DIADesigner-AX Software Manual

www.deltaww.com

Smarter. Greener. Together.

Copyright notice

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

Limitation of Liability

The contents of this user manual are only for the use of 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	2020/12/08	The first version was published.
2nd	2021/02/19	1. Chapter 3.1: Added Support Controller AX-308EA0MA1P, AX-364ELA0MA1T, AX-300NA0PA1, AX- 324NA0PA1P 2. Chapter 4.1.1: Added how to Add RIO Module Process. 3. Chapter 4.1.2: Added Models that Support the firmware update function. 4. Chapter 4.2.2: Updated the running clock configuration tab to the system settings tab and add the device IP address setting field. 5. Chapter 4.2.5: Updated Ethernet General Tab. 6. Chapter 4.2.7: Added high-speed IO Settings,divided into AX-364EL/AX-308 and AX-324. 7. Chapter 4.4.4: Added Introduction to the new free encoder. 8. Chapter 7.1: Added Introduction to the new device storage library.

Table of Contents

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 the user a convenient and efficient development environment.

1.2 DIADesigner-AX Overview

Features

- 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.

There are 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: Software Setup

2.1 Installation and Uninstallation

2.1.1 Installing DIADesigner-AX

Pre-requisites:

Ensure that the target computer follows the minimum criteria mentioned in *2.2 System Requirements*, and DIAInstaller is operating in it.

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

https://diastudio.deltaww.com/home/downloads

Follow the steps to install software:

1. Open DIAInstaller, and check for latest version of DIADesigner-AX.
2. Click *Download* to download DIADesigner-AX installation file, as shown in Figure 2-1.

