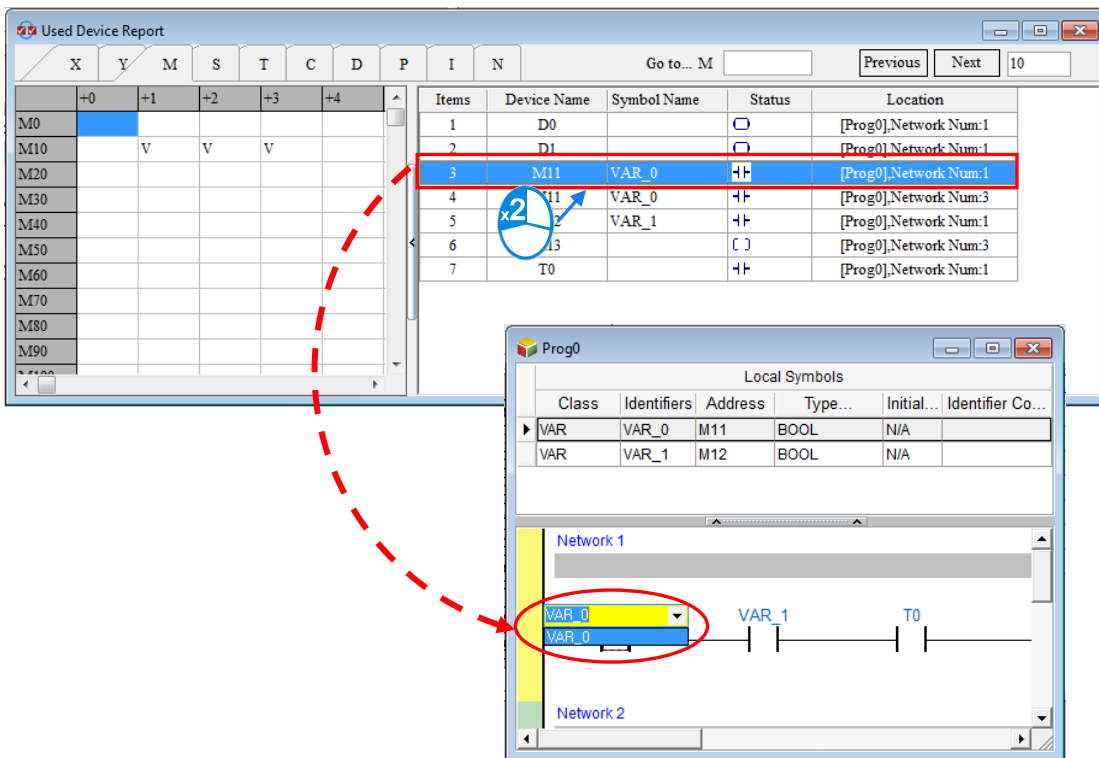
	X	Y	M	S	T	C	D	HC	E	SM	SR	FR
				+0	+1	+2	+3					
X0	V			V								
X10												
X20												

Items	Device Name	Symbol Name
1	D1	
2	X0.1	
3	X1	

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.

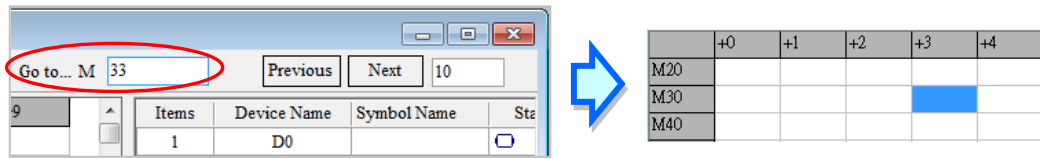


After the users double-click a row on the right part of the window, the system will lead the users to the corresponding device location.

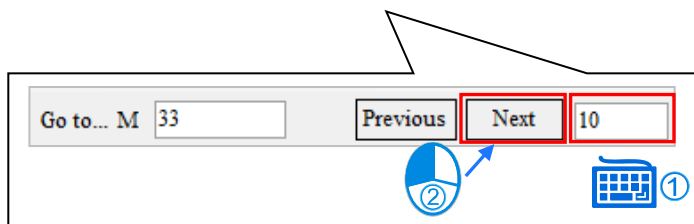
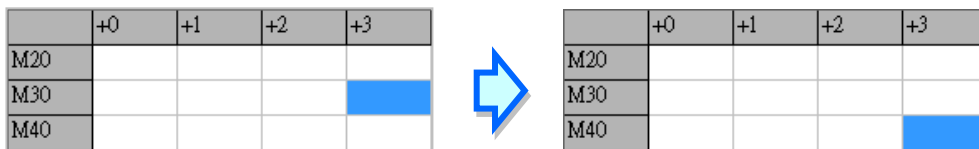
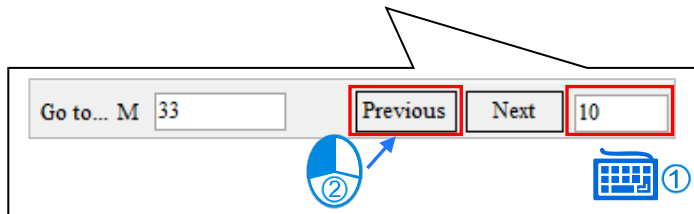
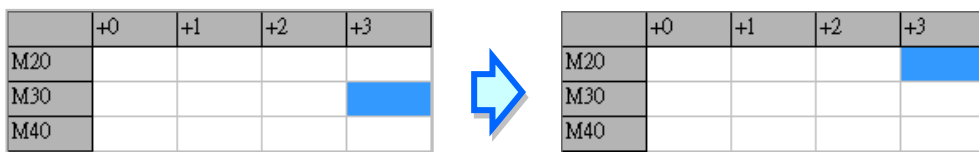


17

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 corresponding device position.

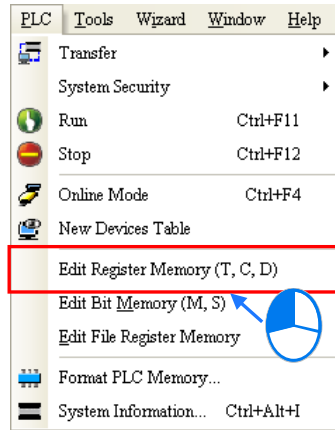


By executing the input number next to the **Previous/ Next** button for forward or backward movement, the blue cursor can move to the assigned device number.



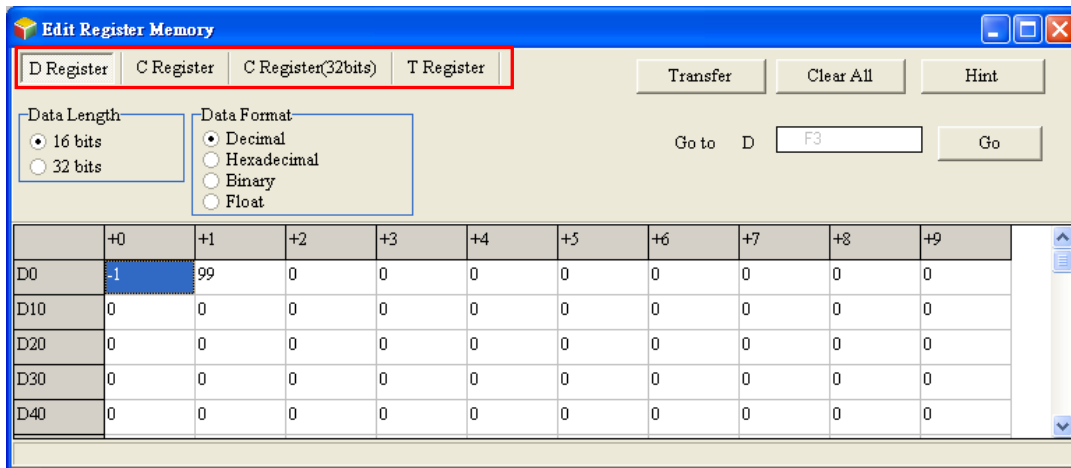
### 17.2.4 Edit Register Memory

Users can edit the values in the T/C/D/L devices in a batch. And the new values in these T/C/D/L devices can be saved and downloaded. After users create a project, they can click **Edit Register Memory** on the **PLC** menu to open the **Edit Register Memory** window.



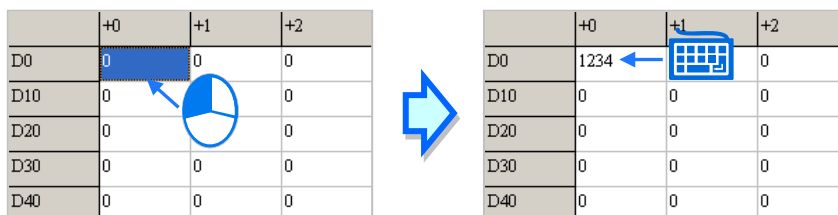
The values in the tables in the **Edit Register Memory** window are not the values uploaded from the PLC. They are the values saved last time. If the **Edit Register Memory** window in the project is opened for the first time, the default values in the tables are 0.

The tabs at the top of the **Edit Register Memory** window indicate device types.

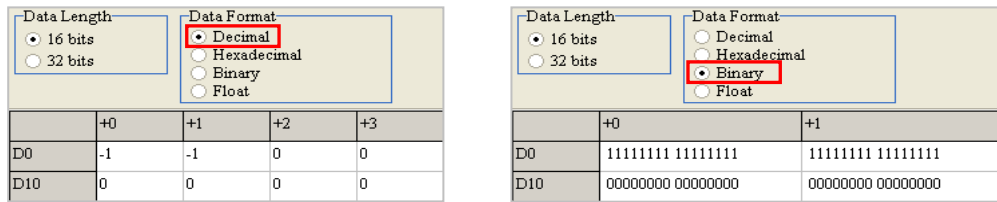


17

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.

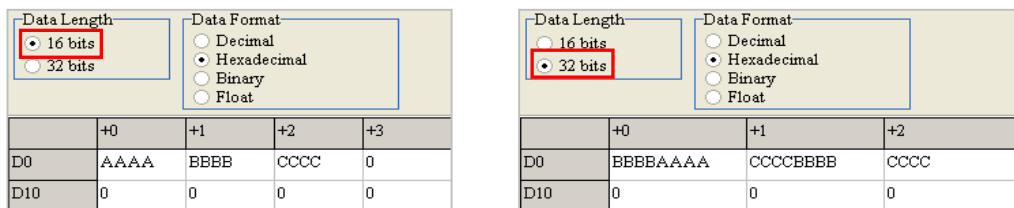


The users can select a data format in the **Data Format** section.

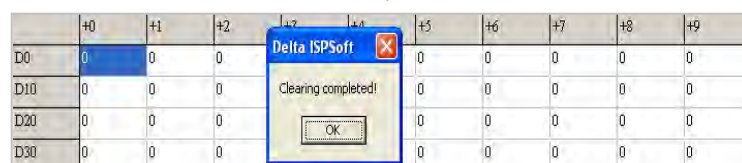
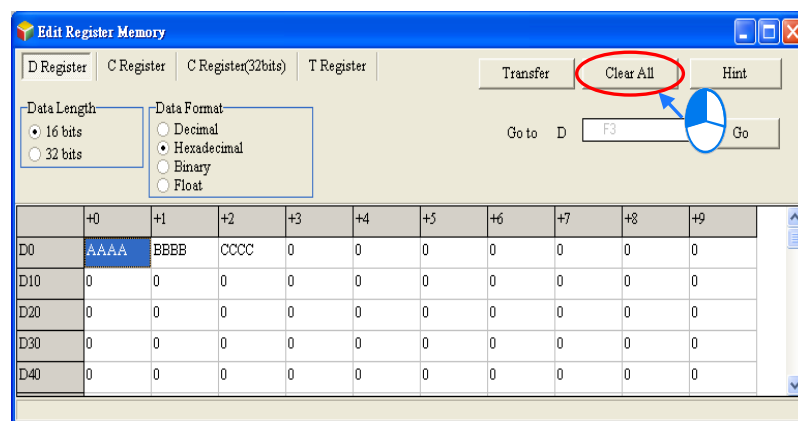


The users can select the **16 bits** option button or the **32 bits** option button in the **Data Length** section. If the **32 bits** option button is selected, the value in the cell for a device includes the value in the next device.

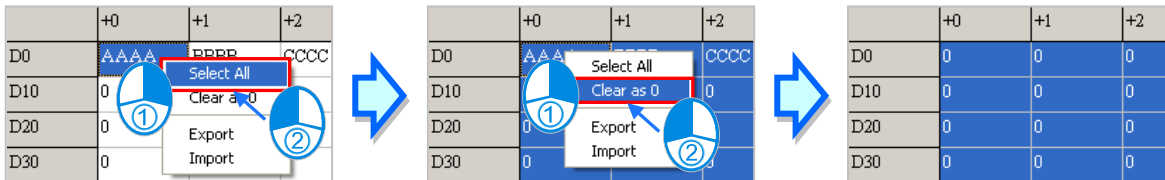
Take the figures below for example. The hexadecimal value in D0 is AAAA, the hexadecimal value in D1 is BBBB, and the hexadecimal value in D2 is CCCC. If the **32 bits** option button in the **Data Length** section is selected, the value in the cell for D0 will include the value in D1, the value in the 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.



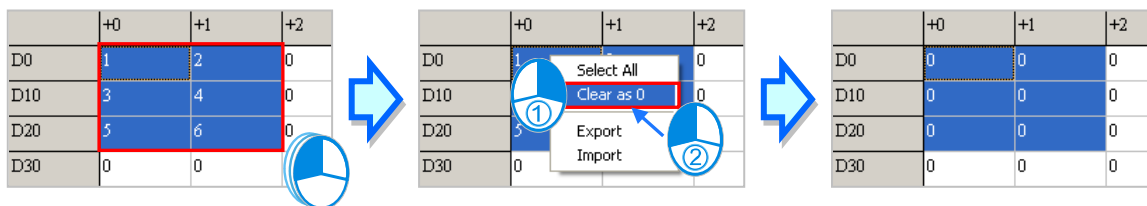
If the users click **Clear All**, the values on the present page will be cleared to 0, and the values on the other pages will remain unchanged.



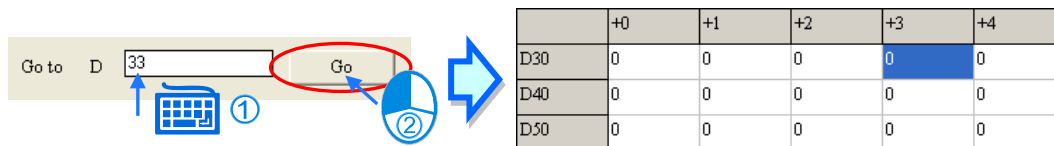
Besides, the users can clear the values on the present page by right-clicking the present page, clicking **Select All** on the context menu, right-clicking the present page again, clicking **Clear as 0** on the context menu.



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 type the device number in the **Go to** field, click **Go**. The blue cursor will move to the number you have typed.

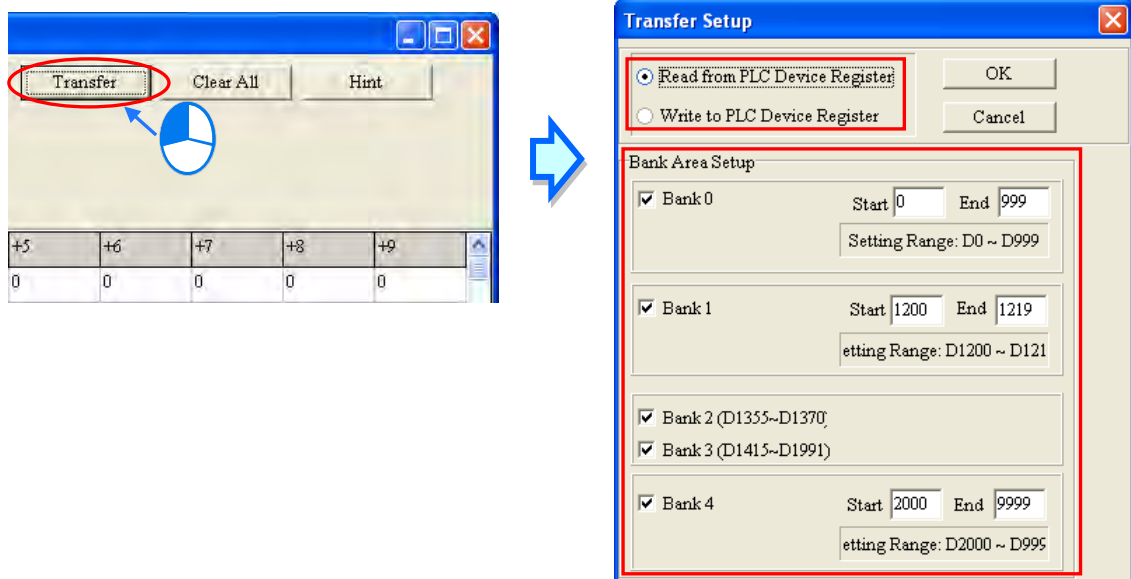


# 17

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.

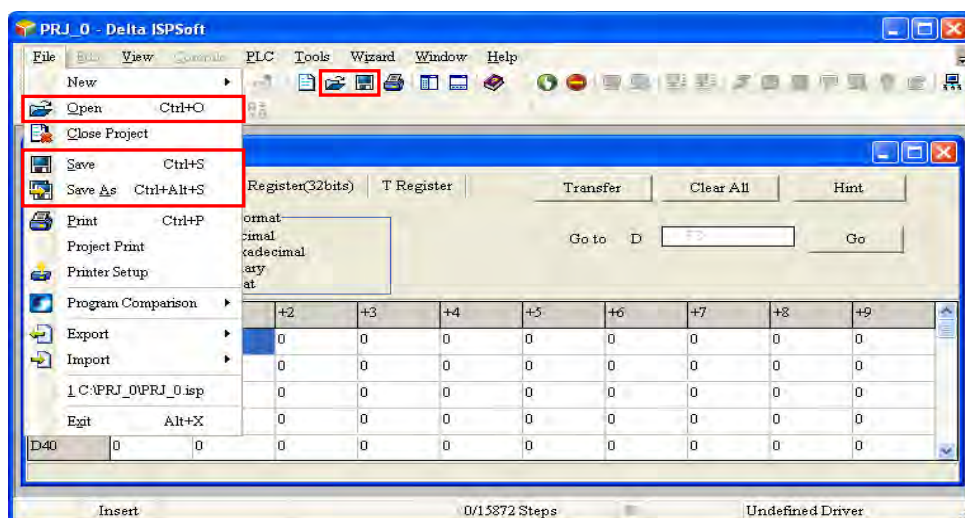
The users can select the **Read from PLC Device Register** option button or the **Write to PLC Device Register** option button. Besides, the users can select device ranges. The values in the devices which are not in the device ranges will remain unchanged after **OK** is clicked. After the setting is 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.



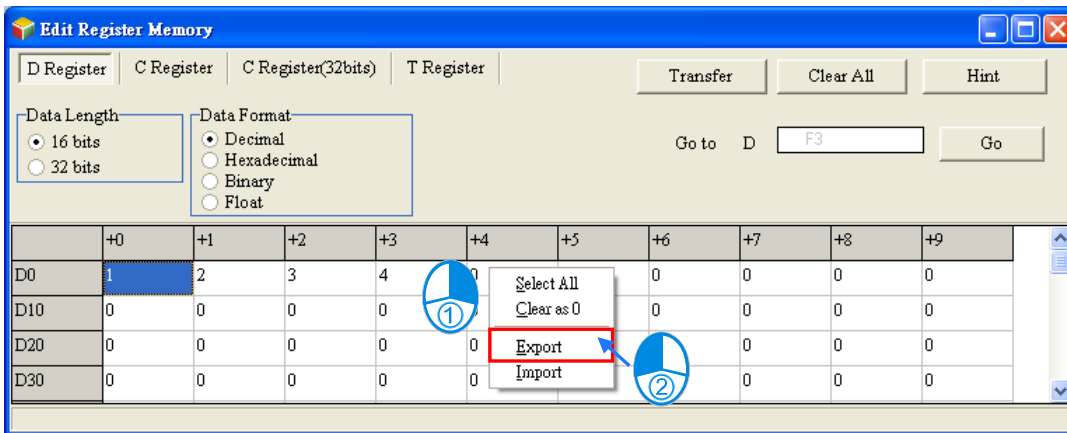


**! Before the values are written into the registers, users have to make sure that the operation does not affect the system, or cause damage to the system and staff.**

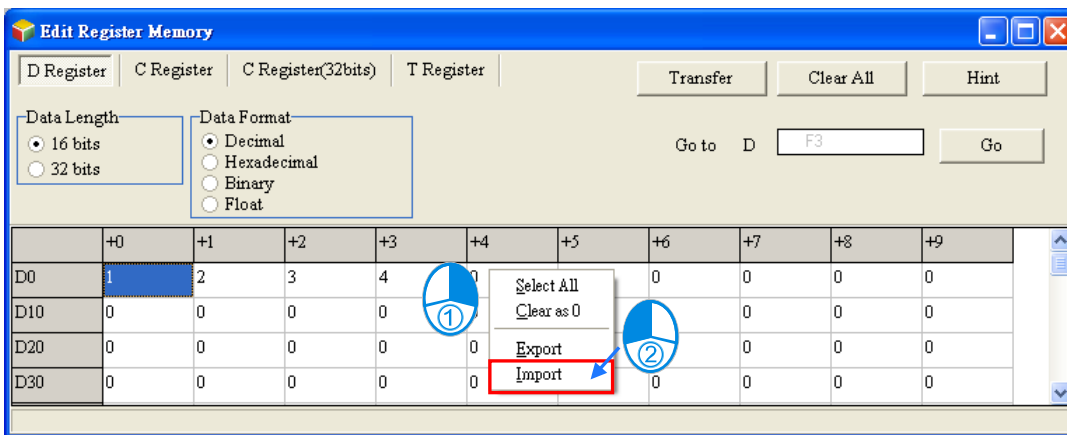
The values in the **Edit Register Memory** window can be saved with the project. If the present window is the **Edit Register Memory** window, the values in the **Edit Register Memory** window will be saved as a .dvl file whose primary filename is the project name in the folder in which the project (\*.isp) 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.



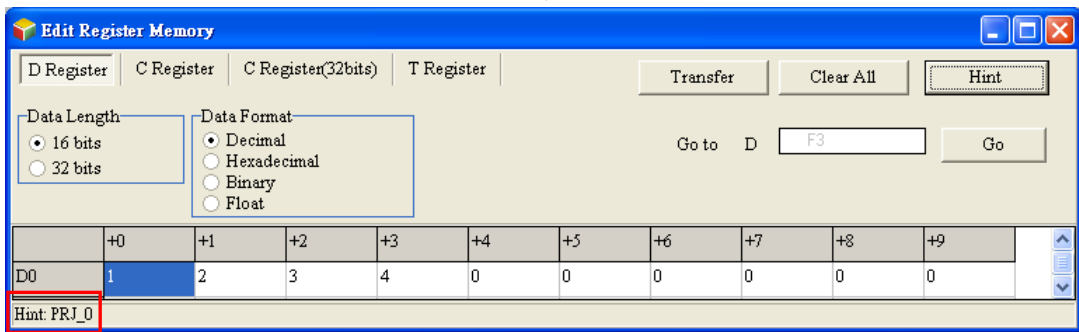
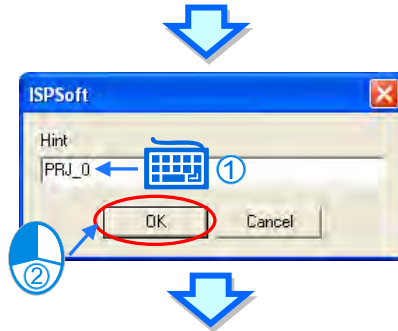
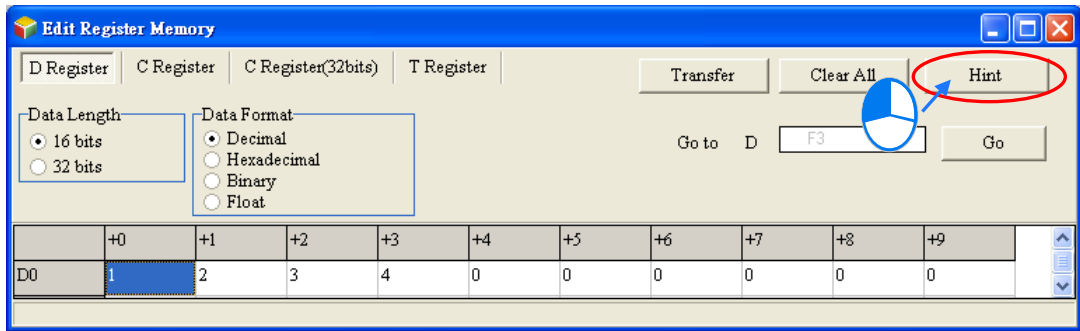
The users can export the values on the present page as a CSV file, and edit the CSV file through Microsoft Excel. If the users want to export the values on the present page, they can right-click the present page, select the range you'd like to export and then click **Export** on the context menu. Select 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.




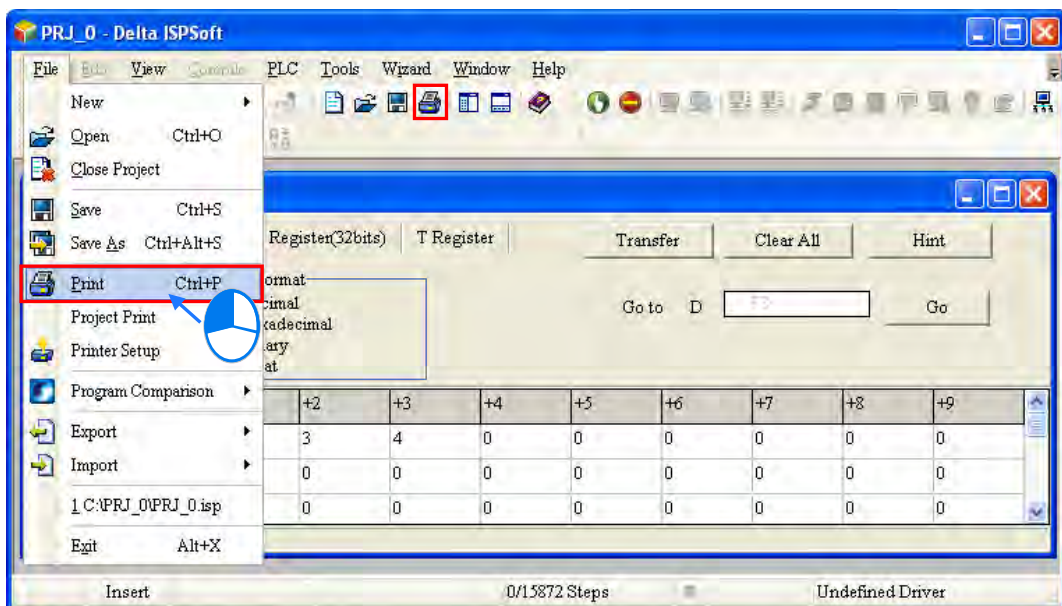
If the users want to import a CSV file into a page, they can right-click the page, click **Import** on the context menu, select the CSV file in the **Open** window, and click **Open**. The file imported must be matched with the device type clicked.



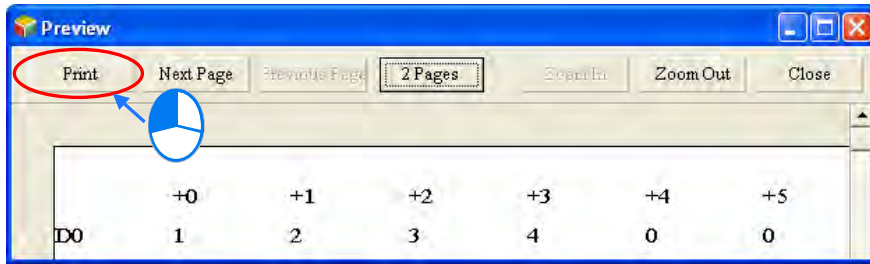
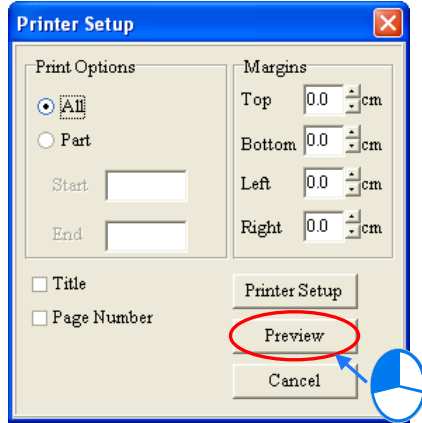
After the users click **Hint** in the **Edit Register Memory** window, the **ISPSOft** window will appear. After a hint is typed in the **ISPSOft** window, the hint will appear at the bottom of the **Edit Register Memory** window. The hint is saved with the values in the **Edit Register Memory** window. 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.

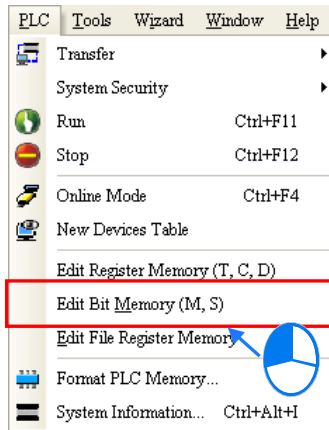


Select the **All** option button or the **Part** option button in the **Print Options** section, set margins, and select items which will be added. After the users click **Printer Setup**, they can select a printer, and set a print format. After the users click **Preview**, they can preview the document which will be printed. After the setting is complete, the users can click **Print** in the **Preview** window.



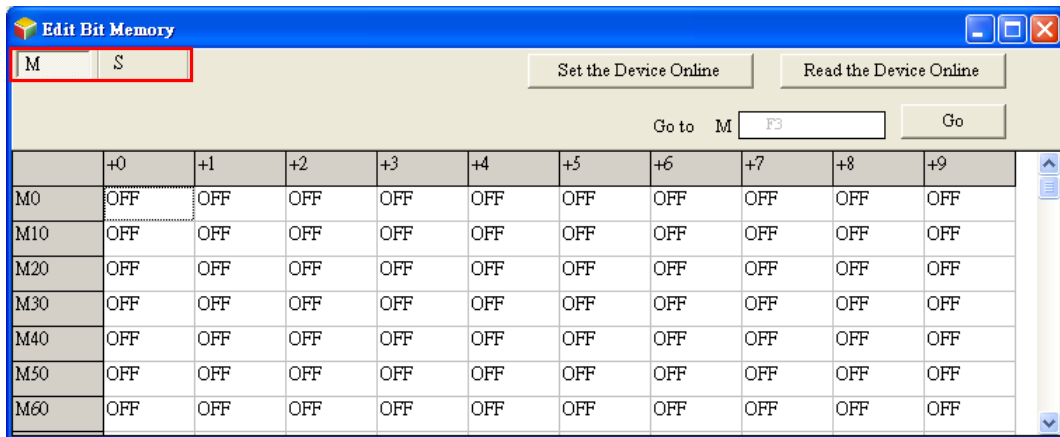
### 17.2.5 Edit Bit Memory

Users can edit the states of several M/S devices in a batch. The new states of these M/S devices can be saved and downloaded. After users create a project, they can click **Edit Bit Memory (M, S)** on the **PLC** menu to open the **Edit Bit Memory** window.

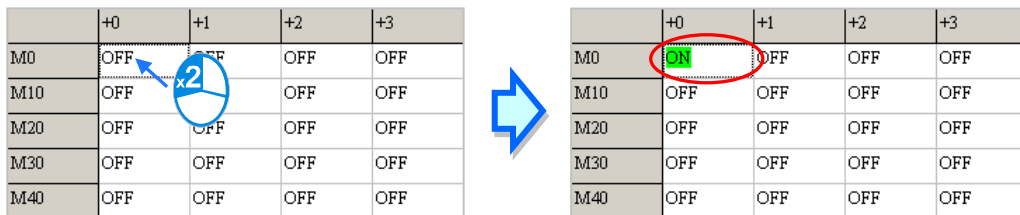


The states in the tables in the **Edit Bit Memory** window are not the states uploaded from the PLC. They are the states saved last time. If the **Edit Bit Memory** window in the project is opened for the first time, the default states in the tables are OFF.

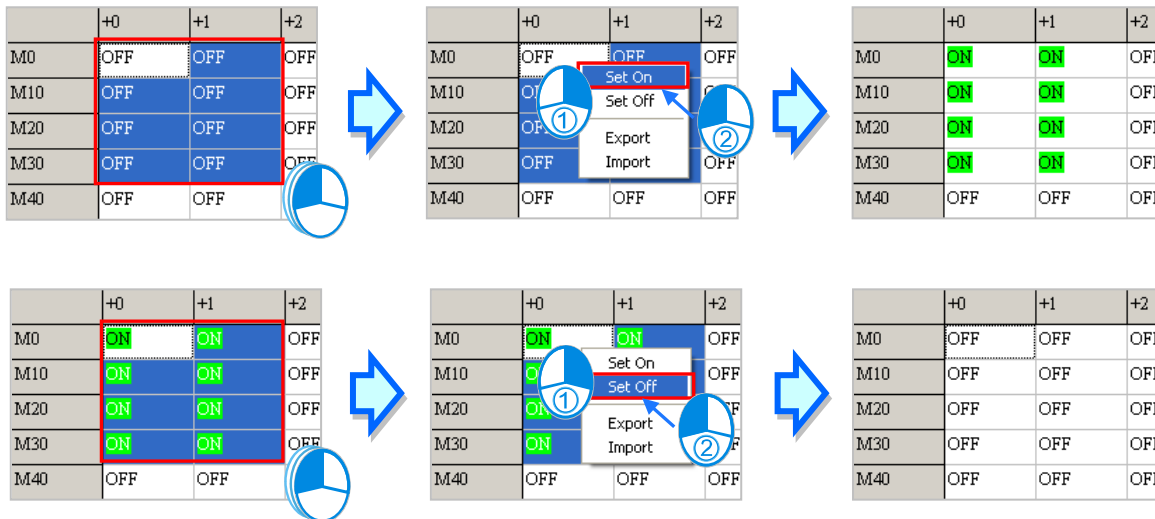
The tabs at the top of the **Edit Bit Memory** window indicate device types.



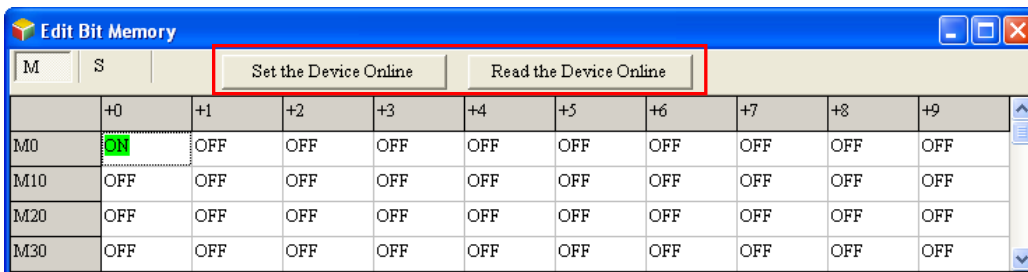
If the users want to change the state in the cell for a device, they can double-click the cell.



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.



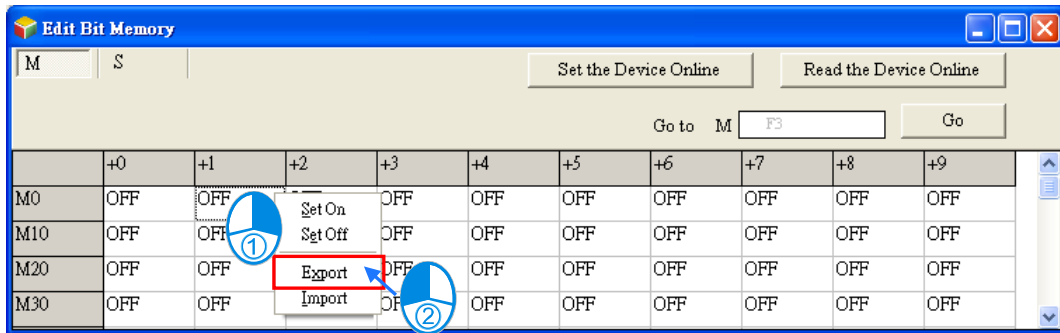
If the users click **Set the Device Online**, the states in the **Edit Bit Memory** window will be written into the PLC. If the users click **Read the Device Online**, the states of the M devices, and the states of the S devices will be read from the PLC. Before **Set the Device Online/Read the Device 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.



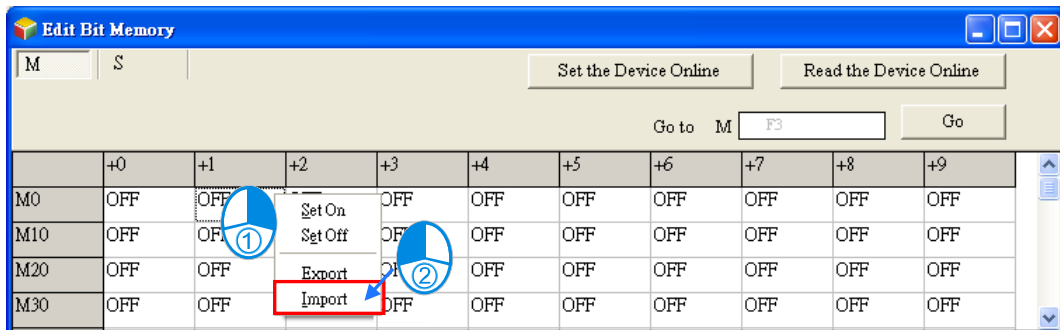
17

**⚠ 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, select the range you'd like to export and then 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.

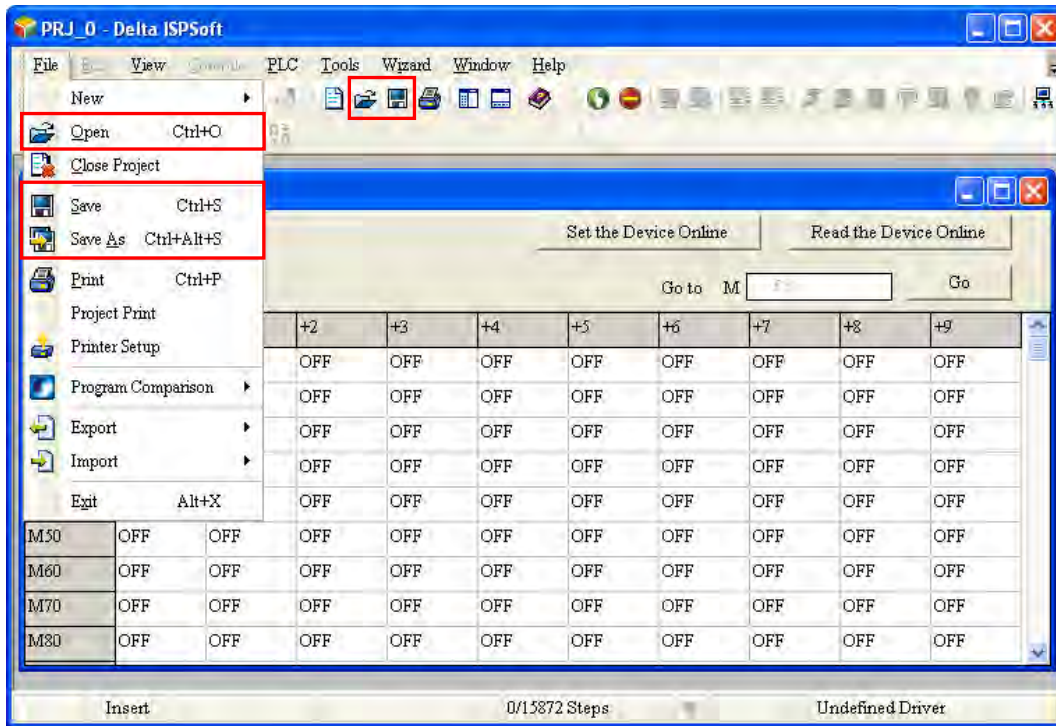



If the users want to import a CSV file into a page, they can right-click the page, click **Import** on the context menu, select the CSV file in the **Open** window, and click **Open**. The file imported must be matched with the device type clicked.

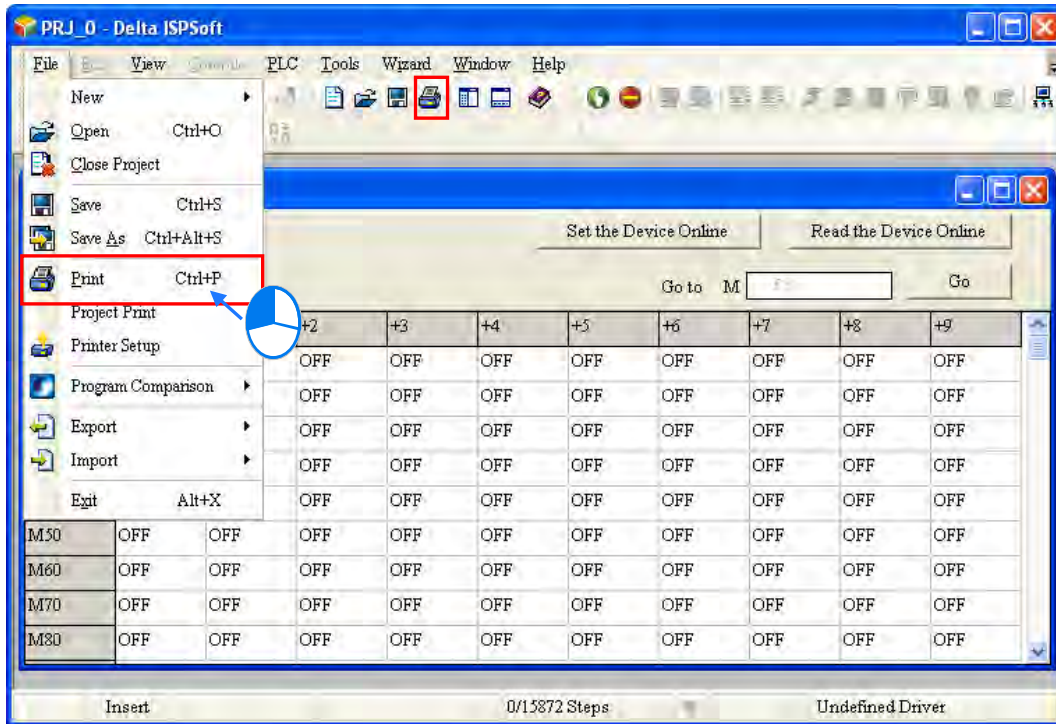


The states in the **Edit Bit Memory** window can be saved with the project. If the present window is the **Edit Bit Memory** window, the states in the **Edit Bit Memory** window will be saved as a dvb file whose primary filename is the project name in the folder in which the project (\*.isp) is saved 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.



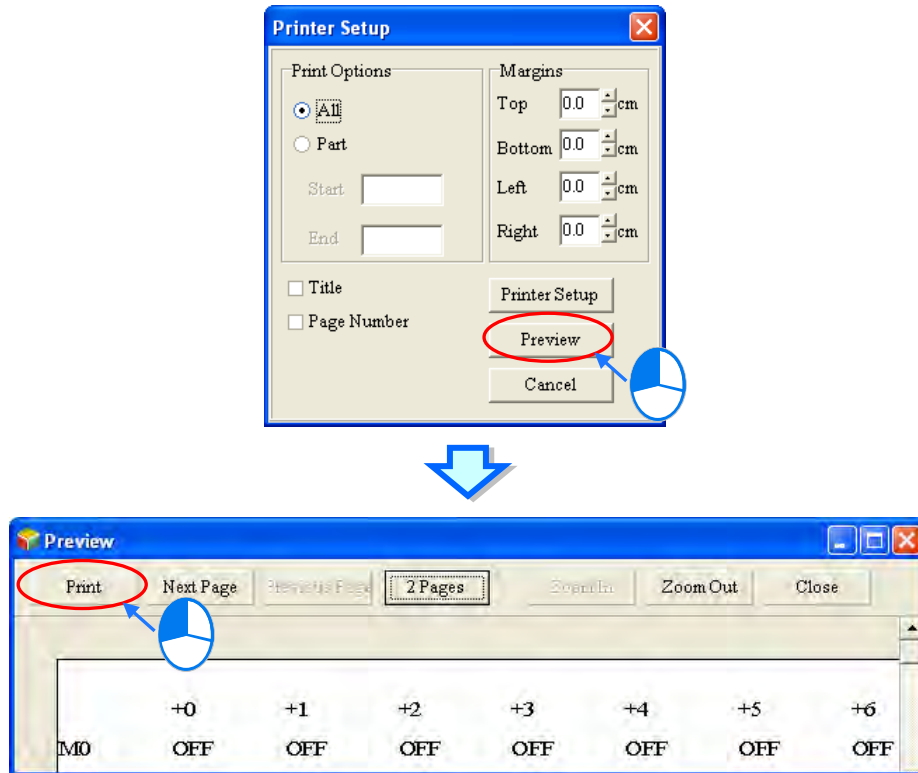
If the present window is the **Edit Bit Memory** window, the users can print the present page in the **Edit Bit Memory** window by clicking **Print** on the **File** menu or  on the toolbar.



17



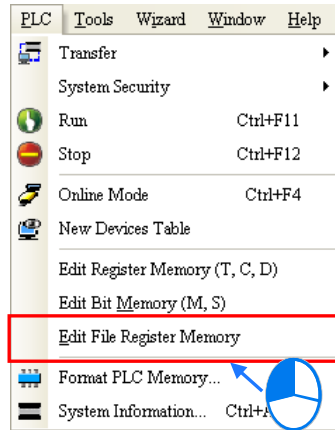
Select the **All** option button or the **Part** option button in the **Print Options** section, set margins, and select items which will be added. After the users click **Printer Setup**, they can select a printer, and set a print format. After the users click **Preview**, they can preview the document which will be printed. After the setting is complete, the users can click **Print** in the **Preview** window.



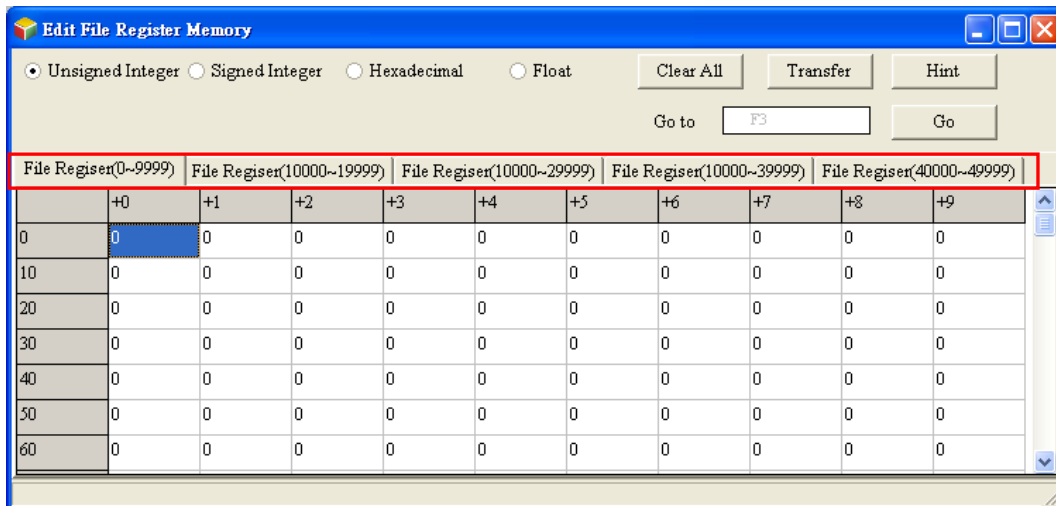
### 17.2.6 Edit DVP Series File Register

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.

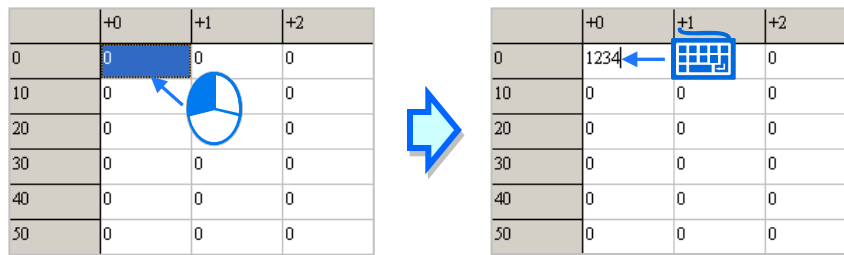


The values in the table in the **Edit File Register Memory** window are not the values uploaded from the PLC. They are the values saved last time. If the **Edit File Register Memory** window in the project is opened for the first time, the default values in the table are 0. Users can select different tap for different register ranges.

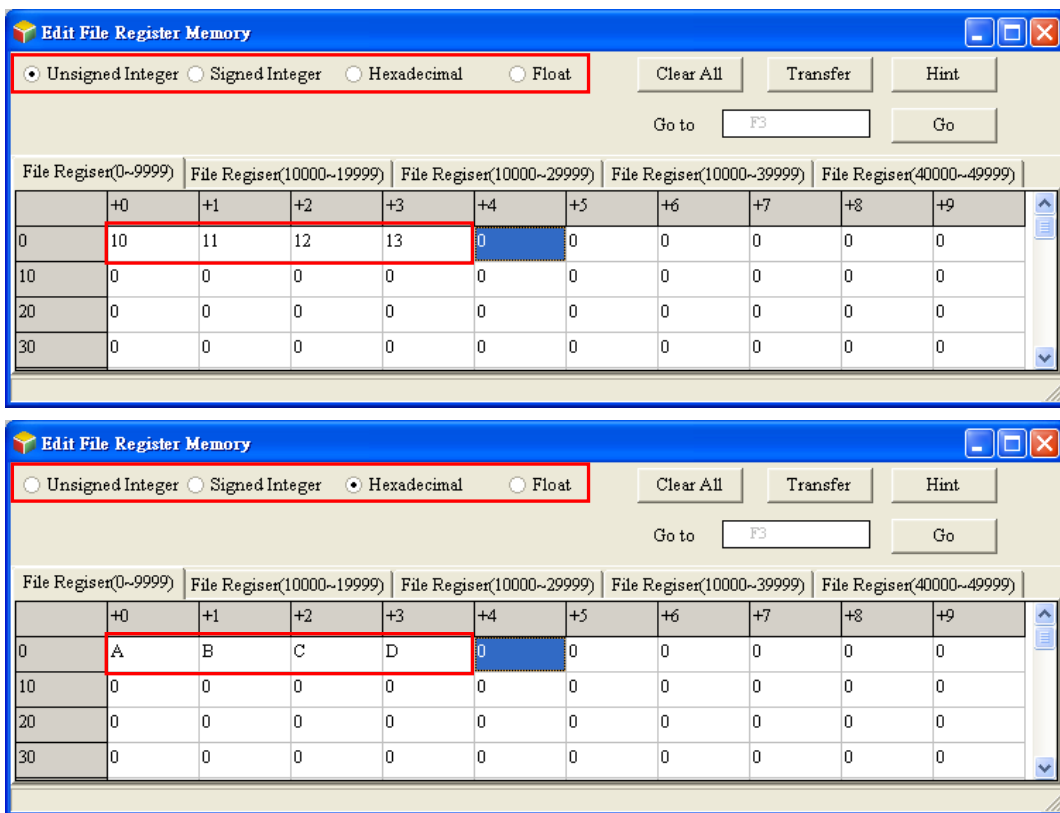


17

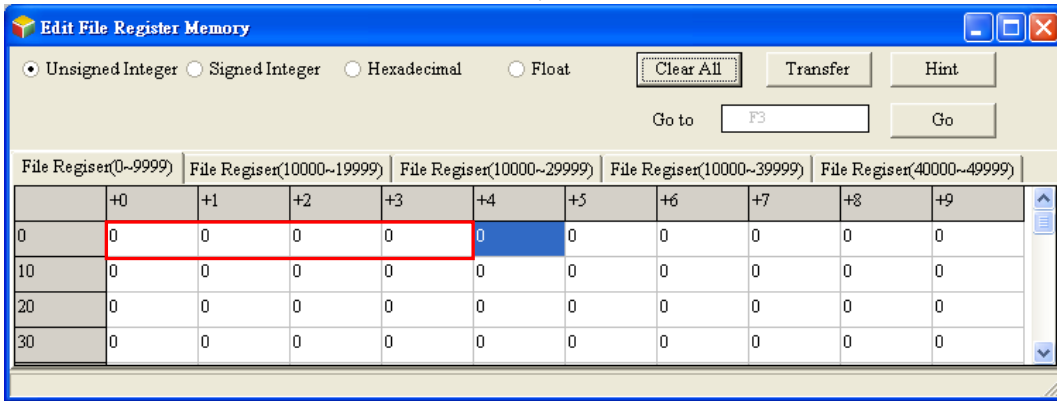
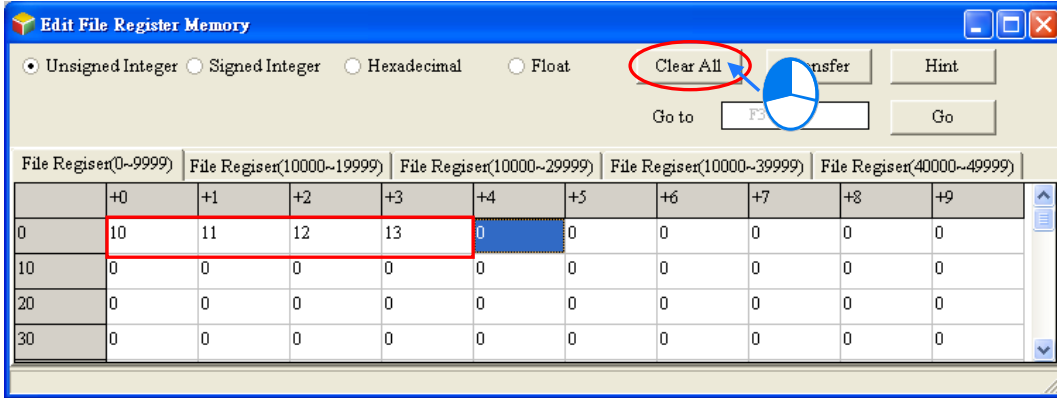
If the users want to edit the value in the cell for a file register, they can click the cell for the file register, and type a value.



The users can select a data format at the top of the **Edit File Register Memory** window.

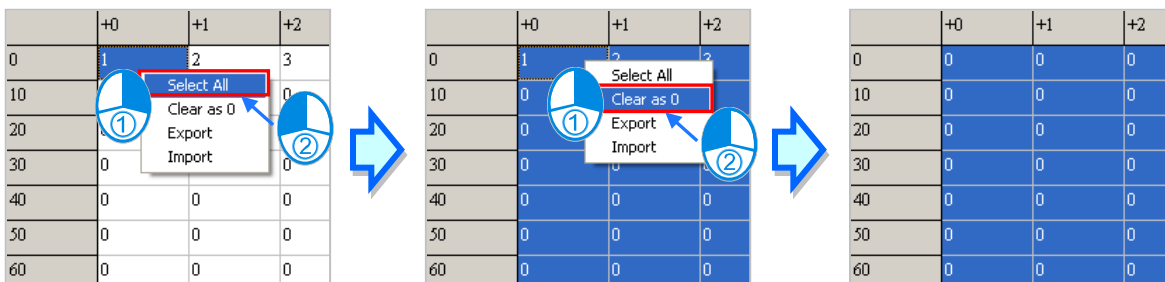


If the users click **Clear All**, the values in the **Edit File Register Memory** window will be cleared to 0.



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.

17

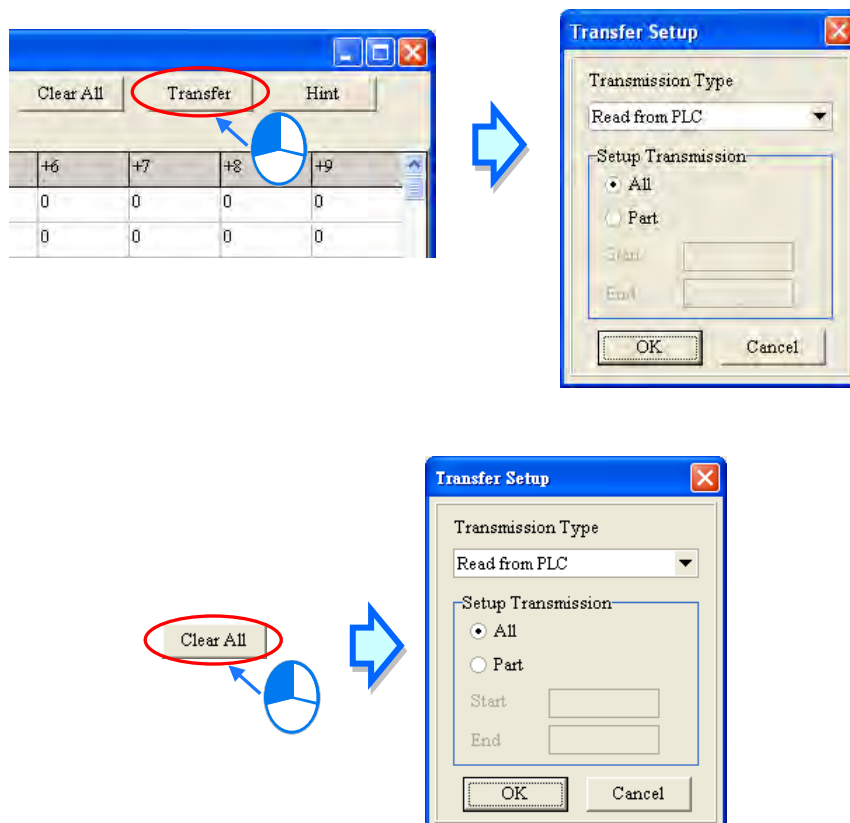


If the users want to clear the values in the cells for a range of file registers on the present page, they can drag across the cells, right-click the present page, and click **Clear as 0** on the context menu.



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.

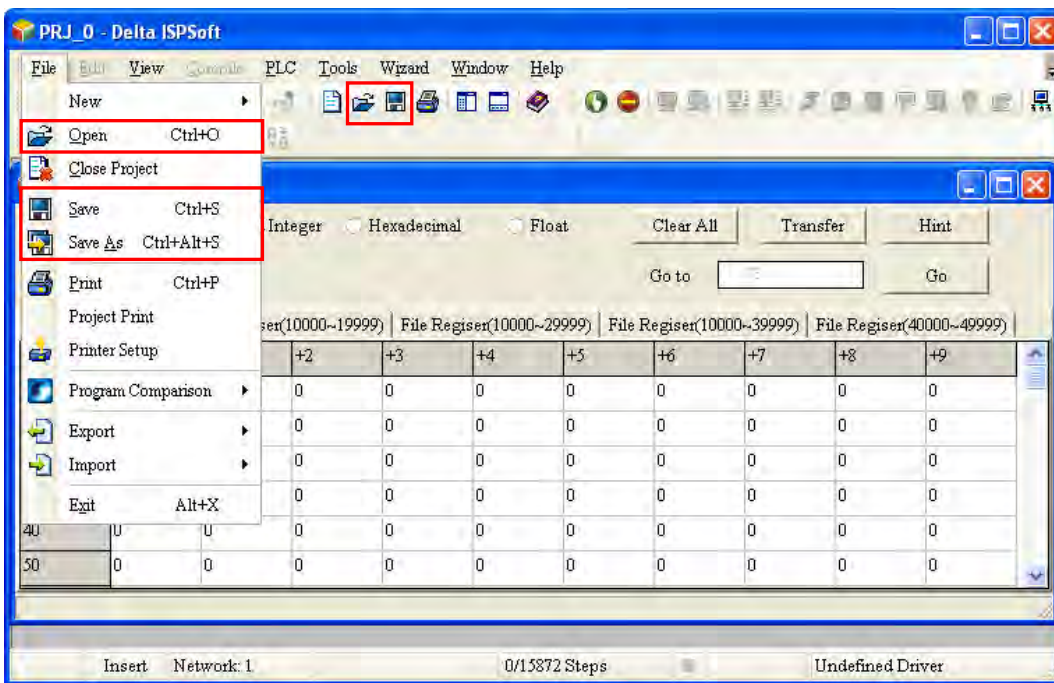
Select **Read form PLC** or **Write to PLC** in the **Transmission Type** drop-down list box, and then select the **All** option button or the **Part** option button. After the setting is complete, the users can click **OK**.



**⚠ 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.**

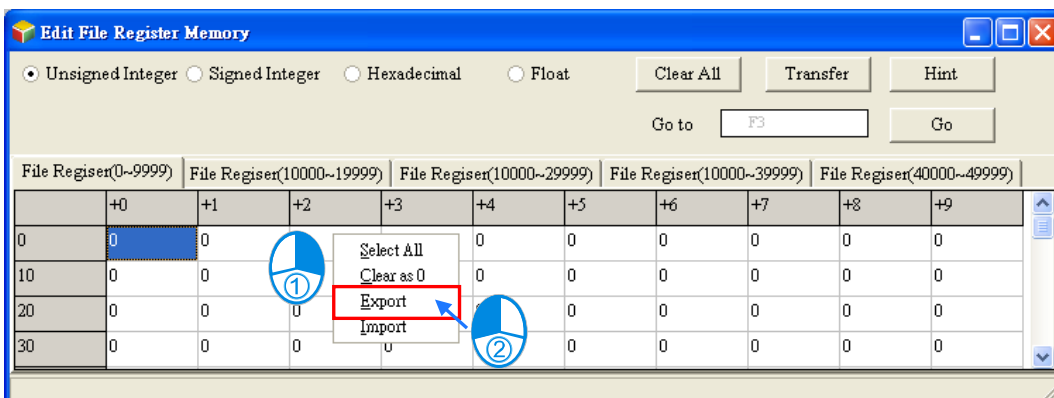
The values in the **Edit File Register Memory** window can be saved with the project. If the present window is the **Edit File Register Memory** window, the values in the **Edit Register Memory** window will be saved as a wft file whose primary filename is the project name in the folder in which the 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.

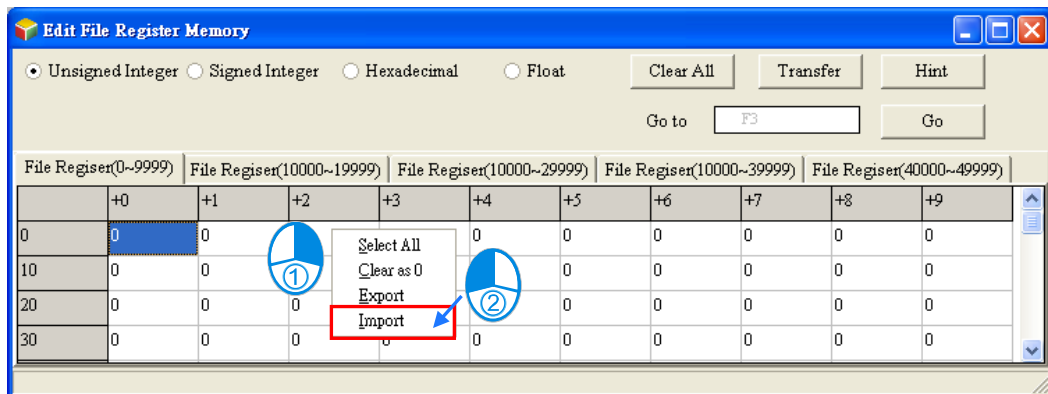


17

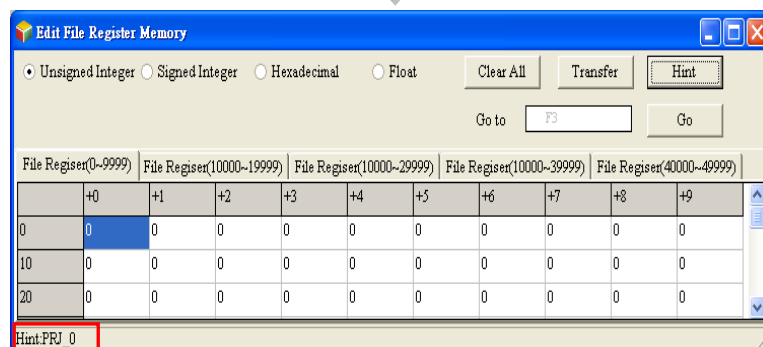
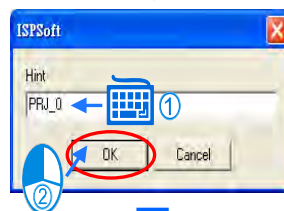
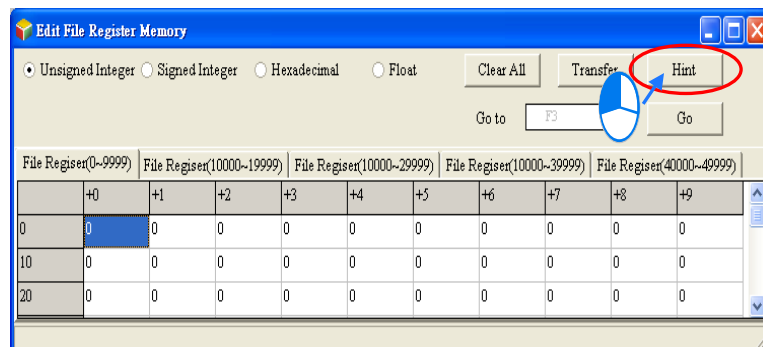
The users can export the values on the present page as a CSV file, and edit the CSV file through Microsoft Excel. If the users want to export the values on the present page, they can right-click the present page, click **Export** on the context menu, select a folder in the **Save in** drop-down list box in the **Save As** window, type a filename in the **File name** box, and click **Save**.




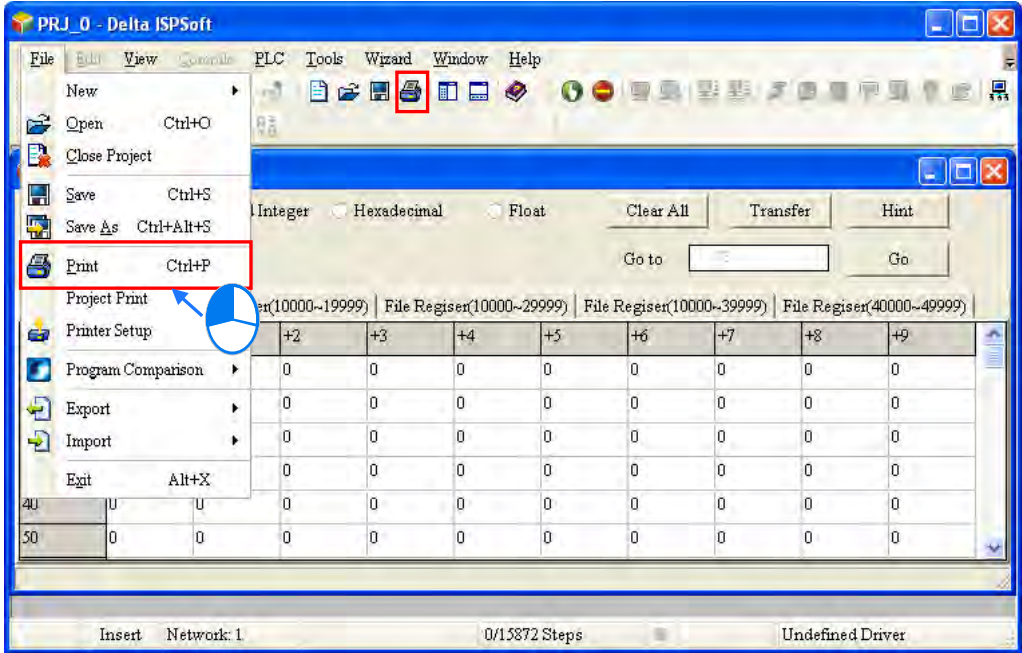
If the users want to import a CSV file into the page, they can right-click the page, click **Import** on the context menu, select the CSV file in the **Open** window, and click **Open**.



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.

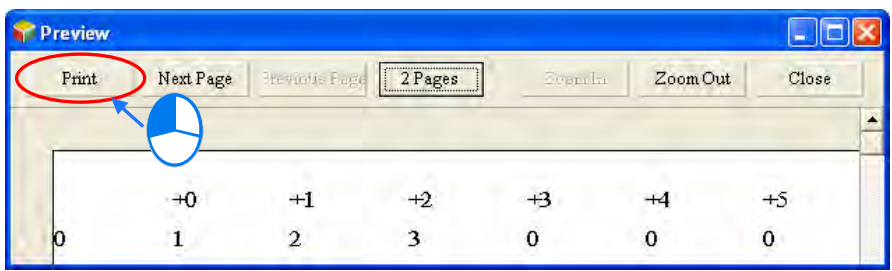
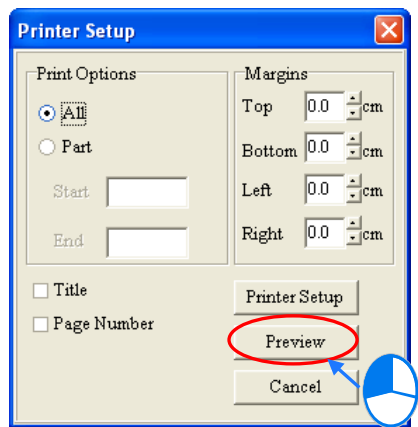


If the present window is the **Edit File Register Memory** window, the users can print the present page in the **Edit File Register Memory** window by clicking **Print** on the **File** menu or  on the toolbar.



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.

17

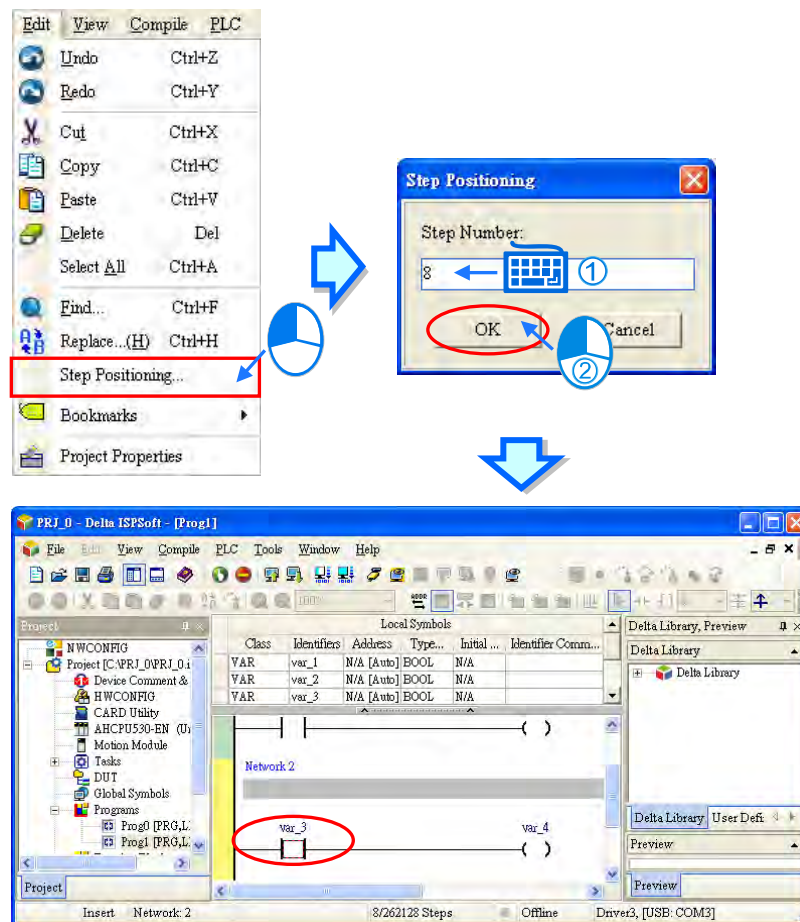




## 17.3 Step Positioning

### 17.3.1 Using Step Positioning

When an error occurred during operation, users can use special registers (e.g. SR1, SR2, the 32 bit registers for storing wrong addresses in the AH series) to know which step went wrong and then go to the specific program to check and fix it. Go to Edit > Step Positioning to open the Step Positioning window. Type the step number you'd like to inquire and press OK. The system will direct you to the corresponding POU and the position of the step in the program, as the image shown below. Be sure to program the project before using the step positioning function.



## 17.4 Program Comparison

### 17.4.1 Introduction to Program Comparison

ISPSOft provides a program comparison tool for users to compare 2 ISPSOft projects, or to compare an ISPSOft project and the PLC internal programs. Here are the comparison rules.

- Comparison items include product series, models, POU's, global symbols, local symbols and programs.
- For a POU comparison, the name of a POU will be checked first. Different POU name will be seen as different, even for those with the same contents.
- For a variable symbol comparison, the order of a variable symbol will be checked first. Different variable symbol order will be seen as different, even for those with the same contents. As the image shown below, the var\_a and var\_b from the left and right are with different order; thus they will be seen as different.

Class	Identifiers	Address	Type...	Initial Value
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A

Class	Identifiers	Address	Type...	Initial Value
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A

- The comparison of variable symbols will include the step skipping in the program. As the image shown below the var\_c only appears on the left and that does not exist on the right.

Class	Identifiers	Address	Type...	Initial Value
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A
VAR	var_e	N/A [Auto]	BOOL	N/A

Class	Identifiers	Address	Type...	Initial Value
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A
VAR	var_e	N/A [Auto]	BOOL	N/A

17

- The comparison of POU's in the languages of LD, FBD, IL, ST will include the step skipping in the program. As the image shown below, the second line on the left does not exist on the right.

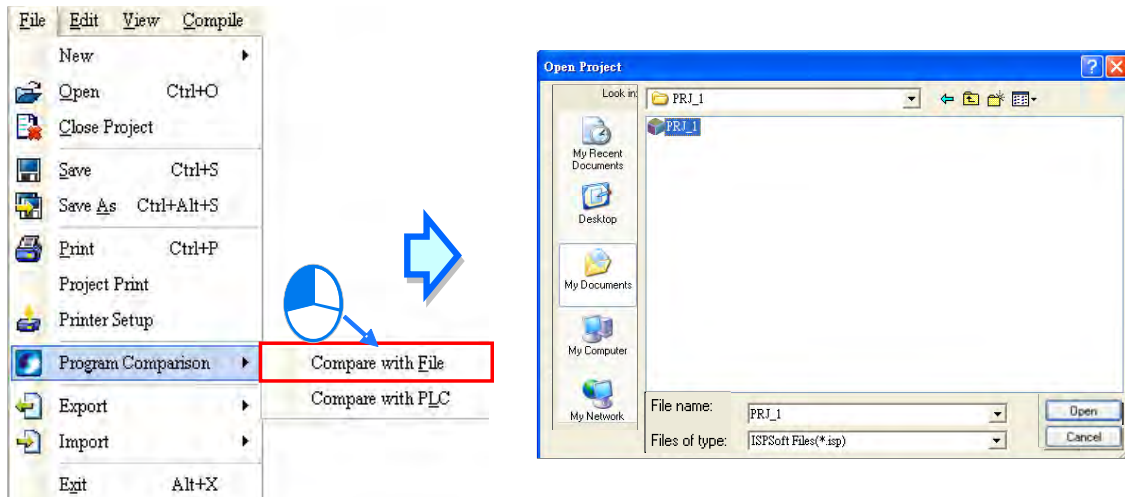
0001	ld var1	0001	ld var1
0002	and var3	0002	out var2
0003	out var2	0003	ld m0
0004	ld m0	0004	out var3
0005	out var3		

- Once there is a difference in the SFC, the comparison of POU's in the language of SFC will be seen as different. When the Action and Transition are different in the SFC, the system will identify which sections or lines of action/transition are different.

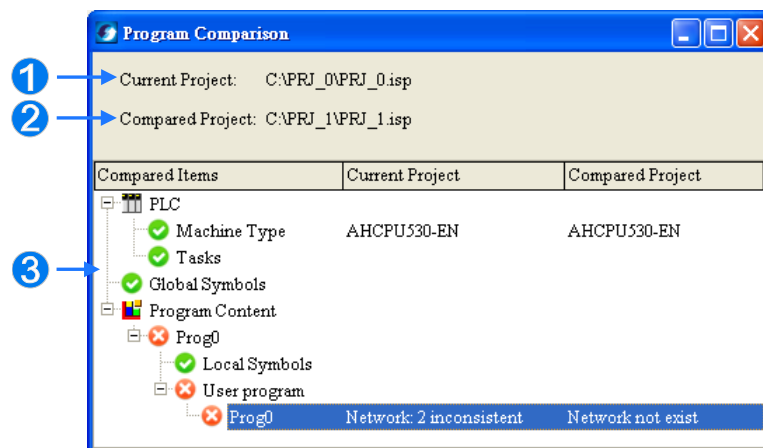
- Projects should be saved before running the program comparison tool.
- Be sure to select the program items while downloading the CPU module; so that the programs in the CPU module can be compared.

## 17.4.2 Compare with File

It is used to compare different projects in the computer. Select File > Program Comparison > Compare with File, and then the Open Project window will appear. Select the project you'd like to compare with and click Open.

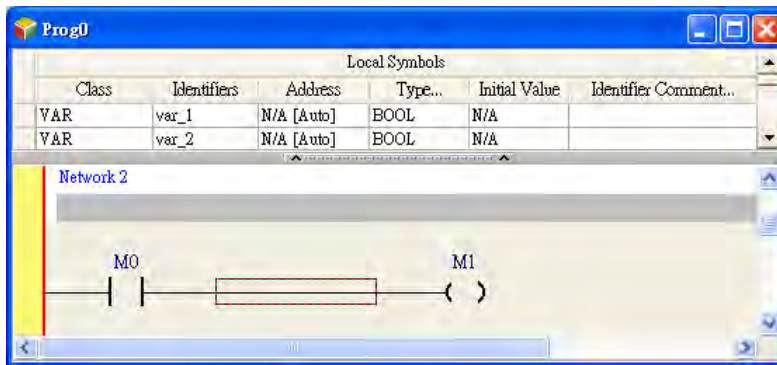
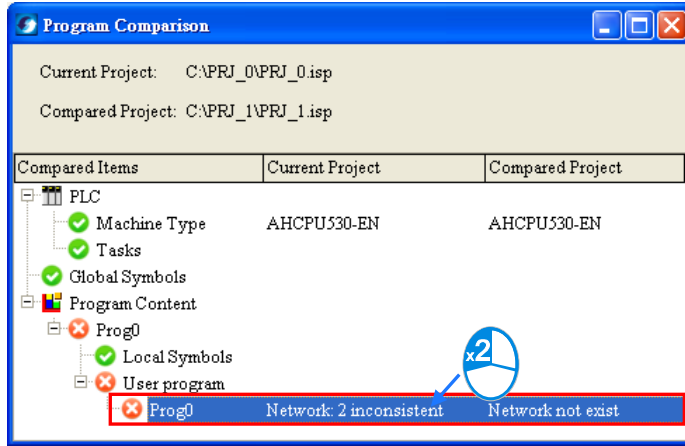


After that you will see a program comparison table, listing the sameness and the differences.



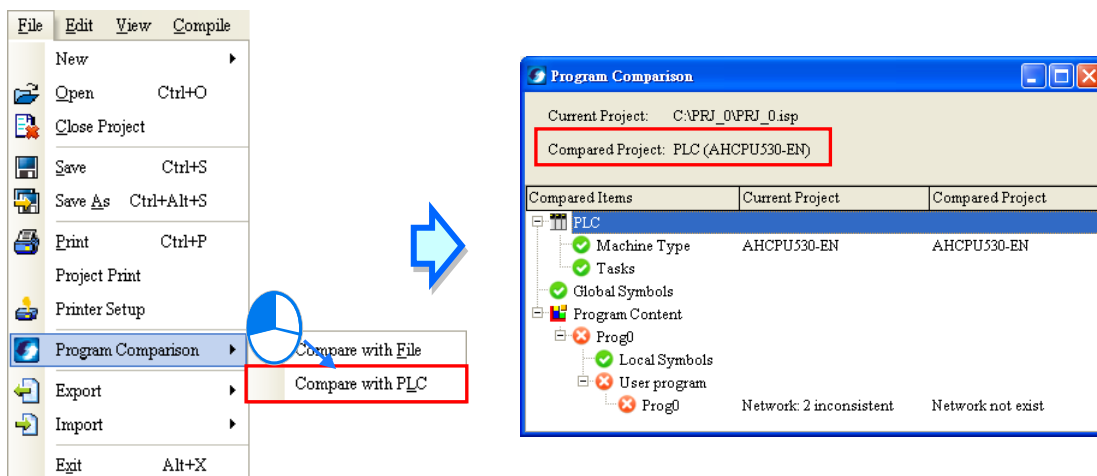
- 1** The location of the current project
- 2** The location of the compared project
- 3** The comparison results, including models, global symbols and program contents.

Double click the items with different comparison results; the system will open the contents for users to review. As the image shown below, click Prog0, the item with a different comparison result, to show its contents.



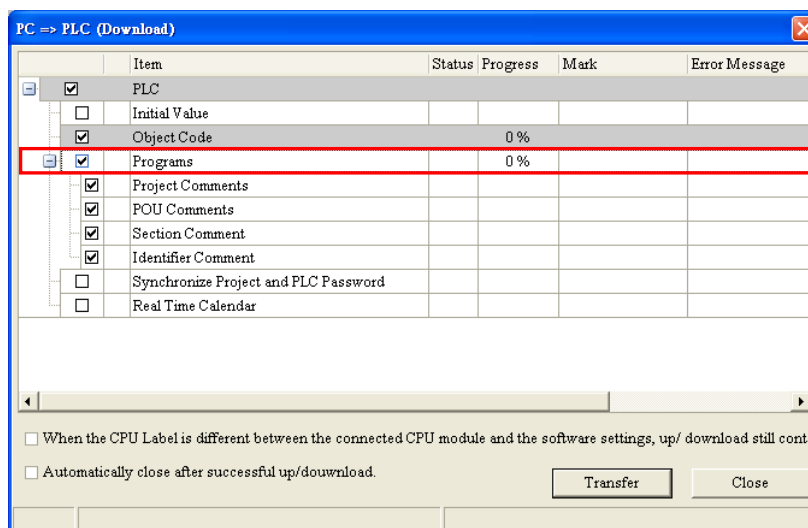
### 17.4.3 Compare with PLC

It is used to compare the CPU module contents and the projects in the computer. Make sure the connection between ISPSOft and the CPU module has been established. Select File > Program Comparison > Compare with PC=LC, and then the Program Comparison window will appear. Open the Program Comparison window and view the compared items (see below). PLC type used in connection with the compared project is also shown.



Select the items from the PC => PLC (Download) you'd like to compare with the CPU module. If not selecting any item to compare, an error message will appear and no comparison can be made.

Refer to section 17.1.3 for more information on Downloading.



17

**MEMO**

17

# Chapter 18 Testing and Debugging Tools





## Table of Contents

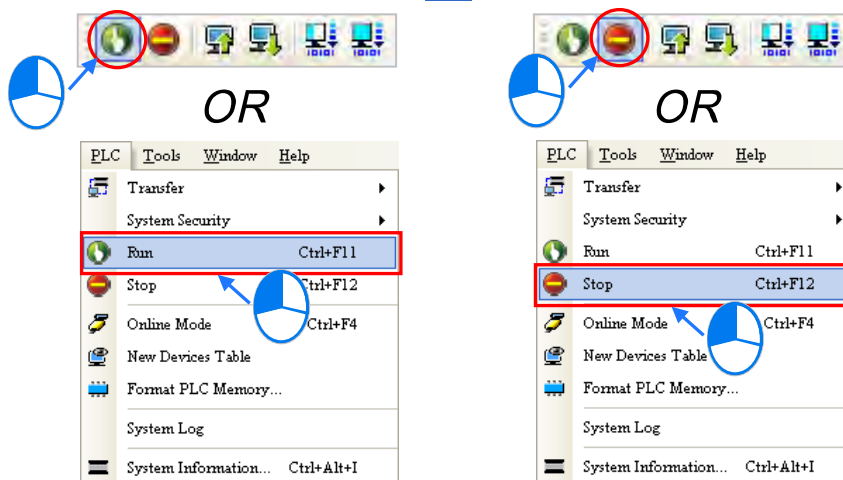
<b>18.1</b>	<b>Online Monitoring Function .....</b>	<b>18-2</b>
18.1.1	Changing the Operating Status of a PLC .....	18-2
18.1.2	Online Monitoring Function and Environment.....	18-2
18.1.3	Changing the States of the X Devices in the Online Mode.....	18-7
18.1.4	Monitoring a Program Online.....	18-8
18.1.5	Device Monitoring Table .....	18-15
18.1.6	Online Editing Function and Online Update Function .....	18-21
<b>18.2</b>	<b>Debugging Mode for DVP Series .....</b>	<b>18-24</b>
18.2.1	Enabling the Debugging Mode for DVP Series .....	18-24
18.2.2	Adding and Clearing Breakpoints .....	18-25
18.2.3	Execution of the Program in the Debugging Mode .....	18-25
<b>18.3</b>	<b>Debugging Mode for AH/AS Series .....</b>	<b>18-27</b>
18.3.1	Enter Debugging Mode .....	18-27
18.3.2	Adding and Clearing Breakpoints .....	18-27
18.3.3	Continuous Execution.....	18-29
18.3.4	Single-Step Execution .....	18-29
<b>18.4</b>	<b>Checking the Status of a PLC.....</b>	<b>18-31</b>
18.4.1	System Information .....	18-31
18.4.2	PLC System Log .....	18-33
<b>18.5</b>	<b>3D Chart .....</b>	<b>18-40</b>
18.5.1	Features.....	18-40
18.5.2	Creating a 3D Chart .....	18-40
18.5.3	Creating a Curve .....	18-41
18.5.4	Display a 3D Chart.....	18-43

## 18.1 Online Monitoring Function

### 18.1.1 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.


### 18.1.2 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.

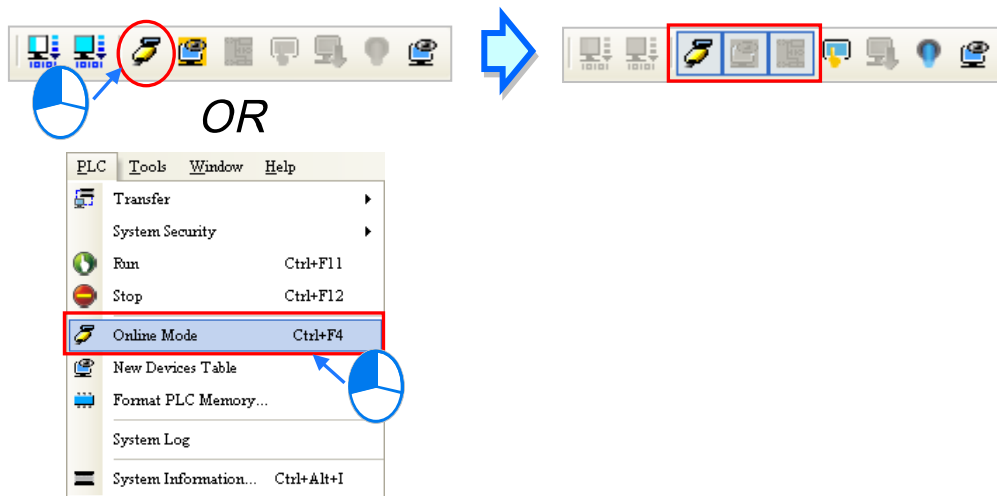
Monitoring mode	Description
 Device monitoring mode	Users can monitor the statuses of the devices in the PLC through the monitoring table. In this mode, ISPSOft only needs to update the statuses of the devices. The present program in ISPSOft does not need to be the same as the program in the PLC.







 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.

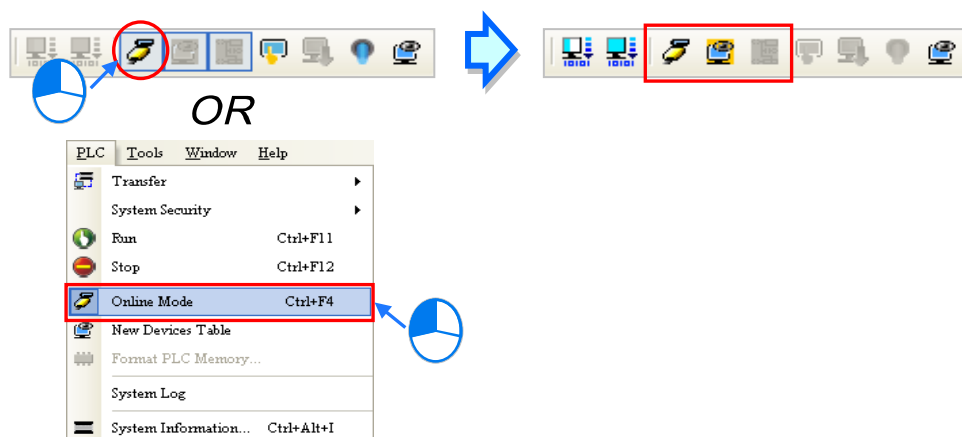
After users click **Online Mode** on the **PLC** menu, or  on the toolbar, the online monitoring function will be enabled. The system will also enable the device monitoring mode and the program monitoring mode.



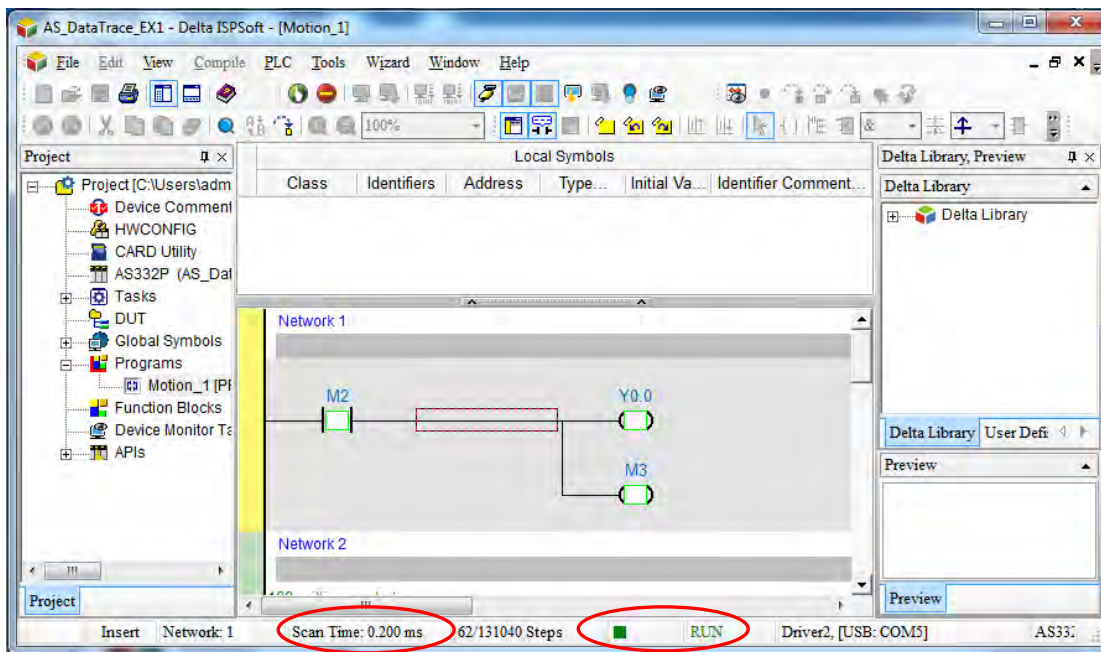
If the users only want to monitor the statuses of the devices in the PLC, they can click  on the toolbar. When  on the toolbar is clicked,  on the toolbar is also clicked. If the users want to monitor the program, they can click  on the toolbar.



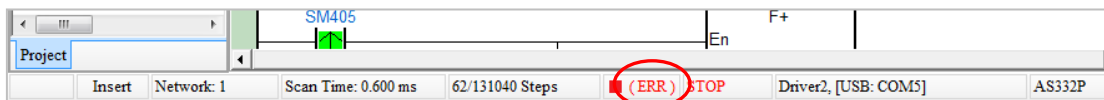
If the users want to disable the online monitoring function, they can click **Online Mode** on the **PLC** menu, or  on the toolbar.



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 green light blinks.

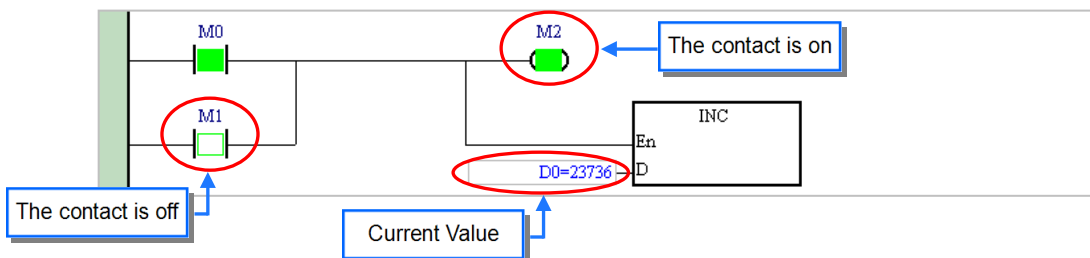


If an error occurs in the PLC, users may not be able to operate the PLC, e.g. the PLC can not run. Please refer to the manual for the model used for more information about troubleshooting.

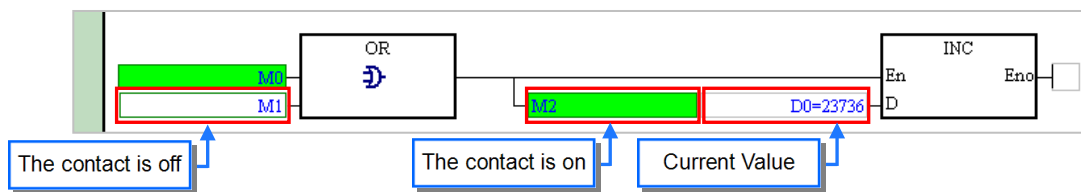


The present statuses of the devices will be displayed in the program editing window after the program monitoring function is enabled. Users can debug and test the program in the program monitoring mode. The way in which the statuses of the devices in a program are displayed depends on the programming language used to create the program.

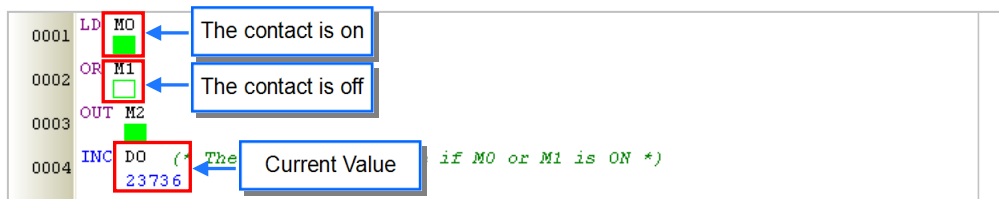
A ladder diagram in the program monitoring mode is shown below.



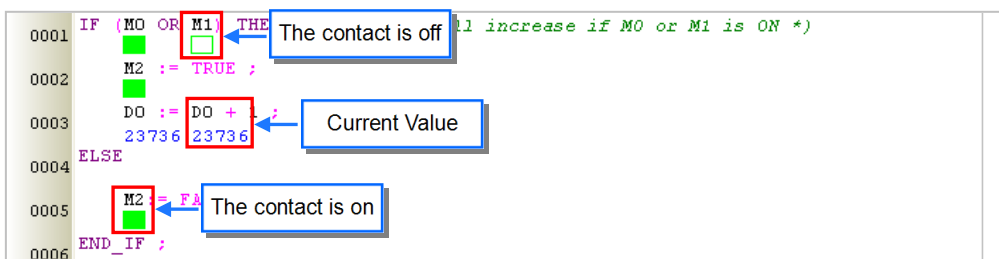
A function block diagram in the program monitoring mode is shown below.



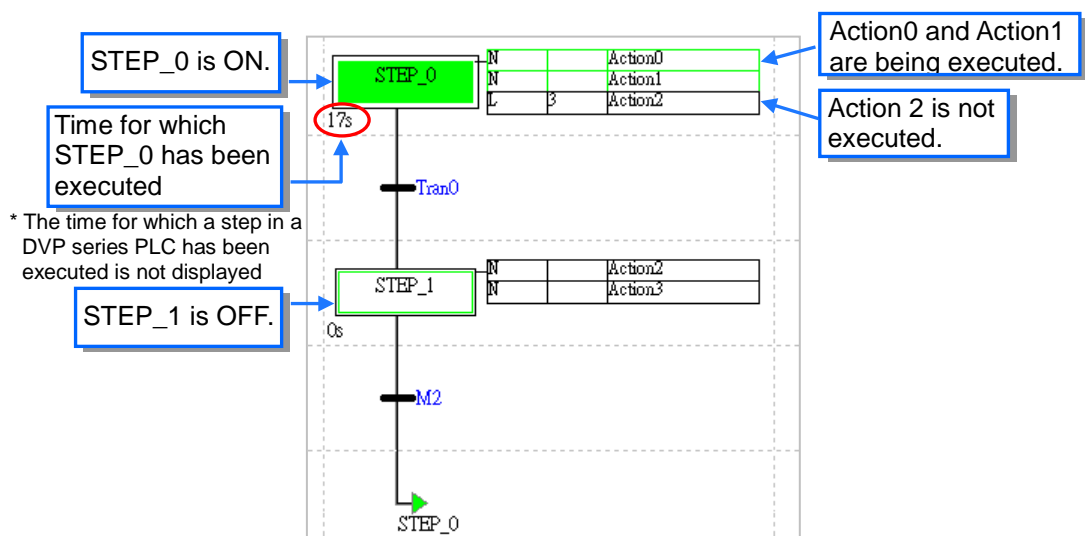
An instruction list in the program monitoring mode is shown below.



A structured text in the program monitoring mode is shown below.

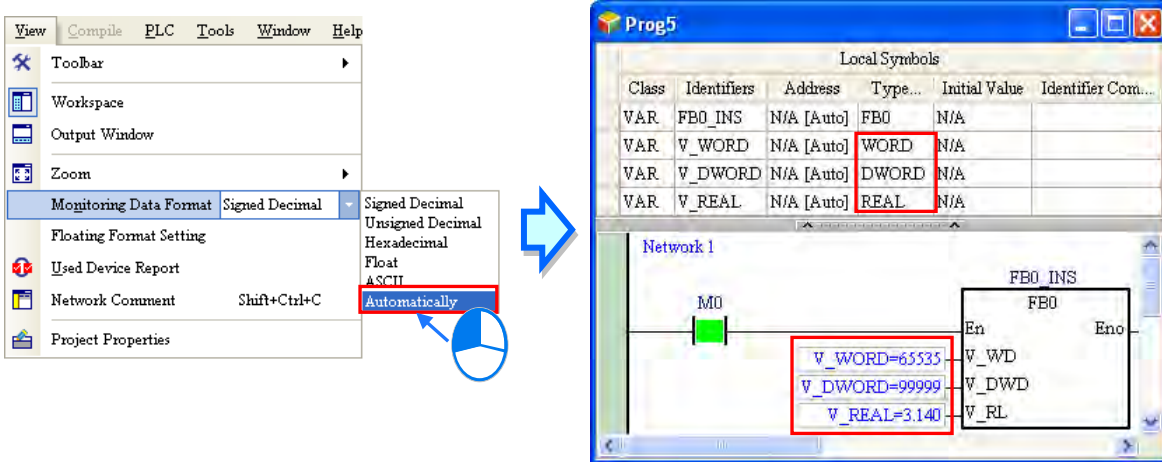


A sequential function chart in the program monitoring mode is shown below. Not only the sequential function chart itself is monitored, but also the actions and the transitions are monitored.

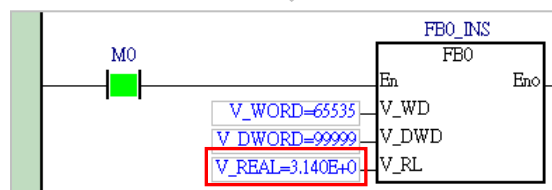
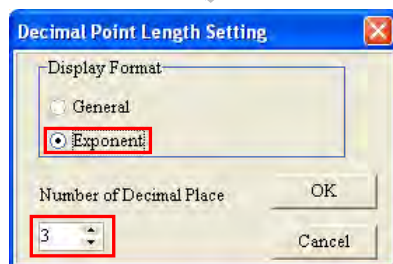
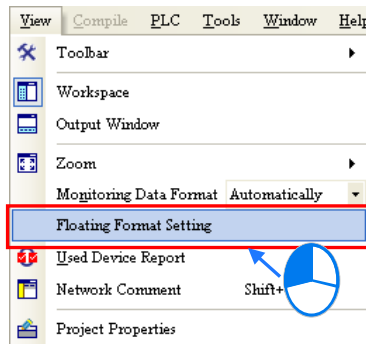


In the program monitoring mode, users can monitor not only the states of the contacts, but also the values in the registers. Besides, users can set the formats of the values monitored.

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.



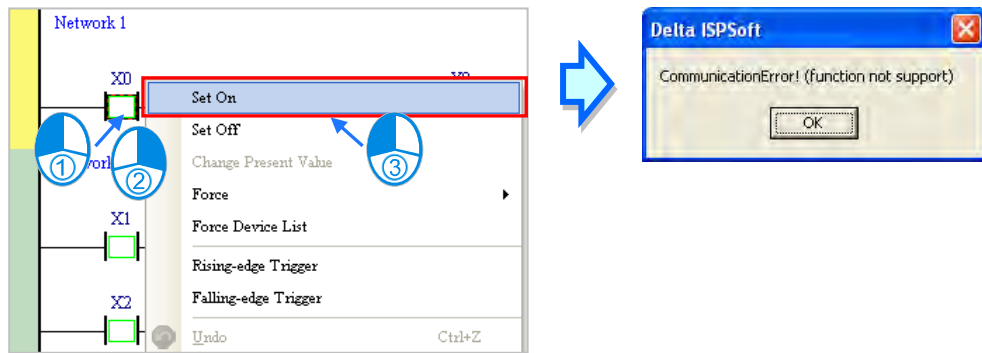
After users click **Floating Format Setting** on the **View** menu, they can set the formats of the floating-point numbers.




