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

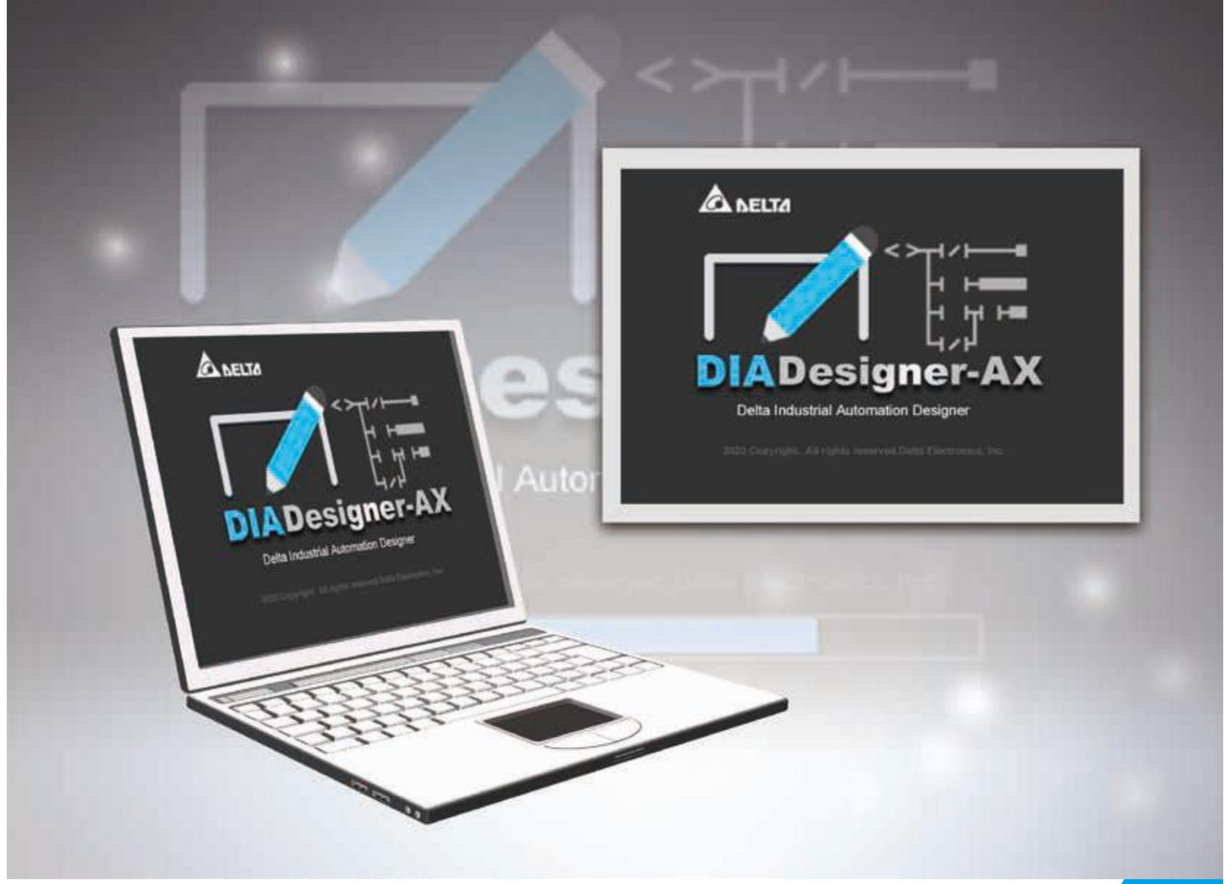
Turkey: Delta Greentech Elektronik San. Ltd. Sti. (Turkey)
Şerifali Mah. Hendem Cad. Kule Sok. No:16-A
34775 Ümraniye – İstanbul
Mail: Sales.IA.Turkey@deltaww.com
TEL: + 90 216 499 9910

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-0014-EN

DIADesigner-AX Software Manual



DIADesigner-AX Software Manual

www.deltaww.com

Smarter. Greener. Together.

Copyright Notice

Limitation of Liability

The contents of this user manual are only for the use of 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 user 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. User shall bear user 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
DIADesigner User Manual	DIAS-Manual-0003-EN
DIAScreen User Manual	DIAS-Manual-0004-EN
Software Download Manual	DIAS-Manual-0005-EN
DIADesigner-AX Online Help	N/A

Revision history

Version	Issue Date	Revision
1st	12/08/2020	The first version was published.
2nd	02/19/2021	1. Section 3.1: Added Support Controller AX-308EA0MA1P, AX-364ELA0MA1T, AX- 300NA0PA1, AX- 324NA0PA1P 2. Section 4.1.1: Added how to Add RIO Module Process. 3. Section 4.1.2: Added Models that Support the firmware update function. 4. Section 4.2.2: Updated the running clock configuration tab to the system settings tab and add the device IP address setting field. 5. Section 4.2.5: Updated Ethernet General Tab. 6. Section 4.2.7: Added high-speed IO Settings,divided into AX-364EL/AX-308 AND AX-324. 7. Section 4.4.4: Added Introduction to the new free encoder. 8. Section 7.1: Added Introduction to the new device storage library.
3rd	07/30/2021	1. Section 3.1: Added support controller AX-304ELA0PA1T, AX-304ELA0PA1P, AX-316EA0MA1T. 2. Section 3.3.1.1: Added update device function description. 3. Section 3.3.1.2: Added the description of editing I/O mapping function. 4. Section 4.1.1.1: Added power module description. 5. Section 4.1.2: Updated the firmware update instructions. 6. Section 4.2.3: Added description of new device diagnosis function. 7. Section 4.2.4.1: Added sending delay time setting. 8. Section 4.2.5.3: Added description of parameter backup and restore functions. 9. Section 4.2.8.1.4: Added Po axis pin selection. 10. Section 4.2.8 and 4.4: Supported floating points for mechanism parameters. 11. Section 4.4.1: Added New axis parameter Dynamic Limits for AX-8xxEP0 series synchronous axis 12. Section 5.3.1.1: Added multiple EtherCAT instructions in AX-8xxEP0 series. 13. Section 7.1: Added device storage library description.
4th	05/31/2022/	1. Section 3.1.7.1: Added Notes on setting array variables. 2. Section 3.5: Added Variable Export and Import. 3. Section 4.1.1.1.1: Added Setting the AS02LC Wizard. 4. Section 4.1.2: Updated Firmware Update Instructions. 5. Updated Section 4.2.2: Added IP address setting support for AX-8 series. 6. Updated Section 4.2.8.1: Added AX-332. 7. Section 4.4.1.3: Added CoE parameters. 8. Revised Section 4.4.2: Added speed axis, revised origin return settings and added modes -1~-4. 9. Chapter 8:Added Password Management and Data Protection Mechanism. 10. Section 8.1: Added Project ID/PLC ID. 11. Section 8.11: Added Permission Settings.

Version	Issue Date	Revision
5th	10/18/2022	1. Updated Section 3.1: Updated List of supported devices. 2. Section 3.6: Added Auto-completion. 3. Updated Section 4.3.1: Added support for AX-8xxEP0 Linux series devices, display IP address and MAC address, and added IP settings to take effect immediately after downloading. 4. Updated Section 4.3.5.1: EtherNet/IP Scanner Operation 5. Updated Section 4.3.5.3: EtherNet/IP Local Adapter Procedure Update
6th	03/31/2023	1. Chapter 2: Overall revised installation and uninstallation steps and system requirements. 2. Section 3.1: Added MH300 series and CH2000 series to supported devices. 3. Section 3.3.1.2: Revised adding a device. 4. Section 3.3.1.3: Revised communication settings and added the ability to show Mac-Address and IP-Address. 5. Section 3.3.1.6: Revised declaring variables. 6. Section 3.4.1: Updated updating device. 7. Section 3.4.1.1: Added the update device list dialog. 8. Chapter 4: Added Programming Reference chapter 9. Section 4.1.1-4.1.3: Added creating array and structure variables and setting up comments 10. Section 4.1.4: Added seting up retain persist variables. 11. Section 4.1.5: move variable export and import to this section. 12. Section 5.1.1.1: Revised adding modules and updated figures to add Network modules. 13. Section 5.1.1.2: Revised adding remote I/O modules. 14. Section 5.1.1.3: Revised adding power module. 15. Section 5.1.1.4: Added function cards. 16. Section 5.1.1.4.1: Added modbus channel configuration page. 17. Section 5.2.2: Revised system settings and added retain settings. 18. Section 5.3.1 Added system parameters for AX-8 series controller. 19. Section 5.3.2 Added system settings for AX-8 series controller. 20. Section 5.4.6 Added SoftMotion version compatibility. 21. Section 6.3.6: Added supporting PROFINET version 4.3.0.0 22. Section 7.1: Updated the overview. 23. Section 7.2: Updated exporting the symbol configuration file. 24. Section 8.1 Revised device repository 25. Section 8.1.1: Revised installing device description file. 26. Section 8.4 Added startup command. 27. Section 8.5: Added resource monitor. 28. Section 8.6: Added tracing variables. 29. Section 8.7: Added software diagnosis tool. 30. Section 9.11.1: Added import management files. 31. Section 9.11.2: Added export management files. 32. Section 9.11.3: Revised adding user and group settings. 33. Section 9.11.4: Revised setting up access right.

TOC

Chapter 1: DIADesigner-AX overview

1.1 Introduction

DIADesigner-AX is the IEC 61131-3 programming tool for Delta's new generation motion controller – AX series and adopts a large number of applied instructions, especially Motion library. The multilingual environment and the user-friendly interface provide a convenient and efficient development environment.

1.2 DIADesigner-AX overview

DIADesigne-AX provides:

- All editors of the IEC 61131-3 (FBD, LD, ST, SFC) and different variants of the standard editors.
- Powerful and proven library concept for the reuse of application.
- Project configuration through wizards.
- Input assistance for the input and configuration of data.
- User-friendly programming with mouse and keyboard in all IEC 61131-3 editors.
- Extensive debugging and online features for the fast optimization of the application code and to speed up testing and commissioning.
- Numerous security features for the protection of the source code and for safeguarding the operation of the controller.
- Programmable devices from different manufacturers.
- The user interface is extendible and adaptable without leaving the framework.
- Transparent internal structures of the development tool and the available components.
- Many seamlessly integrated tools for different kinds of automation tasks.
- Two built-in configurations:

  Hardware Configuration: It is used to configure hardware for a system, and manage parameters.

  Network Configuration: It is used to configure networks for a PLC system, and manage data exchanges.

DIADesigner-AX provides various solutions for motion control including PLCopen MC function block, G-code editor, E-CAM editor, positioning planning chart tool and many more.

- Support PLCopen POUs for single and multi-axis movements

- Support PLCopen POUs for add-on functions like diagnostics, stop, CAM controller
- Additional POUs for different tasks like monitoring dynamic data or following error, operating CAMs and CAM controllers
- Integrated graphical CAM editor with extensive configuration options
- Virtual and logical axes are supported.
- Integrated drivers for numerous, CANopen and EtherCAT drives.
- Configuration of the Drives/Servo based on standard field devices.

Chapter 2: Installation and uninstallation

2.1 System requirements

The following table lists the system requirements to run DIADesigner-AX.

Item	System Requirements
Operating System	Windows 10/11 (64-bit)
CPU	Intel Celeron 540 1.8 GHz (min.) Intel Core i5 M520 2.4 GHz (min.)
Memory	4 GB or above
Hard Disk Drive	10 GB or above
Monitor	Resolution: 1920 x 1080 Pixels recommended
Keyboard/Mouse	General keyboard mouse or Windows-compatible device
PC interface	EtherNet, USB, Serial port (depends on product interface)
Software	Need to install Microsoft .Net Framework 4.6.2

2.2 Install, uninstall, and update DIADesigner-AX

This chapter introduces how to use DIAInstaller to install, uninstall, and update DIADesigner-AX.

2.2.1 Install DIADesigner-AX

DIAInstaller is used to easily download, install, uninstall, and update all DIAStudio software. You can download DIAInstaller from diastudio.deltaww.com. For more information about DIAInstaller, see *DIAInstaller User Manual*.

Prerequisites

- DIAInstaller is installed.

To install DIADesigner-AX

1. Open DIAInstaller.
2. Click at the upper-right corner to sign in.
3. Go to **All Apps** and find DIADesigner-AX.
4. In **Available Versions**, select a version.
5. Click and then select **Install.**
6. In the **DIADesigner-AX** dialog, follow the on-screen instructions to complete the installation.

Note:

1. You can click **Updates** to check if there are any new versions or patches.
2. You can configure the settings of DIAInstaller by clicking .

2.2.2 Uninstall DIADesigner-AX

Follow the steps to uninstall DIADesigner-AX in DIAInstaller.

To uninstall DIADesigner-AX

1. In DIAInstaller, go to **Installed**, and select **Uninstall**.
2. In the **DIADesigner-AX** dialog, follow the on-screen instructions to complete the uninstallation.

2.2.3 Update DIADesigner-AX

Follow the steps to update DIADesigner-AX.

To update DIADesigner-AX

1. In DIAInstaller, go to **Updates**.
2. Find DIADesigner-AX.
3. In **Available Versions**, select a version.
4. Do one of the following:
    - Select **Update** and then follow the on-screen instructions to complete the update.
    - Click  and select **Download** to download the installation file to the local path, and then select **Show in folder** to find the .exe file to install.

Chapter 3: Getting started with DIADesigner-AX

3.1 Supported devices

DIADesigner-AX v1.4.0 supports the following devices. Refer to the device documentation for more information on the devices.

Type	Product Series
Controller	• AX-8xxEP0 Linux series • AX-8xxEP0 Windows series • Motion PLC: AX-308EA0MA1T, AX-308EA0MA1P, AX-364ELA0MA1T, AX-304ELA0PA1T, AX-304ELA0PA1P, AX-316EA0MA1T • Logic PLC: AX-300NA0PA1, AX-324NA0PA1P
AC Motor Drive	C2000 plus series, MS300 series, MH300 series, CH2000 series
AC Servo Drive	ASDA-A2-E series, ASDA-B3-E series, ASDA-A3-E series
Remote I/O	• EtherCAT: R1-EC series, R2-EC series, RTU-ECAT series. • EtherNet/IP: AS 200 series, AS 300 series.

3.2 Device and function guide

The devices and functions supported by DIADesigner-AX v1.4. 0 are shown in the following table. Subsequent versions will gradually increase device support and features.

<table>
<tr><th></th><th>Communication Setup</th><th>Network Configuration</th><th>Hardware Configuration</th><th>Parameter Setup</th><th>Program Edit</th></tr>
<tr><td>Controller</td><td rowspan="3">Sec 6.3</td><td rowspan="3">Ch.6</td><td rowspan="3">Ch.5</td><td>Ch.4</td><td>Ch. 3.3.1.7</td></tr>
<tr><td>AC Motor Drive</td><td>Sec 3.3.1.2</td><td rowspan="2">N/A</td></tr>
<tr><td>AC Servo Drive</td><td>N/A</td></tr>
</table>

<table>
<tr><th></th><th>Download</th></tr>
<tr><td>Controller</td><td rowspan="3">CODESYS Development System > Updating an Application on the PLC > Execution of a Download
Link:
https://help.CODESYS.com/webapp/_cds_performing_a_download;product=CODESYS;version=3.5.10.0</td></tr>
<tr><td>AC Drive</td></tr>
<tr><td>Servo Drive</td></tr>
</table>

3.3 Quick start

3.3.1 Example settings

In this example, user will program a simple refrigerator controller. The completed project **RefrigeratorControl.project_archive** can be found in the DIADesigner-AX installation directory in the Projects sub-directory. In addition to the sample project, user will create one here step-by-step.

- As with a conventional refrigerator, the temperature is specified by the user via rotary control.
- The refrigerator determines the actual temperature using a sensor. When it is too high, the refrigerator starts the compressor with an adjustable delay.
- When the desired temperature is reached, the compressor will start to cool down minus one degree of hysteresis. The hysteresis here is to prevent the actual temperature from fluctuating too much within the range of the set temperature, causing the compressor to be turned on and off continuously.
- When the door is open, a lamp lights up inside the refrigerator. When the door is open too long, a beeping acoustic signal sounds.
- If the compressor does not reach the set temperature despite activity of the motor over a long period of time, then the buzzer emits a steady acoustic signal.

Engineering:

The cooling function is controlled by the main program in the application; the signal management is controlled by another POU. There are required function blocks in the standard library. There is no physical temperature sensor and physical actuator in the example, so a program needs to be written to simulate heating/cooling. In this way, the operation of the refrigerator controller can be monitored in online mode.

3.3.1.1 Create a project

After the project is created, it is stored in a .project file in the folder you specified.

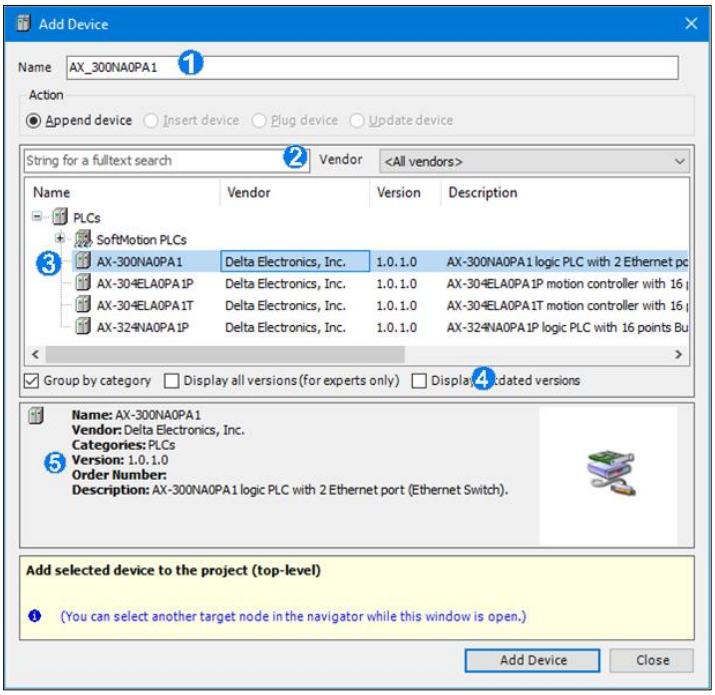
To create a project

1. Do one of the following:
   - On **Start Page**, click **New Project**.
   - On the quick access toolbar, click  .
   - Select **File** > **New Project**.
2. In the **New Project** dialog, select **Standard Project as a template,** specify a name and a storage location for the project, and then click **OK**.
3. In the **Standard Project** dialog, select the controller, DDF version, and programming language, and then click **OK**.

3.3.1.2 Add a device

DIADesigner-AX supports controllers, servo drives, and AC drives.

The **Add Device** dialog consists of the following information.

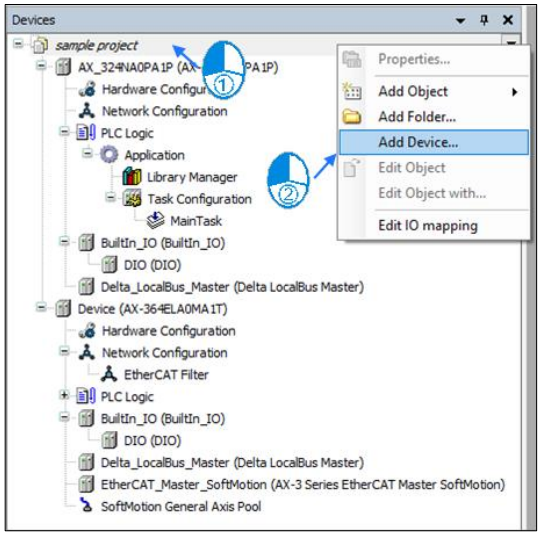


Legend	Description
1	Enter the device name and select to either append a device, insert a device, plug a device, or update a device.
2	Search based on the device name and vendor.
3	Display the available device with its name, vendor, version, and description.
4	Select to group the devices by category, to display either all versions or only display outdated versions.
5	Display the details of the selected device.

If you want to learn more about appending a device and inserting a device and other detailed information, see CODESYS Development System > Reference, User Interface > Menu Commands > Menu 'Project' > Command 'Insert Device'.

To add a device at the project level

1. Create a **project** (See 3.3.1.1 Create a project).
2. On the **Devices** pane, right-click the project name, and then select **Add Device**.



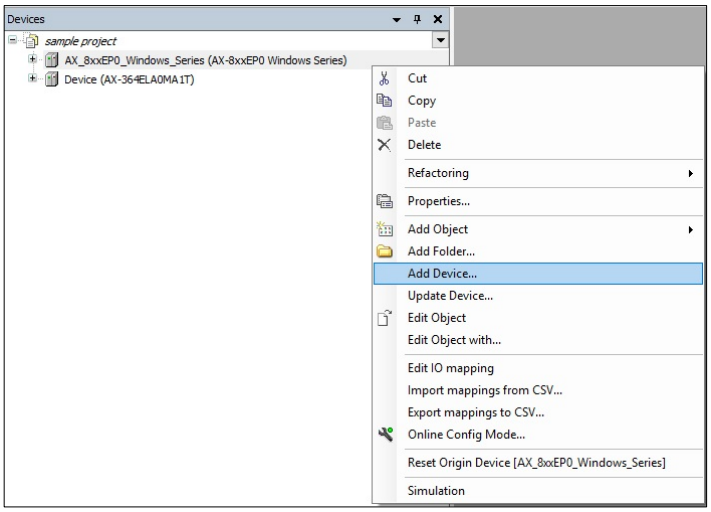
3. In the **Add Device** dialog, select the device, enter **Name**, and then click **Add Device**.

   Only available devices are listed.

   Only the latest version of the device is listed by default. If you want to select the earlier versions device, select the **Display all version (for experts only)** checkbox.

To add a device to the project at the device level

1. Create a project.
2. On the **Devices** pane, right-click the device that you want to add the device to, and then select **Add Device**.



3. In the **Add Device** dialog, select the device, enter **Name**, and then click **Add Device**.

   Only available devices are listed.

   Only the latest version of the device is listed by default. If you want to select the earlier versions device, select the **Display all version (for experts only)** checkbox.

3.3.1.3 Communication settings

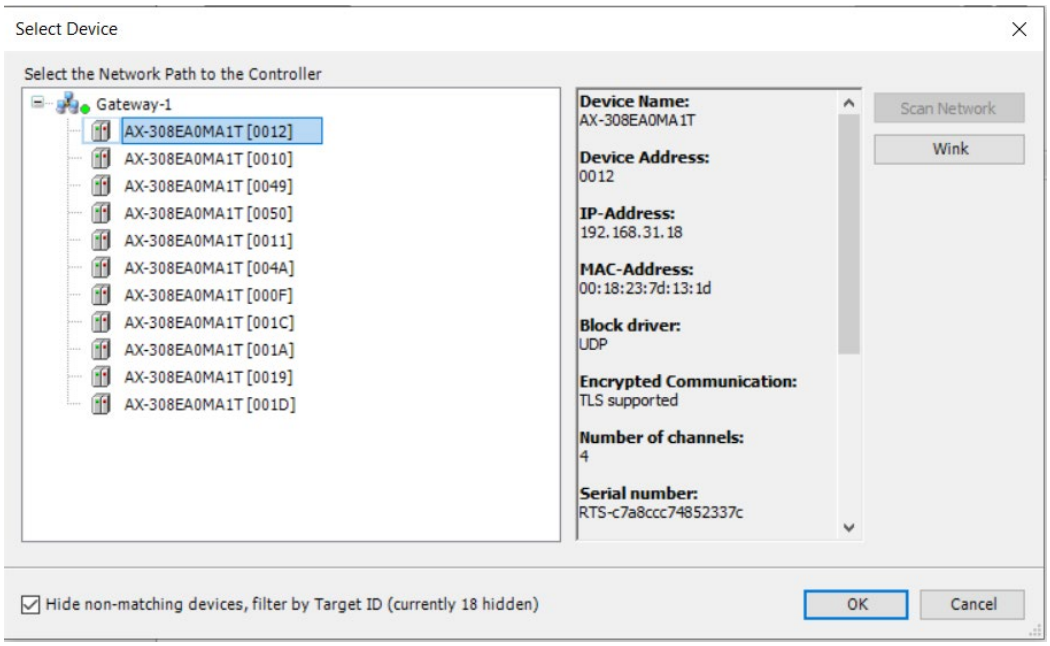
You can use **Communication Settings** to configure and establish communications with the controller.

Prerequisites

- Set up the local gateway.

To establish communications with the controller

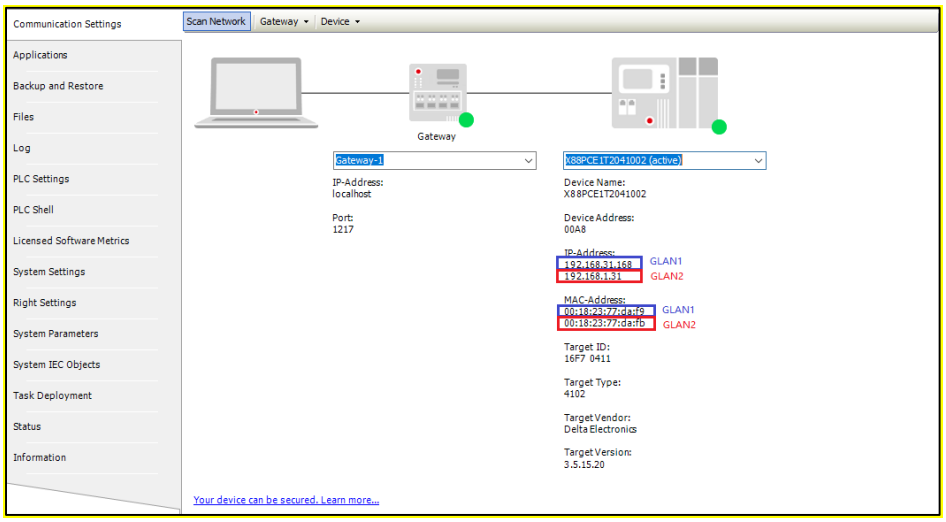
1. On the **Devices** pane, double-click the controller.
2. Go to **Communication Settings**.
3. Click **Scan Network**.
4. In the **Select Device** dialog, select your controller.



Note:

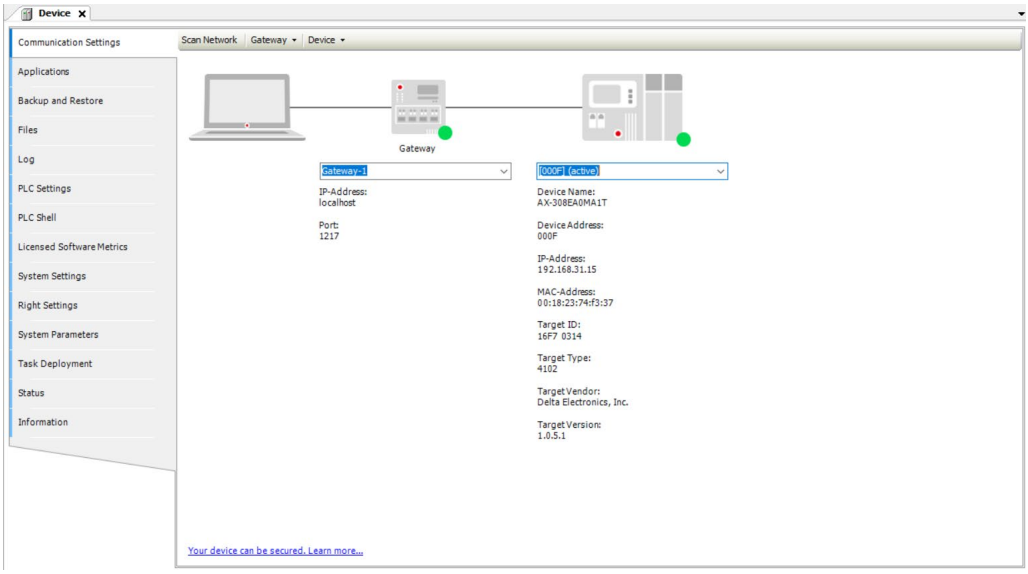
- The IP-Address and MAC-Address are N/A if you specify the username and password in the controller.
- If you select an AX-8xxEP0 Linux series controller, the IP-Address and MAC-Address will show if the controller firmware version is 1.0.4.4 or later.
- If you select an AX-8xxEP0 Windows series controller, the IP-Address and MAC-Address will show if the controller firmware version is 1.0.4.12 or later.

There are two EtherNet ports on both AX-8xxEP0 Linux and AX-8xxEP0 Windows series controllers. The first IP-Address and MAC-Address are for GLAN1 and the second IP-Address and MAC-Address are for GLAN2.



- If you select an AX-3 series controller, the MAC-Address will show if the controller version is 1.0.5.0 or later.

5. Click **OK**. The device is active and the information shows on the **Communication Settings** pane.



6. Go to **System Settings**.
7. In the **Network** area, click **Read from PLC** to synchronize the IP address from the controller.

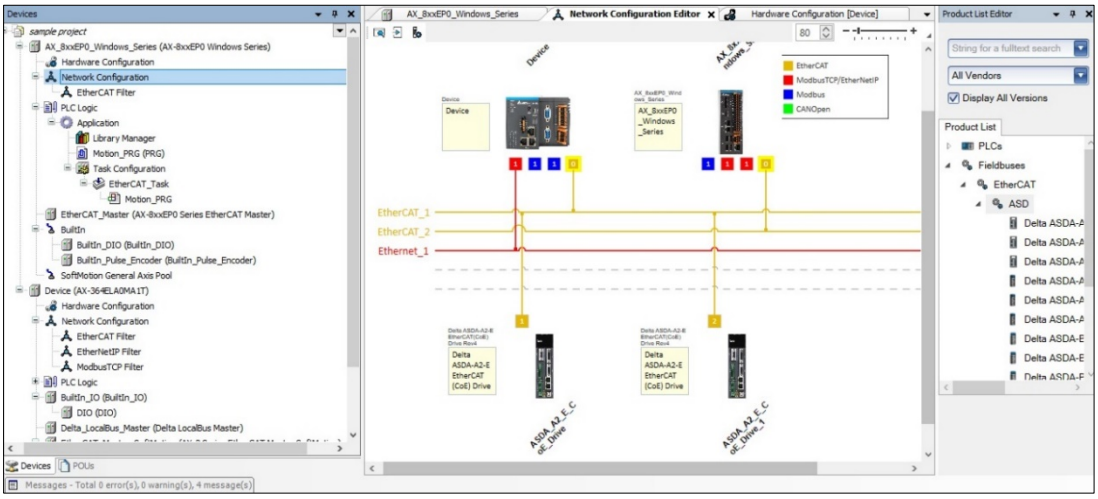
Note:

- If you select an AX-8xxEP0 Linux series controller, the **Read from PLC** function will be available if the controller DDF version is 3.5.15.21 or later and the controller firmware version is 1.0.4.4 or later.
- If you select an AX-8xxEP0 Windows series controller, the **Read from PLC** function will be available if the controller DDF version is 3.5.14.17 or later and the controller firmware version is 1.0.4.12 or later.

3.3.1.4 Network configuration

The **Network Configuration Editor** shows the industrial network topology of all Delta Industrial Automation products. Users can configure and plan the network here.

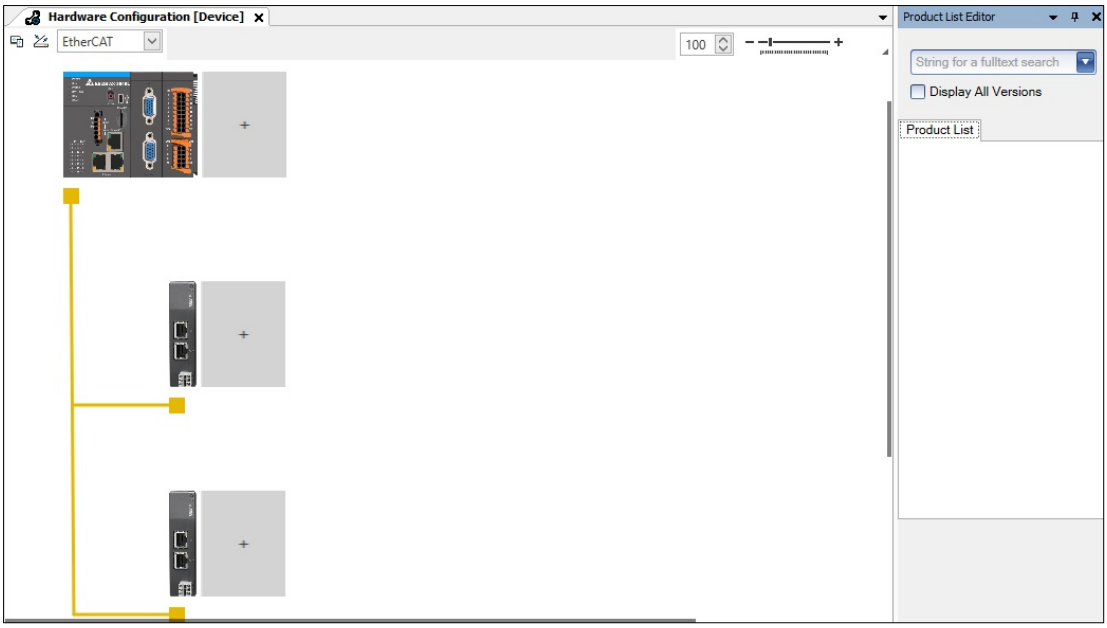
It uses an icon method to display the connections of devices and devices,allowing users to simply plan their networks.



3.3.1.5 Hardware configuration

In **Hardware Configuration**, user can perform functions such as configuration of modules, manage RIO, parameter settings for modules, I/O Scan with the help of Hardware Configuration.

A sample Hardware Configuration is shown in the following figure.



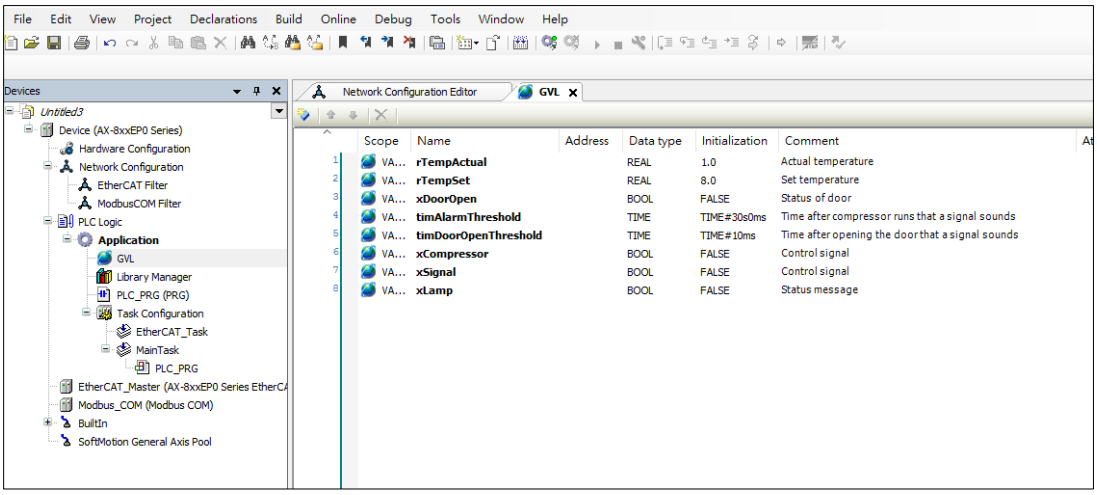
3.3.1.6 Declare variables

Before creating a program, you need to declare the variables for the application by creating a global variable list under **Application**.

To create a global variable list

1. Right-click **Application** and select **Add object** > **Global Variable List**.
2. In the **Add Global Variable List** dialog, enter the name for the list. The default name is GVL.
3. Click  to switch to the tabular view.
4. Click  to insert a new variable.
5. In the **Name** box, enter **rTempActual**.
6. Double-click the **Data type** box and click  , and then select **Input Assistant**.
7. In the **Input Assistant** dialog, on the **Categories** tab, select **Standard Types** > **Real**, and then click **OK**.
8. In the **Initialization** box, enter a number value, for example, 8.0.
9. Declare the following variables in the same way:

Name	Data Type	Initialization	Description
rTempActual	REAL	1.0	Actual temperature.
rTempSet	REAL	8.0	Set temperature.
xDoorOpen	BOOL	FALSE	Status of door.
timAlarmThreshold	TIME	TIME#30S	Time after compressor runs that a signal sounds.
timDoorOpenThreshold	TIME	TIME#10S	Time after opening the door. that a signal sounds.
xCompressor	BOOL	FALSE	Control signal.
xSignal	BOOL	FALSE	Control signal.
xLamp	BOOL	FALSE	Status message.


File Edit View Project Declarations Build Online Debug Tools Window Help
Devices
Untitled3
Device (AX-8xxEP0 Series)
Hardware Configuration
Network Configuration
EtherCAT Filter
ModbusCOM Filter
PLC Logic
Application
GVL
Library Manager
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task
MainTask
PLC_PRG
EtherCAT_Master (AX-8xxEP0 Series EtherCA
Modbus_COM (Modbus COM)
BuiltIn
SoftMotion General Axis Pool
Network Configuration Editor
GVL
Scope Name Address Data type Initialization Comment
1 VA... rTempActual REAL 1.0 Actual temperature
2 VA... rTempSet REAL 8.0 Set temperature
3 VA... xDoorOpen BOOL FALSE Status of door
4 VA... timAlarmThreshold TIME TIME#30s0ms Time after compressor runs that a signal sounds
5 VA... timDoorOpenThreshold TIME TIME#10ms Time after opening the door that a signal sounds
6 VA... xCompressor BOOL FALSE Control signal
7 VA... xSignal BOOL FALSE Control signal
8 VA... xLamp BOOL FALSE Status message

3.3.1.7 Programming

Creating the main program for the cooling control in the CFC editor

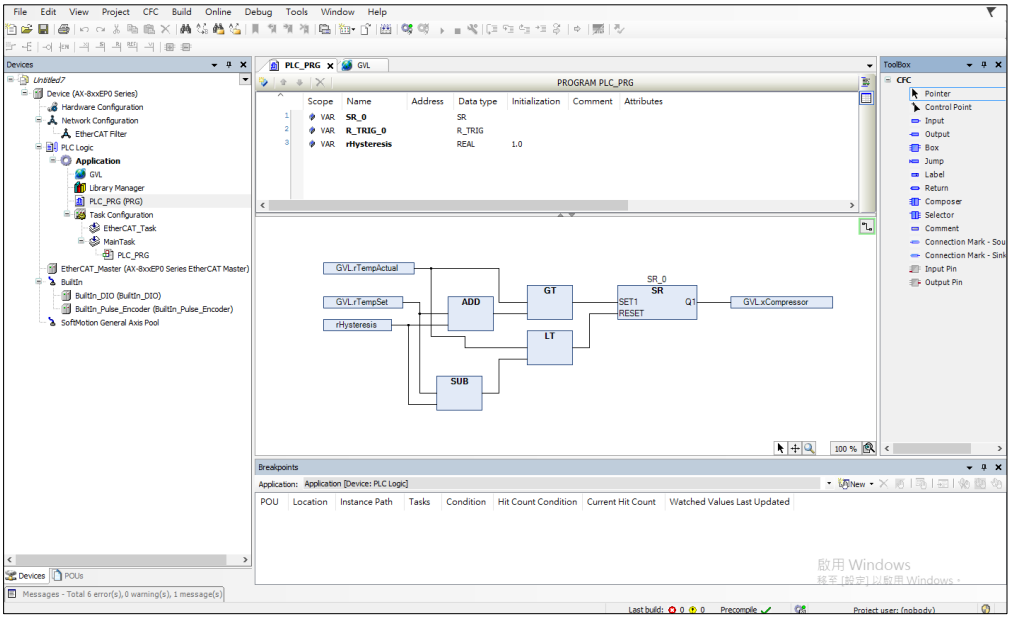
User can describe the main function of the application program in the main program block PLC_PRG, which is created by default. The compressor is activated and cools if the actual temperature is higher than the temperature set point plus a hysteresis. The compressor is switched off if the actual temperature is lower than the set point minus the hysteresis.

Note: Create a new project with Standard template.

To describe the functionality in the CFC implementation language, follow these steps:

1. Double-click **PLC_PRG** in the device tree.

   Result: The CFC editor opens. At the top of the editor window, the declared variable area will be displayed in text or table format. The **Toolbox** tab is displayed on the right side of the window.



2. In the **Toolbox** window on the right, select the input component and drag it to the CFC editing area.

   Result: Added the unnamed ??? component.

3. In the CFC editing area, click the ??? component, and then click ... to open

the **Input Assistant**. In the variable category, please select the variable **rTempActual** from **Application** > **GVL**.

Result: The input name is GVL.rTempActual.

4. As in Step 3, create another input with the name of the global variable GVL.rTempSet.
5. Create another input, and then click ??? in addition, replace them with the name rHysteresis.

   Result: Because this is not the name of a known variable, the **Declare Variable** dialog opens. The name is already used in the dialog.
6. In the **Declare Variable** dialog, specify **Data Type** with REAL and **Initialization value** with 1. Click **OK.**

   Result: The variable rHysteresis appears in the declaration editor.
7. In the **ToolBox** view, select **Box** element and drag it to a point in the CFC editor.

   Result: The POU opens in the CFC Editor.
8. Replace ??? with ADD.

   Result: The POU adds all inputs that are connected to it.
9. Connect the **GVL.rTempSet** input pin to the **ADD** operation block.
10. As the above steps, connect the **rHysteresis** input pin to the ADD operation block.

    Result: **rHysteresis** and **GVL.rTempSet** become the input of ADD operation block
11. If user want to move a component in the editor, click the blank area in the component or click the outer frame to select the component (red box) and drag it to the position user want.
12. In order to compare the sum of **GVL.rTempActual** and **GVL.rTempSet and rHysteresis**, create another **GT** operation block on the right side of the ADD operation block.

    Result: The operation of the **GT** operation block is as follows:

“IF (upper input > lower input) THEN output := TRUE;”

13. **GVL.rTempActual** input is connected to the upper pin of the **GT** calculation block.
14. ADD operation block output pin is connected to the lower pin of **GT** operation block.

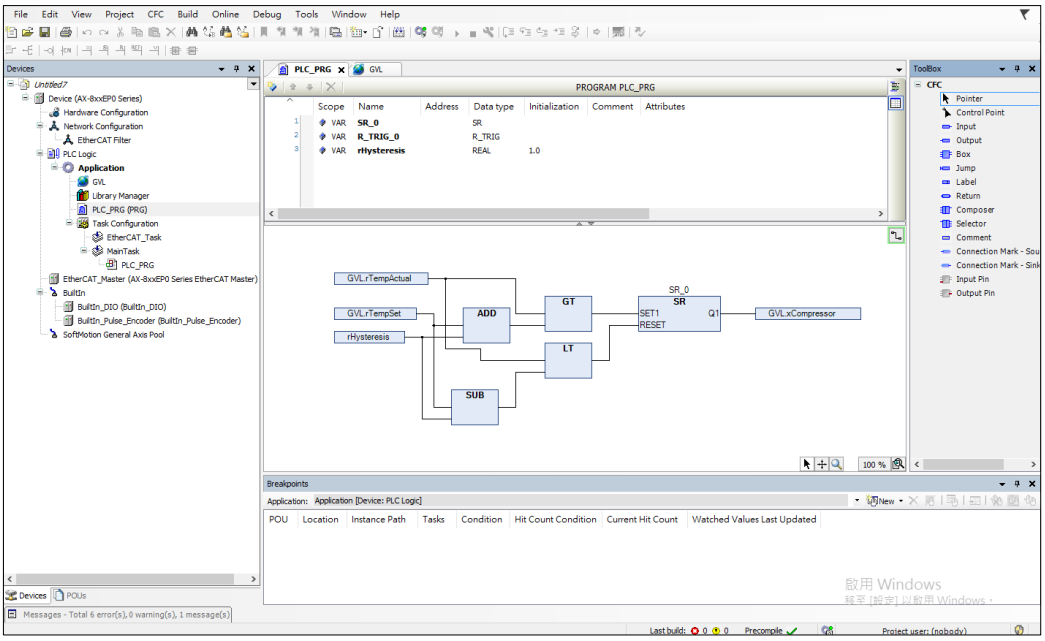
    In this step, create an arithmetic block on the right side of the **GT** arithmetic block, which will be based on the input conditions. (Set-Reset) Start/stop the cooling compressor. Press Enter in the **???** field of the operation block.

    Result: Open the declare variable dialog.
15. Declare that the variable name is **SR_0** and the data type is **SR**. Click **OK**.

    Result: The SR function block from the standard library is instantiated. (Input) **SET1** and **RESET** appear.
16. Connect the output pin on the right side of the **GT** operation block to the **SET1** input of the **SR_0** function block.

    Result: **SR** can set the Bool variable from **FALSE** to **TRUE**, and vice versa. When the input **SET1** condition is met, the Boolean variable will be set to **TRUE**. When the **RESET** condition is met, the variable will be reset. Here is an example of setting **GVL.xCompressor** as a Bool (global) variable.
17. Create an output component and assign it to the global variable **GVL.xCompressor**. Pull out the connecting wire between **GVL.xCompressor** and **SR** output pin **Q1**.

Now specify a condition under which the compressor should be turned off again (under this condition, the RESET input of the SR function block will get a TRUE signal). To do this, please make an opposite condition; use SUB (subtraction) and LT (less than) operation blocks.



Creating a POU for signal management in the ladder diagram editor

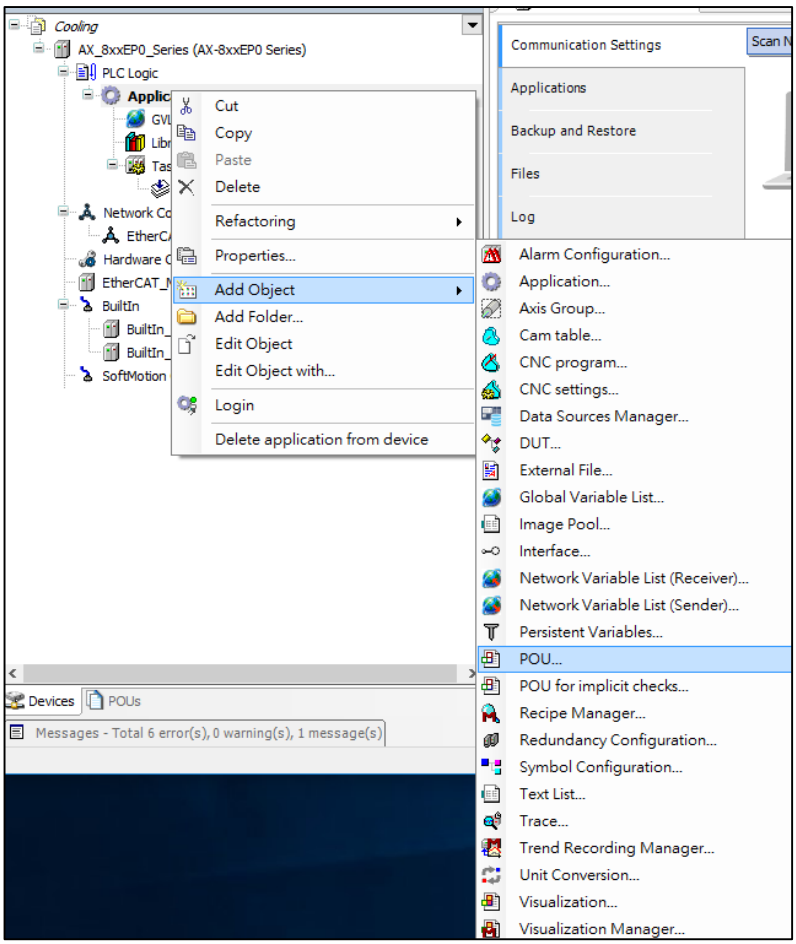
In another POU, user will now implement the signal management for the alarm buzzer and for switching the lamp on and off. The ladder diagram (LD) implementation language is suitable for this.

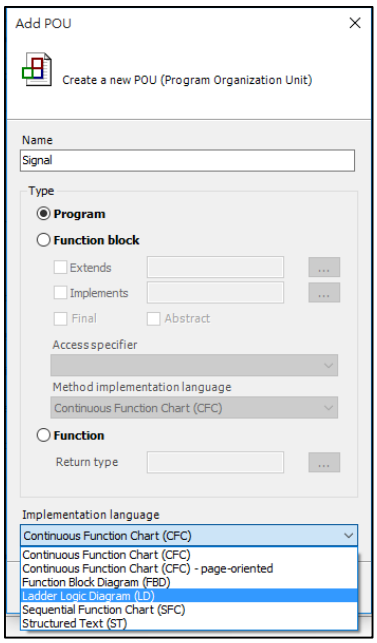
Handle each of the following signals in their own networks:

- When the compressor runs too long because the temperature is too high, a continuous acoustic signal sounds.
- When the door is open too long, an intermittent signal sounds.
- The light is on as long as the door is open.

-

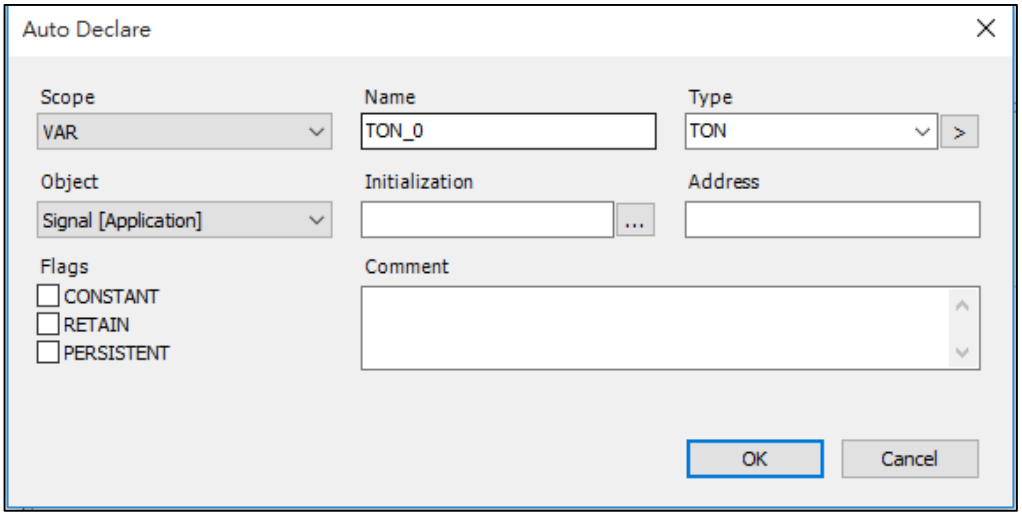
1. Below **Application** in the device tree, create a POU object of type **Program** using the **Ladder Diagram (LD)** implementation language. Specify the name Signals for the program.





Result: Signals is listed in the device tree next to PLC_PRG**.** Double-click the ladder diagram, editor opens in the Signals tab. The declaration editor appears in the upper part, the **ToolBox** view to the right. The LD contains an empty network.

2. Add a new network to the **POU**: "When the compressor runs for too long and does not reach the target set temperature, an acoustic signal will be issued." To do this, insert a **TON** timer function block. After a specified period of time, it will switch the Boolean signal to **TRUE**. Select **TON** under the function block in the toolbox, and then drag it to the empty network and the start from here box is displayed. Release the mouse button when the arrow turns green.
3. Press Enter: User have confirmed the instance name. The **Declare Variable** dialog opens (**Auto Declare**).



4. Click **OK** to confirm the dialog.

   Result: Now the inserted POU TON is instantiated with the name TON_0. Moreover, the top input is displayed by default as a contact preceding the POU.

   Note: To read the help for the function block TON, place the cursor in/on the POU in the “TON” string and press F1.

5. Now user program so that the function block is activated as soon as the cooling compressor starts to run. For this purpose, name the contact at the upper input of the POU GVL.xCompressor. User have already declared this Boolean variable in the GVL Glob_Var.

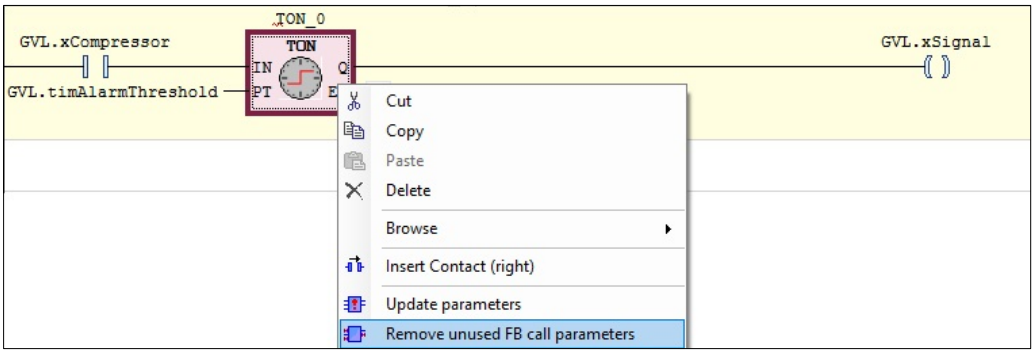
   Note: When user begin to type a variable name at the input position, user automatically get a list of all variables with names that begin with the typed characters and can be used at this point. This assistance is a default setting in the DIADesigner-AX options for smart coding.

6. Insert the signal that is to be activated. To do this, drag a *Coil* from the **Ladder Diagram Elements** toolbox category to the output Q of TON POU. Specify the name GVL.xSignal for the coil.

7. Define the time from the activation of the POU TON_0 until the signal should sound. This definition takes place via the variable GVL.timAlarmThreshold, which user insert for this purpose at the input PT of TON_0. To do this, click the fine-edged box to right of the input pin and enter the variable name.

8. Select the POU TON and click **Remove unused FB call parameters** in the context menu.

   Result: The unused output ET is removed.



9. In the second network of the LD, program so that the signal sounds intermittently when the door is open too long. In the editor window, click below the first network. In the context menu, click **Insert network**.

   Result: An empty network with the number *2* appears.

10. As in the first network, implement a POU TON for time-controlled activation of the signal. This time it is triggered by the global variable GVL.xDoorOpen at the input IN. At the input PT, add the global variable GVL.timDoorOpenThreshold.

11. In addition, from the library Unit, add a POU BLINK at the output Q of the POU TON in this network and instantiate it as Blink_0.

12. The POU BLINK_0 clocks the signal forwarding Q and therefore GVL.xSignal.

    First, drag two **Contact** elements from the **ToolBox** view to the OUT output of the POU. Assign the variable TON_1.Q to the contact directly after the output Q and the global variable GVL.xDoorOpen to the second contact.

13. Insert a **Coil** element after the two contacts and assign the global variable GVL.xSignal to it.

14. To do this, declare the local variable timSignalTime : TIME := T#1S; and insert this variable at the inputs TIMELOW and TIMEHIGH. The cycle time is 1 second for TRUE and 1 second for FALSE.

15. Select the POU TON and click **Remove unused FB call parameters** in the context menu.

Result: The unused output ET was removed.

16. In the third network of the LD, program so that the lamp lights up as long as the door is open. To do this, insert another network. In that network on the left, insert a contact GlobVar.xDoorOpen, which leads directly to an inserted coil GVL.xLamp.

17. DIADesigner-AX processes the networks of an LD in succession. Now install a jump to Network 3 at the end of Network 1 in order to ensure that either only Network 1 or only Network 2 is executed:

    Select Network 3 by clicking in the network or in the field with the network number. In the context menu, click **Insert Label**. Replace the text Label: in the upper left part of the network with DoorIsOpen:

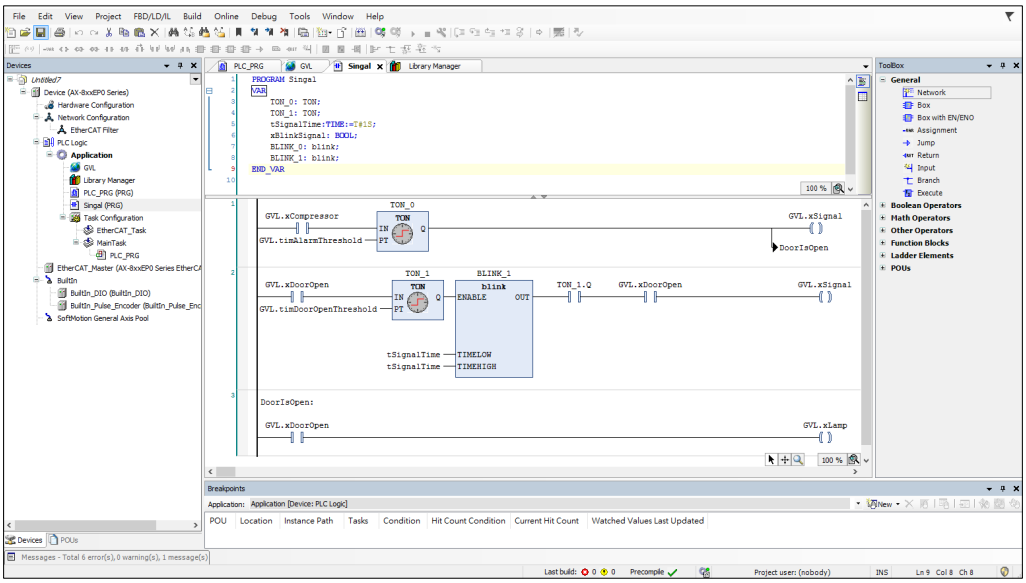
    Select Network 1. In the **ToolBox** view, in the **Common** category, drag a **Jump** element to the network. Place it on the **Add output or jump here** box that appears.

    Result: The jump element appears. The jump destination is still specified as ???.

18. Select ??? and click [...]. Select DoorIsOpen from the possible label identifiers and click *OK* to confirm.

    Result: The label to Network 3 is implemented.

The LD program now looks as shown in the following figure



Calling the Signals program in the main program

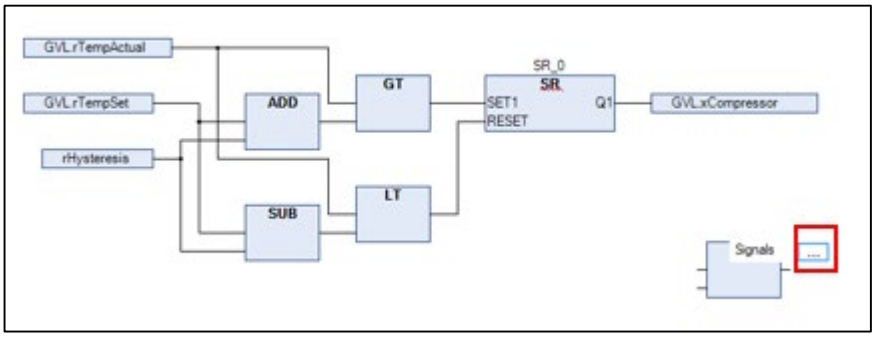
In our program example, the main program PLC_PRG should call the Signals program for signal processing.

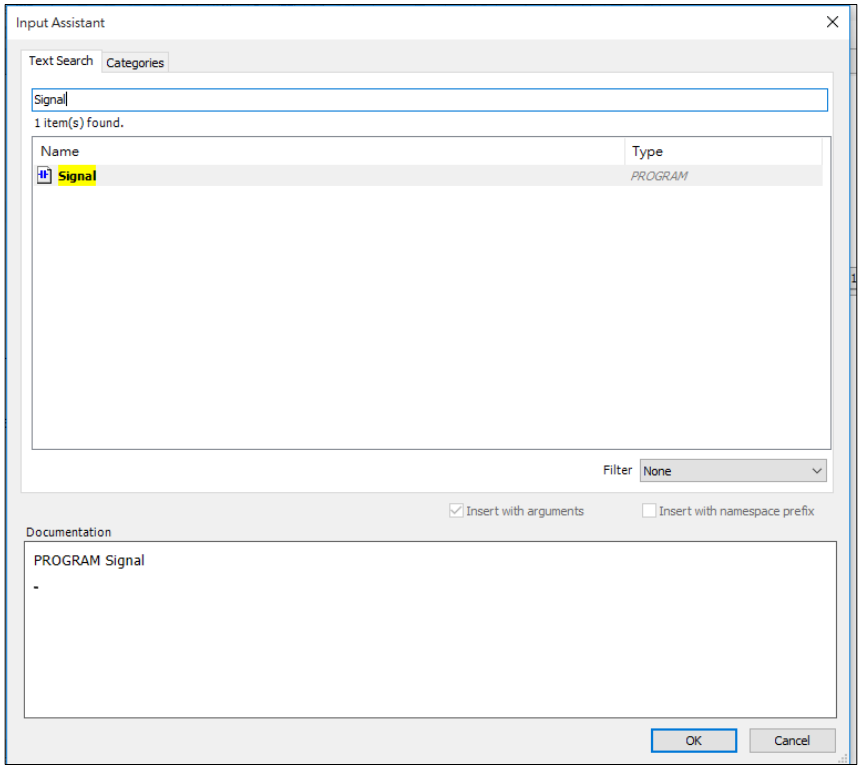
1. In the device tree, double-click PLC_PRG.

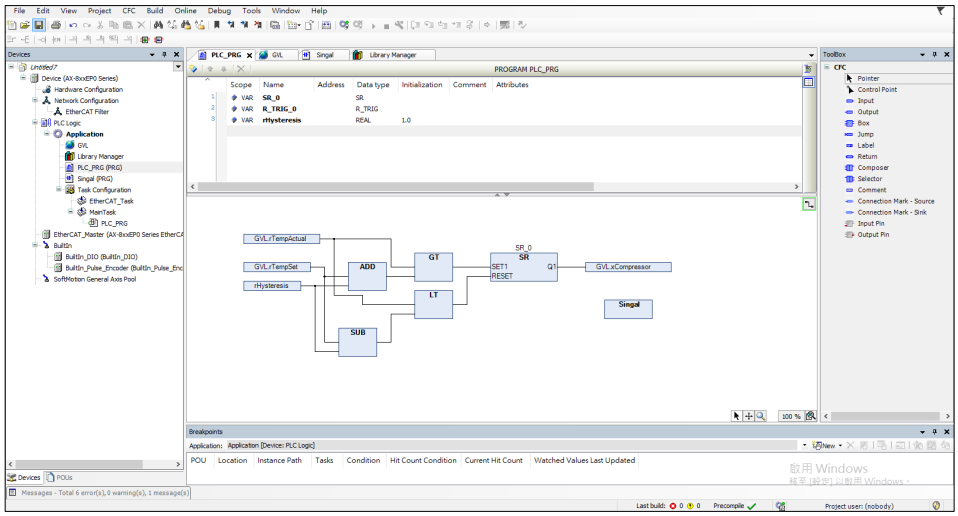
   Result: PLC_PRG opens in the editor.

2. In the **ToolBox** view, drag a **Box** element to the editor of PLC_PRG.

3. Using **Input Assistant,** add this POU from the **POU Calls** category to the call of the Signals program.







Creating an ST program block for a simulation

Since the application in this example project is not linked to real sensors and actuators, user must now also write a program for the simulation of rises and falls in temperature. This will allow user to monitor the operation of the refrigerator controller afterwards in online mode.

User can create the simulation program in structured text.

The program increases the temperature until the main program PLC_PRG determines that the temperature set point has been exceeded and activates the cooling compressor. The simulation program then lowers the temperature again until the main program deactivates the compressor once more.

1. Under the application, insert a POU of the type Program and the implementation language *ST* and with the name Simulation.
2. Implement the following in the ST editor:

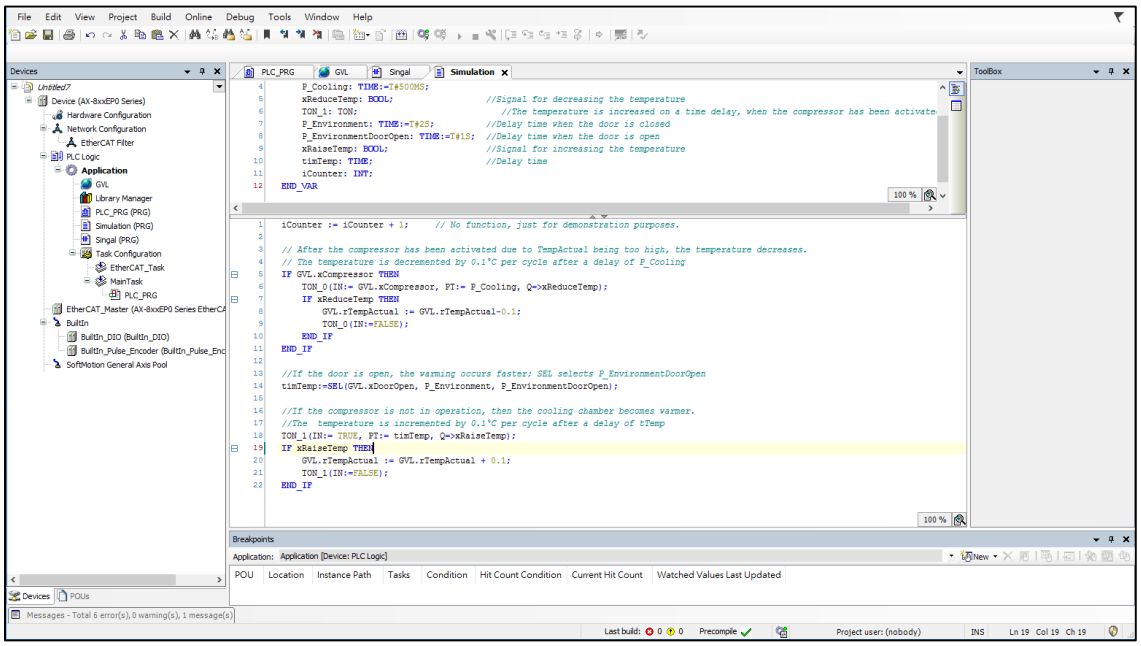
```
PROGRAM Simulation
VAR
    T1: TON;                        //The temperature is decreased on a time
delay, when the compressor has been activated
    P_Cooling: TIME:=T#500MS;
    xReduceTemp: BOOL;              //Signal for decreasing the temperature
```

```
    T2: TON;                          //The temperature is increased on a time
delay, when the compressor has been activated
    P_Environment: TIME:=T#2S;        //Delay time when the door is closed
    P_EnvironmentDoorOpen: TIME:=T#1S;  //Delay time when the door is open
    xRaiseTemp: BOOL;                 //Signal for increasing the temperature
    timTemp: TIME;                    //Delay time
    iCounter: INT;
END_VAR

iCounter := iCounter+1;       // No function, just for demonstration purposes.

// After the compressor has been activated due to TempActual being too high, the
temperature decreases.
// The temperature is decremented by 0.1°C per cycle after a delay of P_Cooling
IF GVL.bCompressor THEN
    T1(IN:= GVL.bCompressor, PT:= P_Cooling, Q=>xReduceTemp);
    IF xReduceTemp THEN
        GVL.rTempActual := GVL.rTempActual-0.1;
        T1(IN:=FALSE);
    END_IF
END_IF

//If the door is open, the warming occurs faster; SEL selects
P_EnvironmentDoorOpen
timTemp:=SEL(GVL.rDoorOpen, P_Environment, P_EnvironmentDoorOpen);
//If the compressor is not in operation, then the cooling chamber becomes warmer.
//The temperature is incremented by 0.1°C per cycle after a delay of tTemp
T2(IN:= TRUE, PT:= timTemp, Q=>xRaiseTemp);
IF xRaiseTemp THEN
    GVL.rTempActual := GVL.rTempActual + 0.1;
    T2(IN:=FALSE);
END_IF
```

Note: The use of a visualization is recommended for convenient operation and monitoring of the entire control program. A visualization created with DIADesigner-

AX is installed in the completed sample project for this tutorial, which is provided with the standard DIADesigner-AX Visualization installation (Projects directory). User can load this project to the controller and start it to see it working together with the visualization.

On starting, the Live_Visu starts up with an illustration of the refrigerator icon that reproduces the operation of the simulation program without user having to make any entries. However, user can bring about the opening and closing of the door with a mouse click the on/off switch and it is possible to adjust the temperature preset on the needle of the rotary control. We will not deal with the creation of the visualization in the context of this tutorial. An appropriate tutorial is planned in the context of DIADesigner-AX Visualization help.



Defining the programs to be executed in the task configuration

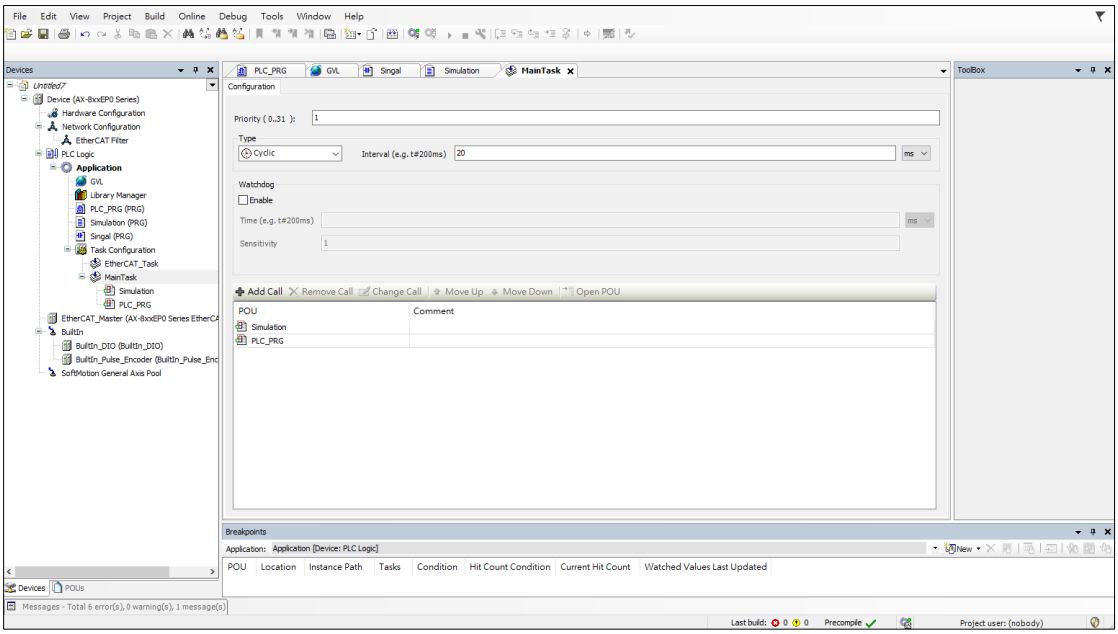
The default task configuration contains the call for the main program PLC_PRG. For our sample project, user also need to add the call for the Simulation program.

1. In the device tree, drag the Simulation entry to Main Task below **Task Configuration**.

   Result: The Simulation program is inserted into the task configuration.

2. To view the task configuration, double-click the Main Task entry to open the editor.

Result: In the table in the lower part of the editor, user will see the POUs that are called by the task: PLC_PRG (entered by default) and Simulation. The call type of the task is **Cyclic** at intervals of 20 milliseconds. In online mode, the task will execute the two POUs one time per cycle.



Debugging the application program

During the input of code, DIADesigner-AX immediately alerts user to syntax errors by a red wavy line underneath the text concerned. Press F11 to have the entire application checked for syntax. The result of the check is shown in the message view. If necessary, open the message view using **View** > **Messages** command. User can then select a message and jump to the corresponding point in the code with the F4 key.

User can only load an error-free application to the controller afterwards.

3.3.1.8 Download and upload

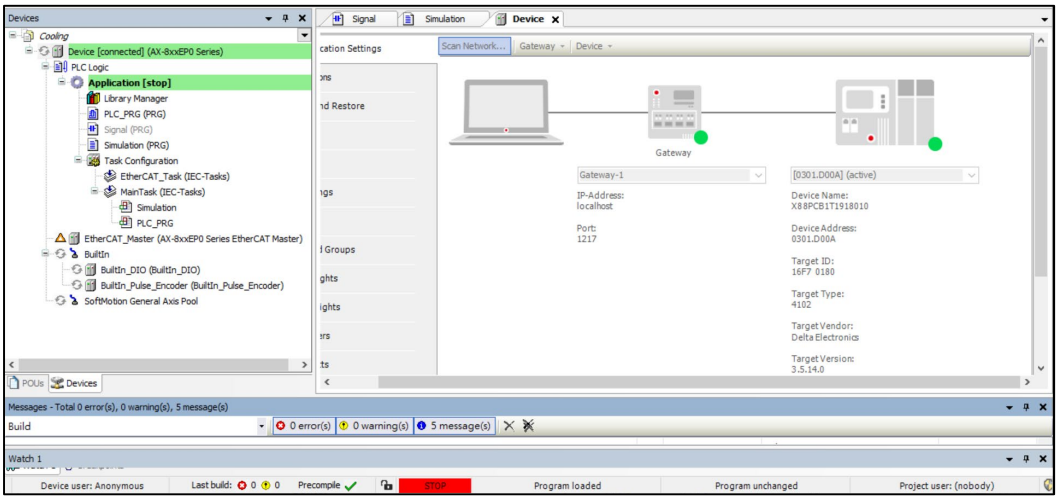
Before downloading the application to the PLC, you can refer to Debugging the application program to make sure the application has been compiled without errors.

1. Click **Online** > **Login**.

   Result: A dialog prompt opens to ask whether or not the application should be downloaded to the controller.

2. Click **Yes**.

   Result: The application is downloaded to the controller. The entries for the controller and the application in the **Devices** view are highlighted in green. Stop appears after the **Application** object. The current status of the controller appears in the taskbar: STOP.



Starting the application

If user have followed this tutorial precisely up until now, then user can use **Application** on the PLC **Device**.

In the **Devices** view, in the context menu of the **Application** object, click **Start**.

Result: The program starts running. The entries for the controller and the application in the **Devices** view are highlighted in green. Run appears after the **Application** object. The current status of the controller appears in the taskbar: RUN.

3.3.1.9 HMI tag sharing

DIADesigner-AX supports export of variables. User can define the variables in DIADesigner-AX and export the xml file of **symbol configuration**. User can import the xml to DIAScreen to communicate between controller and Delta HMI functions. Please refer to ***HMI Interactive*** HMI Interactive for more information.

3.4 Engineering

3.4.1 Update devices

DIADesigner-AX provides a device update function. You can update devices in the project to devices of different versions or types.

To update devices

1. On the menu bar, select **Tools** > **Options** > **General** > **Enable all the devices (include sub devices) update**.
   All the supported sub-devices will be updated and added, and unsupported sub-devices will be removed.
2. Do one of the following:
   - On the menu bar, select **Project** > **Update Device**.
   - Right-click the device that you want to update and select **Update Device**.
3. In the **Update Device** dialog, select the device to be updated to.
4. (optional) You can select **Update same devices in project** to update all the devices with the same ID and type in the project.
5. Click **Update Device**.
6. In the **Update Device List** dialog, check the update information and select the version for the target device.
7. Click **Update**.

3.4.1.1 Update Device List dialog

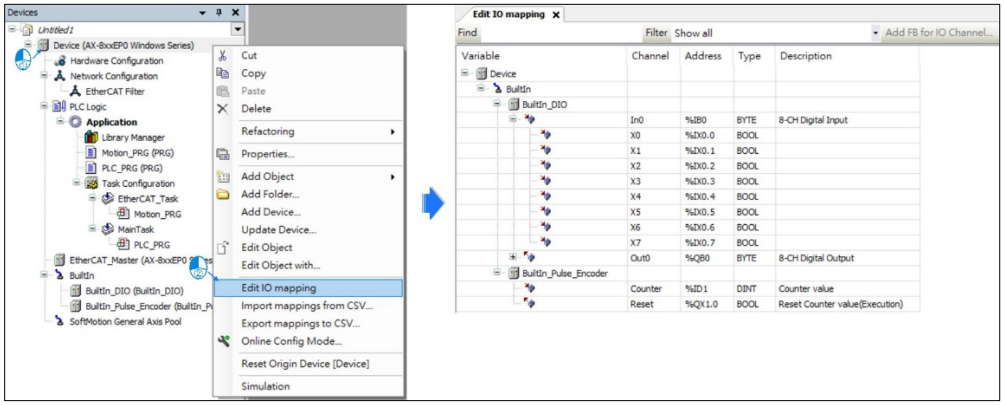
The following table lists the settings in the **Update Device List** dialog.

Item	Description
Device Name	The name of the device in the project tree.
Source	The type and version of the original device.
Target	The type and version of the target device. You can select the version in the dropdown list.
	Some data and configurations may not be updated to the target device due to incompatibility.
	This device will be removed.
	The device will be successfully updated and the configuration will be transferred to the target device.
Don't show again	If you select the checkbox, the **Update Device List** dialog will not show again. If any updating error occurs, you need to check it manually.

3.4.2 Edit I/O mapping

The following are the steps to Edit I/O Mapping

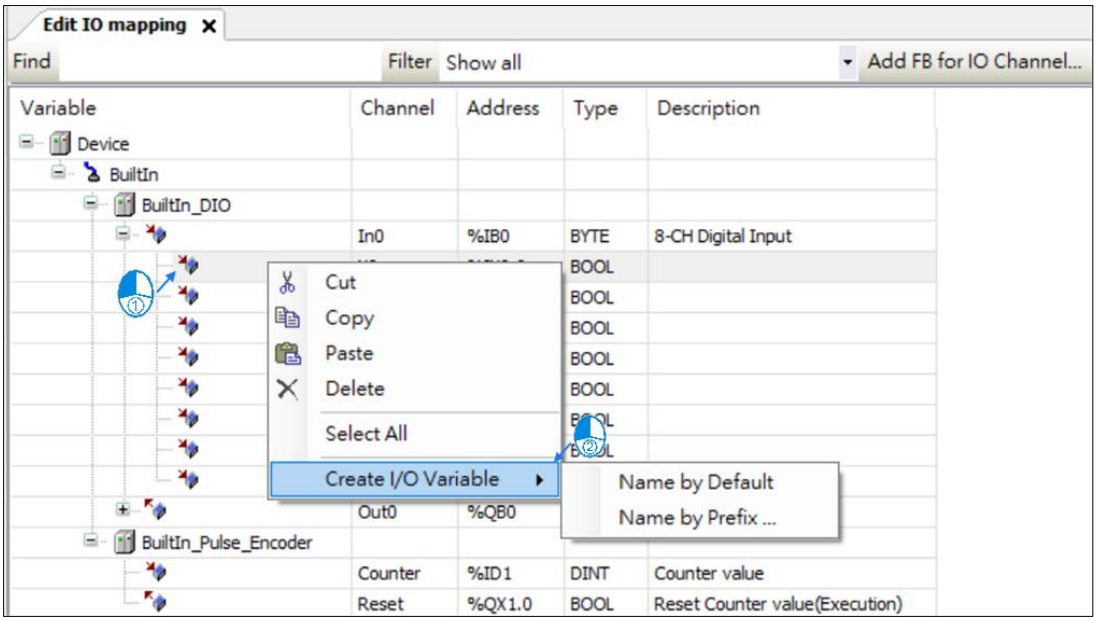
1. Right-click the **Device** in the project tree and select **Edit IO mapping** in the context menu.
   Result: On the **Edit IO mapping** tab, all IO maps of the currently selected device will be displayed, including the IO maps of all additional devices in the project tree.



The following table lists the description of the **Edit IO mapping** window:

Function	Description
Find	After entering the keyword, the search result will be marked in yellow.
Filter	The I/O allocation filter drop-down menu displayed in the mapping table: • Show all. • Show only inputs. • Show only mapped variables. • Show only mapping to existing variables. • Show only mapping to new variables. • Show only outputs. • Show only unmapped variables.

According to the configuration, the system can automatically name each variable: right-click each blank variable field > **Create I/O Variable** > **Name by Default/Name by Prefix**.


Edit IO mapping
Find
Filter
Show all
Add FB for IO Channel...
Variable
Channel
Address
Type
Description
Device
BuiltIn
BuiltIn_DIO
In0
%IB0
BYTE
8-CH Digital Input
BOOL
Cut
Copy
Paste
Delete
Select All
Create I/O Variable
Name by Default
Name by Prefix ...
Out0
%QB0
BuiltIn_Pulse_Encoder
Counter
%ID1
DINT
Counter value
Reset
%QX1.0
BOOL
Reset Counter value(Execution)

Chapter 4: Programming reference

4.1 Variable

Use variables to store and process information. The variable types include:

- Local variable (VAR)
- Input variable (VAR_INPUT)
- Output variable (VAR_OUT)
- I/O variable (VAR_IN_OUT)
- Temporary variable (VAR_TEMP)
- Global variable (VAR_GLOBAL)

You can also add flags to the variable to specify the attribute:

- Retain variables (RETAIN)
- Persistent variable (PERSISTENT)
- Constant variable (CONSTANT)

You can set up variables in POU, Global Variable List, Network Variable List, Persistent Variables, and the DUT editor.

The following table identifies the variable parameters and their descriptions.

Item	Description
Scope	Type of the variable.
Name	Name of the variable.
Address	The address that stores the variable.
Data Type	Data type of the variable. A standard data type or a user-defined data type.
Initialization	The value of a variable when a controller starts running for the first time.
Comment	User-defined text for the variable.
Attribute	Properties of the variable.
Dimension	The number of elements of an array. For example: [1..10,1..5] represents a two-dimensional array containing a total of 50 elements.

4.1.1 Create an array variable

An array is a collection of data elements of the same data type. One and multi-dimensional arrays are supported. Use the following steps to create array variables in the LD POU editor. You can use the same steps for the other supported editors.

To create an array variable

1. Create a project.
2. On the **Devices** pane, right-click **Application,** and then select **Add Object** > **POU**.
3. In the **Add POU** dialog, enter the POU name and select **Ladder Logic Diagram (LD)** in **Implementation language**, and then click **Add**.
4. In the variable declaration editor, click  to switch to Tabular View, and then click  **to insert** a variable.
5. Enter a name for the variable.
6. Double-click the **Data Type** box and click  , and then select **Array Wizard.**
7. In the **Array dialog**, enter **Dimensions** and **Base Type**. If you want to find a base type, click  and select **Input Assistant**. In the **Input Assistant** dialog, search the base type under **Text Search**, or select one from **Standard Types** or **Structured Types** under **Categories**.

4.1.2 Create a structure variable

You can create structures in a project as DUT (Data Unit Type). Use the following steps to create structure variables.

To create a structure variable

1. Create a project.
2. On the **Devices** pane, right-click **Application,** and then select **Add Object** > **DUT**.
3. In the **Add DUT** dialog, enter the name and select a type, and then click **Add**.
4. In the **DUT** editor, enter the text for the structure.
5. Create a POU.
6. In the variable declaration editor, click  to switch to Tabular View, and then click  **to insert** a variable.
7. Enter a name for the variable.
8. Double-click the **Data Type** box and click  , and then select **Input Assistant.**
9. In the **Input Assistant dialog**, on the **Categories** tab, select **Structured Types**.
10. Expand the **Application** node, and select the DUT you created, and then click **OK**.

4.1.3 Set up comments for array and structure variables

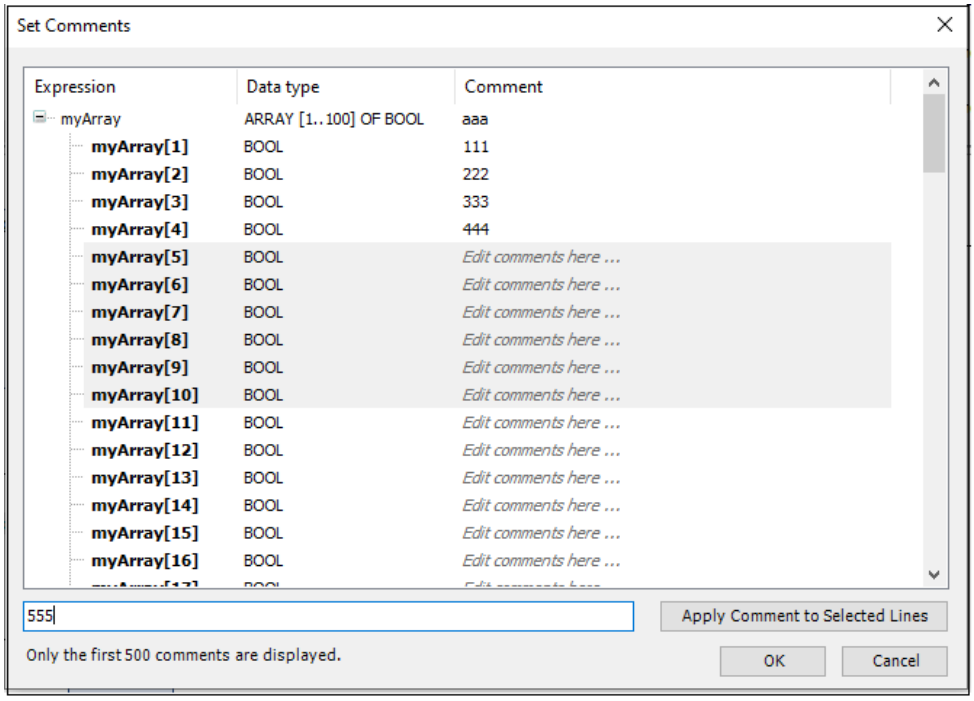
You can set up comments for array and structure variables in:

- POU editors (FBD, LD, ST, CFC, and SFC), and the Method editor
- Global Variable List
- Global Variable List (tasklocal)
- Network Variable List (Sender)
- Persistent Variables

To set array and structure variable comments

1. Create an array or structure variable.
2. Do one of the following to open the **Set Comments** dialog:
   - Right-click the variable, and then select **Set Comments.**
   - Double-click the **Comment** box and then click ... .
3. In the **Set Comments** dialog, enter the comments for the variable and the sub-elements.

   If you want to enter the same comment for multiple lines, select the lines and enter the comment in the bottom textbox, and then click **Apply Comments to Selected Lines**.

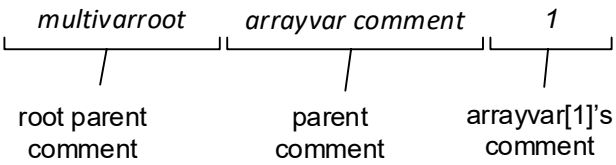


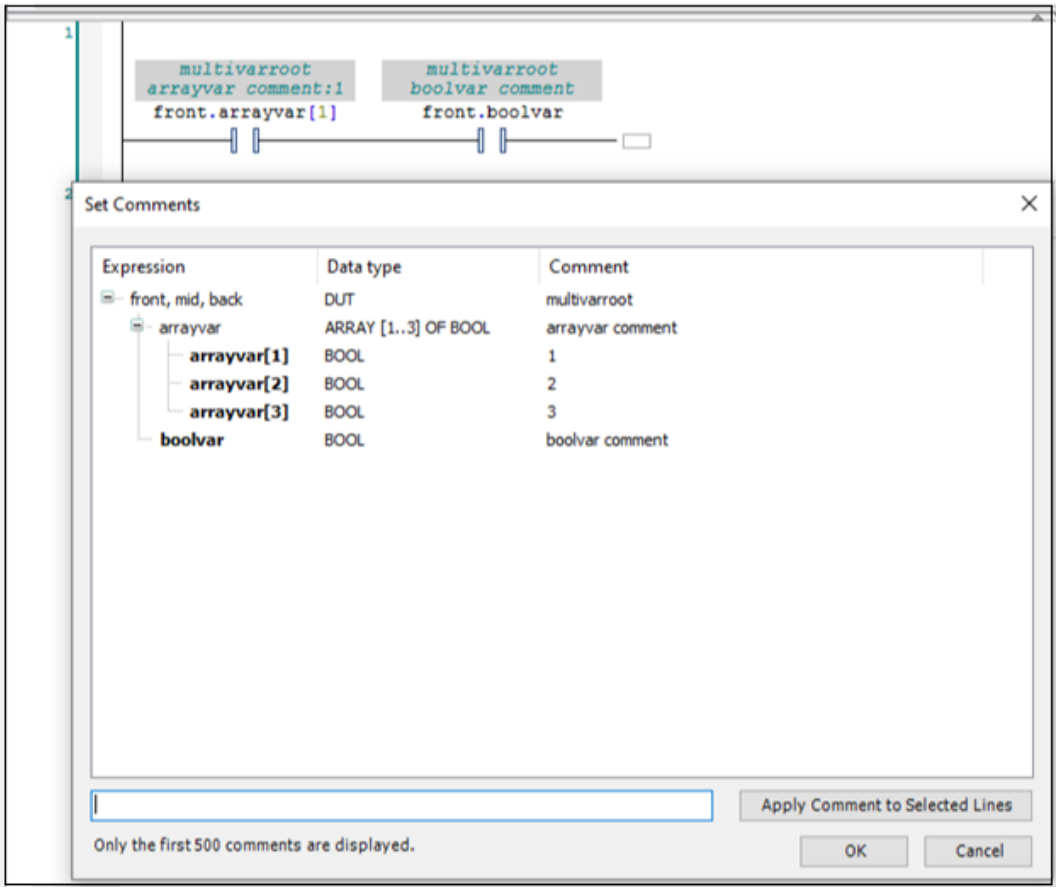
4. Click **OK**. The sub-comments will be synchronized and shown in the **Attributes** column.

To view the comments in the POU editor

1. Insert an element in the **POU** editor and assign a variable to it.
2. Do one of the following:
   - Check the tooltip by hovering over the element.
   - Go to **Tools** > **Options** > **FBD**, **LD and IL editor**, and select **Show symbol comment**, then go back to the **POU** editor and view the symbol label.

Note: After the comment is set, it will be shown with all the comments of its parent levels. In the following example, arrayvar[1] is assigned to Contact, and when you view the comments of arrayvar[1], it will show *multivarroot arrayvar comment:1*.





4.1.4 Set up retain persistent variable

Retain persistent variables can protect the variables from power failure. The value of the variable will retain even after a restart.

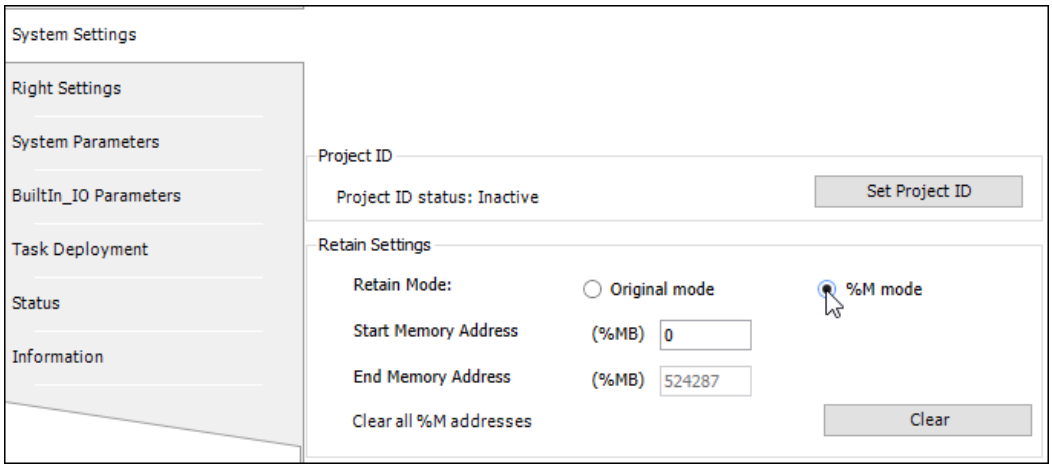
Note: Only recommend setting up variables that do not exist in CODESYS persistent variables.

Supported devices:

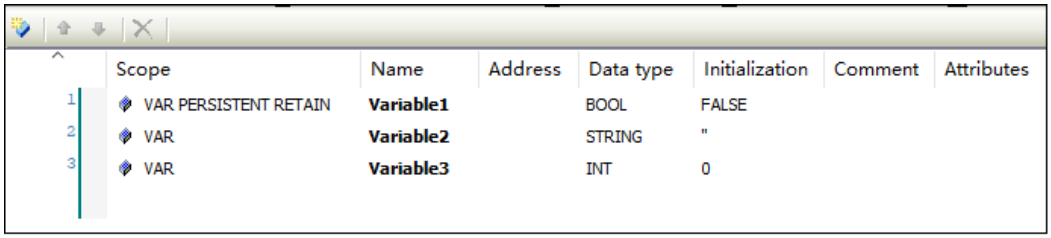
- AX-3 v1.0.4.1 or later
- AX-8 (Linux) v3.5.15.45 or later
- AX-8 (Windows) v3.5.14.30 or later

To set up retain persistent variables

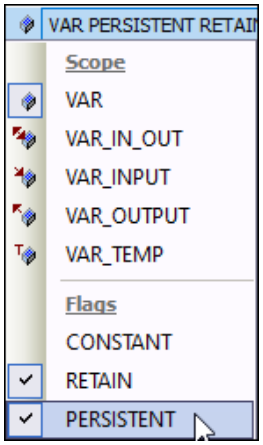
1. On the **Devices** pane, double-click the controller.
2. Go to **System Settings** > **Retain Settings** and select **%M mode**.
3. Enter **Start Memory Address**. The value can start with the default value in **Start Memory Address** and cannot exceed the value in **End Memory Address**.



4. Go to **Application**, add a POU program or open an existing one.
5. Insert variables in the POU program. Enter **Name**, and select **Data Type** and **Initialization**.

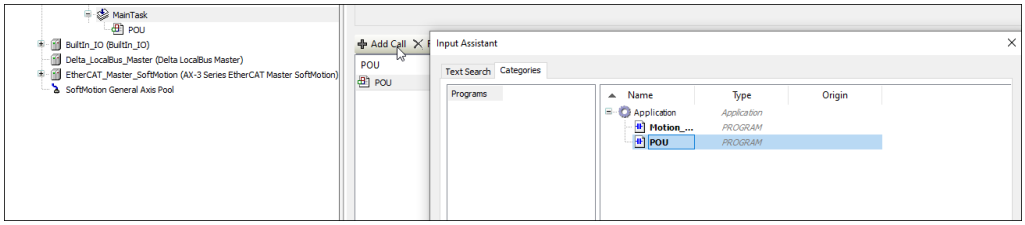


6. Select the variable and click  , and then select **Retain** and **Persistent**.



7. To call this POU:
   - Go to **Devices** > **Task Configuration**.
   - Double-click **MainTask**.
   - On the **MainTask** tab, click **Add Call**.
   - Select the POU, and click **OK**.

   Note: The persistent variables in the **Global Variable List** also need to be called in the task under the application.



8. Click  to generate the code.
9. Go back to the POU and check the **Address** column of the variables. The retained variable memory addresses are assigned.
10. Connect to the controller, click  to log in, and download the application to the device.

4.1.5 Variable export and import

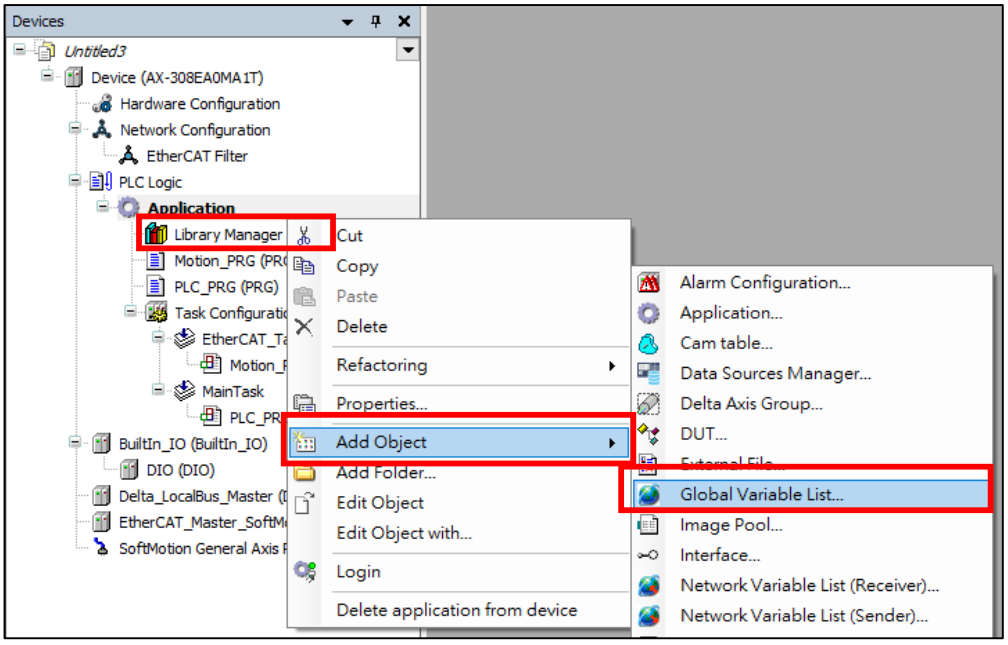
Variable export and import support the following three methods:

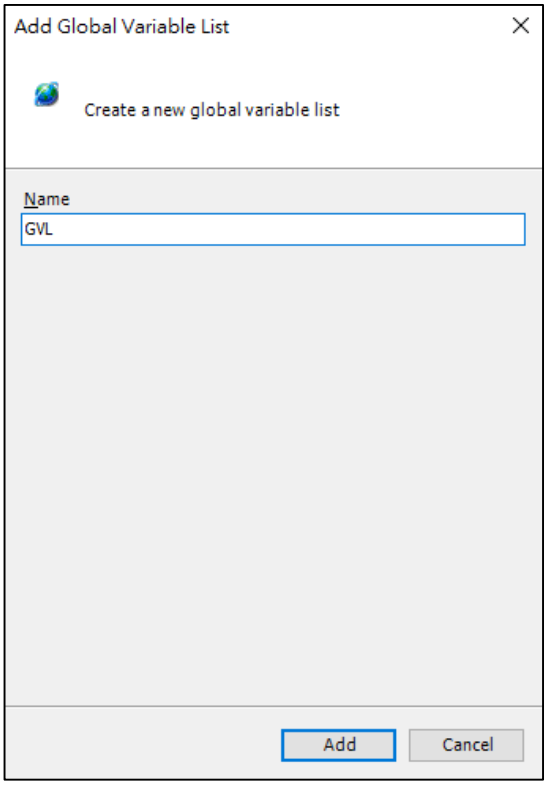
	Export variable	Import variable	File format	
			.xml	.xls
Global Variable List	V	V	V	V
Motion_PRG	V	V	V	V
PLC_PRG	V	V	V	V

Export variable

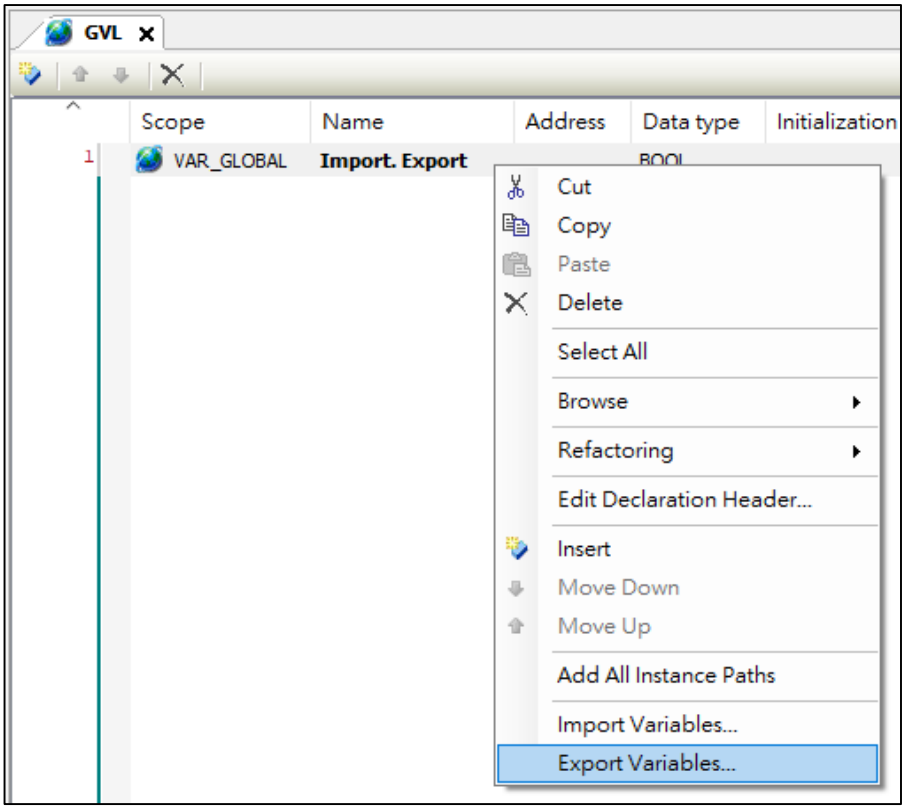
Let us take the **Global Variable List** as an example and follow the steps to export the variables:

1. Right-click **Application** and select **Add Object** > **Global Variable List**. Click **Open** to complete the variable list creation.

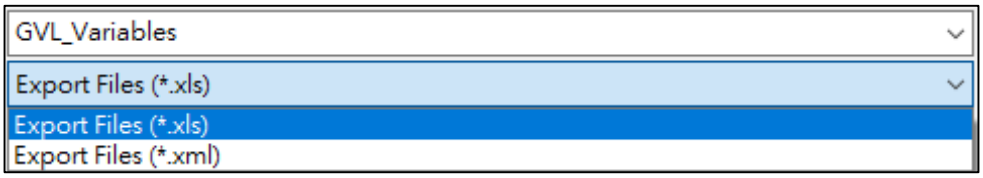




2. After adding a variable, right-click on the screen and select **Export Variables**.



3. Select the desired file format.



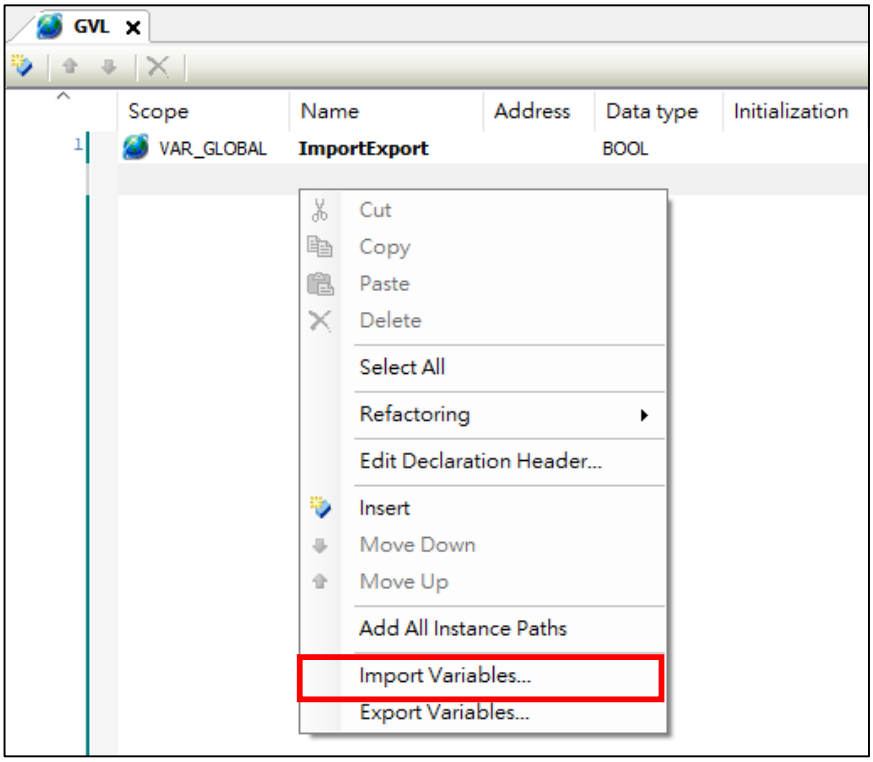
Result: A window pops up showing that the exported variable was successful.

Note: It can be edited in the .xls file, and the comments support Chinese.

Import variable

Let us take the **Global Variable List** as an example and follow the steps to import the variables:

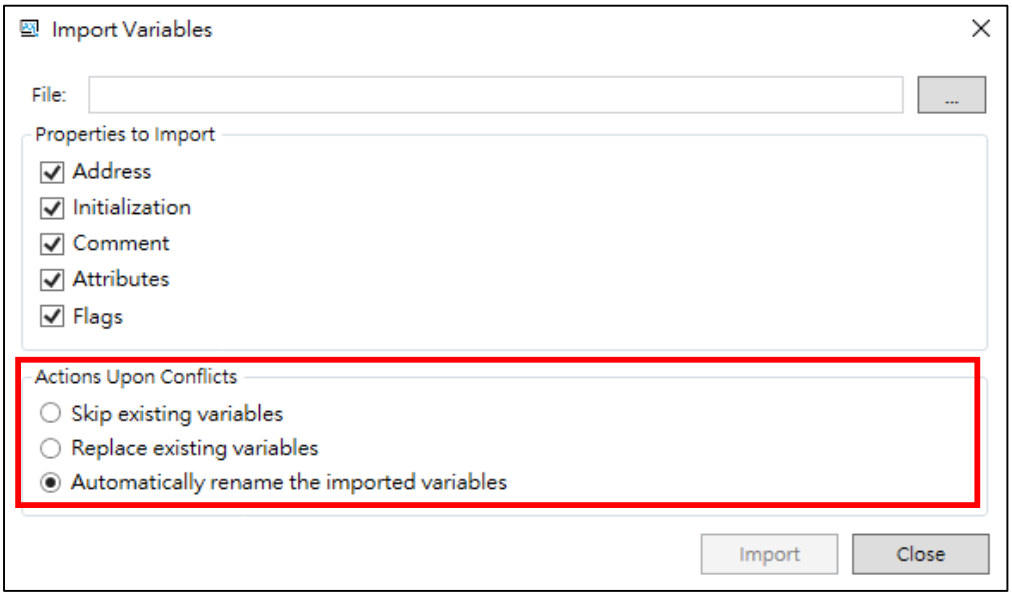
1. Right-click on the **GVL** screen and select **Import Variables** from the context menu to open **Import Variables** window.



2. In the **Import Variables** window, select the following:

- **File**: Click ... to select the xml file to be imported.
- **Properties to Import**: Select the properties to be imported.

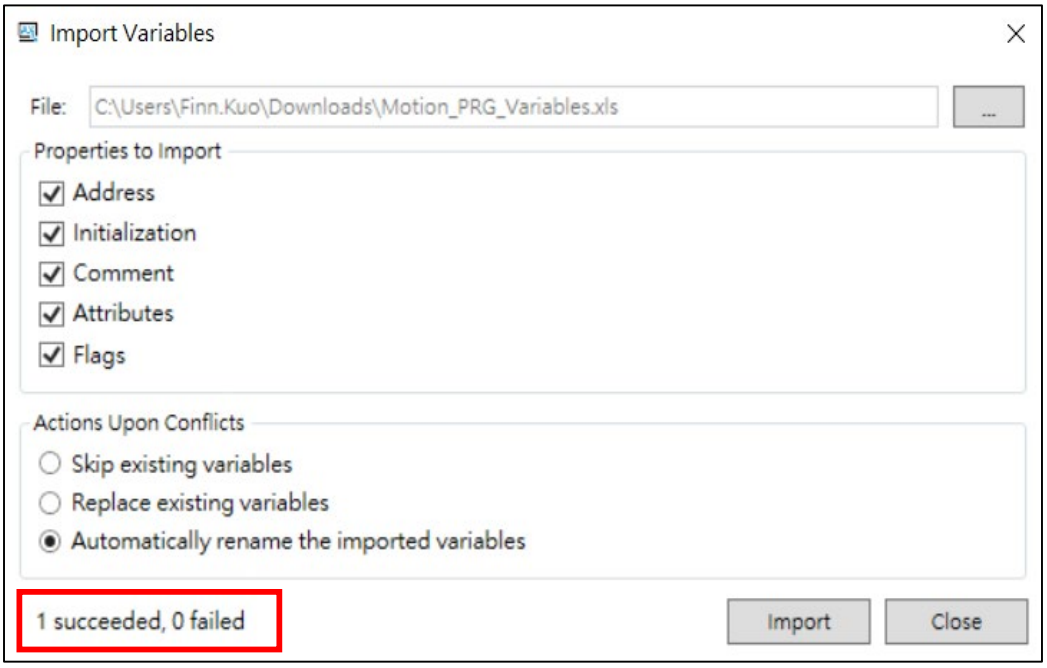
- **Actions Upon Conflicts**: It suggests the actions a user can choose if the variable to be imported conflicts with the variable name of the current software page.



3. Click **Import.**

Result: The selected xml file variable was imported successfully.

Note: The export result information will be displayed in the lower left of the **Import Variables** window.

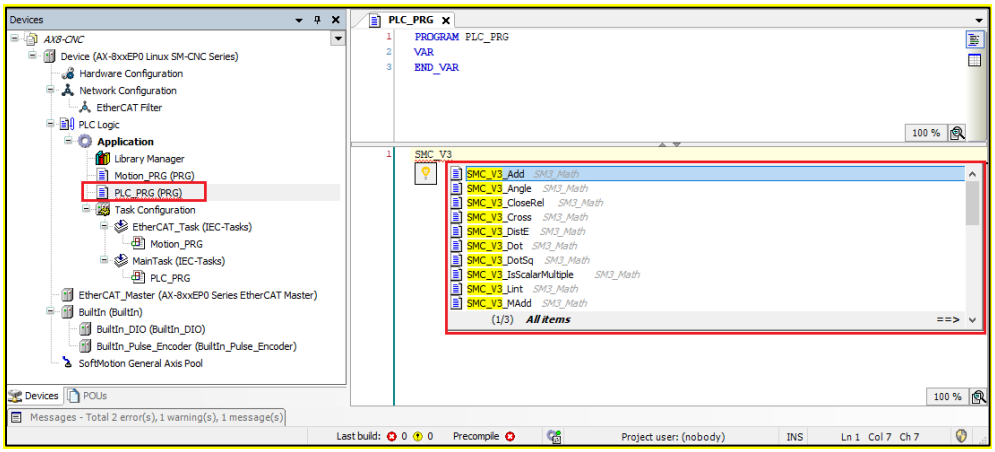


4.2 Auto-completion

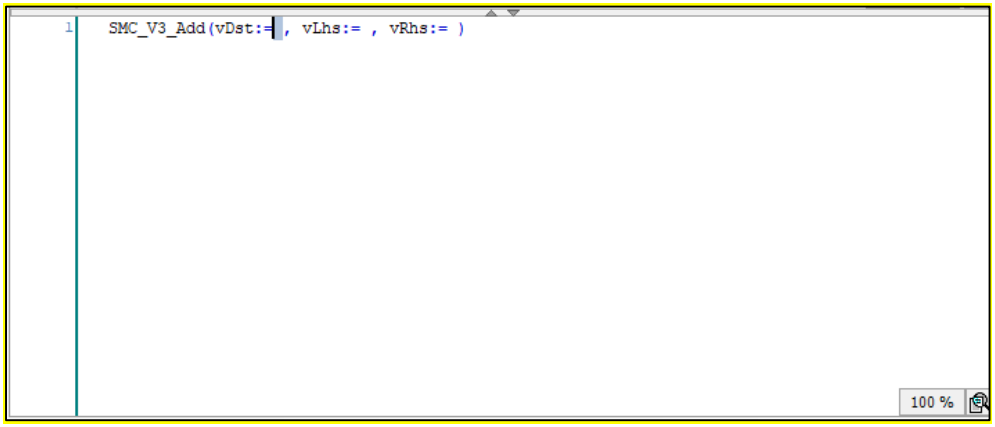
Enter the ST statement type keyword to automatically jump out of the menu, and press the Tab key to auto-complete the parameter template. It supports function block (FB), function (Function), and IF statement.

Example:

1. Entering a keyword will automatically pop up the drop-down menu for selection.



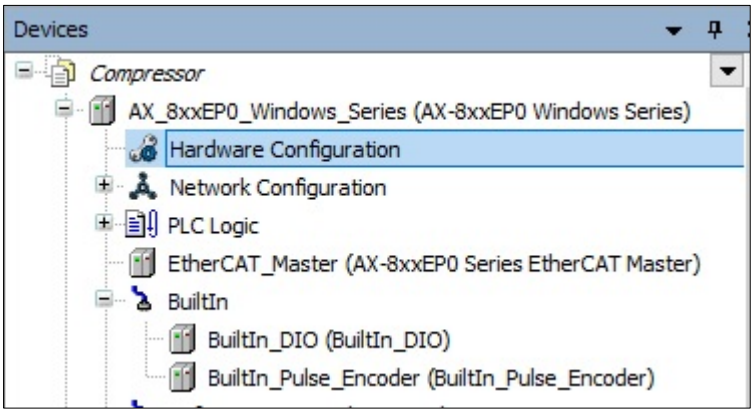
2. Press the Tab key to auto-complete the parameter template.



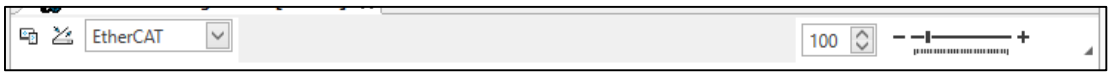
Chapter 5: Hardware configuration

5.1 Hardware module configuration

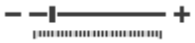
In **Hardware Configuration**, user can perform functions such as configuration of modules, parameter settings for modules, simple on-line detection and diagnosis.



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



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

Function	Description
	Click to display module information.
	Click to perform device scan.
	Select Remote I/O 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.

In the **Hardware Configuration** screen, when the device is connected, a small icon will be displayed on the left of the device to provide information about the device status:

Online Status

- : The PLC is connected, the application is running, the device is in operation, and data is being exchanged. The **Refresh I/Os in stop** check box on the **PLC Settings** tab can be selected or cleared.
- : The PLC is connected and in **STOP**; and the **Refresh I/Os in stop** check box on the **PLC Settings** tab is cleared.
- : The PLC is connected and the application is running. Diagnostic information is available.
- : The device is in preoperative mode and is not running yet. Diagnostic information is available.
- : The device is not exchanging data; bus error, no configuration, or simulation mode.
- : The device is running in demo mode for 30 minutes. When this time has elapsed, demo mode will expire and the fieldbus will end the data exchange.
- : The device is configured, but not operational. No data is exchanged. Example case: CANopen devices when booting and in preoperative mode.
- : Redundancy mode is active. The fieldbus master is not sending any data because another master is active.
- : The device description could not be found in the device repository.
- : The device itself is running, but a child device is not running. The child device is not visible due to a collapsed device tree

5.1.1 AX-3 series module configuration

This section applies to AX-300NA0PA1, AX-308EA0MA1T, AX-308EA0MA1P, AX-364ELA0MA1T, AX-304ELA0PA1T, AX-304ELA0PA1P, AX-316EA0MA1T, AX-324NA0PA1P, and AX-332EP0 series controllers.

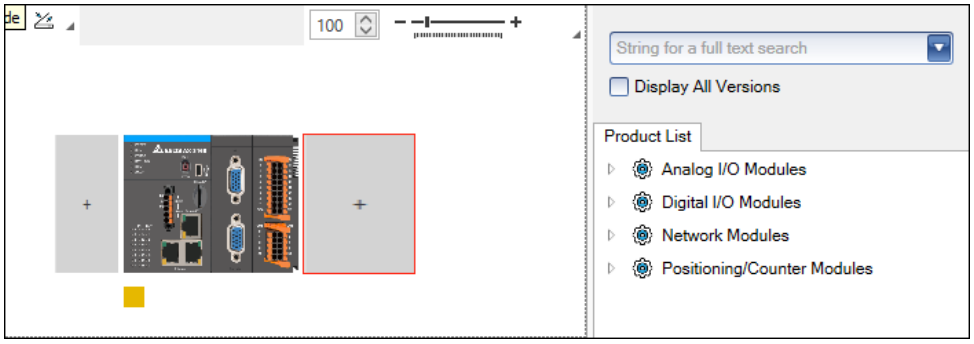
5.1.1.1 Add modules

You can add expansion modules on the right side of the controller in **Hardware Configuration**. There are four types of modules you can select to add:

- Analog I/O Modules
- Digital I/O Modules
- Network Modules
- Positioning/Counter Modules

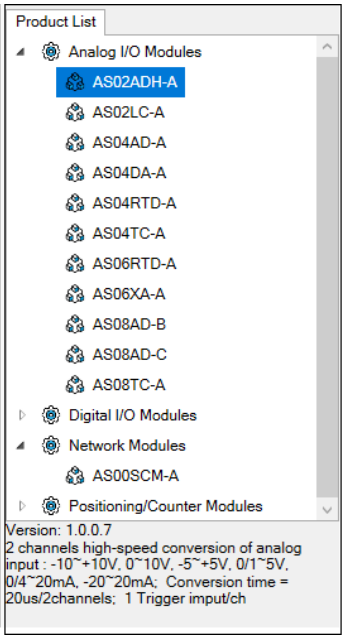
To add modules

1. On the **Devices** pane, double-click **Hardware Configuration**.
2. In the **Hardware Configuration** editor, click the expansion placeholder on the right side of the controller, and then modules will be listed in **Product List**.



3. Expand the module list and add the module. Do one of the following:
   - Double-click the module.
   - Drag the module to the placeholder.

   When you click the module, you can view the module information at the bottom of the **Product List** pane.



Note: The controller supports adding up to four AS00SCM-A modules.

4. (optional) You can right-click the module, and then cut, copy, paste or delete the module if needed.

5.1.1.2 Add a remote I/O module

You can follow the following steps to add a remote I/O module.

To add a remote I/O module

1. **On the Devices pane, double-click** Network Configuration**.**

   Note: The logic controllers do not support adding remote I/O modules.

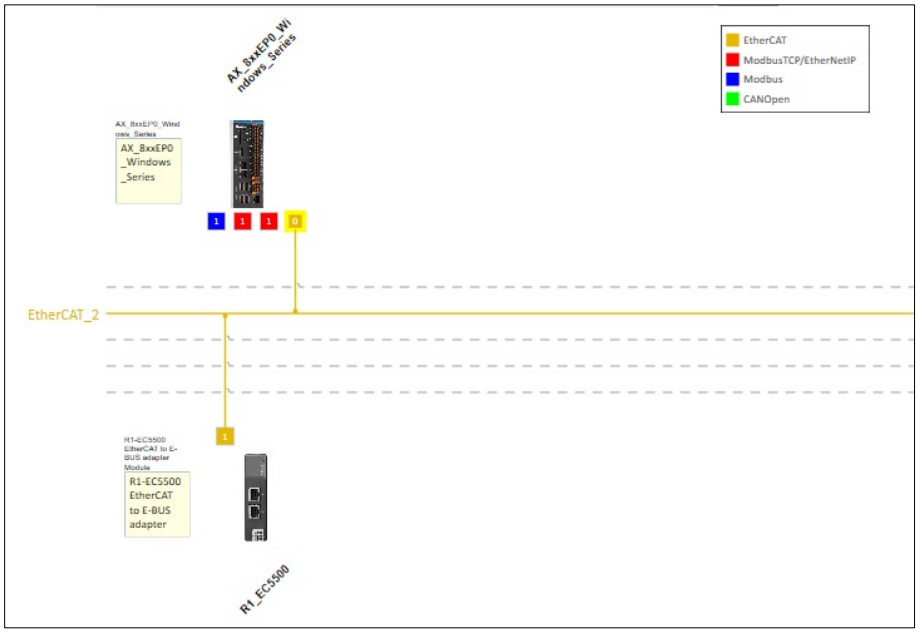
2. In **Product List**, select a remote I/O module, for example, **Fieldbuses** > **EtherCAT** > **R1-EC** > **R1-EC5500 EtherCAT to E-BUS adapter Module**.

3. Add the remote I/O module. Do one of the following:

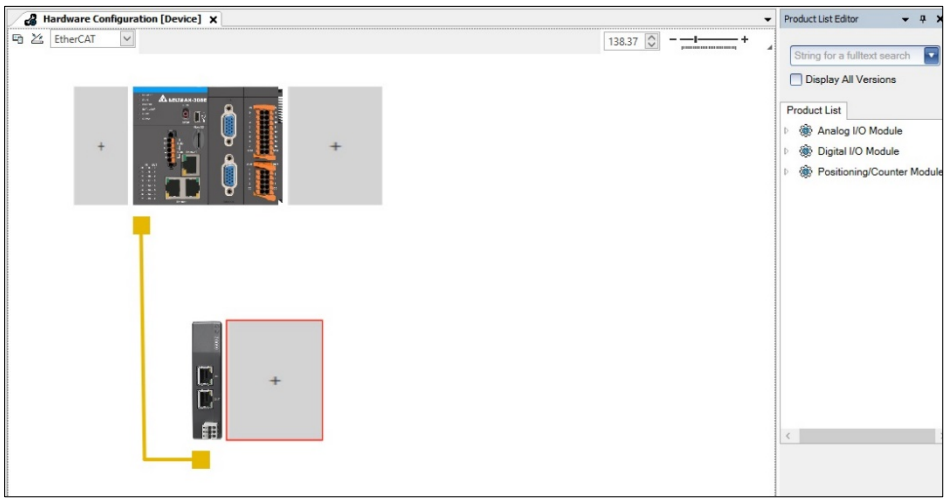
   - Double-click on the module.
   - Drag the module to **Network Configuration Editor**.

4. (optional) You can right-click the module, cut, copy, paste or delete the module if needed.

5. Connect the module to the same yellow EtherCAT line as the master station.



6. On the **Devices** pane, double-click **Hardware Configuration** and the remote I/O module shows under the I/O module.



Note:

- Up to 64 remote I/O modules can be added to the AX-308/AX-316 series controller.
- Up to 16 remote I/O modules can be added to the AX-304 series controller.
- Up to 96 remote I/O modules can be added to the AX-364 series controller.

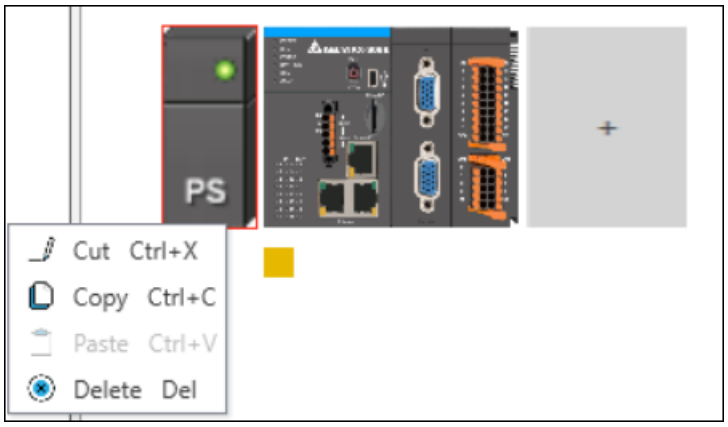
5.1.1.3 Add a power module

The power supply modules supply power to the controller. It can only mount to the left of the controller and do not have any configuration data.

To add a power module

1. On the **Devices** pane, double-click **Hardware Configuration**.
2. In the **Hardware Configuration** editor, click the expansion placeholder on the left, and the power supply modules will be listed in **Product List**.
3. Expand the module list and add the power supply module. Do one of the following:
   - Double-click the module.
   - Drag and drop the module to the placeholder.

   When you click the module, you can view the module information at the bottom of the **Product List** pane.
4. (optional) You can right-click the module, cut, copy, paste or delete the module if needed.



5.1.1.4 Add and configure a function card

If you add a Network Module AS00SCM-A to the controller, you can also add a Function Card to expand the function of the module. AS00SCM-A module provides two slots to add the following function cards:

- AS-F232
- AS-F422
- AS-F485
- AS-FCOPM (slot two only)

Prerequisites

- Add the AS00SCM-A module to the controller.

To add a function card

Do one of the following:

- In the **Hardware Configuration** editor, click the slot in the AS00SCM-A module, and then add the function card from **Product List**.
- Under AS00SCM-A, right-click the **<Empty>** node and select **Plug Device**. In the **Plug Device** dialog, select the function card and click **Plug Device**.

To configure the function card as a Modbus master COM port

1. Add one of the following function cards:
    - AS-F232
    - AS-F422
    - AS-F485
2. Double-click the function card and go to **Configuration**.
3. Select the **Data Exchange** checkbox.
4. Click **Add Channel**. Up to 32 channels can be added.
5. In the **Modbus Channel** dialog, configure and enable the channel.
6. On the **AS-XXX Mapping** pane, configure the I/O mappings to map the input and output.

To configure the function card as a Modbus slave COM port

1. Add one of the following function cards:
    - AS-F232
    - AS-F422
    - AS-F485
2. Double-click the function card and go to **Configuration**.
3. Clear the **Data Exchange** checkbox.
4. Configure the settings and I/O mappings of the function card. When configured as a Modbus slave, the function card can provide 100 input and 100 output Word addresses.

To configure the function card as a CANopen slave

1. Add the function card AS-FCOPM in slot two under the AS00SCM-A module.
2. On the **Configuration** pane, enter **Station ID** from 0 to 63.
3. On the **AS-FCOPM IO Mapping** pane, configure the I/O mappings to map the input and output.

5.1.1.4.1 Modbus channel configuration page

In the **Modbus Channel** dialog, you can configure the Modbus protocol to exchange data with controllers.

The following table lists the settings and their descriptions to configure the Modbus channel.

Setting	Description
Enable	Enable or disable the data exchange channel.
Channel Name	Enter the name of the channel.
Slave Address	Set up the slave address. The valid range is from 1 through 240.
Device Type	Select the device type.
Access Type	Set up the **Read** and **Write** access type. If **Read/Write Multiple Registers** is selected for one of the **Read** or **Write** options, it will also synchronize with the other.
Device Address	Select the device address and enter the value in hexadecimal format.
Length	Set up the word length that reads or writes from or to the register. The value varies with different **Access Type** and **Access Address** which can be 0–1, 0–100, 0–256, and 1–100.

5.1.1.5 Set up AS02LC wizard

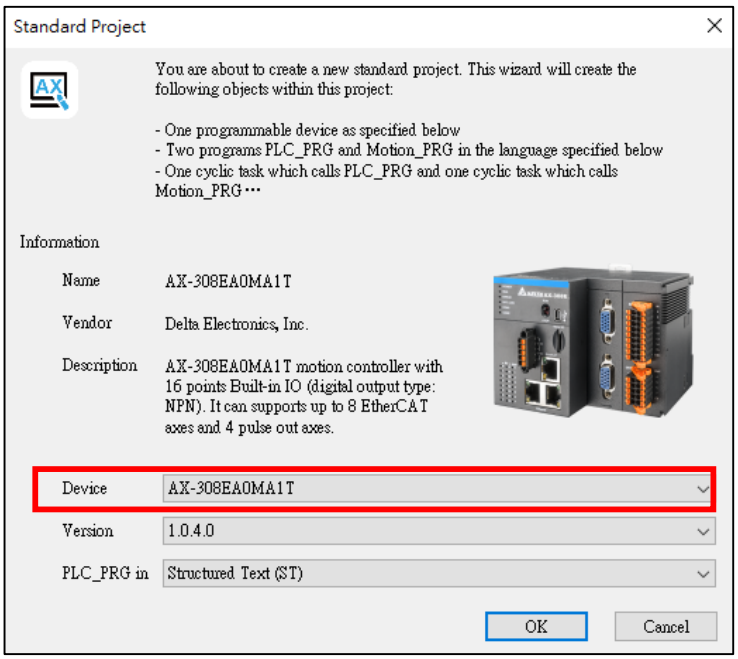
The AS02LC calibration wizard is to make the displayed value of the module weight match the weight of the load cell.

Note:

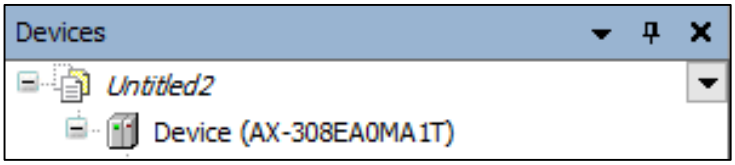
1. This applies to AX-3 series controllers, and the firmware version must be greater than or equal to 1.0.3.0.
2. Only online tuning is supported.

Follow the steps below to start using the Calibration Wizard:

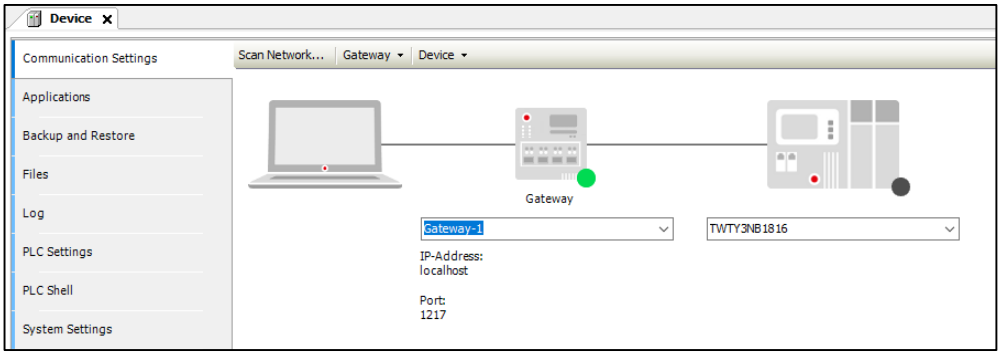
1. Create an AX-3 series project, take AX-308EA0M1T as an example.



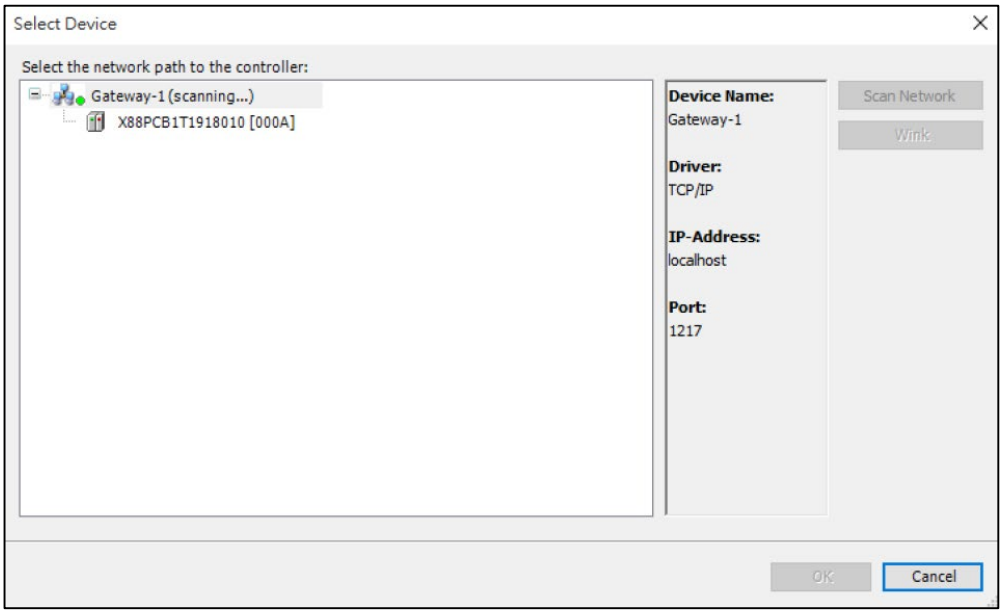
2. **Double-click** on the device AX-308EA0M1T to open the Device setting window.



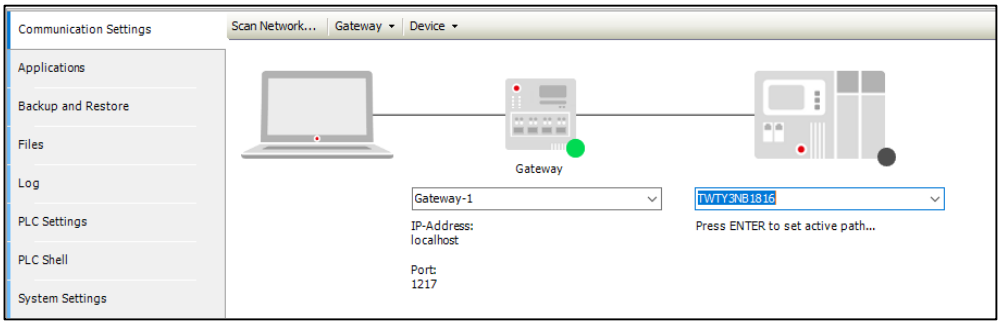
3. Click **Scan Network** in **Communication Settings**, and the **Select Device** window appears. (Please make sure the network cable is plugged in).



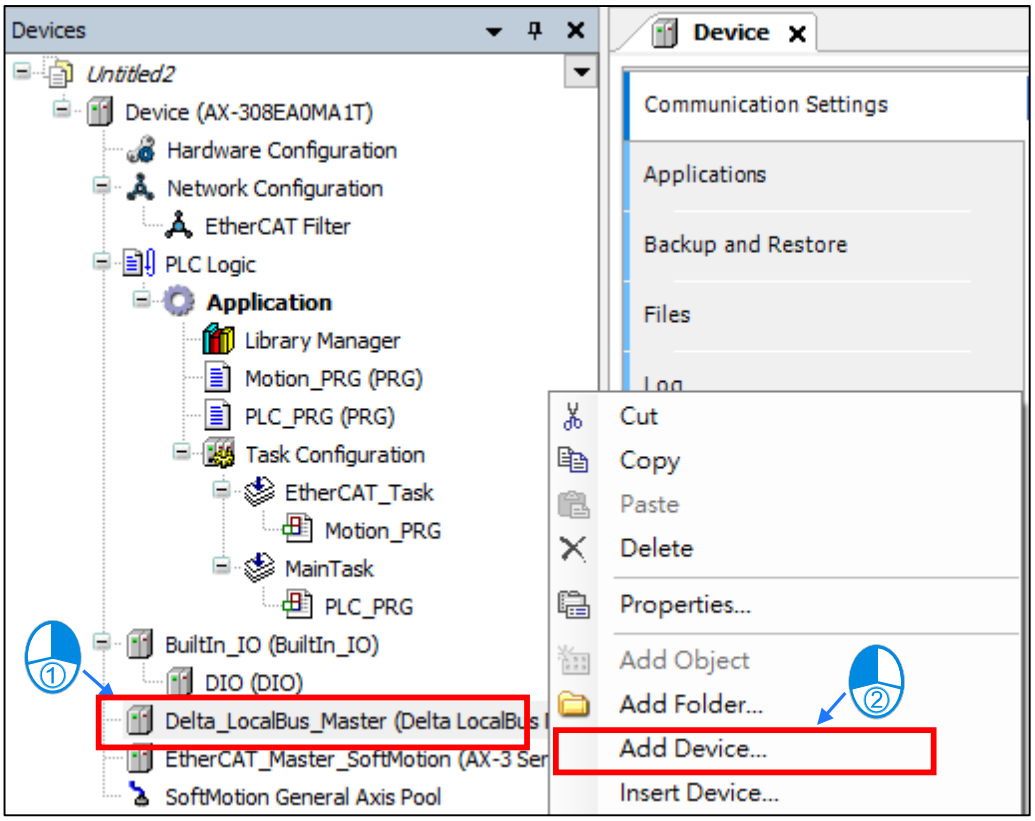
4. In the window, select the device to be connected and click **OK**.



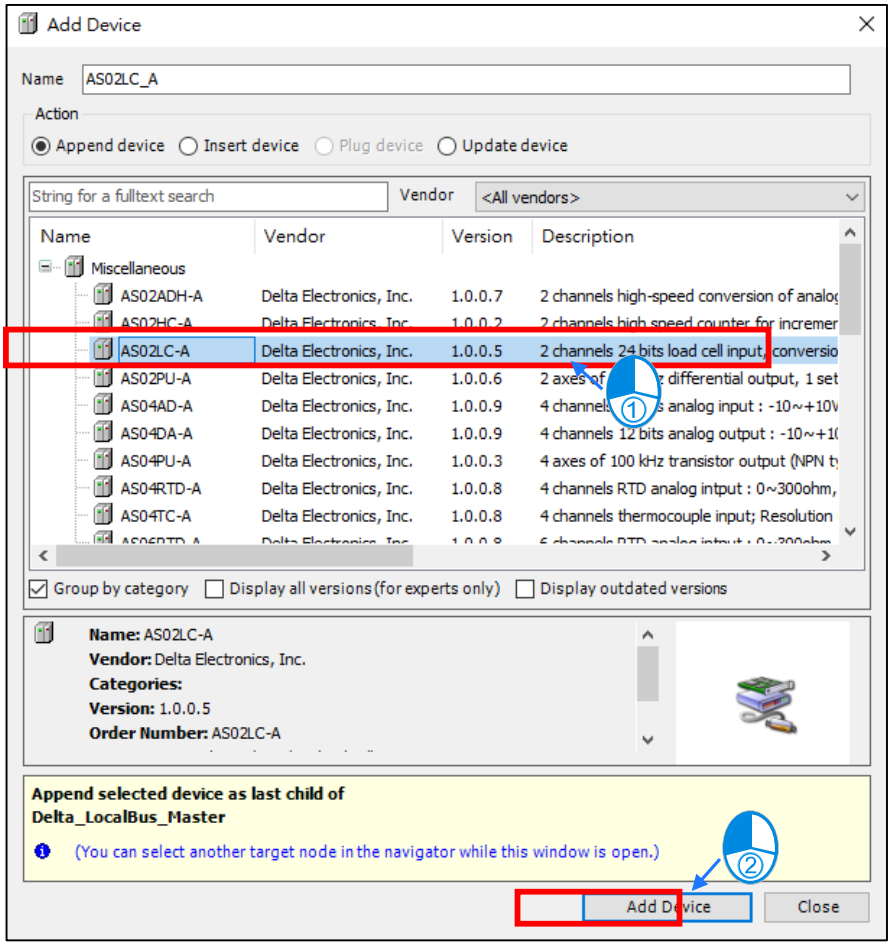
5. The software scans the device automatically and displays **Activation** to indicate successful connection.



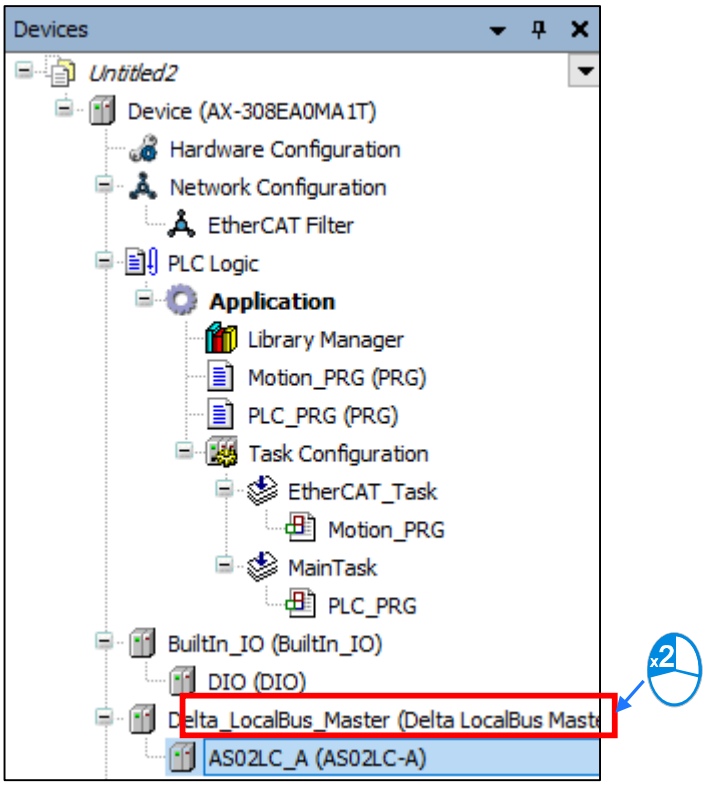
6. Right-click **Device** > **Delta_LocalBus_Master** and click **Add Device**.



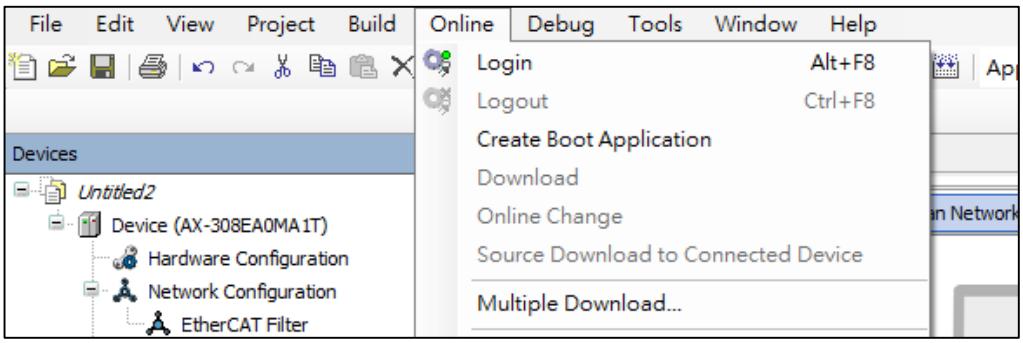
7. Select the device AS02LC-A and click **Add Device**.

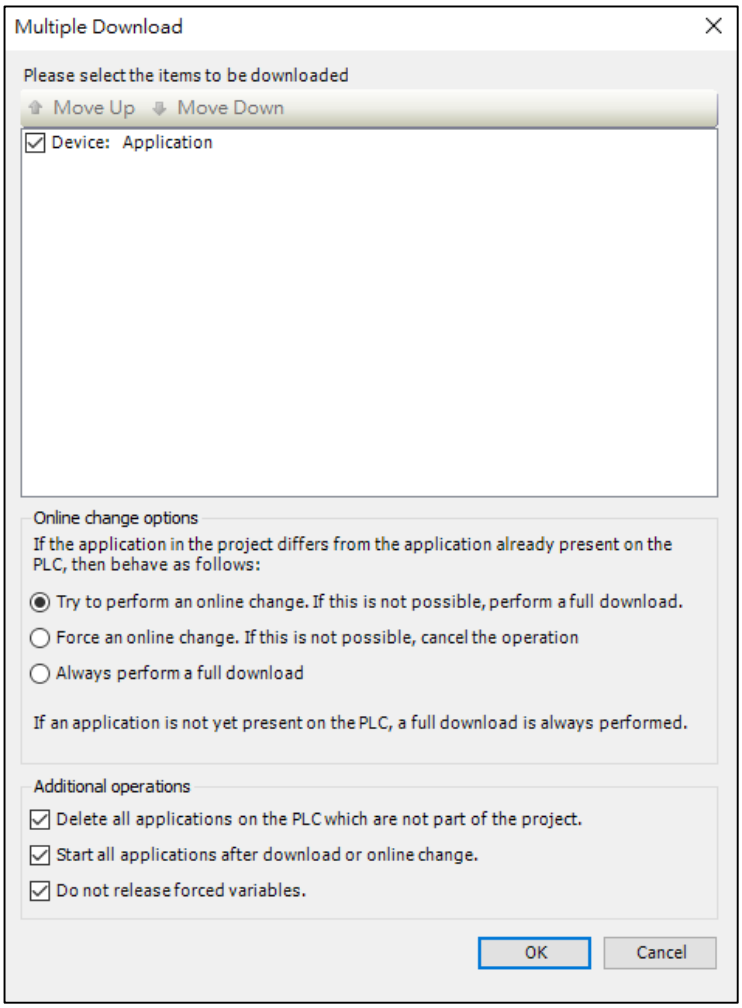


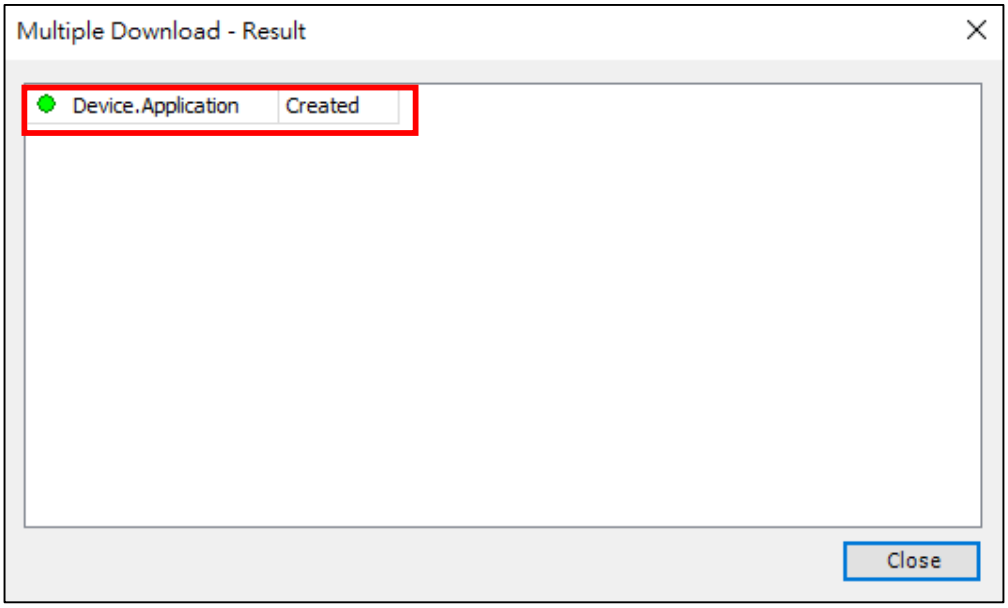
8. After adding, double-click AS02LC_A on the device.



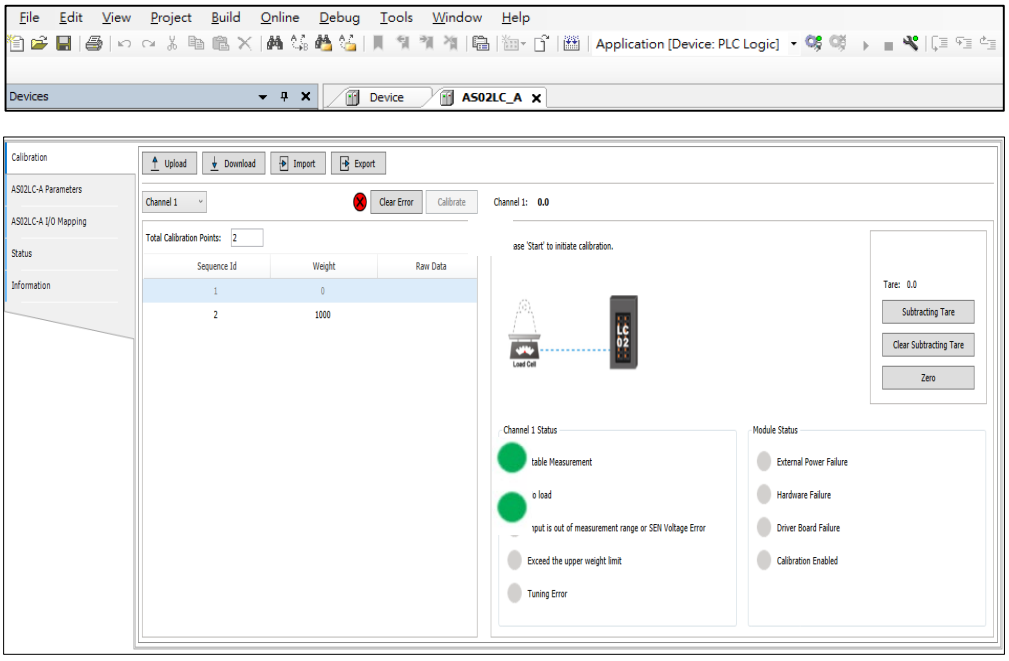
9. From Toolbar, select **Online** and click **Multiple Download**. Select the item you want to download in the **Multiple Download** window and click **OK**. Windows display as shown in the figures given below.







1. Click for **online configuration mode** on the toolbar to enable it.

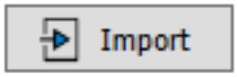
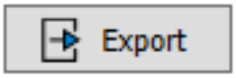


Result:

The Calibration Wizard window appears.

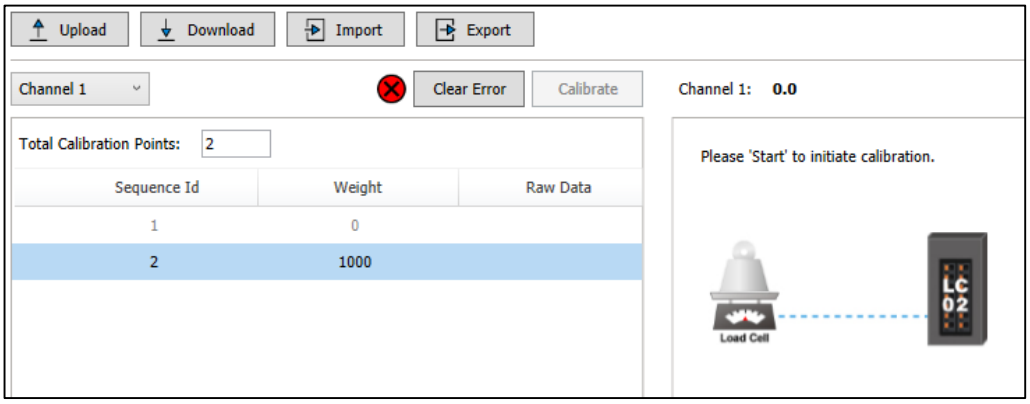
Correction Wizard Window Toolbar Description:

Toolbar	Description
Upload	Upload the parameters of the controller to the software.
Download	Download parameters to the controller.

Toolbar	Description
	Import the original parameters. **Note**: The file format must be .cal.
	Export and save device parameters. **Note:** The file format must be .cal.

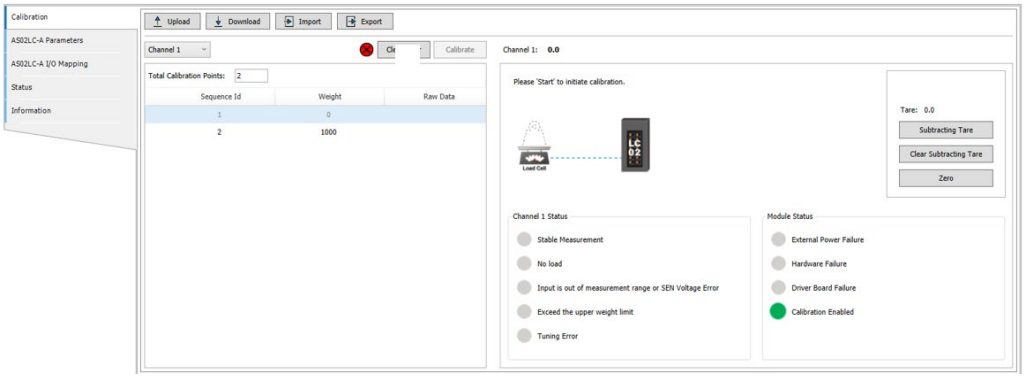
Follow the steps for the calibration process:

1. Set the **number of calibration points** and the **weight** of calibration points for **channel 1/channel 2**. The example below is a two-point adjustment, the first point is 0 g and the second point is 425 g.



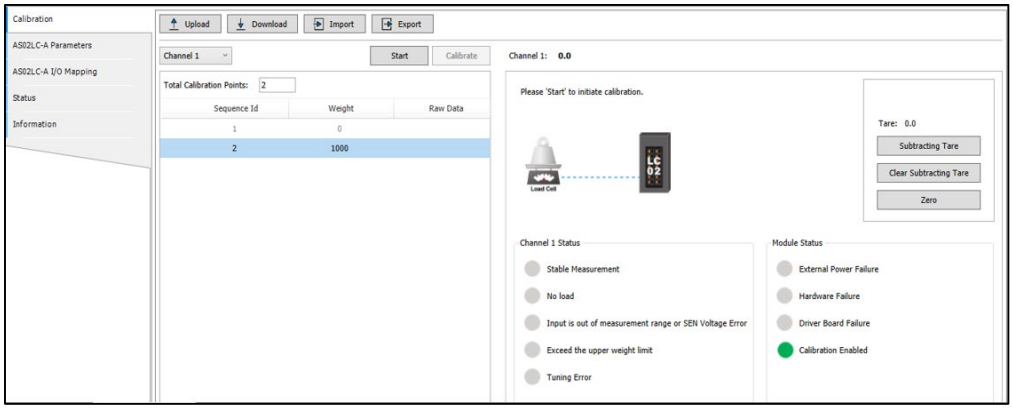
2. Select the first point, keep the LC sensor unloaded and do not put any weight. click **Start**, and then click **Calibration**.

   Result: Get the first point of **raw data**



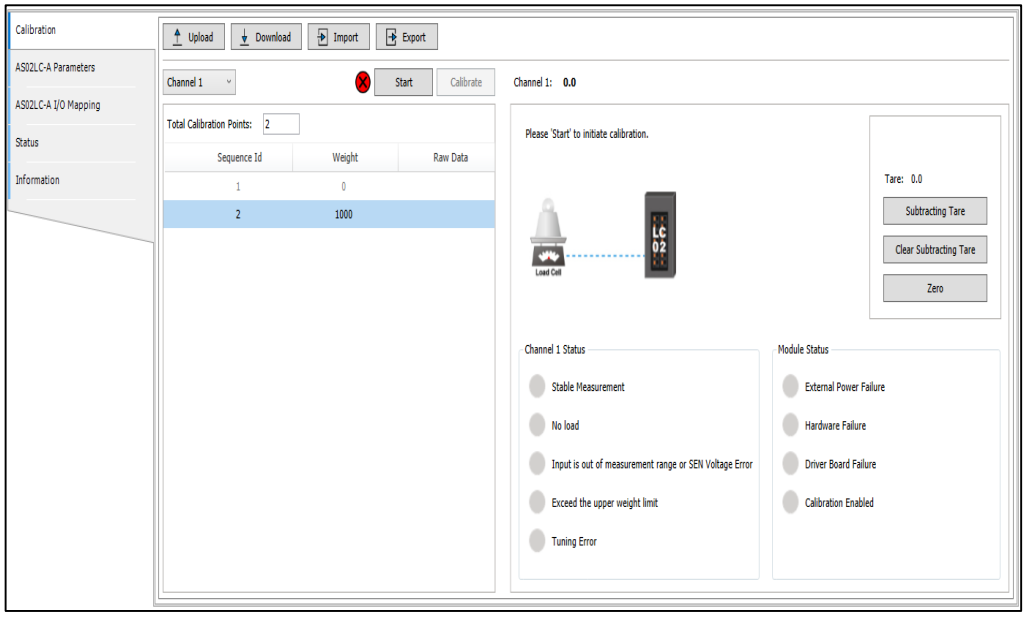
3. Select the second point and place the actual weight corresponding to the second point on the LC sensor. In this example, a weight of 425 g is placed. Click **Calibration** (if it is multi-point calibration, repeat this step several times).