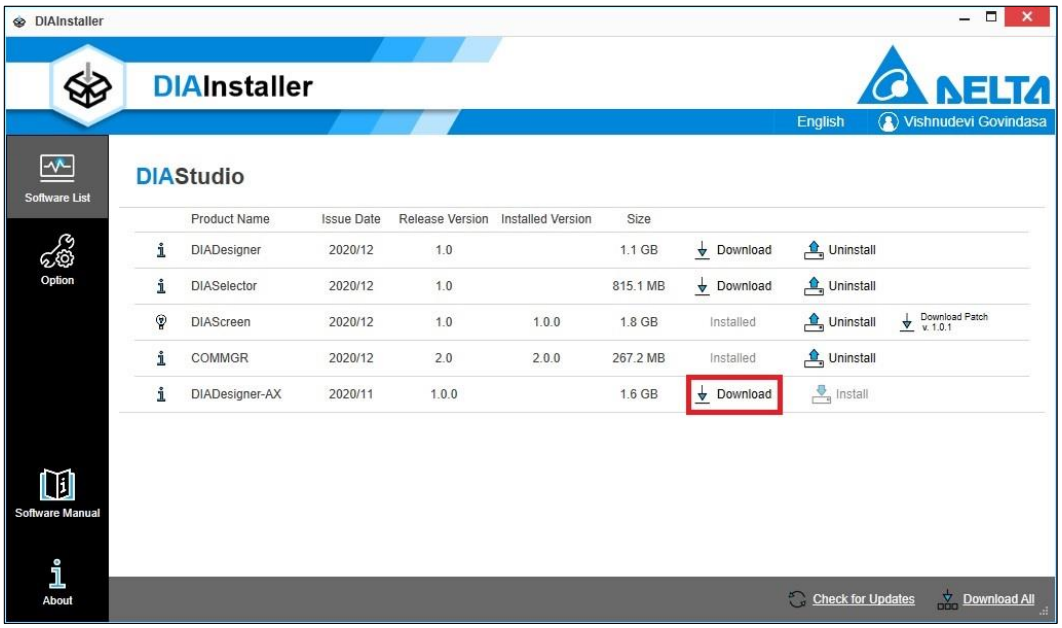


Figure 2 - 1: Downloading DIADesigner-AX

3. After completing the download, click *Install* to start installation in background.

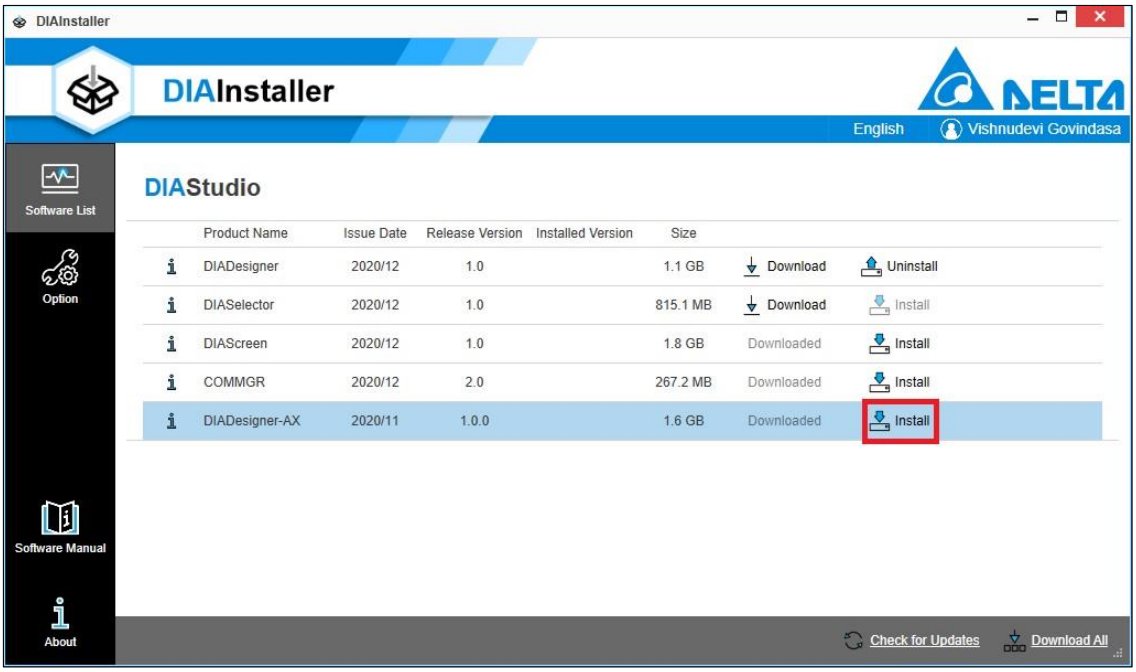


Figure 2 - 2: Installation of DIADesigner-AX

Result: DIADesigner-AX has been installed.

***NOTE**:*

1. *Click Check for Updates to refresh in case user install or uninstall software themselves without using DIAInstaller.*
2. *Click Download All to download all available software.*
3. *User can assign a dedicated directory to install by clicking Option button.*
4. *User can set options in Option.*

Figure 2 - 3: DIAInstaller Option

2.1.2 Uninstalling DIADesigner-AX

Follow the steps to uninstall DIADesigner-AX:

1. Open DIAInstaller, and then click *Uninstall* in target software.

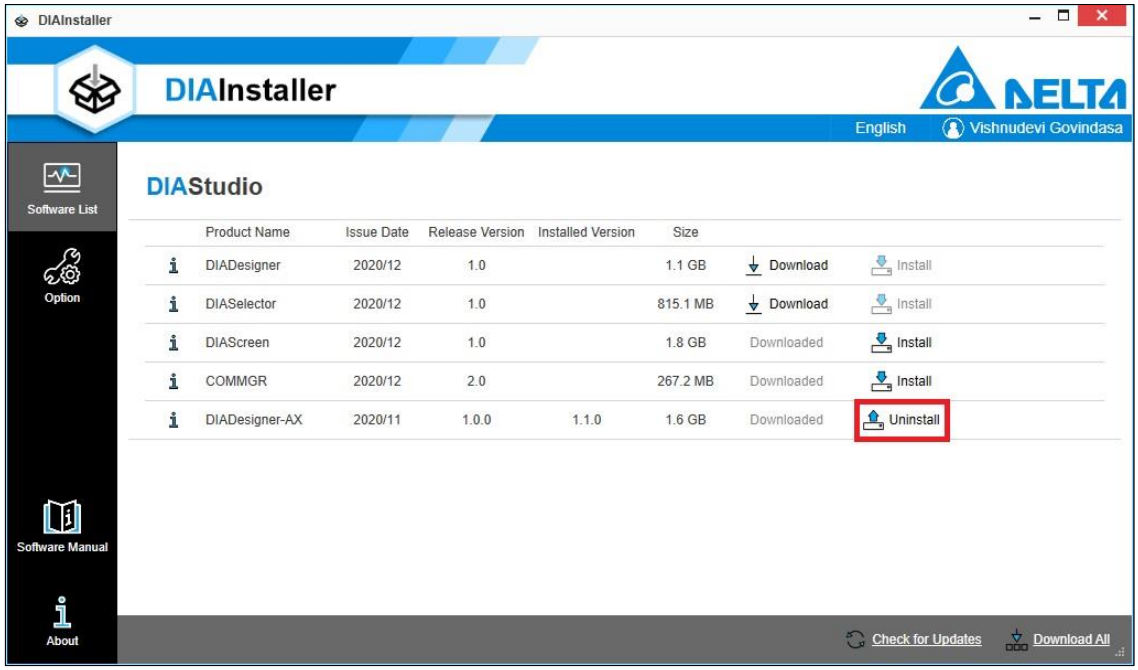


Figure 2 - 4: Uninstallation of DIADesigner-AX

2. Uninstallation process will be started in the background.

   Result: DIADesigner-AX is uninstalled.

