DIADesigner
Software User Manual

www.deltaww.com

Smarter. Greener. Together.

Copyright Notice

©Delta Electronics, Inc. All rights reserved.

All information contained in this user manual is the exclusive property of Delta Electronics Inc. (hereinafter referred to as "Delta ") and is protected by copyright law and all other laws. Delta retains the exclusive rights of this user manual in accordance with the copyright law and all other laws. No parts in this manual can be reproduced, transmitted, transcribed, translated or used in any other ways without the prior consent of Delta.

Limitation of Liability

The contents of this user manual are only for the use of the DIAStudio manufactured by Delta. Except as defined in special mandatory laws, Delta provides this user manual “as is” and does not offer any kind of warranty through this user manual for using the product, either express or implied, including but not limited to the following:

(i) this product will meet your needs or expectations;
(ii) the information contained in the product is current and correct;
(iii) the product does not infringe any rights of any other person. You shall bear your own risk to use this product.

In no event shall Delta, its subsidiaries, affiliates, managers, employees, agents, partners and licensors be liable for any direct, indirect, incidental, special, derivative or consequential damages (including but not limited to the damages for loss of profits, goodwill, use or other intangible losses) unless the laws contains special mandatory provisions to the contrary.
Delta reserves the right to make changes to the user manual and the products described in the user manual without prior notice and afterwards

Related Documents

Document Name	Document ID
Software Download and Installation Manual	DIAS-Manual-0005-EN
DIAScreen User Manual	DIAS-Manual-0004-EN

Revision History

Version	Issue Date	Revision
1st	2020/12/08	The first version release.

Chapter 1: Overview 1

1.1 DIAStudio Integrated Engineering Software 2

1.1.1 DIAStudio Overview 2

1.2 DIADesigner Overview 3

Chapter 2: Getting Started 5

2.1 Support Devices 6

2.2 Device and Function Guide 6

2.3 System Requirement 7

2.4 Quick Start and example 8

2.4.1 Scenario 1 8

2.4.2 Scenario 2 10

Chapter 3: Installation and Update 11

3.1 Installation and Uninstallation 12

3.1.1 Installing DIADesigner or COMMGR 12

3.2 Launch DIADesigner 13

Chapter 4: User Interface 14

4.1 UI Introduction 15

4.2 Basic Designer 16

4.2.1 Basic Designer Tabs 17

4.2.2 Project 18

4.2.3 Device 23

4.2.4 Programming 26

4.2.5 Debug 29

4.2.6 Setting 31

4.2.7 Guide 33

4.3 Advanced Designer 41

4.3.1 Menu bar 45

4.3.2 Standard Toolbar 48

4.3.3 Editor Toolbar 49

4.3.4 Development Toolbar 49
4.3.5 Communication Toolbar 51
4.3.6 Project Explorer 51
4.4 Creating Project and Device 54
4.4.1 Add Device 55
4.4.2 Device Name 59
4.4.3 Update Device 62
Chapter 5: Communication Setting 65
5.1 Communication via COMMGR 66
5.2 Communication without COMMGR 67
5.3 COMMGR (Communication Manager) 69
5.3.1 Overview 70
5.3.2 Launch COMMGR 71
5.3.3 Main Screen 72
5.3.4 Driver Setting 77
5.3.5 Driver Start/Stop 95
5.3.6 Driver Search 96
5.3.7 Bypass Setting 100
5.3.8 Communication Setting of Inverter with Built-in PLC 102
Chapter 6: Setting Device (Hardware Configuration) 106
6.1 Hardware Configuration 107
6.1.1 Creating Controller Configuration 108
6.1.2 Creating Inverter Configuration 130
6.2 Parameter Setting 145
6.2.1 Parameter Settings for PLC devices 145
6.2.2 DTC parameter editor 148
6.2.3 Parameter Settings for Servo devices 151
6.2.4 Parameter Settings for Inverter/ASD 154
6.2.5 Perform Parameter Setting After Servo Drives/Inverters Upload Parameters 157
Chapter 7: Setting Network (Network Configuration) 162
7.1 Network View 163

7.1.1 Network View Description 163
7.1.2 Creating Network View in Advanced View 165
7.1.3 Network View Toolbar 168
7.1.4 Network Parameter Setting 172
7.2 Data Exchange Table 172
7.2.1 Exchanging Data through Data Exchange Table 173
7.2.2 Setting up Data Exchange Table 176
7.2.3 EtherNet/IP Data Exchange Table 178
7.2.4 CANopen Data Exchange Table 179
7.2.5 MODBUS/MODBUS TCP Data Exchange Table 181
Chapter 8: Programming 182
8.1 IEC61131-3 Programming 183
8.2 POU, VAR, Task 184
8.2.1 POU 184
8.2.2 VAR 212
8.3 Programming Language Introduction (LD, SFC, ST) 265
8.3.1 LD – Ladder Diagram 265
8.3.2 SFC - Sequential Function Chart 301
8.3.3 ST - Structured Text 339
8.3.4 C Language Editor (C Language) 379
8.4 Compile and Check 407
8.4.1 Check 407
8.4.2 Compile 408
8.5 Program Monitoring 409
8.5.1 LD Program Monitoring 410
8.5.2 ST Program Monitoring 411
8.5.3 SFC Program Monitoring 411
8.6 Online Edit 412
8.6.1 POU Online Edit 413
8.7 Debug Mode 425
8.7.1 Enable Debug Mode 425

8.7.2 Break Point 426
8.7.3 Continue Run Execution 429
8.7.4 Single-Step Execution 430
Chapter 9: Library Management tools 432
9.1 Library Project 433
9.1.1 Create Library Project 433
9.1.2 Project Information 436
9.1.3 Repository project 437
9.2 Library Manager 438
9.2.1 Resource Library 440
9.2.2 Add Library 441
9.2.3 Delete Library 442
9.2.4 Library Usage 442
Chapter10: Motion Tools 445
10.1 Position Plan Table 446
10.1.1 About Position Plan Table 446
10.1.2 Setting the Position Plan Table 446
10.1.3 Axis Setting 447
10.1.4 Password 449
10.1.5 Position Table 452
10.1.6 Position Plan Table Simulation 454
10.1.7 Upload and Download Position Plan Table 457
10.1.8 Control mode 459
Chapter 11: HMI Interaction 468
11.1 Tag Sharing to DIAScreen 469
Chapter 12: Simulation 474
12.1 PLC Simulation 475
Chapter 13: Download / Upload and Online / Offline Mode 477
13.1 Batch Download and Upload 478
13.1.1 Download 478
13.1.2 Upload 480
13.1.3 Download/Upload Components 482

13.2 Online / Offline Mode 483
Chapter 14: Commission and Diagnostic 484
14.1 Device Information 485
14.2 Monitor Table 486
14.2.1 Creating a Monitor Table 487
14.2.2 Adding Variable and Registers 491
14.2.3 Monitor Table Context Menu 493
14.2.4 Using Monitor Table 495
14.3 Tuning Tool 496
14.3.1 Calculate the Inertia Ratio 497
14.3.2 Set up the bandwidth 500
14.4 Trace and Oscilloscope 501
14.4.1 Data Tracer 502
14.4.2 Data Logger 510
14.4.3 Servo Device Oscilloscope 518
14.4.4 Inverter Device Trace 525
14.5 Communication Test Tools 532
Chapter 15: Temperature control tool 540
15.1 Monitor 541
15.1.1 Turn on the thermostat monitoring function 541
15.1.2 Monitor Screen 542
15.1.3 Monitoring settings 544
15.2 Program planning 550
15.2.1 Program and step editing 553
15.2.2 Status display area 554
15.3 Station number automatic synchronization 555
15.3.1 Start the station number automatic synchronization function 555
15.3.2 Turn off the station number automatic synchronization function 556
Chapter 16: Auxiliary Tools 559
16.1 Register Comment 560
16.1.1 Adding items in Address Comment window 562

16.1.2 Deleting items in Register Comment window 564
16.1.3 Deleting Register Comments 564
16.1.4 Deleting comments for selected Register type 564
16.1.5 Deleting All Comments 565
16.2 Register Usage 565
16.3 Resources 568
16.4 Register Edit 569
16.4.1 Register Edit Toolbar 571
16.4.2 Data format 571
16.4.3 Data length 572
16.4.4 Address Edit – Write 574
16.4.5 Address Edit – Read 575
16.4.6 Address Edit – Clear all values 576
16.4.7 Address Edit – Move to 577
16.5 Step Position 578
16.6 Set RTC 579
16.6.1 Updating Computer time to PLC time 581
16.6.2 Updating Custom time to PLC time: 581
16.7 Format PLC Program Memory 582
16.7.1 Format Program Memory 583
16.7.2 Reset PLC Memory (Factory Setting) 584
16.8 Parameter Backup & Restore 586
16.9 Retain Variable Initialization 588
Chapter 17: Other Function 590
17.1 Find and Replace 591
17.1.1 Find – POU 591
17.1.2 Replace – POU 593
17.1.3 Batch Replacement Register 595
17.2 Bookmark 599
17.2.1 Bookmark in Ladder Diagram (LD) POU 600
17.2.2 Bookmark in Structured Text (ST) POU 606

17.2.3 Bookmark function of C language 613
17.3 Option 621
17.3.1 Project Settings 621
17.3.2 Environment Settings 622
17.3.3 Programming Settings 624
Chapter 18: Trouble Shooting 627
18.1 Error Log 628
18.1.1 Error Information 628
18.1.2 PLC Download Log 630
18.1.3 Status Change Log 630
18.2 Error Checklist 631
Chapter 19: Appendix 634
19.1 Installing USB driver for AS series CPU module 635
19.1.1 Installing USB driver in Windows 7 635
19.1.2 Installing USB driver in Windows 8.1 641
19.1.3 Installing USB driver in Windows 10 646
19.2 Important points related to CPU modules 649
19.2.1 Addresses in AS series CPU modules 650
19.2.2 Address resources in AS series CPU modules 654
19.3 Servo Setting 654

Chapter 1: Overview

1.1 DIAStudio Integrated Engineering Software

The DIAStudio is an all-in-one integrated engineering platform, on which user can efficiently develop machinery systems, from product selection, programming, to exporting.

It simplifies the process, and helps to save time and cost for building machinery systems.

Figure 1 - 1: DIAStudio Characteristics

1.1.1 DIAStudio Overview

The DIAStudio comprises of five key software:

- **DIASelector**: quick, easy, smart selection tool.
- **DIADesigner**: Integrated development & engineering software.
- **DIAScreen**: Intuitive visualization software.
- **DIAInstaller**: System installation & update management.
- **COMMGR**: Communication management.
- **DIADesigner-AX**: Motion development and engineering software.

The DIAStudio supports efficient and flexible data transmission between software. It also facilitates for tag sharing between software.

1.2 DIADesigner Overview

DIADesigner is a member of the DIAStudio integrated software suite, which develops software for Delta's new generation of automation products. Provide user with Delta's full range of Industrial Automation products for program editing, network and hardware layout configuration, parameter settings and other functions, and complete work with a consistent user interface and operating experience. In addition, it also provides many auxiliary tools and multi-language systems, providing user with a convenient and efficient development environment.

Features are as follows:

- Integration Function
    - DIADesigner is an integrated software solution where functions of different legacy software have been integrated into a common platform, such that the user does not need to switch between multiple software to complete different workflow routines. A project can support integration of multiple Delta products in a systematic topological representation.
    - Consistent use interface, friendly guidance, intuitive experience. Provides two design interfaces - **Basic View** and **Advanced View**.
    - Projects compatible with Delta’s legacy software can be migrated into current development environment. (***NOTE*** 1 below).
- Program Editing Function
    - Adopt the international standard IEC 61131-3 programming architecture and built a rich application instruction set.
    - Support for multiple languages, including ladder diagram (LD), sequential function diagram (SFC), structured language (ST) and C language (C).

- Configuration Function
  - Hardware Planning Tool (**Hardware Configuration**): Used to configure and parameterize the hardware architecture on the system.
  - Network Planning Tool (**Network View**): Network architecture and data exchange mechanism for managing and configuring PLC systems.
- Convenient Function
  - Instant check function and keyword filtering function help user to confirm the correctness when inputting commands or parameters.
  - Supports import/export functions of multiple objects, and can be exported/imported into txt, csv and other text formats for further use. (***NOTE*** 2 below).
  - Provides On-line mode and Off-line mode for different operating functions for wired/offline devices.
  - Provides a convenient search/replacement function that can be selected from the current work window or the entire project.
  - Multilingual support including Traditional Chinese, Simplified Chinese, English and others in the subsequent versions.
  - Project management uses a clear and easy-to-use Windows-level tree-like interface.

***NOTE 1**:*

Currently only compatible ISPSoft projects are supported, and subsequent versions will gradually support Delta's other legacy software products.

***NOTE 2**:*

The import/export function can be compatible with the files exported by the legacy software. For example, the parameter file exported by ASDASoft can be imported into the parameter table of DIADesigner.

Chapter 2: Getting Started

2.1 Support Devices

The Delta products supported by DIADesigner v1.0 are as shown in table. Refer device documentation for more information on the devices.

Type	Product Series
Controller	AS200 series AS300 series
AC Servo Drive	ASDA-A2-L/M series ASDA-B2-B/F series ASDA-A3-L/M series ASDA-B3-L/M series
Frequency Converter	C2000 series MS300 series MH300 series VFD-EL series
Temperature Controller	DTC series

2.2 Device and Function Guide

The devices and functions supported by DIADesigner v1.0 are as shown in the following table. Subsequent versions will gradually increase device support and features.

	Communication Setup	Network Configuration	Hardware Configuration	Parameter Setup	Program Edit	Tuning Tool			Download
						Monitoring Table	Trend & Oscilloscope	Tuning	
Controller	CH5	CH7	CH6	CH6.2.1	CH8	CH14.2	CH14.4	N/A	CH13
AC Drive				CH6.2.4	CH8 (Built-in PLC)	N/A			
Servo Drive				CH6.2.3	N/A	N/A		CH14.3	

2.3 System Requirement

The following table provides the specifications for the DIADesigner operating environment:

Item	System Requirement	
Operating System	Windows 7 / 8.1 / 10 Server 2012 R2 32/64 bits	
CPU	Intel Celeron 540 1.8GHz (min.) , Intel Core i5 M520 2.4 GHz (min.)	
Memory	2GB or above (recommend to use 4GB or above)	
Hard Disk Drive	10GB or above	
Monitor	Resolution: 1024 x 768 Pixels, 1920 x 1080 Pixels	
Keyboard/Mouse	General Keyboard Mouse or Windows compatible device	
Printer	Printer with Windows driver (optional for printing of project content)	
USB	Used in Connection with the device	According to the communication interface provided by the device
Ethernet	Used in Connection with the device	
Software	Need to install Microsoft.Net Framework 4.6.2	

2.4 Quick Start and example

2.4.1 Scenario 1

This example is a system that includes a touch-sensitive pedestrian interface (HMI)/programmable controller (PLC)/servo drive (ASDA).The scenario includes HMI/PLC/ASDA

Create a Project

1. Open the software ..... Section 3.2
2. Create a New Project ..... Section 4.2.2

Add Devices

3. Add device in project Explorer ..... Section 4.4.1

Communication Setting

4. COMMGR Configuration ..... Section 5.3
5. Device Communication Settings ..... Section 5.1

Hardware Configuration Settings

6. Module Configuration ..... Section 6.1
7. Parameter Setting ..... Section 6.2

Network Configuration Settings

8. Network View Configuration ..... Section 7.1
9. Data exchange table ..... Section 7.2

Program Editing

10. Establish a POU ..... Section 8.2.1
11. Establish Variables ..... Section 8.2.2

Motion Control

12. Positioning Plan ..... Section 10.1

HMI Variable Sharing

13. Export the Variable to DIAScreen..... Chapter 11
14. Synchronize with DIAScreen Variables ..... Chapter 11

Download

15. Batch Upload/Download..... Section 13.1

Tuning (Commissioning)

16. Variables and register monitoring table..... Section 14.2
17. Debug Mode..... Section 8.7
18. Online Change ..... Section 8.6

2.4.2 Scenario 2

The single inverter or Servo Drive scenario

Create a Project

1. Inverter Wizard /Servo Wizard .....Section 4.2.7.3 ~ Section 4.2.7.4

Add Devices

2. Select Device ..... Section 4.2.7.3 ~ Section 4.2.7.4

Communication Setting

3. Communication Setting .....Section 4.2.7.3 ~ Section 4.2.7.4

Hardware Configuration Settings

4. Parameter Setting..... Section 6.2

Tuning (Commissioning)

5. Oscilloscope..... Section 14.4.3 ~ Section 14.4.4
6. Gain and Inertia Estimation (Applicable to servo drive).....Section 14.3

Chapter 3: Installation and Update

3.1 Installation and Uninstallation

3.1.1 Installing DIADesigner or COMMGR

Ensure that the target computer follows the minimum criteria mentioned in *2.3 System Requirement*, and the DIAInstaller is operating in it.

DIAInstaller is a resident program to manage all Delta Industrial Automation software. User can check for download, install and update Delta Industrial Automation software in DIAInstaller without any burden, and everything will be handled in background. DIAInstaller can be download from below website: https://diastudio.deltaww.com

Please refer Software Download manual for detailed information.

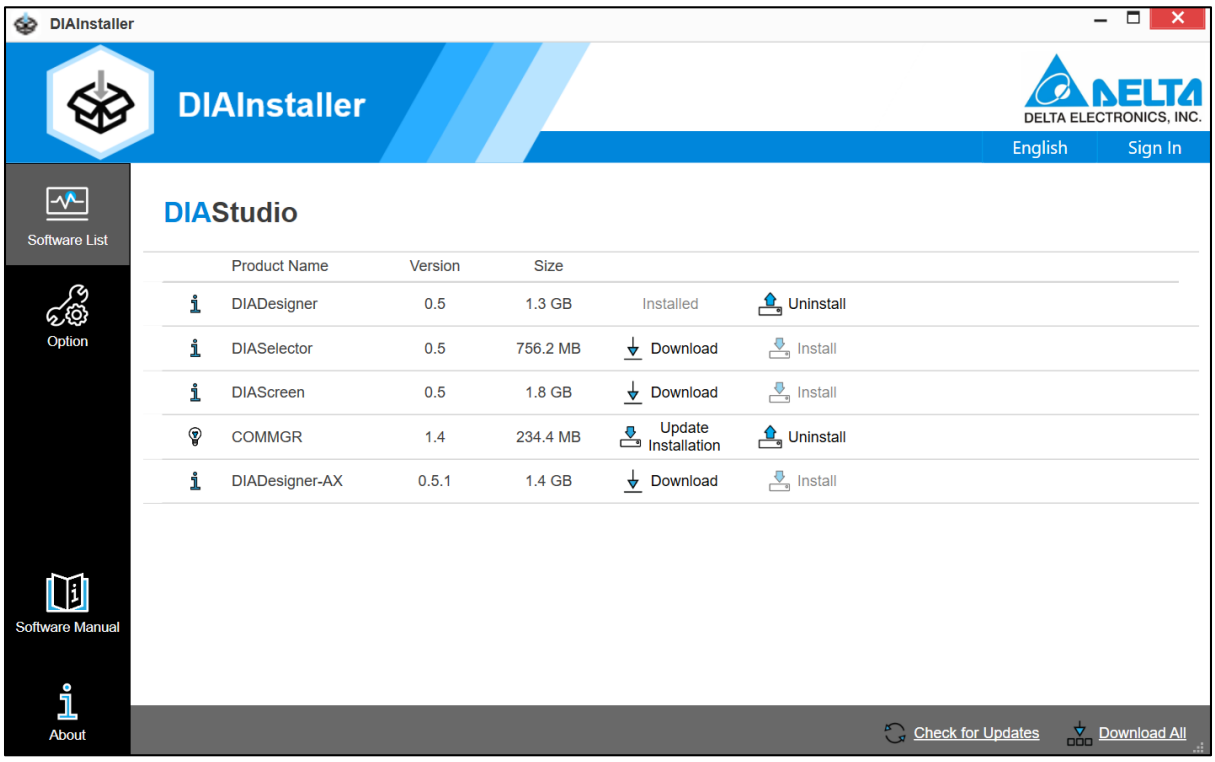


Figure 3 - 1: DIAInstaller - Download

3.2 Launch DIADesigner

After installing DIADesigner as mentioned in 3.1.1 Installing DIADesigner or COMMGR, double-click on the DIADesigner shortcut to launch DIADesigner.

Figure 3 - 2: DIADesigner shortcut

DIADesigner will display the startup screen as shown in the following figure.

Figure 3 - 3: DIADesigner startup screen

Chapter 4: User Interface

4.1 UI Introduction

The initial User Interface of DIADesigner is shown in the following figure:

Figure 4 - 1: Basic Designer Start Page

The Basic Designer UI opens by default. To switch to Advanced Designer, click on **ADV.** at the left-side bottom. User can set the Advanced Designer UI as default option during opening of the project by selecting **Tool** > **Option** > **Environment Settings** > **General Settings** > **Official Website Default Mode**.

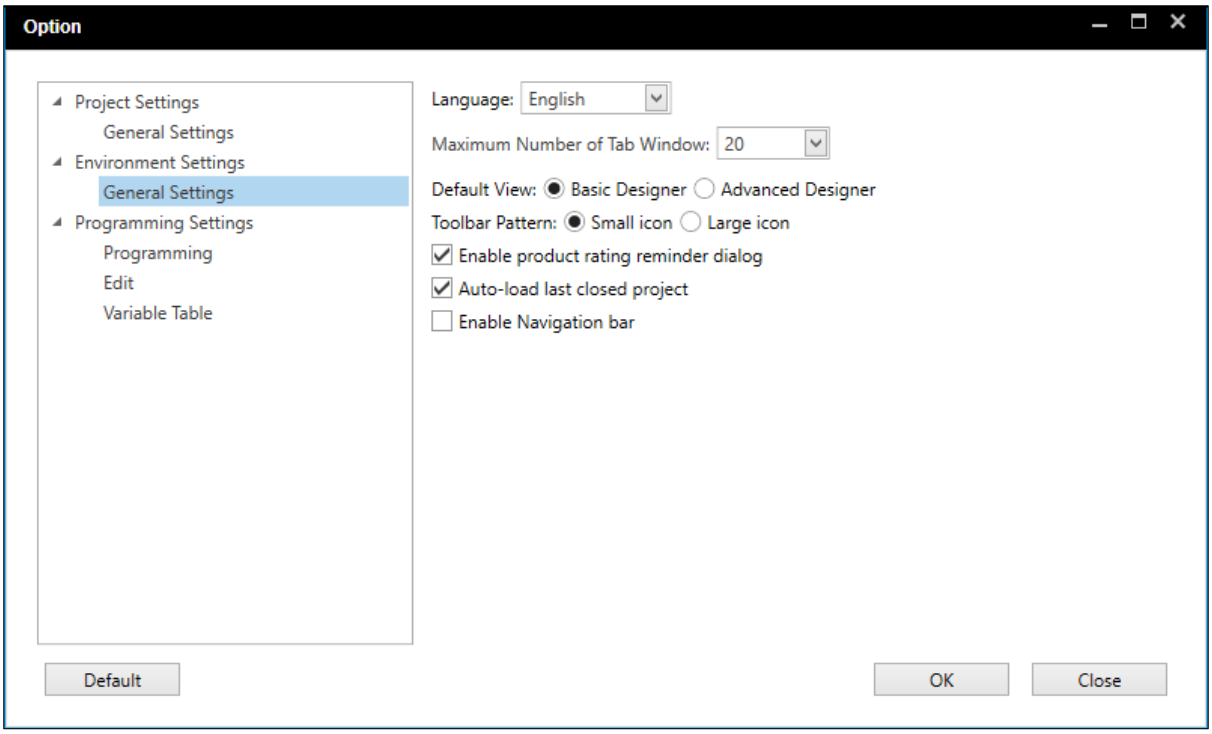


Figure 4 - 2: Option - default mode

4.2 Basic Designer

The Basic Designer provides basic settings and wizards to guide the beginner to do the basic setup. Basic Designer allows user to perform the following functions:

- Provides a simple interface for beginners to get started.
- Sequence the user to complete the basic settings.
- Provide the necessary basic information to understand the configuration and software status.
- Provides the shortcut on the Basic Designer to smoothly switch to Advanced Designer in order to perform the complete configuration.
- Provide common function items to help user smoothly perform the desired functional operations.
- Provide commonly used functions to be marked as favorites.
- Provide quick assistance by opening online learning videos or related documents.

- Provide customer feedback to Delta Electronics Inc.
- Provide Servo/Inverter Wizard

4.2.1 Basic Designer Tabs

The Basic Designer tabs are shown in the following figure:

Figure 4 - 3: Basic Designer Tabs

Tab	Description
Project	Allows user to • Open the project • Create a new project

Tab	Description
	• Migrate (Conversion) • Save the project • Close the project
Device	Allows user to • Add devices • Show devices
Program	Allows user to • Add program • Add variable
Debug	Allows user to • Access communication Setting • Error log • Monitor table
Setting	Allows user to • Set communication parameters • Switch language
Guide	Allows user to • View Official Site • Go to document center • Open servo wizard • Open inverter wizard

4.2.2 Project

The **Project** tab in Basic Designer helps user to create a project. The **Open Project** window displays by default when **Project** tab is clicked. The **Open Project** window is shown in the following figure.

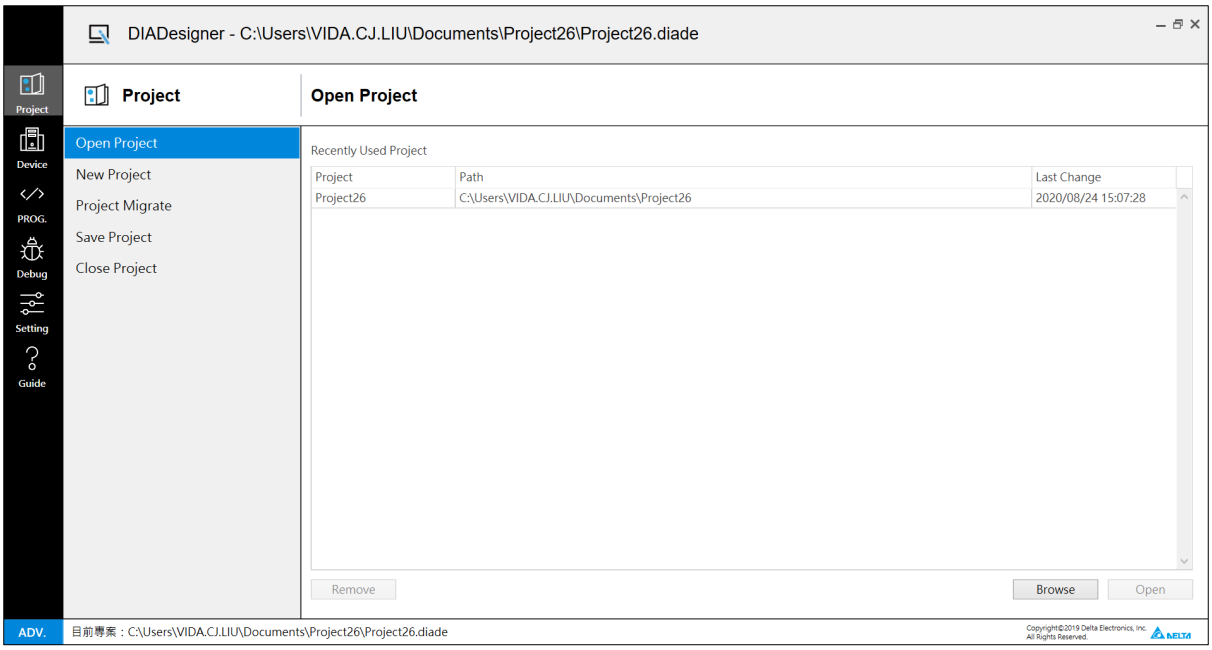


Figure 4 - 4: Open Project

The **Project** tab displays the following functions:

- Open Project
- New Project
- Project Migrate
- Save Project
- Close Project

4.2.2.1 Open Project

User can open previously created project in this window. The functions in the **Open Project** window are described in the following table:

Function	Description
Recently used project	Displays the list of recently opened projects.
Project Types	User can select standard project and library project.

Function	Description
Project Column	Displays the project name.
Path Bar	Displays the project path. ***NOTE***: *User can enter the path manually*.
Last Change Column	Displays the date and time of last change.
Remove	Removes the project from the Recently used project list but the project files are not removed from the computer.
Browse	Opens the window to browse a project.
Open	Opens the project from the Recently used project list.

4.2.2.2 New Project

User can create new project in this window. The **New Project** window is shown in the following figure.

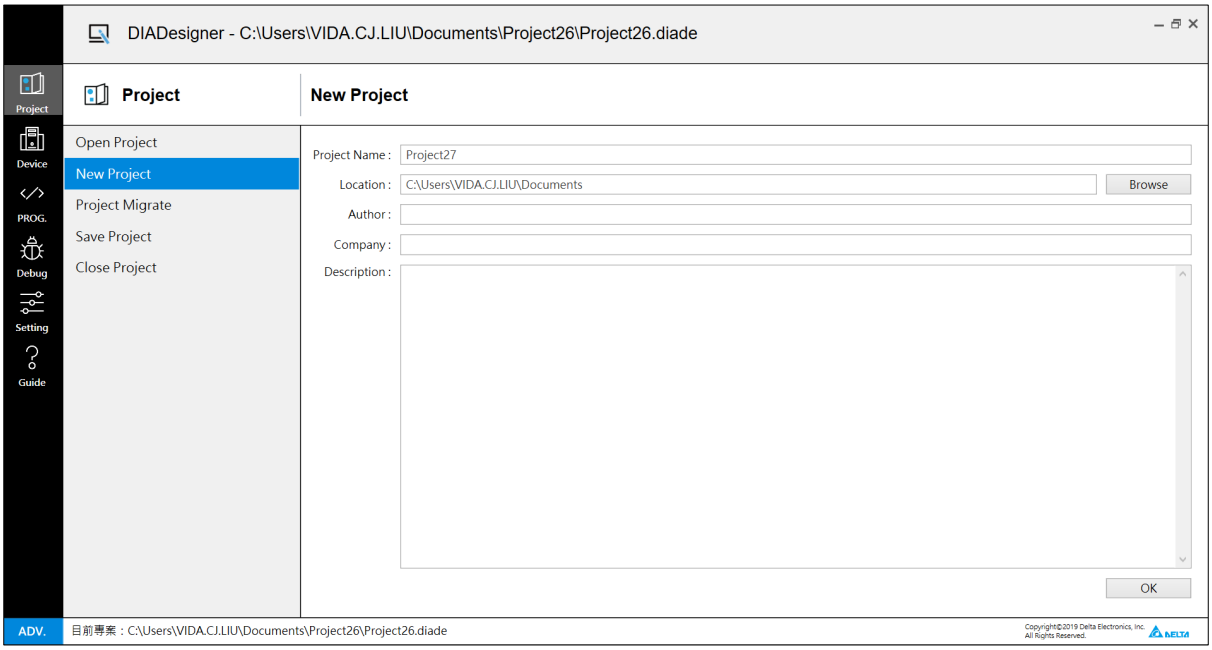


Figure 4 - 5: New Project

The functions in the **New Project** window are described in the following table:

Function	Description
Project Name	Enter a project name.
Location	Displays the project location. User can enter the path. By default, this field displays the location selected from browse function.
Author	Enter Author name
Company	Enter Company name
Browse	Click on **Browse** button to select the path.
Description	Enter project description.
OK	Click to create a new project.

4.2.2.3 Project Migrate

User can migrate the projects created in legacy software like ISPSoft to DIADesigner projects in this window. The **Project Migrate** window is shown in the following figure.

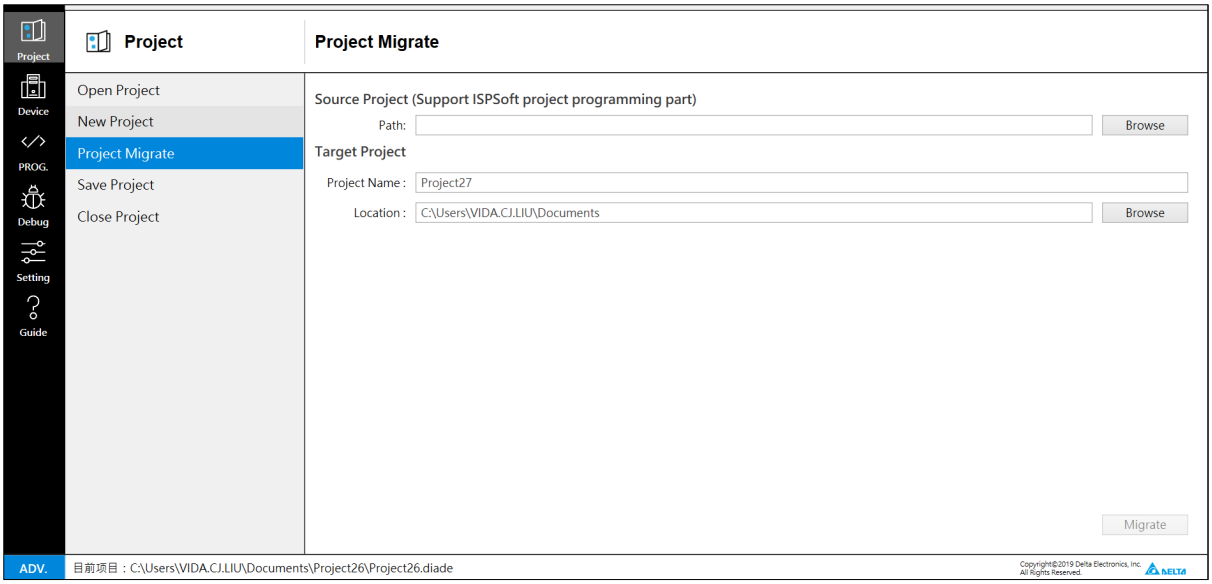


Figure 4 - 6: Project Migrate

The functions in the **Project Migrate** window are described in the following table:

Source Project Function	Description
Path	Displays the path of legacy project made in ISPSoft. User can also enter the file location. By default, this field displays the selection done in **Browse** function.
Browse	Click on **Browse** button to select the legacy project.
Target Project Function	**Description**
Project Name	Enter the DIADesigner project name to be created for the project under migration.
Location	Displays the project location. User can select the path. By default, this field displays the location selected from **Browse** function.
Browse	Click on **Browse** button to select the path.
Migrate	Click on **Migrate** button to migrate the legacy project to DIADesigner project.

4.2.2.4 Save Project

The **Save Project** function allows to save a project opened in DIADesigner.

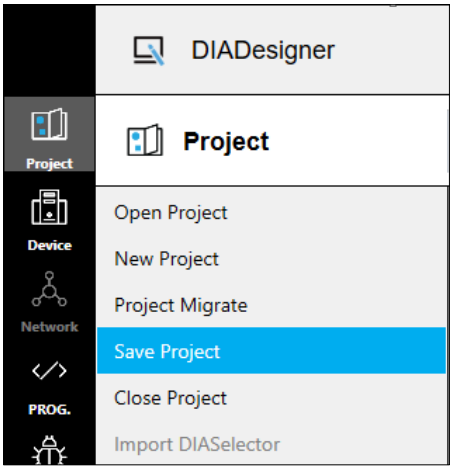


Figure 4 - 7: Save Project

4.2.2.5 Close Project

The **Close Project** function enables a project to be closed.

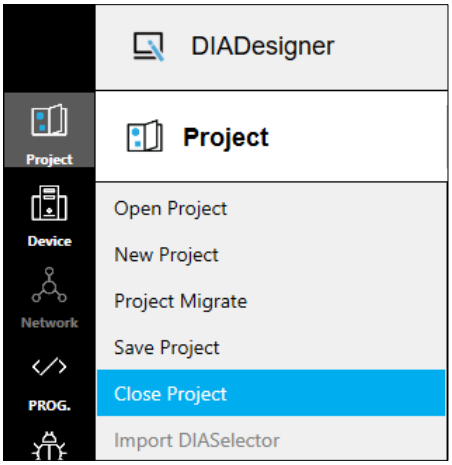


Figure 4 - 8: Close Project

4.2.3 Device

The **Device** tab in Basic Designer helps user to add devices. The **Add Device** window displays by default when **Device** tab is clicked. The **Add Device** window is shown in the following figure.

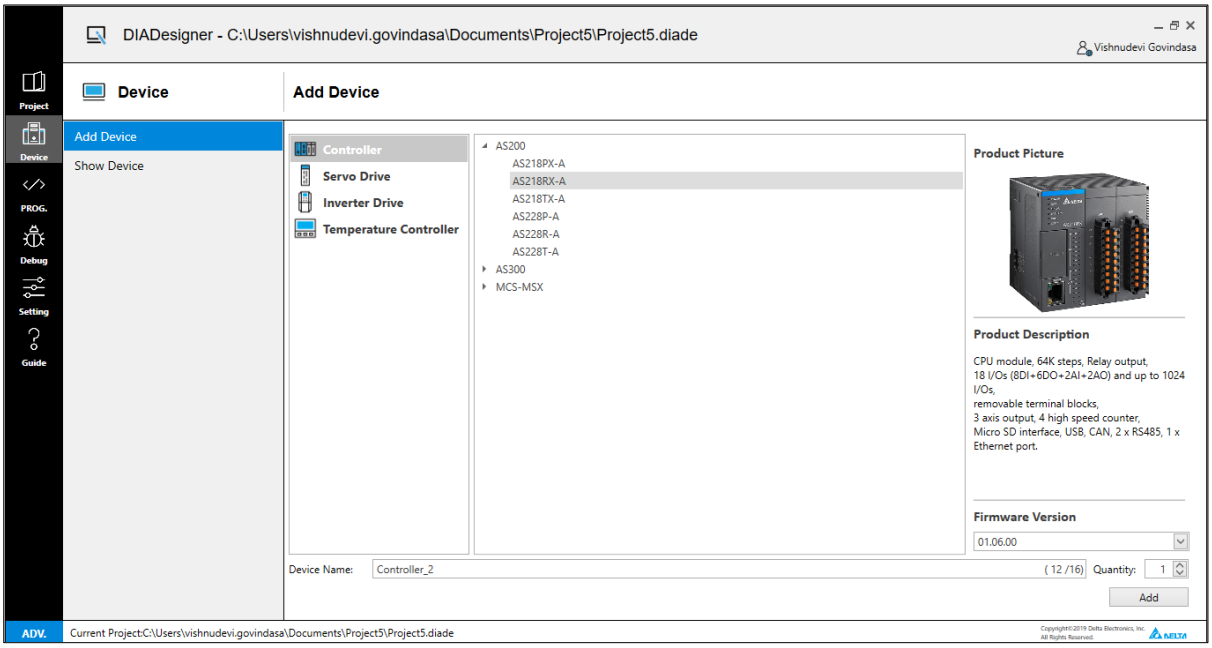


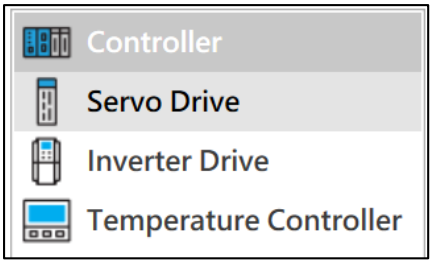
Figure 4 - 9: Add Device

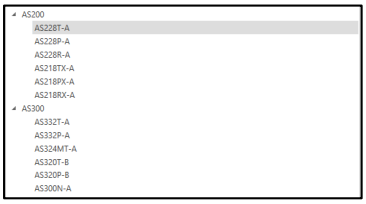
The **Device** tab contains the following options:

- Add Device
- Show Device

4.2.3.1 Add Device

User can add device in this window. The functions in the **Add Device** window are described in the following table:

Function	Description
	**Displays the device types.**

Function	Description
	**Displays the device (product) series and the devices (products).**
Product Specification	Displays the device (product) image and part number.
Product Description	Displays the device (product) description.
Firmware Version	Select the version needed.
Device Name	Enter a name for the device.
Quantity	Select quantity as per requirement.
Add	Click to add devices to the project.

4.2.3.2 Show Device

The **Show Device** window allows user to view the devices created previously in the current project. The **Show Device** window is shown in the following figure.

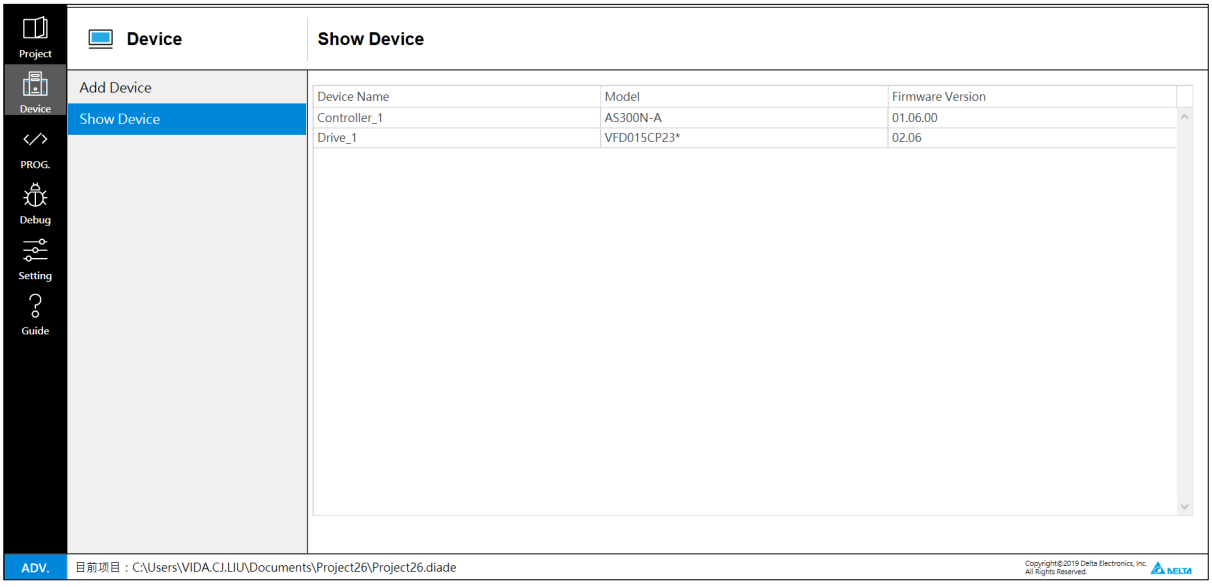


Figure 4 - 10: Show Device

The functions in the **Show Device** window are described in the following table:

Function	Description
Device Name	Displays the device name created by user.
Model	Displays the device model.
Firmware Version	Displays the device version selected by user.

4.2.4 Programming

The **PROG.** tab in Basic Designer helps user to create programs. The **Program** window displays by default when **PROG.** tab is clicked. The **Program** window is shown in the following figure.

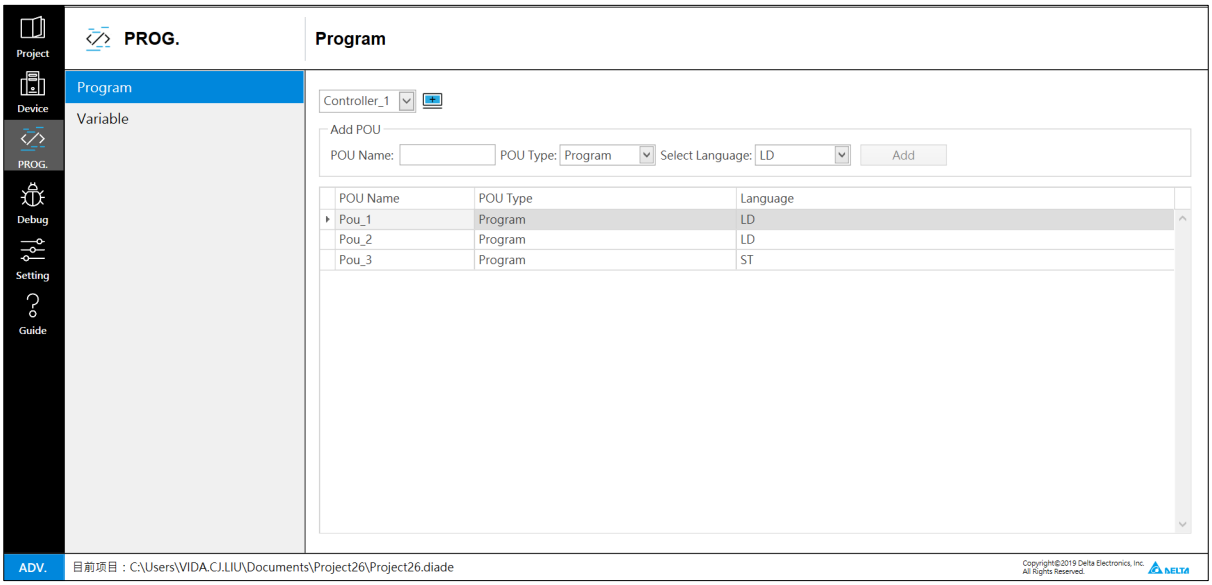


Figure 4 - 11: program

The **PROG.** tab contains the following options:

- Program
- Variable

4.2.4.1 Program

User can create POUs of **Program** or **Function Block** in this window. The functions in the **Program** window are described in the following table:

Function	Description
Controller drop-down list	Select the controller from the drop-down list.
Add POU – POU Name	Enter a POU name.
Add POU – POU Type	Select a POU type. Options are: • Program • Function Block ***NOTE**: The default value is **Program**.*
Add POU – Select Language	Select programming language. Options are: • ST • LD • SFC • C ***NOTE**: The default value is **LD**.*
Add	Click to add POU to the selected controller.
POU Name column	Displays POU names in the selected controller.
POU Type column	Displays POU type.
Language column	Displays POU language.

4.2.4.2 Variable

User can set the variables in this window. The **Variable** window is shown in the following figure.

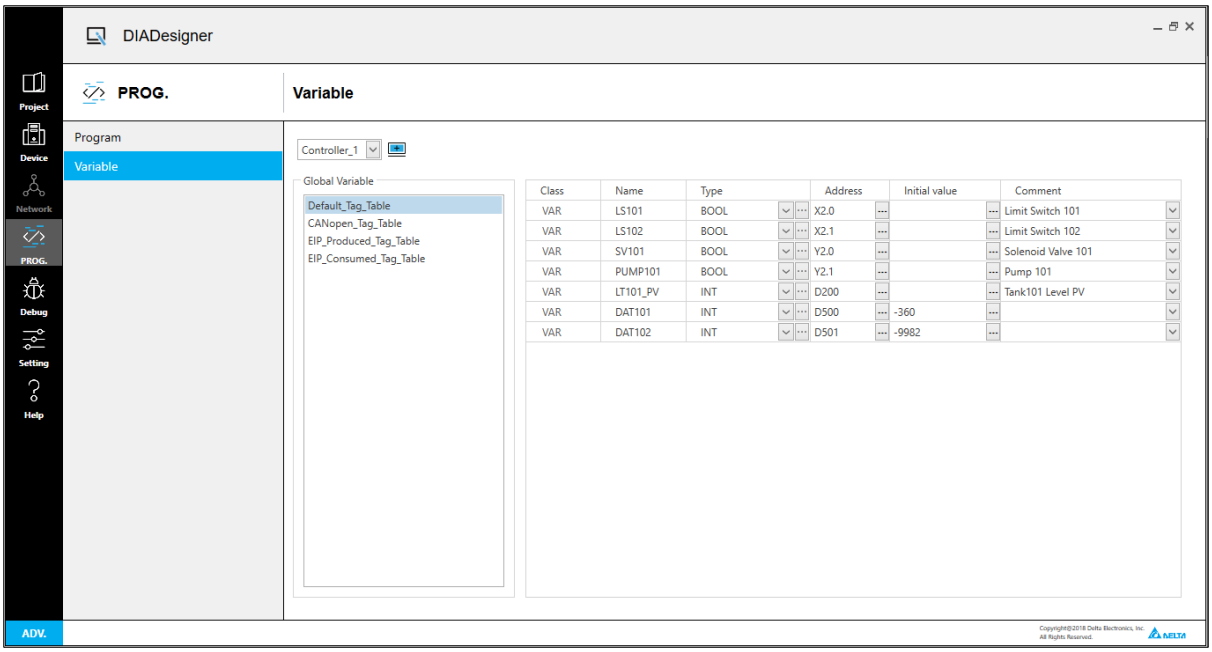


Figure 4 - 12: Variable

The functions in the **Variable** window are described in the following table:

Function	Description
Controller drop-down list	**Select the controller from the drop-down list.**
Global Variable	Select the **Global Variable** type. • Default Tag Table • CANopen Tag Table • EIP Produced Tag Table • EIP Consumed Tag Table • C Tag Table
Class	Select the Variable Class. • VAR • RETAIN ***NOTE**: The default value is **VAR**.*
Name	Enter the Variable name.
Type	Select the variable datatype. • BOOL • WORD • DWORD • INT

Function	Description
	• DINT • REAL • STRING • ARRAY • TIMER • COUNTER ***NOTE**: The default value is **BOOL**. Click on ... to open **Data Type** window to define the variable.*
Address	Enter the variable address. Click on ... to open the **Reference Register** window and select the register and address.
Initial Value	Enter initial value for variable. Click on ... to open the **Reference Initial Value Setting** window to select the initial value.
Comment	Enter variable comment.

4.2.5 Debug

The **Debug** tab in Basic Designer helps user to debug and monitor the program. Click on **Debug** tab to display the **Monitor Table** window as shown in the following figure.

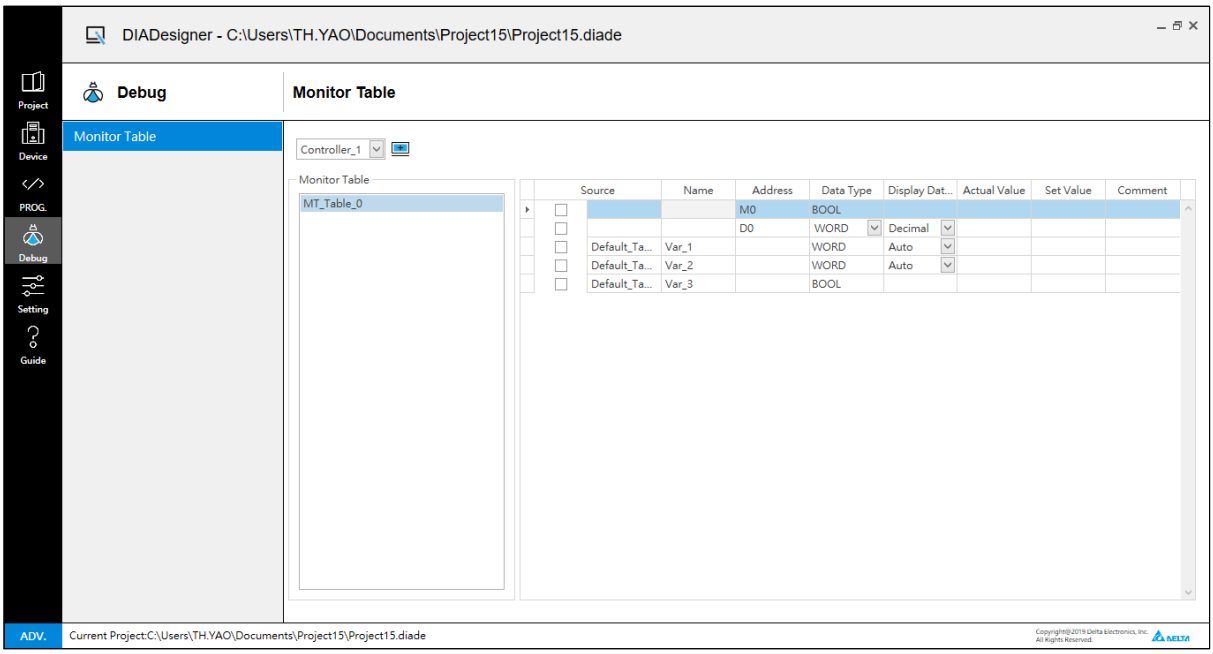


Figure 4 - 13: Debug

The **Debug** tab contains the following options:

- Monitor Table

4.2.5.1 Monitor Table

User can view and manage the monitor tables in this window. The **Monitor Table** window is shown in the following figure.

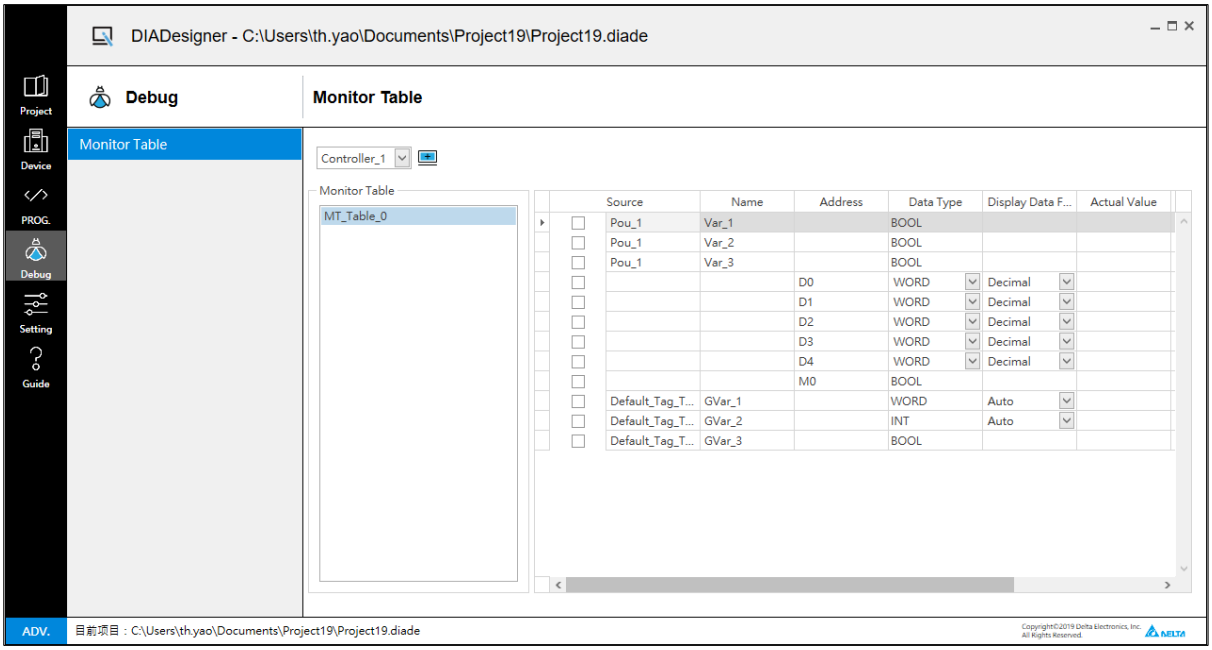


Figure 4 - 14: Monitor Table

The functions in the **Monitor Table** window are described in the following table:

Function	Description
Controller drop-down list	Select the controller from the drop-down list.
Monitor Table	Displays the list of monitor tables created in the controller device.
Check	Click to select the line item.
Source	Displays the source. **Example**: Default Tag Table, POU name etc.

Function	Description
Name	Displays the variable name.
Address	Displays the variable address.
Datatype	Displays the variable data type.
Display Data Format	Displays the data format. Options are: • Decimal • Unsigned Decimal • Binary • Hexadecimal • ASCII ***NOTE**: The default value for non-Boolean variables is **Auto**.*
Actual Value	Displays the value in variable.
Set Value	Enter variable value to write to controller.
Comment	Displays variable comment.

4.2.6 Setting

The **Setting** tab helps user to perform basic settings in Basic Designer. The **Device Repository** window displays by default when the **Setting** tab is clicked. The **Setting** window is shown in the following figure.

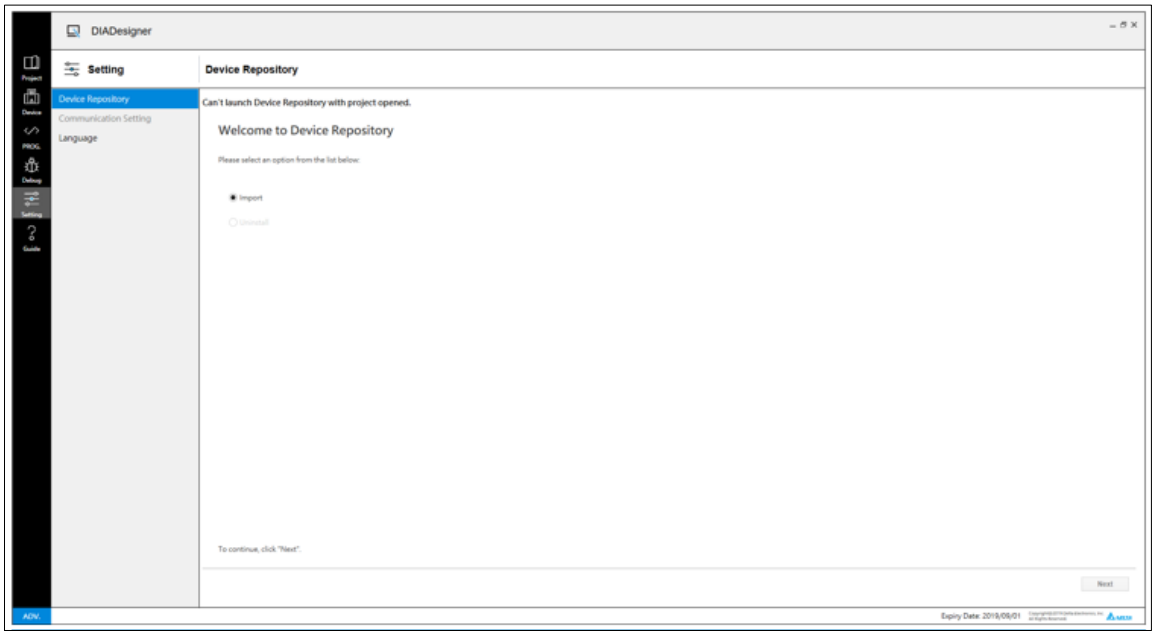


Figure 4 - 15: Setting

The **Setting** tab contains the following options:

- Device Repository
- Language

4.2.6.1 Device Repository

User can perform the communication settings in this window. **Device Repository** is a management tool for the device description files. A device can be used in DIADesigner only when its device description file is installed. The functions in the **Device Repository** window is described in the following table.

Function	Description
Import	**To import device description file of devices, including EDS and so on.**
Uninstall	To uninstall the installed device description file of devices, including DDF, EDS, and so on.

4.2.6.2 Language

User can switch application language using this function. The **Language** window is shown in the following figure.

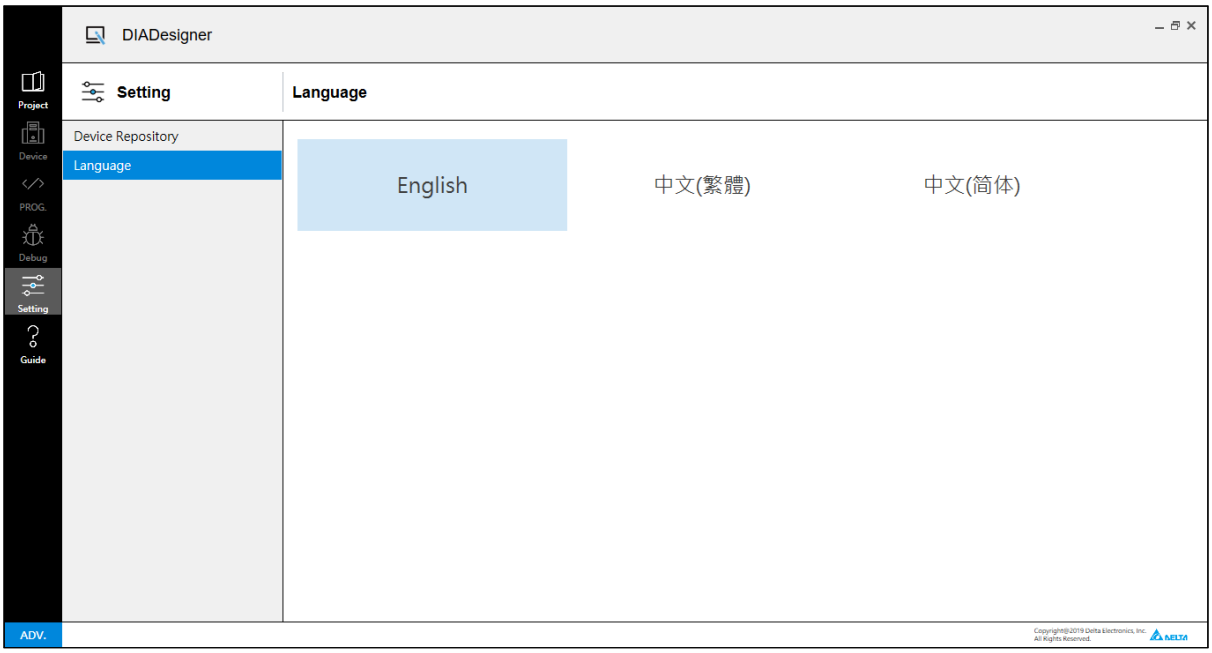


Figure 4 - 16: Choose a Language

The options in the **Language** window are described in the following table:

Function	Description
English	Click to switch to English language environment.
中文(繁體)	Click to switch to Traditional Chinese language environment.
中文(简体)	Click to switch to Simplified Chinese language environment.

4.2.7 Guide

The **Guide** tab in Basic Designer helps user to get around help, documentation and wizards.

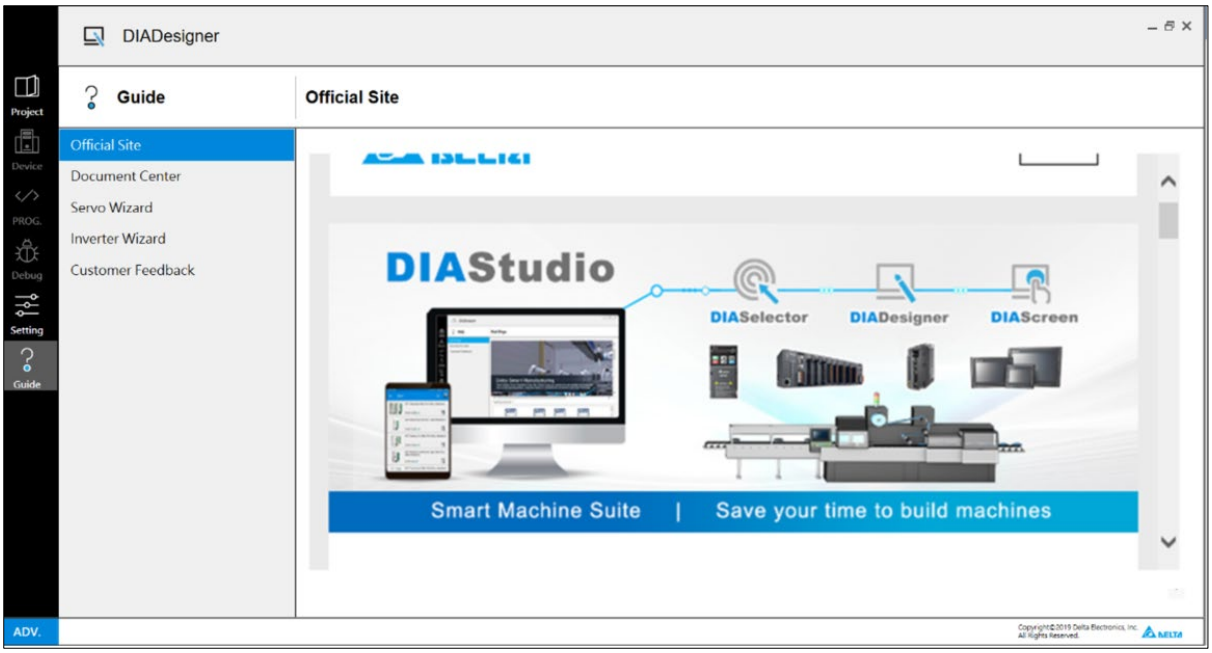


Figure 4 - 17: Guide

The **Guide** tab contains the following options:

- Official Site
- Document Center
- Servo Wizard
- Inverter Wizard
- Customer Feedback

4.2.7.1 Official site

Official site displays the DIAStudio web page and the teaching resources.

4.2.7.2 Document Center

User can access project documents in this window. The **Document Center** window is shown in the following figure.

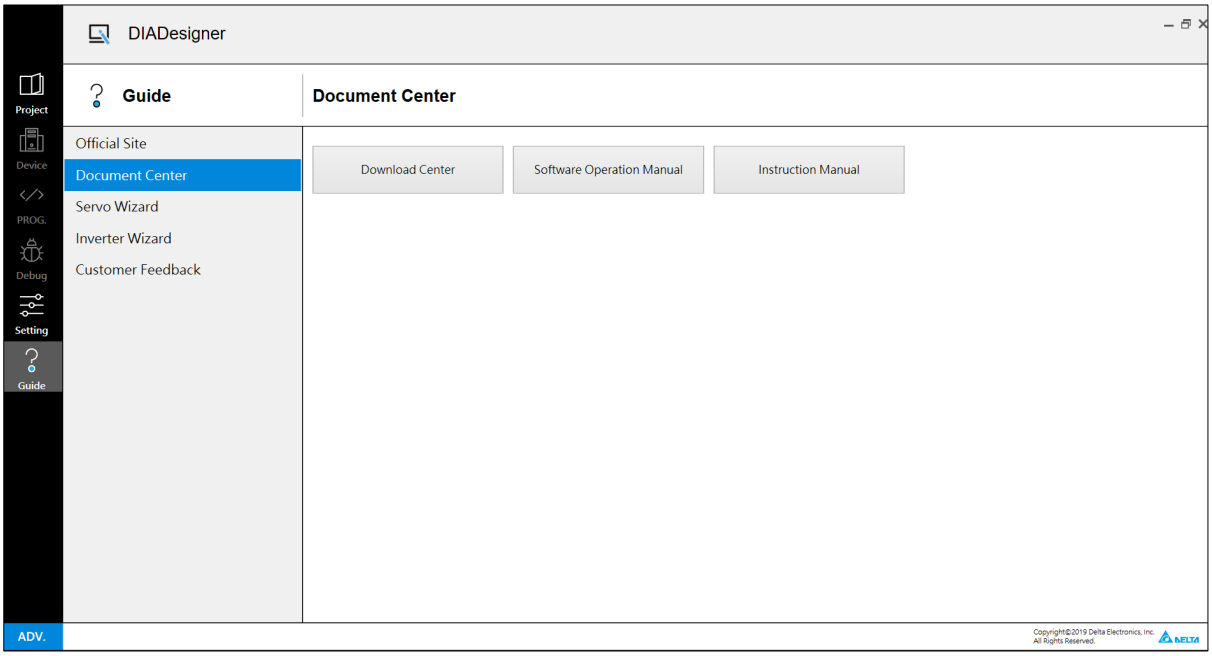


Figure 4 - 18: Document Center

The functions in the **Document Center** window are described in the following table:

Function	Description
Download Center	Click to open **DIAStudio web home page** in the default web browser.
Software Operation Manual	Click to open Software Operation Manual.
Instruction Manual	Click to open Instruction Manual.

4.2.7.3 Servo Wizard

User can add Servo Drive devices into projects in this window. **The Servo Wizard** window is shown in the following figure.

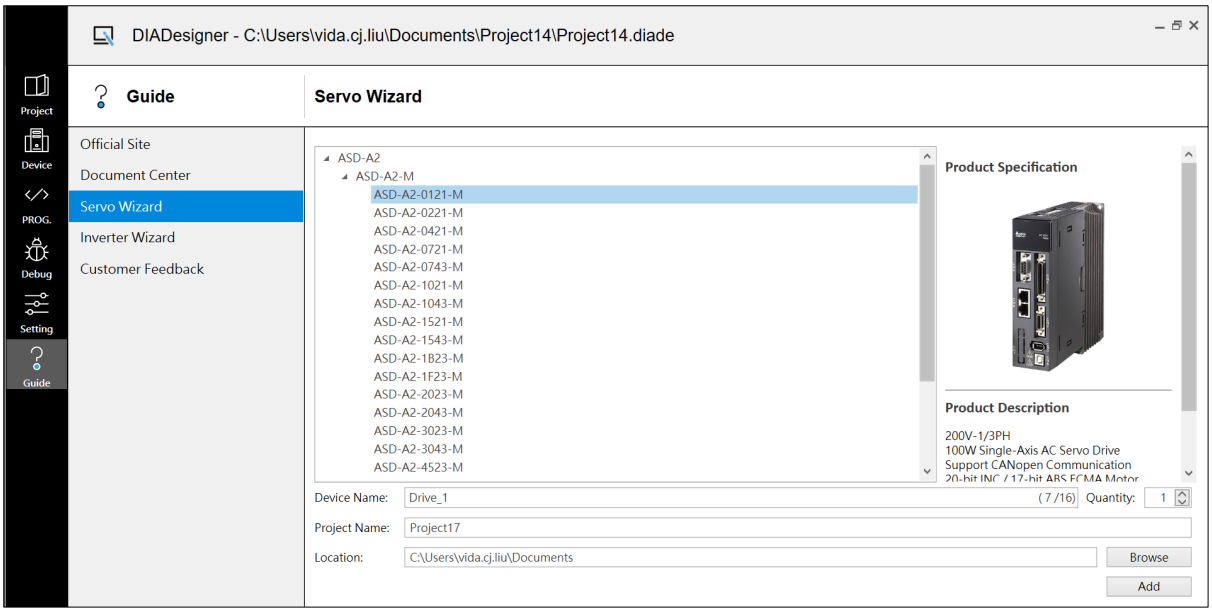
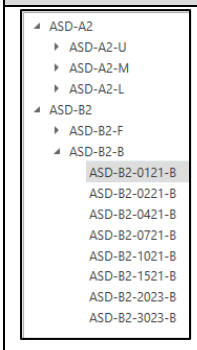


Figure 4 - 19: Servo Wizard

The functions in the **Servo Wizard** window are described in the following table:

Function	Description
	Displays the Servo Drive series and the part numbers.
Product Specification	Displays the Servo Drive image and part number.
Product Description	Displays the Servo Drive description.
Firmware Version	Select the version needed.
Device Name	Enter a name for the Servo Drive.
Quantity	Select quantity as per requirement.
Project Name	Enter a project name.
Location	Displays the project location. User can type the path. By default, this field displays the location selected from Browse function.
Browse	Click to select the path.

Function	Description
Add	Click to add Servo Drive to the project.

After a servo drive model is selected from **Servo Wizard**, a servo device project is created in **Project Explorer** with **Communication Setting** window opened. User can connect to the servo device directly and start the servo setting as shown in the following figure.

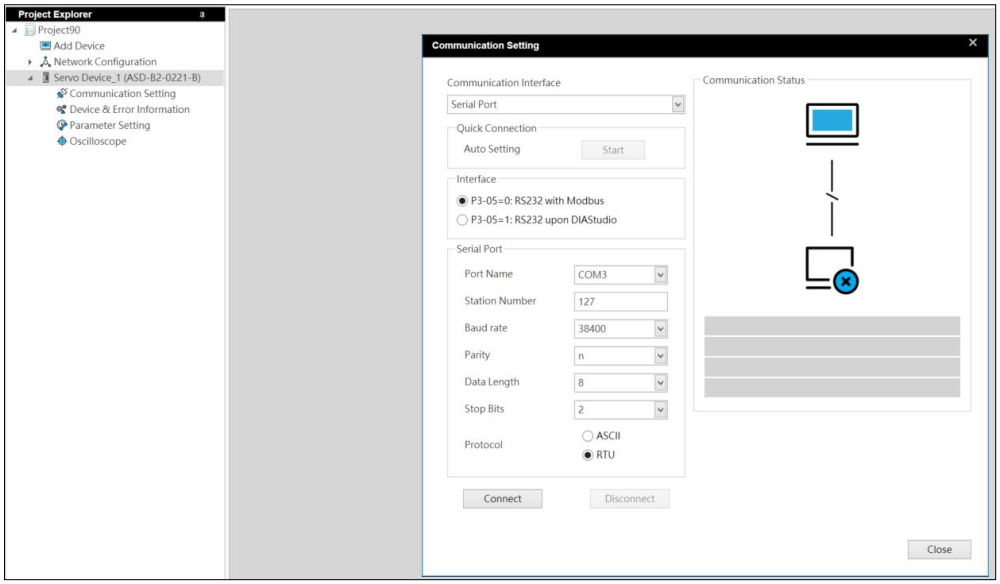


Figure 4 - 20: Communication Setting

4.2.7.4 Inverter Wizard

User can add inverter devices into projects in this window and establish connection with the inverter devices. The **Inverter Wizard** window is as shown in the following figure.

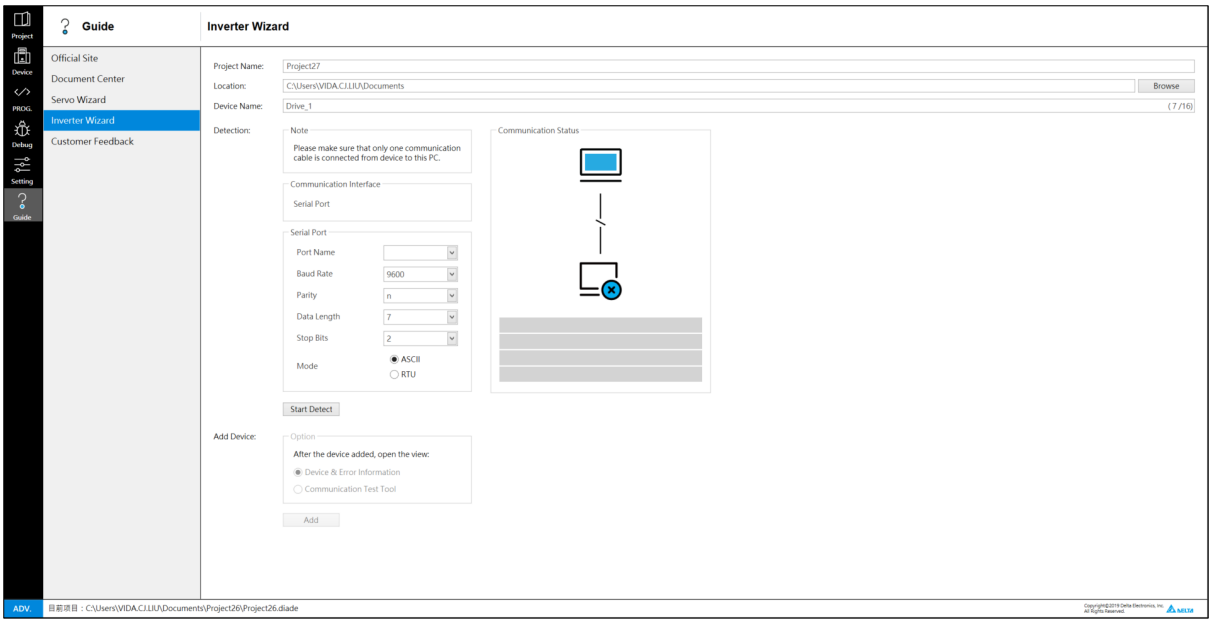


Figure 4 - 21: Inverter Wizard

The functions of **Inverter Wizard** are described in the following table:

Function	Description
Project Name	Enter a name for the project
Location	Displays the project location. User can enter the path. By default, this field displays the location that was previously selected.
Device Name	Enter a name for the inverter
Detection	User can set the communication of inverter. Click **Start Detect** Start Detect to auto-detect the inverter. For stopping detection, click **Stop Detect.**
Add Device	After clicking **Start Detect**, device will be detected, and then this function is enabled. User can select a radio button of **Device & Error Information** or **Communication Test Tool** in **Option**. After that, click **Add** to create a project with the device.

Click **Add** in **Inverter Wizard**, the window switches to the **Advanced Designer** view, and auto-connects to the device. Then, device information and communication test tools display. For the related information of **Advanced Designer**, please refer to *4.3 Advanced Designer*.

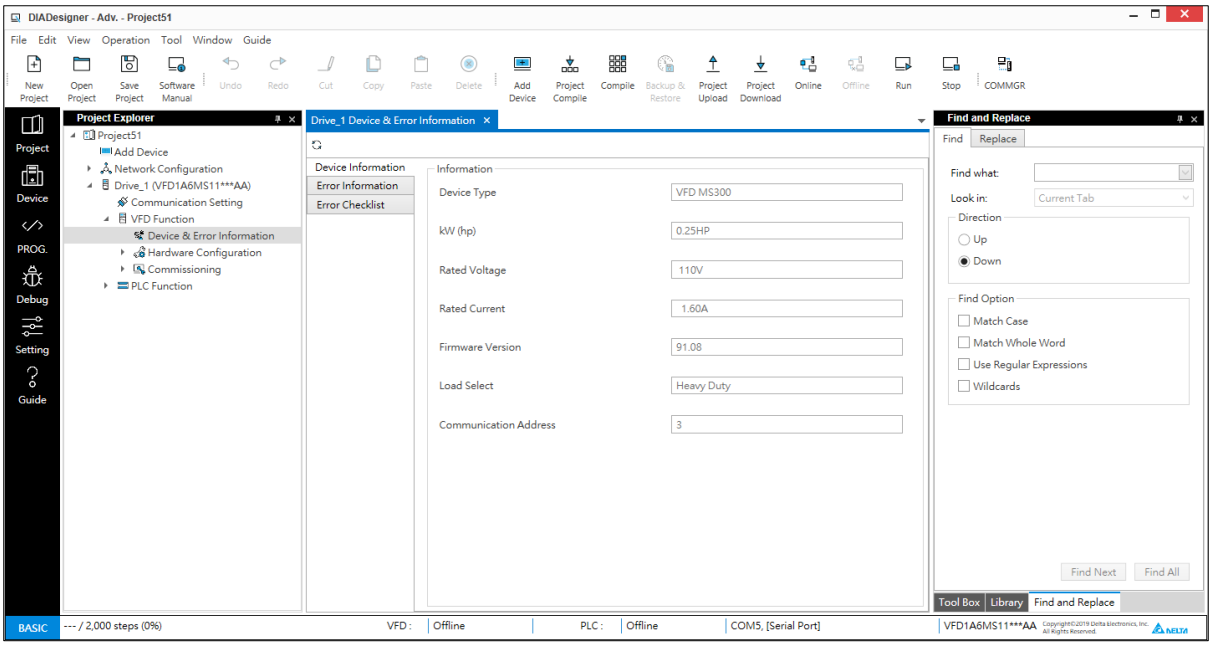


Figure 4 - 22: Device Information

4.2.7.5 Customer Feedback

User can send their feedback of opinions and software issues in this window. The **Customer Feedback** window is shown in the following figure.

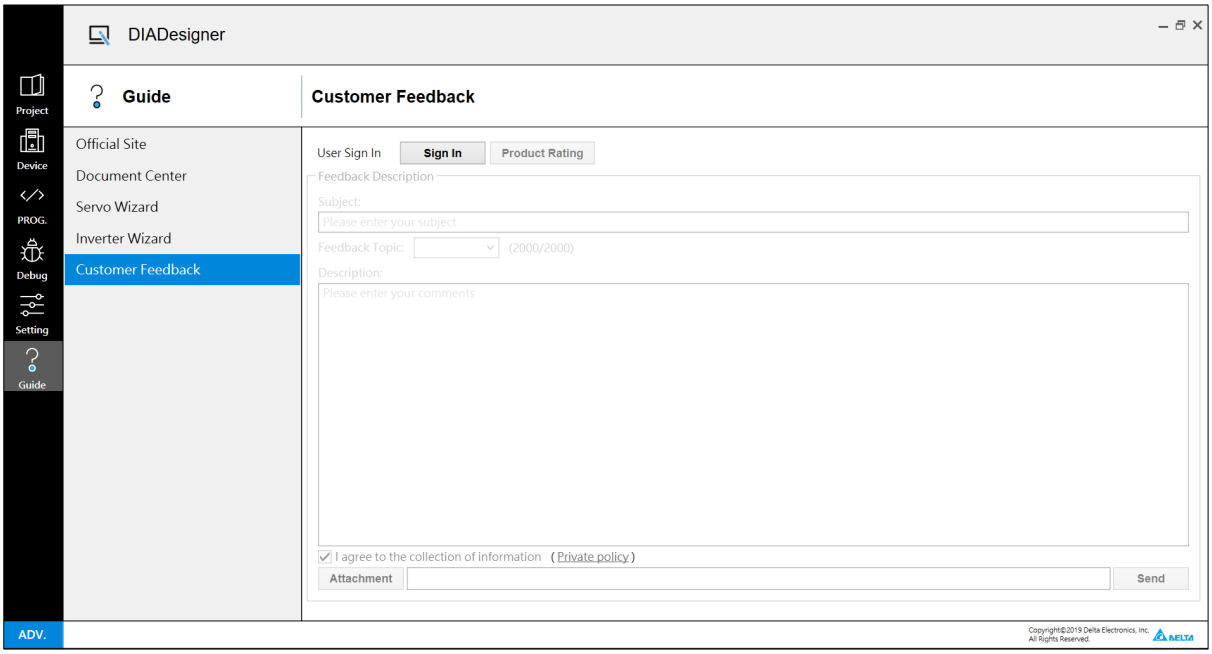
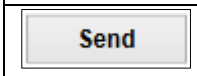


Figure 4 - 23: Customer Feedback

The functions in the **Customer Feedback** window are described in the following table:

Function	Description
	Click **Sign In** button launches the **Sign In/Sign Up** window. After Signing in, the functions in the **Customer Feedback** window are enabled.
	Open **Rating** dialog to send rating for DIADesigner.
Feedback Description	The area of entering customer opinion.
Subject	Feedback subject
Feedback Topic	Select a related topic of user feedback subject
Description	Feedback content
I agree to the collection of information (Private policy)	Click the box to approve sending the information of user PC to Delta. Click **Private policy** to check the detail information.
	Select attachment files to send to Delta for analysis. User can select most 3 files at a time, 2 MB max. for the files.
	Click to send the feedback to Delta.

Follow these steps to send customer feedback:

1. Click **Sign In**, and enter user account and password to sign in Delta account.

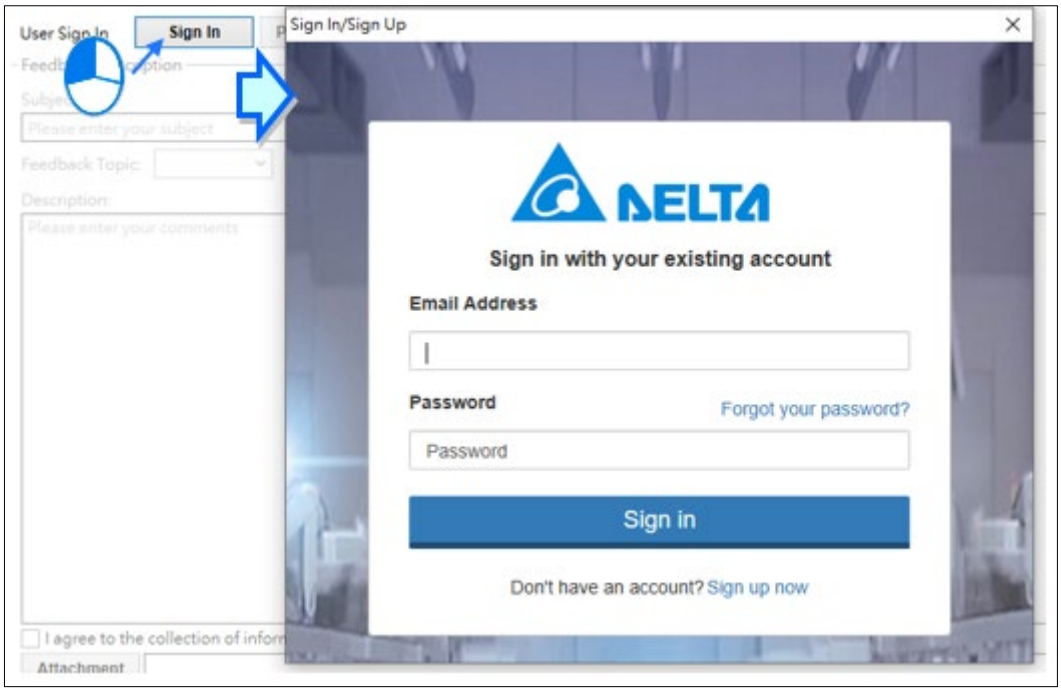


Figure 4 - 24: Sign – In window

2. Enter the content of user feedback and attach the related attachment. Click the check box and click **Send** to send the feedback.

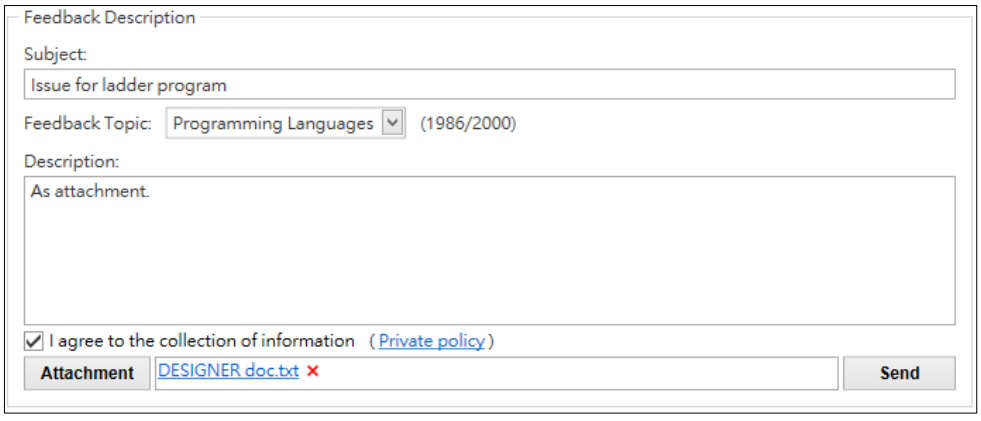


Figure 4 - 25: Feedback Window

4.3 Advanced Designer

The Advanced Designer provides a complete integrated development environment to perform the configuration/programming/networking of devices. User can switch

between Basic Designer and Advanced Designer by clicking on the respective options in the left-side bottom icon.

In the default state, when switching to the ADV page, the navigation bar of BASIC is closed. The user can choose whether to enable the navigation bar when entering the ADV page through Tools> Options> Environment Settings> General Settings, as shown in the figure 4-26.

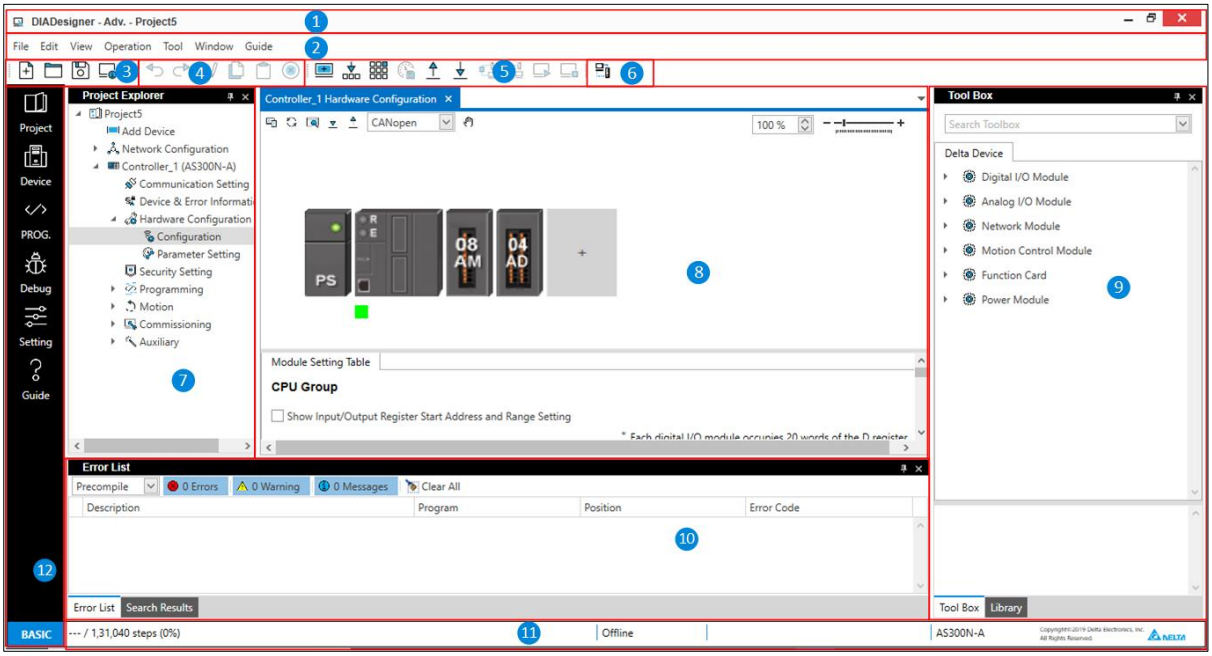


Figure 4 - 26: Advanced Designer

1. Windows title bar
2. Menu bar
3. Standard Toolbar
4. Editor Toolbar
5. Development Toolbar
6. Communication Toolbar
7. Project Explorer
8. Work area

⑨ Auxiliary functions area

⑩ Results area

⑪ Status bar

⑫ Basic navigation bar

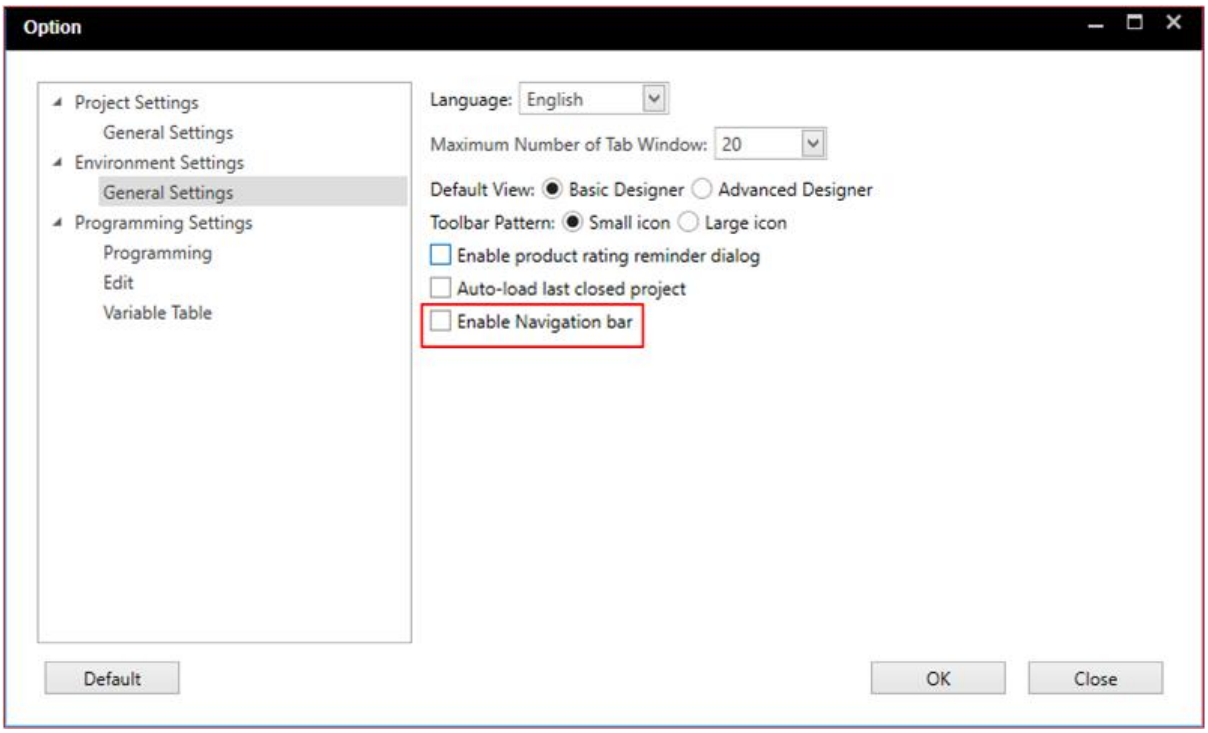


Figure 4 - 27: Option – Enable Navigation bar

When switching to the ADV page without changing any settings, the default value of the toolbar icon is a small icon, as shown in Figure 4-27, the user can go to Tools> Options> Environment Settings> General Settings> Toolbar Style Select to change the toolbar icon as a large icon, please refer to Figure 4-28, the icon after the change is shown in Figure 4-29.

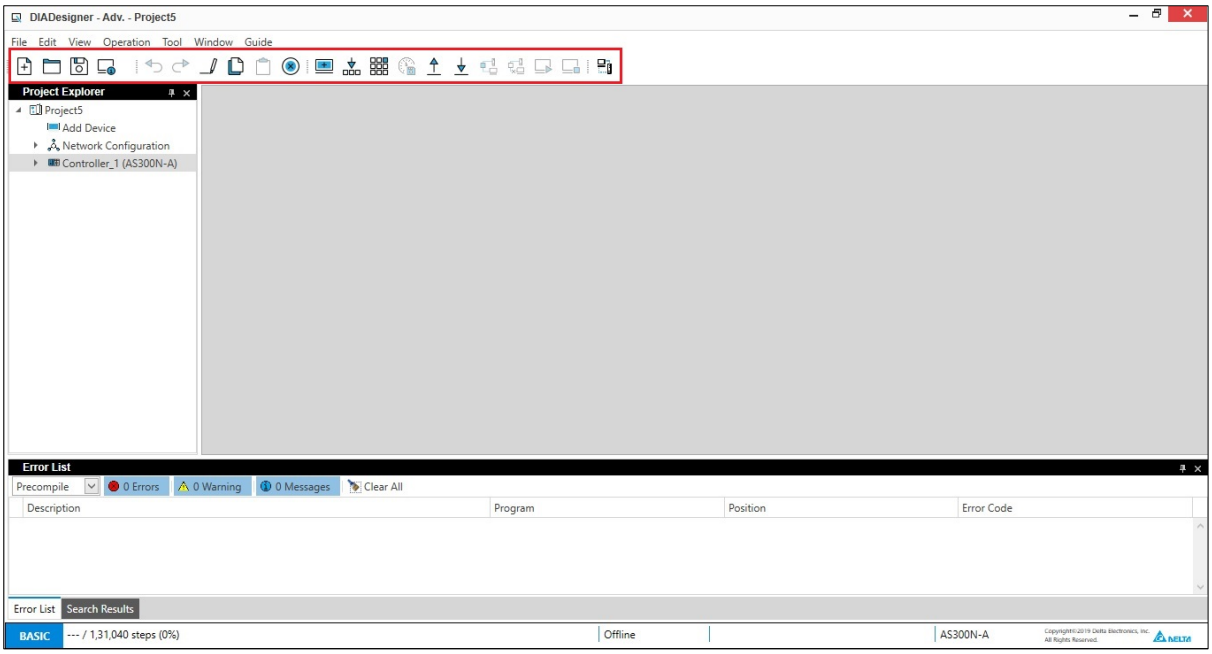


Figure 4 - 28: Default icon – small icon

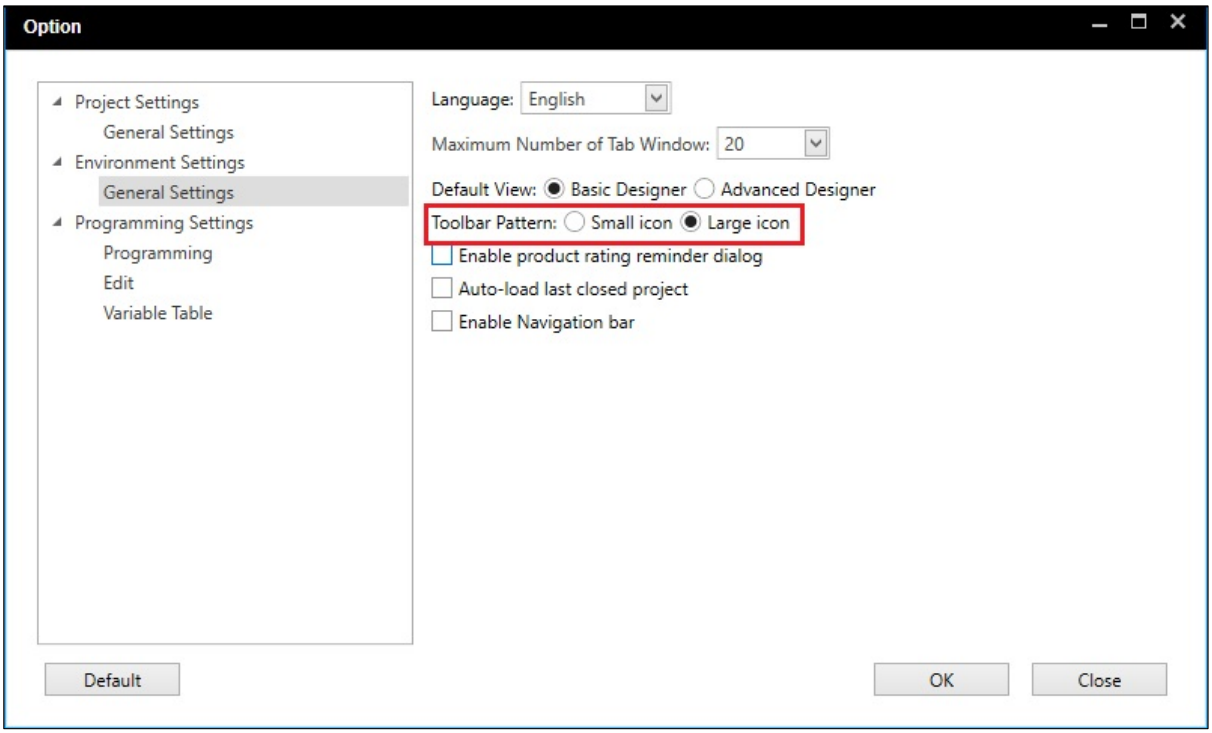


Figure 4 - 29: Tool Pattern changed to Large icon

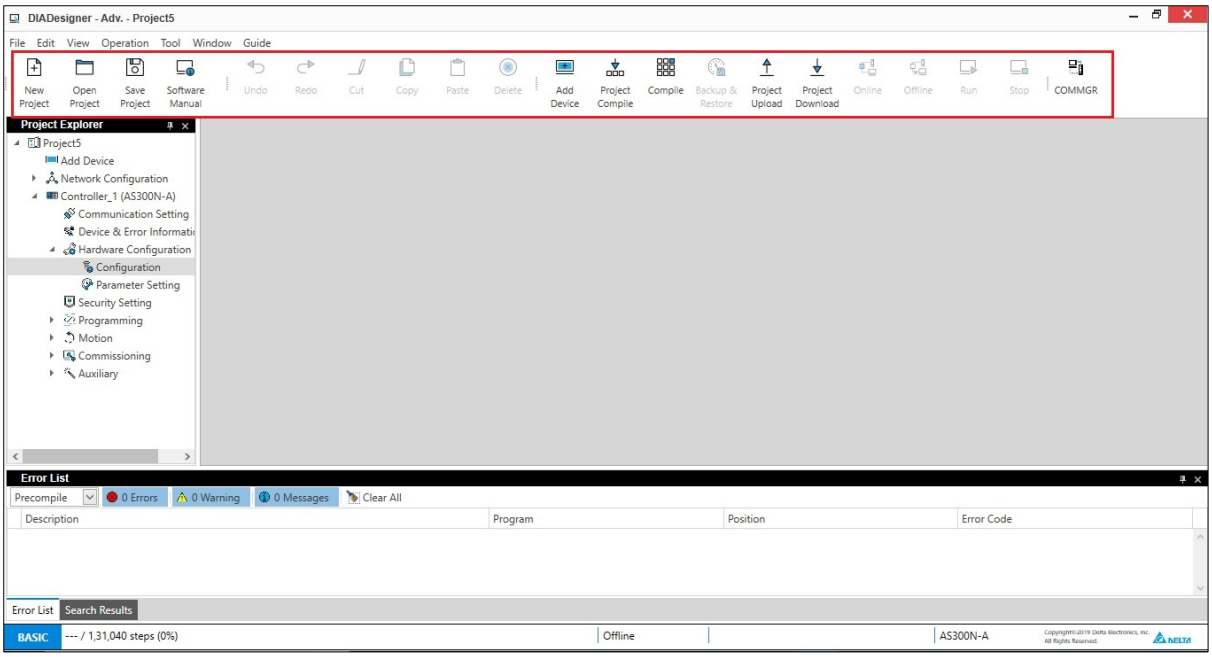


Figure 4 - 30: Tool bar - Large icon

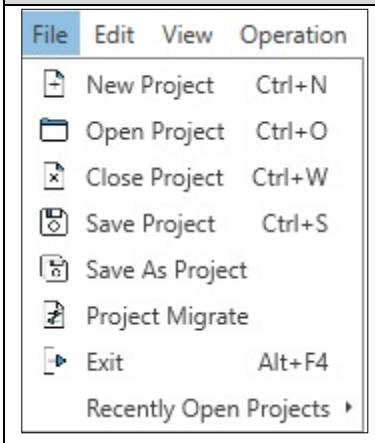
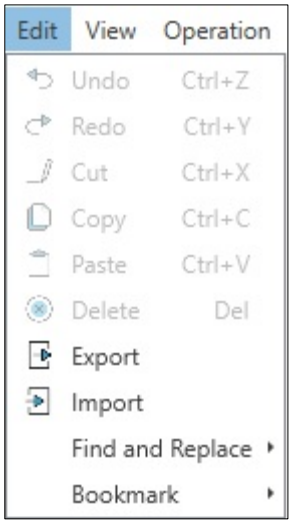
4.3.1 Menu bar

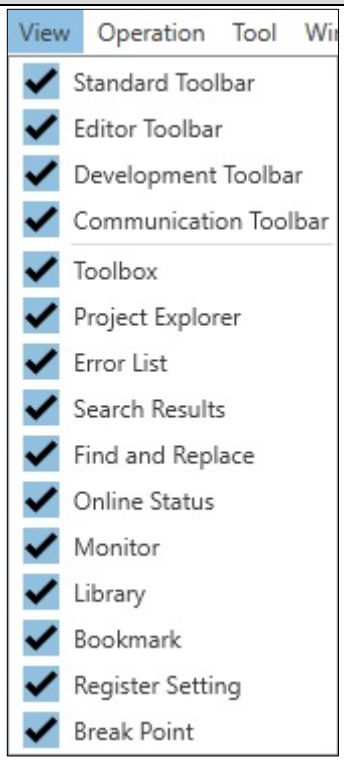
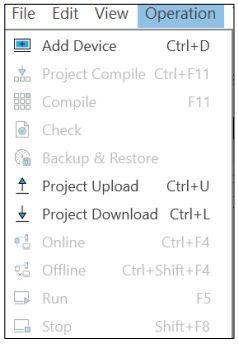
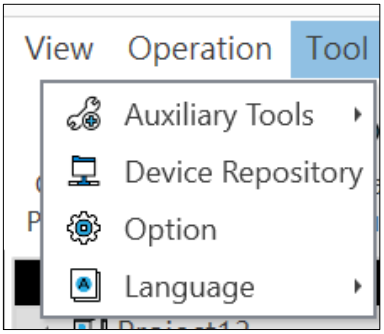
The Advanced Designer **Menu** bar is as shown in the following figure.

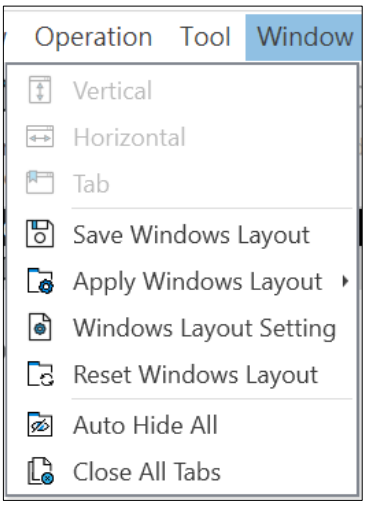
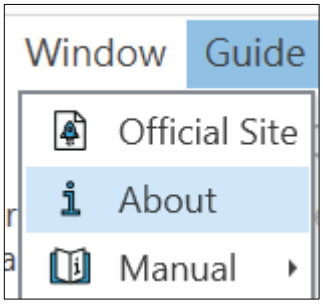
Figure 4 - 31: Menu bar

The functions in the **Menu** bar are described in the following table:

Function	Description
File	Allow user to • Create a new project • Open a created project • Close the current project • Save the current project • Save this project in another name/location

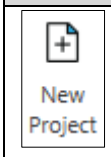
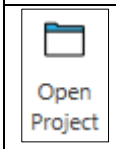
Function	Description
	• Convert legacy software projects into DIADesigner format • Exit the current project • Access the recently opened projects
Edit 	Allows user to • Undo • Redo • Cut • Copy • Paste • Delete • Open the export function • Open the import function • Open the Find and Replace function • Bookmark and the related functions
View	Allows user to hide/display • Show/Hide Standard Toolbar • Show/Hide Editor Toolbar • Show/Hide Development Toolbar • Show/Hide Communication Toolbar • Show/Hide ToolBox • Show/Hide Project Explorer • Show/Hide Error List • Show/Hide Output • Show/Hide Search Results • Show/Hide Find and Replace • Show/Hide Online Status • Show/Hide Monitor • Show/Hide Library

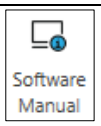
Function	Description
	• Show/Hide Bookmark • Show/Hide Register Setting • Show/Hide Break Point
Operation 	Allows user to • Add Device • Compile all device contents in the current project • Compile the selected one device • Check the selected one device • Open the Backup & Restore function • Open the Project Upload function • Open the Project Download function • Switch to Online mode • Switch to Offline mode • Switch to Run mode • Switch to Stop mode
Tool 	Allows user to • Open Auxiliary Tools • Open Device Repository • Open Option • Change Language

Function	Description
Window 	Allows user to • Tile Windows Vertically • Tile Windows Horizontally • Tile Windows as Tabs • Save the current Windows Layout • Apply the Saved Window Layout • Open Windows Layout Setting • Reset Windows Layout • Open Auto Hide all Windows function • Close All Windows
Guide 	Allows user to • Open Official Site • Update Device Database • Open **About** Window • Open Manuals

4.3.2 Standard Toolbar

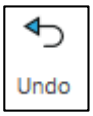
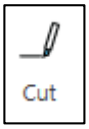
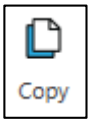
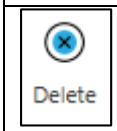
The functions in the **Standard** toolbar are described in the following table:

Function	Description
	Click to add new project
	Click to open project

Function	Description
	Click to save the project
	Click to open software manual

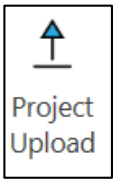
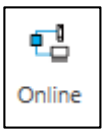
4.3.3 Editor Toolbar

The functions in the **Editor** Toolbar are described in the following table:

Function	Description
	Click to undo the previous action
	Click to redo the last action
	Click to cut
	Click to copy
	Click to paste
	Click to delete

4.3.4 Development Toolbar

The functions in the **Development** toolbar are described in the following table:

Function	Description
	Click to add a new device
	Click to compile project
	Click to compile device
	Click to execute back & restore of the inverter devices in the project
	Click to upload projects from the device to PC
	Click to download projects from PC to the device
	Click to go online between PC and the device
	Click to go offline between PC and the device
	Click to run the device
	Click to stop the device

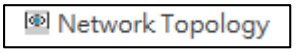
4.3.5 Communication Toolbar

The functions in the **Communication** toolbar are described in the following table:

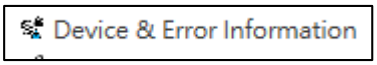
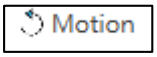
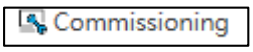
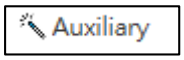
Tool bar icon	Description
	Click to launch COMMGR

4.3.6 Project Explorer

The following table shows the main functions in **Project Explorer**:

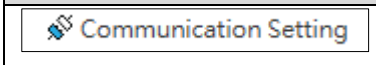
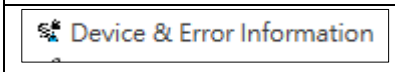
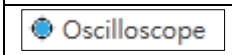
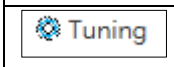
Features	Description
	Add the required device
	Set up device network
	Click to open Network View
	Displays the connected devices
	Displays the unconnected devices
	Displays the added controller
	Displays the added servo drive
	Displays the added inverter
	Displays the added thermostat.

The following table shows the sub-items of the controller functions in **Project Explorer**:

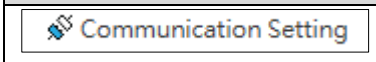
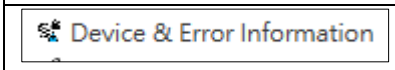
Features	Description
	Set the device communication
	Click to open the view of devices and error information
	Configure hardware, including configuration and parameter setting
	Click to open the view of security setting
	Click to program, including global variables, user-defined data type, task control, program, and function block
	Set the position plan table of motion
	Click to execute commission, including monitor table, data tracer, and data logger
	Provides supporting tools, including register comment, register usage, register edit, resource, step positioning, RTC setting, format PLC memory, and the retain variables initialization

The following table shows the sub-items of the servo drive functions in **Project Explorer**:

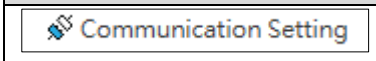
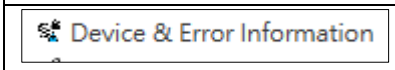
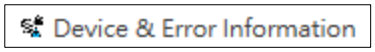
1. Sub-items of ASD-A2 and ASD-B2 functions in Project Explorer

Features	Description
	Set the device communication
	Click to open the view of devices and error information
	Click to open the view of parameter setting
	Click to open the view of oscilloscope
	Click to execute tuning, gain adjustment

2. Sub-items of ASD-A3 and ASD-B3 functions in Project Explorer

Features	Description
	Set the device communication
	Click to open the view of devices and error information
	Click to open the view of oscilloscope
	Click to open the view of Servo Setting Please refer to note of Section 17.3 Servo Setting.

The following table shows the sub-items of the inverter functions in **Project Explorer**:

Features	Description
	Set the device communication
	Set inverter functions
	Click to open the view of devices and error information
	Configure hardware, including configuration and parameter setting
	Parameter backup and restore
	Click to execute commission, including trace and communication test tools
	Set PLC functions
	Click to open the view of devices and error information (for PLC Function)
	Click to open the view of security setting
	Click to program, including global variables, task control, program, and function block
	Click to execute commission, including monitor table (for PLC Function)

Features	Description
Auxiliary	Provides supporting tools, including register comment, register usage, register edit, resource, register resource allocation, step position and format PLC memory.

4.4 Creating Project and Device

User can create projects in Basic Designer as well as Advanced Designer. In Advanced Designer, user can create a new project from **Menu** bar or **Standard** Toolbar or by using keyboard shortcuts.

Follow the following steps to create a new project:

1. Click on New Project icon in the **Standard** toolbar.

   or

   Click **File** > **New Project** from **Menu** bar.

   Result: The **New Project** window displays as shown in the following figure.

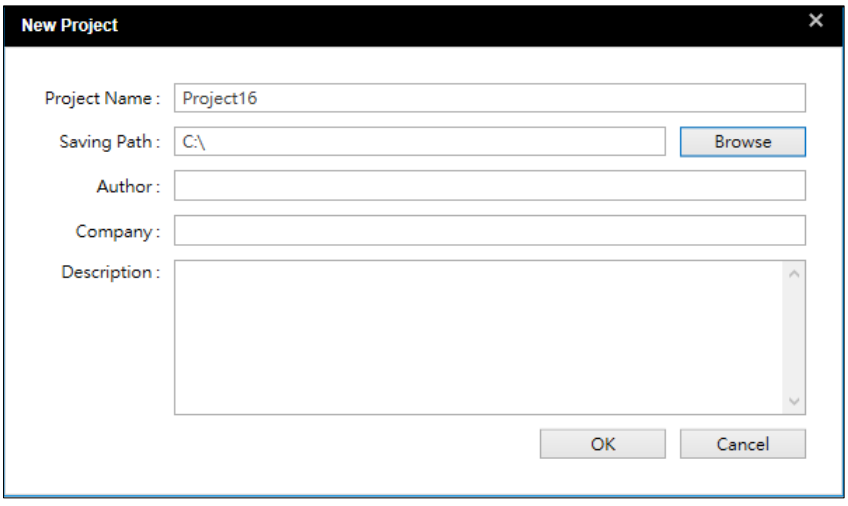


Figure 4 - 32: New Project

***NOTE**: The functions in the **New Project** window are the same as described in the Basic Designer.*

2. Enter the project name, path, and description.
3. Click on **OK** button.

   Result: Project is saved in the given location with *.diade extension.

4.4.1 Add Device

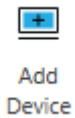
User can add devices to the project in Basic Designer as well as in Advanced Designer. Devices can be added to the project in 2 ways:

- Add new devices manually.
- Add by scan

4.4.1.1 Add New Devices Manually

In Advanced Designer, user can add devices to the project from **Menu** bar or **Development** Toolbar or **Project Explorer**.

Follow these steps to add devices to the project:

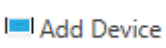
1. Click on 

icon in the **Development** Toolbar.

   or

   Click **Operation** > **Add Device** in the **Menu** bar.

   or

   Click on 

icon in the **Project Explorer**.

   Result: The **Add Device** window displays as shown in the following figure.

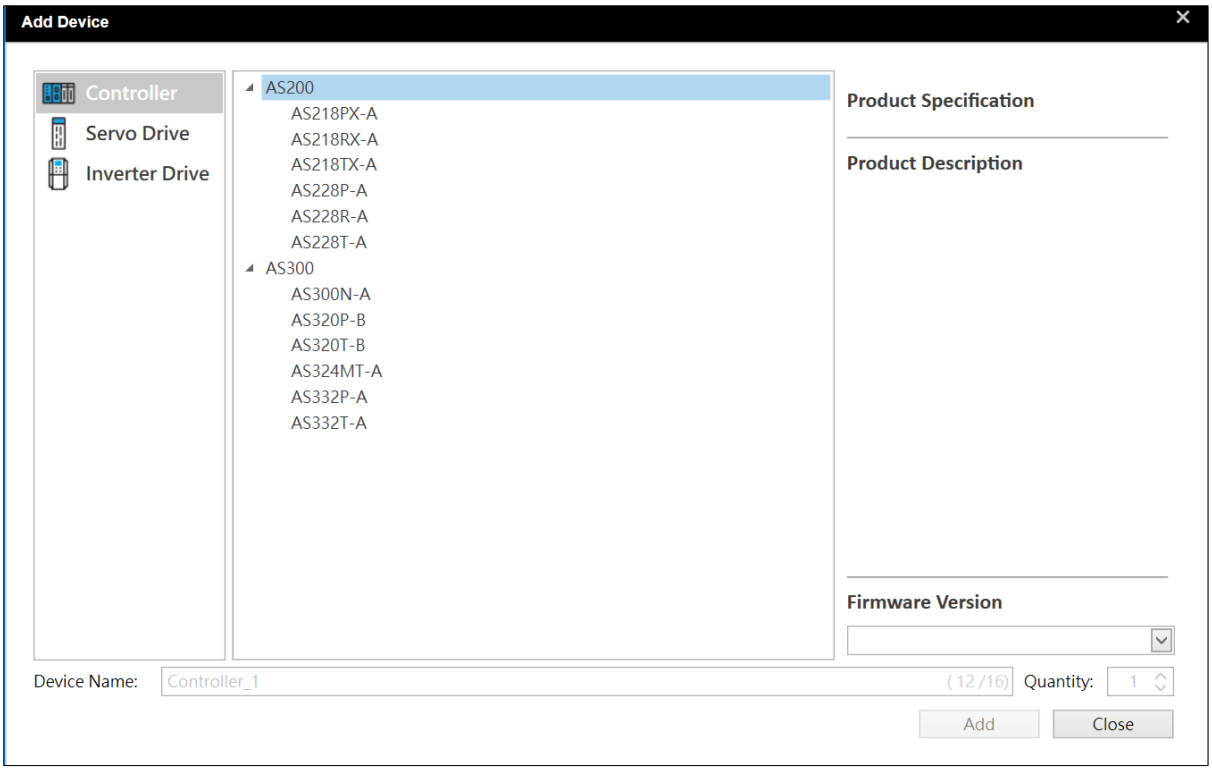


Figure 4 - 33: Add Device

***NOTE**: The **Add Device** window functions are the same as mentioned in Basic Designer.*

4.4.1.2 Add by Scan

User can add Ethernet devices to DIADesigner by scanning them with the help of **Add by Scan** function.

***NOTE**: Only EtherNet/IP device scan is supported in the current version.*

Follow these steps to add devices by scan:

1. Create an **Ethernet** Driver in **COMMGR**.
2. In DIADesigner, navigate to **Network Configuration** > **Network View**.
3. Click on **Network Scan by PC** icon as shown in the following figure.

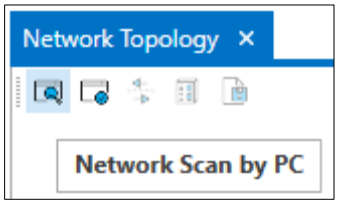


Figure 4 - 34: Network Scan by PC

Result: The **Network Scan** window displays as shown in the following figure.

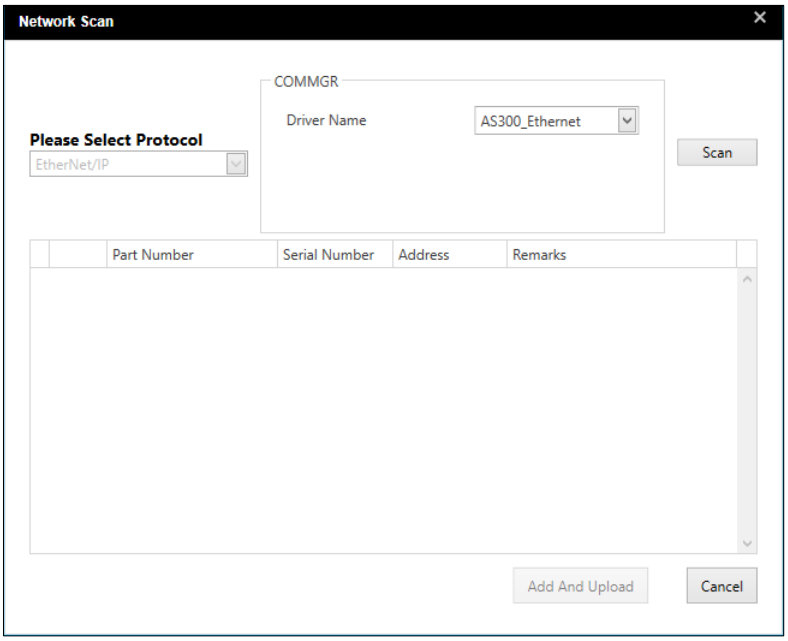


Figure 4 - 35: Network Scan

4. Select the Ethernet driver in **Driver Name** drop-down list
5. Click on **Scan** button to start scanning.

   Result: The **Network Scan** window starts scanning the devices as shown in the following figure.

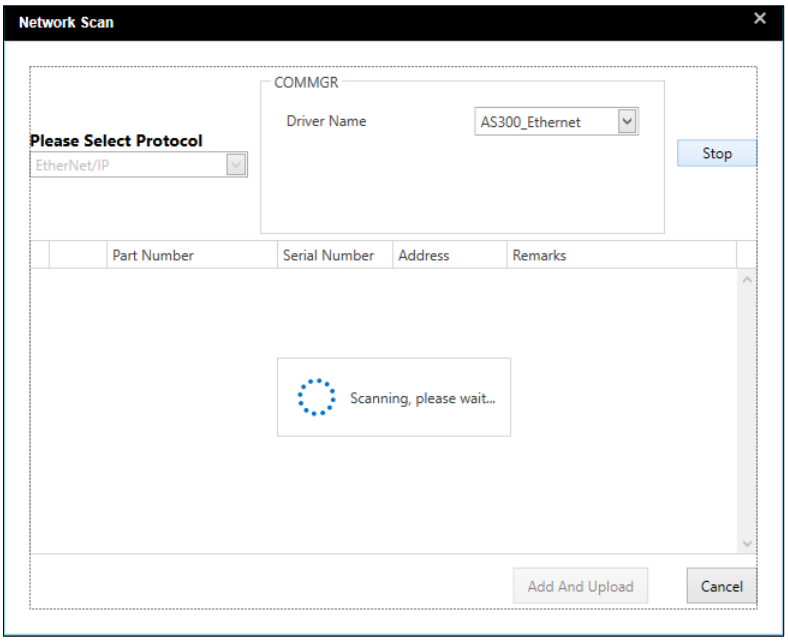


Figure 4 - 36: Network Scan – Scanning

***NOTE**: Wait for the system to scan the devices on the network.*

6. Click on **Stop** button to stop the scanning.

   Result: The **Network Scan** window stops scanning for devices and displays the updated scanned devices as shown in the following figure.

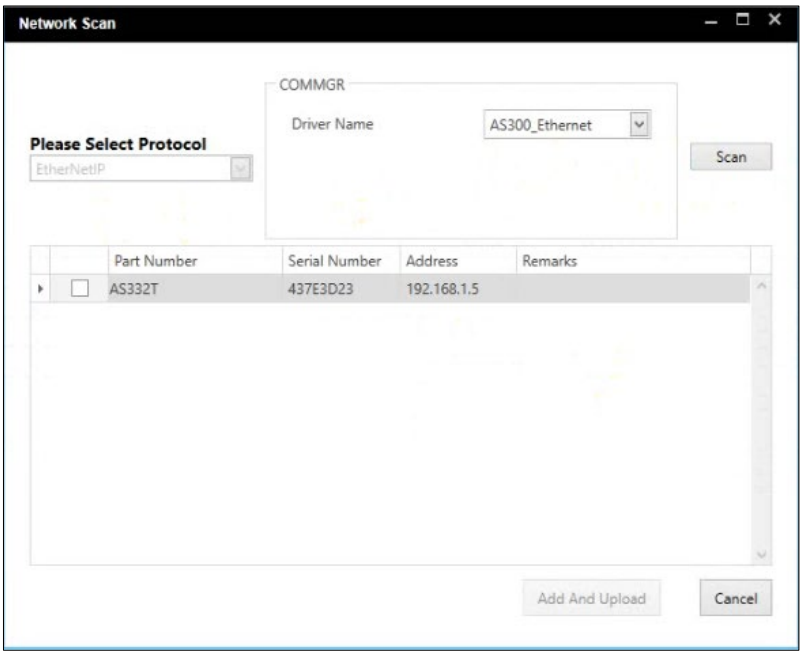


Figure 4 - 37: Network Scan – Result

NOTE: *The* ***Part Number, Serial Number, IP Address*** *of the scanned devices will be displayed in the* ***Network Scan*** *window.*

7. Select the devices which user wants to add to the project and click on **Add And Upload** button.

   Result: The selected devices are updated to the DIADesigner project.

4.4.1.3 Directly Connect to Inverter

Please refer to *4.2.7.4 Inverter Wizard* for the details about directly connecting to inverters.

4.4.2 Device Name

User can modify the default device name by the following methods:

1. Right-click on the device and opens the context menu. Select **Update Device**.

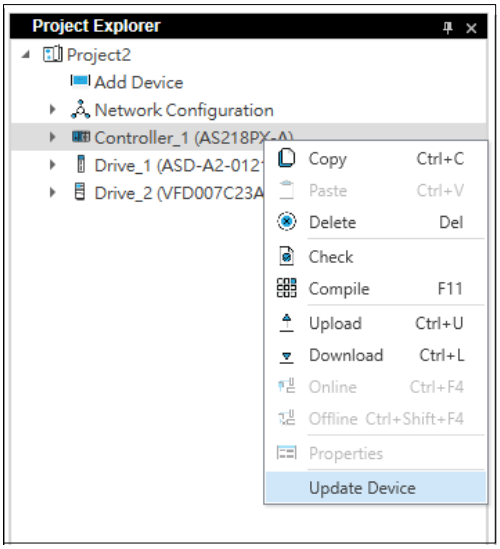


Figure 4 - 38: Context Menu to Update Device

Result: The **Update Device** window will pop up, and user can enter a new name.

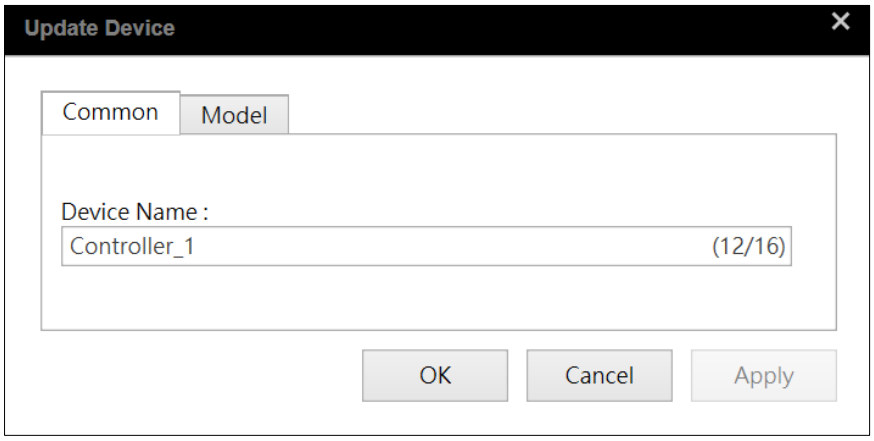


Figure 4 - 39: Update Device Pop-up Window

2. Select the device tree node, then click on the device.

   Result: User can enter a new name.

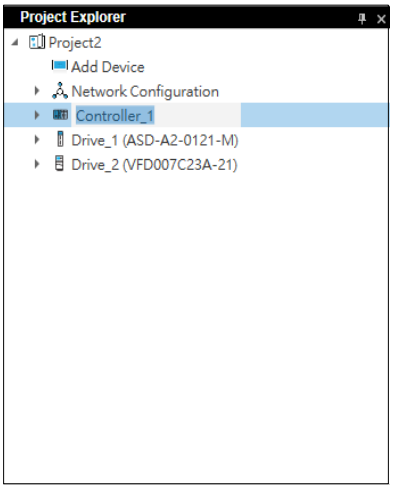


Figure 4 - 40: Device Tree Node to Update Device Name

3. Select the device tree node, and press F2

   Result: User can enter a new name.

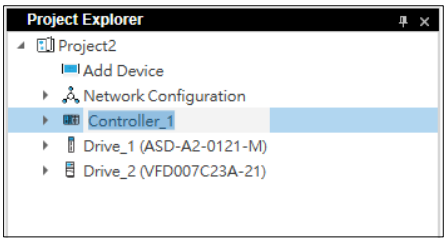


Figure 4 - 41: Hotkey F2 to Update Device Name

4. Double-click on the device name in **Network View** view.

   Result: User can enter a new name.

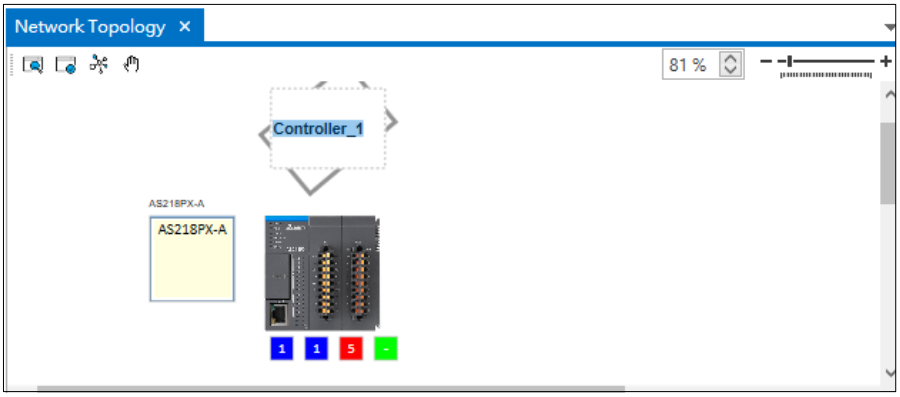


Figure 4 - 42: Network View to Update Device Name

5. Click the device name in **Network View** view, and press F2

   Result: User can enter a new name.

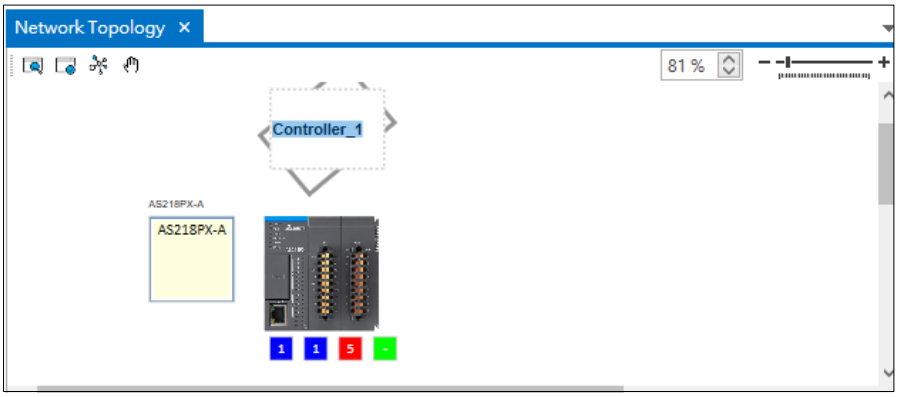


Figure 4 - 43: Hotkey F2 to Update Device Name in Network View

4.4.3 Update Device

User can update devices by the following methods:

1. Select the device tree node, right-click on the device to open its context menu. Select **Update Device > Model**.

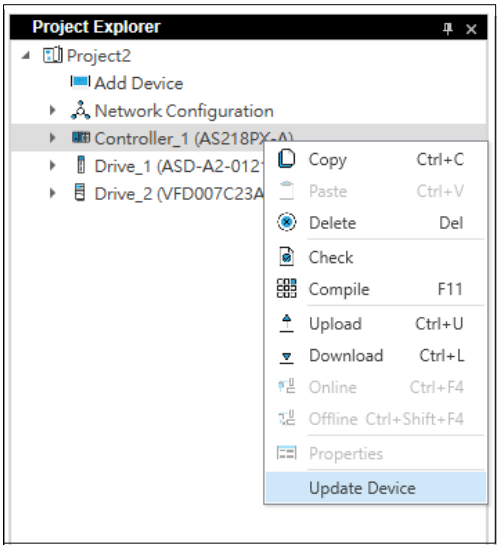


Figure 4 - 44: Context Menu to Update Device

Result: The **Update Device** window will pop up.

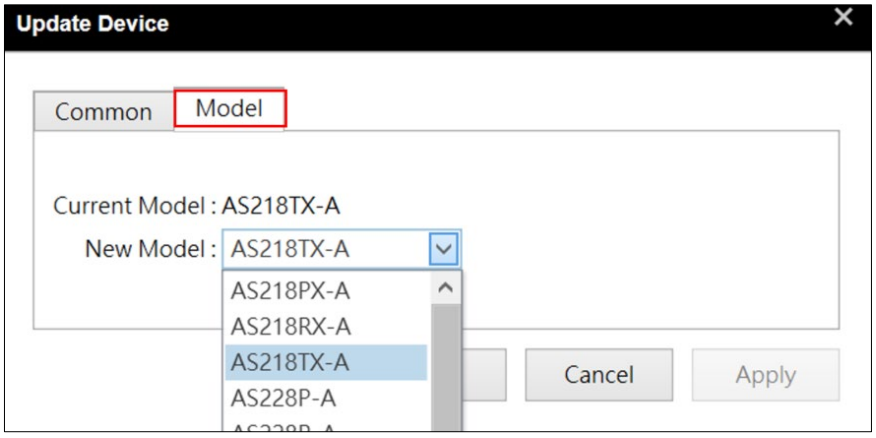


Figure 4 - 45: Update Device Model

2. In **Network View** view, right-click on the device image and opens the context

menu. Select **Update Device > Model**.

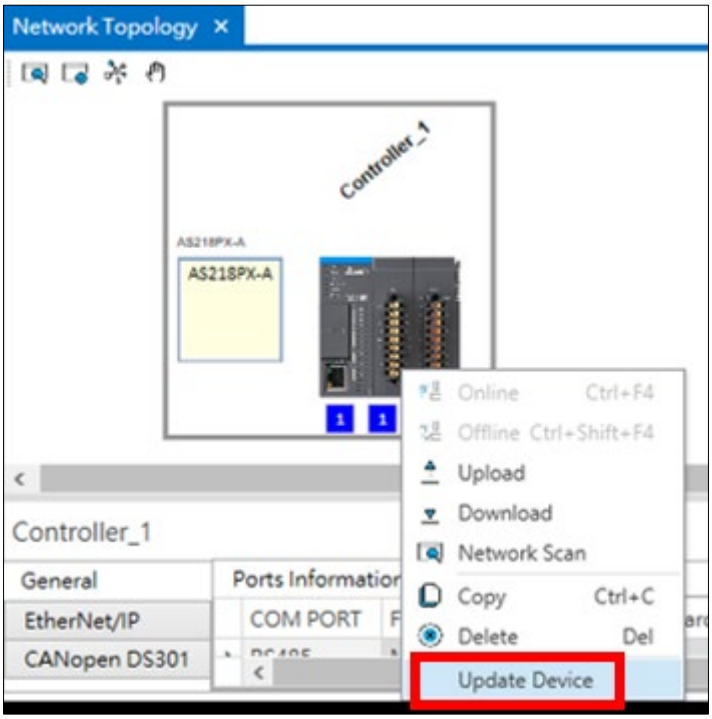


Figure 4 - 46: Update Device Model in Network View

Result: The **Update Device** window will pop up.

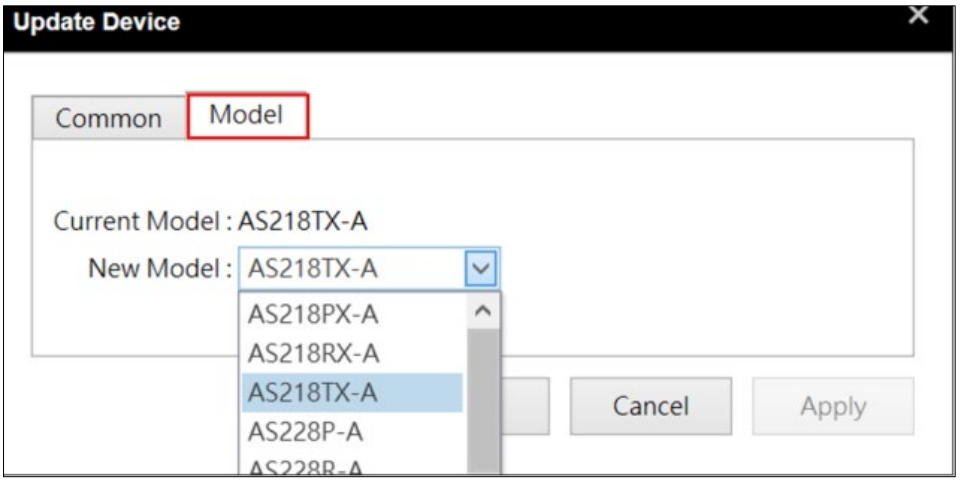


Figure 4 - 47: Update Model

3. In **Hardware Configuration** view, right-click on the device image and open the context menu. Select **Update Device > Model**.

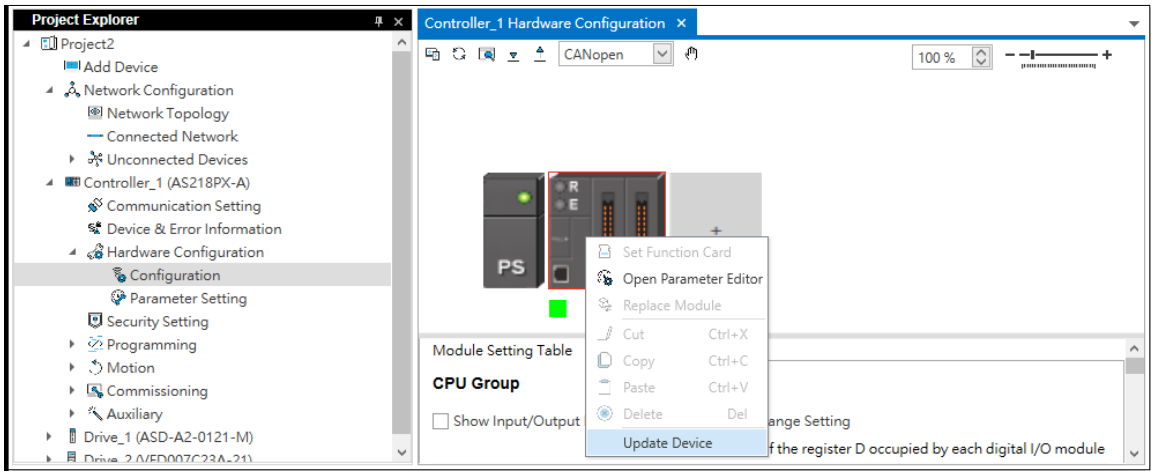


Figure 4 - 48: Hardware Configuration

Result: The **Update Device** window will pop up.

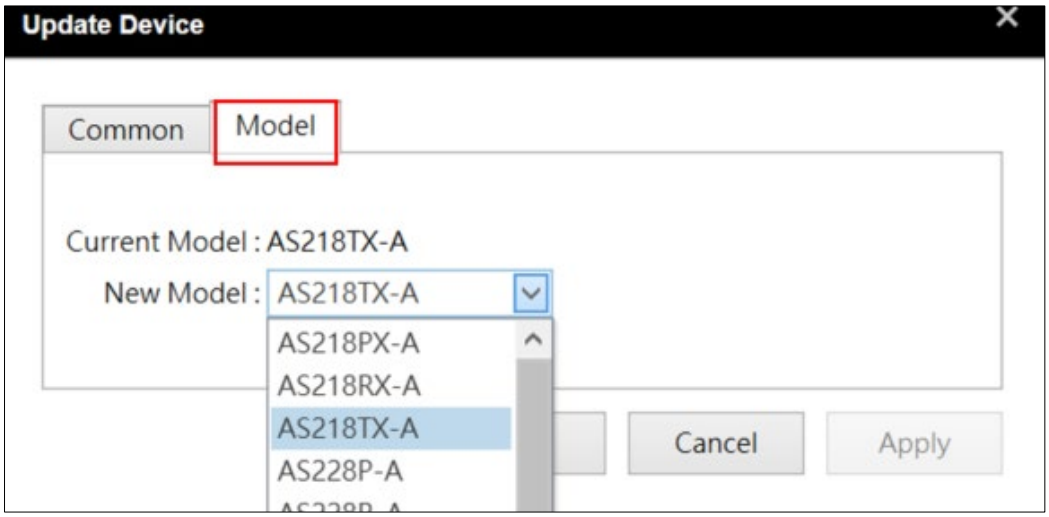


Figure 4 - 49: Update Device Window

Chapter 5: Communication Setting

5.1 Communication via COMMGR

User can enable communication between DIADesigner and connected devices using **COMMGR** (Communication Manager). This is done with the help of interface function in **Communication Setting** window inside DIADesigner. The **Communication Setting** window can be accessed from the **Project Explorer** window through **Device** > **Communication Setting**. An example of **Communication Setting** window for PLC (Controller) device is shown in the following figure.

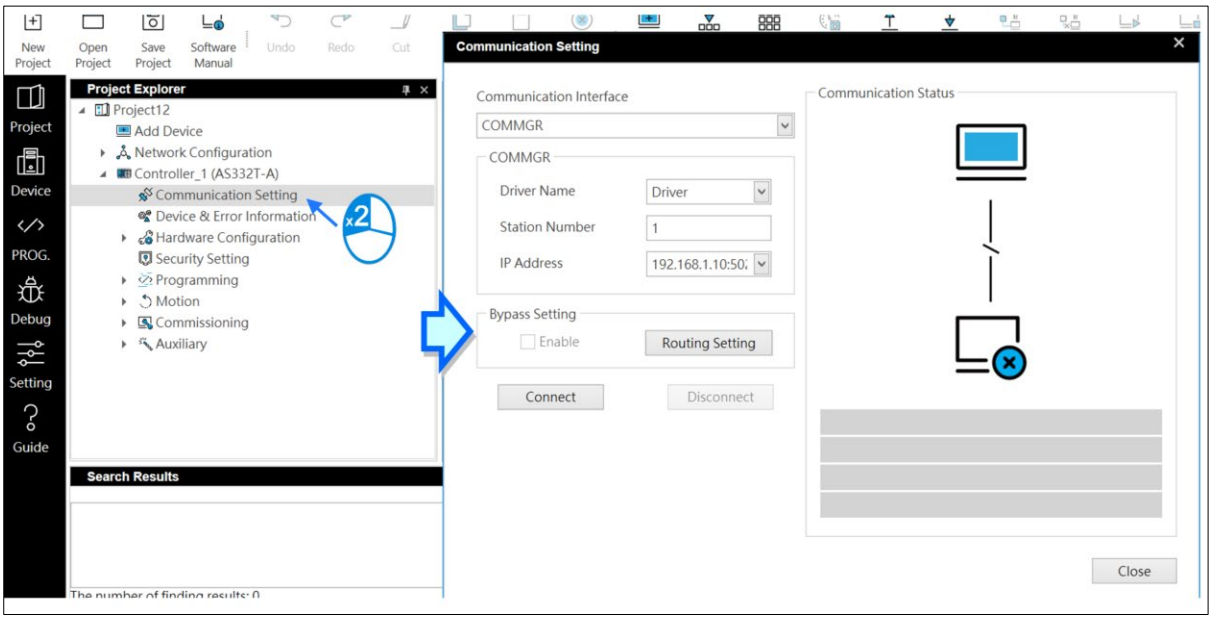


Figure 5 - 1: Communication Setting Window

The functions in the **Communication Setting** window are described in the following table:

Function	Description
Interface	Select the device interface. Options are: • COMMGR • Serial Port • Ethernet (When using inverter communication card to

Function	Description
	connect PC) ***NOTE**: Serial Port is available for devices other than Controller devices such as Servo Drive or AC Drive.*
COMMGR – Driver Name	Select the Driver name created in COMMGR in this drop-down list.
COMMGR – Station Number	Enter the **Station Number** defined in COMMGR.
Bypass - Enable	Click to enable bypass settings.
Bypass – Routing Setting	Click to open **Routing Setting** window to set path.
Connect	Click to connect to the device.
Disconnect	Click to disconnect the device
Communication Status	Displays the communication status

5.2 Communication without COMMGR

User can communicate DIADesigner with devices like Servo Drive, AC Drive and so on, with the help of Serial Port option available in **Communication Settings** – **Interface**. Select the **Connector** on the device and the **Serial COM Port** on the connected computer.

An example for communication using Serial COM Port with **CN4** Connector of ASD-A2 Servo Drive is shown in the following figure.

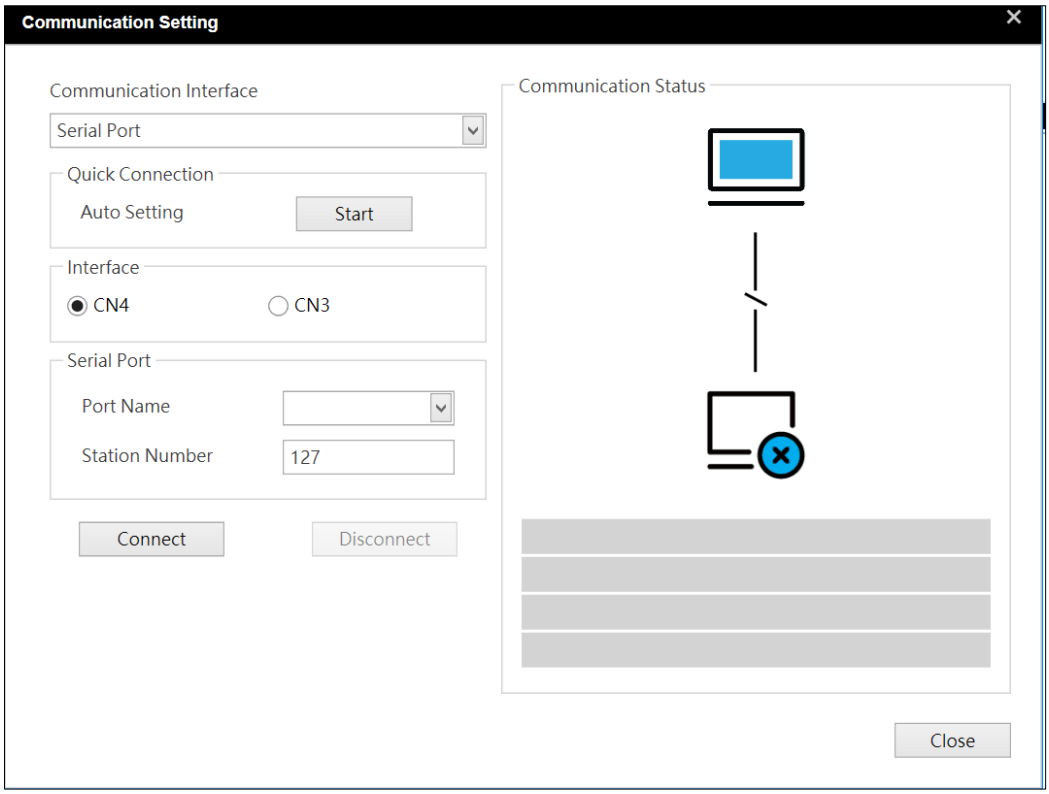


Figure 5 - 2: Communication Setting – Serial Port

Click on **Start** button in **Quick Connection** box to setup communication protocol automatically. The communication protocol parameter is pre-filled.

Connector	Description
CN4	CN4 is the USB communication port on the Servo Drive.
CN3	CN3 is the RS-485/RS-232 Serial Communication port on the Servo Drive.

An example for communication using Serial COM Port with **CN3** Connector of ASD-A2 Servo Drive is shown in the following figure.

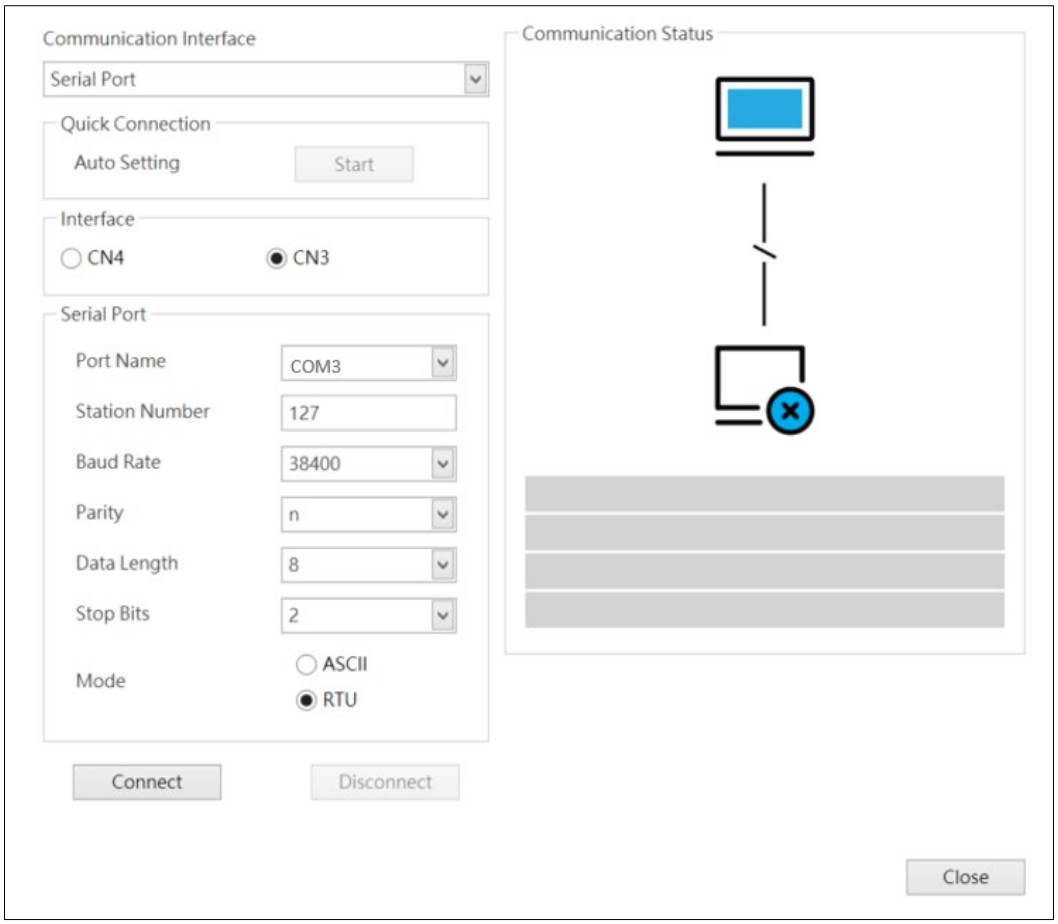


Figure 5 - 3: Communication Setting – Connector and Drive

Serial Port Function	Description
Port Name	Select the COM port on the computer.
Station Number	Enter the station number of the device.
Baud Rate	Select the baud rate.
Parity	Select the parity.
Data Length	Select the data length.
Stop Bits	Select the number of stop bits.
Protocol	Select ASCII/RTU protocol.

5.3 COMMGR (Communication Manager)

The COMMGR is the new generation communication management tool developed by Delta Electronics, Inc. It functions as a communication bridge between DIAStudio

and connected recognized hardware devices. Communication becomes more convenient and more efficient through the management of COMMGR.

5.3.1 Overview

User 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. If a driver is started, a connection is created automatically. After the user specifies a driver in DIADesigner, communication is carried out at the same time.

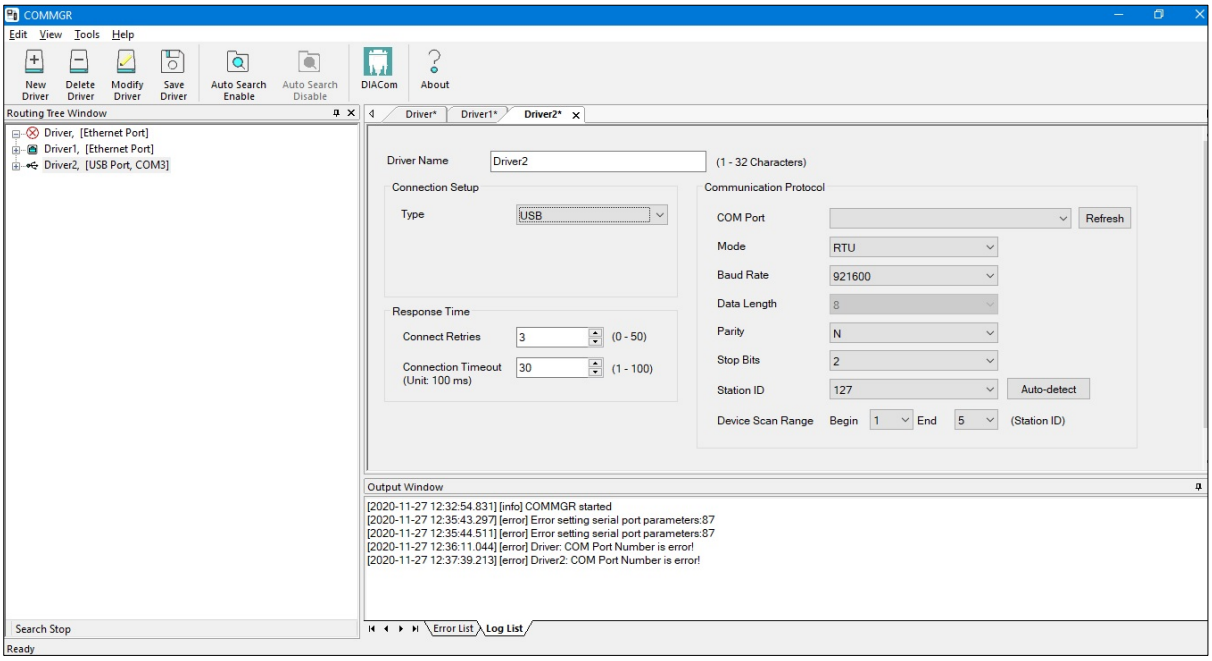


Figure 5 - 4: Setting – COMMGR Window

The communication between DIADesigner and Delta hardware product via Drivers is shown in the following figure. The COMMGR is a communication interface between DIADesigner and Delta hardware product.

Figure 5 - 5: Communication Diagram

The COMMGR is a software independent of DIADesigner. It must be installed and uninstalled separately. If an older version of COMMGR has been installed on a computer, user have to uninstall it before installing the new version.

Refer to *Chapter 3* for more details.

5.3.2 Launch COMMGR

Follow these steps to launch COMMGR:

1. Double-click on the COMMGR shortcut on the desktop or

   Click Windows icon on the windows toolbar and click **Delta Industrial Automation** > **COMMGR** as shown in the following figure.

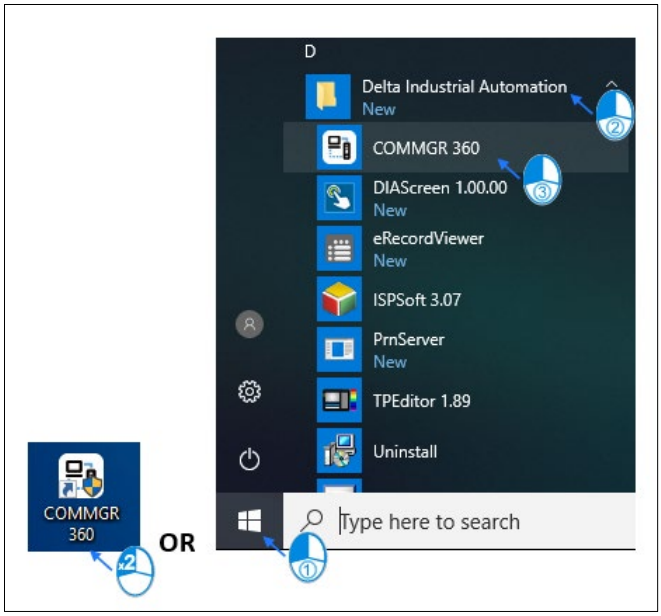


Figure 5 - 6: Launch COMMGR

2. Right-click on icon in the windows toolbar area and click on the **Open COMMGR** option as shown in the following figure.

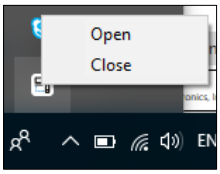


Figure 5 - 7: Launch COMMGR - window toolbar

Result: The **COMMGR** application launches.

5.3.3 Main Screen

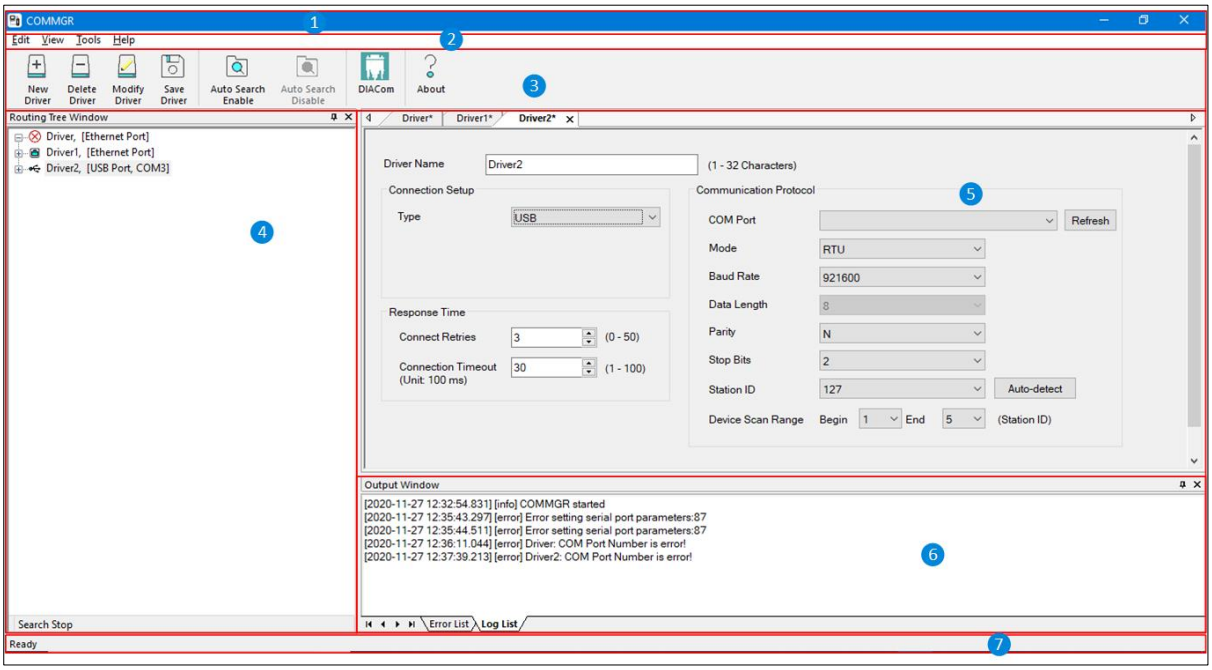


Figure 5 - 8: COMMGR – Main Screen

1. Windows title bar
2. Menu bar
3. Toolbar

4. Routing Tree window
5. Driver Setup window
6. Output window
7. Status bar

5.3.3.1 Windows Title Bar

The COMMGR Windows Title bar displays the name of the product, product icon and icons to minimize, resize/maximize and close the COMMGR. The COMMGR **Windows Title Bar** is shown in the following figure.

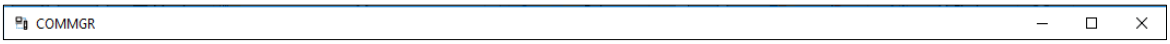


Figure 5 - 9: COMMGR - Windows Title Bar

5.3.3.2 Menu Bar

The COMMGR Menu bar provides functions to perform driver management in the form of menu items. The COMMGR **Menu Bar** is shown in the following figure.

Figure 5 - 10: COMMGR - Menu Bar

The functions in the COMMGR **Menu Bar** are described in the following table:

Menu - Function	Description
Edit – New Driver	Click to add new driver.
Edit – Delete Driver	Click to delete selected driver.

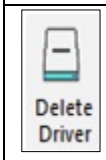
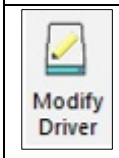
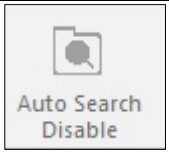
Menu - Function	Description
Edit – Modify Driver	Click to modify selected driver.
Edit – Save Driver	Click to save the modified driver.
View – Routing Tree Window	Click to view/hide **Routing Tree Window**.
View – Status Bar	Click to view/hide status bar.
View – Output Window	Click to view/hide **Output Window**.
Tools – IP Manager	Click to open **IP Manager** window.
Tools – Ping Test	Click to open **Ping Test** window.
DIACom	Click to open **DIACom** setting tool.
Tools - Language	Click and select language. Languages supported are: • Traditional Chinese • Simplified Chinese • English (US)
Help – About	Click to open **About** window.

5.3.3.3 Tool Bar

The COMMGR tool bar provides functions for driver management. The COMMGR **Tool Bar** is shown in the following figure.

Figure 5 - 11: COMMGR - Tool Bar

The functions in the COMMGR **Tool Bar** are described in the following table:

Tool Bar - Icons	Description
	Click to add new driver.
	Click to delete selected driver.
	Click to modify selected driver.
	Click to save the modified driver setting.
	Click to open the automatic search function.
	Click to turn off automatic search.
	Click to open **About** window.

5.3.3.4 Routing Tree Window

The COMMGR **Routing Tree Window** displays the drivers created by user and the driver status in the bottom of the Routing Tree. The **Routing Tree Window** is shown in the following figure.

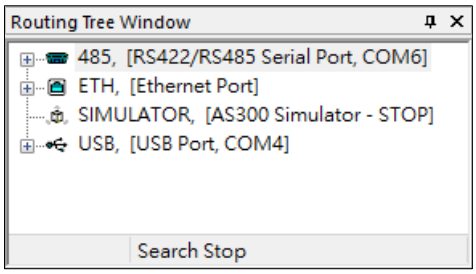


Figure 5 - 12: COMMGR - Routing Tree Window

5.3.3.5 Driver setup

The COMMGR **Driver** setup window displays the driver setting. Double-click on a driver in **Routing Tree Window** to open the **Driver** setup window. The **Driver** setup window is shown in the following figure.

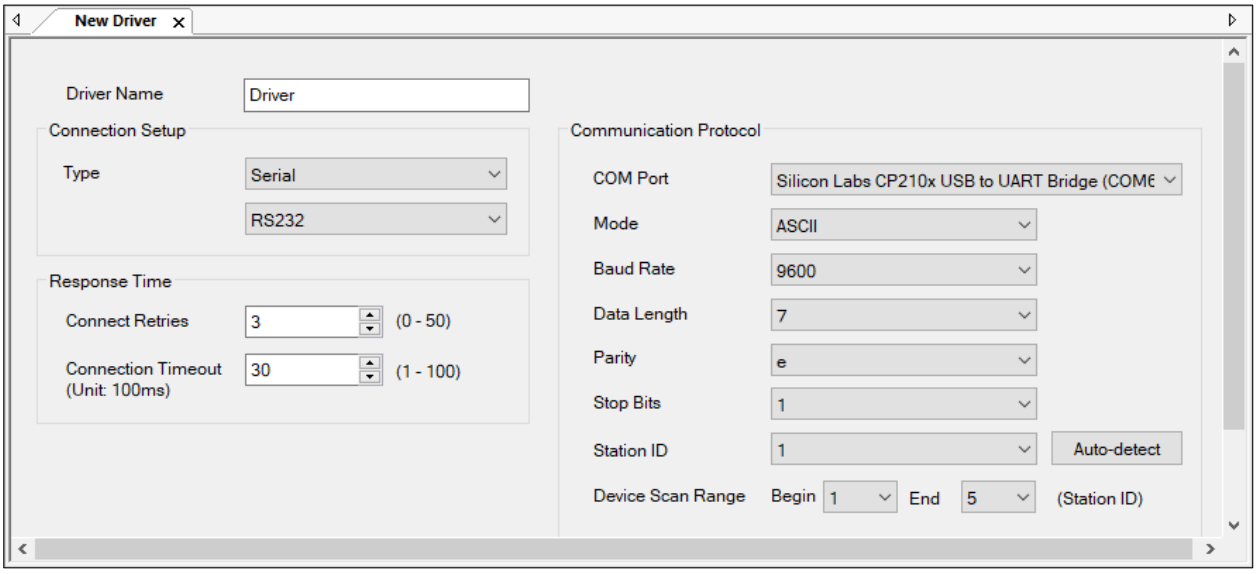


Figure 5 - 13: Driver Setup Window

5.3.3.6 Output Window

The COMMGR **Output Window** displays the **Error List** and **Log List** of Drivers in two different tabs. **Output Window** is shown in the following figure.

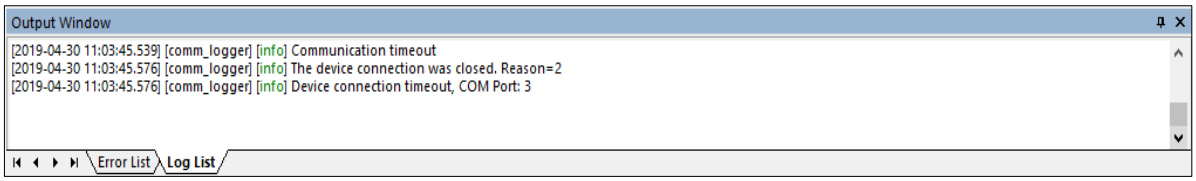


Figure 5 - 14: Output Window

5.3.3.7 Status Bar

The COMMGR **Status Bar** displays the status of the software. The COMMGR **Status Bar** is shown in the following figure.

Figure 5 - 15: COMMGR - Status Bar

5.3.4 Driver Setting

The drivers listed in the COMMGR window connects software and the communication ports. When user add a new driver and set the communication parameter for it, COMMGR will create a connection to the communication port specified for that driver.

Whenever the computer restarts, COMMGR will also start the driver automatically. However, if the connectivity of the driver created is lost (for example, the Ethernet Network card or the USB connection cable is removed), COMMGR will stop the driver and display Error state for the driver. Once the connectivity is back, the driver status is OK again. An example for error state is shown in the following figure.

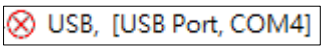


Figure 5 - 16: COMMGR - Driver in error state

5.3.4.1 Driver setup

Follow these steps to open COMMGR and perform driver management:

1. Click **Edit** > **New Driver** in the **Menu Bar**.

   or

   Click on

   icon in the **Toolbar**.

   Result: The **Driver** setup window displays as shown in the following figure.

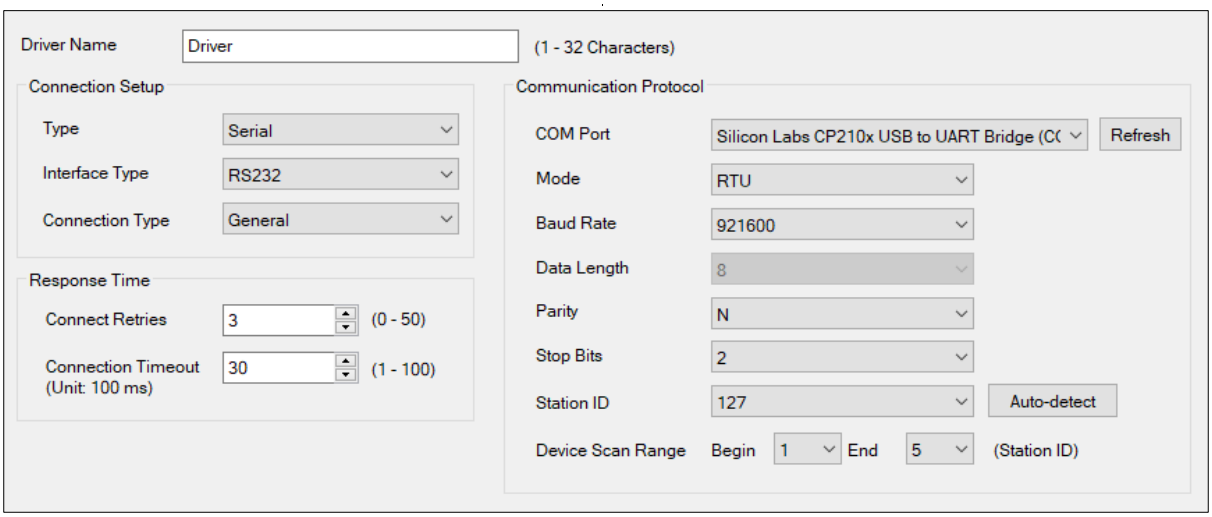
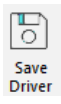


Figure 5 - 17: Driver setup

2. Enter **Driver Name**, select **Connection Setup** > **Type** and set the **Communication Protocol** parameters as explained in the following sections as per user requirement.

3. Click on 

   icon in the **Toolbar**.

   Result: Communication **Driver** is created.

Driver Name:

*User can type a name in the **Driver Name** box. A driver name is composed of maximum 31 characters. Special characters such as *, #, ?, \, %, @ and etc. cannot be used except _. **Driver Name** is shown in the following figure.*

Figure 5 - 18: Driver Name

Connection Setup > Type:

*User can select a connection type in **Type** drop-down list. The supported connection types are shown in the following figure.*

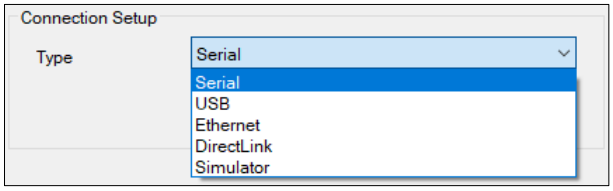


Figure 5 - 19: Communication Setup > Type

Connection Types: -

***Serial**:*

A computer communicates with a PLC through a communication port on the computer using RS-232/RS-422/RS-485.

***USB**:*

A computer can connect to a PLC equipped with a USB port through a USB cable. However, user have to make sure that a USB driver is installed on a computer before the computer connects to a PLC equipped with a USB port. Refer 17.1 Installing USB driver for AS series CPU module for more information about installing PLC USB driver.

***Ethernet**:*

A computer communicates with a PLC through an Ethernet port on the computer.

DIACloud:

Use DIACom to create a computer's virtual Ethernet port (DIACom Ethernet Adaptor) or virtual communication port (DIACom Virtual COM) using RS-232/RS-422/RS-485 to communicate with Delta cloud-connected devices.

***DirectLink**:*

***DirectLink** is a communication function provided by Delta Human Machine Interfaces (HMI). If a PLC connects to a HMI normally, a computer can connect to the HMI through a USB cable or Ethernet, and connect to the PLC indirectly. Refer Delta Human Machine Interface (HMI) manuals for more information about setting a connection.*

Simulator:

*AS **Simulator** type of driver functions as a virtual channel connecting a Simulator. If user specifies this type of driver in DIADesigner, COMMGR will transmit the data between DIADesigner and a Simulator once related connection operation is performed. It is as if COMMGR is connected to a real PLC.*

***Communication Parameters**:*

*Communication Parameters are set according to the connection type selected. Different connection types have different connection parameters. The setting of parameters for different connection types is defined in the below sections. **Setup Response Time** settings are shown in the following figure.*

*User can select the number of times the sending of a command is re-tried if a connection error occurs in the **Connect Retries** box and select an interval of retrying the sending of a command in the **Connection Timeout** box.*

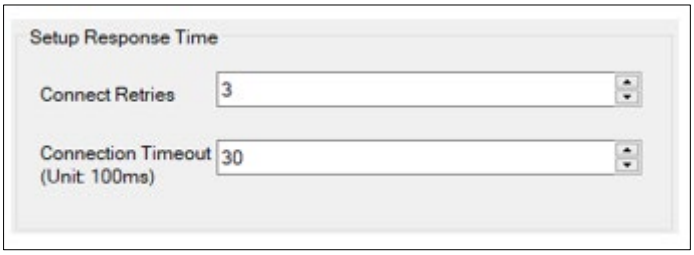


Figure 5 - 20: Setup Response Time

5.3.4.2 RS232/422/485 Setting

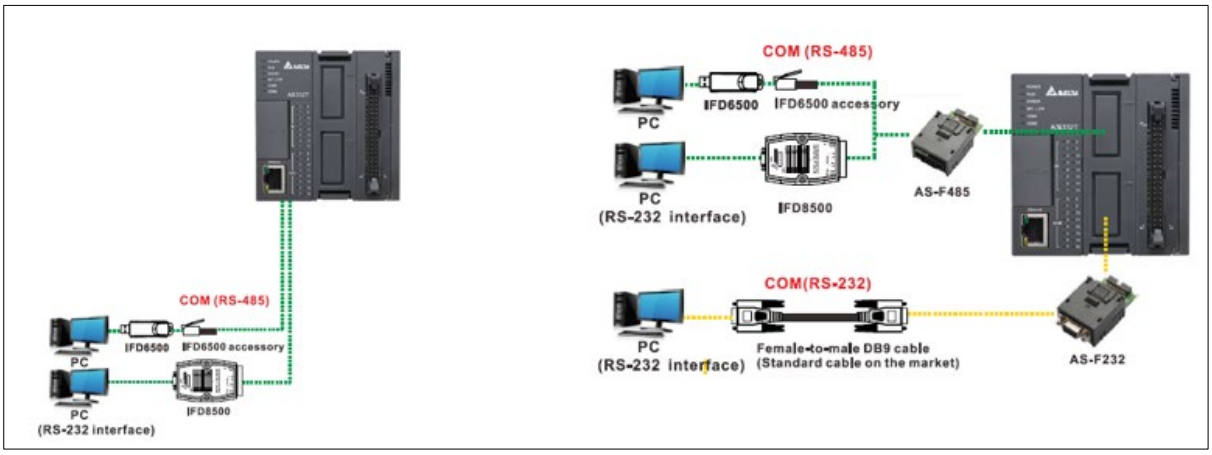


Figure 5 - 21: RS232/422/485 Setting

Pre-requisite:

Before setting up a RS-232/RS-422/RS-485 driver, make sure:

a) A driver program is successfully installed in the computer

b) The connection between the computer and the PLC has been established and is working properly.

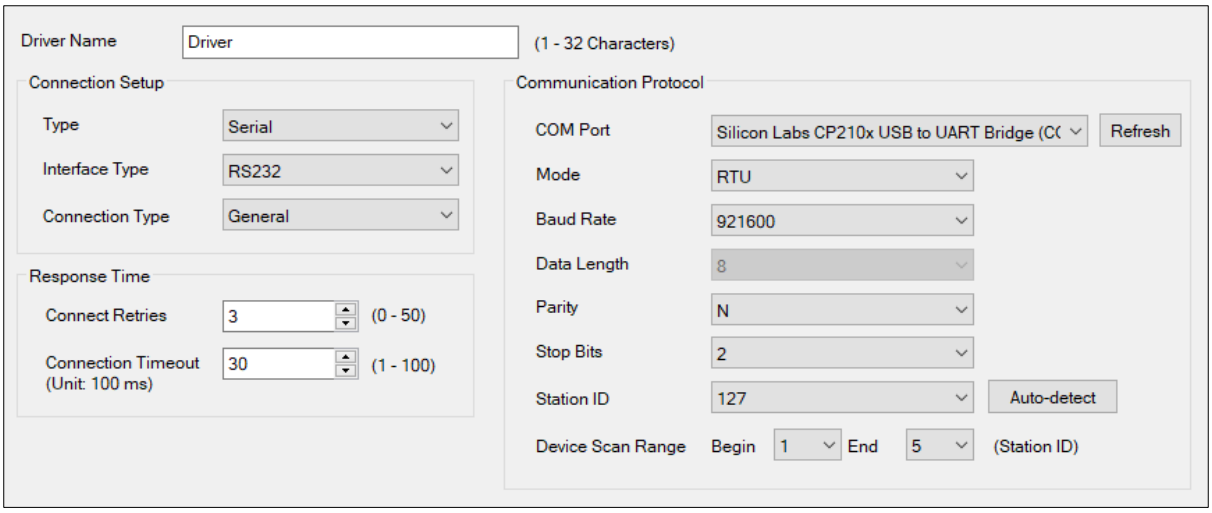


Figure 5 - 22: Drive setup – RS232

- User can type a driver name in the **Driver Name** field. Special characters cannot be used except _.
- Select **Serial** in the **Type** drop-down list in the **Connection Setup** section.
- Select a communication port in **COM Port** drop-down list. Each item in the **COM Port** drop-down list is composed of a device name and a communication port number. The communication ports in the **COM Port** drop-down are the same as the communication ports in the Windows **Device Manager**. If the pre-requisites mentioned in (a) and (b) are not complete, some options will not be shown here.
- The communication **Mode** can be **ASCII** or **RTU**.
- The communication protocol for exchanging the data through a communication port selected must be the same as the communication protocol for exchanging data through the communication port on a device connected.

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

5.3.4.3 USB Setting

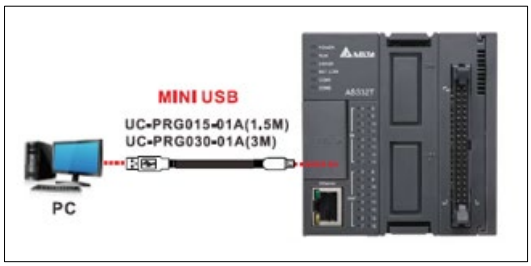


Figure 5 - 23: USB Setting

Pre-requisite:

Before setting up a USB driver, make sure:

a) A USB driver is installed on the computer.