   Result: Get the second point **raw data**.



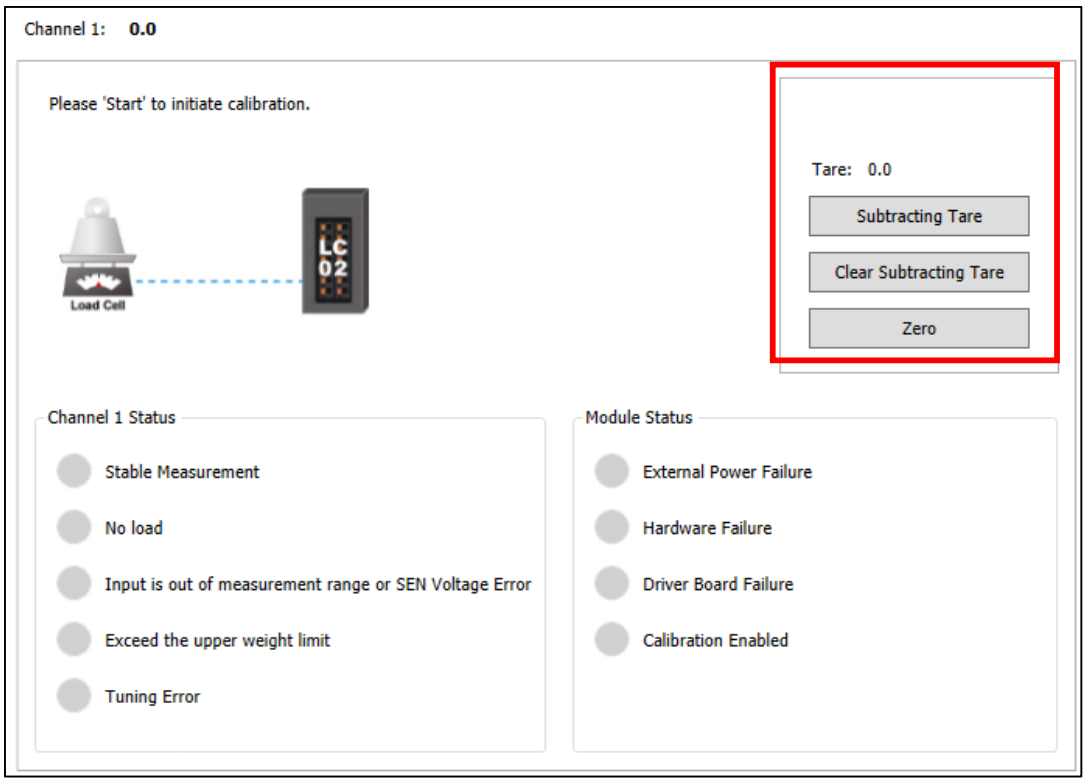
4. Click **Stop**.

   Result: The weight correction is completed, as shown in the following figure.



LC Calibration Wizard Measurement Setup Function

There are measurement settings fields on the right side of the **AS02LC calibration** page, which can perform **Subtracting Tare, Clear Subtracting Tare**, and **Zero** according to the measurement requirements.

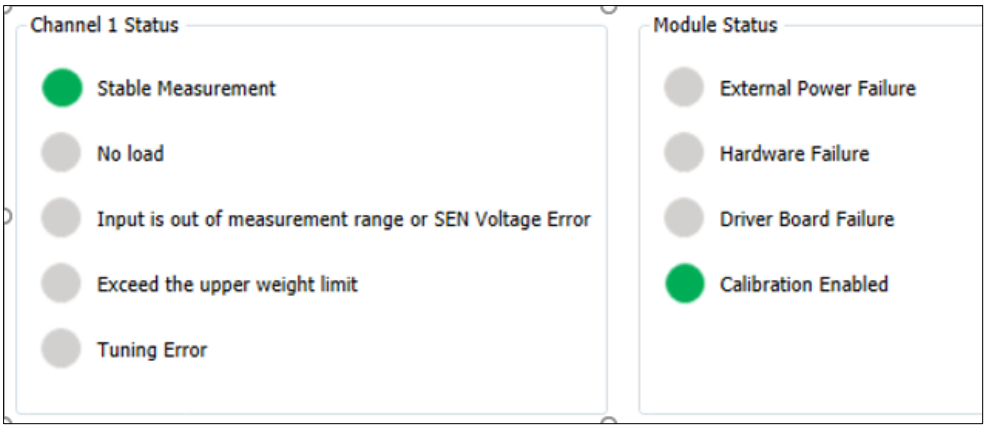


The LC Calibration Wizard measurement setup functions are as follows:

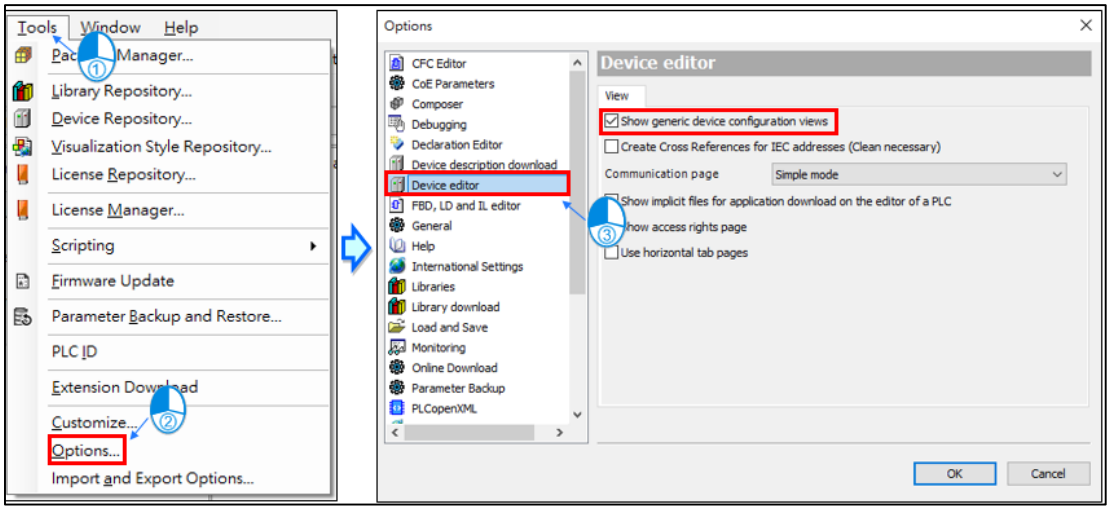
Function	Description
Subtracting Tare	After clicking, the weight of the item currently placed on the channel will be removed (continuous peeling is possible).
Clear Subtracting Tare	After clicking, the tare weight recorded in the current channel will be restored.
Zero	After clicking, the display value of the current channel will be reset to 0.

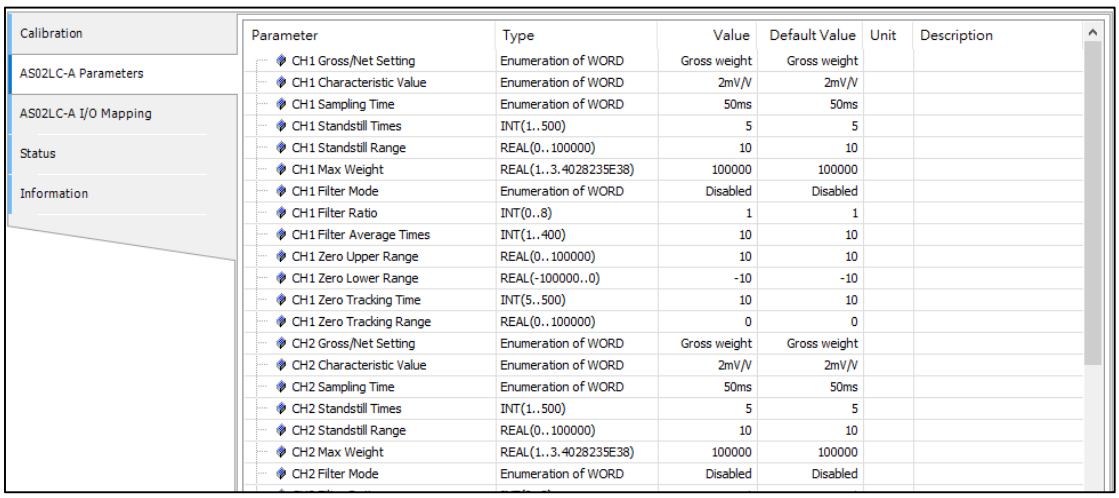
LC Calibration Wizard Status Monitoring

After the AS02LC module enters the calibration mode, you can view the status of the current scale/module.

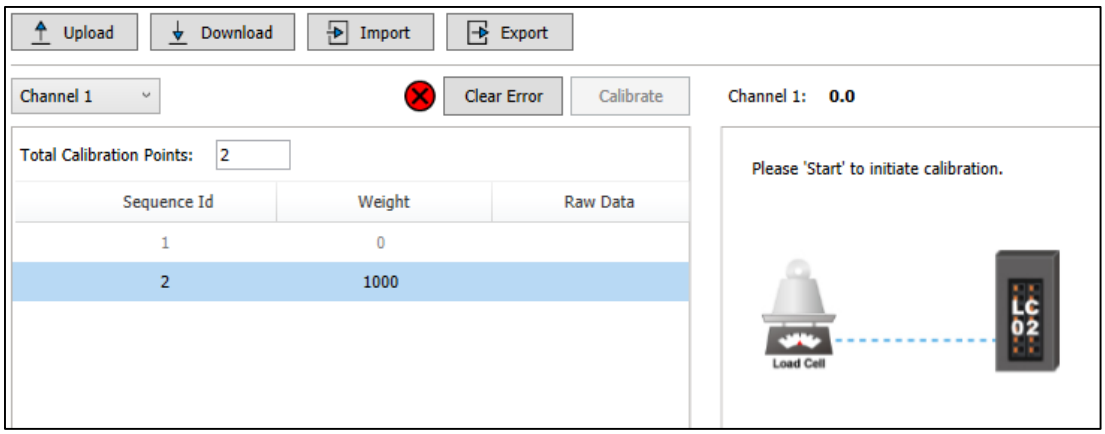


Note: To call the **AS02LC-A parameter** tab, go to main menu **Tools** > **Options** > **Device Editor** > check **Show generic device configuration views**





Parameter	Type	Value	Default Value	Unit	Description
CH1 Gross/Net Setting	Enumeration of WORD	Gross weight	Gross weight		
CH1 Characteristic Value	Enumeration of WORD	2mV/V	2mV/V		
CH1 Sampling Time	Enumeration of WORD	50ms	50ms		
CH1 Standstill Times	INT(1..500)	5	5		
CH1 Standstill Range	REAL(0..100000)	10	10		
CH1 Max Weight	REAL(1..3.4028235E38)	100000	100000		
CH1 Filter Mode	Enumeration of WORD	Disabled	Disabled		
CH1 Filter Ratio	INT(0..8)	1	1		
CH1 Filter Average Times	INT(1..400)	10	10		
CH1 Zero Upper Range	REAL(0..100000)	10	10		
CH1 Zero Lower Range	REAL(-100000..0)	-10	-10		
CH1 Zero Tracking Time	INT(5..500)	10	10		
CH1 Zero Tracking Range	REAL(0..100000)	0	0		
CH2 Gross/Net Setting	Enumeration of WORD	Gross weight	Gross weight		
CH2 Characteristic Value	Enumeration of WORD	2mV/V	2mV/V		
CH2 Sampling Time	Enumeration of WORD	50ms	50ms		
CH2 Standstill Times	INT(1..500)	5	5		
CH2 Standstill Range	REAL(0..100000)	10	10		
CH2 Max Weight	REAL(1..3.4028235E38)	100000	100000		
CH2 Filter Mode	Enumeration of WORD	Disabled	Disabled		

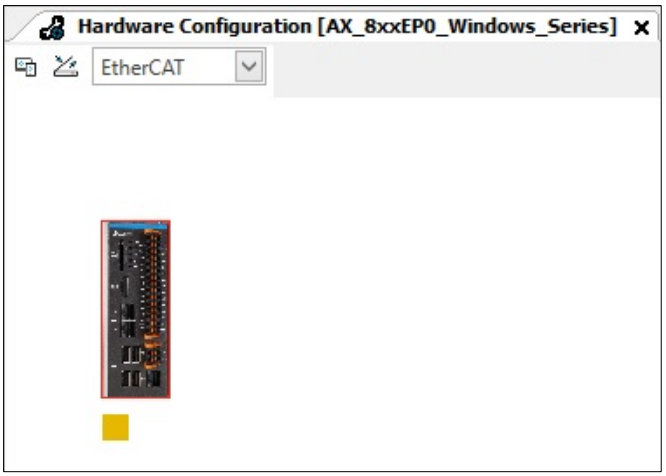

Upload
Download
Import
Export
Channel 1
Clear Error
Calibrate
Channel 1: 0.0
Total Calibration Points: 2
Sequence Id
Weight
Raw Data
1
0
2
1000
Please 'Start' to initiate calibration.
Load Cell
LC
02

5.1.2 AX-8xxEP0 series configuration:

This section applies to AX-8xxEP0 Linux series and AX-8xxEP0 Windows series.

AX-8xxEP0 Windows series as an example.

The AX-8xxEP0 series PAC does not support the expansion of modules. User can add only RIO modules in **Network Configuration** in the case of AX-8xxEP0 series.

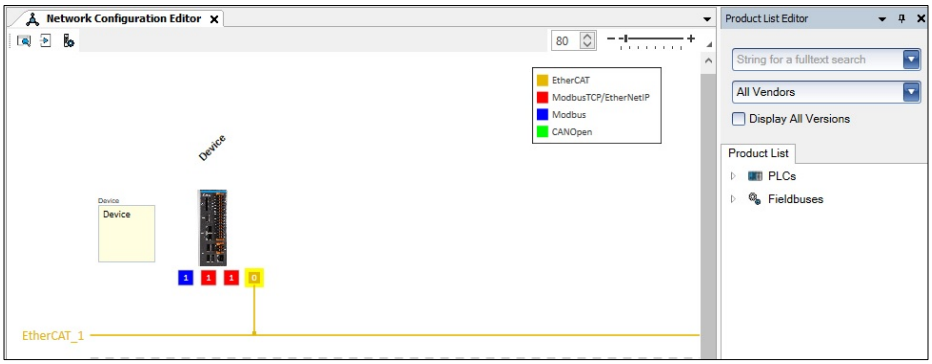


Follow the steps to create Hardware Configuration for AX-8xxEP0 series:

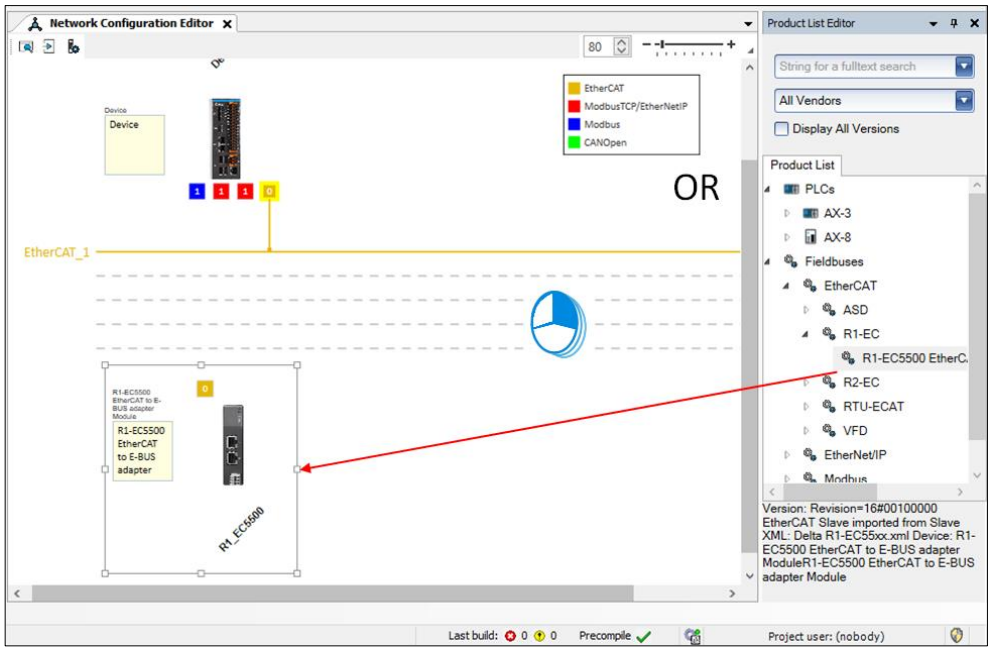
1. Double-click the **Network Configuration** in the **Devices** tree.

   Result: The **Network Configuration** window displays with

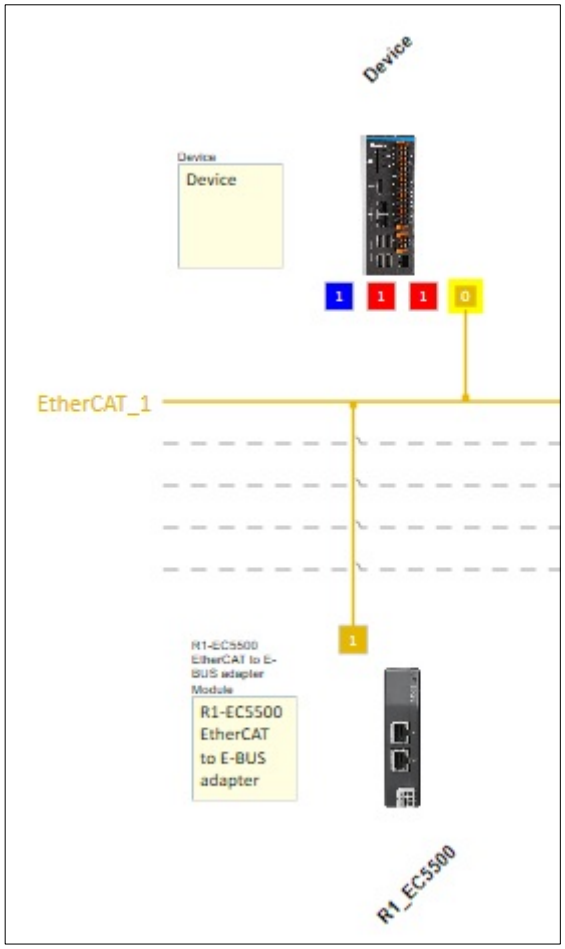
   - A **Product List Editor** pane to the right of the **Hardware Configuration** window.



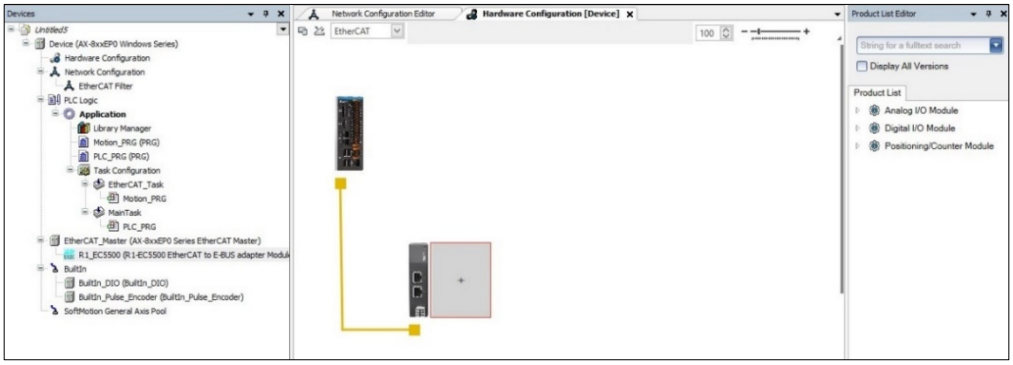
2. Expand the product list on the right side of the screen: **Fieldbuses** > **EtherCAT** > **R1-EC** > **R1-EC5500 EtherCAT to E-BUS adapter** Module. Double-click R1-EC5500 or drag R1-EC5500 to enter the **Network Configuration Editor**.



3. Connect this R1-EC5500 to the yellow EtherCAT line.



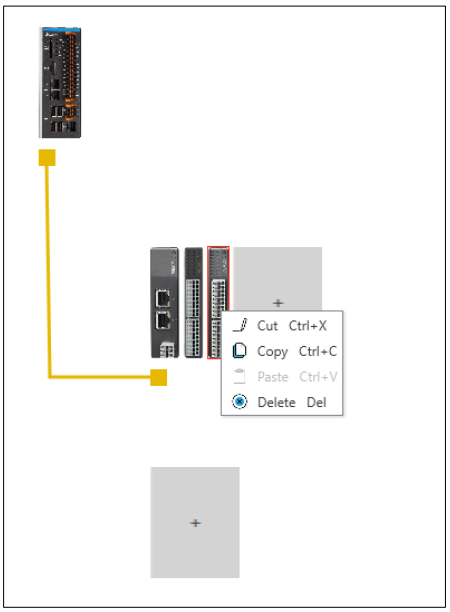
4. Double-click the **Hardware Configuration** in the **Devices** tree, user can see the newly added R1-EC5500. Users can click the reserved position (+) on the right to add RIO modules, as shown in the following figure.



Device Context Menu of RIO-R1 EC series

The device context menu provides the option to cut, copy, paste or delete the module.

Function	Description
Cut	Cut the module and paste to other place
Copy	Copy the module and paste to other place
Paste	Paste the cut or copy module
Delete	Delete the selected module



5.1.3 Firmware update of device

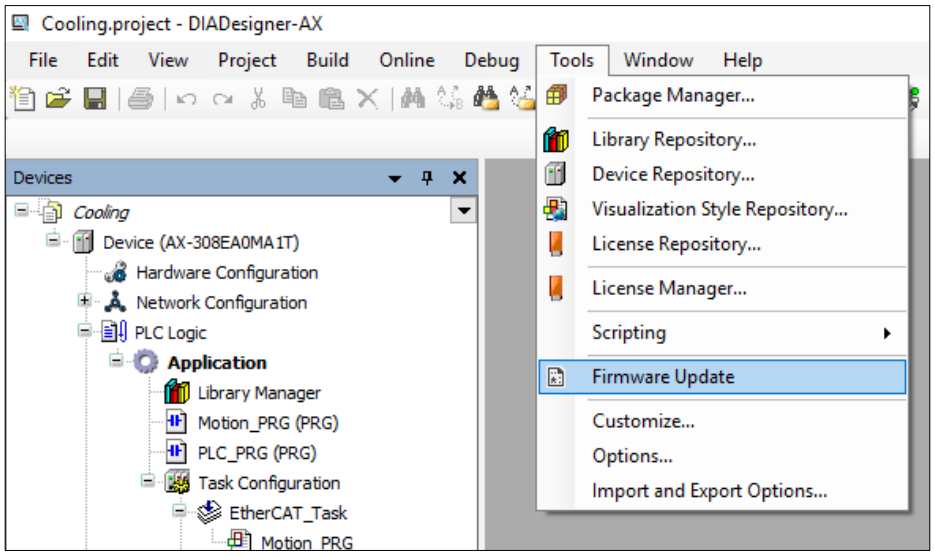
With the update of the host and module versions, different controllers may have different available functions and parameters. DIADesigner-AX provides a device description file that defines functions and parameters; when the software is installed, the latest version of the device description file will be installed into the system. The version of the device description file will include all the firmware version information. When there is a need for new functions or parameters, the firmware version of the device needs to be updated. The firmware of the device can be updated through the **firmware update** function supported by DIADesigner-AX version.

Device series currently supporting **firmware update**:

- AX-8xxEP0 Linux SM series
- AX-8xxEP0 Linux SM-CNC series
- AX-300NA0PA1
- AX-304ELA0PA1P
- AX-304ELA0PA1T
- AX-308EA0MA1T
- AX-308EA0MA1P
- AX-316EA0MA1T
- AX-324NA0PA1P
- AX-364ELA0MA1T

Follow the steps below to perform a firmware update.

1. Click **Tools** > **Firmware Update** on the toolbar menu.



Result: The **Firmware Update** window appears.

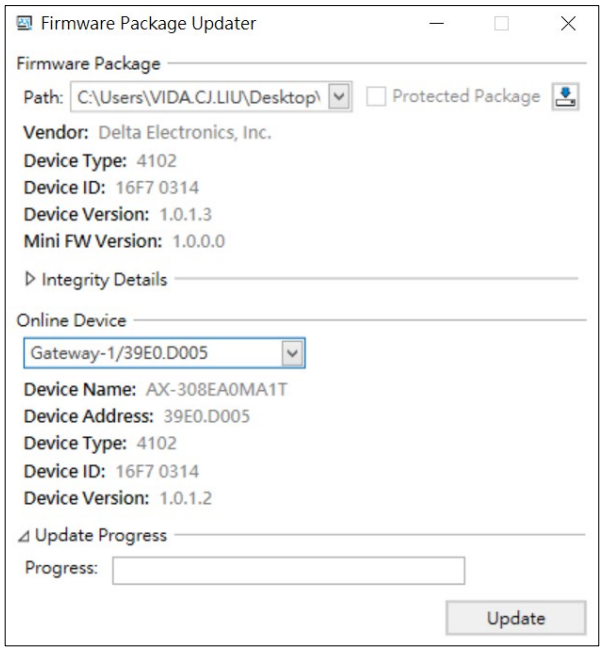
2. Click **Browse** in the **Path** drop-down menu to select the firmware package that needs to be updated

   Or

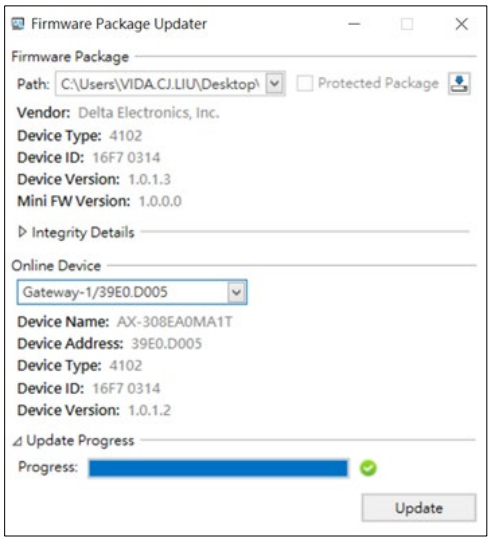
   Click **Online Download** icon  to connect to the Delta server to download and update the Delta controller firmware.

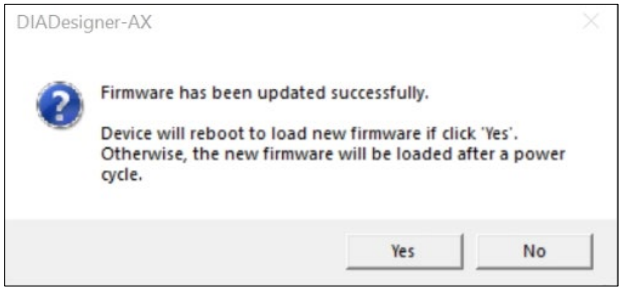
Result: The user can download/select the desired firmware in the online download window.

3. Select the updated device in the **Online Device**.



After confirming that the device type of the hardware matches the ID, click **Update** to start the firmware update.





The following table explains the **Firmware Update** window options:

Features	Description
Firmware Package	
Path	Select the path where the firmware is stored. After selecting the firmware file that needs to be updated, the manufacturer/device type code/device ID/device version number information will be displayed automatically.
Vendor	Firmware package manufacturer
Device Type	Firmware package device type number
Device ID	Firmware package device ID
Device Version	Firmware package device version number
Mini FM Version	The firmware package is compatible with the minimum hardware version that can be updated.
 Online download	After clicking the online download icon, user can download the controller-related firmware. User can select Global or China from the server drop-down menu.
Integrity Details	
Signature verification	Verify the firmware signature file, it will be automatically verified after the firmware path is selected.
Hash value verification	Verify the firmware hash value, it will be automatically verified after the firmware path is selected.
Online Device	
Online device	After clicking **Browse** in the drop-down menu, select the controller to be updated in the drop-down window
Device Name	Controller name
Device Type	Controller device type number
Device ID	Controller device ID

Features	Description
Device Version	Controller device version number
Update Progress	
Progress	After clicking **Update**, the update progress will be displayed.

5.2 AX-3 series setting

The content in this chapter is applicable to the following models. The following will use AX-308EA0MA1T as an example of operation.

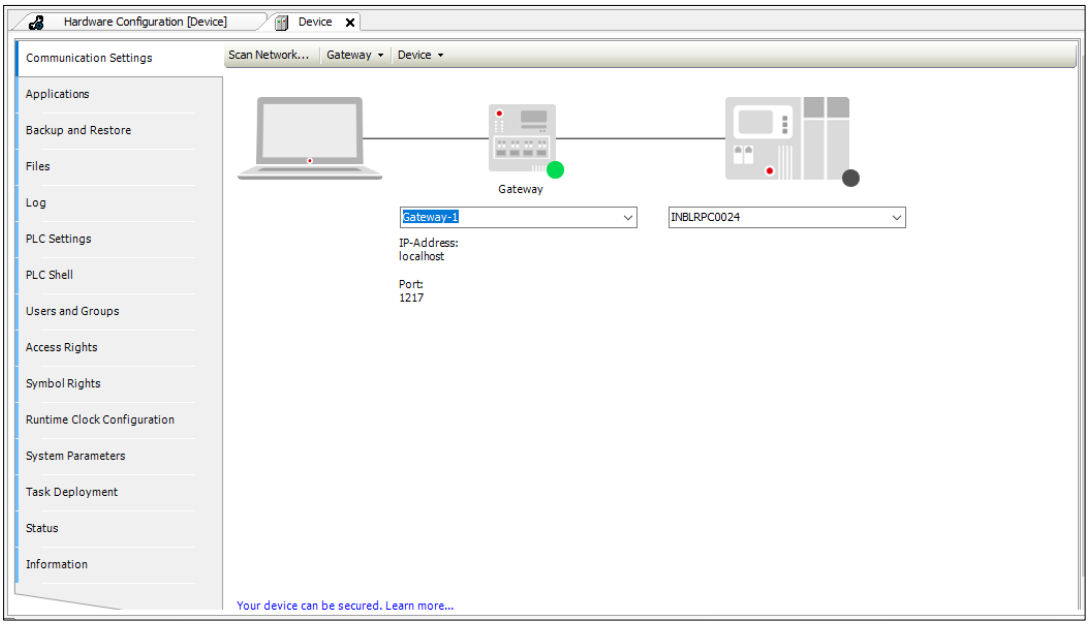
Motion PLC	Logic PLC
AX-308EA0MA1T, AX-308EA0MA1P, AX-364ELA0MA1T, AX-304ELA0PA1T, AX-304ELA0PA1P, AX-316EA0MA1T, AX-332EP0MB1T	AX-300NA0PA1, AX-324NA0PA1P

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

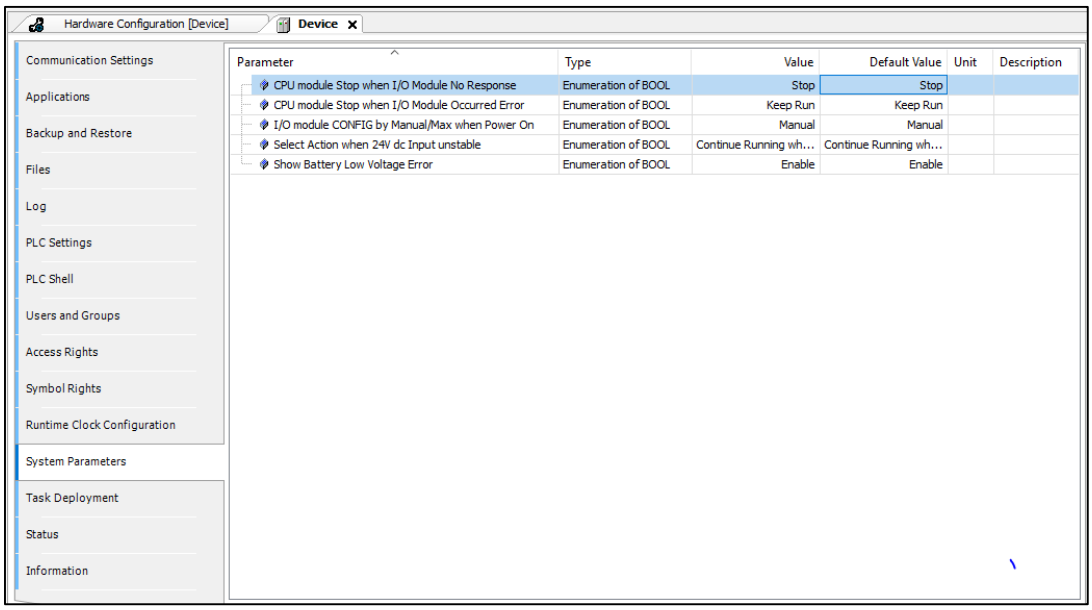
Before the performing Parameter Setting in a device, user have to refer to the Operation Manual. User have to make sure of the facts that affect the CPU module and the whole system, to prevent damage to the system or staff. Parameter Settings are not effective until they are downloaded to the CPU device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

5.2.1 System Parameter

Double-click on the **AX-308EA0MA1T** series to open the system-setting page. The **Communication Settings** tab displays by default as shown in the following figure.



Click the **System Parameters** tab. The device parameters are displayed as shown in the following figure.



The **System Parameters** tab displays the following parameters:

Item	Description
Parameter	Parameter name, not editable

Item	Description
Type	Data type of the parameter, not editable
Value	Initially displays the default value of the parameter, directly or the corresponding symbol name. Non-editable parameters are displayed in light gray. If the parameter is editable user can open an input field, a drop-down list or a file selection dialog with a double-click in the table field and use it to change the value.
Default value	Default value of the parameter defined by the device description, not editable
Unit	Unit of measure for the value (**Example**: ms for milliseconds; not editable).
Description	Short description of the parameter specified by the device description, not editable.

5.2.2 System Settings

On the **Devices** pane, double-click the device and go to **System Settings**.

Runtime Clock

Setting	Description
Read PLC Time	Read time information from the controller.
Date	Set the date.
Write PLC Time	Write the date to the controller.
Time Sync with Local Time	Set the time or sync with the local time.

Time Zone

Setting	Description
PLC Timezone Read Timezone	Read the timezone information from the controller.
Timezone Write Timezone	Set a new time zone and write back to the controller.

Network

Setting	Description
IP Address Mode	Select the IP address mode of Static or DHCP.
IP address	Set the IP address, subnet mask, and default gateway in Static mode.
DNS Server	Select **Obtain DNS server address automatically** or enter **Preferred DNS server** and **Alternate DNS server**.
Read from PLC	Read and update the IP address from the active device.

Project ID

Setting	Description
Set Project ID	Either set up a new project ID or synchronize the project ID with the online device.

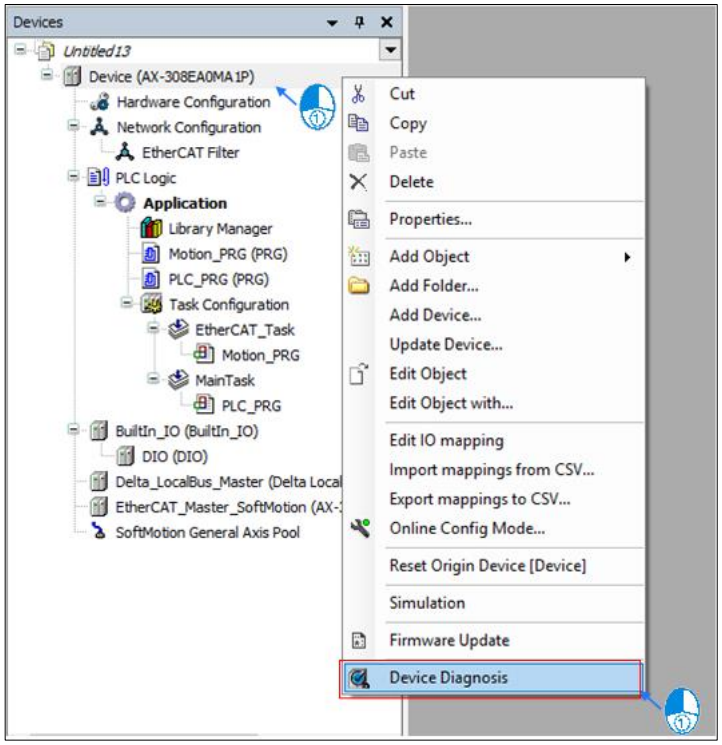
Retain Settings

Setting	Description
Original mode	Keep CODESYSpersistent variable mode.
%M mode	Set up the %M address to retain persistent variables. • **Start Memory Address**: Enter the value for the start memory address. The default value is 0 and cannot exceed the value in **End Memory Address**. • **End Memory Address**: The end memory address is 524287.
Clear	Clear all the %M addresses assigned to the variables.

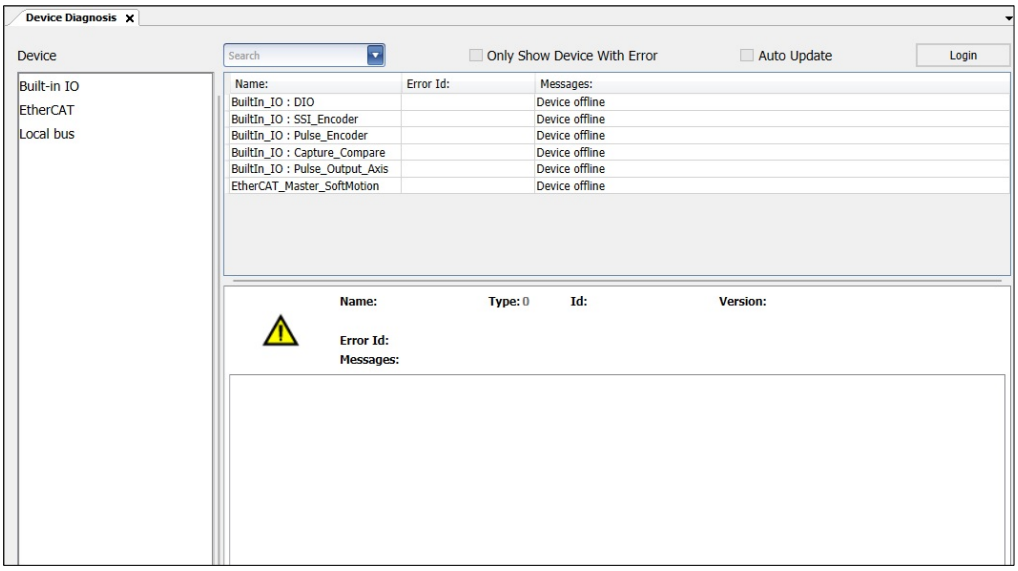
5.2.3 Device Diagnosis

When the AX-3 device is running, if there is an error, user can go to the **Device Diagnosis** to inquire about hardware & related errors.

1. Right Click **Device** > **Device Diagnosis**.



Result: The **Device Diagnosis** tab is displayed, as shown in the following figure.



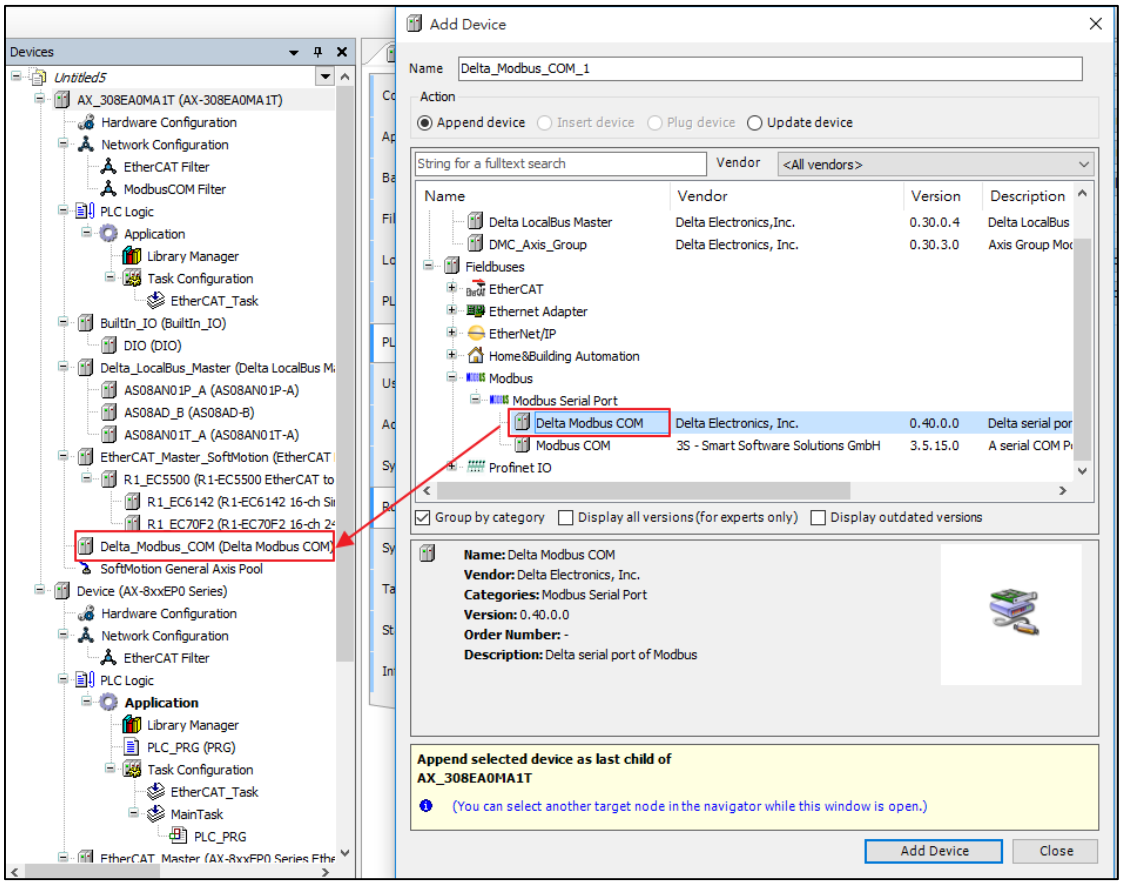
The functions in the **Device Diagnosis** tab are described in the table below.

Function	Description
Device	Device category in the project tree. **Example**: Click **Built-in IO** to display all devices under Built-in IO; double-click Built-in IO to display all devices in the project tree.
	Enter keywords to search for related equipment.
Only Show Device With Error	After checking, only devices with errors will be displayed in the window
Auto update	Automatically update device status
Update	Update the device status
Name	Display the device name. After clicking, detailed error messages will be displayed at the bottom of the window, such as the cause of the error alarm, inspection and treatment, and troubleshooting methods.
Error Code	Please refer to the corresponding manual for the device code: ● Built-In IO: Please refer to the AX-3 series operation manual - A.3 The troubleshooting chapter of the library. ● EtherCAT: please refer to the following 1. Delta ASDA-A2 Series Servo Drive User Manual 2. Delta ASDA-A3 Series Servo Drive User Manual 3. Delta ASDA-B3 Series Servo Drive User Manual 4. For non-Delta servos, please refer to the slave manual. ● Local Bus: Please refer to the following 1. AX-3 Series Operation Manual - A.5 CPU module error code corresponding to the lamp number and state description chapter. 2. The troubleshooting chapter of the AS module corresponding manual
Message	Display device status messages.

5.2.4 COM Port Setting

Right-click **AX-308EA0MA1T** Device and choose Add Device and select **Fieldbus** > **MODBUS** > **MODBUS Serial Port** to add Delta_MODBUS_COM Device.

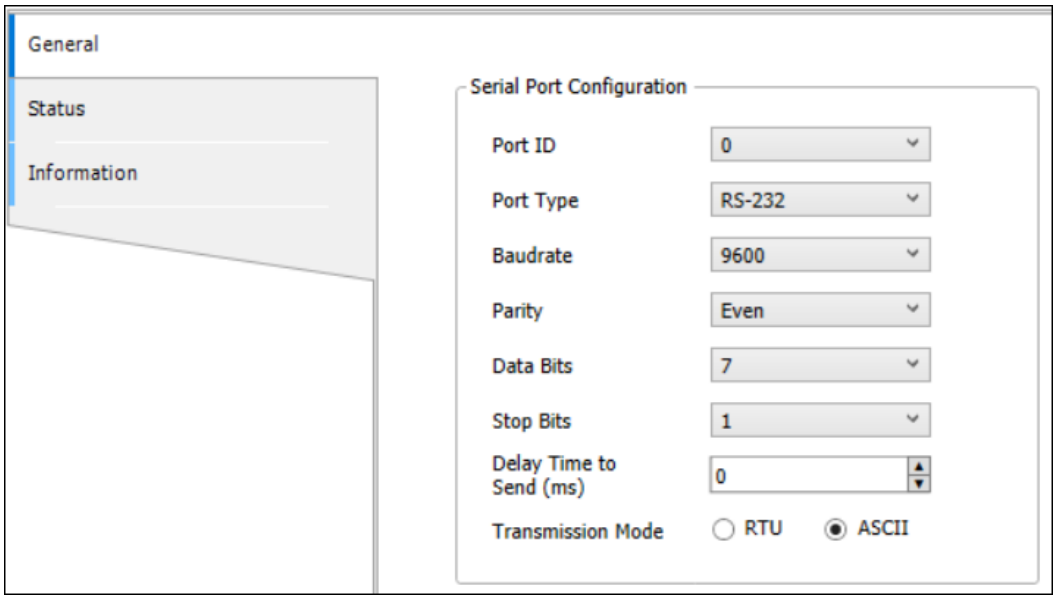
User can set the MODBUS serial port parameters of the controller on this fieldbus.



5.2.4.1 Delta MODBUS COM

User can set the serial port parameters of the controller on this page, such as selecting COM Port (RS-232/RS-485) and setting the communication format.

General

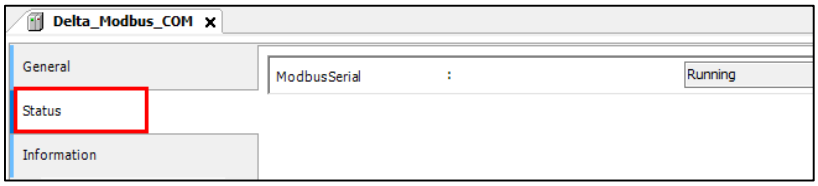


Functions in the **Delta_Modbus_COM – General** tab are described in the table.

Name	Description	
Port ID	Port ID	0/1 (Only shows for Delta Modus COM version 1.0.0.2)
COM Port	Communication interface	RS-232/RS-485
Baudrate	Tandem transmission rate	9600/19200/38400/57600/115200
Parity	Parity	None/Odd/Event
Data Bits	Data length	7/8 (when selecting RTU communication format, the length must be 8)
Stop Bits	-	1 bit/2bits
Send delay time	Delay the time to send the packet. Unit: milliseconds.	
Transmission Mode	Transmission mode	RTU/ASCII

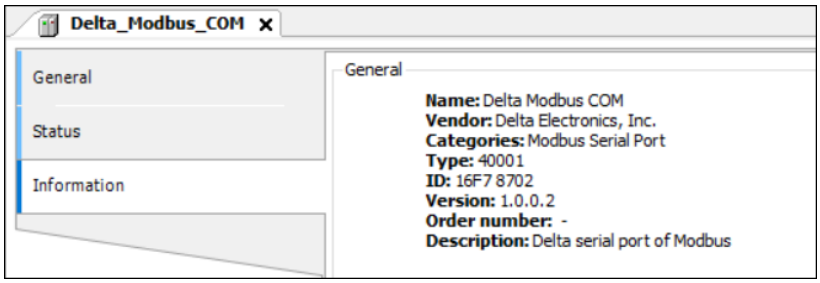
Status

User can monitor the status of MODBUS Serial port in the **Delta_Modbus_COM – Status** tab as shown in the following figure.



Information

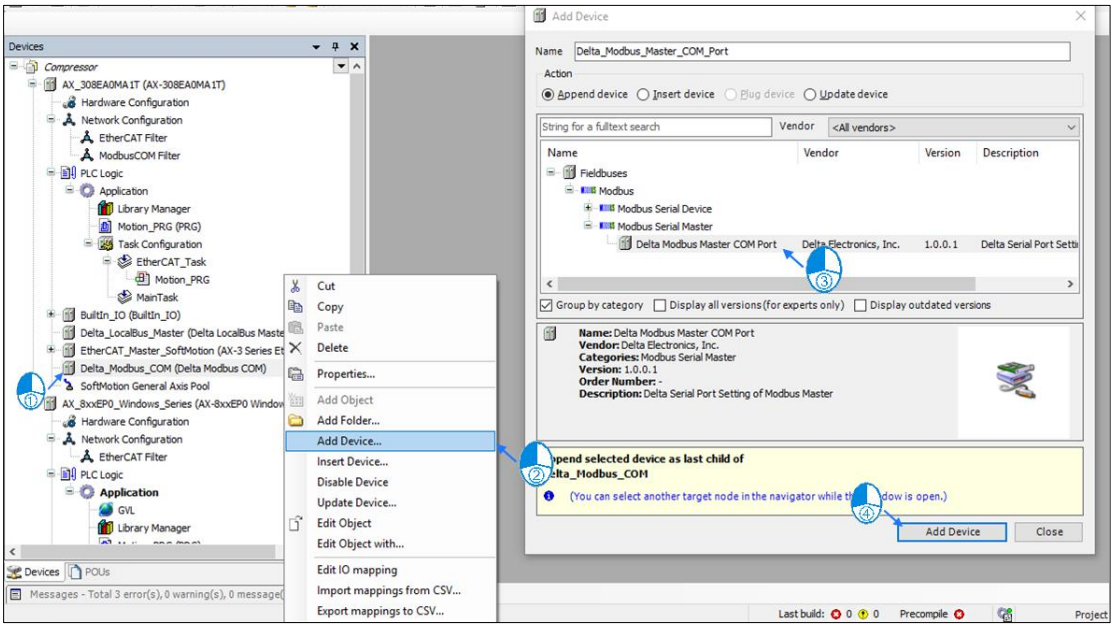
Delta_Modbus_COM – Information tab displays -- Device information, including Device ID, Type, Vendor etc.

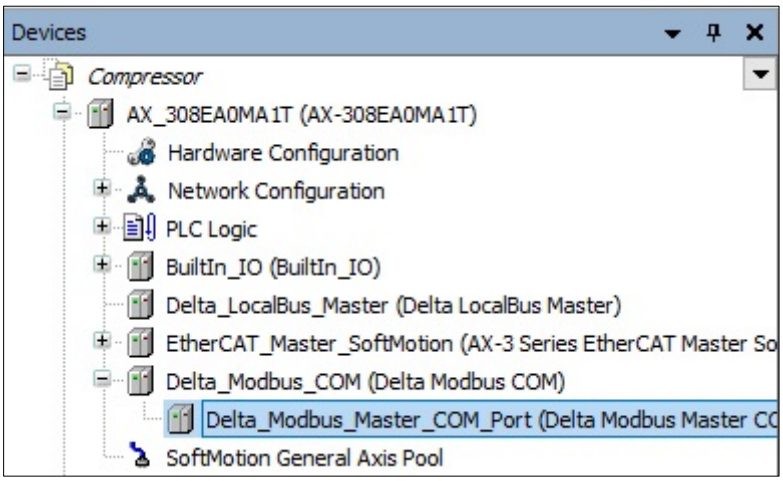


5.2.4.2 Delta MODBUS Master COM Port

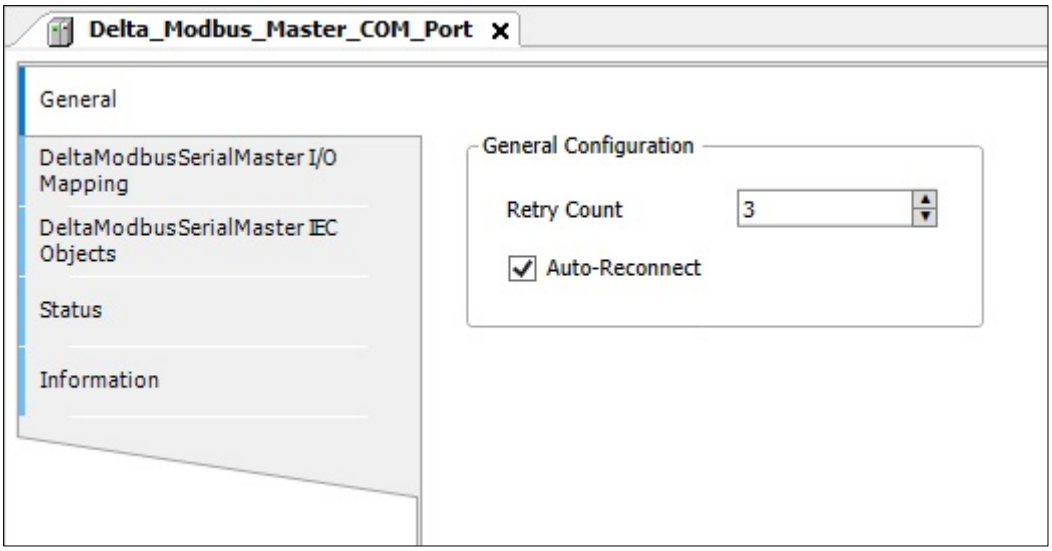
The Delta MODBUS Master COM Port function is used to set the entry when the AX-3 series controller is used as the MODBUS master station.

AX-308E series not only provide the MODBUS standard communication, but also covert the address of Delta PLC controller (X, M, D devices etc), which can save user looking up address table. Right-click **Delta MODBUS COM** and choose **Add device** in **MODBUS Serial Master** to add **Delta MODBUS Master COM port** to project tree.





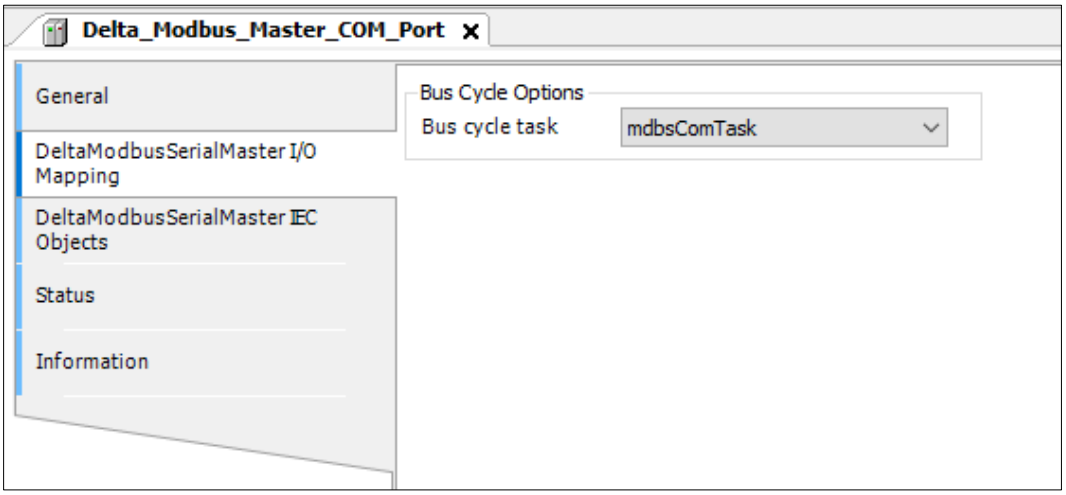
General



Name	Description
Retry Count	Set COM Port re-communication times.
Auto-Reconnect	After a communication error, CODESYS automatically confirms the error and attempts to continue executing the MODBUS command.

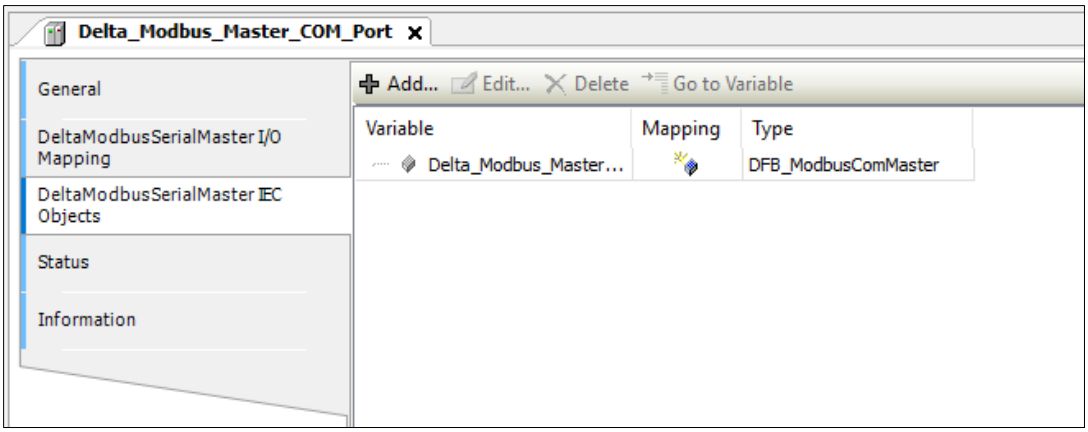
MODBUS Serial Master I/O Mapping

The user can set the MODbus Serial Master **Bus cycle task**. For more PLC settings, please refer to the Hardware Manual.



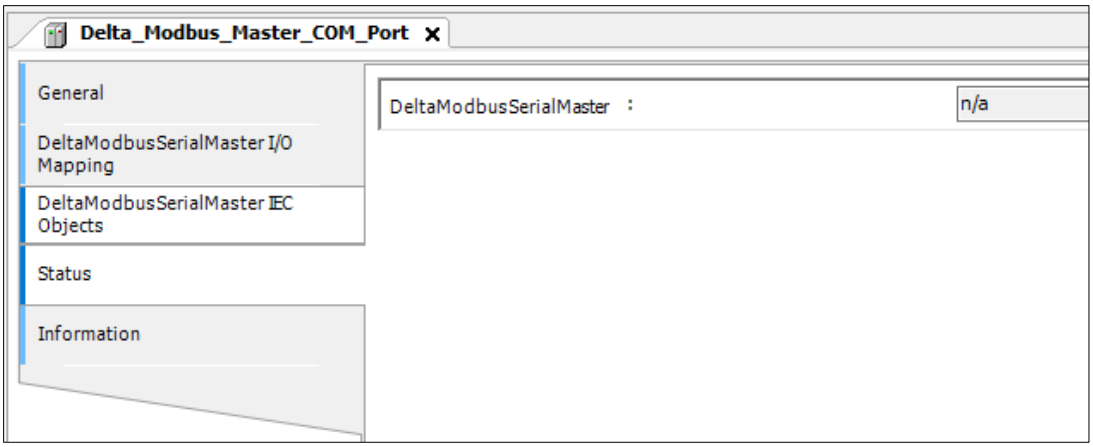
Delta MODBUS Serial Master IEC Objects

This tab shows the definition status of the device. The user can know the running status from the status.



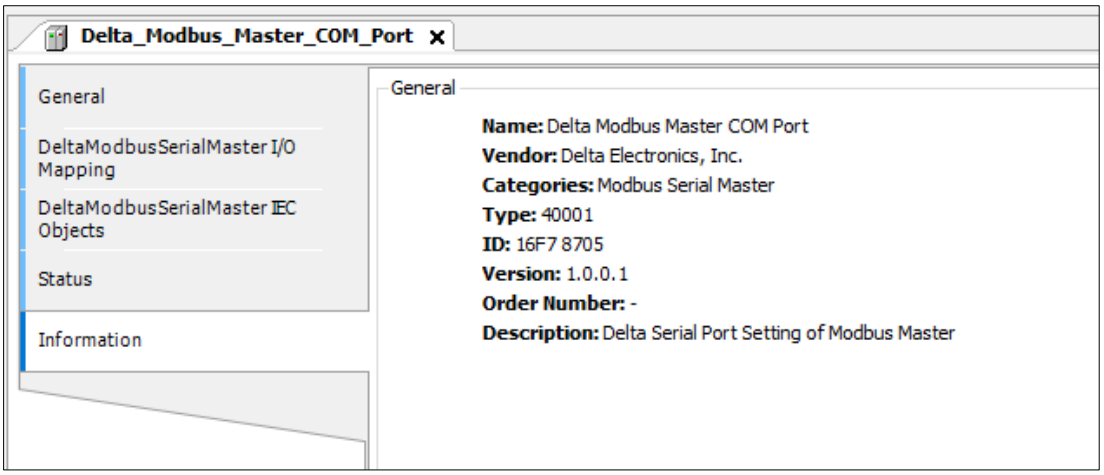
Status Tab

Display the execution status of MODBUS Serial Master.



Information tab

Display current device information, including device ID, type and manufacturer, and so on.



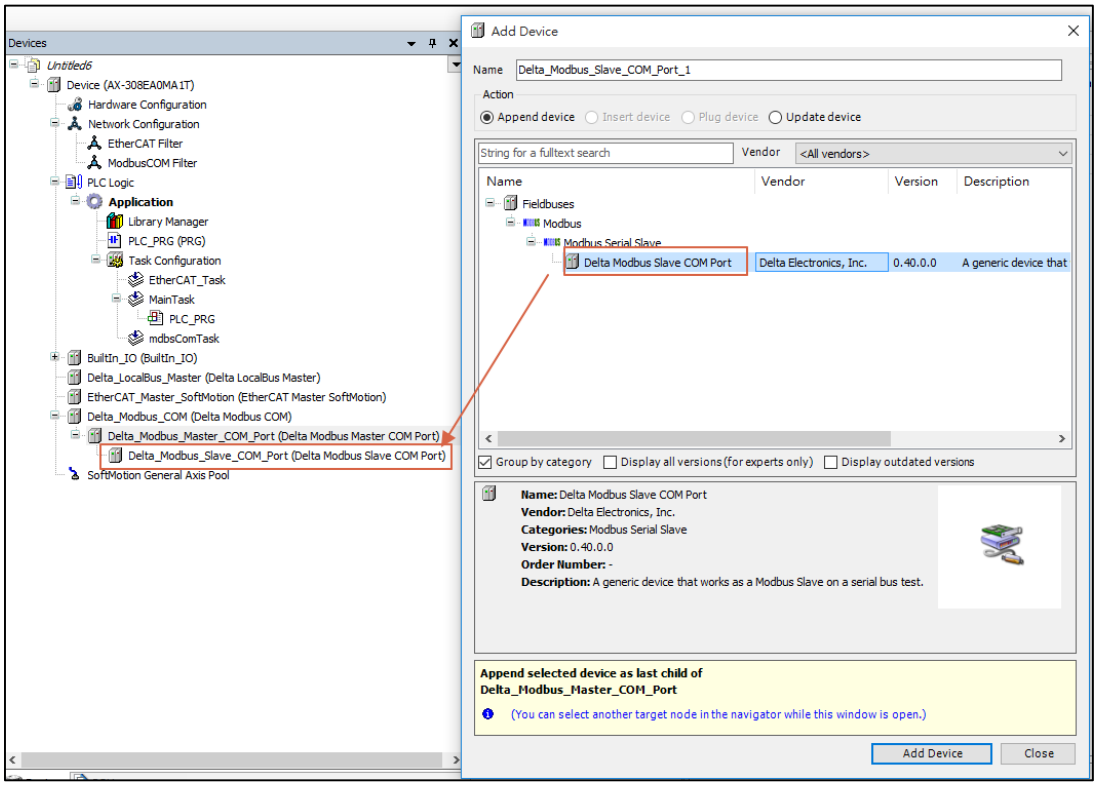
Note:

1. Each MODBUS Serial Port only can add one MODBUS Serial Master and each MODBUS Serial Master can add 32 MODBUS Slave COM Port.
2. User can add 32 MODBUS Slave COM Port with RS232, but RS232 supports only 1-to-1 communication. There is only first MODBUS Slave COM Port operation. RS485 is without this limitation.

5.2.4.3 Delta MODBUS Slave COM port

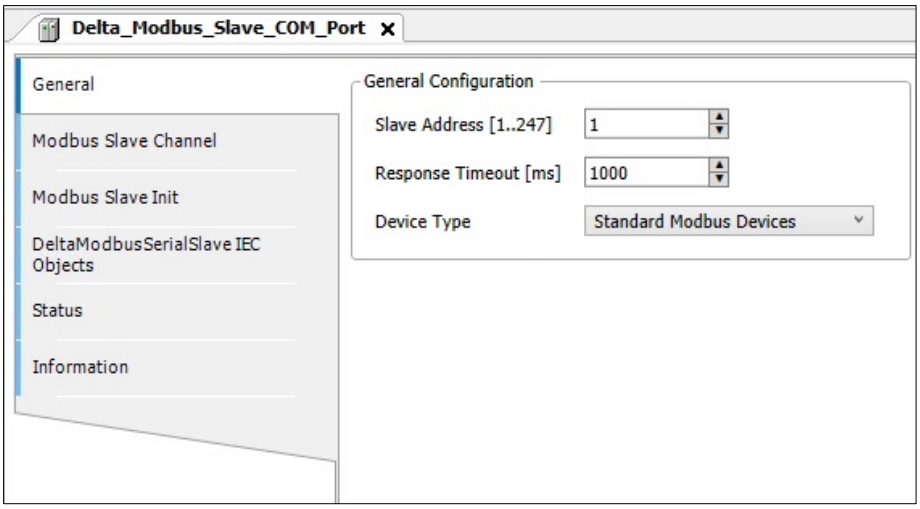
The Delta MODBUS Slave COM Port function is used to set the target slave station to be connected when the AX-3 series controller is used as the MODBUS master station.

User can right-click **Delta MODBUS Master COM Port** and choose **Add Device** to add **Delta MODBUS Slave COM port** under **Delta MODBUS Master COM Port**.



General:

User can set the MODBUS Slave COM Port related information such as station number, response timeout time, communication device format, etc. on this page.

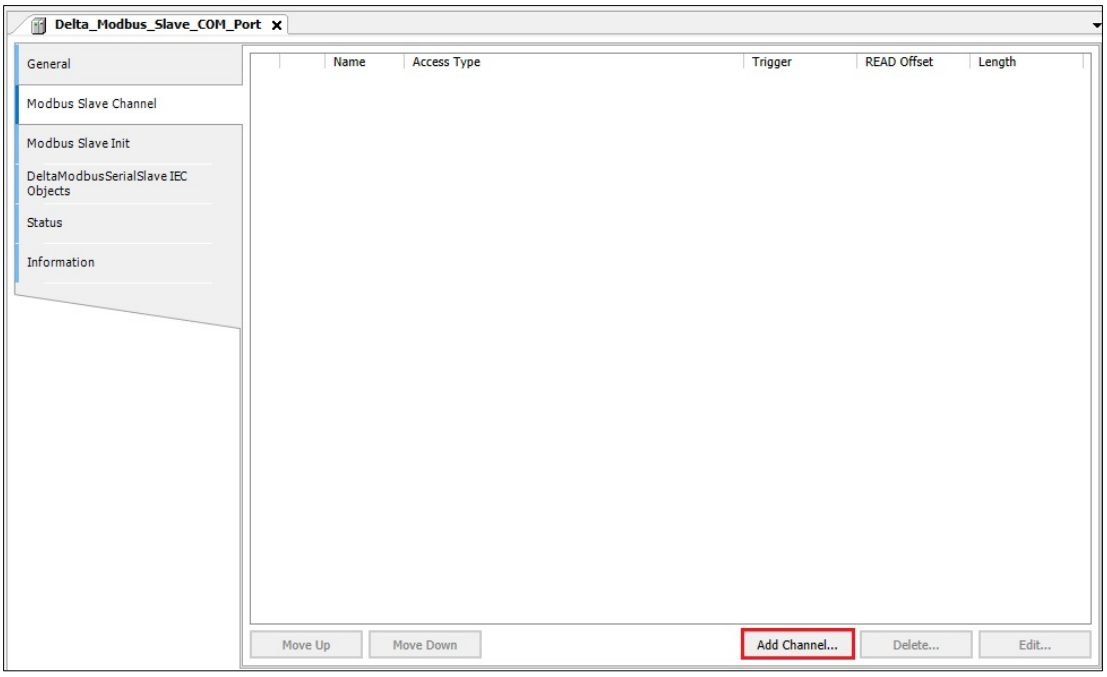


Name	Description
Slave Address	Slave station number.
Response Timeout	Slave response timeout.
Device Type	Slave type can choose standard MODBUS communication or Delta specific controller. (If Delta series controller is selected, the device and MODBUS address will be automatically converted to Delta controller address).

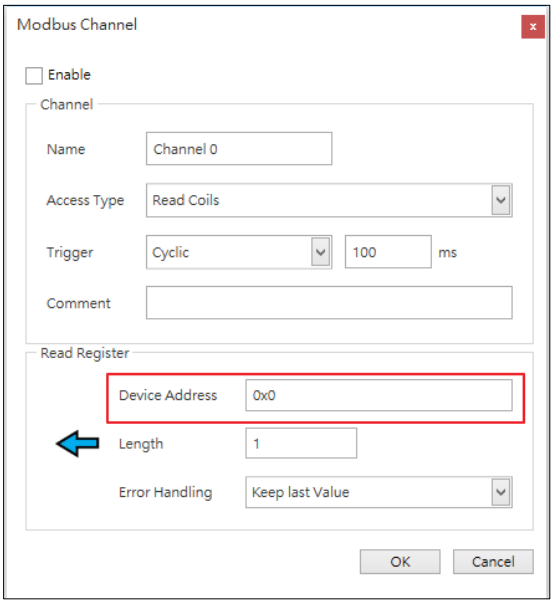
MODBUS slave channel tab

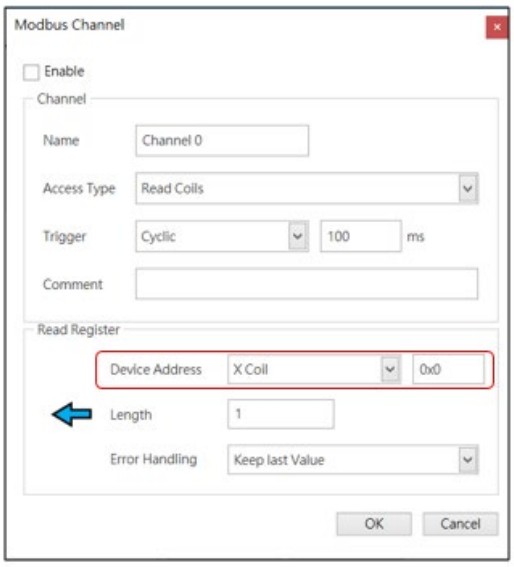
User can use this table to define the slave channels. Each channel represents a MODBUS request packet. At present, each slave station can establish a maximum of 10 packets.

The AX-3 series controller will send MODBUS request packets in order in the table, and all channels use a MODBUS connection in common.

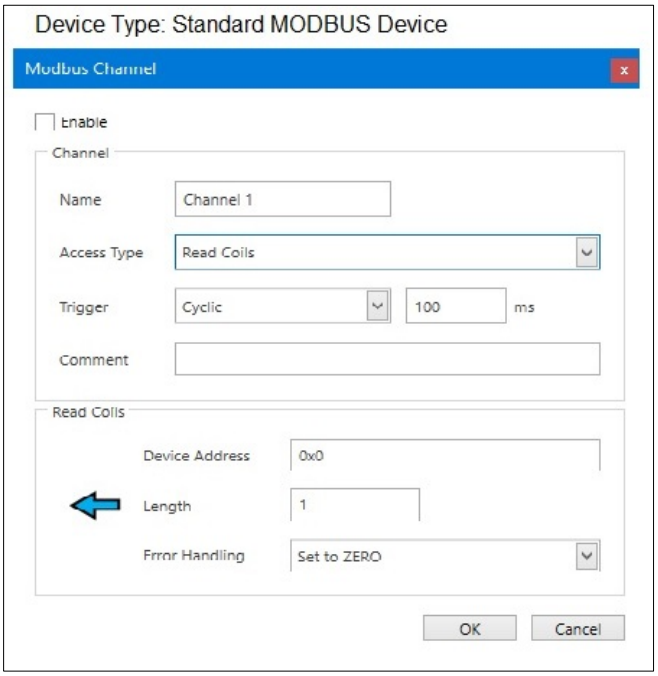


As shown above, there is an option to add channels in the MODBUS slave channel page.The MODBUS Channel will vary with the Device type set on the General page. If user choose a Delta specific controller, it will automatically convert the MODBUS address of the device, eliminating the need to look up



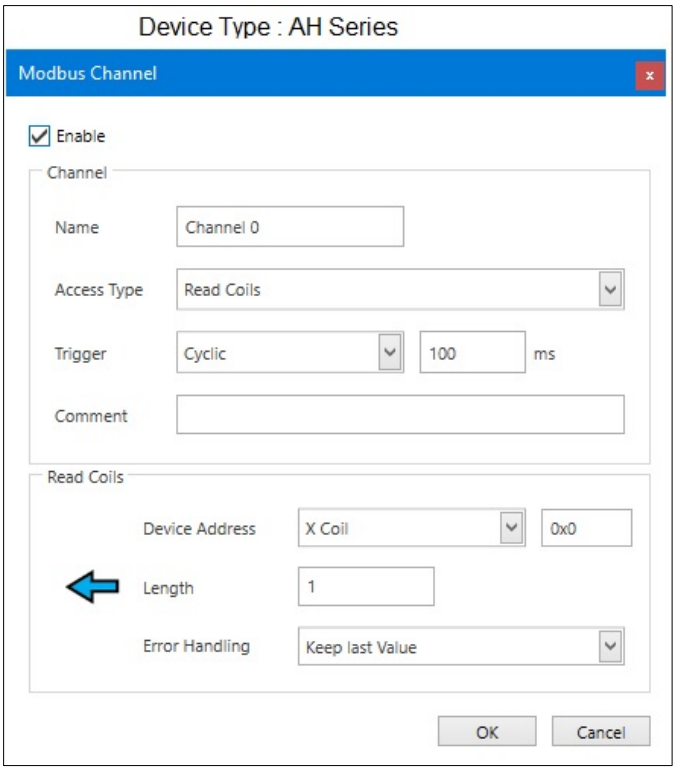


MODBUS Channel Description:



Name	Description
Enable	Decide whether this channel is effective.
Name	Define this channel name.
Access Type	MODBUS Communication function code: • Read coils (0x01). • Read discrete inputs (0x02).

Name	Description
	• Read holding registers (0x03). • Read input registers (0x04). • Read single coil (0x05). • Write single register (0x06). • Write multiple coils (0x0F). • Write multiple registers (0x10). • Read/Write multiple registers (0x17).
Trigger	Cyclic: MODBUS request is triggered according to the set cycle time Rising edge: MODBUS request is triggered by a Bollinger variable, which is defined on the I/O Mapping page. Application: MODBUS request can be triggered through the function block MODBUS Channel.
Comment	Channel annotations.
Device Address	Delta Series Controller address of Register.
Length	Length of data read/written (Maximum length is 100 coils and 100 registers).
Error Handing	When a communication error occurs, the setting action of Register data • Set To ZERO • Keep last value



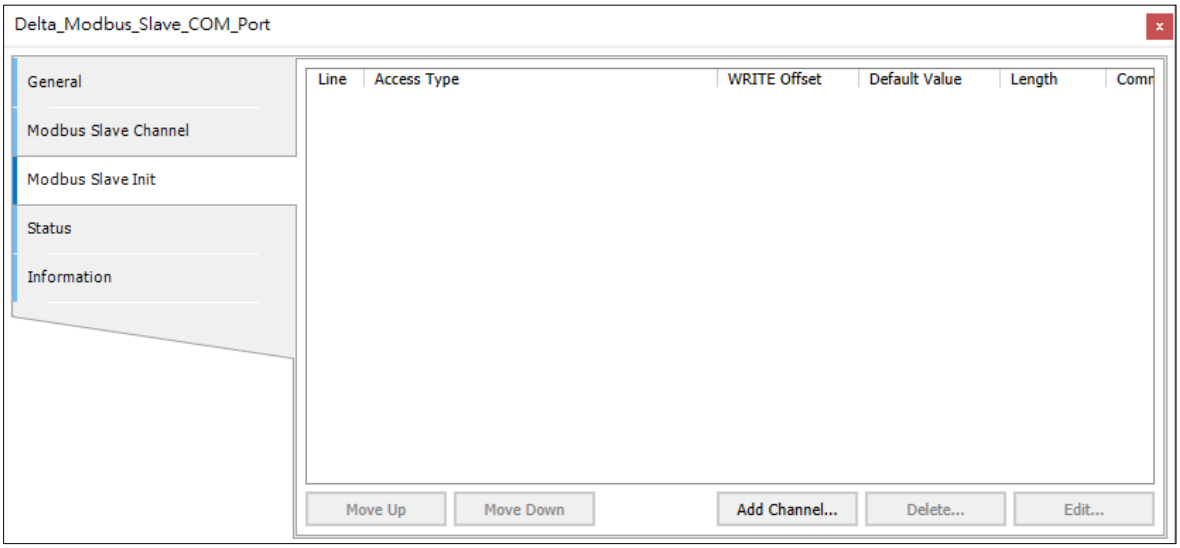
Name	Description
Enable	Decide whether this channel is effective.
Name	Define this channel name.
Access Type	Channel read and write actions: • Read coils • Read registers • Write coils • Write register **Note**: The controller will select the corresponding MODBUS function code according to the type of device being read/written.
Trigger	Cyclic: MODBUS request is triggered according to the set cycle time. Rising edge: MODBUS request is triggered by a Bollinger variable, which is defined on the I/O Mapping page.

Name	Description
	Application: MODBUS request can be triggered through the function block MODBUS Channel.
Comment	Channel annotations.
Device Address	Device Address of Delta Series Controller.
Length	Length of data read/written (Maximum length is 256 coils and 100 registers).
Error Handing	When a communication error occurs, the setting operation of the Registers data is performed. • Set To ZERO • Keep last value

MODBUS Slave Initialization tab

The Delta MODBUS Slave COM Port function is used to set the entry when the AX-3 series controller is used as a MODBUS slave.

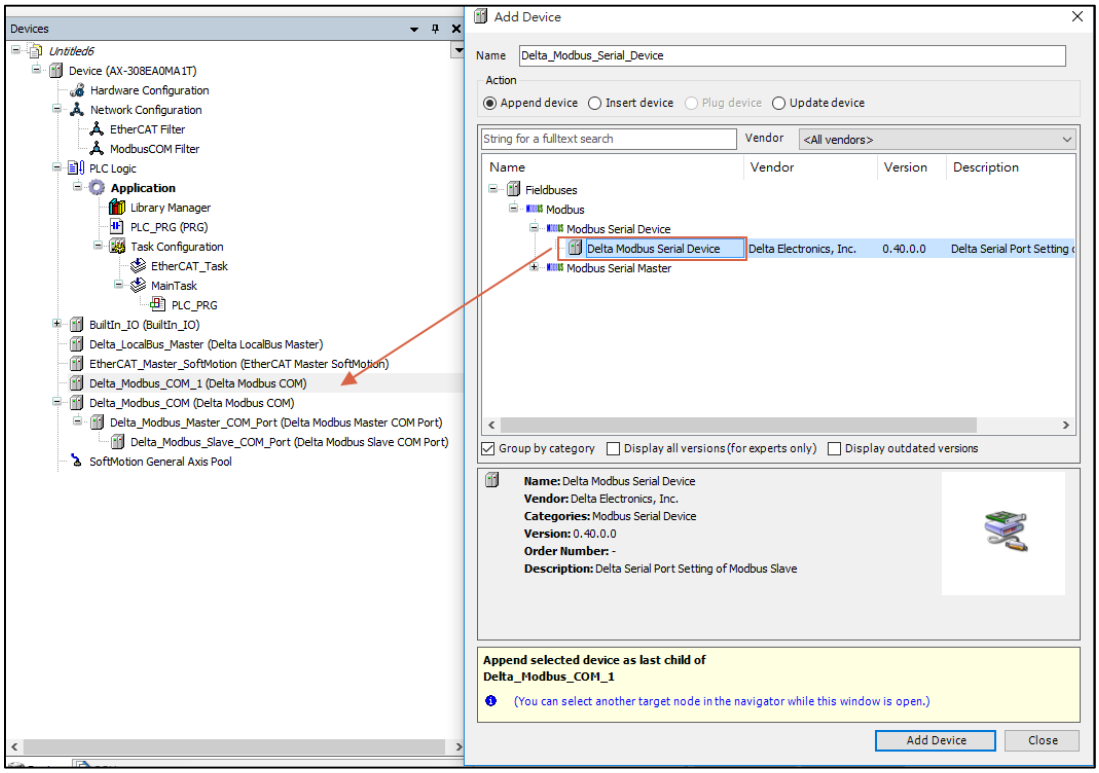
The user can use this table to decide whether to write the initial values to the slave's Coils/Registers after the AX-3 series controller successfully establishes a MODBUS connection with the slave.

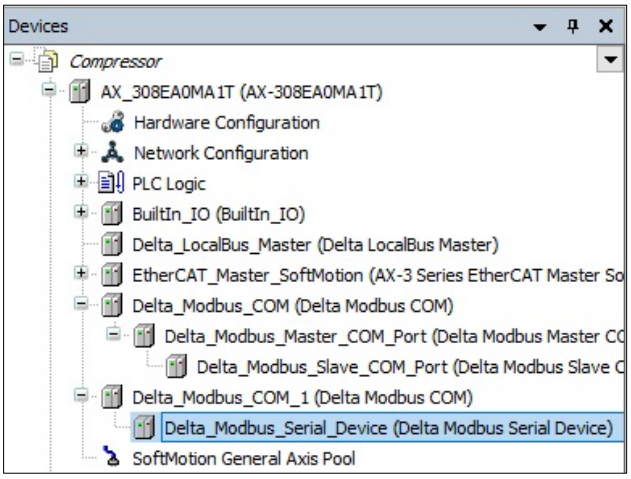


5.2.4.4 Delta MODBUS Serial Device

After completing the procedure *4.2.4.2 Delta MODBUS Master COM Port* to add **Delta MODBUS Master COM Port**. Right-click **AX-308EA0MA1T** > **Add Device**> **MODBUS serial Port** > **Delta MODBUS COM**, and the project tree displays **Delta_MODBUS_COM_1**.

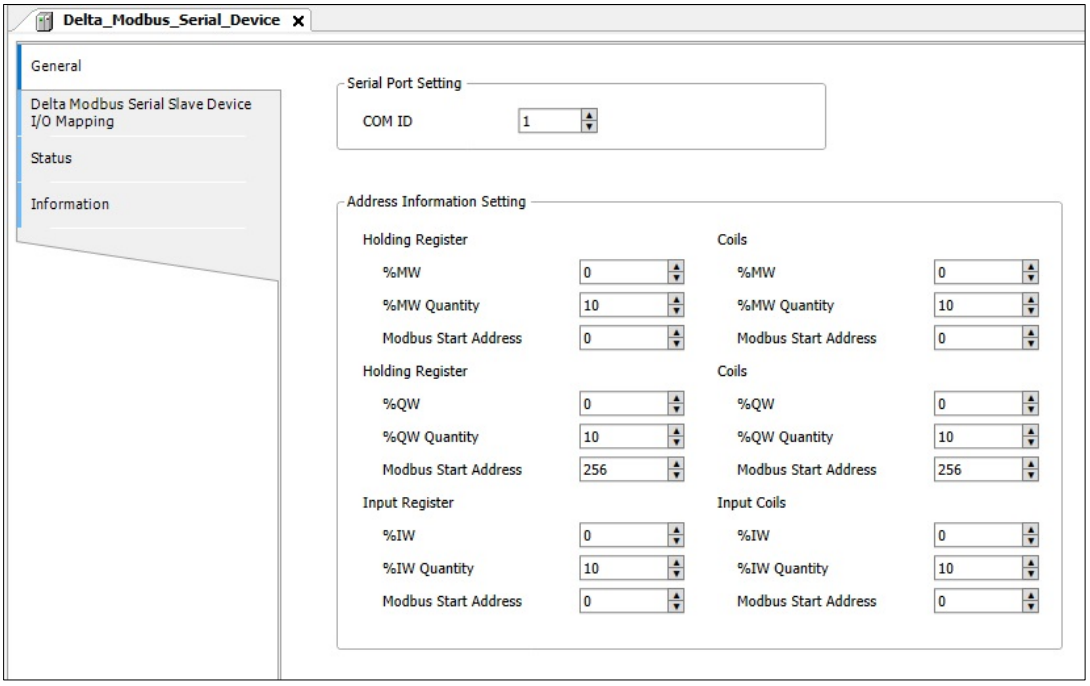
Right-click **Delta_MODBUS_COM_1** and select **Add device**, after that, select **Delta MODBUS Serial Device** under MODBUS Serial Device.





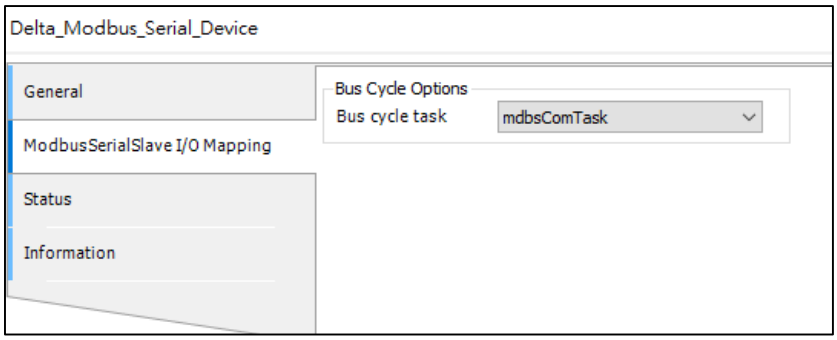
General

User can set the station number of the AX-3 series controller and the allowed Coils/Registers section on this page. If MODBUS Serial Master uses Delta specific communication protocol, all sections can be accessed without restrictions.



MODBUS Serial Slave I/O Mapping:

User can set **Bus Cycle Task** on MODBUS TCP Slave Device. For more information on the bus cycle task, please refer to AX-3 operation manual “PLC Settings”.

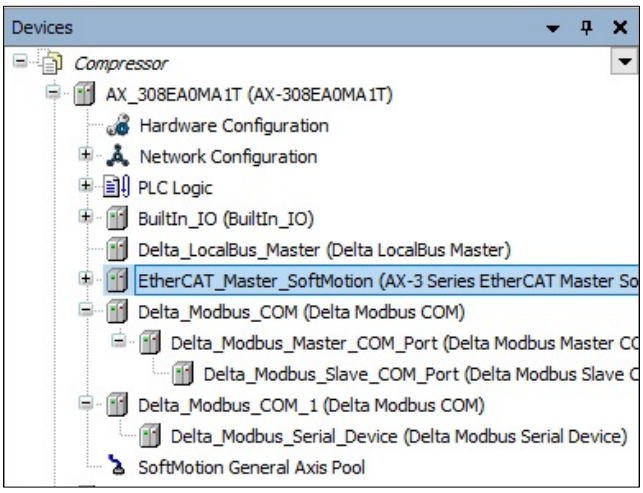


5.2.5 EtherCAT Setting

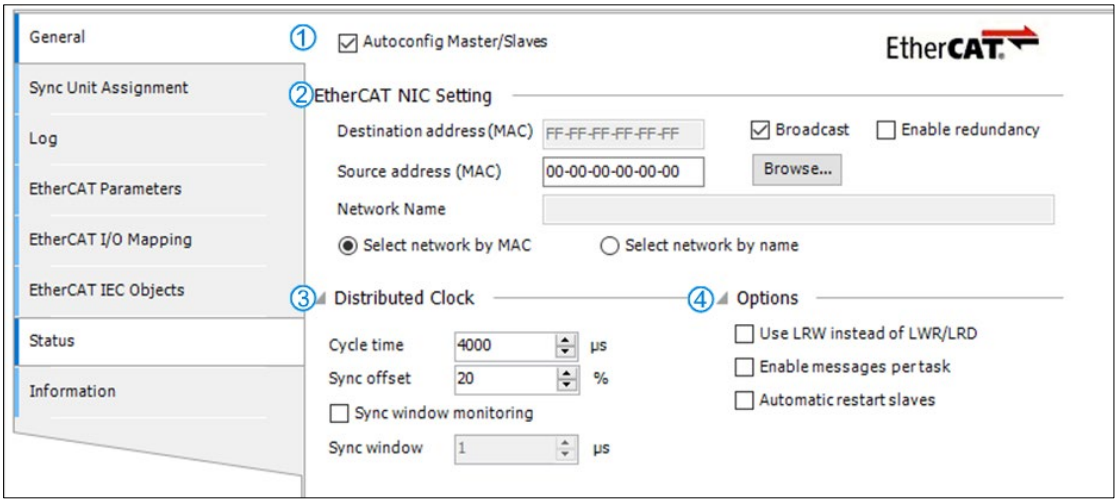
Note: The EtherCAT settings in this section are not applicable to AX-300NA0PA1 and AX-324NA0PA1P.

5.2.5.1 EtherCAT Master Setting

Click on the **Devices** > **EtherCAT_Master_SoftMotion**.



Result: The EtherCAT Master Setting window displays as shown in the following figure.



General

① Autoconfig Master/Slaves: This option activates the master for quick and easy basic configuration. It is generally recommended to check this setting.

② EtherCAT NIC Setting

- Destination address (MAC): EtherCAT MAC address of the device on the network that will receive the telegram.
- Source address (MAC): MAC address of PLC network interface (Use Browse to scan slaves, please select CPSW1 module).
- Network Name: The name or MAC of the network.
- Select Network by MAC: Net-route MAC-ID designation (The default name is CPSW1 * *Note* 1).
- Select network by name: Network routing network name identification, and the project has nothing to do with the device.

③ Distributed Clock

- Cycle time: The master station will send the corresponding data message to the slave station according to the cycle time.
- Sync offset sets the delay time of the PLC cycle time corresponding to the EtherCAT slave SYNC interrupt. For example, if the sync offset is set to 20%, the PLC cycle is delayed by 20% of the cycle time of the SYNC interrupt of the EtherCAT slave, which means that the PLC cycle still has 80% of the cycle time margin. If the PLC cycle is within this time margin, delayed without losing synced data.
- Sync window monitoring: When started, the synchronization of the slaves can be monitored.
- Sync window: Set the synchronization time range of the monitoring slaves.

④ Options (Click 

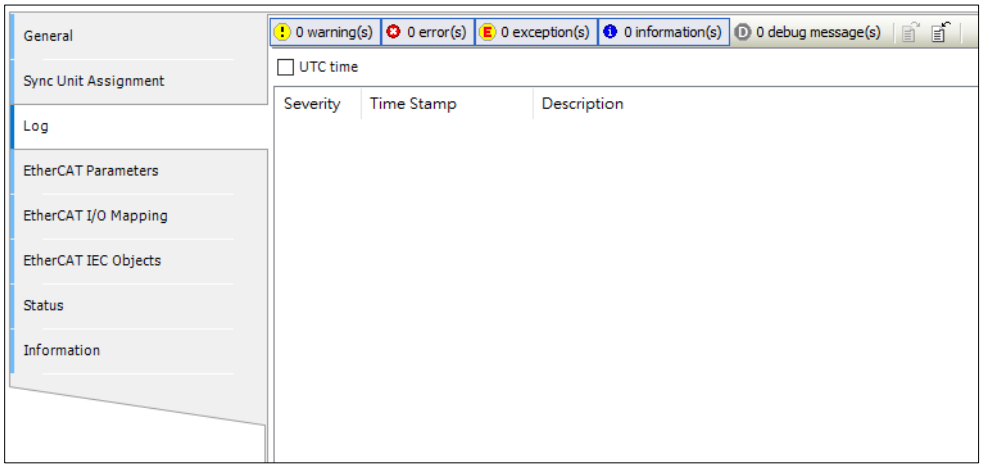
options icon to expand the option field)

- Use LRW instead of LWR/LRD: preset with separate read (LRD) and write commands (LWR). When the option is activated, the PDO is issued with a combined read / write command (LRW).
- Enable messages per task.
- Automatically restart slaves: When the EtherCAT master is disconnected after restart, the slaves will be restarted immediately.

Note: EtherCAT_Master_SoftMotion is preset to Network by MAC and the name is CPSW1.

- Log

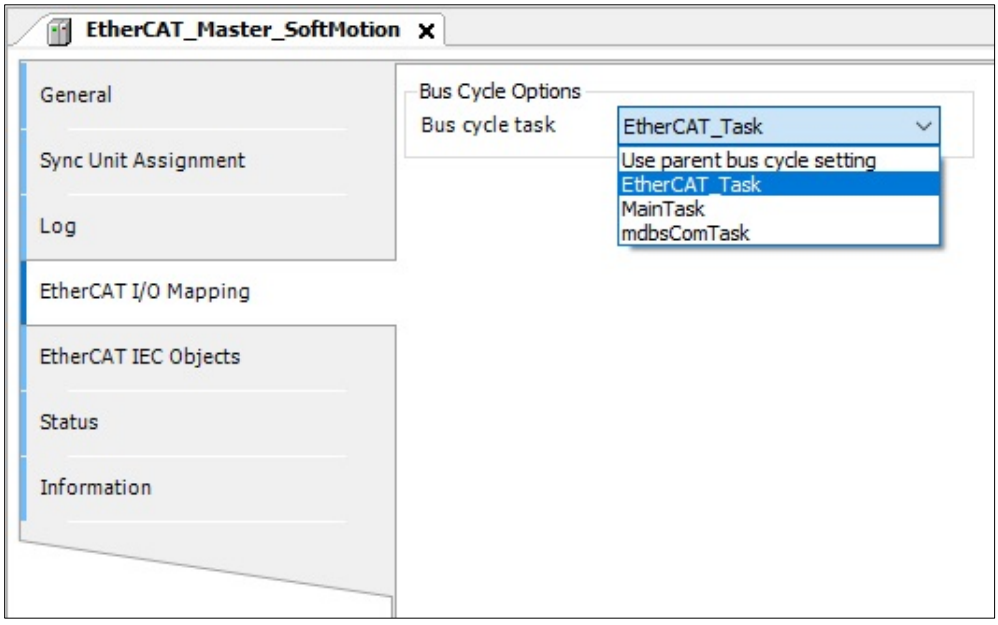
  This screen is recorded and displayed after the error is abnormal.



- EtherCAT I/O Mapping

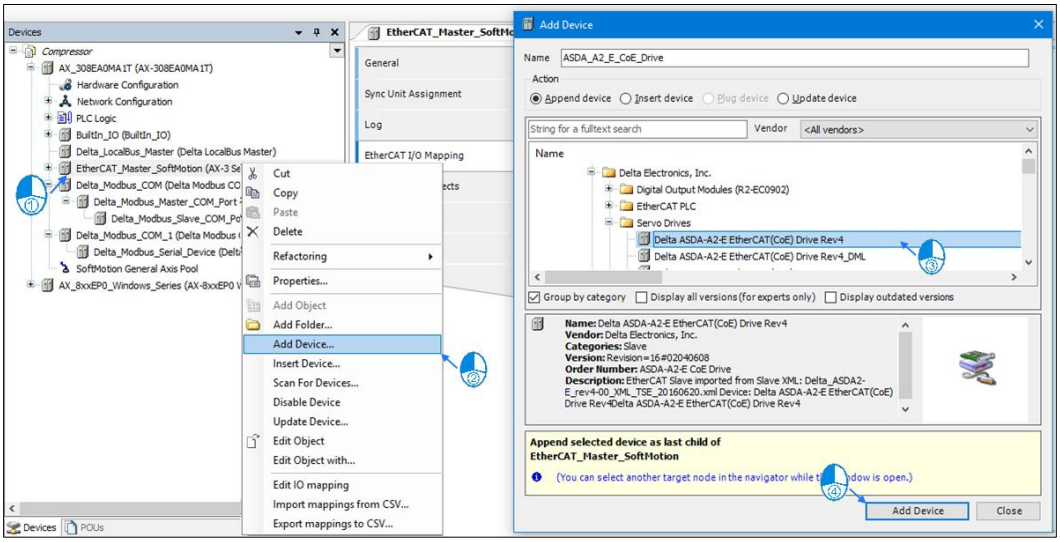
  This chapter introduces that EtherCAT needs to select the corresponding task. When the task is selected, the scan time of the task will be synchronized according to the build cycle.

  Bus cycle task: Select a Task to synchronize time with EtherCAT. If user select parent bus cycle setting, the system will automatically select the shortest cycle time in the task and change it to EtherCAT cycle time.

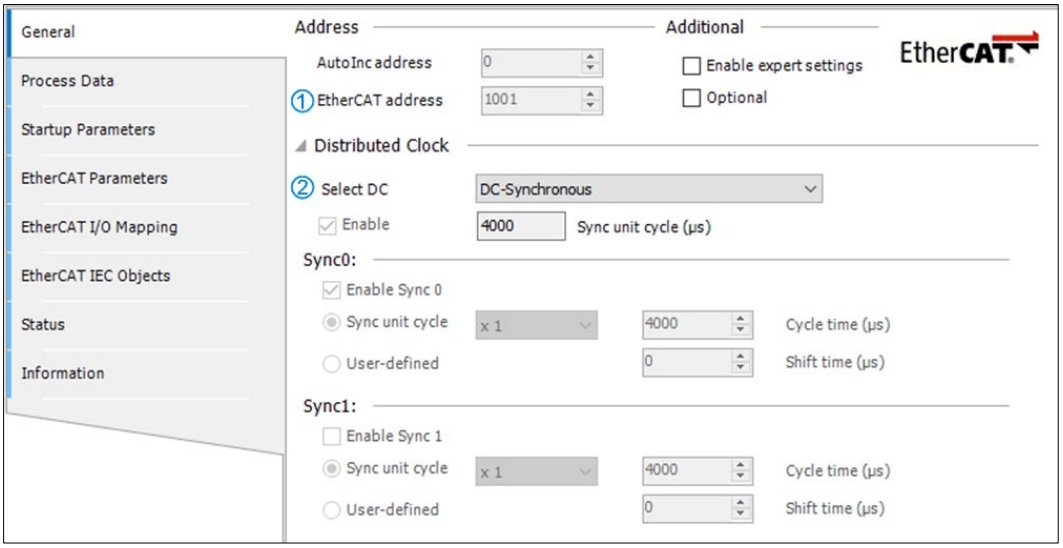


5.2.5.2 EtherCAT Slave Settings

Select EtherCAT Slave from the following path: right-click **EtherCAT_Master_SoftMotion** > **Add Device** > **Slave**. This chapter introduces the **EtherCAT_Slave screen**.



There are two ways to add slaves. The first is to add the slave device of the selected axis, and the second is to add the scanning method.



General

- Address

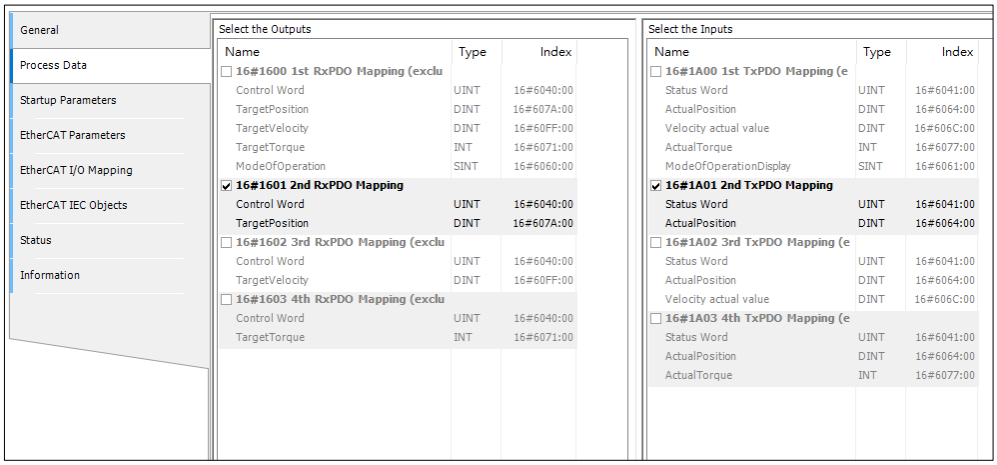
  ① EtherCAT address: address assigned to the slave by the master during startup.

- Distributed Clocks

② Select DC: Set the slave distributed clock.

- Process Data

  The data exchange on the EtherCAT network is a PDO mapping of CoE, which is a periodic data exchange between the master and the slave architecture; the data that the slave sends to the master will be packaged in TxPDO, and the slave will read The data is included in RxPDO. The input settings and output settings on the **PDO Mapping** page list the PDOs available for data exchange, and the PDO content can be edited. In the ESI file of the device, the contents of the preset checked PDO and PDO have been defined, and according to the definition of ESI, the PDO content of some devices allows user to edit it by themselves.

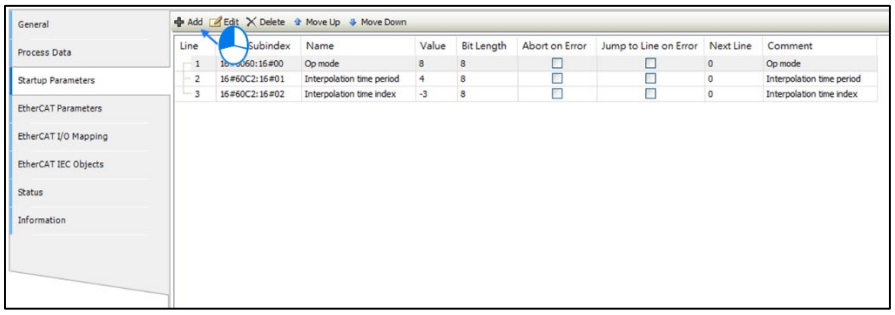


User can choose different groups of PDO to use according to their needs. The more data user choose for PDO, the greater userr PLC performance.

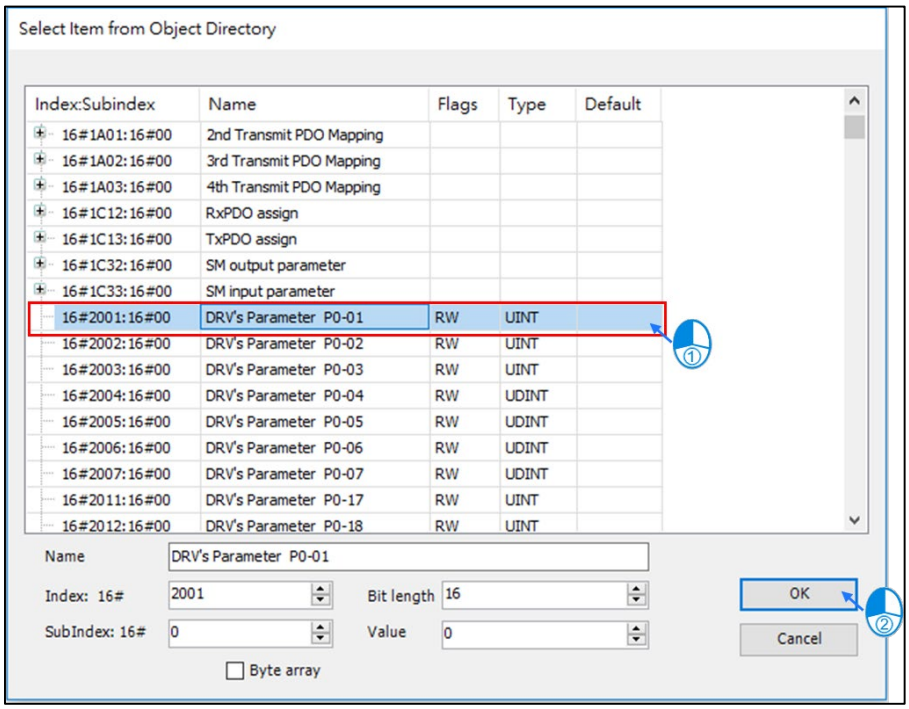
- Startup Parameters

  After the EtherCAT communication is established, the master station will download all the parameters in the table to the slave station, and the list will bring out the pre-defined commands of the ESI file. The user can add, delete or modify as required.

Features	Description
Add	Added parameters
Edit	Editing parameters
Move Up	Move selection parameter up one line
Move Down	Move selection parameter down one line



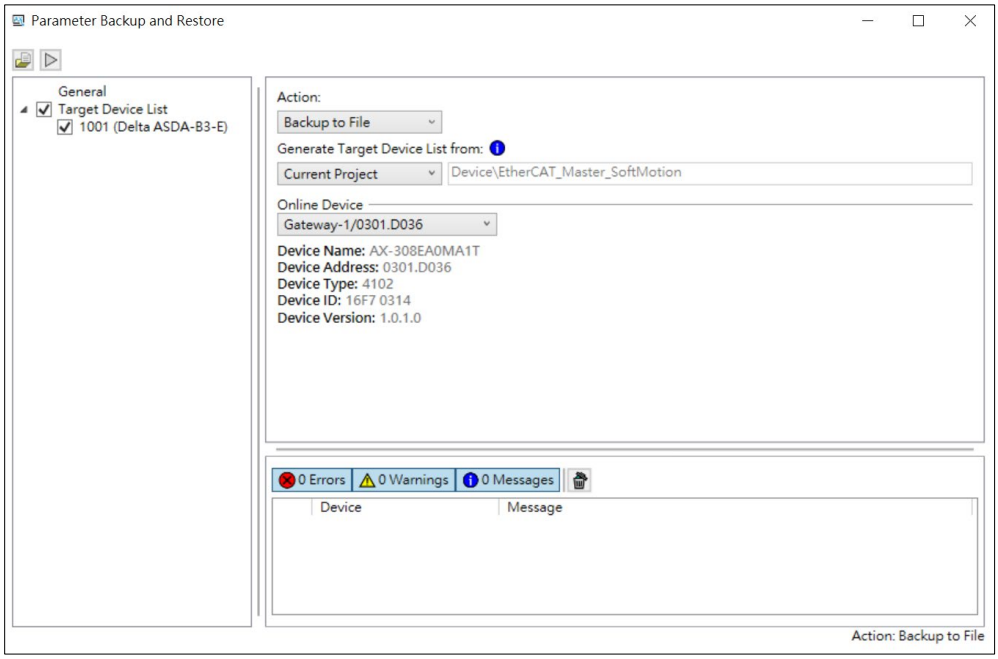
After pressing the **Add** button, the **Select Item from Object Directory** window will appear. First, select the parameter to be written from the directory. After clicking **OK**, it will be added to the command list.



5.2.5.3 Parameter Backup and Restore

The Parameter Backup and Restore function mainly performs backup/restoration for the parameters of the EtherCAT slave device under the AX-3 series controller.

- This function is applicable to the following controller models: AX-308EA0MA1T, AX-364ELA0MA1T, AX-308EA0MA1P, AX-316EA0MA1T, AX-304ELA0PA1P, AX-304ELA0PA1T.
- This function is applicable to the following slave device models: ASDA-A3-E, ASDA-B3-E.



Function	Description
General	When the **Generate Target Device List from** selects the current project or online topology, the general window will display the corresponding target slave device list.
Action	The **Action** menu can be divided into the following: • Backup to file: Backup the parameters of the slave station device to a computer file. • Backup to SD card: The parameters of the slave device are backed up to the SD card of the PLC. • Restore from file: restore slave parameters from computer files. • Restore from SD card: Restore slave station parameters from SD card.
Generate Target Device List from	The **Generate Target Device List from** menu of the target device list can be divided into the following: • Archive File • Current project • Online topology
Online device	Browse and select the controller currently on the network. The selected device information will be displayed below. • Device Name • Device Address • Device Type • Device ID • Device Version
Archive 	Click to pop up the **Save As** in Paramater backup and restore window. User can choose the archive path.
Start	Click **Start** to perform parameter backup and restore.

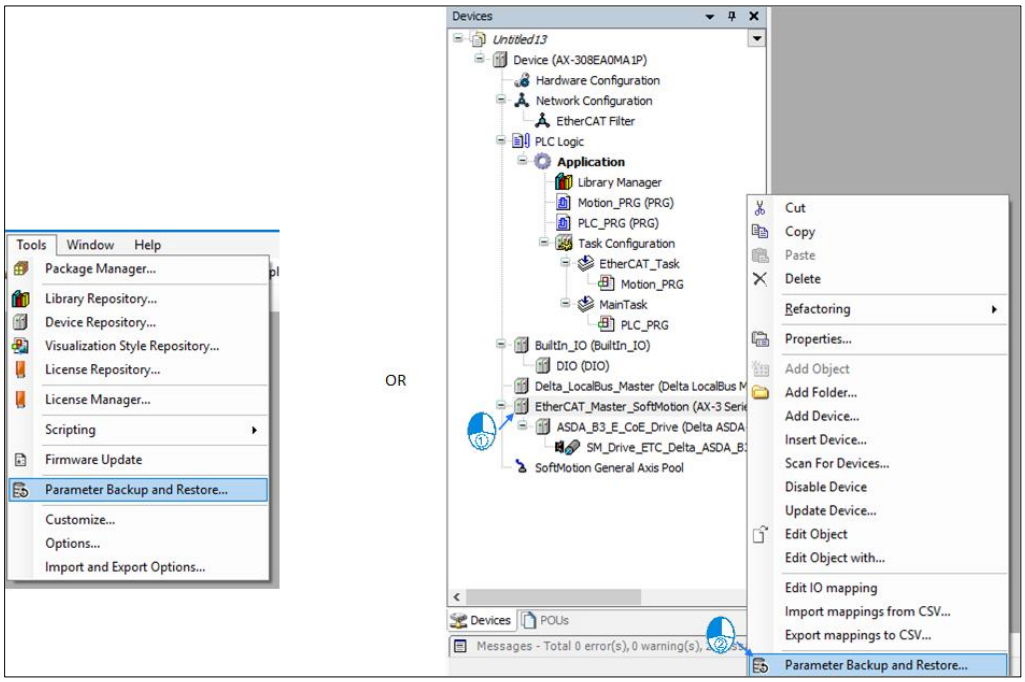
Function	Description
	

Please follow the steps below to perform parameter backup and restore:

1. Click **Tools** > **Parameter Backup and Restore** in the toolbar menu.

   Or

   Right-click **EtherCAT_Master** > **Parameter Backup and Restore** under the AX-3 series controller device.



2. User performs the following as required

- Back up the parameters of the archive file/current project/online topology slave device to a computer file.

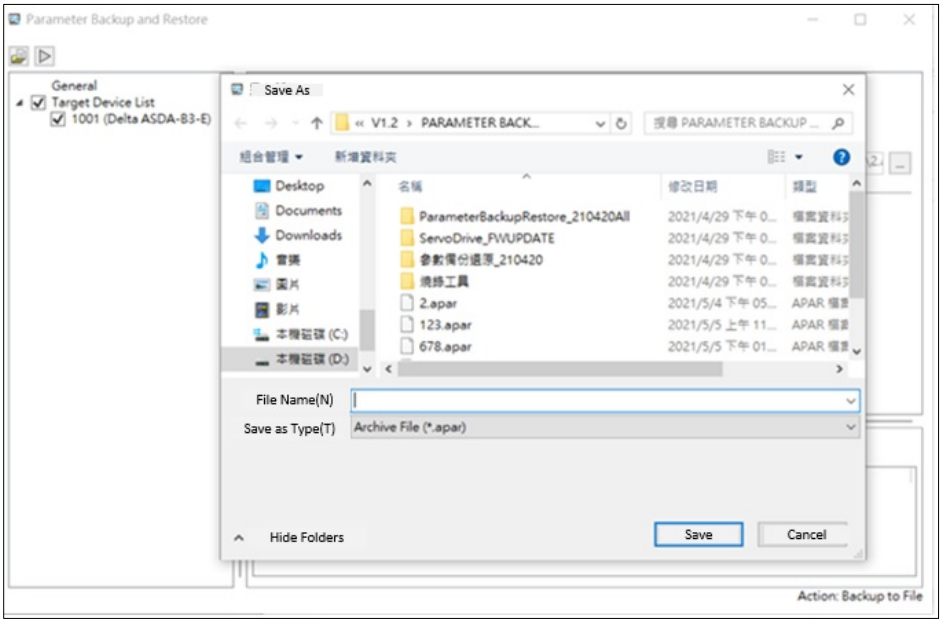
  Action: backup to file

  Generate Target Device List from: archive file/current project/online topology

  a. When selecting the archive file as the list source, first select the file path from the computer (the software will automatically read the current project/online topology).

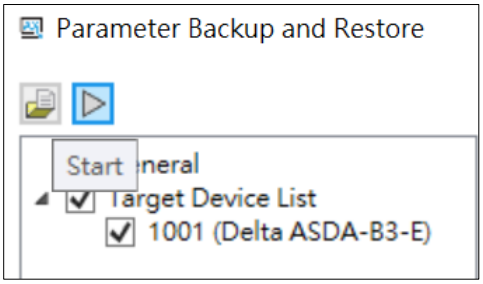
b. If user wants to archive the current device list and its configuration, click **Archive**, and the **Save As** New File window will pop up.

Result: The window will be displayed as shown in the following figure.

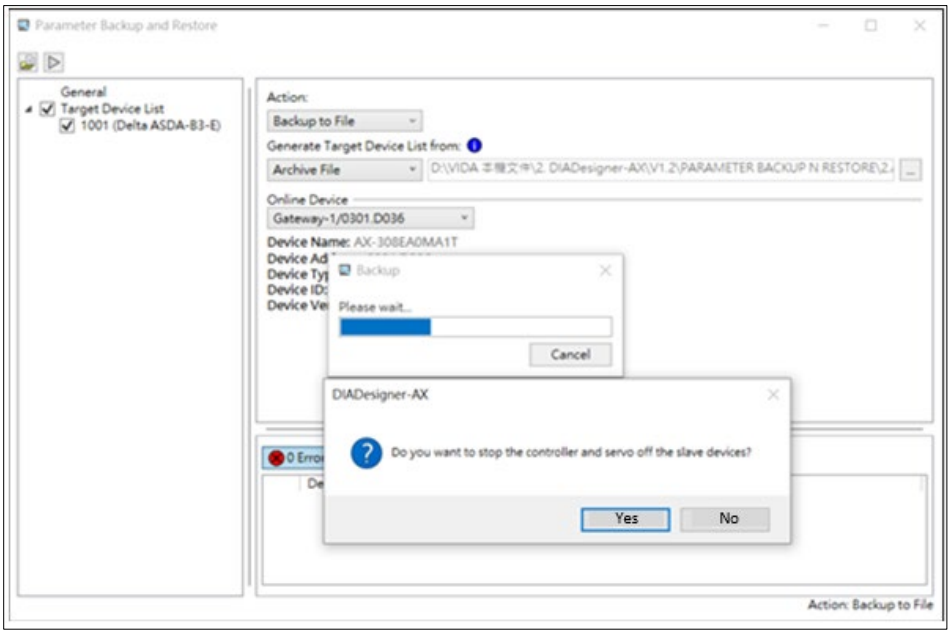


c. Select the path to save and type in the file name, and click **Save**.

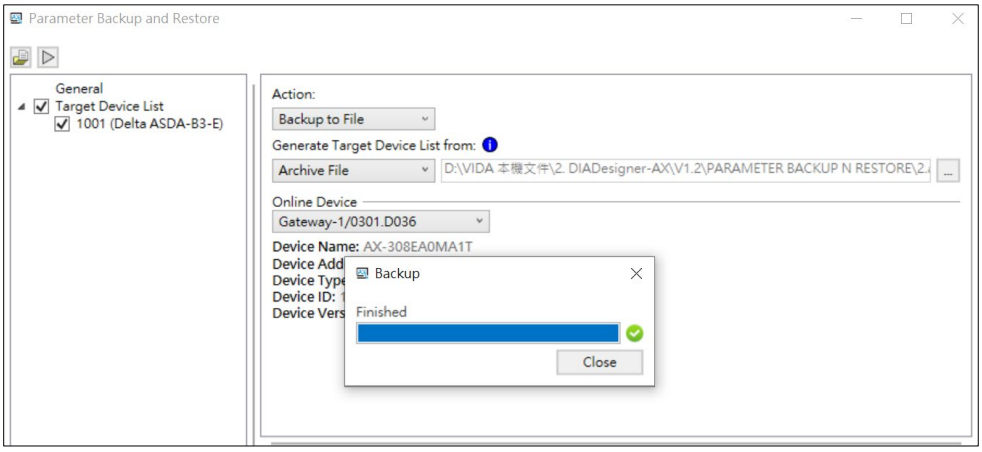
d. Click **Start** to start Parameter backup



e. Click **Yes** in the pop-up dialog box.



f. After the backup is completed, the progress bar will show that it has been completed.



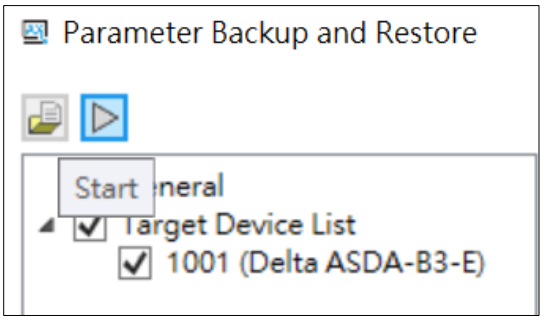
- Back up the archive file/current project/online topology slave device parameters to the SD card.

  Action: backup to SD card

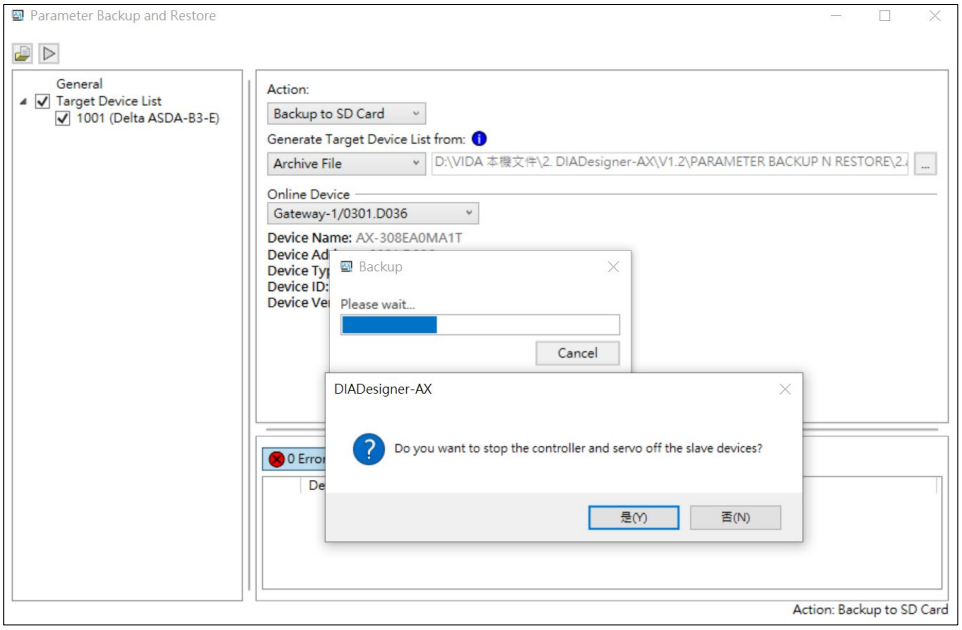
  Generate Target Device List from: archive file/current project/online topology

  a. When selecting the archive file as the list source, first select the file path from the computer (the software will automatically read the current project/online topology).

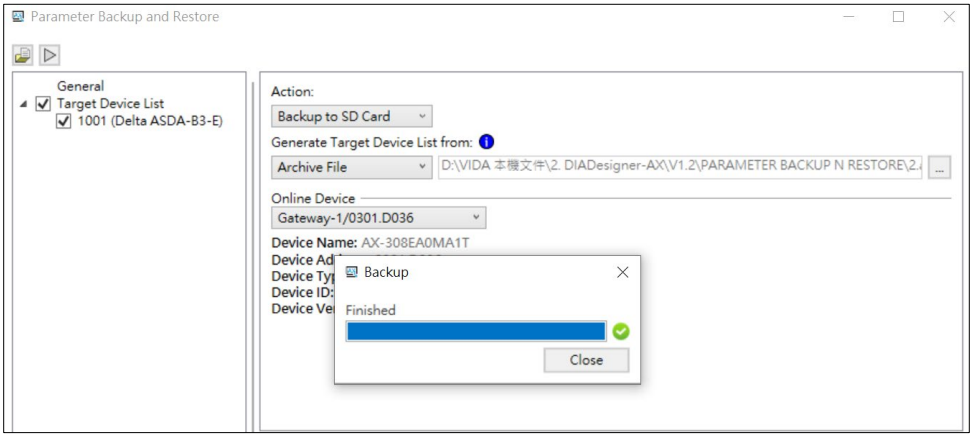
  b. Click **Start** to Start Parameter Backup.



c. Click **Yes** in the pop-up dialog box.



d. After the backup is completed, the progress bar will show that it has been completed.



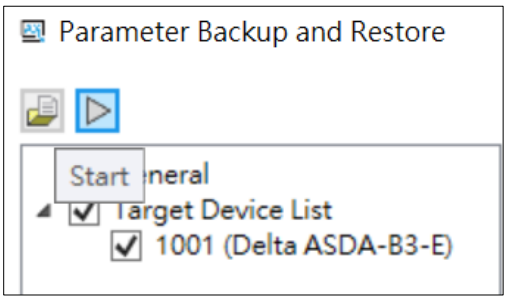
- Restore slave parameters from computer files.

  Action: Restore from file

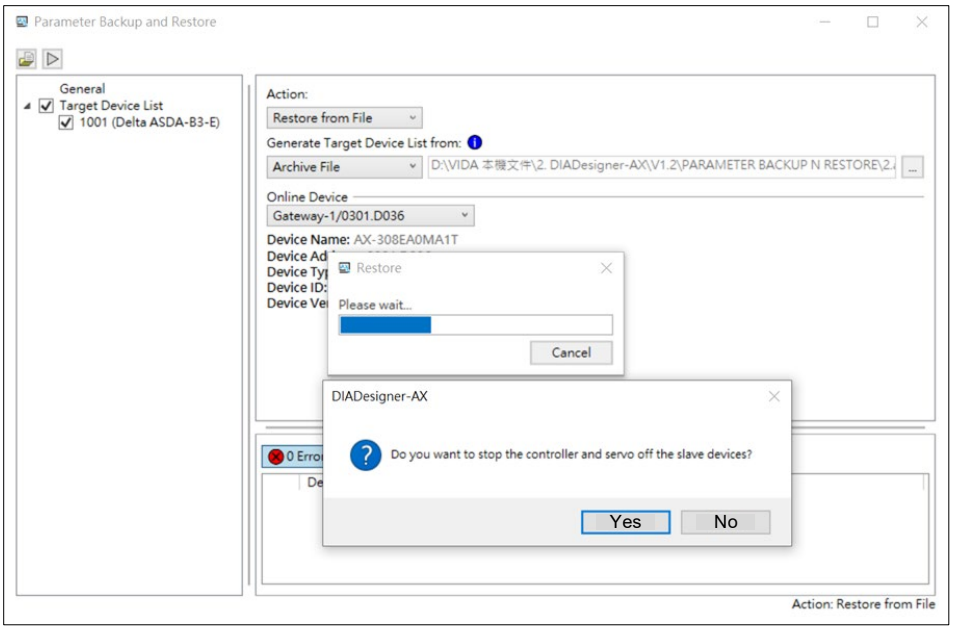
  Generate Target Device List from: archive file

a. When selecting the archive file from the list source, first select the archive file path from the computer (the software will automatically read the current project/online topology).

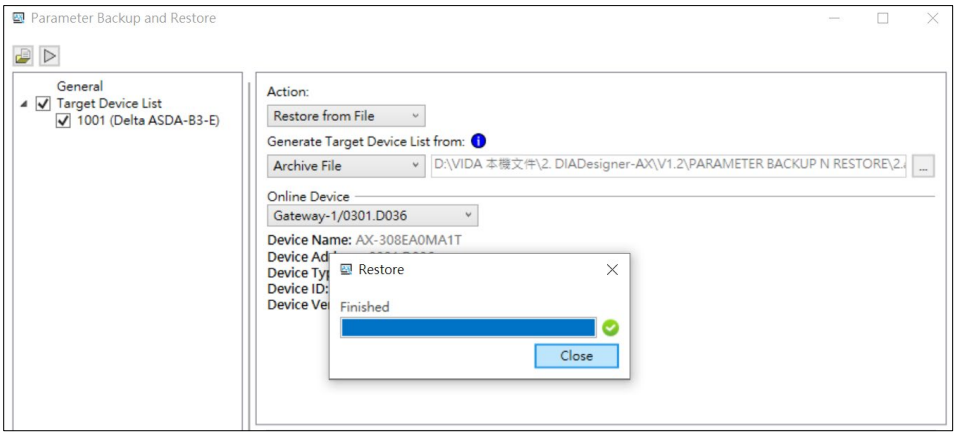
b. Click **Start** to start Parameter Restoration.



c. Click **Yes** in the pop-up dialog box.



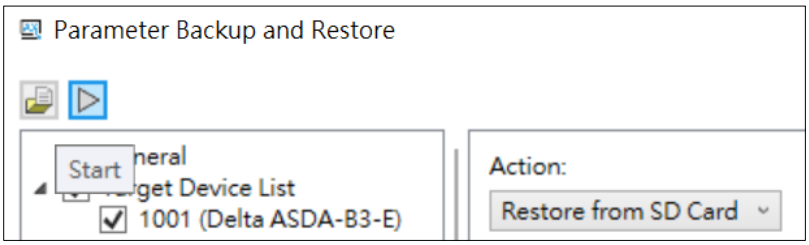
d. After the restoration is complete, the progress bar will show that it has been completed.



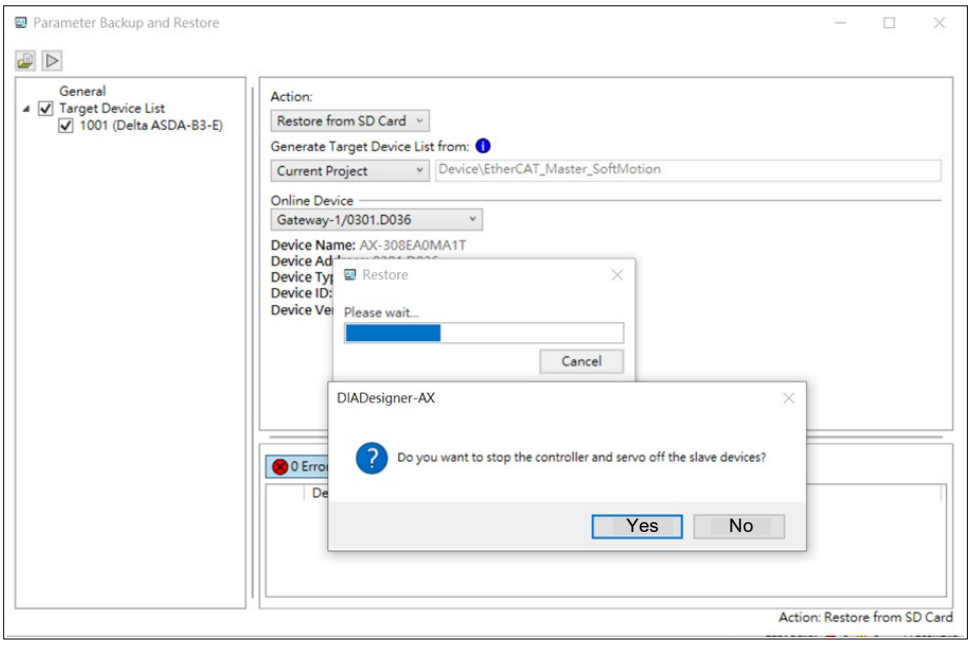
- Restore slave station parameters from SD card.

  Action: Restore from SD card

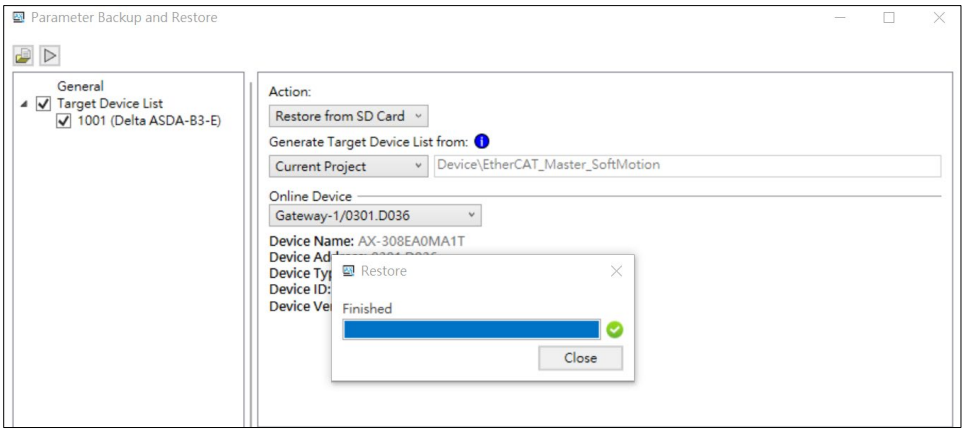
  a. Click **Start** to start Parameter Restoration.



  b. Click **Yes** in the pop-up dialog box.



  c. After the backup is completed, the progress bar will show that it has been completed.

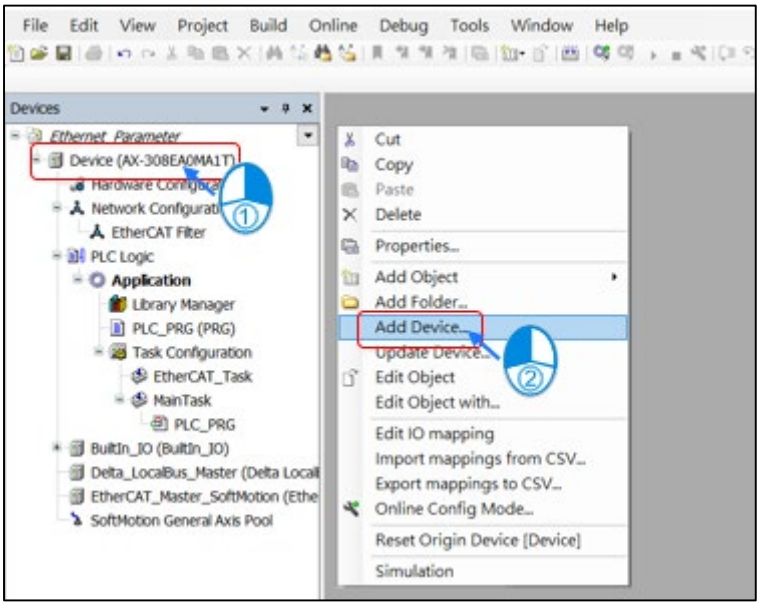

Parameter Backup and Restore
General
Target Device List
1001 (Delta ASDA-B3-E)
Action:
Restore from SD Card
Generate Target Device List from:
Current Project
Device\EtherCAT_Master_SoftMotion
Online Device
Gateway-1/0301.D036
Device Name: AX-308EA0MA1T
Restore
Finished
Close

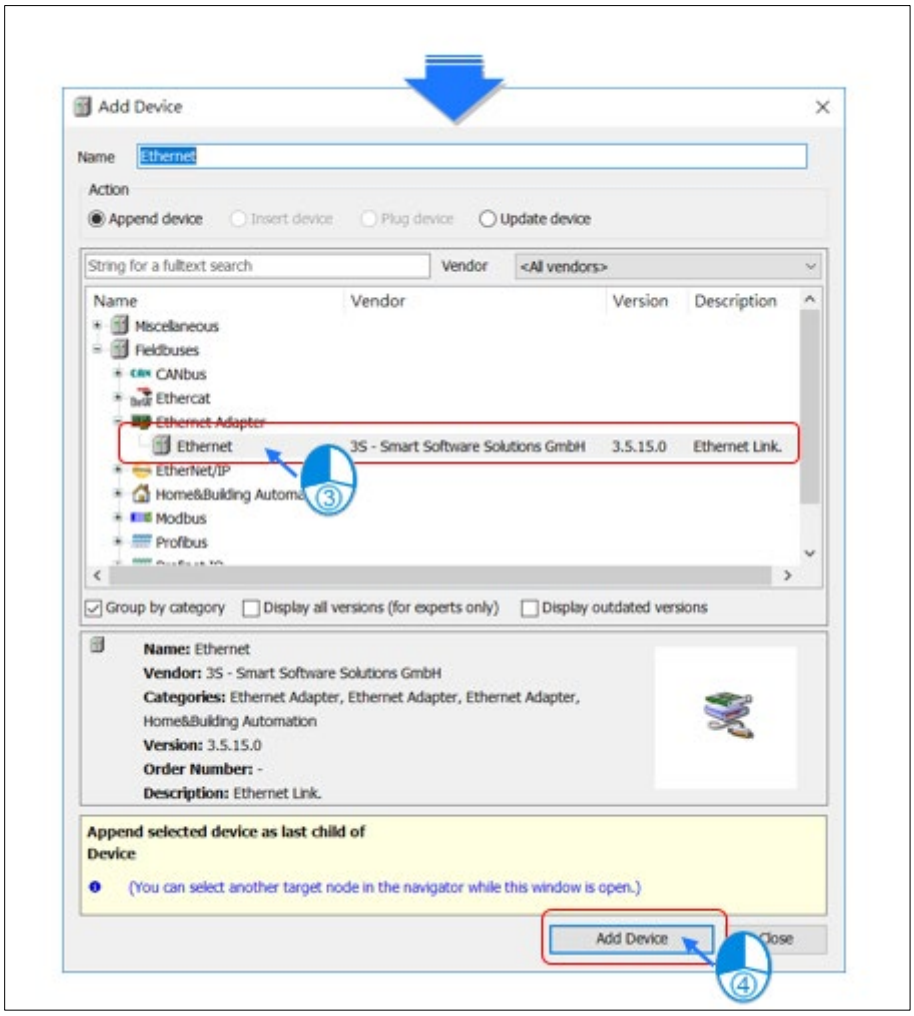
5.2.6 Ethernet Setting

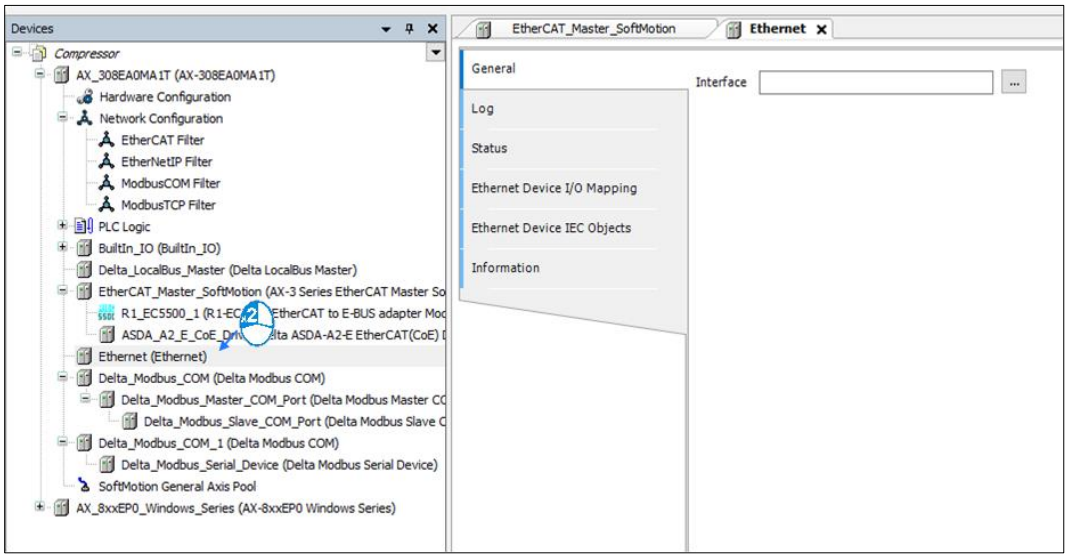
Ethernet Setting is related with Controller Ethernet parameter setting. User has to add the Ethernet Adapter Device to set parameter and internet related function, and all internet function is under **Ethernet** device, like MODBUS TCP and EtherNet/IP.

- Add **Ethernet Adapter Device**

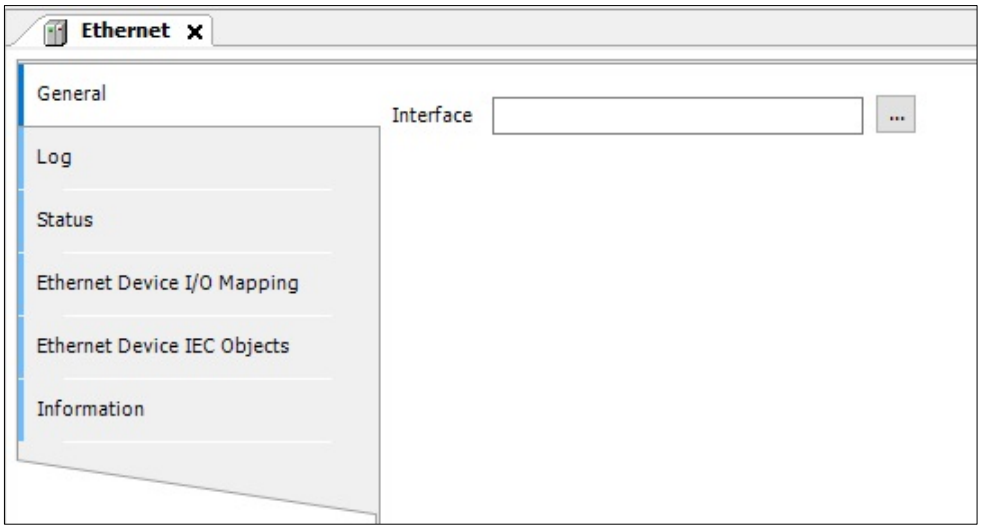
  Right-Click on the **AX-3 Series device** and click **Add Device** > **Fieldbus** > **Ethernet Adapter** > **Ethernet** to add the EtherNet device.







- EtherNet Device – General



Name	Description
Interface	Current controller communicate interface

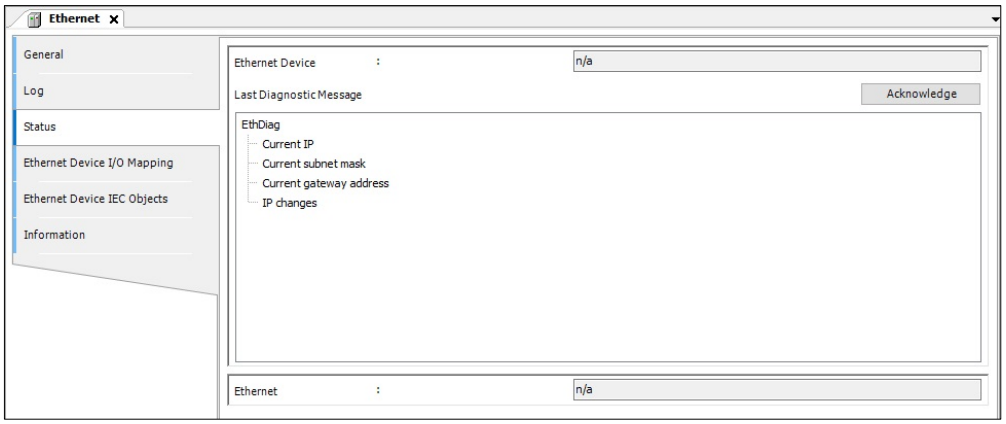
- Ethernet Device – Status

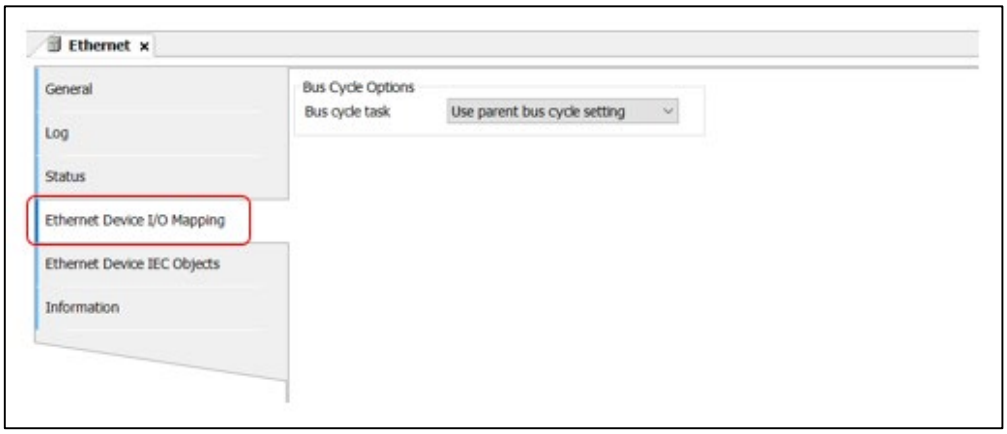
  User can check this page to know operation status of EtherNet Device, like Running/Stopped status.

Name	Description
Ethernet Device	Controller Internet status
Last Diagnostic Message	Diagnosis Status

- **Ethernet Device – EtherNet Device I/O Mapping**

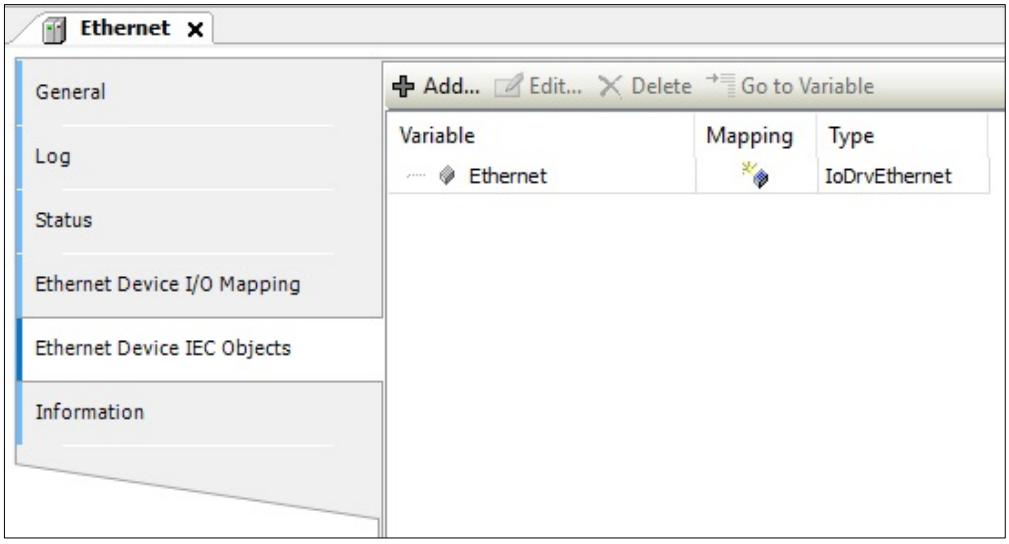
  User can set **Bus Cycle Task** on MODBUS TCP Slave Device. For more information on the bus cycle task, please refer to AX-3 operation manual “PLC Settings”.





- **Ethernet Device IEC Objects**

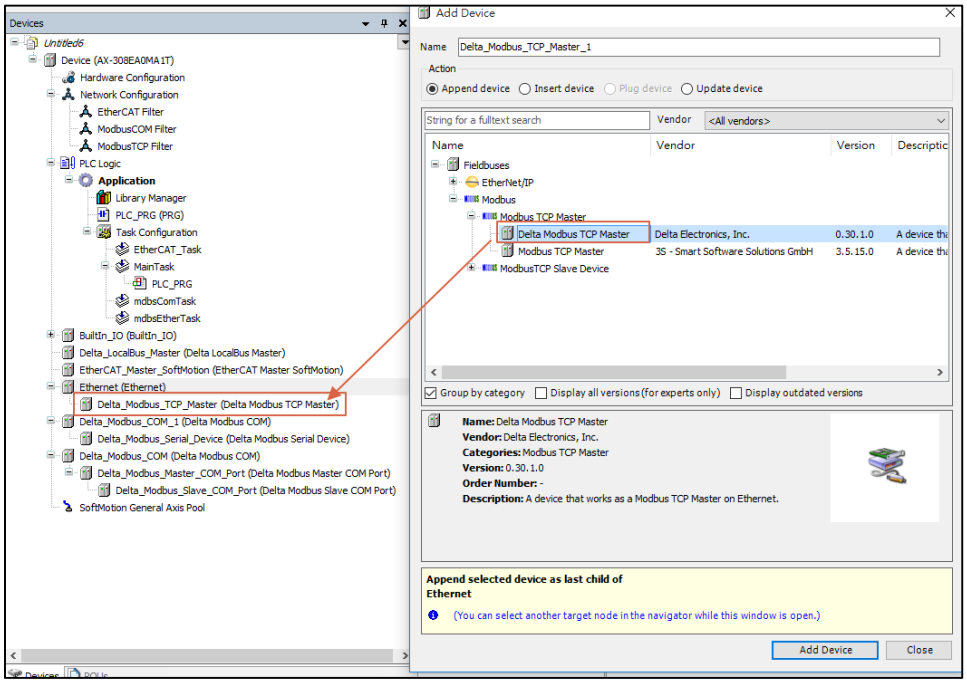
  This tab displays the status of the Ethernet Device. User can check the eState to monitor the operation status.



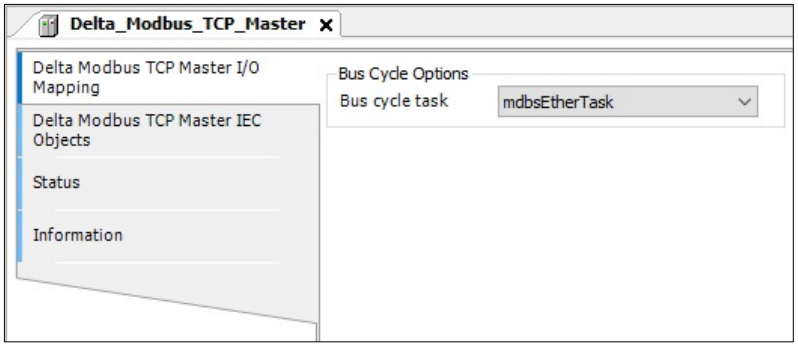
5.2.6.1 Delta MODBUS TCP Master

The Delta MODBUS TCP Master function is used to set the entry when the AX-3 series controller is used as the MODBUS TCP master station.

The AX-3 series not only supports standard MODBUS communication, but also supports MODBUS TCP. Right-click the **EtherNet** device and choose **Add device** > **Modbus** > **Modbus TCP Master** > **Delta MODBUS TCP Master**.



- Delta MODBUS TCP Master I/O Mapping:

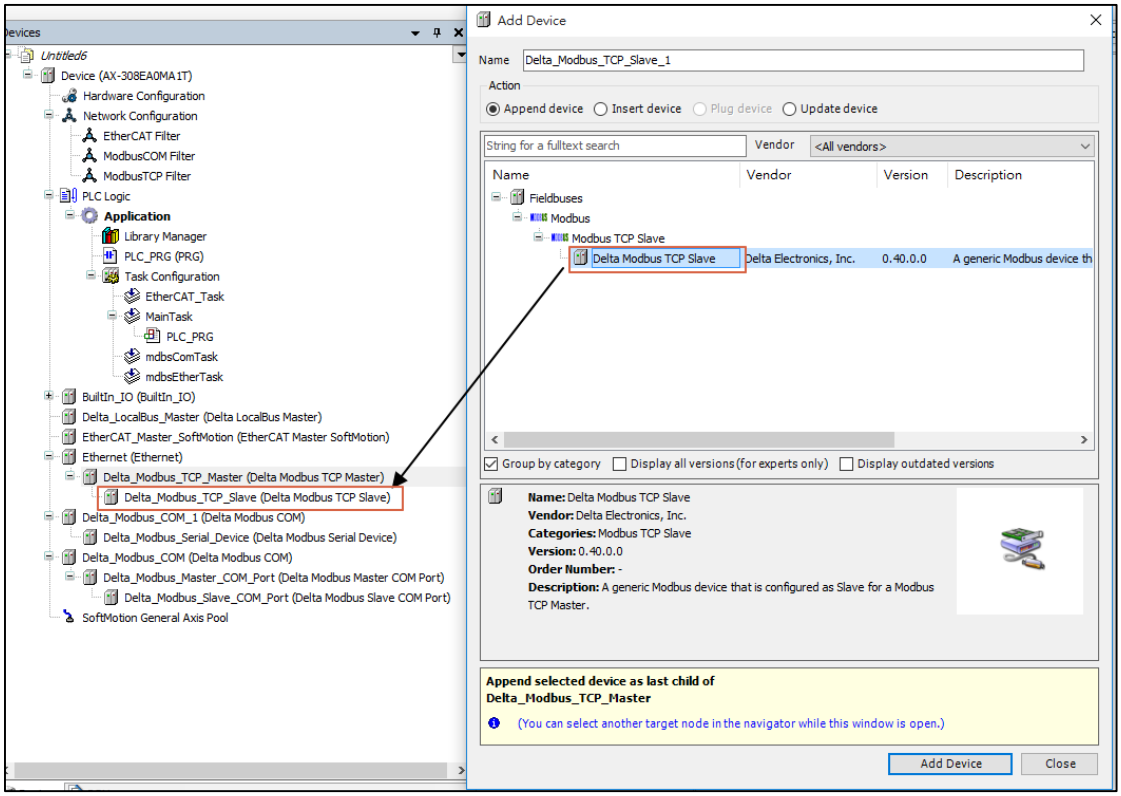


Name	Description
Delta MODBUS TCP Master I/O Mapping	User can set **Bus Cycle Task** on MODBUS TCP Slave Device. For more information on the bus cycle task, please refer to AX-3 operation manual “PLC Settings”.

Name	Description
Status	MODBUS TCP Master current operation status
Information	MODBUS TCP Master information, including the Vendor, Version etc.

5.2.6.2 Delta MODBUS TCP Slave

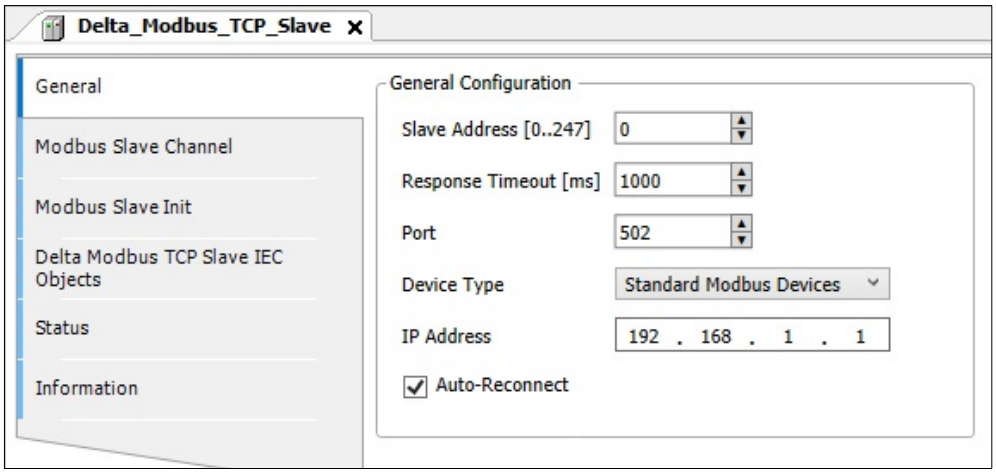
The Delta MODBUS TCP Slave function is used to set the target slave station to be connected when the AX-3 series controller is used as the MODBUS TCP master station. Right-click **Delta MODBUS TCP Master node** > **Add Device** > **MODBUS** > **MODBUS TCP Slave** > **Delta MODBUS TCP Slave**.



As shown above, there is an option to add channels in the MODBUS slave channel page, The MODBUS channel will vary with the device type setting in the general page. If user select a specific Delta controller, they can directly select the register location of the Delta controller, eliminating the need to look up the table (as shown below).