NOTE: *Refer to DIAInstaller User Manual to complete DIADesigner-AX uninstallation.*

2.1.3 Update of DIADesigner-AX

Follow the steps to update DIADesigner-AX:

Open DIAInstaller. If there is a new version available, a  new version icon is displayed as shown in the following figure.

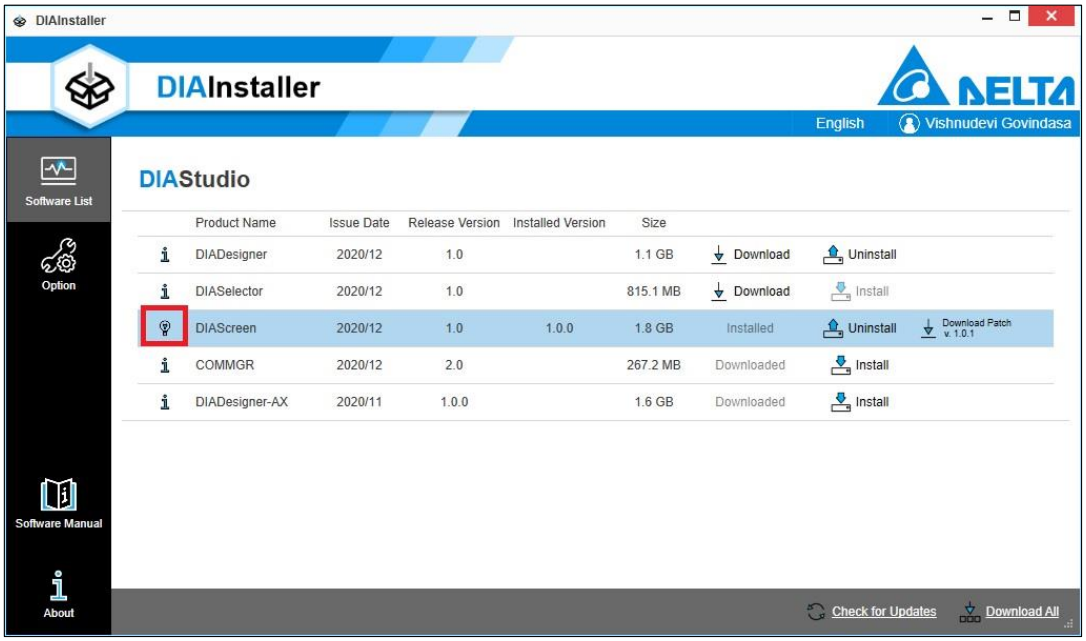


Figure 2 - 5: New version icon of DIAInstaller

1. Click *Download Patch* button, the latest update file will be downloaded.
2. After download is completed, click *Install* button to install update in background.

***NOTE**: User can set options* like *check for update frequency in Option.*

2.2 System Requirements

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

Please refer to the table below for software installation.

Item	System Requirement
Runtime System	The DIADesigner-AX runs all versions of the system in real time AX-8xxEP0, AX-308EA0MA1T
Operating System	Windows 7 / 8.1/10 32/64 bits
CPU	Intel Celeron 540 1.8 GHz (min.), Intel Core i5 M520 2.4 GHz (min.)
Memory	2GB or above (recommend to use 4GB or more)
Hard Disk Drive	10GB or more
Monitor	Resolution 1920 x 1080 Pixels recommend
Keyboard/Mouse	General Keyboard Mouse or Windows compatible device
PC interface	EtherNet, USB, Serial port (depends on product interface)
Software	Need to support Microsoft .Net Framework 4.6.2

Chapter 3: Getting Started

3.1 Supported Devices

DIADesigner-AX V1.1 supports the following devices. Refer the device documentation for more information on the devices.

Type	Product Series
Controller	• AX-8xxEP0 Linux series • AX-8xxEP0 Windows series • MotionPLC: AX-308EA0MA1T, AX-308EA0MA1P, AX-364ELA0MA1T • Logic PLC: AX-300NA0PA1, AX-324NA0PA1P
AC Motor Drives	C2000 series, MS300 series
Servo Drive	ASDA-A2-E series, ASDA-B3-E series, ASDA-A3-E series
Remote IO	• 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 v1.1 are shown in the following table. Subsequent versions will gradually increase device support and features.