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

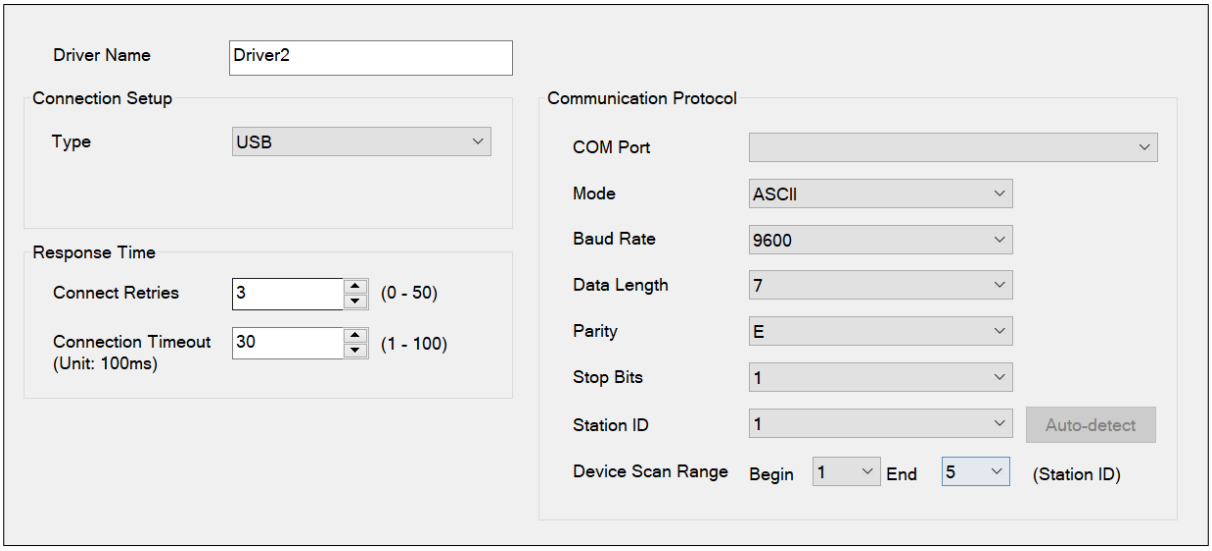


Figure 5 - 24: Drive setup – USB

- User can type a driver name in the **Driver Name** box. Special characters cannot be used except _.
- Select **USB** in the **Type** drop-down list in the **Connection Setup** section.

- Select a communication port in the **COM Port** drop-down list. The communication ports in the **COM Port** drop-down list are the same as the communication ports in the Windows **Device Manager**. If the pre-requisites mentioned in (a) and (b) are not complete, some options will not display.

***NOTE**: Refer 17.1 Installing USB driver for AS series CPU module for more information about installing a **USB driver**.*

5.3.4.4 DirectLink (USB) Setting

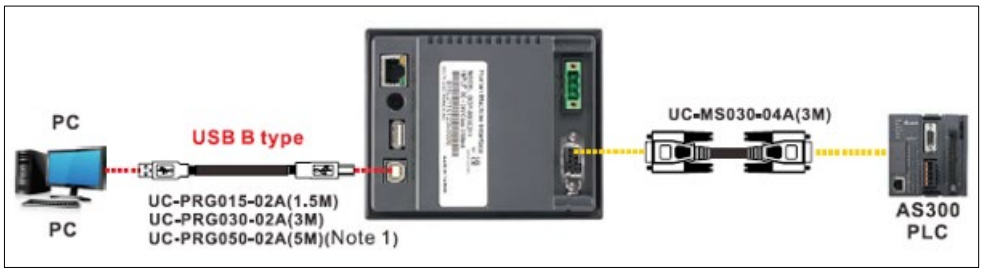


Figure 5 - 25: DirectLink (USB) Setting

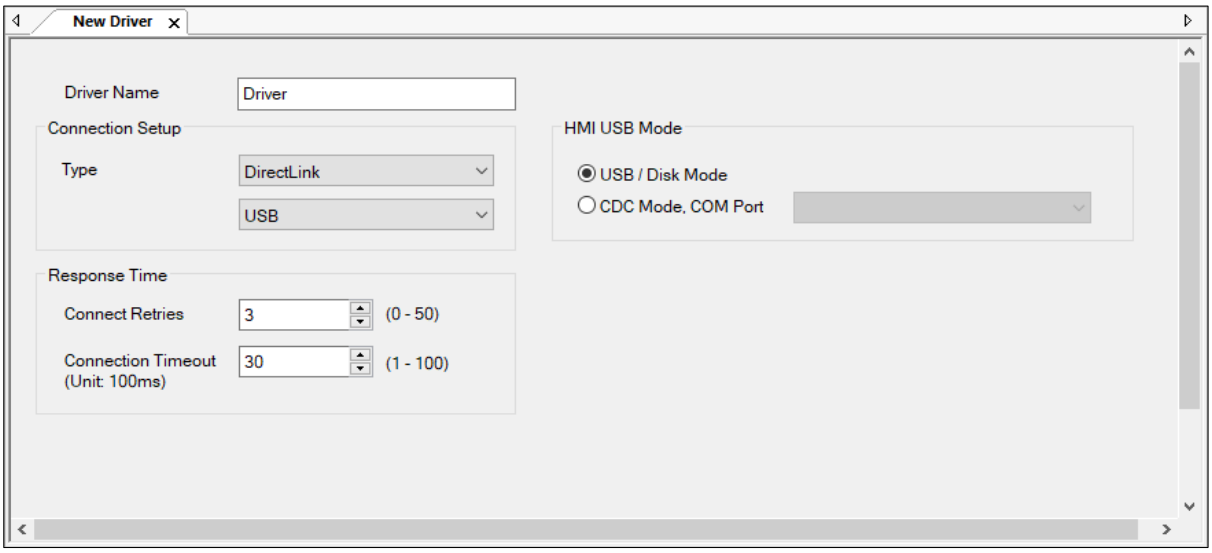


Figure 5 - 26: Driver setup – Direct Link (USB)

- User can type a driver name in the **Driver Name** box. Special characters cannot

be used except _.

- Select **DirectLink** in the **Type** drop-down list in the **Connection Setup** section. By default, the value of sub-type (drop-down list **Type**) is **USB**.
- Select **CDC Mode, COM Port** in the **HMI USB Mode** section and use the drop-down list to see the complete **COM Port** number and device name; the contents are the same as the **Windows Device Manager**. Refer Delta HMI related manuals for more information.

5.3.4.5 Ethernet Setting

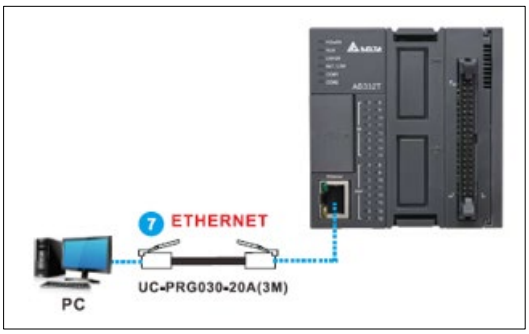


Figure 5 - 27: Ethernet Setting

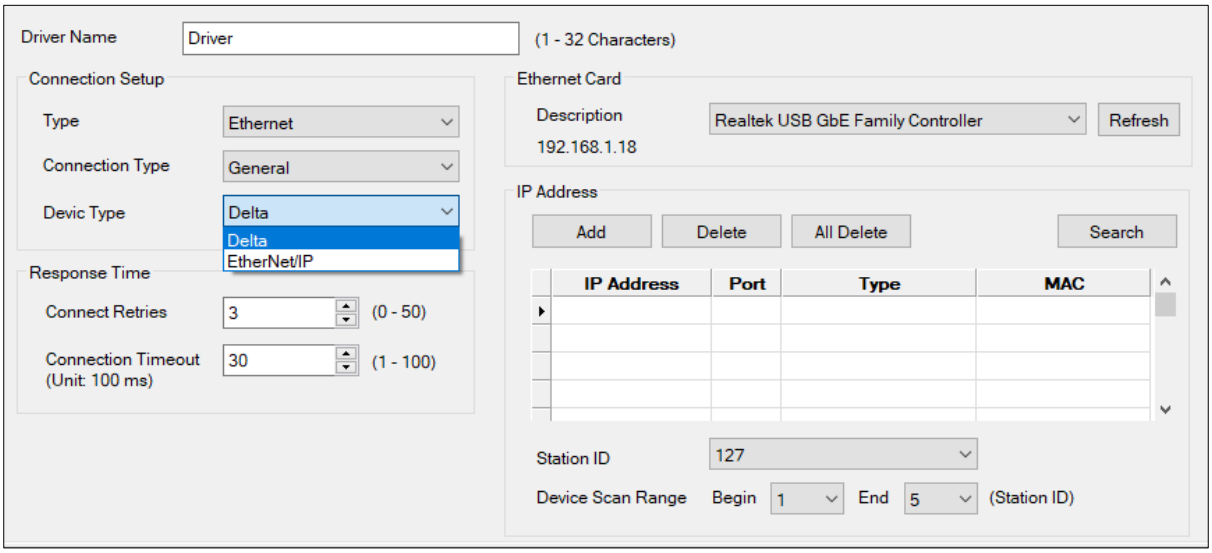


Figure 5 - 28: Driver setup - Ethernet

- User can type a driver name in the **Driver Name** box. Special characters cannot be used except _.
- Select **Ethernet** in the **Type** drop-down list in the **Connection Setup** section.
- Select
- Select a Network Interface Card (NIC) from the **Description** drop-down list option in the **Ethernet Card** section. An IP address assigned to a network interface card selected displays in the lower left corner of the **Ethernet Card** section.
- Owing to the characteristics of Ethernet, a computer can communicate with all devices on a network. User can create the IP addresses of devices connected to this driver in the **IP Address** section.
  - Some devices support the **Search** function. After user click **Search** to search for IP addresses, results display in the **IP Address** section.
  - After user click **Add** to add new IP address to the list of IP addresses in the **IP Address** section, they can type related information in the **IP Address**, the **Port** number, the **Label** and the **Type**.
    1. User can type the IP address of a device connected in **IP Address** column
    2. User can type the communication port number specified. If it is not necessary to specify a communication port number, use the default communication port number **502** in the **Port** column.
    3. User can type a comment in **Label** column.
    4. User cannot type a device in **Type** column. After user click **Search**, a device which is found displays in this column.
- User can delete a particular device by selecting it and clicking **Delete** or pressing **Delete** key in the keyboard.

***NOTE**: When a **Driver** setup window is opened, the information about the network interface card in the computer is retrieved once. However, the information in the **Description** drop-down list will not be updated. If a network interface card is added to the computer system after the **Driver** setup window is opened, the network*

*interface card will not be displayed in the **Description** drop-down list. User have to close the **Driver** setup window, and open it again.*

5.3.4.6 DirectLink (Ethernet) Setting

Figure 5 - 29: DirectLink (Ethernet) Setting

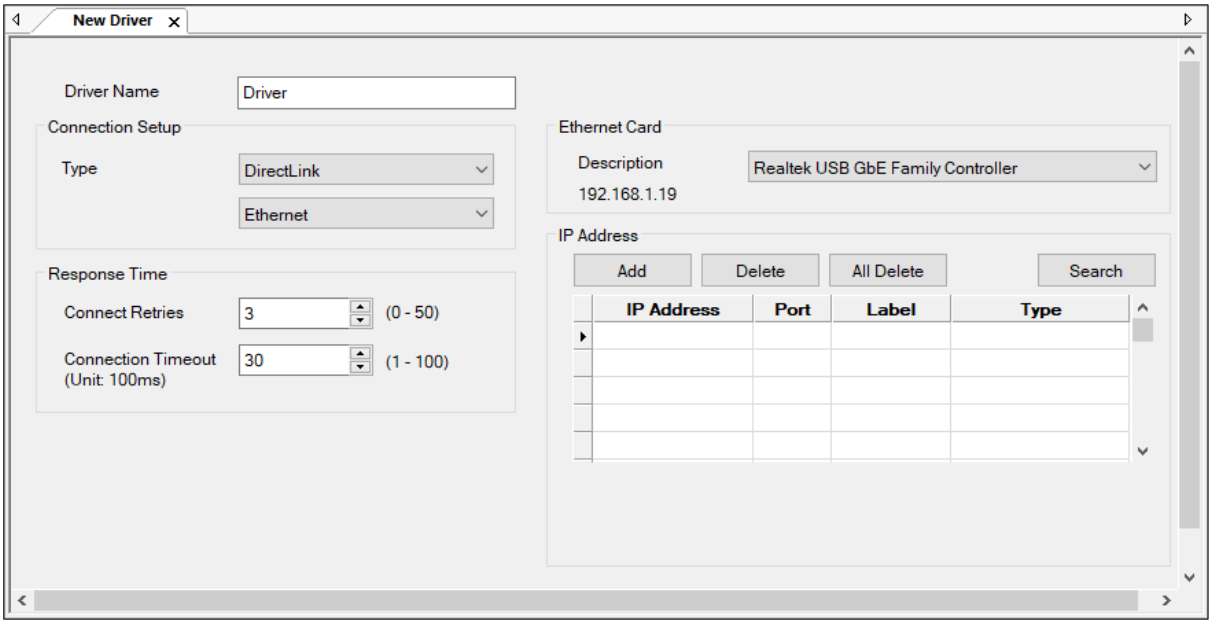


Figure 5 - 30: Driver setup – DirectLink (Ethernet)

- User can type a driver name in the **Drive Name** box. Special characters cannot be used except _.
- Select **DirectLink** in the **Type** drop-down list in the **Connection Setup** section. Select **Ethernet** in the drop-down list the **Type** drop-down list. By default, the value of sub-type (drop-down list **Type**) is **USB**. Select Ethernet here.

- Select a Network Interface Card (NIC) from the **Description** drop-down list option in the **Ethernet Card** section. An IP address assigned to a network interface card selected displays in the lower left corner of the **Ethernet Card** section.
- Owing to the characteristics of Ethernet, a computer can communicate with all devices on a network. User can create the IP addresses of devices connected to this driver in the **IP Address** section.
  - Some devices support the **Search** function. After user click **Search** to search for IP addresses, results display in the **IP Address** section.
  - After user click **Add** to add new IP address to the list of IP addresses in the **IP Address** section, they can type related information in the **IP Address**, the **Port** number, the **Label** and the **Type**.
    1. User can type the IP address of a device connected in **IP Address** column
    2. User can type the communication port number specified. If it is not necessary to specify a communication port number, use the default communication port number 502 in the **Port** column.
    3. User can type a comment in **Label** column.
    4. User cannot type a device in **Type** column. After user click **Search**, a device which is found displays in this column.
- User can delete a particular device by selecting it and clicking **Delete** or pressing **Delete** key in the keyboard.

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

5.3.4.7 Simulator Setting

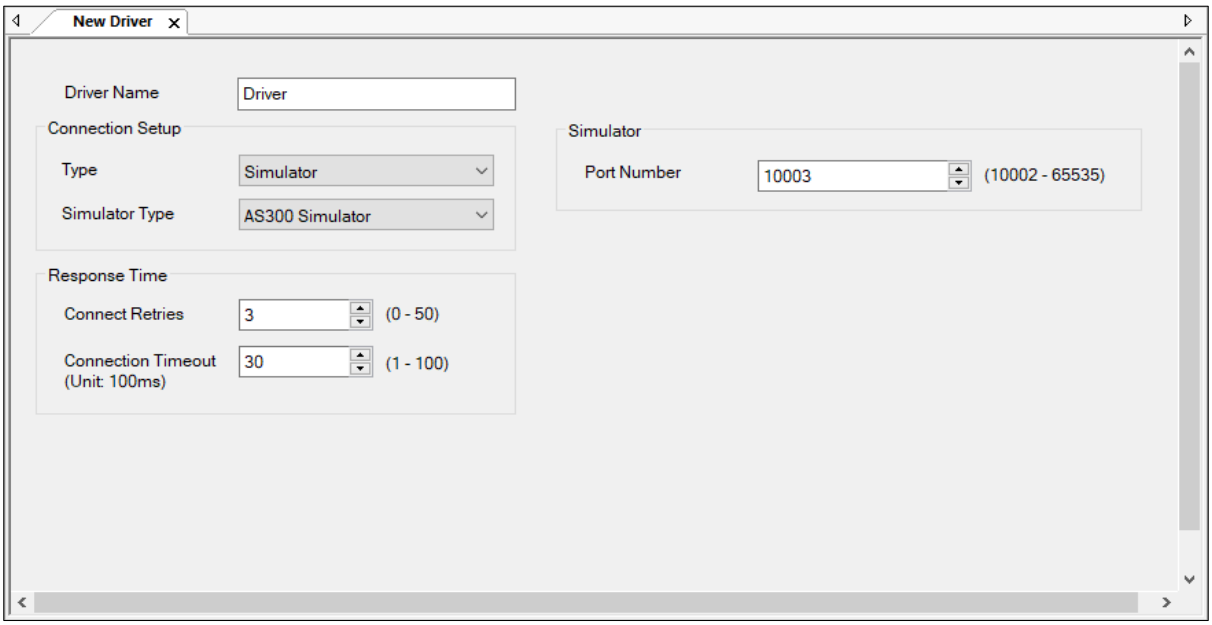


Figure 5 - 31: Driver setup - Simulator

- User can type a driver name in the **Driver Name** box. Special characters cannot be used except _
- Select **Simulator** in the **Type** drop-down list in the **Connection Setup** section
- Select **AS200 Simulator** or **AS300 Simulator** in the **Simulator Type** drop-down list
- User can type a communication port number specified. If it is not necessary to specify a communication port number, use the default communication port number

***NOTE**: User can create several Simulator drivers. But there are limitations in run mode. Maximum of following simulators can run at same time.*

- *4 AS300 simulators*
- *4 AS200 simulators*
- *4 AH simulators*

- *16 DVP simulators*

  (Total 28 simulators can run at same time.)

When the maximum limit for creating drivers of particular type is reached, COMMGR will display the pop-up as shown in the following figure.

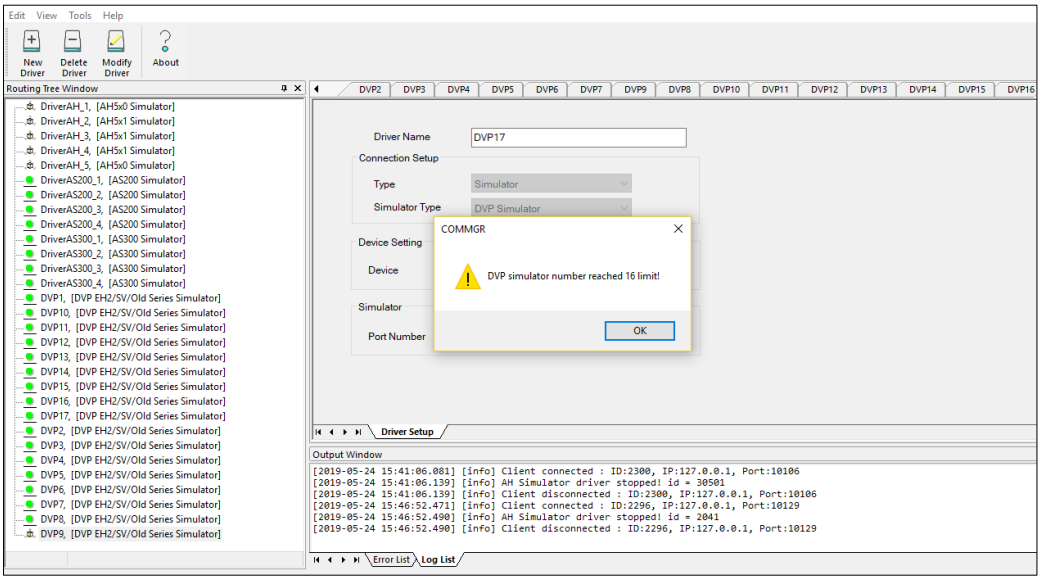


Figure 5 - 32: Maximum Limit of Simulators

User have to assign different communication port numbers to different AS PLC Simulators.

5.3.4.8 Support DIACloud function, create DIACom driver

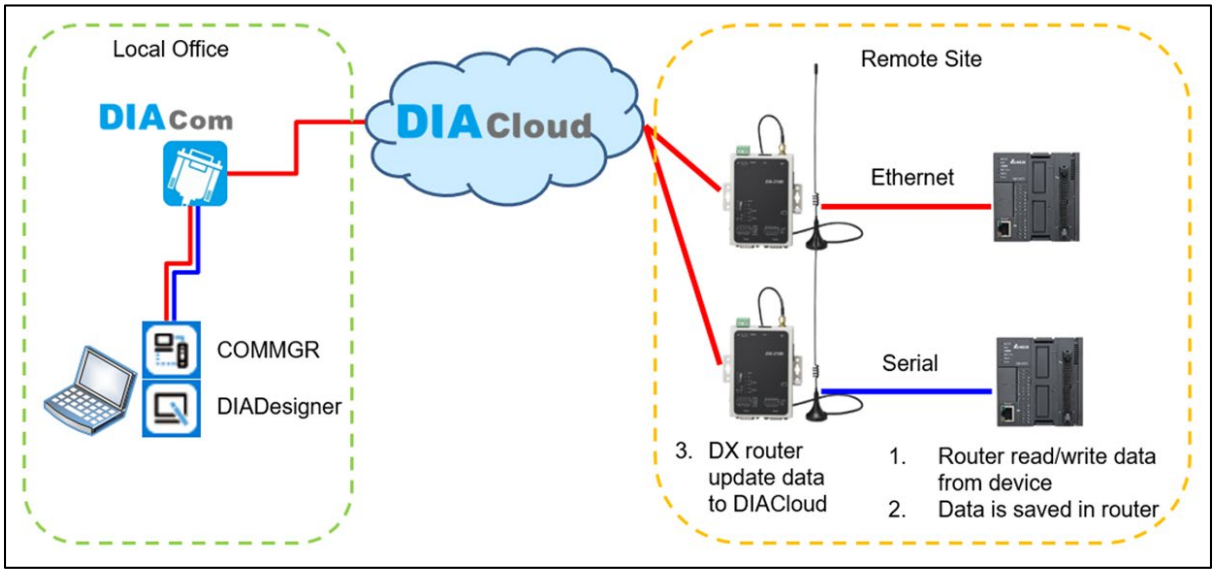


Figure 5 - 33: DIACloud Diagram

Prerequisites:

When you want to create a DIACom type driver, please confirm the following items first:

1. The DX Ethernet cloud router equipment environment has been established.
2. A DIACom account has been established and the DX device has been bound to DIACloud.

For detailed connection settings and precautions for use, please refer to the relevant user manual of Delta DX.

Establish DIACom (Ethernet) driver

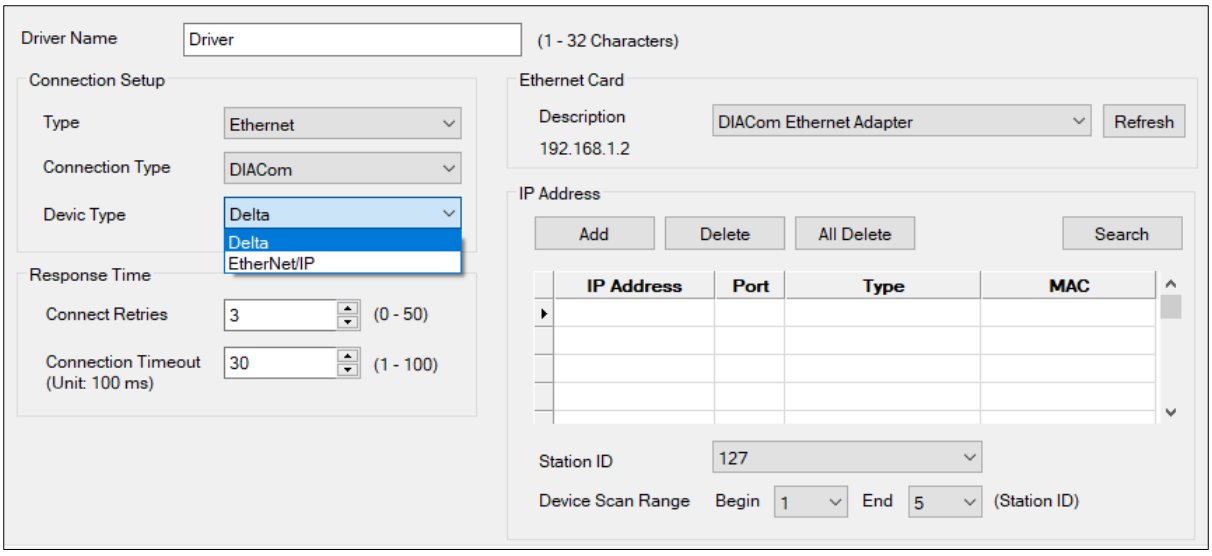


Figure 5 - 34: New driver – DIACom (Ethernet)

- The user can enter the identification name in the driver name field, except for the underline, no special symbols can be used for the entered content.
- Select Ethernet in the category selection drop-down menu of the communication category setting.
- Select DIACom from the drop-down list of connection type in the communication type setting.
- Select Delta or EtherNet/IP from the drop-down list of device type for communication type setting.
    - Delta: Search for Delta devices.
    - EtherNet/IP: Search for EtherNet/IP devices.
- In the description field of the Ethernet card area, select the virtual network card (DIACom Ethernet Adapter) created by DIACom from the drop-down menu. The number in the lower left corner of the Ethernet card area is the current Ethernet card selected The IP address to be set. If you have not connected to the virtual network card when you turn it on, you can click the update button after the connection is completed to select the correct virtual network card through the drop-down menu.
- According to the characteristics of the Ethernet itself, the computer can communicate with all devices connected to DIACloud, so the user can pre-create the device IP that may communicate through this driver in the area of

IP address setting.

- Some devices support the search function. After pressing search, the search will start and the result will be displayed in the IP address setting area. If you have not used DIACom to open the channel, a prompt message will appear and you can open DIACom directly to perform the action of establishing the channel.
- Press the add button to add a group of IP addresses in the IP address setting area, and then you can enter relevant information in the IP address, port number, and model.
  1. Enter the IP address of the connected device in the IP address.
  2. Enter the designated port number in the port number, but if no special designation is required, please use the default port number 502.
  3. The model field cannot be edited, but is used to display the searched device type when using the search function.

- After selecting a group of IPs, press the delete button or directly press Delete on the keyboard to delete the group of IPs.
- Performing a DIACom search through COMMGR will use network traffic.

***NOTE**: When the Add Driver window is opened, the system will retrieve the network card information once, but will not keep real-time updates. Therefore, if the virtual network card is added to the system or changes the network card settings after opening the Add Driver window, the drop-down menu described will not be updated. You can update the drop-down menu by clicking the update button in the Ethernet card area.*

Establish DIACom (Serial) driver:

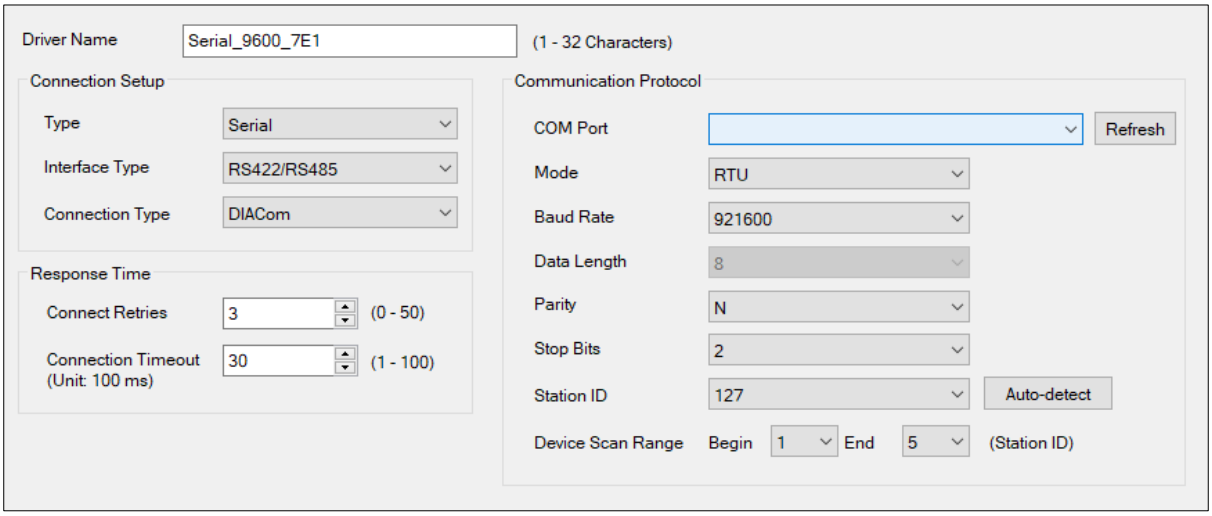


Figure 5 - 35: New driver-DIACom (Serial)

- The user can enter the identification name in the driver name field, except for the underline, no special symbols can be used for the entered content.
- Select Serial in the category selection drop-down menu of the communication category setting.
- Select RS232 or RS422/RS485 from the drop-down menu of street type in the communication type setting.
- Select DIACom from the drop-down list of connection type in the communication type setting.
- In the drop-down menu of the communication protocol, select the port that you want to establish a connection channel. When a virtual communication port has not been established, it will automatically ask whether to open DIACom to create a virtual communication port, then you can open DIACom to create a virtual communication port . After the establishment is completed, click the update in the communication protocol, and the complete device name (ELTIMA Virtual Serial Port) and port number will be displayed in the drop-down menu of the communication port.
- The mode can be ASCII or RTU.
- The computer port and device port used to perform data exchange must have the same communication settings.
- Using COMMGR to perform a DIACom search will use network traffic.

5.3.5 Driver Start/Stop

As long as the device and the computer is connected, COMMGR will start executing automatically; there is no need to do any extra setup. Only **simulator drivers** need to be set the start or stop mode.

Example:

Let us take the example of **AS300 Simulator Driver**.

5.3.5.1 Driver Start

If user want to start AS300 Simulator Driver, they can select the respective driver in the COMMGR window, and right-click on the driver to open its context menu and select **Start**. The description of the Simulator driver status to **Start** is shown in the following figure.

Figure 5 - 36: Driver - Start

5.3.5.2 Driver Stop

If user want to stop a running Simulator driver in the COMMGR window, they can select the driver and right-click on the driver to open its context menu and click **Stop**. The description of the Simulator driver status to **Stop** is shown in the following figure.

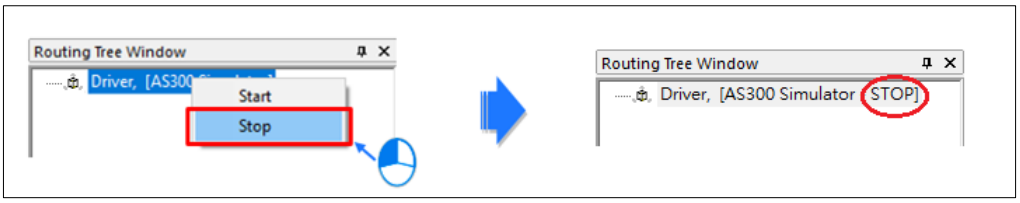


Figure 5 - 37: Driver - Stop

5.3.6 Driver Search

When the device is connected to the computer normally, you can search for connected devices in COMMGR.

5.3.6.1 Execute Search

If the created driver is normally connected to the computer, click on the front of the driver in the Routing Tree window to start searching for the device connected to this driver. The scope of the search can be modified to the settings in the driver, as shown in the figure below, modify this setting will take effect after saving the driver.

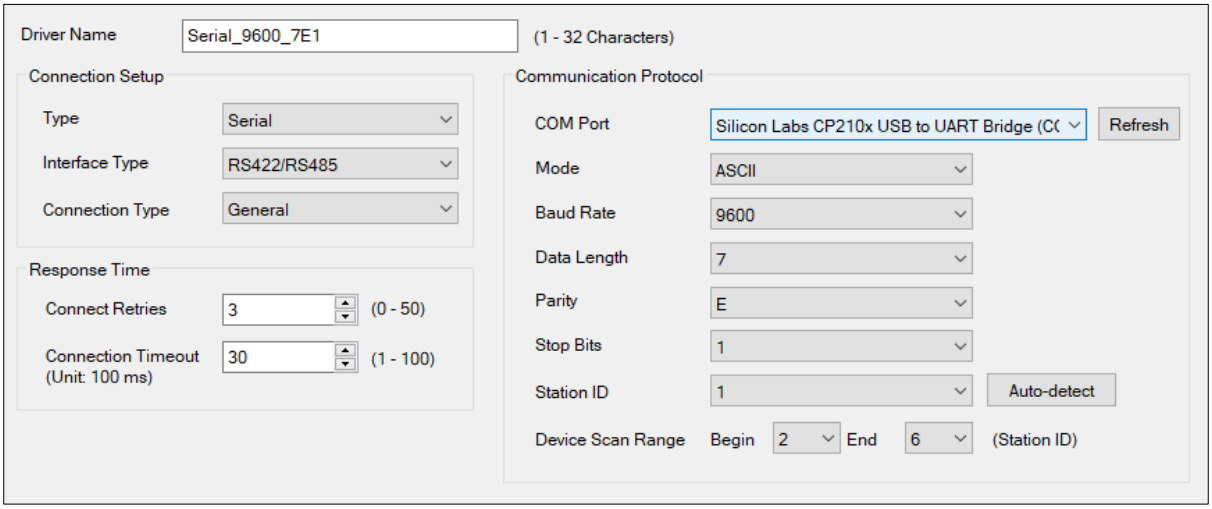


Figure 5 - 38: Modify device search range

For devices that have been searched, when the device is offline, the offline device will remain in the list when you search again, and an error symbol will be displayed in front of the device , indicating that the device is not connected normally; you can check the connection status and run again Search, or right-click on the device in the Routing Tree to delete this device, as shown in the figure below.

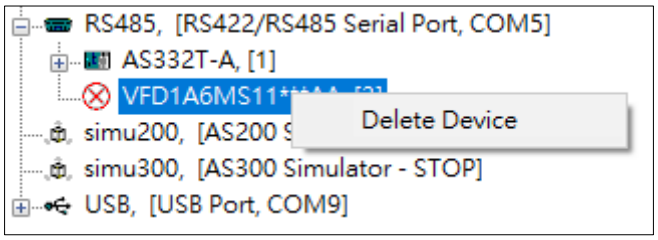


Figure 5 - 39: Delete Device

COMMGR can perform penetrating search. When the PLC supports the penetrating function (ex: AS332T), you can expand the node through the PLC to continue the search. Take AS332T as an example, as shown in the figure below, you can expand RS485 (COM1, COM2, Function Card), Ethernet or CANopen nodes, click on the front of each node to search for connected devices.

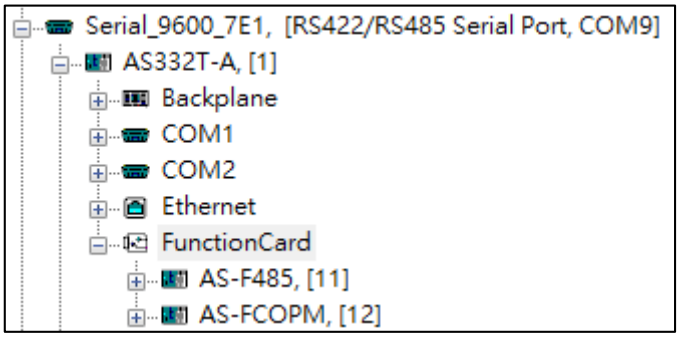


Figure 5 - 40: AS332T expands each node to perform penetration search

The result of the penetration search will be shown in the following figure. Note that if the driver is Serial Port, the penetration search of the same Serial Port can no longer be performed; when the driver is Ethernet, the Ethernet penetration search cannot be performed anymore. Search through.

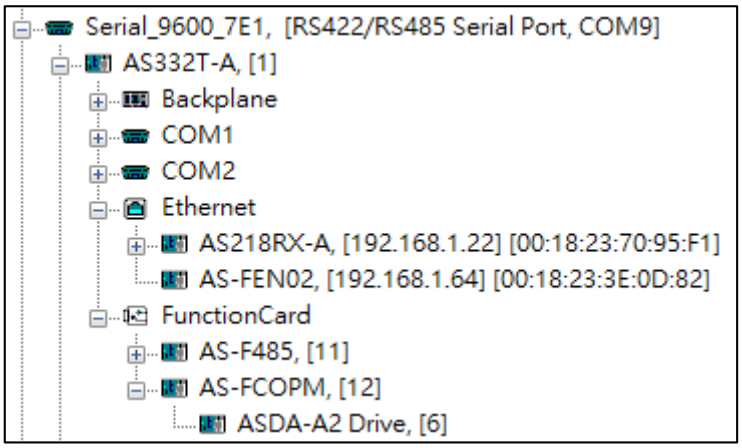


Figure 5 - 41: AS332T penetration search results

For the search range, right-click the device scan range on each node to open the device search range setting window, as shown in the figure below, after modification, click OK to take effect.

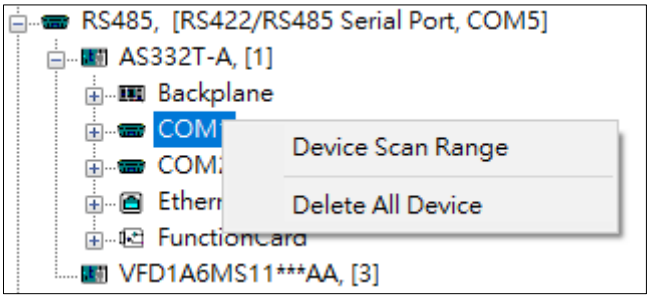


Figure 5 - 42: Right click on the node to set the scan range of the penetration search

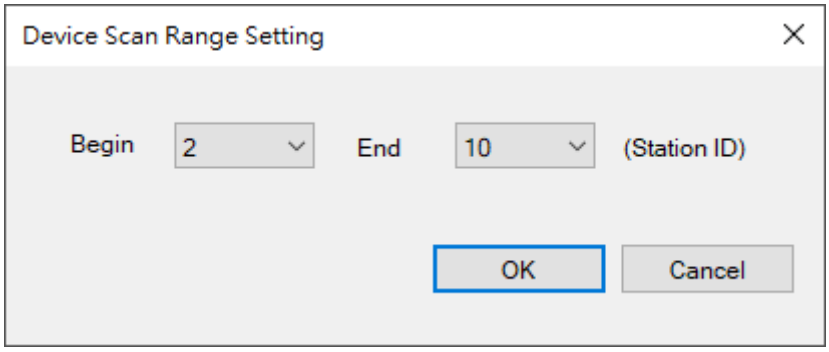


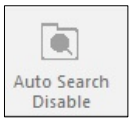
Figure 5 - 43: Device search range setting window

5.3.6.2 Automatic Search

When you need to continuously search for the status of the online device, you can click

in the toolbar to open the automatic search. After opening, the icon button will be grayed out and you can no longer click it. Then click on the front of the driver in the Routing Tree window to continue searching The device status of this driver.

When a new device is connected, the new device will be displayed in the Routing Tree window; when the device is not connected normally, an error symbol will be displayed on the screen immediately. At this time, you can check the connection status and perform the search again, or you can Right-click on the device in the Routing Tree and select delete this device.

When you want to stop the continuous search, click 

in the toolbar to turn off the automatic search function, and it will stop the continuous search for the device status of all drivers.

Follow these steps to modify the driver created previously.

1. Select the driver to be modified
2. Click **Edit > Modify Driver** in **Menu** bar or

   Click

icon in the **Toolbar** or

   Right-click on the driver to open its context menu and select **Modify Driver** option

   Result: The **Driver Setup** window displays.

3. User can modify the parameters in the **Driver Setup** window according to their requirements as mentioned in *5.3.4 Driver Setting*.

4. After modification, click

icon in the tool bar to save driver setting.

5.3.7 Bypass Setting

User can connect to Delta field devices through a Delta PLC to PC using the bypass setting function in DIADesigner.User can then scan the **Routing Tree Window** by opening the driver as shown in the following figure.

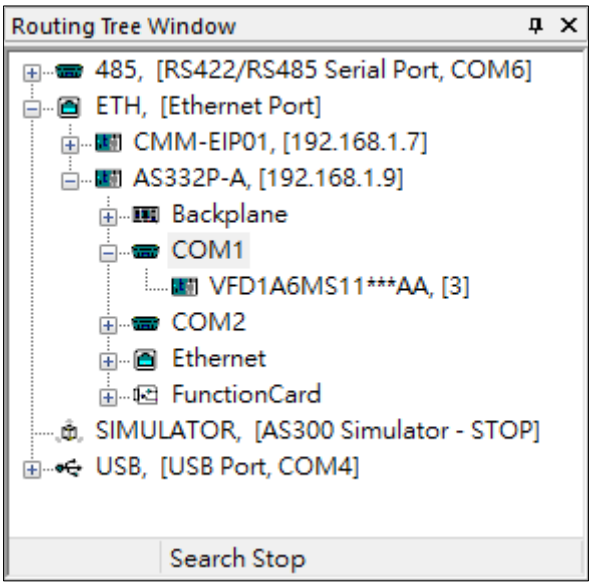


Figure 5 - 44: Routing Tree Window

Follow these steps to perform bypass setting in DIADesigner:

1. Create a driver in COMMGR
2. User can open DIADesigner where the devices to be configured are created
3. Locate the device in DIADesigner and open **Communication Setting** of the device.
4. User can select COMMGR option in the Communication Interface drop-down

list in Communication Setting window, select the Driver Name and Station Number.

5. Click **Routing Setting.**

   Result: The **Routing Setting** window displays as shown in the following figure.

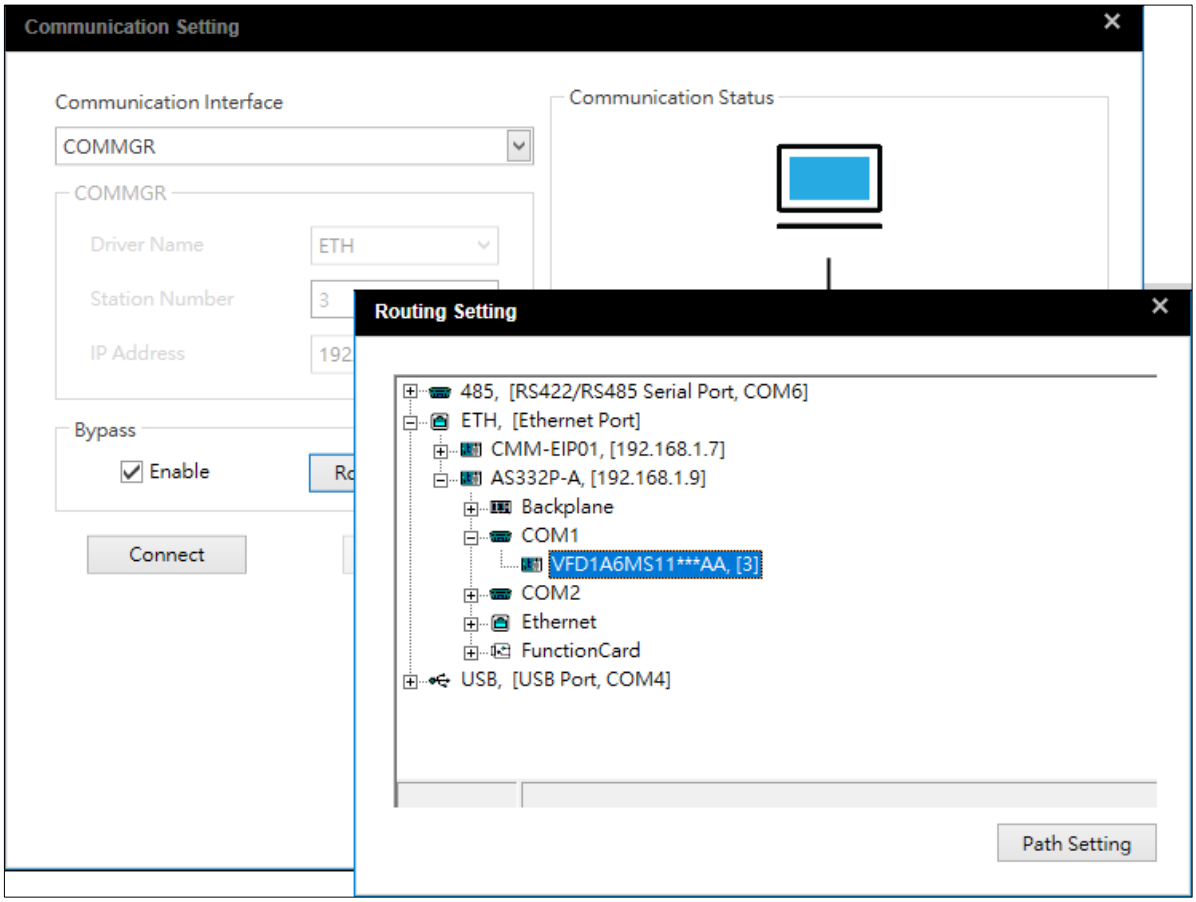


Figure 5 - 45: Routing Setting

6. In the Routing Setting window, click the terminal device to be connected then click **Path Setting.**
7. Click on **Enable** check box.
8. Click on **Connect** to connect the device and configure it.
9. Click on **Disconnect** to disconnect the device.

5.3.8 Communication Setting of Inverter with Built-in PLC

Communication of inverters can be set via COMMGR or Serial Port.

1. *Connection via COMMGR*

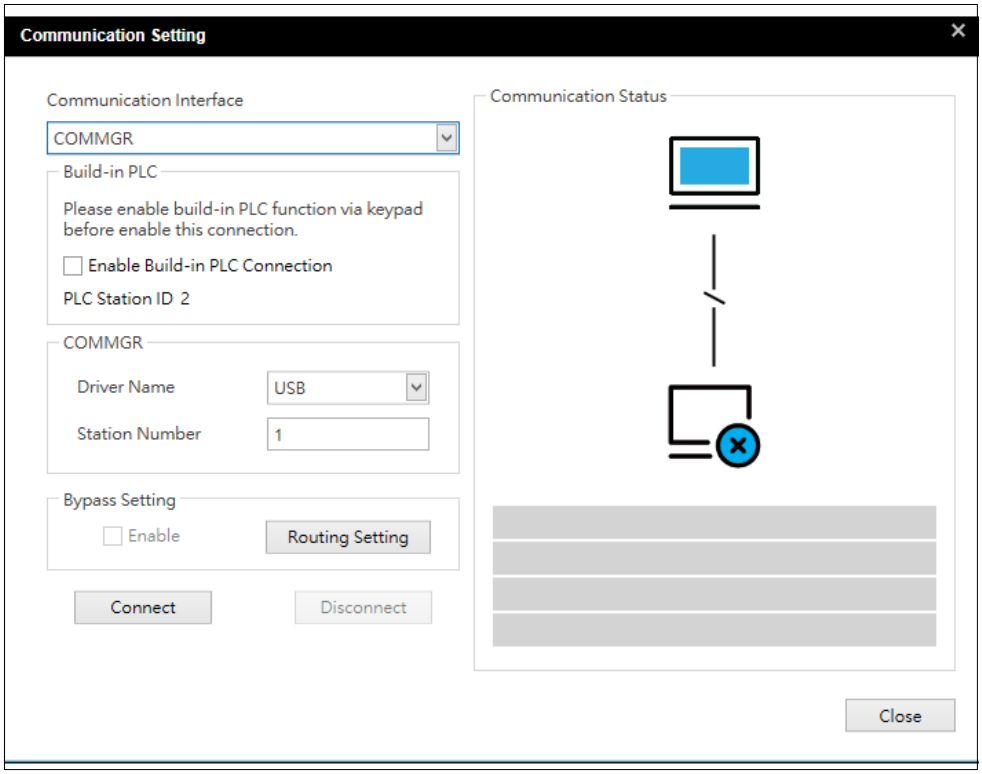


Figure 5 - 46: Communication Setting - Inverter Built-in PLC

Please follow the steps below to set up communication:

1. Create a driver in COMMGR.
2. Open DIADesigner and add a new project. After that, add a new inverter device.
3. Click and expand the inverter device and select **Communication Setting**.
4. In the **Communication Setting** window, select **COMMGR** in the drop-down list of **Communication Interface**. Then, select **Driver Name** and **Station Number**.
5. Please enable the built-in PLC functions on the operation panel before

enabling the built-in PLC in the Inverter.

6. Check the box of **Enable build-in PLC Connection**.
7. Click **Connect**.

   Result: The communication status shows **Connected**. Communication setting is successful as shown in the following figure.

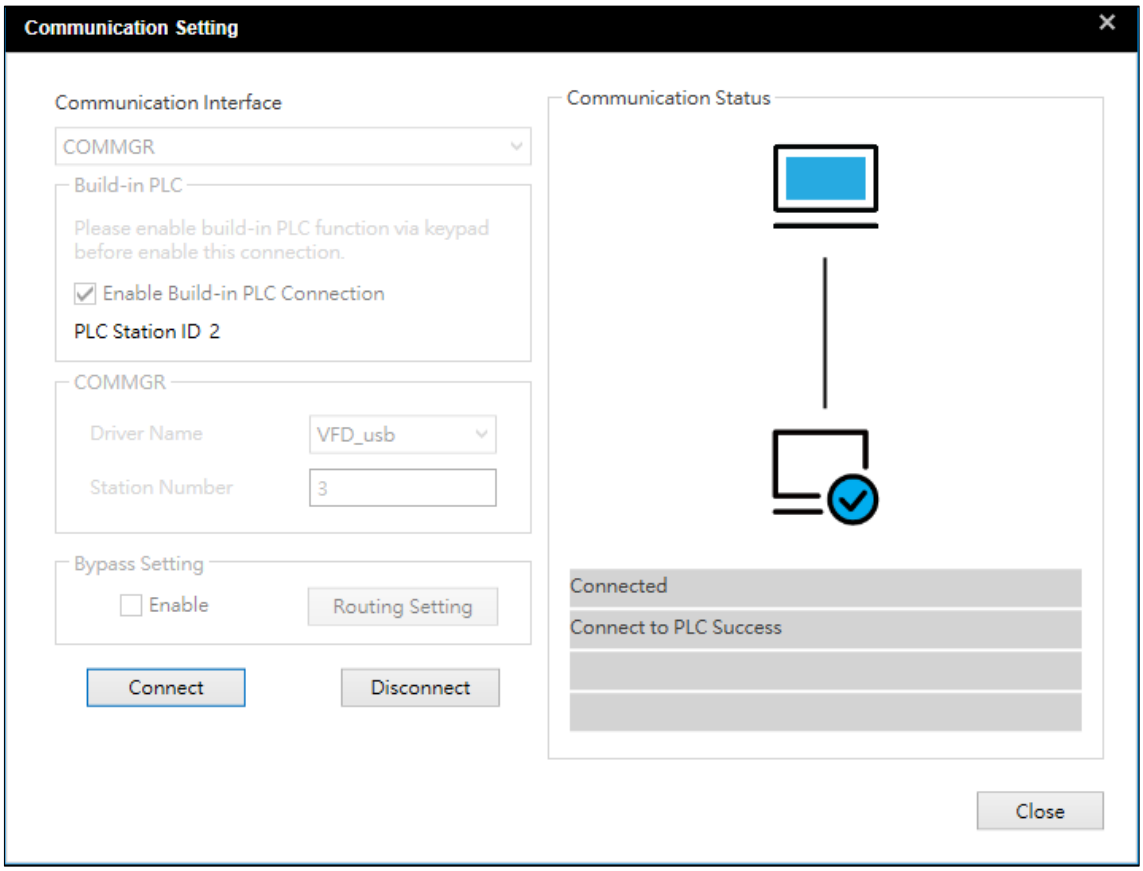


Figure 5 - 47: Communication Setting – COMMGR Interface

Connection via Serial Port

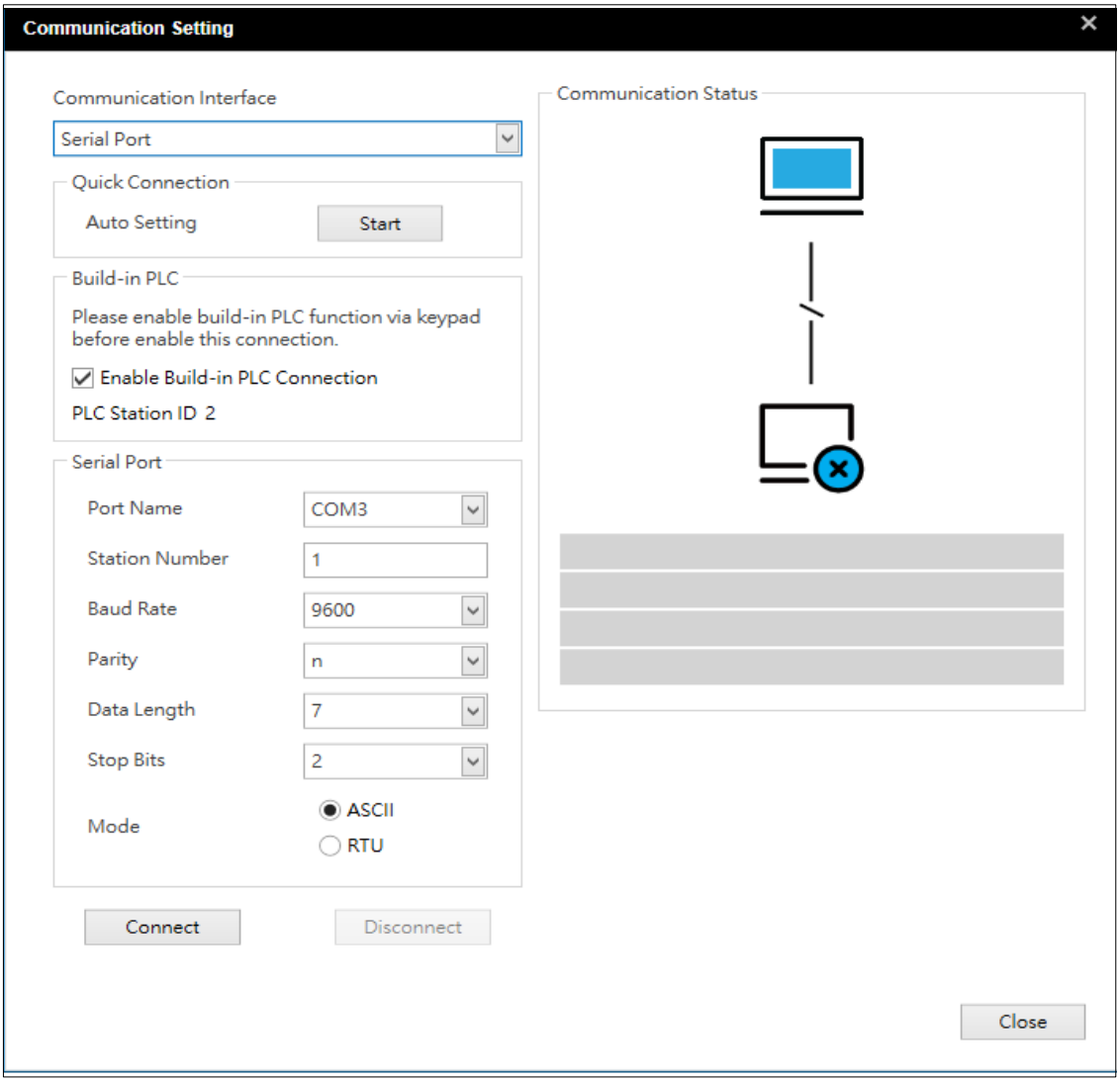


Figure 5 - 48: Communication Setting – Serial Port Interface

Follow the steps below to set up communication:

1. Open DIADesigner and add a new project. Then, add a new inverter.
2. Click and expand the inverter and select **Communication Setting**.
3. In the **Communication Setting** window, select **Serial Port** in the drop-down list of **Communication Interface**. Select the right serial port, or click **Start** of **Auto Setting** in **Quick Connection**.
4. Please enable the built-in PLC functions on the operation panel before enabling inverter with built-in PLC.

5. Check the box of **Enable built-in PLC Connection**.
6. Click **Connect**.

   Result: The communication status shows **Connected**. Communication setting is successful as shown in the following figure.

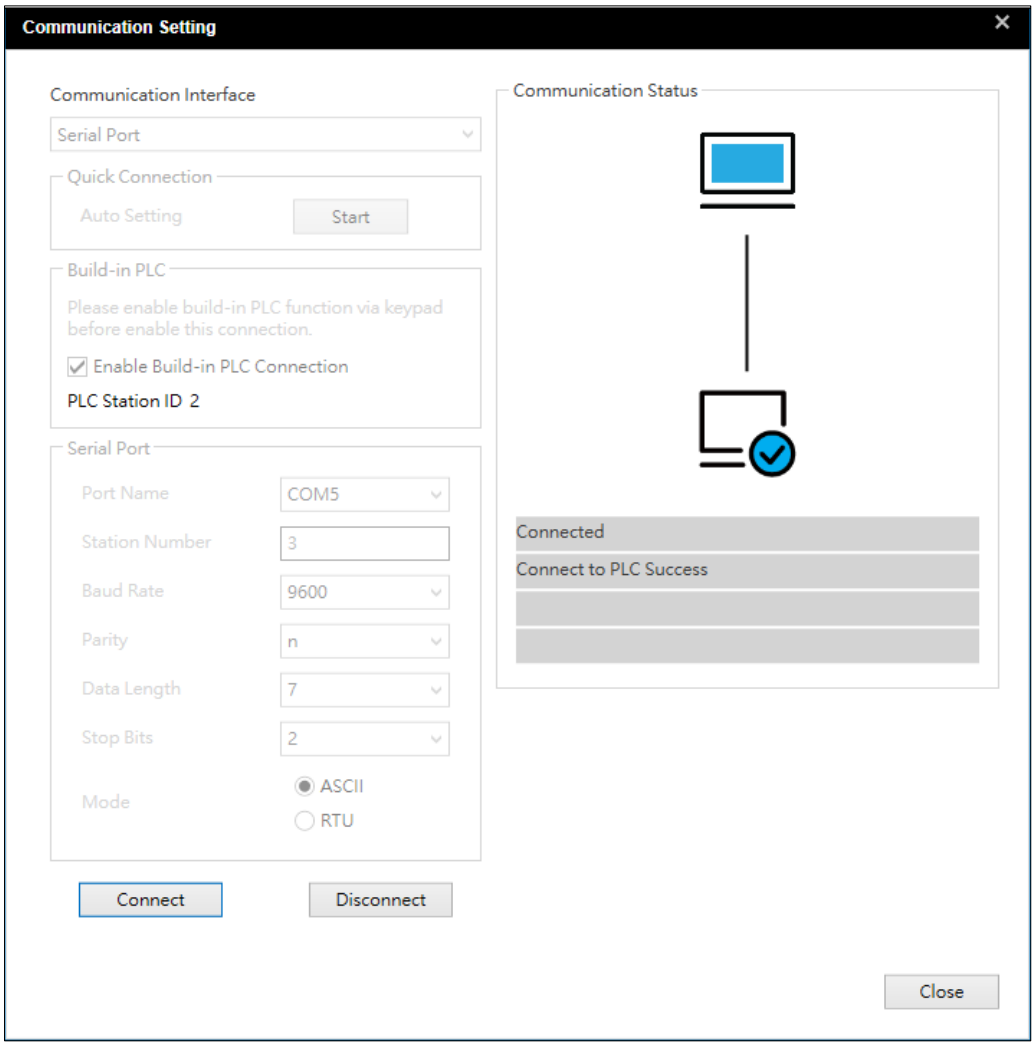


Figure 5 - 49: Communication Setting – Serial Port Interface

Chapter 6: Setting Device (Hardware Configuration)

6.1 Hardware Configuration

User can perform functions like configuration of modules, racks, parameter settings for modules, download/upload - hardware parameters, simple on-line detection and diagnosis with the help of Hardware Configuration.

User can locate **Hardware Configuration** in **Controller** > **Hardware Configuration** as shown in the following figure.

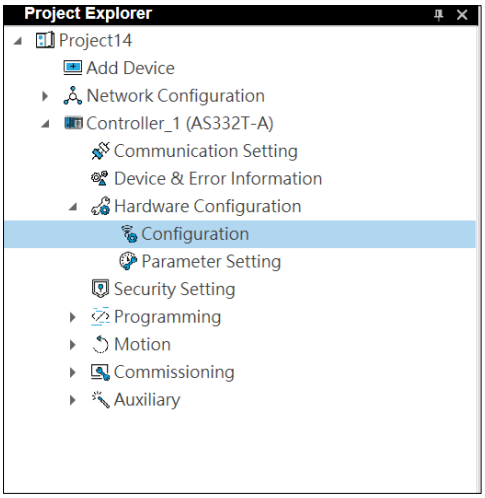


Figure 6 - 1: Project Explorer - Hardware Configuration

AS series PLC:

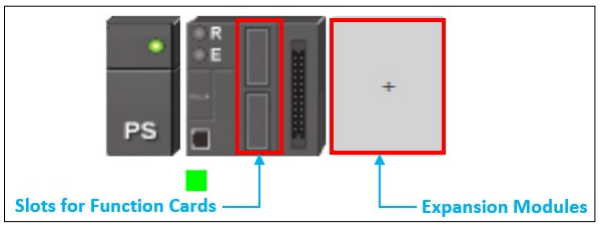


Figure 6 - 2: AS PLC - Function Cards and Expansion Modules

The AS series PLC has expansion modules with non-backplane design placed on the right and slots for function cards. AS300 series CPU and SCM modules have

function card slots and AS200 series CPU have in-built communication port for CANopen.

The **Hardware Configuration** includes **Configuration** and **Parameter Setting**.

Configuration provides a graphical user interface to plan the hardware configuration.

Parameter Setting provides the option to set the parameter values in the device or the module. It also provides the parameter basic information for the user to understand the parameter characteristic.

6.1.1 Creating Controller Configuration

Follow these steps to open Configuration:

1. Expand the **Controller** > **Hardware Configuration** in **Project Explorer**.
2. Double-click on **Configuration** in **Project Explorer**.

   Result: The **Hardware Configuration** window displays as shown in the following figure. Upper pane displays hardware view and bottom pane displays **Module Setting Table** as shown in the following figure.

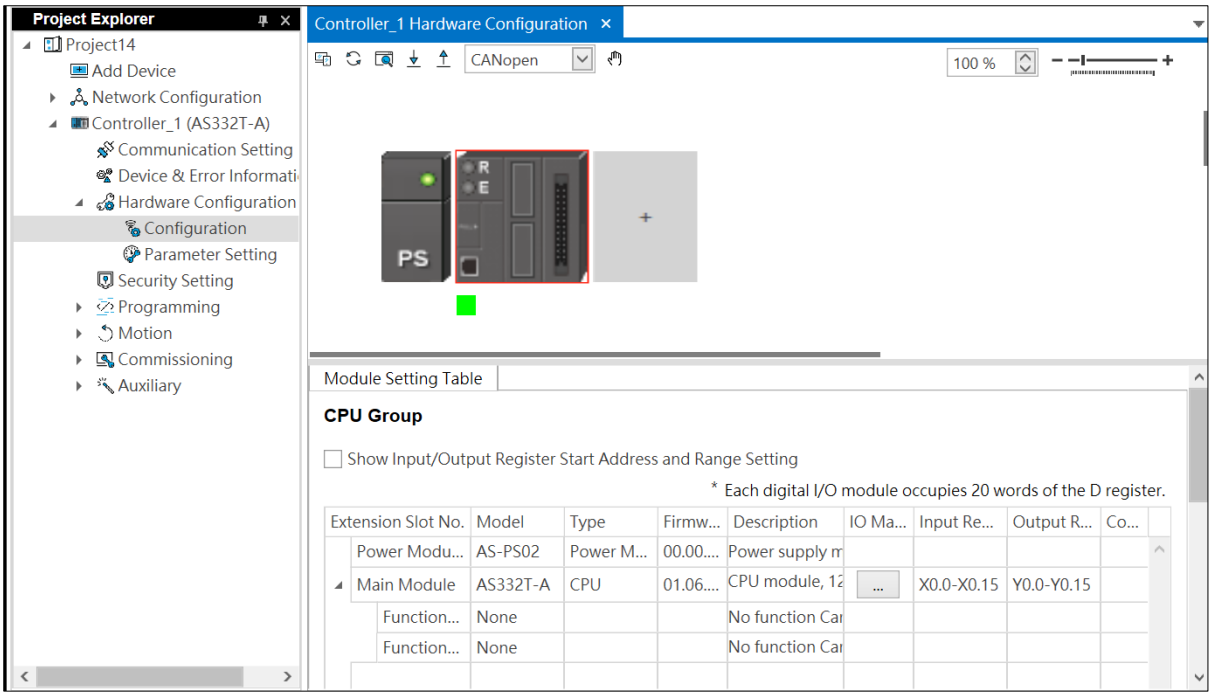


Figure 6 - 3: Hardware Configuration Window

The **Hardware Configuration Toolbar** is shown in the following figure.

Figure 6 - 4: Hardware Configuration - Toolbar

The functions in **Hardware Configuration Toolbar** are described in the following table:

Function	Description
	Click to display module information.
	Click to reset register address.
	Click to perform I/O scan.
	Click to download.

Function	Description
	Click to upload.
	Select remote IO network from the drop-down list.
	Enter or select value to zoom-in or zoom-out.
	Click and drag to zoom-in and zoom-out.
	Switch to hand mode

These functions are described in detail in their respective chapter.

6.1.1.1 Adding Module

The AS series PLC has a non-backplane design with expansion module placed only on the right side (+ position) and power supply module to the left side (+ position) of the CPU in the **Hardware Configuration**. User can continue to add modules to the right side of + position, while only one power supply module can be added on the left + position of the CPU.

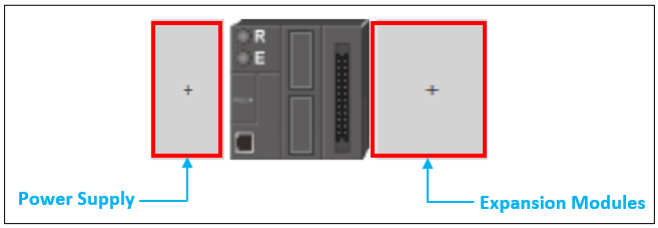


Figure 6 - 5: Power Supply and Expansion Modules position

Adding modules is supported in different methods:

- From **Module List** pop-up window double-click.
- From **Toolbox** drag-drop.

Method 1 (from Module List):

Follow these steps to add modules from **Module List** window:

1. Double-click on the **+** position

   Result: The **Module List** pop-ups.

   ***NOTE**: The **Module List** pop-up window displays only those modules which are supported in the particular context.*

Example:

If user double-clicks on **+** position on the left-side of the PLC CPU, the **Module List** window pop-ups with **Power Module** option as shown in the following figure.

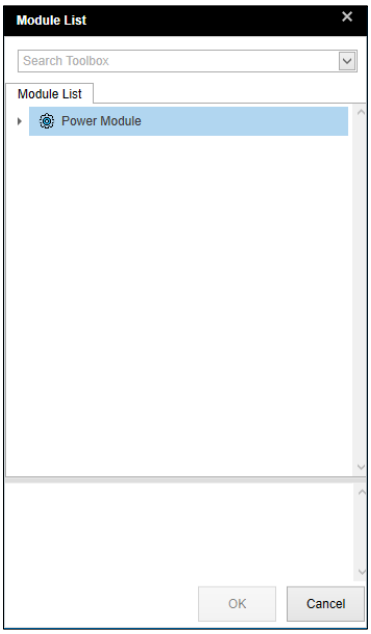


Figure 6 - 6: Module List for CPU left + position

If user double-clicks on **+** position on the right-side of the PLC CPU, the **Module List** window pop-ups with **Digital I/O Module**, **Analog I/O Module**, **Network Module** and **Motion Control Module** options as shown in the following figure.

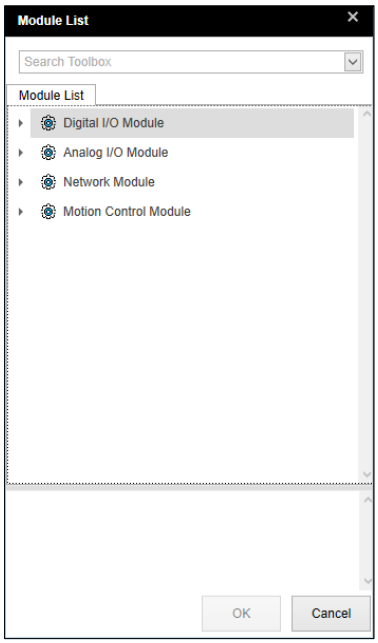


Figure 6 - 7: Module List for CPU right + position

2. Expand the required option.

   ***NOTE**: The bottom pane of the **Module List** window displays the description of the module as shown in the following figure.*

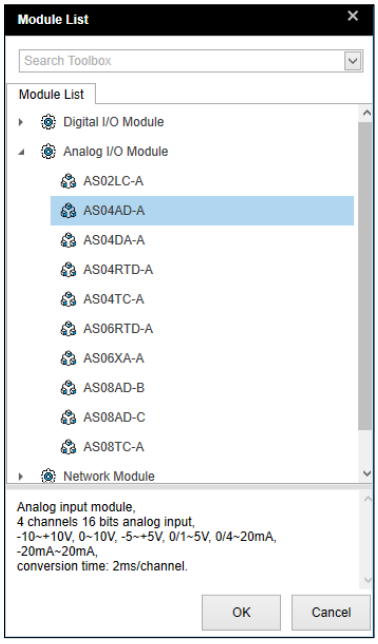


Figure 6 - 8: Module List window - description of selected module

3. Double-click on required modules to add them to the **Hardware Configuration** as shown in the following figure.

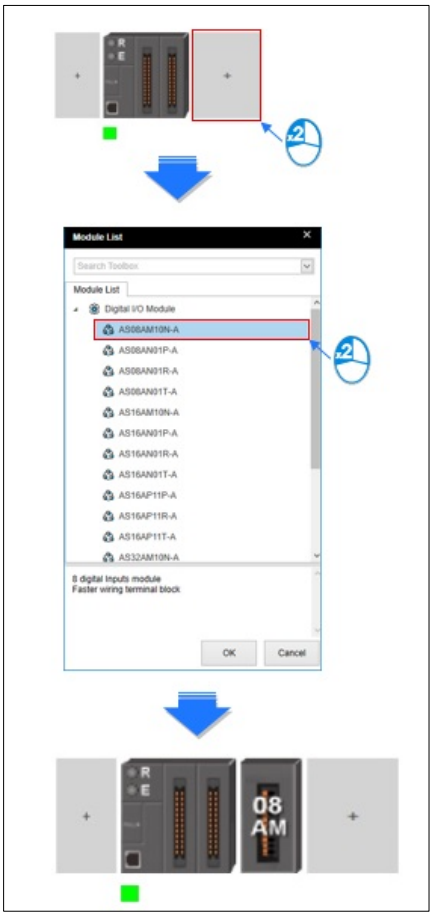


Figure 6 - 9: Adding module from Module List

Method 2 (from Toolbox):

Follow these steps to add modules from **Toolbox** window.

The **Toolbox** window displays by default on the right side of the DIADesigner.

1. Select the required module.

   ***NOTE**: The bottom pane of the **Toolbox** window displays the description of the module as shown in the following figure.*

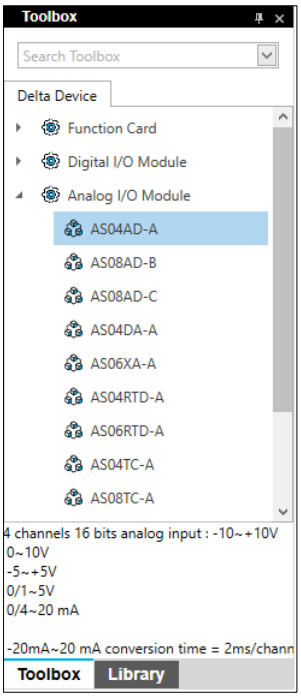


Figure 6 - 10: Toolbox with description of module selected

2. Drag the modules from **Toolbox** to the **Hardware Configuration** or Double-click the modules in **Toolbox** after clicking the **+** position.

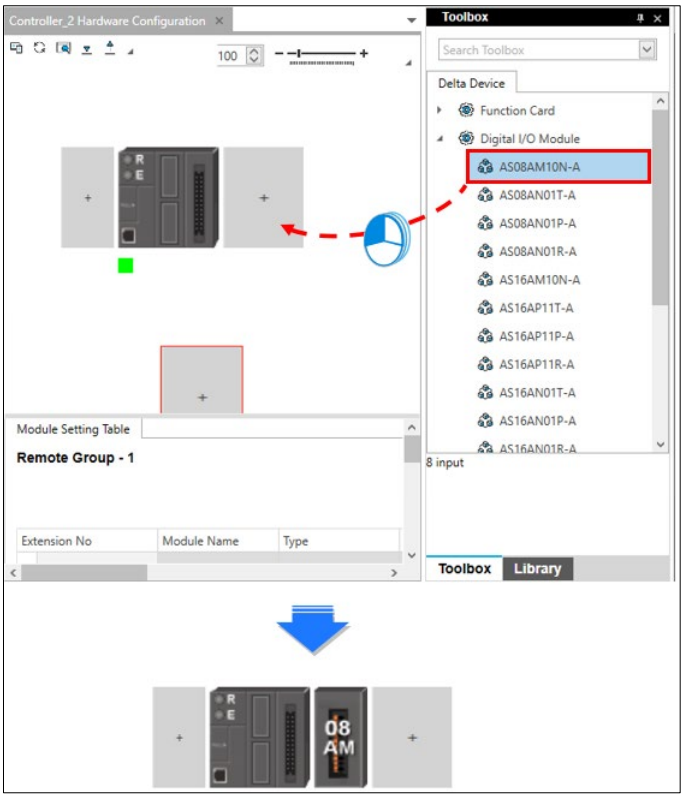


Figure 6 - 11: Adding Module to the Hardware Configuration Window

Result: Modules are added to the **Hardware Configuration** after validation for the correct positions.

6.1.1.2 Address assignment of modules

When a configuration is created in hardware view (upper pane), the **Module Setting Table** (bottom pane) automatically gets updated with address assignment. An example for address assignment in **Module Setting Table** is shown in the following figure.

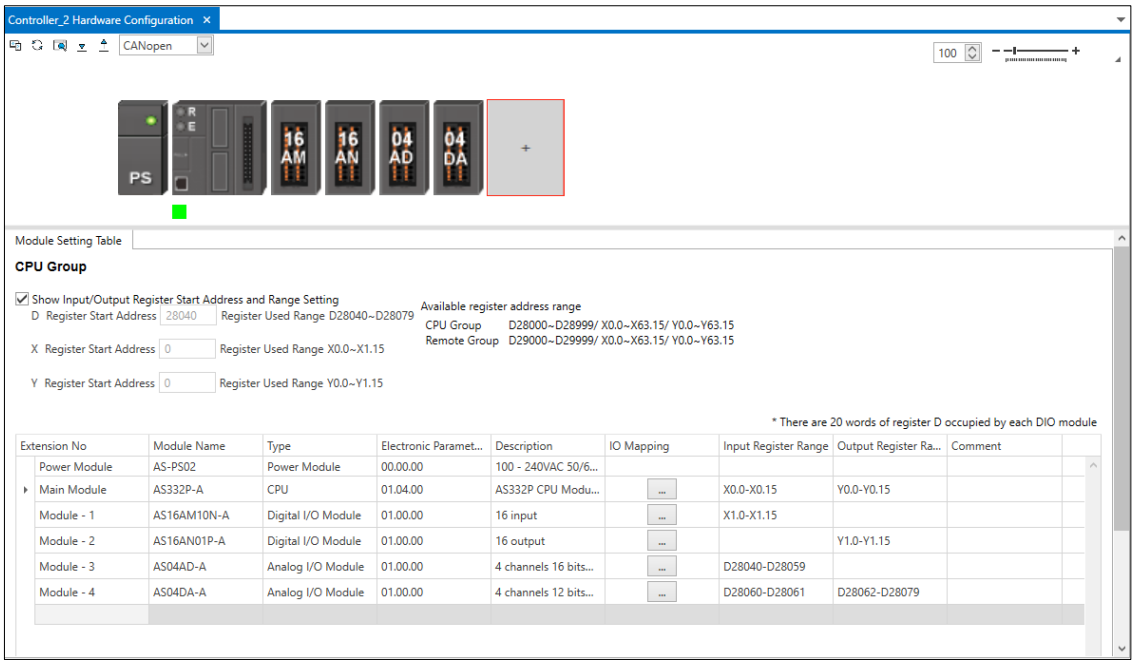


Figure 6 - 12: Hardware Configuration with Modules

The functions in the **Module Setting Table** are described in the following table:

Function	Description
Show Input/Output Register Start Address and Range Setting	Check the box to display the IO Register start address and range setting
Group name	Displays group name
Extension No	Displays extension module number
Module Name	Displays module name
Type	Displays module type
Firmware Version	Displays module firmware version
Description	Displays module description
IO Mapping	Click ... to open **IO Mapping Table** for the module
Input Register Range	Displays input register range for the module
Output Register Range	Displays output register range for the module
Comment	Enter comments

6.1.1.3 Adding RTU station

User can add RTU in the configuration as per their requirement.

Follow these steps to add a module in the **Hardware Configuration**:

1. Select RTU fieldbus type to be added as shown in the following figure.

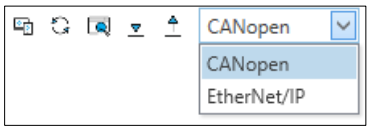


Figure 6 - 13: RTU Fieldbus type

2. Double click RTU position as shown in the following figure.

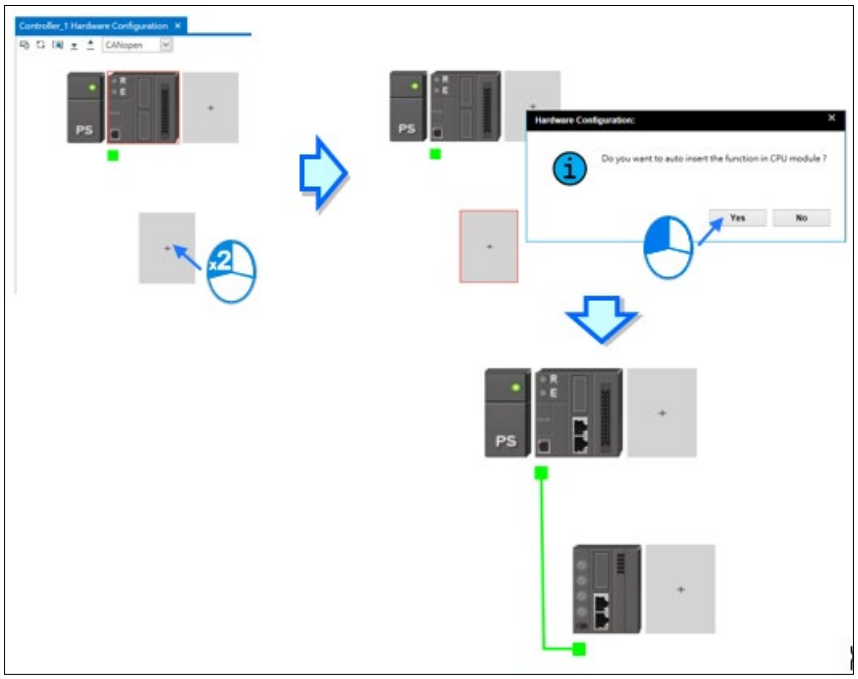


Figure 6 - 14: Adding RTU Station

Result: RTU is added, and related parameters are set automatically.

6.1.1.4 Deleting a Module

User can delete modules in the configuration as per their requirement.

Select a module to be deleted and follow any of the following methods:

- Click

icon in the **Editor** Toolbar.
- Right-click on the module to open its context menu and click **Delete**.

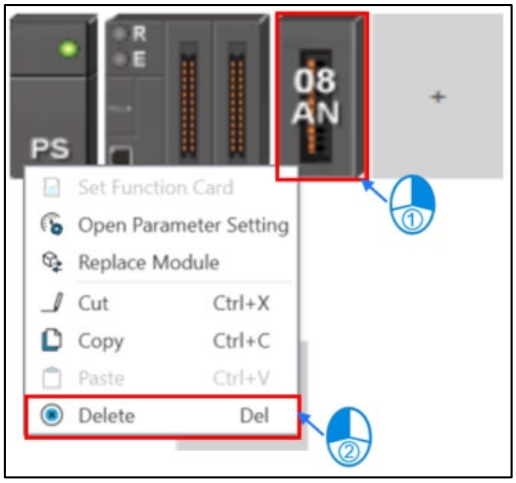


Figure 6 - 15: Delete module from context menu

- Press **Delete** key in the keyboard.

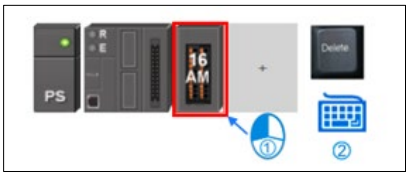


Figure 6 - 16: Delete module using Delete key

- Right-click on the module in **Module Setting Table** to open its context menu and click **Delete**.

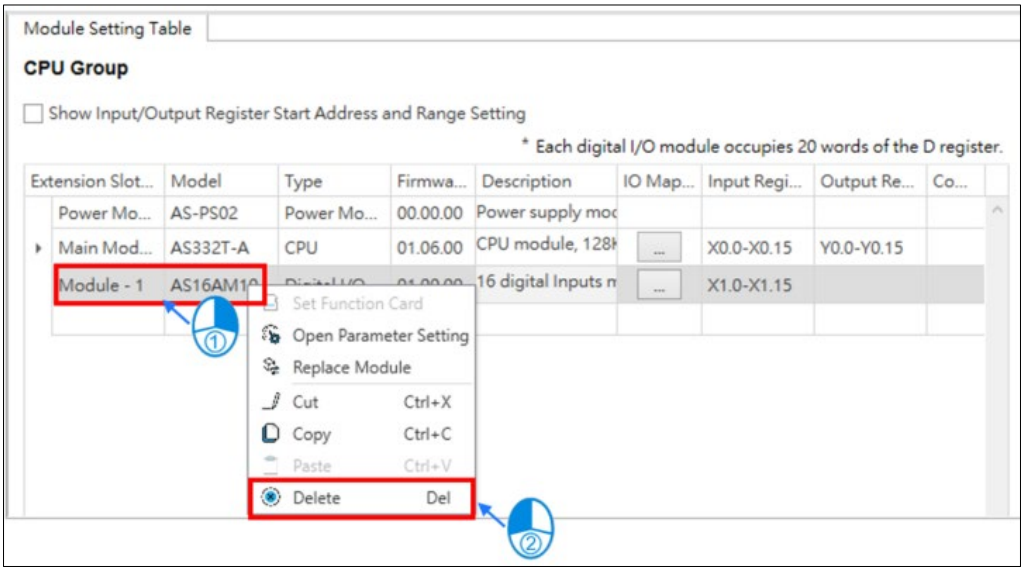


Figure 6 - 17: Delete module from Module Setting Table

- Click **Edit** > **Delete** in **Menu** bar.

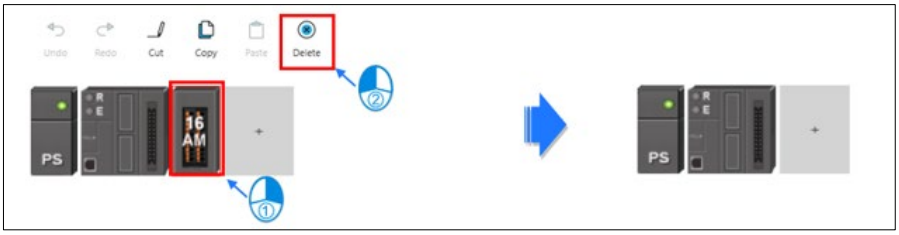


Figure 6 - 18: Delete module from Menu bar

6.1.1.5 Replace Module

User can replace modules in the configuration as per their requirement. **Replace Module** function supports replacement of selected module or all modules of similar type. Except CPU and RTU module, all other modules can be replaced.

Follow these steps to replace a module in **Hardware Configuration**:

1. Select the module to be replaced

2. Right-click on the module and click **Replace Module** in the context menu or

   Right-click on the module in **Module Setting Table** to open its context menu and click **Replace Module**.

   Result: The **Module List** window displays.

3. Expand on supported modules in the **Module List** window.

   Result: The **Module List** window expands the supported modules.

4. Double-click on the module which user intends to replace with the old one

   Result: Displays a pop-up window to confirm – Yes/No.

5. Click **Yes** to continue or **No** to cancel the operation.

   Result: Old module is replaced if **Yes** is clicked. Replacing Module cancels if **No** is clicked.

***NOTE**:*

***Replace All**: If all the modules of same type need to be replaced as another type, click **Replace All** check box in **Module List** window.*

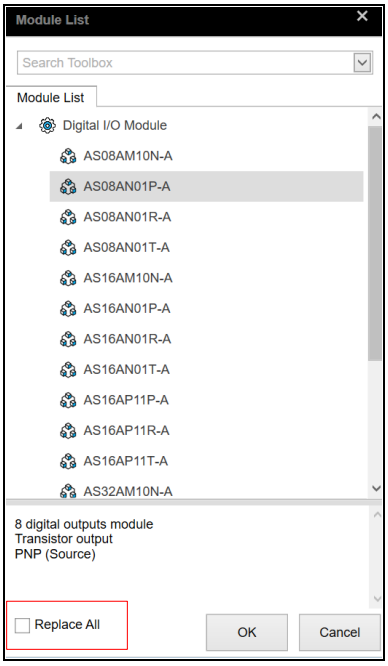


Figure 6 - 19: Replace all

Replace operation is shown in the following figure:

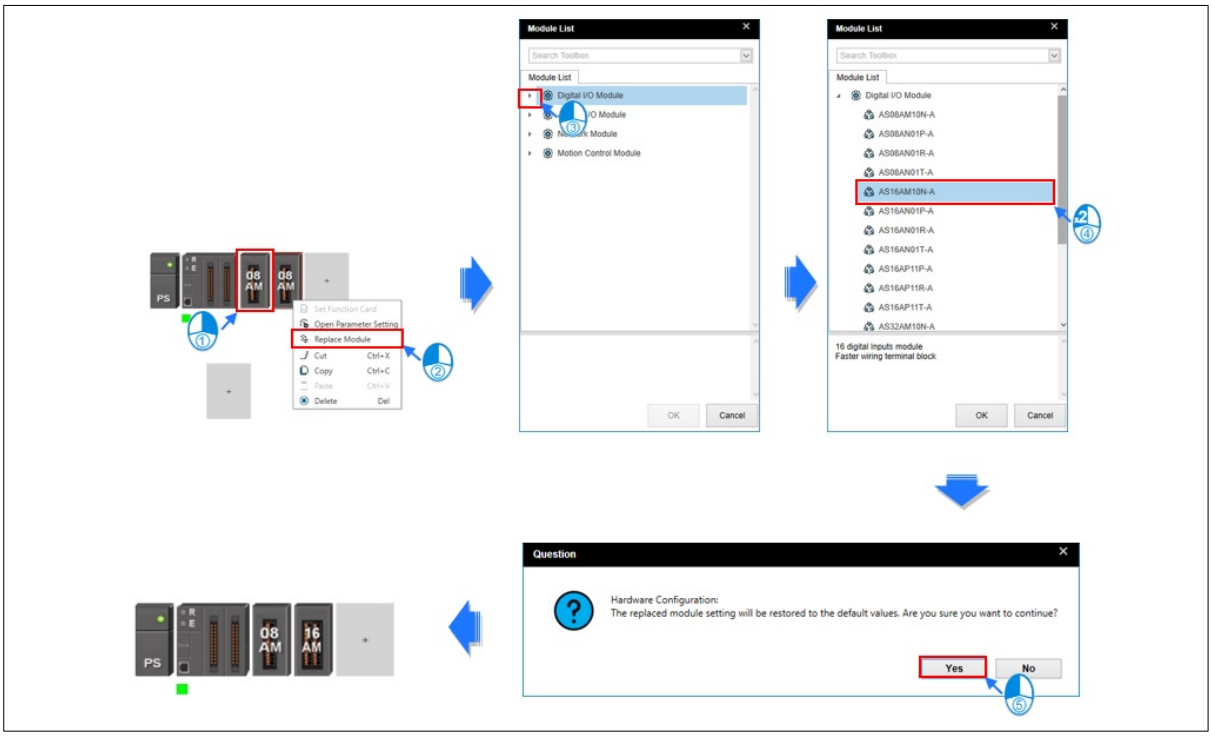


Figure 6 - 20: Replace Module in Hardware Configuration

6.1.1.6 Copy/Paste

User can Copy/Paste modules in the configuration as per their requirement.

Follow these steps to copy a module in **Hardware Configuration**:

1. Select the module to be copied.
2. Select any of these methods to copy a module:

   - Click on

 icon in the **Editor** Toolbar.
   - Right-click on the module to open its context menu and click **Copy.**
   - Press **Ctrl** + **C** keys simultaneously in the keyboard.
   - Right-click on the module in **Module Setting Table** to open its context

menu and click **Copy**.

- Click **Edit** > **Copy** in **Menu** bar.

Follow these steps to paste a module in **Hardware Configuration**:

1. Select the location to paste the module.
2. Select any of these methods to paste the module:

- Click

icon in the **Editor** Toolbar.
- Right-click on the **+** position to open its context menu and click **Paste**.
- Press **Ctrl** + **V** keys simultaneously in the keyboard or
- Right-click on the module in **Module Setting Table** to open its context menu and click **Paste**.
- Click **Edit** > **Paste** in **Menu** bar.

***NOTE**:*

- *Since AS series has a non-backplane design, when a module is pasted over another, the pasted module is inserted to the left of the existing module and the existing module shifts to the right position.*
- *Input/Output Register Range: Register allocation is done automatically to the pasted modules.*
- *Comment: Comment is copied from the source module and retained in the pasted positions.*
- *Internal Parameters: Parameters **are not copied** to the pasted module; and will reset to default.*
- *Corresponding D registers: Register allocation is done automatically to the pasted modules.*

6.1.1.7 Cut/Paste

User can Cut/Paste modules in the configuration as per their requirement. Except Power Supply and CPU, other modules can be cut.

Follow these steps to cut a module in **Hardware Configuration**:

1. Select the module to be cut.
2. Choose any of these methods to cut a module.

   - Click

 icon in the **Editor** Toolbar
   - Right-click on the module to open its context menu and click **Cut**.
   - Press **Ctrl** + **X** keys simultaneously in the keyboard
   - Right-click on the module in **Module Setting Table** to open its context menu and click **Cut**.
   - Click **Edit** > **Cut** in **Menu** bar.

Follow these steps to paste the module in **Hardware Configuration**:

1. Select the location to paste the module.
2. Choose any of these methods to paste the module.

   - Click

 icon in the **Editor** Toolbar.
   - Right-click on the **+** position to open its context menu and click **Paste**.
   - Press **Ctrl** + **V** keys simultaneously in the keyboard.
   - Right-click on the module in **Module Setting Table** to open its context menu and click **Paste**.
   - Click **Edit** > **Paste** in **Menu** bar.

***NOTE**:*

- *While executing **Cut/Paste** function, modules that are cut can only be pasted once.*
- *Comment, Internal Parameters will be transferred to the pasted module.*
- *Input/Output Register Range: Register allocation is done automatically to the pasted module location.*
- *Corresponding D Registers: Register allocation is done automatically to the pasted module location.*

6.1.1.8 IO Mapping

For installed modules and CPU, user can assign a variable for every channel.

Follow these steps to proceed with **IO Mapping**:

1. Click the IO Mapping button in **Module Setting Table**.
2. Select a variable table and variable in it from drop down list.

   or

   Select a variable table and enter a new variable name in **variable** column.

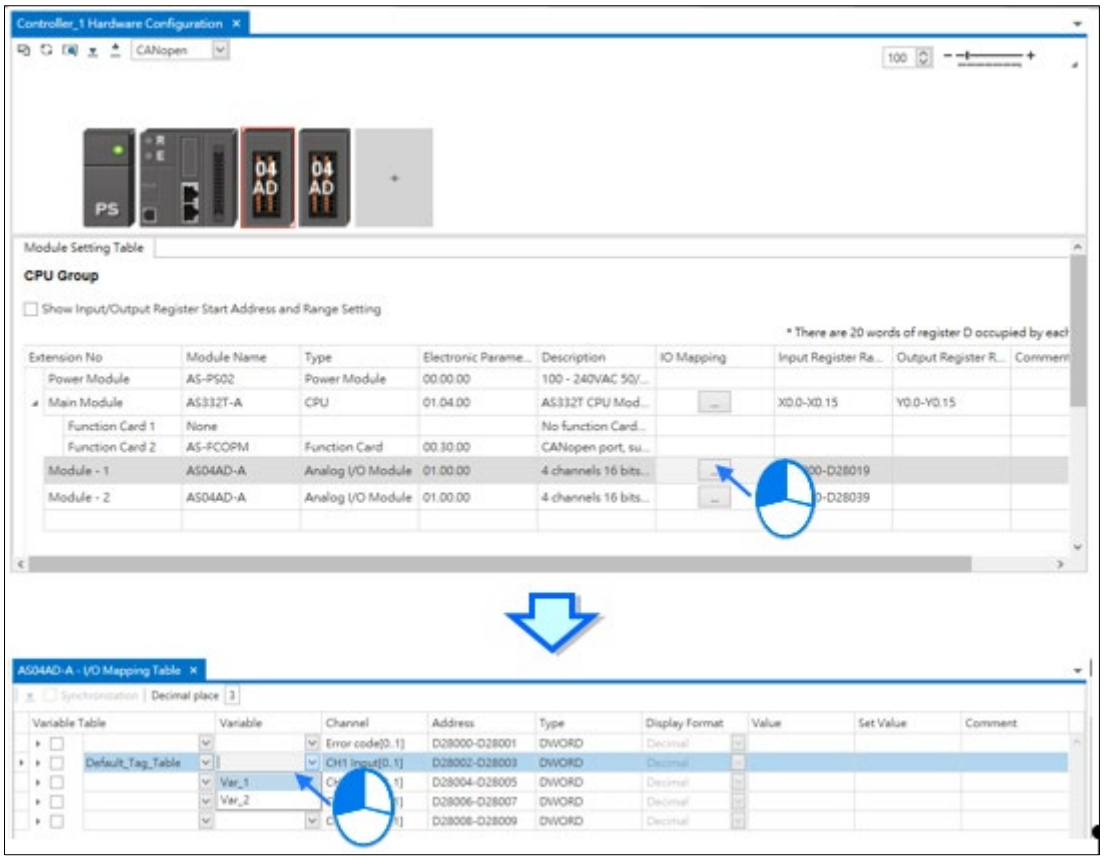


Figure 6 - 21: IO Mapping

Result: Register **Address** is set to specified variable **Name** in the **Default Tag Table** (Global Variable Table) as shown in the following figure.

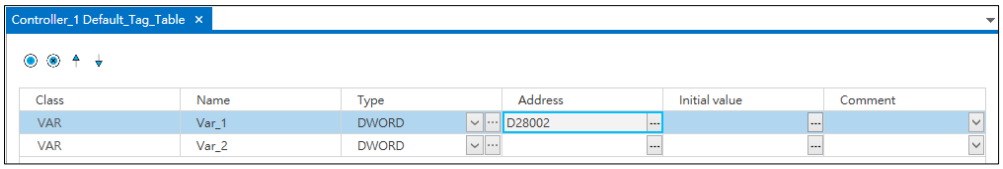


Figure 6 - 22: Default Tag Table (Global Variable Table)

6.1.1.9 Reset Register Address

For AS PLC series, since the process of module configuration involves many executions to copy/paste, cut/paste and so on, the *Input/Output register range* for each module can be automatically assigned by the system and not according to a continuous sequence.

User can make use of the **Reset Register Address** function to reset the register addresses.

Follow these steps to reset the register address:

1. Click icon in the toolbar in **Hardware Configuration** window.

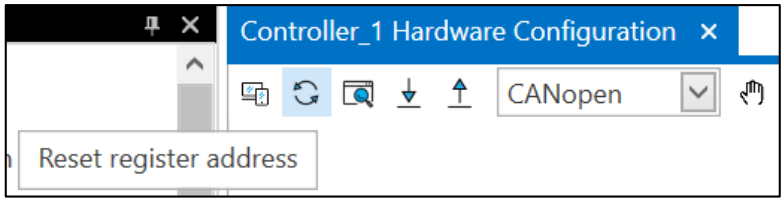


Figure 6 - 23: Reset Register Location Icon

Result: A pop-up window displays as shown in the following figure.

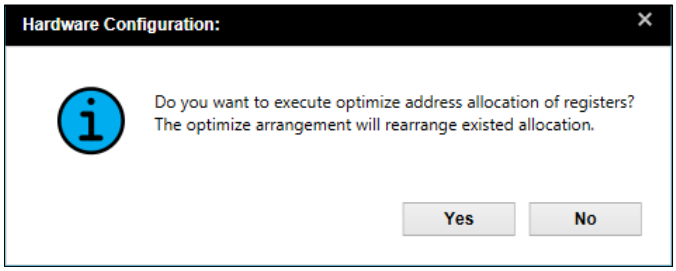


Figure 6 - 24: Reset Register Address – Confirmation

2. Click on **Yes** to reset register address or **No** to cancel the operation.

6.1.1.10 Show/Hide Module Information

User can view the part number of modules, total module width (mm), Internal and External Current Consumption (mA), Power Current Output (mA), RTU numbers and so on, with the help of **Show Module Information**.

Follow these steps to show/hide module information:

1. Click icon in the toolbar of **Hardware Configuration** window.

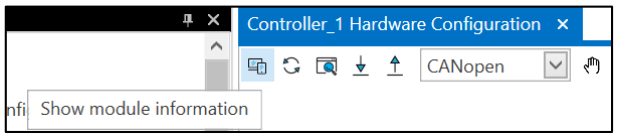


Figure 6 - 25: Display Module details Icon

Result: Module information displays as shown in the following figure.

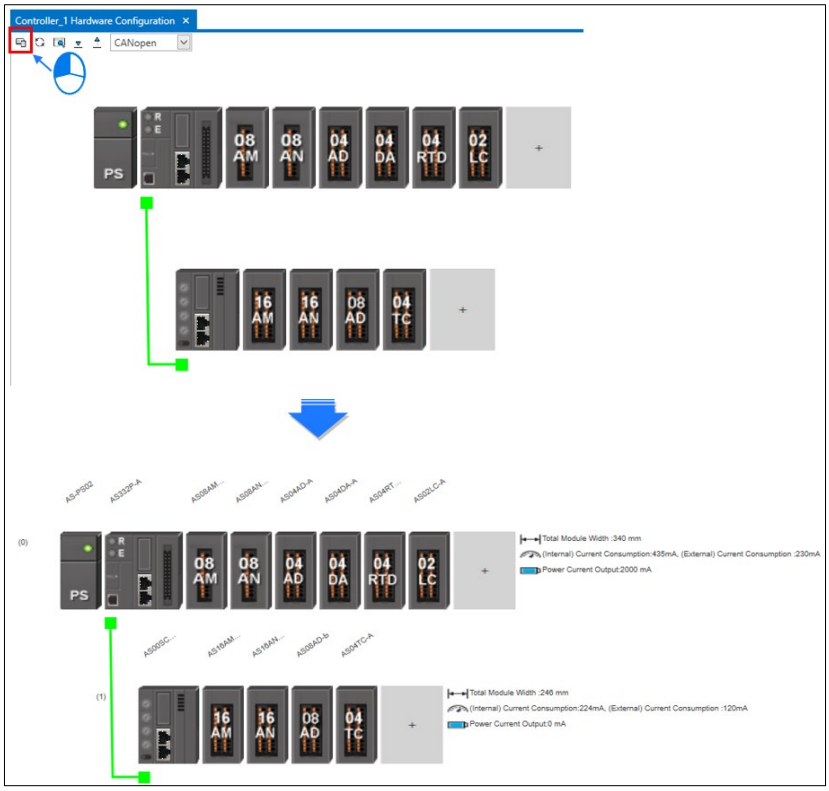


Figure 6 - 26: Show/Hide Module Information

2. Click on the icon again.

Result: Module information is hidden.

6.1.1.11 Set Function Card of Controller

The **Set Function Card** feature allows to set the function card. This function is available only for CPU & NIO modules that support function card.

Follow these steps to set function card:

1. Right-click on the CPU or NIO module as shown in the following figure.

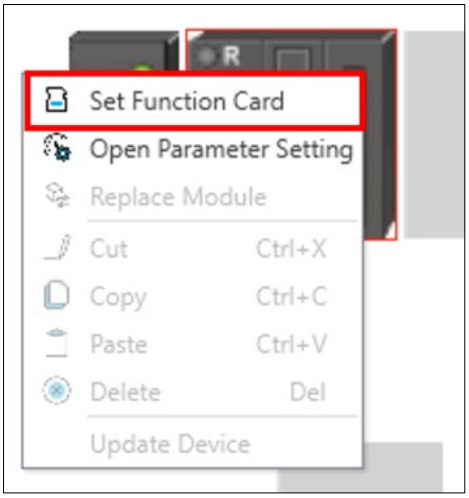


Figure 6 - 27: Set Function Card

2. Click on **Set Function Card**.

   Result: Displays **Function Card Slot List** pop-up menu as shown in the following figure.

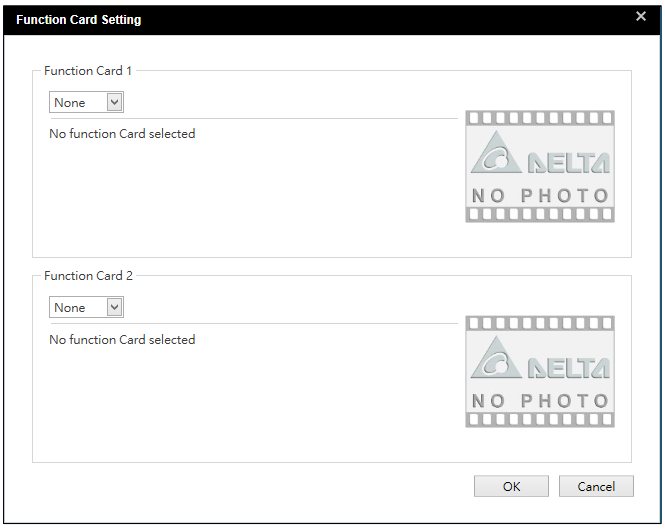


Figure 6 - 28: Function Card Setting

3. Select the required function card in **Slot 1** and/or **Slot 2** and click on **OK** button as shown in the following figure.

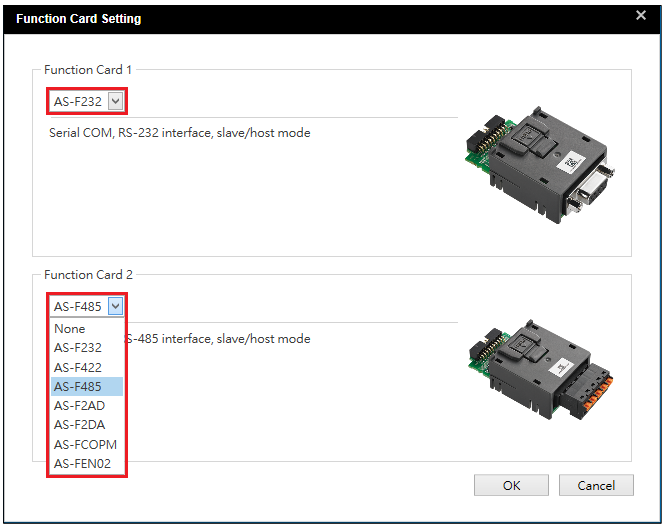


Figure 6 - 29: Function Card Slot List

Result: Selected **Function Cards** are added to the empty slots of CPU as shown in the following figure. Related parameters are updated accordingly.

Figure 6 - 30: Example of Function Card selected

6.1.2 Creating Inverter Configuration

This function is used to set communication cards. When the device supports installing communication cards, the option will be available. User can follow the two methods below to set communication cards.

1. In project Explorer, navigate **Inverter Device** > **VFD Function** > **Hardware Configuration** > **Configuration**.
2. The device, communication card inserted, can perform **Project Upload** (automatically load the communication card settings) after setting communication is complete.

   Result: **Hardware Configuration** window is as shown below. User can manually switch communication card settings or use I/O scan to detect communication card settings after setting communication is complete.

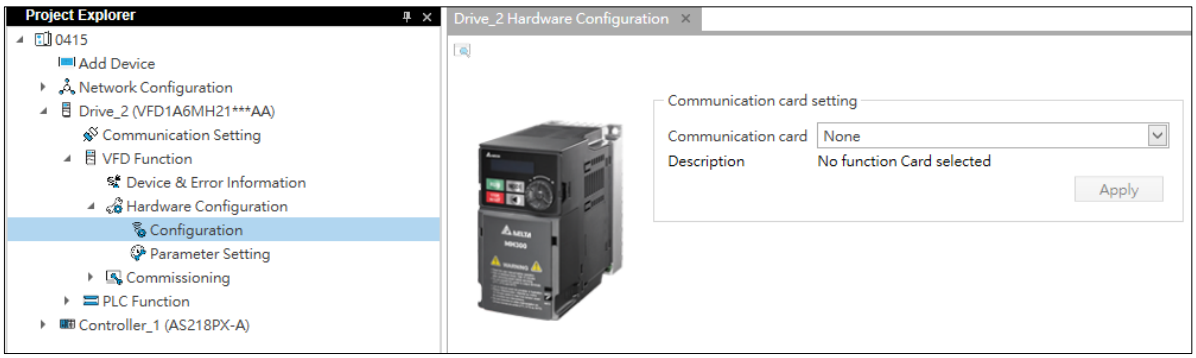


Figure 6 - 31: Hardware Configuration Window

Follow the steps below to set the communication card:

1. User can select the communication card from the drop-down list as per their choice.

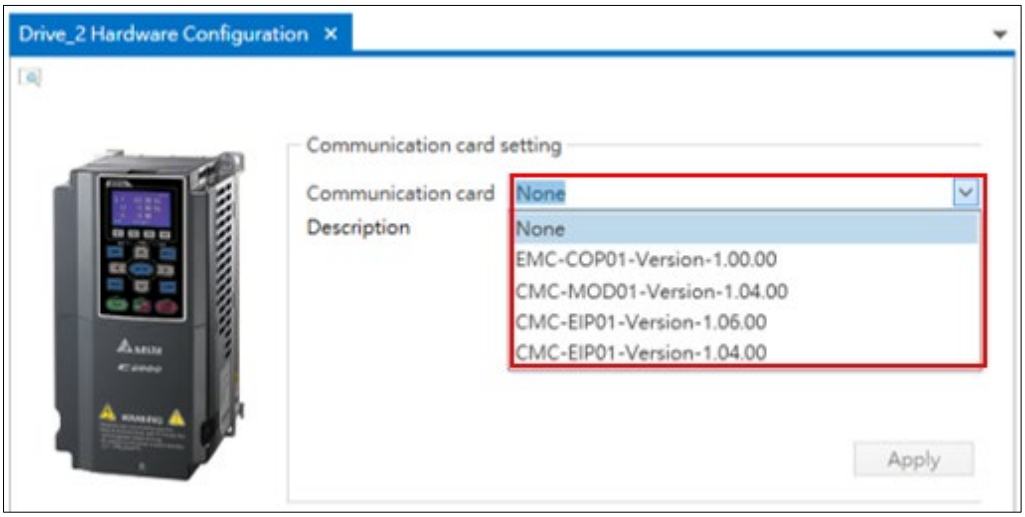


Figure 6 - 32: Communication Card Drop-down List

2. Click **Apply**.

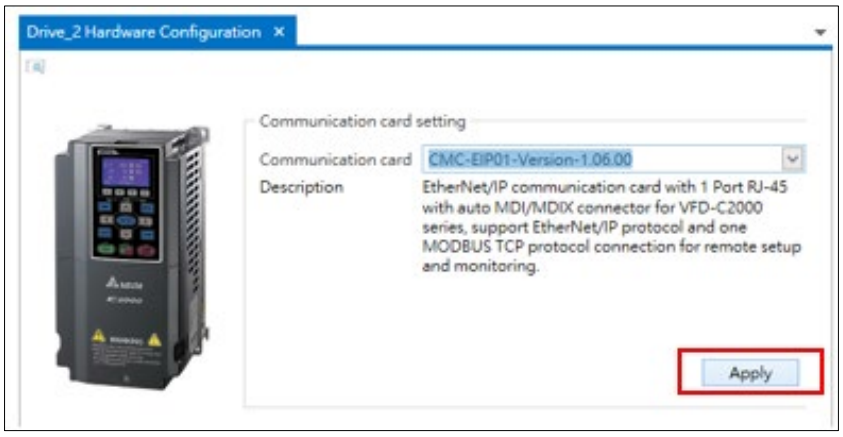


Figure 6 - 33: Communication Card Setting Tab

3. After communication setting is complete, user can open **Hardware Configuration** view. Click to perform I/O Scan.

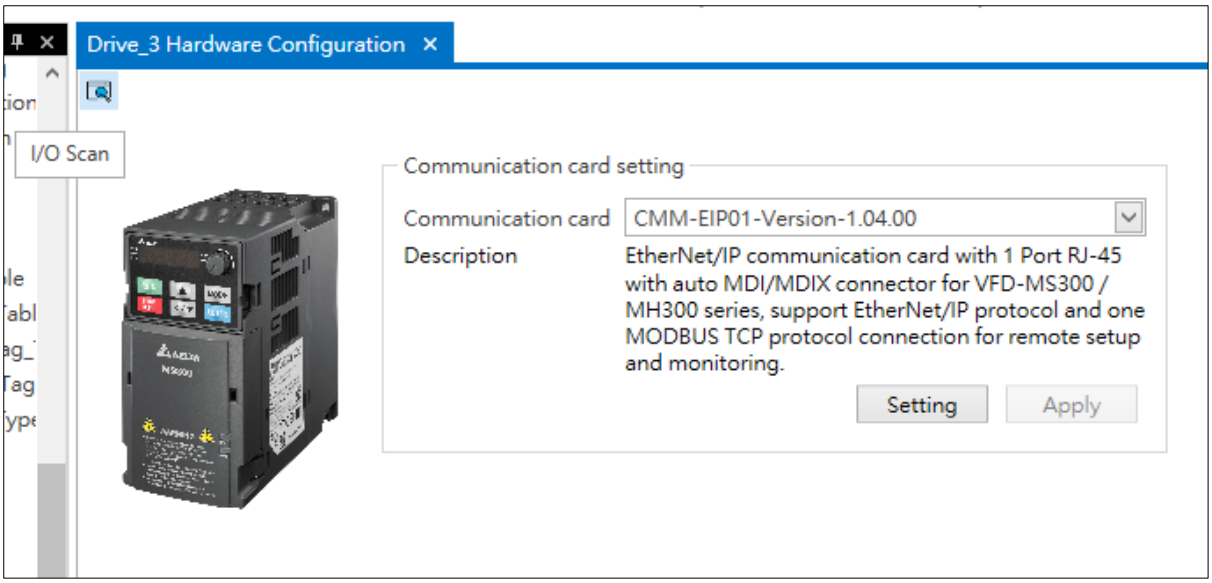


Figure 6 - 34: Communication Card Setting of I/O Scan

Result: After communication card is inserted, the corresponding ports display in Network View view.

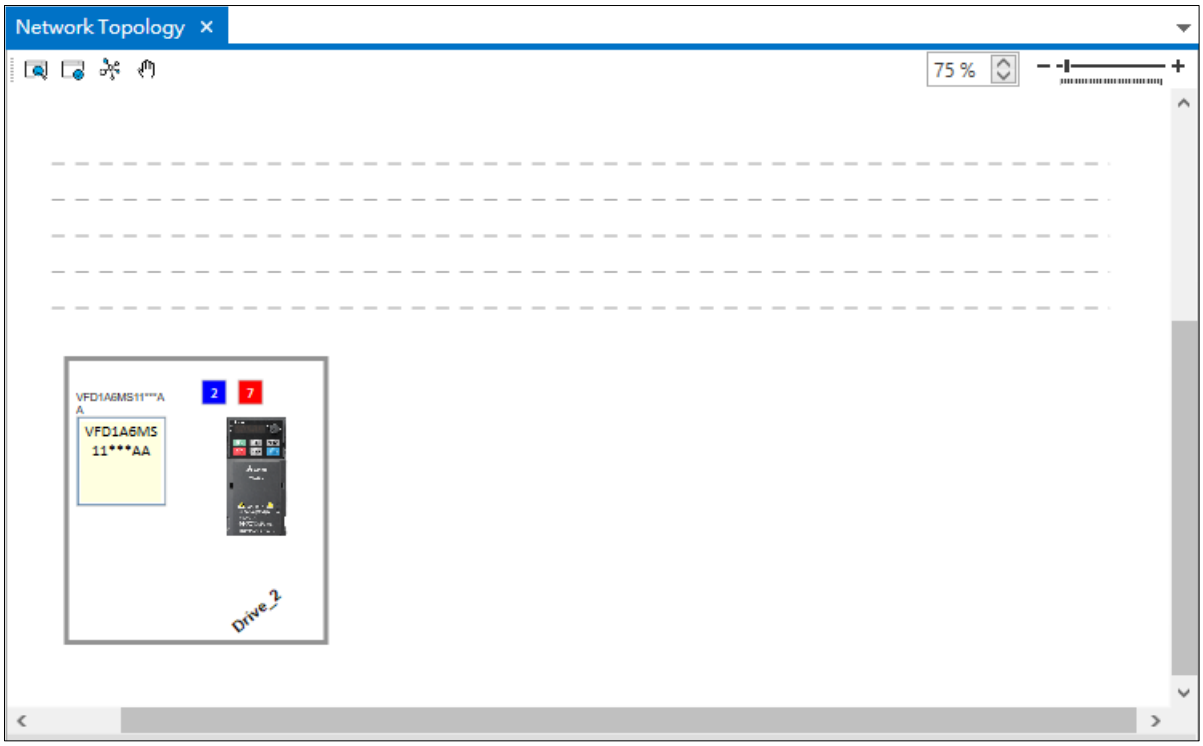


Figure 6 - 35: Insert Ethernet Communication Card

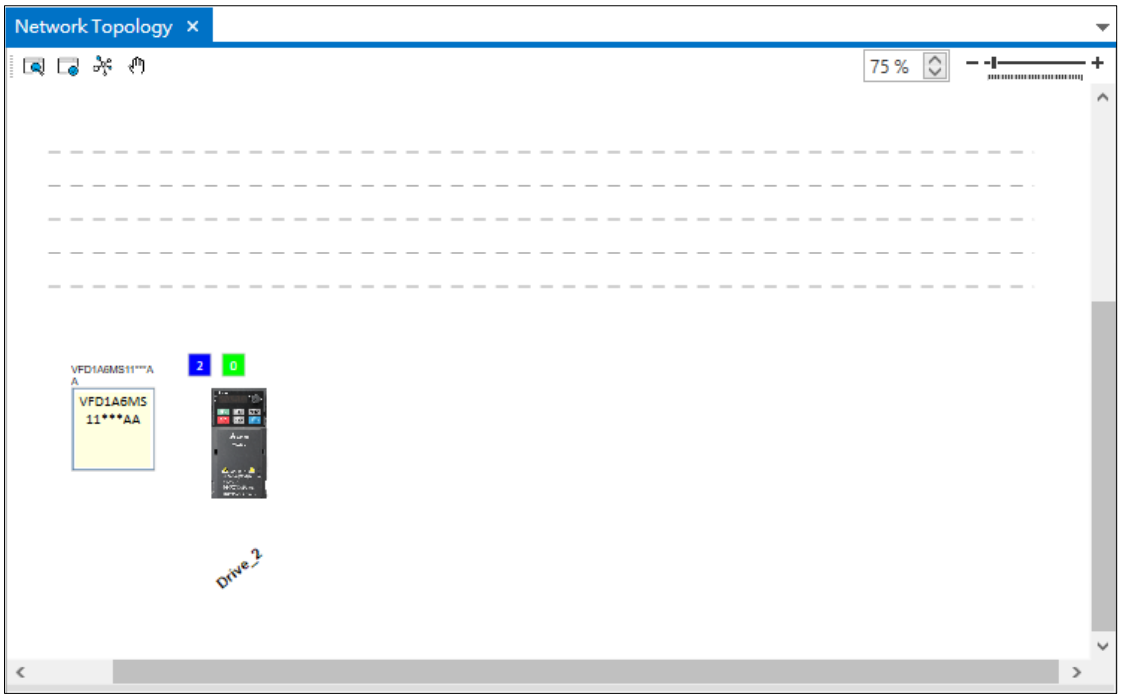


Figure 6 - 36: Insert CANopen Communication Card

Parameter setting of Ethernet communication card:

After inserting the Ethernet communication card, the **Hardware Configuration** view displays the Setting button, as shown below:

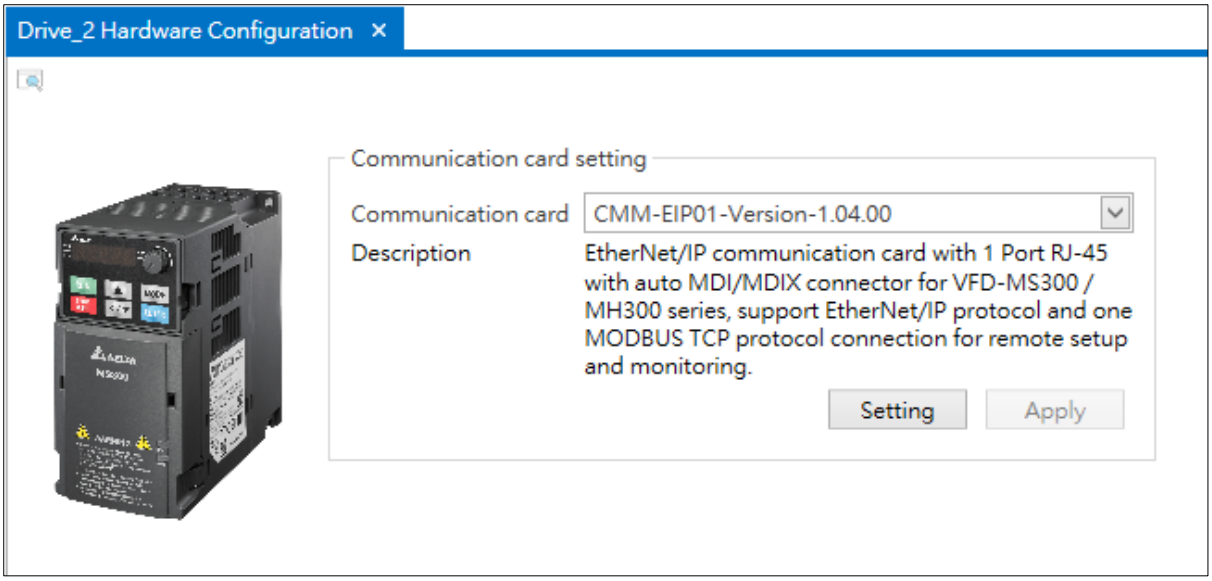


Figure 6 - 37: Setting Button of Parameter Setting of Ethernet Communication Card

Before setting the communication card, please make sure that the computer is connected with the communication card via Ethernet. In addition, COMMGR has established Ethernet drive, and the communication setting of the device in DIADesigner has been successfully connected.

Click the Setting button in hardware configuration to open the function of communication card parameter setting. At normal connection condition, the parameter setting window displays, and shows the device content, as shown below:

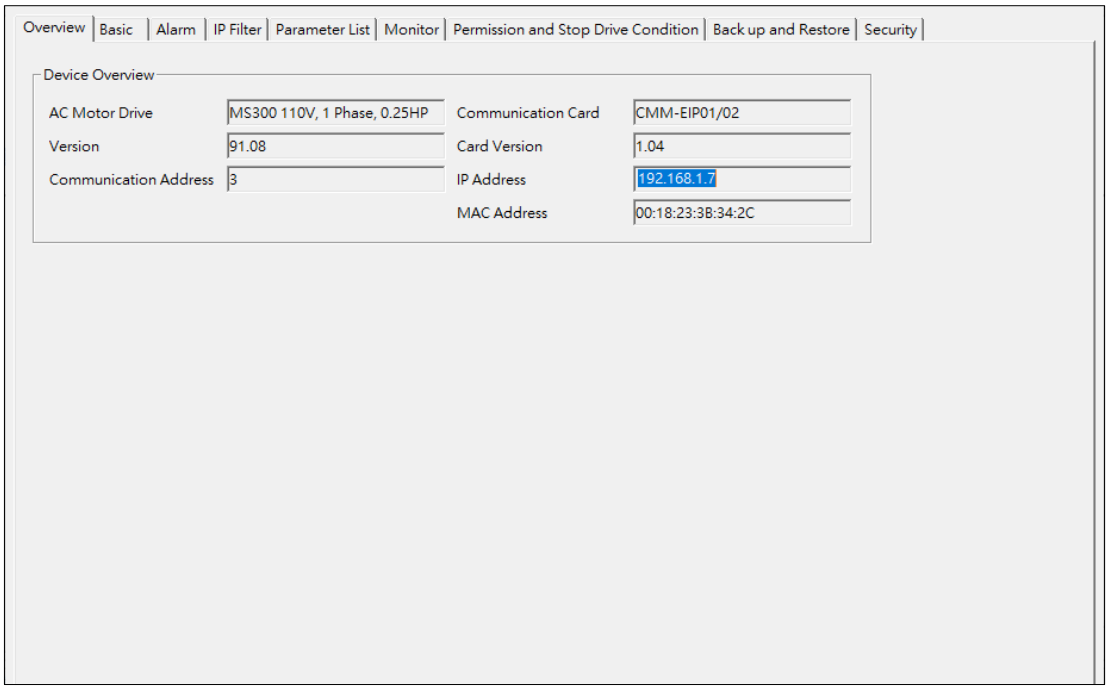


Figure 6 - 38: Function Setting Window of Ethernet Communication Card Parameter Setting

The menu bar of the communication card parameter setting is as shown below:

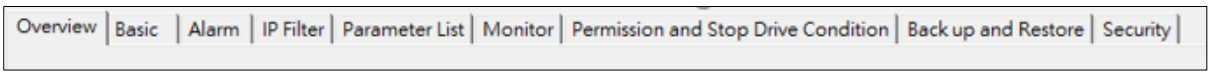


Figure 6 - 39: Menu Bar of Communication Card Parameter Setting

Click **Apply** or **OK** after communication card parameter setting is complete. It will be effective after the set value is saved in the communication card.

1. **Basic**

   Before all network devices connect to the Internet, each of them needs an IP address (Internet Protocol Address), which is like an ID number. User can identify the ID of each device on the network by the ID number.

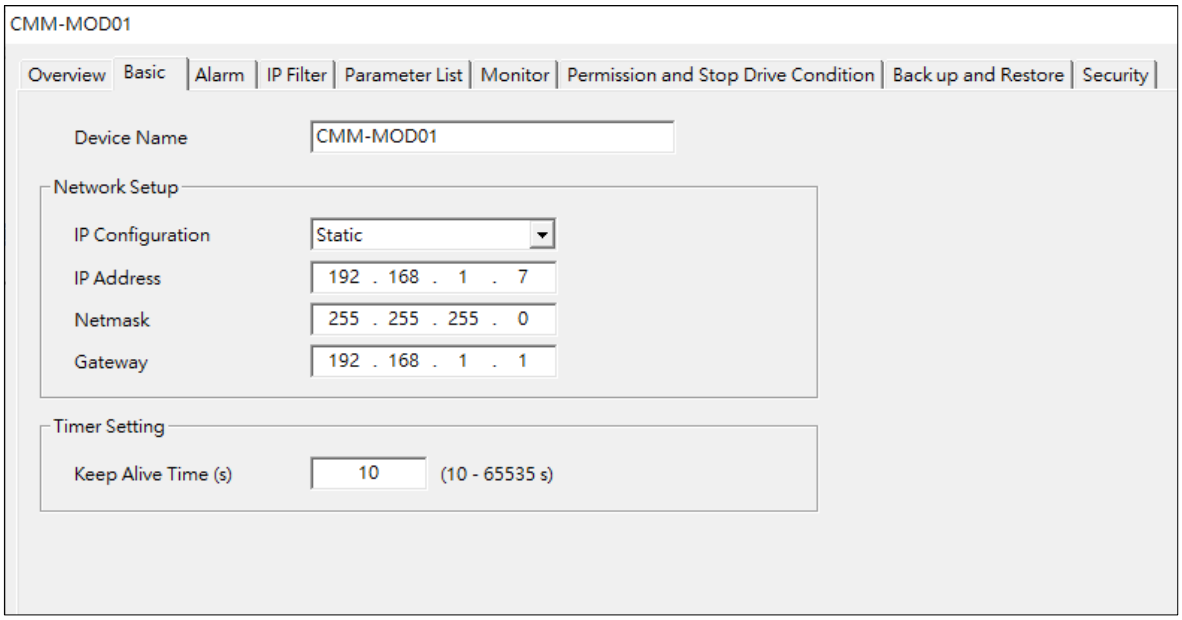


Figure 6 - 40: Network Setting of Communication Card

a. Device Name:

   There might be many CMM-MOD01 cards on the network. To tell which CMM-MOD01 card user wants to set, user has to set the device name, which helps to distinguish devices during research.

b. IP Configuration:

   Choose a way to obtain IP, Static IP or DHCP/BOOTP.

   - Static IP: user-preset or user manually modify.
   - DHCP/BOOTP: automatically update through server; therefore, there have to be servers on local network.

c. IP Address:

IP Address is a unique address that devices use to identify itself and communicate with other devices in the IP network. Any device connected to the IP network must have a unique IP address within the network. If a wrong IP address is used, the connection might fail, even causing other devices to disconnect.

d. Netmask:

Netmask is used to set important parameters of subnet and to determine whether the IP address of the destination device is in the same subnet as the local device. If the destination address is not in the same subnet, the device will send the packet to a gateway, and the gateway will send the packet to another subnet. If the setting is incorrect, it might cause abnormal communication.

e. Gateway:

A gateway is a networking device that connects two networks using different protocols. For instance, local network needs a gateway to connect with Wide Area Network (WAN). Gateway IP and devices must be in the same subnet.

f. Timer Setting:

Set communication time

NOTE: *For correct connection in DIADesigner, after a new IP address is downloaded, click* ***Inverter device > Communication Setting >*** *select* ***COMMGR*** *from Communication Interface drop-down list > select a new* ***IP address*** *from IP Address drop-down list under COMMGR.*

2. **Alarm**

When the condition of triggering email is met, the alarm messages will be sent to user via email.

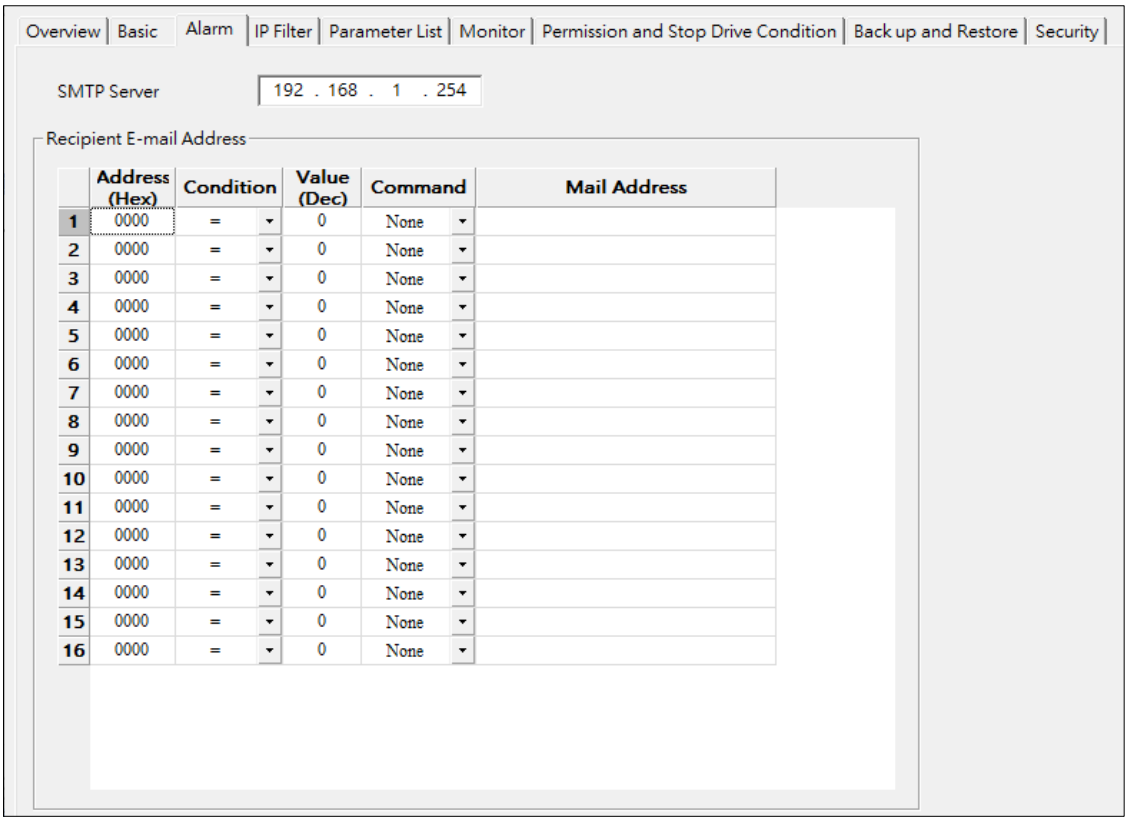


Figure 6 - 41: Alarm Setting of Communication Card

a. Mail Server (SMTP):

Emails are sent to SMTP, and then SMTP sends the emails to designated addresses. For example, if an email is sending to Test@delta.com.tw, and SMTP IP is “172.16.144.122”; then the email will be firstly sent to SMTP, and SMTP will send it to Test@delta.com.tw.

b. Recipient email address:

When the condition of sending an email is triggered, the email will be sent to the recipient, which user set.

3. **IP Filter**

IP filter is used to restrict network connection lest there is uncertain/error IP communication. The connection can be established only in the range that user set, and the rest IP addresses will be rejected.

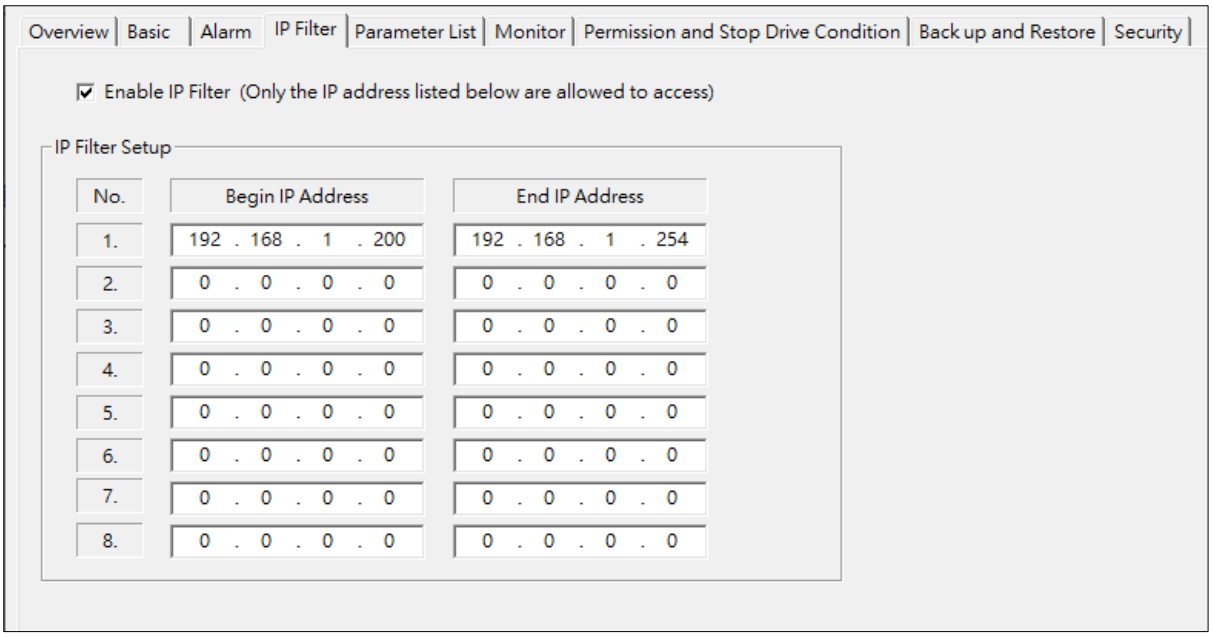


Figure 6 - 42: IP Filter Setting of Communication Card

a. Enable IP Filter

Click the checkbox to enable IP filtering. The filter function starts working according to user-defined setting.

b. IP filter setting:

Up to eight IP address ranges can be set to establish connection at a time

4. **Monitor**

Provides real-time parameter monitoring of devices, and decimal/hexadecimal register present value monitoring per requirement.

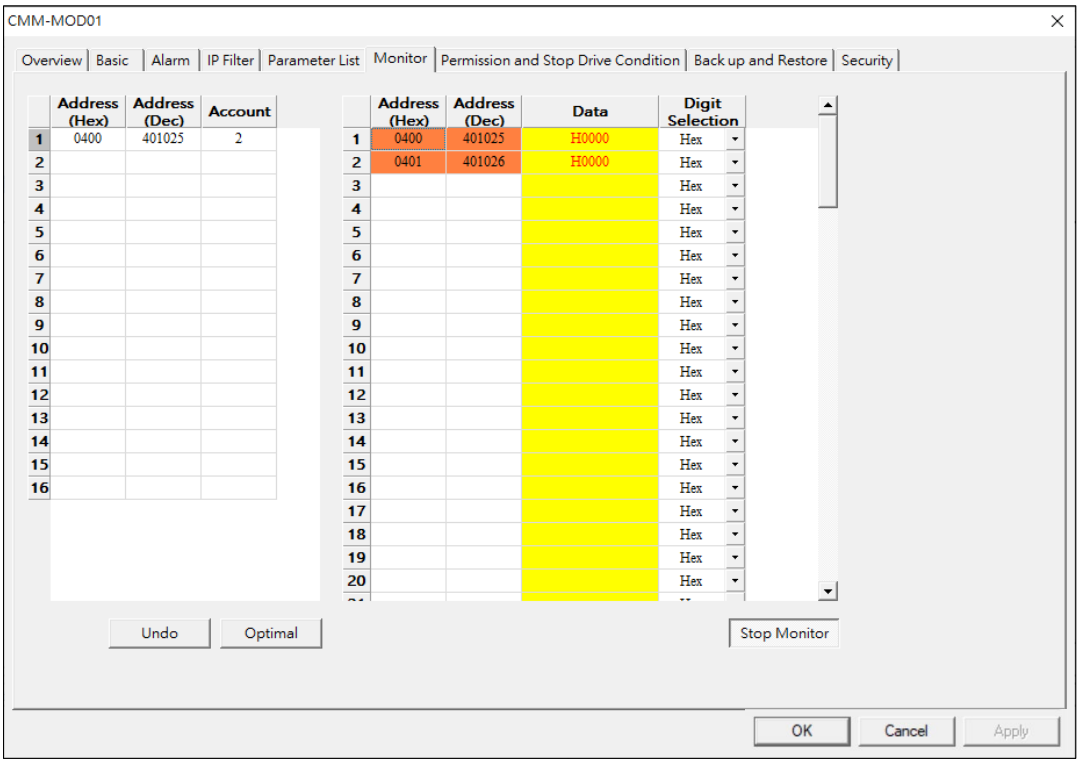


Figure 6 - 43: Communication Card Monitoring Setting

a. Select the parameters that user would like to monitor from the parameter list. Up to 16 sets of discontinuous parameters can be selected.

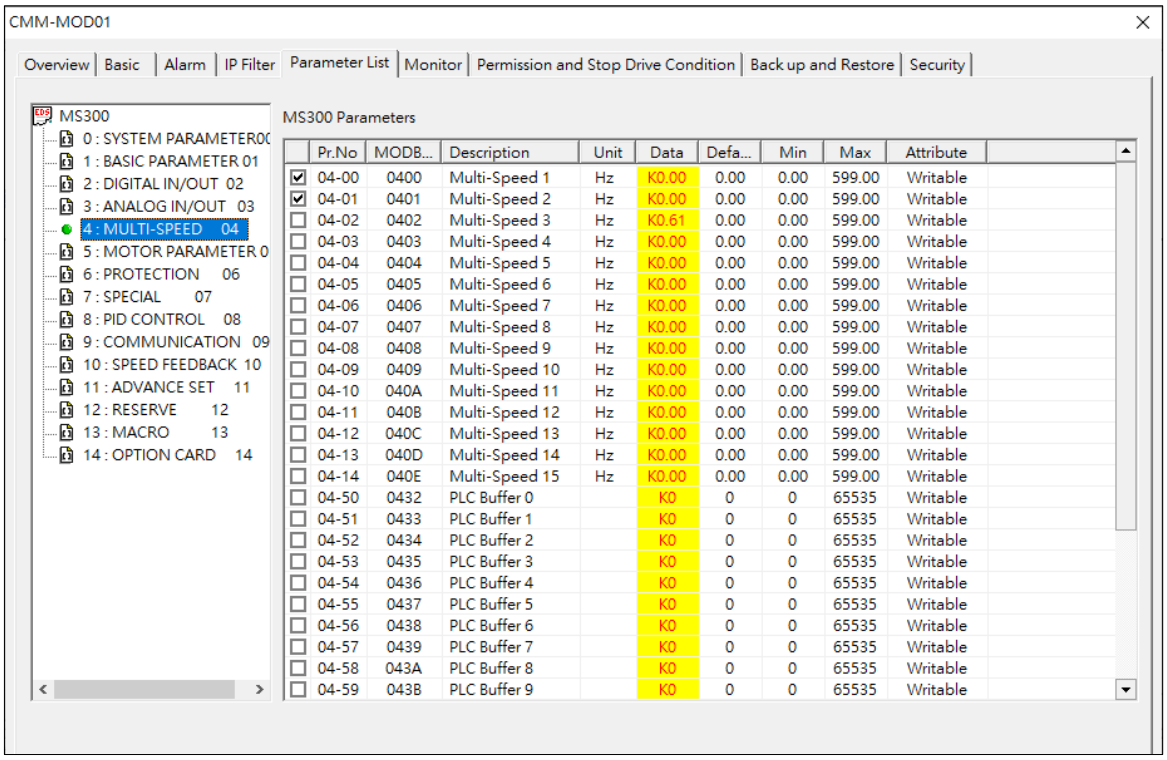


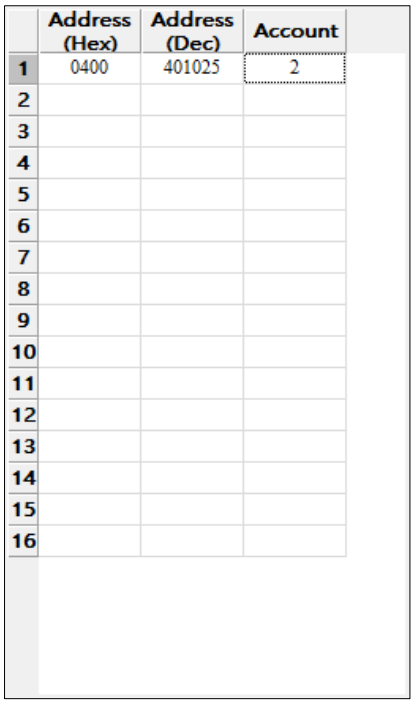
Figure 6 - 44: Parameter List of Communication Card

b. Switch to the **Monitor** tab. User can see the parameter addresses that just selected, including the following:

   i. Address (Hex): 4-digit hexadecimal communication address of parameters

   ii. Address (Dec): 6-digit decimal communication address of parameters

   iii. Account: the continuous parameters that user wants to monitor (Max. 100)



	Address (Hex)	Address (Dec)	Account
1	0400	401025	2
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Figure 6 - 45: Parameter List of Communication Card Monitoring Setting

c. Click **Apply** to save the parameter addresses to be monitored to the communication card, then directly start monitoring. The content includes the following:

   i. Digit Selection: from the drop-down list, user can select decimal or hexadecimal registers.

   ii. Start/Stop Monitor: start/stop the saved parameter monitoring; the unsaved parameters will be deleted.

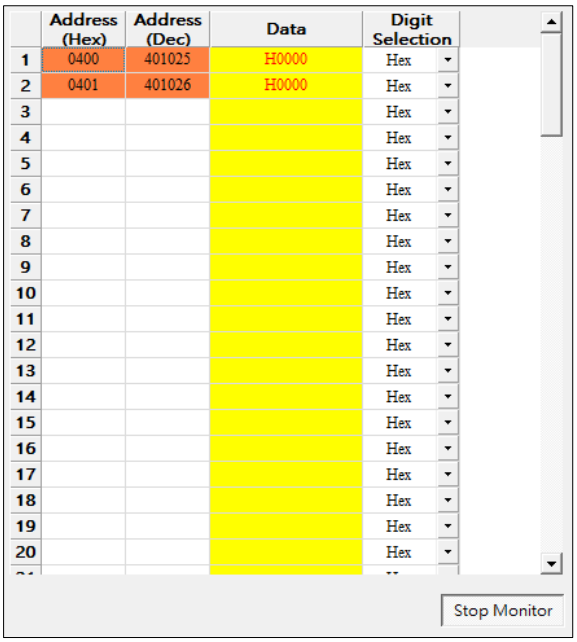


Figure 6 - 46: Present Value of Communication Card Monitoring Parameter

5. Permission and Stop Drive Condition

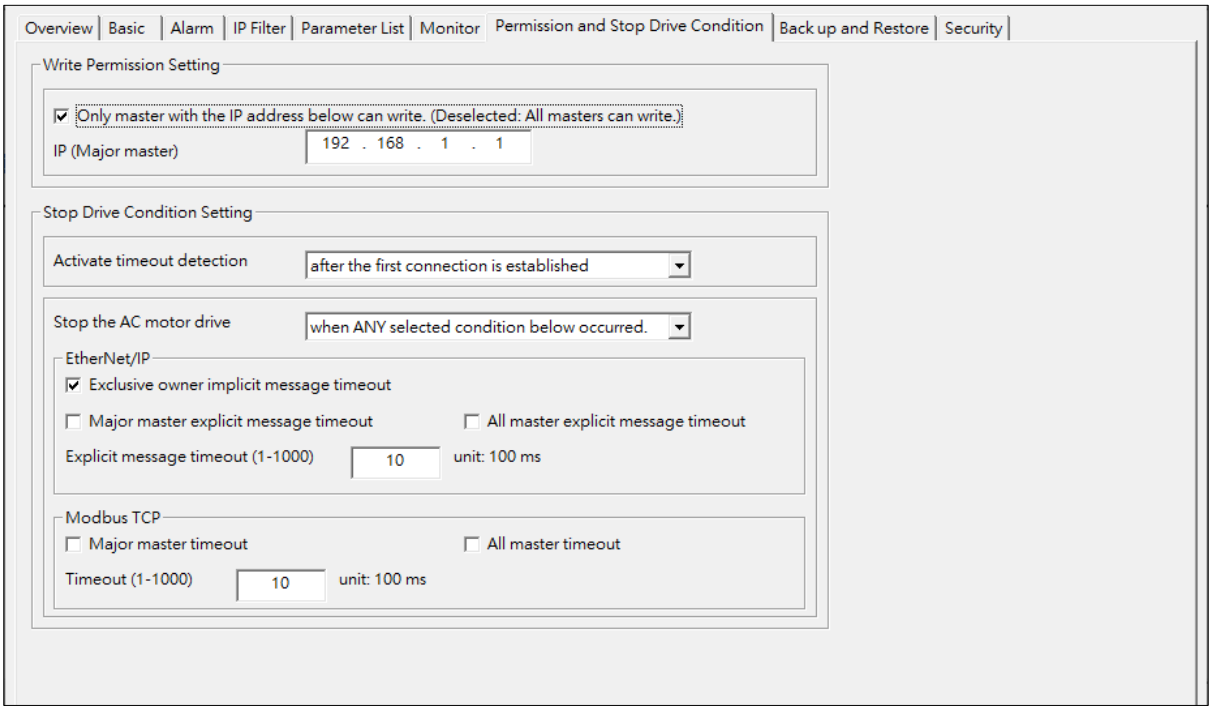


Figure 6 - 47: Permission and Stop Drive Condition

a. Grant write permission: Enable **Only master with the IP address below**

can write. Only IP major master has write permission.

b. Timeout stop setting: select the start conditions of stopping the inverter. Separately choose “timeout setting & the stop conditions of communication timeout” for “EtherNet/IP & Modbus TCP”.

6. **Back up and Restore**

In Backup and Restore tab, user can back up the current parameter files or replace the current parameter setting by reading parameter files.

Figure 6 - 48: Parameter Backup and Restore of Communication Card

a. Backup to File: Click the **Backup to File** button to display the **Save As File** window, where user can choose the parameter file storage location and save the current setting to the parameter file.

b. Restore from File: Click on it, the **Restore Option** window displays, and click **OK**, the **Open** window displays, where user can open a parameter file. The set value of this parameter file will replace the current setting.

7. **Security**

In the **Security** tab, user can set password in order to prevent parameters from being modified

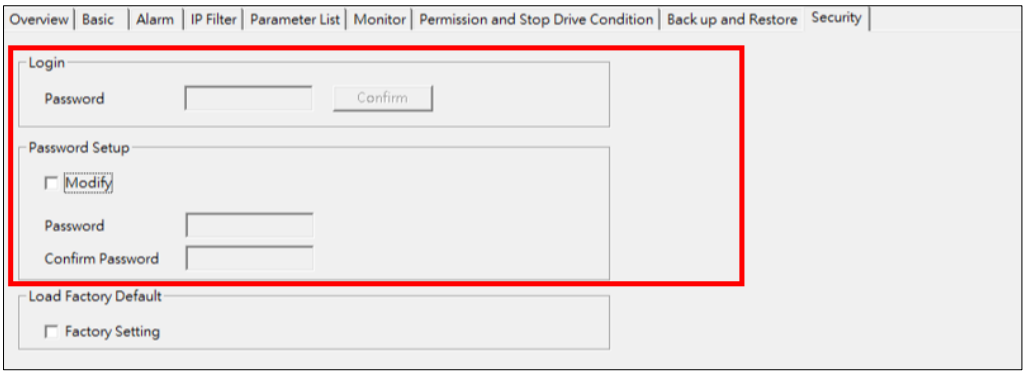


Figure 6 - 49: Parameter Password Authentication of Communication Card

a. Password setup: check the modify box and enter password

b. Confirm Password: re-enter password

c. Login: After password setting is saved in the communication card, every time user open the parameter setting page of communication card, the login password needs to be confirmed to unlock the lock state.

   ***NOTE**: Once the password state is "locked", all parameters cannot be set/changed until the password is "unlocked". If password is lost, by serial port (RS-485) connection, user can restore the system to factory setting to unlock the "locked" state, and all parameters will be back to the default values.*

d. Load Factory Default: To make all parameters back to the default values, check the **Factory Setting** box.

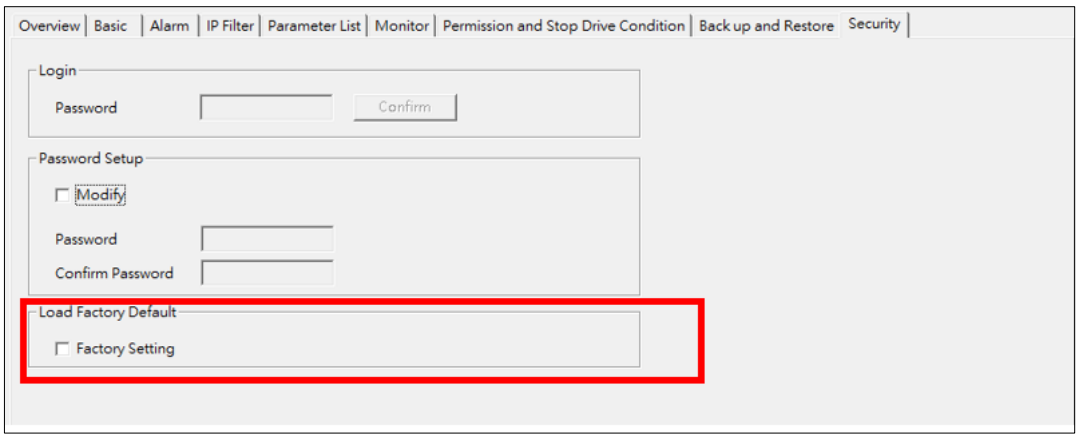


Figure 6 - 50: Factory Reset of Communication Card Parameters

NOTE: *While returning to factory setting through serial port (RS-485) communication, modules need 10 seconds to change the settings. Do not turn off the power at this time.*

After opening the parameter setting of the communication card, the **Setting** button in the hardware configuration of DIADesigner cannot be clicked. Until the parameter setting page of communication card is all closed, **Setting** will be enabled again.

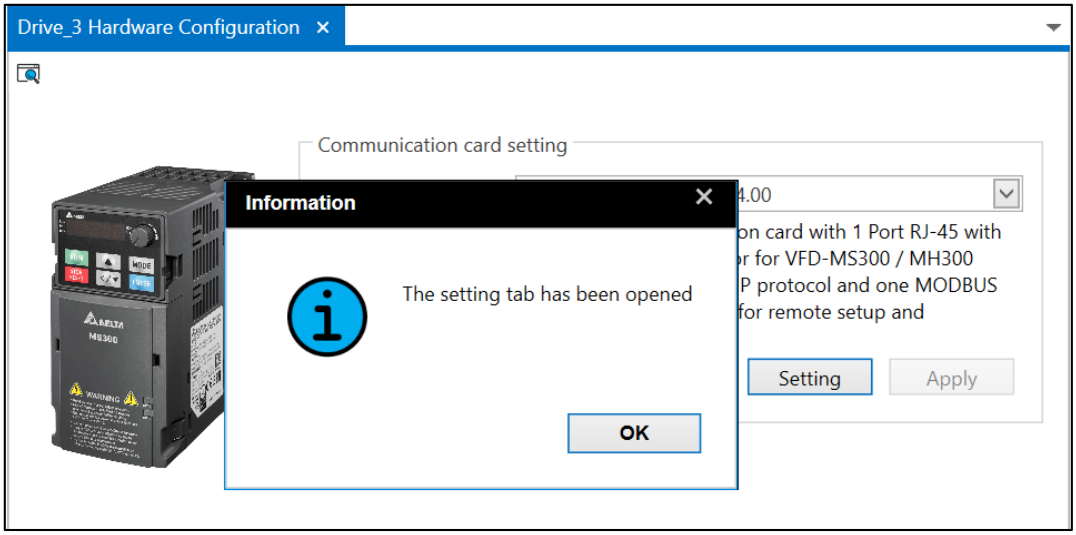


Figure 6 - 51: Performing Parameter Setting of Ethernet Communication Card

6.2 Parameter Setting

6.2.1 Parameter Settings for PLC devices

After finishing **Hardware Configuration**, user can set the CPU & module parameters with the help of **Parameter Setting**. For more information on the hardware and operations of the CPU or modules, refer *Catalogs*, *Hardware Manual* and *Operation Manual*.

***NOTE**:*

*Before the performing **Parameter Setting** in a CPU module, user have to refer the Operation Manual. To prevent damage to the system or staff, user have to make sure of the facts that affect the CPU module and the whole system. **Parameter Setting** are not effective until they are downloaded to the CPU module.*

The **Parameter Setting** window can be opened from different locations: -

- Double-click **Controller** > **Hardware Configuration** > **Parameter Setting** or
- Double-click the icon of CPU or modules in the **Configuration** window.

  Result: The **Parameter Setting** window displays. PLC **Parameter Setting** window is shown in the following figure.

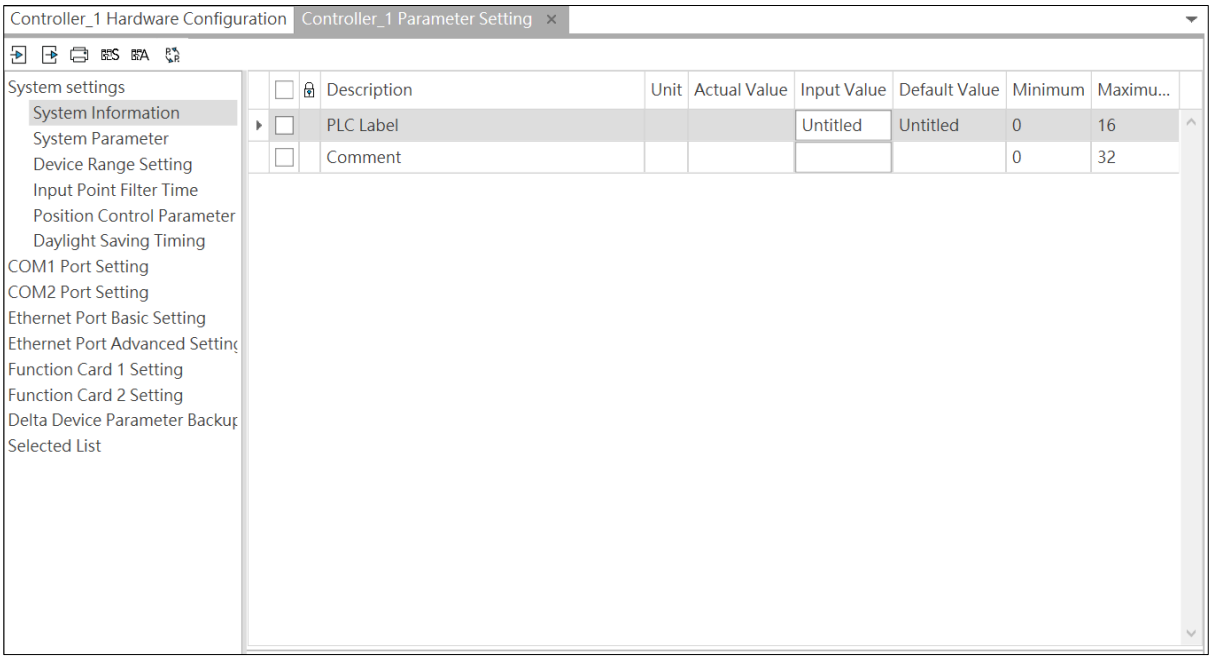


Figure 6 - 52: PLC – Parameter Setting

The toolbar in **Parameter Setting** window is shown in the following figure.

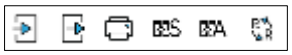


Figure 6 - 53: PLC – Parameter Setting window – Toolbar

The functions in the **Toolbar** of **Parameter Setting** window are described in the following table:

Function	Description
	Click to import the parameters. Upon click, **Open** window displays, where user can select the parameter file.
	Click to export the parameters. Upon click, **Save As** window displays, where user can save the parameter file.
	Click to print the parameter list. Upon click, **Print…** window displays, to perform print operation.
	Click to replace the selected parameter with its default value.
	Click to reset all the parameters to their default values.
	Click to compare the properties of CPU. Upon click, **Parameter Compare** window displays.

Parameters are classified into several tabs in the **Parameter Setting** window. There are primary tabs and secondary tabs. Parameters vary upon the PLC module series and the modules selected.

All the Parameter Setting tabs have the columns described in the following table:

Function	Description
Description	Displays the parameter description
Unit	Displays the parameter unit
Actual Value	Display the parameter actual value after upload
Input Value	Select/Set the parameter value

Function	Description
Default Value	Displays the default value of parameter
Minimum	Displays the minimum value for parameter
Maximum	Displays the maximum value for parameter

Follow these steps to edit the PLC parameters:

1. Enter parameter in **Input Value** column.

   Result: If the **Input Value** is different from the **Default Value**, then the **Default Value** column will be colored as shown in the following figure.

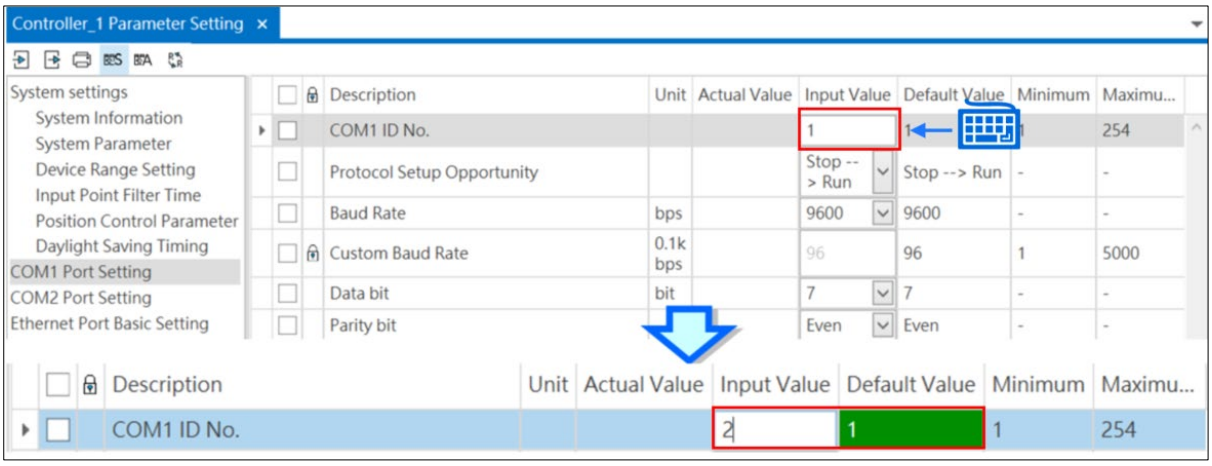


Figure 6 - 54: PLC - Edit Parameter

6.2.2 DTC parameter editor

In the project tree, double-click the parameter editor of the temperature controller device.

Result: The parameter editor window will be displayed. The thermostat parameter tab is a preset tab. The parameter editor window of the thermostat is shown as follows:

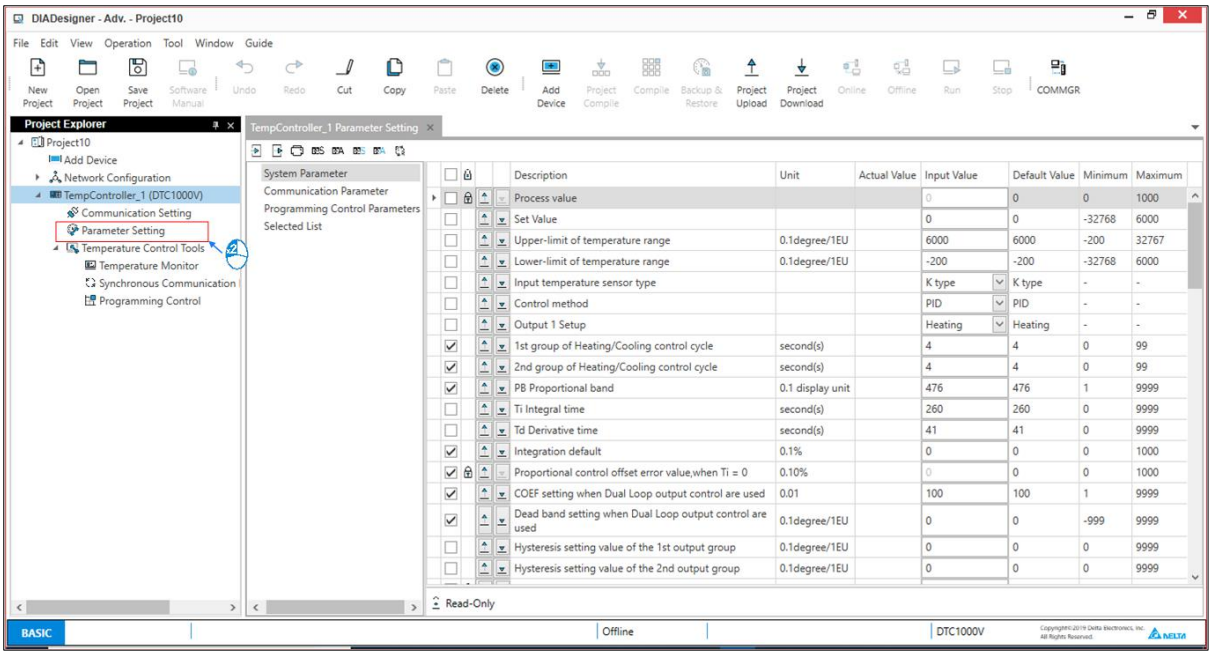


Figure 6 - 55: Thermostat parameter editor

There is a toolbar in the parameter editor window, the icon is as follows

Figure 6 - 56: Toolbar

The functions in the toolbar are described in the following table:

Features	Description
	Parameter import function. After clicking, an open window will be displayed for the user to select the parameter file to be imported.
	Parameter export function. After clicking, the Save New File As window will be displayed for the user to select the storage location where the parameter file is expected to be exported.
	Parameter printing function. After clicking, a print window will be displayed for user to perform print operations.

Features	Description
	After clicking, the selected parameter input value will be replaced with the default value.
	After clicking, all parameter input values will be replaced with preset values.
	After clicking, the selected parameter input value will be replaced with the actual value.
	After clicking, all parameter input values will be replaced with actual values.
	Parameter comparison function of the same model CPU host. After clicking, the comparison parameter window will be displayed.

In the parameter editor window, the parameters will be classified into several group tabs, and their groups will vary by series and models.

All parameter setting tabs have the following fields:

Features	Description
	Indicates that this parameter is a read-only parameter.
	Indicates that this parameter can be set during operation.
	Provide this parameter independent single parameter upload/download function.
Name	Display parameter number
Address	Display the actual memory address of the parameter
description	Display parameter description
Unit	Display parameter unit
Actual value	Display the actual value of the parameter after successful upload/download
input value	Set parameter value
Default value	Display parameter preset value
Minimum	Display parameter minimum
Max	Maximum display parameter

Edit parameters according to the following procedure:

1. Enter the parameter value in the input value field.

   Result: When the input value is different from the preset value after editing, the preset value field will be marked with color.

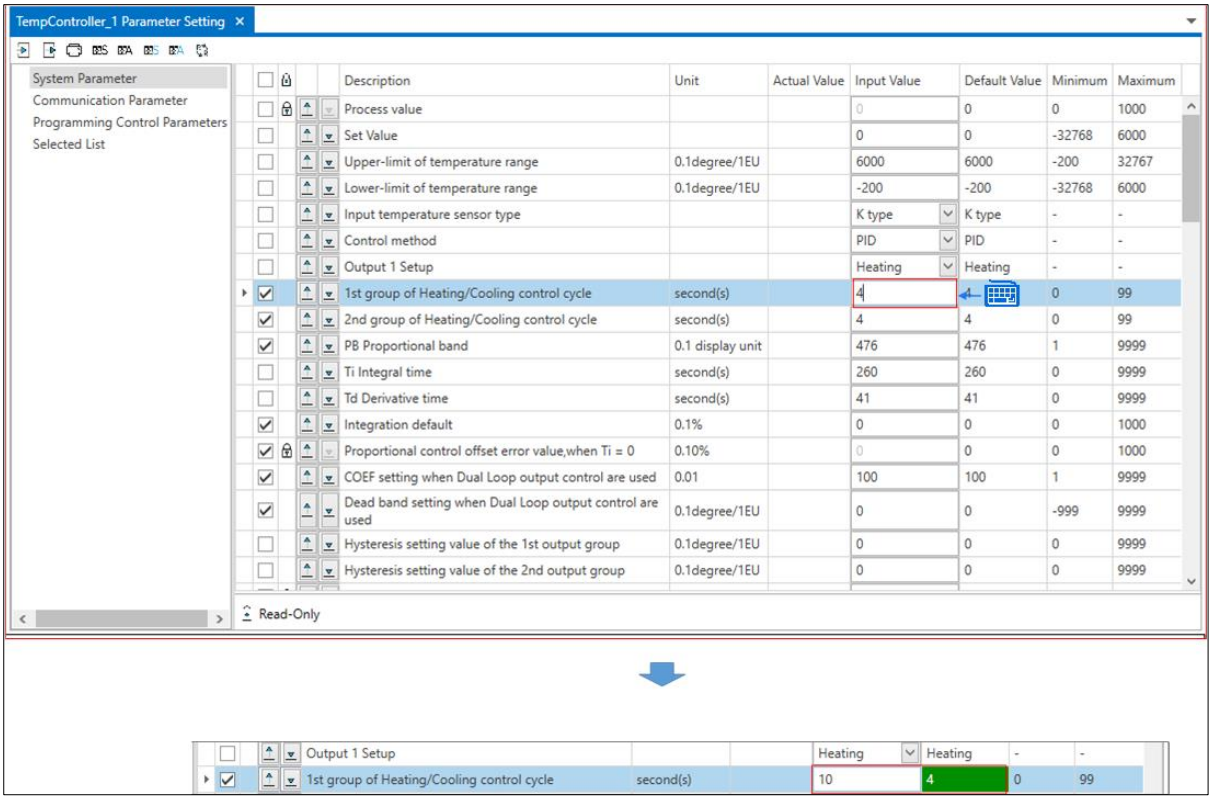


Figure 6 - 57: Edit parameters

6.2.3 Parameter Settings for Servo devices

Double-click **Servo Device** > **Parameter Setting** in the **Project Explorer** under the desired device.

Result: The **Parameter Setting** window displays. Servo Device **Parameter Setting** window is shown in the following figure.

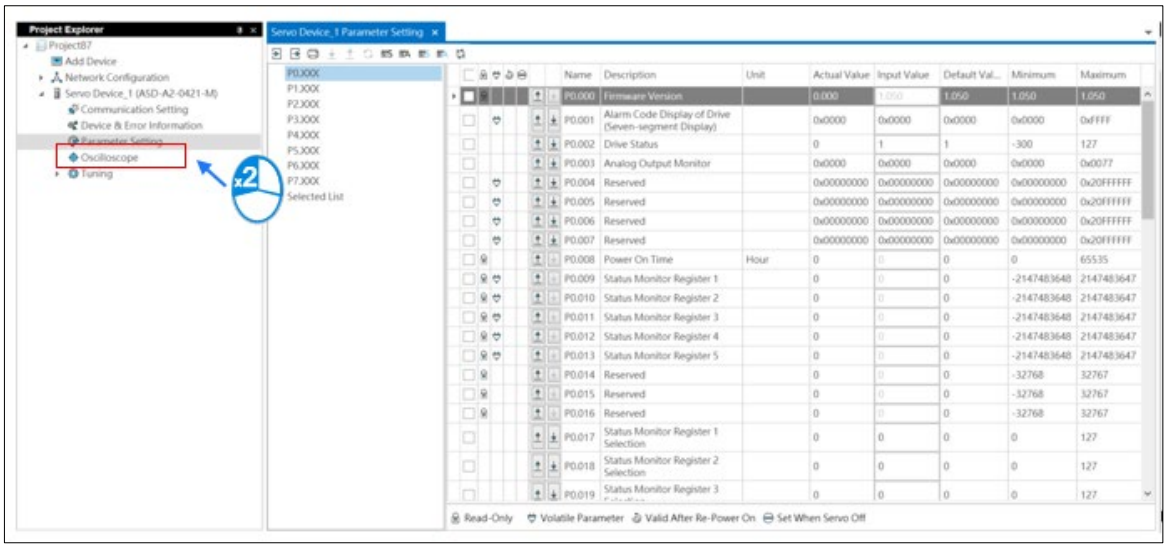


Figure 6 - 58: Servo Device – Parameter Setting

The toolbar in Servo Device **Parameter Setting** window is shown in the following figure.

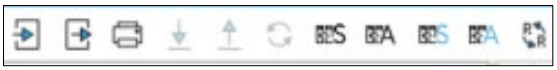


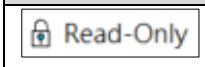
Figure 6 - 59: Servo Device – Parameter Setting – Toolbar

The functions in the Toolbar of Servo Device **Parameter Setting** window are described in the following table:

Function	Description
	Click to import the parameters. Upon click, **Open** window displays, where user can select the parameter file.
	Click to export the parameters. Upon click, **Save As** window displays, where user can save the parameter file.
	Click to print the parameter list. Upon click, **Print** window displays to perform print operation.
	Click to download the parameters. Upon click, **Download View** window displays, where user can select the parameters they want to download.

Function	Description
	Click to upload the parameters. Upon click, **Upload View** window displays, where user can select the parameters they want to upload.
	Click to continuously update the parameters one-by-one for the display group.
	Click to replace the selected parameter with its default value.
	Click to reset all the parameters to their default values.
	Click to replace the selected parameter with its actual value.
	Click to reset all the parameters to their actual values.
	Click to compare the parameters. Upon click, **Parameter Compare** window displays, where user can compare the Parameters.

Parameters are classified into groups (groups depends on series of Servo Device) in the Servo Device **Parameter Setting** window. The columns in each group are described in the following table:

Function	Description
	To indicate if this parameter is read-only parameter
	To indicate if this parameter is reset after power off
	Parameter value will be taken into account after power recycle on the device
	To indicate if this parameter cannot be written while servo on.
	To upload this parameter to drive and to download this parameter from drive.
Name	Display the parameter number.
Description	Displays the parameter description.
Unit	Displays the parameter unit.
Actual Value	Display the parameter actual value after upload.

Function	Description
Input Value	Select/Set the parameter value.
Default Value	Displays the default value of parameter.
Minimum	Displays the minimum value for parameter.
Maximum	Displays the maximum value for parameter.

Follow these steps to edit the Servo Device parameters:

1. Enter parameter in **Input Value** column.

   Result: If the **Input Value** is different from the **Default Value**, then the **Default Value** column will be colored as shown in the following figure.

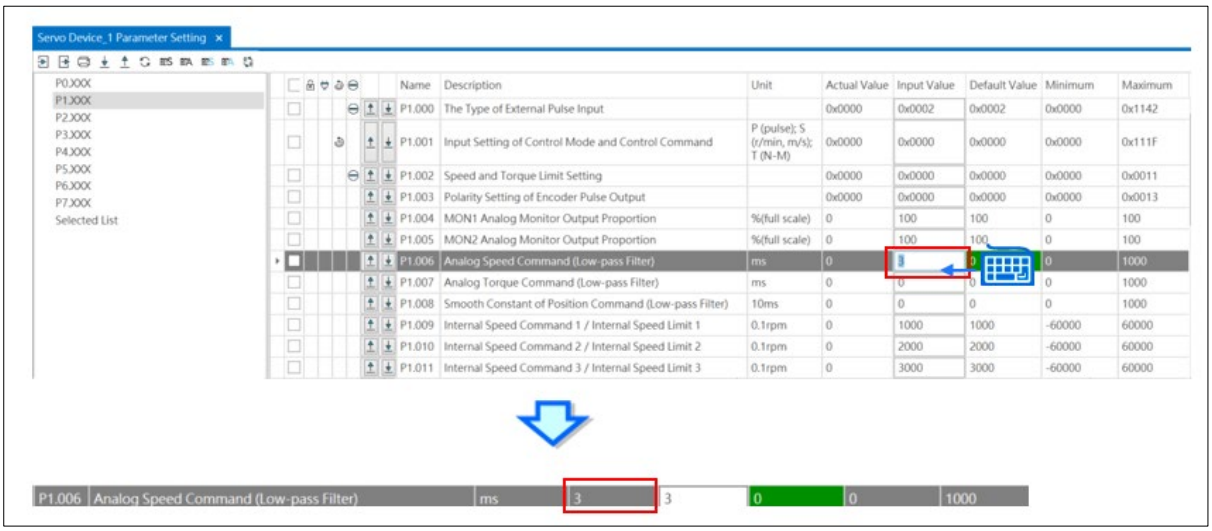


Figure 6 - 60: Servo Device – Edit Parameter

6.2.4 Parameter Settings for Inverter/ASD

Double-click **Drive** > **Hardware Configuration** > **Parameter Setting** in the **Project Explorer** under the desired device.

Result: The **Parameter Setting** window displays. **SYSTEM PARAMTER** tab displays by default. Drive **Parameter Setting** window is shown in the following figure.

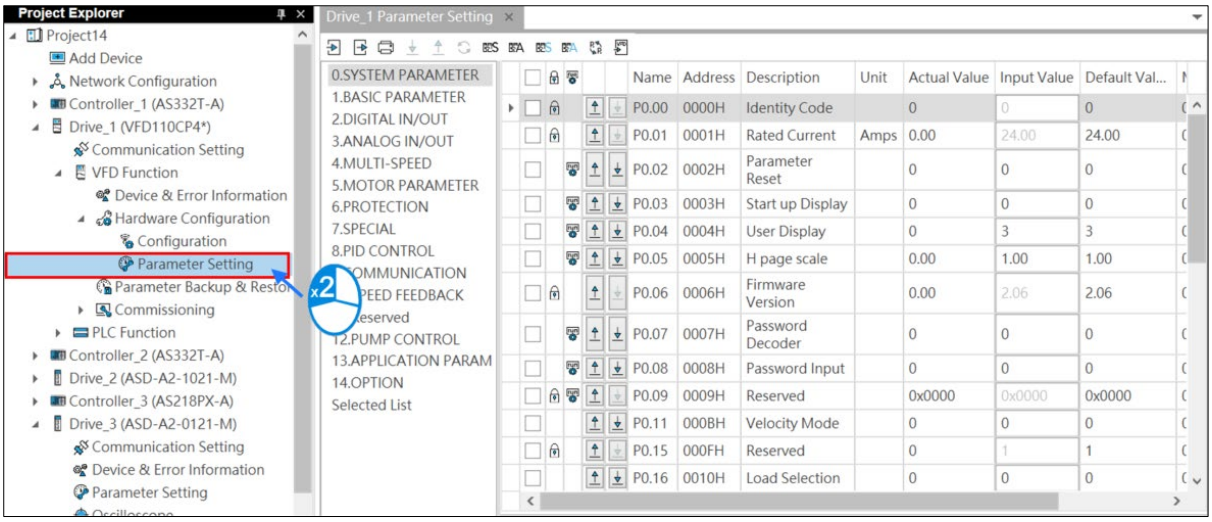


Figure 6 - 61: Drive – Parameter Setting

Drive **Parameter Setting** window **Toolbar** is shown in the following figure.

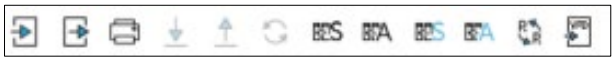


Figure 6 - 62: Drive – Parameter Setting Toolbar

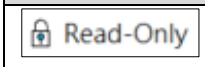
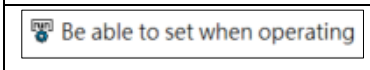
The functions in **Toolbar** for **Parameter Setting** window are described in the following table:

Function	Description
	Click to import the parameters. Upon click, **Open** window displays, where user can select the parameter file.
	Click to export the parameters. Upon click, **Save As** window displays, where user can save the parameter file.
	Click to print the parameter list. Upon click, **Print** window displays to perform print operation.
	Click to download the parameters. Upon click, **Download View** window displays, where user can select the parameters they want to download.

Function	Description
	Click to upload the parameters. Upon click, **Upload View** window displays, where user can select the parameters they want to upload.
	Click to continuously update the parameters one-by-one for the display group.
	Click to replace the selected parameter with its default value.
	Click to reset all the parameters to their default values.
	Click to replace the selected parameter with its actual value.
	Click to reset all the parameters to their actual values.
	Click to compare the parameters. Upon click, **Parameter Compare** window displays.
	Click to import VFDSoft file. Upon click, **Open** window displays, where user can select the VFDSoft parameter file.

Parameters are classified into fifteen groups in the Drive **Parameter Setting** window.

The columns in each group are described in the following table:

Function	Description
	To indicate if this parameter is read-only parameter.
	To indicate if this parameter can be set when operating.
	To upload this parameter to drive and to download this parameter from drive.
Name	Display the parameter number.
Address	Display the memory address of parameter.
Description	Displays the parameter description.
Unit	Displays the parameter unit.
Actual Value	Display the parameter actual value in online.
Input Value	Select/Set the parameter value.
Default Value	Displays the default value of parameter.