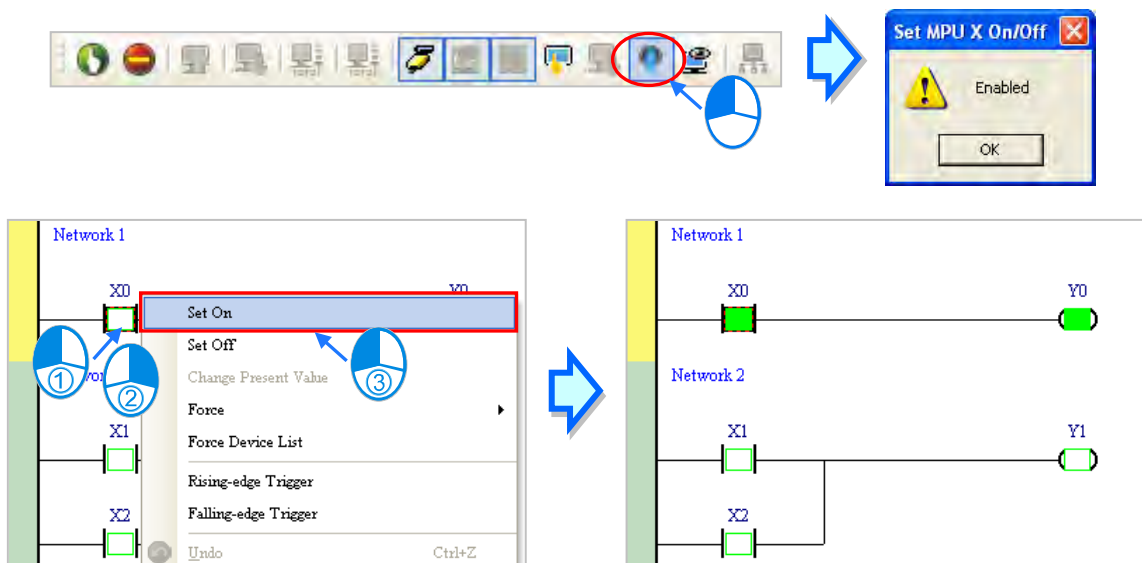
18

### 18.1.3 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.



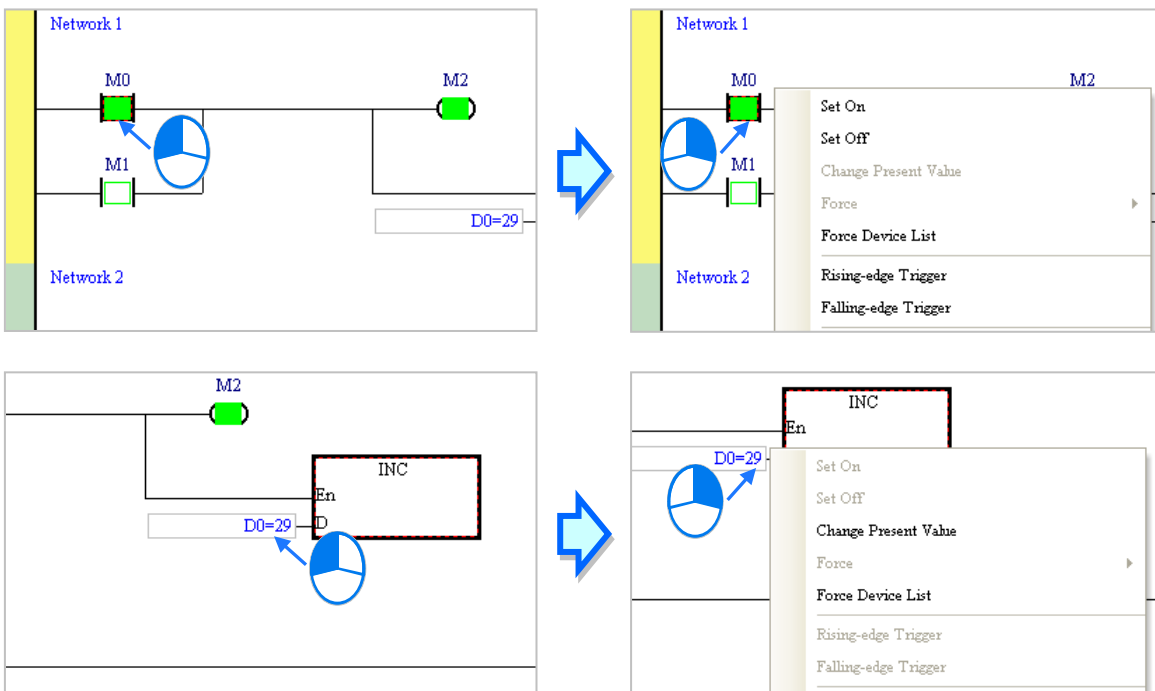
After the online monitoring function is disabled, the function of setting the X device will not be disabled automatically. As a result, be sure to disable the function of setting the X device before the online monitoring function is disabled lest the state of the X device should not be updated during 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.**

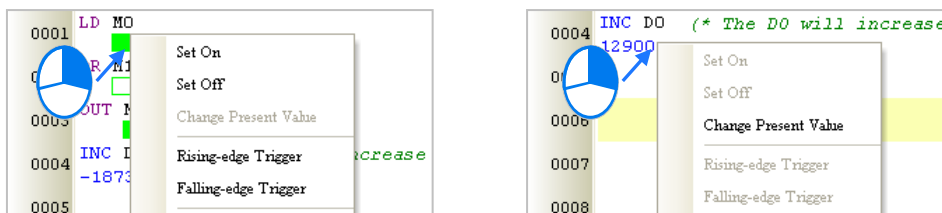
### 18.1.4 Monitoring a Program Online

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.

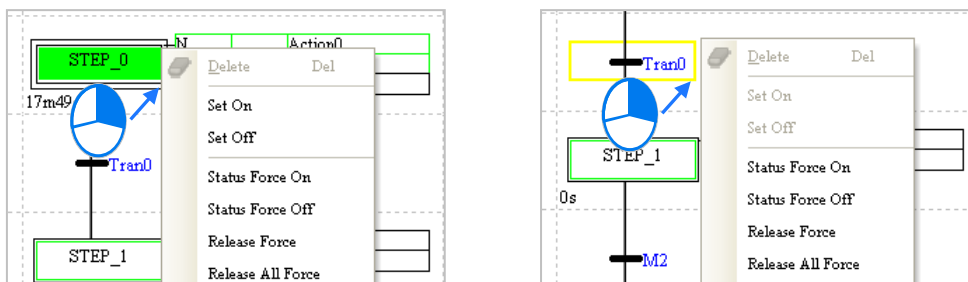


18

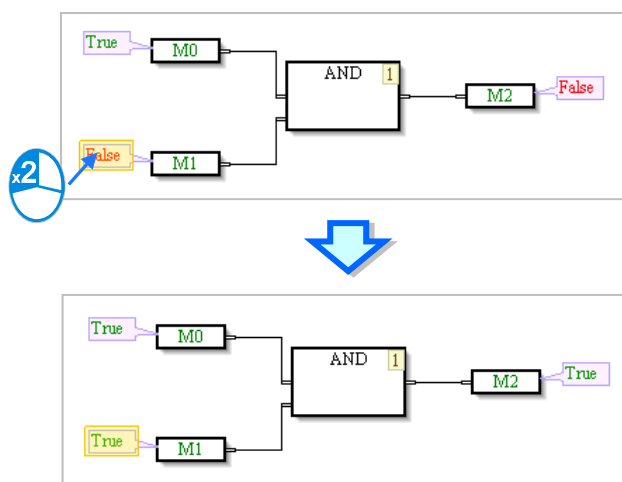
If users want to manipulate an operand on an instruction list or in a structured text, they can right-click the operand, and select an item on the context menu.



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.



If users want to manipulate a step or a transition in a continuous function chart, they can double click the shown status to switch from True to False or vice versa.



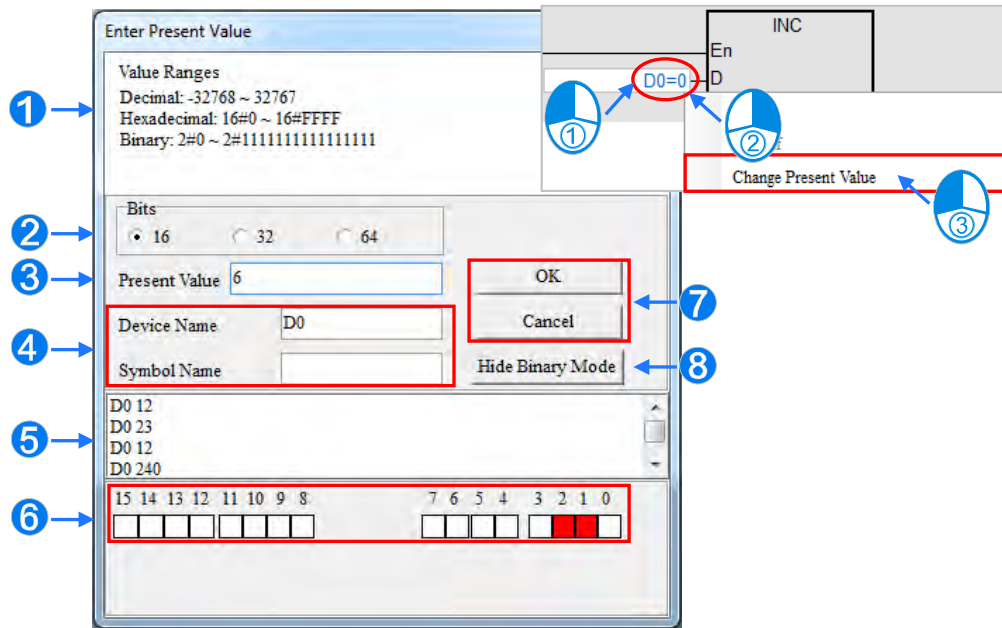
The items on the context menu are described below.

Item	Description
<b>Set On</b>	Setting the contact selected to ON
<b>Set Off</b>	Setting the contact selected to OFF
<b>Rising-edge Trigger</b>	No matter what the state of the contact selected is, the system set the contact to OFF, and then set it to ON.
<b>Falling-edge Trigger</b>	No matter what the state of the contact selected is, the system set the contact to ON, and then set it to OFF.
<b>Change Present Value</b>	Users can set the value of the operand selected.
<b>Force</b>	Forcing an input contact or output contact ON or OFF
<b>Force Device List</b>	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

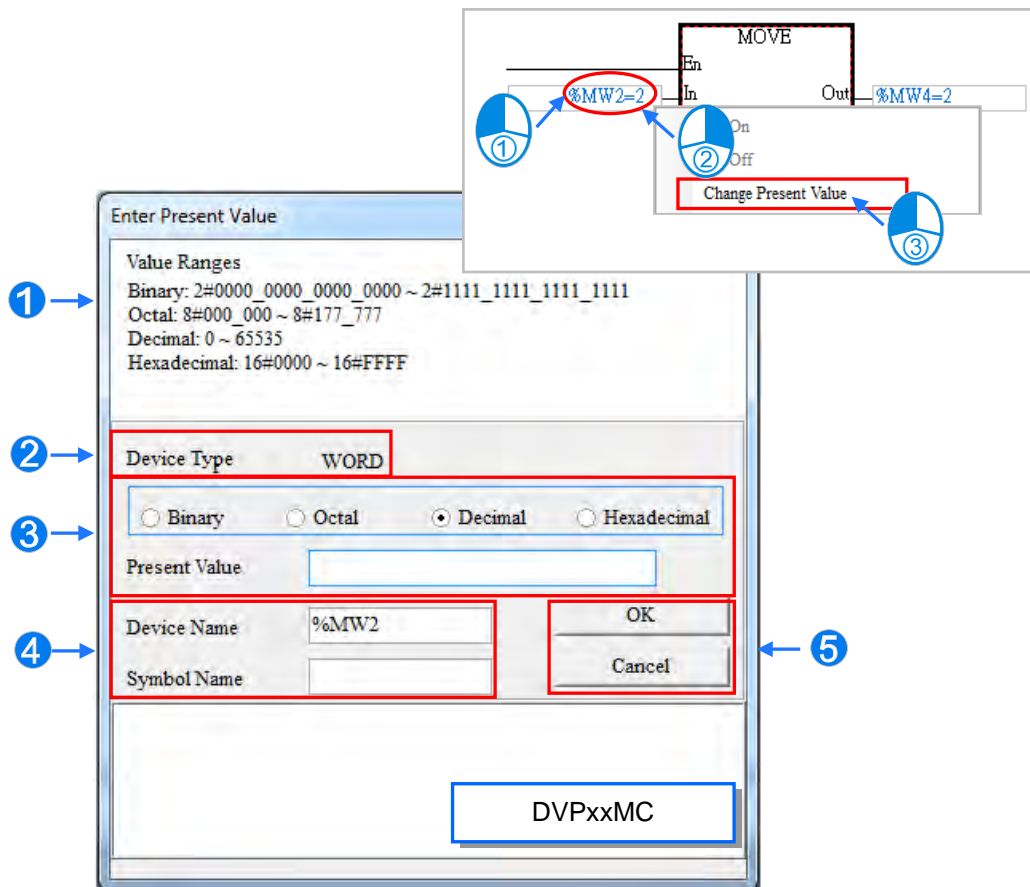
First, select an operand, and right-click the operand. Next, select **Change Present Value** on the context menu. Finally, set the value of the operand in the **Enter Present Value** window.



- ① Message.
- ② Users can select 16 bits/ 32 bits/ 64bits. (The content changes according to the selected PLC series.)
- ③ Input the present value to be changed.
- ④ Users can type a Device Name or Symbol Name for the change present 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.

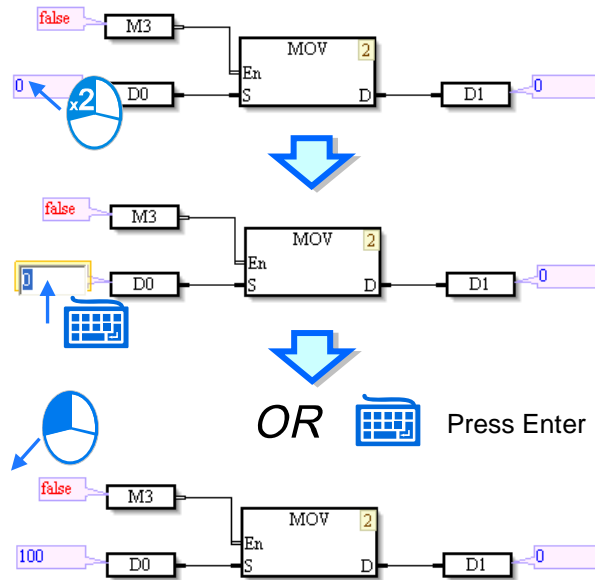


Users can select DVPxxMC series and the following image appears.



- 1 Message.
- 2 Users can select 16 bits/ 32 bits/ 64bits. (The content changes according to the selected PLC series)
- 3 Input the present value and select format including Binary/ Octal/ Decimal/ Hexadecimal.
- 4 Users can type a Device Name or Symbol Name for the change present value.
- 5 Press Enter to apply the setting values; press Cancel to close the setting window.

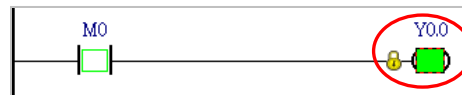
In the continuous function charts, double click the value that you'd like to change, the field will become editable. Type the value in and click on the blank area or press Enter on the keyboard to complete the editing.



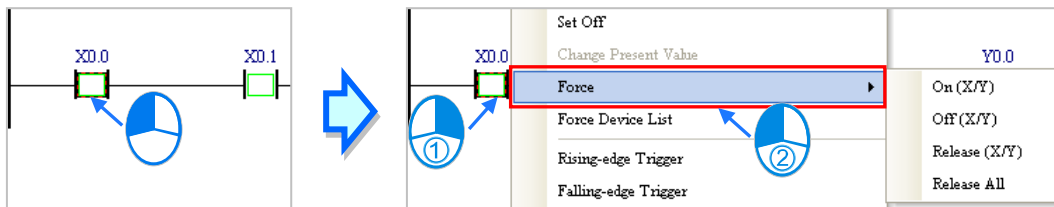
● **Force**

Only X contact and Y contacts can 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.

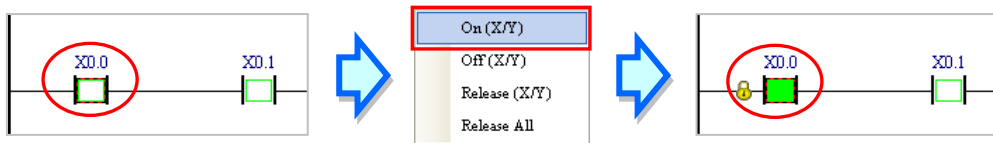
Y0.0 in the figure below is forced ON. It will be ON even if M0 is OFF. Besides, if a contact is forced ON or OFF, a lock symbol will appear at the left side of the contact.



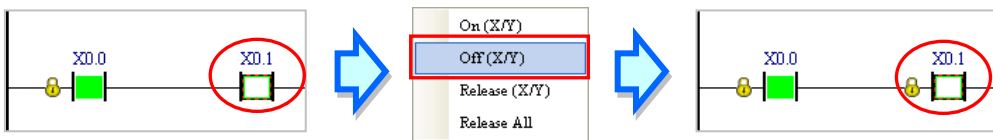
Click an X contact or a Y contact which will be manipulated, right-click the X contact or the Y contact, point to Force on the context menu, and select **On (X/Y)**, **Off (X/Y)**, **Release (X/Y)**, or **Release All**.



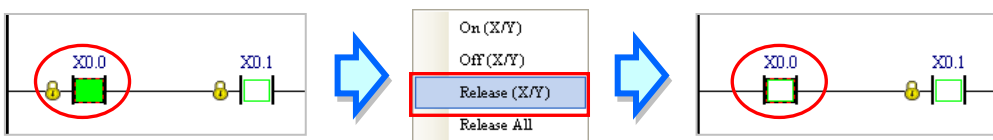
➤ **On (X/Y):** Forcing the X contact or the Y contact selected ON



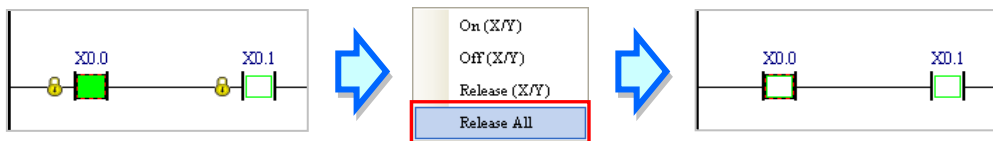
➤ **Off (X/Y):** Forcing the X contact or the Y contact selected OFF



➤ **Release (X/Y):** Releasing the contact from the locked state



➤ **Release All:** Releasing all the contacts from the locked states

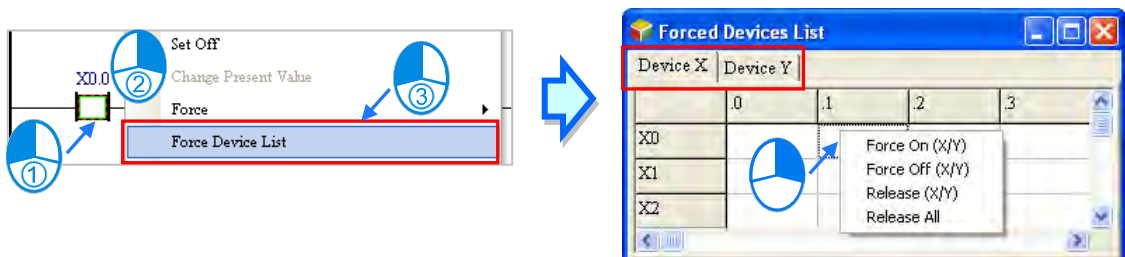


\*1. The mechanism for forcing contacts ON/OFF, and the limit on the number of contacts which can be forced ON/OFF vary from model to model. Please refer to the manual for the model selected for more information.

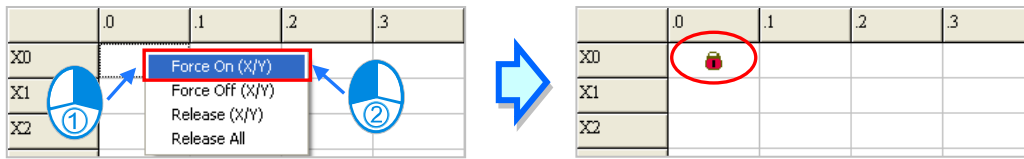
\*2. If the online monitoring function is disabled, the contacts will not be automatically released from the locked states. Please check whether the present state will cause any problem.

● **Force Device List**

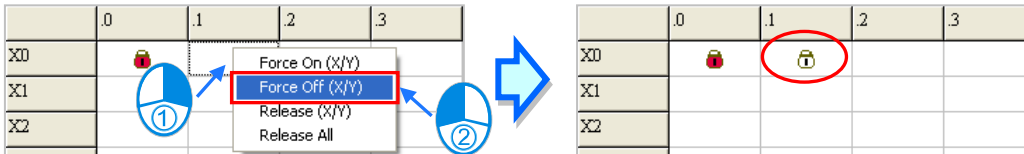
Right-click the program editing window, and then select **Force Device List** on the context menu. Click the **Device X** tab or the **Device Y** tab in the **Forced Devices List** window. Select a contact in the table, right click the contact, and select an item on the context menu.



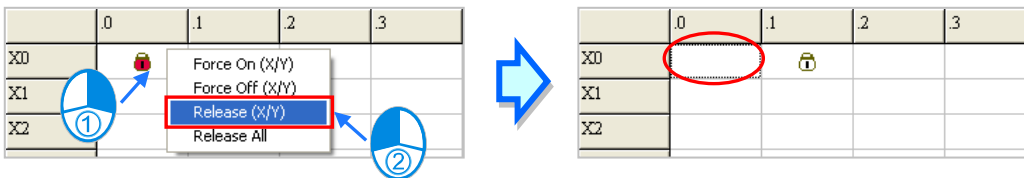
➤ **Force On (X/Y):** Setting the contact selected to ON



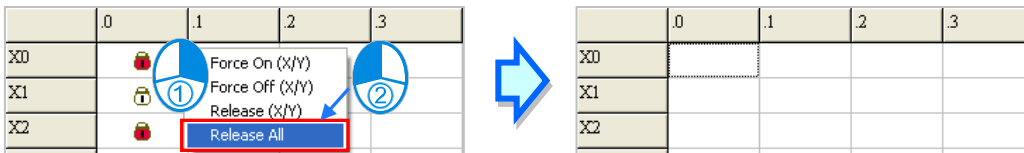
➤ **Force Off (X/Y):** Setting the contact selected to OFF



➤ **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.

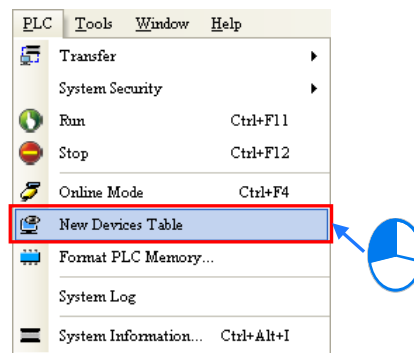
### 18.1.5 Device Monitoring Table

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.

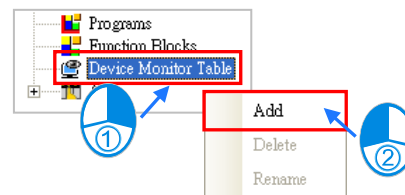
#### ● Method 1

Click **New Devices Table** on the **PLC** menu.




#### ● Method 2

Right-click **Device Monitoring Table** in the project management area and click **Add**.

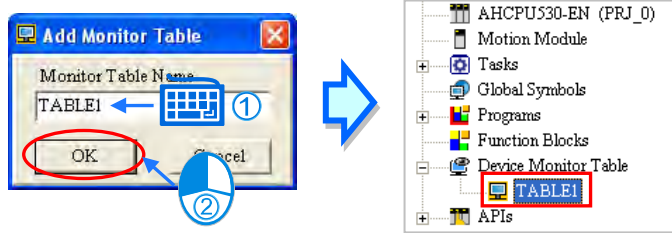


#### ● Method 3

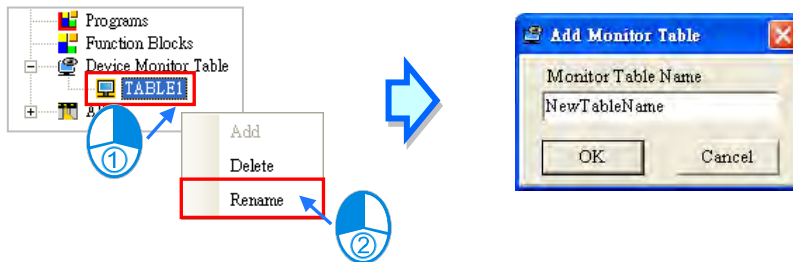
Click  on the toolbar.



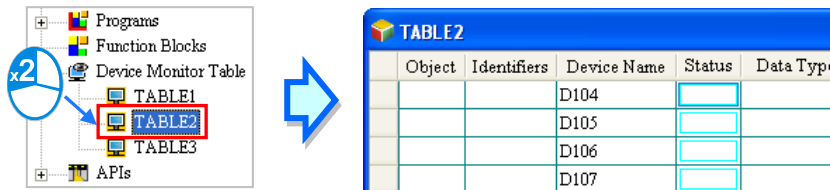
Type a table name in the **Add Monitor Table** window, and then click **OK**. An item will be under **Device Monitor Table** in the project management area, and a device monitoring window will appear in the working area. Users can manage the items in the window.



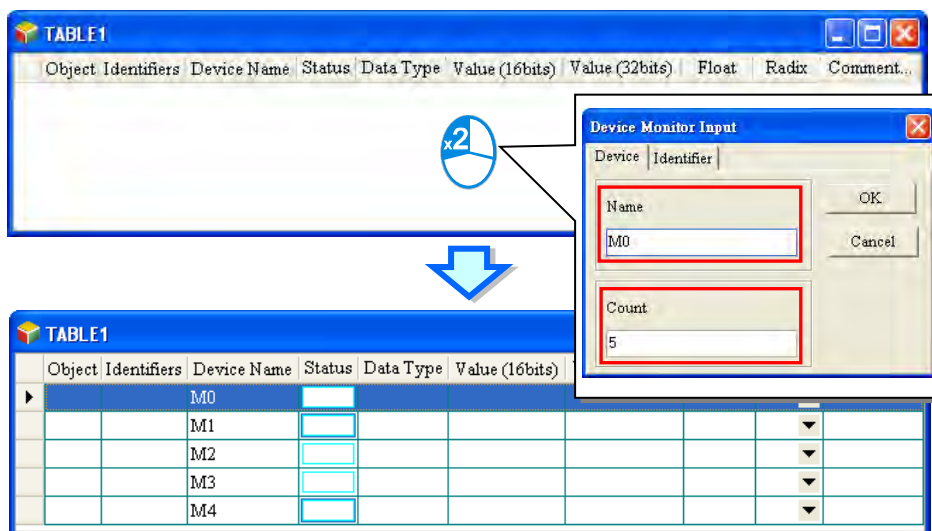
If users want to rename a device monitoring table created, they can click the device monitoring table in the project management area, right-click the device monitoring table click **Delete**. As for renaming a device, click **Rename** and type a new table name in the **Add Monitor Table** window.



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 any blank area 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.

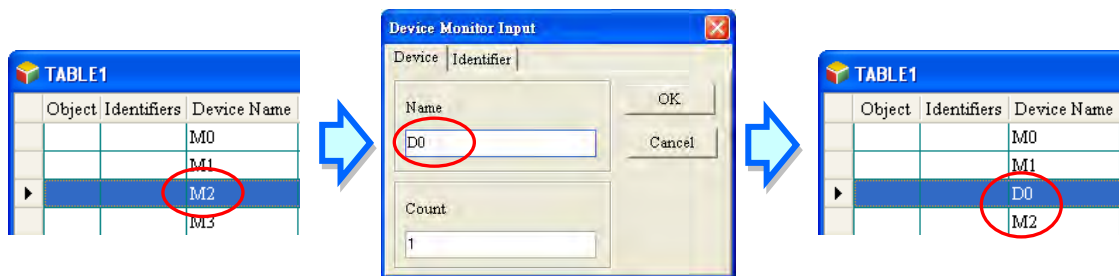


18

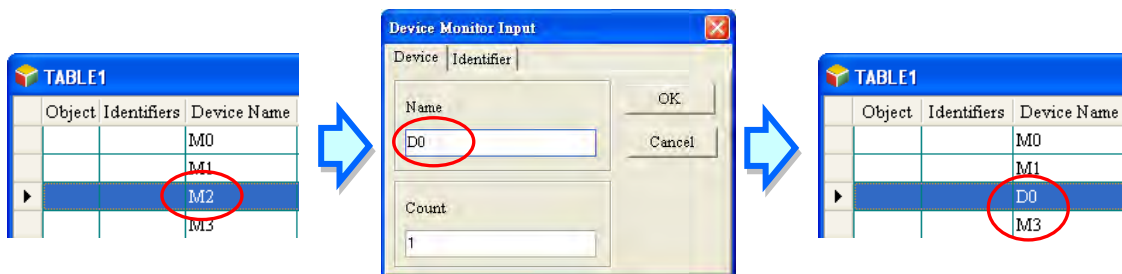
Users can press Insert on the keyboard to switch between inserting an item in the present device monitoring table and replacing an item in the present device monitoring table. The mode which is selected is displayed in the status bar in ISPSOft.



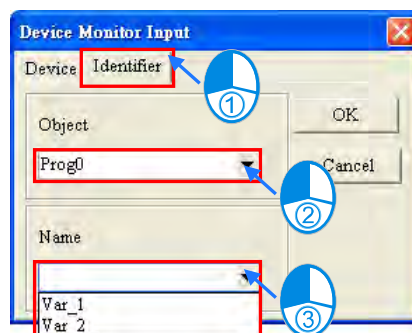
If the insertion mode is selected, the item added will be inserted above the item selected in the device monitoring table.



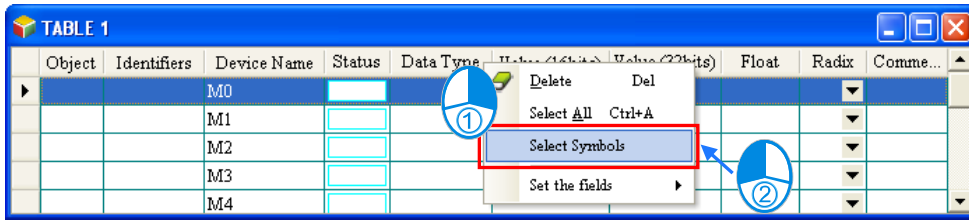
If the replacement mode is selected, the item added will overwrite the item selected in the device monitoring table.



Users can click the Identifier tab and add identifiers in this Device Monitor Input window.

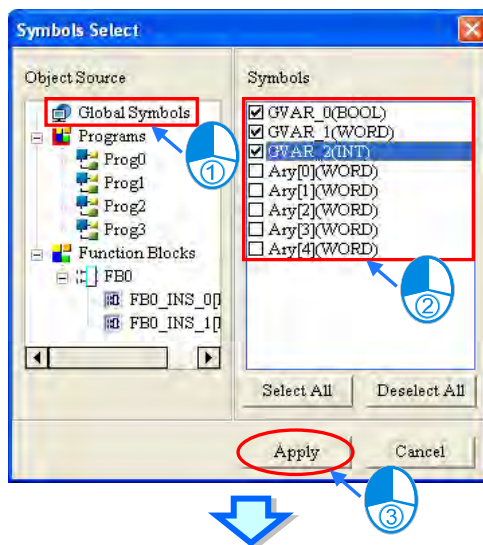


Users can add symbols as well as devices to a device monitoring table. After they right-click a device monitoring table, and select **Select Symbols** on the context menu, they can add symbols to the device monitoring table.



The users can select global symbols or local symbols in the POU's 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.

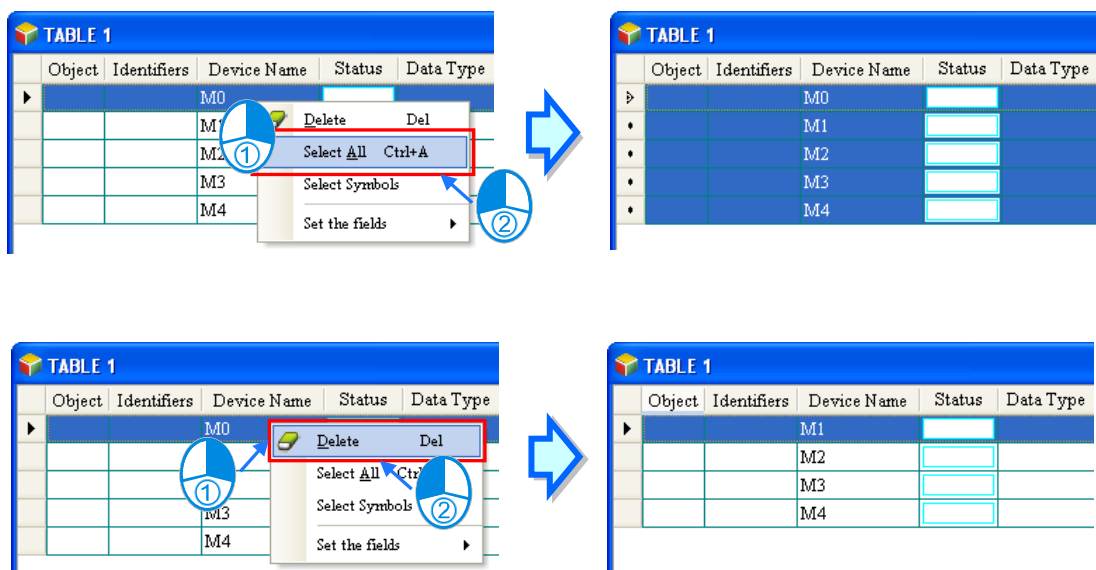


Object	Identifiers	Device Name	Status	Data Type	Value (16bits)	Value (32bits)	Float	Radix	Comment...
		M0							
		M1							
		M2							
		M3							
		M4							
Global Symbols	GVAR_0			BOOL					ERROR
Global Symbols	GVAR_1			WORD				Signed Decimal	DATA
Global Symbols	GVAR_2			INT				Signed Decimal	OFFSET

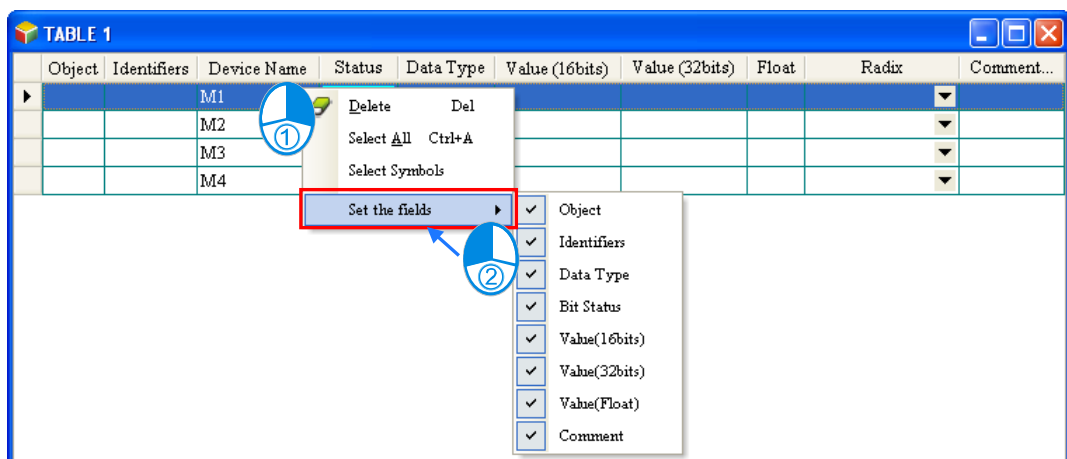
\*. The symbols whose data types are POINTER, the symbols whose data types are T\_POINTER, the symbols whose data types are C\_POINTER, and the symbols whose data types are HC\_POINTER can not be monitored.



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.



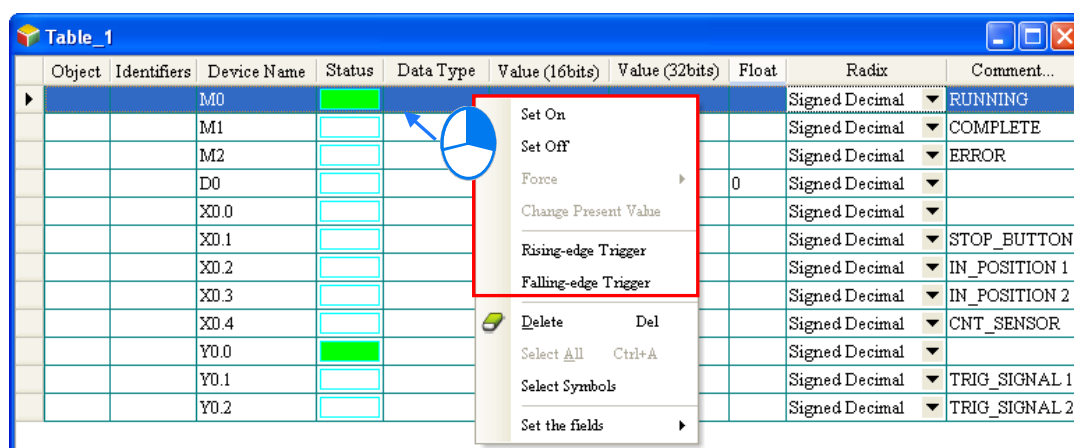
The users can hide some columns in a device monitoring table. If the users want to hide some columns in a device monitoring table, they can right-click the device monitoring table, point to **Set the Fields**, and unselect items which correspond to the columns they want to hide. Please refer to the description of the columns below for more information.



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

<b>Column</b>	<b>Description</b>
<b>Object</b>	The sources of the symbols are displayed.
<b>Identifier</b>	The identifiers of the symbols are displayed.
<b>Device Name</b>	The names of the devices monitored are displayed.
<b>Status</b>	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.
<b>Data Type</b>	If the objects monitored are symbols, the data types of the symbols will be displayed in this column.
<b>Value ( 8 bits)</b>	When the devices and the symbols are monitored, the 8-bit values are displayed in this column.
<b>Value (16 bits)</b>	When the devices and the symbols are monitored, the 16-bit values are displayed in this column.
<b>Value (32 bits)</b>	When the devices and the symbols are monitored, the 32-bit values are displayed in this column.
<b>Value ( 64 bits)</b>	When the devices and the symbols are monitored, the 64-bit values are displayed in this column.
<b>Float (32-bit floating-point)</b>	When the devices and the symbols are monitored, the 32-bit floating-point numbers are displayed in this column.
<b>Float (64-bit floating-point)</b>	When the devices and the symbols are monitored, the 64-bit floating-point numbers are displayed in this column.
<b>Radix</b>	Users can select the formats of the values monitored.
<b>Comment...</b>	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 18.1.4 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.


## **18.1.6 Online Editing Function and Online Update Function**

\*. Currently, NOT supported by DVPxxMC

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.

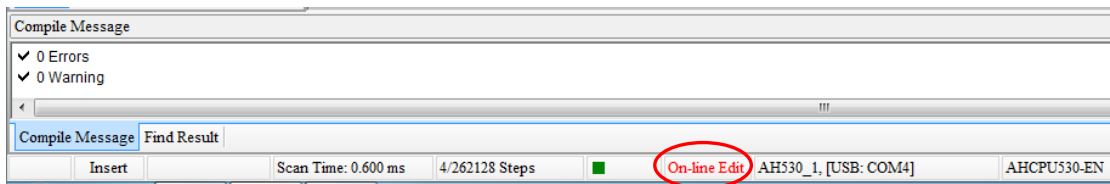
<b>Programming language</b>	<b>Description</b>
Ladder diagram	One network is edited at a time. Users are allowed to add new networks. After a new network is added, the network which can be edited will be the new network, that is, the previous networks can not be edited.
Function block diagram	One network 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	A sequential function chart can not be modified online, but actions and transitions


Programming language	Description
function chart	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.
Continuous function chart	No limit on the editing.
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.




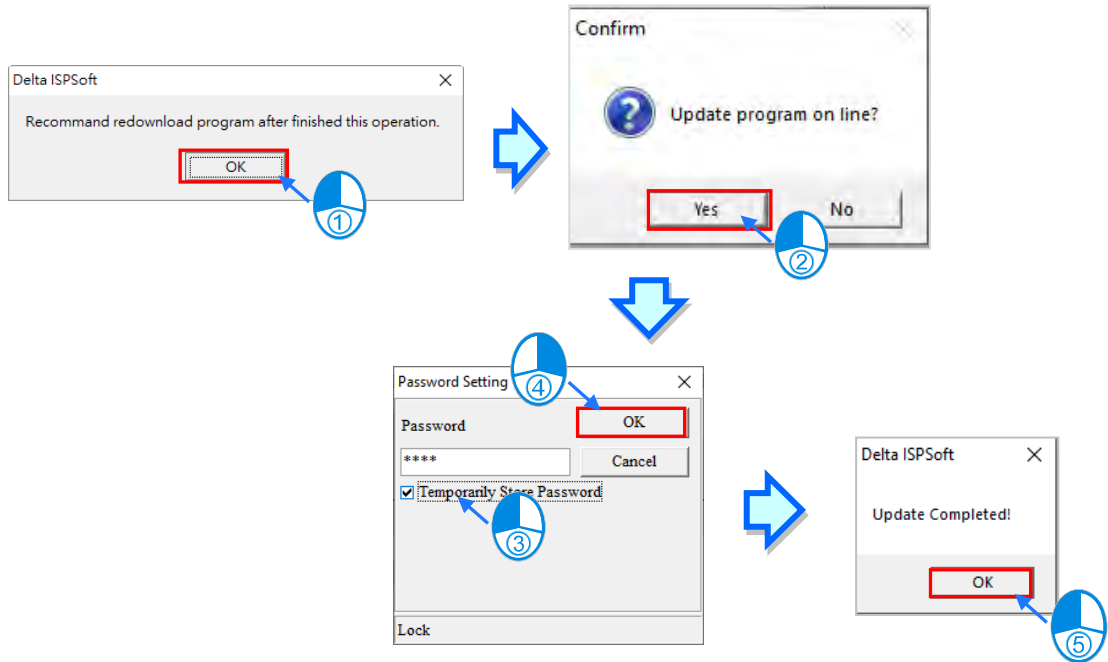
If the online editing function is enabled, **On-line Edit** will be displayed in the status bar in ISPSOft.



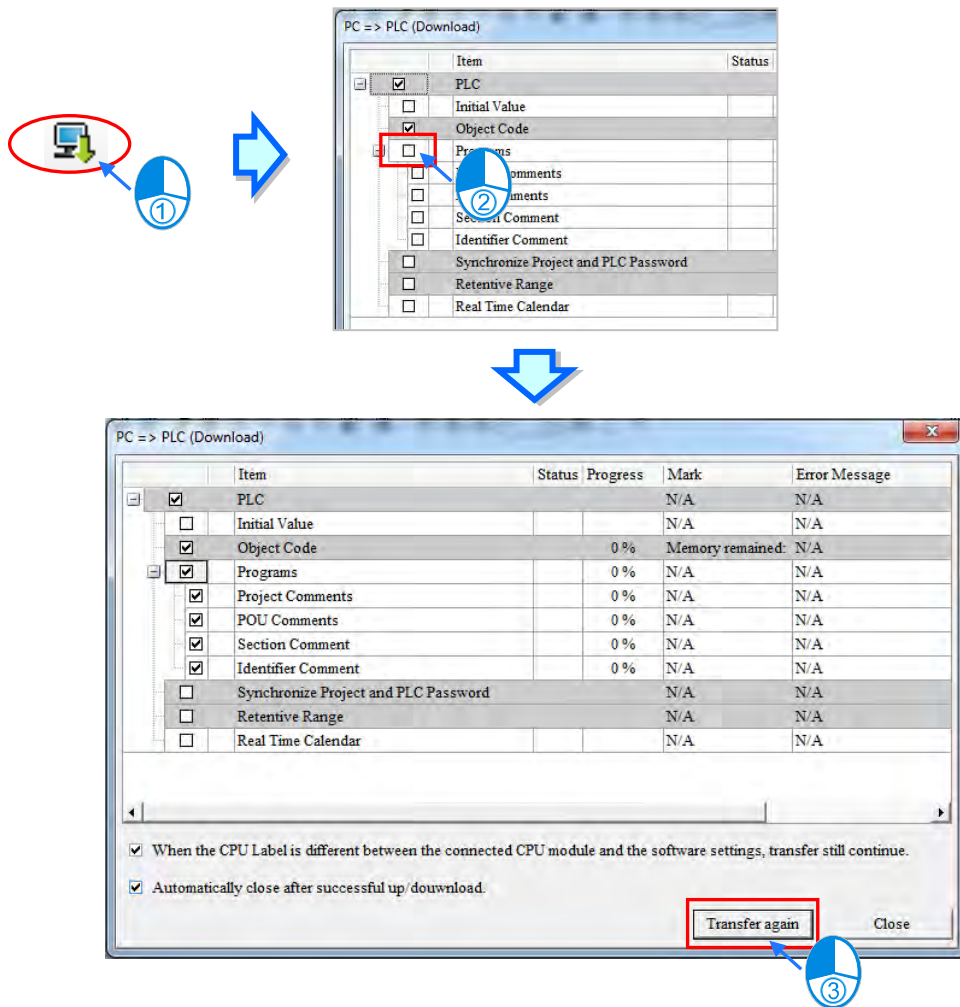
If the editing of the program is complete and the program is compiled successfully, the program will be downloaded to the PLC after users click  on the toolbar. If the program is not compiled, the system will ask users to compile the program.



When users select DVP series and click  on the toolbar, the screen will appear **Recommend redownload program after finished this operation** window. Click **OK** and a confirmation shows up, asking you to **Update program on line**, click **Yes** and a **Password Setting** window appears. Tick the option **Temporarily Store Password** and click **OK**, after that a confirmation shows up to inform you the update is complete, click **OK**.



Click  and select **Programs** to download the PLC source code for synchronization.





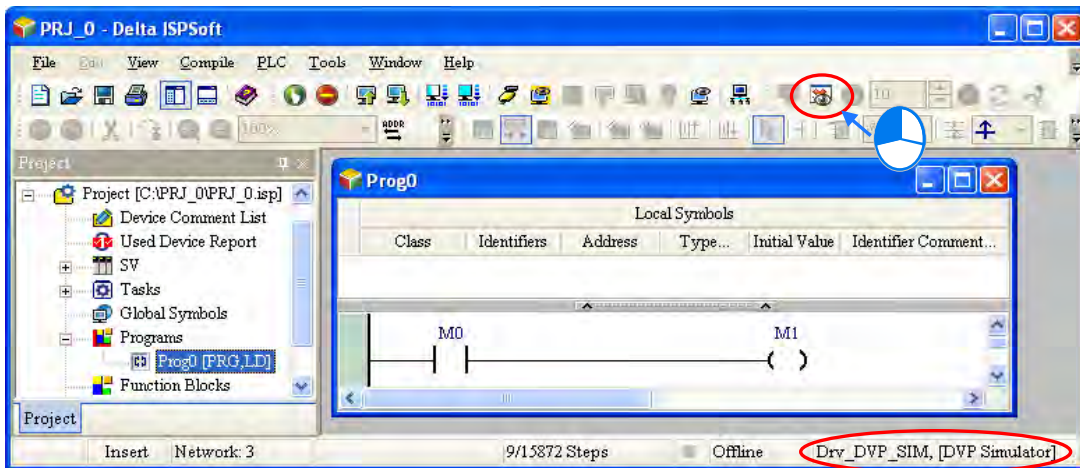
## 18.2 Debugging Mode for DVP Series

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.

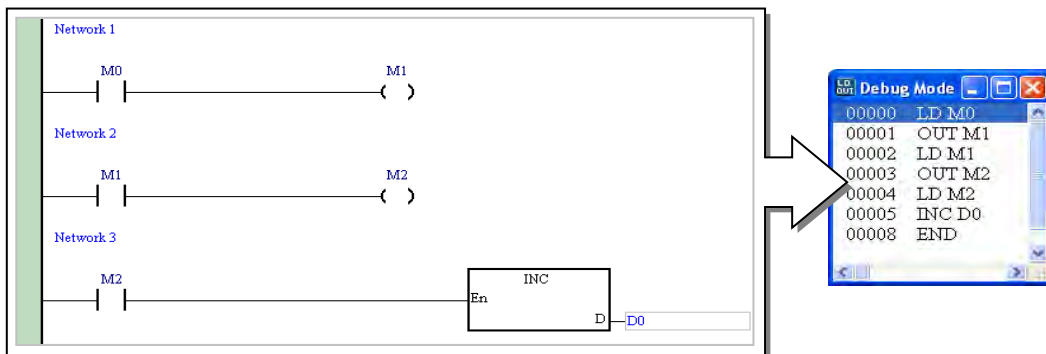
- (a) The debugging mode for a DVP series PLC can be executed only if the connection type is a simulator.
- (b) If the online mode is enabled, the debugging mode can not be enabled. However, the device monitoring function can be enabled after the debugging mode is enabled.

### 18.2.1 Enabling the Debugging Mode for DVP Series

The connection type must be a DVP simulator. Please refer to section 2.4 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.

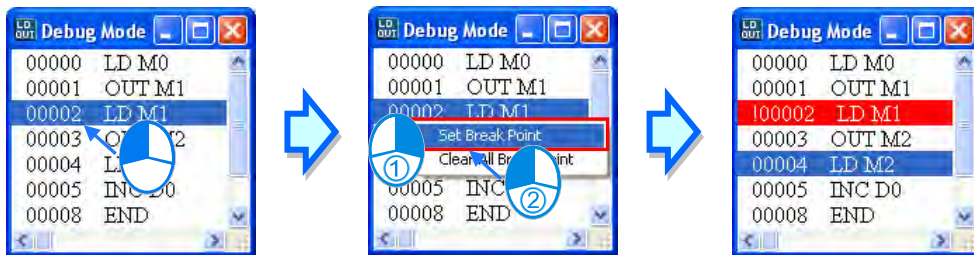


After the program in a DVP series PLC is translated into an instruction list, the instruction list will appear in the debugging window for the DVP series PLC. The part of the instruction list which is on a blue ground in the debugging window is the position where the execution of the program stops. The position where the execution of the program stops is the part of the program which has not been executed.

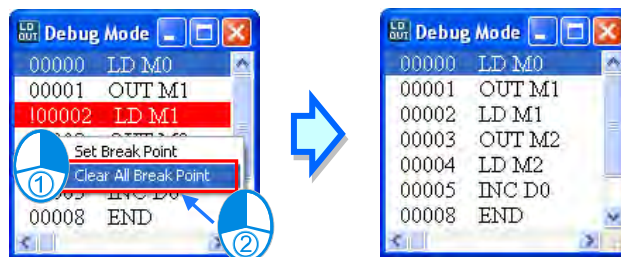


### 18.2.2 Adding and Clearing Breakpoints

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 mark appears at the beginning of the line to which a breakpoint is added, and the line is on a red ground.






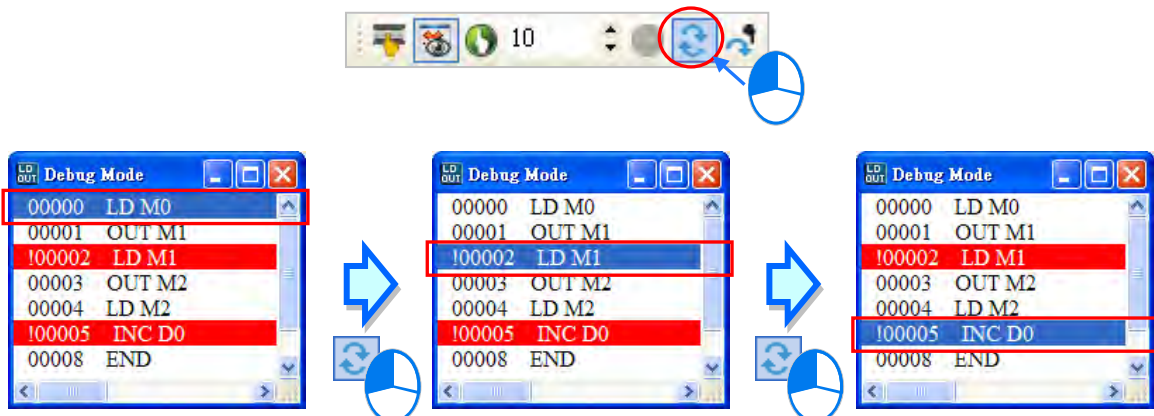
If users want to clear the breakpoints in the debugging window, they can right-click a line, and select **Clear All Break Points** on the context menu.




### 18.2.3 Execution of the Program in the Debugging Mode

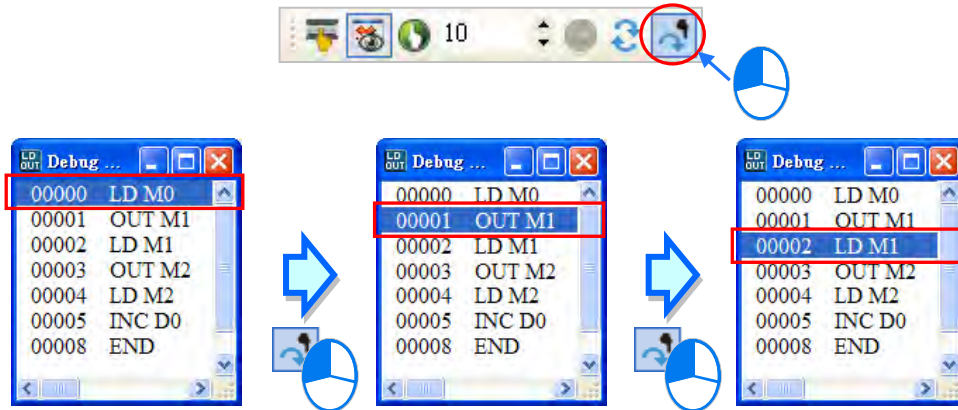
#### • Continuous execution

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.




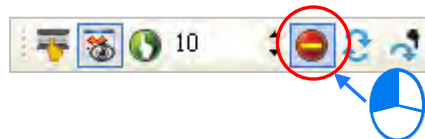
● **Setting the number of times the program is executed**


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  on the toolbar. The maximum number which can be set is 32767. Click  on the toolbar after the setting is complete.

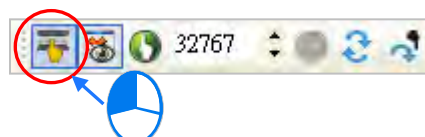


● **Stopping the execution of the program and resetting the execution order**

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 from 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.



## 18.3 Debugging Mode for AH/AS Series

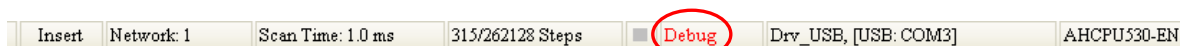
### 18.3.1 Enter Debugging Mode

The debugging mode which ISPSOft provides for AH/AS series CPU modules supports all the programming languages. (Currently, when the AH560 Redundant System series synchronizes, it does NOT provide debugging mode.) 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.

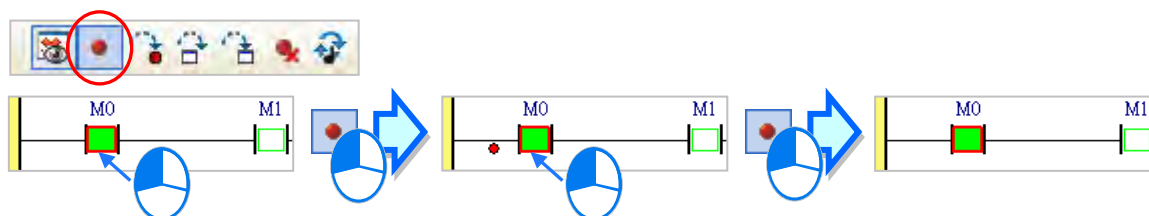



If the debugging mode is not enabled, but the CPU module is in the debugging mode, the debugging mode can not be enabled. (When the CPU module is in the debugging mode, **Debug** is displayed in the status bar in ISPSOft.) The reason is that the debugging mode was not disabled normally last time or another 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.





### 18.3.2 Adding and Clearing Breakpoints

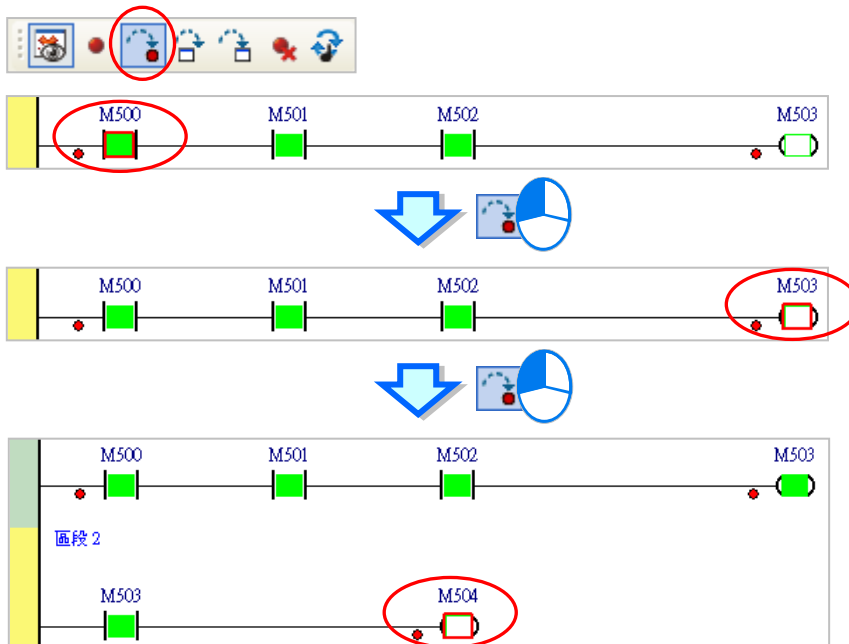
Ten breakpoints at most can be added to the program in a project for an AH500 series CPU module. After users click a position in a program editing window and click  on the toolbar, a breakpoint will be added to the position. If the users click the position again and click  on the toolbar, the breakpoint 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.

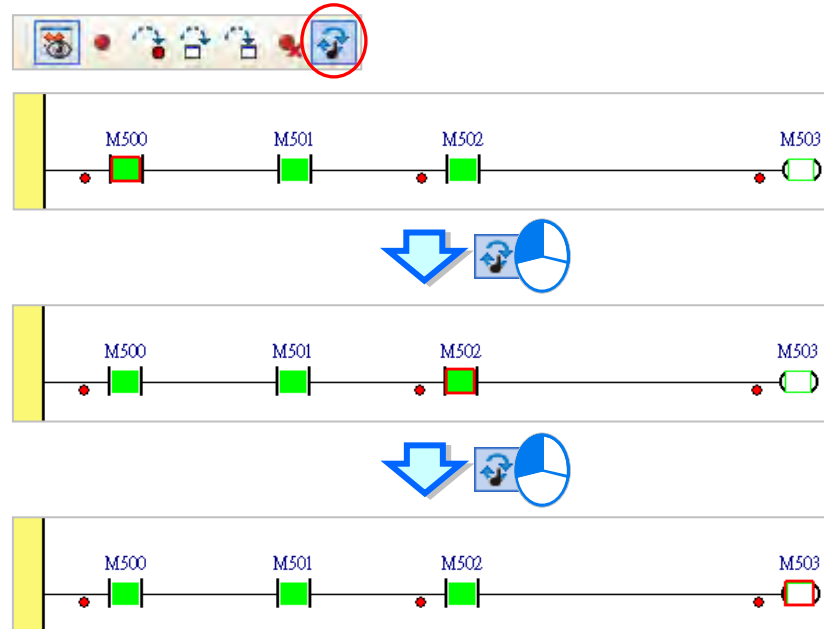


After  on the toolbar is clicked, the execution of the program will stop at a breakpoint. If users click  again, the execution of the program will continue, and will stop at the next breakpoint. The position where the execution of the program stops is the part of the program which has not been executed.









### 18.3.3 Continuous Execution

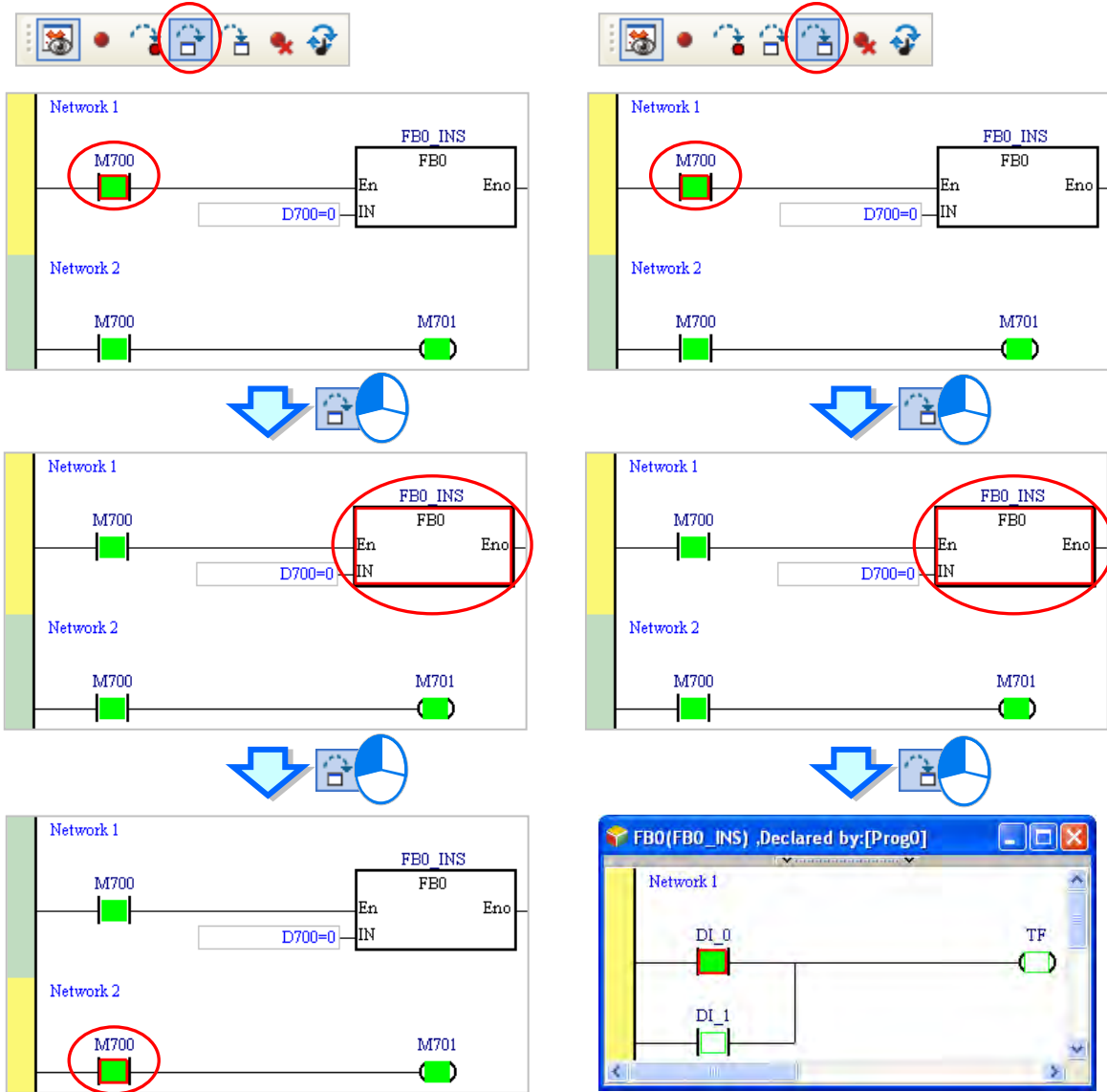
If  on the toolbar is clicked, the execution of the program will stop at a breakpoint. If users click  again, the execution of the program will continue, and will stop at the next breakpoint. If there are no breakpoints, the program will be scanned repeatedly after  is clicked.



### 18.3.4 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.

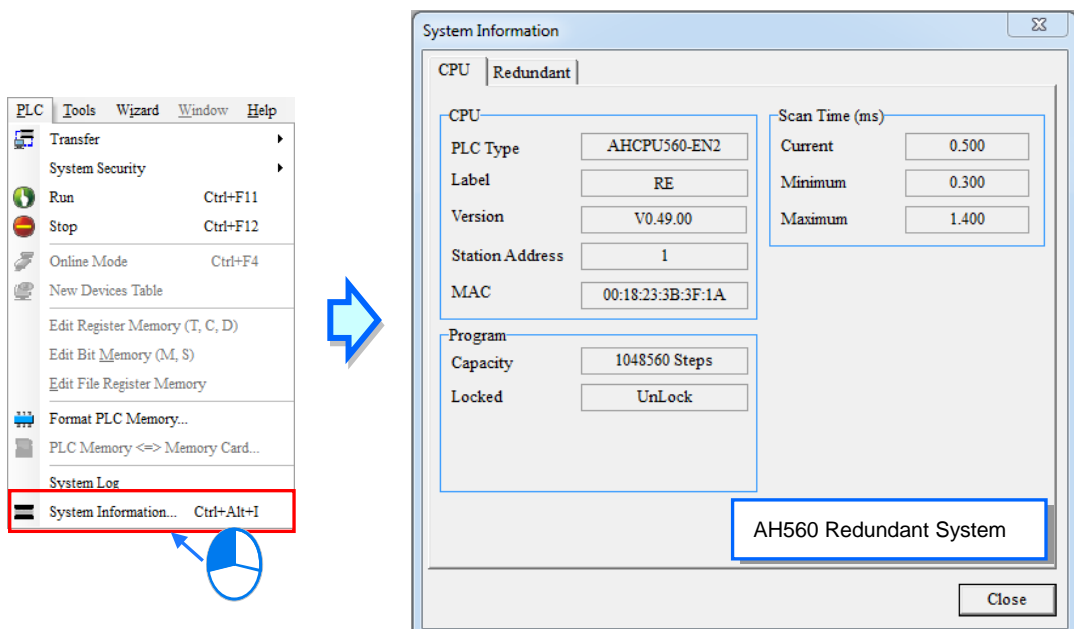
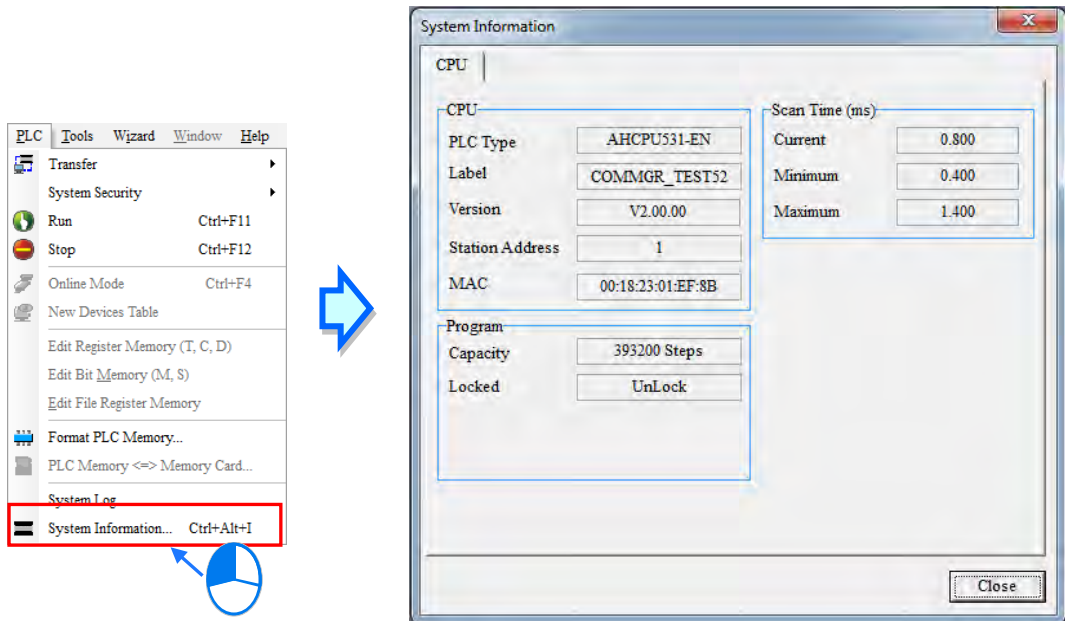
The different between  and  is shown below.



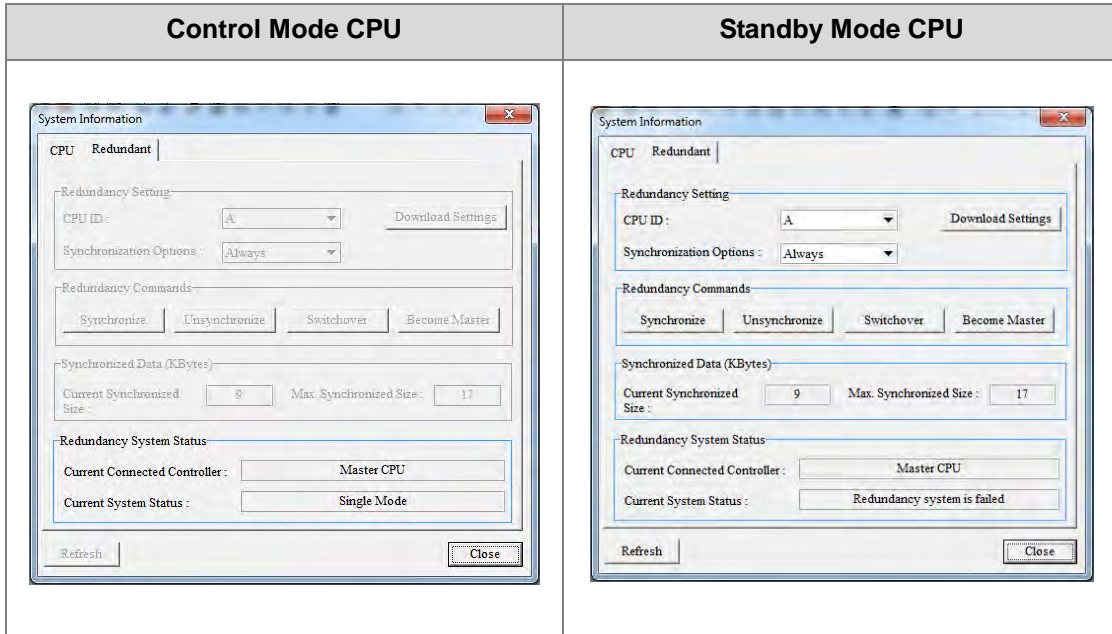
## 18.4 Checking the Status of a PLC

### 18.4.1 System Information

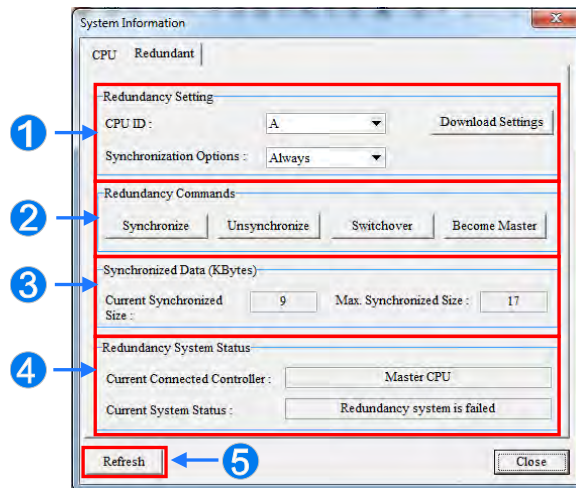
If ISPSOft is connected to the PLC normally, the system will retrieve information from the PLC after users click **System Information...** on the **PLC** menu. For AH560 redundant system series, a **Redundant** page is added.



In **Redundant** page provides current redundant system status or related settings. The current system can only operate under **standby mode CPU**; only redundancy system status appears in **Control Mode CPU** (see below).



Introducing Redundant page content:



18

❶ CPU ID: an id for A and B controller to check whether the controller is switched in the redundant system and uses API2901 RCS command for setting or reading.

**Synchronization Options:** contains 'Always' and 'Condition' which also allows users to check the redundant system and synchronization settings, please refer to related product manuals.

**Download Settings:** Finish setting CPU ID and Synchronization Options, users need to click this button to download to PLC hosts.

❷ **Synchronize:** triggers the master CPU to check the standby CPU and will only initiate synchronization under qualified standby CPU.

**Unsyncronize:** cancelling the standby CPU

**Switchover:** refers to switching controllers under qualified standby CPU.

**Become Master:** allows the standby CPU to become the master CPU when the master CPU does not exist.

③ The section displays synchronized data in Current Synchronized Size and **Max. Synchronized Size (Kbytes)**, using special registers 'SR18' ~ 'SR21' for confirmation.

④ **Current Connected Controller:** can connect to either a master CPU or a standby CPU

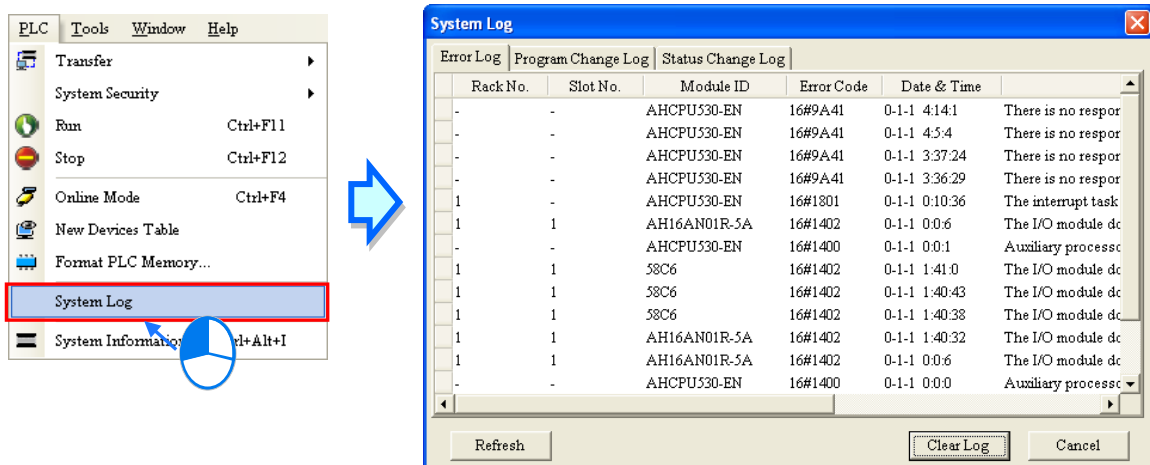
**Current System Status:** contains **Checking the redundancy system**, **Redundancy system is ready**, **Control mode CPU**, **Redundancy system is failed** and **Disconnected with synchronous optical network**.

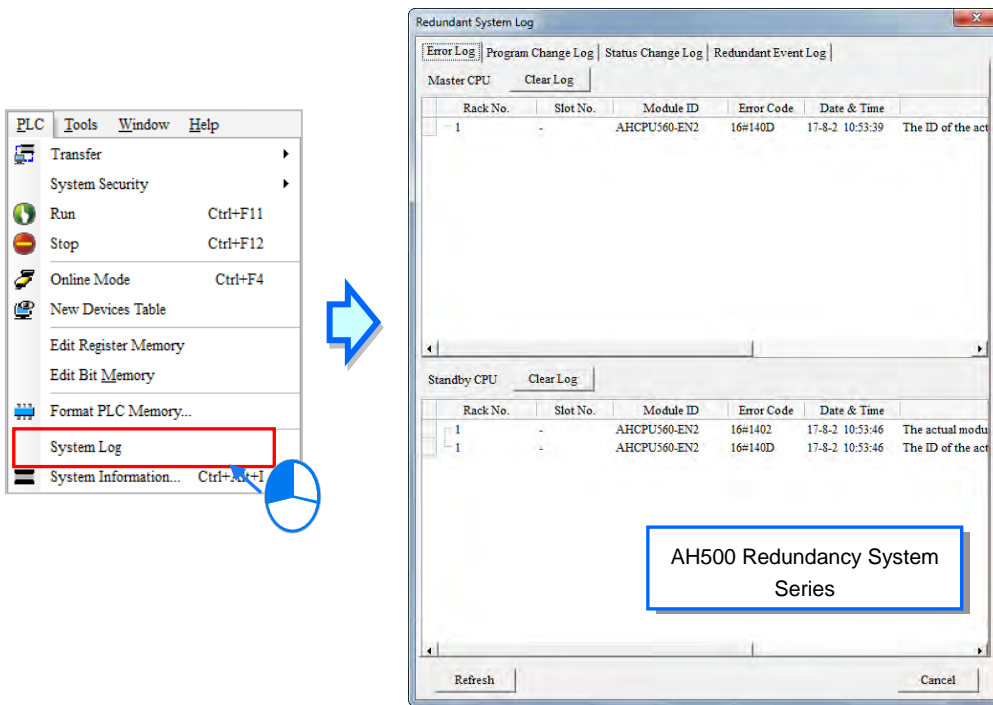
⑤ **Refresh:** used to update all information in the Redundant page.

\*. For more information containing AH560 redundant system series and **Redundant** page, please refer to related product manuals.

## 18.4.2 PLC System Log

Under normal connection with hosts, users can select **System Log** from the **PLC** menu to view logs regarding modification and errors; For AH560 redundant system, users can check the logs of both master and standby controllers, if no master or standby controller is detected, the display areas are in gray color. The **System Log** page contains several pages and can use the labels on top to switch to different functions. The data in system log is not updated in real-time, but users can click **Refresh** to update.

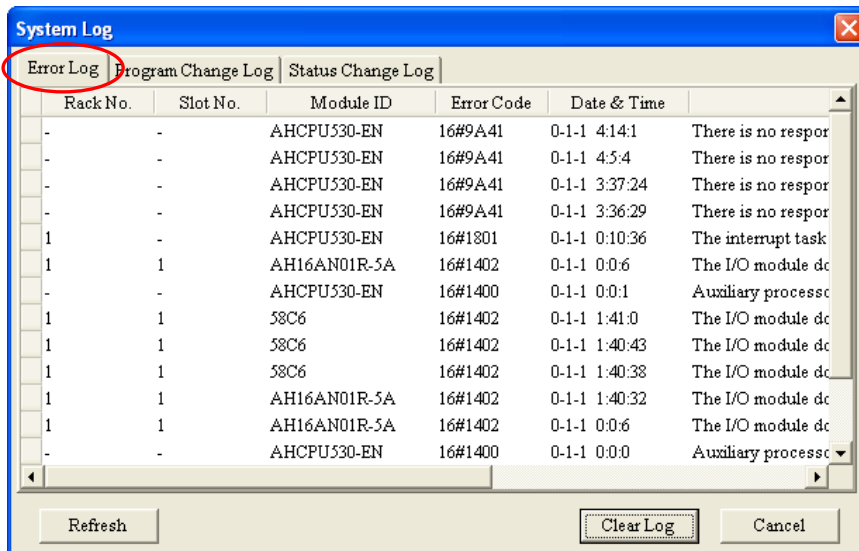




● Error Log

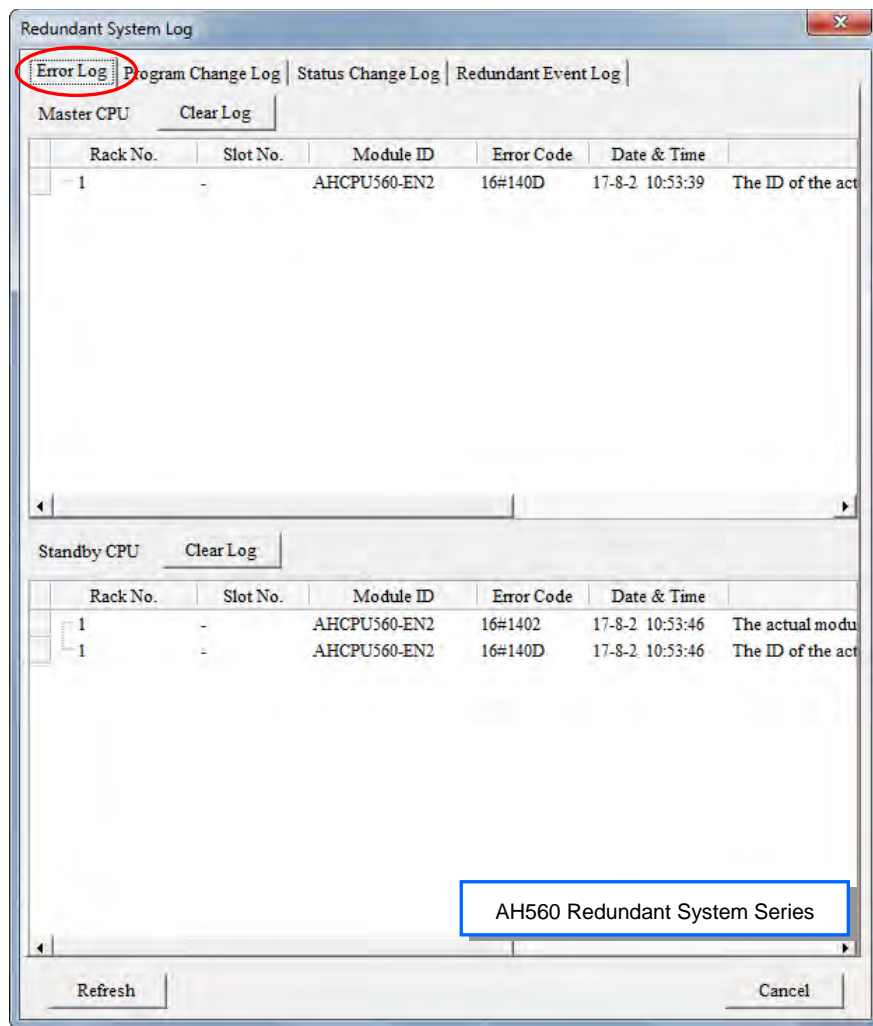
Users can view hosts and all modules error codes, date & time as well as error description. In addition, it will also contain the error module ID and backplane no. as well as slot no.


When users click **Refresh**, the system will retrieve data from the host and update the data in the log; click **Clear Log**, to clear the table and internal host error log as well as reset the host error status. But, when the error reason still exists, the host remains in error status and new error log is formed.

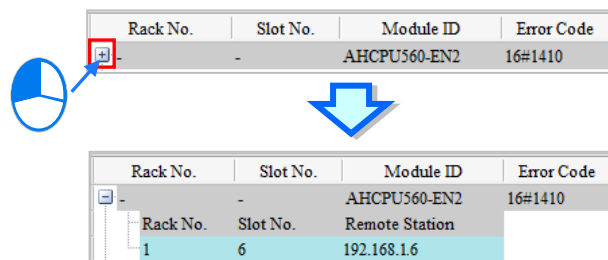


18

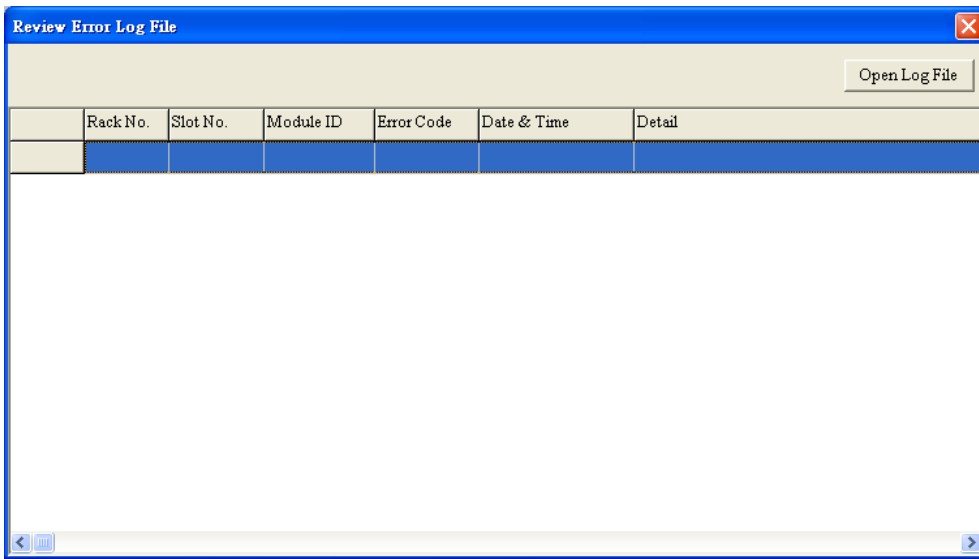




When the CPU module is connected to a remote module, users can click  to view the remote module's rack no., slot no. and RTU remote IP address from the error logs.

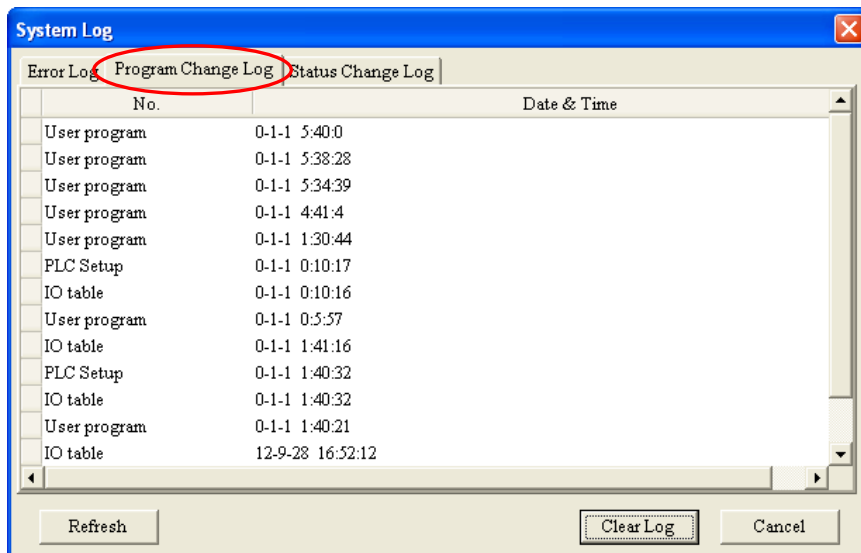


Error logs can be stored in the SD card and then the logs can be moved to the PC for better management. Refer to character 3 for more information on how to save error logs to the SD card. After that users can go to Tools > Review Error Log File and press **Open Log File** to select the path where the error log saved to see the contents of the error log.

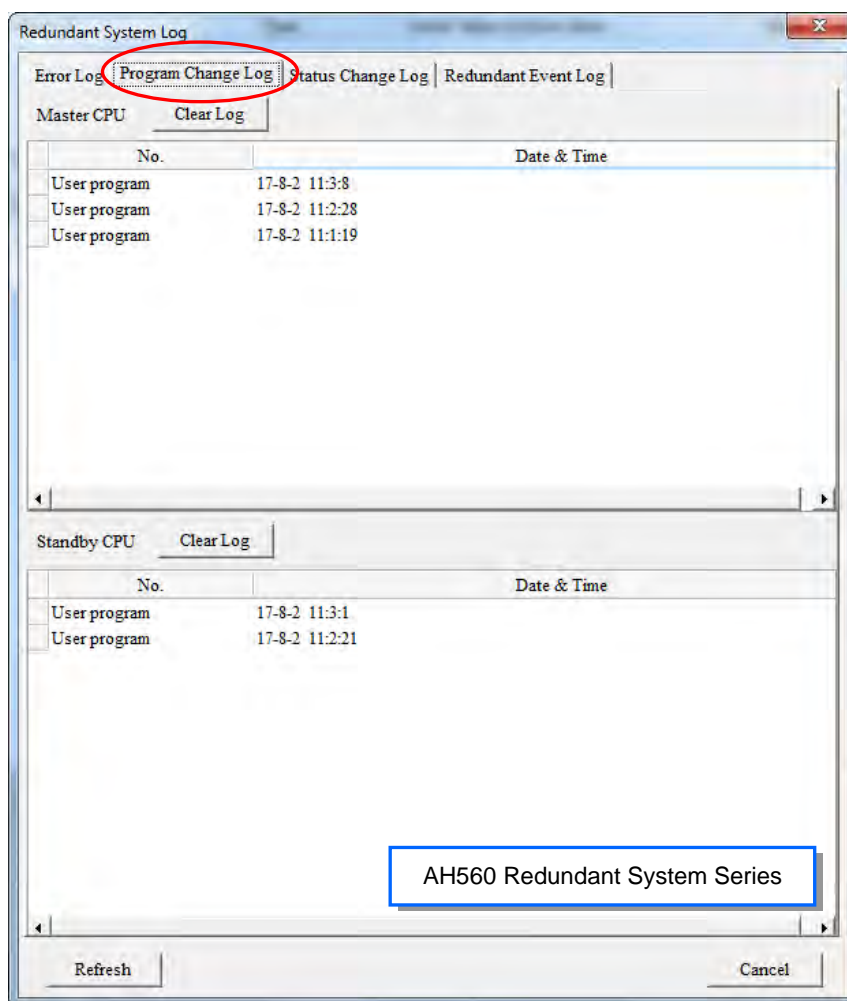


● Program Change Log

After users click the **Program Change Log** tab, they can view the log related to the downloading/uploading of programs and parameters. After users click **Refresh**, the system will retrieve data from the CPU module, and the data displayed in the window is updated. After users click **Clear Log**, the program change log in the pop-up window and the program change log in the CPU module is cleared.

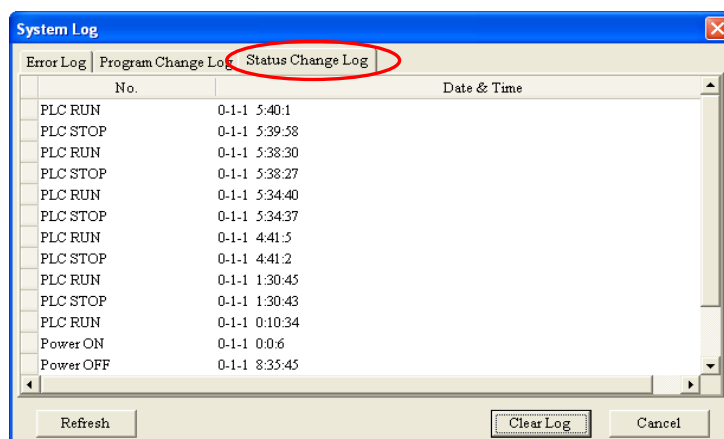


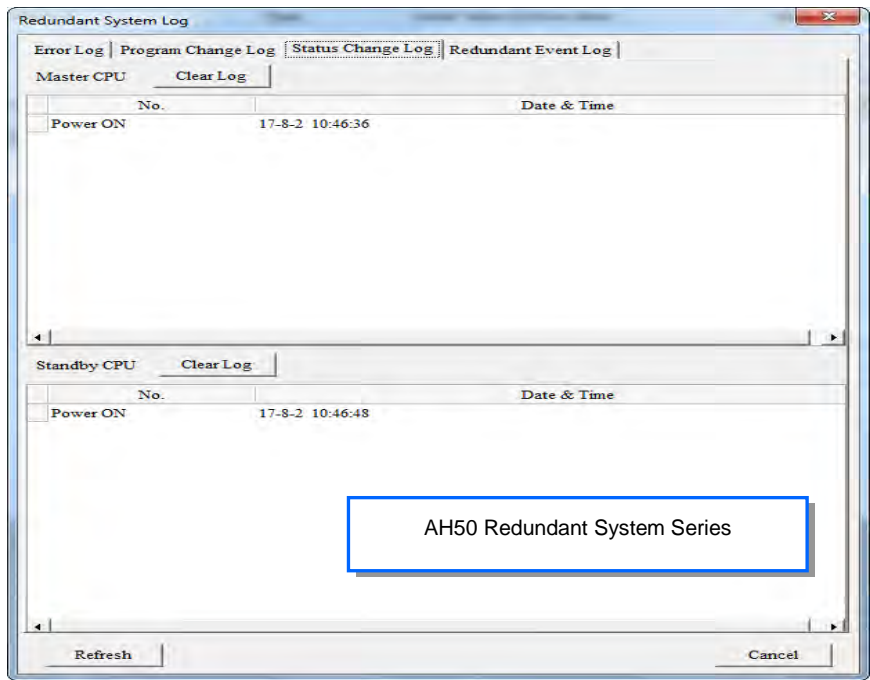
18



### ● Status Change Log

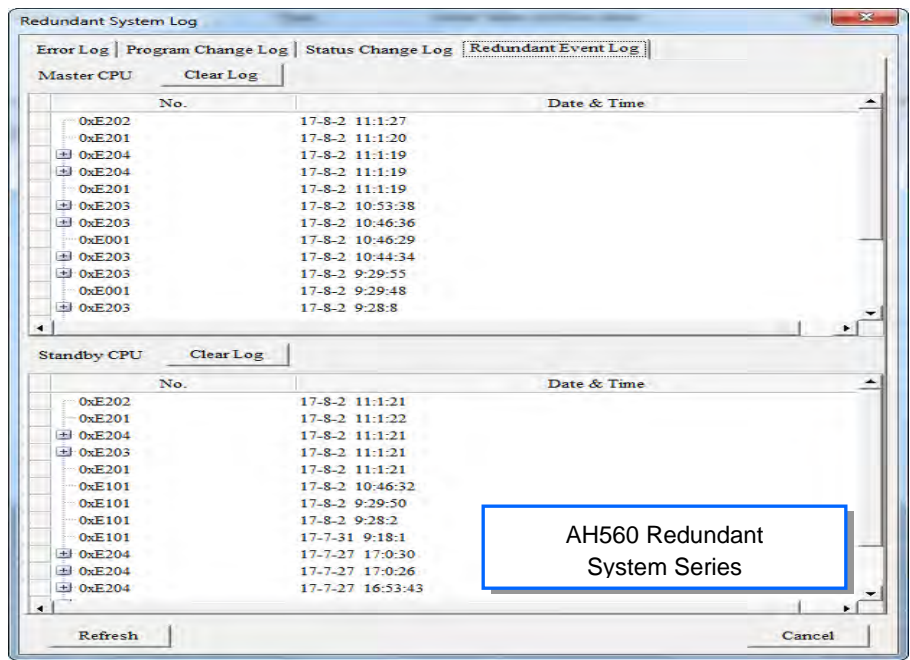
After users click the Status Change Log tab, they can view the log related to the change of the status of the CPU module. After users click **Refresh**, the system will retrieve data from the CPU module, and the data displayed in the pop-up window is updated. After users click **Clear Log**, the status change log in the pop-up window and the status change log in the CPU module is cleared.





● Redundant Event Log

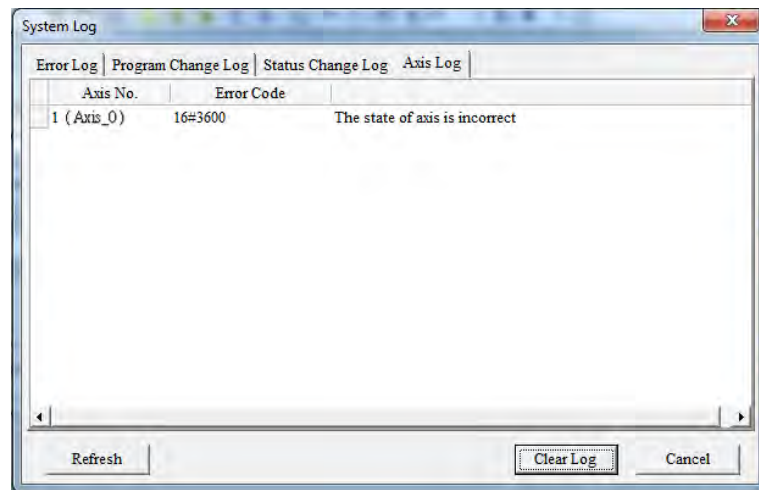
The tab displays redundant event logs of the master and standby CPU. After users click **Refresh**, the system will retrieve data from the master and standby CPUs, and the data displayed in the pop-up window is updated. After users click **Clear Log** in the master CPU, the redundant event log in the pop-up window and the redundant event log in the master CPU is cleared; if users click Clear Log in the standby CPU, the redundant event log in the pop-up window and the redundant event log in the standby CPU is cleared.



\*. The redundant event log only supports the AH560 redundant system series.

**● Axis Log**

After users click the Axis Log tab (the axis number and the axis name inside the bracket are shown in the Axis. No column), they can view the log related to the axis. After users click **Refresh**, the system will retrieve data from the CPU module, and the data displayed in the window is updated. After users click **Clear Log**, the axis log in the window and the axis log in the CPU module is cleared.



\*.Axis log is only available for AHxxEMC.

## 18.5 3D Chart

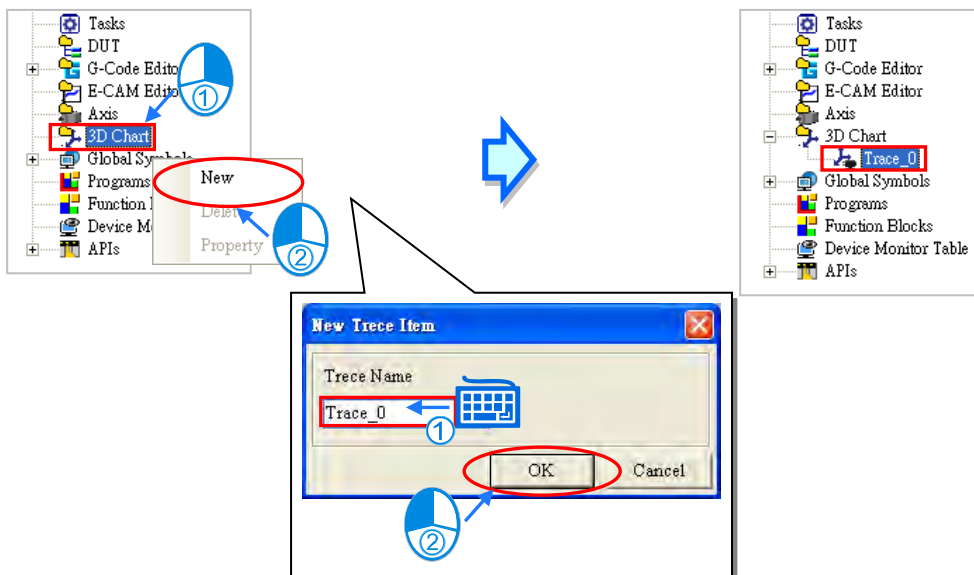
### 18.5.1 Features

During the operation of the multi-axis interpolation in the monitoring mode, users can use the 3D chart to see the motion traces, variables and devices, providing a way to check the accuracy of the motion and a way to analyze the value trend.

\*. This function is only available for AHxxEMC.

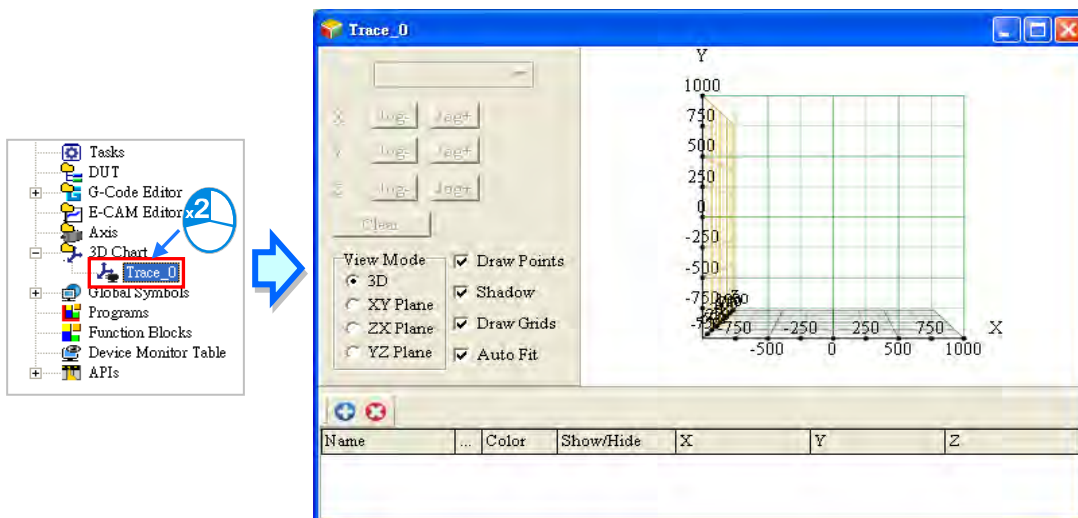
### 18.5.2 Creating a 3D Chart

Right-click the **3D Chart** in the project management area and then click **Add** to bring out a New Trace Item window, type the trace name in the field to create a trace. One project can add up to 10 sets of 10 traces.