	Communication Setup	Network Configuration	Hardware Configuration	Parameter Setup	Program Edit
Controller	CH4	CH5	CH4	CH4	CH7
AC Drive				CH 3.3.1.2	N/A
Servo Drive				CH 3.3.1.2	

	Download
Controller	CODESYS Development System > Updating an Application on the PLC > Execution of a Download

	Download
AC Drive	Link: https://help.CODESYS.com/webapp/_cds_performing_a_download;product=CODESYS;version=3.5.10.0
Servo Drive	

3.3 Quick Start

3.3.1 Example

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 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.
- The compressor cools until the desired temperature is reached, minus a hysteresis of 1 degree. The hysteresis is intended to prevent the actual temperature from fluctuating too much around the set temperature, which would result in the compressor constantly switching itself off and on.
- 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 activity is controlled in the main program of the application. The signal management is controlled in another POU. The required standard function blocks are available in Standard library. Because no physical temperature sensors and no physical actuators are connected in this sample project, user will also write a program to simulate the increase and decrease of the temperature. This will allow user to monitor the operation of the refrigerator controller afterwards in online mode.

Preparation

User has installed DIADesigner-AX and launched it with the default profile. The frame window of the development system opens with the standard menu bar: No project is open yet.

3.3.1.1 Create a Project

Follow the steps to create a project:

1. Double-click on DIADesigner-AX shortcut icon on the desktop to open the software.

Figure 3- 1: DIADesigner-AX shortcut icon

2. Click  new project icon in the quick access toolbar.

   Or

Click *File* > *New Project*.

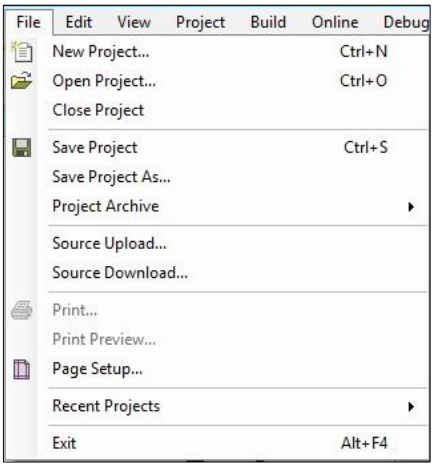


Figure 3- 2: Creating new project

Result: The *New Project* window opens with *Standard Project* as shown in the following figure. Specify a name and a storage location for the project and click *OK*.

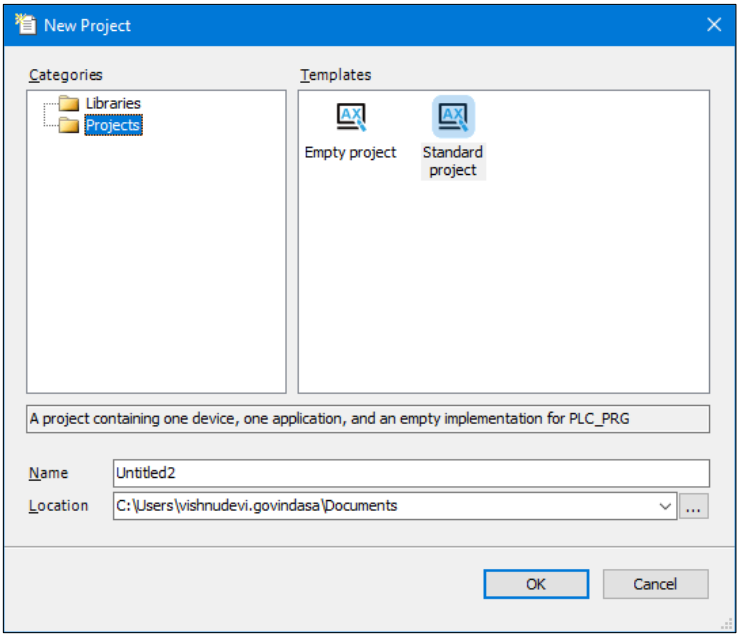


Figure 3- 3: New project template

3. Select the required model and PLC_PRG in *CFC*.

   Result: The project opens in the DIADesigner-AX frame window as shown in the following figure.