Function	Description
Minimum	Displays the minimum value for parameter.
Maximum	Displays the maximum value for parameter.

Follow these steps to edit the Drive parameters:

1. Enter parameter in **Input Value** column.

   Result: The **Actual Value** changes and the **Default Value** column is colored to show the difference of default value as shown in the following figure.

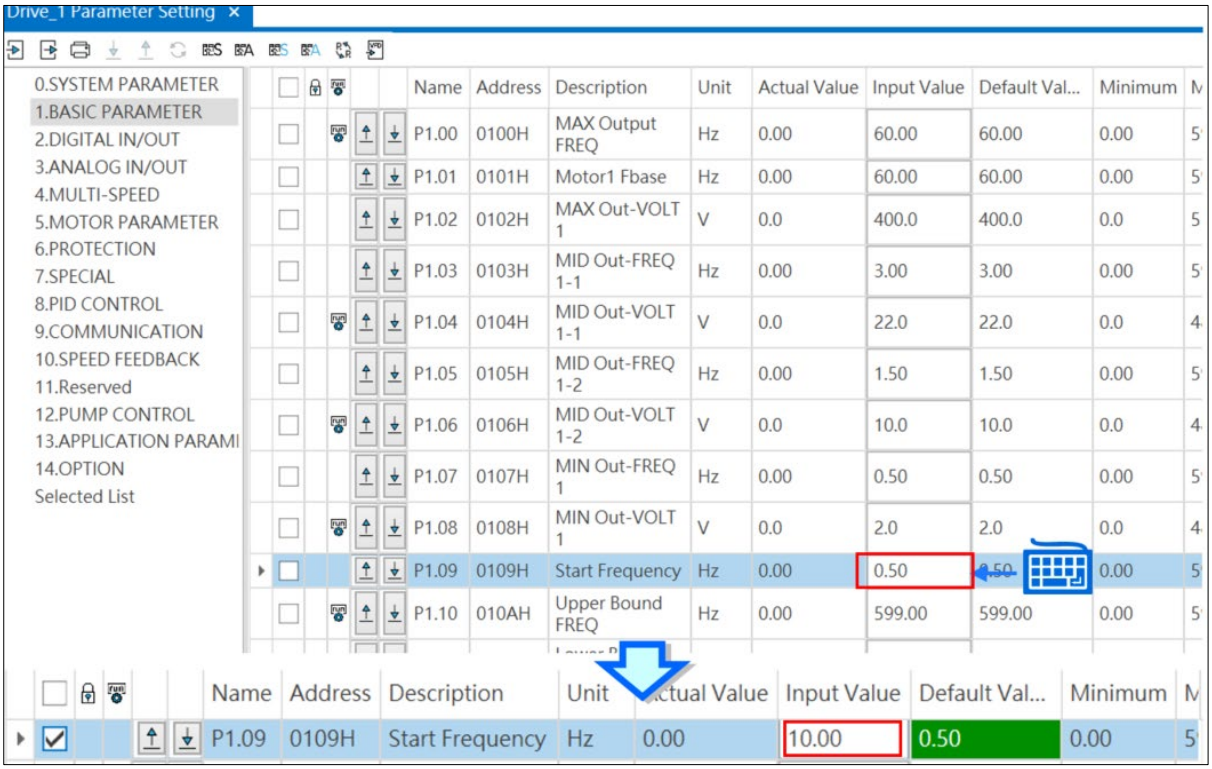


Figure 6 - 63: Drive - Edit Parameter

6.2.5 Perform Parameter Setting After Servo Drives/Inverters Upload Parameters

Parameter Setting provides the **Actual Value** and **Input Value** fields and the purpose is to separate the current device values and the values set by uses. Actual value and Input value can be synchronized through the following methods.

Upload Servo Drives to Read Parameter Actual Values and Edit Parameters

After communication is established, the user reads the parameter actual values. By clicking the **Project Upload** or **Device -> PC** button, as shown in the following figure.

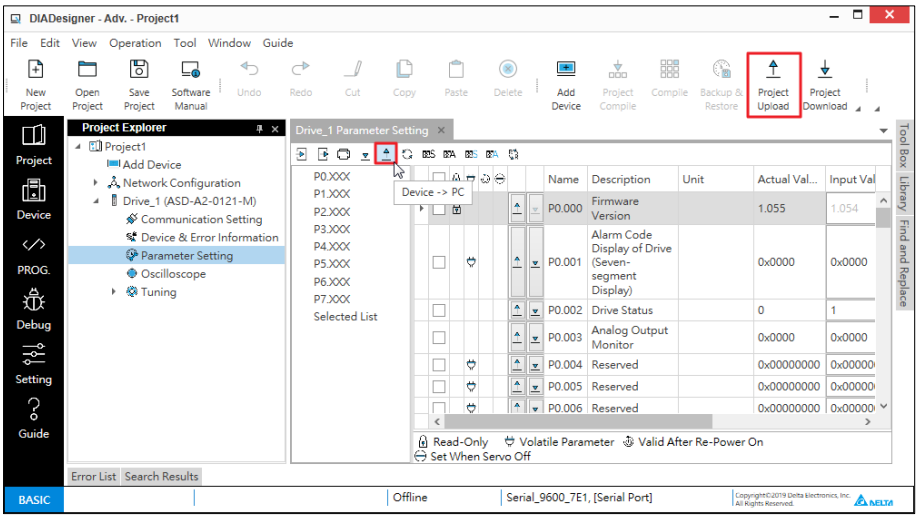


Figure 6 - 64: Parameter Upload

After the upload is complete, the readback parameter values are shown in the Actual Value column, as shown in the following figure.

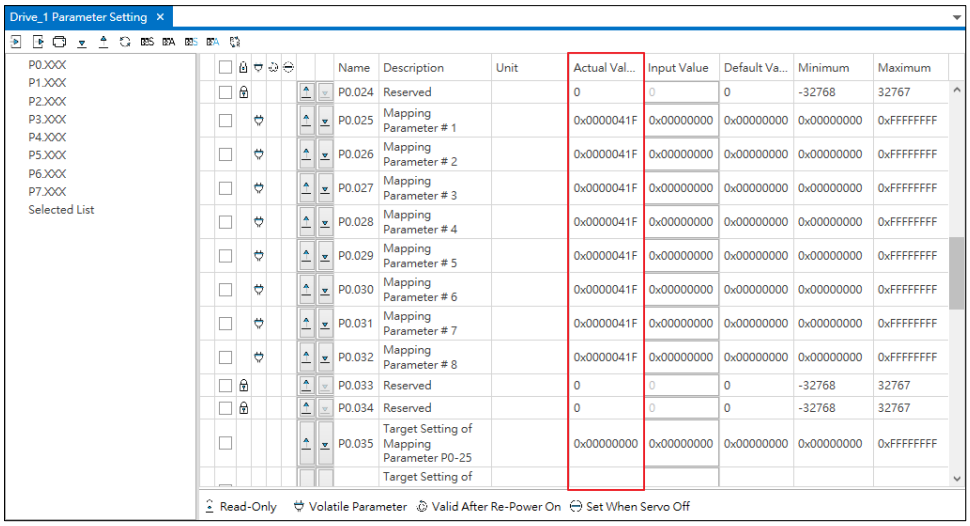


Figure 6 - 65: Actual Value After Upload Complete

If user wants to export the parameters or edit the parameters, then download to the device, click the **Replace the selected parameters with the actual value** or **Replace all parameters with the actual value** button, as shown in the following figure.

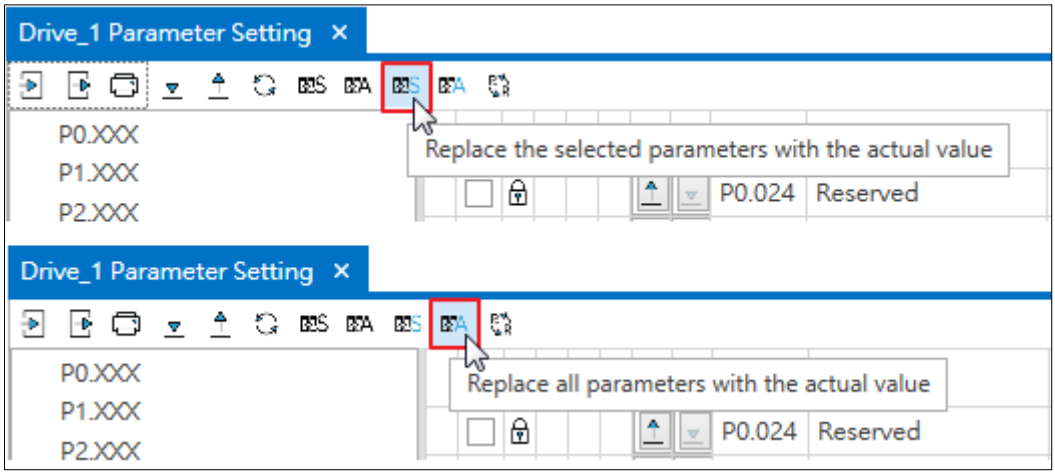


Figure 6 - 66: Write Actual Values to Parameters

The actual values will be written to input values for subsequent editing, as shown in the following figure,

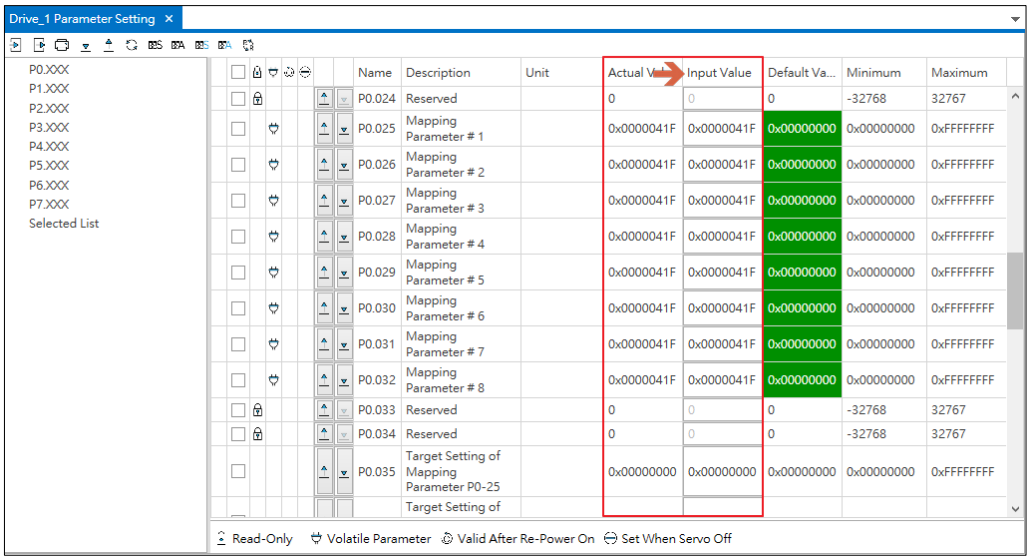


Figure 6 - 67: Actual Values Written to Input Values

Upload Inverters to Read Parameter Actual Values and Edit Parameters

After communication is established, the user read parameter actual values by clicking the **Project Upload** or **Device -> PC** button, as shown in the following figure.

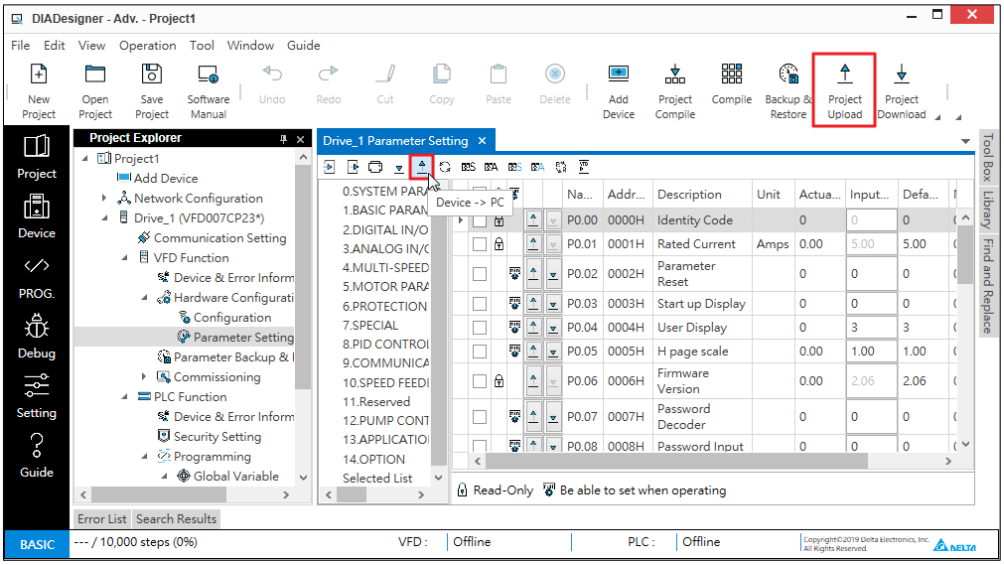


Figure 6 - 68: Parameter Upload

After the upload is complete, the readback parameter values will be written in the Actual Value column, as shown in following figure.

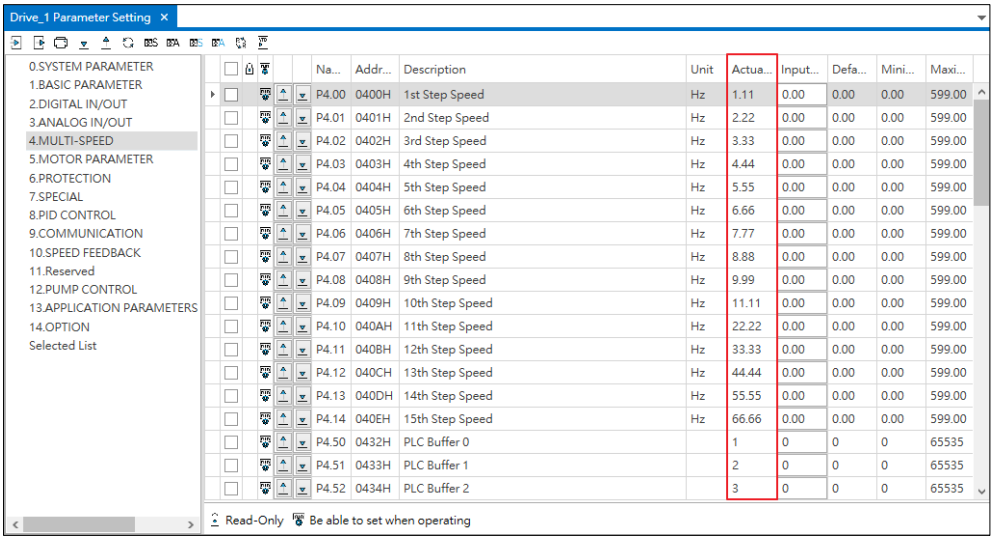


Figure 6 - 69: Actual Value After Upload Complete

If user wants to export the parameters or edit the parameters then download to the device, click the **Replace the selected parameters with the actual value** or **Replace all parameters with the actual value** button, as shown in the following figure.

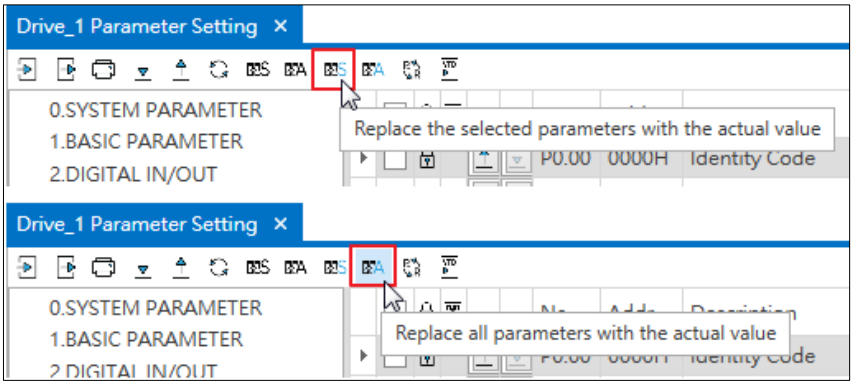
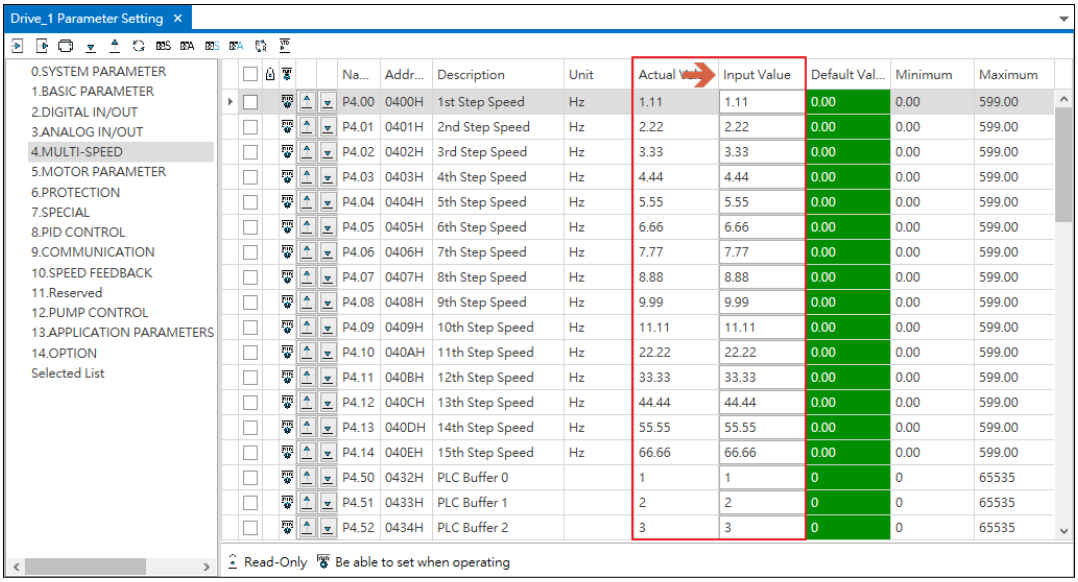


Figure 6 - 70: Write Actual Values to Parameters

The actual values will be written to input values for subsequent editing, as shown in the following figure.



Na...	Addr...	Description	Unit	Actual Val...	Input Value	Default Val...	Minimum	Maximum
P4.00	0400H	1st Step Speed	Hz	1.11	1.11	0.00	0.00	599.00
P4.01	0401H	2nd Step Speed	Hz	2.22	2.22	0.00	0.00	599.00
P4.02	0402H	3rd Step Speed	Hz	3.33	3.33	0.00	0.00	599.00
P4.03	0403H	4th Step Speed	Hz	4.44	4.44	0.00	0.00	599.00
P4.04	0404H	5th Step Speed	Hz	5.55	5.55	0.00	0.00	599.00
P4.05	0405H	6th Step Speed	Hz	6.66	6.66	0.00	0.00	599.00
P4.06	0406H	7th Step Speed	Hz	7.77	7.77	0.00	0.00	599.00
P4.07	0407H	8th Step Speed	Hz	8.88	8.88	0.00	0.00	599.00
P4.08	0408H	9th Step Speed	Hz	9.99	9.99	0.00	0.00	599.00
P4.09	0409H	10th Step Speed	Hz	11.11	11.11	0.00	0.00	599.00
P4.10	040AH	11th Step Speed	Hz	22.22	22.22	0.00	0.00	599.00
P4.11	040BH	12th Step Speed	Hz	33.33	33.33	0.00	0.00	599.00
P4.12	040CH	13th Step Speed	Hz	44.44	44.44	0.00	0.00	599.00
P4.13	040DH	14th Step Speed	Hz	55.55	55.55	0.00	0.00	599.00
P4.14	040EH	15th Step Speed	Hz	66.66	66.66	0.00	0.00	599.00
P4.50	0432H	PLC Buffer 0		1	1	0	0	65535
P4.51	0433H	PLC Buffer 1		2	2	0	0	65535
P4.52	0434H	PLC Buffer 2		3	3	0	0	65535

Figure 6 - 71: Actual Values Written to Input Values

Chapter 7: Setting Network (Network Configuration)

7.1 Network View

Network View is an industrial network connected to the full range of Delta Industrial Automation products, which can be used according to the planned network view.

Network View is the graphical representation of connections between devices that allows user to plan their networks.

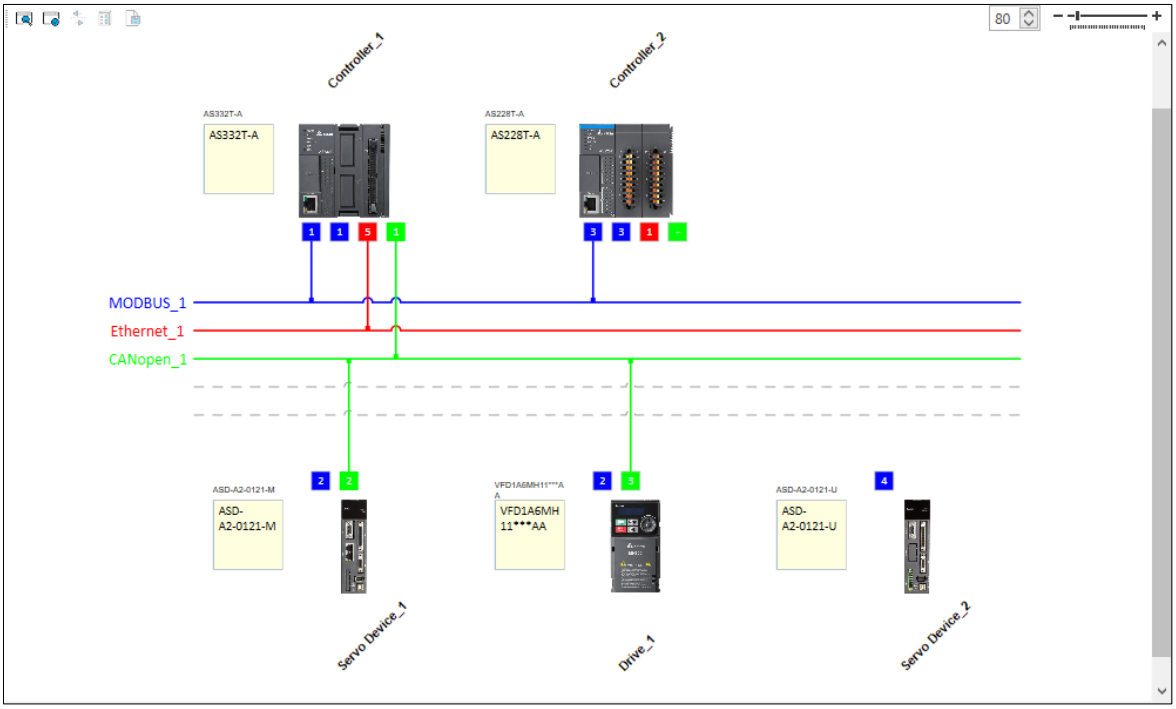


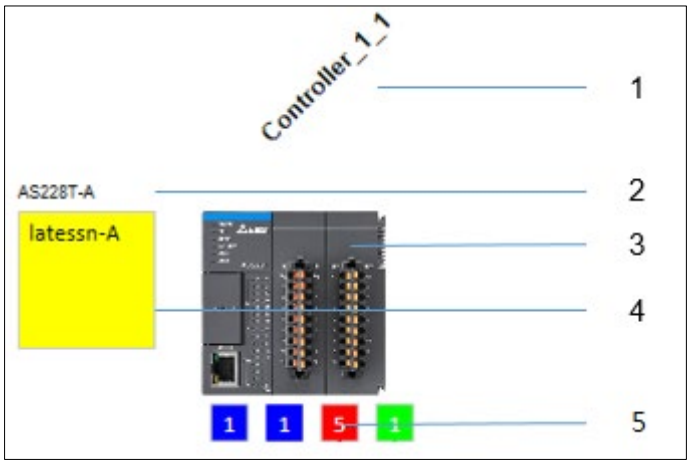
Figure 7 - 1: Network View

7.1.1 Network View Description

Network View displays two types of information:

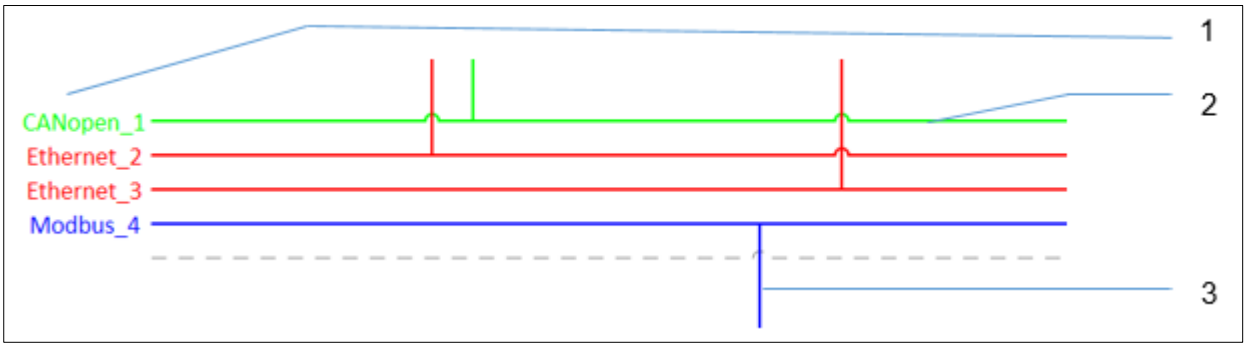
- Device information
- Network information

Device information includes:



1. Device name
2. Part number
3. Device image
4. A NOTE to add text
5. Communication ports on the device with protocol support.

Network information includes:



1. Name of the Network
2. Network lines with Colors

3. Connections to the Devices

When the network connection is made between the controllers, ports, field devices, the network lines appear in the color described in the following table:

Fieldbus or Network	Network line color
EtherNet/IP	Red
Modbus TCP	Red
Modbus	Blue
CANopen	Green

7.1.2 Creating Network View in Advanced View

Follow these steps to create a Network View in Advanced view:

1. Expand **Network Configuration** in the **Project**.
2. Double-click on the **Network View** as shown in the following figure.

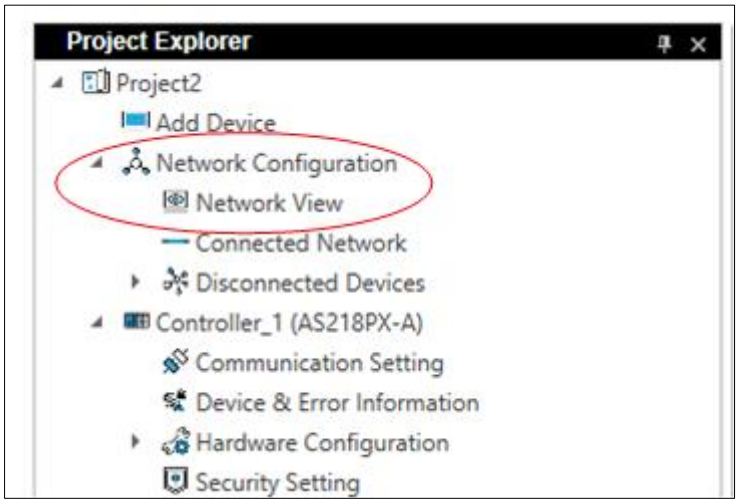


Figure 7 - 2: Network Configuration > Network View

Result: Displays **Network View** window as shown in the following figure. By default, dotted lines display, so that user can create connections between the devices/networks.

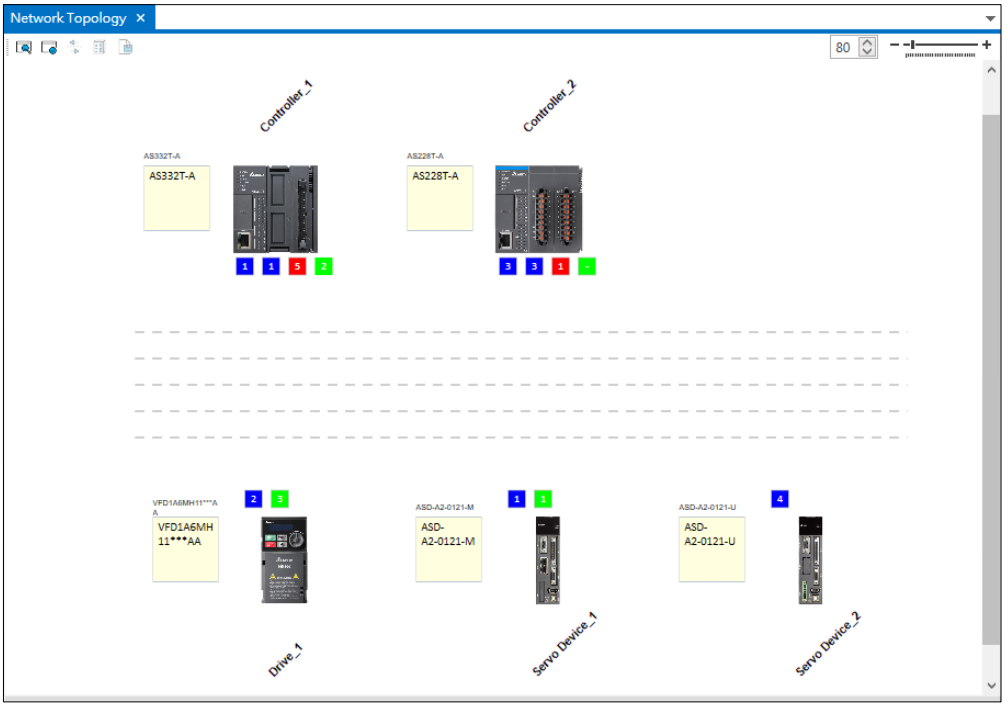


Figure 7 - 3: Network View

3. Drag and drop or double click from **Toolbox** area to add devices in the **Network View**.

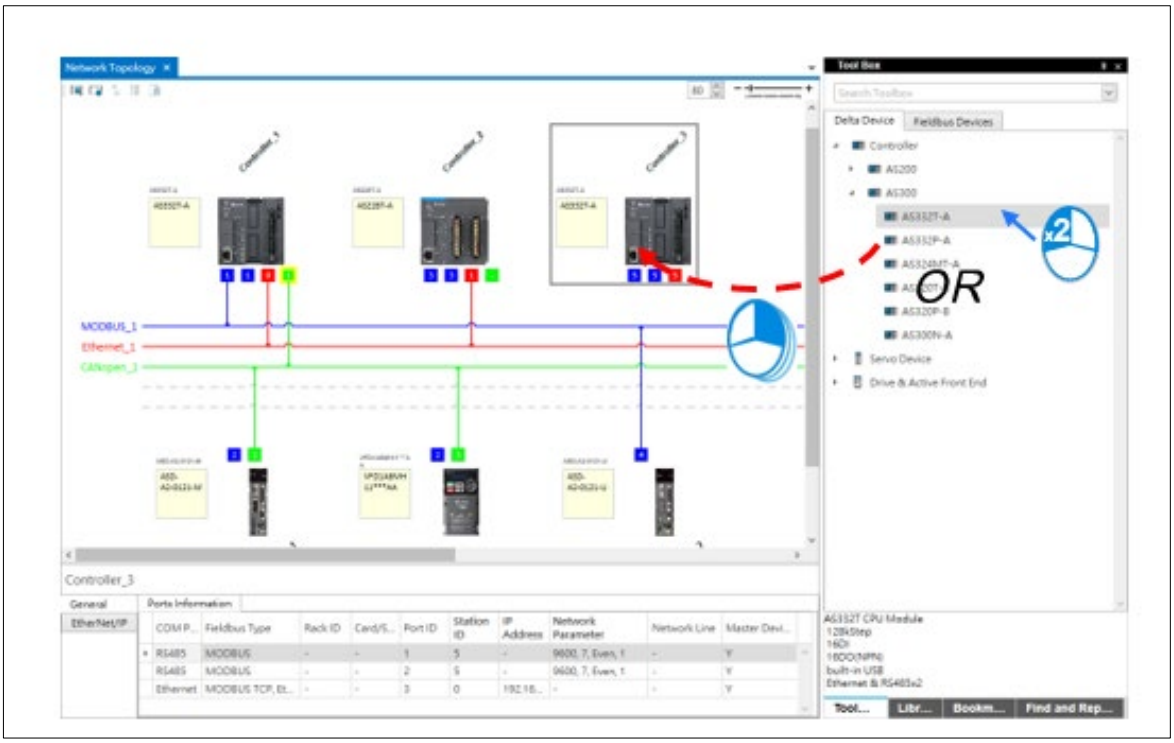


Figure 7 - 4: Adding device to Network

Result: The selected device is added to topology.

***NOTE**: User can also use **Add Device** in Project Explorer.*

4. Click on a communication port and connect it to a dotted line with drag & drop.

   Result: Dotted line will change to a line with same color as that of the device communication port (Protocol) selected as shown in *Figure 7-4: Adding device to Network*.

***NOTE**:*

- *User can also connect directly between two devices by clicking and dragging from one device port to another as shown in the following figure.*
- *If user double-click on the device in the Network View, it takes to **Hardware Configuration** window.*

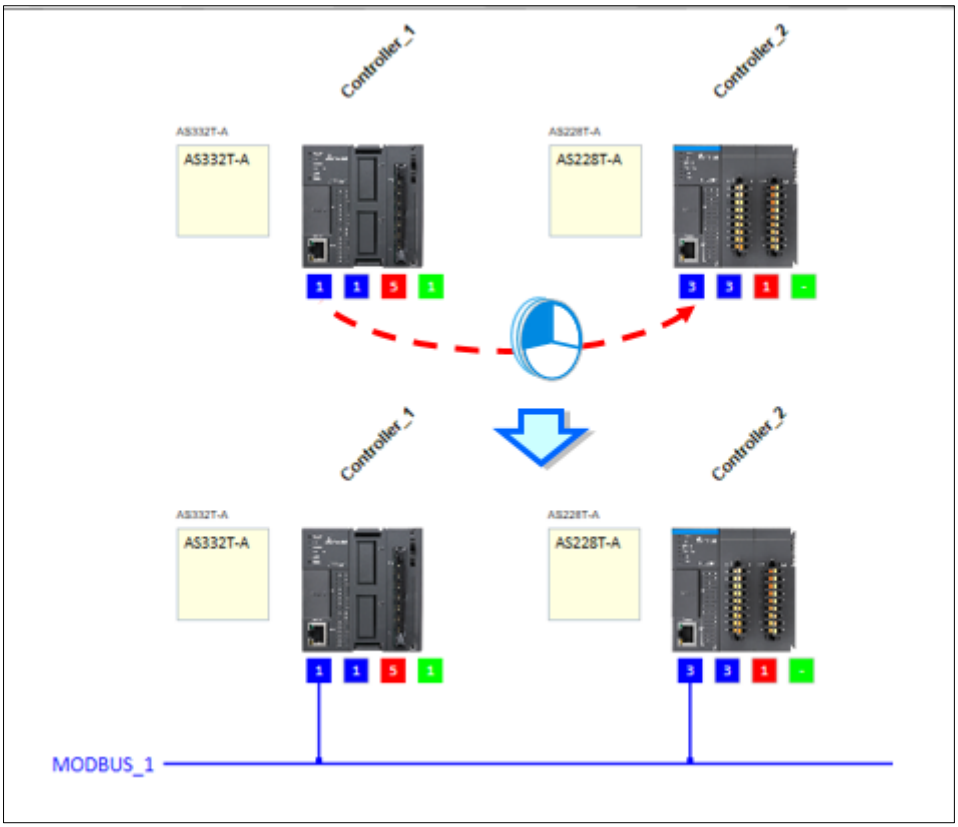


Figure 7 - 5: Network View - Example with MODBUS

7.1.3 Network View Toolbar

Toolbar Icon	Function
	Network Scan by PC
	Check Network for the related parameter settings
	Prominently remind the user of the current device connection
	Hand mode

7.1.3.1 Network Scan

User can click on (Network Scan) button to scan connected devices, the result will be listed in **Network Scan** dialog as shown in the following figure. User can select devices from the list and add it into the project.

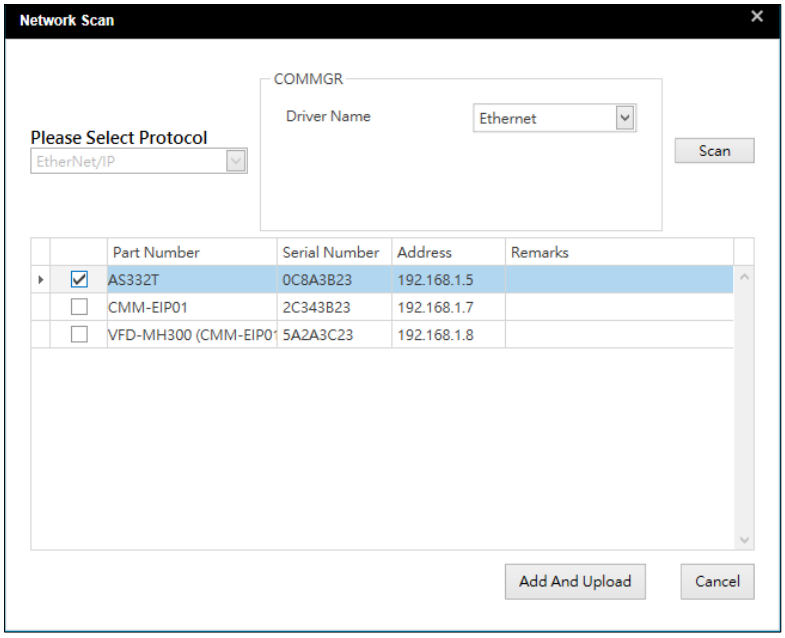


Figure 7 - 6: Network Scan

***NOTE**: EtherNet/IP is the only protocol supported in **Network Scan** for the present version.*

7.1.3.2 Network Check

User can click on (Network Check) button to check network setting correctness. If there is any error, the error symbol & error message displays. The error due to parameter mismatch of a network is shown in the following example figure.

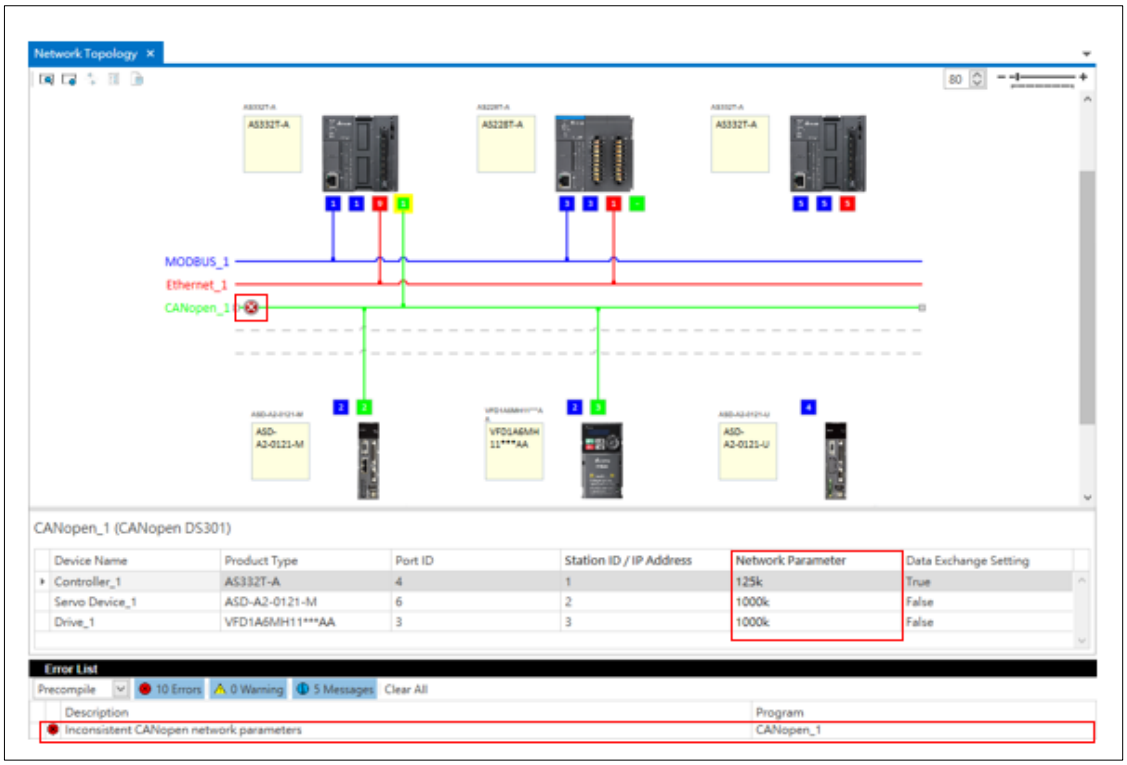


Figure 7 - 7: Network Check

7.1.3.3 Highlight Connections of User Selected Device

User can click the icon to active the function. After that, when user select a device, all the device connections and the connected devices will be highlighted, which is convenient for user to view the network view.

- Device: In bold blue box
- Bold horizontal/vertical network connection

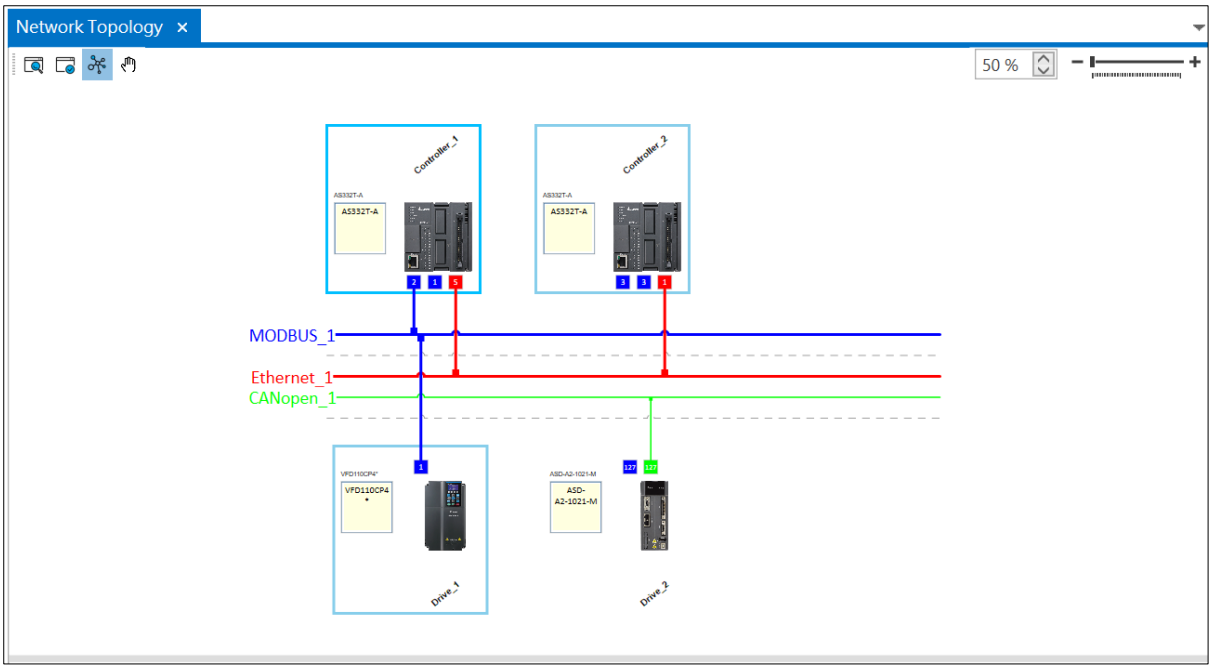


Figure 7 - 8: Highlight Connection

***NOTE**: user can click* *again to stop the function.*

7.1.3.4 Hand Mode

User can click the icon to active the function. When the cursor is an arrow, besides using scroll wheel, user can use scrollbar to move the canvas. When Hand Mode is active, user can directly click and drag in the blank space of the canvas to move the canvas as shown in the following figure.

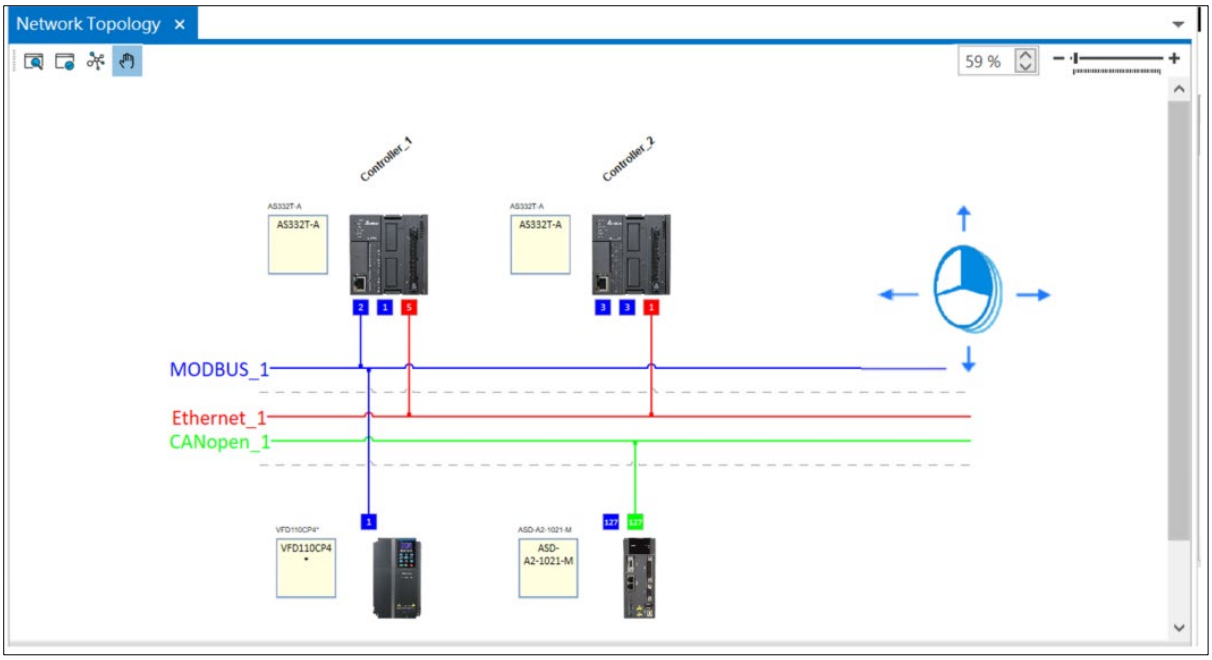


Figure 7 - 9: Hand Mode

***NOTE**: user can click* *again to stop the function.*

7.1.4 Network Parameter Setting

User can directly modify network parameters and IP addresses in the table at bottom of Network View. The modified values will be directly synchronized with the corresponding parameter values of parameter setting in hardware configuration.

7.2 Data Exchange Table

Data Exchange Table is a tabular tool for user to set up the data to be read/written from master to slave. With the help of data exchange table, user does not need to know the function code or detail protocol definition, DIADesigner will handle it.

7.2.1 Exchanging Data through Data Exchange Table

Follow these steps to exchange data in Advanced view:

1. Right-click to open context menu of a master device, and click on required protocol in **Data Exchange Table** as shown in the following figure.

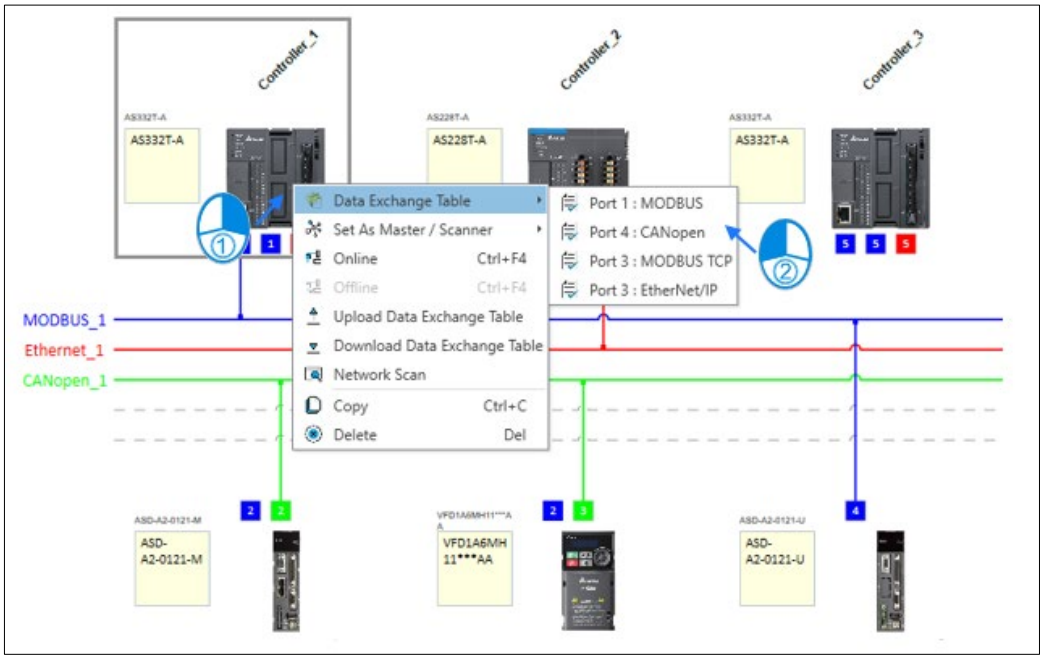


Figure 7 - 10: Data Exchange Table configuration

Result: A **Data Exchange Table** tab of the selected protocol opens, and the connected devices of this protocol displays in it. One row of default data exchange table is also created at the bottom of this tab as shown in the following table.

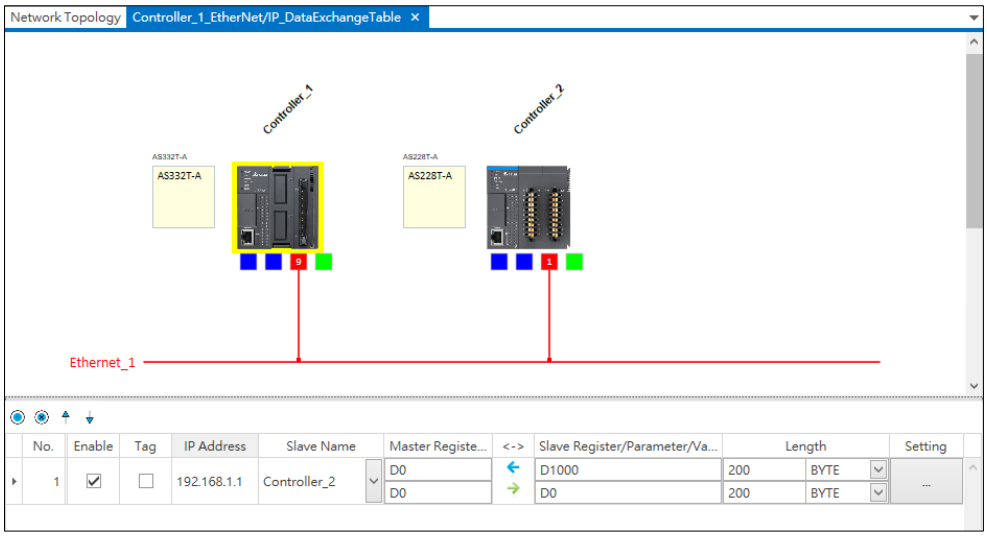


Figure 7 - 11: Data Exchange Table

2. Add or delete a row of data exchange table and then move up/down by clicking the respective buttons in the Data Exchange Table Toolbar as shown in the following table.

Toolbar Icon	Function
	Add
	Delete
	Up
	Down

3. Set up the parameters in each data exchange table. Refer the section *7.2.2 Setting up the Parameters in Data Exchange Table.*

4. To download/upload the data exchange table to/from the master device, right-click to open context menu and select **Download Data Exchange Table/Upload Data Exchange Table** as shown in Figure 7-12.

   Result: The **Upload/Download Manager** dialog displays, and the Configuration node including **Data Exchange Table** is selected to proceed with upload/download as shown in Figure 7-13.

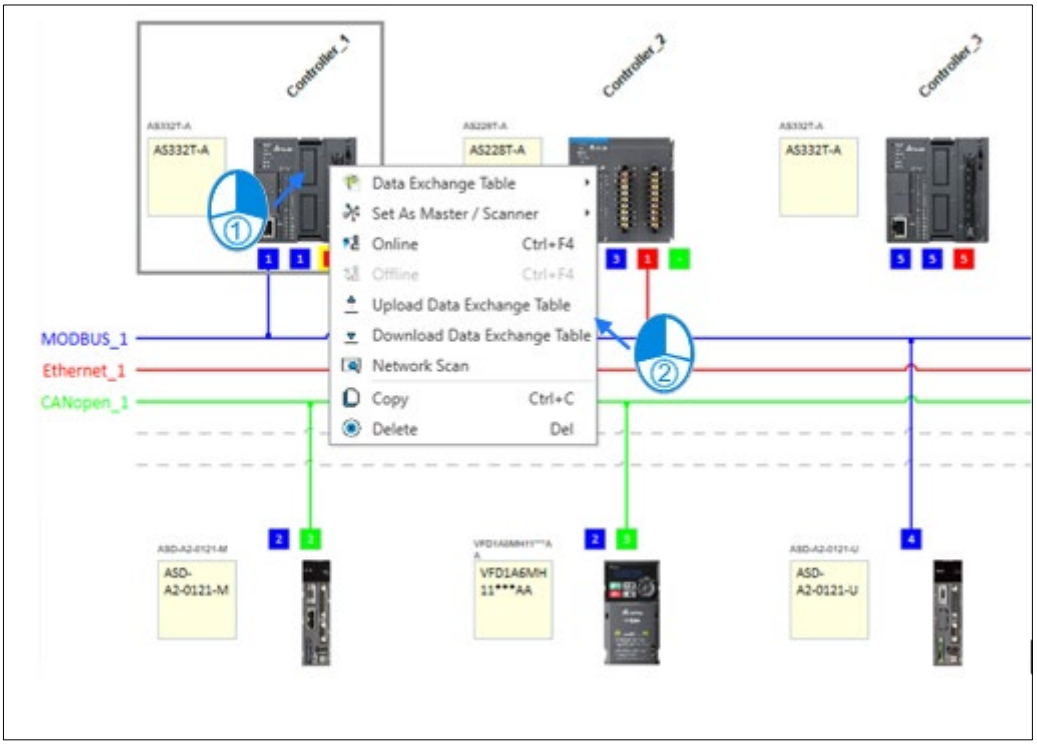


Figure 7 - 12: Upload/Download Data Exchange Table

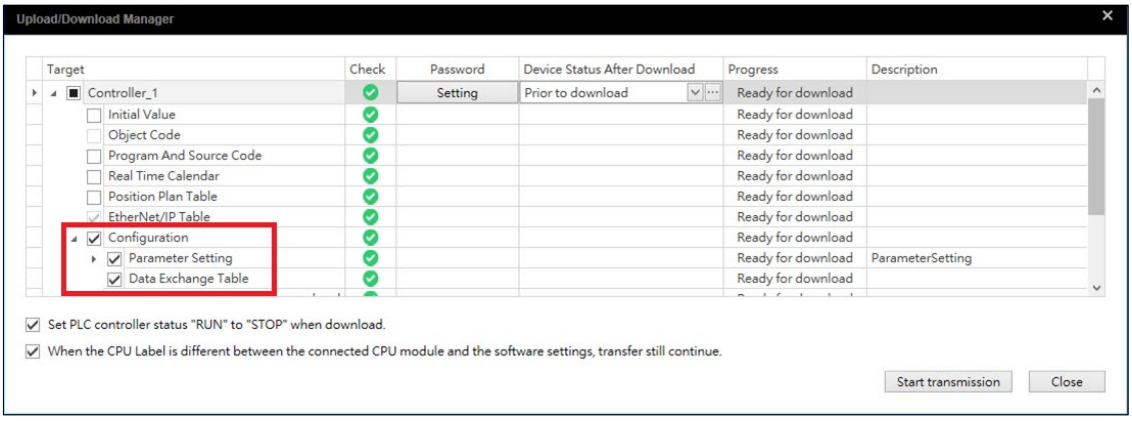


Figure 7 - 13: Upload/Download Manager

***NOTE**: User can also execute **Project Download** to open **Upload/Download Manager** dialog for **Data Exchange Table** download.*

7.2.2 Setting up Data Exchange Table

The examples of setting up the parameters in Data Exchange Table are given below:

Example 1:

Take Ethernet/IP data exchange table as an example, the value in D1000 of Controller_2 (192.168.1.1) is read into D0 of master, and the value in D0 of master is written into D0 of Controller_2.

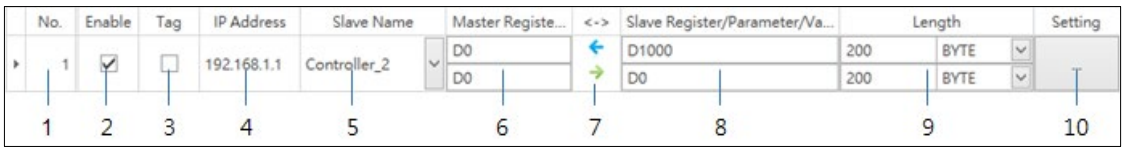


Figure 7 - 14: Data Exchange Table - Example 1

Example 2:

In CANopen protocol, the data exchange table means the PDO mapping between master and slave devices.

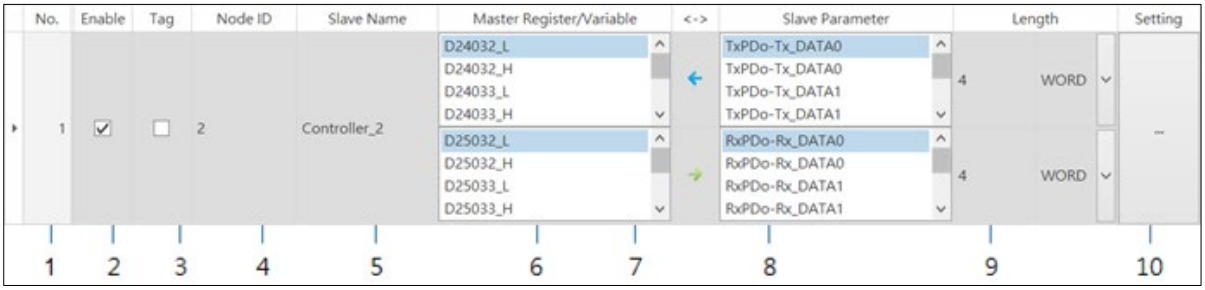


Figure 7 - 15: Data Exchange Table - Example 2

The functions in Data Exchange Table are described in the following table:

Function	Description
No.	The row number of the data exchange table.

Function	Description
Enable	Enables this row of data exchange table. It can be enabled or disabled by the SM register while PLC run.
Tag	Enables to map the variables in the specific global variable table. Only CANopen & EtherNet/IP supports this function.
IP Address/Node ID	Target slave IP Address.
Slave Name	Target slave device name.
Master Register/Variable	Target register or variable in master device.
<->	means read. means write.
Slave Register/Parameter/Variable:	Target register/parameter/variables in slave device.
Length	The data length to be transfer between master and slave.
Setting	Detail setting dialog. To set up parameters in for this data exchange table of the protocol. For detailed definition of each parameter, refer related technical documents.

7.2.2.1 Tag

Checking the Tag enables to map the variables in the specific global variable table.

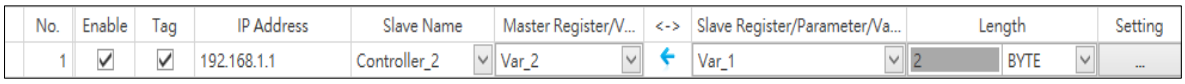


Figure 7 - 16: Tag

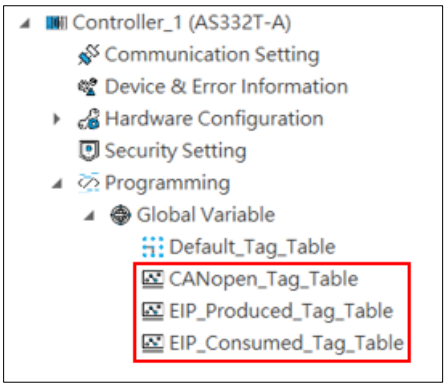


Figure 7 - 17: Global Variable Table

7.2.3 EtherNet/IP Data Exchange Table

The tag is enabled for EtherNet/IP only when both master & slave devices had already declared produced or consumed variable in **EIP_produced_Tag_Table** or **EIP_Consumed_Tag_Table**.

If the slave is 3rd party device or the Delta PLC which is not in this project, user can type variable name in **Slave Register/Parameter/Variable** directly.

Figure 7 - 18: 3rd party device or other PLC Slave Register/Parameter/Variable

Click the setting button to open the setting dialog of **EtherNet/IP Data Exchange Table** as shown in the following figure:

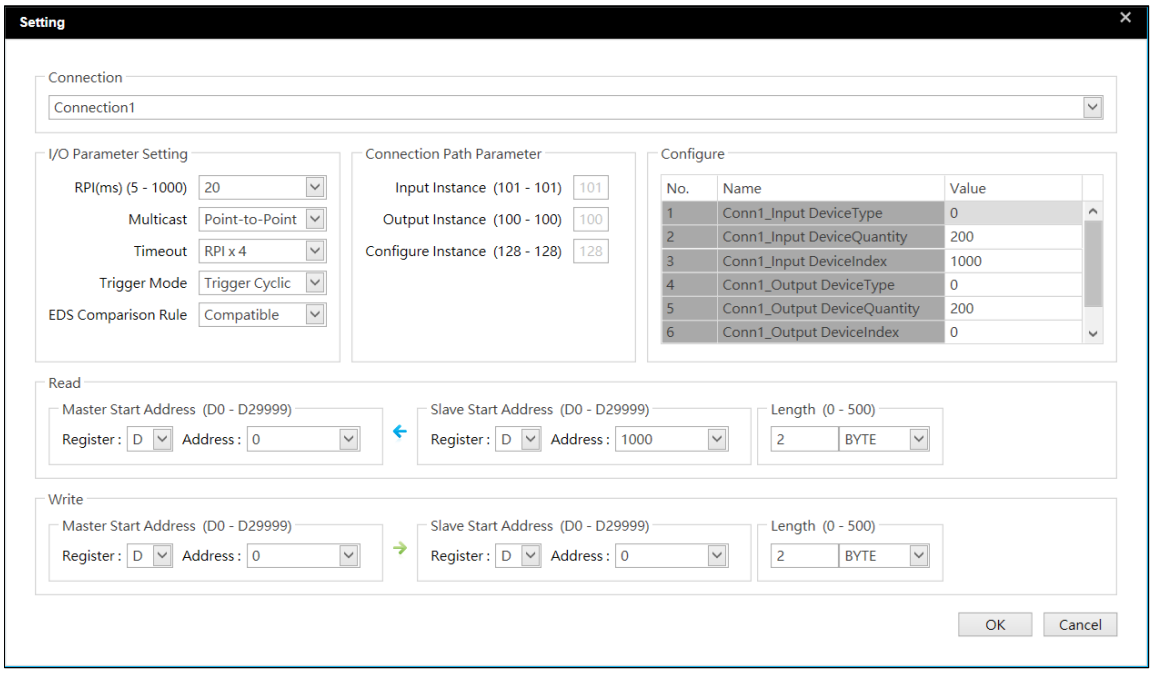


Figure 7 - 19: EtherNet/IP Data Exchange Table Setting

7.2.4 CANopen Data Exchange Table

User cannot declare variable in **CANopen_Tag_Table** directly. When user checks **Tag** box in CANopen data exchange table, the default variables are created in **CANopen_Tag_Table** automatically.

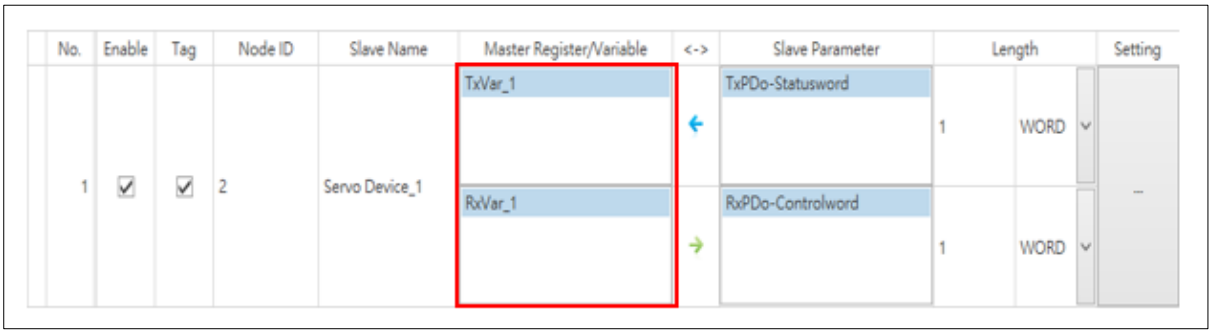


Figure 7 - 20: CANopen Data Exchange Table

Click the **Setting** button to open the **Setting** dialog of **CANopen Data Exchange Table** as shown in the following figure:

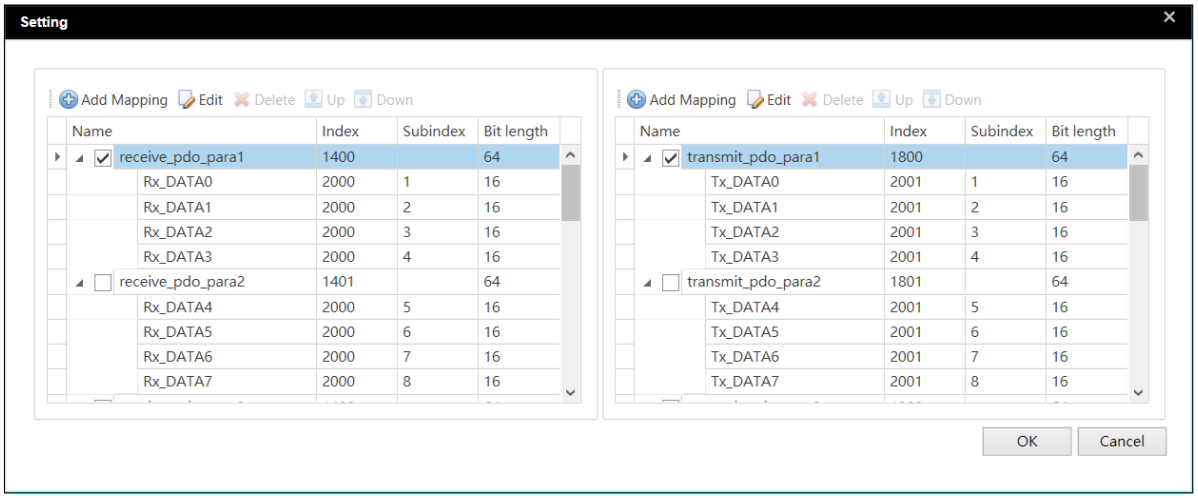


Figure 7 - 21: CANopen Data Exchange Table Setting

Click **Add Mapping** button, then add or delete object in each PDO pocket in the pop-up dialog.

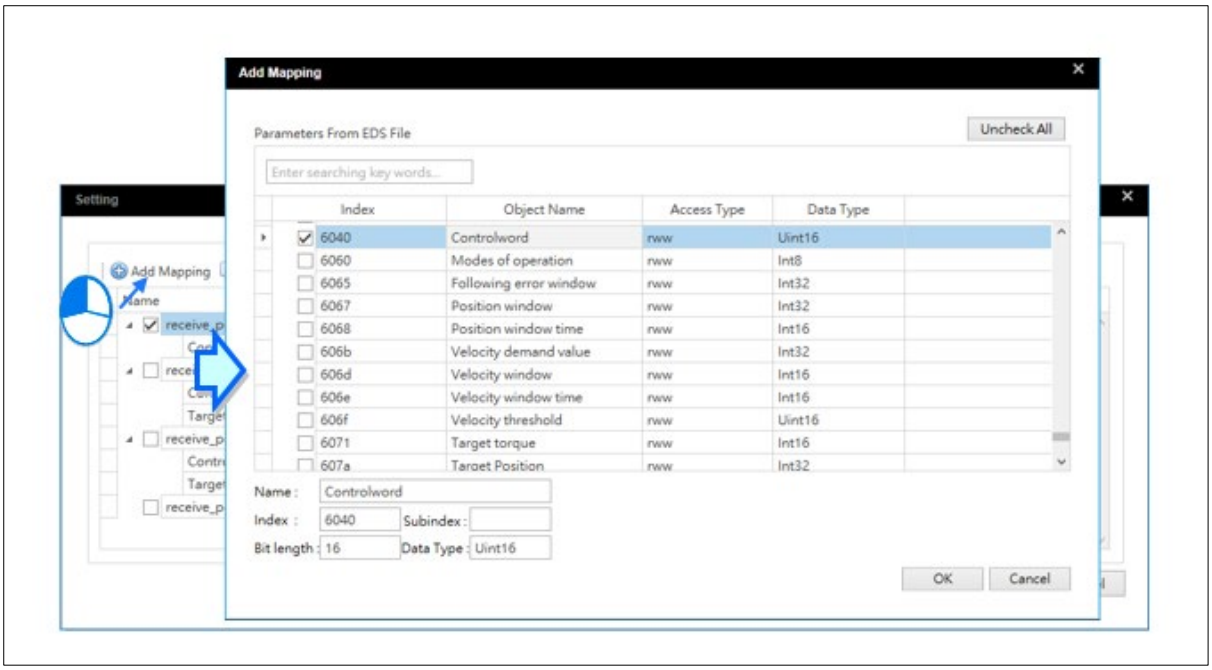


Figure 7 - 22: Add Mapping

7.2.5 MODBUS/MODBUS TCP Data Exchange Table

Modbus and Modbus/TCP do not support the function of designated variable exchange. The **Setting** dialog of **MODBUS/MODBUS TCP Data Exchange Table** is as shown in the following figure.

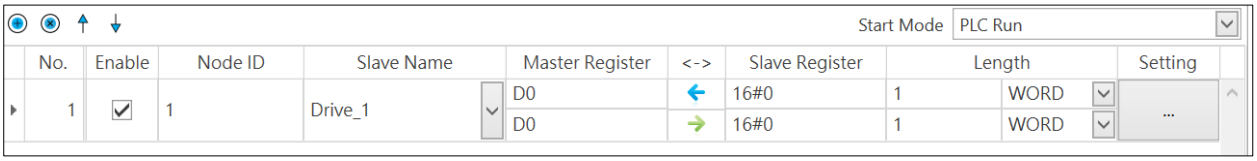


Figure 7 - 23: Modbus and Modbus/TCP Data Exchange Table

Click the **Setting** button to open the **Setting** window of Modbus and Modbus/TCP data exchange table.

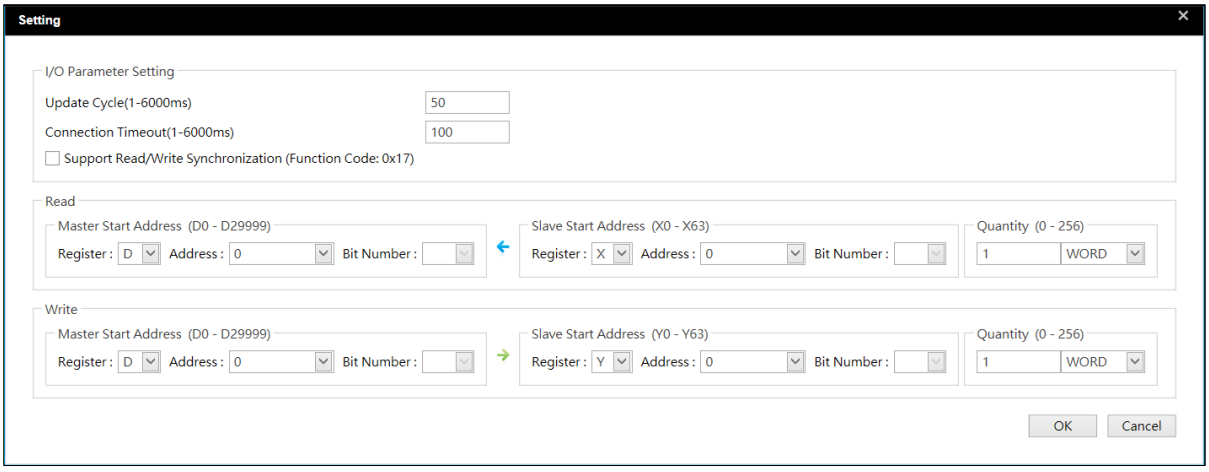


Figure 7 - 24: MODBUS/MODBUS TCP Data Exchange Table Setting

Chapter 8: Programming

8.1 IEC61131-3 Programming

IEC 61131-3, which was published by International Electrotechnical Commission (IEC), standardizes the characteristics of the PLC programming languages. The program codes which meet the standards are more readable and compatible, which greatly reduces user's learning curve. Based on the IEC 61131-3 standards, DIADesigner integrated programming languages that manufacturers and user could follow. The description and outline of IEC 61131-3 is as follows.

- IEC 61131-3 integrated the programming languages established by PLC developers and defined five programming languages for programming control systems:
  - Instruction List (IL)
  - Structured Text (ST)
  - Ladder Diagram (LD)
  - Sequential Function Chart (SFC)
  - Function Block Diagram (FBD)

  Besides, Continuous Function Charts (CFC) derived from the Function Block Diagrams(FBD) is also a common PLC Programming language.
- The concept related to Variables is adopted. User can replace an address with a Variable. A Program is more readable and the time of assigning addresses is saved.
- The architecture related to Program Organization Units (POU) 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 (POU) 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.

***NOTE**: Instruction List (IL), Function Block Diagram (FBD) and Continuous Function Chart (CFC) are not supported in DIADesigner, Version 1.0.*

8.2 POU, VAR, Task

8.2.1 POU

Program Organization Units (POUs) and Tasks are important concepts related to IEC 61131-3. They upgrade programming from traditional program writing to project management. The difference between traditional Ladder Diagram (LD) and IEC61131-3 are described in the following table:

Traditional Ladder Diagram	IEC 61131-3
Main Program	• A single program is divided into several independent Program Organization Units (POUs). • A POU created is assigned to a cyclic task
General Subroutine	• A subroutine is a function block • A function block created can be called by a POU
Interrupt Subroutine	• Different interrupt subroutines are independent POUs • A POU is assigned to a interrupt task

A comparison of traditional Ladder Diagrams and IEC 61131-3 is represented in *Figure 8-1: Comparison between Traditional Ladder Diagram and IEC 61131-3*. In the classic architecture, user write a source code. In the IEC 61131-3 architecture, a program is divided into several POUs. Every POU can be developed independently and can be assigned to a task.

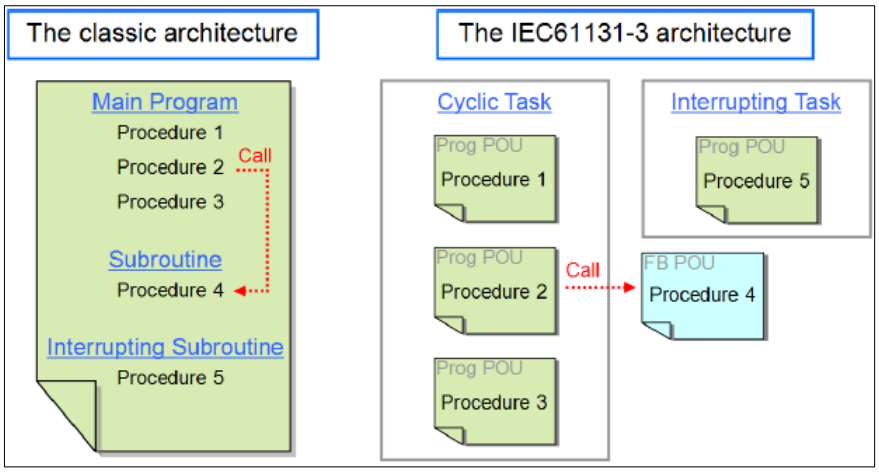


Figure 8 - 1: Comparison between Traditional Ladder Diagram and IEC 61131-3

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

8.2.1.1 POU architecture and types

Program Organization Units (POUs) are basic elements of a PLC Program. 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 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 the program is bigger, the maintenance and the debugging of the program becomes difficult. 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 the POUs are modular, different POUs can be developed by different designers. It benefits the distribution of manpower and the execution of the project.

There are two types of POUs in DIADesigner:

- Program
- Function Block

Program:

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 acts as a main program. If the POU is assigned to an interrupt task, the POU of the program acts as a subroutine. Besides, a POU of program type can call a function block (FB).

Function Block:

A static variable can be declared in a Function Block (FB). As a result, the value of the variable after an operation can be retained. Owing to the fact that the operation is performed on the value memorized in the function block and an input value, the output values may be different even if the input values are the same. Besides, a function block can call another function block.

The relation among POUs is shown in the following figure. A function (FC) is a type of POU. Owing to the fact that DIADesigner currently does not support Function POU, this part will not be covered in this chapter.

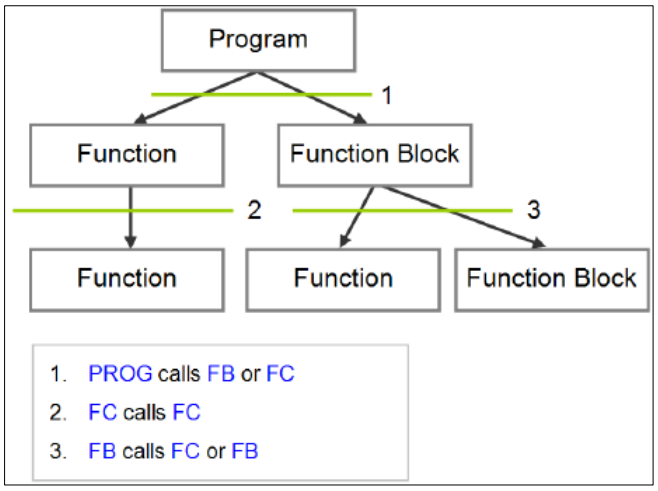


Figure 8 - 2: Function Block

POUs in DIADesigner

In DIADesigner, User can create/manage the POUs in the **Project Explorer**. 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.

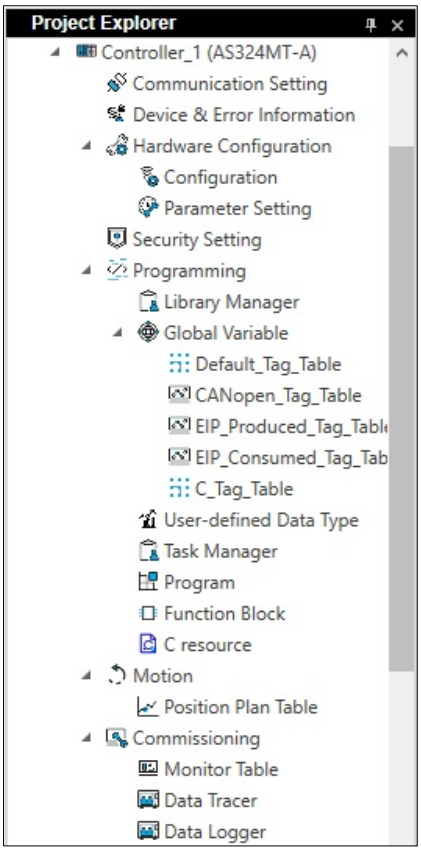


Figure 8 - 3: POUs in Project Explorer

Icon	Description
	POUs of Program types are organized here
	POUs of Function Block type are organized here
	POU is in Ladder Diagram (LD)
	POU is in Structured Text (ST)
	POU is in Sequential Function Chart (SFC)
	POU is in C language (C)

Upon double clicking on a Program or Function Block in the **Project Explorer** window, a POU editing window will appear. The POU editing window is composed of two parts:

A table of local variables is at the upper part of the window and the body of the 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.

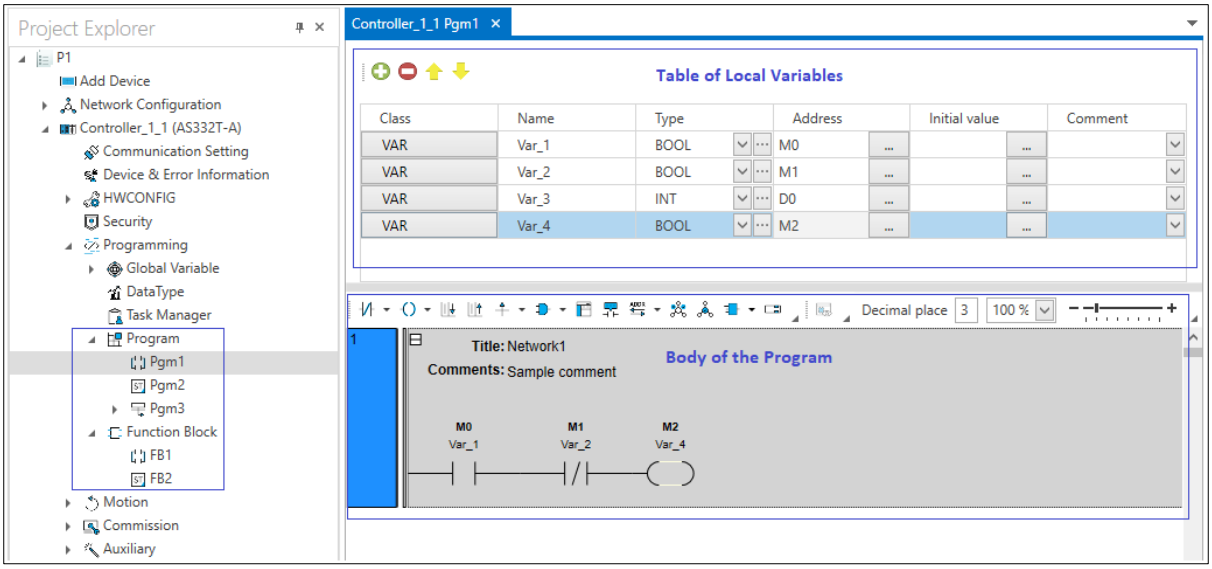


Figure 8 - 4: Programming

Refer below sections for more information about variable tables and program editing.

8.2.1.2 Managing a Program Organization Unit (POU)

Creating and setting a POU:

Program Type POU:

User can create program type POUs in **Project Explorer** > **Controller device** > **Programming** > **Program**.

Follow these steps to create a Program type POU:

1. Right click on the **Program**.

   Result: Context menu of Program displays.

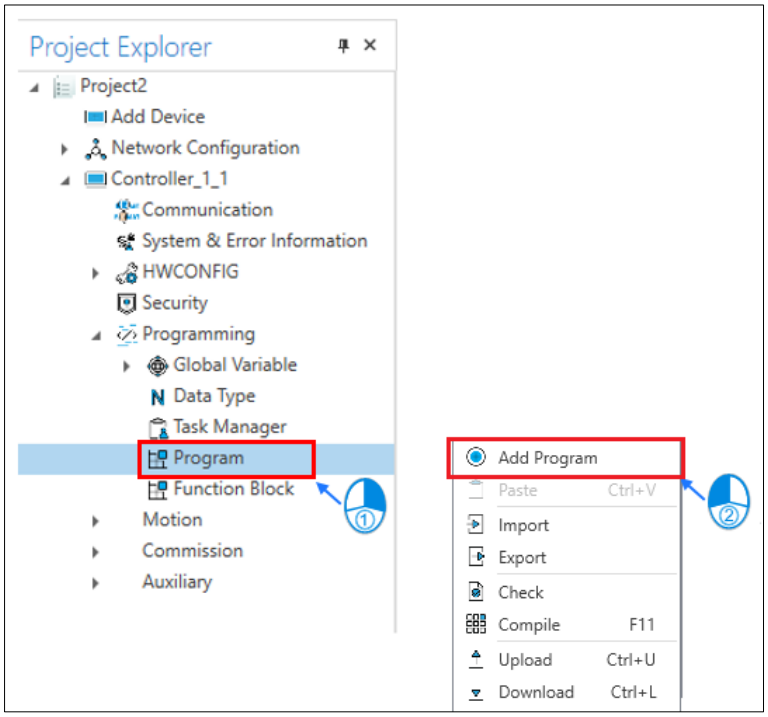


Figure 8 - 5: Program - context menu

2. Click on **Add Program** option.

   Result: Displays **Add Program** window.

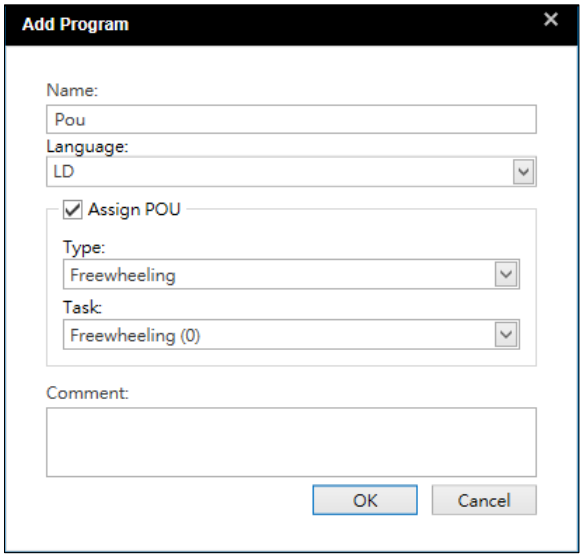


Figure 8 - 6: Add Program

The functions in the **Add Program** window is described in the following table:

Property/Icon	Description
POU name field	Enter a POU name.
Type	Select Program POU.
Language	Select Implementation language. Options are: • ST • LD • SFC • C ***NOTE**: **LD** is the default value.*
Comment	Enter the comments for POU.

3. Enter a POU Name, select program Type, a programming Language, enter Comment (optional) and click **OK**.

   Result: A Program type POU is created with the name entered and language selected

Function Block Type POU:

User can create function block type POUs in **Project Explorer** > **Controller device** > **Programming** > **Function Block**.

Follow these steps to create a Function Block type POU:

1. Right click on the **Function Block**.

   Result: Context menu of function block displays.

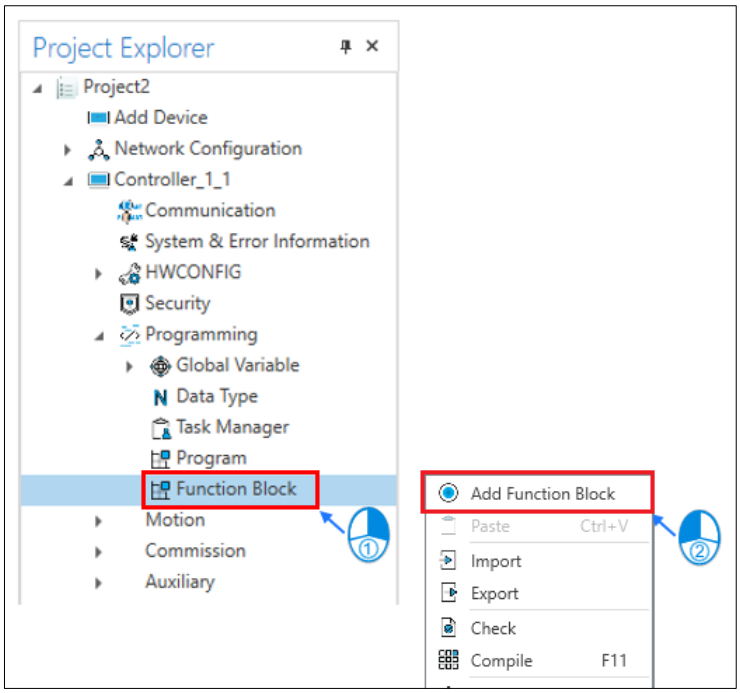


Figure 8 - 7: Function Block - context menu

2. Click on **Add Function Block**

   Result: Displays **Add Function Block** window.

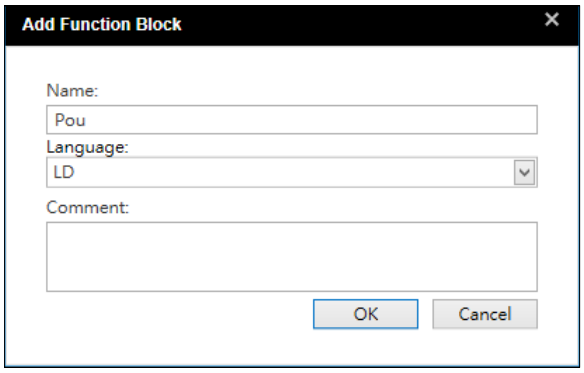


Figure 8 - 8: Add Function Block

3. Enter a POU **Name**, select a programming **Language**, enter **Comment** (optional) and click **OK**

   Result: A Function Block type POU is created with the name entered and language selected. User have to set the attributes of the POU added in the

Add Function Block window which appears. The attributes of the POU of the program type is slightly different from those of the POU of the function block type and they are described in the following table:

Property/Icon	Description
POU name field	Enter a POU name.
Language	Select Implementation language. Options are: • LD • ST • C ***NOTE**: **LD** is the default value.*
Comment	Enter the comments for POU.

POU Name has the following characteristics:

1. Special characters and spaces cannot be used. For example, *, #, ?, \, %, @ etc. cannot be used.
2. A POU name is composed of thirty characters at most. User have to notice that a Chinese character occupies 2 characters.
3. Underline can be used, but they cannot be used continuously or used at the end of a POU name. For example, POU_1 is a valid name, but POU__1 and POU_1_ are invalid names.

POU properties:

User can view the properties of POUs by *POU name* > *context menu* > **Properties**.

Result: **POU Properties** window displays as shown in the following figure.

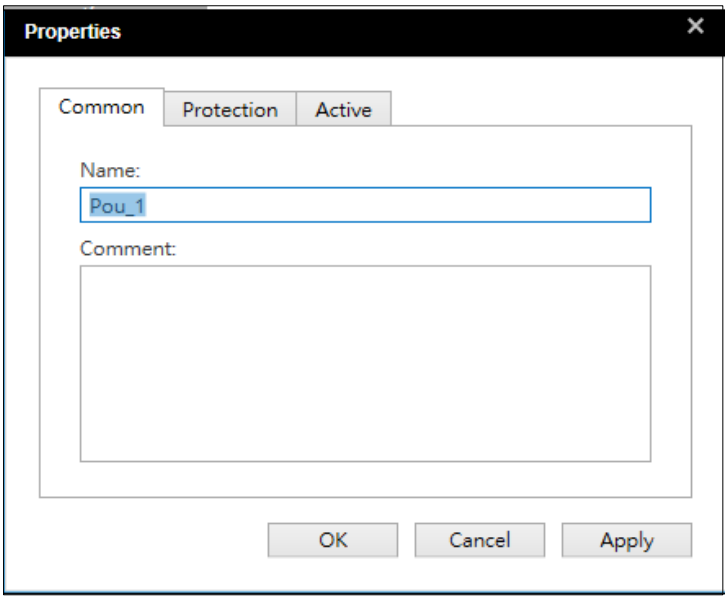


Figure 8 - 9: POU Properties

The **POU Properties** window displays the properties in three different tabs. POU Properties are described in the following table:

Tab	Property	Description
Common	Name	Edit the program name.
	Comment	Edit the comment for program.
Protection	Please enter password (4 ~ 16 character)	Enter a password.
Password	Reconfirm	Enter the password again.
	Show Password	Select to enable/disable the visibility of characters in the password.
Active	Active check box	Select to enable/disable the program type POU.
	OK	Click to save the settings.
	Cancel	Click to cancel the changes made.
	Apply	Click to save the settings and close the window.

Enabling a POU:

DIADesigner provides the function of disabling a POU temporarily. If a POU is disabled, it is skipped when the program is compiled and the program in the POU is not executed. However, only POUs of the program type can be disabled.

This function can be applied to the test or debug a program. If user want to skip certain procedure during the test or debug of the program, they can disable the POU. After the test or debug of the program is complete, they can enable the POU.

Example:

The equipment shown in the following figure is composed of five workstations and the working practices are divided into five POUs of the program type.

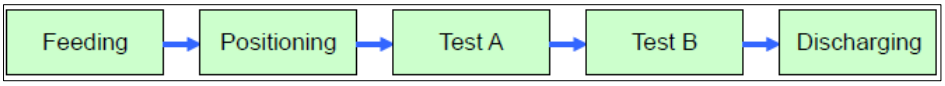


Figure 8 - 10: Process 1

If the assembling of the equipment is complete, but the assembling of the test instrument is not complete, user can disable the POUs corresponding to the Test A and Test B. the two workstations are skipped when the other parts of the equipment are adjusted.

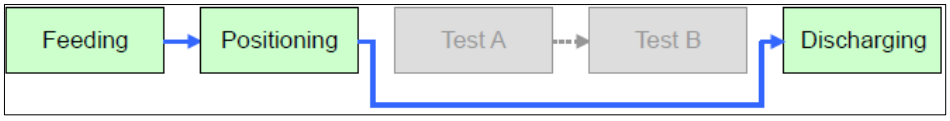


Figure 8 - 11: Process 2

If user want to enable a POU, they can select the **Active** checkbox in the **Properties** window. If the Active checkbox is not selected, then the POU is disabled. A POU which is disabled is represented by a gray icon.

Deleting a POU:

User can delete a POU of Program type or Function Block type. But the POUs delete cannot be recovered. When a POU is deleted, it is removed from the Task associated earlier.

User can delete a POU in the **Project Explorer** by any of these methods:

- Right click on the **POU** to open the context menu and click **Delete**

  or

- Select the **POU** to be deleted and press **Delete** key in the keyboard

  or

- Select the POU to be deleted and click Delete icon in the DIADesigner toolbar

  or

- Select the POU to be deleted and click on **Edit** > **Delete** in the menu item.

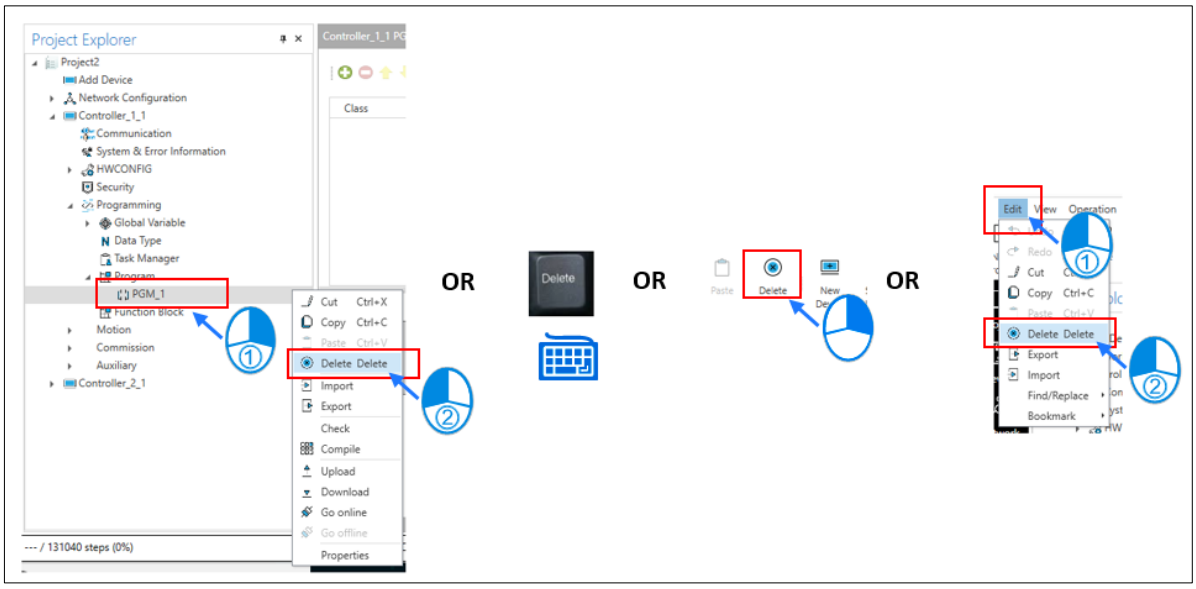


Figure 8 - 12: Delete POU

Result: POU is deleted.

Copying a POU:

User can copy a POU in the **Project Explorer** by any of these methods:

- Right click on the **POU** to open the context menu and click on **Copy**

  or

- Select the **POU** to be copied and press **Ctrl** + **C** keys simultaneously in the keyboard

  or

- Select the POU to be copied and click Copy icon in the DIADesigner toolbar

  or

- Select the POU to be copied and click on **Edit** > **Copy** in the menu item

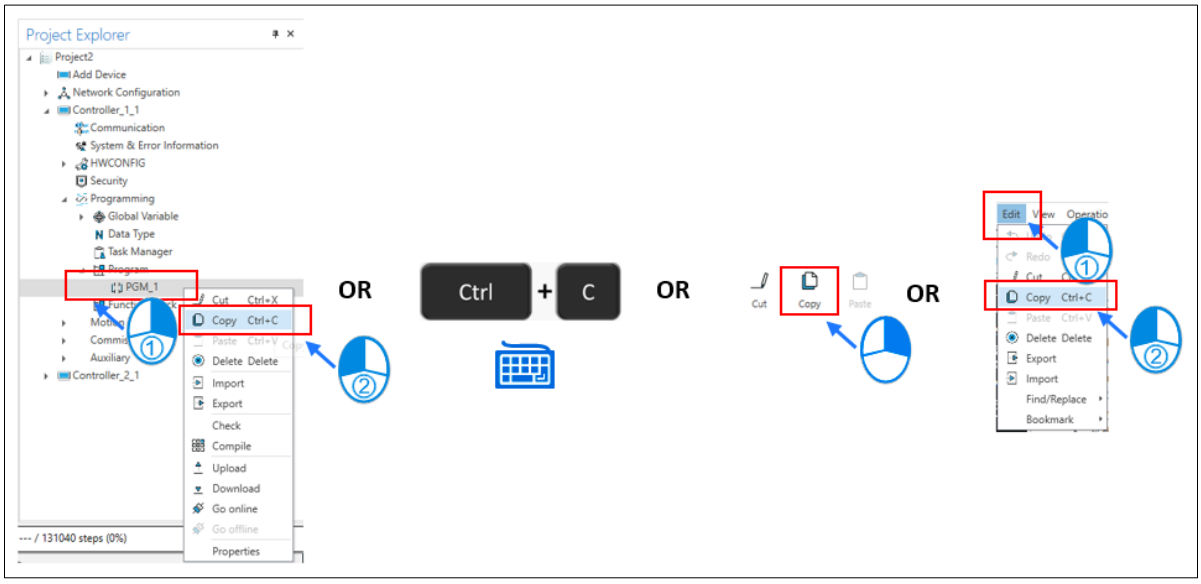


Figure 8 - 13: Copy POU

Result: POU is copied.

Pasting a POU:

User can paste a POU in the **Project Explorer** by any of these methods:

- Right click on the **Program** or **Function Block** to open the context menu and click on **Paste**.

  or

- Select the **Program** or **Function Block** and press **Ctrl** and **V** keys simultaneously in the keyboard.

  or

- Select the **Program** or **Function Block** and click on Paste icon in the DIADesigner toolbar.

  or

- Select the **Program** or **Function Block** and click on **Edit > Paste** in the menu item.

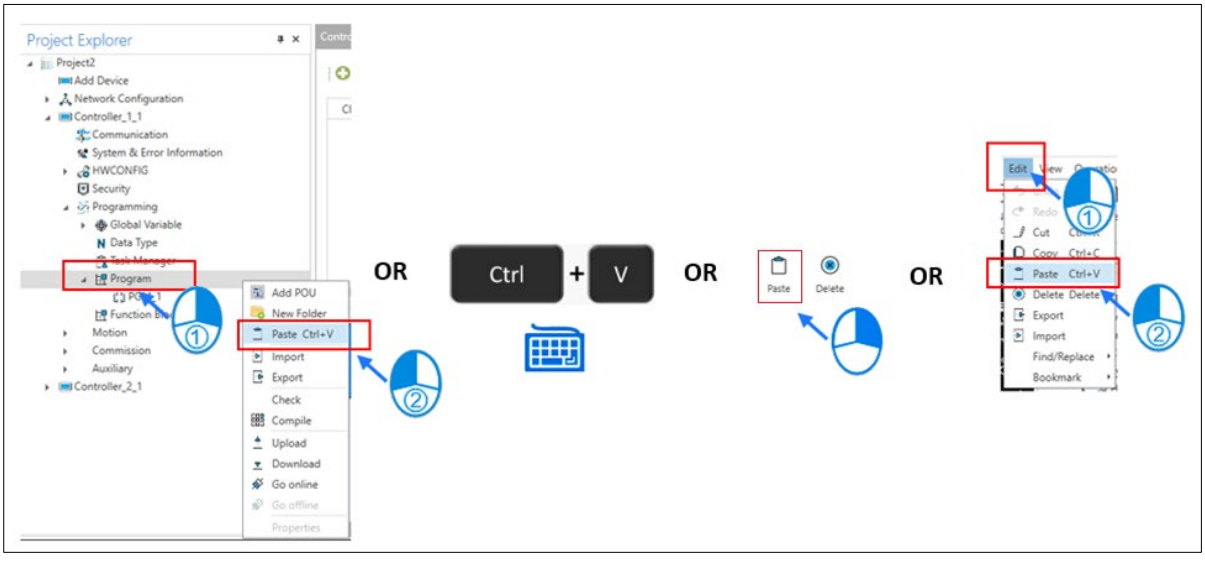


Figure 8 - 14: Copy POU

Result: POU is pasted.

***NOTE**: A POU can be copied in to the same or different device in the project. If the POU copied is a POU of the program type, it cannot be pasted in to the Function Blocks. If the POU copied is POU of the function block type, it*

cannot be pasted in to the Program. Besides, if the POU copied is a POU of the program type, the POU pasted will not be assigned to any task.

Setting and unlocking a POU Password:

User can set a password for a POU in DIADesigner. The setting and unlocking the POU password is described below:

When user create a POU, they can set a password for the POU. If user do not set a password for the POU during the time of its creation, they can protect the POU with a password later by opening the Properties window by clicking the **Properties** option in the context menu of the POU.

Follow these steps to set a password to POU:

1. Right click on the POU on which password must be set.

   Result: Context menu of the POU displays.

2. Click **Properties** in the context menu.

   Result: The **Properties** window displays.

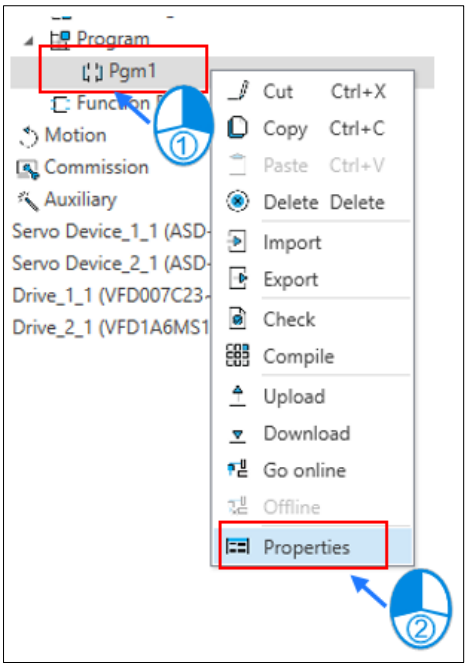


Figure 8 - 15: POU Properties

3. Type the same password in the **Please enter password** box, then in the **Reconfirm** box and click **OK/Apply** button.

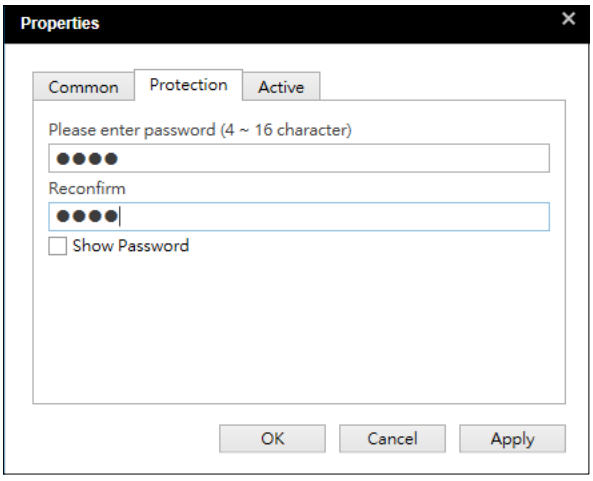


Figure 8 - 16: Properties

***NOTE**: English letters, numeric digits, special characters can be typed in the boxes. Once the POU is protected with password, the systems ask the user to type the password whenever the POU window is opened.*

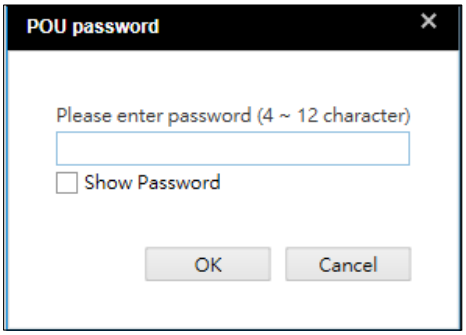


Figure 8 - 17: POU password

If user want to unlock the password, they have to open the **Properties** window again, type the password in the **Please enter the password** box and click **OK**.

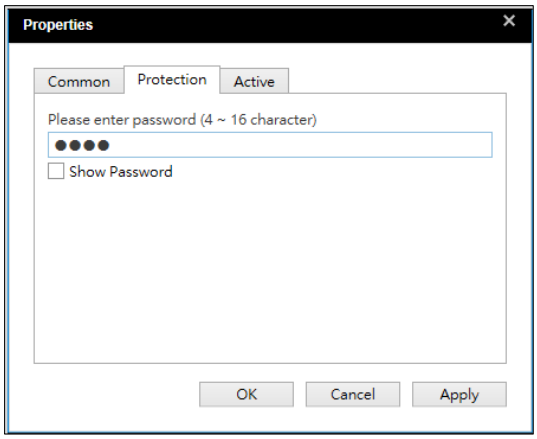


Figure 8 - 18: Unlock POU

Exporting a POU of program type:

DIADesigner supports the function of exporting POUs of program type.

Follow these steps to export a POU of program type:

1. Right click on the POU to be exported.

   Result: Context menu of the POU displays.

2. Click on **Export** in the context menu.

   or

   Click on **Edit** > **Export** in the menu item.

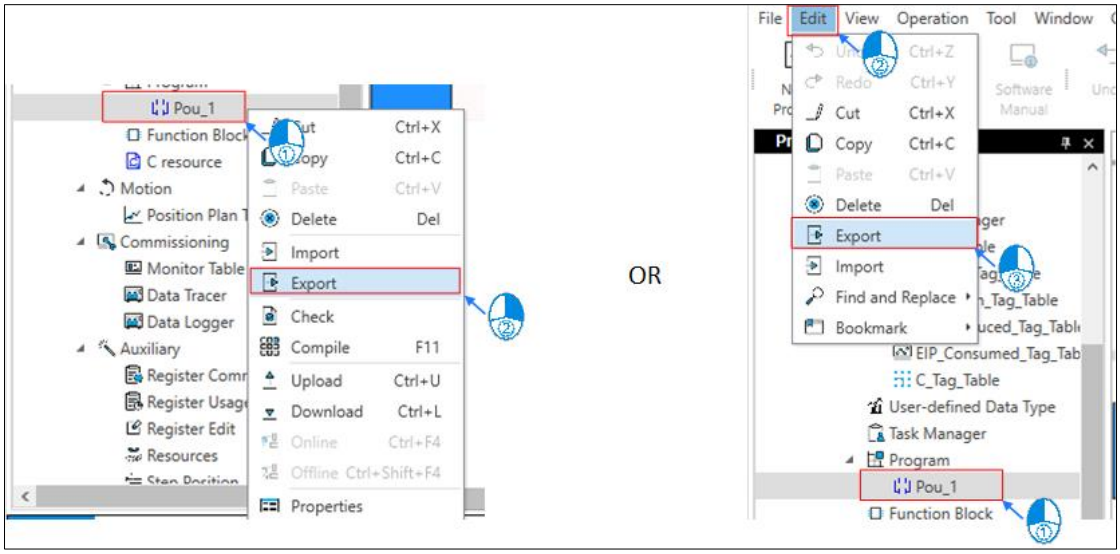


Figure 8 - 19: POU Export

Result: The **Export** window displays as shown in the following figure. POU to be exported is selected by default.

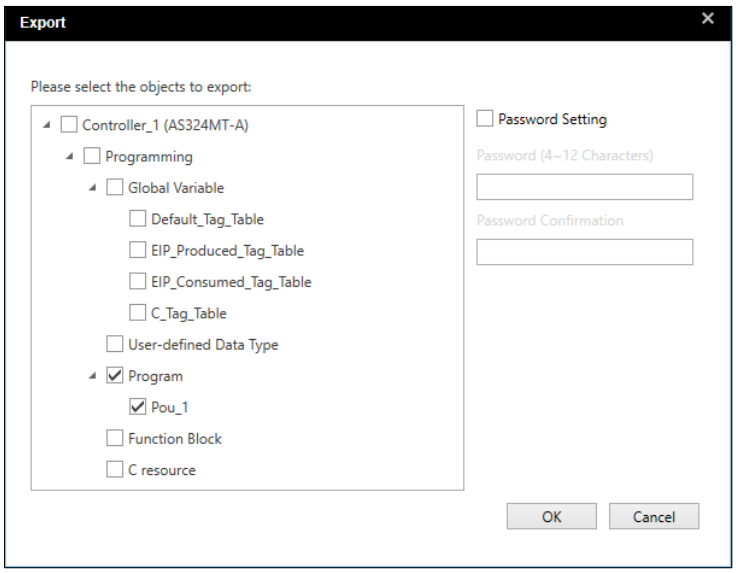


Figure 8 - 20: POU Export

3. Select the object(s) to export. Click **OK**. If User want to protect a file to be exported with a password, they have to select the **Password setting** checkbox, enter the password in **Password** box and enter the same once again in **Confirmation** box.

   Result: The **Save As** window displays.

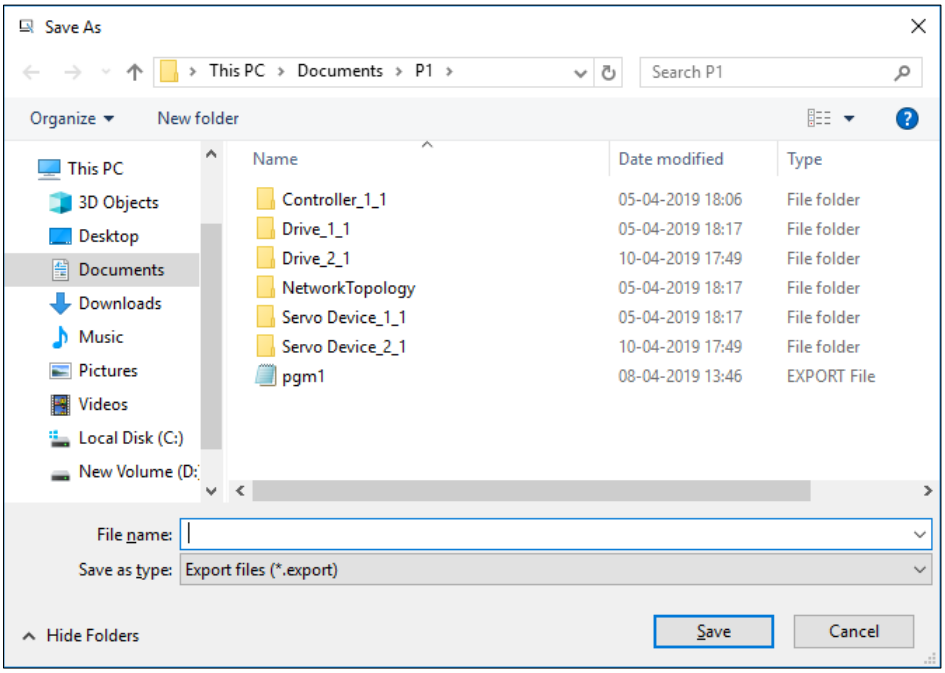


Figure 8 - 21: Save As Window

4. Select a particular location and click on **Save**.

   Result: The **POU** is exported.

Importing a POU of program type:

DIADesigner supports the function of importing POUs of program type.

Follow these steps to import a POU of program type:

1. Select a device in the **Project Explorer** to which the program type POU has to be imported.

2. Right click on the **Program** in the Device to open the context menu and click on **Import**

   or

   Click on **Edit** > **Import** in the menu item.

   Result: The **Open** window displays as shown in Figure 8-21.

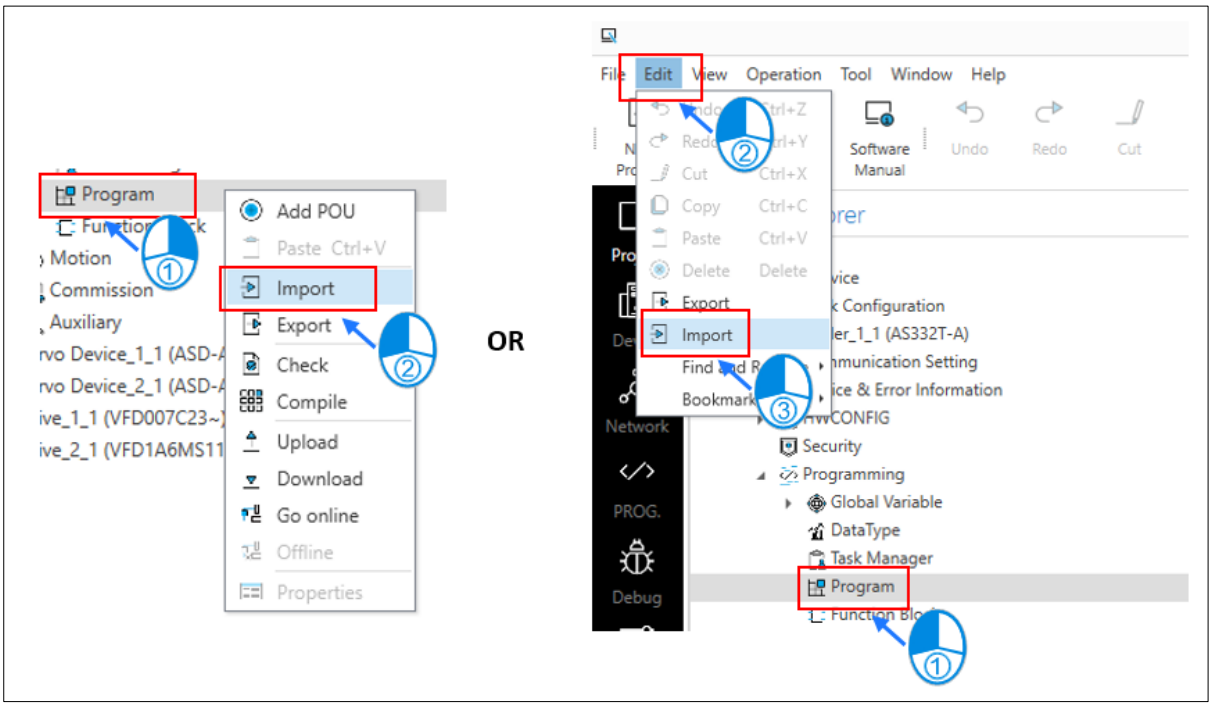


Figure 8 - 22: POU Import

3. Select the file to be imported and click **Open** button.

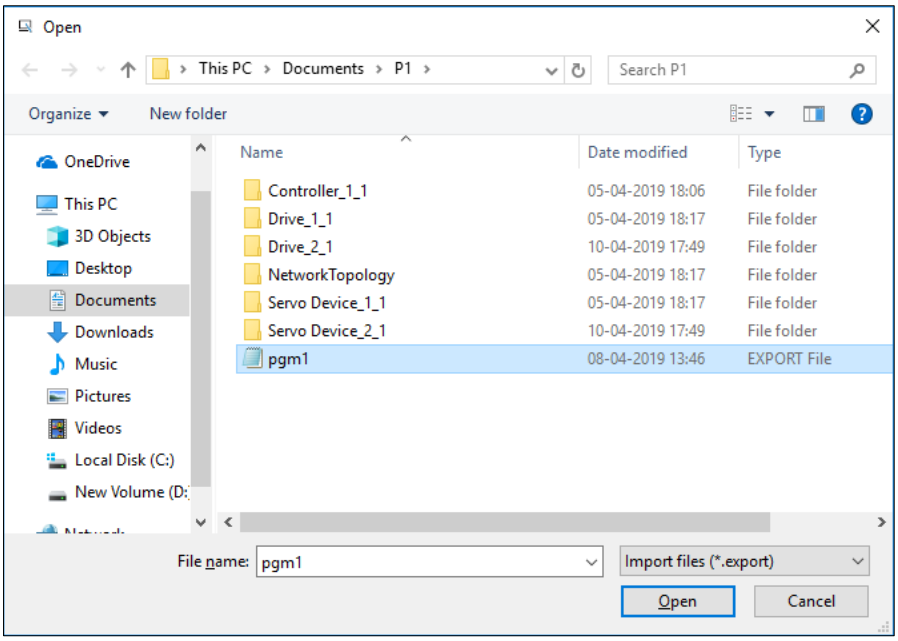


Figure 8 - 23: Open Window

Result: **Import** window displays as shown in the following figure.

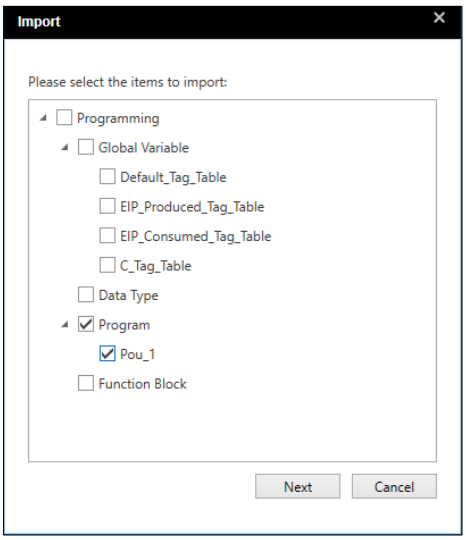


Figure 8 - 24: Import

4. Select the items to be imported and click **Next** button.
5. Select the device in which the POU should be placed and click OK.

   Result: **POU** is imported.

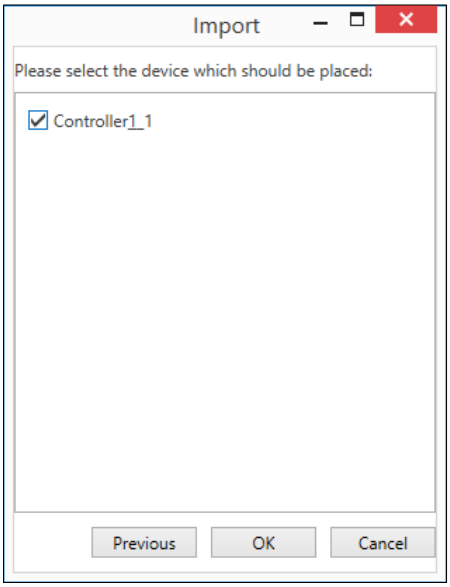


Figure 8 - 25: Import – POU

NOTE: If the file is protected with a password, user have to type the password in the **Password** for decrypting window and click **OK**. The POUs which are imported will not be assigned to any tasks. Besides, if the name of the POU which is to be imported is the same as the name of a POU in the project, it will prompt to rename or replace for proceeding.

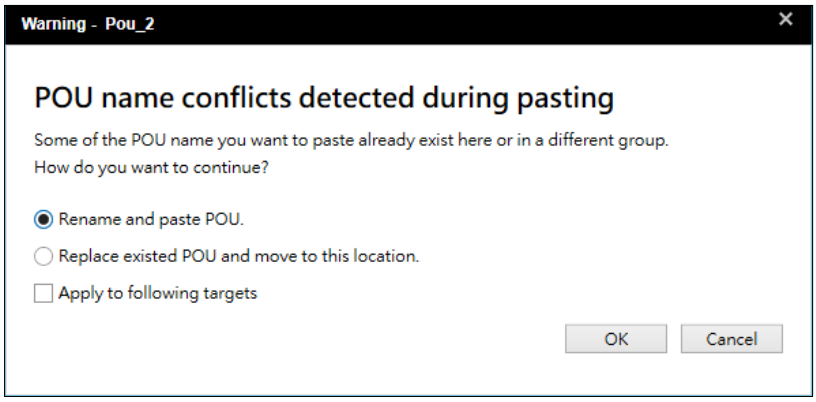


Figure 8 - 26: POU Import - naming conflict

Exporting a POU of Function Block type:

DIADesigner supports the function of exporting POUs of function block type.

Follow these steps to export a POU of function block type:

1. Select the function block type POU to be exported.
2. Right click on the **POU** to open the context menu and click on **Export**

   or

   Click on in the **Edit** > **Export** menu item.

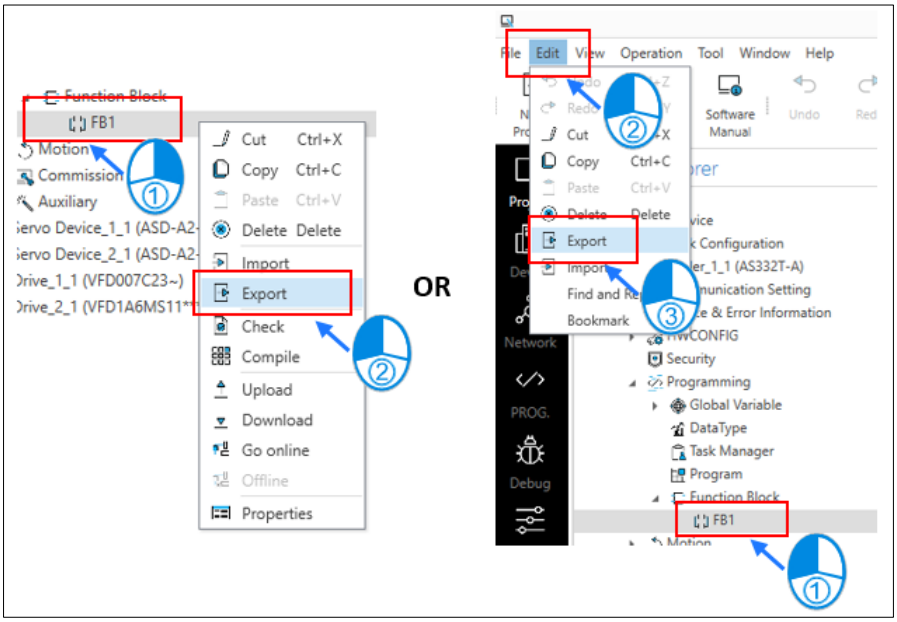


Figure 8 - 27: Function Block – Export

Result: The **Export** window displays as shown in Figure 8-26. POU to be exported is selected by default.

3. Select the objects to export. Click on **OK** button.

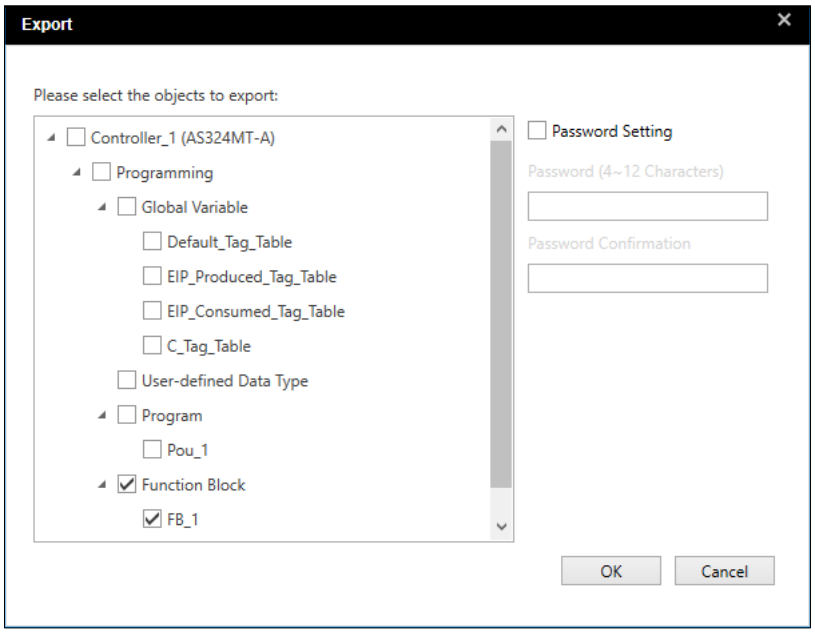


Figure 8 - 28: Function Block - Export

***NOTE**: If user want to protect a file to be exported with a password, they have to select the **Password setting** checkbox, enter the password in **Password** box and enter the same once again in **Confirmation** box.*

Result: The **Save As** window displays.

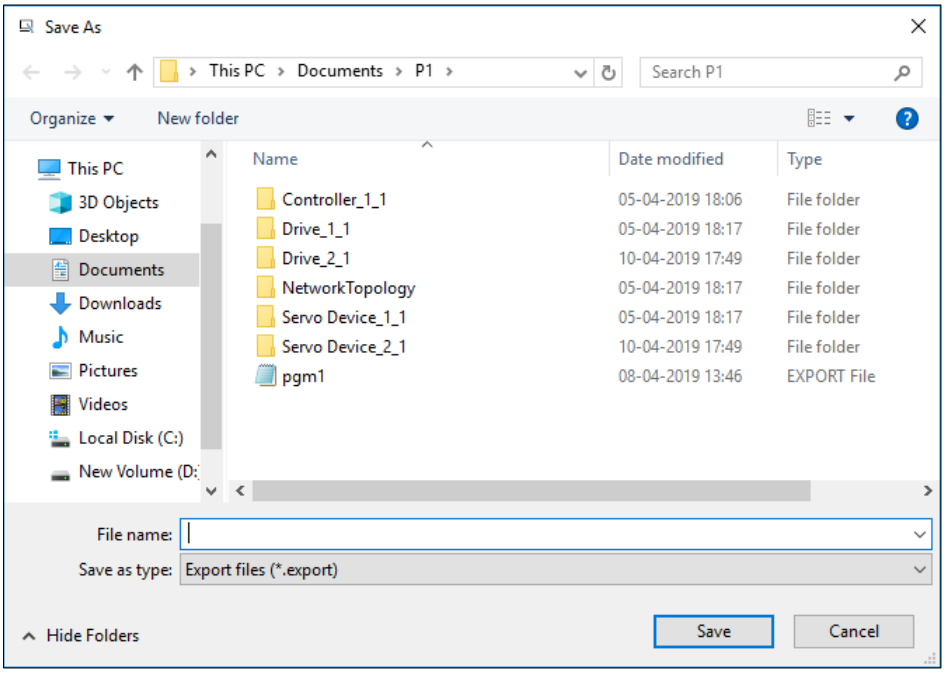


Figure 8 - 29: Save As Window

4. Select a particular location and click **Save**.

   Result: The **POU** is exported.

Importing a POU of the Function Block type:

DIADesigner supports the function of importing POUs of function block type.

Follow these steps to import a POU of function block type:

1. Select a device in the Project Explorer to which the function block type POU has to be imported.
2. Right click on the **Function Block** in the Device to open the context menu and click on **Import**.

   or

   Click on **Edit** > **Import** in the menu item.

   Result: The **Open** window displays as shown in Figure 8-29.

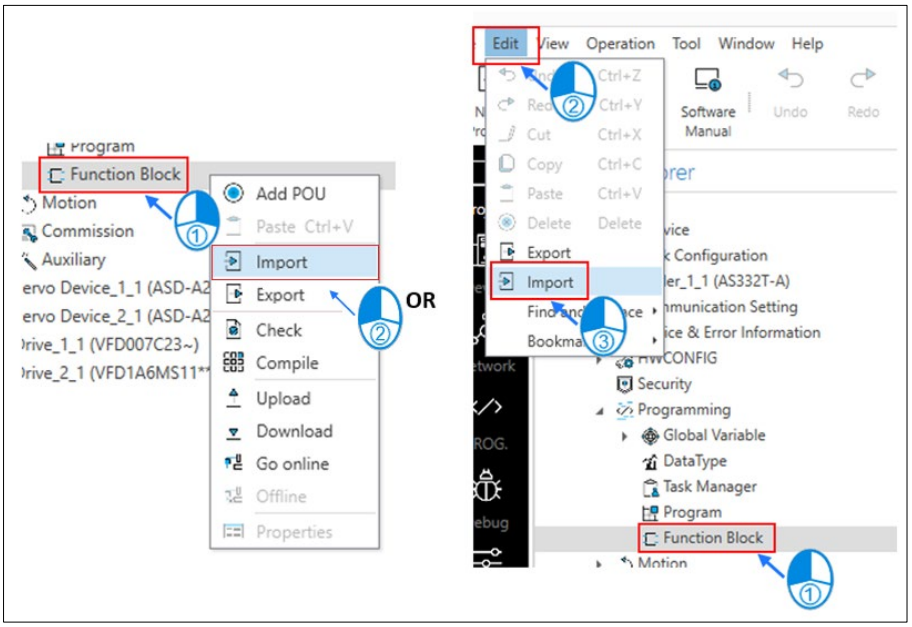


Figure 8 - 30: Function Block - Import

3. Select the file to import and click **Open** button.

   Result: The **Import** window displays as shown in the following figure.

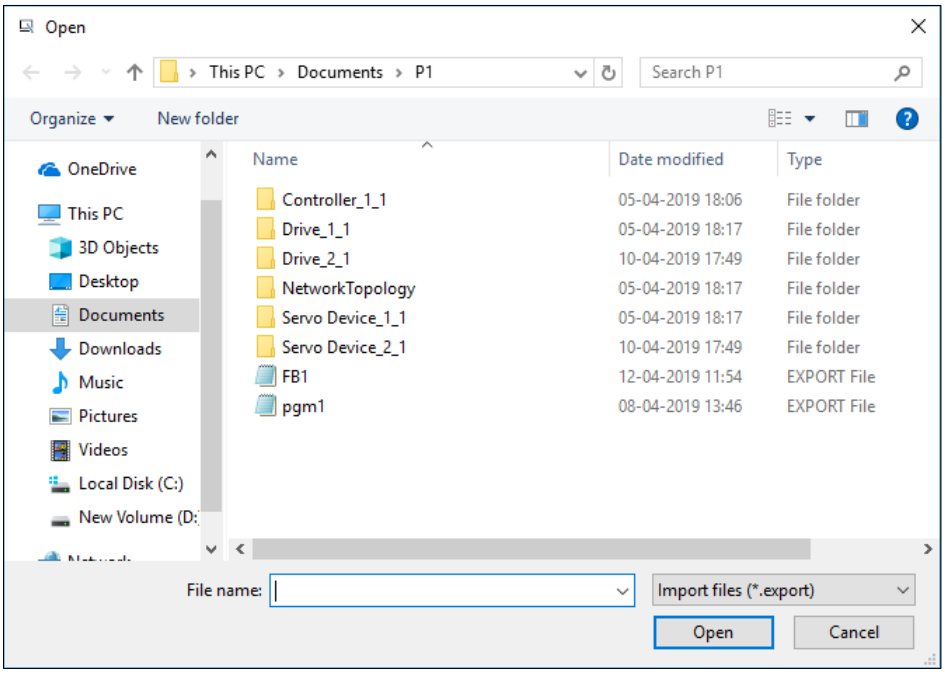


Figure 8 - 31: Open Window

4. Select the items to be imported and click **Next** button. Select the device in which the POU should be placed.

   Result: The **POU** is imported.

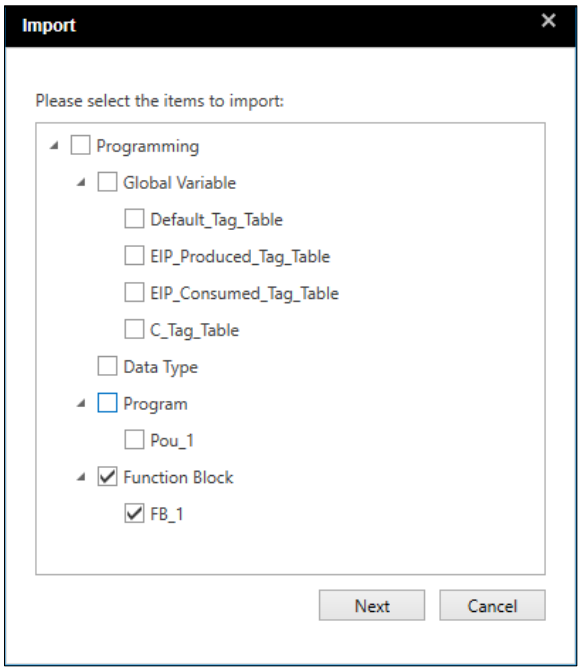


Figure 8 - 32: Import - Function Block

***NOTE**: If the file is protected with a password, user have to type the password in the **Password** for decrypting window and click **OK**. The POUs which are imported will not be assigned to any tasks. Besides, if the name of the POU which is to be imported is the same as the name of a POU in the project, it will prompt to rename or replace for proceeding.*

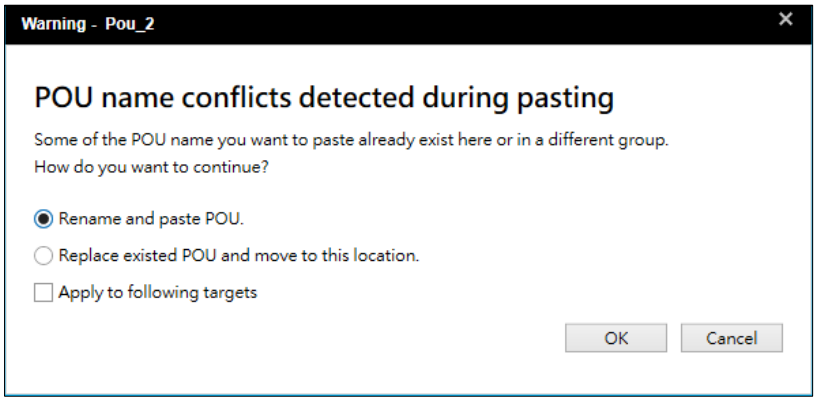


Figure 8 - 33: Function Block Import - naming conflict

8.2.2 VAR

During the process of developing a traditional program for a PLC, it generally takes much time to manage addresses. Besides, managing or debugging the program in a big project is difficult for user. As a result, the concept of **Variables** in a high-level programming language is introduced in IEC 61131-3. An Address in a PLC can be represented by a variable and an address can be automatically assigned to a variable. This saves the time for assigning addresses, increases the program readability and the efficiency of developing a program increases.

8.2.2.1 Application of variables and creation of variables:

A variable has to be declared before it is used. There are two types of variables – local variables and global variables. Local variables can only be used in the POU in which they are declared. Global variables can be used in all the POUs in a project. Besides, the identifier of a local variable in a POU can be the same as that of identifier of a local variable in another POU. However, if the identifier of a local variable declared is the same as the identifier of a global variable, the system will automatically regard the local variable in a POU as a local variable.

The regulations of creating the identifier of a variable are as follows:

- An identifier can contain a maximum of 30 characters (a Chinese character occupies 2 characters)
- The identifier of a variable cannot be a name reserved by the system, example an instruction code, an address or a keyword. However, if a name reserved by the system is a part of the identifier of a variable, the identifier is a valid name. For e.g. *M0* is an invalid name, but *_M0* is a valid name.
- The name of a variable and the name of an enumeration cannot be the same.
- Spaces cannot be used. For example, *INPUT CH0* is an invalid 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 valid identifier, but

INPUT__CH0 and *INPUT_CH0_* are invalid identifiers.

- Special characters cannot be used. For example, *, #, ?, \, %, @ etc. cannot be used.

Variable Classes:

In terms of functions, variables can be classified into five classes. Variable classes supported in global variables are the same as programs (local variables). The characteristics and usage description of the programs (local variable) and function blocks (local variable) are as below:

Variable Classes	Programs (Local Variable)			Function Blocks (Local		
	AH/AS	DVP	DVPxxMC	AH/AS	DVP	DVPxxMC
VAR	✓	✓	✓	Cannot specify location	Support only ES3 and cannot specify	Cannot specify location
VAR_RETAIN	Cannot specify location (*1)	Support only ES3 and cannot specify	✓			Cannot specify location
VAR_INPUT				Cannot specify location	Cannot specify location	Cannot specify location
VAR_OUTPUT				Cannot specify	Cannot specify	Cannot specify
VAR_IN_OUT				Cannot specify location	Cannot specify location	

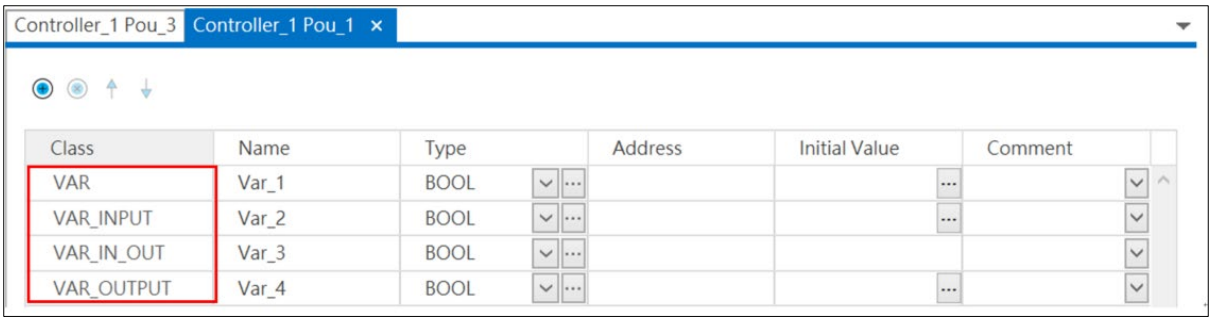


Figure 8 - 34: Variable Classes

VAR – General variable:

The variables of this class are for general operations only. The significance of a variable of this class depends on the data type of the variable or the corresponding address.

VAR_RETAIN – Latched variable:

AS series PLC support the variables of latching class. User cannot declare the variables of this class in a function block. The function of the variables of the VAR_RETAIN class is the same as the function of the variables in the VAR class. The difference between the variables of the VAR_RETAIN and the VAR classes lies in the fact that the addresses automatically assigned to the variables of VAR_RETAIN class are latched addresses. As a result, the values of the variables of this class are retained after the PLC is disconnected.

NOTE: *User cannot assign an address to a variable of this class. In addition, the data type cannot be Timer, Counter or Step.*

VAR_INPUT – variable used as an input pin of a function block:

A variable of this class is used as an input pin of a function block and is declared in the function block. If a function block is called, the variable of this class can receive the input value from the caller. Besides, in a ladder diagram/function block diagram,

the variables of this class are put at the left side of the function blocks and the pins which receive the values from the caller are assigned to the variables of this class.

VAR_OUTPUT – variable used as an output pin of a function block:

A variable of this class is used as an output pin of a function block and is declared in the function block. After the execution of a function block is complete, the operation result is sent to the caller through the variable of this class. Besides, in a ladder diagram /function block diagram, the variables of this class are put at the right side of the function blocks and the pins which sent the operation results to the caller are assigned to the variables of this class.

VAR_IN_OUT – variable used as a feedback pin of a function block:

A variable of this class is used as a feedback pin of a function block and is declared in the function block. For example, refer the below function block FB_DT. When the function block is called, the caller sends the value D1 to DT_IO, which is a variable of the VAR_IN_OUT class. After the operation comes to an end, the final value of DT_IO is sent to D1. Besides in a ladder diagram/function block diagram, the variables of this class are put at the right side of the function block and the blue pins which connect to the caller are assigned to the variables of this class.

8.2.2.2 Data Types

The data type of a variable determines the significance of the value of the variable. Suppose there are two variables - 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 address which can be involved in arithmetic or data transfer.

The data types supported in DIADesigner are listed in the following table. Different models support different data types and the variables which can be declared in the

POU of the program type are different from the variables which can be declared in a POU of the function block type.

Data Type	Description	AS Controller	
		Program	FB
BOOL	Boolean data type A Boolean value represents a state of a contact	✓	✓
BYTE	8-bit value 8-bit data can be stored		
WORD	16-bit value 16-bit data can be stored	✓	✓
DWORD	32-bit value 32-bit data can be stored	✓	✓
LWORD	64-bit value 64-bit data can be stored	✓	✓
SINT	8-bit signed integer The highest bit represents a positive/negative sign		
INT	16-bit integer The highest bit represents a positive/negative sign	✓	✓
DINT	32-bit integer The highest bit represents a positive/negative sign	✓	✓
LINT	64-bit integer The highest bit represents a positive/negative sign	✓	✓
USINT	8-bit unsigned integer		
UINT	16-bit unsigned integer		
UDINT	32-bit unsigned integer		
ULINT	64-bit unsigned integer		
REAL	32-bit floating–point value A decimal is represented by a 32-bit value	✓	✓
LREAL	64-bit floating–point value A decimal is represented by a 64-bit value	✓	✓
ARRAY	Array data type	✓	✓

Data Type	Description	AS Controller	
		Program	FB
	When a variable is declared, the length of an array and the data type which is to be stored are specified (max. length of Array is 512)		
STRING	String data type An ASCII code is represented by an 8-bit value. When a variable is declared, the length of the string is specified (A String is composed of 128 ASCII codes at most)	✓	✓
STEP	It is used as a recognition flag for a Step	✓	
Function Block	It represents the name of a function block	✓	✓
COUNTER	Counter data type It represents a Counter	✓	
TIMER	Timer It represents a Timer	✓	
POINTER	It is a pointer for a word address		✓
T_POINTER	It is a pointer for a Timer		✓
C_POINTER	It is a pointer for a Counter		✓
HC_POINTER	It is a pointer for a High Speed Counter		✓
TIME	Time format T#XXXXXXdXXhXXmXXsXXX.XXXXXXm, Unit: ms. Display Range: T#0ms~213503d23h34m33s709ms		
DATE	Time format D#Y-M-D. Display Range: D#1970-01-01~D#2106-02-07. Unit: sec.		
TOD	Time format. TOD#Hr:Min:Sec.ms Display Range: TOD#00:00:00~23:59:50:59.00. Unit: ms. If value=0, it displays TOD#00:00:00; If value=1, it displays TOD#00:00:00.001; If value=86399999, it displays TOD#23:59:59.999;		

Data Type	Description	AS Controller	
		Program	FB
	If value=86400000, it displays TOD#00:00:00; If value=4294967295 it displays TOD#17:2:47.295		
DT	Time format D#Y-M-D-Hr-Min-Sec. Display Range: DT#1970-01-01-0:0:0~2106-02-07-6:28:15. Unit: s.		

Assigning an address to a variable and setting the initial value of a variable:

A device is assigned to a variable according to the data type of the variable. User can set the initial value of a variable. If a project is downloaded, the initial values of the variables can be written in to the devices assigned to the variables.

The principle of assigning devices to variables are described in the following table. Different models have different assignment principle.

Model	Assignment Principle
AS Controller	• User can assign devices to the global variables and the local variables declared in the POUs of the program type. The system can also automatically assign devices to the global variables and the local variables declared in the POUs of the program type. • The system automatically assigns the devices to the local variables declared in the function blocks. User cannot assign devices to the local variables declared in the function blocks by themselves. • The devices assigned by the system are memories reserved by the CPU module and are not data registers or auxiliary relays. If the data type of a variable is STEP, TIMER or COUNTER, the system will automatically assign a stepping relay, a timer or a counter to this variable.

The relation between the data types and the device types which can be assigned is described in the following table:

Model	AS Controller	
	Device assigned by user	**Device assigned by system**
BOOL	Contact M/SM or bit in the device D/L/X/Y	Internal memory
WORD	D/L/X/Y/E/SR Register	Internal memory
DWORD	D/L/X/Y/E/SR Register	Internal memory
LWORD	D/L/X/Y/E/SR Register	Internal memory
INT	D/L/X/Y Register	Internal memory
DINT	D/L/X/Y Register	Internal memory
LINT	D/L/X/Y Register	Internal memory
REAL	D/L/X/Y Register	Internal memory
LREAL	D/L/X/Y Register	Internal memory
STRING	D/L/X/Y Register	Internal memory
STEP	S Register	S Register
COUNTER	C/HC Register	C Register
TIMER	T Register	T Register
ARRAY	The devices assigned to a variable whose data type is ARRAY depend on the array type specified. The array is composed of the devices starting from the device assigned by user or the system and the number of devices in the array conforms to the size of the array. The devices assigned to the variable whose data types is ARRAY cannot be SR/SM/E/F devices.	

Indirect Assignment and Modification for Register Symbol Variable:

In DIADesigner, a variable can be modified by an index register. The format is Identifier@Index register. An index register can be an E register or a variable to which an E register is assigned.

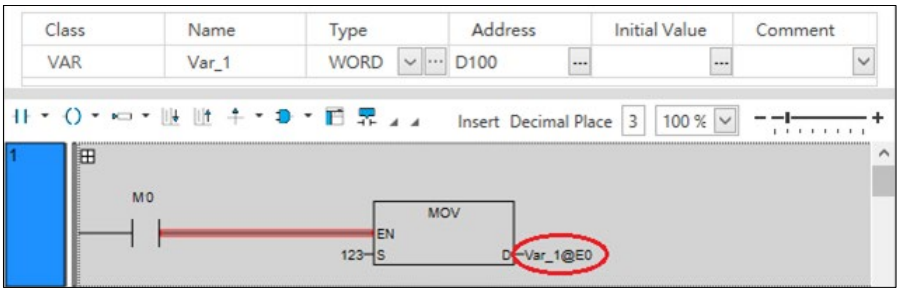


Figure 8 - 35: Indirect Assignment

The data stored in an index register indicates the offset for an object modified. Take the above figure for example. If the value in the index register E0 is 2, VAR_1@E0 indicates that 2 is added to the device address (D100) assigned to VAR_1, that is, VAR_1@E0 represents D102. When M0 is ON, the value 123 is transferred to the data register D102.

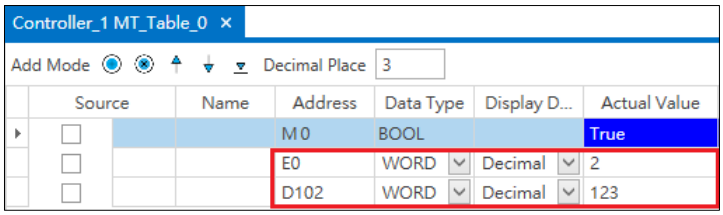


Figure 8 - 36: Value transferred to Index Register

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 variable, the variable whose data type is ARRAY cannot be modified by the index register. In the following figure, 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, user have to be more careful when they modify symbol whose data type is ARRAY with an index register. Please refer the following section more information about the usage of arrays.

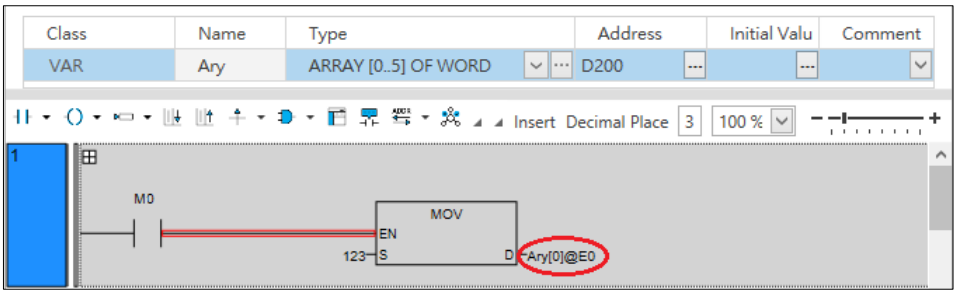


Figure 8 - 37: Index Register with Array

Additional remark:

- 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 user do not know which device is assigned to the symbol.
- 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, user 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.
- If the value in an index register is changed, the register which actually operates differs from the original register. As a result, if the original register is not used in the program, the final value in the original register is retained. Take AHCPU530-EN for example. In the following figure, 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.

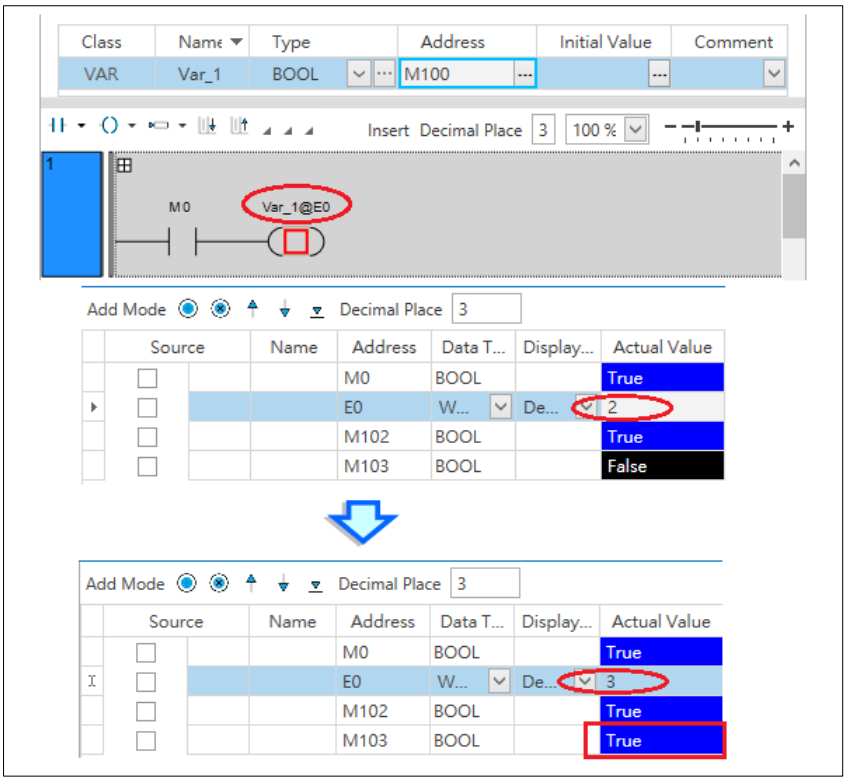


Figure 8 - 38: Value change

- 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, user have to make sure that the device assigned to the symbol is not an E register.

Bit in a Device represented by a variable:

An AS series CPU module supports the manipulation of the bits in a word device. The format is word register.bit number. For example, D0.2 indicates the 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 following figure, the data type of VB is WORD. If 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.

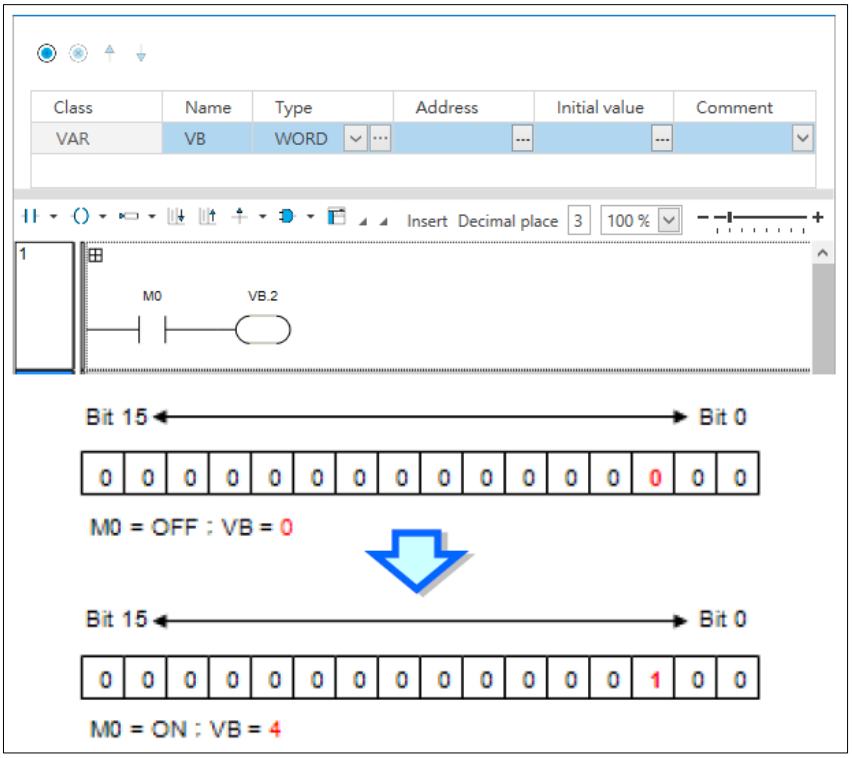


Figure 8 - 39: Bit representation

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.

Example 2: VB@E0.1 (VB represents D100. The value in E0 is 3.)

VB@E0 is interpreted first. As a result, VB@E0.1 represents D103.1. The bit which is actually manipulated is bit 1 in D103.

Example 3: VB@E0.1@E1 (VB represents D100. The value in E0 is 3, and the value in E1 is 2.)

VB@E0 and 1@E1 are interpreted separately. As a result, VB@E0 represents D103, and 1@E1 represents 3. VB@E0.1@E1 represents D103.3. The bit which is actually manipulated is bit 3 in D103.

Data Unit Type:

User-defined Data Type/Data Unit Type:

Sometimes, basic datatypes are not sufficient for user to write their own programs for the PLC. In the IEC-61131-3 standard, a high-level language such as the user-defined datatype is adopted to create a new datatype as user desired, allowing user to define their own variables and improving the readability and the efficiency of developing a program.

Structure:

Definition of a Structure:

A Structure is a collection of data, consisting of same amount of various datatypes. 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.

Creating a Structure:

Follow these steps to create a Structure in DIADesigner:

1. Right-click on the **User-defined Data Type** option in the **Project View** to open its context menu.
2. Click **Add User-defined Data Type** option in the context menu.

   Result: **Add User-defined Data Type** window displays as shown in Figure 8-39.

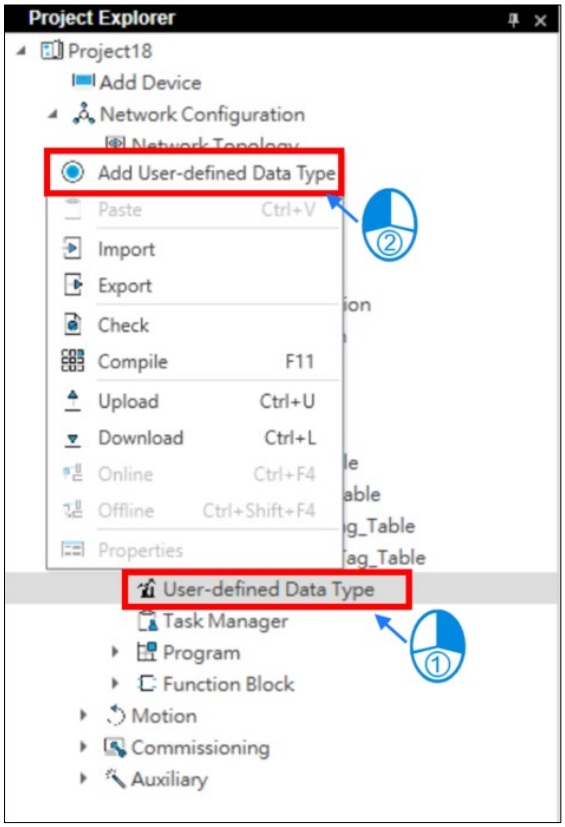


Figure 8 - 40: Add User-defined Data Type in Project Explorer

3. Enter a **Name** for the Structure DUT, select **Structure** in **Select Type**, enter **Comment** (optional) for DUT and click **OK**.

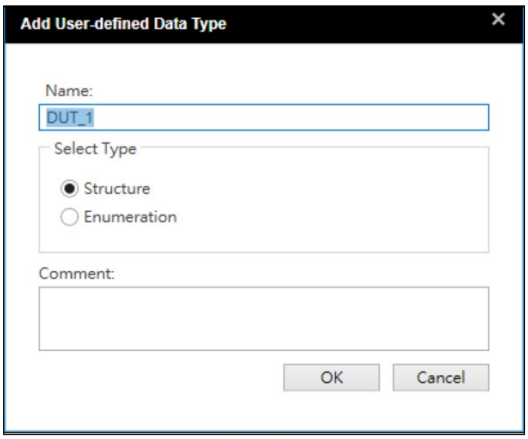


Figure 8 - 41: Add User-defined Data Type window

Result: DUT of Structure is created.

After a Structure DUT is created, user can open the **User-defined Data Type** window by double-clicking the DUT created. **User-defined Data Type** window is shown with examples and definitions for reference as shown in the following figure. User can edit the examples to make them as their own structures.

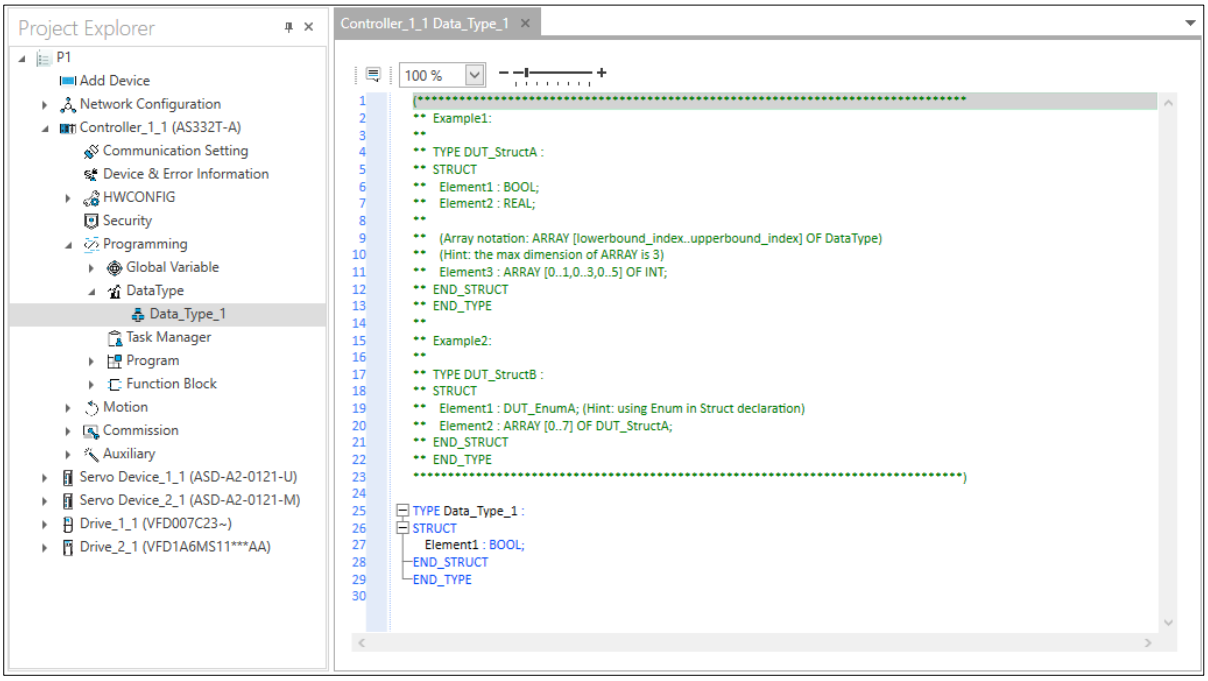


Figure 8 - 42: Data Type window - Structure

The Structure format is:

- Words in BLUE color are the System Keywords
- Words in BLACK color are the user-defined structure names and element names.

A sample Structure is shown in the following figure

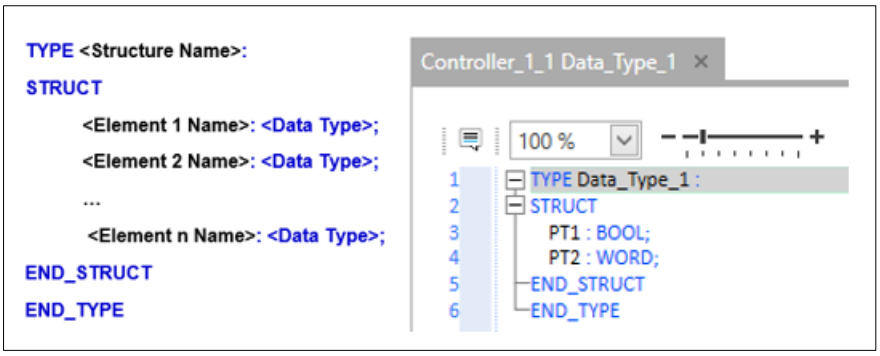


Figure 8 - 43: Structure - Example

Definitions:

Structure Name: Enter the Structure Name here. It is recommended to use the same name as the name in the Project Explorer.

Element Name: Define the Element Name.

Data Type: Define the Datatype of the Elements such as BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, REAL, LREAL, ARRAY or another Structure or Enumeration or UNION.

When the element datatype is ARRAY, the format is –

ARRAY [0..n,0..n,0..n**] OF <data type>;**

***NOTE**: Maximum value of n is 65535.*

If the element datatype is Structure, write another Structure Name. But if the datatype is ARRAY, user cannot import ARRAY again. DIADesigner supports multi-dimensional Array, up to 3-dimensions.

The format of a two-dimensional array is –

ARRAY [0..n**] [**0..n**] OF <data type>;**

The format of a three-dimensional array is –

ARRAY [0..n**] [**0..n**] [**0..n**] OF <data type>;**

An example of ARRAY is shown in the following figure:

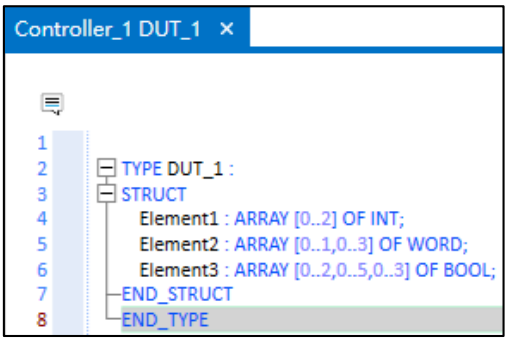


Figure 8 - 44: Structure - Array - Example

The naming rules of a Structure and an element are stated below:

- An identifier cannot be the same as names of the user-defined datatypes. 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 Element can be composed of 30 characters at most and a Chinese character takes two characters.
- Keywords (names reserved for the system) are not allowed as names. Example, 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 valid. For example, M0 is not a valid name, but _M0 is a valid name.
- The identifiers of a Structure and an Element cannot contain spaces. For example, INPUT CH0 is not a valid name.
- The identifiers of a Structure and an Element can contain underline, but it is not allowed to use 2 underlines continuously. Underlines cannot appear at the end even. For example, INPUT_CH0 is valid, but INPUT__CH0 and INPUT_CH0_ are not valid.
- The identifier of a Structure and an Element cannot contain special characters, including *, #, ?, \, %, @ etc.

Using a Structure Variable:

After defining a Structure DUT, user can create the variables in the Global variable table or local variable table or in function block variable table. Create a Variable and select the respective DUT type created as per requirement.

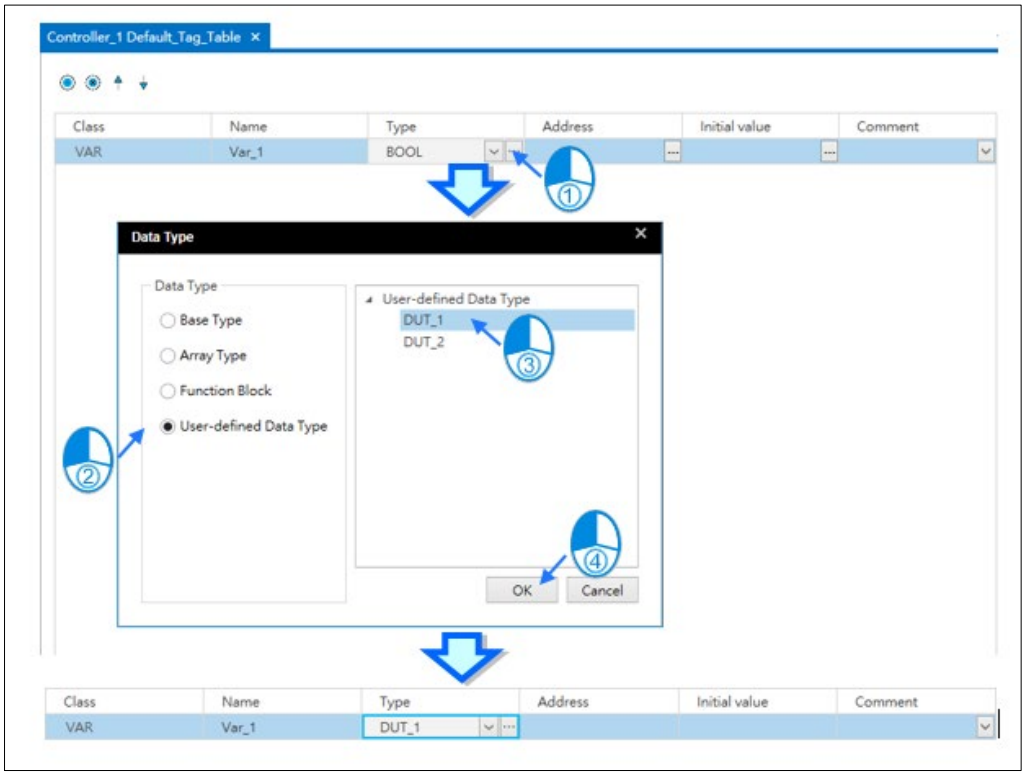


Figure 8 - 45: Data Type – Creating Structure

Application of Structures:

After the Structures and Variables are declared, they can be used in the POU. The datatype of the Variable struct_var0 is Struct_0 in the function block as shown in the following figure. User can use “.” To assign the element for the structured variables. For example, struct_var0.pt1 means the first element pt1 is assigned as a contact to the datatype BOOL operand and struct_var0.pt2 means the second element pt2 is assigned as a pin to the datatype WORD operand as shown in the following figure.

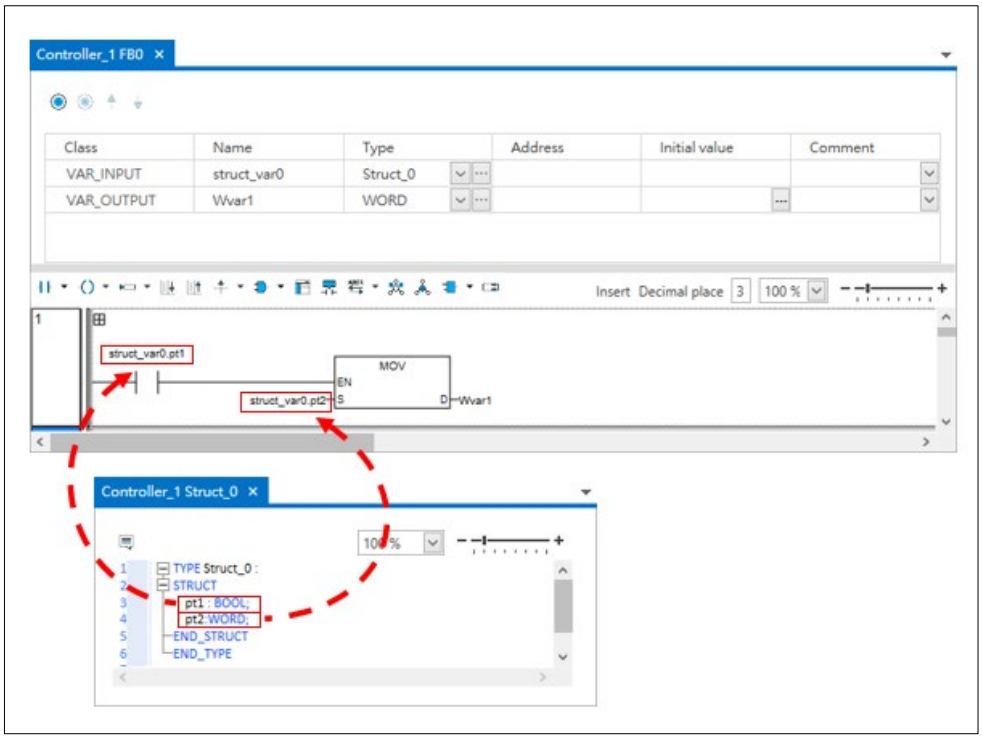


Figure 8 - 46: Assigning Structure to a Pin

When using the aforementioned function block, Struct_0 datatype variable should be declared in the POU, so that the Struct_0 datatype input pin and output pin can be consistent. The elements with the same structure can use “.” to assign the element for the structure type variables, for example POUStruct_var is shown in the following figure.

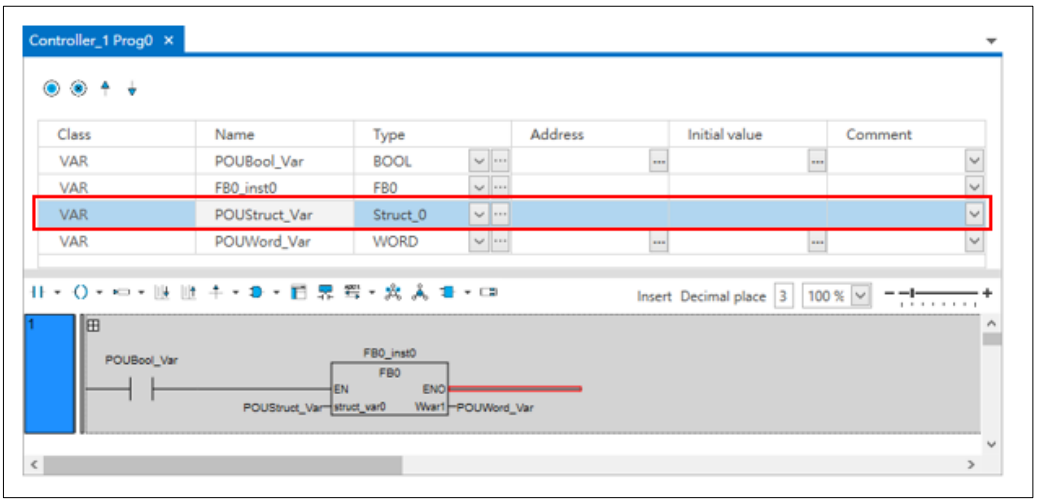


Figure 8 - 47: Declaring Structure in POU

From the description above, we know that when the structure type variable POUStruct_Var is used in the FB0, it also delivers all the elements from the variable for struct_var0.pt1 and struct_var0.pt2 to operate. Thus, user can define the structure type variables as the input/output pin for the function block to deliver all the elements from the variable. So that the pin defining time can be saved and the program can be simpler.

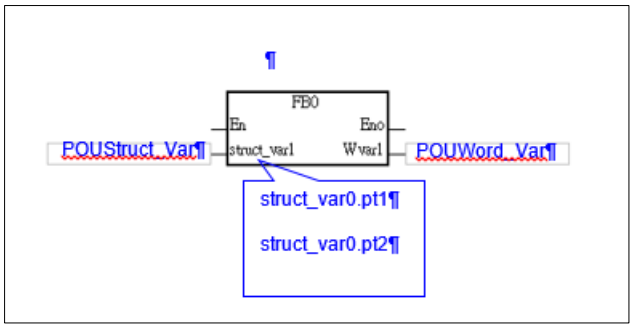


Figure 8 - 48: Structure used in a POU

When the structure is constructed with an array element, the variables in the structure can be written as shown in the following figure. The status of M0 can be written in the [1] of the appointed element pt3 in the variable Pstruct_var1.

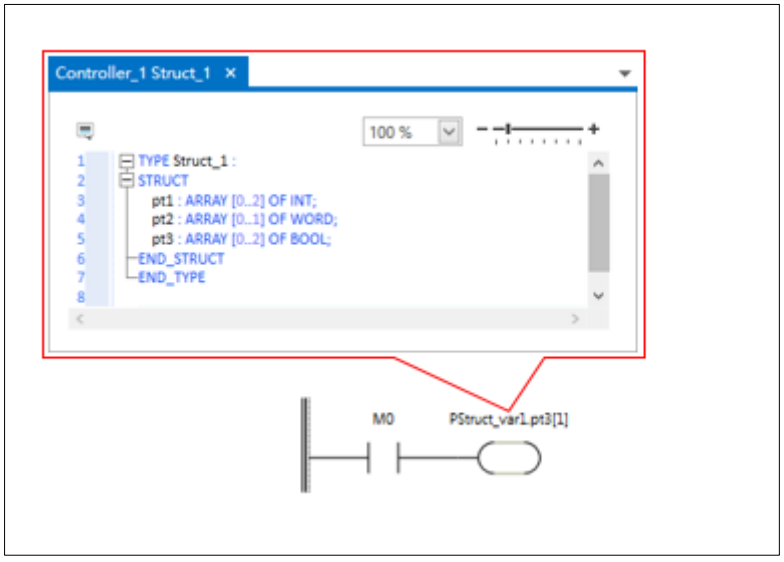


Figure 8 - 49: Writing value to a Structure variable

Change Structure to Enumeration:

User 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.

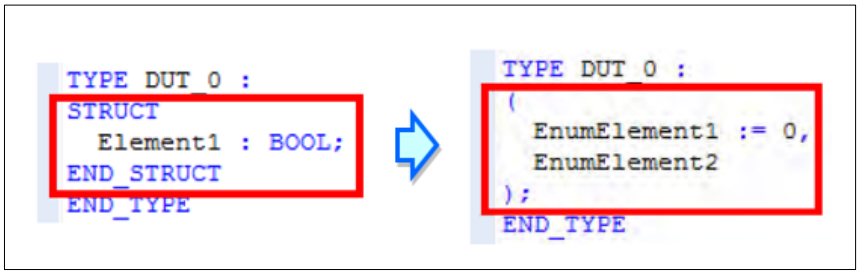


Figure 8 - 50: Changing Structure to Enumeration

Enumeration:

Definition of Enumeration:

User 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 the values that the enumeration represents.

Creating an Enumeration:

Follow these steps to create an Enumeration in DIADesigner:

1. Right-click on the **User-defined Data Type** option in the **Project View** to open its context menu
2. Click **Add User-defined Data Type** option in the context menu.

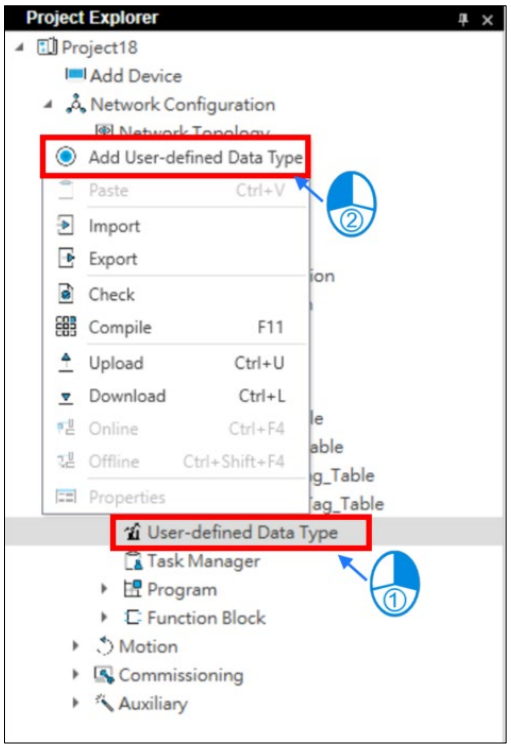


Figure 8 - 51: Add User-defined Data Type in Project Explorer

Result: **Add User-defined Data Type** window displays as shown in the following figure.

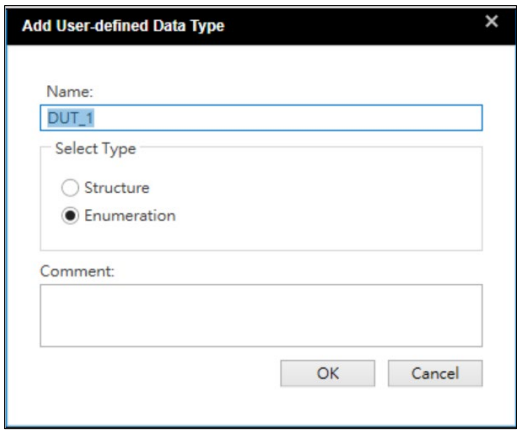


Figure 8 - 52: Add User-defined Data Type window

3. Enter a **Name** for the Enumeration DUT, select **Enumeration** in **Select Type**, enter **Comment** (optional) for DUT and click **OK**.