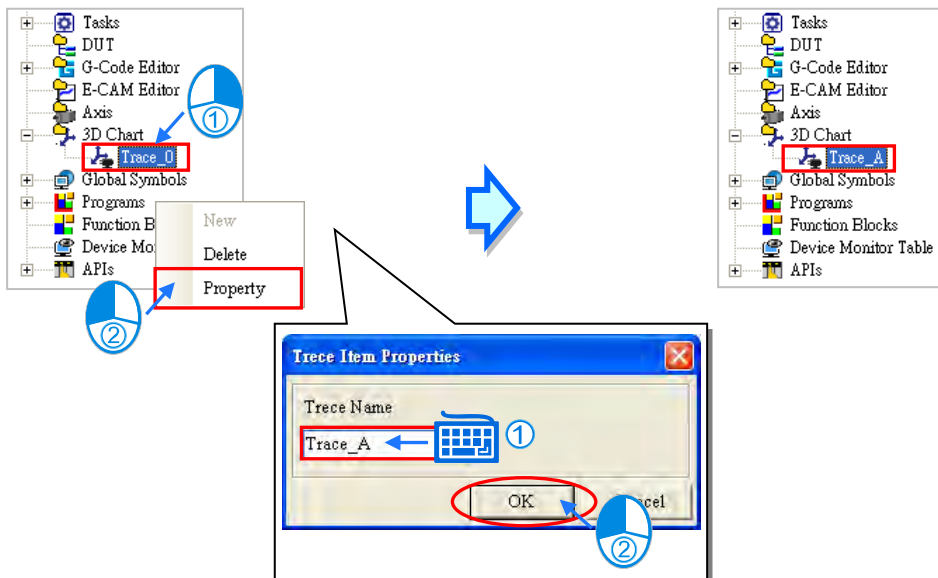


Once a trace is created, the 3D chart window will show up automatically. Users can also find it under the option 3D Chart and double click it to open its editing window.

18

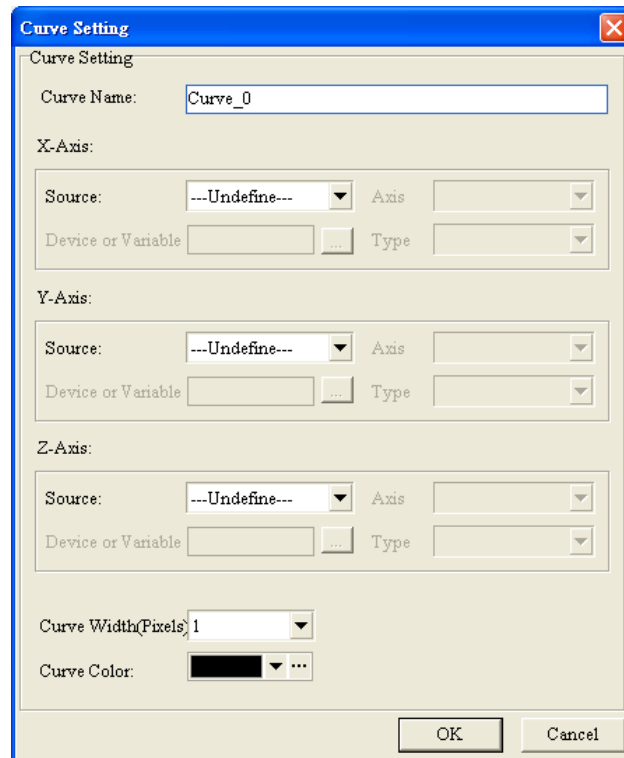


Right-click the created trace, users can use the options to delete or click Property to rename the selected trace.

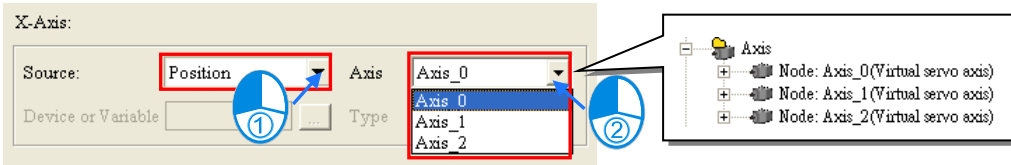


### 18.5.3 Creating a Curve

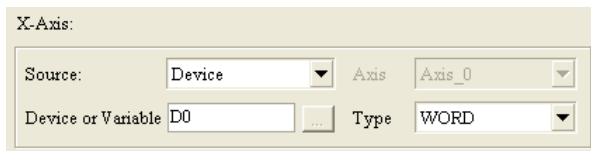
Click the button on the 3D chart editing page to bring out a Curve Setting page.



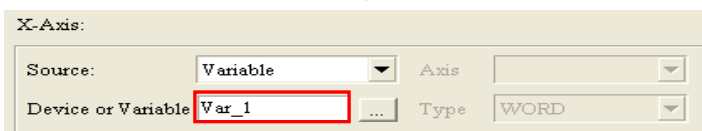
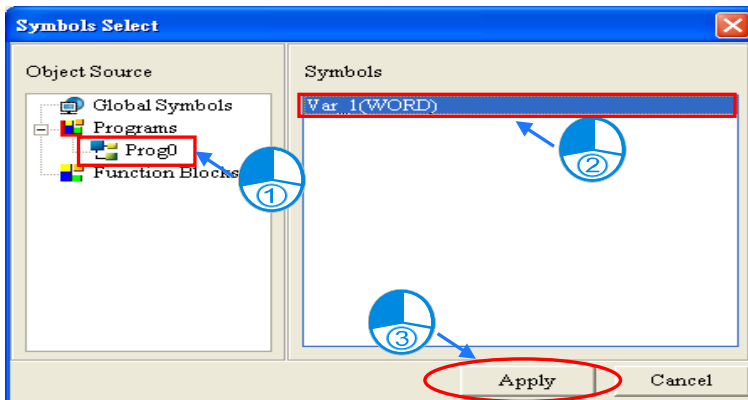
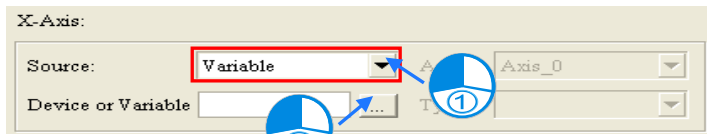
Type a name for the new curve and define the sources and other related settings for the X-, Y-, Z- axes. Setting the source to Position or Speed for the axes means the data of the 3D chart is from the position or speed of the axes. Users can use the drop-down list to select the created axes. When the PLC runs the motion control program, the position or speed of the set axis can be monitored in real time.



Setting the source to Device for the axes means the data of the 3D chart is from the device of the axes. Users can use the drop-down list to select the created device address and select the type. When the PLC runs the program, the address and type of the device of the set axis can be monitored.



Setting the source to Variable for the axes means the data of the 3D chart is from the variable of the axes. Users can use the ... to open a Symbol Select page and select the created variable from the Symbols section. The system will bring in the suitable type automatically.




18




Setting the source to Time for the axes means the data of the 3D chart is from the time of the axes and the value of axes will be increased by 1 every second.

X-Axis:

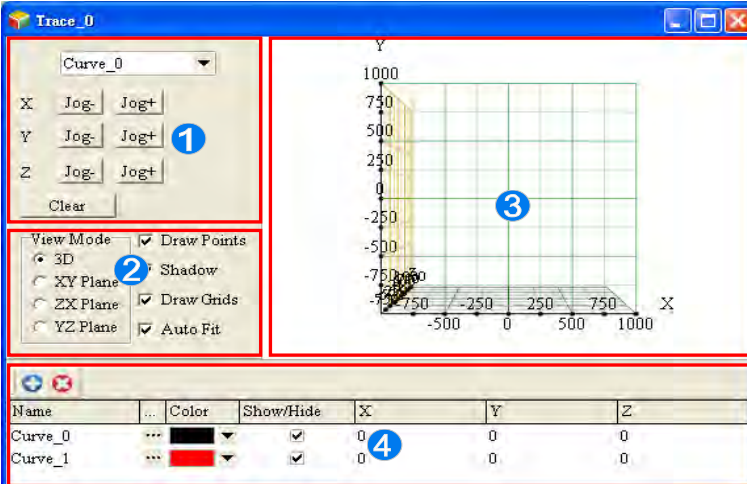
Source:	Time	Axis:	
Device or Variable:		Type:	



Set up the width and the color for the curve and then click **OK** on Curve Setting page to complete the creation of a curve. After that the created curve will be listed below the 3D chart. Click to select a curve you'd like to edit. Click  to delete the selected or click ... to open the curve set up page and set up again. Use the drop-down list to change the color of the selected curve and select or deselect the checkbox to show or hide the curve on the 3D chart. Up to 10 curves can be added on one 3D chart.

Name	Color	Show/Hide	X	Y	Z
Curve_0		<input checked="" type="checkbox"/>	0	0	0

### 18.5.4 Display a 3D Chart

Follow the procedures aforementioned to download program and complete the monitoring setups. After a 3D chart is created, make sure the motion control CPU is connected to your computer and then the 3D chart will be displayed.



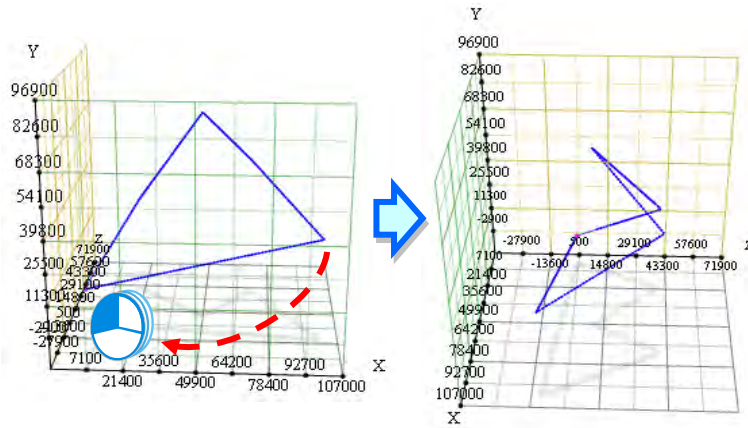
Name	Color	Show/Hide	X	Y	Z
Curve_0		<input checked="" type="checkbox"/>	0	0	0
Curve_1		<input checked="" type="checkbox"/>	0	0	0

- 1 To edit a created curve, users can use the drop-down list on the area 1 or click to select a curve from the list 4. After selection, click **Jog-** and **Jog+** to edit the curve or click **Clear** to delete the selected curve.
- 2 Use options in the View Mode section to change the view of the chart; selections are 3D, XY Plane, ZX Plane and YZ Plane. Other view options include to show or not to show the Draw

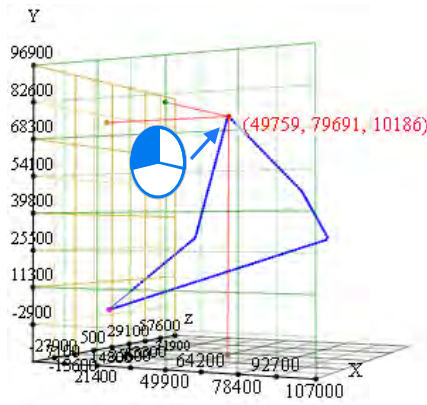
Points, Shadow, Draw Grids and Auto Fit. When Auto Fit is selected, the display area will auto adjust the X, Y, Z axes according to the traces to fit the screen.

- 3 3D chart display area.
- 4 List of the curves created.

Users can use click and drag the trace on the 3D chart to see the chart from different angles. Use the mouse wheel to scroll and adjust the scale of the chart.



Move the cursor pointer to the trace on the 3D chart and it will show its coordinates, as the image shown below.



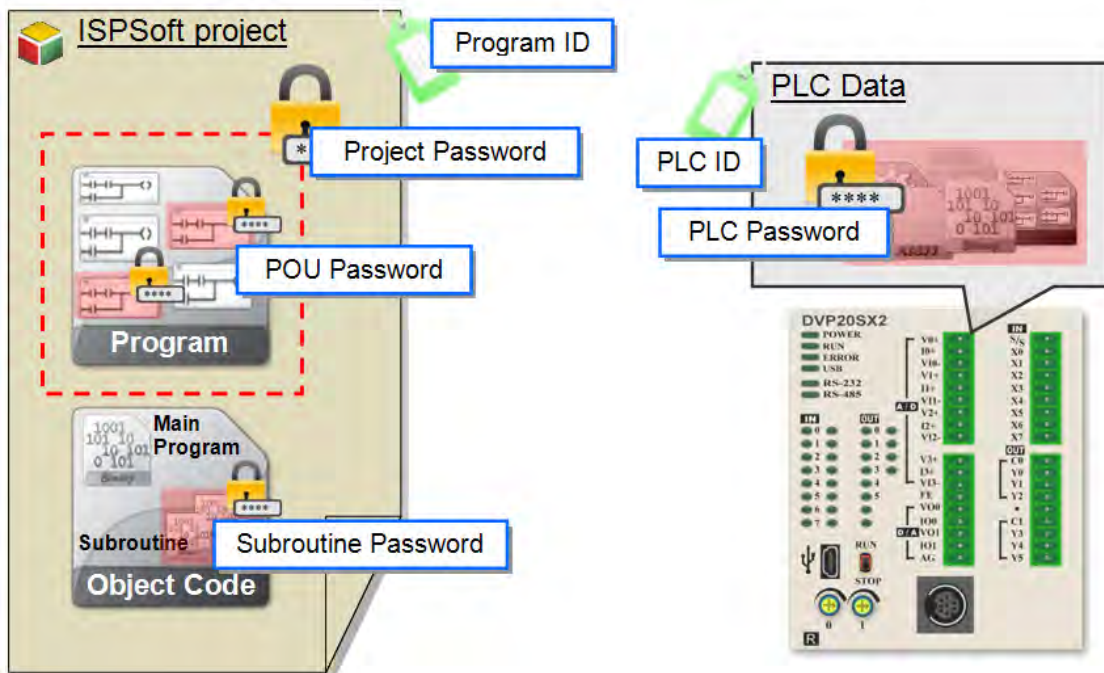
# Chapter 19 Password Management and Data Protection

## Table of Contents

<b>19.1</b>	<b>Password Protection Mechanisms Provided by ISPSoft .....</b>	<b>19-2</b>
<b>19.2</b>	<b>Program ID and PLC ID .....</b>	<b>19-3</b>
19.2.1	Setting and Unlocking a Program ID.....	19-3
19.2.2	Setting and Unlocking a PLC ID .....	19-4
<b>19.3</b>	<b>Project Password and PLC Password.....</b>	<b>19-5</b>
19.3.1	Setting and Unlocking a Project Password.....	19-6
19.3.2	Setting and Unlocking a PLC Password .....	19-8
19.3.3	Synchronize Project and PLC Password .....	19-9
<b>19.4</b>	<b>POU Password.....</b>	<b>19-10</b>
<b>19.5</b>	<b>Subroutine Password .....</b>	<b>19-11</b>
19.5.1	Introduction of a Subroutine Passwords.....	19-11
19.5.2	Setting and Unlocking a Subroutine Password .....	19-12
<b>19.6</b>	<b>Other Password and Data Protection Functions .....</b>	<b>19-13</b>
19.6.1	Disable Program Upload .....	19-13
19.6.2	Setting Read-only Area .....	19-14
19.6.3	Setting TC-01 Password Key .....	19-14

## 19.1 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.



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 POU's 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.

Password type		Description
	Settings Password	The password is used to protect all design data. Different PLC types provides design tools and password protection, including E-CAM and Position Planning Table password introduced in each chapter.
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.

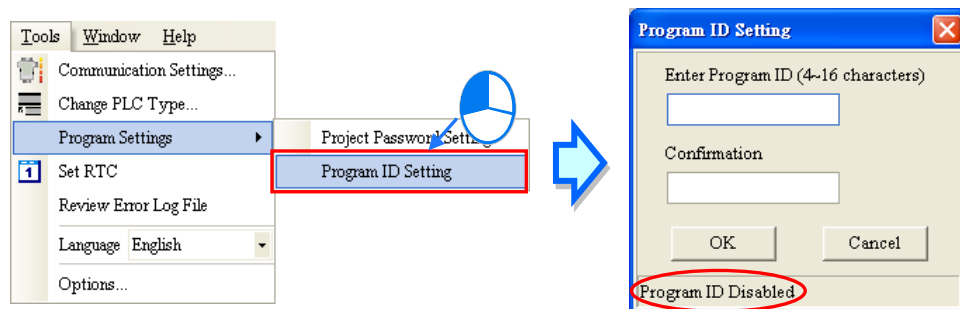
## 19.2 Program ID and PLC ID

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.

### 19.2.1 Setting and Unlocking a Program ID

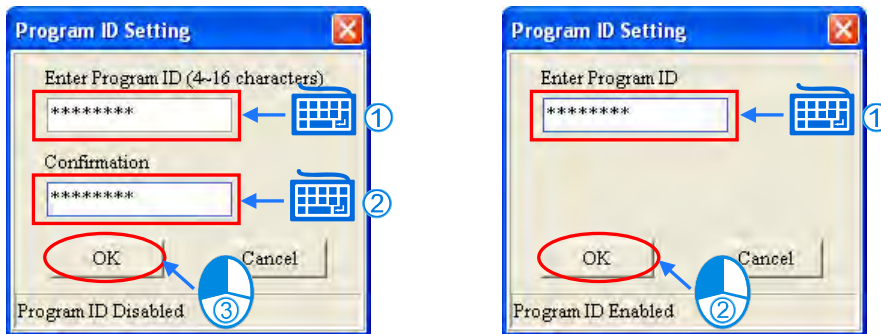
After users click the **Tools** menu, point to **Program Settings**, and click **Program ID Setting**, the **Program ID Setting** window will appear. The message at the bottom of the window indicates whether the project is protected by a program ID.



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**.

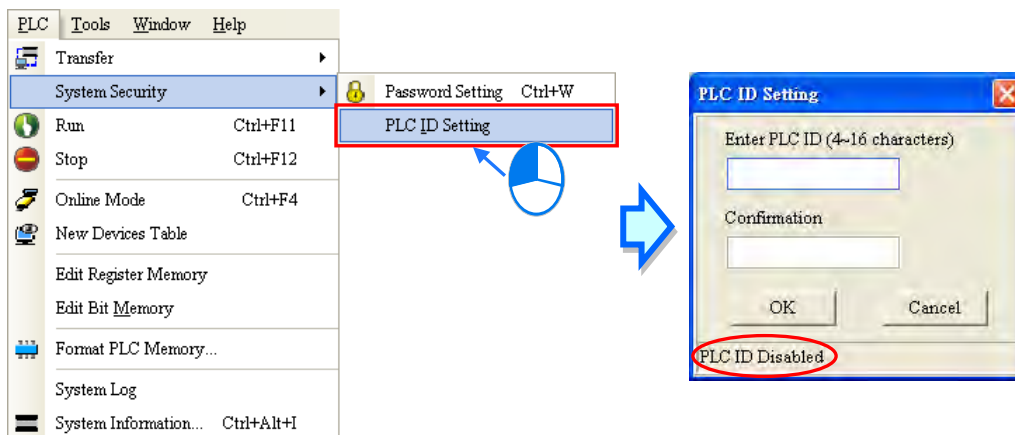
If the users want to set a program ID, they can only type English letters, numerical digits, and spaces in the window. A program ID is case-sensitive. Besides, the limit on the number of characters which can be typed depends on the model selected. The users have to type a program ID according to the 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**.



### 19.2.2 Setting and Unlocking a PLC ID

To set a PLC ID, please make sure that ISPSOft and PLC host is connected.

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.

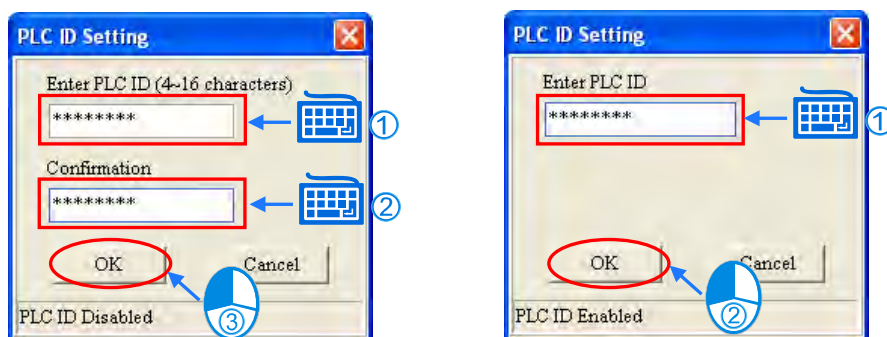


19

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**.



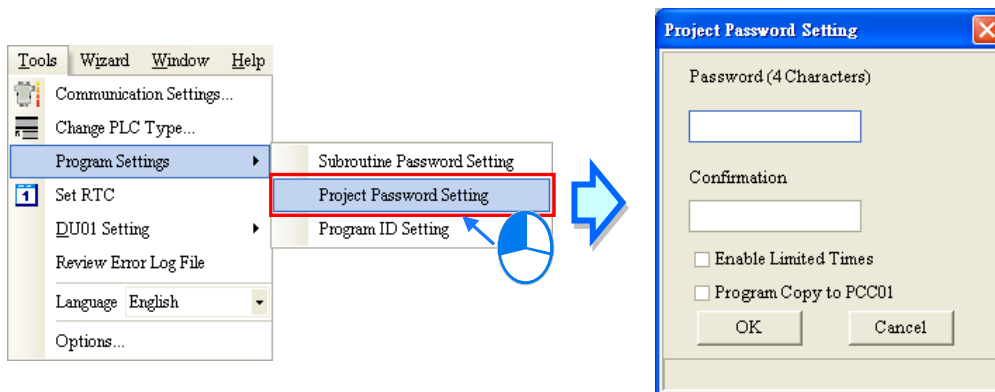
### 19.3 Project Password and PLC Password

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.

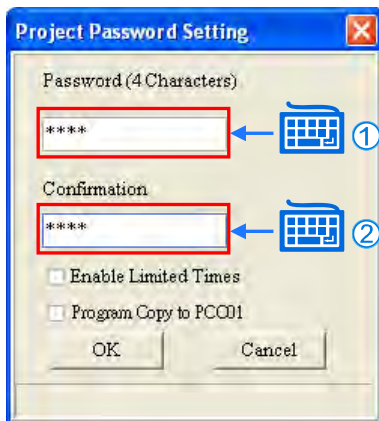
The PLC password set in a PLC is used to protect the data in the PLC. If users want to download or upload a project, the system will ask the users to type the correct password. Besides, the users can limit the total number of password guesses that can be made. If the number of password guesses that is 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.

### 19.3.1 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.



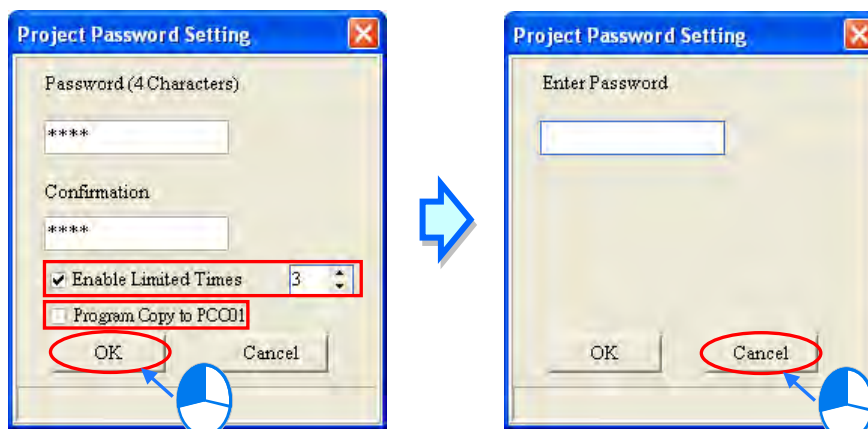
If the project is not protected by a project password, there will be two boxes used to set a project password in the **Project Password Setting** window. If the users want to set a project password, they have to type the same password in the two boxes, and click **OK**. The users can only type English letters, numerical digits, and spaces in the window. A project password is case-sensitive. 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.



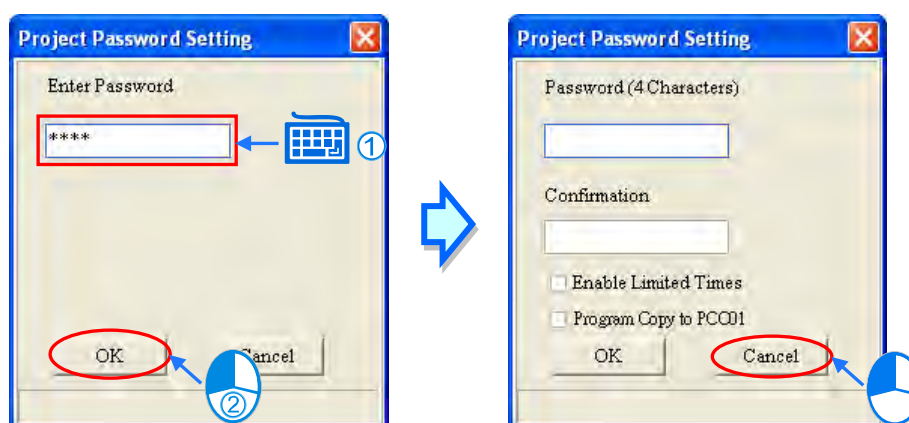
If the users want to limit the total number of password guesses that can be made, they can select the **Enable Limited Times** checkbox, and select a number in the box at the right side of the checkbox. If users want to copy the program into DVPPCC01, they can select the **Program Copy to PCC01** checkbox. (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 be protected by the project password. The window will be closed after the users click **Cancel**.



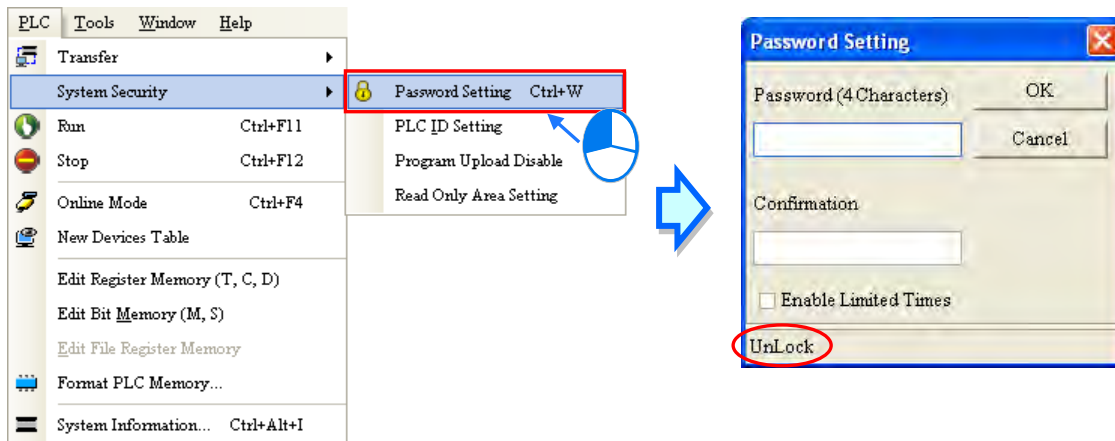
### **Additional remark**

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 onto 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.

### 19.3.2 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.

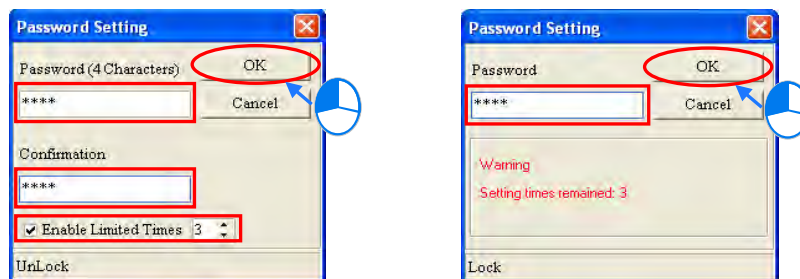
After users click the **PLC** menu, point to **System Security**, and click **Password Setting**, ISPSOft will connect to the PLC, and read the information from the PLC. If the reading of the information from the PLC is successful, ISPSOft will open the **Password Setting** window. The message at the 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 PLC password is case-sensitive. 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**.

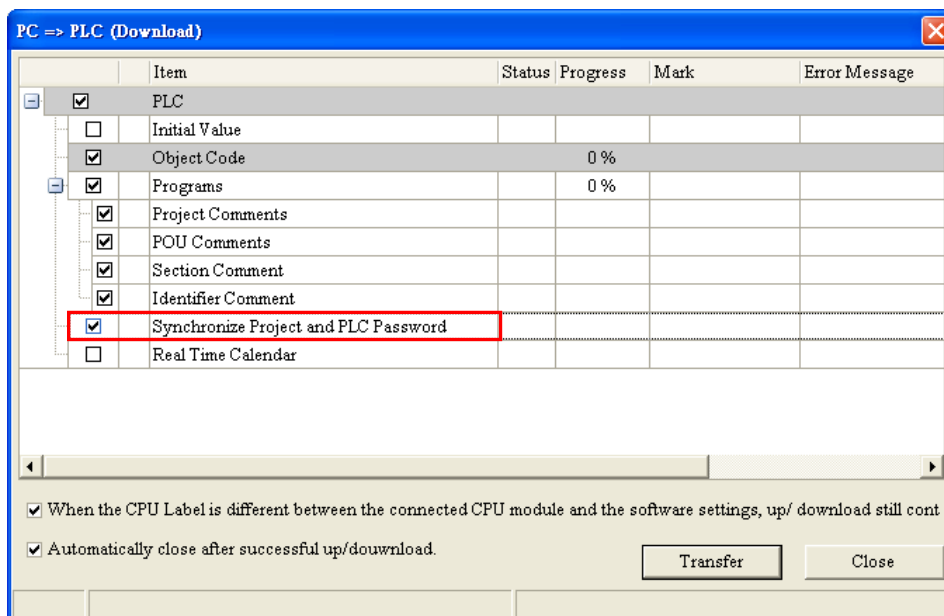


### 19.3.3 Synchronize Project and PLC Password

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.

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.

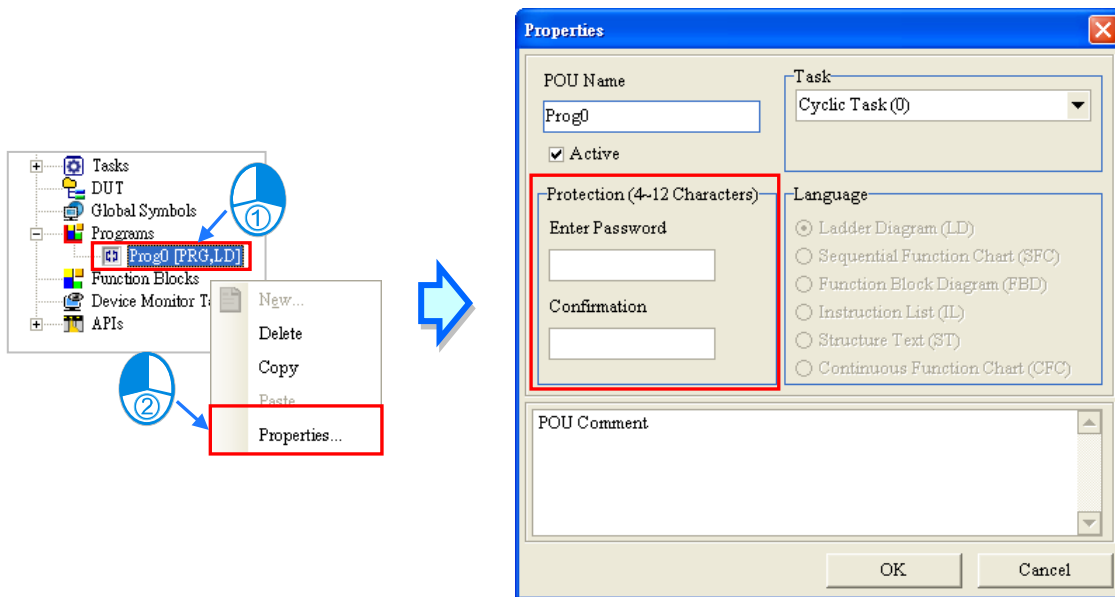
If the users do not set a project password, the system will ask the users to set a project password after the users select the **Synchronize Project and PLC Password** checkbox in the **Transfer Setup** window. When users open a project like an old file and the file is protected by project password, the **Synchronize Project and PLC Password** option is forced to select.



## 19.4 POU Password

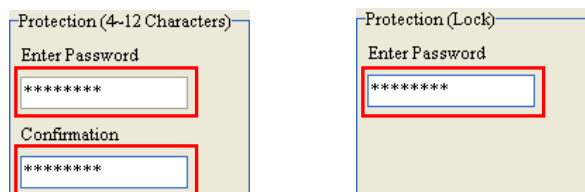
ISPSOft allows users to set a password in a POU. If users want to open a POU protected by a password, the system will ask the users to type the correct password. The password set in a POU protects the source code or the core technique in the POU. The setting of a POU password and the unlocking of a POU password are described below.

Right-click a POU in the project management area 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 **OK**.

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**.

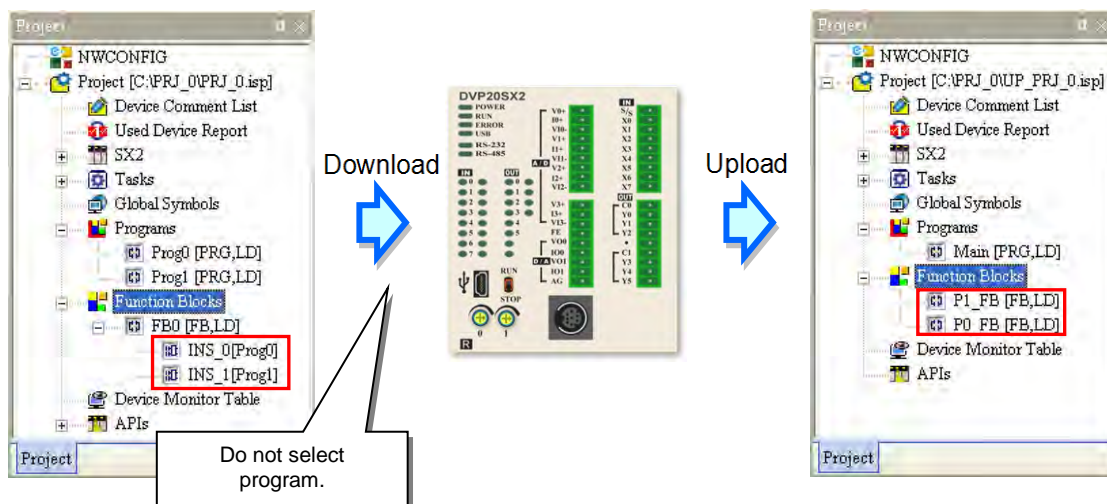


## 19.5 Subroutine Password

\* Currently, DVPxxMC series does not support subroutine password .

### 19.5.1 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 POU's 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 POU's 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 POU's transformed from the object code are not protected by POU passwords. Users can open the POU's of the function block type at will, and analyze the contents of the POU's.

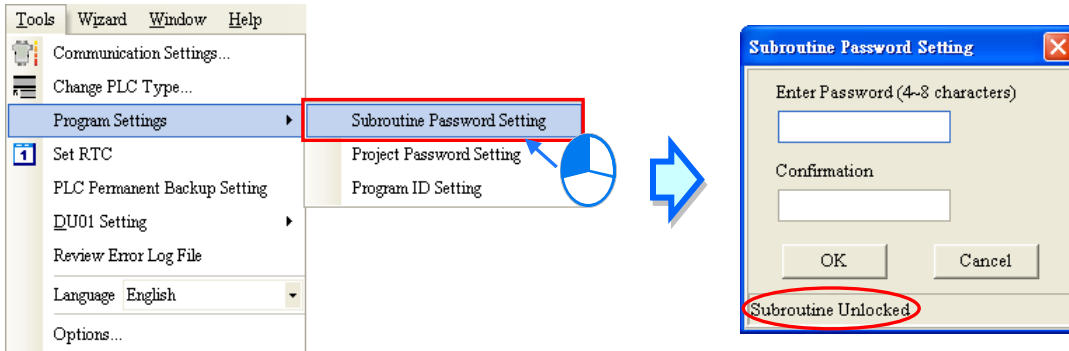


Generally speaking, developers pack core techniques or low-level programs in function blocks. If object code is gotten in the way described above, all the techniques may be analyzed. To prevent this situation, the users can set a subroutine password before the project is downloaded. If the users set a subroutine password, the subroutine password will become the POU passwords which protect the POU's of the function block type transformed from the object code which is uploaded. As a result, if the users want to open the POU's of the function block type, the system will ask the users to type the correct passwords.



## 19.5.2 Setting and Unlocking a Subroutine Password

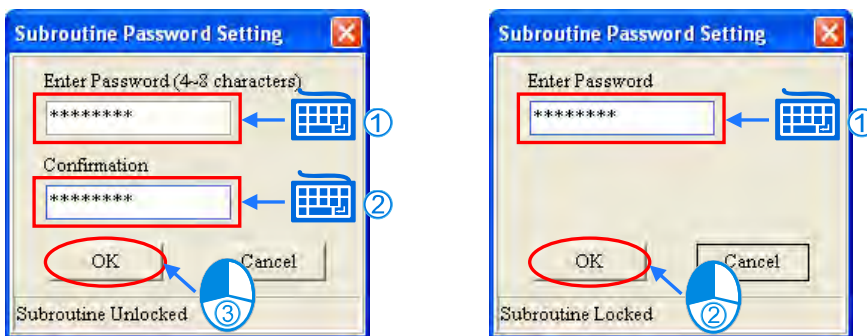
After users click the **Tools** menu, point to **Program Settings**, and click **Subroutine Password Setting**, the **Subroutine Password Setting** window will appear. The message at the bottom of the window indicates whether the subroutine is protected by a subroutine password.



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 subroutine 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**.

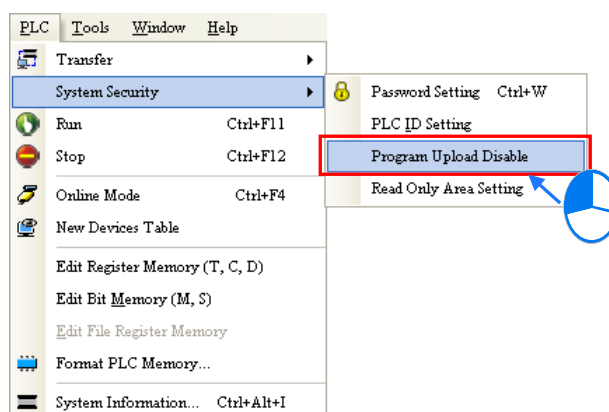


## 19.6 Other Password and Data Protection Functions

### 19.6.1 Disable Program Upload

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 does 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.

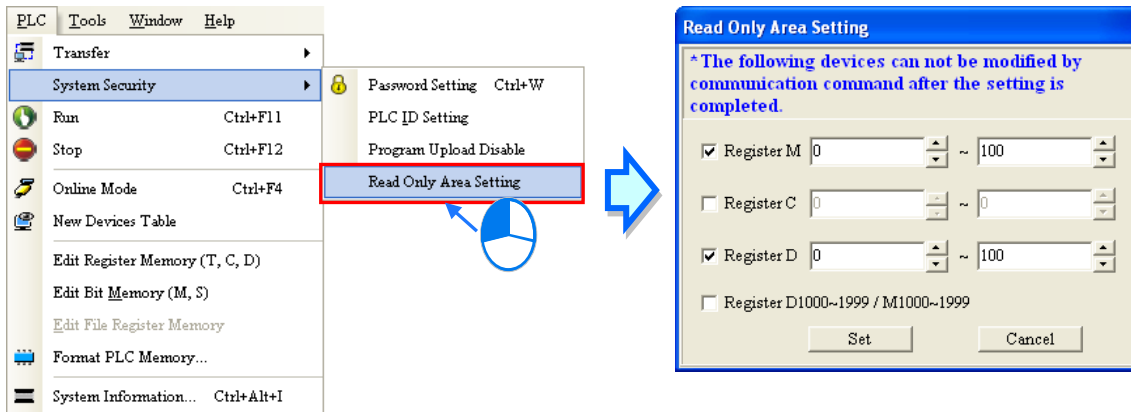


\*. Before the function is used, users have to check whether the model selected and the firmware version of the model support this function. Please refer to operation manuals or technical documents for more information.

### 19.6.2 Setting Read-only Area

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.

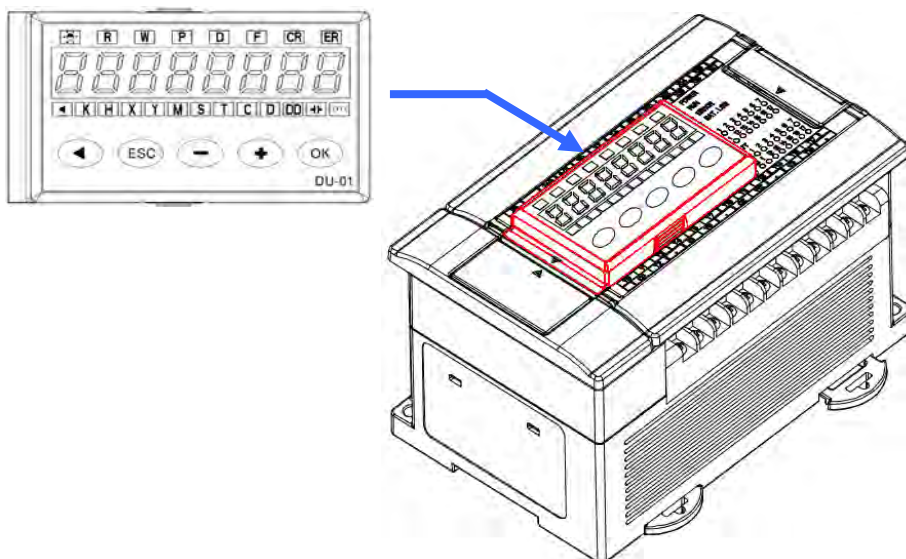
If the users want to set read-only device ranges in the DVP series PLC, they can click the **PLC** menu, point to **System Security**, click **Read Only Area Setting**. In the **Read Only Area Setting** window, the users have to select device types, and set read-only device ranges. After the setting is complete, the users can click **Set** 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 supports this function. Please refer to operation manuals or technical documents for more information.

### 19.6.3 Setting TC-01 Password Key

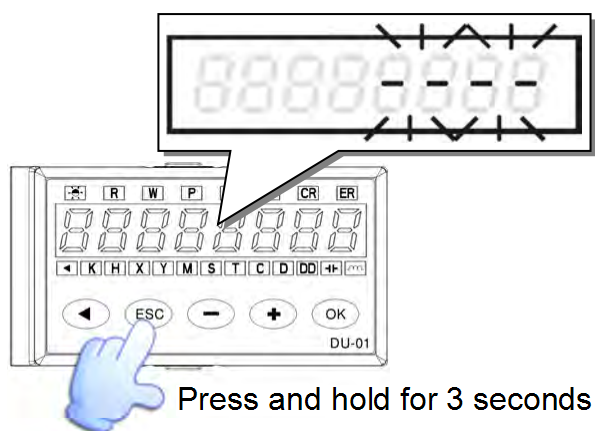
Users can set a TC-01 password key in DVPDU-01, a digital setup display. A drawing of DVPDU-01 is shown below. DVPDU-01 provides a simple operation interface for a DVP series PLC.





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. 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.

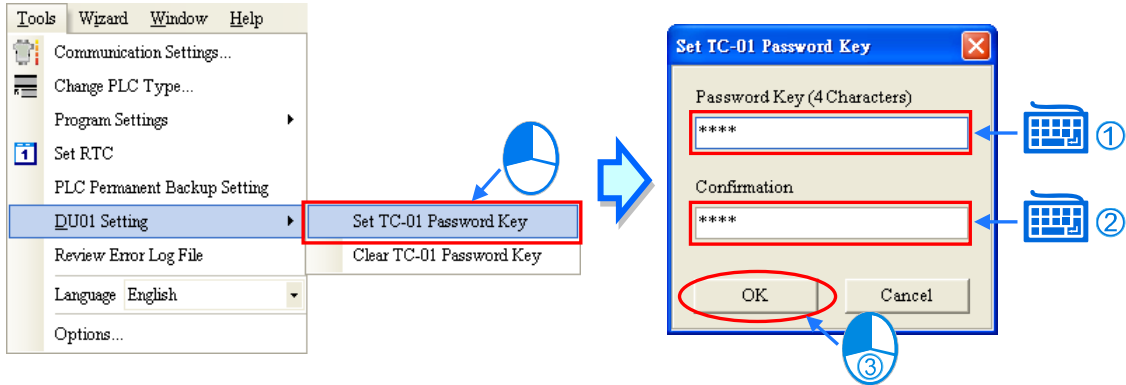
- (1) After the users press **ESC** on DVPDU-01 for three seconds, the password setting screen will appear.



- (2) Set a TC-01 password key by means of ISPSOft, or unlock the TC-01 password key by means of ISPSOft.

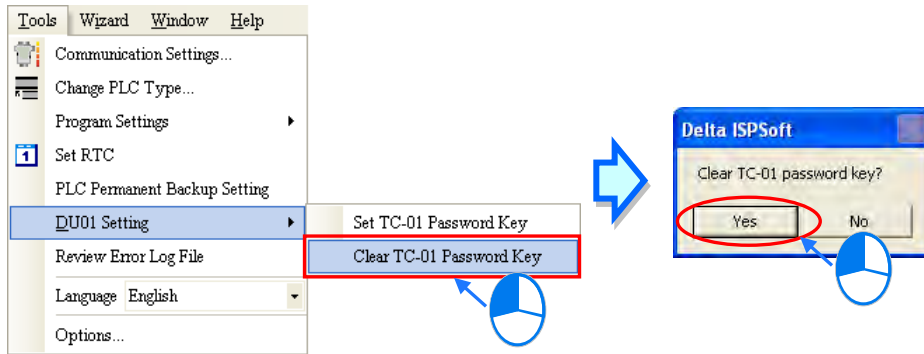
- **Setting a TC-01 password key**

After the 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.

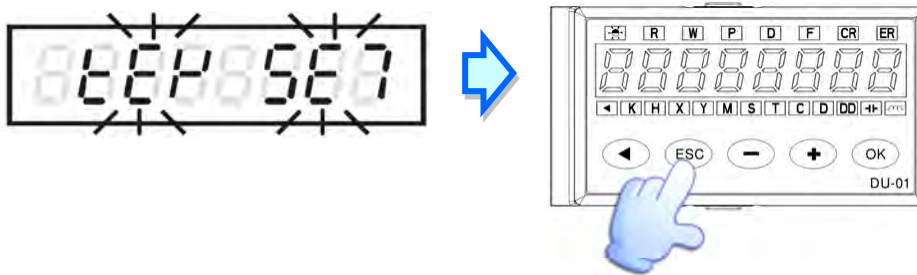


- **Unlocking the TC-01 password key**

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 20 Network Configuration and Data Exchange

## Table of Contents

<b>20.1</b>	<b>Network Configuration Tool - NWCONFIG .....</b>	<b>20-3</b>
20.1.1	Introduction of NWCONFIG .....	20-3
20.1.2	Basic Knowledge.....	20-4
20.1.3	Communication Setting in NWCONFIG.....	20-6
20.1.3.1	Connection Mechanism in NWCONFIG.....	20-6
20.1.3.2	Setting Communication Parameters.....	20-8
20.1.4	Workflow .....	20-9
<b>20.2</b>	<b>Creating a Network Architecture .....</b>	<b>20-14</b>
20.2.1	Deploying Nodes.....	20-14
20.2.2	Connecting to a Network .....	20-18
20.2.3	Adjusting or Deleting Devices or Networks.....	20-23
20.2.4	Setting the Attributes of a Node/Network.....	20-26
20.2.5	Hiding/Displaying Devices or Networks.....	20-31
20.2.6	Correct Network Architecture .....	20-34
20.2.7	Downloading Routing Tables.....	20-37
20.2.8	Testing Routing .....	20-39
<b>20.3</b>	<b>Constructing a PLC Link.....</b>	<b>20-41</b>
20.3.1	Opening the PLC Link Table Editor Window.....	20-41
20.3.2	Select Master Station Device (Step 1) .....	20-43
20.3.3	Communication Parameter Settings (Step 2) .....	20-44
20.3.4	Create Data Exchange Table (Step 3) .....	20-46
20.3.4.1	Introduction of a Data Exchange Table.....	20-46
20.3.4.2	Setting a Data Exchange Group.....	20-48
20.3.4.3	Device Synchronization in Data Exchange Table.....	20-50
20.3.4.4	Managing Data Exchange Table.....	20-52
20.3.5	Monitoring a PLC Link.....	20-53
20.3.6	Notifications on PLC Link.....	20-58
<b>20.4</b>	<b>Constructing an Ether Link .....</b>	<b>20-59</b>
20.4.1	Introduction of an Ether Link.....	20-59
20.4.2	Open Ether Link Configuration.....	20-60
20.4.3	Create and Manage Data Exchange Table.....	20-62
20.4.4	Node List and Display Area .....	20-66
20.4.5	Start Mode of an Ether Link .....	20-68
20.4.6	Download Ether Link Configuration .....	20-70

20.4.7	Upload Ether Link Configuration .....	20-72
20.4.8	Deleting Asynchronous Device .....	20-74
20.4.9	Enable/Disable Online Monitoring Function.....	20-76
20.4.9.1	Enabling a Monitoring Function.....	20-77
20.4.9.2	Monitoring Status .....	20-79
20.4.9.3	Disabling a Monitoring Function.....	20-80
20.4.10	Online Start/Stop Ether Link (SM Flag).....	20-81
20.4.10.1	Starting the Execution of an Ether Link.....	20-81
20.4.10.2	Stopping the Execution of an Ether Link .....	20-83
20.4.11	Monitoring Table and Error Log .....	20-85
<b>20.5</b>	<b>NWCONFIG Management and Application .....</b>	<b>20-86</b>
20.5.1	Save and Print .....	20-86
20.5.2	Downloading.....	20-87
20.5.2.1	Downloading Parameters .....	20-87
20.5.2.2	Description of Downloading.....	20-87
20.5.3	ISPSOft Routing Application .....	20-89

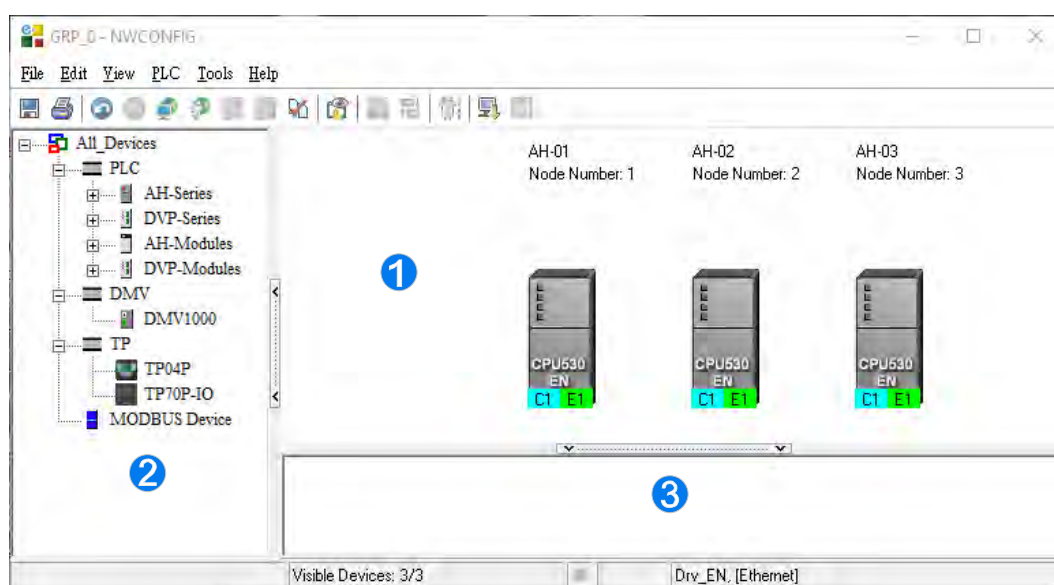
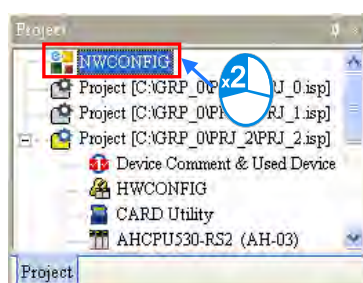
## 20.1 Network Configuration Tool - NWCONFIG

### 20.1.1 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 NWCONFIG. 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

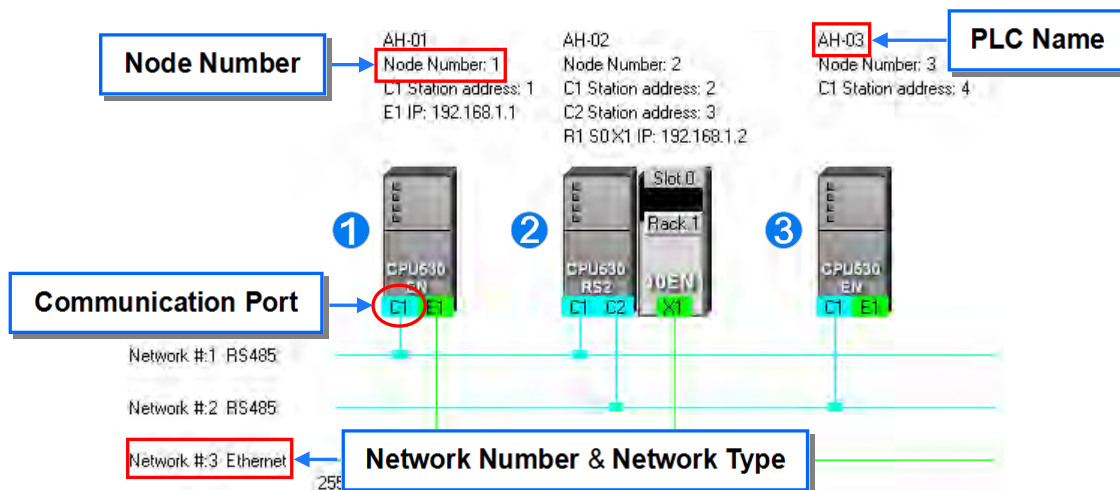
NWCONFIG is used to create a network framework for projects, and therefore it is at the top of the project management area. If users want to start NWCONFIG, they can double-click **NWCONFIG** in the project management area.



- ① **Working area:** It is a main working area. Users can create a network framework in this area.
- ② **Device list:** All the devices which can be used are listed in a catalog.
- ③ **Message display area:** The messages related to operation are displayed in this area.

### 20.1.2 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 20.2.2 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 AH5x0 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 AH5x0 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. ① ~ ③ in the figure above are nodes. ② consists of a CPU module and a network module. The network module cannot operate by itself, and therefore the CPU module and the network module are regarded as one node.

Besides, AH5x0 series CPU modules can forward packets and perform routing. For example, ③ in the figure above can be monitored through ①. 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 AH5x0 series CPU modules can be assigned node numbers, and the node number of a node in a network cannot 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**

This is to identify RS-485 stations on the internet. The station address is configured base on communication port. Basically, a communication port represents a station, and the workstation address that has the same internet no. cannot be repeated. Therefore, when a node contains several COM ports and is connected, a station address has to be assigned.

- **IP address and DHCP mode**

IP address is the address for communication ports on the internet. Basically, a communication port represents a internet point. Therefore, when several communication ports are included and connected to the internet, an IP address has to be assigned and cannot be repeated.

DHCP is the mechanism for dynamic allocation of IP addresses. A host allocated by the DHCP has to exist in the internet. When a communication port adopts DHCP mode, the port automatically obtains an IP address from DHCP host.

- **Subnet mask**

A subnet mask is used to differentiate the subnetwork parameter values.

- **PLC Link**

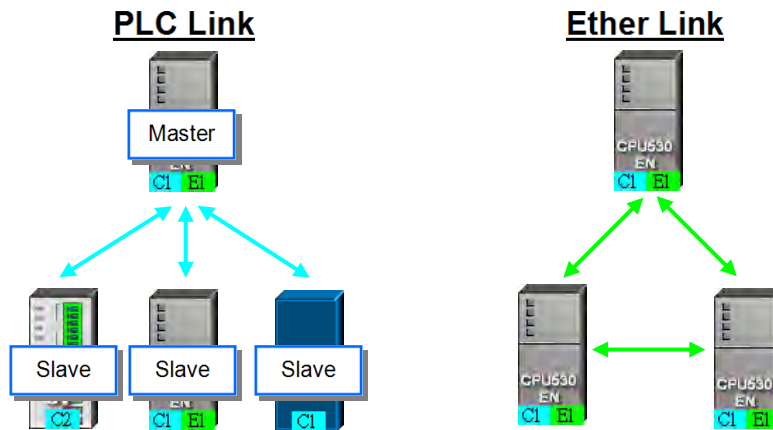
A network mechanism for data exchange performed through an RS-485 cable. If there are several nodes in an RS-485 network, users can create a mechanism for data exchange in the network. If the parameters which are set are downloaded to the PLC which functions as a master station, the system of the PLC 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 cannot exchange data. They have to exchange data through the master station.

- **Ether Link**

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.

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 cannot 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.

### 20.1.3 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.

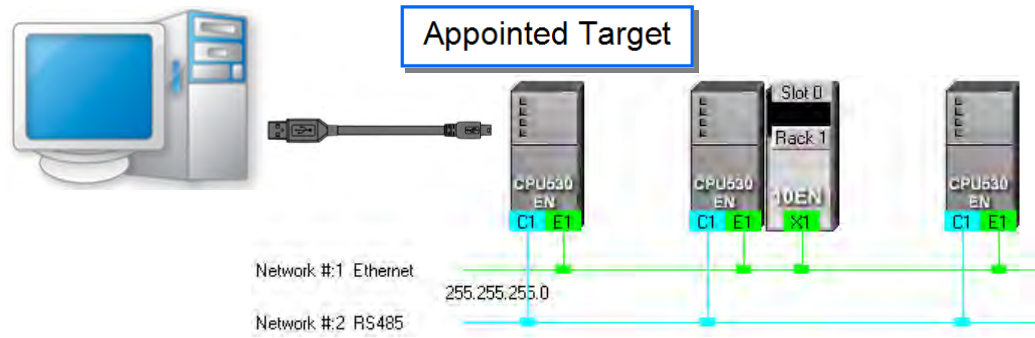
#### 20.1.3.1 Connection Mechanism in NWCONFIG

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**

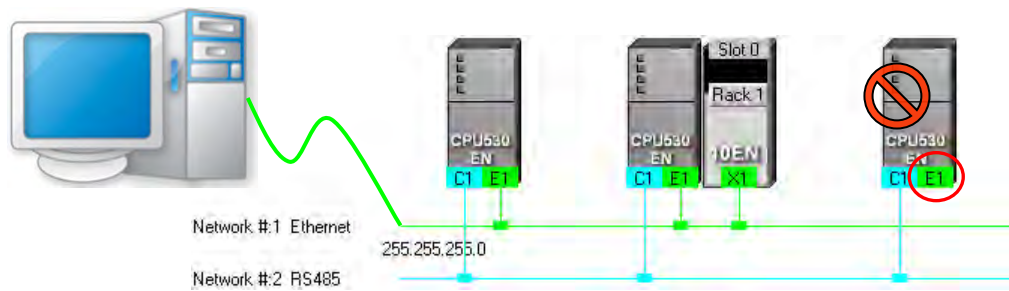
Users can download parameters to a single device, upload parameters from a single device, and monitor a single device. Before users download parameters to a single device, upload parameters from a single device, or monitor a single device, they have to make sure that the device specified is the same as the device which is actually connected to the computer.



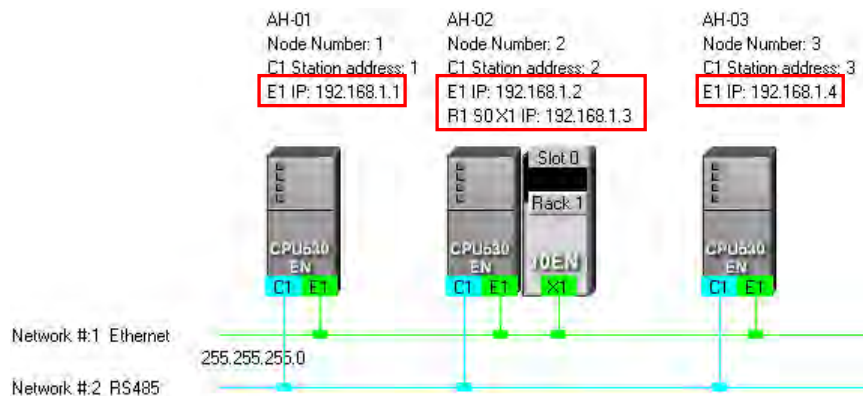


● **Multiple nodes**

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.

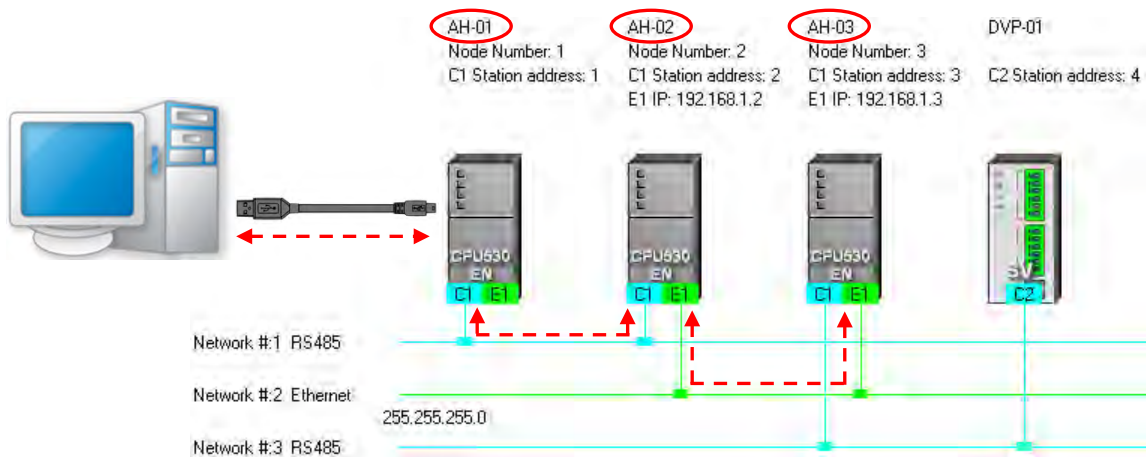


If the connection type that the driver selected uses is Ethernet, the system will carry out communication according to the IP addresses assigned to the devices in NWCONFIG. Before the communication is carried out, users have to make sure that the IP addresses actually assigned to the devices are the 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 carry out communication through routing.

Routing is a function provided by AH5x0 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.

- Users have to create networks in NWCONFIG, and download the routing tables produced to nodes in the networks. Please refer to section 20.2 for more information.
- AH5x0 series CPU modules support routing whereas DVP series PLCs and other devices do not support routing. Although DVP series PLCs and other devices cannot function as intermediate nodes through which packets pass, they can function as destinations to which packets are transmitted.

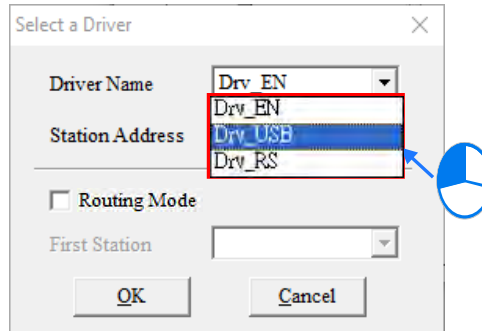
### 20.1.3.2 Setting Communication Parameters

The steps of setting the communication parameters in NWCONFIG are as follows. Please follow the steps to ensure normal connection.

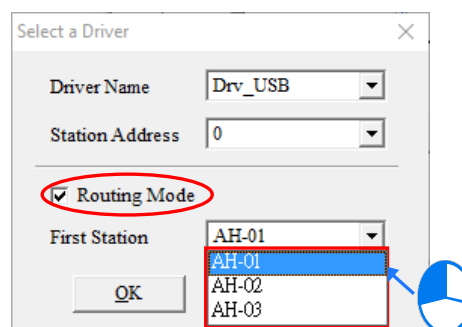
- Start the communication manager COMMGR, and check the status of the driver as OK.
- When using **Single node**, please check the object for operation is the same as the actual device for connection; When operating in **Multiple node**, please check that all operating devices are linked to the Ethernet and as planned in NWCONFIG.
- 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.



- (5) Select the **Routing Mode** and select a device in the **First Station** from the drop-down list box that is the actual device connected to the PC.

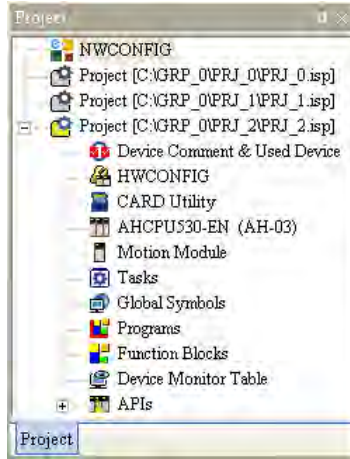


### 20.1.4 Workflow

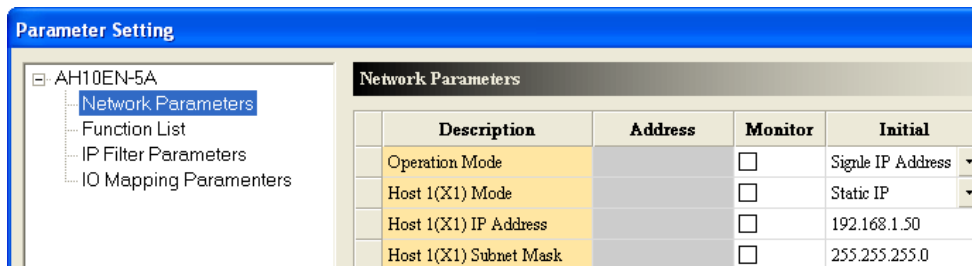
The creation of networks involves the operation of a system, and therefore the workflow needed must consist of a sequence of connected steps. The workflow needed to create networks is introduced briefly in this section, and will be described in length in the following sections. The workflow introduced 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.

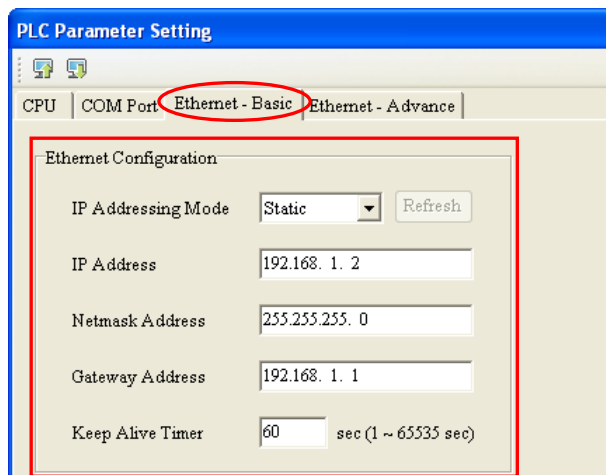
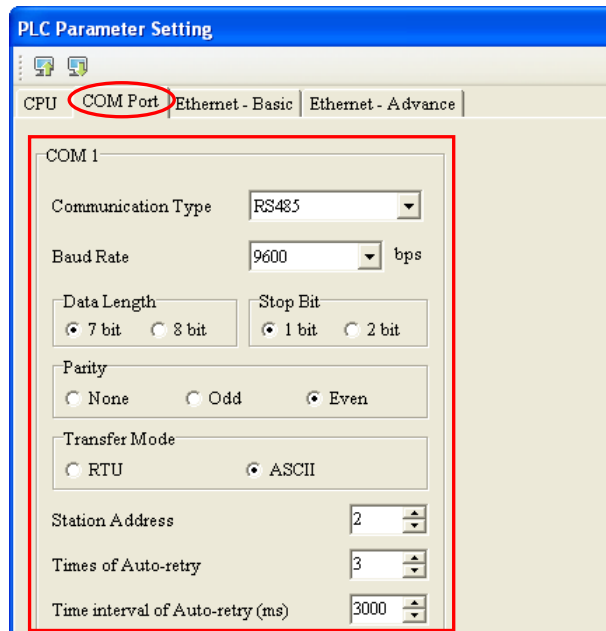
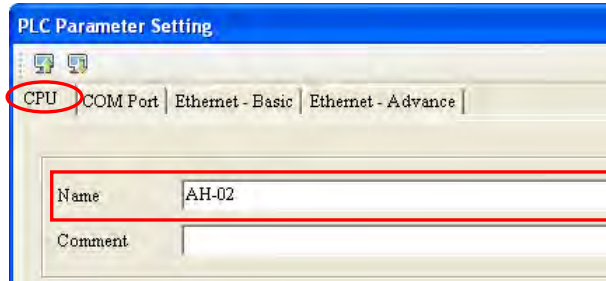
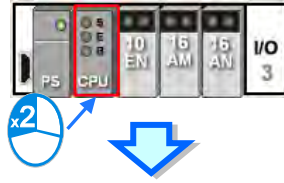
- (1) Before users create a system by means of ISPSOft, they have to design networks. The users have to decide what PLCs or devices are used in the networks, whether a PLC needs to be connected to a network module, how the nodes in the networks are connected, what IP address or RS-485 station address 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.

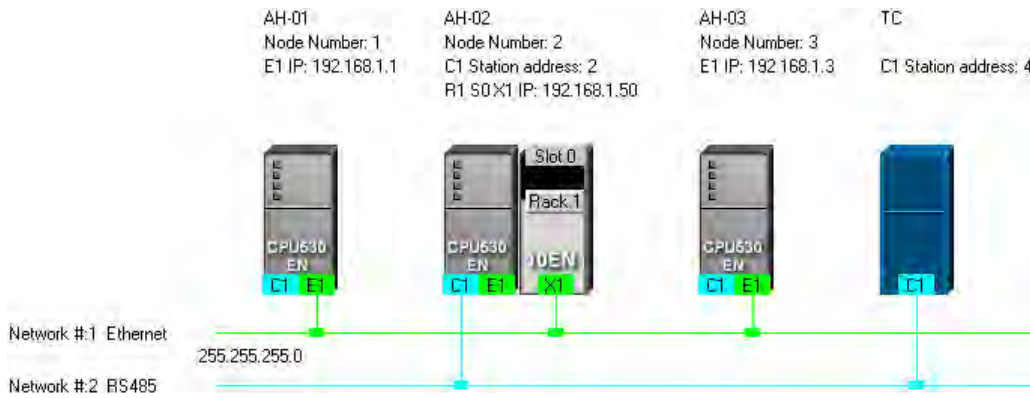


- (3) If there are projects for AH5x0 series CPU modules, the users have to open the HWCONFIG windows in the projects, and complete hardware configurations. The users have to configure modules, set the parameters in network modules, gives names to the CPU modules, set ports, and set Ethernet ports. Please refer to chapter 3 for more information.





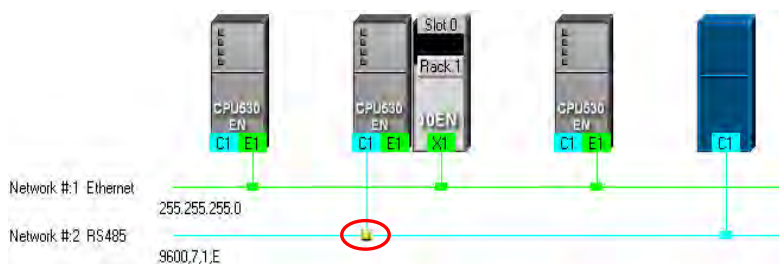
(4) Complete a network configuration in NWCONFIG.



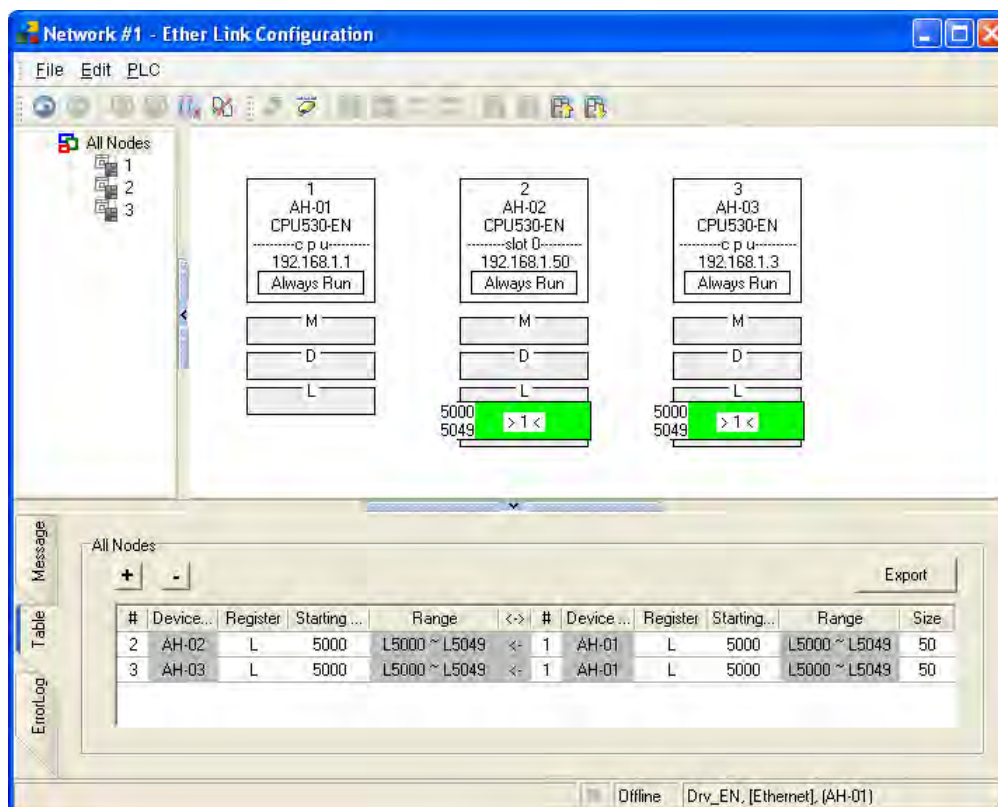
(5) Create a mechanism for data exchange performed by means of a PLC Link or an Ether Link. A PLC Link and an Ether Link operate independently. The users can create them in any order. The addresses involved in data exchange cannot overlap, otherwise an error will occur after the data exchange is performed. 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.

#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
1	4	R	D3000~D3000	<=	16#1000~16#1000	1	Enabled	MODBUS Device
		W	D3001~D3001	=>	16#1001~16#1001	1		
2	0	R	D100	<=	D4096	0	Disabled	Unknown
		W	D100	=>	D4096	0		
3	0	R	D200	<=	D4096	0	Disabled	Unknown
		W	D200	=>	D4096	0		
4	0	R	D300	<=	D4096	0	Disabled	Unknown
		W	D300	=>	D4096	0		
5	0	R	D400	<=	D4096	0	Disabled	Unknown
		W	D400	=>	D4096	0		

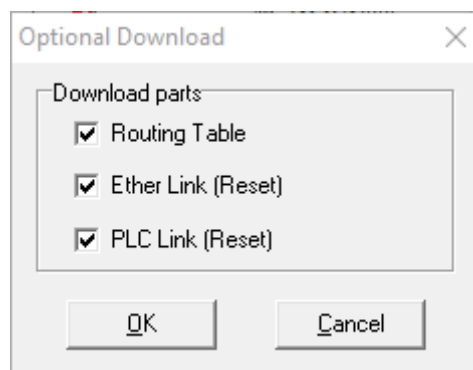


The figure below is a table related to data exchange performed by means of an Ether Link.



- (6) Download the programs in the projects, the parameters set in HWCONFIG, 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 AH5x0 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.

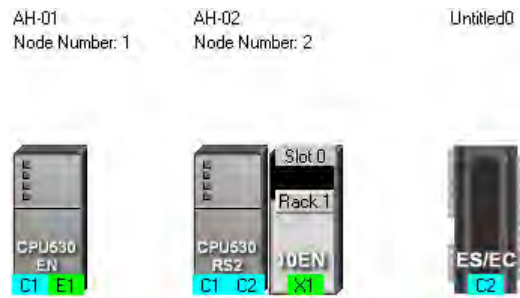
The **Optional Download** window in NWCONFIG is shown below. The items which can be downloaded vary with the node selected.



## 20.2 Creating a Network Architecture

### 20.2.1 Deploying Nodes

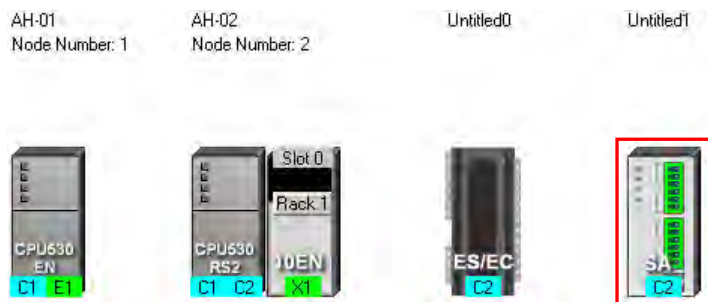
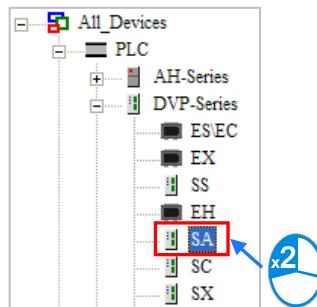
After users open the NWCONFIG window for projects for the first time, the devices for which the projects are created will be deployed in the working area in the NWCONFIG window. If the devices deployed in the working area include AH5x0 series CPU modules, the modules connected to the CPU modules, the 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 cannot 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**

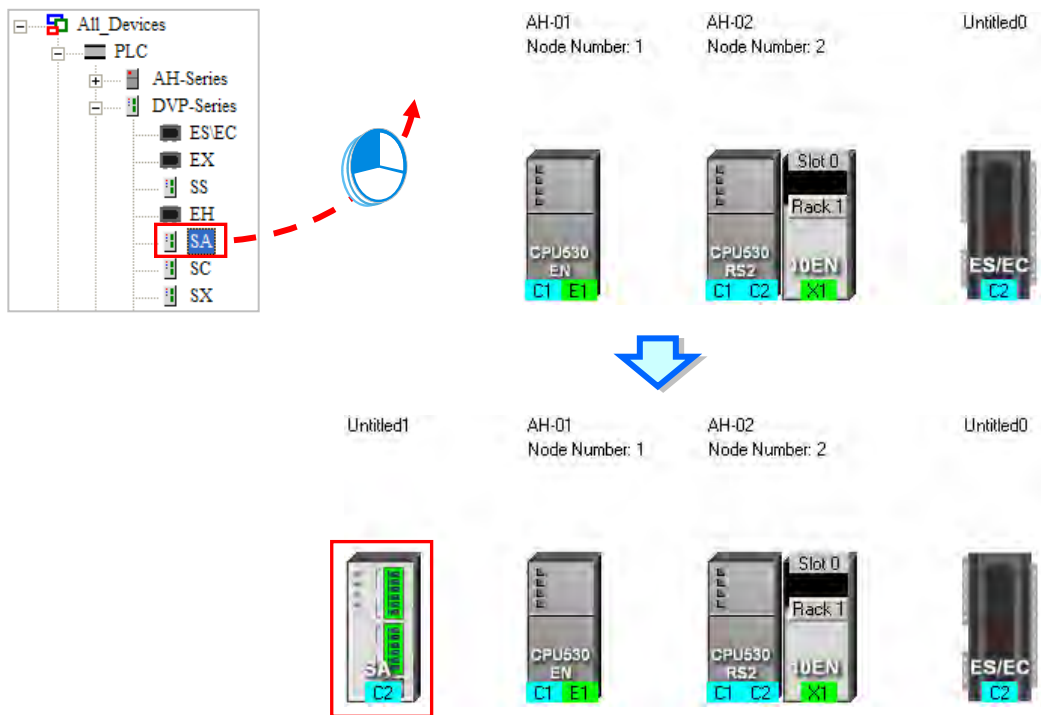
Select a PLC or a device on the device list. After the users double-click the PLC or the device, the PLC or the device will be put at the right side of the rightmost device in the working area.



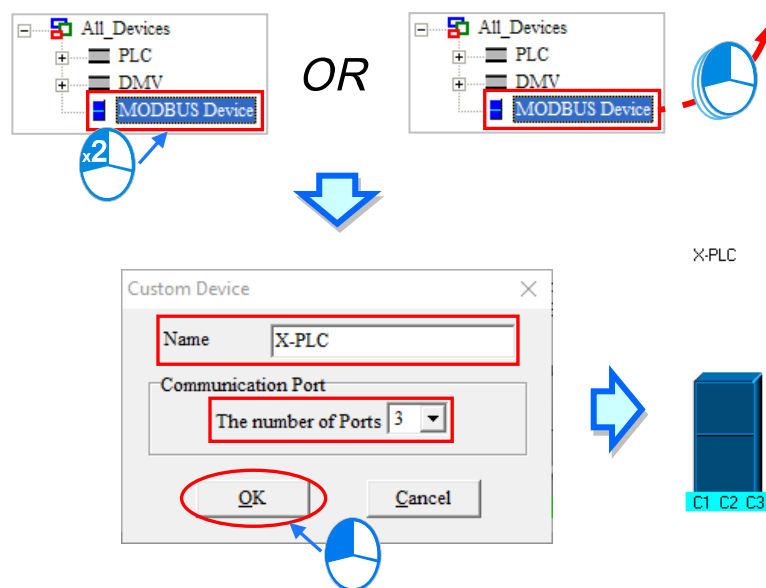


● **Method 2**

Select a PLC or a device on the device list, and then drag it to a position in the working area.



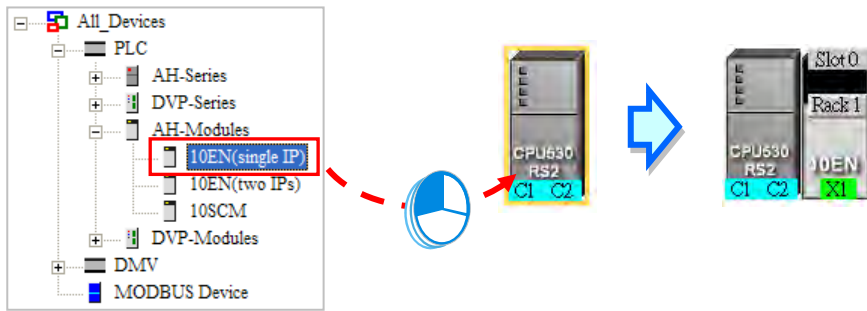
The users can add a user-defined Modbus device to the working area. After the users select **MODBUS Device** on the device list, and add it to the working area in one of the two ways described below, the **Custom Device** window will appear. The users have to type a name in the **Name** box, select a number in the **The number of ports** from drop-down list box, with maximum of three ports and click **OK**.




Base on the project content, the first AH5xo host is auto-selected with module configuration in sync with HWCONFIG. Therefore, it is not possible to add modules in NWCONFIG, but configuration changes can be made in HWCONFIG settings.

● **Method 1**

Select a network module on the device list, and then drag it to a PLC in the working area.

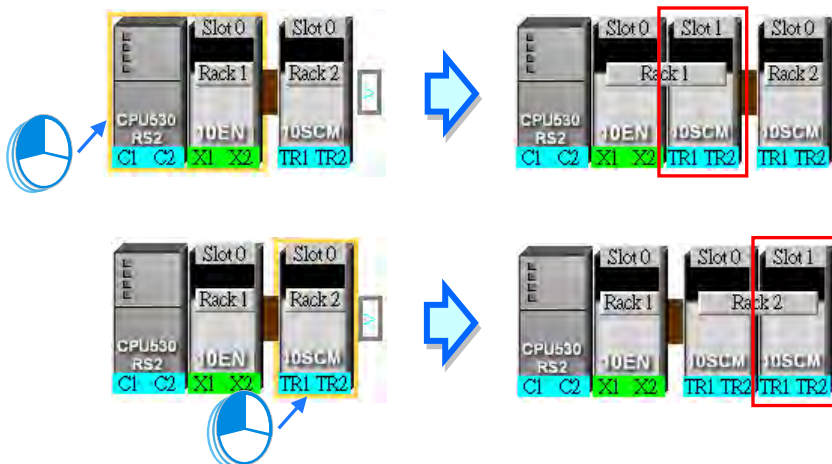



If the PLC does not support the network module selected, the mouse cursor becomes .

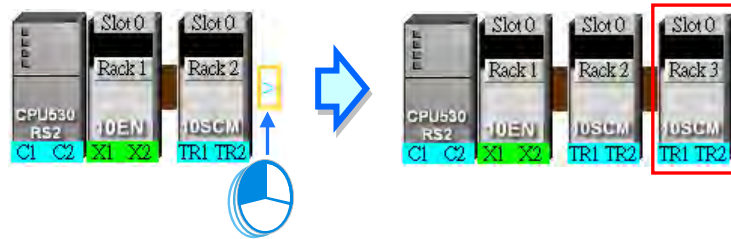


For AH5x0 series, 10EN module can be placed on the main backplane, with a maximum of at least 8 modules. The SCM module can be put on the extension backplane and the number is limited according to the slot number of each backplane. .

When choosing to drag added SCM modules, users can decide the backplane position to drag and drop.

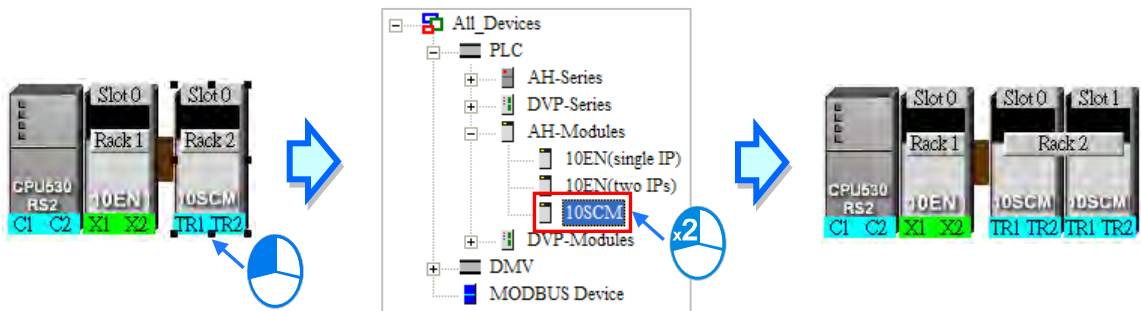


After the users drag an AH10SCM series module to  at the right side of a node, an extension rack will be added to the node.

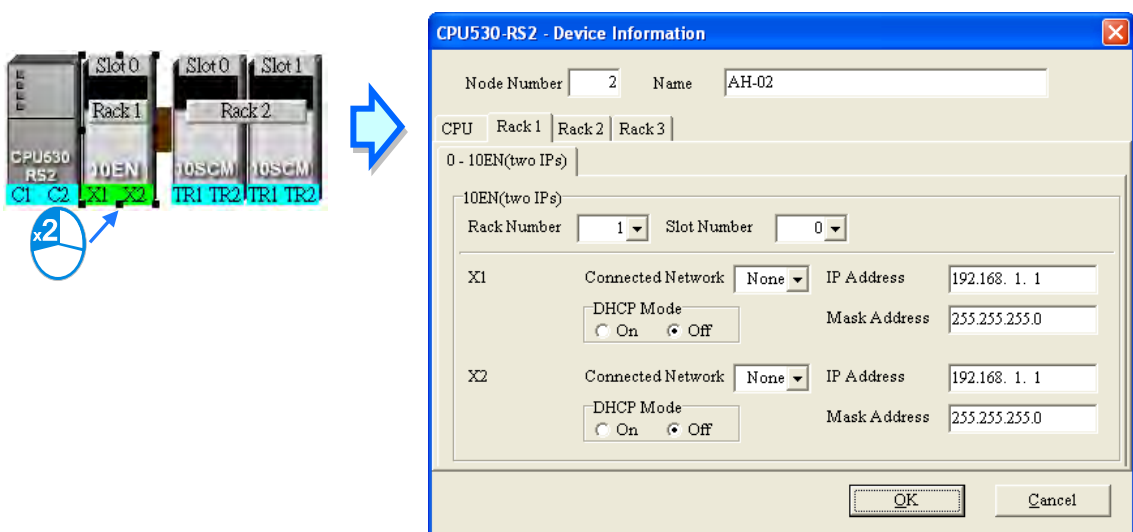


● **Method 2**

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.





## 20.2.2 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.

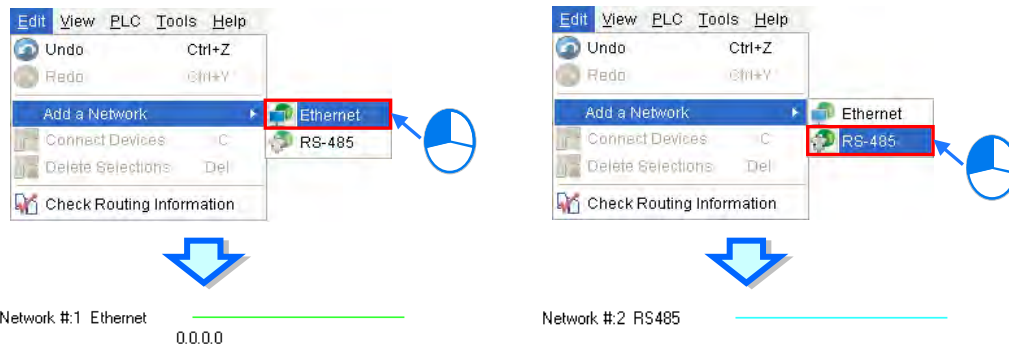
### ● Method 1

After users click  on the toolbar, an Ethernet network is added. After the users click  on the toolbar, an RS-485 network is added.



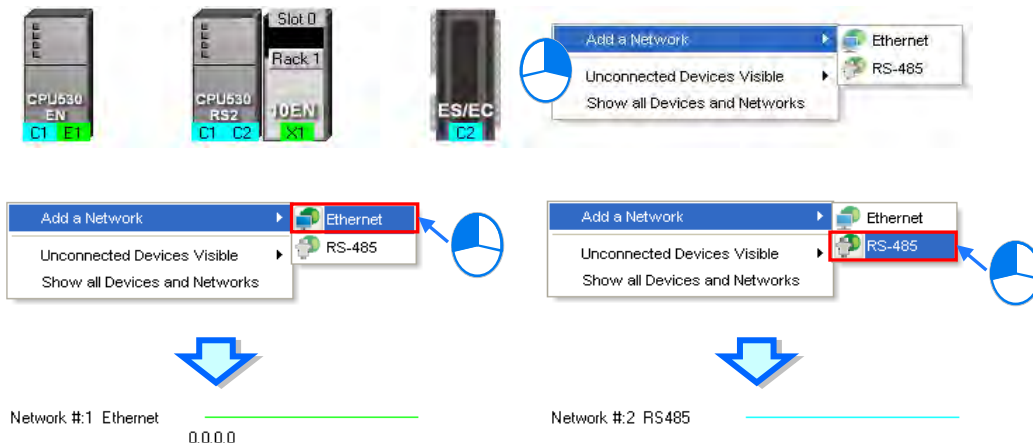
### ● Method 2

Click the **Edit** menu, point to **Add a Network**, 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**.



20

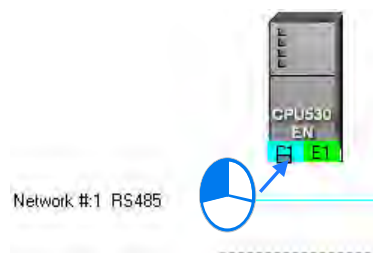
The ports of a device are displayed at the bottom of the device. If a port is blue, it is an RS-485 port. If a port is green, it is an Ethernet port. The port number assigned to a port of a device is consistent with the definition of the port. For example, E1 represents the first Ethernet port, C1 represents 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 cannot connect to any network.



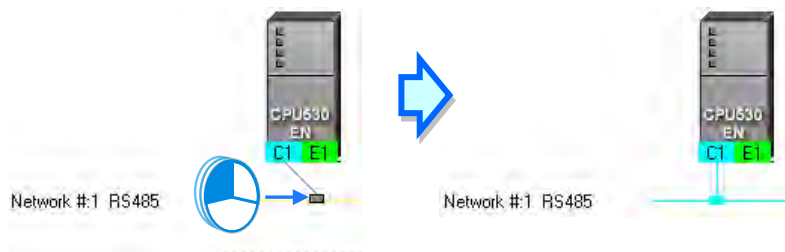
There are several ways to connect the nodes in the working area to networks.

- **Connecting a port to a network by means of dragging the port**

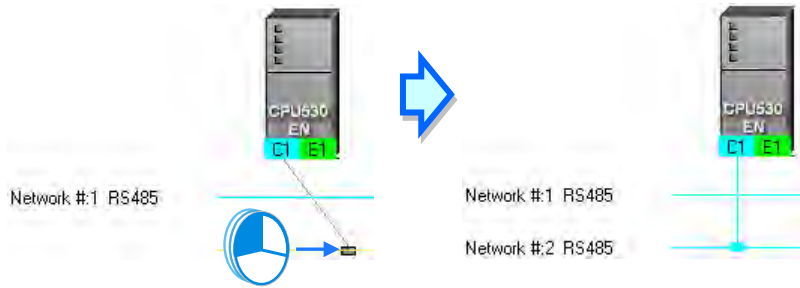
The users press the left mouse button while the mouse cursor hovers over a port. A dotted line is under the existing network.



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.

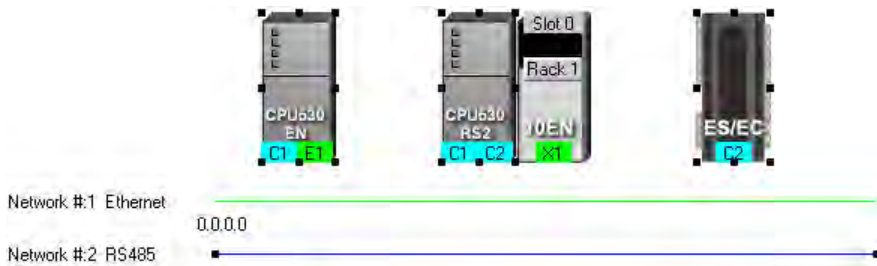



If the users move the mouse cursor to the dotted line while holding the left mouse button down, the port will connect to a network which matches the port.

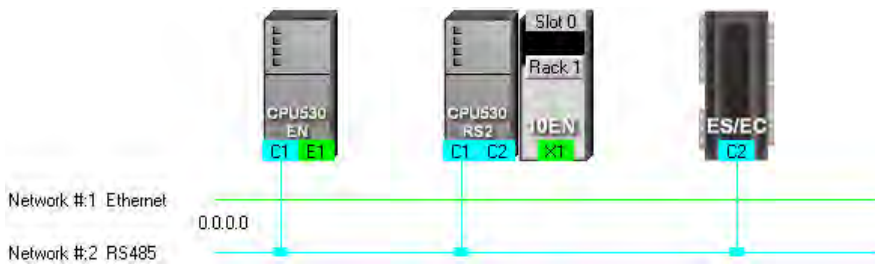
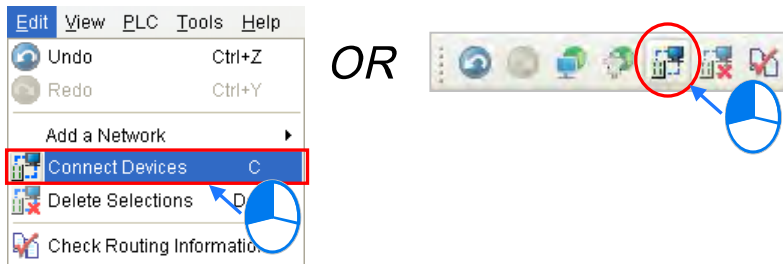


● **Connecting a single device or several devices to an existing network**

- (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.

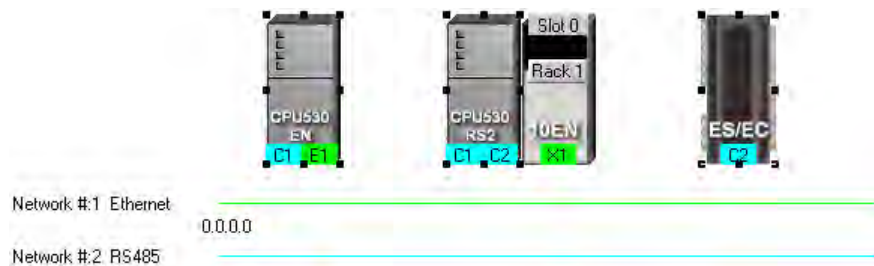



**Additional remark**

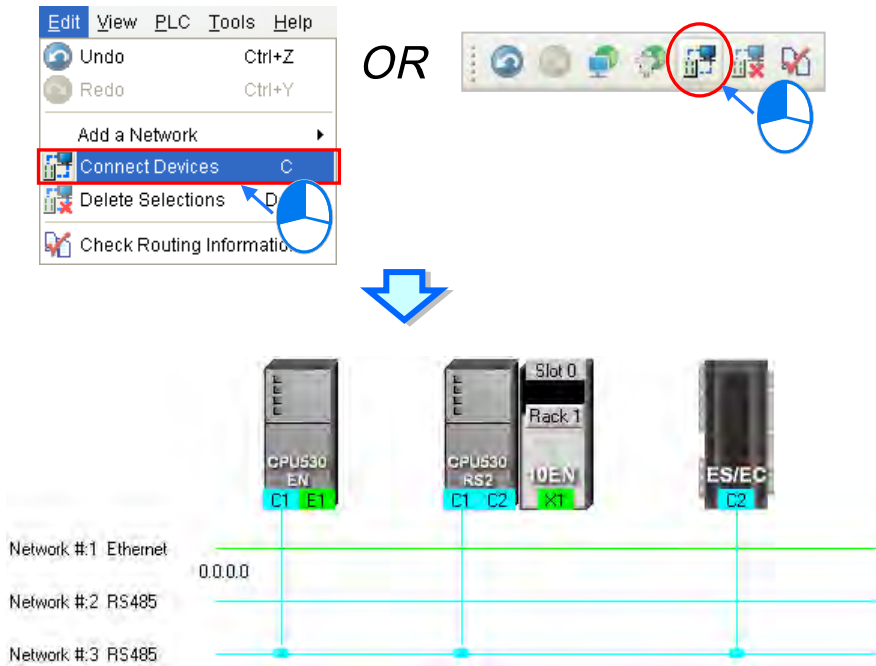
- (a) If the objects selected do not conform to the two principles listed above, the system will not connect the devices selected to the network selected, and a warning message will appear.
- (b) If a device selected has more than one port which is not connected to any network and matches the network 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.

**● 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**

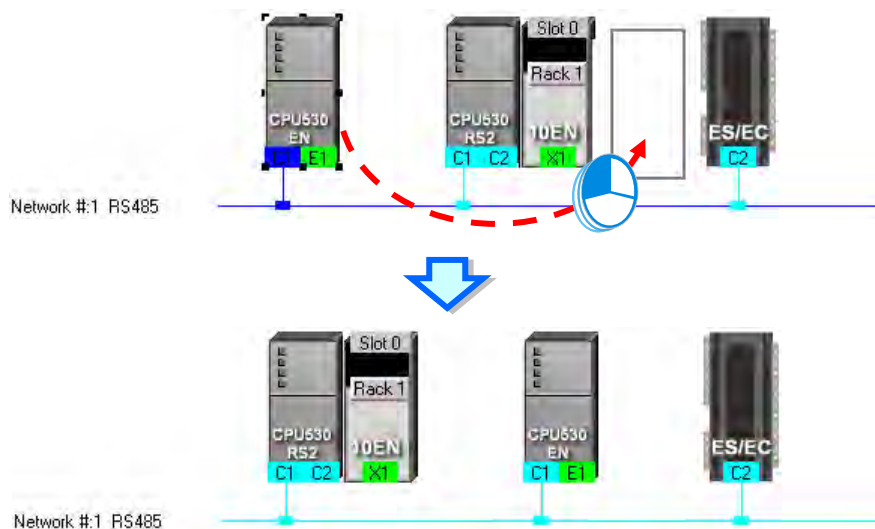
- (a) If the devices selected do not conform to the principle described in (1), the system will not connect the devices selected to a new network, and a warning message will appear.
- (b) If a device selected has more than one port which is not connected to any network and matches the new network added, 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.



### 20.2.3 Adjusting or Deleting Devices or Networks

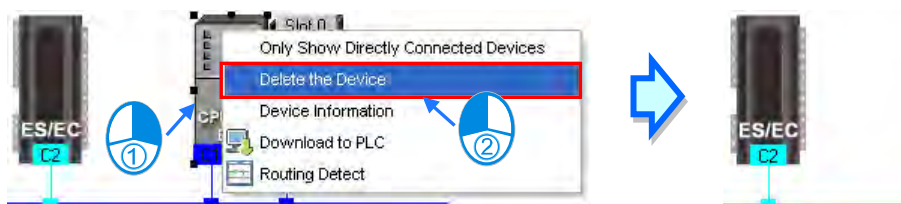
- **Adjusting the order in which the nodes in the working area are arranged**

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 cannot drag a node to a position above or under another node.