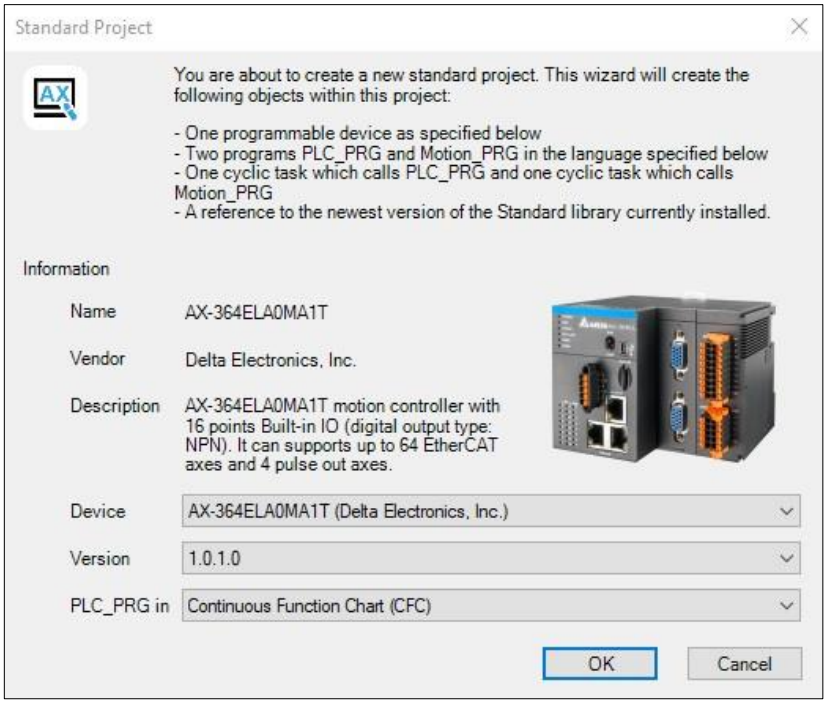


Figure 3- 4: **Standard project**

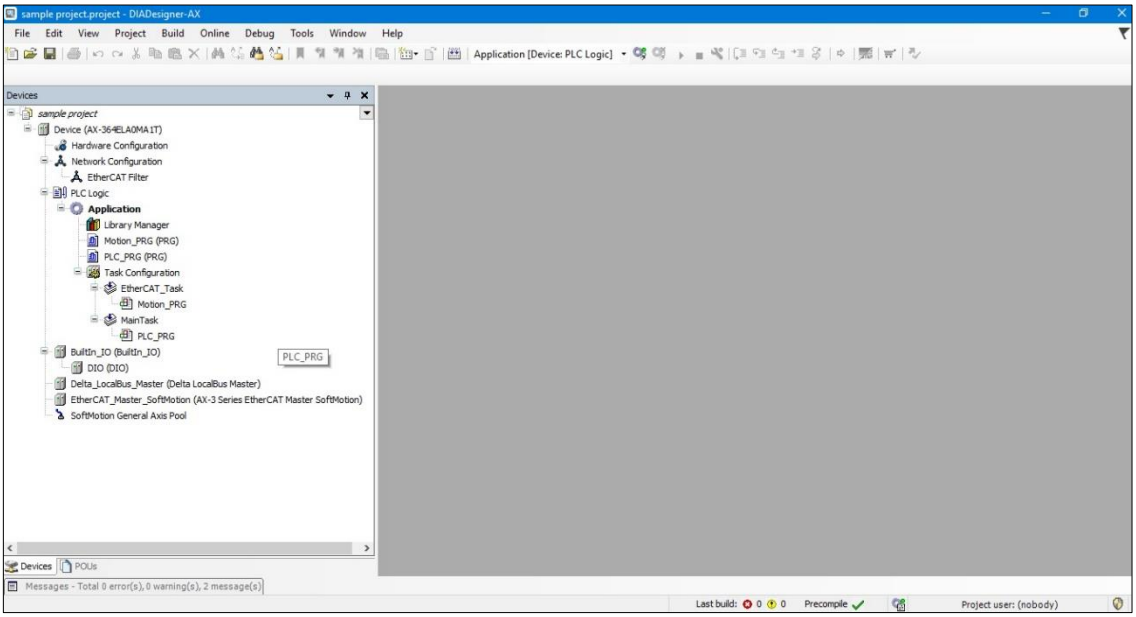


Figure 3- 5: **New project window**

3.3.1.2 Add Devices

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

A sample *Add Device* controllers is shown in the following figure.