- General

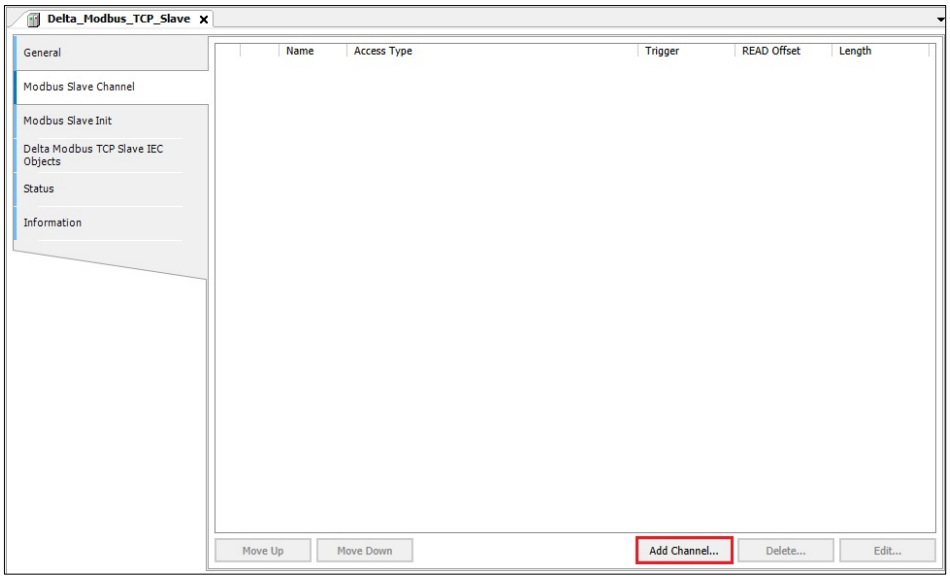
  User can set MODBUS TCP Slave related information such as station number, IP, communication port, communication format, etc. on this page.



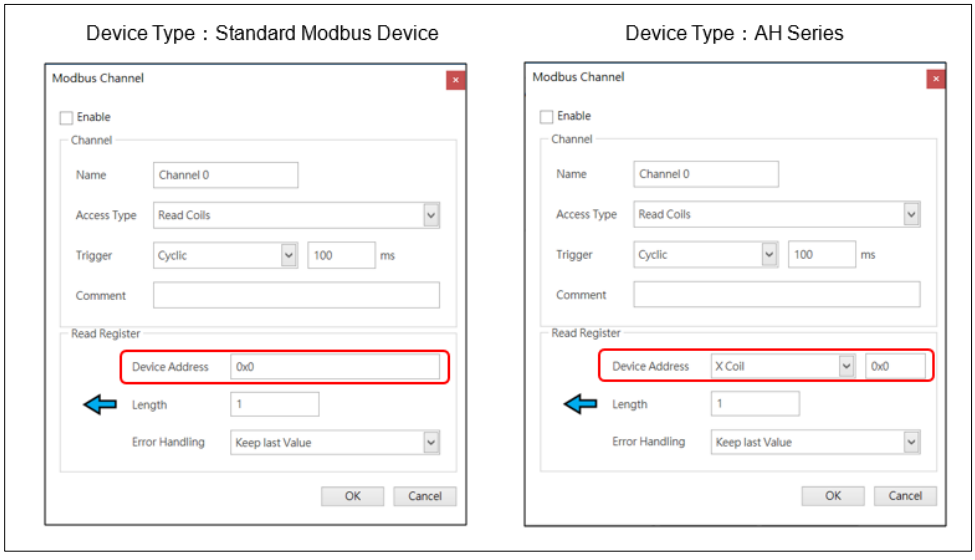
Name	Description
Slave Address	Slave station number.
Response Timeout	Slave response timeout.
Port	Slave communication port.
Device Type	Slave type can choose standard MODBUS communication or Delta specific controller (If Delta series controller is selected, the device and MODBUS address will be automatically converted).
IP Address	Slave IP address.
Auto-Reconnect	When a timeout or error occurs, reconnection is performed automatically.

Modbus Slave channel tab

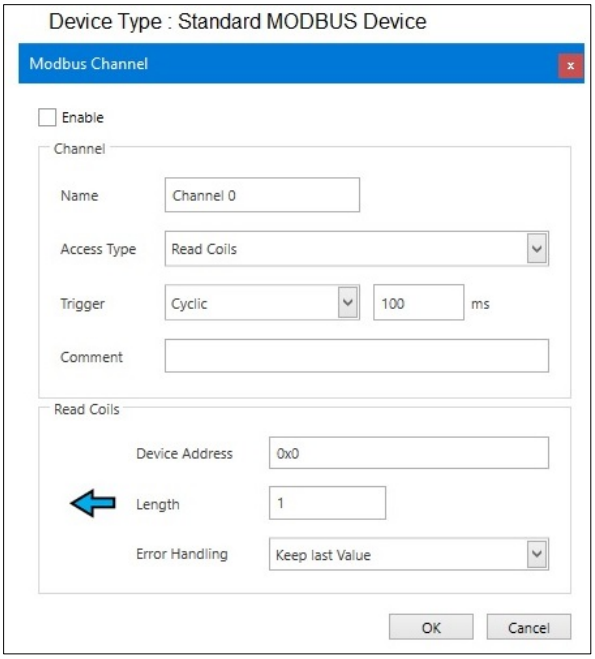
- MODBUS Slave Channel:
  User can use this table to define slave channels. Each channel represents a MODBUS request packet. At present, each slave station can create up to 100 Channels, the AX-308E series controller will send MODBUS request packets in order in a table order, and all channels share a TCP connection.



As shown above, there is an option to add channels in the MODBUS slave channel page > Slave Modbus Slave Channel tab.The MODBUS Channel will vary with the **Device Type** setting on the **General** page > **General** tab. If user choose a Delta specific controller, they can directly select the **Device Address** of the Delta controller, eliminating the need to look up the table as shown in the following.

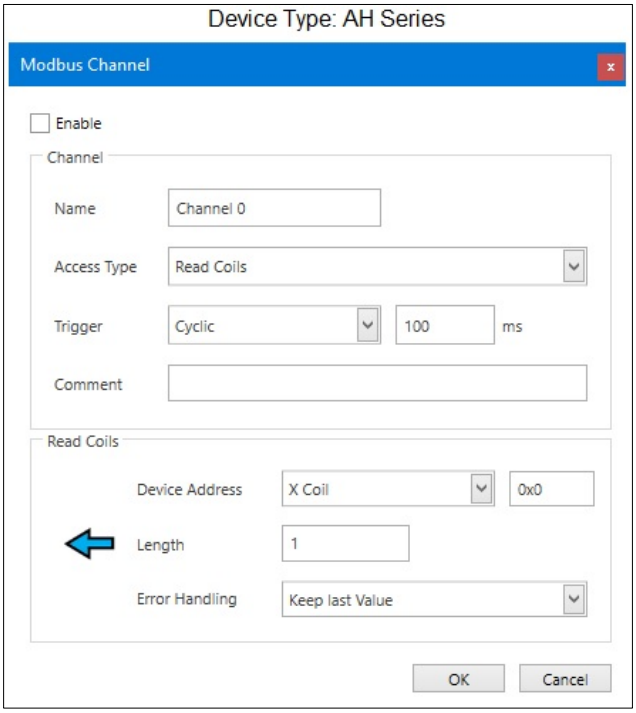


MODBUS Channel Description:



Name	Description
Enable	Decide whether this channel is effective.
Name	Define this channel name.
Access Type	MODBUS Communication function code: • Read coils (0x01) • Read discrete inputs (0x02) • Read holding registers (0x03) • Read input registers (0x04) • Read single coil (0x05) • Write single register (0x06) • Write multiple coils (0x0F) • Write multiple registers (0x10) • Read/Write multiple registers (0x17)
Trigger	Cyclic: MODBUS request Trigger at a set cycle time Rising edge: MODBUS request is triggered by a Bollinger variable, which is defined on the I/O Mapping page.

Name	Description
	Application: MODBUS request can be triggered through the function block **MODBUS Channel**.
Comment	Channel annotations
Device Address	MODBUS address of Register
Length	Read / write data length
Error Handing	When a communication error occurs, the setting operation of the Registers data is performed. • Set To ZERO. • Keep last value.

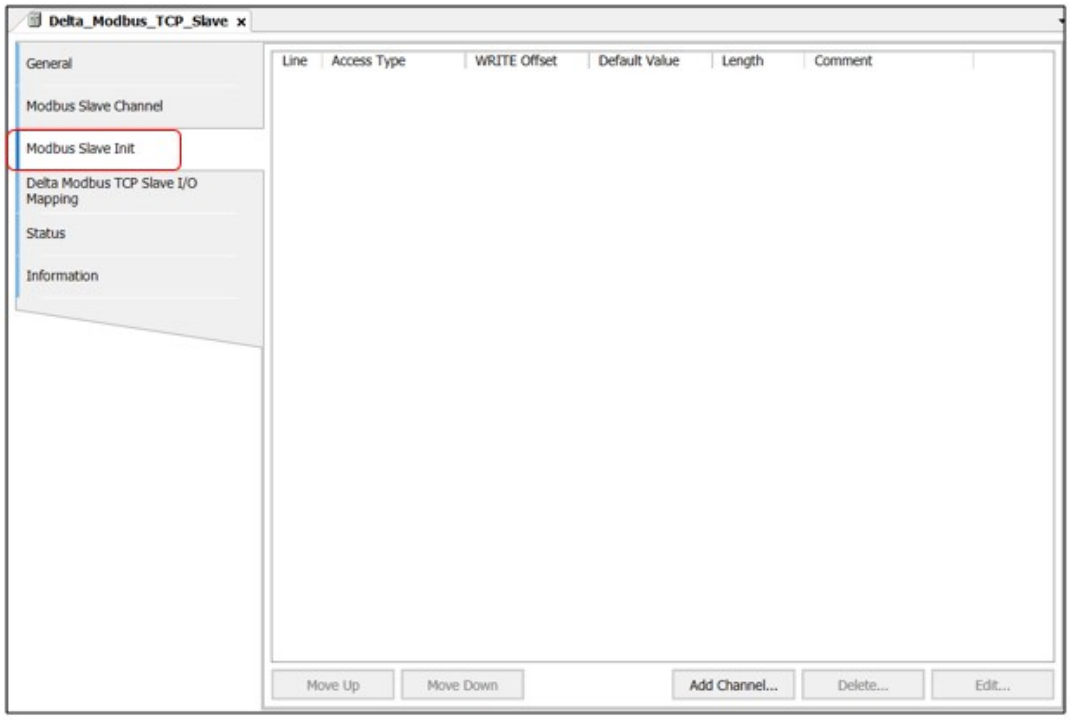


Name	Description
Enable	Decide whether this channel is effective.
Name	Define this channel name.
Access Type	Channel read and write actions: • Read coils

Name	Description
	• Read registers • Write coils • Write register ***Note**: The controller will select the corresponding MODBUS function code according to the type of device being read/written.*
Trigger	Cyclic: MODBUS request is triggered according to the set cycle time. Rising edge: MODBUS request is triggered by a Bollinger variable, which is defined on the I/O Mapping page. Application: MODBUS request can be triggered through the function block **MODBUS Channel**.
Comment	Channel annotations.
Device Address	Device Address of Delta Series Controller.
Length	Length of data read/written (Maximum length is 256 coils and 100 registers).
Error Handing	When a communication error occurs, the setting operation of the Registers data is performed. • Set To ZERO. • Keep last value.

MODBUS Slave Initialization

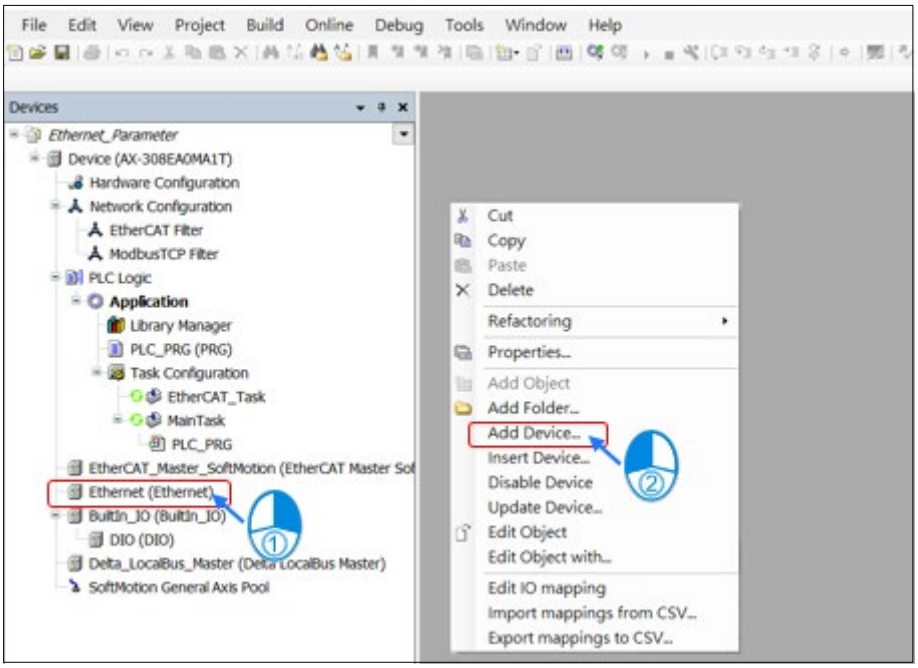
The user can use this table to decide whether to write the initial value to the registers of the slave station after the AX-308E series controller successfully establishes a TCP connection with the slave station.

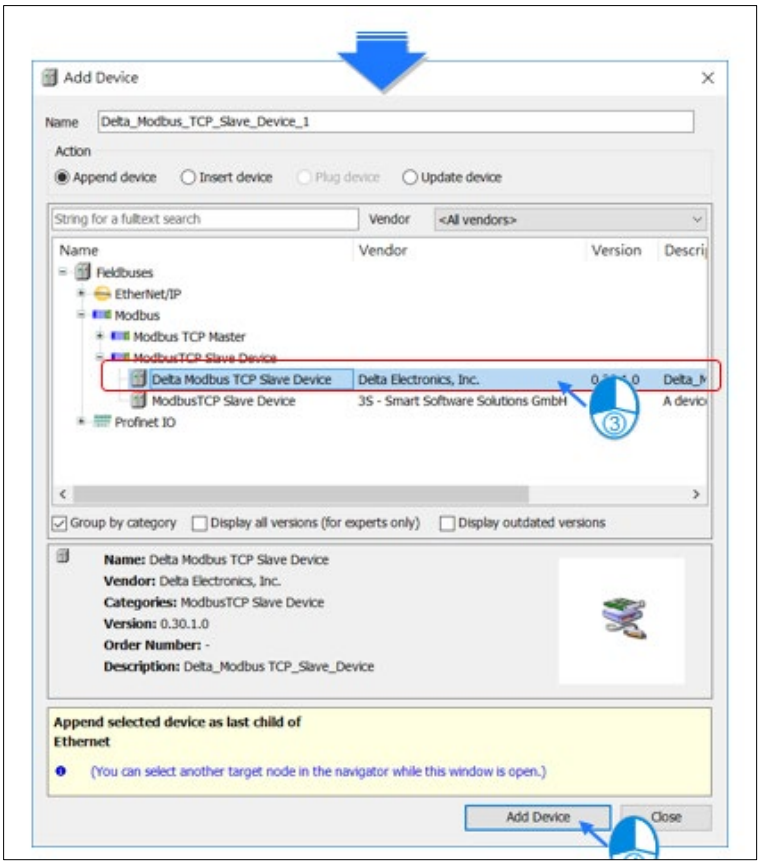

Delta_Modbus_TCP_Slave
General
Modbus Slave Channel
Modbus Slave Init
Delta Modbus TCP Slave I/O Mapping
Status
Information
Line
Access Type
WRITE Offset
Default Value
Length
Comment
Move Up
Move Down
Add Channel...
Delete...
Edit...

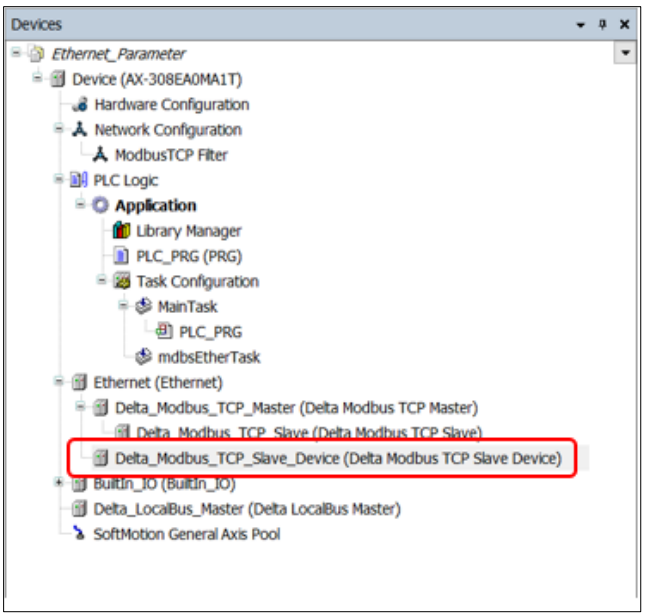
5.2.6.3 Delta MODBUS TCP Slave Device

The Delta MODBUS TCP Slave function is used to set the entry when the AX-3 series controller is used as a MODBUS slave.As AX-308E controller is MODBUS TCP Slave, user adds the **Delta MODBUS TCP Slave Device** tool to set the register area. If user uses the Delta specific fieldbus on MODBUS TCP Master, the register area is not limited and can read/write to all registers.

Right-click the **EtherNet node** > **Add device** > **MODBUS TCP Slave Device** > **Delta MODBUS TCP Slave Device**.

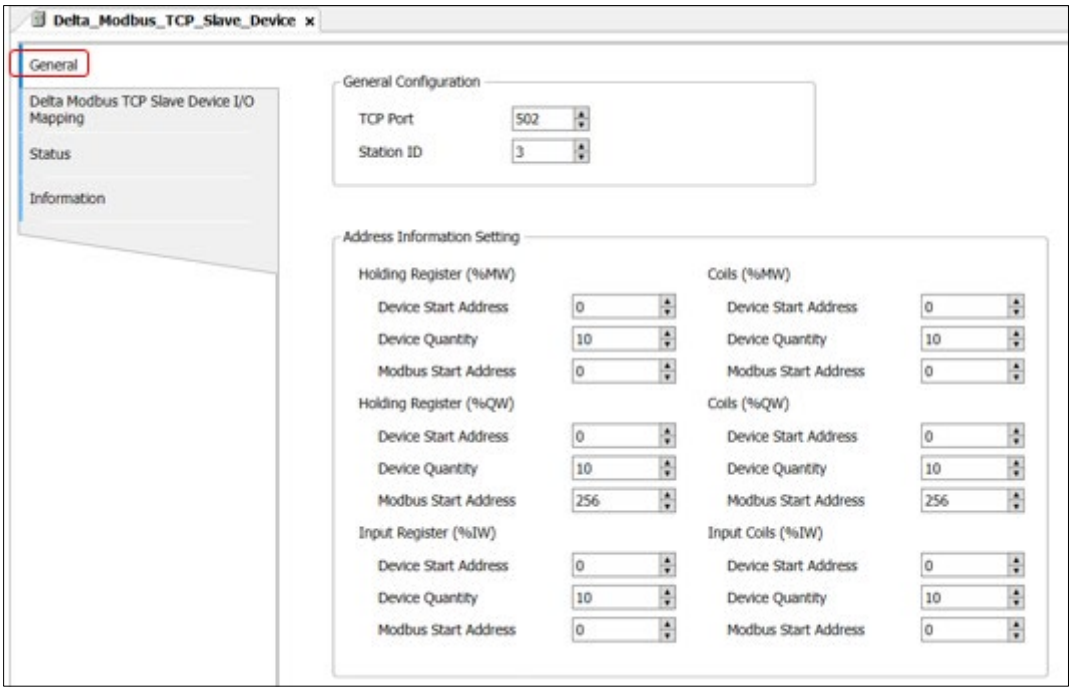






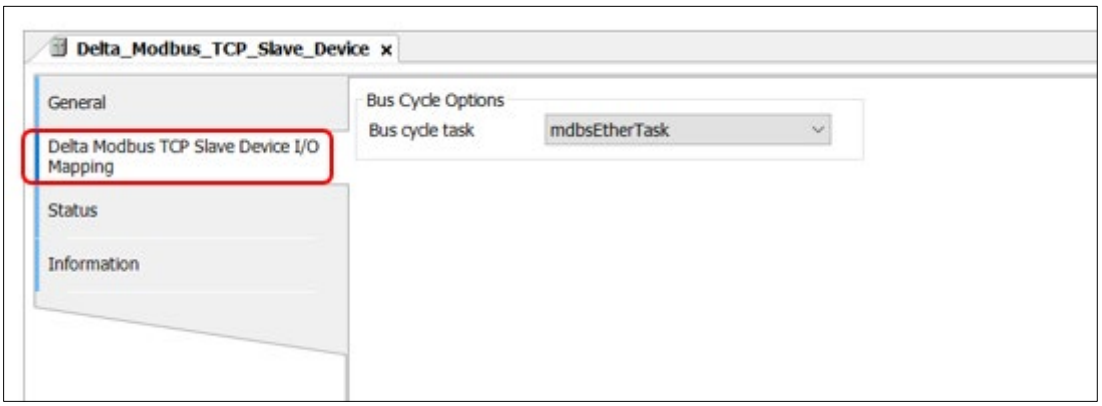
General:

User can set the TCP Port, Station ID, and the register area in this page. If the MODBUSTCP Master is using Delta specific communication, the register is not limited.



Delta MODBUS TCP Slave Device I/O Mapping:

User can set **Bus Cycle Task** on **MODBUS TCP Slave Device**. For more information on the bus cycle task, please refer to *AX-308E Operation manual PLC Settings*.



5.2.7 EtherNet/IP Settings

The EtherNetIP plug-in provides a dialog box for setting up the EtherNet/IP scanner and target device. This dialog box will be displayed in the device editor dialog box.

EtherNet/IP based on standard TCP and UDP allows communication between office networks and control systems. The EtherNet/IP target device supports DHCP and BootP to assign IP addresses. Start the EtherNet/IP network (diagnosis), user can use the Web server integrated with the logic control interface module, or user can use the Web server of other EtherNet/IP devices.

Like other standard EtherNet-based networks, the EtherNet/IP protocol is not suitable for real-time applications.

(< 1 millisecond), such as servo system, because the typical cycle of EtherNet/IP network is 10 milliseconds.

Install and add EtherNet/IP devices

In order to add and configure EtherNet/IP devices in the project tree, user need to install related device description files (EtherNet/IP device description files (*.eds)) in **Tools** > **Device Repository**.

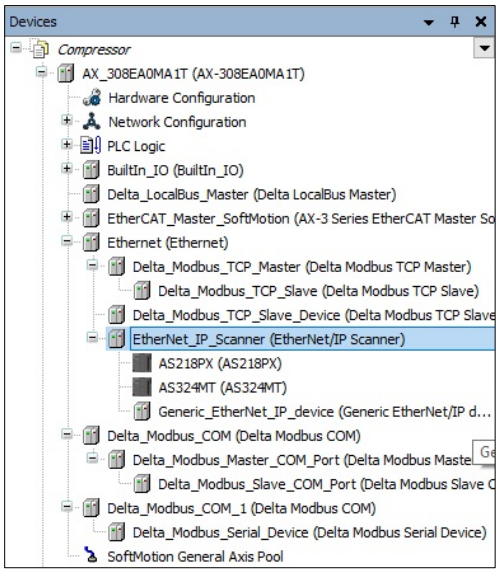
Adding an EtherNet/IP scanner to the project tree will add the specific library collection of the related device to the library manager. There are two ways to insert one or more EtherNet/IP remote adapters or devices into the scanner:

- Module configuration

  Under one target device is another target device, called the **Chassis**, which can be inserted into a fixed number of slots. Use the **Insert Device To** command to insert the so-called **IO Point** into the slot, thereby controlling the input and output.

- Non-modular configuration

  The target device is equipped with input and output.



Double-click on EtherNet/IP object in the project tree, or select a device in the project tree and call the edit object command in the menu to open the device editor for configuring EtherNet/IP devices. (The settings of the editor itself, such as displaying the general configuration dialog box, are implemented in the device editor options).

The title bar of the EtherNet/IP configuration dialog box is marked with the name of the specified EtherNet device; depending on the device, the label included in the dialog box will be different. Note that the bus cycle time is adjusted in the PLC settings.

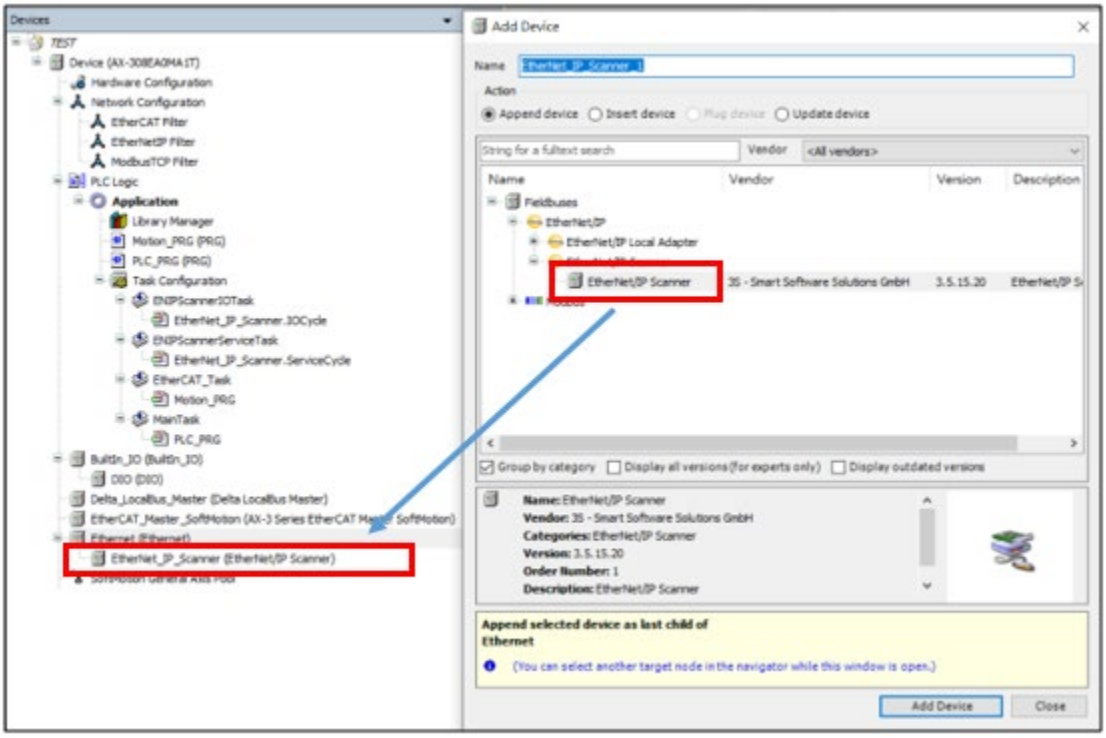
- EtherNet/IP Scanner
- EtherNet/IP Remote Adapter

Access EtherNet/IP configuration through the app

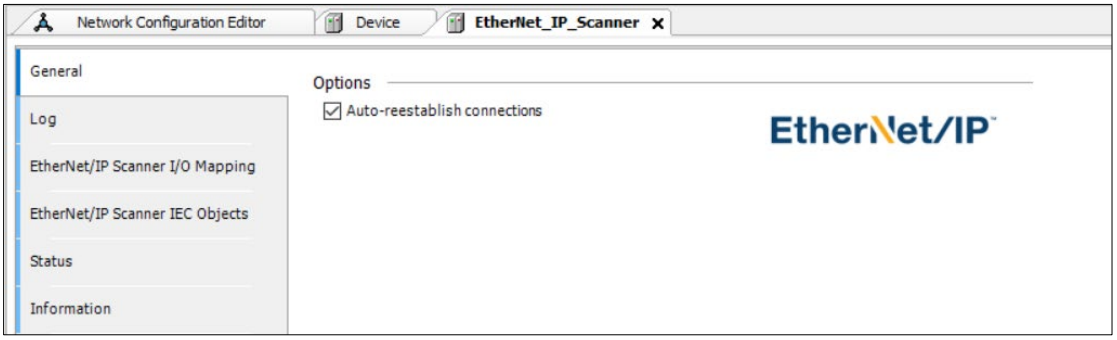
Each EtherNet/IP scanner will automatically add a related function block instance to the project tree. The name of this variable and the type of the function block will appear in the EtherNetIP scan I/O map of the IEC object tab. Variables can be configured through the project address, such as allowing start, stop or application scan status check.

5.2.7.1 EtherNet/IP

The AX-3 series supports EtherNet/IP scanner function. Right-click on the **EtherNet node** > **Add Device** > **Fieldbus** > **EtherNet** > **EtherNetIP Scanner** > **EtherNet/IP Scanner**.

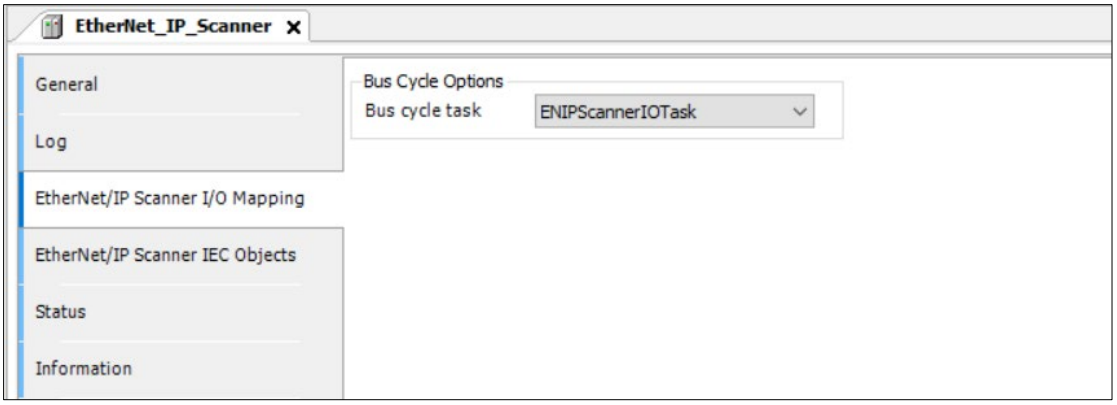


EtherNet/IP Scanner General tab



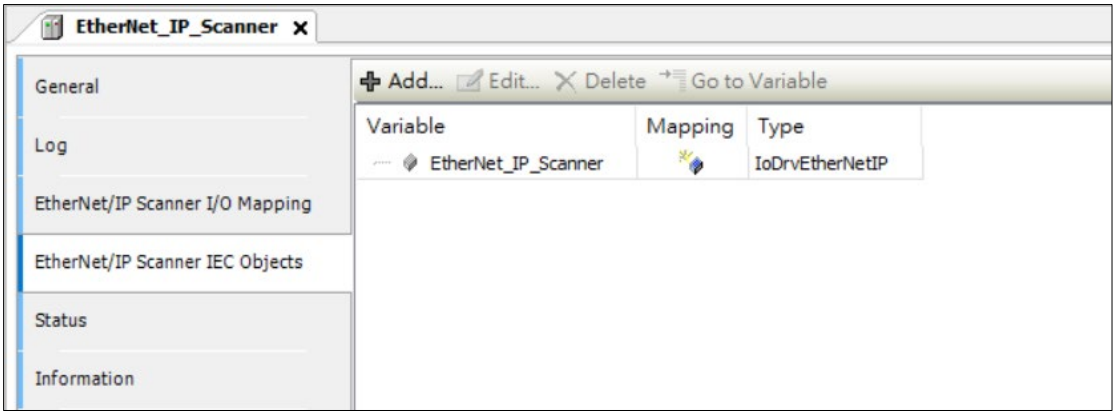
Project	Description
Auto re-establish connections	When a timeout or error occurs, it will automatically reconnect.

EtherNet/IP Scanner I/O Map tab



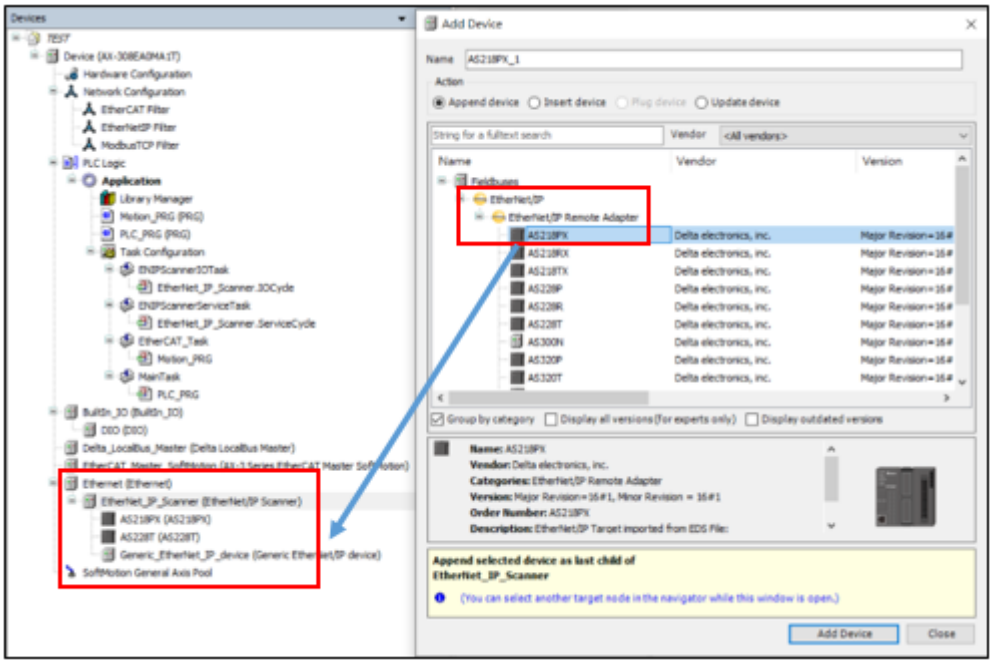
Project	Description
Bus cycle task	Select bus cycle task. For more information about bus cycle tasks, please refer to AX-3 Operation Manual: "PLC Settings".

This tab displays the definition status of EtherNet/IP devices. User can know the running status from the status.



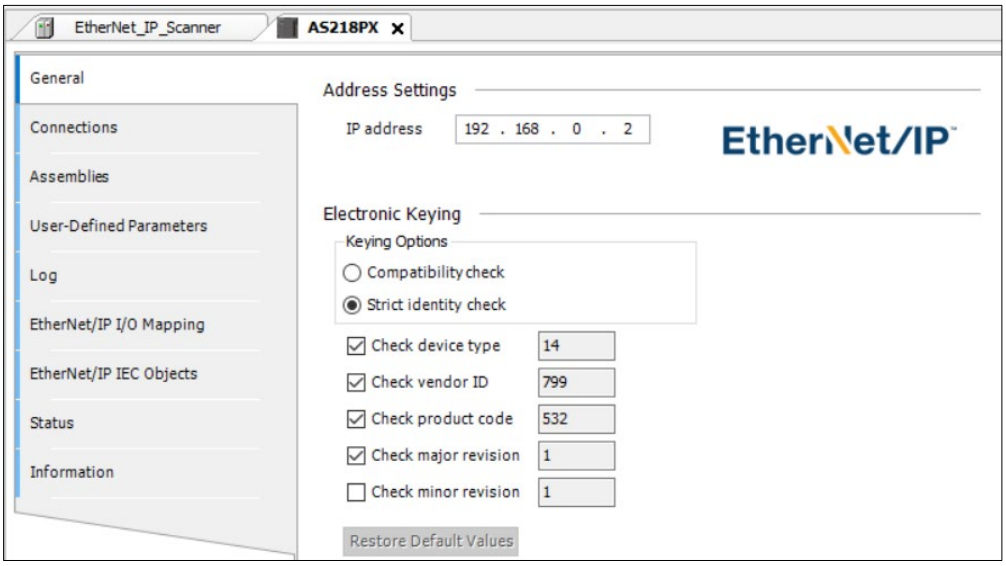
5.2.7.2 EtherNet/IP Remote Adapter

AX-308E supports EtherNet/IP remote adapter function. Right-click on the **EtherNetIP Scanner node** > **Add Device** > **Fieldbus** > **EtherNetIP** > **EtherNet/IP Remote Adapter**.



If user want to use a third-party EtherNet/IP remote adapter, please go to **Tools** > **Device Library** > Import the EtherNet/IP standard EDS file, and then add it in the **Add Device window**.

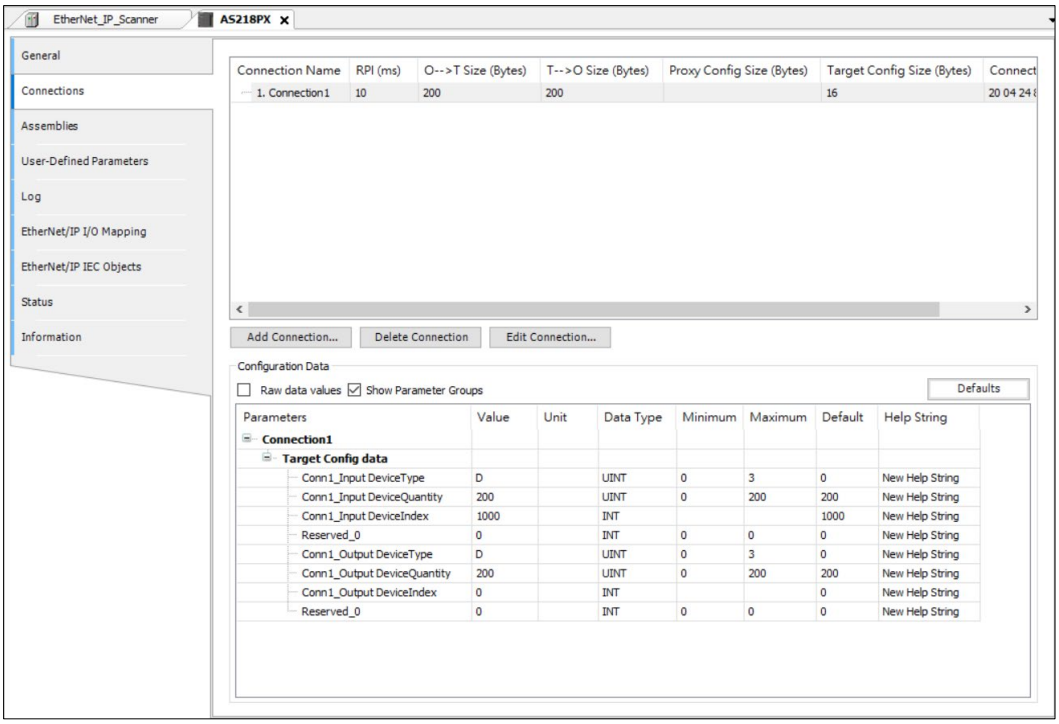
EtherNet/IP Remote Adapter General tab



Project	Description
Address Settings – IP Address	EtherNet/IP remote adapter IP Address
Electronic keying	
Keying Options-Compatibility check	Start: The adapter performs a compatibility check on its EDS file. All key values will be sent to the device. The device decides whether it is compatible with the received value.
Keying options-Strict identity check	The adapter performs a compatibility check on its own EDS file. The user decides which keying information should be checked. If the check fails, no I/O connection with the device will be established, and an error message will be posted on the status page.
Check device type	Check device type
Check vendor ID	Check supplier code
Check product code	Check product code
Check major revision	Check major revisions
Check minor revision	Check minor revisions
Restore Default values	Restore preset values for general equipment

EtherNet/IP Remote Adapter Connection tab

The top of this setting screen displays a list of all configured connections. If there is an **exclusive owner** connection in the EDS file, it will be automatically inserted when adding a device. The configuration data of these connections can be modified at the bottom of the screen.

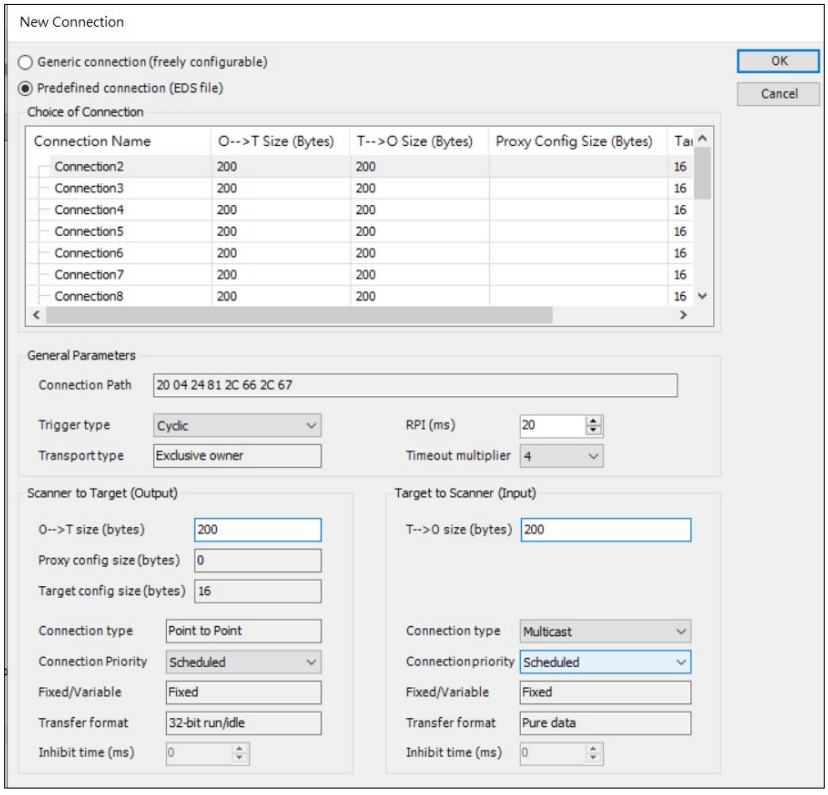


The configurable data will be defined in the EDS file and transmitted when the connection with the adapter is opened.

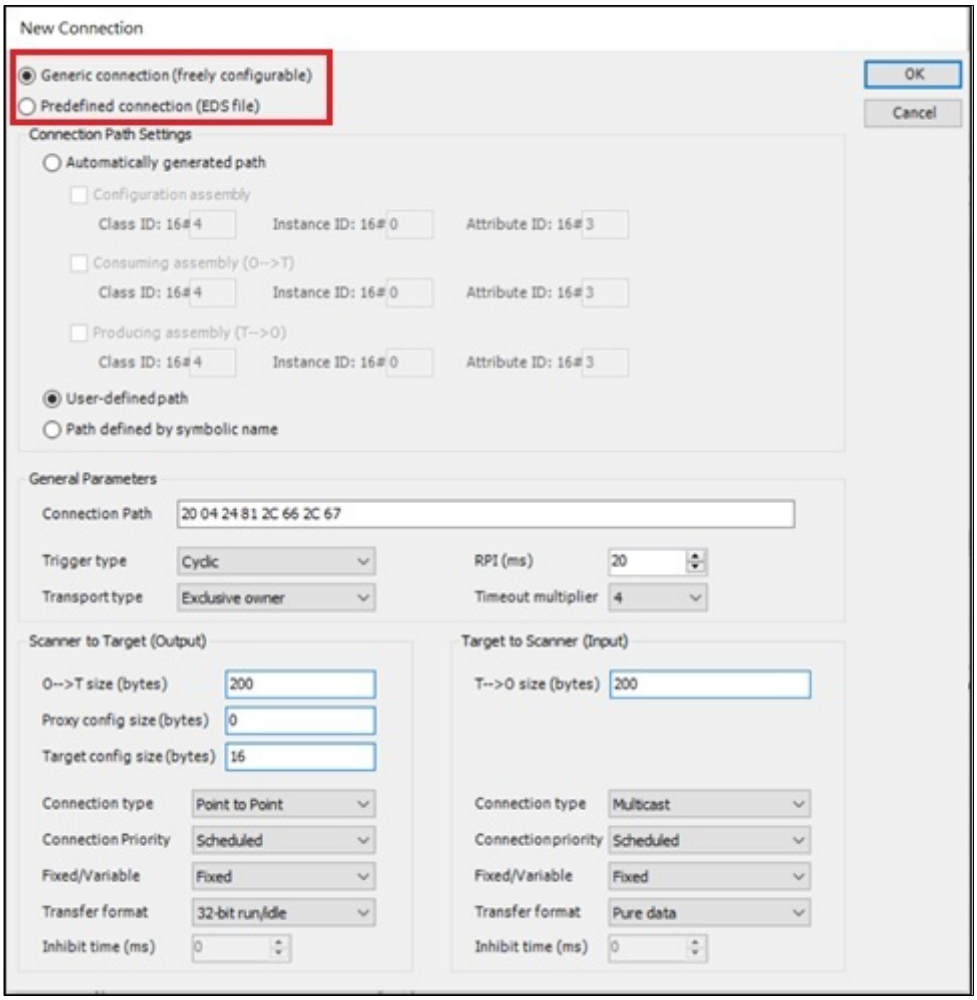
Project	Description
Connection name	Connection name
RPI (ms)	Requested packet interval: the exchange interval of input/output data
O → T size (Bytes)	The size of producer data from scanner to adapter （Originator → Target）
T → O size(Bytes)	Consumer data size from adapter to scanner （Target → Originator）
Proxy Config size (Bytes)	The size of the proxy configuration data
Target Config size (Bytes)	Target configuration data size
Connection path	The connection path data is represented as: address-configuration object-input object-output object

Project	Description
Add connection	Open the "New Connection" dialog box. Determine the parameters of the new connection here.
Delete connection	Remove the selected connection from the list
Edit connection	Open the Edit Connection dialog box. The parameters of the existing connection are modified here

- **Add connection**: Click **Add Connection**, the following dialog box will display,which contains the parameters of the new connection.



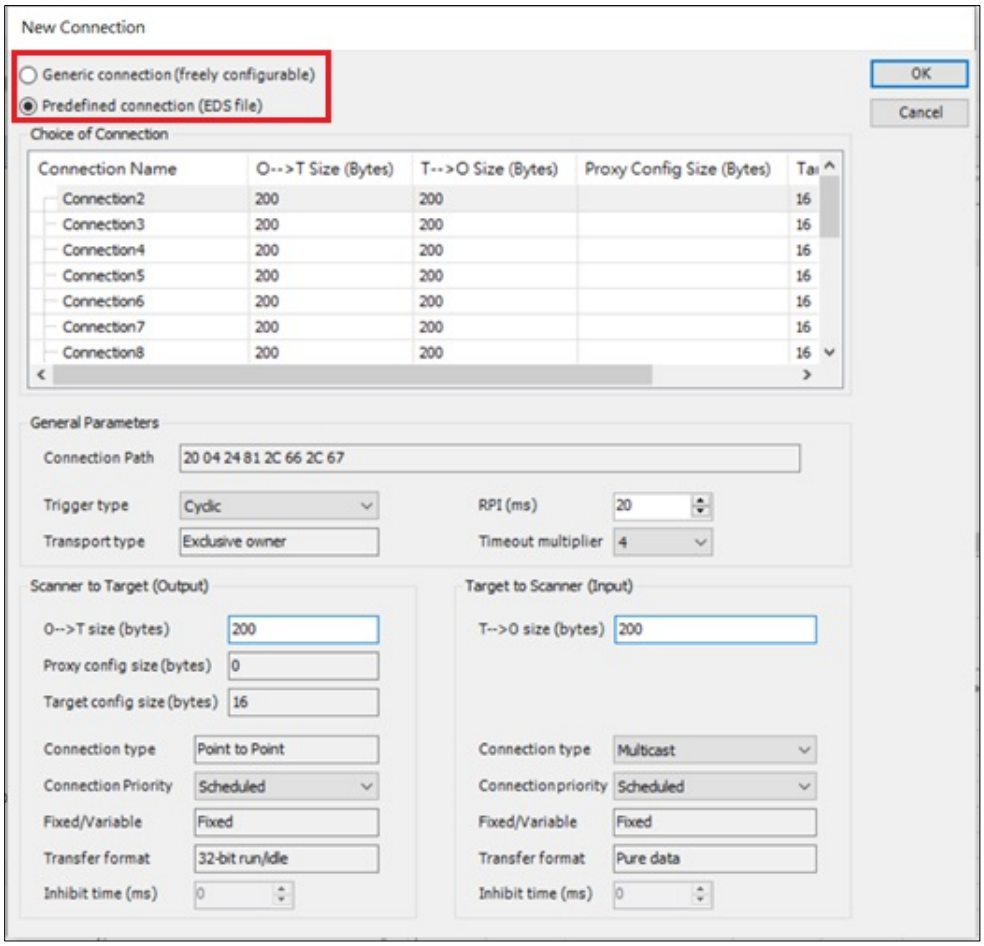
- **Generic connection** (free configuration): The dialog contains the parameters of the new connection.



Project	Description
Connection Path Settings	
Automatically generated path	The connection path is automatically generated based on the combined configuration, combined consumption and combined production values.
User-defined path	The connection path is manually specified in the corresponding input field.
Path defined by symbolic name	The path is specified by the symbolic name. ***Note**: The connection path setting must support symbolic connection path.*
General Parameters	

Project	Description
Connection Path	The connection path is used to address one or more objects in the adapter that provide input data and receive output and configuration data. ***Note**: The connection path is set to a custom path.*
Path defined by symbolic name	Use ANSI strings instead of ordinary connection paths. For the allowed ANSI strings, please refer to the corresponding EtherNet / IP adapter manual. ***Note**: The connection path is set as the connection label.*
Trigger type	• Cyclic: periodically exchange data at intervals set by RPI. • Status change: After changing the scanner output or adapter input, data will be exchanged automatically. • Application: Not implemented
Transport type	For details, please refer to the CIP specification.
RPI (ms)	The time interval (in milliseconds) at which the transmitting application requests data to be transmitted to the target application. The value must be a multiple of the bus cycle task.
Timeout multiplier	If the device fails, there will be a time delay (RPI *timeout multiplier) before the device status switches to **Error**.

- **Pre-defined connection (EDS file**): Use this option to use the existing connection in the EDS file, and the user can change the configuration data defined in the EDS file.



Project	Description
Scanner to Target (Output)	
O → T size (Bytes)	The size of producer data from scanner to adapter（Originator → Target）
Proxy Config size (Bytes)	The size of the proxy configuration data
Target config size (Bytes)	Target configuration data size
Connection Type	• Empty: no network connection is established. • Multicast: The network connection has been established. Connection data can be received by multiple user.

Project	Description
	• Point-to-point: A network connection has been established. Connection data can only be received by one user.
Connection Priority	Using two scanners with different priorities for a target may cause conflicts. Adjusting the connection priority can solve this problem.
Fixed/variable	For detailed information on parameters, see CIP specifications.
Transfer format	Conversion format
Inhibit time	Prohibited time
Heartbeat multiplier	Extend the time interval for the scanner to send heartbeat messages to the adapter. This value is multiplied by the RPI value. **Example**: RPI = 10ms, and heartbeat multiplier = 10 causes a message to be sent every 100ms. ***Note**: The transmission format is Heartbeat*
Target to Scanner (Input)	
T → O Size (bytes)	Consumer data size from adapter to scanner（Target –> Originator）
Connection Type	• Empty: No network connection established. • Multicast: A network connection has been established. Connection data can be received by multiple user. • Point-to-point: A network connection has been established. Connection data can only be received by one user.

Project	Description
Connection Priority	Using two scanners with different priorities for a target may cause conflicts. Adjusting the connection priority can solve this problem.
Fixed/variable	Fixed/variable
Transfer format	Conversion format
Inhibit time	Prohibited time

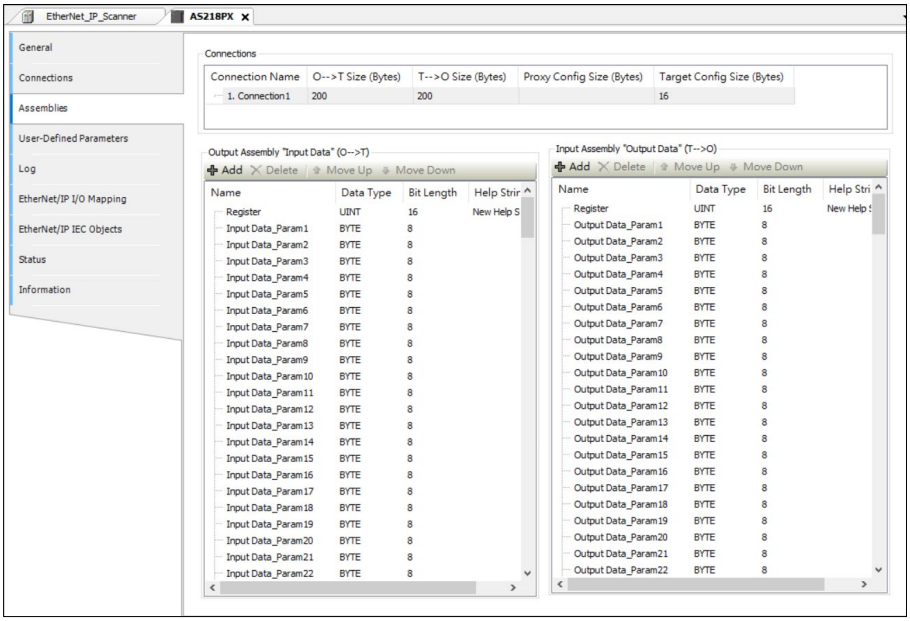
- **Configuration data**

  On the connection page, under the configuration data, the connection with the configuration parameters in the EDS file is displayed. The connections are subdivided into configuration groups.

Project	Description
Raw data value	If scaling parameters are defined for the data in the EDS file, the value can be displayed as raw data or converted data. • Startup: Display data without any conversion. For the Enum data type, the index of the enumeration value will be displayed. • Not started: Display data and convert. For the Enum data type, the enumeration value will be displayed.
Display parameter group	Display parameter group
default	Set as default
value	Double-click to change the value. According to the data type, user can specify the value directly in the input field, or user can select from the drop-down list.

Assemblies tab

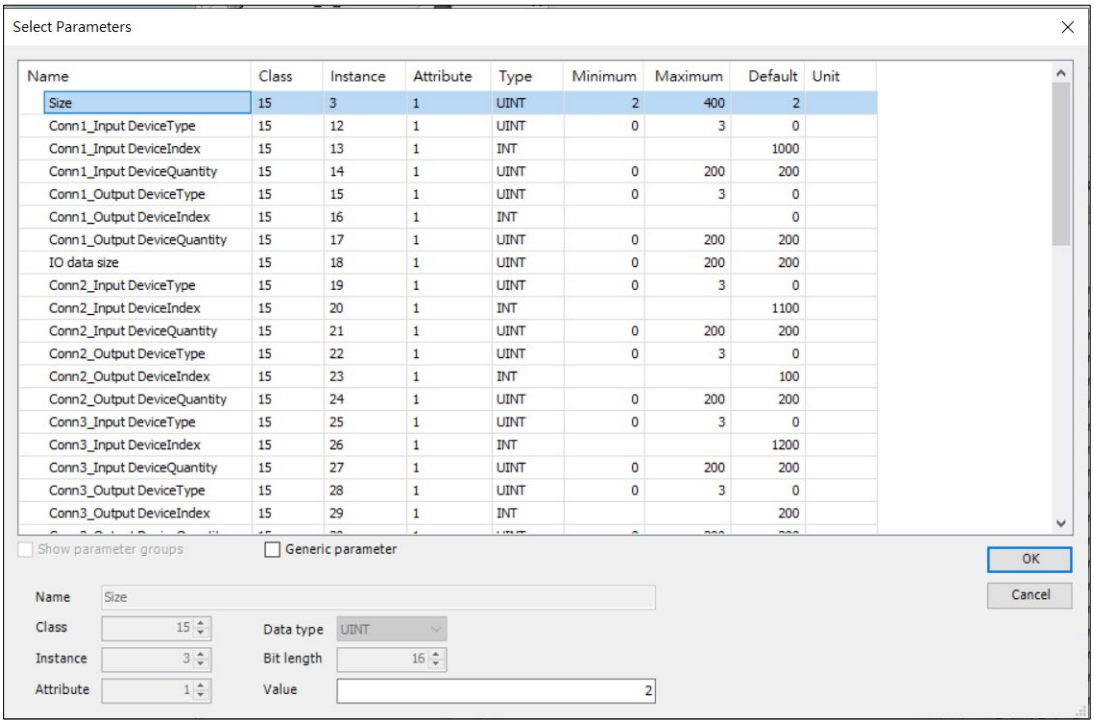
The upper half of the screen on this tab displays a list of all configured connections. After selecting the connection, the relevant data configuration will be displayed in the lower part of the screen.



- **Output component Input Data / Input Component Output Data**

Project	Description
Add	Open the select parameter dialog box, select the parameter to be added.
Delete	Delete selected parameter.
Move Up/Move Down	Move the selected parameter up/down in the list, the order in the list determines the order in the I/O mapping.
Name/Data Type/ Bit Length/Help String	These values can be changed by double-clicking on the text field.

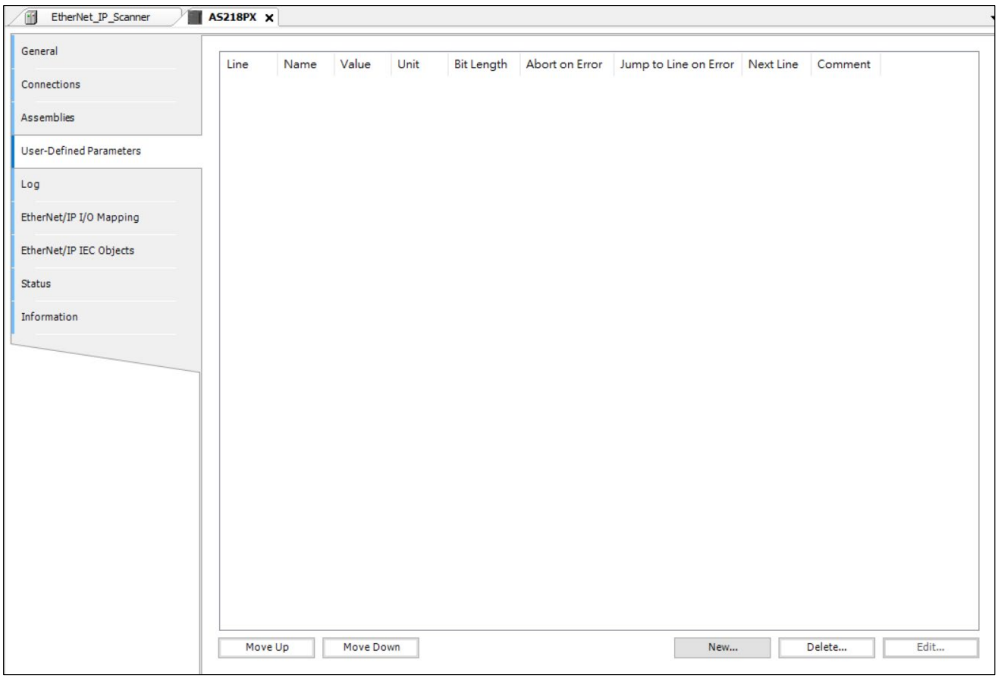
- **Export/Import Components>Add > Select parameter window**



Project	Description
Show parameter groups	• Start: This dialog box displays all parameters in the EDS file according to the group. • Not activated: This dialog box displays all parameters in the EDS file according to the structure. By clicking **OK**, user can select each parameter in this list and add it to the parts list.
Generic parameters	User can add common parameters and edit the values of the parameters.

User-defined Parameters tab

This tab shows all other parameters that are only transferred to the bus system during the phase of the startup process assigned to it. When the connection is re-established (for example, after the remote adapter fails), the user parameters will be sent again.

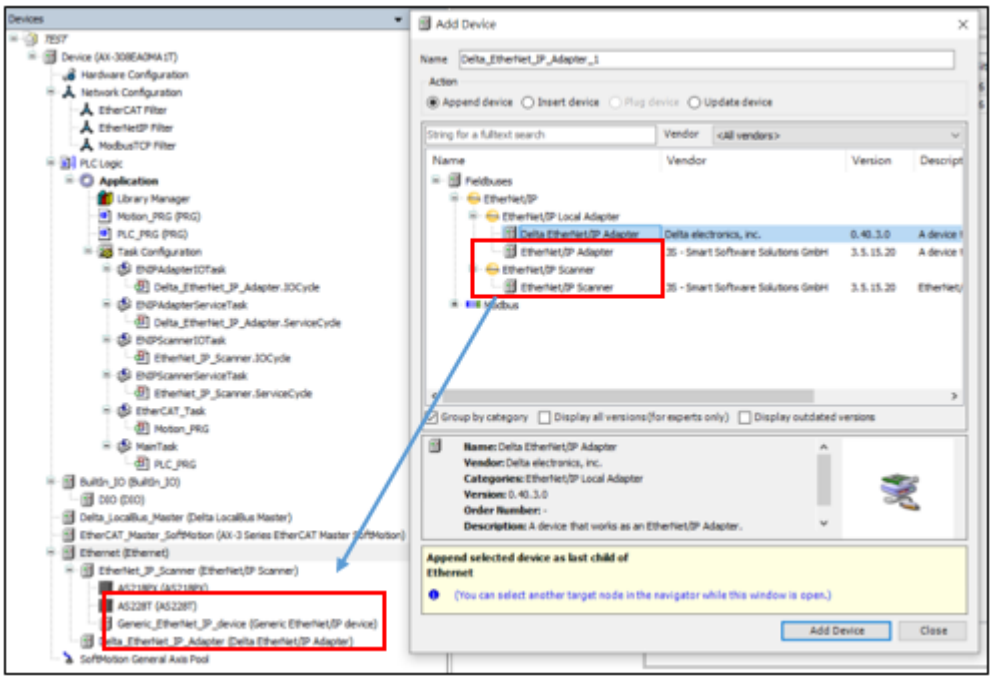


Project	Description
New	Open the **Select Parameters** dialog to add new parameters. The new parameter will be inserted in front of the selected row.
Delete	Select the parameter to be deleted to delete
Edit	Open the **Select Parameters** dialog to change existing parameters.
Move up/Move Down	Change the order of user parameters. The order of the parameters in the list corresponds to the order of initialization.
Value	User can directly change the value of the corresponding parameter by double-clicking the value.
Abort on error	Start: When an error occurs, the complete transmission of parameters is aborted.
Jump to the line on Error	When an error occurs, the program will continue from the line specified in the **Next Line** column. Therefore, complete blocks can be skipped during initialization, or a return can be defined.

Project	Description
	***Note**: If user can never write a specific parameter, returning will result in an infinite loop.*

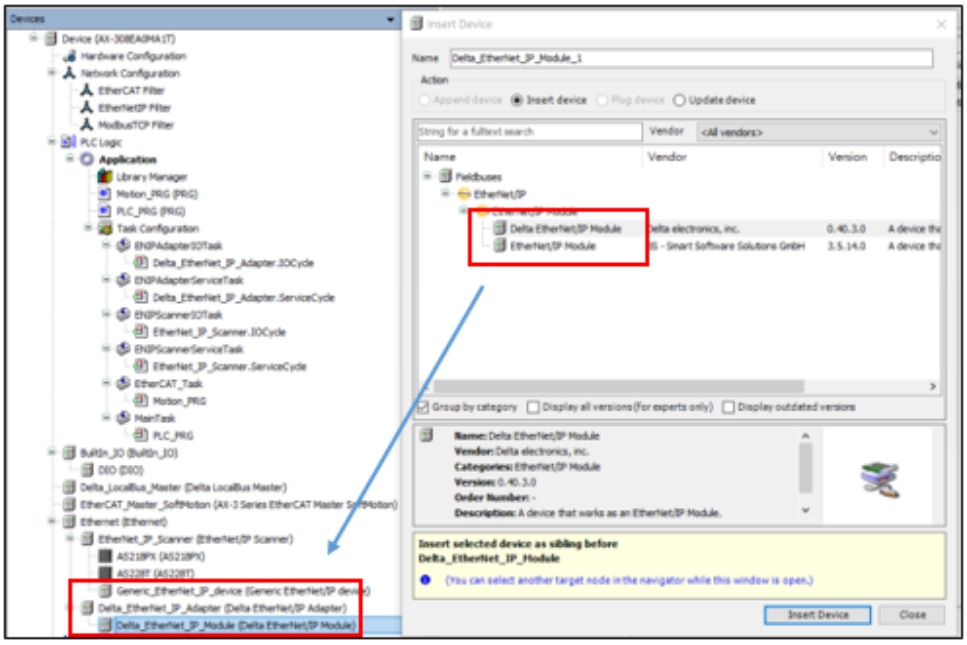
5.2.7.3 EtherNet/IP Local Adapter

AX-3 series supports EtherNet/IP local adapter function. Right-click on the **EtherNet node** > **Add Device** > **Fieldbus** > **EtherNet/IP Local Adapter** > **Delta EtherNet/IP Adapter.**

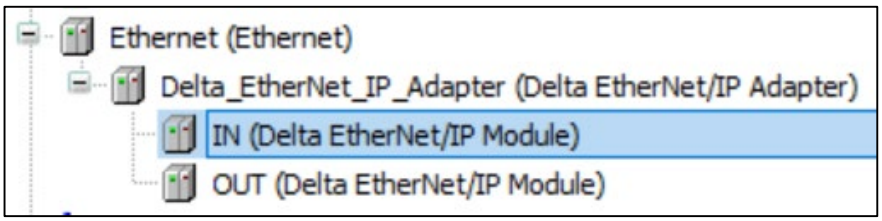


Follow the procedure to set the AX-3 series adapter:

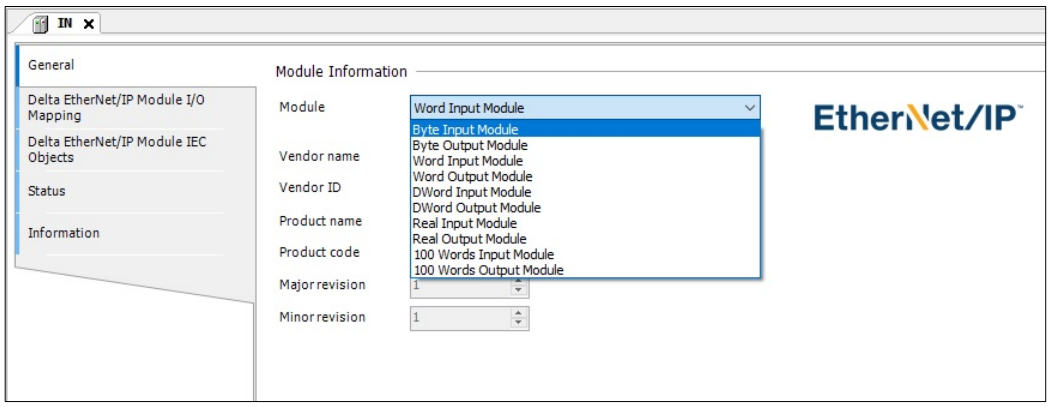
1. Right-click on **Add device** > **Fieldbus** > **EtherNetIP Module** > **Delta EtherNetIP Module**。



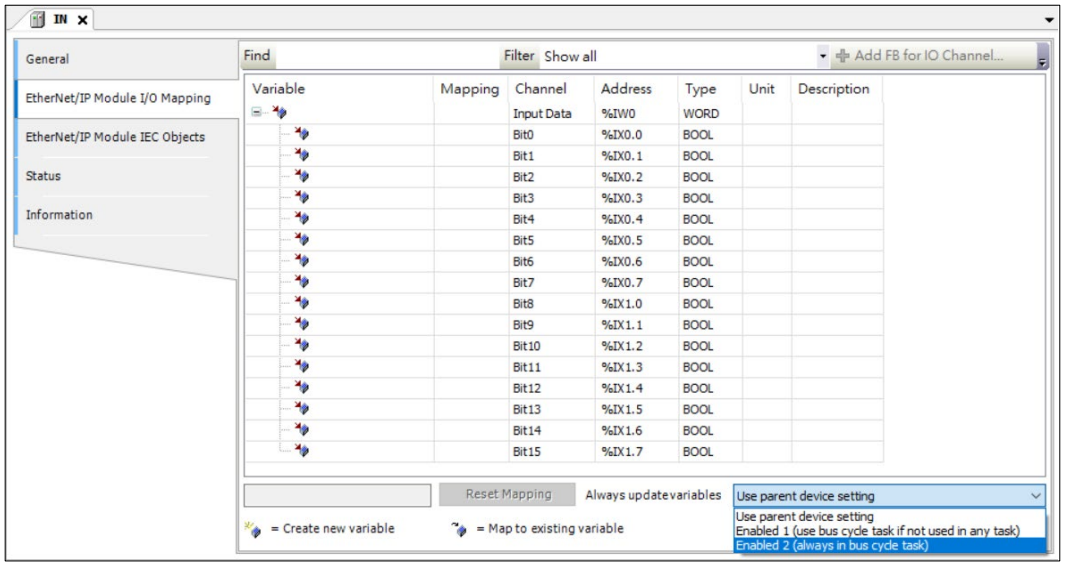
2. Repeat the previous step to add 2 Delta EtherNet/IP Modules and change the names to **IN** and **OUT**. (Right-click **Delta EtherNet/IP Module** > **Properties** > Modify the name in the **General** tab).



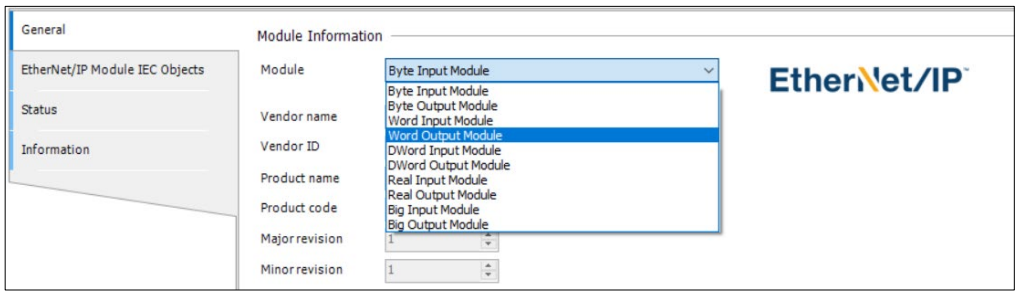
3. Double-click on **IN (Delta EtherNet/IP Module)** > **General tab** > **Module** > **Word Input Module.**



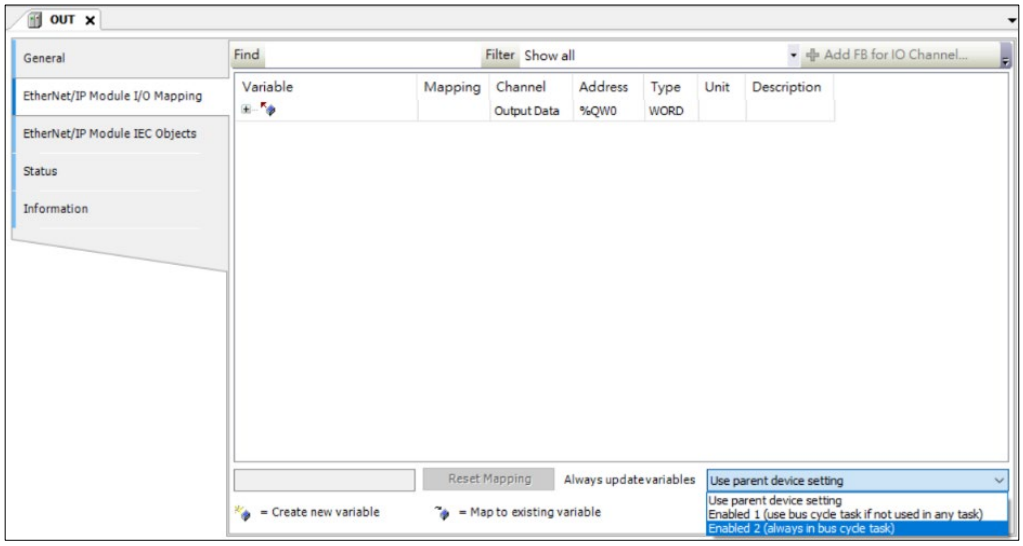
4. **EtherNet/IP Module I/O Mapping tab** > **Always update variables** > **Enabled 2 (always in bus cycle task)**.



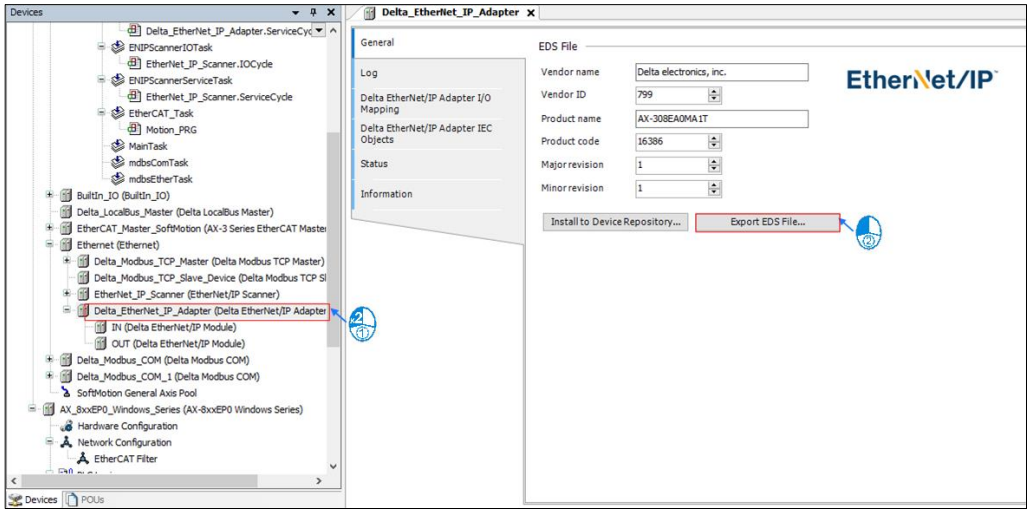
5. Double-click on **OUT (EtherNet/IP Module)** > **General tab** > **Module** > **Word Output Module.**



6. **EtherNet/IP Module I/O Mapping** > **Always update variables** > **Enabled 2 (always in bus cycle task)**.

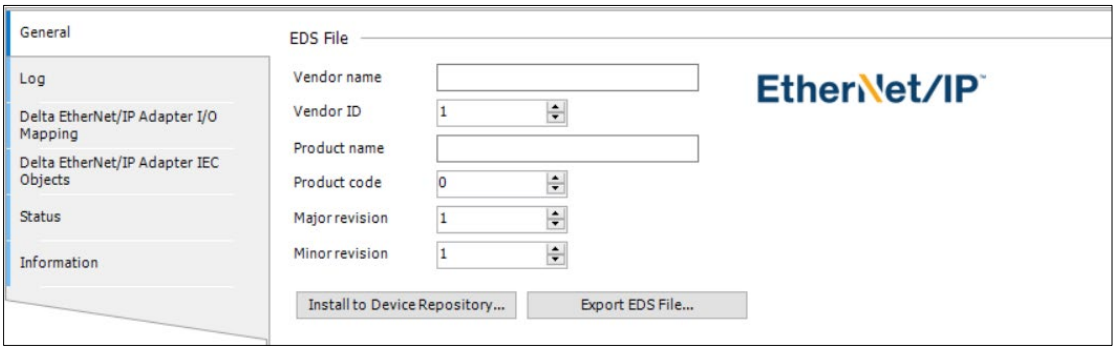


7. Double-click **EtherNet_IP_Adapter (EtherNet/IP Adapter)** in the project tree to open the setting page. In the **General** tab, click **Export EDS File** to export the EDS file to a third-party configuration environment.



Delta EtherNet/IP Local Adapter General tab

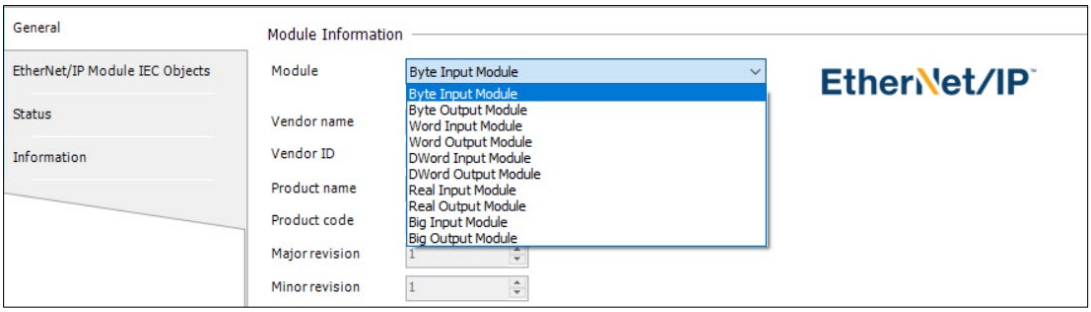
The local adapter displays the general information in the device description file, and the user can define this information to export the EDS file.



Project	Description
Vendor name	Supplier name
Vendor ID	Provided by ODVA Association
Product name	EDS file name
Product Code	Product Code
Major revision	Major revision
Minor revision	Minor revision
Install to Device Repository	If a device with the same device ID is already installed, it will ask if the device should be overwritten. If the device is plugged under the EtherNet/IP scanner as a remote adapter, user will be asked to update the device automatically.
Export EDS file	The EDS file will be created and stored on the local computer. This can use EDS files in an external setting environment.

EtherNet/IP Module General tab

Supports modules of different data types. After setting multiple data type modules, return to the local adapter to output EDS files.


General
EtherNet/IP Module IEC Objects
Status
Information
Module Information
Module
Byte Input Module
Byte Input Module
Byte Output Module
Word Input Module
Word Output Module
DWord Input Module
DWord Output Module
Real Input Module
Real Output Module
Big Input Module
Big Output Module
Vendor name
Vendor ID
Product name
Product code
Major revision
1
Minor revision
1
EtherNet/IP

5.2.8 High-Speed IO Setting

Note: This section does not apply to AX-300NA0PA1.

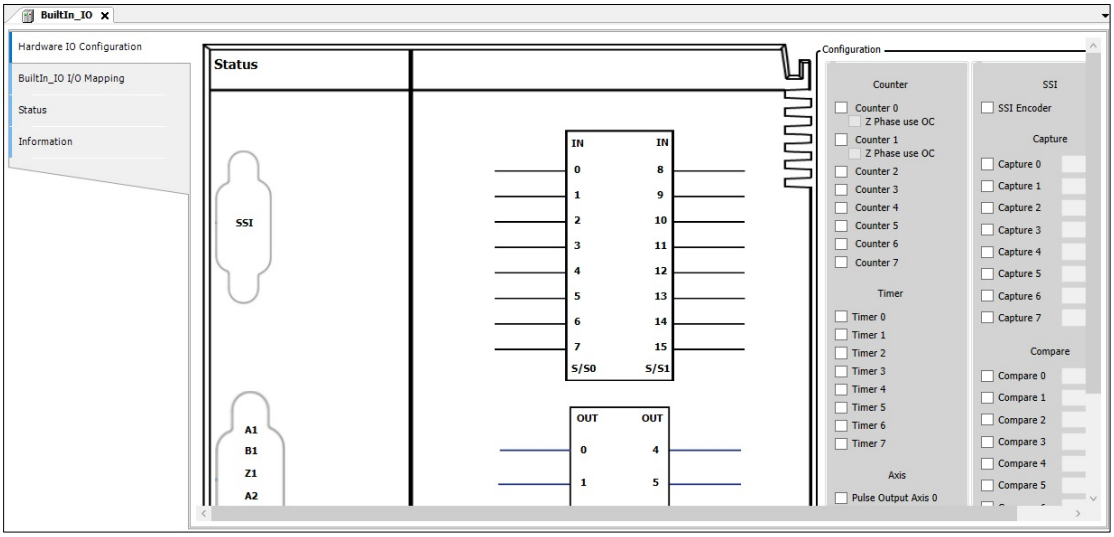
The following table is a comparison table of high-speed IO configurations:

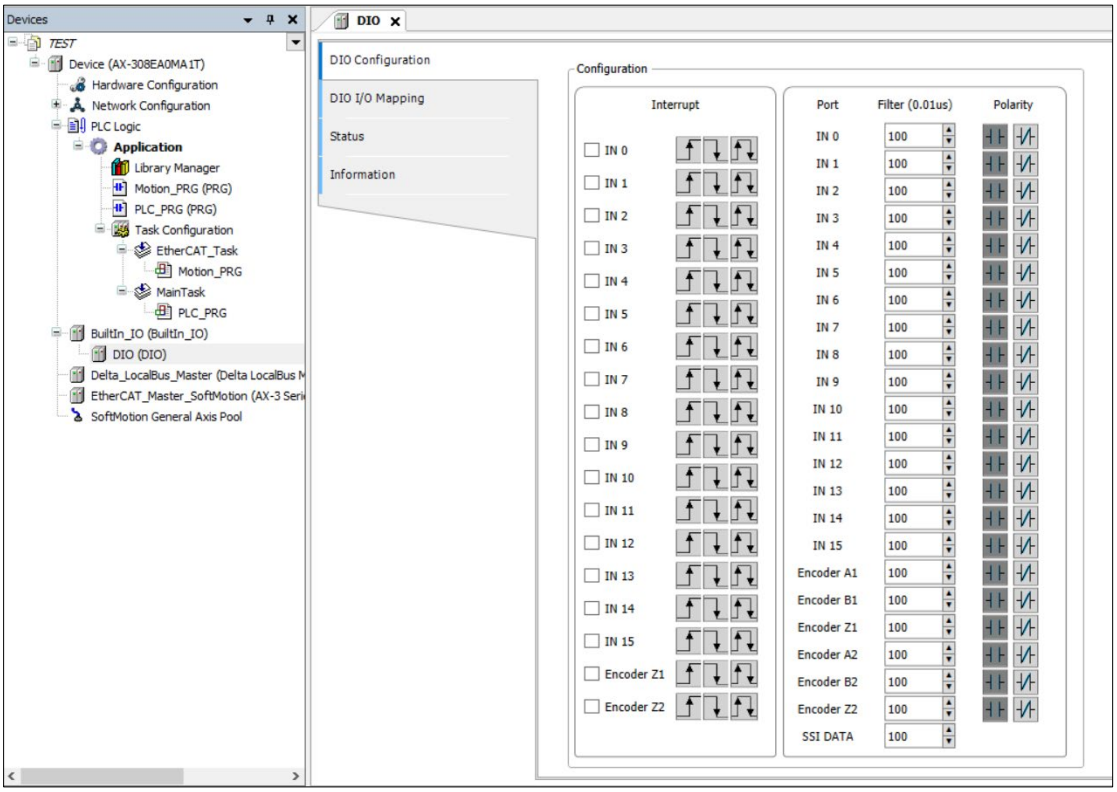
	AX-324/AX-304	AX-364EL/AX-308/AX-316	Description
DIO Configuration	✓	✓	AX-324/AX-304 lacks a differential encoder
SSI	-	✓	-
Counter/Timer	6 groups	8 groups	AX-324/AX-304 lacks a differential encoder
Capturer/Comparator	✓	✓	-
Pulse output shaft	-	✓	-
Interrupt	✓	✓	AX-324/AX-304 is missing Channel 16 & 17

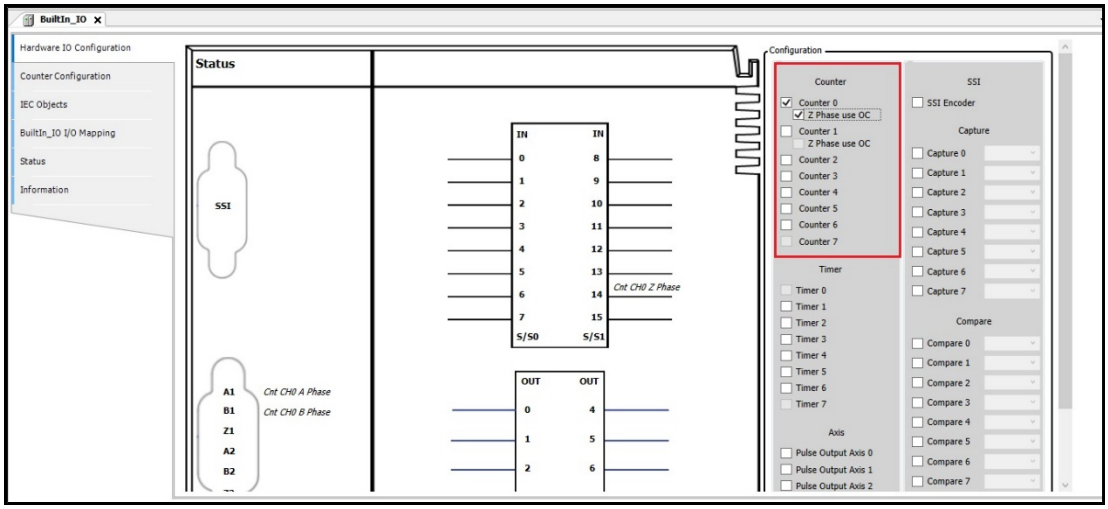
5.2.8.1 High-speed IO setting (AX-364EL/AX-308/AX-316/AX-332)

5.2.8.1.1 BuiltIn IO: Hardware IO configuration

Double-click **BuiltIn_IO** from the device tree to configure hardware IO.







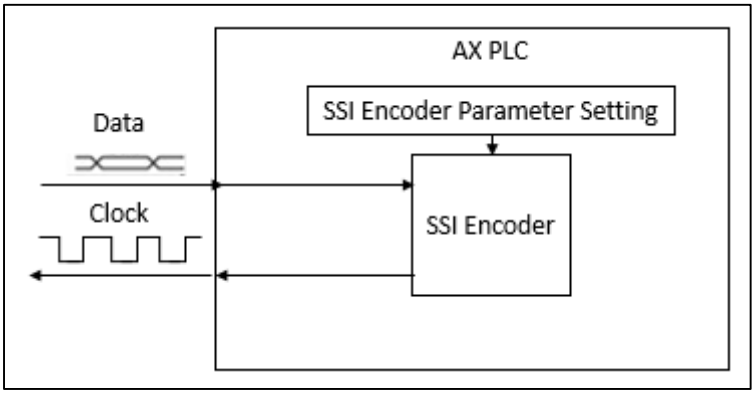
High-speed I/O on the host provides the following function settings:

- DIO: Can set interrupt, filter, polarity and other functions.
- SSI Encoder: Can set SSI encoding type, clock frequency, SSI data length and other functions.
- Counter: Can set high-speed counter variable and counting mode, whether Z-phase signal is enabled or other functions or high-speed timer variable declaration.
- Capture/Compare: User can declare the Variables of the high-speed capture or high-speed comparators.
- Pulse Output: user can set the pulse output mode, direction, and origin return mode and parameters.

5.2.8.1.2 BuiltIn IO: SSI Encoder configuration

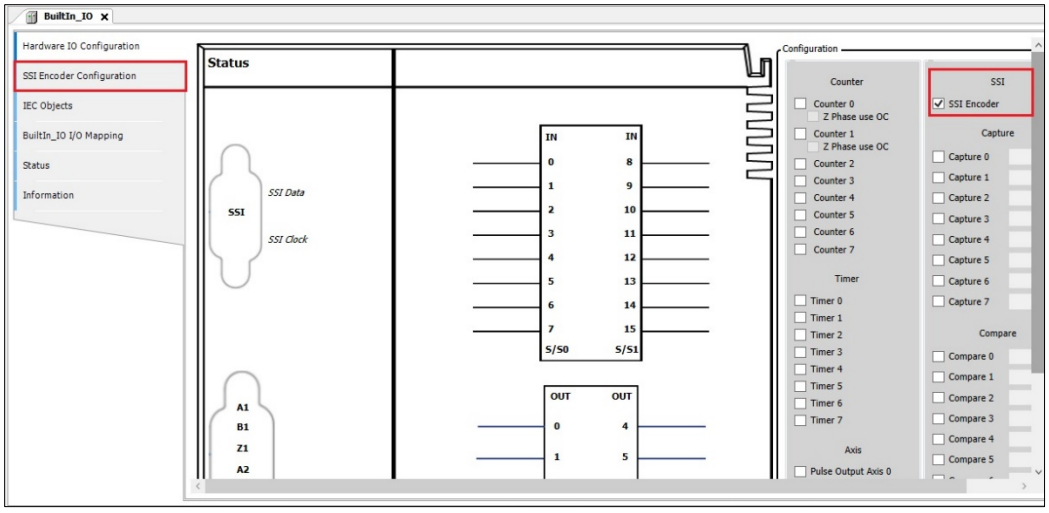
The AX controller body supports a set of SSI encoder functions on the IO end. It is connected to the PLC through the D-SUB interface. This interface provides 5V power output for the encoder power. The parameters required by the encoder can be received through the hardware configuration channel.

1. SSI Encoder architecture.

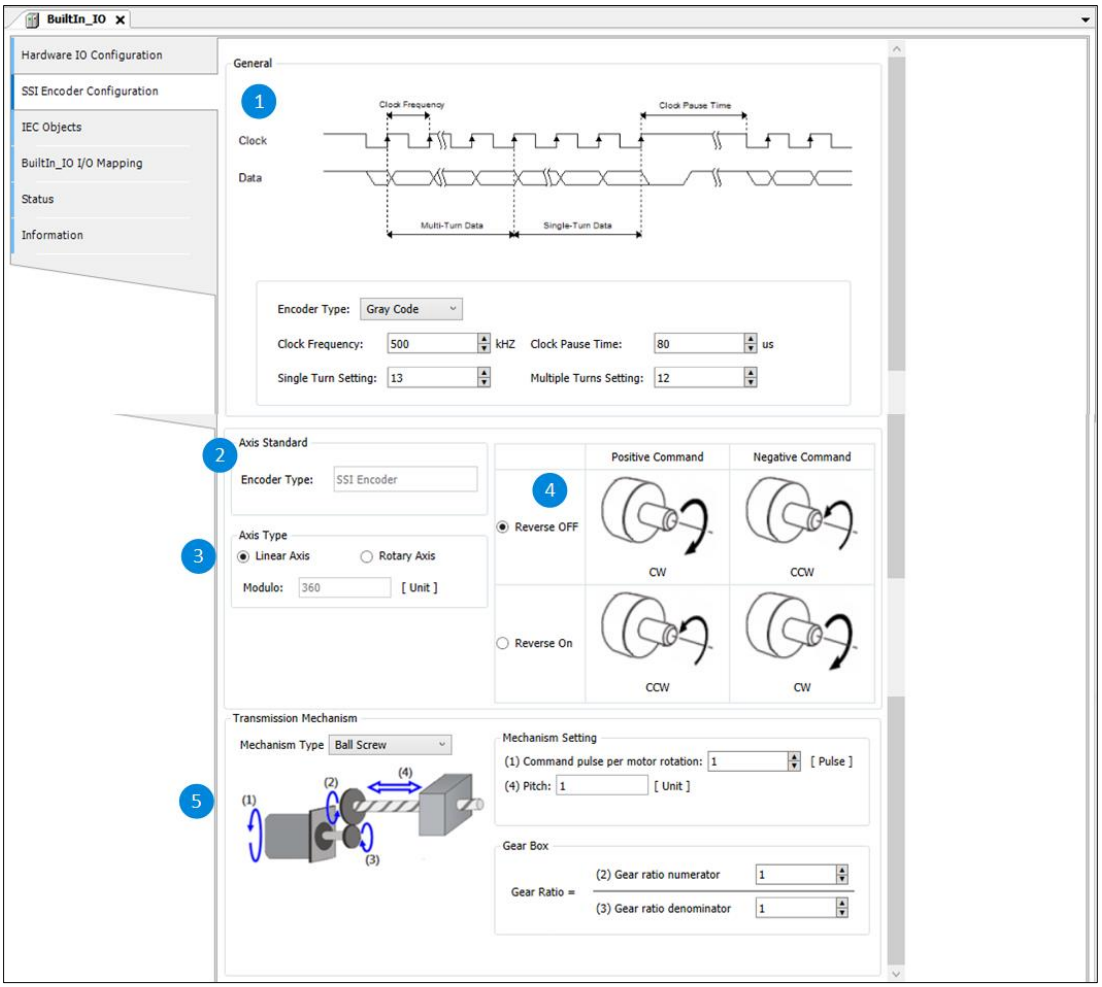


2. SSI Encoder function activated

   In the **BuiltIn_IO** screen, click **SSI Encoder Configuration** and then select **SSI Encoder Configuration** tab.



SSI related settings can be set in the **SSI Encoder Configuration** screen. The following explains the settings:



① General

Name	Features	Set value (default value)
Encoder Type	SSI Encoder type	Gray code/ Binary code (Gray code)
Clock Frequency	Frequency of the transmission clock signal used for SSI encoder communication (requires the SSI encoder data sheet as a reference)	(500)
Single Turn Setting	Data length of single turn data (Need to use the SSI encoder data sheet as a reference)	(13)
Multiple Turns Setting	Data length of multi-turn data	(12)

Name	Features	Set value (default value)
	(Need to use the SSI encoder data sheet as a reference)	
Clock Pause Time	After the last falling edge of the clock, the data line will remain low for a period of time until the data line is pulled high (Need to use the SSI encoder data sheet as a reference)	(80)

② Axis Standard

Name	Features	Set value (default value)
Encoder Type	Encoder type display	-

③ Axis Type

Name	Features	Set value (default value)
Linear Axis/ Rotary Axis	Set linear or rotary axis	Linear Axis Rotary Axis (Linear Axis)
Modulo	When the rotation axis is selected, the value of one rotation of the rotation axis can be set.	(360)

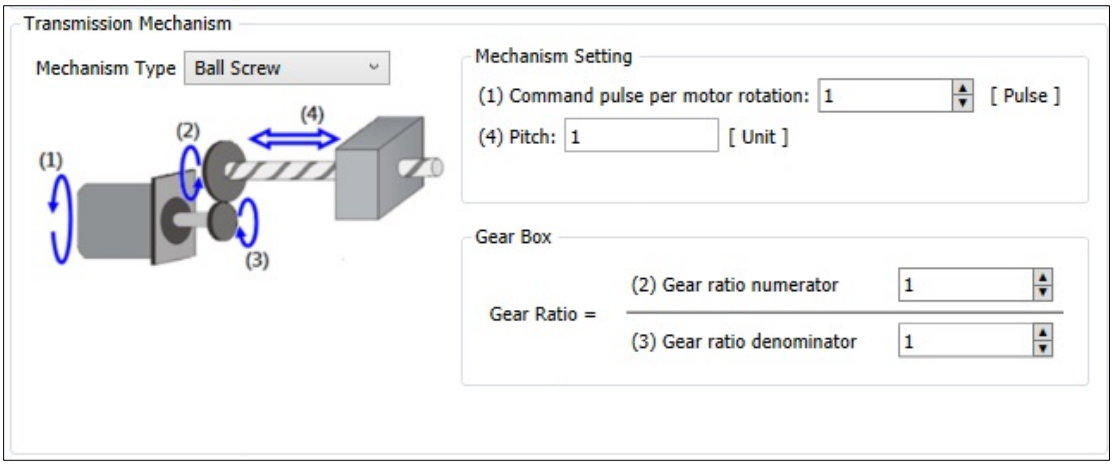
④ Positive/Negative Command

Name	Features
Reverse OFF/On	Set forward and reverse commands

⑤ Transmission Mechanism

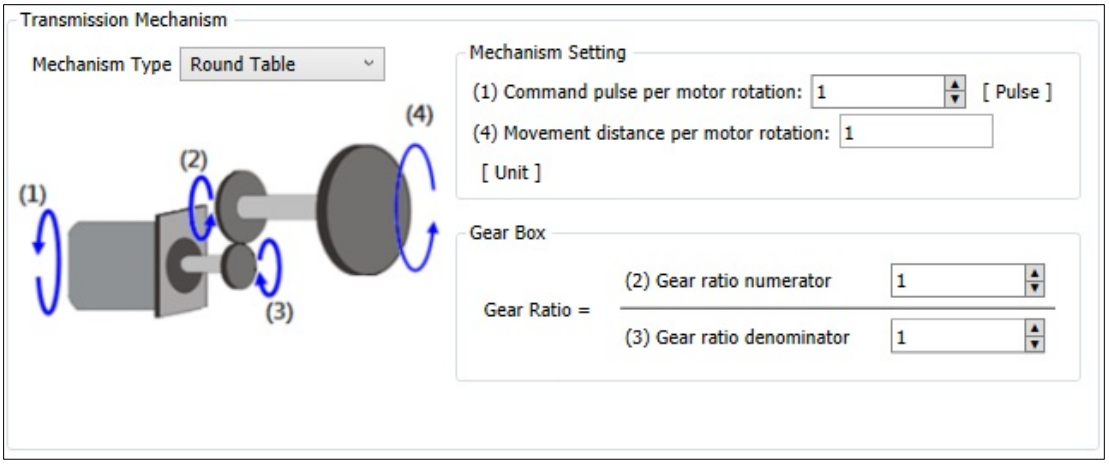
The following are introduced for different architectures:

Ball Screw



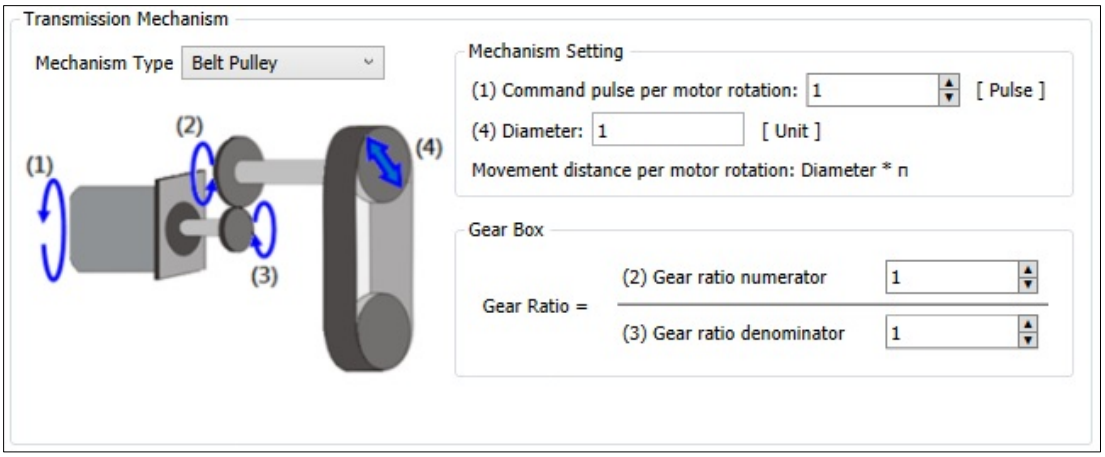
Legend	Features
(1)	Command pulse per motor rotation
(4)	Pitch
(2)	Gear ratio numerator
(3)	Gear ratio denominator

Round Table



Legend	Features
(1)	Command pulse per motor rotation
(4)	Movement distance per motor rotation
(2)	Gear ratio numerator
(3)	Gear ratio denominator

Belt Pully

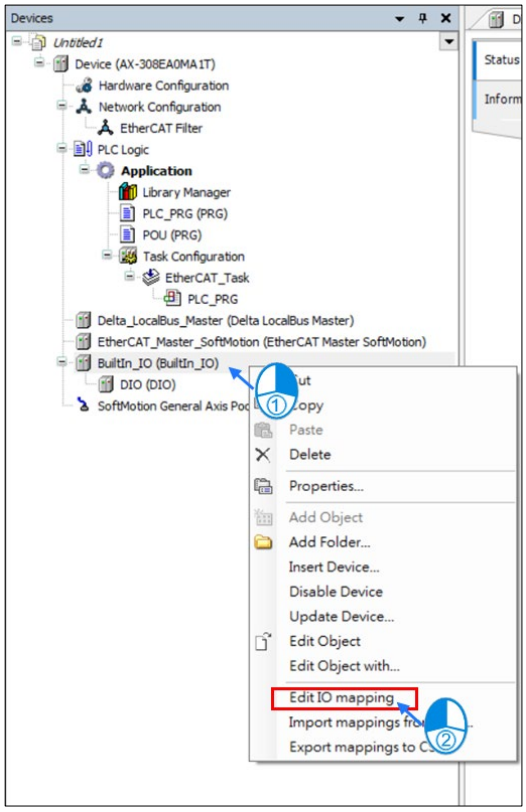


Legend	Features
(1)	Command pulse per motor rotation

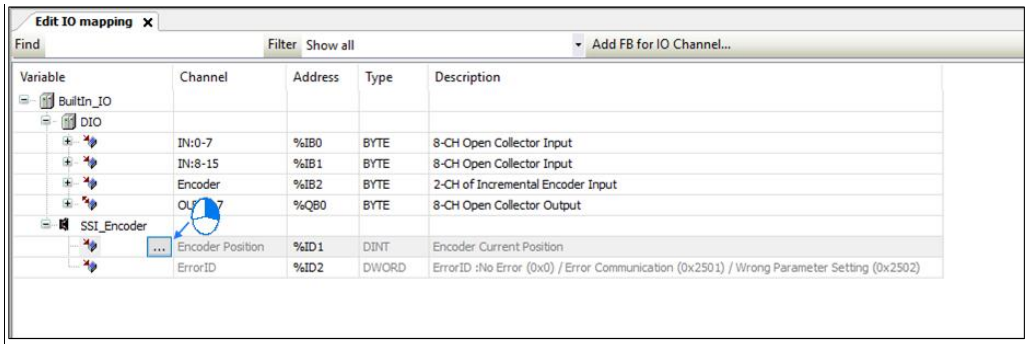
Legend	Features
(4)	Diameter (Movement distance per motor rotation: Diameter * n)
(2)	Gear ratio numerator
(3)	Gear ratio denominator

Follow the steps to set up SSI Encoder Mapping variable.

1. Right-click **BuiltIn_IO**, select **Edit IO Mapping**.



2. In the **Edit IO mapping** screen, click ... to add variables.



Name	Content
Encoder Position	SSI Actual location
ErrorID*Note	SSI Encoder Communication status, the status is as follows: 0 : No Error 1 : Error Communication 2 : Wrong Parameter Setting

Note:

- ErrorID *:*

  ErrorID = 1 when SSI Encoder is not connected or the SSI Encoder is disconnected from the CPU.

  When MultiTurns + SingleTurns is greater than 32, ErrorID = 2

  The above error conditions will cause BusCycle to stop updating the EncoderPosition, and the EncoderPosition will maintain the previous value. The purpose is to prevent other slave axes from bouncing if the Encoder has a synchronously moving spindle.

- ErrorID Clear :

  When the SSI Encoder is not connected or the SSI Encoder is disconnected from the CPU, Status Data = 1, BusCycle will stop updating, and the EncoderPosition will maintain the previous value. The purpose is to prevent other slave axes from occurring if the Encoder has a synchronously moving spindle beat:

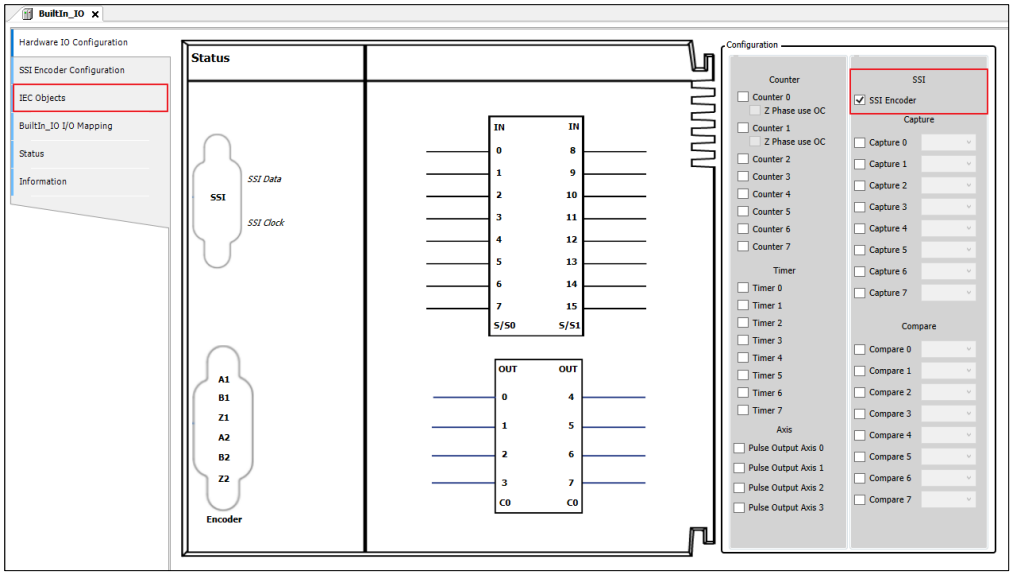
  Check the connection between the SSI Encoder and the CPU. After the correction, the firmware will determine that the connection communication channel is correct and the BusCycle's EncoderPosition update will resume. There may be many reasons for the error, such as: SSI Encoder not plugged in, SSI Encoder broken, Drive board Abnormal.

  When MultiTurns + SingleTurns is greater than 32, Status Data = 2 : Download the parameters of MultiTurns + SingleTurns within 32.

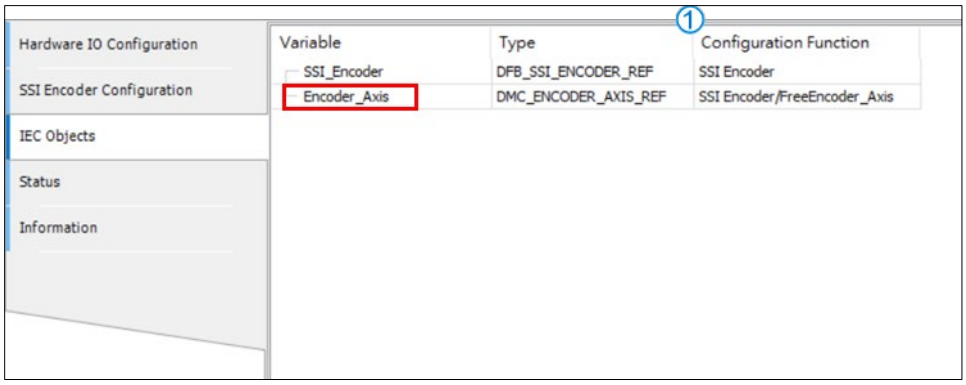
- *SSI Encoder Used in programs :*

The encoder axis variable that comes with the SSI encoder device. This variable is used by the user's motion function block in the POU (**Example**: MC_CamIn).

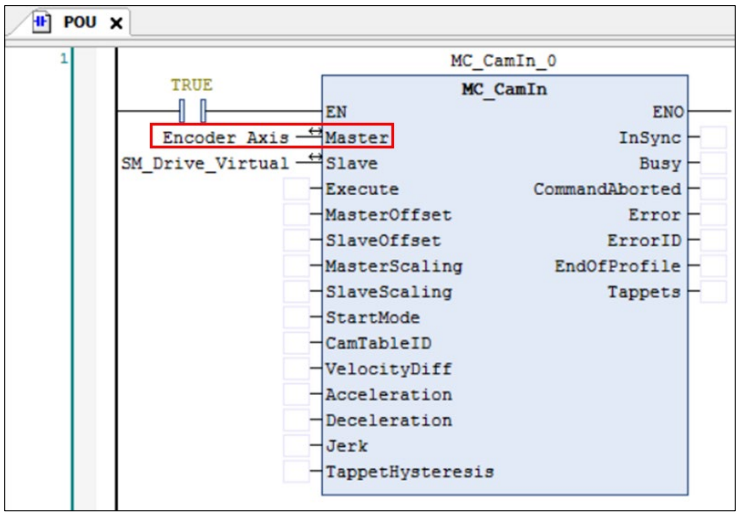
1. In **BuiltIn_IO** sceen, click the **IEC Objects** tab.



2. **IEC Objects** in the picture: Is an example of configuration function, and **Encoder_Axis** must be filled in for the axis name used in the POU.



3. For the **MC_CamIn** function block in the POU, the source of the spindle can use SSI as the source. The input axis name is Encoder_Axis.


POU
1
MC_CamIn_0
MC_CamIn
TRUE
EN
ENO
Encoder_Axis
Master
InSync
SM_Drive_Virtual
Slave
Busy
Execute
CommandAborted
MasterOffset
Error
SlaveOffset
ErrorID
MasterScaling
EndOfProfile
SlaveScaling
Tappets
StartMode
CamTableID
VelocityDiff
Acceleration
Deceleration
Jerk
TappetHysteresis

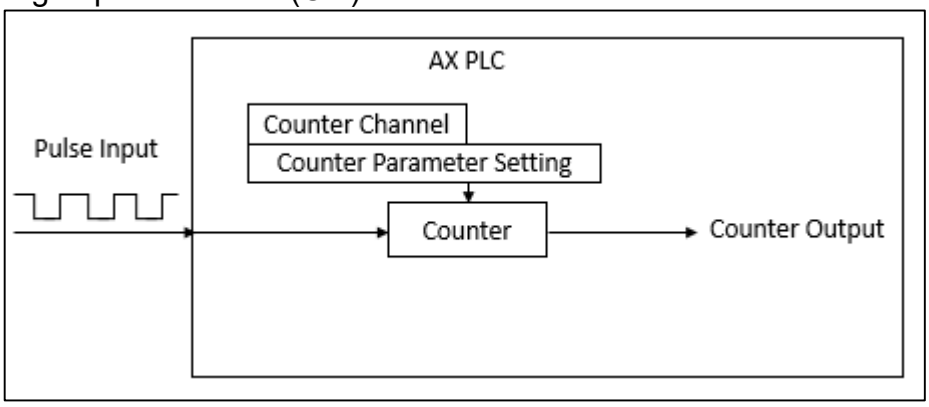
5.2.8.1.3 BuiltIn IO: Counter/timer configuration

The connection between AX-308/AX-364/AX-316 and pulse encoder supports differential input (2 groups) and open collector input (56 groups) interfaces. The differential interface is connected to the PLC through the D-SUB15 interface. This interface has 2 sets of high-speed counting The (timer) device can count the pulse value or frequency of the (timer) encoder; the external encoder with open collector input needs to be connected through the Input point on the IO board. There are 6 sets of high-speed counter (timer) in this area to count (Timing) The pulse wave value or frequency of the encoder, the user needs to check the pulse wave type encoder function to enable and set the parameters required by the encoder, and the encoder data can be received through the hardware configuration channel.

Note: AX-324 and AX-304 do not support differential input.

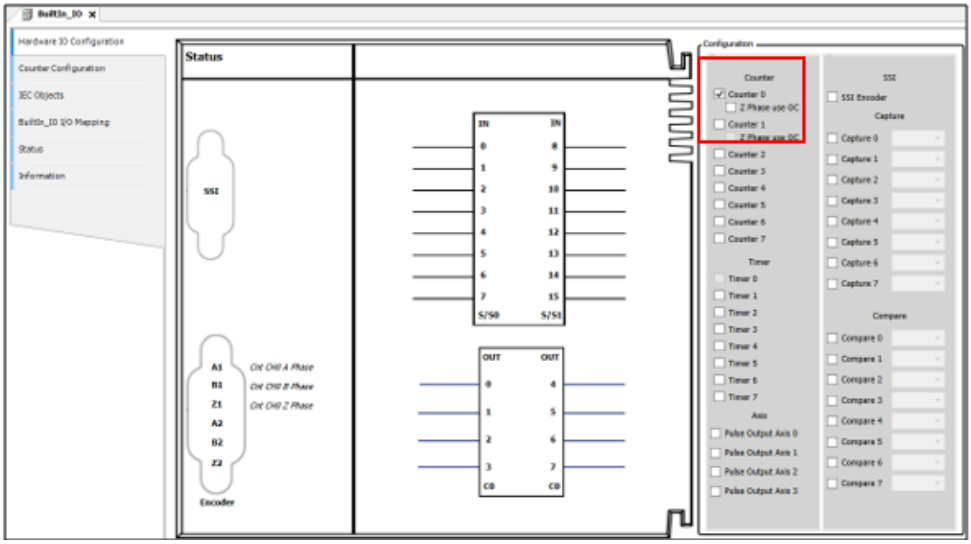
In this section, user can find the describtion of the pulse encoder function module of the body IO. The AX-308 supports up to 8 groups of high-speed counters and high-speed timers.

- High-speed counter (Cnt)

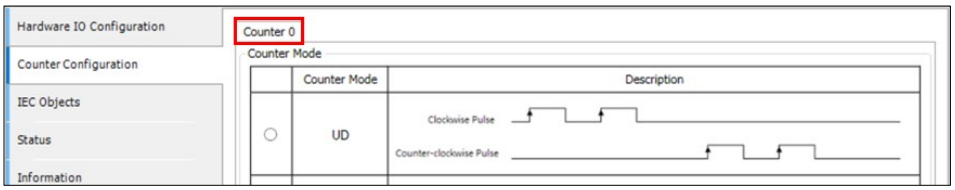


- High-speed counting function is activated.

1. In the **BuiltIn_IO** screen, there are 8 counters in total. Click **Counter 0** and then click **Counter Configuration**.

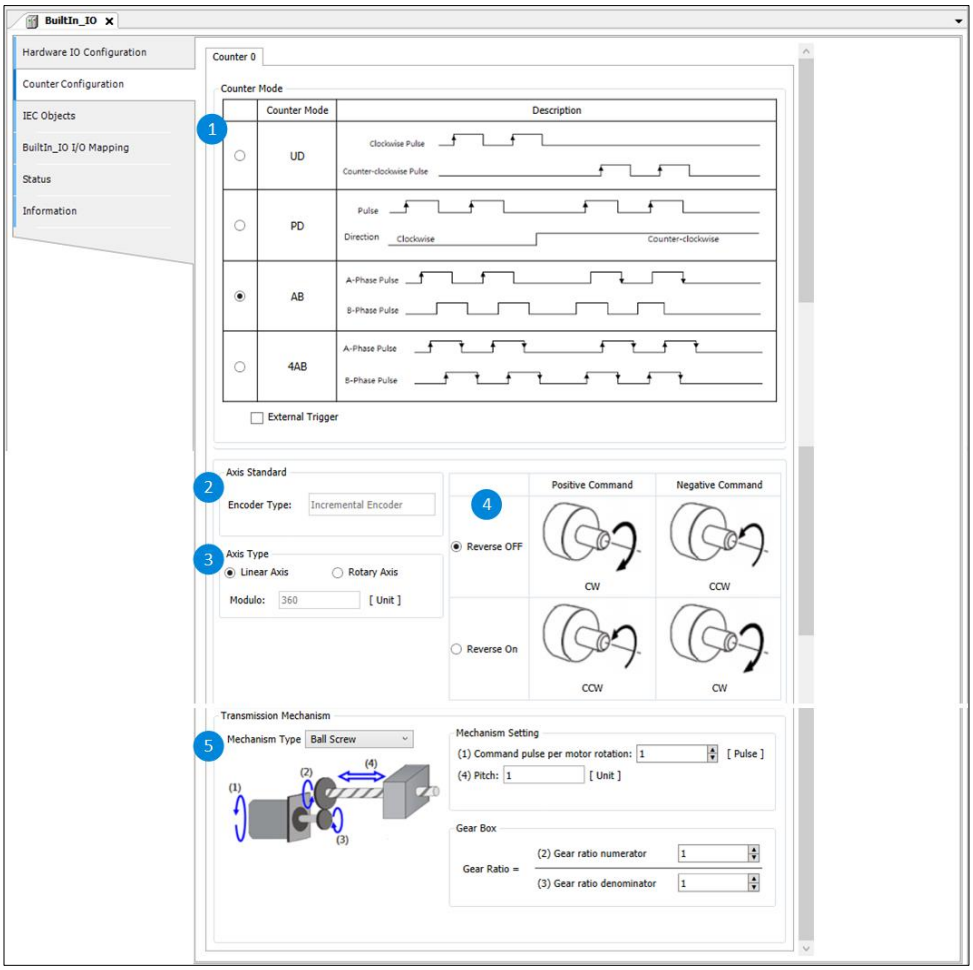


2. See the **counter 0** page in the **counter configuration** screen .



3. Counter related settings can be set in the **Counter Configuration** screen.

   The following explains the settings:



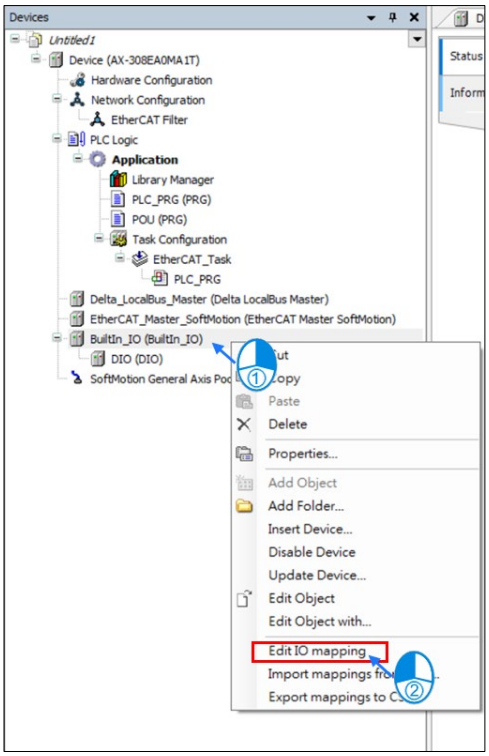
① Counter Mode

Name	Features
UD	Forward pulse train and reverse pulse train
PD	Pulse train + direction
AB	AB Phase pulse train
4AB	AB phase pulse train (4x)
External Trigger	Phase Z signal is enabled.

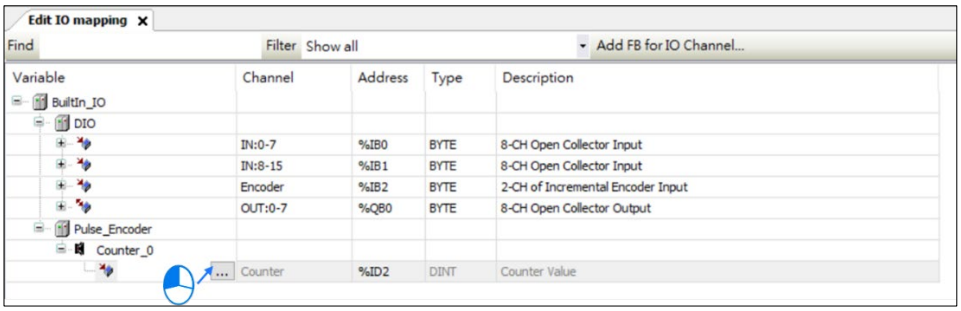
② ③ ④ ⑤ Software setting screen: refer to *SSI Encoder Function Settings*

Set Counter mapping variable

1. Right-click **BuiltIn_IO** and select **Edit IO Mapping**.



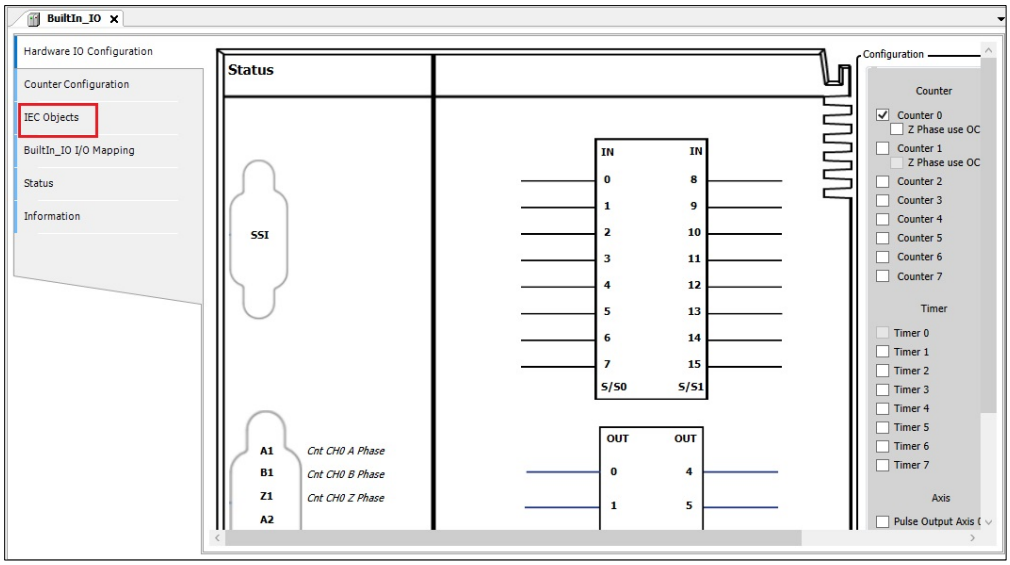
2. In the **Edit IO mapping** screen, click to ... add a new variable.



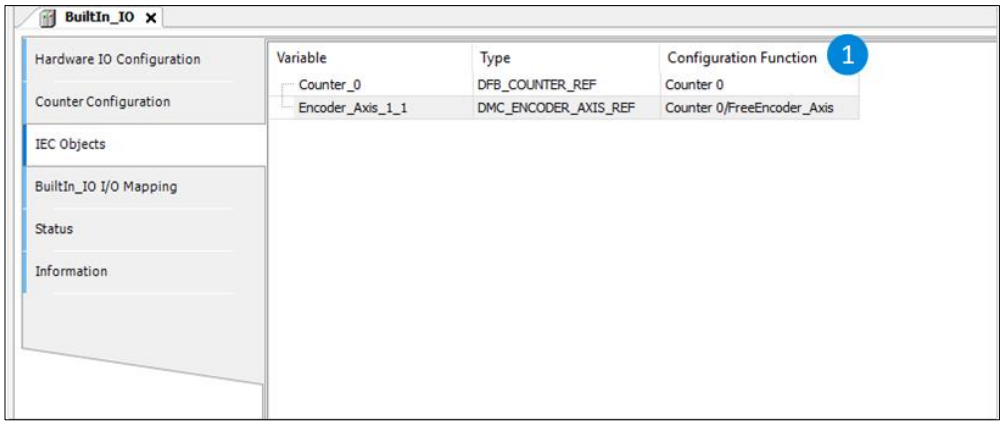
- Counter Used in programs:

  High-speed counter's own encoder axis variable. This variable is used by the user in the POU's motion function block.

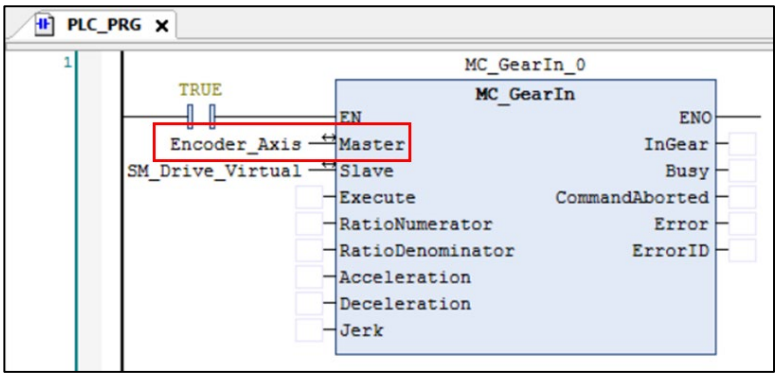
1. In **BuiltIn_IO** screen, select the **IEC Objects** tab.



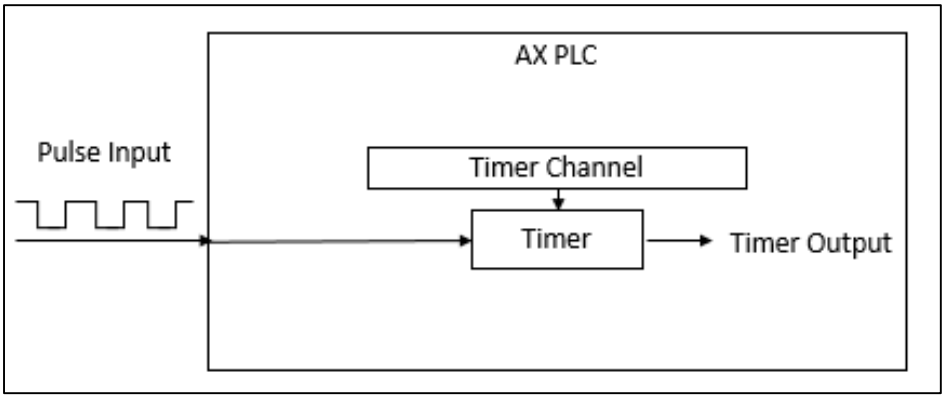
2. In the **IEC Objects** tab,is an example of configuration function, and **Encoder_Axis** must be filled in for the axis name used in POU.



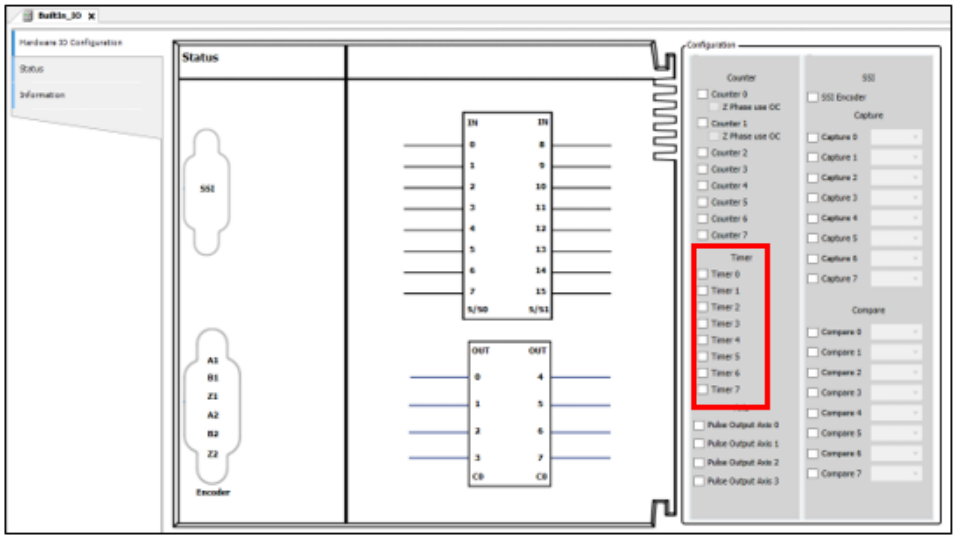
3. In the POU for the **MC_GearIn** function block, the source of the spindle can use Counter as the source, and the entered axis name is Encoder_Axis.



- High-speed timing (Tmr)

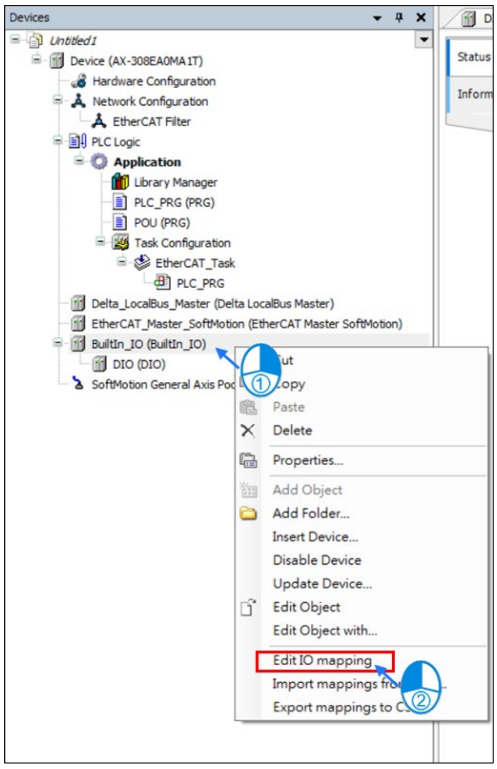


- Quick timer function activated: In the **BuiltIn_IO** screen, there are 8 groups of timers. Click **Timer 0**. Timer does not need to set the screen, so user can start it after selecting it. High-speed timer in AX series internal clock is 0.1 μs.

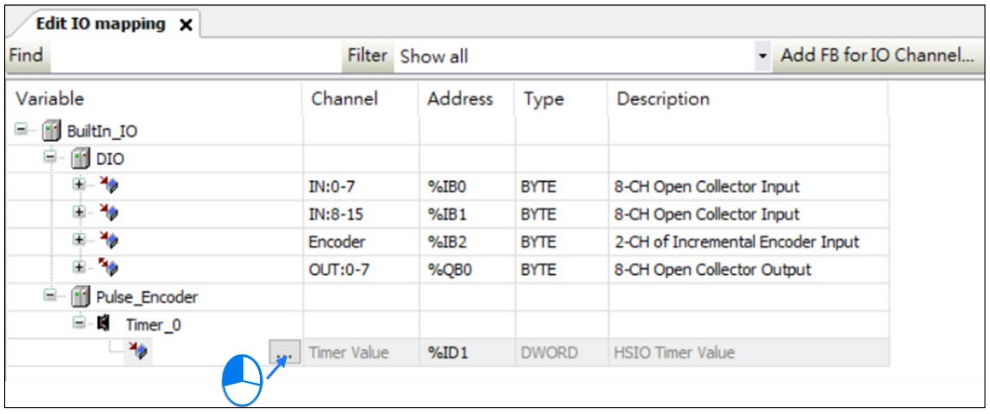


- Set Timer mapping variables

1. Right-click **BuiltIn_IO** and select **Edit IO Mapping**.

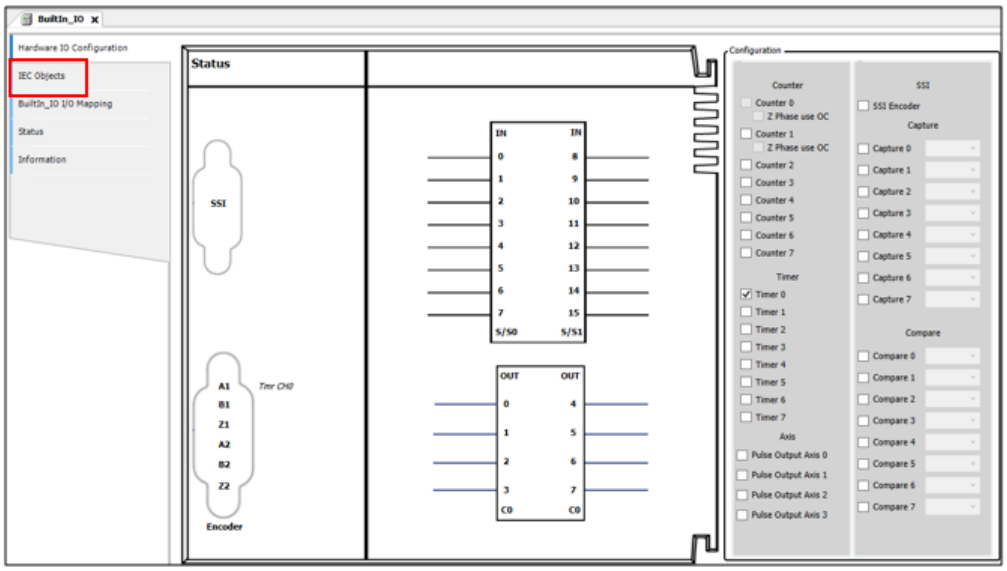


2. In the **Edit IO Mapping** screen, click ... to add a new variable.

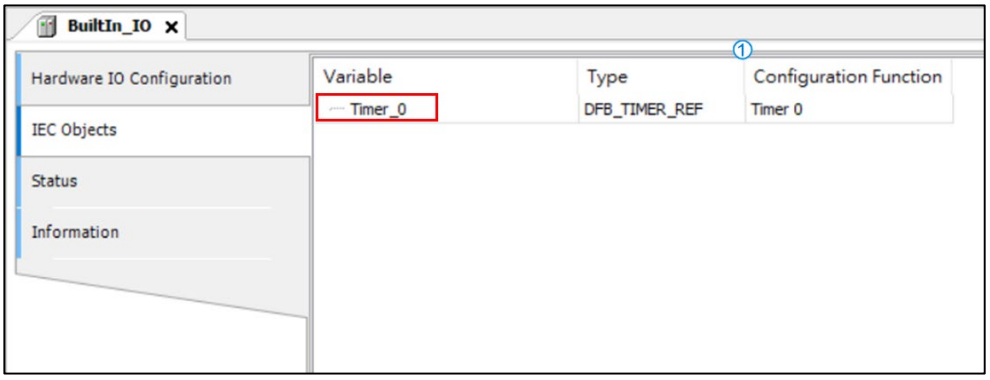


Timer Used in programs: This variable is used by the user's motion function block in the POU.

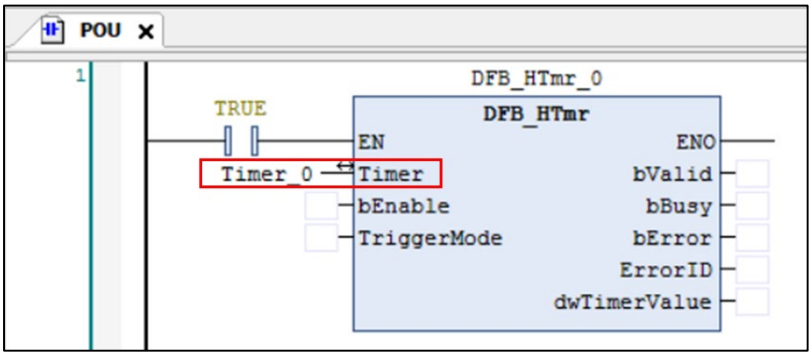
1. In **BuiltIn_IO** screen, select the **IEC Objects** tab.



2. In the **IEC Objects** tab: The following is an example of configuration function, and **Timer_0** is required for the axis name used in the POU. the POU.



3. For the DFB_HTmr function block in the POU, enter the axis name as Timer_0.

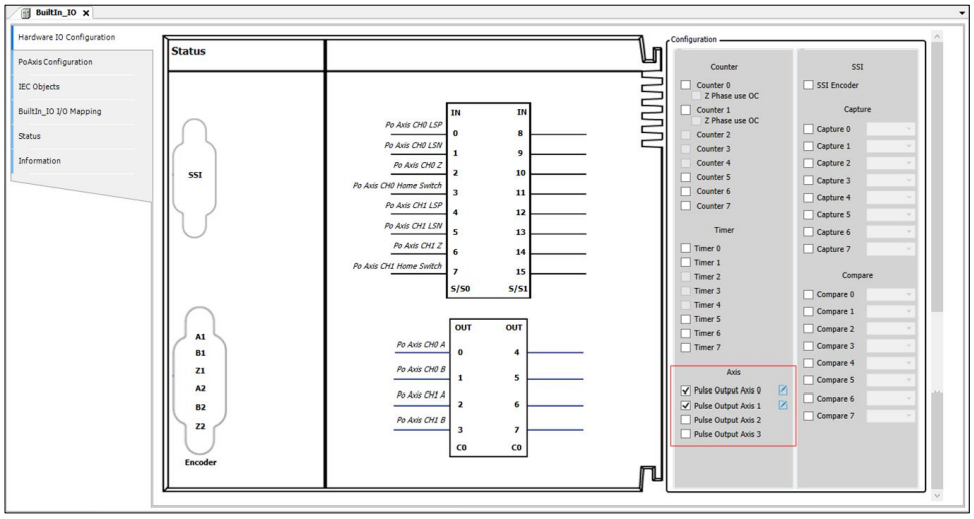


5.2.8.1.4 BuiltIn IO: Po axis configuration

This section describes the IO pulse output function module. It supports up to 4 sets of pulse output devices for Selection.

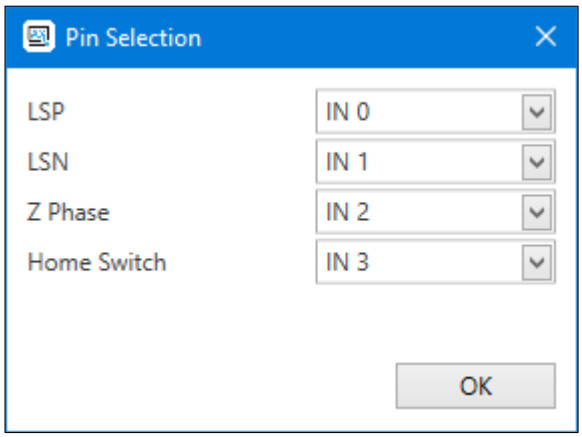
- Axis Function Activated

1. In the **BuiltIn_IO** screen, there are 4 sets of pulse output axes, check one of them to start.



2. After checking, user can click the  icon on the right of the Pulse Output Axis to select the pin.

   Result: The **Pin Selection** window pops up, and user can select the IO terminal number of LSP, LSN, Z Phase, Home Switch from the drop-down menu according to the needs.

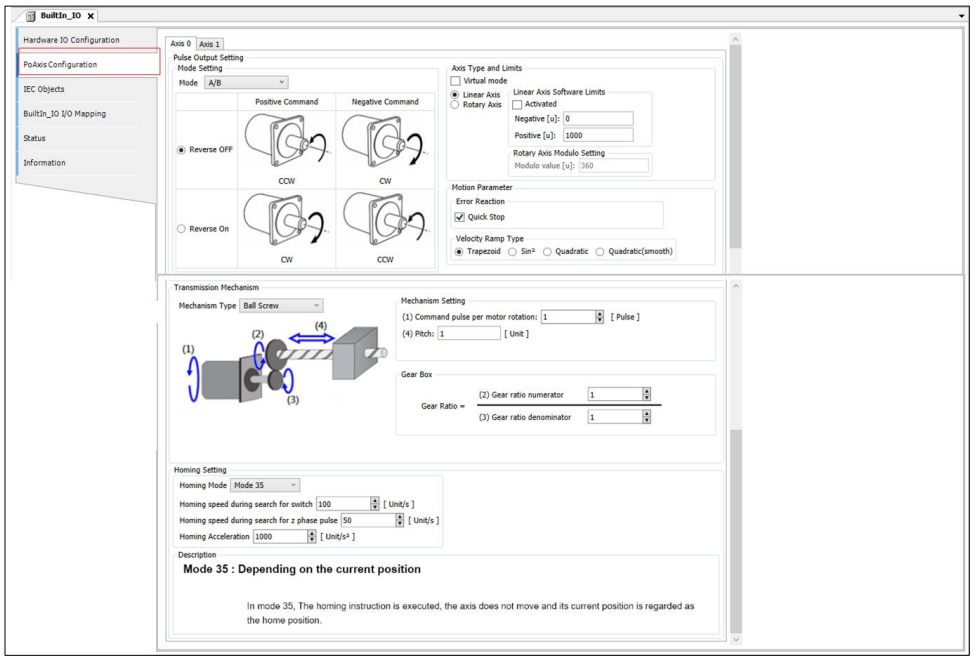


   Note:

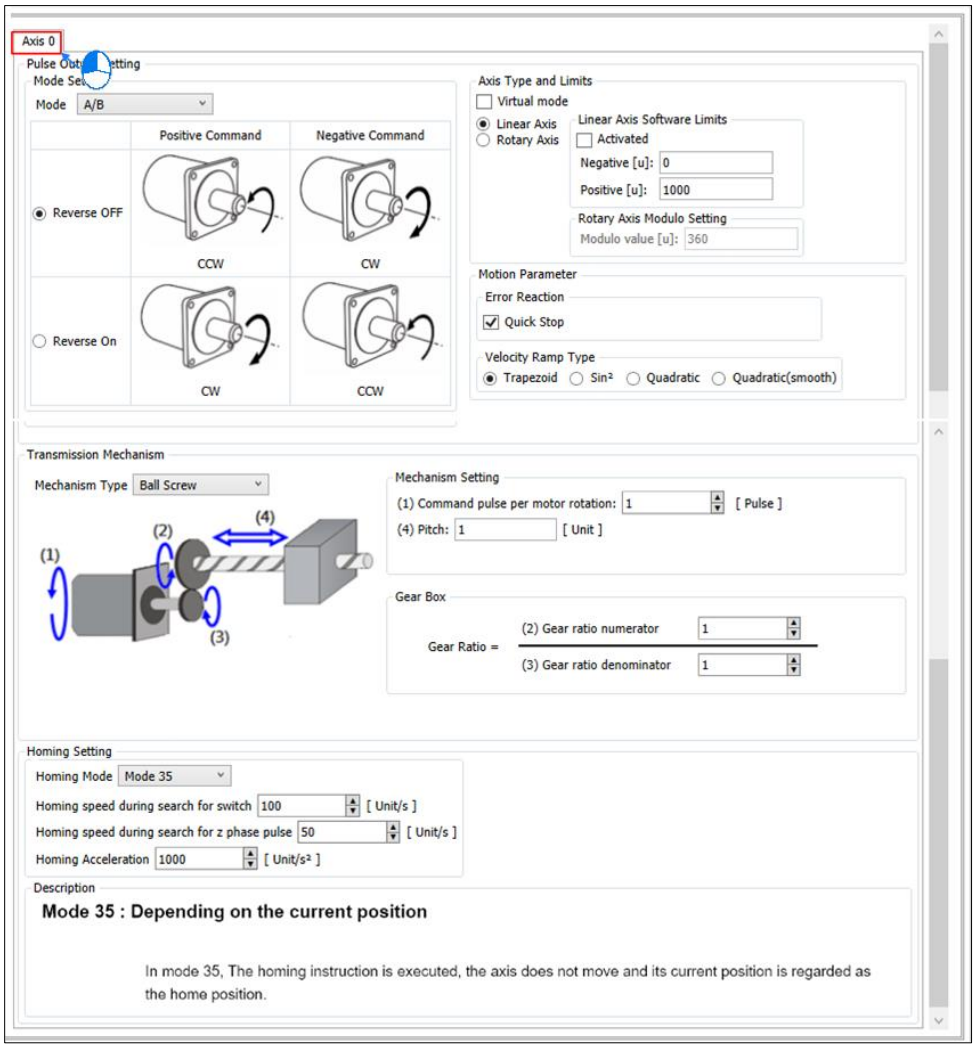
1. In the **Pin Selection** window, if two items select the same IO terminal number, the system will automatically change the previous conflicting number to None.
2. After selecting the pin, the options with pin conflicts in **Counter** and **Timer** will be blacked out and cannot be selected.
3. After selecting the pin, check **Capture** or **Compare**, and the selected IO terminal number will no longer be displayed in the drop-down menu.
4. After selecting the pin, double-click **DIO** in the project tree. The options with pin conflicts in the DIO configuration tab will be blacked out and cannot be selected.
5. If any pin of the current Po axis configuration is already occupied, the pin selection window will pop up automatically when this Po axis is checked, so that **user** can confirm or change the conflicting pin selection.

3. After starting Axis, click **PoAxis Configuration**.

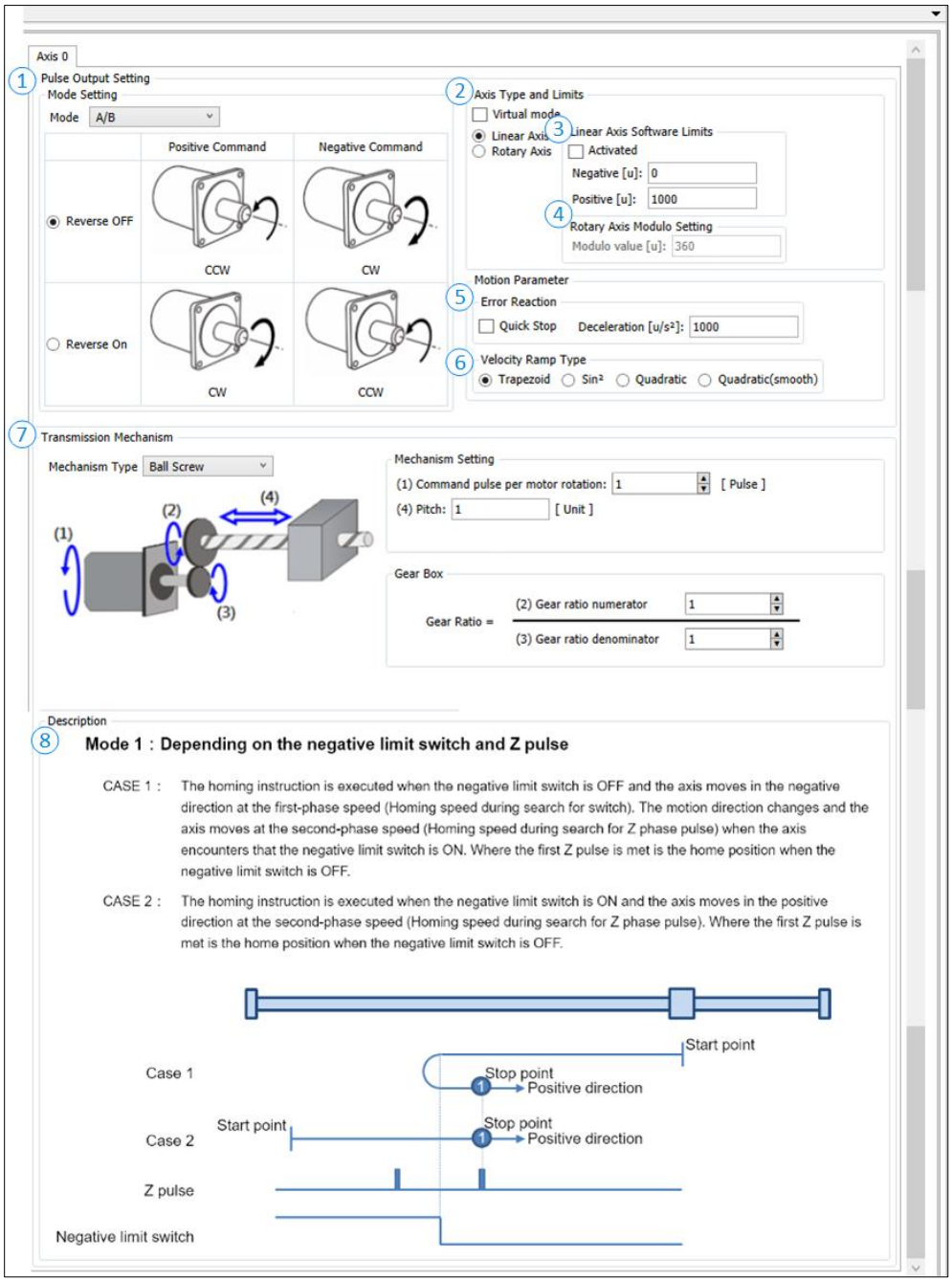
   Result: The window displays as shown in the following figure。



4. Click the **Axis 0** tab after opening the **PoAxis Configuration** window.



5. The related settings of Axis can be set in the **PoAxis Configuration** screen:


Axis 0
1 Pulse Output Setting
Mode Setting
Mode A/B
Positive Command
Negative Command
Reverse OFF
CCW
CW
Reverse On
CW
CCW
2 Axis Type and Limits
Virtual mode
Linear Axis
Rotary Axis
3 Linear Axis Software Limits
Activated
Negative [u]: 0
Positive [u]: 1000
4 Rotary Axis Modulo Setting
Modulo value [u]: 360
Motion Parameter
5 Error Reaction
Quick Stop
Deceleration [u/s²]: 1000
6 Velocity Ramp Type
Trapezoid
Sin²
Quadratic
Quadratic(smooth)
7 Transmission Mechanism
Mechanism Type Ball Screw
(1)
(2)
(3)
(4)
Mechanism Setting
(1) Command pulse per motor rotation: 1 [Pulse]
(4) Pitch: 1 [Unit]
Gear Box
Gear Ratio =
(2) Gear ratio numerator 1
(3) Gear ratio denominator 1
Description
8 Mode 1 : Depending on the negative limit switch and Z pulse
CASE 1 : The homing instruction is executed when the negative limit switch is OFF and the axis moves in the negative direction at the first-phase speed (Homing speed during search for switch). The motion direction changes and the axis moves at the second-phase speed (Homing speed during search for Z phase pulse) when the axis encounters that the negative limit switch is ON. Where the first Z pulse is met is the home position when the negative limit switch is OFF.
CASE 2 : The homing instruction is executed when the negative limit switch is ON and the axis moves in the positive direction at the second-phase speed (Homing speed during search for Z phase pulse). Where the first Z pulse is met is the home position when the negative limit switch is OFF.
Start point
Case 1
Stop point
Positive direction
Start point
Case 2
Stop point
Positive direction
Z pulse
Negative limit switch

① Mode Setting

Name	Features	Set value (default value)
Mode	Set output pulse type	CW/CCW Pulse and Direction (A/B)
Reverse On /Reverse OFF	Set pulse axis forward/reverse	Reverse On Reverse OFF (Reverse OFF)

② Axis Type and Limits

Name	Features	Set value (default value)
Virtual mode	Start virtual axis	TRUE FALSE (FALSE)
Linear Axis/ Rotary Axis	Set linear or rotary axis	Linear Axis Rotary Axis (Linear Axis)

③ Linear Axis Software Limits

Name	Features	Set value (default value)
Activated	Start software limit (only supports linear axis)	TRUE/FALSE (FALSE)
Negative [u]	Reverse Software Limit	(0)

Name	Features	Set value (default value)
Positive [u]	Forward software limit	(1000)

④ Rotary Axis Modulo Setting

Name	Features	Set value (default value)
Modulo Value [u]	Set a circle range (only supports rotation axis)	(360)

⑤ Error Reaction

Name	Features	Set value (default value)
Quick Stop	Emergency stop	(360)
Deceleration [u/s^2]	Axis decelerating to stop (only effective when Quick Stop is not activated)	(10000)

⑥ Velocity Ramp Type

Name	Features	Set value (default value)
Trapezoid/Sin2/Quadratic/ Quadratic(Smooth)	Set axis motion curve	(Trapezoid)

⑦ Software setting screen: refer to SSI Encoder Function Settings

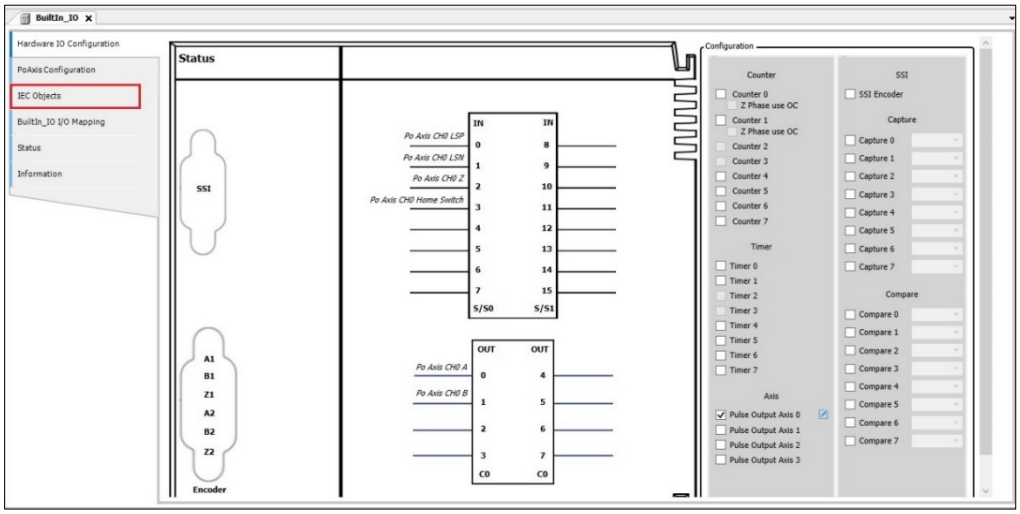
⑧ Homing Setting

Name	Features	Set value (default value)
Homing Mode	Set origin mode	(Mode 35)
Homing speed during search for switch	Set the homing switch speed	(100)
Homing speed during search for z phase pulse	Set Z-Seek Speed	(50)
Homing Acceleration	Origin speed	(1000)

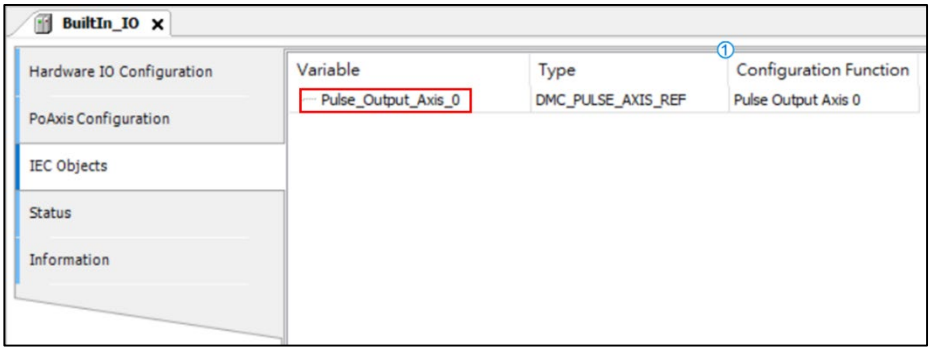
- Pulse Axis Used in programs:

  Pulse Axis is used in the POU. User need to use this variable in the motion function block in the POU.

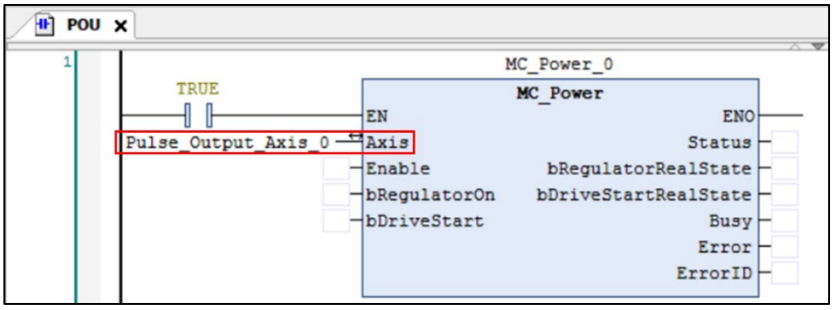
1. Select **IEC Objects** in the **BuiltIn_IO** screen.



2. In the **IEC Objects** screen, in the following IEC object screen is an example of configuration function. For the axis name used in the POU，**Pulse_Output_Axis_0** should be filled in.