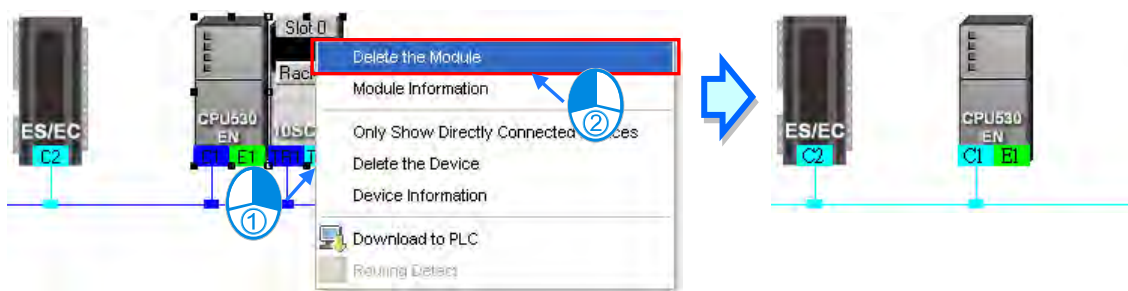


- **Deleting a single device by means of a context menu**

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 cannot be deleted.

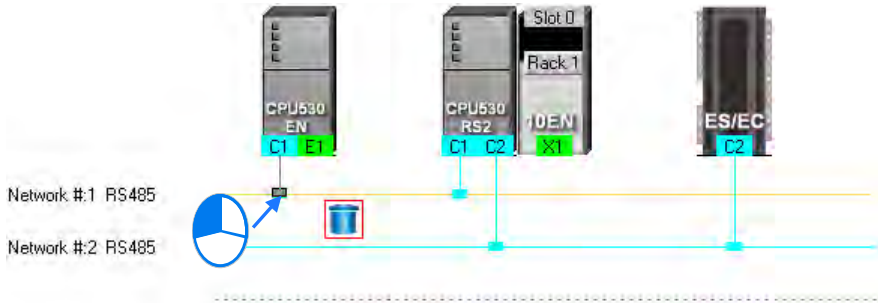


After the users right-click a module, and click **Delete the Module** on the context menu, the module will be deleted.

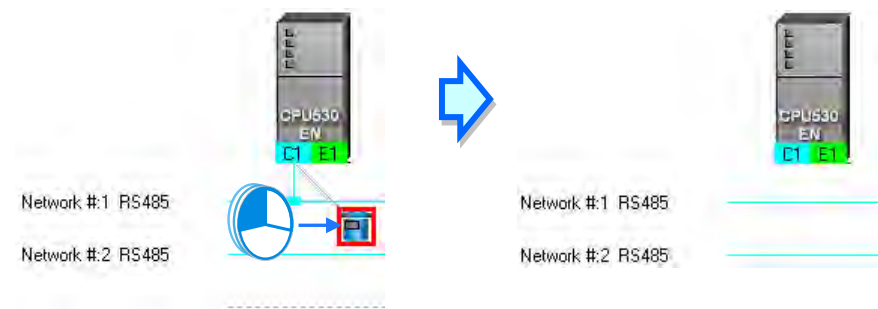


● **Adjusting a connection**

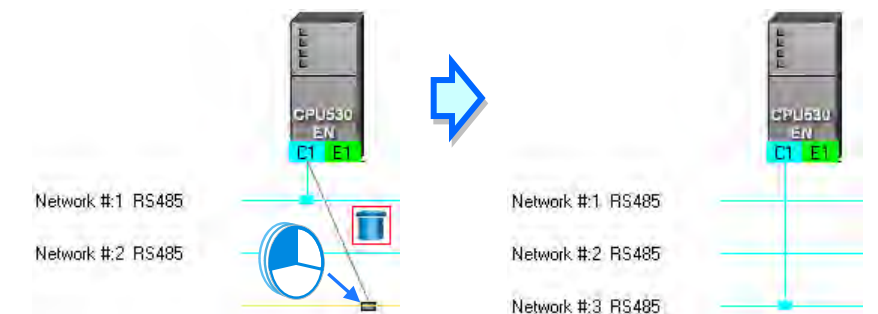
If users press the left mouse button while the mouse cursor hovers over a connection point which connects a network and a port, a small picture representing a trash can and a dotted line will appear.



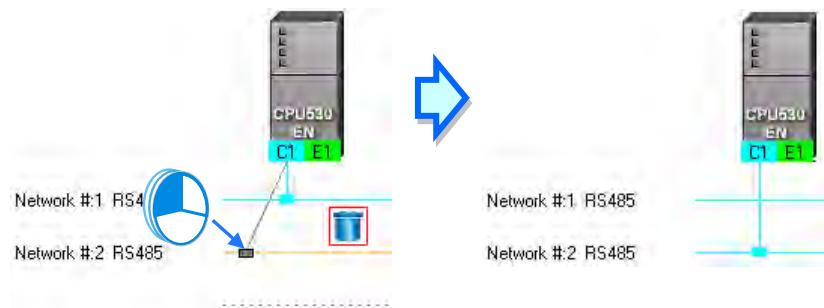
If the users release the left mouse button after they drag the connection point to the small picture representing a trash, the connection between the network and the port will be canceled.



If the users drag the connection point to the dotted line, the system will connect the port to the new network added.

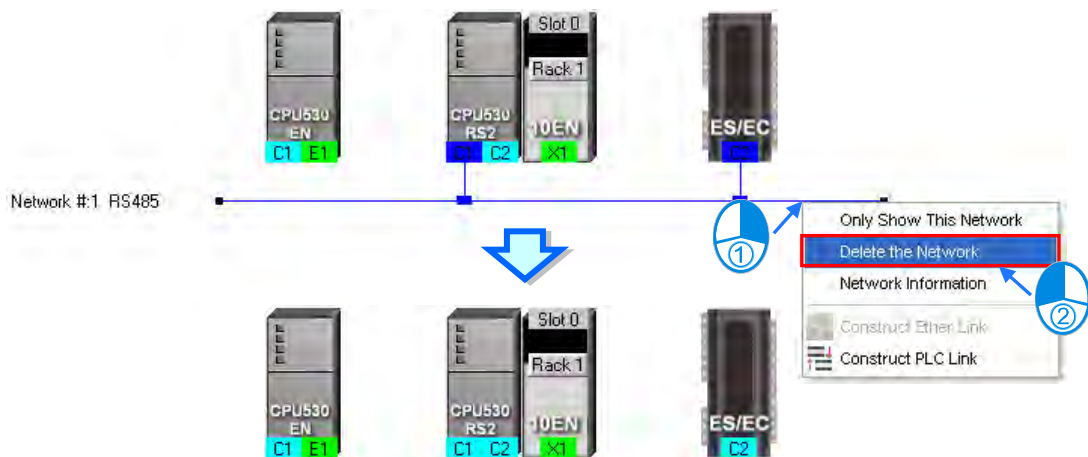


If the users release the left mouse button after they drag the connection point to another network which matches the port, the port will be connected to the network.



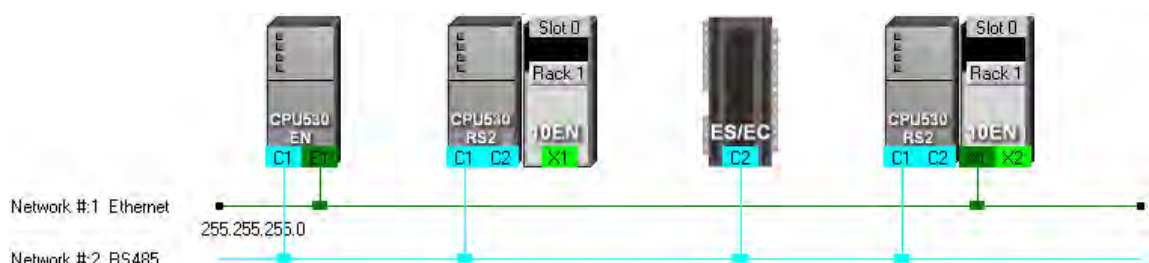
- **Deleting a single network by means of a context menu**


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.

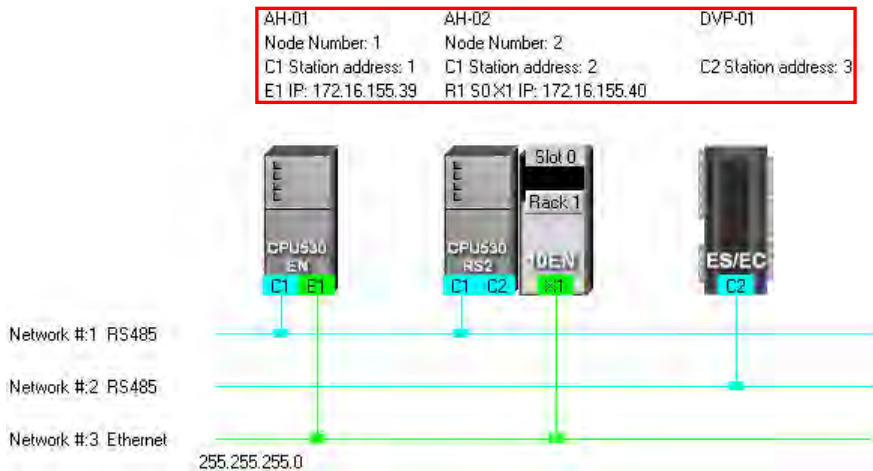


After the users click **Delete Selections** on the **Edit** menu, click  on the toolbar, or press Delete on the keyboard, the objects selected will be deleted. However, the PLC for which a project is created and the modules connected to the PLC cannot be deleted. Besides, if a PLC is deleted, the modules connected to the PLC will also be deleted.

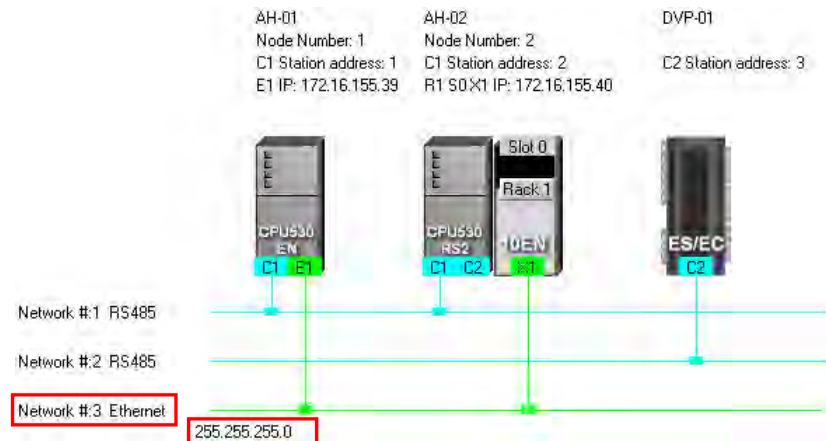


### 20.2.4 Setting the Attributes of a Node/Network

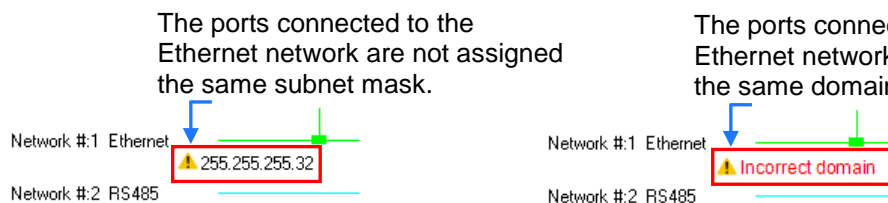
After users deploy the nodes in the NWCONFIG window, the information about the nodes will put above the nodes. The information includes PLC names and node numbers. The attributes of the ports connected to networks are also displayed. If a port is an RS-485 port, a station address will be displayed. If 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.



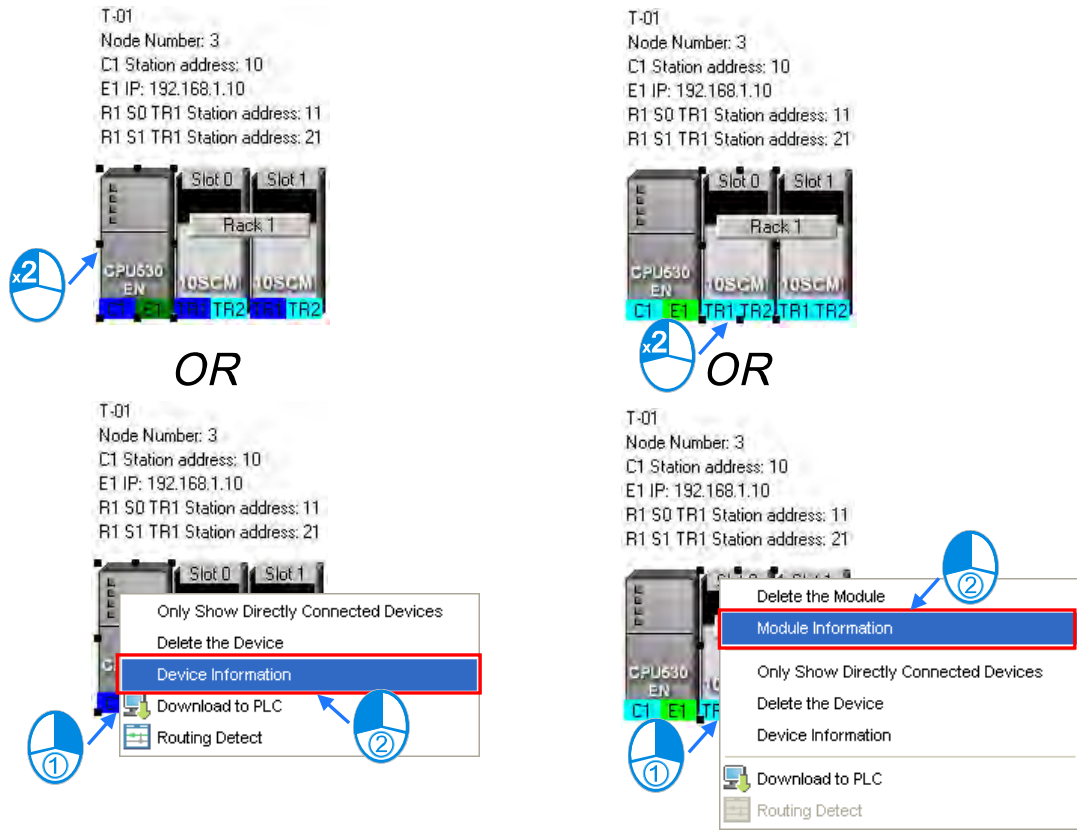
If the ports connected to an Ethernet network are not in the same domain, or are not assigned the same subnet mask, a warning sign will appear. If the ports connected to an Ethernet network are not assigned the same subnet mask, the strictest subnet mask will be shown.



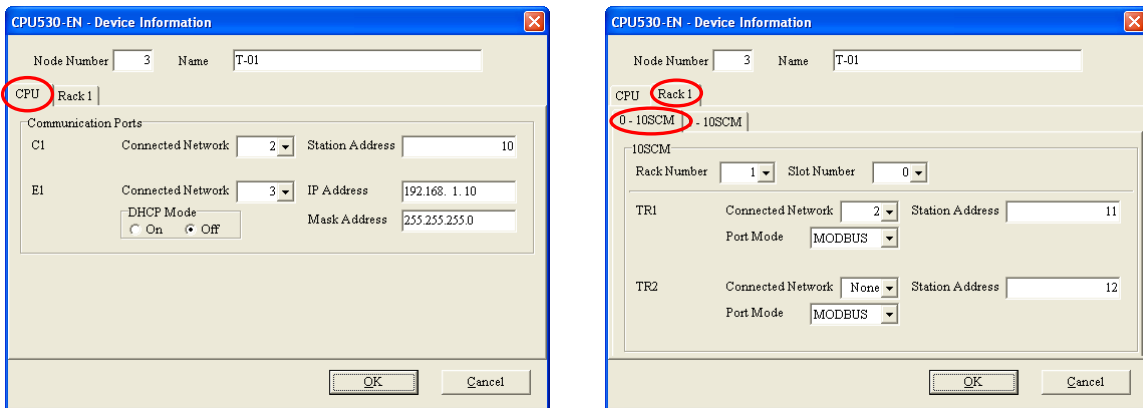
● **Setting the attributes of a node**

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.

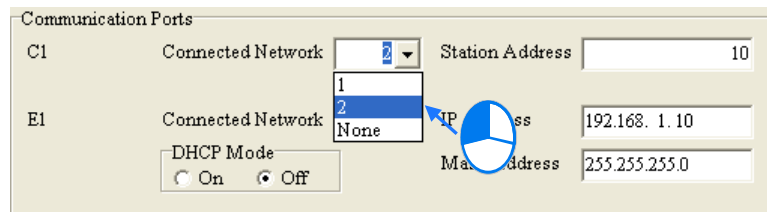


In the **Device Information** window, there are two tabs. The page displayed in the window depends on the device selected.

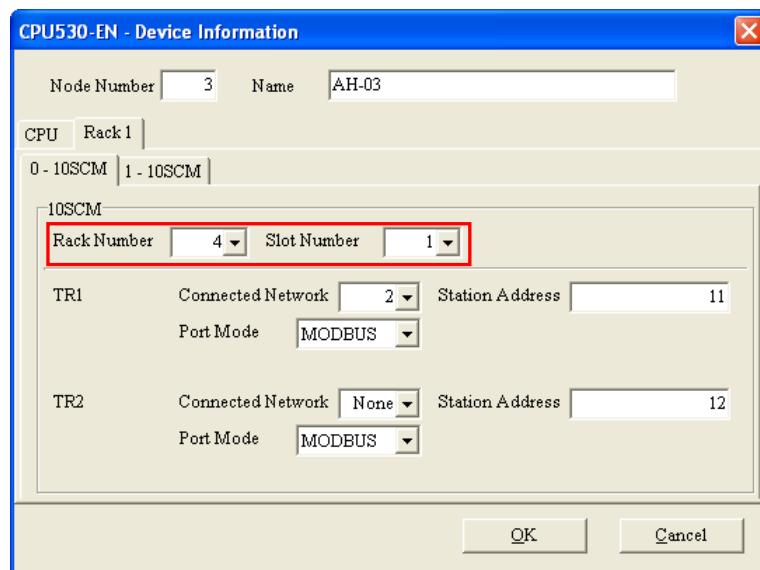


When the device selected based on project setting is AH5x0 series, most of the columns in the property window is greyed, so property settings can only be completed through HWCONFIG.

In the setting page, every communication port has a corresponding **Connected Network number** to choose from the drop-down list box. When **None** is selected, the port connection is deleted.



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.

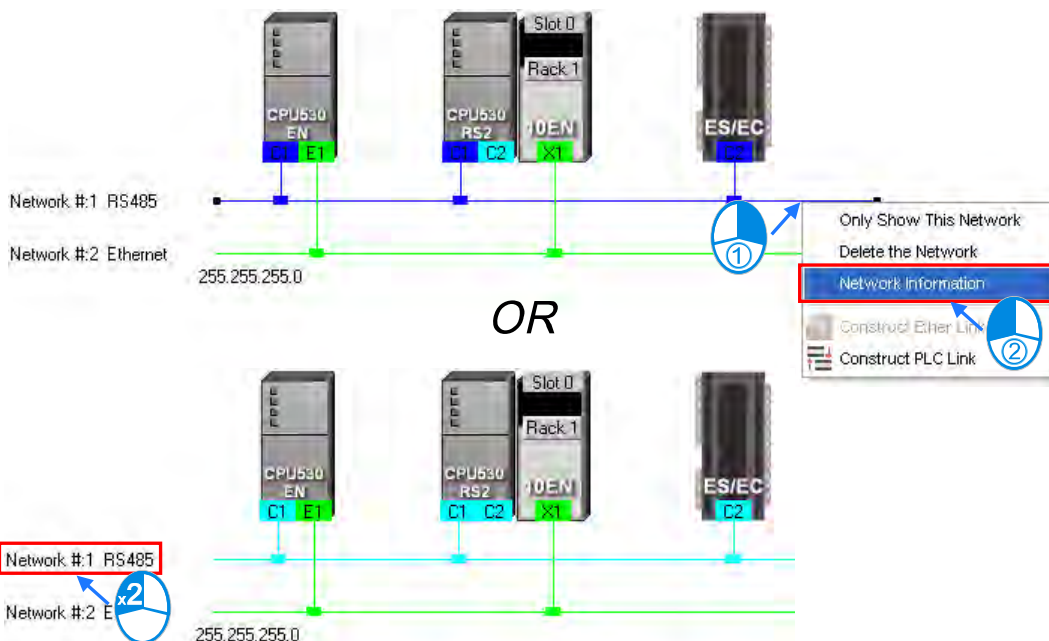


After the users complete the setting of the attributes of a node, and click **OK** in the **Device Information** window, the attributes of the node will be updated immediately.

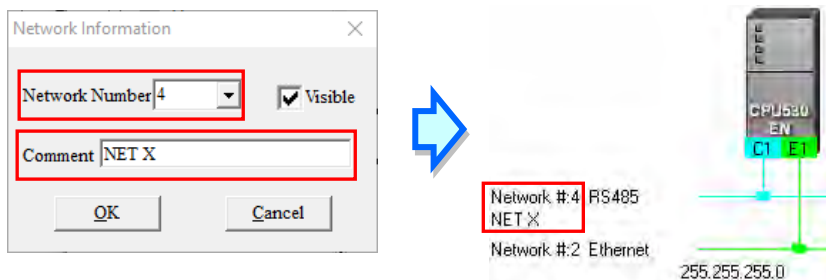


● **Setting the attributes of a network**

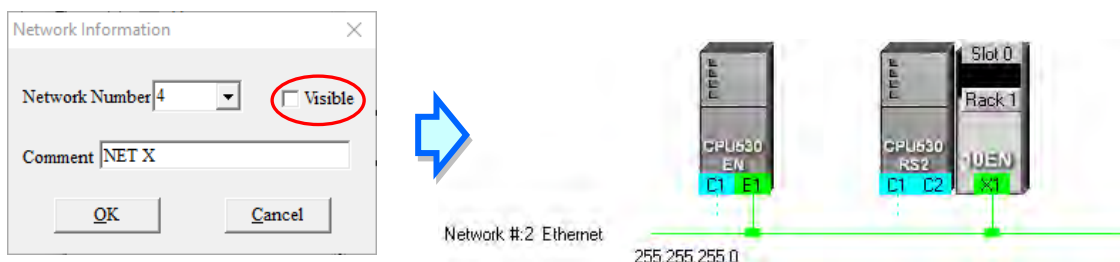
After users double-click a network, the **Network Information** window will appear. The users can also open the **Network Information** window by right-clicking the network, and clicking **Network Information** on the context menu.



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.



Un-click **Visible** in the **Network Information** window, the network will be hidden and connected communication ports are displayed in dotted lines.

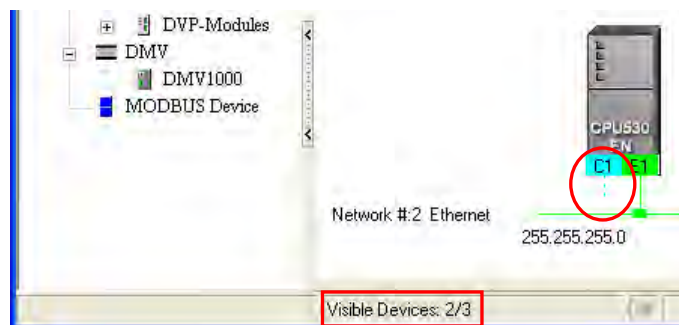


20



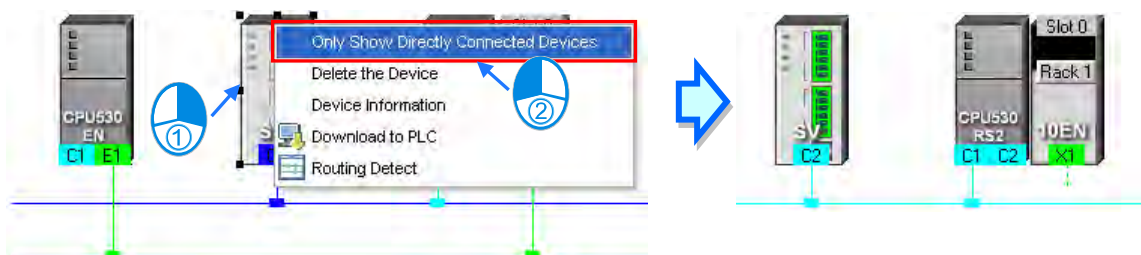
### 20.2.5 Hiding/Displaying Devices or Networks

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.



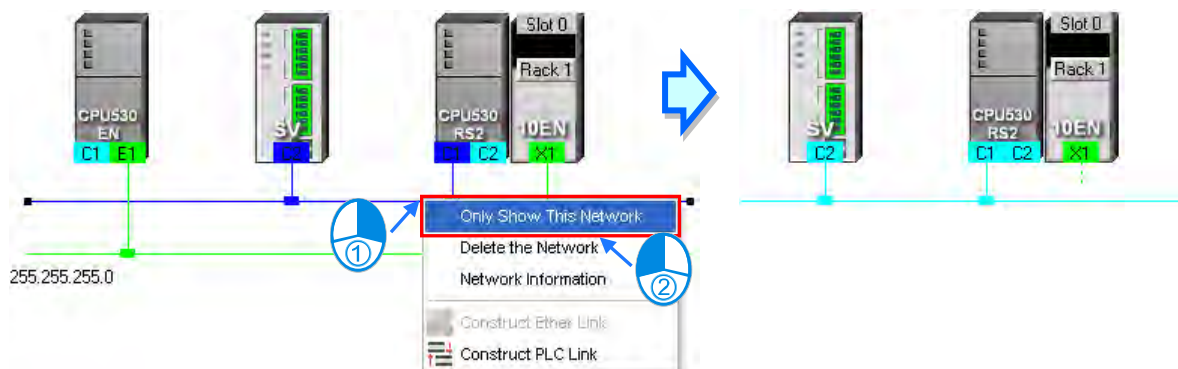
- **Only displaying the objects connected to a device**

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.



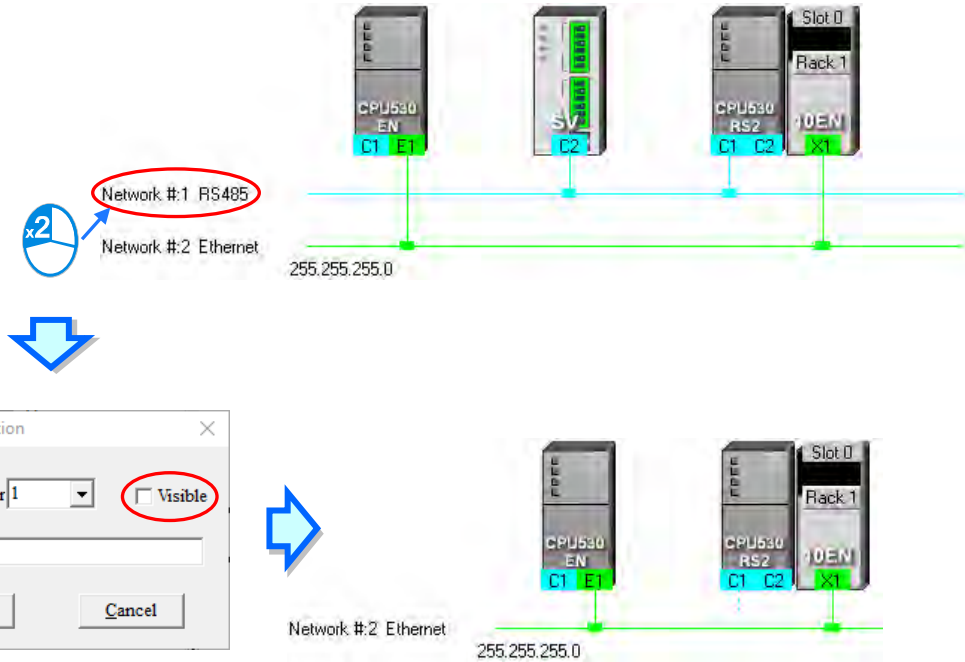
- **Only displaying the devices connected to a network**

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.



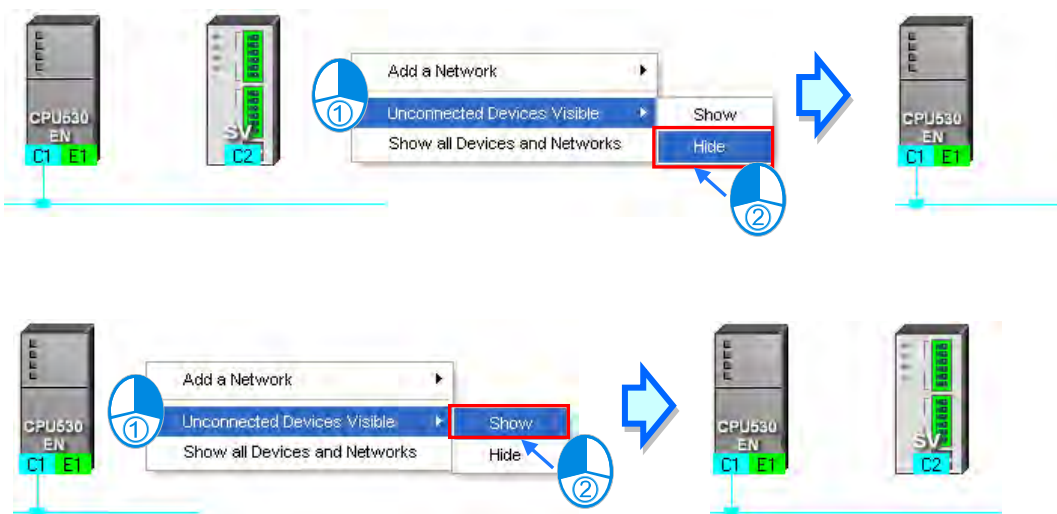
● **Hiding a network and the devices connected to the network**

Unclick **Visible** in the **Network Information** window, the network is hidden and connected communication ports are displayed in dotted lines.



● **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**

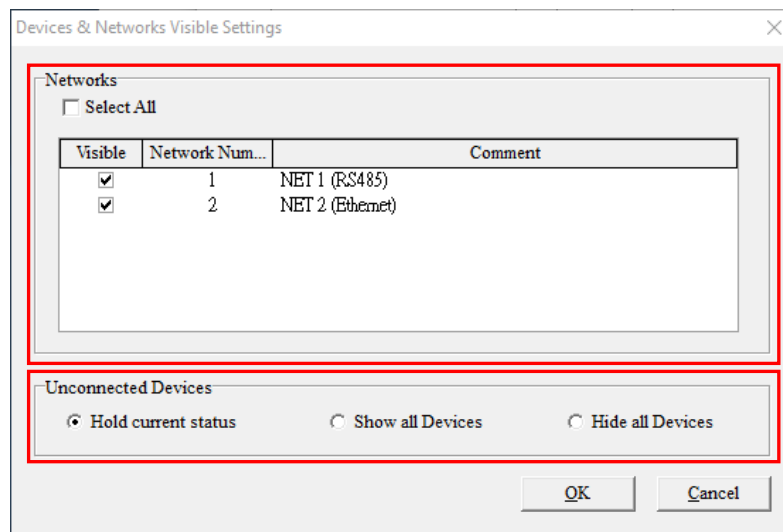
Users can set the display states of all the devices at the same time.

Click **Device & Network Visible Settings** on the **View** Menu, or  on the toolbar.



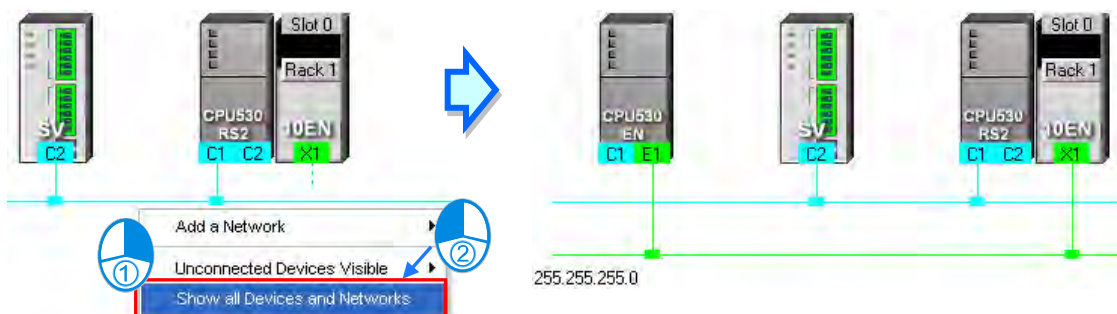
Select networks which will be displayed in the **Networks** section in the **Devices & Networks Visible Settings** window. The networks which are not selected will not be displayed in the working area. If the users select the **Select All** checkbox, all the networks in the **Networks** section will 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 later.




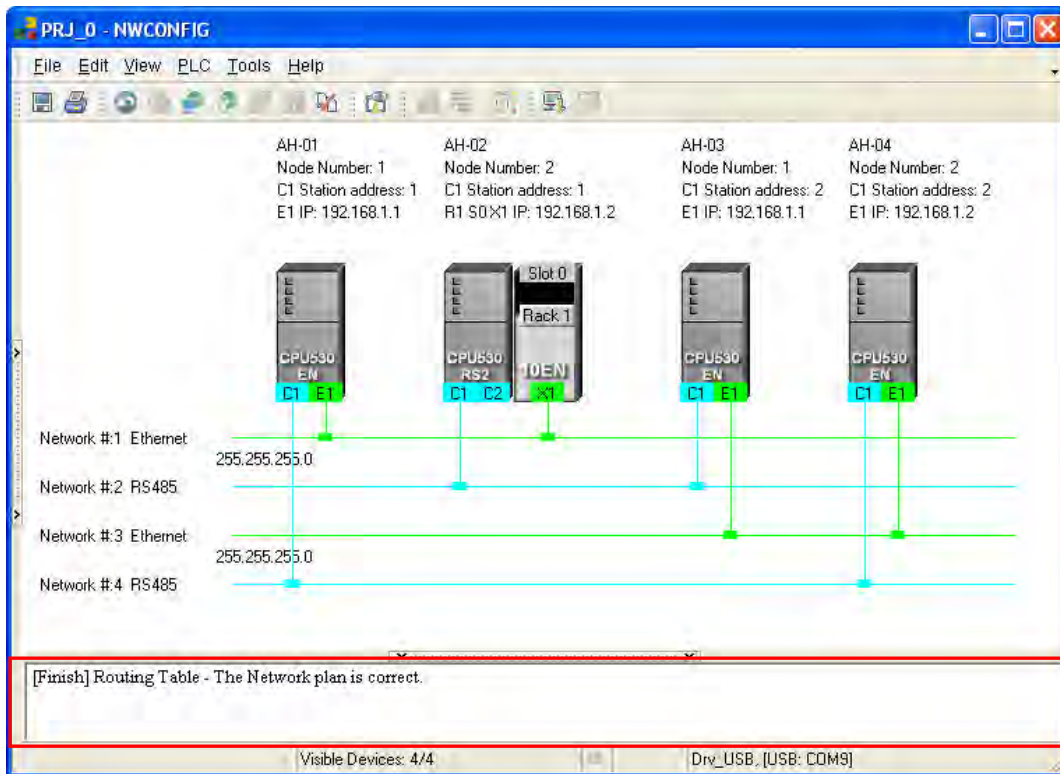
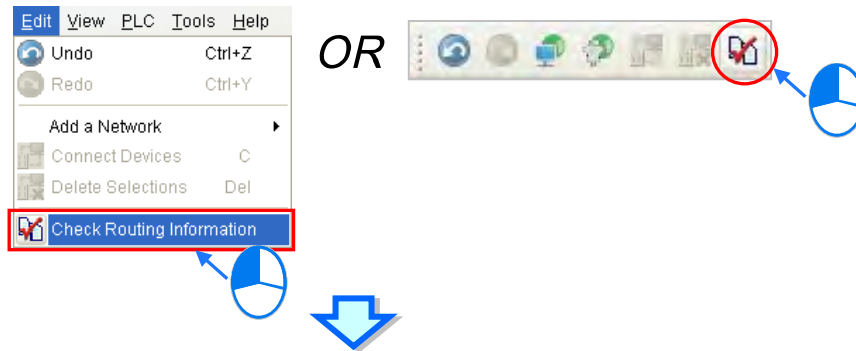
● **Displaying all the objects**

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.



## 20.2.6 Correct Network Architecture

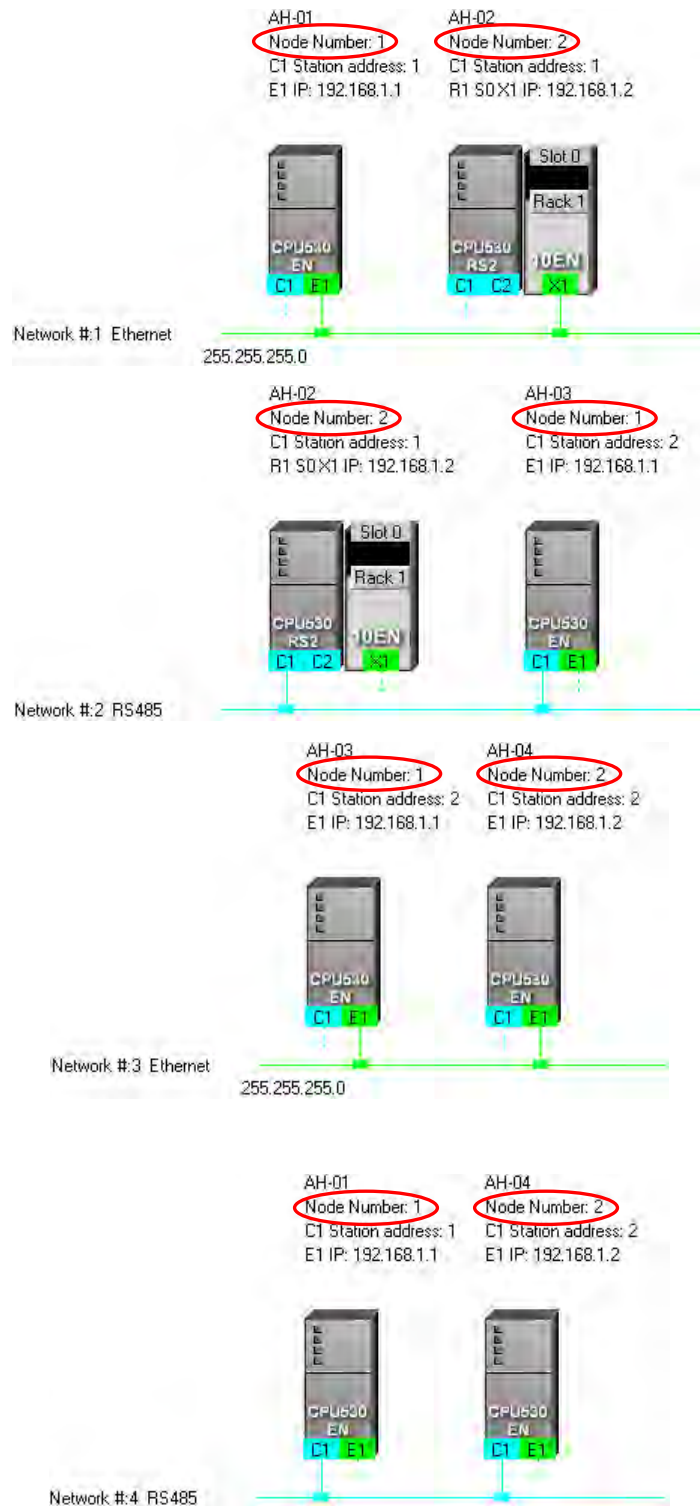
After users click **Check Routing Information** on the **Edit** menu, or  on the toolbar, the system will check whether the network architecture the users create is correct, and the check result will be displayed in the message display area.



When checking, the system only focuses on the connected ports or the nodes and internet information in the Graphic Workspace for confirmation. The items to check include **nodes number**, **RS485 station address** and **IP address**. From the above image, the issue of repetition in nodes number, RS485 station address and IP address appears, therefore, we can check an internet one at a time to make sure that it is connected to the same node on the internet and the **nodes number**, **RS485 station address** and **IP address** are not repeated.

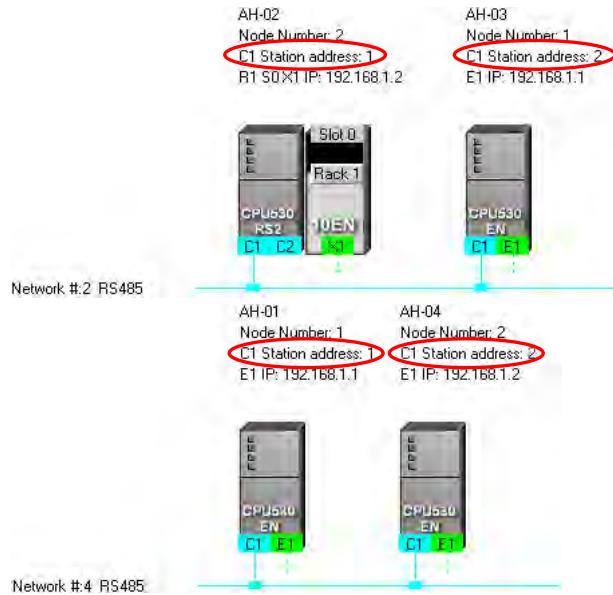
● **Node number**

In principle, the node number assigned to a node in a network cannot 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.



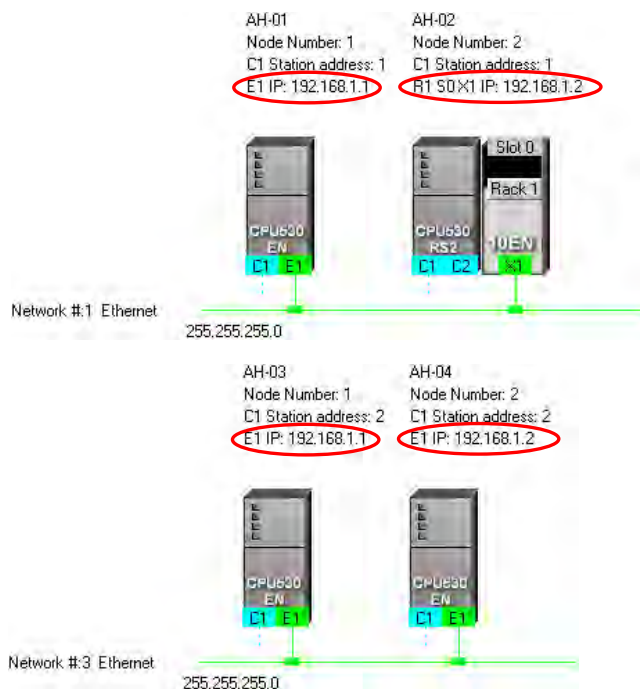
● **RS-485 station address**

In principle, the RS-485 station address of a port in a network cannot 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 cannot 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.



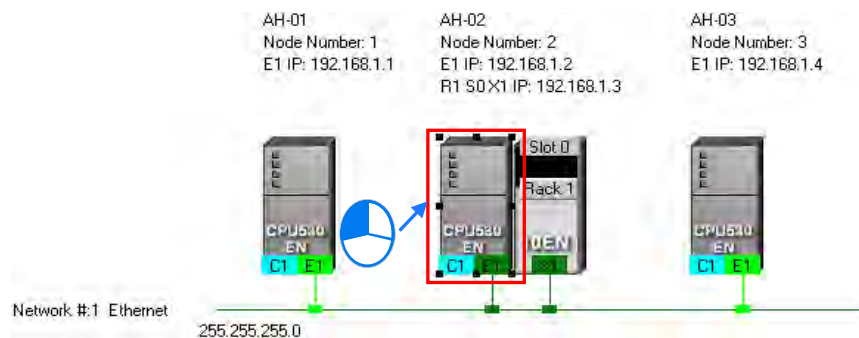
## 20.2.7 Downloading Routing Tables


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.

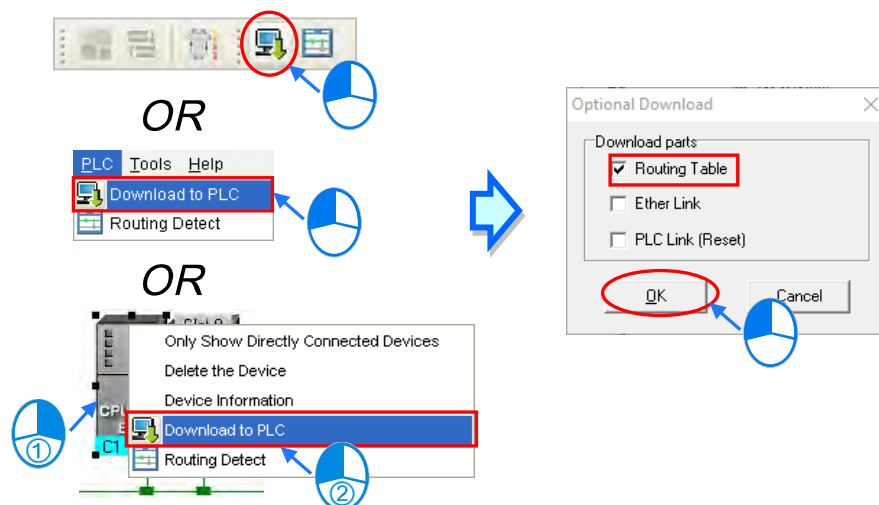
The users can download the routing tables produced to nodes one by one, or download the routing tables produced to nodes at the same time. If the users want to download the routing tables produced to nodes, the **Routing Mode** checkbox in the **Select a Driver** window must be unselected.

### ● Single node

First, select a node to download and click on the node in the host part from the graphic workspace. Since the routing function only supports AH5x0 series, therefore, selecting other PLC types or devices will be unable to download the **Routing Table**.

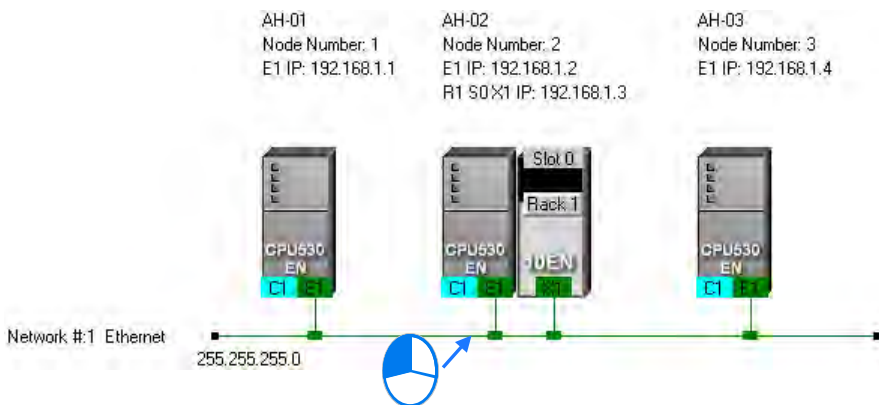



After the users click **Download to PLC** on the **PLC** menu, or  on the toolbar, the **Optional Download** window will appear. The users can also open the **Optional Download** window by right-clicking the device they select, and clicking **Download to PLC** on the context menu. After the users select the **Routing Table** checkbox in the **Optional Download** window, and click **OK**, a routing table will be downloaded to the device.

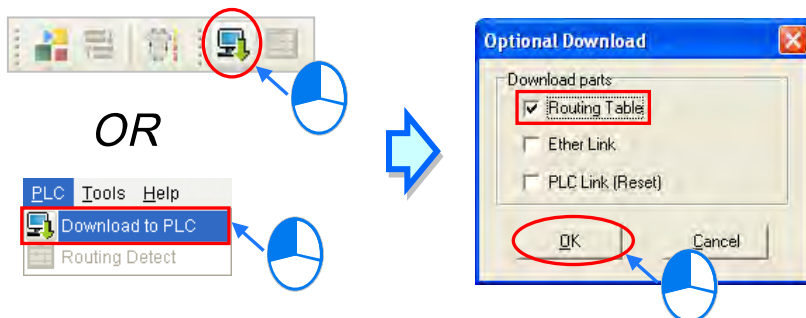


● **Multiple nodes**

The COMMGR Driver has to be under Ethernet structure in the communication setting. Then, choose an Ethernet network from the graphic area and parameters can download to connected nodes.



After the users click **Download to PLC** on the **PLC** menu, or  on the toolbar, the **Optional Download** window will appear. 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 cannot be selected. Please refer to the following sections for more information about Ether Links and PLC Links.

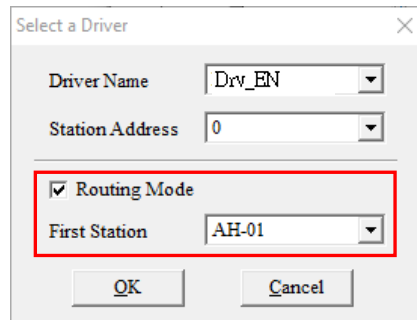
Besides, the routing data stored in a PLC is data related to the PLC itself. The users cannot upload the routing data stored in a PLC. The system does not provide the function of uploading routing data.




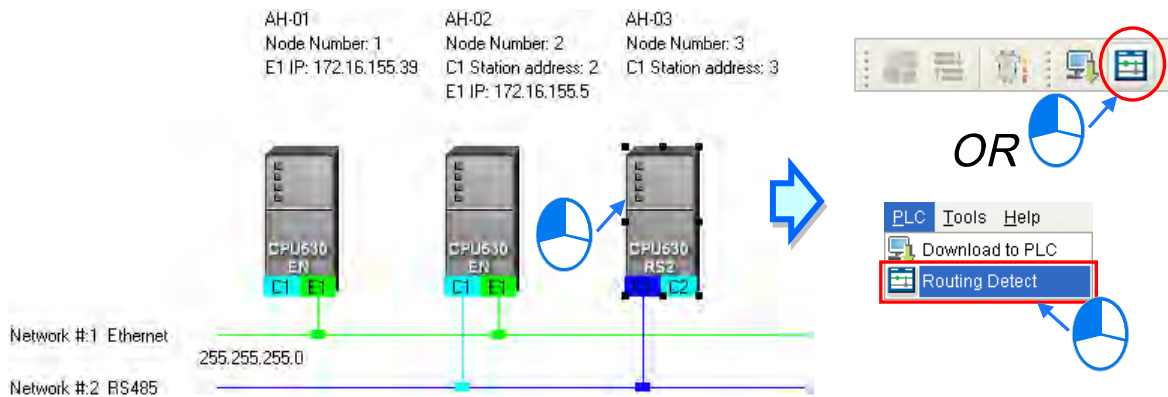
### 20.2.8 Testing Routing

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.

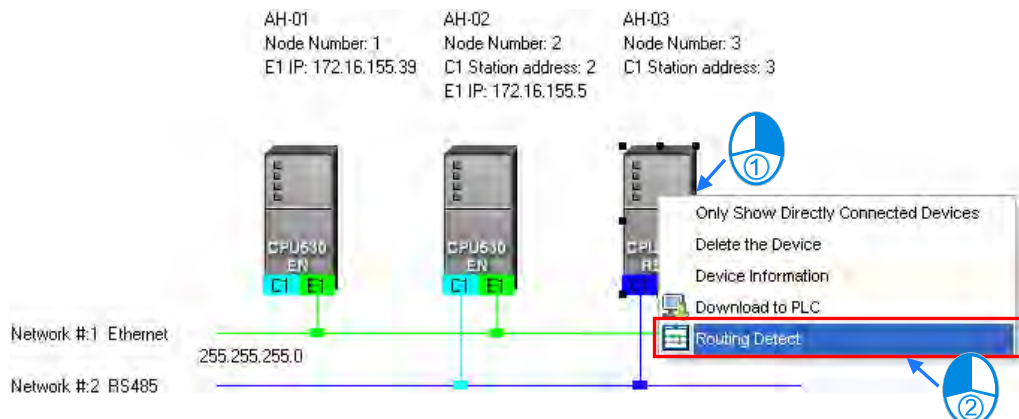
- (1) The users have to make sure that all the nodes are wired according to the configuration in NWCONFIG, and operate normally. The setting of the devices has to be consistent with the setting in NWOCNIFG. In the **Select a Driver** window, the users have to select the **Routing Mode** checkbox, and select a device in the **First Station** drop-down list box.



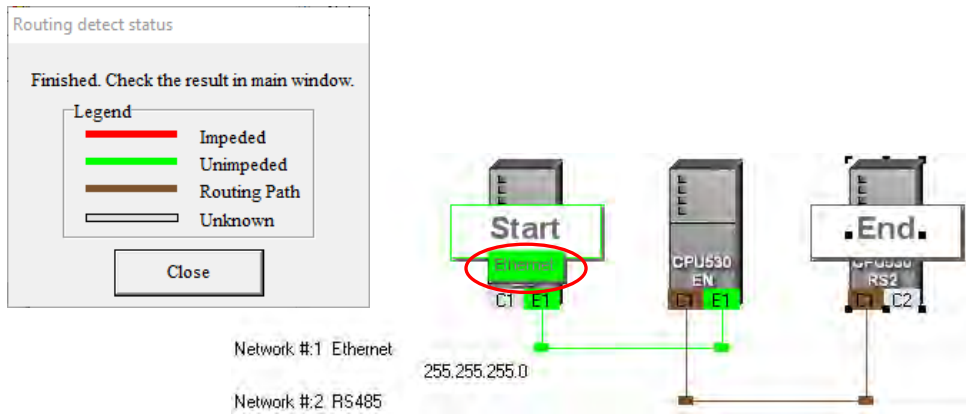
- (2) After the users select the destination device toward which packets are relayed, they have to click  on the toolbar, or **Routing Detect** on the **PLC** menu.



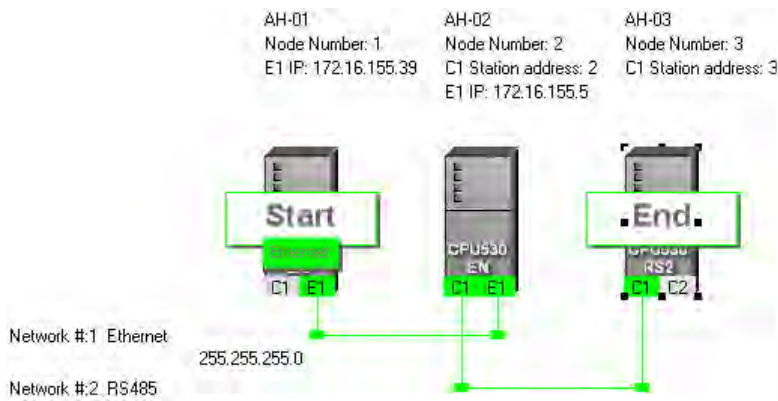
The users can also right-click the destination device, and then click **Routing Detect** on the context menu.



- (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. 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**

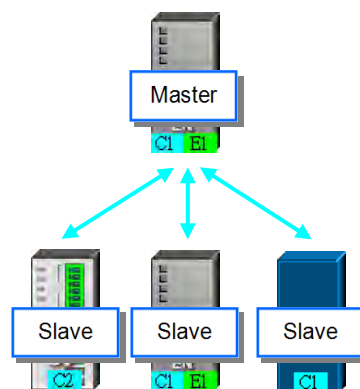
If the detection fails, the users have to make sure of the following points.

- (a) Please confirm hardware configuration is the same as NWCONFIG configuration and every node is correctly connected to the internet and functioning.

## 20.3 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 cannot exchange data. They have to exchange data through the master.



### 20.3.1 Opening the PLC Link Table Editor Window

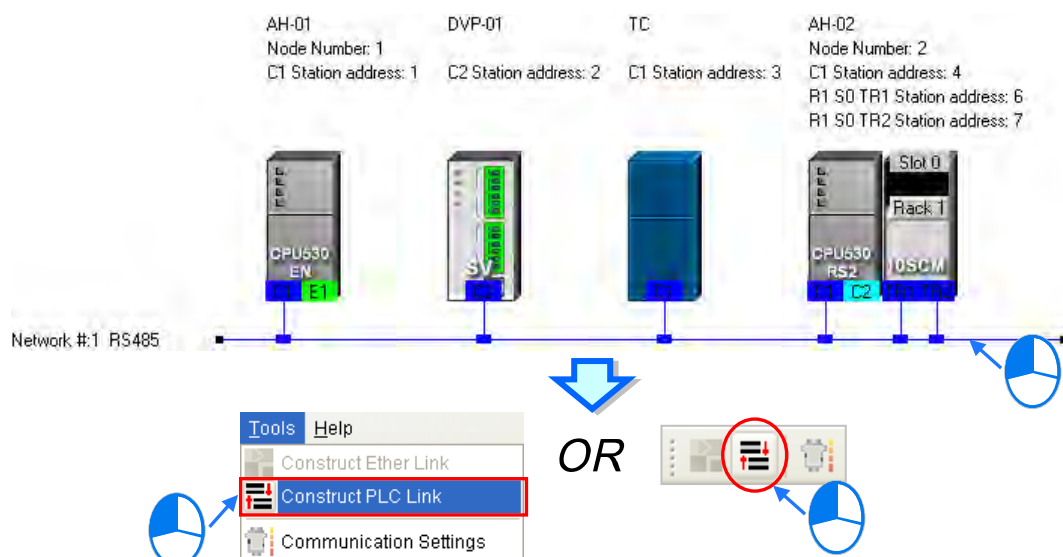
Before creating a PLC Link, users have to make sure that all the network setting is correct. Please use one of the methods described below.

- **Method 1**

After the users select a network, they have to click **Construct PLC Link** on the **Tools** menu, or

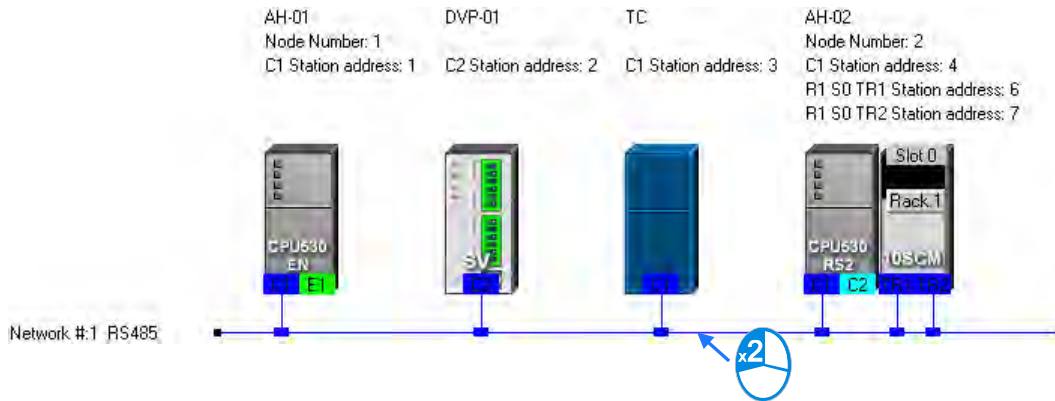


on the toolbar.



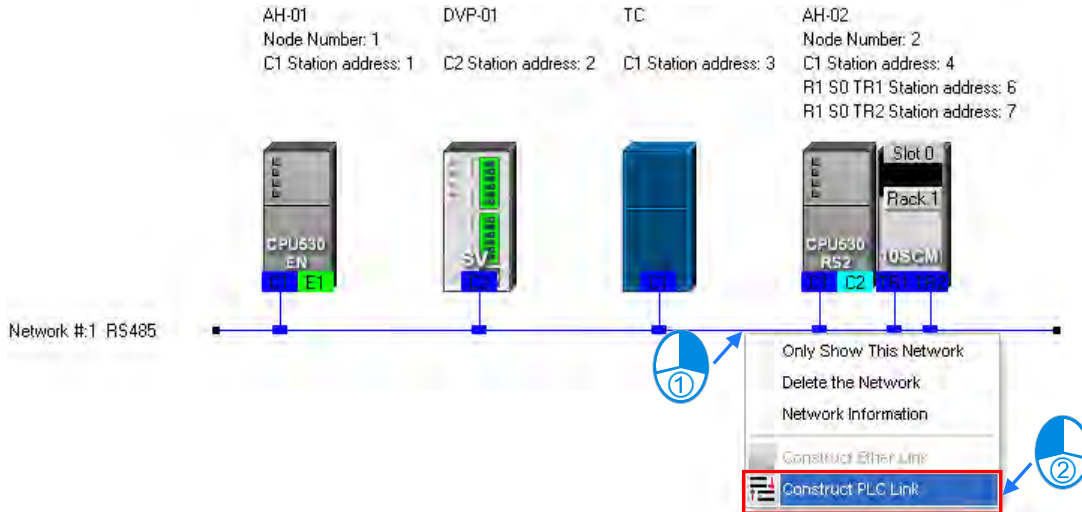
● **Method 2**

Double-click a network.



● **Method 3**

Right-click a network, and then click **Construct PLC Link** on the context menu.

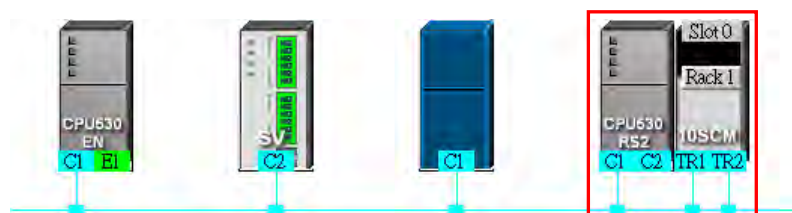


After the users use one of the methods described above, the **PLC Link Table Editor** window will be opened. The **PLC Link Table Editor** window leads the users to construct a PLC Link step by step. The steps of constructing a PLC Link are designating a port as a master station, setting communication 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.

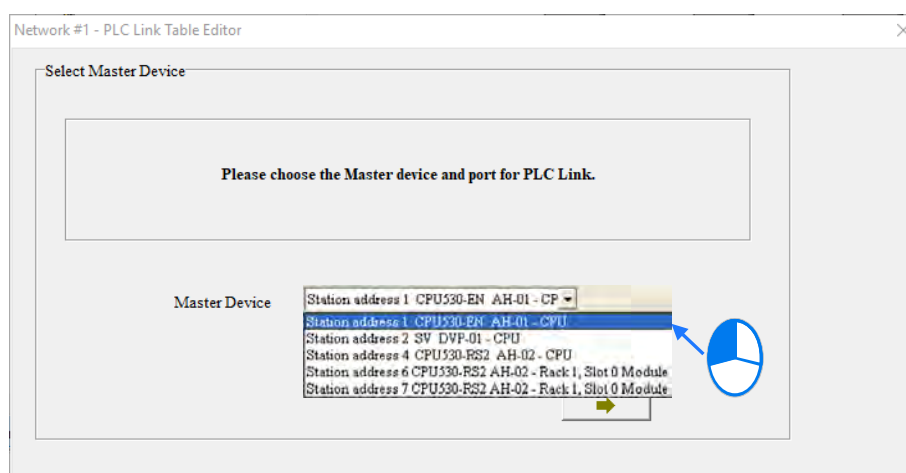
Besides, a PLC Link is executed through special relays and special registers. The construction tool introduced here is just a friendly user interface which helps users download the parameters related to a PLC Link to the related special relays and the related special registers. The exact execution of 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.


### 20.3.2 Select Master Station Device (Step 1)

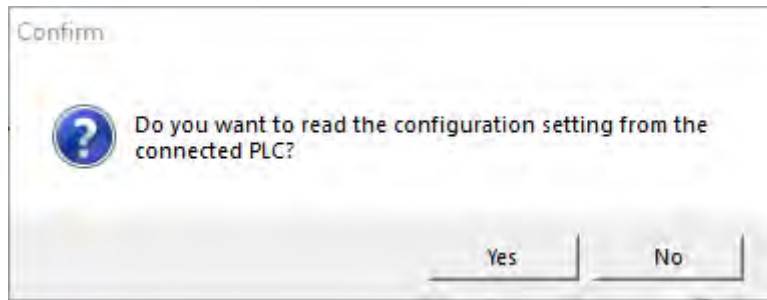
Only an AH5x0 series CPU module, an AH500 series module, a DVP series PLC, or a DVP series module can be designated as the master station. Not all ports of a PLC or a module can be designated as the master station, and therefore users have to read the usage of the PLC or the module before they designate 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.

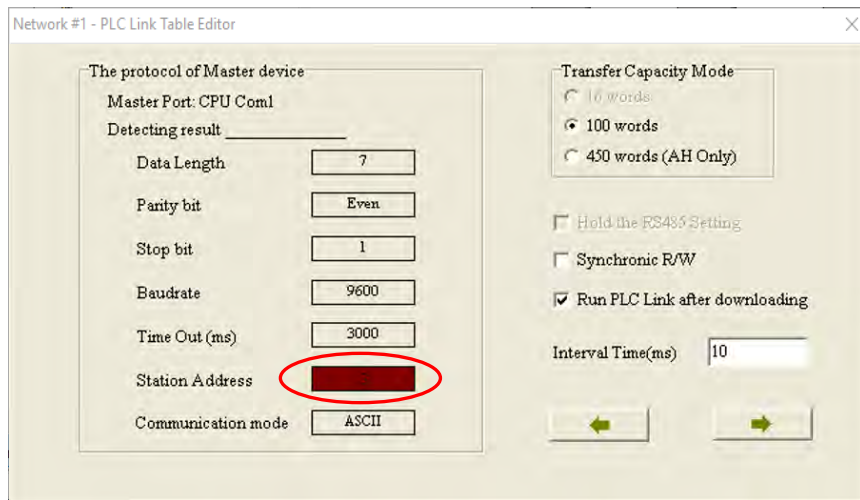


- (2) After users click , the system will ask the users whether they want to upload the setting related to a PLC Link through the master station. If the users click **No**, they will be led to the second step. If the users click **Yes**, the setting related to a PLC Link in the PLC which is a part of 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 status is OK.
  - (c) The users have to make sure that they have completed the communication setting in NWCONFIG.

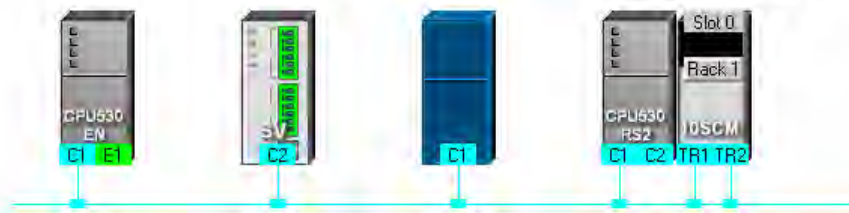


### 20.3.3 Communication Parameter Settings (Step 2)

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 from the station address assigned to the master station, the **Station Address** box will become red.



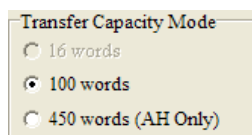
AH-01 Node Number: 1 C1 Station address: 1	DVP-01 C2 Station address: 2	TC C1 Station address: 3	AH-02 Node Number: 2 C1 Station address: 4 R1 S0 TR1 Station address: 6 R1 S0 TR2 Station address: 7
--	---------------------------------	-----------------------------	--



Network #1 RS485

- **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.



Setting	Description	DVP	AH5x1
16 Words	A single data exchange transfer capacity length is 16 Words. Use only Device D.	V	V
100 Words	A single data exchange transfer capacity length is 100 Words. Use only Device D.	V	V
450 Words	A single data exchange transfer capacity length is 450 Words and can use Device M, D or L. The device type for master and slave station can be		V

- **Hold the RS485 Setting**

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 host will switch from STOP to RUN and re-download the saved communication parameters; for more information on DVP COM port communication parameters, please refer to manuals regarding PLC types.

- **Synchronic R/W**

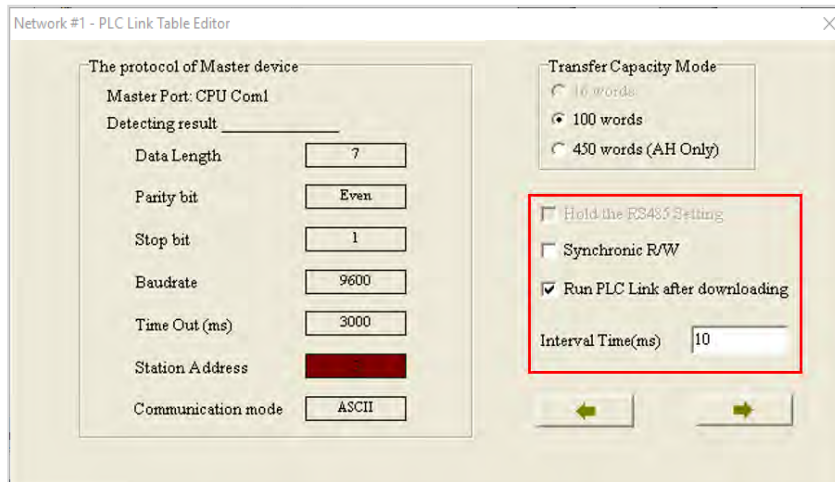
A master station sends a writing command and a reading command to a slave station separately. If the **Synchronic R/W** checkbox is selected, the master station can complete reading and writing simultaneously by means of a specific Modbus function code (the hexadecimal code 17), and the efficiency of 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 cannot be identified, and the reading/writing of data will fail after they receive the commands from the master station. For mode, choose 450 Words or users can only select Device D to enable **Synchronic R/W**.



- **Run PLC Link after downloading**

Select **Run PLC Link after downloading**. When finish planning PLC Link, download to hosts will also enable PLC Link.

● 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.

### 20.3.4 Create Data Exchange Table (Step 3)

#### 20.3.4.1 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.

#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
1	2	R	D0~D99	<=	D100~D199	100	Enabled	CPU530-RS2
		W	D0~D99	=>	D200~D299	100		
2	3	R	D100~D199	<=	D100~D199	100	Enabled	CPU530-RS2
		W	D100~D199	=>	D200~D299	100		
3	4	R	D200~D299	<=	D100~D199	100	Disabled	CPU530-RS2
		W	D200~D299	=>	D200~D299	100		
4	0	R	D300	<=	16#1000	0	Disabled	Unknown
		W	D300	=>	16#1000	0		
5	0	R	D400	<=	16#1000	0	Disabled	Unknown
		W	D400	=>	16#1000	0		

Buttons: Import, Export, Reset, Check Settings, Upload, Download, Monitor and Download, , Finish



Column	Description
<b>Serial Number (#)</b>	The data exchange groups in the data exchange table in the <b>PLC Link Table Editor</b> window are numbered. Users can set 16 data exchange groups or 32 data exchange groups, depending on the model selected.
<b>Station Addr.</b>	Indicate the station address for data exchange group. The same station address can contain multiple sets of slave station from different sections. Besides, the station address 0 represents an undefined slave station, and is not a broadcast station address.
<b>R/W</b>	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.
<b>Master Device Data</b>	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.
<b>Slave Device Data</b>	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.
<b>Length</b>	A data length is indicated. A length indicates the number of devices used.
<b>Status</b>	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.
<b>Device Type</b>	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.

#	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		
2	2	R	D3000~D3009	<=	D3000~D3009	10	Enabled	SV
		W	D3200~D3209	=>	D3200~D3209	10		
3	0	R	D0~D99	<=	16#0000~16#0063	100	Disabled	Unknown
		W	D0~D99	=>	16#0000~16#0063	100		
4	3	R	D3400~D3424	<=	16#1000~16#1018	25	Enabled	MODBUS Device
		W	D3500~D3524	=>	16#1025~16#103D	25		
5	6	R	D3600~D3699	<=	D3000~D3099	100	Enabled	CPU530-RS2
		W	D3700~D3799	=>	D3100~D3199	100		

Buttons: Import, Export, Reset, Check Settings, Upload, Download, Monitor and Download, Finish

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 data from slave station no.2 (SV series) in D3000~D3009, a total of 10 words and stores the data in D3000~D3009. Meanwhile, the master station data in D3200~D3209, a total of 10 words is written in slave station D3200~D3209.
#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~D3799 in the master station is written into D3100~D3199 in the slave station.

20

20.3.4.2 Setting a Data Exchange Group

Double-click on the data exchange group for setup to open the setting window.

Network #1 - PLC Link Table Editor

#	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		
2	2	R	D3000~D3009	<=	D3000~D3009	10	Enabled	SV
		W	D3200~D3209	=>	D3200~D3209	10		
3	0	R	D0~D99	<=	16#0000~16#0063	100	Disabled	Unknown
		W	D0~D99	=>	16#0000~16#0063	100		
4	3	R	D3400~D3424	<=	16#1000~16#1018	25	Enabled	MODBUS Device
		W	D3500~D3524	=>	16#1025~16#103D	25		
5	6	R	D3600~D3699	<=	D3000~D3099	100	Enabled	CPU530-RS2
		W	D3700~D3799	=>	D3100~D3199	100		

Import Export Reset Check Settings Upload Download Monitor and Download Finish



Parameter Setting

Linked Device

Station Address: 2 Device Type: SV

Linked Status:  Disable  Enable

Read

Master Parameter Setting: Starting Address: D 3000 Data Length: 100 Words

Slave Parameter Setting: Starting Address: D 2500

Write

Master Parameter Setting: Starting Address: D 3100 Data Length: 50 Words

Slave Parameter Setting: Starting Address: D 2600

OK Cancel

● **Linked Device**

The **Station Address** is the slave station for data exchange group. Do not select the master station address but slave station address used in other data exchange group can be selected. When choosing 0, Unknown appears in the Device Type column meaning not defined. Otherwise, a hardware device name will appear in corresponds to the slave station address.

Linked Device

Station Address: 4 Device Type: CPU530-RS2

Linked Status:  Disable  Enable

● **Linked Status**

To set on the linked status to enable data exchange for data exchange groups.

Linked Device		Linked Status
Station Address	Device Type	<input type="radio"/> Disable
4	CPU530-RS2	<input checked="" type="radio"/> Enable

● **Read**

The device type on both sides should be the same. The data length is the number for exchange. To read from the device starting address of Slave Parameter Setting the device starting address of Master Parameter Setting.

Read	
Master Parameter Setting	Slave Parameter Setting
Starting Address	Starting Address
Data Length	
D	D
3000	2300
100 Words	

● **Write**

The device type on both sides should be the same. The data length is the number for exchange. To read from the device starting address of Master Parameter Setting the device starting address of Slave Parameter Setting.

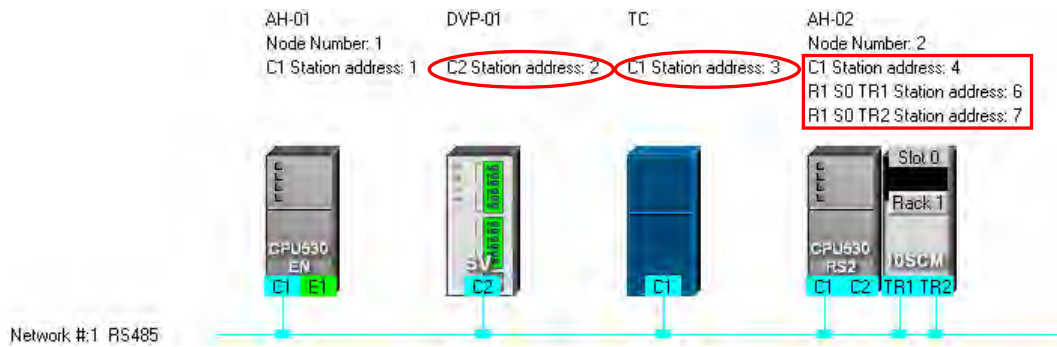
Write	
Master Parameter Setting	Slave Parameter Setting
Starting Address	Starting Address
Data Length	
D	
3100	2600
50 Words	

For section planning, the same section data can write into multiple different positions but does not allow different section data to store in the same position. Base on the image above, the content data in master station D3100~D3149 can repeat and write in different slave station, but section in master station D3000~D3099 cannot receive data from other slave stations.

**20.3.4.3 Device Synchronization in Data Exchange Table**

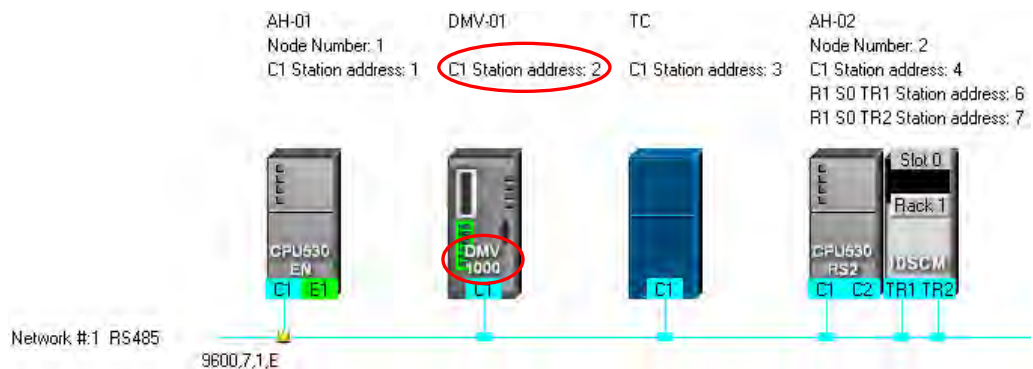
In **Device Type** column of the data exchange table, the appearing content is based on the host model type of each station address as well as according to the current configuration in the **NWCONFIG** Graphic Work space; when a host has multiple station address, the host type is shown in the **Device Type** column.

#	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		



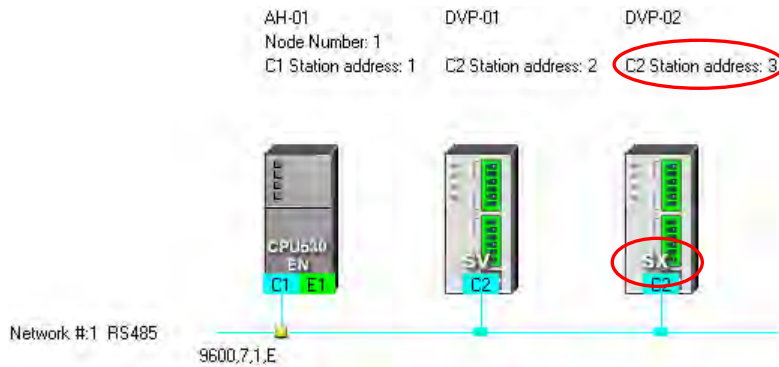
When finish planning and data setting is saved, the modified host type is marked in red to remind users and the status is auto-switched to **Disabled**.

#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
1	2	R	D3000~D3099	<=	D2500~D2599	100	Disabled	DMV1000
		W	D3100~D3149	=>	D2600~D2649	50		




When the master station device is AH5x0 series and executing upload but finds the current NWCONFIG not matched, the modified PLC type is marked in red to remind users and the status is auto-switched to **Disabled**.

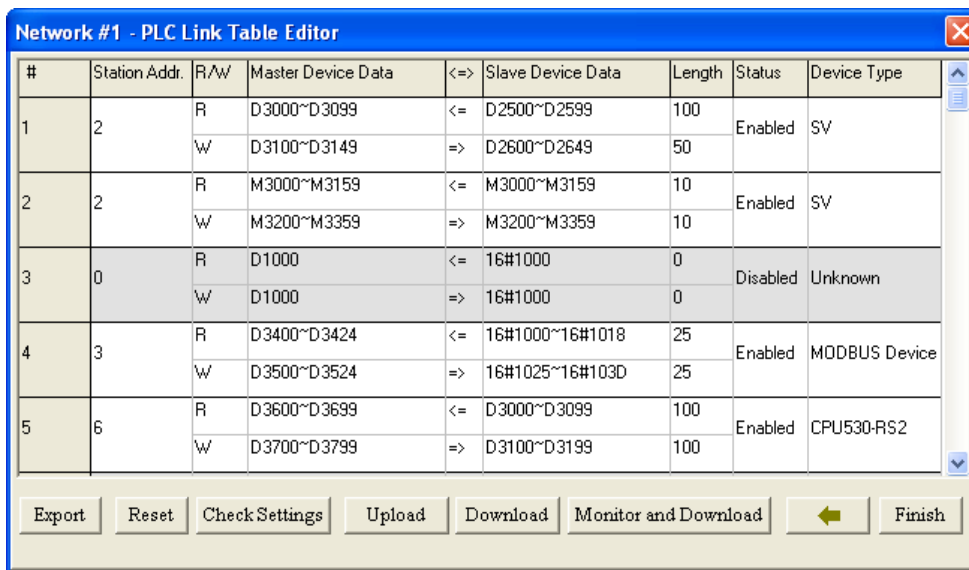
#	Station Addr.	R/W	Master Device Data	<=>	Slave Device Data	Length	Status	Device Type
2	3	R	D100~D115	<=	D100~D115	16	Disabled	SX
		W	D100~D115	=>	D200~D215	16		



When the master station device is DVP series, since the host of DVP series cannot save the device type in data exchange table, it is not possible to check on the PLC type base on the data uploaded. Therefore, the Device Type column shows the name of the PLC type base on the Graphic Workspace. For unreasonable settings, the status is auto-switched to **Disabled**.

### 20.3.4.4 Managing 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.

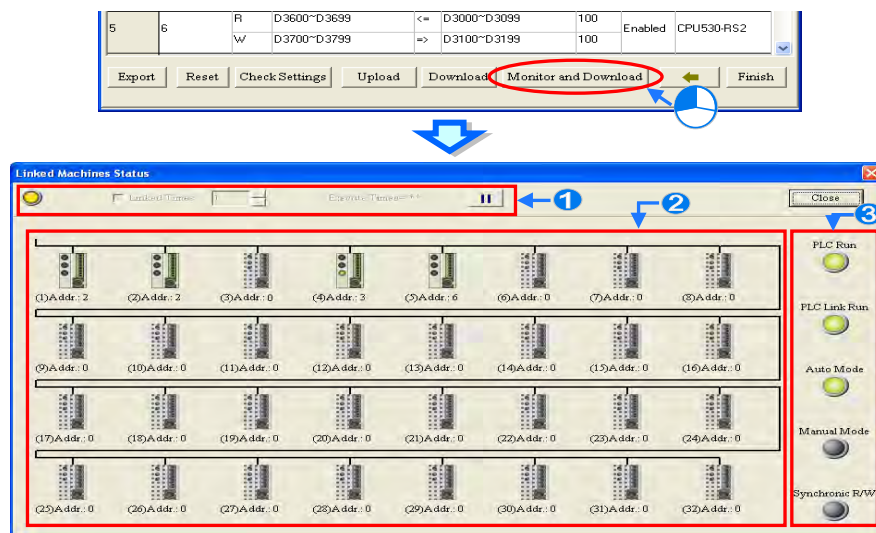


Item	Description
<b>Export</b>	Exports the data exchange table in CSV file format.
<b>Reset</b>	Set the data exchange table content to initial state.
<b>Check Settings</b>	Check the data exchange table to be correct.
<b>Upload</b>	Upload the data exchange table in master station.
<b>Download</b>	Download data exchange table to master station.
<b>Download &amp; Monitor</b>	Download data exchange table to master station and automatically opens monitoring window once downloading is complete.
<b>Finish</b>	The system will ask to save the modifications before closing the window.

Before executing uploading & downloading or monitoring in PLC Link, please make sure the actual device is well-connected.

### 20.3.5 Monitoring a PLC Link

Before monitoring, please make sure the system and host is well-connected. Then, click **Monitor and Download** to open the monitoring window in PLC Link; the host of the master station must be in RUN state, so that PLC Link operation can execute.

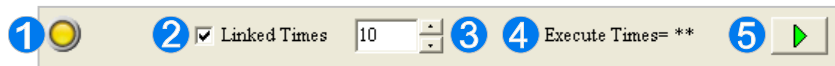


- ① **Function Operation area:** To operate PLC Link data exchange function.
- ② **Monitoring area:** Displays the read or write status of master to slave station and execute online data read or write in this area.
- ③ **Lighting Status area:** The lighting indicates current PLC Link status. Please refer to the following table for each lighting description.

Lighting	Description
<b>PLC Run</b>	When lighting is ON means PLC host is RUN.
<b>PLC Link Run</b>	When lighting is ON means PLC Link enabled.
<b>Auto Mode</b>	When lighting is ON means PLC Link execution in Auto Mode.
<b>Manual mode</b>	When lighting is ON means PLC Link execution in Manual Mode. When executing number reaches the polling number, the lighting is OFF.
<b>Synchronic R/W</b>	When lighting is ON means Synchronic R/W enabled.

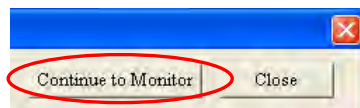
● **Setting area**

To execute setting or operating PLC Link data exchange function, 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. When leaving PLC Link monitoring function, the PLC Link will remain in its last operating status, therefore, please make sure the current PLC Link status is correct before leaving the monitoring page.



- 1 When lighting is ON means the computer and PLC host remains connected.
- 2 Select the item and the setting bases on 3 polling number to execute. Maximum number is 65535.
- 4 When 2 is selected and execution starts, the number of Execution Times is shown; the calculation is to complete one round of data read and write in slave station.
- 5 To start or stop monitoring.

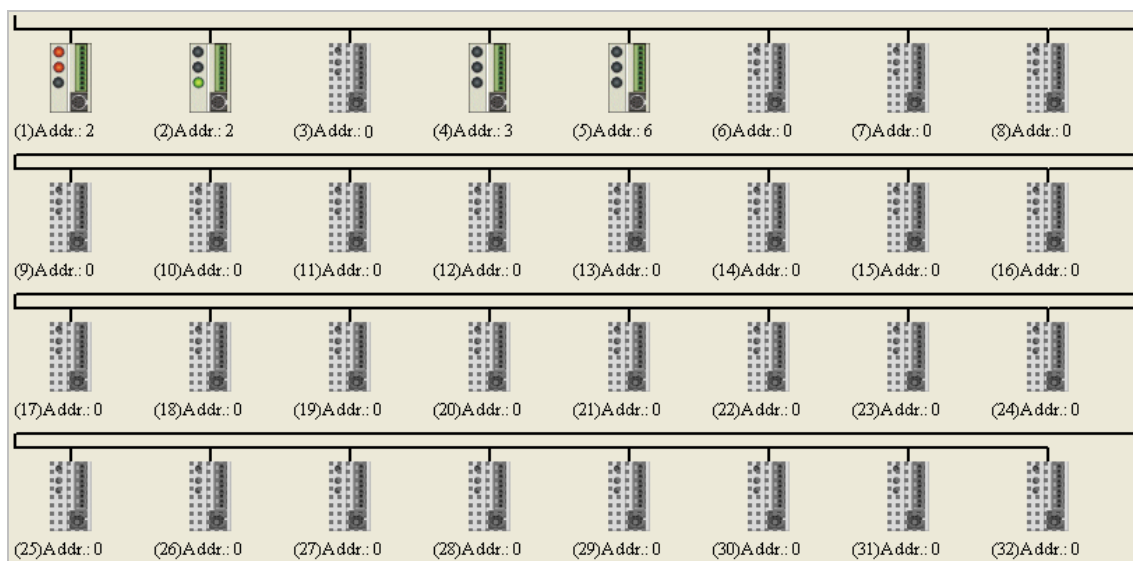
When disconnection occurs during monitoring process, **Continue to Monitor** button appears on the right upper corner of the window; when resolved, click the button to restart the connection.



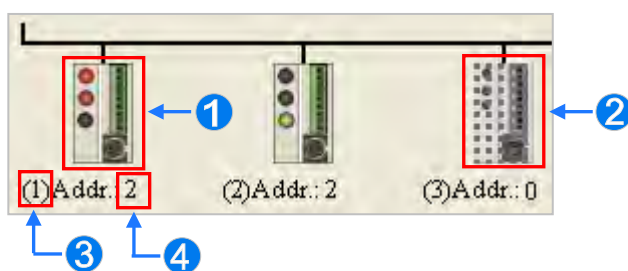
● **Monitoring area**

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.

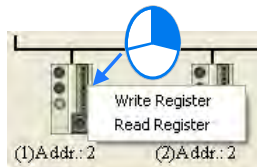


- ① The indicators on the small picture indicate the status of the data exchange group represented by the small picture.
- ② The data exchange group represented by the small picture is not involved in data exchange.
- ③ The number corresponds to a serial number in the data exchange table. It represents 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.

Since PLC Link error flags for AH5x0 series and DVP series differs slightly, therefore it is displayed differently. Please refer to the following table.

Master station PLC Type	Small picture	Description
AH5x0 series		<p>① When lighting is ON means read slave station data is in error.</p> <p>② When lighting is ON means writing data in slave station data is in error.</p> <p>③ When lighting is ON means data is exchanging.</p>
DVP series		<p>① When lighting is ON means read or write slave station data is in error.</p> <p>② When lighting is ON means data is exchanging.</p>

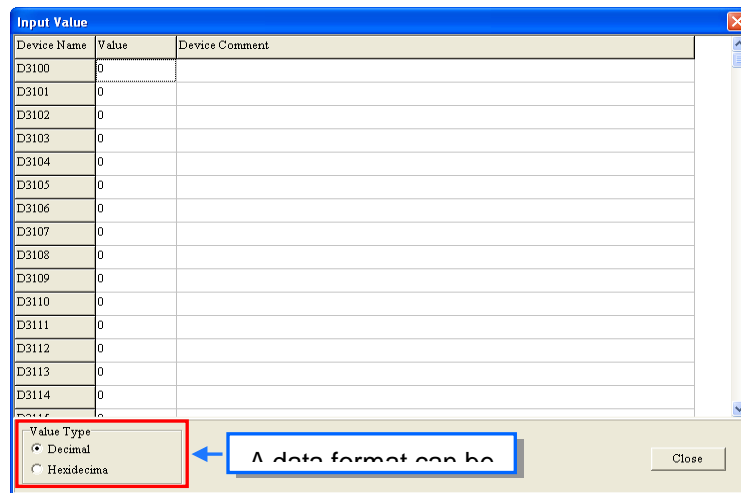
After the users select a small picture, and right-click the small picture, they can click **Write Register** or **Read Register** on the context menu.



- **Write Register:** To write in the device position data of slave to master station, for example D3100~D3149 (see image below).
- **Read Register:** To read the device position data of slave from master station, for example D3000~D3099 (see image below).

#	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 corresponding 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.

Input Value	Device Name	Value	Device Comment
	D3100	0	
	D3101	0	

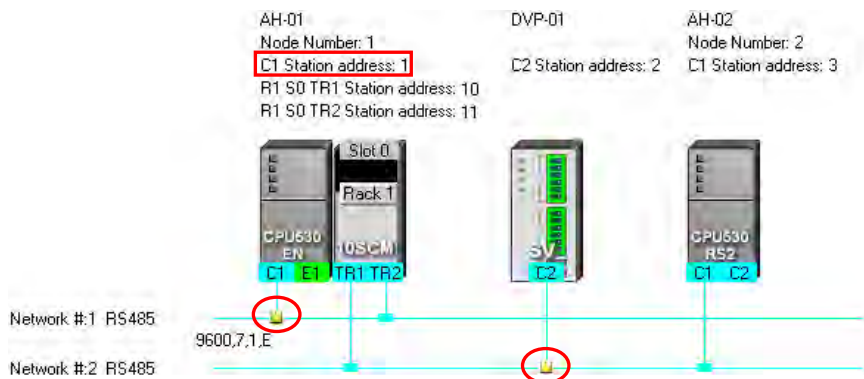
OR

Input Value	Device Name	Value	Device Comment
	D3100	0	
	D3101		

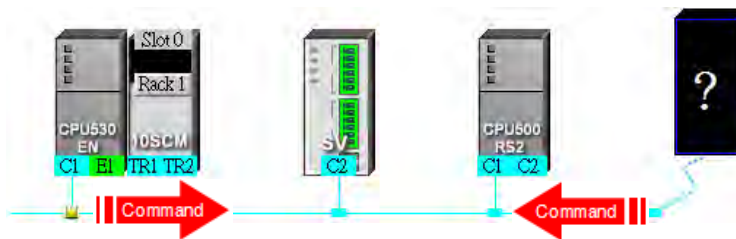
- ① Users can type a value in the box.
- ② Users can select a data format.
- ③ Users can display or hide the binary mode.
- ④ In the binary mode, users can set the states of the bits through the mouse.

### 20.3.6 Notifications on PLC Link

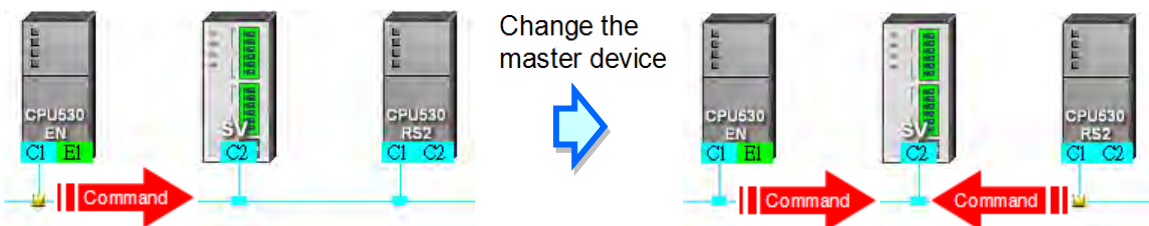
When planning PLC Link, the same RS-485 network can only have one master station. When the nodes set as the master station has many COM ports, the nodes cannot be used as master station in other RS-485 network, but only as slave station. For example (see below), AH-01 shows that node in communication port C1 is the master station for Network #1 and cannot be the master station assigned to Network #2 but only as slave station. While, C1 and TR1 port in AH-01 are connected to Network#1 and both port station addresses are independent, therefore, TR2 assigned as slave station is allowed for Network#1 in PLC Link planning.



One network can only have one master station to send command at the same time. Therefore, when finish network planning, please make sure the actual network is the same as in NWCONFIG to avoid similar error (see below).



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.

## 20.4 Constructing an Ether Link

### 20.4.1 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 PLC,

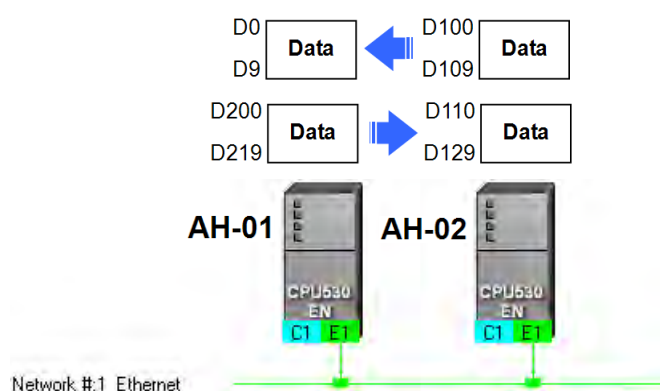
the systems of the PLCs perform data exchange according to the start mode selected when the PLCs run. The users do not have to write a redundant program.

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 cannot 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.

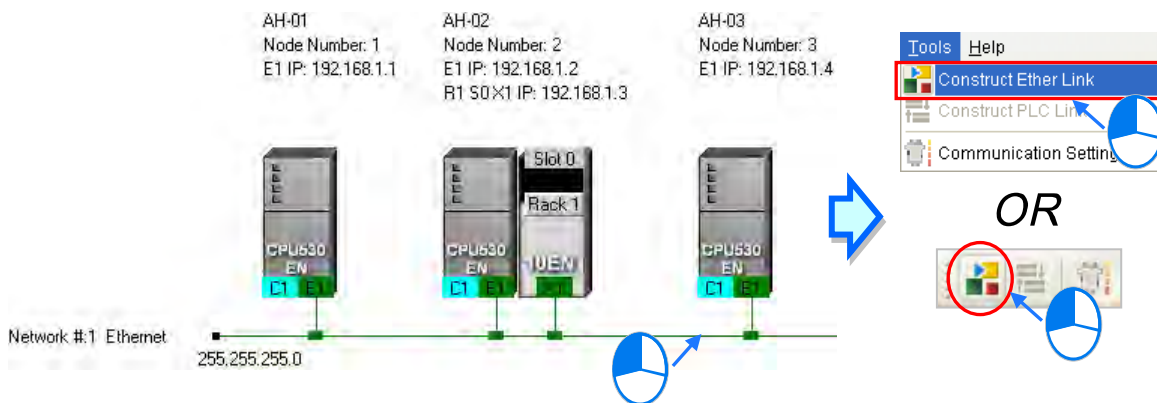
### 20.4.2 Open Ether Link Configuration

To 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**

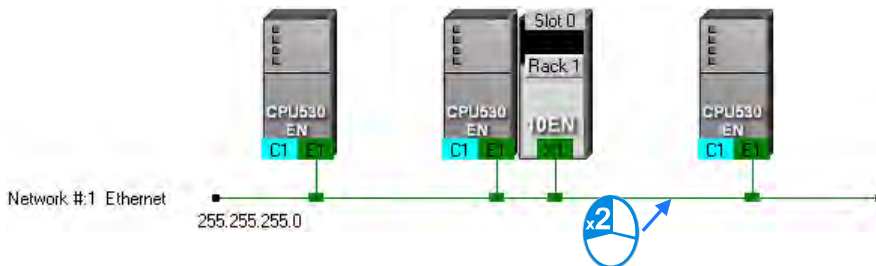
After the users select a network, they have to click **Construct Ether Link** on the **Tools** menu, or

 on the toolbar.



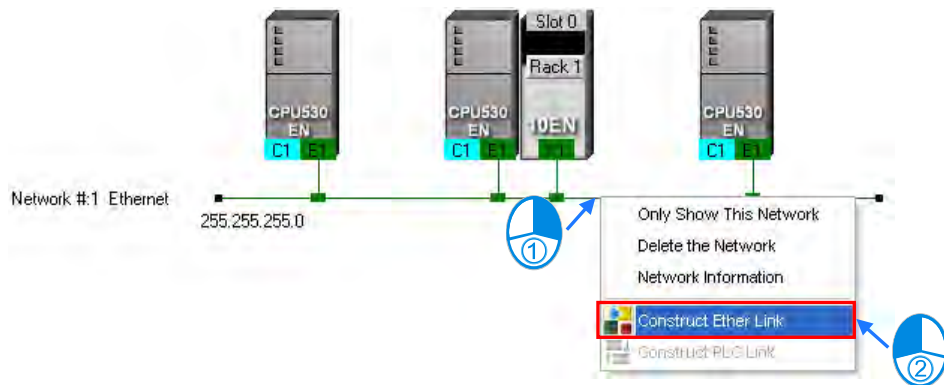
- **Method 2**

Double-click a network.

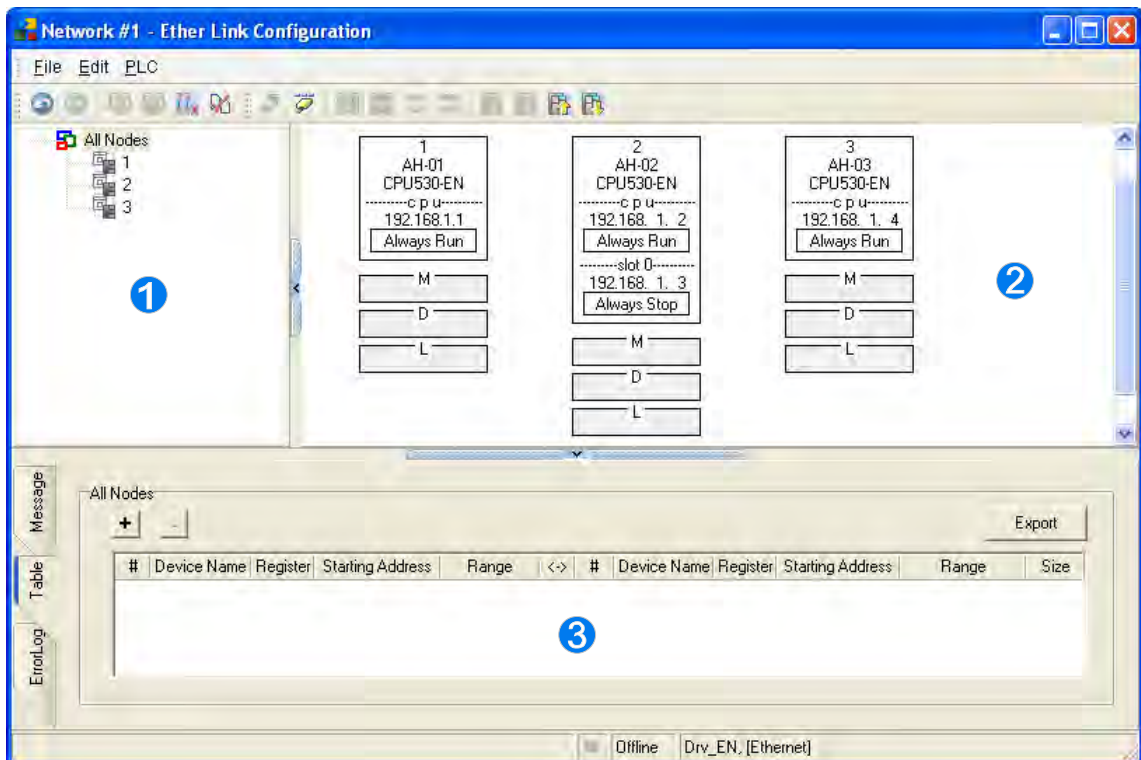


● **Method 3**

Right-click a network, and then click **Construct Ether Link** on the context menu.

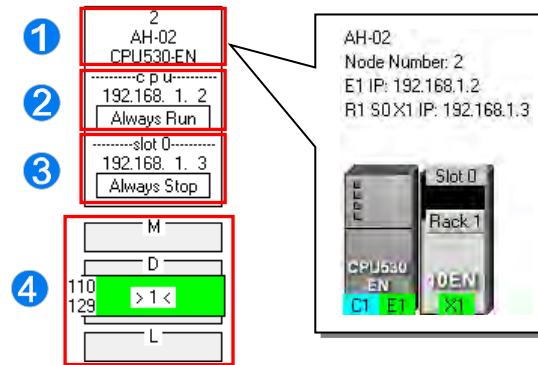


The **Ether Link Configuration** window is shown below.



- ❶ **Node list:** Displays the node list.
- ❷ **Graphic area:** Use graphics to demonstrate node settings.
- ❸ **Information window:** Display three pages each contains information on data exchange table, message and device error logs.