   Result: DUT of Enumeration is created.

After an Enumeration DUT is created, user can open the **DataType** window by double-clicking the DUT created. DataType window displays with examples and definitions for reference as shown in the following figure. User can edit the examples to make them as their own enumerations.

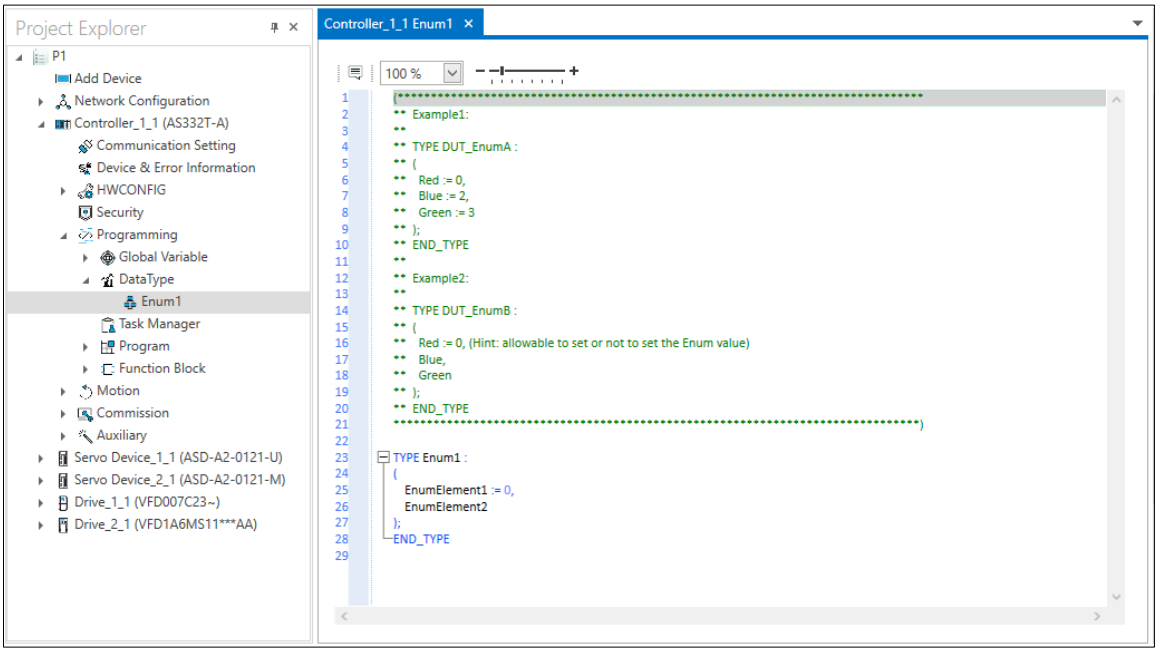


Figure 8 - 53: DataType window - Enumeration

The Enumeration format is:

- Words in BLUE color are the System Keywords
- Words in BLACK color are the user-defined enumeration names, element names and values.

A sample enumeration is shown in the following figure.

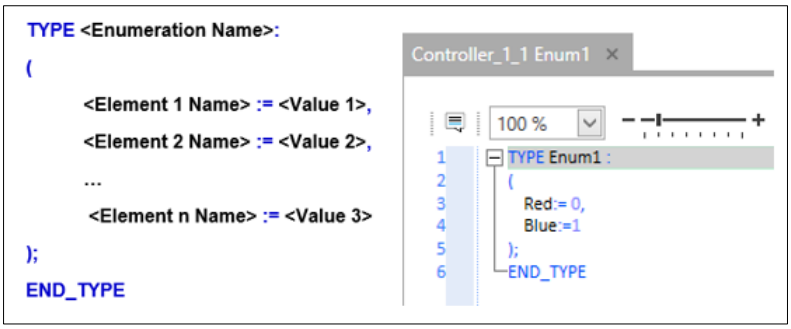


Figure 8 - 54: Enumeration – format

Definitions:

Enumeration Name: Enter the Enumeration Name here. It is recommended to use the same name as the name in the Project Explorer.

Element Name: Define the Element Name.

Value: Define values for every element, ranging from -32768 ~ 32767. It is acceptable to leave this field empty (blank). The system will use defaults according to the pervious element value and avoid duplications. The system will set the value of the element Blue to 1 and Yellow to 6 in the example in following figure:

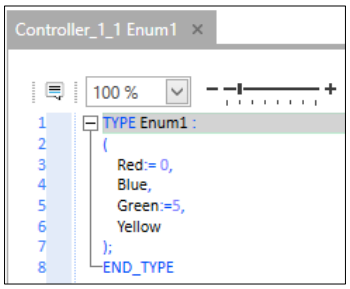


Figure 8 - 55: Enumeration - Example

The naming rules of an Enumeration and an element are stated below:

- An identifier cannot be the same as names of the user-defined datatypes. 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 Element can be composed of 30 characters at most and a Chinese character takes two characters.
- Keywords (names reserved for the system) are not allowed as names. Example, 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 an Enumeration or an Element, it is valid. For example, M0 is not a valid name, but _M0 is a valid name.
- The identifiers of an Enumeration and an Element cannot contain spaces. For example, INPUT CH0 is not a valid name.
- The identifiers of an Enumeration and an Element can contain underline, but it is not allowed to use 2 underlines continuously. Underlines cannot appear at the end even. For example, INPUT_CH0 is valid, but INPUT__CH0 and INPUT_CH0_ are not valid.
- The identifier of an Enumeration and an Element cannot contain special characters, including *, #, ?, \, %, @ etc.
- The data size of Enumeration variables are the same as INT data size.

Using an Enumeration Variable:

After defining an Enumeration DUT, user can create the variables in the Global variable table or local variable table or in function block variable table. Create a Variable and select the respective DUT type created as per requirement.

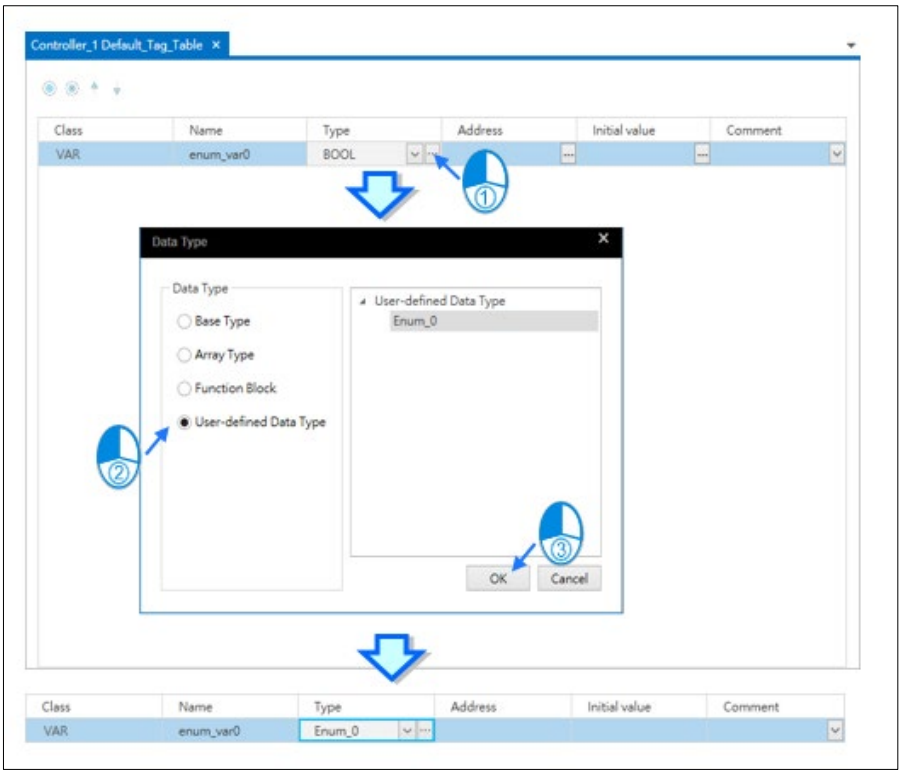


Figure 8 - 56: Data Type - Creating Enumeration

Applications of Enumerations:

After the Enumeration and variable are declared, they can be used in a POU. The datatype of the variable POUEnum_Var0 is Enum_0. The initial value can be set by the element name or values. The following figure shows the element name – Red.

User can use “.” to assign the element for the enumeration variables. For instance, user can assign Enum_0.Blue to POUenum_Var0, the variable Enum_0 and execute the instruction MOV; the defined element or value is the enumeration value and after the execution of MOV, the value of POUenum_var0 is 1.

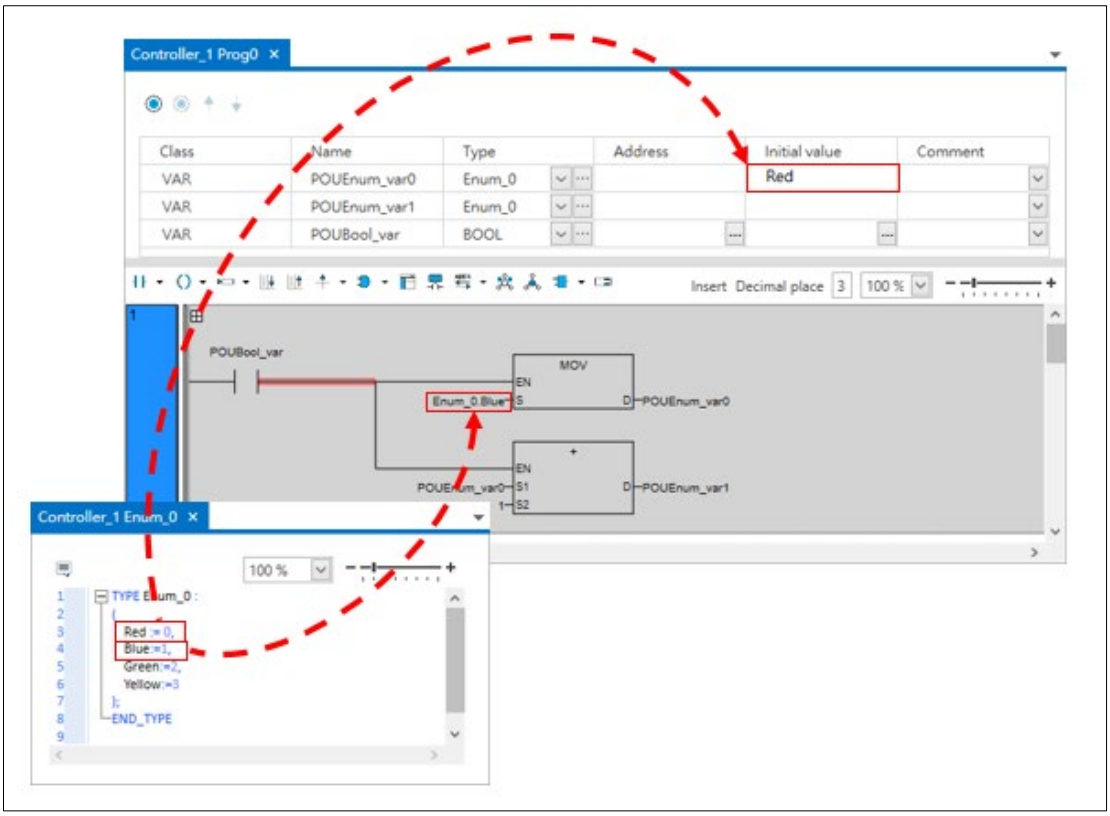


Figure 8 - 57: Applying Enumeration to a Pin

User can also perform numerical operations using the instruction + and set the value to 1, POUEnum_Var0 is incremented by 1 as the example shown in the above figure. When the value of POUEnum_Var0 is 1, after executing the instruction +, the value result is 2 and its element is Green as defined.

When running and monitoring the program, the enumeration variables will show their current defined element names. For example, when the value is 1, the POUEnum_Var0 is seen as Blue(1) as they are defined in the following figure. But if the value of the enumeration variable is not in the defined element range, the value is 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 is stated as shown in the following figure.

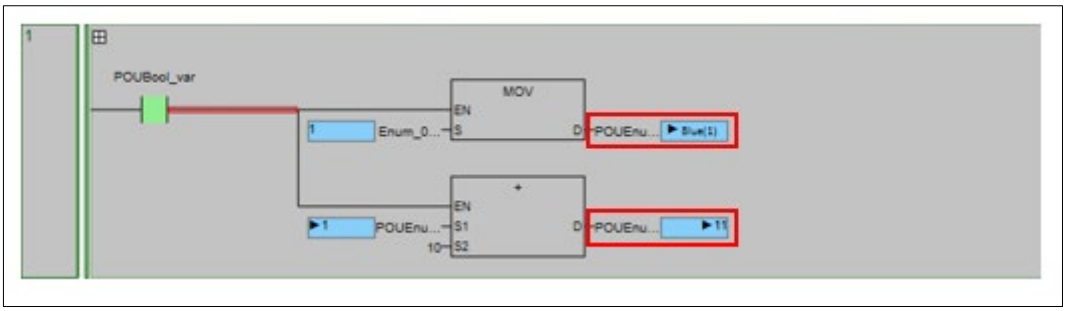


Figure 8 - 58: Declaring Enumeration in a POU

Change Enumeration to Structure:

User 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.

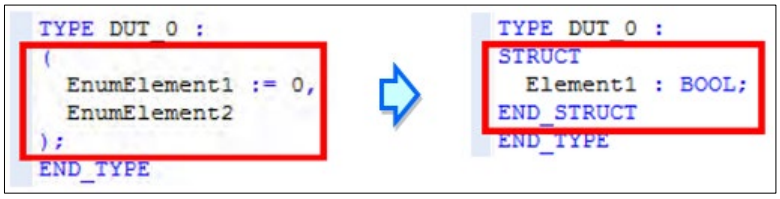


Figure 8 - 59: Changing Enumeration to Structure

Managing the Variables in DIADesigner:

Variable Tables:

There are two types of variable tables in DIADesigner:

- Global variables table
- Local variables table

Global variables table:

Double-click on the Project Explorer - **Global Variable** > **Default_Tag_Table** to open the Global variables table. **Default_Tag_Table** table is shown in the following figure:

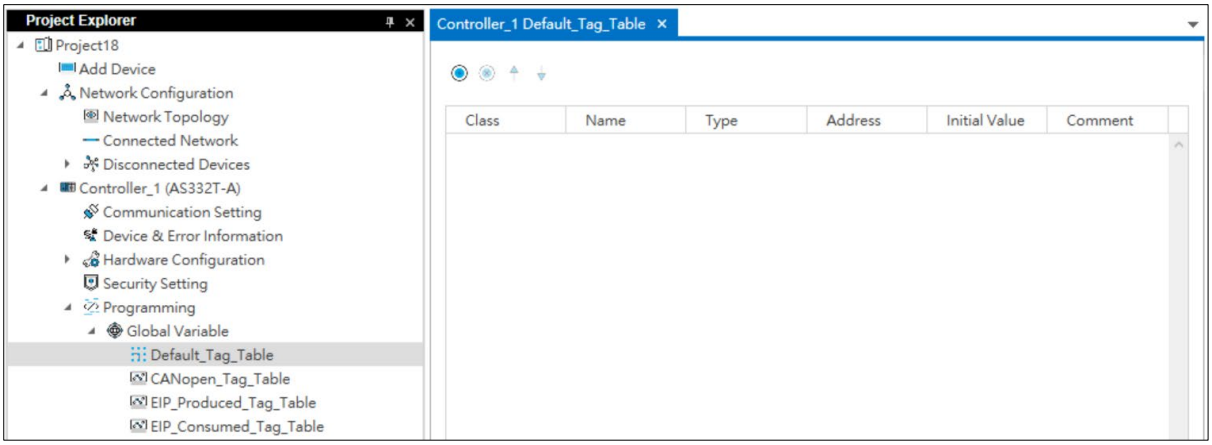


Figure 8 - 60: Global Variables

Local variables table:

Local Variables table in a POU is located at the top of the POU window.

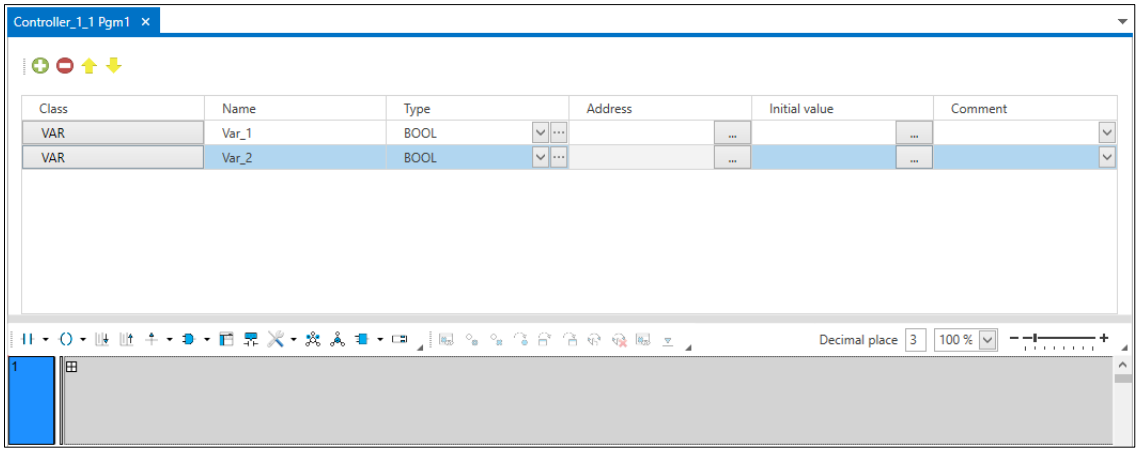


Figure 8 - 61: Local Variables

Adding a Variable:

User can add a Variable in Global or Local variable table by means of the following steps:

- Click on the icon in the toolbar of variable table

  or

- Right-click in the variable table and click on **Add**

  or

- By pressing **Insert** key on the keyboard

  Result: Variable is created.

The attributes of a Variable are described in the following table:

Attribute	Description
Class	Select the class of the variable
Name	Enter a name for the variable
Data Type	Select the datatype of the variable
Address	Select the address of the variable
Initial value	Select or enter an initial value of the variable
Comment	Enter a comment for the variable

Variable Class:

Variable Class drop-down list vary with the variable table created (Global/Local). The items in the Class drop-down list box for a global or a local variable in a POU are VAR and VAR_RETAIN.

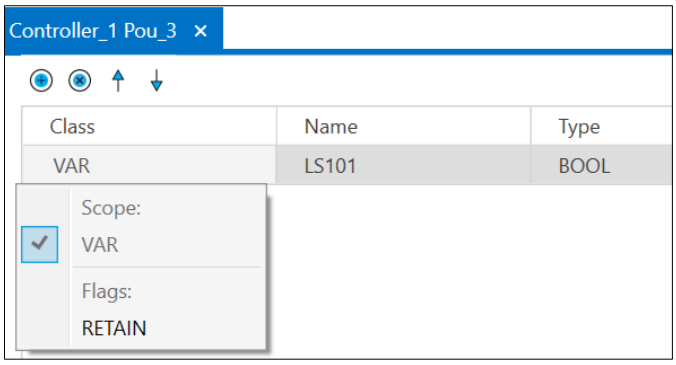


Figure 8 - 62: Variable Class in a Program POU

The items in the Class drop-down list box for a POU of the function block type are VAR, VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT.

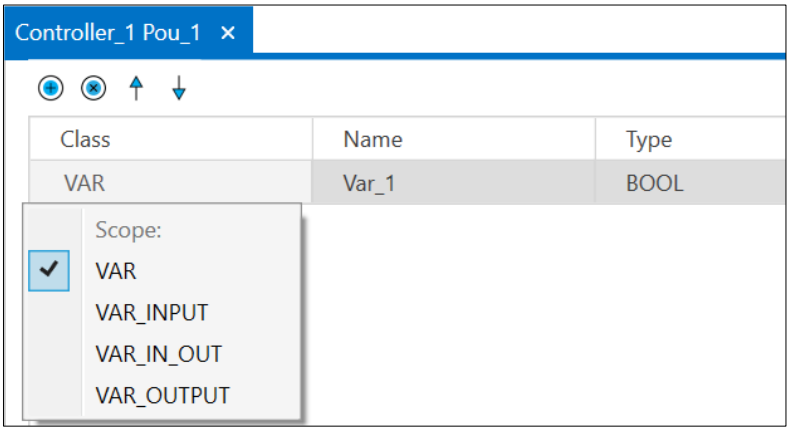


Figure 8 - 63: Variable Class in a Function Block POU

Variable Type:

Click on the icon in the **Type** attribute to display the **Data Type** window as shown in the following figure.

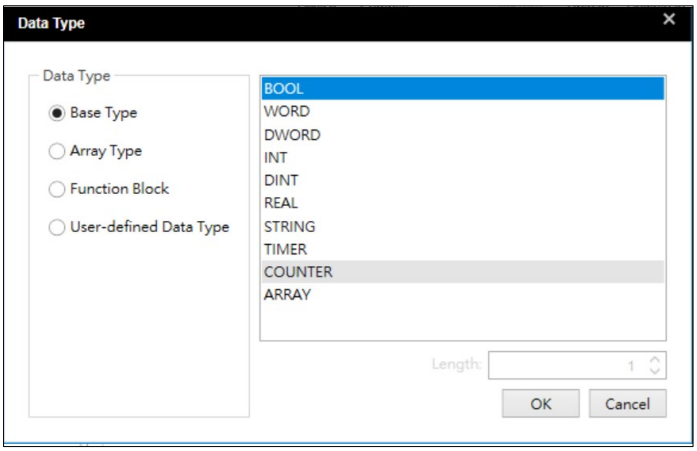


Figure 8 - 64: Data Type

User can select the datatype of the variable needed as per their requirements.

Variable Type – Base Type:

Base Type Variables are:

- BOOL
- WORD
- DWORD
- INT
- DINT
- REAL
- STRING
- ARRAY
- TIMER
- COUNTER

Variable Type – Function Block:

Select **Function Block** option button in the **Data Type** section and then select a function block definition in the User defined function block section. If the datatype of the variable is a function block, a function block instance is declared.

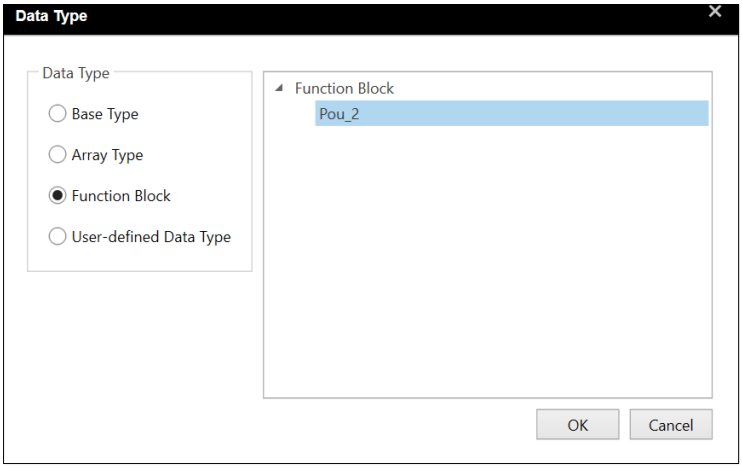


Figure 8 - 65: Data Type - Function Block

Variable Type – User-defined Data Type:

Select the **User-defined Data Type** option button in the **Data Type** section and then select a data unit type in the Data Type Unit section.

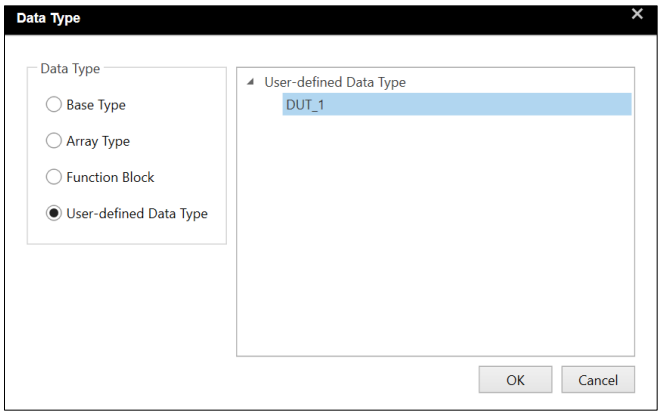


Figure 8 - 66: Data Type - User-defined Data Type

STRING:

If the datatype of a variable is String, user 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 user set is the maximum number of characters in the string. User can click on the increment/decrement buttons to increase or decrease the string size by 1. A String variable with size of 8 is shown in the following figure:

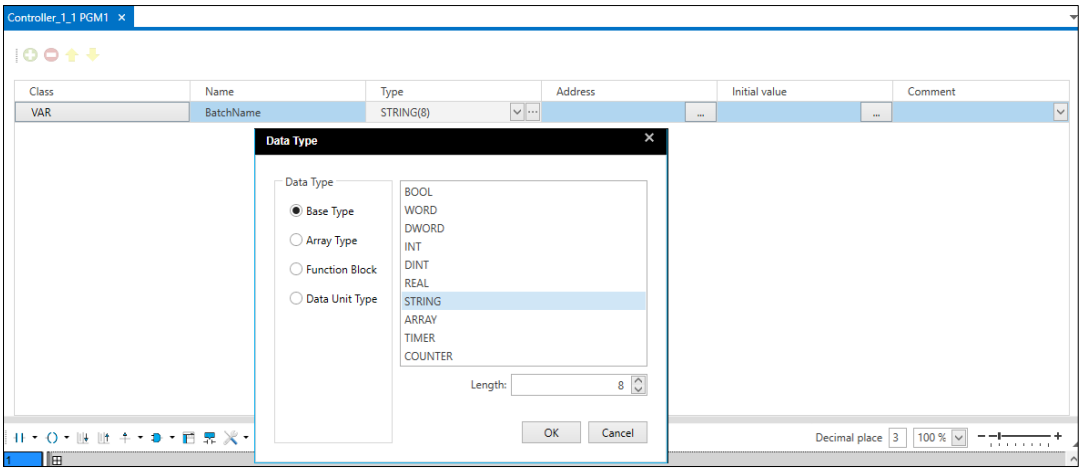


Figure 8 - 67: Data Type with String Variable

ARRAY:

If the datatype of a variable is ARRAY, then user can select **Array Type** in the **Data Type** section. In the **Ranks and base type specification** section, at present user may select up to 3 dimensional arrays, set the data size [0..n] and the datatype in **Base Type**. **Result** section displays the datatype selected. An example of 1-dimensional array with INT datatype with data size 3 is shown in the following figure.

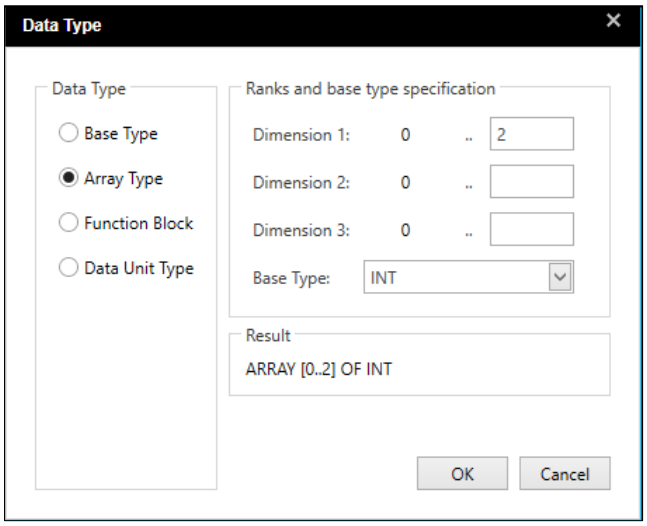


Figure 8 - 68: Data Type with Array Variable

Address:

Follow these steps to assign address to Variables:

1. Entering the address directly in the field by keyboard entry

   Or

   Click the ... icon to open the **Reference Register** window.

2. Select the **Register**, **Address** and **Bit Number**.
3. Click on **OK**.

A sample address of X0.0 is shown in the following figure.

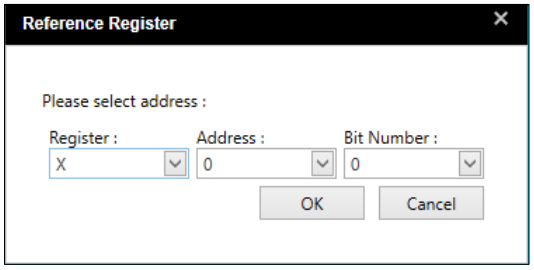


Figure 8 - 69: Reference Register window

Initial Value:

If user want to set Initial value for a variable, they can type the initial value directly. For variable of BOOL datatype, it can be 0 or 1. For STRING datatype, initial value cannot be more than maximum data size characters. Characters **do not** need to be put in double quotes. If the datatype of a variable is Array, then upon clicking the ... icon in initial value field, the **Reference Initial Value Setting** window displays as shown in the following figure.

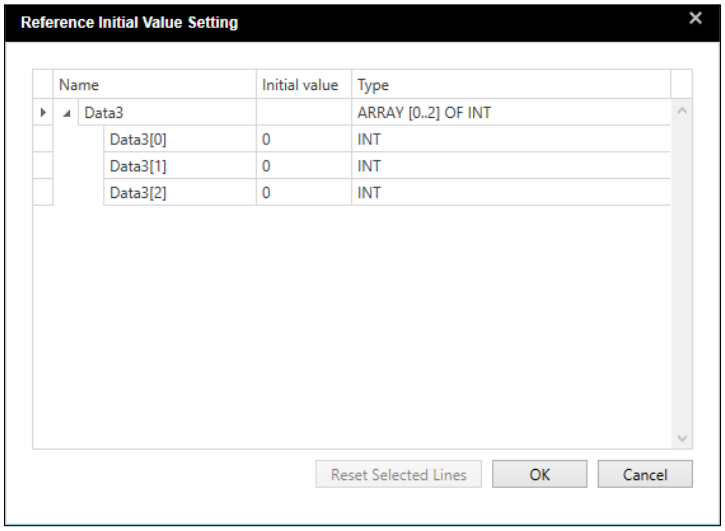


Figure 8 - 70: Reference Initial Value Setting window

Comments:

User can enter comments for the variables by directly typing.

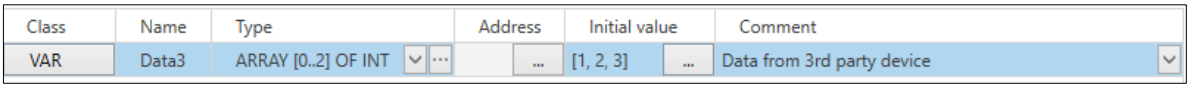


Figure 8 - 71: Variable Comments

Variable whose datatype is an Array:

If user want to use a variable whose datatype 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 a AS PLC CPU module is a Structured Text, the index can be a variable. The minimum index value must be 0, whether the index is a constant or a variable. 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 0 ~ 9. If the index is a variable then, the value of the variable cannot be larger than the number of elements in the array. Otherwise an error will occur during the operation.

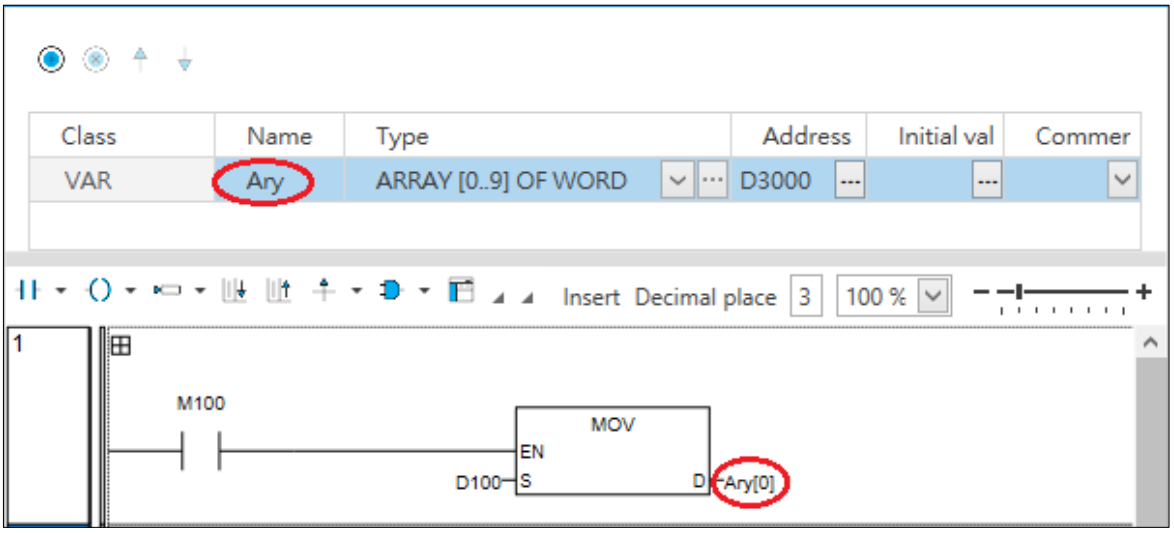


Figure 8 - 72: Array Variable linked to a Pin

If user declare a variable whose datatype is an 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 1 ~ 512. A start device address is assigned to a variable whose datatype is array according to the array type. Notice that the start device address is assigned to a variable whose datatype is ARRAY cannot be an SR device, an SM device, an E device or and F device. Besides, an array composed of the devices starting from the device assigned by user or the system and the number of devices in the array conforms to the size of the array.

In the following figure, ARRAY[0..2] of DWORD in the Type… cell 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 the start device address is D0. As a result, the array is composed of D0 ~ D5 (6 WORD devices).

ARRAY[0..4] 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 Register address in the address cell for Ary_1 indicates that the start address is M0. As a result, the array is composed of M0 ~ M4.

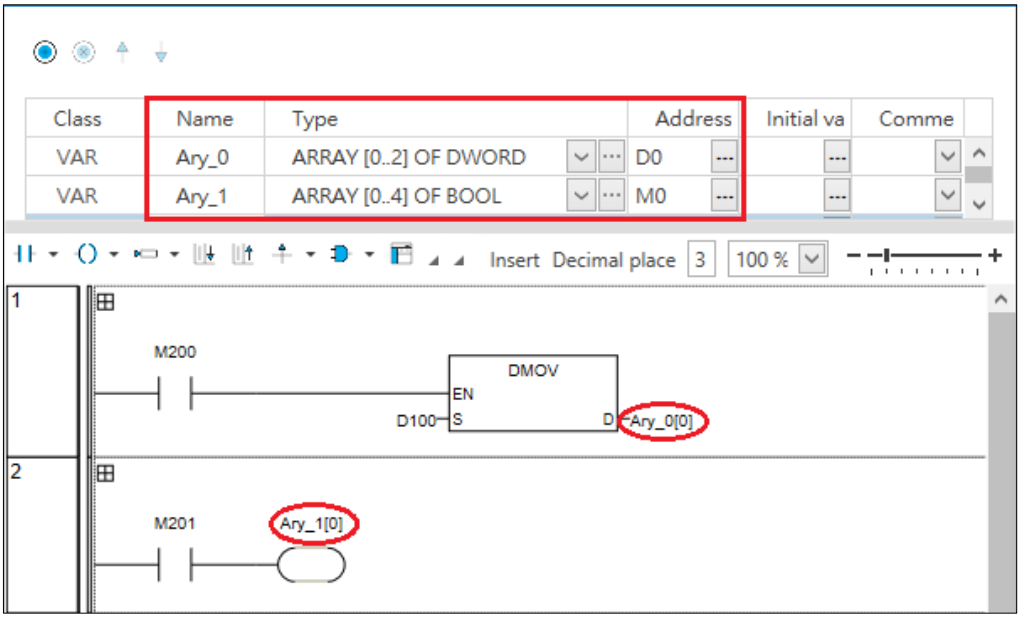


Figure 8 - 73: Array – examples

When user declare a variable whose datatype is ARRAY, they can set the initial value of the variable.

Example: [1,2,3,4,5] in the **Initial Value** attribute for Ary_1 indicates that the initial value of Array elements:

Figure 8 - 74: Array - Initial Value

Array Element	Initial Value	Array Element	Initial Value
Data1[0]	**1**	**Data2[0]**	**0**
Data1[1]	**2**	Data2[1]	1
Data1[2]	3	Data2[2]	4
Data1[3]	4	Data2[3]	5
Data1[4]	5	Data2[4]	6

When the element is with a two-dimensional array or three-dimensional array in the data unit type, the variable can use [n,n] or [n,n,n]. In the above example, the element pt1 is with a three-dimensional array, the datatype of the variable can be vardut.pt1[1,1,1] to assign the element [1,1,1] from the array.

Variable whose datatype is a STRING:

In DIADesigner, a String is composed of ASCII codes and the ASCII codes are surrounded by double quotes. E.g. “ABCD”, a variable whose datatype is STRING is generally used with an applied instruction. Refer programming manual for more information.

AS series PLC support the variables whose datatypes are STRING and the datatype of a variable of the VAR_INPUT class, the VAR_OUTPUT class or the VAR_IN_OUT class in a function block cannot be STRING. As a result, if user want to process a string in a function block, they can create the String as a global variable.

When user declare a variable whose datatype is STRING, they can specify the maximum size of the string. The number of characters in a string is within the range 1 ~ 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 is a string does not occupy the 2 bytes in a WORD device, the ending character is assigned to another device.

In the following figure, the number in the parentheses in the **Type** attribute for a variable whose datatype is String indicates the maximum size of the string. Besides,

the characters in the **Initial Value** attribute for a variable whose datatype is String cannot be put in double quotes and the number of characters in the **Initial Value** cell for the variable whose datatype is string must be less than or equal to the characters in the string.

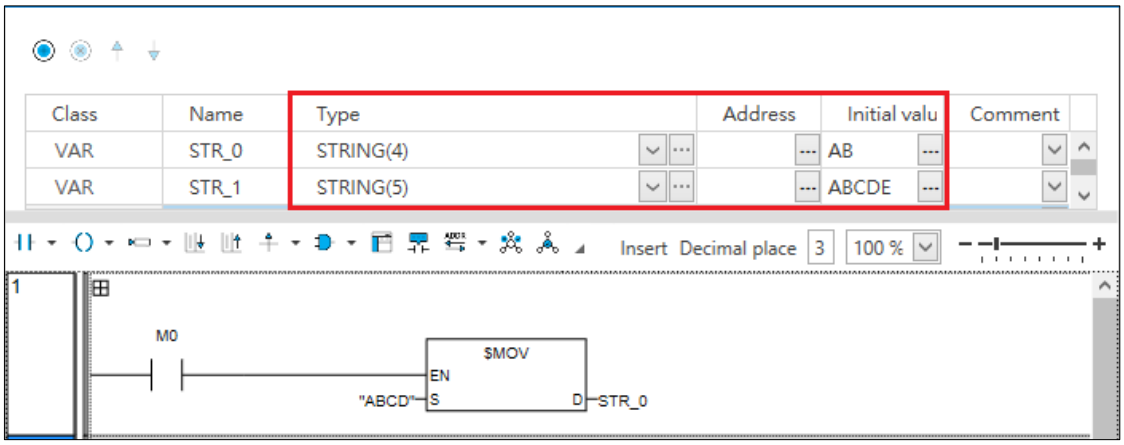


Figure 8 - 75: Initial Value attribute

Modifying a variable and editing a variable table:

If user want to modify the attributes of a variable, they can edit the variable table directly.

User can also modify the variable items in the variable table with the help of context menu as well as the functions in **Edit** menu.

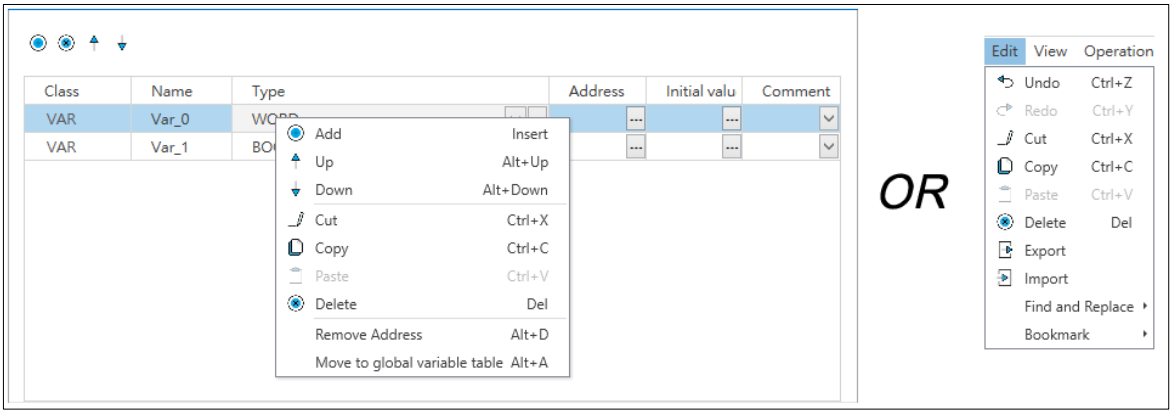


Figure 8 - 76: Edit Variables

The attributes in **Edit** Variables are described in the following table:

Item	Function
Undo	Undoing the previous action
Redo	Redoing the action which has been undone
Cut	Cut the variable selected
Copy	Copy the variable selected
Paste	Paste the variable which is cut/copied
Delete	**Delete the variable selected**
Up	Move the variable one level up
Down	Move the variable one level down

Remove the Register Address assigned to the variables:

User can clear all the device addresses assigned to the variables in a variable table, whether the register are assigned by the user or by the system. After the program is compiled again, the system will automatically assign addresses to these variables. Besides, only the device addresses in the variable table selected are cleared, the device addresses in the other variable tables are not cleared.

User can delete the Addresses assigned to variables in any of the below methods:

- Right click on the variables top open the context menu and click **Remove Address**

  or

- Press **Alt & D** keys simultaneously on the keyboard

  Result: Addresses are cleared in the Variable table of POU.

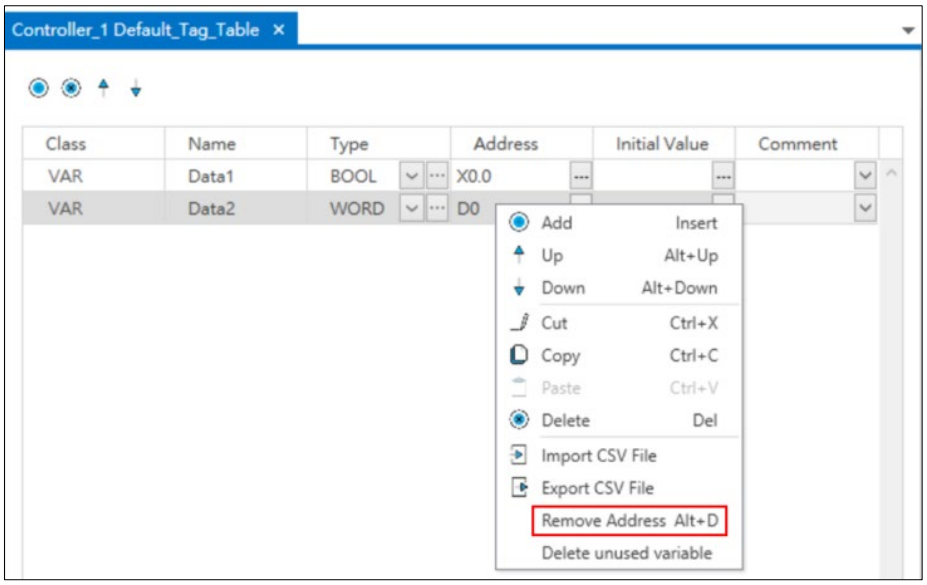


Figure 8 - 77: Removing Register Address

Downloading Initial Values of the variables:

User can set the initial values of a variable when the declare it. If a project is downloaded, the initial values of the variables can be written into the devices assigned to the variables. However, the initial values of the variables in a project are written into the devices assigned to the variables only when the project is downloaded. If a PLC is disconnected or stops running and then starts again, the values of the variables in the program in the PLC will not be the initial values. In order to ensure that the values of the variables in a PLC are the initial values whenever the PLC begins to run, it is suggested that user download the initial values of the variables. Besides, when the online edit is executed, the system does not download the initial values of the variables.

Follow these steps to write the initial value of variable to a PLC while downloading a program:

1. Click **Project Download** after the program is compiled without error.
2. Select **Initial Value** feature.
3. Click **Start Transmission.**

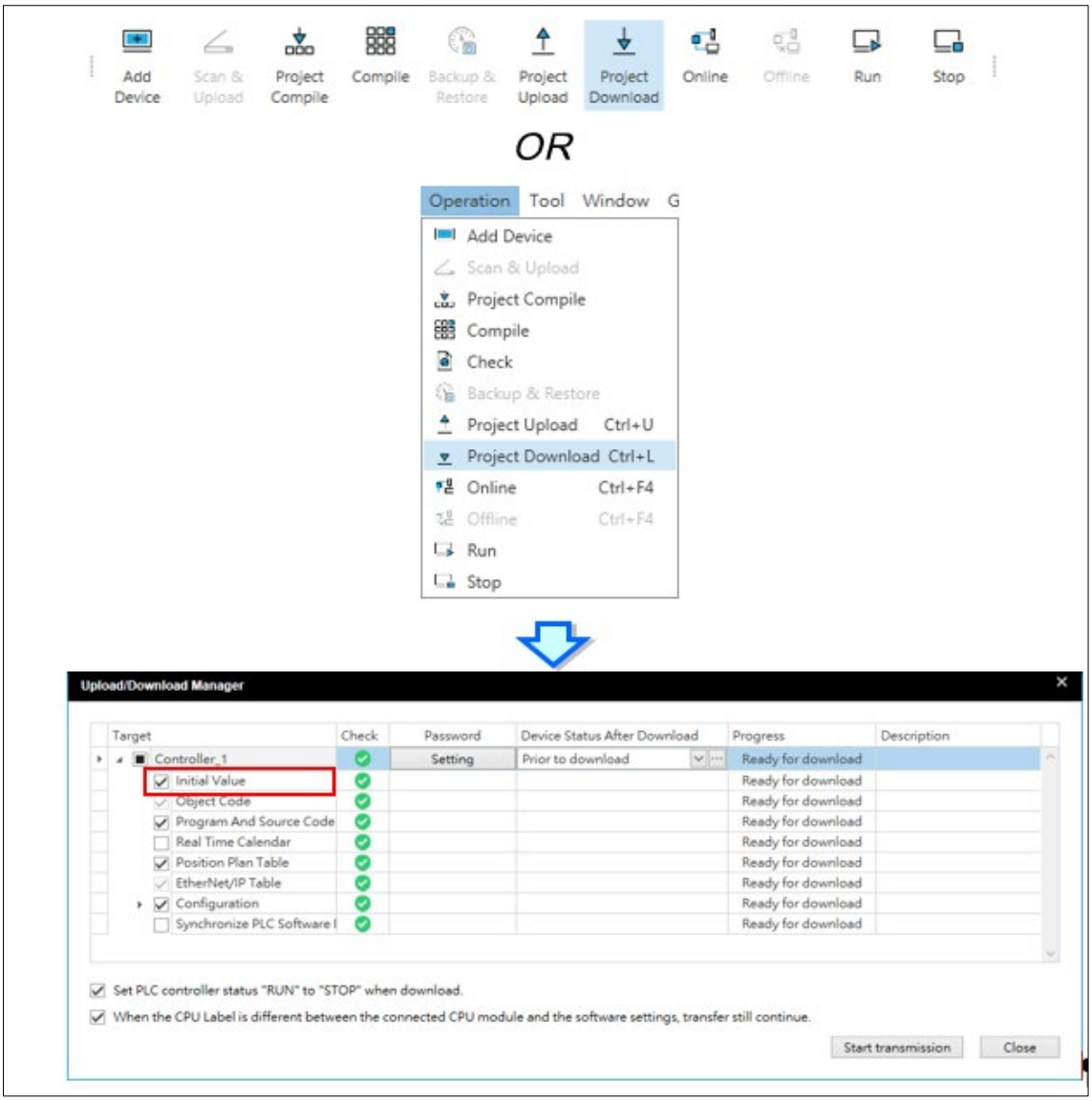


Figure 8 - 78: Download - Initial Value of Variables

Exporting a Variable Table:

The variables created in a project can be exported as CSV files, which can be opened in Excel. User can edit/manage the exported data.

To export variables, right click the variable table, and then select **Export CSV file** in the context menu.

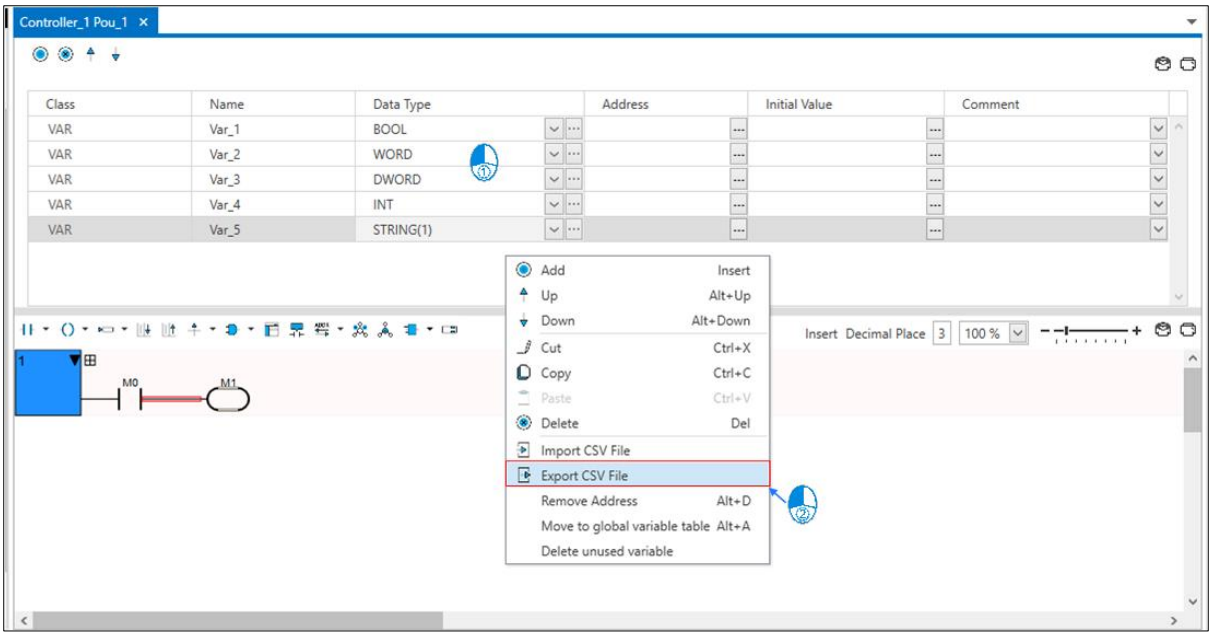


Figure 8 - 79: Export CSV File

There are two methods to export global variable table:

(1) Menu bar

Click **Edit > Export** to export the global variable table of the project that is currently editing.

(2) Project explorer

Right-click the project node that is currently editing and select **Export** in the context menu.

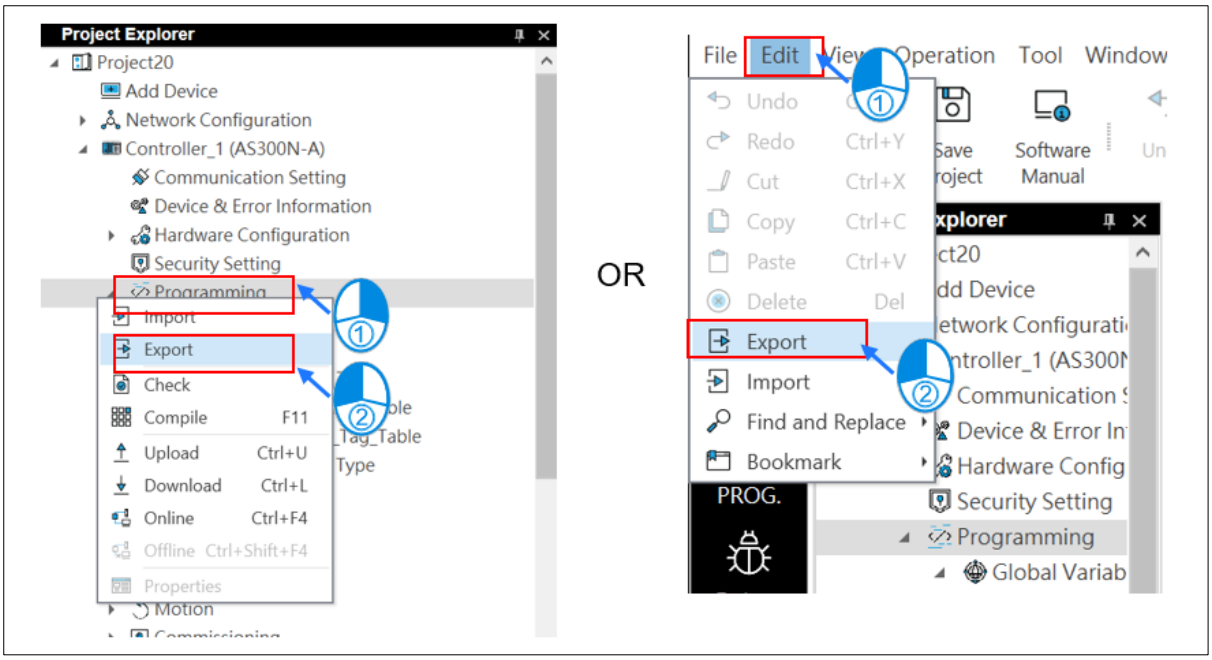


Figure 8 - 80: Export Global Variable Table

Importing a Variable Table:

To import variables, right click the variable table, and then select **Import CSV file** in the context menu.

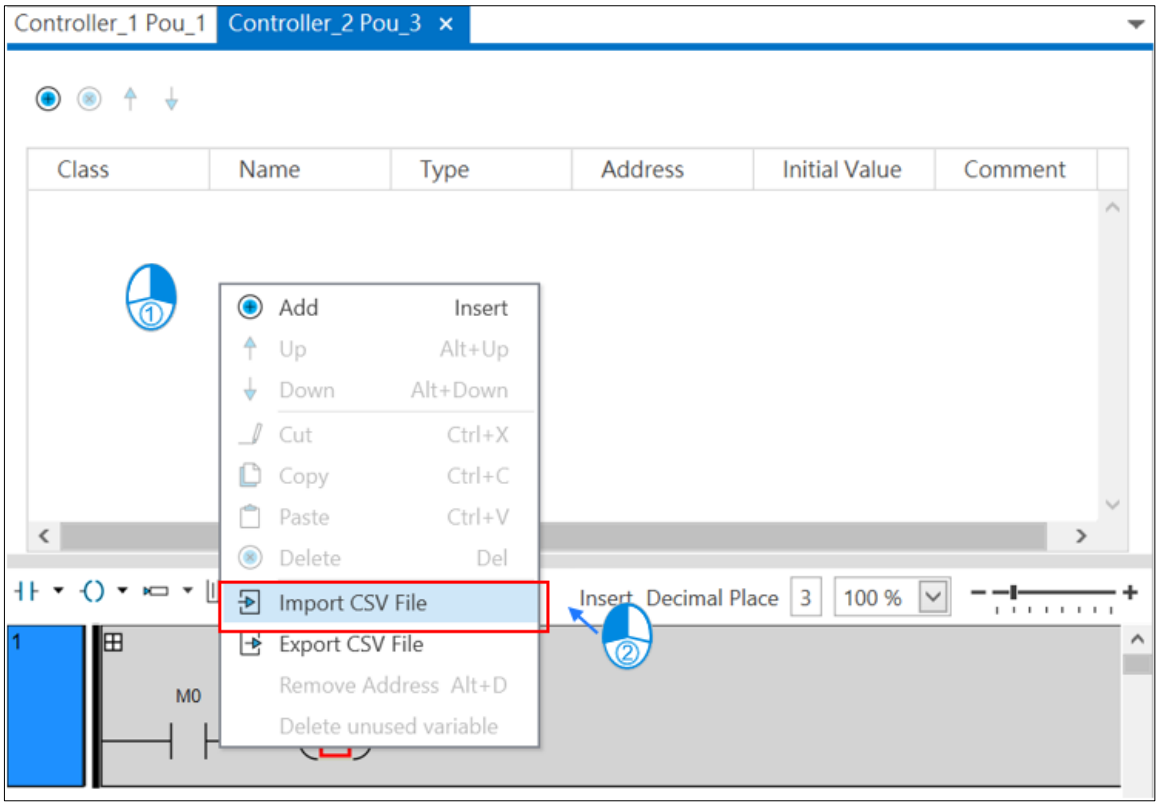


Figure 8 - 81: Import CSV File

There are two methods to import global variable table:

(1) Menu bar

Click **Edit > Import** to import the global variable table of the project that is currently editing.

(2) Project explorer

Right-click the project node that is currently editing and select **Import** in the context menu.

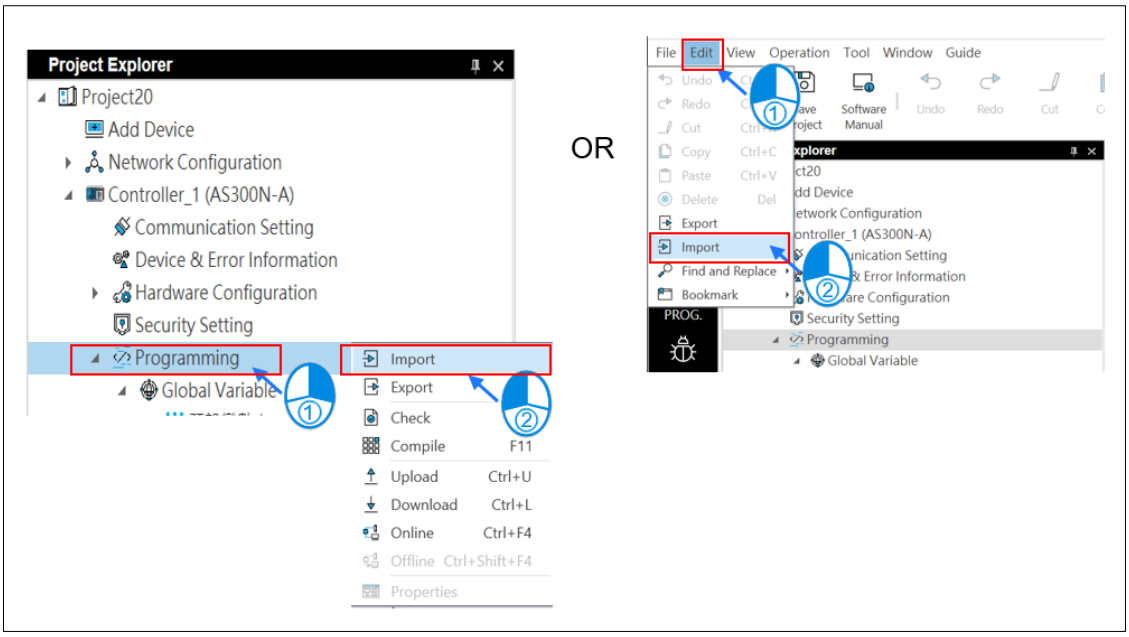


Figure 8 - 82: Import Global Variable Table

Arranging the Variables:

In the local variable table in a function block, the order in which the variables of the VAR_INPUT class, the variables of VAR_IN_OUT class and the variables of the VAR_OUTPUT class are arranged affects the order in which the operands in the function block are arranged. If user want to arrange the variables in the local variable table in a function block, they can right click a variable or use the Up/Down button and then move it up or down.

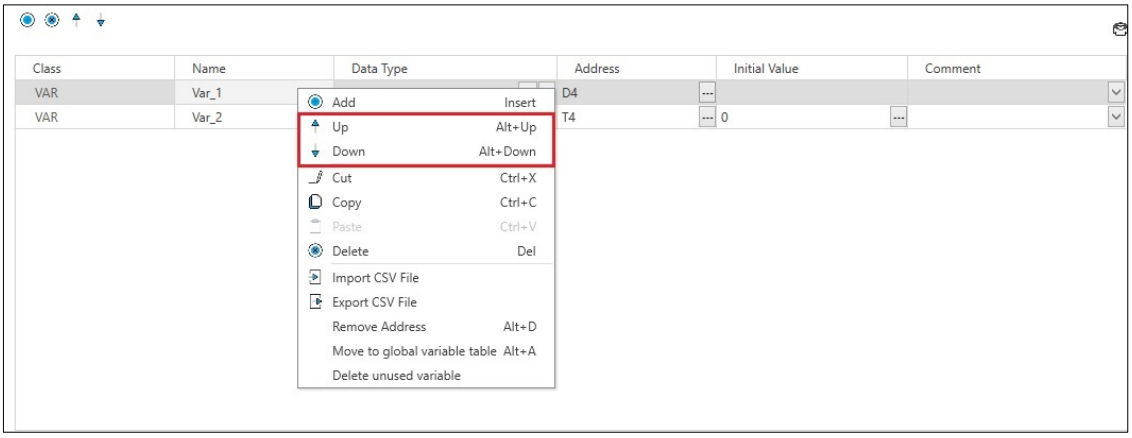


Figure 8 - 83: Arranging Variables

8.2.2.3 TASK

Every POU of the program type is assigned to a specific Task. Tasks determine the order in which the POUs of the program type are executed or enabled. Task is like a mission.

Managing the Tasks in DIADesigner:

In DIADesigner, not all the POUs of the program type in the project are executed. A POU of the program type is executed after it is assigned to a Task and the Task determines how the POU is executed. If a POU is not assigned to a Task, the POU is taken as a general source code and is saved with the project, but not to be translated into an execution code. Only the POUs of program type needs to be assigned to Tasks, Function Blocks are called by POUs.

The characteristic of the IEC 61131-3 architecture is that a program is divided into several POUs which can be developed independently. When all the POUs are compiled, they are rearranged and combined into an execution code which can be scanned step-by-step. The POUs are rearranged and combined to the tasks to which the POUs are assigned.

Task Management:

In DIADesigner, a POU of the program type is assigned to a task. More than one POU can be assigned to the same task and the order in which these POUs are executed can be specified. Besides, after user add a POU of the program type, they must assign it to a default task. There are three types of tasks:

- Freewheeling
- Cyclic
- Event

Freewheeling Task:

A POU assigned to freewheeling task is scanned back and forth. Owing to the fact that there is only one freewheeling task in a project for a PLC, the POUs which are scanned are assigned to this freewheeling task. However, there are 32 freewheeling tasks in a project, which are numbered from 1 to 31. The smaller the task number, earlier the task is executed. User can enable or disable a task during the operation.

A POU which is made with Sequential Function Chart (SFC) can only be assigned to a freewheeling task. It cannot be assigned to an Interrupt task.

Cyclic Task:

A POU of the program type which is assigned to a cyclic task is like a cyclic timed interrupt subroutine. If the time when cyclic should occur is reached, the POUs assigned to the cyclic task are executed in order. Besides, the number of the cyclic tasks depends on the PLC selected. The number of cyclic sources that a PLC supports is the number of cyclic tasks that the system has.

Event Task:

There are several types of event tasks. For example, external interrupts, I/O interrupts, counting interrupts etc. Different PLCs provide different event tasks. User have to make sure of the conditional interrupt tasks which are supported by the PLC before they assign the POUs of the program type to tasks. A POU of the program type which is assigned to an event task is like an interrupt subroutine. If the condition of an event is met, for example, if the value of a counter reaches the set point, the POUs assigned to the interrupt task are executed in order.

Tasks in Project Explorer area:

The POUs which are assigned to tasks are listed in the Task Manager in the **Project Explorer** area. In the Project Explorer, double-click the **Task Manager**.

Result: Task Manager window displays as shown in the following figure.

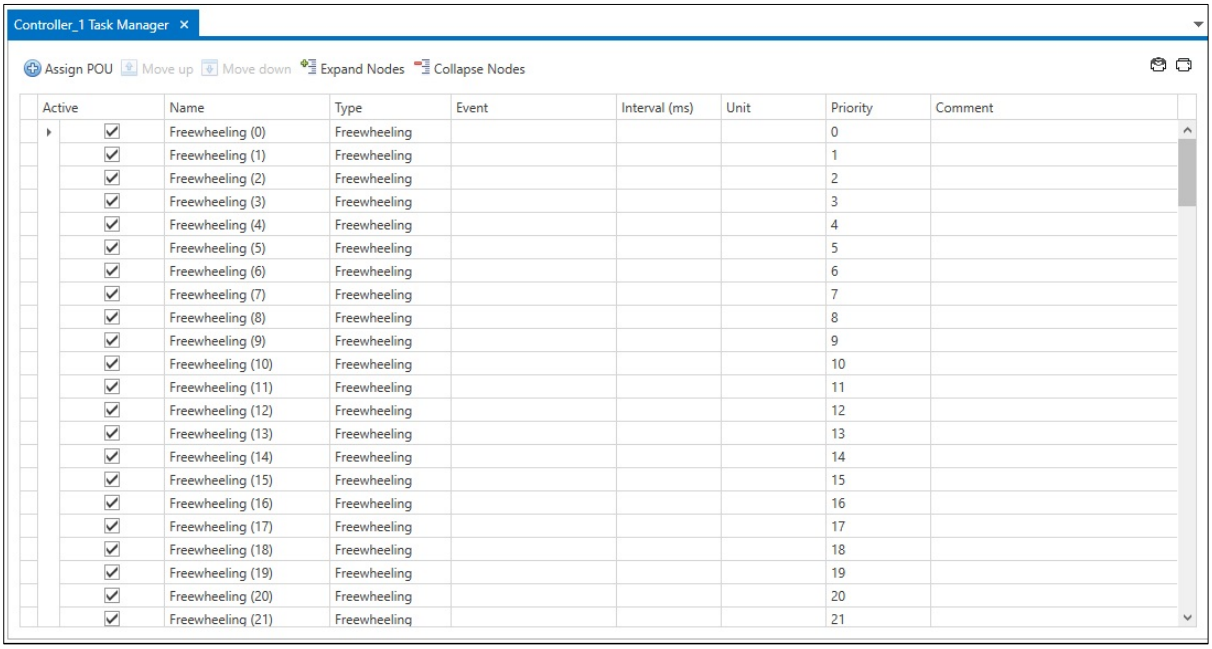


Figure 8 - 84: Task Manager

Task Manager window displays the Tasks in PLC. For AS PLC, the tasks applicable are:

- Cyclic
- External Input Interruption
- Hardware High-Speed Compared Interrupt
- Communication Interrupt
- External Interrupt
- High Speed Output Interruption
- Timed Interruption

All the tasks display with their names, type, event type, priority and comments.

Follow these steps to associate a program type POU to a Task:

1. Click on a particular task to which the POU has to be assigned.
2. Click **Assign POU** in **Task Manager** window

   or

   Right click on a particular task to open the context menu and click on **Assign POU.**

   Result: The **Assign POU** pop-up displays as shown in the following figure and it displays the list of program type POUs in the device which are not yet assigned to any task.

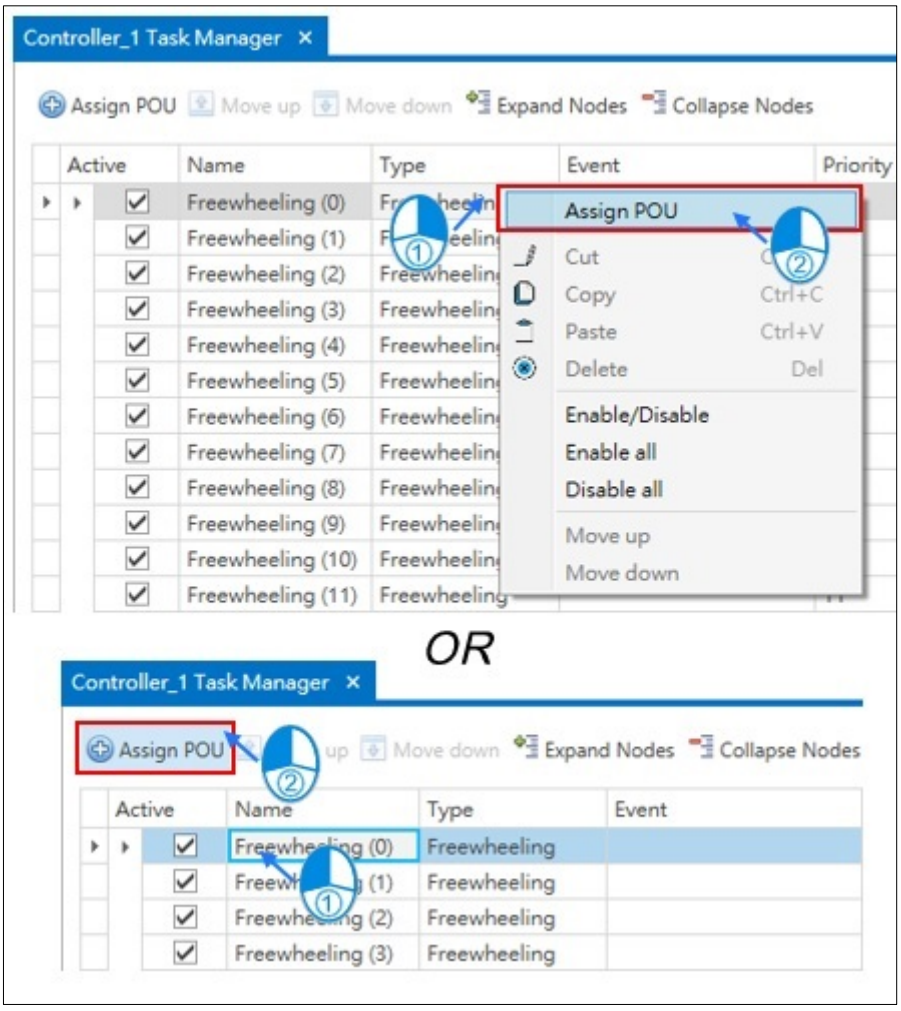


Figure 8 - 85: Assign POU

3. Select the necessary POU from the drop-down list and click **OK** to assign the POU to the task selected.

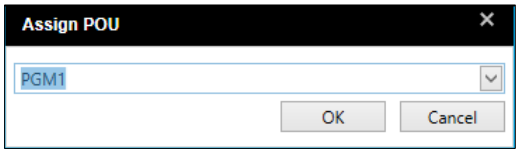


Figure 8 - 86: Assign POU window

Result: POU is assigned to the task selected.

The Toolbar available in **Task Manager** window is shown in the following figure.

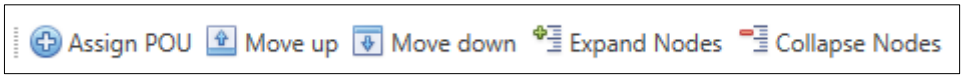


Figure 8 - 87: Task Manager - Toolbar

The functions available in the Task Manager window are described in the following table:

Function	Description
Assign POU	Select a Task and click Assign POU to assign the POU to that Task
Move up	Select a POU assigned to a Task and click Move up one row icon to execute the current POU before the earlier one
Move down	Select a POU assigned to a Task and click Move down one row icon to execute the current POU after the below one
Expand Nodes	**Click to expand all nodes**
Collapse Nodes	Click to collapse all nodes

Arranging the POU order:

User can arrange the POU order (order in which POU is executed in PLC runtime), by rearranging them with the help of Move up or Move down one functions in the **Task Manager** window.

Follow these steps to re-arrange a POU to be executed before another one:

1. After creating several POUs, assign them to suitable Task in Task Manager.
2. Click on a POU assigned to the above Task, example – PGM_2.
3. Click on the Move up function to re-arrange (make PGM_2 executed before PGM_1) the order as shown in the following figure.

   Result: POU orders are re-arranged (PGM_2 will now be executed before PGM_1).

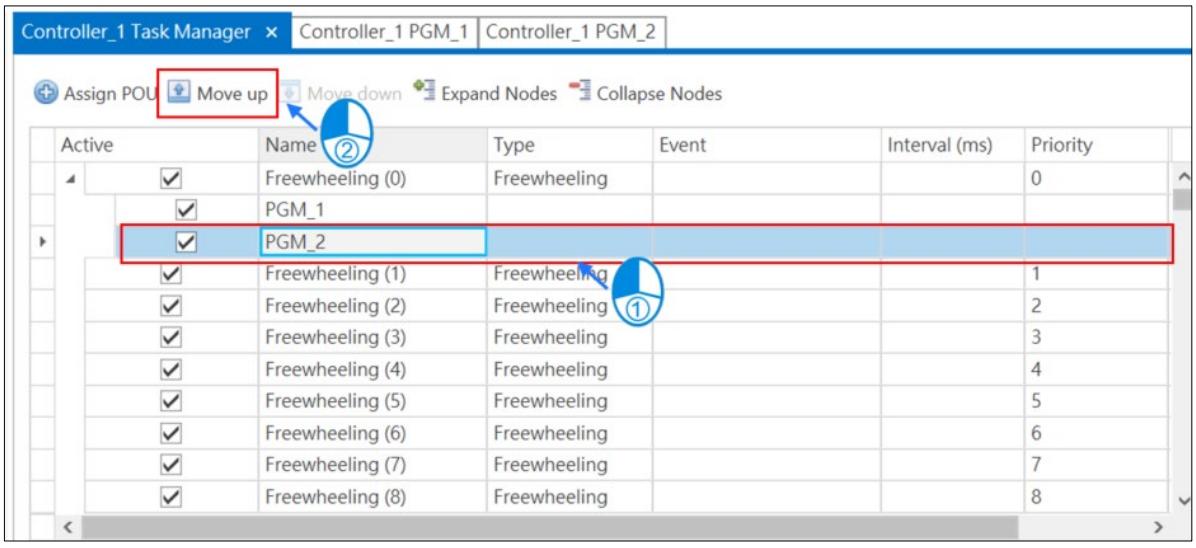


Figure 8 - 88: Move Up

Follow these steps to re-arrange a POU to be executed after another one:

1. After creating several POUs, assign them to suitable Task in Task Manager.
2. Click on a POU assigned to the above Task, example – PGM_2.

3. Click on the Move Down function to re-arrange (make PGM_2 executed after PGM_1) the order as shown in the following figure.

   Result: POU orders are re-arranged (PGM_2 will now be executed after PGM_1)

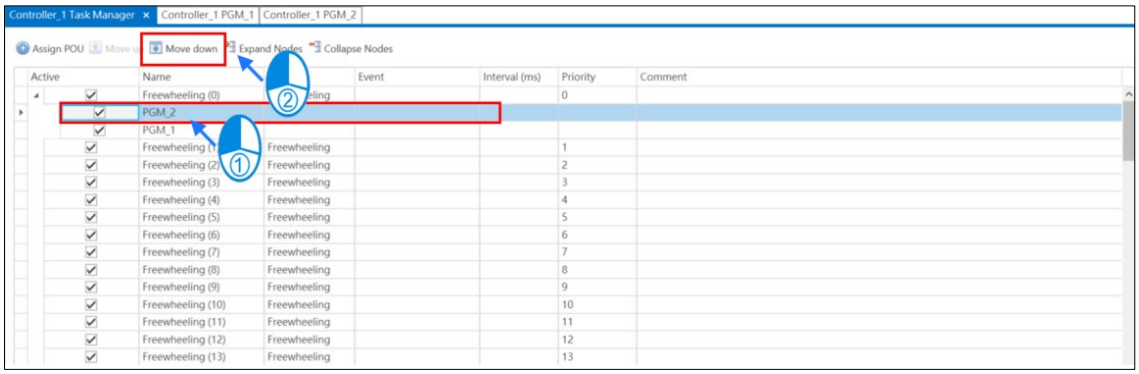


Figure 8 - 89:Move Up

8.3 Programming Language Introduction (LD, SFC, ST)

8.3.1 LD – Ladder Diagram

The Ladder Diagram (LD) is one of the programming languages defined by IEC 61131-3 and is widely used to create a PLC program. LD 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 DIADesigner displays as shown in the following figure.

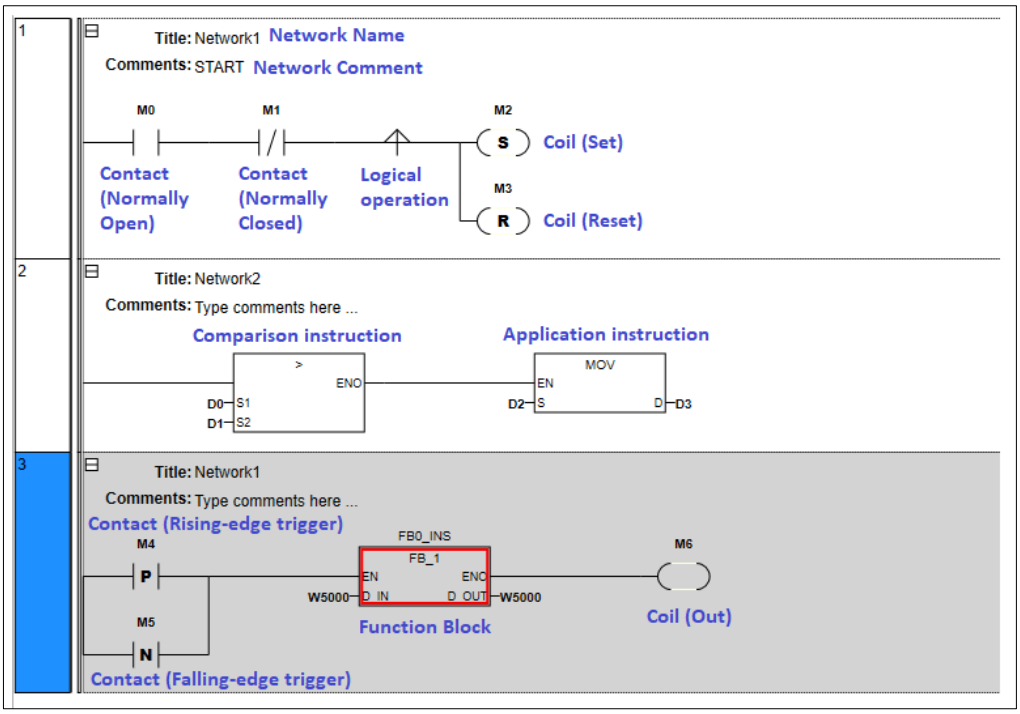


Figure 8 - 90: LD Program

8.3.1.1 Add LD Program

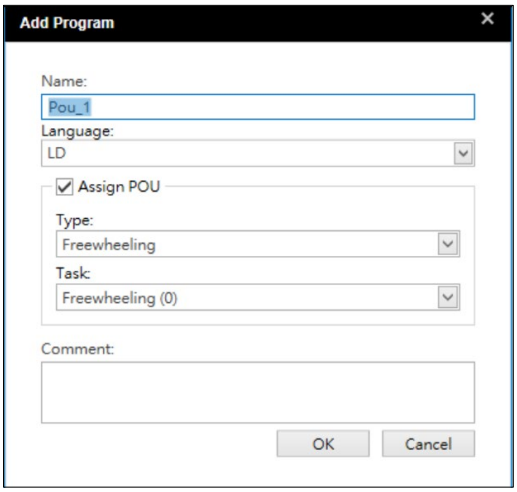


Figure 8 - 91: Add LD POU

User can select LD in the Language field in Add POU window while creating a POU of program type or function block type.

The environment in which a LD POU can be edited is shown in the following figure. In any POU window (program type or function block type), there are two areas:

- Local Variable table area
- POU Working area

The table in the upper part of the LD POU window is a Local Variable Table and the lower part is POU working area. After a LD POU program editing window is opened, the corresponding tool bar will appear in the DIADesigner window. A blank Network is created by default.

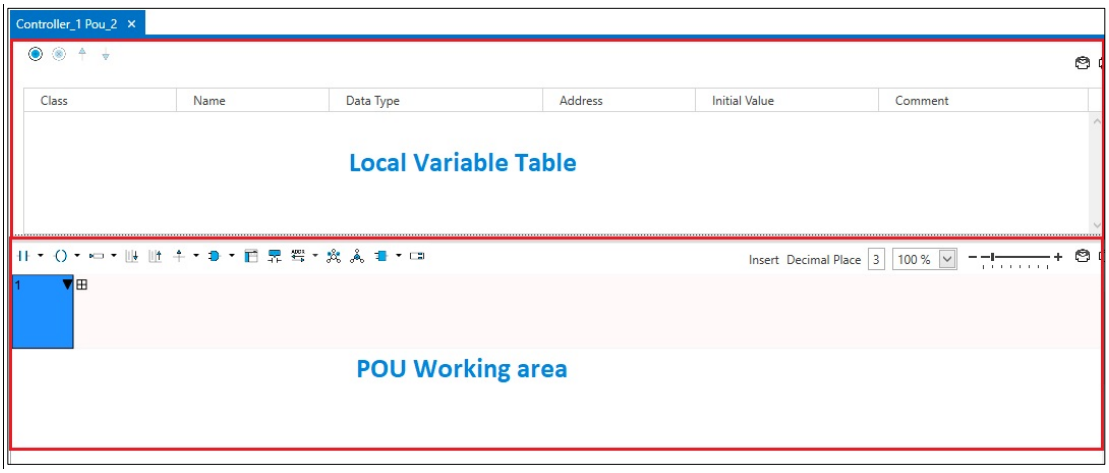


Figure 8 - 92: LD POU - Local Variable Table and POU working area

8.3.1.2 LD Elements

After the POU window is opened, the toolbar for editing the ladder diagram displays as shown in the following figure.

Figure 8 - 93: LD POU Toolbar

Editing Environment:

The functions in LD POU window are described in the following table:

Function/Icon	Description
	Contact (Normally Opened)
	Contact (Normally Closed)
	Contact (Rising-edge)
	Contact (Falling-edge)
	Insert Parallel Contact (Under)
	Insert Parallel Contact (Above)
	Coil (Output)
	Coil (Set)
	Coil (Reset)
	Coil (Not) ***NOTE:*** *The function works only for certain devices.*
	Insert Parallel Coil (Under)
	Insert Parallel Coil (Above)
	Jump
	Insert and jump (Under)
	Insert and jump (Above)
	Insert Network (Under)
	Insert Network (Above)
	Trigger (Rising-edge)
	Trigger (Falling-edge)
	Trigger (Not transition)
	Insert Parallel Trigger (Under)
	Insert Parallel Trigger (Above)
	FB Trigger (Positive Transition-sensing)
	FB Trigger (Negative Transition-sensing)
	Insert Parallel FB Trigger (Under)

Function/Icon	Description
	Insert Parallel FB Trigger (Above)
	& and Logical operators
	Show/Hide Network Comment
	Show/Hide Register Comment
	Mode of display – Address, Variable, Variable and Address
	Enable/Disable selected Networks
	Enable all Networks
	Insert an Instruction/Block
	Insert Parallel Instruction (Under)
	Insert Parallel Instruction (Above)
	Label
	Click to start Debug mode

8.3.1.3 Edit LD

Networks in a Ladder Diagram:

A Ladder Diagram consists of Networks. Every Network is an independent program. There is no limit on the number of objects which can be inserted in a Network in DIADesigner and therefore there is no mark which is used to connect two Networks.

Network names display in Ladder Diagram in DIADesigner. The color at the left side of a network indicates the state of the network. A network can be enabled or disabled and a network can be selected or unselected. By default, the Network created is activated. If user wants to disable a network, they can select them and click on the icon to disable them. Click the icon again incase to enable the disabled network.

Figure 8 - 94: Network disable and enable icon

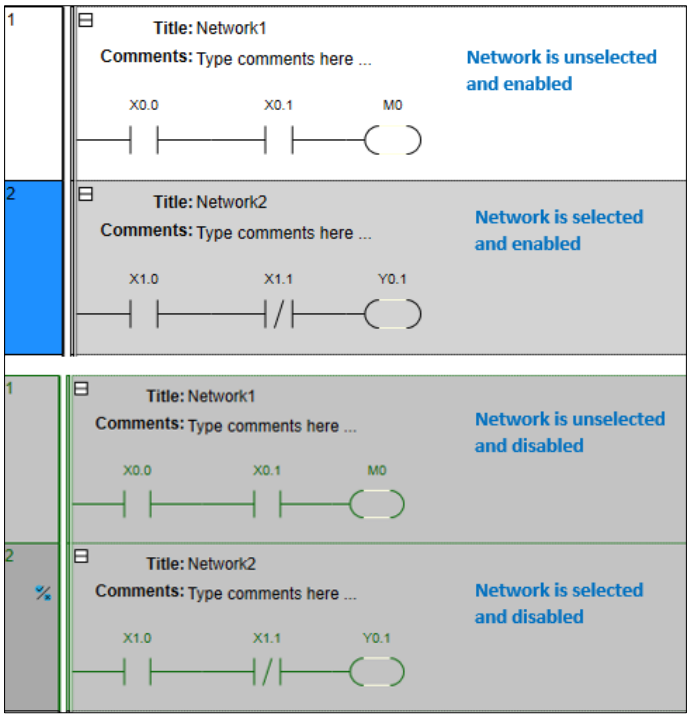


Figure 8 - 95: LD POU - Network status

Color of the network number indicates the enable/disable and selected/unselected network as mentioned in the following table:

Selection	Enable/Disable	Network Color (Left Side)
Not Selected	Enabled	White
Selected	Enabled	Blue
Not Selected	Disabled	Grey
Selected	Disabled	Dark Grey

Add Network:

When a POU of program type or function block type with LD is opened, the system inserts a blank network in the POU window.

Add Network below the current Network:

If user want to insert a network below the current network, they can select the network and click icon in the toolbar.

Result: New Network is added below the current network.

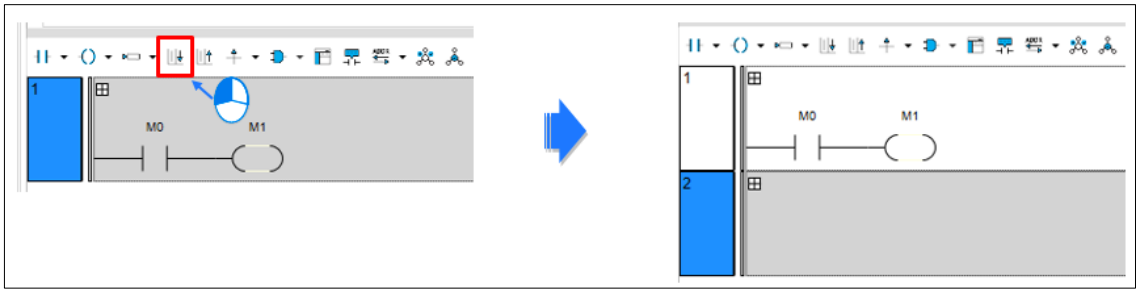


Figure 8 - 96: LD POU - Insert a Network below the current Network

Add Network above the current Network:

If user want to insert a network above the current network, they can select the network and click icon in the toolbar.

Result: New Network is added above the current network.

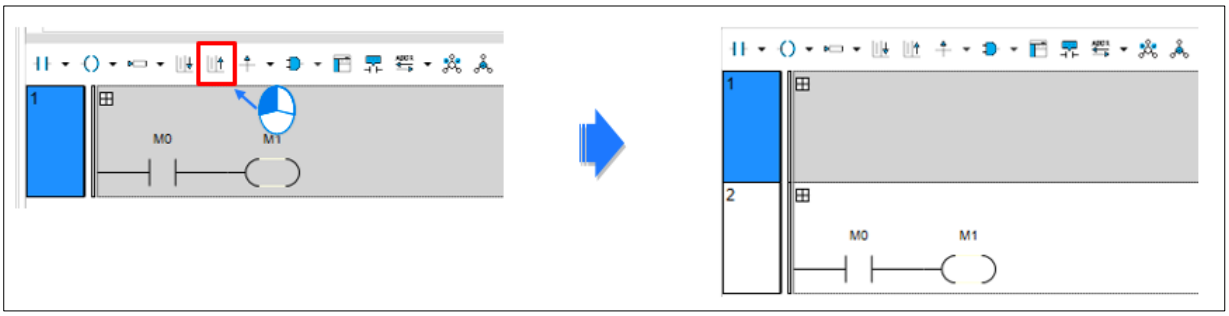


Figure 8 - 97: LD POU - Insert a Network above the current Network

Selecting Objects:

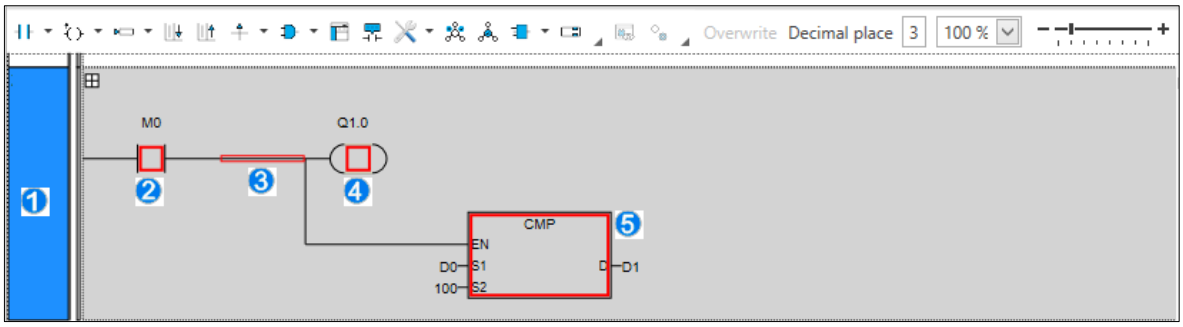


Figure 8 - 98: LD POU - selecting objects

1. Selecting the Network
2. Selecting the Input Contact
3. Selecting the Network
4. Selecting the Output Coil
5. Selecting the block

If user want to select many networks, they can hold down **Ctrl** key on the keyboard while they click the Networks. User can also select a range of networks by pressing **Shift** key on the keyboard, clicking the first network within the range and the last network within the range.

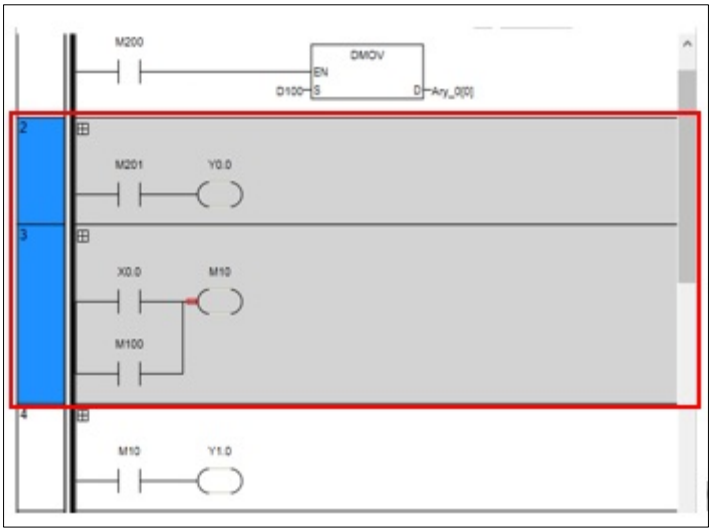
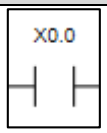
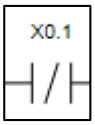
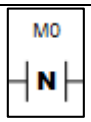
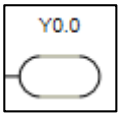
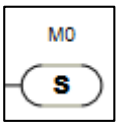
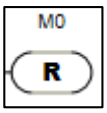


Figure 8 - 99: LD POU - working area elements

Creating a Contact and Coil:

Ladder Element	Description	Representation
Contact	Normally-Open Contact (NO)	
	Normally-Closed Contact (NC)	
	Rising Edge-Triggered Contact (RE)	
	Falling Edge-Triggered Contact (FE)	
Coil	OUT	
	SET	
	RESET	

Inserting a Contact and changing a Contact Type:

After user make sure of the position in which a contact is inserted, they can click on **Contact** icon on the toolbar as shown in the following figure.

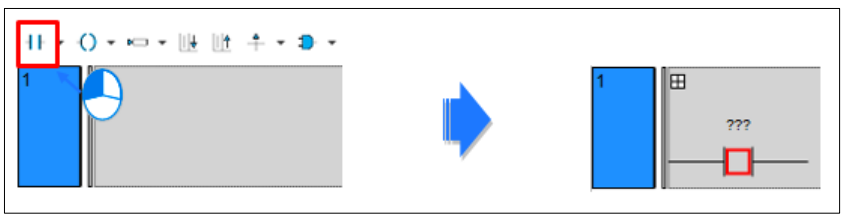


Figure 8 - 100: LD POU - inserting a Contact

After creating a contact, user can drag the contact to different locations as shown in the above figure.

- right side of a contact
- left side of a contact
- bottom of a contact

The new position in which it can be inserted is shown by the arrows in those positions. Position the cursor on the arrow, so that the arrow displays in yellow color. Release the cursor to insert the LD element in that position. The inserted element can be dragged to a new position as shown in the following figure

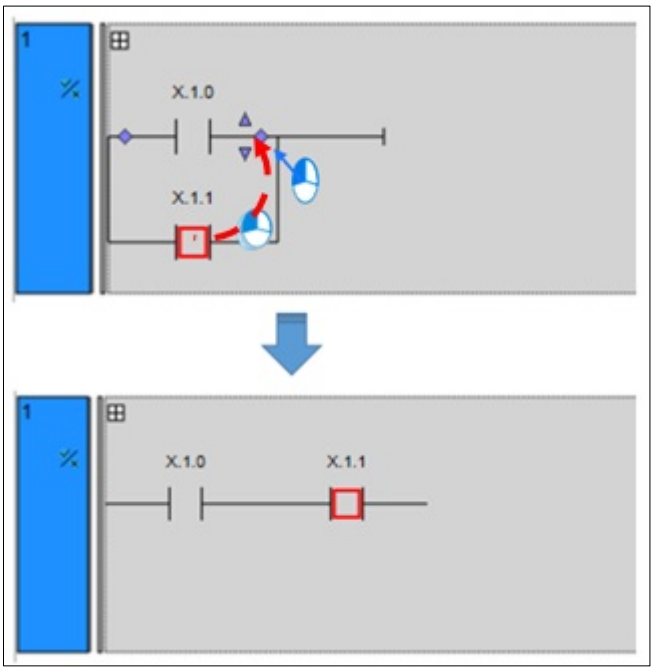


Figure 8 - 101: LD POU - drag the Contact

User can click an element which is inserted previously and click again to insert a contact as shown in the following figure.

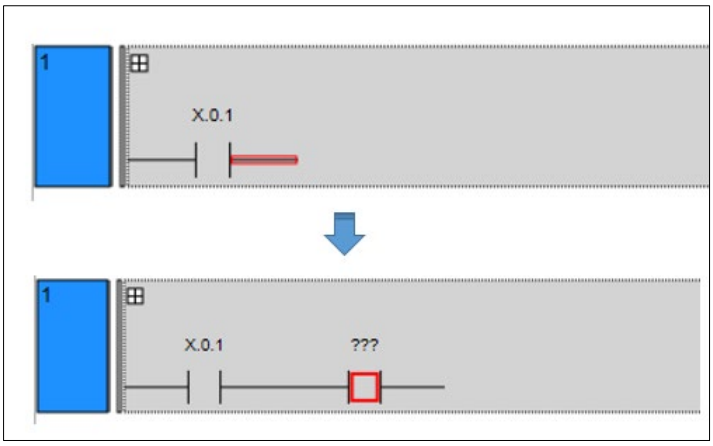


Figure 8 - 102: LD POU - continue inserting Contact

To change a contact type to a different one, user can right-click on a contact and select the contact type from the context menu according to the requirement.

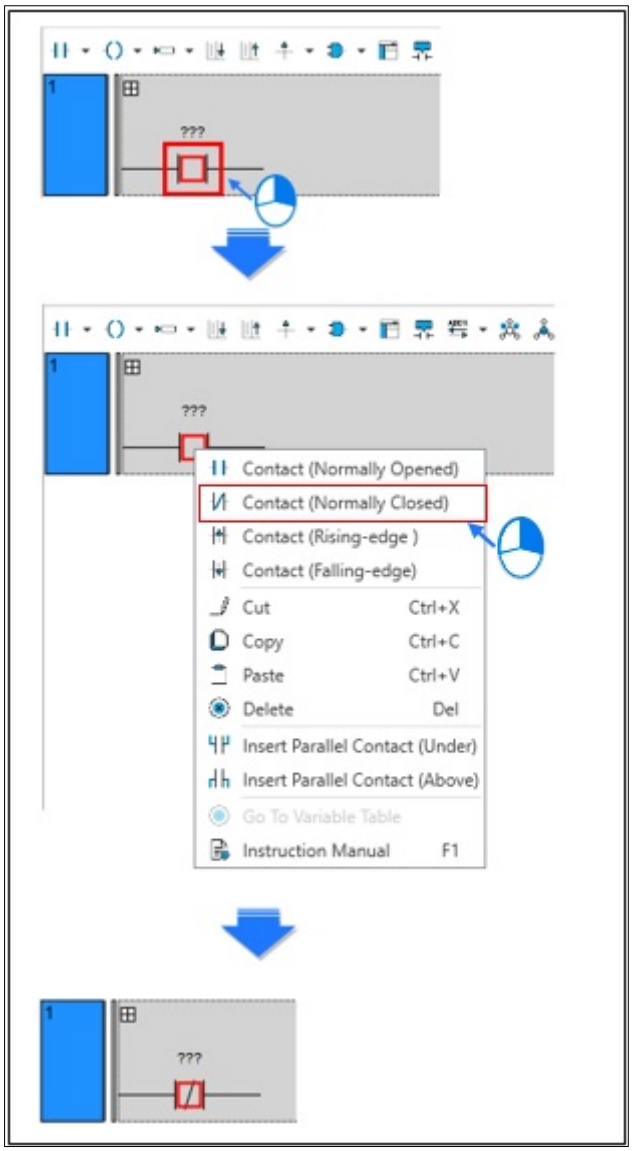


Figure 8 - 103: LD POU - change Contact Type

Inserting a Coil and Changing a Coil type:

Select a line, a contact, a coil or an applied instruction. After user make sure of the position in which a coil is inserted, they can click ( ) icon on the toolbar as shown in the following figure.

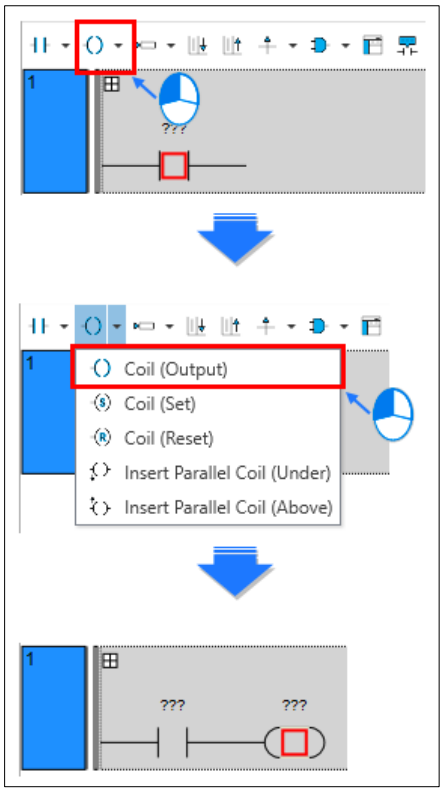


Figure 8 - 104: LD POU - insert Coil

To change a coil type to a different one, user can right-click on a coil and select the coil type from the context menu according to the requirement.

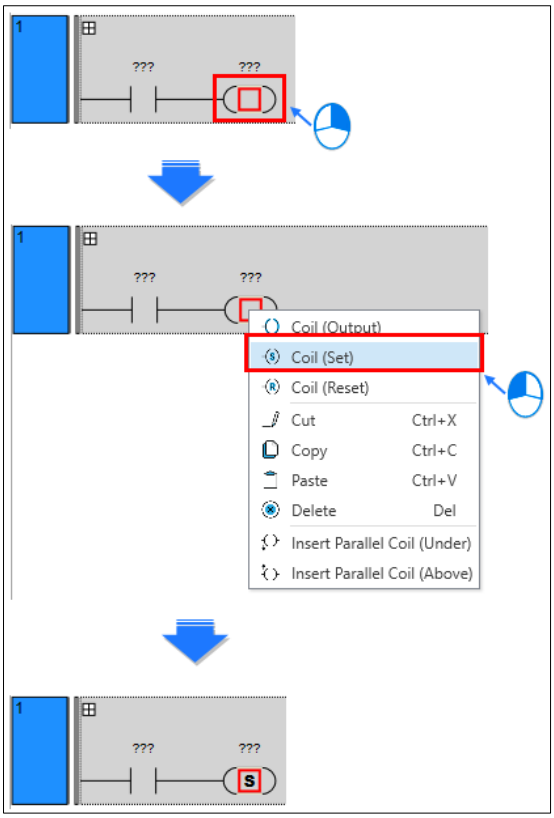


Figure 8 - 105: LD POU - Change Coil Type

Vertical Operation:

DIADesigner provides intuitive way to add or remove vertical line in LD logic. Please follow these steps.

Add Vertical Line:

1. Click on the horizontal line of logic.
2. Drag & drop to the next side of horizontal line to connect between both horizontal line as shown in the following figure.

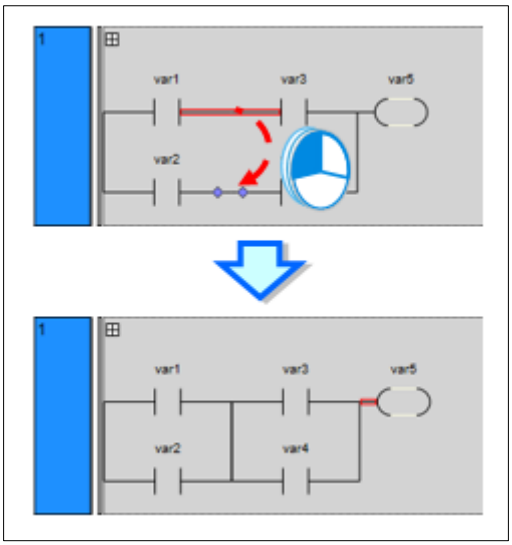


Figure 8 - 106: LD POU - Add Vertical line

Delete Vertical Line:

1. Click the vertical line.
2. Press **Delete** or click right key to open context menu then click **Delete**. The vertical line is deleted.

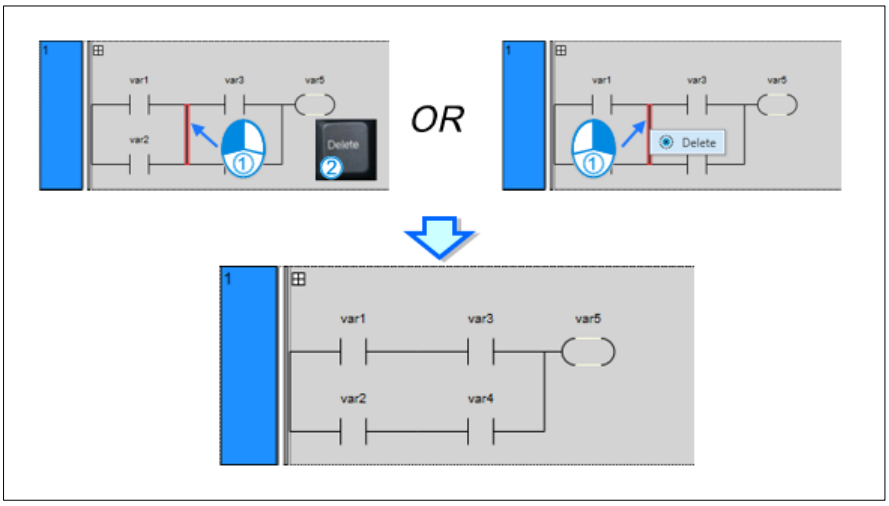


Figure 8 - 107: LD POU - Delete Vertical line

Using Address, Variables and Constants in a Ladder Diagram:

After user click ??? above an object inserted in a network, they can type an operand as shown in the following figure. Operand can be a Variable or Address.

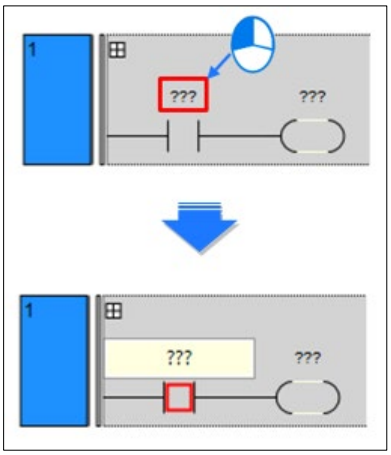


Figure 8 - 108: LD POU - add Variable or Address

User can click on ??? on any LD element and enter a Variable name or Address. Address can be typed directly. In case of Variables, there are two type of assignments:

- Assigning Variables created previously
- Enter a Variable name and declare it

Assigning Variables created previously:

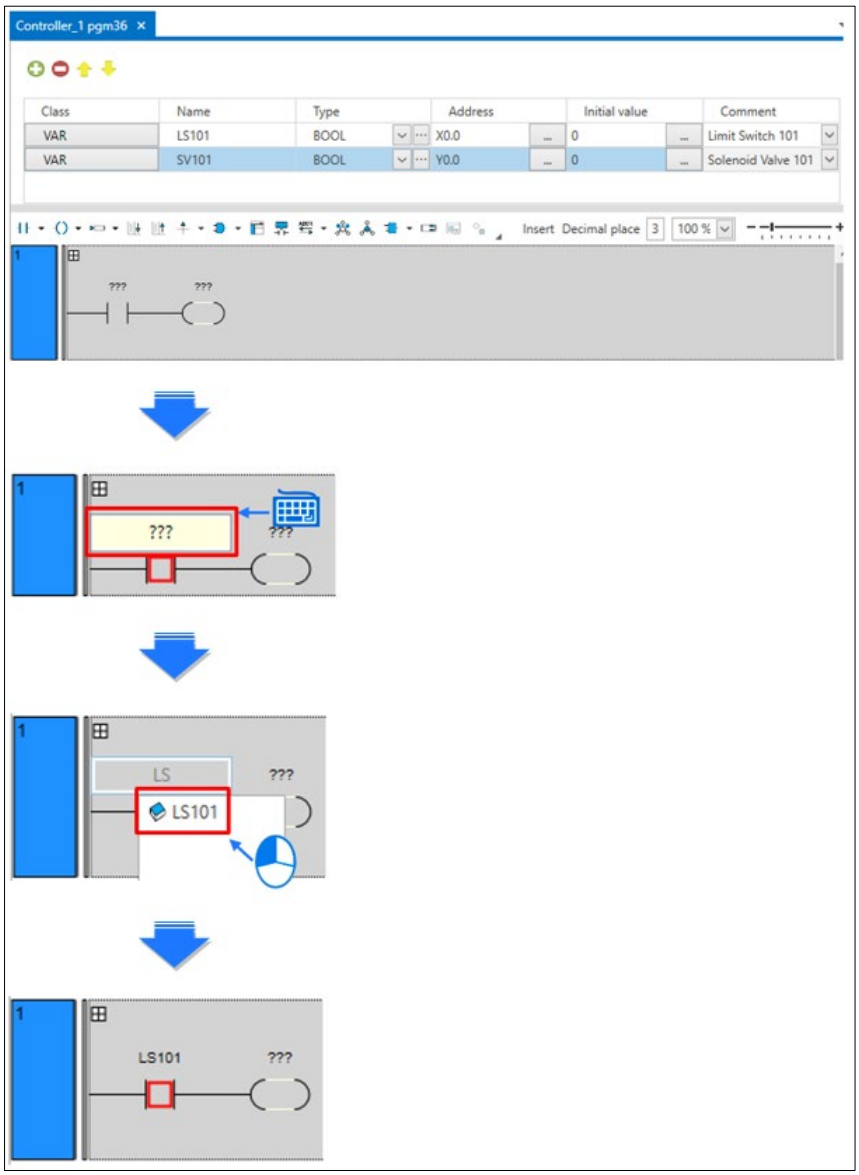


Figure 8 - 109: LD POU - assign Variables which are created previously

Enter a Variable name and declare it:

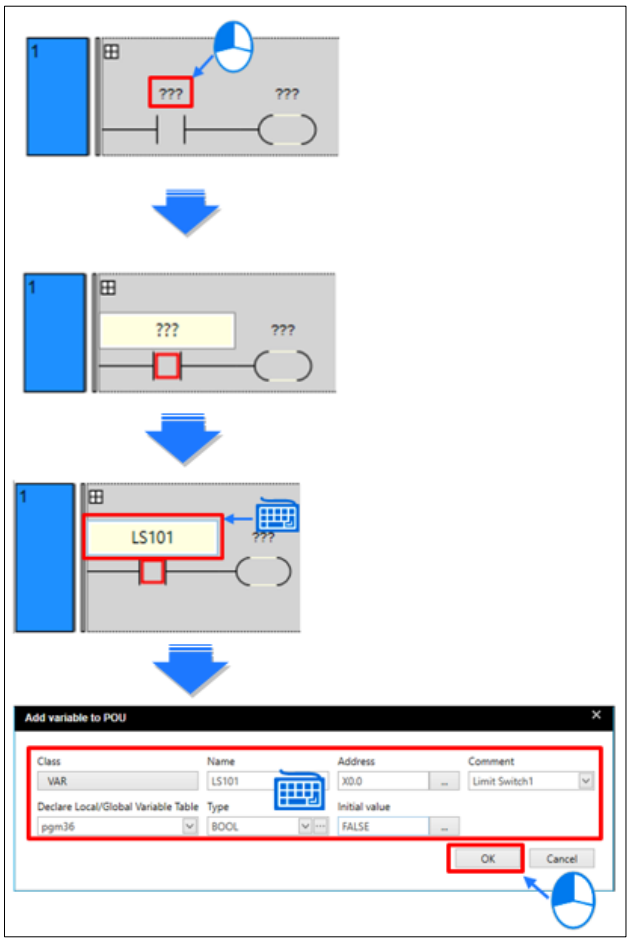


Figure 8 - 110: LD POU - create Variable

Additional remark:

If user want to use constants in a Ladder Diagram in DIADesigner, the constants must be represented in the following ways:

- Decimal value: 23456 (a value which is not preceded by any mark is regarded as a decimal value)
- Hexadecimal value: 16#5BA0 (a value which is preceded by 16# is regarded as a hexadecimal value)
- Octal value: 8#55640 (a value which is preceded by 8# is regarded as an octal value)

- Binary value: 2#1011 (a value which is preceded by 2# is regarded as a binary value)
- String value: “XYZ” (a value which consists of characters and inserted in double quotes is regarded as string value)
- Bool value: AS PLC SM400 (NO Contact) or SM401 (NC Contact) is used.

If user want to use a variable whose datatype is ARRAY, the expression format is 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 separated by a comma. For example, Ary[1,2,3]. Since the Ladder Diagram (LD) doesn’t support variable type index, the indexes in a ladder diagram cannot be variables.

Typing Instructions:

User can add instructions to a Ladder Diagram in DIADesigner by typing instructions. Follow these steps to insert instructions in a LD POU.

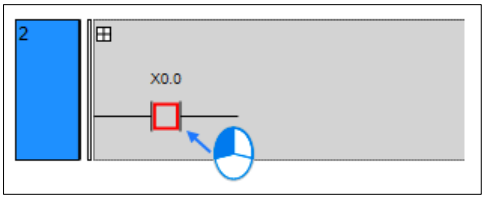


Figure 8 - 111: LD POU - click to select and insert

1. Select a position in which a LD object has to be inserted.
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, user can press **Enter** key on the keyboard or click **OK** at the right side of the editing box.

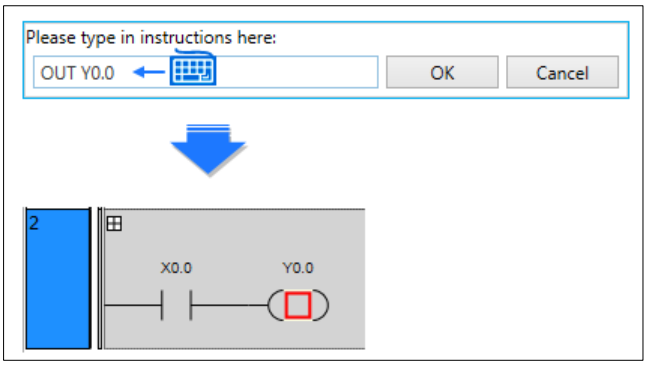


Figure 8 - 112: LD POU - type instructions

To insert a NO contact, type **A** followed by address and click **OK** in the editing box.

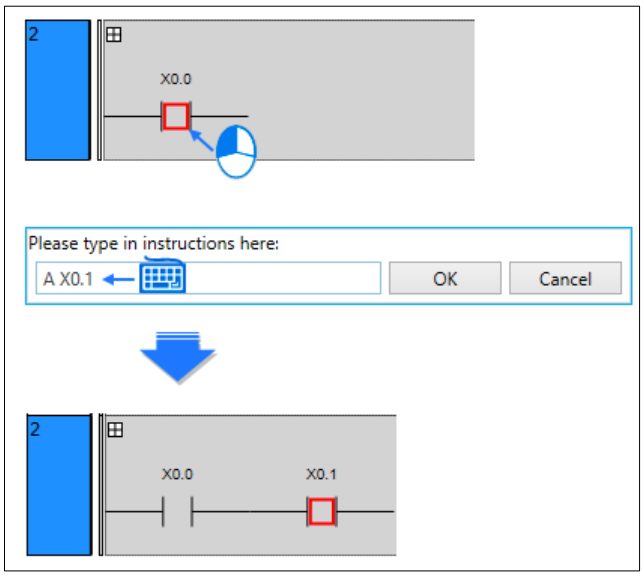


Figure 8 - 113: LD POU - insert NO Contact

To insert a NC contact, type “B” followed by address and click **OK** in the editing box.

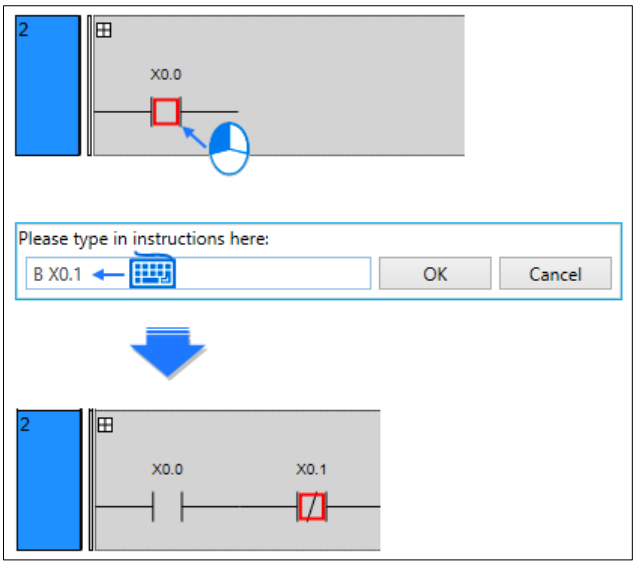


Figure 8 - 114: LD POU - insert NC Contact

Inserting Applied Instructions and Function Blocks:

The applied instructions and function blocks in a ladder diagram in DIADesigner 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 variables 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 of the function block is 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 cannot be followed by any object. Besides, user can type a function block instance in the box above a function block.

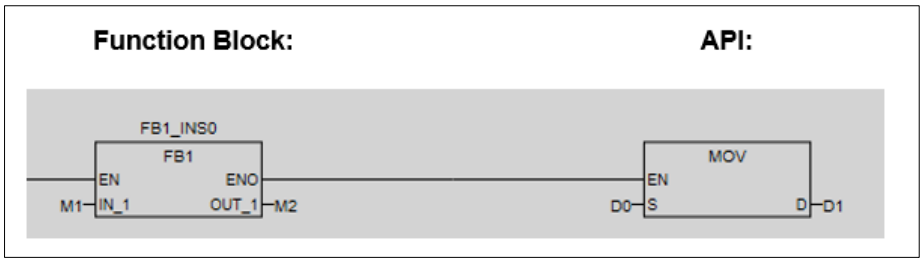


Figure 8 - 115: LD POU - Function Block and API

User can insert an applied instruction or a function block in a ladder diagram in DIADesigner in one of the three ways described below:

Method1:

Unfold the APIs section or the Function Blocks section in the Project Explorer and find the item which is inserted. Select the item and drag it to the position in which it is inserted as shown in the following figure:

Function Block:

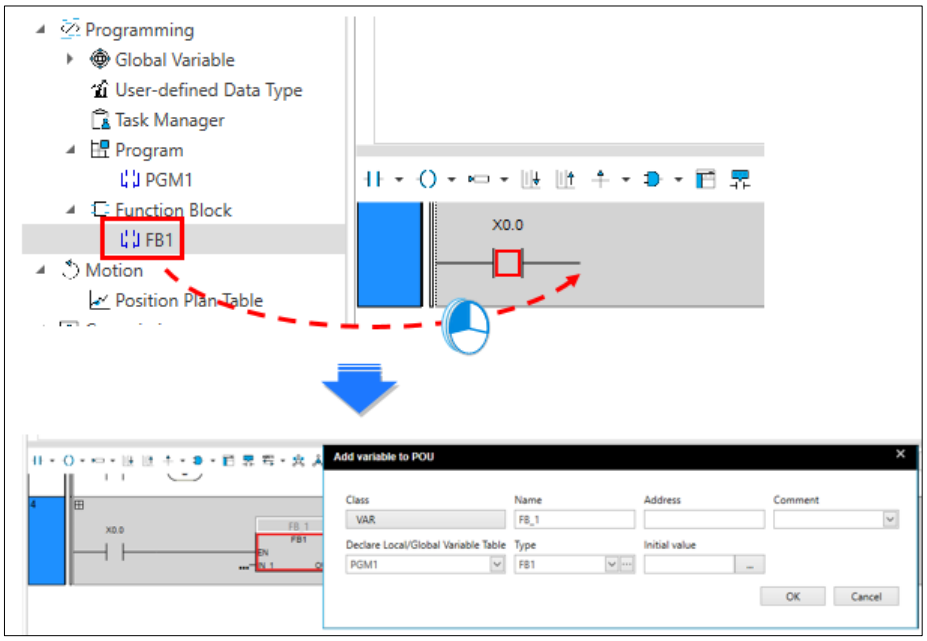


Figure 8 - 116: LD POU - drag-drop Function Block

API:

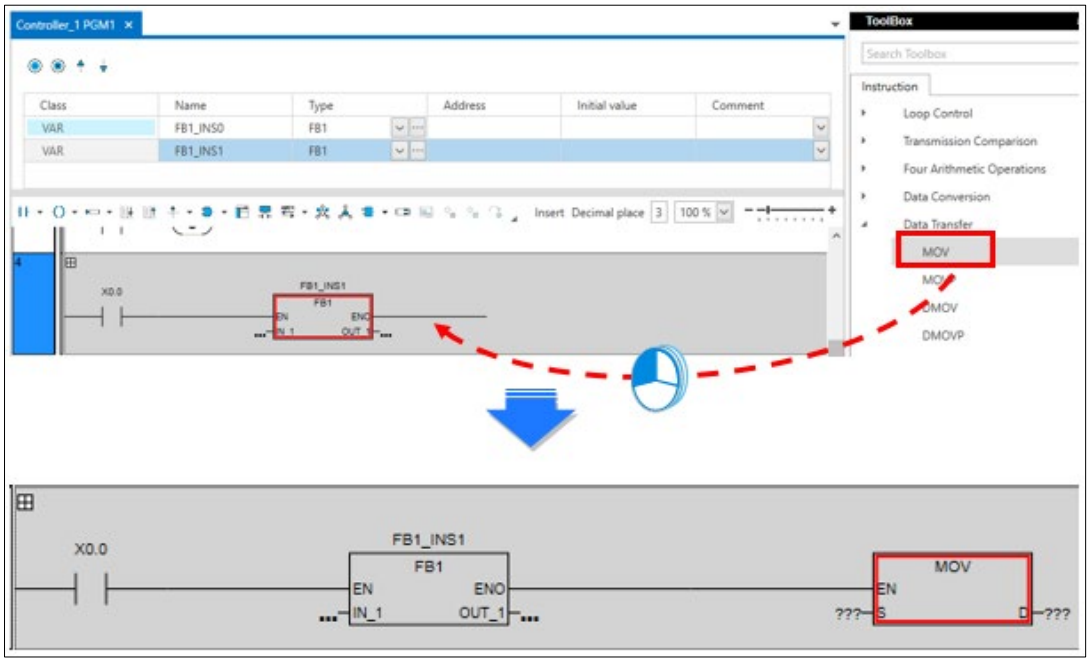


Figure 8 - 117: LD POU - drag-drop API

Method2:

Click the position in which an applied instruction or a function block is inserted, click the icon on the toolbar to insert an empty block, enter an instruction or function block name in the added block as shown in the following figure.

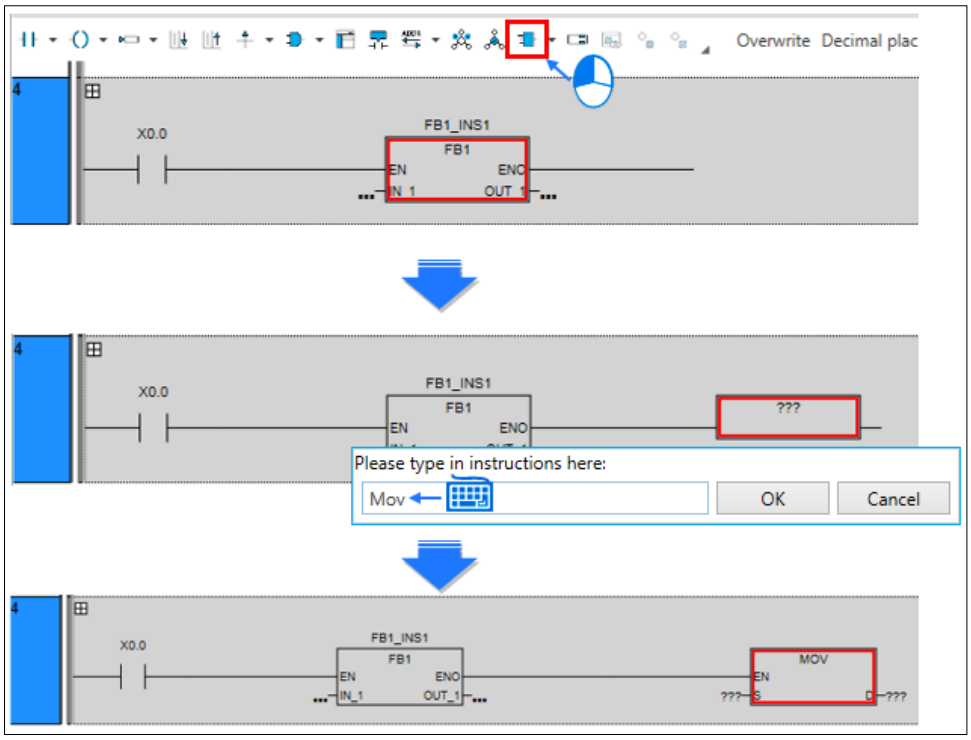


Figure 8 - 118: LD POU - insert Function Block from Toolbar

Method3:

Click the position in which an instruction or a function block is 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 typing of the applied instruction or the function block definition is complete, user can press **Enter** key on the keyboard or click **OK** at the right side of the box. (the applied instruction and the function block diagram are case-insensitive).

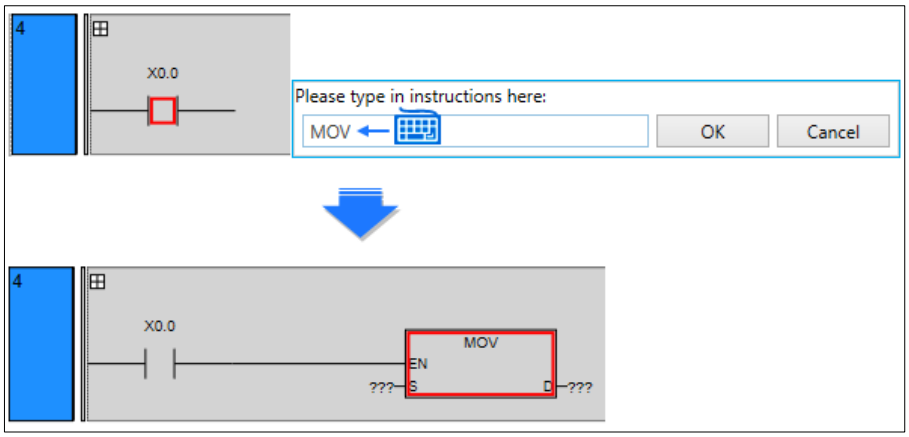


Figure 8 - 119: LD POU - type name and insert

If user want to insert a function block they have to type the function block definition and they cannot type the operands specified in the function block. If user want to insert an applied instruction, they have to type the applied instruction and they can type the operands specified in the applied instruction. User 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 variable which is not declared, the **Add a Variable to the POU** window will appear after the user press **Enter** key on the keyboard. Refer programming manuals for more information about instructions.

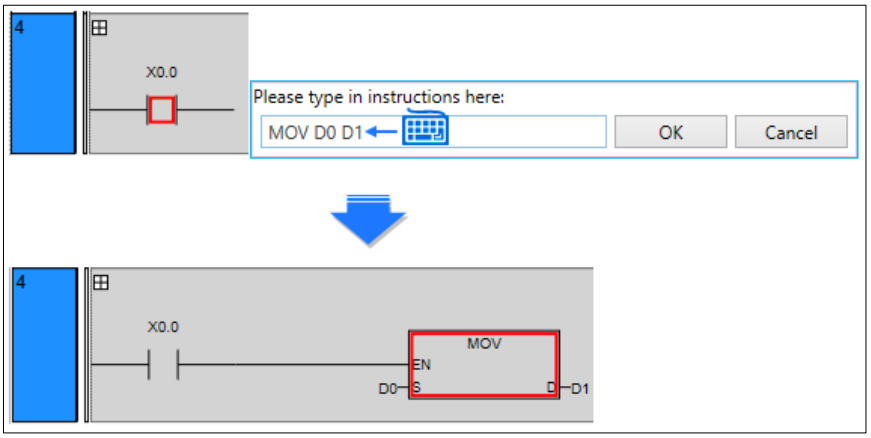


Figure 8 - 120: LD POU - Type instructions and insert

Creating Comparison Contact:

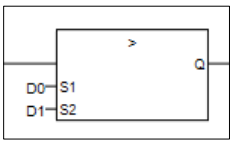


Figure 8 - 121: LD POU - Comparison Contact

An example of a Comparison Contact in DIADesigner is shown in the above figure. **>** in the block represents a comparison type. **S1** and **S2** are objects which are compared. The data format which is accepted depends on the comparison type. **Q** can drive a coil or connect to a contact in series.

The comparison contacts in a ladder diagram are similar to general contacts in that they can be connected to other contacts in series or parallel, as shown in the following figure.

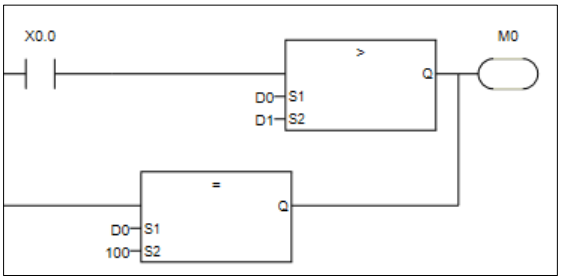


Figure 8 - 122: LD POU - examples of Comparison Contacts

A comparison contact can be inserted not only in one of the three ways described in previous section, but also by means of the following steps:

1. Click on a line or contact or an instruction.

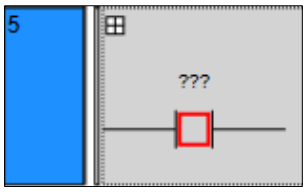


Figure 8 - 123: LD POU - select a Line or Contact

2. Click the icon on the toolbar and then select a type of comparator.

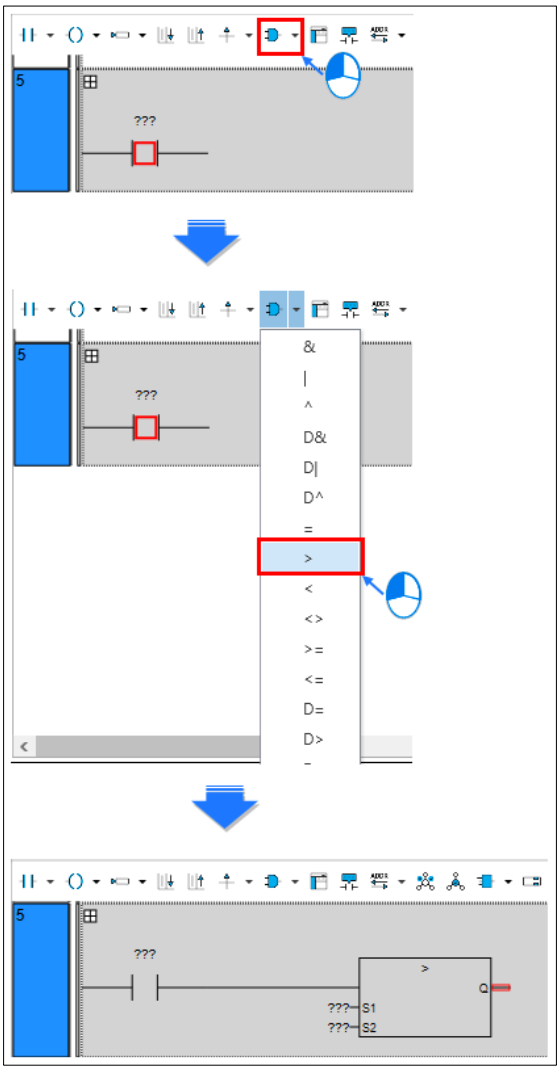


Figure 8 - 124: LD POU - insert Comparison Contact

Result: Comparison contact is inserted.

3. Type the Variables or Address for the S1 and S2 pins.

Inserting a Block Logic Instruction:

If user 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, user can combine two networks by means of using a block logic instruction.

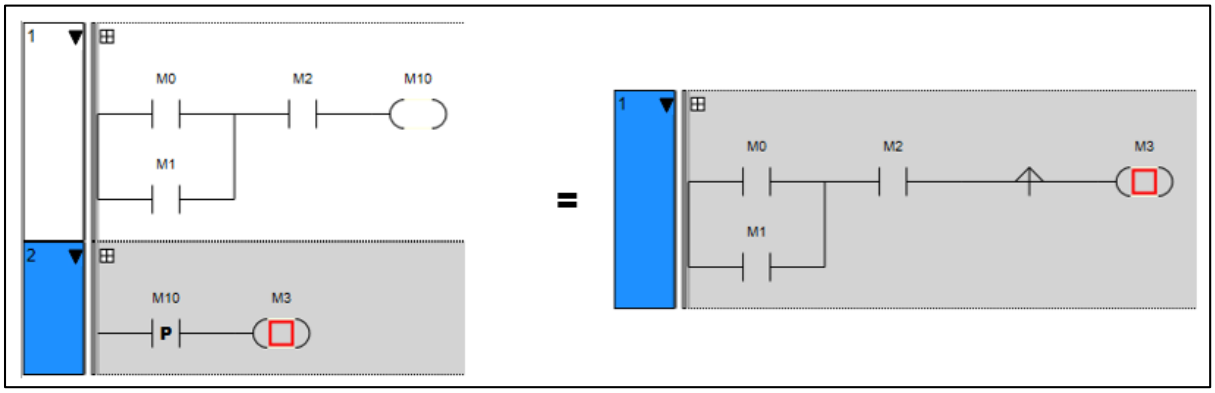


Figure 8 - 125: LD POU - Block Logic Instruction

The block logic instructions available in DIADesigner are described in the following table:

Representation	Instruction	Function
	NP	**Click to insert Positive Transition Trigger. When the block preceding this block logic instruction is tuned from OFF to ON, the output device is ON**
	PN	Click to insert Negative Transition Trigger. When the block preceding this block logic instruction is turned from ON to OFF, the output device is ON
	INV	Click to insert Not Transition Trigger. The operation result of the block preceding this block logic instruction is inverted and the inversion result is sent
	NP-	Click to insert a Parallel Trigger in parallel under the existing line or contact

Representation	Instruction	Function
	-NP	Click to insert a Parallel Trigger in parallel above the existing line or contact

Refer Programming Manuals for more information about FB_NP and FB_PN.

User can insert a block logic instruction by following methods:

- Clicking on respective icon in toolbar
- Typing the instruction

Clicking respective Block Logic Instruction from the toolbar:

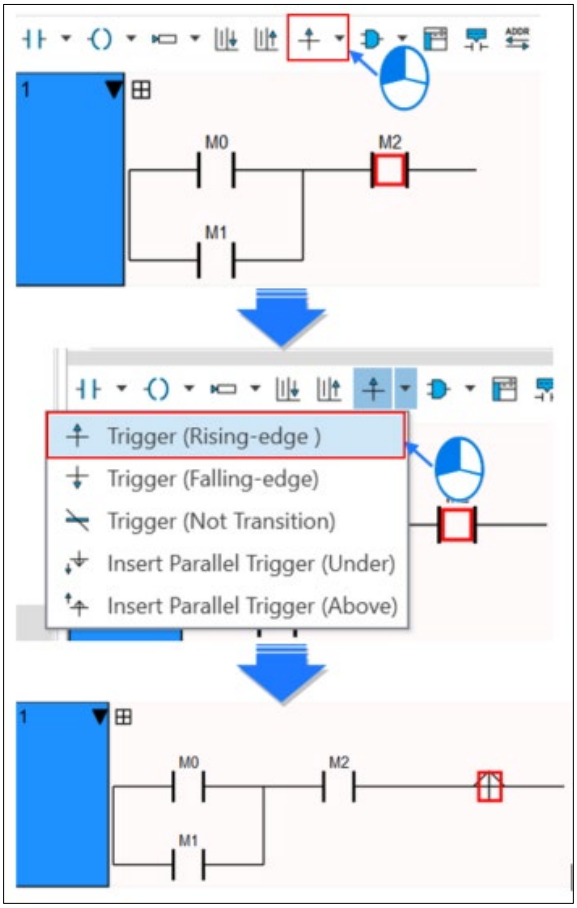


Figure 8 - 126: LD POU - inserting Block Logic Instruction from LD POU Toolbar

Typing Block Logic Instructions:

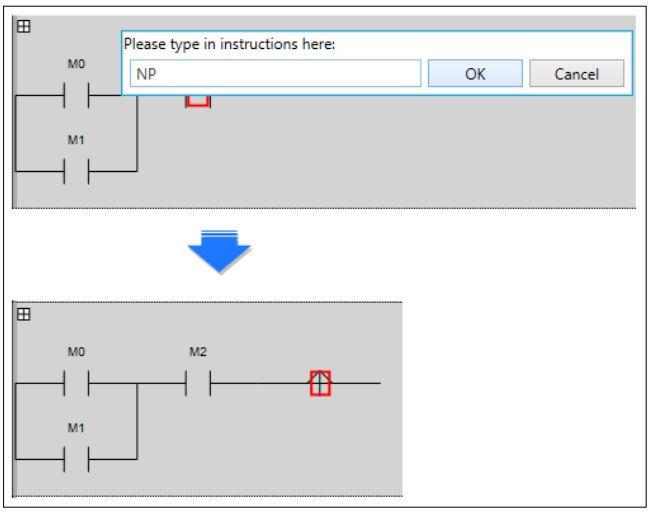


Figure 8 - 127: LD POU - type Block Logic Instruction

Creating Multiple Outputs:

If user want to create multiple outputs, they can insert a coil or an applied instruction first. After this, user insert multiple outputs from Toolbar or from the context menu of the output coil/applied instruction.

Clicking Multiple Output icon in Toolbar:

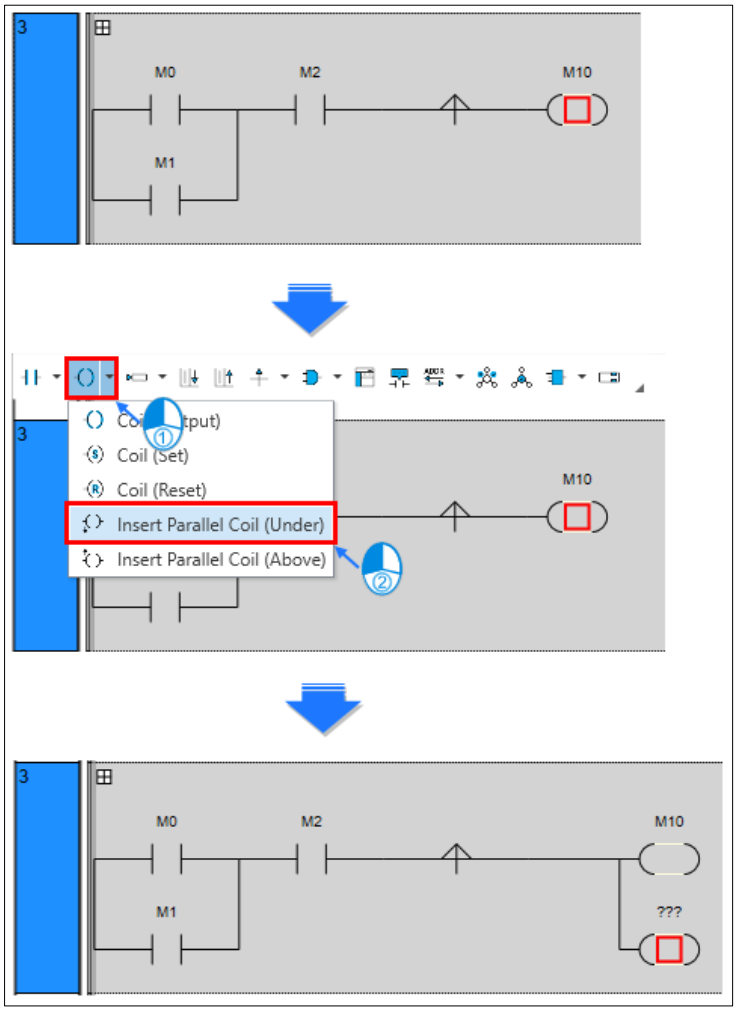


Figure 8 - 128: LD POU – insert Multiple Output from Toolbar

Selecting Multiple Output coils from context menu:

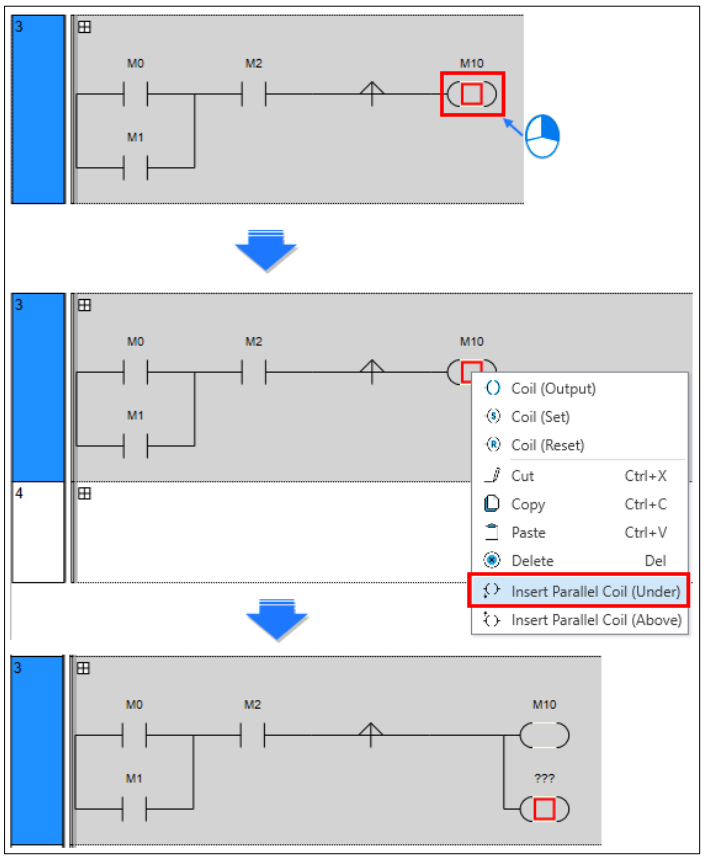


Figure 8 - 129: LD POU - insert Multiple Output from context menu

Putting a label:

If user 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 top side of the network.

Figure 8 - 130: LD Network - Label

Follow these steps to insert a Label in a LD Network:

1. Click the network at which a label to be added.
2. Click in toolbar or in the context menu.
3. Enter a Label name.
4. Press Enter key on the keyboard after a label is typed. The system will automatically put a colon at the right side of the label.

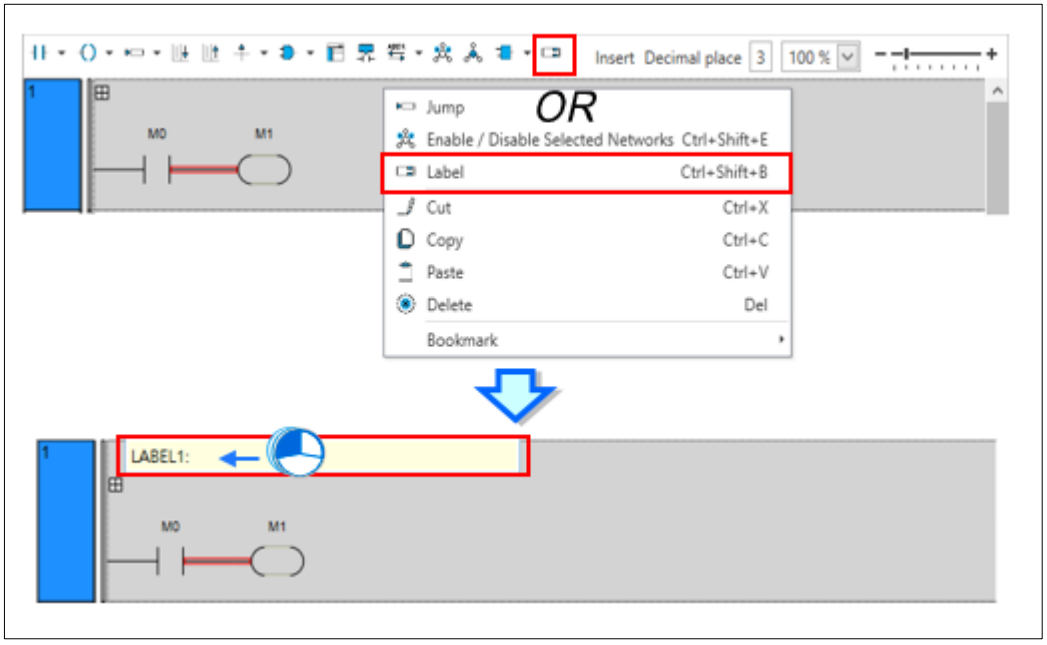


Figure 8 - 131: LD Network - insert Label

Editing Title and Comment:

In a Ladder Diagram in DIADesigner, user can not only make comments on Variables and Address, but add descriptive titles and comments on the Networks.

User can click on the icon above Networks to expand Title and Comments boxes where user can enter title and comments as shown in the following figure.

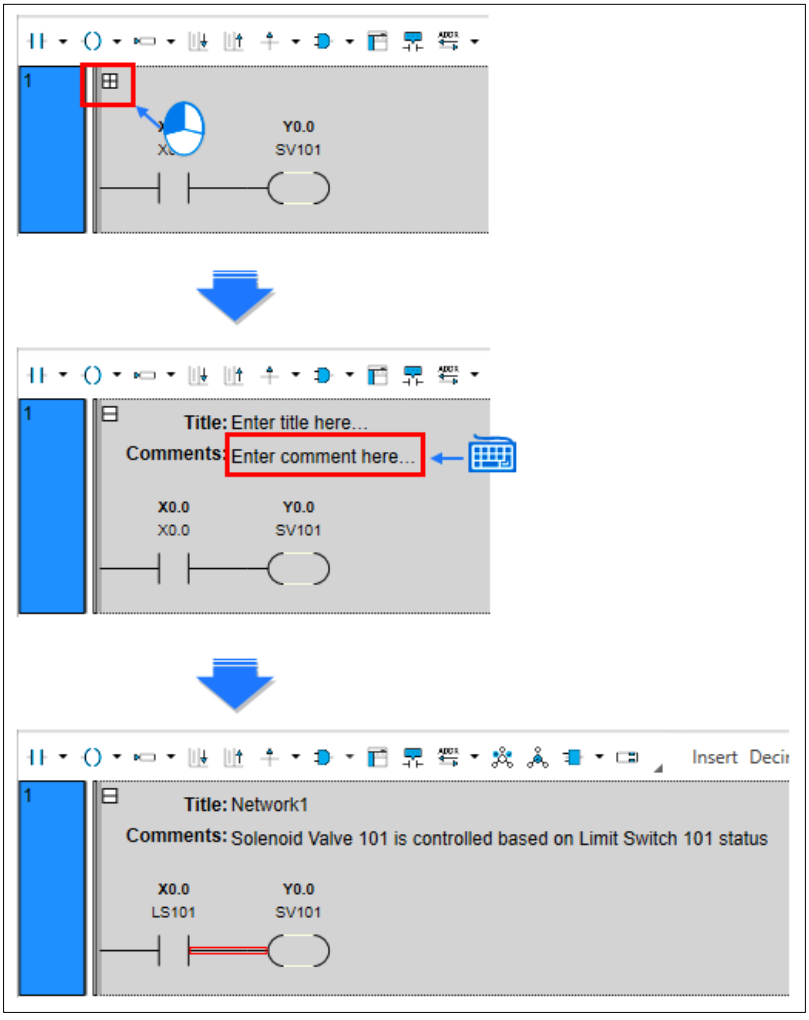


Figure 8 - 132: LD Network Comment

Show/Hide title and comments on a network:

After user click icon on the toolbar, all the titles and comments on the networks of the current POU will show/hide.

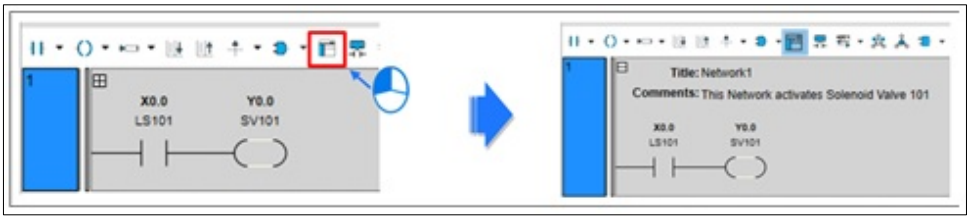


Figure 8 - 133: LD Network - Show Network Title and Comment

User can click on the same icon once again to hide them.

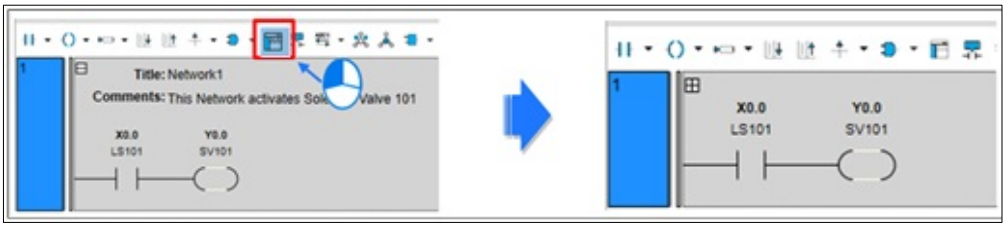


Figure 8 - 134: LD Network - Hide Network Title and Comment

Editing the comment on a Variable in LD element:

After user click icon on the toolbar, the comments on the Ladder elements display.

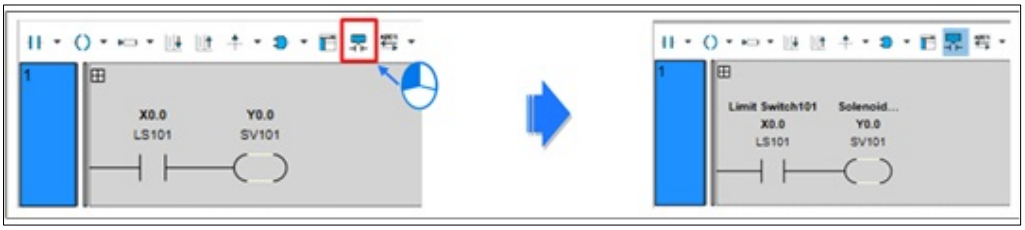


Figure 8 - 135: LD Network - Display Variable Comment

User can click on the same icon once again to hide them.

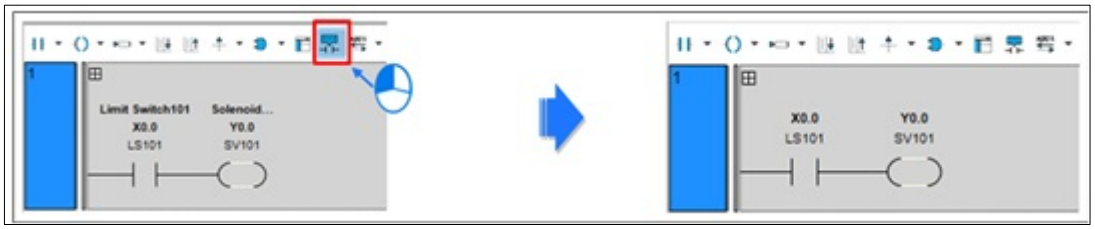


Figure 8 - 136: LD Network - Hide Variable Comment

After user click the position above a device name, they can type a comment in the box which appears. Press **Enter** key on the keyboard after the editing is complete.

Variable Mode and Address Mode:

If a LD element is assigned to a Variable declared in the Local Symbol Table in an LD POU, user can change the mode in which the Variable or Address is defined by means of clicking icon.

User can click the icon and select **Mode: Variable and Address** to display the variable and address associated with the LD elements.

Figure 8 - 137: LD Network - Variable and Address Mode

User can click the icon and select **Mode: Variable** to display the variable associated with the LD elements.

Figure 8 - 138: LD Network - Variable Mode

User can click the icon and select **Mode: Address** to display the address associated with the LD elements.

Figure 8 - 139: LD Network - Address Mode

8.3.2 SFC - Sequential Function Chart

A Sequential Function Chart (SFC) 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 is shown in the following figure. The main components of a sequential function chart are the Steps with associated Actions, Transitions and associated Logic conditions and directed links between Steps and Transitions.

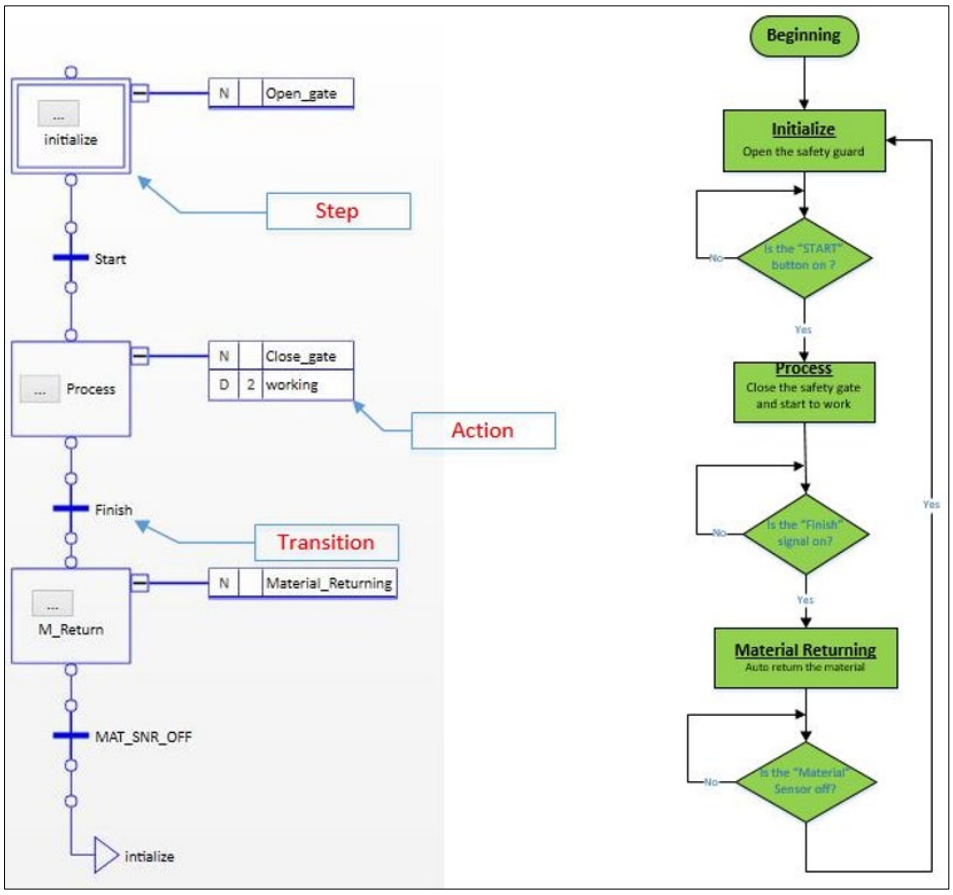


Figure 8 - 140: Sequential Function Chart

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 stop and allows 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 following 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 DIADesigner, the Steps must alternate with Transitions. If there are two continuous Steps or two continuous Transitions in a Sequential Function Chart, the program cannot be compiled.

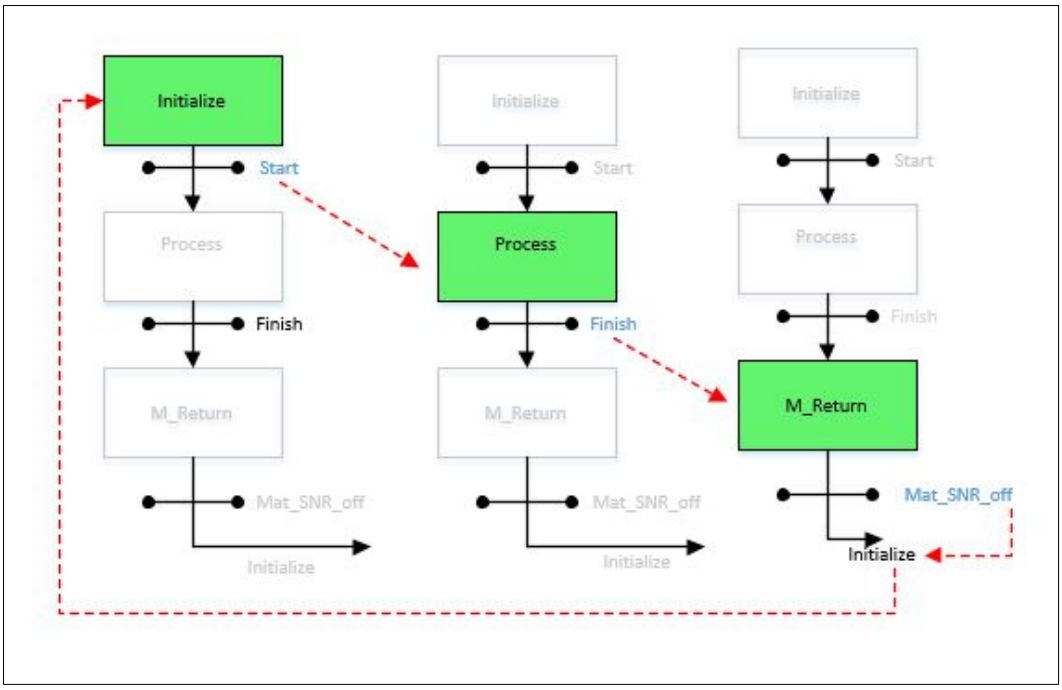


Figure 8 - 141: Work flow of a Sequential Flow Chart

The above example is the structure of a Sequential Function Chart. DIADesigner 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 are introduced in the sections below.

8.3.2.1 Add SFC program

User can select **SFC** in the **Language** field in **Add Program** window while creating a POU of program type. Refer *8.2.1 POU* for detailed information.

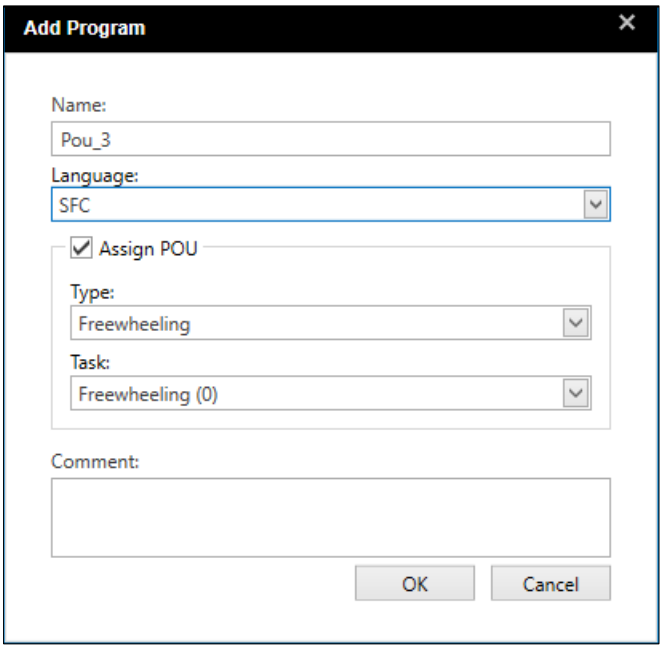


Figure 8 - 142: Add SFC POU

The environment in which a SFC can be edited is shown in the following table. In any POU window (program type or function block type), there are two areas:

- Local Variable table area
- POU Working area

The table in the upper part of the SFC POU window is a Local Variable Table and the lower part is POU working area. After a program editing window to create a SFC POU is opened, the corresponding tool bar will appear in the DIADesigner window.

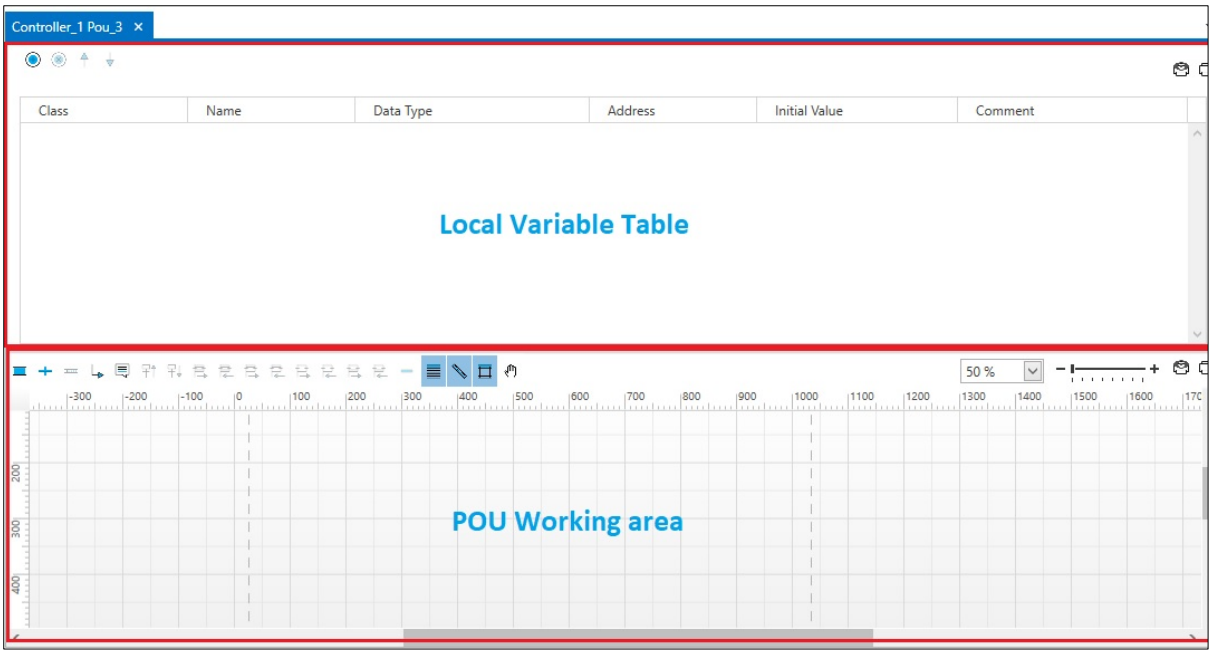


Figure 8 - 143: SFC POU - Local Variable Table and POU working area

SFC Tool bar:

After the POU window is opened, the toolbar for editing the Sequential Function Chart displays as shown in the following figure.

Figure 8 - 144: SFC POU Toolbar

The functions in SFC POU window are described in the following table:

Function/Icon	Description
	Click to add a Step
	Click to add a Transition
	Click to add Action

Function/Icon	Description
	Click to collapse Action List
	Click to insert a Jump
	Click to add Comment
	Click to insert a Step and a Transition above
	Click to insert a Step and a Transition below
	Click to add a simultaneous Step to the right direction
	Click to add a simultaneous Step to the left direction
	Click to add a Step and a Transition to the right direction
	Click to add a Step and a Transition to the left direction
	Click to add a Transition to the right direction
	Click to add a Transition to the left direction
	Click to add a Divergence of Sequence Selection & Simultaneous Convergence to the right direction
	Click to add a Divergence of Sequence Selection & Simultaneous Convergence to the left direction
	Click to display/hide Grid
	Click to display/hide Ruler
	Click to display/hide Page Range
	Click to switch to Hand mode

8.3.2.2 SFC Elements

Sequential Function Chart in DIADesigner

Steps and Actions:

Every Step in DIADesigner must be assigned a variable whose datatype is **STEP**, and functions as a state flag. If the flag corresponding to a Step is ON, the Step is active. Every variable in AS series PLC whose datatype is STEP occupies an **S**

register in the PLC. Refer the operation manuals for more information about the S registers.

After a Step is activated, the actions associated with the Step are executed according to the qualifiers. In DIADesigner, 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.

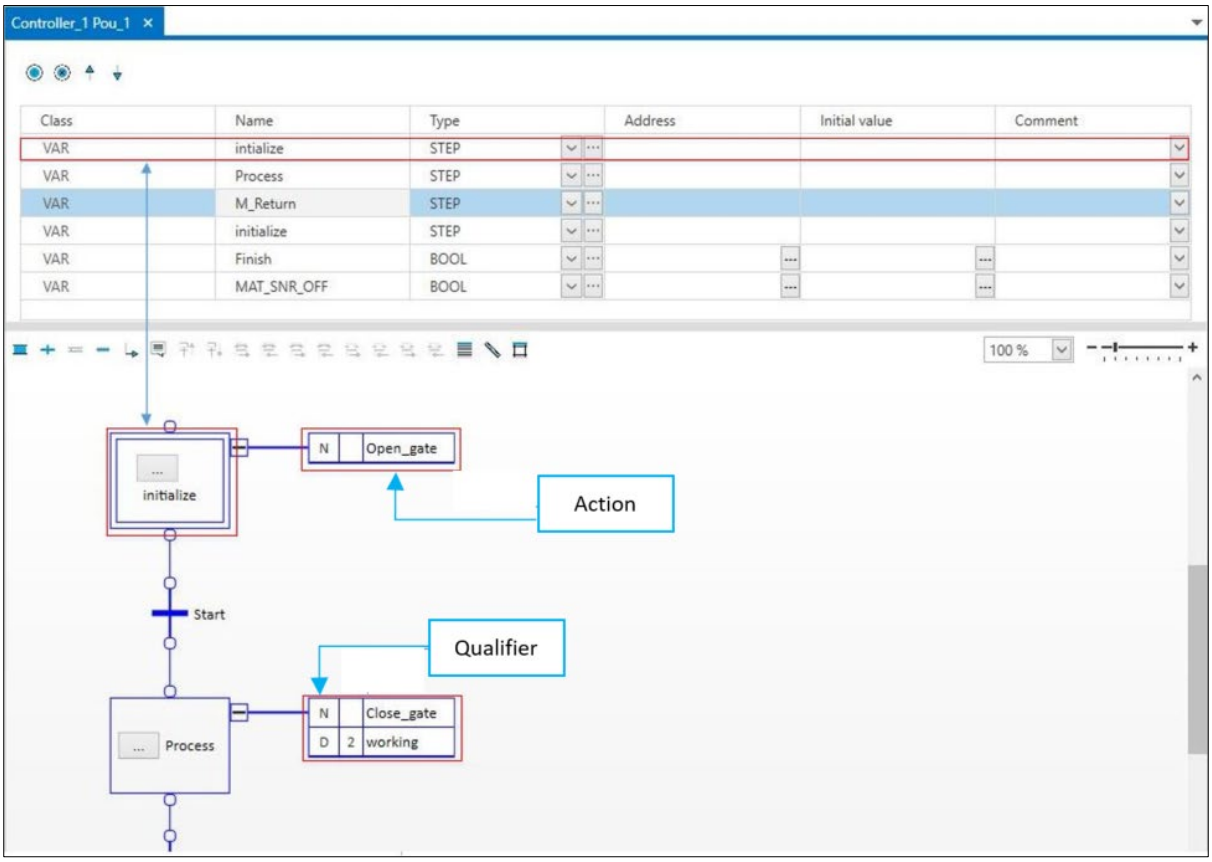


Figure 8 - 145: SFC POU – example

In DIADesigner, an action is a program code, and can be developed by means of a ladder diagram, a structured text, etc. The actions which are created are listed in the Project Explorer. The POU window in which an action can be created differs from the window for a programming language in that there is no local variable table in the POU window in which an action can be created. In DIADesigner, the actions and the transitions in a sequential function chart share the same local variable table.

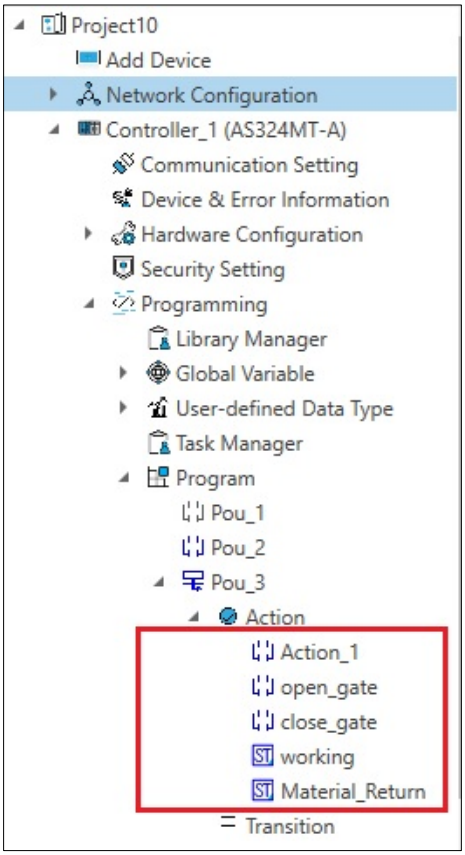


Figure 8 - 146: SFC POU in Project Explorer

If user want to edit a sequential function chart, they can firstly create actions, then assign the actions to steps and assign qualifiers to the actions. To perform this, right-click on the step to be set, then click **Action Setting** in the context menu. In the **Action Settings** window, user can add actions and set qualifiers as shown in the following figure.

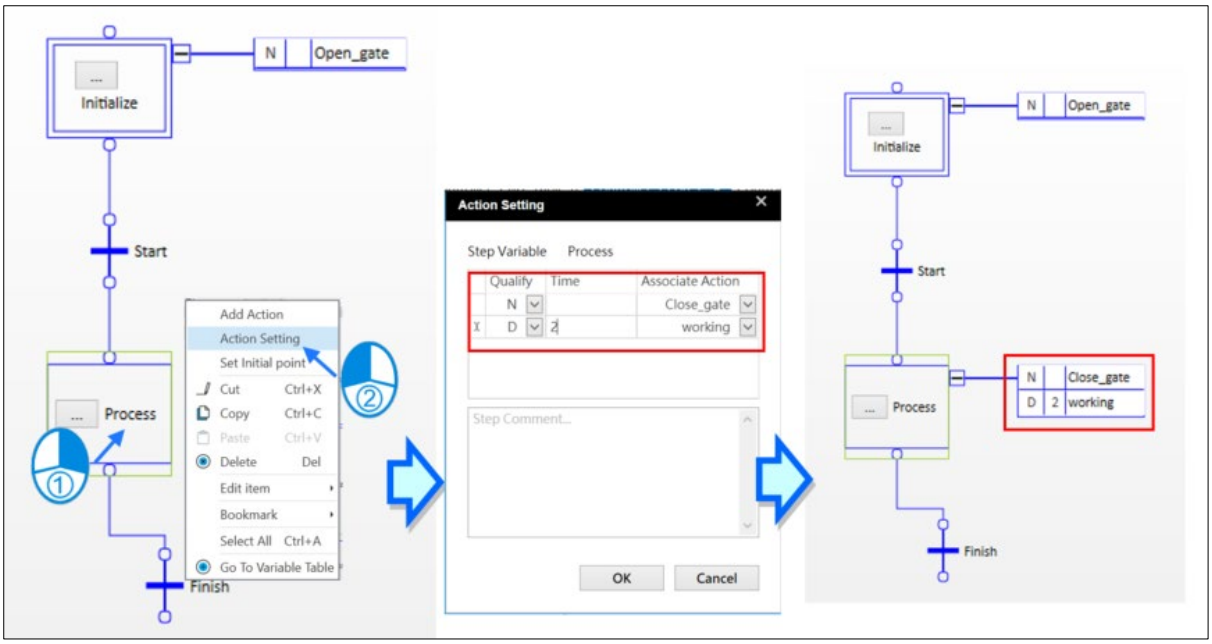


Figure 8 - 147: SFC POU - Action Setting window

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 an action. If a final scan is executed, the coils driven by the instruction OUT is OFF, the application instructions and the function blocks will not be executed.The Timers are 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 contact of the counters will remain unchanged, and the values in the counters will remain unchanged.

Action0 in the following figure 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 the final scan.

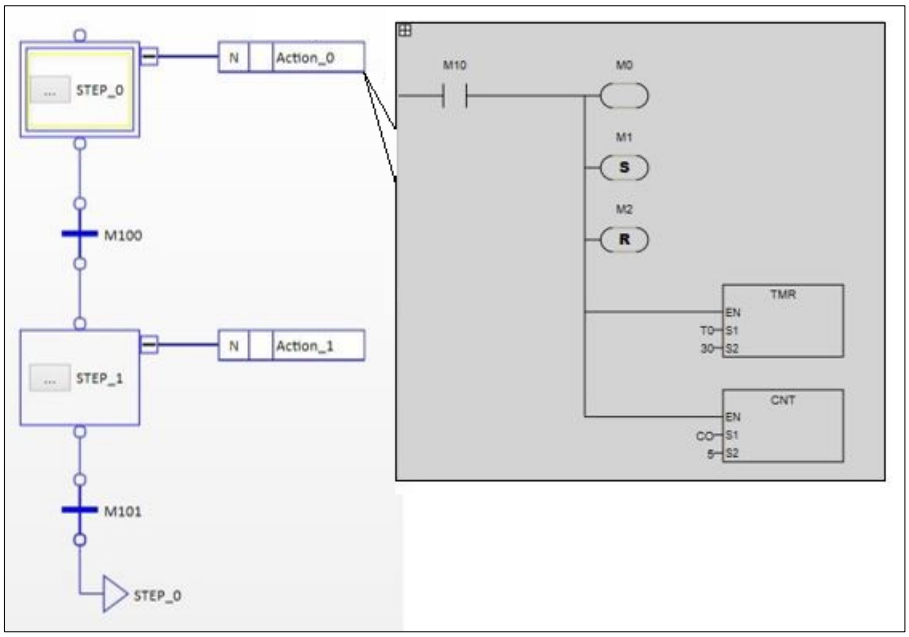


Figure 8 - 148: SFC Action - created with LD

STEP_0 in the following figure 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.

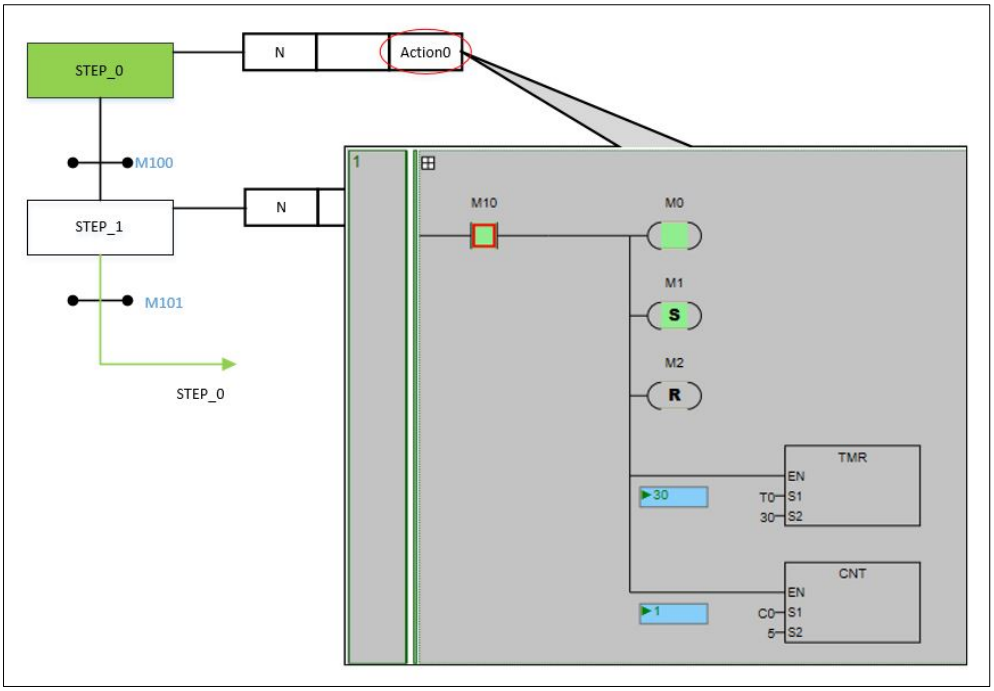


Figure 8 - 149: SFC Action - Online

STEP_1 in the following figure 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.

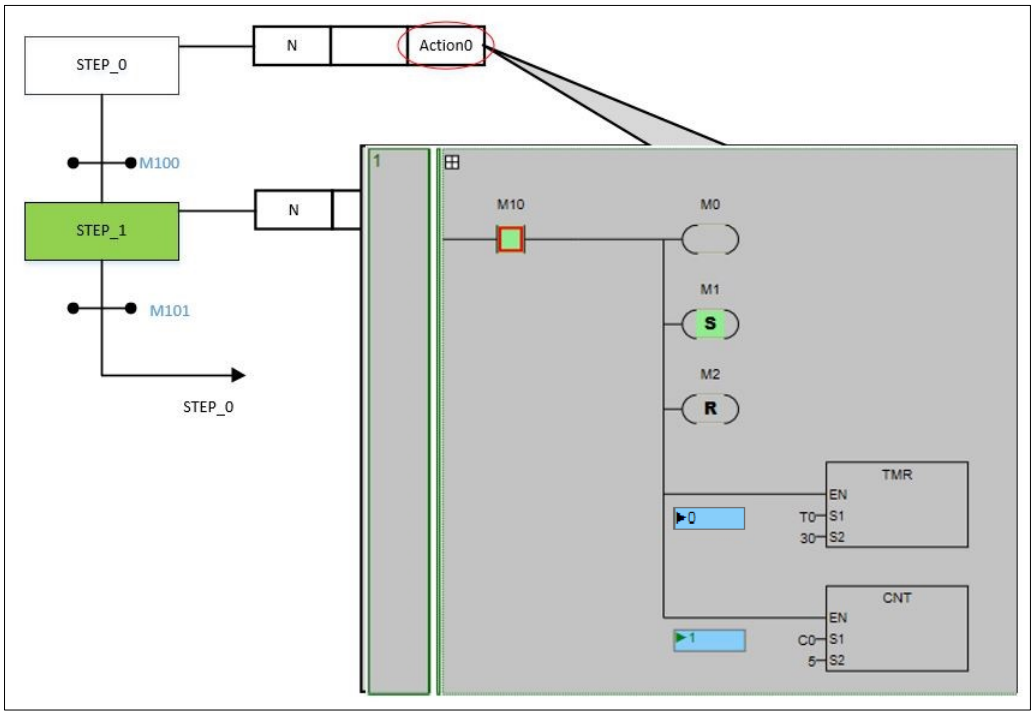


Figure 8 - 150: SFC Action - Active Step – Online

For AS series PLC, user can set the Final Scan to run or not for every action. The default is to run the Final Scan in the DIADesigner. To disable this function, user can right-click the action that does not require a Final Scan to open its context menu and then uncheck the **Action/Transition** > **Reset**. After that user may notice that the action that does not required a Final Scan is changed from blue color icon to gray color icon. 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 is cleared.