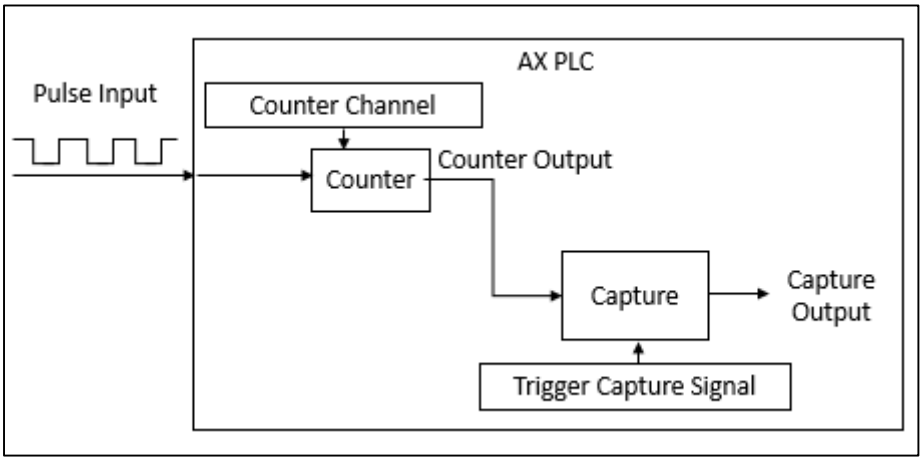
3. In the POU for the **MC_Power** function block, the input axis name is Pulse_Output_Axis_0.



5.2.8.1.5 BuiltIn IO: IEC object

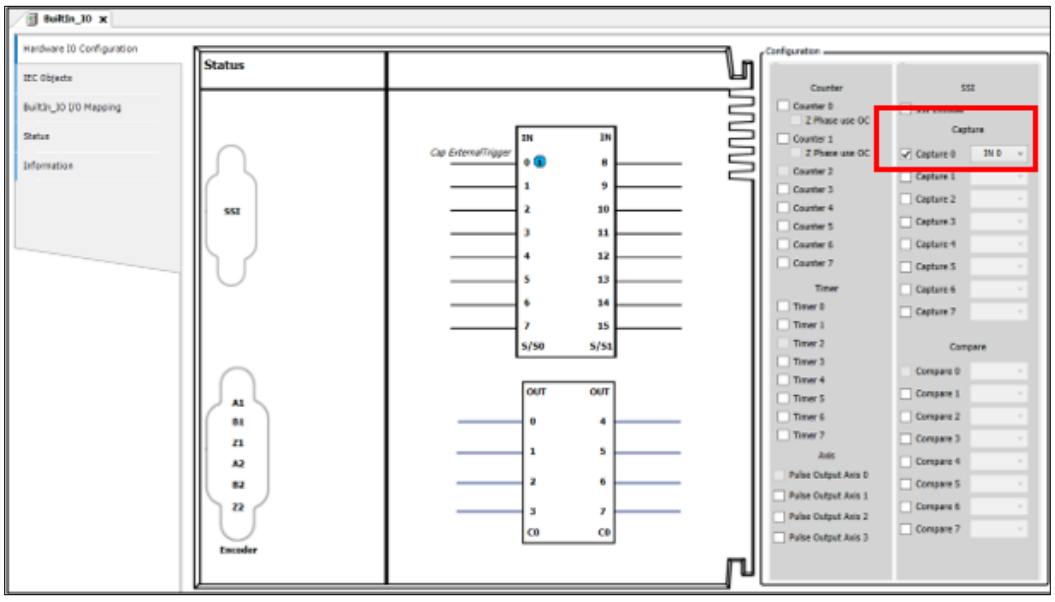
This section describes the high-speed counter comparator and capture function modules of the main body IO. The AX-3 series supports up to 8 sets of high-speed capturer and high-speed counter comparators.

- High-Speed Counter Capture

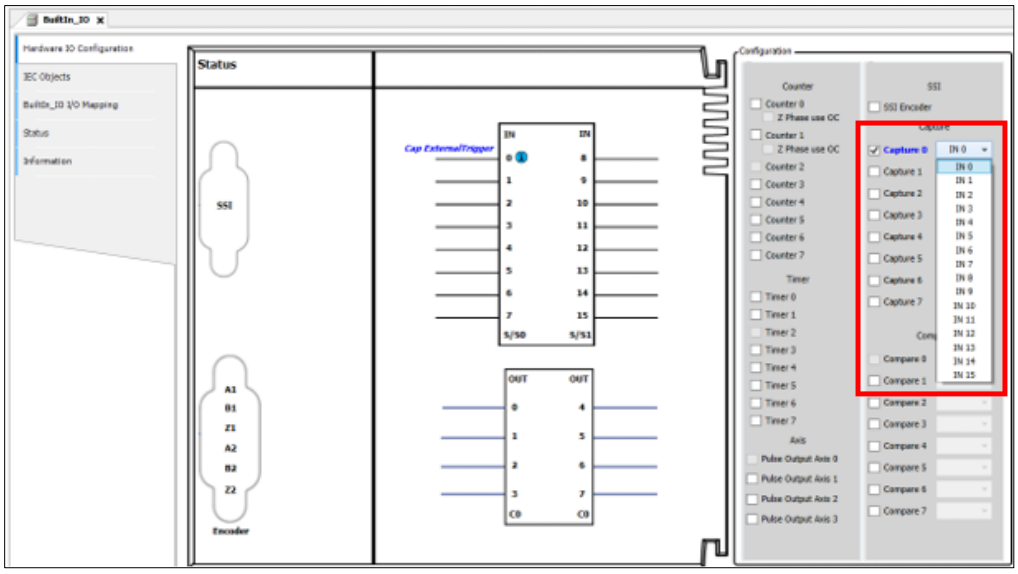


- Capture function is activated.

1. In the **BuiltIn_IO** screen, **Capture** has a total of 8 points, click one of them to start.

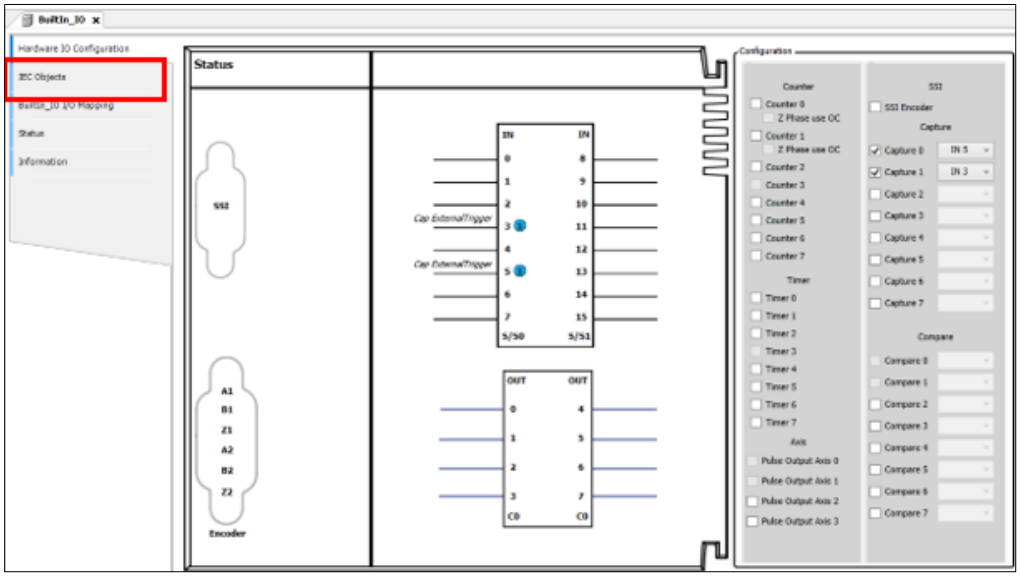


2. After launching **Capture**, click the drop-down menu to select the external trigger source.

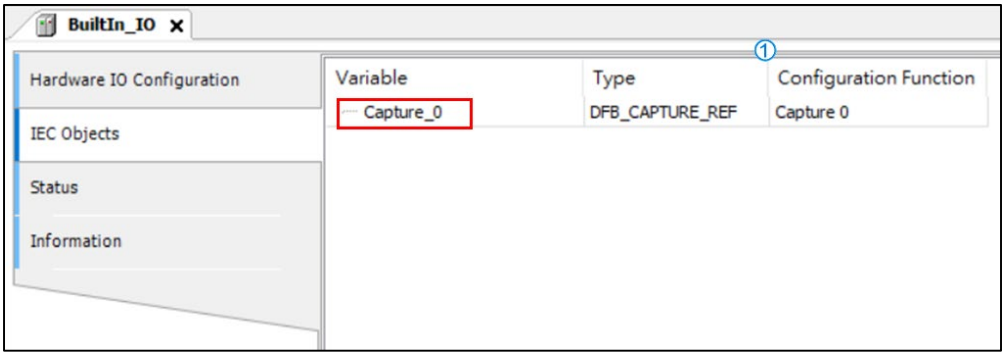


- **Capture** is used in the program: This variable is used by the user's motion function block in the POU.

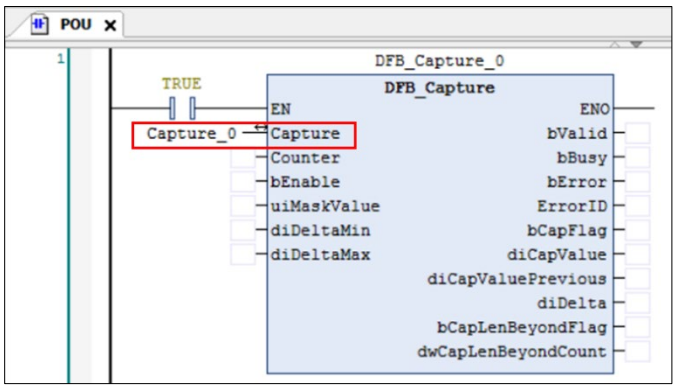
1. Select **IEC Objects** in the **BuiltIn_IO** screen.



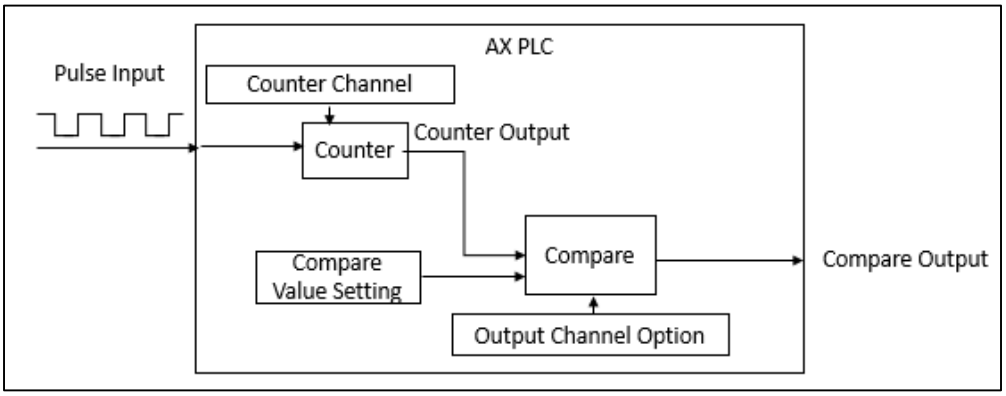
2. In the following **IEC Objects** screen, is an example of configuration function introduction, Capture_0 is required for the axis name used in POU.



3. For the DFB_Capture function block in the POU, the axis name entered is Capture_0.

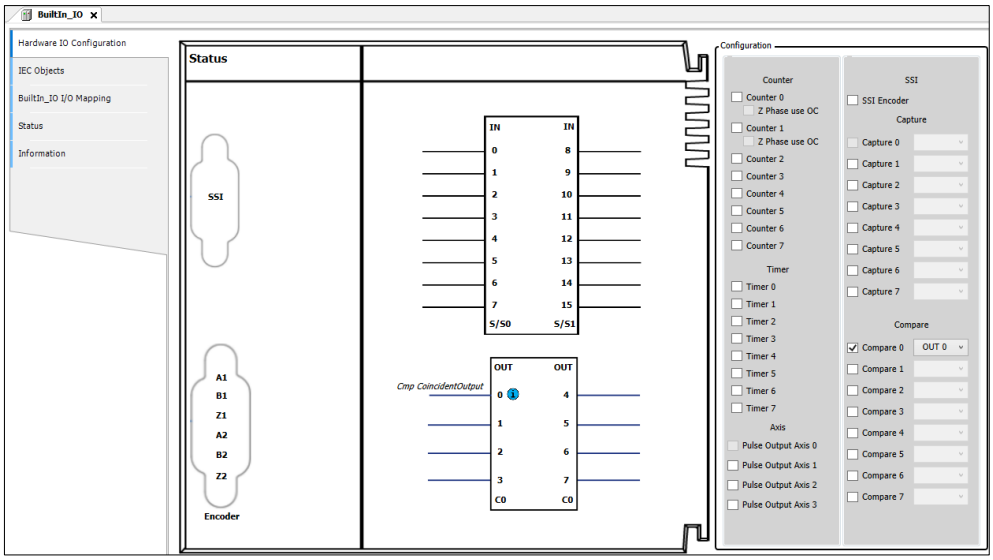


- High-Speed Counting Comparator (Compare)

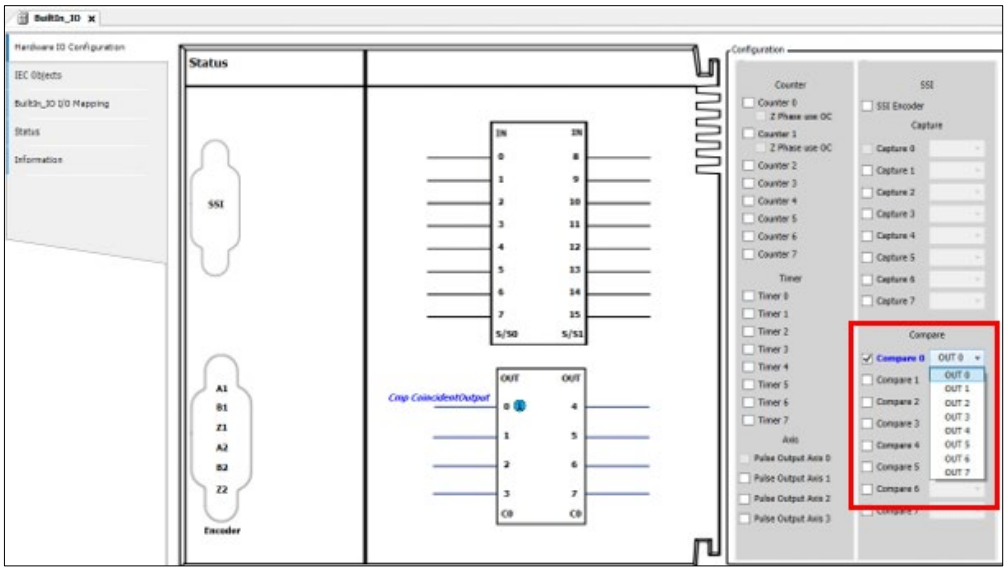


- Compare Function start

1. In the **BuiltIn_IO** screen, **Compare** has a total of 8 points, click one of them to start.



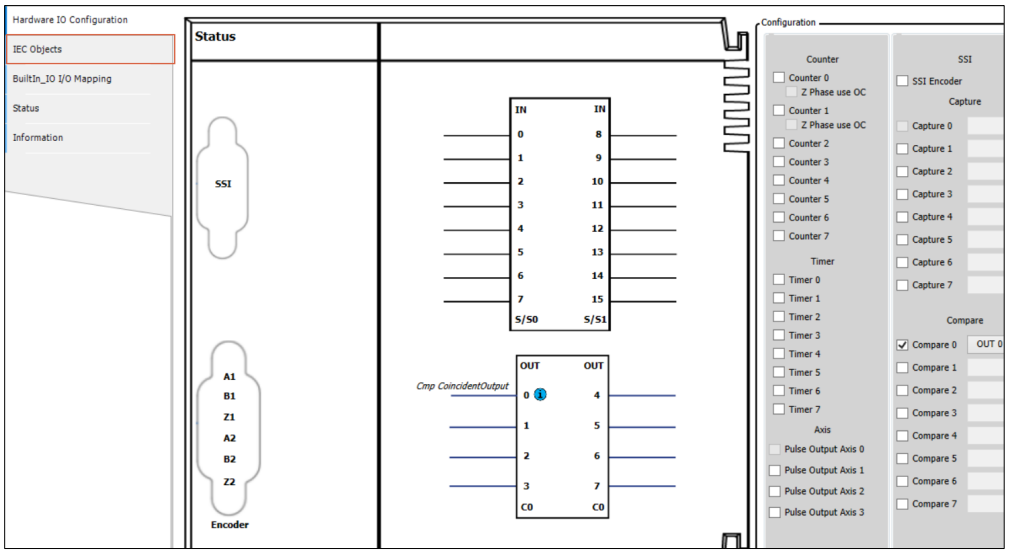
2. After clicking **Compare**, click the drop-down menu, select **Compare** to output the external contacts successfully.



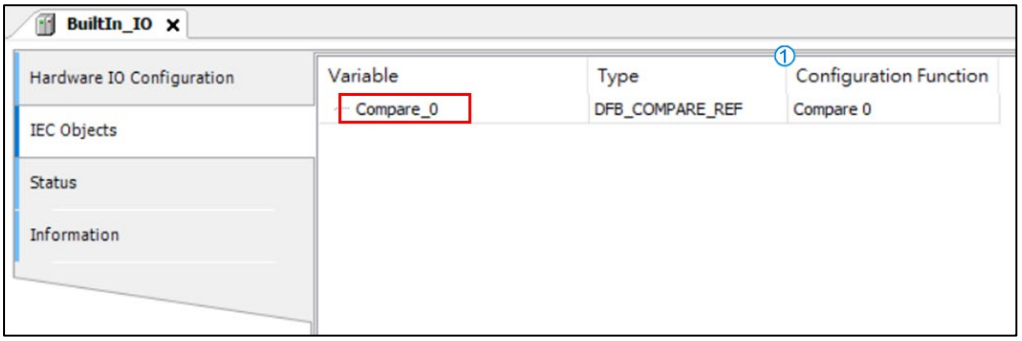
- **Compare** is used in the program :

  This variable is used by the user's motion function block in the POU.

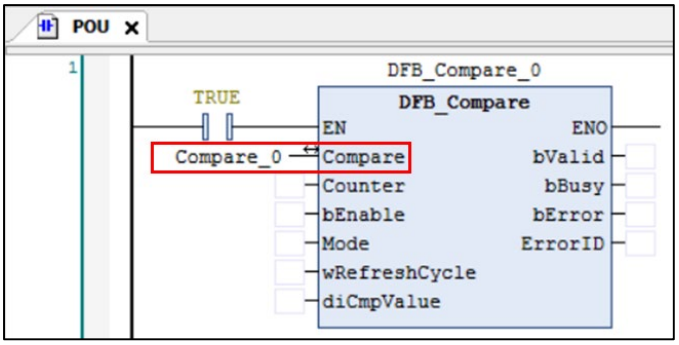
1. Select **IEC Objects** in the **BuiltIn_IO** screen.



2. In the following **IEC Objects** screen, is an example of configuration function introduction, for the axis name used in the POU, user need to fill in Compare_0.



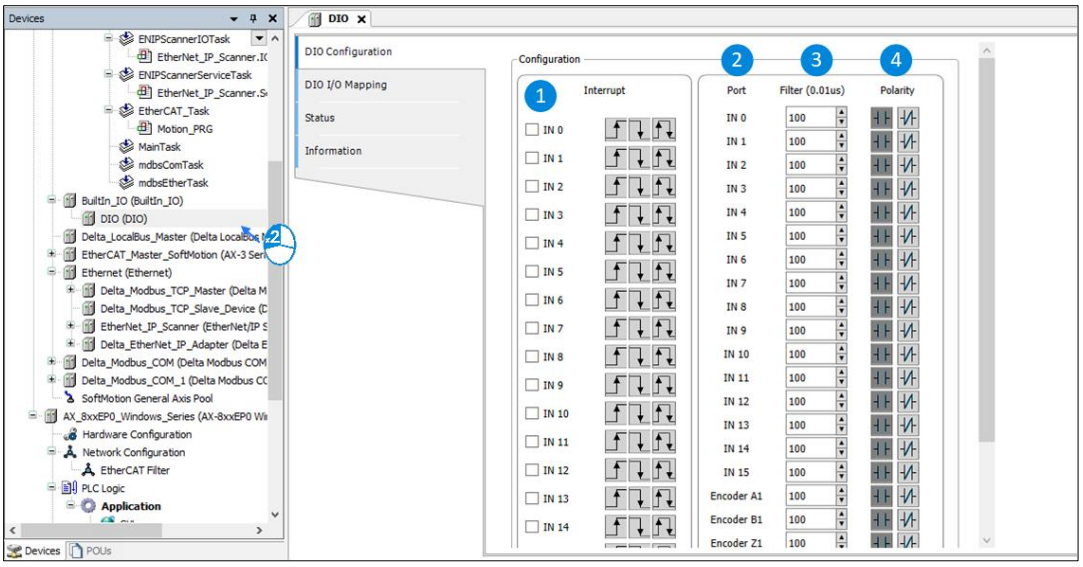
3. For the DFB_Compare function block in the POU, enter the axis name as Compare_0.



5.2.8.1.6 DIO: DIO configuration

This section describes the DIO device, which sets the I/O points of the host to interrupt, filter, and polarity.

Double-click **DIO** to enter the setting screen.



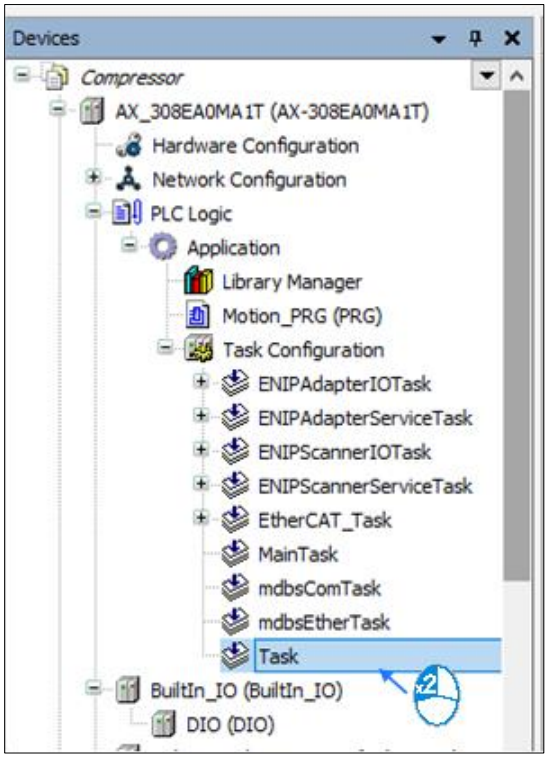
- Configuration

Features	Description
① Interrupt	default value
	Enable external interrupt
	When external interrupt is enabled, set the input contact to upper differential signal.
	When external interrupt is enabled, set the input contact to the lower differential signal.
	When external interrupt is enabled, set the input contacts to the upper and lower differential signals.
② Port	Corresponding external contact number
③ Filter	100 Set filtering time (0.01us), the setting range is (0 ~ 30000000), the default is 100x0.01us.

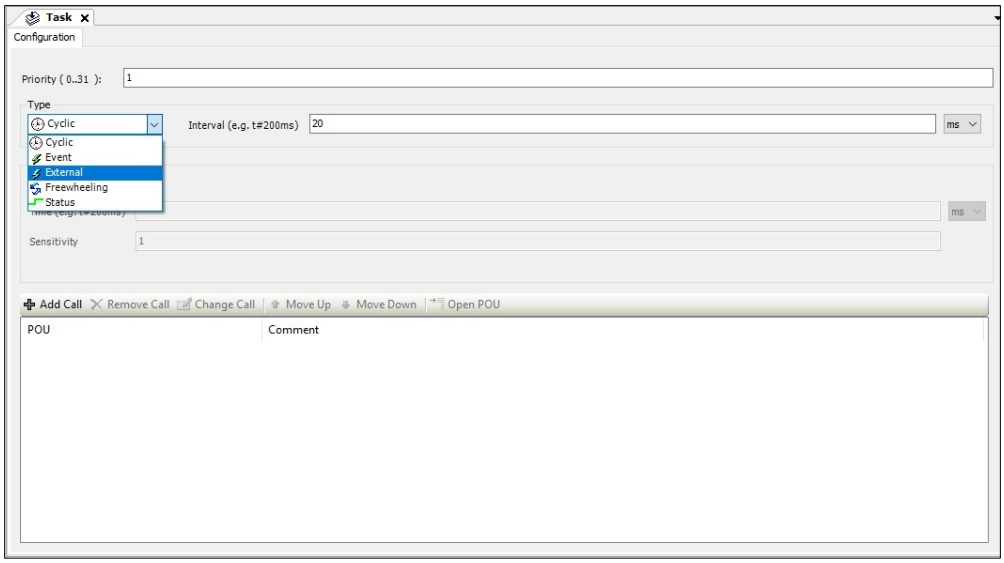
Features	Description
④ Polarity	Set the input contact polarity. The default value is A contact.
	Set the input contact polarity to B contact

- I/O Interrupt Usage:

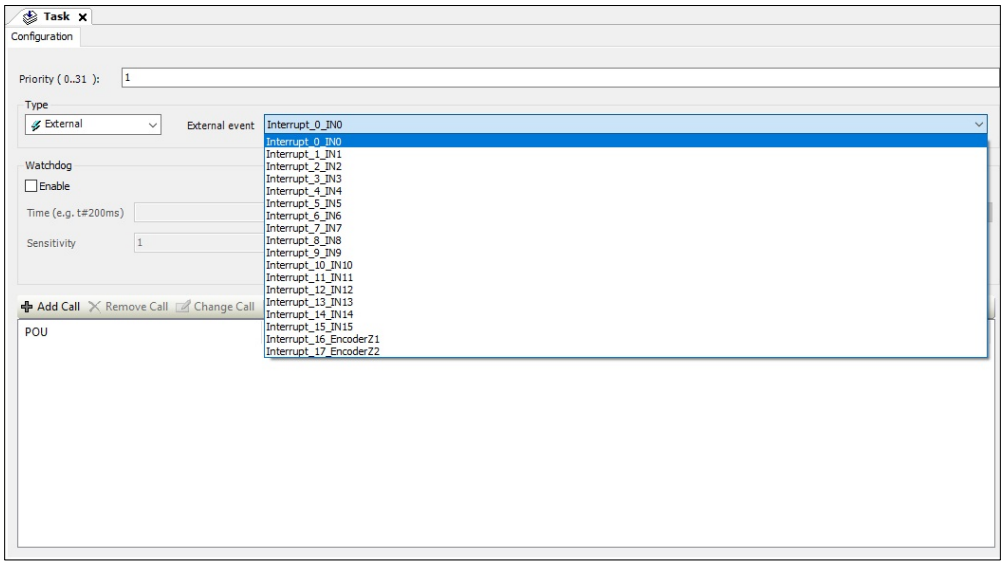
  After enabling the interrupt function on the DIO setting screen, click on the **Task** that needs the interrupt function.



1. Open **Task** setting screen, click **Type** and select **External**.

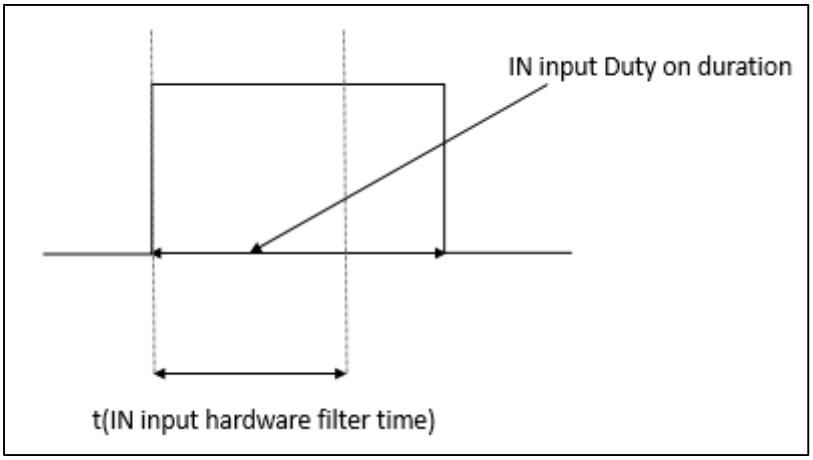


2. After selecting **External**, click **External event** to select the corresponding interrupt contact.



- The setting time of the hardware filtering time is less than the time of inputting Duty On at the IN point, as shown in the following figure:

1. Hardware filter input range is 1 ~ 50000000, the unit is 0.1us.



2. The relationship between filtering frequency and time is as follows:

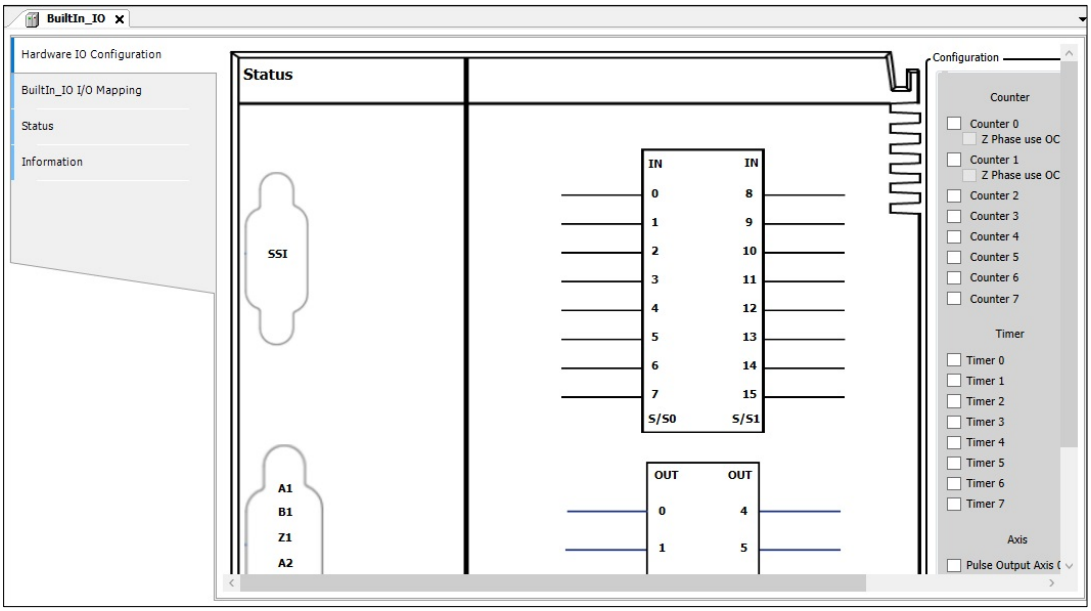
   Filtering frequency * 2 (Hz): Filtering frequency = 1/(2*t); t is the setting value of filtering time (unit 0.01 μs).

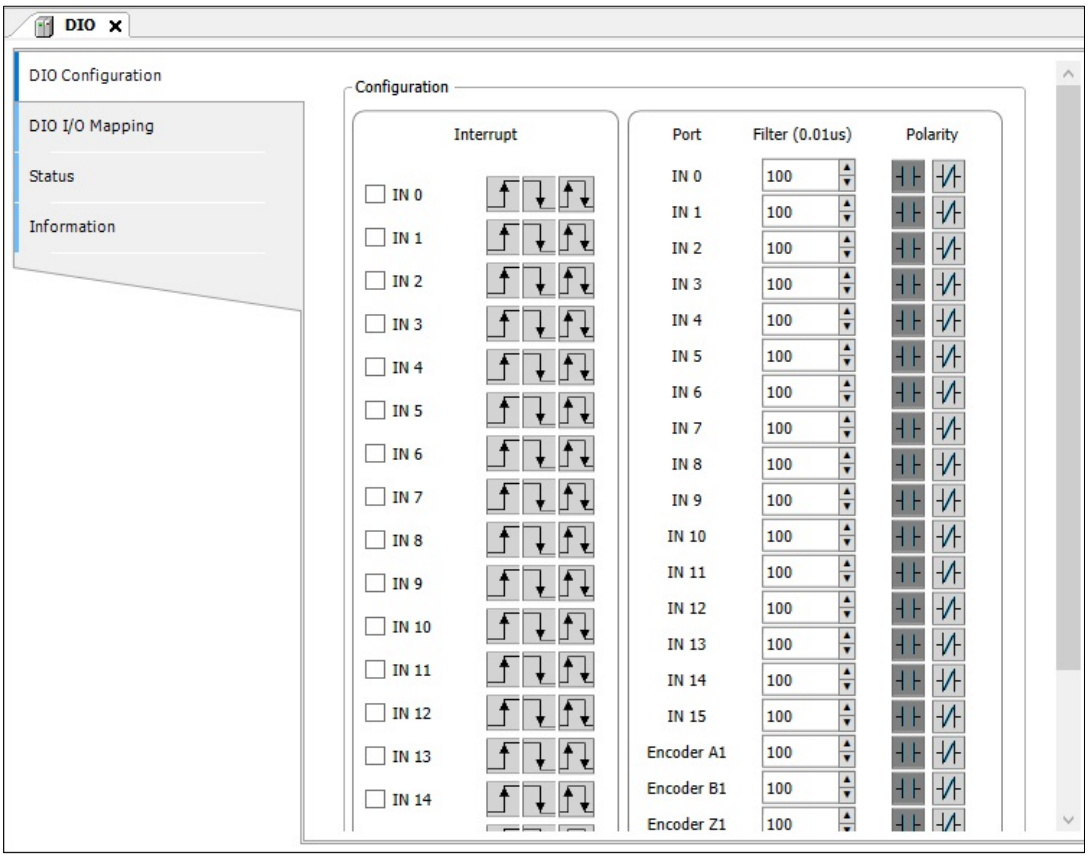
   When the input frequency is higher than the filtering frequency range, the signal will be filtered.

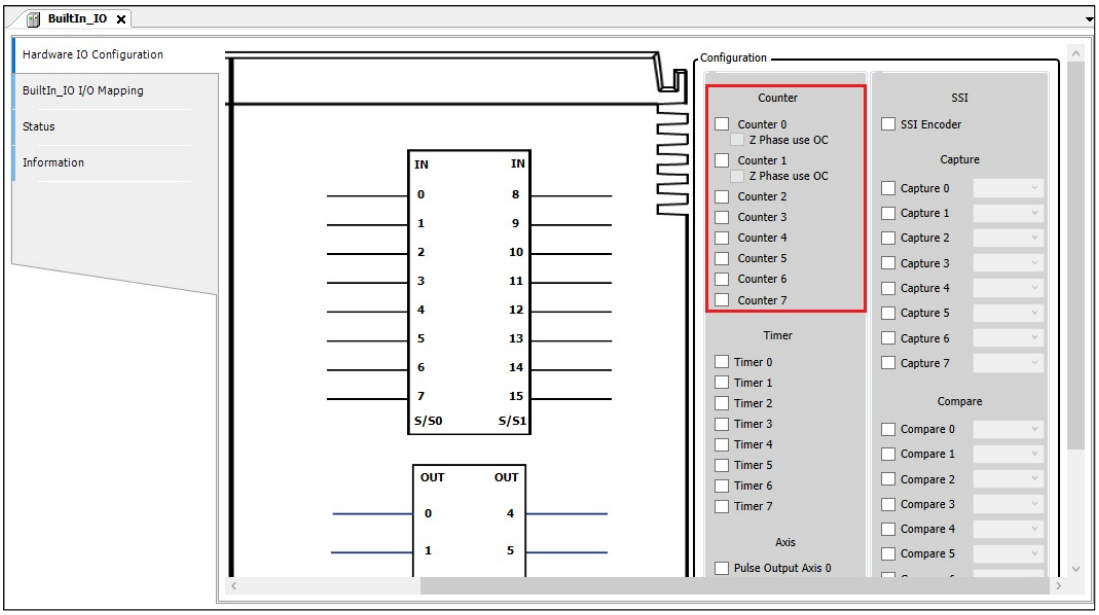
   This function is for the IN input points such as DFB_Capture, DFB_Hcnt, DFB_HTmr, DFB_Compare, and the body IO interrupt.

5.2.8.2 High-speed I/O setting (AX-324/AX-304)

5.2.8.2.1 BuiltIn I/O: Hardware IO configuration







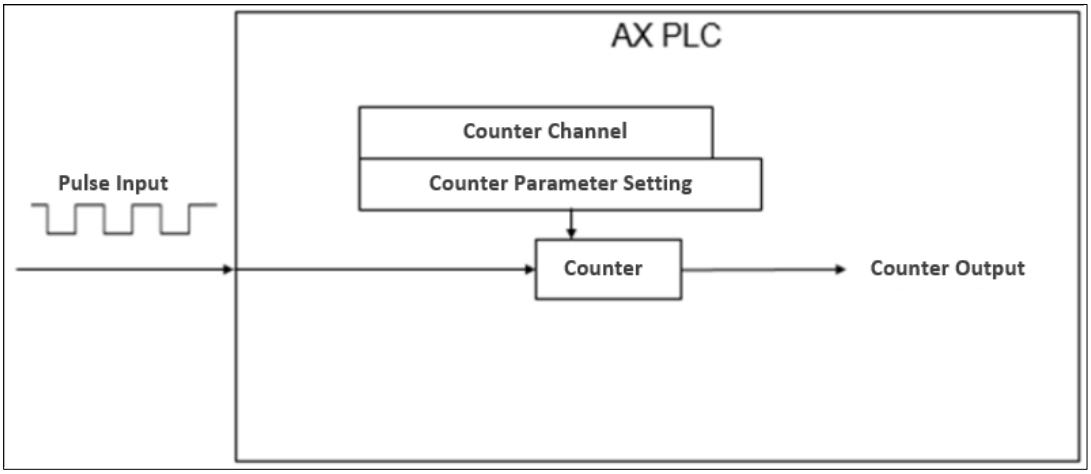
On the host, high-speed I/O provides the following function settings:

- DIO: interrupt, filter, polarity and other functions can be set.
- Counter: User can set the variable and counting mode of the high-speed counter and whether the Z-phase signal is enabled or not, or the high-speed timer variable declaration.
- Capturer/Comparator: User can declare the variables of the high-speed capture or high-speed comparator.

5.2.8.2.2 BuiltIn IO: Counter/timer configuration

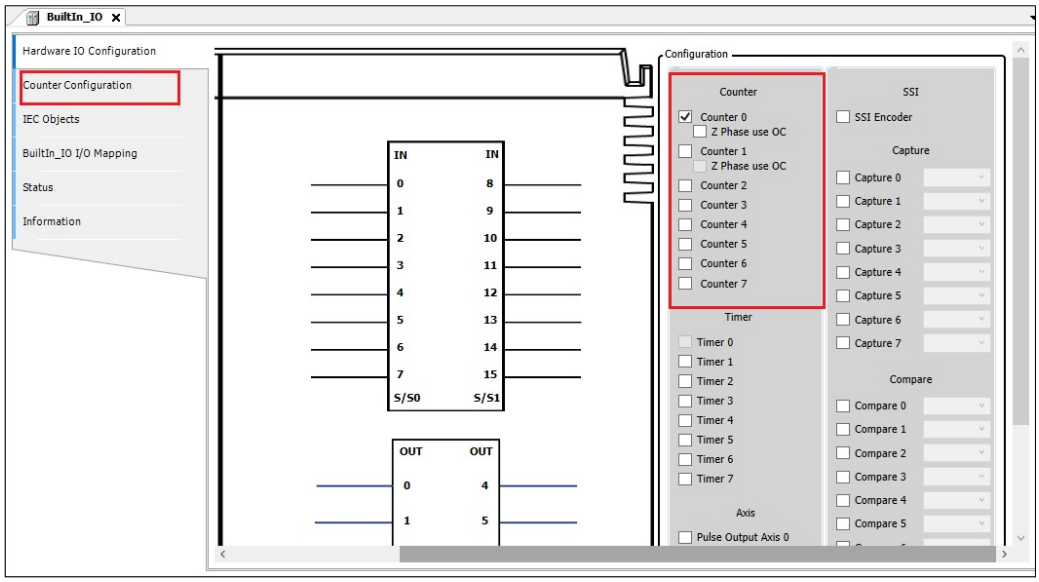
Here will describe the pulse encoder function module of the main body IO. The AX-324/AX-304 supports up to 6 groups of high-speed counters and high-speed timers.

High-speed counter (Cnt)

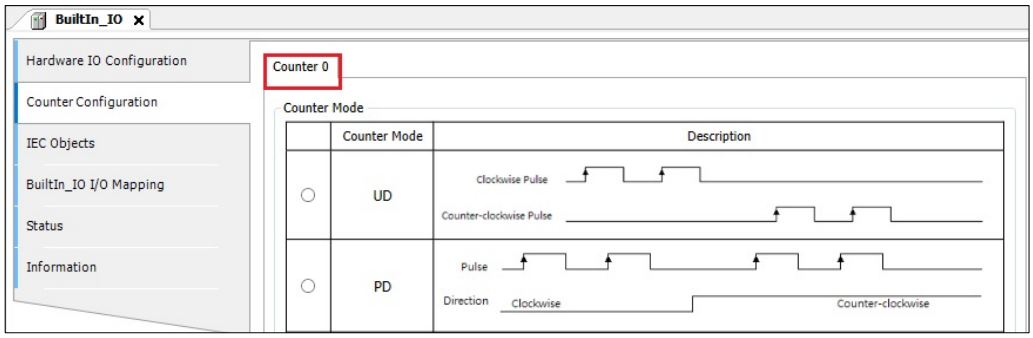


High-speed counting function activated

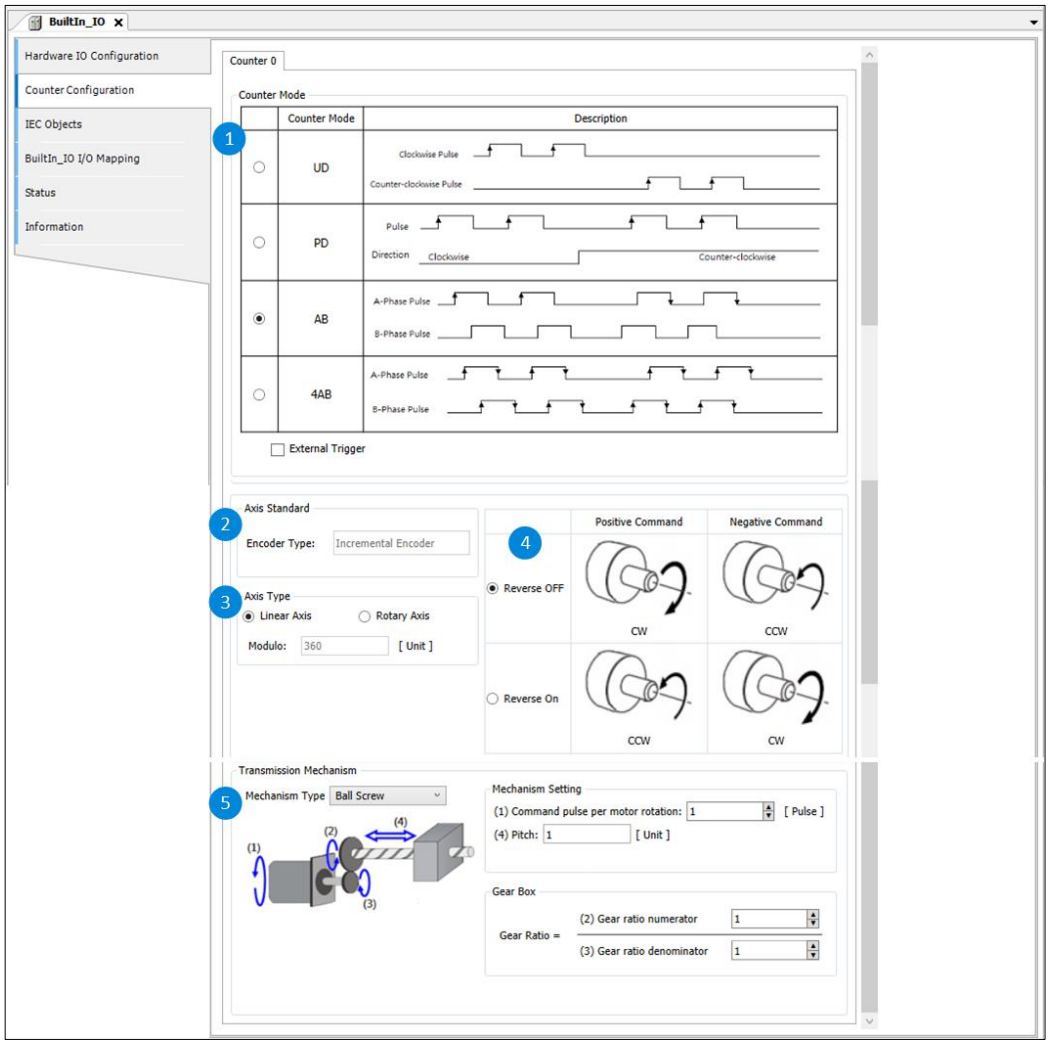
1. In the BuiltIn_IO screen, there are 6 groups of counters, check counter 0 and then click counter configuration.



2. See the counter 0 page in the counter configuration screen.



3. The relevant settings of the counter can be set in the counter configuration screen. The following is a description of the setting:



① Counting Mode

Name	Features
UD	Forward pulse train and reverse pulse train
PD	Pulse train + direction

Name	Features
AB	AB phase pulse train
4AB	AB phase pulse train (4x)
External Trigger	Z phase signal enabled

② Shaft standard

Name	Features	Set value (default value)
Encoder type	Encoder type display	(Incremental encoder)

③ Shaft Type

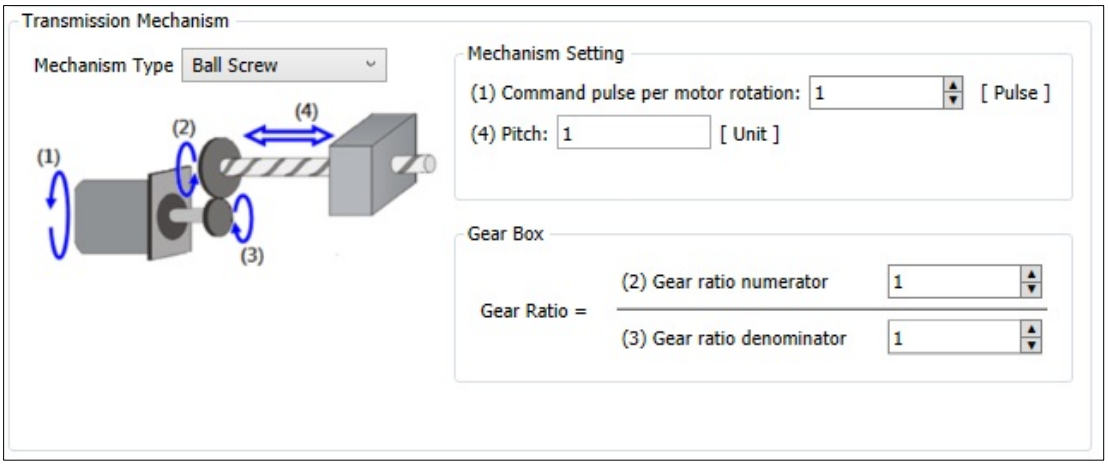
Name	Features	Set value(default value)
Linear axis/rotary axis	Set linear axis or rotary axis	Linear axis Axis of rotation (Linear axis)
Modulus value	When the rotation axis is selected, the rotation axis can be set to a larger value	(360)

④ Positive/Negative Instructions

Name	Features
Reverse Close/Reverse Open	Set forward/reverse command

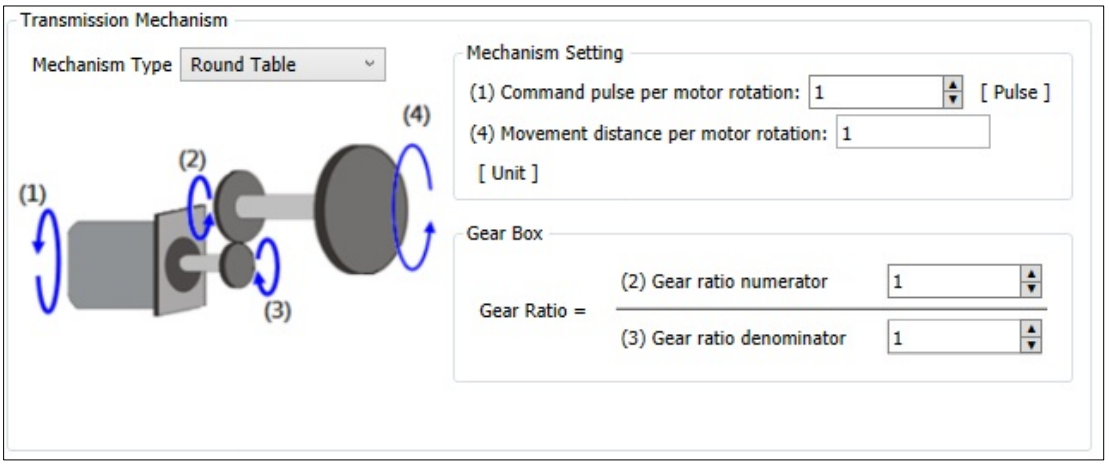
⑤ Transmission mechanism: The following is introduced for different architectures

Ball screw drive



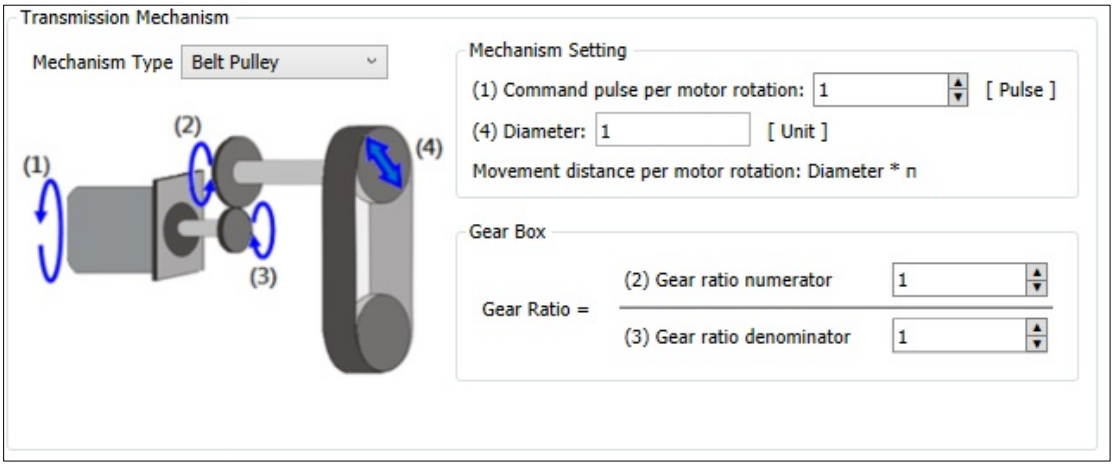
Name	Features
(1) Command pulse per motor revolution	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Pitch	Screw Pitch

Round Table Drive



Name	Features
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) The moving distance of the motor per rotation	Movement distance per motor rotation

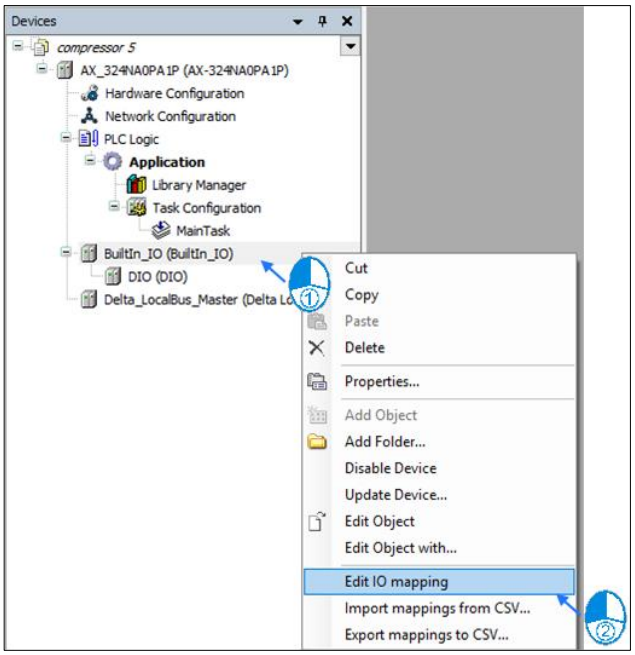
Belt Drive



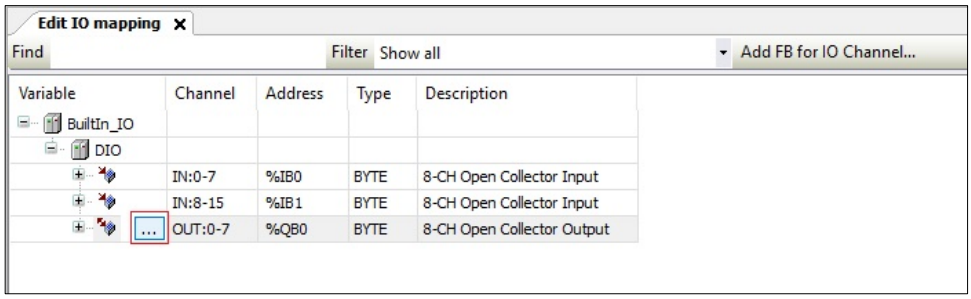
Name	Features
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Diameter (moving distance per revolution of the motor: diameter*n	Diameter (moving distance per motor revolution: diameter*n)

Set counter mapping variable

1. Right click **BuiltIn_IO** > **Edit IO mapping**.

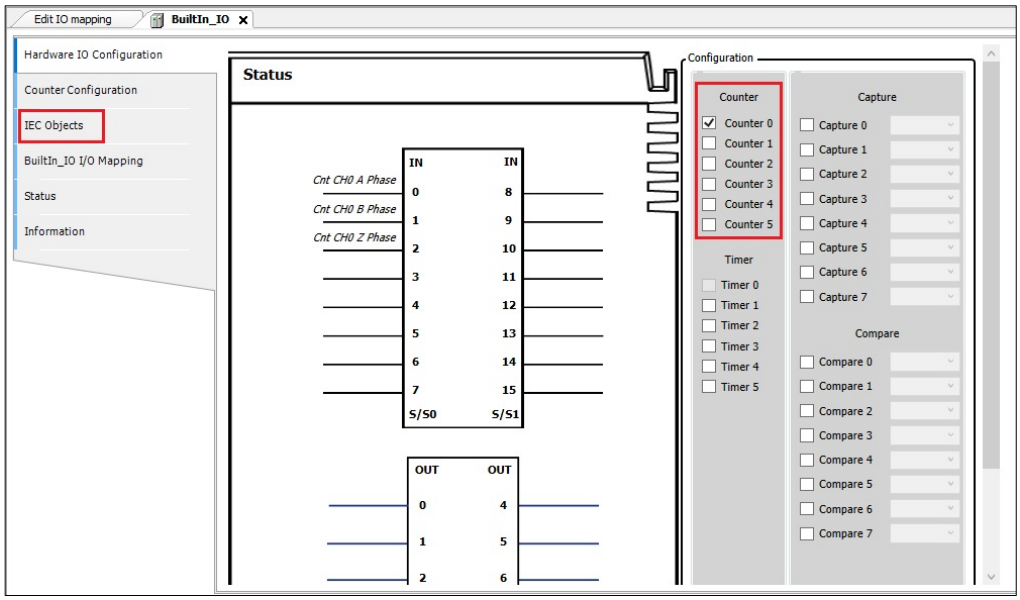


2. In the edit IO mapping screen, click to **add variables**.

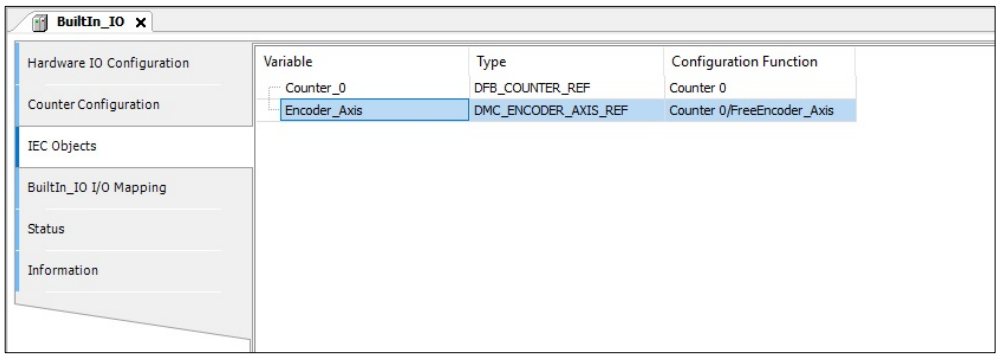


The counter is used in the program: the encoder axis variable of the high-speed counter, this variable is used by the user in the motion function block in the POU.

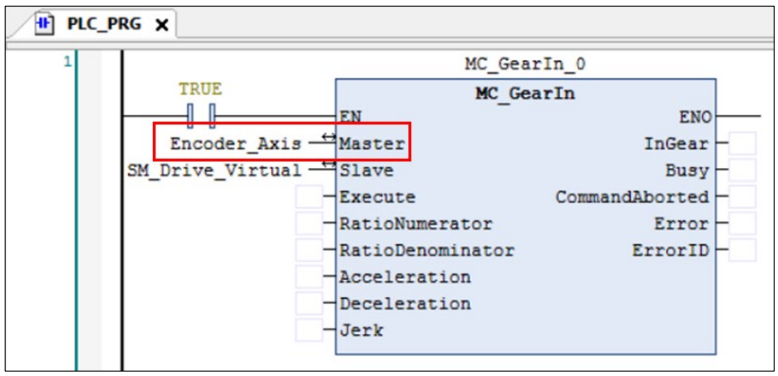
1. Click the **IEC object** on the **BuiltIn_IO screen**.



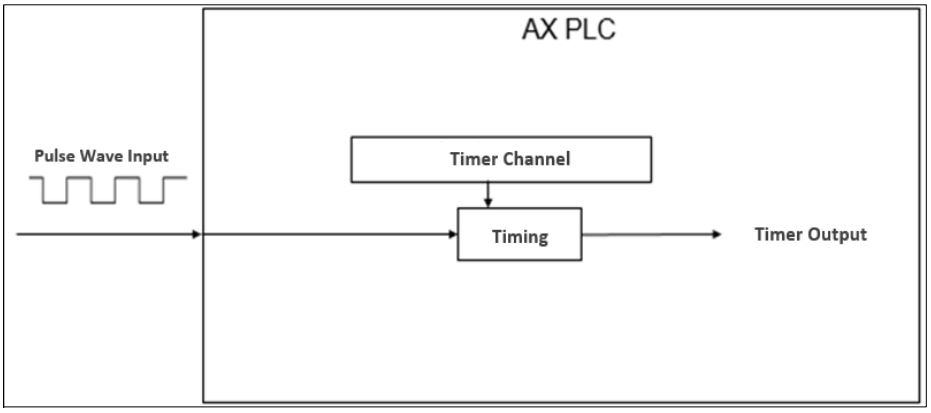
2. The following is an example of configuration function. Encoder_Axis is required for the axis name used in POU.



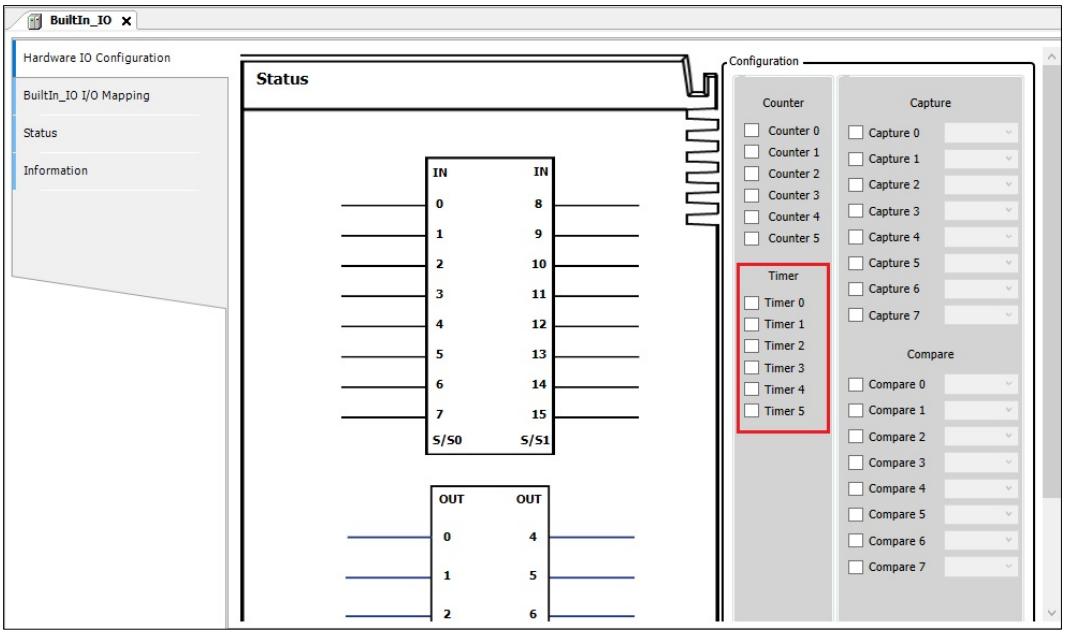
3. For the MC_GearIn function block in the POU, the spindle source can use the counter as the source, and the input axis name is Encoder_Axis.



High-speed timing (Tmr)

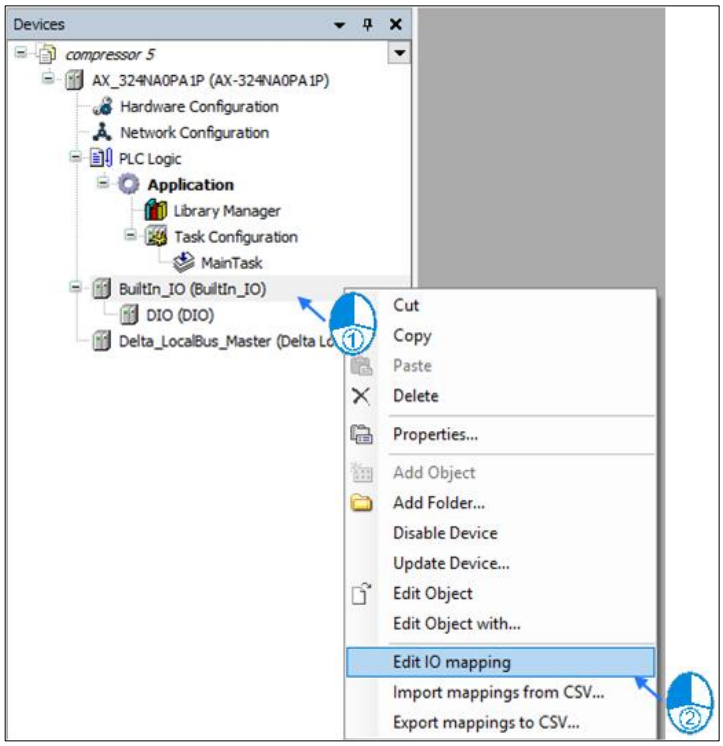


The high-speed timing function is activated: In the BuiltIn_IO screen, there are 6 groups of timers. Click **Timer 0**. Since the timer does not need to set the screen, it can be started after checking. The internal timing clock of the high-speed timer in the AX series is 0.1 μs.

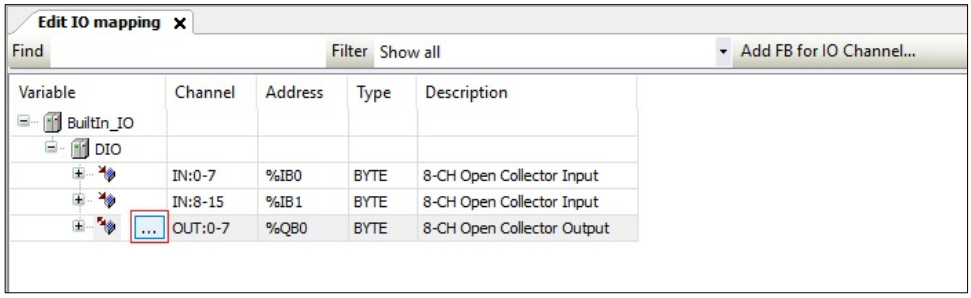


Set timer mapping variable

1. Right-click **BuiltIn_IO** and select **Edit IO mapping**.

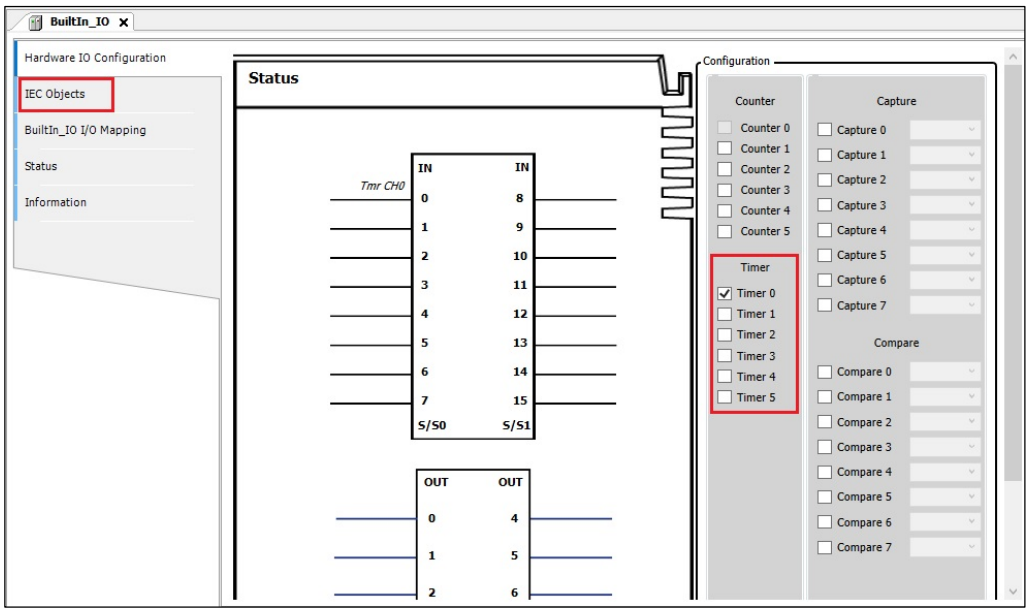


2. In the edit **IO mapping** screen, click on **add variables**.

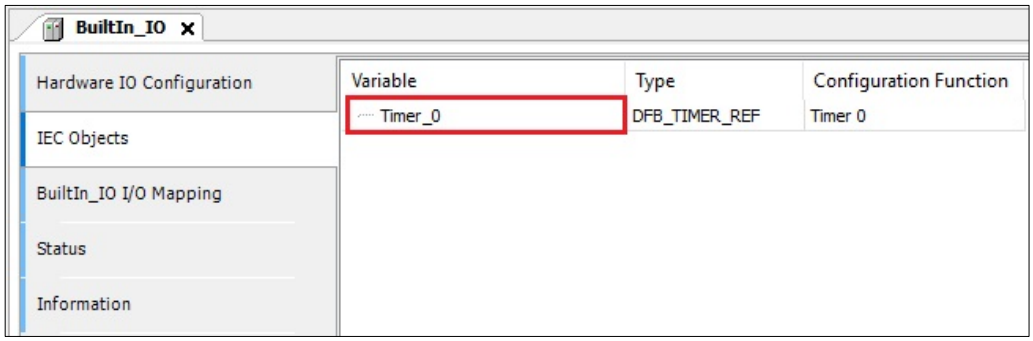


The timer is used in the program: this variable is used by the user in the motion function block in the POU.

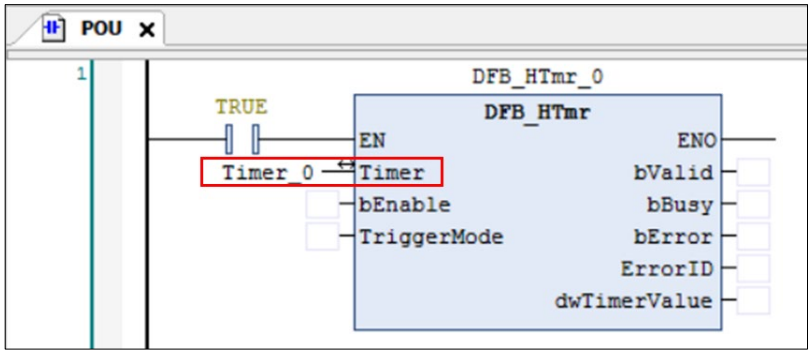
1. Click the **IEC objects** on the **BuiltIn_IO** screen.



2. The following is an example of the configuration function. For the axis name used in the POU, Timer_0 is required.



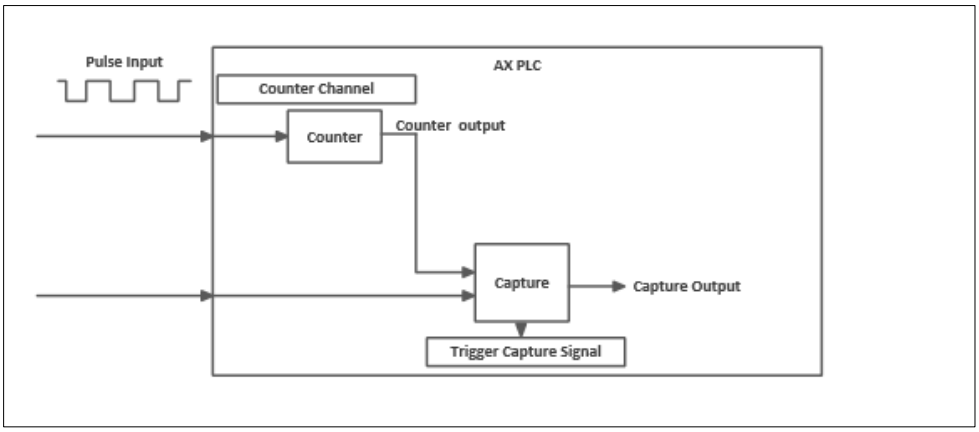
3. For the DFB_HTmr function block in the POU, the input axis name is Timer_0.



5.2.8.2.3 BuiltIn IO: IEC object

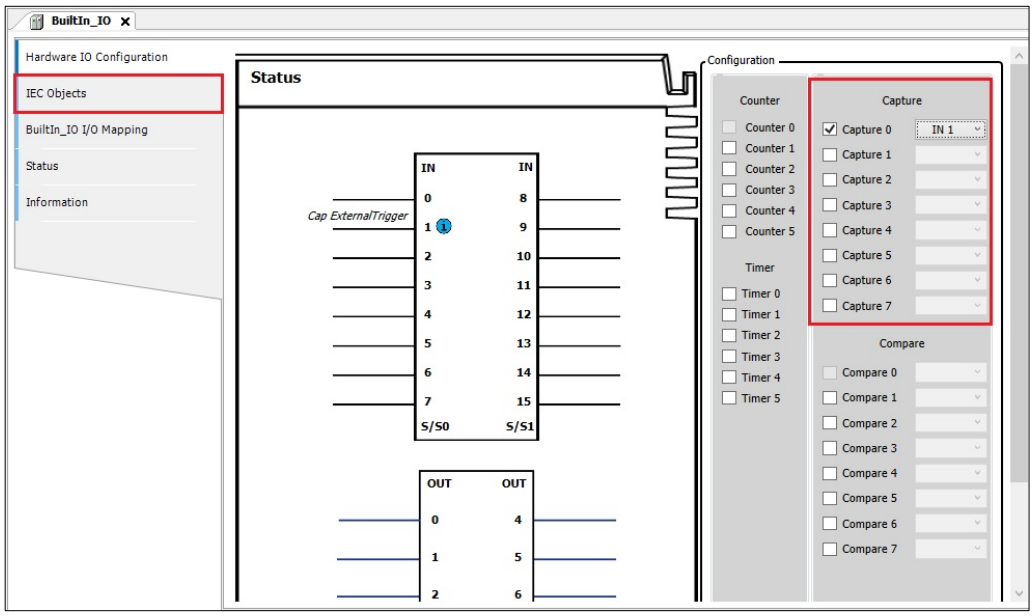
Here is a description of the high-speed counter comparator and catcher function modules of the main body IO. The AX-324/AX-304 supports up to 8 groups of high-speed catcher and high-speed counter comparator.

High-speed counter capture

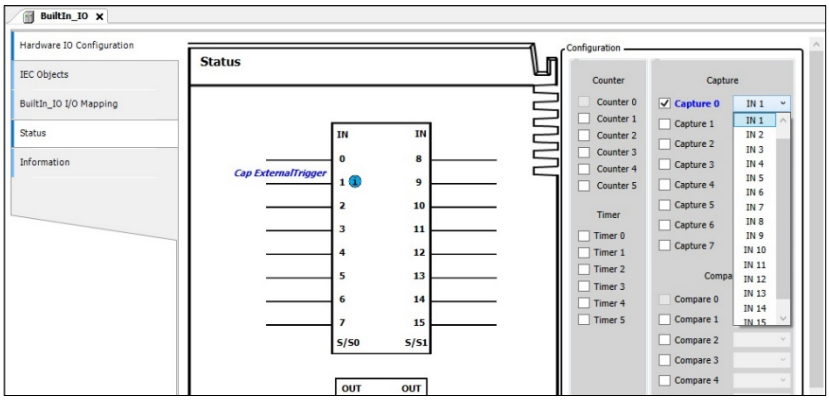


Capture function starts:

1. In the BuiltIn_IO screen, there are a total of 8 points for the capture, click one of them to start.

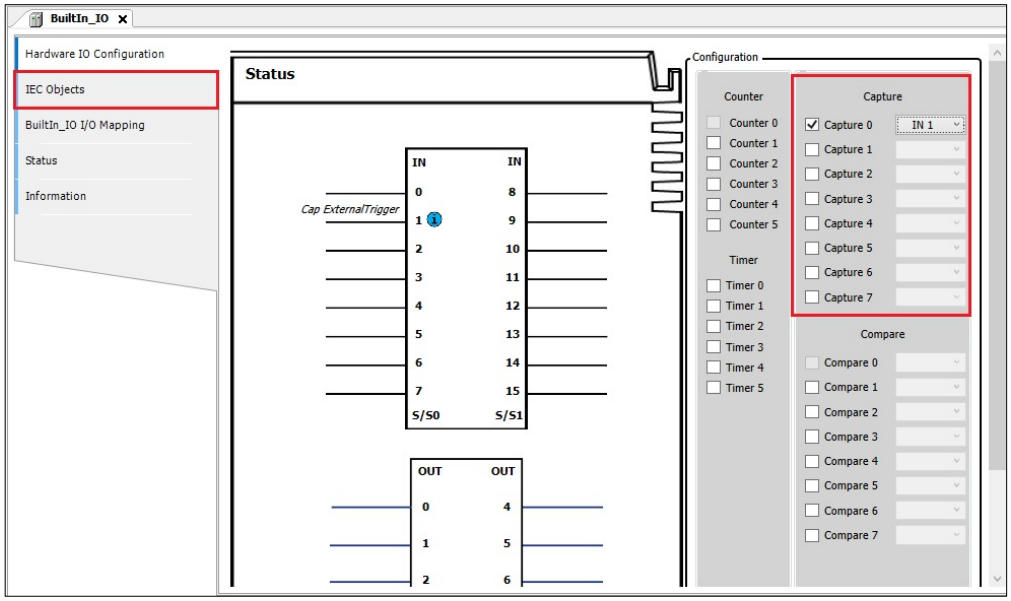


2. After starting the capturer, click the drop-down menu and select the external trigger source.

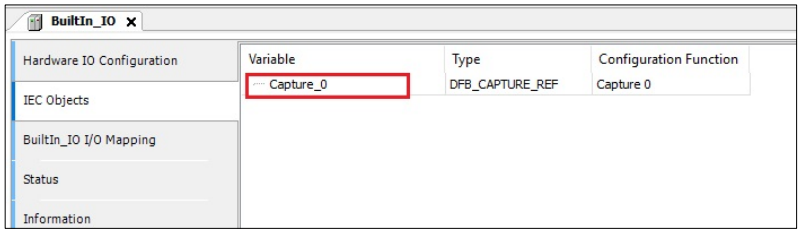


The catcher is used in the program: this variable is used by the user in the motion function block in the POU.

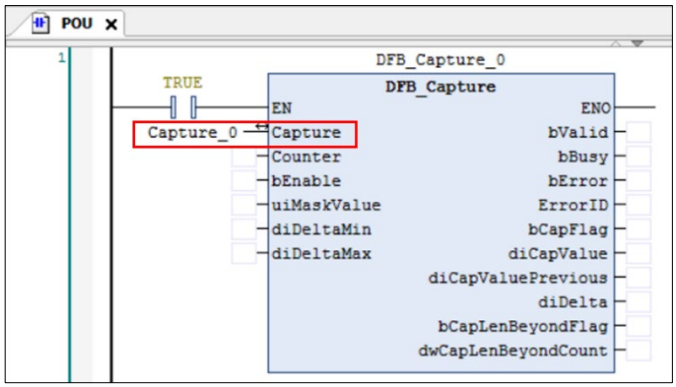
1. Click the **IEC objects** on the BuiltIn_IO screen below.



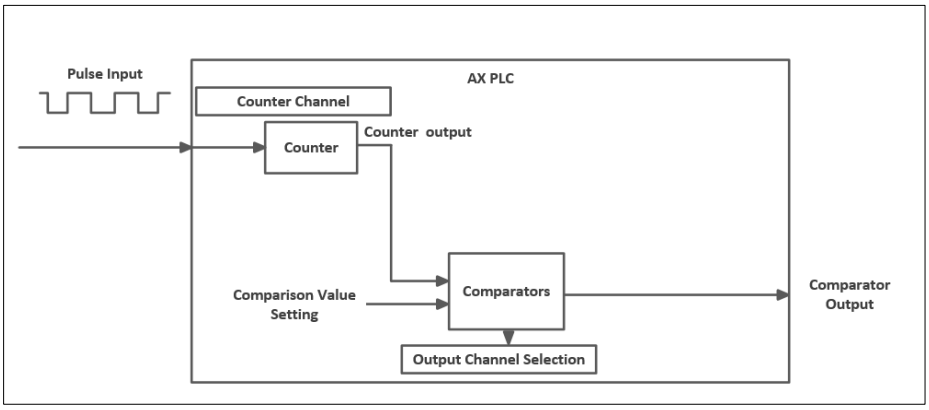
2. The following is an example of the configuration function. Capture_0 is required for the axis name used in the POU.



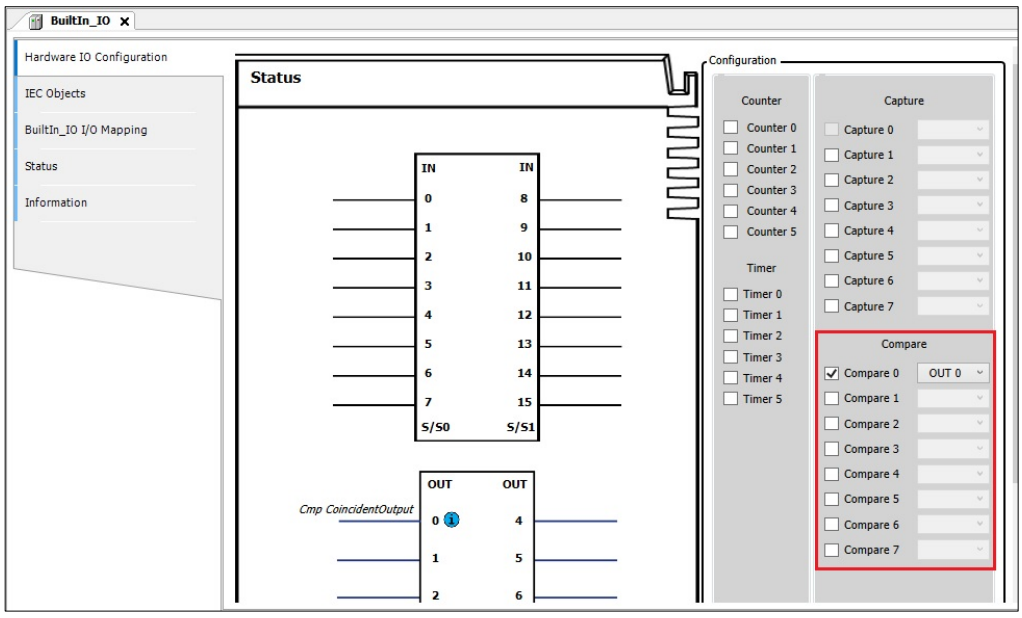
3. For the DFB_Capture function block in the POU, the input axis name is Capture_0.



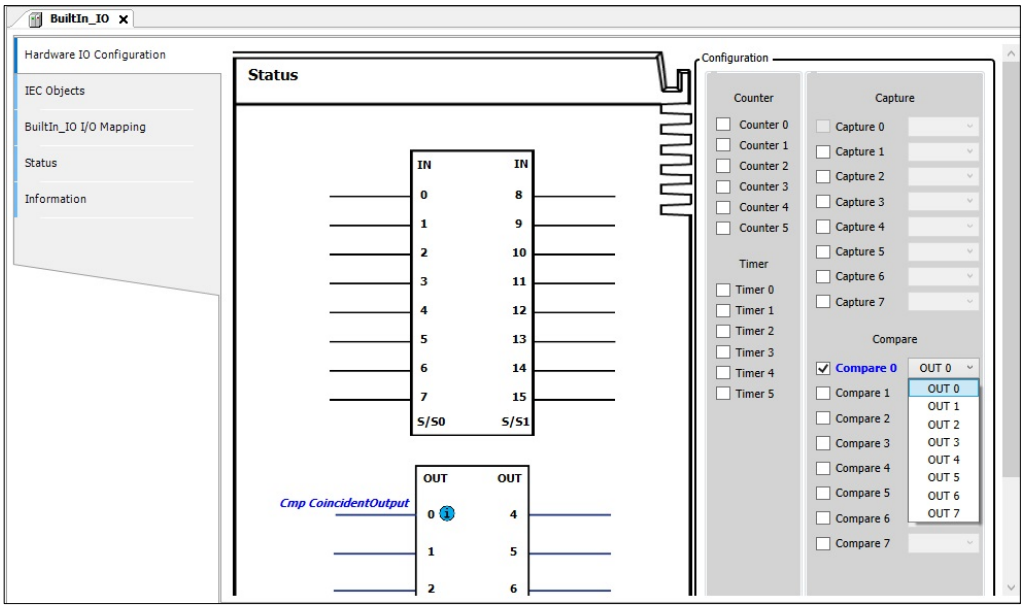
High-speed counting comparator



1. Comparator function start: In the BuiltIn_IO screen, the comparator has a total of 8 points, click one of them to start.

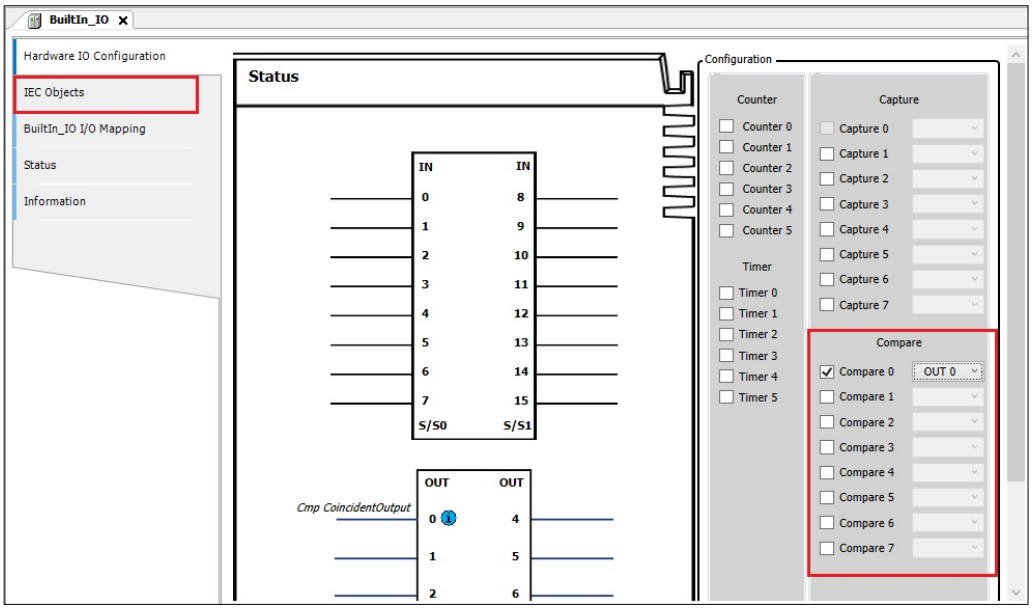


2. After starting the comparator, click the drop-down menu to select the comparator output external contact.

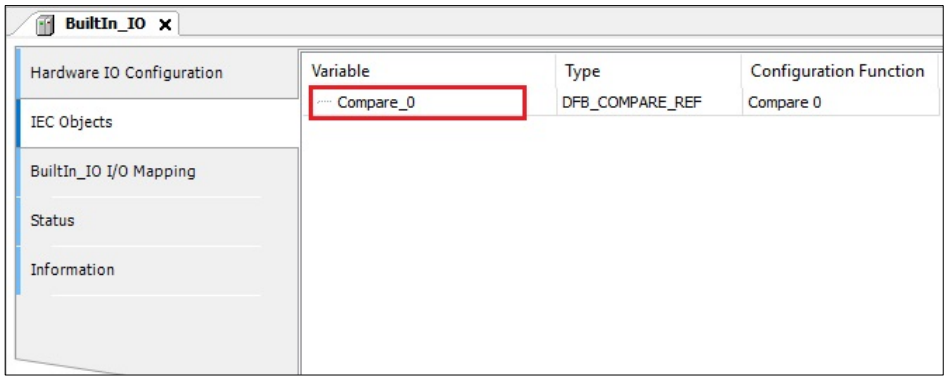


The comparator is used in the program: this variable is used by the user in the motion function block in the POU.

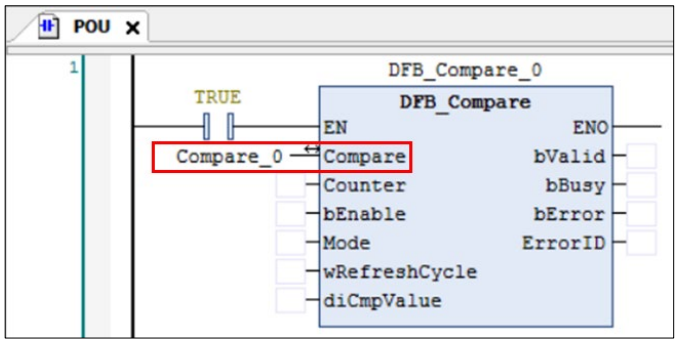
1. Click the **IEC objects** on the BuiltIn_IO screen below.



2. The following is an example of configuration function. Compare_0 should be filled in for the axis name used in POU.



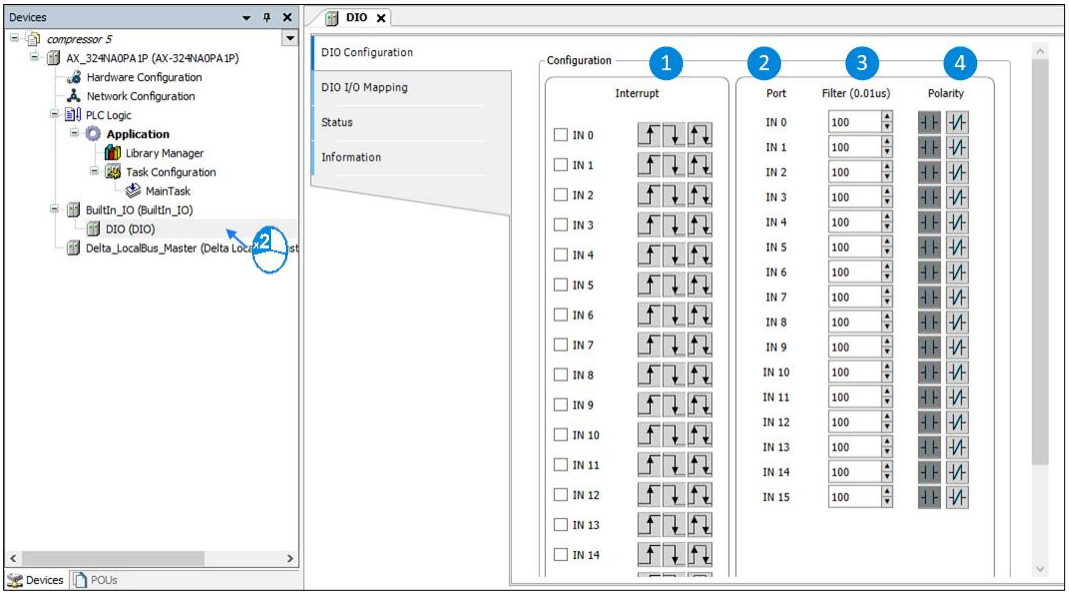
3. For the DFB_Compare function block in the POU, the input axis name is **Compare_0**.



5.2.8.2.4 DIO: DIO configuration

Here describes the DIO device, setting the interrupt, filtering, polarity and other functions of the host I/O point.

Click **DIO** twice to enter the setting screen.

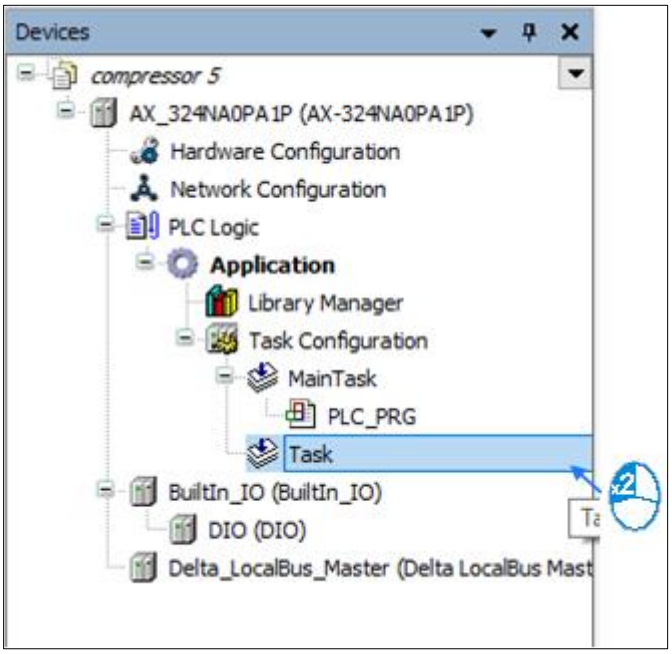


The following table lists detailed function descriptions:

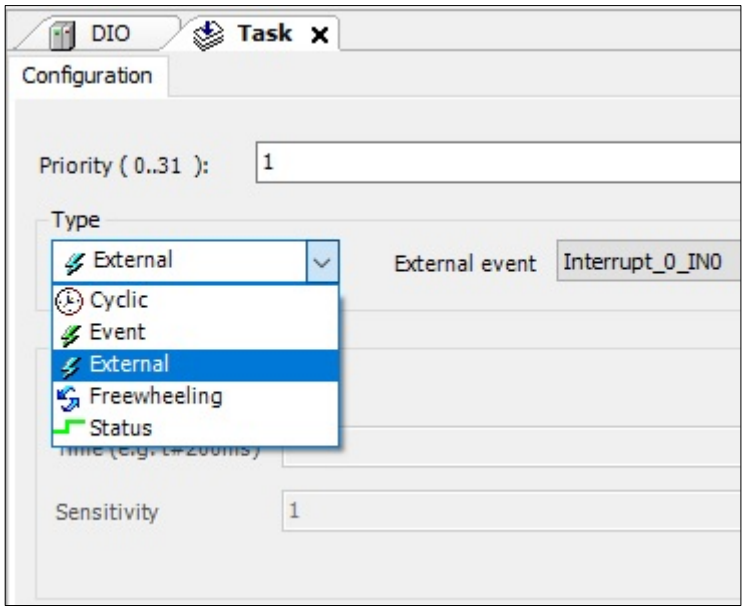
Features	Description
① Interrupt	default value
	Start external interrupt
	When the external interrupt is activated, the input contact is set as the upper differential signal.
	When the external interrupt is activated, the input contact is set as the lower differential signal.
	When the external interrupt is activated, the input contact is set as the upper and lower differential signal.
② port	Corresponding external contact number
③ Filter	100 Set the filter time (μs), the setting range is (0~100000000), and the default is 100μs.

Features	Description
④ polarity	Set the polarity of the input contact, the default value is A contact.
	Set the input contact polarity to B contact.

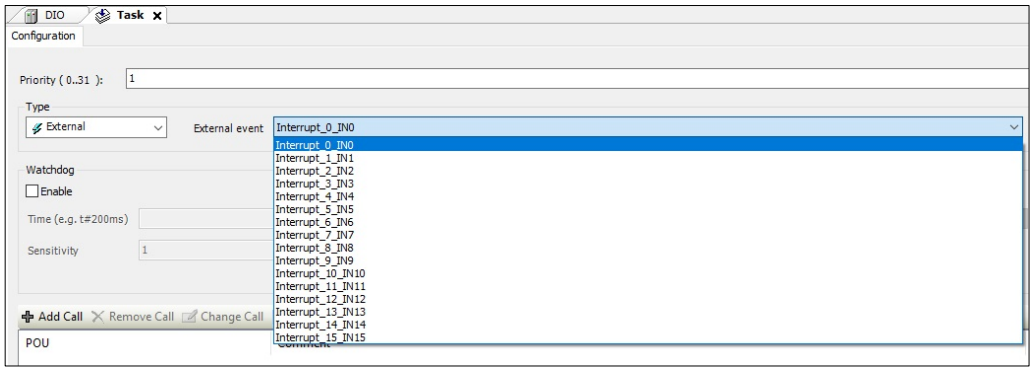
How to use I/O interrupt: After starting the interrupt function on the DIO setting screen, click the **Task** that needs interrupt function.



1. Enter the Task setting screen, and select **External** from the **Type** drop-down menu.

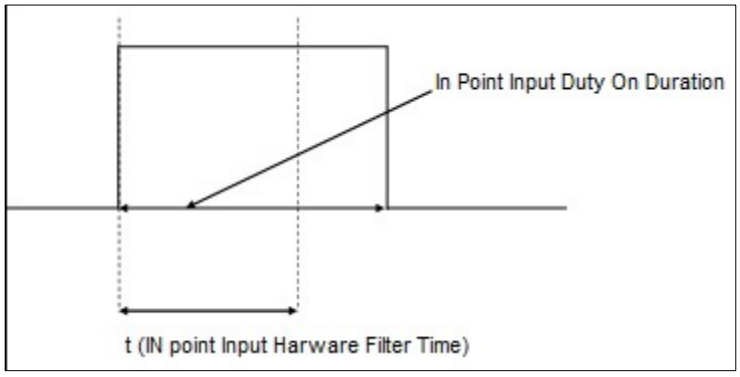


2. After selecting external, click on **External Event** to select the corresponding **Interrupt** contact.



The hardware filter time setting value is less than the input Duty On time of IN point, as shown in the following figure:

1. The input range of hardware filter is 1~50000000, and the unit is 0.1 μs.



2. The relationship between filtering frequency and time is as follows:

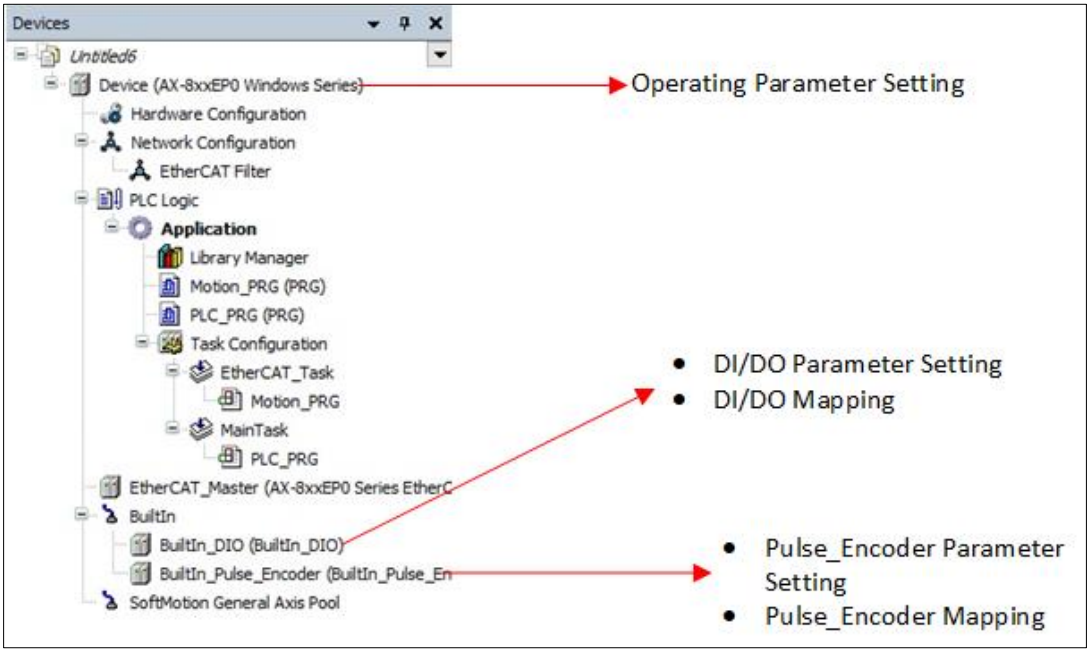
Filter frequency *2 (Hz): Filter frequency = 1 / (2 * t); t is the filter time setting value (unit 0.01 μs)

When the input frequency is higher than the filter frequency range, the signal will be filtered out. This function is for IN input points, such as DFB_Capture, DFB_Hcnt, DFB_HTmr, DFB_Compare, and body IO interrupt.

5.3 AX-8xxEP0 series setting

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

Before performing Parameter Setting in a device, user have to refer to the Operation Manual.



AX-8 is PAC-based standalone controller and supports the DIO and Pulse Encoder on controller. In addition, it can connect the remote IO R1EC.

5.3.1 System Parameters

How do I open the **System Parameters** tab?

- On the **Devices** pane, double-click the device and go to **System Parameters**.

System Parameters displays the communication and controller basic parameters.

Use the following tables to view and configure the parameters.

Item	Description
Parameter	Display the parameter name, not editable.
Type	Display the data type of the parameter, not editable.
Value	Display the initial value of the parameter or the corresponding symbol name. Double-click the field to change the value if the parameter is editable.
Default value	Display the default value of the parameter defined by the device description, not editable.
Unit	Display the unit of measure for the value (example: “ms” for milliseconds), not editable.
Description	Display the short description of the parameter specified by the device description, not editable.

Item	Description
Application Run Mode	Set whether the PLC applications are to be run automatically after the controller is powered on again. • Default Value: False **Note**: You can only change it when the controller is offline.
Serial Communication Mode	Set the Serial port communication mode. • Default Value: RS-485
Network Interface Settings Enable	Enable to apply the changed IP settings while downloading. • Default Value: True If the value is set to True, the **Apply IP settings while download** checkbox on the **System Settings** tab is selected.

5.3.2 System Settings

How do I open the **System Settings** tab?

- On the **Devices** pane, double-click the device and go to **System Settings**.

Use the following tables to view and configure the settings.

Runtime Clock

Setting	Description
Read PLCTime	Read the time and time zone from the controller. PLC Time and PLC Timezone are updated accordingly.
Date	Set the date.
Time	Adjust the time.
Write PLC Time	Write the date and time based on the selected timezone to the controller. **PLC Time** and **PLC Timezone** are updated accordingly.
Sync with Local Time	Sync all the time, date, and timezone settings with the computer time.

Time Zone

Setting	Description
Read Timezone	Read time zone information from the controller.
Write Timezone	Write the time and timezone to the controller. **PLC Time** and **PLC Timezone** are updated accordingly.

Network

Setting	Description
Apply IP settings while download	Select to apply the changed IP address after download.
IP Address Mode	Select the IP address mode of Static or DHCP.
IP address	Set the IP address, subnet mask, and default gateway in Static mode.
Read from PLC	Read and update the IP address from the active device.

Retain Settings

Setting	Description
Original mode	Keep CODESYS persistent variable mode.
%M mode	Set up the %M address to retain persistent variables. • **Start Memory Address**: Enter the value for the start memory address. The default value is 5144576. The maximum value is 5242879. • **End Memory Address**: The end memory address is 5242879, not editable.
Clear	Clear all the %M addresses assigned to the variables.

Retain Size

This function is only available for AX-8 Linux series controllers.

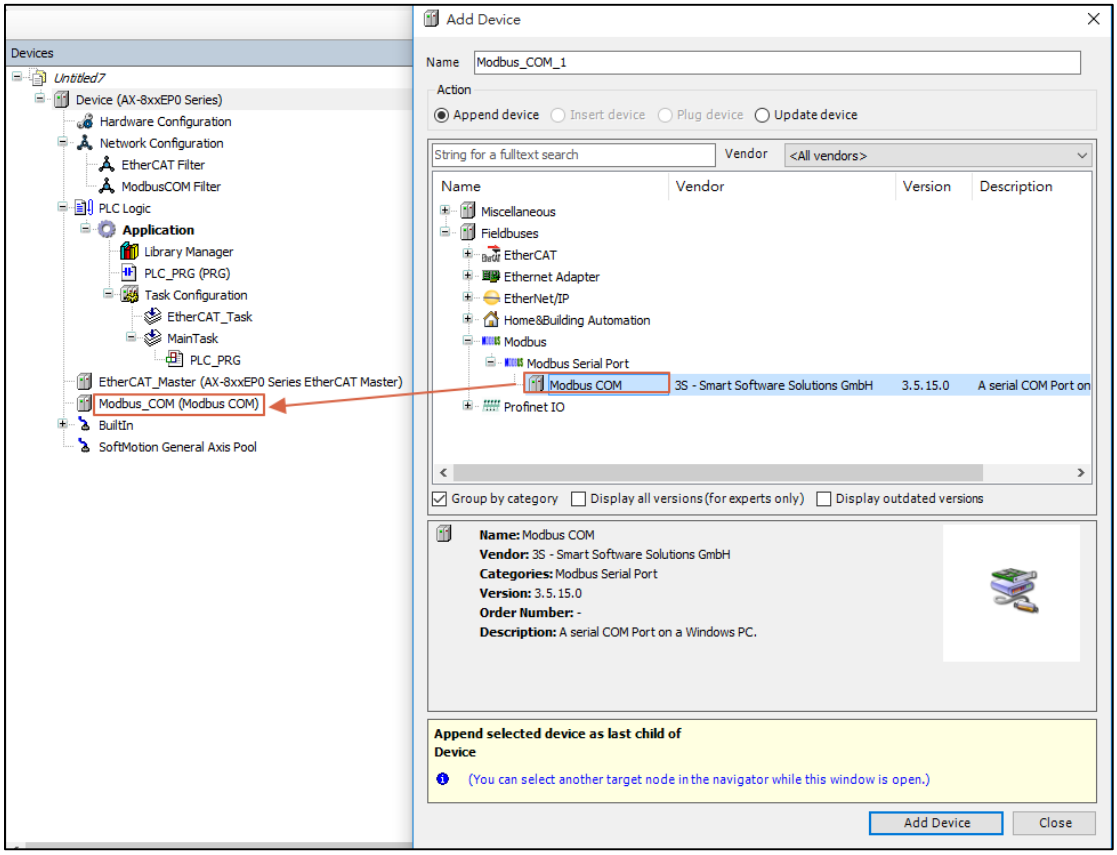
Setting	Description
Default size	Select to use the default retain memory size.
Extended size	Select to extend the retain memory size.

5.3.3 COM Port Settings

AX-8 series support MODBUS COM Setting. A MODBUS network consists of a MODBUS master and one or more MODBUS slaves.

5.3.3.1 MODBUS COM Port

Right-click on the AX-8xxEP0 device **node** > **Add Device** > **Fieldbus** > **MODBUS** > **MODBUS Serial Port** > **MODBUS COM**.



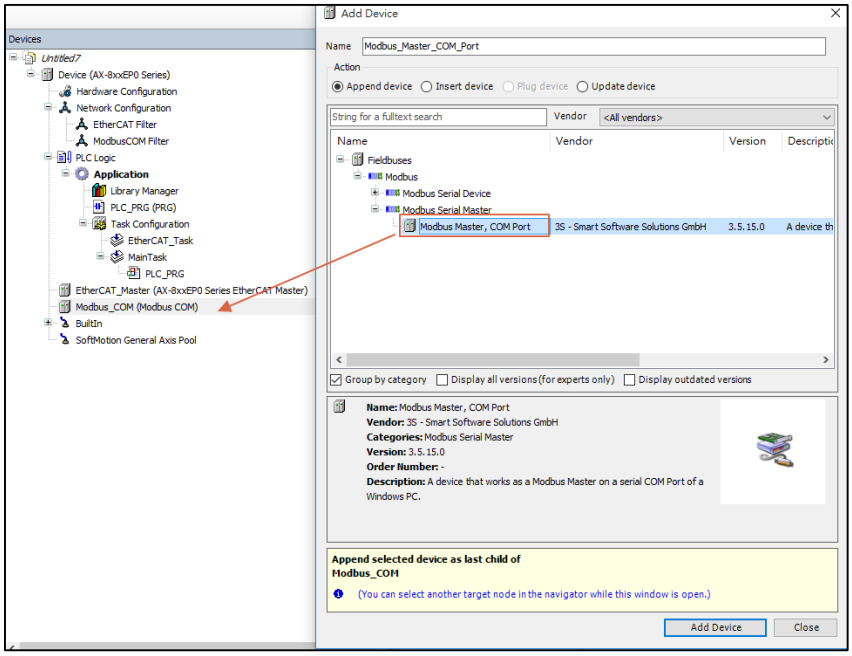
- General Tab

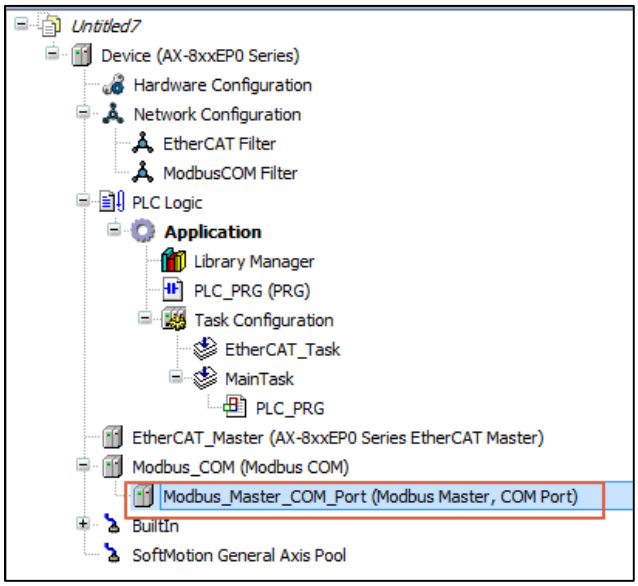
Serial Port Configuration	
Item	**Description**
COM port	Number of the serial port
Baud rate	Parameter of the serial port
Parity	Parity
Data bits	Data Length
Stop bits	Stop bits

5.3.3.2 MODBUS Master COM Port

The MODBUS Master function is used to set the entry when the AX-8 series controller is used as the MODBUS master station.

Right-click on the MODBUS COM **node** > **Add Device** > **MODBUS** > **MODBUS Serial Master** > **MODBUS Master, COM Port** to add to the project tree.





General Tab

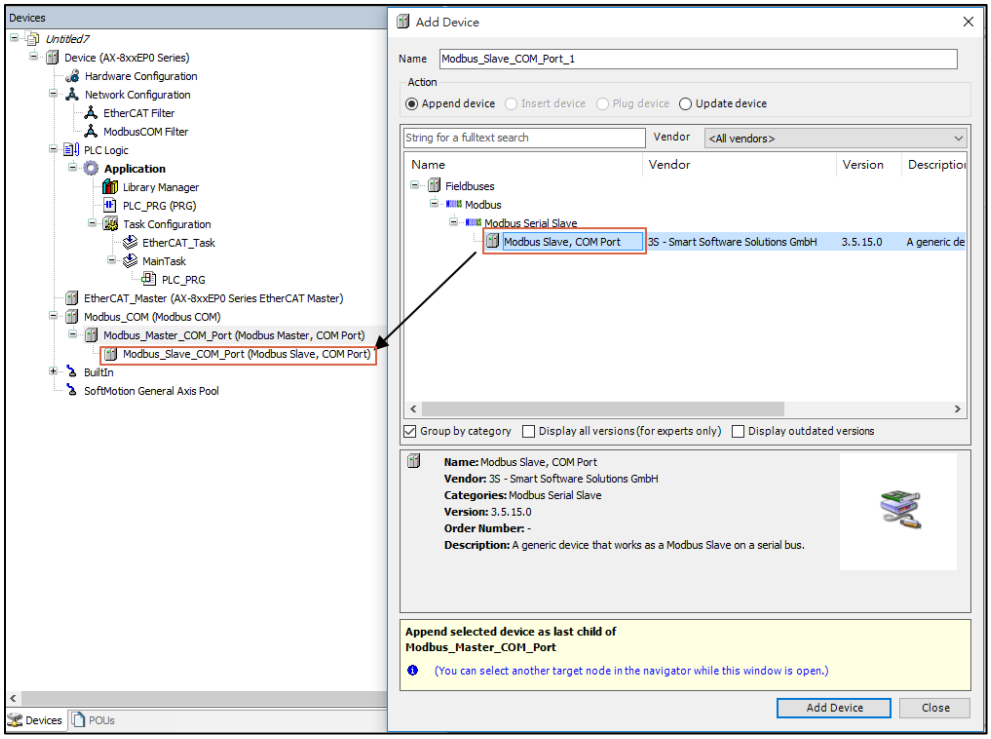
This includes the basic settings for MODBUS communication.

Item	Description
Transmission mode	• RTU: Transmission in binary • ASCII: Transmission in ASCII code (currently not supported by all drivers)
Response timeout (ms)	The response timeout sets the time interval that the master will wait for the answer from a slave node. If the nodes do not answer within this time interval, then an error is recorded for the implicit slave function block. The value entered for the time interval is also the default value for each node. For each node, user can still set a specific value within its slave configuration.
Time between frames (ms)	The time between the frames is the send pause of the master between the last response (and a timeout) and the next request. Use this parameter to make the communication less liable to disruption.
Auto-restart communication	• : After a communication error, CODESYS automatically confirms the error and attempts to continue executing the MODBUS command. • : The error must be explicitly confirmed in the slave function block.

5.3.3.3 MODBUS Slave COM Port

The MODBUS Slave function is used to set the target slave station to be connected when the AX-8 series controller is used as the MODBUS master station.

Right-click **MODBUS Master COM port** to choose **Add Device** to add **MODBUS Slave COM port**.



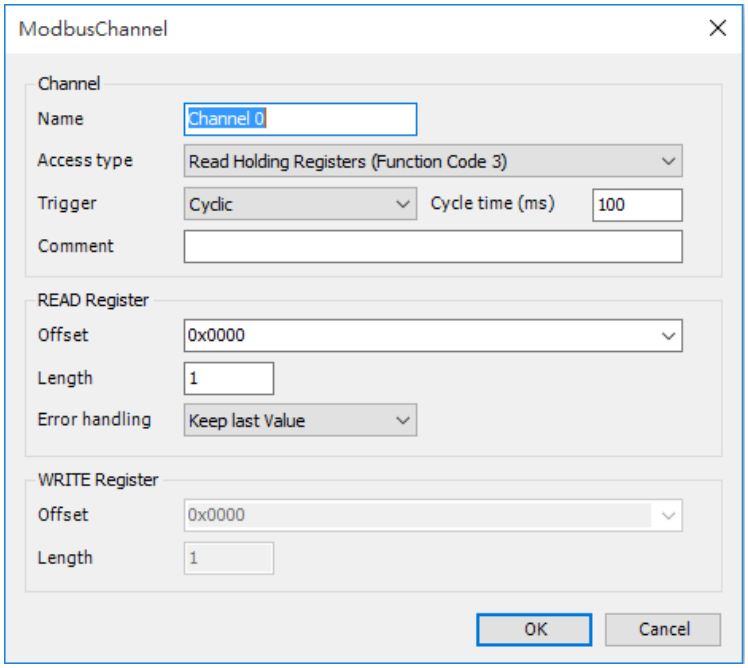
General

Item	Description
Slave address [1..247]	Address of a serial MODBUS device (value between 1 and 247)
Response timeout [ms]	Time interval for the master to wait for the response from the slave. This is especially configured for this slave node and overwrites the general response timeout setting of the respective master.

Channel

User use this tab to define slave channels. Each channel represents a single MODBUS request.

Item	Description
Add Channel	Opens the dialog **MODBUS Channel**. Here user can create new channels for this MODBUS slave. In the device description file of the MODBUS slave, user can predefine descriptions for individual MODBUS registers or entire channels.



The following table describes each item in the MODBUS channel window:

Aisle:

Channel	
Item	**Description**
Name	A string that contains the name of the channel
Access type	• Read coils (function code 1) • Read discrete inputs (function code 2) • Read holding registers (function code 3)

	• Read input registers (function code 4) • Read single coil (function code 5) • Write single register (function code 6) • Write multiple coils (function code 15) • Write multiple registers (function code 16) • Read/Write multiple registers (function Code 23)
Trigger	• **CYCLIC**: The request occurs periodically. • **Rising edge**: The request occurs as a reaction to a rising edge of the Boolean trigger variables. The trigger variable is defined in the tab **I/O Mapping**. • **Application**: The MODBUS request is triggered by the PLC application. This happens by means of the MODBUS Channel function block, which is included in the respective I/O driver library. This function block provides complete control of and information about the execution of this command, for example the start time, the processing time, and the result.
Cycle time (ms)	For Trigger = **CYCLIC**: Request interval **Note:** The request interval should be the same as or a multiple of the cycle time of the application.
Comment	Description of the channel

READ Register

Item	Description
Offset	Start address where reading should start (value range 0 ~ 65535)
Length	Number of registers to be read (for word access) or number of discrete inputs to be read (for bit access)
Error handling	Defines what should happen to the data in case of a communication error • Set to ZERO

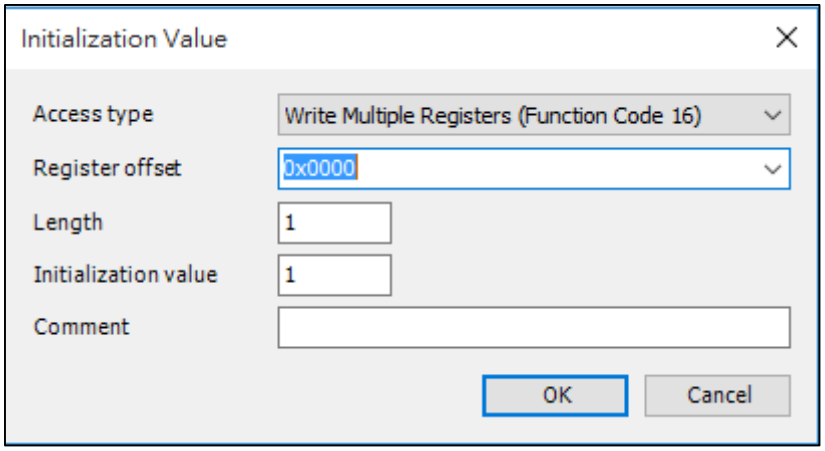
READ Register	
	• Keep last value
WRITE Register	

Item	Description
Offset	Number of the register to be written to (value range 0 ~ 65535)
Length	Number of registers to be written to (= Words) The value range of the parameter depends on function code.

MODBUS slave initialization tab:

User uses this tab to define initialization commands. Initialization commands are executed one time when starting the bus or activating the slave (setting the *Enabled* flag of the slave instance). When setting up or editing a slave initialization value, the following parameters are available in the respective dialog boxes:

Item	Description
Move Up Move Down	The order of channels also determines the order of initialization.
New	Opens the dialog box **Initialization Value**. The initialization commands are defined.

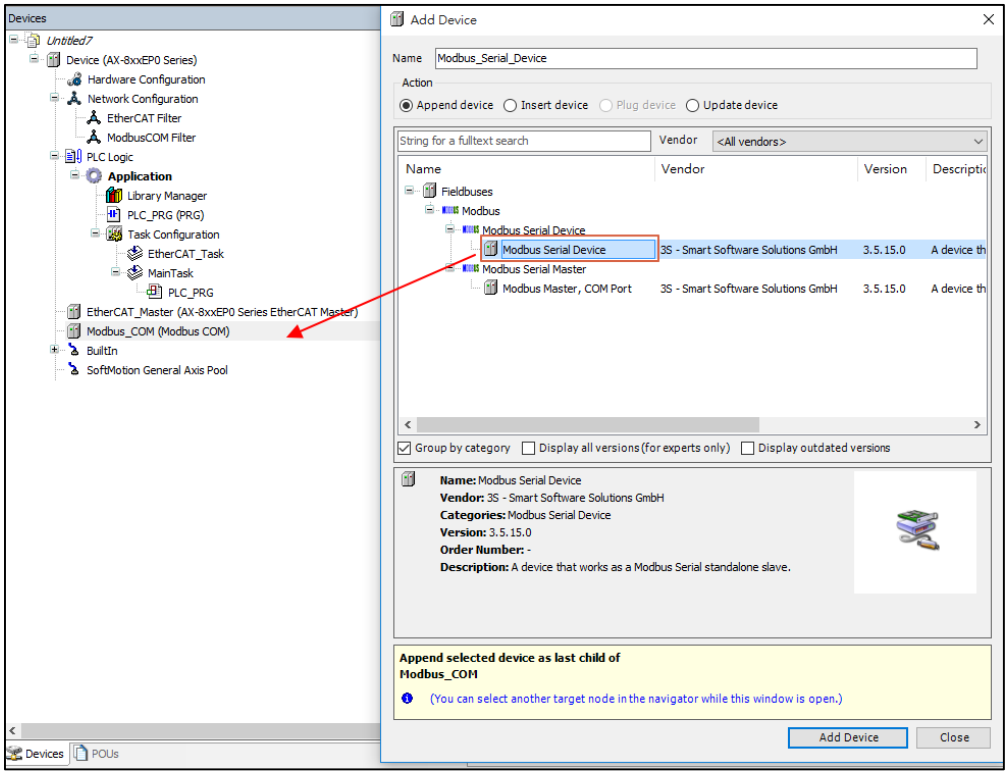


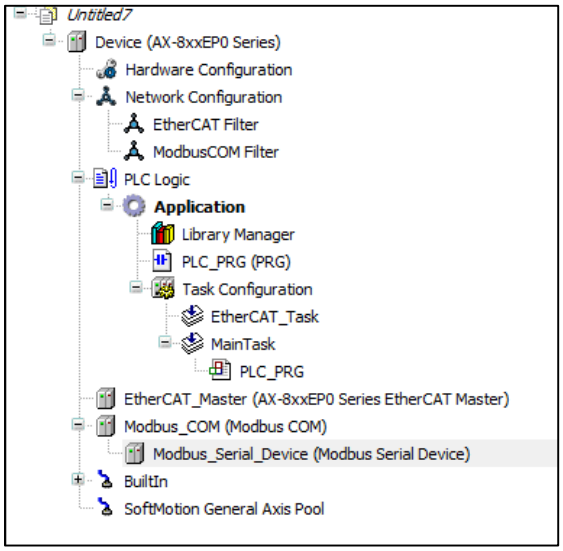
Item	Description
Access Type	• Write single coil (function code 5) • Write single register (function code 6) • Write multiple coils (function code 15) • Write multiple registers (function code 16)
Register offset	Number of the register to be written to (value range 0 ~ 65535)
Length	Number of registers to be written to (= Words) Value range of the parameter depends on function code
Initialization value	Initialization value for the register
Comment	Short description of the data

5.3.3.4 MODBUS Serial Device

The MODBUS Serial Device function is used to set the entry when the AX-8 series controller is used as a MODBUS slave station.

Right-click on the **MODBUS_COM node** > **Add Device** > **MODBUS Serial Device** > **MODBUS Serial Device** to add it to the project tree.





General

Item	Description
Unit ID	Unit ID of the slave
Watchdog	☑ Activates the watchdog function The incoming data (Holding Registers / %I range) is set to zero when the MODBUS device does not receive any valid query from the master.
Holding registers (%IW)	Number of holding registers: possible values are 1 - 500. The maximum number can be limited in the device description. **Writable**: ☑: For the holding register, writable I/O mappings (%QW addresses) are generated instead of read-only I/O mappings (%IW addresses). This allows the holding registers to be set by the MODBUS device application (= server application) by means of the usual I/O mapping.
Input registers (%QW)	Number of input registers: possible values are 1 - 500. The maximum number can be limited in the device description.

MODBUS defines for I/O data a data model with four areas (Start addresses).

- Coils: Outputs of type BIT
- Discrete Inputs: Inputs of type BYTE
- Holding Register: Outputs of type WORD (16-bit)
- Input Register: Inputs of type WORD (16-bit)

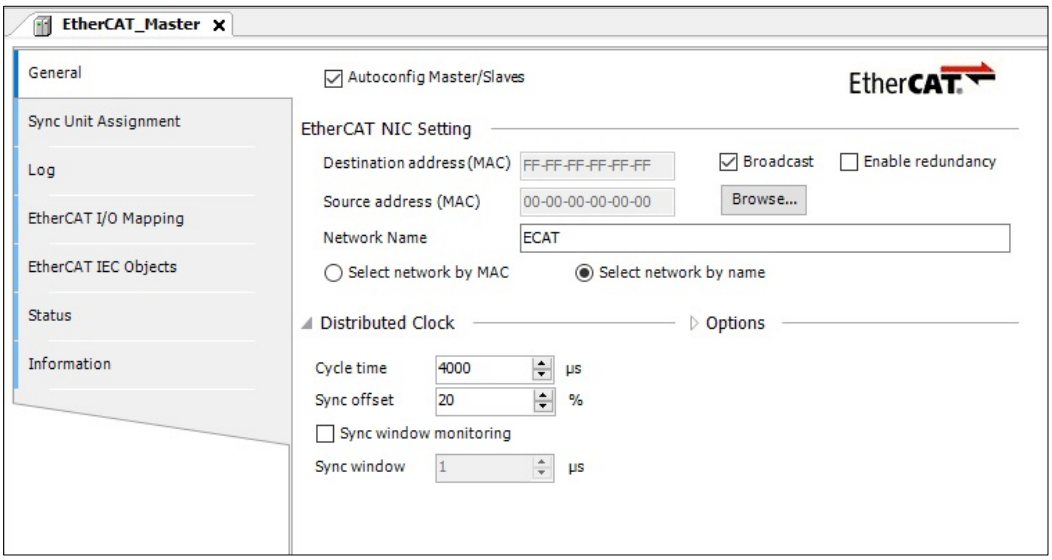
Inputs and outputs are included as with all fieldbuses, but from the point of view of the host. These data areas have any number of logical start offsets as defined by the manufacturer. The I/O data behind the MODBUS data areas can overlap or be independent of each other.

5.3.4 EtherCAT Settings

The configuration of EtherCAT modules is based on the device description files for the master and slave devices employed and can be adapted in the project in configuration dialogs. In order to ensure the simplest and most error-free use possible, we recommend for standard applications that user activate the option for **Automatic Configuration** of the master, so that the majority of the configuration settings are performed automatically.

5.3.4.1 EtherCAT Master

The following figure shows the general tab window displayed after clicking EtherCAT _Master in the project tree. The basic configuration settings of EtherCAT master can be set in the window.



- General

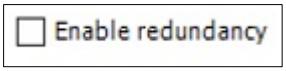
Item	Description
Autoconfig Master/Slaves	☑ The greater part of the master and slave configuration is accomplished automatically, based on the device description file and implicit calculations. The dialog for the FMMU/Sync settings is not available. Even if this option of the master is activated, an expert mode can be activated explicitly for each individual slave that permits the manual editing of the automatically generated process data configuration.

EtherCAT NIC Setting

Item	Description
Destination address (MAC)	MAC address of the device in the EtherCAT network that is to receive the telegrams. Options • **Broadcast***:* no **destination address (MAC)** needs to be specified. • **Enable redundancy**: is activated if the bus is constructed in a ring topology and redundancy is to be supported. With this function, the EtherCAT network remains functional even in the case of a cable breakage. If this function is activated, the parameters must be defined in **Redundancy EtherCAT NIC Settings** area.
Source address (MAC)	MAC address of the controller (target system) or network name (name of the card, i.e. PLC (target system))
Network Name	Name or MAC of the network, depending on which of the following options is activated:

Select network by MAC	☑: Network is specified by the MAC-ID. The project then cannot be used on another device, since each network adaptor has a unique MAC-ID.
Select network by Name	☑: Network is identified by the network name and the project is device-independent.
Browse	Scans the network for the MAC-IDs or names of the target devices that are presently available.

Redundancy EtherCAT NIC Setting

These settings are shown only if **Enable redundancy** 

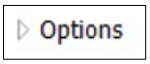
option is activated. Here the parameters of the additional device are defined in accordance with the description to **EtherCAT NIC Settings**.

Distributed Clocks

Item	Description
Cycle time	Time interval after which a new data telegram is dispatched on the bus. If **Distributed clocks** function is activated in the slave, the master cycle time specified here is transferred to the slave clocks. In this way a precise synchronization of the data exchange can be achieved, which is important in particular if spatially distributed processes require simultaneous actions. Simultaneous actions are, for example, applications in which several axes must execute coordinated movements at the same time. A very precise, network-wide time-based with a jitter of substantially less than 1 microsecond can be achieved in this way.
Sync offset	Enables the time delay of the sync interrupt of the EtherCAT slave to be adjusted to the cycle time of the PLC. Normally the PLC cycle starts 20% later than the sync interrupt of the slave. This means that the PLC cycle could be delayed by 80% of the cycle time without a message being lost.

Sync window monitoring	☑ Synchronization of the slaves can be monitored.
Sync window	Time for **Sync window monitoring**. If the synchronization of all slaves lies within this time window, then the variable xSyncInWindow (IoDrvEthercat) is set to TRUE, otherwise to FALSE.

Options

Click the **option** 

to expand the following items:

Item	Description
Use LRW instead of LWR/LRD	☑ Direct communication from slave to slave is possible. In place of separate read (LRD) and write commands (LWR), combined read/write commands (LRW) are used.
Enable messages per task	☑ Read and write commands, i.e. the handling of the input and output messages, can be controlled with various tasks.
Automatic restart slaves	☑ In the event of a communication breakdown, the master immediately attempts to restart the slaves.

Master Setting

User can edit this setting only when user disable the **Automatic configuration** 

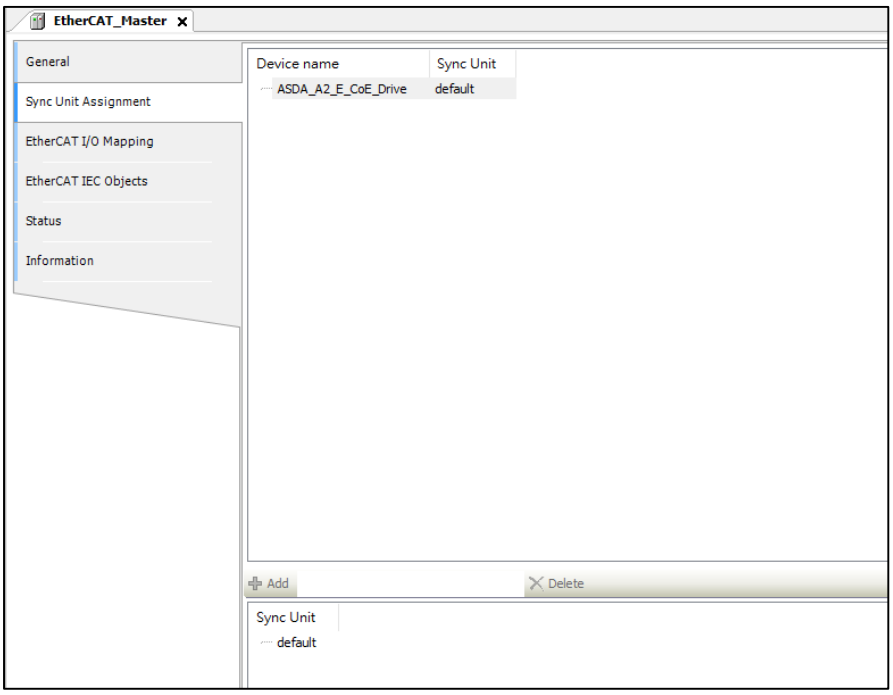
master/slave option, otherwise it will be configured automatically, and this function will be displayed at the bottom of the general tab.

Image in address	First logical address of the first slave for input data
Image out address	First logical address of the first slave for output data

- Sync Unit Assignment

  This tab shows all slaves that are inserted below a particular master with an assignment to the sync units. With the EtherCAT sync units, several slaves are configured into groups and subdivided into smaller units. For each group,

the working counter can be monitored for an improved and more precise error detection. As soon as a slave is missing in a sync unit group; the other slaves in the group are shown as missing. Detection occurs immediately in the next bus cycle, as the working counter is continuously checked. With the device diagnostics, the missing group can be remedied as quickly as possible.



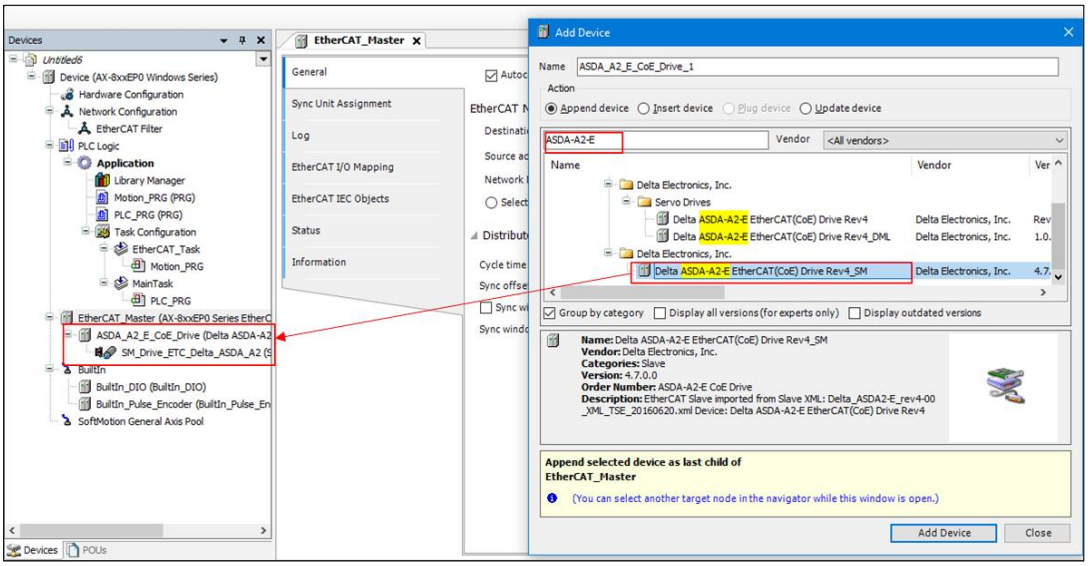
Item	Description
Device Name	Name of the slave
Sync Unit	Name of the selected sync unit. User can combine individual devices or whole groups (multiple selection) into one sync unit group.
Add	When user type a name in the text field, user can create a new sync unit.
Delete	Removes the selected sync unit. If slaves are assigned to the group to be deleted, then a warning is displayed. If user confirm the dialog prompt by clicking **Yes**, then these devices are reassigned to the default group.

5.3.4.2 EtherCAT Slave

Support ASDA-A2-E, B3-E, A3-E under EtherCAT Master. User can install 3rd party Device xml file to add under EtherCAT Master.

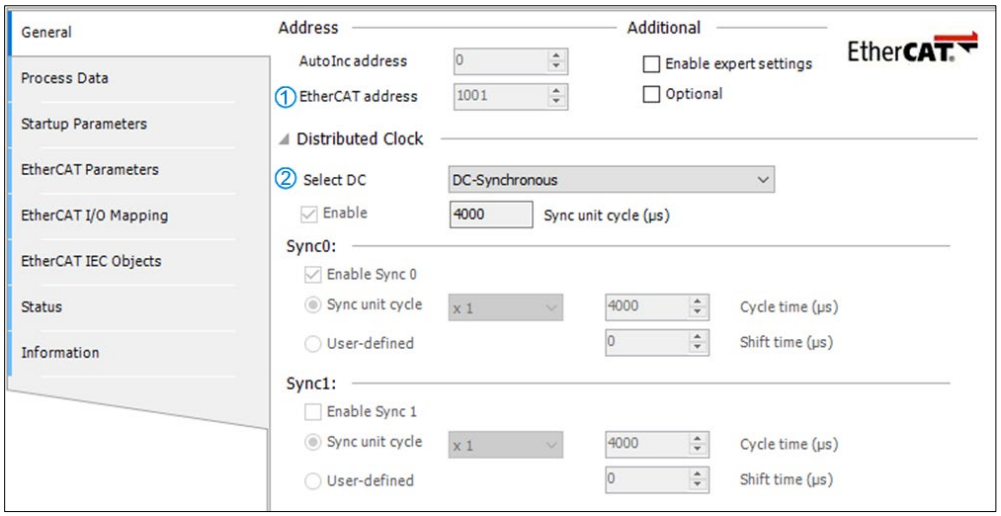
Right-click **EtherCAT Master** to choose **Add Device** and add **ASDA-A2-E_SM**.

Note: Users can install third-party device xml files in the device storage library. For detailed steps, please refer to Section 7.1 Device Storage Library.



- **General Tab**

  Here we will introduce the EtherCAT_Slave window (double-click the device tree ASDA_A2_E_CoE_Drive). There are two ways to add a slave: the first is to add the slave device of the selected axis, and the second is to add a scan method.



1. Address

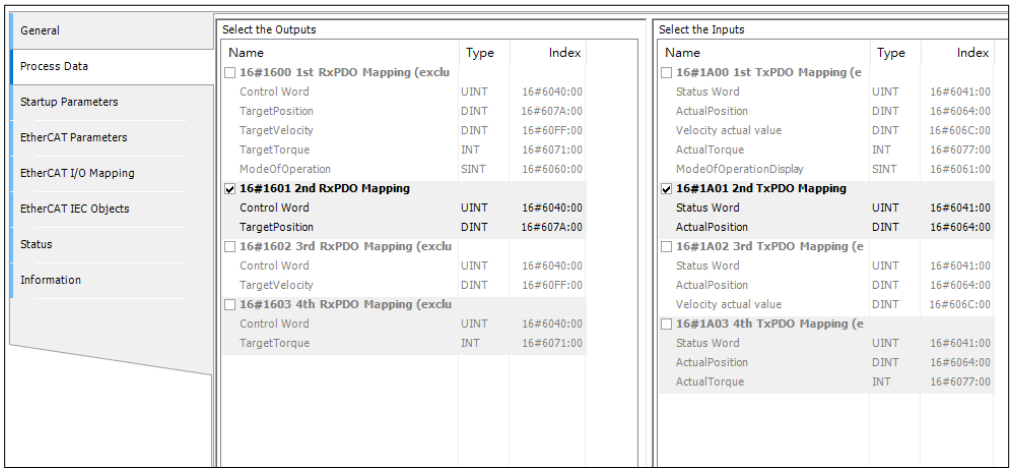
① EtherCAT address: address assigned to the slave by the master during startup.

2. Distributed Clocks

② Select DC: Set the slave distributed clock.

- **Process Data**

The data exchange on the EtherCAT network is a PDO mapping of CoE, which is a periodic data exchange between the master and the slave architecture; the data that the slave sends to the master will be packaged in TxPDO, and the slave will read the data is included in RxPDO. The input settings and output settings on the "PDO Mapping" page list the PDOs available for data exchange, and the PDO content can be edited. In the ESI file of the device, the contents of the preset checked PDO and PDO have been defined, and according to the definition of ESI, the PDO content of some devices allows user to edit it by themselves.



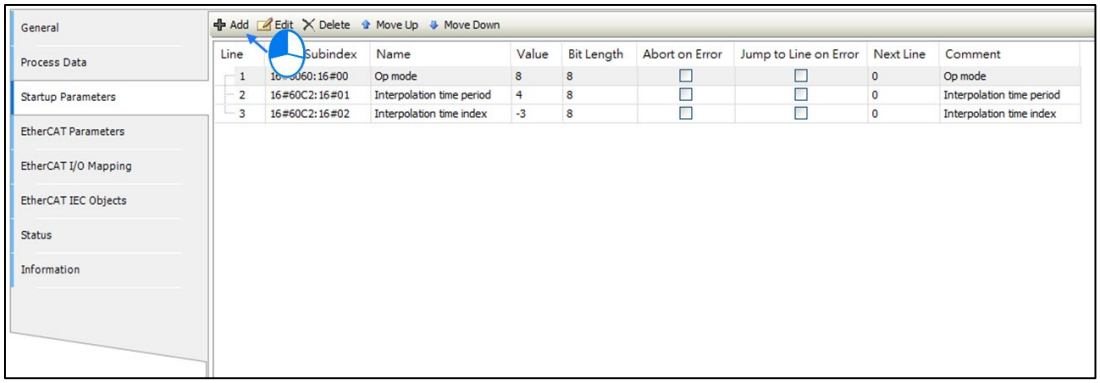
User can choose different groups of PDO to use according to their needs. The more data user choose for PDO, the greater userr PLC performance.

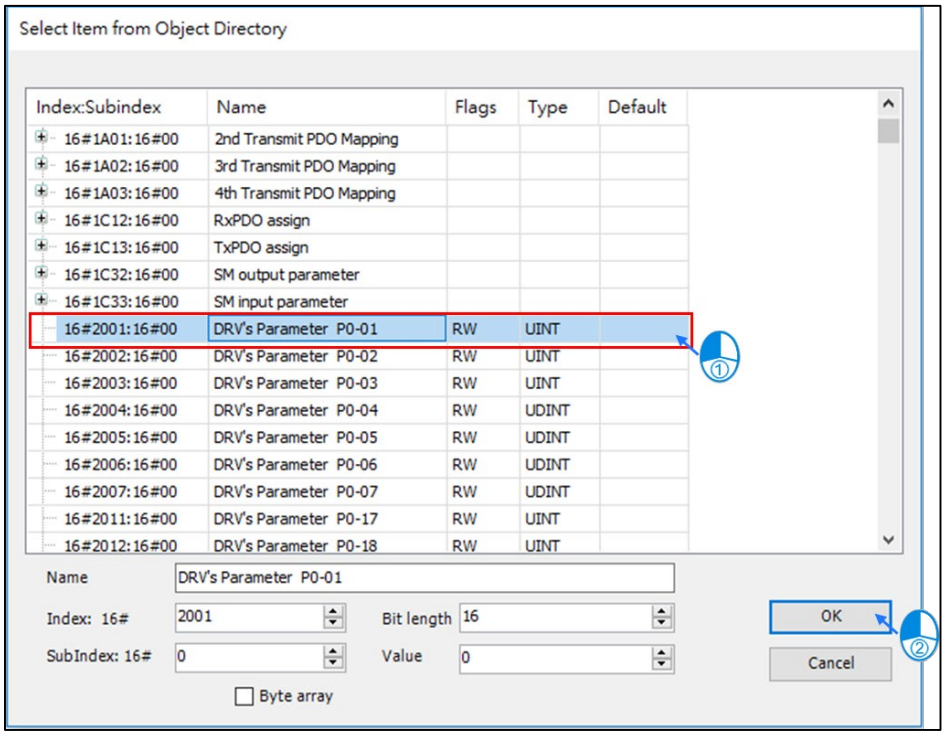
- **Startup Parameters**

After the EtherCAT communication is established, the master station will download all the parameters in the table to the slave station, and the list will bring out the pre-defined commands of the ESI file. The user can add, delete or modify as required.

Item	Description
Line	Line number
Bit Length	Bit length of the SDO
Abort on Error	☑: The transfer is interrupted in the case of an error with error status.
Jump to Line on Error	☑: In case of error the transfer is resumed with the SDO at the specified **Line**.
Next Line	☑: The transfer is resumed with the SDO at the next line.
Comment	Input field for comment
Move Up	Moves the selected line upwards by one line
Move Down	Moves the selected line downwards by one line
Add	Opens the dialog **Select and entry from the object directory**. In this dialog, user can change the parameters of the SDO before the SDO is added to the configuration. By specifying new index/sub index entries, a new object can be added to the SDO that is not yet described in the EDS file. This is useful if only an incomplete object directory or none at all is present.
Delete	Removes the selected entry.
Edit	Opens the dialog **Select an Entry from the Object Directory** in order to change the parameters of the selected SDOs or IDNs in the table.

After pressing the **Add** button, the **Select Item from Object Directory** window will appear. First, select the parameter to be written from the directory. After clicking **OK**, it will be added to the command list.


General
Process Data
Startup Parameters
EtherCAT Parameters
EtherCAT I/O Mapping
EtherCAT IEC Objects
Status
Information
Add
Edit
Delete
Move Up
Move Down
Line
Subindex
Name
Value
Bit Length
Abort on Error
Jump to Line on Error
Next Line
Comment
1 Op mode 8 8 0 Op mode
2 16#60C2:16#01 Interpolation time period 4 8 0 Interpolation time period
3 16#60C2:16#02 Interpolation time index -3 8 0 Interpolation time index

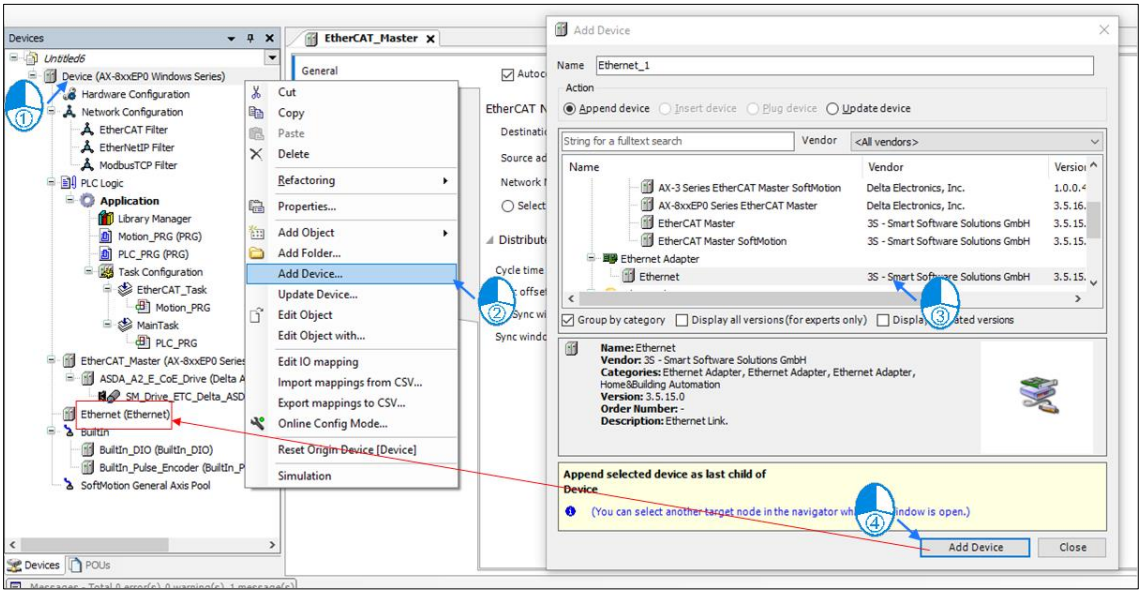

Select Item from Object Directory
Index:Subindex
Name
Flags
Type
Default
16#1A01:16#00 2nd Transmit PDO Mapping
16#1A02:16#00 3rd Transmit PDO Mapping
16#1A03:16#00 4th Transmit PDO Mapping
16#1C12:16#00 RxPDO assign
16#1C13:16#00 TxPDO assign
16#1C32:16#00 SM output parameter
16#1C33:16#00 SM input parameter
16#2001:16#00 DRV's Parameter P0-01 RW UINT
16#2002:16#00 DRV's Parameter P0-02 RW UINT
16#2003:16#00 DRV's Parameter P0-03 RW UINT
16#2004:16#00 DRV's Parameter P0-04 RW UDINT
16#2005:16#00 DRV's Parameter P0-05 RW UDINT
16#2006:16#00 DRV's Parameter P0-06 RW UDINT
16#2007:16#00 DRV's Parameter P0-07 RW UDINT
16#2011:16#00 DRV's Parameter P0-17 RW UINT
16#2012:16#00 DRV's Parameter P0-18 RW UINT
①
Name
DRV's Parameter P0-01
Index: 16#
2001
Bit length
16
SubIndex: 16#
0
Value
0
Byte array
OK
Cancel
②

5.3.5 EtherNet Setting

Ethernet Setting related with Controller Ethernet parameter setting. User needs to add the Ethernet Adapter Device to set parameter and internet related function, and all internet functions are under **Ethernet** device, like MODBUS TCP and EtherNet/IP.

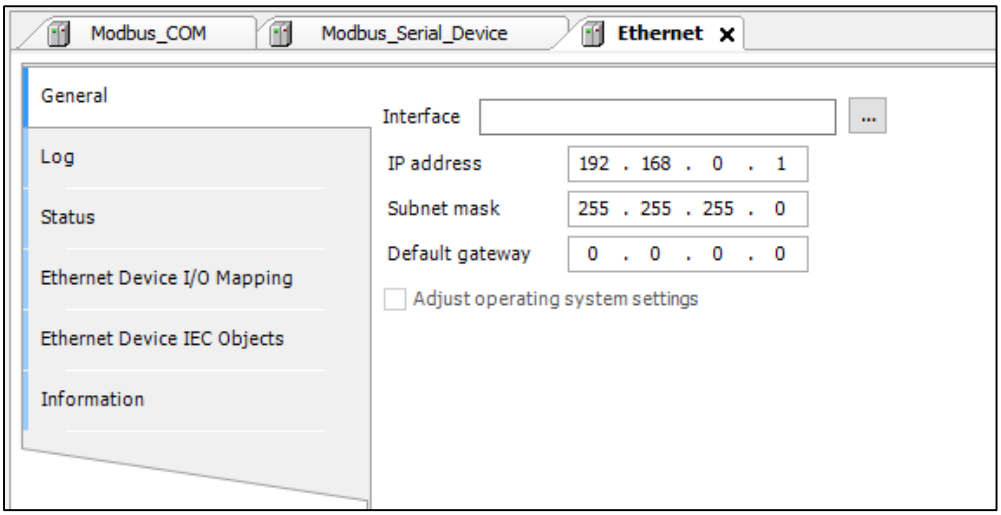
Add Ethernet Adapter Device*:*

Right-click on the Device node > **Add Device** > **Ethernet Adapter** > **Ethernet** > **Add Device**.



- EtherNet Device – General

  The user can set the controller network IP address and gateway interface on this tab.

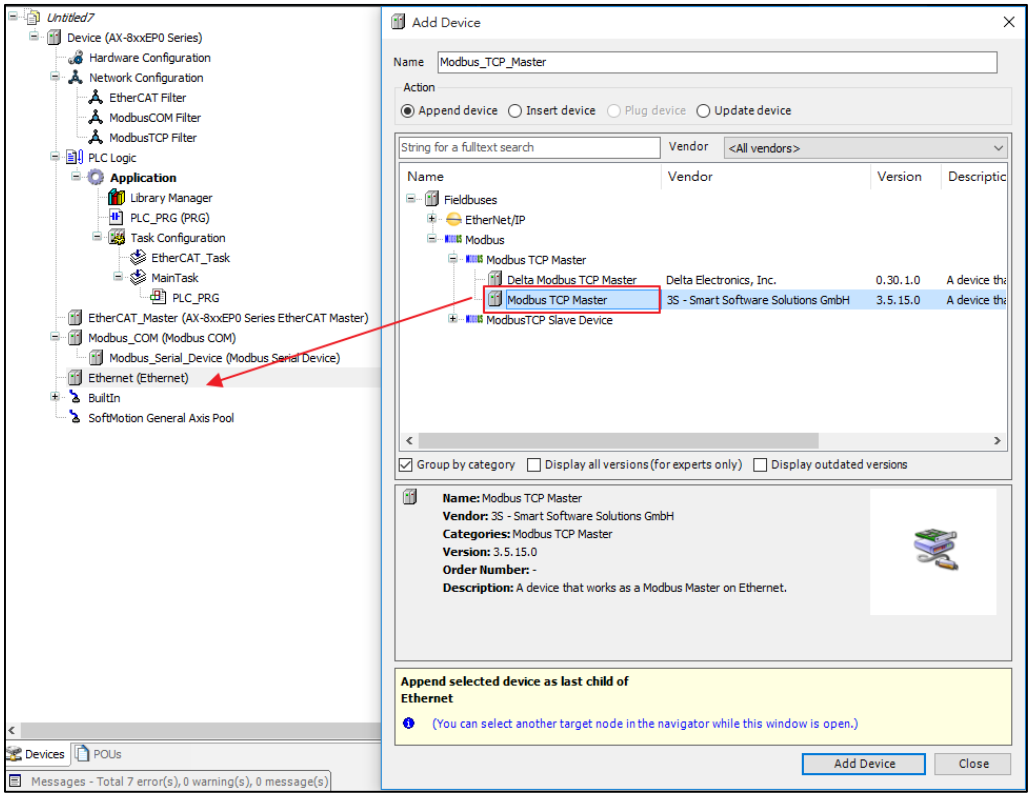


Interface	Current controller communicate interface
Interface	Current controller communication interface
IP address Subnet mask Default gateway	IP address Setting
Adjust operating system settings	Replace the current parameter to Controller

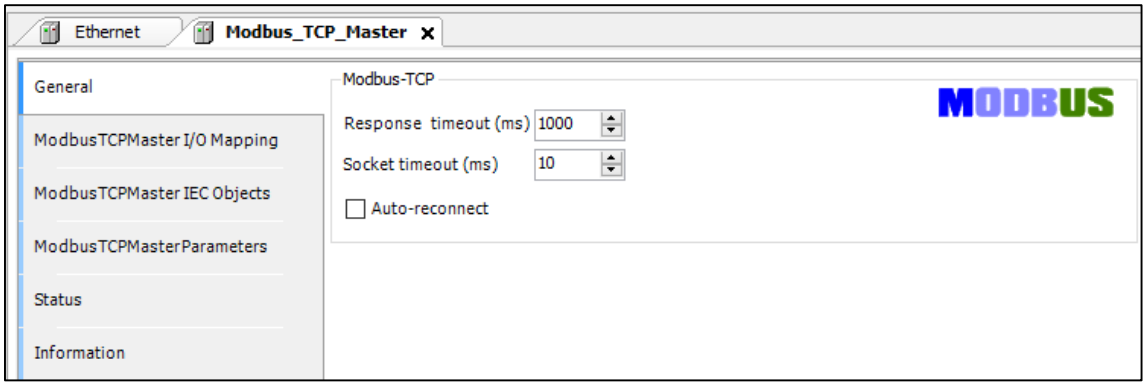
5.3.5.1 MODBUS TCP Master

The MODBUS TCP Master function is used to set the entry when the AX-8 series controller is used as the MODBUS TCP master station.

Right-click on the **Ethernet node** > **Add Device** > **MODBUS** > **MODBUS TCP Master** > **MODBUS TCP Master**.



General

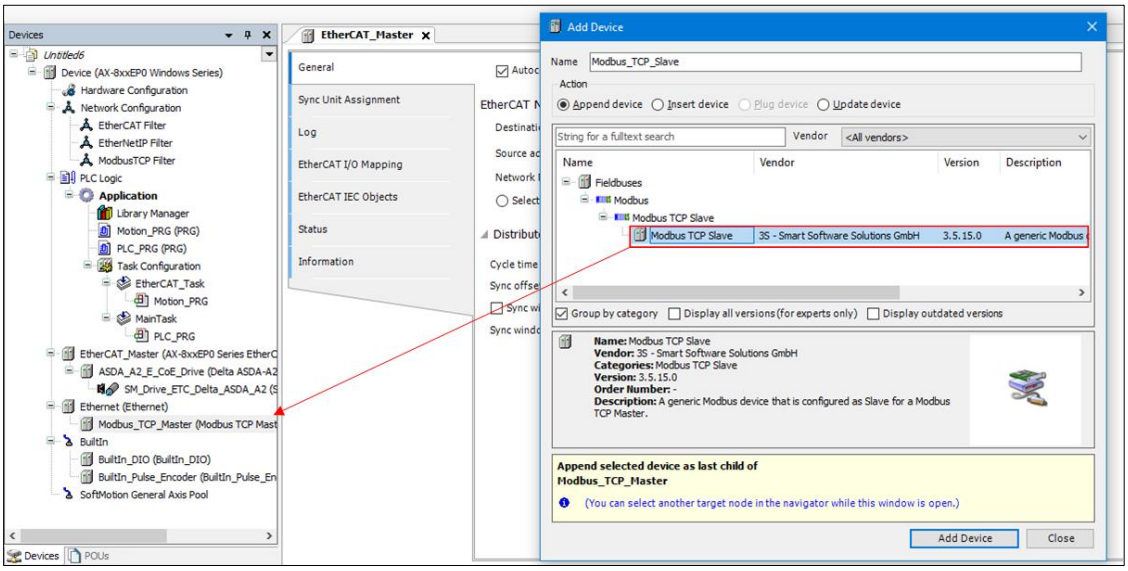


Item	Description
Response timeout (ms)	The response timeout sets the time interval that the master will wait for the answer from a slave node. If the nodes do not answer within this time interval, then an error is recorded for the implicit slave function block. The value entered for the time interval is also the default value for each node. For each node, user can still set a specific value within its MODBUS TCP slave configuration.
Socket timeout (ms)	Maximum time to wait for incoming TCP/IP packages. The bus cycle task can be blocked during this time, for example if a MODBUS TCP slave is disconnected.