A node shown in the display area in the figure above is described below.




- 1 The information is composed of a node number, a PLC name, and a model name.
- 2 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**.
- 3 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**.
- 4 >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.

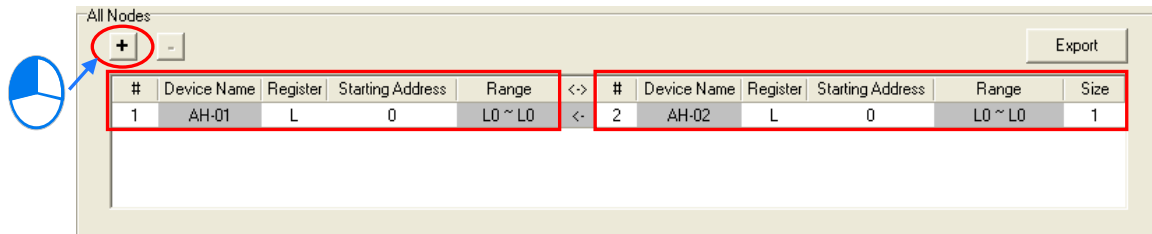
### 20.4.3 Create and Manage Data Exchange Table

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.



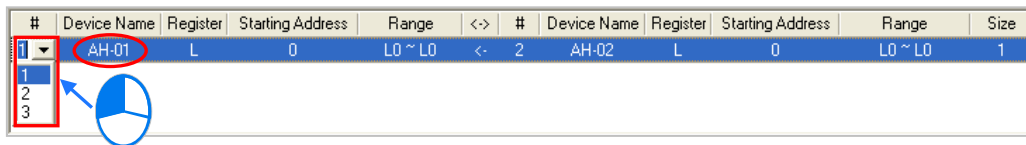


If the users click , a new piece of data will be added to the table. A piece of data is composed of two parts. The left part of the data in the figure below indicates that the data demanding node AH-01 will store the data demanded in L0 in itself, and the right part of the data in the figure below indicates that the data supply node AH-02 will supply the data in L0 in itself.

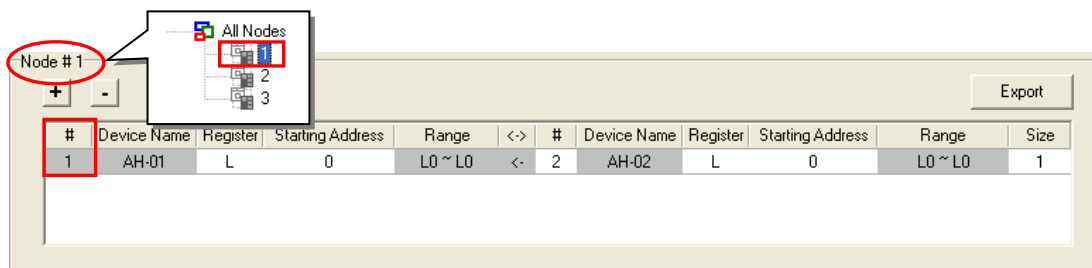


The steps of setting a data exchange group are as follows.

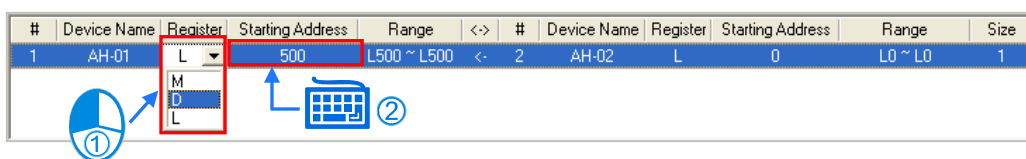
- (1) Select a node number in the # cell for the data demanding node. After the users select a node number, the PLC name corresponds to the node number will be appear in the **Device Name** cell for the node number.



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 cannot 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.

#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D500	<->	2	AH-02	L	0	L0 ~ L0	1

- (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.


#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D500	<->	2	AH-02	M	1000	M1000 ~ M1015	1

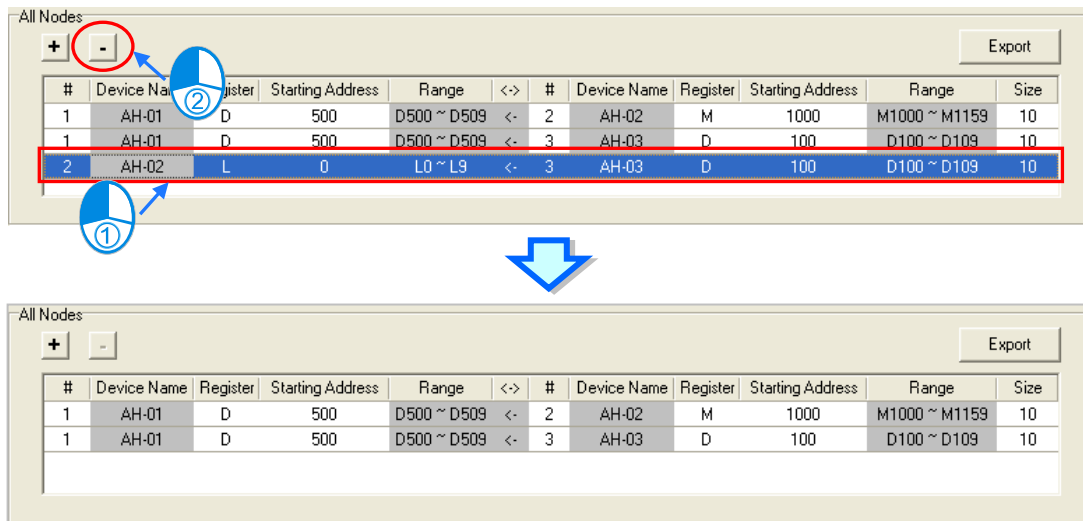
- (5) Type a data length in the **Size** cell. A word is a unit. The maximum data length is 1900 words. After the users type a data length, the device ranges in the **Range** cells will change according to the data length typed.

#	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

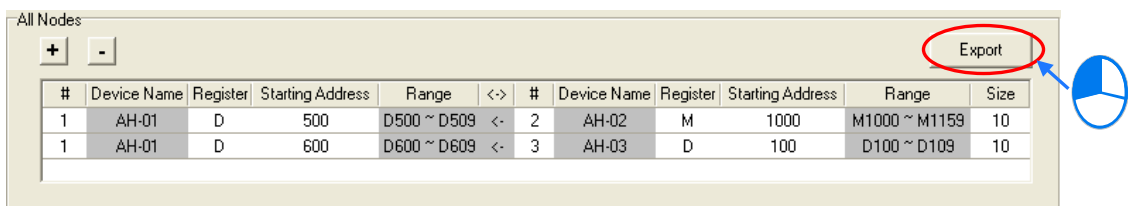
The users can create data exchange groups by following the steps described above. The device range in the **Range** cell for a data demanding node cannot overlap the device range in the **Range** cell for another data demanding node whereas the device range in the **Range** cell for a data supply node 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 cannot 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

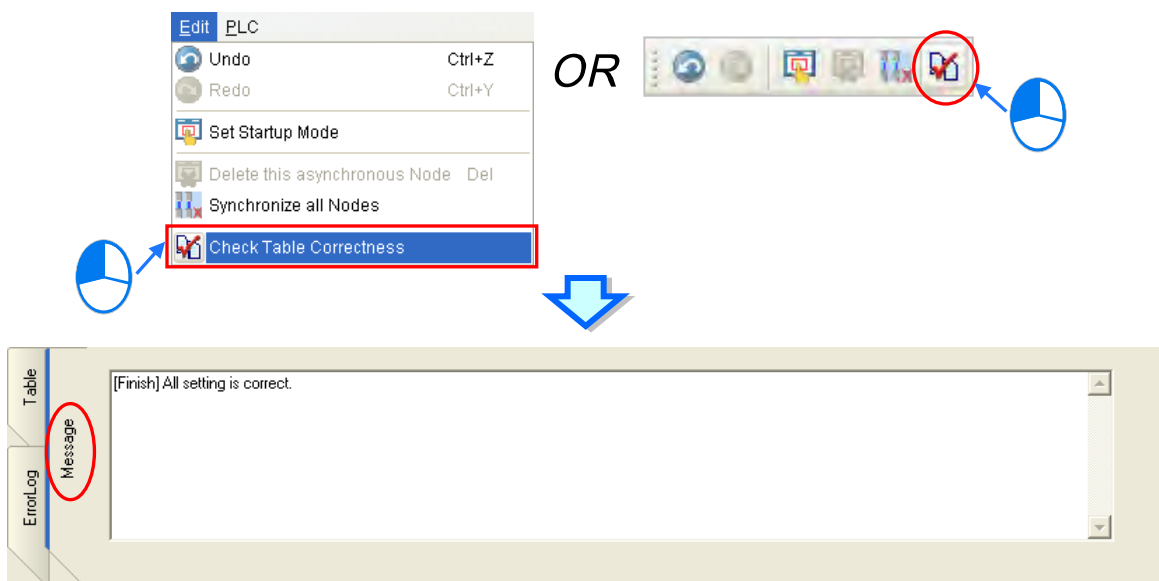
If the users want to delete a piece of data, they can click the piece of data, and click .



After the users click **Export**, the data in the data exchange table can be exported as a CSV file.



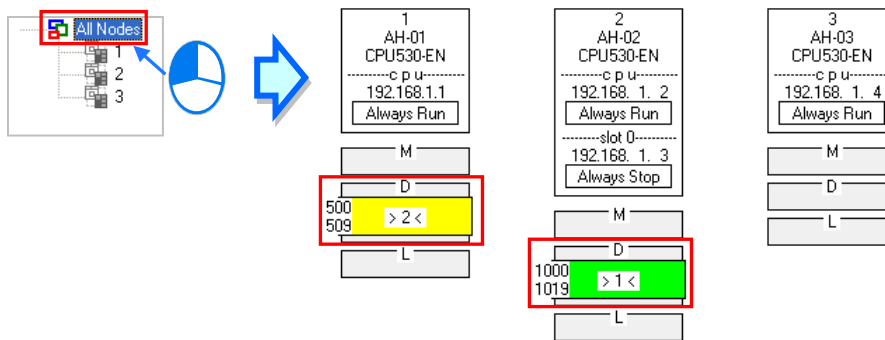
After the users create a data exchange table, they can click **Check Table Correctness** on the **Edit** menu or  on the toolbar if they want to check the data set in the table. The check result will be displayed in the **Message** page.



### 20.4.4 Node List and Display Area

If users click **All Nodes** 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.

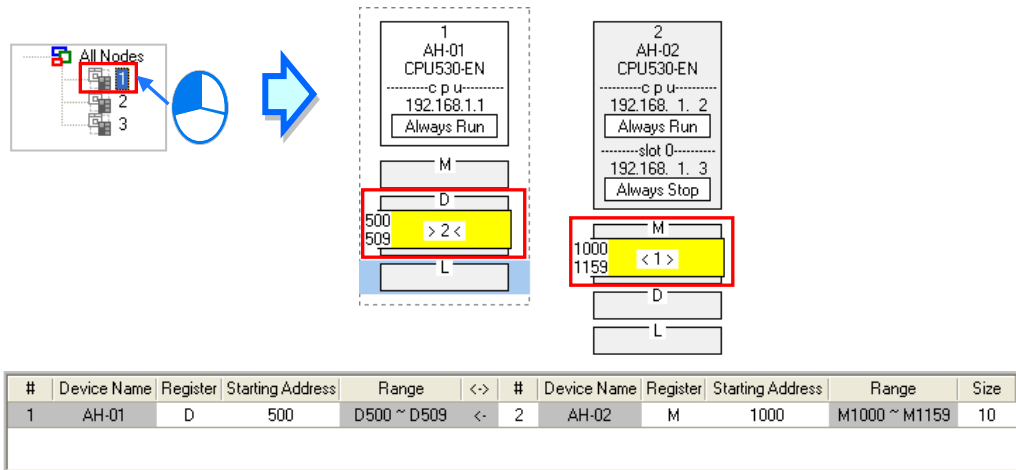


#	Device Name	Register	Starting Address	Range	<->	#	Device Name	Register	Starting Address	Range	Size
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	<-	1	AH-01	L	0	L0 ~ L19	20

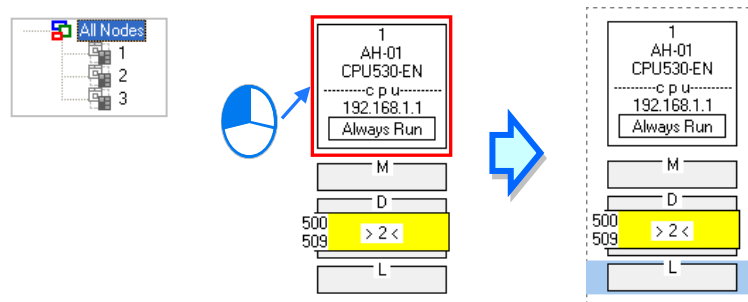
If the users click a specific node on the node list, the node and the nodes of which the node demands data will be shown in the display area, and the data in the data exchange table under the display will be related to the specific node selected.

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 cannot 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.

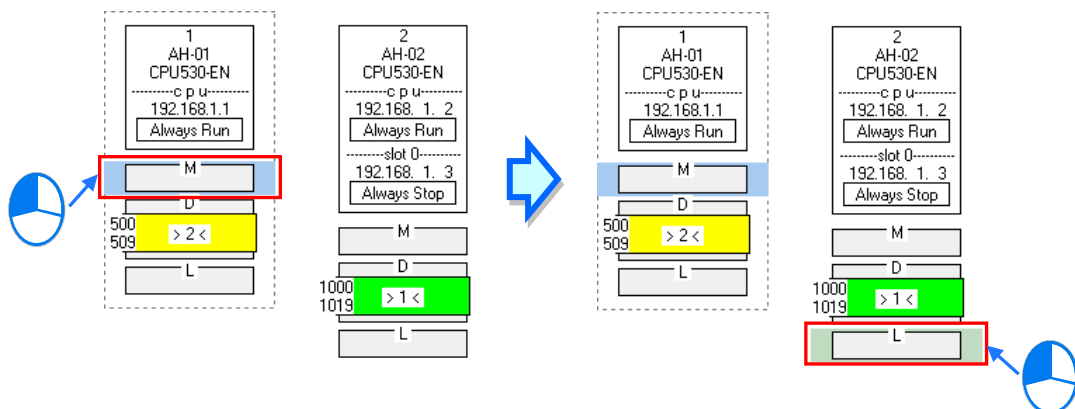
The numbers at the left side of the D block in node 1 indicates that the data demanded of node 2 will be stored in D500~D509 in node 1. <1> in the M block in node 2 indicates that the data in M1000~M1129 in node 2 will be supplied to 1.




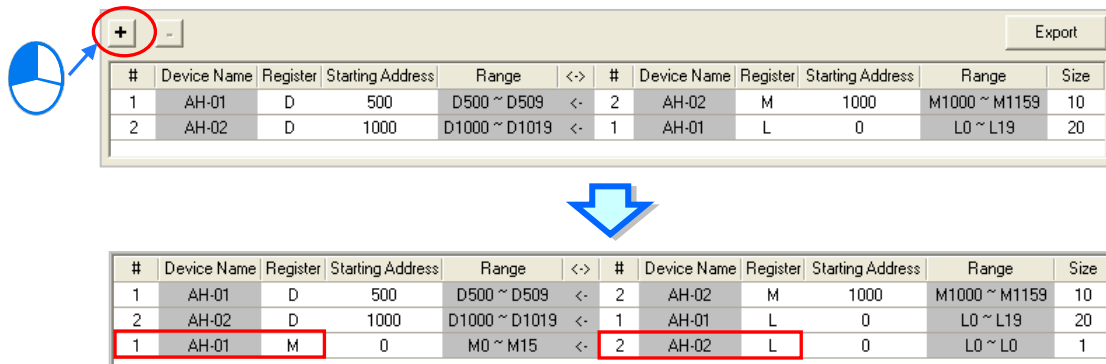
If a specific node on the node list is selected, the node in the display area will be selected. After the users select **All Nodes** on the node list, they can click the information about a node in the display area if they want to select the node. If the users click a device block in a node, the node will not be selected. After a node is selected, a dotted frame will appear.



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 cannot 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.



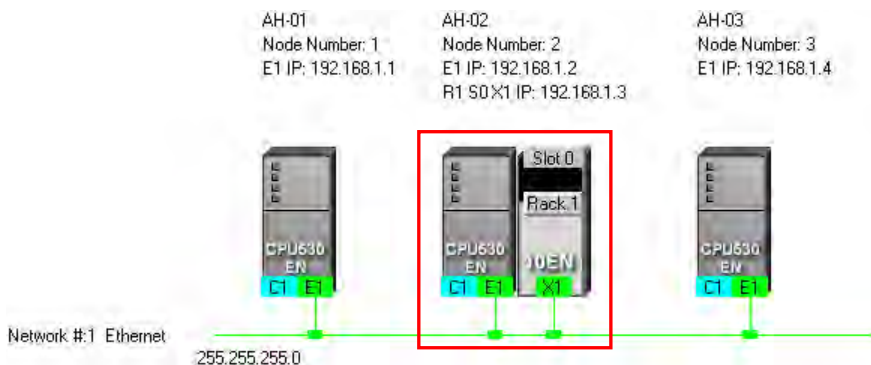
### 20.4.5 Start Mode of an Ether Link

There are three start modes. Please refer to the table below for more information.

Start mode	Description
<b>Always Stop</b>	During the operation of the PLC, no data exchange is performed.
<b>Always Run</b>	When the PLC runs, data exchange is performed.
<b>SM Flag</b>	The performance of data exchange depends on a special relay in the PLC. After users select <b>SM Flag</b> , 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.

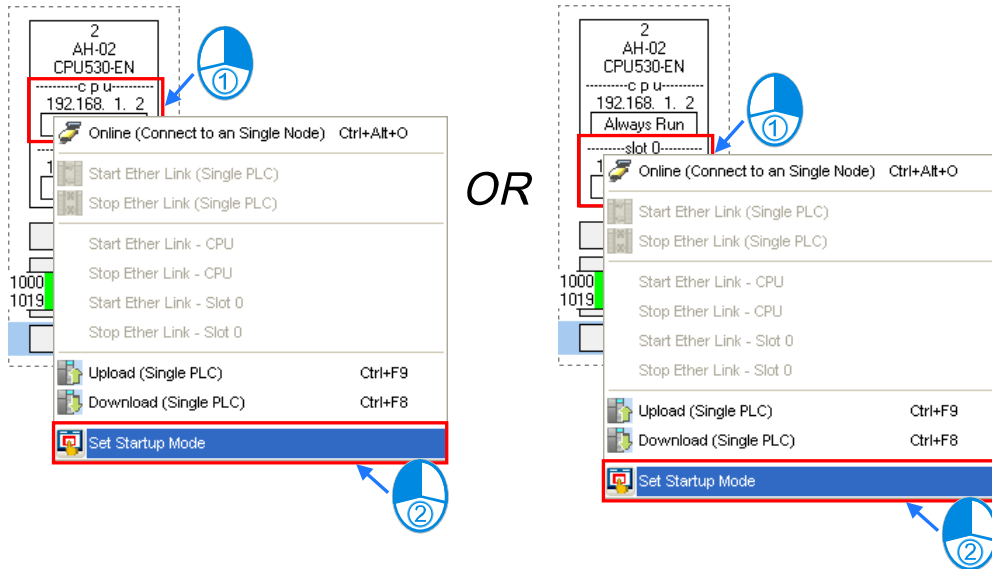
The execution of an Ether Link is based on the nodes in a network. If some of the ports that a node has are connected to a network, users can set the start modes of the ports separately. When the Ether Link constructed is executed, the system automatically distributes reading/writing work to the ports 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.

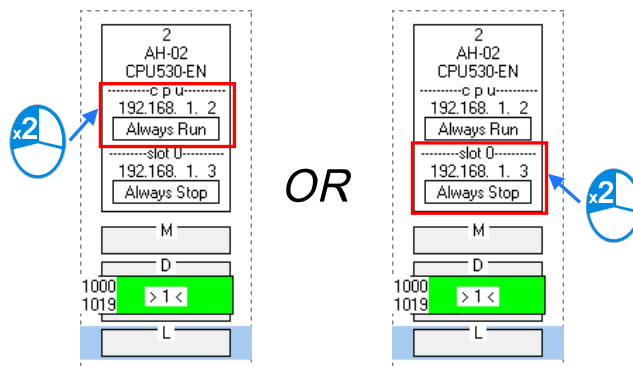
● **Method 1**

Right-click CPU information or module information, and then click **Set Startup Mode** on the context menu.




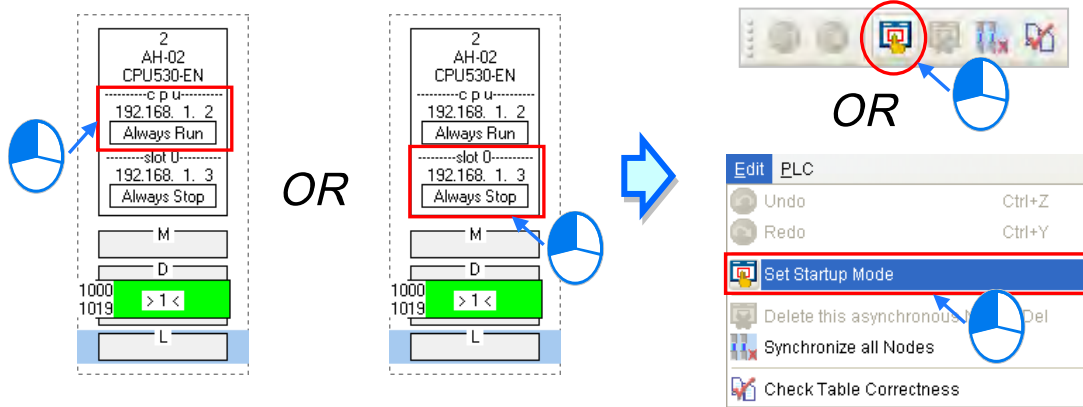
● **Method 2**

Double-click CPU information or module information.



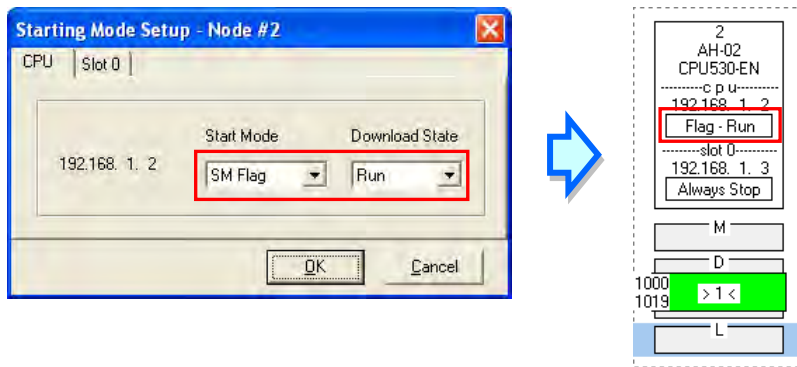
● **Method 3**

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**.



### 20.4.6 Download Ether Link Configuration

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.


20

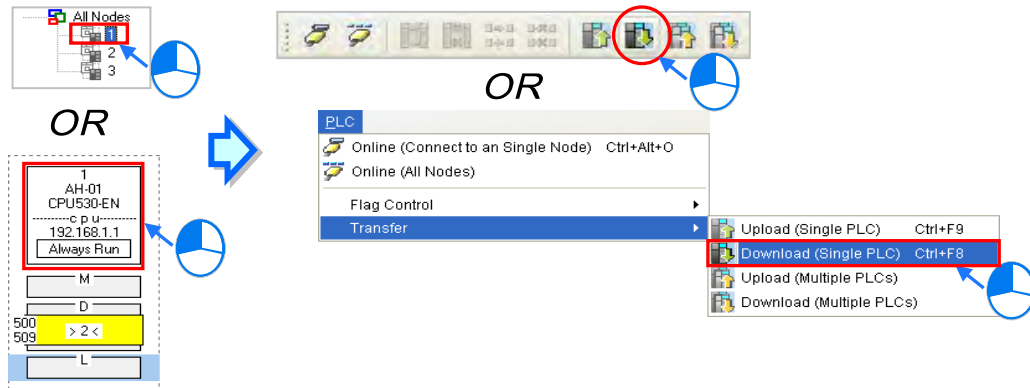
● **Single node**

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 downloading the related parameters, please make sure the system is well-connected to hosts or modules.



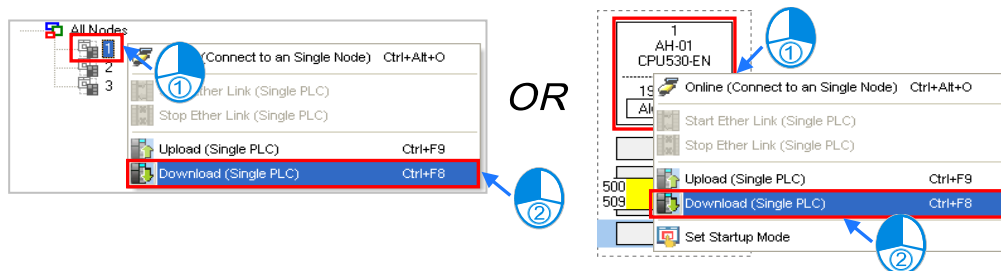
➤ **Method 1**

Select a data demanding node to download parameters, 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.

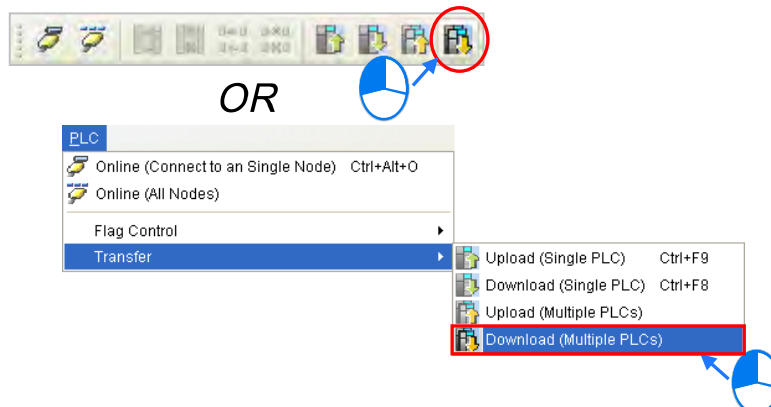


● **Multiple nodes**

Download all settings and enabling mode for data demanding nodes; before downloading, please check the system and hosts or modules are well-connected.

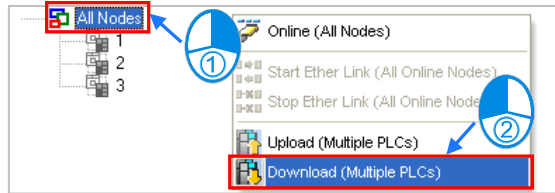
➤ **Method 1**

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.



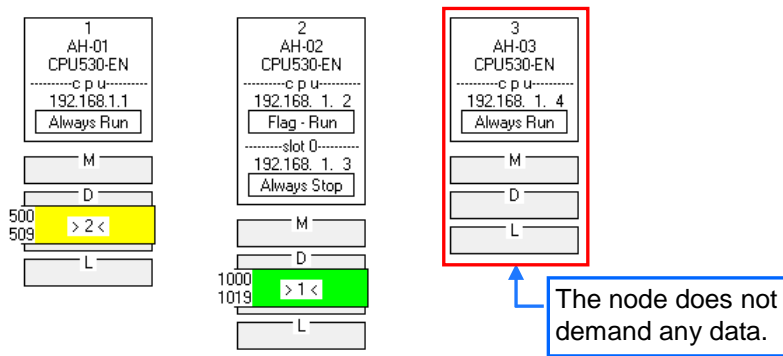
➤ **Method 2**

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.




**20.4.7 Upload Ether Link Configuration**

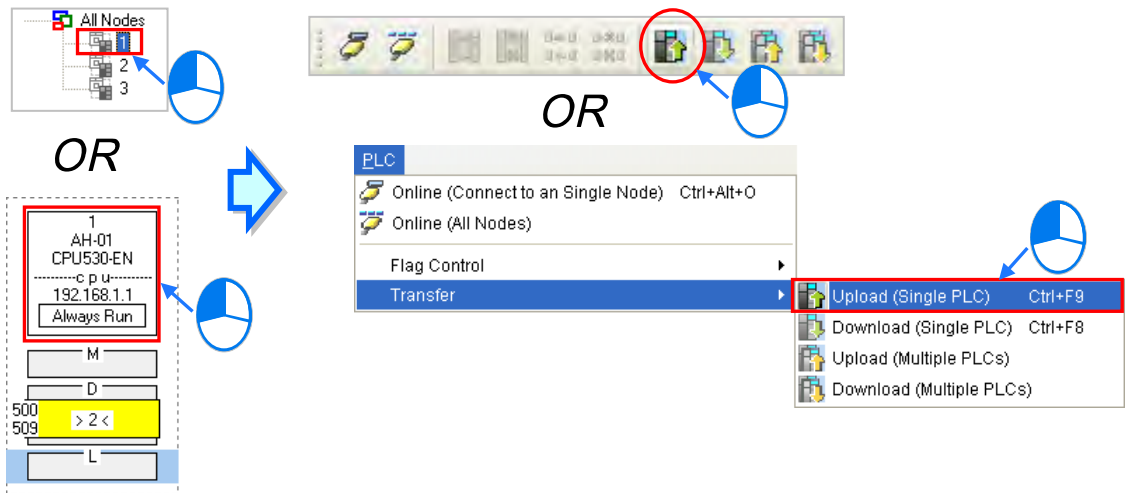
Users can upload the parameters related to an Ether Link in a PLC.

● **Single node**

Upload only the Ether Link setting parameter of single node; before uploading, please check the system and hosts or modules are well-connected.

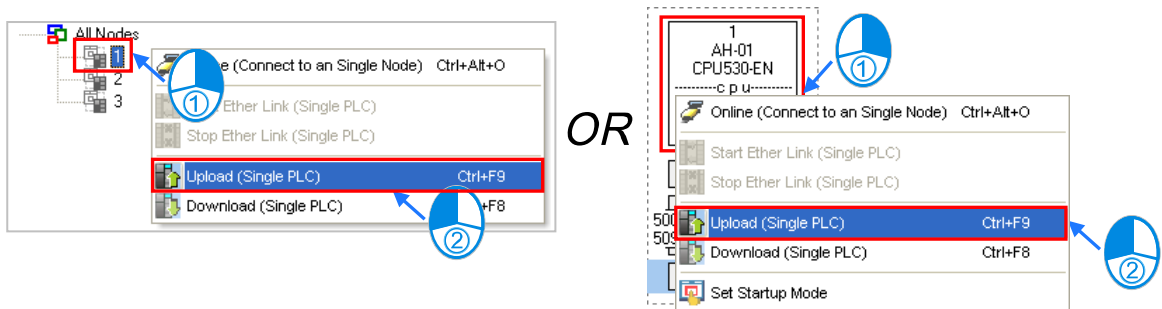
➤ **Method 1**

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.

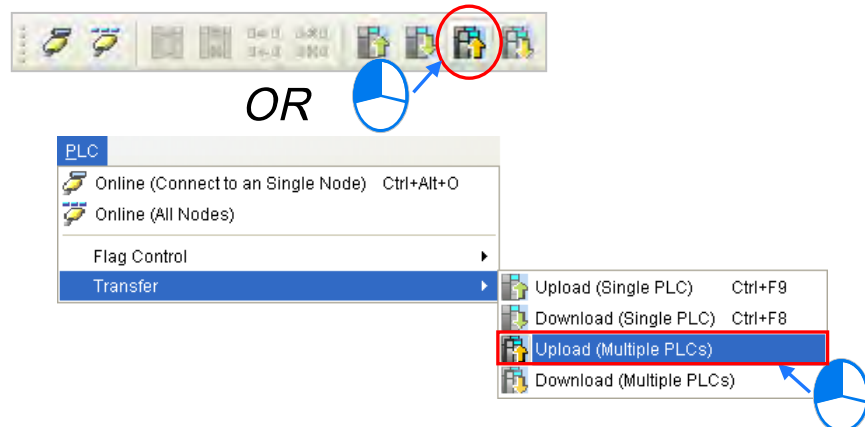


● **Multiple nodes**

Upload all Ether Link setting parameters on nodes; before uploading, please check the system and hosts or modules are well-connected.

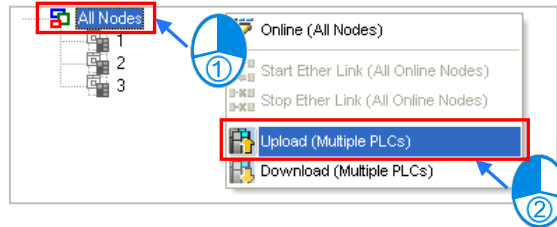
➤ **Method 1**

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.



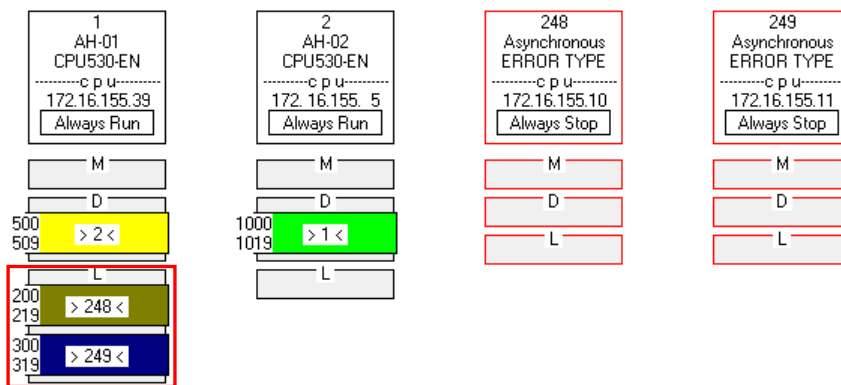
➤ **Method 2**

Select **All Nodes** on the node list, right-click **All Nodes**, and click **Upload (Multiple PLCs)** on the context menu.




**Additional remark**

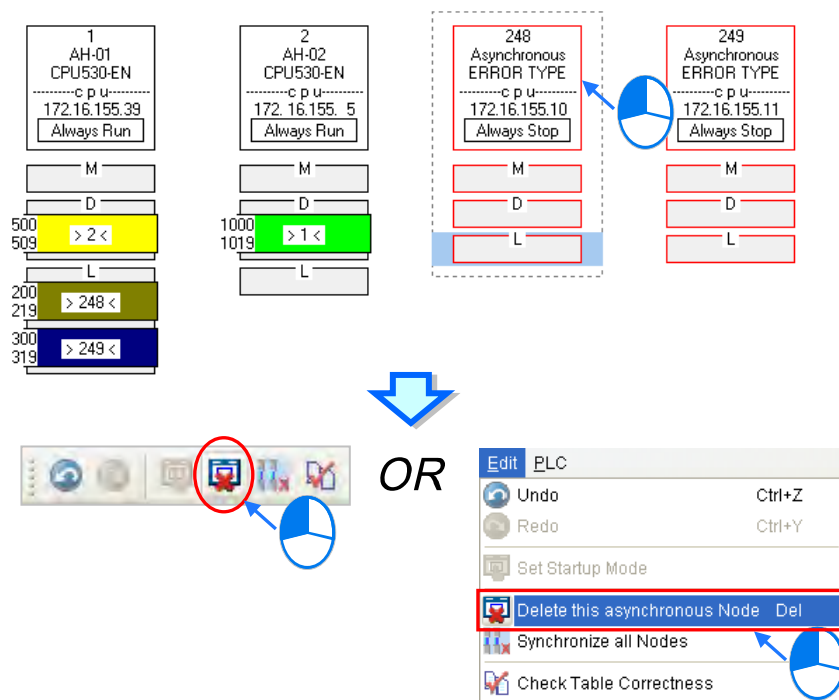
After re-uploading Ether Link parameters, if nodes that are not assigned in NWCONFIG, the node will be framed in red to display as **Asynchronous Device**. When Ether Link setting contains asynchronous device, monitoring or down loading are not allowed.




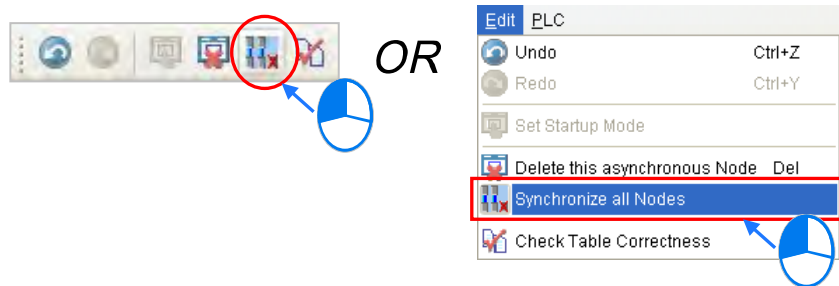
**20.4.8 Deleting Asynchronous Device**

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.



### 20.4.9 Enable/Disable 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 <b>Driver Name</b> drop-down list box in the <b>Select a Driver</b> window uses must be Ethernet.



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.

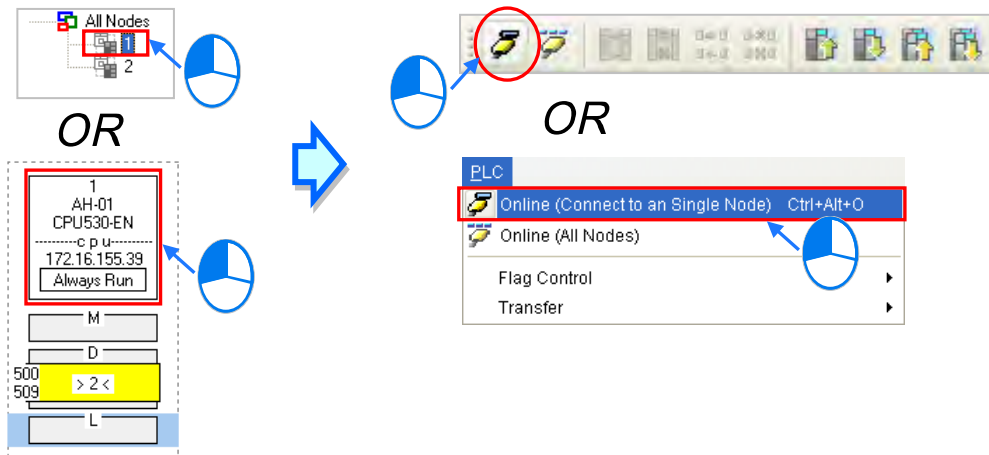
- (a) Every node has been connected to a network according to the network framework created in NWCONIFG.
- (b) The users have set the parameters for Ethernet ports of the nodes by means of HWCOFNIG, and the parameters have been downloaded to the PLCs and the modules. The setting of the parameters must 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.

### 20.4.9.1 Enabling a Monitoring Function


- **Enabling the function of monitoring a single node**

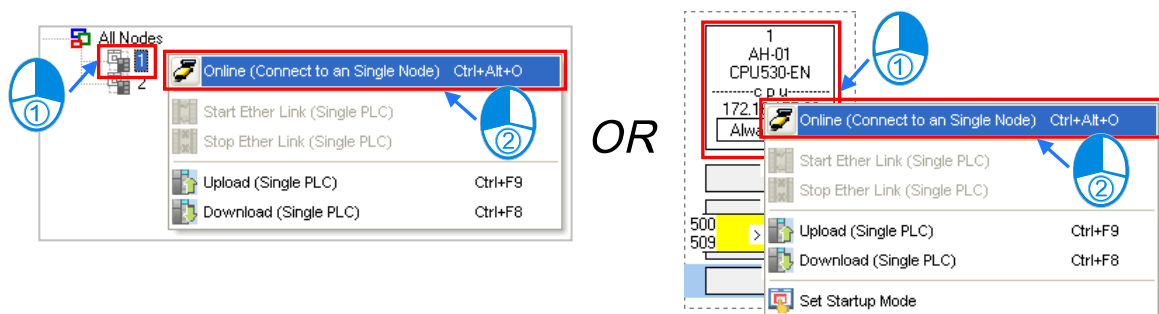
- **Method 1**

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**

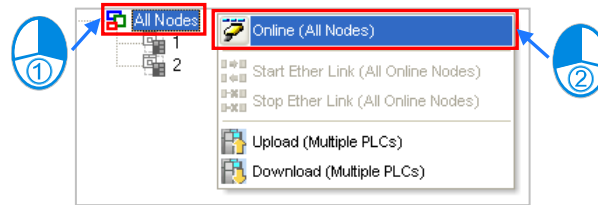
- **Method 1**

Click **Online (All Nodes)** on the **PLC** menu, or  on the toolbar.



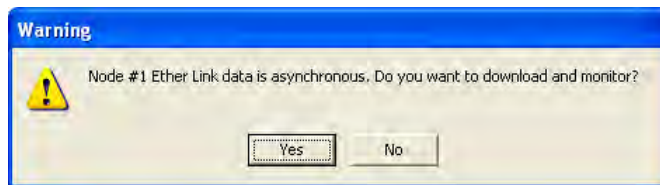
➤ **Method 2**

Select **All Nodes** on the node list, right-click **All Nodes**, and click **Online (All Nodes)** on the context menu.



**Additional remark**

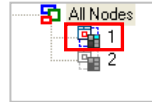
Before the system enters a monitoring mode, it checks whether the Ether Link constructed in the **Ether Link Configuration** window is consistent with the setting in the PLCs. If the Ether Link constructed in the **Ether Link Configuration** window is not consistent with the setting in the PLCs, the system will ask the users to download the related parameters again.



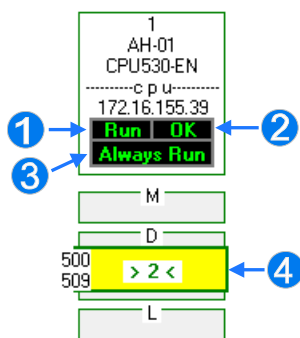


### 20.4.9.2 Monitoring Status

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.



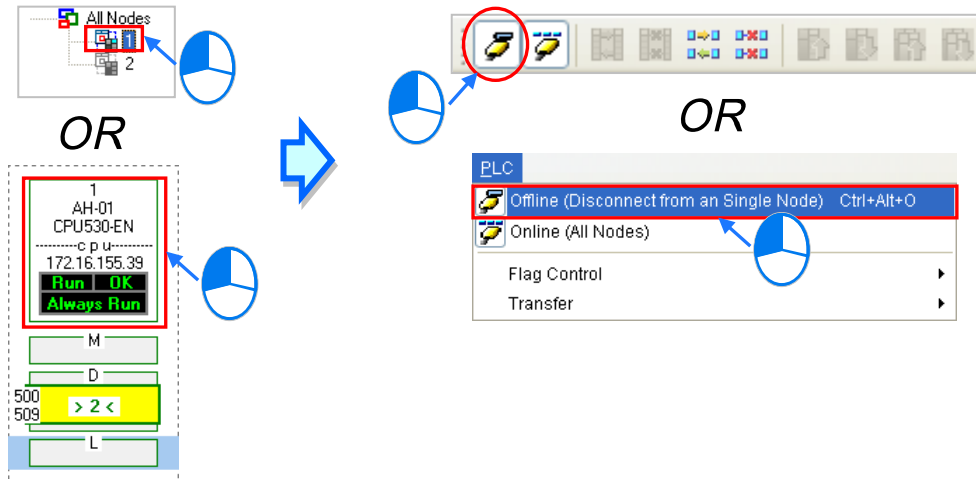
Status	Description
① Running/Stopping	<b>Run</b> : The Ether Link constructed is executed.
	<b>Stop</b> : The Ether Link is not executed.
② Operating status	<b>OK</b> : The Ether Link constructed is executed normally.
	<b>Error</b> : The Ether Link constructed is not executed normally.
③ Start mode	<b>Always Stop</b> : Always Stop
	<b>Always Run</b> : Always Run
	<b>Flag Mode</b> : SM Flag
④ Data block	500 509 <b>&gt; 2 &lt;</b> : The data exchange is being performed.
	500 509 <b>≠ 2 ≠</b> : The performance of the data exchange stops.

### 20.4.9.3 Disabling a Monitoring Function

- Disabling the function of monitoring a single node

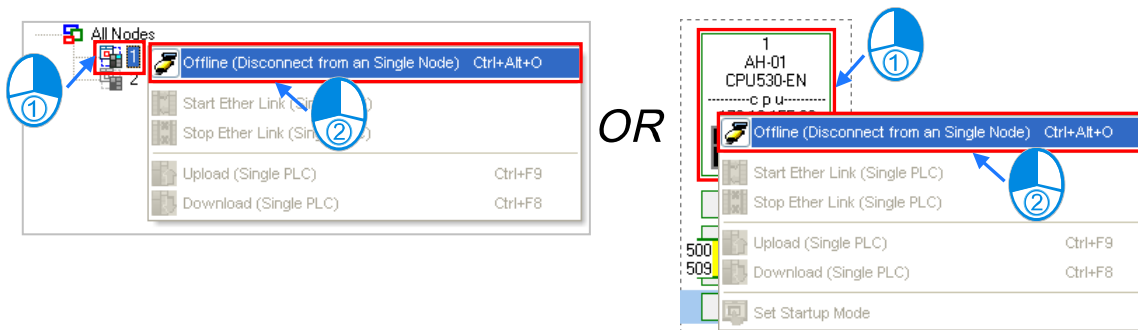
- Method 1



Select a data demanding node, and then click  on the toolbar, or **Offline (Disconnect from a Single node)** on the **PLC** menu.

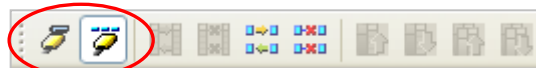


- Method 2

Select a data demanding node, right-click the data demanding node, and click **Offline (Disconnect from a Single Node)** on the context menu.



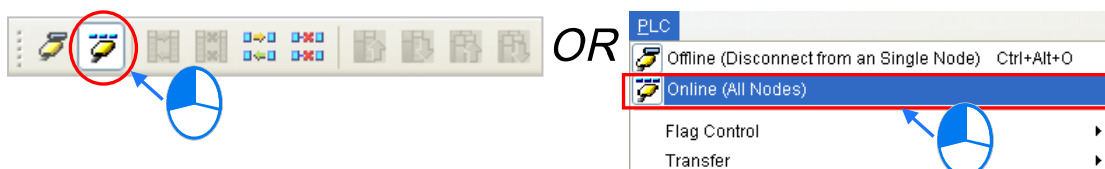
When the data demanding node selected is not monitored,  on the toolbar is not pressed. If another node is monitored,  will be pressed.



- Disabling the function of monitoring multiple nodes

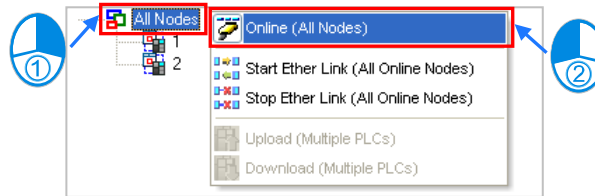
- Method 1

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.




### 20.4.10 Online Start/Stop Ether Link (SM Flag)

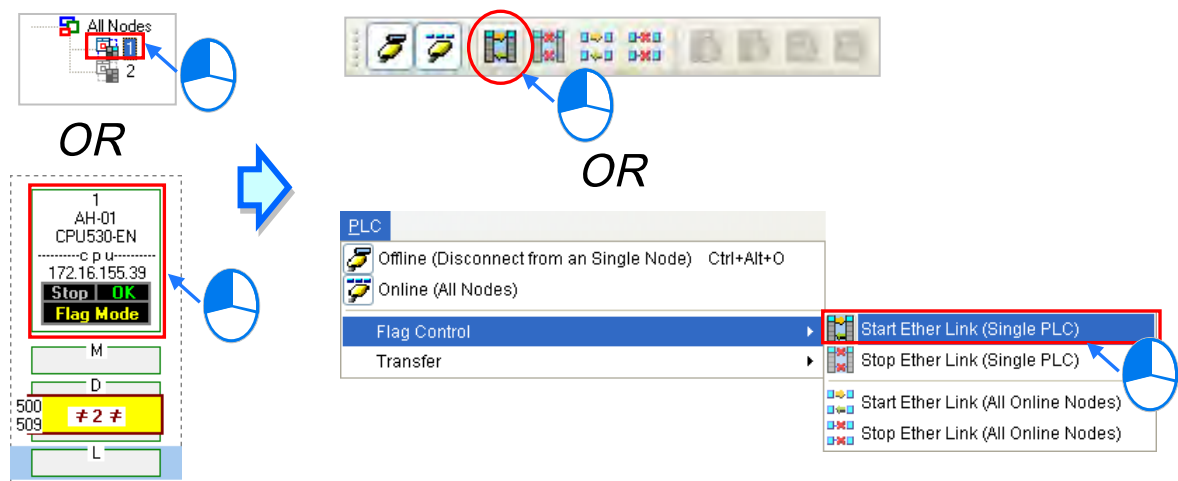
When the start mode of a node is in **SM Flag**, users can directly operate the flags under online monitoring state and start or stop a node under Ether Link operation; however, it is limited to nodes in monitoring.

#### 20.4.10.1 Starting the Execution of an Ether Link

- **Making a single node start the execution of an Ether Link**

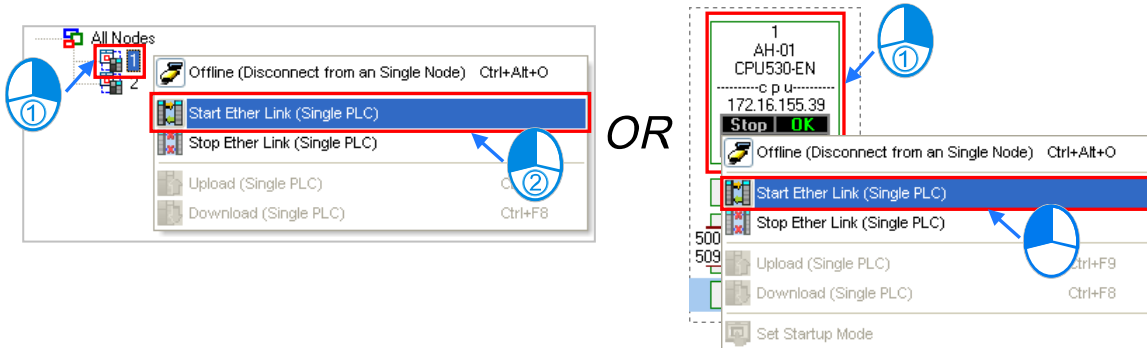
➤ **Method 1**

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**

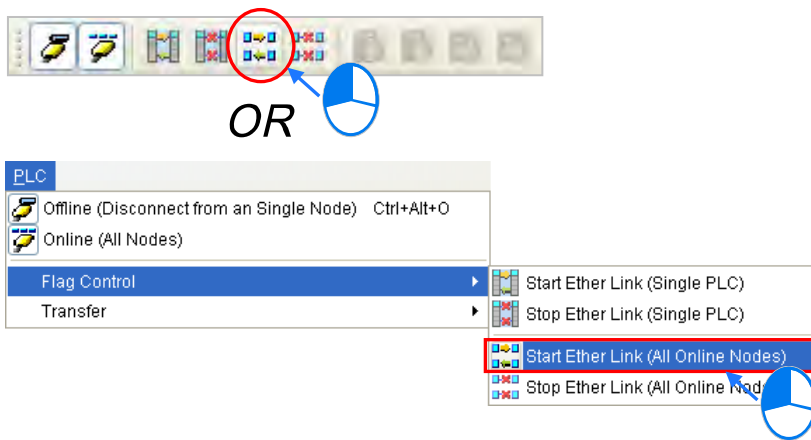
Select a node, right-click the node, and click **Start Ether Link (Single PLC)** on the context menu.



● **Making multiple nodes start the execution of an Ether Link**

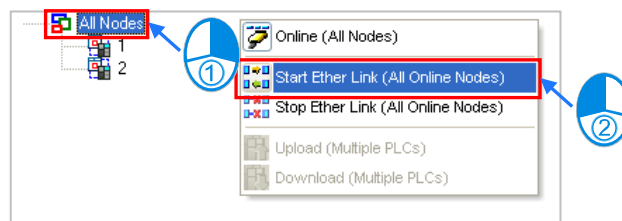
➤ **Method 1**

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.

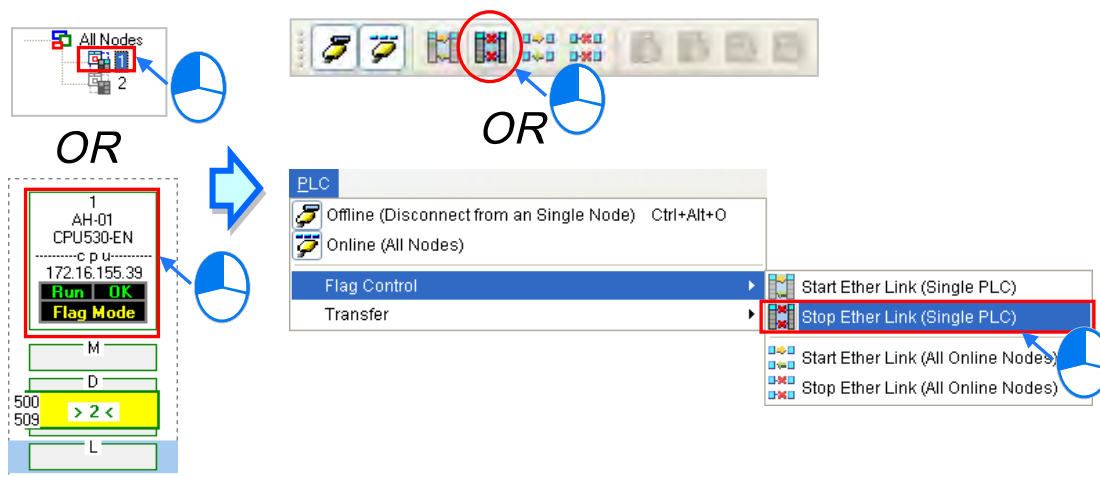


### 20.4.10.2 Stopping the Execution of an Ether Link

- **Making a single node stop the execution of an Ether Link**

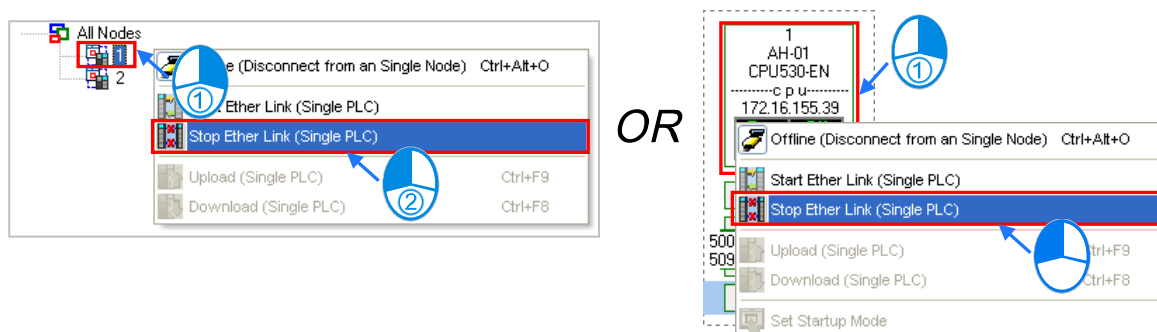
- **Method 1**

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**

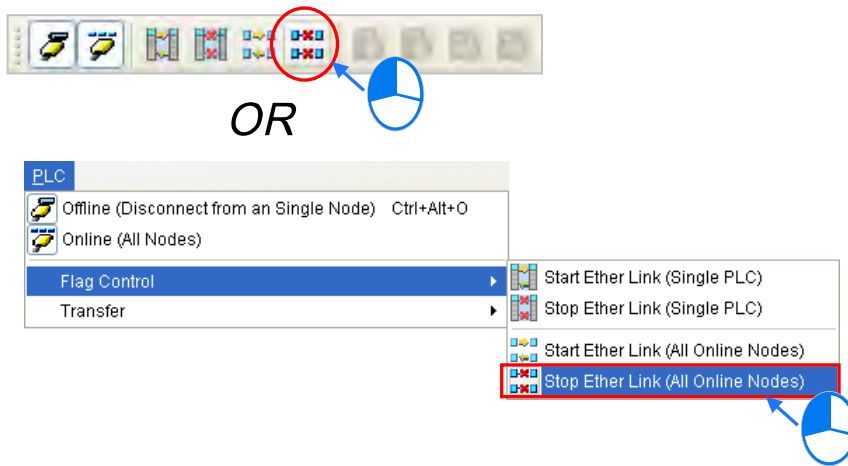
Select a node, right-click the node, and click **Stop Ether Link (Single PLC)** on the context menu.



- **Making multiple nodes stop the execution of an Ether Link**

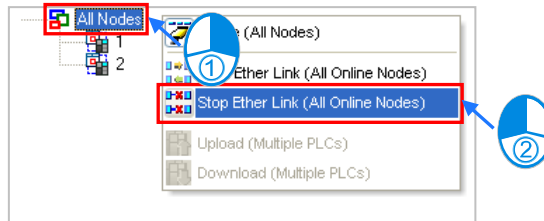
- **Method 1**

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.



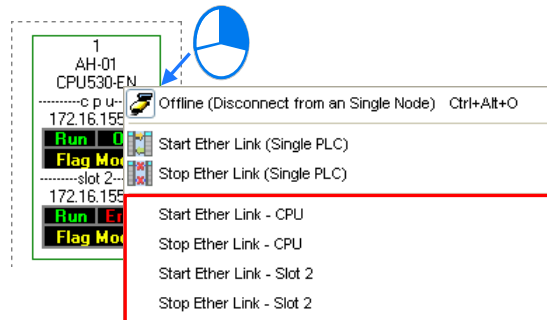
➤ **Method 2**

Select **All Nodes** on the node list, right-click **All Nodes**, and click **Stop Ether Link (All Online Nodes)** on the context menu.



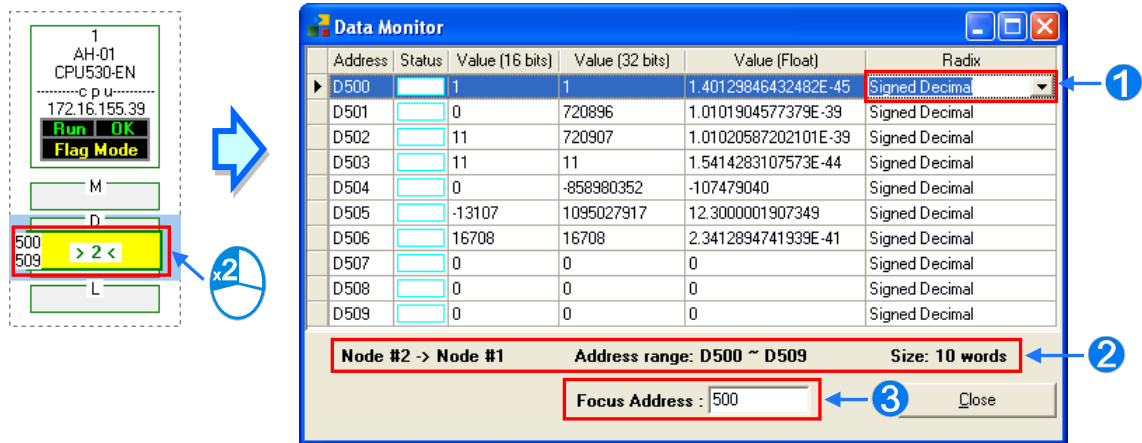
**Additional remark**

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.



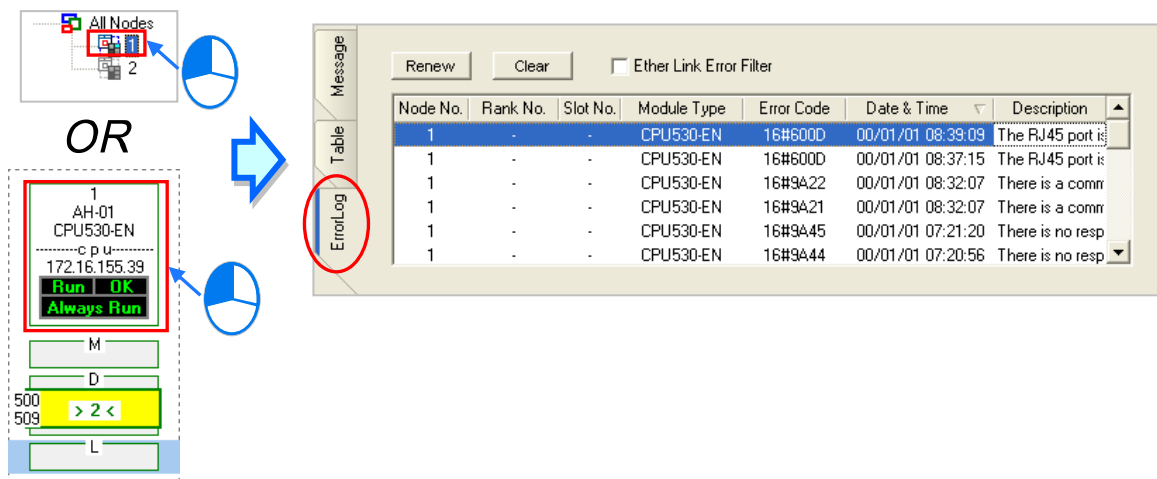
### 20.4.11 Monitoring Table and Error Log

After users double-click a data block, the **Data Monitor** window will appear. The users can view the values in the window. They cannot change the values in the window.



- 1 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.
- 2 The information about the data block which is monitored is displayed.
- 3 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.




The **Error Log** page contains error records from hosts and modules. When an error log is formed by modules, the PLC model type is known in the **Device Type** column and the module position can be found in **Rack No.** and **Slot No.** columns. However, when error occurs by CPU, no numbers are shown in **Rack No.** and **Slot No.** columns.

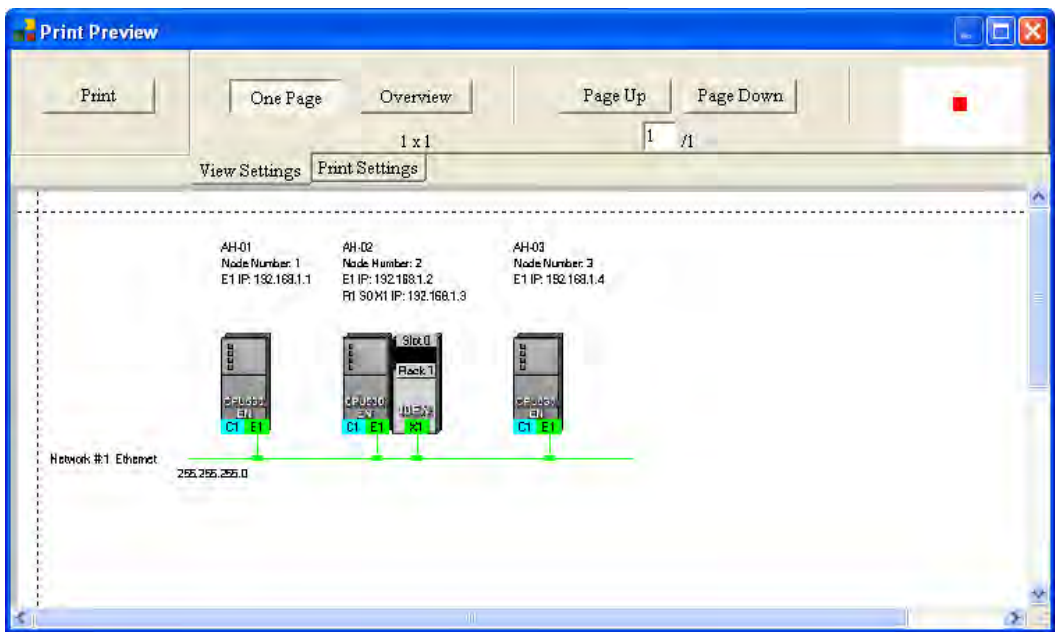
## 20.5 NWCONFIG Management and Application

### 20.5.1 Save and Print

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.



20

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.



## 20.5.2 Downloading

In NWCONFIG Graphic Workspace, 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.

### 20.5.2.1 Downloading 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**

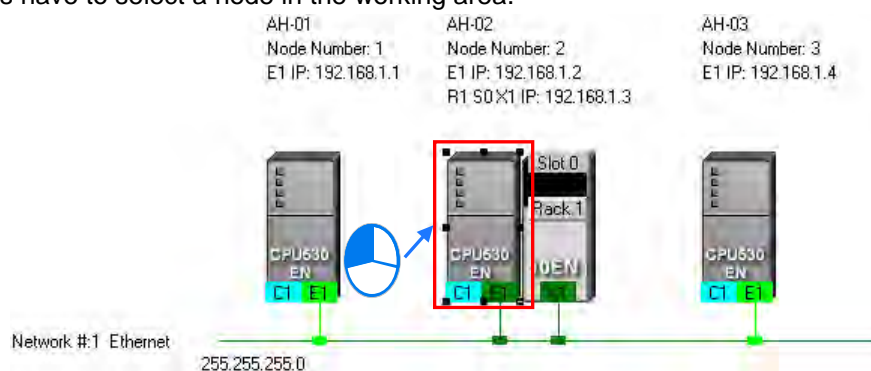
The parameters related to an Ether Link must be downloaded to the data demanding nodes. However, the data exchange table stored in a PLC is a table related to the PLC itself. Besides, if the parameters set include a node which does not demand any data, the data in the node will be cleared after the parameter 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.


### 20.5.2.2 Description of Downloading

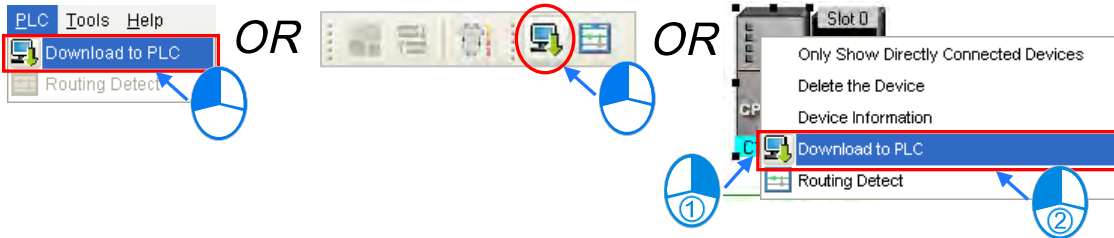
There are two ways to download. One is **Single node**, the other is **Multiple node**. When downloading communication settings, do not use **Routing Mode**.

- **Single node**


The users have to select a node in the working area.

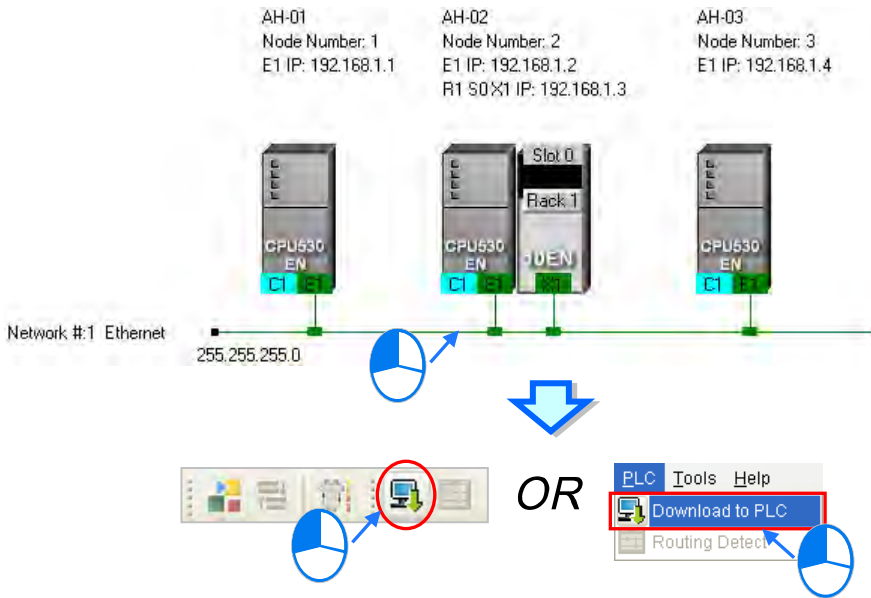


After the users click **Download to PLC** on the **PLC** menu, or  on the toolbar, the **Optional Download** window will appear. The users can also open the **Optional Download** window by right-clicking the device they select, and clicking **Download to PLC** on the context menu.

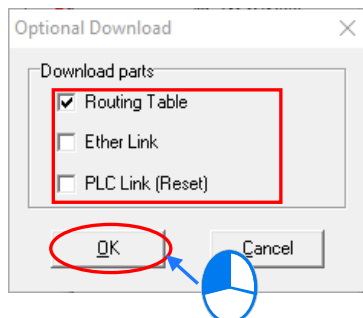


● **Multiple nodes**

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.



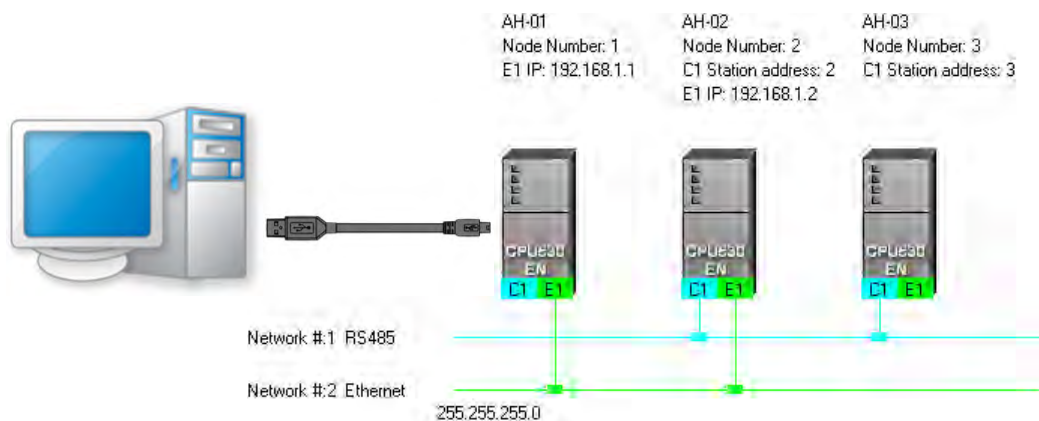
Base on the steps above, the screen will appear Optional Download window. When certain items are in grey, the function is not supported; select the time to download and click **OK**.



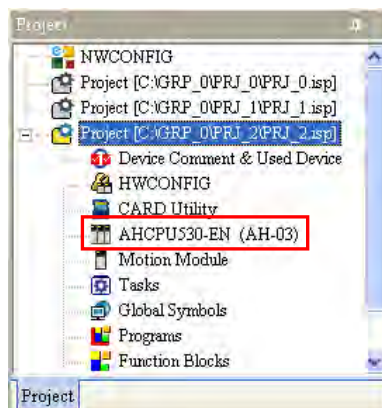
### 20.5.3 ISPSOft Routing Application

When completing the network structure in NWCONFIG and successfully downloaded Routing Table for each node, users can download and upload or monitor hosts (non-directly connected) through routing function in ISPSOft; but, to use the function in ISPSOft, the objective needs to be AH5xo series.

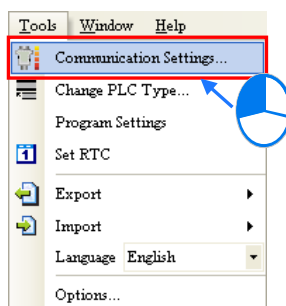
In the figure below, the actual connected host is AH-01. But, it is possible to monitor AH-03 through routing function.



- (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.

The screenshot shows a dialog box titled "Select a Driver". It contains the following fields and controls:

- Driver Name:** A dropdown menu with "Drv\_USB" selected.
- Station Address:** A dropdown menu with "0" selected.
- Routing Mode:** A checked checkbox.
- First Station:** A dropdown menu with "AH-01" selected.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

Red boxes in the image highlight the "Driver Name" field and the "Routing Mode" checkbox with its "First Station" dropdown.

# Chapter 21 Data Backup and Data Restoration



## Table of Contents

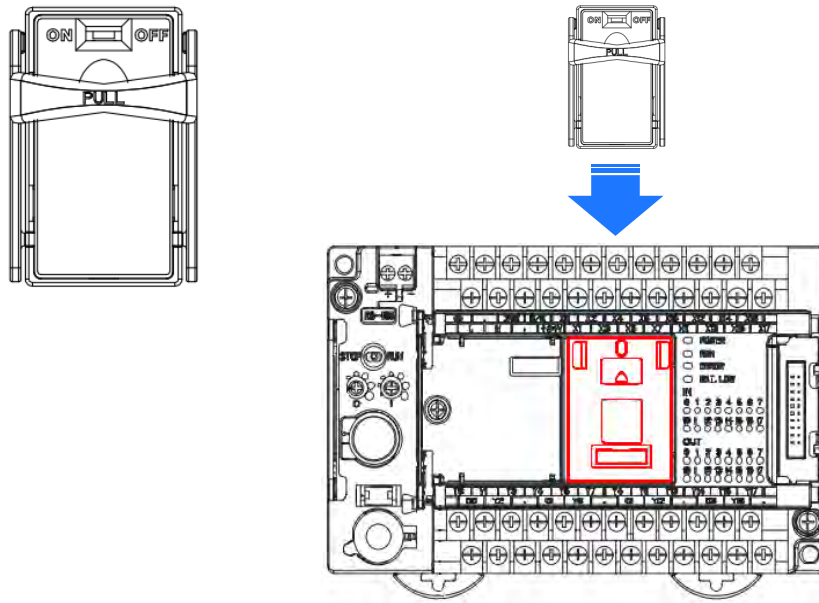
<b>21.1</b>	<b>Data Backup Memory Cards</b> .....	<b>21-2</b>
21.1.1	Introduction of Data Backup Memory Cards .....	21-2
21.1.2	Operating Data Backup by Memory Card.....	21-3
<b>21.2</b>	<b>Permanent Data Backup</b> .....	<b>21-4</b>
<b>21.3</b>	<b>CARD Utility</b> .....	<b>21-5</b>
21.3.1	Introduction of CARD Utility .....	21-5
21.3.2	Backup.....	21-7
21.3.3	Restoration .....	21-12

## 21.1 Data Backup Memory Cards

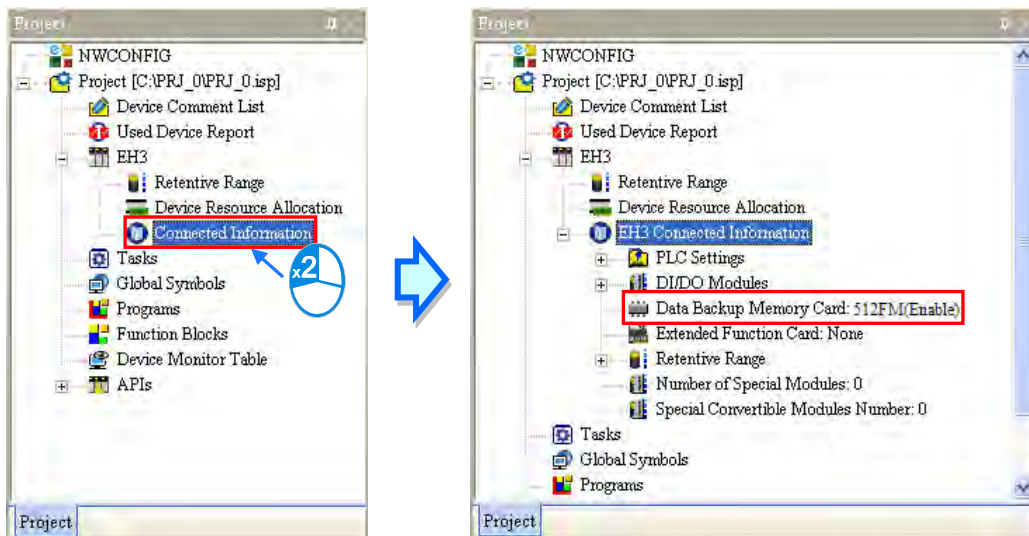
### 21.1.1 Introduction of Data Backup Memory Cards

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.

21

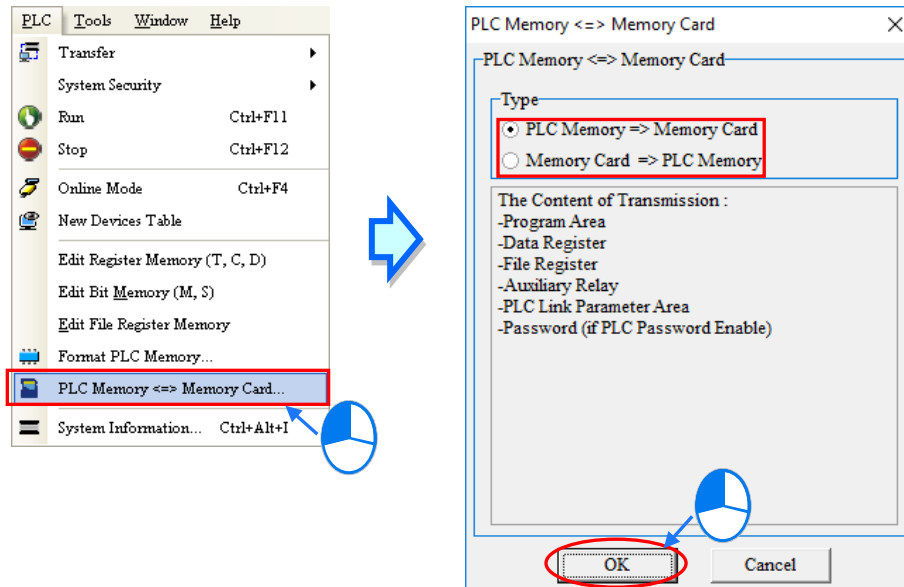


Install a memory card in a PLC, power the PLC up, unfold the PLC section in the project management area in ISPSOft, and double-click **Connected Information**. If ISPSOft is connected to the PLC normally, the system information about the PLC will be under the **Connected Information** section. The **Data Backup Memory Card** item indicates whether the memory card is enabled.



### 21.1.2 Operating Data Backup by Memory Card

Before executing data backup by memory card, please make sure that ISPSOft and PLC hosts are connected. 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**.



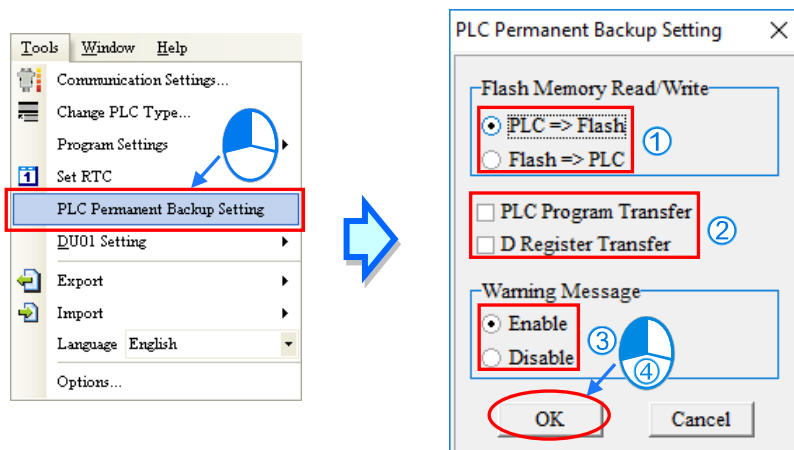
\*. Please refer to the related user manuals or technical documents for more information about data backup memory cards, the models which are supported, and the firmware versions which are supported.

## 21.2 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.

**21**

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**.





## 21.3 CARD Utility

### 21.3.1 Introduction of CARD Utility

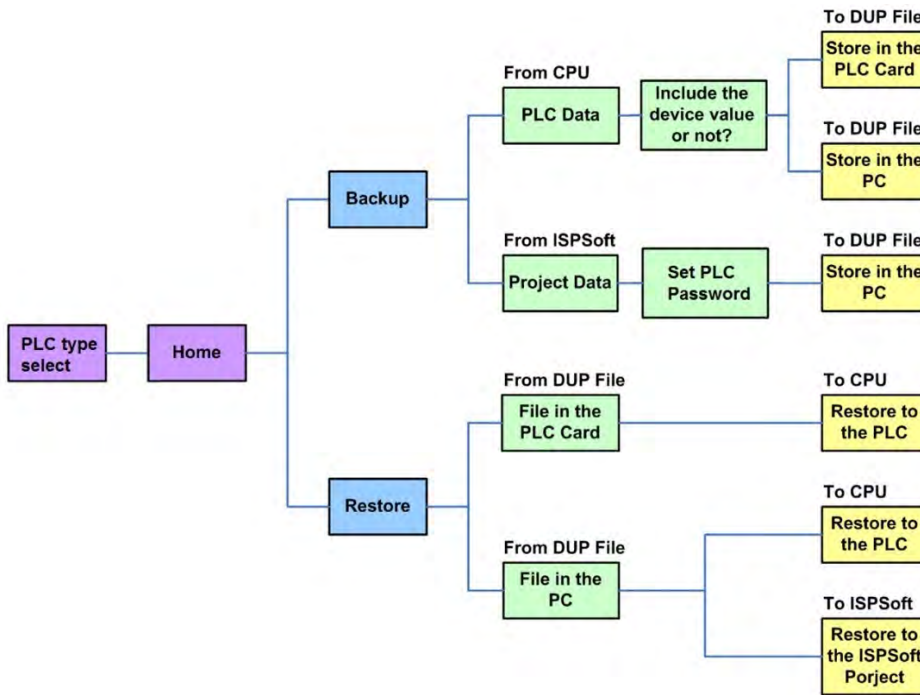
There are SD slots on AH/AS series CPU modules. Users can back up/restore data in an AH/AS series CPU module through a memory card. Besides, ISPSOft provides CARD Utility for users to back up/restore data or an ISPSOft project on AH/AS series CPU modules. The program codes, the parameter settings, the hardware configurations, and the network configurations in an AH/AS series CPU module or an ISPSOft project can be backed up. The values in the devices in an AH/AS series CPU module can also be backed up. For the module software in the HWCONFIG, only ECAT Builder supports data backup and restoration. Please refer to operation manuals or technical documents for more information about the SD card specifications.

As for the network configurations, the AH5x0 series will only back up data that is related to their own modules and ISPSOft CPU projects, such as Routing Table and Ether Link. Likewise, not all network configurations can be restored back to the ISPSOft project. Please refer to chapter 20 for more information about network configurations.

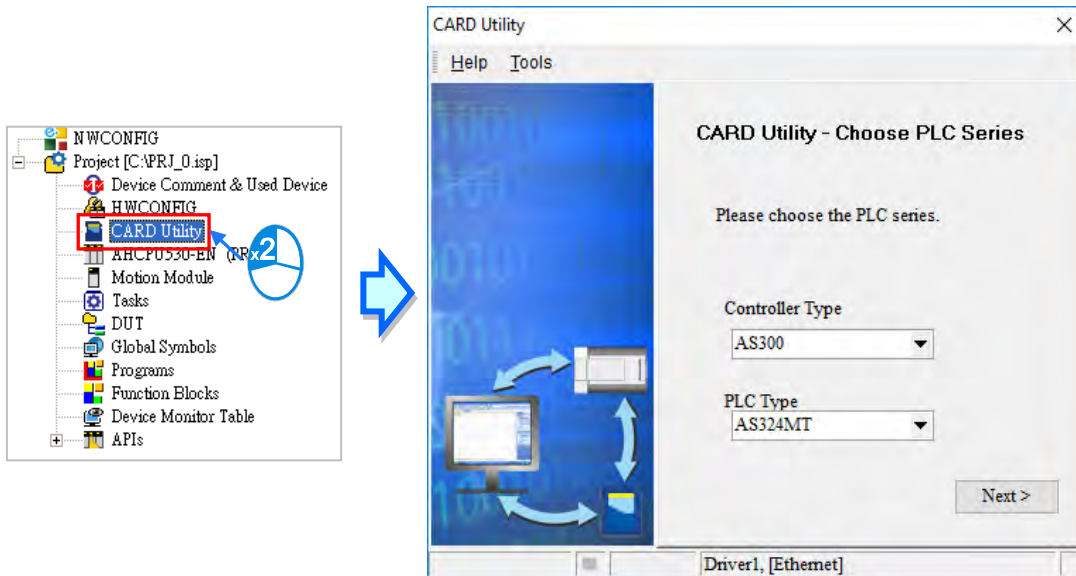
The functions supported by CARD Utility are described below. The diagram below is a flowchart.

- If users export data from an AH series CPU module as a backup file (\*.dup), the data exported can be saved in the memory card or in the computer. Users can also export the values in the devices in the AH/AS series CPU module.
- If users export an ISPSOft project of an AH500 series CPU module as a backup file (\*.dup) can only be saved in the computer and the values in the devices of the AH500 series CPU module are not included. Data such as register editing (\*.dvl), device status editing (\*.dvb), file register editing (\*.wft) of AS series and AHxxEMC can be seen as values in the device and can be exported and saved as backup files.
- Users can download the backup file saved in the inserted memory card back to the CPU module.
- Users can import a backup file (\*.dup) saved in a computer back to the AH series CPU module or restore the backup file to an ISPSOft project. For AS series, 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. Data such as register editing (\*.dvl), device status editing (\*.dvb), file register editing (\*.wft) of AS series and AHxxEMC can be seen as values in the device and can be imported back to restore the ISPSOft project.
- For AHxxEMC series users can backup the G-Code files on the SD card of the CPU module to the computer. And when restoring, users can restore the G-Code files on the computer to the SD card of the CPU module or to the ISPSOft project.

21



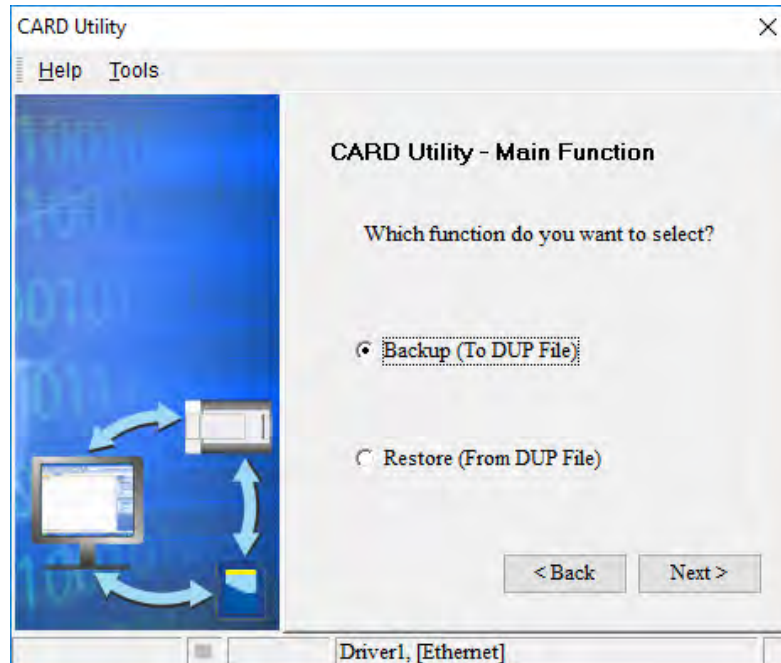
After users double-click **CARD Utility** in the project management area, the system will open the **CARD Utility** window. Select the controller and PLC type from the drop-down list and then click **Next** to start.



### 21.3.2 Backup

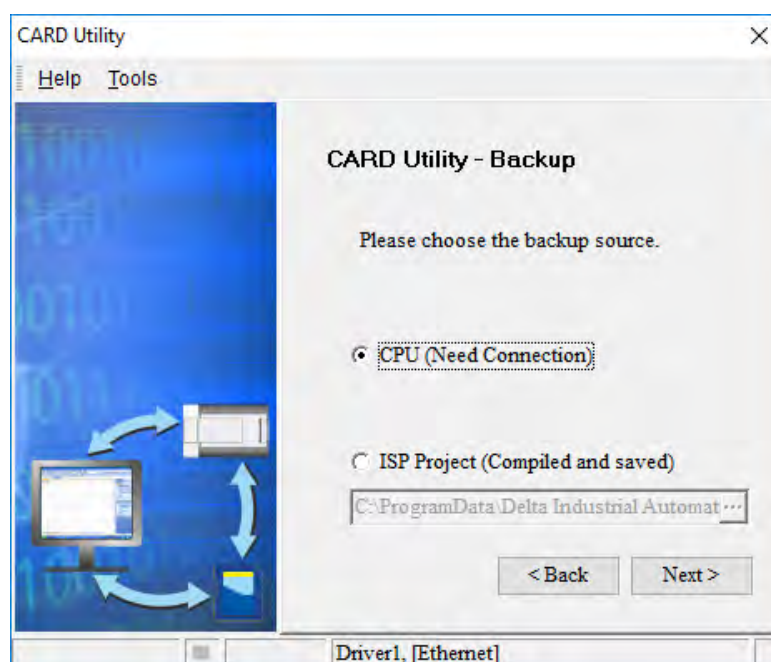
When the backup source/backup destination contains a host CPU or memory card, please check that ISPSOFT and the hosts is connected.

- (1) Select **Backup (To DUP File)** option button in the **CARD Utility** window, then click **Next**.

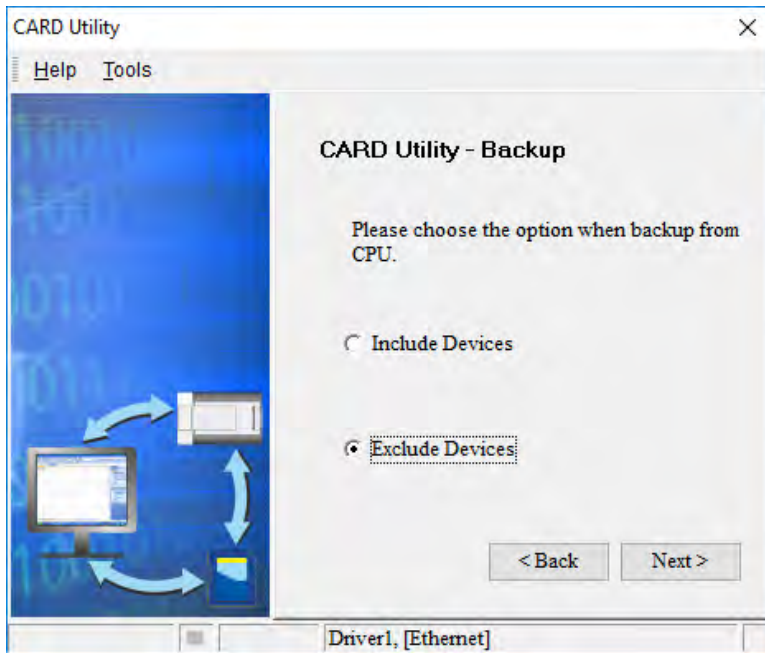


- (2) Select a backup source, and then click **Next**.

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; meanwhile, please re-compile and save to re-execute data backup.




- (3) When backup source is **CPU** , a window appears to choose whether the backup should include or exclude device content.

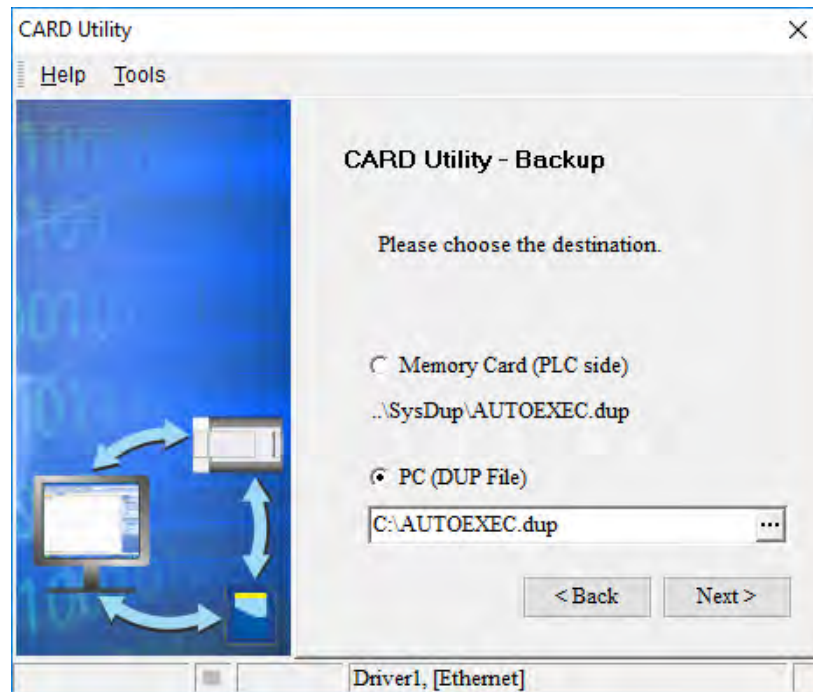


- (4) Select a backup destination. If the backup source is an ISPSOft project, the backup destination must be a computer.
- a. If the **Memory Card (PLC Side)** option button is selected, the filename of the backup file which will be produced will be **AUTOEXEC.dup**, and the path which points to the backup file is **Root directory of the memory card\PLC CARD\AH500\SysDup\AUTOEXEC.dup** for AH series.

For AS300 series, it is **Root directory of the memory card\ SDCard \PLC CARD\AS300\SysDup\AUTOEXEC.dup**

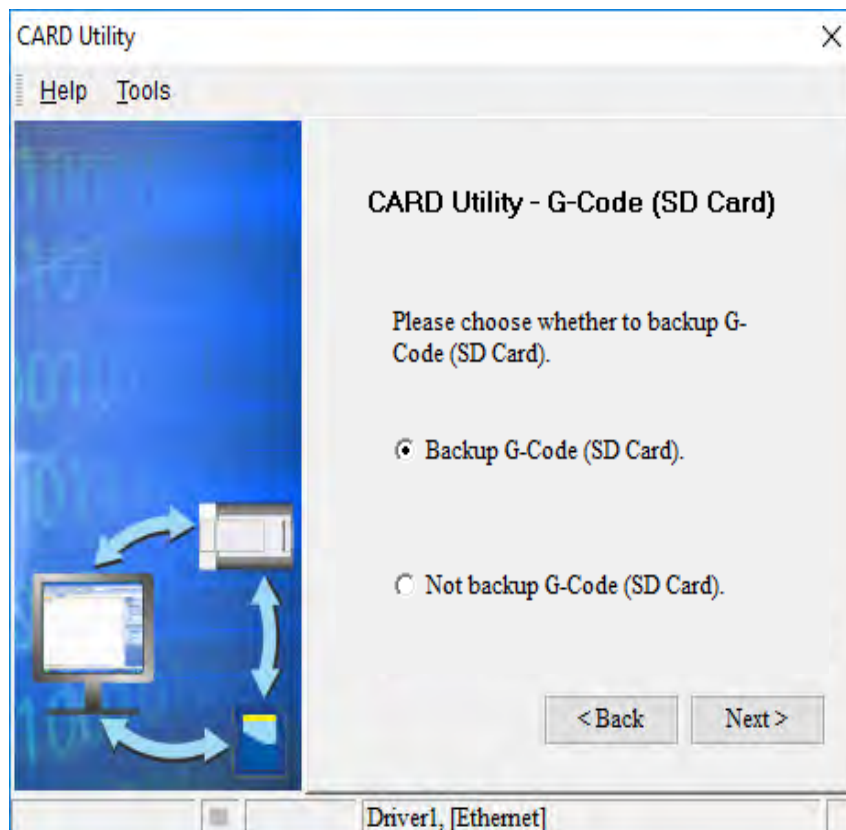
For AS200 series, it is **Root directory of the memory card\ SDCard \PLC CARD\AS200\SysDup\AUTOEXEC.dup**

- 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.



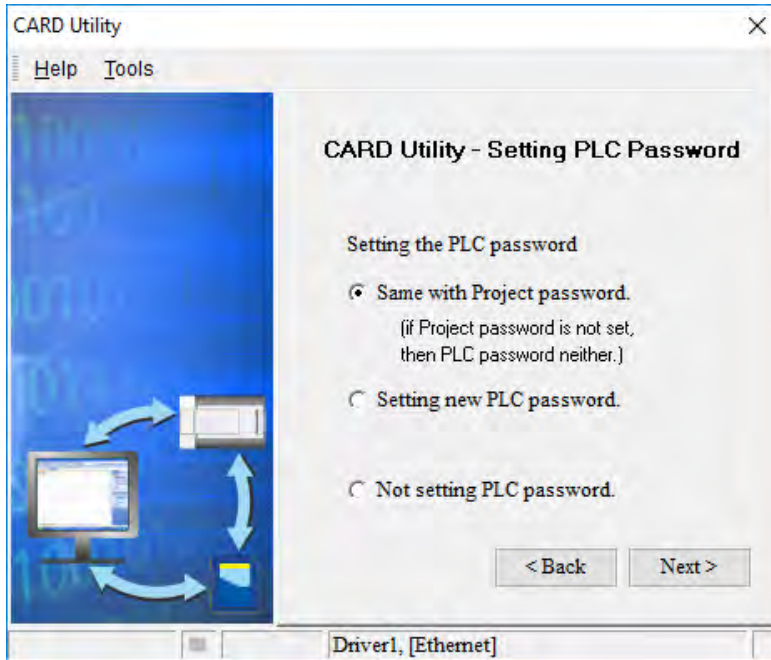
21

- (5) For AHxxEMC series when the backup is from a CPU module to the computer, users can also choose to backup the G-Code files on the SD card of the CPU module to the computer.

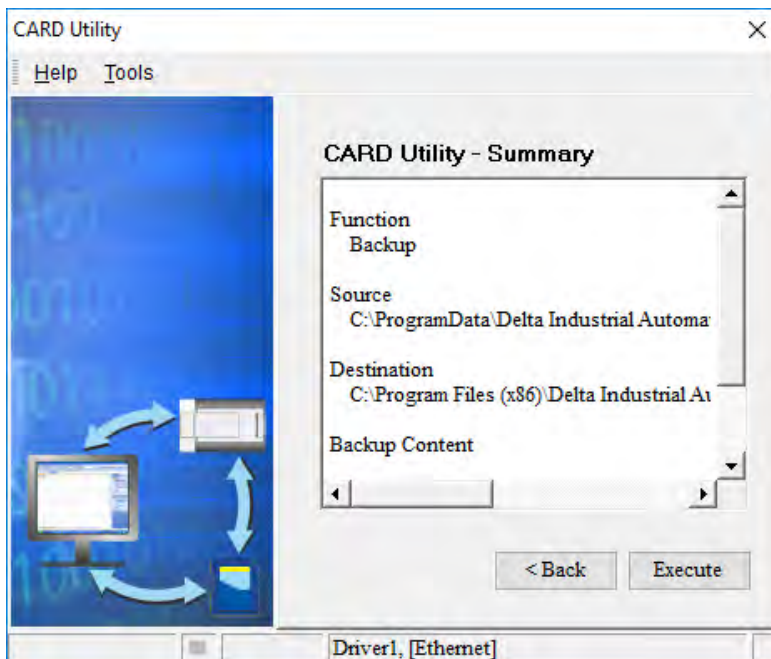


- (6) When the backup source is selected to the ISPSOft Project, users can set up the PLC password. The options are to set the password the same as the Project (If the password for the Project is not set, the same goes to the PLC password.), set a new PLC password, and do not set a PLC password. When the option “Setting new PLC password” is selected, the setup window will show up.

21

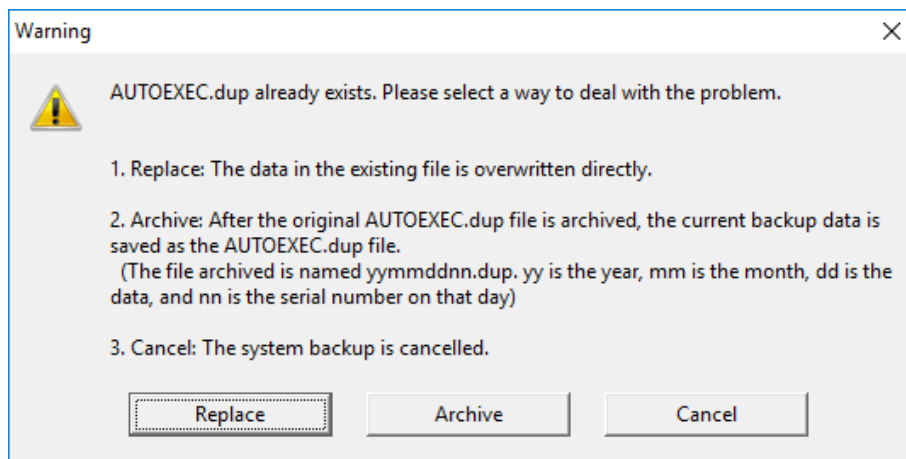


- (7) After the users make sure that the summary in the **CARD Utility** window is consistent with the data backup which will be performed, they can click **Execute**.



Even if the users click **Cancel** to stop ISPSOft from performing the data backup in the process of backing up data in the CPU module onto the memory card inserted in the CPU module, the CPU module will still performs the data backup. The users can turn off the 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 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.