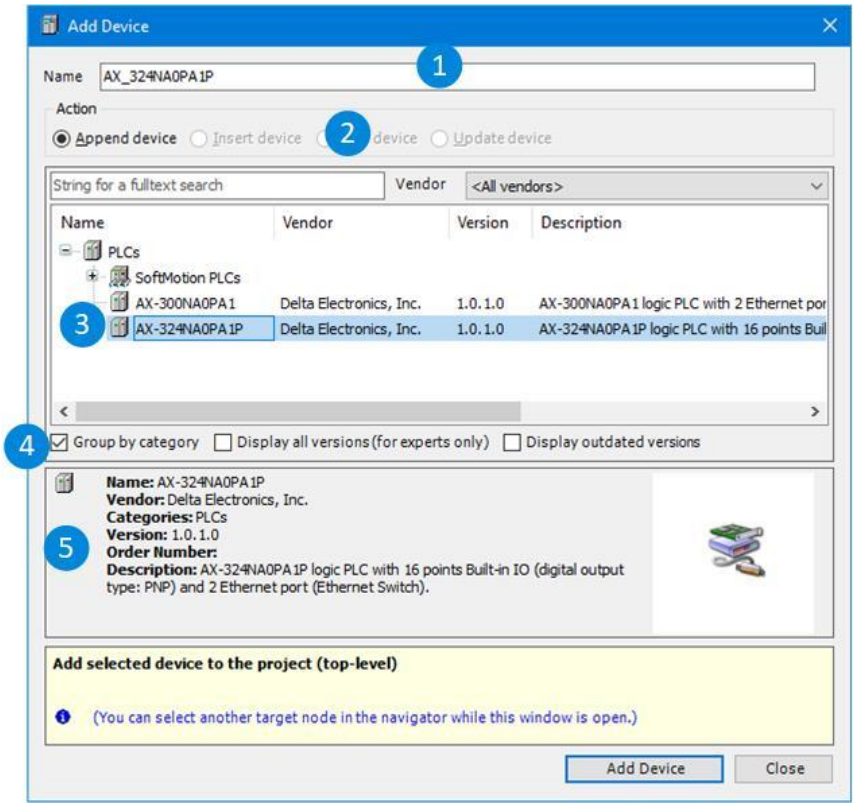


Figure 3- 6: **Add device window**

The *Add Device* window consists of the following information:

Legend	Description
1	Allows to either Append device, Insert device, Plug device or Update device.
2	Allows to search based on name and Vendor.
3	Displays the device tree with the device name, vendor, version and description of the devices.

Legend	Description
4	Allows to group the devices by category. To display either all versions or display only updated versions.
5	Displays the details of the selected device.

Follow the steps to add devices at project level:

1. Create a project (Refer to *3.3.1.1 Create a Project* for creating a project).
2. Right-click on the project name and click *Add Device*.

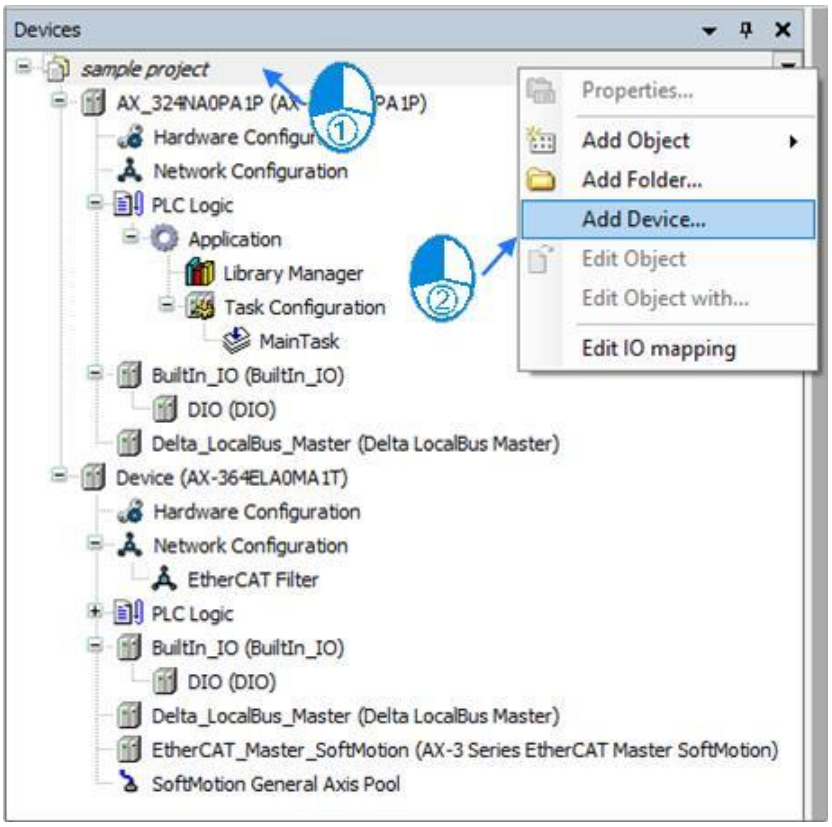


Figure 3- 7: **Adding device**

Result: The *Add Device* window displays with a list of devices as shown in the following figure.