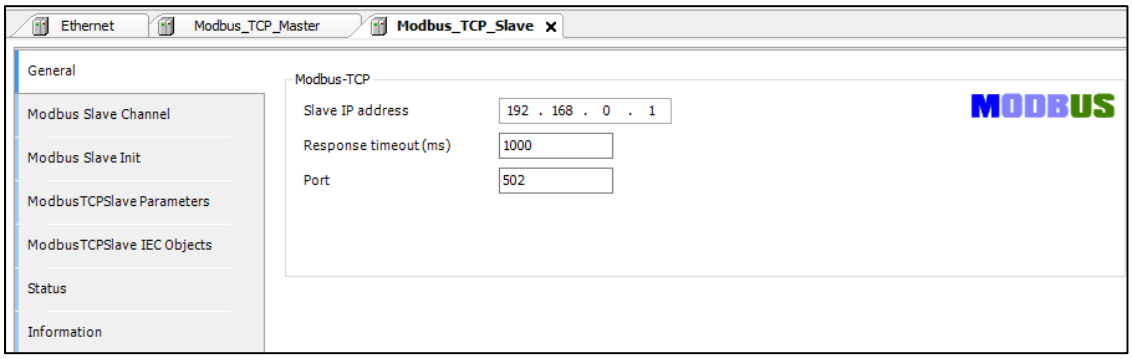
5.3.5.2 MODBUS TCP Slave

The MODBUS TCP Slave function is used to set the target slave station to be connected when the AX-8 series controller is used as the MODBUS TCP master station.

Right click on **Delta MODBUS TCP Master** > **Add Device** > **MODBUS** > **MODBUS TCP Slave** > **MODBUS TCP Slave**.



General

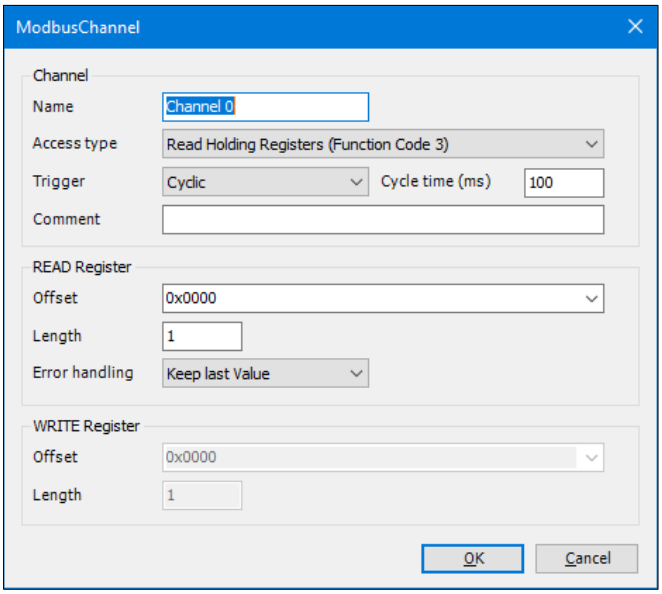


Item	Description
Slave IP address	IP Address
Response timeout (ms)	Time interval for the master to wait for the response from the slave. This is especially configured for this slave node and overwrites the general response timeout setting of the respective master.
Port	Port number (TCP/IP) of the slave 502 is default for MODBUS

- **MODBUS Slave Channel Tab**

  User will define the slave channel on this tab. Each channel represents a MODBUS request.

Item	Description
Add Channel	**MODBUS slave channel tab** > click **Add Channel**. Here user can add a new channel for the MODBUS slave. In the device description file of the MODBUS slave, user can predefine individual MODBUS registers or the description of the entire channel.



Channel	
Item	**Description**
Name	A string that contains the name of the channel
Access type	• Read coils (function code 1) • Read discrete inputs (function code 2) • Read holding registers (function code 3)

Channel	
	• Read input registers (function code 4) • Read single coil (function code 5) • Write single register (function code 6) • Write multiple coils (function code 15) • Write multiple registers (function code 16) • Read/Write multiple registers (function Code 23)
Trigger	• CYCLIC: The request occurs periodically. • Rising edge: The request occurs as a reaction to a rising edge of the Boolean trigger variables. The trigger variable is defined in the tab **I/O Mapping**. • Application: The MODBUS request is triggered by the PLC application. This happens by means of the MODBUS Channel function block, which is included in the respective I/O driver library. This function block provides complete control and information about the execution of this command, for example the start time, the processing time, and the result.
Cycle time (ms)	For Trigger = **CYCLIC**: Request interval ***Note****: The request interval should be the same as or a multiple of the cycle time of the application.*
Comment	Description of the channel

READ Register	
Item	**Description**
Offset	Start address where reading should start (value range 0 ~ 65535)
Length	Number of registers to be read (for word access) or number of discrete inputs to be read (for bit access)

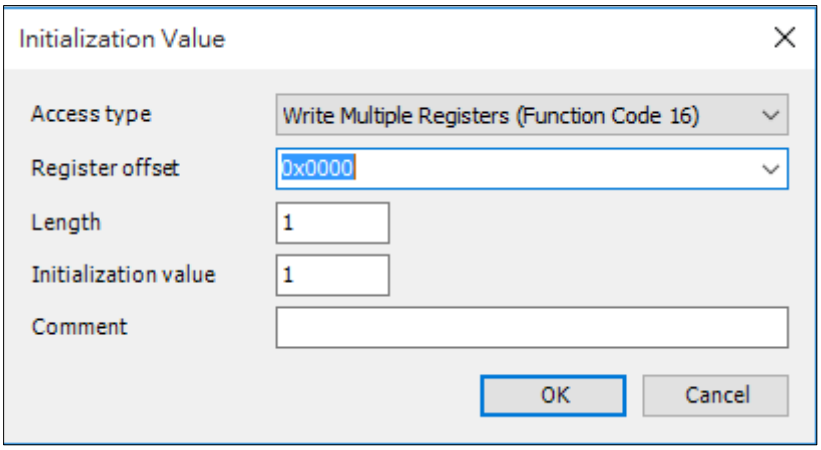
Error handling	Defines what should happen to the data in case of a communication error • Set to ZERO • Keep last value

WRITE Register	
Item	**Description**
Offset	Number of the register to be written to (value range 0 ~ 65535)
Length	Number of registers to be written to (= Words) The value range of the parameter depends on function code.

Init

User uses this tab to define initialization commands. Initialization commands are executed one time when starting the bus or activating the slave (setting the “Enabled” flag of the slave instance). When setting up or editing a slave initialization value, the following parameters are available in the respective dialog boxes:

Item	Description
Move Up Move Down	The order of channels also determines the order of initialization.
New	Opens the dialog box **Initialization Value**. The initialization commands are defined.

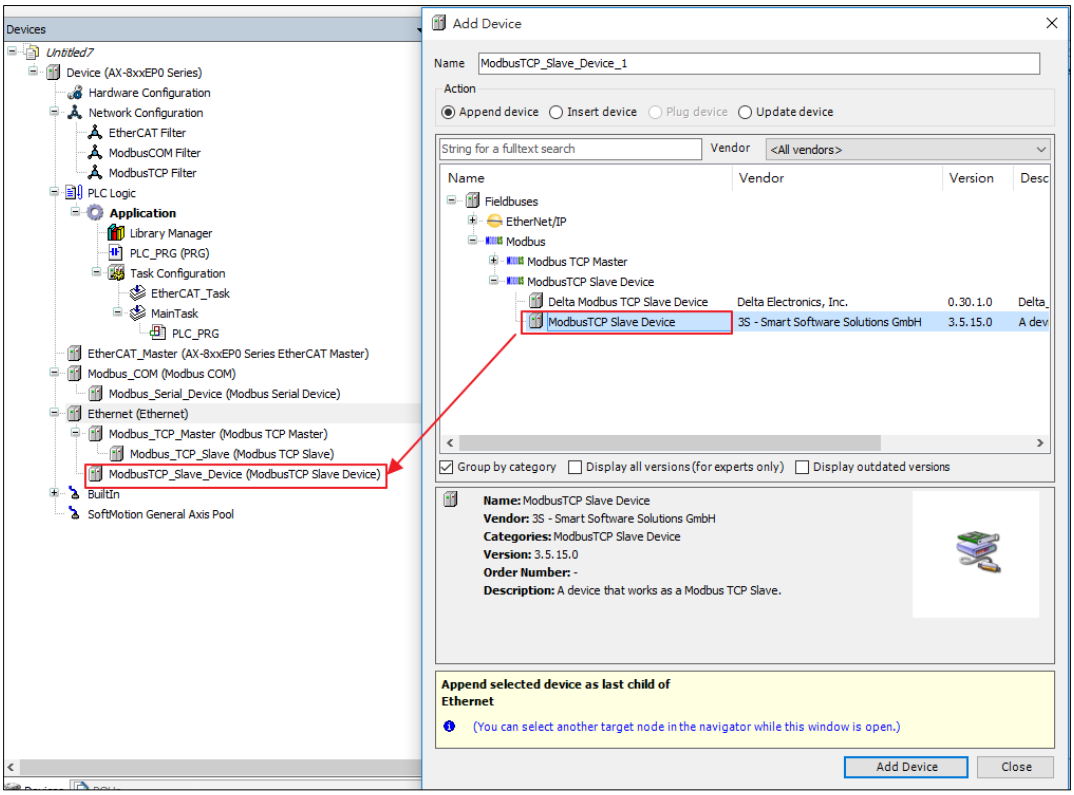


Item	Description
Access Type	• Write single coil (function code 5) • Write single register (function code 6) • Write multiple coils (function code 15) • Write multiple registers (function code 16)
Register offset	Number of the register to be written to (value range 0 ~ 65535)
Length	Number of registers to be written to (= Words) Value range of the parameter depends on function code
Initialization value	Initialization value for the register
Comment	Short description of the data

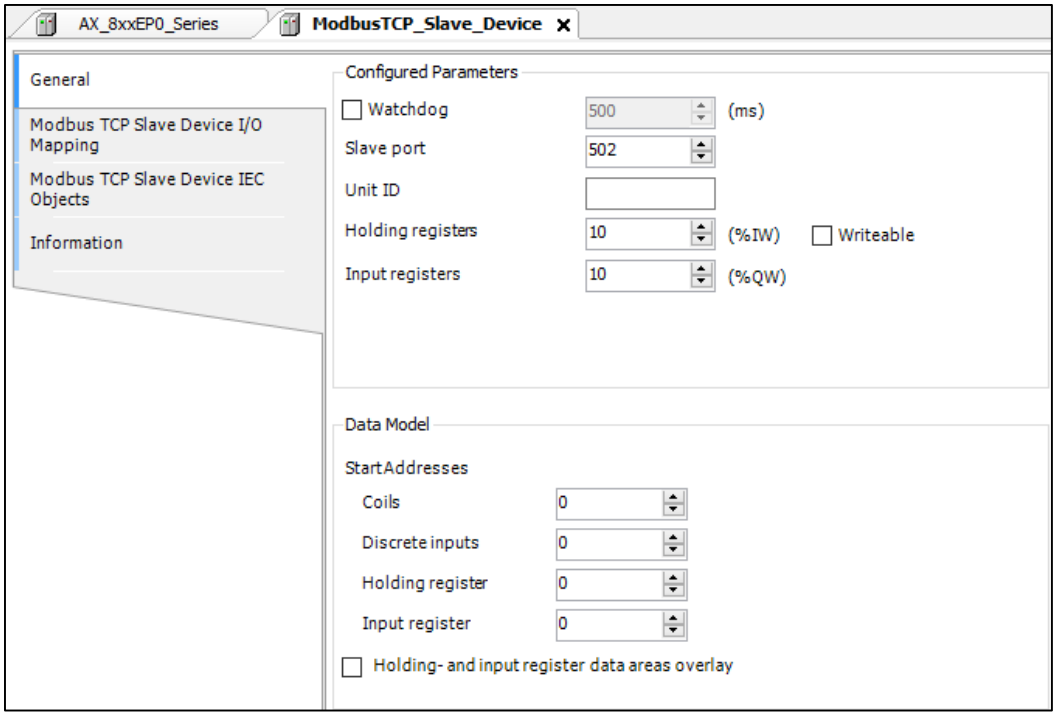
5.3.5.3 MODBUS TCP Slave Device

The MODBUS TCP Slave Device function is used to set the entry when the AX-8 series controller is used as a MODBUS TCP slave.

Right-click the **EtherNet** device and choose **Add device** to add **MODBUS TCP Slave Device**.



Double-click the newly added MODBUSTCP_Slave_Device, the following picture will be displayed:



General Tab

Item	Description
Watchdog	☑ Activates the watchdog function The incoming data (Holding Registers / %I range) is set to zero when the MODBUS device does not receive any valid query from the master.
Slave port	Port number of the slave
Unit ID	Optional: Unit ID of the slave. Only for a TCP/RTU gateway.
Holding registers (%IW)	Number of holding registers: possible values are 2 - 4096. The maximum number can be limited in the device description. Writable:☑ For the holding register, writable I/O mappings (%QW addresses) are generated instead of read-only I/O mappings (%IW addresses). This allows the holding registers to be set by the MODBUS device application (= server application) by means of the usual I/O mapping.

Item	Description
Input registers (%QW)	Number of input registers: possible values are 2 - 4096. The maximum number can be limited in the device description.
IP Master address	IP address of the master; Example: “192.168.0.1”. The device description determines whether the option is displayed in the dialog.
I/O Mapping active	☑: Activates the I/O scanner. The device description determines whether the option is displayed in the dialog.

MODBUS TCP defines for I/O data a data model with four areas (starting address).

- Coils: Outputs of type BIT
- Discrete Inputs: Inputs of type BYTE
- Holding Register: Outputs of type WORD (16-bit)
- Input Register: Inputs of type WORD (16-bit)

Inputs and outputs are included as with all fieldbuses, but from the point of view of the host. These data areas have any number of logical start offsets as defined by the manufacturer. The I/O data behind the MODBUS TCP data areas can overlap or be independent of each other.

5.3.6 EtherNet/IP Settings

EtherNet/IP based on standard TCP and UDP allows communication between office networks and control systems. The EtherNet/IP target device supports DHCP and BootP to assign IP addresses. Start the EtherNet/IP network (diagnosis), user can use the Web server integrated with the logic control interface module, or user can use the Web server of other EtherNet/IP devices.

Like other standard Ethernet-based networks, the EtherNet/IP protocol is not suitable for real-time applications (< 1 millisecond), such as servo system, because the typical cycle of EtherNet/IP network is 10 milliseconds.

Install and add EtherNet/IP devices

In order to add and configure EthernNet/IP devices in the project tree, user need to install related device description files (EthernNet/IP device description files (*.eds)) in **Tools** > **Device Repository**.

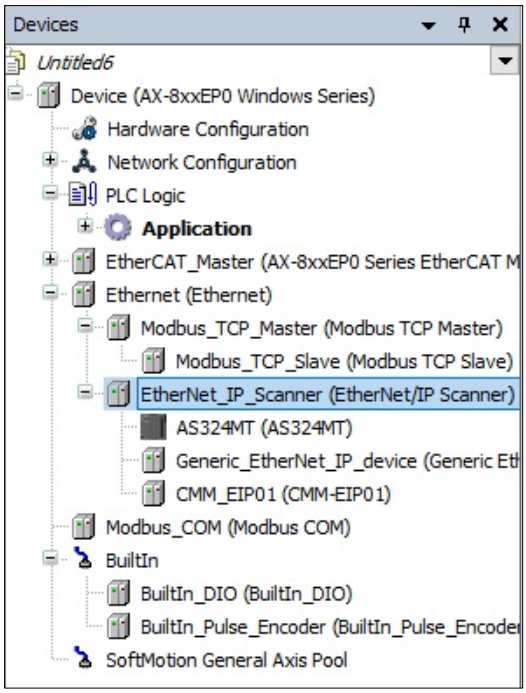
Adding an EtherNet/IP scanner to the project tree will add the specific library collection of the related device to the library manager. There are two ways to insert one or more EtherNet/IP remote adapters or devices into the scanner.

- **Module configuration**

  Under one target device is another target device, called the **chassis**, which can be inserted into a fixed number of slots. Use the **insert device** to... command to insert the so-called **IO point** into the slot, thereby controlling the input and output.

- **Non-modular configuration**

  The target device is equipped with input and output.



Double-click an EtherNet/IP object in the project tree, or select a device in the project tree and call the edit object command in the menu to open the device editor for configuring EtherNet/IP devices. (The settings of the editor itself, such as displaying the general configuration dialog box, are implemented in the device editor options.

The title bar of the EtherNet/IP configuration dialog box is marked with the name of the specified EtherNet device; depending on the device, the label included in the dialog box will be different. Note that the bus cycle time is adjusted in the PLC settings.

- EtherNet/IP Scanner
- EtherNet/IP Remote Adapter

Access EtherNet/IP configuration through the app

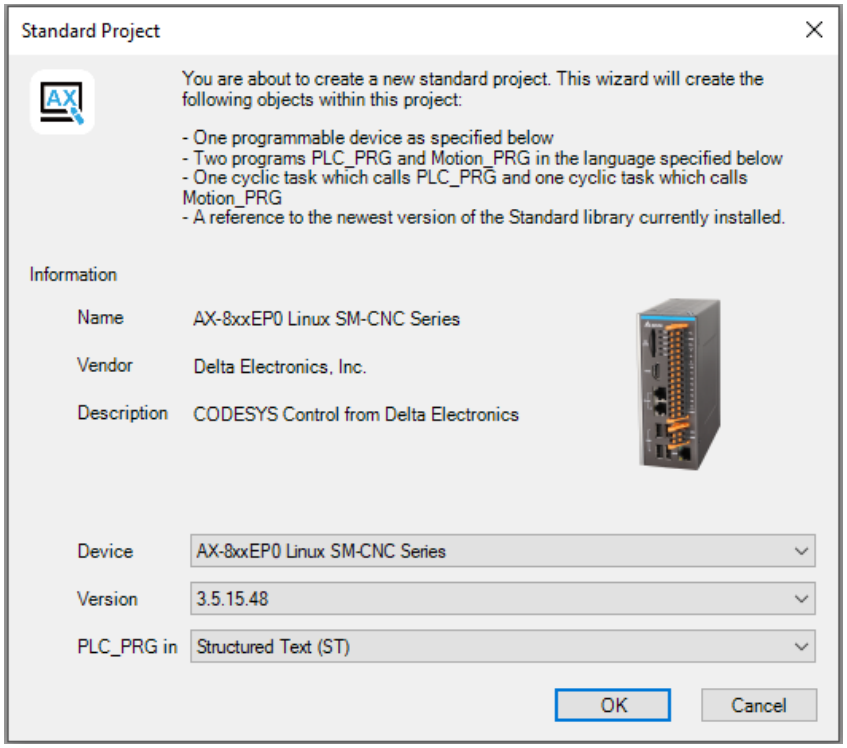
Each EtherNet/IP scanner will automatically add a related function block instance to the project tree. The name of this variable and the type of the function block will appear in the EtherNetIP scan I/O map of the IEC object tab. Variables can be configured through the project address, such as allowing start, stop or application scan status check.

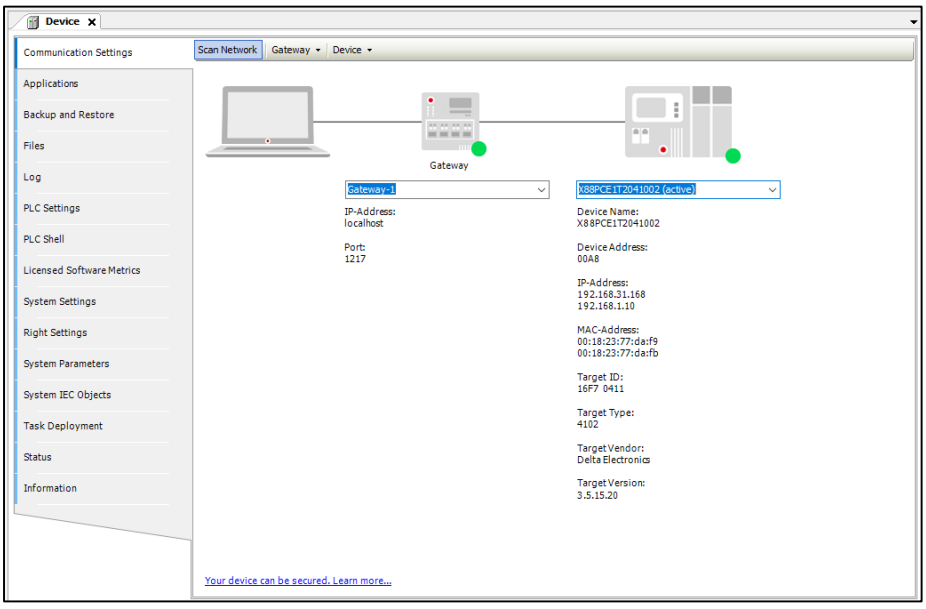
5.3.7 EtherNet/IP Scanner

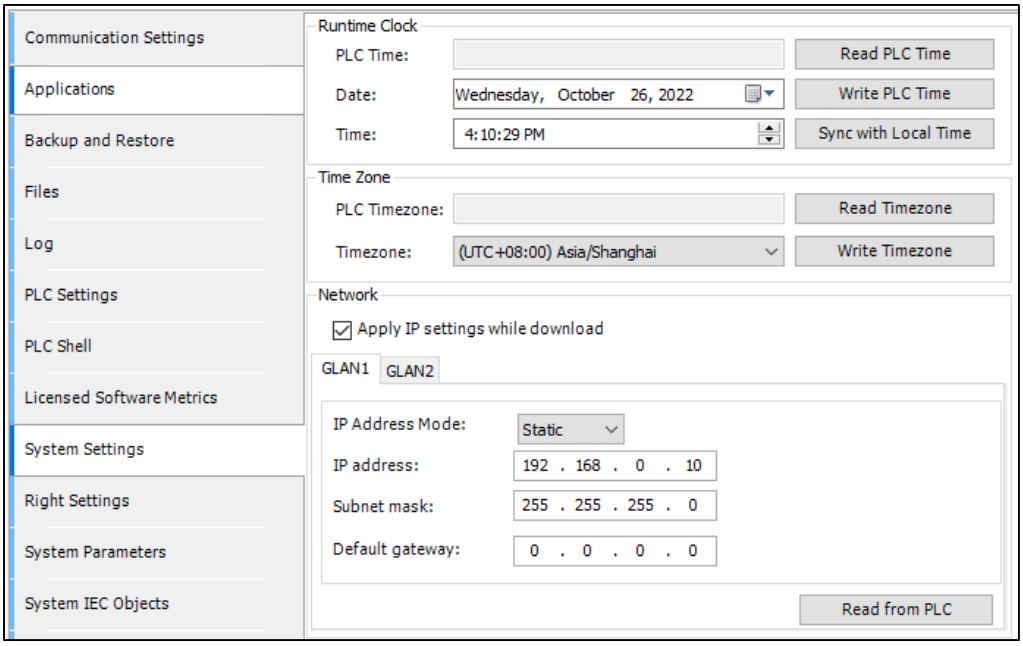
Applicable models: AX-8 Linux series models

Take AX-8xxEP0 Linux SM-CNC Series as an example:

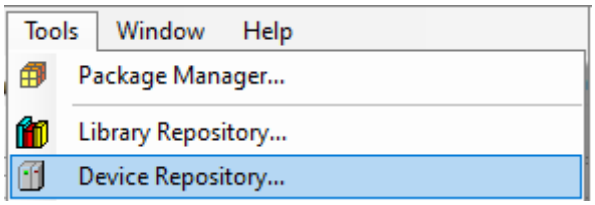
1. Create a project > **Scan Network** > Go to **System Settings** > **Read from PLC**.

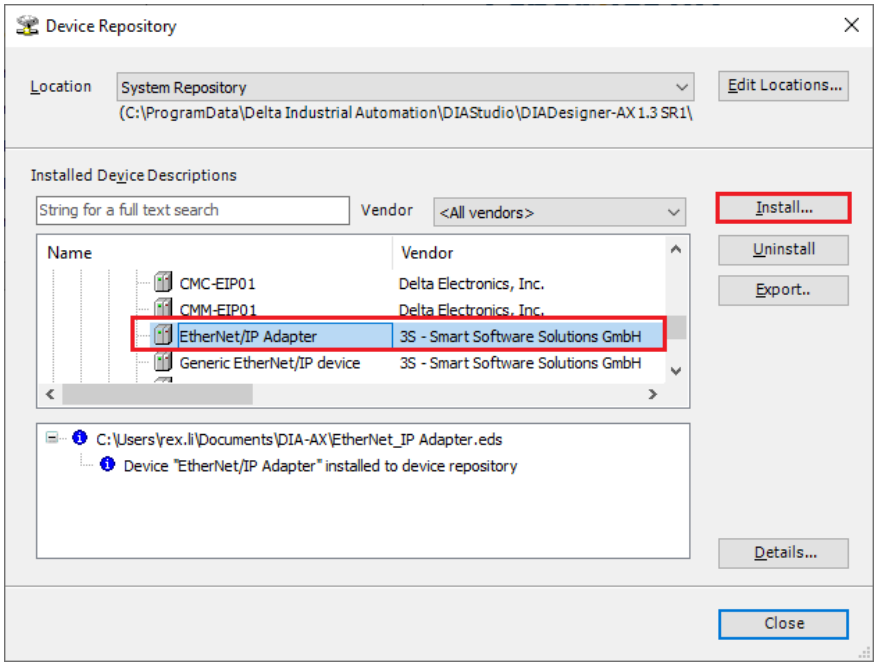




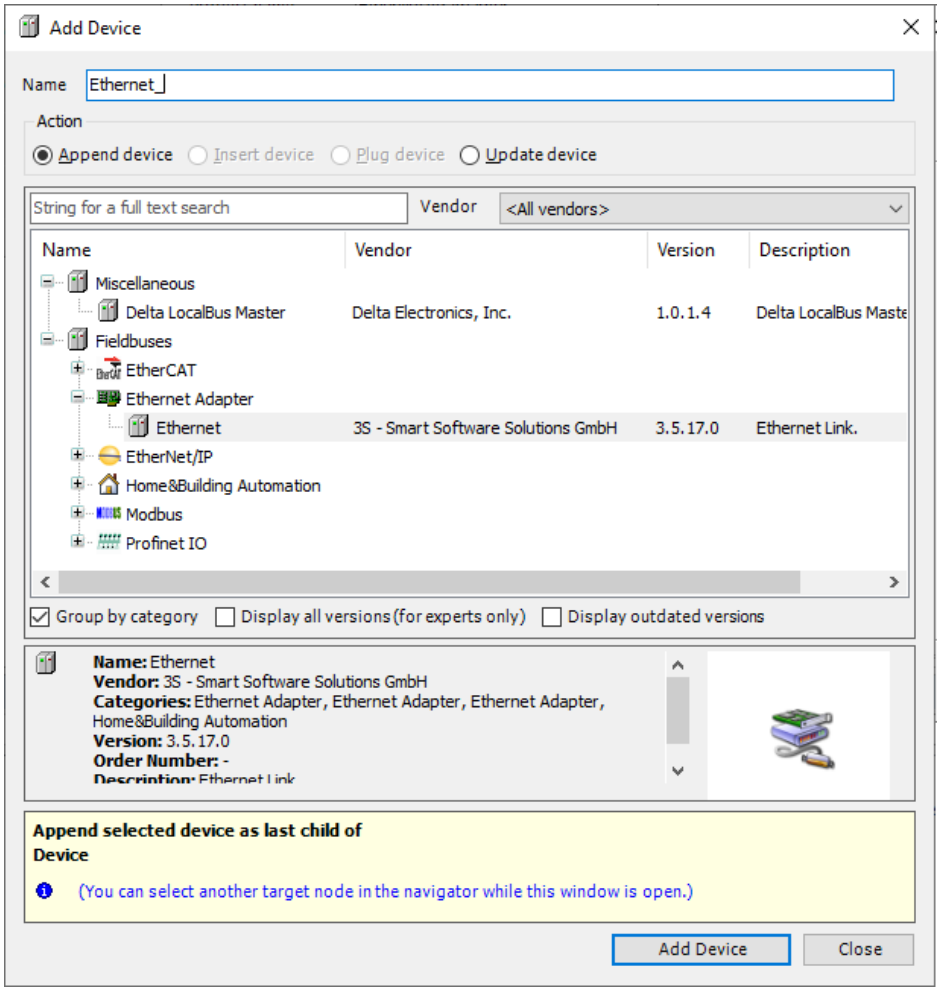


2. Go to **Tools** > **Device Repository** > **Install** EDS File.

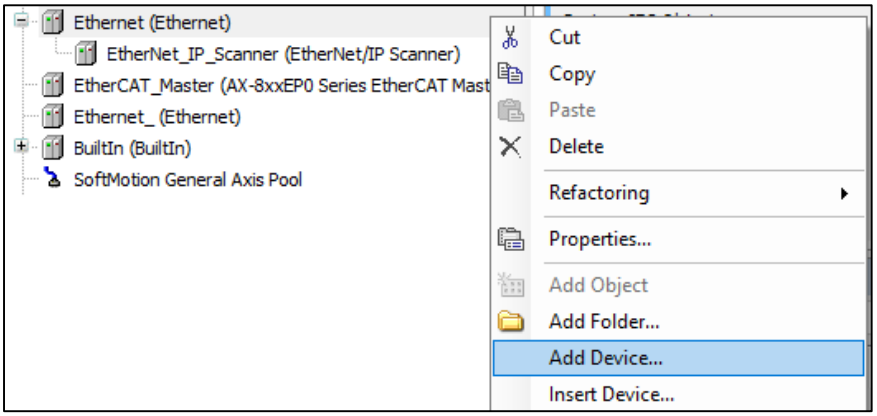




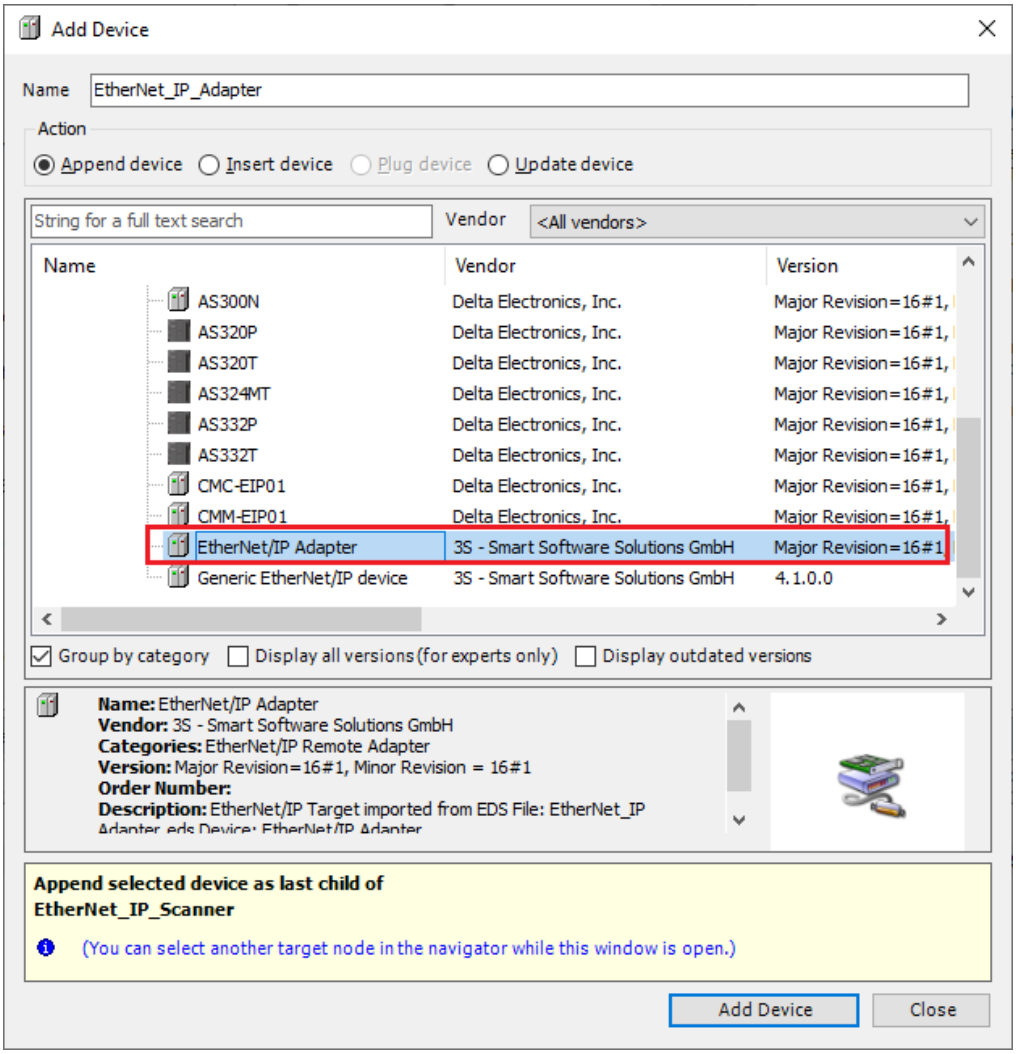
3. Right click **Device (AX-8xxEP0 Linux SM-CNC Series)** > **Add Device** > select **Ethernet Adapter** > **Ethernet**.



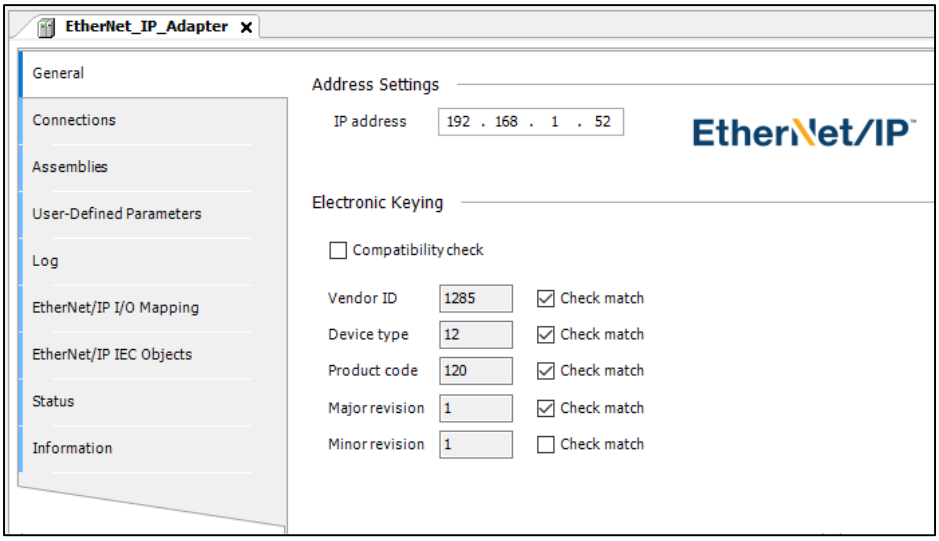
4. Right click **Ethernet** > **Add Device** > select **EtherNet/IP Scanner**.



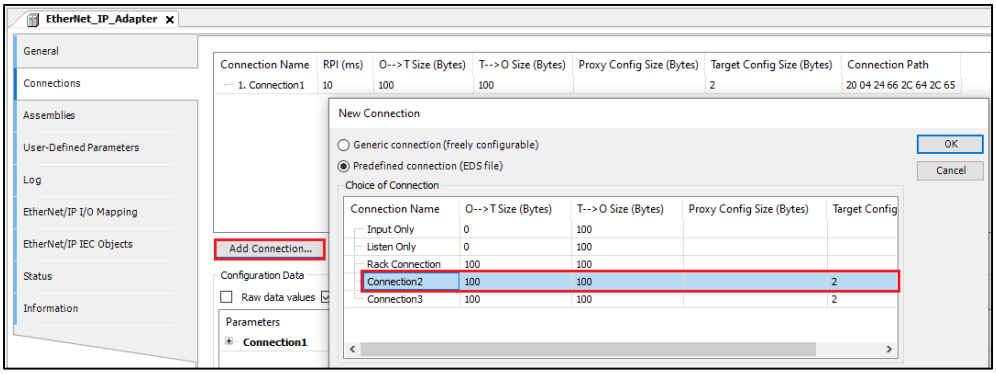
5. Right-click **EtherNet/IP Scanner** > **Add Device** > **EtherNet/IP Adapter**, and call the Adapter's EDS file for use.



6. Confirm that the IP address must match the target (external) device.

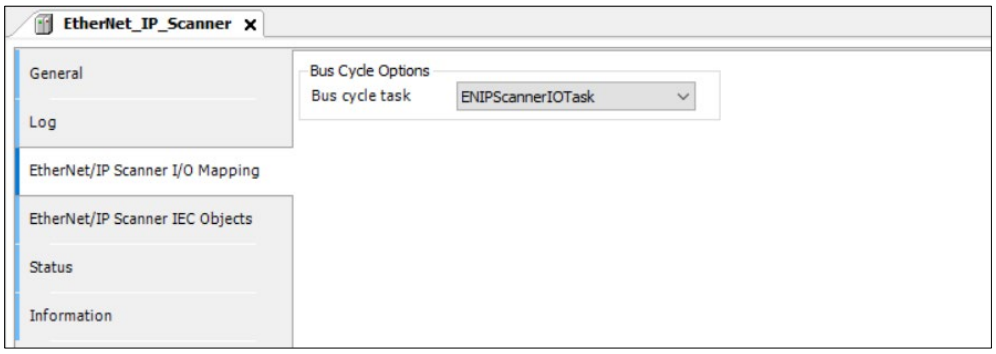


7. To Connections, a second connection in the EDS file can be added via **Add Connection**.



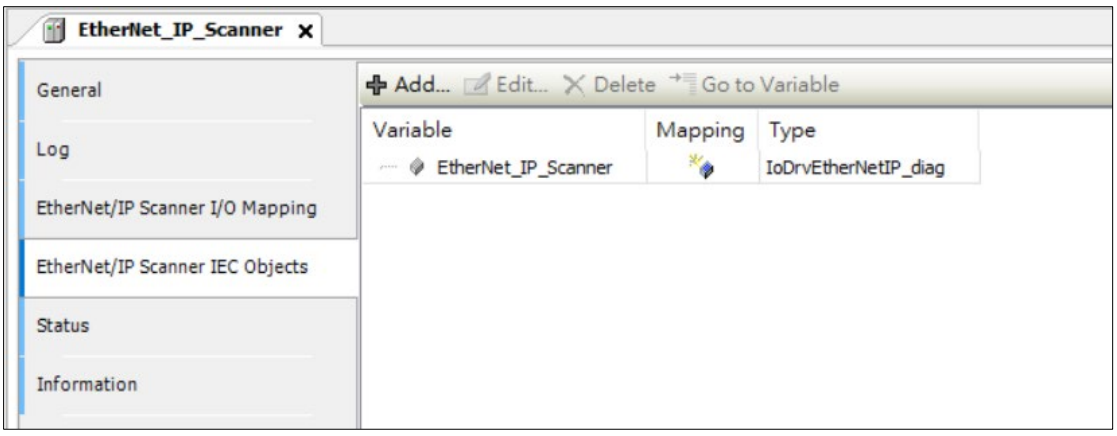
8. Click **Login** to finish.

EtherNet/IP Scanner I/O Mapping Tab



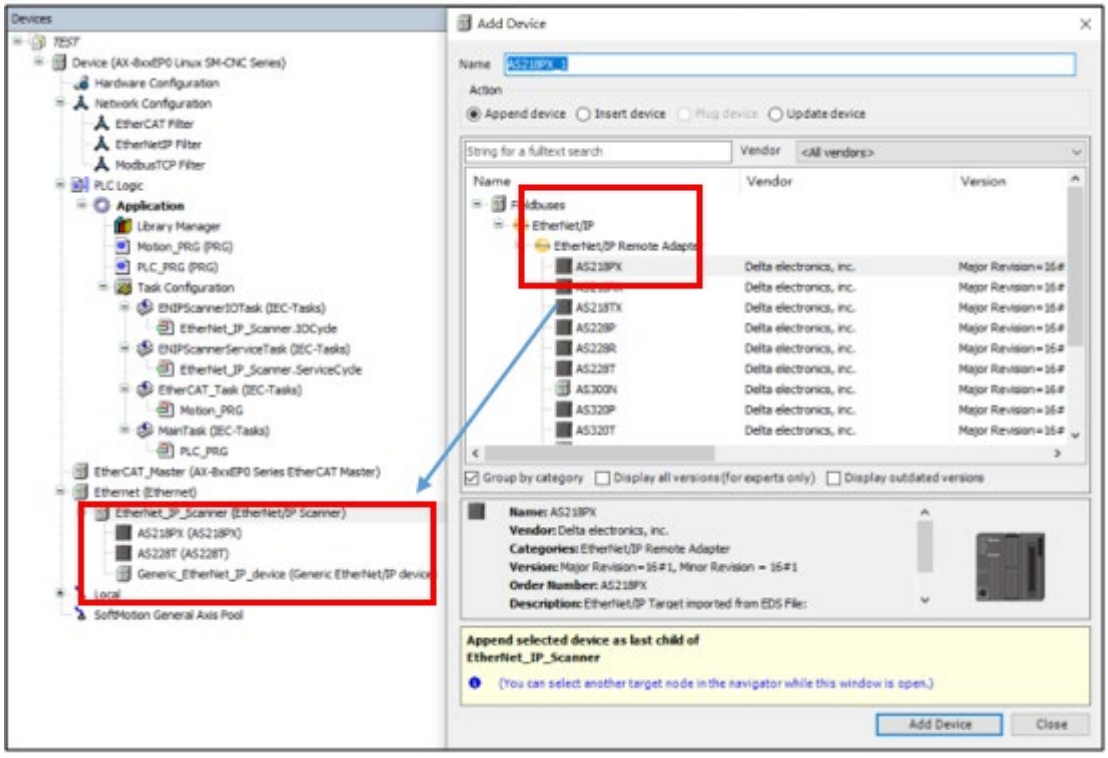
Project	Description
Bus cycle task	Select bus cycle task. For more information about bus cycle tasks, please refer to AX-8 Operation Manual: PLC Settings.

This tab displays the definition status of EtherNet/IP devices. User can know the running status from the status.



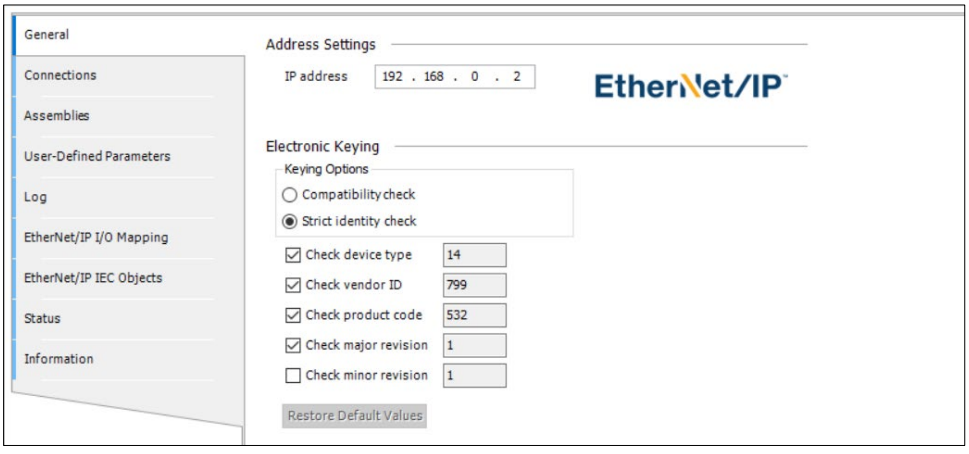
5.3.7.1 EtherNet/IP Remote Adapter

AX-8xxEP0 Supports EtherNet/IP Remote Adapter function. Right-click on the **EtherNetIP Scanner node** > **Add Device** > **Fieldbus** > **EtherNetIP** > **EtherNet/IP Remote Adapter**.



If user want to use a third-party EtherNet/IP remote adapter, please go to **Tools** > **Device Library** > **Import the EtherNet/IP standard EDS file**, and then add it in the **Add Device window**.

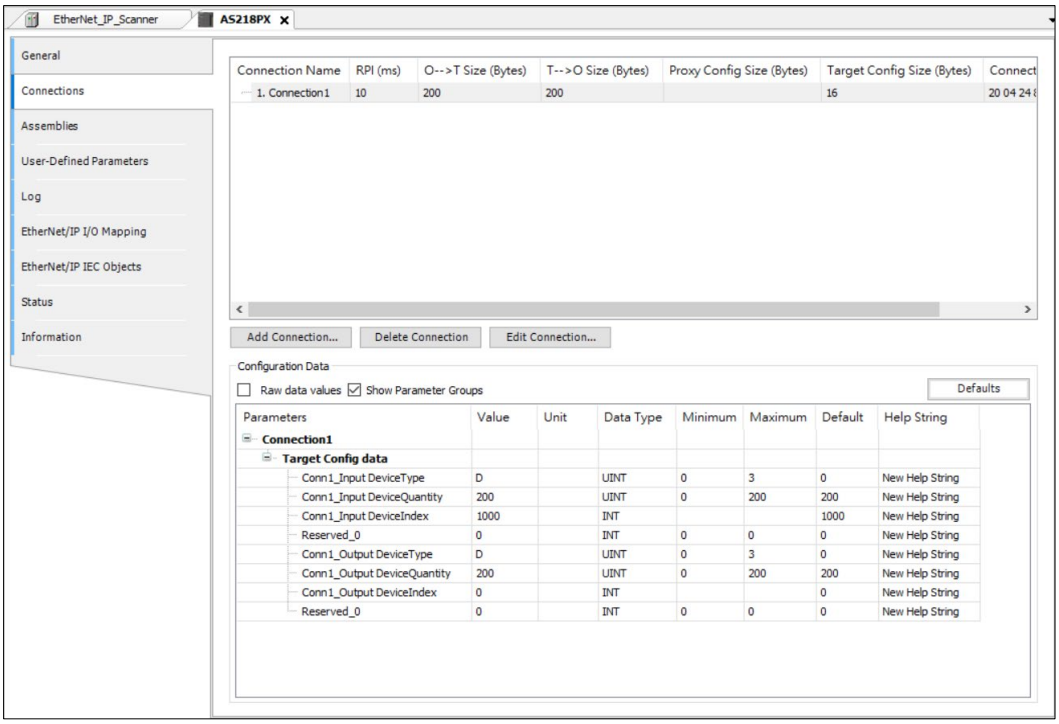
EtherNet/IP Remote Adapter General Tab



Project	Description
Address Settings-IP address	EtherNet/IP IP location of the remote adapter
Electronic keying	
Keying options-compatibility check	Start: The adapter performs a compatibility check on its EDS file. All key values will be sent to the device. The device decides whether it is compatible with the received value.
Keying options-Strict identity check	The adapter performs a compatibility check on its own EDS file. The user decides which keying information should be checked. If the check fails, no I/O connection with the device will be established, and an error message will be posted on the status page.
Check device type	Check device type
Check vendor ID	Check supplier code
Check product code	Check product code
Check major revision	Check major revisions
Check minor revision	Check minor revisions
Restore Default Values	Restore preset values for general equipment

Connection tab

The top of this setting screen displays a list of all configured connections. If there is an exclusive owner connection in the EDS file, it will be automatically inserted when adding a device. The configuration data of these connections can be modified at the bottom of the screen.

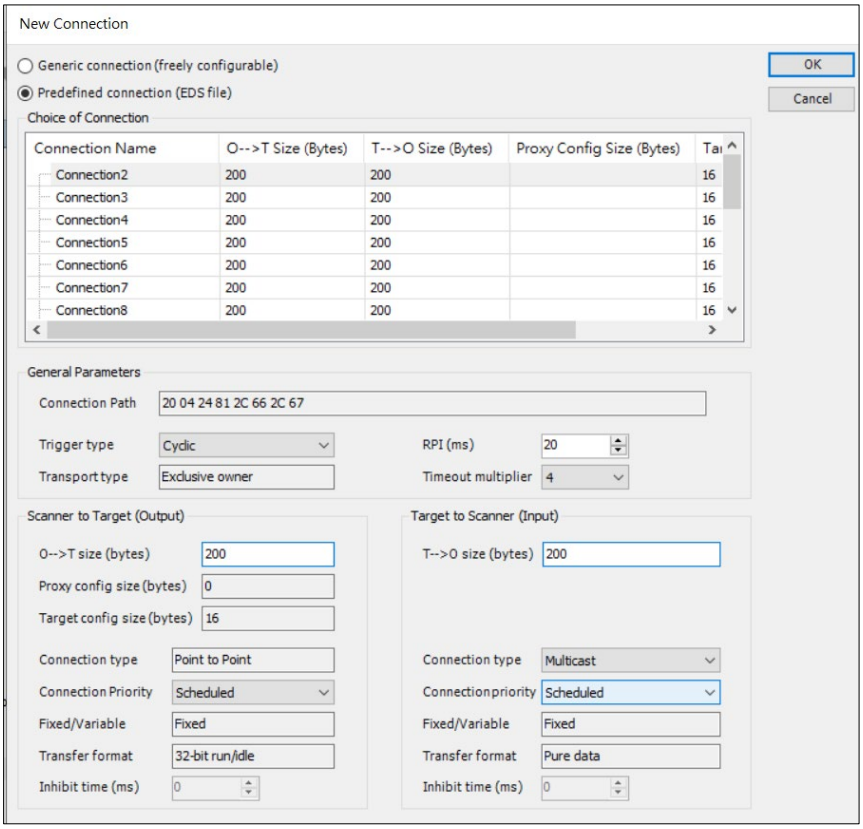


The configurable data will be defined in the EDS file and transmitted when the connection with the adapter is opened.

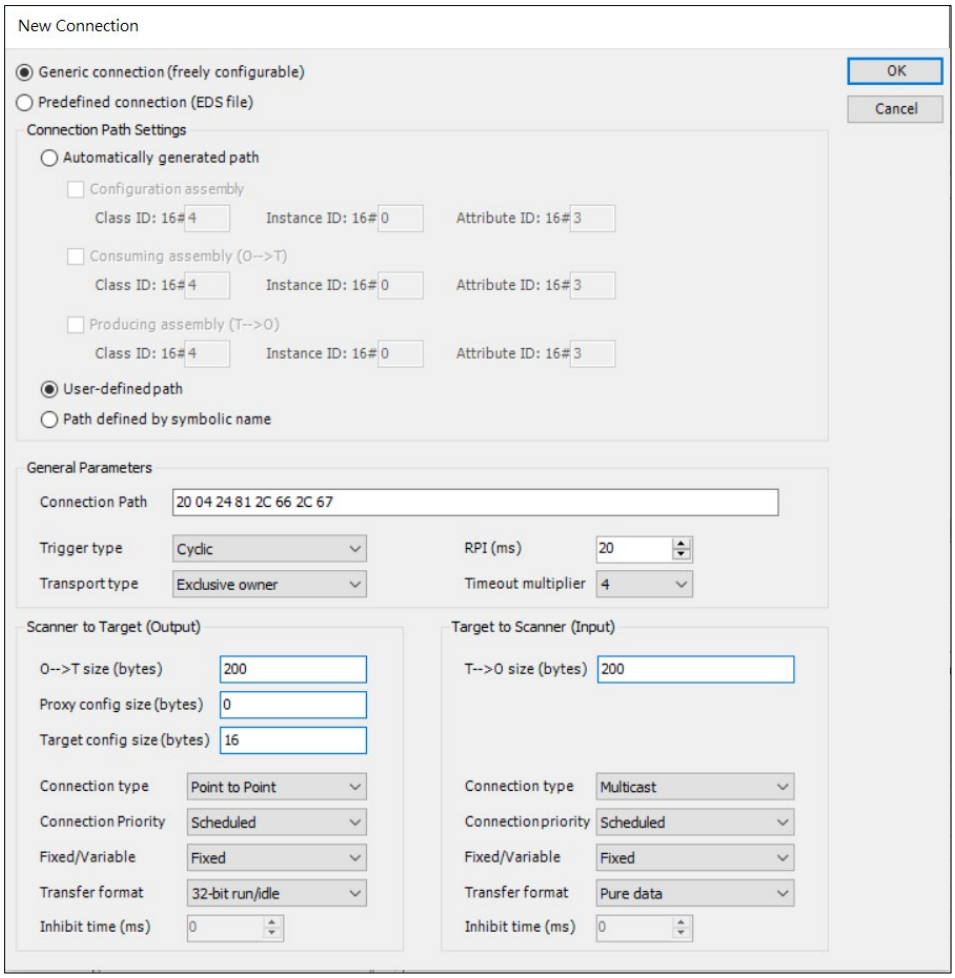
Project	Description
Connection name	Connection name
RPI (ms)	Requested packet interval: the exchange interval of input/output data
O → T size (Bytes)	Connection The size of producer data from scanner to adapter （Originator → Target）
T → O size (Bytes)	Consumer data size from adapter to scanner （Target → Originator）
Proxy Config size (Bytes)	The size of the proxy configuration data
Target Config size (Bytes)	Target configuration data size

Project	Description
Connection path	The connection path data is represented as: Address-configuration object-input object-output object.
Add Connection	Open the **New Connection** dialog box. Determine the parameters of the new connection here.
Delete Connection	Remove the selected connection from the list
Edit Connection	Open the Edit Connection dialog box. The parameters of the existing connection are modified here.

Click **Add Connection**, the following dialog box will pop up, which contains the parameters of the new connection.



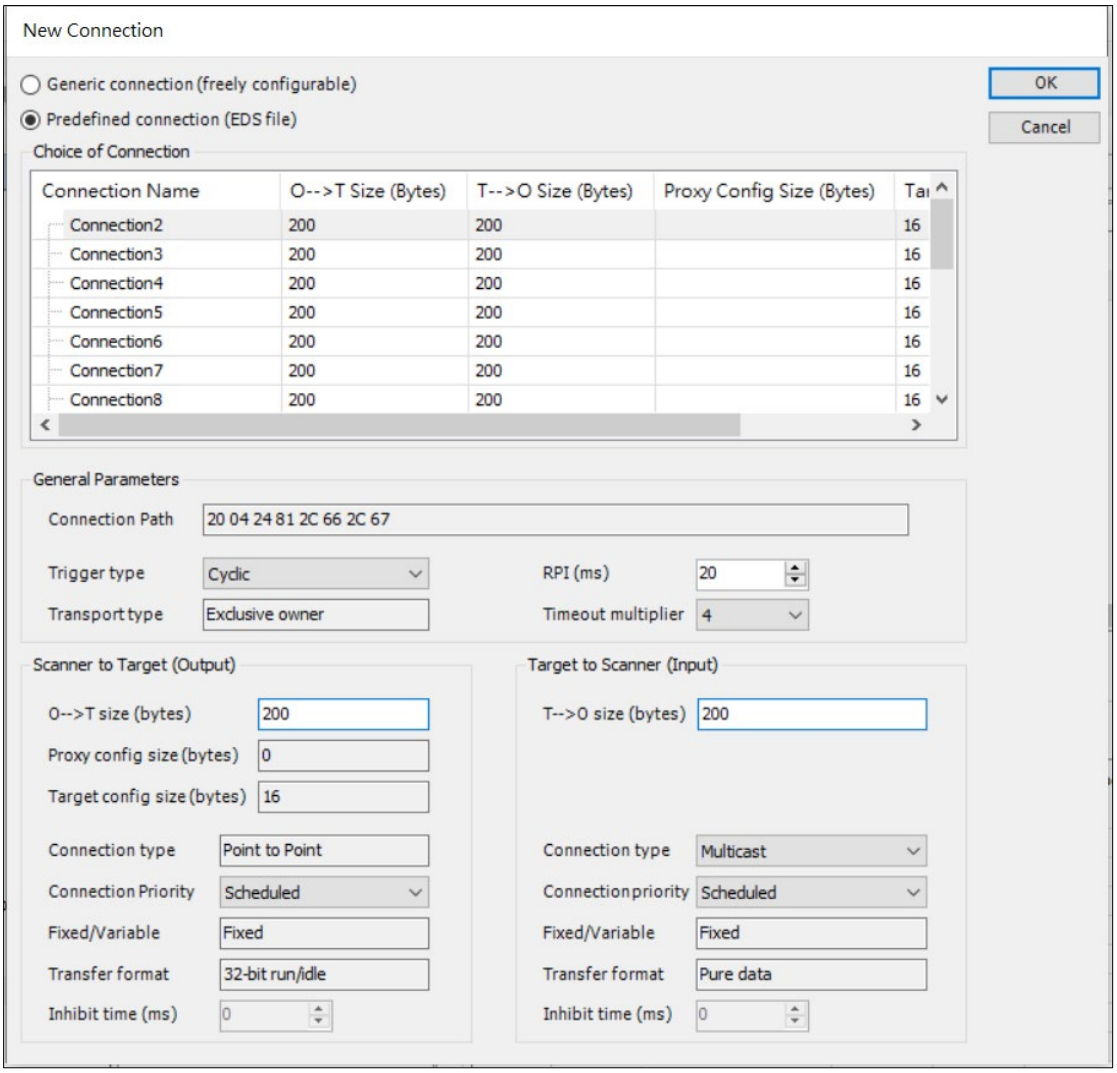
- **Generic Connection (freely configurable)**: The dialog contains the parameters of the new connection.



Project	Description
Connection Path Settings	
Automatically generated path	The connection path is automatically generated based on the combined configuration, combined consumption and combined production values.
User-defined path	The connection path is manually specified in the corresponding input field.
Path defined by symbolic name	The path is specified by the symbolic name. ***Note:*** *Connection path setting must support symbolic connection path.*
General parameters	

Project	Description
Connection Path	The connection path is used to address one or more objects in the adapter that provide input data and receive output and configuration data. ***Note**: The connection path is set to a custom path.*
Path defined by symbolic name	Use ANSI strings instead of ordinary connection paths. For the allowed ANSI strings, please refer to the corresponding EtherNet / IP adapter manual. ***Note**: The connection path is set as the connection label.*
Trigger type	• Cyclic: Periodically exchange data at intervals set by RPI. • Status change: After changing the scanner output or adapter input, data will be exchanged automatically. • *Application: Not implemented.*
Transport type	For details, please refer to the CIP specification.
RPI (ms)	The time interval (in milliseconds) at which the transmitting application requests data to be transmitted to the target application. The value must be a multiple of the bus cycle task.
Timeout multiplier	If the device fails, there will be a time delay before the device state switches to "error" (RPI * timeout multiplier).

- Pre-defined connection (EDS file): Use this option to use the existing connection in the EDS file, and the user can change the configuration data defined in the EDS file.

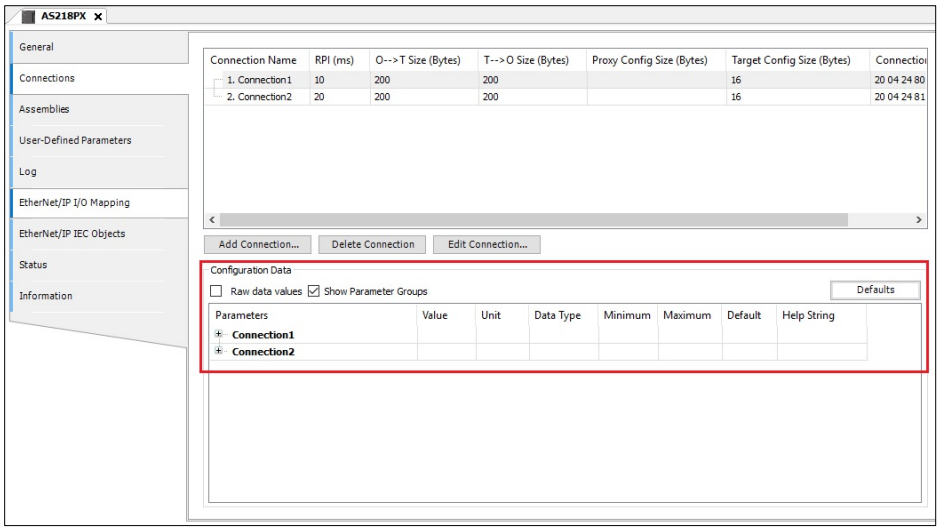


Project	Description
Scanner to Target (Output)	
O →T Size (bytes)	The size of producer data from scanner to adapter（Originator → Target）
Proxy config size (bytes)	The size of the proxy configuration data.
Target config size (Bytes)	Target configuration data size
Connection type	• Empty: no network connection is established. • Multicast: The network connection has been established. Connection data can be received by multiple user.

Project	Description
	• Point-to-point: A network connection has been established. Connection data can only be received by one user.
Connection Priority	Using two scanners with different priorities for a target may cause conflicts. Adjusting the connection priority can solve this problem.
Fixed/Variable	For detailed information on parameters, see CIP specifications
Transfer format	Conversion format
Inhibit time	Inhibit time
Timeout multiplier	Extend the time interval for the scanner to send heartbeat messages to the adapter. This value is multiplied by the RPI value. Example: RPI = 10ms, and heartbeat multiplier = 10 causes a message to be sent every 100ms. **Note:** The transmission format is Heartbeat
Target to Scanner (Input)	
T →O size (bytes)	Consumer data size from adapter to scanner （Target –> Originator）
Connection type	• Empty: No network connection established. • Multicast: A network connection has been established. Connection data can be received by multiple user. • Point-to-point: A network connection has been established. Connection data can only be received by one user.

Project	Description
Connection Priority	Using two scanners with different priorities for a target may cause conflicts. Adjusting the connection priority can solve this problem.
Fixed/Variable	Fixed/variable
Tranfer format	Conversion format
Inhibit time	Inhibit time

- **Connections Tab - Configuration Data**



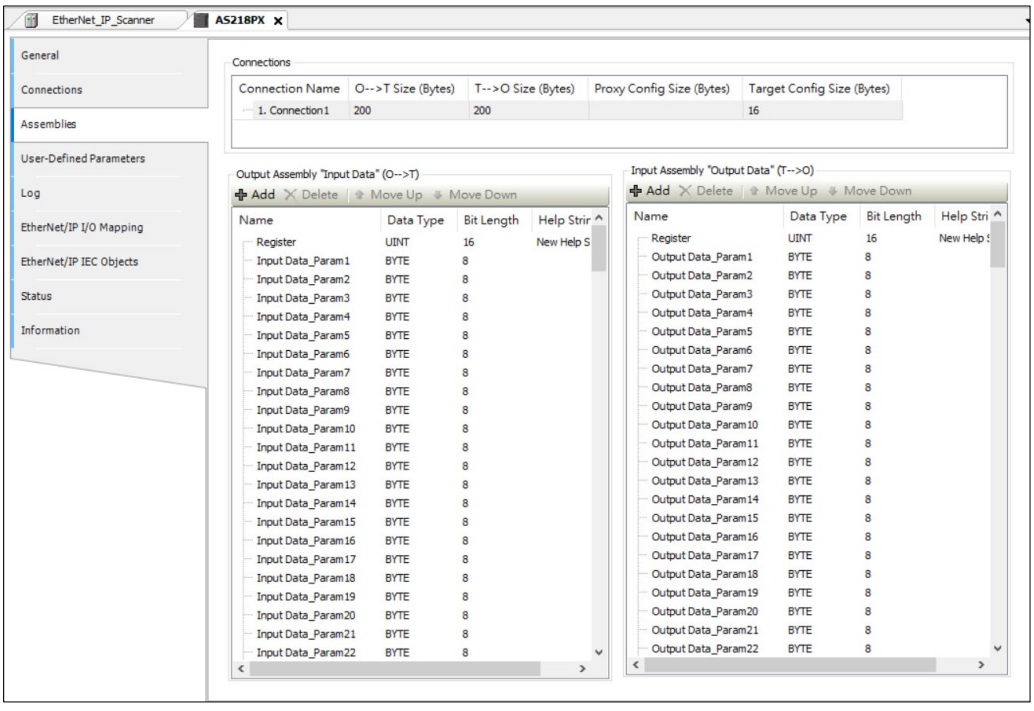
The following table shows connections with configuration parameters in the EDS file. Connections are subdivided into configuration groups.

Project	Description
Raw data value	• If scaling parameters are defined for the data in the EDS file, the value can be displayed as raw data or converted data. • Startup: Display data without any conversion. For the Enum data type, the index of the enumeration value will be displayed.

Project	Description
	• Not started: Display data and convert. For the Enum data type, the enumeration value will be displayed.
Display parameter group	Display parameter group
default	Set as default
value	Double-click to change the value. According to the data type, user can specify the value directly in the input field, or user can select from the drop-down list.

Component Tab

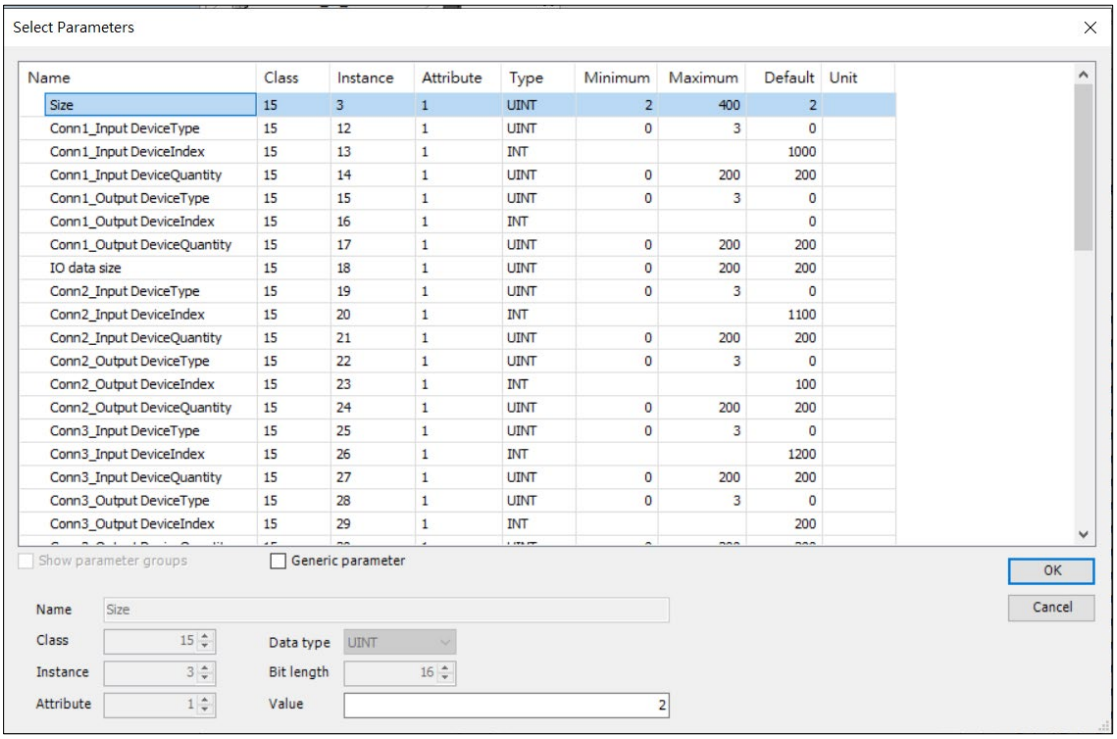
The upper half of the screen on this tab displays a list of all configured connections. After selecting the connection, the relevant data configuration will be displayed in the lower part of the screen.



Output component Input Data / Input component Output Data

Project	Description
Add	Open the select parameter dialog box, select the parameter to be added.
Delete	Delete selected parameter
Move Up/ Move Down	Move the selected parameter up/down in the list, the order in the list determines the order in the I/O mapping.
Name/Data type/ Bit Length/Help String	These values can be changed by double-clicking on the text field.

Add > Select Parameter window

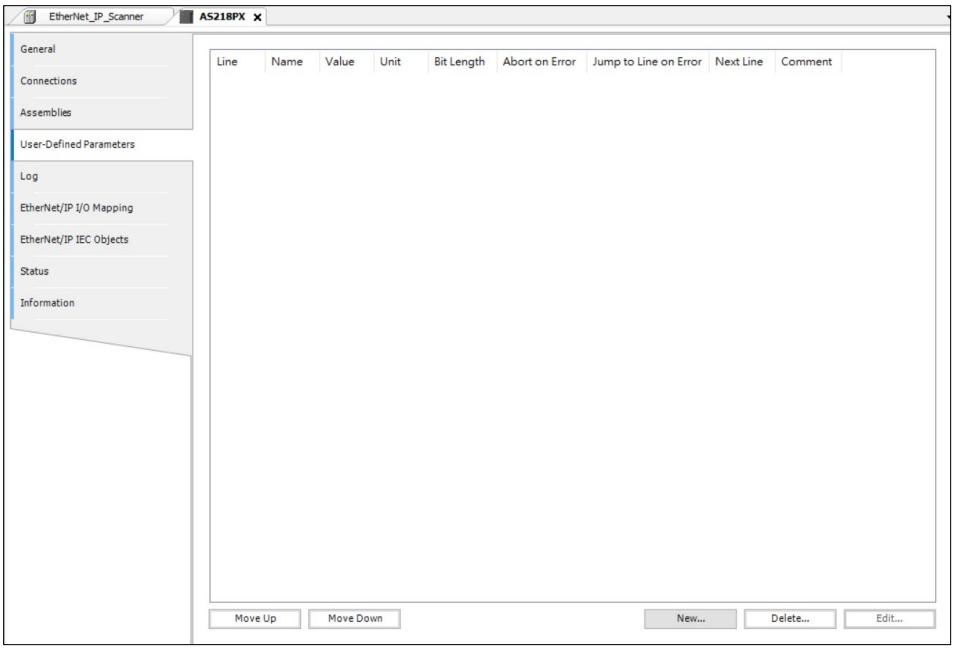


Project	Description
Display parameter group	• Start: This dialog box displays all parameters in the EDS file according to the group. • Not activated: This dialog box displays all parameters in the EDS file according to the structure

Project	Description
	By clicking **OK**, user can select each parameter in this list and add it to the parts list.
General parameters	User can add common parameters and edit the values of the parameters.

User parameter tab

This tab shows all other parameters that are only transferred to the bus system during the phase of the startup process assigned to it. When the connection is re-established (for example, after the remote adapter fails), the user parameters will be sent again.



Project	Description
New	Open the **Select Parameters** dialog to add new parameters. The new parameter will be inserted in front of the selected row.
Delete	Select the parameter to be deleted to delete
Edit	Open the **Select Parameters** dialog to change existing parameters.

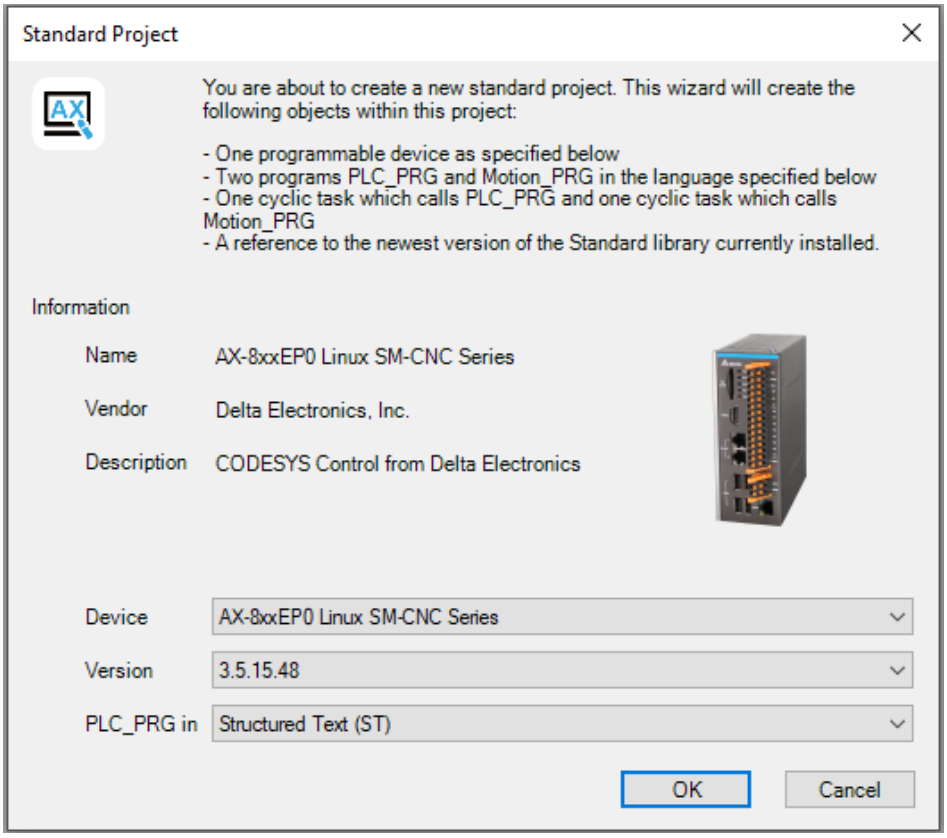
Project	Description
Move Up/ Move Down	Change the order of user parameters. The order of the parameters in the list corresponds to the order of initialization.
Value	User can directly change the value of the corresponding parameter by double-clicking the value.
Abort an Error	Start: When an error occurs, the complete transmission of parameters is aborted.
Jump to Line on Error	When an error occurs, the program will continue from the line specified in the **Next Line** column. Therefore, complete blocks can be skipped during initialization, or return can be defined. **Note:** *If user can never write a specific parameter, returning will result in an infinite loop.*

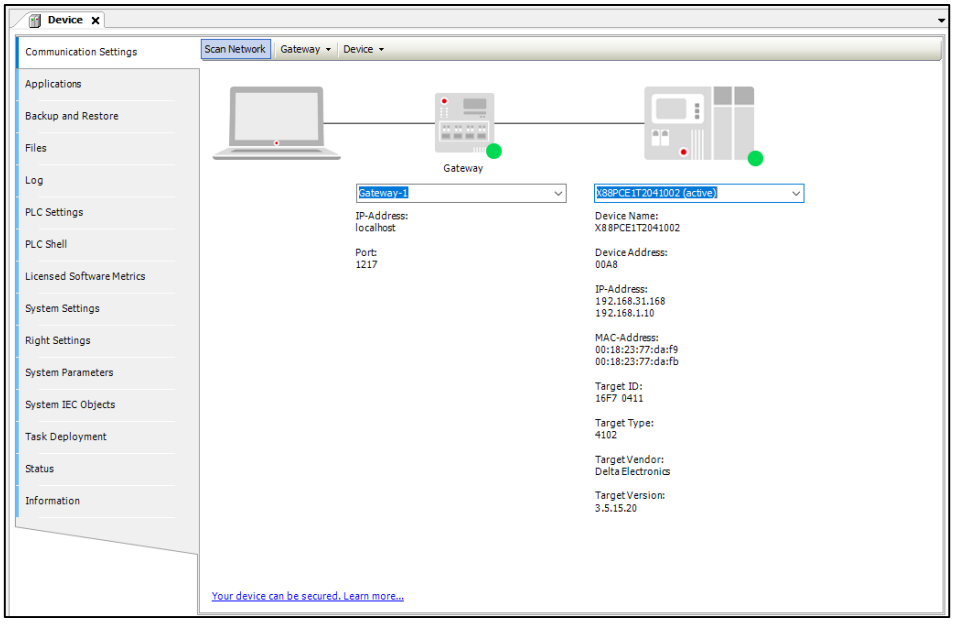
5.3.7.2 EtherNet/IP Local Adapter

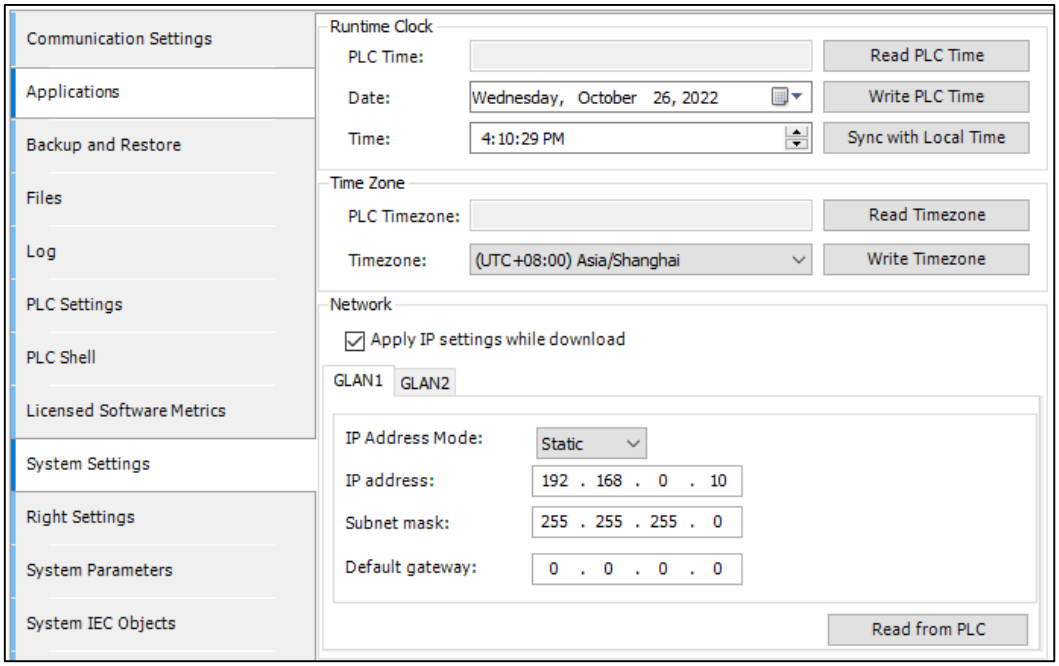
Applicable models: AX-8 Linux series, supports up to 10 scanners

Take AX-8xxEP0 Linux SM-CNC Series as an example:

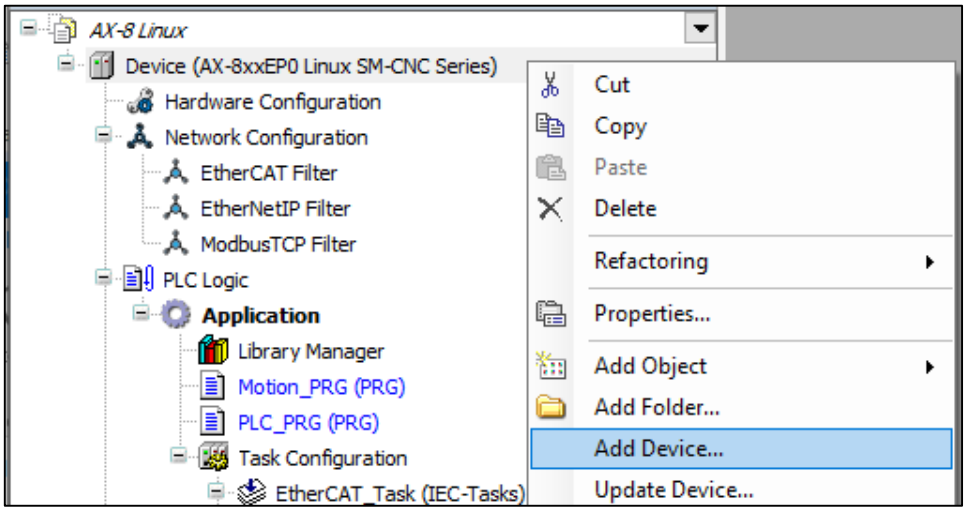
1. Create a project > **Scan Network** > Go to **System Settings** > **Read from PLC.**

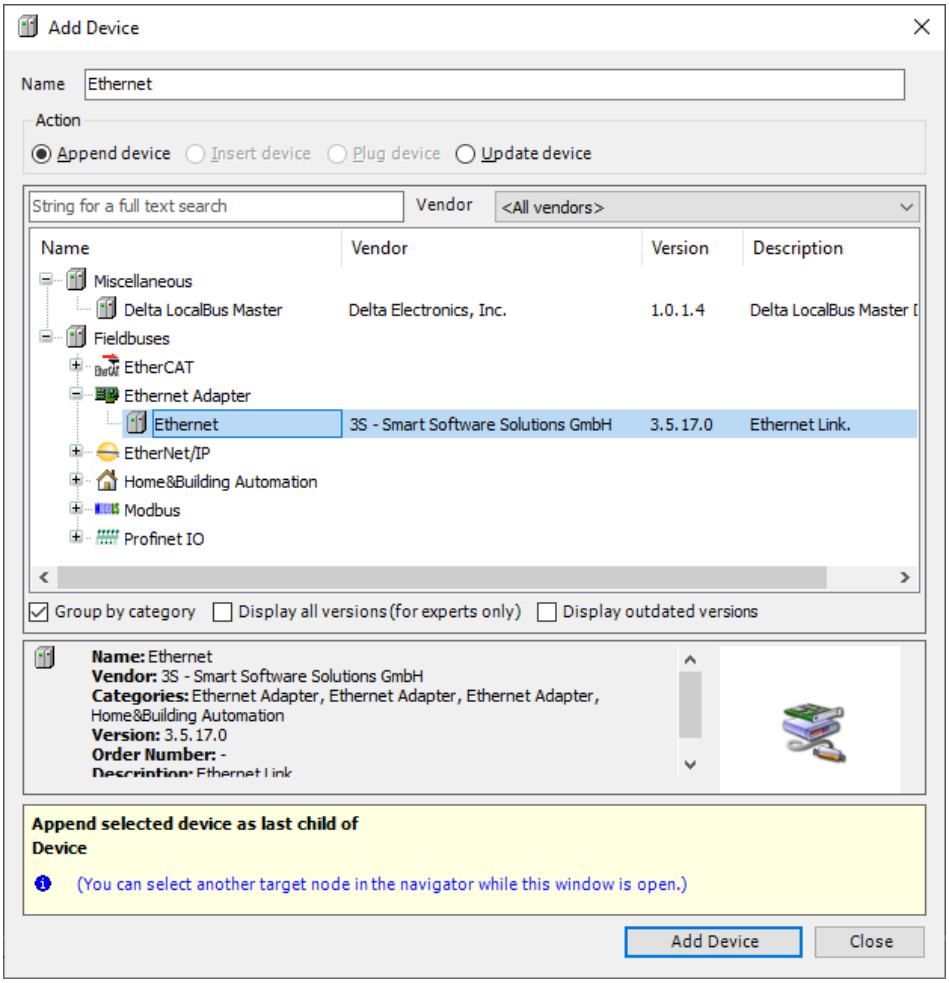




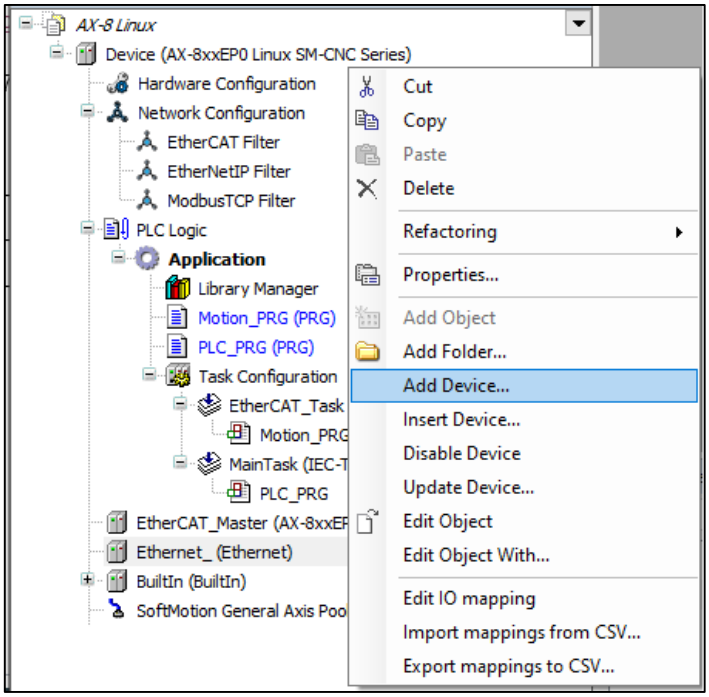

Communication Settings
Applications
Backup and Restore
Files
Log
PLC Settings
PLC Shell
Licensed Software Metrics
System Settings
Right Settings
System Parameters
System IEC Objects
Runtime Clock
PLC Time:
Read PLC Time
Date:
Wednesday, October 26, 2022
Write PLC Time
Time:
4:10:29 PM
Sync with Local Time
Time Zone
PLC Timezone:
Read Timezone
Timezone:
(UTC+08:00) Asia/Shanghai
Write Timezone
Network
Apply IP settings while download
GLAN1
GLAN2
IP Address Mode:
Static
IP address:
192 . 168 . 0 . 10
Subnet mask:
255 . 255 . 255 . 0
Default gateway:
0 . 0 . 0 . 0
Read from PLC

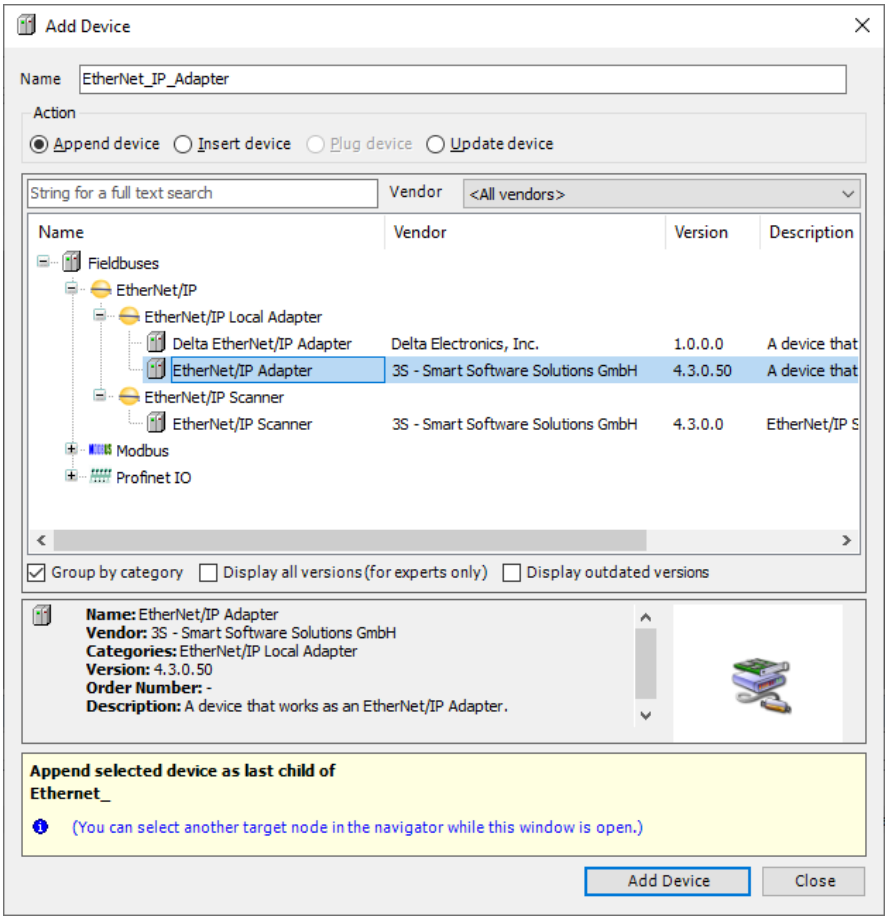
2. Right-click **Device (AX-8xxEP0 Linux SM-CNC Series)** > **Add Device** > select **Ethernet Adapter** > **Ethernet**.



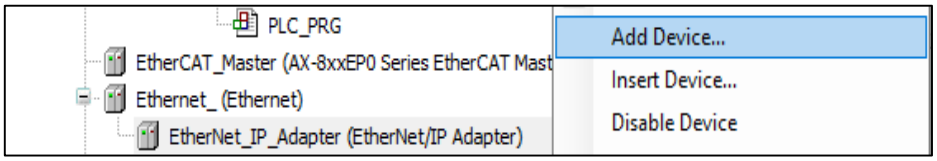


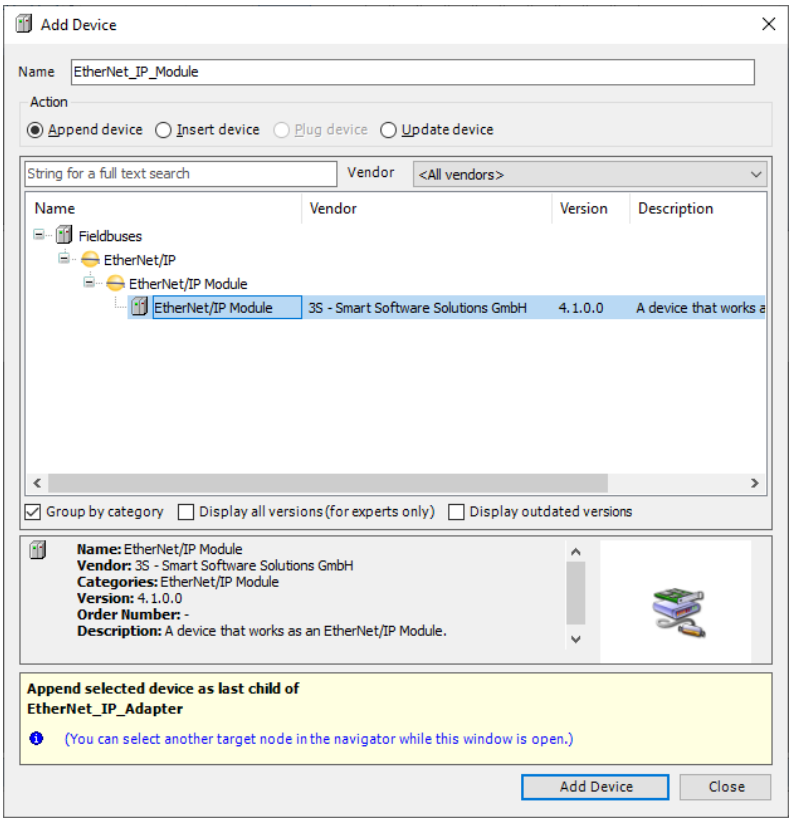
3. Right-click **Ethernet** > **Add Device** > select **EtherNet_IP Adapter**.


AX-8 Linux
Device (AX-8xxEP0 Linux SM-CNC Series)
Hardware Configuration
Network Configuration
EtherCAT Filter
EtherNetIP Filter
ModbusTCP Filter
PLC Logic
Application
Library Manager
Motion_PRG (PRG)
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task
Motion_PRG
MainTask (IEC-T
PLC_PRG
EtherCAT_Master (AX-8xxEP
Ethernet_ (Ethernet)
BuiltIn (BuiltIn)
SoftMotion General Axis Poo
Cut
Copy
Paste
Delete
Refactoring
Properties...
Add Object
Add Folder...
Add Device...
Insert Device...
Disable Device
Update Device...
Edit Object
Edit Object With...
Edit IO mapping
Import mappings from CSV...
Export mappings to CSV...

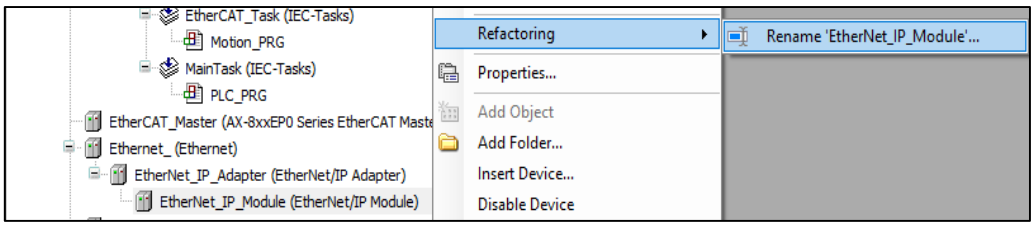

Add Device
Name
EtherNet_IP_Adapter
Action
Append device
Insert device
Plug device
Update device
String for a full text search
Vendor
<All vendors>
Name
Vendor
Version
Description
Fieldbuses
EtherNet/IP
EtherNet/IP Local Adapter
Delta EtherNet/IP Adapter
Delta Electronics, Inc.
1.0.0.0
A device that
EtherNet/IP Adapter
3S - Smart Software Solutions GmbH
4.3.0.50
A device that
EtherNet/IP Scanner
EtherNet/IP Scanner
3S - Smart Software Solutions GmbH
4.3.0.0
EtherNet/IP S
Modbus
Profinet IO
Group by category
Display all versions (for experts only)
Display outdated versions
Name: EtherNet/IP Adapter
Vendor: 3S - Smart Software Solutions GmbH
Categories: EtherNet/IP Local Adapter
Version: 4.3.0.50
Order Number: -
Description: A device that works as an EtherNet/IP Adapter.
Append selected device as last child of
Ethernet_
(You can select another target node in the navigator while this window is open.)
Add Device
Close

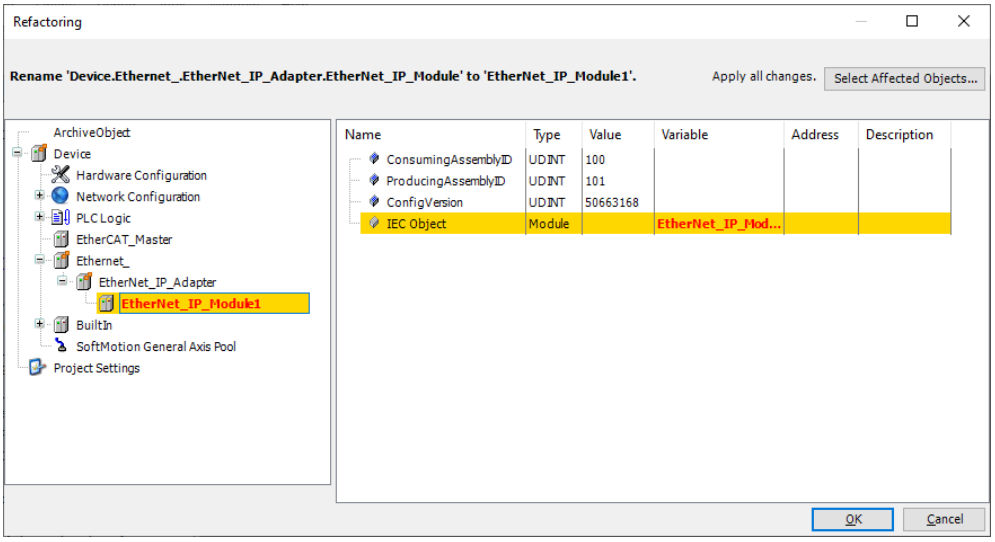
4. Right-click **EtherNet_IP Adapter** and select **Add Device** > **EtherNet/IP Module**.



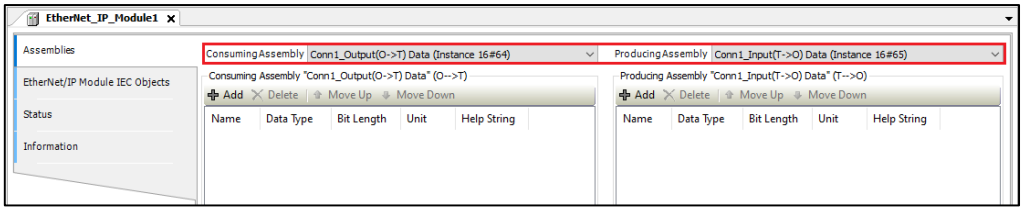


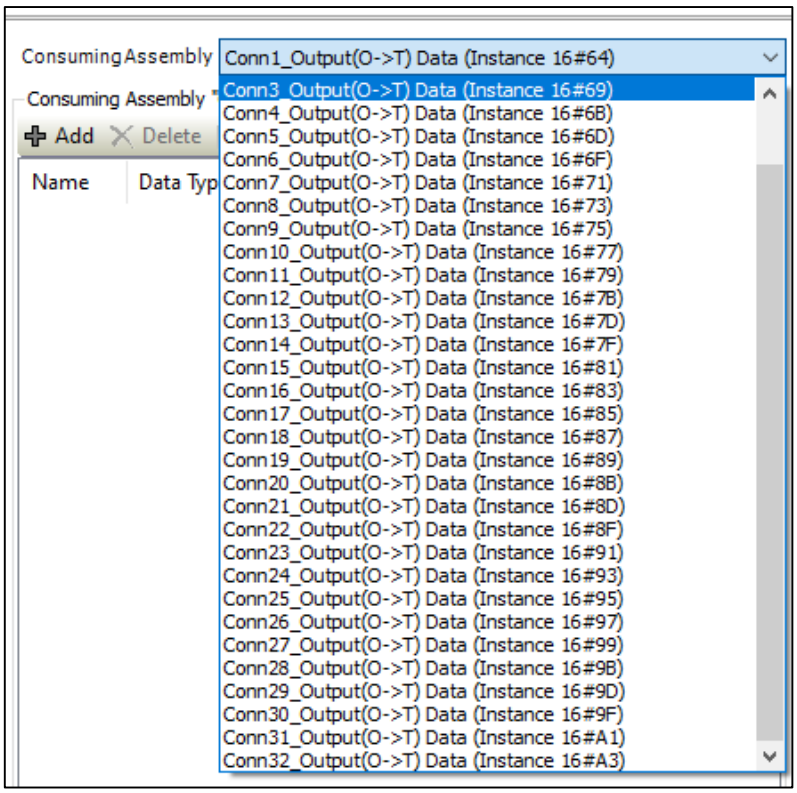
5. Right-click **EtherNet_IP _Module** > **Refactoring** > Rename it for easy identification, here it is named **EtherNet_IP _Module1**.

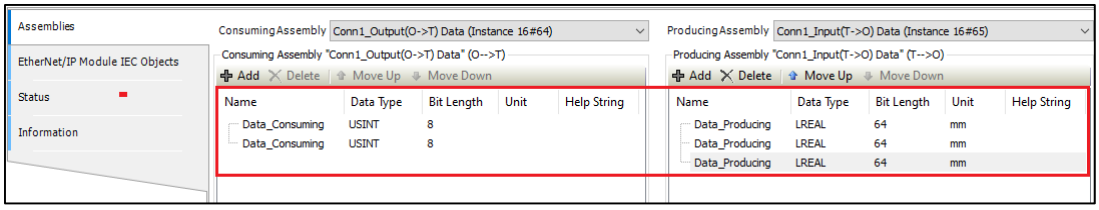




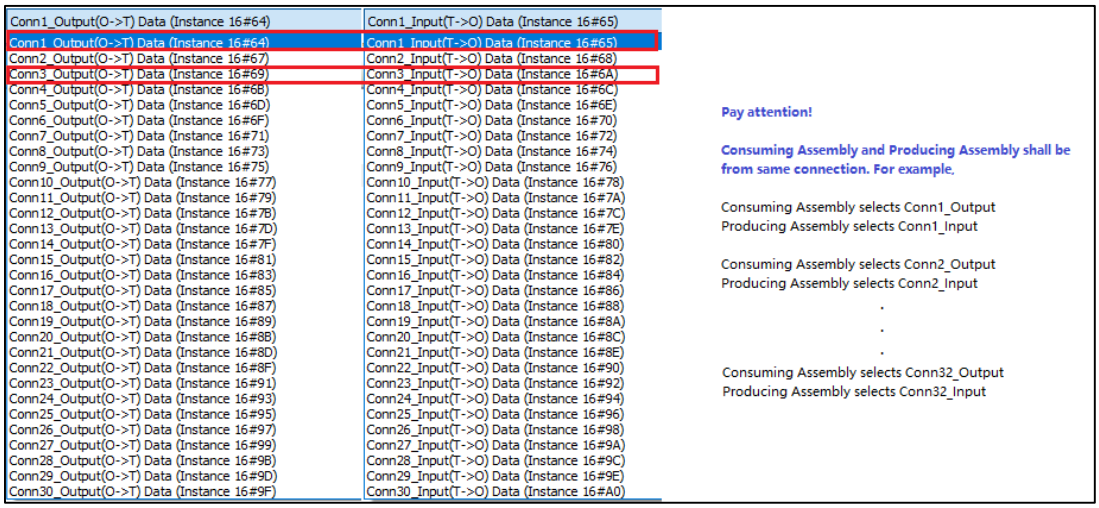
6. **EtherNet_IP _Module1** > **Assembly** can add data exchange for output/input, and can specify which connection to set the data on. There are 32 connections to choose from.



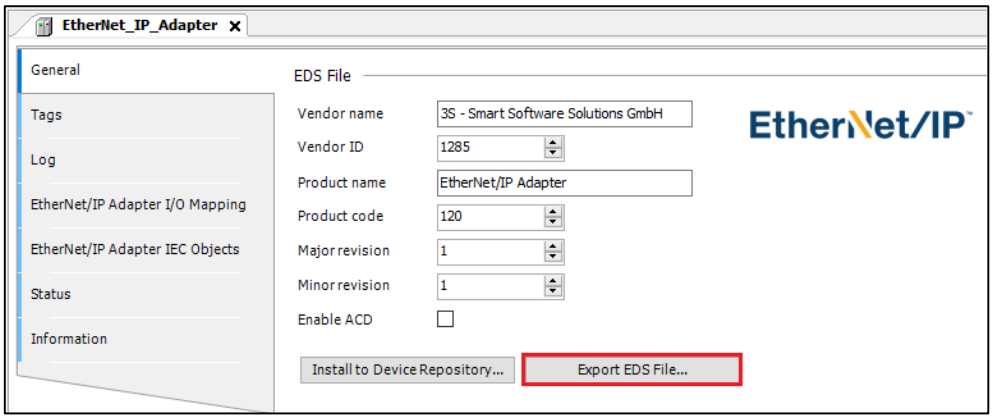




Note: Consuming Assembly and Producing Assembly Connection have the same connection, as shown in the following figure:

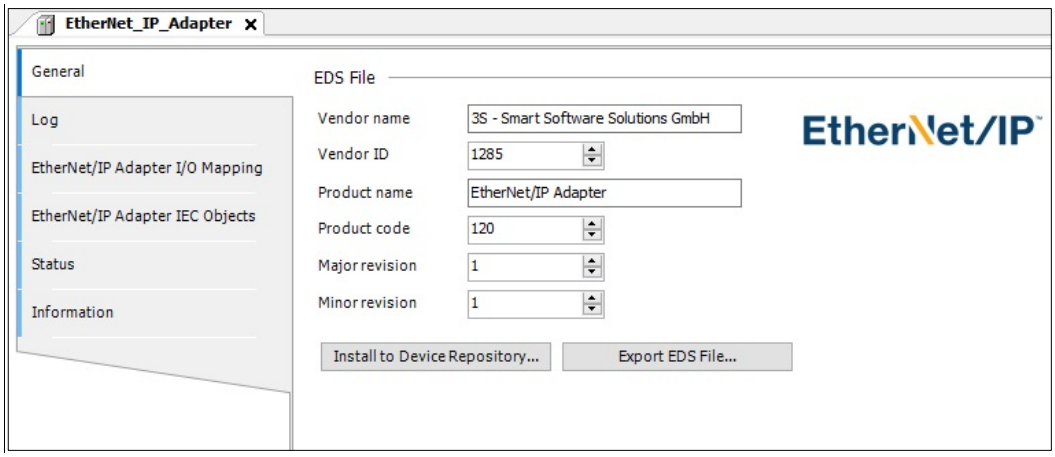


7. Repeat steps 4~6 to create the second module in sequence. After setting the two modules and the number of connections, go to **General** tab to modify the content of the EDS file, and **Export EDS file** after completion, which is convenient to use when setting up the EtherNet/IP Scanner.



EtherNet/IP Local adapter general tab

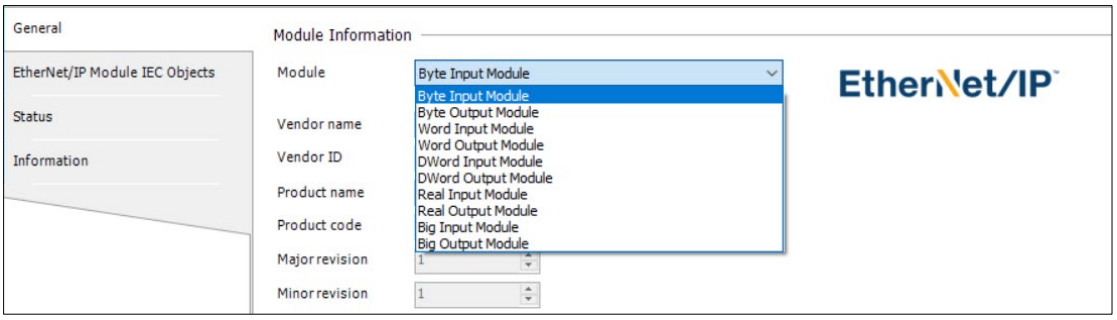
The local adapter displays the general information in the device description file, and the user can define this information to export the EDS file.



Project	Description
Vendor name	Supplier name
Vendor ID	Provided by ODVA Association
Product name	EDS file name
Product code	Product Code
Major revision	Major revision
Minor revision	Minor revision
Install to Device Repository	If a device with the same device ID is already installed, it will ask if the device should be overwritten. If the device is plugged under the EtherNet/IP scanner as a remote adapter, user will be asked to update the device automatically.
Export EDS File	The EDS file will be created and stored on the local computer. This can use EDS files in an external setting environment.

EtherNet/IP Module General Tab

Supports modules of different data types. After setting multiple data type modules, return to the local adapter to output EDS files.


General
EtherNet/IP Module IEC Objects
Status
Information
Module Information
Module
Byte Input Module
Byte Input Module
Byte Output Module
Word Input Module
Word Output Module
DWord Input Module
DWord Output Module
Real Input Module
Real Output Module
Big Input Module
Big Output Module
Vendor name
Vendor ID
Product name
Product code
Major revision
1
Minor revision
1
EtherNet/IP

5.3.8 High-Speed IO Setting

5.3.8.1 BuiltIn_DIO Parameter

Parameters Tab

After double-clicking BuiltIn_DIO in the device tree, the BuiltIn_DIO parameter tab is displayed.

DI Filter Select: Set X0 ~ X7 input filter time

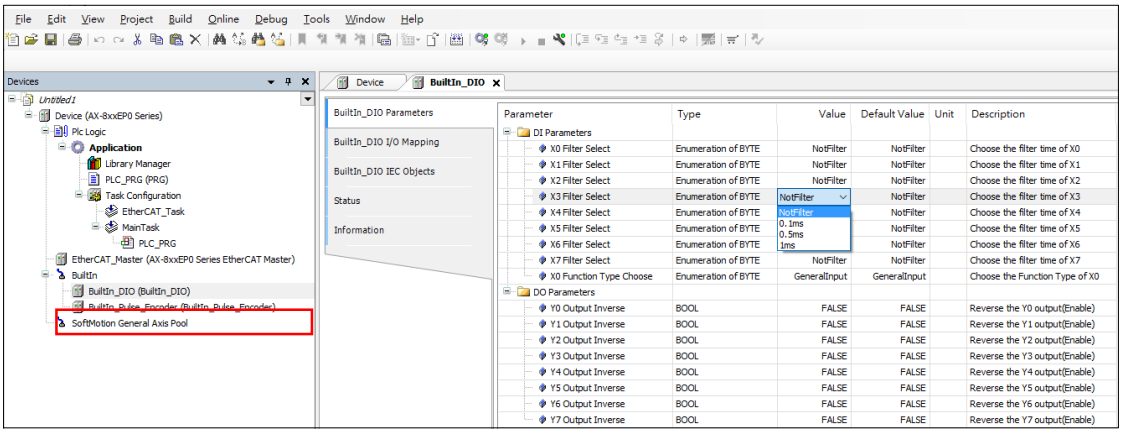
- Setting Value: Not Filter, 0.1ms, 0.5ms, 1ms
- Default Value: Not Filter

Note*:* The setting can be changed when offline. After Download, the status will be Run.

DO Output Inverse: Set Y0 ~ Y7 out reserve

- Setting Value: True, False
- Default Value: False

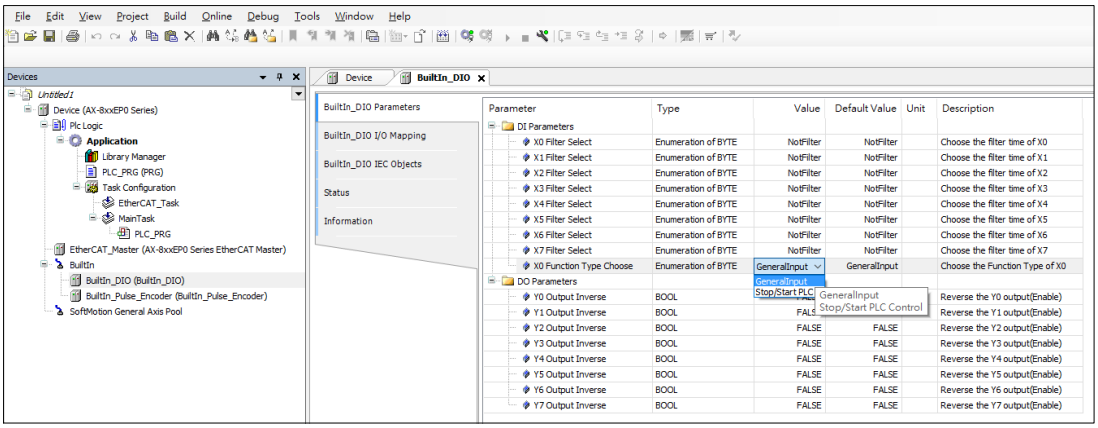
Note: The setting can be changed when offline. After Download, the status will be Run.



X0 Function Type Choose: Set X0 as general output or special software functions

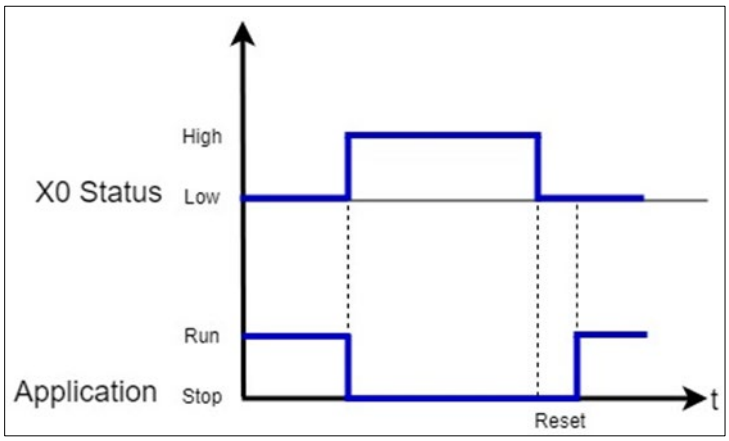
- Value: GeneralInput, Stop/Start PLC Control (default value: GeneralInput)

Note: The setting can be changed when offline. After Download, the status will be Run.



Note: Stop/Start PLC Control Timing diagram.

- When X0 is Low, Application → Reset → Run
- When X0 is High, Application Stop

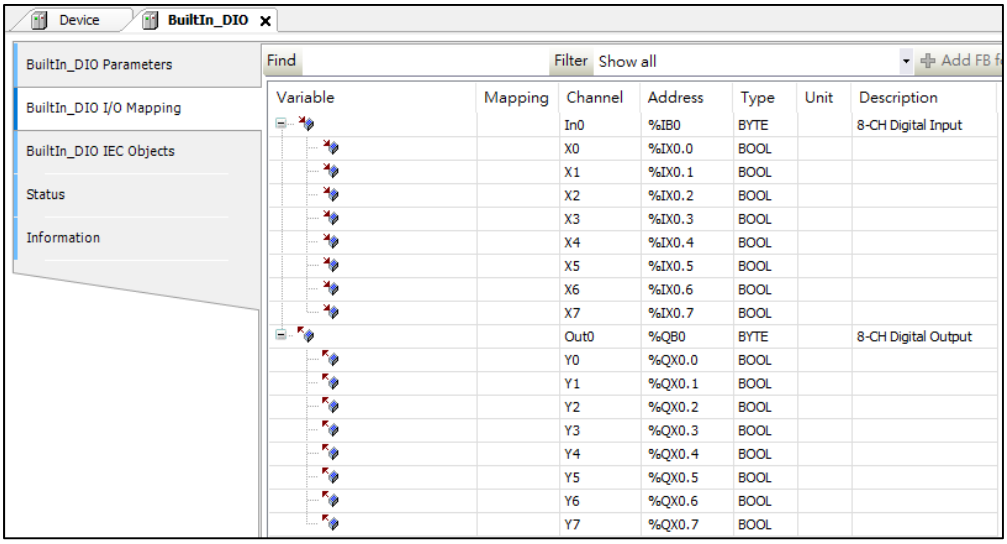


BuiltIn_DIO I/O Mapping

After double-clicking BuiltIn_DIO in the device tree, the BuiltIn_DIO parameter tab is displayed, and then switch to the BuiltIn_DIO I/O mapping tab.

I/O Mapping: Input or output Mapping

- Input 0 or *Output* 0 Mapping (Type: Byte)
- X0 ~ X7 or *Y0* ~ Y7 Single in/output Mapping (Type: BOOL)


Device
BuiltIn_DIO
BuiltIn_DIO Parameters
BuiltIn_DIO I/O Mapping
BuiltIn_DIO IEC Objects
Status
Information
Find
Filter
Show all
Add FB f
Variable
Mapping
Channel
Address
Type
Unit
Description
In0 %IB0 BYTE 8-CH Digital Input
X0 %IX0.0 BOOL
X1 %IX0.1 BOOL
X2 %IX0.2 BOOL
X3 %IX0.3 BOOL
X4 %IX0.4 BOOL
X5 %IX0.5 BOOL
X6 %IX0.6 BOOL
X7 %IX0.7 BOOL
Out0 %QB0 BYTE 8-CH Digital Output
Y0 %QX0.0 BOOL
Y1 %QX0.1 BOOL
Y2 %QX0.2 BOOL
Y3 %QX0.3 BOOL
Y4 %QX0.4 BOOL
Y5 %QX0.5 BOOL
Y6 %QX0.6 BOOL
Y7 %QX0.7 BOOL

5.3.8.2 BuiltIn_Pulse_Encoder Parameter

Parameters Tab

After double-clicking BuiltIn_Pulse_Encoder in the device tree, the BuiltIn_Pulse_Encoder parameter tab is displayed.

Pulse Encoder input type:

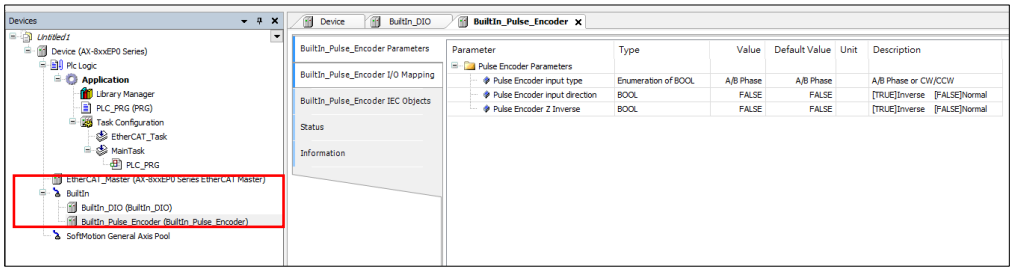
- *Setting* Value: A/B Phase, CW/CCW
- Default Value: A/B Phase

Pulse Encoder input direction:

- Setting Value: True(Inverse), False(Normal)
- Default Value: False

Pulse Encoder Z Inverse:

- Setting Value: True(Inverse), False(Normal)
- Default Value: False



Note: The setting can be changed when offline. After Download, the status will be Run.

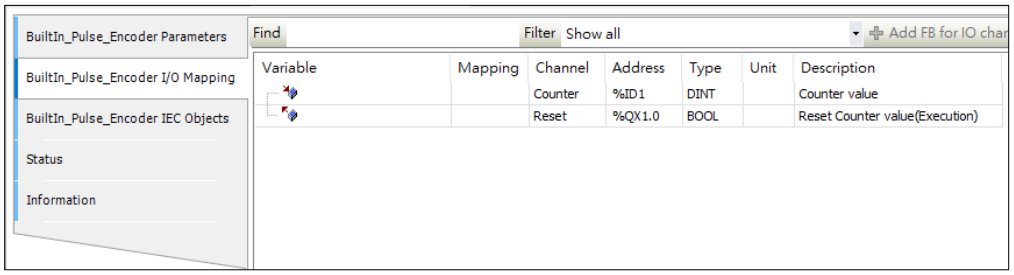
BuiltIn_Pulse_Encoder I/O Mapping

Double-click BuiltIn_Pulse_Encoder in the device tree to display the BuiltIn_Pulse_Encoder parameter tab, and then switch to the BuiltIn_Pulse_Encoder I/O mapping tab.

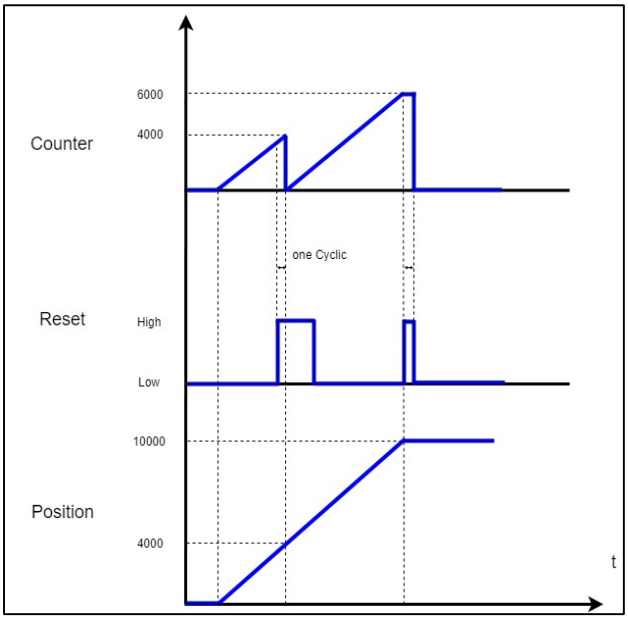
Pulse Encoder Mapping: Pulse Encoder Input and Reset Mapping

- Counter (Type: DINT)

- Reset (Type: BOOL & Execution)

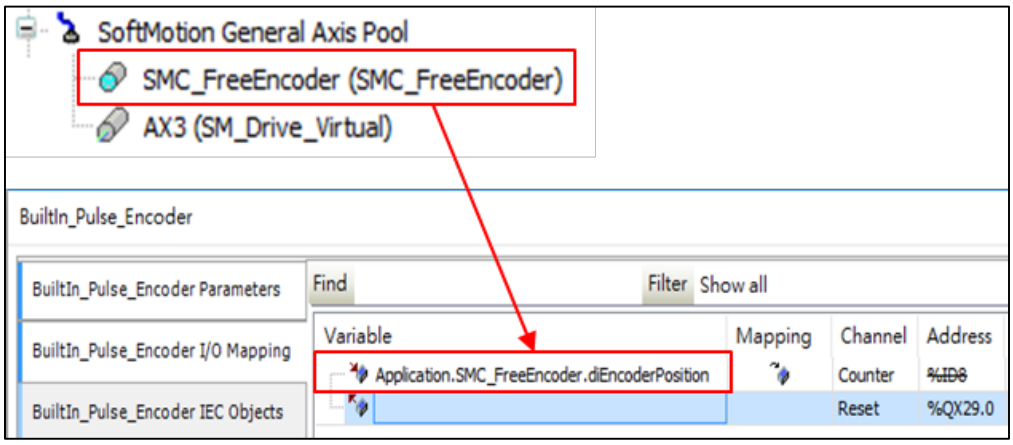


- Reset Timing diagram



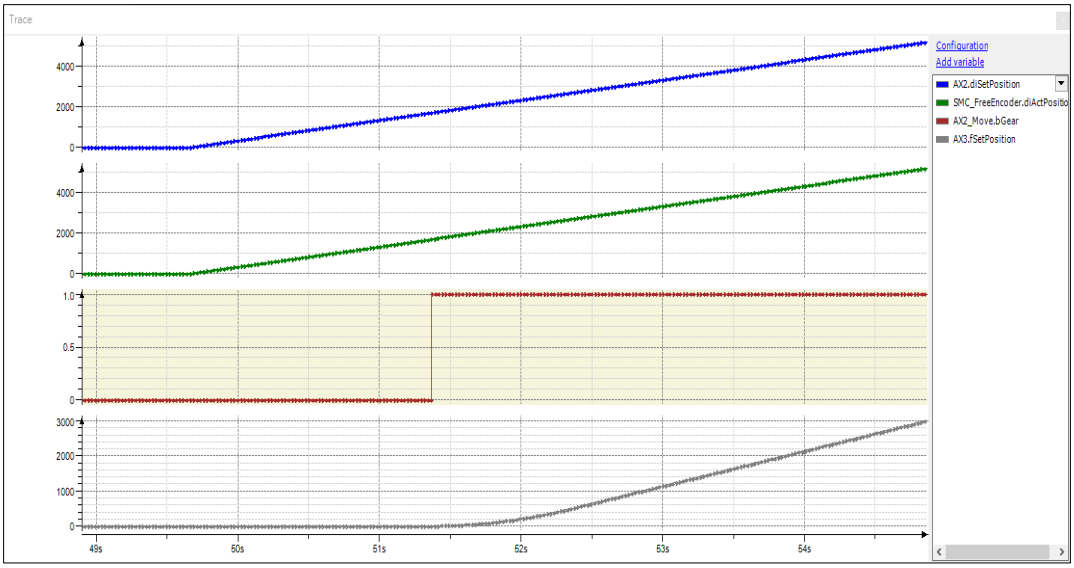
Pulse Encoder example:

Right-click the **SoftMotion General Axis Pool** node in the device tree > **Add Device** > **Free Encoder** > Add **SMC_FreeEncoder** and then map it with Counter.



Use Encoder of R1-EC5621 as output to AX-8xxE Pulse Encoder.

Use SMC_FreeEncoder as master axis and do GearIn with virtual axis.



5.4 Motion control device

Axis setting

Motion control equipment is mainly used to set axis parameters. In most applications, user can set axis parameters in DIADesigner-AX. DIADesigner-AX provides user with a convenient editing environment. In this editing environment, the axis parameter setting of motion control is defined as its single data structure. This data structure contains multiple related parameters.

Axis introduction

In a motion control system, the object of motion control is called an axis. This axis includes physical servo drives, encoders and virtual servo drives. The following table lists all axis types:

Types	Description
Positioning Axis*1	Perform basic positioning control such as absolute positioning, relative positioning etc and other functions through EtherCAT communication.
Synchronous axis *2	Servo control is performed through EtherCAT communication, in addition to basic positioning control, it also includes functions such as synchronous motion such as electronic cams.
Pulse type servo axis	Physical servo control through pulse impulse
Virtual axis	The unconnected physical servo used only in the program can execute motion control commands.
Encoder axis	Use physical encoder (SSI encoder, incremental encoder) as signal source
Virtual encoder shaft	The encoder without wiring is only used in the program.

Note:

1. Positioning axis should use **DL_MotionControl** Library
2. Synchronous axis needs to be used with DL_MotionControl and SM3_Basic library.

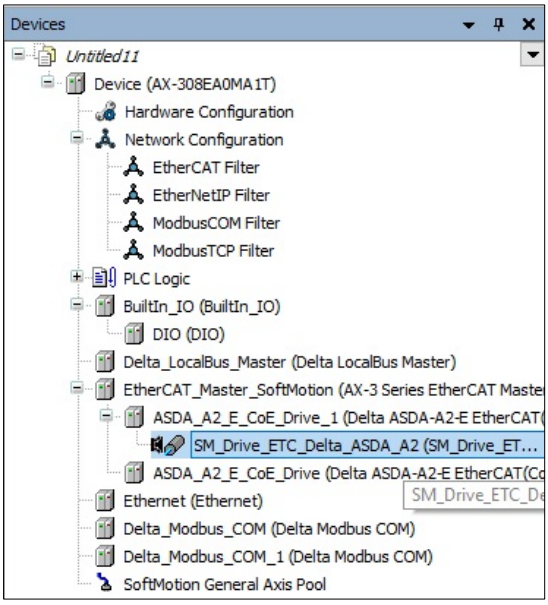
5.4.1 Synchronous Axis

Support slave device:

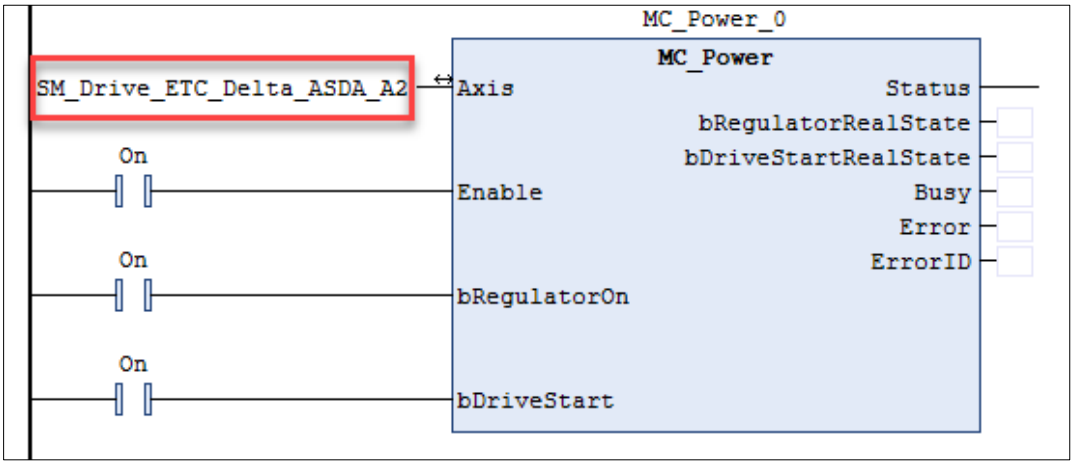
- ASDA-A2-E, ASDA-A3-E, ASDA-B3-E
- R1EC-5621

Use the axis in the program:

When the servo axis is added to the project, the system will automatically generate its servo axis name (the user can also change its name), as shown in the following figure. The name of the servo axis generated by DIADesigner-AX is SM_Drive_ETC_Delta_ASDA_A2.



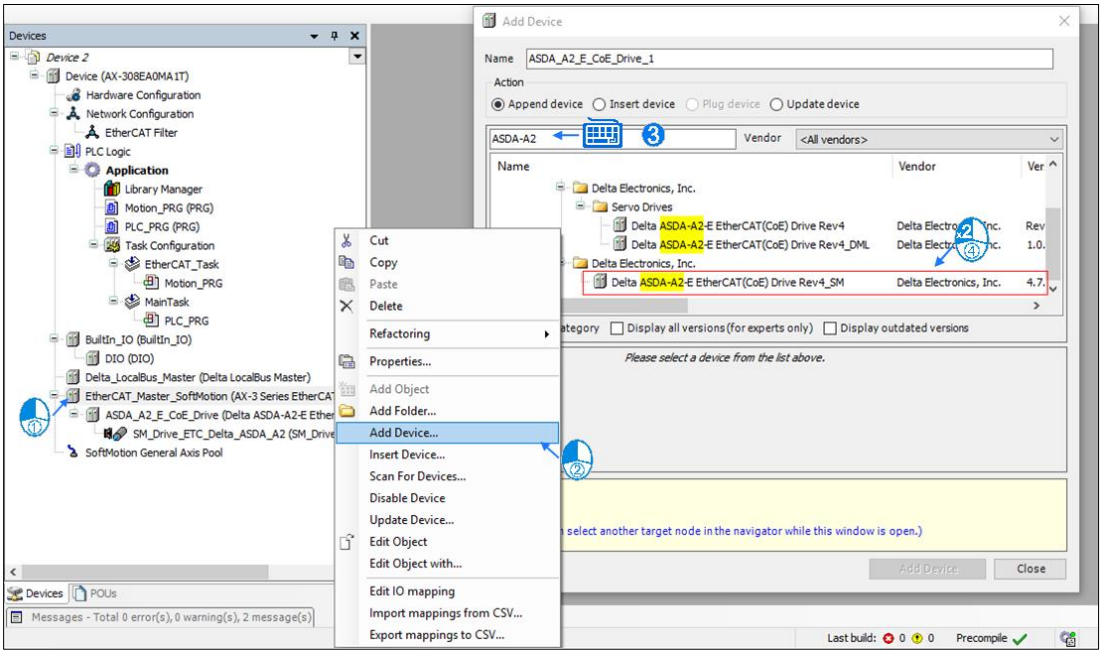
Therefore, when using the motion function block, if the user needs to input the corresponding axis, user can input SM_Drive_ETC_Delta_ASDA_A2 in the **Axis** pin as shown below.



5.4.1.1 AX-3 series

Please follow the step below to add a synchronous axis:

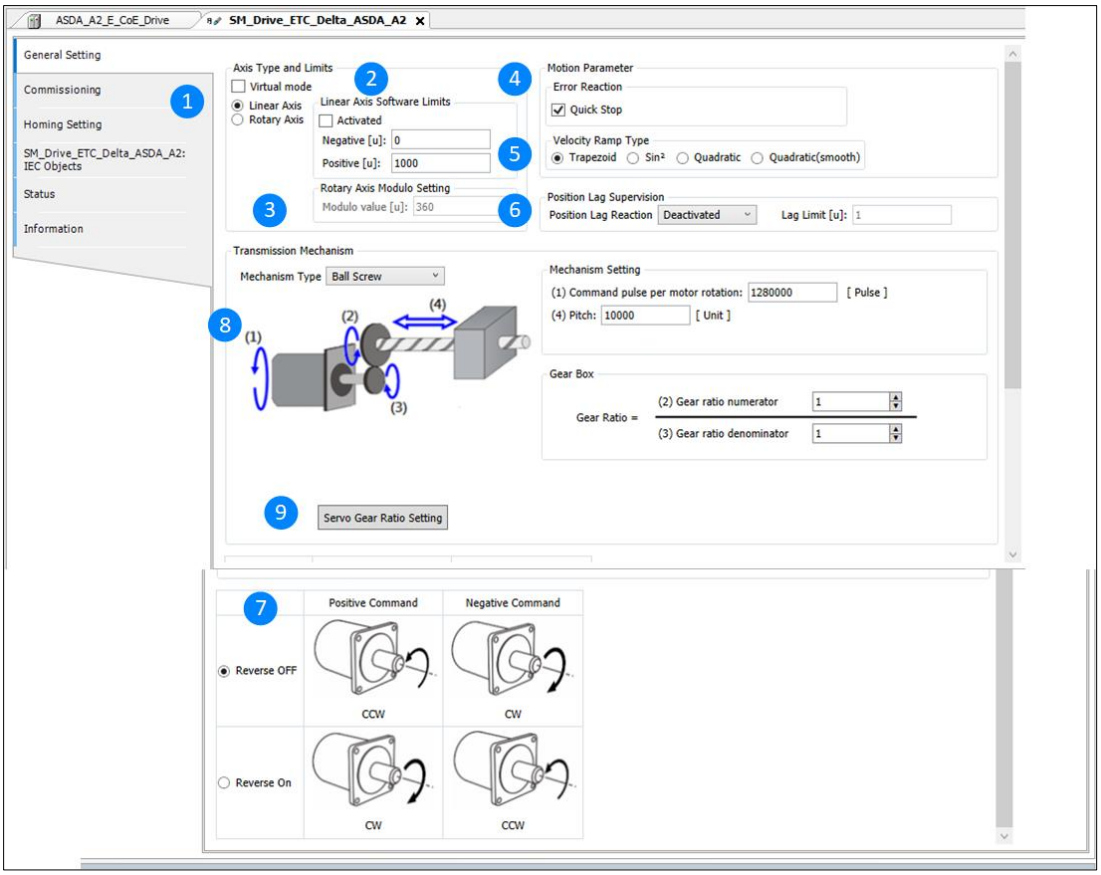
Right-click **EtherCAT_Master_SoftMotion** in the device tree > **Add Device** > Search **ASDA-A2 > ASDA-A2-E_SM** in the device tree.



Perform basic positioning control such as absolute positioning, relative positioning and other functions through EtherCAT communication.

Axis parameter introduction

When creating a servo axis or virtual axis, the axis will generate corresponding axis parameters. The following table lists its related instructions.



① Axis Type and Limits

Name	Description
Virtual mode	Enable virtual mode after checking
Linear Axis/Rotary Axis	Select linear axis/rotary axis

② Linear Axis Software Limits

Name	Description
Activated	Start software limit (only support linear axis)
Negative [u]	Reverse software limit
Positive [u]	Forward software limit

③ Rotary Axis Modulo Setting

Contour (Body)	Description
Modulo value [u]	Set a circle range (only support rotation axis)

④ Motion Parameter

Name	Description
Quick stop	Axis emergency stop
Deceleration [u/s2]	The axis decelerates to a stop (only effective when the quick stop is not started)

⑤ Velocity Ramp type

Name	Description
Trapezoid/Sin2/ Quadratic/ Quadratic (Smooth)	Set axis motion curve

⑥ Position Lag Supervision

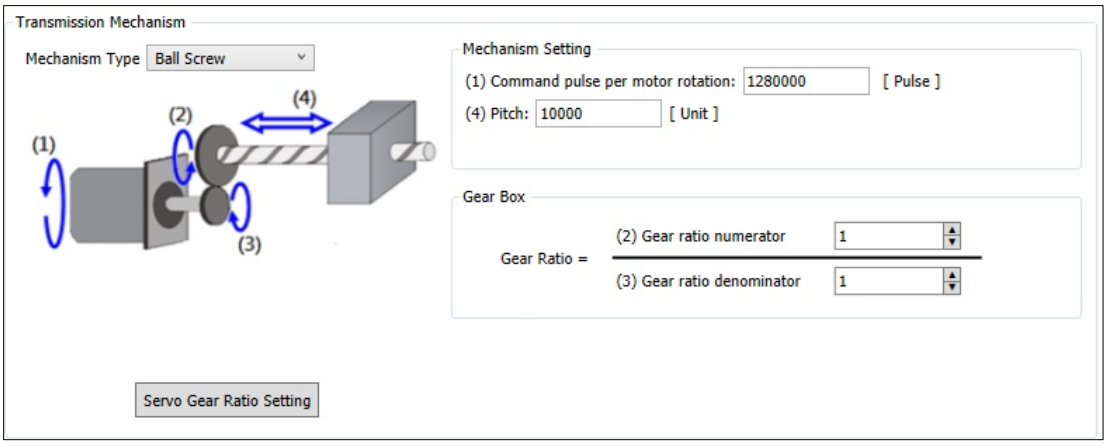
Name	Description
Position Lag Reaction	Set the behavior of the axis after overflow
Lag Limit [u]	Set overflow value

⑦ Positive/Negative Command

Name	Description
Reverse OFF/Reverse On	Set forward/reverse command

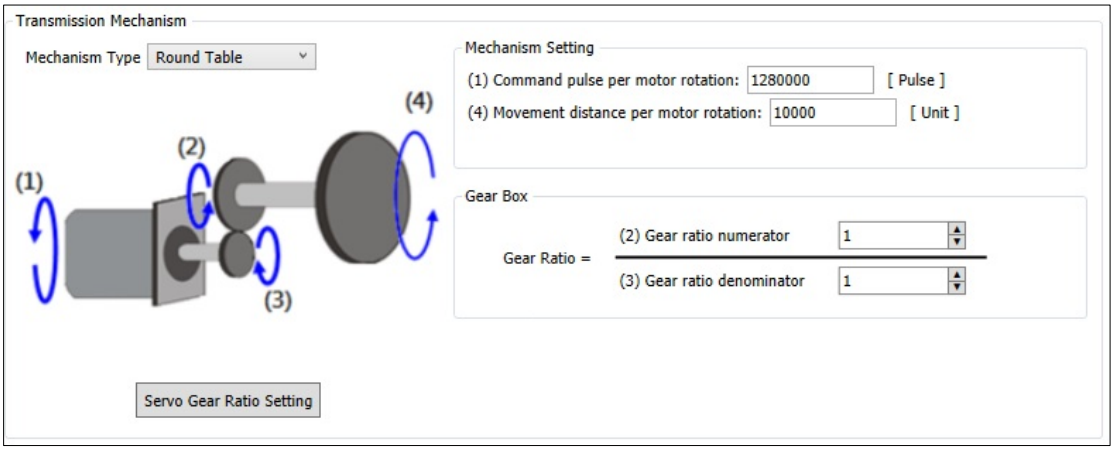
⑧ Transmission Mechanism

Ball Screw Drive



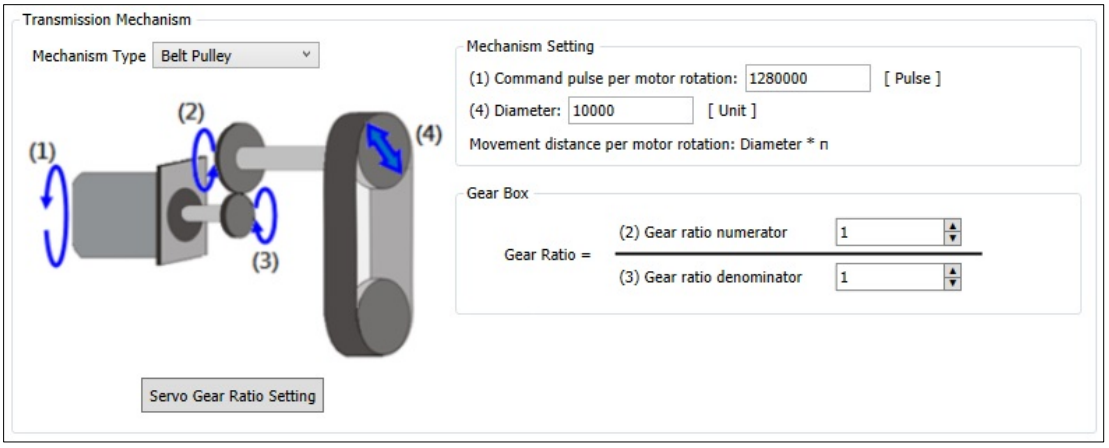
Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Pitch	Pitch

Round table transmission



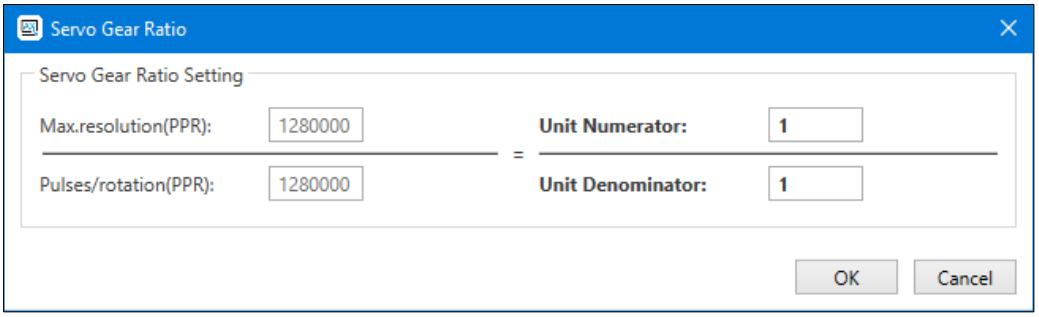
Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Movement distance of the motor per rotation	Movement distance per motor rotation = diameter*n

Belt drive



Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Diameter (moving distance per motor r rotation: diameter *n)	The moving distance of the motor per r rotation = diameter *n

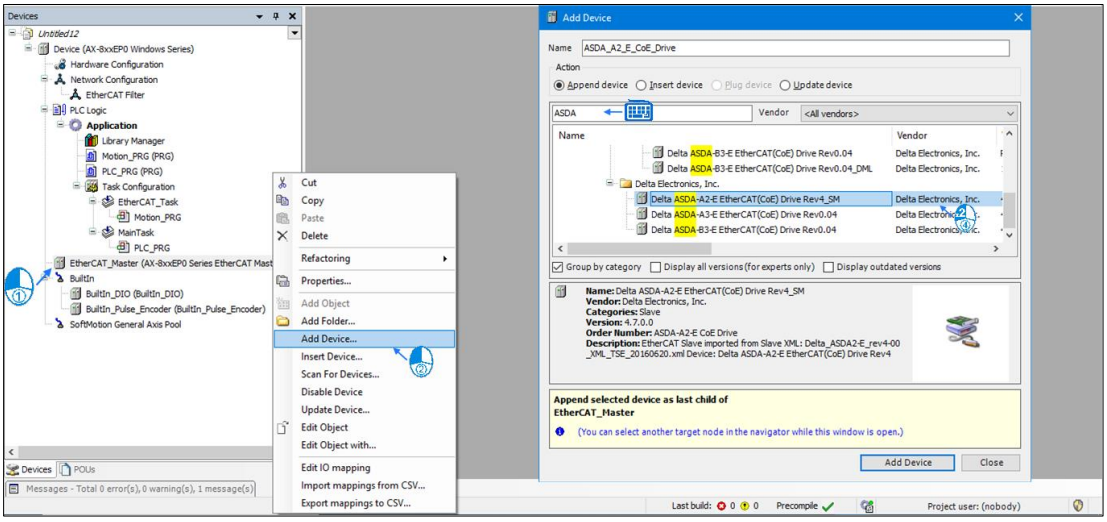
⑨ Servo gear ratio setting



Name	Features
Maximum resolution (PPR)	Maximum resolution (PPR)
Electronic gear ratio numerator	Electronic gear ratio numerator
Electronic gear ratio denominator	Electronic gear ratio denominator
Pulse number/revolution (PPR)	Pulse number/revolution (PPR)

5.4.1.2 AX-8xxEP0 series

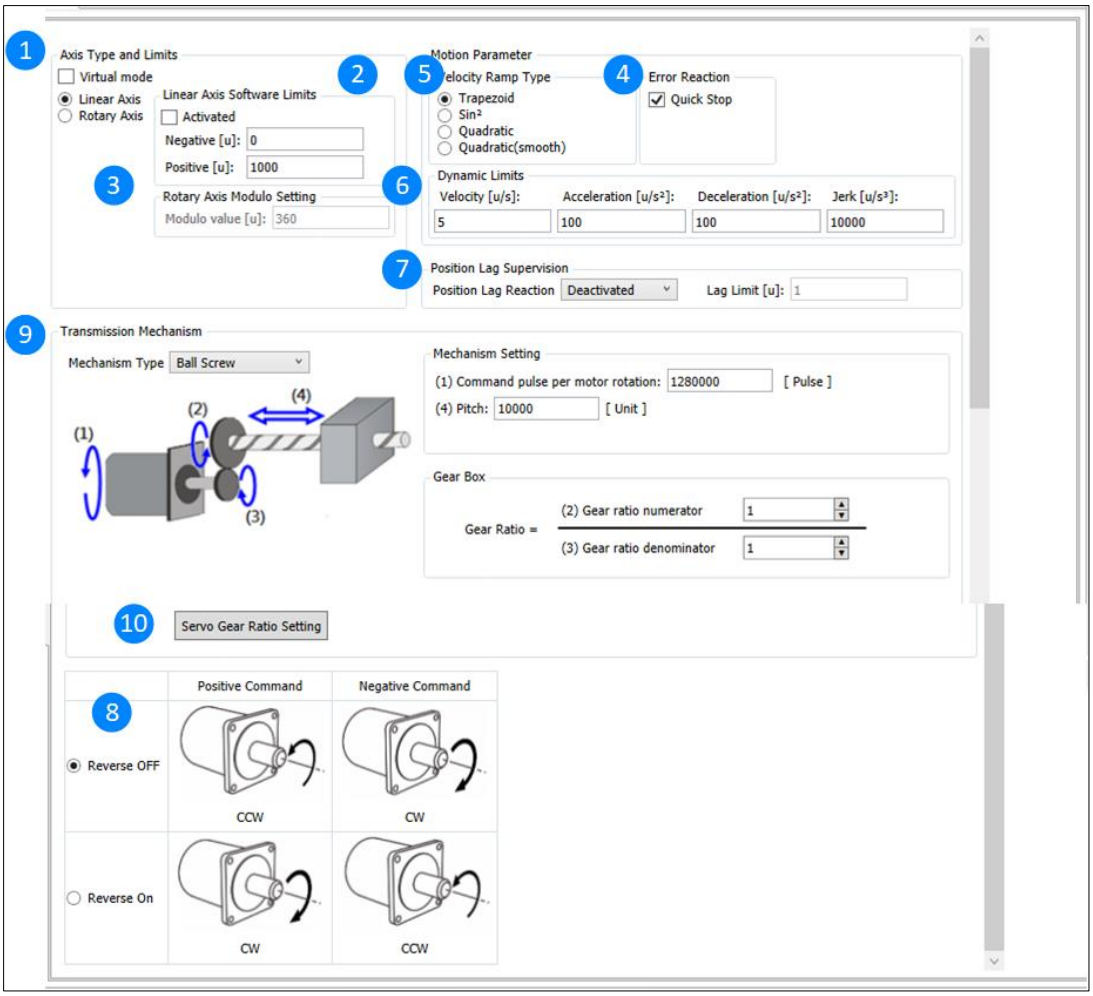
Please follow the steps below to add a synchronous axis from AX-8xxEP0: Right-click **EtherCAT_Master** in the device tree > **Add Device** > Search **ASDA-A2** > **ASDA-A2-E_SM**.



Perform basic positioning control such as absolute positioning, relative positioning... and other functions through EtherCAT communication.

Axis Parameter Introduction

When creating a servo axis or a virtual axis, this axis will generate corresponding axis parameters. The following table lists its related descriptions.



1. Shaft Type and limit

Name	Description
Virtual Mode	Enable virtual mode after checking
Linear axis/rotary axis	Select linear axis/rotary axis

2. Linear axis software limit

Name	Description
Enable	Start software limit (only support linear axis)
Negative[u]	Reverse Software Limit
Positive[u]	Forward software limit

3. Rotary axis modulus value setting

Name	Description

Modulus Value[u]	Set a circle range (only support rotation axis)

4. Wrong reaction

Name	Description
Modulus Value[u]	Set a circle range (only support rotation axis)

5. Rate Ramp Type

Name	Description
Trapezoid/Sin2/ Quadratic/ Quadratic (smoothing)	Set axis motion curve

6. Dynamic Limits

Name	Description
Speed [u/s]	Set the dynamic limit speed. Default value: 5 u/s.
Acceleration [u/s^2]	Set the dynamic limit acceleration. Default value: 100 u/s.
Deceleration [u/s^2]	Set dynamic limit deceleration. Default value: 100 u/s.
Jerk[u/s^2]	Set dynamic limit jerk. Default value: 10000 u/s.

7. Position hysteresis monitoring

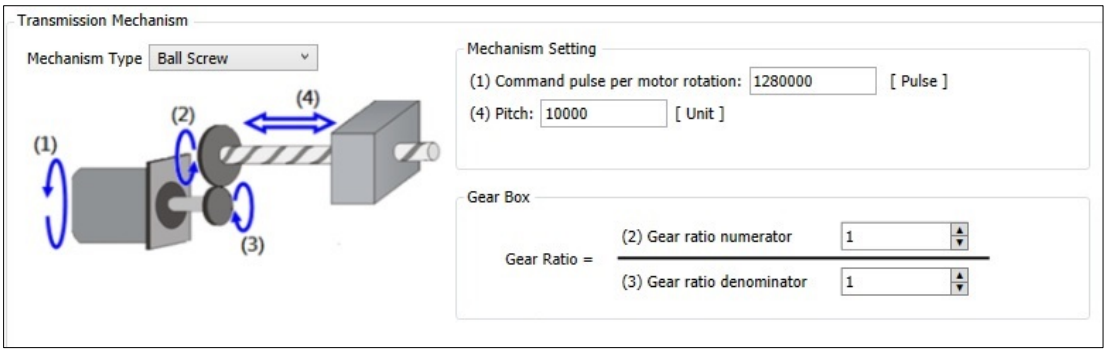
Name	Description
Position lag response	Set the behavior of the axis after overflow
Hysteresis limit [u]	Set overflow value

Positive/negative instructions

Name	Description
Reverse close/reverse open	Set forward/reverse rotation command

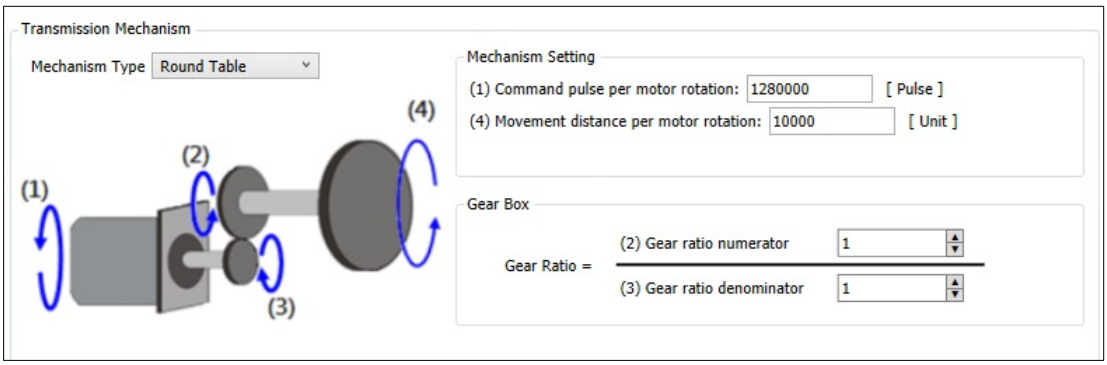
8. Transmission mechanism

Ball screw drive



Name	Features
(1) Command pulse per motor revolution	Command pulse per motor revolution
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Pitch	Pitch

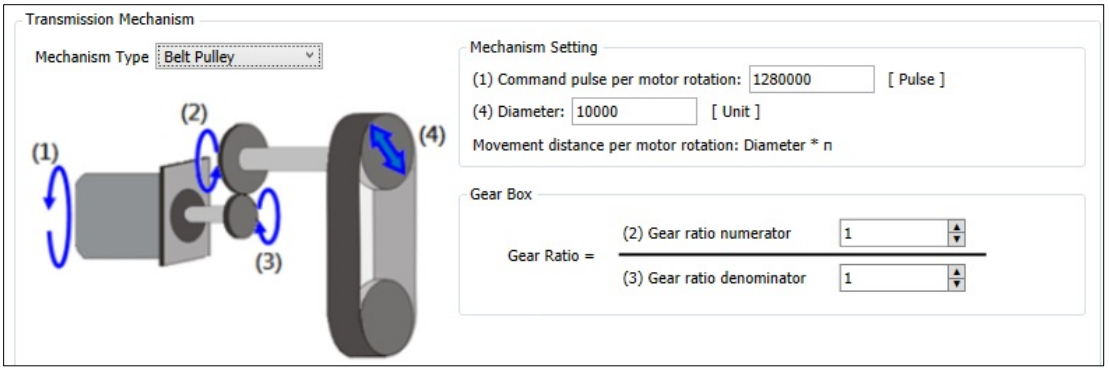
Round Table Transmission



Name	Features
(1) Command pulse per motor revolution	Command pulse per motor revolution

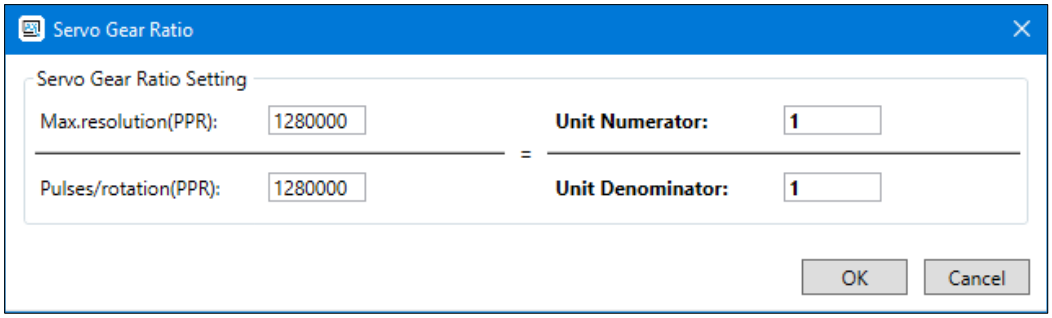
Name	Features
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) The moving distance of the motor per revolution	Moving distance per motor revolution = diameter * n

Belt drive



Name	Features
(1) Command pulse per motor revolution	Command pulse per motor revolution
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Diameter (moving distance per motor revolution: diameter * n)	Moving distance per motor revolution = diameter * n

9. Servo gear ratio setting



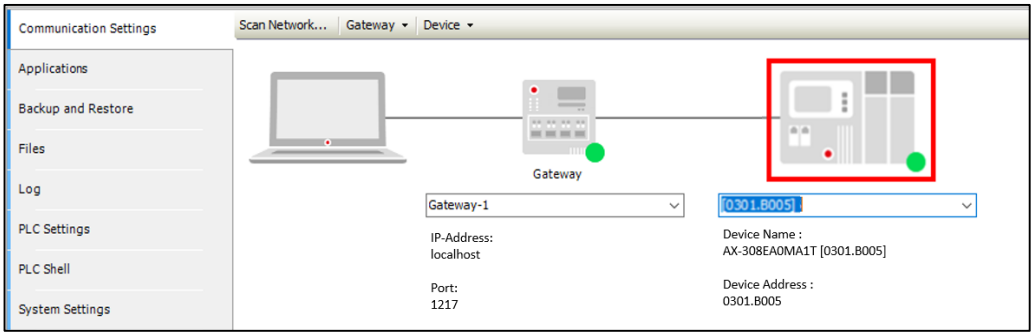
Name	Features
Maximum resolution (PPR)	Maximum resolution (PPR)
Numerator of electronic gear ratio	Numerator of electronic gear ratio
Electronic gear ratio denominator	Electronic gear ratio denominator
Number of pulses/revolution (PPR)	Pulse number/revolution (PPR)

5.4.1.3 COE parameters

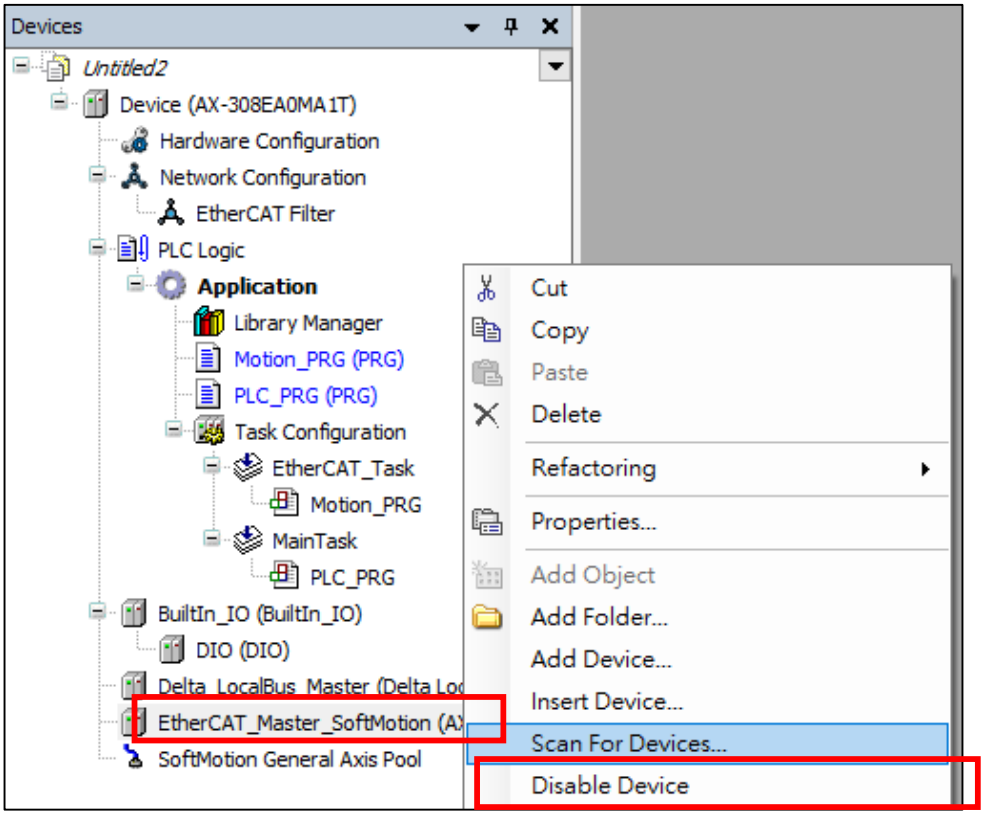
This function is applicable to devices: ASD-A3-E, ASD-B3-E, supporting AX-3/AX-8 series controllers. Take AX-308EA0MA1T and ASD-B3-0421-E as examples.