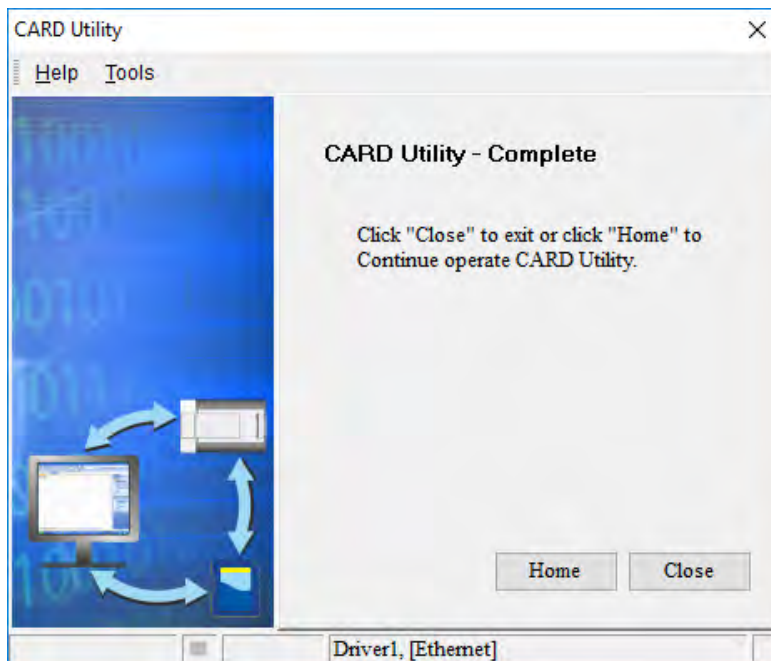


If the data backed up is protected by passwords, these passwords will also be backed up.

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, the project password and the PLC password set while setting up the backup.

- (8) After the data backup is performed, the users can click **Home** or **Close** in the **CARD Utility** window.

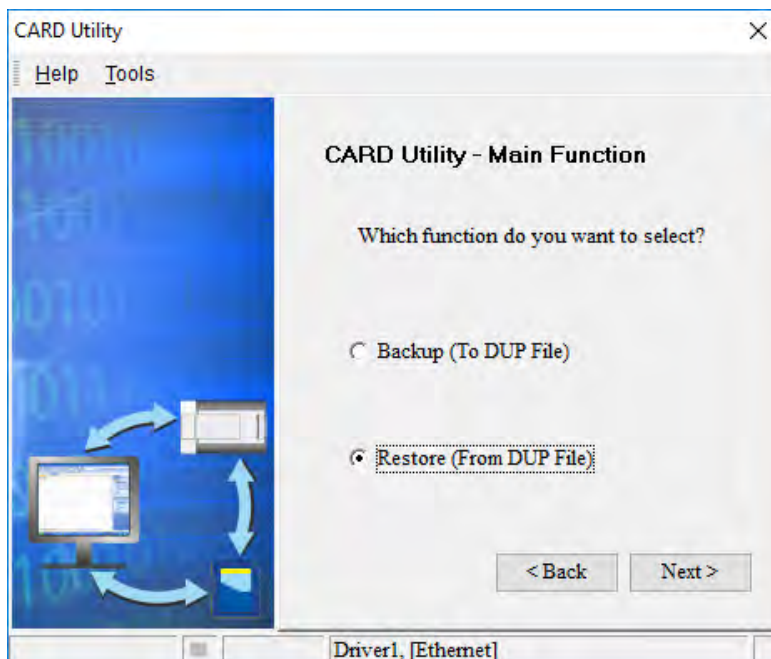
21




### 21.3.3 Restoration

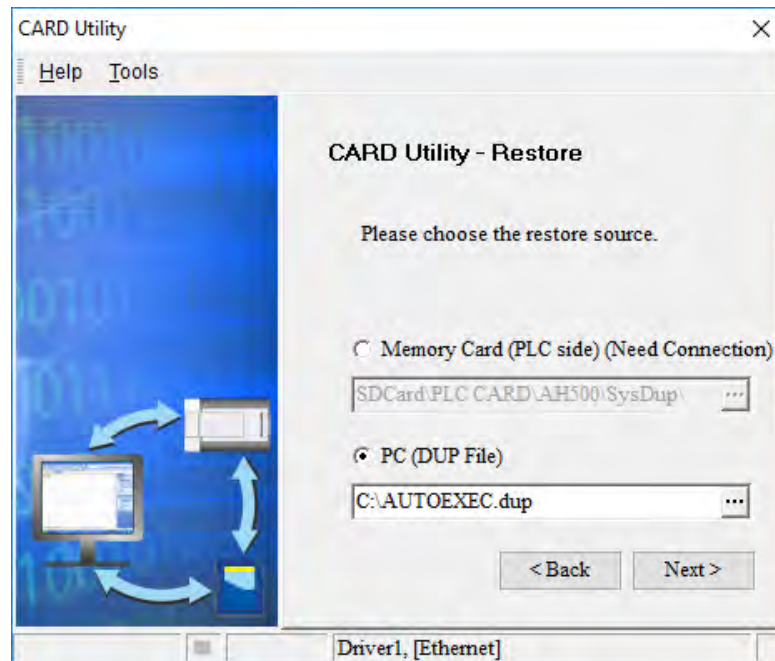
When restoration source/restoration destination has a host CPU or memory card, please make sure that ISPSOft and the host is connected

- (1) Select **Restore (From DUP File)** option button in the **CARD Utility** window, and then click **Next**.

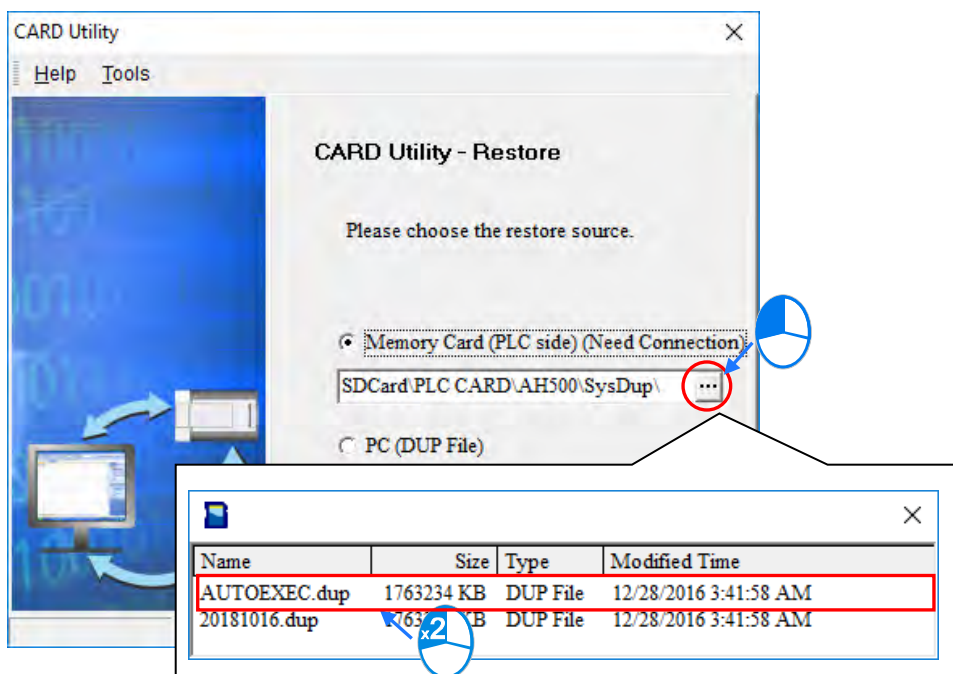




- (2) Click  at the right of the path column and select a backup file to restore.

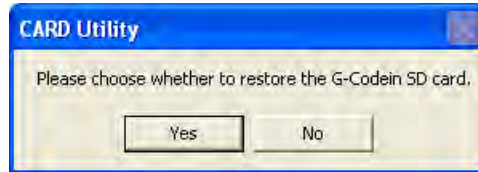


When **Memory Card (PLC side) (Need Connection)** is selected, the system automatically connect to a PLC and displays all backup files by memory card in the window. Double-click on the backup files for restoring.



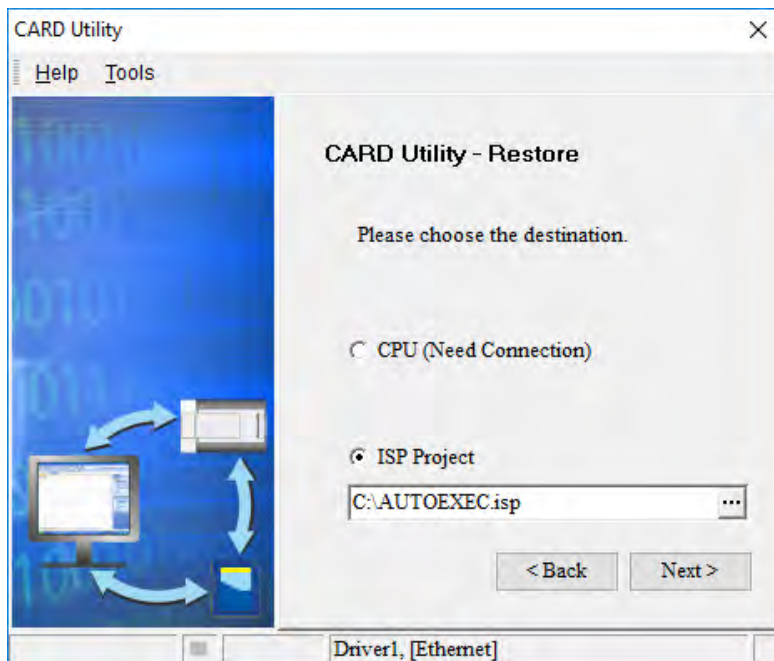
For AHxxEMC series when the backup is from a computer, the system will prompt to ask if you'd like to restore the G-Code in the computer and save the restored G-Code files to the SD card of the CPU module or ISPSOft project. If yes, the .nc files that share the same directory with the DUP files will be saved to the selected location.

# 21

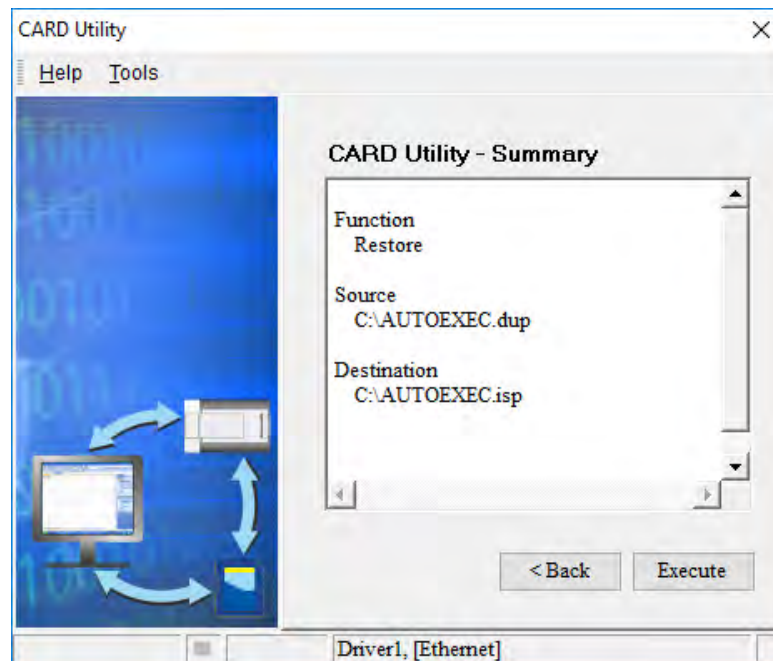


\*.G-Code filenames will be 100.nc~227.nc correspond to the G-Code items (100~227) in the SD card from the G-Code editor of the ISPSOft. For more information, please refer to chapter 22.

- (3) Select a restoring destination, and then click **Next**.
  - a. If the users want to put the backup file selected into the CPU module which is connected to ISPSOft, they have to select the **CPU (Need Connection)** option button.
  - b. If **ISP Project** is selected, click **...** in the directory column to setup the file name and path after restoring. When the path already contains the same file name, the file content will be covered by the data restored.



- (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**.



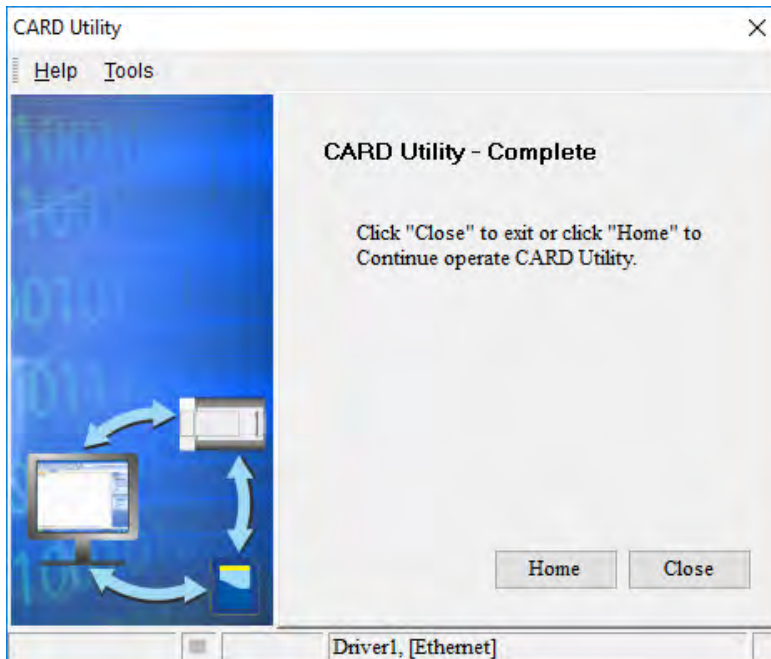
If the users click **Cancel** in the process of restoring data to the CPU module, the data will not be completely restored. To prevent the CPU module from operating incorrectly, the users have to restore the CPU module to the factory setting if they do not perform the data restoration again. Besides, the 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 CPU module. The users can turn off the CPU module to stop the data restoration from being performed.

If the restoration source/restoration destination contains a password and an ID, the process is as below.

Data restoration	Description
Memory card→CPU module	<p>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.</p> <p>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.</p> <p>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.</p>

<p>Computer→CPU module</p>	<p>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 executed.</p> <p>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.</p> <p>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.</p>
<p>Computer→ ISPSOft project</p>	<p>While restoring, if the backup file is PLC password protected, users will be asked for the password to proceed. After restoration, the ID and the password in the backup file will become the program ID and the project password in ISPSOft project but the PLC ID and PLC password will not be saved to ISPSOft.</p>

(5) When execution is complete, users can choose **Home** to return to the main page or or **Close**.



# Chapter 22 G-Code Editor and E-CAM Editor



## Table of Contents

<b>22.1 G-Code Editor .....</b>	<b>22-2</b>
22.1.1 About G-Code .....	22-2
22.1.2 Using G-Code Editor .....	22-2
22.1.3 G-Code Command Structure .....	22-7
22.1.4 G-Code Functions .....	22-10
22.1.5 G-Code Example of Drawing Delta Logo through Three-Axis Motion ...	22-31
<b>22.2 E-CAM Editor .....</b>	<b>22-33</b>
22.2.1 About E-CAM (Electronic CAM) .....	22-33
22.2.2 The Significance and Description of E-CAM .....	22-33
22.2.3 Using E-CAM Editor .....	22-35

## 22.1 G-Code Editor

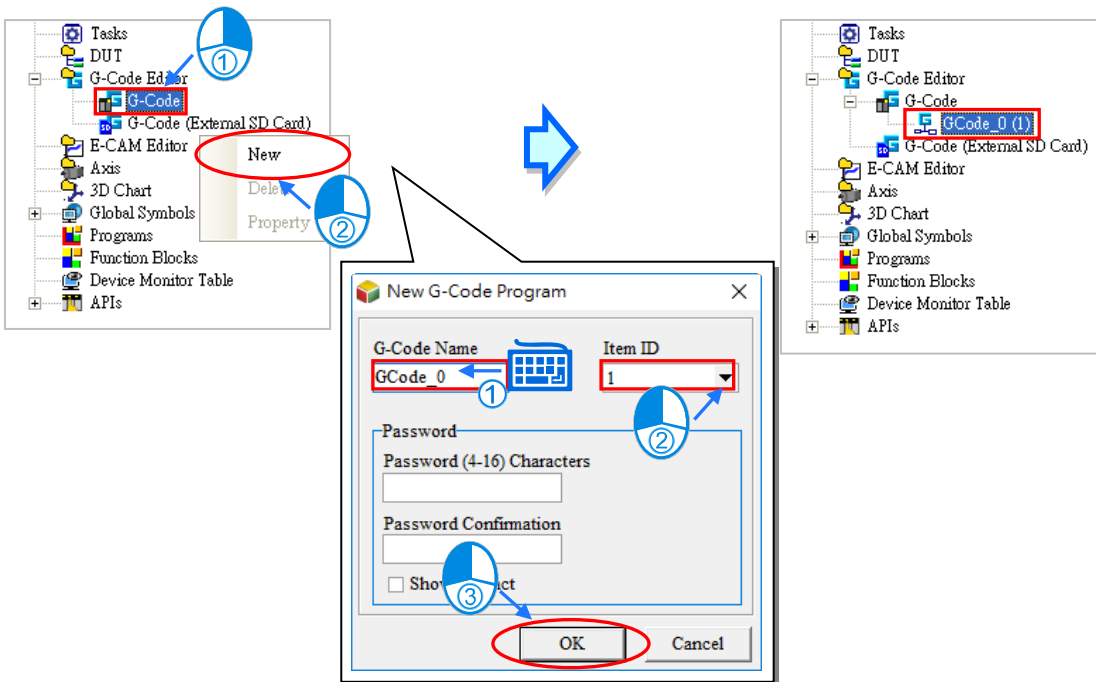
### 22.1.1 About G-Code

During the control of motion, users can generate G-codes for the complex processing paths of two axes (or more than two axes) by means of computer-aided manufacturing software, and the G-codes can be input to a controller which can drive the axes. The time of developing a motion program can be saved. G-code is a CNC (computer numerical control) programming language widely used in automatic equipment. Delta motion controllers and ISPSOft support the majority of G-codes on the market. The minorities of G-codes which are not supported are skipped, and are not executed.

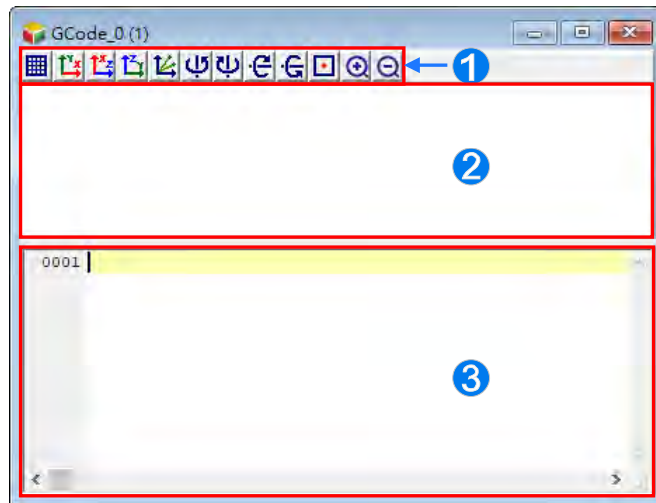
22

### 22.1.2 Using G-Code Editor

There are 2 options under the G-Code Editor, select G-Code to save the codes in the PLC and G-Code (External SD Card) to save the codes to the external SD Card on the PLC. (Currently, the ISPSOft does not support this function in DVPxxMC). Select one, right-click the selected option and then click New to open the New G-Code Program setting window. Type a name and an item ID for this G-Code. Users can also set up a password to protect the G-Code data. When a password is set, the system will ask for the password upon next editing. Up to 64 sets of G-Code can be created (max. 250KB for each set) in the PLC.















After the G-Code is created, an editing window will appear.



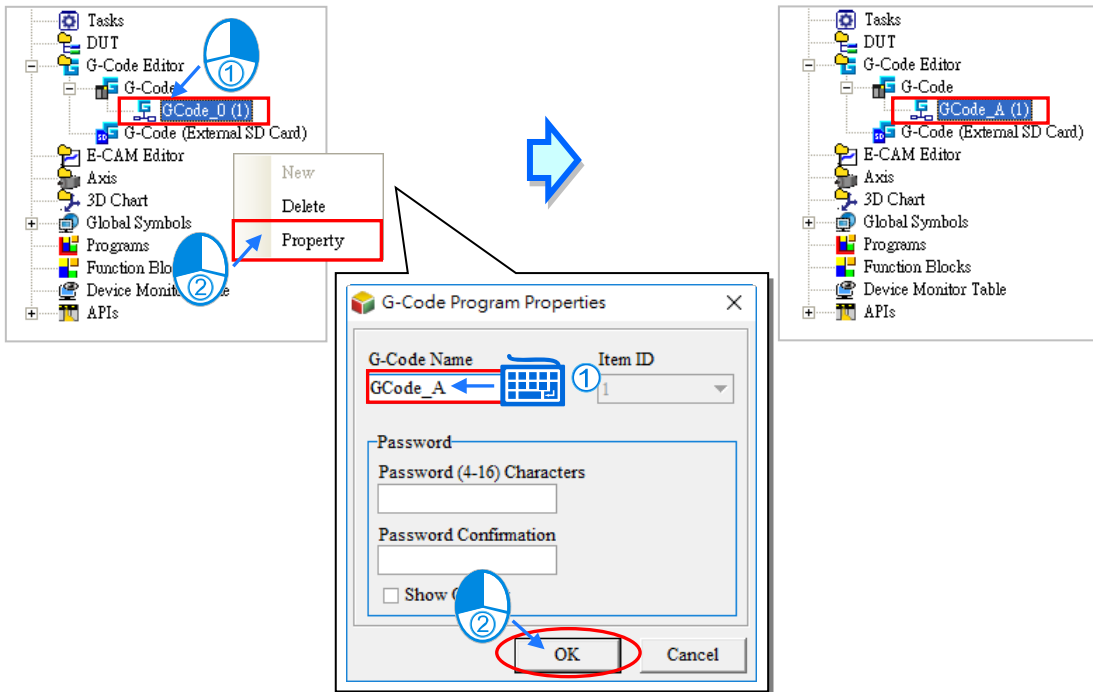
- ❶ Toolbar icons: functional buttons are provided for users to modify the G-Code easily.
- ❷ Toolpath view area: a toolpath is created according to the programs written in programming area.
- ❸ G-Code programming area: type the G-Code here.

The function buttons of the G-Code Editor are described below.

Icon	Name	Description
	Show/Hide Grid	Display grids in the toolpath view area, show/hide grids in the XY, YZ, XY plane respectively.
	Switch to the XY plane	Switch the view seen to the XY plane.
	Switch to the XZ plane	Switch the view seen to the XZ plane.
	Switch to the YZ Plane	Switch the view seen to the YZ plane.
	Show/Hide G0 Trace	Show/Hide the display of dotted lines representing rapid moves (G0)
	Vertical CCW Rotation	Rotate the toolpath view counter-clock wisely.
	Vertical CW Rotation	Rotate the toolpath view clock wisely.
	Horizontal Top Rotation	To reverse the toolpath view upwardly and horizontally
	Horizontal Down Rotation	To reverse the toolpath view downwardly and horizontally
	Auto Fit	Adjust the toolpath view area automatically to fit the screen
	Zoom In	Increase the magnification of the toolpath view.
	Zoom Out	Decrease the magnification of the toolpath view.

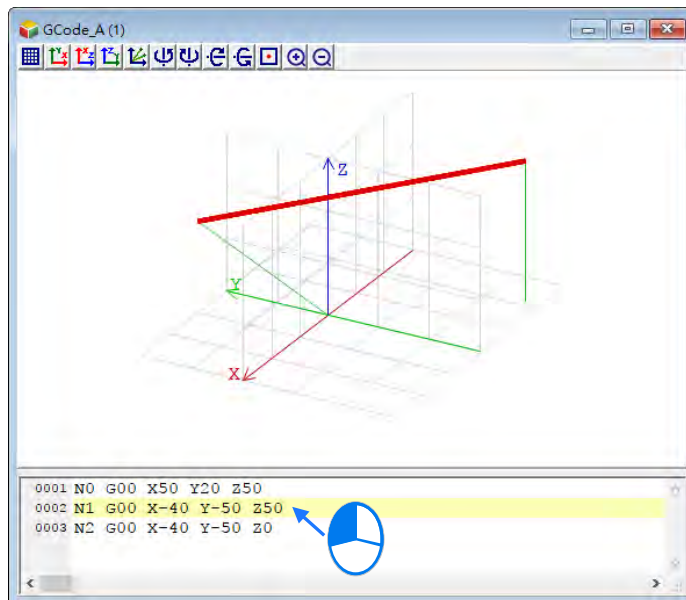
Users can right-click the created G-Code item to open the context menu and select **Delete** to delete

this item. Or select **Property** to edit the name and the password. Note that Item ID is not allowed to be changed.



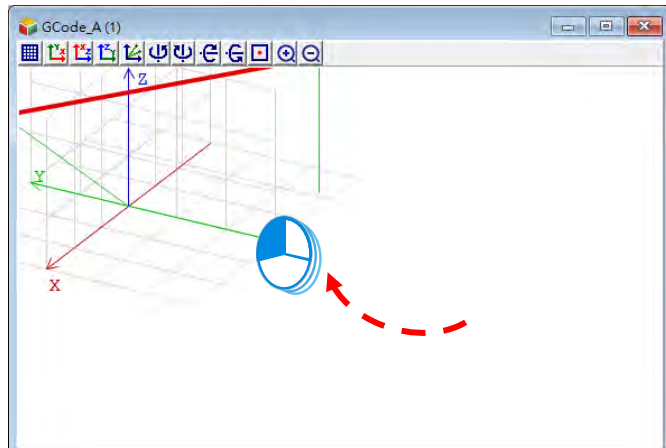
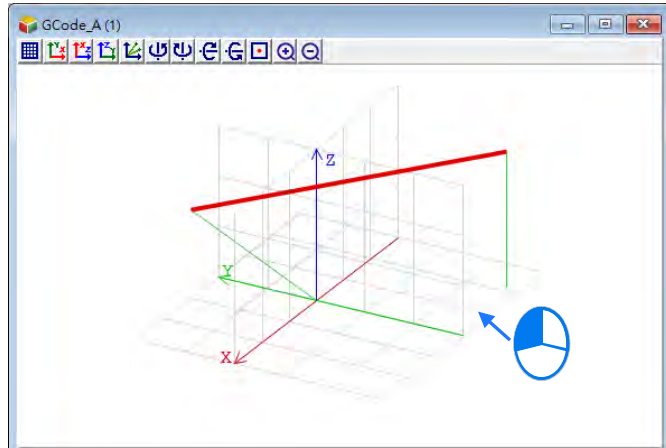
22

While writing a G-Code in the programming area, the current drawing G-Code will be shown in read in the toolpath view area.



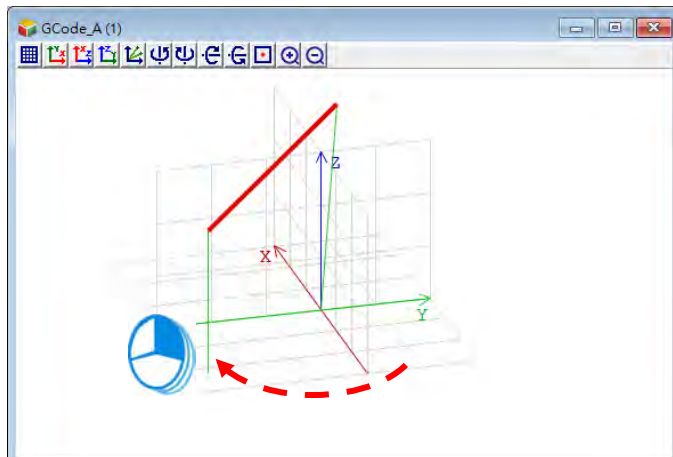
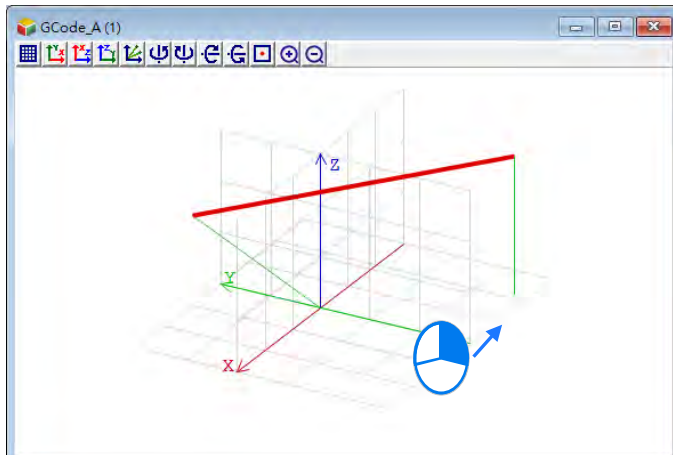


Users can click, hold and drag the mouse to move the toolpath in the toolpath view area.



22

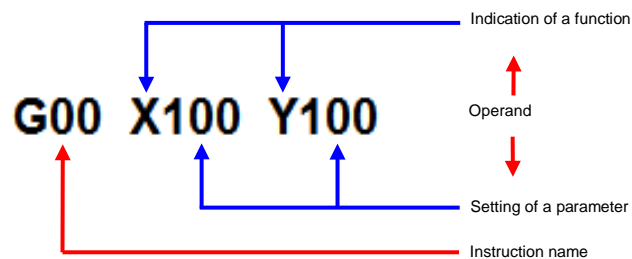
Users can right-click, hold and drag the mouse to rotate the toolpath in the toolpath view area.



After all the setups are done, download the parameters and the project to the PLC. Refer to section 17.1.3 for Downloading / Uploading a Project.

### 22.1.3 G-Code Command Structure

A G-code is composed of an instruction name and operands, and an operand is composed of the indication of a function and the setting of a parameter. In the figure below, the instruction name represents the function which is executed, the indication of functions represents the targets of the operands, and the setting of the parameters represents the values of the operands. The instruction G00 indicates that the x-axis and the y-axis are moved at the maximum speed to the target position (100, 100).



22

Here are some basic items of the supported formats and basic G-Code instructions.

- In the beginning of every G-Code, there is a parameter mark "N" to indicate a new line for example N1 or N2.
- As long as the format of an instruction is not destroyed, leaving spaces is allowed. For example, G01X50 Y15F20 is legal, and is equivalent to G01 X50 Y15 F20.
- 
- G00/G01/G02/G03 supports inheritance function (meaning the next row of codes without G code can inherit the previous G codes) while some G codes do not possess this function. The speed parameter (F) of G01/G02/G03 can be extended.

```
N0 G00 X50 Y15
N1 X-40 Y-50 ← G00 X-40 Y-50

N2 G01 X10 Y25 F20
N3 G03 X-40 Y-50 R100
      ↑
      G03 X-40 Y-50 R100 F20
```

- Input type of constant operand parameter: Arabic numerals and floating points  
E.g. N0 G0 X100.0 Y100.0
- Quick positioning (G0) does not need to set speed  
E.g. N0 G0 X100.2 Y500.0

- At least one of the position unit of axis parameters X/Y/Z is saved in G00, G01, while other omitted axis parameters can refer to the closest parameter values from the previous rows of codes.
- There are two ways to execute comments :
  - Use “and” to wrap comments, but nested comments are not allowed. E.g. (Comment)
  - Add words after “%”, e.g. %comment
- The operand parameters X, Y, Z, I, J, K, R can be written in any order and not a particular order to fulfill the needs of writing G codes.

E.g: N0 G1 X100 Z300 Y200.45 F400 = > N0 G1 X100 Y200.45 Z300 F400



The G codes and input formats supported by DVPxxMC series motion controllers are listed in the following table.

G code	Function	Format
G0	Rapid Positioning	N_G0 X_Y_Z_A_B_C
G1	Linear interpolation	N_G1 X_Y_Z_A_B_C_F_
G2	Clockwise circular arc/helical interpolation	N_G2 X_Y_Z_I_J_F_ N_G2 X_Y_Z_I_K_F_ N_G2 X_Y_Z_J_K_F_
		N_G2 X_Y_Z_R_F_
G3	Anticlockwise circular arc/helical interpolation	N_G3 X_Y_Z_I_J_F_ N_G3 X_Y_Z_I_K_F_ N_G3 X_Y_Z_J_K_F_
		N_G3 X_Y_Z_R_F_
G4	Dwell instruction	N_G4 X_ N_G4 P_
G17	XY plane for circular interpolation	N_G17
G18	XZ plane for circular interpolation	N_G18
G19	YZ plane for circular interpolation	N_G19
G50	Precise stop	N_G50

<b>G code</b>	<b>Function</b>	<b>Format</b>
G51	Circular arc transition	N_G51 D_
G52	Smooth path transition	N_G52
G90	Absolute mode	N_G90
G91	Relative mode	N_G91
M0~M99	M Code	N_M_D_

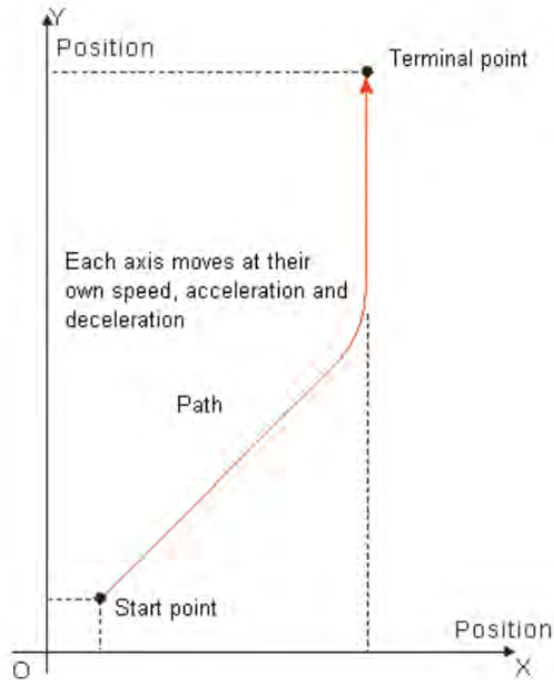
## 22.1.4 G-Code Functions

### 22.1.4.1 G0 (Rapid Positioning)

- Function:

Axes move from current position to the terminal position at the given speed. Maximum 8 axes can be controlled and they are independent with each other in motion.

And the motion path figure is displayed below.



➤ Format: N\_G0 X\_Y\_Z\_A\_B\_C

➤ Parameter explanation:

N\_: The row number of G code in NC program.

X\_: Specify the terminal position of axis X, Unit: unit, data type: REAL.

Y\_: Specify the terminal position of axis Y, Unit: unit, data type: REAL.

Z\_: Specify the terminal position of axis Z, Unit: unit, data type: REAL.

A\_: Specify the terminal position of axis A, Unit: unit, data type: REAL.

B\_: Specify the terminal position of axis B, Unit: unit, data type: REAL.

C\_: Specify the terminal position of axis C, Unit: unit, data type: REAL.

- Instruction explanation:

1. G0 can control one or more axes and for other axes they remain default values.
2. The speed, acceleration, deceleration and jerk of each axis in motion can be set via DMC\_SetG0Para instruction.

- Example:

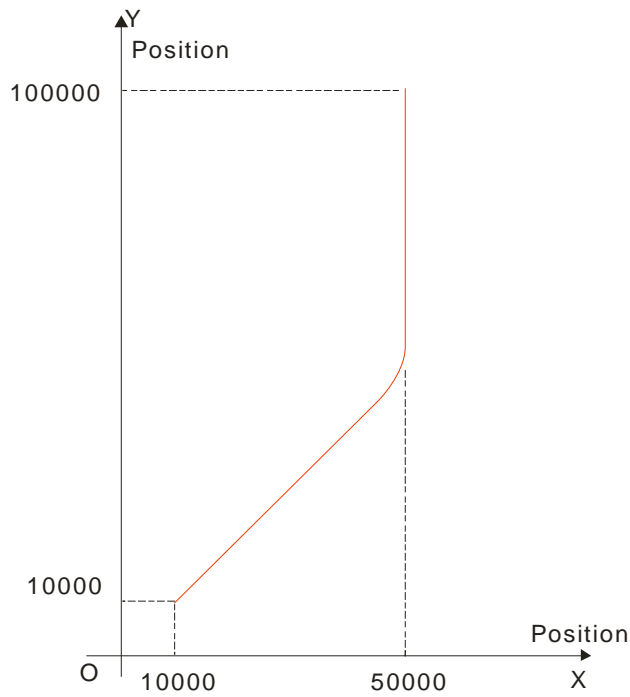
The initial positions of axis X, Y are both 10000 units and their axis parameters are both default values.

The G codes to be executed are:

N00 G90

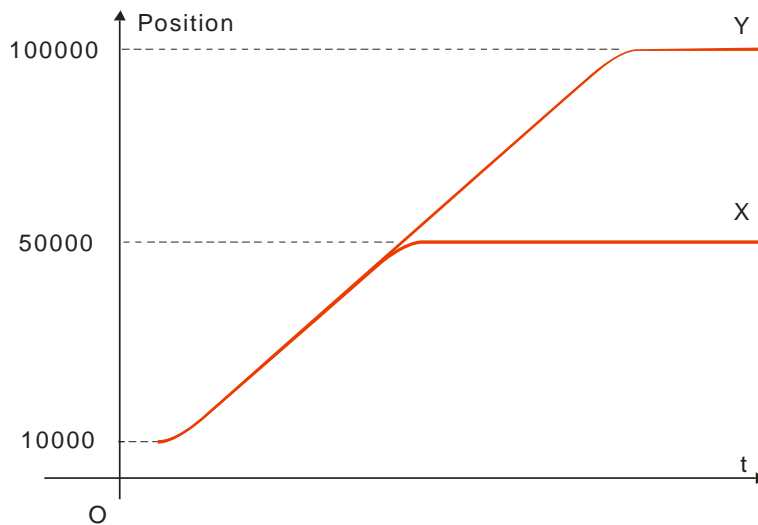
N01 G0 X50000 Y100000

After G codes are executed, the Y/X curve for the whole movement process is shown below:



22

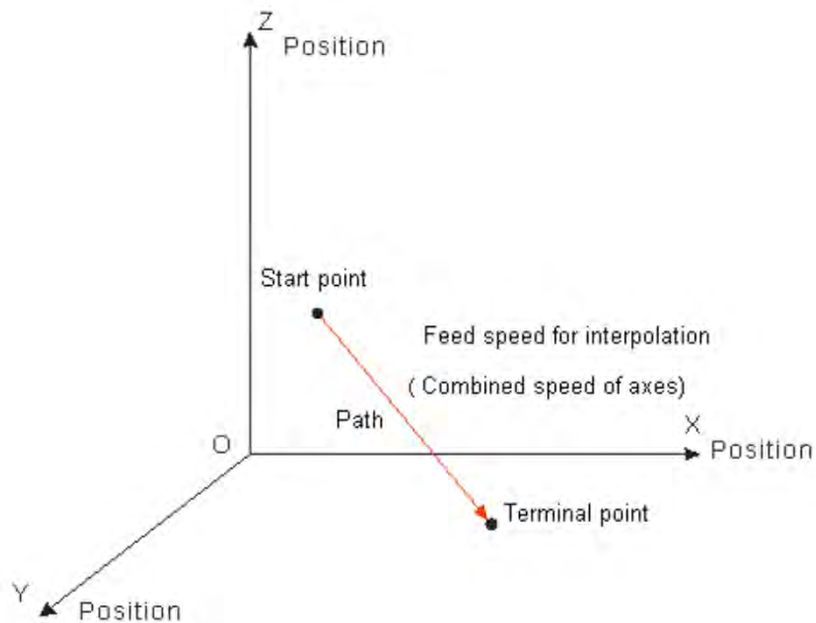
After G codes are executed, the Position/Time curve for the whole movement process is shown below:



### 22.1.4.2 G1 (Linear Interpolation)

- Function:

The cutter starts off from one point and moves straight to the target position at a given speed. The instruction can control up to 8 axes and all axes start up or stop simultaneously. Three axes control the position of the cutter together as the figure shows below.



- Format: N\_G1 X\_Y\_Z\_A\_B\_C \_F\_

- Parameter explanation:

N\_: The row number of G code in NC program

X\_: Specify the terminal position of axis X, Unit: unit, data type: REAL.

Y\_: Specify the terminal position of axis Y, Unit: unit, data type: REAL.

Z\_: Specify the terminal position of axis Z, Unit: unit, data type: REAL.

A\_: Specify the terminal position of axis A, Unit: unit, data type: REAL.

B\_: Specify the terminal position of axis B, Unit: unit, data type: REAL.

C\_: Specify the terminal position of axis C, Unit: unit, data type: REAL.

F\_: Specify the feed speed of the cutter, unit: unit/second, data type: REAL.

When the cutter moves at a constant speed, the combined speed of all axes in G code is equal to F value.

The method of calculation is shown as below.

When two axes exist,  $F = \sqrt{V_1^2 + V_2^2}$ .

When three axes exist,  $F = \sqrt{V_1^2 + V_2^2 + V_3^2}$ .



For more axes, F value could be calculated in the same way as above.

- Instruction explanation:

1. G1 can control one or more axes and for other axis they remain default values.
2. The velocity, acceleration, deceleration and jerk can be set via DMC\_SetG1Para.

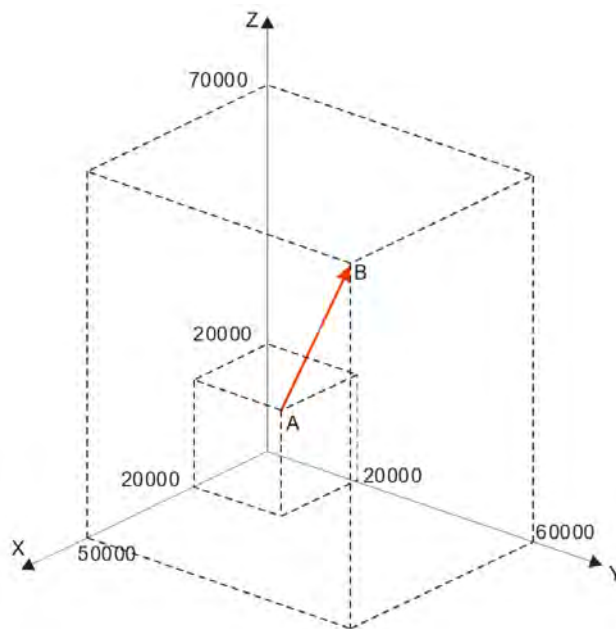
- Example:

The initial positions of axis X, Y, Z are all 20000 units and their axis parameters are all default values. The G codes to be executed are below:

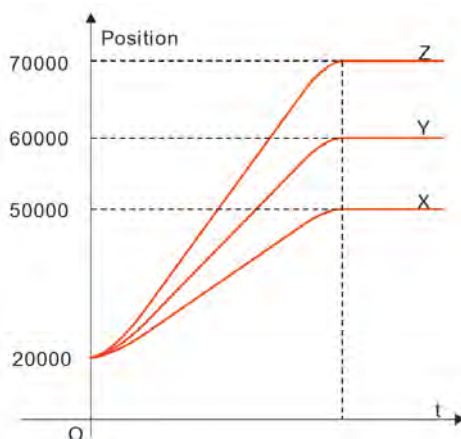
N0 G90

N1 G1 X50000 Y60000 Z70000

After G codes are executed, the Y/X curve for the whole movement process is shown below:



After G codes are executed, the Position/Time curve for the whole movement process is shown below:

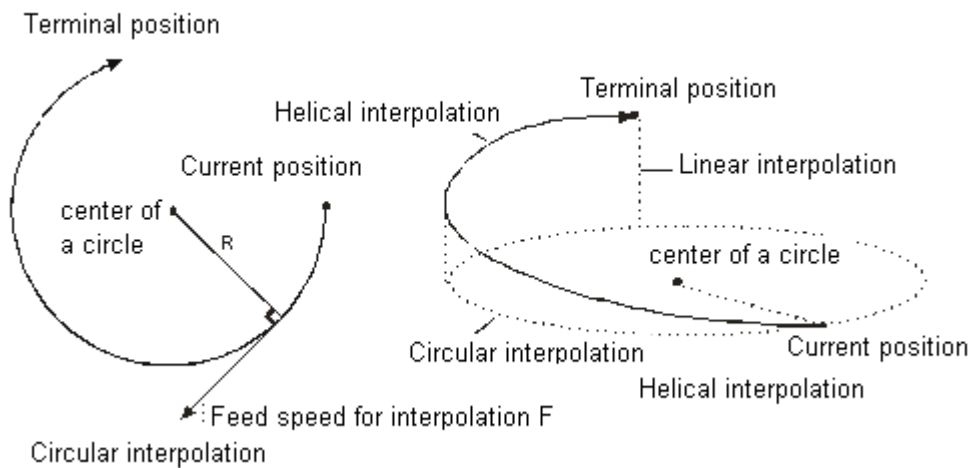


### 22.1.4.3 G2 (Clockwise Circular/ Helical Interpolation)

- Function:

Circular interpolation: The cutter conducts the cutting of the object which is to be processed in the clockwise direction at the feed speed given by parameter F on the circular arc with the fixed radius or fixed circle center of the specified plane.

Helical interpolation: The cutter moves in the clockwise direction on the circular arc of the specified plane, which is circular interpolation and simultaneously moves in a straight line perpendicular to the specified plane at the feed speed given by parameter F, which is linear interpolation.



- Format:

Format 1: N\_G2 X\_Y\_Z\_I\_J\_F\_

Format 2: N\_G2 X\_Y\_Z\_I\_K\_F\_

Format 3: N\_G2 X\_Y\_Z\_J\_K\_F\_

Format 4: N\_G2 X\_Y\_Z\_R\_F\_

- Parameter explanation:

N\_: The row number of G code in NC program

X\_: Specify the terminal position of axis X; Unit: unit, data type: REAL..

Y\_: Specify the terminal position of axis Y; Unit: unit, data type: REAL..

Z\_: Specify the terminal position of axis Z; Unit: unit, data type: REAL..

I\_J\_: Specify the coordinate position of a circle center on XY plane, Unit: unit, data type: REAL.

I\_K\_: Specify the coordinate position of a circle center on XZ plane, Unit: unit, data type: REAL.

J\_K\_: Specify the coordinate position of a circle center on YZ plane, Unit: unit, data type: REAL.

R\_: Specify the radius of the arc, data type: REAL.

F: Specify the feed speed of the cutter, Unit: unit/second, data type: REAL.

When the cutter moves at a constant speed, the combined speed of all axes in G code is equal to F value.

The method of calculation is shown as below.

When two axes exist,  $F = \sqrt{V_1^2 + V_2^2}$ .

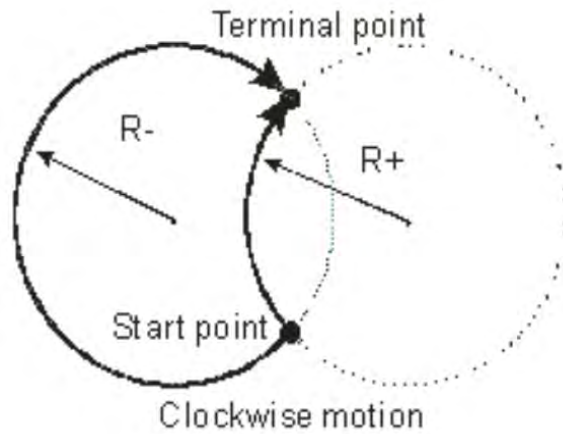
When three axes exist,  $F = \sqrt{V_1^2 + V_2^2 + V_3^2}$ .

For more axes, F value could be calculated in the same way as above.

- Instruction explanation:

1. Two axes among axis X, Y and Z make the circular interpolation on the plane specified by instruction G17/G18/G19. The third axis makes the linear interpolation in the direction perpendicular to the specified plane.
2. F can be omitted. If there is only one row of code in the G-code editing area and F is omitted, the velocity, acceleration and deceleration of the cutter are decided by axis group parameters. They can be set via DMC\_SetG1Para instruction.  
If there are multiple rows of codes and F in G2 code is omitted, the velocity, acceleration, deceleration of the cutter are based on valid F in the previous rows of codes above the row where G2 is. If the previous rows of G codes have not specified F, "maximum velocity", "maximum acceleration" and "maximum deceleration" among axis group parameters will prevail.
3. In absolute mode for G90, the terminal point of the circular arc is the absolute coordinate value regarding 0 unit in their own directions as reference. In relative mode for G91, the terminal point of the circular arc is the incremental value relative to the start point of the circular arc.
4. No matter whether in the absolute mode or in relative mode, the coordinates of the circle center I\_J\_(I\_K\_/J\_K\_) are always relative coordinates with the start point as reference
5. Regarding the relation between a coordinate plane and I, J, and K, at most two of I, J, and K exist in a circular instruction. Which two they are is determined by the corresponding plane, e.g. only I and J are allowed to be on XY plane.
6. Different from format 1, format 2 and format 3, format 4 decides a circular arc via the start

point, terminal point and radius. If the input value on the right of R parameter is a positive number (R+), the circular arc is the minor arc less than 180 degrees; if the input value on the right of R parameter is a negative number (R-), the circular arc is the major arc more than 180 degrees.



- Example

Current position (1000, 3000), axis parameters: default values,

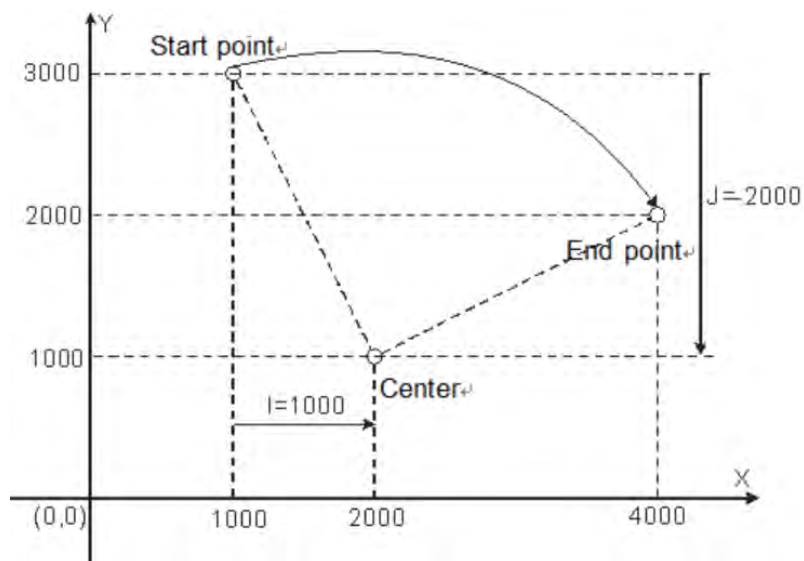
The G codes to be executed are below:

```
N00 G90
```

```
N01 G17
```

```
N02 G2 X4000 Y2000 I1000 J-2000 F5000
```

After G codes are executed, the Y/X curve for the whole movement process is shown below:

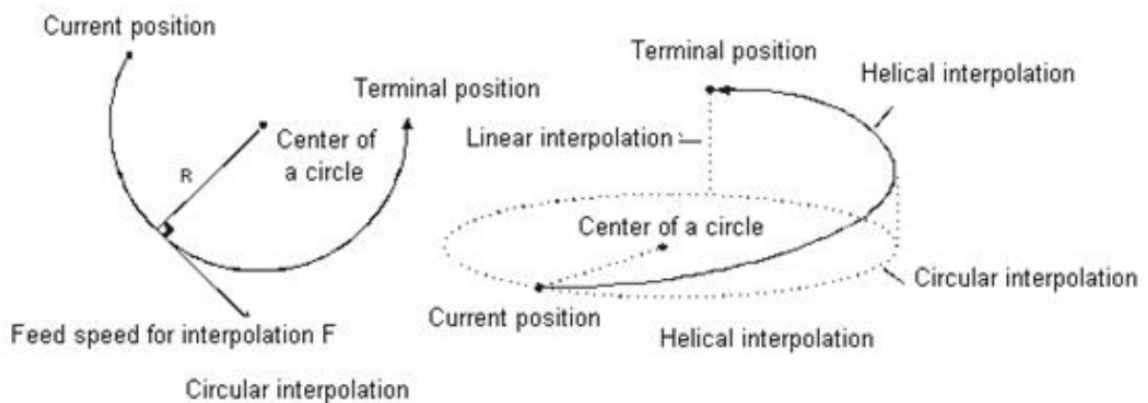


#### 22.1.4.4 G3 (Anticlockwise Circular /Helical Interpolation)

- Function explanation:

Circular interpolation: The cutter conducts the arc cutting of the object to be processed in the anticlockwise direction at the feed speed given by parameter F on the circular arc with the fixed radius or the fixed center on the specified plane.

Helical interpolation: The cutter moves in the anticlockwise direction on the circular arc of the specified plane, which is a circular interpolation and simultaneously moves in a straight line perpendicular to the specified plane at the feed speed given by parameter F, which is a linear interpolation.



- Format:

Format 1: N\_G3 X\_Y\_Z\_I\_J\_F\_

Format 2: N\_G3 X\_Y\_Z\_I\_K\_F\_

Format 3: N\_G3 X\_Y\_Z\_J\_K\_F\_

Format 4: N\_G3 X\_Y\_Z\_R\_F\_

- Parameter explanation:

N\_: The row number of G code in NC program

X\_: Specify the terminal positions of axis X, Unit: unit, data type: REAL.

Y\_: Specify the terminal positions of axis Y, Unit: unit, data type: REAL.

Z\_: Specify the terminal positions of axis Z, Unit: unit, data type: REAL.

I\_J\_: Specify the coordinate position of the circle center on XY plane, Unit: unit, data type: REAL.

I\_K\_: Specify the coordinate position of the circle center on XZ plane, Unit: unit, data type: REAL.

J\_K\_: Specify the coordinate position of the circle center on YZ plane, Unit: unit, data type: REAL.

R\_: Specify the radius of the arc, data type: REAL.

F: Specify the feed speed of the cutter, Unit: unit/second, data type: REAL.

When the cutter moves at a constant speed, the combined speed of all axes in G code is equal to F value.

The method of calculation is shown as below.

When two axes exist,  $F = \sqrt{V_1^2 + V_2^2}$ .

When three axes exist,  $F = \sqrt{V_1^2 + V_2^2 + V_3^2}$ .

For more axes, F value could be calculated in the same way as above.

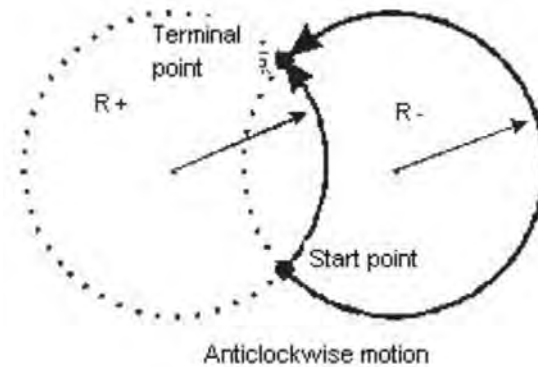


- Instruction explanation:

1. Two axes among axis X, Y and Z make the circular interpolation on the plane specified by G17/G18/G19. The 3rd axis makes the linear interpolation in the direction perpendicular to the specified plane.
2. F\_ can be omitted. If there is only one row of code in the G-code editing area and F is omitted, the velocity, acceleration and deceleration of the cutter are determined by axes group parameters. The parameters for velocity, acceleration and deceleration can be set by DMC\_SetG1Para.  
  
If there are multiple rows of codes and F in G3 code is omitted, the velocity, acceleration and deceleration of the cutter are based on valid F in the previous rows of codes above the row where G3 is. If the previous rows of G codes have not specified F, "maximum velocity", "maximum acceleration" and "maximum deceleration" among axis group parameters will prevail.
3. In absolute mode for G90, the terminal point of a circular arc is of absolute coordinate values regarding 0 unit in their respective directions as reference. In relative mode for G91, the terminal point of a circular arc is of incremental values relative to the start point of the circular arc.
4. No matter whether in the absolute mode or in relative mode, the coordinates of the circle center I\_J\_(I\_K\_/J\_K\_) are always the relative coordinates with the start point as reference.
5. Regarding the relation between a coordinate plane and I, J, and K, at most two of I, J, and K exist in a circular instruction. Which two they are is determined by the corresponding plane, e.g. only I and J are allowed to be on XY plane.
6. Different from format 1, format 2 and format 3, format 4 determines a circular arc via the start point, terminal point and radius. If the input value on the right of R parameter is a

positive number (R+), the circular arc is a minor arc less than 180 degrees; if the input value on the right of R parameter is a negative number (R-), the circular arc is a major arc more than 180 degrees.

The following solid curves are the motion paths when G3 selects R+ and R- and the arrows on the arcs refer to the motion direction



22

- Example

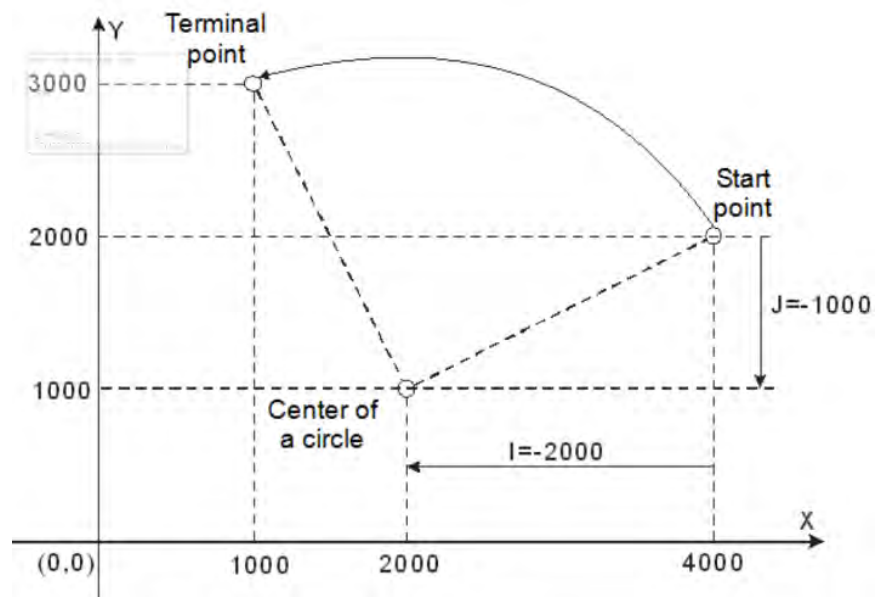
Current position (4000, 2000), axis parameters: default values.

The G codes to be executed are below:

N00 G90

N01 G17 N02 G3 X1000 Y3000 I=-2000 J=-1000

After G codes are executed, the Y/X curve for the whole movement process is shown below:



### 22.1.4.5 G4 (Dwell Instruction)

- Function: Dwell instruction

- Format:

Format 1: N\_G4 X\_

Format 2: N\_G4 P\_

- Parameter explanation:

N\_: The row number of G code in NC program

X\_: Specify the delay time, unit: second, data type: REAL.

P\_: Specify the delay time, unit: millisecond, data type: REAL.

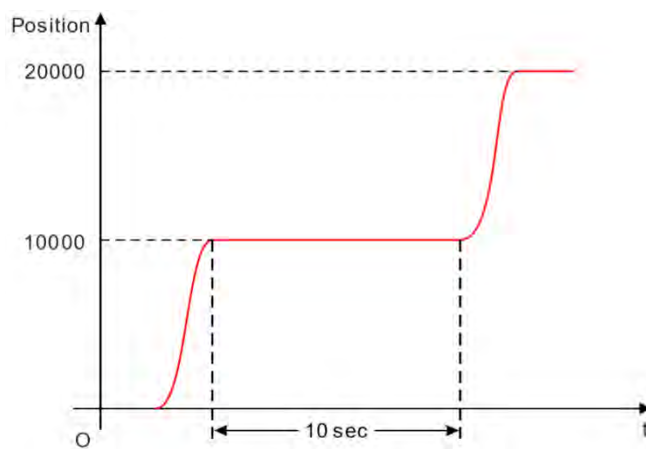
- Example:

```
N00 G1 X10000
```

```
N01 G4 X10 (or N01 G4 P10000)
```

```
N02 G1 X20000
```

After G codes are executed, the Position/t curve for the whole movement process is shown below:





### 22.1.4.6 G17/G18/G19 (Specify Circular Interpolation Plane)

- Function:

The three instructions are used for deciding the plane selection of circular interpolation or helical interpolation and have no impact on the linear interpolation.

While the program is being executed, the three work planes can be switched with each other.

If no plane option is set, the initial state of the system is XY plane (G17).

- Format:

N\_G17

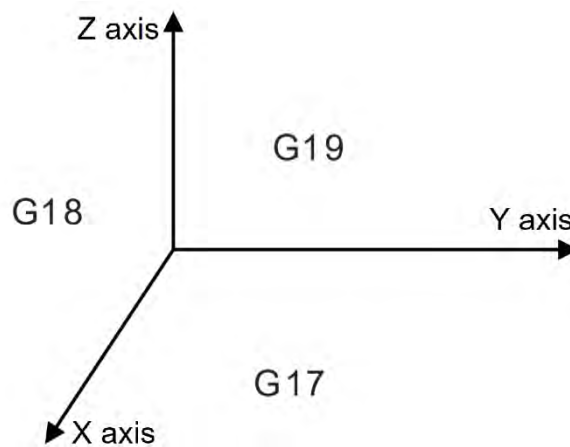
N\_G18

N\_G19

- Parameter Explanation:

N\_: The row number of G code in NC program

The figure of the planes is shown as follows:



### 22.1.4.7 G50 (Precise Stop)

- Function:

Change the transition mode into the precise stop. And afterwards the transition mode is always the precise stop. G51/G52 can be used for the transition mode switch in the execution. The terminal actuator will reduce its speed to 0 between G codes.

- Format:

N\_ G50

- Parameter Explanation:

N\_: The row number of G code in NC program

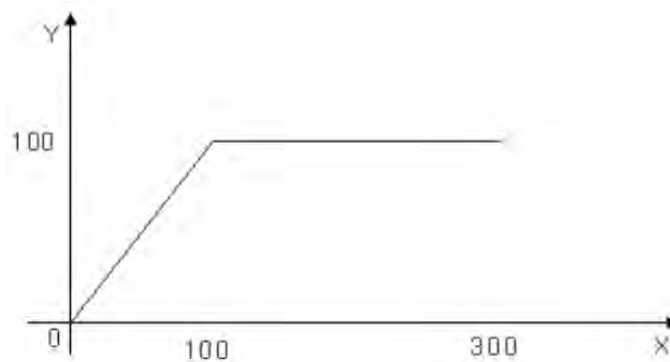
- Example:

N00 G50

N01 G1 X100 Y100

N02 G1 X300 Y100

After G codes are executed, the Y/X curve for the whole movement process is shown below:



### 22.1.4.8 G51 (Round Path Transition)

- Function:

G51 changes the transition mode into the arc transition. And afterwards the transition mode is always the arc transition. G50/G52 can be used for the switch of transition modes in the execution process. The terminal actuator will not reduce its speed between G codes and the transition curve is an arc. In this mode, the speed of the first G code prevails as the entire motion speed and F can not be used for changing the speed in the motion. VelOverride parameter of DMC\_CartesianCoordinate instruction can be set to control the speed of terminal actuator.

- Format:

N\_ G51 D\_

- Parameter Explanation:

N\_ : The row number of G code in NC program

D\_ : Radius of the arc

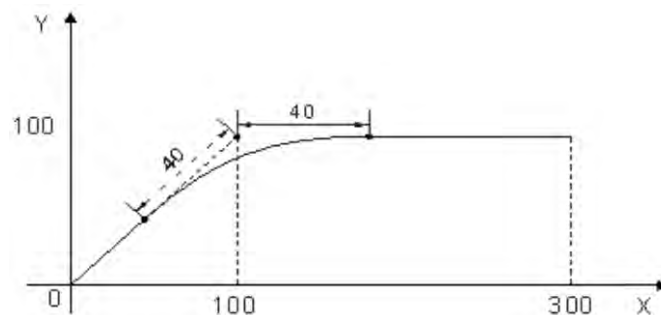
- Example:

```
N00 G51 D40
```

```
N01 G1 X100 Y100
```

```
N02 G1 X300 Y100
```

After G codes are executed, the Y/X curve for the whole movement process is shown below:



### 22.1.4.9 G52 (Smooth Path Transition)

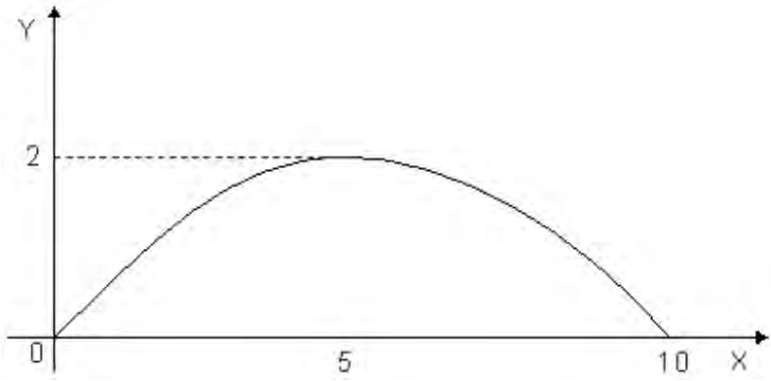
- **Function:**  
 G52 changes the transition mode into the smooth path transition and then the later transition is always the smooth path transition. G50/G51 can be used for the switch of transition modes in the execution. The terminal actuator will not reduce its speed between G codes. It is suitable for continual interpolation of small segments. In this mode, the speed of the first G code prevails as the entire motion speed and F can not be used for changing the speed in the motion. VelOverride parameter of DMC\_CartesianCoordinate instruction can be set to control the speed of the terminal actuator.
- **Format:**  
 N\_ G52
- **Parameter Explanation:**  
 N\_ : The row number of G code in NC program
- **Example:**

N00 G52	N01 G1 X0 Y0 E5 E-5 F5
N02 G1 X0.1 Y0.06282151816	N03 G1 X0.2 Y0.1255810391
N04 G1 X0.3 Y0.1882166266	N05 G1 X0.4 Y0.2506664671
N06 G1 X0.5 Y0.3128689301	N07 G1 X0.6 Y0.3747626292
N08 G1 X0.7 Y0.4362864828	N09 G1 X0.8 Y0.4973797743
N10 G1 X0.9 Y0.5579822121	N11 G1 X1 Y0.6180339887
N12 G1 X1.1 Y0.6774758405	N13 G1 X1.2 Y0.7362491054
N14 G1 X1.3 Y0.7942957813	N15 G1 X1.4 Y0.8515585831
N16 G1 X1.5 Y0.9079809995	N17 G1 X1.6 Y0.9635073482
N18 G1 X1.7 Y1.018082832	N19 G1 X1.8 Y1.07165359
N20 G1 X1.9 Y1.124166756	N21 G1 X2 Y1.175570505
N22 G1 X2.1 Y1.225814107	N23 G1 X2.2 Y1.274847979
N24 G1 X2.3 Y1.322623731	N25 G1 X2.4 Y1.369094212
N26 G1 X2.5 Y1.414213562	N27 G1 X2.6 Y1.457937255
N28 G1 X2.7 Y1.500222139	N29 G1 X2.8 Y1.541026486
N30 G1 X2.9 Y1.580310025	N31 G1 X3 Y1.618033989
N32 G1 X3.1 Y1.654161149	N33 G1 X3.2 Y1.688655851
N34 G1 X3.3 Y1.721484054	N35 G1 X3.4 Y1.75261336

N36 G1 X3.5 Y1.782013048	N37 G1 X3.6 Y1.809654105
N38 G1 X3.7 Y1.835509251	N39 G1 X3.8 Y1.859552972
N40 G1 X3.9 Y1.881761538	N41 G1 X4 Y1.902113033
N42 G1 X4.1 Y1.920587371	N43 G1 X4.2 Y1.937166322
N44 G1 X4.3 Y1.951833524	N45 G1 X4.4 Y1.964574501
N46 G1 X4.5 Y1.975376681	N47 G1 X4.6 Y1.984229403
N48 G1 X4.7 Y1.991123929	N49 G1 X4.8 Y1.996053457
N50 G1 X4.9 Y1.999013121	N51 G1 X5 Y2
N52 G1 X5.1 Y1.999013121	N53 G1 X5.2 Y1.996053457
N54 G1 X5.3 Y1.991123929	N55 G1 X5.4 Y1.984229403
N56 G1 X5.5 Y1.975376681	N57 G1 X5.6 Y1.964574501
N58 G1 X5.7 Y1.951833524	N59 G1 X5.8 Y1.937166322
N60 G1 X5.9 Y1.920587371	N61 G1 X6 Y1.902113033
N62 G1 X6.1 Y1.881761538	N63 G1 X6.2 Y1.859552972
N64 G1 X6.3 Y1.835509251	N65 G1 X6.4 Y1.809654105
N66 G1 X6.5 Y1.782013048	N67 G1 X6.6 Y1.75261336
N68 G1 X6.7 Y1.721484054	N69 G1 X6.8 Y1.688655851
N70 G1 X6.9 Y1.654161149	N71 G1 X7 Y1.618033989
N72 G1 X7.1 Y1.580310025	N73 G1 X7.2 Y1.541026486
N74 G1 X7.3 Y1.500222139	N75 G1 X7.4 Y1.457937255
N76 G1 X7.5 Y1.414213562	N77 G1 X7.6 Y1.369094212
N78 G1 X7.7 Y1.322623731	N79 G1 X7.8 Y1.274847979
N80 G1 X7.9 Y1.225814107	N81 G1 X8 Y1.175570505
N82 G1 X8.1 Y1.124166756	N83 G1 X8.2 Y1.07165359
N84 G1 X8.3 Y1.018082832	N85 G1 X8.4 Y0.9635073482
N86 G1 X8.5 Y0.9079809995	N87 G1 X8.6 Y0.8515585831
N88 G1 X8.7 Y0.7942957813	N89 G1 X8.8 Y0.7362491054
N90 G1 X8.9 Y0.6774758405	N91 G1 X9 Y0.6180339887
N92 G1 X9.1 Y0.5579822121	N93 G1 X9.2 Y0.4973797743
N94 G1 X9.3 Y0.4362864828	N95 G1 X9.4 Y0.3747626292

N96 G1 X9.5 Y0.3128689301	N97 G1 X9.6 Y0.2506664671
N98 G1 X9.7 Y0.1882166266	N99 G1 X9.8 Y0.1255810391
N100 G1 X9.9 Y0.06282151816	N101 G1 X10 Y0

After G codes are executed, the Y/X curve for the whole movement process is shown below:



22

### 22.1.4.10 G90 (Absolute Mode)

- Function:

After G90 is executed, the terminal position of each axis in G code is based on 0 unit and G91 can be used to switch into the relative mode in the execution process. It is absolute mode for NC program by default.

- Format:

N\_G90

- Parameter Explanation:

N\_ : The row number of G code in NC program

- Example:

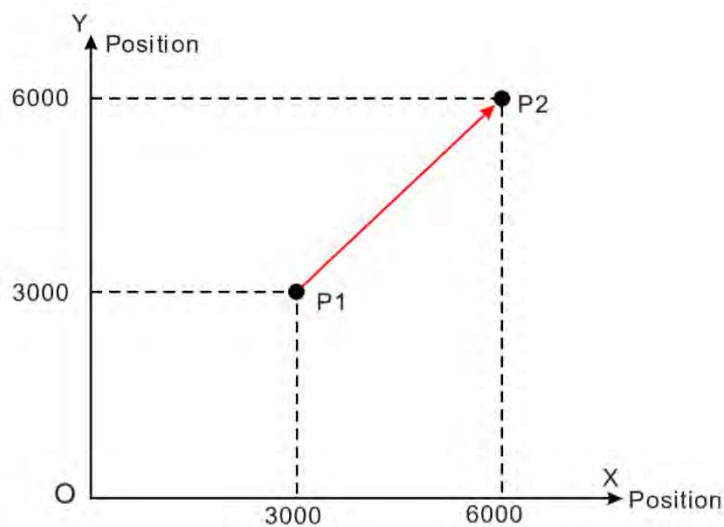
The initial positions of axis X and Y are both 3000 units and the axis parameters are both default values.

The G codes to be executed are as follows:

```
N00 G90
```

```
N01 G0 X6000 Y6000
```

After G codes are executed, the Y/X curve for the whole movement process is shown below:



### 22.1.4.11 G91 (Relative Mode)

- Function:

After G91 is executed, the terminal position of each axis in G code is counted in the incremental method beginning from the current position and G90 can be used to switch into the absolute mode in the process.

- Format:

N\_G91

- Parameter Explanation:

N\_: The row number of G code in NC program

- Example:

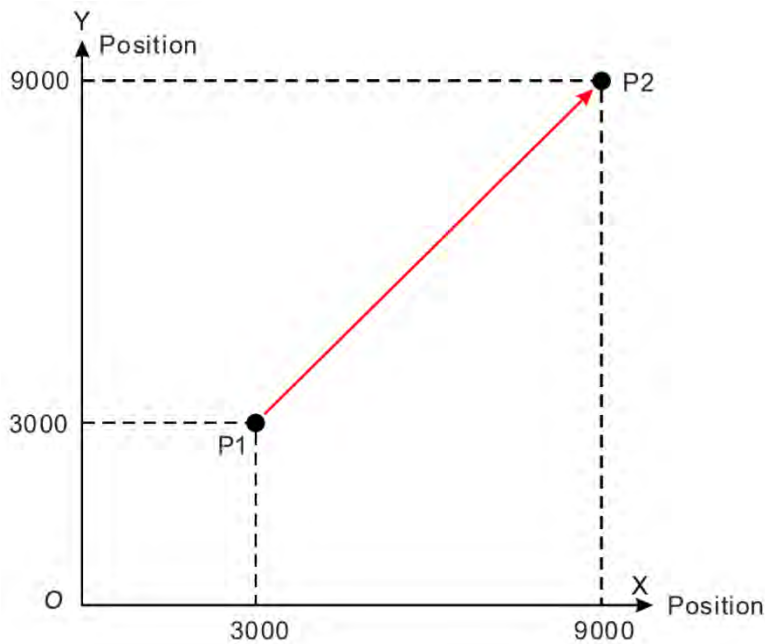
The initial positions of axis X and Y are both 3000 units and the axis parameters are both default values.

The G codes to be executed are as follows:

```
N00 G91
```

```
N01 G0 X6000 Y6000
```

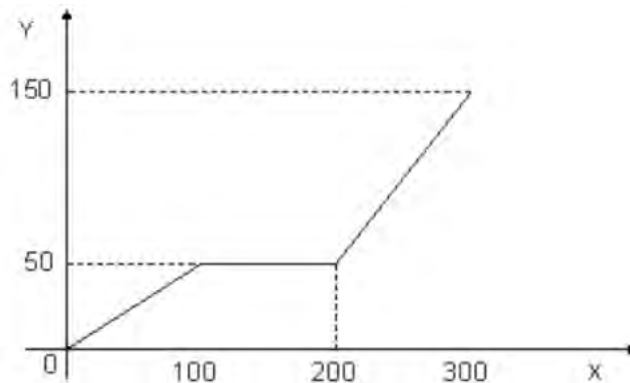
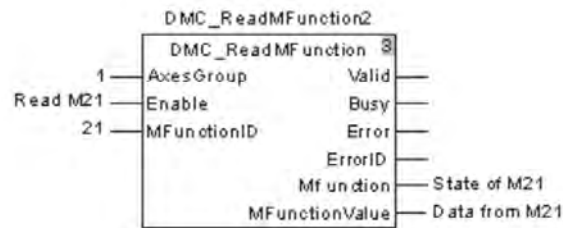
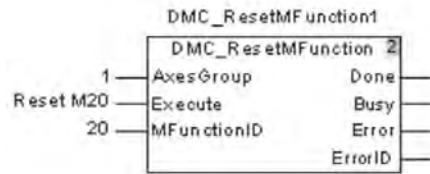
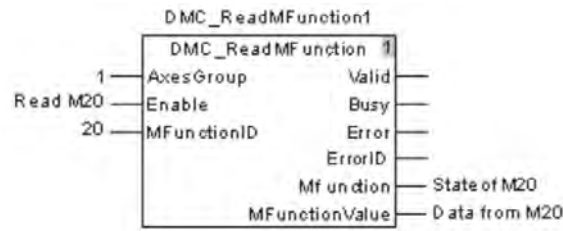
After G codes are executed, the Y/X curve for the whole movement process is shown below:





### 22.1.4.12 M Code

- Function:  
Interacts with general programs.
- Format:  
N\_ M\_ D\_
- Instruction Explanation:
  1. D\_ can be omitted and then no parameter is output after M code is executed.
  2. Two methods of using M code: one is to write it outside the row of G code; the other is to write it in the row of G code.
  3. When M code and G code are not in the same row, e.g. N0 M10 D10.02 ; N1 G1 X100 Y100. When arriving at the row N0, G code execution will stop. Meanwhile, DMC\_ReadMFunction is used to read the state of M code, MFuntion is TRUE and the value of MFunctionValue is 10.02. After M code is reset by using DMC\_ResetMFunction instruction, G code execution will continue and then N1 row will be executed.
  4. When M code and G code are in the same row and M code can only be placed after G code, e.g. N0 G1 X100 M10 D10.02. When the execution arrives at N0 row, G1 is executed. Meanwhile, DMC\_ReadMFunction is used to read the state of M code, MFuntion is TRUE and the value of MFunctionValue is 10.02. After G1 execution is finished the following execution will continue without the resetting of M code by the reset instruction.
- Example:  
N0 G1 X100 Y50  
N1 M20  
N2 G1 X200 Y50 M21 D3.14  
N3 G1 X300 Y150  
After G codes are executed, the Y/X curve for the whole movement process is shown below:

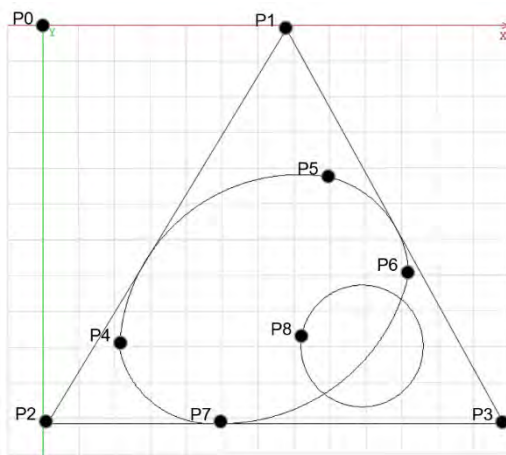


When the two variables “Read M20” and “Read M21” are TRUE, G code execution starts and the terminal actuator stops at the position (100 · 50). At the moment, the variable “State of M20” is TRUE. After the execution of other actions is done, the variable “Reset M20” changes to TRUE and M code is reset. Then G code execution continues. The terminal actuator starts to move to (200, 50). Meanwhile, the variable “state of M21” changes to TRUE and the value of “Data from M21” variable is 3.14. The terminal actuator will not stop at that time. After reaching the position (200, 50), it will keep moving till the position (300, 150) is reached and then the execution is completed.

### 22.1.5 G-Code Example of Drawing Delta Logo through Three-Axis Motion

Through G Codes and absolute addressing, edit and draw the paths to form Delta logo.

[Delta logo to be drawn]



[G-Code Program]

```

0001 N0 G90 (P0)
0002 N1 G0 Z5000
0003 N2 G0 X32500 Y-500 (P1)
0004 N3 G0 Z0
0005 N4 G1 X600 Y-53400 F10000 (P2)
0006 N5 G1 X61500 Y-53400 F10000 (P3)
0007 N6 G1 X32500 Y-500 f10000 (P4)
0008 N7 G0 Z5000
0009 N8 G0 X23100 Y-53400 (P5)
0010 N9 G0 Z0
0011 N10 G2 X10300 Y-43600 R12600 (P6)
0012 N11 G2 X38440 Y-20500 R23900 (P7)
0013 N12 G2 X48880 Y-33300 R13900 (P8)
0014 N13 G2 X23100 Y-53400 R25600 (P5)
0015 N14 G0 Z5000
0016 N15 G0 X34500 Y-43000 (P9)
0017 N16 G0 z0
0018 N17 G2 X34500 Y-43000 I8150 J0 F5000 (P9)
0019 N18 G0 z5000
0020 N19 G0 x0 y0 (P0)
0021

```

**[Path Description]**

Steps to execute the program:

Step 1: Lift the tool in the direction of Z axis and then move it from P0 to P1.

Step 2: Lower the tool until P1 point in the direction of Z axis, move it from P1 to P2, from P2 to P3 and from P3 to P1 and then lift the tool in the direction of Z axis to finish the drawing of the triangle.

Step 3: Move the tool from P1 to P4 and then lower the tool until P4 point in the direction of Z axis. Subsequently move it in the way from P4 to P5, from P5 to P6, from P6 to P7 and from P7 to P4 and then lift the tool in the direction of Z axis to finish the drawing of the oval shape.

Step 4: Move the tool from P4 to P8, lower the tool until P8 point in the direction of Z axis, draw a circle straight from P8 to P8 and then lift the tool in the direction of Z axis to finish the drawing of the circle.

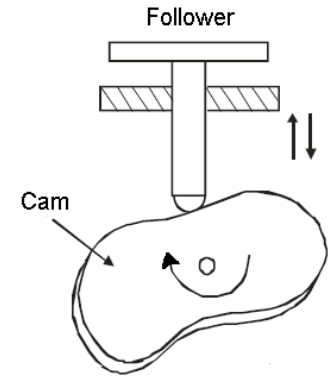
Step 5: Move the tool to the home point P0 to complete the drawing of the entire Delta logo.



## 22.2 E-CAM Editor

### 22.2.1 About E-CAM (Electronic CAM)

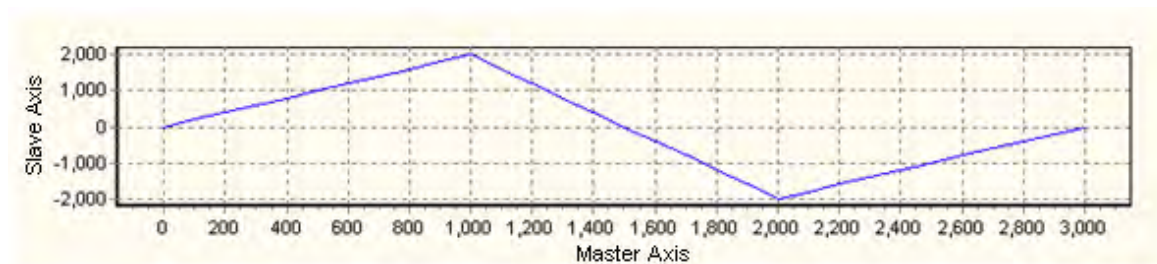
A mechanical cam is an input object with irregular shape. It makes a follower move regularly by coming into contact with the follower. A traditional mechanical cam is composed of a cam, a follower, and a support. As the image shown, when a cam rotates, the follower will go move according to the shape of the cam. For an operation of motion control, the relative movements between axes are achieved by the rotation of a cam.



Use a cam chart to define the relation between a follower and a cam to simulate the movements of a cam controlled by a PLC, making a slave axis to move by the master axis according to their defined relation. The benefit of using a E-CAM is that users can modify the electronic cam data in an electronic cam in software. Users do not need to modify the mechanical design and no depreciation costs.

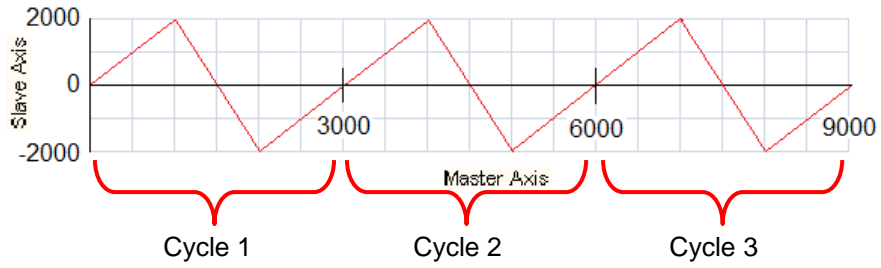
### 22.2.2 The Significance and Description of E-CAM

A created E-CAM chart is as the image shown below. In the cam chart, the horizontal axis represents the master axis used and the vertical axis represents the slave axis used. Values (such as number of pulses sent) on the cam chart are all single digits. The range of 0-3000 on the horizontal axis represents an E-CAM cycle and the vertical axis represents the units of the relative slave axis outputs.



If an electronic cam operates cyclically, the slave axis of the electronic cam moves in accordance with electronic cam data when the master axis of the electronic cam moves. After the master axis completes an electronic cam cycle (3000 units sent), the number of pulses that the master axis sends will continue to increase. However, after the slave axis completes an electronic cam cycle, it will repeat the electronic cam cycle. The relation between the positions of a master axis and the positions of a slave axis is the repeated extension of electronic cam data.

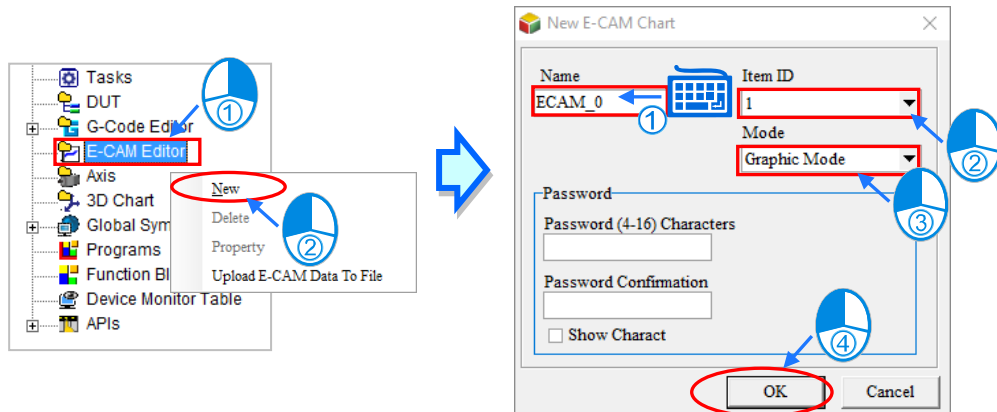
If a motion controller is connected to a master servo motor and a slave servo motor, the master servo motor will be regarded as a drive shaft rotating in a direction, and the slave servo motor will be regarded as a processing shaft which rotates back and forth. This characteristic can be applied to flyingsaws and flyingcuts.



22

### 22.2.3 Using E-CAM Editor

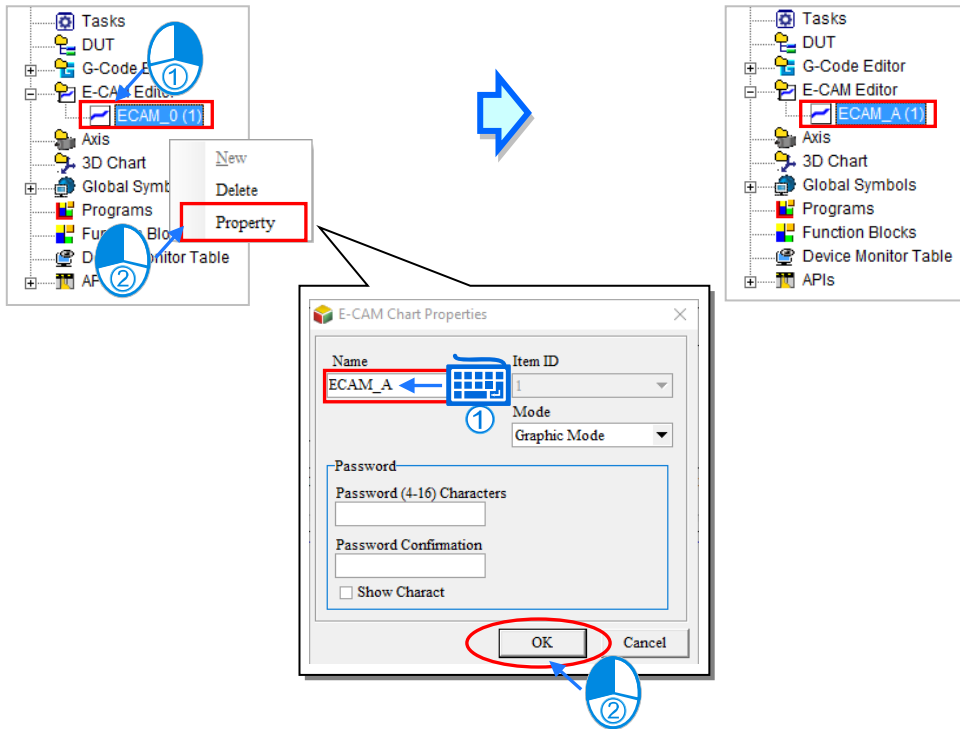
Right-click the E-CAM Editor in the project management area and then click **New** to see a New E-CAM Chart setting window. Type a name and select an item ID and a mode. Users can also set up a password to protect the E-CAM data. After a password is set, the system will ask for the password upon next editing when the mode is set to **Graphic Mode**. When the mode is set to **File mode**, the password will be asked for upon each import and export.



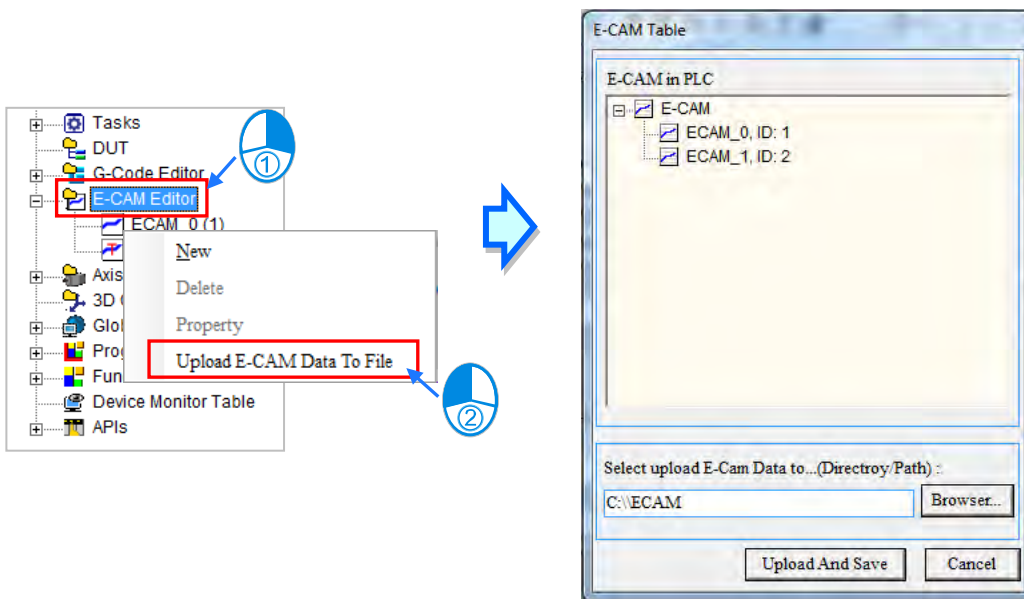
22

Name	Function Description	AS	AHxxEMC	DVPxxMC
Item ID	E-CAM table item ID	Max.16	Max.32	Max.32
ASDA	Using ASDA file format in	V		
File Mode	E-CAM table in file mode		V	
Graphic	E-CAM table in graphic		V	V
Flag Cycle	When dragging the first and last control point, data synchronization of the two			V
Password	Password for E-CAM table		V	V
Upload E-CAM to	Upload master axis and slave axis position from PLC		V	

Right-click on the created E-CAM item and choose Delete to remove the item; or select Property to reset item name or password but the item ID cannot be changed.

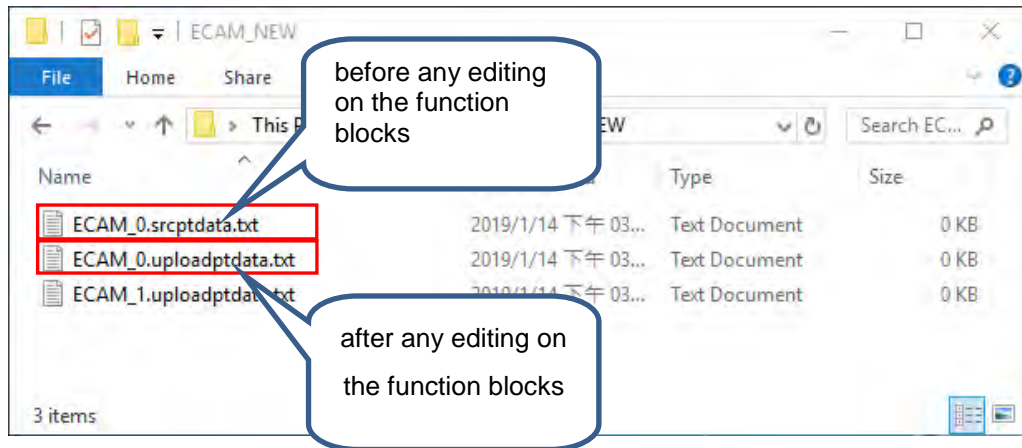


Use the AH-Motion function block in Delta library, e.g. DFB\_CamWire to modify E-Cam graph point position online and then implement **Upload E-CAM Data to File** function to save the master axis and slave axis position in E-CAM table in another file. The function requires connection with AHxxEMC hosts first then right-click **Upload E-CAM Data to File** in **E-CAM Editor**. Meanwhile, the **E-CAM in PLC** window will appear.



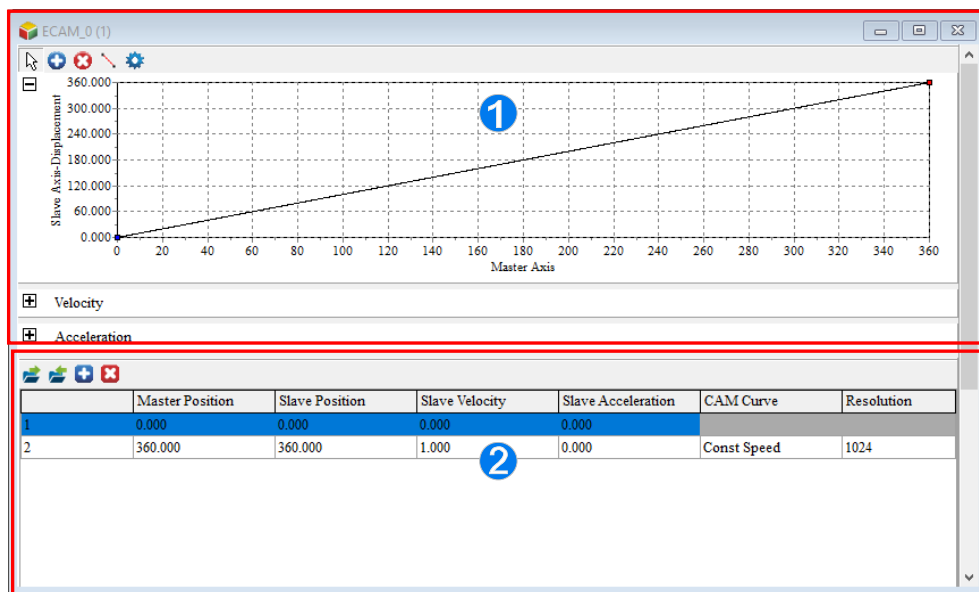


When the directory/path for file saving is selected, click **Upload and Save**. Two files, one is the xxxx\_scrptdata.txt that appears in the directory before any editing on the function blocks and xxxx\_uploadptdata.txt that appears after any editing on the function blocks. The xxxx is the E-CAM name (see below).



22

When selecting **Graphic Mode**, it will open E-CAM edit window once E-CAM graph is created. The description are as followed.






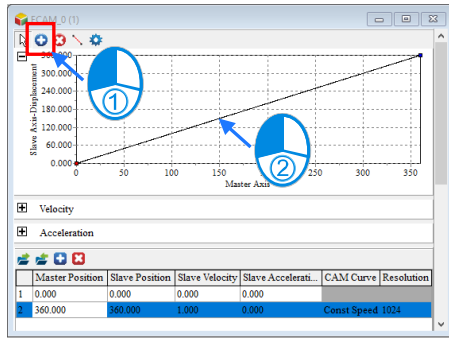
- 1 E-CAM curve area: the chart is created according to the position, velocity and acceleration of the E-CAM. Users can also use their mouse to move the points directly in the chart or use the function buttons to add new data points or sections on the created chart.
- 2 Data setting area: users can use the function buttons to add, edit, or delete the created charts. The data in the Data setting area and the E-CAM chart will be synchronized with one another.

Explanation of E-CAM Editor function icons (see below).

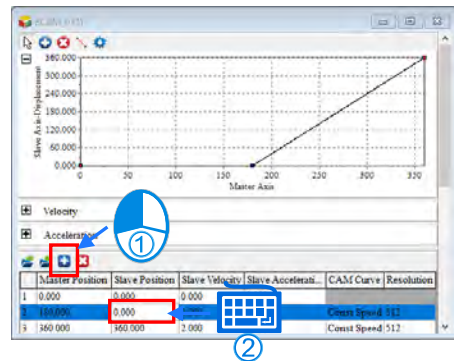
Icon	Name	Description	AS	AHxxEMC	DVPxxMC
	Select tools	Switch to the selected tool	V	V	V
	Add a key point	Add a key point in the diagram	V	V	V
	Delete a key point	Delete a key point in the diagram	V	V	V
	Add line segment	Add a line segment that include two key points in the diagram.		V	V
	Setting	Open setting window	V	V	V
	Export	Export E-CAM in corresponding format	.csv	.dst	.dst
	Import	Import E-CAM in corresponding format	.csv	.dst	.dst
	Upload slave axis data	Upload data from data registers/servers	V		
	Download slave axis data	Download data to data registers/servers	V		
	Add a row	Add a key point data in the planning area	V	V	V
	Delete a row	Delete a key point data in the planning area	V	V	V
	Add tappet point	Add a tappet point icon  on the curve			V
	Display CAM table	Display CAM table			V
	Display tappet point table	Display tappet point table			V

22

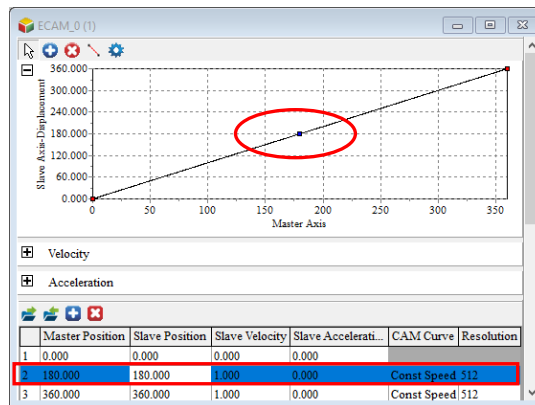
Click  to add new point then click on the diagram for the added key point position. Meanwhile, the section below will automatically add the key point data or select  in the planning section and input master axis data as 180 (see below). To add several key points, click  and then click several times on the diagram to add the key points data.






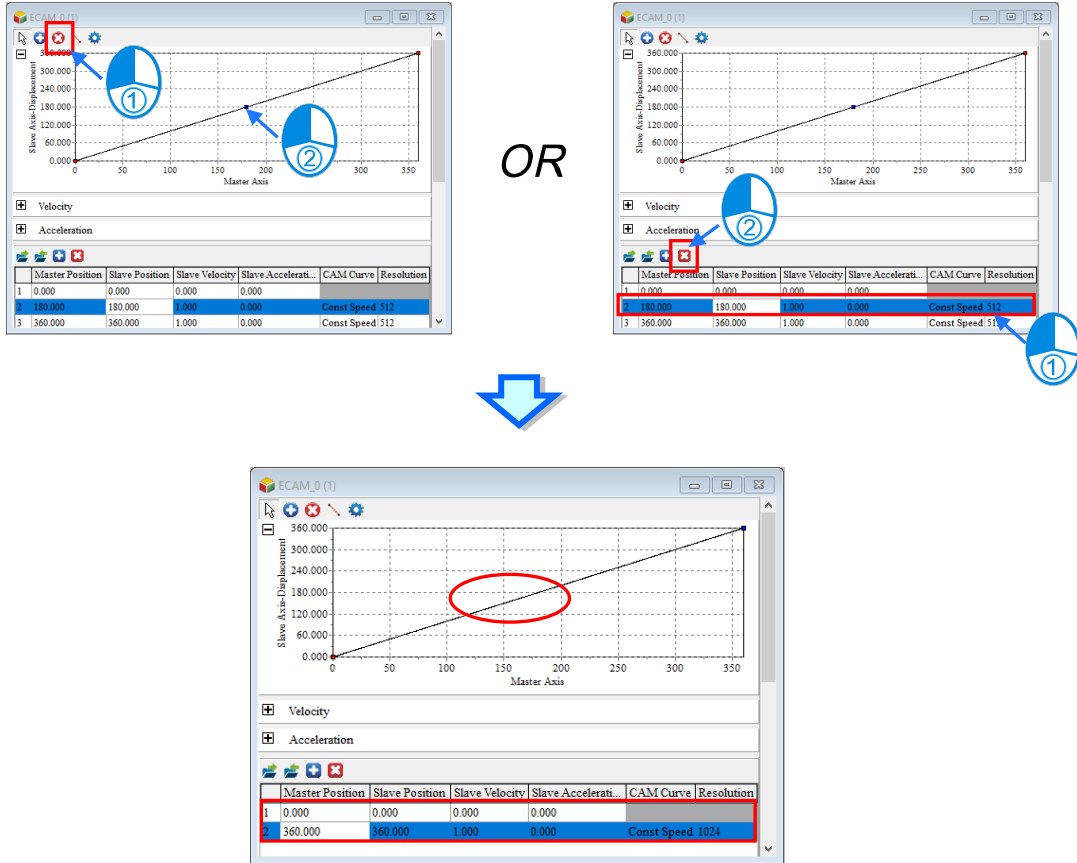
OR



22



Users can click  then select the key point to delete on the diagram and the section below will automatically delete the key point data or select  in the planning section to delete the key point data (see below). To delete several key points, click  and then continuously click on the key points to delete.