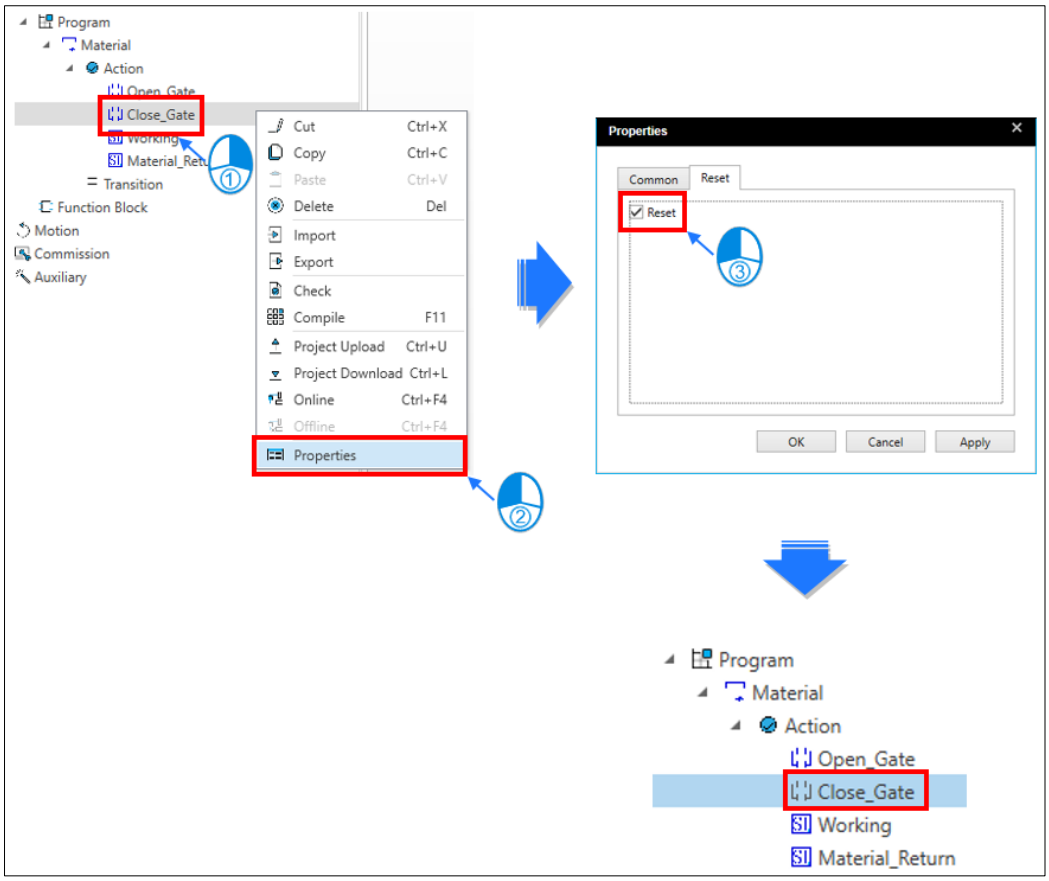


Figure 8 - 151: SFC Action - Reset

Transitions:

When a Transition is true, the active step immediately before this transition becomes inactive and the step immediately after this 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 DIADesigner, a Transition can be a Boolean register or variable or a program code. If a transition is a logical operation, user can create a transition program, and assign the program to the transition. During operation, click on a transition to be set, then (1) use keyboard to enter Transition or (2) select Transition/ Boolean register in the drop-down list as shown in the following figure.

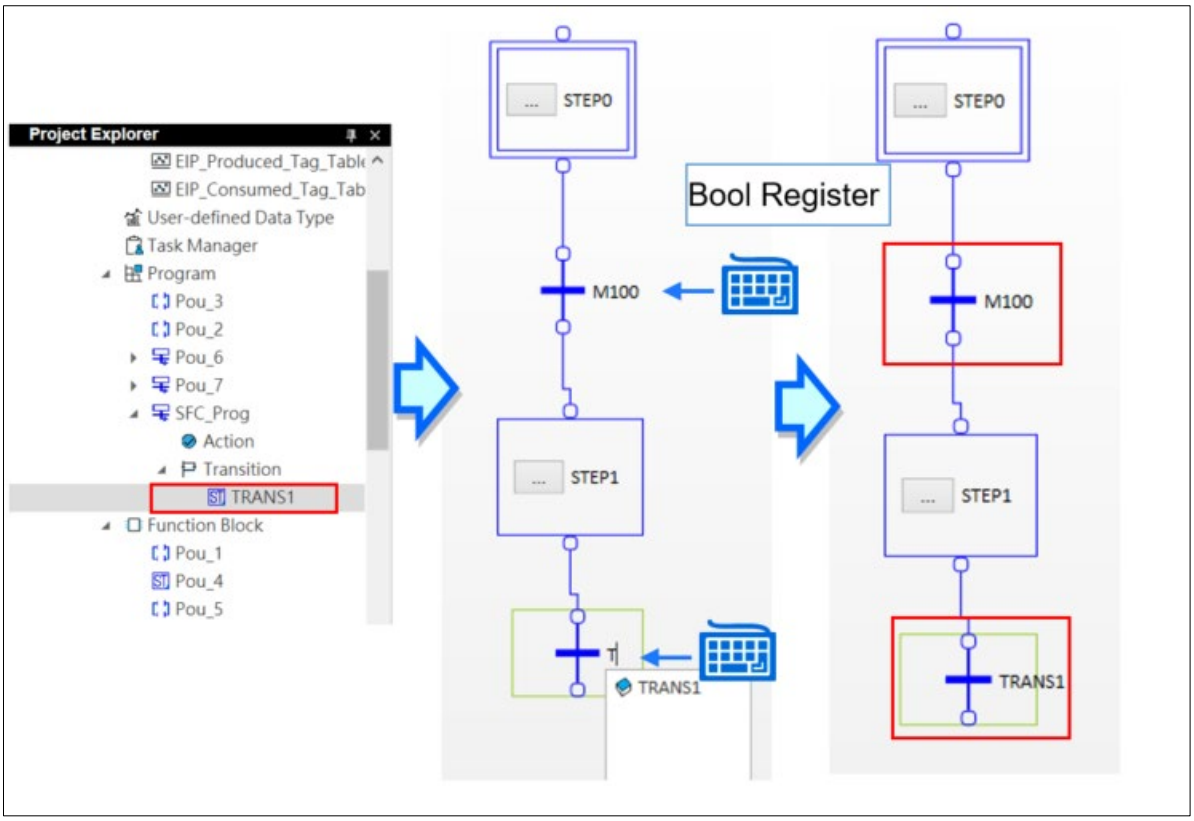


Figure 8 - 152: Transition: Boolean Register or Variable

The programming languages which currently can be used to create transition programs include ladder diagrams (LD) and structured texts (ST). The Boolean state in a transition program must be sent to a variable name which is the same as the transition program name, whether the programming language used to create the transition program is a ladder diagram or a structured text. The variable need not be declared. However, if there is a variable name in a variable table which is the same as a transition program name, an error occurs when the program is compiled. Besides, applied instructions and function blocks cannot 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 variable 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.

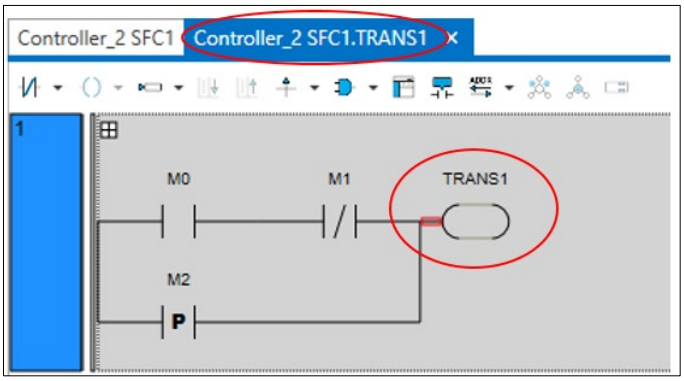


Figure 8 - 153: SFC - Transition in LD

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 variable name which is the same as the transition program name.

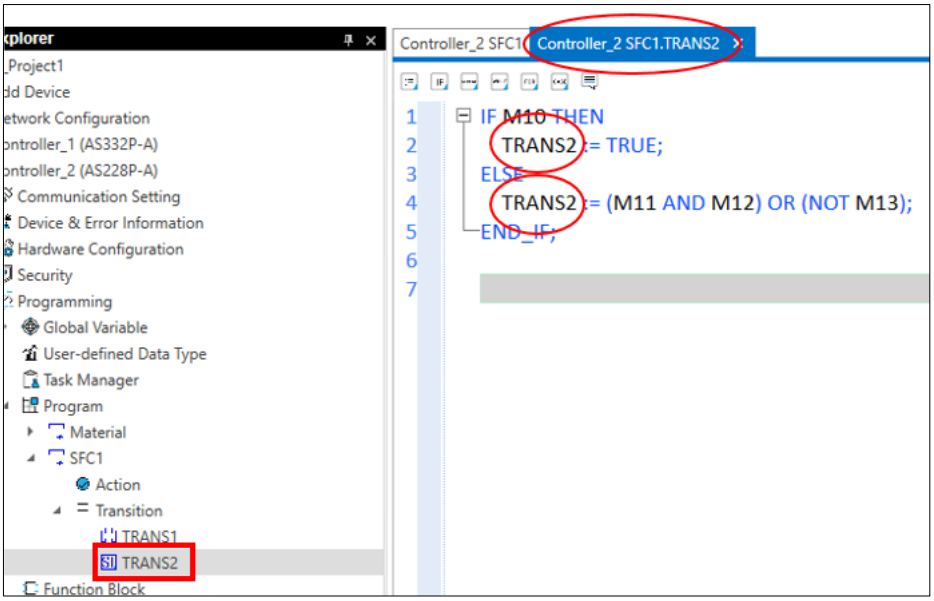


Figure 8 - 154: SFC - Transition in ST

When user create a transition program, a variable name which is the same as the transition program name is added to the program code of the transition program. The user 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 variable table, there are not local variable tables in the program editing windows for the transitions.

User cannot create a complex transition program. If a transition condition must undergo a complex operation, user 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 user do not need to declare the variables in the transition programs, the system assigns memories which function as flags to the variables. As a result, user have to make sure that the variables 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, TRANS4 is ON if the result of (M0 AND M1) is ON. Since TRANS4 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 is executed again. If the result of (M0 AND M1) is OFF, the statement following THEN will not be executed. TRANS4 will still be ON because it is not updated. The system will evaluate the transition condition as TRUE, and the control is passed to the next step.

Adding an **ELSE** section to the transition program in the following figure is a better way to ensure that the variable TRANS4 is assigned a specific logic state. User also have to pay attention to the same problem in a CASE structure.

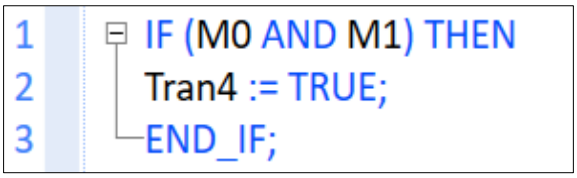


Figure 8 - 155: SFC - Transition in ST with IF or CASE

Simultaneous Divergence and Divergence of Sequence Selection:

User can create divergent paths in a sequential function chart. There are two types of divergences:

- Simultaneous divergence
- Divergence of sequence selection

Simultaneous divergence:

A Simultaneous divergence is shown in the following figure. The divergence steps are connected by a double line. The characteristics of a simultaneous divergence is that before entering the divergent steps, there is a shared transition above. When the transition is true, the steps are active.

When Trans0 is true, STEP_0 becomes inactive and STEP_1 and STEP_2 becomes active.

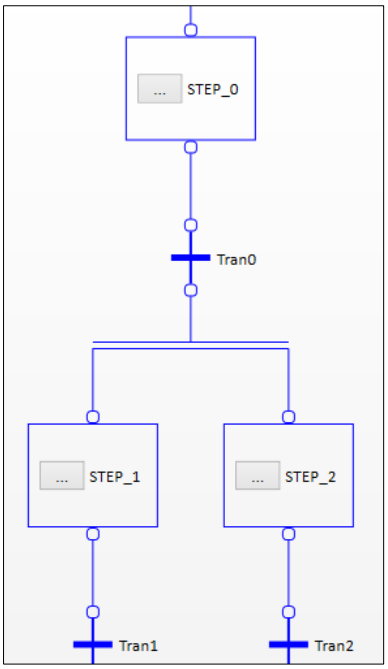


Figure 8 - 156: Simultaneous Divergence

Divergence of Sequence Selection:

A Divergence of Sequence Selection is shown in the following figure. The characteristics of divergence of sequence selection is that the divergent steps have their own transitions above. 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 left-most transition is activated.

When the Tran2 is true, STEP_0 becomes inactive and STEP_2 becomes active. Trans1 is not evaluated and STEP_1 remains inactive.

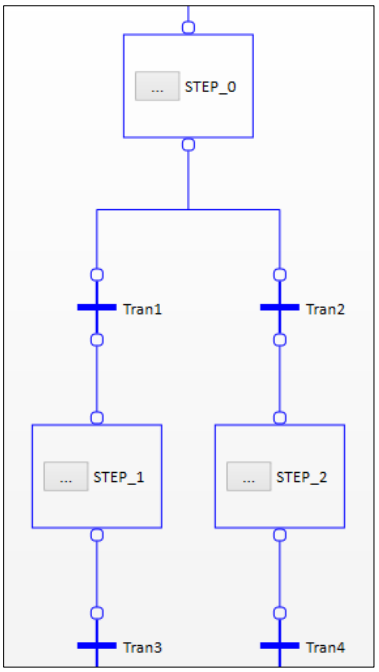


Figure 8 - 157: Divergence of Sequence Selection

Simultaneous Convergence and convergence of Sequence Selection:

The divergent paths in a SFC must be converged. There are two types of convergences:

- Simultaneous Convergence
- Convergence of Sequence Selection

Simultaneous Convergence:

A Simultaneous Convergence is shown in the following figure. The convergent steps are connected by a double line. The characteristics of a simultaneous convergence are that the convergent steps are connected to the same transition below. The transition will not be evaluated until the convergent steps preceding the transition are activated.

After STEP_1 and STEP_2 are activated, Tran3 is evaluated. When Tran3 is true, STEP_1 and STEP_2 becomes inactive and STEP_3 becomes active.

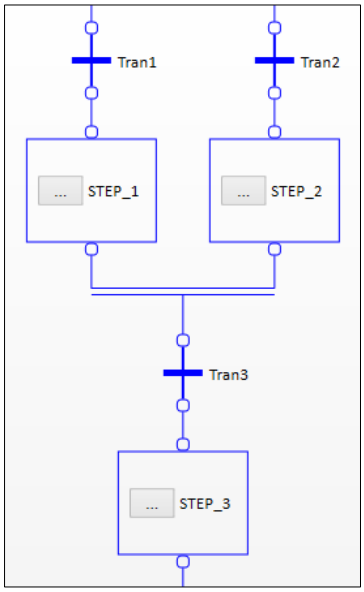


Figure 8 - 158: Simultaneous Convergence

Convergence of Sequence Selection:

A convergence of sequence selection is shown in the following figure. The characteristics of a convergence of sequence selection is that the convergent steps have their own transitions. When one transition is true, the next step become active. If there is another convergent step which is activated, the step is active until the transition following the step is true. When the transition is true, the step following the transitions becomes active. Consequently, if a convergence of sequence selection is used in a SFC, several steps following the convergence may be activated. If user use a convergence of sequence selection in a SFC, 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 is inactivated and STEP_3 is activated again.

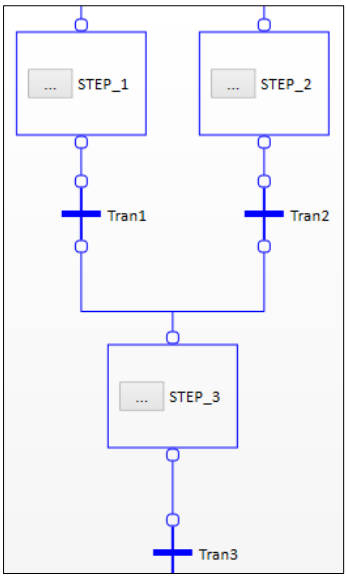


Figure 8 - 159: Convergence of Sequence Selection

Jump:

The use of a jump structure in a SFC 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 SFC. User can use jump structures quite freely. If they use a jump structure in a SFC, they have to make sure that the program is not confused.

The program in the following figure is a SFC where there is a divergence of sequence selection. If Tran2 is true, control is passed to STEP2. When Tran4 is true, it passes control to STEP1 through a jump. When Tran5 is true, it passes control to STEP0.

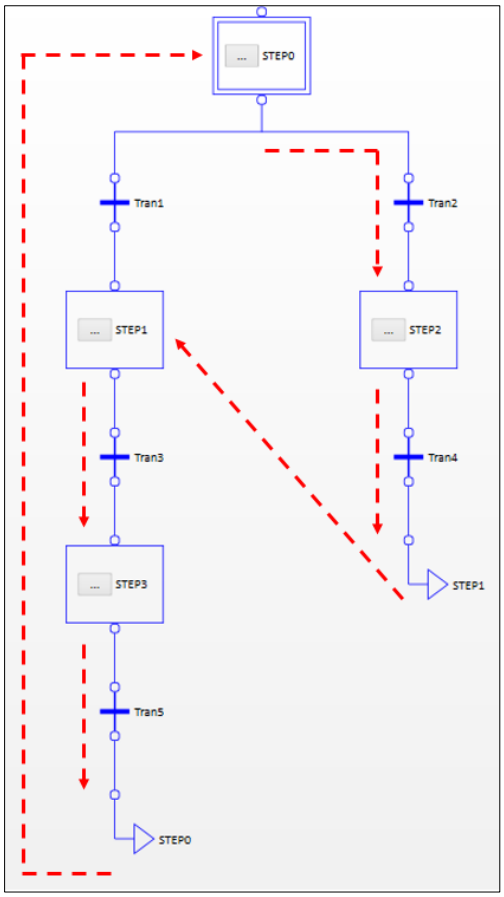


Figure 8 - 160: SFC with Jump

If user use a jump structure in a SFC, the step specified must be a step in the same SFC. If the step specified is not in the same SFC and is declared in a local variable table or the global variable table, an error occurs when the program is compiled.

If control needs to be passed from a step in a SFC to a step in another SFC, user can make use of the characteristics of steps and the characteristics of transitions. User can declare the step or the transition which is the destination jump in the global variable 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 following figure shows that SFC_A is executed first. SFC_B is executed after SFC_A is executed and SFC_A is executed after SFC_B is executed. User can achieve the jumps by means of the characteristics of transitions.

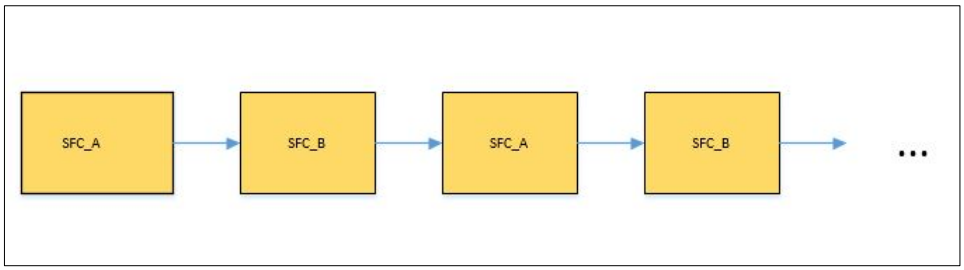


Figure 8 - 161: SFC Step Sequence Execution

The steps SP_A and SP_B in the following example are used to express the sequential function chart which is executed. They are preceded and followed by steps. Besides, the transitions RUN_A and RUN_B are declared in the global variable 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 AS series CPU module, user can assign the qualifier P or P1 to the actions associated these steps.

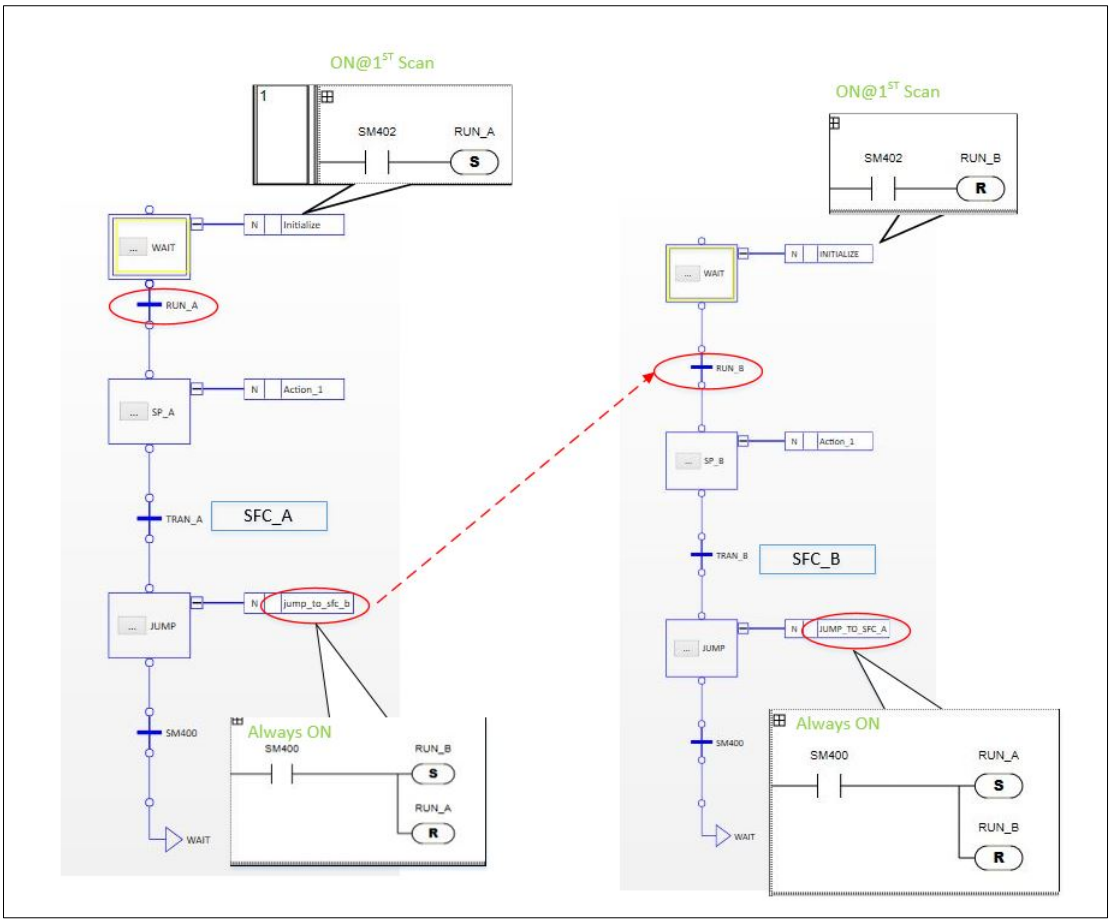


Figure 8 - 162: Jump between two SFCs

Set Qualifier of Action:

In DIADesigner, user can define the execution of an action by qualifying the action.

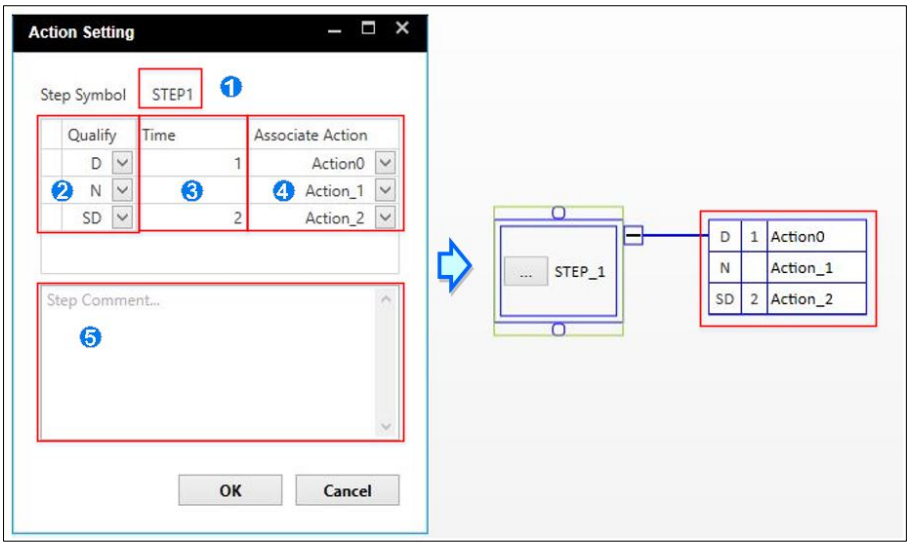


Figure 8 - 163: Qualifier of an Action

1. The step name displays here
2. User can select qualifiers for the actions associated with the step
3. User can select qualifiers related to time
4. User can select actions which are qualified
5. User can make comments on the step

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.

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 following figure 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.

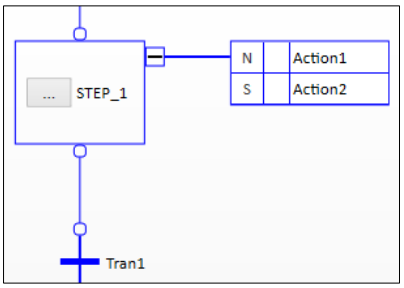


Figure 8 - 164: Set Action

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 is inactivated, control is passed to the next step, and the D action is not executed. When the step is activated again, the delay is measured again.

Action1 in the following figure 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.

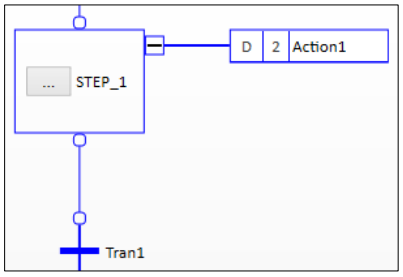


Figure 8 - 165: Delay Action

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 is 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 are 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.

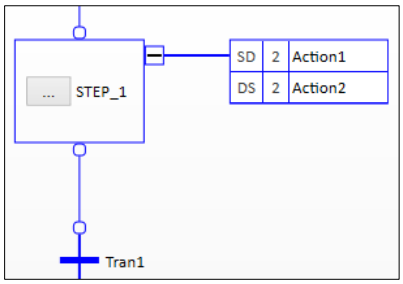


Figure 8 - 166: Delay Set Action

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.

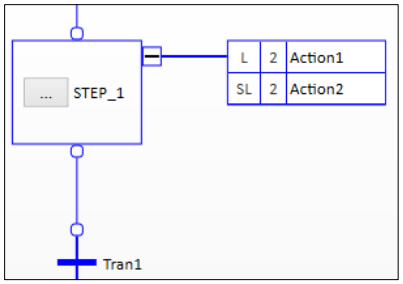


Figure 8 - 167: Set Limit Action

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 is 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.

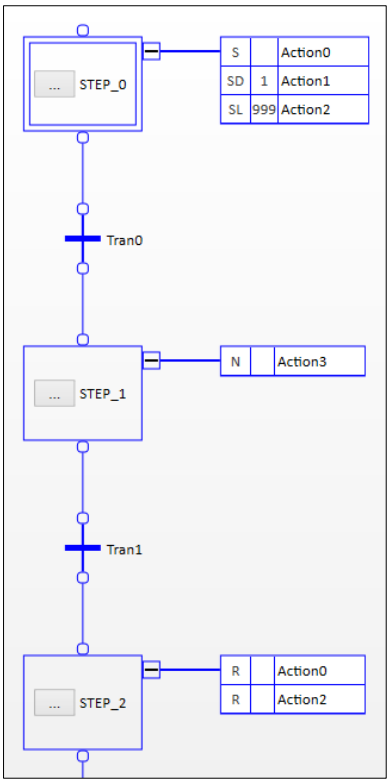


Figure 8 - 168: Reset Action

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 are 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.

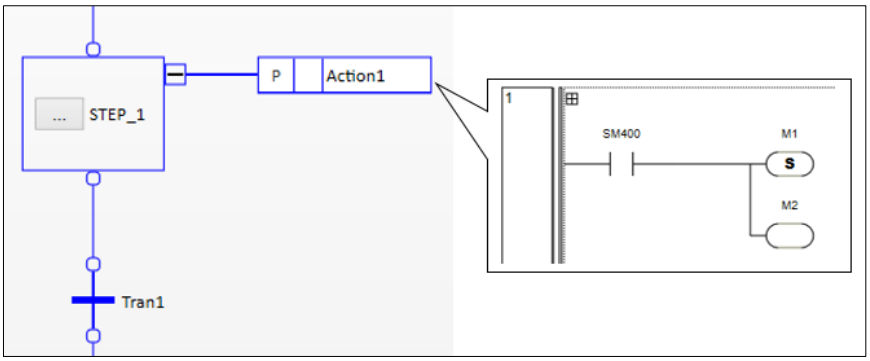


Figure 8 - 169: Pulse Action

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 is 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.

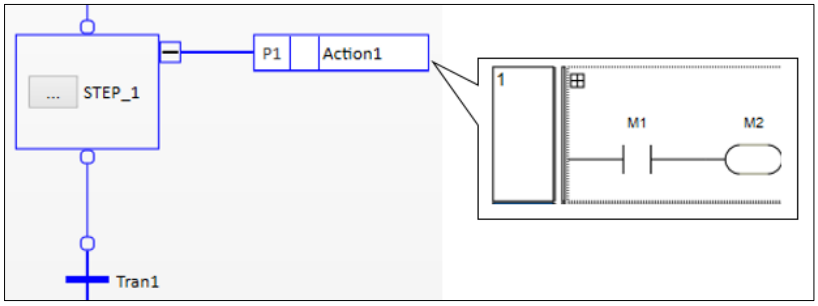


Figure 8 - 170: Raising Pulse Action

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 is 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.

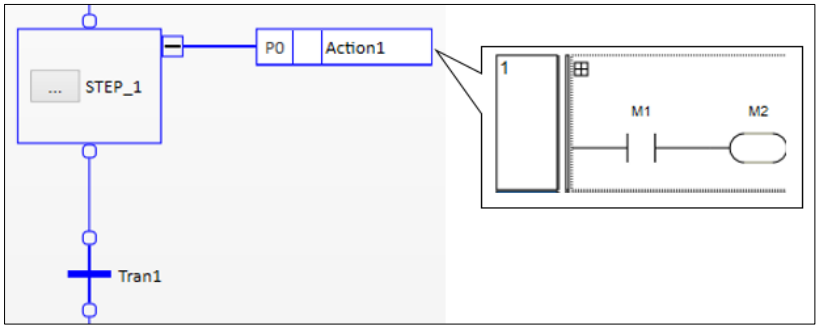


Figure 8 - 171: Falling Pulse Action

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, user 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 is executed all the time.

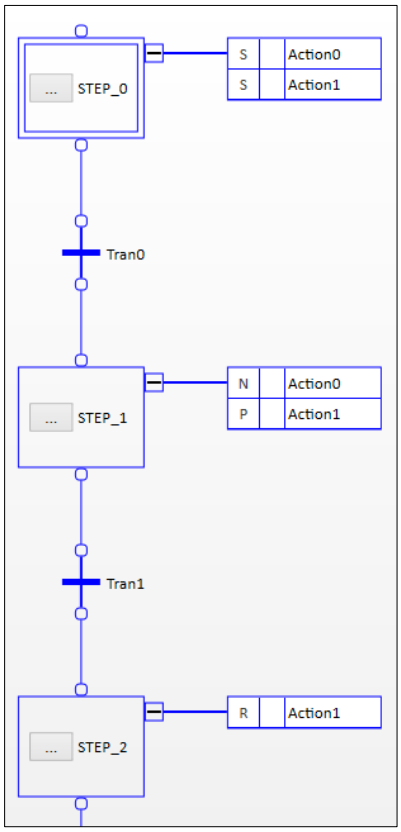


Figure 8 - 172: Qualifying an Action - Example

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.

In the following example, 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 Lin the STEP_0 and R in the STEP_1 is correct, since R can be added to an Action with more than 1 qualifier.

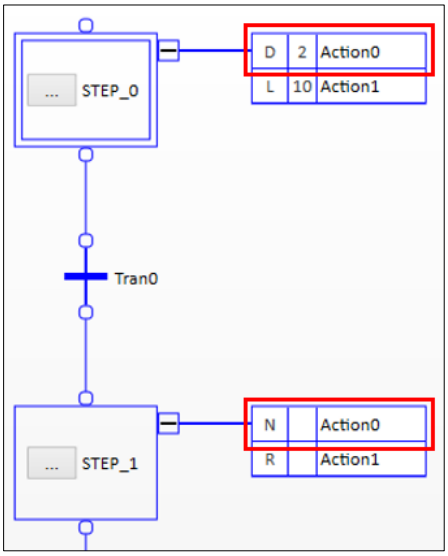


Figure 8 - 173: Qualifying an Action - Example 2

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 DIADesigner, 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.

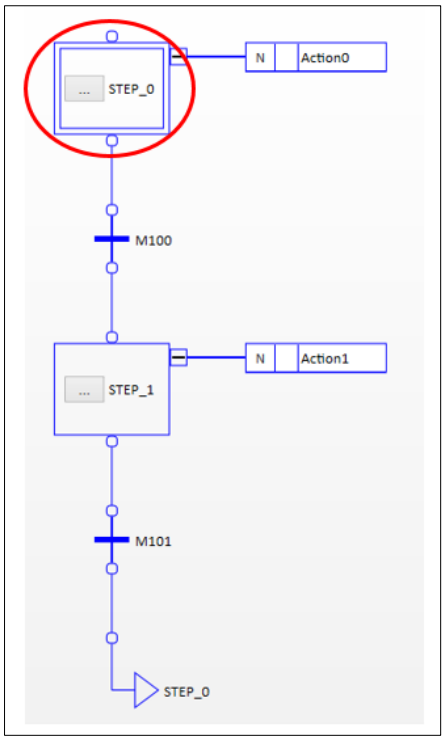


Figure 8 - 174: Initial Step

Any step in a sequential function chart in an AS series CPU module can be defined as the initial step in the sequential function chart. When a sequential function chart in an 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.

Right-click on a step that user wants it to be as the initial step and select **Set Initial Point** to complete the setting as shown in the following figure.

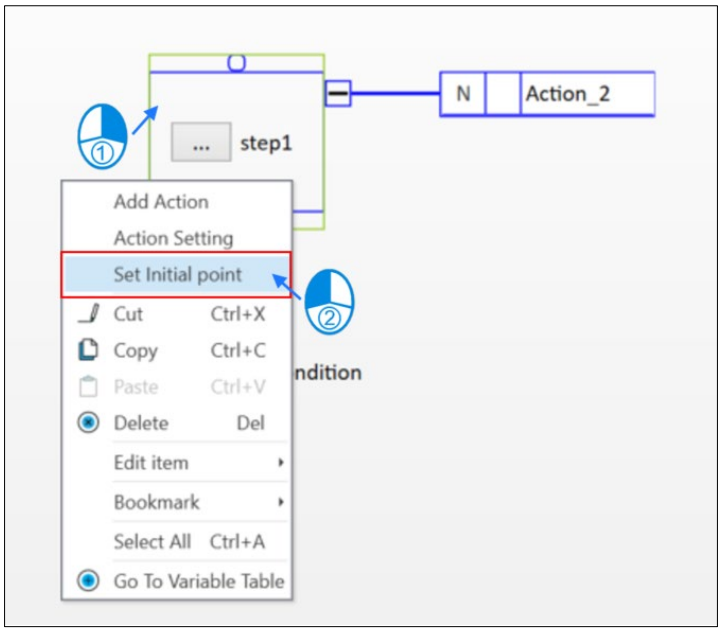


Figure 8 - 175: Set Initial Step

8.3.2.3 SFC Internal Properties

To make it easier for user to use certain property parameters of SFC while programming, internal properties are available for user to call among POUs. The property parameters can be divided into three categories: Step, Action, and Transition, which will be introduced in the following sub-sections.

Step Property Parameter

If user need to call Step property parameters in the POU of SFC. Property parameter is presented in the form of the structure as shown in the following figure. The structure is composed of “POU name of SFC where Step is at”, “Step name”, and “Step member”. User can select appropriate POU and Step names first, and then decide the needed Step member.

Figure 8 - 176: Property Parameter

Type	Member	Description	Read/Write
Step	X	When Step is running, the SFC property parameter will be ON, and this SFC property parameter usage will replace the past S register.	**Read only**
	done	When **T** property of Step (currently, Step has executed time) >= **timePreset** property of Step, the value will be ON.	Read only
	counter	When **X** property of Step changes from OFF to ON, the property parameters count adds one.	Read only
	timePreset	Determine the length of time that **done** property of Step changes from OFF to ON. (Unit: **ms**; data length: **DWORD**)	Read only
	maxTime	Record **T** property of Step, the Max. time that has ever been timed. (Unit: **ms**; data length: **DWORD**)	Read only
	T	How long has Step been executing. (Unit: **ms**; data length: **DWORD**)	Read only

Take the member X of Step in SFC as example: If user want to start execution at Step 1 in POU SFC_Prog of SFC to trigger a normally open contact in POU of LD program as ON, the program description will be as the following. Use a normally open contact in POU of the LD program, click above the contact to open the drop-down list, select the POU name of SFC program, Step name, and the name of member property parameter, and finish the selection.

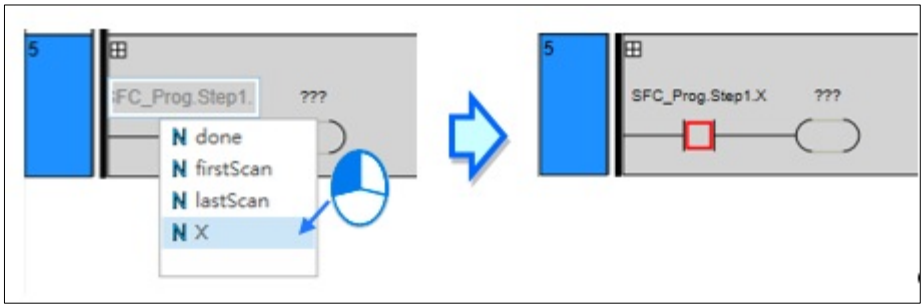


Figure 8 - 177: Use Property Parameter

After that, when Step1 in POU of SFC program is running, the normally open contact, SFC_Prog.Step1.X, of the LD program will be ON.

Action and Transition Property Parameters

If user need to call Action or Transition property parameter in the POU of SFC. Property parameters are presented in the form of the structure as shown in the following figure. The Action structure is composed of “POU name of SFC where Action is at”, “Action name”, and “Action member”. User can select appropriate POU and Action names first, and then decide Action member.

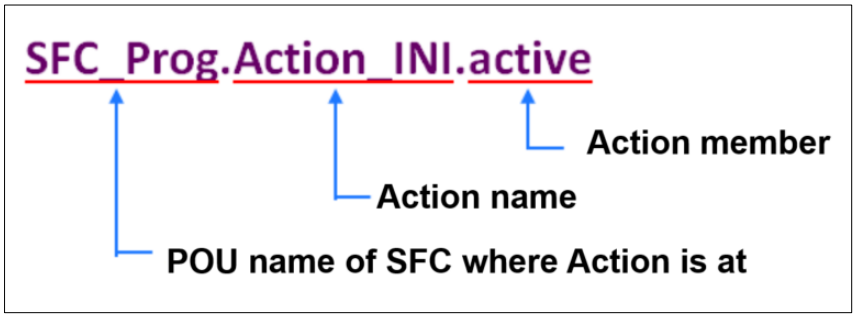


Figure 8 - 178: Action Property Parameter

The Transition structure is composed of “POU name of SFC where Transition is at”, “Transition name”, and “Transition member”. User can select appropriate POU and Transition names first, and then decide Transition member.

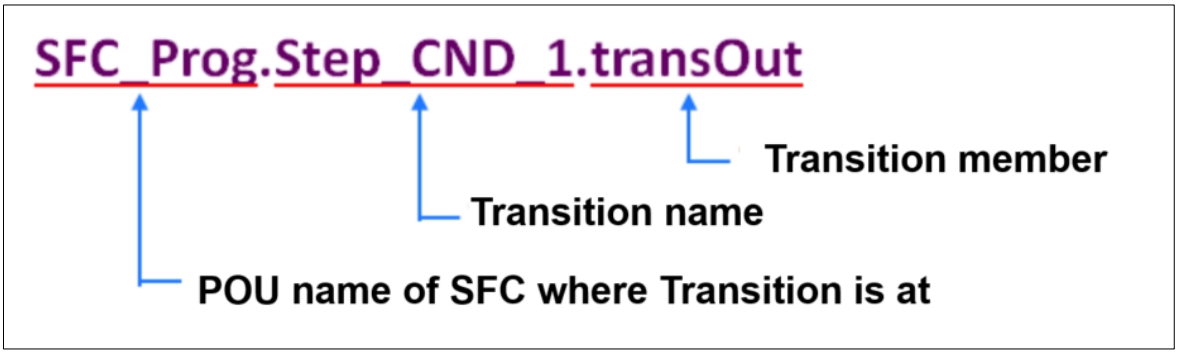


Figure 8 - 179: Transition Property Parameter

The following table shows different members of Action and Transition and their functions.

Type	Member	Description	Read/Write
Action	enabled	When qualifier of Action is TRUE, this SFC property parameter is ON, and user can tell if Action is enabled.	Read only
	active	When the first Action execution is complete, the SFC property parameter will be ON.	Read only
	rstDisable	When the Final scan function is enabled, the value is OFF. When the Final scan function is disabled, the value is ON.	Read only
	counter	When the **active** property of Action changes from OFF to ON, the property parameter count adds one.	Read only
	setTimer	Time display of the time-related qualifier (Unit: **ms**; data length: **DWORD**)	Read only
Transition	transOut	When the Transition condition is True: ON Or When Transition is forced on: ON	Read only

Take the member counter of Action in SFC as example: If user want to monitor the number of times the active property parameter of Action changes from 0 to 1, and

send this active property parameter to D100 register, the program description will be as the following. Use SM400 normally open contact and SMOV instruction in the POU of LD program, click the S pin of DMOV, select POU SFC_Prog of SFC program, Action name, the name of member property parameter, and finish the selection. After that, enter D100 in the D pin of the DMOV instruction.

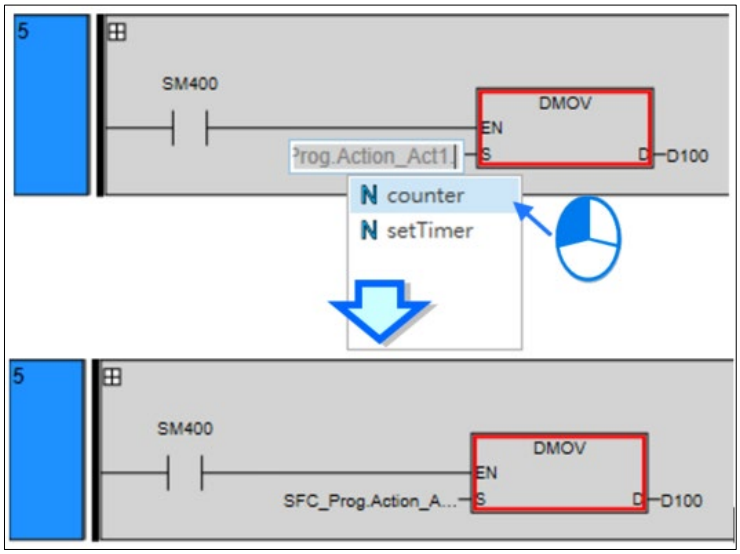


Figure 8 - 180: Use Property Parameter

After that, when the program is running, every time when the SFC_Prog.Action.Act_1.Active action is complete, the SFC_Prog.Action.Act_1.counter counter adds one.

8.3.3 ST - Structured Text

The Structured Text (ST) is a high level language similar to C or PASCAL. It is composed of texts. Compared with Instruction Lists (IL), Structured Text is more readable. It is very flexible and intuitive for writing control algorithms. People who are trained in programming languages often find it easiest language to use for programming control logic. Besides, Structured Texts provide familiar programming environment for programmers who are familiar with C language.

Basic structure of Structured Text:

Structured Text has the same function as the ladder diagram in the following figure. Structure of the structured text is similar to the syntax of a high level language. A structured text is composed of statements.

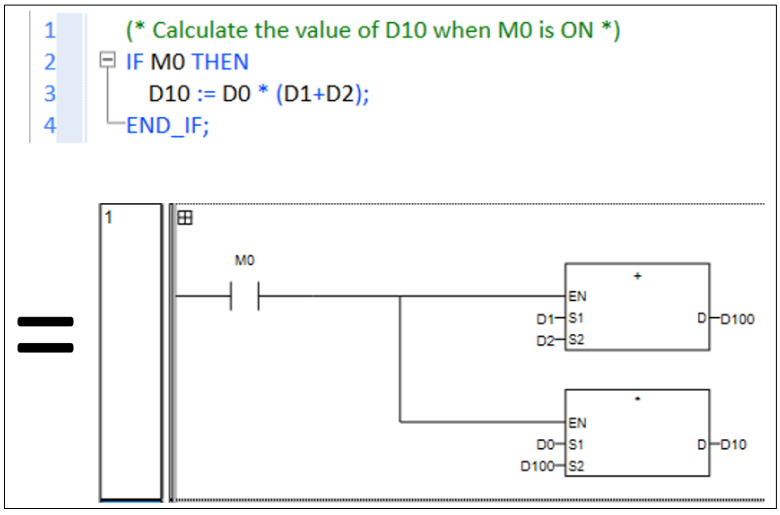


Figure 8 - 181: Example of similar ST and LD programs

The Structure of the structured text is shown in the following figure.

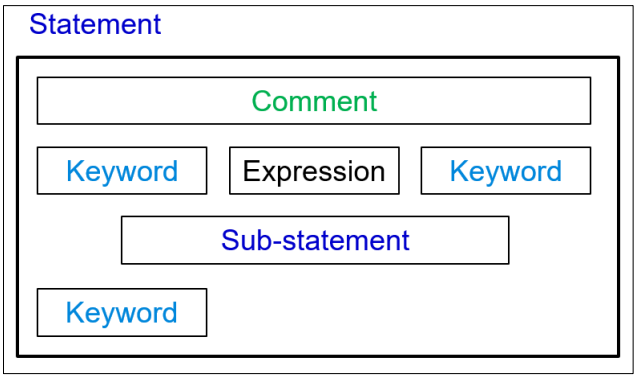


Figure 8 - 182: ST Structure

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.

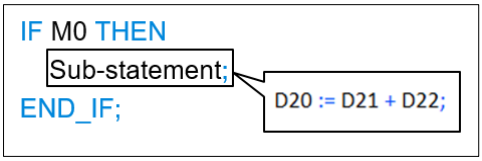


Figure 8 - 183: ST Statement

A complete statement corresponds to a network in ladder diagram. It represents a task. For example, let's take the scenario in the following figure. 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 following figure is not a valid statement. It is an expression. It is not a concrete task. It is like the ladder diagram shown in the following figure. The ladder diagram lacks an operand. It is not a complete program which can be executed.

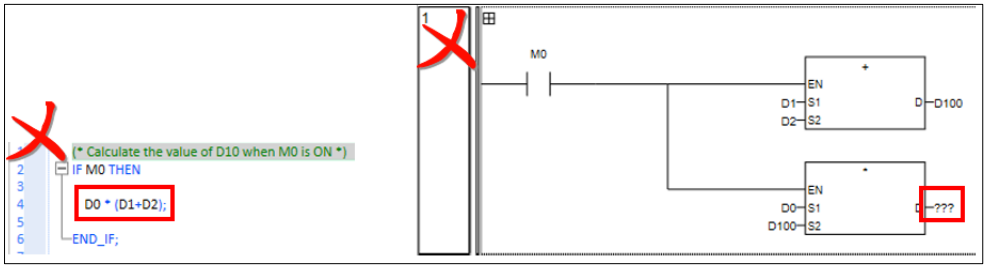


Figure 8 - 184: Example of incomplete ST and LD programs

A structured text is composed of statements and a statement is composed of elements. An example of a structured text is shown in the following figure. The complete statements in the structured text are in the red frames.

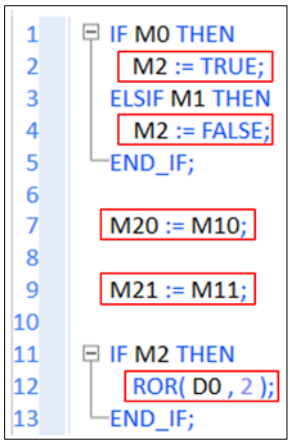


Figure 8 - 185: Example of ST Statement

NOTE:

In a Structured Text, ":=" means 'to assign' and

"=" means "if equal to'

Expression:

An expression is an important element of a statement. It represents a value. An expression can be an arithmetic expression, a constant, a variable, or a device.

Examples:

- M0 & M1 (Boolean value)

The logical operator & takes M0 and M1 and performs the logical & operation on M01 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 if OFF, then 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

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.

In 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 is calculated and the result of the operation is 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.

```
1   (* This logic calculates the sum of D21 and D22 and stores in D20 when M0 is TRUE *)
2   IF M0 THEN
3       D20 := D21 + D22;
4   END_IF;
5
6   //This logic sets the value of M1 based on value of D20
7   IF D20 = D0 * (D1 + D2) THEN
8       M1 := TRUE;
9   ELSE
10      M1 := FALSE;
11  END_IF;
```

Figure 8 - 186: Example IF Statement

Operand and Operator:

Operand and Operators are basic elements of an expression. An Operand is the Object of an expression and an operator is a variable 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, variables or constants.

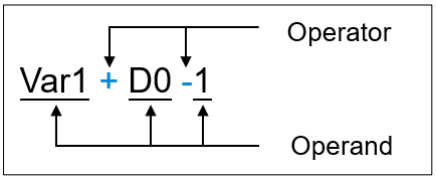


Figure 8 - 187: Operator and Operand

Rules of precedence affect which values form operands for which operators. If the operator in an expression share the same precedence, the order of the operation is carried out from left to right. The operands used in a Structured Text in DIADesigner are described in the following table.

Symbol	Function	Data Format		Example of an expression		Precedence
		Operand	Result of an operation (Value of an expression)	Expression	Value	
( )	The parenthetical part of an expression is calculated first	Unspecified	Unspecified	(D0 + 6) * 3	33	Highest
**	Exponentiation	Floating-point number	Floating-point number	2.0 ** 5.0	3.2E+1	
-	Negative sign	Signed number	Signed number	-(D0 + 3)	-8	
NOT	Logical Negation	Boolean value	Boolean value	**NOT** M0	TRUE	
*	Multiplication	Any value	Any value	D0 * 3	15	
/	Division	Any value	Any value	15 / D0	3	
MOD	Remainder	Integer	Integer	D0 MOD 3	2	
+, -	Addition, Subtraction	Any value	Any value	D0 + 3	8	
<, >, <=, >=	Comparison	Any value	Boolean value	D0 > 2	TRUE	
=, <>	Equal to, Not Equal to	Any value	Boolean value	D0 <> 2	TRUE	
		Boolean value		M0 = TRUE	FALSE	
AND, &	Logical AND operation	Boolean value	Boolean value	M0 & M1	FALSE	
XOR	Logical Exclusive OR operation	Boolean value	Boolean value	M0 XOR M1	TRUE	
OR	Logical inclusive OR operation	Boolean value	Boolean value	M0 OR M1	TRUE	Lowest

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 are 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 variables. However, the words which are regarded as keywords in a program can be parts of the variables in the programs. For example, FIFO and _IF are allowed.

In a structure text, the words between (* and *) are regarded as Comments. When a program is compiled, the system automatically skips the words between (* and *) in the program. Besides, as long as the structure of a structured text is not destroyed, comments can be put anywhere in the structured text. However, user have to be careful about the readability of the structured text.

Example: The comments in the left example are valid. The comment in the right side example divides the keyword END_IF into two parts and therefore an error will occur if the program is compiled.

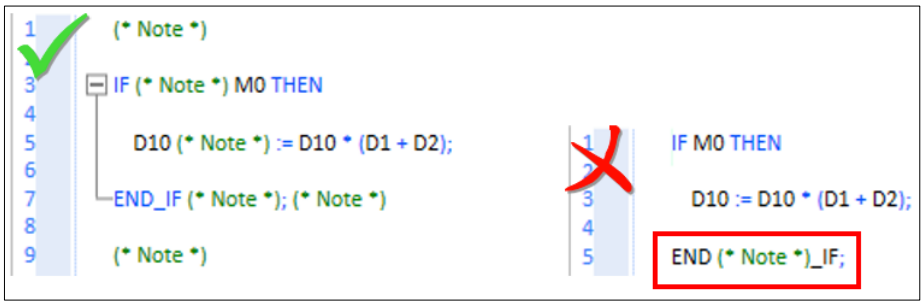


Figure 8 - 188: Example of ST Comment

Using a Variable whose datatype is ARRAY in a Structured Text:

If user want to use a variable whose datatype is ARRAY in a structured text, the expression format is identifier[index]. The index is a constant or a variable. However only the indexes in a project for an AS series CPU module can be variables. The minimum index value must be 0, whether the index is a constant or a variable. The maximum index value cannot be larger than or equal to the number of elements in an array. For example, if the number of elements in an Array is 10, the index values must be between 0 and 9.

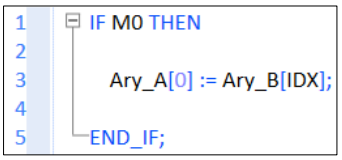


Figure 8 - 189: Array Variable in ST

Besides, user have to pay attention to the following points when they use a variable whose data type is array.

- User can only manipulate a single element in an array. A variable which represents an array cannot be used as an operand.
- The arrays that DIADesigner 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 shown in the following figure.

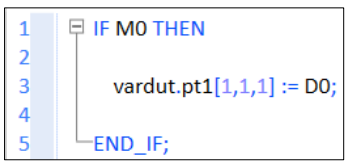


Figure 8 - 190: Thee-dimensional Array in ST

- If an index is a variable, and the value of the variable is not within a valid 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 variable is constant, the variable can be modified by an index register. If the index register for a variable is variable, the variable cannot be modified by an index register.

Important points about Structured Texts:

- 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 DIADesigner are regarded as different characters. Avoid using full width characters lest they should be identified incorrectly.
- Structured Text language is case-insensitive. IF and if are considered to be equivalents of if (The C language is case-sensitive)
- If user want to use constants in a ST POU in DIADesigner, the constants must be represented in the following ways:
  - Decimal value: 23456 (A value which is not preceded by any mark is regarded as a decimal value)
  - Hexadecimal value: 16#5BA0 (A value which is preceded by 16#)
  - Octal value: 8#55640 (A value which is preceded by 8#)
  - Binary value: 2#1011 (A value which is preceded by 2#)
  - String: “XYZ” (Characters are put in double quotes)
  - Boolean value (BOOL): TRUE or FALSE
- In a ST POU in DIADesigner, the data types of the operands in an expression must be the same. However, if the data type of one of the 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 of the operand is DINT and the datatype 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, then the operands cannot take part in the operation.

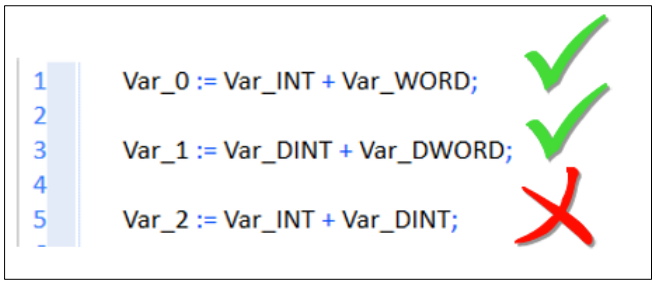


Figure 8 - 191: Data Types in ST

- If the data type of a Variable is WORD, DWORD, the system regards it as a signed integer during the operation.
- Although there is no limit to the number of lines, user still have to consider the capacity of the memory in the PLC.
- The sections of a ST in DIADesigner can be copied/cut/pasted. User can copy the edited text into a ST in DIADesigner.
- The following instructions are not supported in a ST:

AS series CPU modules
All the Instruction List (IL) instructions are not supported. For example, LD, LDI, OUT, SET, RST, LD and 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 counter in subroutine program are not supported.

- The following instructions are supported in a ST but with limitation:

AS series CPU modules
All the high speed output instructions (API 27xx) for example, DPLSY and DDRVI are supported. Refer DDRVI instruction for application.

AS series CPU modules
The user of DCNT instruction for high speed counters is supported. But, statements need to be executed so that the 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.

8.3.3.1 Add ST program

User can select **ST** in the **Language** field in **Add POU** window while creating a POU of program type or function block type. Refer *8.2.1 POU* for detailed information.

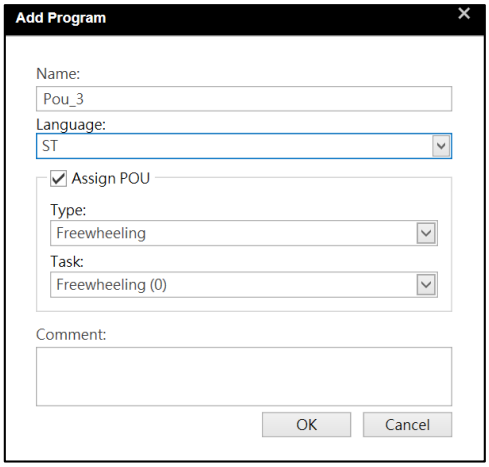


Figure 8 - 192: ST POU

The environment in which a ST can be edited is shown in the following figure. In any POU window (program type or function block type), there are two areas:

- Local Variable table area
- POU Working area

The table in the upper part of the ST POU window is a Local Variable Table and the lower part is POU working area. After a program editing window to create a ST POU is opened, the corresponding tool bar will appear in the DIADesigner window. There are line numbers at the left side of the working area.

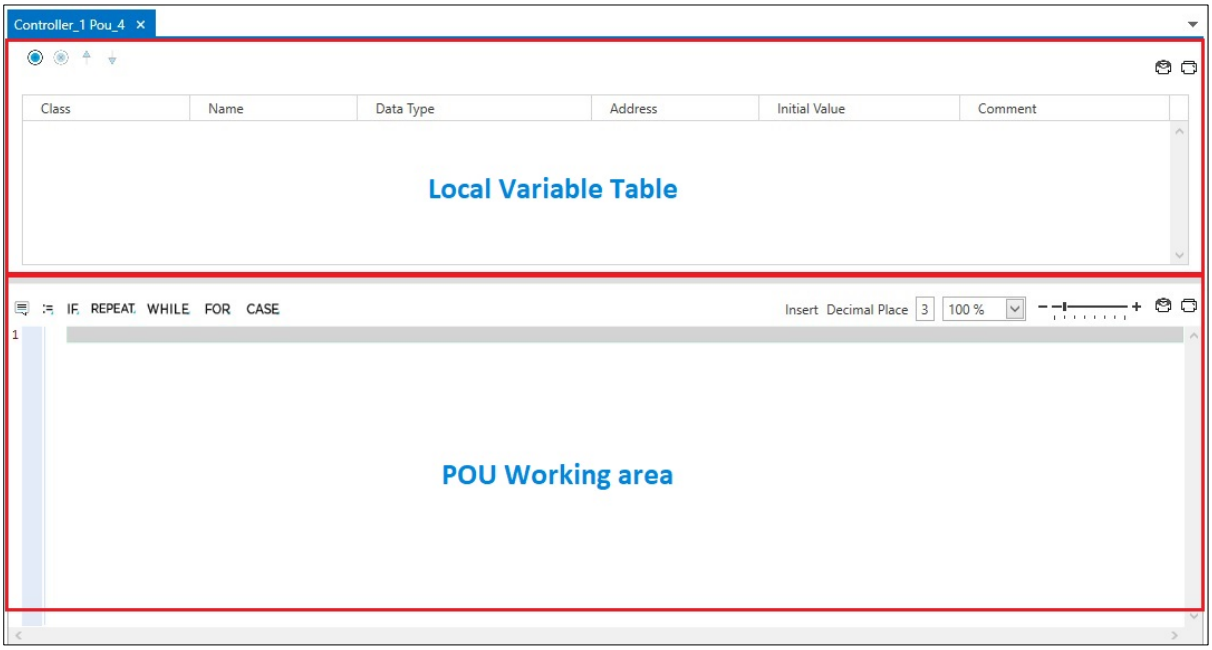


Figure 8 - 193: ST POU - Local Variable Table and POU working area

ST Tool bar:

After the POU window is opened, the toolbar for editing the Structured Text displays as shown in the following figure.

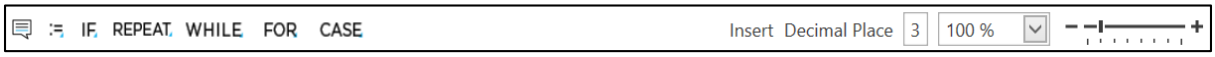


Figure 8 - 194: ST POU Toolbar

The functions in ST POU window are described in the following table:

Function/Icon	Description
	Edit/Delete Line Comment
	Assignment Structure :=
	Conditional Structure IF

Function/Icon	Description
REPEAT	Loop Structure REPEAT
WHILE	Loop Structure WHILE
FOR	Loop Structure FOR
CASE	Conditional Structure CASE

8.3.3.2 ST Expressions and Statements

Structure of a statement

Assignment structure - :=:

Format:

```
Address/Variable := Expression;
```

Description:

The value represented by the expression at the right side of **:=** is assigned to the address or the variable at the left side of **:=**.

Rules and Limits:

- The expression at the right side of **:=** can be a constant, an address, a variable or an arithmetic expression. Only a address or a variable can be at the left side of the **:=**.
- The operands at both sides of **:=** must conform to the rules listed below.
  - One operand is an M/S/T/C/HC address, and the other operand is a variable whose data type is BOOL/STEP/TIMER/COUNTER (the address assigned to the operand which is a variable is a bit address)
  - One operand is a D/L address, and the other operand is a variable whose data type is not BOOL/STEP/REAL/STRING.

- One operand is a T/C address, and the other operand is a D/L address or a variable whose data type is WORD/INT/TIMER/COUNTER.
- One operand is a D/L/HC address or a variable whose datatype is COUNTER and to which a HC address is assigned. The other operand is a variable whose datatype is DWORD/DINT
- One operand is a variable whose datatype is WORD/DWORD/LWORD and the other operand is a variable whose data type is INT/DINT/LINT. The data length of the operand at the left side of **:=** must be larger than or equal to the data length of the operand at the right side of **:=**.
- Both operands are variables whose datatypes are REAL/LREAL/STRING. The data lengths of the operands must be the same.

Examples:

Example 1: The value of M0 is assigned to the variable OUT_0

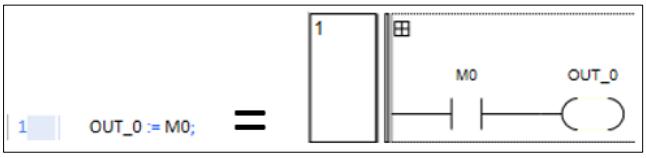


Figure 8 - 195: Assignment Structure – Example 1

Example 2: M0 is set to ON.

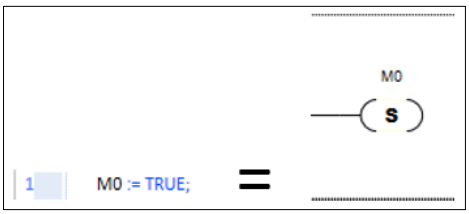


Figure 8 - 196: Assignment Structure - Example 2

Example 3: The value in D2 is added to the value in D1. The sum is stored in D0.

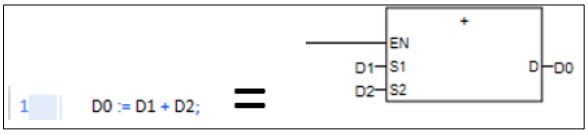


Figure 8 - 197: Assignment Structure - Example 3

Example 4: 3 is assigned to the variable DATA.

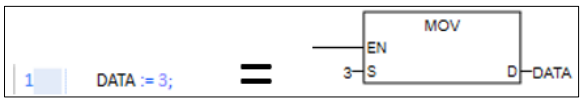


Figure 8 - 198: Assignment Structure - Example 4

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.

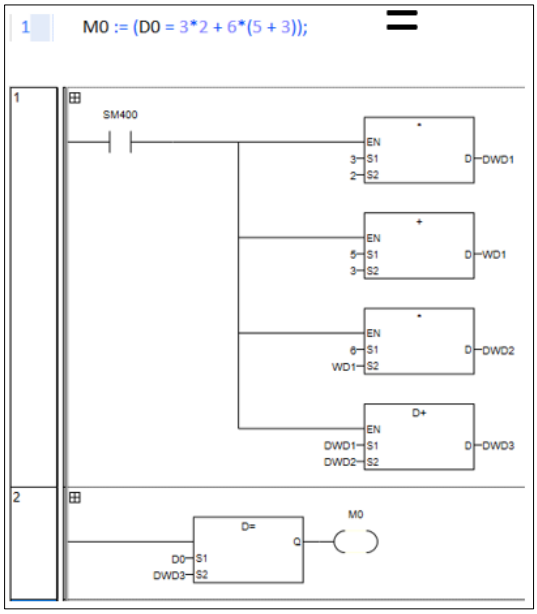


Figure 8 - 199: Assignment Structure - Example 5

Example 6: Examples of incorrect use of the assignment structure are mentioned in the following table:

Statement	Description
D0:= M0;	The datatype of one operand must not be different from the datatype of the other operand.
V_WORD:= V_DWORD;	The data length of the operand at the left side of **:=** must not be less than the data length of the operand at the right side of **:=**.
D0:= V_REAL;	The value of a variable whose datatype is REAL cannot be assigned to a D address.
V_LREAL:= V_REAL;	If one operand is a variable whose datatype is REAL, the other operand must be a variable whose datatype is REAL. If one operand is a variable whose datatype is LREAL, the other operand must be a variable whose datatype is LREAL.
V_DWORD:= V_TIMER;	The value of a variable whose datatype is Timer cannot be assigned to a variable whose datatype is DWORD, but can be assigned to a variable whose datatype is WORD.
V_WORD:= HC0;	The value in a HC address cannot be assigned to a variable whose datatype is WORD, but can be assigned to a variable whose datatype is DWORD

Conditional Structure - IF:

Format:

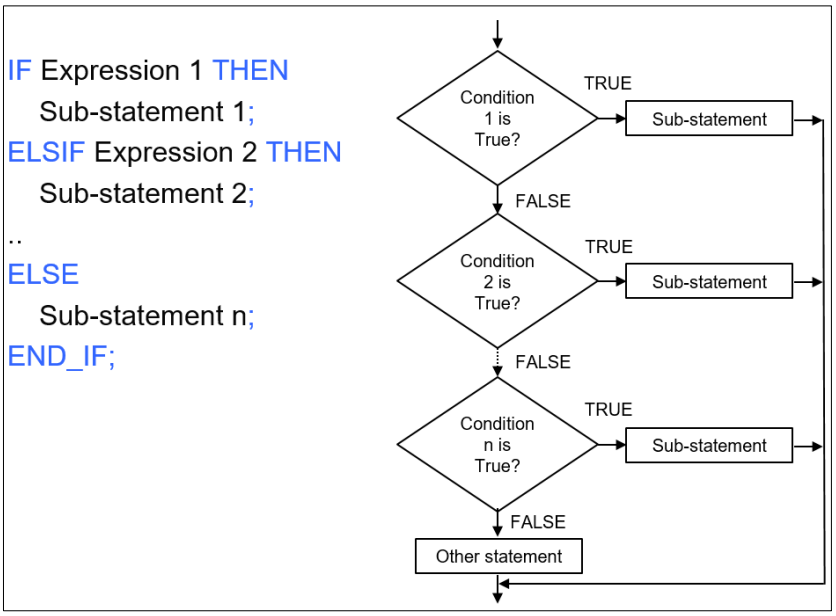


Figure 8 - 200: Condition Structure - IF - format

Description:

If the value of **Expression 1** is TRUE, **Sub-statement 1** is executed. If the value of **Expression 1** is FALSE, **Expression 2** is evaluated. If the values of all the Expressions are FALSE, **Other sub-statement** are executed.

Rules and Limits:

- A Boolean expression can be an address, a variable or an arithmetic expression. It cannot be a constant.
- A sub-statement can be a valid statement, or an **IF** statement.
- There is not 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 is executed.
- User can add **ELSIF** sections to a conditional structure at will. There is only one **ELSE** section in a condition structure. The **ELSE** section in a condition structure is at the last section of the conditional structure.

Examples:

Example 1: If M0 is ON, M1 is set to ON and M2 is set to OFF. If M0 is OFF, then the step will end. No matter the value of M0 is, the value of M0 is assigned to M0.

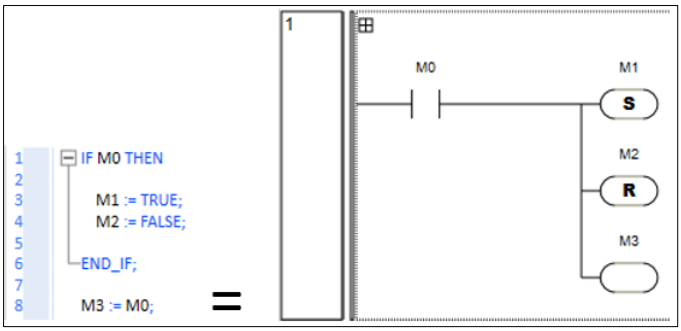


Figure 8 - 201: Conditional Structure IF - Example 1

Example 2: If the value of DT is larger than 5, OUT_0 is set to ON and OUT_1 is reset to OFF. If the value of DT is not larger than 5, the value of OUT_2 is assigned to Out_0 and Out_1.

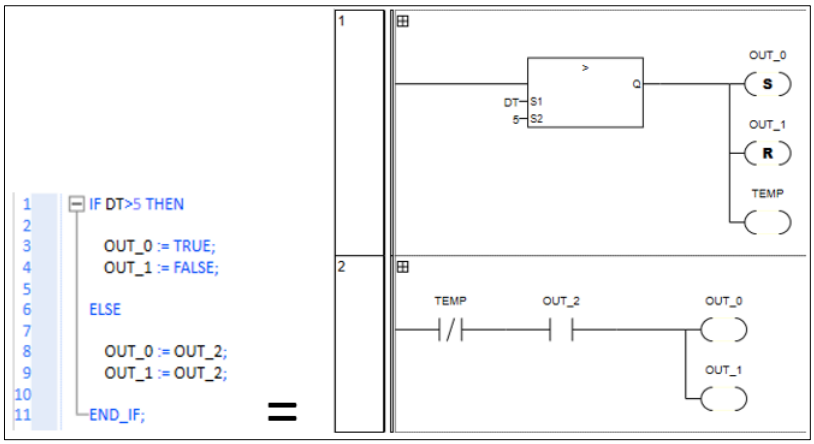


Figure 8 - 202: Conditional Structure IF - Example 2

Example 3: If M1 is ON, the value in D0 is 1. If M1 is OFF, the PLC will evaluate M2. If M2 is ON, the value in D0 is 2, If M2 is OFF, the value in D0 is 0.

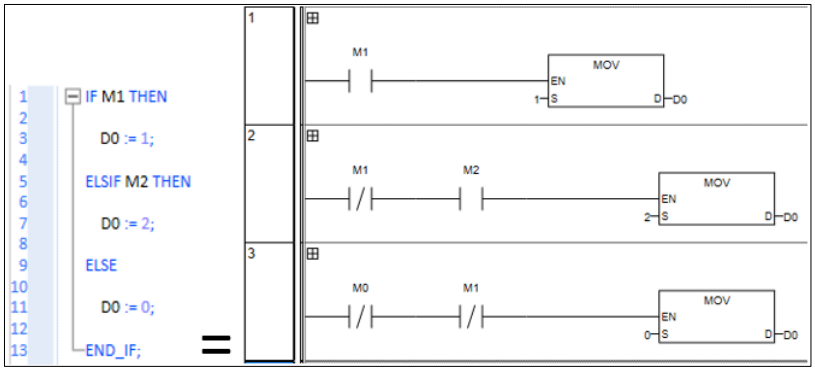


Figure 8 - 203: Conditional Structure IF - Example 3

Conditional Structure – CASE:

Format:

```
CASE Expression OF
   Conditional value 1 := Sub-statement 1;
   Conditional value 2 := Sub-statement 2;
   .
   .
   Conditional value n := Sub-statement n;
ELSE
   Sub-statement;
END_CASE;
```

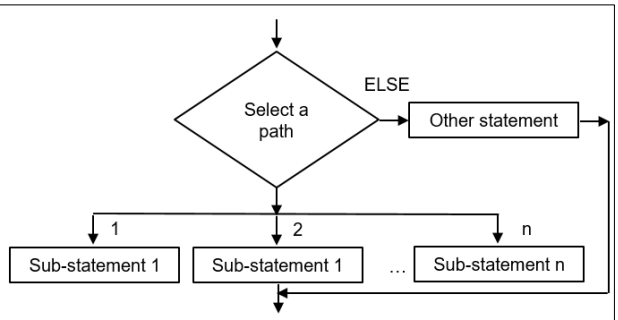


Figure 8 - 204: Conditional Structure CASE - format

Description:

The PLC judges whether the value of **Expression** is on the list of condition values. If the value of **Expression** is the same as a condition value on the list, the sub-statement following the condition value are executed. If the value of **Expression** is not on the list of condition values, **Other sub-statement** under the keyword **ELSE** are executed.

Rules and Limits:

- **Expression** in the above figure can be an address, a variable or an arithmetic expression. It cannot be a constant. The datatype of **Expression** must be INT (-32768 ~ 32767) or DINT (-2147483648 ~ 2147483647). If an address is used as **Expression**, the system will automatically regard the value in the address as an integer. Besides, if an arithmetic expression is used as **Expression** and the result of the operation is not an integer, the result is rounded down to the nearest whole digit.

- A condition value on the list cannot 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 datatype of **Expression**. If the datatype is INT, the condition values must be in the range of -32768 ~ 32767. If the datatype is DINT, the condition values must be in the range of -2147483648 ~ 2147483647.
- If the sub-statements following some conditions values are same, user 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 is executed if the value of **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 is executed if the value of **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 not valid.
- A sub-statement can be valid statement or another structure.
- There is not 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 **Expression** is not on the list of condition values, no sub-statement is executed.
- After a sub-statement is executed, the execution of the program will jump out of the **CASE** structure. No Jump instruction is needed.

Examples:

Example 1: If the value of D0 is 1, M1 is ON. If the values of D0 is 2, M2 is ON. If the value of D0 is neither 1 nor 2, M0 is ON.

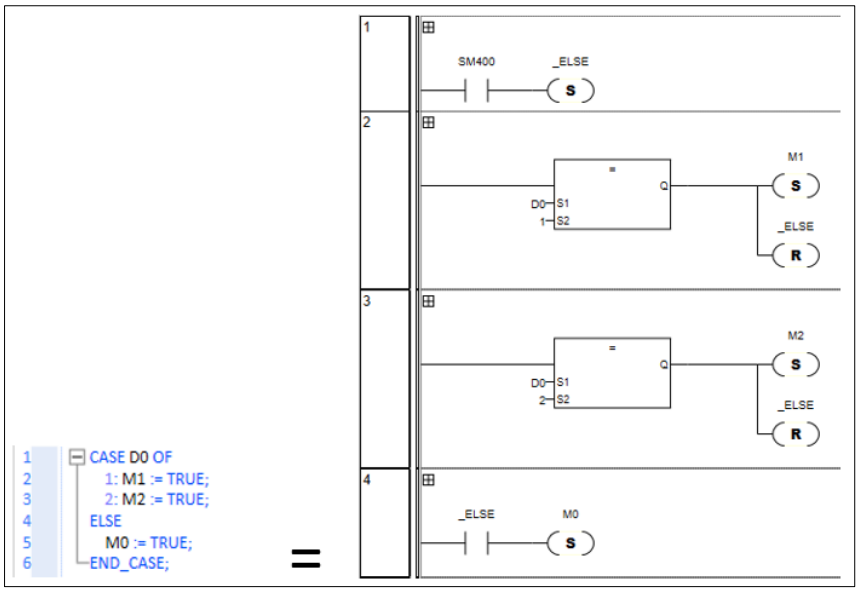


Figure 8 - 205: Conditional Structure CASE - Example 1

Example 2: The value of arithmetic expression D0+D1 is calculated first. If the result of the operation is in the range of 1 to 3, M0 is ON. If the result of the operation is 5 or 7, M1 is ON. If the result of the operation is not on the list of condition values, no sub-statement is executed and the execution of the program will jump out of this structure.

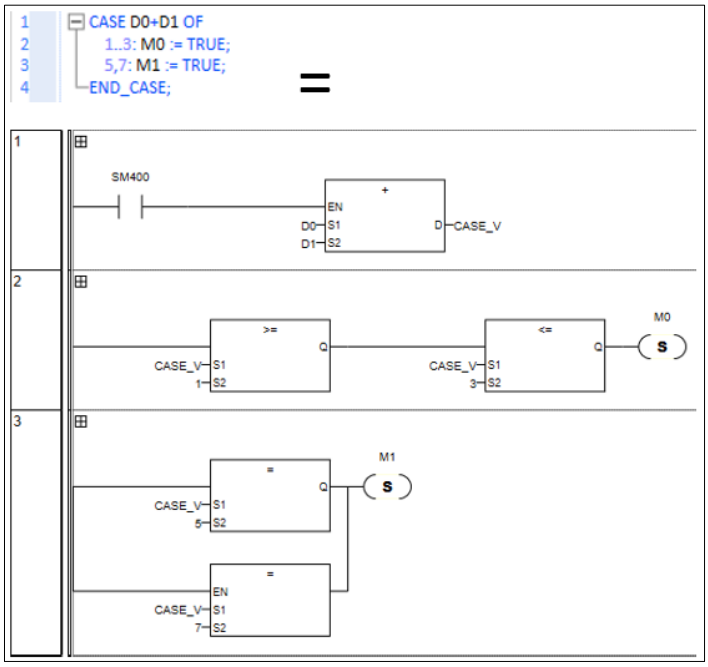


Figure 8 - 206: Conditional Structure CASE - Example 2

Loop Structure – REPEAT:

Format:

```
REPEAT
   Sub-statement;
UNTIL Expression;
END_REPEAT;
```

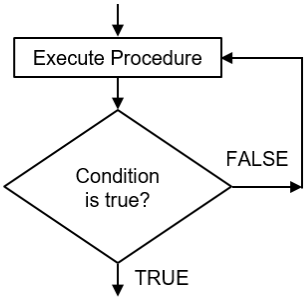


Figure 8 - 207: Loop Structure - REPEAT – format

Description:

The sub-statement is executed and then the **Expression** (BOOL) is evaluated. If the **Expression** is evaluated as TRUE, the loop is terminated. If the value of the **Expression** is FALSE, the program in the structure is executed repeatedly until the value of the **Expression** is TRUE.

Rules and Limits:

- The Boolean expression can be an address, a variable or an arithmetic expression. It cannot be a constant.
- The sub-statement is executed before the **Expression** is evaluated and therefore the sub-statement is executed at least once.
- The sub-statement can be a valid statement or another structure. There is not 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 **Expression** following **UNTIL** is TRUE, the loop is

terminated. To prevent an infinite loop, the operand in the **Expression** following **UNTIL** cannot be an address or a variable which cannot be updated.

- The X address in a program are updated after the program is scanned. If user want to use and a X address as the operand of the **Expression**, they have to add an I/O update instruction to the structure, or update the X through an interrupt subroutine.

Examples:

Example 1: The initial value of DT is 0. In the loop structure, the values of DT increases progressively by 2. The loop will not be terminated until the values of DT is larger than 100.

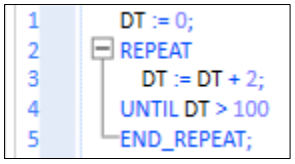


Figure 8 - 208: Loop Structure - REPEAT - Example 1

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 is %*4*3*2*1.

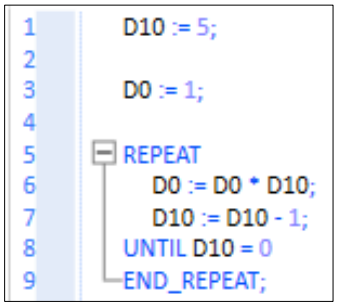


Figure 8 - 209: Loop Structure REPEAT - Example 2

Loop Structure – WHILE:

Format:

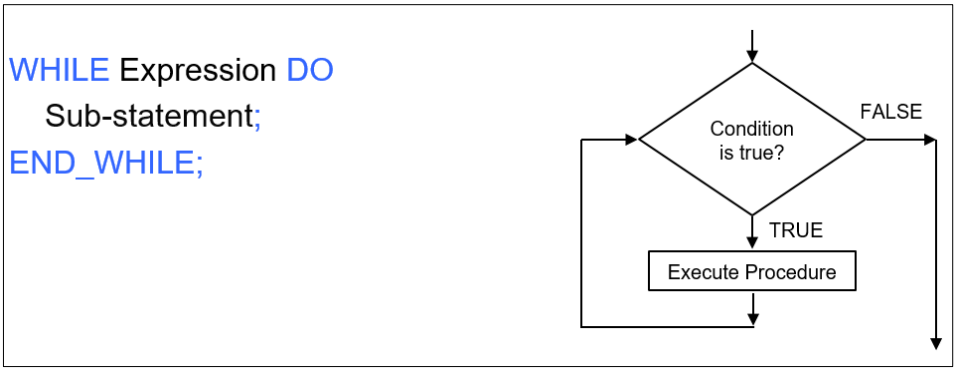


Figure 8 - 210: Loop Structure WHILE - format

Description:

The **Expression** (BOOL) is evaluated. If the value of the **Expression** is TRUE, the sub-statement is executed. This repeats until the value of the **Expression** becomes FALSE.

Rules and Limits:

- The **Expression** can be an address, a variable or an arithmetic expression. It cannot be a constant.
- The sub-statement is executed if the value of the **Expression** is TRUE and therefore the sub-statement may not be executed.
- The sub-statement can be a valid 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 **Expression** following **WHILE** is FALSE, the loop is terminated. To prevent an infinite loop, the operand in the **Expression** following **WHILE** cannot be an address or a variable which cannot be

updated.

- The X address in a program are updated after the program is scanned. If user want to use an X address 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.

Examples:

Example 1: The initial value of DT is 0. In the loop structure, the value of DT increases progressively by 2. The loop will not be terminated until the value of DT is larger than 100.

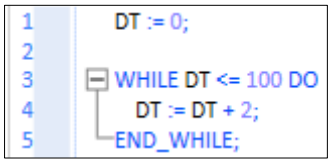


Figure 8 - 211: Loop Structure WHILE - Example 1

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 becomes 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.

```
1     M0 := TRUE;
2     D0 := 0;
3     D1 := 0;
4
5     REPEAT
6         D0 := D0 + 1;
7     UNTIL M0
8     END_REPEAT;
9
10    WHILE (NOT M0) DO
11        D1 := D1 + 1;
12    END_WHILE;
```

Figure 8 - 212: Loop Structure WHILE - Example 2

Loop Structure – FOR:

Format:

```
FOR Address or Variable := Start Value TO End Value BY Increment DO
   Sub-statement;
END_FOR;
```

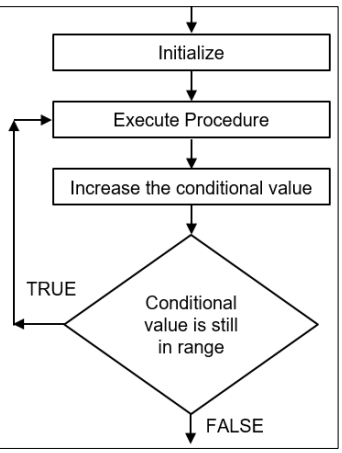


Figure 8 - 213: Loop Structure FOR - format

Description:

The initial value of the **Variable** or the **Address** is the **Start Value**. Whenever the loop body is executed, the value of the **Variable** or the **Address** increases by the **Increment**. If the value of the **Variable** or the **Address** is in the range of the **Start Value** to the **End Value**, the loop body is executed. If the value of the **Variable** or the **Address** is not in the range of the **Start Value** to the **End Value**, the loop is terminated.

Rules and Limits:

- The **Start Value**/**End Value**/**Increment** value can be an address, a variable, an arithmetic expression or a constant. The datatype of the **Start Value**/**End Value**/**Increment** must be INT (-32768 ~ 32767) or DINT (-2147483648 ~ 2147493647). If an address is used as the **Start Value**/**End Value**/**Increment**, the system will automatically regard the value in the address 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 is rounded down to the nearest whole digit.
- If the **End Value** is less than the **Start Value**, the increment must be given a negative number. Otherwise there is an infinite loop.
- The datatype of the variable must be INT or DINT. If the variable or the address takes part in the operation, prevent the value of the variable or the address from resulting in an infinite loop.
- The sub-statement can be valid statement or another structure. There is not 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 variable or the address increases by the **Increment**, the system will check whether the value of the variable or the address is in the range of the **Start Value** to the **End Value**. If an overflow occurs after the value of the variable or the address increases by the **Increment**, there is an infinite loop.

Example:

The initial value of the variable INDEX is 1. Whenever the loop body is executed, the value of the INDEX increases by 2. After the loop body is executed for the fifth time, the value of the INDEX will increase by 2 and the value of the INDEX becomes 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 variable SU is 0. After the loop is terminated, the value of SU becomes $1^2+3^2+5^2+7^2+9^2$.

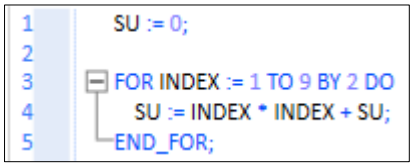


Figure 8 - 214: Loop Structure FOR - Example

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 in the following figure shows that no action is executed. Owing to the blank statement, the execution of the program is delayed. User can adjust the time for which the execution of the program is delayed by adjusting the number of times the loop body is executed.

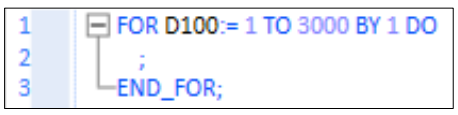


Figure 8 - 215: Blank Statement in FOR Loop

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 FB1 is a Structured Text. After the upper part of the program is executed, IN_1 is compared with IN_2. If IN_1 is equal to IN_2, then the execution will leave the current program. If IN_1 is not equal to IN_2, then, the program in the Function Block is executed.

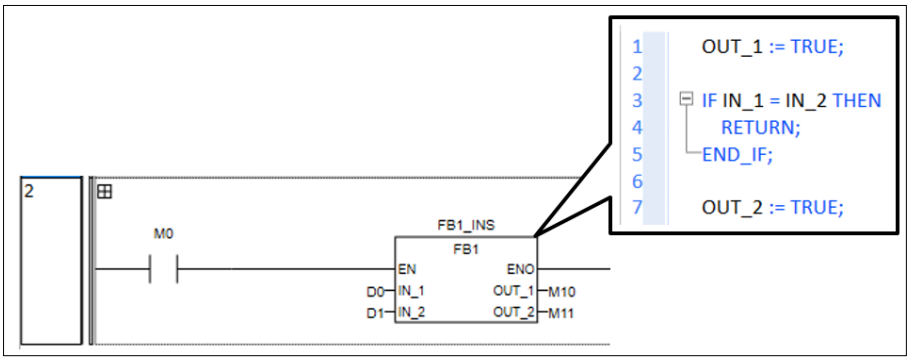


Figure 8 - 216: RETURN Statement - Example

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.

Examples:

Example 1: The loop in the following figure is a **REPEAT** loop. The loop will not be terminated until the value of DT is 0. The IF structure in the loop is used with an **EXIT** statement. The initial value in D0 is 0. It increases by 1. The loop will not be terminated until the value in D0 is 100. As a result, the loop body is executed 100 times at most. In the following figure, the initial value of DT is 110. After the loop body is executed for the 100th time, the value of DT becomes 10. In other words, the final value of DT is 10. The value of DT will not decrease to 0.

```
58      DT := 110;
59      D0 := 0;
60
61      REPEAT
62          DT := DT - 1;
63          D0 := D0 + 1;
64
65          IF D0 = 100 THEN
66              EXIT;
67          END_IF;
68
69      UNTIL DT = 0
70      END_REPEAT;
```

Figure 8 - 217: EXIT Statement – Example 1

Example 2: The loop in the following figure is a FOR loop. The value of the arithmetic expression D0 * D10 is calculated and the result of the operation is stored in D0. The IF structure in the loop is used with an EXIT statement. The loop will not be terminated until the value in D0 is larger than 100. When the value in D10 increases to 5, the value in D0 is 1*2*3*4*5 and the loop is terminated.

```
72      DT := 10;
73      D0 := 1;
74
75    FOR D10 := 1 TO DT BY 1 DO
76
77        D0 := D0 * D10;
78
79    IF D0 > 100 THEN
80        EXIT;
81    END_IF;
82
83  END_FOR;
```

Figure 8 - 218: EXIT Statement - Example 2

8.3.3.3 Edit ST

Editing ST Program:

The way in which a ST program is edited is similar to the way in which a text file is edited with a general text editor. If user want to edit ST program, they can type or modify the text in the working area. If the user want to start a new line of text at a specific point, they can press Enter on the keyboard.

Figure 8 - 219: ST - Edit from keyboard

Besides, the user can click a line which can be edited and insert a conditional or loop structure by clicking on the respective icon in the tool bar.

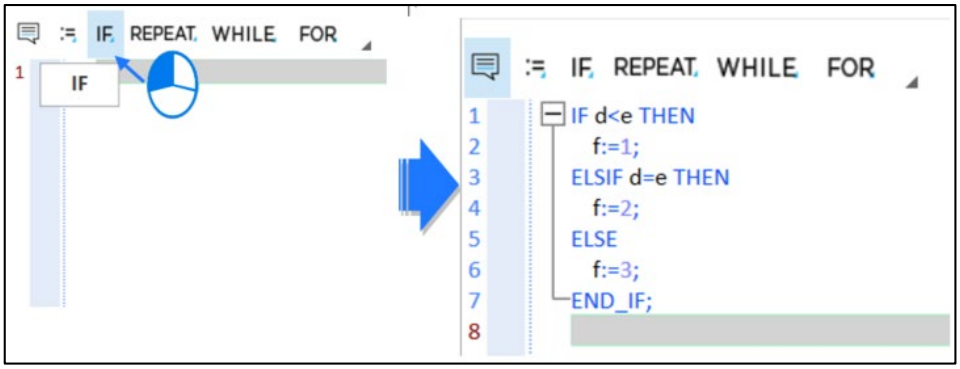


Figure 8 - 220: ST - Edit from Toolbar

If user 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 user 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 user type a Variable which has not been declared, they can right-click the variable to open its context menu. Click **Add Variable** in the context menu to open **Add a Variable to POU** window. Select the required properties and click **OK**.

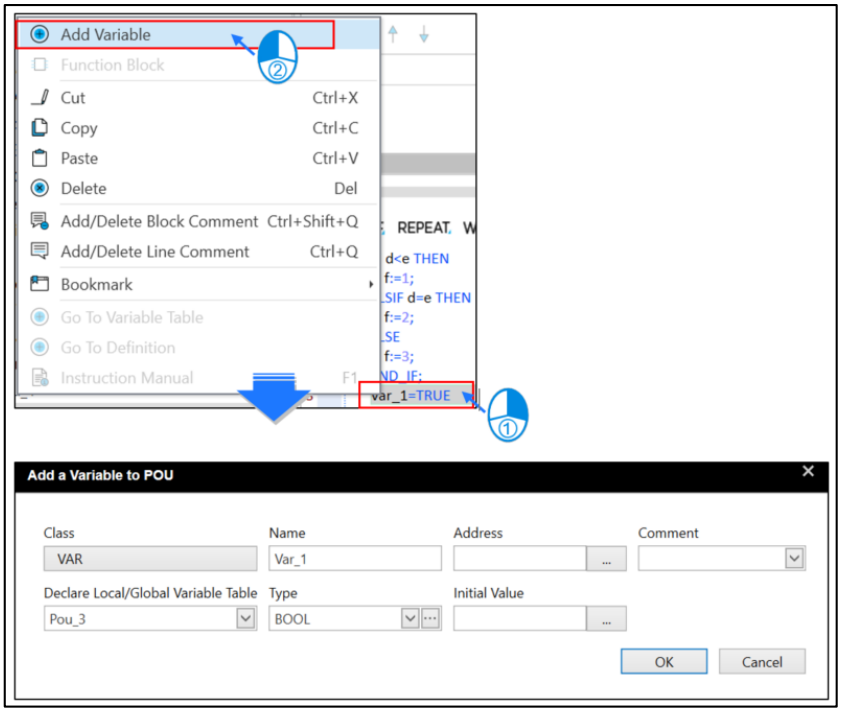


Figure 8 - 221: Add Variable to ST POU

Line Comment:

After creating a program, if user don't want to compile and execute some portion of the ST, they can comment them with the help of **Line Comment**. Compiler will ignore the lines which are commented and do not execute them. Commented lines display in green color prefixed with //.

Follow these steps to Edit/Delete Line Comment:

1. Select the lines to be commented
2. Click icon on the **ST POU** Tool bar or

   Right-click on the lines to open the context menu and select **Edit/Delete Line Comment** or

   Press **Ctrl + Q** keys simultaneously on the keyboard.

   Result: Selected lines are commented as shown in the following figure.

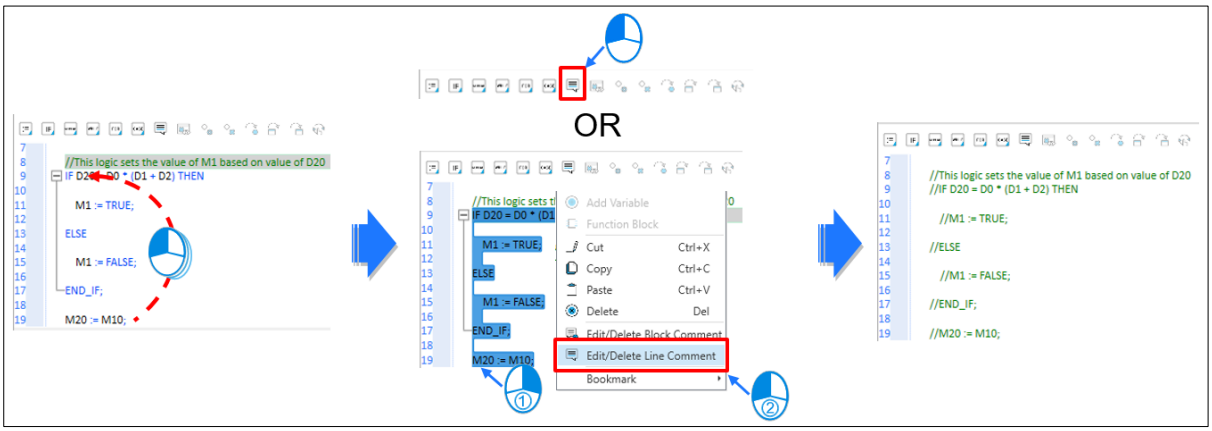


Figure 8 - 222: ST Line Comment

3. Repeat the above step once again to delete the Line Comment

   Result: Line Comment is deleted and the ST lines are Compiled next time when user compiles this program.

Block Comment:

Block Comment is one simpler way to comment the lines in a ST program. Instead of prefixing all the lines, the compiler will insert (* in the beginning of the first commented line and a *) at the end of last commented line. Commented lines display in green color.

Follow these steps to Edit/Delete a Block Comment:

1. Select the lines to be commented
2. Right-click on the lines to open the context menu and select **Edit/Delete Block Comment** or

   Press **Ctrl +Shift + Q** keys simultaneously on the keyboard.

   Result: Selected lines are commented.

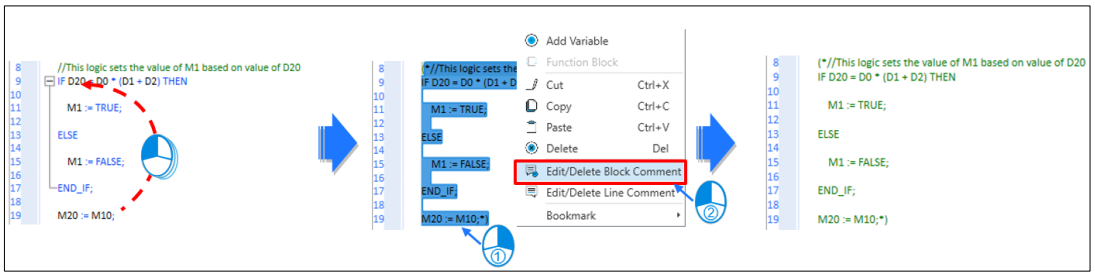


Figure 8 - 223: Block Comment

3. Repeat the above step once again to delete the Block Comment.

   Result: Block Comment is deleted and the ST lines are Compiled next time when user compiles this program.

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 need not 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 parenthesis.
- The order in which the operands are arranged cannot be changed. Refer instruction manuals for more details.
- The applied instruction in DIADesigner do not support the assignment of values and therefore `M0 := CMP( D0 , D10 , M0 );` is an invalid statement.
- As long as the structure of the applied instruction is unchanged, starting new lines is allowed.

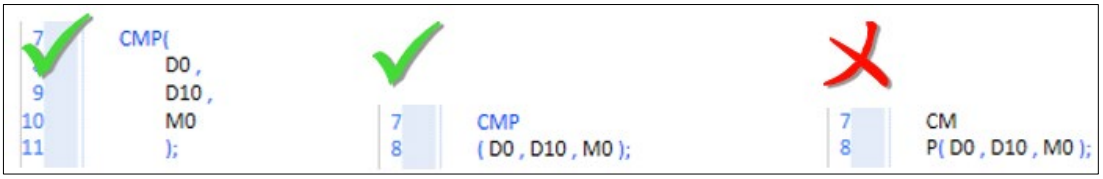


Figure 8 - 224: Applied Instruction Structure

Examples:

Example 1: The instruction **CMP** is used with an **IF** statement.

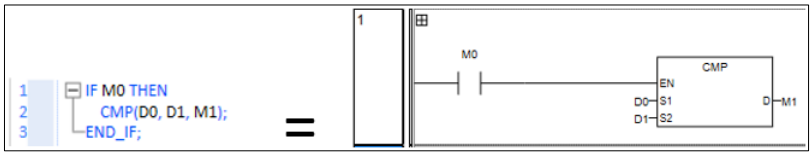


Figure 8 - 225: Applied Instruction Structure - Example 1

Example 2: The state of **EN** pin need not 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 4 in the following figure is a statement about resetting the timer.

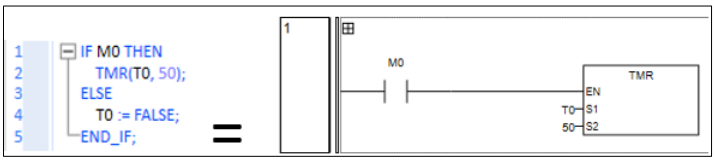


Figure 8 - 226: Applied Instruction Structure - Example 2

Function Block structure:

Format:

```
Name of FB instance (Input pin 1 := Input operand 1,
                     Input pin 2 := Input operand 2,
                     …
                     Output pin 1 => Output operand 1,
                     Output pin 2 => Output operand 2,
                     …
                     Output pin n => Output operand n  );
```

Figure 8 - 227: ST Function Block Structure - format

Description:

The Function Block (FB) is executed.

Rules and Limits:

The state of the EN pin of the **Function Block** instance need not be specified. When the statement is executed, the FB instance 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 FB is a statement and therefore it ends with a semicolon.

The input/output pins of the FB and the corresponding operands must be put in parenthesis. A pin and the corresponding operand form a group. The groups are separated by commas. The datatypes of the operands are specified in the FB definition.

The FB before a parenthesis is a FB instance rather than a FB definition, that is, the variable before the parenthesis is a variable whose datatype is a function block.

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 FB. The arrangement cannot be changed. The **EN** pin and the **ENO** pin are executed. The input pins are listed above all the output pins. A pin of the VAR_IN_OUT class is an input pin. Refer the ladder diagram and the structure text in the following example for more information.

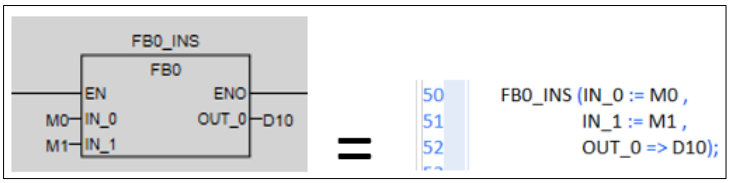


Figure 8 - 228: Function Block Structure - Example

Examples:

Example 1: When a statement which calls a function block is executed, the function block instance is executed.

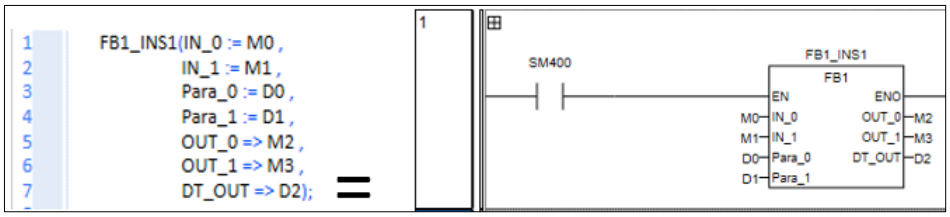


Figure 8 - 229: ST Function Block Structure - Example 1

Example 2: A function block is used with an IF statement.

Figure 8 - 230: ST Function Block Structure - Example 2

Example 3: The value of the variable CH determines which function block is executed.

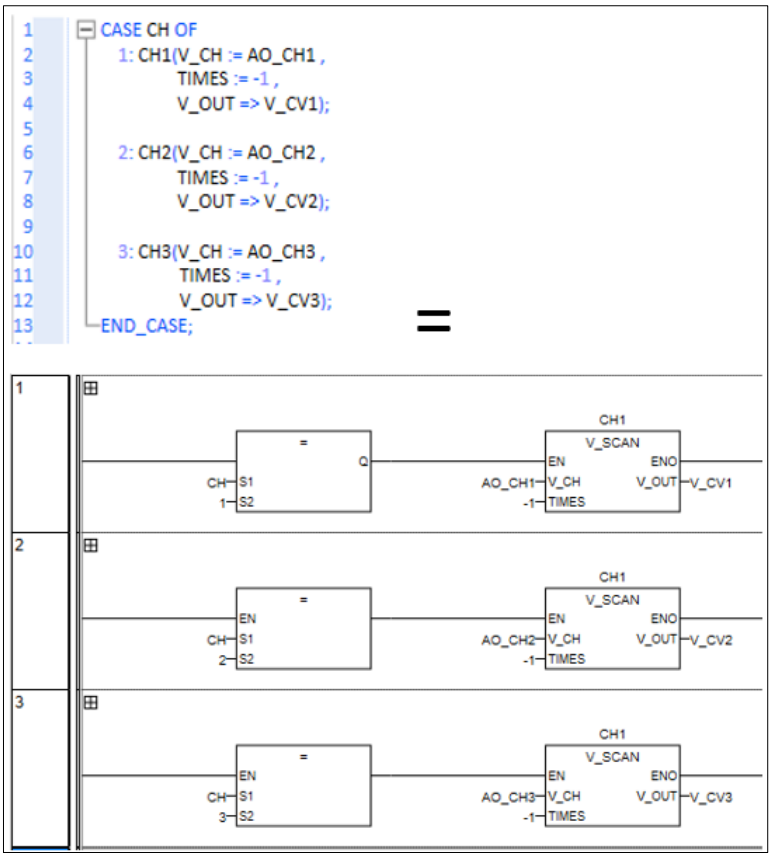


Figure 8 - 231: ST Function Block Structure - Example 3

8.3.4 C Language Editor (C Language)

The characteristic of C language is that the entire program is constructed by many text descriptions. Compared with the instruction list (IL), the C language is more readable and can express a grammar similar to mathematical equations. Complex arithmetic, so for a program that needs to perform complex calculations, C language is a more suitable development language for such applications.

NOTE: *Before downloading the C language program to the host, you must confirm whether the AS host firmware is V1.06.00 (inclusive) or higher.*

8.3.4.1 C language Structure Description

This editing language is suitable for programmers who are already familiar with C language editing. If you want to start editing in C language, please refer to the following figure for the editing language structure supported by C language; user can first use the common DEV-C and TurboC on the market And editing software such as Visual Studio for editing, compiling and testing, and then using the import function provided by DIADesigner, import the completed C language ".C" and ".h" files and use them in C language (C) FB or C language (C) POU, as shown in the figure below, can also use C language (C) FB in structured language (ST) POU or ladder diagram (LD) POU.

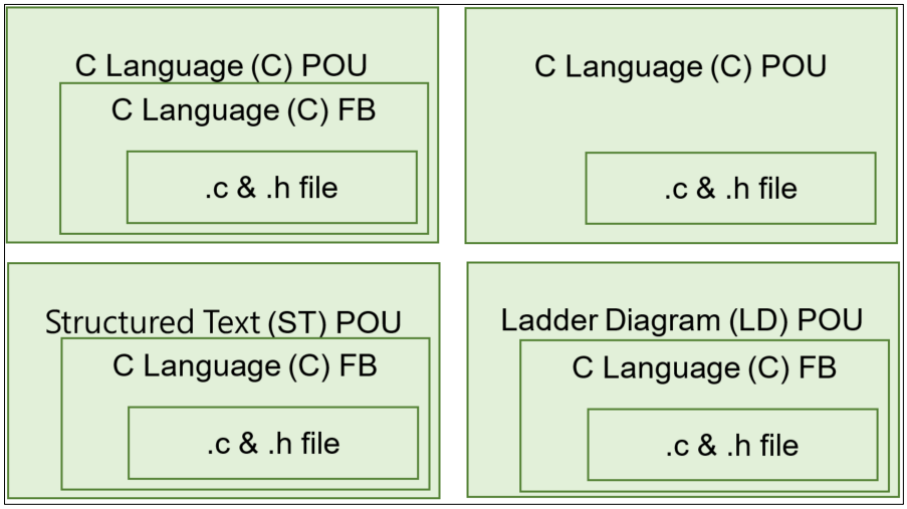


Figure 8 - 232: C language architecture

8.3.4.2 C language editing environment description

The C Code execution code storage space (ROM) provided by AS series hosts is 64K Bytes, and the variable declaration available space (RAM) is 8K Bytes. The above two storage spaces are independent of the existing PLC program space. After each compilation, it is recommended to pay attention to the remaining space from the compilation message box.

When the user creates a program or function block POU, the C programming language can be selected in the Language field of the Add Program window.

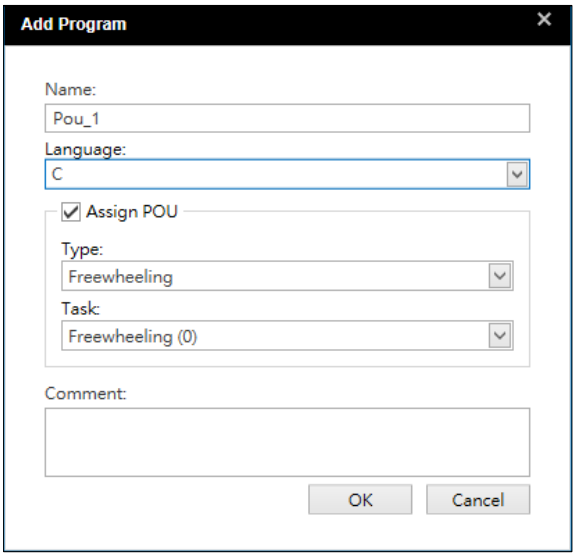


Figure 8 - 233: Add C POU

The C POU editing environment is shown in the figure below, the editor contains one part:

- POU working area

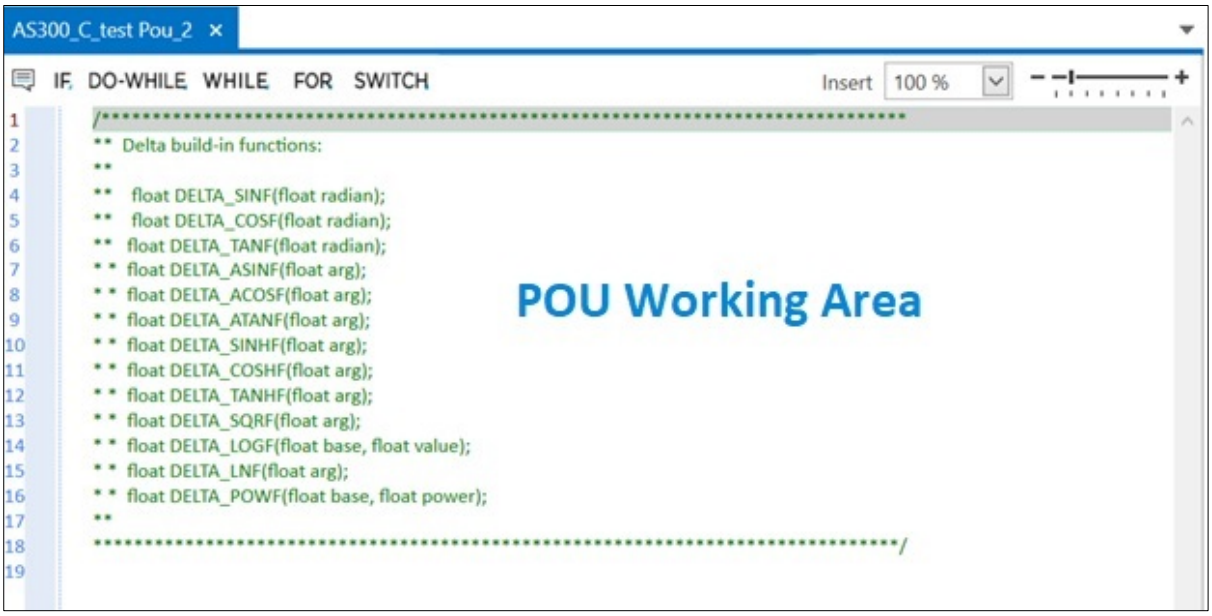


Figure 8 - 234: C POU Program editor

C working column

Open the C POU program editor, the task bar is shown in the figure below.

Figure 8 - 235: C POU program editor taskbar

The description of each function item in the C working column is as follows:

Toolbar icon	Function Description
IF	Conditional structure IF
DO-WHILE	Loop structure DO-WHILE
WHILE	Loop structure WHILE
FOR	Loop structure FOR
SWITCH	Condition Structure SWITCH CASE
	Add / delete line comment

Set the start number and C_Tag_Table of the data exchange area

When declaring variables, you can use matrix (Array) and other methods, but it must be noted that the memory that can be declared for the variables is limited, so it cannot be used beyond the scope, otherwise it may seriously cause the PLC to crash. You must add variable symbols in the C language main symbol table of global symbols. There are 1000 D devices in the device addresses that can be used by C language symbols. You can set the starting address range by yourself. In "C_Tag_Table", click the mouse Right-click and then click "Set the start number of the data exchange area" in the shortcut menu to set the start position of the data exchange area. The start address can be filled in from D0~D29000, and the system will automatically set 1000 consecutive numbers in the back D device.

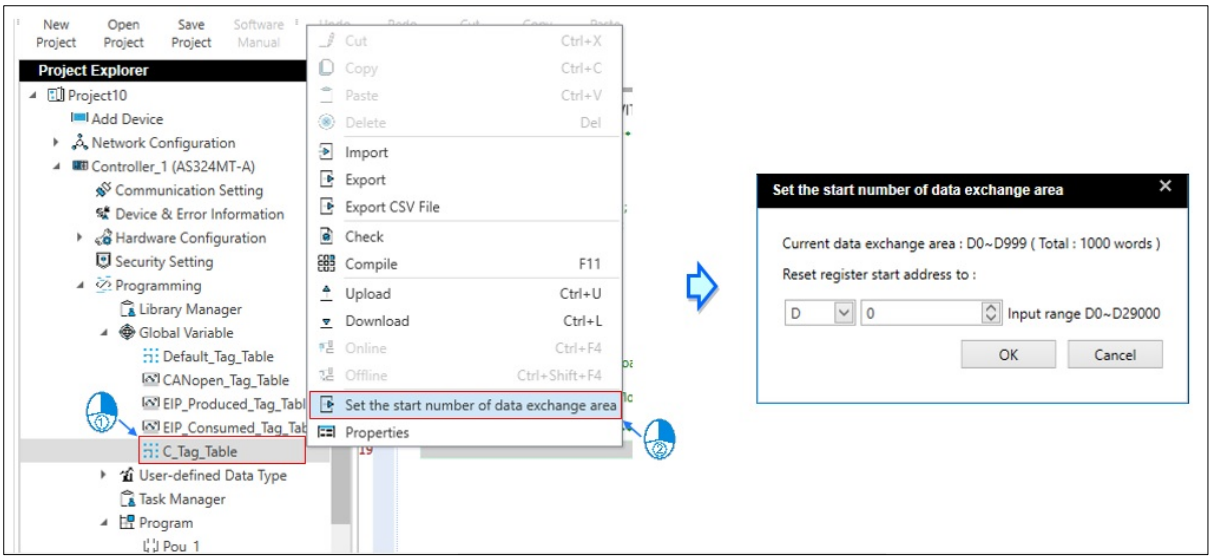


Figure 8 - 236: C_Tag_Table sets the starting number of the data exchange area

After setting D0 as the starting position, double-click "C_Tag_Table" and the content of C_Tag_Table will appear.

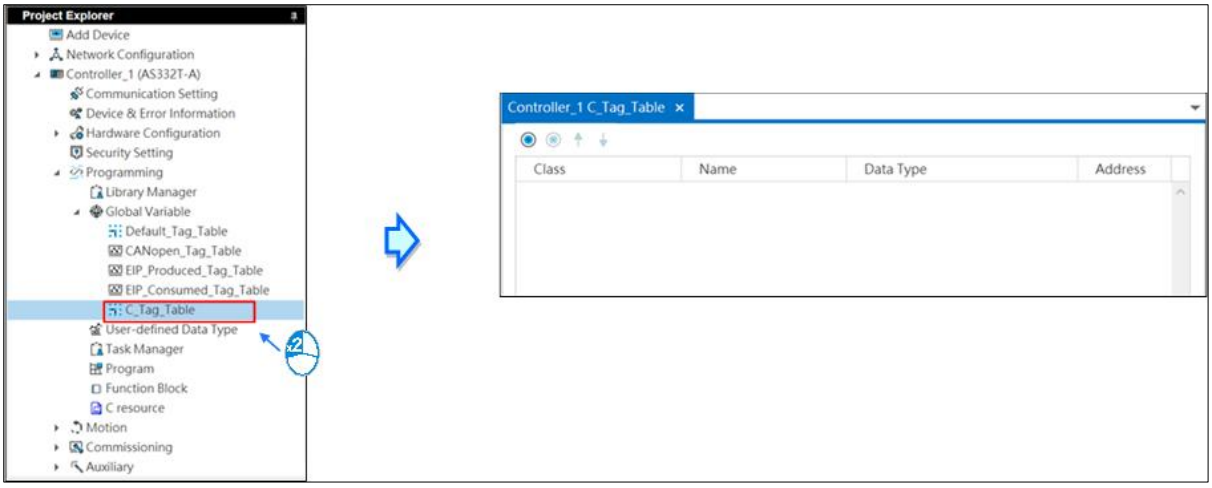


Figure 8 - 237: Open C_Tag_Table

In the blank space of C_Tag_Table, right-click with the mouse, and then click "Insert" in the shortcut menu, and the default variables will appear for editing.

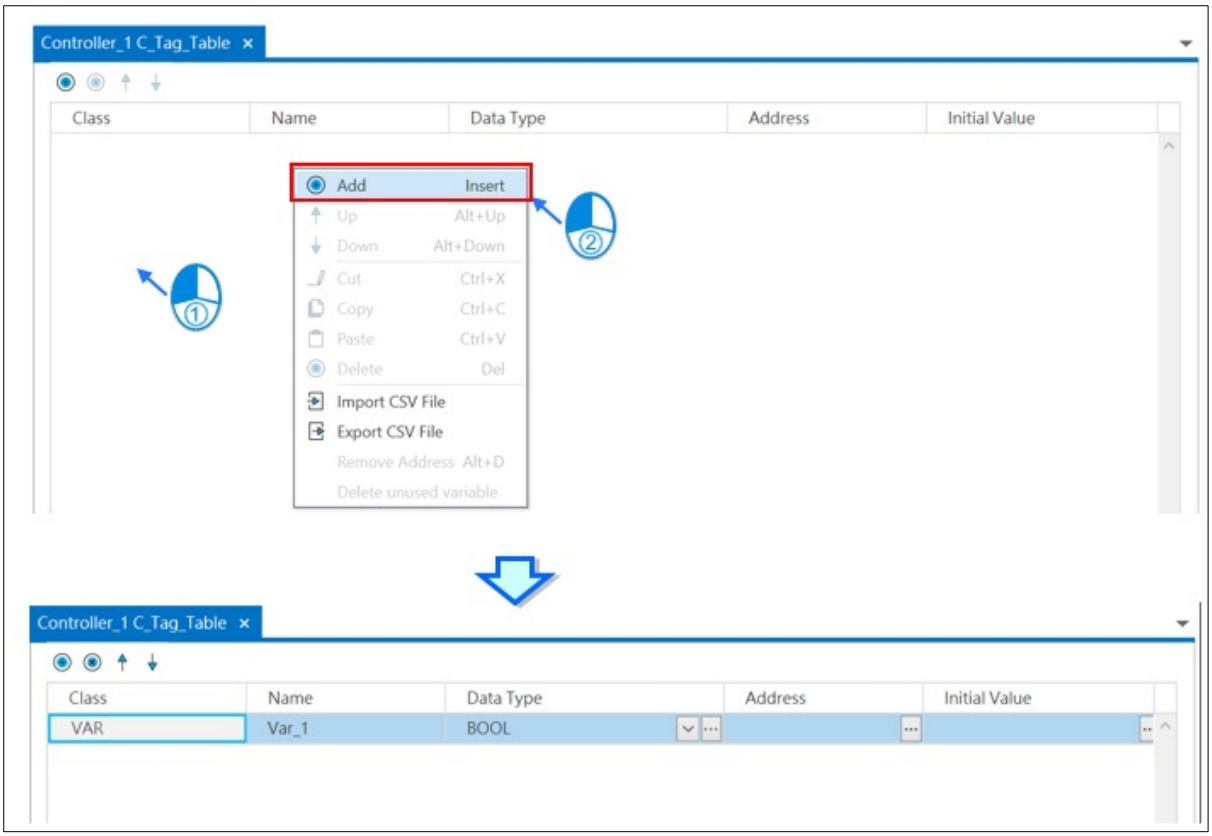


Figure 8 - 238: C_Tag_Table new variable

C_Tag_Table only supports four types: BOOL, WORD, DWORD and REAL. The starting address of the address is the range set by the C language data exchange area. The example in the figure below shows the D0.0 address setting for the newly added variable. The address field of the variable can only be filled in the range set by the C data exchange area, and this variable can be used directly in the C language program, but also equivalent to a global variable that can be used in other programming languages.

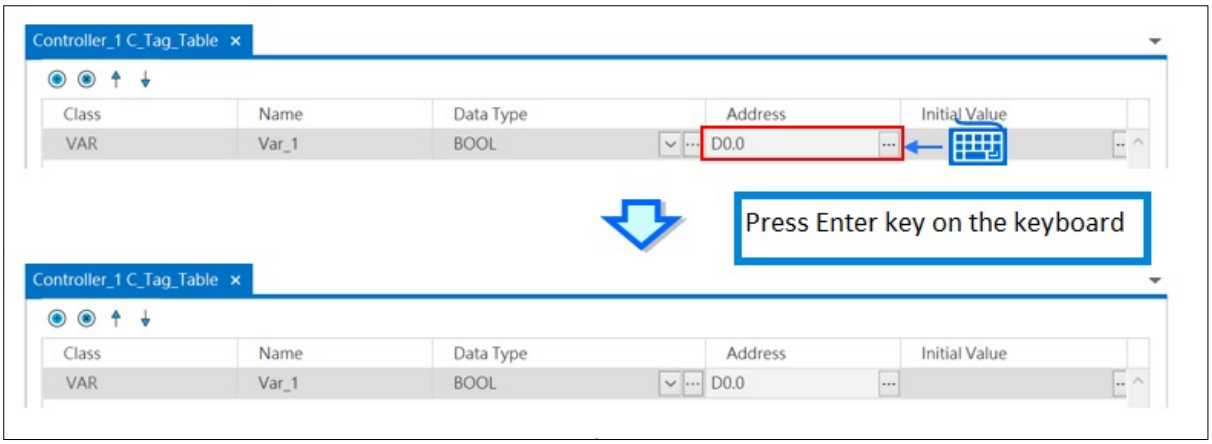


Figure 8 - 239: C_Tag_Table input variable address

You can also input undeclared variable symbols in the C language editing area. After inputting, double-click the text with the mouse and press the right button, and then click "Add Variable" in the shortcut menu to automatically export the variable symbols After finishing the setting, the corresponding setting variable will appear in C_Tag_Table.

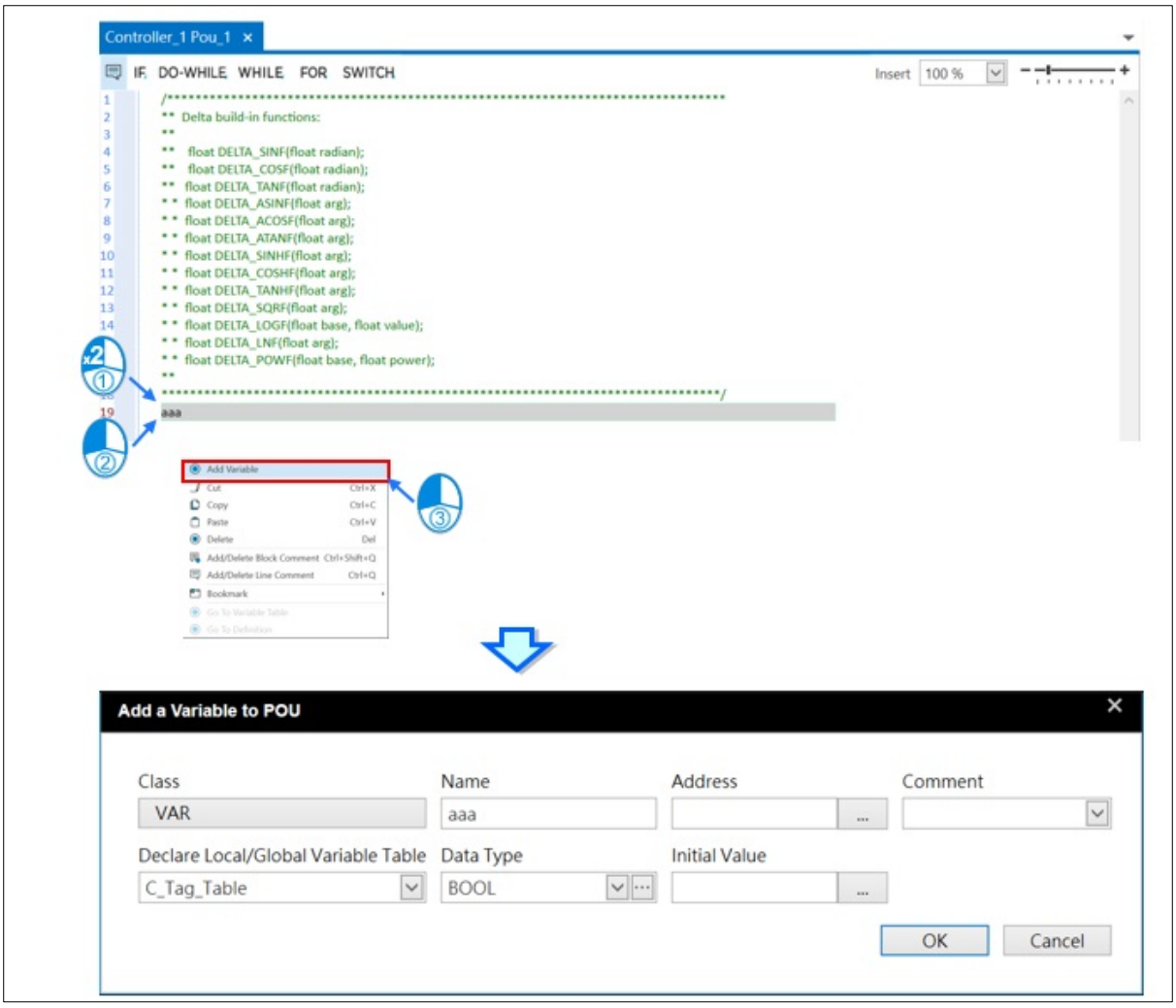


Figure 8 - 240: Add variable to POU

In the process of program input, the matching variable symbols will be automatically filtered from the created variable symbols for user to choose. You can use the " ↑ " and " ↓ " keys on the keyboard, or directly click the left mouse button in the menu to complete enter.

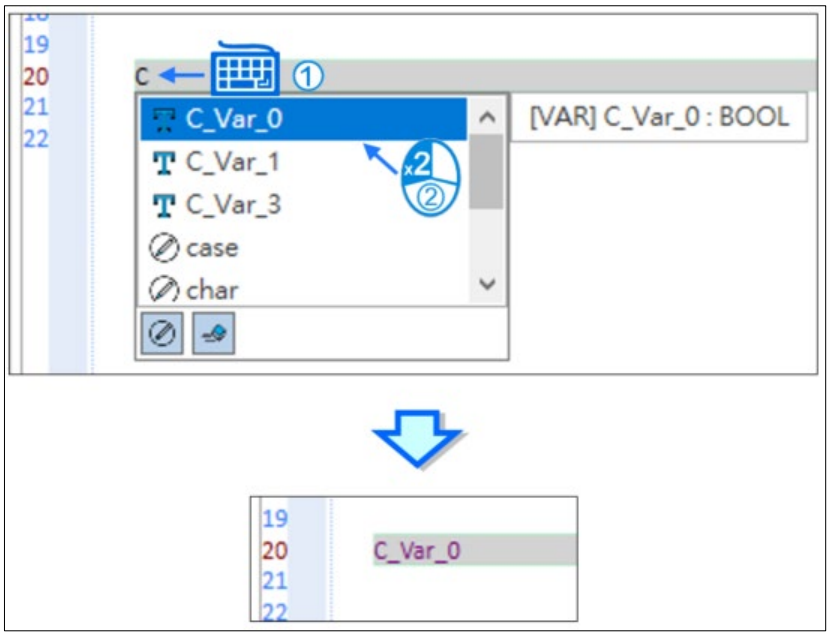


Figure 8 - 241: Filter out the matching variable symbols from the variable symbols

Edit C language

In the C editing workspace, only standard C language writing is supported, other high-level syntax such as PLC instructions, C++ or Visual C... are not supported. The operation method is similar to that of a general text editor. You only need to directly use the keyboard to input or modify text on the work area according to the format of C grammar when editing, and you can execute line break when pressing the keyboard [Enter] key action.

Figure 8 - 242: C language wrap action

In the process of program input, the matching variable symbols will be automatically filtered from the created variable symbols for user to choose. You can use the " ↑ "

and " ↓ " keys on the keyboard, or directly click the left mouse button in the menu to complete enter.

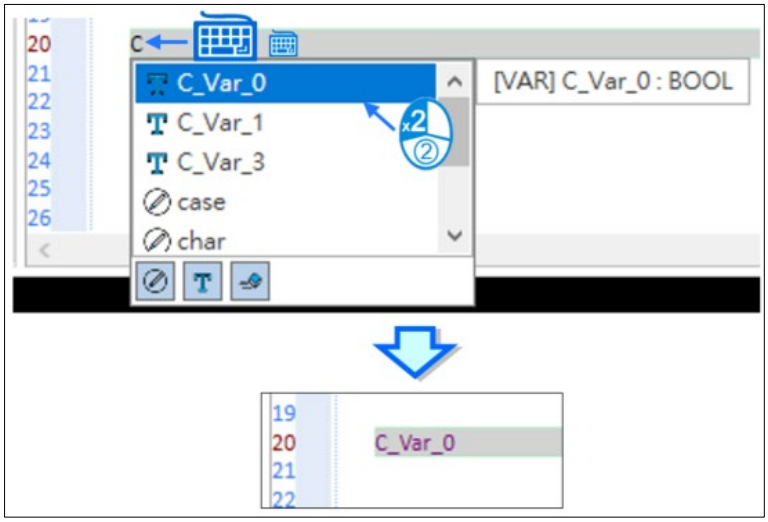


Figure 8 - 243: Variable symbols can be automatically filtered out

Line Comment

After the C language program is created, if the user wants to add explanatory text, or part of the description of the C language program does not need to be compiled, the line annotation function can be used at this time to eliminate the description that does not need to be compiled. The keyword for line comments is //, the string after this keyword turns green, and the green string represents the comment string that the compiler will ignore.

The following are the steps for adding/deleting line comments:

1. Select one or more lines to be executed.
2. Click the pattern button on the toolbar,
   Or
   Click the right mouse button in the editing tool area to open the right-click menu, and then click the left mouse button to add/delete line comments.

Or

Click CTRL + Q on the keyboard at the same time.

Result: The selected line becomes line comment text (green string).

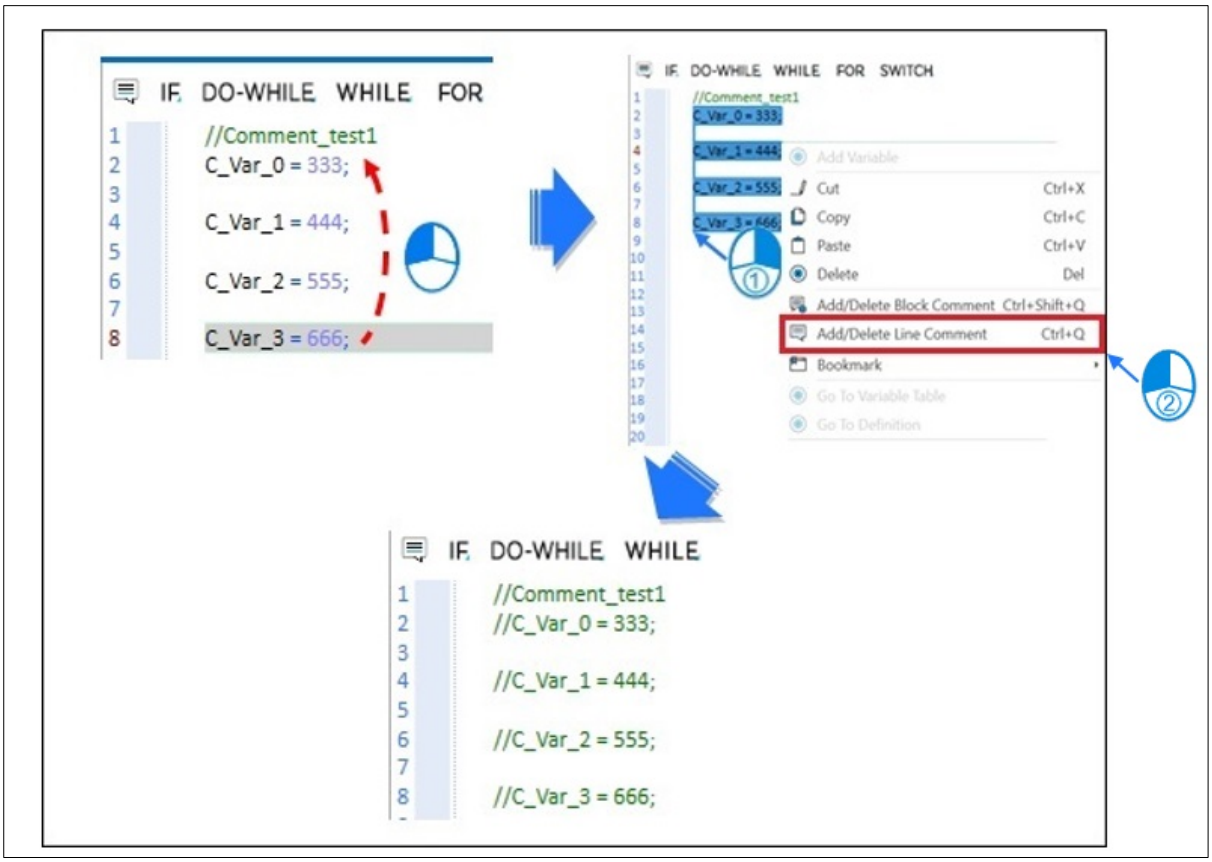


Figure 8 - 244: Add line comment

3. Repeat the above steps to delete the line comment keywords.
   Result: Delete the line annotation keywords, make the C language program effective.

Block comment

Block comments provide another way of commenting besides line comments. The keywords of the block comment are /* and */, which are placed before and after the string to be annotated, respectively. The text between these two keywords will turn

green, representing the comment string, which will be automatically ignored by the compiler during compilation Comment string.

The following are the steps for adding/deleting block comments:

1. Select the string to be annotated
2. In the editing tool area, click the right mouse button to open the right-click menu, and then click the left mouse button to add/delete block comments.
   Or
   Click CTRL + SHIFT + Q on the keyboard at the same time.
   Result: The selected string becomes block comment text (green string).

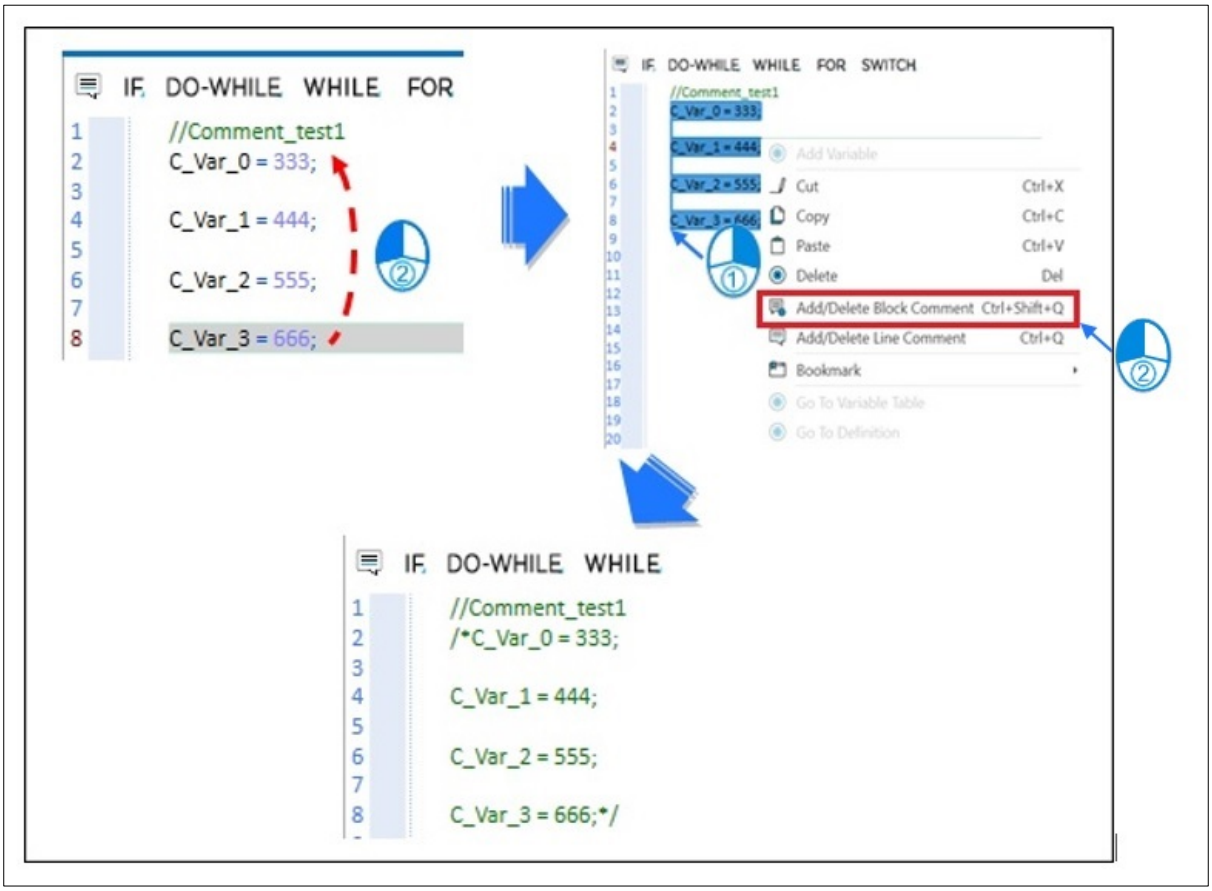


Figure 8 - 245: Add block comment

3. Repeat the above steps to delete the block comment keywords.
   Result: Delete the block comment keyword, make the C language program description become effective.

Function block (FB) call structure

The editing method of C function block is the same as that of C program. According to the format of C syntax, you can directly use the keyboard to input or modify text on the work area.

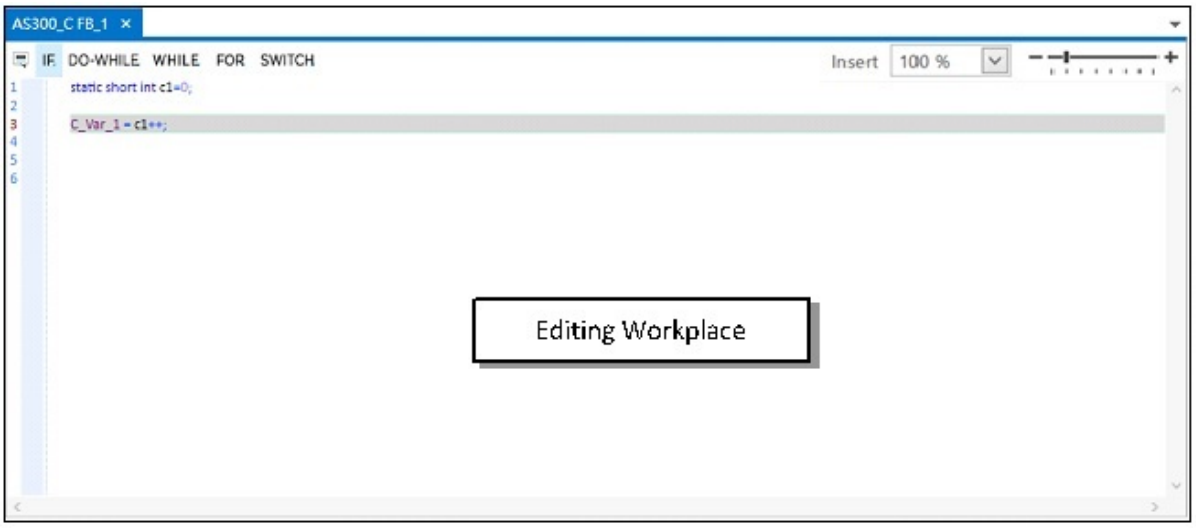


Figure 8 - 246: C function block editing workspace

There are two ways to insert C language function block into LD language program.

Method 1

According to the function block name, you can directly use the keyboard to input in the editing work area of LD language.

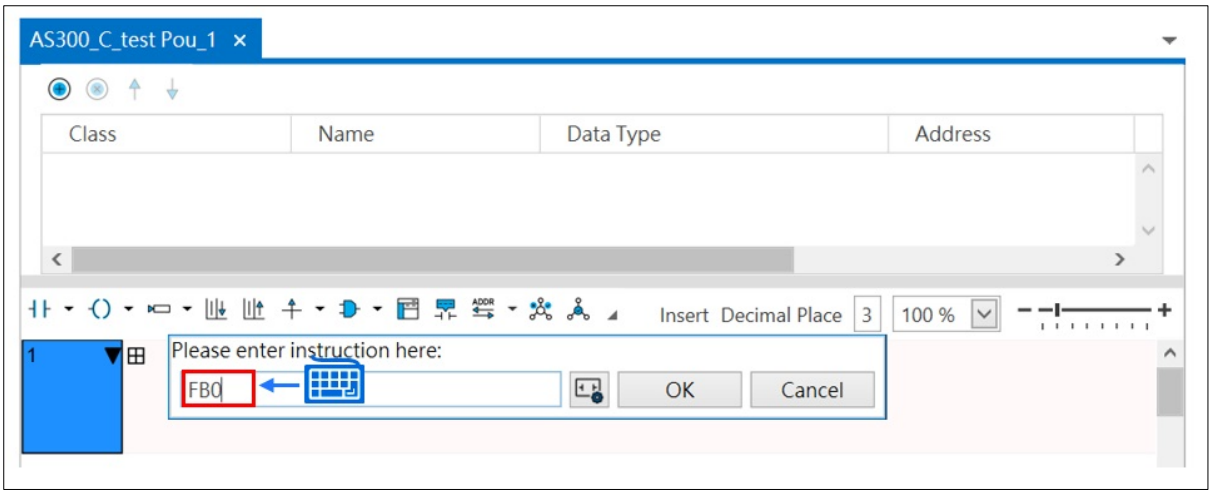


Figure 8 - 247: Input C language function block into LD language program

Method 2

Expand the list of function blocks in the project tree, find the item you want to insert in it, press and hold the left mouse button on the item, then drag it to the position you want to insert and release the left button.

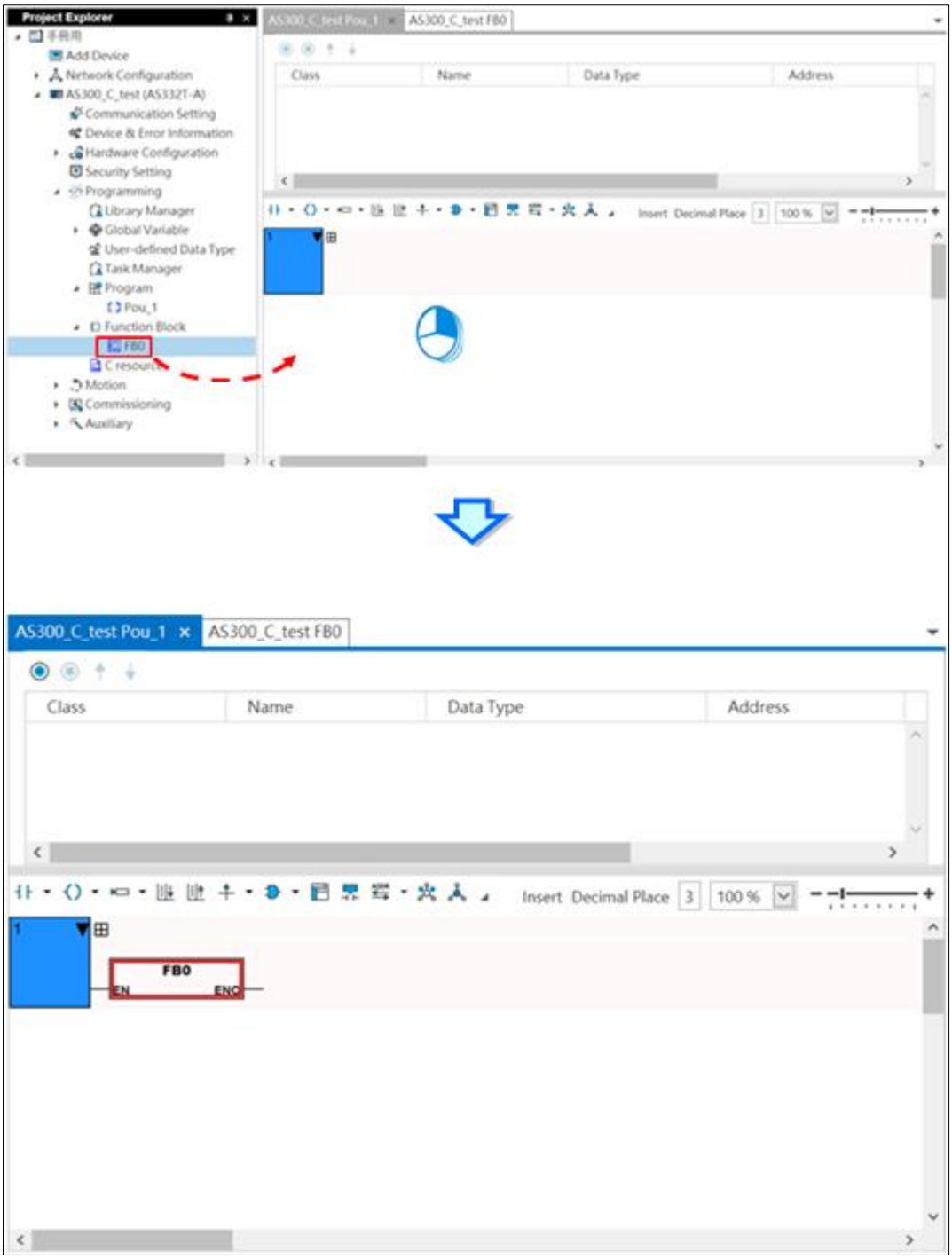


Figure 8 - 248: Drag C language function block into LD language program

Import/Export C resource file content

To import the completed C language ".c" and ".h" files, first click the right mouse button on the "C resource file", then click the "Import" item in the shortcut menu, and

then select the file to be imported Add the C file window and select the C file to be imported, as shown in the figure below. To import .H files, you need to select C headers (*.h) in the "File Type" item.

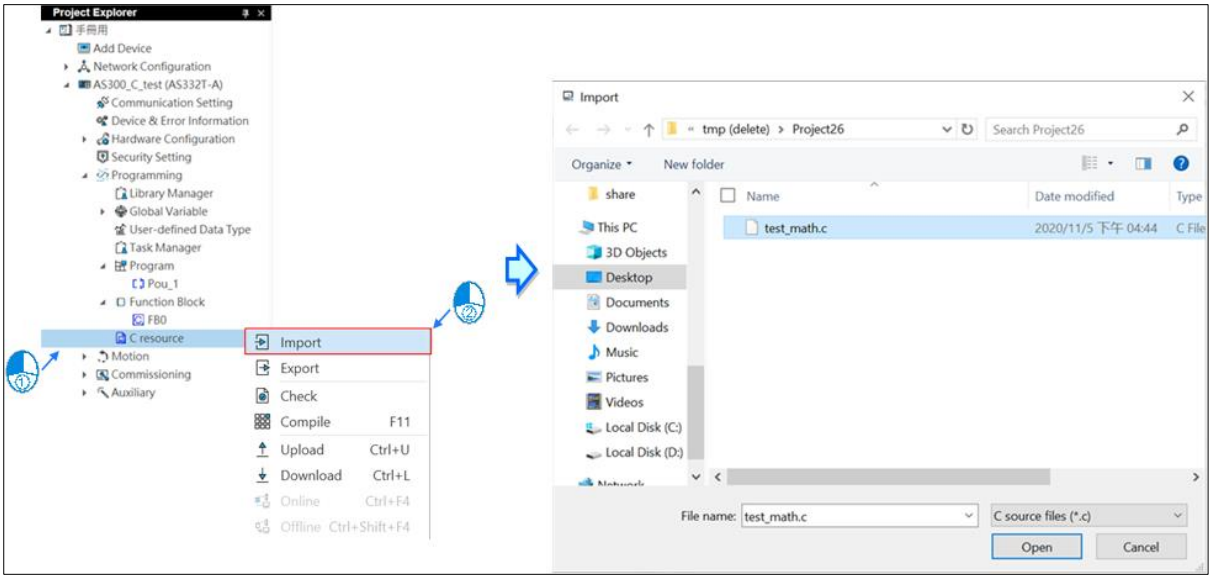


Figure 8 - 249: Import.C file

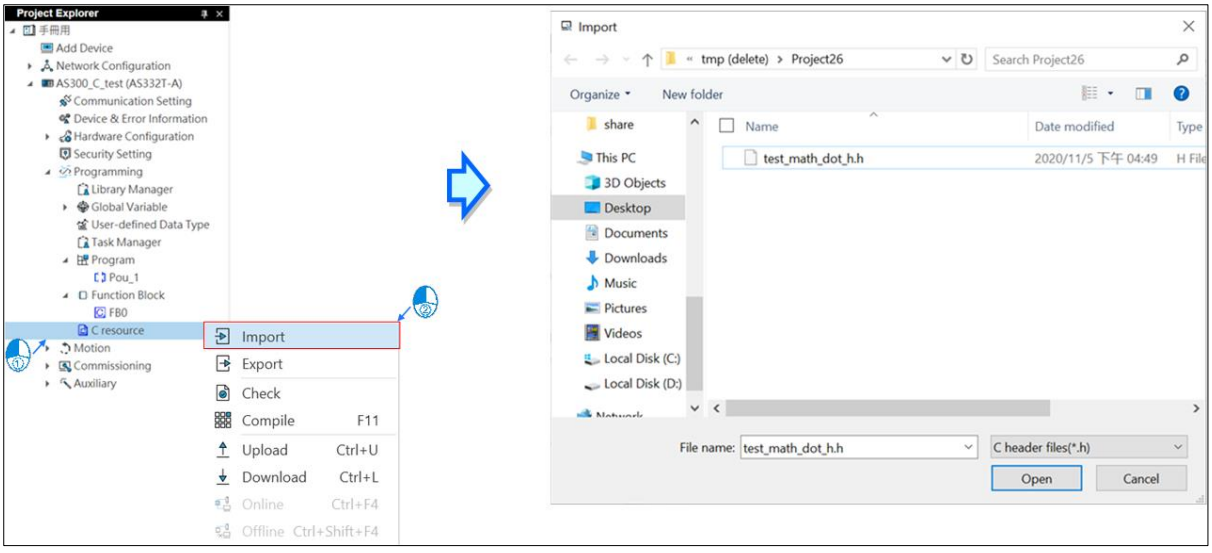


Figure 8 - 250: Import.h file

The user can also choose to display both the ".c" and ".h" files. At this time, the ".c" and ".h" files can be double-selected for batch import.

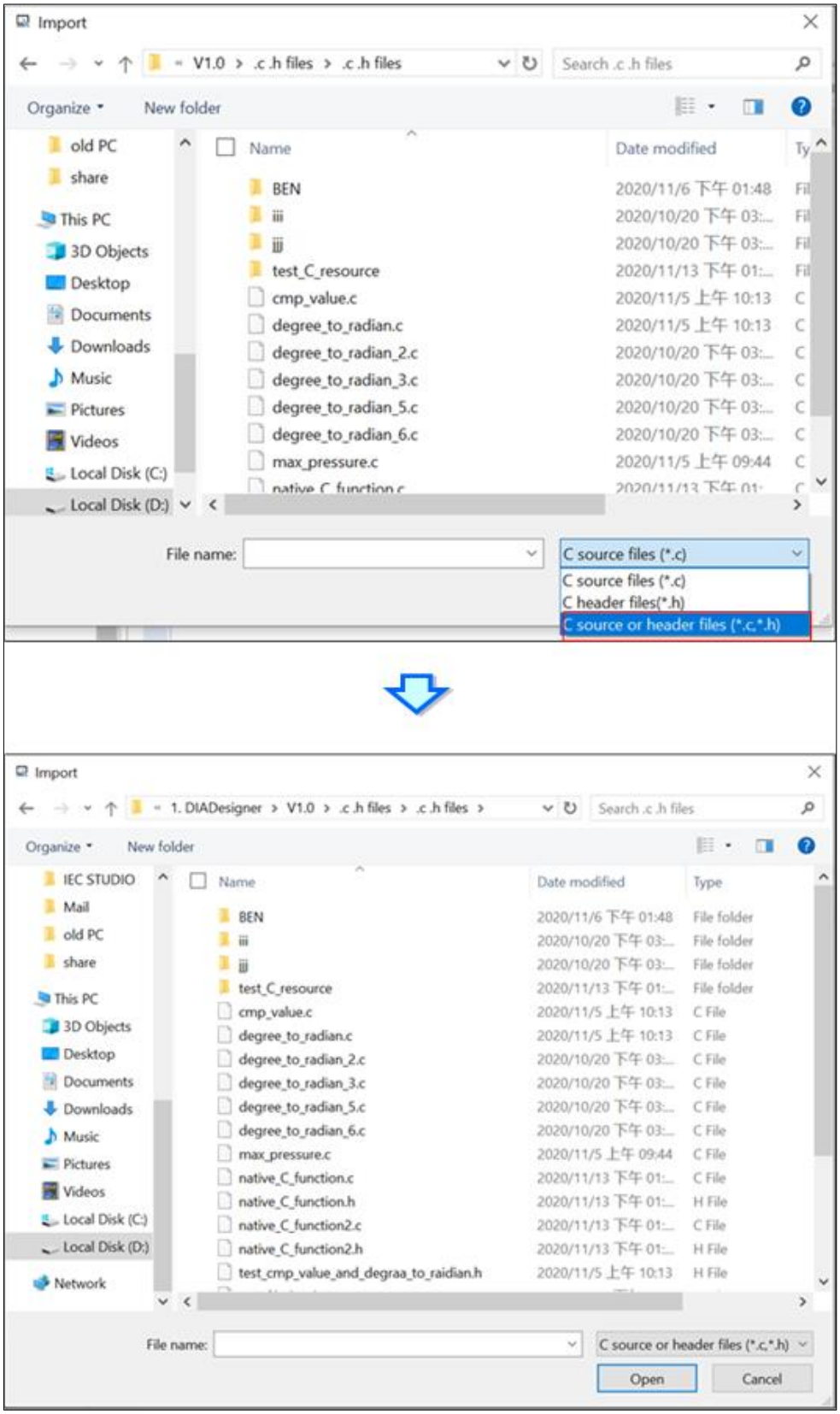


Figure 8 - 251: Import .c/.h file

Similarly, user can also click the "Export" item in the C resource file to do the export, or directly click the "Export" item in the .c/.h file to do the export.

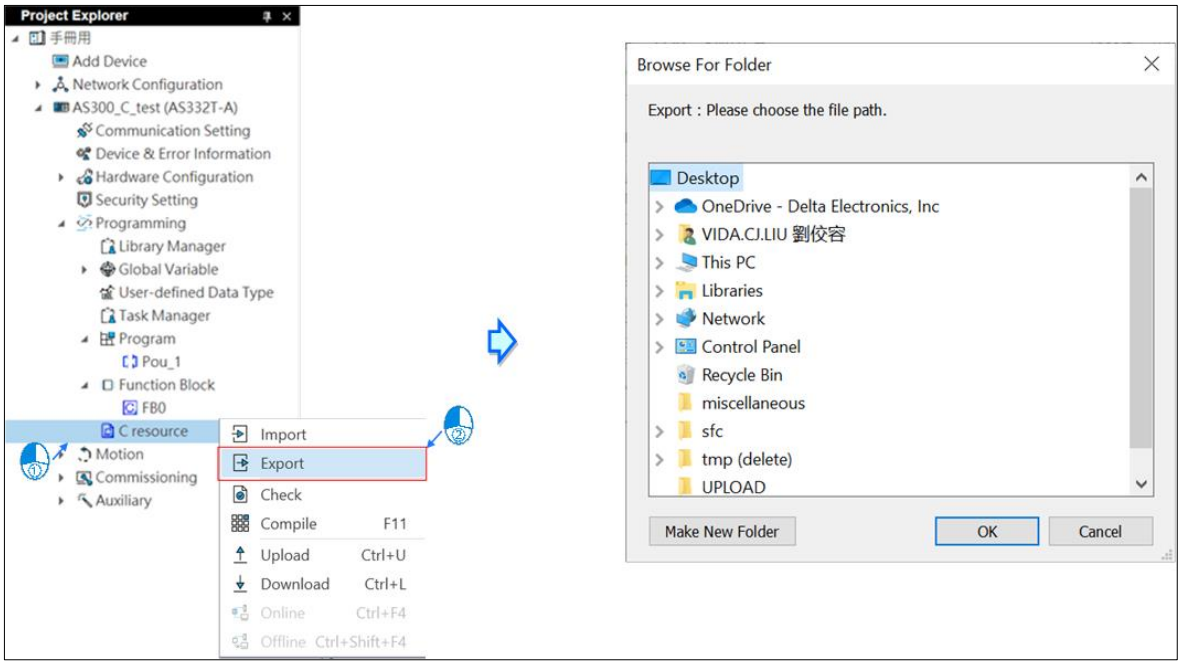


Figure 8 - 252: Export.C fie

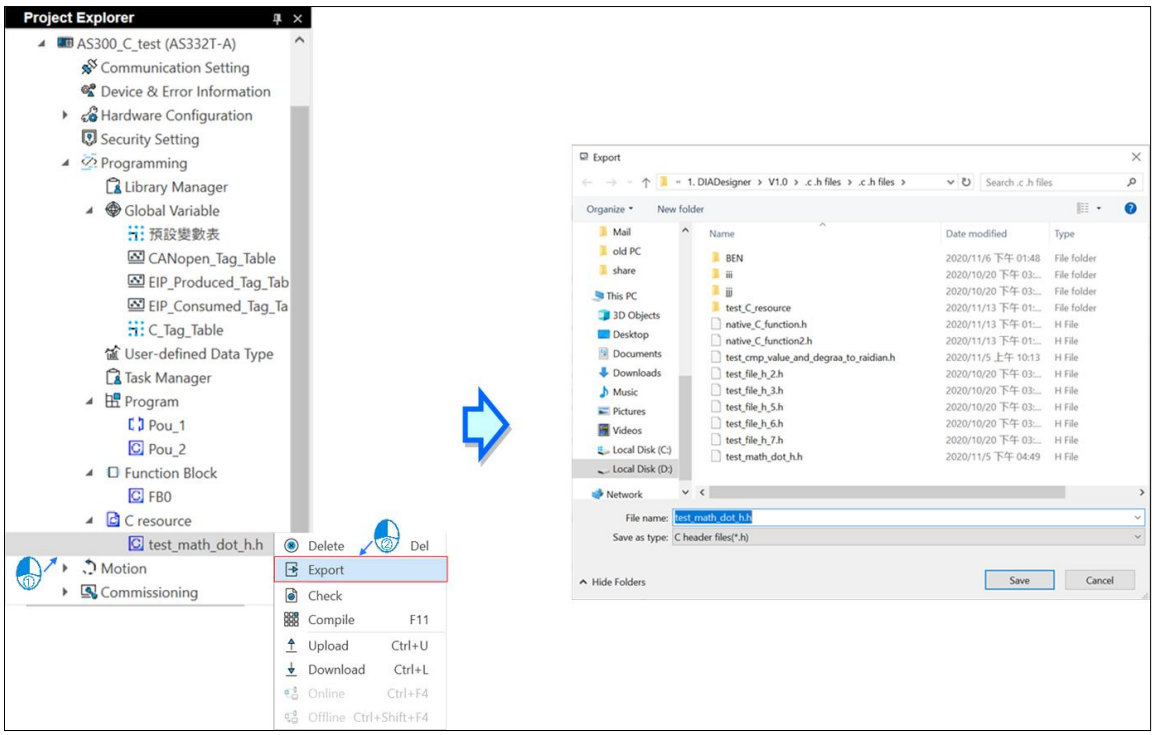


Figure 8 - 253: Export.h file

8.3.4.3 C Language compilation and debugging

DIADesigner provides PLC built-in trigonometric function and exponential library, which can be used by user without announcing, but it does not provide other external standard C library (such as math.h) on the market to save PLC internal provision C code storage space. DIADesigner will display the compiled error message. After double-clicking the error description, you can jump directly to the error location to modify the function. The user can monitor the existing PLC programs and function blocks online, but cannot perform line by line editing of the C language program. C Code for monitoring.

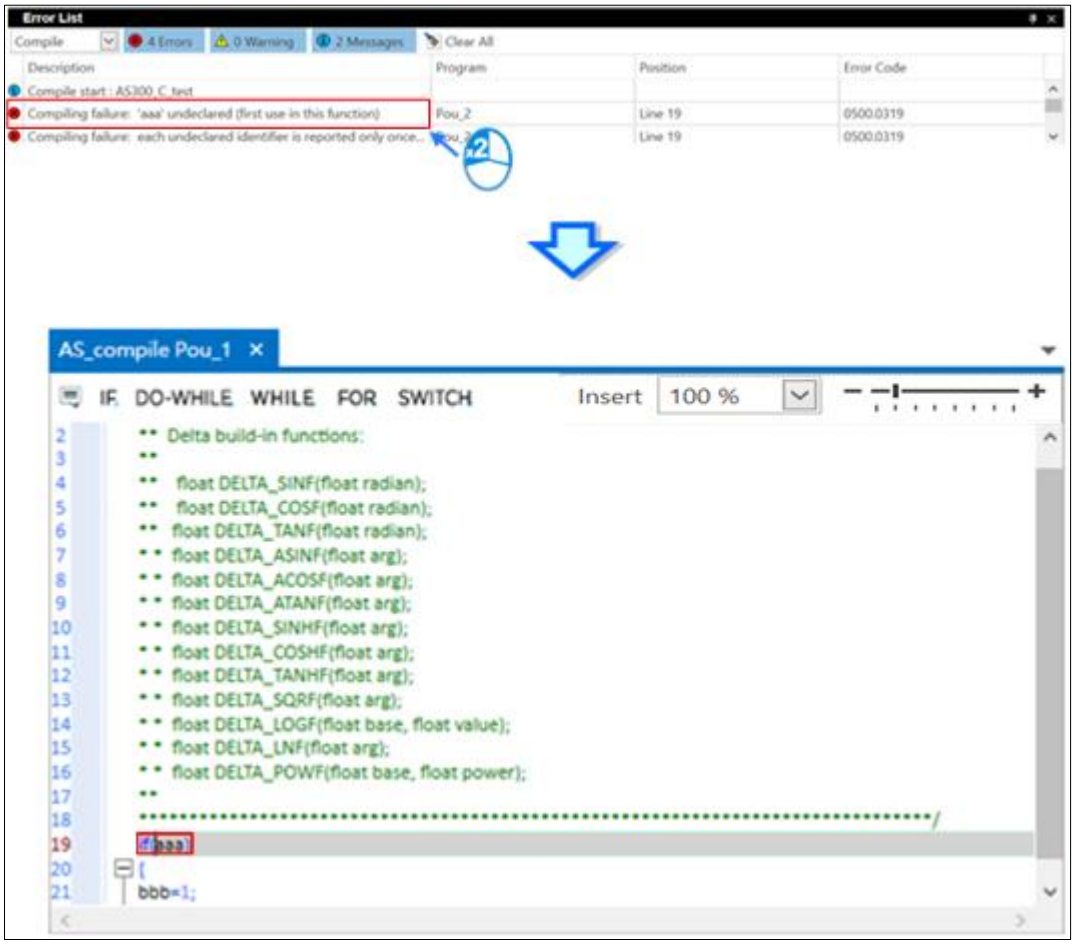


Figure 8 - 254: Double-click the error description to jump directly to the error location

8.3.4.4 Built-in library command description

The four arithmetic "add, subtract, multiply, and divide" commonly used integer type or floating-point number type are all built-in in DIADesigner and can be used without additional declaration.

The name of the built-in trigonometric function:

C function name	DIADesigner Edit name	Instruction Format	Function Description
sinf	DELTA_SINF	Float DELTA_SINF (float radian) ;	Sin operation
cosf	DELTA_COSF	Float DELTA_COSF (float radian) ;	Cos operation
tanf	DELTA_TANF	Float DELTA_TANF (float radian) ;	Tan opertion
asinf	DELTA_ASINF	Float DELTA_ASINF (float arg) ;	Sin^{-1} operation
acosf	DELTA_ACOSF	Float DELTA_ACOSF (float arg) ;	Cos^{-1} operation
atanf	DELTA_ATANF	Float DELTA_ATANF (float arg) ;	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 name of the built-in exponential function

C function name	DIADesigner Edit name	Instruction format	Function 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）;	Exponentiation

8.3.4.5 Example operation instructions

Example: Output a Sin function value every 10ms with each 5 degree transformation, and use the data oscilloscope function to record the output value of the Sin function.

Step 1: First create the symbol names of the variables shared by PLC and C language program, please refer to section 8.3.4.2, as shown in the figure below.

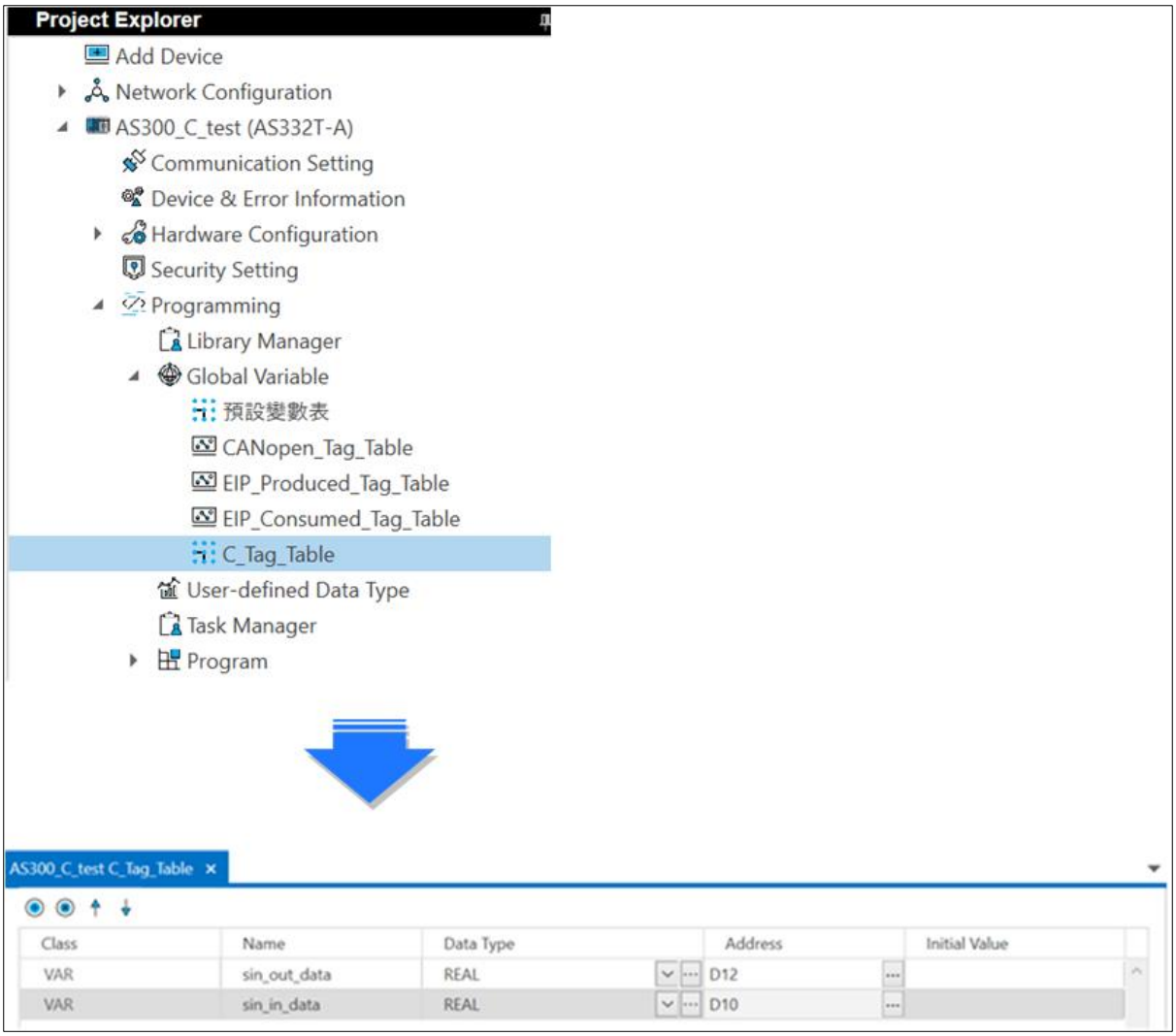


Figure 8 - 255: Create C_T_Table content

Step 2: Import a function degree_to_radian.c that you have edited and completed the test. Refer to Section 8.3.4.2. This step can be deleted according to the situation.

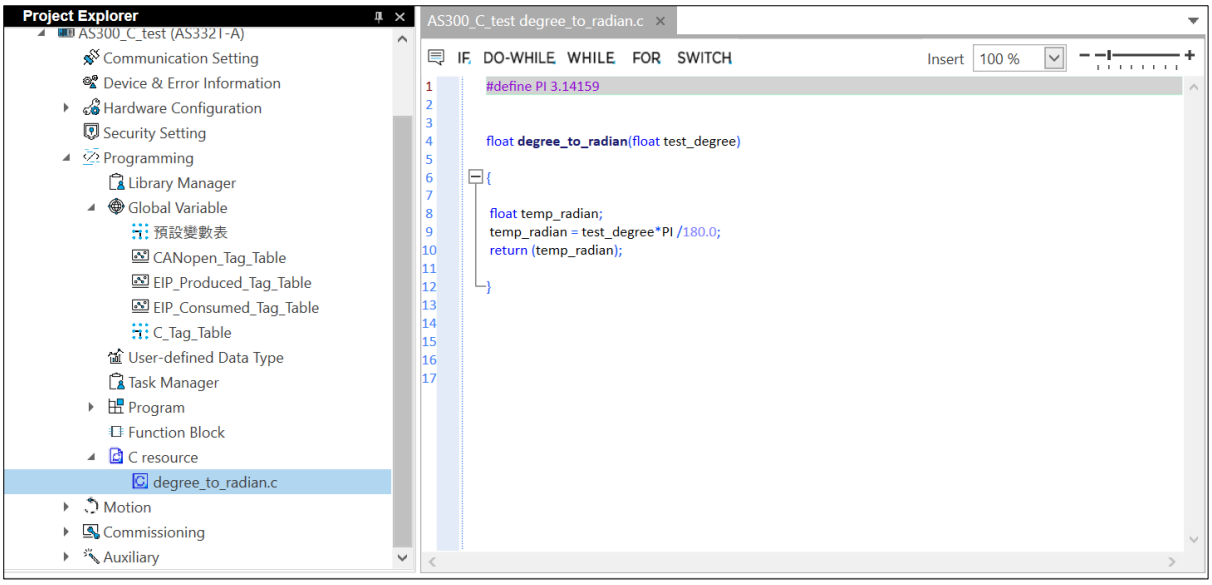


Figure 8 - 256: Import test function degree_to_radian.c

Step 3: Edit a new function block test_sin and write the calculation process, as shown in the figure below. The degree_to_radian function library.

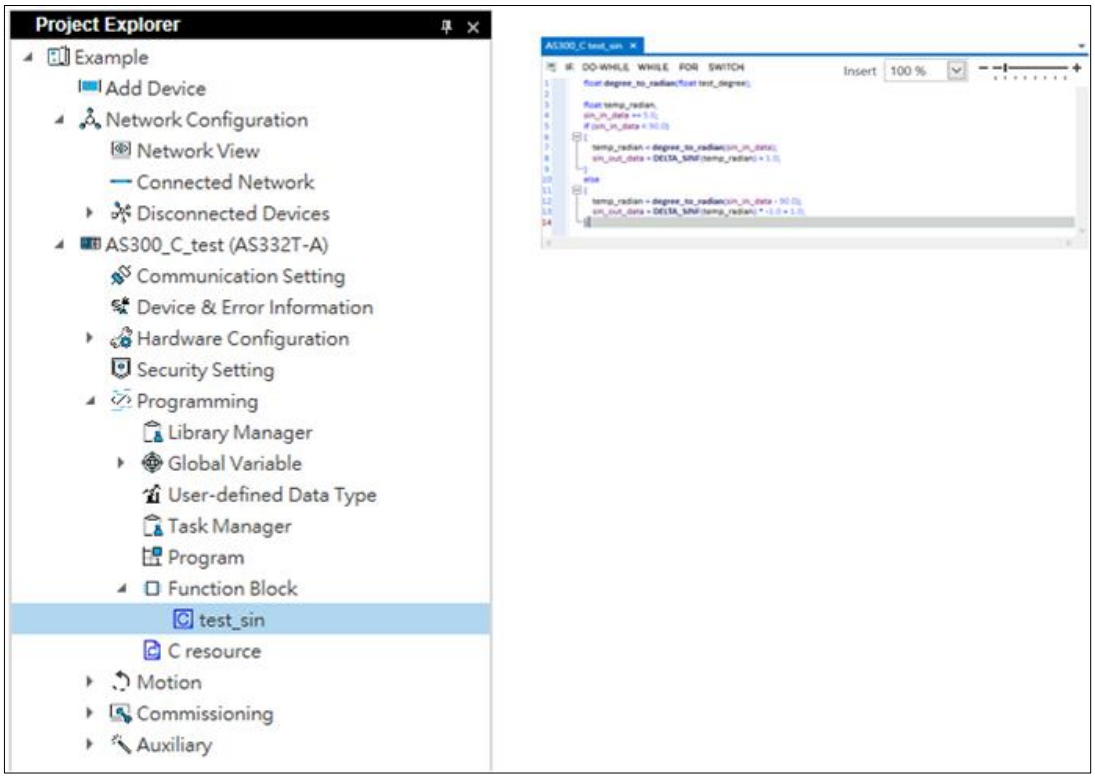


Figure 8 - 257: Edit new function block

Step 4: Create a program (POU), and write a function block (FB) that calls test_sin every 10ms, as shown in the figure below.

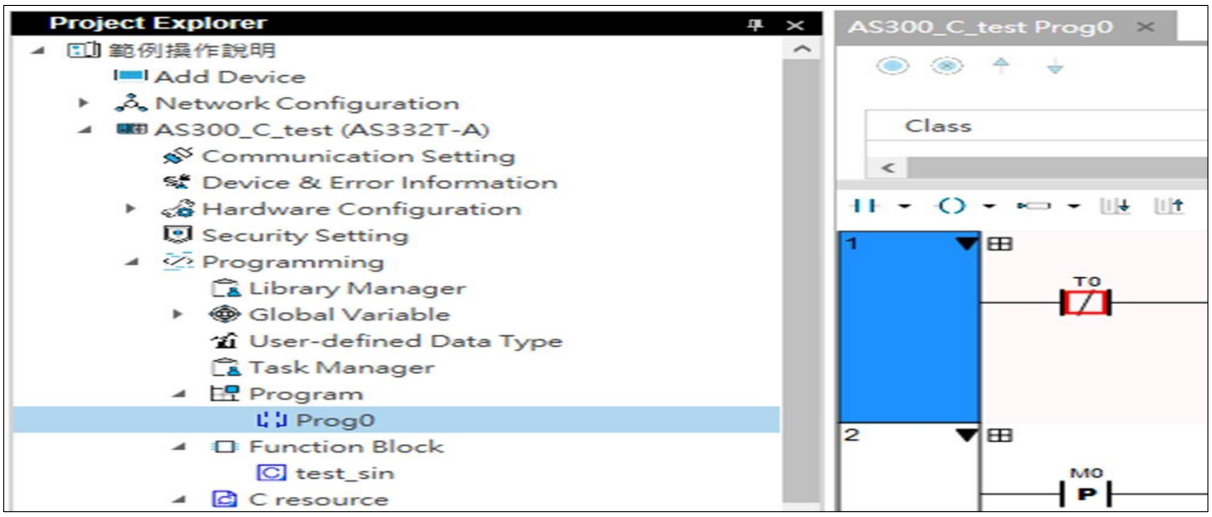


Figure 8 - 258: Create program (POU), call function block

Step 5: Press Compiler. Assuming no error occurs, the result is shown in the figure below.