Follow the steps to open the COE parameter setting window:

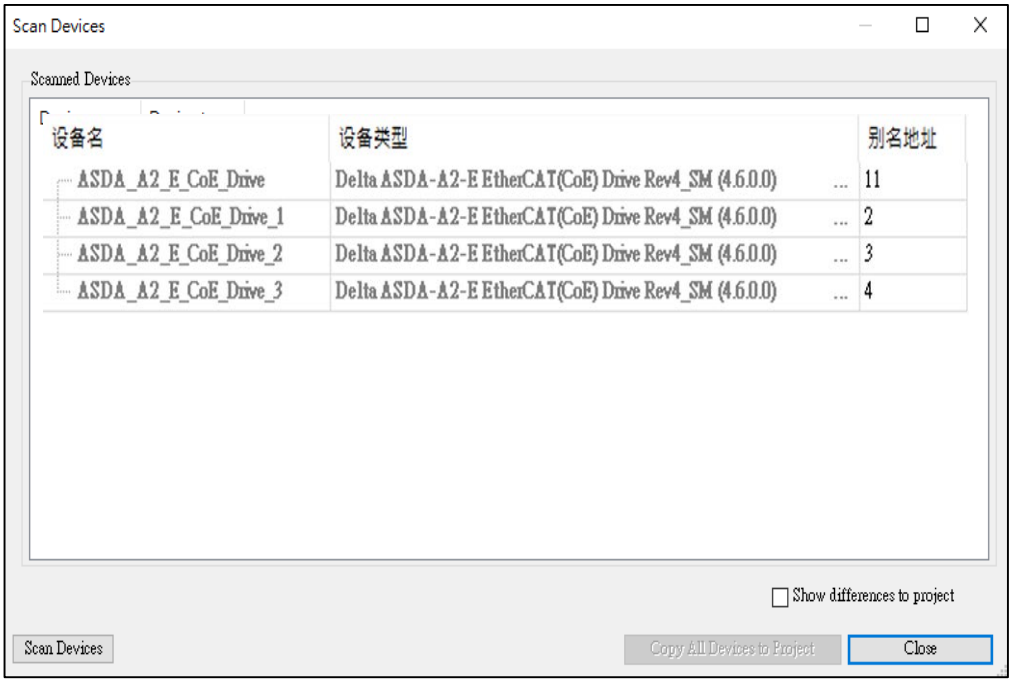
1. Create a project and complete the network connection.



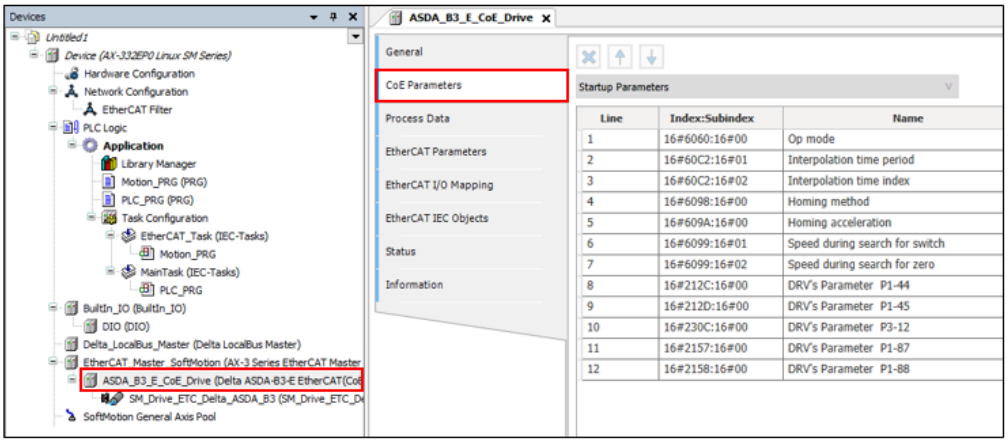
2. Right-click on EtherCAT_Master_SoftMotion and select **Scan For Devices**.



3. Select the desired version.



4. Double-click ASDA_B3_CoE_Drive, select the **CoE Parameters**, and complete the setting page.



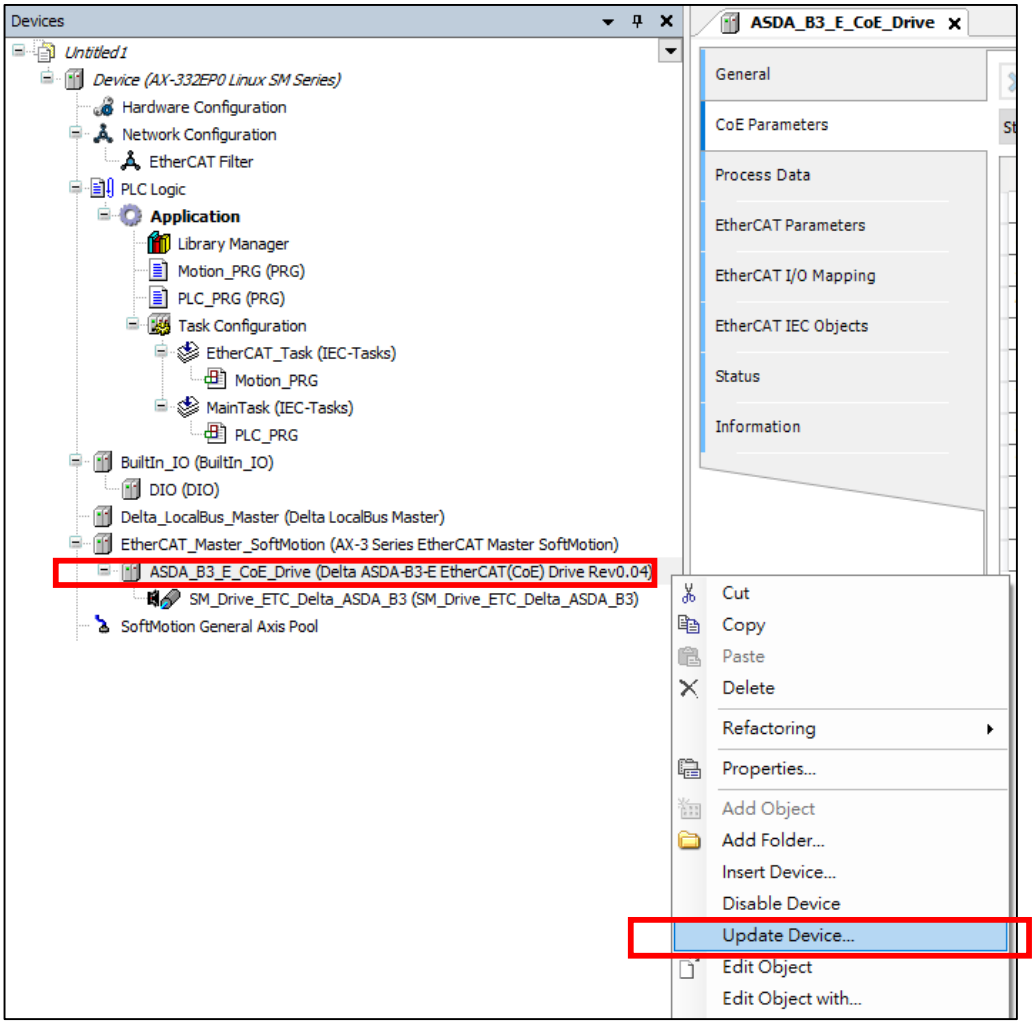
Toolbar Description

Icon	Features	Description
	Compare	Give results where the **value** does not match the **current value** data.
	Upload selected parameters	Upload the selected parameters to the software.
	Upload all parameters	Upload all parameters to the software.
	Download selected parameters	Download the selected parameters to the device.
	Download all parameters	Download all parameters to the device.
	Sync to selected startup parameters	Update the value of the corresponding item in "Startup Parameters" with the value of the currently selected item.
	Sync to all startup parameters	Update the values of all items in **Startup Parameters** according to **All Parameters**.

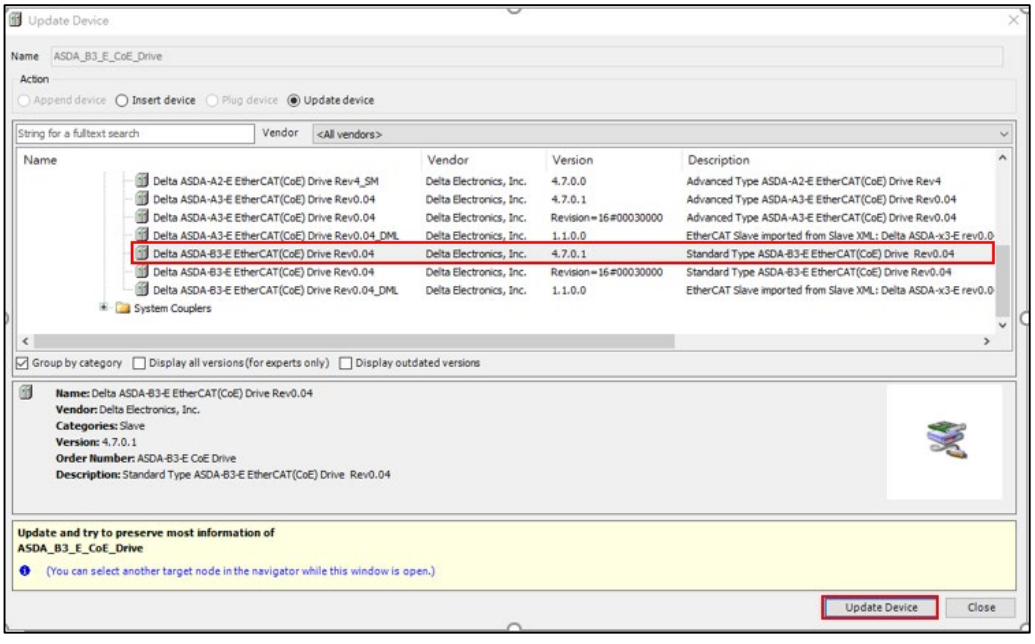
Support for updating devices

This update function only supports ASD-A3-E and ASD-B3-E. The following takes ASD-A3-E to update device parameters to ASD-B3-E as an example.

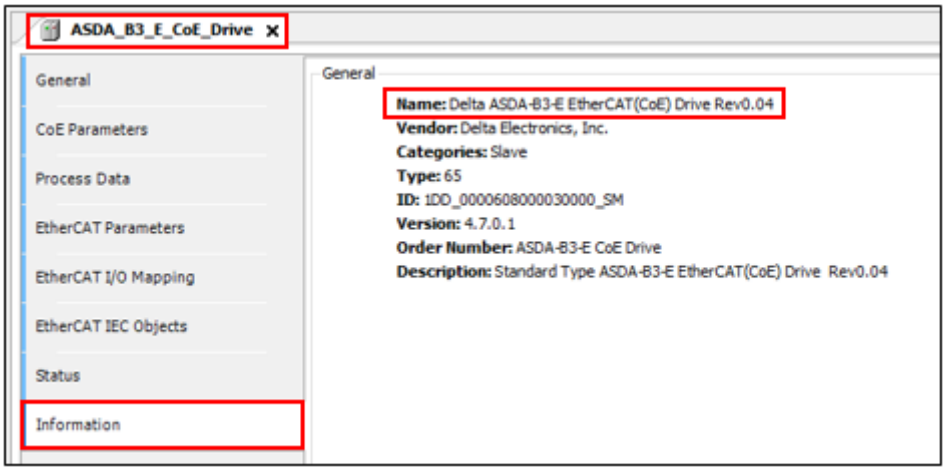
1. Right-click on **EtherCAT Master_Softmotion** > **ASD_A3_CoE_Drive** and select **Update Device**.



2. Select the device to be updated and click **Update Device**.



Result: The update is successful.Go to the **Information** to confirm the updated device name.



5.4.2 Positioning Axis and Speed Axis

This section applies to the following devices:

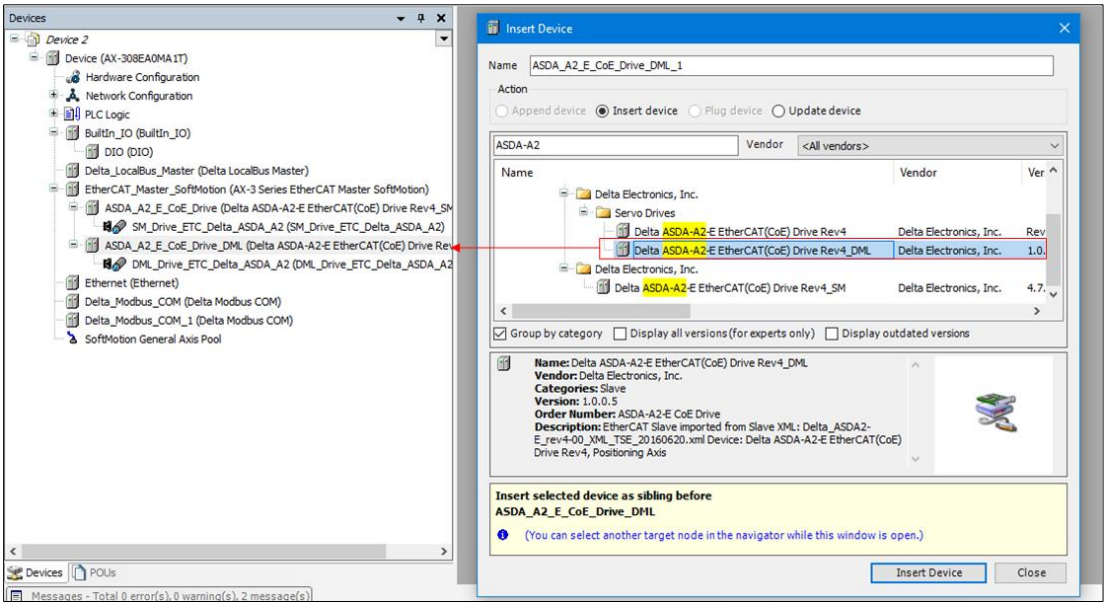
Supporting Device		Position Axis	Velocity Axis
AC Motor Drives	C2000Plus	V	V
	CH2000	V	V
	MH300	-	V
	MS300	-	V
Servo Drives	ASD-A2-E	V	-
	ASD-A3-E	V	-
	ASD-B3-E	V	-

Positioning axis Reference (Axis_REF_DML)

Category	Type	Function group	Description
Single-axis motion control instructions	Motion control type	Positioning control	"MC_": Motion control instructions based on PLCopen "DMC_": Delta custom motion control instruction "SMC_": Motion instruction "MC_XXX_DML": Delta custom motion control instructions, used for positioning axis
		speed control	
		Torque control	
		Synchronization control	
	Management	Management function	
Multi-axis motion	Motion control type	Axis group movement	Multi-axis motion

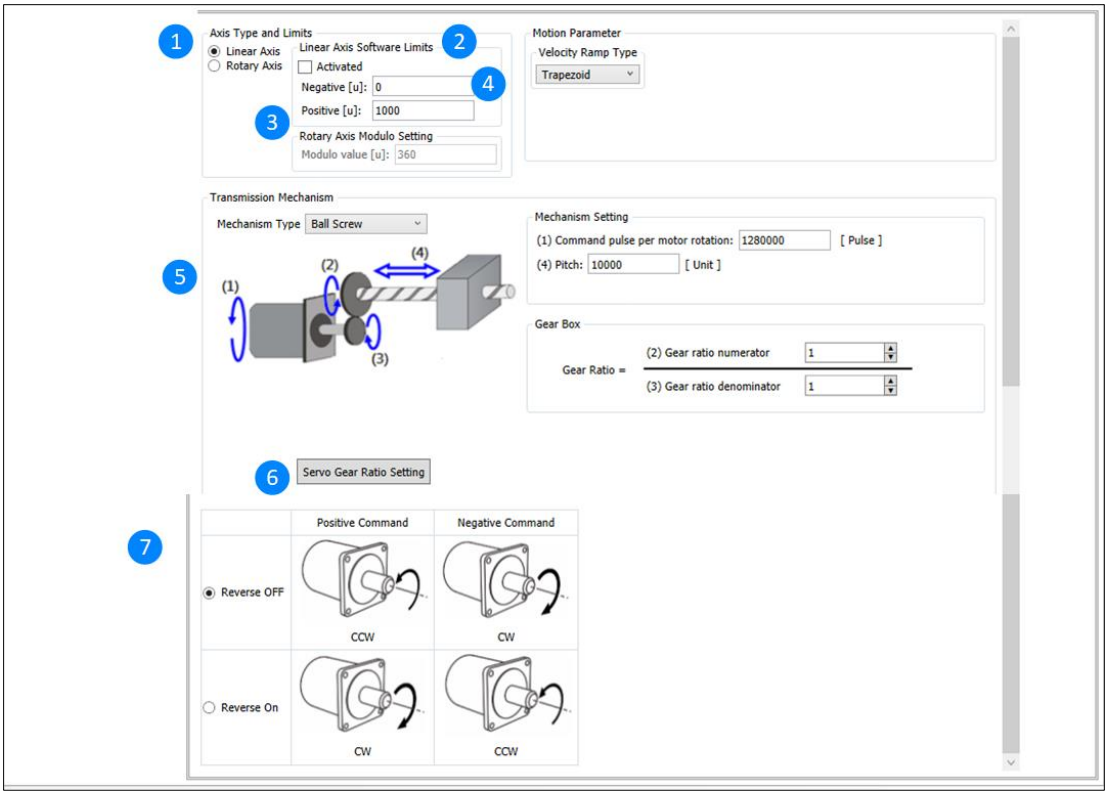
Category	Type	Function group	Description
control finger	Management	Multi-axis management function	Multi-axis setting, monitoring, reset

Please follow the steps below to add the positioning axis: Right-click **EtherCAT_Master_SoftMotion** in the device tree > **Add Device** > **Search ASDA-A2** > **ASDA-A2-E_DML**.



Perform basic positioning control such as absolute positioning, relative positioning... and other functions through EtherCAT communication. The positioning axis needs to be matched with the DL_MotionControlLight library.

Positioning Axis General Setting page



- **Axis Type and Limits**

Name	Features
Linear Axis/Rotary Axis	Set linear axis or rotary axis

- **Linear Axis Software Limits**

Name	Features
Activated	Start the software limit (only support linear axis)
Negative [u]	Reverse software limit
Positive [u]	Forward software limit

- **Setting of Rotary Axis Modulo value**

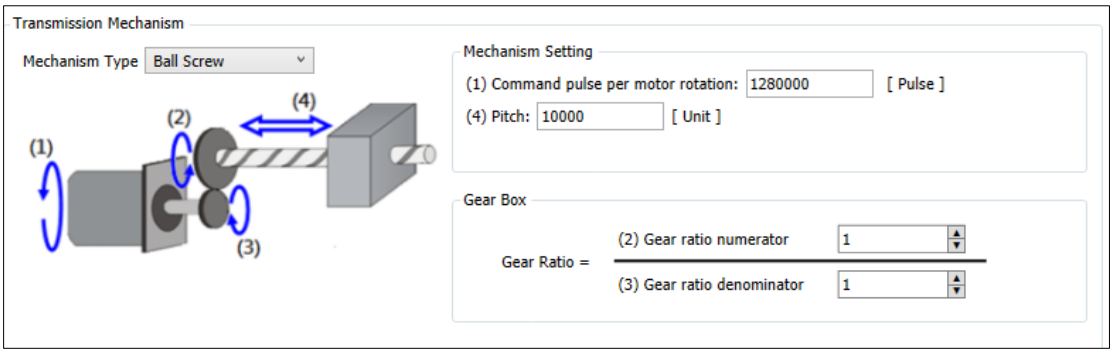
Name	Features
Modulo value [u]	Set a lap range (only support rotation axis)

- **Velocity Ramp Type**

Name	Features
Trapezoid	Set axis motion curve

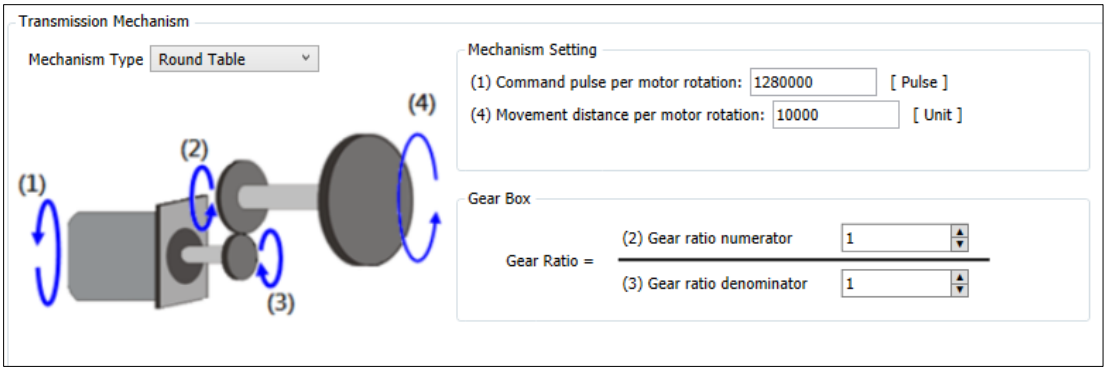
Transmission Mechanism

Ball Screw Drive



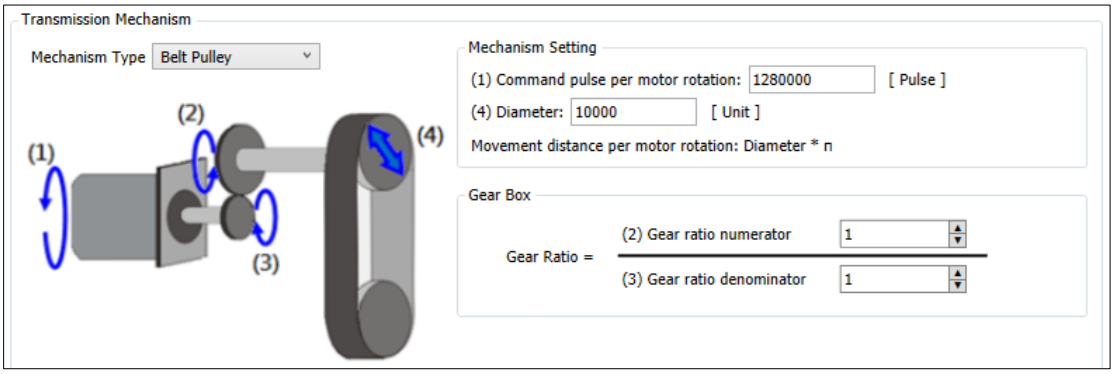
Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Pitch	Pitch

Round Table Transmission



Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Movement distance per motor rotation	The moving distance of the motor per rotation = diameter*n

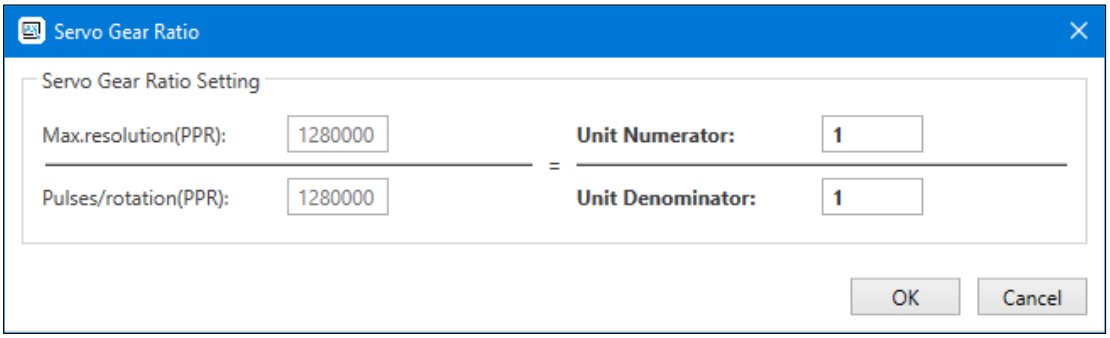
Belt Drive Gear Ratio Denominator



Belt drive	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator

Belt drive	Description
(4) Diameter (Movement distance per motor rotation: Diameter *n)	The Movement distance per motor rotation: Diameter *n

Servo Gear Ratio



Name	Features
Maximum resolution (PPR)	Maximum resolution (PPR)
Unit Numerator	Unit Numerator
Unit Denominator	Unit Denominator
Pulses/rotation (PPR)	Pulse/rotation (PPR)

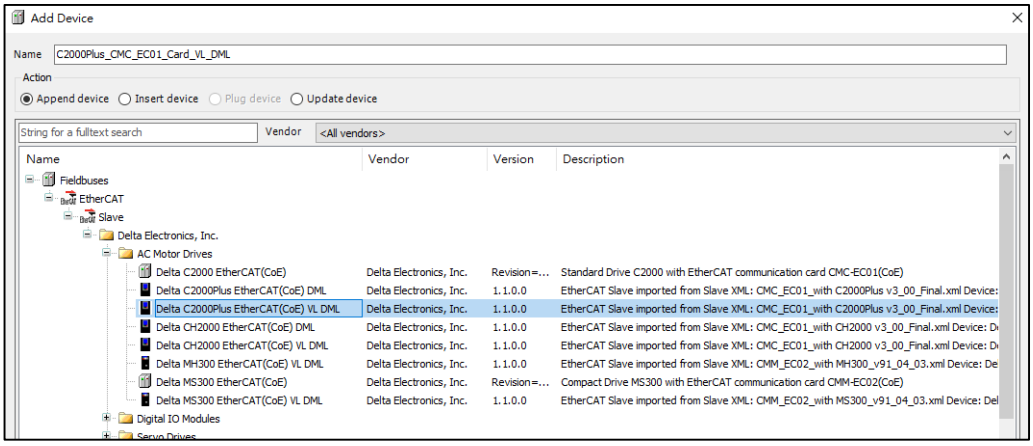
Positive/Negative Command

Name	Features
Reverse OFF/Reverse On	Set forward and reverse commands.

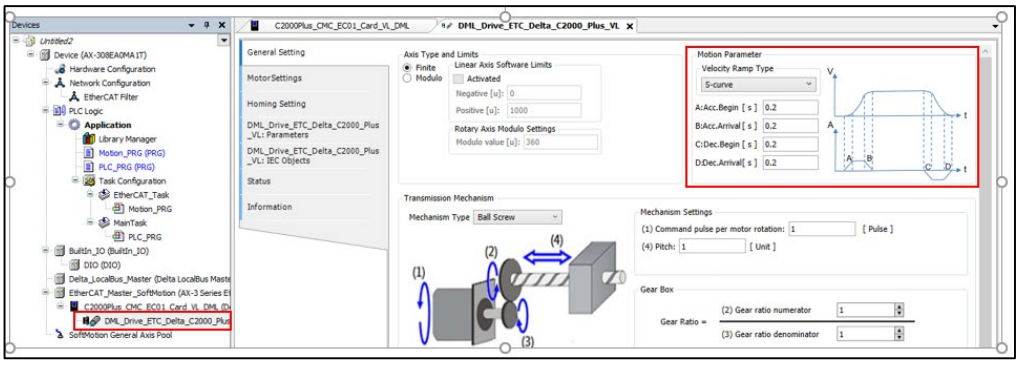
Speed Axis

Consider the following example of the newly added C2000Plus speed axis:

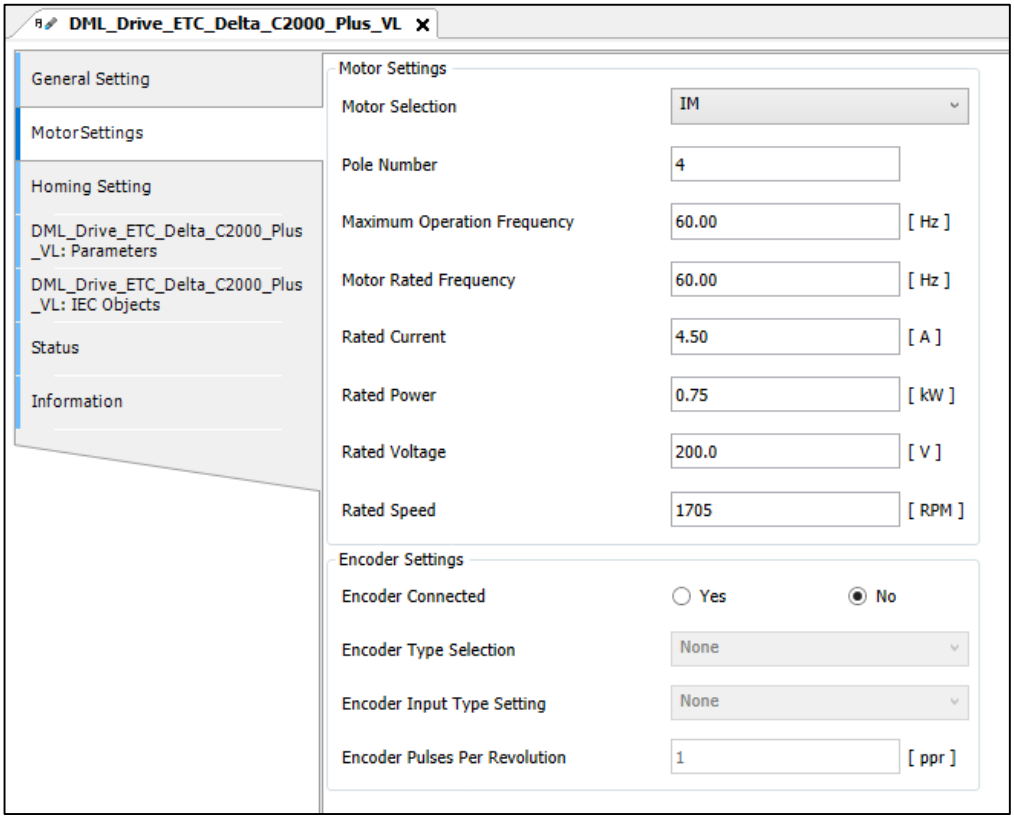
1. **Right-click** on EtherCAT_Master_Softmotion, select **Add Device**, and select Delta C2000Plus EtherCAT(CoE)VL DML.



2. Double-click DML_Drive_ETC_Delta_C2000_Plus_VL, select **General Settings**, and open the settings window.



3. For Motor Settings, the existing motor or encoder can be set.



Homing Setting

Please refer to the following mode/function table/device correspondence table:

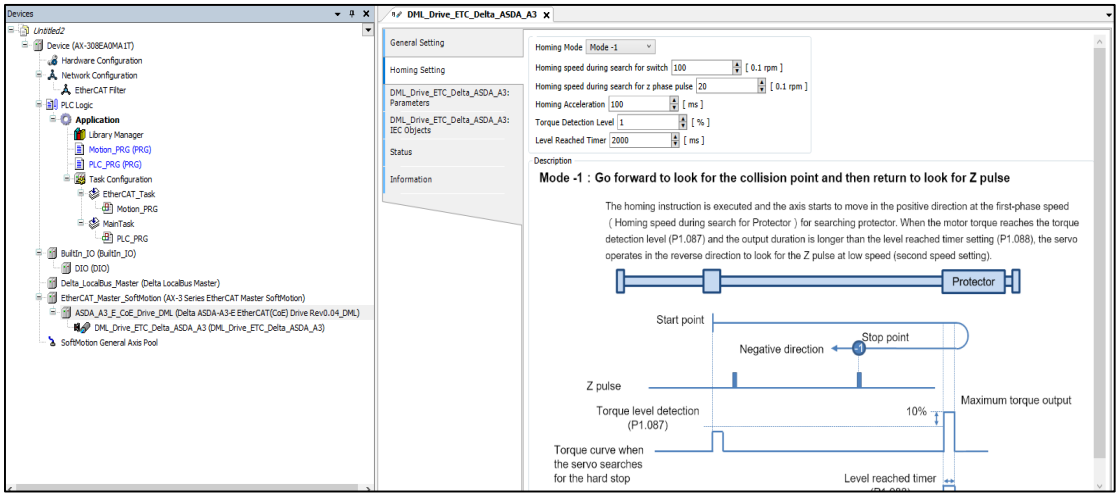
Homing Setting		Function					Device				
		Homing speed during search for switch	Homing speed during search for z phase switch	Homing acceleration	Detect torque level	Quasi-level arrival timing	ASDA-A2E	ASDA-A3E	ASD-B3E	C2000Plus	CH2000
-4	Reverse the direction to find the collision pad	V	V	V	V	V		V	V		
-3	Look for the collision pad in the forward direction	V	V	V	V	V		V	V		
-2	Reverse the direction to find the collision pad, then go back to find Z	V	V	V	V	V		V	V		
-1	Turn forward to find the collision pad, then go back to find Z	V	V	V	V	V		V	V		
1	Homing setting depending on negative limit switch and Z-phase pulse signal	V	V	V			V	V	V	V	V
2	Homing setting depending on positive limit switch and Z-phase pulse signal	V	V	V			V	V	V	V	V
3	Homing setting depending on home switch and Z-phase pulse signal	V	V	V			V	V	V	V	V
4		V	V	V			V	V	V	V	V
5		V	V	V			V	V	V	V	V
6		V	V	V			V	V	V	V	V
7	Homing setting depending on home switch and Z-phase pulse signal and positive limit switch signal	V	V	V			V	V	V	V	V
8		V	V	V			V	V	V	V	V
9		V	V	V			V	V	V	V	V
10		V	V	V			V	V	V	V	V
11	Homing setting depending on home switch, Z-phase pulse signal and negative limit switch signal	V	V	V			V	V	V	V	V
12		V	V	V			V	V	V	V	V
13		V	V	V			V	V	V	V	V
14		V	V	V			V	V	V	V	V
17	Homing setting depending on the negative limit switch is similar to mode 1 but does not	V	V	V			V	V	V	V	V

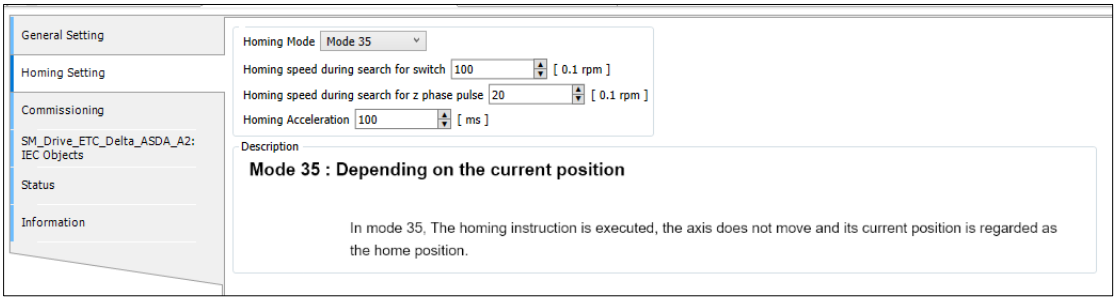
	consider the Z-phase pulse signal										
18	Homing setting depending on the positive limit switch is similar to mode 2 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
19	Homing setting depending on the home switch is similar to mode 3 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
20		V	V	V			V	V	V	V	V
21	Homing setting depending on the home switch is similar to mode 5 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
22		V	V	V			V	V	V	V	V
23	Homing setting depending on the home switch and the positive limit switch signal is similar to the mode 7 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
24		V	V	V			V	V	V	V	V
25	Homing setting depending on the home switch and the positive limit switch signal is similar to the mode 10 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
26		V	V	V			V	V	V	V	V
27	Homing setting depending on the home switch and the negative limit switch signal is similar to the mode 11 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
28		V	V	V			V	V	V	V	V
29	Homing setting depending on the home switch and the negative limit switch signal is similar to the mode 14 but does not consider the Z-phase pulse signal	V	V	V			V	V	V	V	V
30		V	V	V			V	V	V	V	V
33	Homing setting (backward direction) that only requires Z pulse	V	V	V			V	V	V	V	V
34	Homing setting (forward) that only requires Z pulse	V	V	V			V	V	V	V	V
35	Homing setting depending on the current position	V	V	V			V	V	V	V	V

Remarks: The Homing Setting -1~-4 is only applicable to ASDA-A3-E and ASDA-B3-E devices.

Follow the steps below to enable the Homing Setting:

Right-click on **EtherCAT_Master_SoftMotion** and click **Add Device**. After selecting the desired device version **ASD_A3_E_CoE_Drive**, double-click **DML_Drive_ETC_Delta_ASD_A3**, select the **Homing Setting** to open **Homing Setting** window.





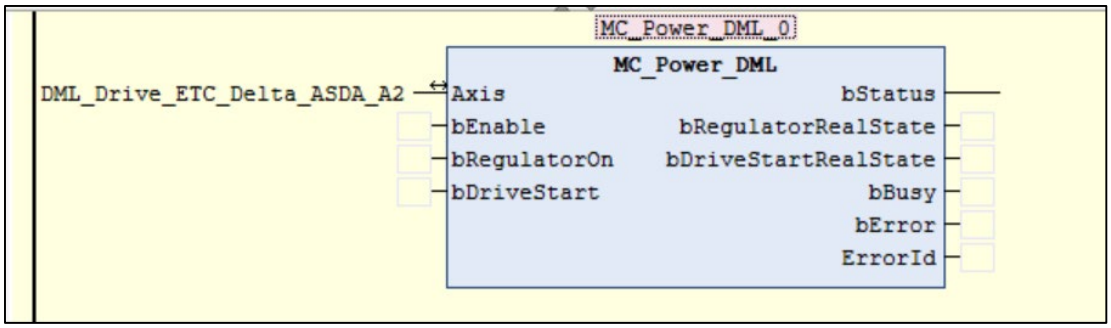
Name	Features
Homing mode	Set Homing mode
Homing speed during search for switch	Set the speed during search for switch
Homing speed during search for z phase switch	Set Homing speed during search for z phase switch
Homing acceleration	Set Homing acceleration

Use axis in the program

After selecting the DML servo device name and adding it to the project, the system will automatically generate its positioning axis name (the user can also change its name). The servo axis name generated by the positioning axis is **DML_Drive_ETC_Delta_ASDA_A2**.

When using the motion function block, if the user needs to input the corresponding axis, user can input DML_Drive_ETC_Delta_ASDA_A2 in the axis pin. As shown below.

Therefore, when using the motion function block, if the user needs to input the corresponding axis, they can input **DML_Drive_ETC_Delta_ASDA_A2** in the axis pin as shown below.

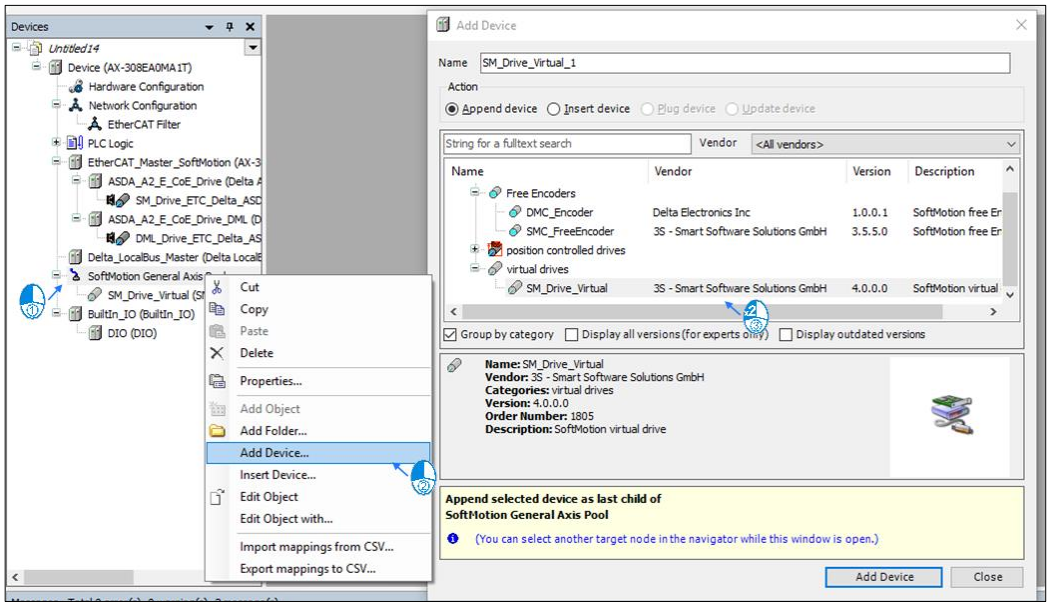


5.4.3 Virtual Drive

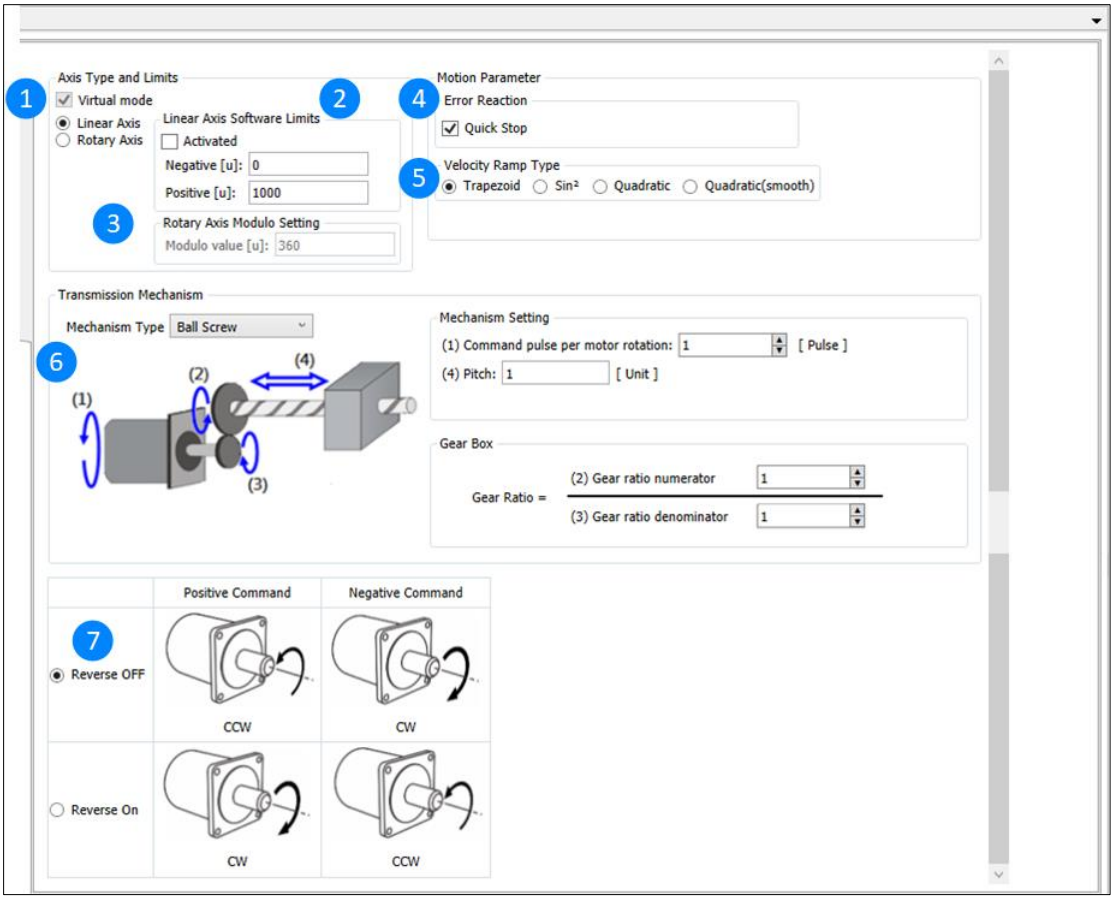
A virtual drive is an analog drive in the software. User can test programs or implement extended functions without connecting hardware, such as axis motion control and cam driver control. The virtual drive is not connected to the physical servo and is only used in the program to execute motion control commands.

Add a virtual drive to the project tree.

1. Right-click on **SoftMotion General Axis Pool** > **Add Device** > **Virtual Drive** > **SM_Drive_Virtual**.



Virtual Axis Page



1. **Axis Type and Limits**

Name	Description
Virtual mode	Enable virtual mode after clicking, which is enabled by default.
Linear axis/Rotary axis	Select Linear axis/Rotary axis

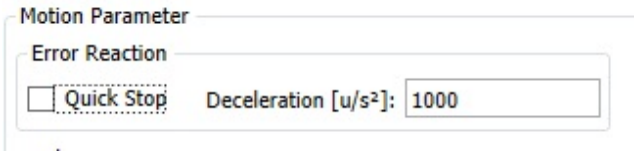
2. **Linear Axis Software Limits**

Name	Description
Activated	Start software limit (only support linear axis)
Negative [u]	Reverse software limit
Positive [u]	Forward software limit

3. **Rotary Axis modulo Setting**

Name	Description
Modulo value [u]	Set a lap range (only support Optional rotation axis)

4. **Error Reaction**

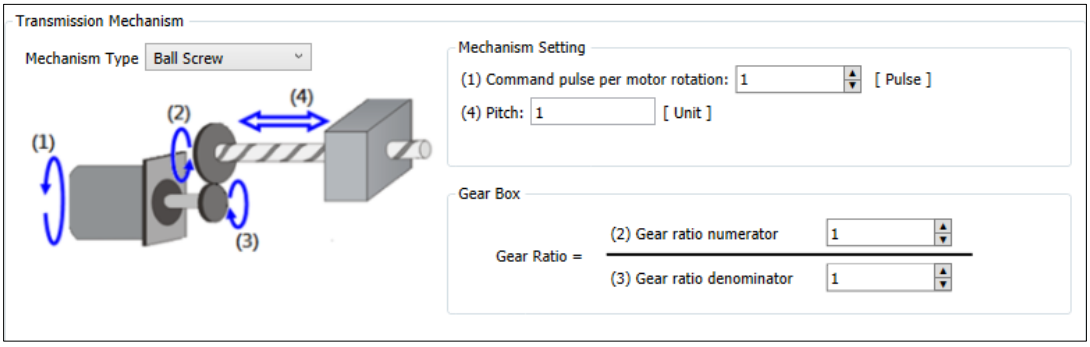
Name	Description
Quick Stop	Axis emergency stop
Deceleration	When the quick stop is unchecked, the deceleration option is displayed. 

5. **Velocity Ramp Type**

Name	Description
Trapezoid/Sin²/ Quadratic/ Quadratic (smooth)	Set axis motion curve

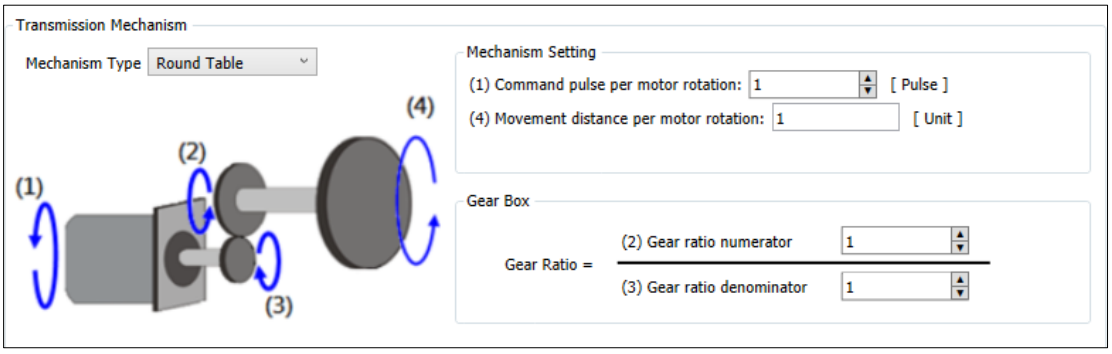
6. Transmission Mechanism

Ball screw drive



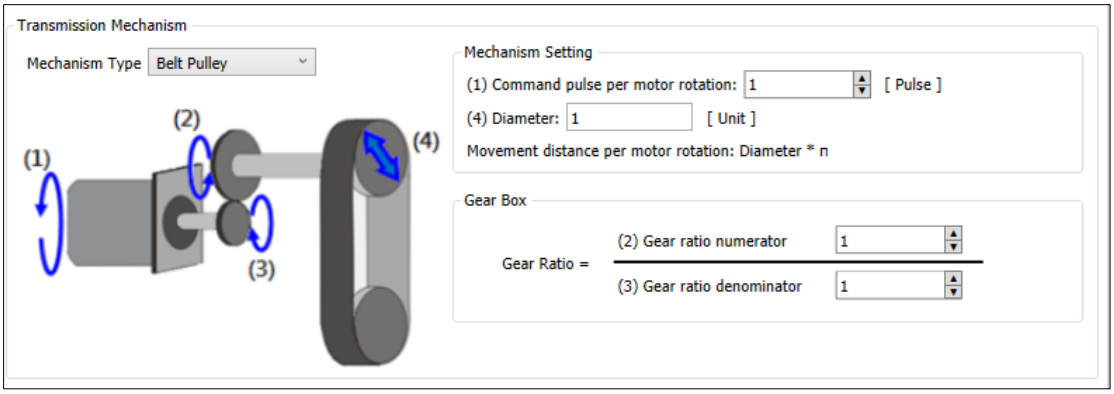
Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Pitch	Pitch

Round table drive



Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Movement distance per motor rotation	Movement distance per motor rotation = diameter*n

Belt drive



Name	Description
(1) Command pulse per motor rotation	Command pulse per motor rotation
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Movement distance per motor rotation: Diameter *n	Movement distance per motor rotation: Diameter *n

7. Positive/Negative Commands

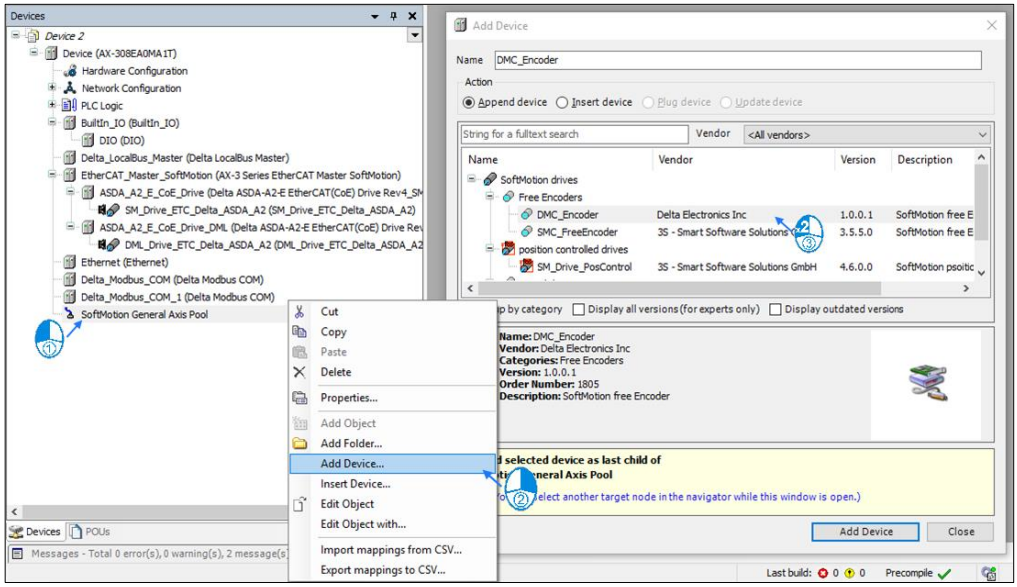
Name	Description
Reverse OFF/Reverse On	Set forward/reverse command

5.4.4 Free Encoder

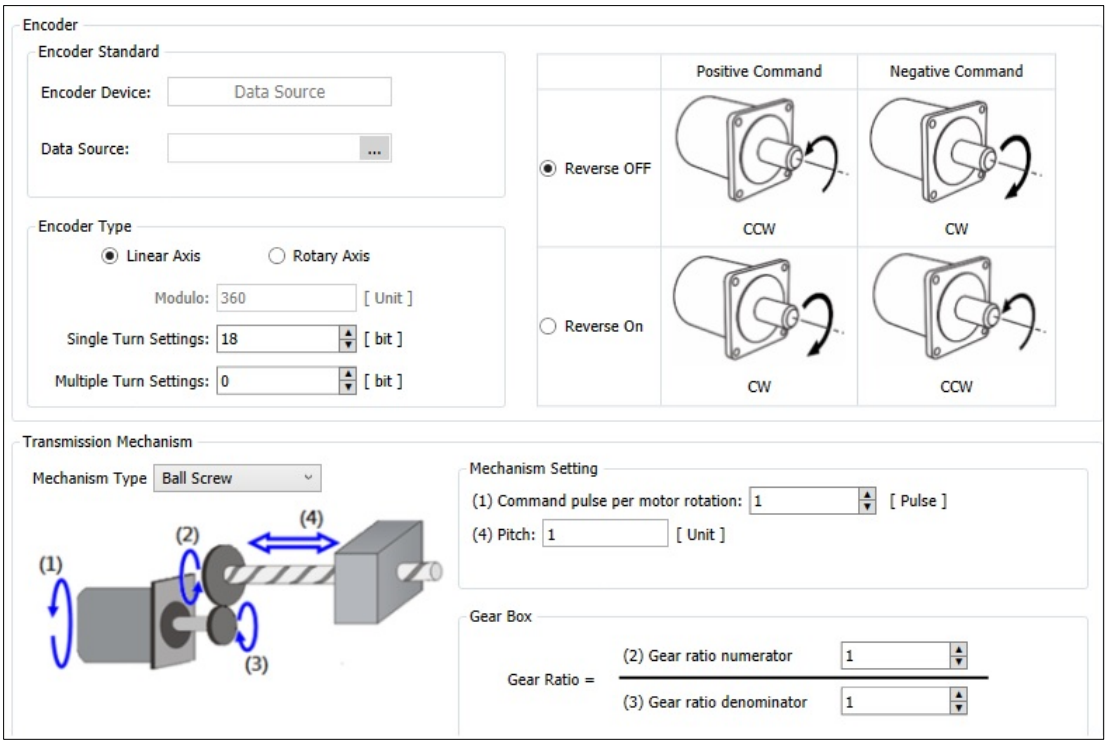
The free encoder function is used to set the connection with the external encoder electronic signal source, which is commonly used for precise positioning during motion control.

Please follow the steps below to add a free encoder to the project tree:

1. Right-click on **SoftMotion General Axis Pool** > **Add Device** > **Free Encoder** > **DMC_Encoder**.

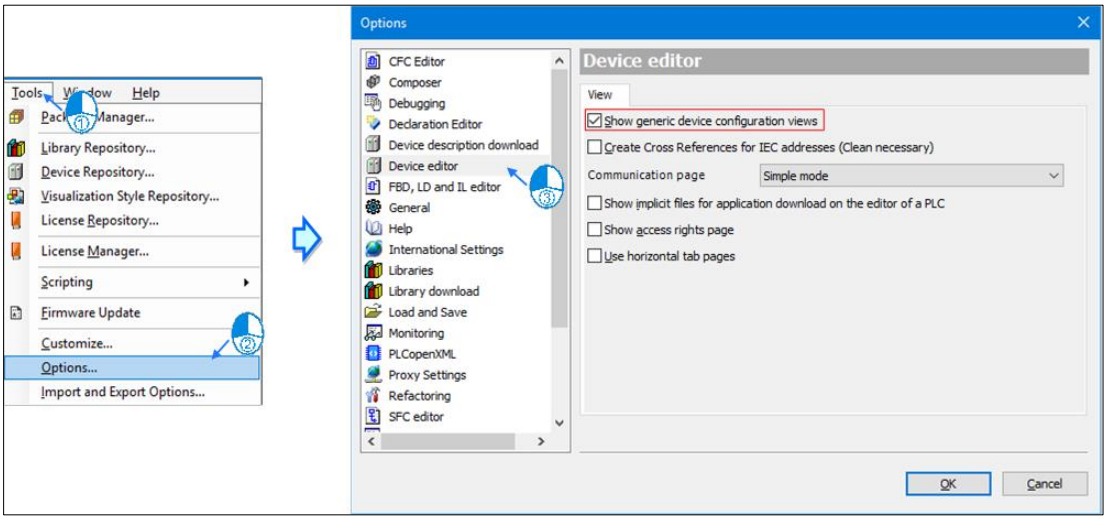


The following figure shows the default display screen after double-clicking DMC_Encoder in the project tree:

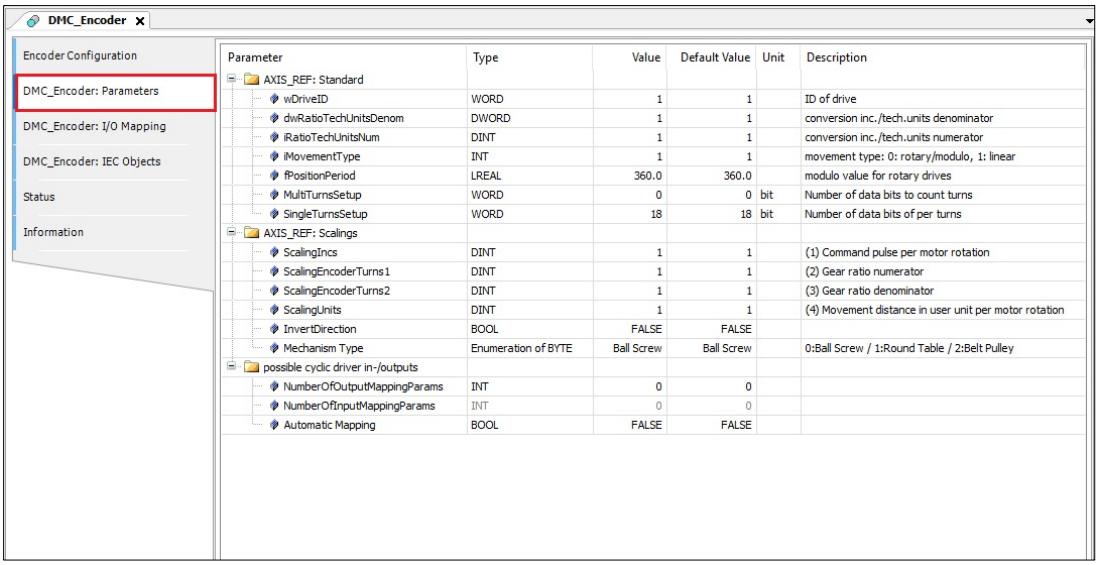


DMX_Encoder Parameters Tab

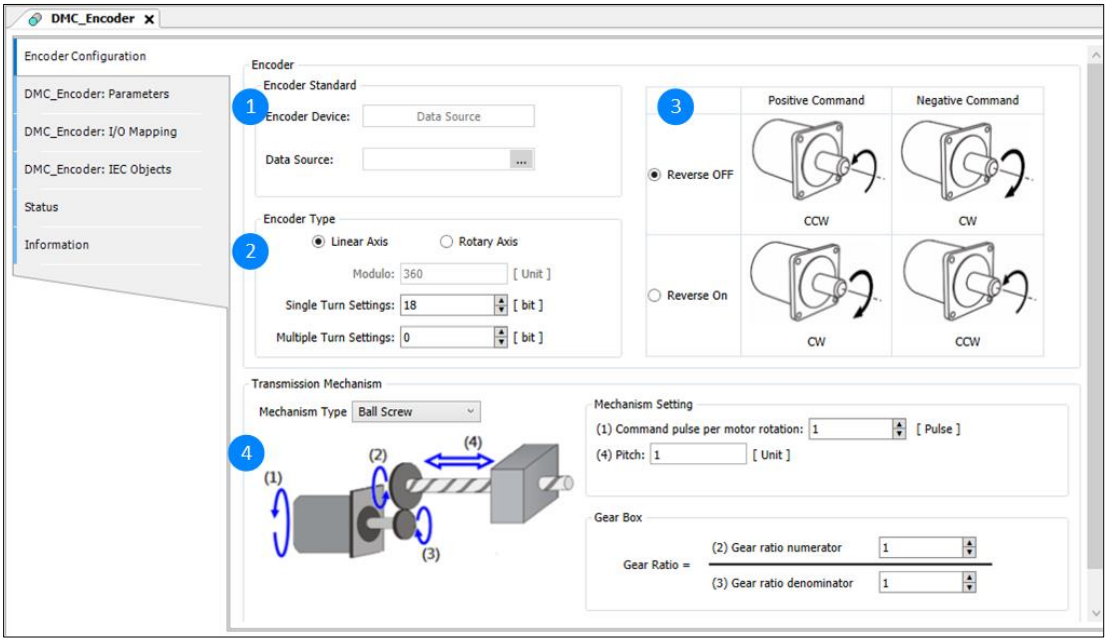
To add the DMC_Encoder parameter setting tab, please go to the main menu **Tools** > **Options** > **Device Editor** > **Select Show generic device configuration views**.



The user can set relevant parameters in the DMC_Encoder parameter tab.



Encoder Configuration Tab



① Encoder Standard

Name	Description
Encoder equipment	Data Source
Data Source	Support variables whose data type is UDINT. Data source Click ... to enter the input assistant, select a variable under Motion_PRG, and press **OK** to complete the data source selection.

② Encoder Type

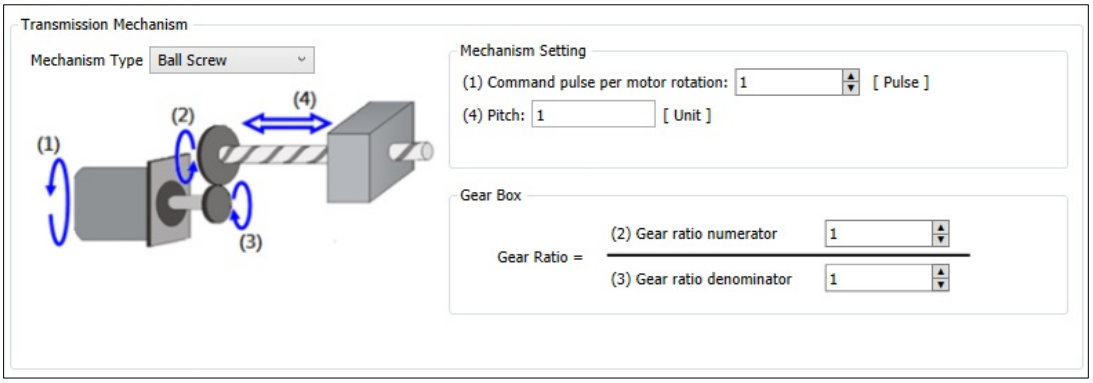
Name	Description
Linear axis/rotary axis	Select linear axis/rotary axis
Modulus value	Set a circle range (only support rotation axis)
Single lap setting	Data bit of single revolution (unit: bit)
Multi-turn settings	Multi-turn data bits (unit: bit)

③ Positive/Negative Instructions

Name	Features
Reverse close/reverse open	Set forward and reverse commands

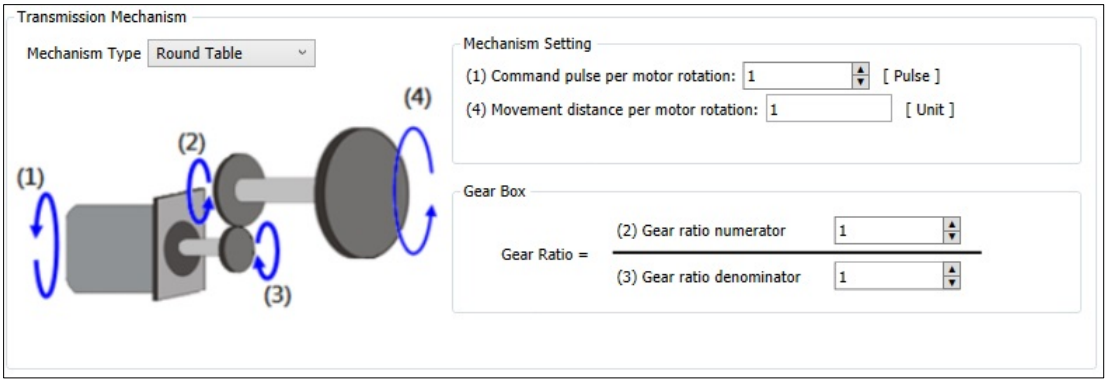
④ Transmission Mechanism

Ball screw drive



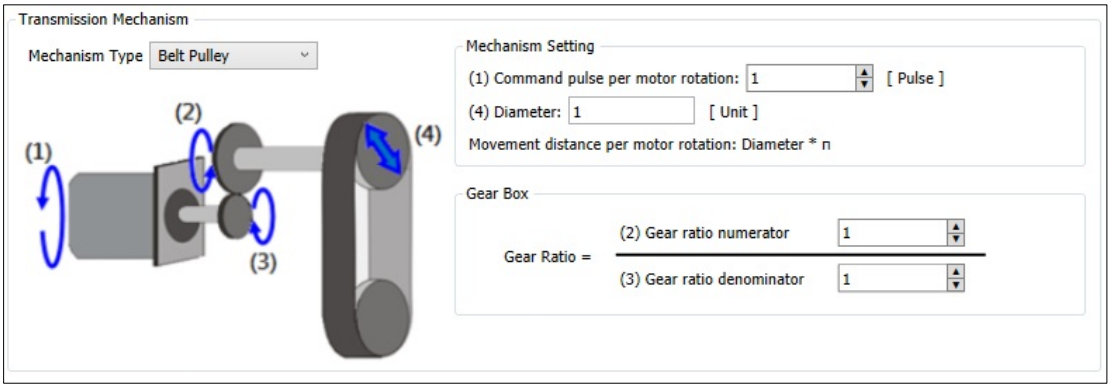
Name	Description
(1) Command pulse per motor revolution	Command pulse per motor revolution
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Pitch	Pitch

Round table drive



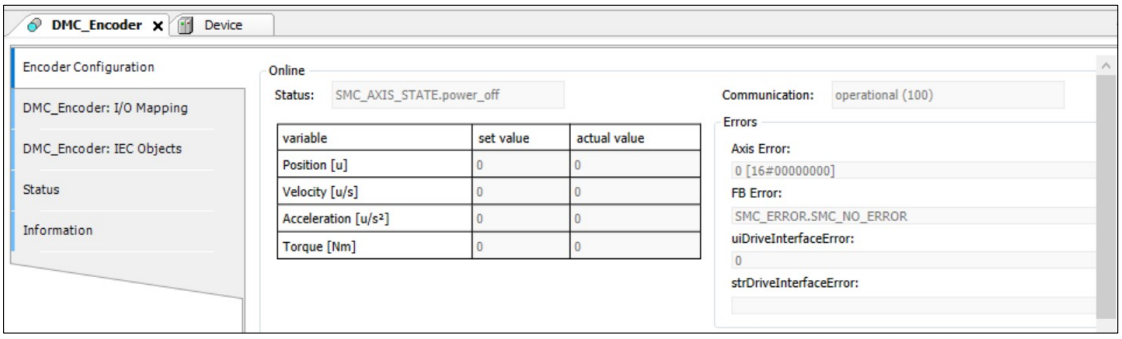
Name	Description
(1) Command pulse per motor revolution	Command pulse per motor revolution
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) The moving distance of the motor per revolution	Movement distance per motor revolution

Belt drive



Name	Description
(1) Command pulse per motor revolution	Command pulse per motor revolution
(2) Gear ratio numerator	Gear ratio numerator
(3) Gear ratio denominator	Gear ratio denominator
(4) Diameter	Moving distance per motor revolution = diameter*n

When the device is connected, online monitoring information will be added to the encoder configuration tab, as shown in the following figure.



5.4.5 Axis group setting

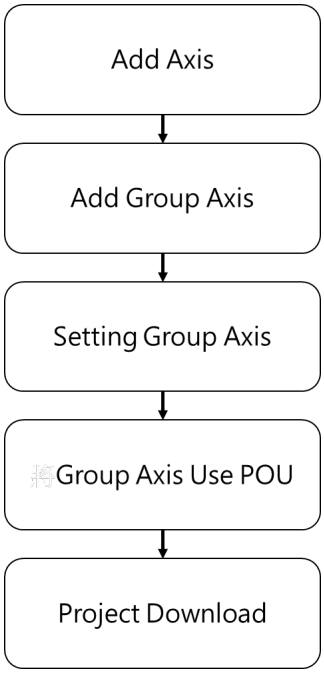
The axis group defines the mechanism relationship between the axes, co-locating and orienting multiple axes in space to match the motion function to be used. When multiple axes are executed at the same time and linear interpolation and helical interpolation functions are required to use the axis group movement, the axis needs to be set as the axis group through DIADesigner-AX.

Number of control axes	Maximum number of axes controlled by linear interpolation	6 axes
	Number of control axes for arc interpolation	6 Axes (3 axes following axes)

Applicable devicet: AX-308E, AX-364E.

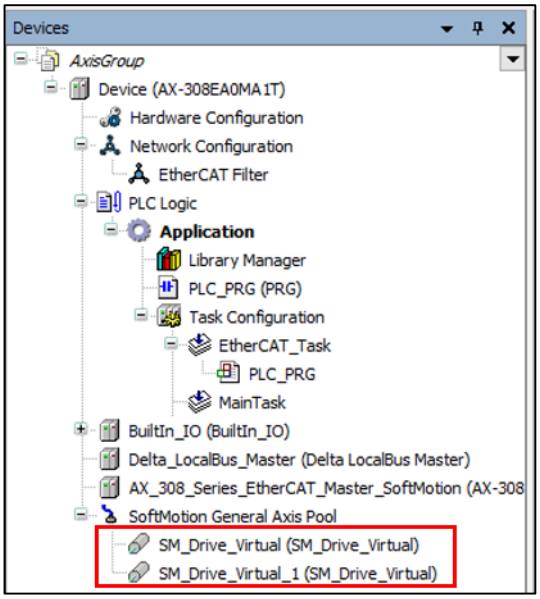
5.4.5.1 Axis Group Parameter Introduction

To use the axis group movement, user need to set the axis group name and corresponding individual axis through DIADesigner-AX. The axis group creation flowchart is as follows:



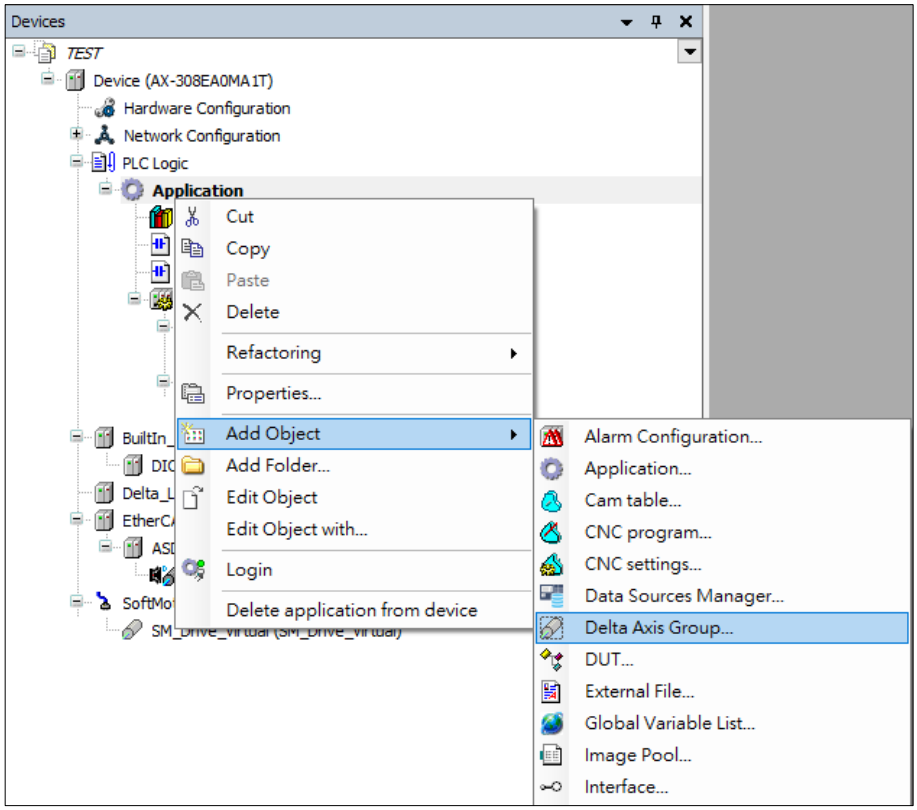
Follow the steps to create an axis group:

1. Add a single axis, this process takes the creation of 2 virtual axes as an example.

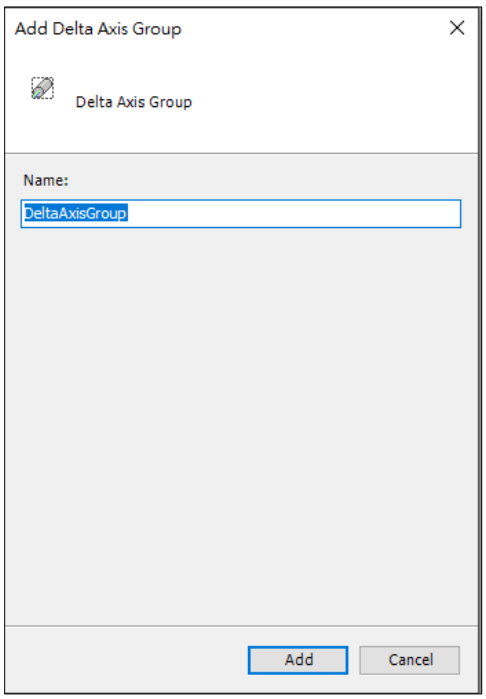


2. After the creation is complete, right-click on the **Application** and select **Add**

Object > **Delta Axis Group** as shown in the following figure.

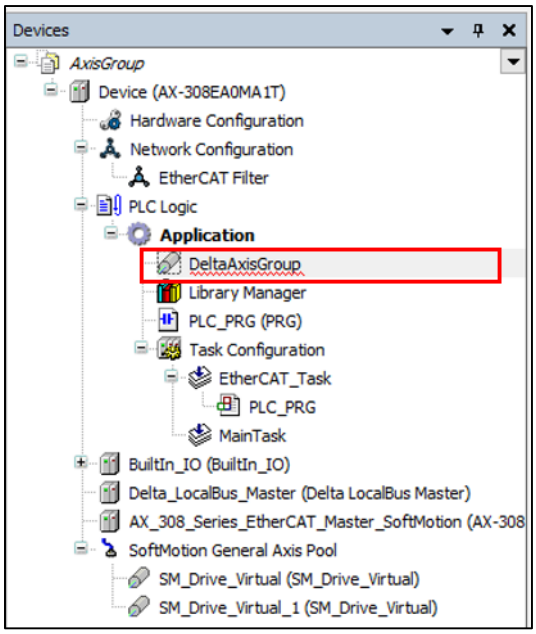


Result: Opens **Add Delta Axis Group** window.

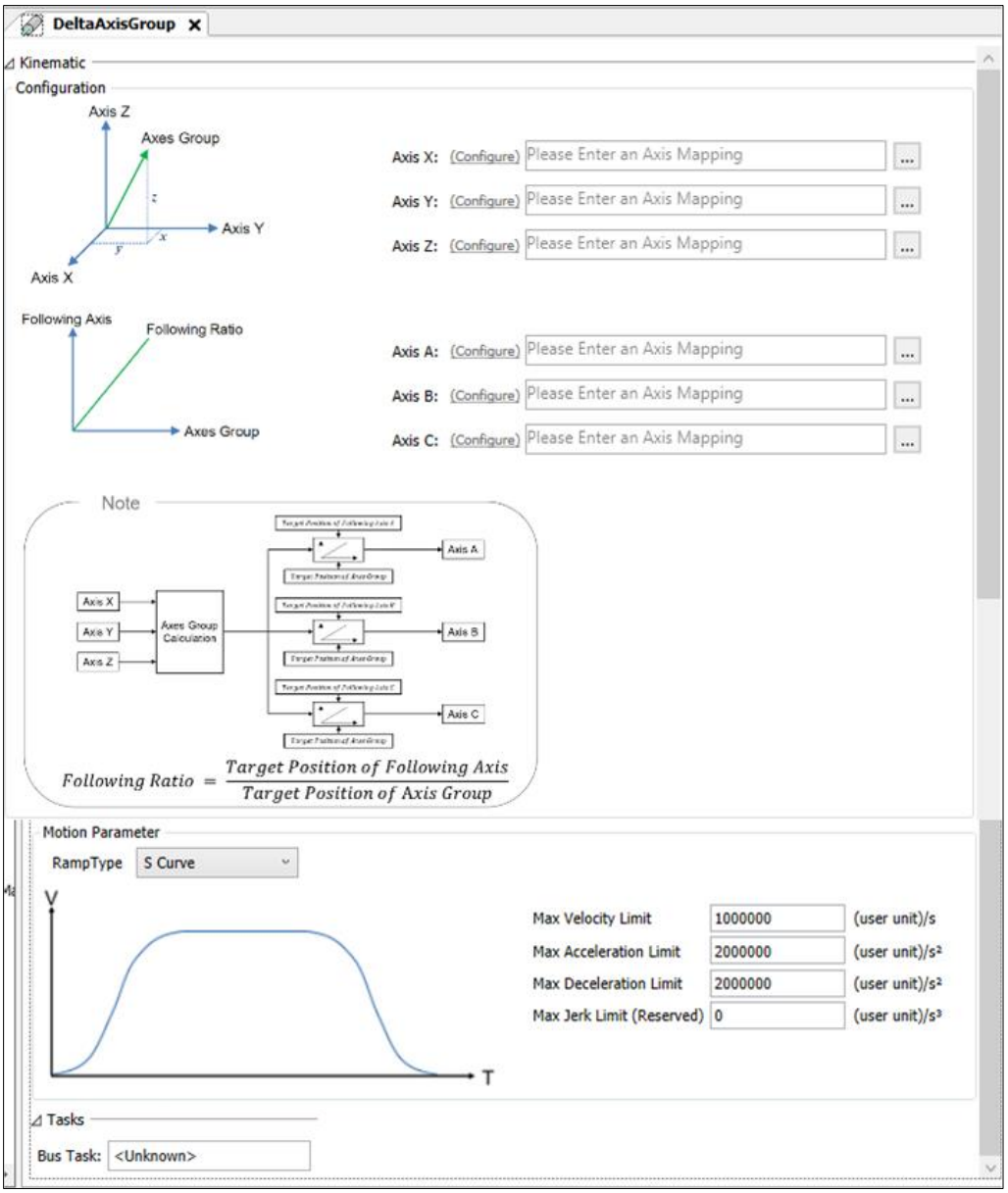


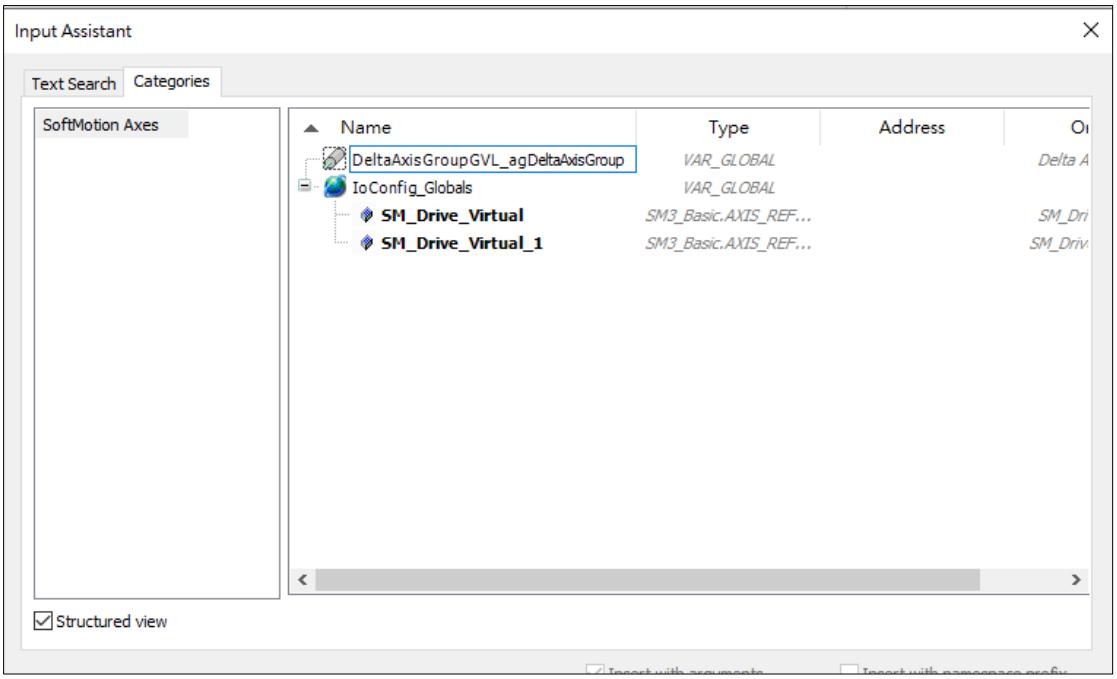
3. In the **Add Delta Axis Group** window, enter name and click **Add** button.

Result: The **DeltaAxisGroup** appears on the project tree.

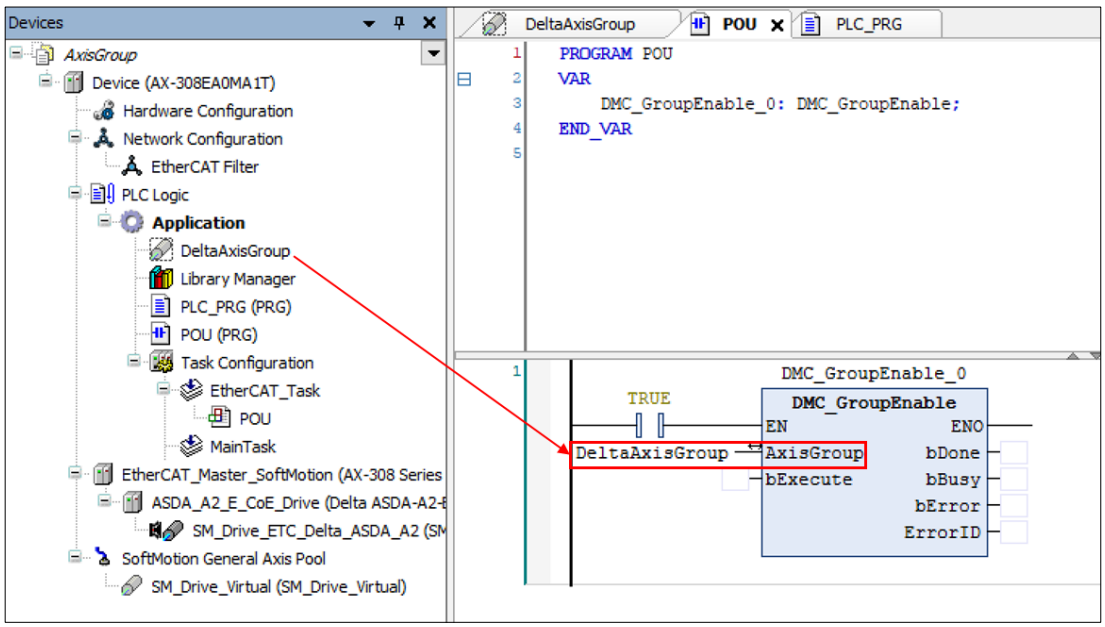


4. Click on **DeltaAxisGroup**, then click on to enter the **input assistant** and enter the names of the two-axis virtual axis into **Axis X** and **Axis Y**.

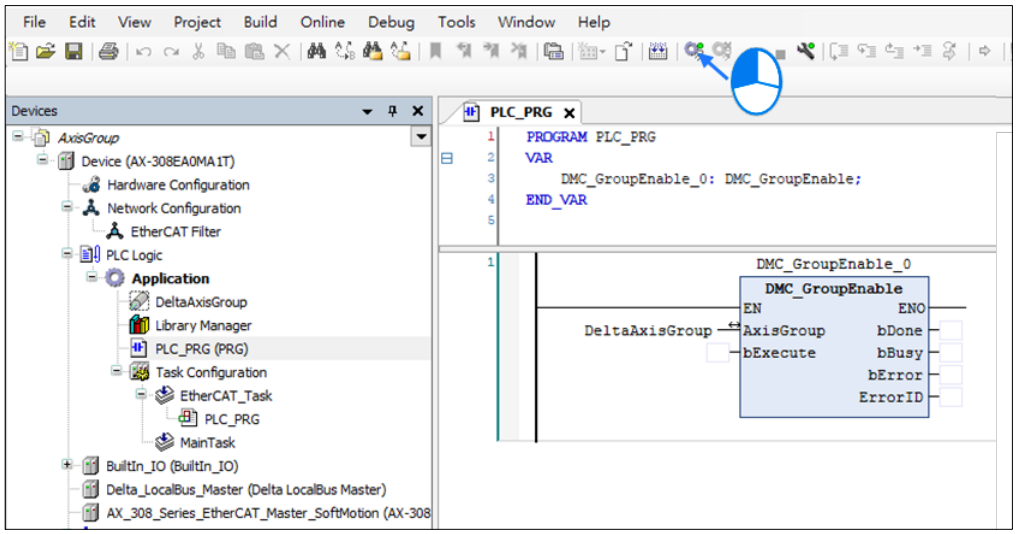

DeltaAxisGroup
Kinematic
Configuration
Axis Z
Axes Group
Axis Y
Axis X
Axis X: (Configure) Please Enter an Axis Mapping
Axis Y: (Configure) Please Enter an Axis Mapping
Axis Z: (Configure) Please Enter an Axis Mapping
Following Axis
Following Ratio
Axes Group
Axis A: (Configure) Please Enter an Axis Mapping
Axis B: (Configure) Please Enter an Axis Mapping
Axis C: (Configure) Please Enter an Axis Mapping
Note
Axis X
Axis Y
Axis Z
Axes Group Calculation
Axis A
Axis B
Axis C
Following Ratio = Target Position of Following Axis / Target Position of Axis Group
Motion Parameter
RampType S Curve
V
T
Max Velocity Limit 1000000 (user unit)/s
Max Acceleration Limit 2000000 (user unit)/s²
Max Deceleration Limit 2000000 (user unit)/s²
Max Jerk Limit (Reserved) 0 (user unit)/s³
Tasks
Bus Task: <Unknown>



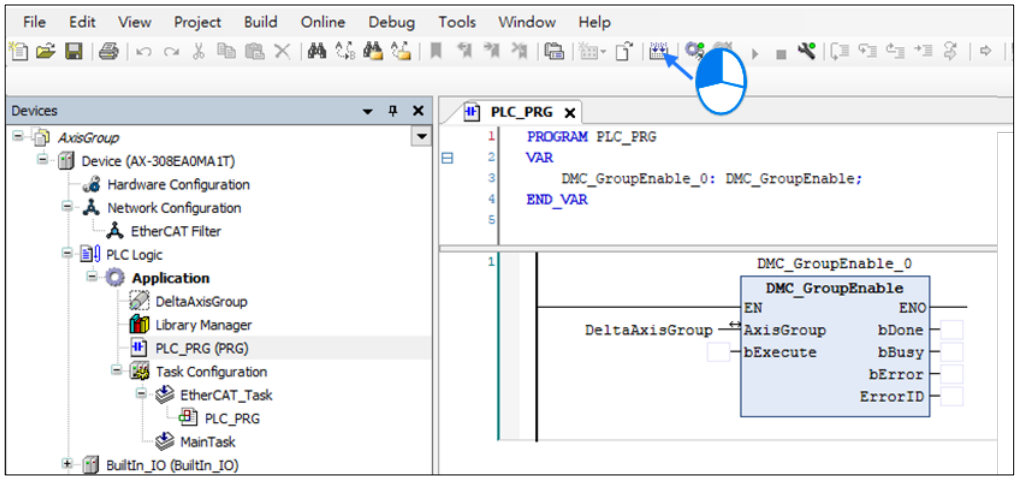
5. Add the **DMC_GroupEnable** function block in Motion_PRG and fill in the axis group name in the AxisGroup pin.



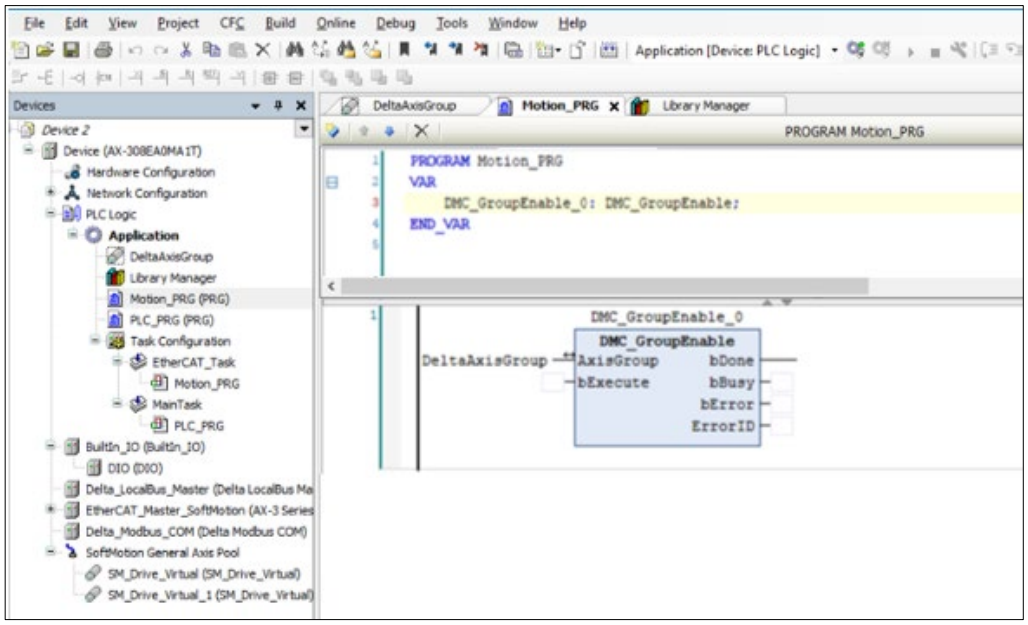
6. When the program is complete, click compile and confirm whether there are errors.



7. After confirming that there are no errors after compiling, click Login icon to download the program.

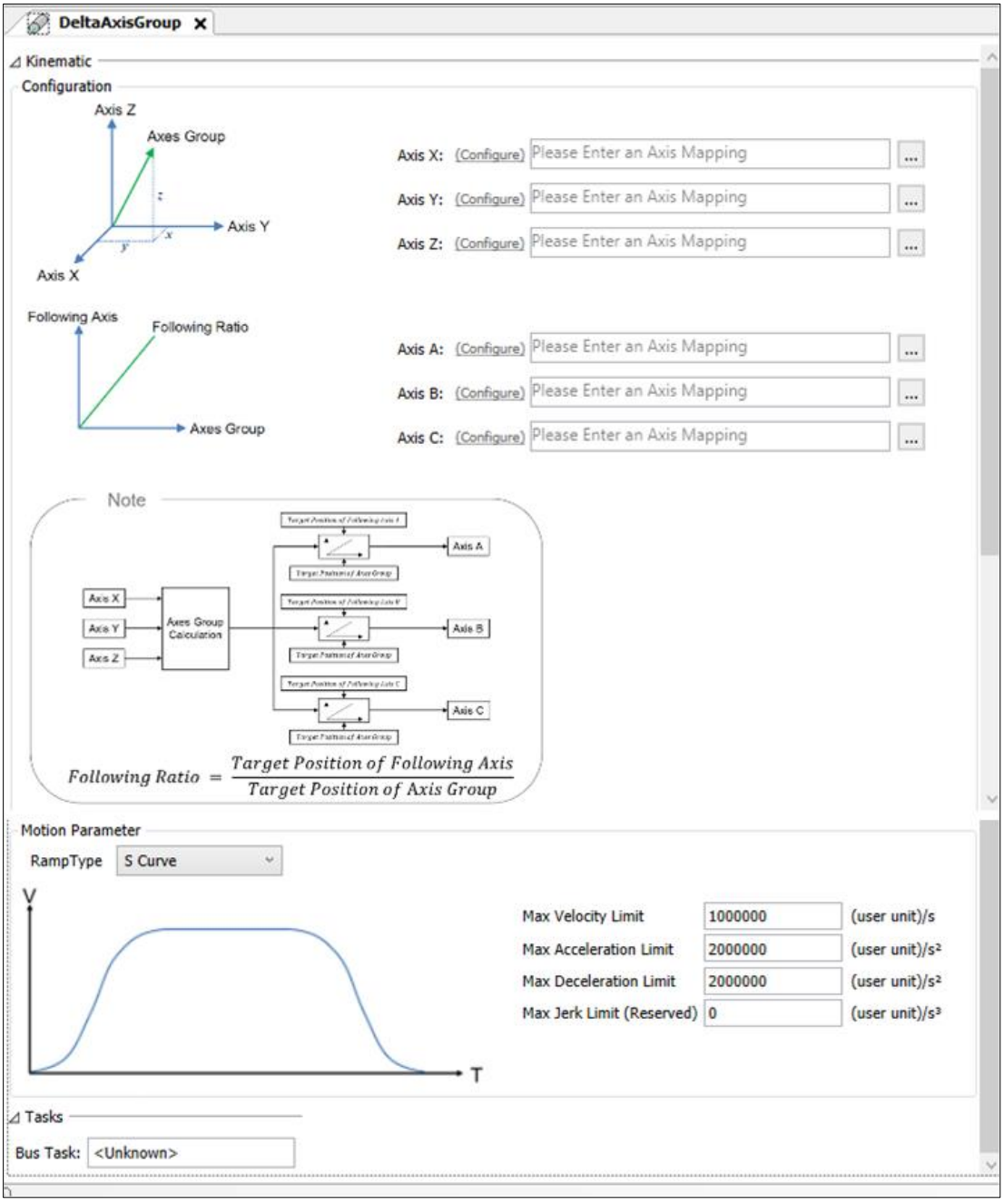


8. After confirming that there are no errors after compiling, click Login to download the program.


File Edit View Project CFC Build Online Debug Tools Window Help
Application [Device: PLC Logic]
Devices
Device 2
Device (AX-308EA0MA1T)
Hardware Configuration
Network Configuration
PLC Logic
Application
DeltaAxisGroup
Library Manager
Motion_PRG (PRG)
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task
Motion_PRG
MainTask
PLC_PRG
BuiltIn_IO (BuiltIn_IO)
DIO (DIO)
Delta_LocalBus_Master (Delta LocalBus Ma
EtherCAT_Master_SoftMotion (AX-3 Series
Delta_Modbus_COM (Delta Modbus COM)
SoftMotion General Axis Pool
SM_Drive_Virtual (SM_Drive_Virtual)
SM_Drive_Virtual_1 (SM_Drive_Virtual)
DeltaAxisGroup
Motion_PRG
Library Manager
PROGRAM Motion_PRG
PROGRAM Motion_PRG
VAR
DMC_GroupEnable_0: DMC_GroupEnable;
END_VAR
DMC_GroupEnable_0
DMC_GroupEnable
DeltaAxisGroup
AxisGroup
bExecute
bDone
bBusy
bError
ErrorID

5.4.5.2 Axis group configuration process and setting steps

The parameters used for axis group motion are as follows:



Kinematic model

Name	Features
Axis X*1	X axis in axis group
Axis Y*1	Y axis in axis group
Axis Z*1	Z axis in axis group
Axis A*1	A axis in the axis group
Axis B*1	B axis in axis group
Axis C*1	C axis in axis group

Motion parameters

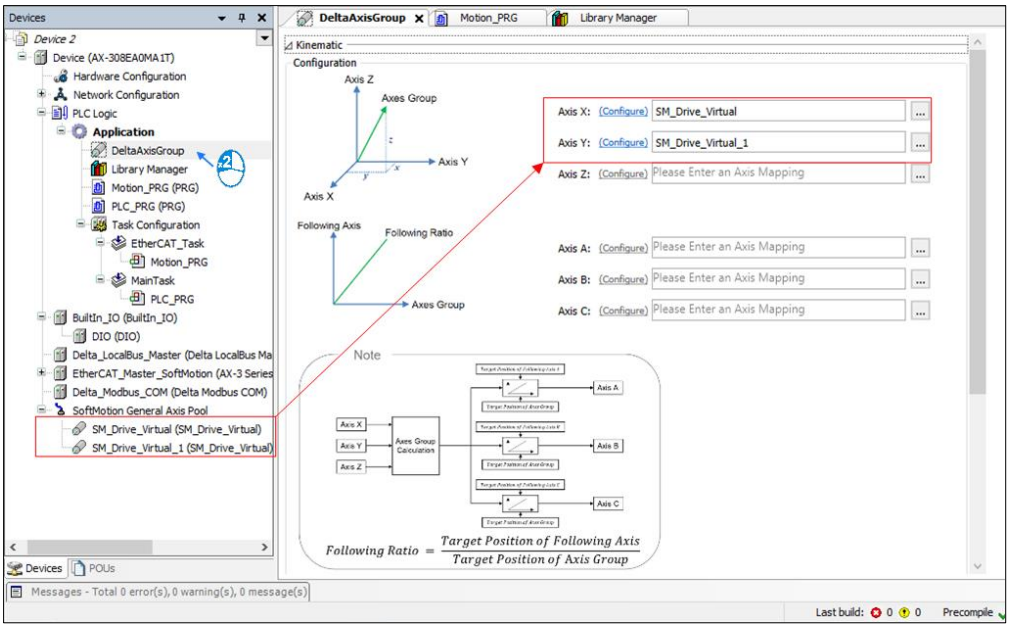
Name	Features
Slope type * 2	Velocity curve type
Maximum Velocity Limit * 3	Maximum Velocity of axis group
Maximum Acceleration Limit * 3	Maximum acceleration of axis group
Maximum Deceleration limit *3	Maximum deceleration of axis group
Maximum jerk Limit (Reserved) * 3	Maximum acceleration (jump) of axis group (this function is reserved)

Task

Name	Features
Bus task	Set axis group update Task

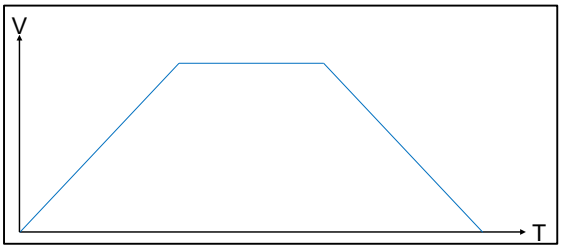
Note:

1. Axis X ~ Axis C：Enter single axis name

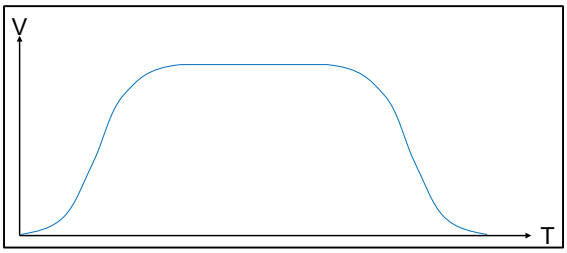


2. Slope type: There are two types of curves, trapezoid and S-curve, and the curves are shown as the following two types:

Trapezoid



S curve



- Maximum Velocity Limit: an error will be reported when the axis group movement velocity exceeds the set parameter.
- Maximum Acceleration Limit: an error will be reported when the axis group movement acceleration exceeds the set parameter.
- Maximum Deceleration Limit: when the axis group motion deceleration exceeds the set parameter, an error will be reported.

5.4.6 SoftMotion version compatibility

Softmotion v4.10 is supported in DIADesigner-AX version 1.4.0. Use the following table to check the version compatibility among SoftMotion, Axis devices, and related libraries. You can modify the version or update the device if any related incompatible errors exist.

DML Axis Device Version (DML or VLDML)	Supported Device								SoftMotion Version	Library	
	C2000+	CH2000	MH300	MS300	A2	A3	B3	R1-EC5621		DML_Drive_ETC_Delta_XX	DL_MotionControlLight
1.0.0.0					V	V	V		4.6.1.0	1.0.0.14	1.0.0.13
1.0.0.5					V		V	V	4.6.1.0	1.0.0.14	1.0.0.13
1.0.0.6						V			4.6.1.0	1.0.0.14	1.0.0.13
1.0.0.14					V	V	V	V	4.6.1.0	1.0.0.14	1.0.0.13
1.1.0.0	V	V	V	V	V	V	V	V	4.6.1.0	1.1.0.0	1.1.0.0
1.3.0.0	V	V	V	V	V	V	V	V	4.6.1.0 4.10.0.0	1.3.0.0	1.3.0.0

To update the Axis device version

1. On the **Devices** pane, right-click the Axis device, and then select **Update Device**.
2. In the **Update Device** dialog, select the latest version (for example, v1.3.0.0), and then click **Update Device**.
3. In the **Update Device List** dialog, check the Source and Target version, and then click **Update**.

To change the SoftMotion version

1. Go to **Project** > **Project Settings**.
2. Select **SoftMontion**.
3. On the **SoftMotion** pane, select the version, and then click **OK**.

To change the library version

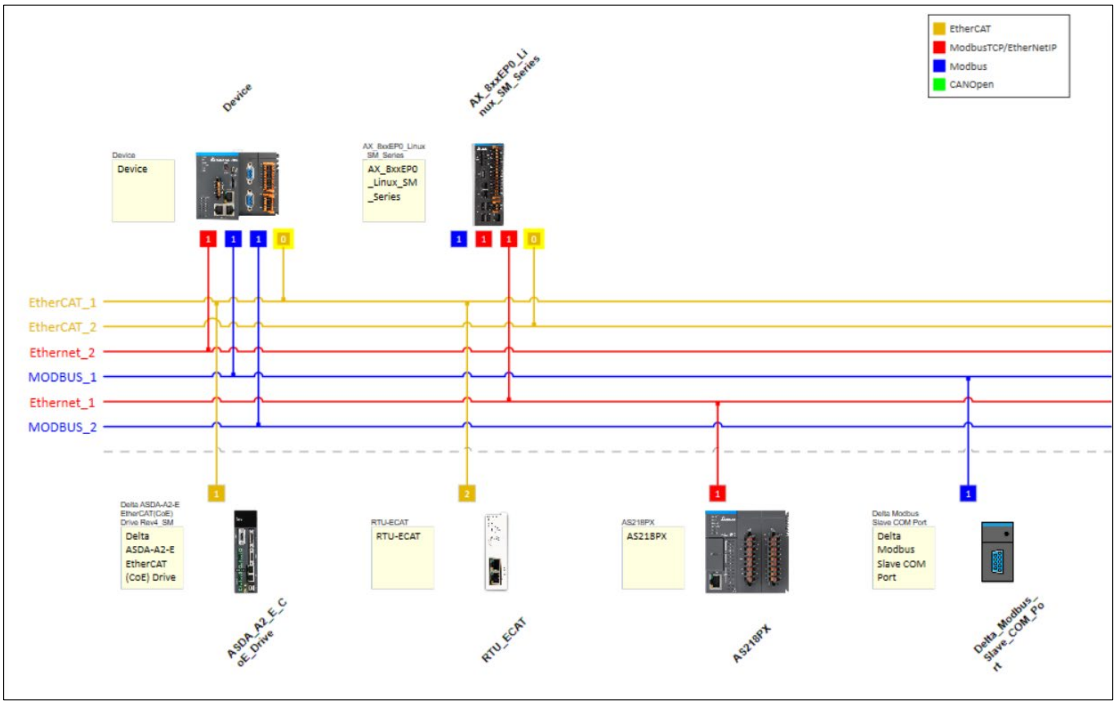
1. On the **Devices** pane, double-click **Library Manager**.
2. On the **Library Manager** pane, click **Placeholders**.
3. In the **Placeholders** dialog, select the library version, and then click **OK**.

Chapter 6: Network Configuration

6.1 Network Configuration

The Network Topology is an industrial network connected to the full range of Delta products, which can be used according to the planned network topology.

The **Network Configuration Editor** is the graphical representation of connections between devices that allows user to plan their networks.

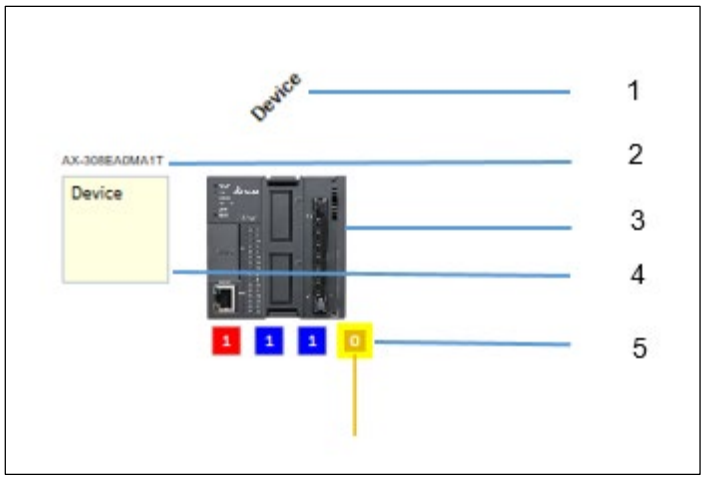


Network View Description

Network Configuration window displays two types of information:

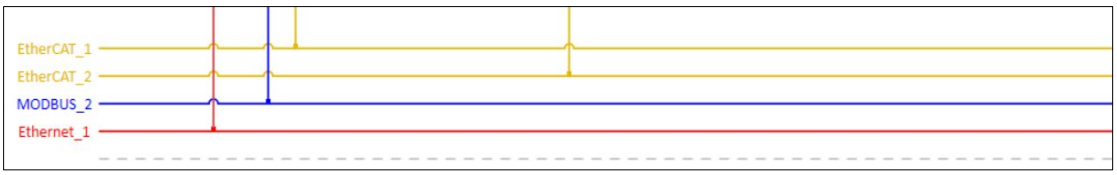
- Device information
- Network information

Device information includes the following:



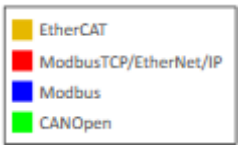
1. Device name
2. Part number
3. Device image
4. A Note to add text
5. Communication ports on the device with protocol support.

Network information includes the following:



1. Name of the Network
2. Network lines with Colors
3. Connections to the Devices

When the network connection is made between the communication ports in the controller and field devices, the network lines appear in the color described as shown in the following figure:

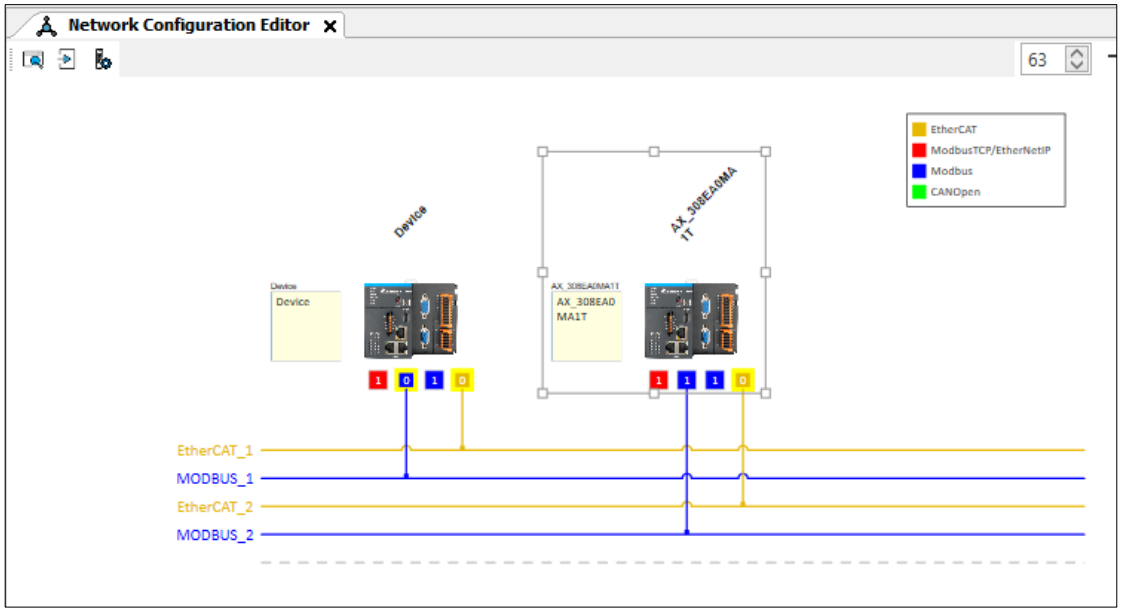


Network configuration diagram toolbar

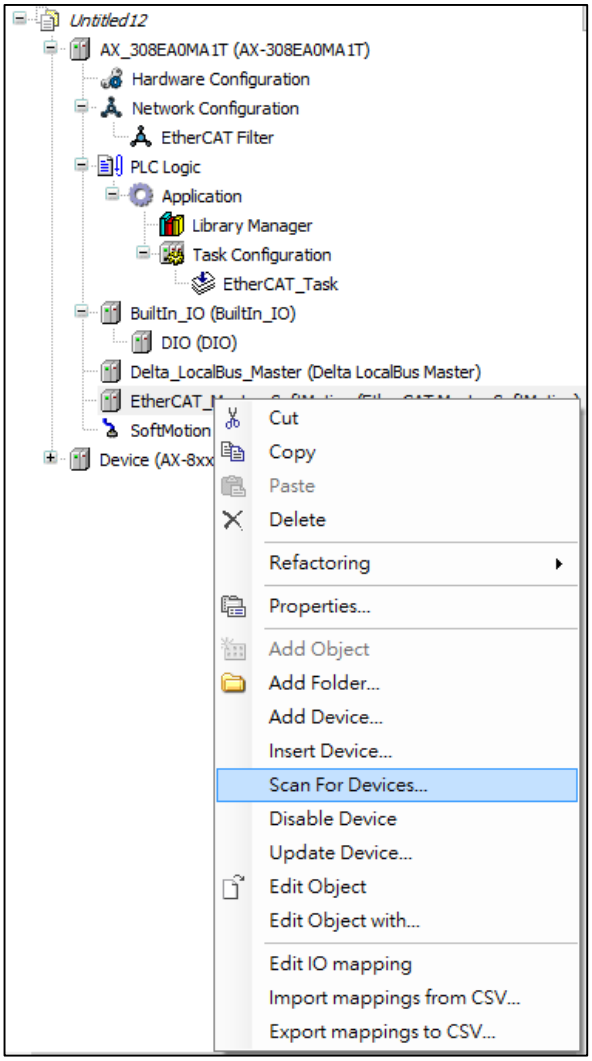
Toolbar Icon	Function
	Network Scan by PC

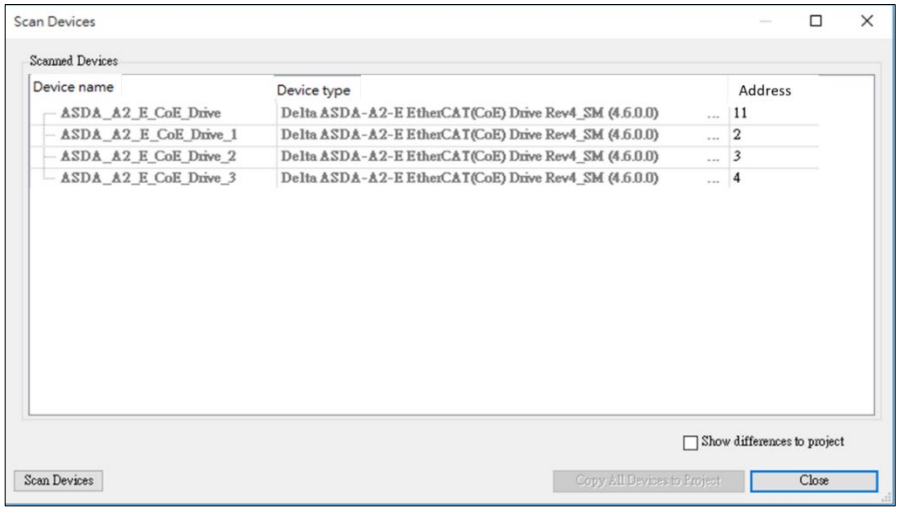
Network Scan

1. User can click  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 them into the project.



2. Right-click the EtherCAT node and choose **Scan For Devices** to scan the device under fieldbus.





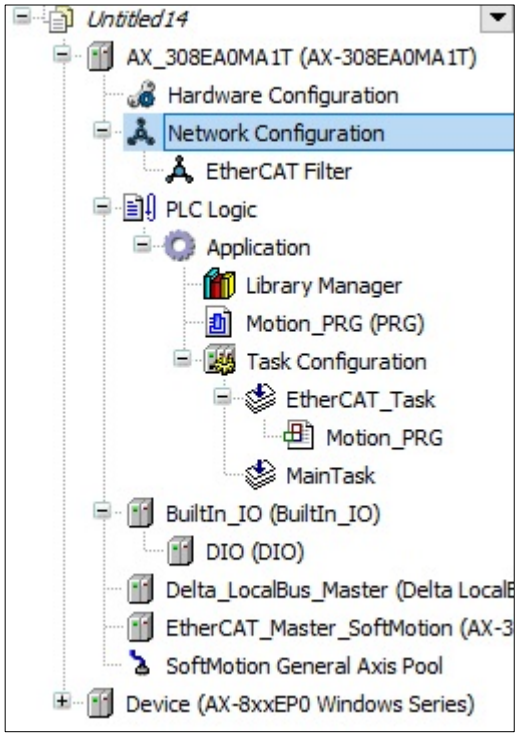
Note: Please LOGIN first to build the map DIADesigner-AX fieldbus scan.

6.2 Configuration

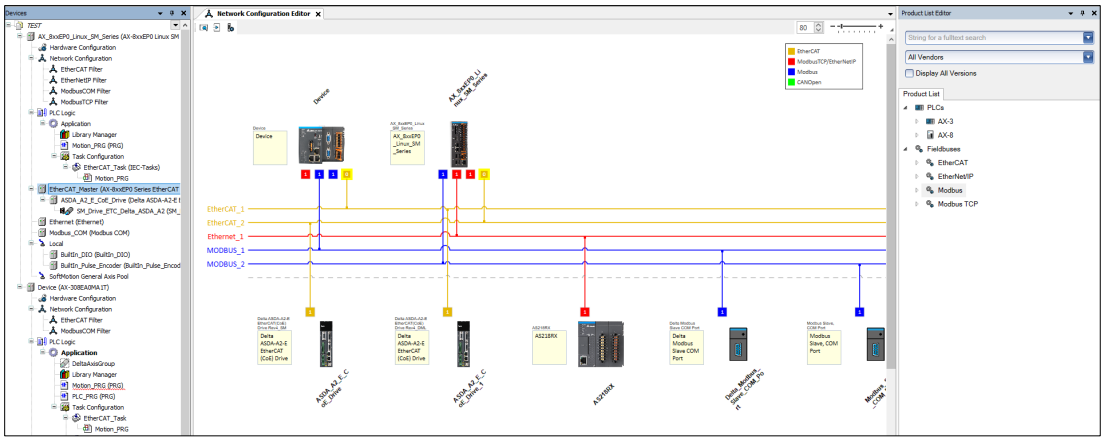
6.2.1 Create a network configuration diagram

Follow the procedure to create a Network Topology:

1. Create a Project (refer to *3.3.1.1Create a project*)
2. Add devices (*3.3.1.2 Add devices*)
3. Double-click **Network Configuration** below the device as shown in the following figure.

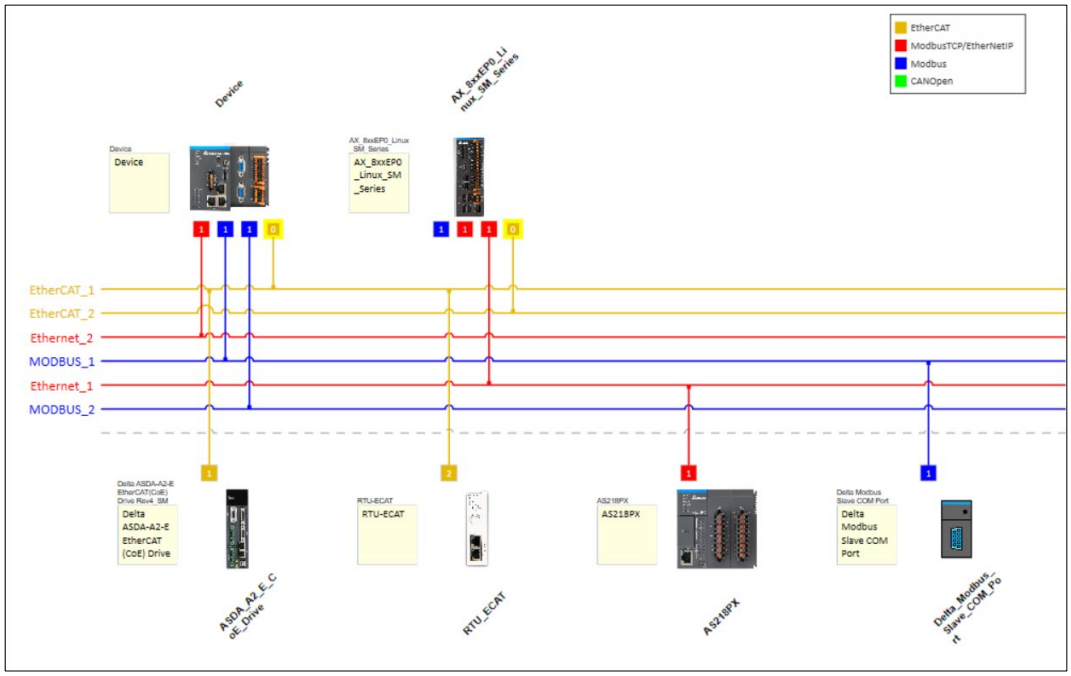


Result: The **Network Configuration Editor** window displays, including the Topology and Product list.

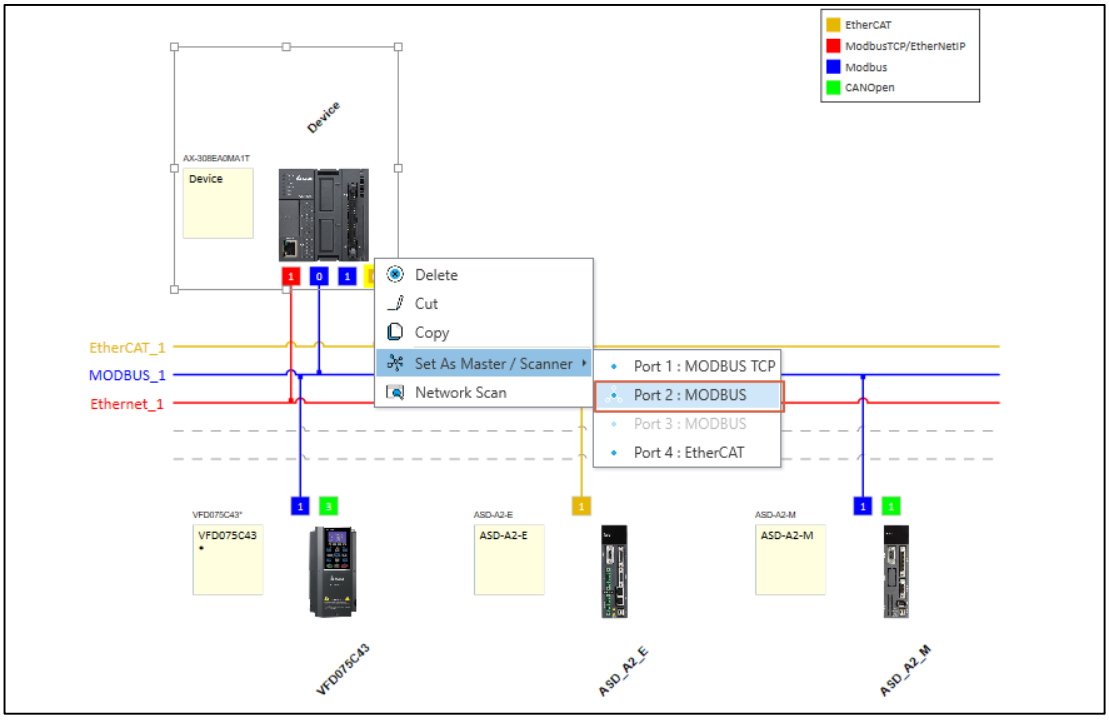


4. User can drag required device from **Product List** to **Network Configuration Editor** and connect the correspond protocol to build project network topology.
5. Press the port to the network protocol line. Connect the devices to the required network.

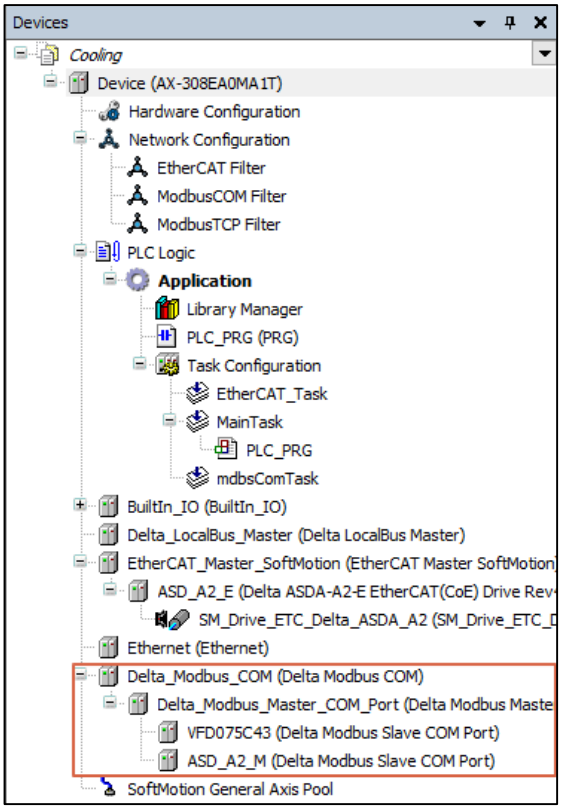
   Result: The network configuration is shown in the following figure.



6. Set as Master to connect the slave device: Right-click controller and choose **Set as Communication Master/Scanner** to set the major protocol.



Then, project tree will show the related slave device. For example, set MODBUS as Master and the MODBUS slave will be on project tree.



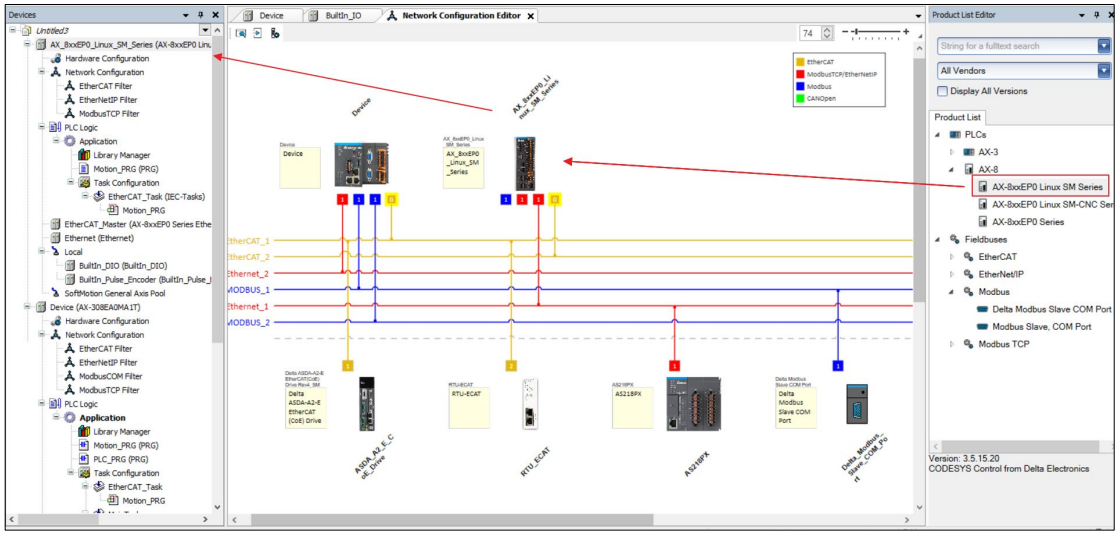
Note:

1. In **Network Configuration Editor** the controller appears above the network line and devices will appear below the network line.
2. Under the condition of different versions of the software, if the user imports the project from a higher version to a lower version, the slave device can only restore part of it. The network configuration diagram may also be different because different network devices support different software versions.
3. Create a new project without NW/HW Config and export the project. Import the project again. There will be “Errors occurred during the import of the Selected files”. This is original CODESYS import error.

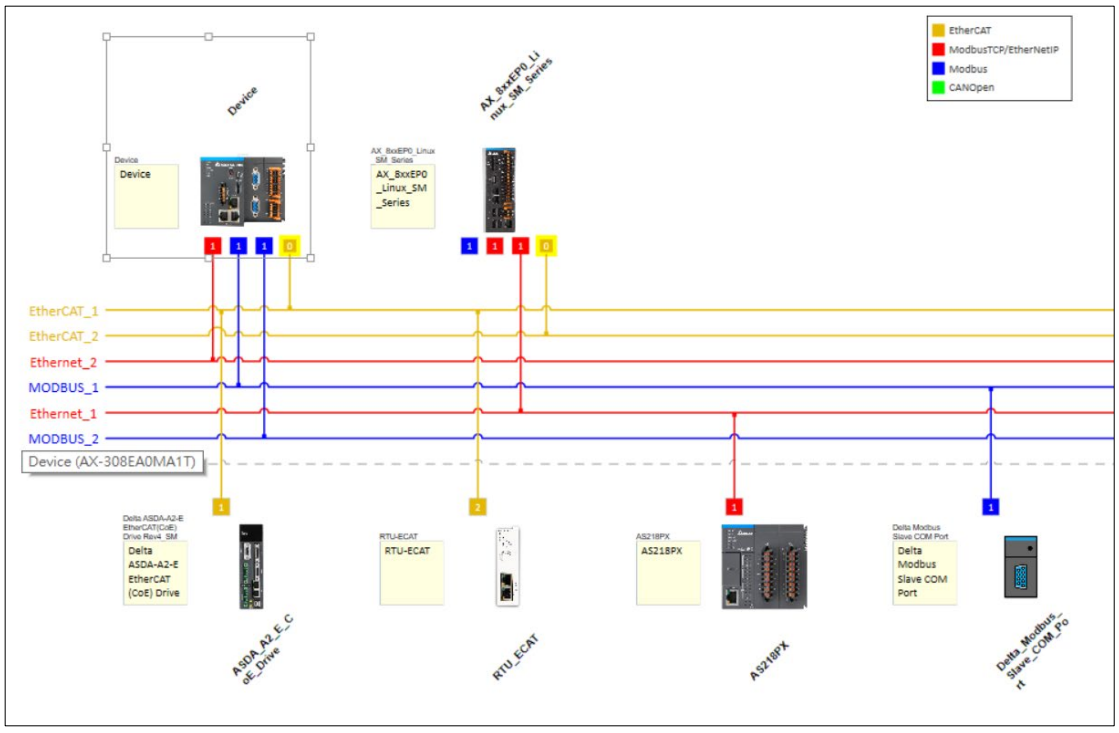
6.2.2 Multi Controller Network Topology

Support Multi controller network configuration.

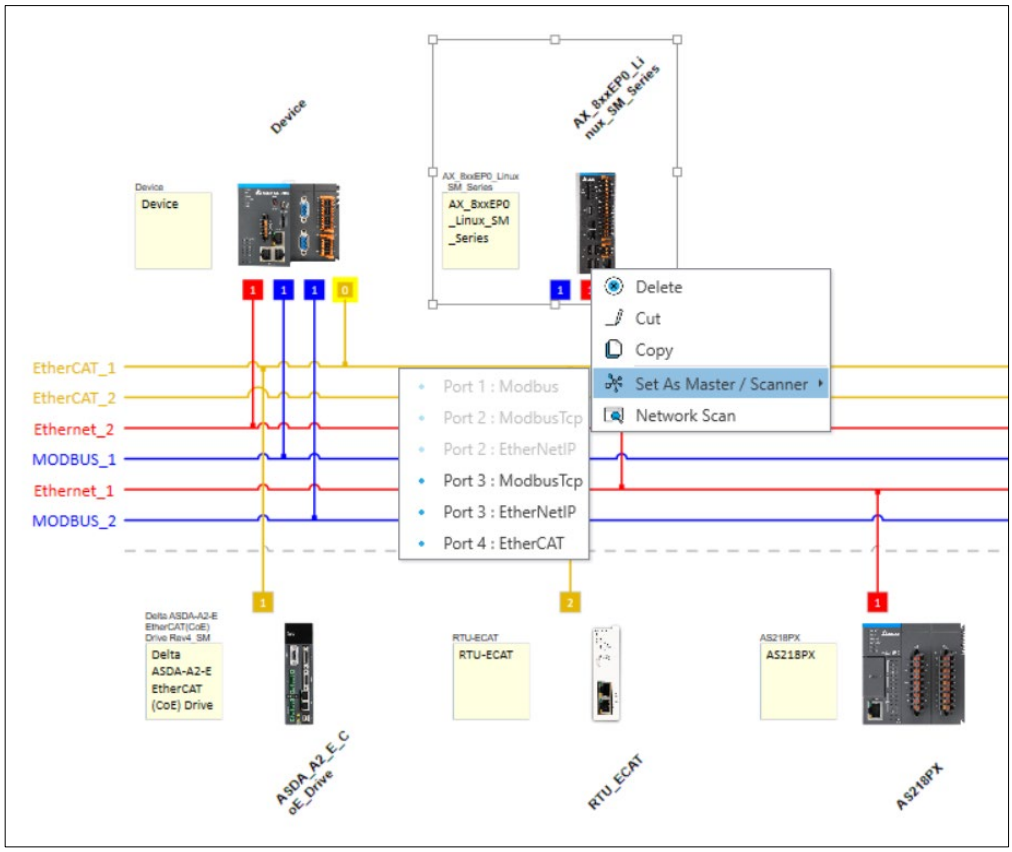
1. Drag a new controller on the topology. The device will be displayed on the topology and shown on the project tree as same time.



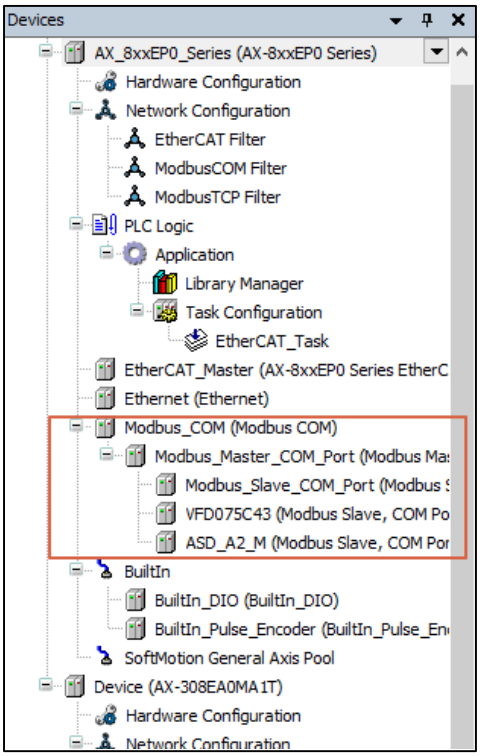
2. Connect the correspond protocol to build project network topology.



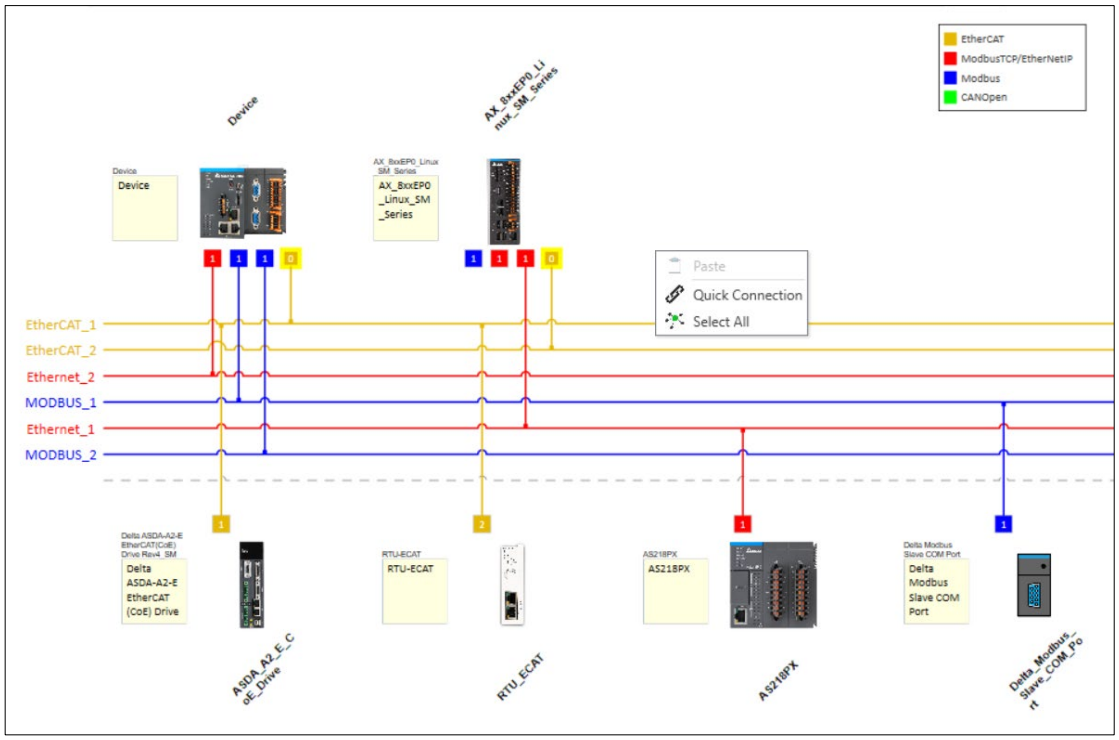
3. Set as Master to connct the slave device: Right-click controller and choose **Set as Master/Scanner** to set the major protocol.

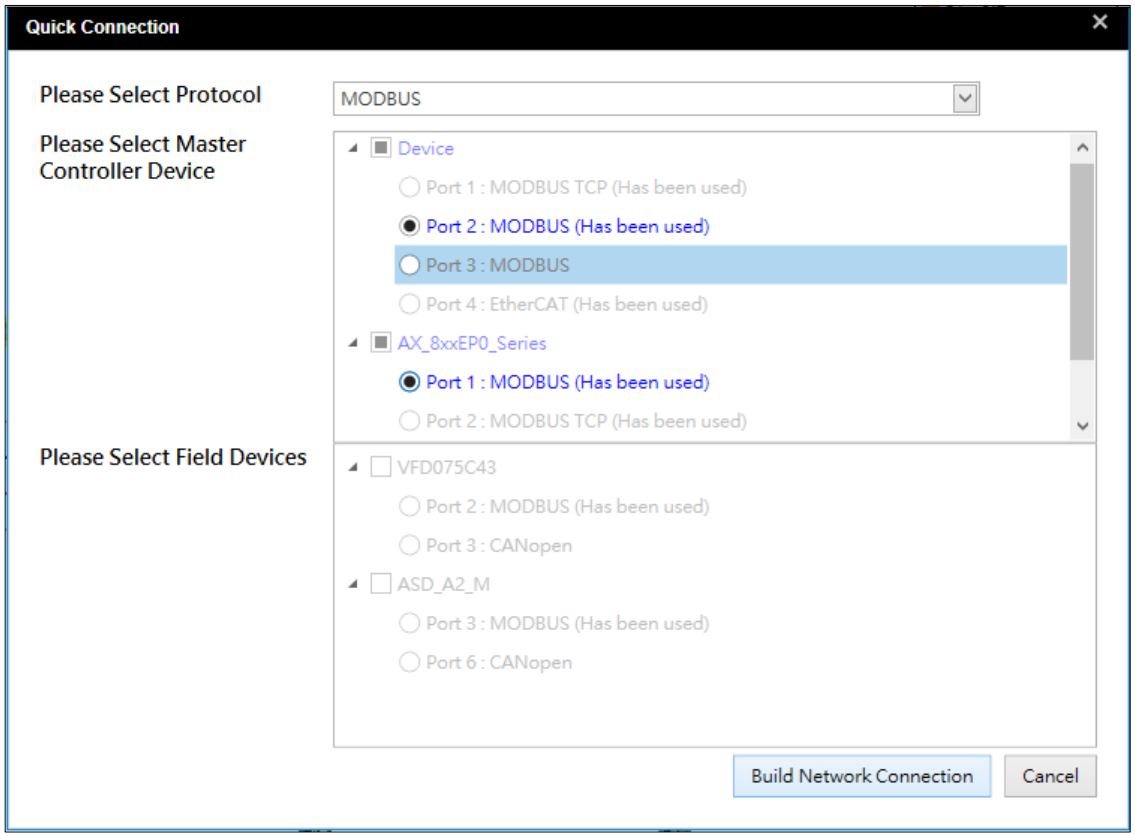


Project will show the related slave devices.



4. If the protocol is too complex, user can right-click on the topology blank place and find **Quick Connection** to build device fieldbus line quickly.


EtherCAT
ModbusTCP/EtherNetIP
Modbus
CANOpen
Device
AX_8xxEP0_Linux_SM_Series
Paste
Quick Connection
Select All
EtherCAT_1
EtherCAT_2
Ethernet_2
MODBUS_1
Ethernet_1
MODBUS_2
Delta ASDA-A2-E EtherCAT(CoE) Drive
RTU-ECAT
AS218PX
Delta Modbus Slave COM Port


Quick Connection
Please Select Protocol
MODBUS
Please Select Master Controller Device
Device
Port 1 : MODBUS TCP (Has been used)
Port 2 : MODBUS (Has been used)
Port 3 : MODBUS
Port 4 : EtherCAT (Has been used)
AX_8xxEP0_Series
Port 1 : MODBUS (Has been used)
Port 2 : MODBUS TCP (Has been used)
Please Select Field Devices
VFD075C43
Port 2 : MODBUS (Has been used)
Port 3 : CANopen
ASD_A2_M
Port 3 : MODBUS (Has been used)
Port 6 : CANopen
Build Network Connection
Cancel

Name	Description
Select Protocol	Select one protocol which user want to build the connection line
Select Master controller device	Select the main Master controller based on the selected protocol
Select fieldbus Devices	Select the connection fieldbus based on the selected protocol.
Establish a Bus Connection	Click the button, the selected fieldbus and device will build and connect

6.3 Communication Setting

In this tab of the generic device editor, user can define the connection between DIADesigner-AX and the device where user application should run.

Select a gateway and a target device from the list boxes. The possible selections depend on the entries in the **Manage gateways** and **Manage favorite devices** dialogs (see the **Gateway** menu).

User can also type in the target device directly with the IP address (**Example**: 192.168.101.109), device address (**Example**: [056D]), or device name (**Example**: My Device). After the device is entered successfully, DIADesigner-AX searches for the device in the network of the gateway.

Note: The option of searching by device name requires unique device names in the network.

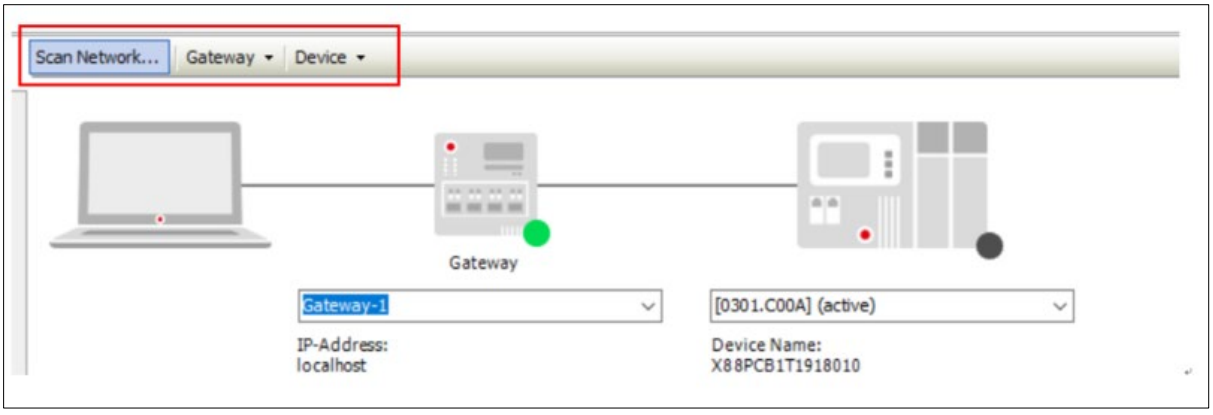
The filled icon on the left side of the device provides information about the connection status:

- : The PLC is connected, the application is running, the device is in operation, and data is being exchanged. The **Refresh I/Os in stop** check box on the **PLC Settings** tab can be selected or cleared.
- : The PLC is connected and in **STOP**; and the **Refresh I/Os in stop** check box on the **PLC Settings** tab is cleared.
- : The PLC is connected and the application is running. Diagnostic information is available.

- : The device is in preoperative mode and is not running yet. Diagnostic information is available.
- : The device is not exchanging data; bus error, no configuration, or simulation mode.
- : The device is running in demo mode for 30 minutes. When this time has elapsed, demo mode will expire and the fieldbus will end the data exchange.
- : The device is configured, but not operational. No data is exchanged. Example case: CANopen devices when booting and in preoperative mode.
- : Redundancy mode is active. The fieldbus master is not sending any data because another master is active.
- : The device description could not be found in the device repository.
- : The device itself is running, but a child device is not running. The child device is not visible due to a collapsed device tree

  Note: If user login while the device description on the target device is more recent than in the project, then a warning prompt opens with the possibility to cancel the process.

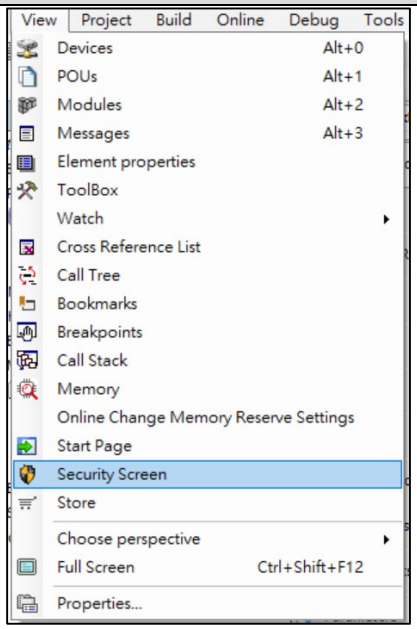
Clicking the filled circle of the target device starts a network scan for the device. This works only if the network is not already being scanned.



Function	Description
Scan Network	Click this button to open the **Select Device** dialog box, which will list all configured gateways and related devices. User can select the target device from the list.

Function	Description
Gateway	This menu includes the following commands: • **Add New Gateway**: Opens the **Gateway** dialog for defining a new gateway channel. • **Manage Gateways**: Opens the **Manage Gateways** dialog with an overview of all gateways. User can add or delete entries here or change their order. • **Configure the Local Gateway**: Opens the **Gateway Configuration** dialog. User can configure the block drivers for the local gateway.
Device	This menu includes the following commands: • (option)Add Current Device to Favorites: Adds the currently set device to the list of favorite devices. • (option)Manage Favorite Devices: open the dialog box for managing favorite devices with a list of favorite devices. User can add/delete items or change their order in this dialog box. At the top is the default device. • (Option) Filter network scan by device ID: Only display devices with the same target ID as the devices currently configured in the project. • (Option) Confirm online mode: • : When calling the following online commands (for security purposes) DIADesigner-AX requires the user to confirm the following: forced value, written value, multiple load, delete forced list, single loop, start, pause. • (Option) Save communication settings to the project: ☑ : DIADesigner-AX will save the communication settings in the project, which can be reused in the same computer later. ***Note**: If user use this project on another computer, user must reset the valid path.*

Function	Description
	☐ : DIADesigner-AX will store the communication settings in the locally installed options, and then user can reuse this setting on the same computer. ***Note**: When using DIADesigner-AX SVN, this option should be avoided to prevent blocking the device target..* • Rename the activated device: User can change the device name. • Flash activated devices: Devices that support this function will emit a flash signal. • Send response service: DIADesigner-AX will send five echo services to the PLC. Similar to the ping function, they are used to test network connections. In the beginning, these services are not sent with the packet, and then will be sent with the packet. The range of the packet depends on the communication buffer of the PLC. Open a message box, which contains the average echo service delay message and the sent packet range message. • Encrypted Communication: ☑ The communication to this controller is encrypted. A certificate of the controller is required in order to log in to the controller. If the certificate is not available, then an error message opens prompting whether or not the certificate should be displayed and installed.Please go to *View > Security Screen*

Function	Description
	 • In the user tab of the security fence window, if user select to enable enforced encrypted communication in the security level, the encrypted communication command will be invalid.  • **Change Communication Policy** Open the Change Communication Method Policy dialog box to change the settings of device communication encryption

Dialog Change Communication Policy

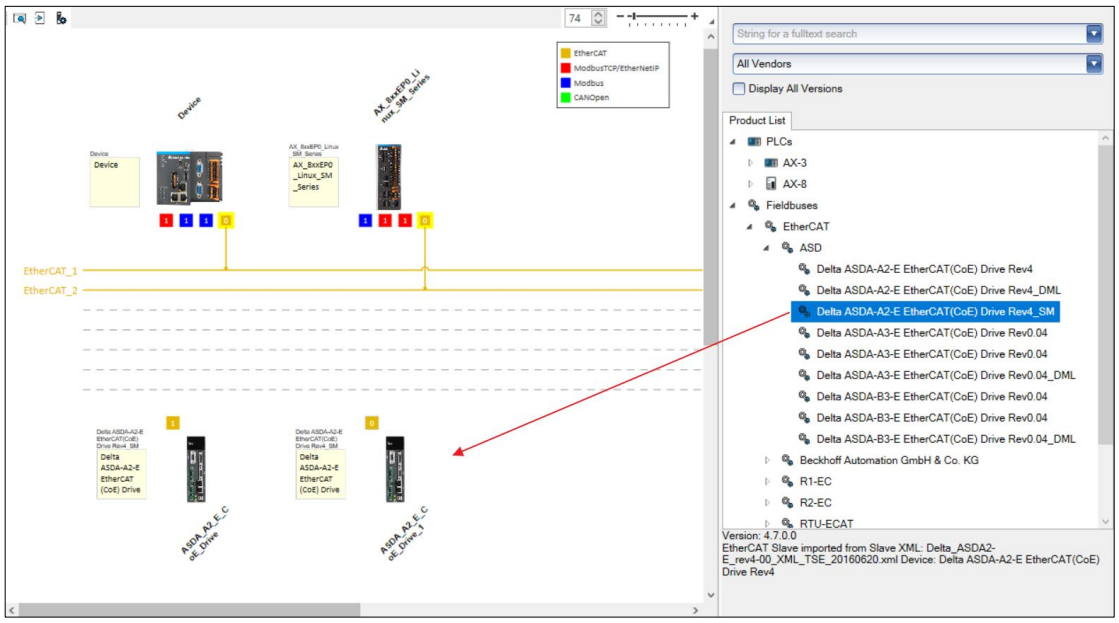
If a new communication policy is selected in this dialog, then the configuration on the controller is changed.
Communication

Current policy	Shows the currently selected policy for the encryption of communication
New policy	Drop-down list for the new policy for encryption. • No encryption: The controller does not support encrypted communication. • Optional encryption(default): The controller supports encrypted and unencrypted communication. • Enforced encryption: The controller supports encrypted communication only.
Device User Management	
Current policy	Shows the currently selected policy for user management
New policy	Drop-down list for the new policy for user management. • Optional user management(default): It is the responsibility of the user to enable user management on the device or leave the device unprotected. • Enforced user management**:** The user management on the device is enabled and cannot be disabled by the user.

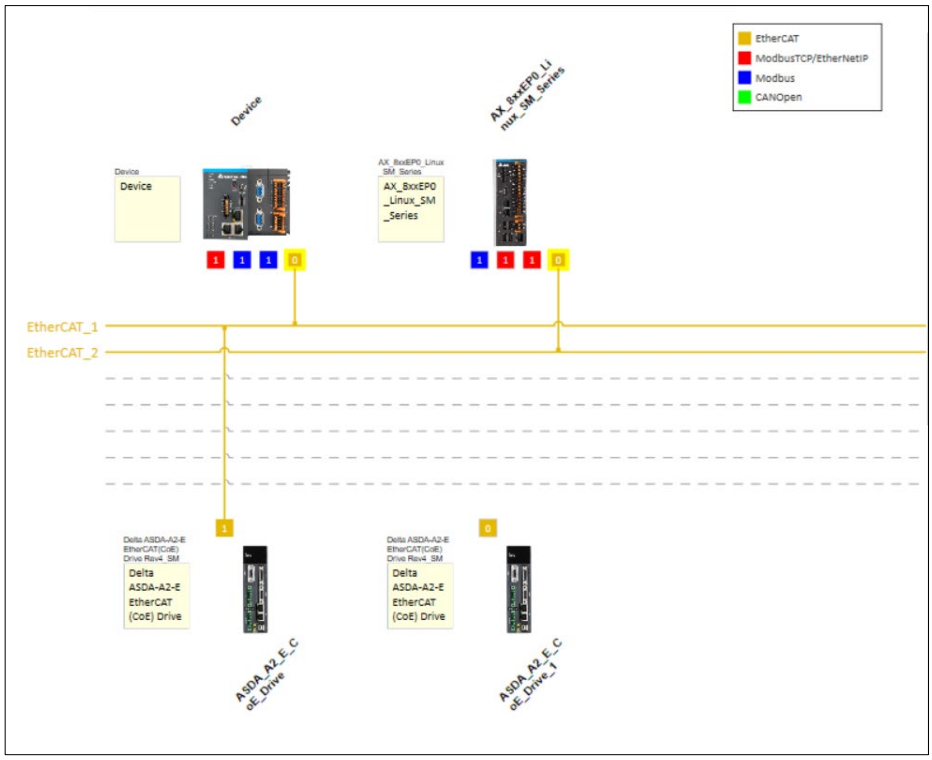
6.3.1 EtherCAT (AX-3 Series)

Follow the procedures to do EtherCAT settings.