22


During E-CAM execution, the CAM may trigger corresponding action in certain specific position and this requires a close/open state setting regarding the assigned position of CAM. When CAM moves to that position, the close/open state change can control the corresponding action and the close/open state is called CAM-tappet.

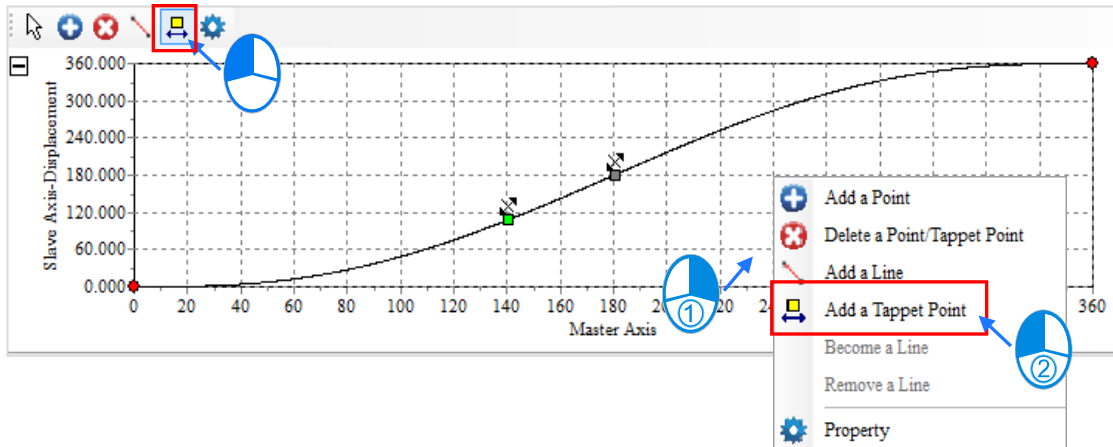
The CAM-tappet can set in any position of CAM and one CAM can set up a maximum of 128 CAM-tappet. When CAM moves across in positive or negative direction, the tappet state can be configured into (close, set, reset, reverse). The tappet state can be obtained through the corresponding command of CAM-tappet in the controller program. The following chart describes the CAM-tappet state:

Close	When CAM passes the corresponding position of tappet, the tappet state does not change.
Set	When CAM passes the corresponding position of tappet, the tappet state is set to TRUE.
Reset	When CAM passes the corresponding position of tappet, the tappet state is set to FALSE.

Reverse	When CAM passes the corresponding position of tappet, the tappet state switch once between TRUE and FALSE. (If TRUE then change to FALSE, if FALSE then change to TRUE).
---------	--

Note: CAM-tappet state will reset once the next CAM cycle starts.


Use the add tappet point icon  on the toolbar or right-click on the diagram to choose add a tappet point (see below).

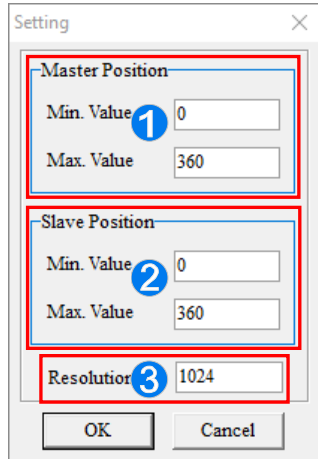


Each key point data in the planning section allows users to setup a CAM curve section. Then, several rows of key points data form the CAM curve and all the created key points data forms a E-CAM cycle. For each key point data, users can input the master and slave axis relations. The descriptions are shown below.


Name	Function	AS	AHxxEMC	DVPxxMC
Key Point	Record E-CAM	Max.721	Max.40	Max.40
CAM Curve Const Speed	Constant speed		V	
CAM Curve Const Acc.	Constant acceleration		V	
CAM Curve Single Hypot.	Single Hypothesis		V	
CAM Curve Cycloid	Cycloid		V	
CAM Curve Polynomial	Polynomial	V	V	V
CAM Curve B-Spline	B- Spline		V	

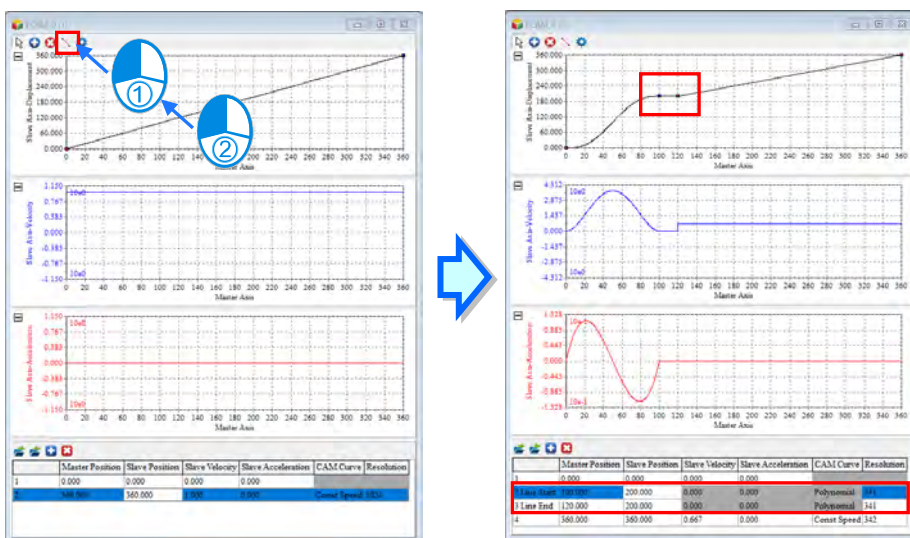
Name	Function	AS	AHxxEMC	DVPxxMC
Resolution	Used data points of the current section.	Auto-configuration	Max. 2047	Auto-configuration

Click Setting or  to open other setting windows. Explanations are as followed.

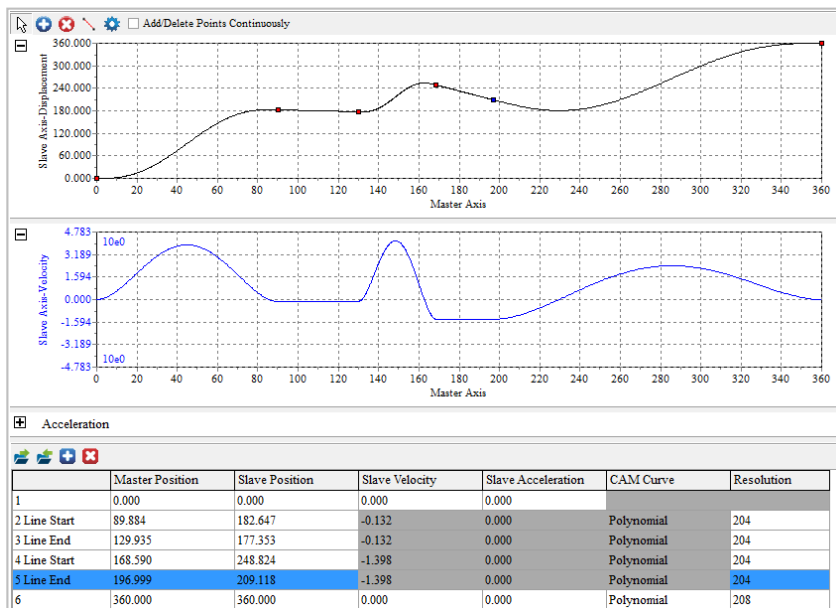
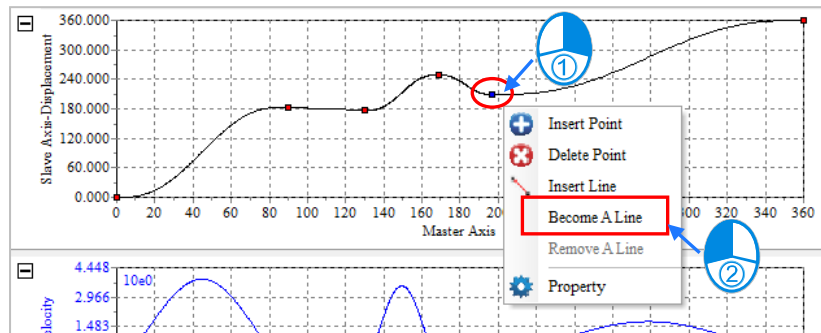
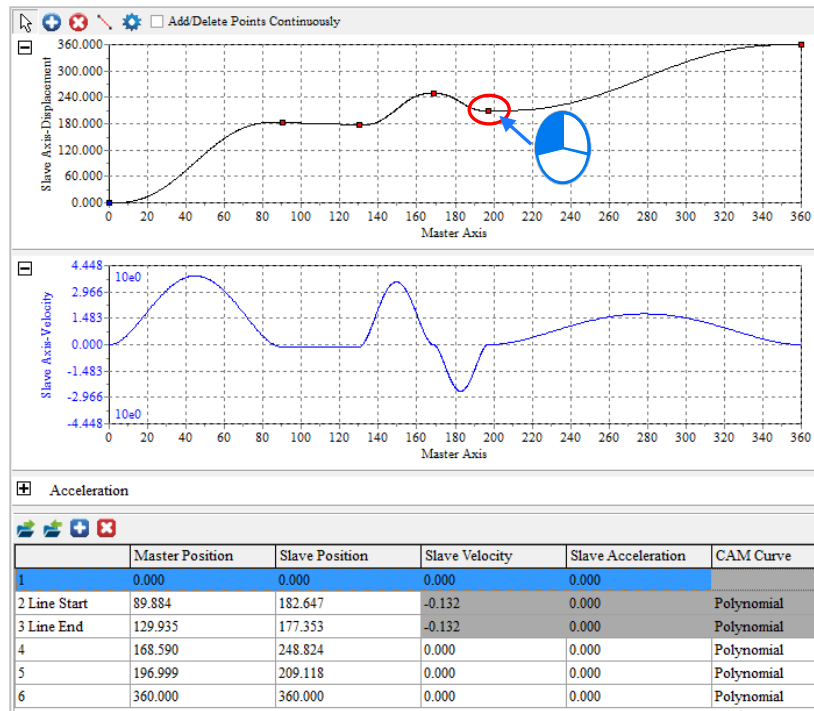


- ❶ Master position: Setting main axis display range in the diagram.
- ❷ Slave position: Setting slave axis display range in the diagram.
- ❸ Resolution: Setting the E-CAM table, the total interval curve resolution

Click **add line segment** or  and click on the position to add in the diagram and the section below will automatically add the linesegment containing two key points data. When the line segment is added, the function of the CAM curve with line segment added is fixed in Polynomial mode. On the left of the line segment, item ID appears and on the right, Line Start and Line End shows.



To set the interval line segment to into a line n Polynomial mode, right-click the key point in the diagram and then select **Become A Line**.

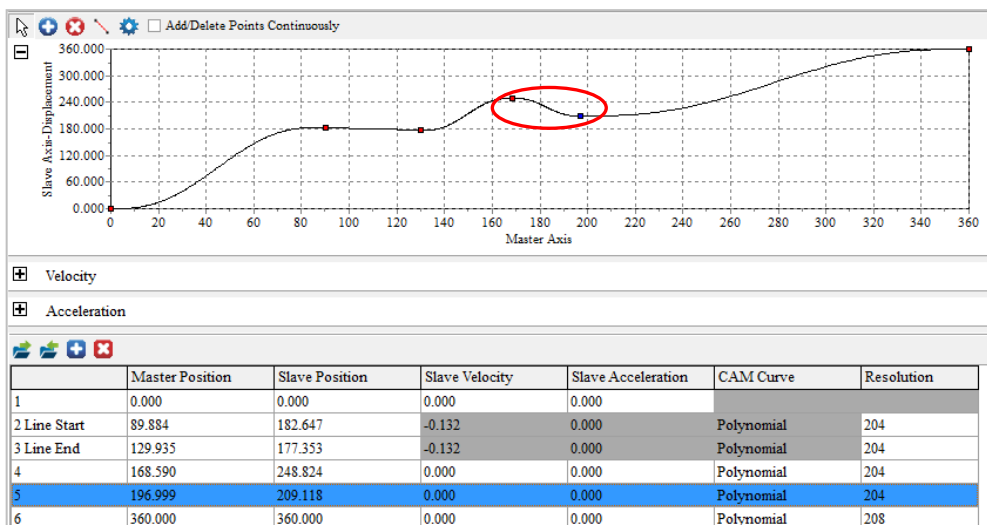
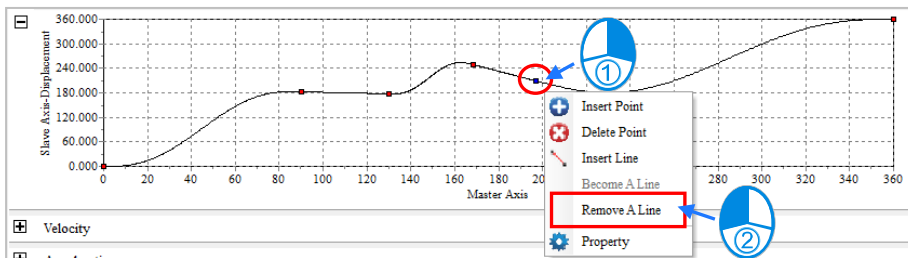
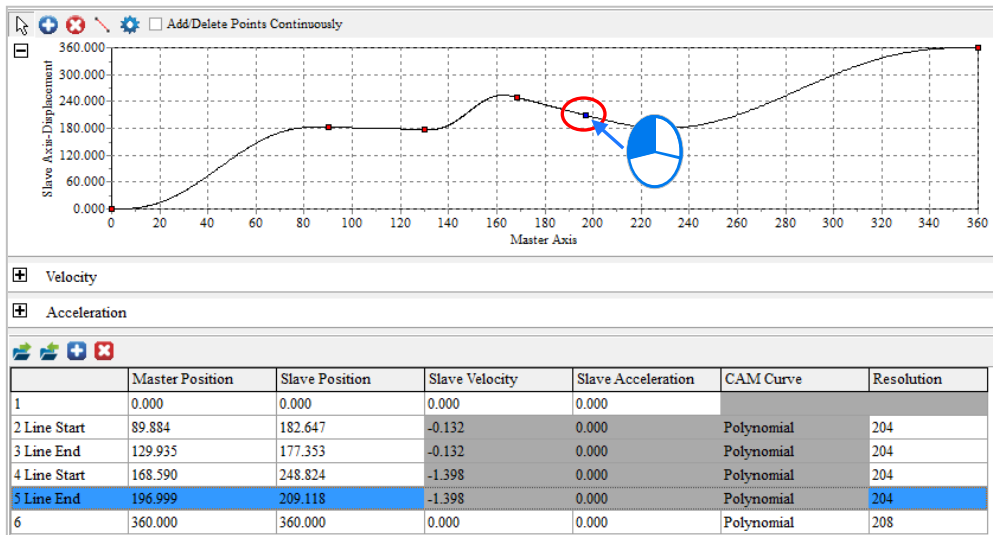


When setting to **Become A Line**, please take note:

- The previous and current section has to be polynomial to use this function.
- The key points of Line Start and Line End cannot share line property with other line intervals.

To remove the created line property, right-click the key point of Line Start or Line End of the polynomial interval line sections in the diagram and then choose **Remove A Line**.


22

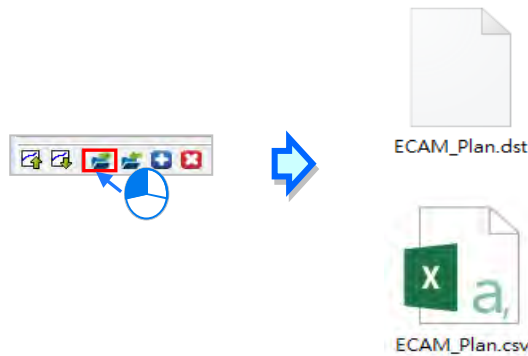





When using **Remove A Line** function, please take note:

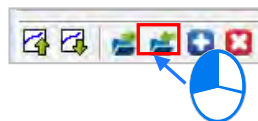
- Property regarding Line Start or Line End key points can use this function.

Click the upper toolbar icon  to export planning table data. However, the file format to export is different based on the project PLC types and cannot be in common use.

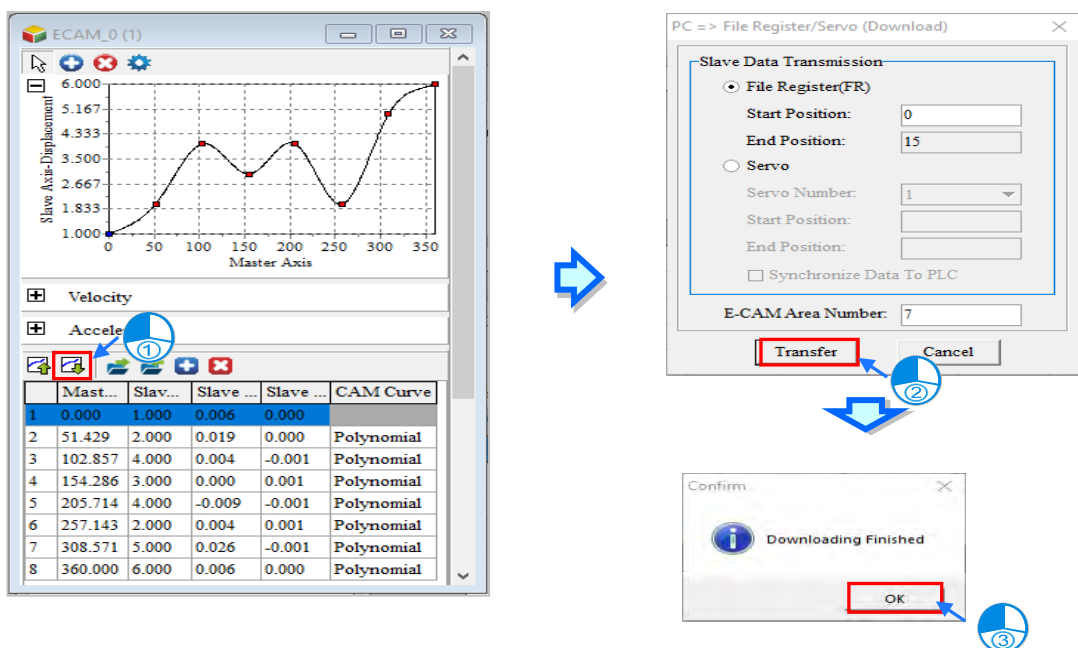


22

To import planning table data, click the icon  to import the selected files.

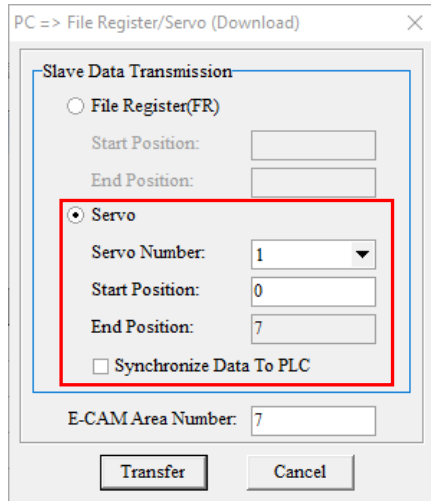


To download slave data, users need to setup connection between ISPSOft and AS series PLCs first. The following example demonstrates by opening an E-CAM table and increase up to 8 key points. Then, select **Download Slave Data [PC=> Data Register/Servo]** and choose to download to file register (FR) or servo from the pop-up window. The E-CAM area number is set to save the curve area. Here, we download 8 key points which equals to 7 curve area. When changing the start position or E-CAM area number, the system automatically calculates the ending position. Click Transfer and a pop-up window appears when transfer is complete.

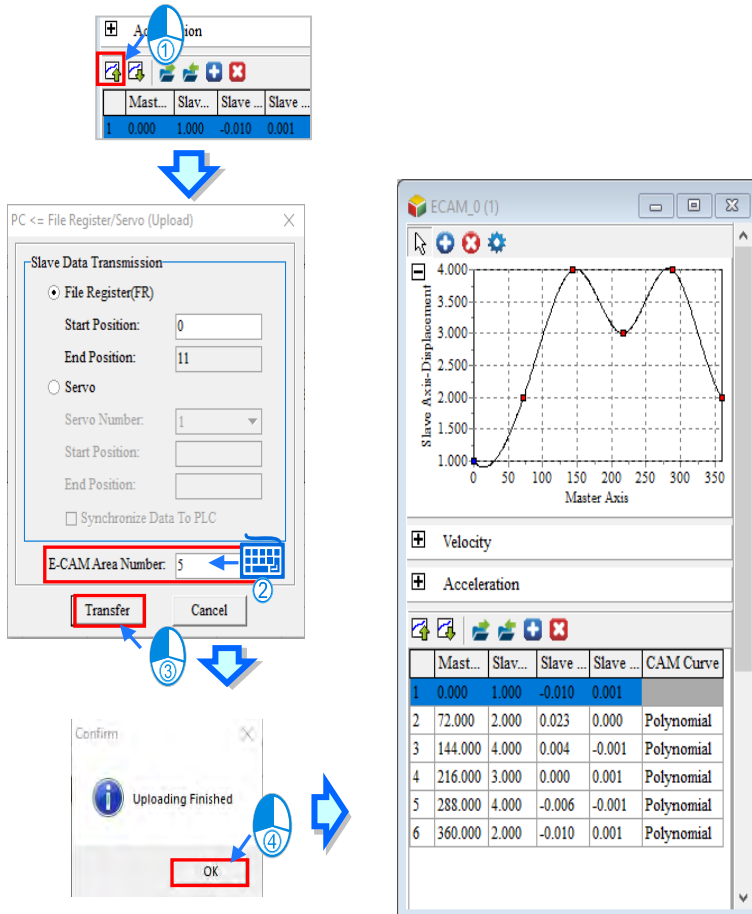


Click Servo to download the files to the servo. First, users need to select a corresponding servo number, then setup the start position. When Synchronization Data to PLC is selected, the current E-CAM data will synchronize into PLC, however, the E-CAM number and item ID need to be the same as in PLC for data synchronization.

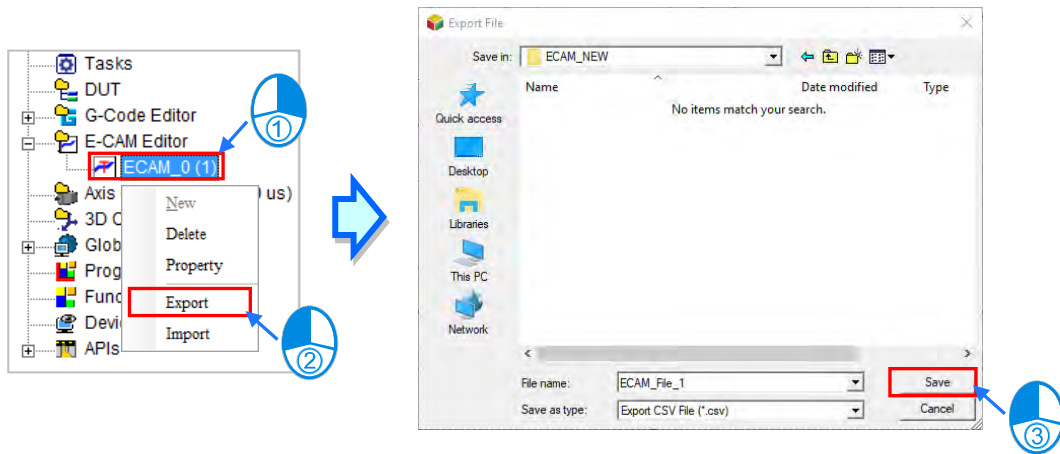
22



Click **Upload Slave Data [PC=> Data Register/Servo]** and in the pop-up window, we can upload 6 key points, so the E-CAM area number is changed to 5 and click Transfer to finish uploading. The following diagram shows that 6 key points are re-drawn on the E-CAM table.



When in **File Mode**, users need to export E-CAM table for edit. Right-click on the E-CAM item and choose **Export**. Select a path and file name, then click **Save** to complete exporting.



22

Input E-CAM master position (X) and slave position (Y) and format as 0 for the export file in csv format; input maximum of 2048 sets of XY position points and at least 2 sets of XY position points. When input is complete, click save and return to ISPSOft to execute import.

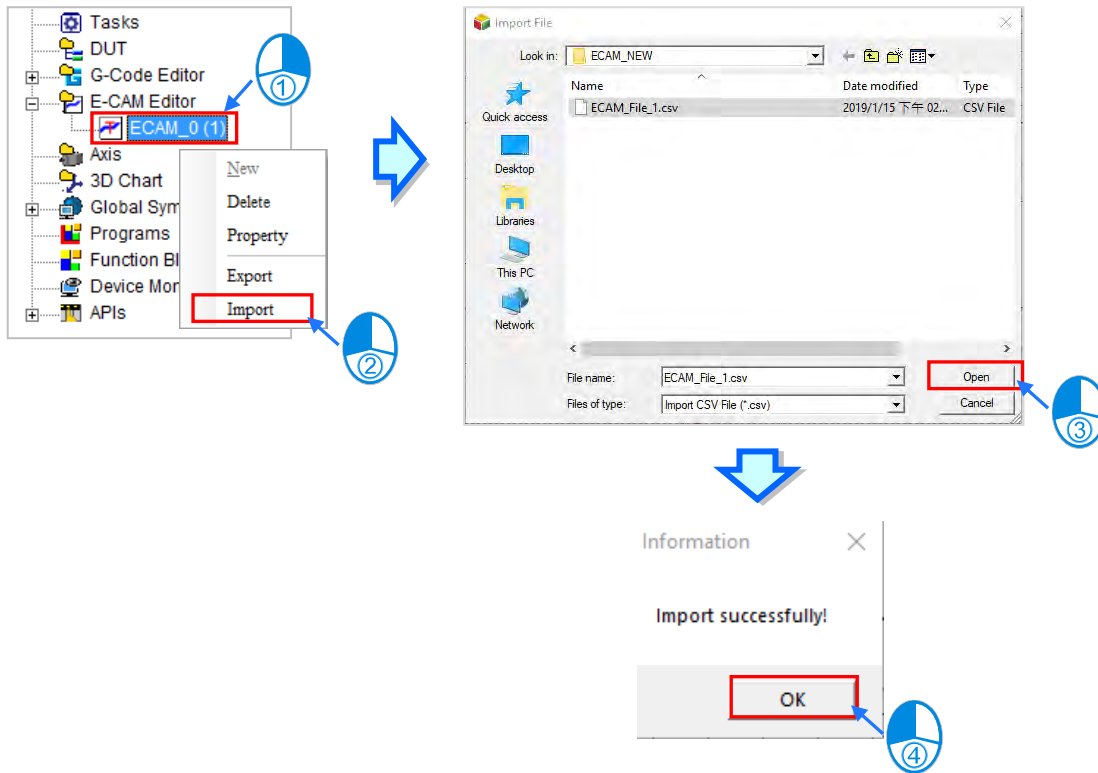
	A	B	C	D
1	:Delta	:E-Cam	Version	V1.0
2	X	Y		
3	0	0		
4	0.280561	0.804357		
5	0.561122	1.608714		
6	0.841682	2.413071		
7	1.122243	3.217427		
8	1.402804	4.021784		
9	1.683365	4.826141		
10	1.963925	5.630497		
11	2.244486	6.434855		
12	2.525047	7.239211		
13	2.805608	8.043568		

Master Position

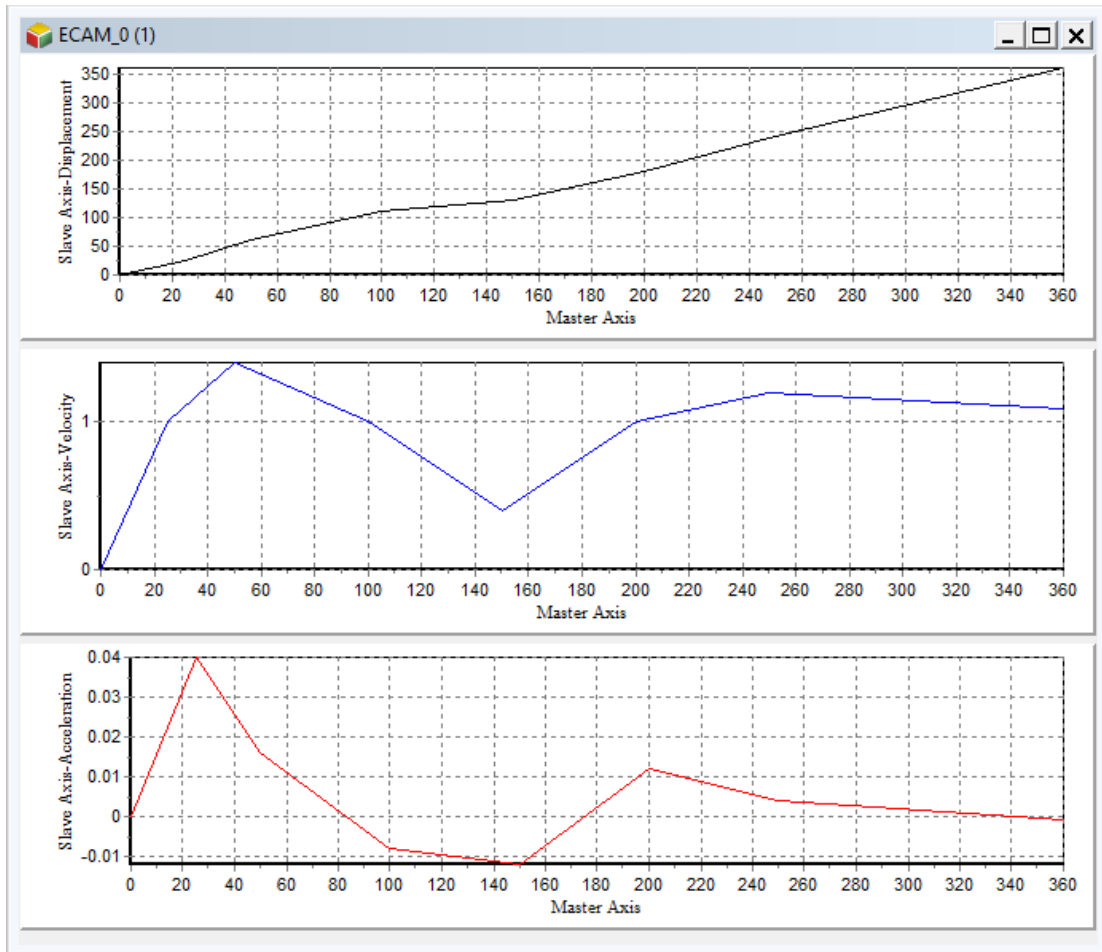
Slave Position

Once return to ISPSoft, right-click on the selected E-CAM item and choose **Import**. Then, select on the file to open and a pop-up window indicating successful import appears. Click **OK** to complete importing.

22

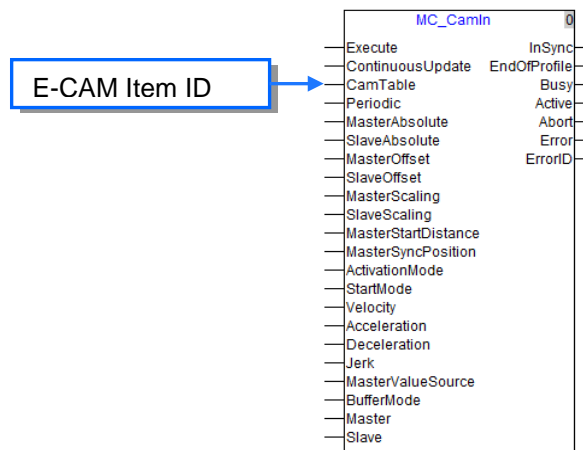


When importing is complete, the corresponding E-CAM diagram is shown to view graphic data regarding imported files.

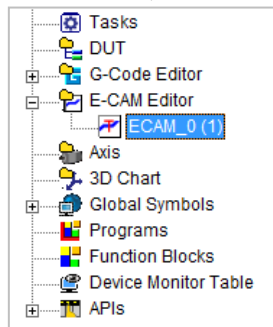
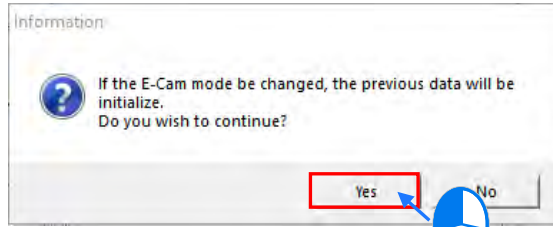


22

When finish programming E-CAM Editor, it can be used together with motion control commands or function blocks in POU. E.g. MC\_CamIn command enables assigned E-CAM item according to E-CAM Editor item ID. For more information on using MC\_CamIn command, please refer to the programming manuals of related PLC types.



When switching to E-CAM graphic mode, click OK and an Information window appears asking “If the E-CAM mode is changed, the previous data will be initialized. Do you wish to continue? ” Choose **Yes** and all E-CAM data is initialized.



22

# Chapter 23 Wizard Tool

## Table of Contents

<b>23.1 Position Planning Table .....</b>	<b>23-2</b>
23.1.1 About the Position Planning Table .....	23-2
23.1.2 Setting the Position Planning Table.....	23-2
23.1.3 Positioning Table Simulation.....	23-6
23.1.4 Upload and Download Position Planning Table.....	23-8
23.1.5 Control Mode – Single axis point-to-point motion .....	23-9
23.1.6 Control Mode – Single axis multi-segment.....	23-10
23.1.7 Control Mode - 2-axis linear interpolation motion .....	23-11
23.1.8 Control Mode - 2-axis Arc Interpolation Motion .....	23-12
<b>23.2 Data Tracer .....</b>	<b>23-14</b>
23.2.1 About Data Tracer .....	23-14
23.2.2 Opening the Data Tracer Window .....	23-14
23.2.3 Sample Parameter Settings .....	23-16
23.2.4 Sampling Modes .....	23-19
23.2.5 Measurement .....	23-21
<b>23.3 Data Logger.....</b>	<b>23-25</b>
23.3.1 About Data Logger .....	23-25
23.3.2 Opening the Data Logger Window .....	23-25
23.3.3 Sample Parameter Settings.....	23-27
23.3.4 Watch and Record.....	23-29
23.3.5 Measurement .....	23-32
<b>23.4 High Speed Counter.....</b>	<b>23-36</b>
23.4.1 About High Speed Counter .....	23-36
23.4.2 Using High Speed Counter.....	23-36

## 23.1 Position Planning Table

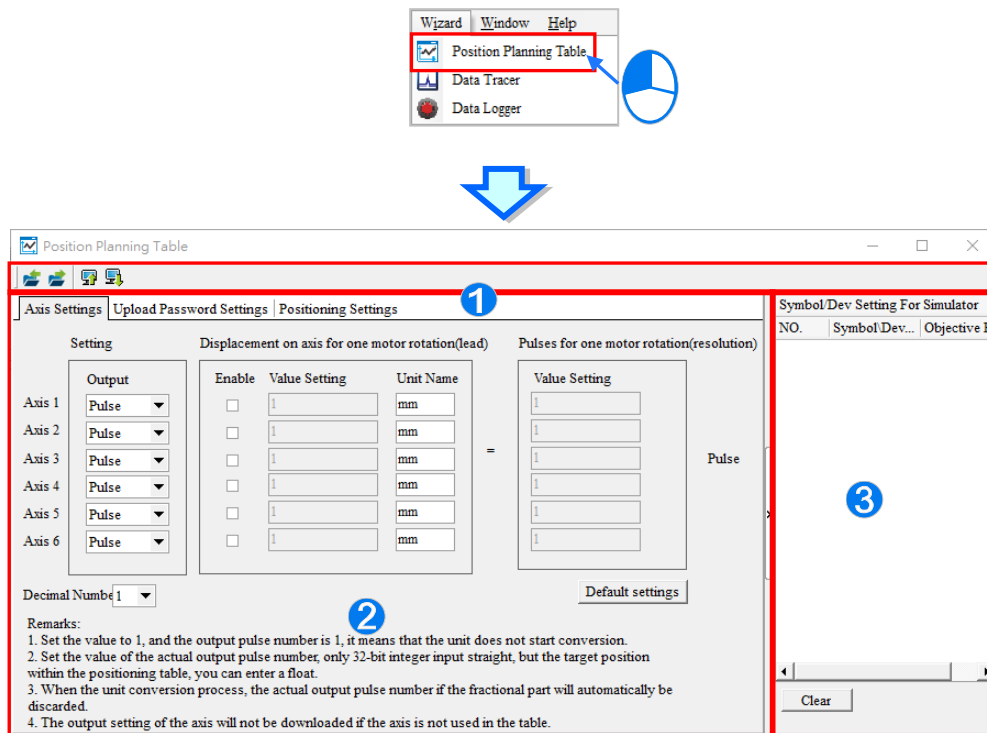
### 23.1.1 About the Position Planning Table

ISPSOft provides an easy table for users to set up the single-axis or 2-axis motion control. It is not required for users to know the complicated motion control rules to complete the setup. Create and download a position planning table and then along with the instructions, the PLC can output axis motions in the set number and order accordingly.

### 23.1.2 Setting the Position Planning Table

Click **Wizard (!) > Position Planning Table** on the tool bar to open the **Position Planning Table** window which is categorized into 3 sections as shown in the image shown below.





23



- ① Icon toolbar: Provide buttons for various functions with function description explained below.
- ② Parameter setting area: Settings regarding axis control. Upload password protection is used for password setting. Provides axis control mode and motion path setting in the position table.
- ③ Setting for Simulator: Use **relative mode (symbol/address)** or **absolute mode (symbol/address)** to simulate for the objective position.

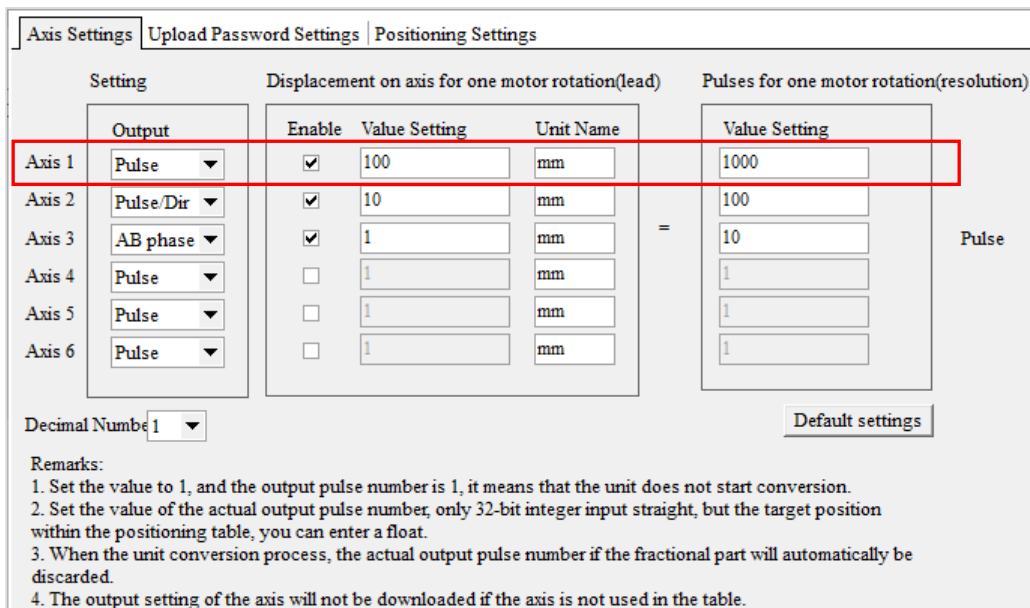


Icon toolbar description:

Icon	Name	Function Description
	Import	Import *.pnt to the current position planning table.
	Export	Export positioning table to *.pnt.
	Upload	Read positioning table settings via hosts.
	Download	Write positioning table settings in hosts.

In Axis Settings page, users can setup each axis control output. The number of axis shown varies according to the supported PLC types. When executing 2-axis interpolation movement, do NOT choose Pulse for **Output** instead choose Pulse/Dir or AB phase. Then click **Enable** next to the **Output** and input self-defined **Value Setting** and **Unit Name** as well as the **Value Setting of Pulses for one motor rotation**. The follow image shows that the setups of Axis 1 for 100 mm can output 1000 pulses in one motor rotation. Click **Default Setting** to restore the initial setup for the unit setting page.

23



Setting	Displacement on axis for one motor rotation(lead)			Pulses for one motor rotation(resolution)
Output	Enable	Value Setting	Unit Name	Value Setting
Axis 1	<input checked="" type="checkbox"/>	100	mm	1000
Axis 2	<input checked="" type="checkbox"/>	10	mm	100
Axis 3	<input checked="" type="checkbox"/>	1	mm	10
Axis 4	<input type="checkbox"/>	1	mm	1
Axis 5	<input type="checkbox"/>	1	mm	1
Axis 6	<input type="checkbox"/>	1	mm	1


Decimal Number 1

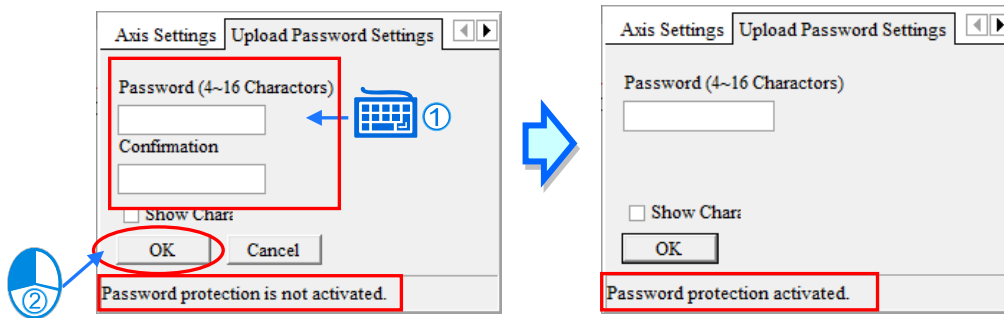
Default settings

Remarks:

1. Set the value to 1, and the output pulse number is 1, it means that the unit does not start conversion.
2. Set the value of the actual output pulse number, only 32-bit integer input straight, but the target position within the positioning table, you can enter a float.
3. When the unit conversion process, the actual output pulse number if the fractional part will automatically be discarded.
4. The output setting of the axis will not be downloaded if the axis is not used in the table.

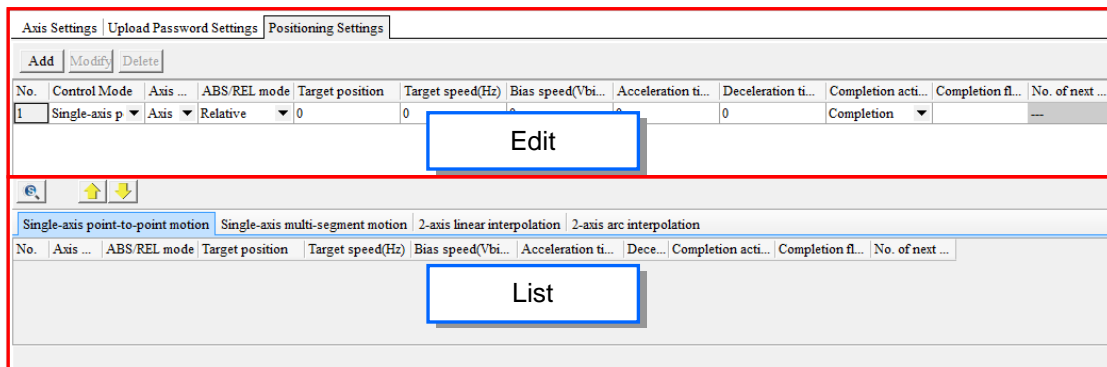
In Upload Password Settings page, the passwords are set to protect during uploading for position planning table downloaded to the host. After typing in the **Password** and again for **Confirmation**, click **OK** and when successful, a hint indicating 'password protection activated' is shown below. To remove the password setting, input the **Password** in the box to click **OK** and when successful a hint indicating 'password protection is not activated' is shown below.

 Please note that Upload Password Settings only provides password for uploading the position planning table. When other planning table is downloaded to hosts with Upload Password Settings, the position planning table will be covered.

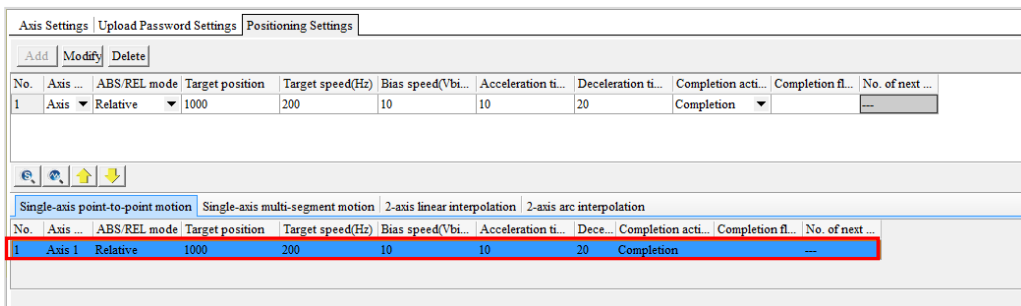
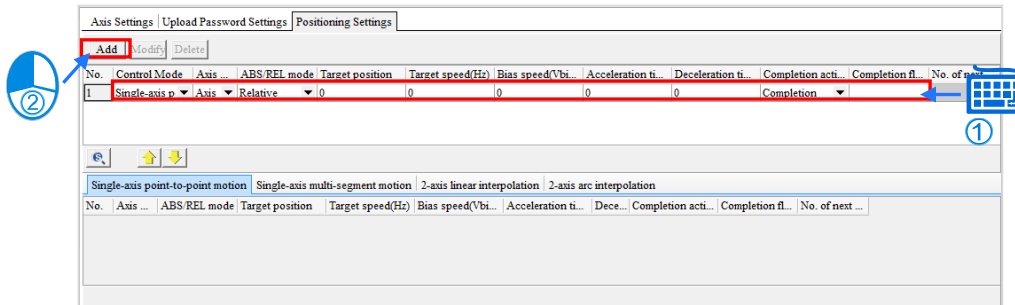


In Positioning Settings page, the edit section provides functions to add, modify or delete. When data are added, it will appear in the list section.

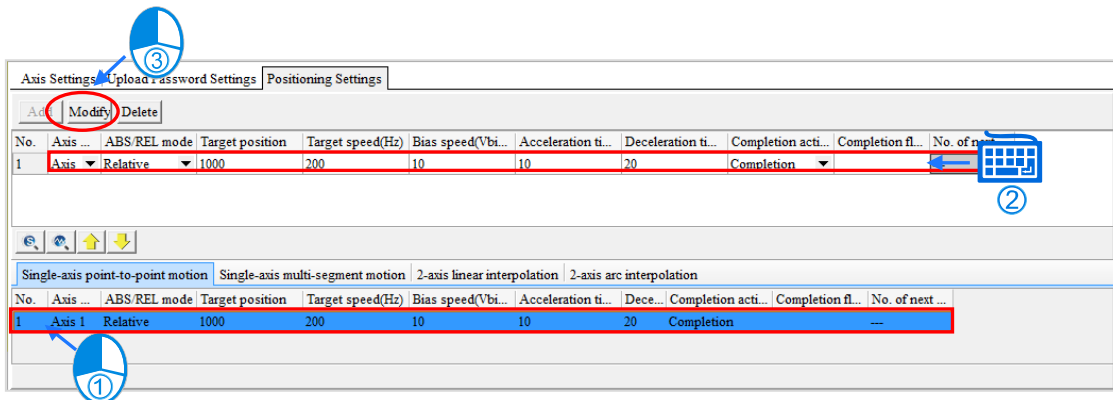
23



To add data, first input settings in each column of the edit section and click **Add** so the data is added to the corresponding axis control mode page (see below) based on **Control Mode**.



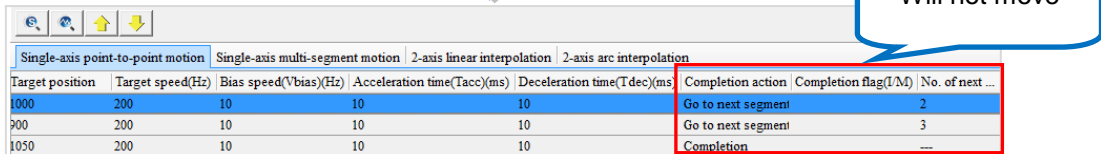
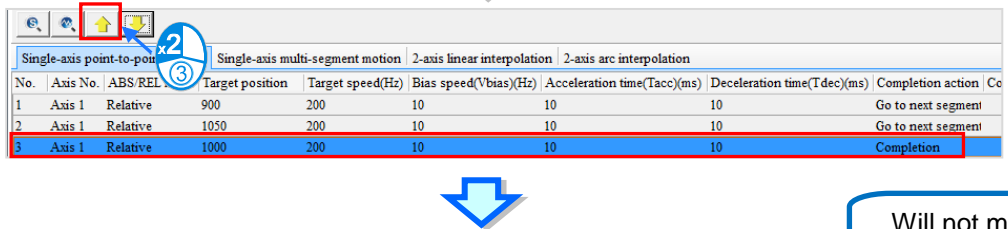
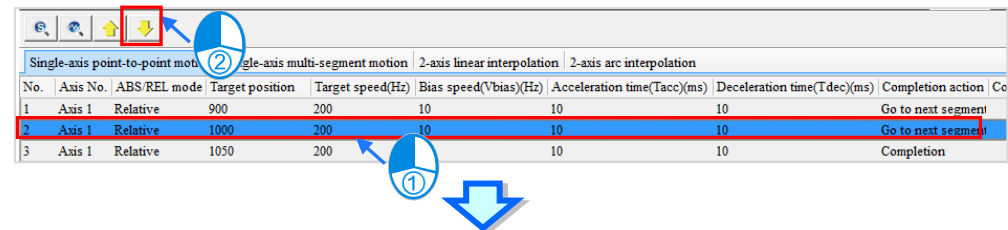
To modify, select the data in the list section then edit its content in the edit section and click **Modify** to update this data in the list section (see below).



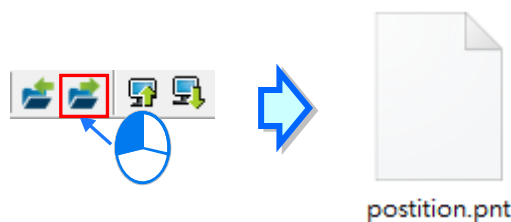
To delete, select the data in the list section and click **Delete** in the edit section. This data is deleted from the list section.

Every number in the list section can move up or down by click / . But please note that **Completion action, No. of next section and Completion flag (I/M)** will not move (see below).


23



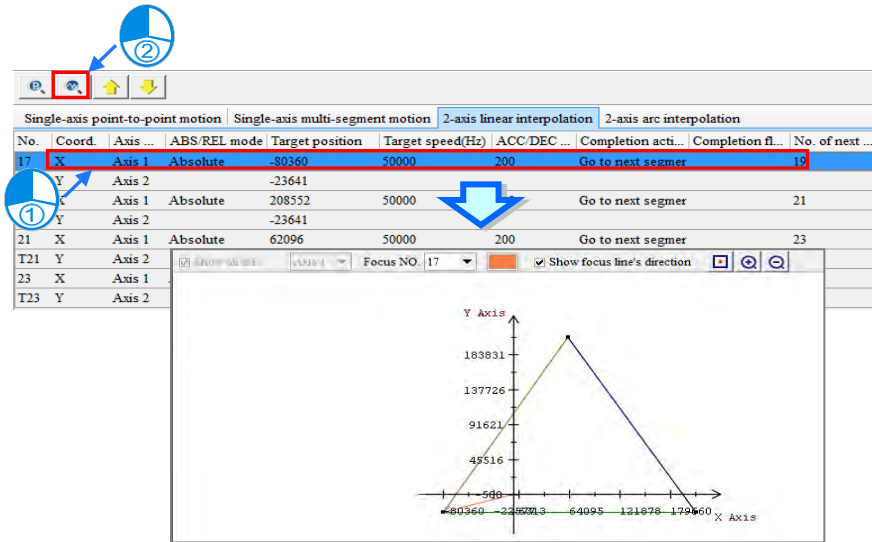
When completing the positioning table data, click **Export** data to \*.pnt (see below). Then, click **Import** and choose \*.pnt file to import the positioning data.











### 23.1.3 Positioning Table Simulation


Select a row in the list section for simulation then click **Assigned Number Diagram**  to view a pop-up window that contains a complete graph-based path planning (see below) that begins with No.17.

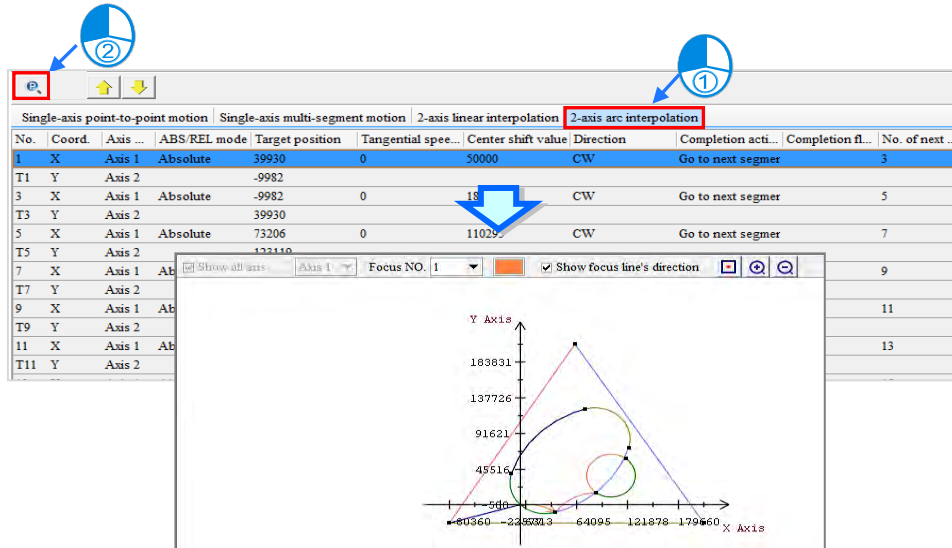
23




Simulation toolbar description:

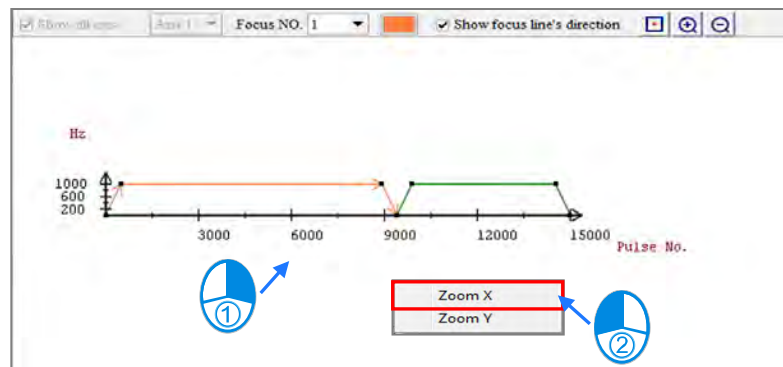
Figure	Function
<input type="checkbox"/> Show All Axis    Axis 1 ▼	Select to show all axes or choose an axis from the
Focus NO 1 ▼ 	Select a No. in the graph to be in bold type or change and choose a color in the right side.
<input checked="" type="checkbox"/> Show focus line's direction	Select to view the arrow indicating motion direction.
	Adjust the graphs to appropriate fit.
	Enlarge the graph in X direction.
	Narrow the graph in X direction.
	Enlarge the graph in Y direction.
	Narrow the graph in Y direction.
	Enlarge the graph.
	Narrow the graph.

By clicking 2-axis arc interpolation in the list section, then click 2-axis simulation mode  (see below), both 2-axis linear interpolation and arc interpolation are shown on the same simulation graph with a mixture image of both continuous paths.




23

By clicking Single-axis point-to-point motion page, then click single axis simulation mode , the single-axis point-to-point and multi-segment simulation graphs only shows the same mode path. Also, right-click on an empty spot in the single simulation graph window and a pop-up list appears for switching the scroll wheel direction (see below).



For Target Simulation functions, users need to declare symbols in DWORD or DINT type in global symbols. Then, select either **Absolute (Symbol/Address)** or **Relative (Symbol/Address)** for ABS/REL mode in edit section, and type in symbol or device for Objective position (see below).

 For **Objective position** in the position planning table, the smallest unit is DWORD. So, please be aware that D100 is estimated to occupy the length of D100 and D101.

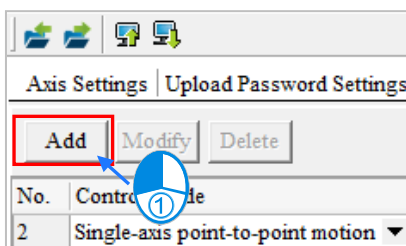
Global Symbols					
Class	Identifiers	Address	Type...	Initial Value (Act...	Identifier Comment...
VAR	Data_1	D100	DWORD	N/A	



No.	Control Mode	Axis No.	ABS REL mode	Target po...	Target	Max speed(Vbias)/Hz	Acceleration ti...	Deceleration t...	Completion action	Completion flag(I,M)	No. of nest segment
2	Single-axis point-to-point motion	Axis 1	Relative/Symbol/Add	Data_1	1000		10	10	Completion		



When completed, click **Add** in the target simulation section to automatically add this data. Meanwhile, type a value in Objective Position (Simulator) item list, users can check the simulation path in **Single simulation mode** or **2-Axis simulation mode**.



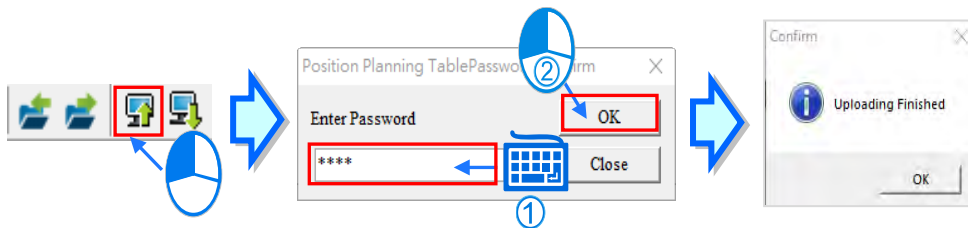
NO.	Symbol Device	Objective Positioe(Simulator)
1	D100	1000
2	Data_1	1000



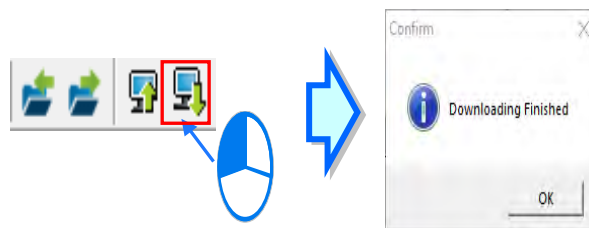
23

### 23.1.4 Upload and Download Position Planning Table

When uploading or downloading position planning table, users need to connect ISPSOft to a host and select **Upload** for the position planning table saved in the host. When the table contains upload password protection, a window referring to password appears. Input the correct password to upload the position planning table (see below).



Choose the Download icon for downloading the position planning table to the host.



### 23.1.5 Control Mode – Single axis point-to-point motion

Description regarding single axis point-to-point mode setting.

No.	Control Mode	Axis No.	ABS/REL mode	Target position	Target speed(Hz)	Bias speed(Vbias)(Hz)	Acceleration time(Tacc)(ms)	Deceleration time(Tdec)(ms)	Completion action	Completion flag(I/M)	No. of next segment
1	Single-axis point-to-point motion	Axis 1	Relative	0	0	0	0	0	Completion		---

- **No.:** The number is generated by the system and the maximum No. is 200.
- **Control Mode:** A motion type. Choose **Single-axis point-to-point motion**.
- **Axis No.:** An axis controlled by the data.
- **ABS/REL mode:**
  - Relative Mode** increases or decreases based on the current coordinates.
  - Absolute Mode** increases or decreases based on the absolute coordinates as target position.
  - Relative mode (Symbol/Address)** increases or decreases based on the current coordinate address, symbol or address are allowed.
  - Absolute mode (Symbol/Address)** increases or decreases based on the absolute coordinates as target position, symbol or address are allowed.
- **Target position:** Target coordinate. For **ABS/REL mode**, the options include symbol/address.
- **Target speed:** Maximum speed of the target.
- **Bias speed:** The start and ending speed.
- **Acceleration time:** The time to reach target speed.
- **Deceleration time:** The time to reach stop.
- **Completion action:**
  - Go to next segment** to complete the data and continue to the next.
  - Completion** refers to an end once data is complete.
- **Completion flag (I/M):** When the data execution is complete, set the register to ON. The register range is set as M0~M8191 or I510~I519 only when **Completion action** is set to **Completion**.
- **No. of next segment:** Continue with the next data number, not limited to the same assigned axis number. But, take note that the same axis number cannot output more than 2 numbers from the table or output position command at the same time. When **Completion action** is set to **Completion**, the number is automatically continued to the next segment ---

23

Notice for table input:

1. The target speed cannot be set to 0. The objective position also cannot be set to 0 in the relative mode.
2. I number can be used more than once. Users must determine through which row number the I number can be completely outputted.
3. The instruction which the start table outputs is API2718 TPO. API2719 TPWS can be used to modify the output parameters while PLC is running.

## 23.1.6 Control Mode – Single axis multi-segment

Description regarding single axis multi-segment mode setting.

No.	Control Mode	Axis No.	ABS/REL mode	Target position	Target speed(Hz)	ACC/DEC time(ms)	Abort signal(I/M)	Completion action	Completion flag(I/M)	No. of next segment
1	Single-axis multi-segment motion ▼	Axis 1 ▼	Relative ▼	0	0	0		Go to next segment ▼		2

- **No.:** The number is generated by the system and the maximum No. is 200.
- **Control Mode:** A motion type. Choose **Single-axis multi-segment motion**.
- **Axis No.:** An axis controlled by the data.
- **ABS/REL mode:**
  - Relative Mode** increases or decreases based on the current coordinates.
  - Absolute Mode** increases or decreases based on the absolute coordinates as target position.
  - Relative mode (Symbol/Address)** increases or decreases based on the current coordinate address, symbol or address are allowed.
  - Absolute mode (Symbol/Address)** increases or decreases based on the absolute coordinates as target position, symbol or address are allowed.
- **Target position:** Target coordinate. For **ABS/REL mode**, the options include symbol/address.
- **Target speed:** Maximum speed of the target.
- **ACC/DEC time:** The time to reach target speed and stop.
- **Abort signal (I/M):** When execution of the table is not yet complete and the register of abort signal is set ON, the next segment of position planning is executed. Users can input I for abort number or M for register bane. For information on supported abort numbers, please refer to related product programming manuals.
- **Completion action:**
  - Go to next segment** to complete the data and continue to the next.
  - Completion** refers to an end once data is complete.
- **Completion flag (I/M):** When the data execution is complete, set the register to ON. The register range is set as M0~M8191 or I510~I519 only when **Completion action** is set to **Completion**.
- **No. of next segment:** Continue with the next data number, not limited to the same assigned axis number. But, take note that the same axis number cannot output more than 2 numbers from the table or output position command at the same time. When **Completion action** is set to **Completion**, the number is automatically continued to the next segment ---

23



Notice for table input:

1. The objective position cannot be set to 0 in the relative mode. The objective positions of two adjacent row numbers cannot be set to the same value in the absolute mode.
2. If the value in the ACC/DEC time is not 0, the acceleration/deceleration is fixed to 1ms. If the value in the ACC/DEC time is set to 0, it means there is no acceleration or deceleration.
3. I number can be used more than once. Users must determine through which row number the I number can be completely outputted.
4. The instruction which the start table outputs is API2718 TPO. API2719 TPWS can be used to modify the output parameters while PLC is running.

### 23.1.7 Control Mode - 2-axis linear interpolation motion

Description regarding 2-axis linear interpolation mode performs the motion on XY plane. Two data are used for this mode.

No.	Control Mode	Coord.	Axis No.	ABS/REL mode	Target position	Target speed(Hz)	ACC/DEC time(ms)	Completion action	Completion flag(I/M)	No. of next segment
1	2-axis linear interpolation	X	Axis 1	Relative	0	0	0	Completion		---
T1		Y	Axis 2		0					

- **No.:** The number is generated by the system, the 2-axis mode occupies two data. The second data No. will add a T in front of the number. Under 2-axis mode, the maximum No. is 100 and T100, a total of 100 set.
- **Control Mode:** A motion type. Choose **2-axis linear interpolation motion**.
- **Coord:** The X or Y coordinates in the simulation graph.
- **Axis No.:** An axis controlled by the data.
- **ABS/REL mode:**
  - Relative Mode** increases or decreases based on the current coordinates.
  - Absolute Mode** increases or decreases based on the absolute coordinates as target position.
  - Relative mode (Symbol/Address)** increases or decreases based on the current coordinate address, symbol or address are allowed.
  - Absolute mode (Symbol/Address)** increases or decreases based on the absolute coordinates as target position, symbol or address are allowed.
- **Target position:** Target coordinate. For **ABS/REL mode**, the options include symbol/address.
- **Target speed:** Maximum speed of the target.
- **ACC/DEC time:** The time to reach target speed and stop.
- **Completion action:**
  - Go to next segment** to complete the data and continue to the next.
  - Completion** refers to an end once data is complete.

- **Completion flag (I/M):** When the data execution is complete, set the register to ON. The register range is set as M0~M8191 or I510~I519 only when **Completion action** is set to **Completion**.
- **No. of next segment:** Continue with the next data number, not limited to the same assigned axis number. But, take note that the same axis number cannot output more than 2 numbers from the table or output position command at the same time. When **Completion action** is set to **Completion**, the number is automatically continued to the next segment ---

Notice for table input:

1. The PLC will automatically run at the set speed when the target speed or the Acc\Dec time is set to 0. The objective position cannot be set to 0 in the relative mode. The objective positions of two adjacent row numbers cannot be set to the same value in the absolute mode.
2. I number can be used more than once. Users must determine through which row number the I number can be completely outputted.
3. The instruction which the start table outputs is API2718 TPO. API2720 TPWL can be used to modify the output parameters while PLC is running.

## 23

### 23.1.8 Control Mode - 2-axis Arc Interpolation Motion

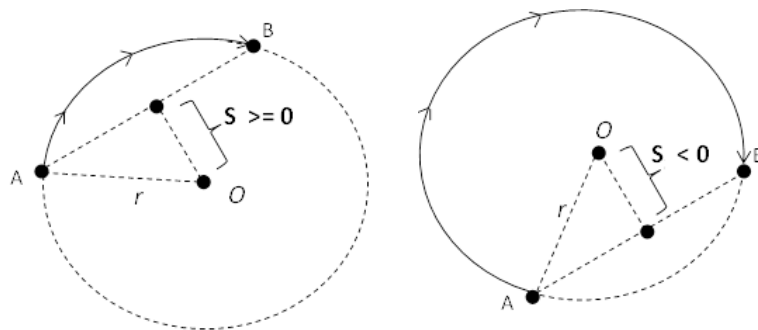
Description regarding 2-axis arc interpolation mode performs the motion on XY plane. Two data are used for this mode.

No.	Control Mode	Coord.	Axis No.	ABS/REL mode	Target position	Tangential speed(Hz)	Center shift value	Direction	Completion action	Completion flag(I/M)	No. of next segment
1	2-axis arc interpolation	X	Axis 1	Relative	0	0	0	CW	Completion		---
T1		Y	Axis 2		0						

- **No.:** The number is generated by the system, the 2-axis mode occupies two data. The second data No. will add a T in front of the number. Under 2-axis mode, the maximum No. is 100 and T100, a total of 100 set.
- **Control Mode:** A motion type. Choose **2-axis arc interpolation motion**.
- **Coord:** The X or Y coordinates in the simulation graph.
- **Axis No.:** An axis controlled by the data.
- **ABS/REL mode:**
  - Relative Mode** increases or decreases based on the current coordinates.
  - Absolute Mode** increases or decreases based on the absolute coordinates as target position.
  - Relative mode (Symbol/Address)** increases or decreases based on the current coordinate address, symbol or address are allowed.

**Absolute mode (Symbol/Address)** increases or decreases based on the absolute coordinates as target position, symbol or address are allowed.

- **Target position:** Target coordinate. For **ABS/REL mode**, the options include symbol/address.
- **Tangential speed (Hz):** Movement in a circular path in three dimensions and the magnitude of that vector is the tangential speed.
- **Center shift value:** The distance from the center of the line between the arc target point and start point to the center of a circle, such as S in the following figures. From point A to point B, a clockwise rotation will generate an arc less than  $180^\circ$  if  $S > 0$  and an arc more than  $180^\circ$  if  $S < 0$ .



- **Direction:** Users can select **CW** clockwise motion or **CCW** counter clockwise motion.
- **Completion action:**  
**Go to next segment** to complete the data and continue to the next.  
**Completion** refers to an end once data is complete.
- **Completion flag (I/M):** When the data execution is complete, set the register to ON. The register range is set as M0~M8191 or I510~I519 only when **Completion action** is set to **Completion**.
- **No. of next segment:** Continue with the next data number, not limited to the same assigned axis number. But, take note that the same axis number cannot output more than 2 numbers from the table or output position command at the same time. When **Completion action** is set to **Completion**, the number is automatically continued to the next segment ---

Notice for table input:

1. The PLC automatically calculates the output speed in this function. The objective position cannot be set to 0 in the relative mode. The objective positions of two adjacent row numbers cannot be set to the same value in the absolute mode.
2. I number can be used more than once. Users must determine through which row number the I number can be completely outputted.
3. The instruction which the start table outputs is API2718 TPO. API2721 TPWL can be used to modify the output parameters while PLC is running.

## 23.2 Data Tracer

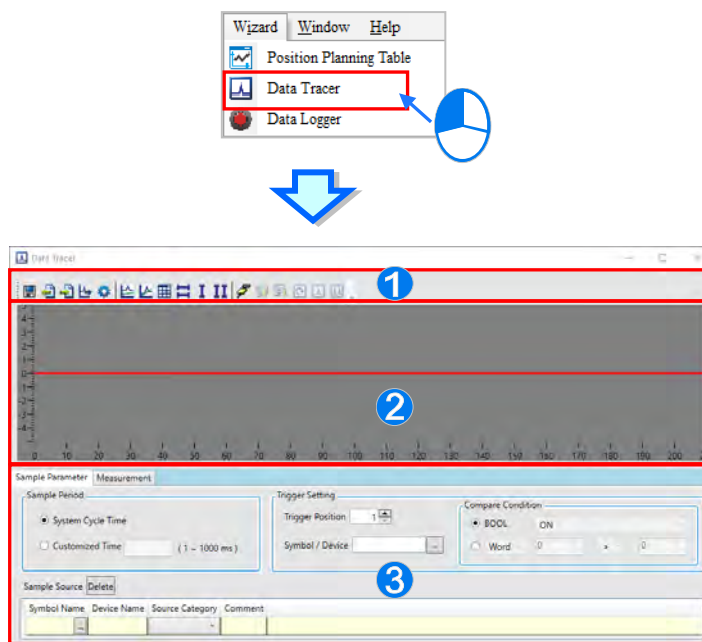
### 23.2.1 About Data Tracer

Data Tracer is used once a triggering condition is reached to collect real-time symbol variables or status and using waveform graphs to analyze the value trends.

### 23.2.2 Opening the Data Tracer Window









Please compile the current project before using the function. Click **Wizard > Data Tracer** on the tool bar to open the **Data Tracer** setup window as the image shown below.

23



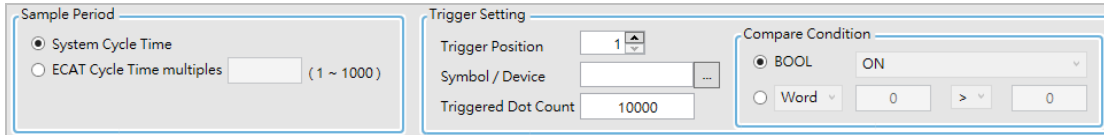
- ❶ Icon toolbar: Provides functions in icons and explanation below.
- ❷ Waveform display area: Displays real-time symbol variables or waveform of the device.
- ❸ Parameter setting area: Sample Parameter page is for setting the sample source and trigger condition. Measurement page is for displaying the values and adjusted waveform colors and proportions through measuring equipment.

Explanation of icon toolbar:

Icon	Name	Function
	Save All	The sample settings and curve data results are saved to the specified path.
	Export Sampling	The sample settings are exported to the specified path.
	Import Sampling	The sample settings are imported from the specified path.
	Export Curve Data	The sample curve data are exported in the format of .bmp and .csv to the specified path.
	Configuration	The colors of curves in the data tracer and environment can be set.
	Separate Curves	Multiple curves are separated automatically.
	Overlap Curves	Multiple curves are overlapped with the center line.
	Show Grid Line	Show or Hide the grid line.
	Horizontal Measuring Lines	There will be two horizontal lines in the curve display area for users to measure the curve.
	Vertical Measuring Lines	There will be one vertical line in the curve display area for users to measure the curve.
	Vertical Measuring Lines	There will be two vertical lines in the curve display area for users to measure the curve.
	Connect/Disconnect	Connect or disconnect the PC and PLC.
	Upload Sampling	The sample settings in the data tracer are read from the PLC.
	Download Sampling	The sample settings in the data tracer are written to the PLC.
	Real Time Monitor	The PC reads the curve of the source from the PLC in real time.
	One-Shot Trigger	After the trigger condition is met, a total 200 points of curve data, including pre- and post- of the triggered point will be read from the PLC.
	Continuous Trigger	Whenever the trigger condition is met, a total 200 points of curve data, including pre- and post- of the triggered point will be read from the PLC.

### 23.2.3 Sample Parameter Settings

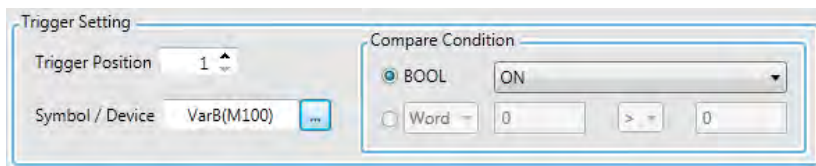
For sample setting page, the following chart shows that the items may differ based on different PLC types. The **Sample Period** is for setting sample time period. While **Trigger Setting** is for trigger position and condition settings. When trigger condition is reached, the host maintains a total of 200 data and reflected in the Waveform area.



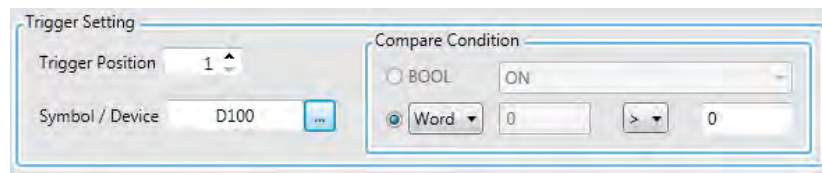
23



Name	Desc	AS	AHxxEMC	AH5x1
System Cycle Time	Host scan time	V	V	V
ECAT Cycle Time	ECAT scan cycle		Max. 1000	
User-defined Time	User-defined time	Max.1000		Max.1000
Trigger Position	Triggering	Max. 200	Max. 10000	Max. 200
Symbol/Device	Triggering device/ symbol	Supports Global Symbol	Supports Symbol, Local Symbol, Axis Symbol, ECAT	Supports Global Symbol
Triggered Points	Max. value for trigger position		Max. 10000	

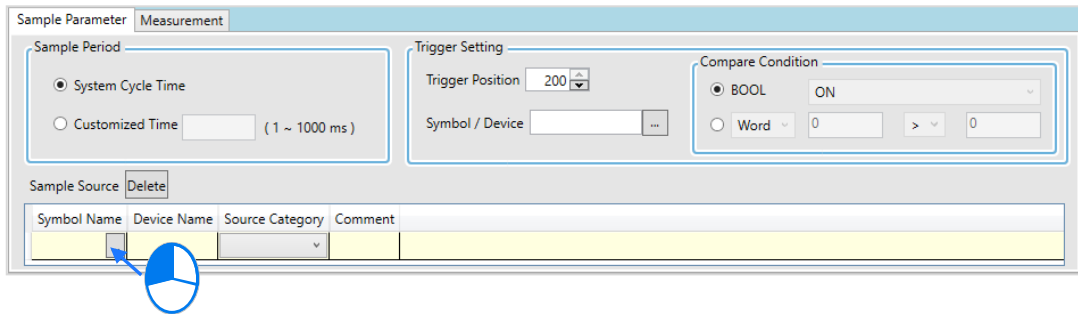
When selecting symbol variables or device in BOOL for Symbol/Device column, the compare condition will auto-select BOOL (see below). Or choose **ON, OFF, rising edge-triggered, falling edge-triggered, rising or falling edge-triggered** from the drop-down list. When the symbol or device fulfills the setting condition, the trigger will show on the data tracer.



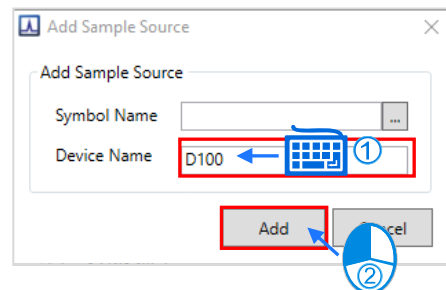
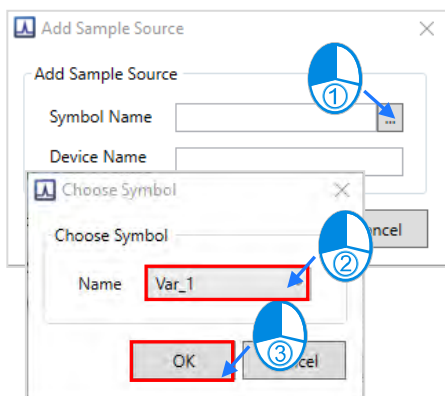
When choosing Word, DWord or Real as symbol variables or devices for Symbol/Device column, the system auto-selects Word (see below) or select DWord or Real from the drop-down list and select conditions by choosing >, <, =, != (not equal), ~ (a range) or !~ (out of a range) together with the numbers in front and back of the setting. When the setting symbol or device fulfills the condition, the trigger is shown on the data tracer. The following image shows value in D100 is 0 > 0 and fulfills trigger condition; when setting is 10 !~ 100, this means D100 is no more than 10 or exceeds 100 to fulfill trigger condition.



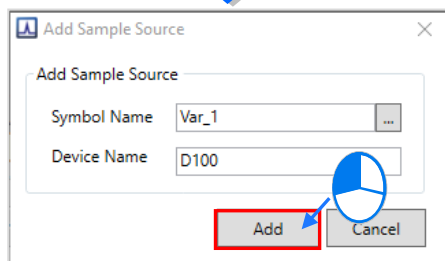
To add sample symbols, click  button in Symbol Name and a pop-up Add Sample Source window appears. Then click the  button to select the added symbol variables or type the device name in the window and click **Add** to the sample source list (see below).



23

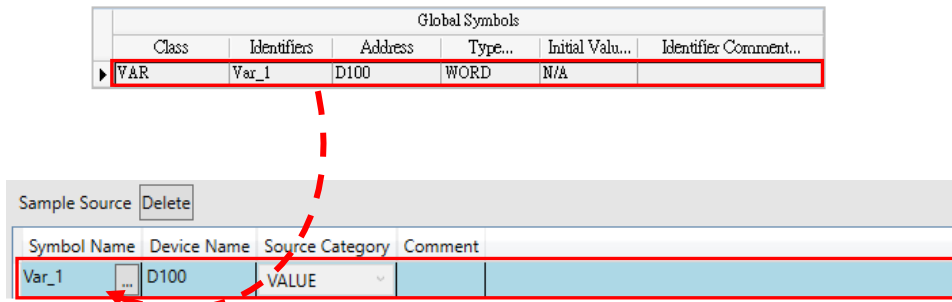


OR



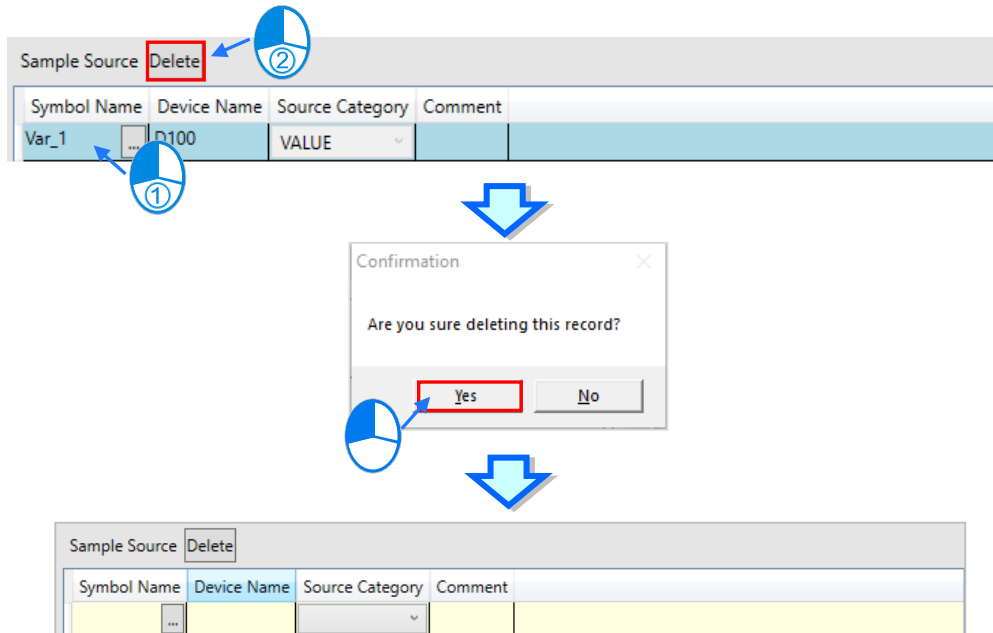
Symbol Name	Device Name	Source Category	Comment
Var_1	D100	VALUE	



Symbols that are declared in global symbols and compiled in projects, are selected here. Manual device settings or identifier comments are also included.



To delete the sample source which has been added, select the deleted item and click **Delete**.


23

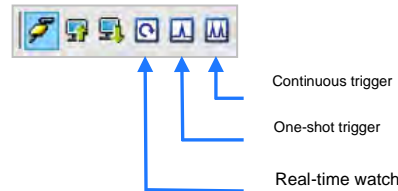


When sample parameter settings are complete, click **Export** or  from the icon toolbar and save the sample parameter settings in .dtr format. Then, users can choose **Import** or  to restore sample parameter settings in new projects.

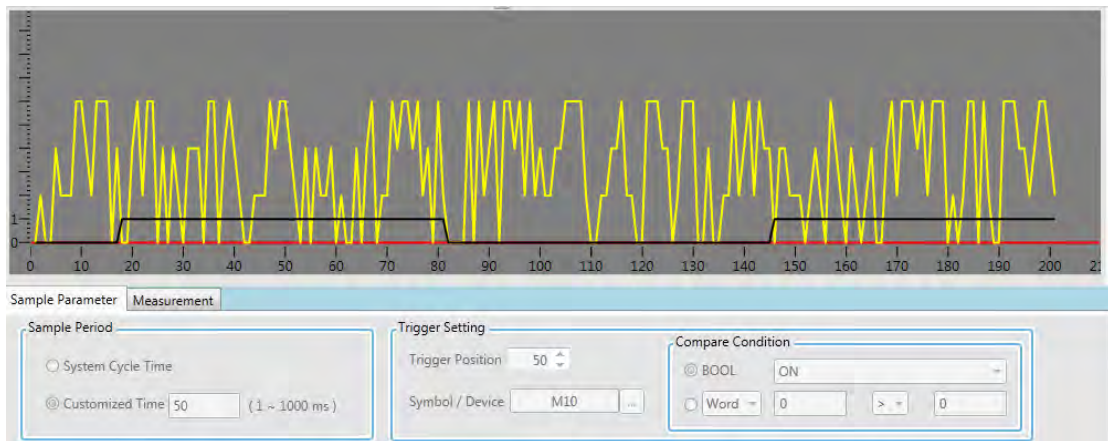


### 23.2.4 Sampling Modes

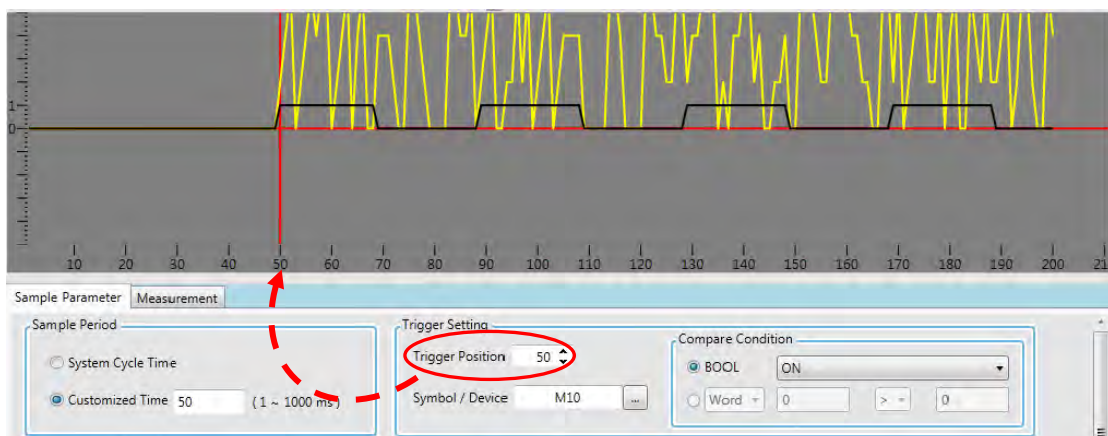
When sample parameter setting for data tracer is complete, download parameter setting during connection mode  and use the icon toolbar to switch and view the waveforms from the following three modes.



- **Real-time watch:** According to the set sample period, the curves for variable symbol or device value or state in the sample source are constantly drawn to the data tracer. The curve chart will keep the latest 200 pieces of data while moving leftward.



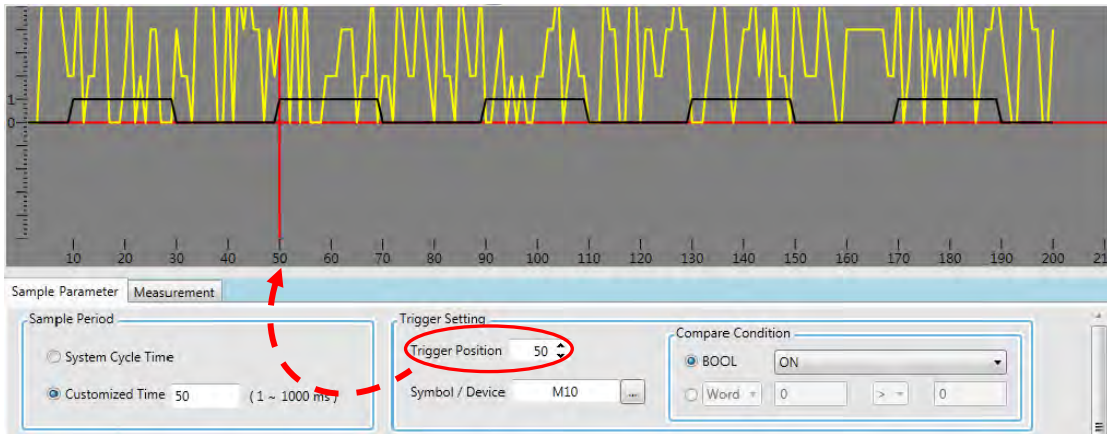
- **One-shot trigger:** According to the trigger position setting, the sample data to the right of the trigger position will be displayed in waveforms when the trigger condition is met and the trigger is done once only. The following diagram shows that the trigger position is set to 50. When trigger condition M10 is ON, the sample source drawing ranges from 50~200 based on sample period. The following red vertical line represents the trigger position in the waveform.



- Continuous trigger: The data tracer will keep the state of constant reading according to the trigger position setting. The sample data of the pre- and post-triggered points will be displayed in the data tracer whenever the trigger condition is met.

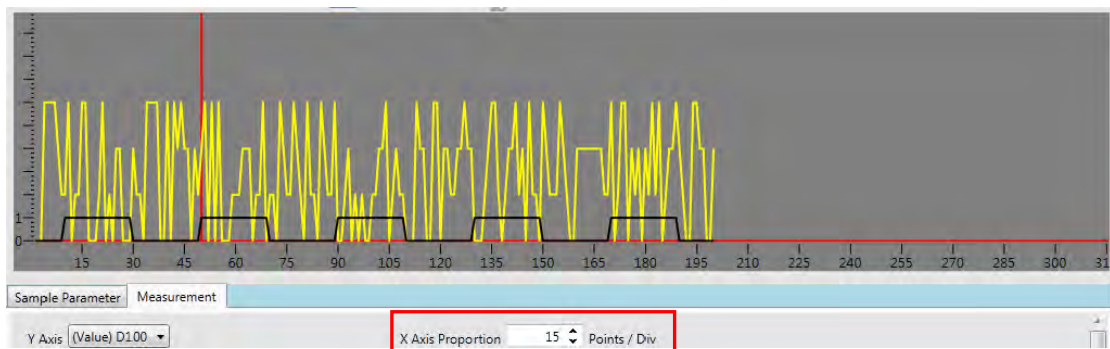
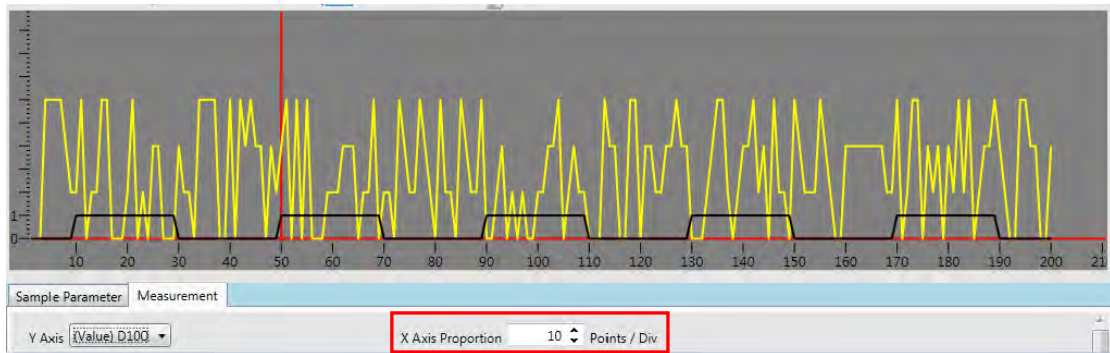
To view the curve chart, the state of the continuous trigger button must be restored from OFF to ON. As the following example shows the trigger position is set to 50 and thus the curves for the sample source according to the sample period setting will be drawn 200 pieces including 49 pieces of the pre-triggered points and 150 pieces of the post-triggered points whenever the trigger condition M10=ON is met. Where the red vertical line is represents the trigger position in the wave shape chart.

23



### 23.2.5 Measurement

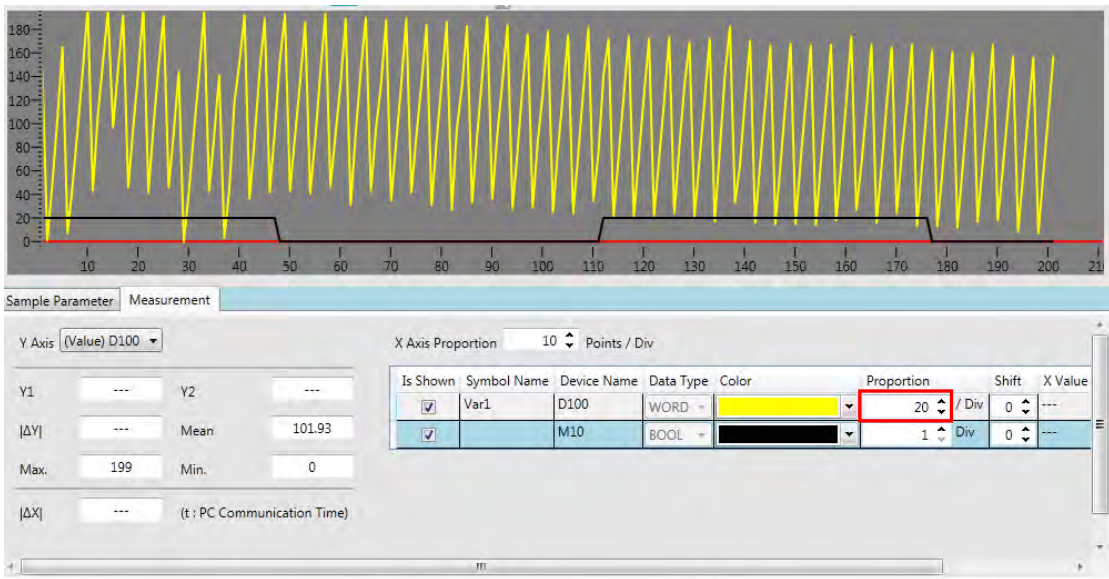
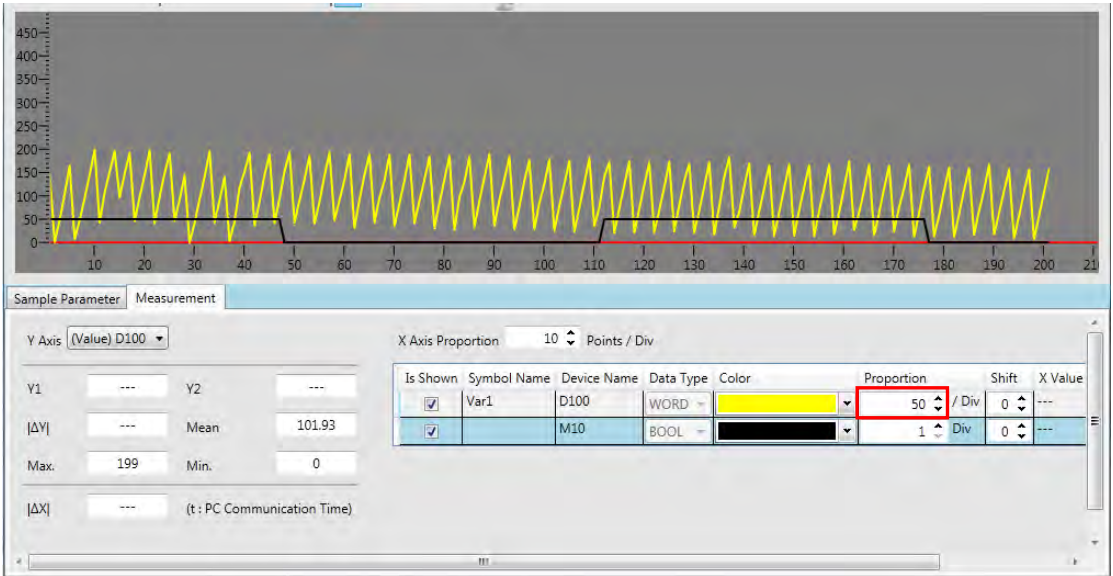
In **Measurement**, the page provides adjustment regarding the waveform shape of the data tracer for enhanced display effect. The following image shows the points of each scale in adjusting the horizontal coordinate for **X Axis Proportion** modification.



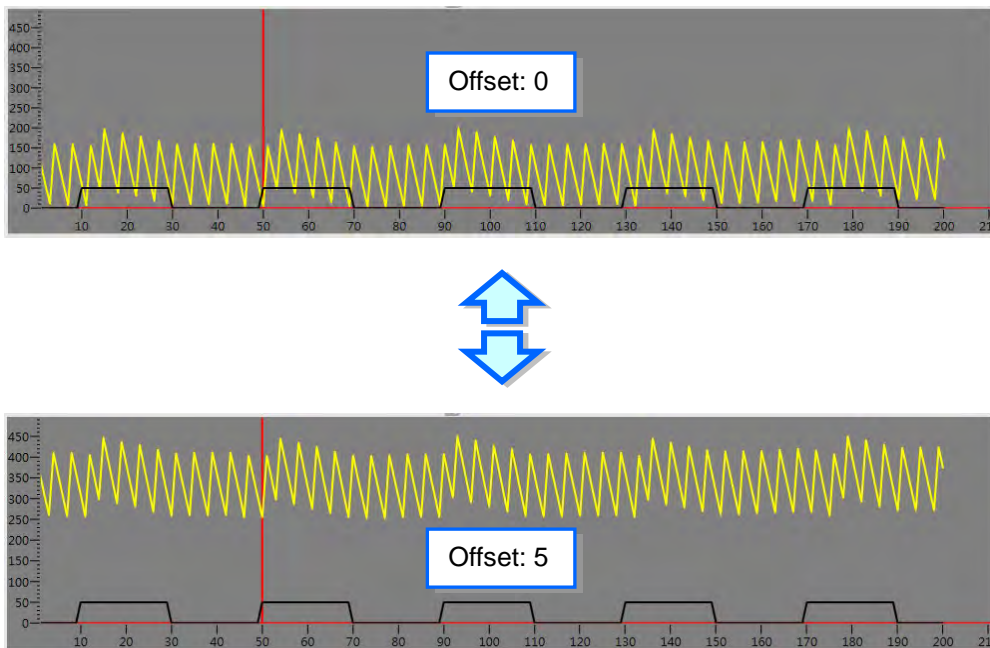
23

In Measurement, the page offers the values of mean, maximum and minimum regarding waveform information of selected device in Y axis. Each waveform has different color, proportion and shift settings. The **Proportion** refers to the proportion of vertical coordinate and the following diagram shows proportion of each scale set in 50 and 20 for comparison.

23



**Shift** can be set to modify the position of the curve in the display as shown in the following figures.



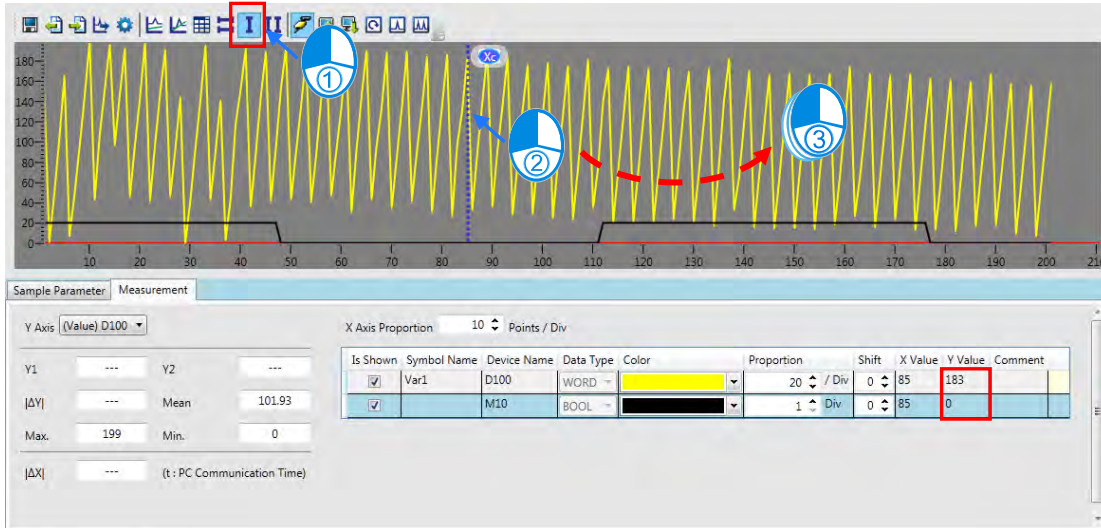
23

Drag the two horizontal measuring lines Y1 and Y2 and adjust to the measured position together with horizontal measuring lines. The parameter setting area displays every value regarding horizontal measuring lines. The Y1 value refers to Y1 column, Y2 value refers to Y2 column and differences between Y1 and Y2 refers to  **$|\Delta Y|$  column**.

The screenshot shows the software interface with a waveform and a parameter table. The waveform has two horizontal measuring lines, Y1 and Y2, and a vertical measuring line Xc. The parameter table below the waveform is as follows:

Y Axis (Value) D100	X Axis Proportion	10	Points / Div								
Y1: 158.82	Y2: 38.82	Is Shown	Var1	D100	WORD	Color	Proportion	Shift	X Value	Y Value	Comment
$ \Delta Y $ : 120	Mean: 101.93	<input checked="" type="checkbox"/>					20 / Div	0	105	175	
Max.: 199	Min.: 0	<input checked="" type="checkbox"/>		M10	BOOL		1 / Div	0	105	0	
$ \Delta X $ : ---	(t: PC Communication Time)										

Drag the vertical measuring line Xc to match with other vertical measuring lines. The Y value column below shows an intersected corresponding value regarding each sample resource and measuring line Xc.



23

The following diagram measures the time value of data tracer by matching two vertical measuring lines. Click **II** from the toolbar to start the two vertical measuring lines and use the mouse to drag the two vertical lines, X1 and X2 in adjusting to the measured position. Time value is shown in the following  $\Delta X1$  column. (t value: the communication time between a PC and PLC)



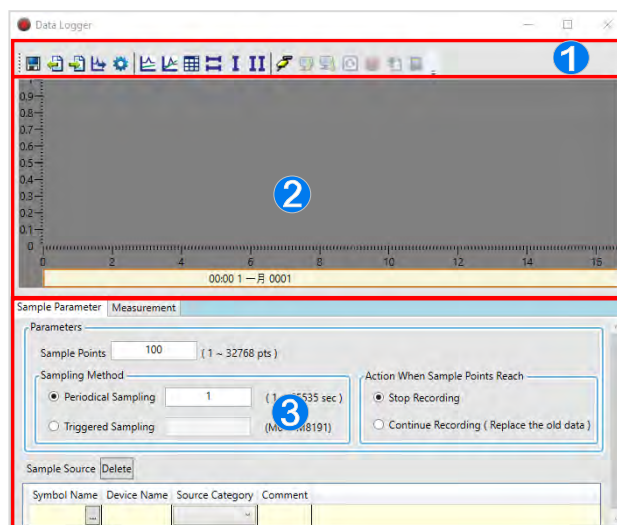
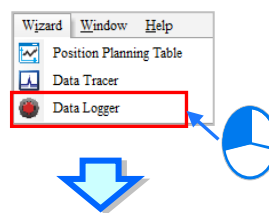
## 23.3 Data Logger

### 23.3.1 About Data Logger

Data logger is used long-term recording of symbol variables or device values and status that uses waveforms for diagram display. The data are saved in the hosts and memory card for convenient value trend analysis.