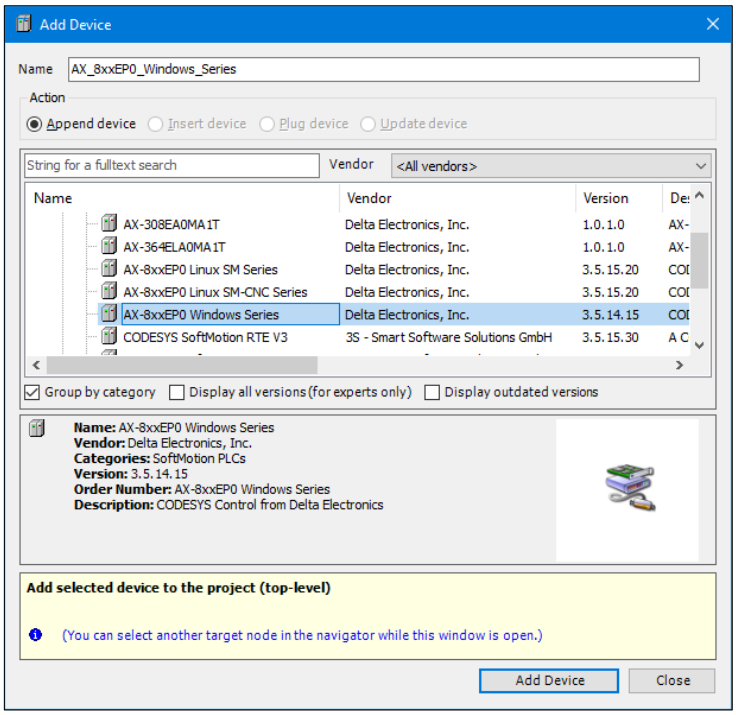


Figure 3- 8: **Device list of Add Device window**

3. Select the required device and click *Add Device* button.

   Result: The selected device is added to the project in *Devices* tree as shown in the following figure.

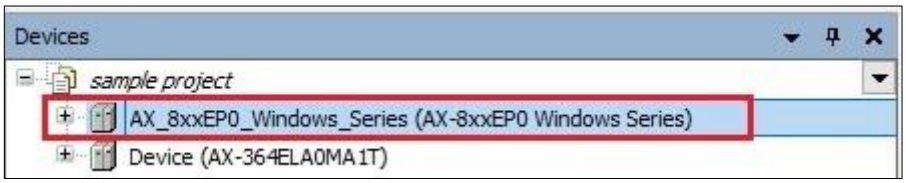


Figure 3- 9: **Add device to device tree**

Follow the steps to add devices to the project at device level:

1. Create a project (*3.3.1.1 Create a Project* for creating a project).
2. Right-click on the device and click *Add Device* as shown in the following figure.

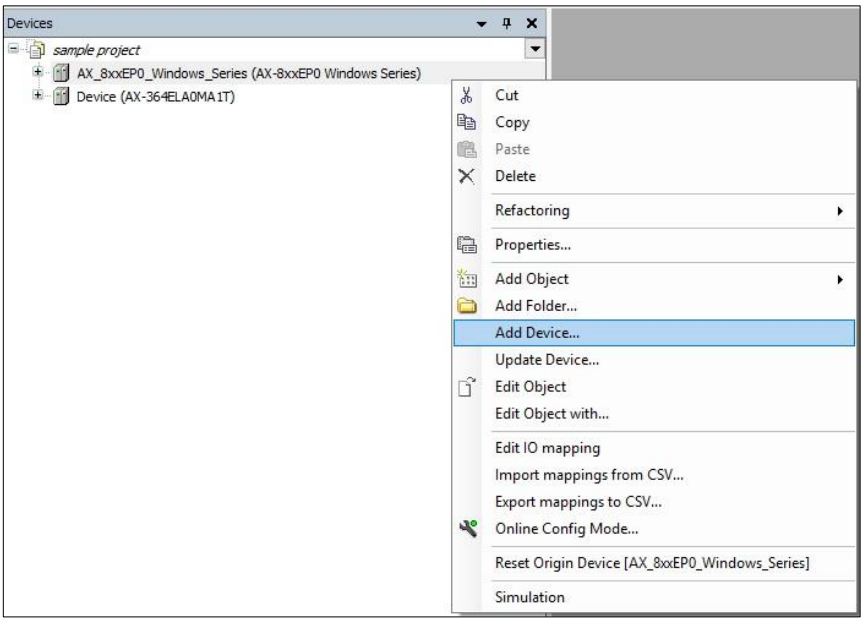


Figure 3- 10: Add Device

Result: The *Add Device* window displays with a list of devices.