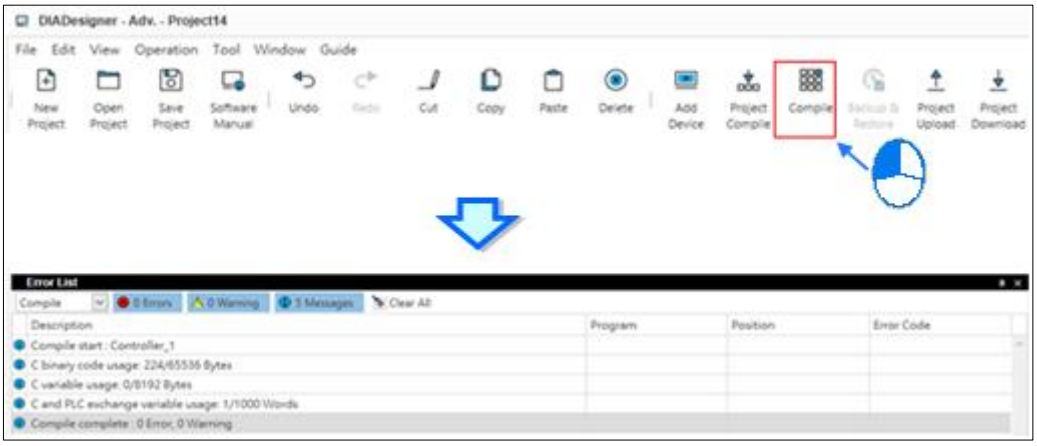


Figure 8 - 259: Compile PLC program

Step 6: After the compiling (Compiler) is successful, the communication is set and the PLC program is downloaded.

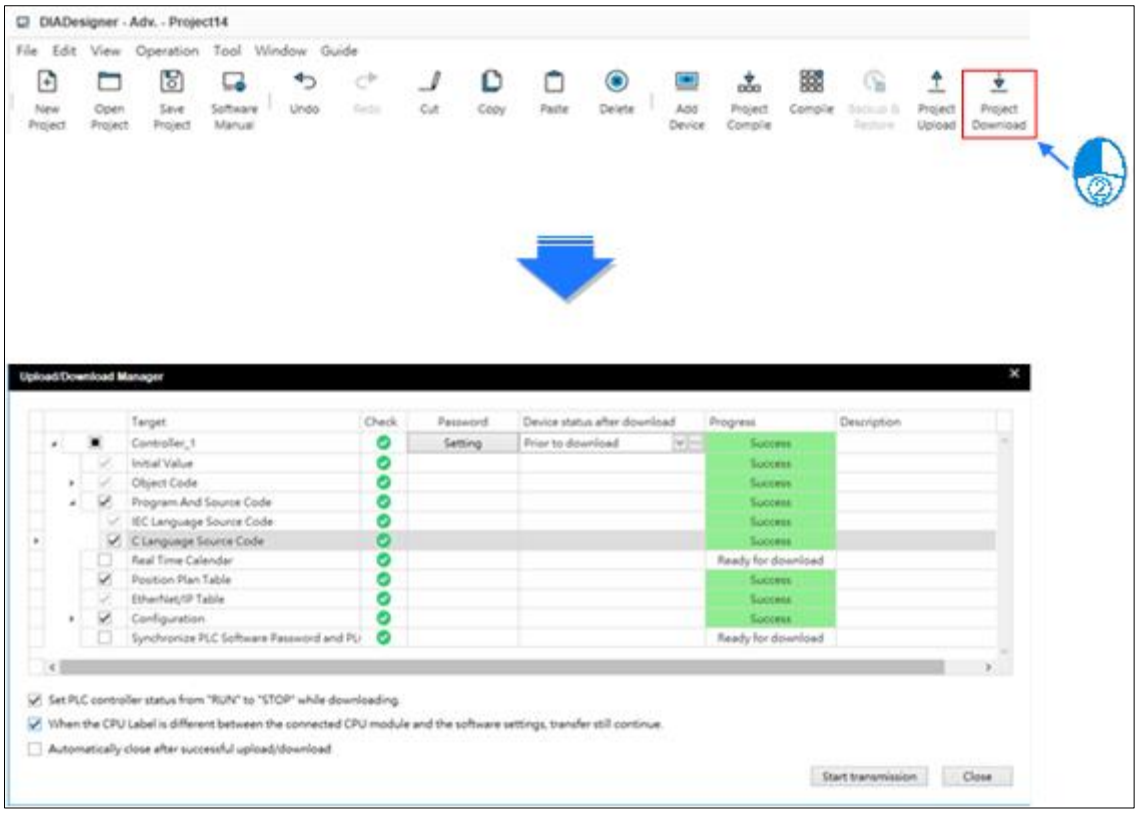


Figure 8 - 260: Download PLC program

Step 7: After starting PLC Run, turn on the data oscilloscope. The process is shown in the figure below.

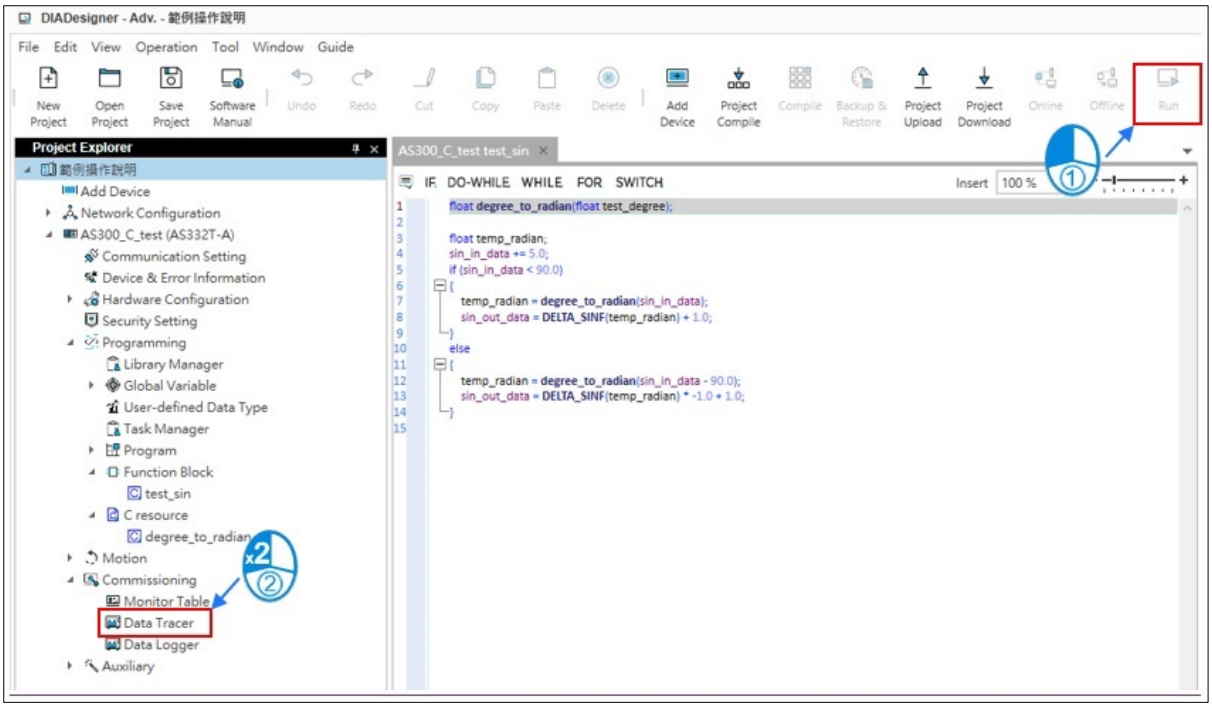


Figure 8 - 261: Open the data oscilloscope

Step 9: The data oscilloscope sets sampling parameters and measurement. The process is as shown in the figure below. Click "Add", double-click "Global Variables", click "C_Tag_Table" in the "Select Variables" window, and select the variables to be monitored. Then click "OK".

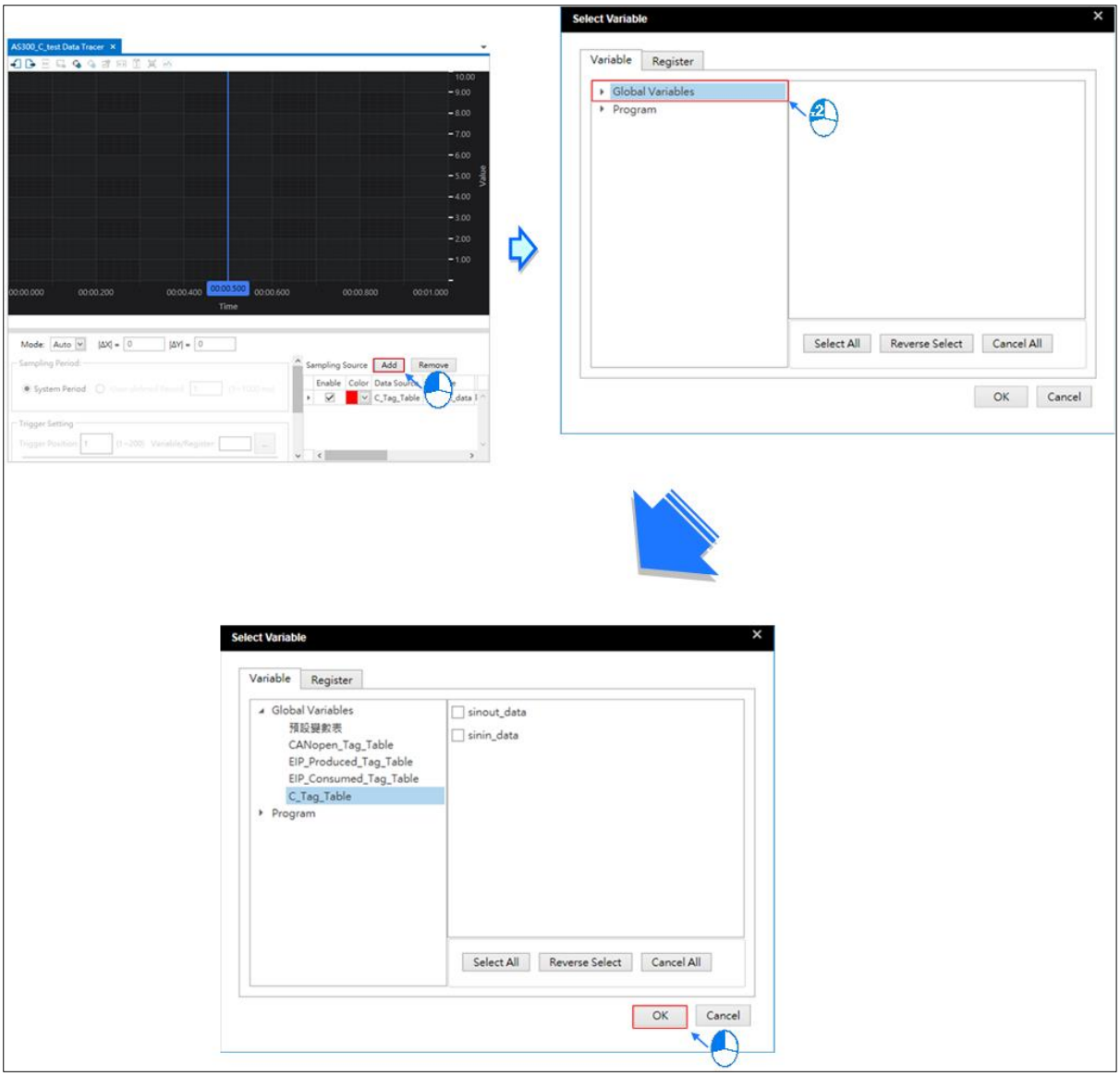


Figure 8 - 262: Data oscilloscope settings

Step 10: Click "Start Oscilloscope" to perform real-time monitoring of the data oscilloscope. The process is shown in the figure below.

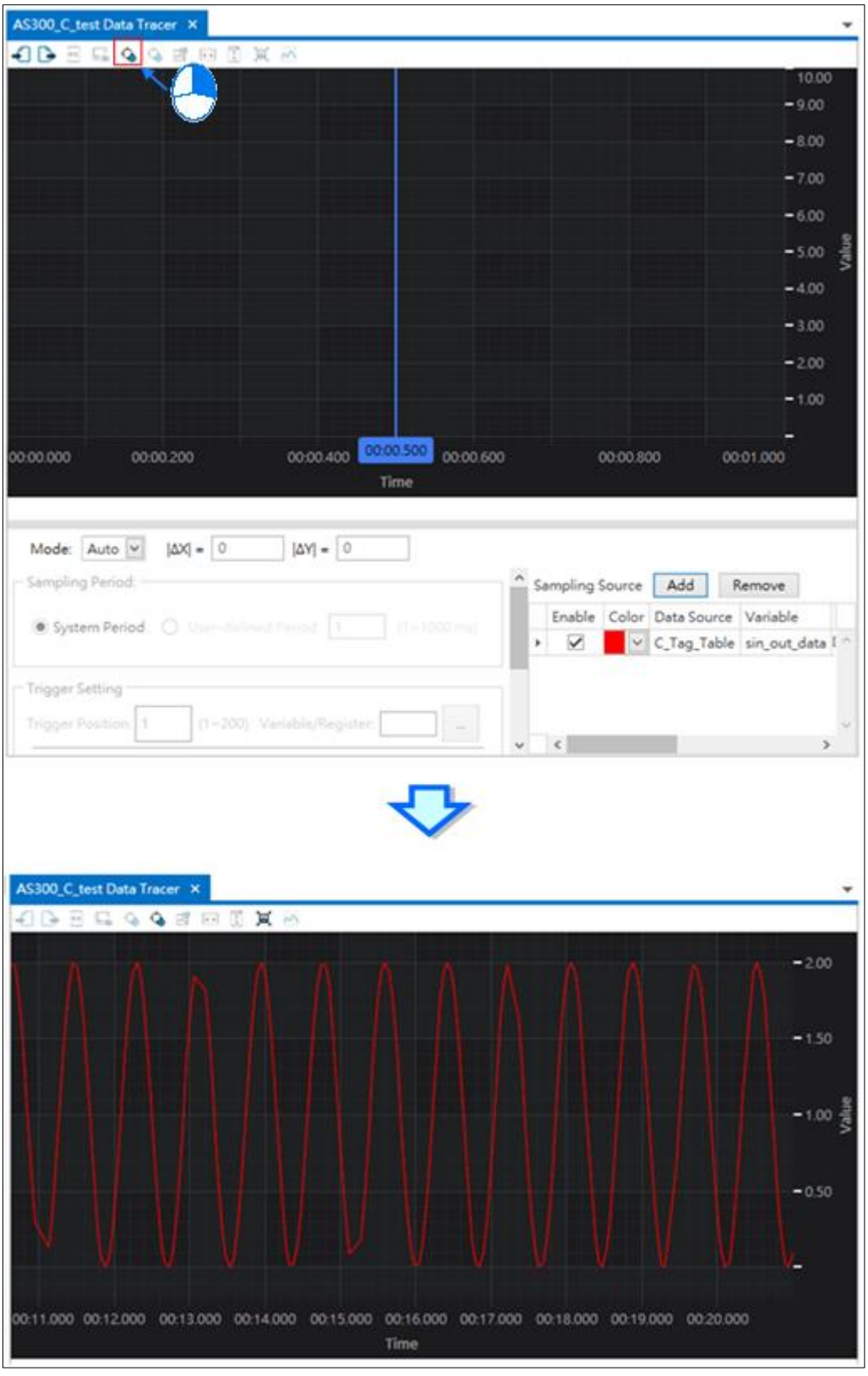


Figure 8 - 263: Real-time monitoring of data oscilloscope

8.4 Compile and Check

After editing the programs in IEC programming languages, user can check the syntax of the programming language with **Check** or compile the project or device with the help of **Compile**. The syntax and the structure in the present window are checked with **Check**. Execution code for project or device or program is generated only if **Compile** is successful, without any errors. It is possible to compile the entire project or individual device or individual program.

8.4.1 Check

Follow these steps to check a program POU:

1. Select the program POU to be checked.
2. Right-click on the program POU to open the context menu and click **Check** or

   Click **Operation** > **Check** in the menu bar.

   Result: The **Precompile** starts the **Check** operation and **Error List** window displays the check results with errors, warnings and messages if any. Sample check result is shown in the following figure.

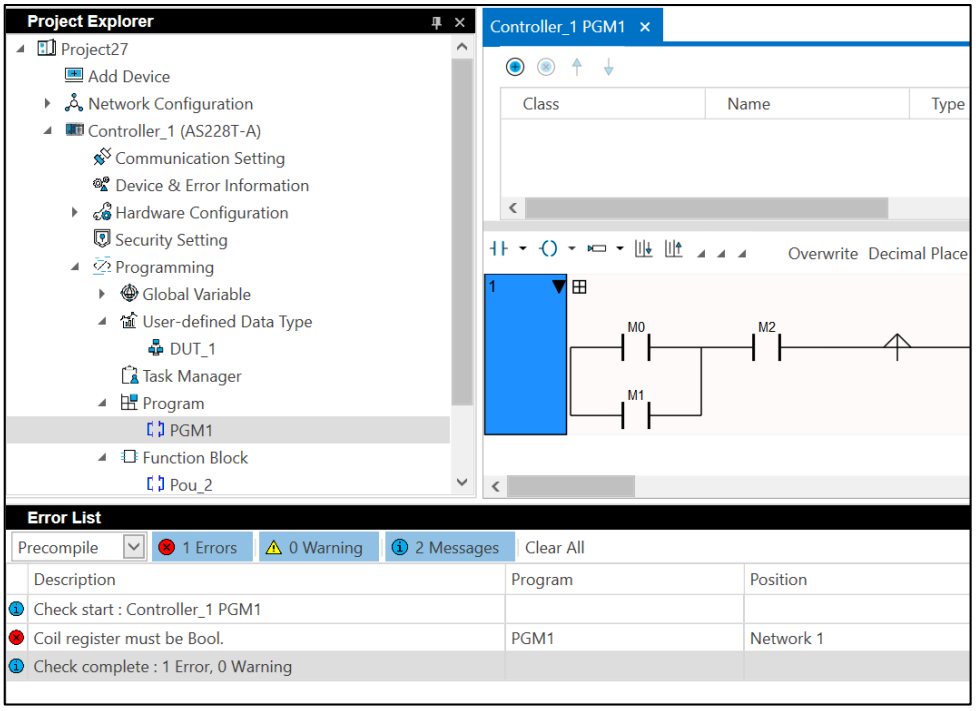


Figure 8 - 264: Check Operation

***NOTE**: If **Check** results contain errors or warnings, double-click on the message in **Error List** window to navigate to the respective item.*

8.4.2 Compile

Follow these steps to compile a project or a device or a program POU:

1. Select the device or program POU to be compiled.

2. Click

icon on the **Development** Toolbar.

   or

   Press **F11** key on the keyboard.

   or

   Right-click on the device or program POU to open context menu and click **Compile**.

Result: The Compiler will start the **Compile** operation and **Error List** window displays the compilation results with errors, warnings and messages if any. Sample compile result is shown in the following figure.

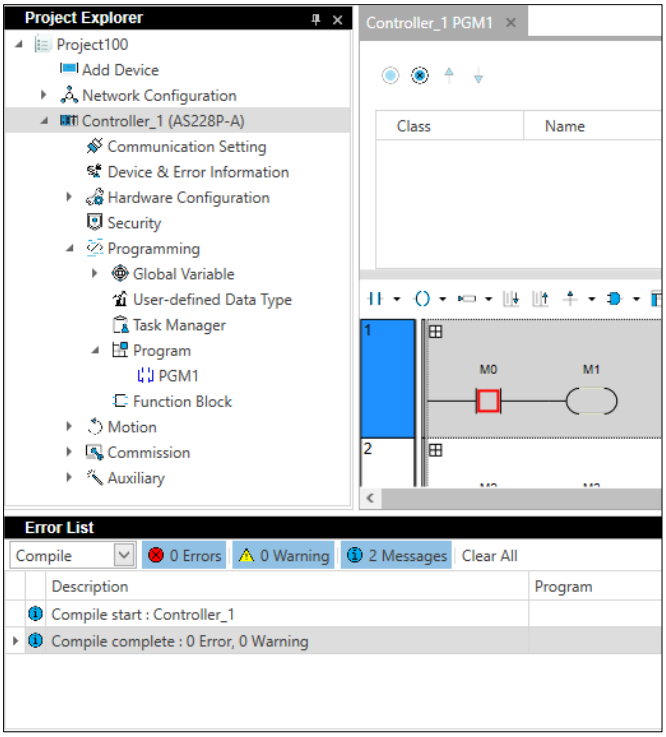


Figure 8 - 265: Compile Operation

***NOTE**: If **Compile** results contain errors or warnings, double-click on the message in **Error List** window to navigate to the respective item.*

8.5 Program Monitoring

In the Online program monitoring, user can view the present scan time and the status of the PLC in the status bar. An example of Online monitoring is shown in the following figure.

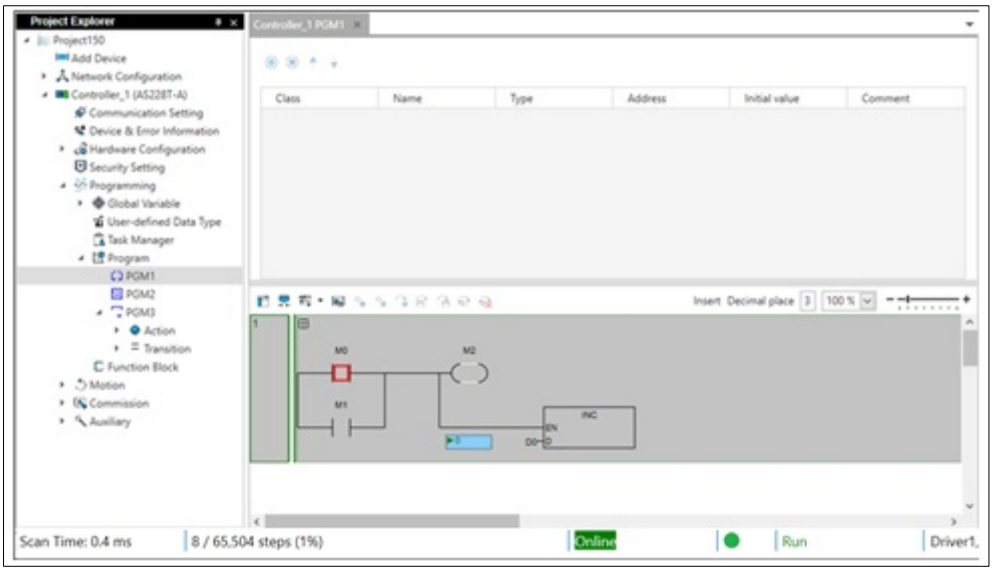


Figure 8 - 266: Program Monitoring

8.5.1 LD Program Monitoring

In Online program monitoring, LD POU will display the status or values of Contacts, Coils, APIs, Address, Variables etc. A Ladder Diagram POU in the program monitoring is shown in the following figure.

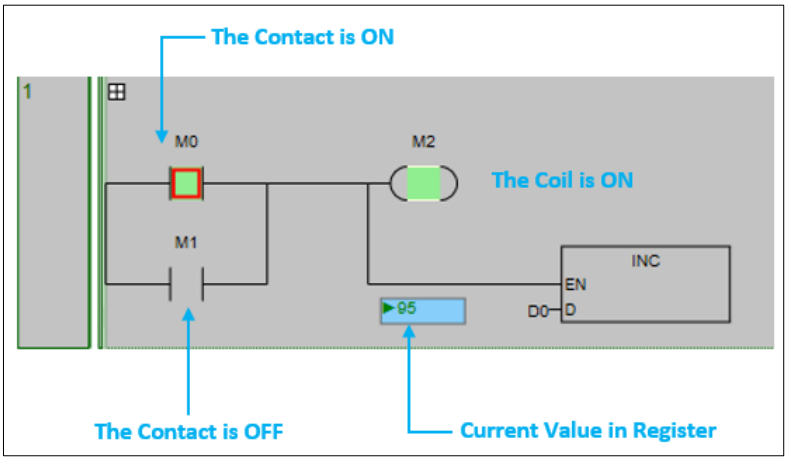


Figure 8 - 267: LD Program Monitoring

8.5.2 ST Program Monitoring

In Online program monitoring, ST POU will display the status of Registers or Values. Boolean Address or Variables display with the TRUE or FALSE status and other registers with the actual values. A Structured Text POU in the program monitoring is shown in the following figure.

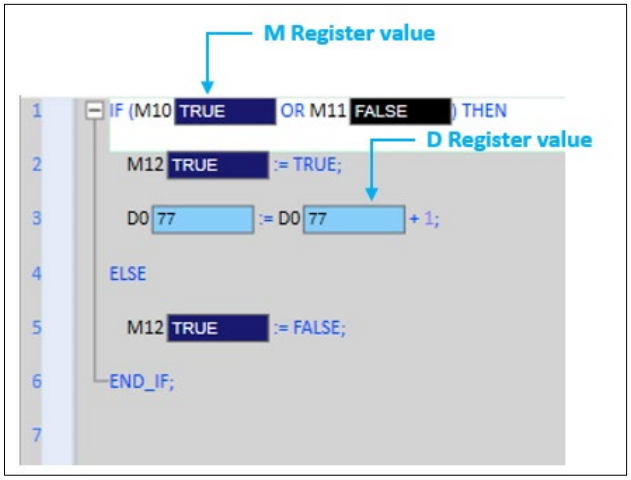


Figure 8 - 268: ST Program Monitoring

8.5.3 SFC Program Monitoring

In Online program monitoring, not only the SFC, but the Actions and Transitions are also monitored. A Sequential Function Chart POU in the program monitoring is shown in the following figure.

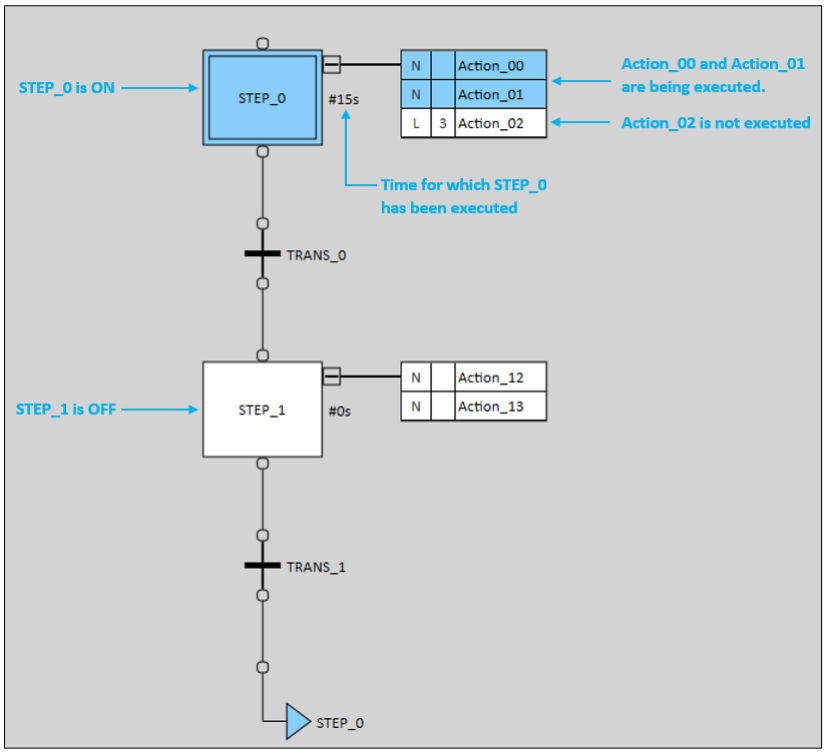


Figure 8 - 269: SFC Program Monitoring

8.6 Online Edit

When the system is in online mode, and the PLC is running, user can modify the program monitored through the function Online Edit. Refer the following table for more information about the limits set on the editing of programs in **Online Edit** mode.

Programming Language	Restriction in Online Edit mode
Ladder Diagram	• Only one Network can be edited at a time • User are allowed to add a Network • After a Network is added, user is allowed to edit only the newly added Network, previous Networks cannot be edited
Structured Text	• There is no limit on the number of lines which can be edited
Sequential Function Chart	• A Sequential Function Chart cannot be edited in online • Only Actions and Transitions can be modified in online • The limit set on the editing of an Action or Transition depends

Programming Language	Restriction in Online Edit mode
	upon the language used to create the Action or Transition
C language	• Online editing is not supported
Common	• DIADesigner must be in **Online** mode with the Controller • Controller must be in **RUN** mode • User can edit only one POU at a time • User cannot modify or add/delete **Variables**

During **Online Edit**, user has the option to **Download** (Only changes) in: -

- Object Code – Compiled code by PC
- Object Code + Program and Source Code – created by user

8.6.1 POU Online Edit

8.6.1.1 Online Edit in Ladder Diagram (LD) POU:

Follow these steps to perform **Online Edit** in a Ladder Diagram (LD) POU: -

1. Open the Ladder Diagram POU and click icon (Online Edit) in the LD POU toolbar.

   Result: DIADesigner enters **Online Edit** mode. Status bar displays **Online Edit** in mode area as shown in the following figure.

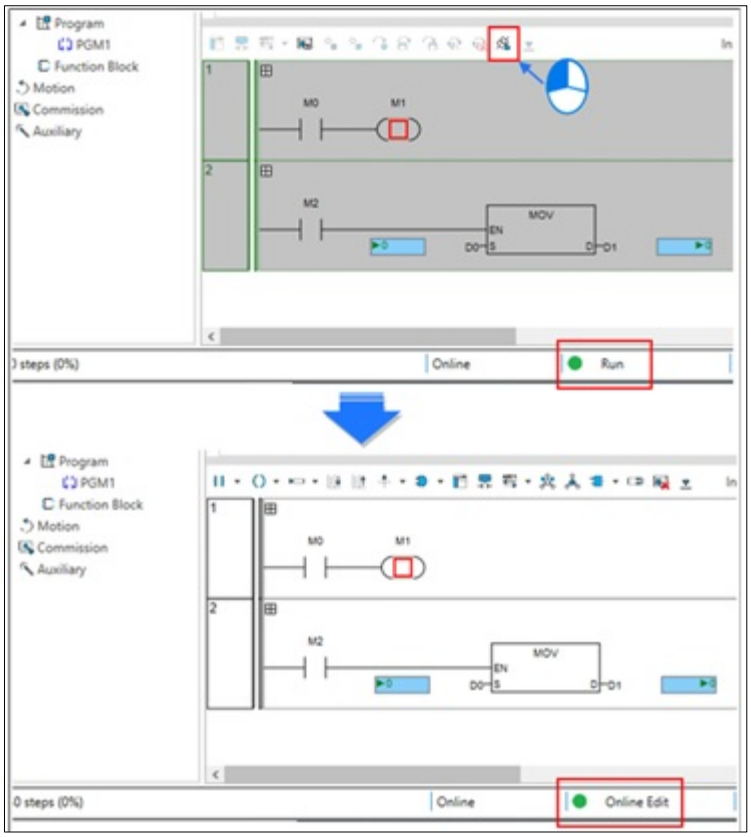


Figure 8 - 270: LD Online Edit

2. Edit the LD POU.

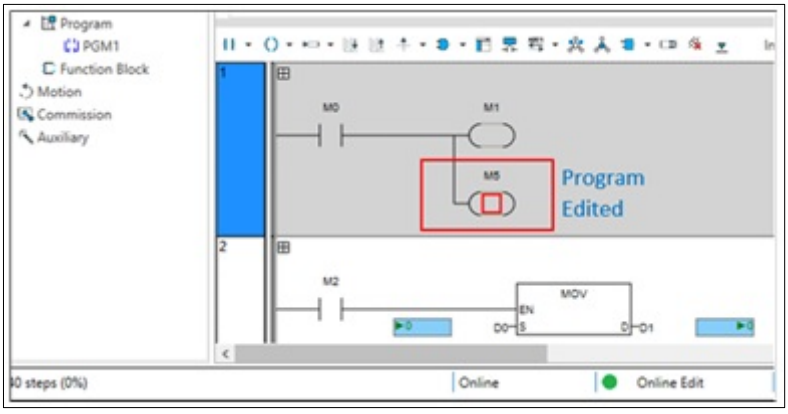


Figure 8 - 271: Edit LD POU

3. After the editing is complete, click icon (**Download Only changes**) in the LD POU Tool bar.

Result: The **Online change code** window displays as shown in the following figure.

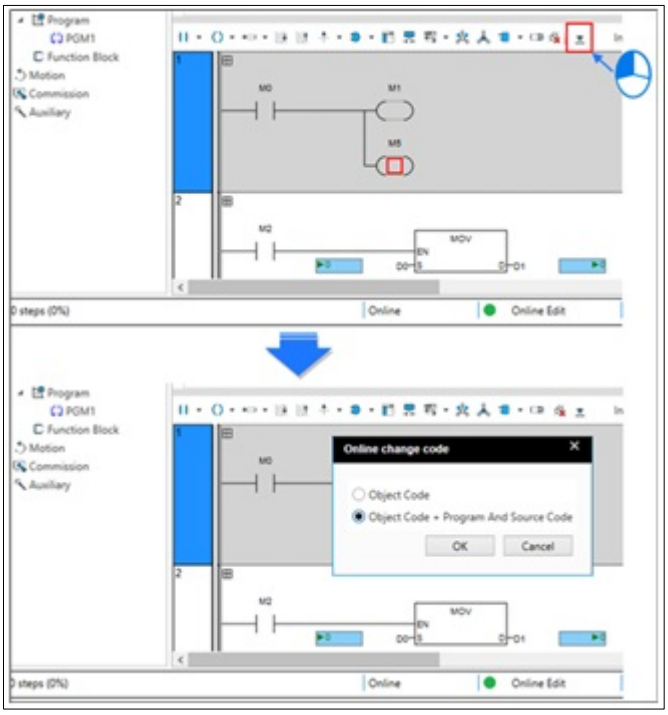


Figure 8 - 272: LD Download Online Changes

4. User may select **Object Code** or **Object Code & Program and Source Code** option as needed and click **OK** in the **Online Change Code** window.

   Result: The pop-up window – **Online edit successful** displays as shown in the following figure.

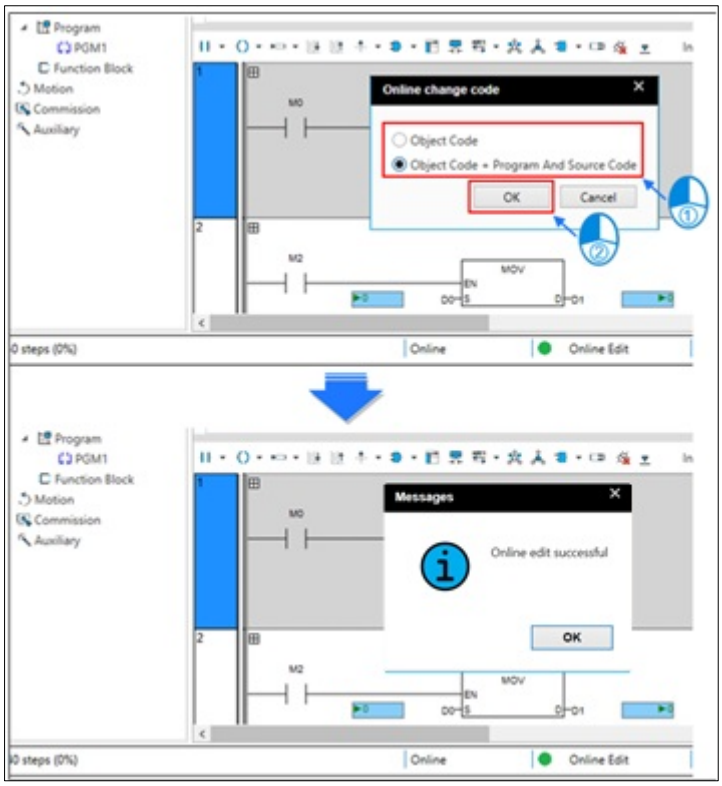


Figure 8 - 273: LD Online edit success

5. Click **OK** in the pop-up window.

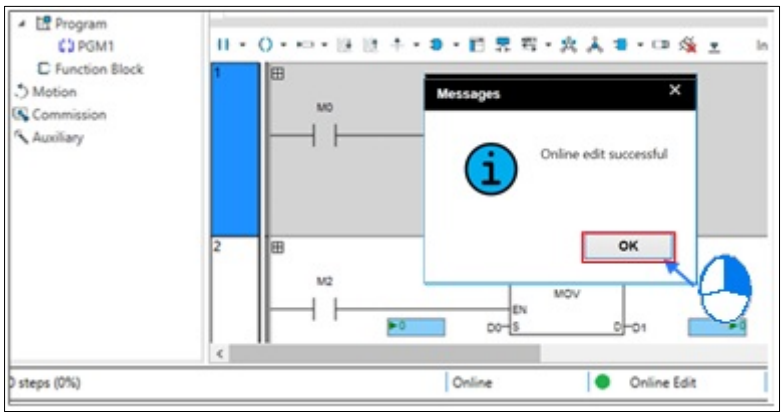


Figure 8 - 274: LD Online edit success - OK

6. Click icon (**Stop Online Edit**) in the LD POU Tool bar.

   Result: The **Online Edit** stops and DIADesigner returns to **RUN** mode.

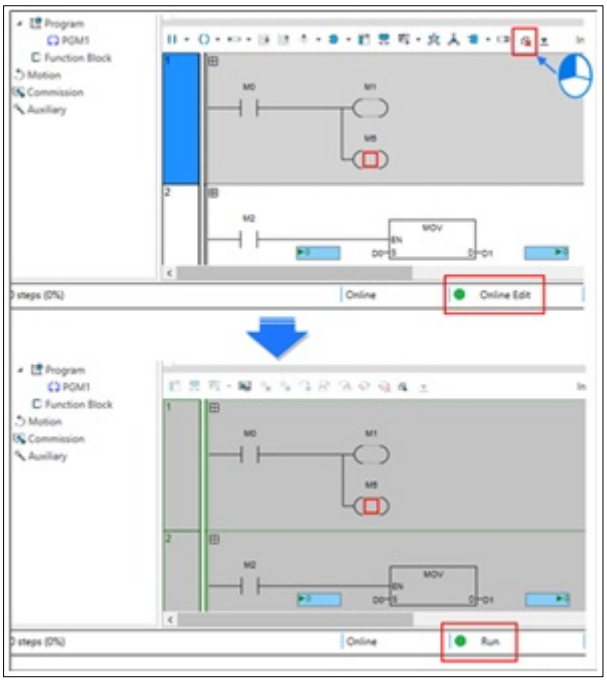


Figure 8 - 275: LD POU RUN after Online Edit

8.6.1.2 Online Edit in Structured Text (ST) POU:

Follow these steps to perform **Online Edit** in a Structured Text (ST) POU: -

1. Open the Structured Text POU and click icon (Online Edit) in the ST POU Tool bar.

   Result: DIADesigner enters **Online Edit** mode. Status bar displays **Online Edit** in mode area as shown in the following figure.

Figure 8 - 276: ST Online Edit

2. Edit the ST POU.

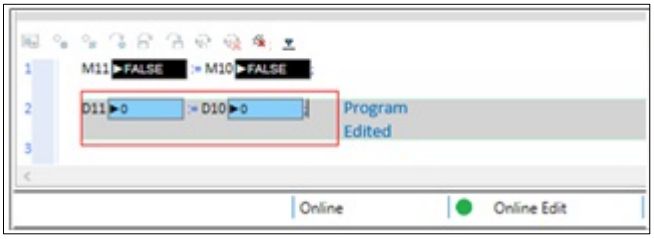


Figure 8 - 277: Edit ST POU

3. After the editing is complete, click icon (**Download Only changes**) in the ST POU Tool bar.

   Result: The **Online change code** window displays as shown in the following figure.

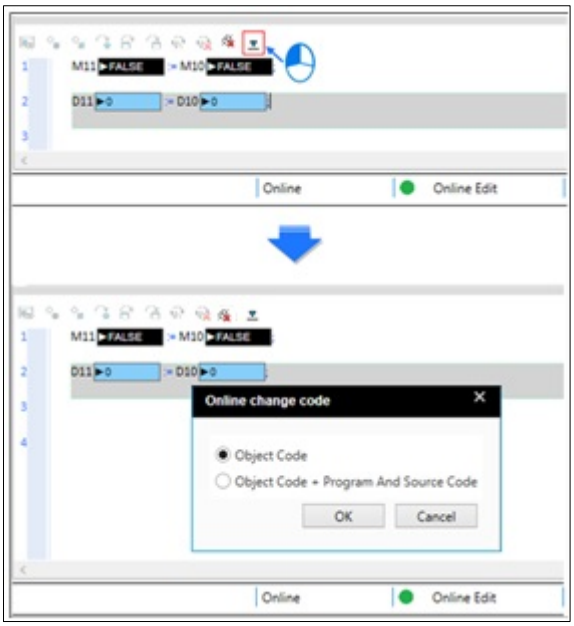


Figure 8 - 278: ST Download Online Changes

4. User may select **Object Code** or **Object Code & Program and Source Code** option as needed and click **OK** in the **Online Change Code** window.

   Result: The **Messages** window displays **Online edit successful** as shown in the following figure.

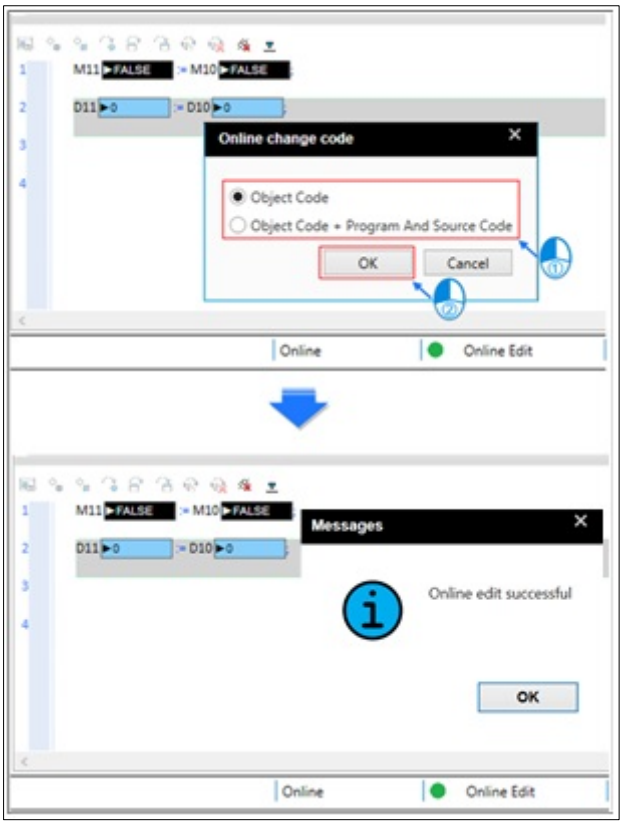


Figure 8 - 279: ST Online edit success

5. Click on **OK** button.

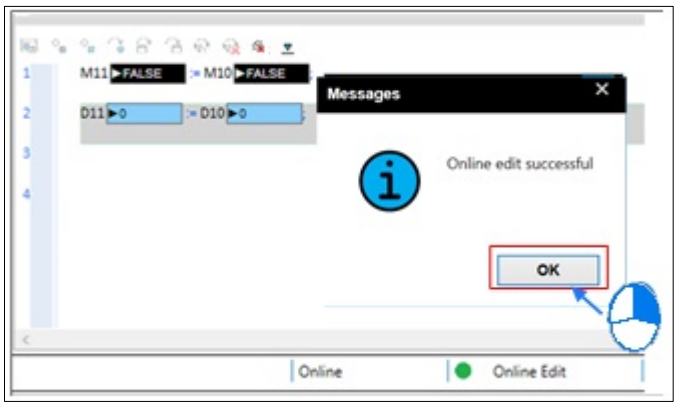


Figure 8 - 280: ST Online edit success - OK

6. Click icon (**Stop Online Edit**) in the ST POU Tool bar

   Result: The **Online Edit** is stopped and DIADesigner returns to **RUN** mode.

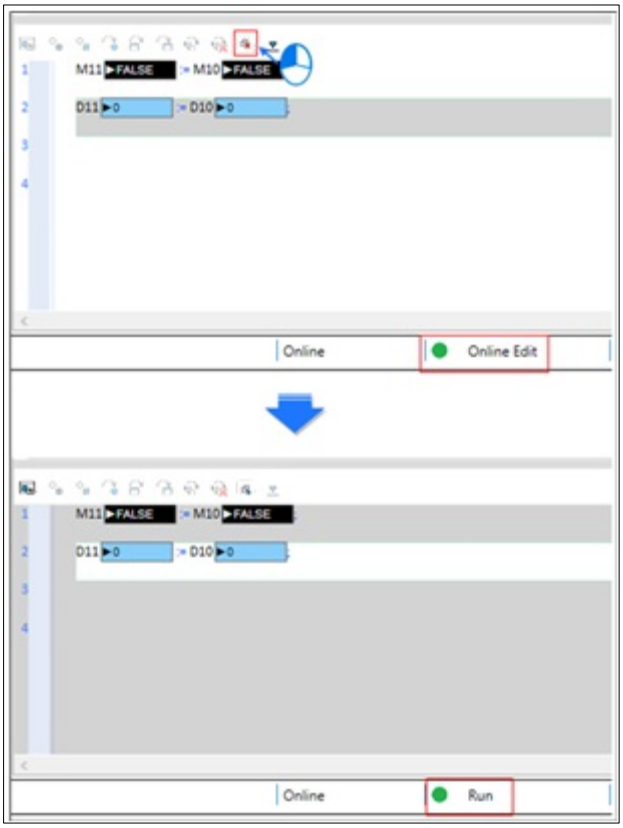


Figure 8 - 281: ST POU RUN after Online Edit

8.6.1.3 Online Edit in Sequential Function Chart (SFC) POU:

SFC POU cannot participate in an Online Edit. But the Actions and Transitions POUs which are made with LD or ST can take part in Online Edit. Procedure is same as that mentioned in the previous sections.

8.6.1.4 Online Edit – Stop and Recover

During Online Edit, after user's finish editing the code, they can choose to **Stop** the **Online Edit**. Stop of Online Edit results in two scenarios: -

- Stop Online Edit and Recover – Yes
- Stop Online Edit and Recover – No

In either ways, DIADesigner goes into **Offline** mode with the Controller. After that, user must compile the program and download the compiled program to the controller later, which is further explained in the following sub-subsection.

8.6.1.5 Stop Online Edit and Recover – Yes

1. During **Online Edit**, after finishing the edit, if user feels to cancel operation, they can do so by clicking the (Stop Online Edit) icon.

   Result: The **Messages** window – **Project is modified, but hasn't update to device. Do you want to recover?** displays as shown in the following figure.

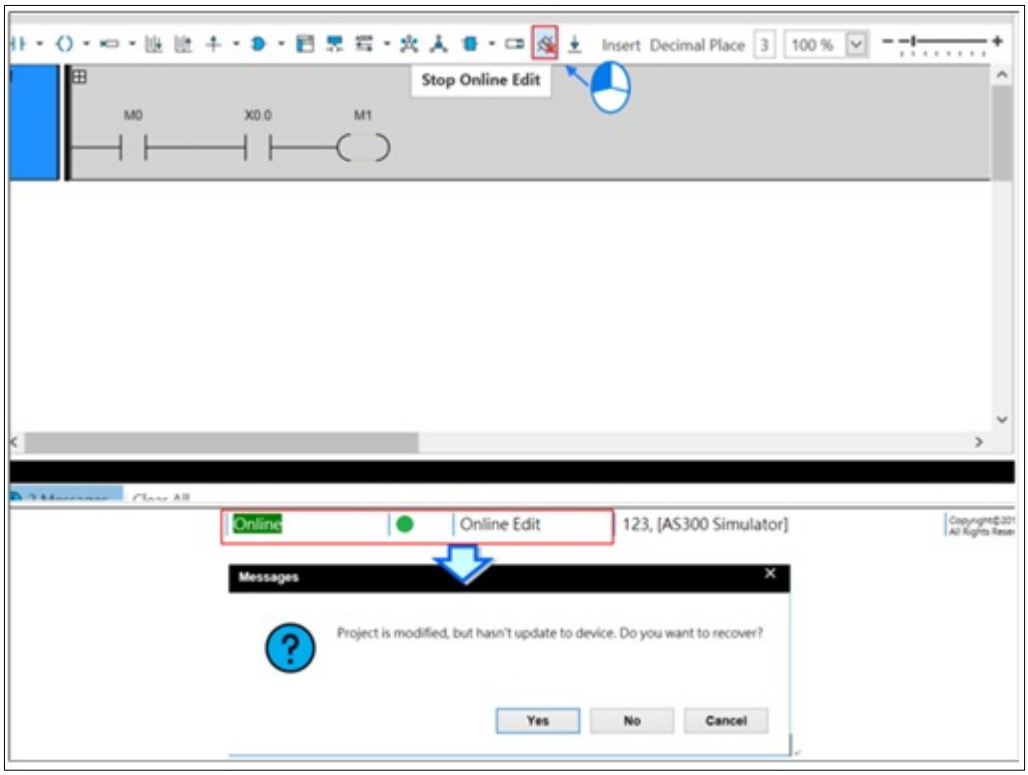


Figure 8 - 282: Stop Online Edit and Recover

2. Click on **Yes** button.

Result: Program edit is ignored, that is - program code before the Online Edit mode is restored and the status becomes **Offline** (Recover) as shown in the following figure.

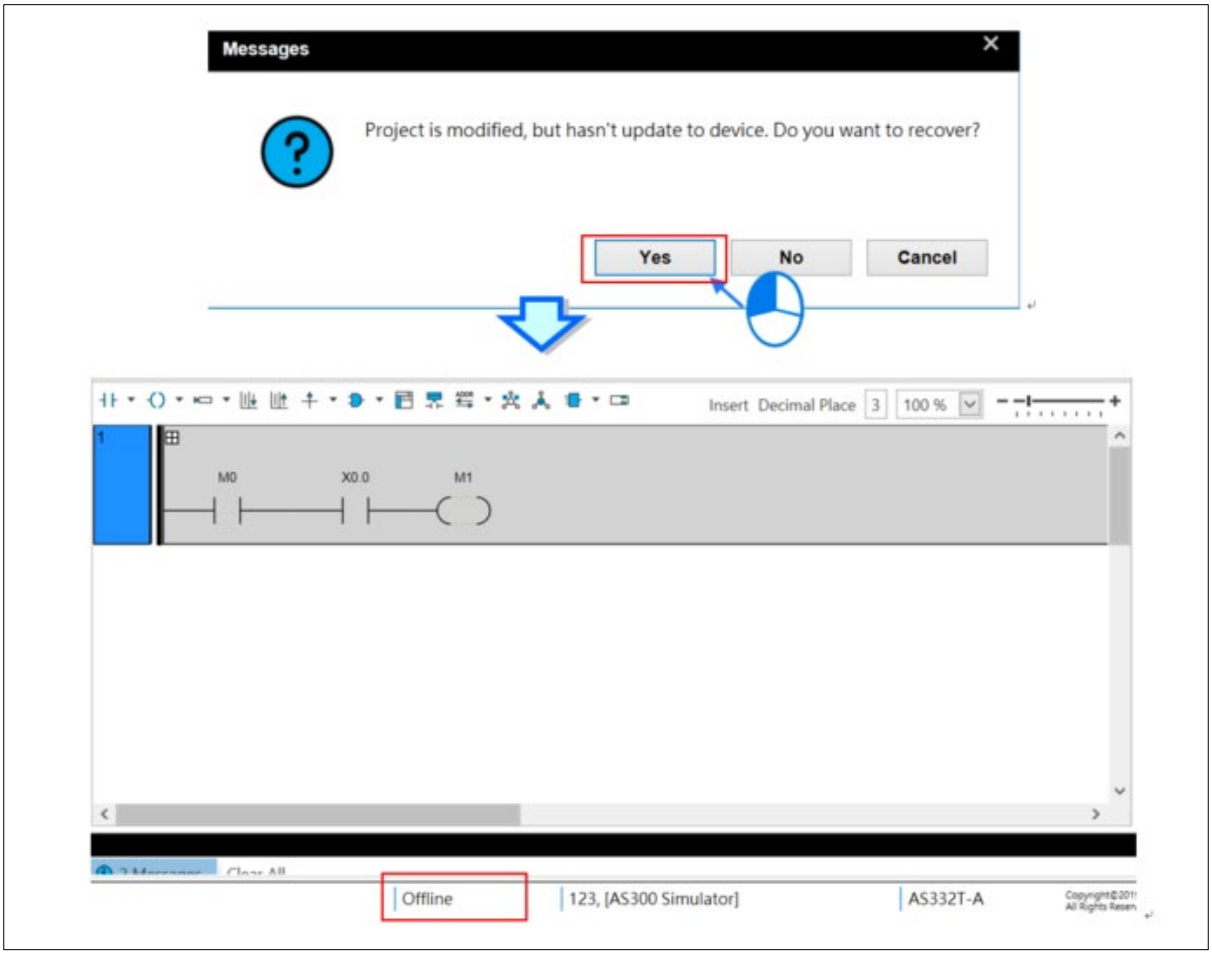


Figure 8 - 283: Stop Online Edit and Recover - Yes

8.6.1.6 Stop Online Edit and Recover – No

1. During **Online Edit**, after finishing the edit, if user feels to cancel operation, they can do so by clicking the (Stop Online Edit) icon.

   Result: The **Messages** window displays as shown in the following figure.

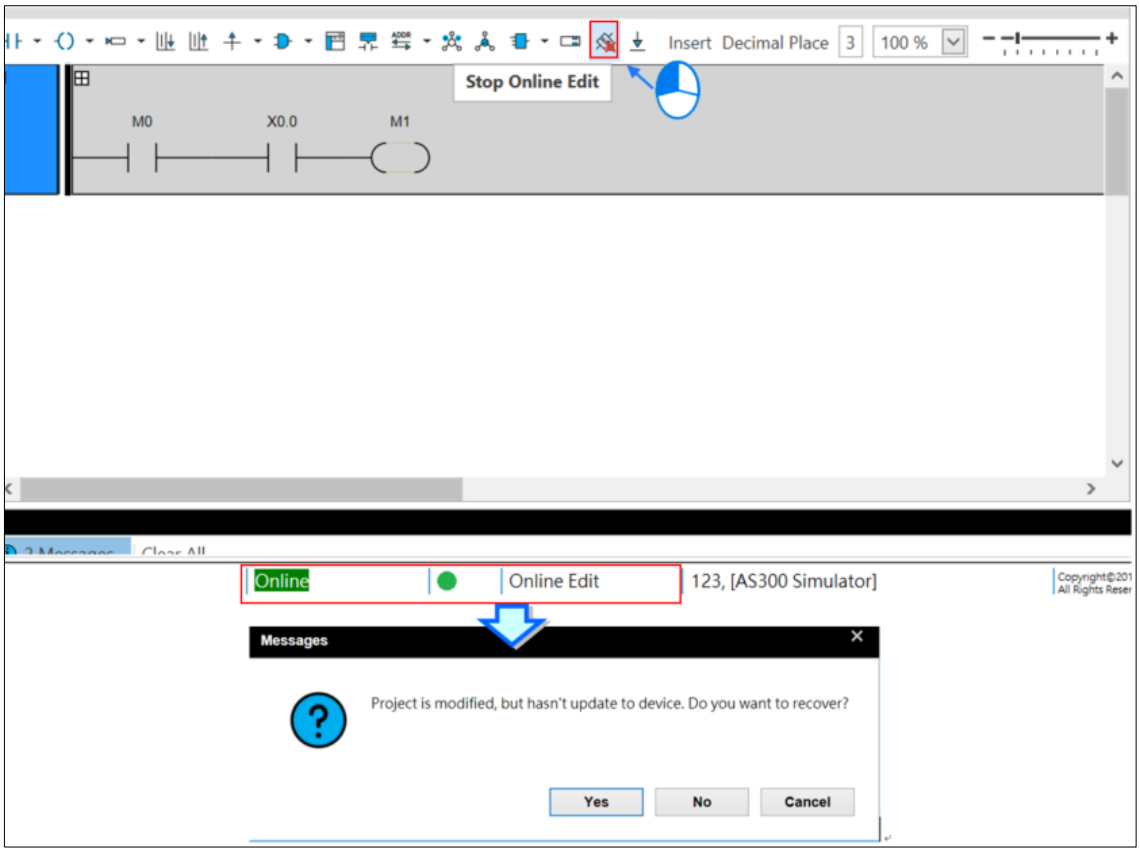


Figure 8 - 284: Stop Online Edit and Recover

2. Click on **No** button.

   Result: Program edited remains intact and the status becomes **Offline** (**No Recover**) as shown in the following figure.

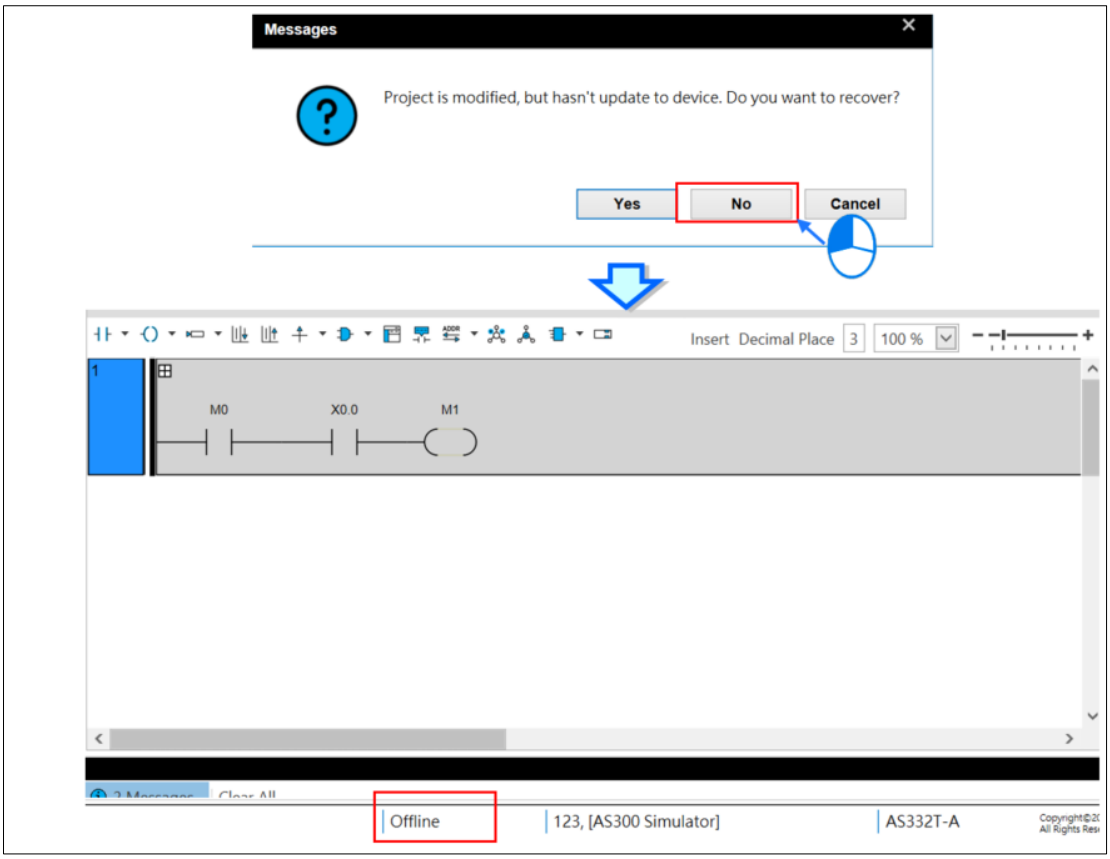


Figure 8 - 285: Stop Online Edit and Recover – No

8.7 Debug Mode

DIADesigner provides debugging functions for PLCs so that user can test the programs by means of **Debug Mode**. In DIADesigner, the debug mode is supported for all CPU modules for all the programming languages. However, debug mode for an SFC can only be executed for the Actions which are ON. Besides, if the state of a Transition in a SFC does not make the transition from one step to another step, user must change the state of the transition.

8.7.1 Enable Debug Mode

Debug Mode can be enabled only when the CPU is in Online mode. When Debug Mode is enabled, the status of the device in DIADesigner displays Debug Mode.

Follow these steps to enable Debug Mode:

1. Click icon in the POU Toolbar.

   Result: PLC enters Debug Mode and status bar displays **Debug Mode** in the status area as shown in the following figure.

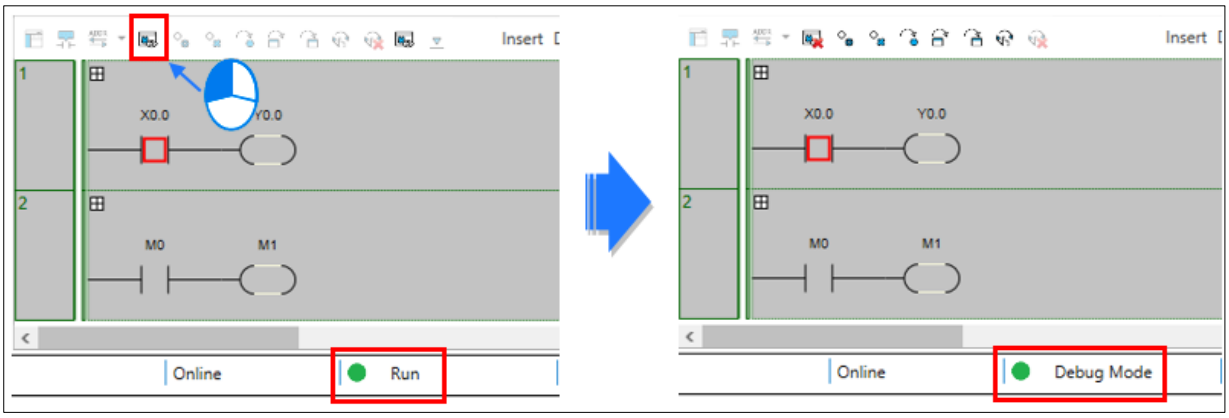


Figure 8 - 286: Debug Mode

***NOTE**: After clicking **Debug Mode** icon, if CPU does not enter Debug Mode, that means another user on the network is debugging the program in the CPU module.*

8.7.2 Break Point

A Break Point is an intentional stop marked in the application code at positions to check the behavior at those stop points. In Debug Mode, user are allowed to set the Break Point in an application. Maximum of 10 Break Points can be added to an AS series PLC.

8.7.2.1 Insert a Break Point

Follow these steps to insert a Break Point:

1. Open the POU and click icon in the POU Toolbar.

   Result: DIADesigner enters Debug Mode.

2. Select the position where user intends to add a Break Point.

3. Click icon in the POU Toolbar.

   Result: The **Break Point** is inserted in the position as shown in the following figure.

Figure 8 - 287: Insert Break Point

8.7.2.2 Delete a Break Point

Follow these steps to delete the Break Point:

1. Select the position where Break Point has to be deleted

2. Click icon in the POU Toolbar

   Result: Break Point is removed as shown in the following figure.

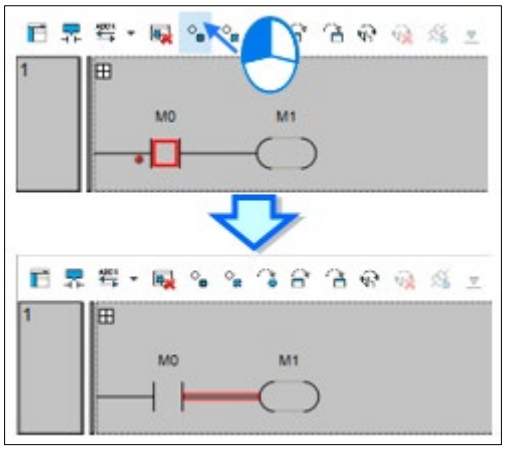


Figure 8 - 288: Delete Break Point

8.7.2.3 Delete all Break Points

Follow these steps to delete all the Break Points:

1. Click icon in the POU Toolbar.

   Result: All the Break Points are deleted as shown in the following figure.

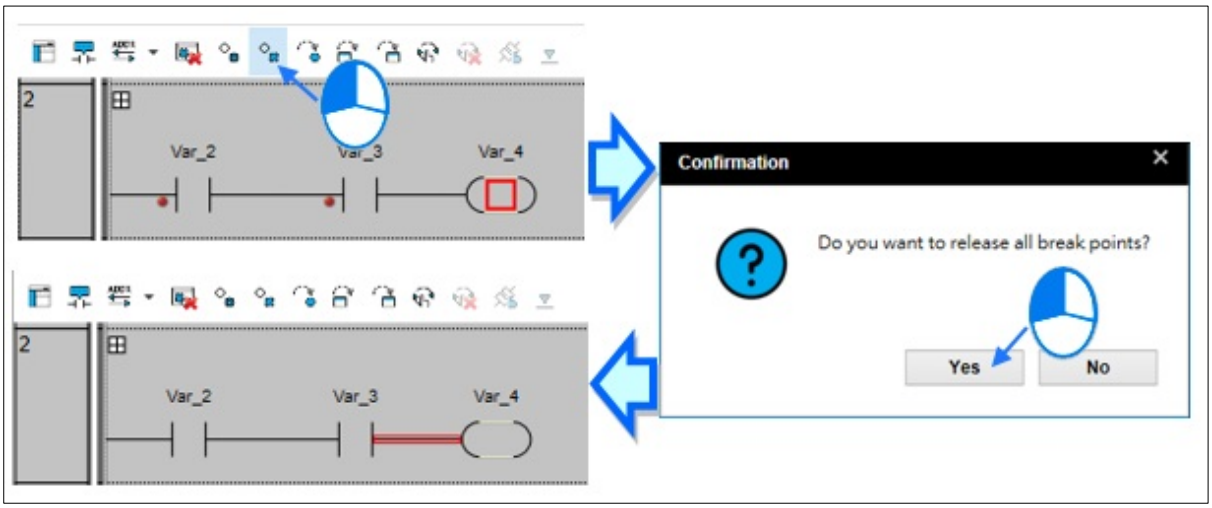


Figure 8 - 289: Delete all Break Points

8.7.2.4 Debug the application code

Follow these steps to debug the application code after inserting Break Points:

1. Click icon in the POU Toolbar.

   Result: The execution of the program will stop at a Break Point as shown in the following figure.

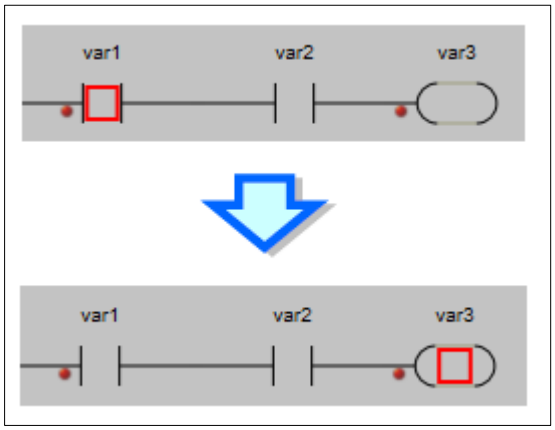


Figure 8 - 290: Debug the Application Code

2. Click icon again in the POU Toolbar.

   Result: The execution of the program will continue, and will stop at the next Break Point.

   ***NOTE**: The position where the execution of the program stops is the part of the program which has not been executed*.

8.7.3 Continue Run Execution

Continue Run Execution function helps user to execute the portion of the application code between one Break Point and the next Break Point.

Follow these steps to perform Continuous Execution:

1. Click icon on the POU Toolbar

Result: The execution of the program will stop at a Break Point.

2. Click icon on the POU Toolbar again.

   Result: The execution of the program will continue and will stop at the next Break Point as shown in the following figure.

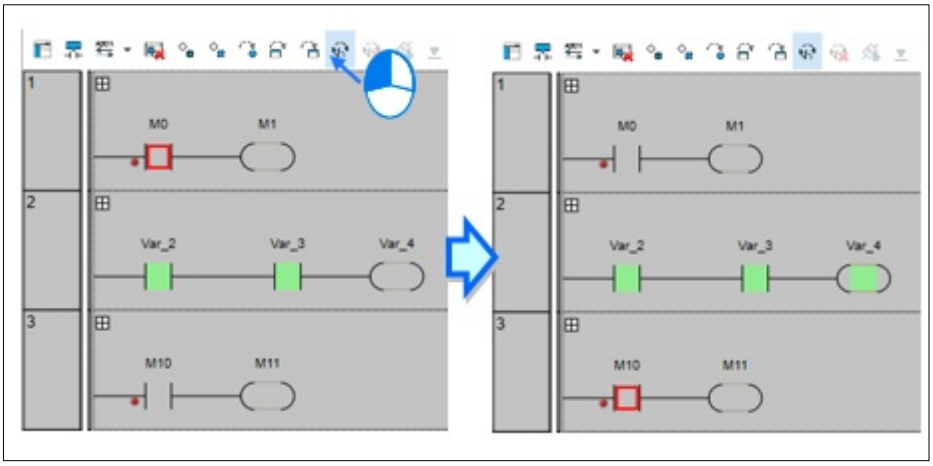


Figure 8 - 291: Continuous Run Execution

***NOTE**: If there are no more Break Points, the program is scanned repeatedly after the Stop Continue Run icon is clicked.*

8.7.4 Single-Step Execution

After or on the toolbar is clicked, the part of the program selected is executed. If a function block is met, the function block is executed after is clicked, and the first step in the function block is 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 user want to perform single-step execution in a structured text, they must add breakpoints to the

structured text. If user 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 in the following figure.

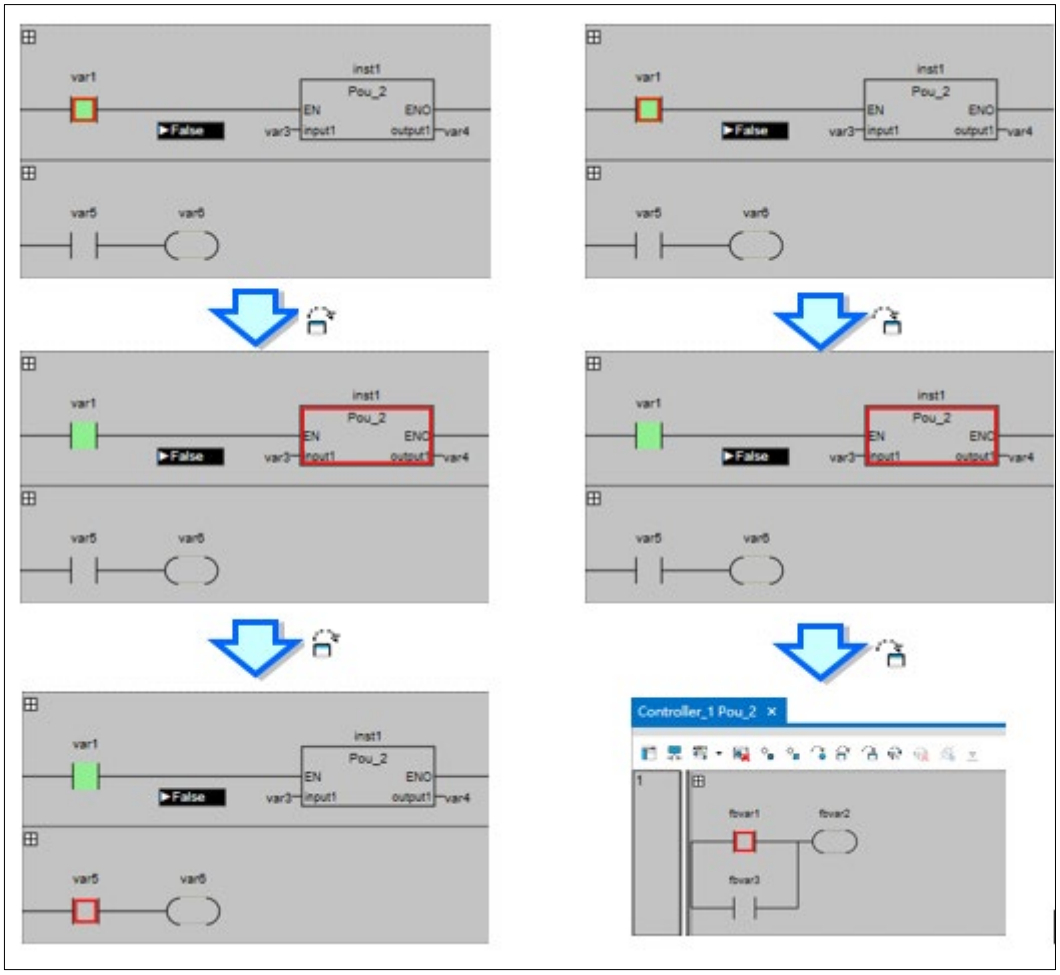


Figure 8 - 292: Single Step Execution

Chapter 9: Library Management tools

9.1 Library Project

The use of library projects can reduce the time spent by user to develop repetitive functions in the project development process. User only need to install the built library project to modify the content and include it in version control for subsequent maintenance. The above can greatly reduce the cost of design, development and implementation, testing and document writing.

9.1.1 Create Library Project

To create a project follow the steps,

1. When adding a project, select the library project type.

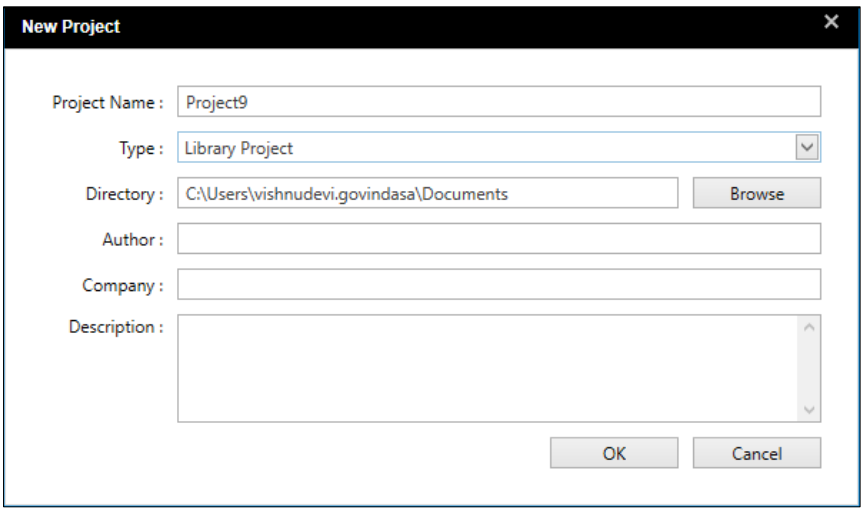


Figure 9 - 1: Add Library Project

2. After the library project is created, double-click the selected library series on the project tree as the corresponding model of the library.

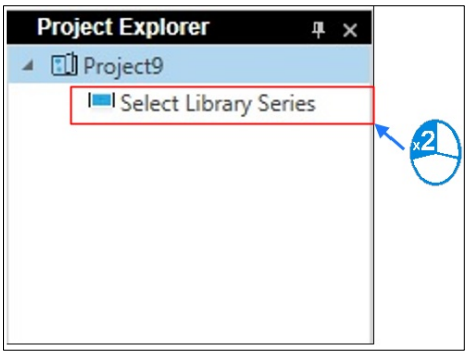


Figure 9 - 2: Library Project – Project Tree

3. Take AS300 as an example, click OK with the mouse; after specifying, the library project only supports the specified AS300 series models, and the library project cannot be installed on other models.

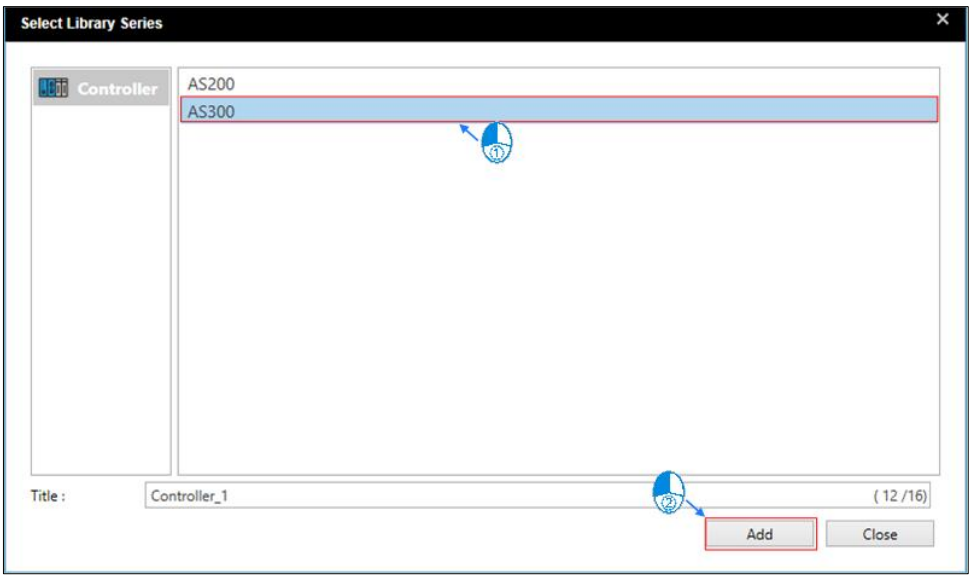


Figure 9 - 3: Select Library Series

4. After selecting the specified library model, you can edit the library content; currently the library content support only supports custom data types (including structure, enumeration and union) and function blocks (LD and ST).

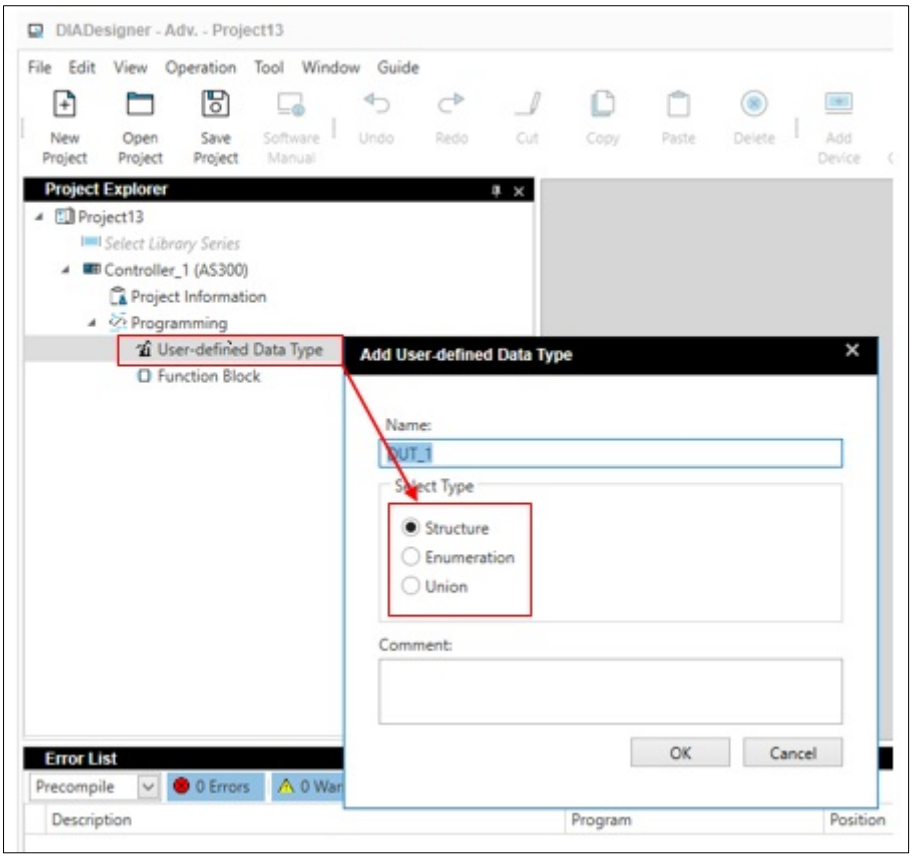


Figure 9 - 4: Custom data type support type list

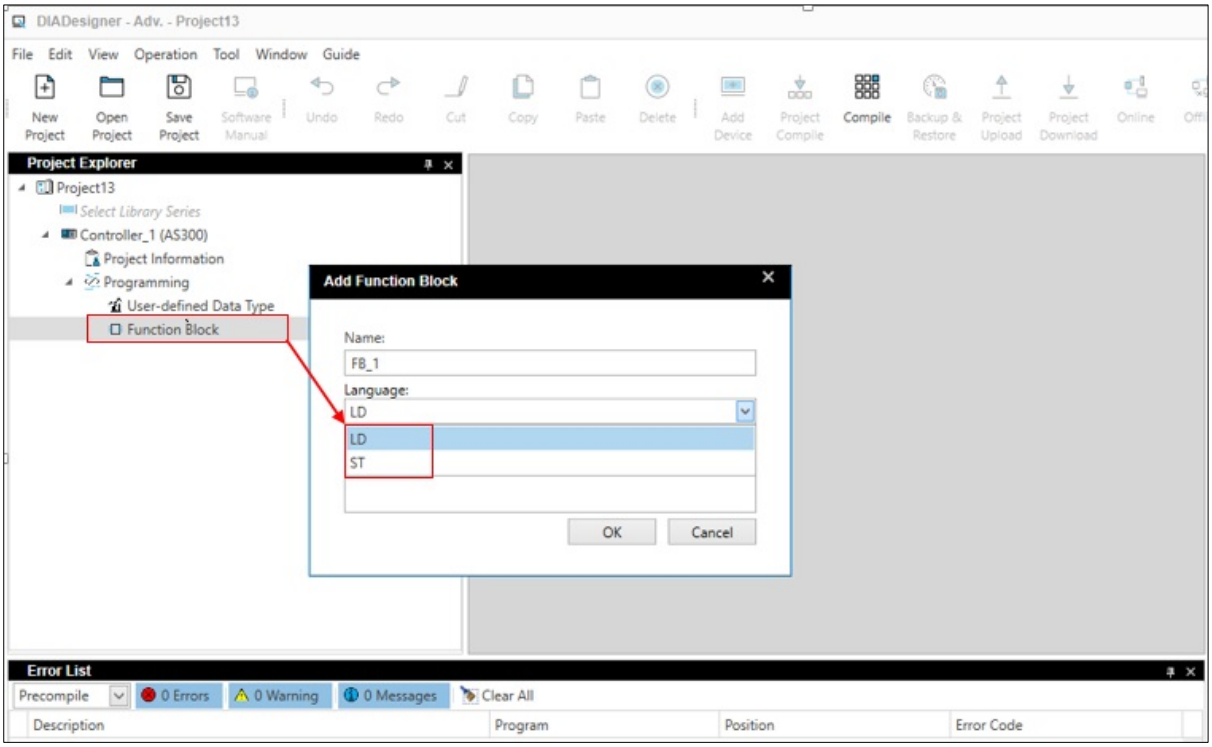


Figure 9 - 5: FB support list

9.1.2 Project Information

After editing the contents of the library project, you need to double-click the project information to edit the library project version information before archiving.

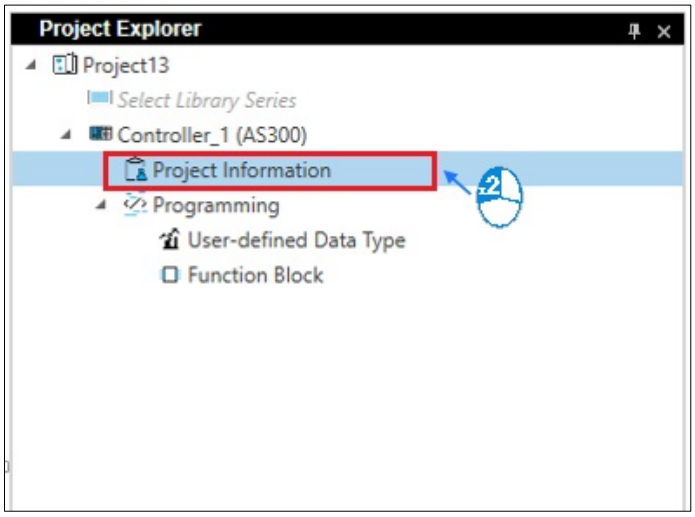


Figure 9 - 6: Project Information

The required fields in the project information are: Company, Title, Version and NameSpace.

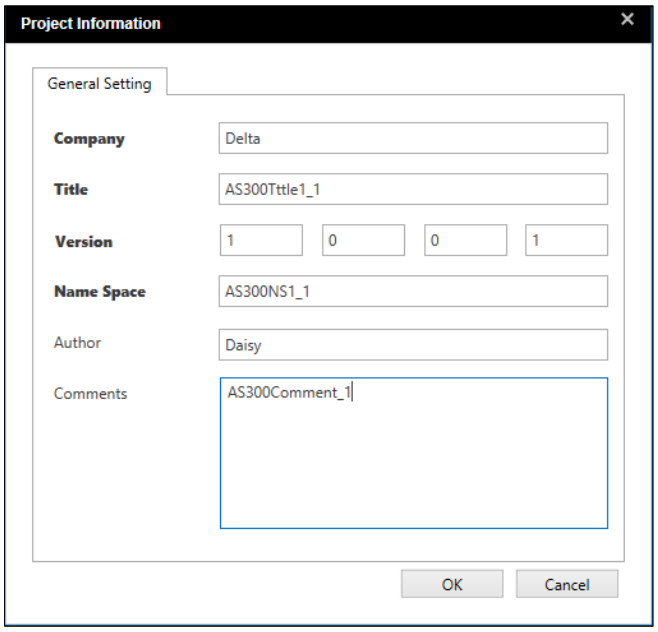


Figure 9 - 7: Project information field information

9.1.3 Repository project

The library project provides two formats for archiving, namely DIA Library Project (DIA Library Project) and Compiled Library Files (compiled library). The difference is that the latter is a library file whose content is protected. Once closed, it cannot be opened and modified again. In the library manager after installing and adding a new compilation library, double-clicking the item inside cannot view the content.

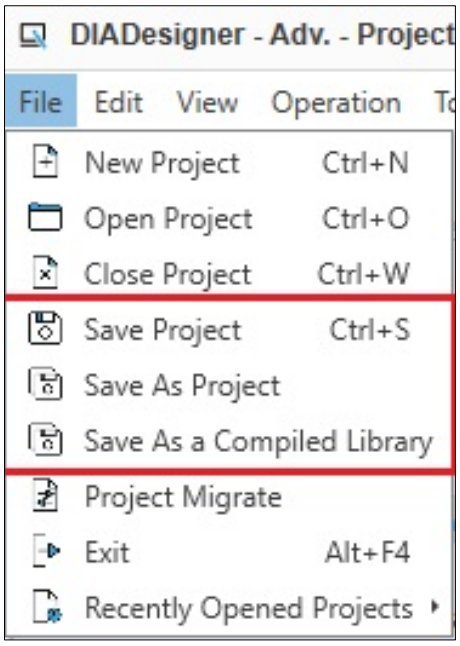


Figure 9 - 8: Library project archive method

9.2 Library Manager

The library can be installed, removed, added, and deleted in the library manager (the format must be DIA Library Project and Compiled Library Files), and can be used for calling in the program.

1. Double-click the library manager in the project tree.

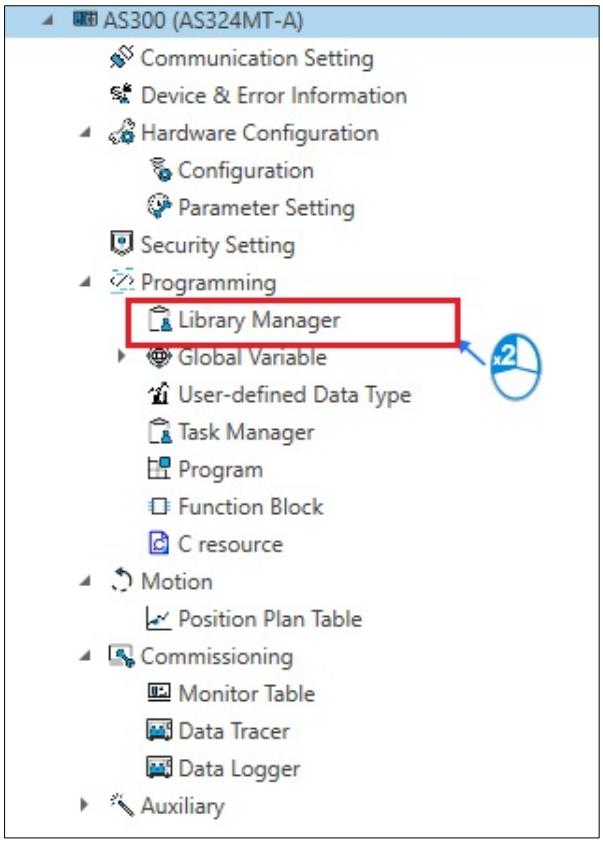


Figure 9 - 9: Project tree - Library manager

2. Library Manager Page

   The page block is defined as follows:

   ❶ Toolbar: Provide library resource library (installation, removal), add library and delete library.

   ❷ List of newly added libraries.

   ❸ The content items covered in DL_HVAC; when the left mouse button clicks on the item, there will be corresponding annotations on the right side (4); when the left mouse button double clicks on the item, the original content code will be opened, if it is Compile the library, a warning message will pop up and it cannot be opened.

   ❹ Annotations corresponding to each item.

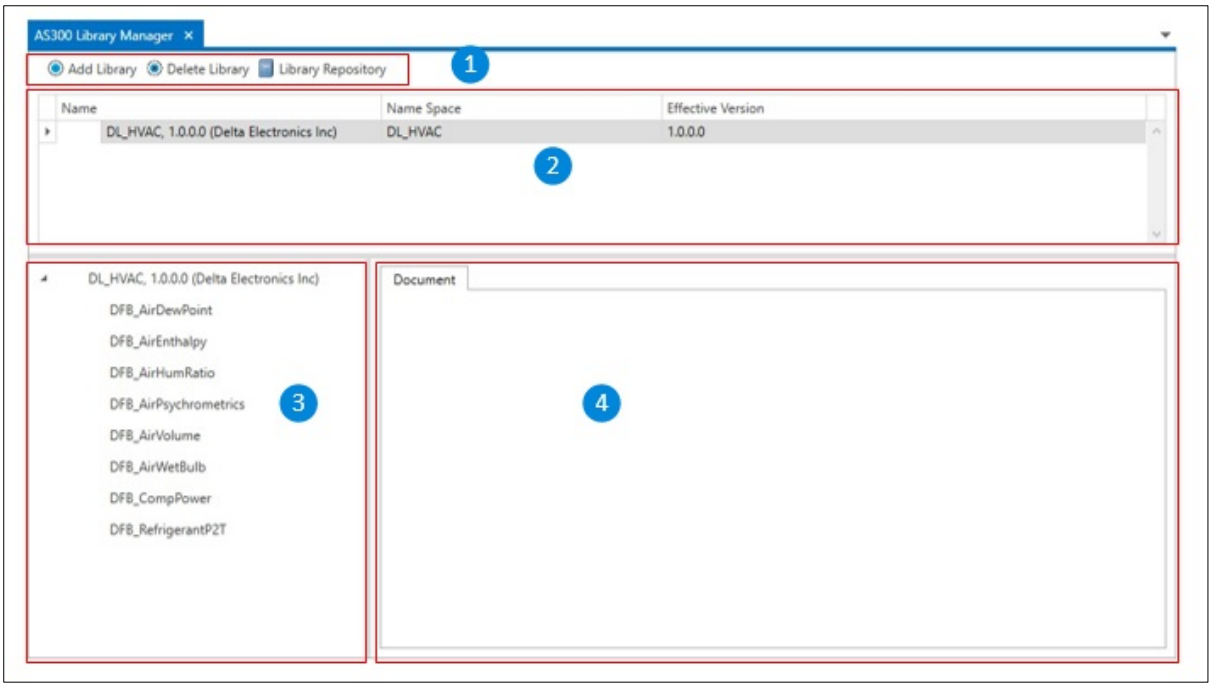


Figure 9 - 10: Library manager page block definition

9.2.1 Resource Library

Figure 9 - 11: Library Repository

The library can be used for installation and removal of library projects and library compilation files. After installation, the library version, company and version number will be displayed. Multiple different versions can be installed with the same library name, and you can click the specified version after installation Do removal; the built-in Delta library cannot be removed.

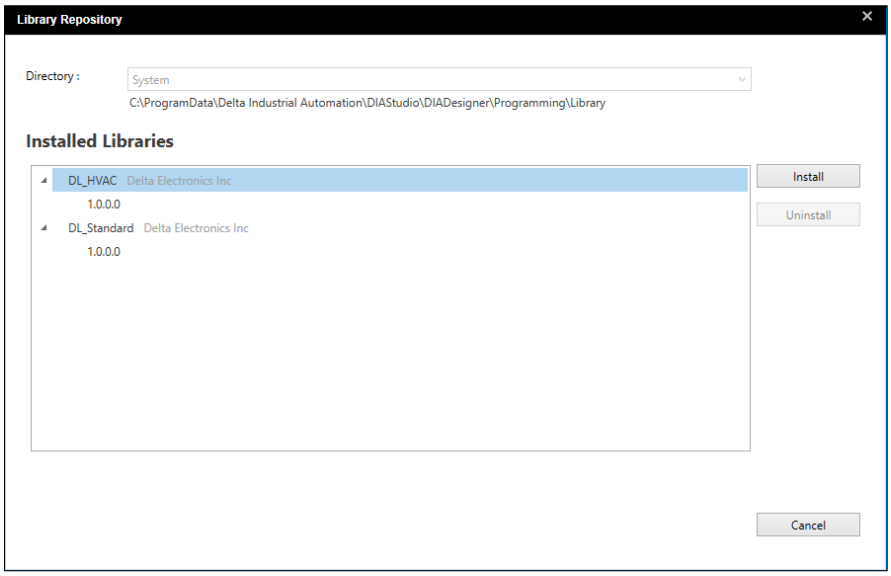


Figure 9 - 12: Resource Library

9.2.2 Add Library

Figure 9 - 13: Add Library

Add version selection for the installed library project. After the addition, it will be displayed on the library manager page, and the user can call the content in the program.

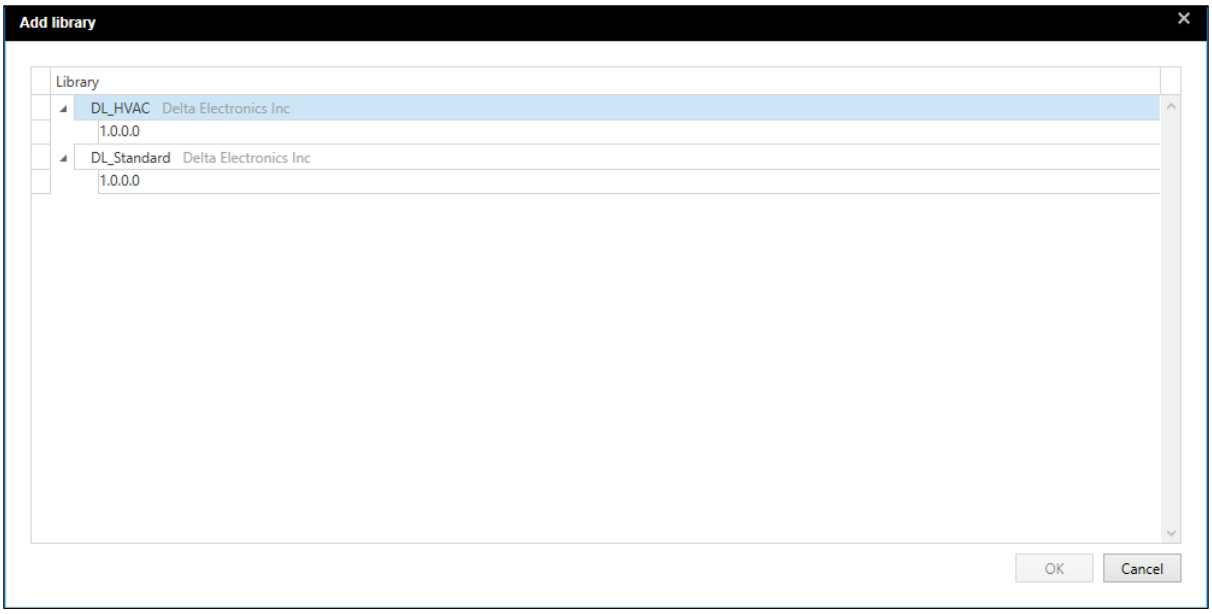


Figure 9 - 14: Add Library window

9.2.3 Delete Library

Figure 9 - 15: Delete Library

Delete the library for the newly added library project. After the deletion is completed, it will disappear on the library manager page. If the user references the deleted library in the program, the compilation will report an error.

9.2.4 Library Usage

After the library project is installed and added, you can call and use the custom data type and FB provided under the library project in the program area. The following takes the built-in DL_HVAC library of Delta as an example, and you want to call in the LD POU DFB_AirDewPoint under DL_HVAC:

The first method: call and select DFB_AirDewPoint under the resource library through the command wizard

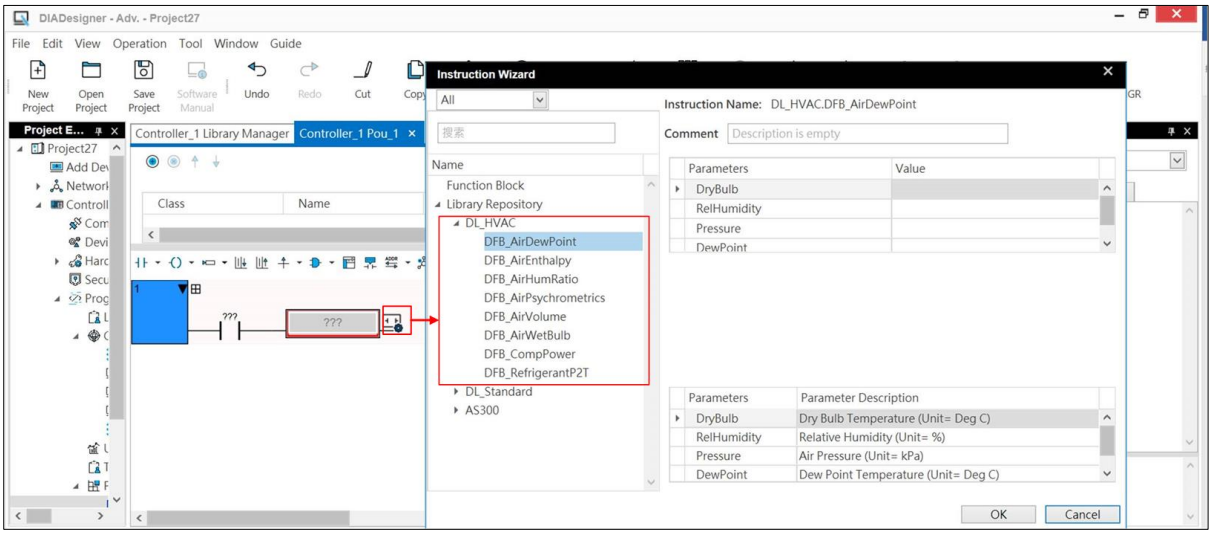


Figure 9 - 16: Add library commands via command wizard

The second way: In the data type of the variable table, select the function block DFB_AirDewPoint

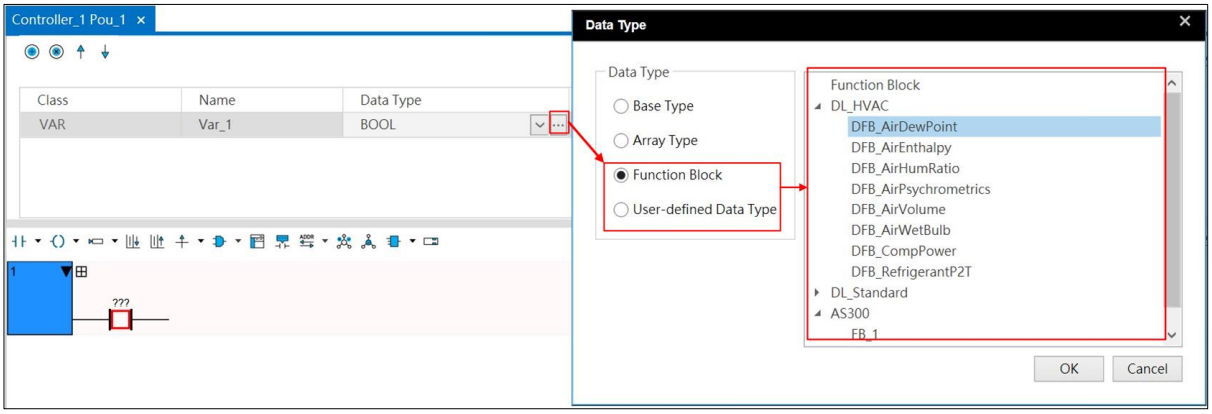


Figure 9 - 17: Add library commands by specifying variable data types

After selecting DFB_AirDewPoint, it can be successfully used in the program area.

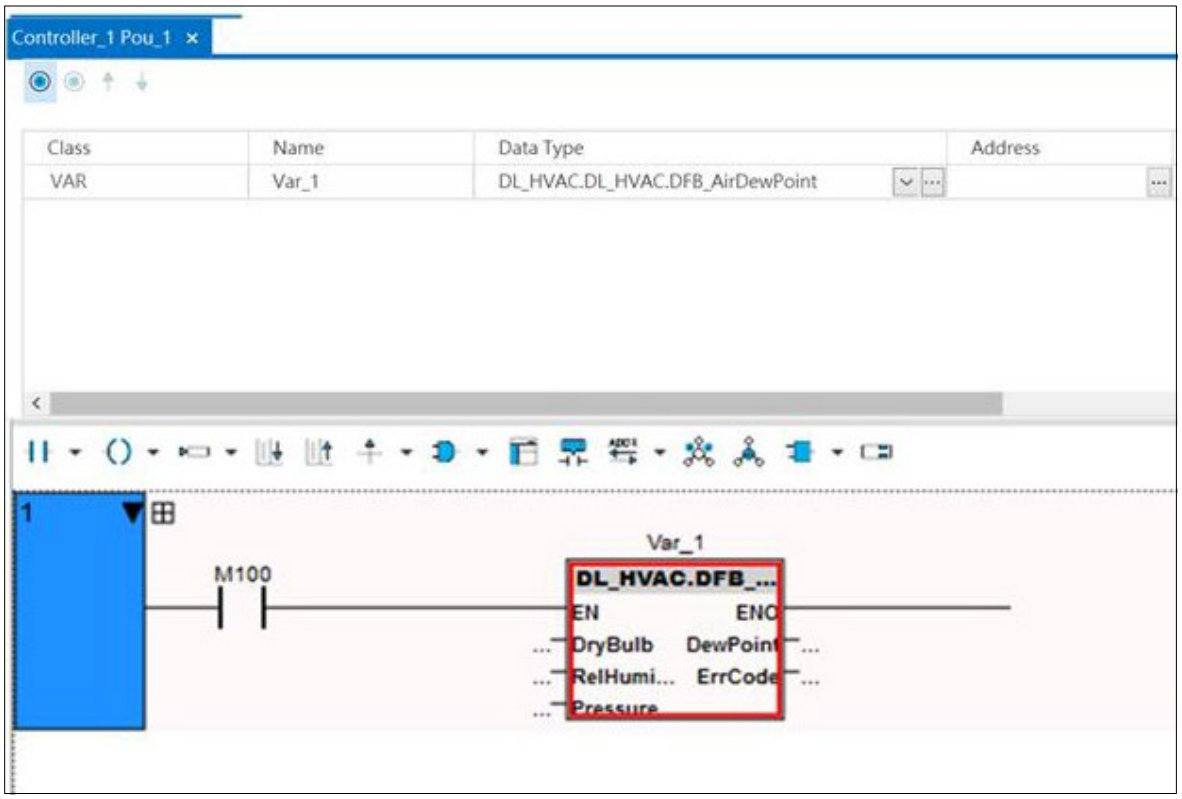


Figure 9 - 18: Program successfully added library commands

Chapter10: Motion Tools

10.1 Position Plan Table

10.1.1 About Position Plan Table

DIADesigner provides an easy table for user to set up the single-axis or 2-axis motion control. It is not required for the user to know the complicated motion control rules to complete the setup. Create and download the **Position Plan Table** and then along with the instructions, the PLC can output axis motions in the set number and order accordingly.

10.1.2 Setting the Position Plan Table

Double-click on the **Controller** > **Motion** > **Position Plan Table**.

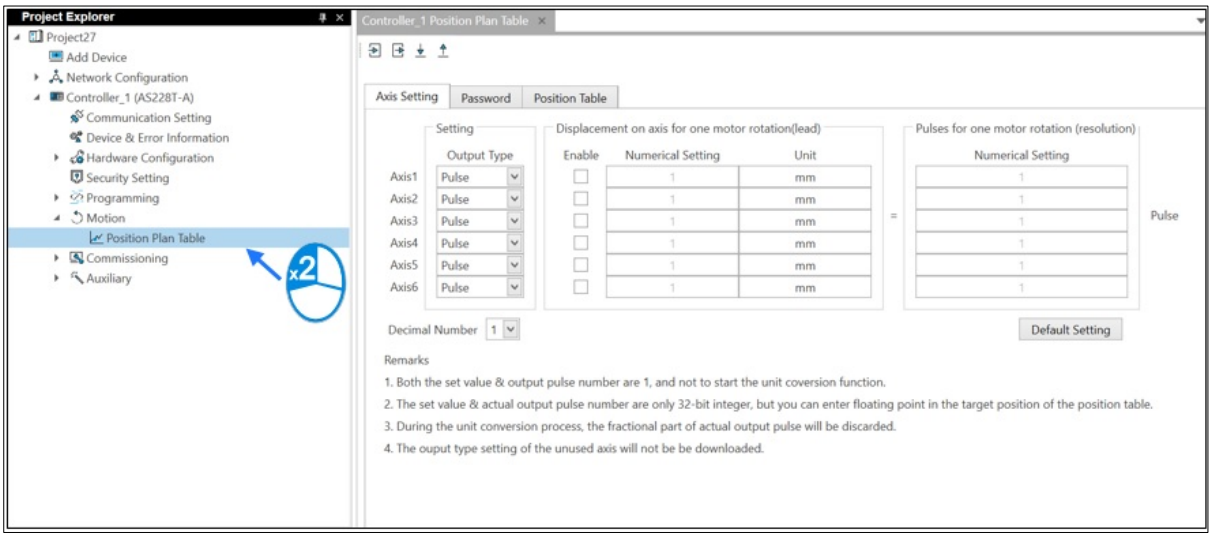


Figure 10 - 1: Position Plan Table

The **Position Plan Table** window has three tabs: -

- Axis Setting
- Password
- Position Table

The icons in Position Plan Table toolbar are described in the following table:

Function	Description
	Click to import **Position Plan Table**.
	Click to export **Position Plan Table**.
	Click to download **Position Plan Table**.
	Click to upload **Position Plan Table**.

10.1.3 Axis Setting

User can setup control output for each axis in the **Axis Setting** tab. The number of the axis display varies according to the supported PLC (e.g. AS300 series – 6 axis).

The functions in **Axis Setting** tab are described in the following table:

Function	Description
Setting – Output Mode	Select the output mode for the axis from drop-down list. Options are: • Pulse • Pulse/Dir • AB Phase ***NOTE**: **Pulse** is the default value. When executing 2-axis interpolation movement, **DO NOT** choose **Pulse** in **Output Mode**, instead choose **Pulse/Dir** or **AB Phase**.*
Displacement on axis for one motor rotation (lead) – Enable	Select **Enable** checkbox of the axis which needs to make a unit conversion.
Displacement on axis for one motor rotation (lead) – Numerical Setting	Enter the value for **Numerical Setting** of the axis.

Function	Description
Displacement on axis for one motor rotation (lead) – Unit	Click to set the unit conversion between a value and actual pulse output in the **Position Plan Table**.
Pulses for one motor rotation (resolution) – Numerical Setting	Enter the pulses for one motor rotation for unit conversion purpose.
Default Setting	Click **Default Setting** button to restore the default values on the **Axis Setting** tab.
Decimal Number	**Decimal Number** decides the upper limit of the number of decimal digits in a simulation view when there are decimals in a unit conversion. Options are: • 0~6 ***NOTE**: 1 is the default value.*

Example:

There are *1000* pulse output every *100* units for Axis 1 with the unit of *mm* in the **Position Plan Table**, which displays in the simulation view as shown in the following figure.

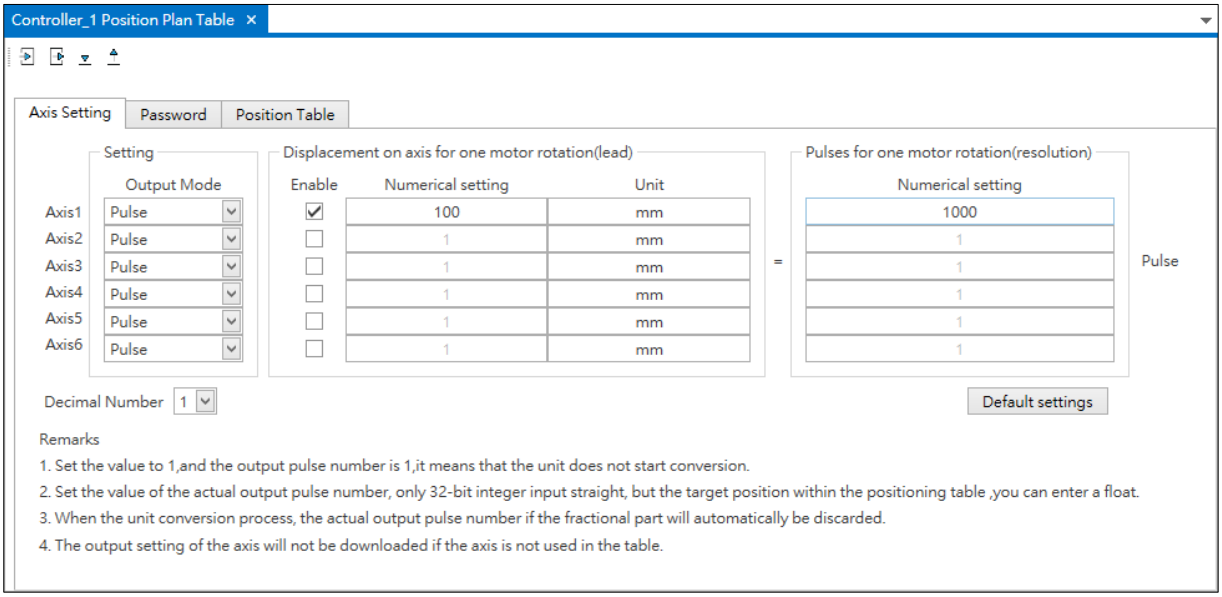


Figure 10 - 2: Position Plan Table – Axis Setting

10.1.4 Password

The **Password** tab is used for protecting the **Position Plan Table** which has been downloaded to the CPU module from being uploaded or read by other people. **Password** tab displays as shown in the following figure.

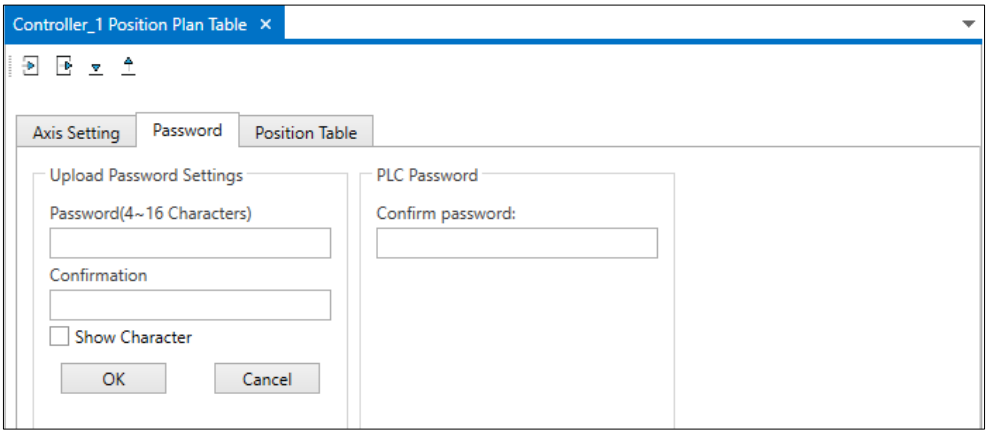


Figure 10 - 3: Position Plan Table - Password

The functions in the **Password** tab are described in the following table:

Function	Description
Upload Password Settings – Password	Enter password to protect the **Position Plan Table** while uploading from PLC. ***NOTE**: 4~16 characters can be provided.*
Upload Password Settings – Confirmation	Enter the above password once again to confirm it.
Upload Password Settings – Show Character	Click to make the characters visible while typing the password.
PLC Password – Confirm Password	Enter the PLC password.
OK	Click to save the settings in **Password** tab.
Cancel	Click to cancel the changes done.

10.1.4.1 Setting a password

Follow these steps to set up password for the **Position Plan Table**:

1. Enter a password in the **Password** box.

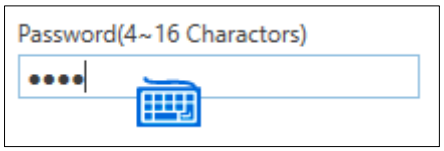


Figure 10 - 4: Position Plan Table – Password

2. Enter the same password in the **Confirmation** box.

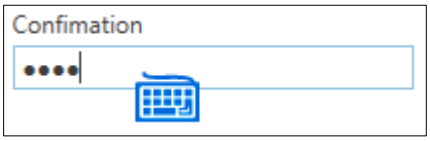


Figure 10 - 5: Position Plan Table - Password Confirmation

3. Click on **OK** button.

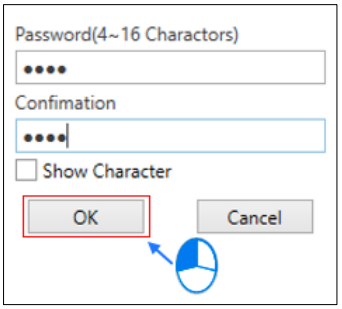


Figure 10 - 6: Position Plan Table - Password Setting

Result: Password is set for the **Position Plan Table**.

10.1.4.2 Cancelling password

Follow these steps to cancel the password set for a **Position Plan Table**:

1. Enter the password in the **Password** box

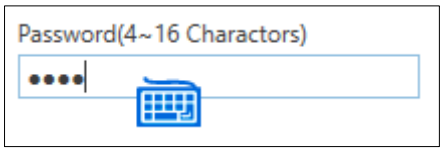


Figure 10 - 7: Position Plan Table - Password for Cancel

2. Click on **OK** button.

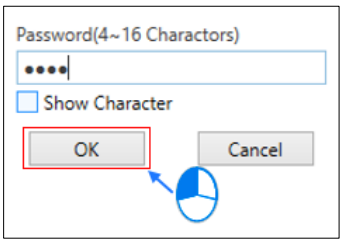


Figure 10 - 8: Position Plan Table - Password Cancel

Result: Password is cancelled for the **Position Plan Table**.

NOTE:

*After the password settings are downloaded to the CPU module, a window appears asking user to type the correct password whenever an upload of the **Position Plan Table** is needed. NOTE that the upload password only protects the **Position Plan Table** in the CPU module from being read by other people. To download a new **Position Plan Table** to the CPU module where there has already been one, a new **Position Plan Table** will overwrite the existing one whether there is an upload password set for the existing one in the CPU module or not.*

10.1.5 Position Table

In the **Position Table** tab, user can add, modify or delete a position plan. **Position Table** tab is shown in the following figure.

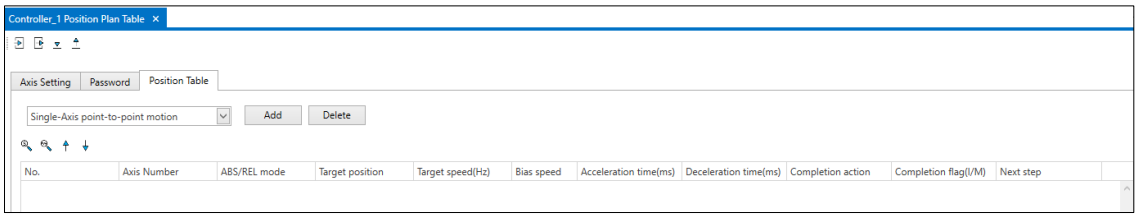


Figure 10 - 9: Position Plan

The functions available in **Position Table** tab are described in the following table:

Function	Description
Position Table mode drop-down list	Select the **Position Table** mode. Options are: • Single-Axis point-to-point motion • Single-Axis multi segment motion • 2-axis linear interpolation • 2-axis arc interpolation ***NOTE**: The **Single-Axis point-to-point motion** is the default value.*
Add	Click to add items.
Delete	Click to delete items.
	Click to perform single-axis simulating mode.
	Click to simulate by selecting item.
	Click to move an item up.
	Click to move an item down.

10.1.5.1 Import Position Plan Table

Follow these steps to import **Position Plan Table**:

1. Click on icon.

   Result: The **Open** window displays as shown in the following figure.

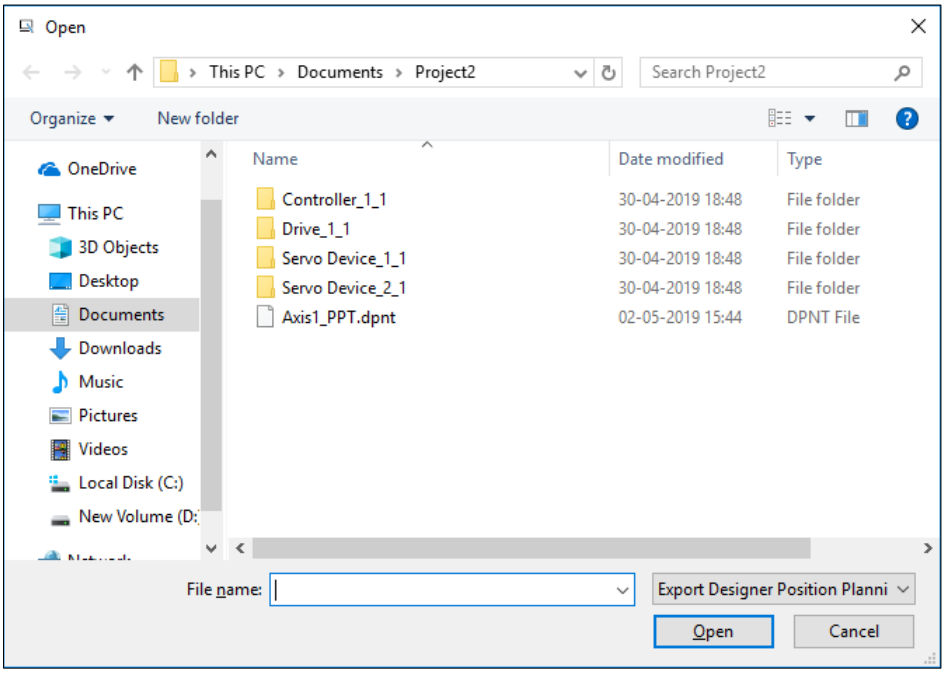


Figure 10 - 10: Open Window – Import Position Plan Table

2. Select the DIADesigner **Position Planning Table** file (*.dpnt) file to import

   ***NOTE**: User can also import the Position Plan Table file (*.pnt) from ISPSoft.*

3. Click on **Open** button.

   Result: **Position Plan Table** is imported

10.1.5.2 Export Position Plan Table

Follow these steps to export **Position Plan Table**:

1. Click on icon.

   Result: The **Save** window displays as shown in the following figure.

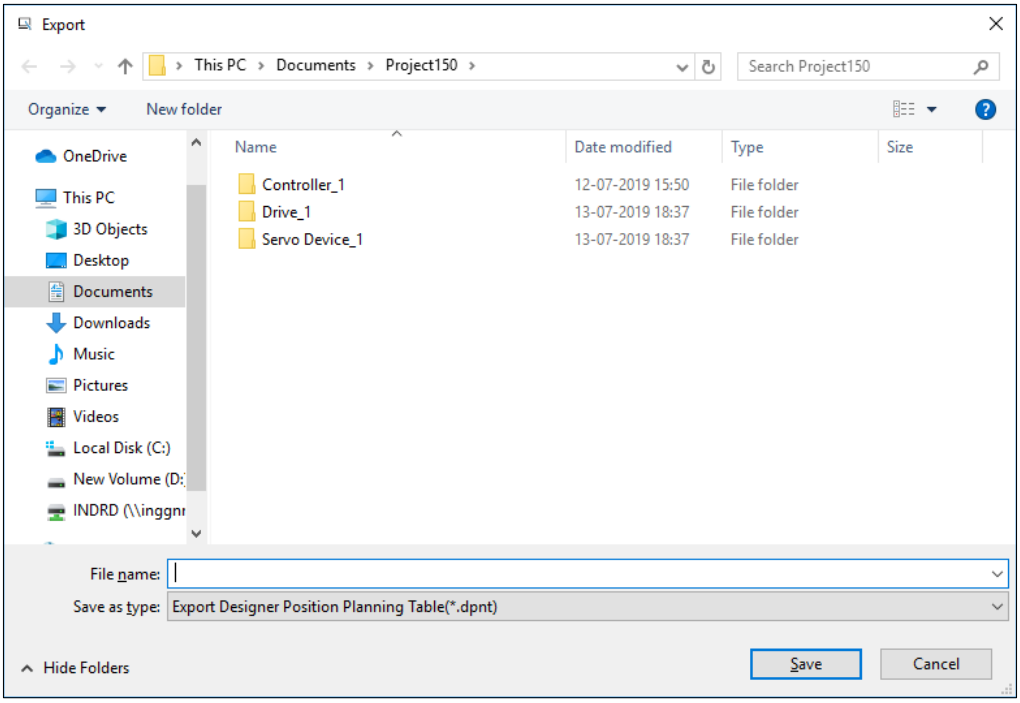


Figure 10 - 11: Export Position Plan Table

2. Enter a name for the *.dpnt file to be created.
3. Click on **Save** button.

   Result: **Position Plan Table** is exported.

10.1.6 Position Plan Table Simulation

After a **Position Plan Table** is set, the path for the table is shown through a graphical simulation. Click any row of the table which is to be simulated and then click or icon. **Position Plane Table Simulation** window displays as shown in the following figure:

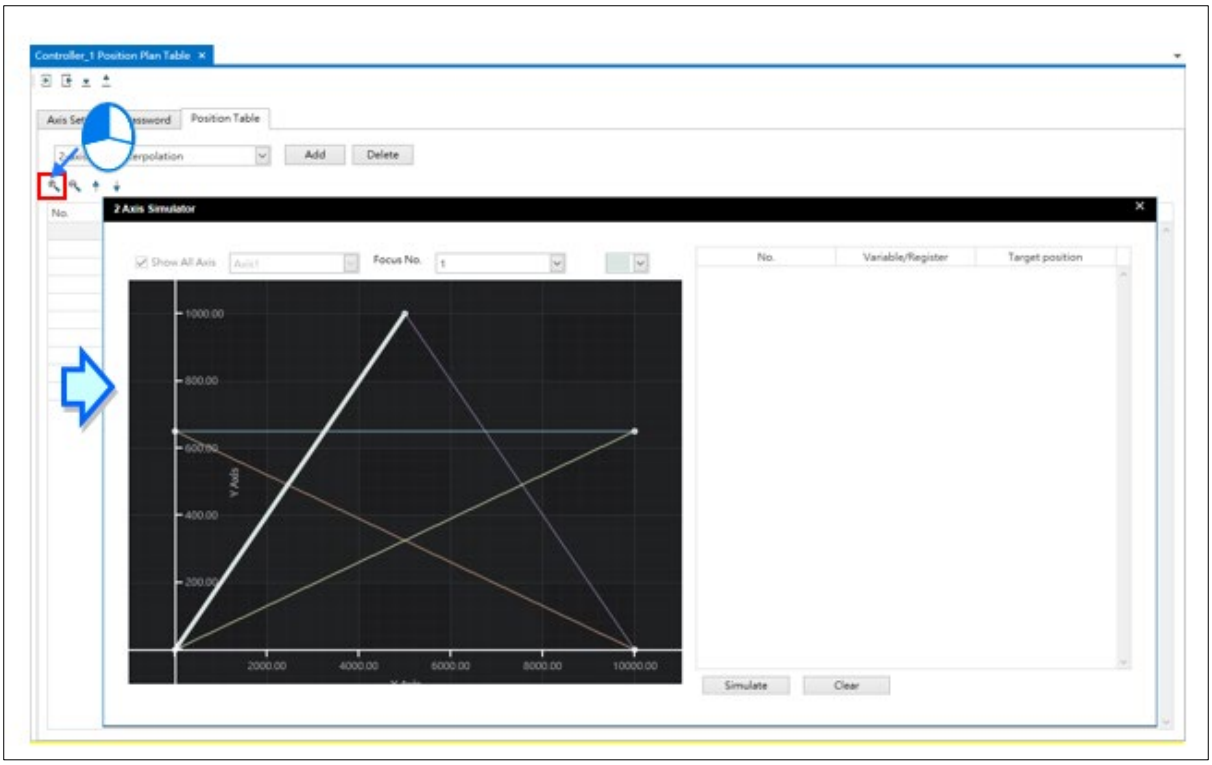


Figure 10 - 12: Position Plan Table Simulation

For the 2-axis linear interpolation and 2-axis arc interpolation, in the mixture of two motion modes, the continuous path is shown in the same simulation view such as that in the above window including the linear interpolation and arc interpolation.

For the single-axis point-to-point motion and single-axis multi-segment motion, only the path of the same motion mode can be shown in the simulation view. The following is the simulation view of the single-axis point to point motion.

The Position Plan Table also supports the simulation starting from a row of the position table in one single mode. First, user must select the row of the table as the start position for the simulation. After the icon is clicked, the simulation window will display a simulation view starting from the selected row of the table until the end.

In the following window, the drawn path starts from number 5, which is part of the linear interpolation in the previous simulation view.

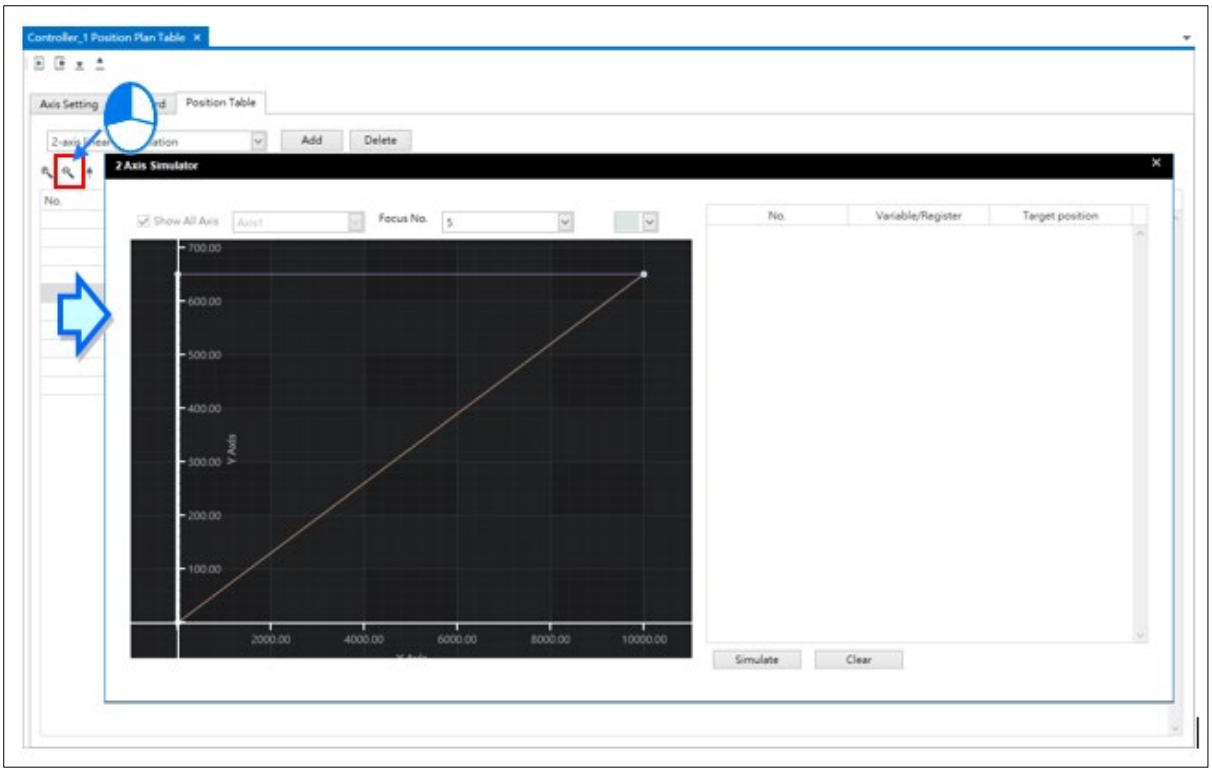


Figure 10 - 13: 2 Axis Simulator

If there are any target position sat by a variable or a register, the variable/register list will display in the right side of simulation window. User can set the value in **Target Position** box of those variables/registers then press **Simulate** to display on the chart in the left side.

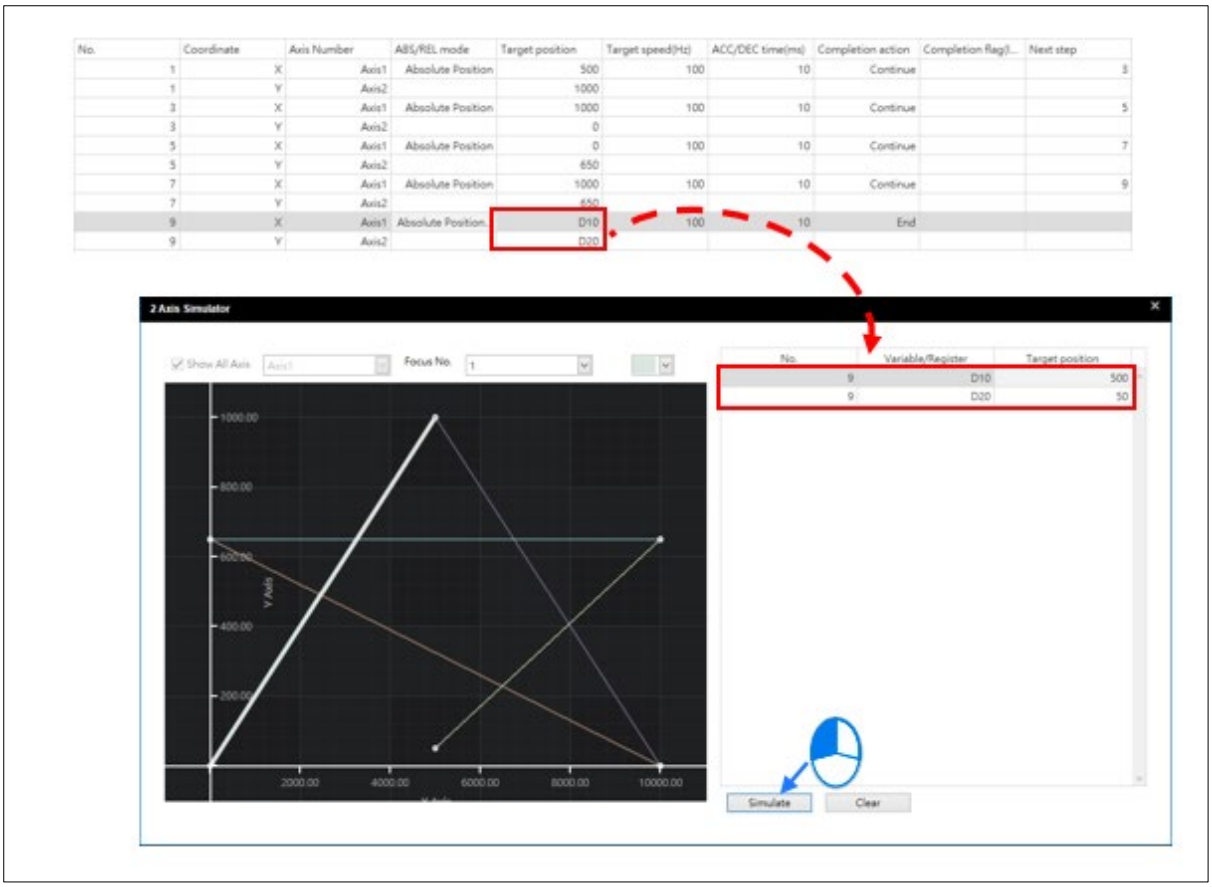


Figure 10 - 14: 2 Axis Simulator

10.1.7 Upload and Download Position Plan Table

When uploading or downloading **Position Plan Table**, user need to connect DIADesigner to a host. If the host is not connected, the following pop-up displays, prompting the user to connect DIADesigner to the host.

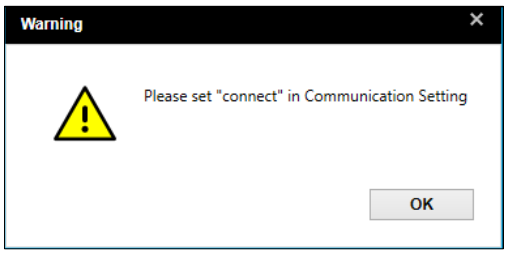


Figure 10 - 15: Warning for Connect

10.1.7.1 Download Position Plan Table

Follow these steps to download **Position Plan Table** to a host:

1. Click icon to download **Position Plan Table** to a host.

   Result: **Position Plan Table** download status **download success** displays as shown in the following figure.

Figure 10 - 16: Position Plan Table - Download success

10.1.7.2 Upload Position Plan Table

Follow these steps to upload **Position Plan Table** from a host:

1. Click icon to upload **Position Plan Table** from a host.

   Result: Pop-up **Position Plan Table Password** displays as shown in the following figure.

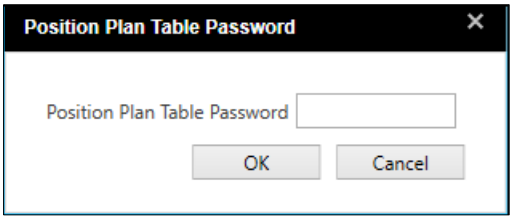


Figure 10 - 17: Position Plan Table Password – Upload

***NOTE**: The **Position Plan Table Password** pop-up displays only if a password is set during previous download! If the password is not set, then DIADesigner will start uploading directly.*

2. Enter the password and click on **OK** button.

   Result: **Position Plan Table** upload status – **upload success!** displays as shown in the following figure.

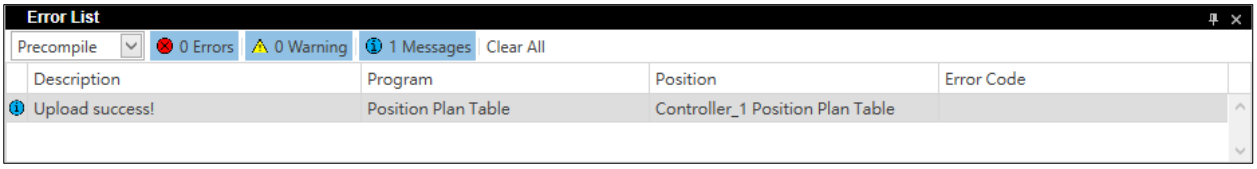


Figure 10 - 18: Position Plan Table - Upload success

3. Click on **OK** to close the pop-up.

10.1.8 Control mode

10.1.8.1 Single-Axis point-to-point motion

User can set the point-to-point positioning control of an axis in this mode. An example of **Single-Axis point-to-point motion** control is shown in the following figure.

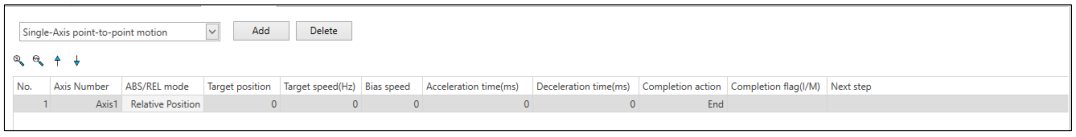


Figure 10 - 19: Single-Axis point-to-point motion

No.: The row number which is created in order automatically by the DIADesigner. Maximum number: **200**.

Axis Number: User can select the axis which is controlled by the **Position Plan Table**.

ABS/REL mode: Select **Relative Position** mode by using the current coordinates to add or reduce coordinate values for position control; or select **Absolute Position** mode to use absolute coordinates as target position for position control. Select **Relative Position (Variable/Address)** mode by using the current coordinates to add or reduce coordinates values (variable/address) for position control; or select **Absolute Position (Variable/Address)** mode to use absolute coordinate and target position (variable/address) for position control.

Target Position: User can type a target coordinate. In the **Position Plan Table**, the **ABS/REL mode** includes **Relative Position (Variable/Address)** or **Absolute Position (Variable/Address)** that requires the use of D register or global variables to input the target position. When using a D register, the data takes up to the length of two D registers.

***Example*:**

If user enter D100, it will take up the length of D100 and D101). When using global variables, the address has to be a variable of D register in DWORD or DINT format which will also take up the length of two D registers!

Target Speed (Hz): User can type a target speed in this field.

Bias Speed: The speed when the positioning starts and ends.

Acceleration time (ms): The time when the **Target Speed** is reached.

Deceleration time (ms): The time when the stop speed is reached.

Completion Action: Selecting **Continue** means to move to the execution of the next row of the table after the present table execution is completed. Selecting **End** means to end the action after the present table execution is completed.

Completion Flag (I/M): The register is set to **ON** when the present table execution is completed. And, I number and or the name of M register can be typed only when **End** in **Completion Action** column is selected. For the interrupt numbers supported, refer application manual. The ten I numbers: - I510 ~ I519 are supported.

Next Step: User can type the number of the next segment of **Position Plan Table**. Blank means completion. The setting is automatically effective after **End** in **Completion Action** column is selected. **Next Step** is not restricted to the same axis

number. NOTE that the same axis number could not have more than two table numbers or position output instructions which are output together.

***NOTE**:*

- *The **Target Speed (Hz)** cannot be set to 0. The **Target Position** also cannot be set to 0 in the **Relative Position** mode.*
- *I number can be used more than once. User must determine through which row number the I number can be completely output.*
- *The instruction which the start table outputs is API2718 TPO. API2719 TPWS can be used to modify the output parameters while the PLC is running.*

10.1.8.2 Single-Axis multi-segment motion

In **Single-Axis multi-segment motion** control mode, user can control the positioning of one axis and make every-segment motion followed by the next-segment motion at the target speed. Or abort the segment and to perform the next-segment motion. An example of **Single-Axis multi-segment motion** control is shown in the following figure.

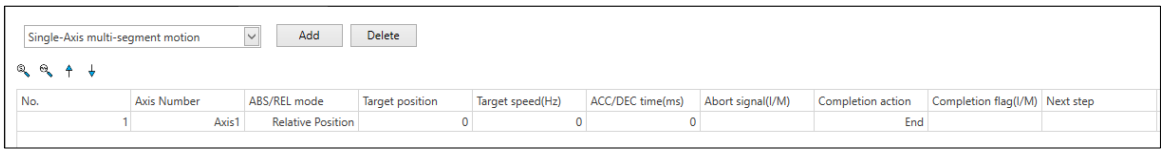


Figure 10 - 20: Single-Axis multi-segment motion

No.: The row number created in order automatically. Maximum number: **200**.

Axis Number: User can select the axis which is controlled by the **Position Plan Table**.

ABS/REL mode: Select **Relative Position** mode by using the current coordinates to add or reduce coordinate values for position control; or select **Absolute Position** mode to use absolute coordinate as target position for position control. Select **Relative Position (Variable/Address)** mode by using the current coordinates to add or reduce coordinate values (variable/address) for position control; select **Absolute**

Position (Variable/Address) mode to use absolute coordinate and target position (variable/address) for position control.

Target position: User can type a target coordinate. In the Position Plan Table, the **ABS/REL mode** includes **Relative Position (Variable/Address)** or **Absolute Position (Variable/Address)** that requires the use of D register or global variables to input the target position. When using a D register, the data takes up to the length of two D registers.

***Example*:**

If user enter D100, it will take up the length of D100 and D101). When using global variables, the address has to be a variable of D register in DWORD or DINT format which will also take up the length of two D registers!

Target Speed (Hz): User can type a target speed in this field.

ACC/DEC time (ms): The time when the **Target Speed** is reached and stop speed is reached.

Abort Signal (I/M): The next segment of Position Plan Table is performed if the register of interrupt signals is set to **ON** before the current table is completed. And, I number or the name of M register can be typed. For the interrupt numbers supported, refer product programming manual.

Completion Action: Selecting **Continue** means to move to the execution of the next row of the table after the present table execution is completed. Selecting **End** means to end the action after the present table execution is completed.

Completion Flag (I/M): The register is set to **ON** when the present table execution is completed. And, I number and or the name of M register can be typed only when **End** in **Completion Action** column is selected. For the interrupt numbers supported, refer application manual. The ten I numbers: - I510 ~ I519 are supported.

Next Step: User can type the number of the next segment of Position Plan Table. Blank means completion. The setting is automatically effective after **End** in **Completion Action** column is selected. **Next Step** is not restricted to the same axis number. NOTE that the same axis number could not have more than two table numbers or position output instructions which are output together.

***NOTE**:*

- *The **Target Position** cannot be set to 0 in the relative mode. The **Target Position** of two adjacent row numbers cannot be set to the same value in the **Absolute Position** mode.*
- *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/deceleration.*
- *I number can be used more than once. User must determine through which row number the I number can be completely output.*
- *The instruction which the start table outputs is API2718 TPO. API 2719 TPWS can be used to modify the output parameters while the PLC is running.*

10.1.8.3 2-axis Linear Interpolation motion

User can set the linear interpolation motion of two axes on the XY plane. Two rows of the **Position Plan Table** are occupied when this mode is used. **2-axis Linear Interpolation motion** control is shown in the following figure.

Figure 10 - 21: 2-axis linear interpolation

No.: The row number is automatically created orderly in the input area. It will take up two rows of the table for 2-axis motion mode. In other words, when in 2-axis mode, there are a total of 100 groups in the table and the maximum number: **100**.

Coordinate: The axis which X and Y axes coordinate stand for in the simulation view.

Axis Number: User can select the axis which is controlled by the **Position Plan Table**.

ABS/REL mode: Select **Relative Position** mode by using the current coordinates to add or reduce coordinate values for position control; or select **Absolute Position** mode to use absolute coordinate as target position for position control. Select **Relative Position (Variable/Address)** mode by using the current coordinates to add or reduce coordinate values (variable/address) for position control; select **Absolute Position (Variable/Address)** mode to use absolute coordinate and target position (variable/address) for position control.

Target Position: User can type a target coordinate. In the Position Plan Table, the **ABS/REL mode** includes **Relative Position (Variable/Address)** or **Absolute Position (Variable/Address)** that requires the use of D register or global variables to input the target position. When using a D register, the data takes up to the length of two D registers.

***Example*:**

If user enter D100, it will take up the length of D100 and D101). When using global variables, the address has to be a variable of D register in DWORD or DINT format which will also take up the length of two D registers!

Target Speed (Hz): User can type a target speed in this field.

ACC/DEC time (ms): The time when the **Target Speed** is reached and Stop Speed is reached.

Completion Action: Selecting **Continue** means to move to the execution of the next row of the table after the present table execution is completed. Selecting **End** means to end the action after the present table execution is completed.

Completion Flag (I/M): The register is set to **ON** when the present table execution is completed. And, I number and or the name of M register can be typed only when **End** in **Completion Action** column is selected. For the interrupt numbers supported, refer application manual. The ten I numbers: - I510 ~ I519 are supported.

Next Step: User can type the number of the next segment of Position Plan Table. Blank means completion. The setting is automatically effective after **End** in **Completion Action** column is selected. **Next Step** is not restricted to the same axis number. NOTE that the same axis number could not have more than two table numbers or position output instructions which are output together.

***NOTE**:*

- *The PLC will automatically run at the set speed when the **Target Speed** or the **ACC/DEC time** is set to 0. The **Target Position** cannot be set to 0 in the **Relative Position** mode. The **Target Position** of two adjacent row numbers cannot be set to the same value in the **Absolute Position** mode.*
- *I number can be used more than once. User must determine through which row number the I number can be completely output.*
- *The instruction which the start table outputs is API2718 TOP. API2720 TPWL can be used to modify the output parameters while the PLC is running.*

10.1.8.4 2-axis Arc Interpolation motion

User can set the Arc Interpolation motion of two axes on the XY plane. Two rows of Position Plan Table are occupied by using the **2-axis Arc Interpolation** control mode every time. Sample **2-axis Arc Interpolation** motion control mode is shown in the following figure.

Figure 10 - 22: 2-axis Arc interpolation

No.: The row number is automatically created orderly in the input area. It will take up two rows of the table for 2-axis motion mode. In other words, when in 2-axis mode, there are a total of 100 groups in the table and the maximum number is 100.

Coordinate: The axis which X and Y axes coordinate stand for in the simulation view.

Axis Number: User can select the axis which is controlled by the Position Plan Table.

ABS/REL mode: Select **Relative Position** mode by using the current coordinates to add or reduce coordinate values for position control; or select **Absolute Position** mode to use absolute coordinate as target position for position control. Select **Relative Position (Variable/Address)** mode by using the current coordinates to add or reduce coordinate values (variable/address) for position control; select **Absolute Position (Variable/Address)** mode to use absolute coordinate and target position (variable/address) for position control.

Target Position: User can type a target coordinate. In the Position Plan Table, the **ABS/REL mode** includes **Relative Position (Variable/Address)** or **Absolute Position (Variable/Address)** that requires the use of D register or global variables to input the target position. When using a D register, the data takes up to the length of two D registers.

***Example*:**

If user enter D100, it will take up the length of D100 and D101). When using global variables, the address has to be a variable of D register in DWORD or DINT format which will also take up the length of two D registers!

Tangential Speed (Hz): Movement in a circular path in three-dimensions and the magnitude of that vector is the tangential speed.

Center Shift value (Pulse): 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 figure. From point A to point B, a clockwise rotation will generate an arc less than 180^0 if S>0 and an arc more than 180^0 is S<0.

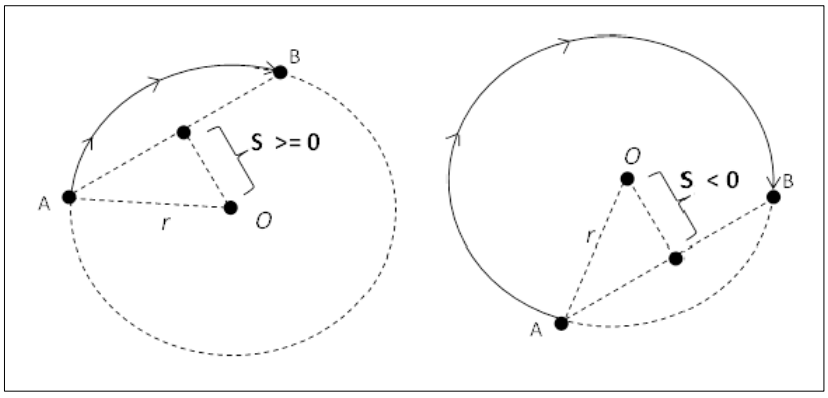


Figure 10 - 23: Clockwise Rotation

Direction: User can select **Clockwise** motion or **Counterclockwise** motion.

Completion Action: Selecting **Continue** means to move to the execution of the next row of the table after the present table execution is completed. Selecting **End** means to end the action after the present table execution is completed.

Completion Flag (I/M): The register is set to **ON** when the present table execution is completed. And, I number and or the name of M register can be typed only when **End** in **Completion Action** column is selected. For the interrupt numbers supported, refer application manual. The ten I numbers: - I510 ~ I519 are supported.

Next Step: User can type the number of the next segment of **Position Plan Table**. Blank means completion. The setting is automatically effective after **End** in **Completion Action** column is selected. **Next Step** is not restricted to the same axis number. NOTE that the same axis number could not have more than two table numbers or position output instructions which are output together.

***NOTE**:*

- *The PLC automatically calculates the output speed in this function. The **Target Position** cannot be set to 0 in the **Relative Position** mode. The **Target Position** of two adjacent row numbers cannot be set to the same value in the **Absolute Position** mode.*
- *I number can be used more than once. User must determine through which row number the I number can be c completely output.*
- *The instruction which the start table outputs is API2718 TPO. API2721 TPWL can be used to modify the output parameters while the PLC is running.*

Chapter 11: HMI Interaction

11.1 Tag Sharing to DIAScreen

Tag means the variable that is inherited between in DIADesigner & DIAScreen. User can share the Variables with **Address** created in DIADesigner with DIAScreen application, so that user can directly access those Variables in DIAScreen via **DIA Tag** function.

***NOTE**:*

*Only **DOP-100 HMI series** supports **DIA Tag** sharing feature in the current version.*

Follow these steps to perform tag sharing to DIAScreen:

1. Create variables with address in any global variable table of **Controller** > **Programming** > **Global Variable**.

   An example for Tag Sharing is shown in the following figure.

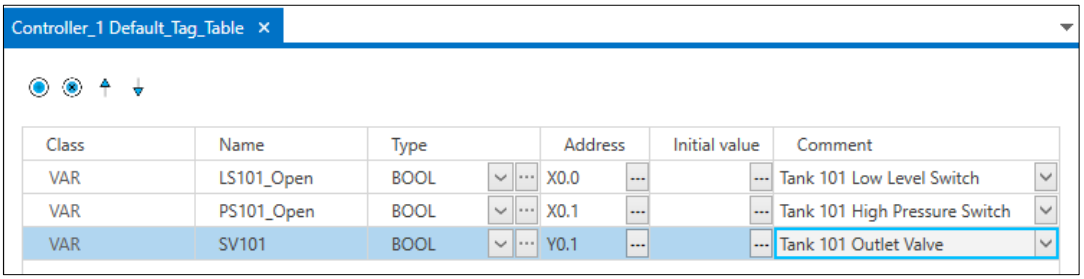


Figure 11 - 1: Tag Sharing

2. Compile the DIADesigner device or project.
3. Launch DIAScreen.
4. Create a HMI project with DOP-100 HMI series. During **Project** creation, enter a **Link Name**, select **Delta** in **Manufacturers** drop-down list, select **Delta AS series PLC** in the **series** drop-down list as shown in the following figure.

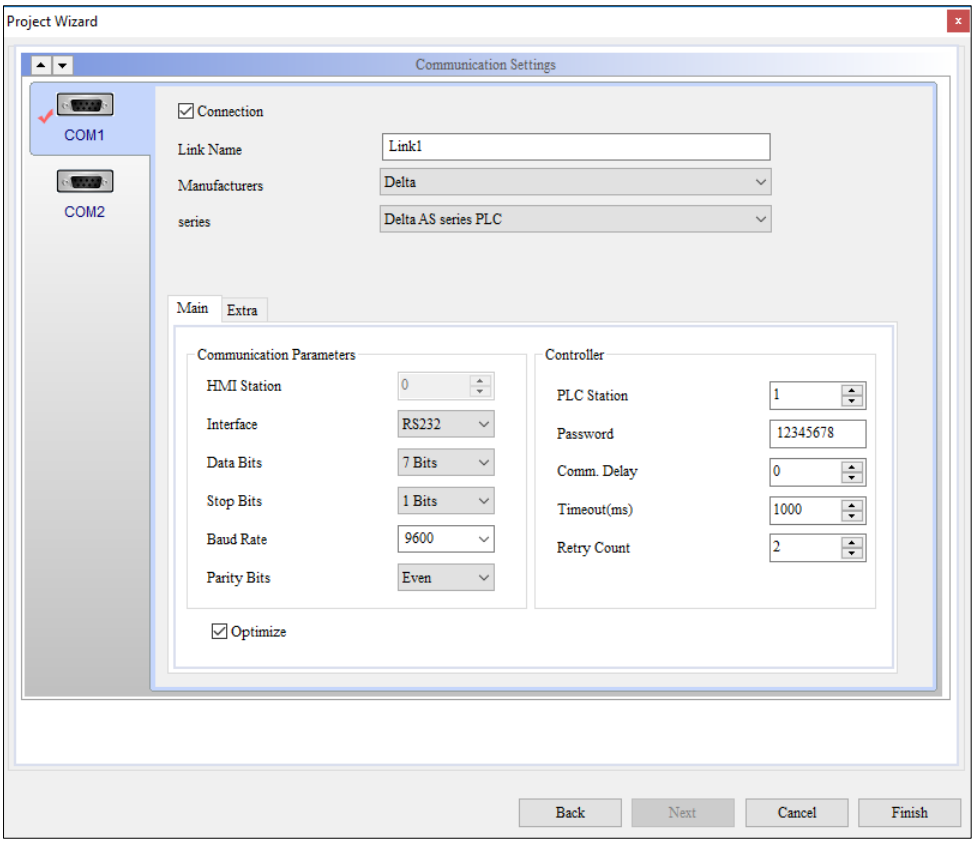


Figure 11 - 2: Project Wizard

Result: The DIAScreen project is created.

5. Double-click **DIA Tag** function in the **Project Explorer** in DIAScreen.

   Result: **DIA Tag** window displays as shown in the following figure.

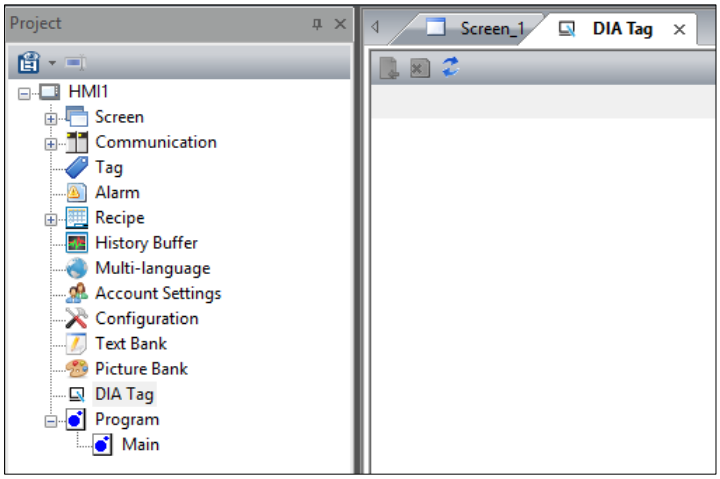


Figure 11 - 3: DIA Tag

6. Click icon in the **DIA Tag** window.

   Result: The **Device Name List** window displays as shown in the following figure.

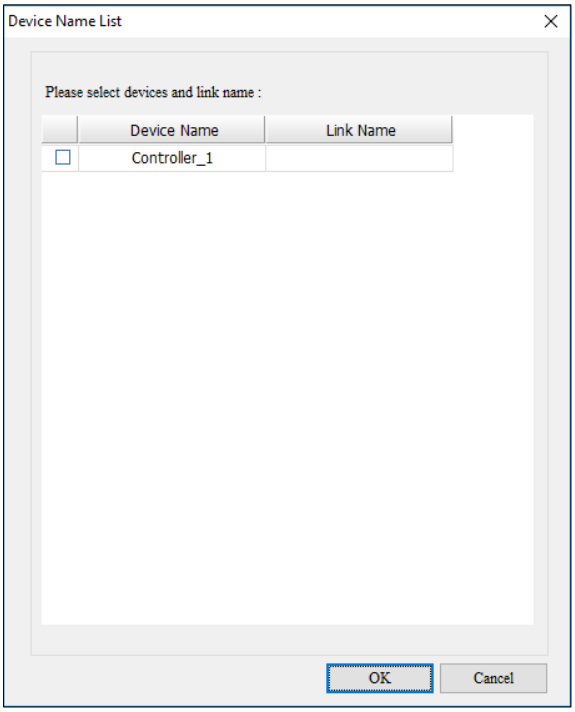


Figure 11 - 4: Device Name List

7. Select the Controllers in the **Device Name** and select **Link Name**.

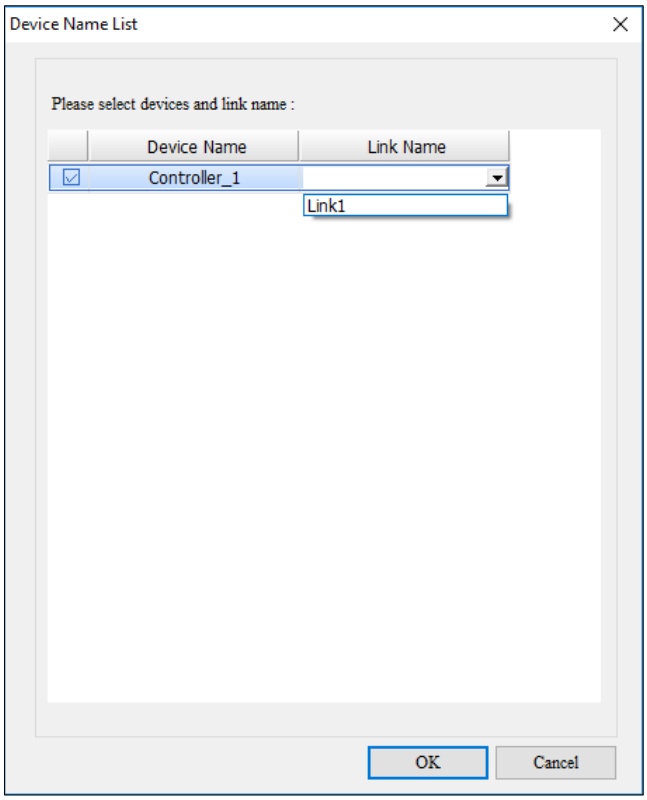


Figure 11 - 5: Device Name List > Link Name

8. Click on **OK**.

   Result: DIADesigner tags are now shared with DIAScreen so that the Variables can be used for different kinds of elements, animations, etc. as shown in the following figure.

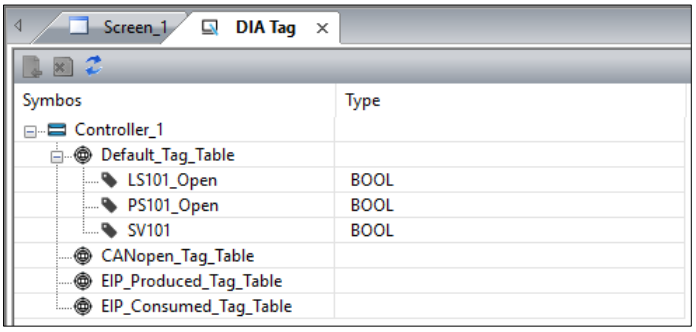


Figure 11 - 6: DIA Tag updated with Tag sharing

Example:

User can select **DIA Tag** function to select the DIADesigner variables as shown in the following figure:

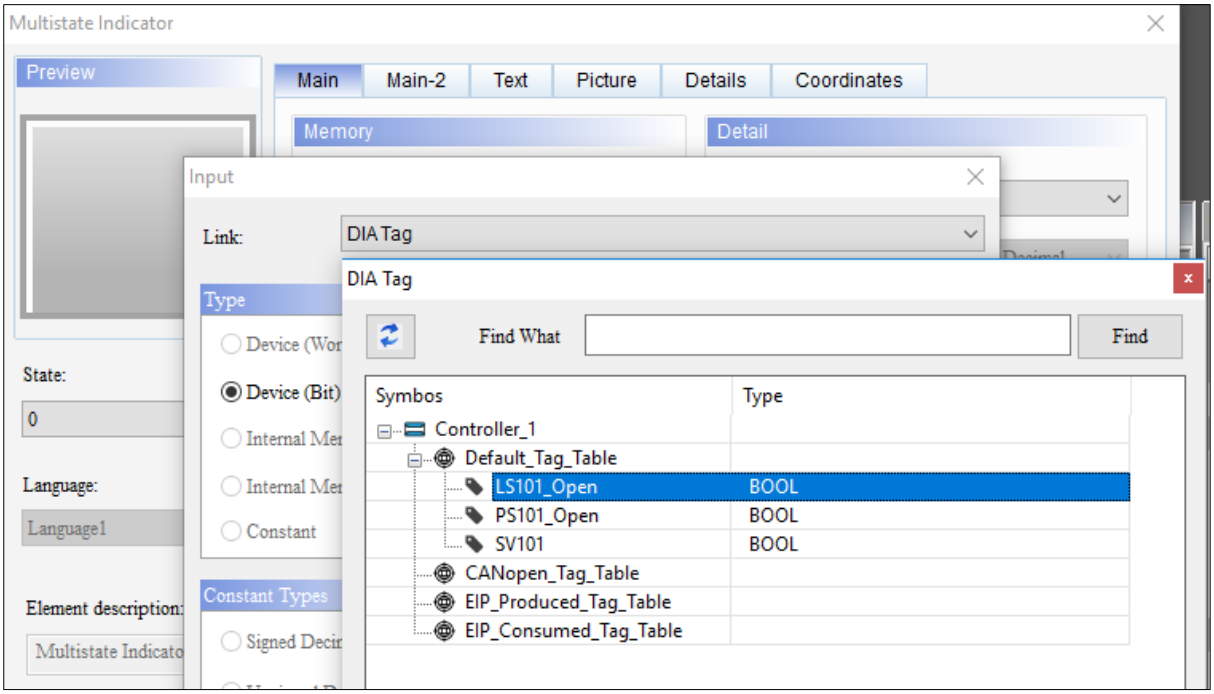


Figure 11 - 7: DIA Tag usage

Chapter 12: Simulation

12.1 PLC Simulation

In the current version of DIADesigner, AS200 and AS300 PLC Simulation is supported.

***NOTE**: Refer 5.3.4.7 Simulator Setting for more information.*

Follow these steps to perform PLC Simulation for AS200/AS300 PLCs:

1. Create a Communication Driver of AS200 Simulator /AS300 PLC Simulator in **COMMGR** and **Start** the driver.

   ***NOTE**: Refer 5.3.4.7 Simulator Setting for more information.*

2. Add AS200/AS300 devices as per user requirements in DIADesigner.
3. Create programs, variables and so on, as per user requirement.
4. Compile the device.

   ***NOTE**: Refer 8.4 Compile and Check for more information*.

5. Double-click on **Controller** > **Communication Setting** in **Project Explorer** of DIADesigner.

   Result: Displays the **Communication Setting** window.

   ***NOTE**: Refer 5.1 Communication via COMMGR for more information.*

6. Select the *Simulation Driver* created in Step (1) in the **Driver Name** field in **Communication Setting** window
7. Click **Connect**.

   Result: The **Connected** status displays in **Communication Settings** window.

8. Perform **Download** operation.

   ***NOTE**: Refer 12.1 Batch Download and Upload for more information.*

9. Go **Online** with the device.

Result: User can open the POU or Monitor Table or other options and perform the monitoring, etc.

***NOTE**: Refer 12.2 Online / Offline Mode for upload/download and online functions.*

Chapter 13: Download / Upload and Online / Offline Mode

13.1 Batch Download and Upload

User can download/upload the programs, functions or parameters in projects from/to multiple devices at one time in DIADesigner.

13.1.1 Download

Follow these steps to download the project from PC to devices.

1. Compile PLC models or inverters with built-in PLC. Other field devices do not need to be compiled.

   ***NOTE**: Compile must be successful without any errors.*

2. Select the communication **Driver** in **Device** > **Communication Setting** window.

3. Select the device and click icon from **development toolbar** or

   Right-click the device to open its context menu and select **Download** or

   Select the device and click **Operation** > **Project Download** in menu bar or

   Select the device and press **Ctrl** + **L** keys simultaneously on the keyboard.

   Result: The **Upload/Download Manager** window displays as shown in the following figure.

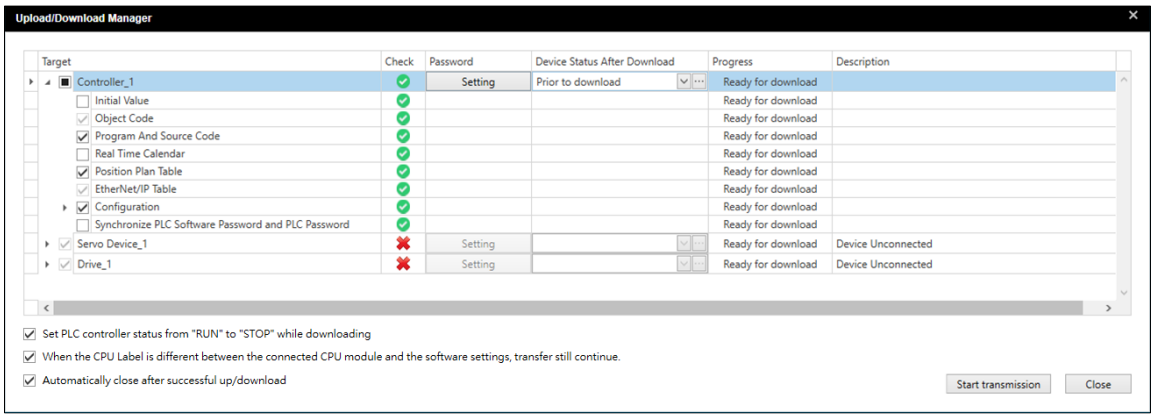


Figure 13 - 1: Upload/Download Manager

The functions in the **Upload/Download Manager** window are explained in the following table:

Function	Description
Target	Displays the target device name as defined in the project in the form or a tree node. Components (functions) display below the device name (depends on device).
Check	Displays whether the download check is passed ( ) or failed ( ).
Password	Click **Setting** to display the **Password** window.
Device Status after Download	Select the device status intended after download action. Options are: • Prior to Download • Run • Stop ***NOTE**: **Prior to download** is the default value.*
Progress	Display the progress. **Example**: Ready for Download, % progress and so on.
Description	Displays the description of the device connection status.
Set the PLC controller status from “RUN” to “STOP” while downloading	If any of PLC is RUN mode, user can check this box to switch to STOP mode when download to those PLC.
When the CPU label is different between the connected CPU module, and the Software settings, transfer still continue	Check to continue download even if the PLC label is different.
Automatically close after successful upload/download	Check to close the **Upload/Download Manager** dialog after successful transformation.
Start Transmission	Click to start upload/download.
Close	Click to close the **Upload/Download Manager** window.

4. Select the components to be downloaded.
5. Click on **Start Transmission**.

   Result: Project or components selected are downloaded to the device.

13.1.2 Upload

Follow these steps to Upload or transfer project from PLC to PC:

1. Create a device or select the device whose Program or components needs to be uploaded.
2. Select the Communication Driver in **Device** > **Communication Setting**.
3. Select the device and click on icon from **development toolbar**

   or

   Right-click on the device to open its context menu and select **Upload**

   or

   Select the device and click **Operation** > **Project Upload** in **Menu** bar

   or

   Select the device and press **Ctrl** + **U** keys simultaneously on the keyboard.

   Result: **Confirmation** window displays. Click **Yes**, to close current project, and upload data to a new project. Click **No**, to upload data to current project. The **Upload/Download Manager** displays

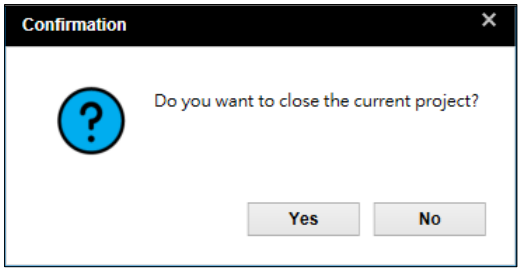


Figure 13 - 2: Confirmation to Upload

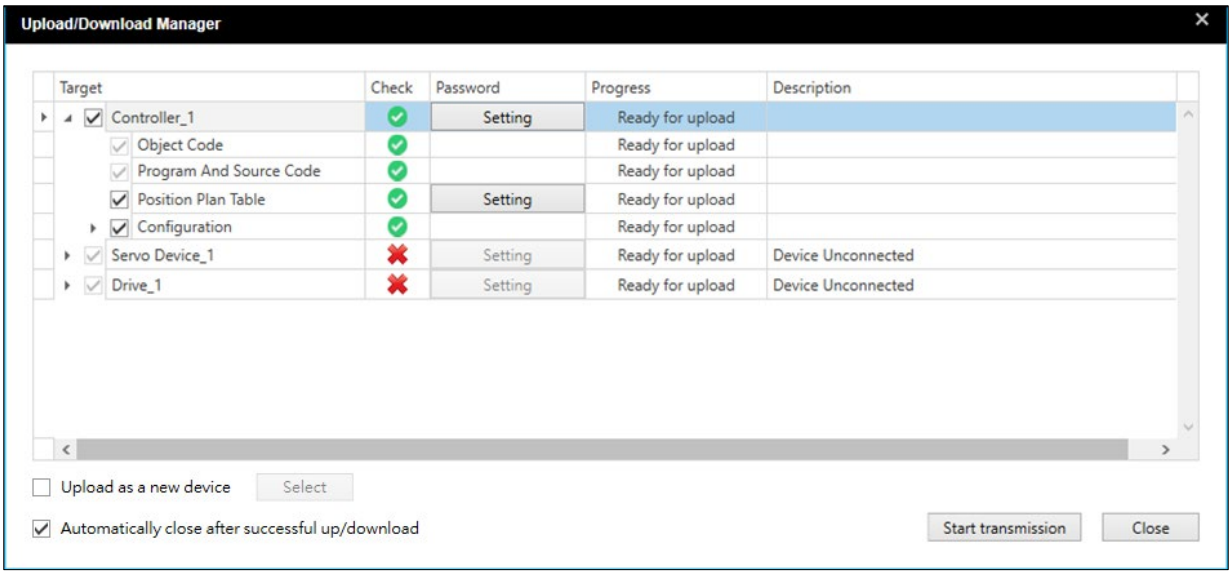


Figure 13 - 3: Upload/Download Manager - Upload

The functions in **Upload/Download Manager** functions during **Upload** are explained in the following table:

Function	Description
Target	Displays the target device name as defined in the project in the form or a tree node. Components (functions) display below the device name (depends on device).
Check	Displays whether the upload check is passed ( ) or failed ( ).
Password	Click **Setting** to display the **Password** window.
Progress	Display the progress **Example**: Ready for Upload, % progress and so on.
Description	Displays the description.
Upload as a new device	Check to add a new device to the project for uploading.
Automatically close after uploading / downloading	Check it to automatically close "Upload and Download Manager" after uploading/downloading.
Start Transmission	Click to start upload.
Cancel	Click to cancel upload.

4. Select the components to be uploaded.
5. Click **Start Transmission**.

   Result: Project or components selected are uploaded to the PC.

13.1.3 Download/Upload Components

The items in the **Upload/Download Manager** window vary according to the device or series selected. The items, which are not editable, are grayed out.

Transmission items in the **Upload/Download Manager** are explained in the following table:

Item	Description
Initial Value	**Initial Value** of **Variables**.
Object Code	Object Code is what a PC compiler produces. ***NOTE**: This item is selected by default*.
Program and Source Code	IEC source code: the programming language specified by IEC61131-3 C source code: a program edited in C language
Real time Calendar	Real Time Calendar (RTC) of PC/device.
Position Plan Table	Position Plan Table created by user.
EtherNet/IP Table	EtherNet/IP Table.
Configuration	**Hardware Configuration** settings created by user.
Synchronize PLC Software Password and PLC Password	During the transmission of the project, the project password and the PLC password become the same.

13.2 Online / Offline Mode

User can perform the addition, deletion, modification of the specified hardware, programs, variables, address assignment and so on, in the **Offline** mode and edit the project/device. After editing is complete, user can compile and generate the object code which is to be downloaded to PLC. At last, the programs and settings will be written to the target device by downloading.

When a controller or device is not connected to the DIADesigner, the status bar of the software will display **Offline** in its status area as shown in the following figure. By default, the operation status of DIADesigner is **Offline** mode.

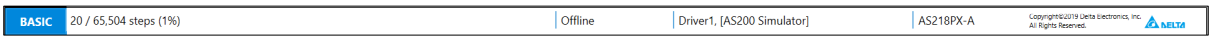


Figure 13 - 4: Status bar - Offline mode

After connecting to devices, user can enter **Online** mode to check the status of devices.

Follow any of these methods to enter **Online** mode:

- Click icon on the **Development** toolbar or
- Click **Operation** > **Online** in Menu bar or
- Right-click on the device to open its context menu and click **Online** or
- Press **Ctrl** + **F4** keys simultaneously on the keyboard.

  Result: DIADesigner enters **Online** mode for the device connected as shown in the following figure.

Figure 13 - 5: Status bar - Online mode

Chapter 14: Commission and Diagnostic

14.1 Device Information

If DIADesigner is connected to a device, the system will retrieve information from the device after user navigate to **Controller** > **Device & Error Information** > **Device Information** in the **Project Explorer**. The **Device Information** tab displays by default. System information is different for different product series.

An example of **Device Information** for AS300 PLC & MS300 inverters are shown in the following figure.

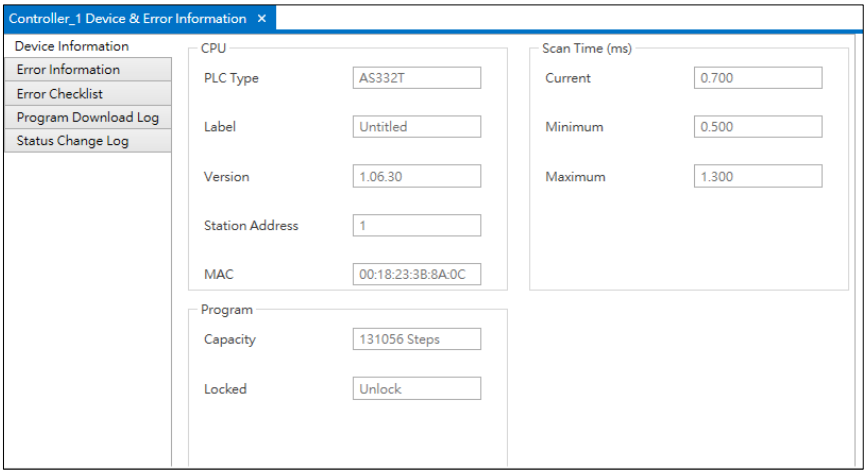


Figure 14 - 1: Device and Error Information tab

An example of **Device Information** for MS300 inverter is shown in the following figure.

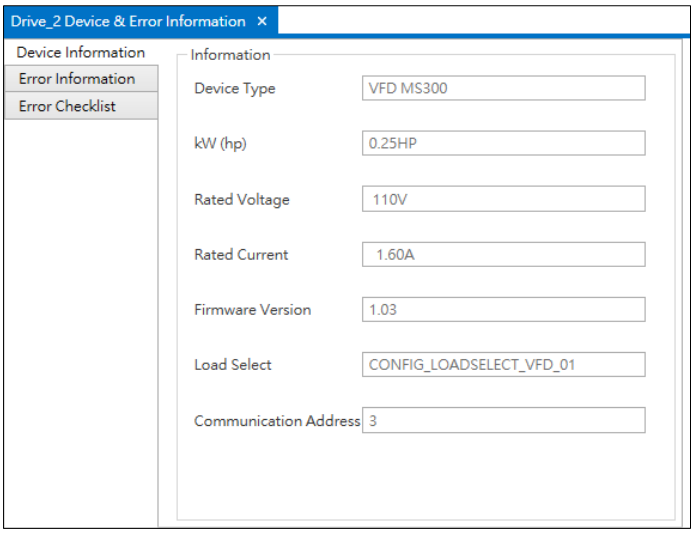


Figure 14 - 2: Device Information - Drive

14.2 Monitor Table

User can check the status and values of Variables – Global Variables or POU Local Variables and Registers during testing and commissioning with the help of **Monitor Table** in DIADesigner. User can locate **Monitor Table** in DIADesigner in **Controller** > **Commissioning** > **Monitor Table**.