### 23.3.2 Opening the Data Logger Window

Please compile the current project before using the function. And then click **Wizard> Data Logger** to open the **Data Logger** window as below.





- ❶ Icon toolbar: Provides function buttons for operating the data logger with function descriptions below.
- ❷ Waveform display area: Displays real-time symbol variables or waveform of the device.
- ❸ Parameter Setting area: Sample Parameter page is for setting sample source and parameters. Measurement page is for displaying the values and adjusted waveform colors and proportions through measuring equipment.



## Explanation of icon buttons on the icon toolbar

Icon	Name	Function
	Save All	The sample settings and curve data result are saved to the specified path.
	Export Sampling Config	The sample settings are exported to the specified path.
	Import Sampling Config	The sample settings are imported from the specified path.
	Export Curve Data	The sample curve data are exported in the format of .bmp and .csv to the specified path.
	Configuration	The colors of curves in the data tracer and environment are set.
	Separate Curves	Multiple curves are separated automatically.
	Overlap Curves	Multiple curves are overlapped with the center line.
	Show Grid Line	Show or Hide grids
	Horizontal Measuring Lines	There will be two horizontal lines in the curve display area for users to measure the curve.
	Vertical Measuring Lines	There will be one vertical line in the curve display area for users to measure the curve.
	Vertical Measuring Lines	There will be two vertical lines in the curve display area for users to measure the curve.
	Connect/Disconnect	The PC and PLC are connected or disconnected.
	Upload Sampling Config from PLC	The sample settings in the data tracer are read from the PLC.
	Download Sampling Config to PLC	The sample settings in the data tracer are written to the PLC.
	Real Time Monitor	The PC reads the curve of the source from the PLC in real time.
	Record Data	Click this button to start recording or cancel the clicking of the button to stop recording.

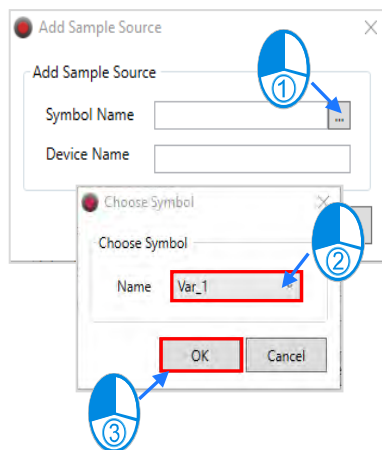
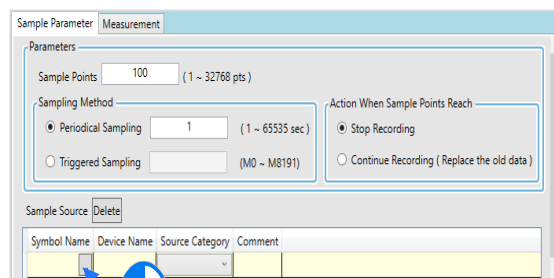


Icon	Name	Function
	Upload Sampling Config from PLC	The recorded data in the data logger are read from the PLC.
	Write to Memory Card	Write the data recorded in the PLC to the memory card.

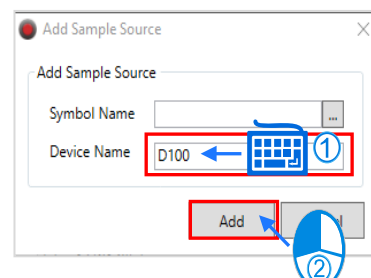
### 23.3.3 Sample Parameter Settings

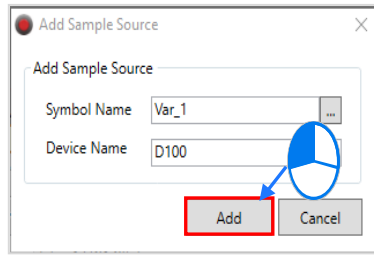
In sample parameter page, the Sample Points are used to setup the number of samples; **Sampling Method** can be categorized into **Periodical Sampling** and **Triggered Sampling**. When using **Periodical Sampling**, users need to setup cycle time once every week; when using **Triggered Sampling**, users need to setup triggered device. When triggered device is ON, the sampling takes place, however, the triggered device is auto-switched to OFF once sampling is complete. When the number of sample points has reached, users can select either **Stop Recording** or **Continue Recording (Replace old data)**. To add sample symbols, click  button in Symbol Name and a pop-up Add Sample Source window appears. Then click the  button to select the added symbol variables or type the device name in the window and click Add to the sample source list (see below).

23



OR





Symbol Name	Device Name	Source Category	Comment
Var_1	D100	VALUE	

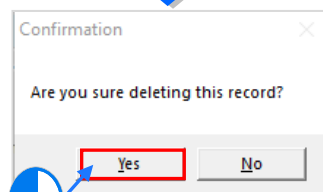
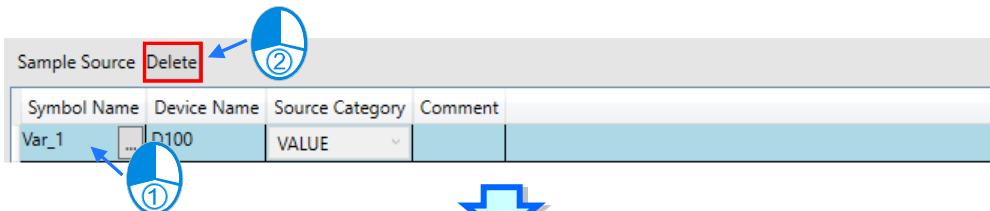
Symbols that are declared in global symbols and compiled in projects, are selected here. Manual device settings or identifier comments are also included.

23



Global Symbols						
Class	Identifiers	Address	Type...	Initial Valu...	Identifier Comment...	
VAR	Var_1	D100	WORD	N/A		

Symbol Name	Device Name	Source Category	Comment
Var_1	D100	VALUE	


To delete the sample source which has been added, select the deleted item and click **Delete**.

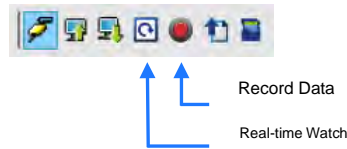


Symbol Name	Device Name	Source Category	Comment

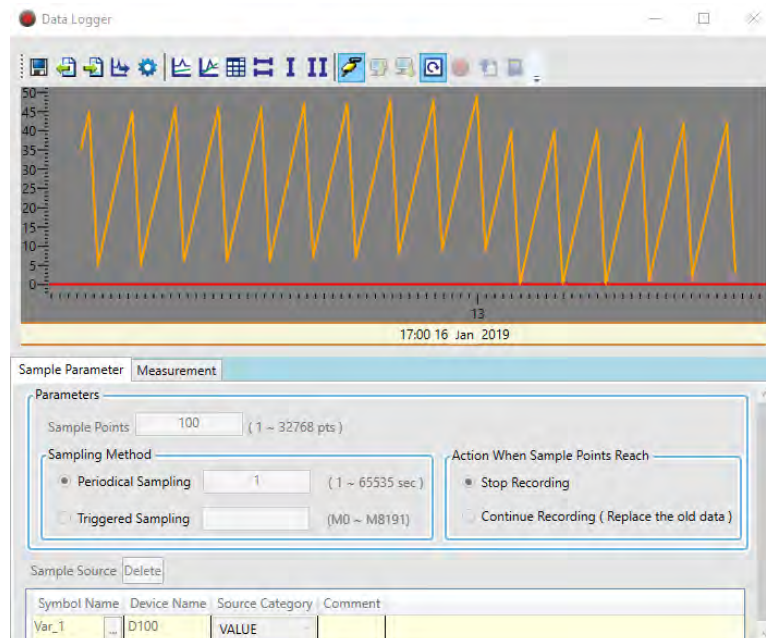
When sample parameter settings are complete, click **Export** or  from the icon toolbar and save the sample parameter settings in .dlg format. Then, users can choose **Import** or  to restore sample parameter settings in new projects.

### 23.3.4 Watch and Record

When sample parameter setting for data logger is complete, download parameter setting during connection mode  and use the icon toolbar to switch and watch or record real-time data.



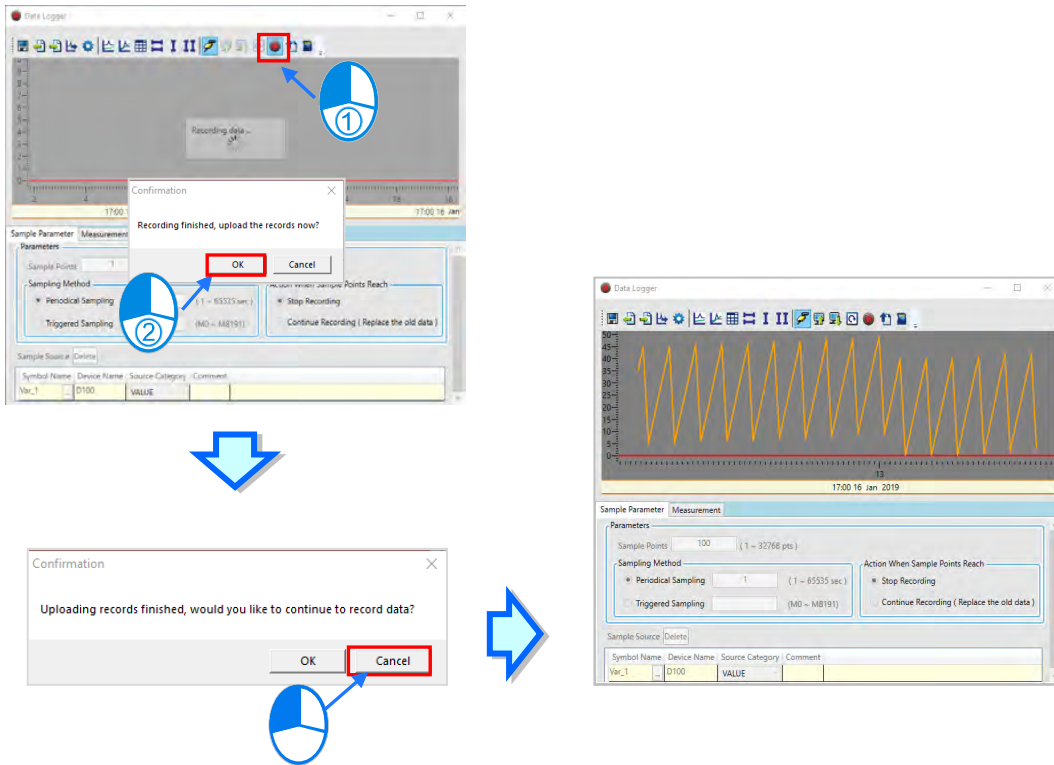
- **Real-time Watch:** According to the set sample period, the curves for the variable symbol or device value or state of the sample source are constantly drawn to the data logger. The screen will be adjusted to the appropriate range based on the curve shape and only the data of the latest sample points are kept. The PLC does not record the sample source in this mode.



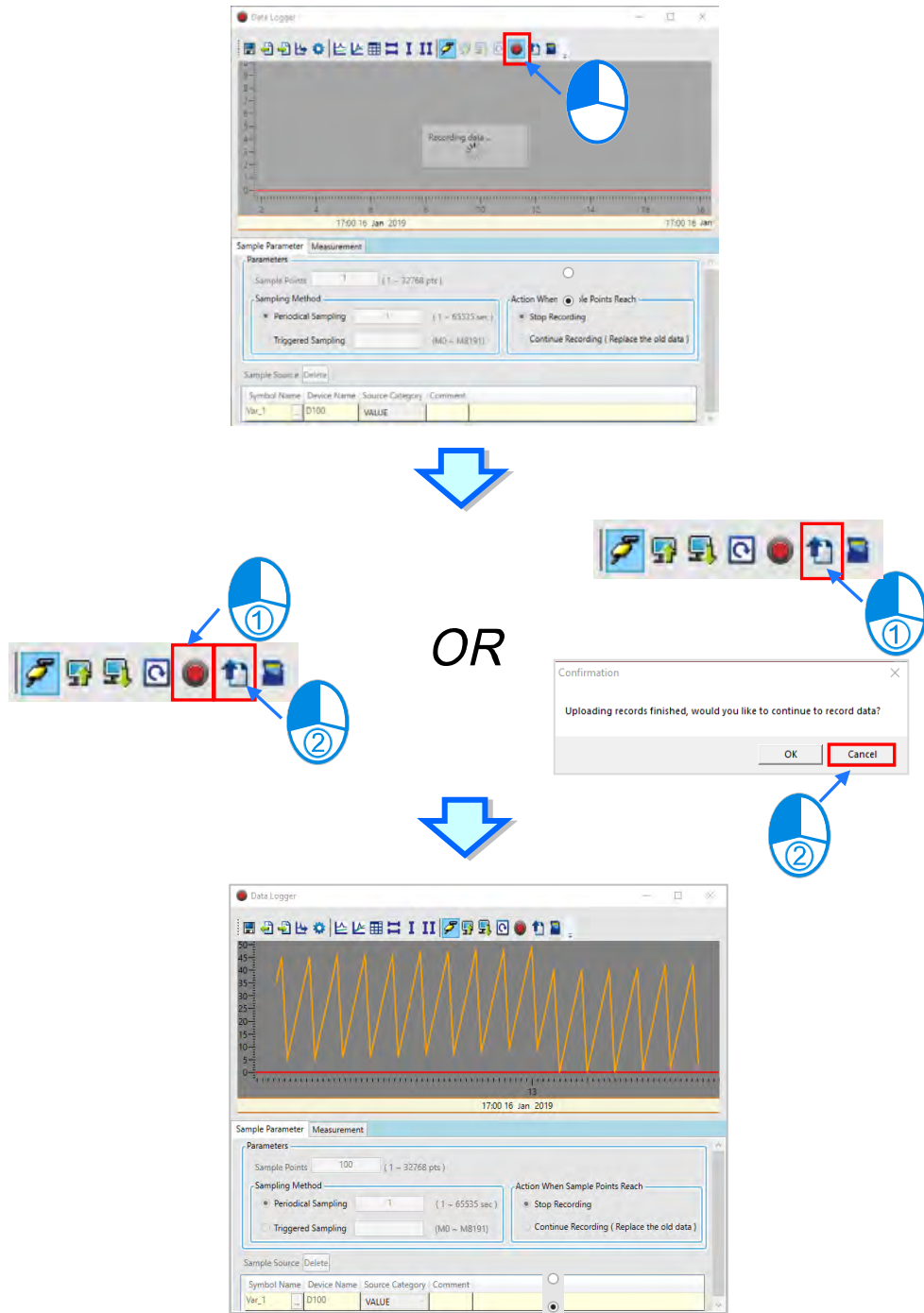
23

- Record Data:** To record symbol variables or device value status of sample sources and save in the internal memory of hosts. Click **Record Data** to start recording. Use **When Sample Points Reach to Stop Recording**, a pop-up window appears to confirm whether or not to upload records. Click OK and a confirmation window pops-up to confirm whether data recording need to be continued, select cancel and the uploaded data appear in the waveform display area.


23



When the sample points reach to **Continue Recording (Replace the old data)**, the hosts will continue to record sample resource data and keep the latest sample point range data. Click record **Record Data** and once recording stops then click **Upload PLC records** or directly click **Upload PLC records** during non-stop recording in the pop-up window to confirm recording again. Click **Cancel** to display recording data from hosts in the waveform display area.



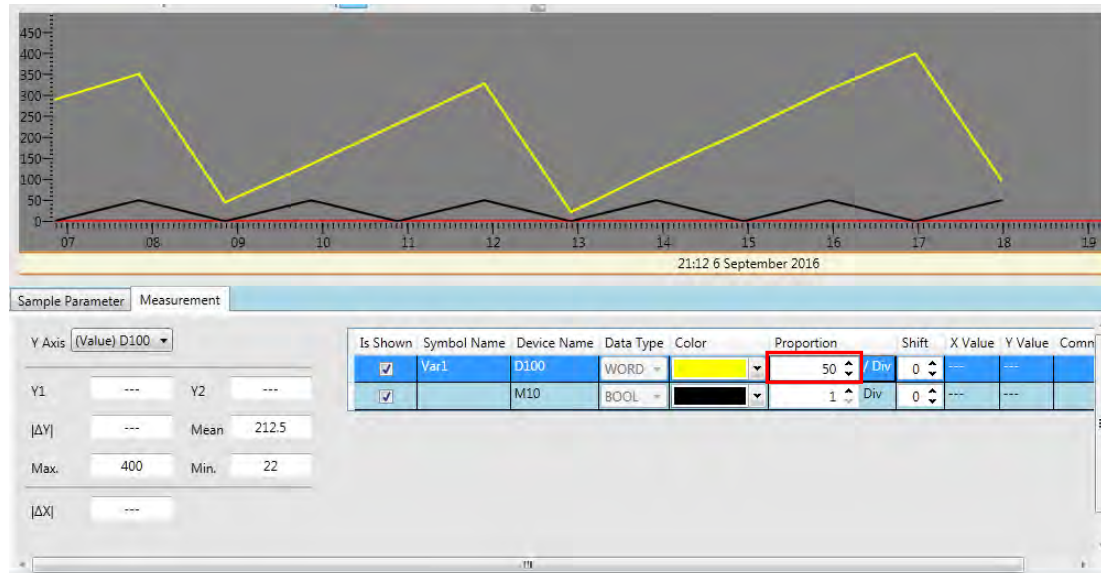
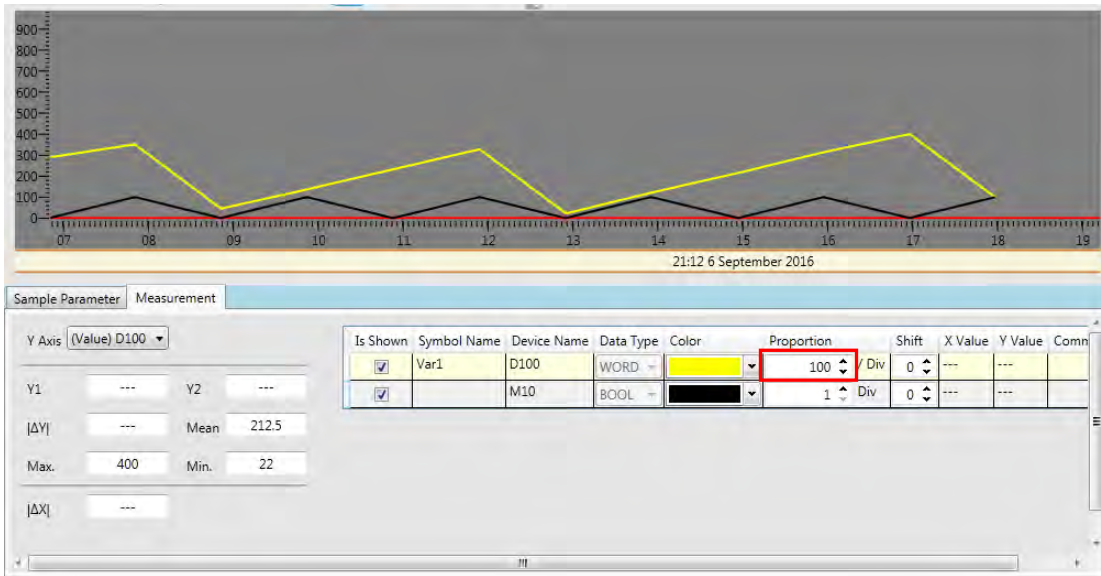
23

Whether recording data or not, users can click Write in Memory Card or  so that recording data from hosts can write in the memory card & content\ PLC **CARD\TypeLog\DATA\_LOGGER\_YMD\_HMS.**

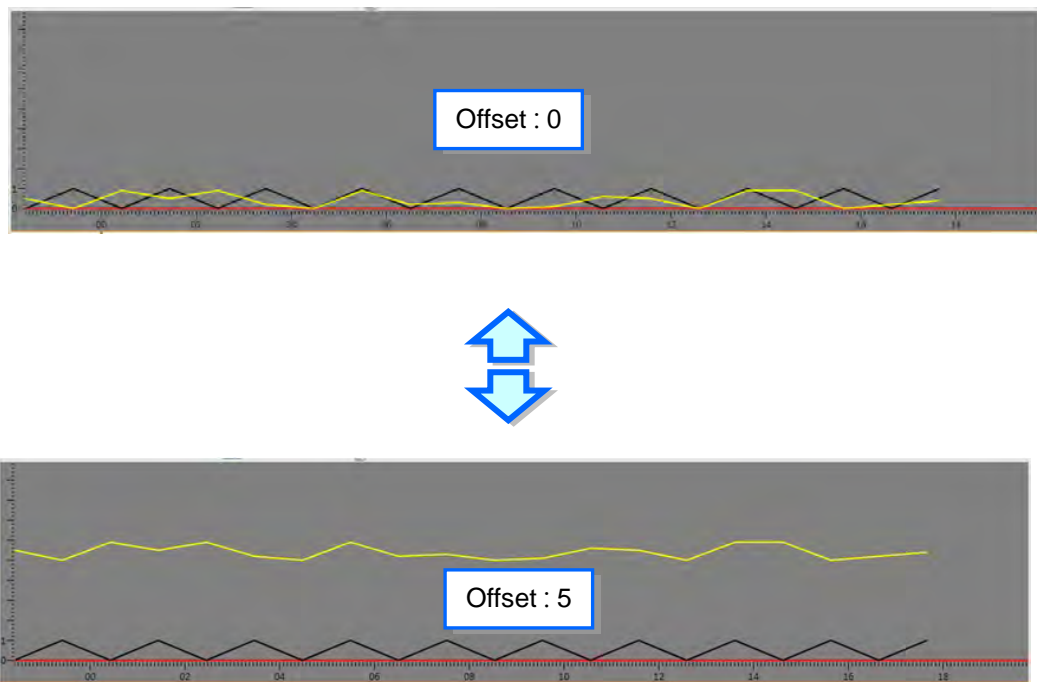
### 23.3.5 Measurement

In Measurement, the page offers the values of mean, maximum and minimum regarding waveform information of selected device in Y axis. Each waveform has different color, proportion and shift settings. The **Proportion** refers to the proportion of vertical coordinate and the following diagram shows proportion of each scale set in 100 and 50 for comparison.

23



To set **Shift**, users can adjust the position of the waveforms in the screen (see below).

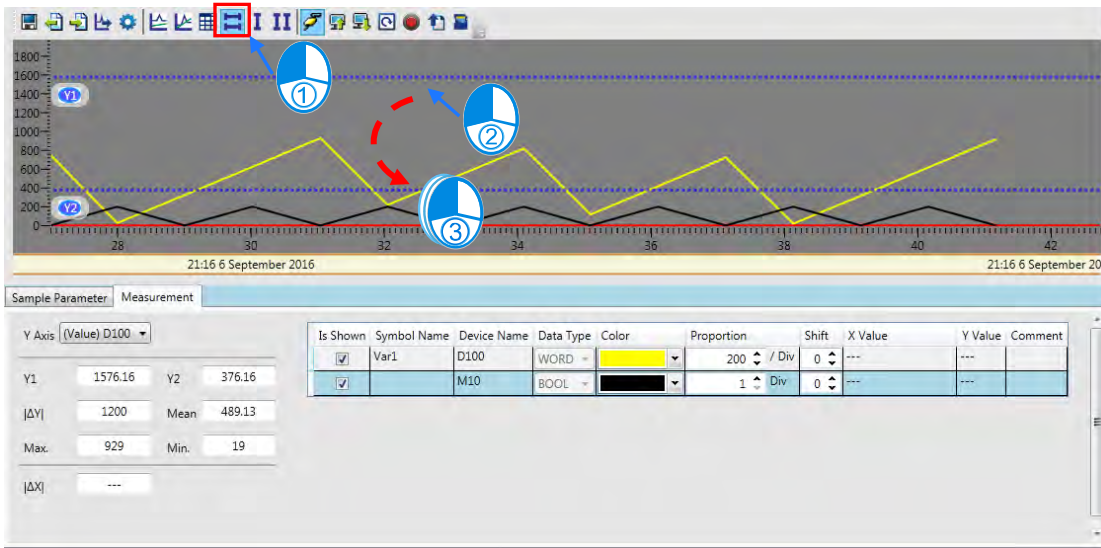


23

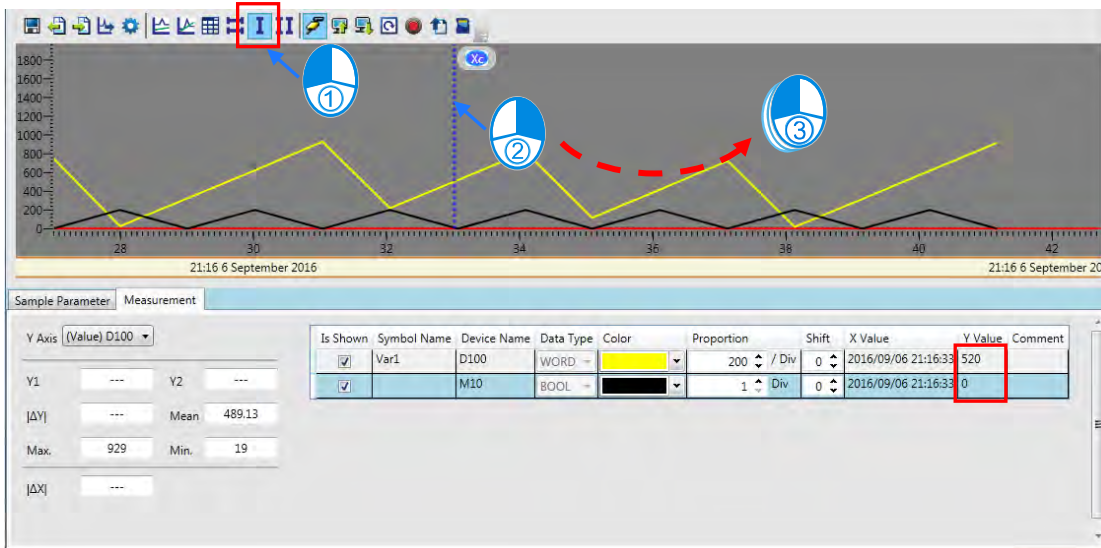
The waveform values in the data logger can be measured by using horizontal and vertical measuring lines after the **Measurement** tab is clicked. Click **H** to open the horizontal measuring line and **V** to open the vertical measuring line.

Is Shown	Symbol Name	Device Name	Data Type	Color	Proportion	Shift	X Value	Y Value	Comment
<input checked="" type="checkbox"/>	Var1	D100	WORD	Yellow	200	Y Div	2016/09/06 21:16:37	723	
<input checked="" type="checkbox"/>		M10	BOOL	Black	1	Div	2016/09/06 21:16:37	0	

Drag the two horizontal measuring lines Y1 and Y2 and adjust to the measured position together with horizontal measuring lines. The parameter setting area displays every value regarding horizontal measuring lines. The Y1 value refers to Y1 column, Y2 value refers to Y2 column and differences between Y1 and Y2 refers to  **$\Delta Y$**  column.

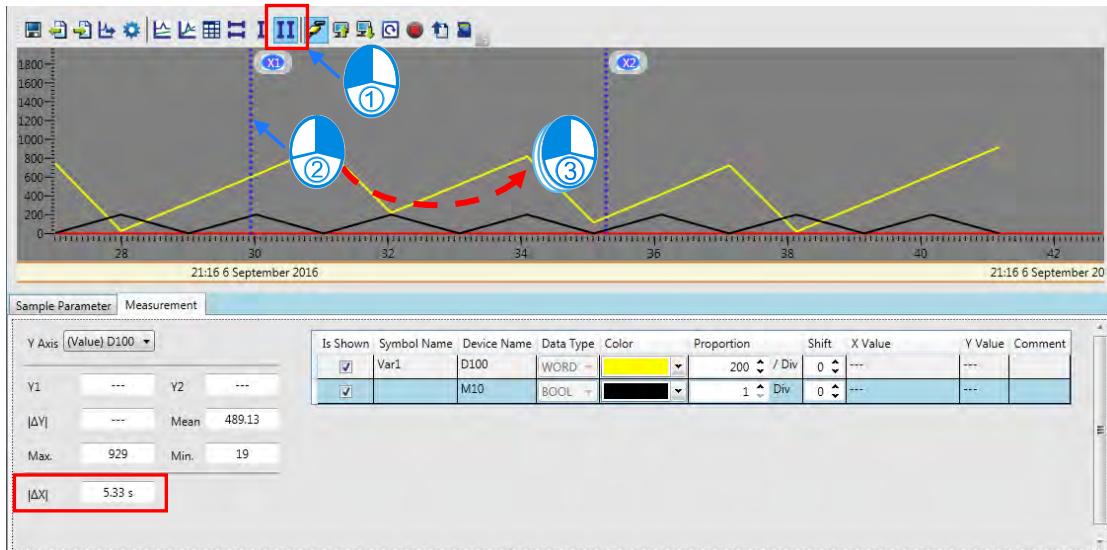


**23** Drag the vertical measuring line Xc to match with other vertical measuring lines. The Y value column below shows an intersected corresponding value regarding each sample resource and measuring line





The following diagram measures the time value of data logger by matching two vertical measuring lines. Click **II** from the toolbar to start the two vertical measuring lines and use the mouse to drag the two vertical lines, X1 and X2 in adjusting to the measured position. Time value is shown in the following  $|\Delta X|$  column.



## 23.4 High Speed Counter

### 23.4.1 About High Speed Counter

The high speed counter tool enables users to select the register number of the corresponding high speed counter based on the type of the high speed counter, format of the value to be counted and input point number so that DCNT instruction can be set quickly.

### 23.4.2 Using High Speed Counter

Double-click DCNT instruction in the ladder diagram to open the **High Speed Counter Setting** window as follows.

23



High Speed Counter Setting

Type: **Hardware - 1 Phase 1 Input** HC200 ~ HC223

Input Point: **X0.0** Counter: **HC200** OK

Arrival Value: **Decimal** **1000** Cancel

5VDC differential inputs  
Maximum differential input frequency: 4MHz

Set Up/Down counter mode by SM300  
Increasing/Decreasing Input point: X0.0

Enable/disable the reset input function by SM291  
Reset input point: X0.12  
(Note: If the PLC does not support X0.12-X0.15, write a reset-related instruction in the program of external interrupts to clear the counters.)

Set the counting mode by SR190  
(Note: The counting mode must be set before the DCNT instruction be enabled.)

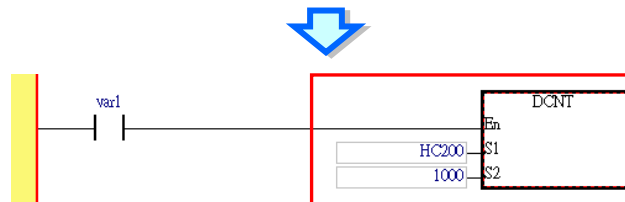
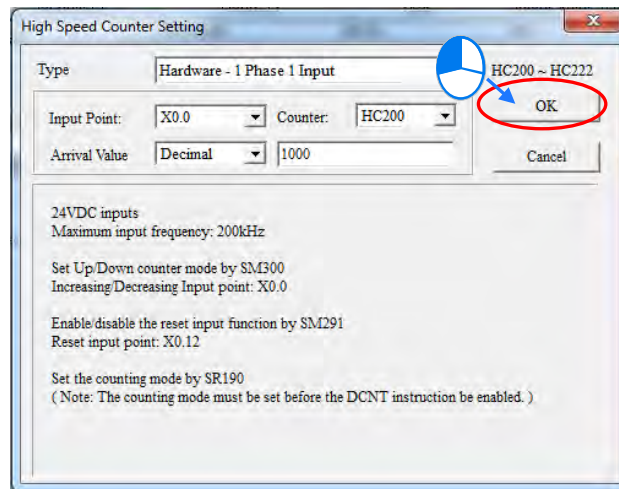
Firstly, select the type of the high speed counter such as **Hardware-1 Phase 1 Input**, **Hardware 1 Phase 2 Input** and so on. Then select **Input Point** or **Counter** device (see below).

Input Point: **X0.0** Counter: **HC200**

Set the arrival value with options including **Decimal**, **Hex** or **D** device from the drop-down menu and input the value or D device number on the right.

Arrival Value: **Decimal** **1000**

Click OK and the high speed counter number and setting values are put in DCNT instruction.



**MEMO**

23



# Appendix A USB Connection

## Table of Contents

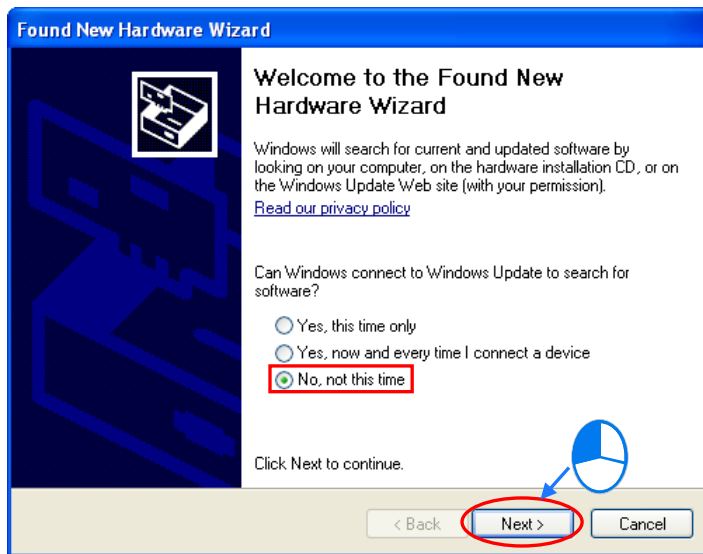
<b>A.1</b>	<b>Installing the USB Driver for an AS Series CPU module .....</b>	<b>A-2</b>
A.1.1	Installing the USB Driver in Windows XP with SP3.....	A-2
A.1.2	Installing the USB Driver in Windows 7.....	A-6
A.1.3	Installing the USB Driver in Windows 8.1 .....	A-11
A.1.4	Installing the USB Driver in Windows 10.....	A-14
<b>A.2</b>	<b>Create USB Driver in COMMGR.....</b>	<b>A-18</b>
<b>A.3</b>	<b>Setting the USB Port on a DVP-SX2 Series PLC.....</b>	<b>A-20</b>

## A.1 Installing the USB Driver for an AS Series CPU module

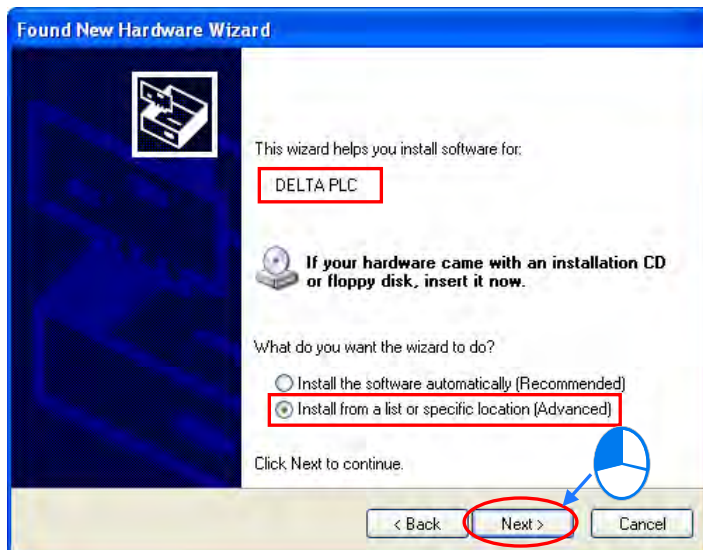
### A.1.1 Installing the USB Driver in Windows XP with SP3

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.

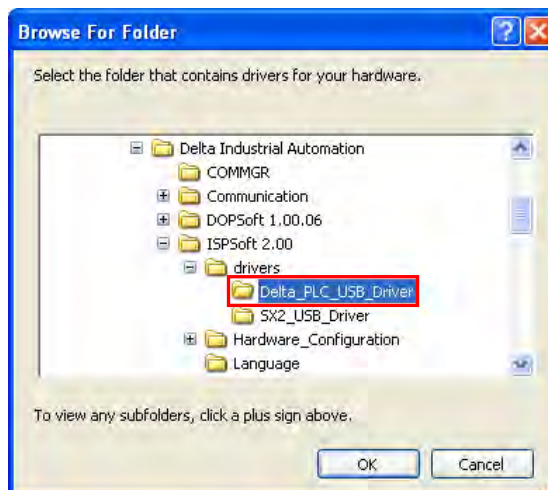
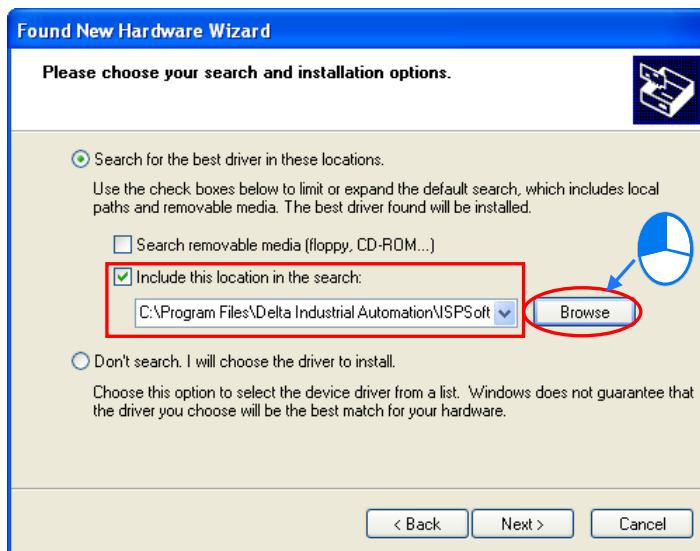


- (3) The Delta PLCs which are equipped with built-in USB interfaces are listed in the table below. After ISPSOft version 2.00 or above is installed, the drivers for DVP-SX2 series PLCs, DVP-SE series PLCs, and AH500 series CPU modules will be in the folders denoted by the paths in the table.

Model	Path
DVP-SX2, AH560EN2	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\

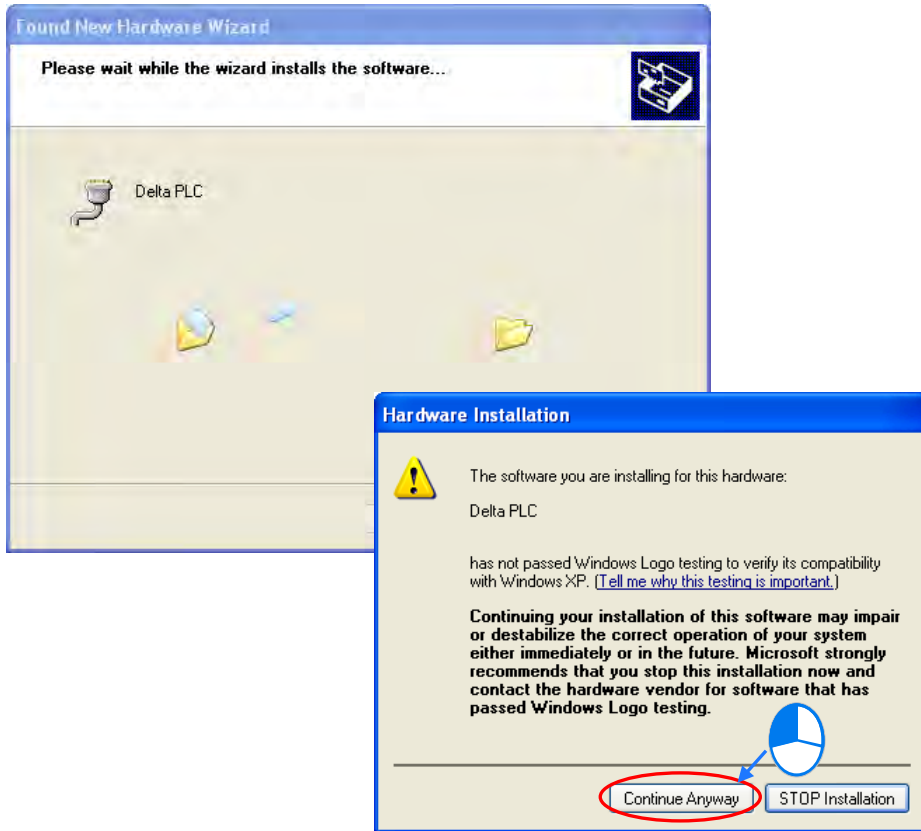
\*. The default installation path of ISPSOft is C:\Program Files(x86)\Delta Industrial Automation\ISPSOftx.xx.

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.



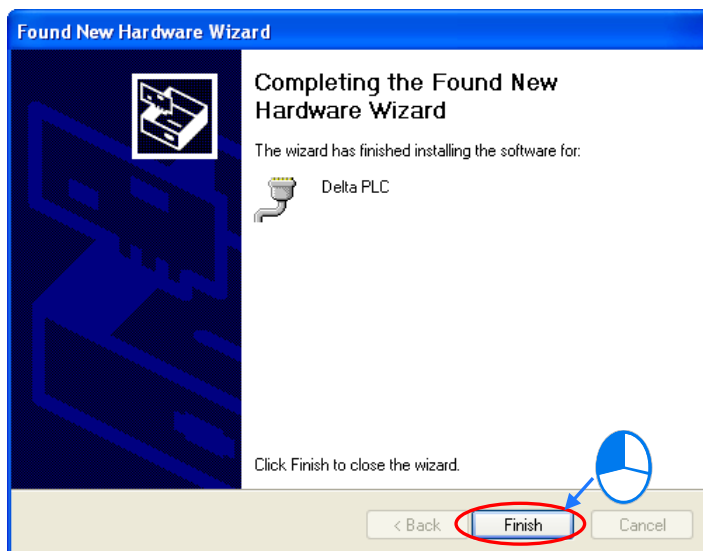
A

- (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**.



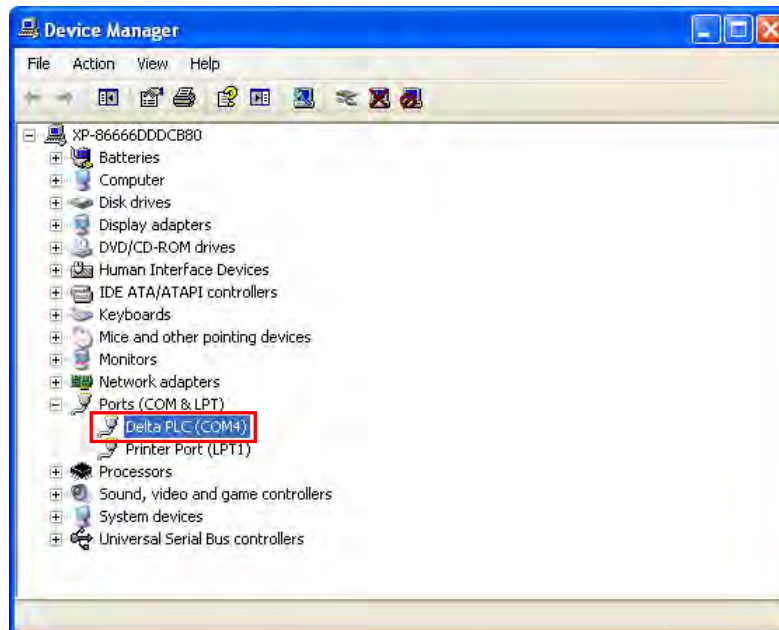
A

- (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.



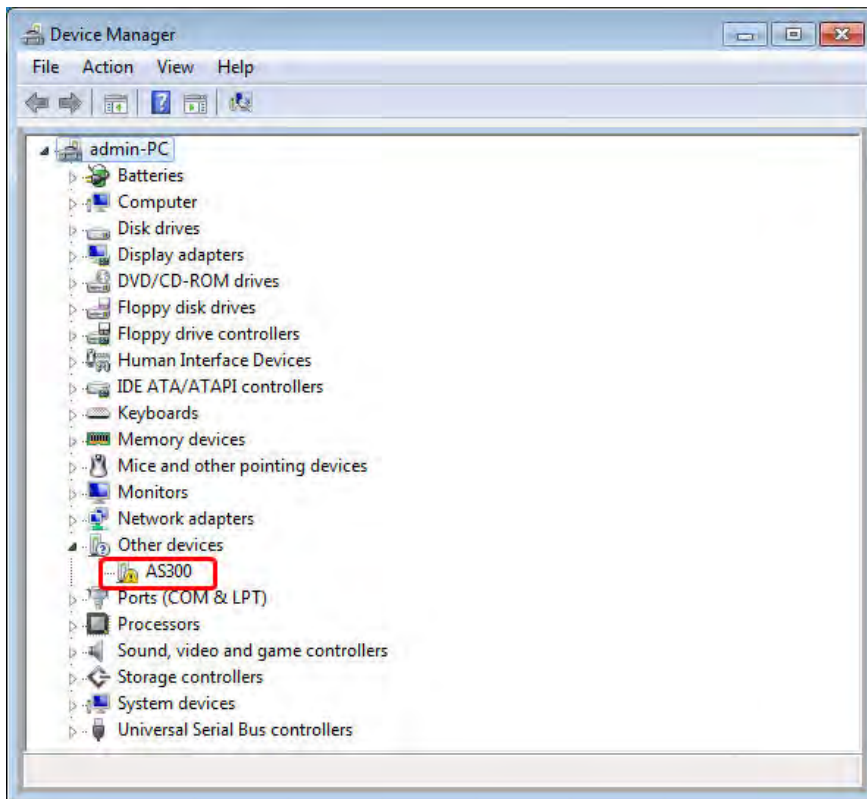
### **Additional remark**

- 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.

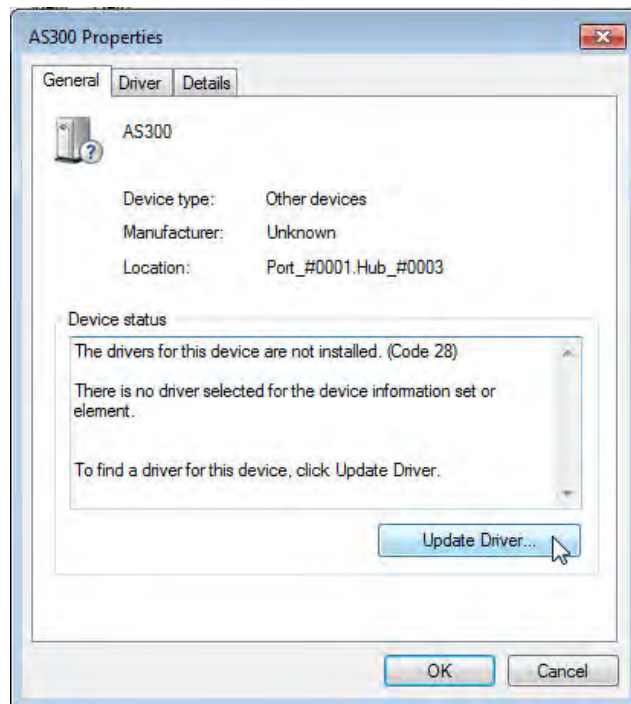
### A.1.2 Installing the USB Driver in Windows 7

The installation of the USB driver for an AS series CPU module on Windows 7 is introduced below. If users want to install the USB driver for an AS series CPU module on another operating system, they have to refer to the instructions in the operating system for more information about the installation of new hardware.

- Make sure that the AS series CPU module is supplied with power normally. Connect the AS series CPU module to a USB port on the computer with a USB cable.
- The name of the USB device detected will be displayed in the Control Panel > Device Manager window. Please select and double-click AS300.

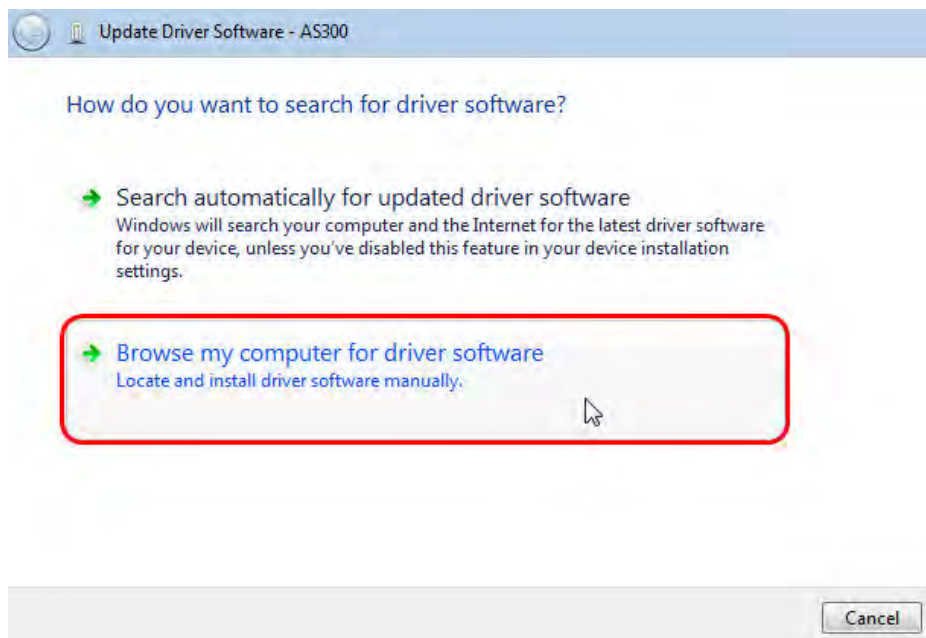


- Click **Update Driver....** in the **AS300 Properties** window.

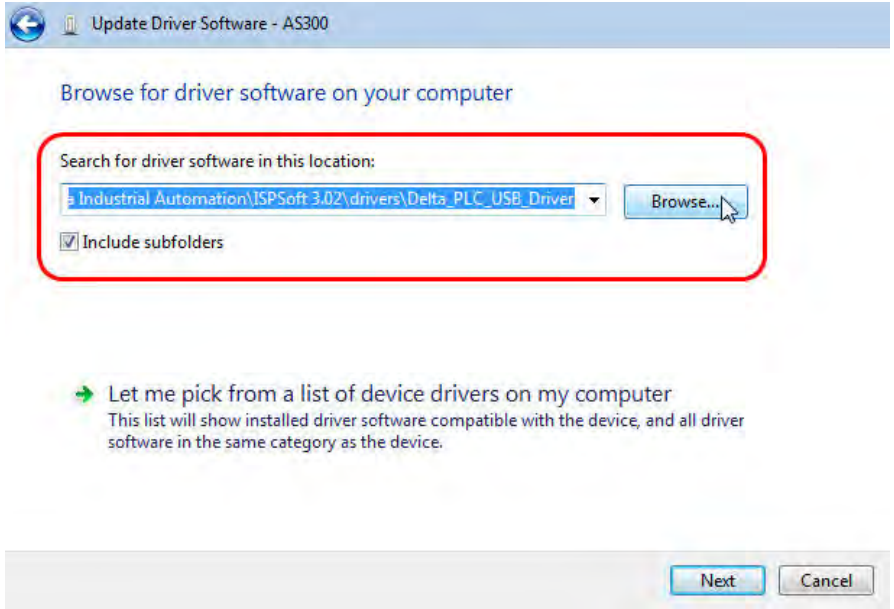


A

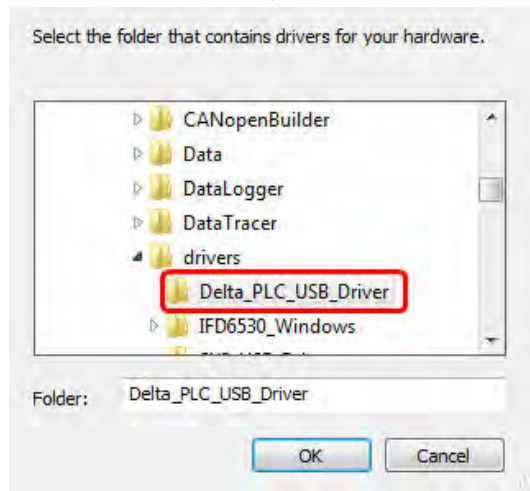
- Click **Browse my computer for driver software.**



- After ISPSOft version 3.00 or above is installed, the driver for the AS series CPU module will be in the folder denoted by the path Installation path of **ISPSOft \drivers\Delta\_PLC\_USB\_Driver\**.
- Specify the correct path. If the driver for the AS series CPU module is gotten in another way, users have to specify the corresponding path. Click **Next** to carry on the installation.



A

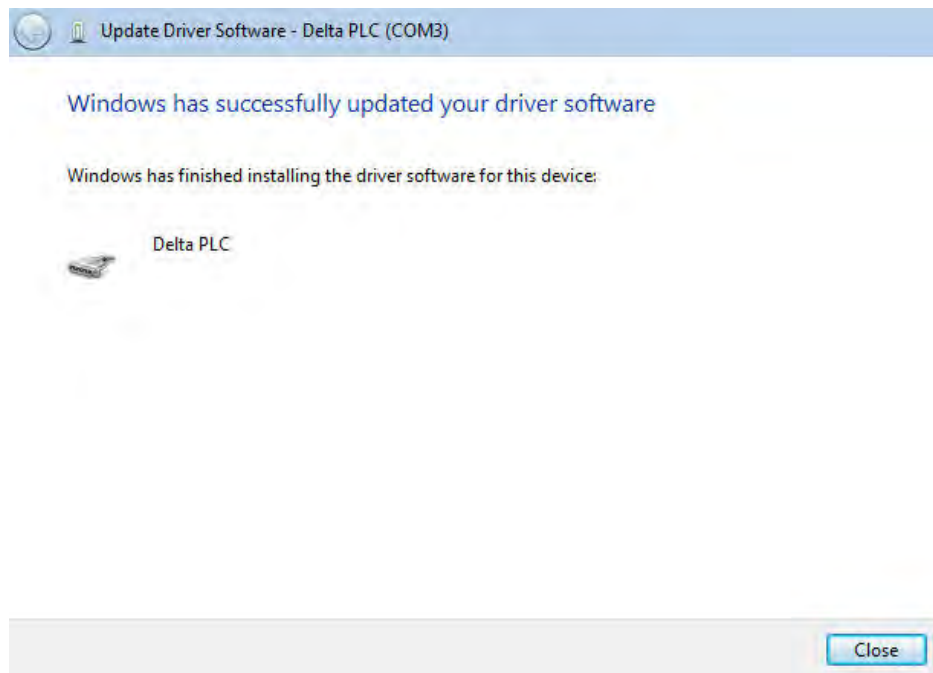


- After the correct driver is found in the folder denoted by the path, the system will install the driver. If the **Windows Security** window appears during the installation, please click **Install this driver software anyway**.

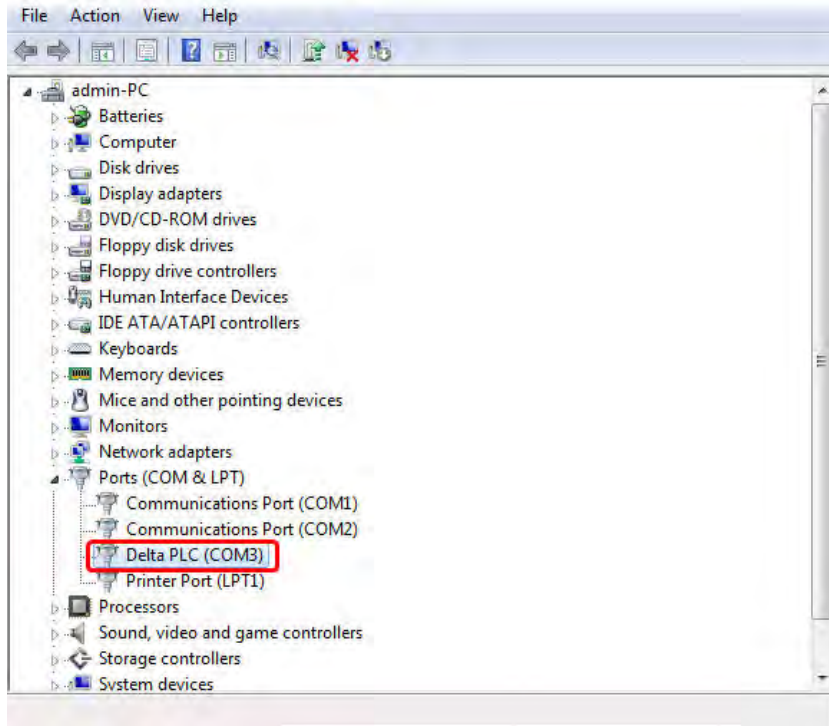


A

- Click **Close** after the installation is finished.



- 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.




**Additional remark**

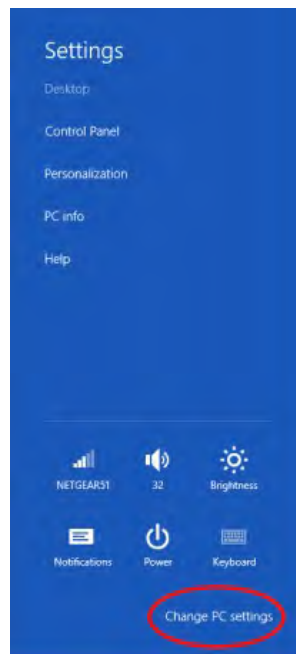
- If the PLC is connected to another USB port on the computer, the system may ask users to install the driver again. The users can follow the steps above, and install the driver again. After the driver is installed, the communication port number that the operating system assigns to the USB device may be different.

### A.1.3 Installing the USB Driver in Windows 8.1

Windows 8.1 driver signature enforcement provides a way to improve the security of the operating system by validating the integrity of a driver or system file each time it is loaded into memory. However since Delta PLC USB driver does not include the driver signature, this section will help users to disable driver signature enforcement functionality in Windows 8.1 to ensure a success Delta PLC USB installation. This act is only valid for a single time. The setting will return to its original state after restarting.

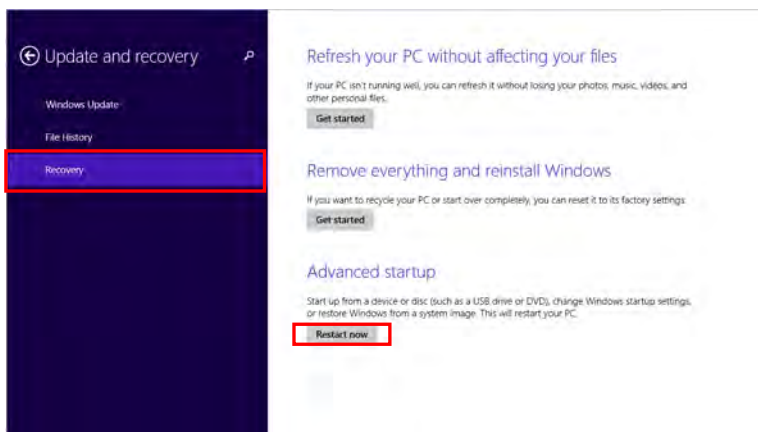
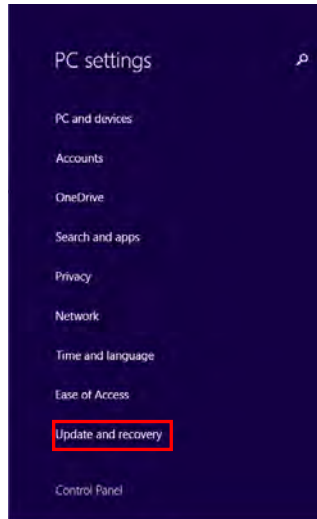
Steps to disable driver signature enforcement in Windows 8:

1. Press the button  **[WIN]+[I]** on the keyboard to see the Setting interface. Click "Change PC settings".



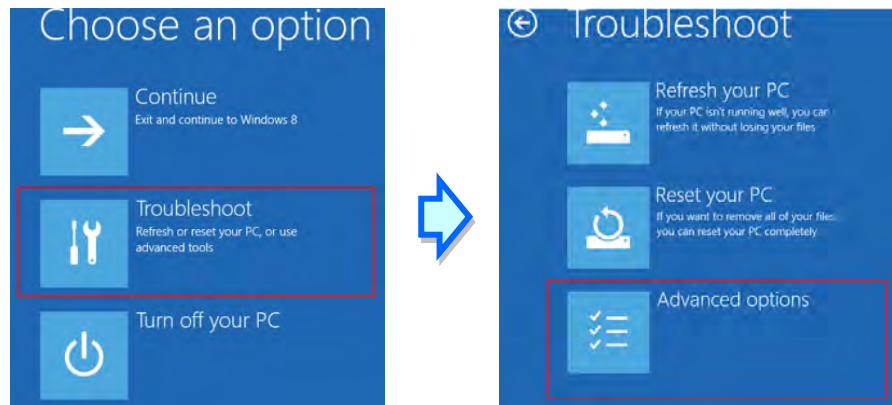
A

2. The PC settings window will appear. Select “General” and then “Restart now” under “Advanced startup”.

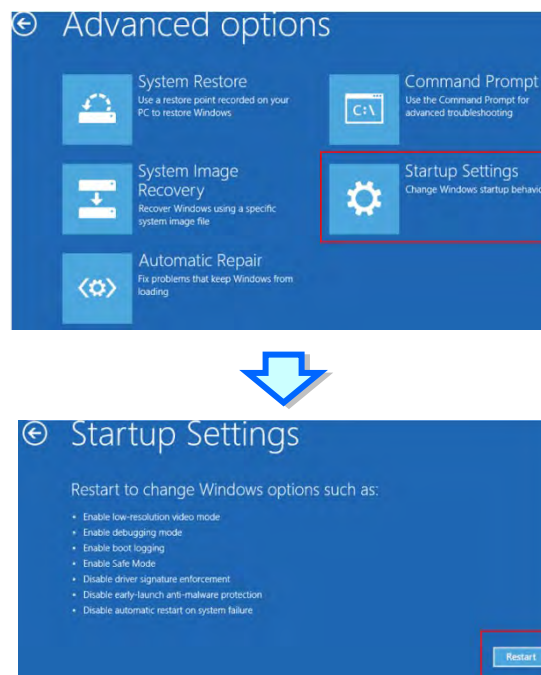




3. After the computer is restarted, select “Troubleshoot” under “Choose an option”. And then select “Advanced options”.



4. From the Advanced options page, select “Startup Settings” to see the Startup Settings. From this page select “Restart” to restart the computer.



A

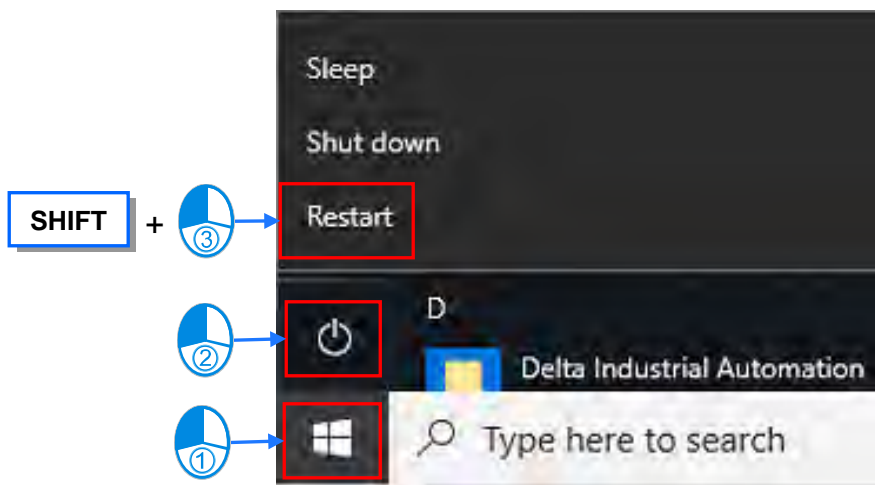
5. Press “7” or “F7” to choose “Disable driver signature enforcement” and the system will direct you to the Windows 8 operating page. Users can then install the Delta PLC USB driver now.
6. For the Delta USB drive installation, please refer to installation in Windows 7 section.

### A.1.4 Installing the USB Driver in Windows 10

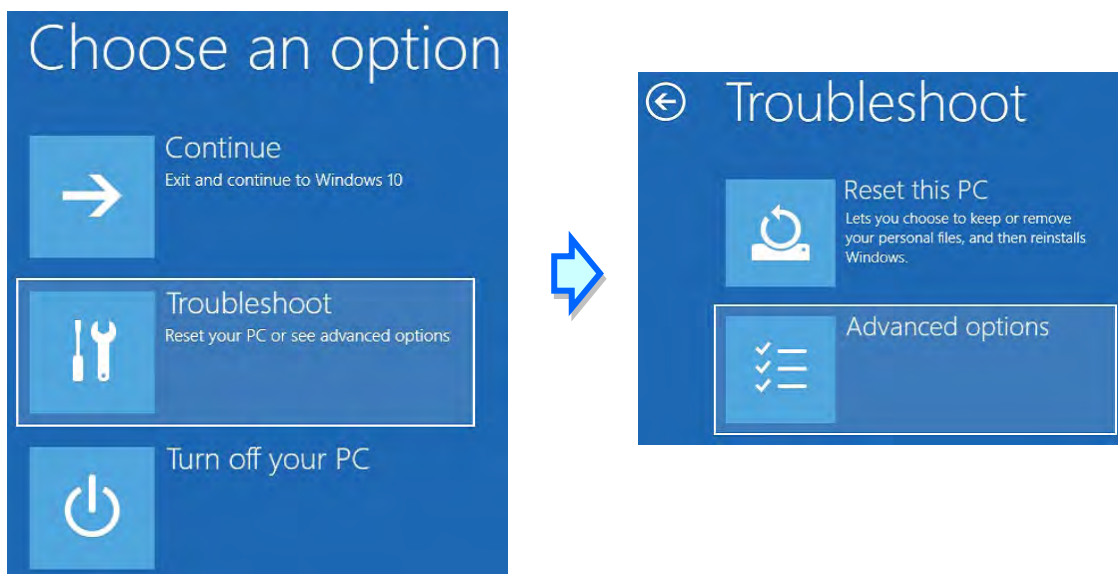
Windows 10 driver signature enforcement provides a way to improve the security of the operating system by validating the integrity of a driver or system file each time it is loaded into memory. However since Delta PLC USB driver does not include the driver signature, this section will help users to disable driver signature enforcement functionality in Windows 10 to ensure a success Delta PLC USB installation. This act is only valid for a single time. The setting will return to its original state after restarting.

Steps to disable driver signature enforcement in Windows 10:

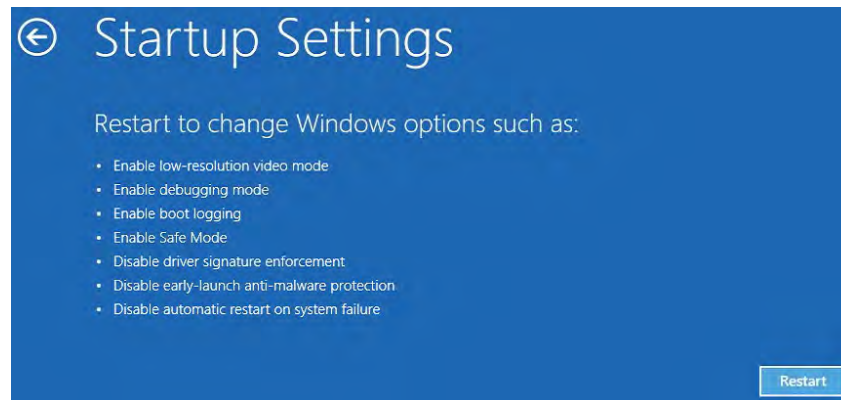
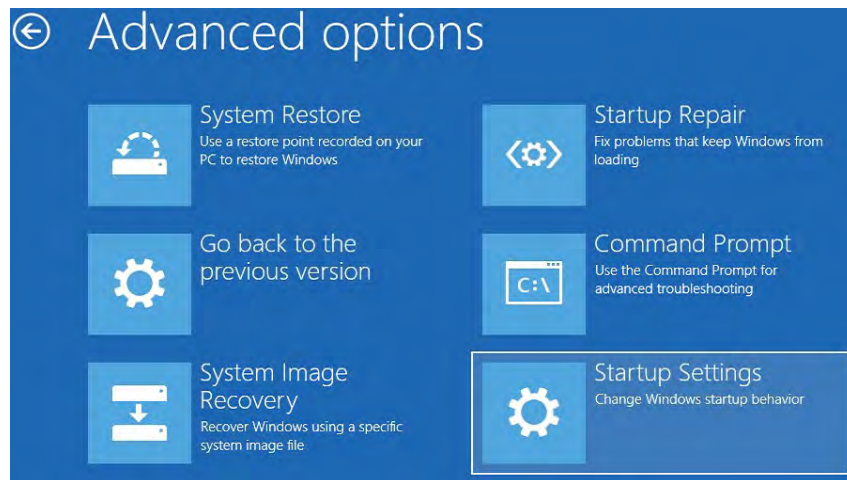
1. Click Start and then Settings. Then, press SHIFT key and click Restart



2. After the computer is restarted, select Troubleshoot, then choose “Advanced options”.



3. Select Startup Settings from the Advanced options page, then click “Restart” .



4. In Startup Setting page, press 7 or F7 to execute “Disable driver signature enforcement” and the system will restart once completed.

## Startup Settings

Press a number to choose from the options below:

Use number keys or functions keys F1-F9.

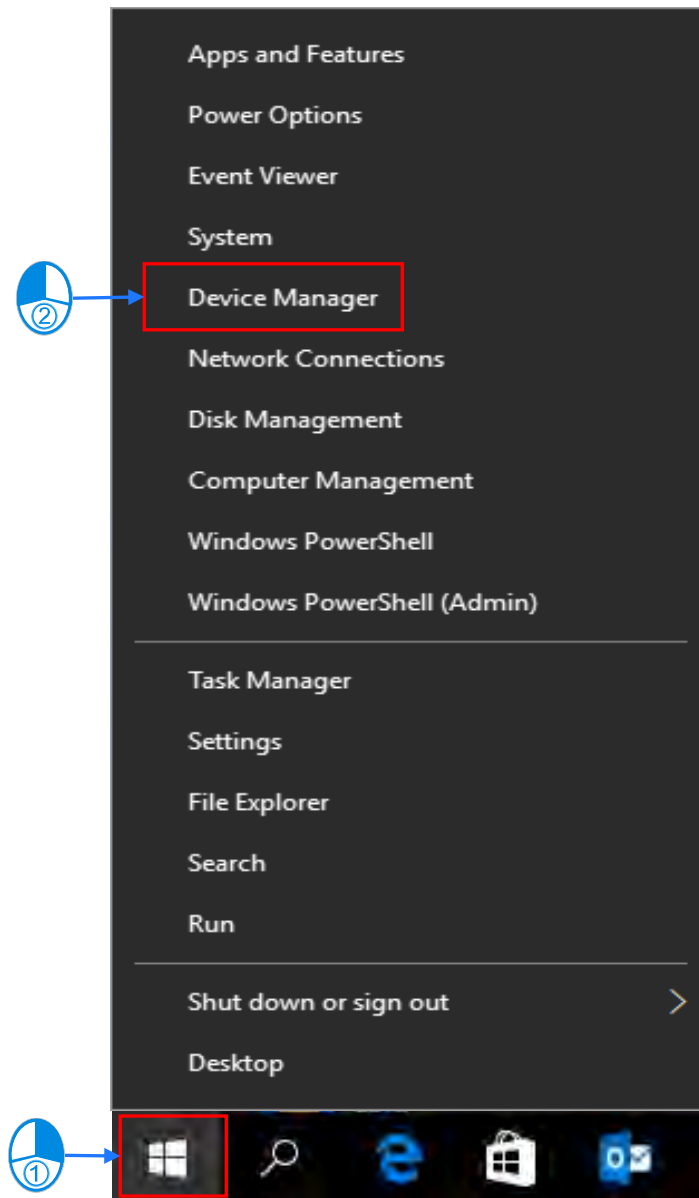
- 1) Enable debugging
- 2) Enable boot logging
- 3) Enable low-resolution video
- 4) Enable Safe Mode
- 5) Enable Safe Mode with Networking
- 6) Enable Safe Mode with Command Prompt
- 7) Disable driver signature enforcement
- 8) Disable early launch anti-malware protection
- 9) Disable automatic restart after failure

Press F10 for more options

Press Enter to return to your operating system



5. To install Delta USB drive, right-click Start and select Device Manager.

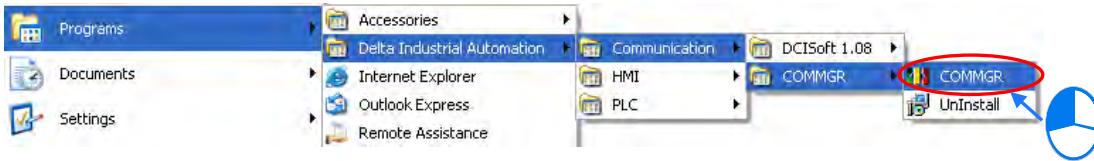


6. Please refer to A1.2 for installing USB drive (Delta\_PLC\_USB\_Driver) in Window 7.

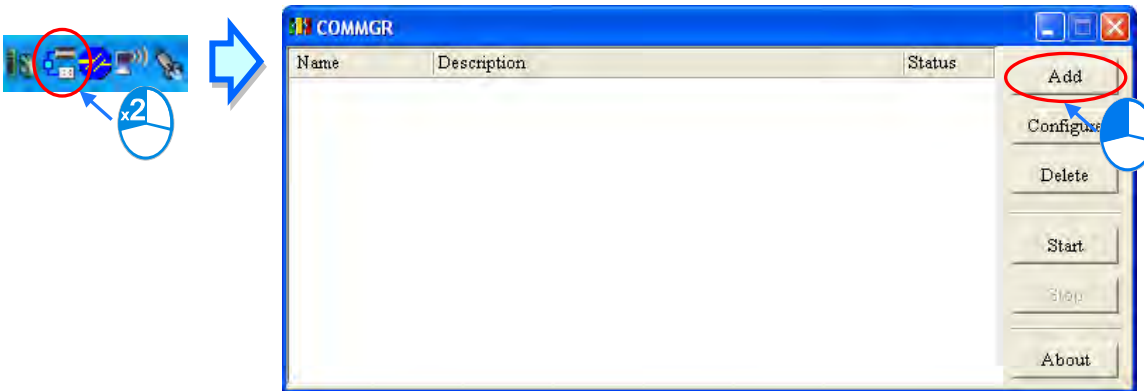
## A.2 Create USB Driver in COMMGR

The steps of creating a driver whose connection type is USB in COMMGR are as follows.

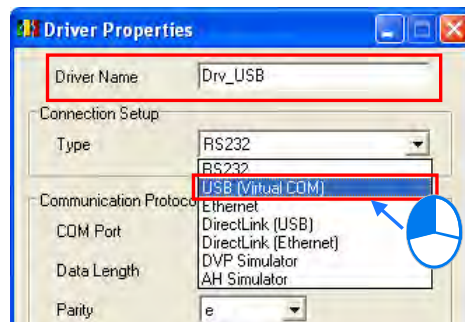
- (1) Make sure that the PLC is supplied with power normally. Connect the PLC to a USB port on the computer with a USB cable.
- (2) Make sure that COMMGR is started, and the icon representing COMMGR is displayed on the system tray. If the icon representing COMMGR is not displayed on the system tray, users can start COMMGR by clicking the shortcut on the **Start** menu (**Start>Programs>Delta Industrial Automation>Communication>COMMGR**).



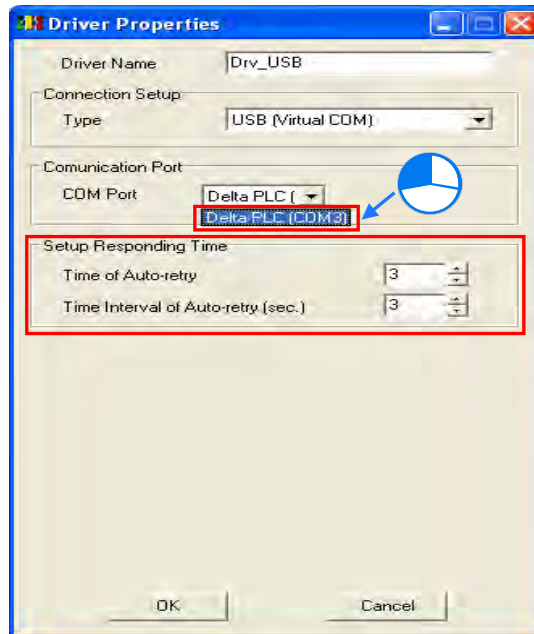
- (3) Double-click the icon representing COMMGR on the system tray, and then click **Add** in the **COMMGR** window.



- (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.



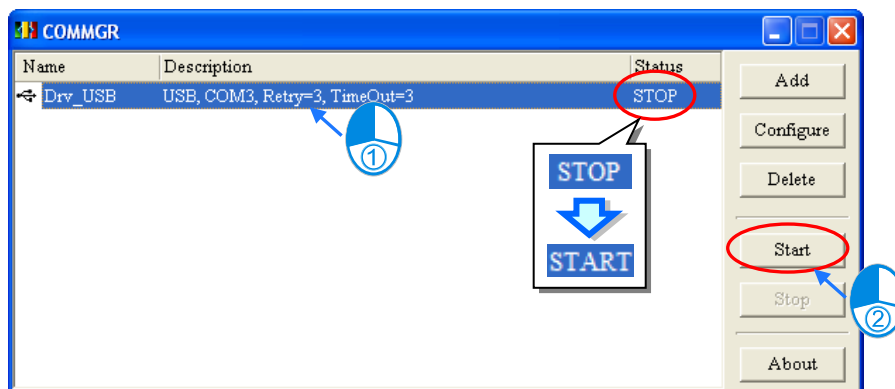
- (5) Select a device name which corresponds to the PLC connected in the **COM Port** drop-down list box in the **Communication Port** section, select the number of times the sending of a command is retried if a connection error occurs in the **Time of Auto-retry** box, and select an interval of retrying the sending of a command in the **Time Interval of Auto-retry** box. Click **OK**.



A

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**.

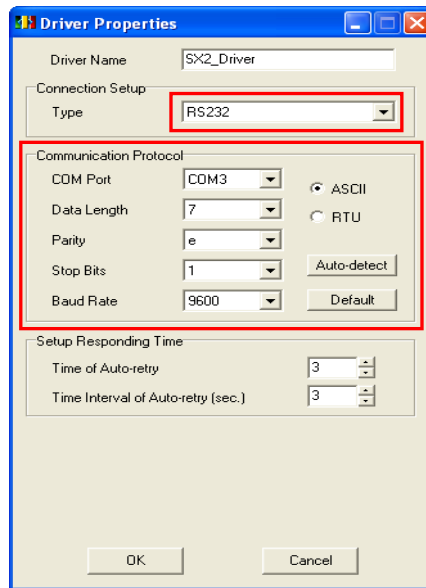
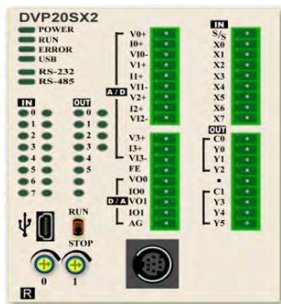


\*. Please refer to section 2.4 for more information about subsequent tests and the operation of COMMFR.

### 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.







# Appendix B Notification for PLC Types

## Table of Contents

<b>B.1</b>	<b>Device Addresses for PLC Types</b> .....	<b>B-2</b>
B.1.1	AH/AS Series Device Types .....	B-2
B.1.2	The X/Y/D/L Device Address Format in AH/AS Series.....	B-4
B.1.3	Real-time Access of X/Y Contact in AH/AS Series .....	B-6
B.1.4	DVPxxMC Device Types.....	B-7
B.1.5	DVPxxMC Series Device Address Format .....	B-8
B.1.6	DVPxxMC Retentive Devices .....	B-8
<b>B.2</b>	<b>Device Resources for PLC Types</b> .....	<b>B-9</b>
B.2.1	AHCPU500-EN/AHCPU500-RS2 .....	B-9
B.2.2	AHCPU510-EN/AHCPU510-RS2 .....	B-10
B.2.3	AHCPU520-EN/AHCPU520-RS2 .....	B-11
B.2.4	AHCPU530-EN/AHCPU530-RS2 .....	B-12
B.2.5	AHCPU501-EN/AHCPU501-RS2 .....	B-13
B.2.6	AHCPU511-EN/AHCPU511-RS2 .....	B-14
B.2.7	AHCPU521-EN/AHCPU521-RS2 .....	B-15
B.2.8	AHCPU531-EN/AHCPU531-RS2 .....	B-16
B.2.9	AHCPU560-EN2 .....	B-17
B.2.10	AHxxEMC .....	B-18
B.2.11	AS200/AS300 .....	B-19
B.2.12	DVPxxMC .....	B-20
<b>B.3</b>	<b>Compile &amp; Uploading/Downloading Time</b> .....	<b>B-20</b>

AH/AS series PLCs 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 AH/AS series CPU module are described in appendix B.

## B.1 Device Addresses for PLC Types

### B.1.1 AH/AS Series Device Types

The device types of AH/AS 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 does not conform 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 does not conform to the setting value.



Device type	Access property	Description
	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.
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)

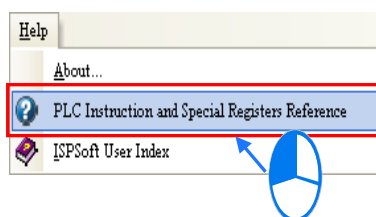
\*1. AH/AS series CPU modules support the manipulation of the bits in the input relays, the output relays, the data registers, and the link registers. For example, X0.1 indicates that bit 1 in X0 is manipulated, Y0.1 indicates that bit 1 in Y0 is manipulated, D0.1 indicates that bit 1 in D0 is manipulated, 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. There is no link register for AS series PLC.

\*3. The index registers that AH/AS series CPU modules have are E devices. AH500 series CPU modules do not have F devices.

\*4. AH/AS 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 Programming Manual or **PLC Instructions and Special Registers Reference** in a project for a CPU module in ISPSOft for more information.





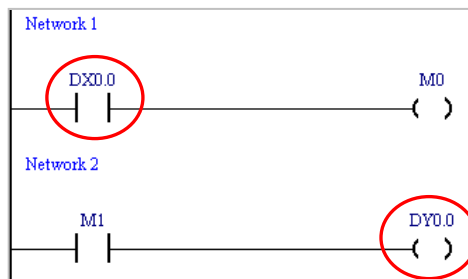
If the bits in the X/Y/D/L devices which are word devices in an AH500 series CPU module are manipulated, the memory blocks that the X/Y/D/L devices occupy are manipulated. However, the memory blocks that the T/C/HC devices used as bit devices occupy are different from the memory blocks that the T/C/HC 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 does not conform 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 Real-time Access of X/Y Contact in AH/AS Series

The way in which the input/output in an AH/AS 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 AH/AS 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.

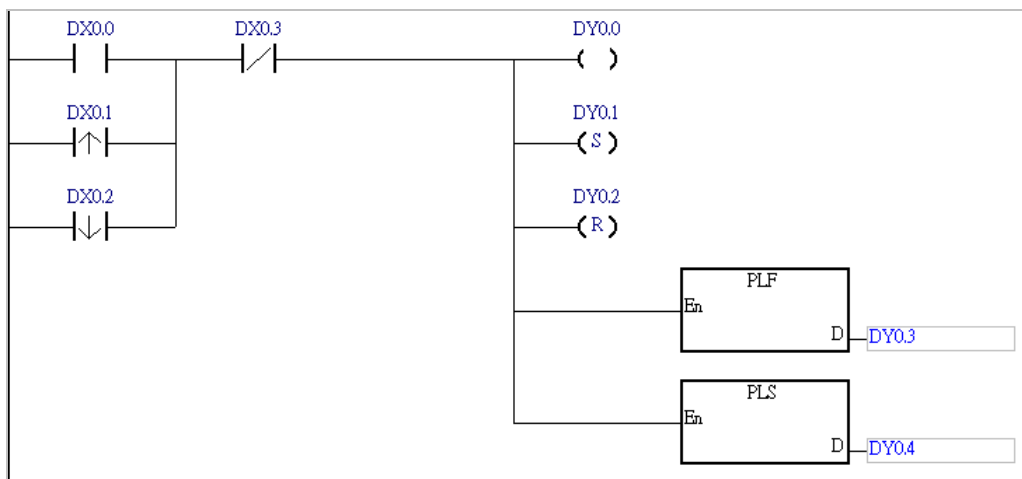


The instructions which support the modification by “D” are listed in the following table. If an X device which is modified by “D” or a Y device which is modified by “D” is used in an instruction which is not listed in the table below, the system will regard it as incorrect syntax, and an error will occur 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”.

B

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.



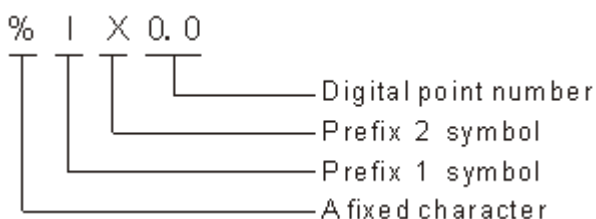
The figure below is an example of applying the modification by “D” illegally. The Y device is modified by “D”. It can not be used as an input contact.



### B.1.4 DVPxxMC Device Types

- **Device Name Explanation**

Unlike AH/AS or DVP series, DVPxxMC series adopts standard IEC (see below).



- **Device List used in DVPxxMC Software**

No.	Item	Content				
1	Prefix 1 symbol	I	Q	M		
2	Prefix 1 name	Input device	Output device	Intermediate device		
3	Prefix 2 symbol	X	B	W	D	L
4	Data type of prefix 2	BIT	BYTE	WORD	DWORD	QWORD
5	Device example	%IX0.0	%IB0	%IW0	%ID0	%IL0
6		%QX0.0	%QB0	%QW0	%QD0	%QL0
7		%MX0.0	%MB0	%MW0	%MD0	%ML0

B

### B.1.5 DVPxxMC Series Device Address Format

- **The Corresponding Relationships of Devices**

The table below shows %ML0 includes %MB0~%MB7, %MD0 includes %MB0~%MB3 and %MW0 includes %MB0~%MB1.

Device name	Corresponding relationships																							
	The 1 <sup>st</sup> WORD						The 2 <sup>nd</sup> WORD						The 3 <sup>rd</sup> WORD						The 4 <sup>th</sup> WORD					
	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15
%MX	%MX0.0~0.7			%MX1.0~1.7			%MX2.0~2.7			%MX3.0~3.7			%MX4.0~4.7			%MX5.0~5.7			%MX6.0~6.7			%MX7.0~7.7		
%MB	%MB0			%MB1			%MB2			%MB3			%MB4			%MB5			%MB6			%MB7		
%MW	%MW0						%MW1						%MW2						%MW3					
%MD	%MD0												%MD1											
%ML	%ML0																							

Below, the table shows %ML1 includes %MB8~%MB15, %MD2 includes %MB8~%MB11, %MW4 includes %MB8~%MB9 and %MB8 includes %MX8.0~8.7.



Device name	Corresponding relationships																							
	The 5 <sup>th</sup> WORD						The 6 <sup>th</sup> WORD						The 7 <sup>th</sup> WORD						The 8 <sup>th</sup> WORD					
	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15
%MX	%MX8.0~8.7			%MX9.0~9.7			%MX10.0~10.7			%MX11.0~11.7			%MX12.0~12.7			%MX13.0~13.7			%MX14.0~14.7			%MX15.0~15.7		
%MB	%MB8			%MB9			%MB10			%MB11			%MB12			%MB13			%MB14			%MB15		
%MW	%MW4						%MW5						%MW6						%MW7					
%MD	%MD2												%MD3											
%ML	%ML																							

### B.1.6 DVPxxMC Retentive Devices

The retentive devices include %MW0~%MW999. Also, variables in the software can set to retentive or non-retentive. The retentive volume is 100K byte.



## B.2 Device Resources for PLC Types

### 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**

<b>Device type</b>	<b>Address range</b>	
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	
SR	SR0~SR2047	
SM	SM0~SM2047	
Number of memory blocks which can be assigned to function block instances	1024	



**B.2.5 AHCPU501-EN/AHCPU501-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~D24575	Bit: D0.0~D24575.15
L	Word: L0~L24575	Bit: L0.0~L24575.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	



**B.2.6 AHCPU511-EN/AHCPU511-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~D49151	Bit: D0.0~D49151.15
L	Word: L0~L49151	Bit: L0.0~L49151.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4096	
SM	SM0~SM4096	



**B.2.7 AHCPU521-EN/AHCPU521-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~D98303	Bit: D0.0~D98303.15
L	Word: L0~L98303	Bit: L0.0~L98303.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	



**B.2.8 AHCPU531-EN/AHCPU531-RS2**

Device type	Address range	
X	Word: X0~X1023	Bit: X0.0~X1023.15
Y	Word: Y0~Y1023	Bit: Y0.0~Y1023.15
D	Word: D0~D131071	Bit: D0.0~D131071.15
L	Word: L0~L131071	Bit: L0.0~L131071.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4095	
SM	SM0~SM4095	





**B.2.9 AHCPU560-EN2**

Device type	Address range	
X	Word: X0~X4095	Bit: X0.0~X4095.15
Y	Word: Y0~Y4095	Bit: Y0.0~Y4095.15
D	Word: D0~D262143	Bit: D0.0~D262143.15
L	Word: L0~L262143	Bit: L0.0~L262143.15
M	M0~M8191	
S	S0~S4095	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4095	
SM	SM0~SM4095	



**B.2.10 AHxxEMC**

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	
SR	SR0~SR2047	
SM	SM0~SM2047	



**B.2.11 AS200/AS300**

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~D29999	Bit: D0.0~D29999.15
M	M0~M8191	
S	S0~S2047	
C	C0~C511	
HC	HC0~HC255	
T	T0~T511	
E	E0~E14	
SR	SR0~SR2047	
SM	SM0~SM4095	



### B.2.12 DVPxxMC

Device type	Address range
%IX	%IX0.0~%IX127.7
%QX	%QX0.0~%QX127.7
%MX	%MX0.0~%MX131071.7
%IB	%IB0~%IB127
%QB	%QB0~%QB127
%MB	%MB0~%MB131071
%IW	%IW0~%IW63
%QW	%QW0~%QW63
%MW	%MW0~%MW65535
%ID	%ID0~%ID31
%QD	%QD0~%QD31
%MD	%MD0~%MD32767
%IL	%IL0~%IL15
%QL	%QL0~%QL15
%ML	%ML0~%ML16383

## B

### B.3 Compile & Uploading/Downloading Time

The duration for ISPSOft to compile programs is involved with the size, quantity of the programs and the variable symbols and the condition of the computer. The duration to upload/download is involved with the size of the project, the connection type and the communication speed. The following information is listed for reference.

Operation System on the computer: CPU Intel(R) I7, RAM 4GB, Windows 7 32-bit

Project contents: 2 variable symbols and a basic ladder diagram with a total of 5000 networks; each network contains 1 contact and 1 output coil.

Program compilation time: about 2 seconds

Program capacity: 10K STEP

Time for uploading: about 4.9 seconds via USB port

Time for downloading: about 4.6 seconds via USB port

# Appendix C Print Management Tool

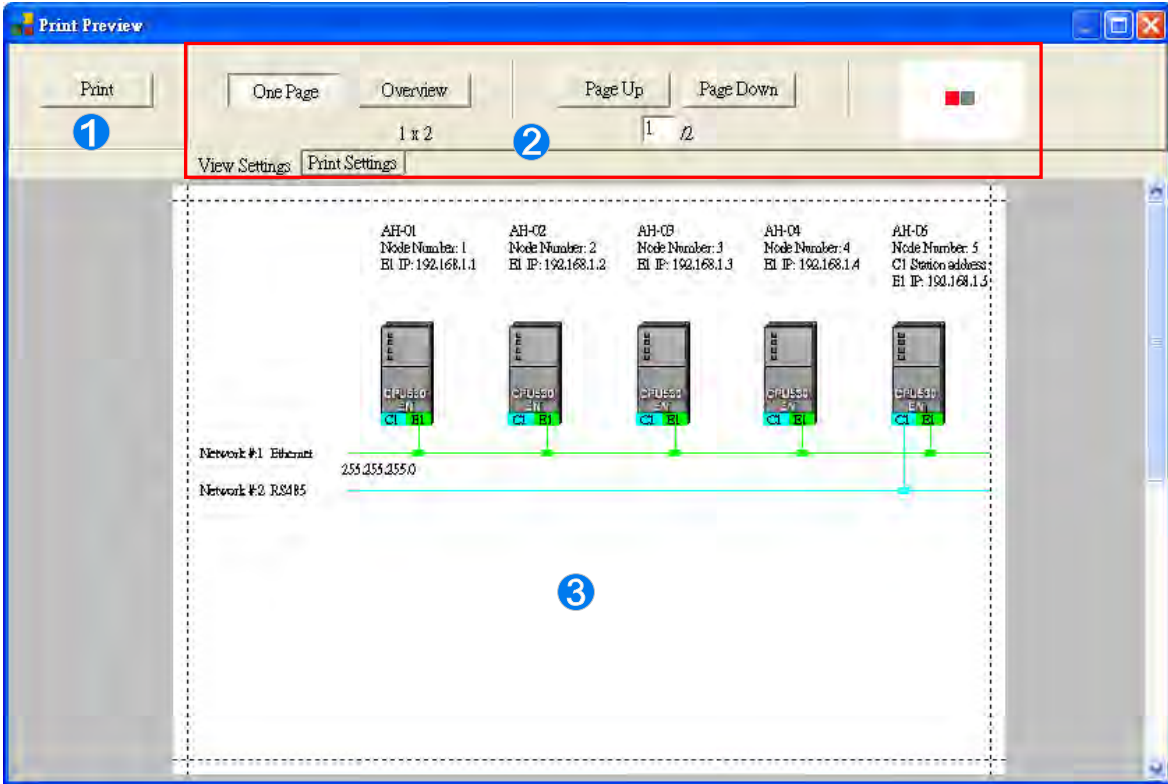


## Table of Contents

C.1	Introduction of the Environment .....	C-2
C.2	Introduction of the Setting Area .....	C-4

## C.1 Introduction of the Environment

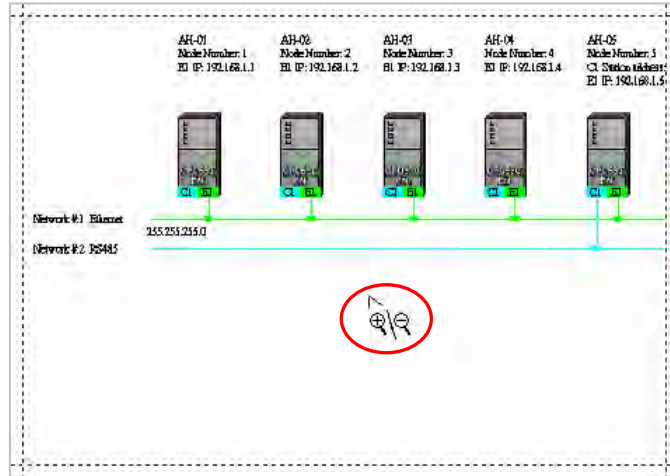
The print management tool provides a convenient operation interface. Users can set a print format, and preview the document which will be printed in the **Print Preview** window.



- 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.

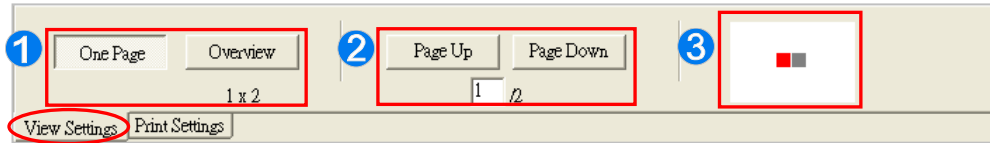


If users move the mouse cursor to the preview area and click the left mouse button, the pointer will become a magnifying glass. After the left mouse button is clicked, ISPSoft will zoom in on the preview area. After the right mouse button is clicked, ISPSoft will zoom out on the preview area.



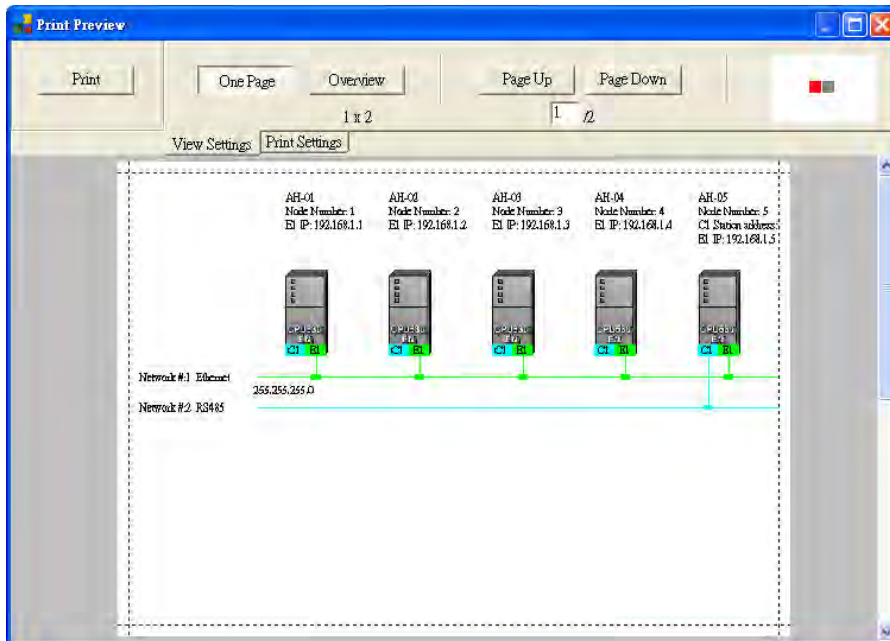
## C.2 Introduction of the Setting Area

After users click the **View Settings** tab, they can adjust the display state of the preview area.



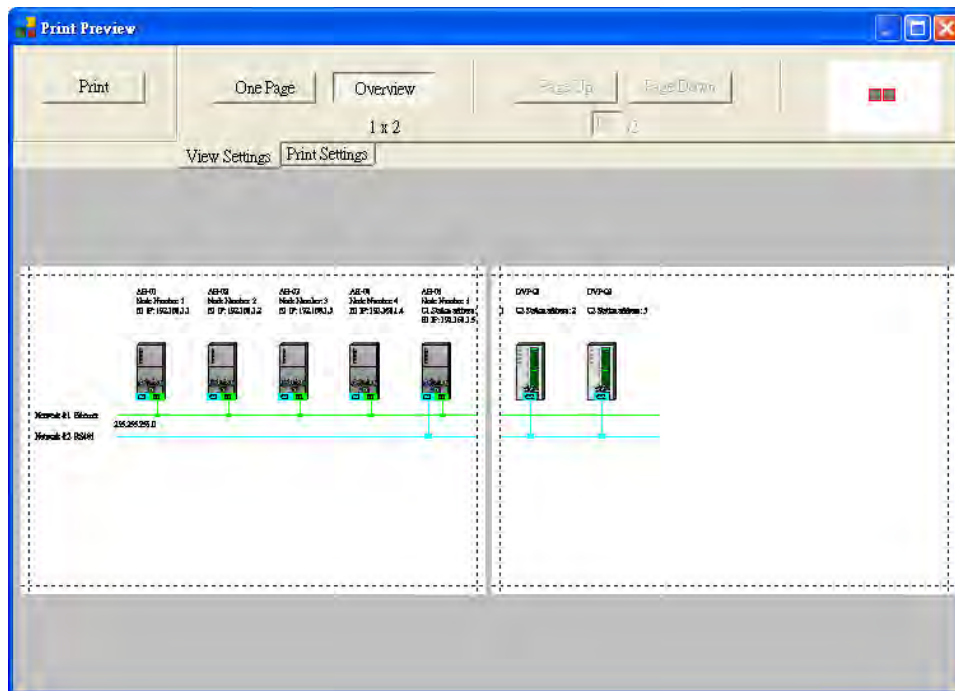
- 1 If the print area does not fit the printed page, the system will divide the print area into several pages. After **One Page** is clicked, one page will be displayed. After **Overview** is clicked, all the pages will be displayed. The number of pages into which the print area is divided is under the **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

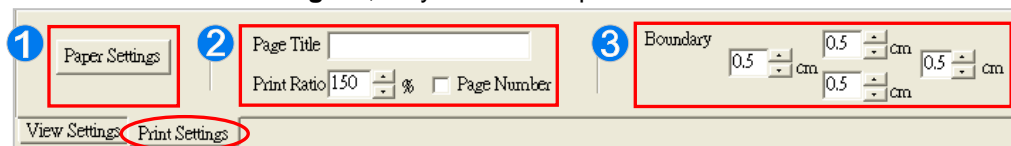




## ● Overview

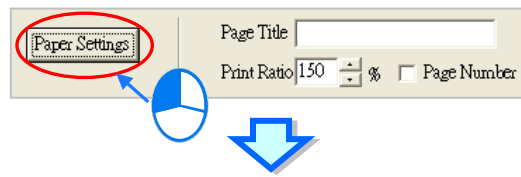


After users click the **Print Setting** tab, they can set the print format.



- 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.