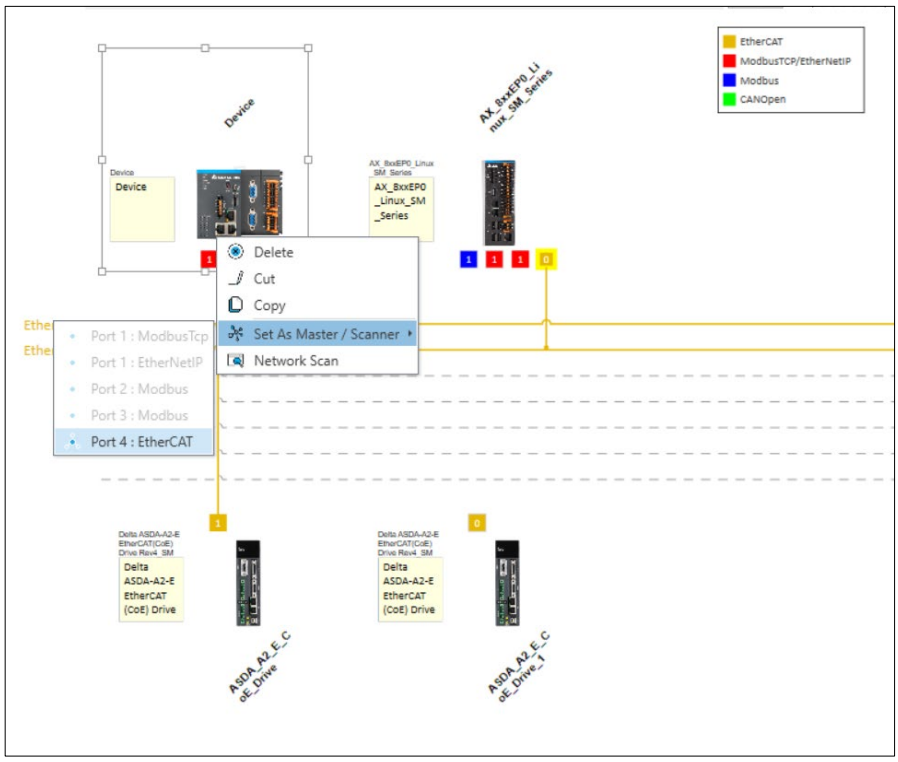
1. Create a project with AX-3 series.
2. Open **Network Configuration Editor**.
3. In the product list window on the right, select **EtherCAT** > **ASD** > **Delta ASDA-A2-E EtherCAT(CoE) Drive Rev4_SM.,** add it to the network configuration window, as shown in the following figure.



4. Connect the device to the network as shown in the following figure.

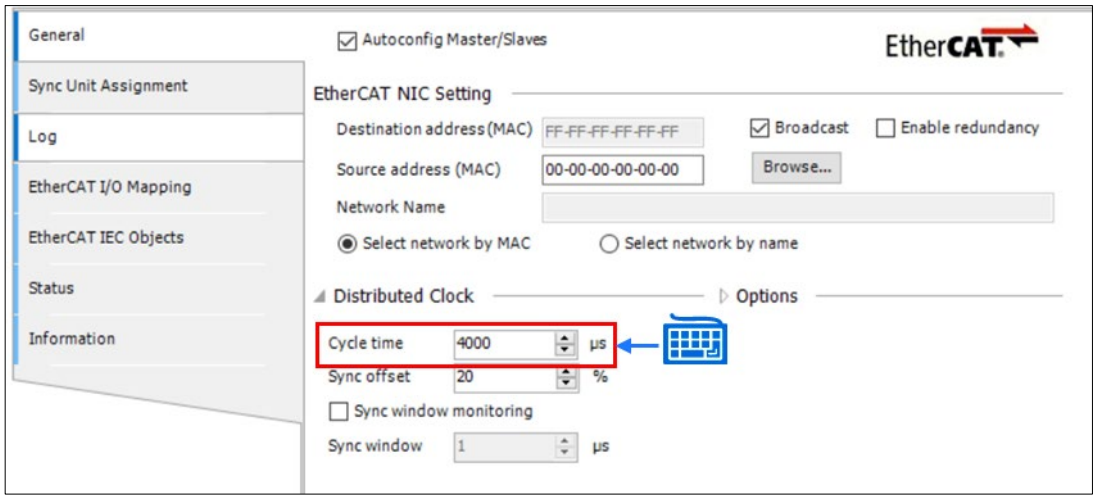


5. Double-click on the master station node as shown in the following figure.

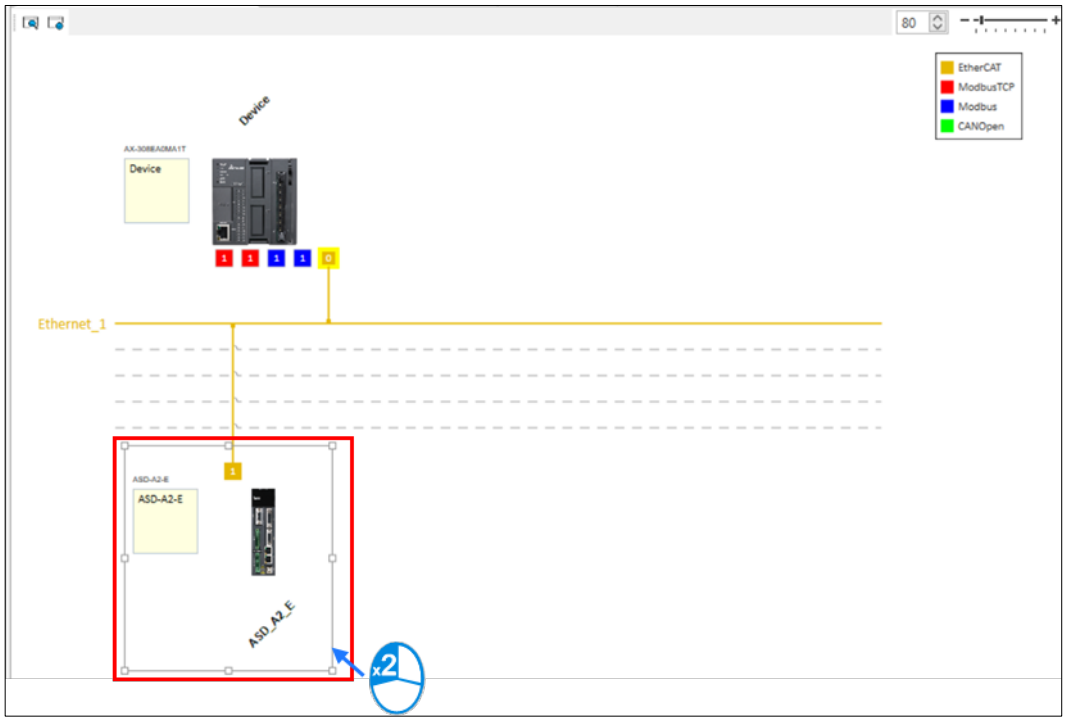


Result: Displays the **EtherCAT** master station.

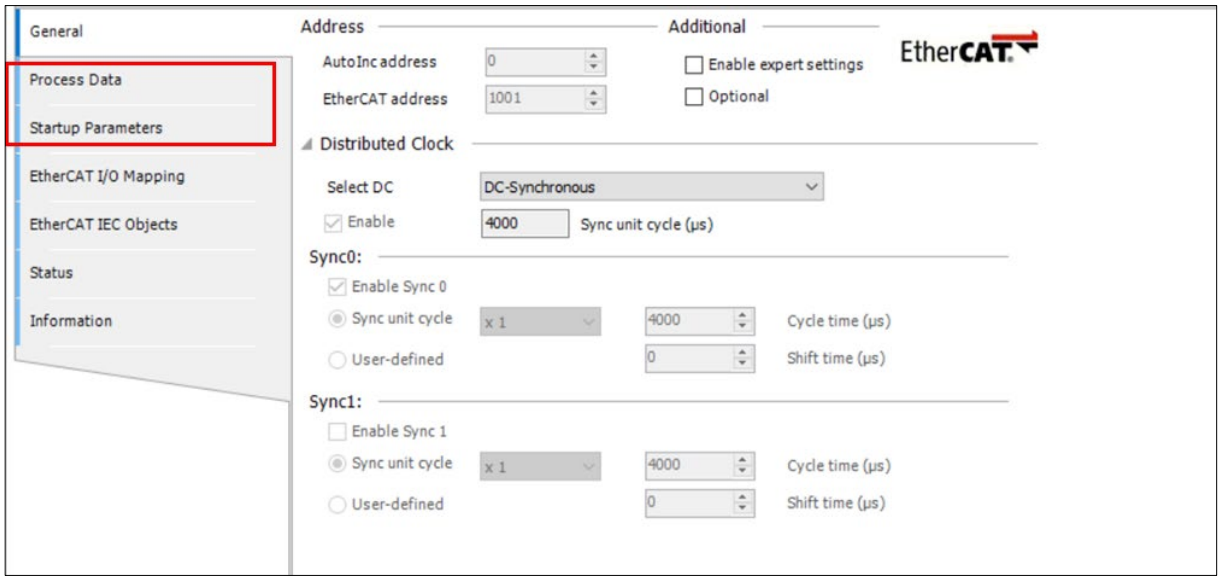
6. Set the EtherCAT synchronization time and other required settings as shown in the following figure.



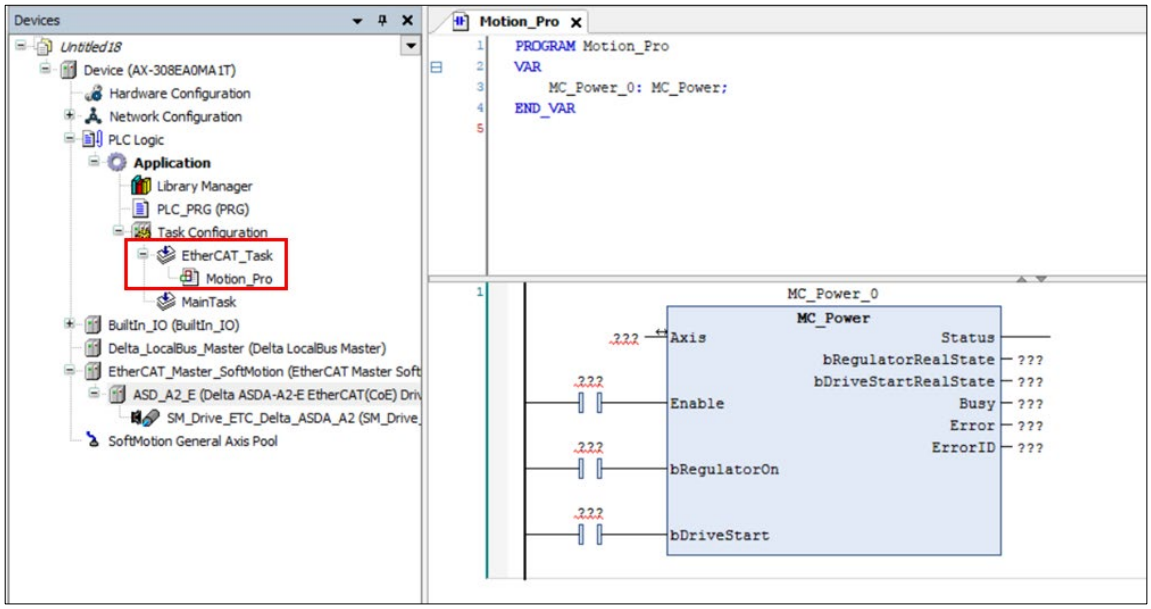
7. Double-click on the slave device to set the EtherCAT settings as shown in the following figure.



8. Set the required EtherCAT settings as show in the following figure.



Note: User can start programming later. It should be Noted that place the created POU under EtherCAT_Task, and use the motion function block to write the program in the POU under EtherCAT_Task to avoid the motion function block from running normally.



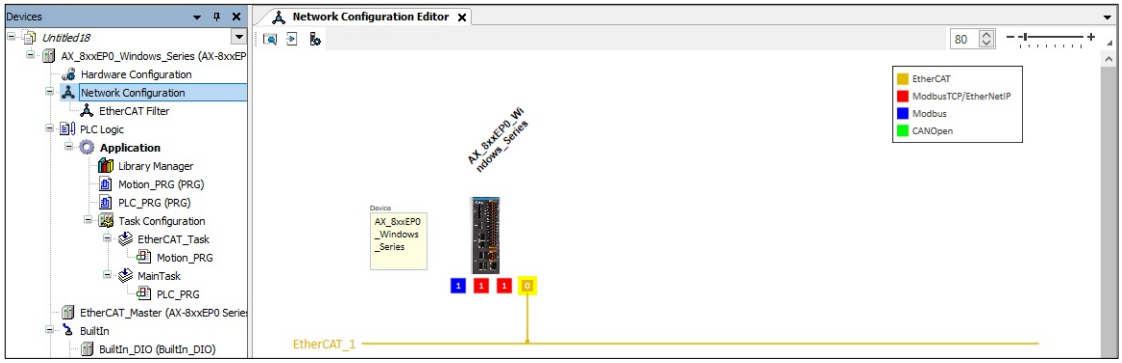
6.3.2 EtherCAT (AX-8xxEP0 series)

For the EtherCAT communication of the AX-8xxEP0 series, the biggest difference from the AX-3 is that the AX-8xxEP0 series can be increased to three EtherCAT ports in the network configuration.

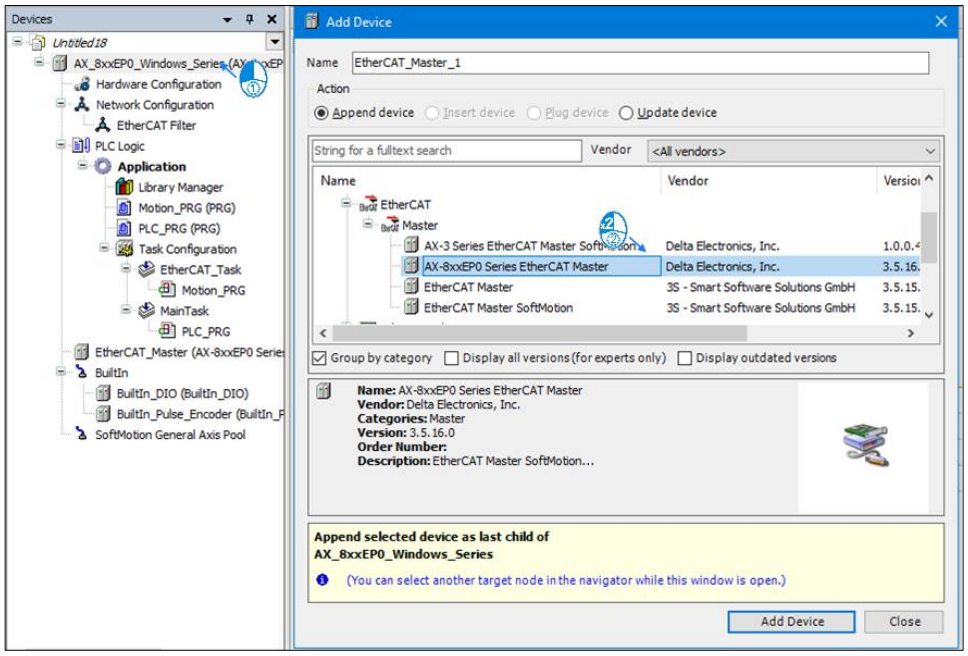
Please follow the steps below to add AX-8xxEP0 series EtherCAT:

1. Double-click **Network Configuration**.

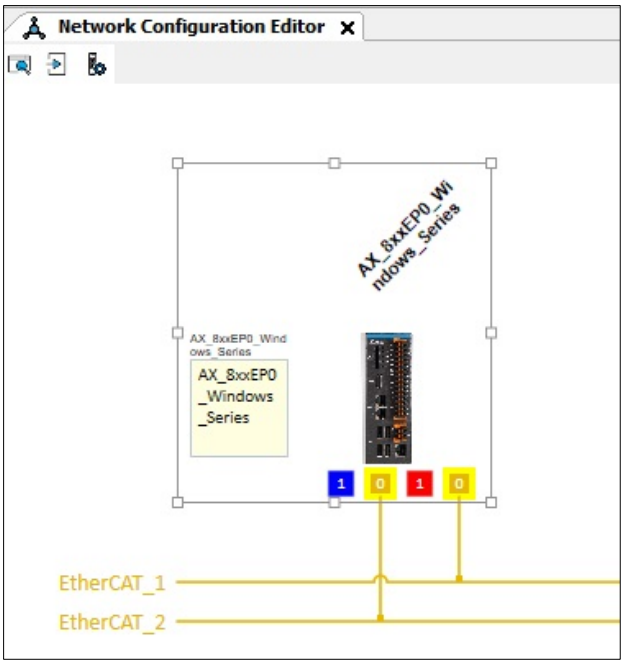
   Result: The figure displays as shown below.



2. To add multiple EtherCAT interfaces, first right-click the device in the project tree. After the Add Device window pops up, double-click the device user want to add, as shown in the following figure**.**



Result: At this time, user can see in the network configuration window that the device has two interfaces that have become EtherCAT.

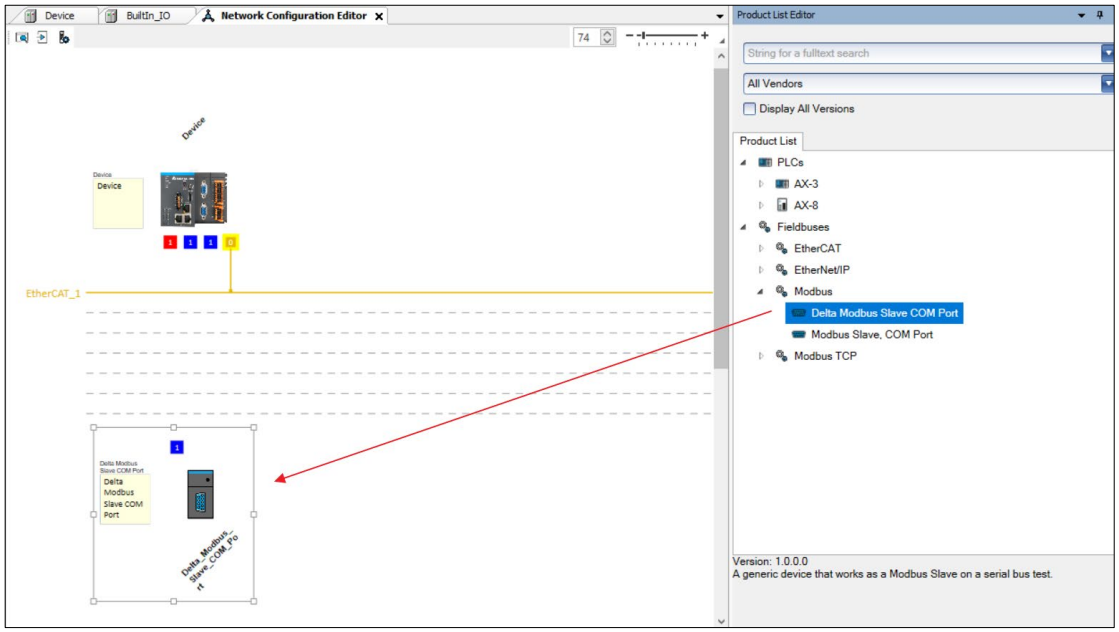


3. Repeat the above steps to add up to three EtherCAT Masters.

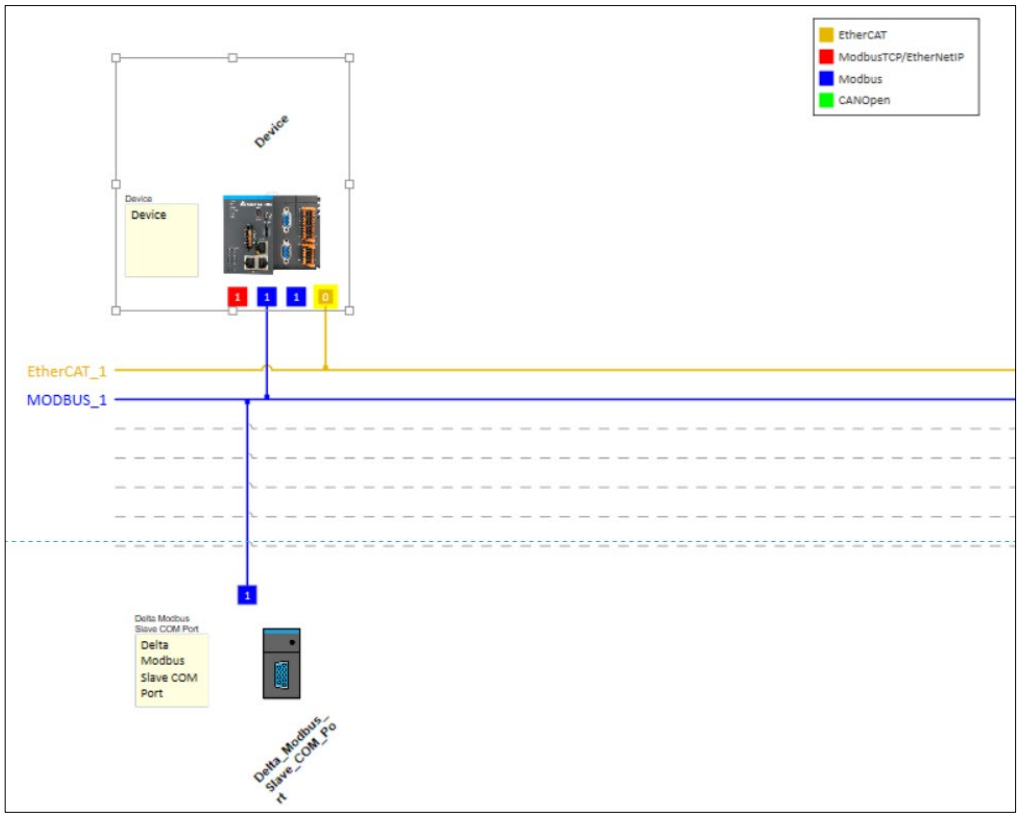
6.3.3 MODBUS COM

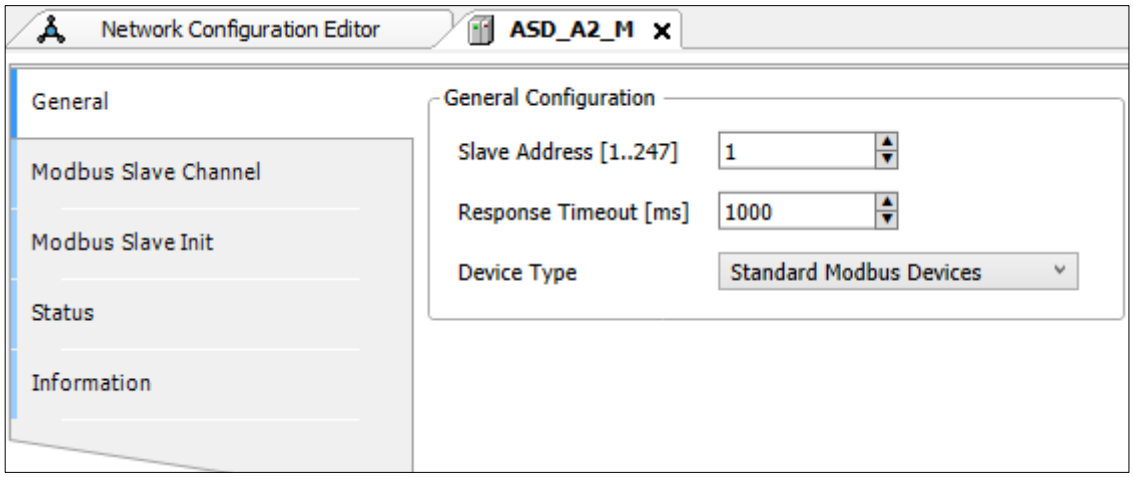
Follow the procedures to do MODBUS settings.

1. In the network configuration window, select a MODBUS COM device. For example: **Modbus**> **Delta Modbus Slave COM Port**, then drag/double-click this device to add it to the network configuration window, as shown in the following figure.



2. Connect to the MODBUS line, Set controller as **MODBUS Master**, and double-click field device to open the parameter page.



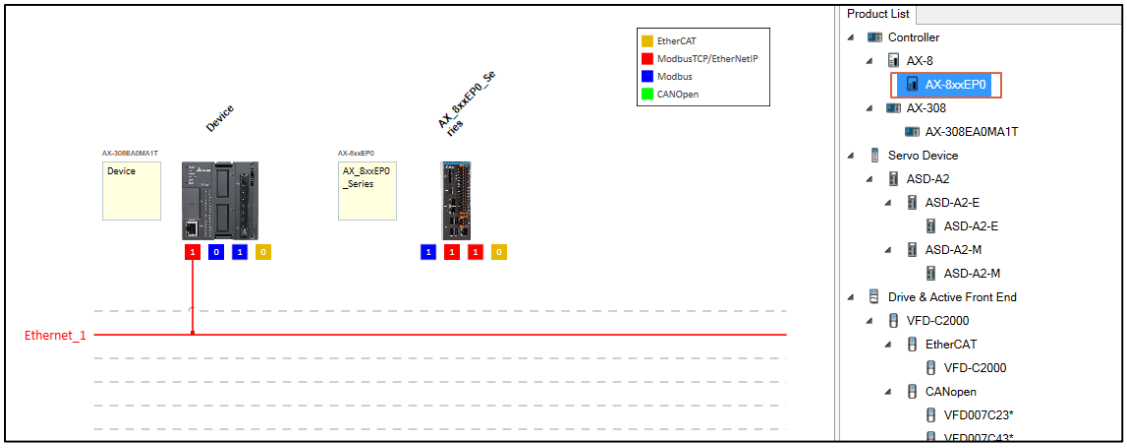


For more information, please refer to *4.2.4 COM Port Setting* and *4.3.2 COM Port Setting* setting.

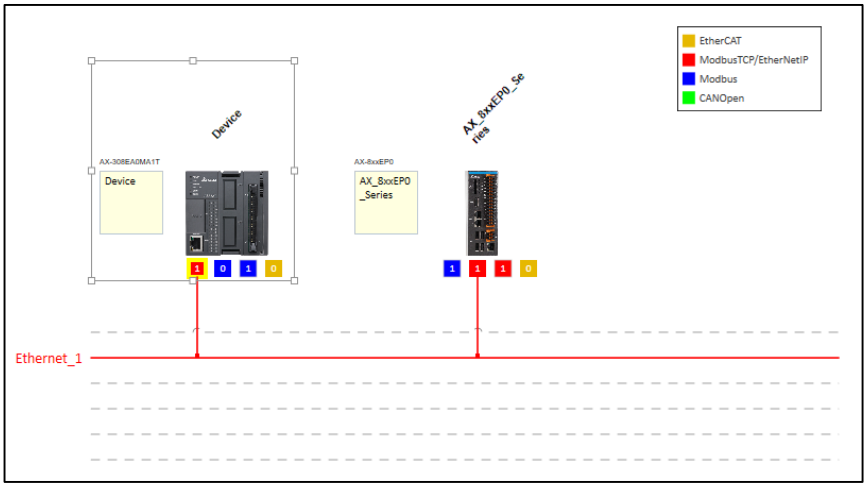
6.3.4 MODBUS TCP

Follow the procedures to do MODBUS settings.

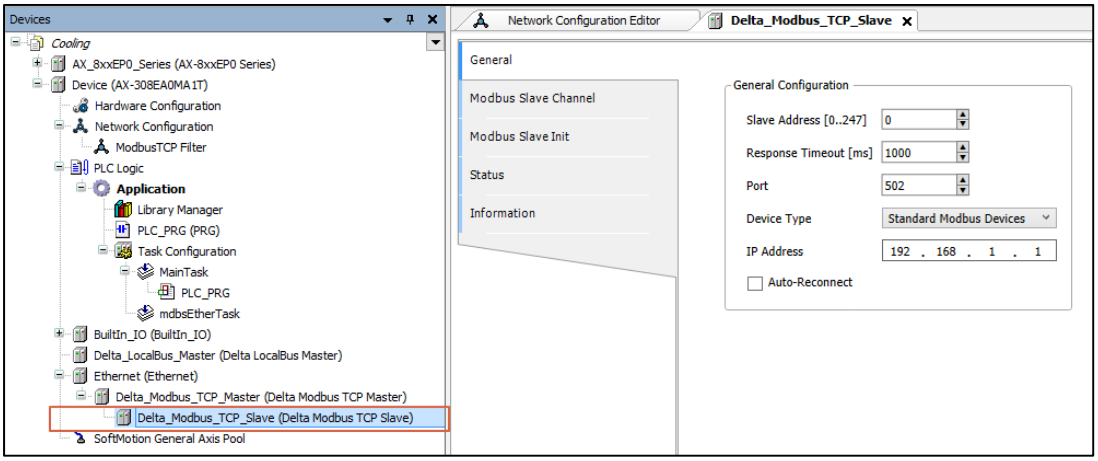
1. From the **Product List Editor** window, select one Ethernet device **Example**: **Controller** > **AX-8** > **AX-8xxEP0** and add it to the **Network Configuration Editor** window as shown in the following figure.



2. Connect to the Ethernet line and Set AX-3 controller as **MODBUS TCP** Master and double-click field device to open the parameter page.



User can go to project tree to open the MODBUS TCP slave parameter.



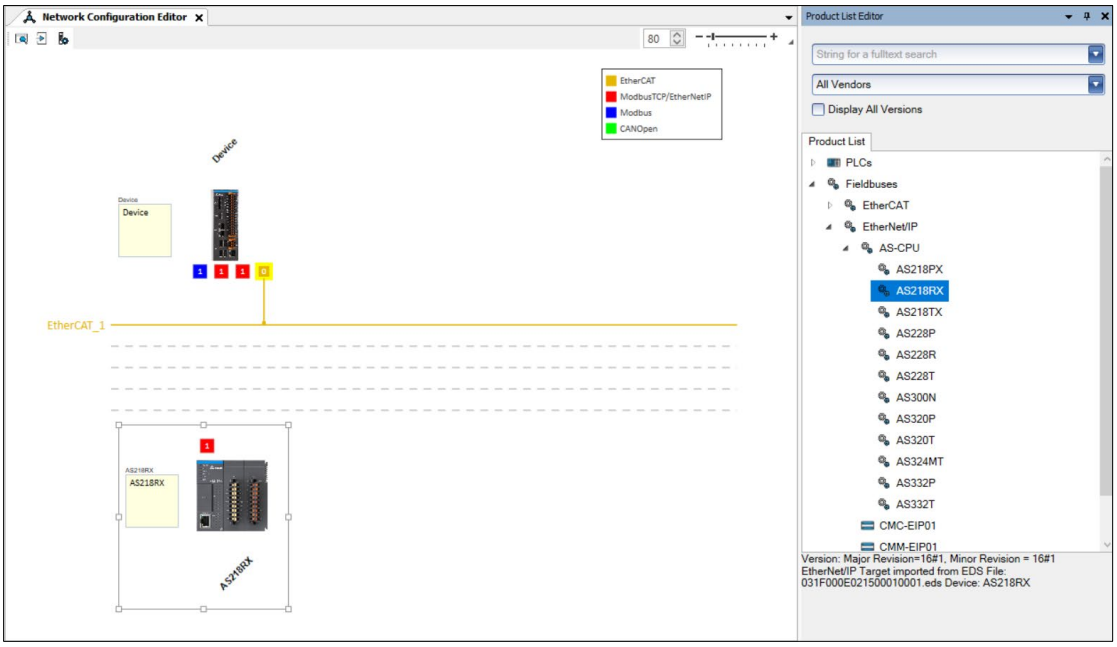
For more information, please refer to *4.2.6 Ethernet Setting* for AX-3 TCP port setting and *4.3.4 EtherNet Setting* for AX-8 TCP port setting.

6.3.5 EtherNet/IP

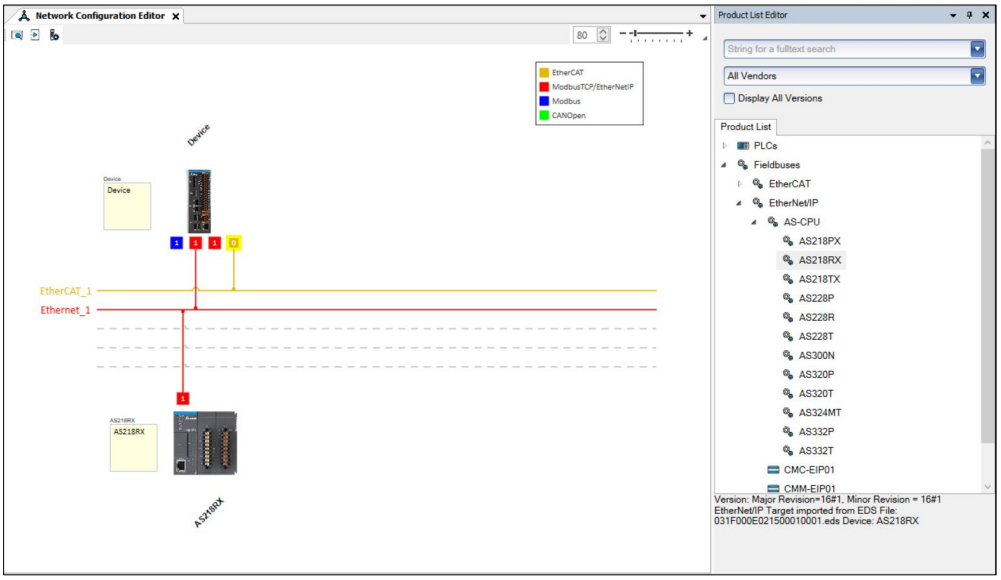
Please follow the steps below to perform EtherNet/IP settings.

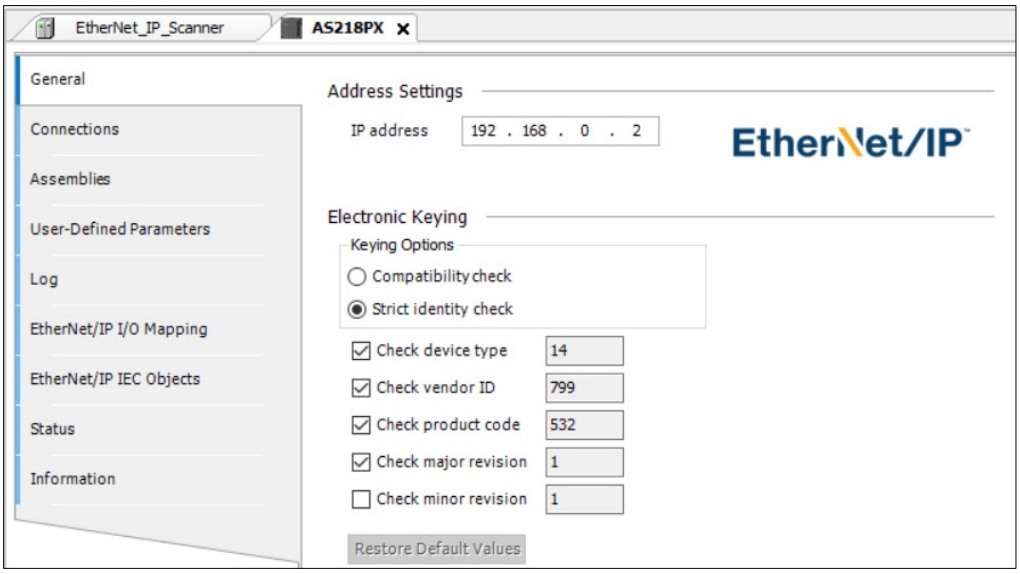
In the network configuration window, select an EtherNet/IP device.

Example: **AS CPU** > **AS218RX**, then drag/double-click this device to add it to the network configuration window, as shown below.



Connect the Ethernet line. After the controller is set as the communication master by pressing the right button, double-click the field device to open the parameter tab.





For more information on AX-308E EtherNet/IP settings, please refer to *4.2.7 EtherNet/IP Settings*. For more information on AX-8 EtherNet/IP settings, please refer to *4.3.5 EtherNet/IP Settings*.

6.3.6 OPC UA

In the AX series controller, the OPC UA server is included, and this service is used through the client access to the control interface of the controller. The OPC UA server communicates with the OPC UA client through a TCP connection. Therefore, these connections must be checked separately for security to protect the client and OPC UA user management with encrypted communication, thereby protecting the OPC UA server.

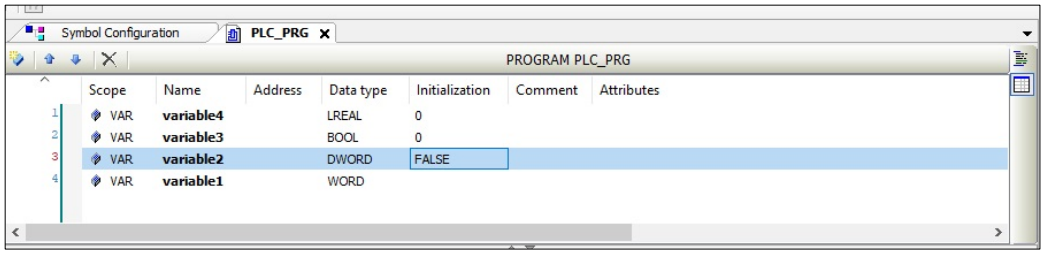
OPC UA Server Support the following functions:

- Browse data types and variables
- Standard read/write
- Value change notification (for subscribed and monitored items)
- According to OPC UA standard, support Basic256SHA256 encrypted communication.

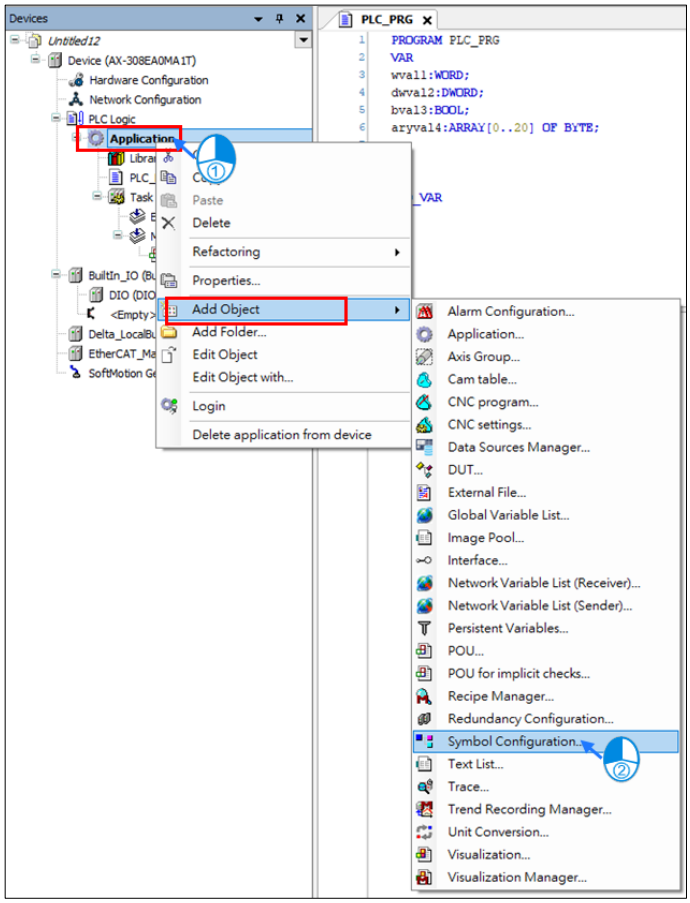
6.3.6.1 Create OPC UA Access Settings in the Project

To use the OPC UA Server function, user must first create the OPC UA access function in the project. The operation steps are as follows:

1. Create a new project in DIADesigner-AX.
2. In the project **PLC_PRG** program, create some different types of variables, as shown below.

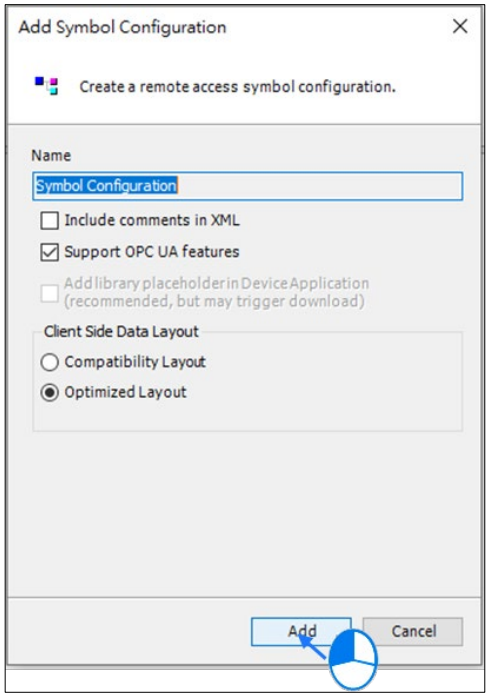


3. To add a symbol configuration from **Application**, right-click **Application> Add Object> Symbol Configuration**.

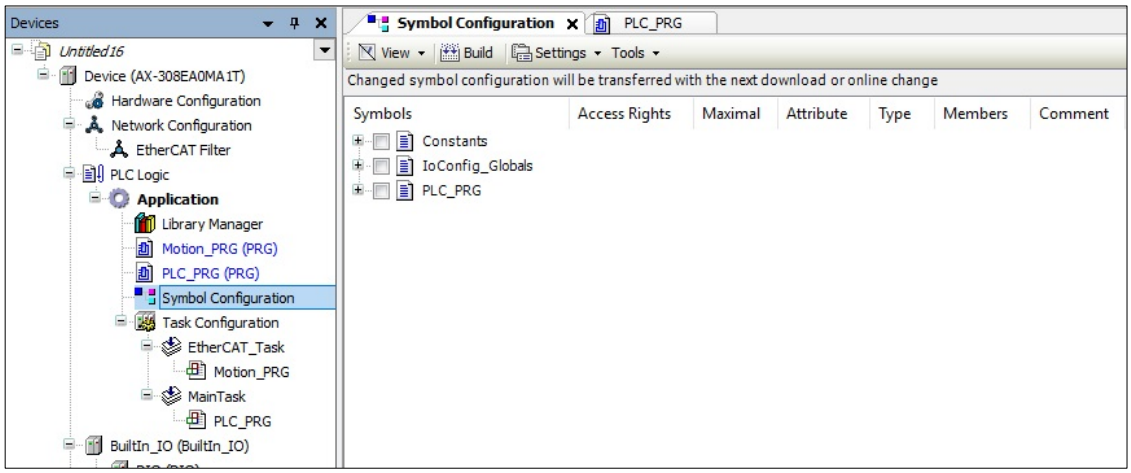


4. In the dialog box for **adding symbol configuration**, confirm that the **OPC UA**

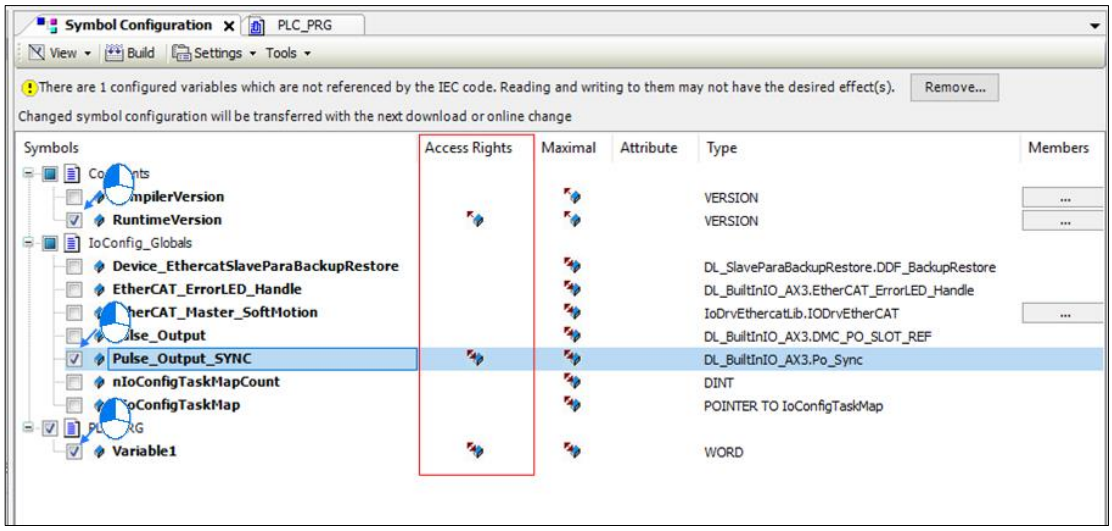
feature is enabled and then click **Open**.



5. Open the symbol configuration setting interface and click **compile**: The created projects and variables are displayed in the tree structure table below.



6. Check to allow the OPC UA client to set the monitored variables. Each variable can set its read/write permission in the access permission field. After changing the setting, it needs to be rebuilt and downloaded to the PLC host.



7. Finally, download the project to the host.

6.3.6.2 Use UaExpert Client to Establish a General Connection

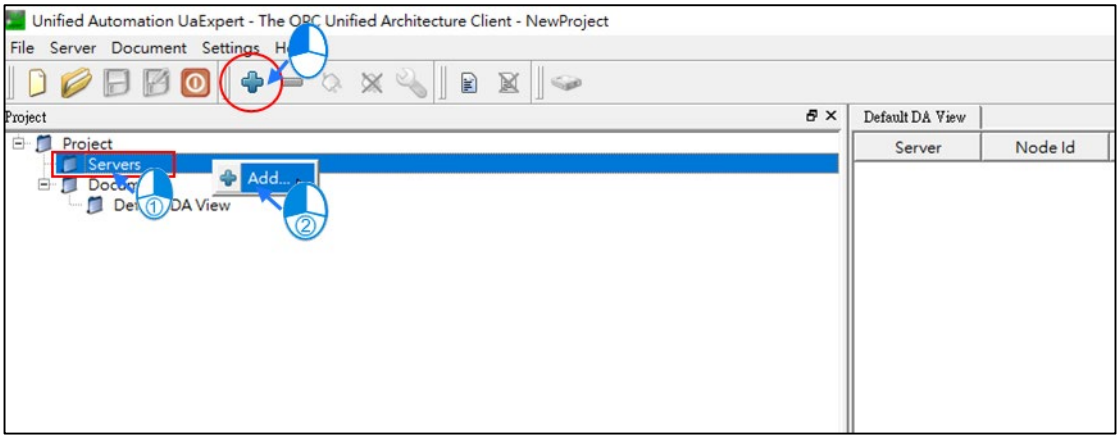
Use OPC UA Client to connect with AX series controller OPC UA server. Generally unencrypted connection.

The following is an explanation of how to set UaExpert client. Other OPC UA Clients are similarly set.

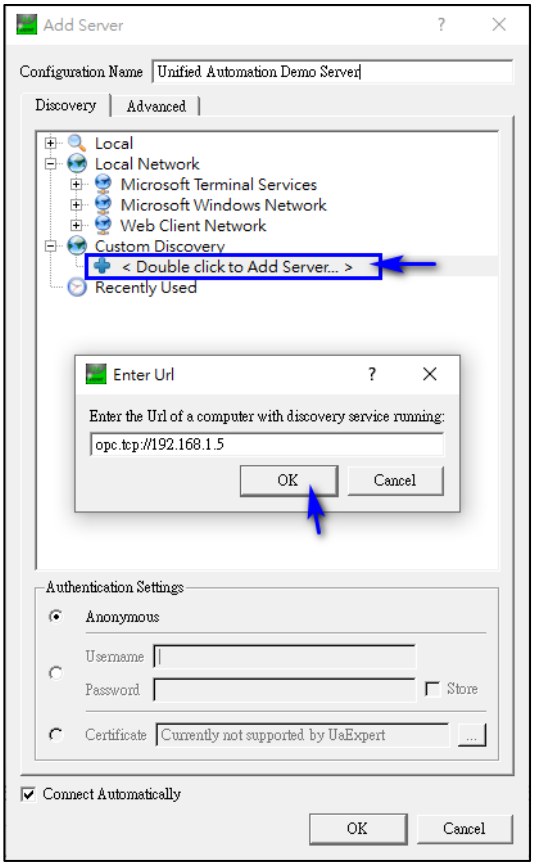
User can first go to UaExpert official website to register for free download.

Download URL：https://www.unified-automation.com/downloads/opc-ua-clients.html

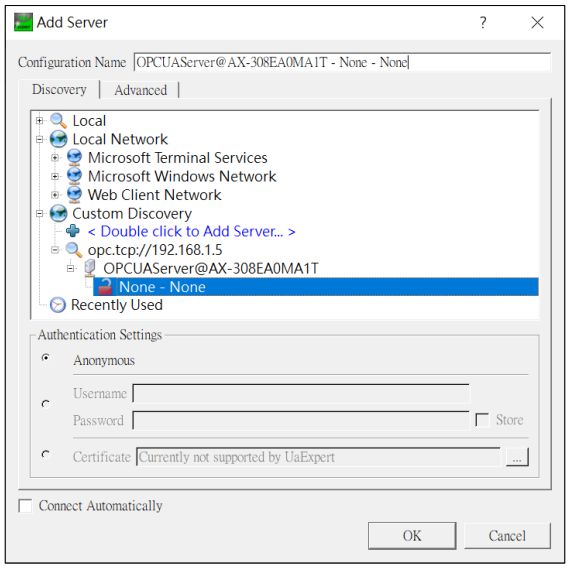
1. Launch UaExpert
2. Right-click **Servers** > **Add: open the Add Server dialog.**



Custom Discovery> **Double click** <**Double click to Add Server**...>> to pop up the **Enter Url** dialog box, enter opc.tcp://192.168.1.5, and click **OK**.

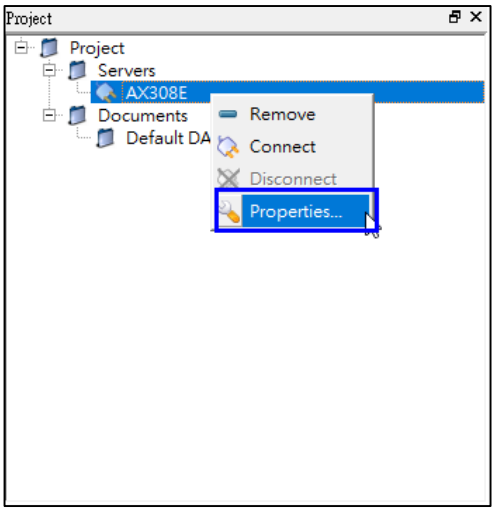


3. After completing the previous step, the following screen will appear and AX308E will be added under Project. If the encryption function is not used, the connection method will only scan out [No encryption] None-None. (No need to enter account password and select encryption format).

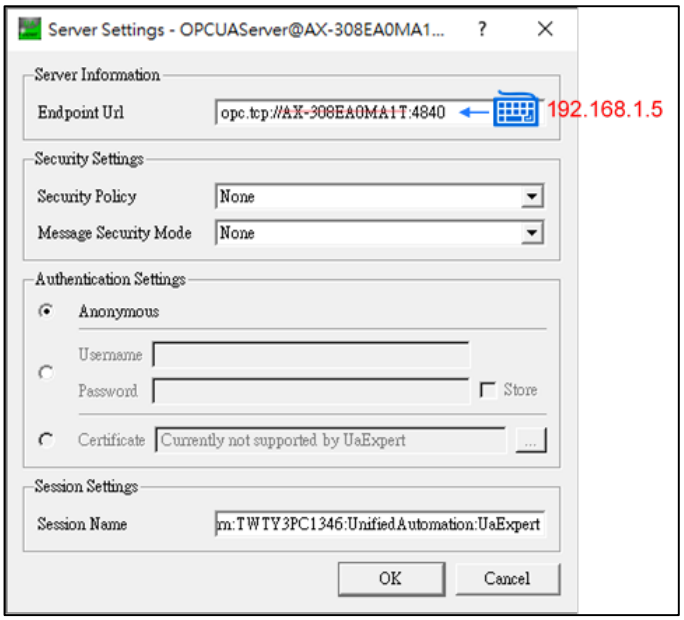


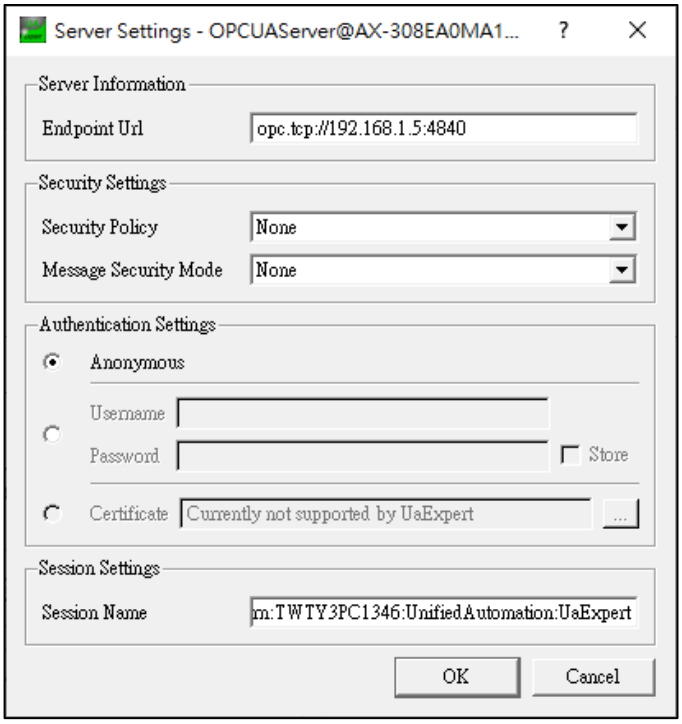
4. Modify the connection content: **Project** > **AX308E** and click the right mouse

button > **Properties.**

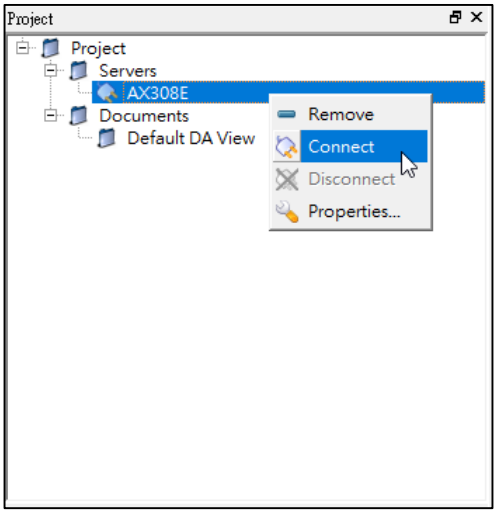


5. Enter in **Endpoint Url** in the Server Settings dialog box **opc.tcp://192.168.1.5:4840**.





6. Enter **opc.tcp://192.168.1.5:4840** in **Endpoint Url** in the Server Settings dialog box, and click **OK**.



Result: The AX-308E variable is displayed in the Address Space column.

7. Change variable data status through OPC UA client.

Select the variable and drag it to the Default DA View, user can see the variable information and status; double-click the variable Value column to modify the value of the variable.

6.3.6.3 Establish an Encrypted Connection

In order to encrypt data and securely exchange data with the client, the server needs a certificate when establishing a connection for the first time, and the client must classify this certificate as a trusted certificate. After the certificate is trusted, the correct user account must be used to establish a connection. The following chapters are the setting and operation instructions required for authentication, please set and operate in order.

- **Set user account password**

The OPC UA Server account password used is the same as the account password of the host encryption.

Log in to the controller for the first time to edit or view user management steps:

1. Double-click the **Controller Node** in the project tree.

   Result: The communication settings tab is displayed.

2. Switch to the user and group tab.

3. Click  the icon.

   Result: A dialog window opens and asks whether device user management should be activated.

4. Click **Ok**.

   Result: The device user login dialog box will open.

5. Specify Administrator as the user name and password.

   Result: The pop-up password is invalid, please re-enter! Dialog box.

6. Specify a new password.

   Result: After clicking OK, the device user management will be displayed in the editor view.

Steps to set up a new user in the user management of the controller:

1. Double-click the controller in the project tree.

   Result: The **communication settings** tab is displayed.

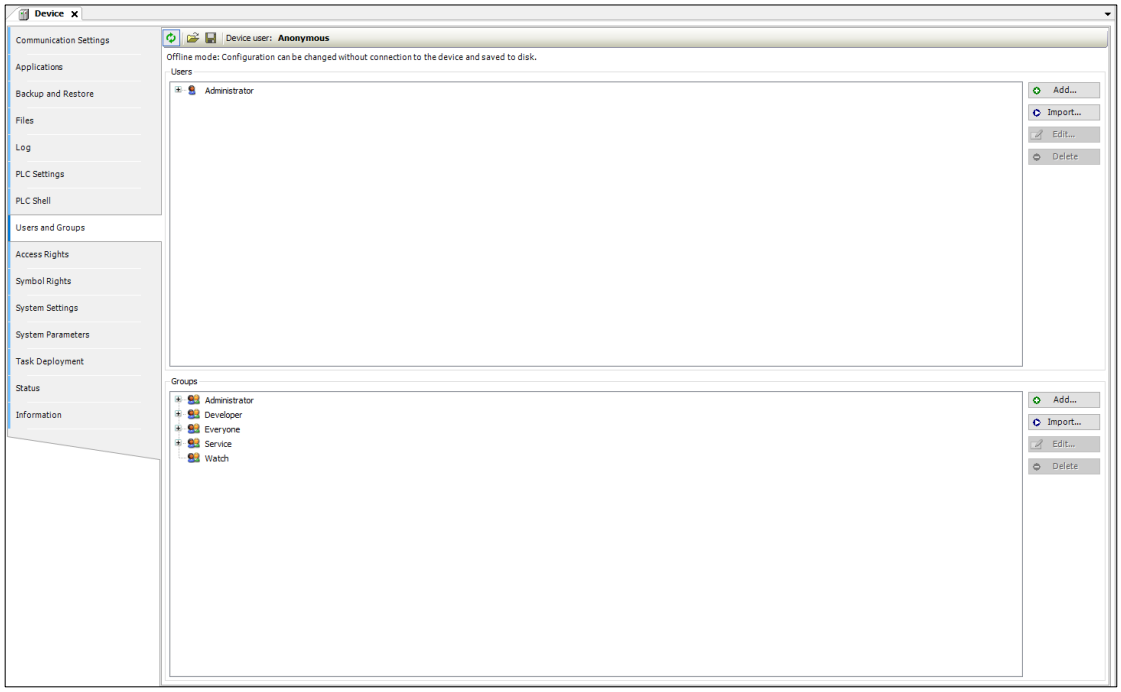
2. Switch to the **user** and group tab.

3. Click  the icon. Load the user management configuration from the controller to the editor. If user have not logged in to the device, the device user login dialog box will open and enter the **user name and password**.
4. To the right of the user field, click **Add**.

   Result: The Add User dialog box is opened.
5. Enter the **name of the new user** > **select the default group** > **set a password**. Click **OK**.

The example is as follows:

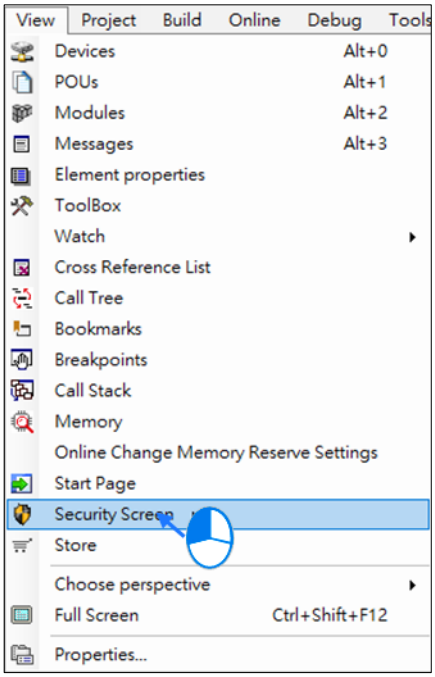
1. Default account：Administrator
2. New account：guest



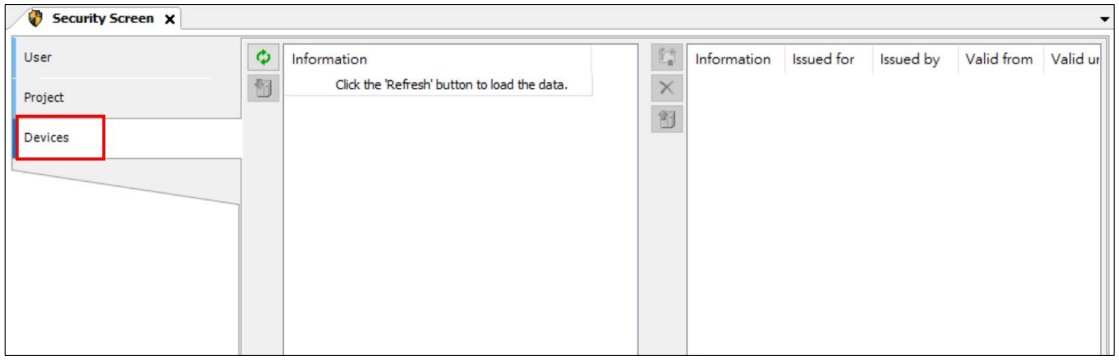
- **Trust Certificate Settings**

  Install the CODESYS Security Agent add-on. DIADesigner-AX is already installed.

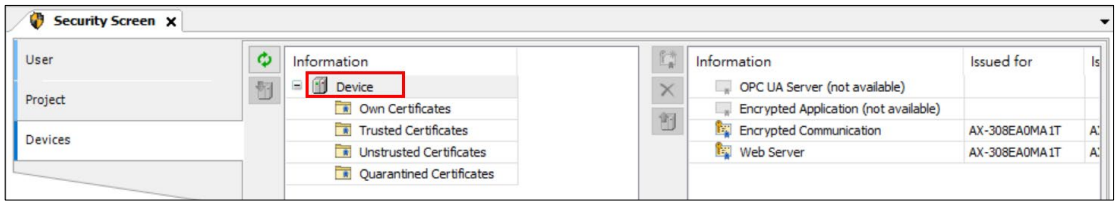
1. Click **View** > **Security Screen.**



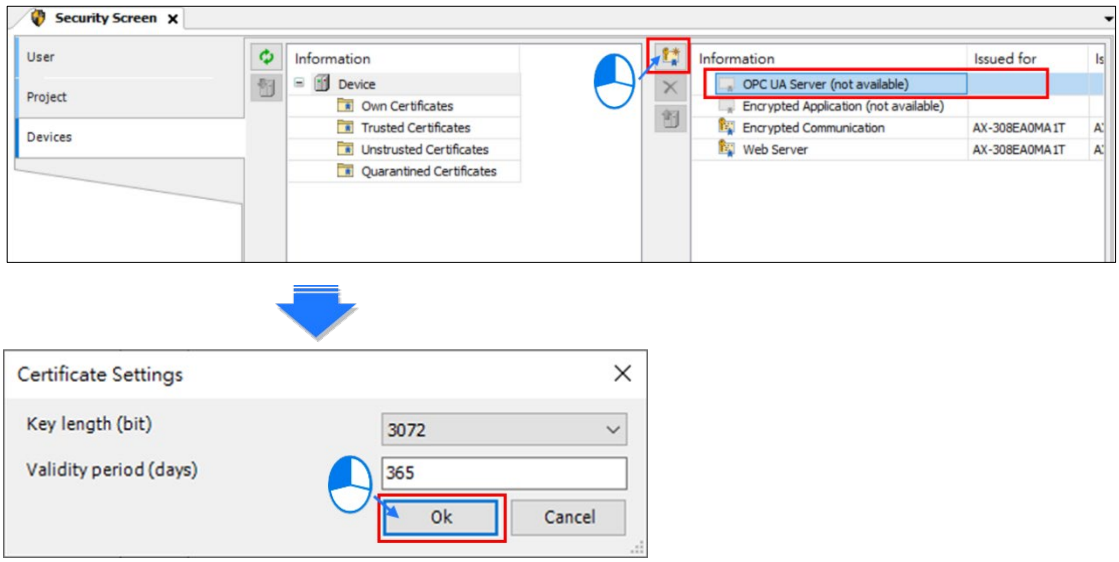
2. Select the **Devices** tab.



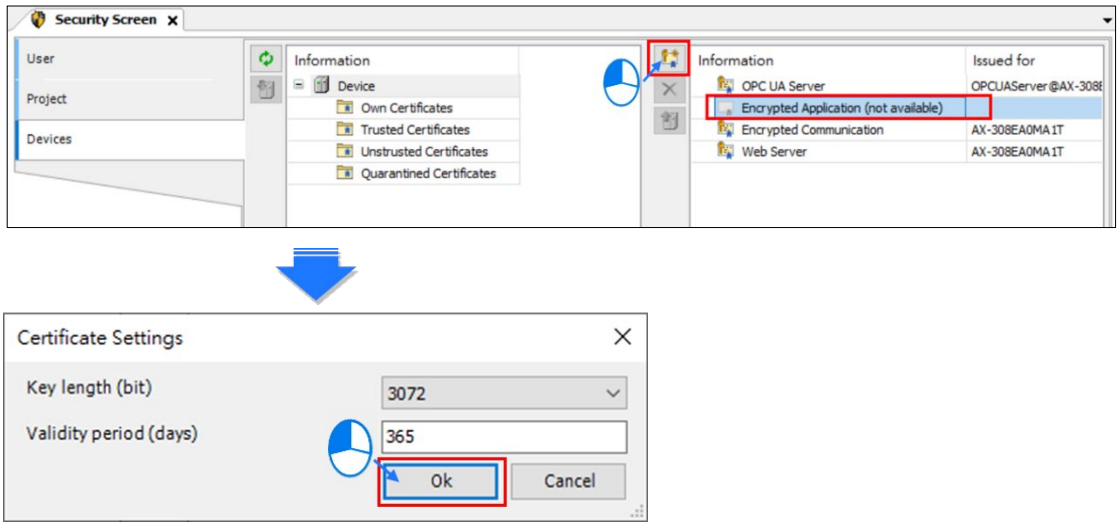
3. Click  , and select the controller **Device** in the left window: all services in the controller that require certificates will be displayed in the right window.



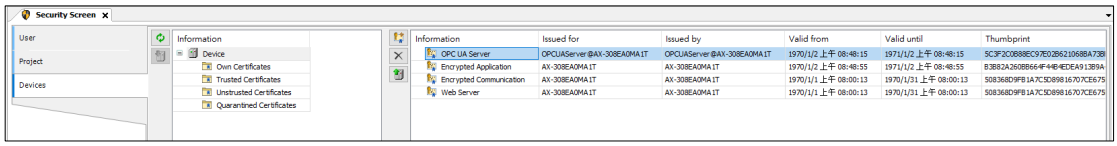
4. Create a new certificate for the controller: click **OPC UA Server**, click  , and set in the **Certificate Settings** dialog box.



5. Select **Encrypted Application**.



6. Complete the **OPC UA Server** and **Encrypted Application** certificate.

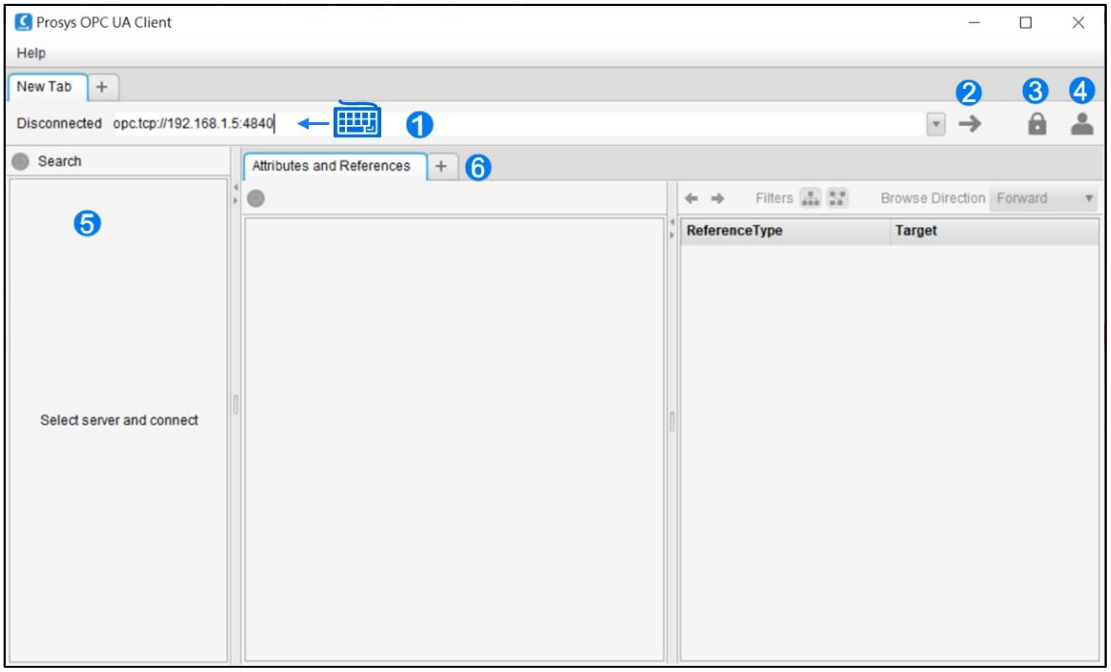


- **Use Prosys OPC UA Client to establish an encrypted connection**

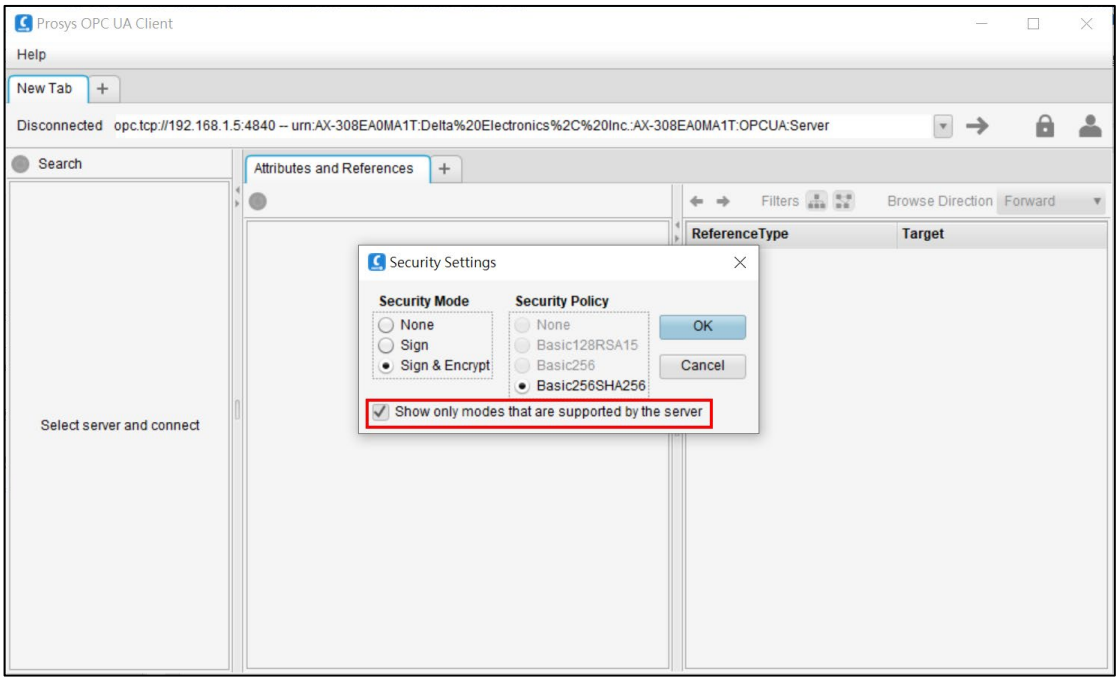
  Here with Prosys OPC UA Client Operation connection. User can first go to the official website of Prosys OPC UA Client to register for free download:

  Download URL：https://downloads.prosysopc.com/opc-ua-client-downloads.php

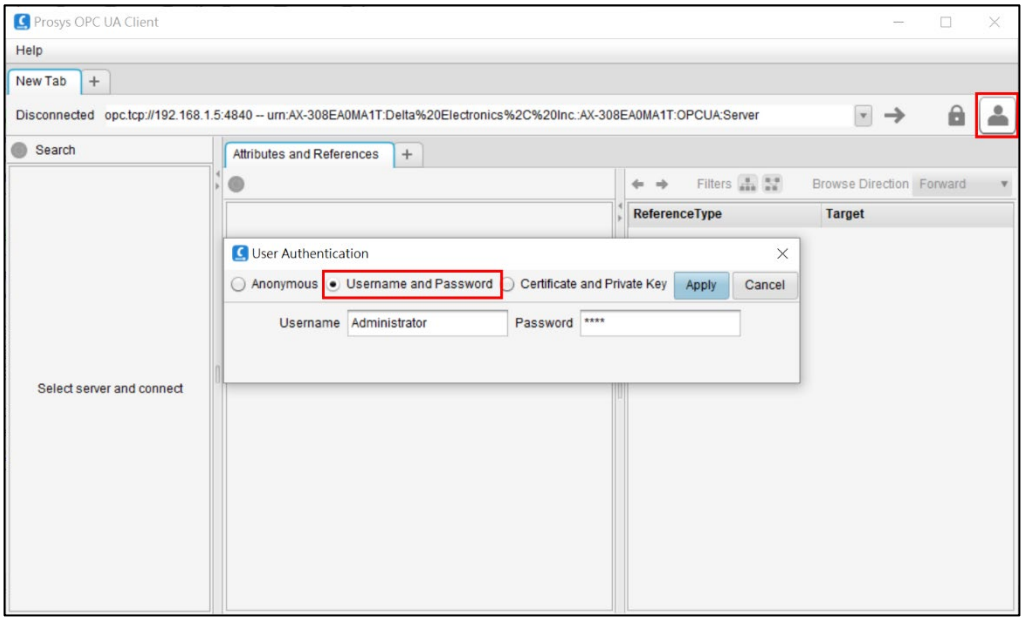
1. Launch Prosys OPC UA Client
2. In the picture below ① **Disconnected**, Enter the OPC UA Server IP address: **opc.tcp://192.168.1.5:4840**.



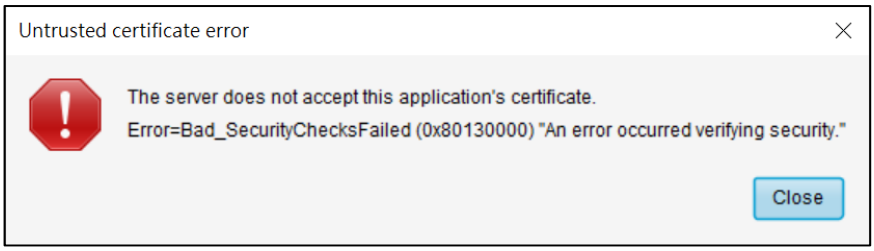
3. Select the connection encryption method ③  rear, **Security Settings** Window (only supports Basic256SHA256) check **Show only modes that are supported by the server**.



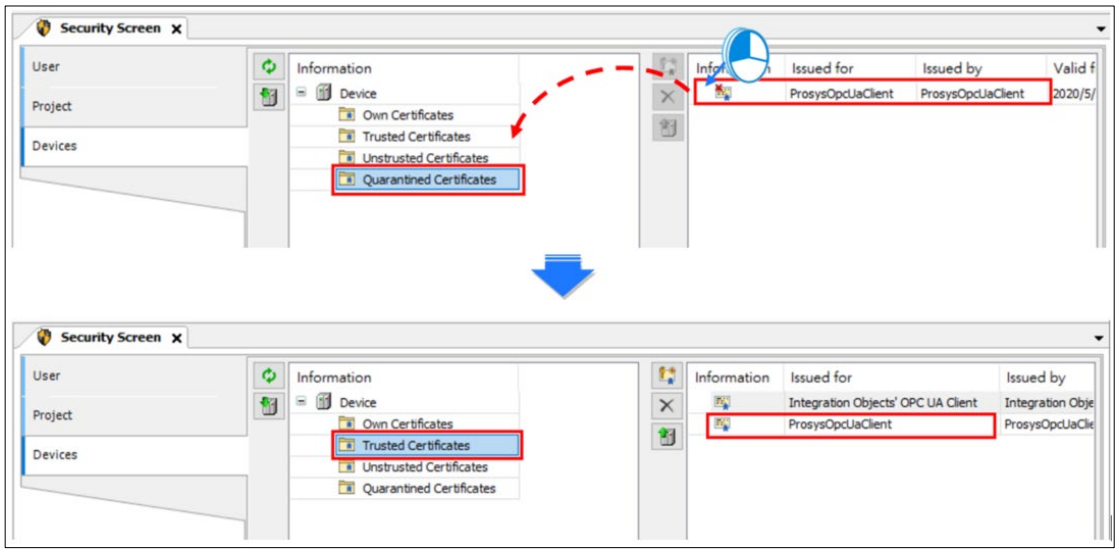
4. Click on ④  After, set the **user account password** > **Apply**, After setting, click ②  Connect.



5. Pop up the prompt window: no Server Allowed App Certificate.

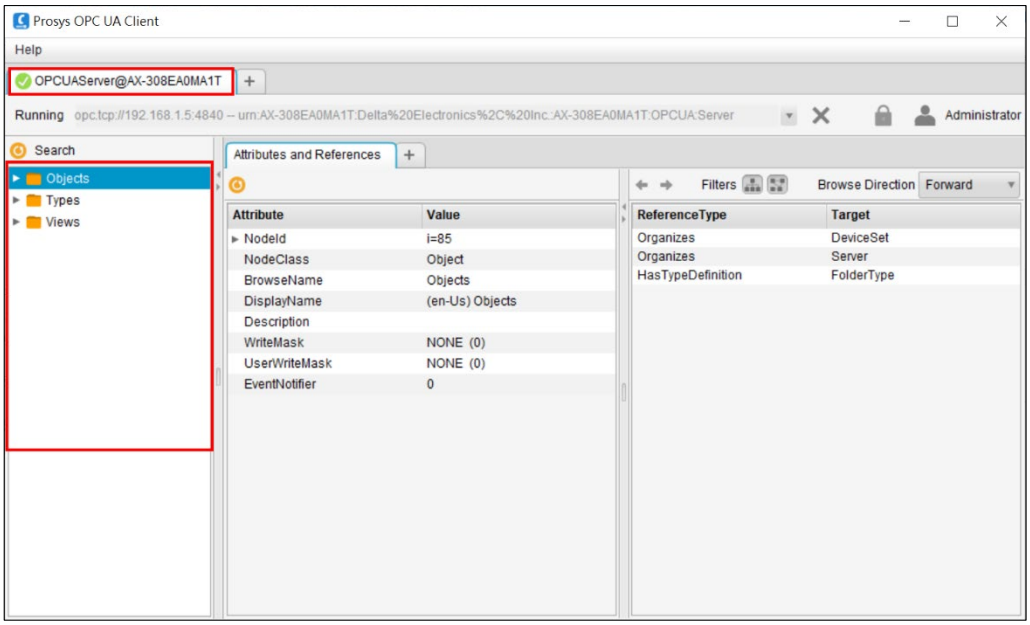


6. Back to the Safety fence page, there is an untrusted certificate ProsysOpcUaClient in Quarantined Certificates, drag this certificate to Trusted Certificates.

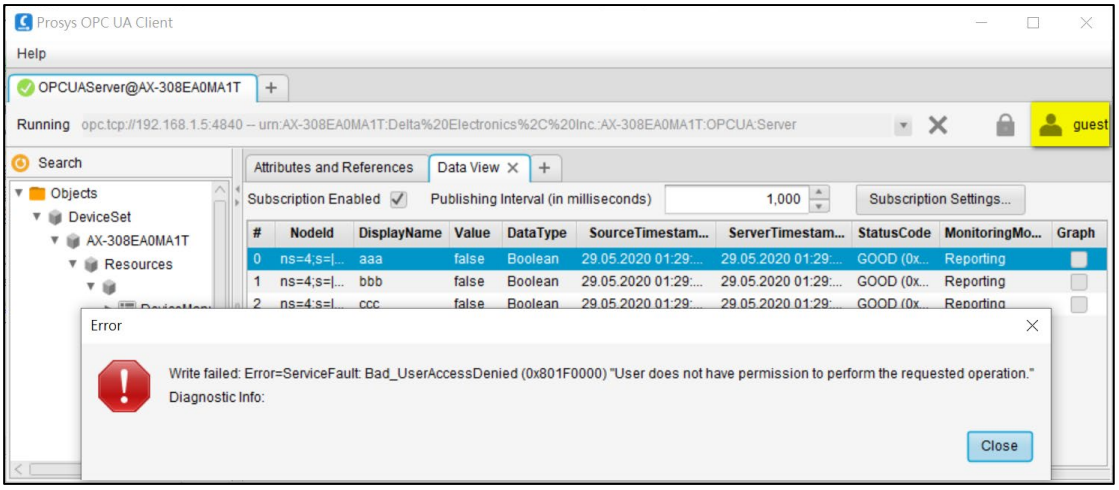


7. Client Reconnect to connect successfully. In the area ⑤, the tree structure organization of information connected to OPC UA Server for reading and writing is displayed. In the ⑥ area, the detailed description of the selected item is displayed. User can also add a paged Data View to pull the devices that user need to read and write to the page for monitoring.

   User：Administrator



User：guest (No write permission)



6.3.7 PROFINET

CODESYS PROFINET version 4.3.0.0 is supported in AX-3 series controllers version 1.0.5.7 and later, except the AX-332 controller.

Chapter 7: HMI Interactive

7.1 Overview

DIADesigner-AX supports exporting variables to DIAScreen. You can define the variables in DIADesigner-AX, export the symbol configuration XML file and import the XML file to DIAScreen to communicate between the controller and Delta HMI functions.

For more information about **Symbol Configuration**, see CODESYS Development System > Working with Controller Networks > Symbol Configuration.

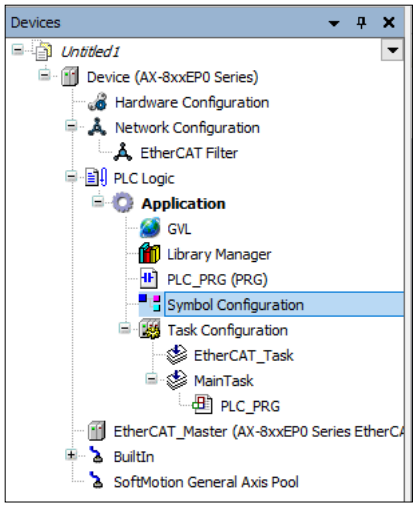
Note: HMI Tag sharing is supported only for the DOP-100 series.

7.2 Export the symbol configuration file

You can export variable information to an XML file for the OPC UA protocol to read variables.

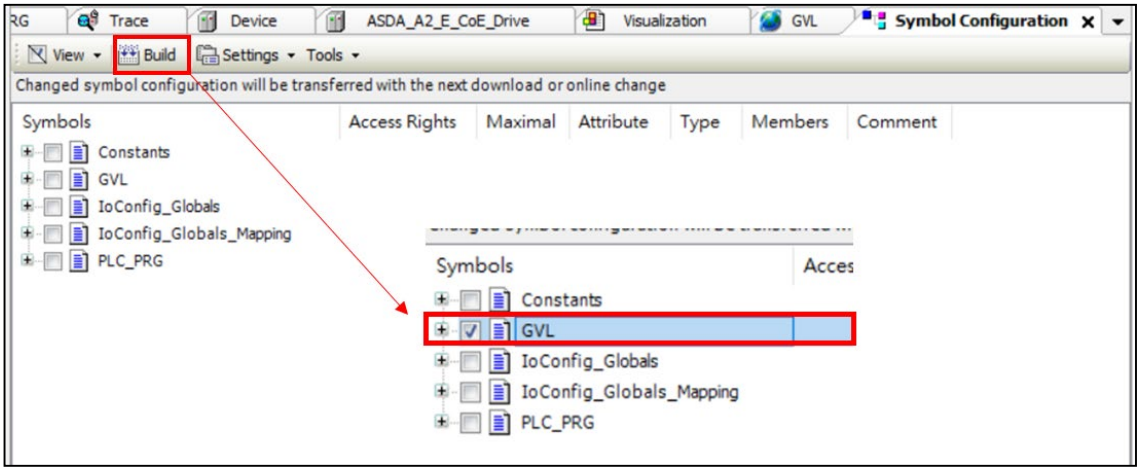
To export the symbol configuration file

1. On the **Devices** pane, right-click **Application**, and then select **Add Object** > **Symbol Configuration**.
2. In the **Add Symbol Configuration** dialog, select **Include comments in XML** and **Support OPC UA features**, and then click **Add**.
   For more information about the settings, see CODESYS Development System > Reference, User Interface > Objects > Object 'Symbol Configuration'.
   You can only add one **Symbol Configuration** in the same application.

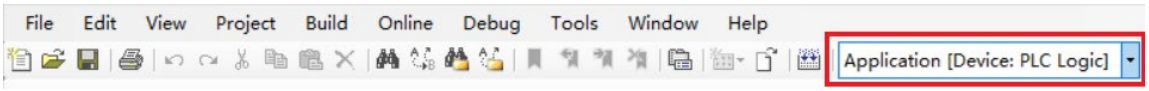


3. On the **Symbol Configuration** page, click 

or press **F11,** and then select the respective **variables** in the symbol list.



4. Generate the current XML file by clicking 

.
5. If there are multiple applications under the same device, to generate the symbol configuration XML file of the active device, do the following:
    1. On the quick access toolbar, select the active application.
    

    2. Do one of the following:
        - Click  next to it.
        - Press **F11**.
        - On the quick access toolbar, select **Build** > **Generate code**.

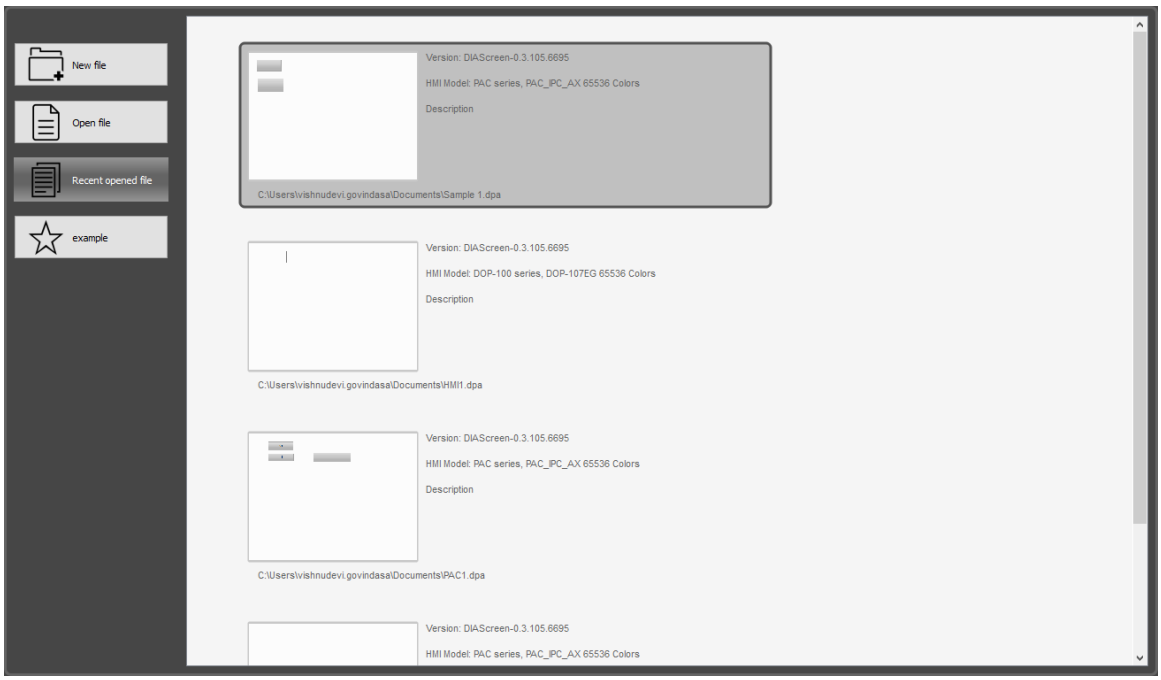
    The XML file will be stored in the same folder as that of the project. You can check the file path from **File** > **Save project as**.

    You can also find all the XML files in the .tgs file.

7.3 DIAScreen with AX-8 Series

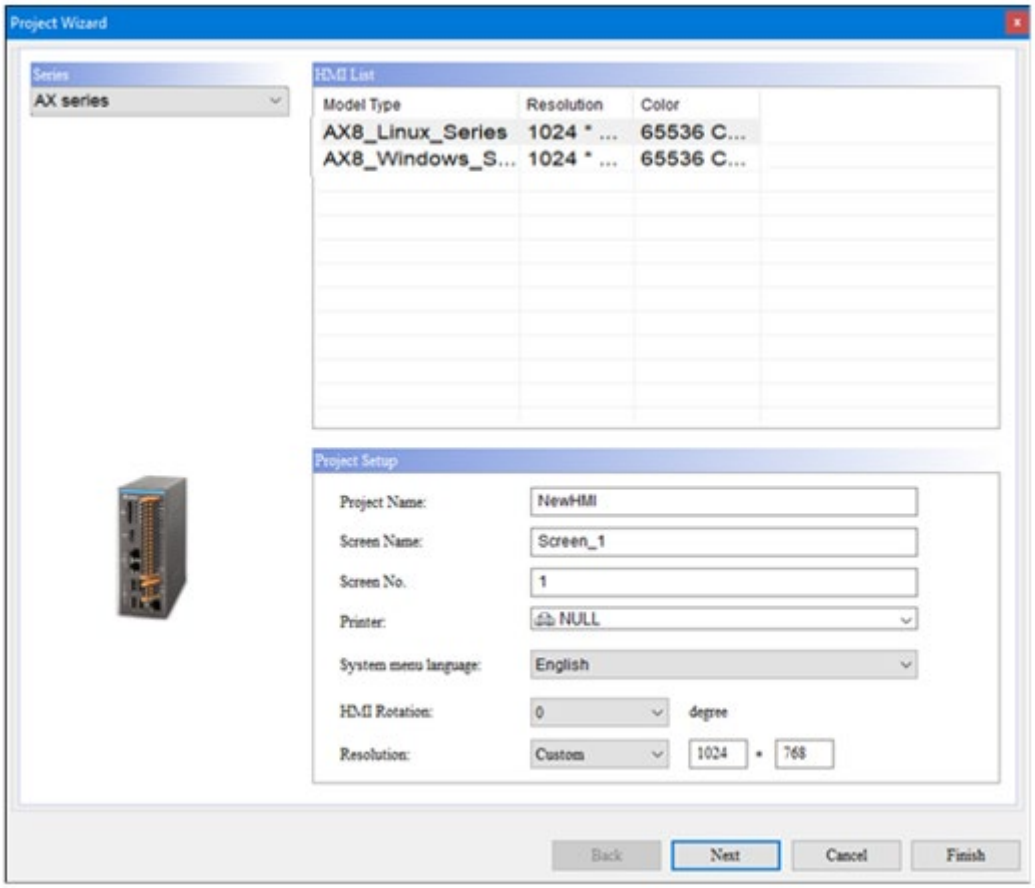
1. Launch DIAScreen.

   Result: DIAScreen displays the home screen as shown in the following figure.



2. Click **New file**.

   Result: A **Project Wizard** window displays as shown in the following figure.



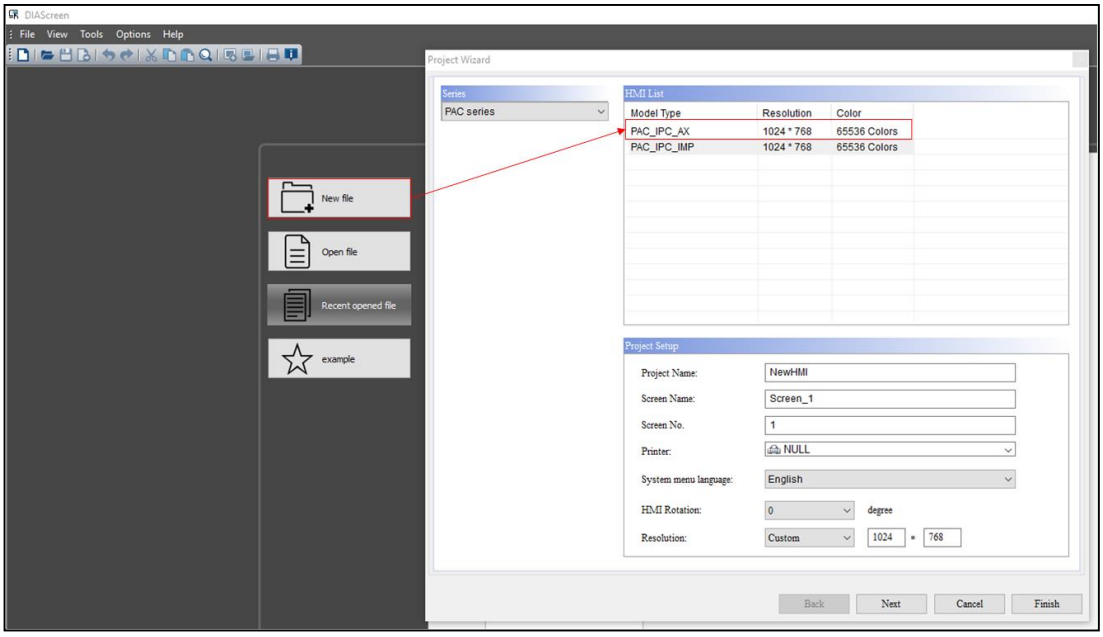
3. Select **AX series** in **Series** and **AX_IPC_AX Model** in **Model Type**.

   Set the following settings in the new project wizard window:

   - Select **AX series** from the drop-down menu of **HMI series**.
   - Select **AX8_Windows_Series** in the **HMI List**.

4. Enter **Project Name, Screen Name, Screen No., Printer, System menu language, HMI Rotation** and **Resolution i**n the **Project Setup** box.Click **Next**.

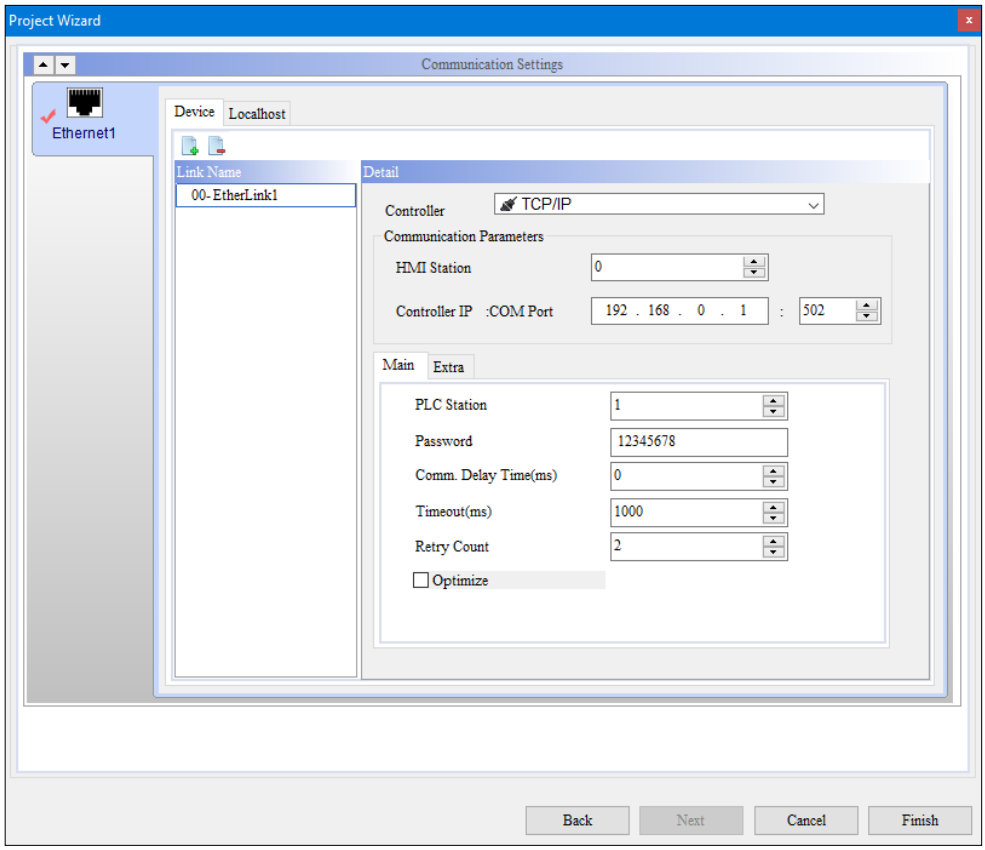
   Result: A **Project Wizard** window displays as shown in the following figure.



5. Enter controller details and communication parameters and click **Finish**.

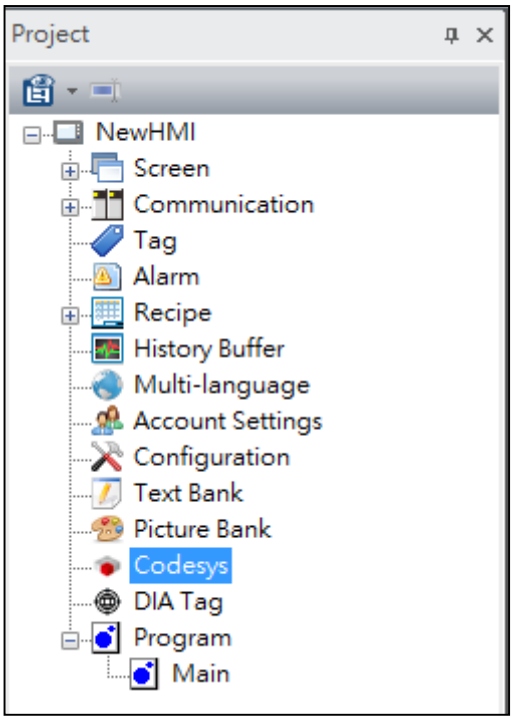
   Result: A **Communication Settings** displays as shown in the following figure.

   Note: Use **New Network Link**  to add Ethernet if required.

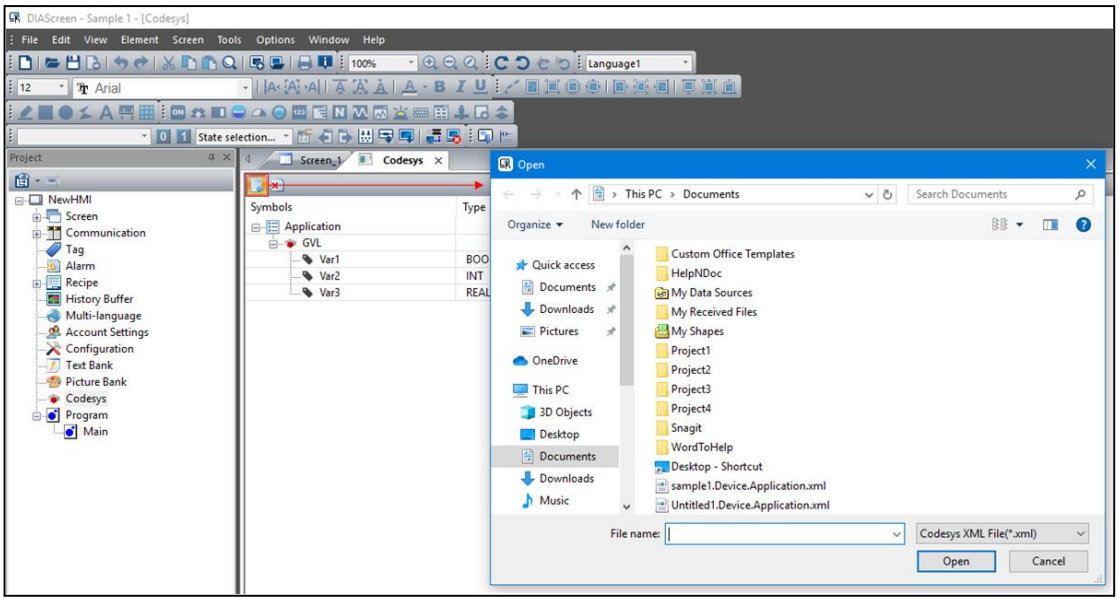


6. Click **New HMI** in the screen.

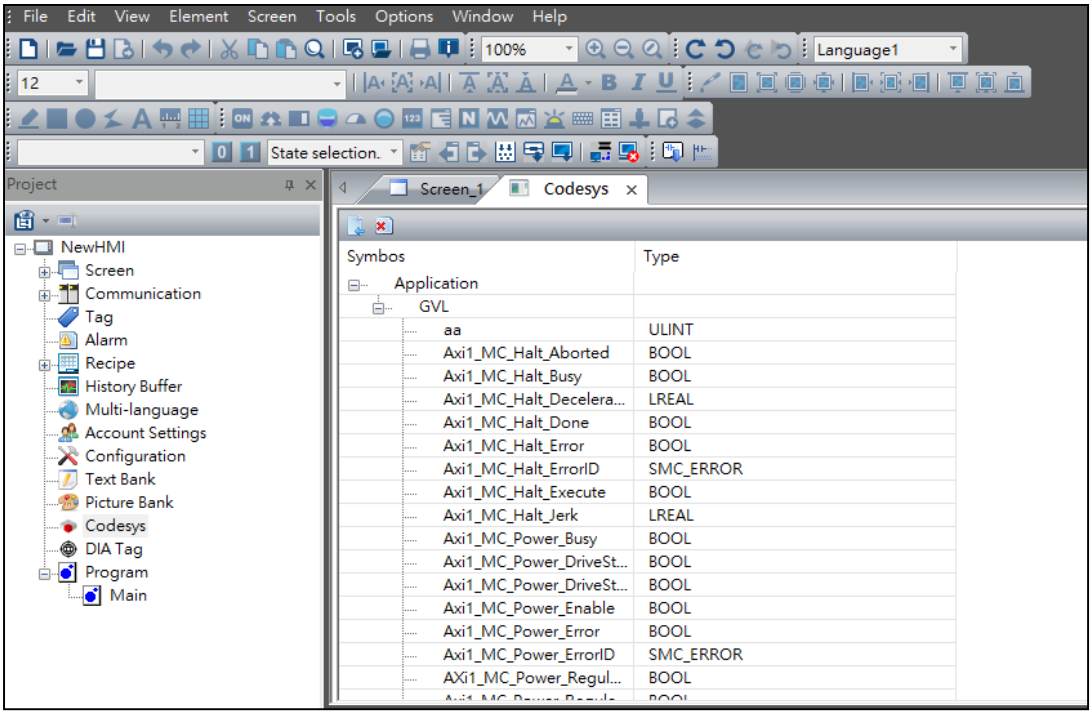
Result: User can view **CODESYS** in the project tree.



7. Click on **CODESYS** > **Import icon** > select the respective **XML file** > **Open**.



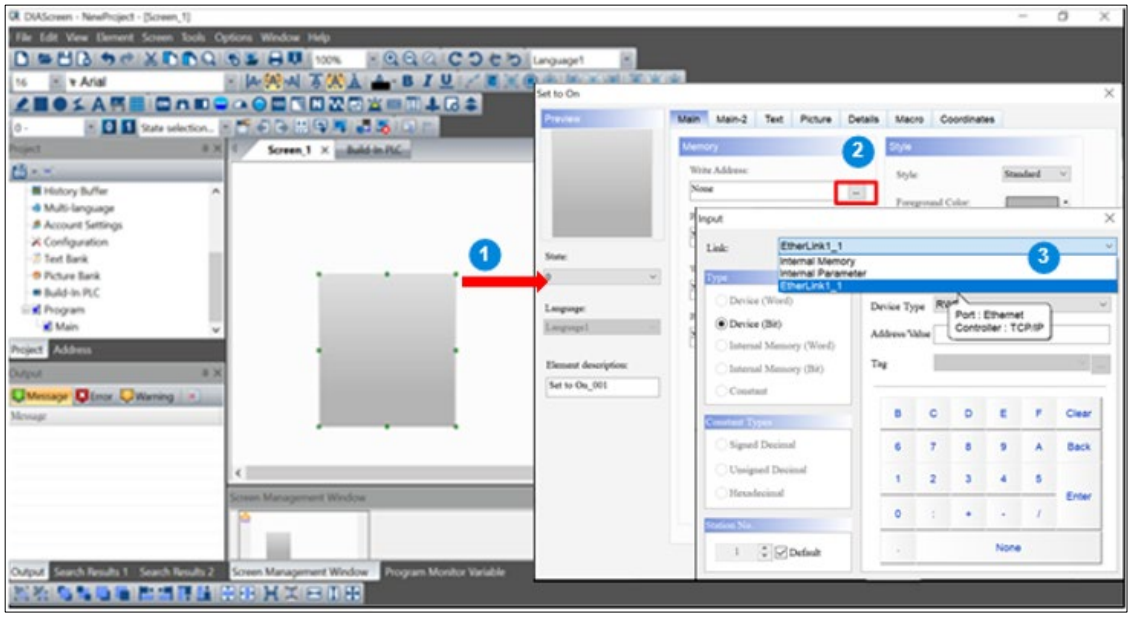
Result: The Imported symbols appears on DIAScreen.



8. Use any of the buttons on the element and double-click the element.

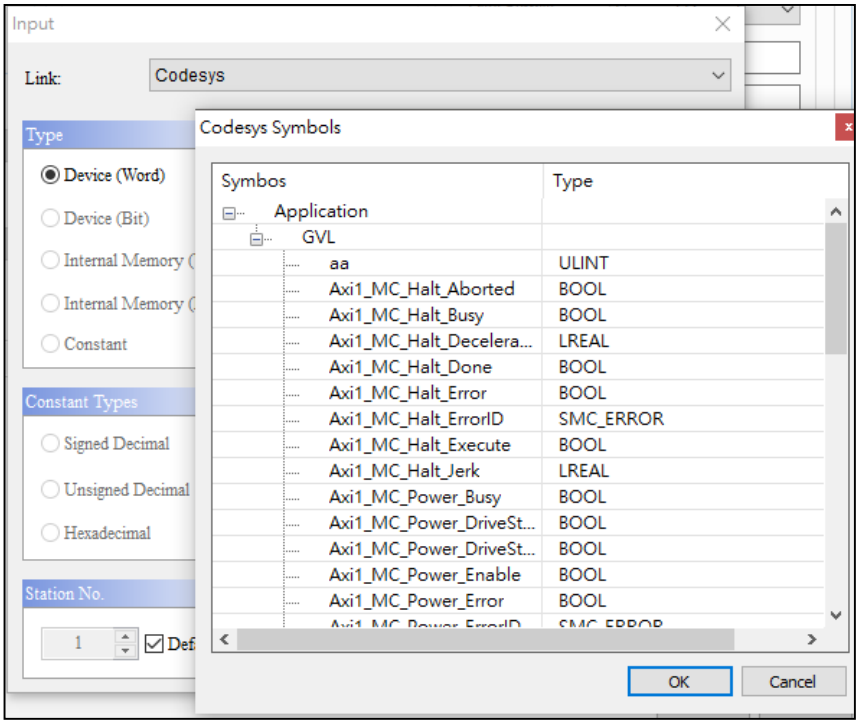
   Example: The **Set to on** button shown in the following figure.

   Note: User can choose the CODESYS parameter to connect with the element.

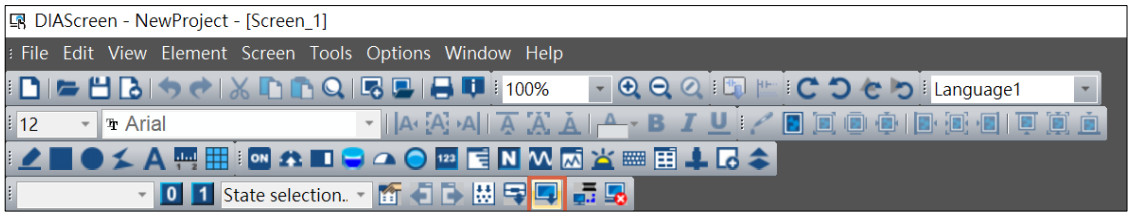


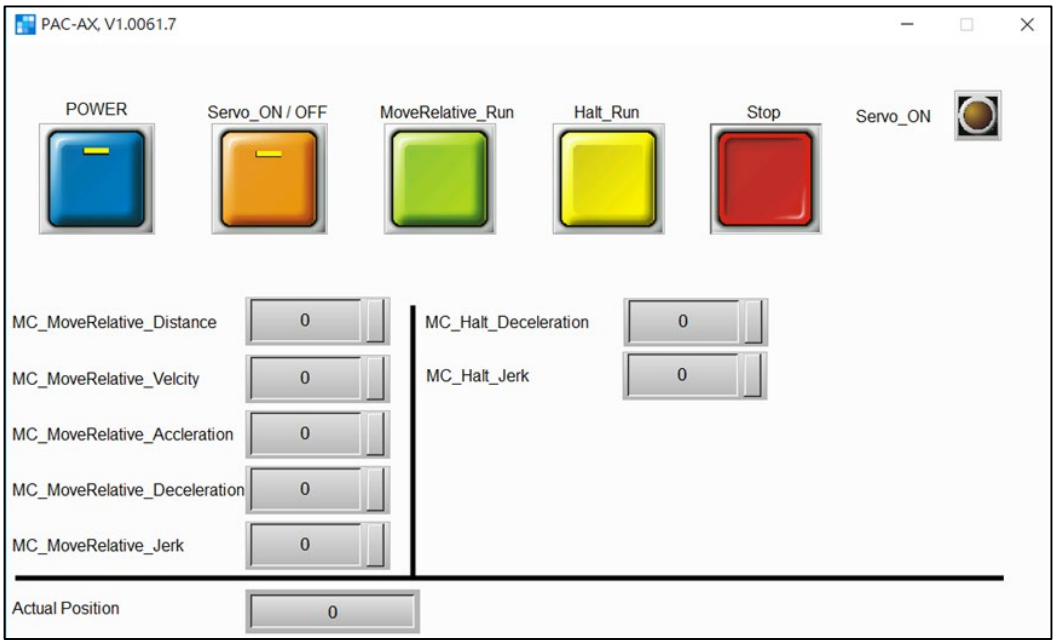
9. Select **CODESYS** in Link.

   Result: CODESYS is linked with the element.



10. After configuration screen, click **Download** and choose the model to download the screen.

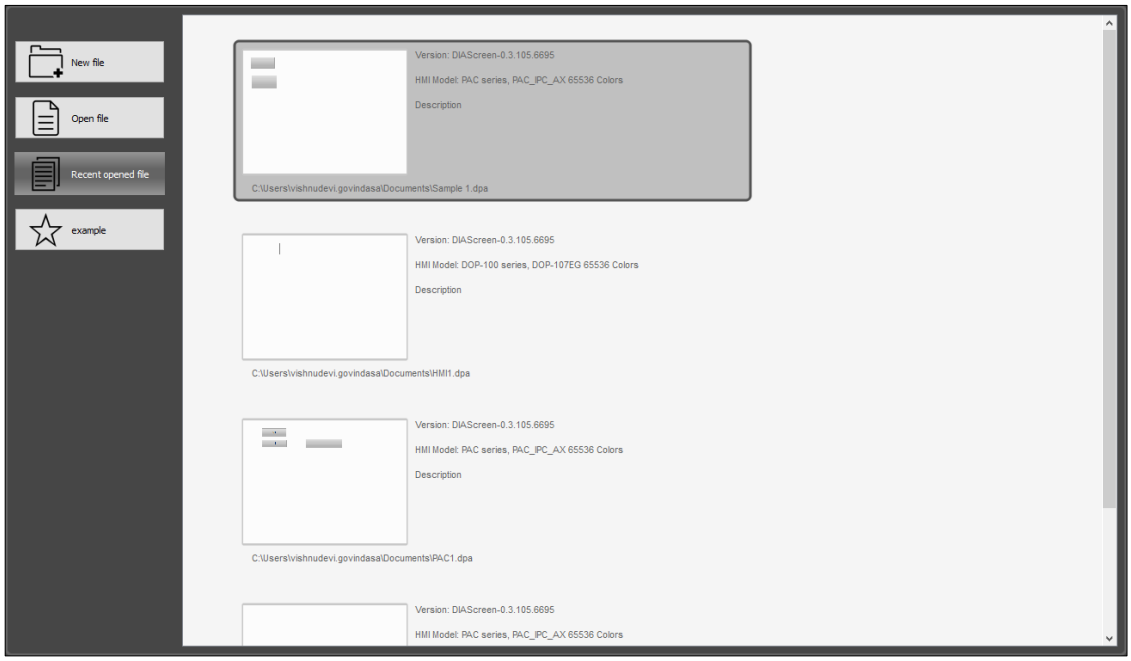




7.4 DIAScreen with PLC AX-308E Series

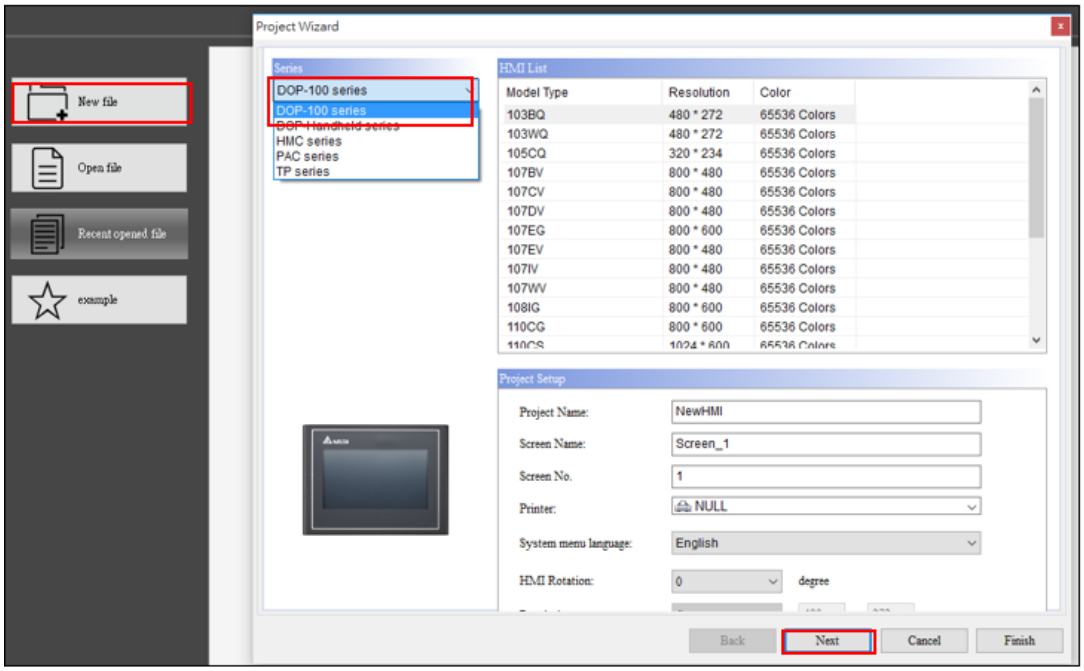
1. Launch DIAScreen.

   Result: DIAScreen displays the home screen as shown in the following figure.



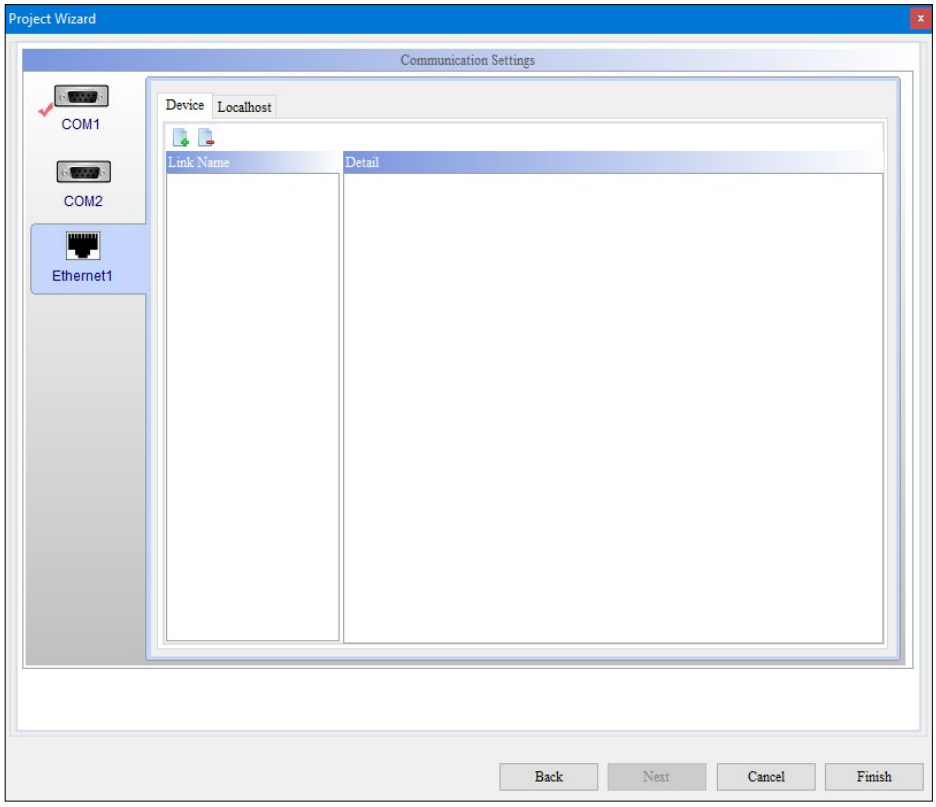
2. Click **New file** in the screen.

   Result: A **Project Wizard** window displays as shown in the following figure.

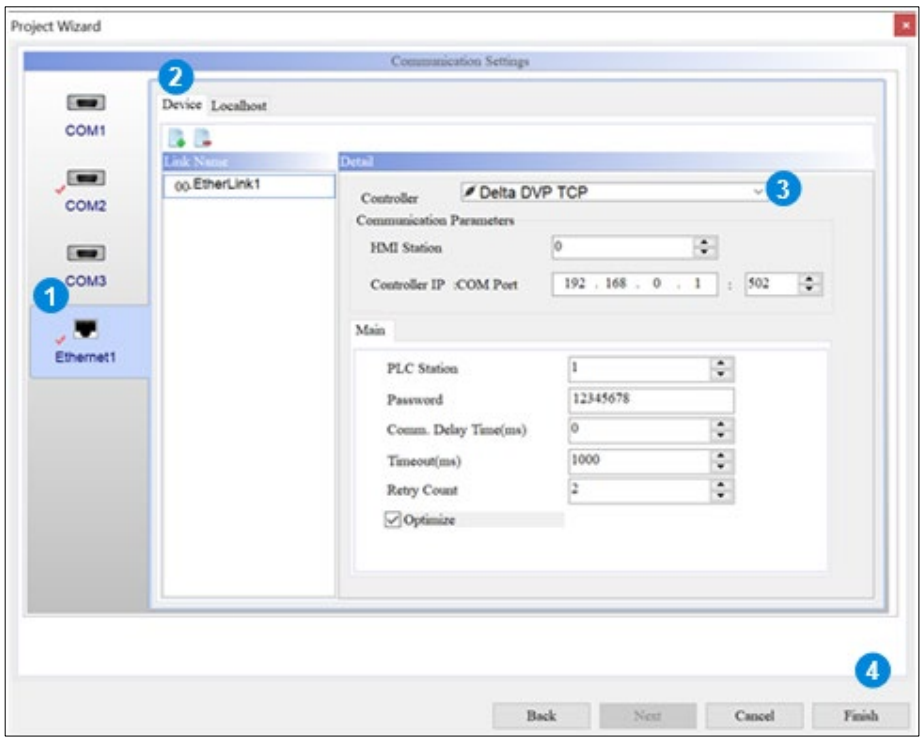


3. Select **DOP-100 Series** and the Required Model type in HMI list. Click **Next**.

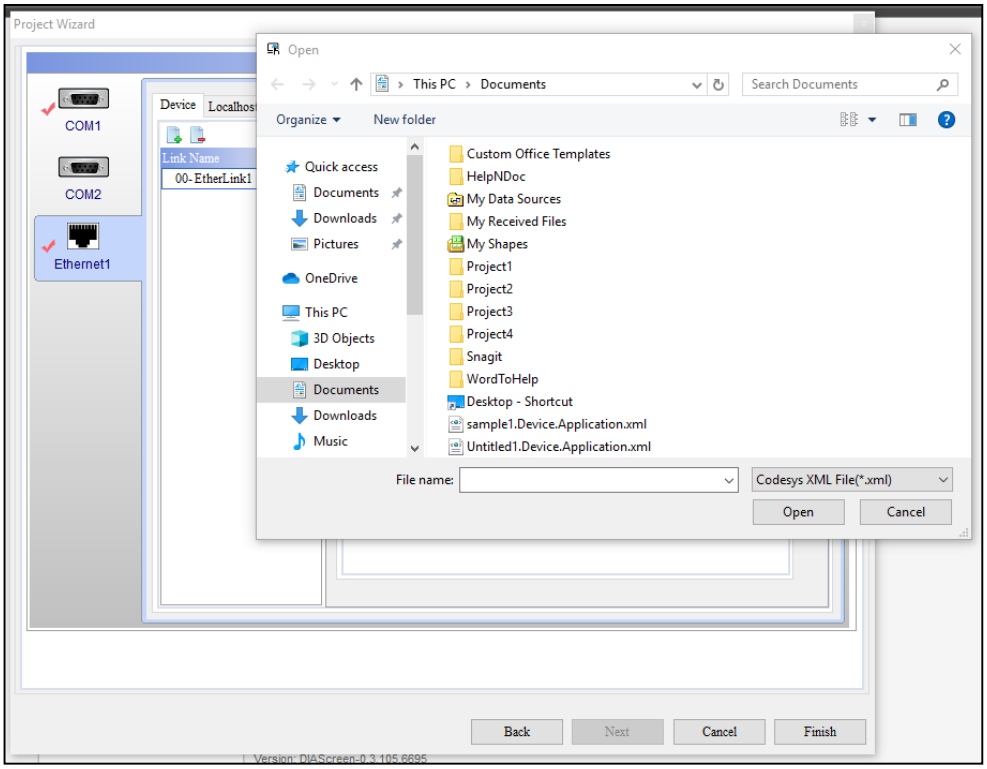
Result: A **Project Wizard** window displays as shown in the following figure.



4. Click **Ethernet1** > icon. Select **CODESYS** in controller. Click **Import**.



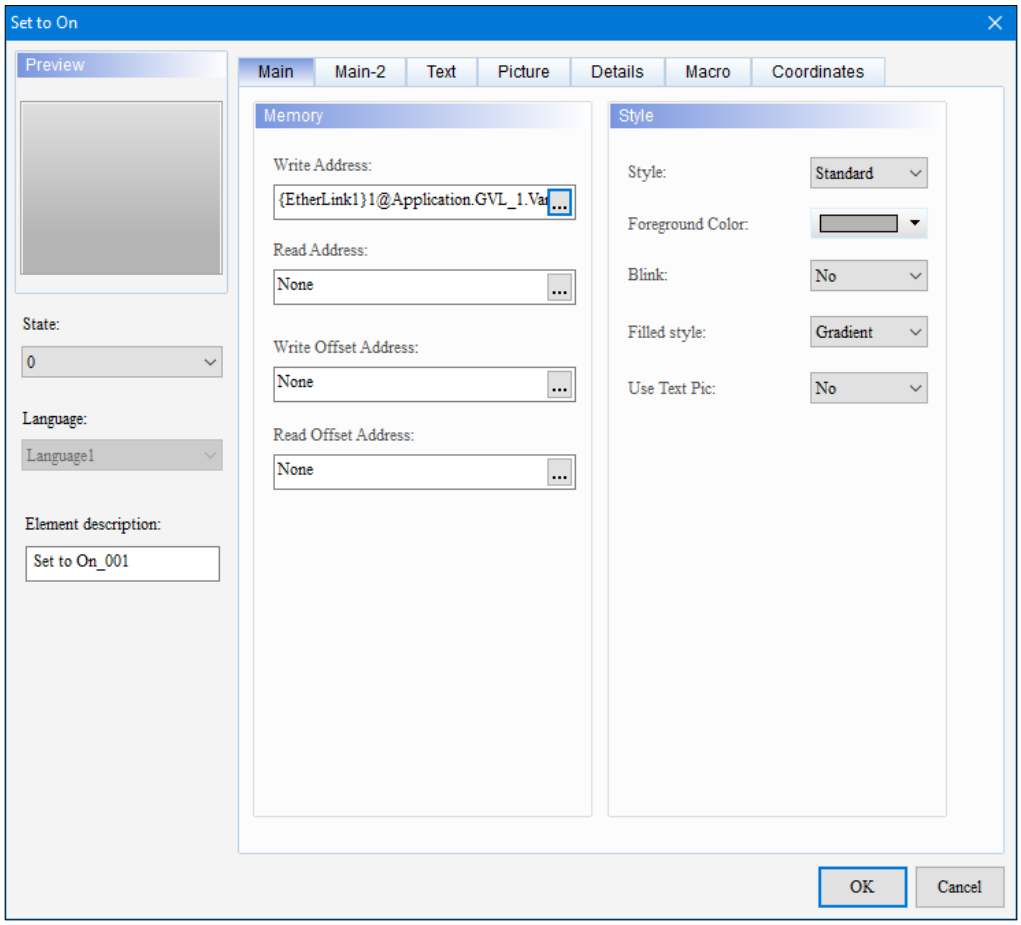
5. Click **Controller** drop-down menu and click **Open**.

6. 

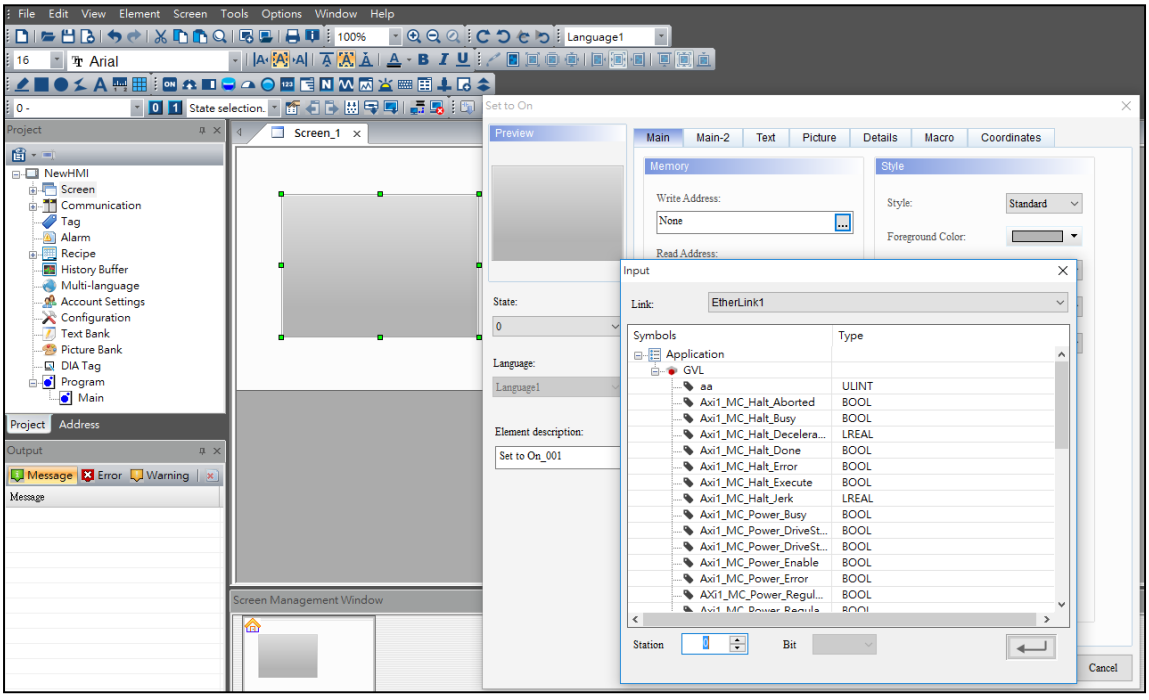
7. Use any button to set the input.

   Example: The **Set to On** is used in the following figure. Enter the required details.

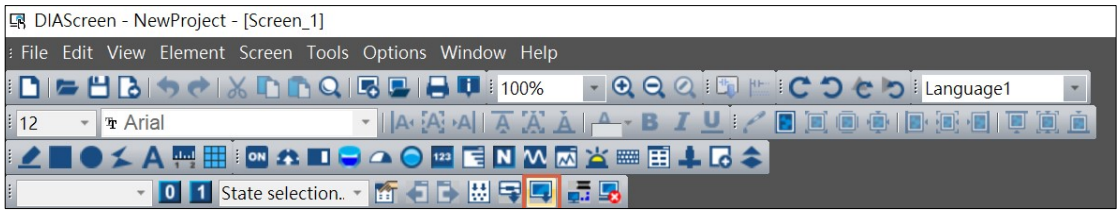
   Click **OK**.



Result: CODESYS is linked with the element.



8. After configuration screen, click **Download** and choose the model to download the screen.


DIAScreen - NewProject - [Screen_1]
File Edit View Element Screen Tools Options Window Help
100%
Language1
12
Arial
State selection..


POWER
Servo_ON / OFF
MoveRelative_Run
Halt_Run
Stop
Servo_ON
MC_MoveRelative_Distance
0
MC_MoveRelative_Velcity
0
MC_MoveRelative_Accleration
0
MC_MoveRelative_Deceleration
0
MC_MoveRelative_Jerk
0
MC_Halt_Deceleration
0
MC_Halt_Jerk
0
Actual Position
0

7.5 Upload and download

The program files can be uploaded and downloaded in the PLC used by the HMI, various running projects can be changed for different program requirements. Upload and download the Application file in the AX series PLC used. After Clicking search, the currently connected PLC device is automatically searched. After the search is over, the searched files can be uploaded to the storage device.

To copy DVP and DIADesigenr-AX files from the storage device to the PLC, click Download.

The Upload Download fetaure, currently supports the following hardware series:

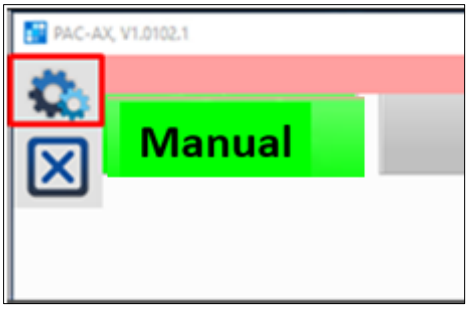
- DOP-100 series
- AX-8 HMIAP

Upload

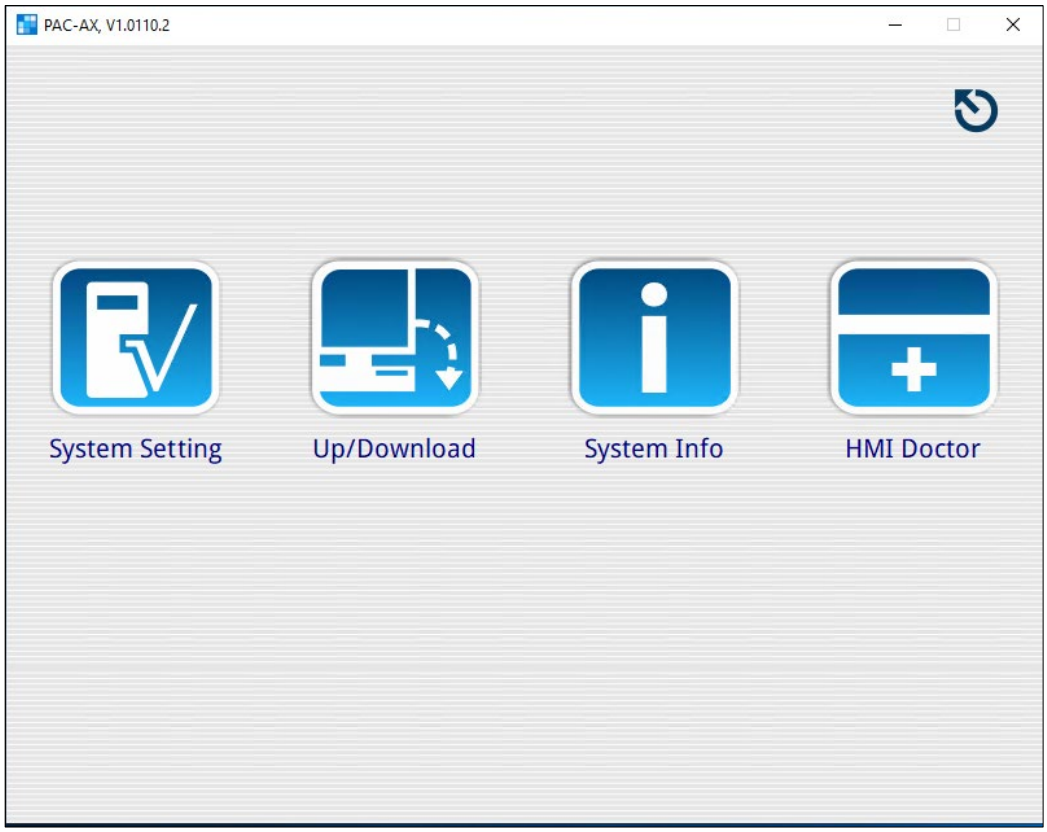
Follow the procedure to upload the Application file.

1. Enter the system screen:

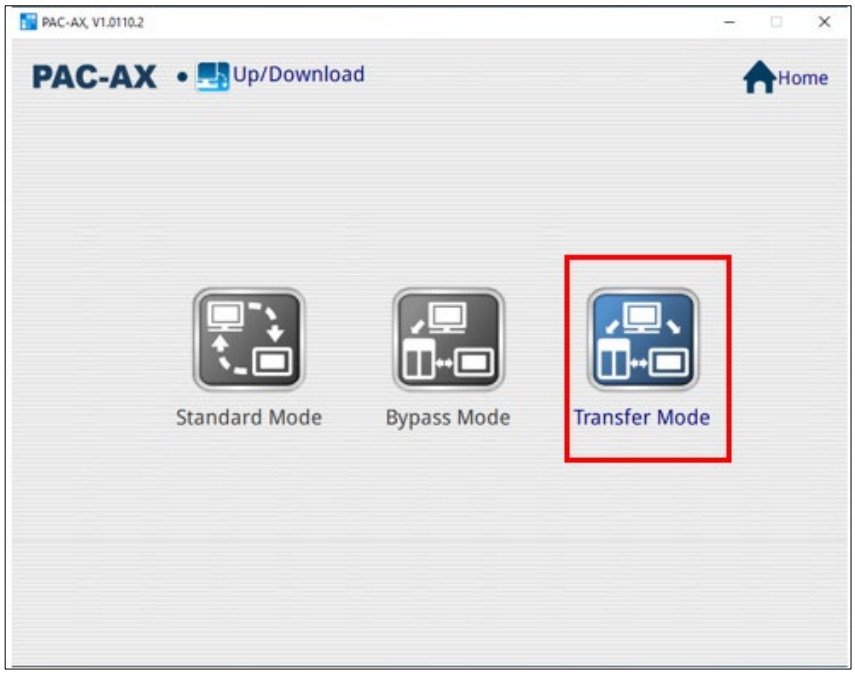
   Press the blank area of the DOP-100 HMI or HMIAPP screen for more than 3 seconds, then click on the system setting icon to enter the system screen.



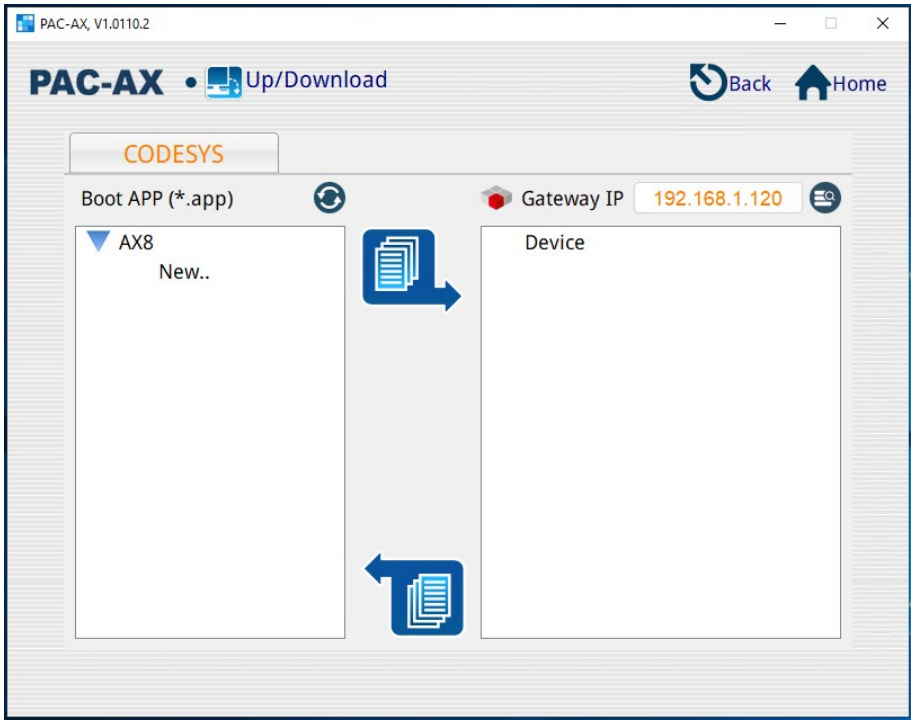
2. Click on **Upload/Download**.



3. Click on **Transfer Mode**: This item is for the Application file in the PLC used. CODESYS Application file only supports **Transfer Mode**.



4. Click on the **CODESYS** tab.



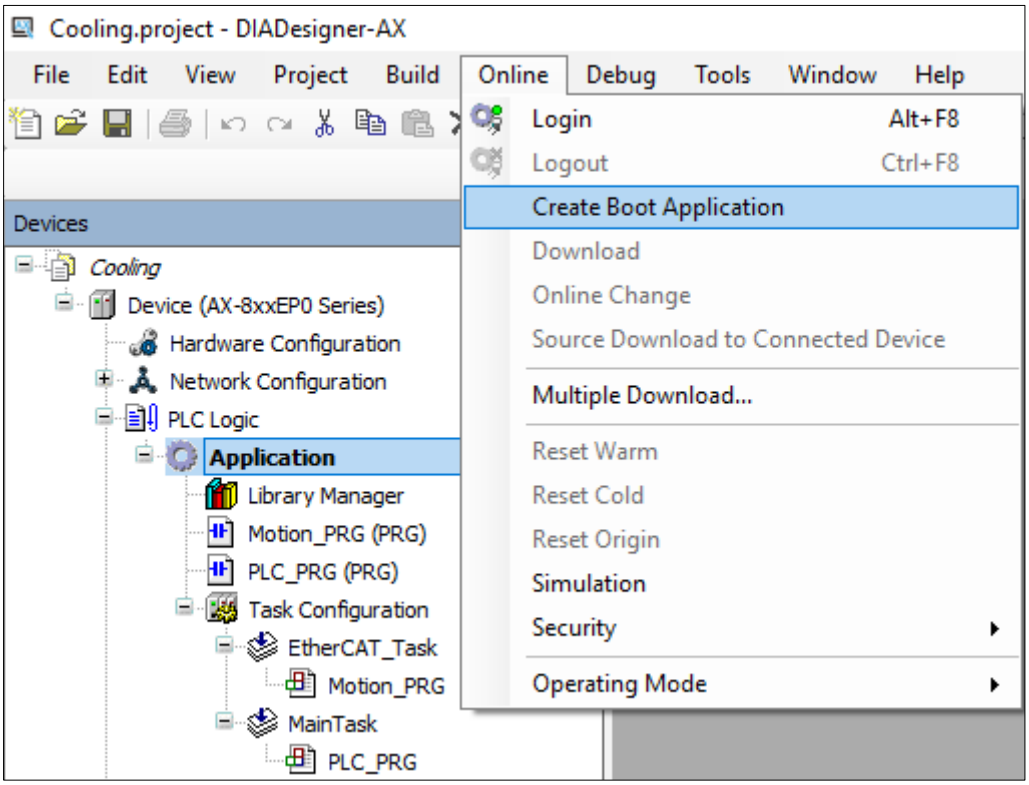
Project	Description
Boot APP	
Path	DOP supports USB flash drives. AX-8 supports D:\Delta Industrial Automation\PLC-APP.
	Refresh the Application file in the USB or D slot.
Gateway IP	
IP Set up	Set the value of PLC IP.
	Search for PLC devices in the same network domain.
Download	
	Download the App. file.
	Upload the App. file.

5. Upload the App. File.

The App file can be uploaded in two Methods:

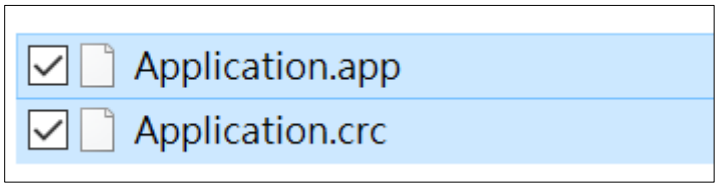
Method 1:

Create .app file: Open the CODESYS project, click on **Online** > **Create Boot Application**.



Copy and save the .app and .crc files in the project directory to the following path:

- DOP-100 series: USB flash drive
- AX-8 HMIAP: Path D:\Delta Industrial Automation\PLC-APP

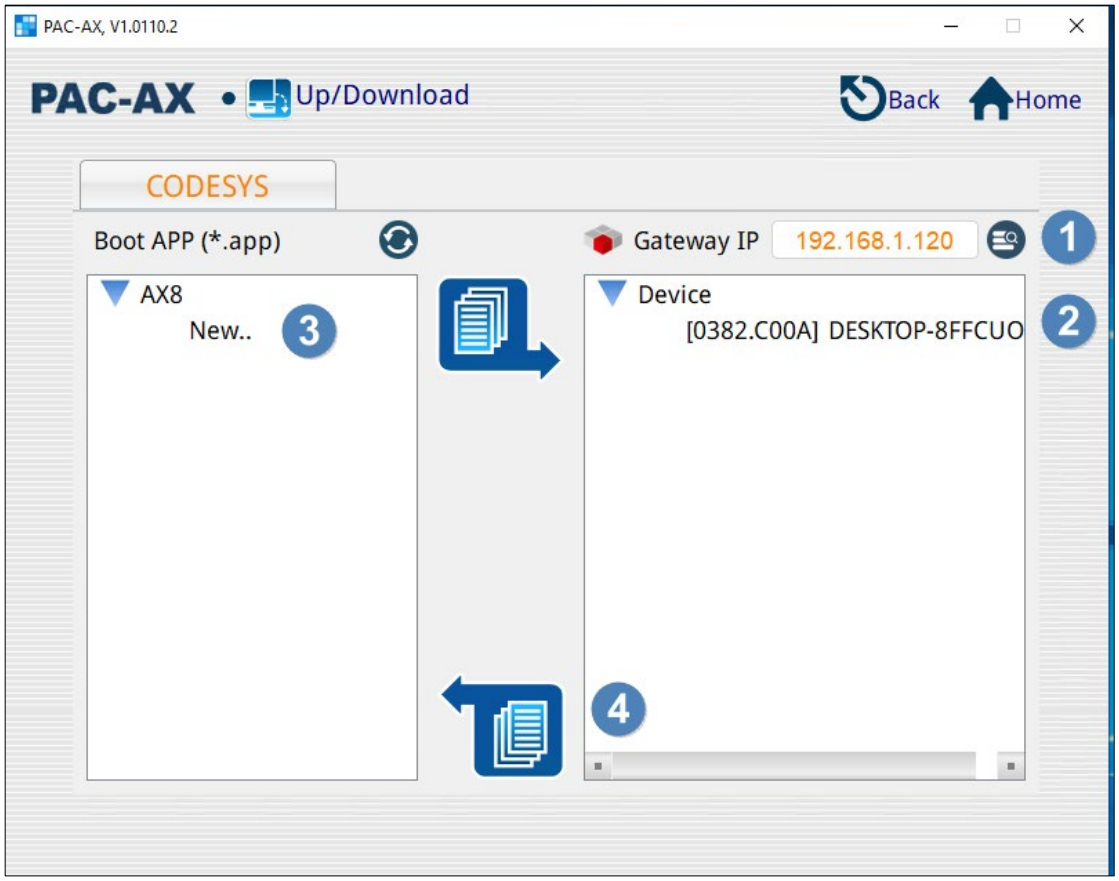


Method 2:

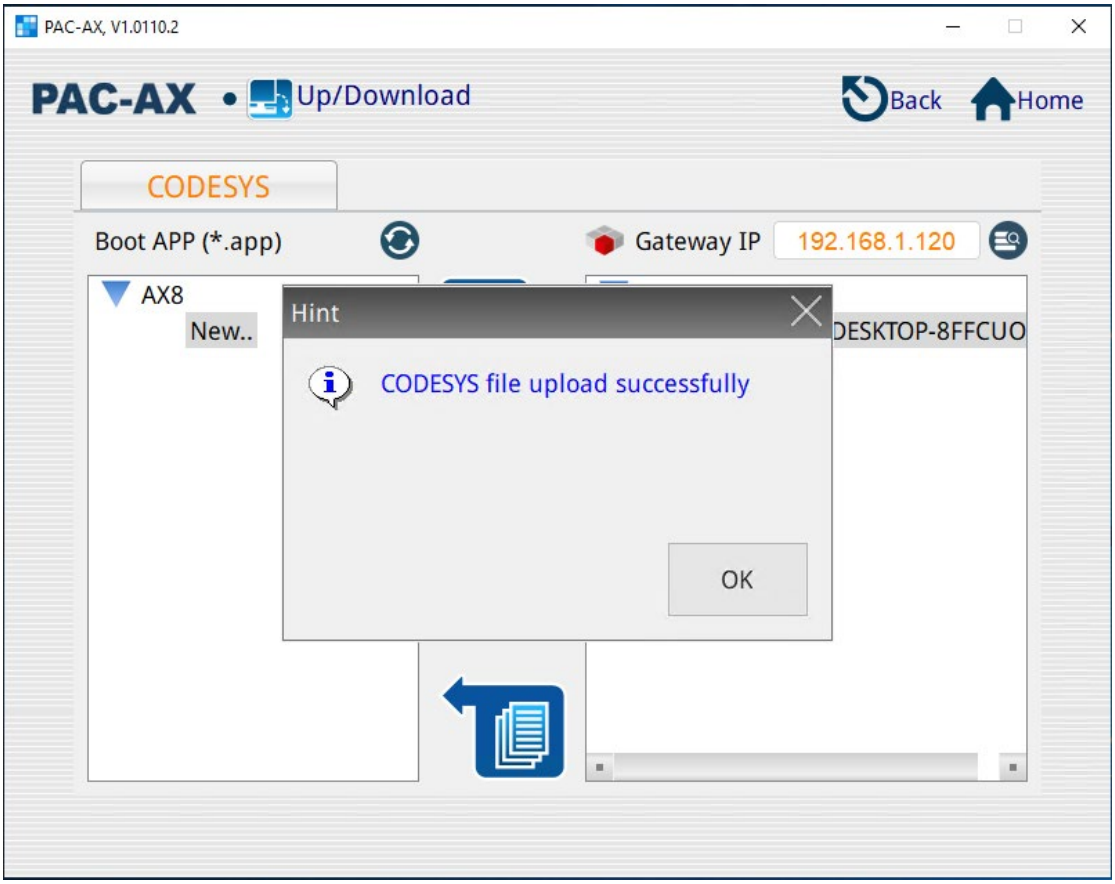
Follow the steps to upload the .app file directly via PLC.

1. Click on  icon to search the PLC device to be uploaded. The PLC needs to be registered and activated in the CODESYS project.
2. After finding the device, select the PLC device to upload the app. file.
3. Select the folder path to upload.

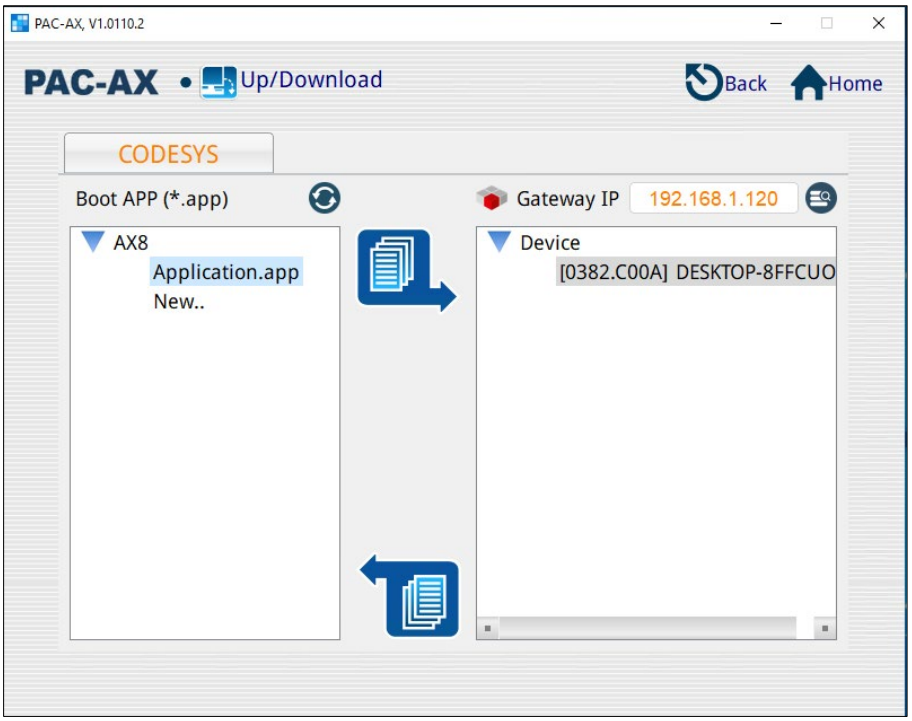
4. Click on upload  icon.



Result: Displays file upload success message.



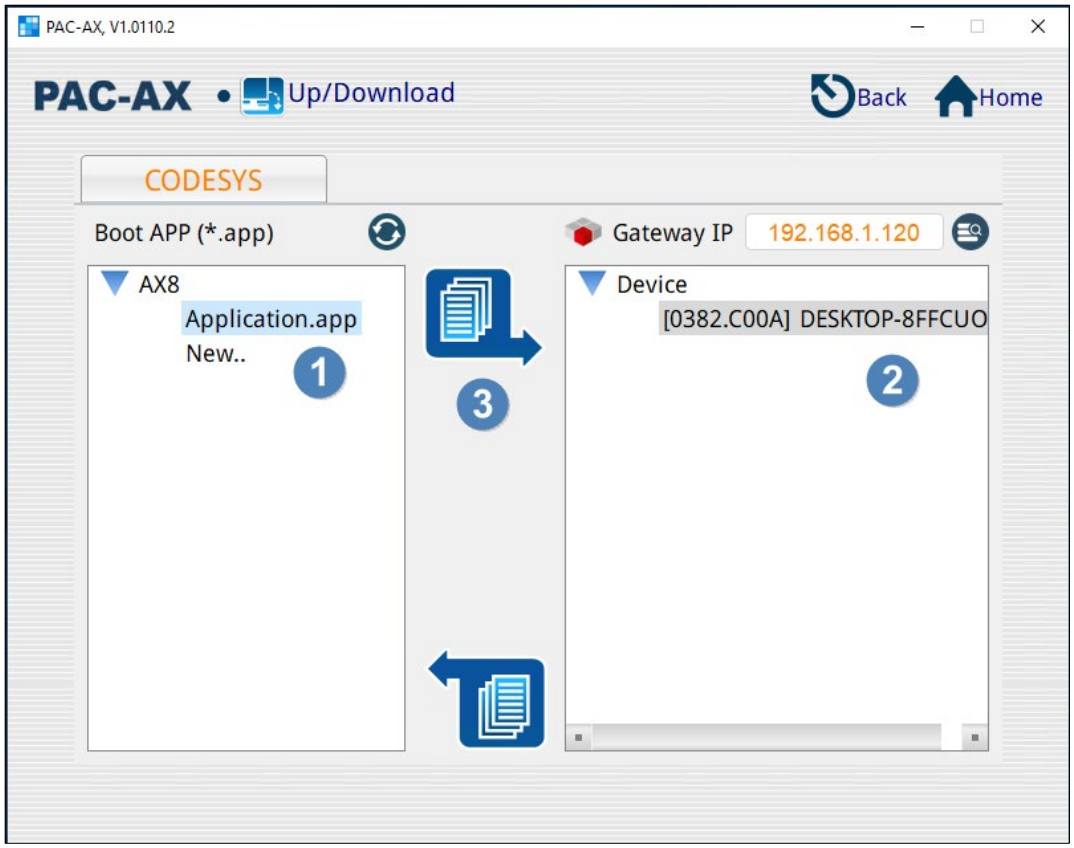
Note: After the upload is complete, there will be more .app files in the path.