A sample **Monitor Table** is shown in the following figure:

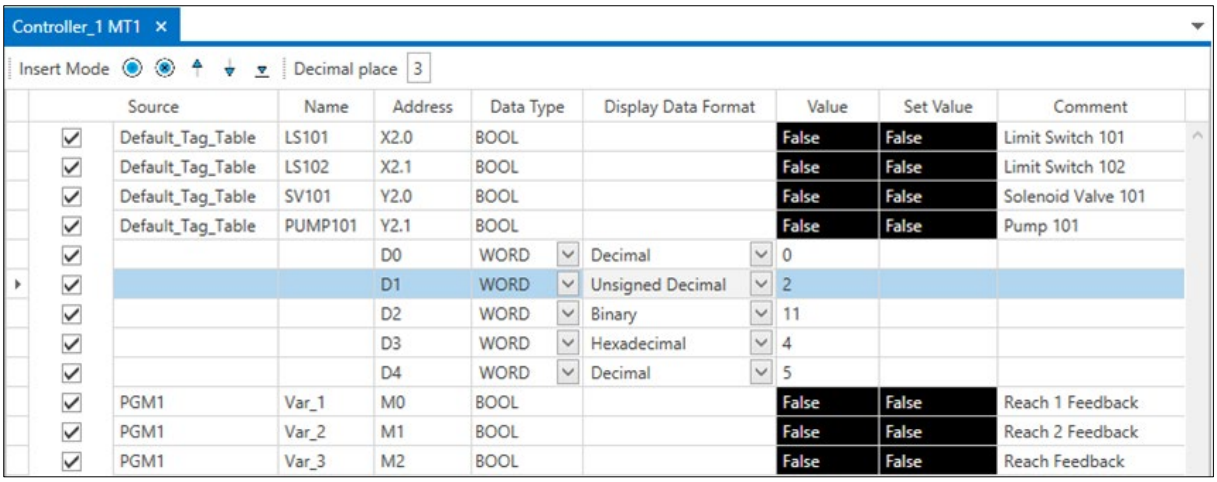


Figure 14 - 3: Monitor Table

The functions in the **Monitor Table** are described in the following table:

Function	Description
Source	Displays the source of the variable.
Name	Displays the name of the variable.
Address	Displays the address of the variable/register.
Data Type	Displays the datatype of the variable/register.
Display Data Format	User can select a format in which the register value is represented. Possible options are: • Decimal • Unsigned Decimal • Binary • Hexadecimal • ASCII • Floating point • Scientific symbols ***NOTE**: The default value is **Decimal**.*
Value	Displays the value of the variable/register in Online mode.
Set Value	User can enter the value to be written to the variable in Online mode.
Comment	Displays the comment of the variables/address.

14.2.1 Creating a Monitor Table

Follow these steps to create a **Monitor Table**:

1. Click on **Controller** > **Commissioning** > **Monitor Table**.
2. Right-click on the **Monitor Table** to open its context menu and select **Add Monitor Table** as shown in the following figure.

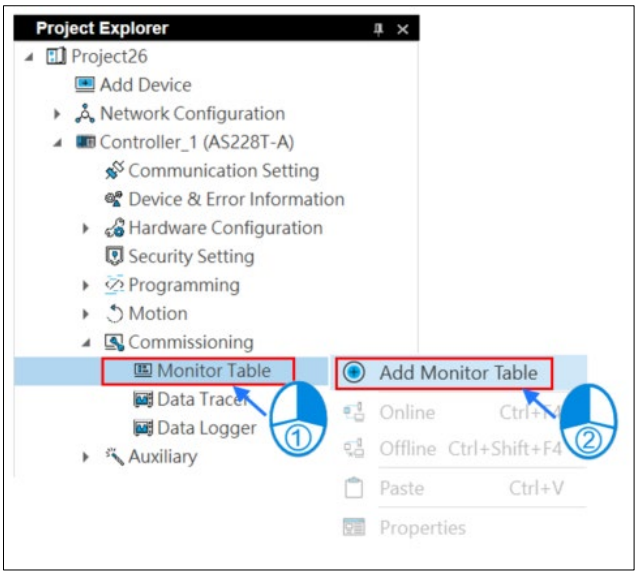


Figure 14 - 4: Monitor Table in Project Explorer

Result: The **Add Monitor Table** window displays as shown in the following figure.

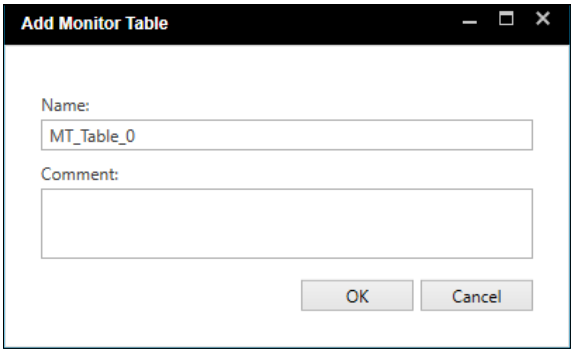


Figure 14 - 5: Add Monitor Table

3. Enter a name in the **Name** field and a comment for this **Monitor Table** in the **Comment** field (optional).

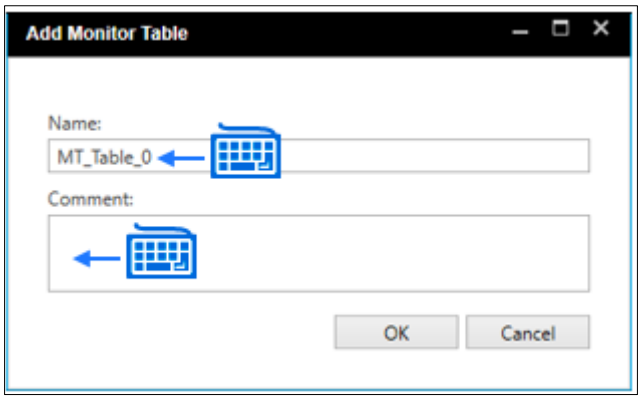


Figure 14 - 6: Add Monitor Table - edit

4. Click on **OK** button.

   Result: A **Monitor Table** is created as shown in the following figure.

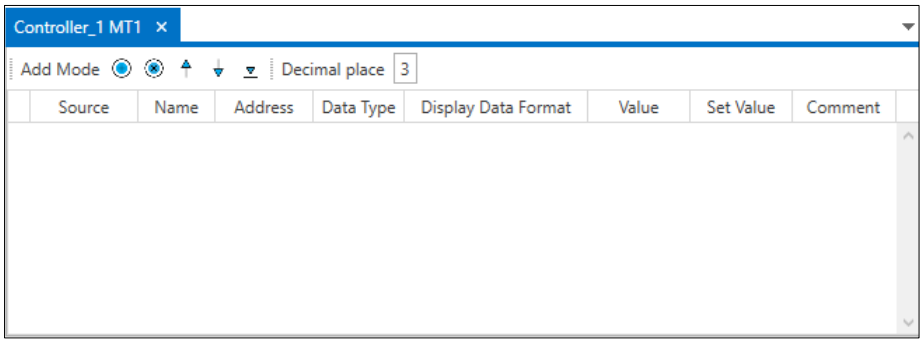


Figure 14 - 7: Monitor Table

***NOTE**:*

*When the **Monitor Table** is created, by default it displays in the DIADesigner work area as the sample shown in the following figure. If the window is closed, user can double-click on the **Monitor Table** in the **Project Explorer** to open the **Monitor Table** window. User can create several Monitor Tables in the Project.*

The functions in the context menu of **Monitor Table** in **Project Explorer** are described in the following table:

Function	Description
Add Monitor Table	Click to open the Add Monitor Table window to add a new monitor table
Online	Set the device as online mode
Offline	Set the device as offline mode
Paste	Paste the monitor table
Properties	Click to open the Properties window

The following table shows the detailed descriptions of the context menu of each monitor table node.

Function	Description
Cut	Cut the selected monitor table
Copy	Copy the selected monitor table
Paste	Paste copied/cut monitor table
Delete	Delete monitor table
Online	Set the device as online mode
Offline	Set the device as offline mode
Properties	Click to rename the monitor table and edit comments

The toolbar of **Monitor Table** is shown in the following figure.

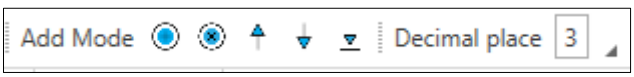


Figure 14 - 8: Monitor Table - Toolbar

The functions in the **Monitor Table Toolbar** are described in the following table:

Function	Description
Add Mode	Click to select **Add mode** so that the entry can be added in the end.
Insert Mode	Click to select **Insert mode** so that the entry can be added in between the existing table lines.
	Click to open **Source selection** window to add Variables/Registers.
	Click to delete the selected items.
	Click to move an item above its present level.
	Click to move an item below its present level.
	Click to download **Monitor Table**.
Decimal place 3	Enter decimal places.

14.2.2 Adding Variable and Registers

Click icon to open the **Source selection** window to add **Variables** and **Registers**. The **Source Selection** window displays as shown in the following figure.

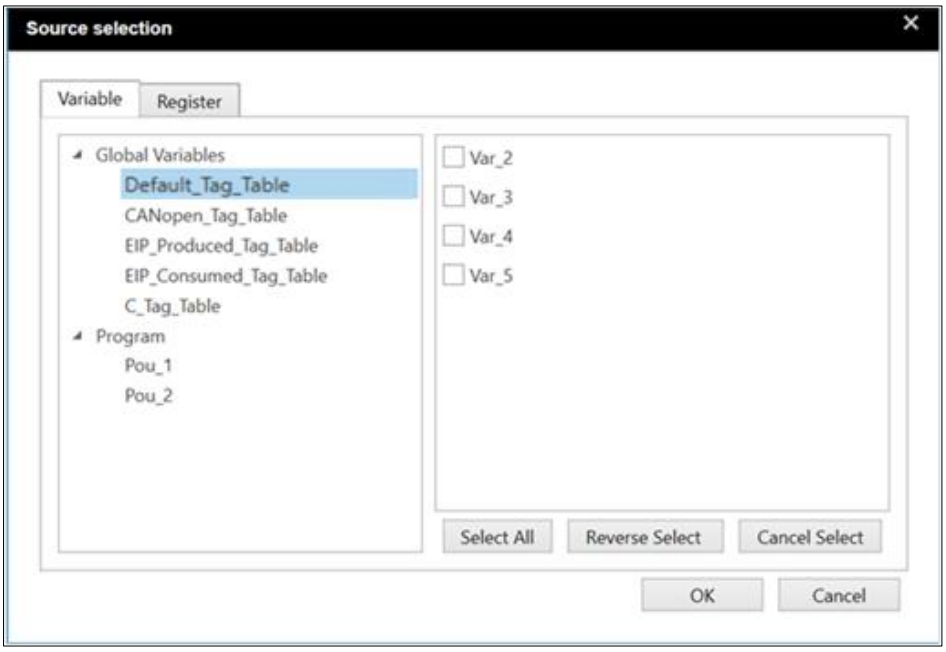


Figure 14 - 9: Source selection – Variable

The **Source selection** window has two tabs:

- Variable
- Register

The **Variable** tab displays by default in the **Source selection** window. User can select **Global Variables** or **Program** category.

The **Global Variables** displays **DefaultVariable Table**, **CANopen_Tag_Table**, **EIP_Produced_Tag_Table** and **EIP_Consumed_Tag_Table** and C_Tag _Table

The **Program** displays all the POUs created in the controller in **Programming** > **Program**. Based on the item selected on left, the respective variables display in the right side of the **Variable** tab.

The functions in the **Variable** tab of **Source selection** window are described in the following table:

Function	Description
Select All	Click to select all variables.
Revert Select	Click to revert the selected variables.
Cancel Select	Click to cancel all the variables.

If user want to add **Register**(s) to the **Monitor Table**, they can click on **Register** Tab. The **Register** tab of **Source selection** window is shown in the following figure.

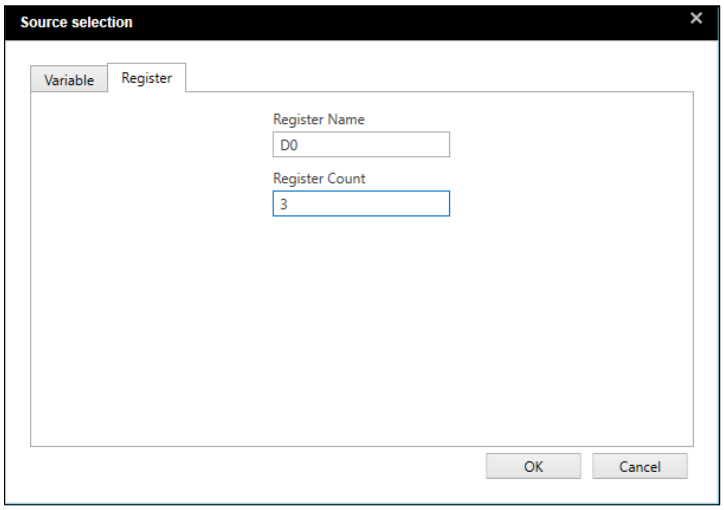


Figure 14 - 10: Source Selection - Register

The functions in the **Register** tab of **Source selection** window are described in the following table:

Function	Description
Register Name	Enter the starting register.
Register Count	Enter the number of registers needed from the starting register.

14.2.3 Monitor Table Context Menu

Context menu in **Monitor Table** window is shown in the following figure.

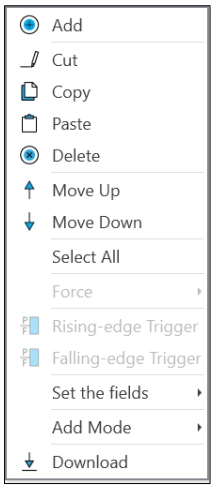


Figure 14 - 11: Monitor Table - Context Menu

The functions in the **Monitor Table** context menu are described in the following table:

Function	Description
Add	Click to open **Source selection** window to add variables/registers.
Cut	Click to cut the selected item.
Copy	Click to copy the selected item.
Paste	Click to paste the cut/copied item.
Delete	Click to delete the selected items.
Move Up	Click to move an item above its present level.
Move Down	Click to move an item below its present level.
Select All	Click to select all items.
Force	Click to Force On/Off Boolean Variables.
Rising-edge Trigger	Send OFF for one scan time and then ON.
Falling-edge Trigger	Send ON for one scan time and then OFF.
Set the fields	Click to hide/display the column property. Tick mark on the left side of the function means the respective column is shown. **Source**, **Data Type**, **Display Data Format**, **Value**, **Set Value**, **Comment** can be displayed or hidden. ***NOTE**: All columns display by default.*

Function	Description
Add mode	Click to select insert mode.
Download	Click to download the selected items

14.2.4 Using Monitor Table

User can monitor items in online mode after adding a monitor table and monitoring variables/registers. In offline mode, user can select the items to be set by check the boxes on the left, then set the values in **Set Value** field. Then, switch to the online mode, user can do batch download of the selected variables or register values by clicking the download button in toolbar or the context menu as shown in the following figure.

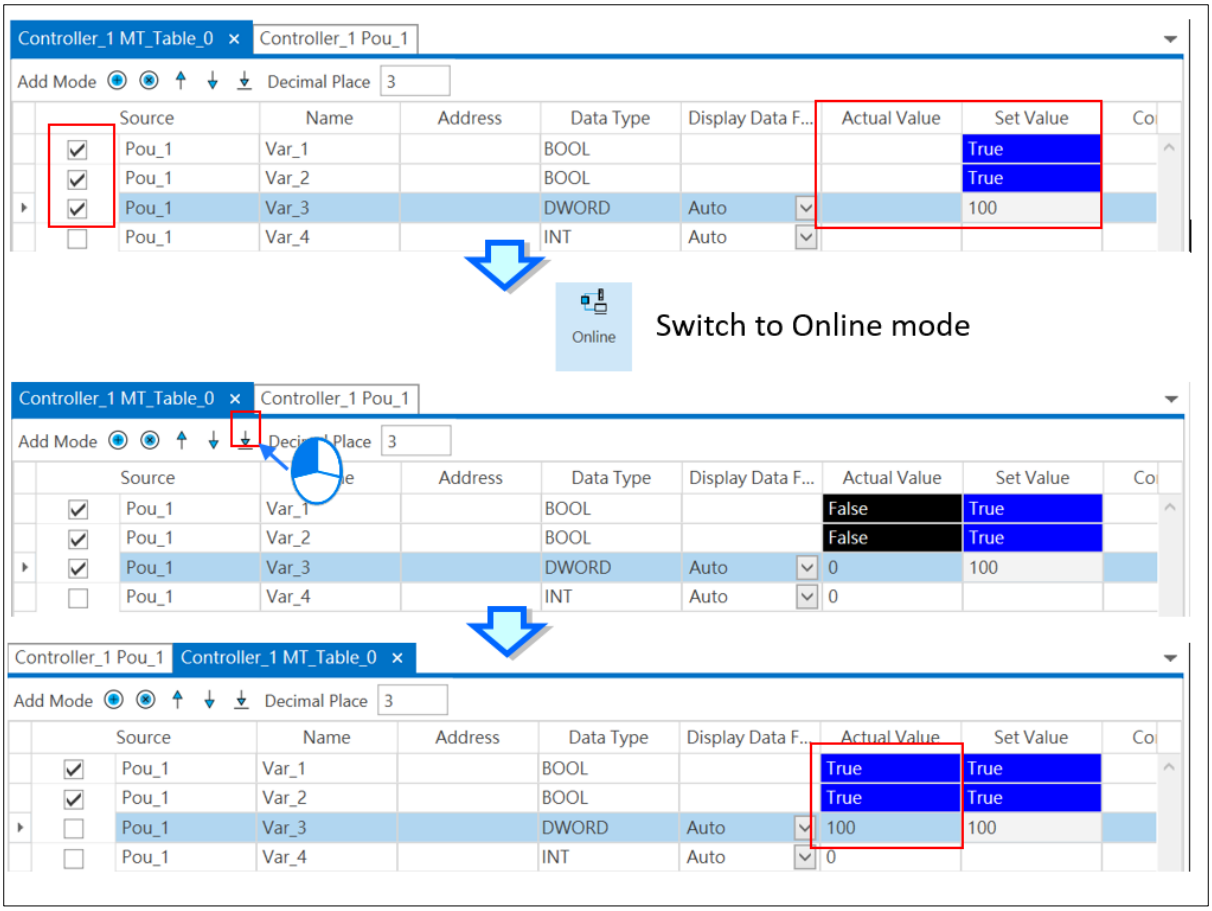


Figure 14 - 12: Batch Download of Monitor Table Values

In the online mode, when setting **Set Value**, user can press the Enter button or click somewhere else with mouse to write the values to **Actual Value**. For BOOL data type, user can just double-click the **Set Value** to toggle state.

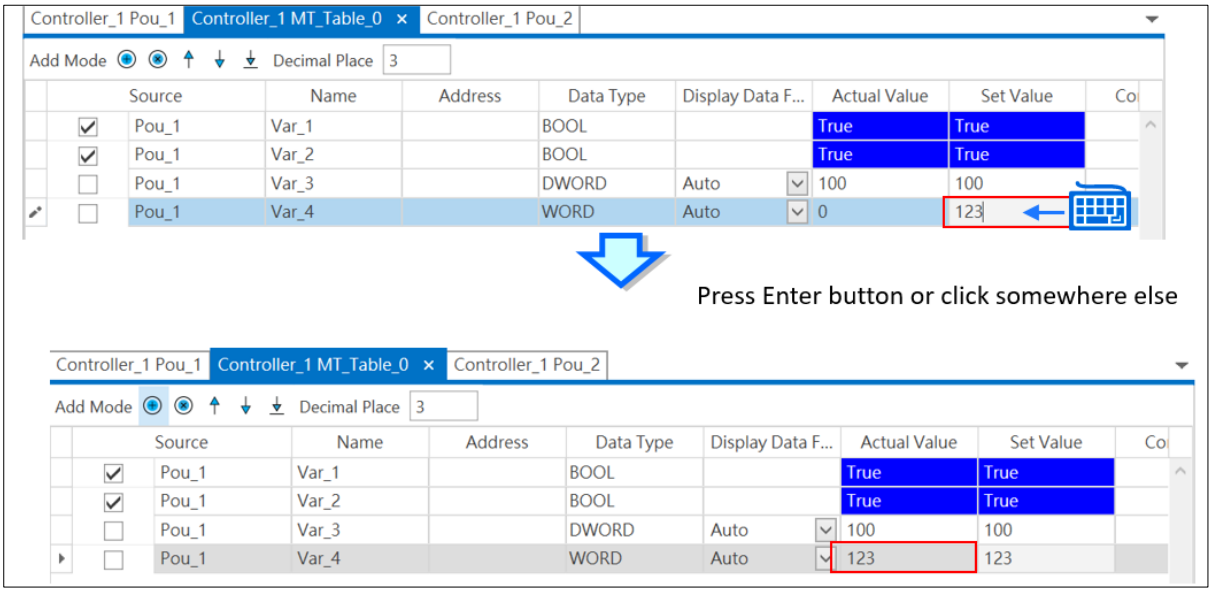


Figure 14 - 13: Real-time Monitor Table Values Setting

14.3 Tuning Tool

User can tune the Servo Drive in **Gain Tuning**. User can locate Gain Tuning in **Project Explorer** in **Servo Device** > **Tuning** > **Gain Tuning**. Double-click on Gain Tuning option to open **Gain Tuning** window. **Gain Tuning** window is shown in the following figure.

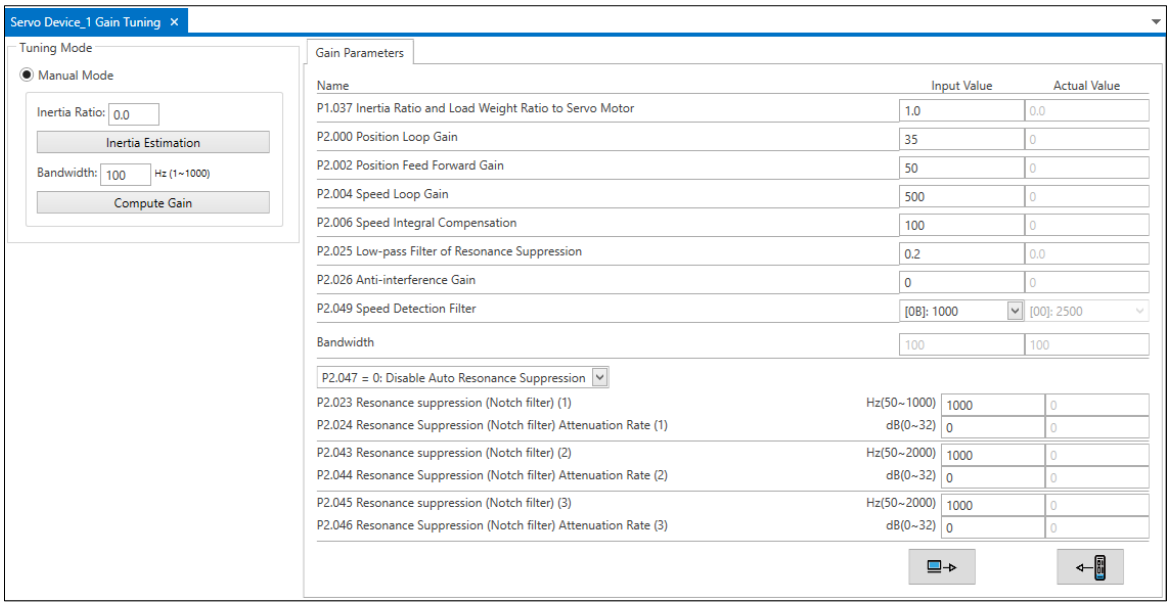


Figure 14 - 14: Gain Tuning

14.3.1 Calculate the Inertia Ratio

Click on the **Inertia Estimation** button.

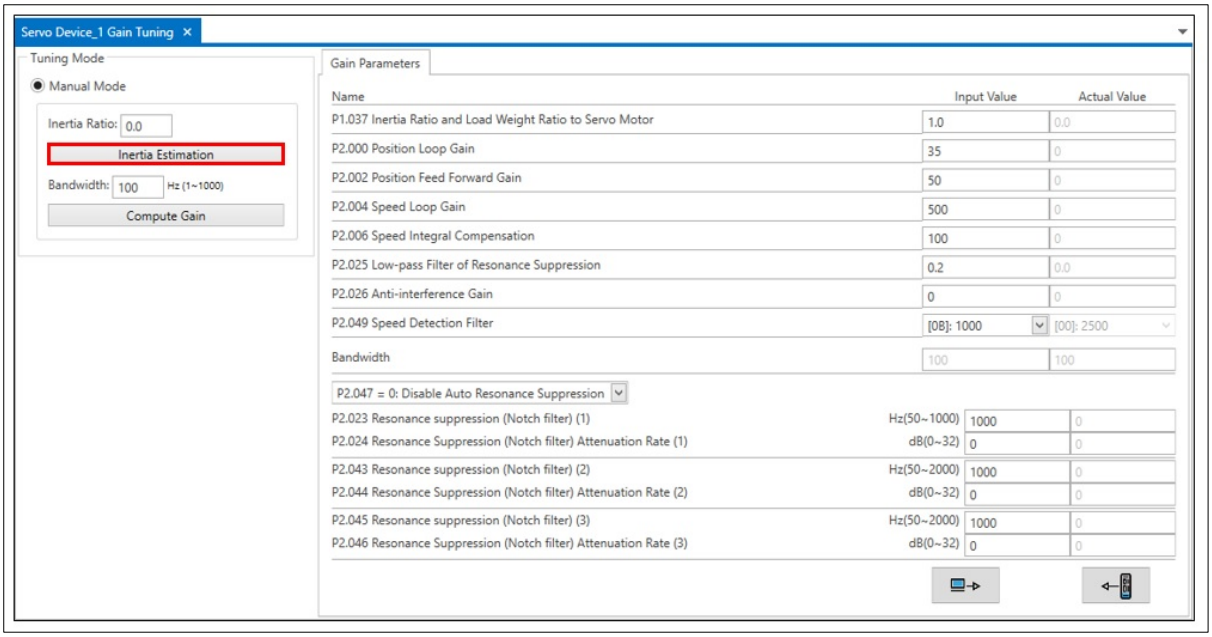


Figure 14 - 15: Click on inertia button

A warning window will pop up, as shown in the following figure. Click **OK** to continue as shown in the following figure.

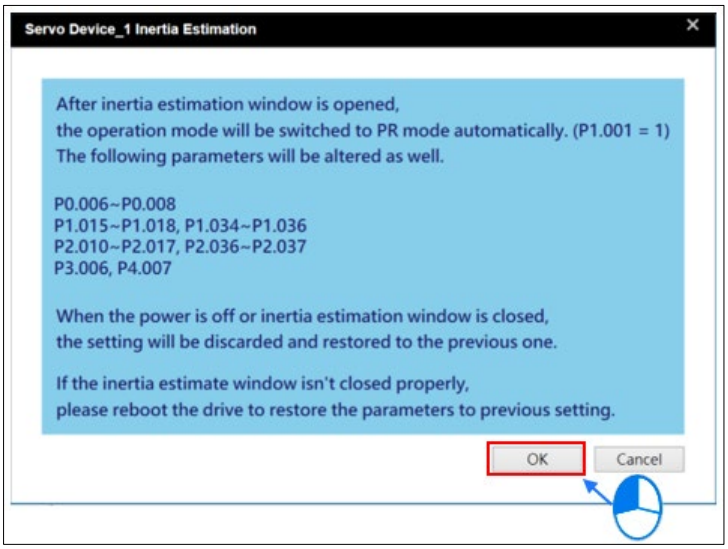


Figure 14 - 16: Inertia Estimation - Warning

The **Inertia Estimation** window opens for user to estimate the Inertia Ratio step-by-step.

1. Click on **Servo On** button. If there is alarm, user need to clear the alarm by clicking the **Alarm Reset** button.

Figure 14 - 17: Inertia Estimation - Step 1

2. Enter the Acceleration/Deceleration Time, **S-curve constant**, and **Jog Speed**, then click **Download** as shown in the following figure.

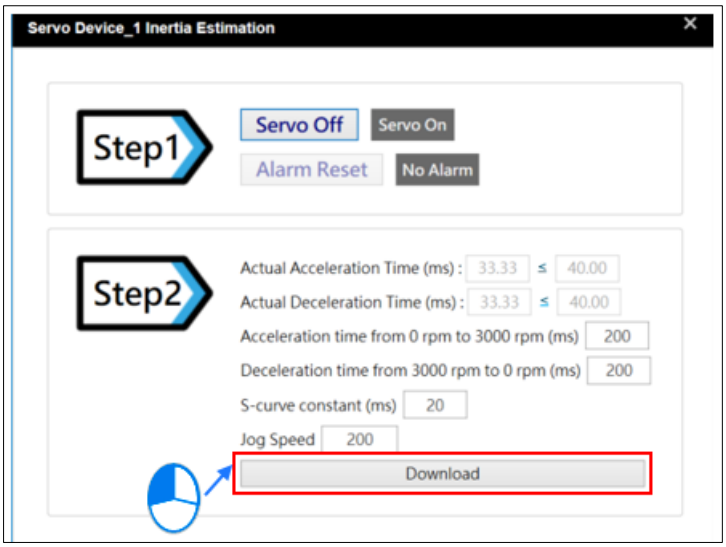


Figure 14 - 18: Inertia Estimation - Step 2

3. Arrows in **1** indicates to move the motor to the desired position and use **Position 1** and **Position 2** in **2** indicates to set the positions. After the positions are set, click **Start Moving** to do the inertia estimation as shown in the following figure.

Figure 14 - 19: Inertia Estimation - Step 3

4. Click **Setting** to update the value of **Inertia Ratio** as shown in the following figure.

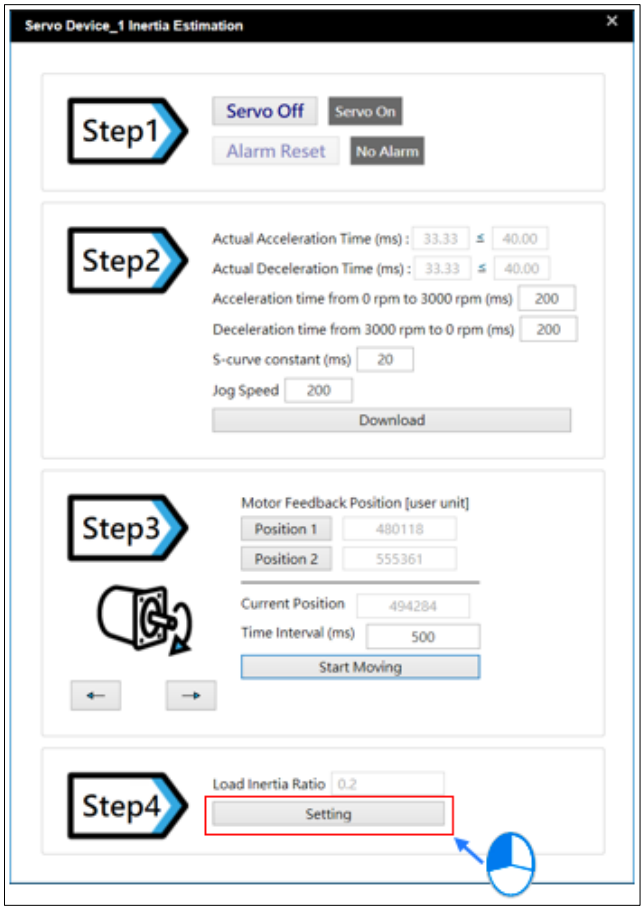


Figure 14 - 20: Inertia Estimation - Step 4

14.3.2 Set up the bandwidth

After calculating **Inertia Ratio**, the next step is to set up the Servo Drive **Gain**.

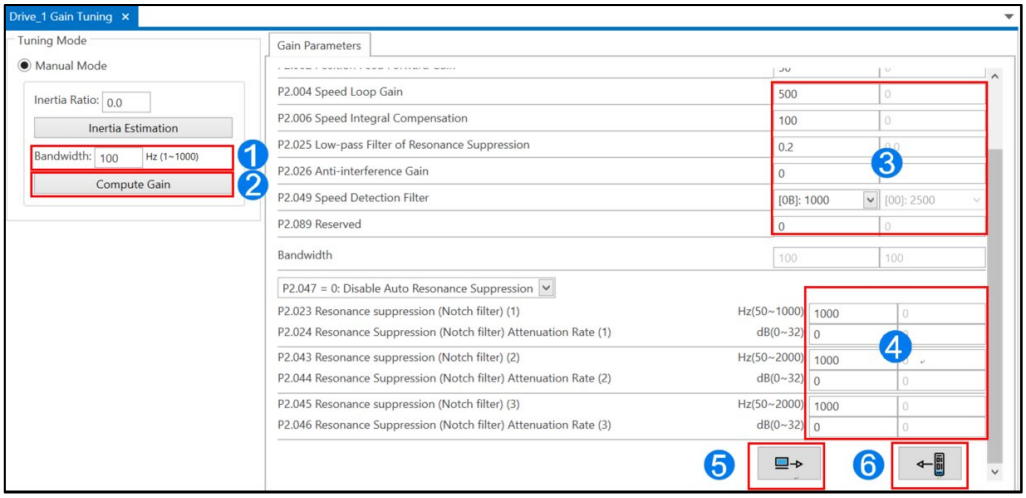


Figure 14 - 21: Gain Parameters

The functions in the **Gain** parameters in **Gain Tuning** window are described in the following table:

Function	Description
1	Enter the **Bandwidth**.
2	Click **Compute Gain** to calculate the gain parameters.
3	Gain parameters. **Input Value** from the calculation of **Bandwidth**. **Actual Value** from the actual value in servo drive.
4	To set up the resonance suppression filter.
5	To download **Input Value** to servo drive (**Actual Value**).
6	To upload the parameters from servo drive to show in **Actual Value**.

14.4 Trace and Oscilloscope

User can view the waveforms or trends of value in **Address** and **Register** with the help of **Data Tracer** and **Data Logger**. User can locate them in **Controller** > **Commissioning**.

14.4.1 Data Tracer

The **Data Tracer** is used for the real-time data collection of values or states in variables or addresses after some trigger condition is met to perform analysis with the waveforms or trends. Once the window is closed, the data tracing will stop. User can locate Data Tracer in the **Project Explorer** in **Controller** > **Commissioning** > **Data Tracer**.

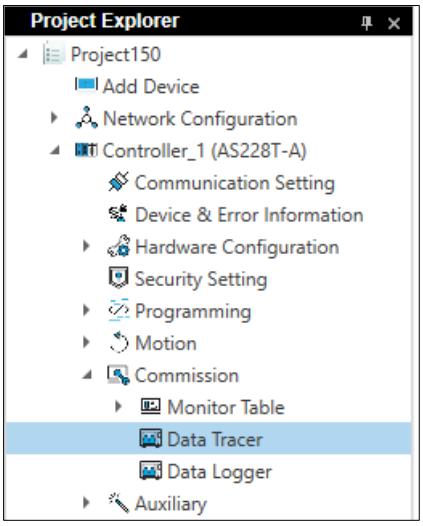


Figure 14 - 22: Data Tracer in Project Explorer

Double-click the **Data Tracer** option to open the **Data Tracer** window as shown in the following figure.

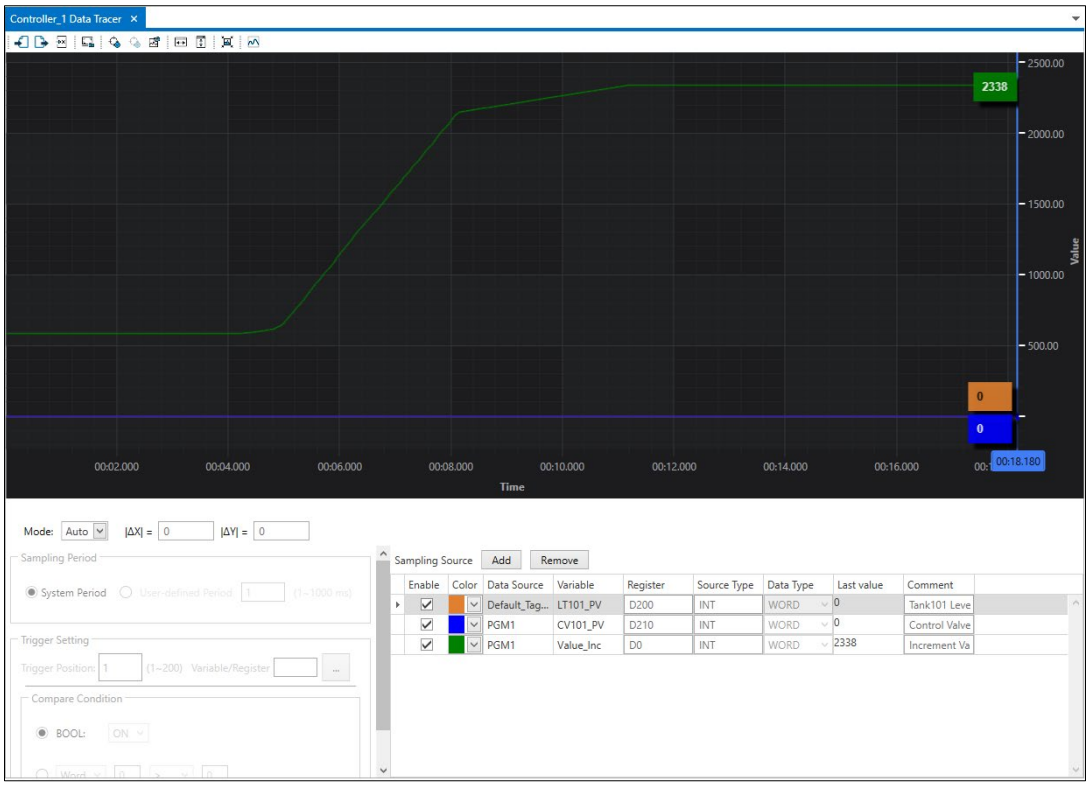


Figure 14 - 23: Data Tracer

The **Data Tracer** window has the following areas:

- Data Trace toolbar
- Trace Display area
- Sampling Source and Parameter Setting

14.4.1.1 Data Trace Toolbar

The **Data Trace Toolbar** provides icons for data trace management as shown in the following figure.

Figure 14 - 24: Data Tracer Toolbar

The functions in Data Tracer toolbar are described in the following table:

Icon/Function	Description
	Open the sample settings from a specified path. Click to display **Open** window (*.trascp file).
	Save the sample settings to a specified path. Click to display **Save As** window (*.trascp file).
	Export the trend data to a specified path. Click to display **Save As** window (*.csv file).
	Click to take a screenshot of the trend display. (*.png, *.jpg & *.bmp)
	Click to start the Data Trace.
	Click to stop the Data Trace.
	Click to clear the Data Trace.
	There are two vertical lines in the trend display area for user to measure the trend.
	There are two horizontal lines in the trend display area for user to measure the trend.
	Click to zoom fit.
	Click to open **FFT Analysis** window.

14.4.1.2 Trace Display area

The variable trends are displayed in real-time. The **Trace Display** area of **Data Tracer** is shown in the following figure.

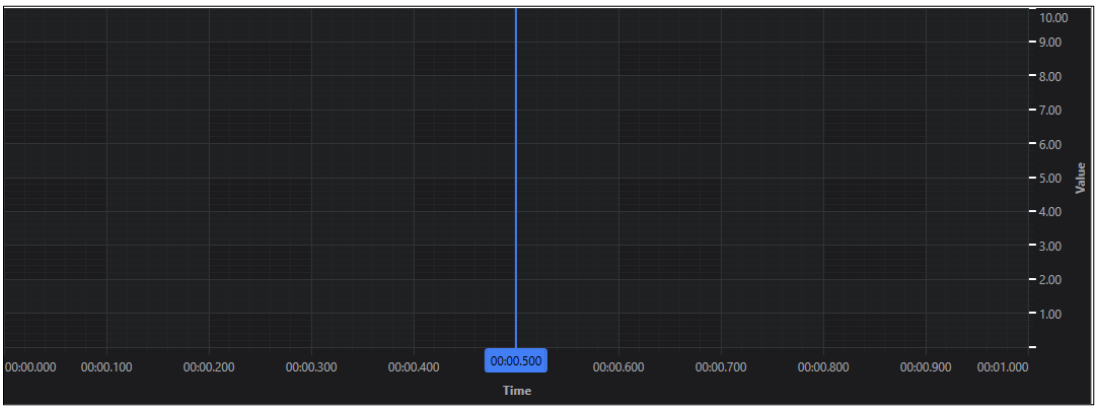


Figure 14 - 25: Trace Display Area

14.4.1.3 Sampling Source and Parameter Setting

The area below the **Trace Display** area is the **Sampling Source and Parameter Setting** area.

The **Sampling Source** area is on the right-side bottom of the **Data Tracer** window. **Sampling Source** area is shown in the following figure.

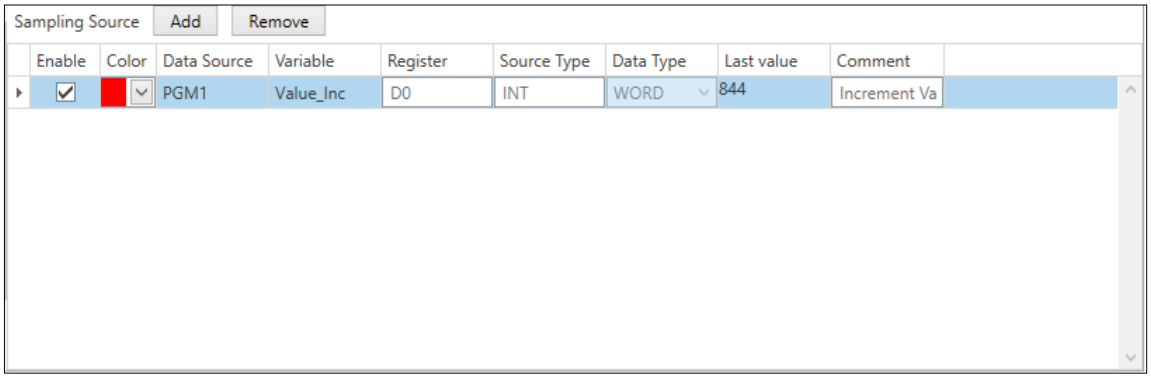


Figure 14 - 26: Data Tracer - Sampling Source area

The functions in the **Sampling Source** area in the **Data Tracer** window are described in the following table:

Function	Description
Add	Click to open **Select Variable** window. Select **Variables**/**Registers** to add sample source.
Remove	Click to remove the sample source selected.
Enable	Click the check box to enable the sample source.
Color	Select the Trace color from the color pallet.
Data Source	Displays the source of Data e.g. **Program** name.
Variable	Displays the variable name e.g. LT101.

Function	Description
Register	Displays the register name e.g. D100.
Source Type	Displays the data type of the variable.
Data Type	Displays the data type of the register.
Last Value	Displays the last value.
Comment	Displays the comment of variable or register.

14.4.1.4 Adding Variables or Registers to Data Tracer

Follow these steps to add **Variables** or **Registers** to **Data Tracer**.

1. Click on **Add** in the **Sample Source** area.

   Result: The **Select Variable** window displays.

   ***NOTE**: The **Select Variable** window contents are the same as the one described in Figure 13 -9.*

2. Click on **OK** button to finish sample source selection for variables or registers.

   Result: An example of sample sources for **Data Tracer** is shown in the following figure.

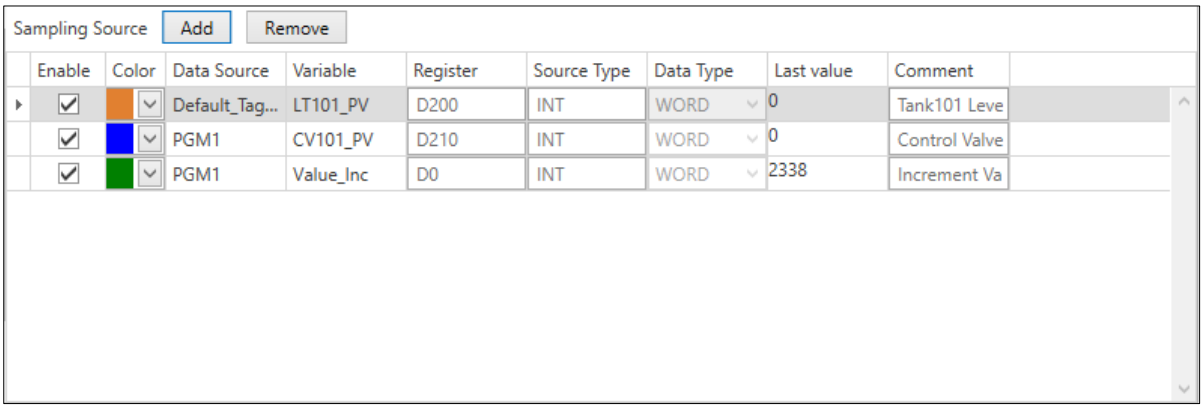


Figure 14 - 27: Example of Data Trace - Sampling Source

14.4.1.5 Parameter Setting

Data Trace can be activated in two modes:

- Auto
- Trigger

Sampling Period and **Trigger Setting** is available on the left-side bottom of the **Data Trace** window. **Sampling Period** and **Trigger Setting** is shown in the following figure.

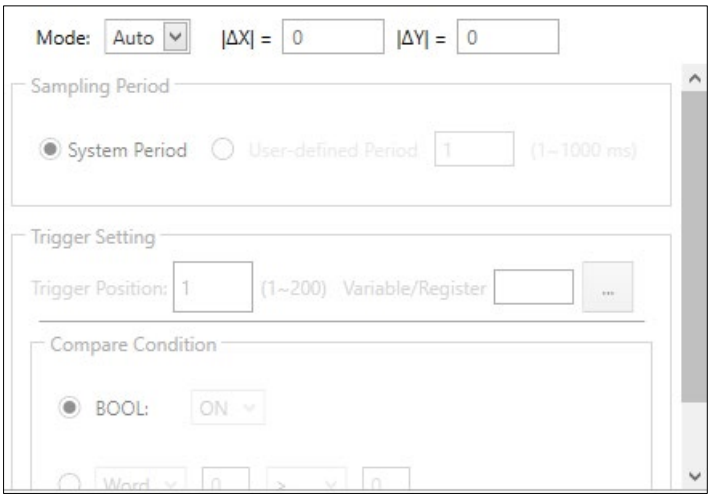


Figure 14 - 28: Data Trace - Sampling Period and Trigger Setting

14.4.1.6 Sampling Period and Trigger Setting Functions

The functions in **Sampling Period** and **Trigger Setting** area in the **Data Tracer** window are described in the following table.

Function	Description
Mode	Select the **Sampling Period** and **Trigger Setting** mode. Options are: • Auto • Trigger ***NOTE**: The default value is **Auto**. In **Auto** mode, **Sampling Period** and **Trigger Setting** fields are greyed out. They are enabled in the **Trigger** mode.*

Function	Description
(ΔX)	Enter the X-axis field (time) to start Data Trace recording
(ΔY)	Enter the Y-axis filed (Sampling Source) value to start Data Trace recording
Sampling Period – System Period	Click to select **System Period**. It means sampling is done once in every scan cycle of the Controller
Sampling Period – User-defined Period	Click to select user-defined **Sampling Period** and enter the sampling period time in the range of (1 ~ 1000ms)
Trigger Setting – Trigger Position	Enter the trigger positions as per user requirement. The PLC read 200 pieces of sample data and display the data in the Data Tracer if the trigger condition is met. The trigger position settings are different for one-shot trigger and continuous trigger
Trigger Setting – Variable/Register	Click to open the **Select Variable** window to select the trigger setting Variable/Register on whose values the trigger for sampling must be performed
Trigger Setting – Compare Condition - BOOL	Click to select **BOOL** radio button if the trigger variable/register is Bool. Select the Trigger condition from the drop-down list. Options are: • ON • OFF • Rising Edge Triggered • Falling Edge Triggered • Rising or Falling Edge Triggered ***NOTE**: **BOOL** is the default Variable/Register type and **ON** is the default trigger value.*
Trigger Setting – Compare Condition -	Select the non-Boolean variable/register for trigger. Options are: • Word • DWord • Real Select the comparison condition and value. Options are: • >= • < • == • != • ~ • !~

Function	Description
	***NOTE**: **Word** is the default non-Boolean trigger variable/register and **>** is the default comparison condition.*

The **Sampling Period** setting is shown in the following figure.

Figure 14 - 29: Data Tracer – Sampling Period

The **Trigger Setting** is shown in the following figure.

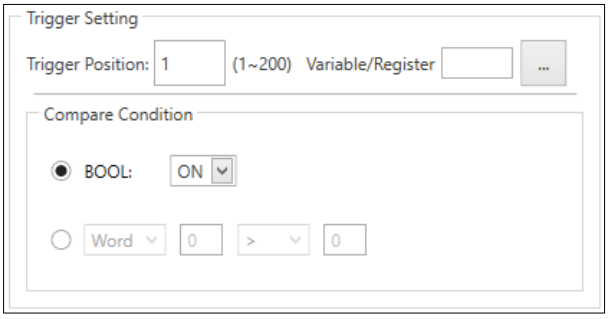


Figure 14 - 30: Data Tracer – Trigger Setting

Sample **FFT Analysis** window where the magnitude and the frequency can be compared is shown in the following figure.

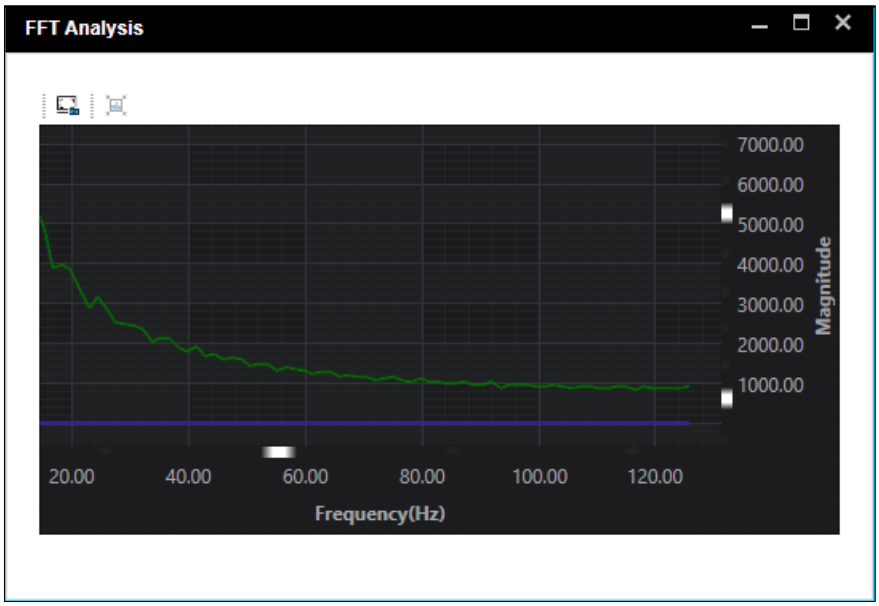


Figure 14 - 31: FFT Analysis

The result of Data Tracer is shown in the following figure.

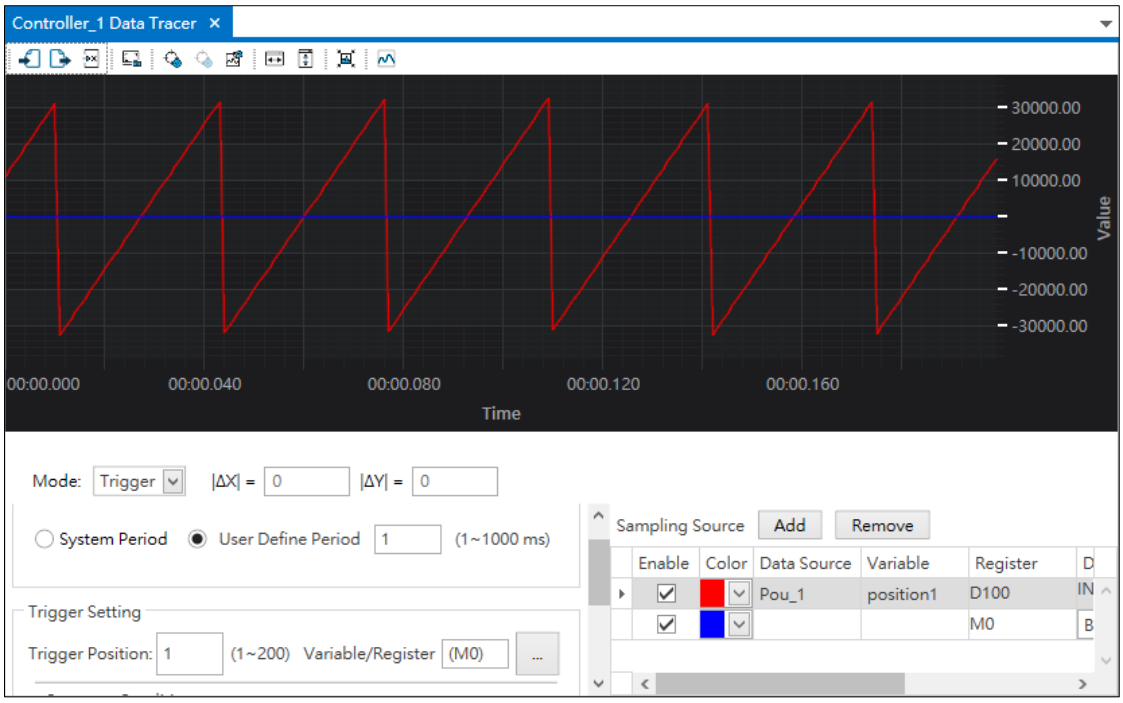


Figure 14 - 32: Data Tracer – Example

14.4.2 Data Logger

The **Data Logger** is used for the long-term recording of values or states in variables/registers, trending and storing them in the PLC and the memory card of the PLC for analysis of trends. User can locate Data Logger in the **Project Explorer** in **Controller** > **Commissioning** > **Data Logger**.

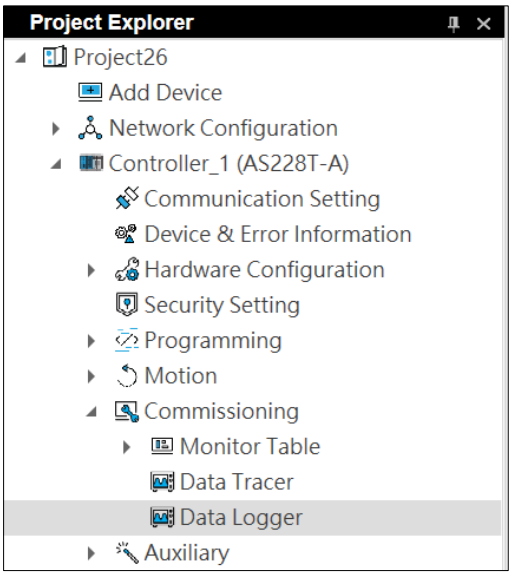


Figure 14 - 33: Data Logger in Project Explorer

Double-click the **Data Logger** option to open the **Data Logger** window as shown in the following figure.

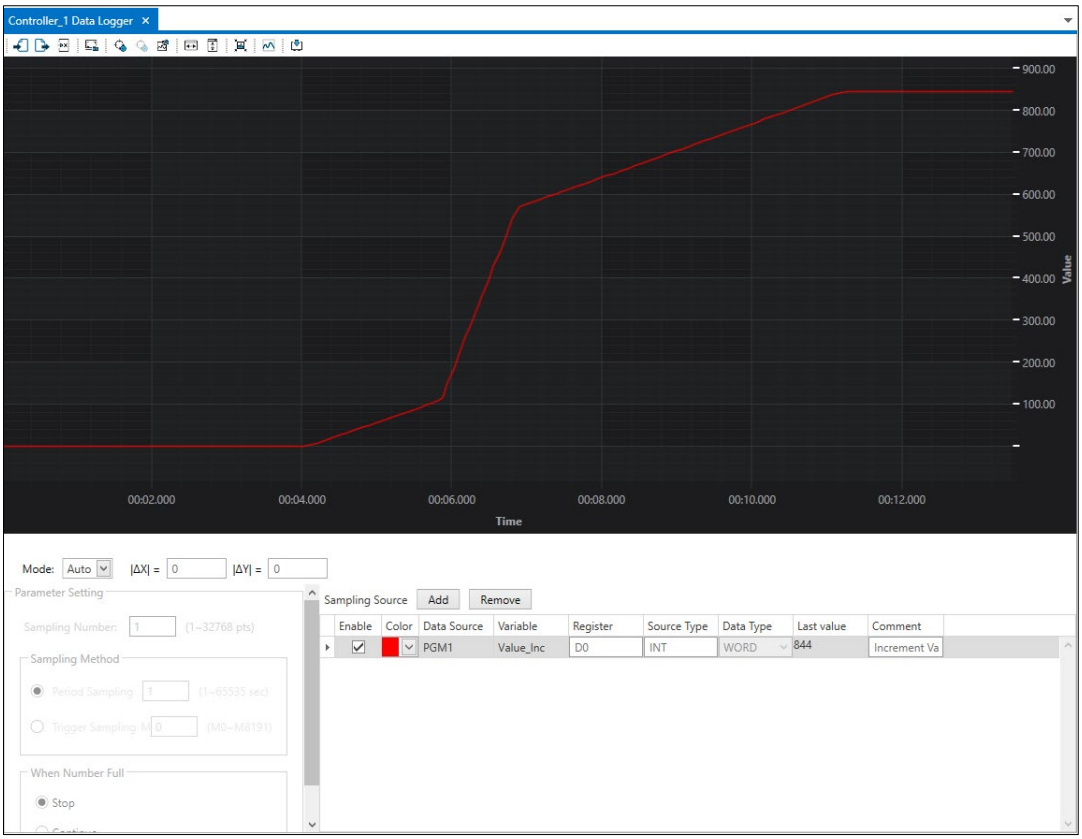


Figure 14 - 34: Data Logger

The **Data Logger** window has the following areas:

- Data Logger toolbar
- Log Display area
- Sampling Source and Parameter Setting area

14.4.2.1 Data Logger Toolbar

The **Data Logger Toolbar** provides icons for data log management as shown in the following figure.

Figure 14 - 35: Data Logger Toolbar

The functions in Data Logger toolbar functions are described in the following table:

Icon/Function	Description
	Open the Sampling settings from a specified path. Click to display **Open** window (*.logscp file).
	Save the Sample settings to a specified path. Click to display **Save As** window (*.logscp file).
	Export the trend data to a specified path. Click to display **Save As** window (*.csv file).
	Click to take a screenshot of the trend display (*.png, *.jpg & *.bmp).
	Click to start the Data Log.
	Click to stop the Data Log.
	Click to clear the Data Log.
	There are two vertical lines in the trend display area for user to measure the trend.
	There are two horizontal lines in the trend display area for user to measure the trend.
	Click to zoom fit.
	Click to open **FFT Analysis** window.
	Click to write the records to memory card.

14.4.2.2 Log Display

The variable trends are displayed in real-time and stored. The **Log Display** area of **Data Logger** is shown in the following figure.

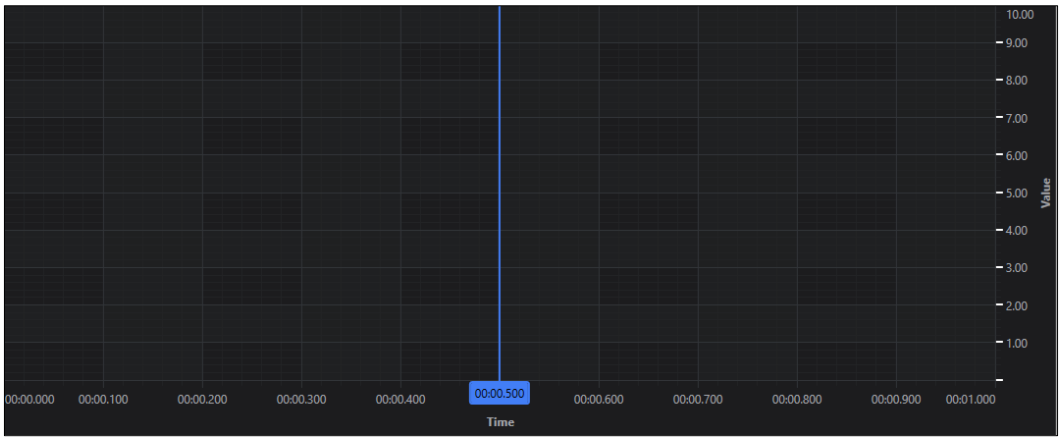


Figure 14 - 36: Log Display

14.4.2.3 Sampling Source and Parameter Setting

The area below the **Log Display** area is the **Sampling Source and Parameter Setting** area.

The **Sampling Source** area is on the right-side bottom of the **Data Logger** window. **Sampling Source** area is shown in the following figure.

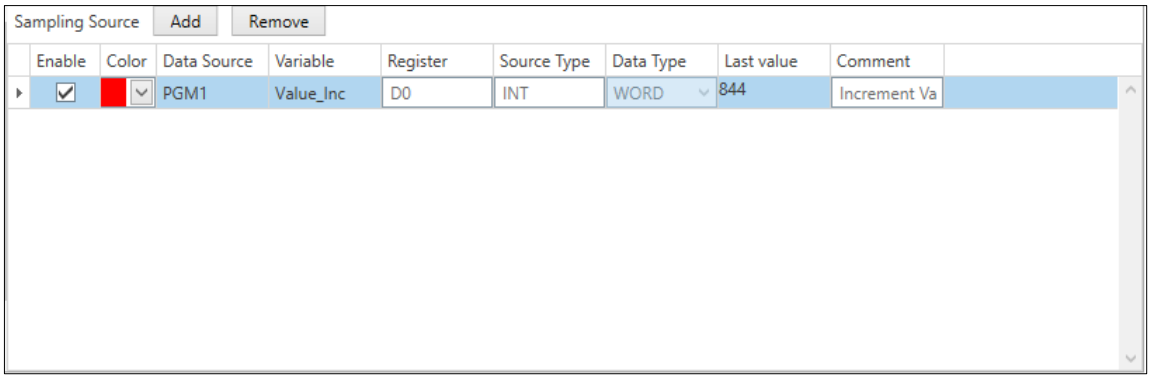


Figure 14 - 37: Data Logger – Sampling Source

The functions in the **Sampling Source** area in the **Data Logger** window are described in the following table:

Function	Description
Add	Click to open **Select Variable** window. Select **Variables**/**Registers** to add sample source.
Remove	Click to remove the sample source selected.
Enable	Click the check box to enable the sample source .
Color	Select the Trace color from the color pallet.
Data Source	Displays the source of Data e.g. **Program** name.
Variable	Displays the variable name e.g. LT101.
Register	Displays the register name e.g. D100.
Data Type	Displays the data type of the register.
Last Value	Displays the last value.
Comment	Displays the comment of variable or register.

14.4.2.4 Adding Variables or Registers to Data Logger

Follow these steps to add **Variables** or **Registers** to **Data Logger**:

1. Click on **Add** in the **Sample Source** area.

   Result: The **Select Variable** window displays.

   ***NOTE**: The **Select Variable** window contents are the same as the one described in Figure 13 -9.*

2. Select variables or registers in **Select Variable** window. Click on **OK** button to finish sample source selection for variables and registers.

   Result: An example of sample sources for **Data Logger** is shown in the following figure.

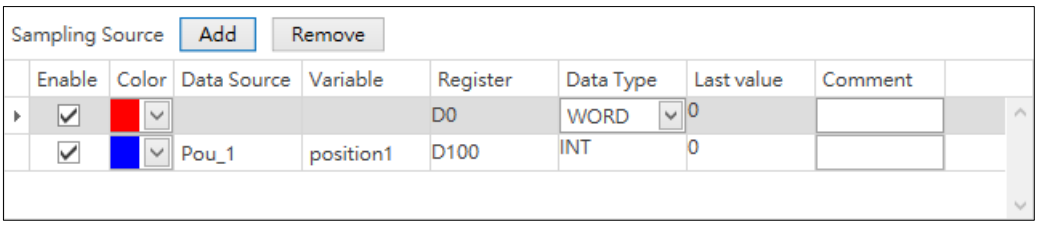


Figure 14 - 38: Example of Data Log – Sampling Source

14.4.2.5 Parameter Setting

The **Parameter Setting** area is used for setting the sample parameters related with the **Data Logger**. **Parameter Setting** area of **Data Logger** is shown in the following figure.

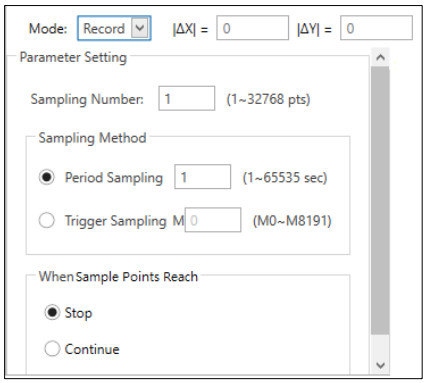


Figure 14 - 39: Data Logger – Parameter Setting

Data Log can be activated in two modes:

- Auto
- Record

The functions in the **Parameter Setting** area in the **Data Logger** window are described in the following table:

Function	Description
Mode	Select the **Sampling** and **Parameter Setting** mode. Options are: • Auto • Record

Function	Description
	***NOTE**: **Auto** is the default value. In **Auto** mode, **Parameter Setting** field is grayed out. They are enabled in the **Record** mode.*
(ΔX)	The difference value of 2 vertical line.
(ΔY)	The difference value of 2 horizontal line.
Parameter Setting – Sampling Number	Enter the sampling number in 1 ~ 32768 pts range.
Parameter Setting – Sampling Method	Select the sampling method. Options are: • Periodic Sampling • Trigger Sampling ***NOTE:** **Periodic Sampling** is the default value. When using **Periodic Sampling**, user needs to setup the sample period and the **Data Logger** will execute from the **Sampling Source** every period. When using the **Trigger Sampling**, user needs to setup the trigger register (M register).*
Parameter Setting – Sampling Method – Period Sampling	Click to select **Sampling Method** - periodic sampling. Enter the sampling period in 1 ~ 65535 sec
Parameter Setting – Sampling Method – Trigger Sampling M	Click to select **Sampling Method** – triggered sampling. Enter the M register for sample trigger. If the trigger register is ON, the Data Logger will execute the sampling once and after completing the sampling process, it will automatically turn the device to OFF. ***NOTE**: If the user program tries to execute other actions to the M Register, then the ON/OFF function of the device can be effected.*
Parameter Setting – When Sample Points Reach	Select the action when the Data Logger has recorded the number of set sample points. Options are: • Stop • Continue ***NOTE**: **Stop** is the default value.*

The result of **Data Logger** is shown in the following figure.

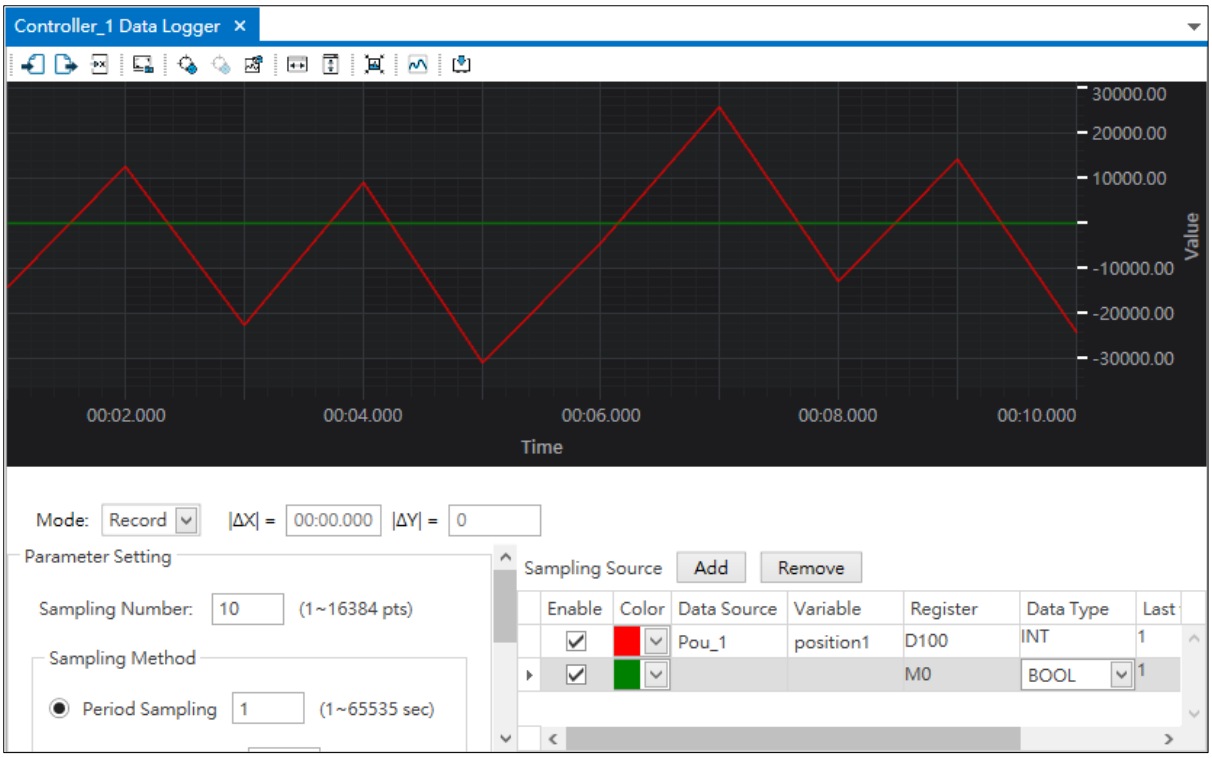


Figure 14 - 40: Data Logger - example

14.4.3 Servo Device Oscilloscope

Servo Device Oscilloscope is used for the real-time servo drive status. User can locate Servo Device Oscilloscope in **Servo Device** > **Oscilloscope** in **Project Explorer** as shown in the following figure.

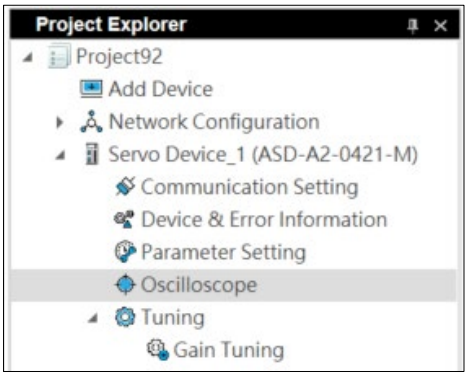


Figure 14 - 41: Servo Device – Oscilloscope in Project Explorer

Double-click on the **Oscilloscope** to open the **Oscilloscope** window as shown in the following figure.

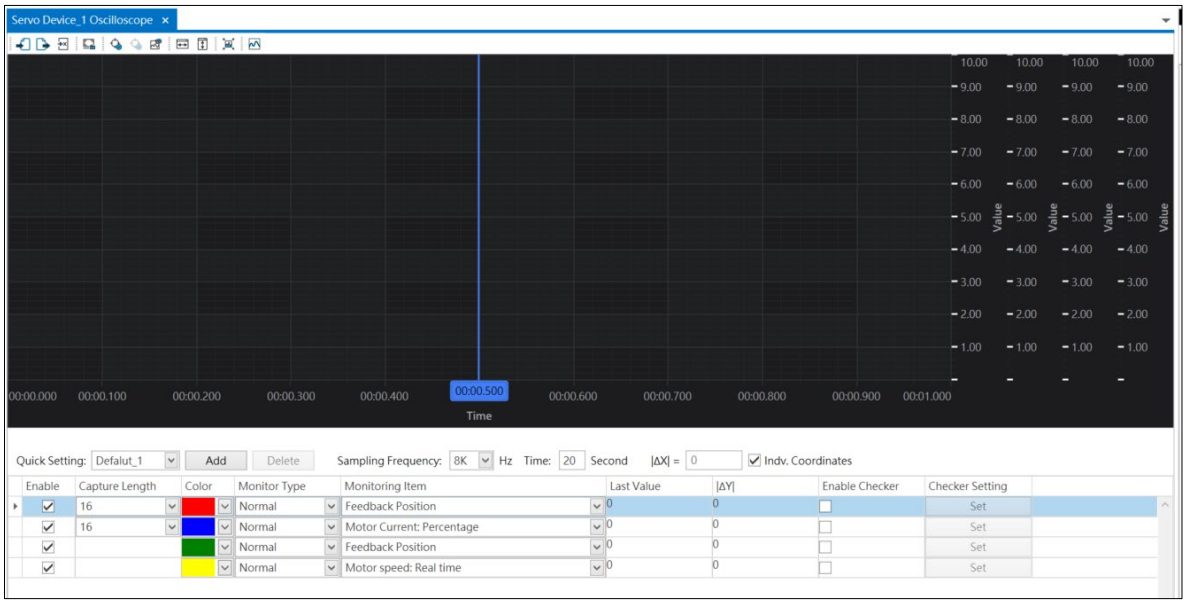


Figure 14 - 42: Servo Device - Oscilloscope

The Servo Device **Oscilloscope** window has the following areas:

- Oscilloscope toolbar
- Display area
- Monitoring Channel Setting

14.4.3.1 Oscilloscope Toolbar

The **Oscilloscope Toolbar** is shown in the following figure.

Figure 14 - 43: Servo Device - Oscilloscope - Toolbar

The functions in Servo Device **Oscilloscope Toolbar** are described in the following table:

Icon/Function	Description
	Open the sample settings & points from a specified path. Click to display **Open** window (*.asdscp file).
	Save the sample settings & points to a specified path. Click to display **Save As** window (*.asdscp file).
	Export the trend data to a specified path. Click to display **Save As** window (*.csv file).
	Click to take a screenshot of the trend display (*.png, *.jpg & *.bmp)
	Click to start the scope.
	Click to stop the scope.
	Click to clear the scope display.
	There are two vertical lines in the trend display area for user to measure the trend.
	There are two horizontal lines in the trend display area for user to measure the trend.
	Click to zoom fit.
	Click to open **FFT Analysis** window.

14.4.3.2 Oscilloscope Display Area

The selected monitoring channels are displayed in real-time. The Display Area is shown as the following figure.

X axis of the screen represents time. Unit is second (sec). Y axis of the screen represents data. Y axis on the right side is allocated according to the channel user select. Data unit displayed in Y axis is marked with the same coordinates when in initial status.

It is suitable for the monitor items which have the same unit. If the monitor items have different unit, it is suggested to select **Indv. Coordinates** in the top part of **Monitoring Channel Setting** area.

Figure 14 - 44: Monitoring channel setting

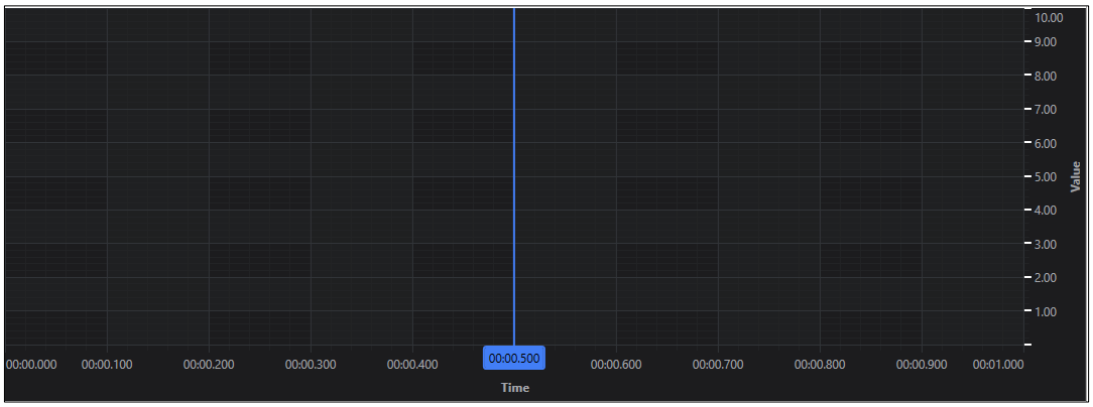


Figure 14 - 45: Servo Device Oscilloscope - Display Area

14.4.3.3 Monitoring Setting

The area below the **Display area** is the **Monitoring Setting** area. Upper part of **Monitoring Setting** area is to set the display condition of **Display Area**.

Figure 14 - 46: Servo Device Oscilloscope - Monitoring Setting area

The functions in the **Monitoring Setting** area in Servo Device **Oscilloscope** window are described in the following table:

Function	Description
Quick Setting	To store the saved monitoring channel list. Click **Add** to save the current selected monitoring channels into list. A **Naming Window** will pop up for user to type the name. User can select the saved monitoring channel list from drop-down manual. Click **Delete** to delete the selected list.

Function	Description
Sampling Frequency	To select the sampling frequency in **Display Area**. To select 8KHz to monitor 4 monitoring channels or 16Hz to monitor 2 monitoring channels.
Time	Adjust the resolution of time axis (X axis of scope). The resolution of scope image can be enhanced. The setting range is from 20 seconds ~ 120 seconds)
|ΔX|	Display the time interval of X axis when enabling the function of vertical line measurement from **Servo Drive Oscilloscope** Toolbar .
Indv. Coordinates	Channel with individual coordinate shows the actual monitor variables.

Channel setting could be found in bottom part of **Monitoring Setting** area.

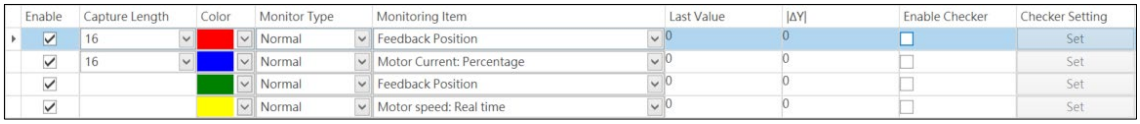


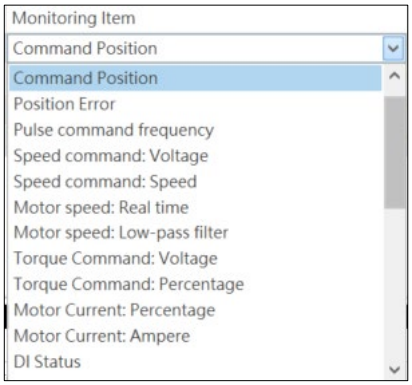
Figure 14 - 47: Servo Device Oscilloscope - Monitoring Setting - Channel setting

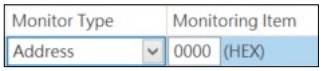
The functions in the **Channel setting** in Servo Device **Oscilloscope** window are described in the following table:

Function	Description
Enable	User can select the desire channel.
Capture Length	Parameter format can be divided into two kinds, 16-bit and 32-bit. The standard setting of channel data is 16-bit. Thus, when user check 32 bit to expand the amount of one channel, ASDA-Soft will automatically close another channel to support the selected one.
Color	To change the color of data curve.
Monitor Type	Select the command source. 5 sources could be selected: • Normal • Address

<table>
<tr><th>Function</th><th>Description</th></tr>
<tr><td></td><td>

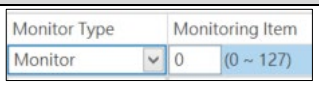
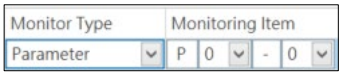
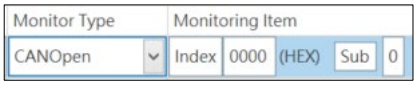
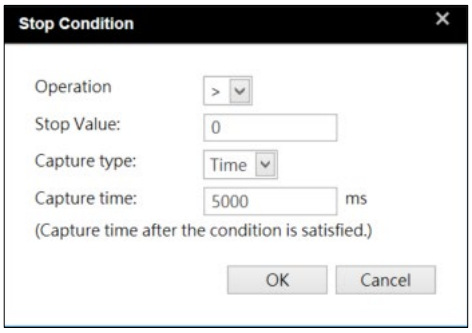
- Variables
- Parameters
- CANopen

***NOTE**: **Normal** is the default value.*
</td></tr>
<tr><td>Monitoring Item</td><td>
Based on the selection of Monitor Type, this area will show different settings.
Normal:
Directly access the built-in monitor items. User can select the desired monitor item from the drop-down manual as shown below.


Address:
Enter the specific position format can access monitor variable, parameter and CANopen Object. The unit is in hexadecimal.


Followings are the descriptions of the setting format:
<table>
<tr><th>Monitor Variable</th><th>Specify Parameter</th><th>CANopen Object</th></tr>
<tr><td>0x100000<u>XX</u>
XX = Monitor variable code
Range: 0x00h ~ 0x7Fh</td><td>0x20002<u>X</u> <u>YY</u>
YY = Parameter number
X = Parameter group
number</td><td>0x20<u>XX</u> <u>YYYY</u>
YYYY = Index
XX = Sub-Index</td></tr>
</table>
For example, enter 0x01 to access 0x10000001, 0x2000212C to access parameter P1-44, and 0x200260C1 to access OB Index 60C1h, Sub 2.

Variables:
It can enter the specific monitor variable code. Unit is 0 ~ 127 in decimal. The monitor variable list could be found in servo drive user manual.
</td></tr>
</table>

Function	Description
	 **Parameters:** User can enter the desire accessing parameter.  **CANopen:** User can enter the desire access CANopen objects data. Enter the specified accessing position, Index, then specify the flag position, Sub-Index. For example, if Index is set to 60C1h, Sub-Index is set to 02 
Last Value	Indicate the value of selected point.
|ΔY|	Display the time interval of Y axis when enabling the function of vertical line measurement from **Servo Drive Oscilloscope** Toolbar .
Enable Checker	Enable the **Checker Setting** option of this channel.
Checker Setting	Click to display **Stop Condition** window as shown in the following figure.  **Operation:** To select the operator from the drop-down manual as shown in the following figure.

Function	Description
	Stop Value: Enter the stop value. **Capture Type:** Select from **Time** or **Points**. **Capture Time:** Enter the capture time in **Display Area** after the condition is satisfied.

14.4.4 Inverter Device Trace

The inverter device trace is used to collect the real-time inverter device status.

Click **Inverter** > **VFD Function** > **Commissioning** > **Trace** to open the trace view.

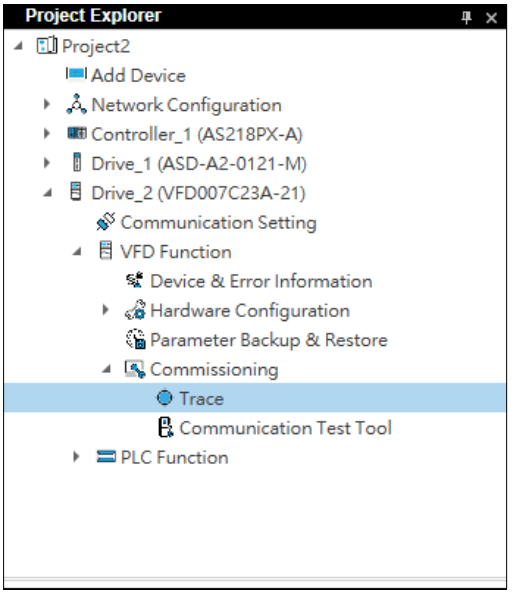


Figure 14 - 48: Trace in Device Tree

The inverter trace view is as below:

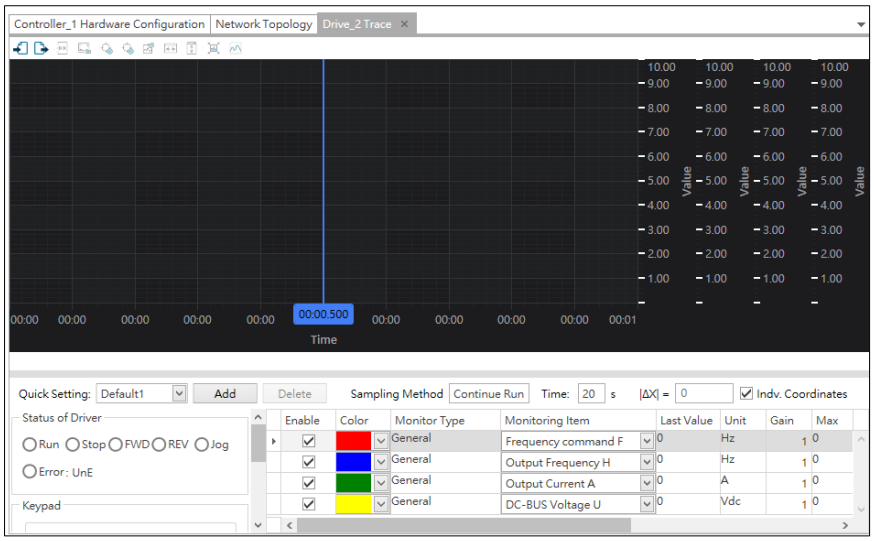


Figure 14 - 49: Inverter Trace

The inverter trace includes the following function blocks:

- Toolbar
- Canvas
- Monitoring Channel Setting
- Status of Driver
- Keypad

14.4.4.1 Toolbar

Toolbar is as below:

Figure 14 - 50: Toolbar

The toolbar functions of trace are described in the following table:

Icon	Description
	Open sampling setting and points in the designated path. Click and the **Open** window displays. (*.vfdscp file)
	Save sampling setting and points in the designated path. Click and the **Save as File** window displays. (*.vfdscp file)
	Export points in the designated path. Click and the **Save as File** window displays. (*.csv file)
	Click to take a snapshot of the trend display (*.png, *.jpg & *.bmp)
	Click to start monitoring
	Click to stop monitoring
	Click to clear canvas
	Displays two vertical lines to measure the distance between waveforms
	Displays two horizontal lines to measure the distance between waveforms
	Zoom to the best size
	Click to open the **FFT Analysis** window

14.4.4.2 Canvas

The real-time display of the selected monitoring channel is as shown in the following figure:

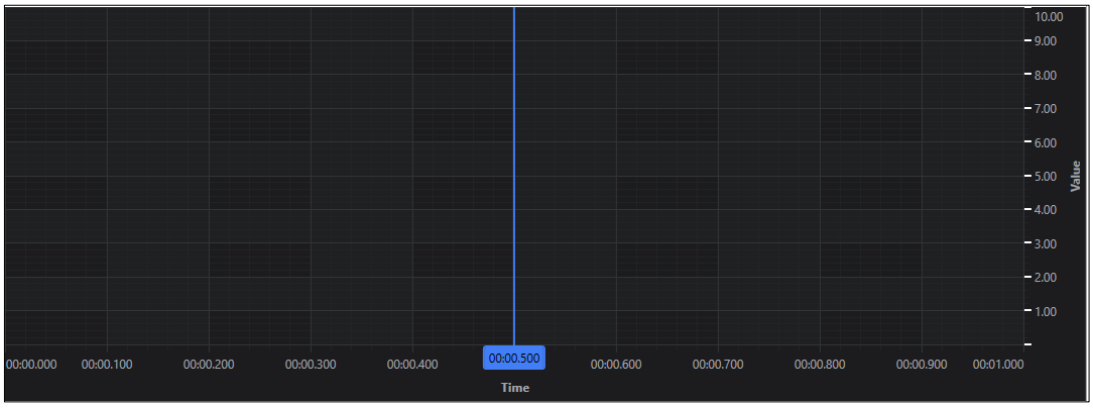
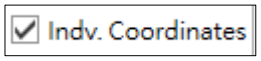


Figure 14 - 51: Inverter Trace

The X axis represents time. The unit is sec.

The Y axis represents the data on the right of the view, which corresponds to the user-selected channels.

At initialization, there is only one Y axis, and it is set to be shared.

It is suitable to monitor the items that have the same unit; however, if the items that have different units, enabling 

Indv. Coordinates above the monitoring channel setting area is recommended.

Figure 14 - 52: Respective coordinate system

14.4.4.3 Monitoring Setting

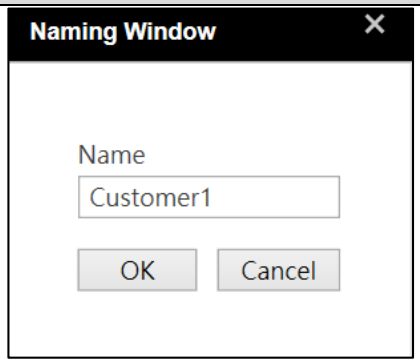
Monitoring channel setting is below the canvas.

The canvas condition setting function is provided above monitoring channel setting.

Figure 14 - 53: Set Trace Condition

The functions are described in the following table:

Function	Description
Quick Setting	Save the set monitoring channel list. Click **Add** to add the selected monitoring channels to the list. Click it and the **Naming Window** pops up, where user enters a name. User can select the saved monitoring list from the drop-down list. Click **Delete** to delete the selected list.

Function	Description
	
Sampling Method	Support continuous sampling
Time	Adjust the resolution of the time axis (the range of the X axis). The length of the canvas can be adjusted according to user input. The setting range is 1 ~ 20 sec.
|ΔX|	Displays the time interval of the X axis. Enable this function from toolbar .
indv. Coordinates	Make each channel corresponds to the individual Y axis of its own.

Keypad is on the bottom left of monitoring channel setting.

The indicator shows the current inverter status.

Keypad can monitor the current device status & the related status of CANopen, and it can give inverter commands.

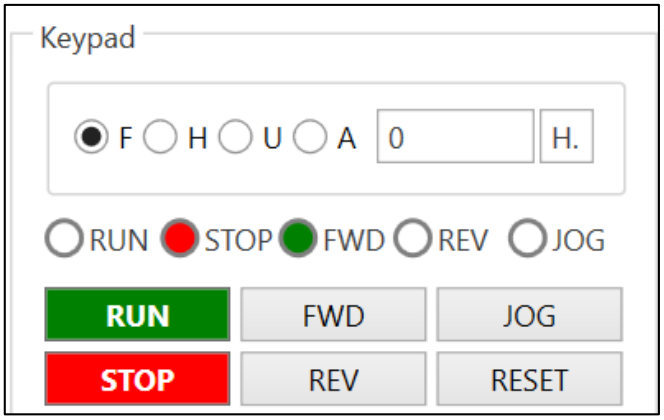


Figure 14 - 54: Keypad

Channel setting is on the bottom right of monitoring channel setting.

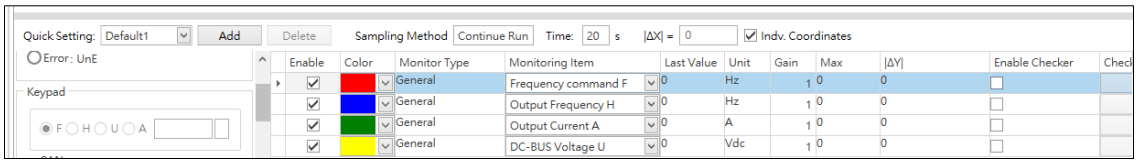
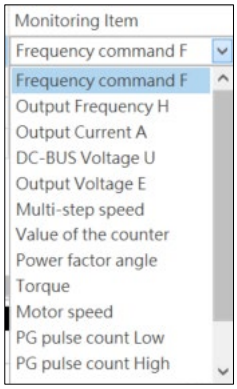
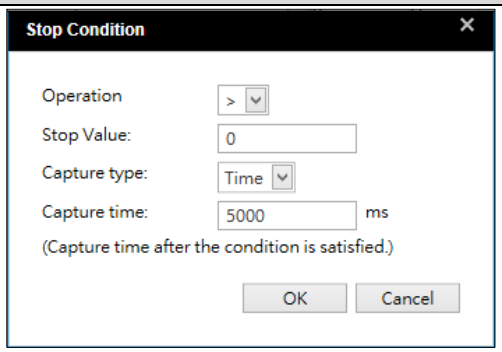


Figure 14 - 55: Trace Monitoring Setting

The functions are described in the following table:

Function	Description
Enable	Select a channel
Color	Select a color of channel
Monitor Type	Support the general monitoring items
Monitoring Item	Read the built-in monitoring items. User can choose monitoring items from the drop-down list as below: 
Last Value	Displays the latest value of channel
Unit	Unit of monitoring item
Gain	Move the waveform position according to gain
Max	Displays the Max. value of the waveform
|ΔY|	Displays spacing between the Y axis and measuring lines. Enable this function from toolbar .
Enable Checker	Enable **Checker Setting** function
Checker Setting	Click **Set** button to open the **Stop Condition** window:

Function	Description
	 **Operation** Choose from the drop-down list. Operators are as follows: **Stop Value** Enter the stop value **Capture Type** Select **Time** or **Points** **Capture Time** When the compare condition holds, capture stops according to the input time

14.4.4.4 Keypad

Keypad is used to control inverter, adjust speed, monitor status, and read/write parameters. To use keypad to adjust speed and start/stop inverter, please set frequency command source and operation command source parameters to be controlled by communication. Take VFD-C2000 Plus as an example, parameters P0.20 and P0.21 need to be set as 1 and 2 respectively. Parameters can be different by different device series. Please refer to the corresponding device series documents.

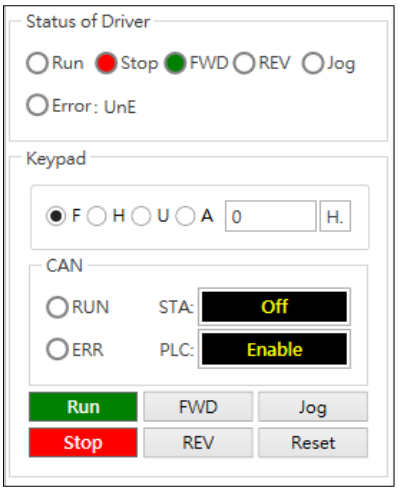
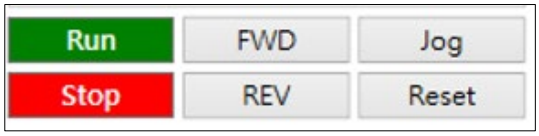


Figure 14 - 56: Keypad

The functions are described in the following table:

Function	Description
F	Displays command frequency In the editing area, double click to enter the command frequency
H	Displays output frequency
U	By default, it displays DC-BUS voltage.
A	Displays output current
	Run/Stop: Inverter runs and stops. FWD/REV: Forward operation of inverter, and reverse operation of inverter Jog: Click it to enable the function Reset: Clear and reset the error code

14.5 Communication Test Tools

A command transceiver is based on Modbus communication. During commissioning testing, it can quickly read/write the related data of inverter (or the built-in PLC) through Modbus communication. Before executing the function, please make sure

that the communication setting of the device has been successfully connected.

Navigate **Inverter Device** (e.g. MH300) **> VFD function > Commissioning > Communication Test Tool**.

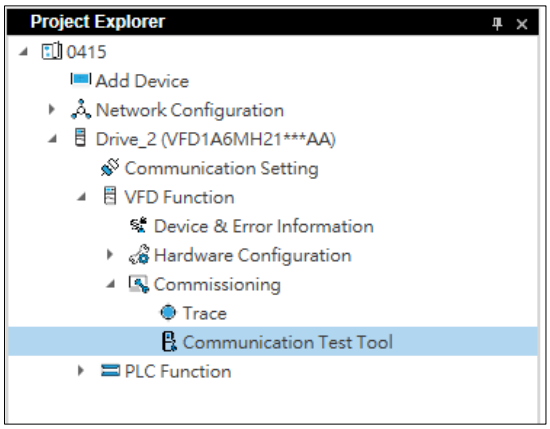


Figure 14 - 57: Communication Test Tool in Device Tree

The tab of Communication Test Tool is as shown in the following figure.

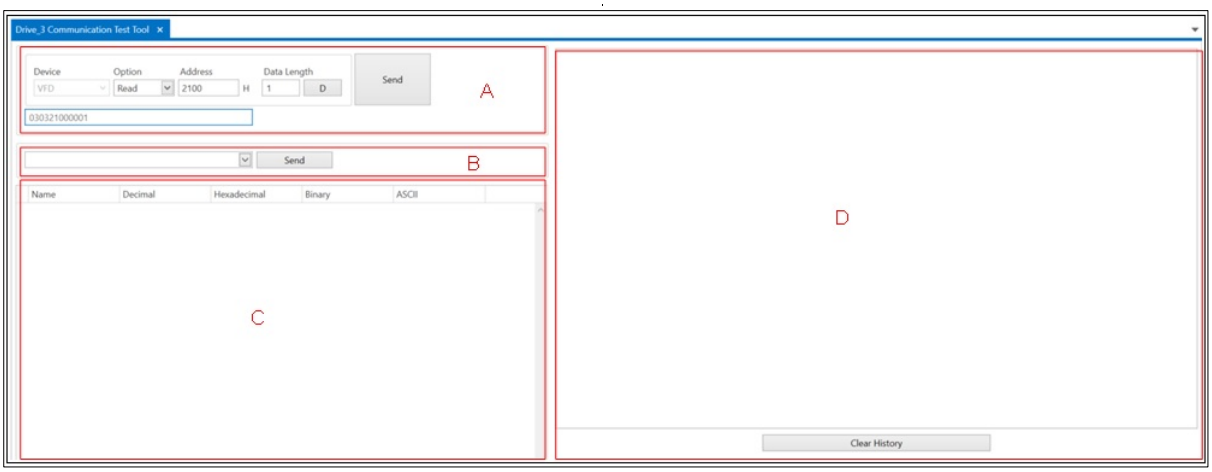


Figure 14 - 58: Communication Test Tool Tab

The view of Communication Test Tool includes the following function blocks:

A. Communication format setting (form)

B. Communication format setting (input box)

C. Data reading result

D. Delivering history

The functions are described as follows:

Function	Description
Device	Select a device user wants to read/write. User can choose this inverter or its built-in PLC. For choosing PLC, please check the **Enable Built-in PLC Connection** box. • VFD • PLC
Option	Choose a command to the device: • Read • Write
Address	Hex. address of read/write Modbus
Data Length	When selecting “read the address data”, assign the data length.
Data (Text box)	When selecting “write the address data”, assign the data content.

Example:

Please refer to the steps below to perform the read/write testing of frequency command functions to VFD 2001H address.

1. Device: VFD

   Option: Read

   Address: 2001

   Data Length: 1

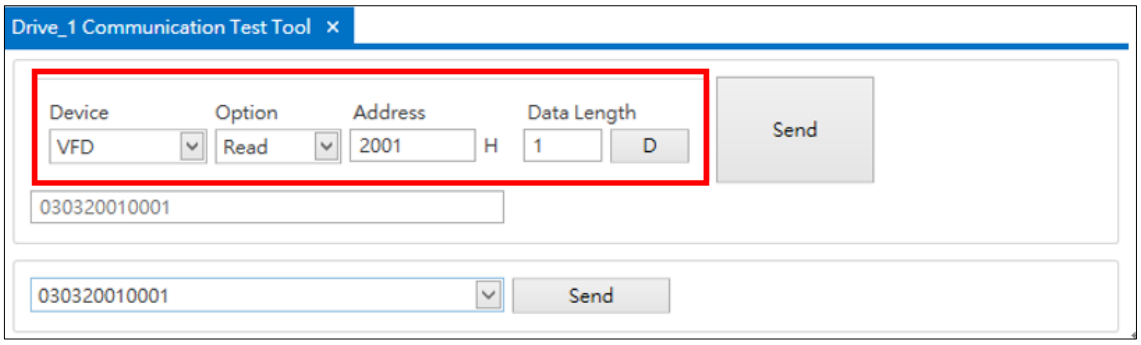


Figure 14 - 59: Communication Test Tool Setting (Step 1)

2. After filling in the columns, user can see the Modbus command value in the below text box.

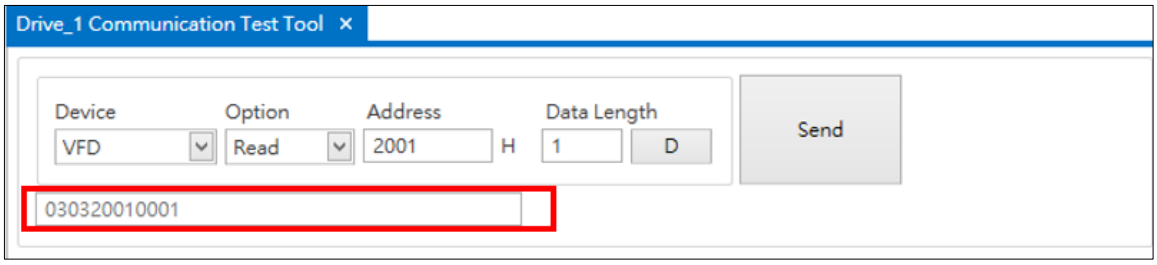


Figure 14 - 60: Communication Test Tool Setting (Step 2)

3. Confirm the value and click **Send**

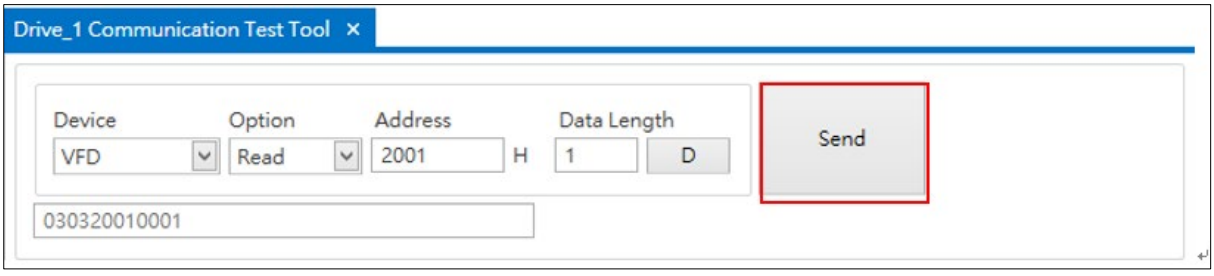


Figure 14 - 61: Communication Test Tool Setting (Step 3)

4. Data reading result shows the 2001H address content value (currently 0) in different ways. On the right, the delivering history block shows the delivery record.

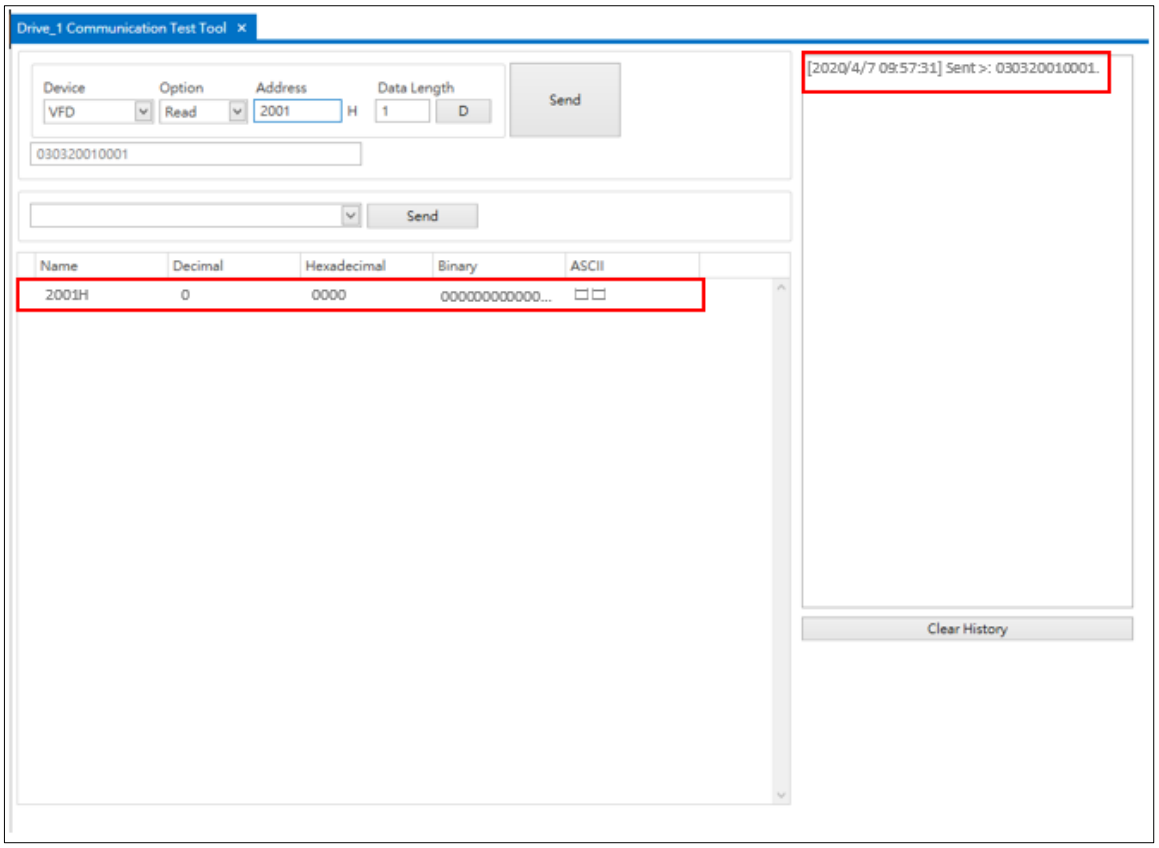


Figure 14 - 62: Communication Test Tool Setting (Step 4)

5. Change the 2001H address content to 55

   Device: VFD

   Option: Write

   Address: 2001

   Data Length: 55 (D: decimal) or 37 (H: hex)

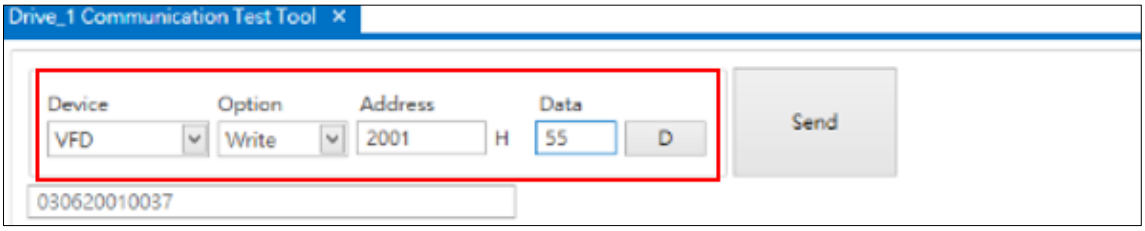


Figure 14 - 63: Communication Test Tool Setting (Step 5)

6. After filling in the columns, user can see the Modbus command value in the below text box.

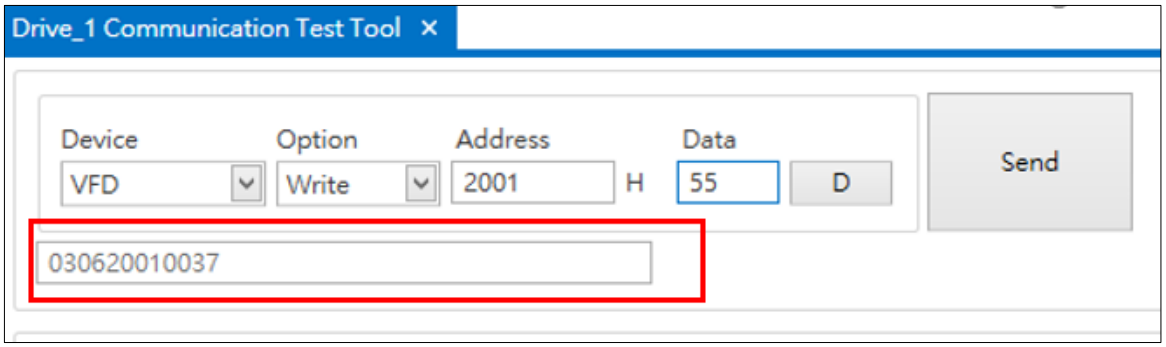


Figure 14 - 64: Communication Test Tool Setting (Step 6)

7. Confirm the value and click **Send**. A new delivering record will be added on the right.

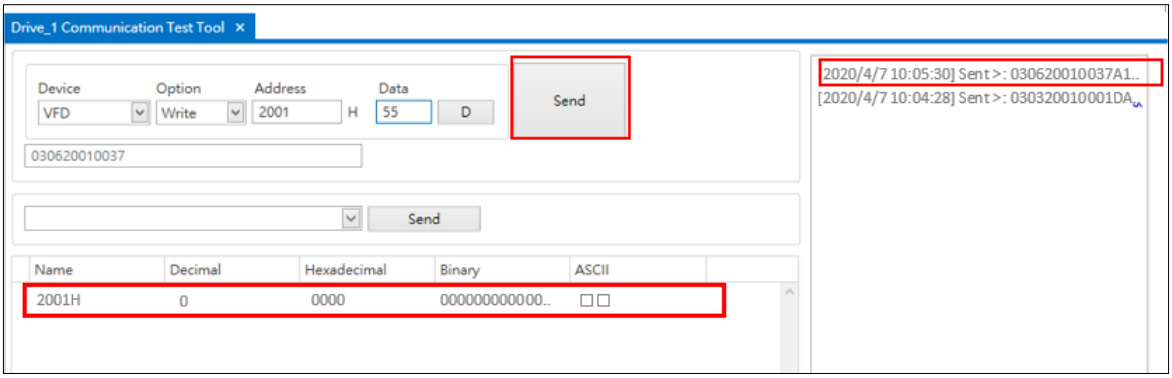


Figure 14 - 65: Communication Test Tool Setting (Step 7)

8. Repeat Step 1 to Step 4, and make sure the current 2001H content value is 55, which represents successful write.

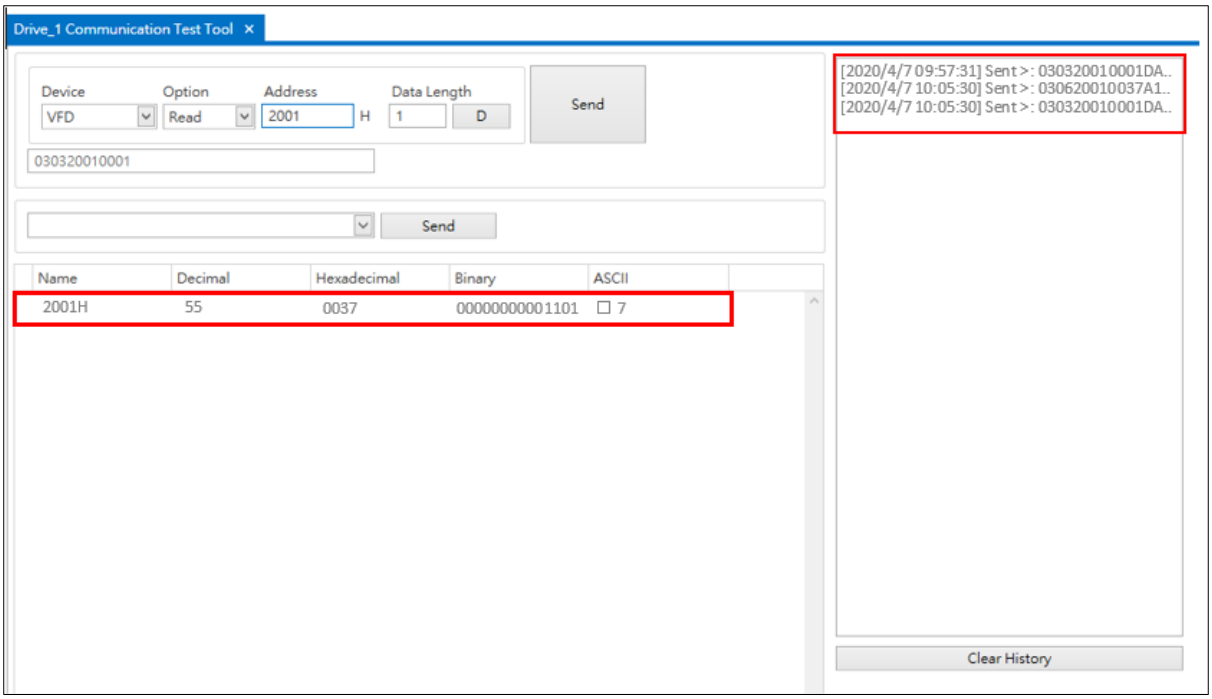


Figure 14 - 66: Communication Test Tool Setting (Step 8)

9. If user want to write value 70 (decimal) to 2001H, (1) in addition to change the column value, user can also (2) directly enter 030320010001 in the second text box. (3) Moreover, user can double-click on a record in the delivery history block. The record will be loaded to the text box.

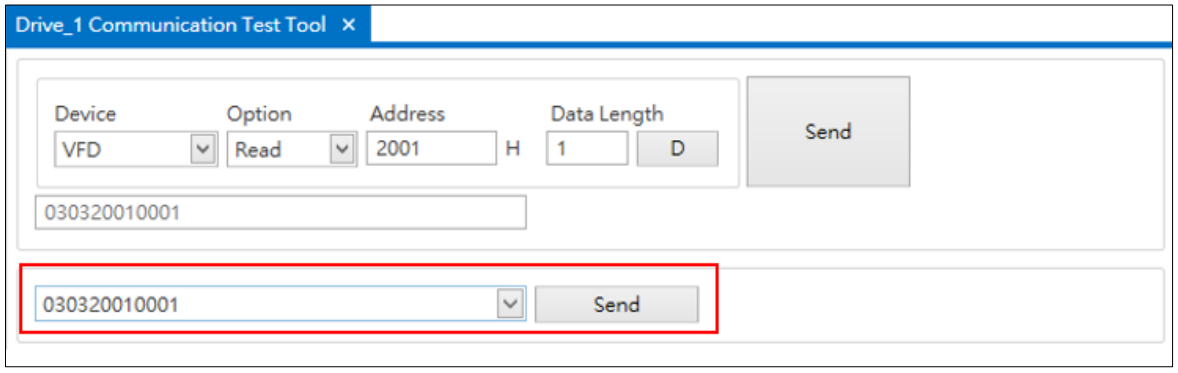


Figure 14 - 67: Communication Test Tool Setting (Step 9)

10. Confirm the value and click **Send**.

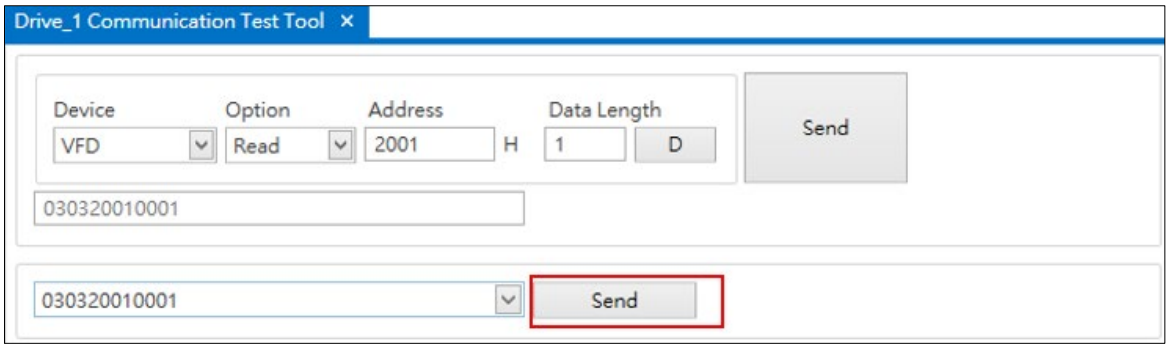


Figure 14 - 68: Communication Test Tool Setting (Step 10)

11. Enter 030320010001, as current 2001H content value, in the second text box. The value 70 represents success.

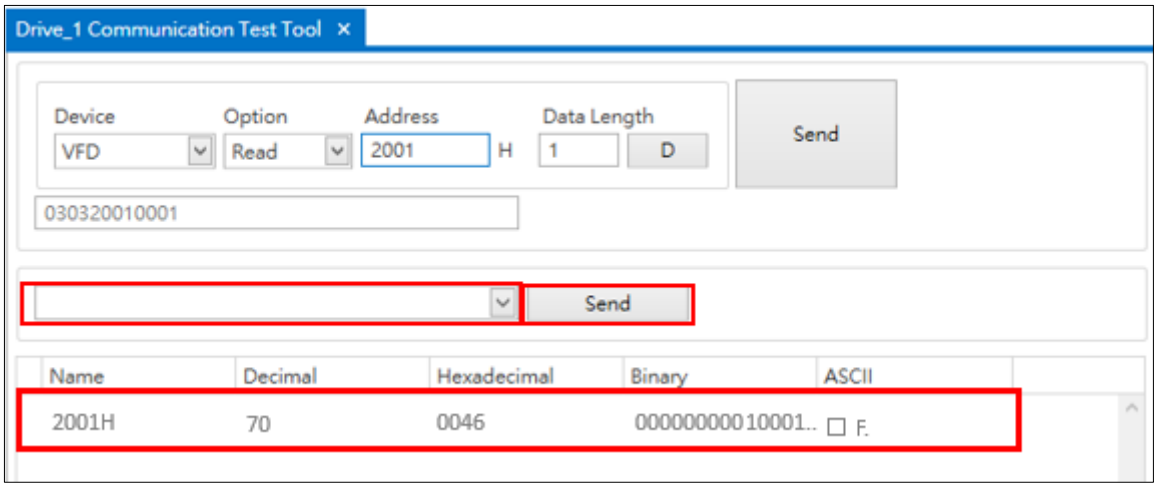


Figure 14 - 69: Communication Test Tool Setting (Step 11)

Chapter 15: Temperature control tool

15.1 Monitor

The user can use the temperature monitor to observe the set temperature and the current temperature value through the monitoring screen, and can also directly edit the control method of the temperature controller online and adjust the related temperature control parameters.

This function can be found in Adaptation Tools> Temperature Monitor.

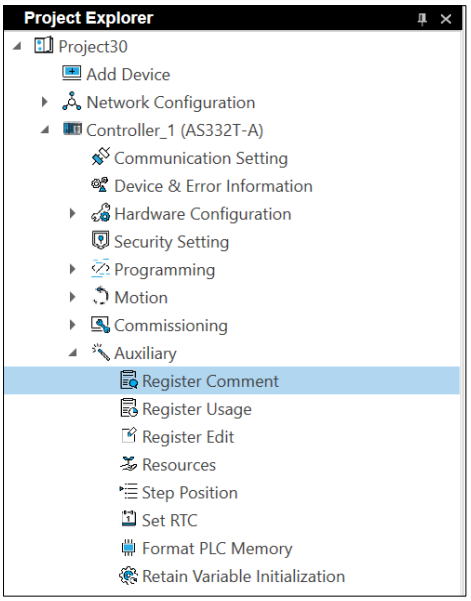


Figure 15 - 1: Project Tree Node-Temperature Monitor

15.1.1 Turn on the thermostat monitoring function

Follow the procedure below to turn on the online monitoring of the thermostat:

1. Click Main Toolbar> Online.

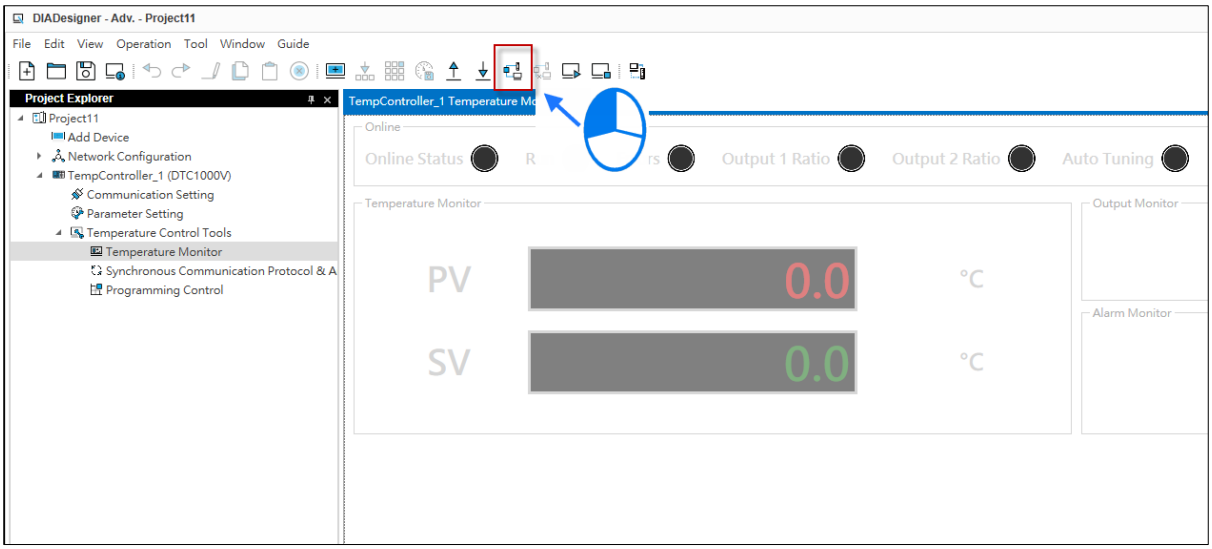


Figure 15 - 2: Online function

Result: The online monitoring screen has been opened, as shown in the figure below.

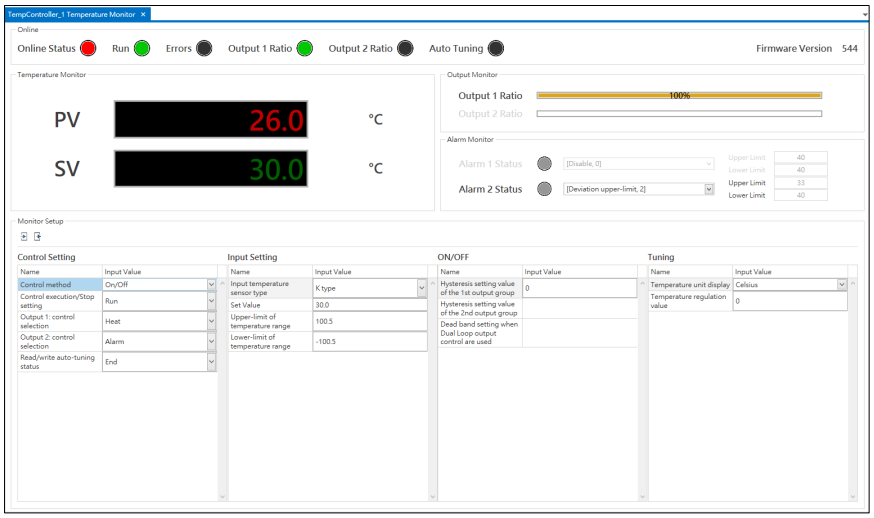


Figure 15 - 3: Online monitoring screen

15.1.2 Monitor Screen

1. Temperature Monitoring

   The user can see the temperature measured by the current devices temperature controller, referred to as PV(Present Value), and the set target

temperature, referred to as SV(Setting Value) here.

Figure 15 - 4: Temperature Monitoring screen

2. Output monitoring

   The user can freely set the output type, and use this block to know the current heating/cooling output status or output percentage.

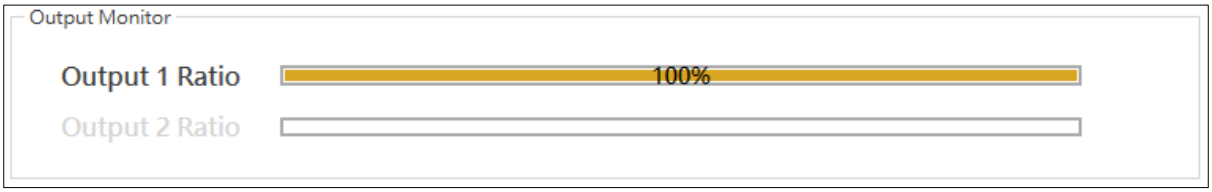


Figure 15 - 5: Output-monitoring screen

3. Alarm monitoring

   The user can freely set the alarm mode through the drop-down menu, and can also set the upper/lower temperature limit of the alarm, and learn the current alarm status of the device from this block.

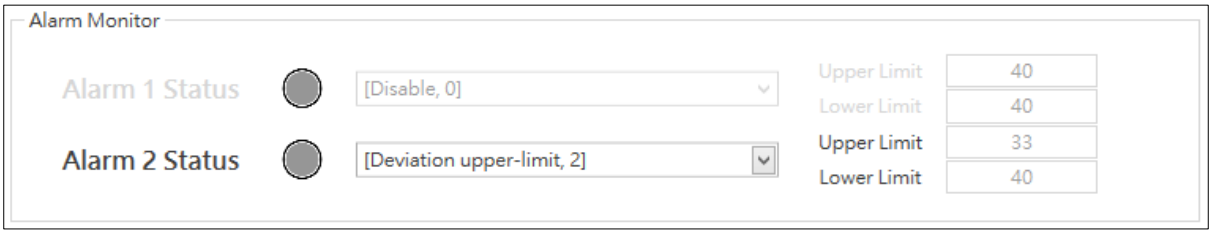


Figure 15 - 6: Alarm monitoring

15.1.3 Monitoring settings

There is a toolbar in the monitoring setting block, the icon is as follows:

Figure 15 - 7: Diagram toolbar

The functions in the toolbar are described in the following table:

Features	Description
	Parameter import function. After clicking, an open window will be displayed for the user to select the parameter file to be imported.
	Parameter export function. After clicking, the Save New File As window will be displayed for the user to select the storage location where the parameter file is expected to be exported.

- Control Settings

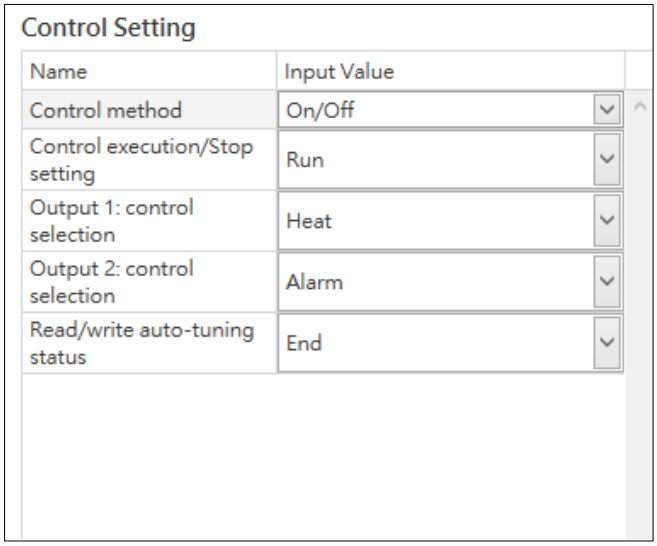


Figure 15 - 8: Control settings

Features	Description
Way to control	1. PID: Through PID control parameters such as proportional, integral and derivative, it can output precise temperature control. 2. On/Off control: switch control, switch control when the temperature reaches the set condition. 3. Manual control: Set the temperature control with a fixed output. 4. PID programmable: for user programmable PID temperature control.
Control execution/stop setting	1. Implementation 2. Stop 3. The Program stops 4. Program Pause
Output-control option	1. Heating 2. Cooling 3. Alarm 4. Proportional output (applicable to equipment models C and L)
Output two control	1. Heating 2. Cooling 3. Alarm
Read and write auto tuning status	1. Auto tuning start: When switching to start, the AT light will flash and PID control parameters will start to be calculated. After tuning is completed, it will automatically switch to "Auto tuning stop". 2. Auto tuning stops

- Input settings

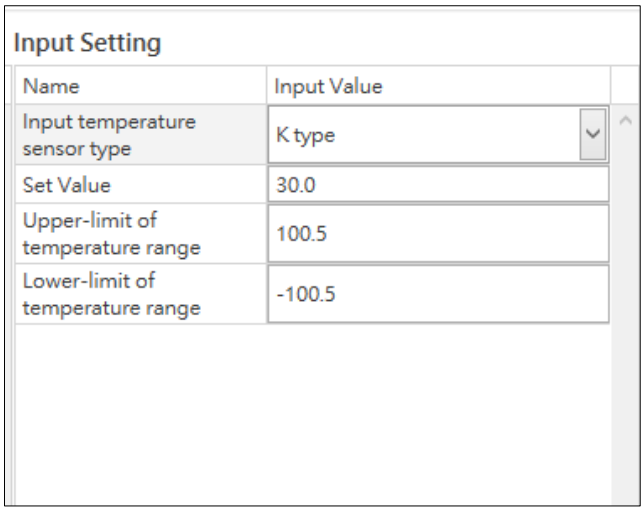


Figure 15 - 9: Input settings

Features	Description
Input sensor type	1. Thermocouple pair K type 2. Thermocouple pair J type 3. Thermocouple pair T type 4. Thermocouple pair E type 5. Thermocouple pair N type 6. Thermocouple pair R type 7. Thermocouple pair S type 8. Thermocouple pair B type 9. Thermocouple pair L type 10. Thermocouple pair U type 11. Thermocouple TXK type 12. Platinum resistance thermometer JPt100 13. Platinum resistance thermometer Pt100 14. 0~5V linear voltage input 15. 0~10V linear voltage input 16. 0~20mA linear current input 17. 4~20mA linear current input 18. 0~50mV linear voltage input
SV set value	Target temperature setting value
Enter the highest value of detection range	SV set value upper limit
Enter the lowest value of the detection range	SV set value lower limit

- PID parameter setting

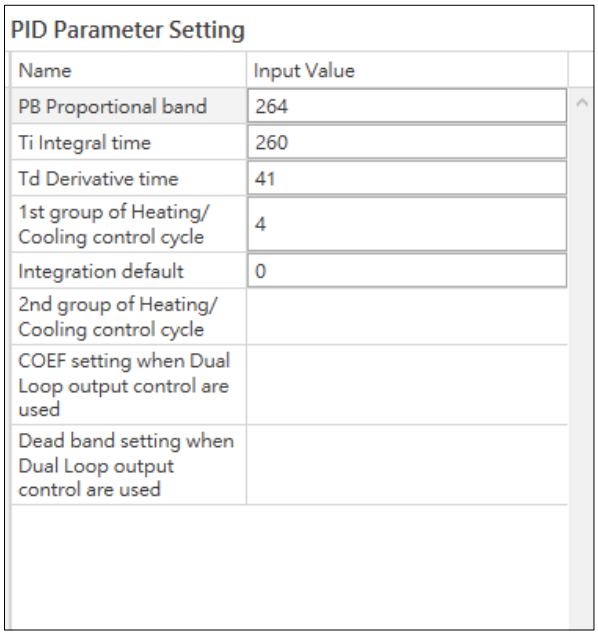
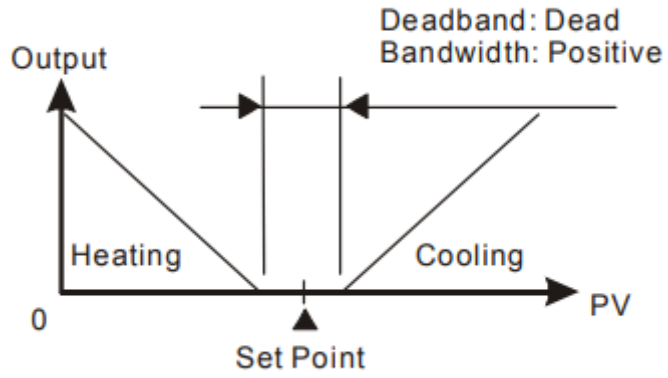
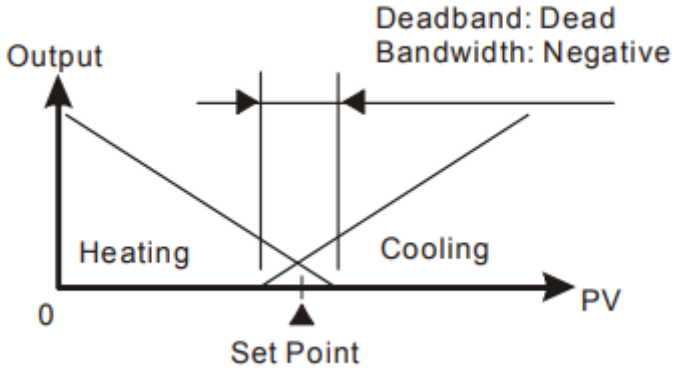


Figure 15 - 10: PAD Parameter setting

Features	Description
PID setting value	1. PB proportional band setting value 2. Ti integral value constant set value 3. Td differential control constant setting value 4. Preset value of integration 5. COEF setting for dual output ***Note**: PID setting value is recommended to be calculated by auto tuning function.*
Control cycle	Control cycle time, unit (seconds). Example: If the output is 60%, the control cycle time is 10 seconds The operation output time occupies 6 seconds, and the stop operation time is 4 seconds.
Dead band setting when dual output	1. Dead band is positive

Features	Description
	 2. Dead band is negative 

- Switch Control

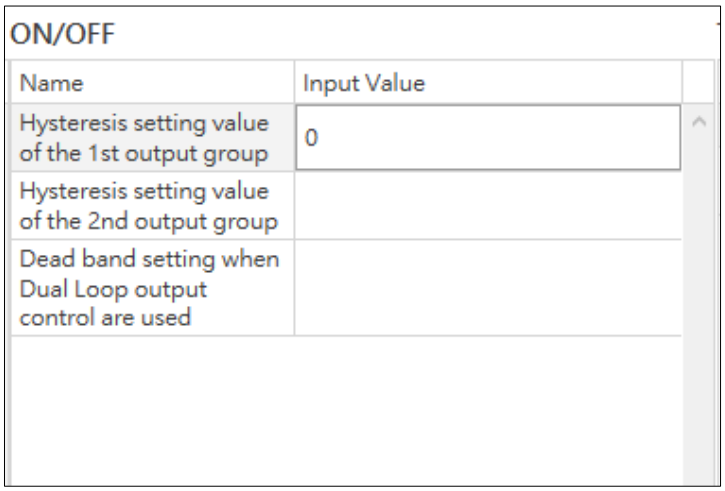
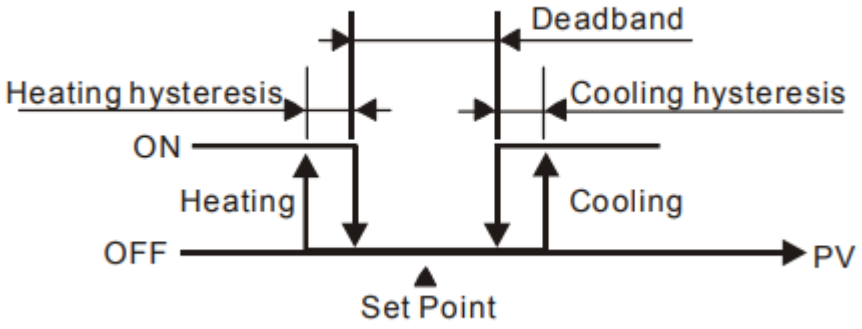


Figure 15 - 11: Switch control

Features	Description
Hysteresis setting value	If it is higher than SV setting value + hysteresis setting value or lower than SV setting value-hysteresis setting value, heating/cooling will be activated.
Dead band setting when dual output	

- Manual output setting

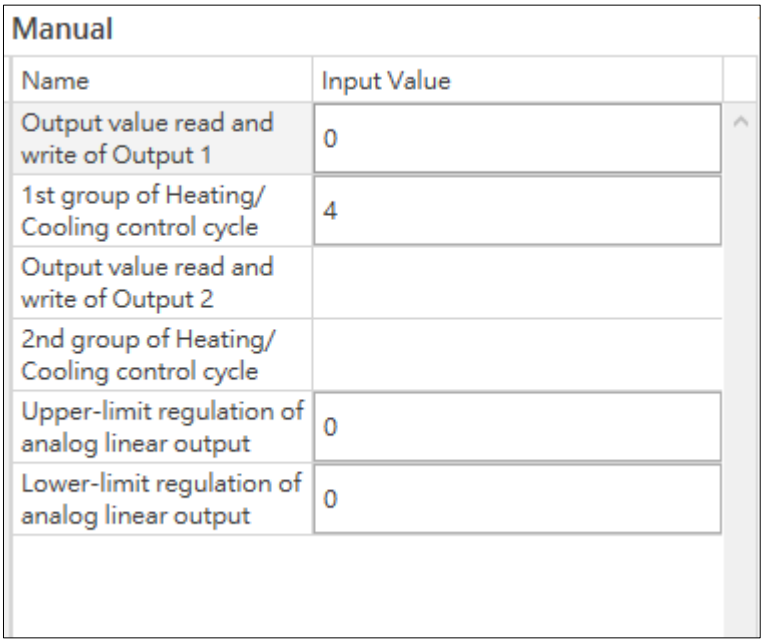


Figure 15 - 12: Manual output setting

Function	Description
Output read and write	Set the percentage of output.
Control cycle	Control cycle time, unit (seconds). Example: If the output is 60%, the control cycle time is 10 seconds

Function	Description
	The operation output time occupies 6 seconds, and the stop operation time is 4 seconds.
Analog linear output upper limit adjustment / offline adjustment	Output upper / lower limit scale adjustment, 2.8μA (1.3mV).

- Tuning

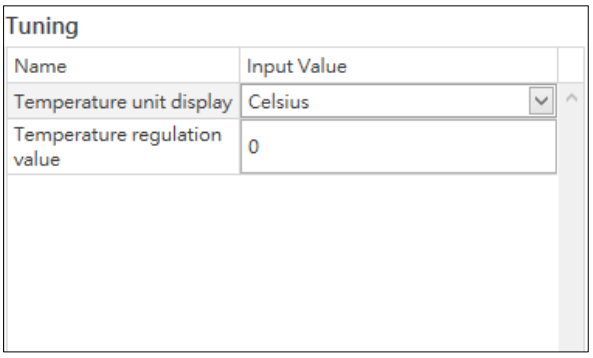


Figure 15 - 13: Set

Features	Description
Temperature unit display selection	1. Fahrenheit 2. Celsius
Temperature error adjustment value	Used to adjust the error temperature

15.2 Program planning

The user can plan the thermostat temperature control program in the program planning function. User can find the program planning function through the thermostat> temperature control tool> program planning.

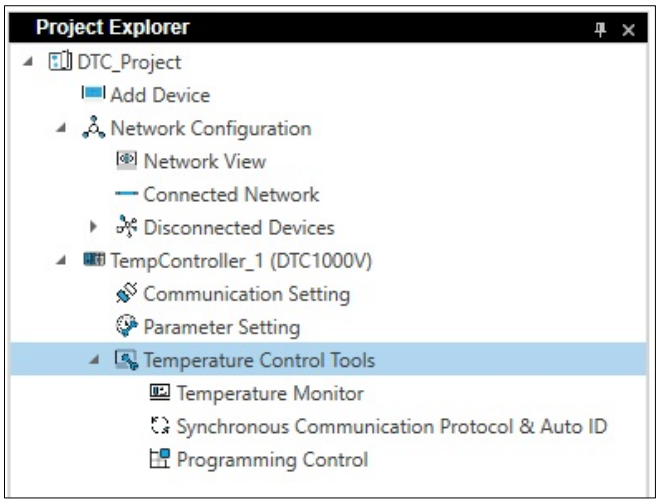


Figure 15 - 14: Program planning location

Double-click the program planning to open the program planning screen.

The program planning screen will be shown as the picture below. In the program planning screen, the upper program editing area can be divided into 8 programs (Pattern0 ~ 7). Each program has 8 steps (Step 0 ~ 7), and each has a link program (Link Pattern), execution loop (Loop Count), and effective step (Step Max.). The lower status display area can display the execution curve after program editing.

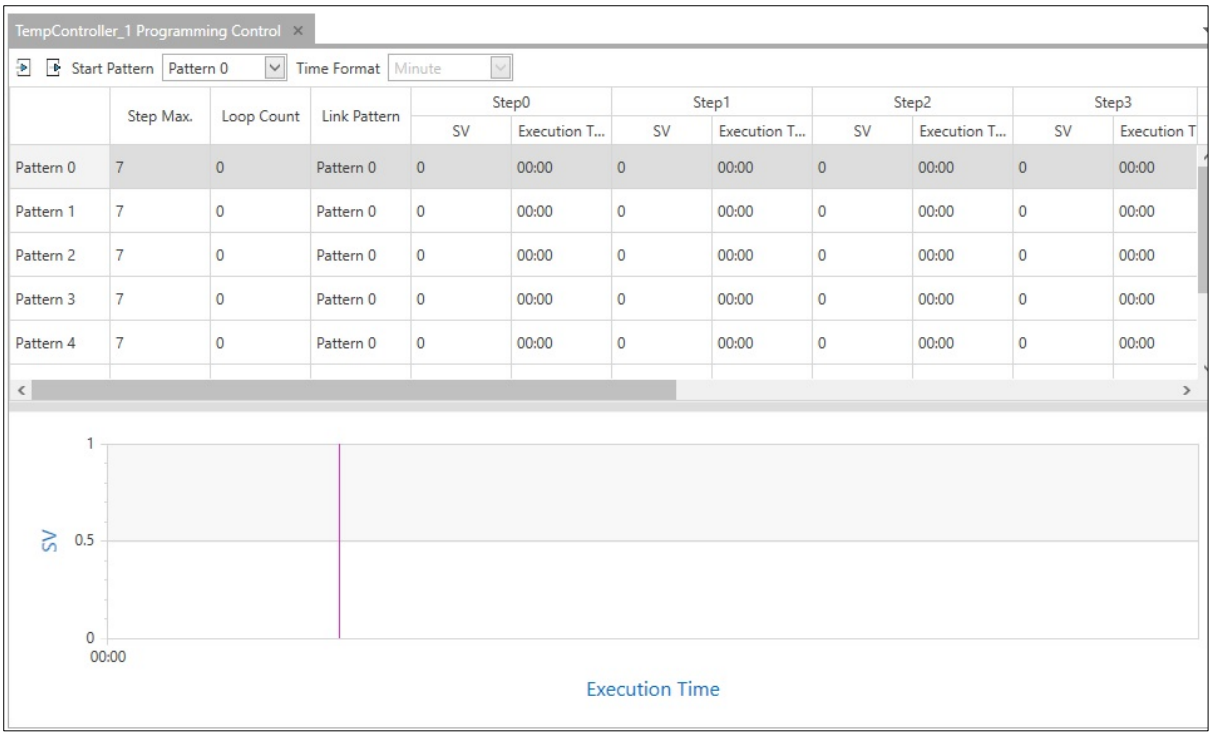


Figure 15 - 15: Program planning

There is a toolbar in the program planning screen to provide user with quick operation program planning. The part of the toolbar is shown in the figure below.

Figure 15 - 16: Program planning toolbar

The toolbar functions of program planning are described in the following table.

Features	Description
	Import function. After clicking, an open window will be displayed to allow the user to select the thermostat program planning file to be imported.
	Export function. After clicking, the Save New File window will be displayed to allow the user to select the storage location where the thermostat program planning file is expected to be exported.

Features	Description
Starting Program	Set the program control from which program to start execution.
Time format	Depending on the model, different time formats of SV can be set. DTC only supports the time format as minutes.

15.2.1 Program and step editing

Edit the program according to the following process:

1. In the program of initial program setting, set the steps, including the setting of two parameters, SV and execution time, which means that the PV value must reach the SV setting value after the execution time.
   Result: The lower status display diagram shows the temperature control curve after program editing, and the screen is as shown in the figure below.

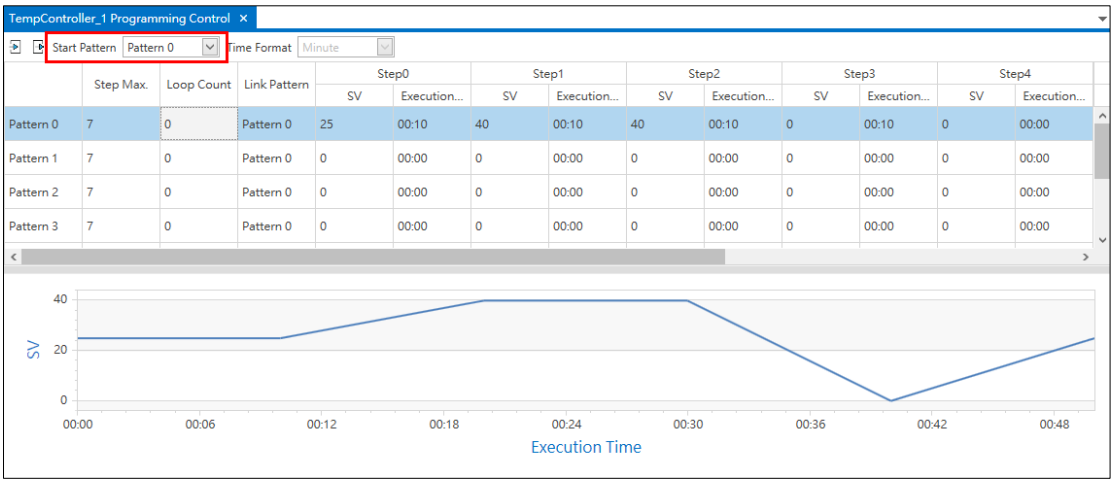


Figure 15 - 17: Single program multi-step editing

2. Set the connection program: the pattern that will be executed after the program is executed. If it is set to a value other than program 0 ~ 7, it means that the program ends after the program is executed, and the target value remains at the setting of the last step.
3. Set loop parameters: the number of additional loops that the program will

execute. For example, if this parameter item is set to 2, it means that the program needs to be executed 2 additional times, even the original one needs to be repeated 3 times in total.

4. Effective steps: the number of steps executed by each pattern (can be set from 0 to 7). For example, if this item is set to 3, it means that the corresponding style will execute step 0 ~ step 3, and the remaining steps will not be executed and cannot be edited.

   Result: The lower status display diagram shows the temperature control curve after program editing, and the screen is as shown in the figure below.

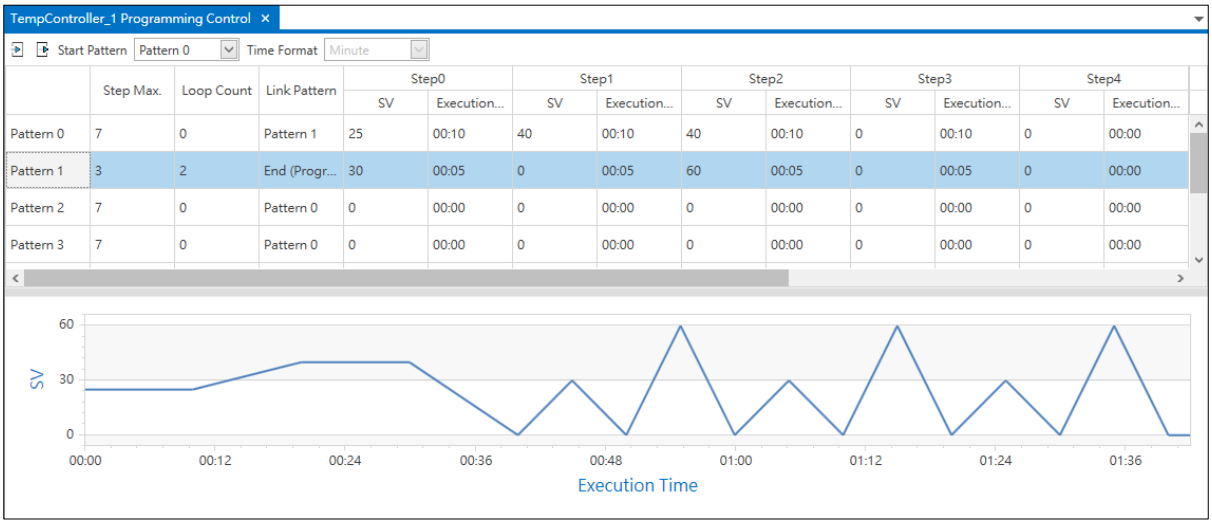


Figure 15 - 18: Multi-program and multi-step editing

15.2.2 Status display area

It can display the effective program temperature control curve after program editing. The horizontal axis is the execution time and the vertical axis is SV. The maximum coordinate value changes automatically according to the maximum time and temperature set by the program. Move the mouse to any position to display the target temperature and estimated execution time at that point, as shown in the figure below.

Figure 15 - 19: state display diagram

15.3 Station number automatic synchronization

The user can use the station number automatic synchronization function to connect all the devices connected together. After the function is turned on, the communication specifications and station numbers of the serially connected devices will be directly changed, and the station number setting will be based on the station number of the first station. After sorting increases.

This function can be enabled or disabled in Adaptation Tools> Automatic synchronization of communication and station number.

15.3.1 Start the station number automatic synchronization function

Follow the procedure below to activate the station number automatic synchronization function:

1. Click Adaptation Tools> Automatic synchronization of communication and station number.
   Result: A window for confirming the opening of automatic synchronization will be displayed, as shown in the figure below.

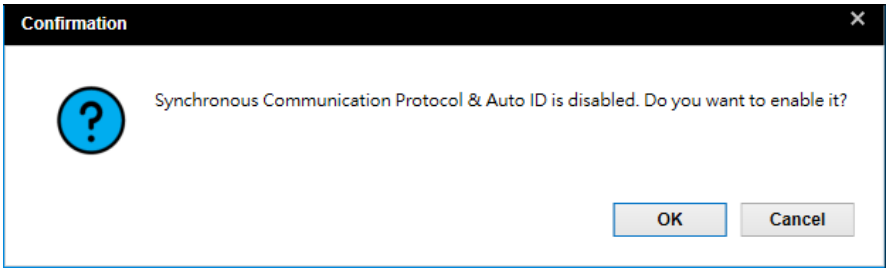


Figure 15 - 20: Confirm that the automatic synchronization function is turned on

2. Click OK
   Result: A window that has successfully opened the automatic synchronization function will be displayed, and the user will be notified to re-power on, as shown in the figure below.

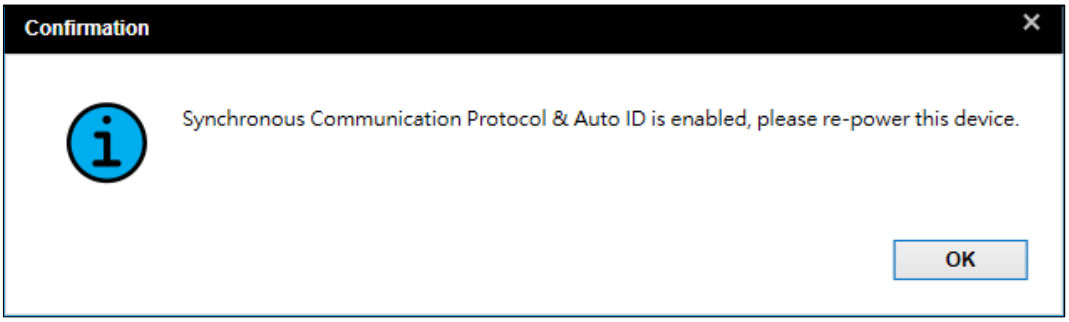


Figure 15 - 21: Re-power to confirm

3. Click OK and power on again.
   Result: The automatic synchronization function has been successfully turned on.

15.3.2 Turn off the station number automatic synchronization function

Follow the procedure below to turn off the station number automatic synchronization function:

1. Click **Adaptation Tools**> **Automatic synchronization of communication** and station number.

   Result: A confirmation window will be displayed to close the automatic synchronization, as shown in the figure below.

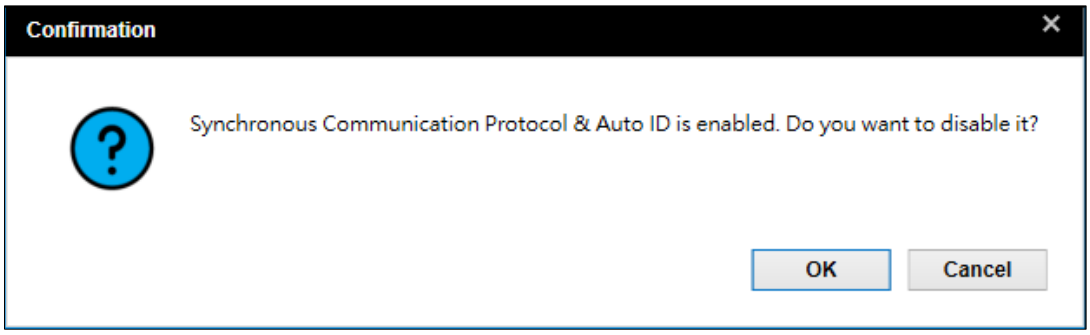


Figure 15 - 22: Auto-sync function close confirmation

2. Click **OK**

   Result: A window with the automatic synchronization function closed will be displayed, as shown in the figure below.

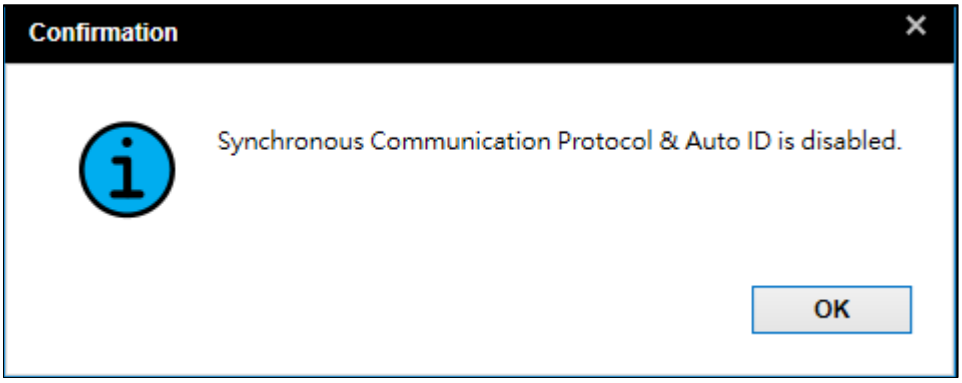


Figure 15 - 23: Automatic synchronization is turned off

3. Click **OK**

   Result: The automatic synchronization function has been successfully turned off.

Chapter 16: Auxiliary Tools

16.1 Register Comment

User can manage the comments for registers with the help of Register Comment.

User can locate register comment in **Controller** > **Auxiliary** > **Register Comment**.

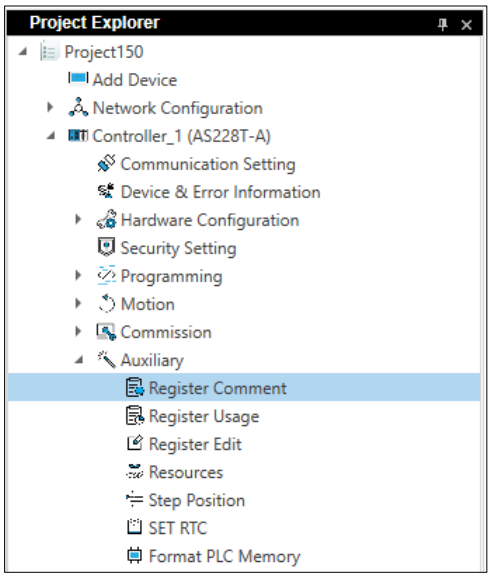


Figure 16 - 1: Register Comment

Double-click on **Register Comment** to open the Register Comment window.

Register Comment window displays as shown in the following figure. **Register Comment** window displays the register names with corresponding comments in the Address and Comments columns respectively.

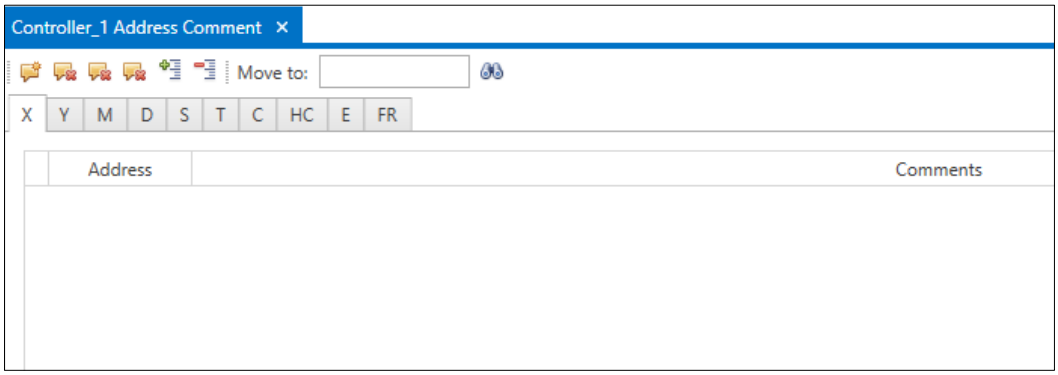


Figure 16 - 2: Register Comment

The **Register Comment** window has a toolbar which helps user to manage the register comments. Toolbar in **Register Comment** window is shown in the following figure.

Figure 16 - 3: Register Comment - Toolbar

The functions in the toolbar in **Register Comment** window are described in the following table:

Function	Description
	Click to add Register Comment. Displays the **Add Register Comment** window
	Click to clear comments of selected items
	Click to clear comments of all items in the page
	Click to clear comments of all items
	Click to expand nodes
	Click to collapse nodes
	Enter a device name to locate
	Click to locate the register whose name is entered in the **Move to** field

User can right click in the **Register Comment** window to open the context menu. Context menu in **Register Comment** window is shown in the following figure.

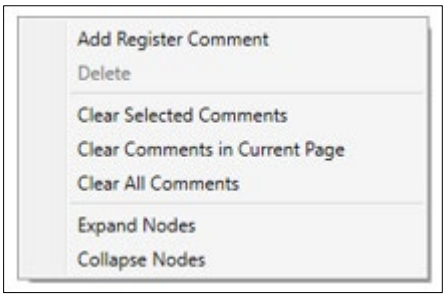


Figure 16 - 4: Register Comment window - context menu

The functions in context menu in **Register Comment** window are described in the following table:

Function	Description
Add Register Comment	Click to add Register Comment. Displays the **Add Register Comment** window.
Delete	Click to delete the item from Register Comment window.
Clear Selected Comments	Click to clear comments of selected items.
Clear Comments in Current Page	Click to clear comments of all items in the page.
Clear All Comments	Click to clear comments of all items.
Expand Nodes	Click to expand nodes.
Collapse Nodes	Click to collapse nodes.

16.1.1 Adding items in Address Comment window

Follow these steps to add items in the Register Comment window:

1. Click icon in the toolbar or

   Right-click in the **Register Comment** window to open context menu and click on **Add Register Comment**.

   Result: **Add Register Comment** window displays as shown in the following figure.

Figure 16 - 5: Add Register Comment

2. Select the **Register**, **Address**, **Bit Number** and **Count** of items needed.
3. Click on **OK** button.

   Result: Registers are added to the **Register Comment** window.

   Example:

   An example of adding X Registers from 0.0 to 0.5 is shown in the following figure.

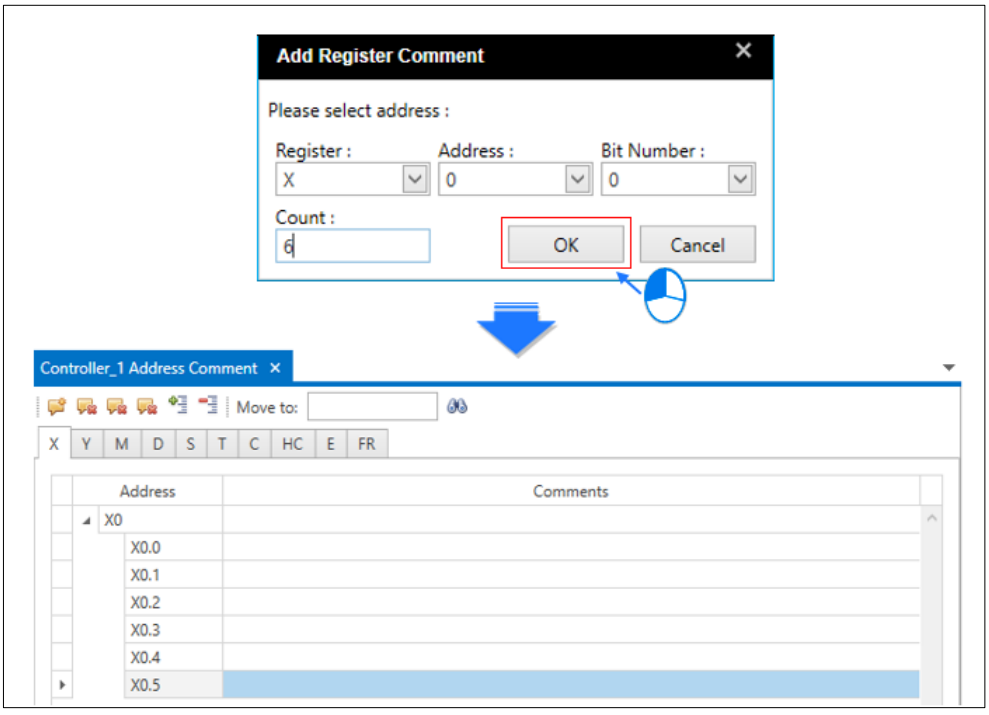


Figure 16 - 6: Add Register Comment

4. User can enter the comments for respective registers in the **Comments** column.

16.1.2 Deleting items in Register Comment window

Follow these steps to delete items in the Register Comment window:

1. Select the item to be deleted.
2. Right-click on the Register and Address which needs to be deleted from the window to open its context menu and select **Delete**

   or

   Press **Delete** key in the keyboard

   Result: Item is deleted from the **Register Comment** window

16.1.3 Deleting Register Comments

Follow these steps to delete the Register Comments:

1. Select the Register(s)
2. Click icon in the toolbar or

   Right-click on the Register Address whose comments has to be deleted and click **Clear Selected Comments** on the context menu.

   Result: **Comments** of the selected Address is deleted

16.1.4 Deleting comments for selected Register type

Follow these steps to delete comments of all Register Addresses in a Register type:

1. Select the Register type (example X, Y, M etc.).
2. Click on icon in the icon toolbar or

   Right-click in the Address Comment window and click **Clear Comments in the Current Page** on the context menu.

Result: **Comments** of all Registers in the selected Register type is deleted.

16.1.5 Deleting All Comments

Follow these steps to delete the comments for all Addresses in the Register Comments:

1. Click on icon in the icon toolbar or

   Right-click in the Address Comment window and click **Clear All Comments** on the context menu.

   Result: Comments of all Address types are deleted.

16.2 Register Usage

User can refer the usage of Registers in the project with the help of Register Usage in DIADesigner. User can locate register usage in **Controller > Auxiliary > Register Usage**.

Double-click on **Register Usage** to open the Register Usage window as shown in the following figure.

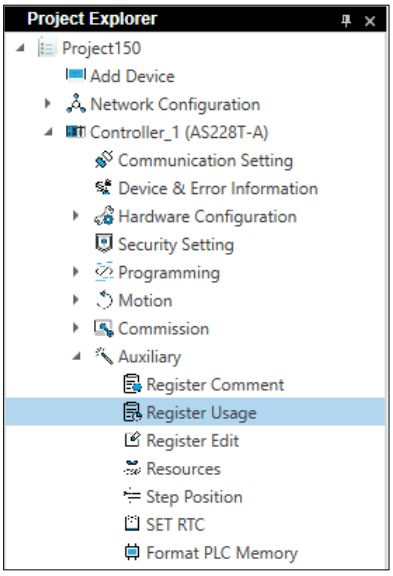


Figure 16 - 7: Register Usage in Project Explorer

The **Register Usage** window displays as shown in the following figure.

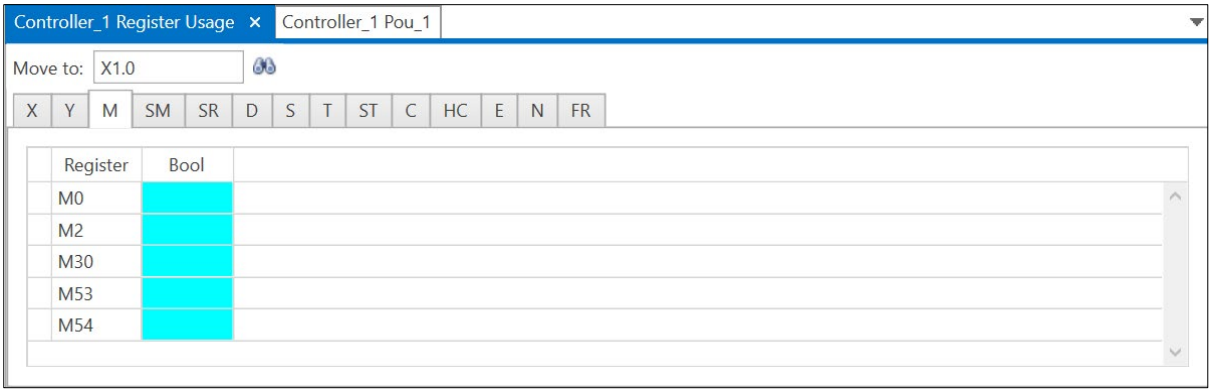


Figure 16 - 8: Register Usage

The **Register Usage** window consists of a toolbar and two panes – top and bottom. Toolbar has a Move to function to locate the particular register. Top pane displays the Register and Addresses. Bottom pane displays the locations where they are used. The Address which are used in the POU, Global Variable – Default Tag Table etc. display in cells filled with blue color.

Example:

The addresses – X0.0, X0.5 in the below figure are used. User can click on the blue color cells; based on the usage, the bottom panel will display the locations where the addresses are used. An example of address X0.0 usage location is shown in the following figure.

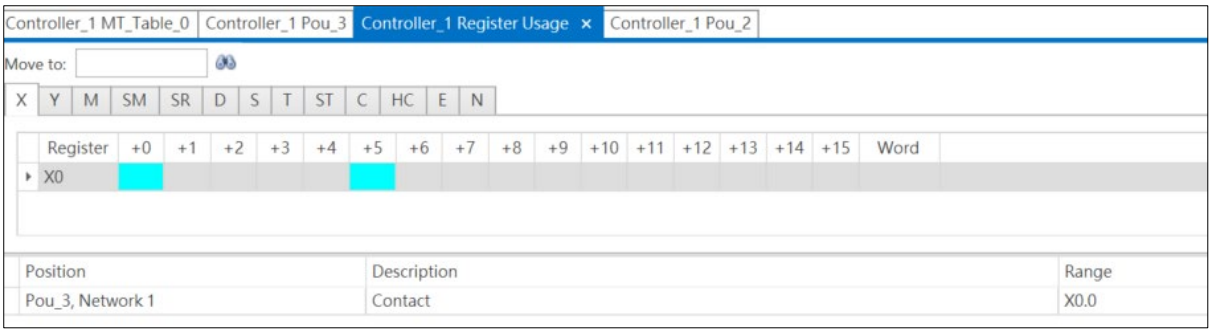


Figure 16 - 9: Address Usage

User can click on any address (blue-color cells) in the upper pane of **Register Usage** and usages of this register displays in the bottom pane. Double-click on a particular line in the bottom pane to navigate to the respective location. Sample navigation is shown in the following figure for Register *X* with Address *0.0* in POU *PGM1*.

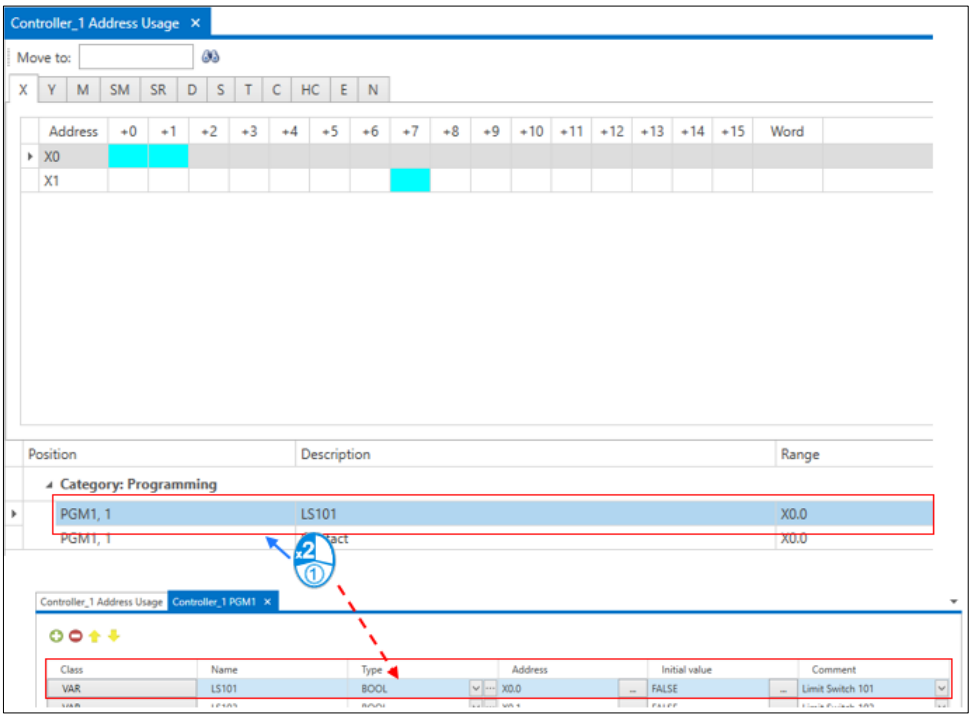


Figure 16 - 10: Register Usage

User can locate a particular address in the Register Usage window with the help of **Move to** function located in the toolbar. Enter an address and click the icon to

navigate to that address. An example of navigation to X1.7 is shown in the following figure.

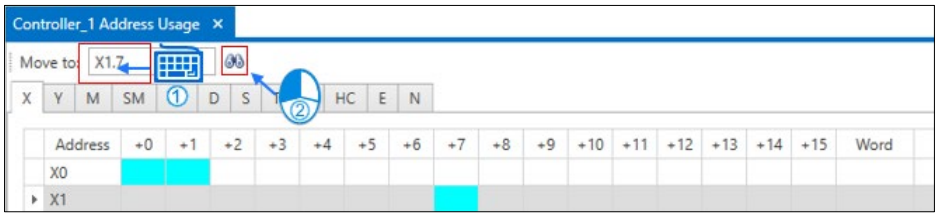


Figure 16 - 11: Navigation in Register Usage window

16.3 Resources

User can keep a check on the memory usage of the controller device with the help of Resources. User can locate the Resources in **Controller** > **Auxiliary** > **Resources**. **Resources** can be located in the **Project Explorer** as shown in the following figure.

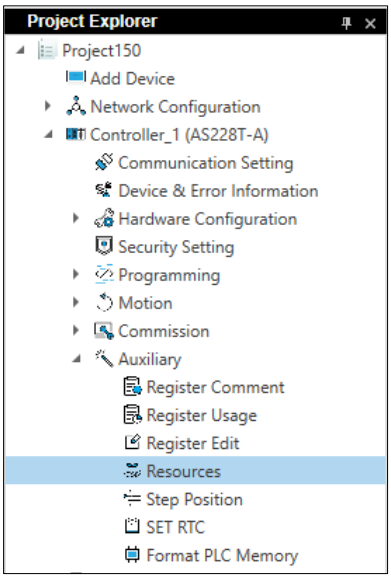


Figure 16 - 12: Resources

Double-click **Resources** to open the Resources window. **Resources** window is shown in the following figure.

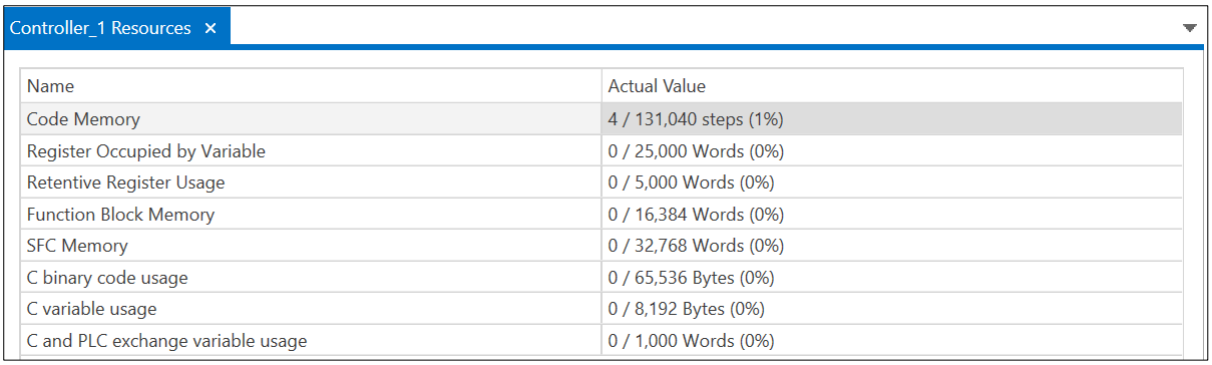


Figure 16 - 13: Resources window

The functions in Resource window are described in the following table:

Function	Description
Code Memory	Percentage of memory used by user programs in Steps
Register Occupied by Variable	Percentage of used retentive register in Words
Retentive Register Usage	Percentage of register occupied by variable declaration in Words
Function Block Memory	Percentage of used Function Block Memory in Words
SFC Memory	Percentage of user SFC Memory in Words
C language execution code usage rate	The usage rate of the C Code execution code in the storage space (64K Byte).
C language variable usage rate	The usage rate of C language variables occupying variable declaration free space (8K Byte).
C language and PLC shared variable utilization rate	The usage rate of the C_Tag_Table variable (1000Words).

16.4 Register Edit

User can edit the values in M, D, T, C, HC and FR registers in PLC by Register Edit. User can read/write value from/to PLC by Register Edit. Register Edit can be located in **Controller** > **Auxiliary** > **Register Edit**.

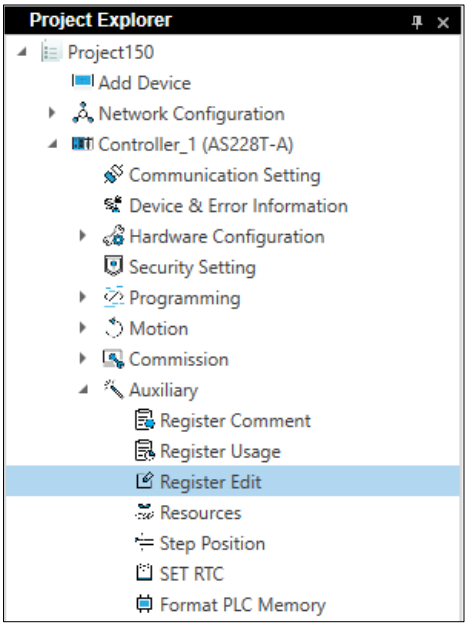


Figure 16 - 14: Register Edit

Double-click on **Register Edit** to open the Register Edit window, as shown below:

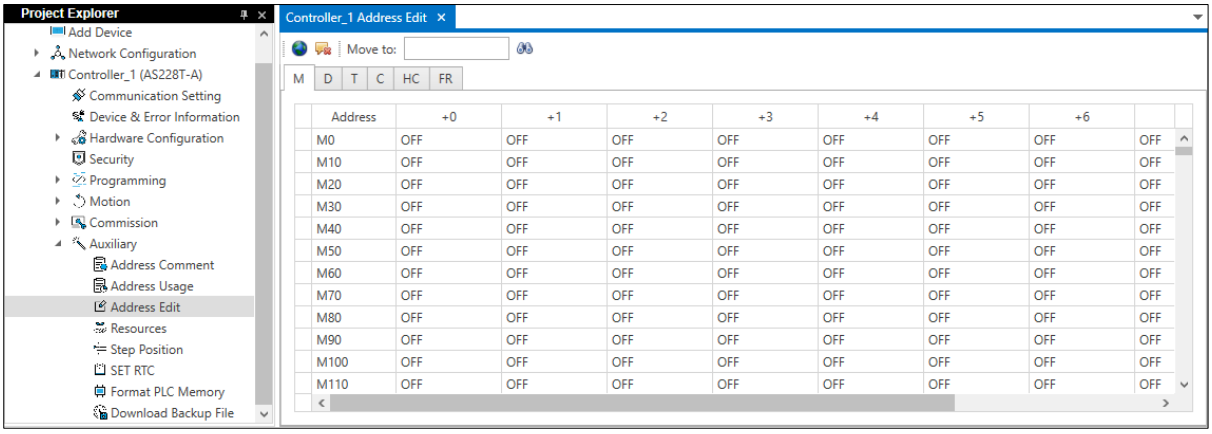


Figure 16 - 15: Address Edit Window

The **Register Edit** window has a toolbar and the register editing area.