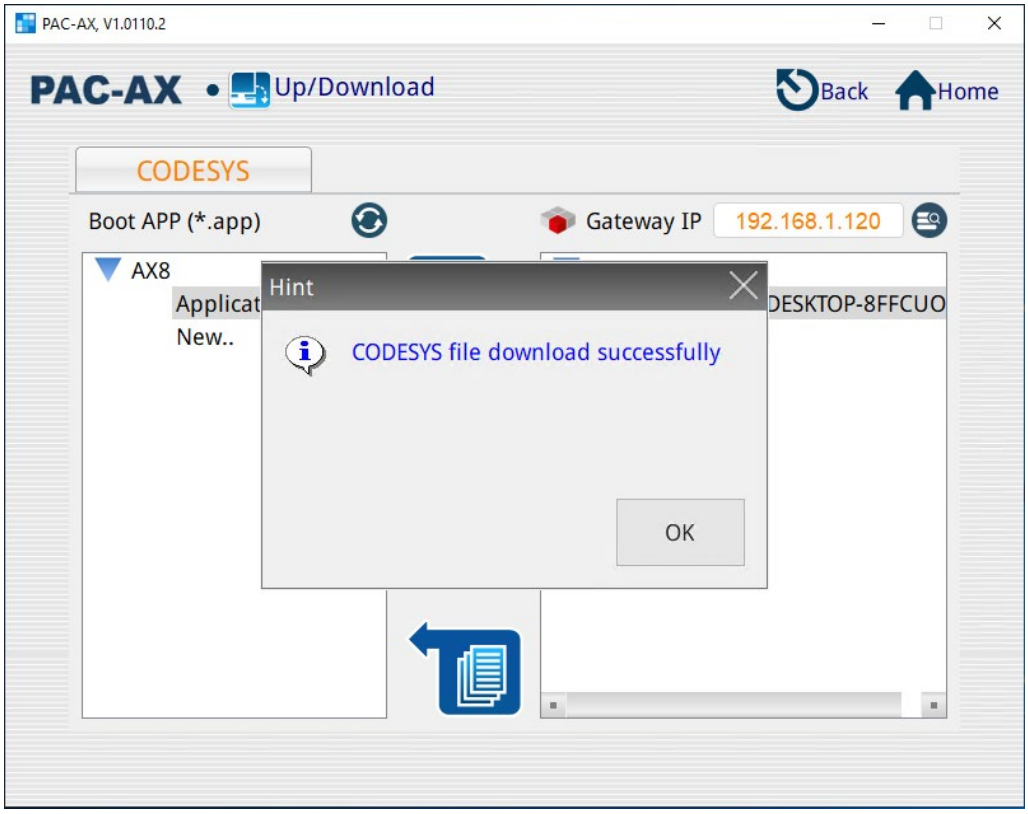
Download

Follow the procedure to download the .app file.

1. Select the .app file to download.
2. Select the PLC device to be downloaded.
3. Click on download  .



4. A message after the download is successful, the newly downloaded Application program will run on the PLC.


PAC-AX, V1.0110.2
PAC-AX
Up/Download
Back
Home
CODESYS
Boot APP (*.app)
Gateway IP
192.168.1.120
AX8
Applicat
New..
DESKTOP-8FFCUO
Hint
CODESYS file download successfully
OK

Chapter 8: Tools and configuration options

8.1 Device Repository

How do I open Device Repository?

- Select **Tools** > **Device Repository.**

The device repository is used to manage the device description files installed in the local system. You can also install a third-party device description file in **Device Repository** and add the device to the project.

Use the following table to configure the device repository dialog.

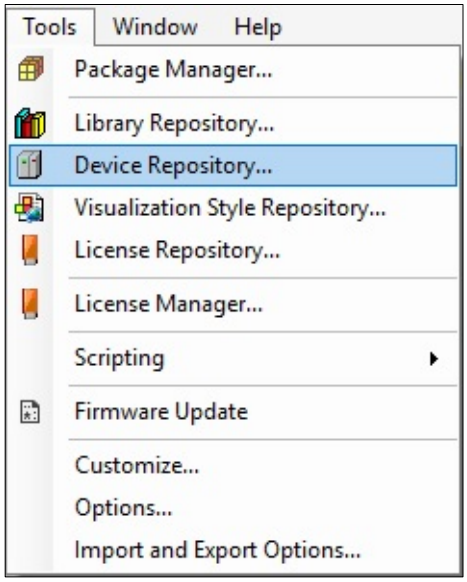
Setting	Description
Location	Select the device repository directory on the local system and the save location currently set.
Edit locations	Click to edit repository locations.
Vendor	Select the manufacturers, and then the available devices will show.
Install	Click to install the device description file.
Uninstall	Click to remove the selected device.
Export	Click to export the device description file.
Details	Click to show detailed information of the selected device.

8.1.1 Install device description file

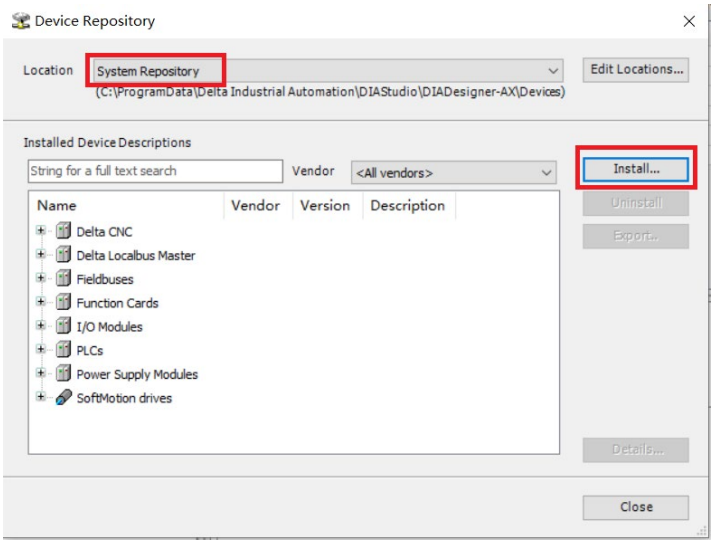
Use the following steps to install device description file.

To install the device description file

1. Select **Tools** > **Device Repository**.



2. In the **Device Repository** dialog, select **System Repository** as the location, and then click **Install**.



3. In the **Install Device Description** dialog, select the file type (recommend selecting **Automatic detection**), and then select the file to be installed and click **Open**.

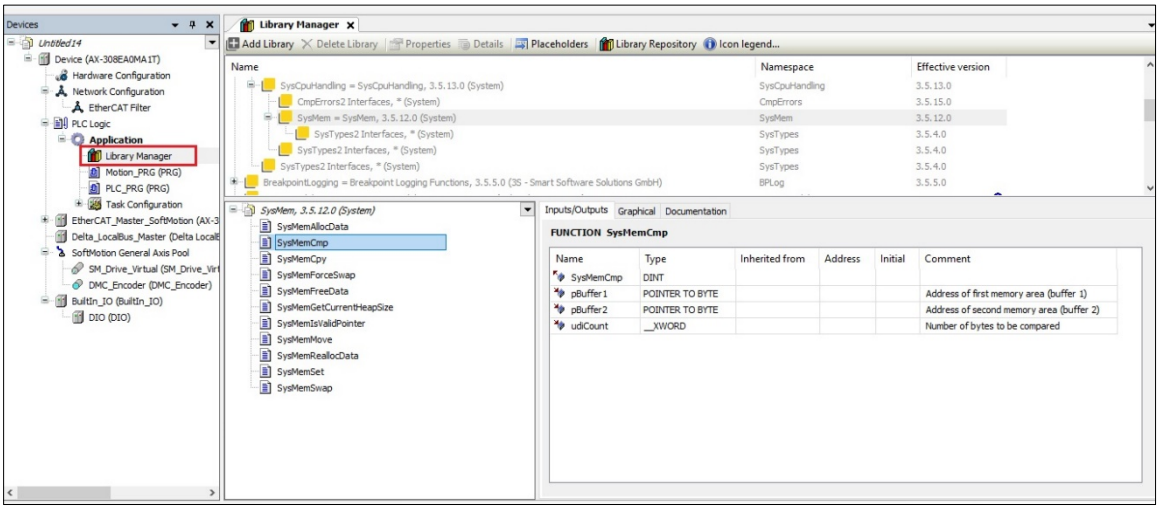
After the installation, you can find the new device description file under the device component it belongs to, and it is available when you add this type of device to the project.

8.2 Library Manager

The library memory is the storage location of the upper programming system library and related metadata. Insert any installed library into the project through the library manager. When installing the library, the version manager will facilitate updating the latest library.

In addition to the pre-installed system and motion control libraries, users can also create and edit more libraries.

The library manager is responsible for managing the libraries used in user projects. Display the library type and attributes according to the corresponding device. User can use more libraries from the library repository, delete them or edit their properties.



The library manager lists all libraries integrated in the project for creating applications. It provides information about the library type, its properties and content. User can expand or collapse the list of integrated libraries, and edit the library properties of non-dependent libraries.

The library manager contains three blocks：

- Upper block: List of integrated libraries.
- Lower left block: tree structure, all modules of the library are selected in the upper view.
- Lower right block: The input/output, graphics and files of the module selected in the tree structure.

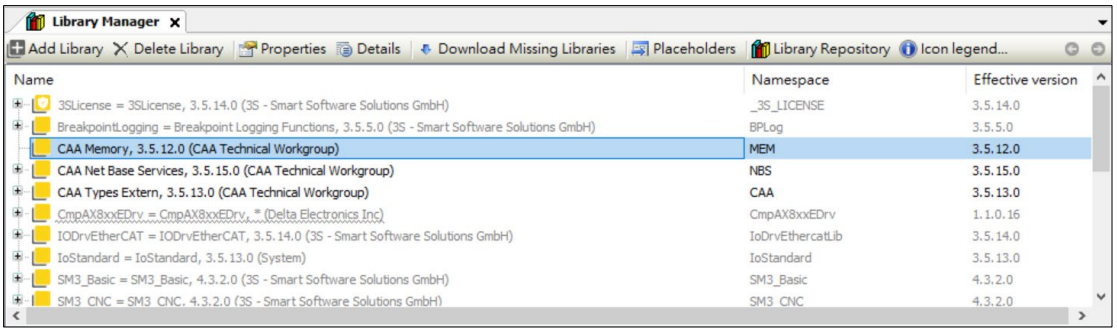
User can insert the library manager into the POU view or the device view. In this way, a project can have an application and a library manager in each POUs view, or a cross-application library manager. Regardless of the application, user can call the library module of the library manager in the POUs view. The library module of the library manager in the device view can only be called in the corresponding application code. In addition, due to different library locations, library placeholders behave differently during download.

The library manager can be inserted into the POUs view or the Devices view. In this way, a project can have one library manager per application and also one cross-application library manager in the POUs view. The library modules of the linked libraries in the POUs view can be called regardless of the application. The library modules of the linked libraries in the Devices view can be called in the respective application code only. Furthermore, placeholder libraries behave differently when downloading depending on their positions.

Libraries that are linked to a specific version in the project also have a placeholder for that version (placeholder library). User can define special placeholder resolutions. User can also use the placeholder resolution that is defined for a device in the device description or that is stored in the library repository for a library. The library manager notifies about the actual placeholder resolution and shows the version that is used when downloading an application (effective version).

If a library manager is linked cross-application in the POUs view, then user have global access to its contents. If placeholder libraries are linked, then only the placeholder resolutions in the device description or library repository are checked.

Normally, a library manager is linked in the Devices view. Then, only the application code below it calls the library modules. Moreover, the special placeholder resolutions are checked first for placeholder libraries. Only after that are the placeholder resolutions checked that are in the device description or that originate from the library repository.



Project	Description
Display in gray font	The library is automatically added to the project through a plug-in.
Display in black font	The library is automatically added to the project through the add library command.
Name	The name of the library definition
Namespaces	Namespace, used to uniquely access the contents of the library. The namespace is usually the same as the library name.
Effective version	A valid version of the library. This library version is used in the project.

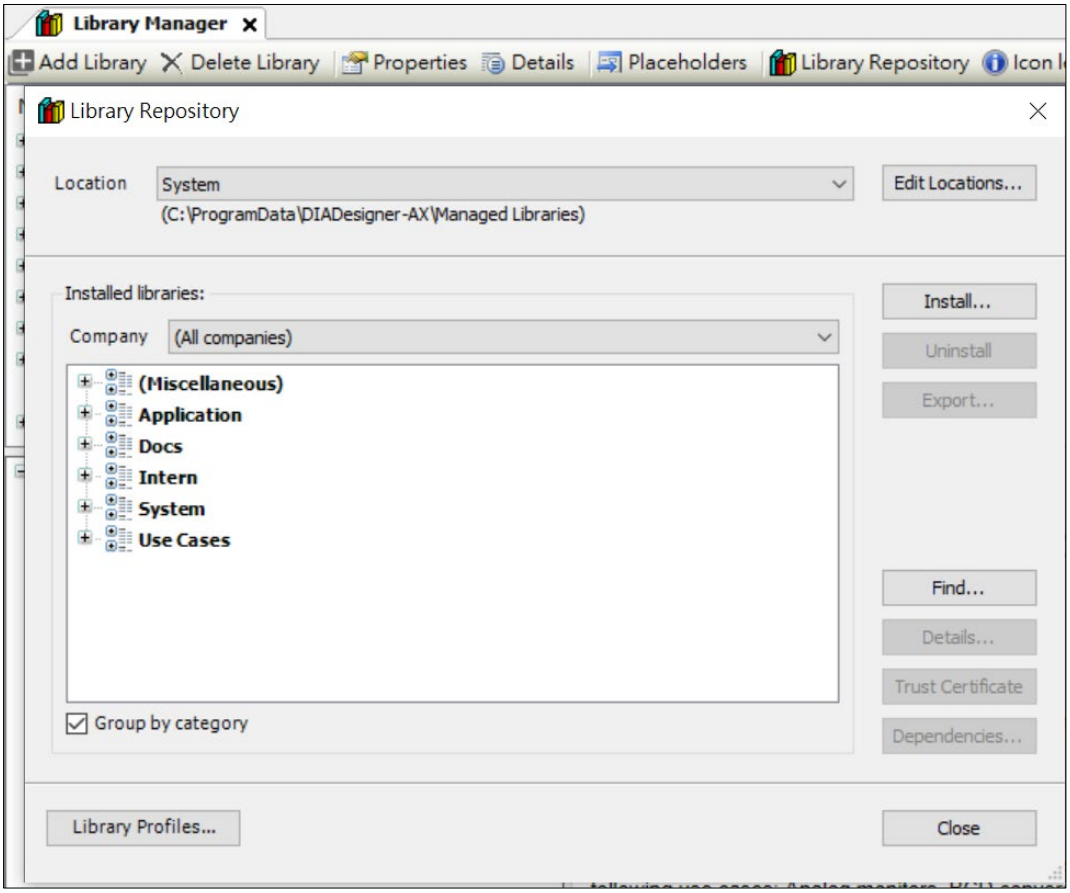
Directory function

Project	Description
Add library	This command opens the Add Library dialog box. In this dialog box, user can add libraries to the Library Manager and then integrate them into userr application
Delete library	Remove the currently selected library from the project.
Properties	This command opens the properties dialog box for the library selected in the library manager.
Details	According to controller and fieldbus classification

Project	Description
Download Missing libraries	CODESYS browses the missing library on the download server of the specified project. Then user can download or install the library
Placeholders	This command will open a placeholder dialog box. This dialog box displays information about the currently selected placeholder library, convert the library version to another version.
Library Repository	To install or uninstall the library and define the address of the library.
Icon legend	The meaning of icon legend information  Library signed with a trusted certificate  A library signed with a trusted certificate, but at least one unsigned library is referenced.  Library signed with private key and token  Unsigned libraries, or libraries signed with untrusted or expired certificates. For untrusted certificates, trusted certificate commands are provided in the context menu.  Defined as optional and currently unavailable library  Library being determined  There is currently no license library for a valid license  The library symbol of the library that cannot be loaded because the signature (encryption) cannot be verified.

8.2.1 Resource Library

Resource library defines which libraries have been installed on the local system and can be used in the application, and new libraries can be installed and used in the application through the interface.



Project	Description
Location	Display the directory on the local system where the library file is located. The libraries in this location are listed in the installed libraries area.
Edit Locations	Open the Edit Repository Location dialog box
Company	List of companies supported by the library.
Install	Open the Select the library to install dialog box. Possible screening procedures: • Compile CODESYS library file (* .compiled-library).

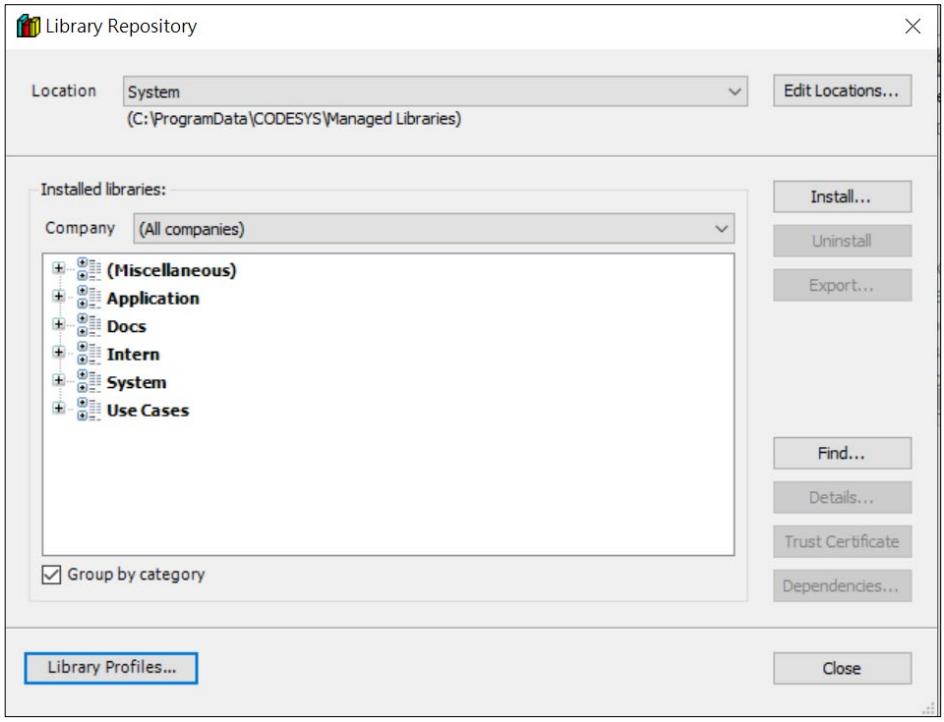
Project	Description
	• Compile CODESYS library file (* .compiled-library-v3) ab V3 SP15 • Library files (*.library) are used for library projects that have not yet been compiled • All files (*.*)
Uninstall	Uninstall selected library
Export	Opens the default dialog box for saving library projects to the local file system. The file type isLibrary files (*.library), Compiled library files (*.compiled-library), or Compiled library files (*.compiled-library-v3).
Find	Search libraries and function blocks. Open the Find Library dialog box. When entering a string in the input box, CODESYS will display the library found with the corresponding string.
Details	Open the detailed information dialog box, which contains detailed information about the selected library version in the library's project information. User can find the following information by clicking More in the details dialog: • Size: in bytes • Created: Date of creation • Change: Last change date • Last Visited: Date • Attributes • Attributes
Dependiencies	For the selected library, the dependency dialog box will open, showing dependencies on other libraries. Display the title, version and company referenced by each library. References that function through placeholders will be

Project	Description
	displayed according to the following syntax: #<placeholder name.
Group by category	• Startup: group by library category • Not activated: sorted alphabetically The category is defined by the external description file "*.libcat.xml".

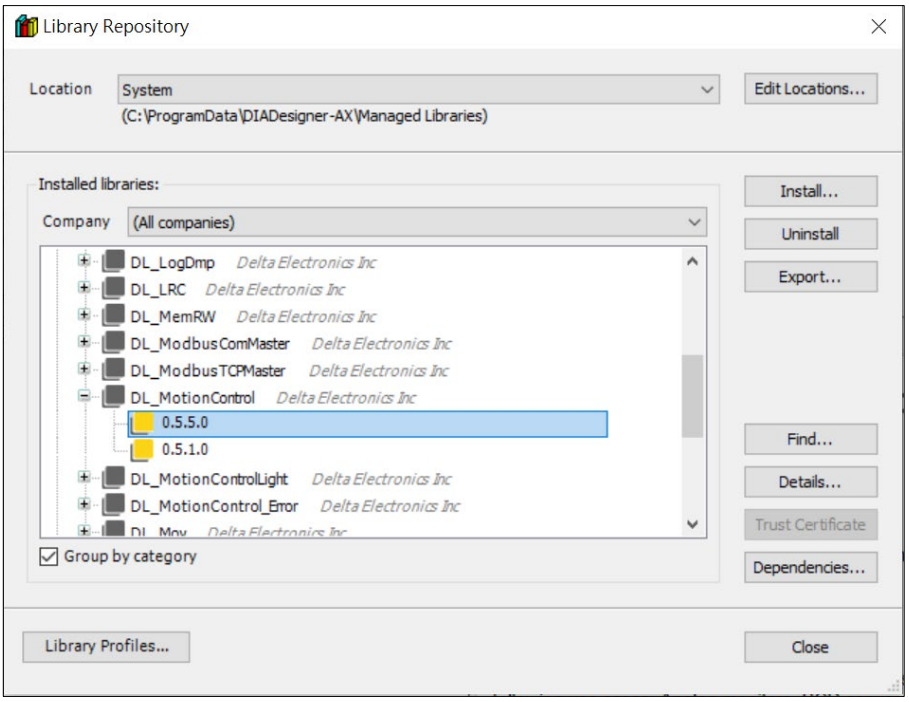
:

The following describes how to install the library in the Library Repository.

1. Select **Libaray Repository** and open it in the editor.
2. Select **Install**, select the library to be installed. User can choose the file format here.

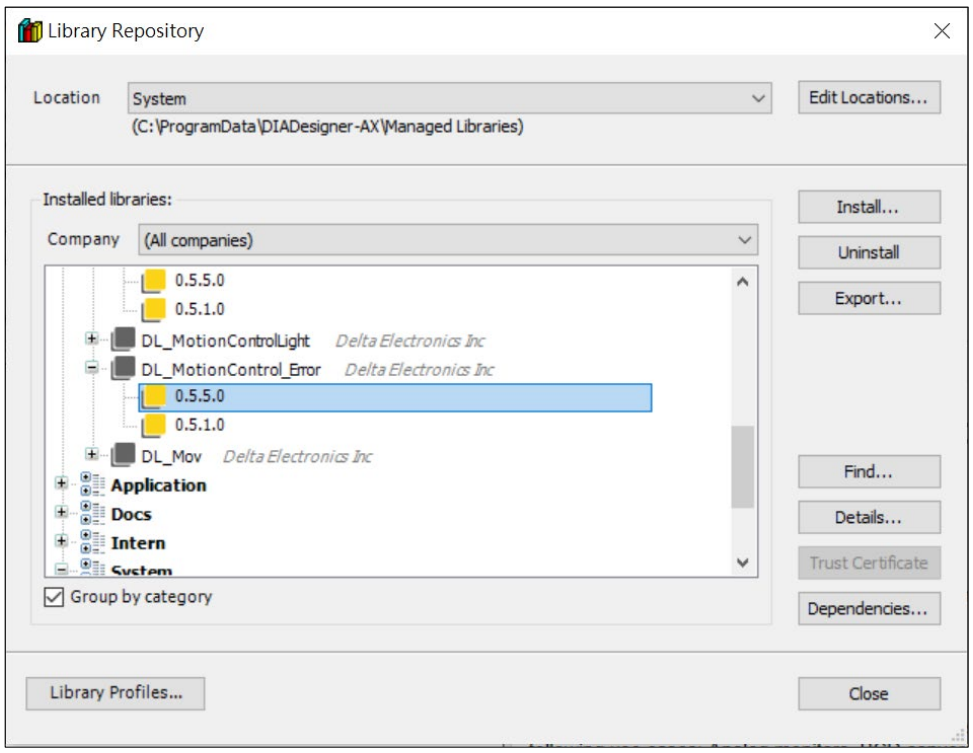


After the installation is complete, the new installation library will appear in the resource library.



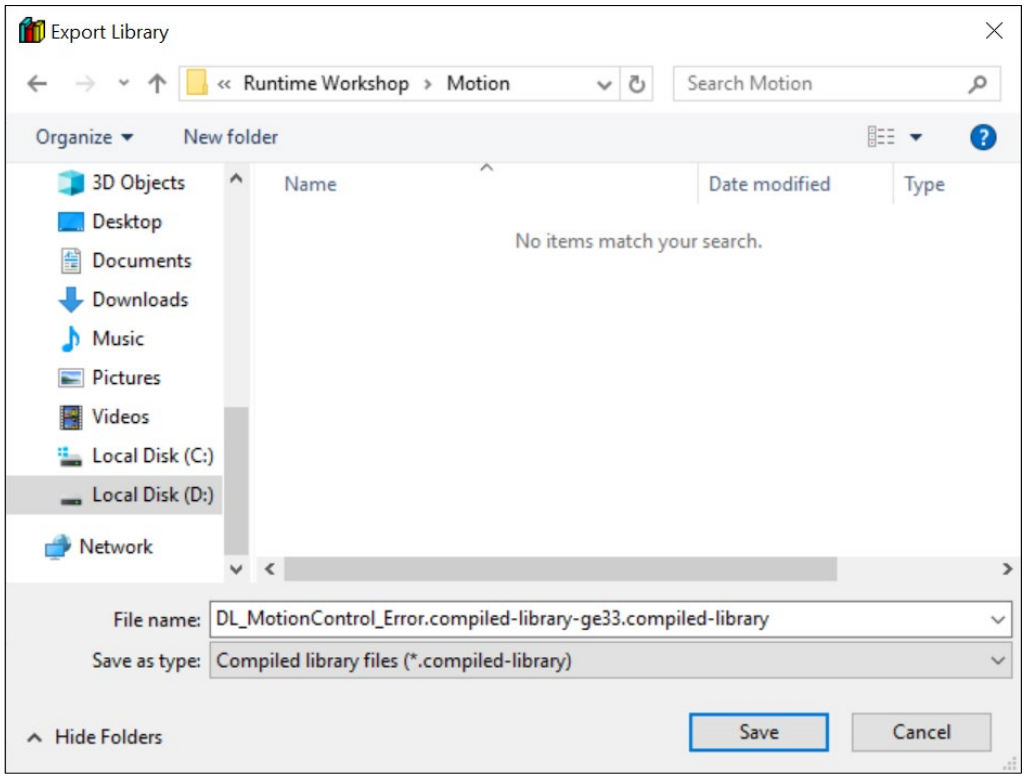
User can export the library from the resource library and save it as a file to the hard drive.

1. Open the application library manager in the project.
2. Select a library in the library manager and click **Export**. The Export Library dialog box will open.



3. If the selected library is not only linked to the project in the form of a compiled library, but also linked to the project in the original format, both file types are in the file type drop-down list. Otherwise, the filter will automatically display the available types: "*.library or *.compiled-library.

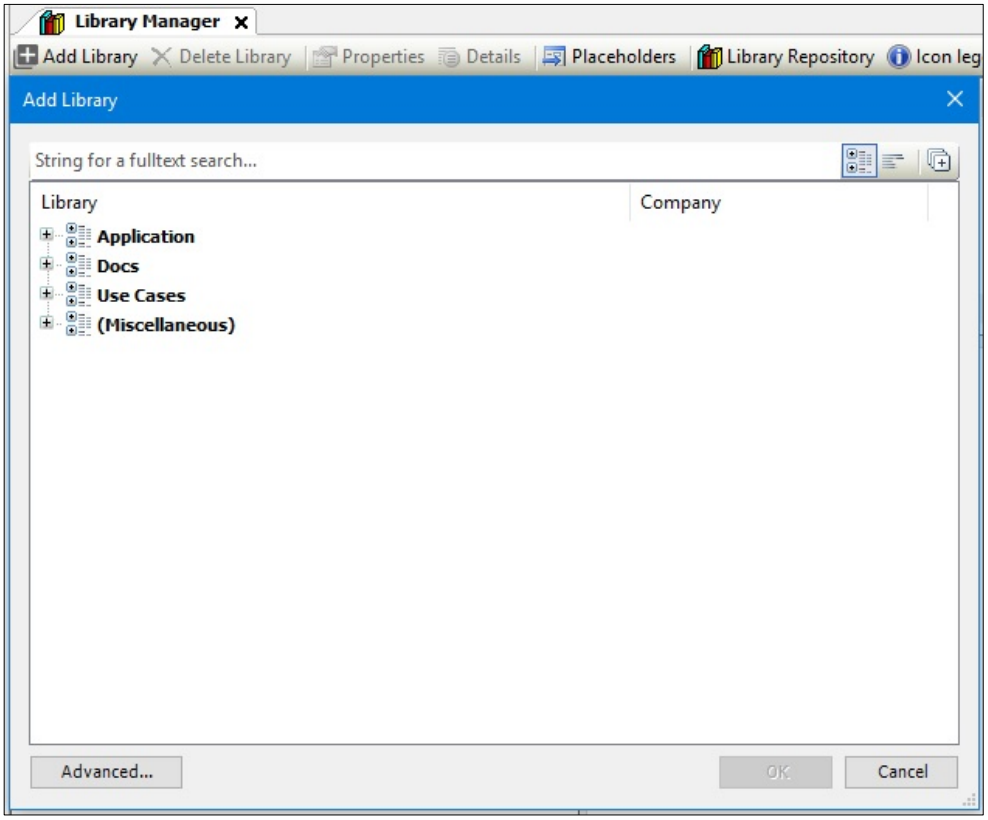
4. Select the file type and storage location, and then click **Save** to export.



8.2.2 Add a library

The following example shows how to add the Util library to the application. The user can add various libraries to use in the device in this way.

1. Double-click the **library manager** in the project tree, and the library manager will open in the editor
2. Select **Add Library** to open the dialog box for adding a library.

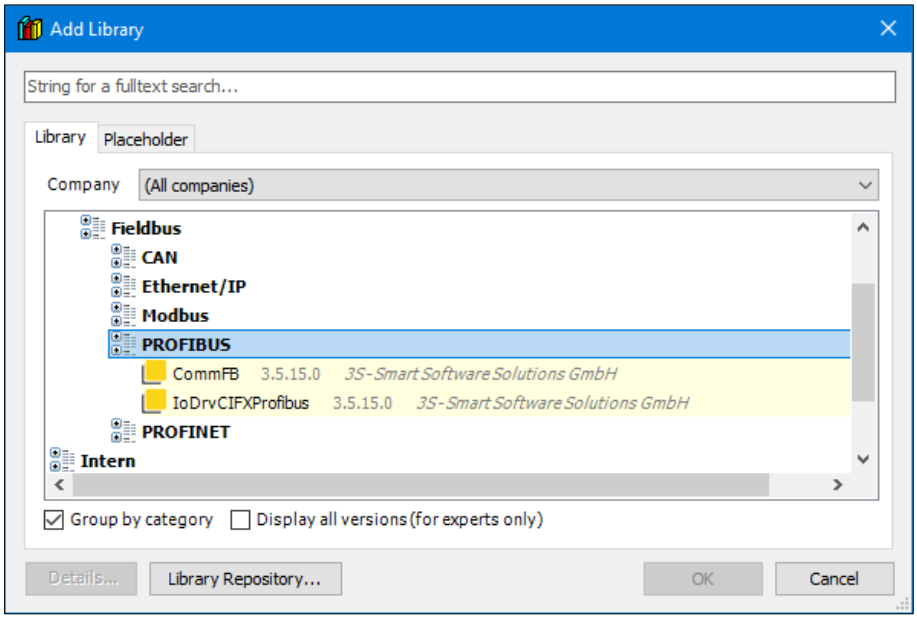


Project	Description
Search for	In the row above the library list, search for the library name or library module by entering a string
Function Library	Appropriate libraries installed in the resource library
The company	Creator of the library
Advanced	Open the advanced add library dialog

Project	Description
	***Note**: It is recommended to be familiar with the related functions of the library before using the advanced add library*
Category View 	Display library category
List View 	List libraries in alphabetical order
Show advanced library 	Show all available libraries

Click the **Advanced** button in the Add Library window, and there will be two tabs: Function Library and Placeholder.

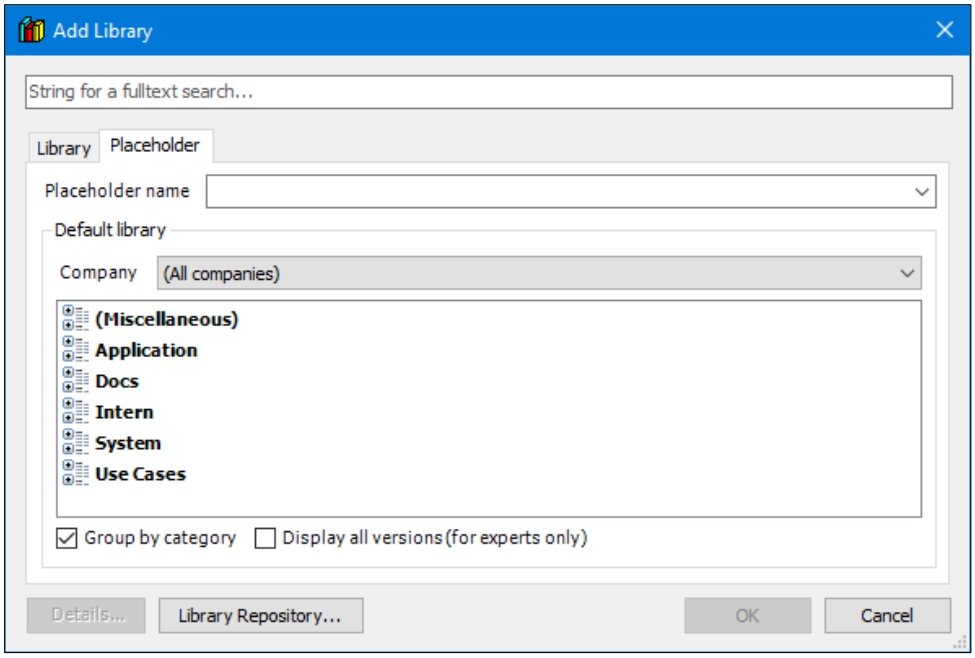
- Function library tab



Project	Description
Company	Filter the list by supplier
Group by category	Checked: Display the library in a tree structure Unchecked: Display libraries in alphabetical order in the plane structure

Project	Description
Show all versions	Checked: Display all versions of the library. "*" means the latest version available in the repository Unchecked: Only the latest version of the library is displayed. In this display, hold down the Shift key to select multiple libraries
Detailed	Details of library modules
Library	Inventory storage dialog box. More libraries can be installed to the local system.

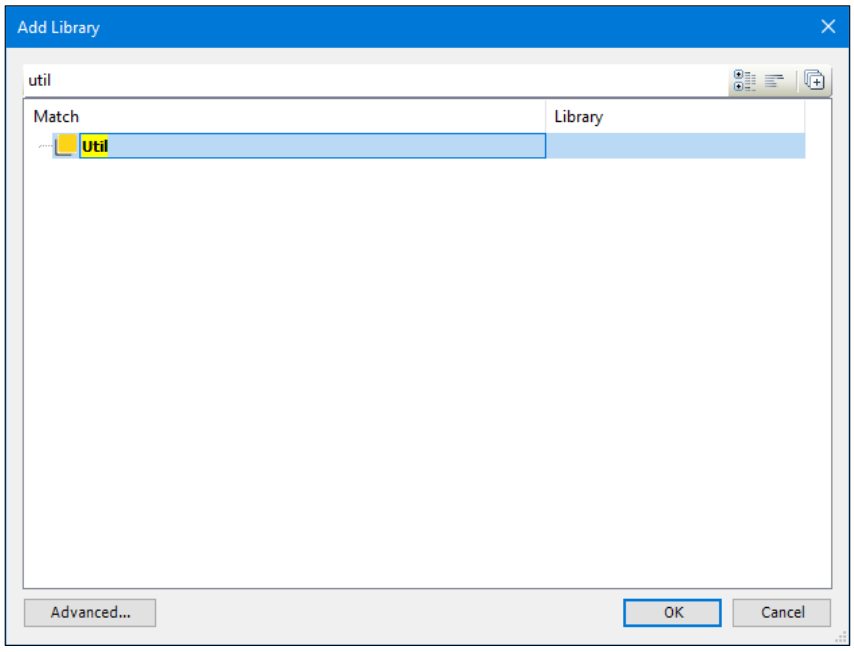
- Placeholder tab



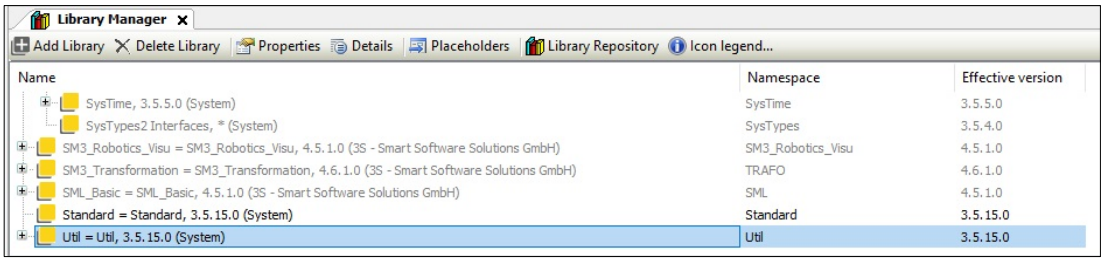
Project	Description
Placeholder name	Enter a valid placeholder name read from the currently accessible device description or select it from the drop-down menu
Company	Filter the list by supplier
Group by category	Checked: the library is classified and displayed in a tree structure

Project	Description
	Unchecked: Display libraries in alphabetical order in the plane structure
Show all versions	Checked: Display all versions of the library. "*" means the latest version available in the repository. Unchecked: Only the latest version of the library is displayed. In this display, hold down the **Shift** key to select multiple libraries
Detailed	Details of library modules
Library	When the defined device is not available, CODESYS will use this library. This will compile the current project correctly.

3. Enter the string Util in the search box in the Add Library window to browse the library, click the **Util** library, and then click **OK**.



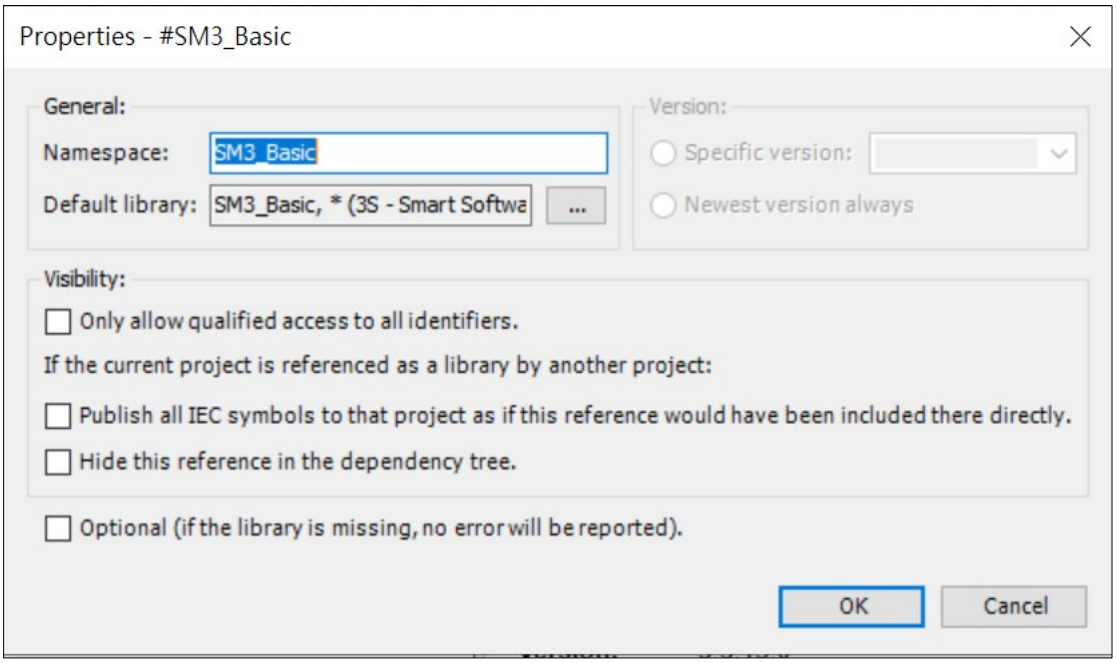
4. The Util library has been added to the library manager and can be used in the library manager.


Library Manager
Add Library
Delete Library
Properties
Details
Placeholders
Library Repository
Icon legend...
Name
Namespace
Effective version
SysTime, 3.5.5.0 (System)
SysTime
3.5.5.0
SysTypes2 Interfaces, * (System)
SysTypes
3.5.4.0
SM3_Robotics_Visu = SM3_Robotics_Visu, 4.5.1.0 (3S - Smart Software Solutions GmbH)
SM3_Robotics_Visu
4.5.1.0
SM3_Transformation = SM3_Transformation, 4.6.1.0 (3S - Smart Software Solutions GmbH)
TRAFO
4.6.1.0
SML_Basic = SML_Basic, 4.5.1.0 (3S - Smart Software Solutions GmbH)
SML
4.5.1.0
Standard = Standard, 3.5.15.0 (System)
Standard
3.5.15.0
Util = Util, 3.5.15.0 (System)
Util
3.5.15.0

8.2.3 Attributes

After selecting any library in the library manager, the property button will be enabled.

*__Note__: It is recommended for people with professional knowledge related to library development to use.



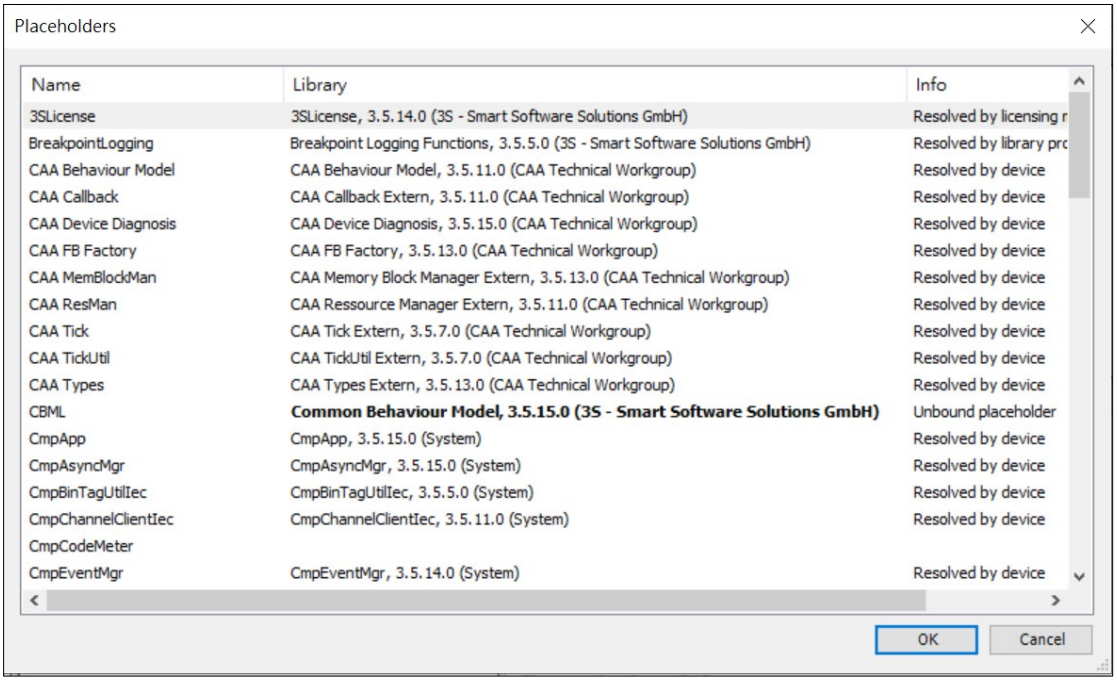
Project	Description
General	
Namespace	Display the current namespace. By default, this is the same library name, unless user explicitly define different standard namespaces when creating library information in the project. User can change the namespace of the local project in the **Properties** dialog box.
Default library	If user have just selected the library placeholder in the library manager, when no other resolution can be defined, then this field contains the name of the library to replace the placeholder
Version	

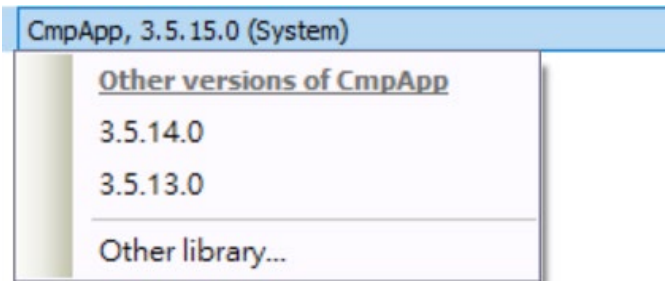
Project	Description
Specific version	To use this version is to select the desired version from the list.
Newest version always	Always use the latest version of the library found in the library's database. If a new version of the library is available, the library module can actually be changed.
Visibility	
Only allow qualified access to all identifiers	Library modules (and variables) can only be called in the project using the pre-name space path.
If the project is referenced as a library by another project	It makes sense to change the following settings only when a library is created using the project and therefore another library project is opened. In this way, the selected library will be referenced in the new library.
Publish all IEC symbols to that project as if this reference would have been included there directly	As a container library, the selected library makes the contents of the referenced library visible at the top level (a higher level in the project). **Requirements**: Use the library project to create a container project. The container library does not implement its own modules, but specifically references other libraries. It is bundled with libraries. User can use container libraries wisely to bundle multiple libraries in a project (in the reference). This option must be enabled for each library reference.
Hide this reference in the dependency tree	The selected library is not displayed as a library reference in the library manager (appears later in the project). The library is a hidden reference. **Warning**: If there are compilation errors caused by hidden library errors, it may be difficult to detect these errors.

Project	Description
Optional	The selected library is considered an optional library. When downloading a project that references the library, even if the library is not available in the library repository, no errors will be reported.

8.2.4 Placeholder

The placeholder describes the configuration and the target device of the library, or has not been defined (*free Placeholder*). The user can transfer each placeholder version to another version of the library.



Project	Description
Name	Placeholder name
Library	The library version currently valid for the project. Double-click the item to display a selection list with available library versions, user can select the version that needs to be used in the project. 
Info	Placeholder device type: • Device analysis: Solve by device description

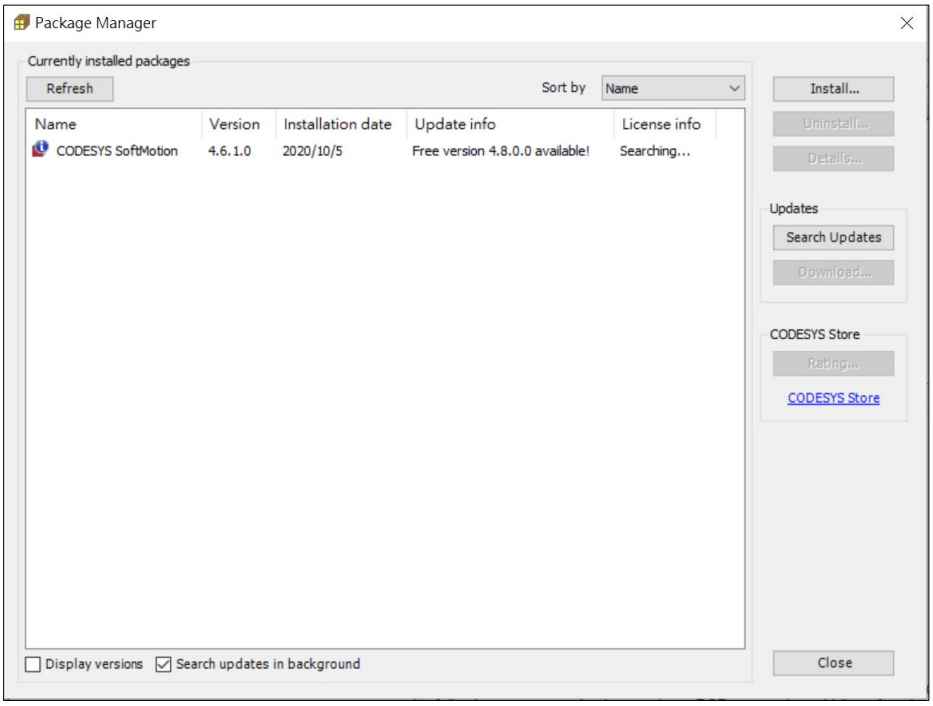
Project	Description
	• Library configuration file analysis: resolved by the database configuration file. • Solved by XXX configuration file: Solved by <specific library>

The placeholder function can mainly convert the currently used library version to the required version. The main functions are as follows:

1. If a specific version is assigned to the placeholder library through the placeholder dialog, the project will apply this version.
2. If a specific version is not defined, it will check whether the version is specified in the device description of the app.
3. The library configuration file will be checked for version definitions.
4. Shown in the library manager below the active version.

8.3 Package Manager

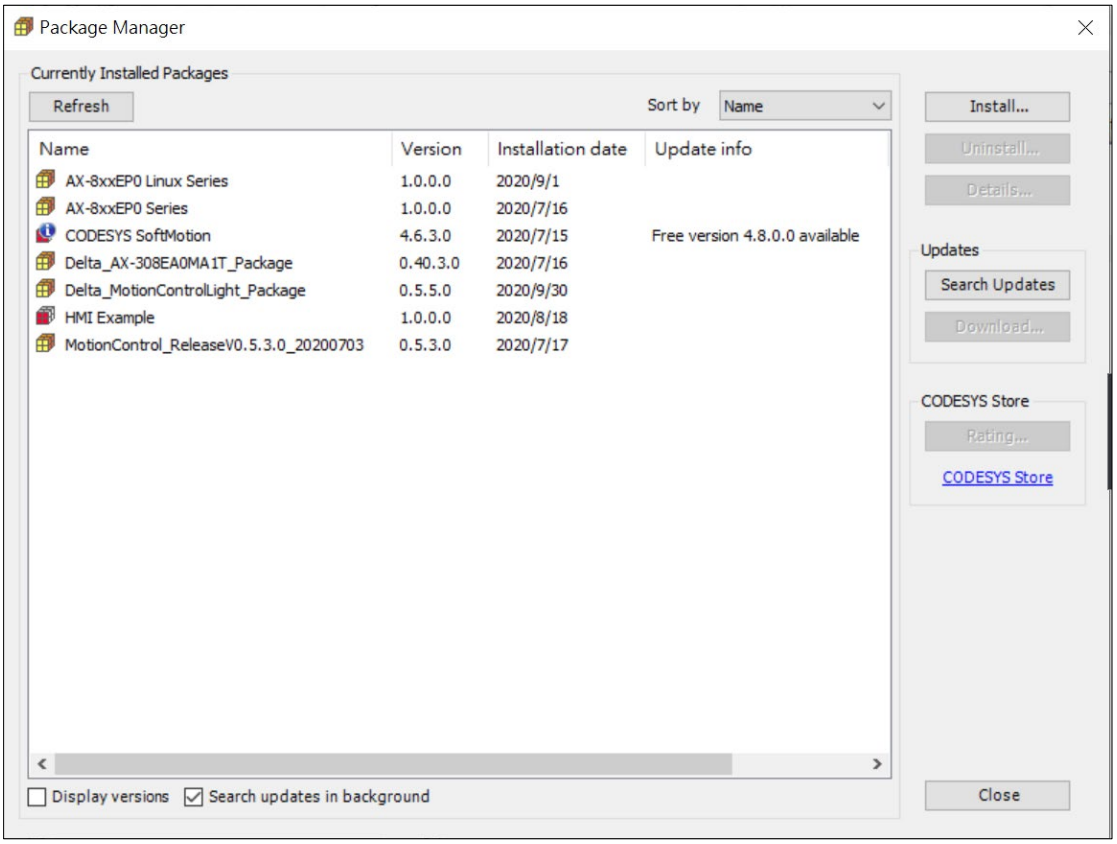
DIADesigner-AX provides a package manager whose functionality extends the standard installation through DIADesigner-AX additional functions and configuration settings. The package manager is located in **Tools** > **Package Management**. The concept is similar to the typical Windows installation mechanism. The package is a ZIP file with a ***.package** file extension.



The possible package components include the following items that can be installed:

- Plug-in
- Library
- Description of the device
- Supplier’s description
- Configuration
- Configuration information
- Configuration changes
- File
- Function table, toolbar, keyboard shortcuts and view configuration extension
- Description module

- Complete menu configuration of special profile
- Complete toolbar configuration for special profile
- Complete keyboard configuration for special profile
- Options
- Library configuration



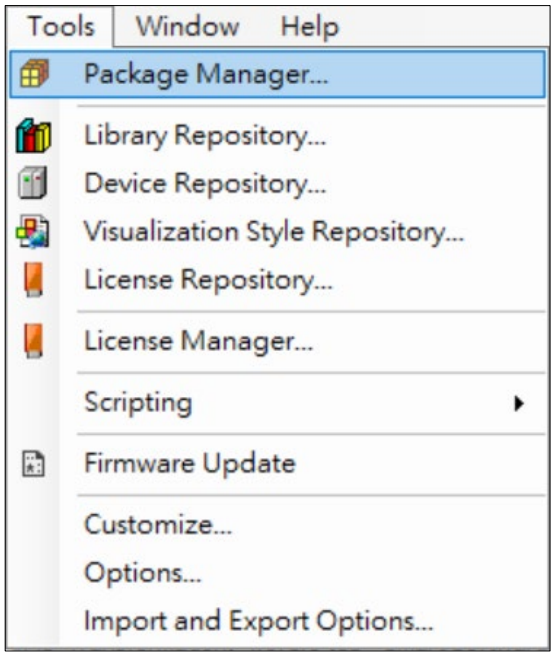
Project	Description
Currently installed Packages	The list of installed packages includes: name, version, installed data, update information, authorization information. If the package comes from the CODESYS Store, it will be marked with a red package symbol instead of a yellow symbol.
Refresh	Update list
Install	Browse the packages in the file system and select the package to install. The standard file types are *.package.

Project	Description
	After selecting the package, the installation steps are shown as follows: • Installation-License Agreement Show package checksum in installation protocol dialog. Only displayed if the package contains a license agreement. • Choose installation type o Full installation: install all components o Typical installation: install the standard set from the components defined in the package • Installation-target system version: select which existing target system version to update through package installation. User must select at least one version profile
Uninstall	Uninstall selected package • If the display version is not started, CODESYS uninstalls all versions of the selected package • If the display version is activated and user select the highest level package node, CODESYS will uninstall all versions of the selected package If the display version is activated and user select a package version, CODESYS will uninstall that version exactly.
Details	Use the following options to open the details dialog of the selected package: • Package details o Name: The name of the package o Version o Checksum: packaged SHA-1 CRC

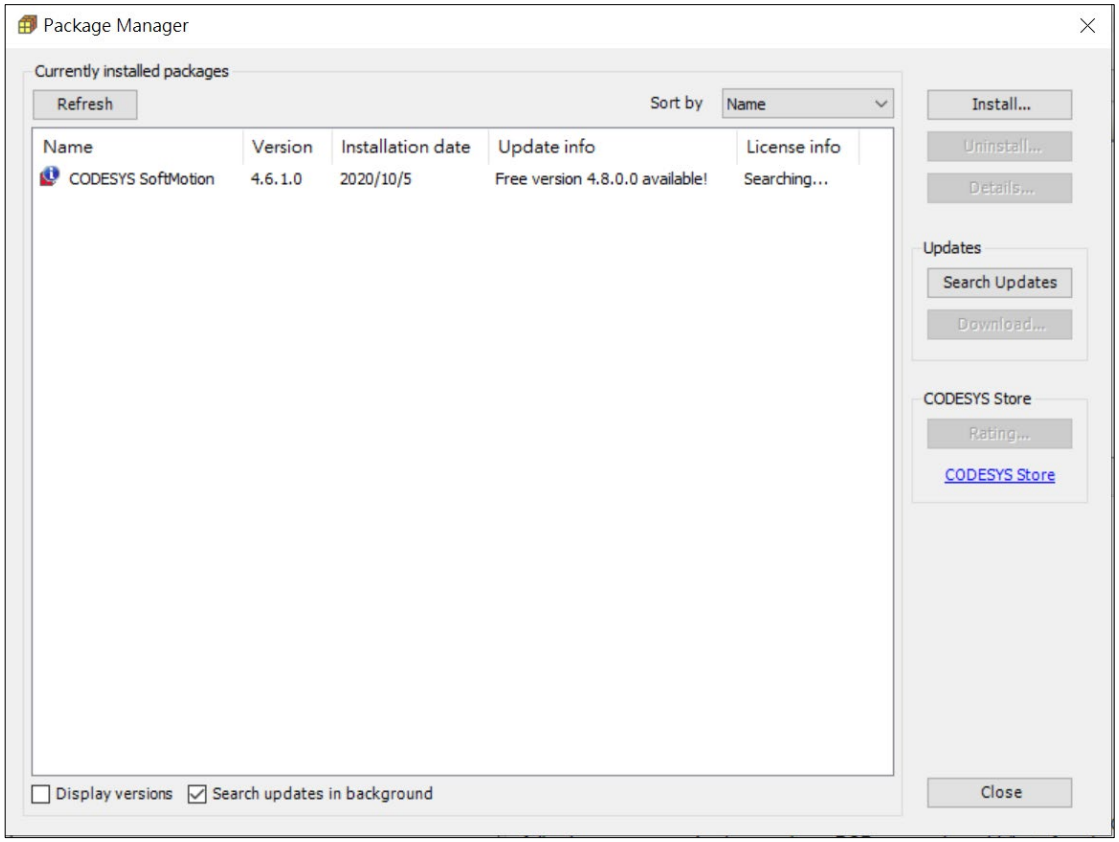
Project	Description
	○ Supplier ○ Copyright ○ Description ○ Installation date • License Agreement
Updates	
Search Updates	Every time the programming system is started, CODESYS automatically searches for updates every hour.
Download	Install the update package with the help of the dialog download package. To do this, click the download and install button in the dialog download package.
CODESYS Store	Link to the homepage of the CODESYS store
Display versions	Show all versions of installed packages.
Search updates in background	Search for updates of the selected package on the system and in the CODESYS Store. CODESYS displays the list of found update packages in the update information bar.

The steps to install a package are as follows:

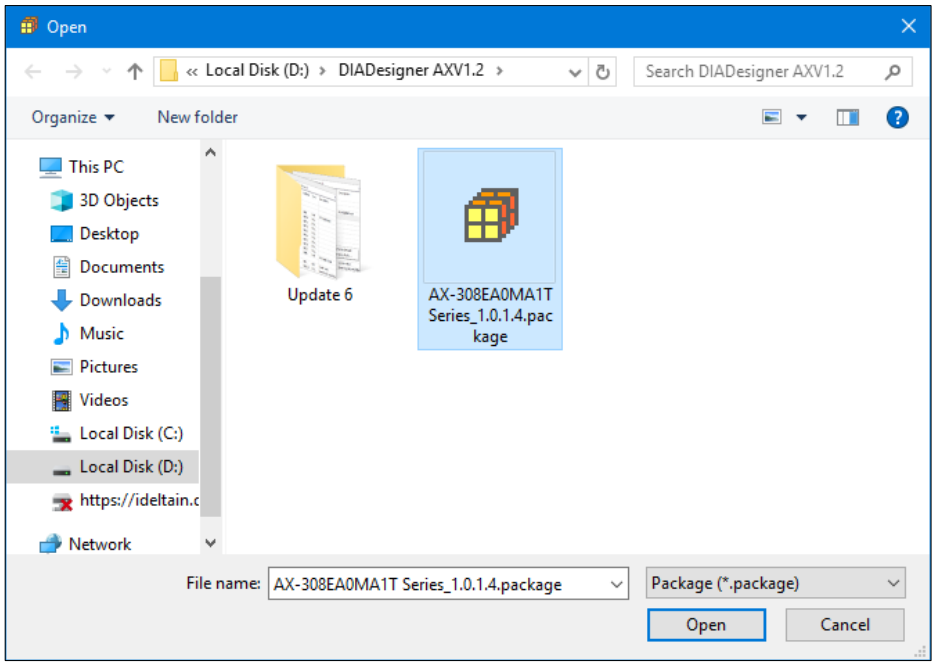
1. Open **Tools** > **Package Manager**.

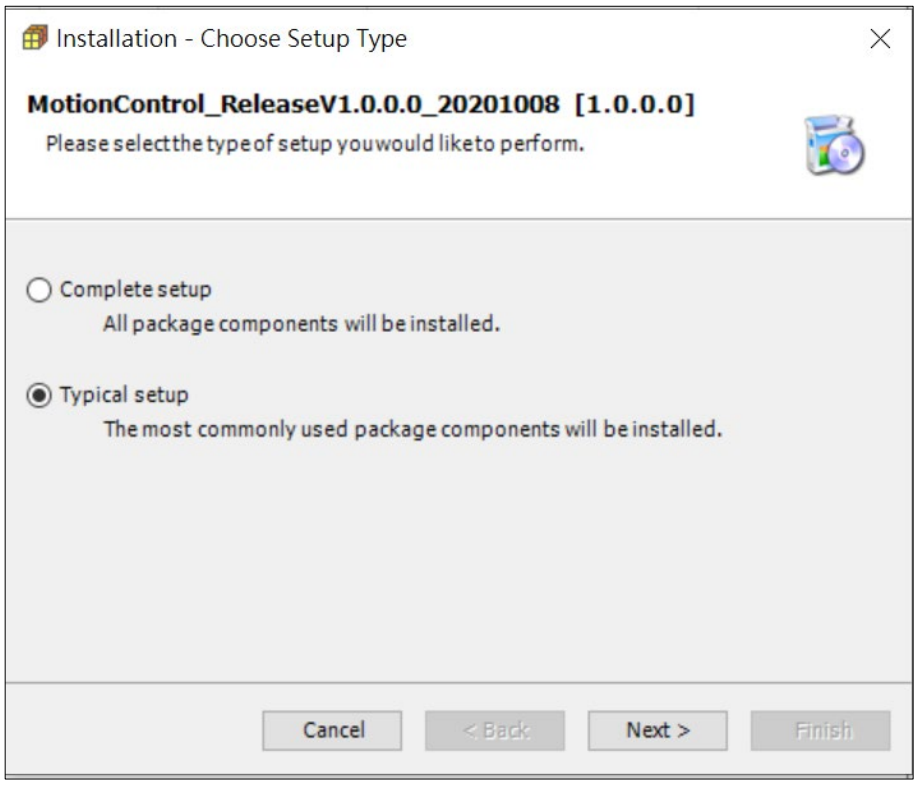


2. Click **Install**, select the package user want to install and open it to install.



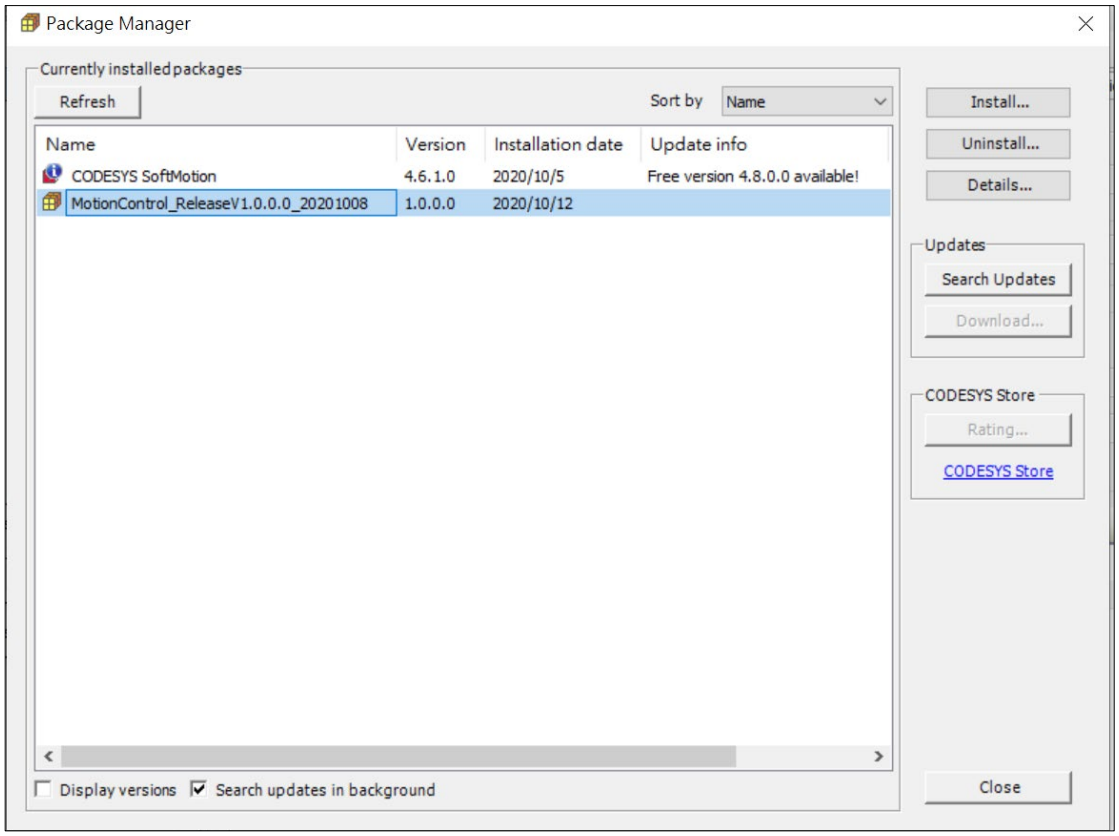
3. In the open dialog box, select a package from the file directory, and then click **Open**, the installation step will start, and the installation is complete, select **Finish**.





Uninstall a package:

1. After selecting the package to be uninstalled, click **Uninstall**.
2. User will be guided through the process of uninstalling the package.


Package Manager
Currently installed packages
Refresh
Sort by
Name
Name
Version
Installation date
Update info
CODESYS SoftMotion
4.6.1.0
2020/10/5
Free version 4.8.0.0 available!
MotionControl_ReleaseV1.0.0.0_20201008
1.0.0.0
2020/10/12
Install...
Uninstall...
Details...
Updates
Search Updates
Download...
CODESYS Store
Rating...
CODESYS Store
Display versions
Search updates in background
Close

8.4 Startup Command

How do I open **Startup Command**?

- On the **Devices** pane, right-click the controller and then select **Startup Command**.
- Select **Tools** > **Startup Command**.

Startup Command supports configuring and generating different commands to a .PLCCmd.ini.scmd file and saving it to a portable device. Once the portable device is attached to the controller, the command will be run after a restart of the controller.

Supported device:

- AX-3
- AX-8 series controllers (Linux operating system only)

Note: AX-8 series and AX-332 controllers only suggest using SD memory card as the portable device. For the AX-332 controller, you need to open the 

cover to find the hidden Micro SD slot. For more information about inserting a memory card, please refer to *AX-3 Series Operation Manual.*

Use the following table to configure the **Startup Command** dialog.

Setting	Description
	Click to open the **Command** dialog to add or insert commands.
	Click to open the **Template** dialog to select the command template.
	Click to export the command configuration file.
	Click to import the command configuration file.
Append Command	Select to add commands.
Insert Command	Select to insert commands before the selected command.

Setting	Description
Restore Application from Current Controller	Select to restore the application file from the current application to the controller.
Restore Application from Portable Device	Select to restore the application file from the portable device to the controller.
Restore Source from Current Project	Select to restore the source file from the current project to the controller.
Restore Source from Portable Device	Select to restore the source file from the portable device to the controller.
Reset to Factory Settings	Select to reset the controller to the factory settings.
Backup Application	Select to back up the application file from the controller to the portable device.
Backup Source	Select to back up the source file from the controller to the portable device.
Firmware Update	Update the controller firmware from the portable device.
Keep	Select to always run this command.
Execute Time	Clear **Keep** and enter **Execution Times** (1–65536). After running the number of times you defined, the command will be deleted.
Exit	Select to stop and exit after running this command.

8.4.1 Add and generate commands

Use backup and restore commands to save and recover the file in case of an unexpected data loss, or to transfer files from one controller to another. Use the firmware update command to update the controller firmware via the file in the portable device. Use the reset to factory settings command to delete all the data and settings in the controller.

Prerequisites

- The portable device is attached.

To add backup or restore commands

1. Open the **Startup Command** dialog.
2. Click  .
3. In the **Command** dialog, select **Append Command**, and then select the backup and restore commands.
4. Click **OK**.
5. In the **Startup Command** dialog, set up the execution of the command via **Keep**, **Execution Times**, or **Exit**.
6. (optional) Click **?** to precompile commands to check the required space, and verify errors if any.
7. Click **Generate**. The command file is generated in the portable device.
   Note: Only the FAT32 format portable device is supported. A message will ask you to format your portable device to FAT32 if it is not. You can follow the steps to format it.
8. Attach the portable device to the target controller. The command will be run after a restart of the controller.
   Note: Make sure the restore file exists before running the command to restore the application or source file from the portable device.

When you select **Restore Source from Current Project**, you can also add the additional files.

To add additional files

1. Open the **Startup Command** dialog.
2. Click  .
3. In the **Command** dialog, select **Append Command** > **Restore Source from Current Project**, and then click **OK**.
4. In the **Startup Command** dialog, in the **Settings** column, click **Additional Files**.
5. In the **Additional Files** dialog, select the files and then click **OK**. These settings are synchronized with **Project** > **Project Settings** > **Source Download** > **Additional Files**, and can be saved with the project.

To add the Update Firmware command

1. Open the **Startup Command** dialog.
2. Click  .
3. In the **Command** dialog, select **Append Command** > **Firmware Update**, and click **OK**.
4. In the **Startup Command** dialog, in the **Settings** column, click , select the firmware package file, and then click **Open**.
5. In the **Startup Command** dialog, set up the execution of the command via **Keep**, **Execution Times**, or **Exit**.
6. (optional) Click **?** to precompile commands to check the required space, and verify errors if any.
7. Click **Generate**. The command file is generated in the portable device.
   Note: only FAT32 format portable device is supported. A message will ask you to format your portable device to FAT32 if it is not. You can follow the steps to format it.
8. Attach the portable device to the target controller. The command will be run after a restart of the controller.
9. After the firmware update completed, restart the controller again.

To add the Reset to Factory Settings command

1. Open the **Startup Command** dialog.
2. Click  .
3. In the **Command** dialog, select **Append Command** > **Reset to Factory Setting**, and click **OK**.
   All the data and settings in the controller will be deleted after running this command.
4. In the **Startup Command** dialog, set up the execution of the command via **Keep**, **Execution Times**, or **Exit**.
5. (optional) Click **?** to precompile commands to check the required space, and verify errors if any.
6. Click **Generate**. The command file is generated in the portable device.
   Note: only FAT32 format portable device is supported. A message will ask you to format your portable device to FAT32 if it is not. You can follow the steps to format it.
7. Attach the portable device to the target controller. The command will be run after a restart of the controller.
8. After running the command, restart the controller again.

8.4.2 Add and generate template commands

The template command provides you with two command combinations that are convenient for widely used situations. Select one of the following options if you want to apply the application or source file to another controller.

- Backup & Restore Application
- Backup & Restore Source

Prerequisites

- The portable device is attached.

To add and generate template commands

1. Open the **Startup Command** dialog.
2. Click  .
3. Select one of the templates, and then click **OK**.
4. Click **Generate**. The command file is generated in the portable device.
   Note: only FAT32 format portable device is supported. A message will ask you to format your portable device to FAT32 if it is not. You can follow the steps to format it.
5. Attach the portable device to the controller to back up the file.
6. Attach the portable device to the other controller to restore the file.

8.4.3 Import and export commands

You can export and import the command configuration to an XML file.

To export commands

1. Open the **Startup Command** dialog.
2. Add the command.
3. Click  .
4. Enter **File Name**, and then select a path to save the command configuration file.
5. Click **Save**.

To import commands

1. Open the **Startup Command** dialog.
2. Click  , and select the command configuration file to be imported.
3. Click **Open**. The commands will be listed in the **Startup Command** dialog.

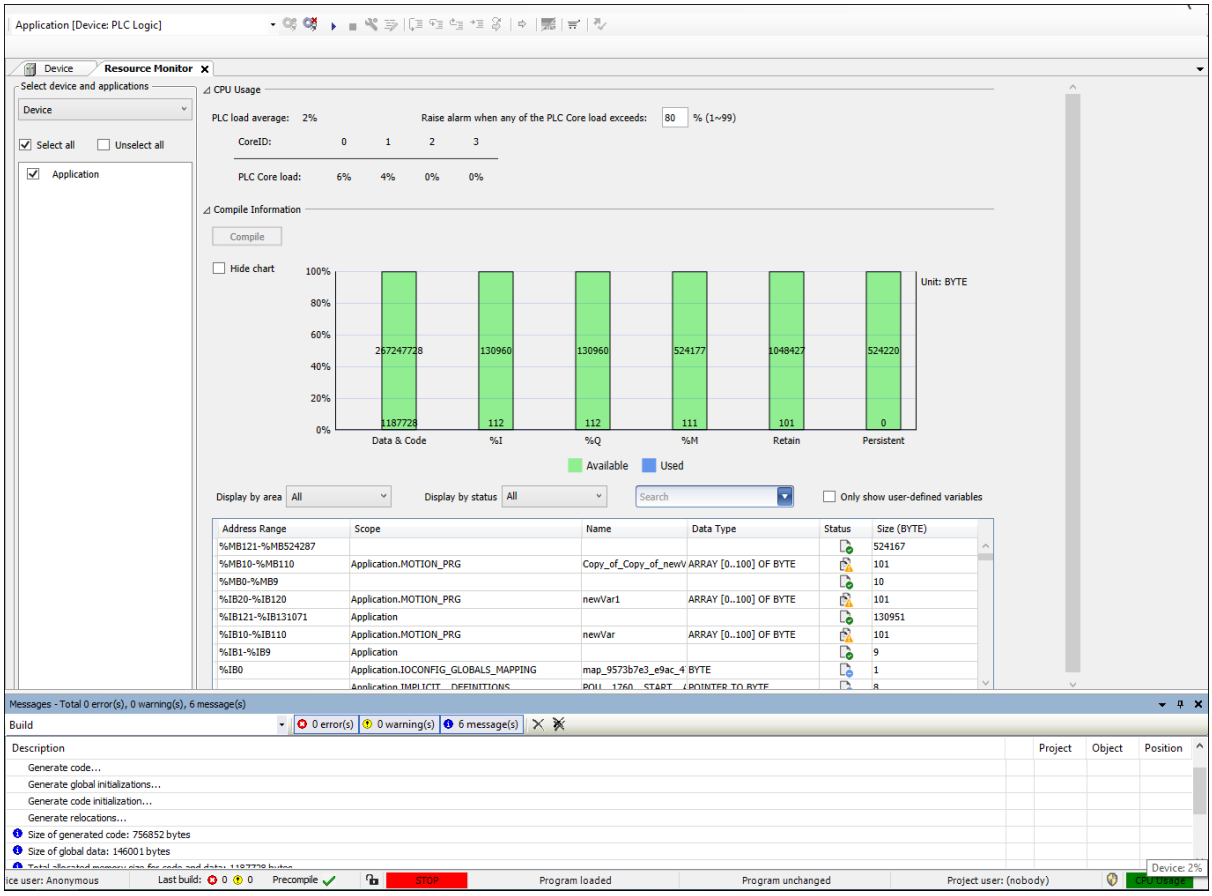
8.5 Resource Monitor

How do I open Resource Monitor?

- Select **Tools** > **Resource Monitor**.

Resource Monitor is only available when there is a project with AX series controllers.

Use **Resource Monitor** to monitor the controller CPU usage in real time and the variable usage, such as the size of variables assigned to different storage areas.



8.5.1 Monitor variable usage

Use the following steps to view the variable usage.

To monitor variable usage

1. On the **Select device and application(s)** pane, select the device and applications that you want to monitor.
   If you edit devices or applications on the **Devices** pane, it will be updated in real-time on the **Select device and application(s)** pane.
2. On the **Compile Information** pane, click **Compile**.
   All user-defined variables and system variables are monitored.

The following table lists the variable compilation information.

Item	Description
Address range	Show the address range of I%, M%, or Q%.
Scope	Show the location of the variable. The scope format: Application name + Signature name.
Name	Show the variable name.
Data type	Show the variable data type.
Status	Show the usage condition of the address. • : Used: There are variables assigned to the address. • : Available: No variables assigned to the address within this range. • : Conflict: Different variables are assigned to the same address.
Size	Show the variable size or the size of the address section.

On the **Compile Information** pane, you can:

- Double-click the variable in the table to see where it is being used.
- Filter the variables by area or status.

- Search the variable by the variable name, address range, or scope.
- Select **Only show user defined variable** to filter out the system variables.
- Select **Hide Chart** to hide the bar chart to expand the variable table.

8.5.2 Monitor the online controller CPU usage

Use the following steps to monitor the controller CPU usage when online.

To monitor the online controller CPU usage

1. On the **Devices** pane, double-click the device.
2. On the **Device** editor, select **Communication Settings** > **Scan Network**, and then select the device.
3. Click **System Settings** > **Read from PLC**.
4. Log in to the controller.
   The **CPU Usage** at the lower-right corner will show. You can also double-click it to launch **Resource Monitor**.
   Resource Monitor will monitor the CPU usage of the corresponding online controller.

The following table lists the items monitored online.

Item	Description
PLC load average	Show the average PLC load.
Raise alarm when any of the PLC Core load exceeds %	The **CPU Usage** at the lower-right corner will turn red when any of the **PLC Core load** exceeds the specified value.
CoreID	Show the Core ID.
PLC Core load	Show the PLC Core load in real time.

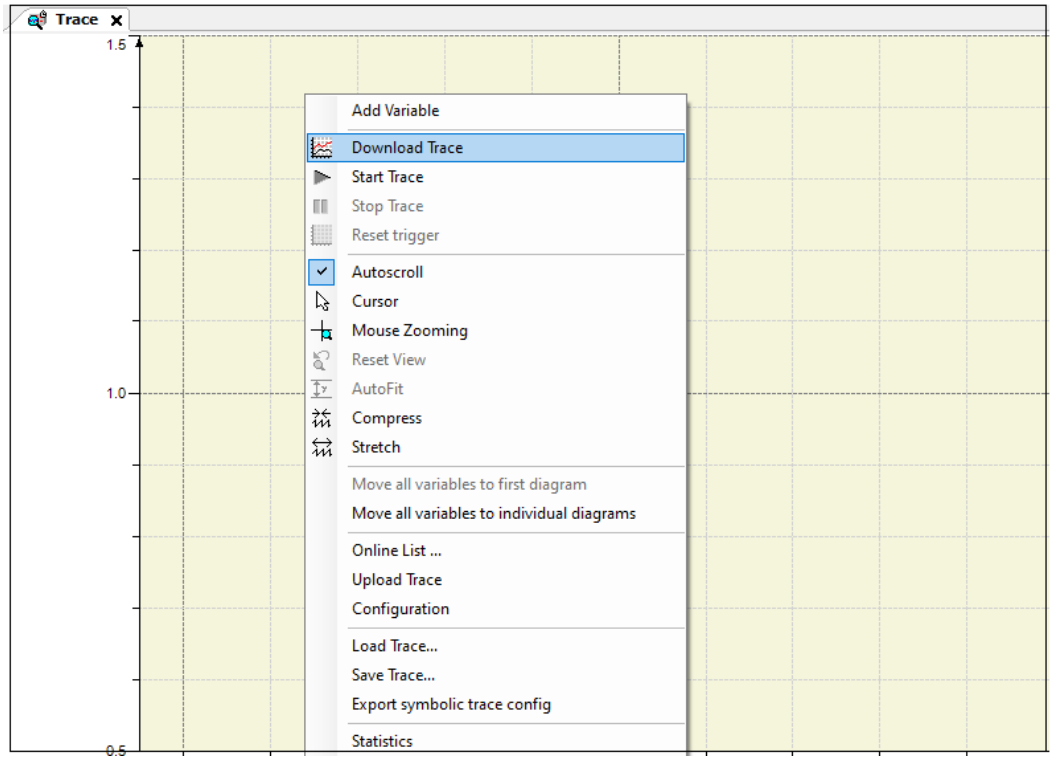
Note: If you log in under simulation mode, CPU usage monitoring will not be available.

8.6 Trace variables

Trace is used to monitor variables and show the variables' value change in amplitude charts.

To trace variables

1. On the **Devices** pane, right-click **Application** and select **Add Object** > **Trace**.
2. In the **Add Trace** dialog, enter a name, select a task, and click **Add**.
3. On the trace tab, click **Add Variable**.
4. In the **Trace Configuration** dialog, click  next to **Variable**.
5. In the **Input Assistant** dialog, select a trace variable or traceable parameter and click **OK**.
6. Log in to the controller.
7. Right-click the Trace field and select **Download Trace** to start monitoring.



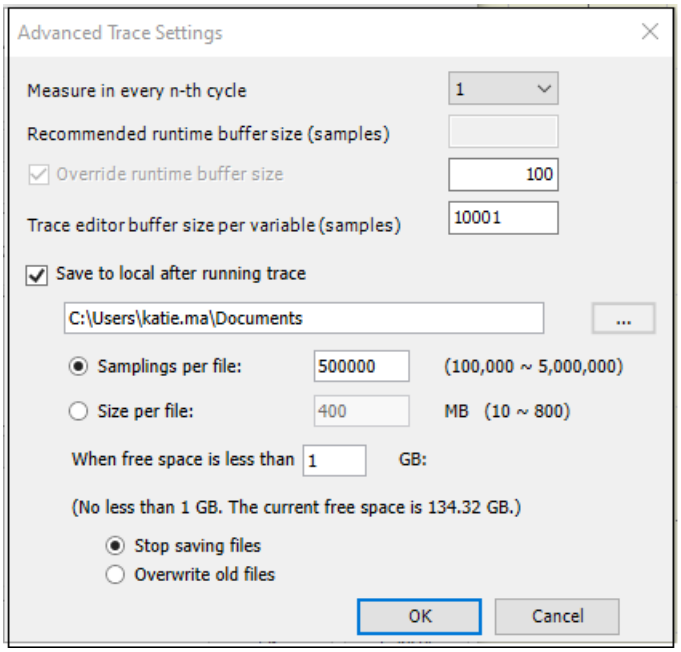
8. (optional) You can select **Move all variables to individual diagrams** to show the trace separately for multiple variables.

8.6.1 Automatically save traces

DIADesigner-AX supports automatically saving the trace data to a local .csv file after the trace starts. Saving samplings will only stop when the free disc space reaches the lower limit you set up.

To automatically save traces

1. On the trace tab, click **Configuration**.
2. In the **Trace Configuration** dialog, click **Advanced**.
3. In the **Advanced Trace Settings** dialog, select **Save to local after running trace** to turn on more options.
4. Select a local path to save the file.
5. Do one of the following:
   - Select **Samplings per file**, and then enter the number of samplings saved to one .csv file.
   - Select **Size per file**, and then enter the size limit of the .csv file.
6. Enter the minimum disc space and select an option when reaching this limit.



8.7 Software Diagnosis

Software Diagnosis is a tool to diagnose the running environment for DIADesigner-AX and collect the system and software information. The tool will show all the compatible and incompatible items.

The tool is installed automatically when installing DIADesigner-AX.

The tool supports diagnosing two versions of DIADesigner-AX (version 1.3.1 and another) installed on the computer at the same time.

To launch Software Diagnosis

1. Go to C:\Program Files\Delta Industrial Automation\DIAStudio\Diagnosis Tool.
2. Double-click **SoftwareDiagnosis**.

To diagnose the environment

1. Click **Collect** to start the diagnosis.
   If you want to stop diagnosing, click **Cancel**.
2. Look for the red items that indicate incompatible issues.
3. Select the red item.
4. On the **Prompt Information** pane, check the solution.

You can click **Save as** to export the diagnosis information as an XML file or a zip file.

You can click **Open File** to open the generated XML file.

You can search for related information in the search box.

Select the **Only show incompatible items** checkbox to filter and show the items that are not compatible with the operating system.

Chapter 9: Password management and data protection mechanism

9.1 Overview

For projects developed by users, Designer-AX provides a protection mechanism to ensure that the programs developed by users can be fully protected.

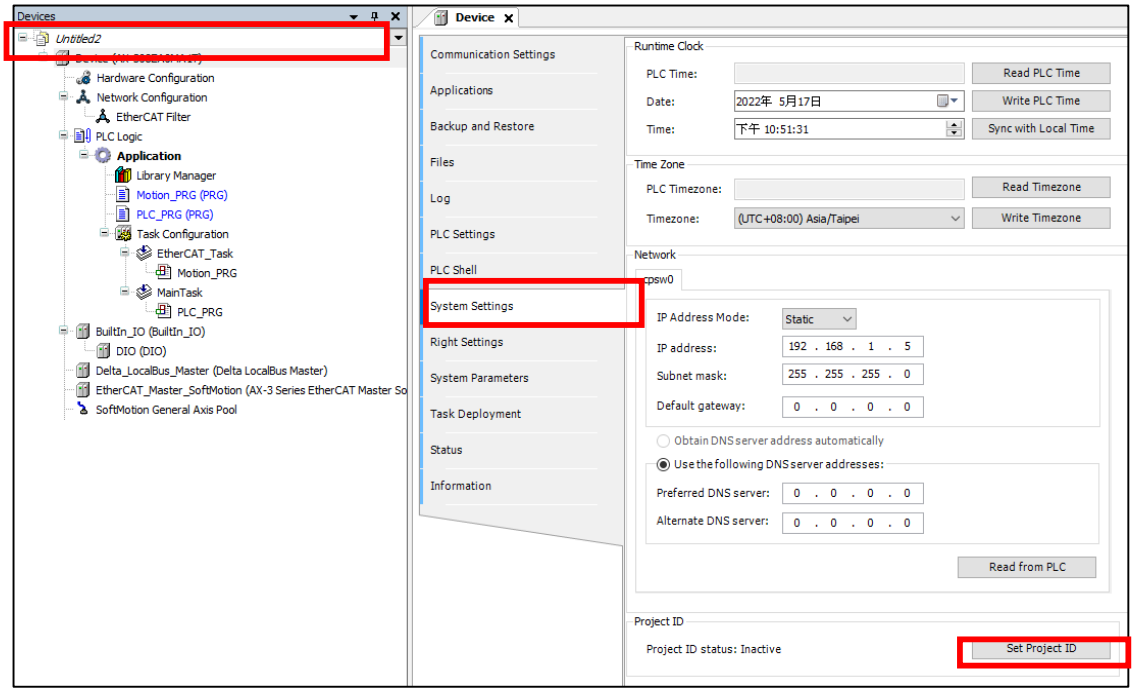
Follow the instructions:

Password type		Instructions
Designer-AX Project	Engineering ID (Project ID)	When downloading a project, the "Project ID" set in the Designer-AX project and the " PLC ID" set in the host must be exactly the same to operate. So it can be used to restrict certain items to download only to certain hosts.
PLC	PLC identification code (PLC ID)	

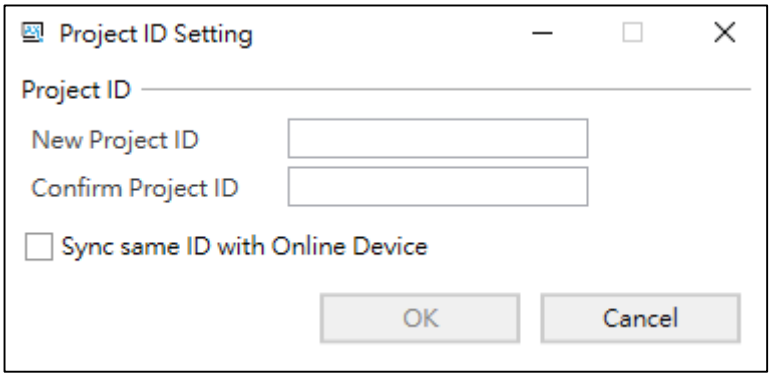
9.2 Setting Project ID

Follow the steps to set Project ID:

1. Create a Project, take AX-308EA0MA1T as an example.
2. Double-click **Device (AX-308EA0MA1T)**.
3. Click **System Settings** > **Set Project ID.**



4. Enter the password (identification code). Note that the length of the password is 4~16 characters. Click **OK** after setting is completed.

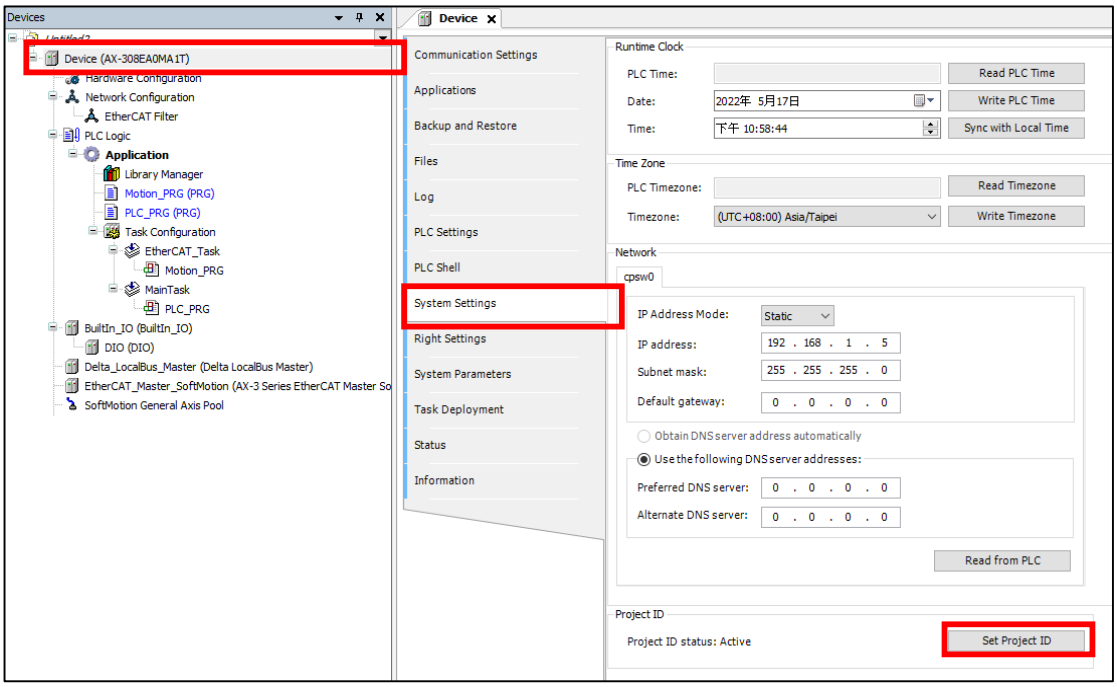


5. Once the setting is successful, the status of the Project ID displays "Active".

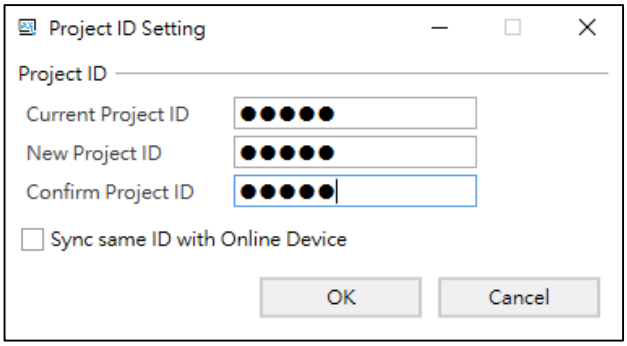
9.3 Modifying Project ID

Follow steps to modify Project ID:

1. Only when the Project ID status is displayed as "Active", it can be modified.
2. Click **Device (AX-308EA0MA1T)** > **System Settings** > **Set Project ID.**

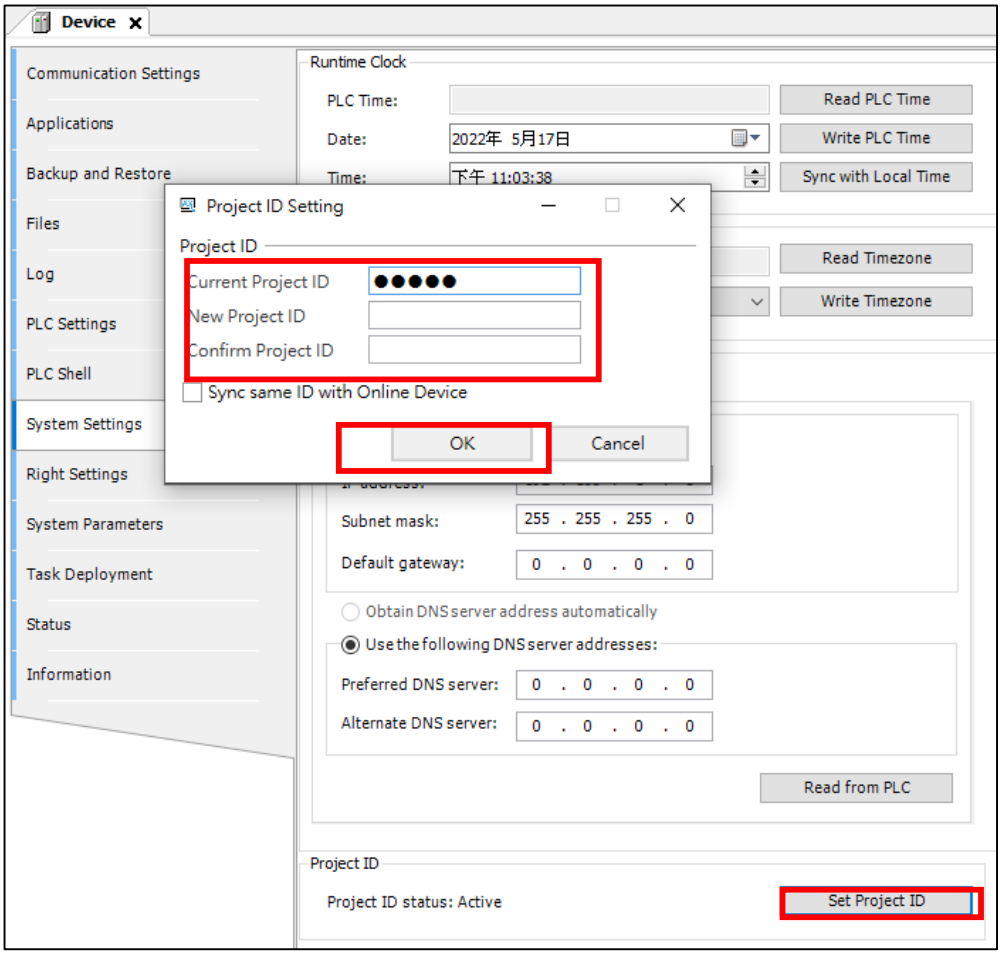


3. Enter the **Current Project ID** (original password), enter **New Project ID** (new password), **Confirm Project ID** (confirm new password), and click **OK**.



9.4 Clear Project ID

1. Click **Device (AX-308EA0MA1T)** > **System Settings** > **Set Project ID.**
2. Enter the **Current Project ID**.
3. Leave the **New Project ID** and **Confirm Project ID** blank**.**
4. Press **OK** to complete the setup.



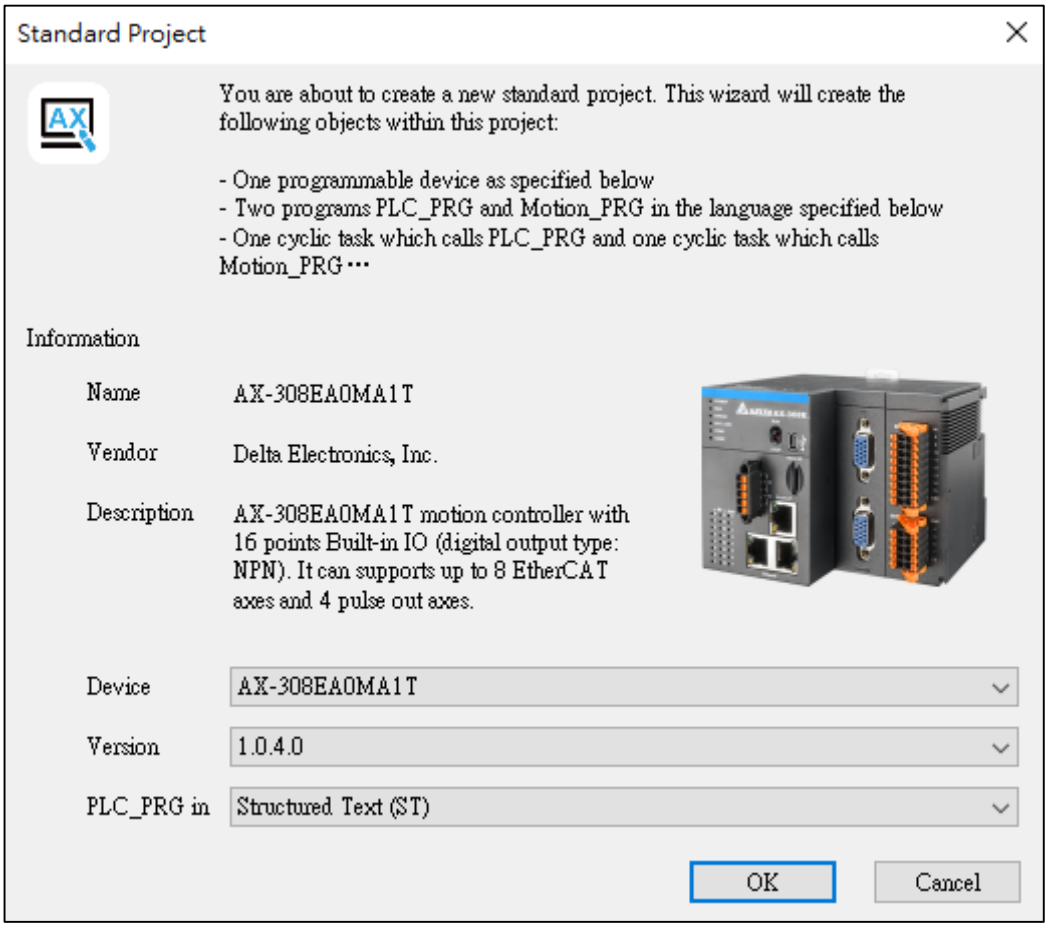
9.5 Set PLC ID

Note:

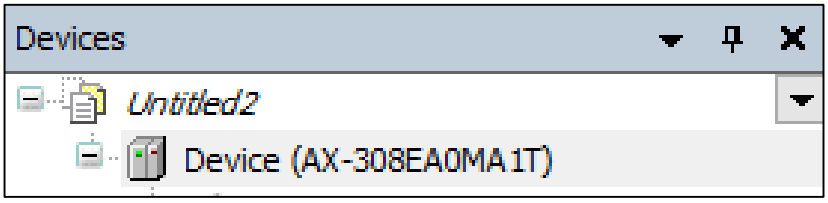
1. This section applies to AX-3 series controllers, and the firmware version must be greater than or equal to 1.0.3.0.
2. It supports online tuning only.

Follow the steps below to set PLC ID:

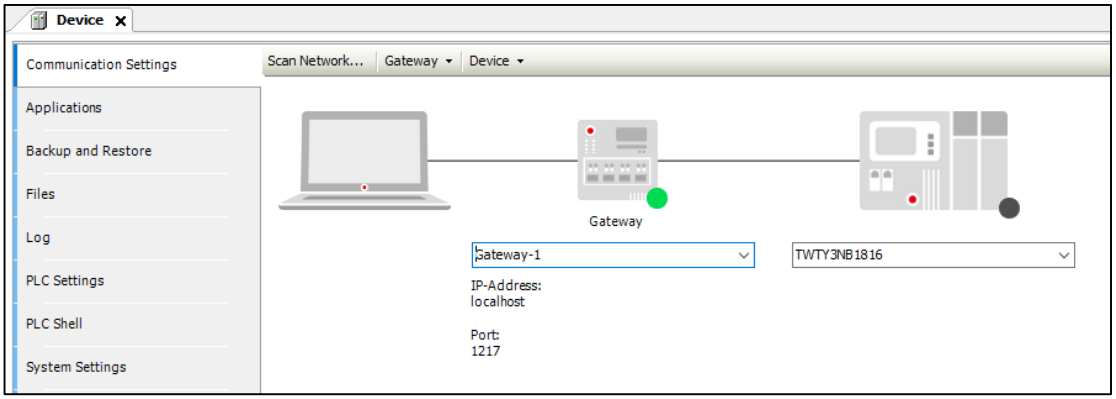
1. Create an AX-3 series project, take AX-308EA0M1T as an example.



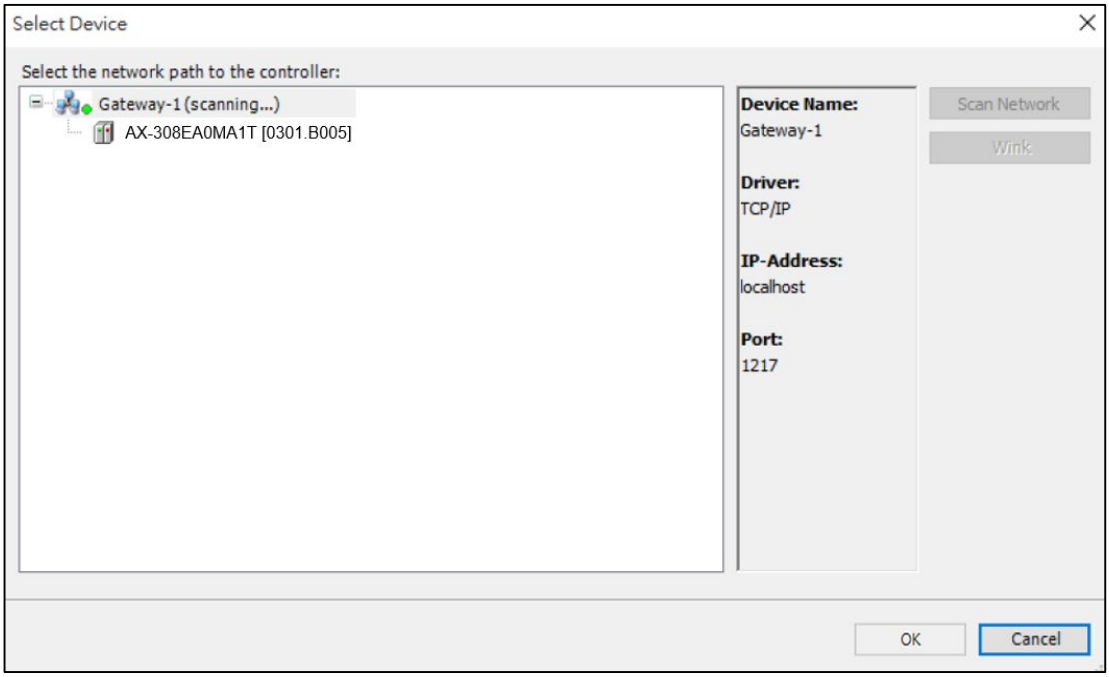
2. Double-click **Device AX-308EA0M1T**.



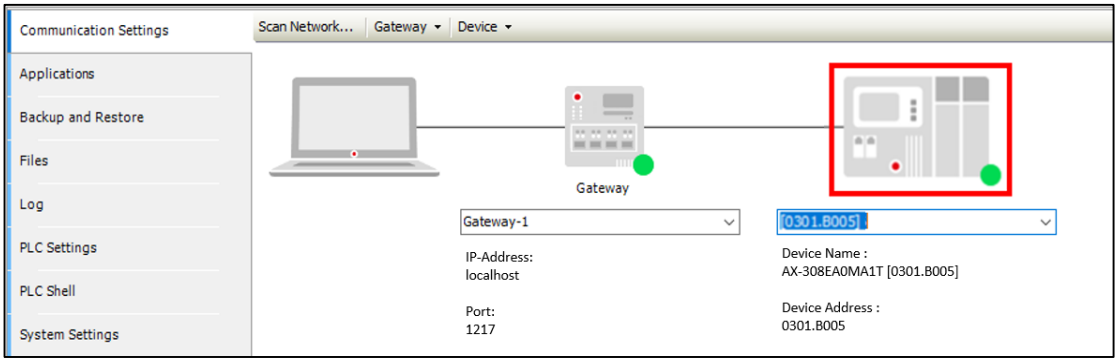
3. Make sure that the network cable is plugged in, and click **Scan Device**.



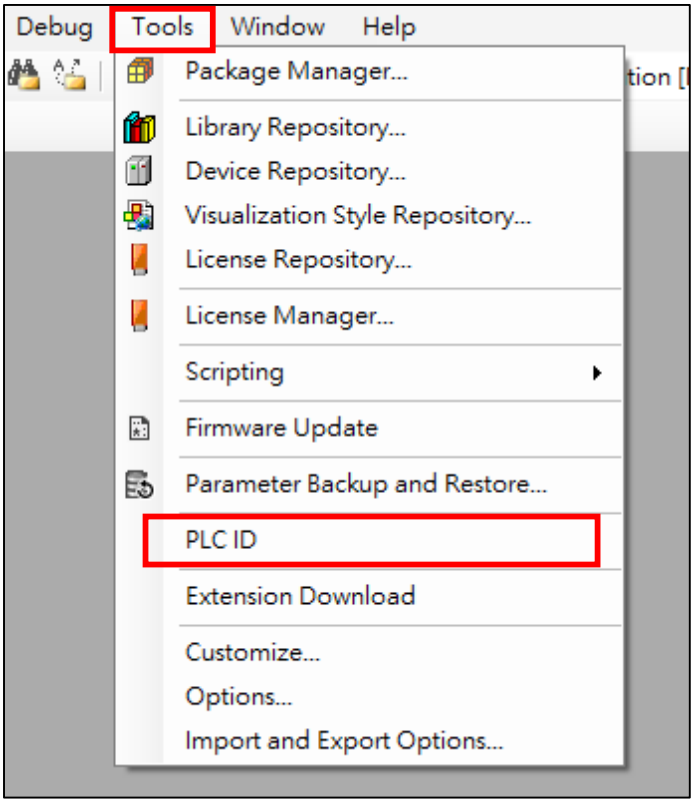
4. Select the device to be connected and click **OK**.



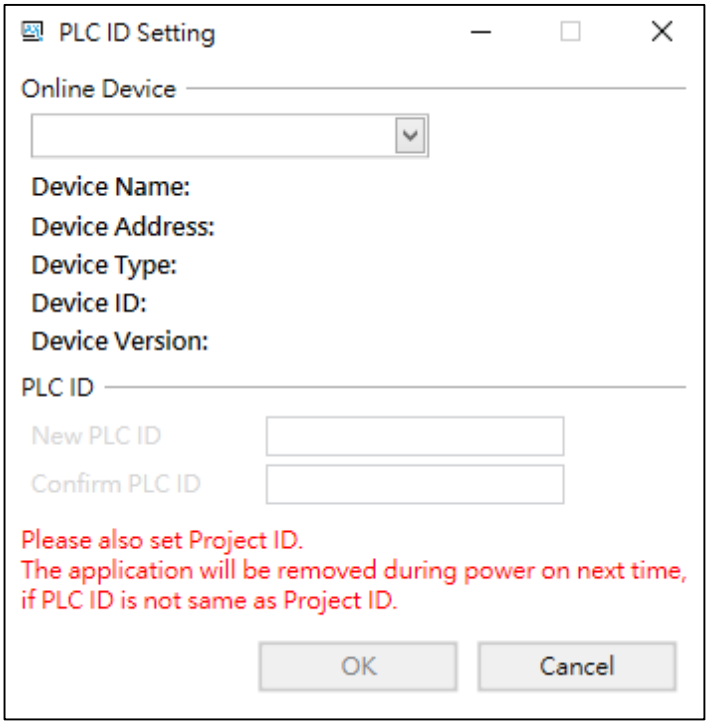
5. The device will be automatically scanned, and the activation (state) indicates that the connection is successful.



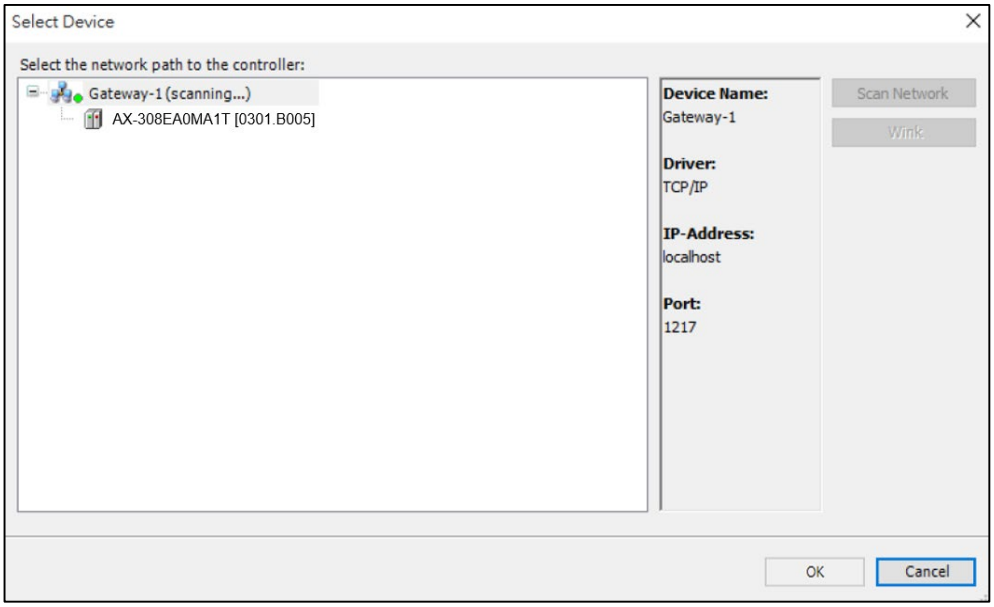
6. Click **Tools** > **PLC ID**.



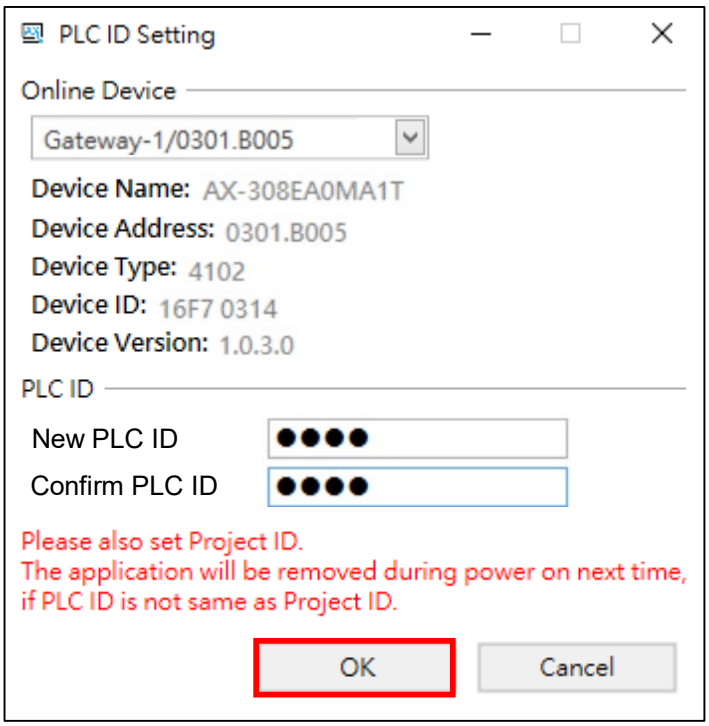
7. Select the device whose PLC ID needs to be set.



8. After selecting, press **OK**.



9. Once the new PLC ID is entered (please enter the password), click **OK**.

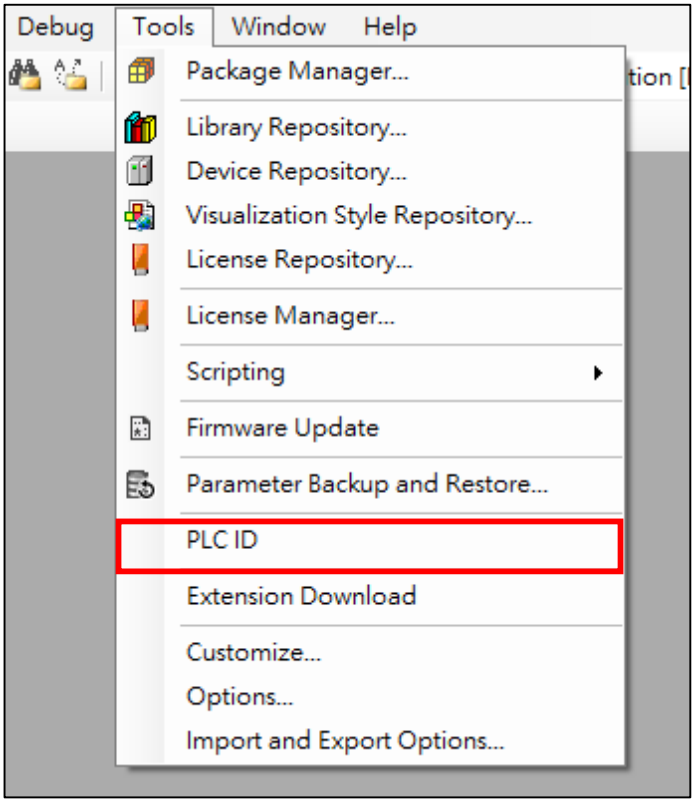


Result: The Project ID is set successfully as shown in the figure.

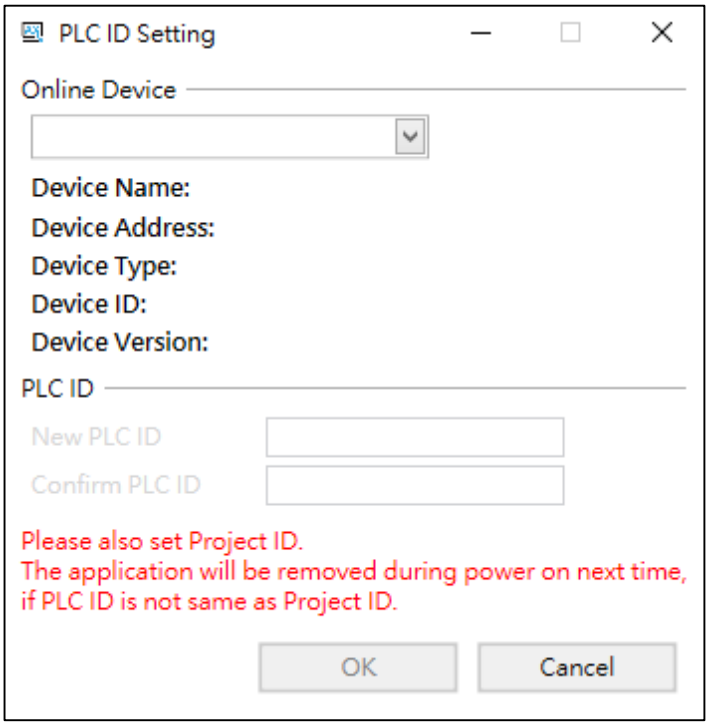
9.6 Modify PLC identification code

Follow the steps to modify the PLC ID:

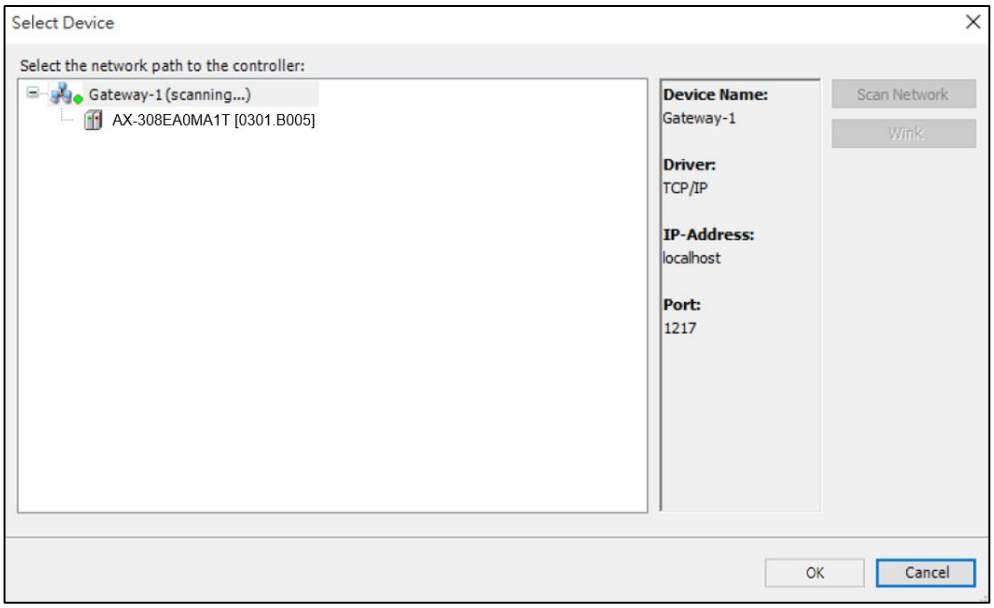
1. Click **Tools** > **PLC ID**.



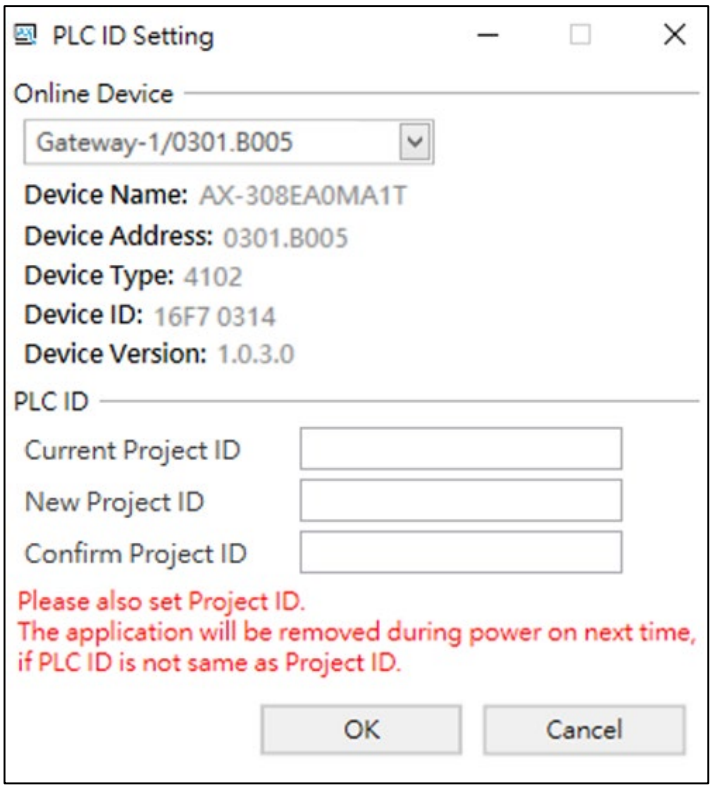
2. Select the device whose PLC ID needs to be set.



3. After selecting, click **OK**.



4. Enter the **Current PLC ID** (original password), enter the **New PLC ID** (new password), **Confirm PLC ID** (confirm the new password), and click **OK**.

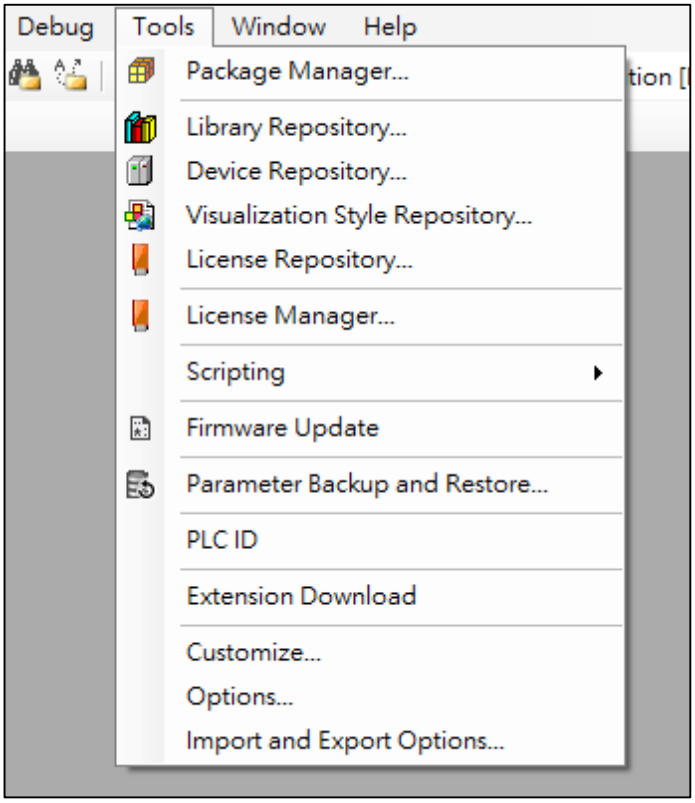


Result: PLC ID is modified successfully.

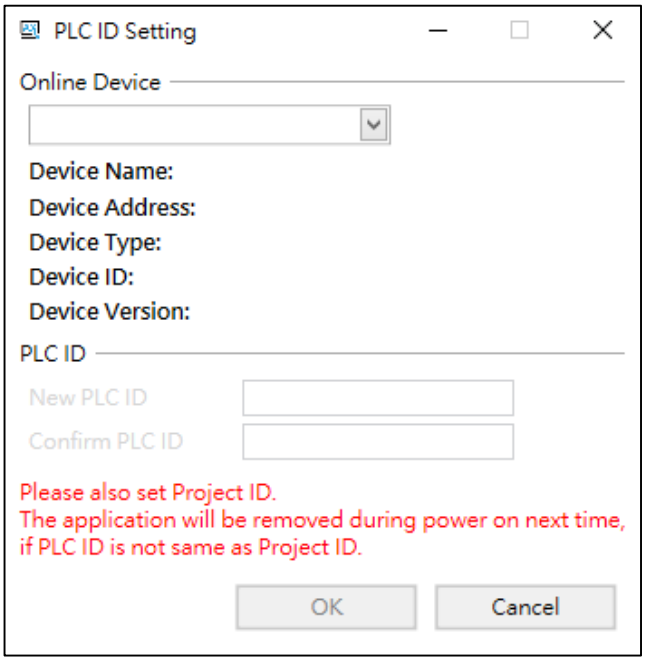
9.7 Clear PLC ID

Follow the steps to clear the PLC ID:

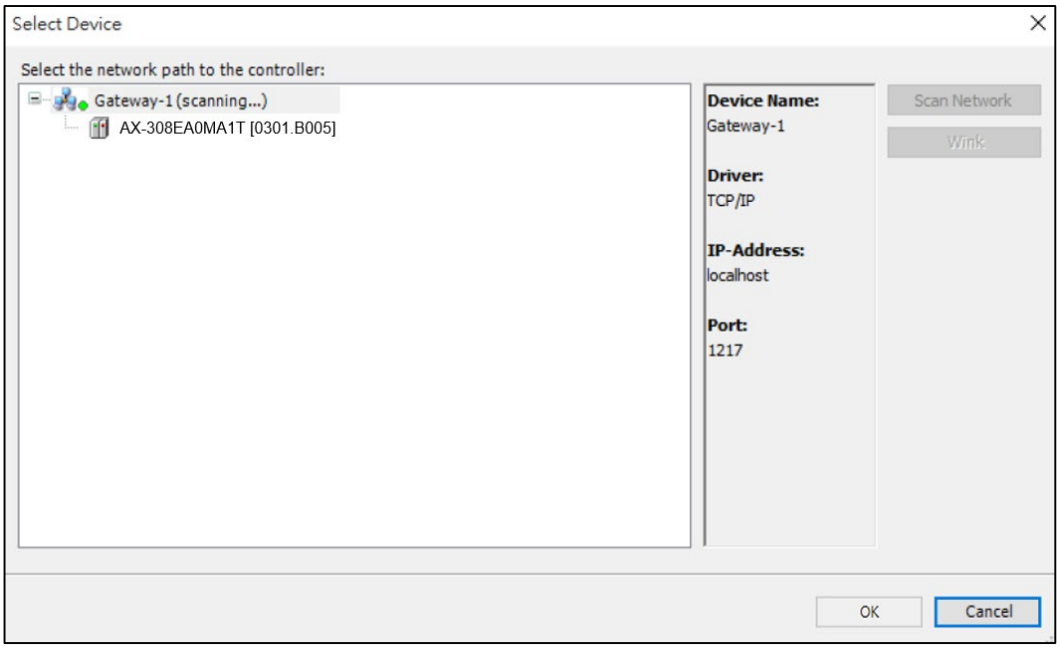
1. Click **Tools** > **PLC ID.**



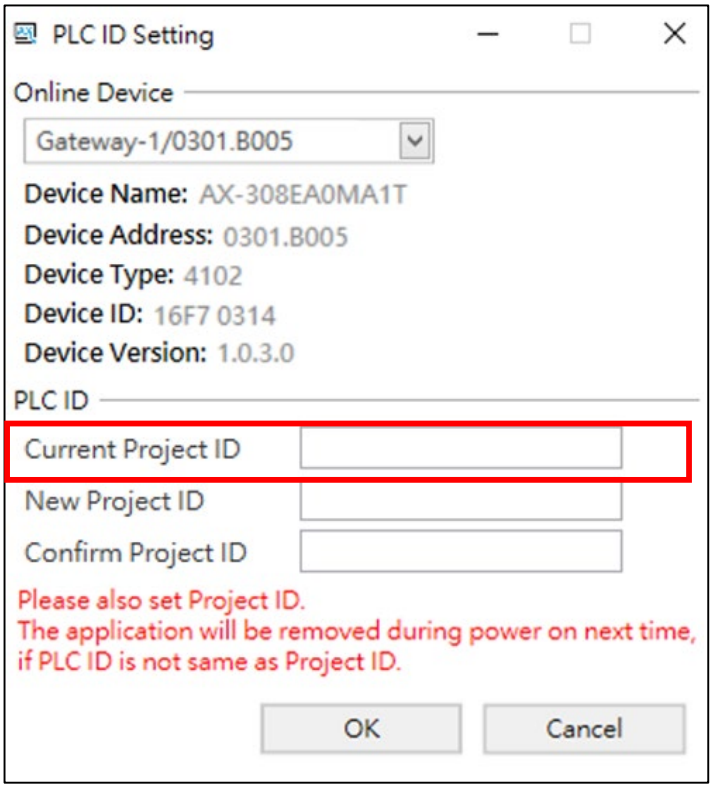
2. Select the device whose PLC ID needs to be set.



3. After selecting, press **OK**.



4. Enter the **Current PLC ID** (original password), leave **New PLC ID** and **Confirm PLC ID** blank, and click **OK**.

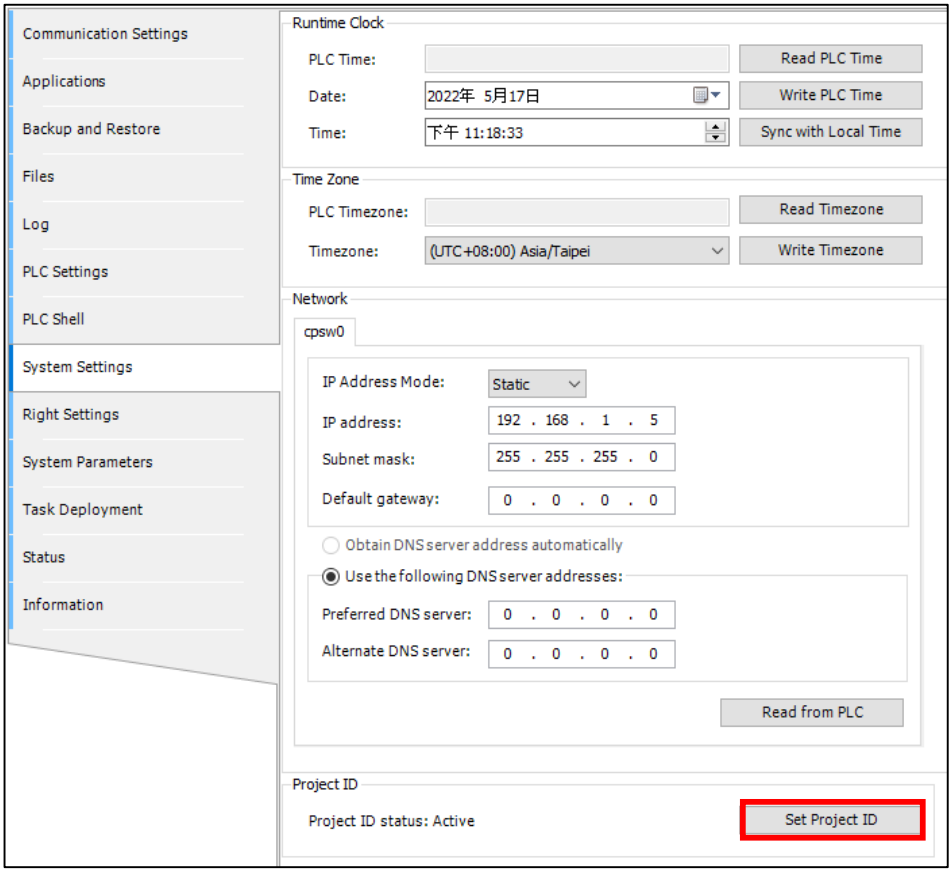


Result: PLC ID is cleared successfully.

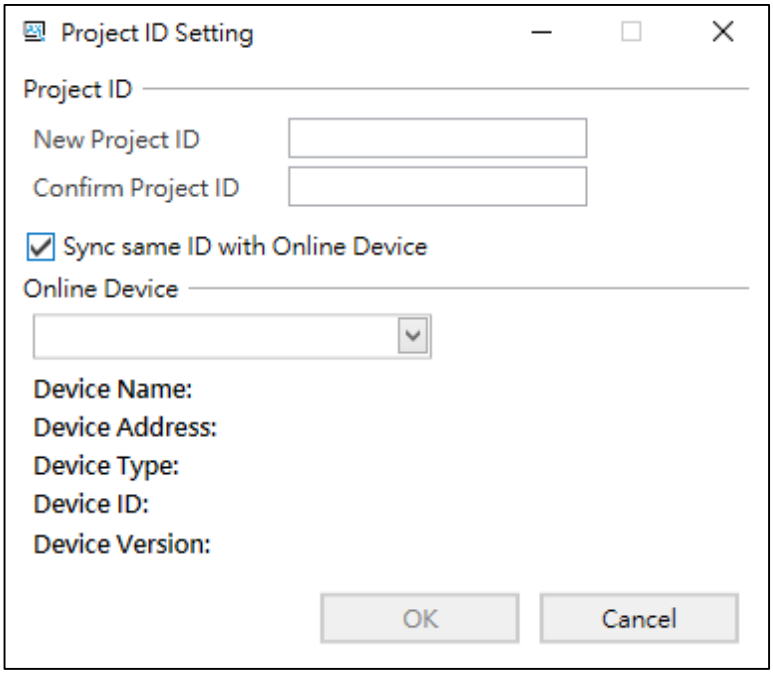
9.8 Set Project ID and PLC ID Simultaneously

Follow the steps to set the Project ID and PLC ID at the same time:

1. Click **Set Project ID**.



2. Enter the **New Project ID** and check **Synchronize the same identification code to the online device**. At this time, the PLC ID will also be set to the same.password as the Project ID. Click **OK**.

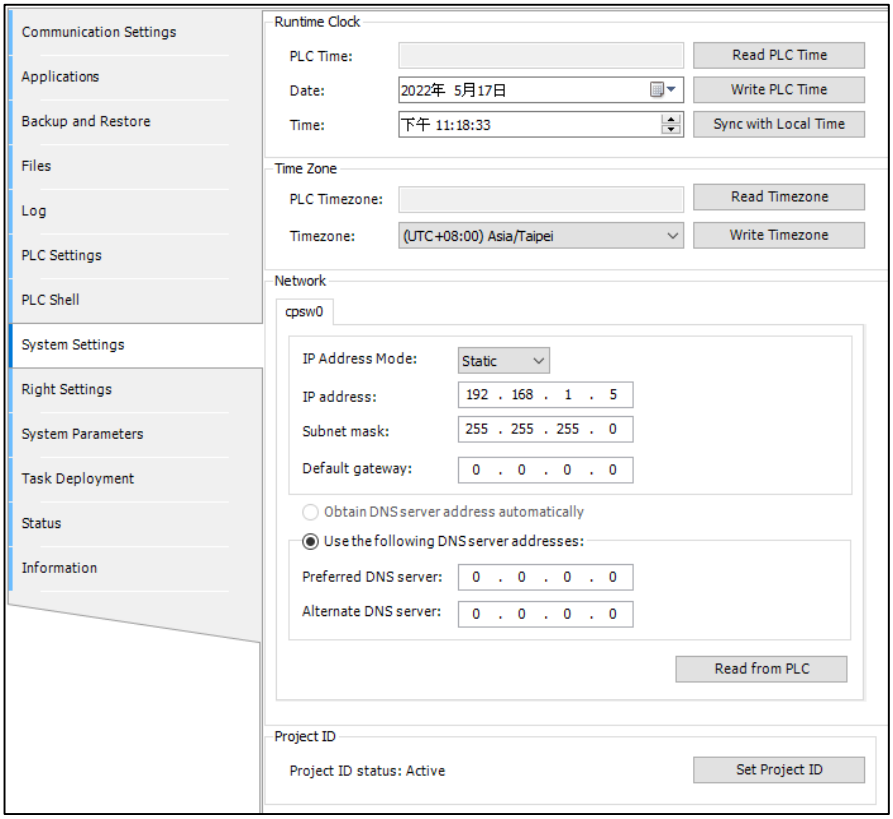


Result: The setting is successful.

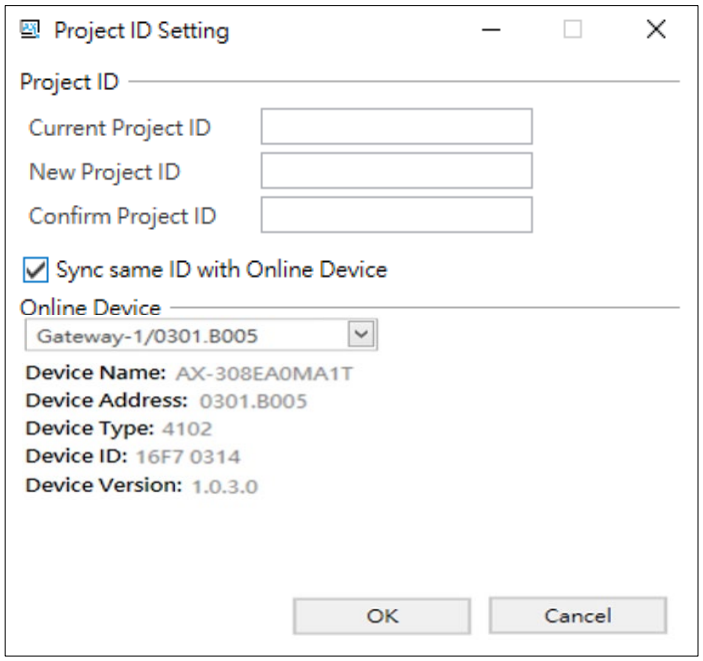
9.9 Modify Project ID and PLC ID Simultaneously

Follow the steps to modify the Project ID and PLC ID at the same time:

1. Click **Set Project ID**.



2. Enter the **Current Project ID** and the **Current PLC ID.** Check **Sync same ID with Online Device**. At this time, the PLC ID will also be set to the same password as the Project ID. Enter the **New Project ID** and **Confirm Project ID**, and press **OK**.

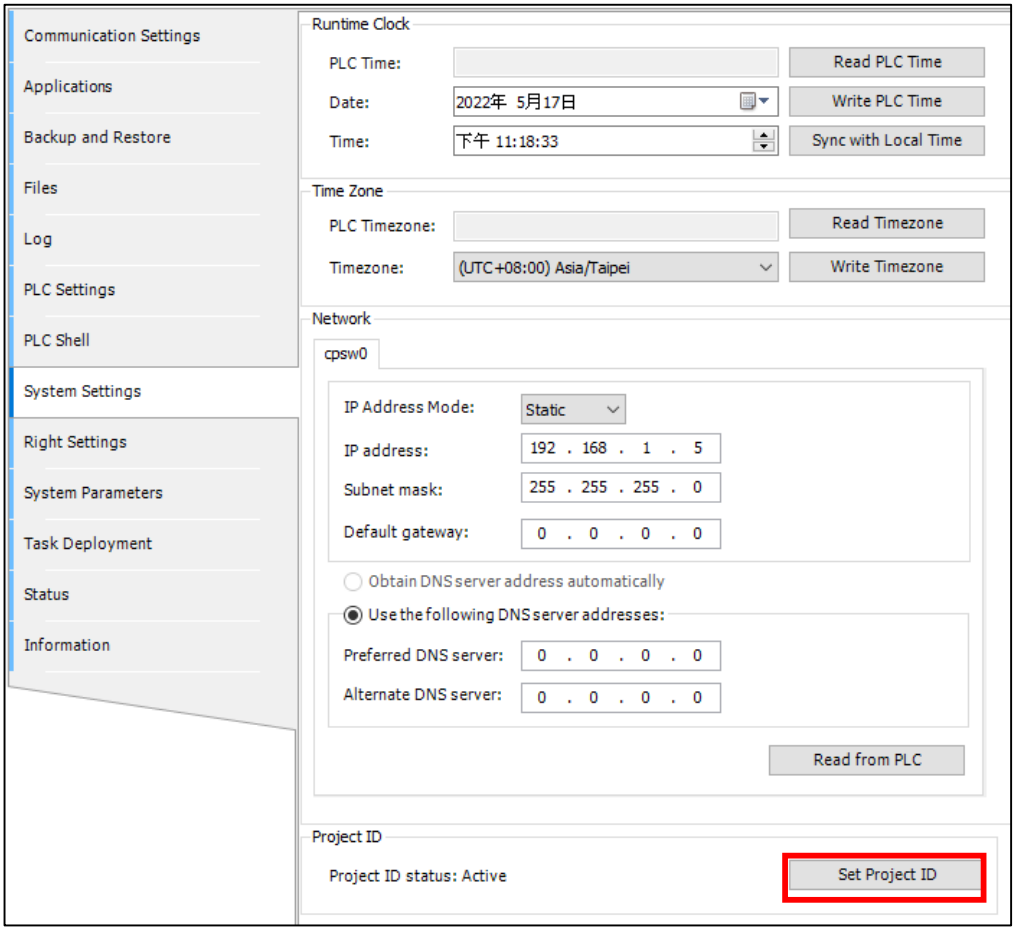


Result: The modification is done successfully.

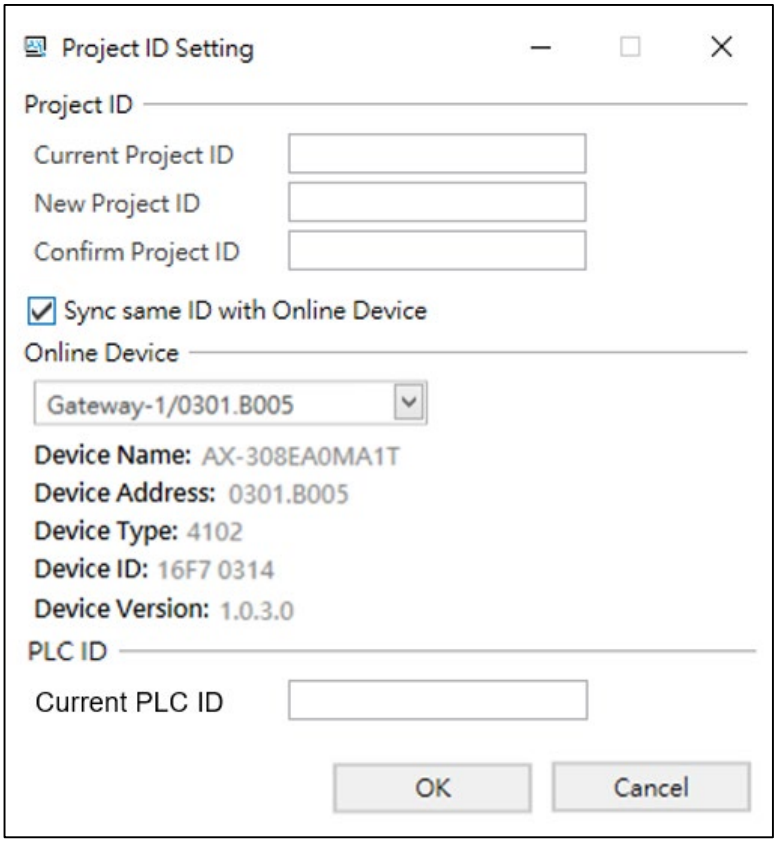
9.10 Clear Project ID and PLC ID Simultaneously

Follow the steps to clear the Project ID and PLC ID at the same time:

1. Click **Set Project ID**.



2. Check **Sync same ID with Online Device**, and enter the **Current Project ID** and the **Current PLC ID**. Leave the **New Project ID** and **Confirm Project ID** blank and click **OK**.



Result: It is cleared successfully.

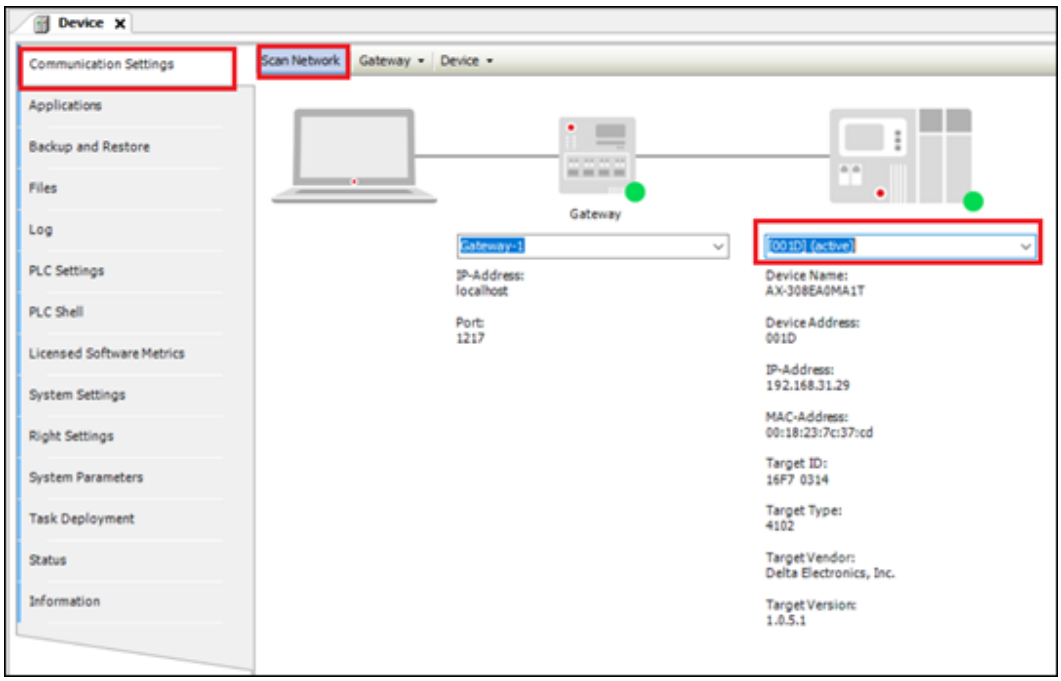
9.11 Right Settings

It applies to AX-3 and AX-8 series controllers. You cannot undo or redo settings on this pane.

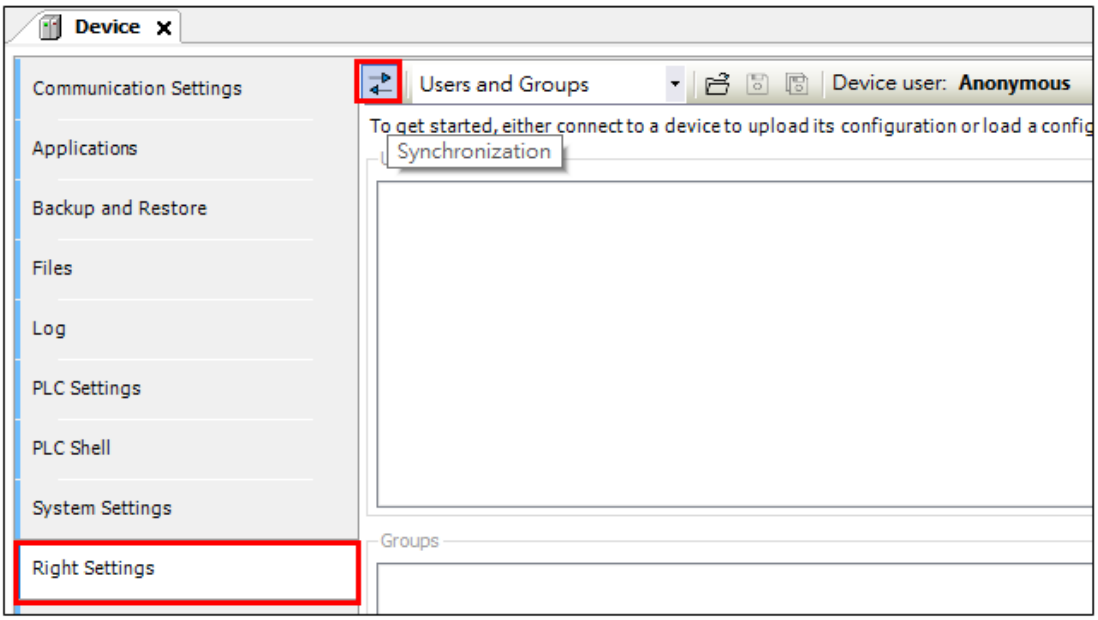
Starting from version 1.4.0, if you select the AX-3 series controller and its version is 1.0.5.0 or later, and select **Users and Groups** on the **Right Settings** pane, an encryption function is available for importing and exporting management files.

To set rights

1. Create a **New Project** or open an existing one**.**
2. Connect to the network first.



3. Select **Right Settings**, and then click  to synchronize with the controller.



4. The account password is “Administrator” by default. When logging in for the first time, you need to change the password from the default “Administrator” to another. If the password is lost, it can only be restored to the factory value.

9.11.1 Import management files

You can import users and user groups and access configurations from the local computer.

To import management files

1. On the **Devices** pane, double-click the controller.
2. On the **Device** editor, select **Right Settings**.
3. Click  .
4. Select the *.rsu2 or *.dum2 file.
   The *.dum2 file is the configuration file for all settings on the **User and Groups** pane.
   The *.rsu2 file is the configuration file for all settings on the **Right Settings** pane.
   In the synchronization mode, DIADesigner-AX only supports importing the *.dum2 file.
5. (optional) In the **Enter Password** dialog, enter **Password**, and then click **OK**.
   The password was assigned when the user management file was exported.
   This encryption function is only available when you select:
   - The AX-3 series controller and its version is 1.0.5.0 or later.
   - Select the **User and Groups** pane.

   Note: The import of a device user management by means of a *.dum2 file completely overwrites the existing user management on the device. To log in to the device again afterward, you need authentication data from the recently imported user management.
6. In the **Device User Logon** dialog, enter **Username** and **Password** and click **OK**.
7. Click  and log in again.

9.11.2 Export management files

You can only export users and user groups and access configurations in the synchronization mode.

To export user management files

1. On the **Devices** pane, double-click the controller.
2. On the **Device** editor, select **Right Settings**.
3. Click  .
4. Click  to only export the configurations in the **Users and Groups** pane or the **Access Rights** pane or click  to export all configurations on the **Right Settings** pane.
5. Enter **File name**, and click **OK**.

(optional) In the **Enter Password** dialog, enter **Password**.

This encryption function is only available when you select:

- The AX-3 series controller and its version is 1.0.5.0 or later.
- Select the **User and Groups** pane.

Note: You need to enter this password when importing the *.dum2 file.

6. Click **OK**.

9.11.3 Set up user and group

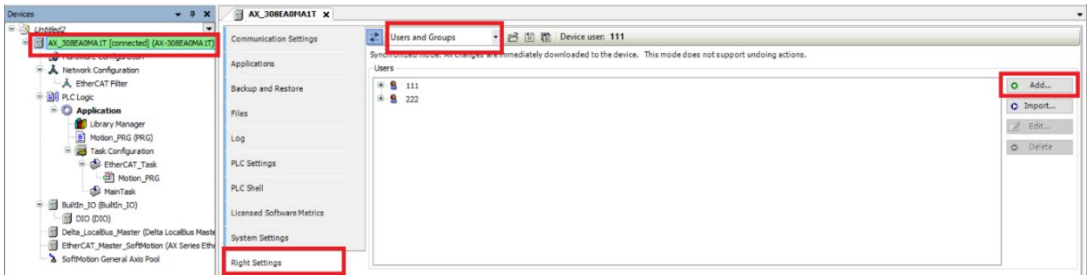
You can adjust user management for the first time in a project and adjust the definitions of users and groups to which they belong.

Prerequisites

- Turn on and synchronize the device where user management will be established and ensure that the configuration has not been adjusted.

To add a user

1. On the **Devices** pane, double-click the controller.
2. On the **Device** editor, select **Right Settings** > **Users and Groups**.
3. Click **Add** next to the **Users** pane.



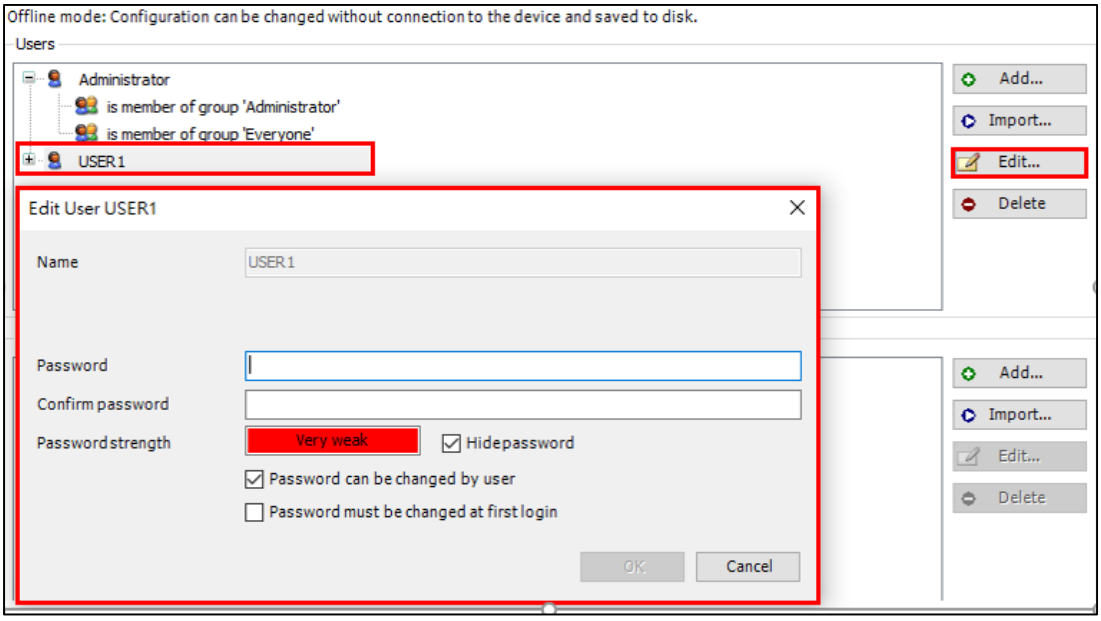
4. In the **Add User** dialog, enter **Name**, select the default group for the user being added, and then enter **Password**, and **Confirm password**.
5. Click **OK**.

To add a user group

1. On the **Devices** pane, double-click the controller.
2. On the **Device** editor, select **Right Settings** > **Users and Groups**.
3. Click **Add** next to the **Groups** pane.
4. Enter **Name**, and then select group members.
5. Click **OK**.

To modify the user password

1. On the **Users** pane, select the user that you want to change the password for.
2. Click **Edit**.
3. In the **Edit User** dialog, enter **Password** and **Confirm password**.
4. Click **OK**.

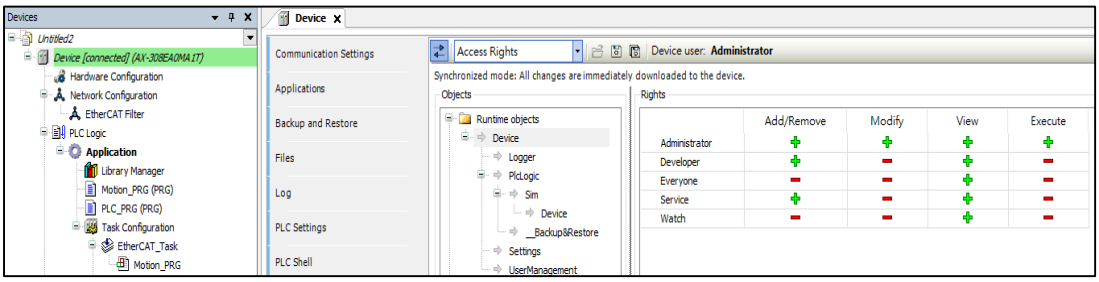

Offline mode: Configuration can be changed without connection to the device and saved to disk.
Users
Administrator
is member of group 'Administrator'
is member of group 'Everyone'
USER1
Add...
Import...
Edit...
Delete
Edit User USER1
Name
USER1
Password
Confirm password
Password strength
Very weak
Hide password
Password can be changed by user
Password must be changed at first login
OK
Cancel
Add...
Import...
Edit...
Delete

9.11.4 Set up access rights

You can configure whether group members have the right to view, edit, or delete objects, and add or delete sub-objects in objects.

To configure the access rights

1. On the **Devices** pane, double-click the controller.
2. On the **Device** editor, select **Access Rights**.



3. On the **Objects** pane, select the object that you want to configure the access right for.
4. On the **Rights** pane, double-click the access.