16.4.1 Register Edit Toolbar

The toolbar functions in the **Register Edit** window are described in the following table:

Function	Description
	Click to open the **Transfer Setup** window, only available in offline mode
	Click to clear or reset the values of all registers in the *current register* window (e.g. M, D etc.) to 0
	Click to select data format type from the drop-down list: • Decimal • Unsigned Decimal • Hexadecimal • Binary • Floating Point • ASCII ***NOTE**: This field displays for all register types except M. **Decimal** is the default value.*
	Click to select data Length type from the drop-down list: • 16bits • 32bits ***NOTE**: This field is applicable for all register types except M and HC. **16bits** is the default value*.
	Enter the register number to be navigated
	Click to navigate to the location of the Register value entered in the **Move to** field
	Click on the register letter to open the respective register tab

16.4.2 Data format

An example for Decimal, Binary and Hexadecimal format for D0, D1 and D2 registers is shown in the following figure.

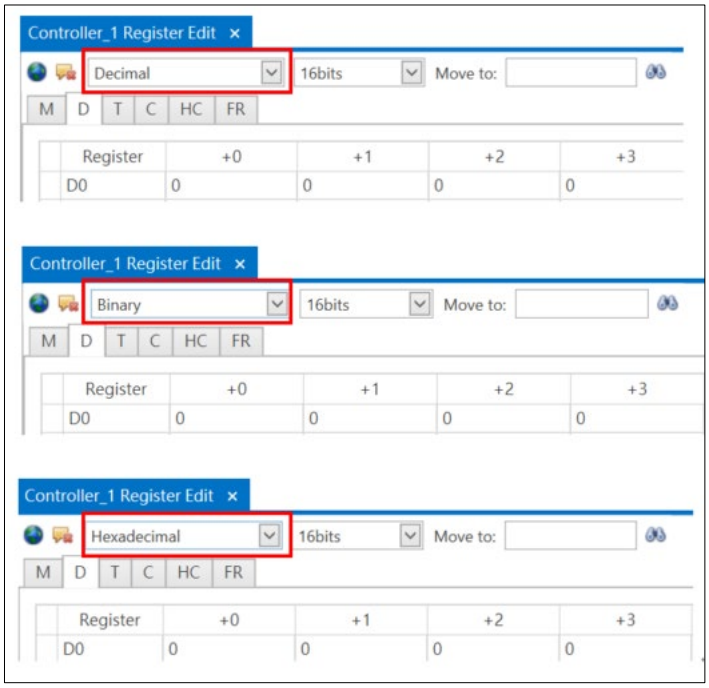


Figure 16 - 16: Scratchpad edit display format switch

16.4.3 Data length

User can select 16bits or 32bits options to display the register value. If 32bit option is selected, then the value in the cell for a register includes the value in the next register.

Assume that the data format selected is Hexadecimal and if the values in registers D0 = AAAA, D1 = BBBB, D2 = CCCC, the **Register Edit** window will display the values as shown in the following figure for 16bits and 32bits options.

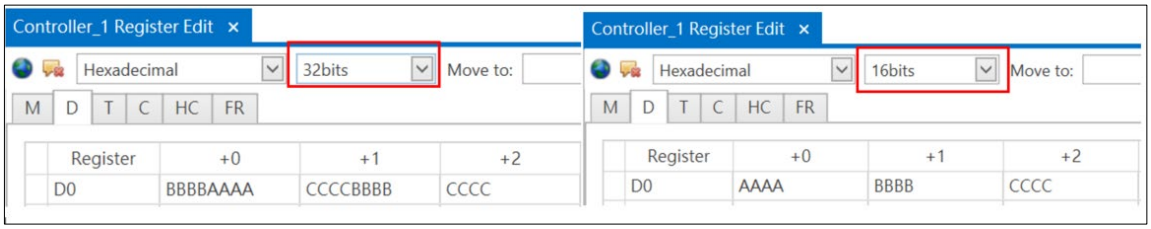


Figure 16 - 17: Register Edit - 16 Bits/32 Bits options

The **Transfer Setup** window is shown in the following figure.

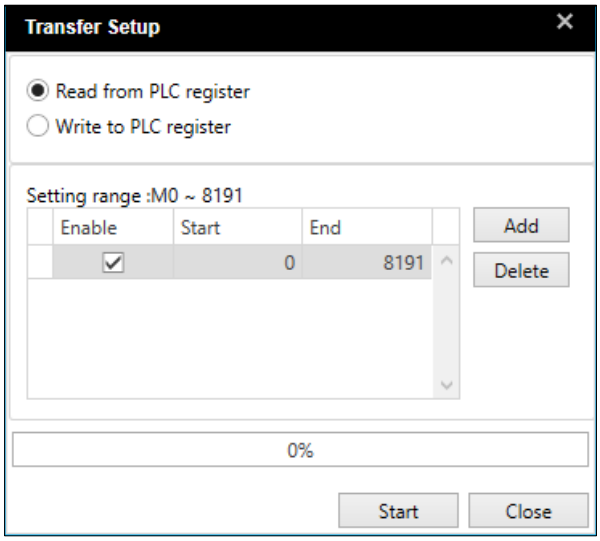


Figure 16 - 18: Transfer Setup

The functions in **Transfer Setup** window described in the following table:

Function	Description
Read from PLC register	Click to select read values from PLC registers ***NOTE**: **Read from PLC register** is selected by default.*
Write to PLC register	Click to select write values to PLC registers
Setting range :	Displays the register range for the selected register type, user can add registers in separate segment. **Example**: M0 ~ M8191 for M, D0 ~ D29999 for D etc.
Enable	Click to check/uncheck the address edit lines ***NOTE**: Address edit lines are checked by default*
Start *(column)*	Enter the start value of address edit ***NOTE**: **0** is the default value*
End	Enter the end value of address edit. ***NOTE**: Default value is the full range of register type selected.*
Add	Click to add address edit range lines.
Delete	Click to delete address edit range lines.

Function	Description
Start *(button)*	Click to start the transfer of register values (read/write).
Close	Click to close **Transfer Setup** window.

16.4.4 Address Edit – Write

Follow these steps to perform write register values to PLC:

1. Click a register type tab

   Result: Register type tab selected above displays.

2. To write register values to the PLC, edit the register values first and click icon

   Result: **Transfer Setup** window displays.

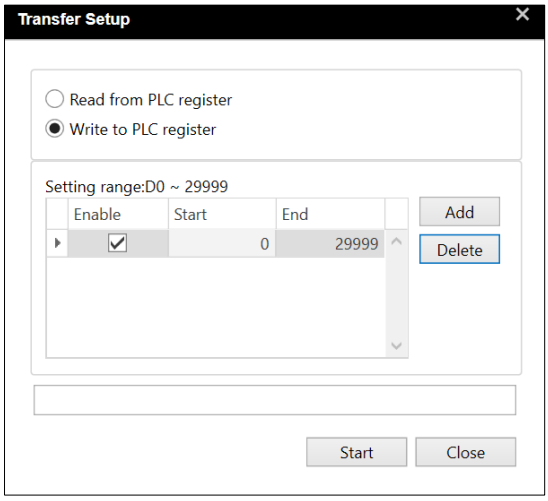


Figure 16 - 19: Transfer Set-Up

Attention:

Before values are written into the PLC registers, user have to make sure that the operation does not affect the system, or cause damage to the system and staff!

3. Click **Write to PLC register.**

4. Click **Add** to add extra address lines if the register range to be written is not continuous.

   Result: Address Edit lines are added.

5. Edit **Start** and **End** values according to user's requirement.

6. Click on **Start** button**.**

   Result: Register values in the PLC is written as per editing done and range selected. The status appears in the **Transfer Setup** window in the progress bar.

7. Click **Close** in the **Transfer Setup** window.

   Result: **Transfer Setup** window closes.

16.4.5 Address Edit – Read

Follow these steps to perform read register values from PLC:

1. Click a register type tab

   Result: Register type tab selected above displays.

2. Click icon.

   Result: **Transfer Setup** window displays (**Read from PLC register** is selected by default)

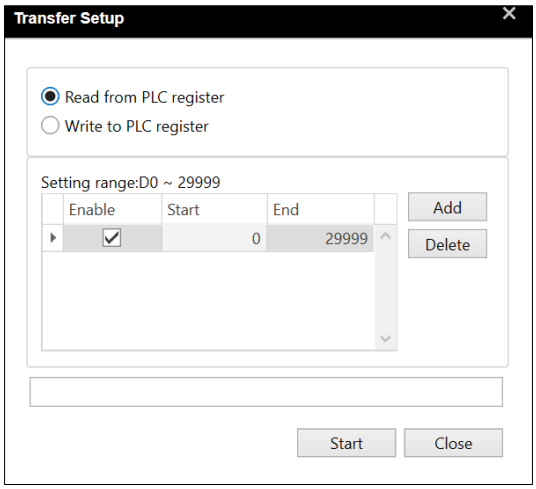


Figure 16 - 20: Transfer complete screen

3. Click **Add** to add extra address lines if the register range to be read is not continuous

   Result: Address Edit lines are added.

4. Edit **Start** and **End** values according to user's requirement.

5. Click on **Start** button.

   Result: Register values in the PLC are read as per editing done and range selected. The status appear in the **Transfer Setup** window in the progress bar.

6. Click **Close** in the **Transfer Setup** window.

   Result: **Transfer Setup** window closes.

16.4.6 Address Edit – Clear all values

Follow these steps to clear the values of all registers in a register tab:

1. Click icon or

   Right-click in the register tab to open its context menu and click **Clear all edited value in this page**.

   Result: Pop-up window displays as shown in the following figure.

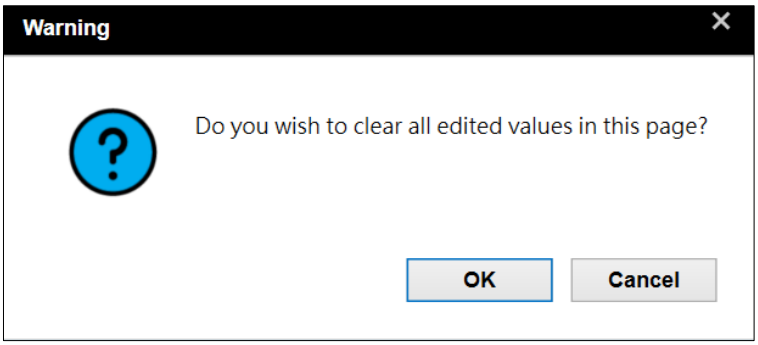


Figure 16 - 21: Clear All Edited value

2. Click on **OK**.

   Result: All register values in the selected tab (register type selected) are reset to 0 and pop-up window **Information** displays as shown in the following figure.

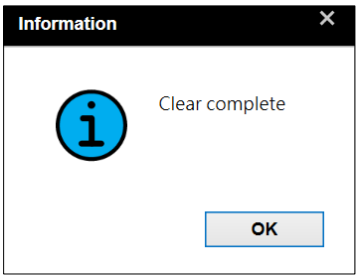


Figure 16 - 22: Clear completed

***NOTE**:*

Only the value in the present edited page (register type) is reset to 0. Values edited in other pages (register types) remains intact.

16.4.7 Address Edit – Move to

User can enter a register address in the **Move to** field and navigate to the respective address.

Follow these steps to move to a register location:

1. Enter a register value in the **Move to** field.
2. Click icon or

   Press **Enter** key on the keyboard.

   Result: System will navigate or the particular location entered in **Move to** field as shown in the following figure.

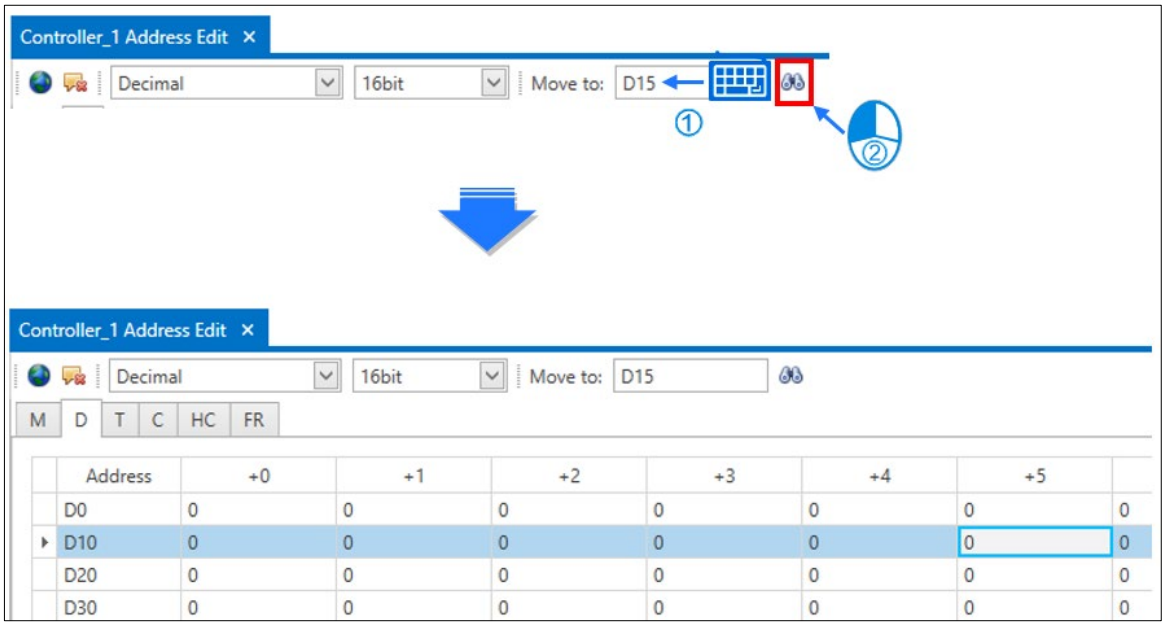


Figure 16 - 23: Address Edit - Move to

16.5 Step Position

When an error occurs during operation, user can use special registers (example SR1, SR2, the 32-bit registers for storing wrong addresses) to diagnose which step went wrong and then navigate to the specific program to check and fix it. Step Position helps user to understand which step number has gone wrong. User can locate Step Position in **Controller** > **Auxiliary** > **Step Position** in DIADesigner Project Explorer.

Follow these steps to perform Step Position:

1. Navigate to **Controller** > **Auxiliary** > **Step Position.**
2. Double-click **Step Position**.

   Result: The **Step Positioning** window displays as shown in the following figure.

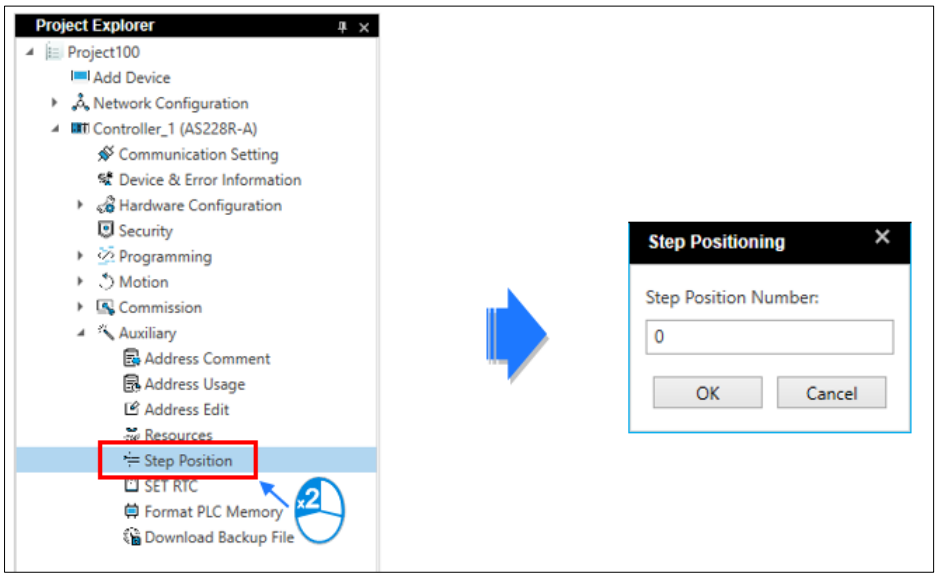


Figure 16 - 24: Step Position

3. Enter the step number in **Step Position Number** field
4. Click **OK.**

   Result: System will direct user to the corresponding POU and the position of the step in the program.

   ***NOTE**: Be sure to program the project before using **Step Position** function.*

16.6 Set RTC

User can set the real time clock in the PLC from DIADesigner with the help of Set RTC. Make sure that DIADesigner is connected to the CPU module before performing this action.

User can locate the Set RTC in **Controller** > **Auxiliary** > **Set RTC** in the **Project Explorer** as shown in the following figure.

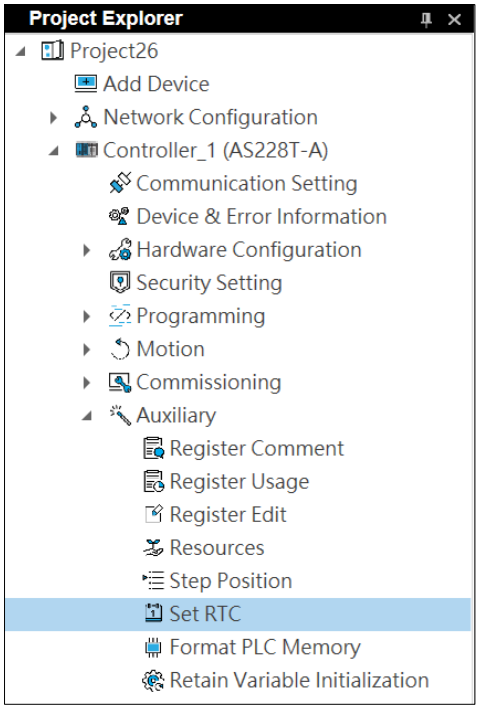


Figure 16 - 25: Set RTC in Project Explorer

Double-click on **Set RTC** to open the Set RTC window as shown in the following figure.

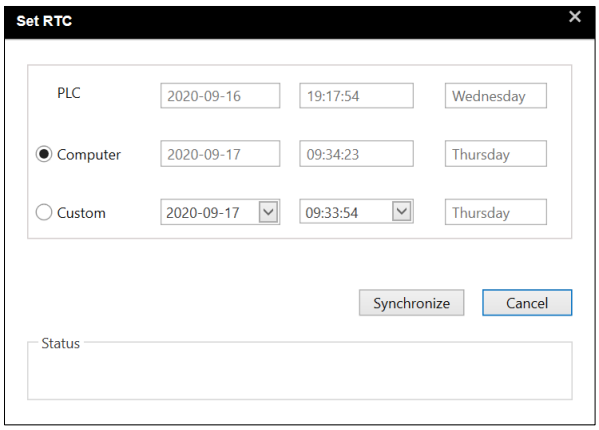


Figure 16 - 26: Set RTC window

16.6.1 Updating Computer time to PLC time

The Set RTC window displays the PLC date, time and day information retrieved from the PLC when the window is opened.

If user want to change the real-time clock in the PLC with the Computer's time, they can select Computer option and click **Synchronize**.

Result: Computer date, time and day information are written to the PLC RTC. DIADesigner updates the status of action – **RTC Setting Complete** as shown in the following figure.

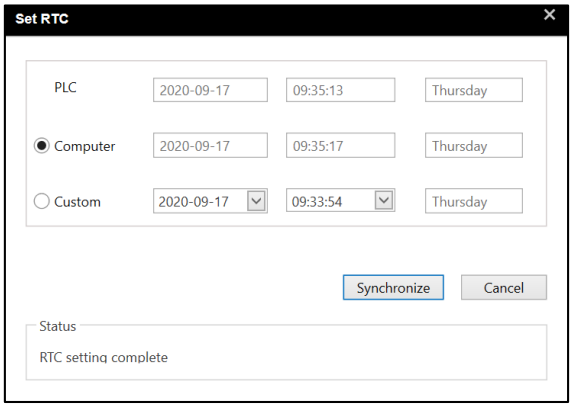


Figure 16 - 27: RTC Setting complete - Computer

16.6.2 Updating Custom time to PLC time:

If user want to set the real-time clock information in the PLC with a custom time, they can select the **Custom** radio button in the **Set RTC** window and enter/select the date, time and day in the respective fields and click **Synchronize**.

Result: DIADesigner updates the status of action – **RTC Setting Completed** as shown in the following figure.

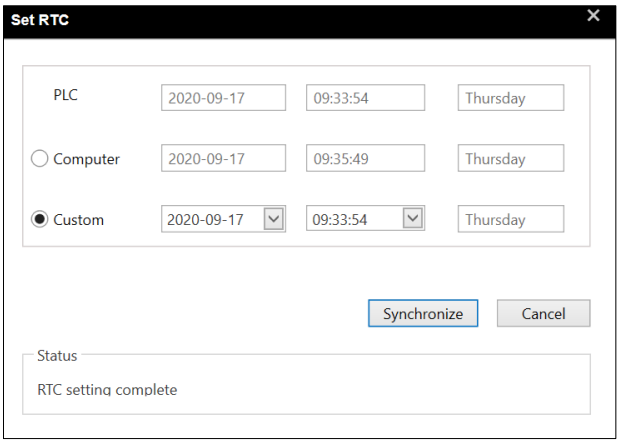


Figure 16 - 28: RTC Setting complete - Custom

16.7 Format PLC Program Memory

User can clear the memory, or restore the setting in the PLC to the factory settings from DIADesigner with the help of Format PLC Memory. Make sure that DIADesigner is connected to the CPU module before performing this action. User can locate the Format PLC Memory function in **Controller** > **Auxiliary** > **Format PLC Memory** in the **Project Explorer** as shown in the following figure.

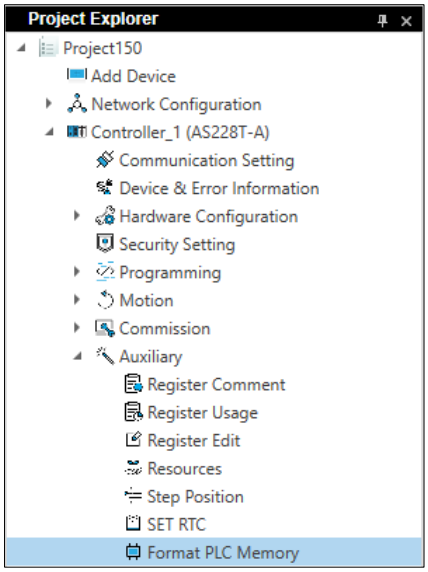


Figure 16 - 29: Format PLC Memory in Project Explorer

Double-click **Format PLC Memory** to open the **Format PLC Memory** window as shown in the following figure.

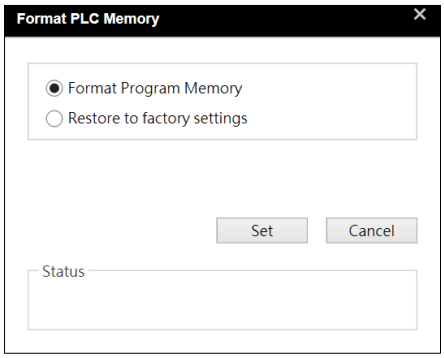


Figure 16 - 30: Format PLC Memory

The functions in **Format PLC Memory** window is described in the following table:

Function	Description
Format Program Memory	Click to format PLC program memory
Restore to factory settings	Click to reset PLC to factory settings
Set	Click to perform the action selected
Cancel	Click to cancel the action

16.7.1 Format Program Memory

To format program memory, user can select Format Program Memory radio button and click Set.

Result: PLC program memory is formatted. DIADesigner updates the status of action – **Finish clearing PLC program!** as shown in the following figure.

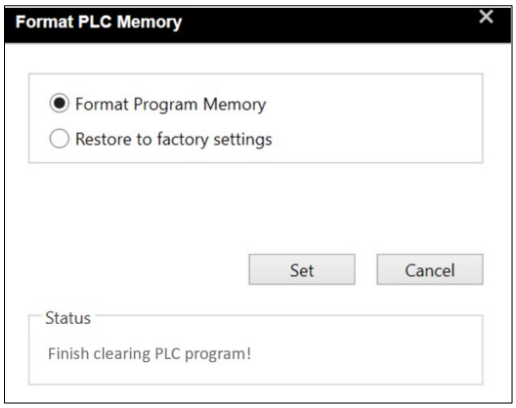


Figure 16 - 31: Format Program Memory - Finish clearing PLC program

***NOTE**: User can format program memory when the controller is in **Offline** and **Stop** mode. In **Online** mode, **Format PLC memory** function is disabled. When the PLC is in **Run** status, a message – **This function cannot be performed in PLC RUN mode** displays as shown in the following figure.*

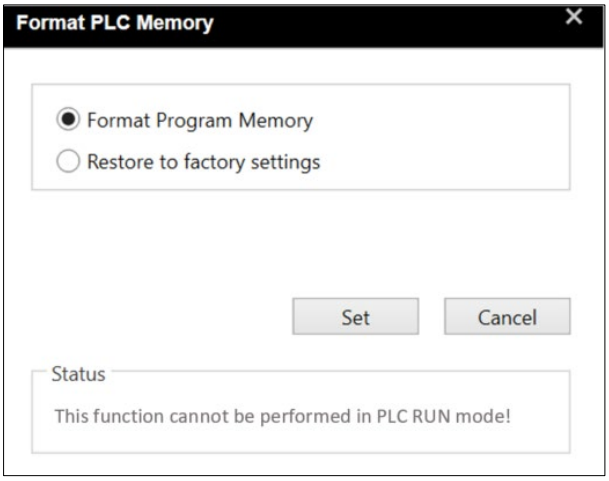


Figure 16 - 32: Format PLC Memory – PLC RUN mode message

16.7.2 Reset PLC Memory (Factory Setting)

To reset the PLC memory to factory settings, user can select **Restore to factory settings** radio button in **Format PLC Memory** window and click **Set**.

Result: DIADesigner updates the status of action – **Initial setup complete, Re-power PLC!** as shown in the following figure.

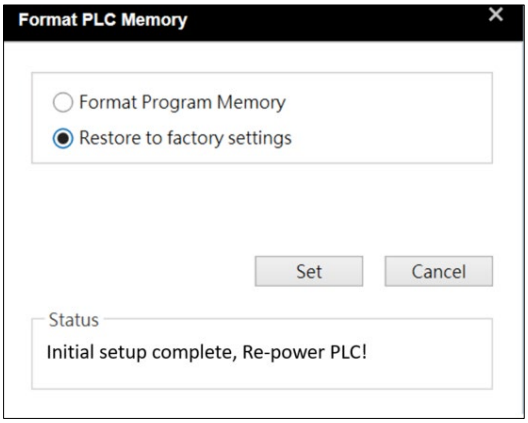


Figure 16 - 33: Format PLC Memory - Initial setup completed, Re-power PLC

***NOTE**: Upon performing a **Reset PLC Memory (Factory Setting)** option, user must restart the PLC power supply to proceed with the operations. User can perform **Reset PLC Memory (Factory Setting)** when the controller is in **Offline** and **Stop** mode. In **Online** mode, **Format PLC memory** function is disabled. When the PLC is in **Run** status, a message – **This function cannot be performed in PLC RUN mode!** displays as shown in the following figure.*

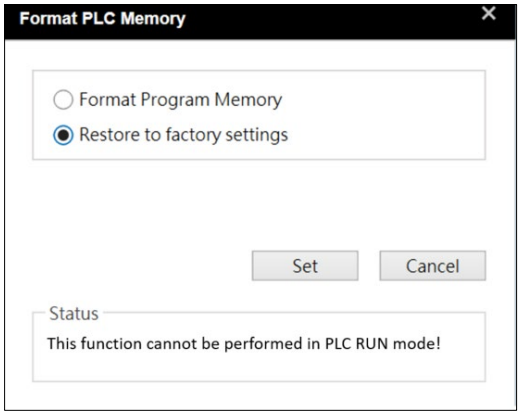


Figure 16 - 34: Format PLC Memory - Online Run message

16.8 Parameter Backup & Restore

***NOTE**: Parameter Back-up & Restore is supported only in the VFD C2000 series now. For other series, it may be supported in future version.*

User can Back-up the parameters in inverters, and restore it afterward.

Navigate **Inverter device > VFD function > Parameter Backup & Restore** in Project Explorer to open the tab. The functions in **Parameter Backup & Restore** is described in the following table:

Function	Description
Device Name	Device name of target device
Backup file	Select a file to export or overwrite
Backup	Button to execute Backup
Export	Export file to a specified directory
Restore file	Select a file to restore to this device
Restore	Button to execute Restore
Import	Import file from a specified directory (*.par file)
Device Password	Enter password which is matching with parameter 00-08 to proceed transformation, keep blank if the connected device has no password
Compatibility Check	Select between **Partially Compatibility**, **Exactly Match** and **Forced Transformation** before proceed restore
Compatibility Setting	Available when **Partially Compatibility** is selected in Compatibility Check
Copy Setting To Other Device	Copy current setting and password to other inverters

Follow these steps to perform Parameter Back-up & Restore:

1. Set up communication setting and connecting. Parameter Back-up & Restore only available when the device is connected directly of via bypass route.

2. Double click Parameter Back-up & Restore to open the tab as shown in the following figure.

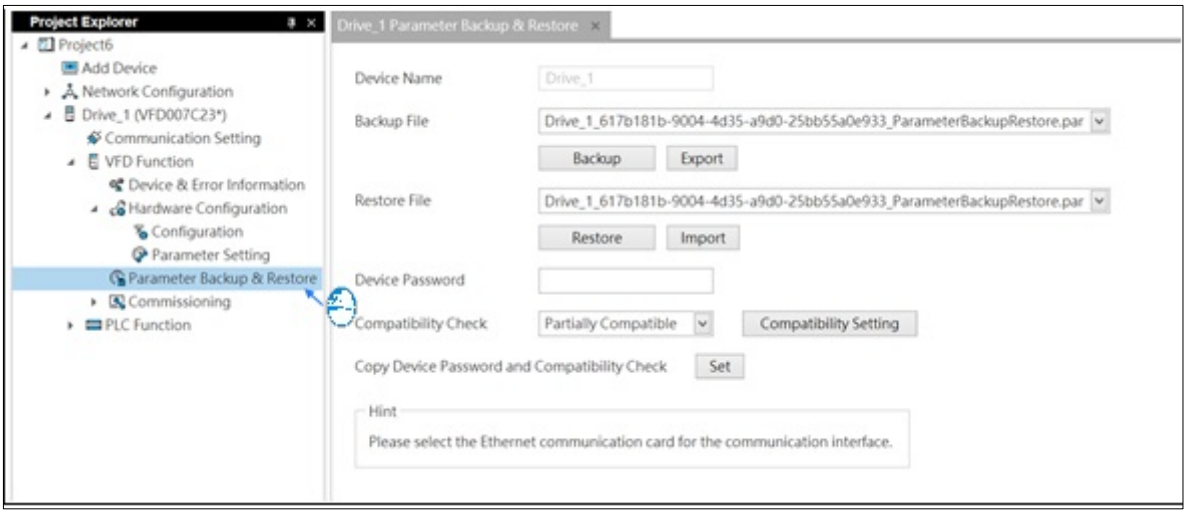


Figure 16 - 35: Parameter Back-up and Restore

3. Back-up: Click Backup to upload all parameters from current connected inverter and save it in a file in the Backup path directory.
4. Restore: Click Restore to overwrite all parameters from the file in Restore Path directory to the connected inverter

In case of different application situation, DIADesigner allow user restore the file to compatible devices, like restore the parameters into different series or version. User can select **Partially Compatible** in Compatibility Check, the restore allows according to the setting of **Compatibility Setting** dialog as shown in the following figure.

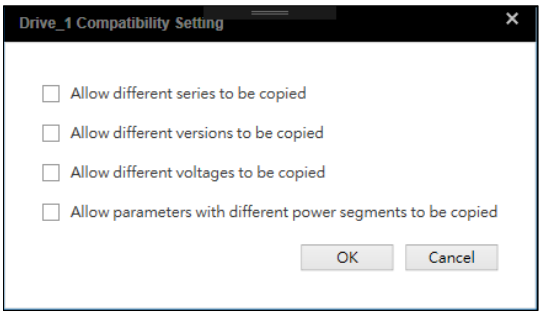


Figure 16 - 36: Drive Compatibility Setting

If user wants to copy current setting to other inverters, user can click Copy Setting to Other Devices to copy Compatibility Setting and password to the selected inverters in project.

16.9 Retain Variable Initialization

User can reset the retain variable to initialize value of a PLC.

User can locate the Retain Variable Initialization in **Controller** > **Auxiliary** > **Retain Variable Initialization** in the **Project Explorer** as shown in the following figure.

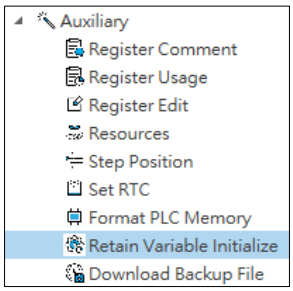


Figure 16 - 37: Retain Variable Initialization - Project Explorer

Click **OK** in the confirmation dialog, the retain variable will initialize.

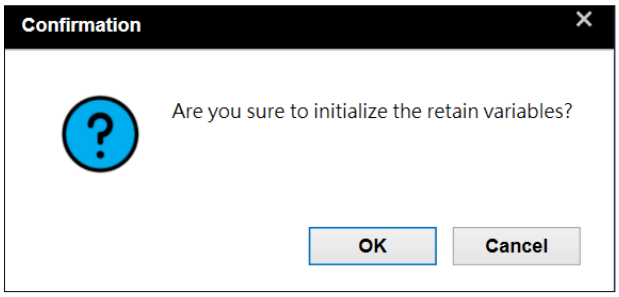


Figure 16 - 38: Confirmation dialog to initialize the retentive Variables

***NOTE**:* Make sure the retain variables are downloaded and the PLC is in STOP mode before execute the function.

Chapter 17: Other Function

17.1 Find and Replace

User can find text objects in DIADesigner with the help of **Find** feature and replace the texts with the help of **Replace** feature. There is also a batch replacement register, which can be used as a batch replacement register function.

The scope of implementation of batch replacement register function is as follows:

- Program POU (including LD, ST, SFC language) and regional variable table
- Function block POU (including LD, ST language) and regional variable table
- Global Variable Table

The **Find** and **Replace** function can be used in following contexts:

- POU – LD, ST, SFC,C Programs and Local Variable Table
- POU – LD, ST,C Function Blocks and FB Variable Table
- Global Variable Table
- DUT – Structure and Enumeration and Union.

17.1.1 Find – POU

User can open any POU and click the POU working area or the local variable table to start the Find operation.

Follow these steps to perform a find operation in DIADesigner POU:

1. Open a POU.
2. Click **Edit** > **Find and Replace** > **Find** or

   Press **Ctrl** + **F** keys simultaneously on the keyboard.

   Result: **Find and Replace** window displays with **Find** tab opened by default as shown in the following figure.

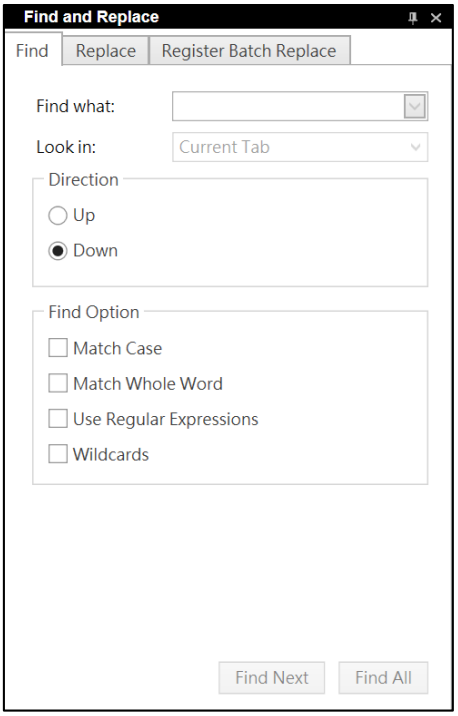


Figure 17 - 1: Find and Replace - Find

The functions available in the **Find** tab in the **Find and Replace** window are described in the following table:

Toolbar	Description
Find what	User can enter the object name which is searched for, or select an object which has been searched.
Look in	Select the area to look into. Options available are: - • Current Tab ***NOTE**: Current Tab is the default value.*
Match Case	Click to select the matching case search.
Match whole word	Click to select the whole word case search.
Use Regular Expressions	Click to use the regular expressions. (Please refer to related technical documents for more information.)
Wildcards	Click to use the wildcard expressions. (Please refer to related technical documents for more information.)
Up	Click to select among previous instances.
Down	Click to select among instances after the present instance.

Toolbar	Description
Find Next	Click to find the next instance.
Find All	Click to find all the instances.

3. Enter the text to be searched in the **Find What:** field, select the options and directions as per user requirements.
4. Click **Find Next** or **Find All.**

   Result: The **Search Results** window displays all the occurrences of the text in search.

17.1.2 Replace – POU

Follow these steps to perform a replace operation in DIADesigner POU:

1. Open a POU.
2. Click **Edit** > **Find and Replace** > **Replace** or

   Press **Ctrl** + **H** keys simultaneously on the keyboard.

   Result: The **Find and Replace** window displays with **Replace** tab as shown in the following figure.

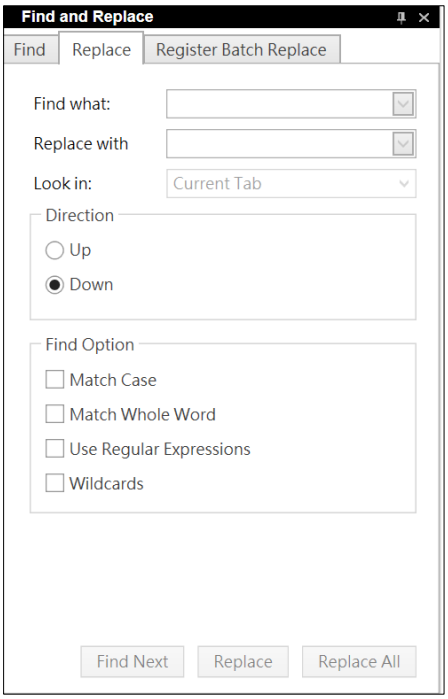


Figure 17 - 2: Find and Replace - Replace

The functions available in **Replace** tab in the **Find and Replace** window are described in the following table:

Toolbar	Description
Find what	User can enter the object name which is searched for, or select an object which has been searched.
Replaced with	User can enter the object name which is replacing the searched text in this operation.
Look in	Select the area to look into. Options available are: - • Current Tab ***NOTE**: **Current Tab** is the default value.*
Match Case	Click to select the matching case search.
Match whole word	Click to select the whole word case search.
Use Regular Expressions	Click to use the regular expressions. (Please refer to related technical documents for more information.)
Wildcards	Click to use the wildcard expressions. (Please refer to related technical documents for more information.)
Up	Click to select among previous instances.

Toolbar	Description
Down	Click to select among instances after the present instance.
Find Next	Click to find the next instance.
Replace	Click to replace the current instance.
Replace All	Click to replace all the instances.

Follow the steps below to perform the replacement operation on the Find and Replace page:

1. Enter the text to be searched in the **Find What:** field, select the options and directions as per user requirements.
2. Click **Replace** or **Replace All**.

   Result: Search result(s) is replaced by the button clicked in the **Replace** tab.

17.1.3 Batch Replacement Register

The user can open the POU and click on the editing workspace or the regional variable table, and then perform the batch replacement operation.

Follow the steps below to perform batch replacement register operation in the POU of DIADesigner:

1. Open POU
2. Click the Find and Replace window, the Batch Replace Register tab.
   Result: The Find and Replace window will switch to the Batch Replace Register tab, as shown in the figure below.

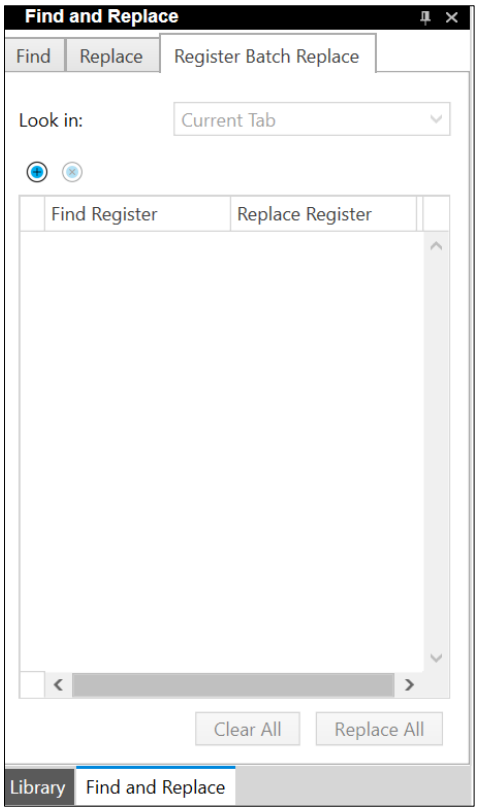


Figure 17 - 3: Batch replace register tab

The functions of the batch replacement register tab are as follows:

Toolbar	Function description
Inquire	The supported replacement ranges are as follows: • Current tab **NOTE: The current tab is the default value.**
	Click this item to add settings "Search Register" and "Replace Register".
	Click this item to delete the set "Search Register" and "Replace Register".
Find register	Enter the search register.
Replace register	Enter the register to replace.
Consecutive number	Enter the number of registers to replace.
Clear All	Clear all "Find Register" and "Replace Register" items.

Toolbar	Function description
Replace All	Replace the "Find Register" content with the "Replace Register" content. (The replacement quantity varies according to the setting of "Continuous Quantity")

Follow the steps below to perform the replacement operation on the Batch Replace Register page:

1. Click **ADD**.

Figure 17 - 4: Add

2. Enter the Search Register to **Find Register**.

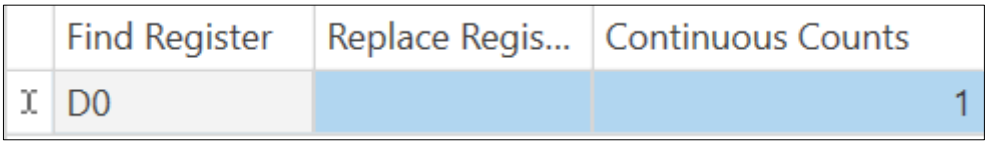


Figure 17 - 5: Find register

3. Enter the Replacement Register to the **Replace Register**.

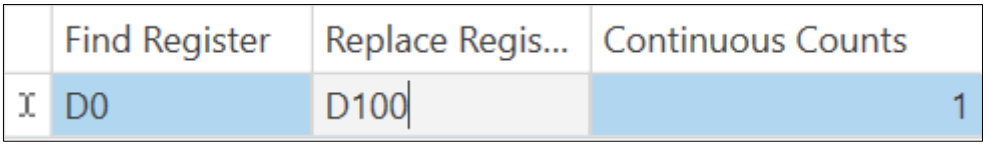


Figure 17 - 6: Replace register

4. Set the Continuous Counts, set to 2.

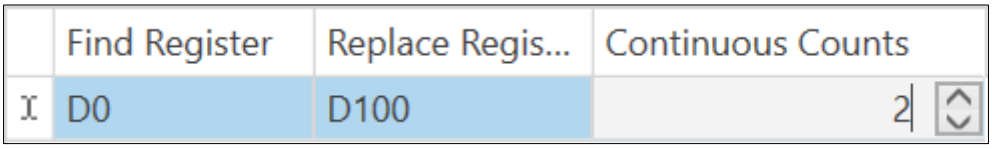


Figure 17 - 7: Set continuous quantity

5. Click **Replace All**

Figure 17 - 8: Replace all

6. Click **OK** to complete the replacement.

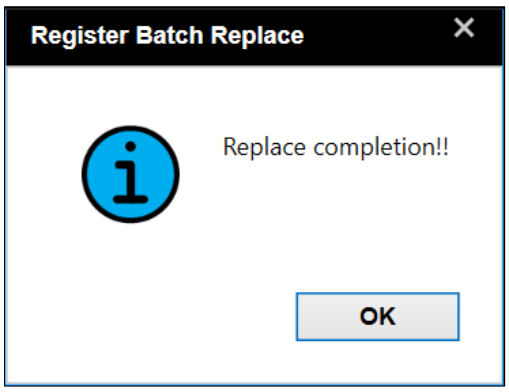


Figure 17 - 9: Replace Success

Result: If a batch is executed to replace the registers, all searchable registers will be replaced. (Refer to Figure XX-X: Batch replace register)

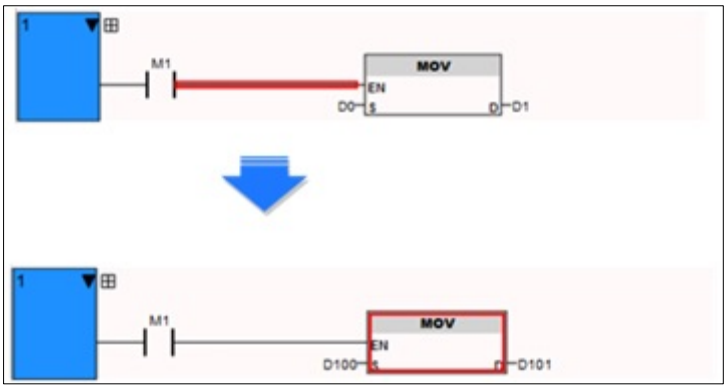


Figure 17 - 10:Batch replacement register

***NOTE**: The upper limit value entered in the "Continuous Quantity" field is 999.*

17.2 Bookmark

Bookmark is used in a POU to mark the location(s) – a line in a Structured Text or a Network in Ladder Diagram POU. Bookmark function is similar to a bookmark which a reader inserts in a book to remember the page the reader has to open or the pages where some important read out are located.

User can perform the following actions with Bookmarks:

- Create Bookmark
- Go to Next Bookmark
- Go to Previous Bookmark
- Delete Bookmark
- Clear All Bookmarks

All the Bookmarks created display in the **Bookmark** window. POU in which Bookmarks are created are shown in a tree structure and all the Network names with Bookmarks are shown in the branch level. User can open Bookmark window from **View** > **Bookmark** from menu bar.

17.2.1 Bookmark in Ladder Diagram (LD) POU

In Ladder Diagram POU, Bookmark is provided for Networks. Bookmark functions are supported in the context menu > **Bookmark > Toggle Bookmark** as well as from the menu bar **Edit** > **Bookmark**.

An example of Bookmark in an LD Network is shown in the following figure.

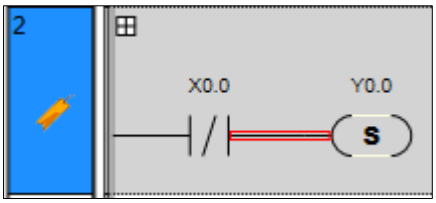


Figure 17 - 11: Bookmark (LD)

The context menu Bookmark options in a LD POU is shown in the following figure.

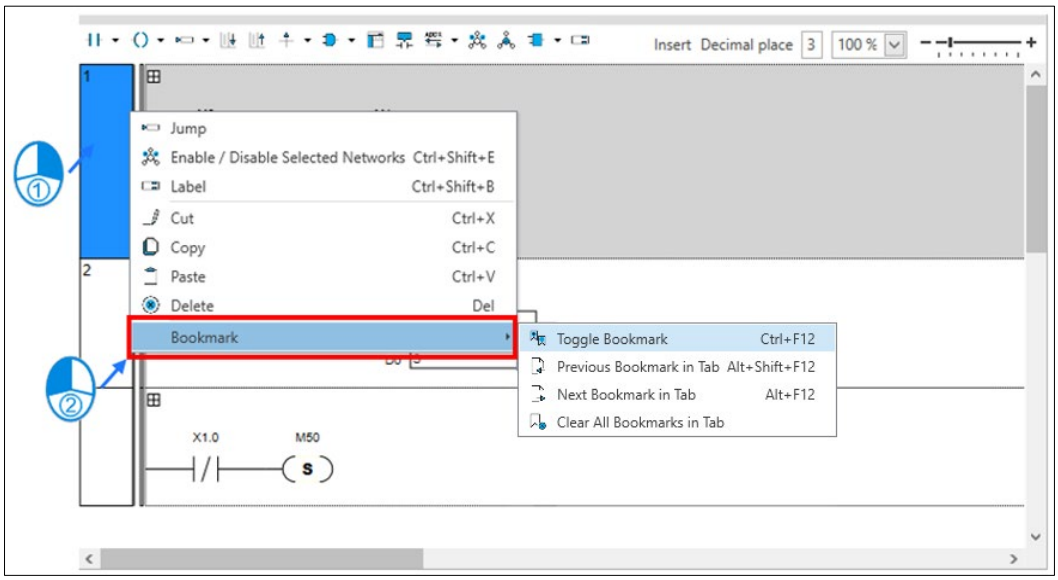


Figure 17 - 12: Bookmark in context menu (LD)

The Menu Bar **Edit** > **Bookmark** options are shown in the following figure.

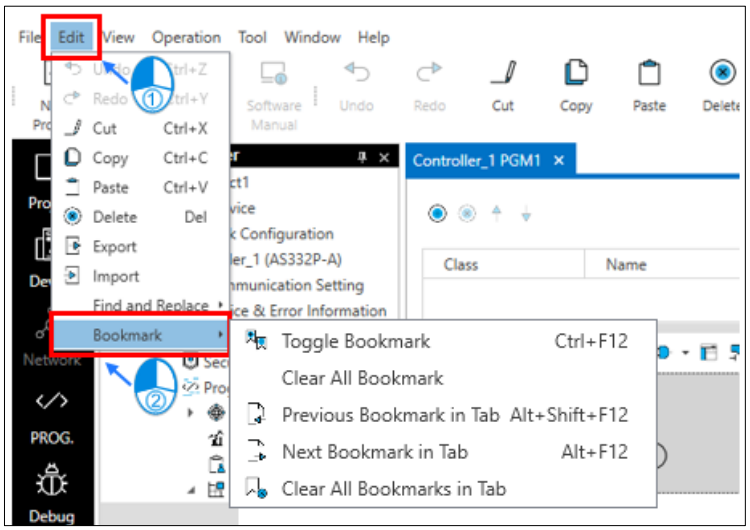


Figure 17 - 13: Edit menu Bookmark (LD)

17.2.1.1 Create Bookmark – LD POU

Follow these steps to create Bookmark in a LD POU:

1. Select the Network(s) where Bookmark has to be created.
2. Right-click to open the context menu and click **Bookmark** > **Toggle Bookmark**

   or

   Select **Edit** > **Bookmark** > **Toggle Bookmark** in the menu bar or

   Press **Ctrl** + **F12** keys simultaneously in the keyboard.

   Result: Bookmark is created in the selected Network(s)

   Example:

   Networks - Network1 and Network5 with Bookmark is shown in the following figure.

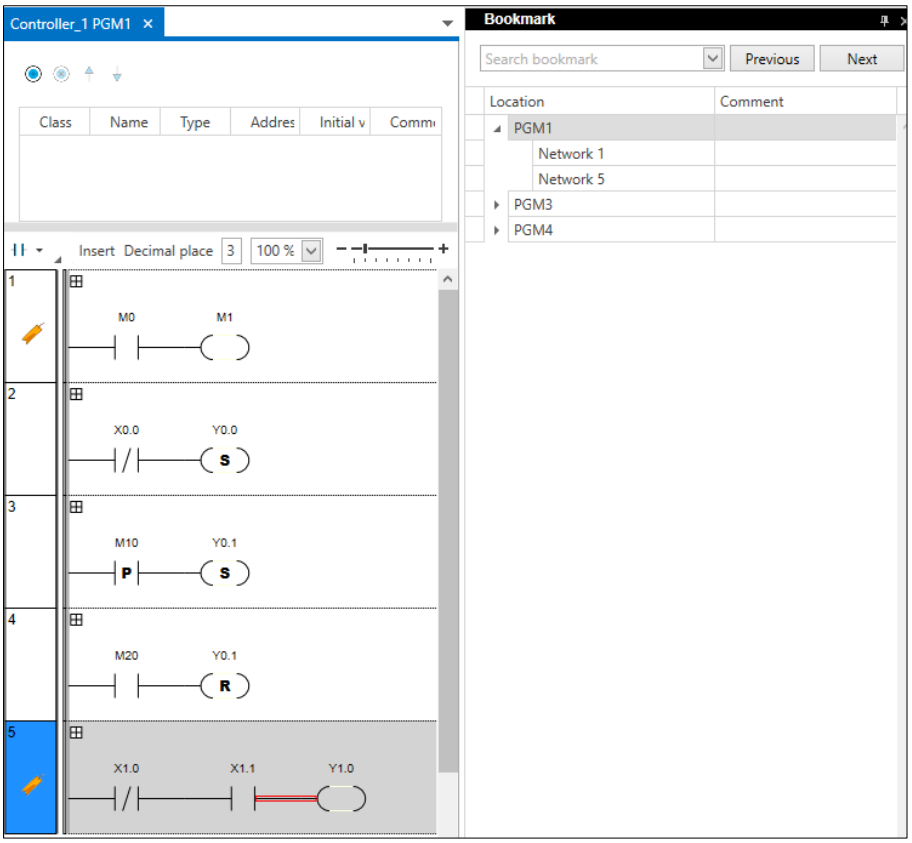


Figure 17 - 14: Bookmark – LD Networks

17.2.1.2 Next Bookmark in Tab – LD POU

Follow these steps to navigate to the next Bookmark in tab in a LD POU:

1. Select the Network with Bookmark.
2. Right-click to open the context menu and click **Bookmark** > **Next Bookmark in Tab** or

   Select **Edit** > **Bookmark** > **Next Bookmark in Tab** in the menu bar or

   Press **Alt** + **F12** keys simultaneously in the keyboard or

   Click **Next** button in the **Bookmark** window.

   Result: Network with the next immediate Bookmark displays.

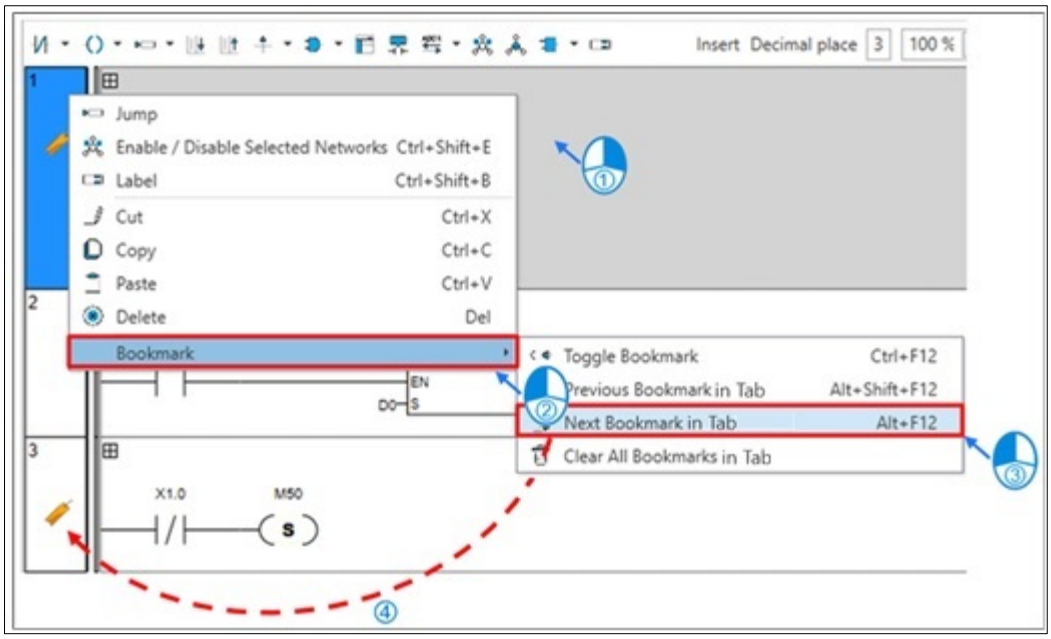


Figure 17 - 15: Next Bookmark in Tab (LD)

17.2.1.3 Previous Bookmark in Tab – LD POU

Follow these steps to navigate to the previous Bookmark in tab in an LD POU:

1. Select the Network with Bookmark.
2. Right-click to open the context menu and click **Bookmark** > **Previous Bookmark in Tab** or

   Select **Edit** > **Bookmark** > **Previous Bookmark in Tab** in the menu bar or

   Press **Alt**, **Shift** + **F12** keys simultaneously in the keyboard or

   Click **Previous** button in the **Bookmark** window.

   Result: Network with previous immediate Bookmark displays.

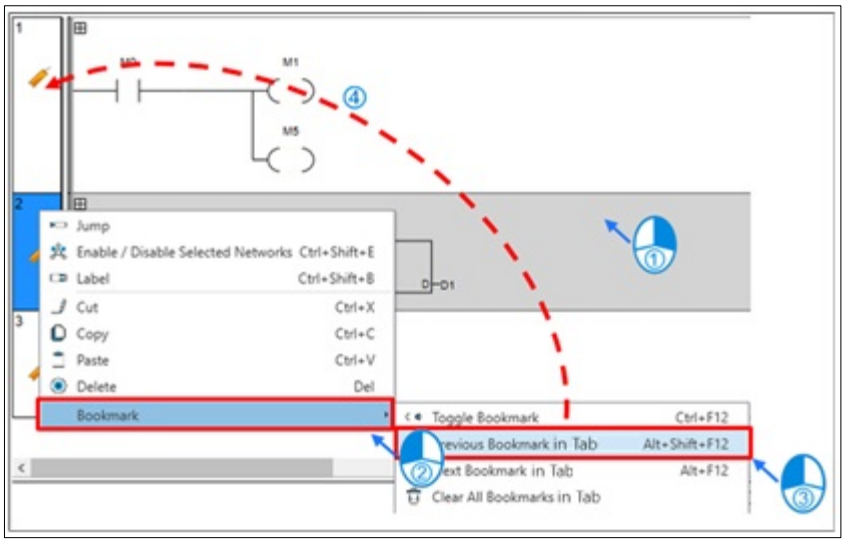


Figure 17 - 16: Previous Bookmark in Tab (LD)

17.2.1.4 Delete Bookmark – LD POU

Follow these steps to delete the Bookmark(s) in a LD POU:

1. Select the Network(s) where Bookmark is/are created
2. Right-click to open the context menu and click **Bookmark** > **Toggle Bookmark**

   or

   Select **Edit** > **Bookmark** > **Toggle Bookmark** in the menu bar or

   Press **Ctrl** + **F12** keys simultaneously in the keyboard.

   Result: Selected Bookmark is deleted.

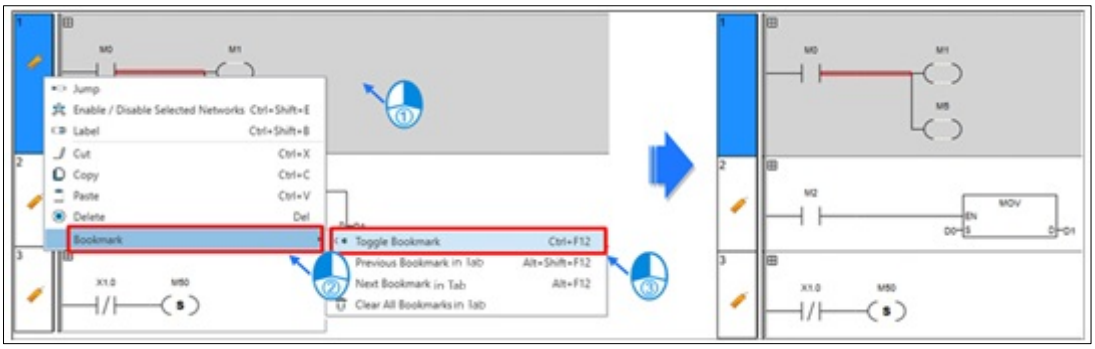


Figure 17 - 17: Clear All Bookmarks in Tab (LD)

17.2.1.5 Clear All Bookmarks in Tab – LD POU

Follow these steps to delete all the Bookmarks in a LD POU:

1. Right-click to open the context menu and click **Bookmark** > **Clear All Bookmarks in Tab** or

   Select **Edit** > **Bookmark** > **Clear All Bookmarks in Tab** in the menu bar or

   Result: All Bookmarks are deleted.

Figure 17 - 18: Clear All Bookmarks in Tab (LD)

***NOTE**:*

*User can click the Bookmark location in the Bookmark tab to navigate to the Bookmark as shown in the following figure. Double-click on the Network name below **Location** in the **Bookmark** window to navigate to the respective Bookmark.*

Figure 17 - 19: Navigate to Bookmark from tab

17.2.2 Bookmark in Structured Text (ST) POU

In Structured Text POU, Bookmark is provided for text line(s). Bookmark functions are supported in the **context menu > Bookmark of ST POU Network** as well as from the Menu Bar **Edit** > **Bookmark**.

An example of Bookmark in an ST line is shown in the following figure.

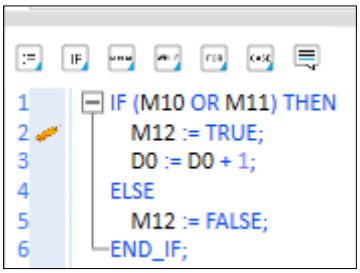


Figure 17 - 20: Bookmark (ST)

The context menu **Bookmark** options of a ST POU is shown in the following figure.

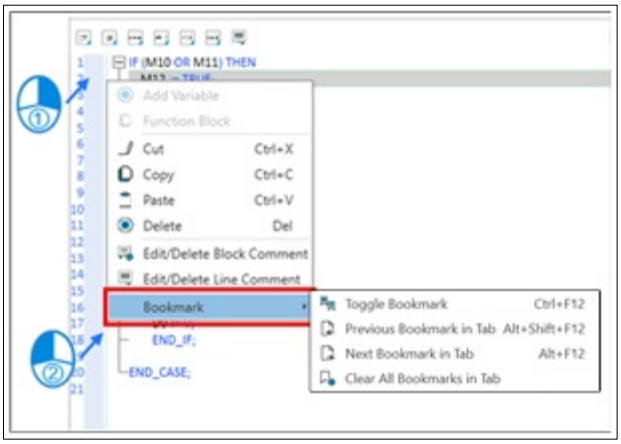


Figure 17 - 21: Bookmark in context menu (ST)

Menu Bar **Edit** > **Bookmark** options are shown in the following figure.

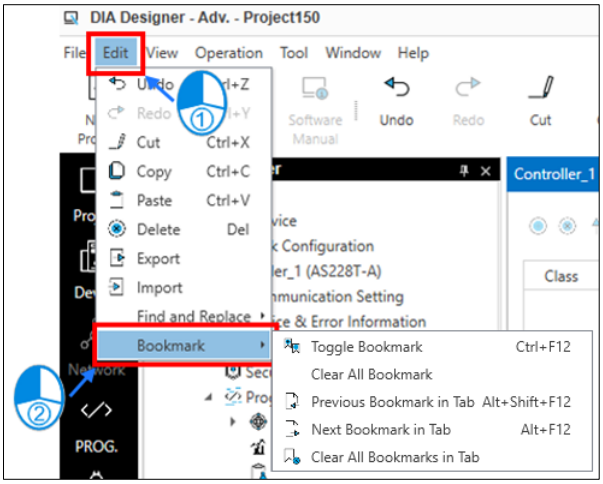


Figure 17 - 22: Edit menu Bookmark (ST)

17.2.2.1 Create Bookmark – ST POU

Follow these steps to create Bookmark(s) in a ST POU:

1. Select the line where Bookmark has to be created.
2. Right-click to open the context menu and click **Bookmark** > **Toggle**

Bookmark

or

Select **Edit** > **Bookmark** > **Toggle Bookmark** in the menu bar or

Press **Ctrl** + **F12** keys simultaneously in the keyboard or

Click on the space between the *Line number* and *editing space*.

Result: Bookmark is created for the line selected.

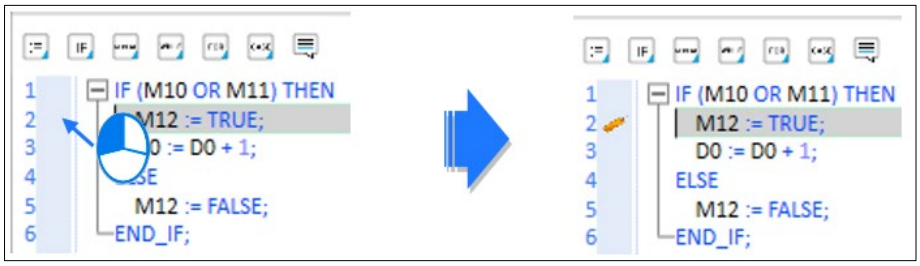


Figure 17 - 23: Bookmark (ST) - click

Example:

Lines - Line2 and Line10 with Bookmark is shown in the following figure.

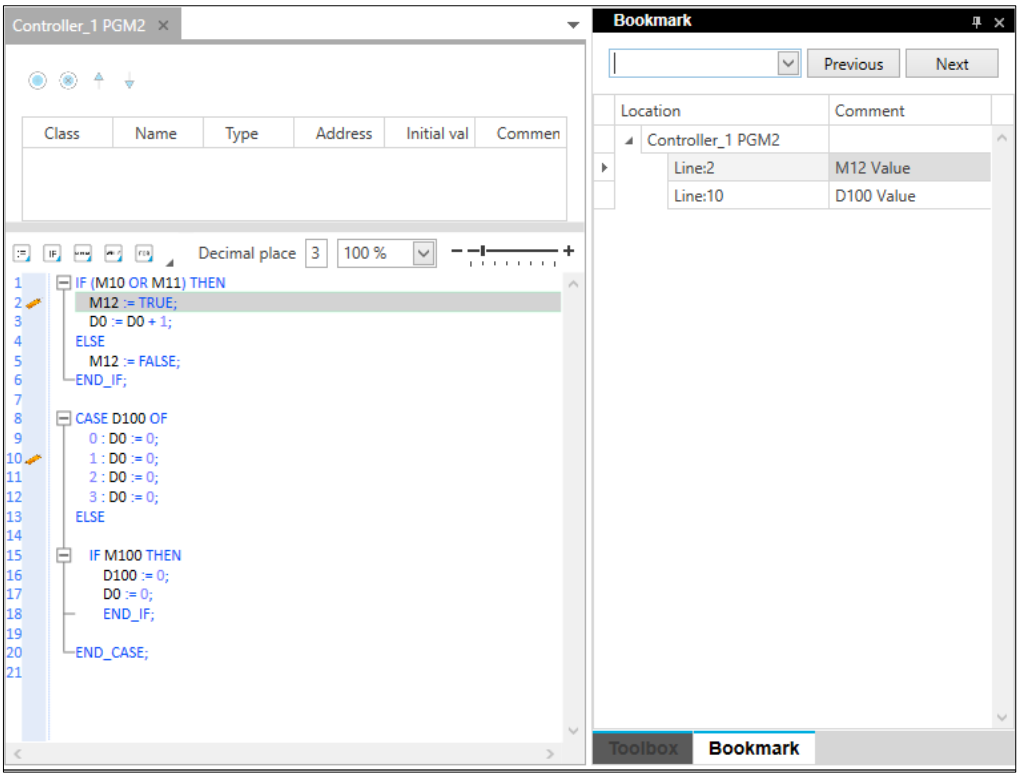


Figure 17 - 24: Bookmark (ST lines)

17.2.2.2 Next Bookmark in Tab – ST POU

Follow these steps to navigate to the next Bookmark in tab in a ST POU:

1. Select the line with Bookmark.
2. Right-click to open the context menu and click **Bookmark** > **Next Bookmark in Tab** or

   Select **Edit** > **Bookmark** > **Next Bookmark in Tab** in the menu bar or

   Press **Alt** + **F12** keys simultaneously in the keyboard or

   Click **Next** button in the **Bookmark** window.

   Result: Line with the next immediate Bookmark displays.

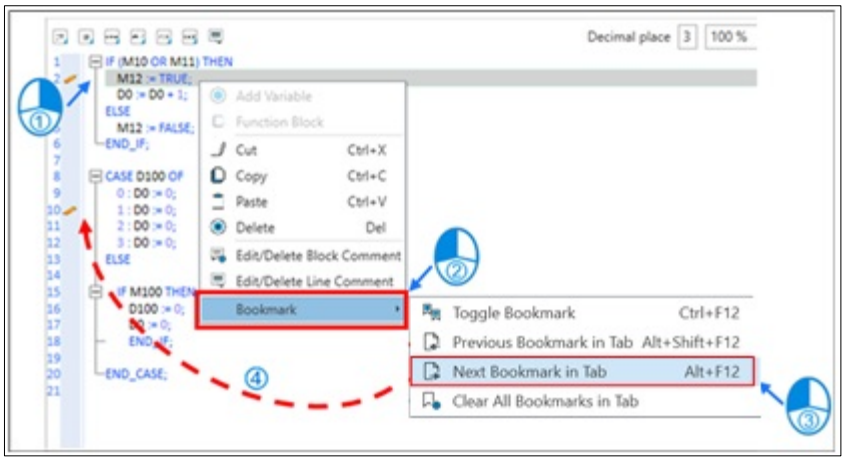


Figure 17 - 25: Next Bookmark in Tab (ST)

17.2.2.3 Previous Bookmark in Tab – ST POU

Follow these steps to navigate to the previous Bookmark in tab in a ST POU:

1. Select the line with Bookmark.
2. Right-click to open the context menu and click **Bookmark** > **Previous Bookmark in Tab** or

   Select **Edit** > **Bookmark** > **Previous Bookmark in Tab** in the menu bar or

   Press **Alt**, **Shift** + **F12** keys simultaneously in the keyboard or

   Click **Previous** button in the **Bookmark** window

   Result: Line with previous immediate Bookmark displays.

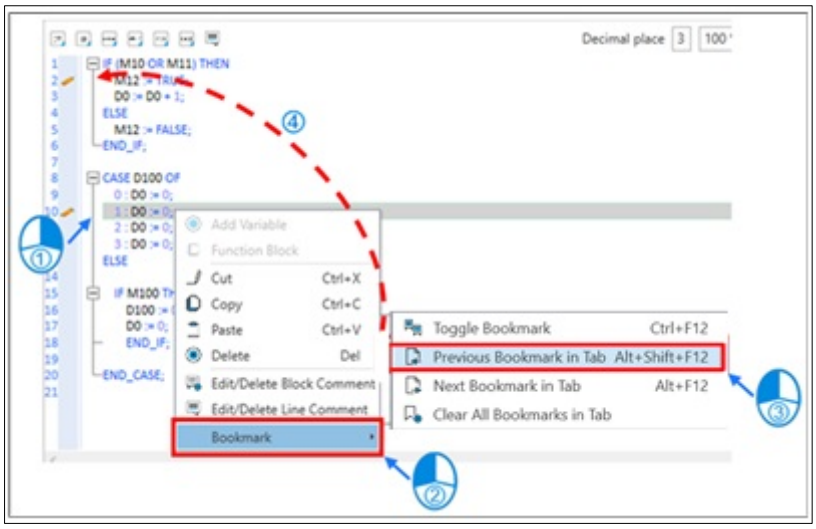


Figure 17 - 26: Previous Bookmark in Tab

17.2.2.4 Delete Bookmark – ST POU

Follow these steps to delete the Bookmark in a ST POU:

1. Select the line where Bookmark is created.
2. Right-click to open the context menu and click **Bookmark** > **Toggle Bookmark**

   or

   Select **Edit** > **Bookmark** > **Toggle Bookmark** in the menu bar or

   Press **Ctrl** + **F12** keys simultaneously in the keyboard.

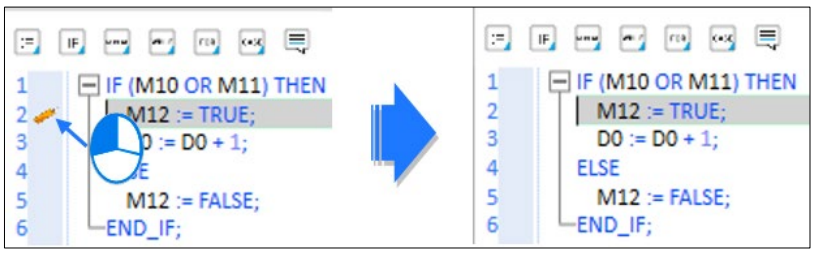


Figure 17 - 27: Delete Bookmark (ST)

Result: Bookmark for the line is deleted.

17.2.2.5 Clear All Bookmarks in Tab – ST POU

Follow these steps to delete all the Bookmarks in a ST POU:

1. Right-click to open the context menu and click **Bookmark** > **Clear All Bookmarks in Tab** or

   Select **Edit** > **Bookmark** > **Clear All Bookmarks in Tab** in the menu bar or

   Result: All Bookmarks are deleted.

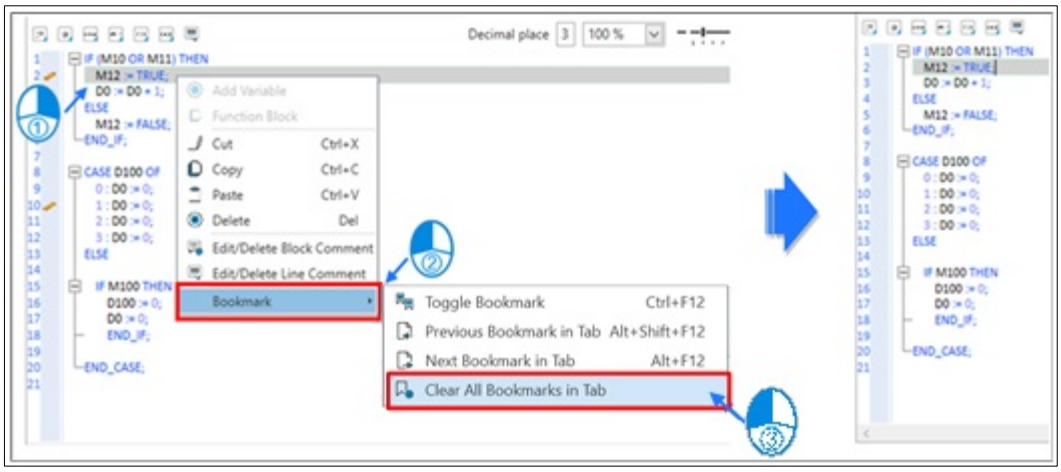


Figure 17 - 28: Delete All Bookmark (ST)

***NOTE**:*

*User can click the Bookmark location in the **Bookmark** tab to navigate to the Bookmark as shown in the following figure. Double-click on the **Line** name below **Location** in the **Bookmark** window to navigate to the respective Bookmark.*

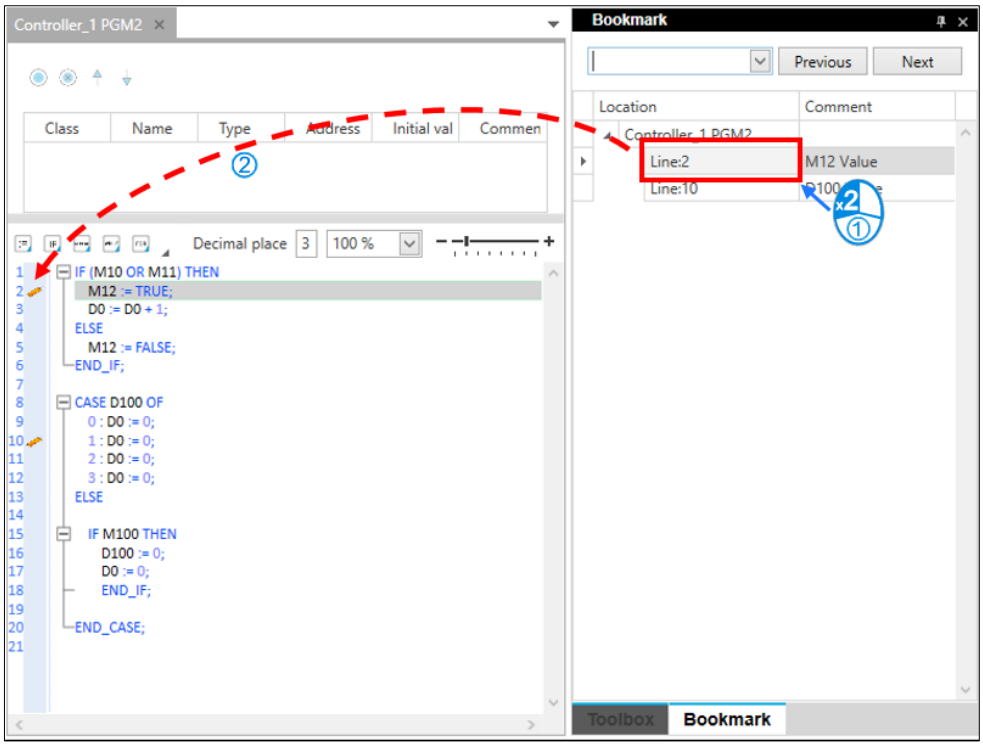


Figure 17 - 29: Navigate to Bookmark from tab

17.2.3 Bookmark function of C language

In C language, left-click the line number header (the blue area in the figure below), click the left mouse button to add a bookmark; or click the right mouse button in the editing workspace to open the right-click menu, and then click the bookmark with the left mouse button> Switch Bookmarks to add bookmarks, or click Edit> Bookmarks> Switch Bookmarks in the main menu to add bookmarks.

The following figure is an example of adding bookmarks in C language:

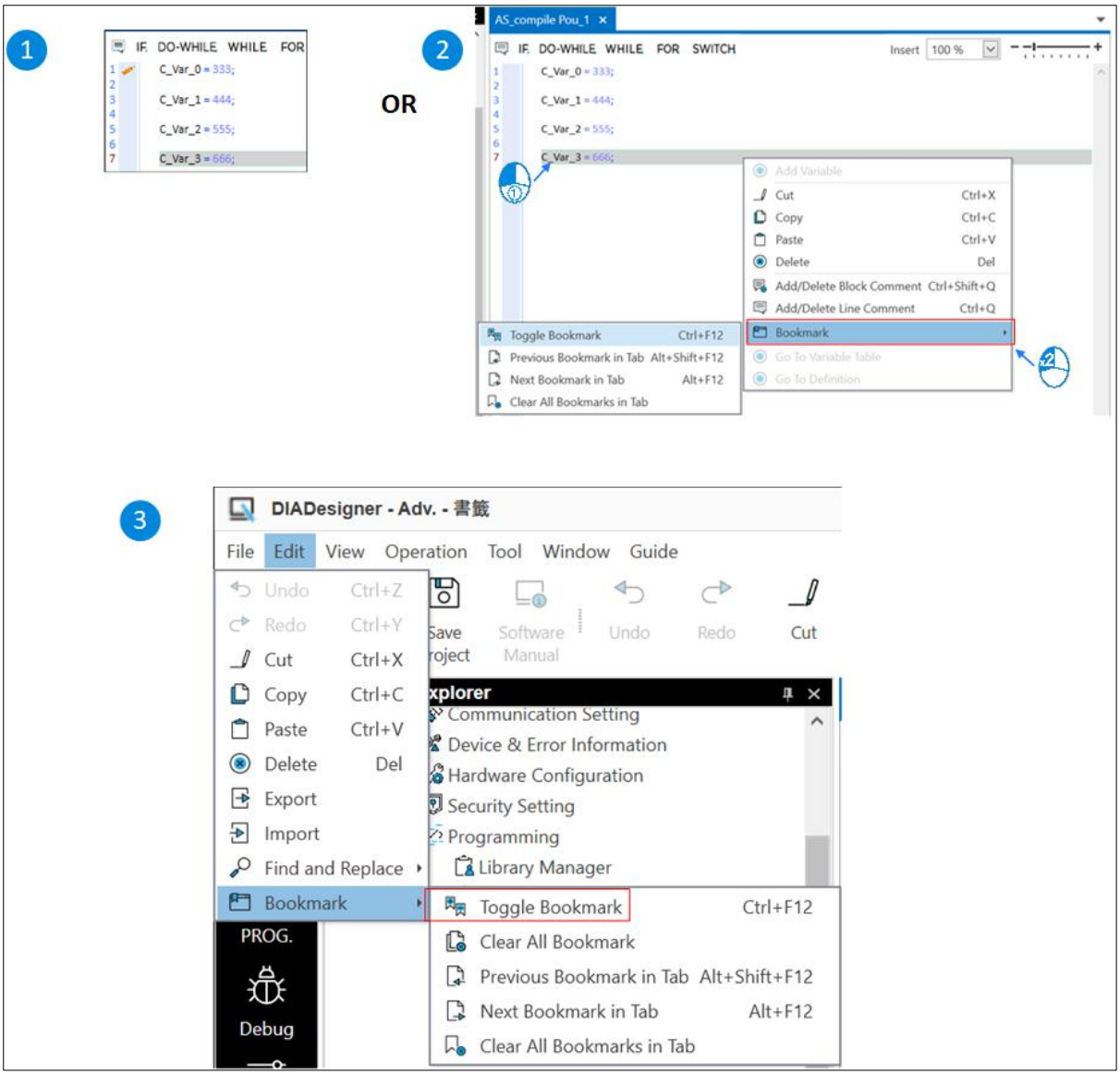


Figure 17 - 30: Add bookmarks in C language

The following figure uses the right-click menu Bookmarks> Switch bookmarks, add bookmarks

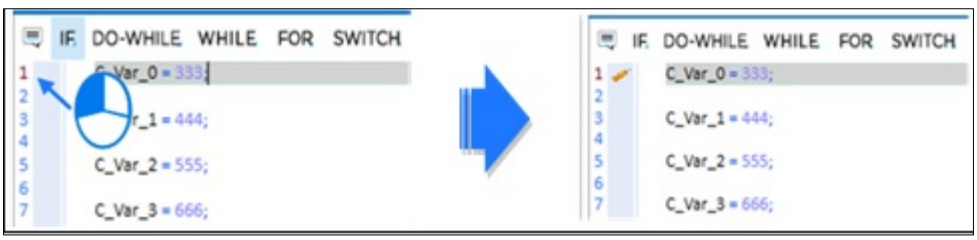


Figure 17 - 31: Add bookmark in the right-click menu

The following figure uses the main menu Edit> Bookmarks> Switch bookmarks to add bookmarks:

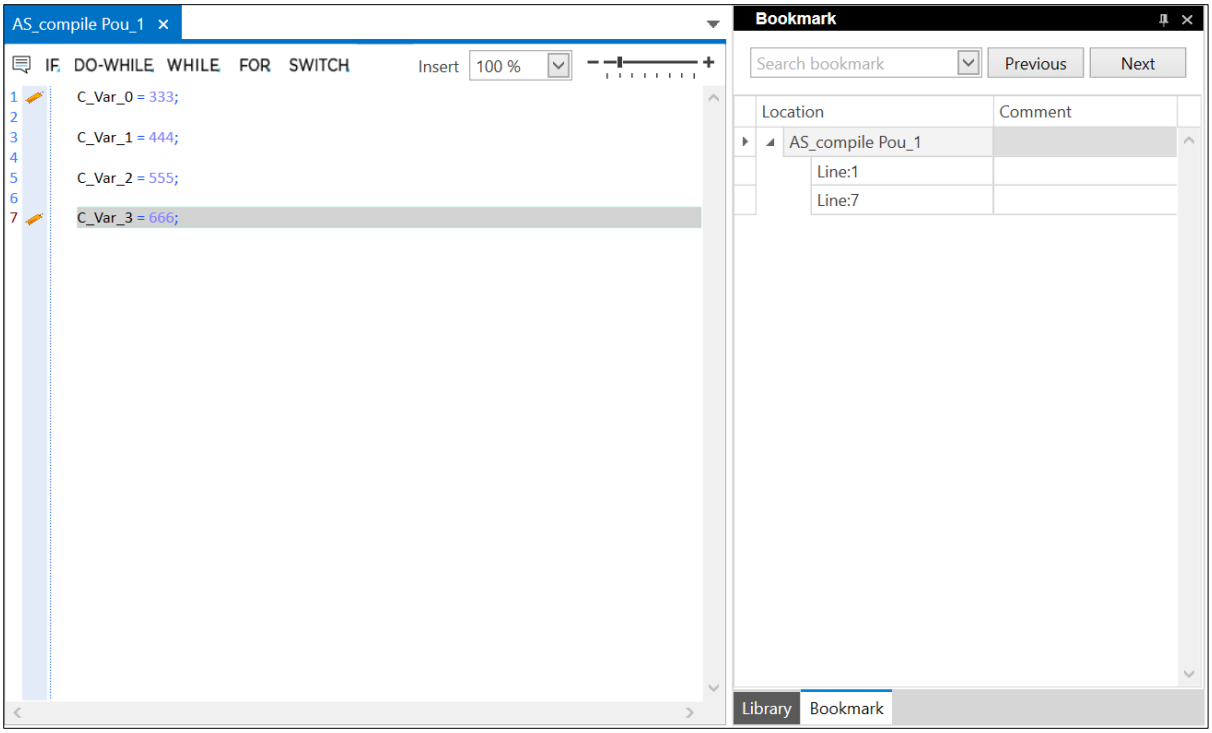


Figure 17 - 32: Edit bookmark

17.2.3.1 Add bookmark

Follow the steps below to add bookmarks in C language:

1. Select the row position where you want to add a bookmark.
2. Click the right mouse button to open the right-click menu, and then click Bookmarks> Switch Bookmarks with the left mouse button to add bookmarks, or click Edit> Bookmarks> Switch Bookmarks in the main menu to add bookmarks. The shortcut key for switching bookmarks is defined as CTRL + F12 by default.

   Result: Add a bookmark to the selected line number header.

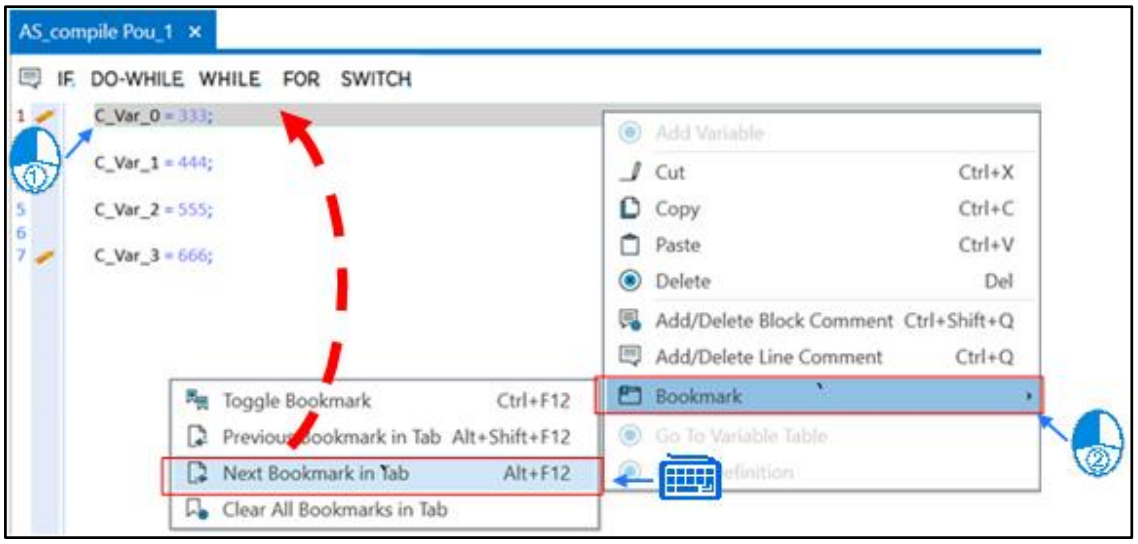


Figure 17 - 33: Left-click on the bookmark bar to add a bookmark

Example:

Line number 1 and line number 7 are the bookmark locations. It can be observed that in addition to the bookmark icon displayed in the line number header of the C language, the bookmark manager on the right will also display all the bookmark locations marked by the POU. As shown below.

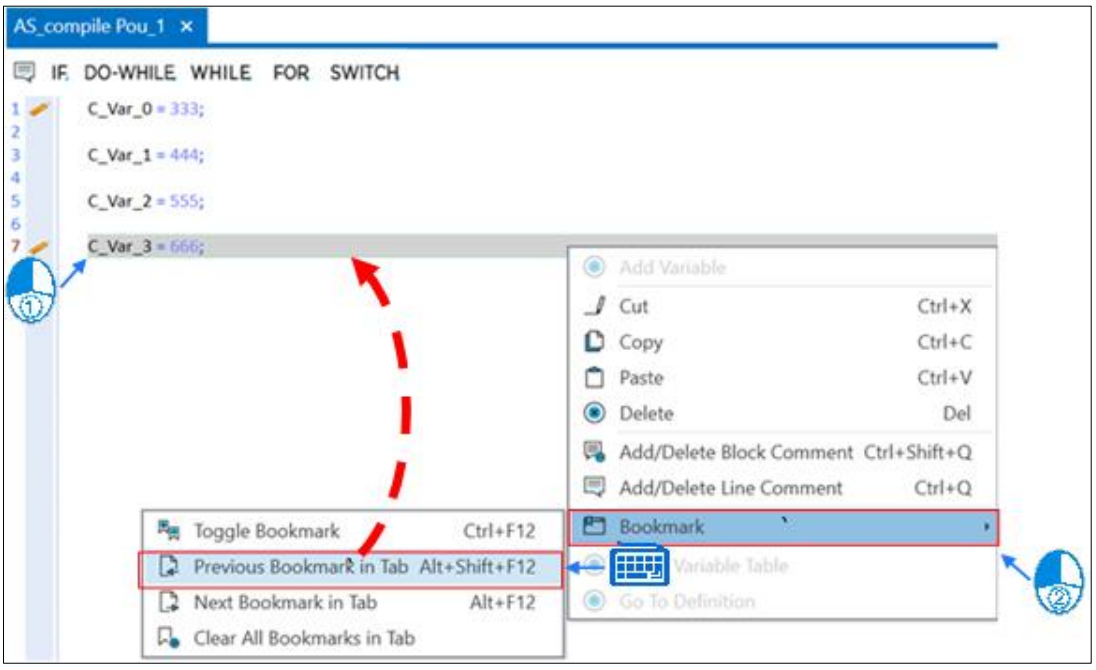


Figure 17 - 34: C POU and bookmark manager

17.2.3.2 Move to the next bookmark position

Follow the steps below to move to the next bookmark position in C language:

1. Select the current row position.
2. Click the right mouse button to open the right-click menu, and then click the left mouse button to click Bookmarks> Next Bookmark in this page to move to the next bookmark position, or click Edit> Bookmarks> Next Bookmark in this page in the main menu. You can also move to the next bookmark position. The shortcut key to move to the next bookmark position is preset as ALT+F12.
   Result: Move to the section position of the next bookmark.

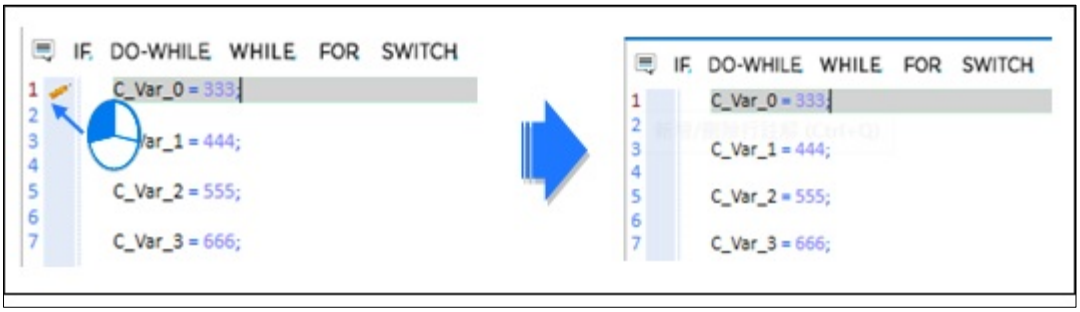


Figure 17 - 35: Next bookmark

17.2.3.3 Move to the previous bookmark position

Follow the steps below to move to the previous bookmark position in C language:

1. Select the current row position.
2. Click the right mouse button to open the right-click menu, and then click the left mouse button to click Bookmarks> Previous Bookmark in this Page to move to the next bookmark position, or click Edit> Bookmarks> Previous Bookmark in this Page in the main menu You can also move to the next bookmark position. The shortcut key to move to the next bookmark position is preset as ALT + Shift + F12.
   Result: Move to the row position of the previous bookmark.

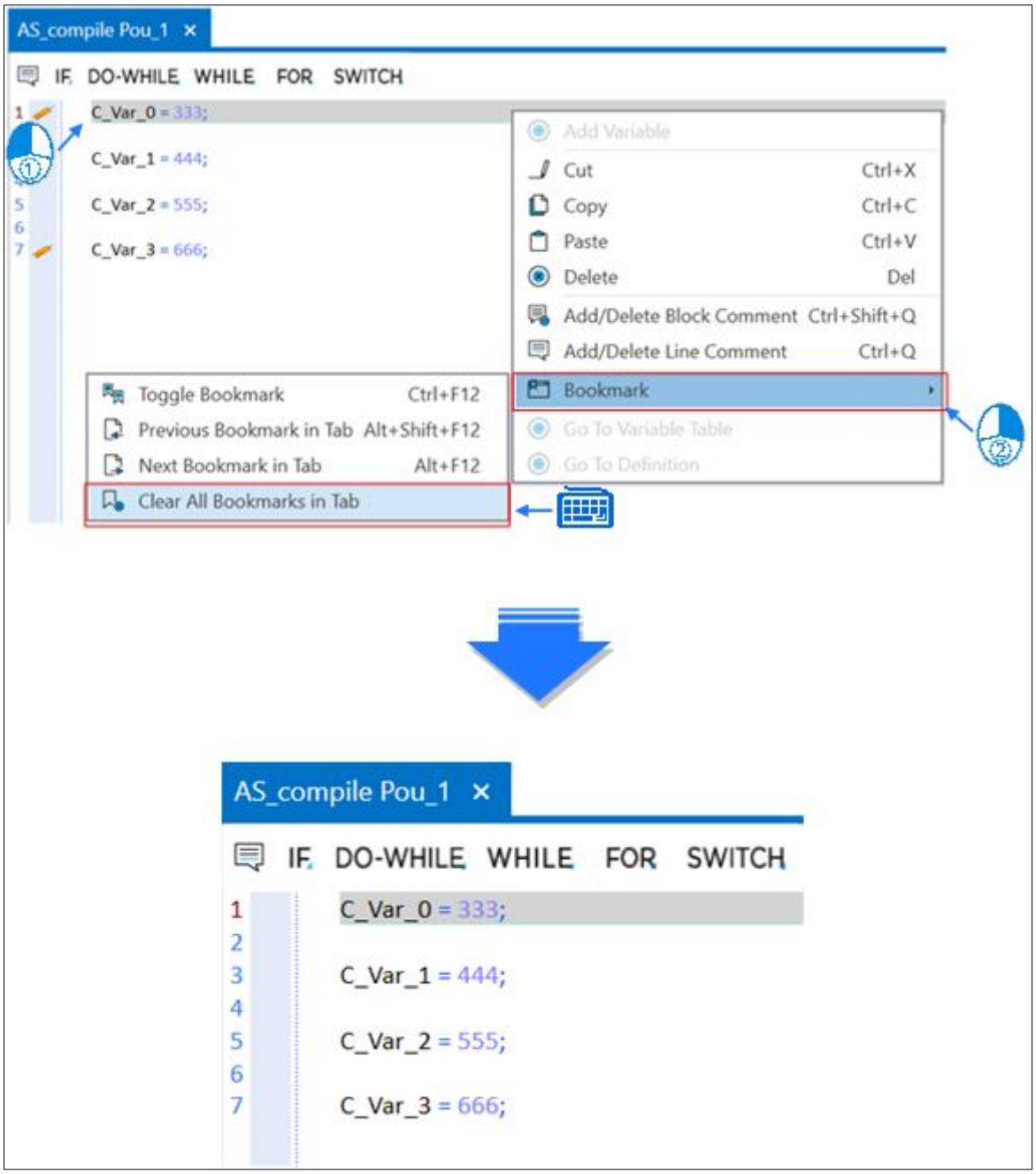


Figure 17 - 36: Previous bookmark

17.2.3.4 Delete bookmark

Follow the steps below to delete bookmarks in C language:

1. User can select the line position where the present bookmark is marked and the bookmark needs to delete.

2. Click the right mouse button to open the right-click menu, and then click the left mouse button to click Bookmark> Switch Bookmark to delete the bookmark, or click Edit> Bookmark> Switch Bookmark in the main menu to delete the bookmark. The shortcut key for switching bookmarks is defined as CTRL + F12 by default.
   Result: Delete the selected line number header bookmark.

Figure 17 - 37: Left-click on the bookmark bar to delete the bookmark

17.2.3.5 Clear all bookmarks

Follow the steps below to delete all bookmarks in C language:

1. Click the right mouse button to open the right-click menu, and then click the left mouse button to click Bookmarks> Clear Bookmarks on this page to delete all bookmarks, or click Edit> Bookmarks> Clear Bookmarks on this page in the main menu to delete All bookmarks.
   Result: delete all bookmarks.

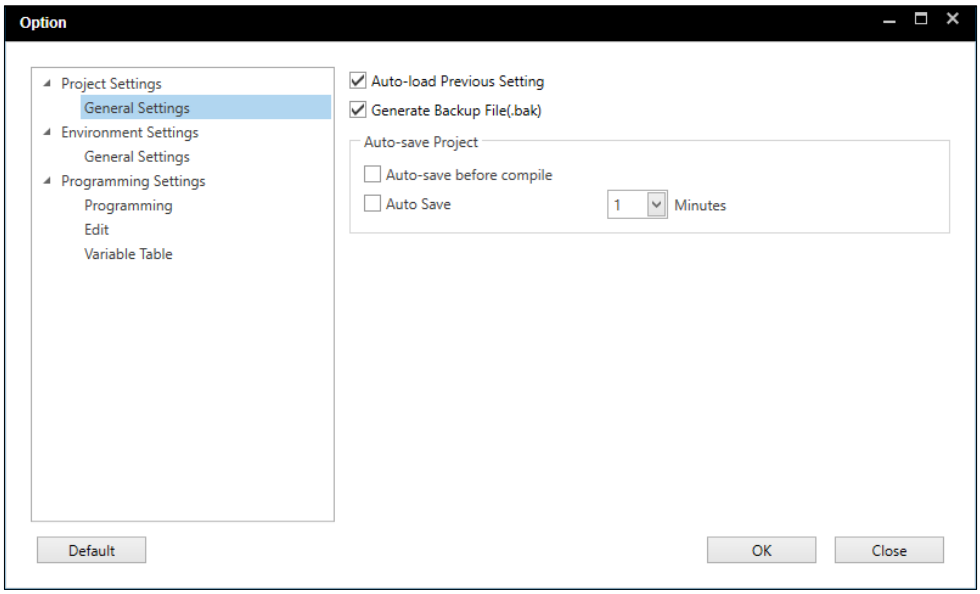


Figure 17 - 38: Clear all bookmarks

***Note**: In the bookmark manager window, double-click any bookmark location, and it will immediately move to the bookmark location*

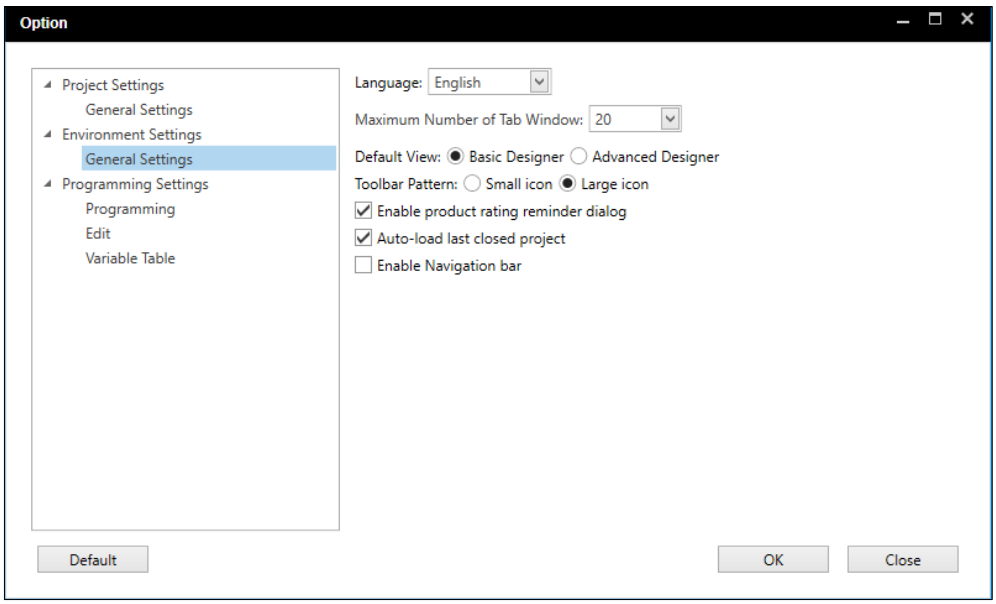


Figure 17 - 39: Double tap to jump to bookmark

17.3 Option

User can access Option function via **Tool** > **Option** in the menu bar of DIADesigner. **Option** window is shown in the following figure.

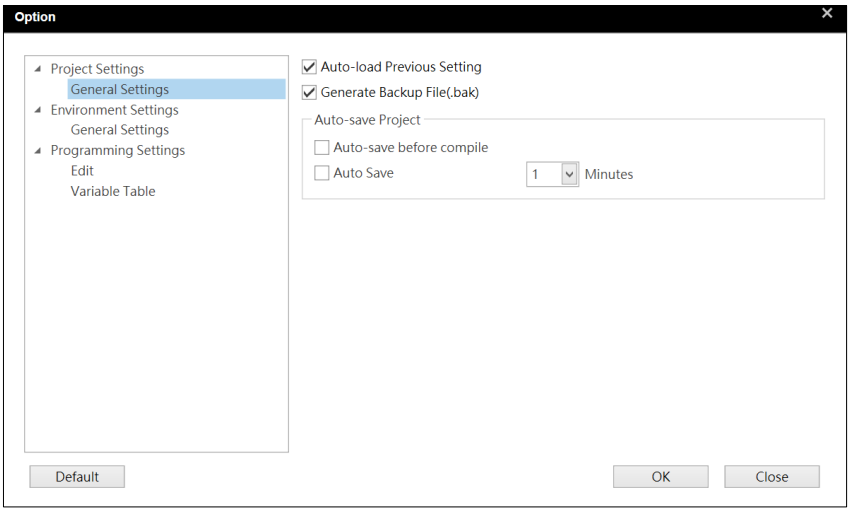


Figure 17 - 40: Option

Option window has the following options:

- Project Settings
- Environment Settings
- Programming Settings

17.3.1 Project Settings

Project Settings option is shown by default when **Option** window displays.

The functions available in **Project Settings** > **General Settings** in the **Option** window are described in the following table:

Function	Description
Auto-load Previous Setting	If this item is selected, then a project which was edited last and which was not closed is opened automatically whenever DIADesigner is started.
Generate Backup file (.bak)	If this item is selected, a project which is saved also creates a backup file and can be restored when the disk drive is damaged.
Auto-save Project – Auto-save Before Compile	If this item is selected, the project is automatically saved before compiling.
Auto-save Project – Auto Save	If this item is selected, the project is saved automatically as per the time interval set in **Auto-Save Project – Minutes**.
Auto-save Project - Minutes	If the **Auto Save** is selected, the **Minutes** drop-down list provides an interval time for the system to automatically save the project. Options for time intervals are: • 1 • 5 • 10 • 30 • 60 ***NOTE**: 1 is the default value.*

17.3.2 Environment Settings

The **Environment Settings** option is shown in the following figure.

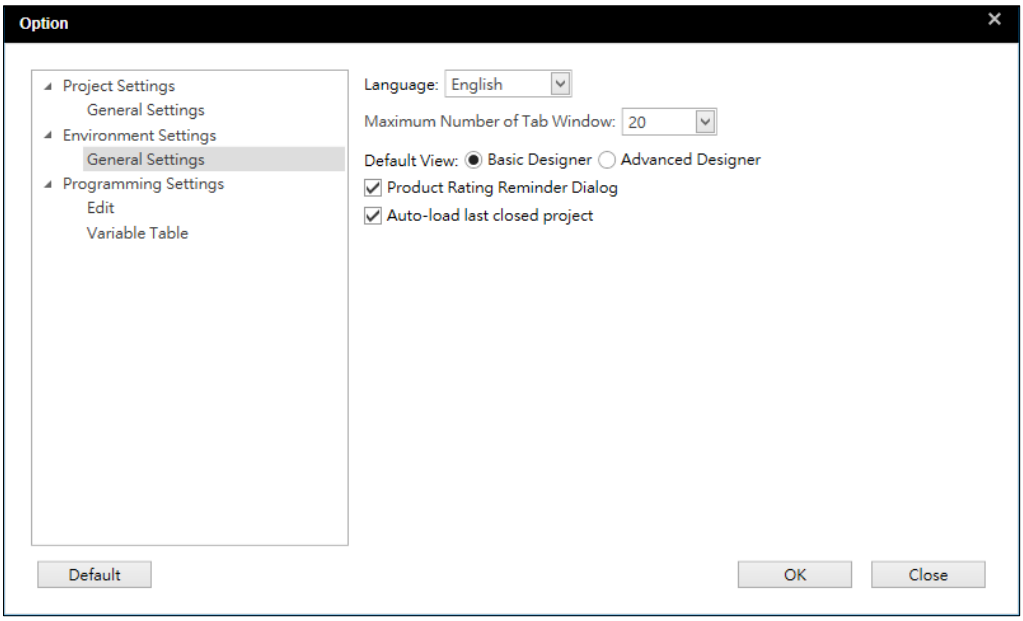


Figure 17 - 41: Environment Settings

The functions available in **Environment Settings** > **General Settings** in the **Option** window are described in the following table:

Function	Description
Language	User can select the language in the drop-down list. Options are: • Traditional Chinese • Simplified Chinese • English (US) ***NOTE**: Language displayed by default is the OS language.*
Maximum number of Tab Window	User can select the maximum number of tab windows to display simultaneously from the drop-down list. Value can be set between 2 ~ 20.
Default View	Select one interface for next starting DIADesigner. • Basic Designer • Advanced Designer
Product Rating Reminder Dialog	Check to open the Rating Reminder Dialog window when exiting DIADesigner.
Auto-load last closed project	Check to open the last closed project for next starting DIADesigner.
Toolbar style	The user can select the display size of the toolbar icon, the options are:

Function	Description
	• Small icon • Large icon ***NOTE**: The default is a small icon.*
Enable product review reminder window	The user can choose whether to display the product evaluation reminder window when closing DIADesigner ***NOTE**: The default is enabled.*
Automatically load the last closed project	User can choose whether to automatically load the last closed project when opening DIADesigner ***NOTE**: The default is enabled.*
Enable navigation bar	The user can choose whether to display the BASIC navigation bar when switching to the ADV page ***NOTE**: The default is not enabled.*

17.3.3 Programming Settings

The **Programming Settings** is shown in the following figure.

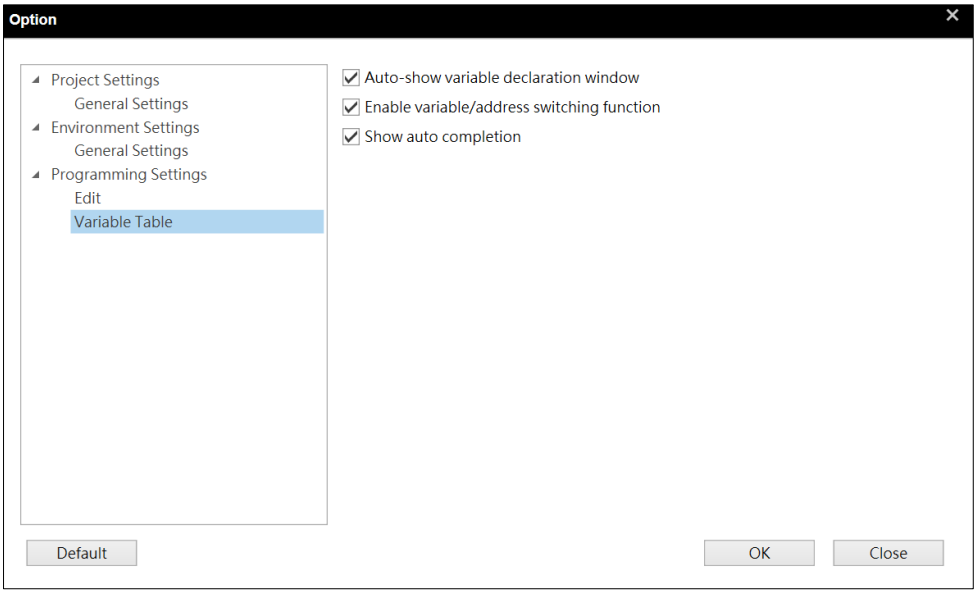


Figure 17 - 42: Programming Settings – Variable Table

The functions available in **Programming Settings** > **Variable Table** in the **Option** window are described in the following table:

Function	Description
Auto-show variable declaration window	If this item is selected, the **Add variable to POU** window displays whenever a variable is created.
Enable variable/address switching function	If this item is selected, the Variable/Address mode can be selected in the icon tool bar so as to change over between Variable and Address, Variable, Address modes.
Show auto-completion	If this item is selected, input the variables in the program, the system will list all the set items for the variable and user can use the mouse or keyboard to select from the list.
Default	Click to set all functions to the default value.
OK	Click to apply the changes.
Close	Click to close the **Option** window.

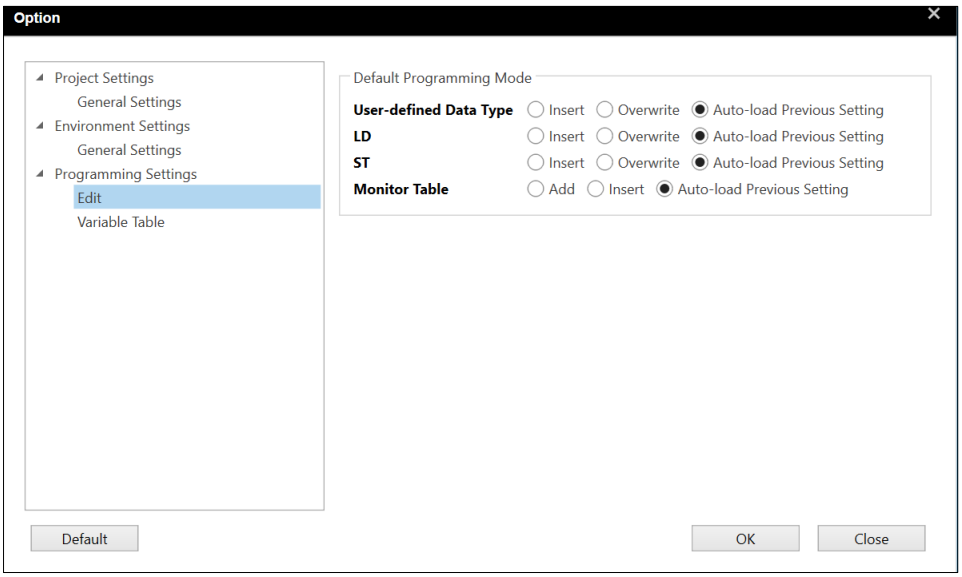


Figure 17 - 43: Programming Settings – Edit

The functions available in **Programming Settings** > **Edit** in the **Option** window are described in the following table:

Function	Description
Default Programming Mode	
User-defined Data Type	Select a programming mode for the keyboard input. • Insert
LD	

Function	Description
ST	• Overwrite • Auto-load Previous Setting
Monitor Table	Select a programming mode for the keyboard input. • Add • Insert • Auto-load Previous Setting

Chapter 18: Trouble Shooting

18.1 Error Log

Under normal connection with the hosts, user can select **Controller** > **Device and Error Information** to view logs regarding the modifications and errors.

Three type of logs are available: -

- Error Information
- Program Download Log
- Status Change Log

18.1.1 Error Information

The **Error Information** tab displays the errors in devices along with the date and time information. **Error Information** tab for different device series may be different. **Error Information** tab for PLC is shown in the following figure.

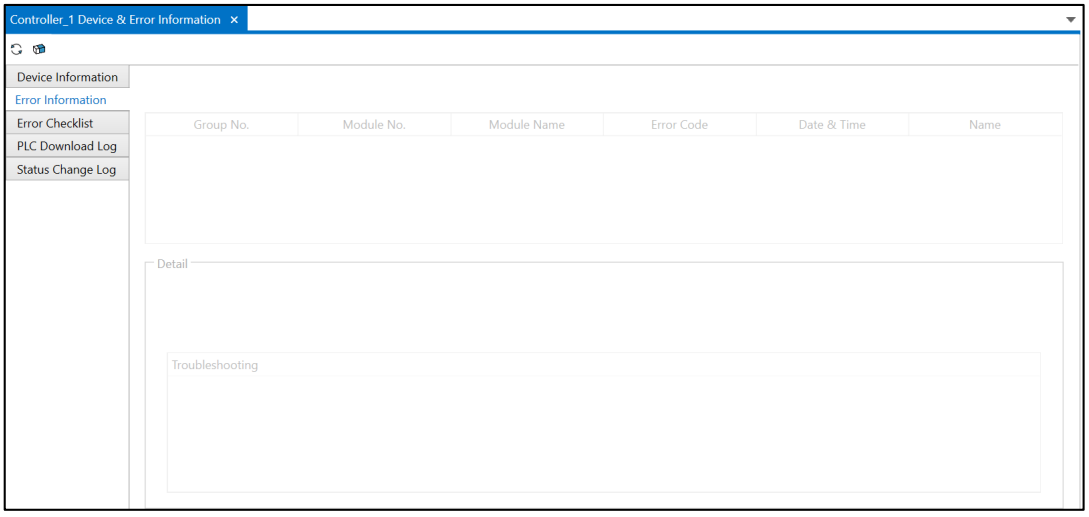


Figure 18 - 1: PLC – Error Information

The functions in **Error Information** tab are described in the following table:

Function	Description
Group No.	**Displays the group number in which the error is active.**
Module No.	Displays the module number in which the error is active.
Module Name	Displays the module name in which the error is active.
Error Code	Displays the error code.
Date and Time	Displays the date and time of the error code.
Name	Displays the error message.
Detail	Displays the details of error.
Troubleshooting	Displays how to troubleshooting and fix the error.
Refresh	Click **Refresh** to let the system retrieve data from the host and update the data in the log.
Reset	Click **Reset** to clear the table and internal host log as well as the host reset status. ***NOTE**: If the error is still active, the entry remains in the error status and new error log is created.*

18.1.2 PLC Download Log

The **PLC Download Log** tab displays the history log related to Download/Upload of programs and Parameters for PLC devices as shown in the following figure.

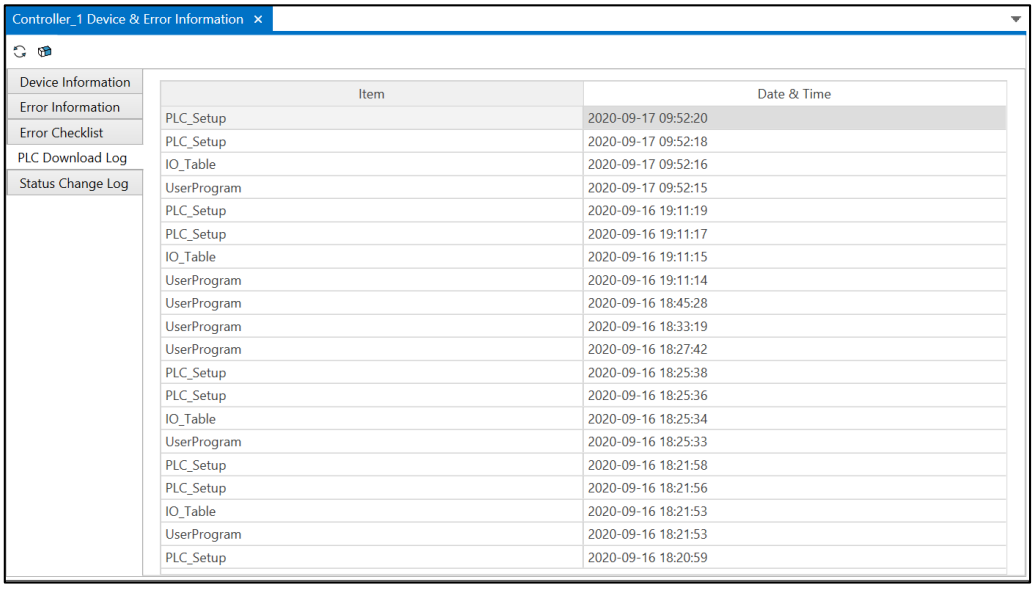


Figure 18 - 2: PLC - PLC Download Log

The functions in the **PLC Download Log** tab are described in the following table:

Function	Description
Item	Displays the name of the item in Download or Upload action. **Example**: User Program, PLC Setup, IO Table etc.
Date & Time	Displays the date and time of the upload or download item.
Refresh	Click **Refresh** to let the system retrieve data from the PLC and update the data in the log.
Reset	Click **Reset** to clear the table and internal host log as well as the host reset status.

18.1.3 Status Change Log

The **Status Change Log** tab displays the log related to the change of PLC CPU status as shown in the following figure.

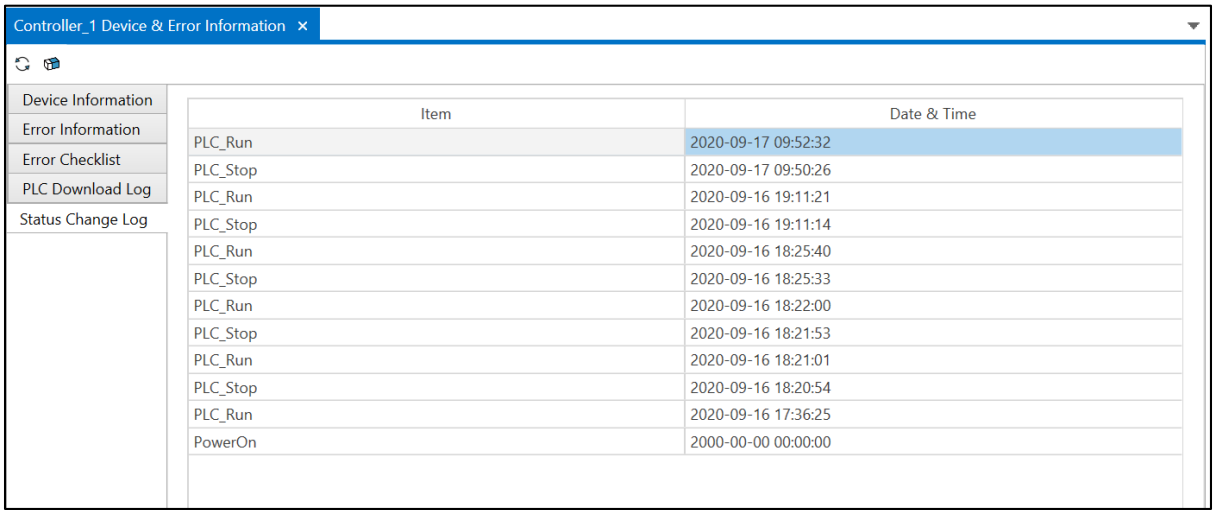


Figure 18 - 3: PLC – Status Change Log

The functions in **Status Change Log** tab are described in the following table:

Item	Description
Item	Displays the name of the item in PLC CPU status. **Example**: PLC RUN, PLC STOP etc.
Date & Time	Displays the date and time of the PLC CPU status change.
Refresh	Click **Refresh** to let the system retrieve data from the PLC and update the data in the log.
Reset	Click **Reset** to clear the table and internal host log as well as the host reset status.

18.2 Error Checklist

User can check the description of the errors and error codes offline with the help of **device** > **Device and Error Information** > **Error Checklist**. **Error Checklist** tab displays all the error codes, names, description, alarm causes, fix methods, corrective actions etc.

The **Error Checklist** tab of a Controller is shown in the following figure.

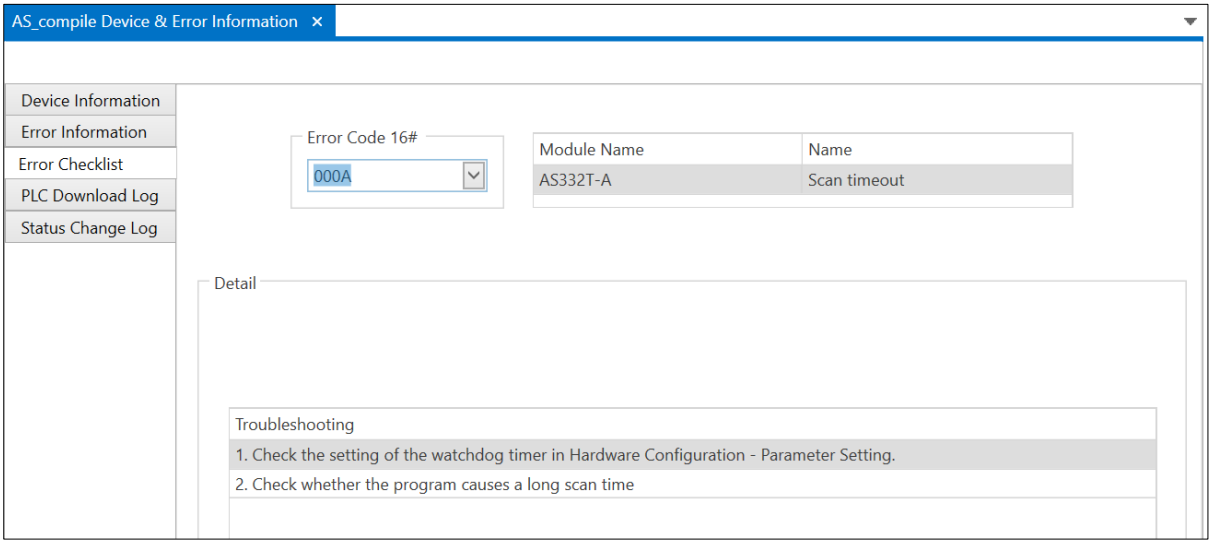


Figure 18 - 4: Retrieve Error Code

The **Error Checklist** tab of a Servo Device is shown in the following figure.

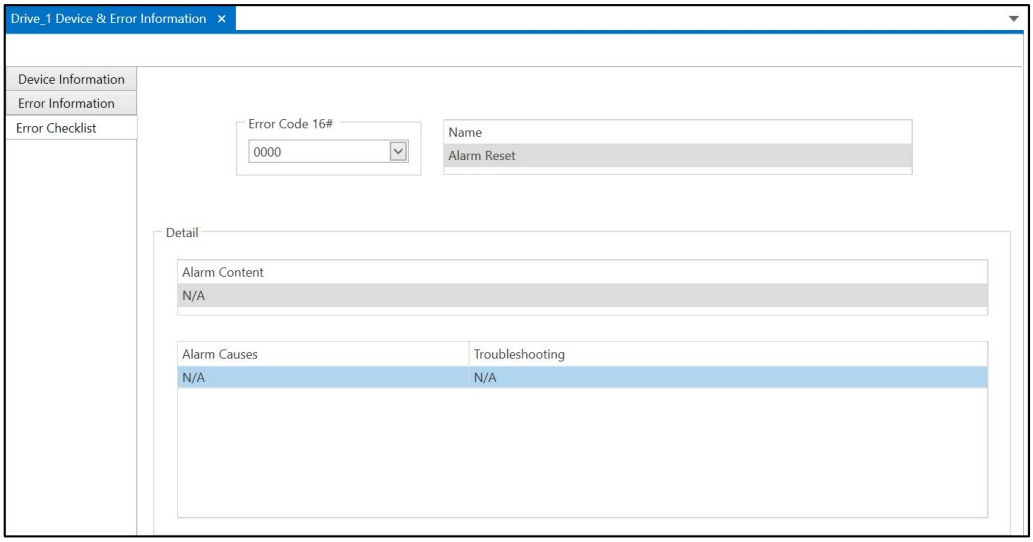


Figure 18 - 5: Error Checklist – Servo Device

The **Error Checklist** tab of a Drive is shown in the following figure.

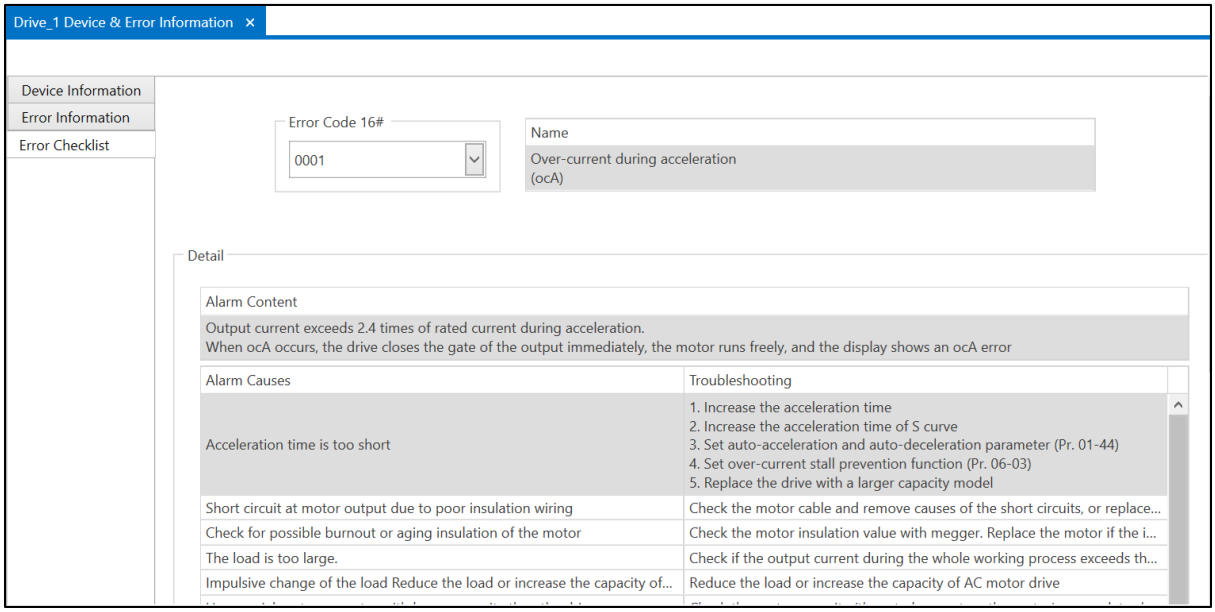


Figure 18 - 6: Error Checklist – Drive

Chapter 19: Appendix

19.1 Installing USB driver for AS series CPU module

19.1.1 Installing USB driver in Windows 7

The installation of USB driver for an AS series PLC CPU module on Windows 7 operating system is introduced below. If user want to install USB driver for an AS series PLC CPU module on another operating system, refer the instructions in that operating system for more information about the installation of new hardware.

Follow these steps for the installation of USB driver in Windows 7 operating system:

Pre-requisite:

Make sure that the PLC is powered up. Connect the PLC to a USB port on the computer with a USB cable.

1. Open **Control Panel** > **Device Manager**.

   Result: **Device Manager** displays the device as shown in the following figure.

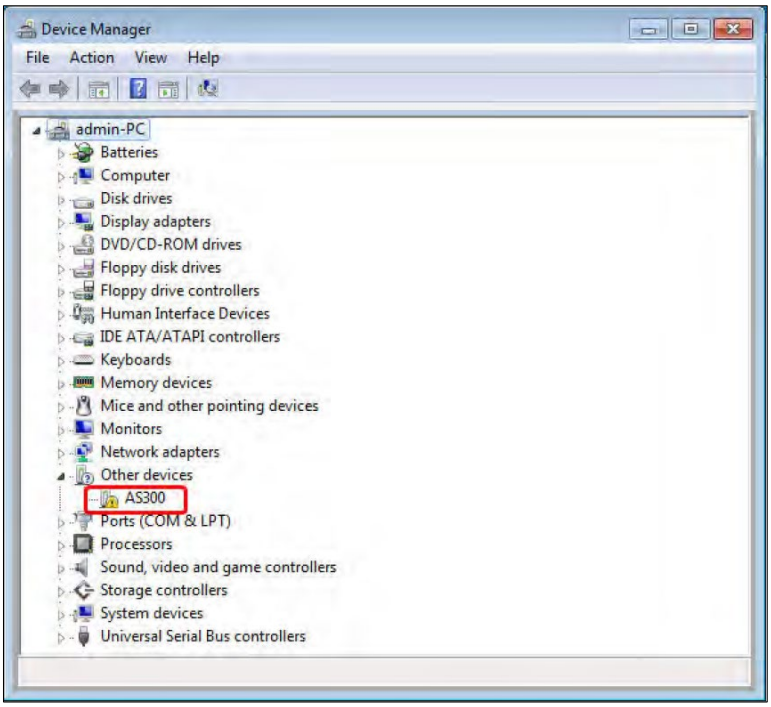


Figure 19 - 1: Device Manager

2. Select and double-click **AS200/AS300**.

   Result: The **AS200 Properties/ AS300 Properties** window displays as shown in the following figure.

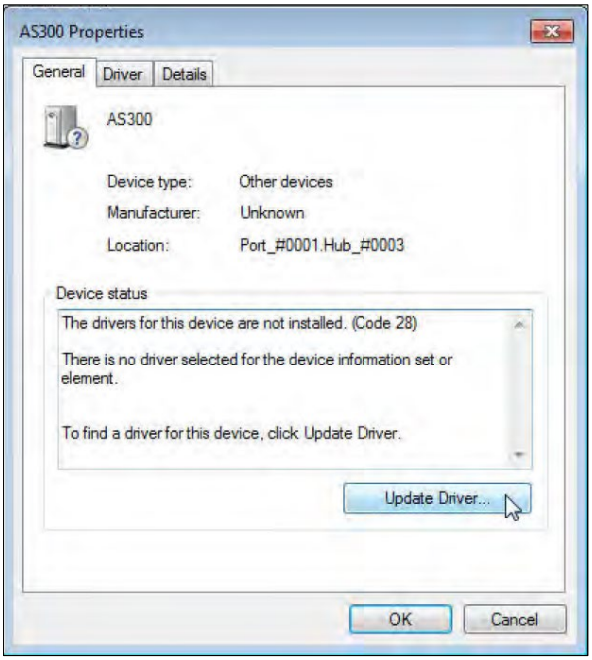


Figure 19 - 2: AS300 Properties

3. Click on **Update Driver…**

   Result: The **Update Driver Software – AS200/ Update Driver Software – AS300** window displays as shown in the following figure.

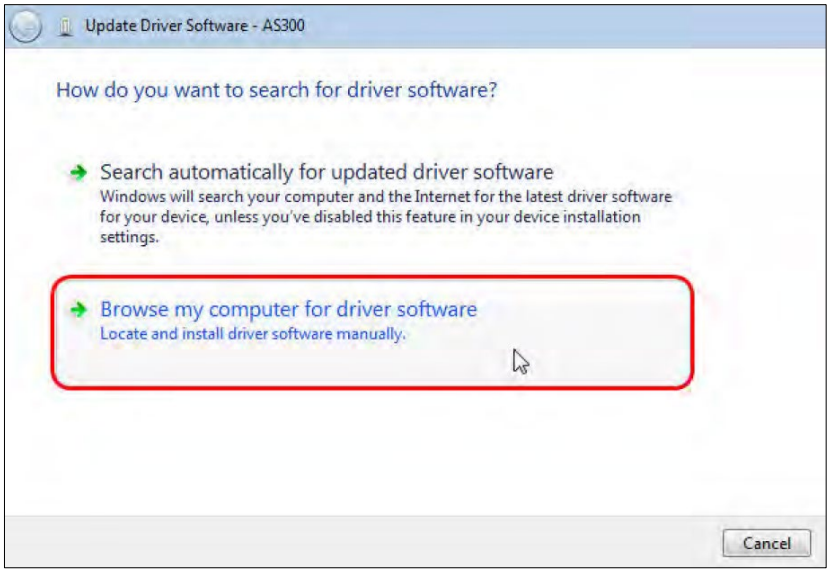


Figure 19 - 3: Update Driver Software - AS300

4. Click on **Browse my computer for driver software.**

   Result: **Update Driver Software – AS300** window displays the path to browse the driver software as shown in *Figure 17-4: Update Driver Software - AS300 - Browse*.

***NOTE**:*

After the installation of DIADesigner, the driver for AS series CPU module is available in the folder of installation path - DIAStudio\DIADesigner\DIADesigner version\drivers\Delta_PLC_USB_Driver

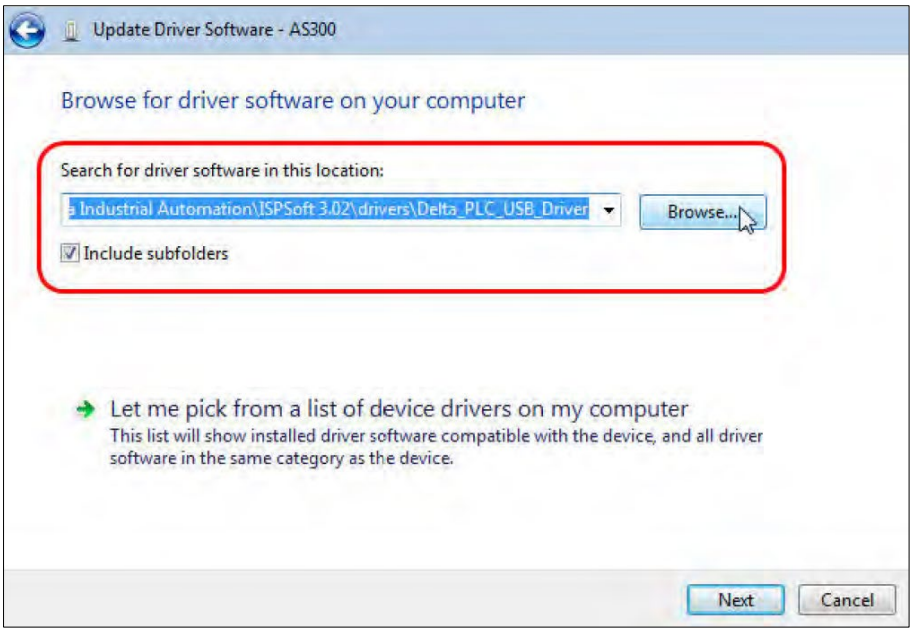


Figure 19 - 4: Update Driver Software - AS300 - Browse

5. Click on **Browse** and locate the correct path as per installation.

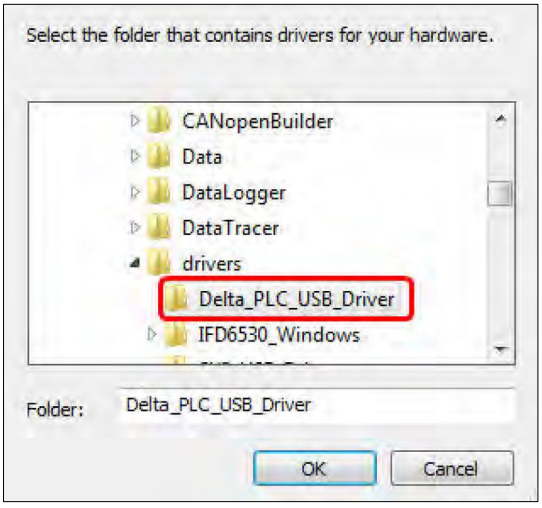


Figure 19 - 5: Folder selection

6. Click on **OK** button.

   Result: Path is selected.

7. Click **Next** in **Update Driver Software – AS300** window.

Result: The system will start installing the driver. If the **Windows Security** window appears during the installation as shown in the following figure.

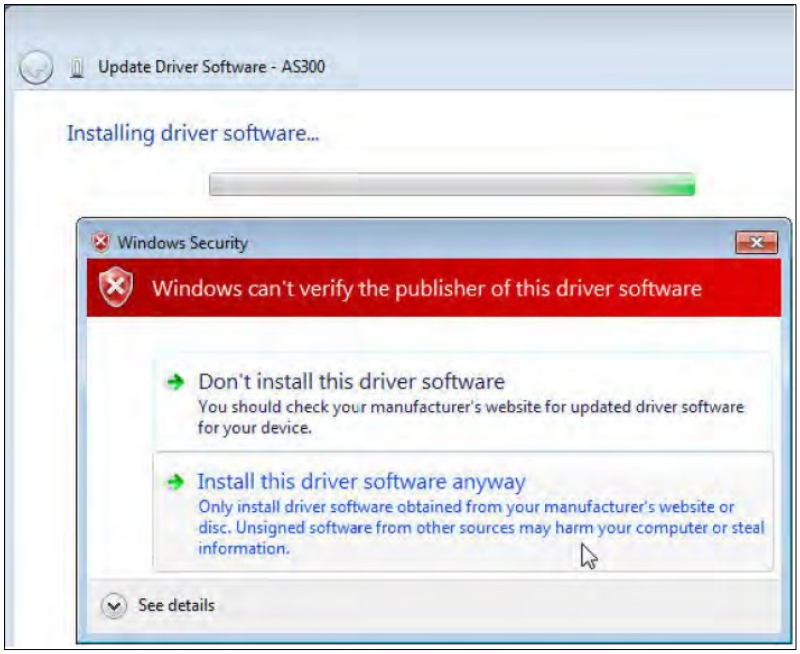


Figure 19 - 6: Windows Security

8. Click **Install this driver software anyway**.

   Result: After the installation is complete, the **Update Driver Software** window displays the status as shown in the following figure.

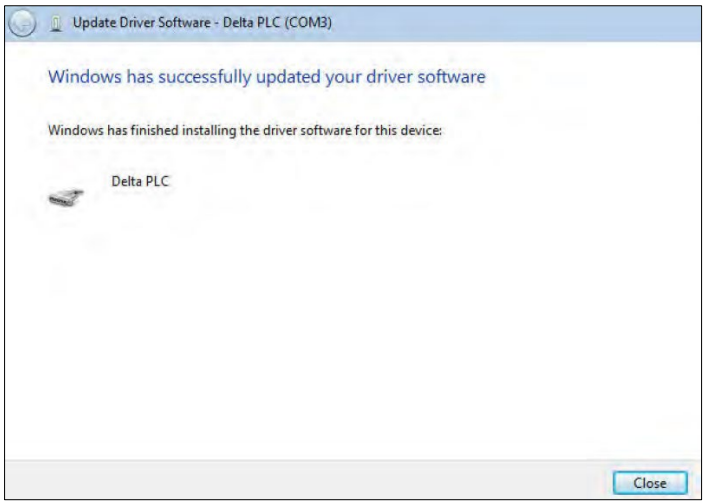


Figure 19 - 7: Update Driver Software Delta PLC - Successful

9. Click on **Close** button.

10. Open the **Device Manager** window after the installation is finished.

    Result: If the name of the USB device connected is under **Ports (COM and LPT)**, the installation of the USB driver is successful. The operating system assigns a communication port number to the USB device as shown in the following example.

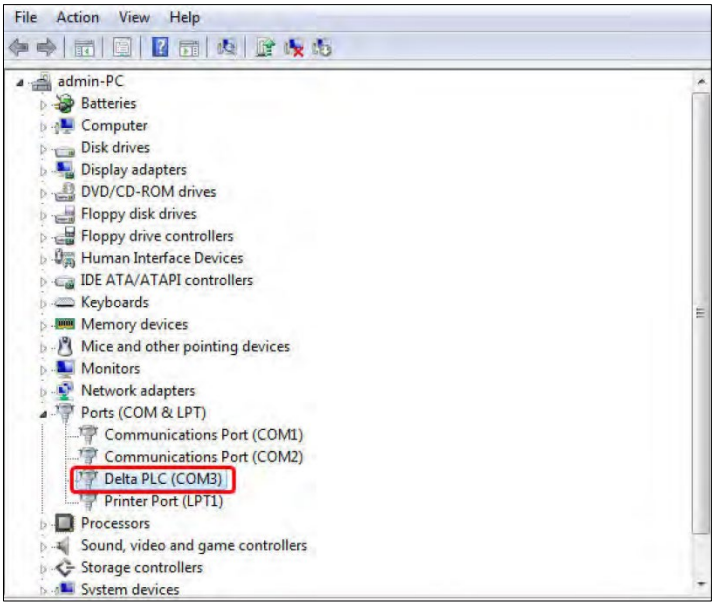


Figure 19 - 8: Example of successful USB Driver installation

***NOTE**:*

If the PLC is connected to another USB port on the computer, the system can ask user to install the driver again. User can follow the same steps as mentioned above, and install the driver again. After the driver is installed, the communication port number that the operating system assigns to the USB device can be different.

19.1.2 Installing 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 user to disable driver signature enforcement functionality in Windows 8.1 to ensure successful installation of USB driver.

Follow these steps to disable driver signature enforcement in Windows 8.1:

1. Press **Windows** ( ) and **I** keys simultaneously on the keyboard.

   Result: The **Settings** interface displays as shown in the following figure.

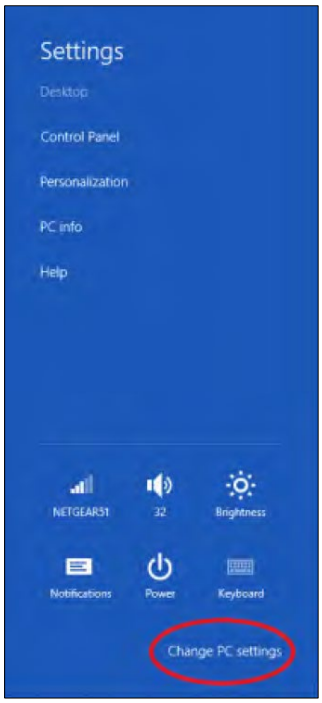


Figure 19 - 9: Settings – Windows 8.1

2. Click **Change PC settings** in the **Settings** interface.

   Result: The **PC settings** interface displays as shown in the following figure.

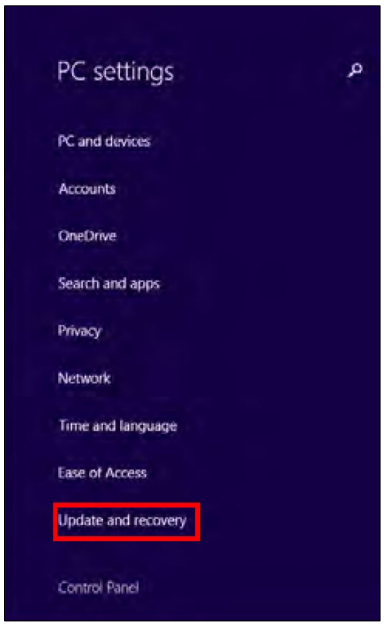


Figure 19 - 10: PC Settings – Windows 8.1

3. Click **Update and recovery** in the **PC settings** interface.

   Result: **Update and recovery** interface displays as shown in the following figure.

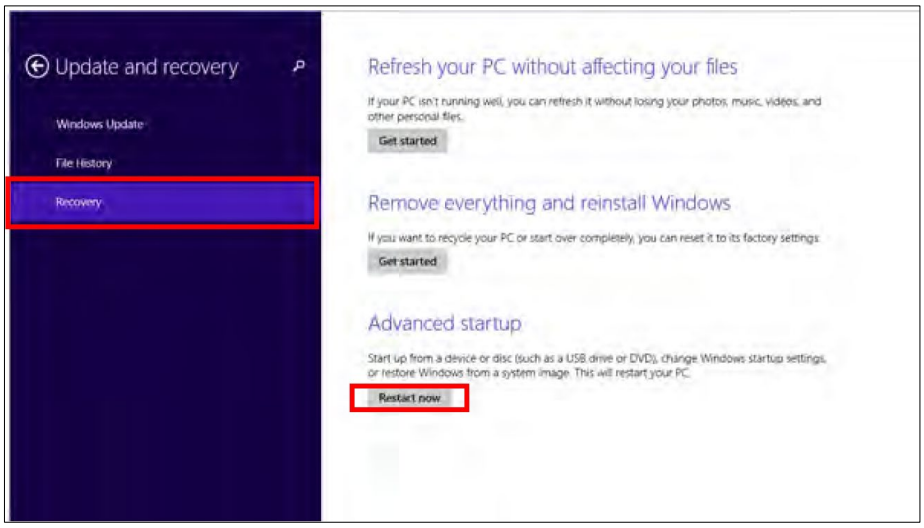


Figure 19 - 11: Update and recovery - Windows 8.1

4. Click **Recovery** in the **Update and recovery** interface.

   Result: **Refresh your PC without affecting your files** displays.

5. Click on **Restart now** in the **Advanced startup** section in the **Refresh your PC without affecting your files**.

   Result: Computer will restart and it will display the **Choose an option** interface as shown in the following figure.

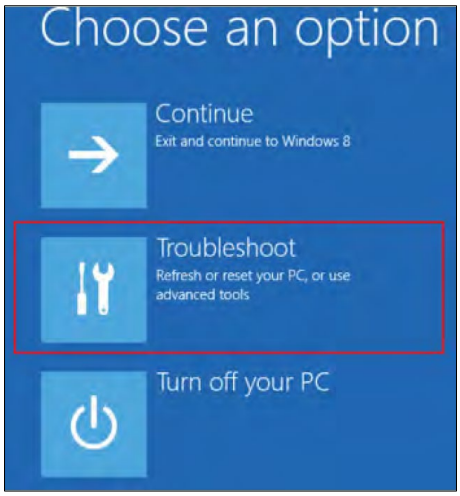


Figure 19 - 12: Choose an option - Windows 8.1

6. Click on **Troubleshoot** in **Choose an option** interface.

   Result: The **Troubleshoot** interface displays as shown in the following figure.

Figure 19 - 13: Troubleshoot - Window 8.1

7. Click on **Advanced options** in **Troubleshoot** interface.

   Result: The **Advanced options** interface displays as shown in the following figure.

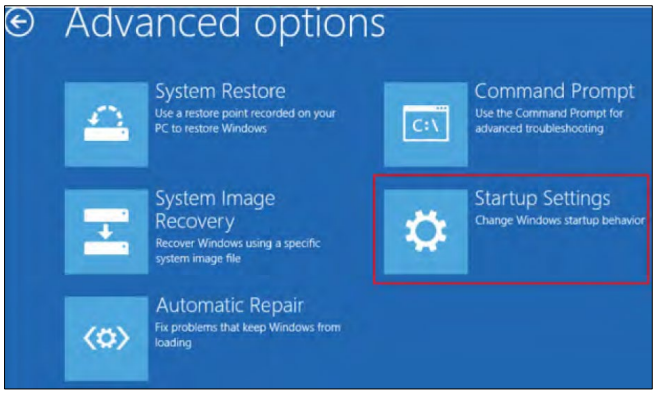


Figure 19 - 14: Advanced options - Windows 8.1

8. Click **Startup Settings** in the **Advanced options** interface.

   Result: The **Startup Settings** interface displays as shown in the following figure.

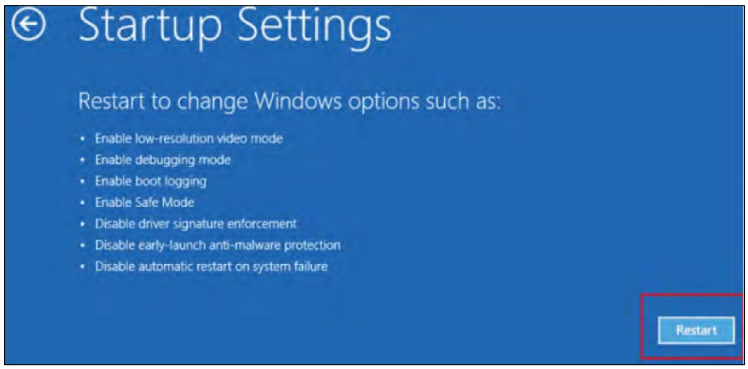


Figure 19 - 15: Startup Settings - Windows 8.1

9. Click on **Restart** button**.**

   Result: Computer restarts.

10. After the computer restarts, press **7** or **F7** to choose **Disable driver signature enforcement**.

    Result: System will direct the user to Windows 8.1 operating page. Now User can install the Delta PLC USB driver. Procedure is same as the one mentioned in Windows 7.

19.1.3 Installing 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 user to disable driver signature enforcement functionality in Windows 10 to ensure successful installation of USB driver. This operation is only valid for a single time. The setting will return to its original state after restarting.

Follow these steps to disable driver signature enforcement in Windows 10 operating system:

1. Click **Windows** icon > **Settings** and press **SHIFT** key and click **Restart.**

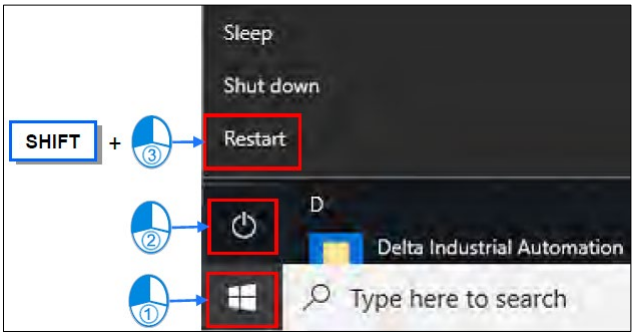


Figure 19 - 16: Restart

Result: Computer restarts. After completing the restart, **Choose an option** window displays as shown in the following figure.

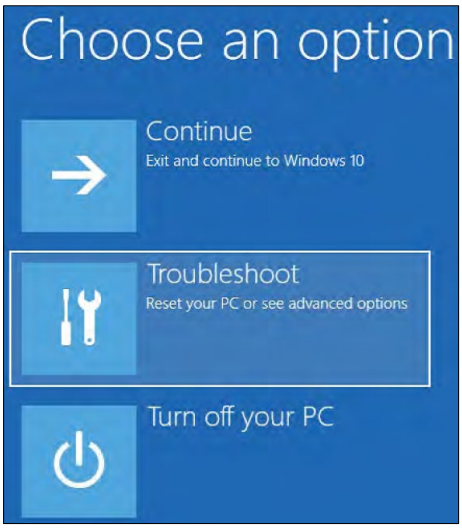


Figure 19 - 17: Choose an option - Windows 10

2. Click **Troubleshoot** in **Choose an option** window.

   Result: **Troubleshoot** window displays as shown in the following figure.

Figure 19 - 18: Troubleshoot - Windows 10

3. Click **Advanced options** in the **Troubleshoot** window.

   Result: **Advanced options** window displays as shown in the following figure.

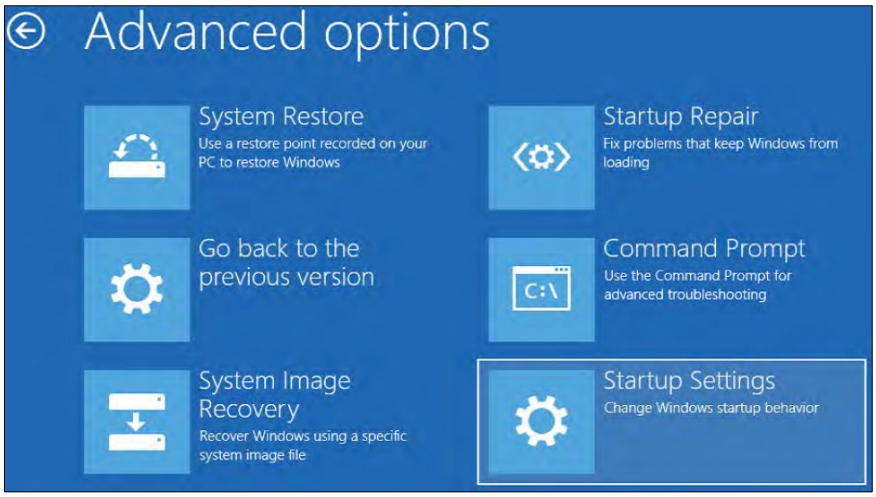


Figure 19 - 19: Advanced options - Windows 10

4. Click **Startup Settings** in the **Advanced options** interface.

   Result: The **Startup Settings** interface displays as shown in the following figure.

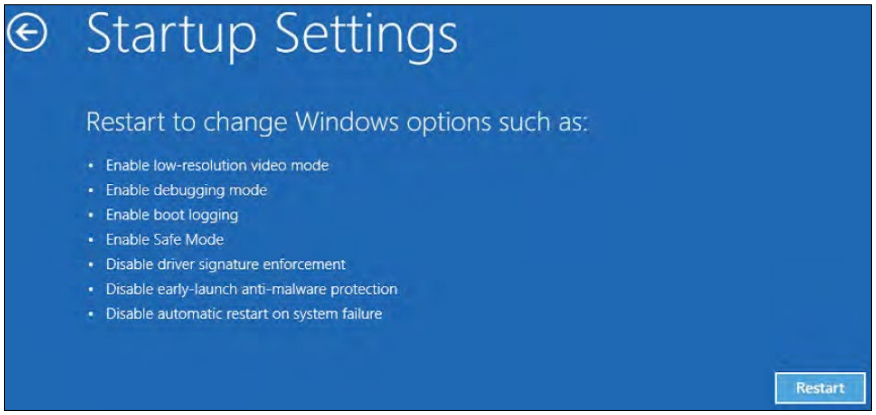


Figure 19 - 20: Startup Settings - Windows 10

5. Click on **Restart**.

   Result: Computer restarts and displays the **Startup Settings** as shown in the following figure.

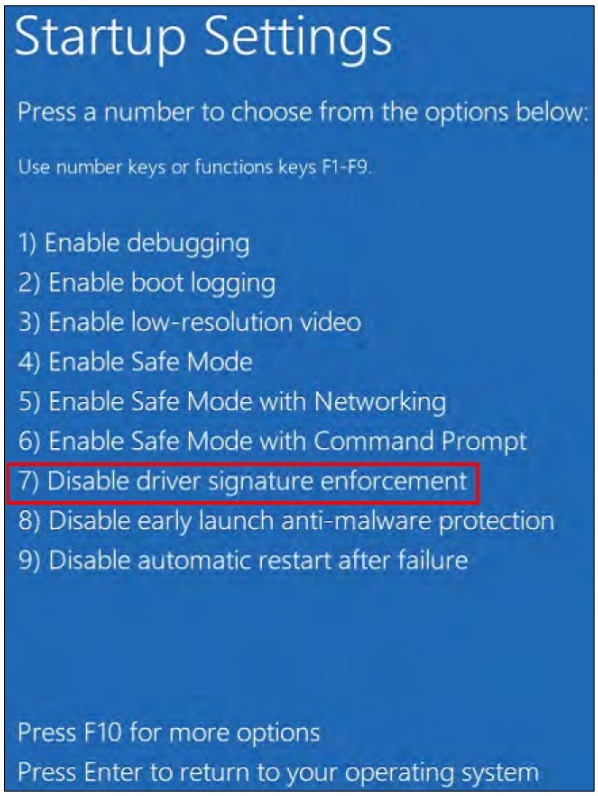


Figure 19 - 21: Startup Settings - Windows 10

6. Press 7 or F7 to choose **Disable driver signature enforcement**.

   Result: System directs the user to Windows 10 operating page. Now User can install the Delta PLC USB driver. Procedure is same as the one mentioned in Windows 7.

19.2 Important points related to CPU modules

User have to read the manuals of different models before they develop projects. The points to which user have to pay attention when they develop a project for an AS series CPU module are discussed in this section.

19.2.1 Addresses in AS series CPU modules

19.2.1.1 Address types of AS series CPU modules

The address type of AS series CPU modules are described in the following table:

Address 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 (SFC).
D	Bit/Word	Data Register Operation data is stored in a data register. (*1)
T	Bit	Timer If a timer is used as a bit address, 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 address, the values stored in the timer is the present value of the timer.
C	Bit	Counter If a counter is used as a bit address, 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.

Address type	Access Property	Description
	Word	If a counter is used as a word address, 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 address, 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
	Word	32-bit counter If a 32-bit counter is used as a word address, 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 an address modified. (*2)
SM	Bit	Special Auxiliary Register It is used as a state flag for a special function of the system. (*3)
SR	Word	Special Data Register It is used as a data register for a special function of the system (*3)

***Footnote**:*

**1 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. Refer following sections for more information.*

**2 The Index Registers that AS series CPU modules have is E address.*

**3. Different from the DVP model, the AS model separates the special auxiliary relay and the special data register into two types of devices, SM and SR.*

Refer Instruction Manual in a project for a CPU module in DIADesigner **Help** > **Manual** > **Instruction Manual** in the menu bar as show in the following figure.

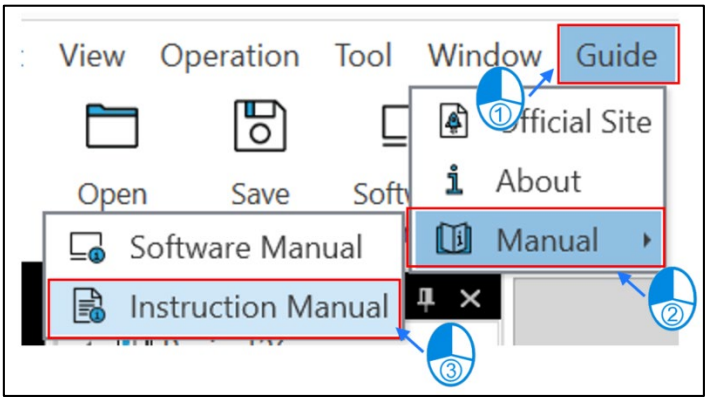


Figure 19 - 22: Instruction Manual

19.2.1.2 Manipulating the Bits in the X/Y/D/L address for AS series CPU modules

The X/Y addresses, which are used as bit, address in AS series CPU modules are represented by X0.0, Y0.0 etc. The AS model uses X0.0 to represent the first bit (Low Significant Bit, LSB) in X0 and X0.15 represents the last bit (Highest Significant Bit, HSB) in X0. The bit-wise structure is shown in the following figure.

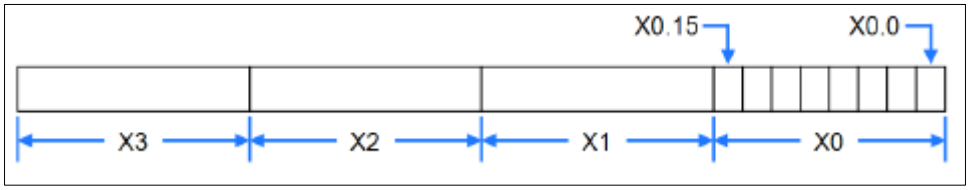


Figure 19 - 23: Manipulating Bits in AS series CPU modules

Example:

In the following scenario, when X0.0 is set, value in D0 is moved to Y1 and the states of Y1.0 ~ Y1.15 will change.

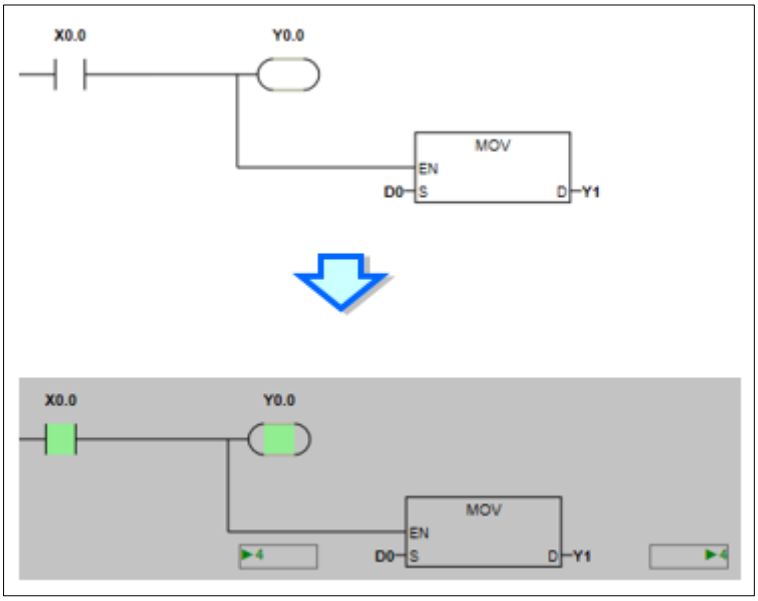


Figure 19 - 24: Example 1 of Bits in AS series CPU

The bits operation can be used except for X/Y address in AS series CPU modules and can be manipulated. In the scenario shown in the following figure, the state of M0 is sent to D0.2. Suppose the initial value in D0 is 0. When M0 is ON, the value of D0 becomes 4.

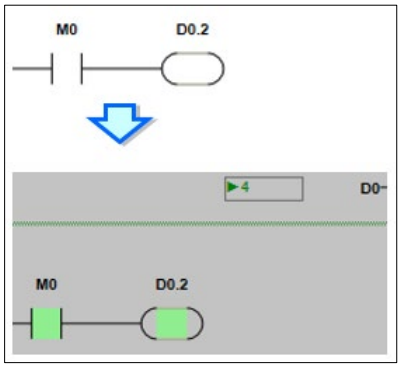


Figure 19 - 25: Example 2 of Bits in AS series CPU

19.2.2 Address resources in AS series CPU modules

19.2.2.1 AS200/AS300 address resources

Address resources in AS series PLC – AS200/AS300 are listed in the following table:

Address 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	

19.3 Servo Setting

User can open ASD-A3/ASD-B3 Servo Setting through clicking **Servo Setting** node or click the context menu of the device in **Network View**. The respective descriptions are as follows:

- Servo Setting node
  Add an ASD-A3/ASD-B3 device, and expand the device, as shown below:

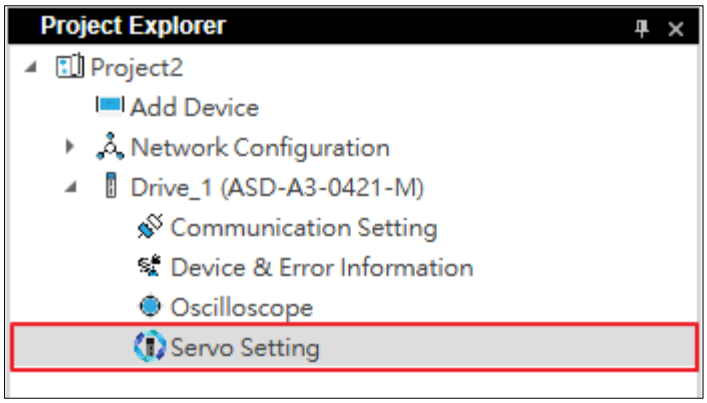


Figure 19 - 26: Servo Setting Node in Project Explorer

Double-click **Servo Setting** to open the Servo Setting window. User can see the device model that is just selected in Window title bar as shown below:

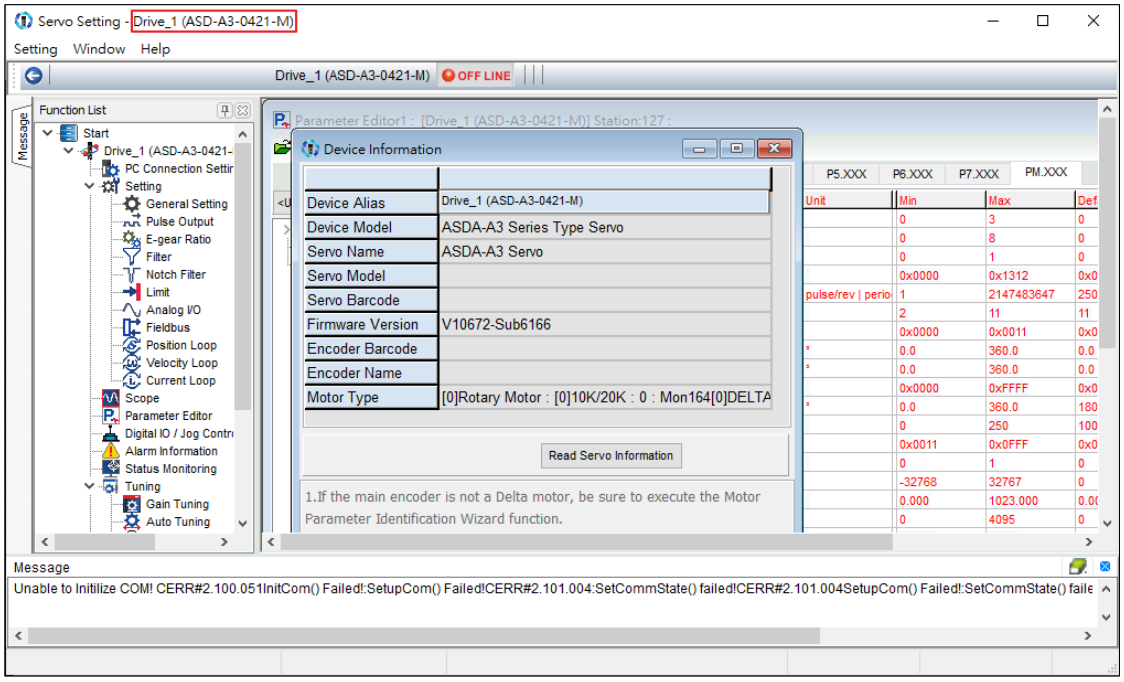


Figure 19 - 27: Servo Setting Node to Open Servo Setting Window

- Context menu of Network View

  After adding an ASD-A3/ASD-B3 device, open Network View. Right-click the device to open the context menu, as shown below:

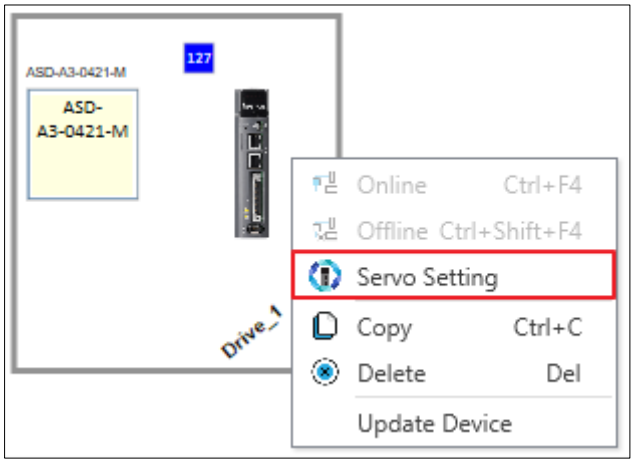


Figure 19 - 28: Context Menu of Network View

Click **Servo Setting** to open the Servo Setting window. User can see the device model that is just selected in Window Title Bar as shown below:

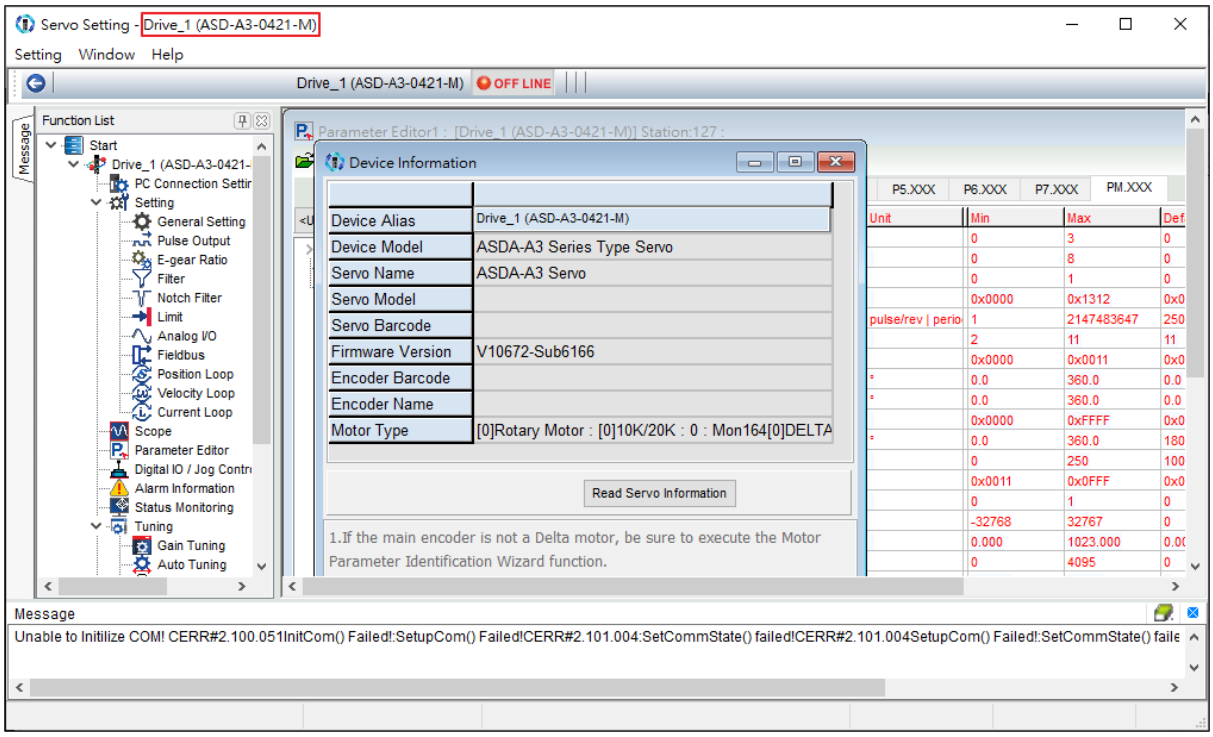


Figure 19 - 29: Context Menu of Network View to Open Servo Setting Window

The Servo Setting windows of different devices can be open at the same time. For more information on Servo Setting window usage, please refer to Servo software manual.

NOTE:

1. *ASD-A3/ASD-B3 devices only support Serial Port connection (Connection through COMMGR is not supported).*
2. *Servo Setting function cannot be used when the built-in oscilloscope is on.*
3. *When Servo Setting window is open, the functions of project editing, device editing, and device upload/download cannot be executed.*
4. *User cannot exit DIADesigner when the Servo Setting window is open.*
5. *If device connection has been established before, user can click* OFF LINE *in Servo Setting to automatically execute the connection.*
6. *If user would like to change the way to establish connection, click* ***PC connection Setting*** *to open the Communication Setting Window.*

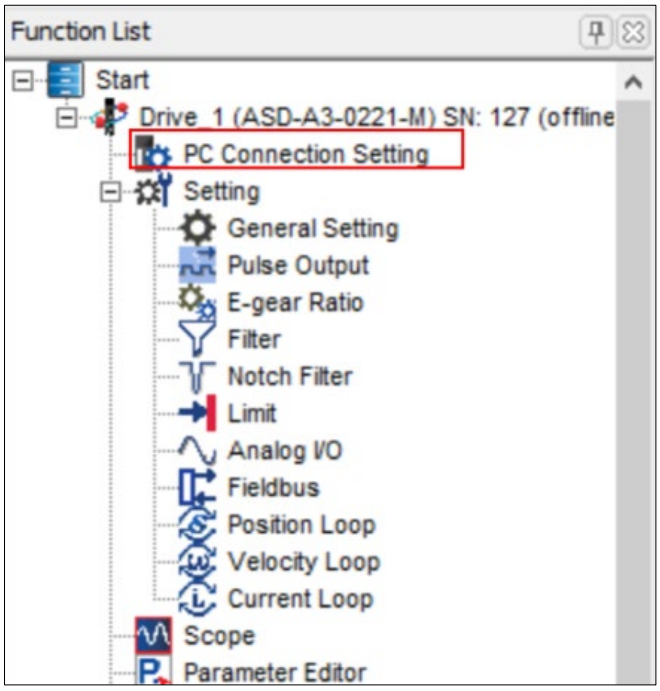


Figure 19 - 30: PC Connection Setting

Smarter. Greener. Together.

Industrial Automation Headquarters

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

Asia

Delta Electronics (Shanghai) Co., Ltd.
No.182 Minyu Rd., Pudong Shanghai, P.R.C.
Post code : 201209
TEL: 86-21-6872-3988 / FAX: 86-21-6872-3996
Customer Service: 400-820-9595

Delta Electronics (Japan), Inc.
Tokyo Office
Industrial Automation Sales Department
2-1-14 Shibadaimon, Minato-ku
Tokyo, Japan 105-0012
TEL: 81-3-5733-1155 / FAX: 81-3-5733-1255

Delta Electronics (Korea), Inc.
Seoul Office
1511, 219, Gasan Digital 1-Ro., Geumcheon-gu,
Seoul, 08501 South Korea
TEL: 82-2-515-5305 / FAX: 82-2-515-5302

Delta Energy Systems (Singapore) Pte Ltd.
4 Kaki Bukit Avenue 1, #05-04, 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

Delta Electronics (Thailand) PCL.
909 Soi 9, Moo 4, Bangpoo Industrial Estate (E.P.Z),
Pattana 1 Rd., T.Phraksa, A.Muang,
Samutprakarn 10280, Thailand
TEL: 66-2709-2800 / FAX : 662-709-2827

Delta Electronics (Australia) Pty Ltd.
Unit 20-21/45 Normanby Rd., Notting Hill Vic 3168, Australia
TEL: 61-3-9543-3720

Americas

Delta Electronics (Americas) Ltd.
Raleigh Office
P.O. Box 12173, 5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3813 / FAX: 1-919-767-3969

Delta Greentech (Brasil) S/A
São Paulo Office
Rua Itapeva, 26 – 3˚ Andar - Bela Vista
CEP: 01332-000 – São Paulo – SP - Brasil
TEL: 55-11-3530-8643 / 55-11-3530-8640

Delta Electronics International Mexico S.A. de C.V.
Mexico Office
Gustavo Baz No. 309 Edificio E PB 103
Colonia La Loma, CP 54060
Tlalnepantla, Estado de México
TEL: 52-55-3603-9200

EMEA

Headquarters: Delta Electronics (Netherlands) B.V.
Sales: Sales.IA.EMEA@deltaww.com
Marketing: Marketing.IA.EMEA@deltaww.com
Technical Support: iatechnicalsupport@deltaww.com
Customer Support: Customer-Support@deltaww.com
Service: Service.IA.emea@deltaww.com
TEL: +31(0)40 800 3900

BENELUX: Delta Electronics (Netherlands) B.V.
De Witbogt 20, 5652 AG Eindhoven, The Netherlands
Mail: Sales.IA.Benelux@deltaww.com
TEL: +31(0)40 800 3900

DACH: Delta Electronics (Netherlands) B.V.
Coesterweg 45, D-59494 Soest, Germany
Mail: Sales.IA.DACH@deltaww.com TEL:
+49(0)2921 987 0

France: Delta Electronics (France) S.A.
ZI du bois Challand 2, 15 rue des Pyrénées,
Lisses, 91090 Evry Cedex, France
Mail: Sales.IA.FR@deltaww.com
TEL: +33(0)1 69 77 82 60

Iberia: Delta Electronics Solutions (Spain) S.L.U
Ctra. De Villaverde a Vallecas, 265 1º Dcha Ed.
Hormigueras – P.I. de Vallecas 28031 Madrid
TEL: +34(0)91 223 74 20

Carrer Llacuna 166, 08018 Barcelona, Spain
Mail: Sales.IA.Iberia@deltaww.com

Italy: Delta Electronics (Italy) S.r.l.
Via Meda 2–22060 Novedrate(CO)
Piazza Grazioli 18 00186 Roma Italy Mail:
Sales.IA.Italy@deltaww.com
TEL: +39 039 8900365

Russia: Delta Energy System LLC
Vereyskaya Plaza II, office 112 Vereyskaya str.
17 121357 Moscow Russia
Mail: Sales.IA.RU@deltaww.com
TEL: +7 495 644 3240

Turkey: Delta Greentech Elektronik San. Ltd. Sti. (Turkey)
Şerifali Mah. Hendem Cad. Kule Sok. No:16-A
34775 Ümraniye – İstanbul
Mail: Sales.IA.Turkey@deltaww.com
TEL: + 90 216 499 9910

GCC: Delta Energy Systems AG (Dubai BR)
P.O. Box 185668, Gate 7, 3rd Floor, Hamarain Centre
Dubai, United Arab Emirates
Mail: Sales.IA.MEA@deltaww.com
TEL: +971(0)4 2690148

Egypt + North Africa: Delta Electronics
Unit 318, 3rd Floor, Trivium Business Complex, North 90 street,
New Cairo, Cairo, Egypt
Mail: Sales.IA.MEA@deltaww.com

DIAS-Manual-0003-EN

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