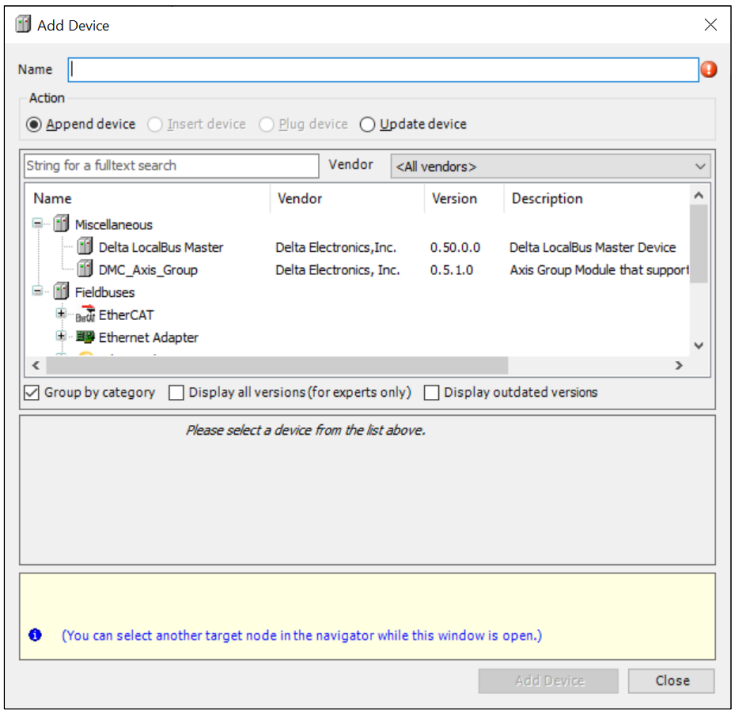


Figure 3- 11: List of devices in Add Device window

3. Select the required device and click *Add Device*.

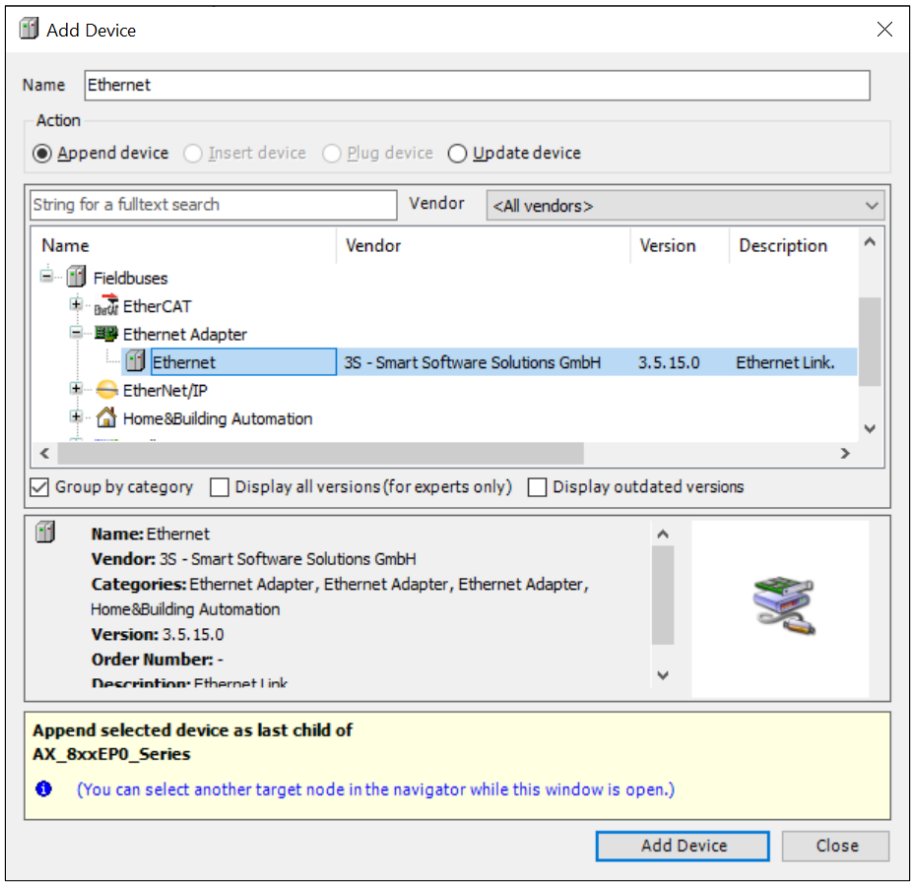


Figure 3- 12: Add Device

Result: The device is added to the *Devices*Tree as shown in the following figure.

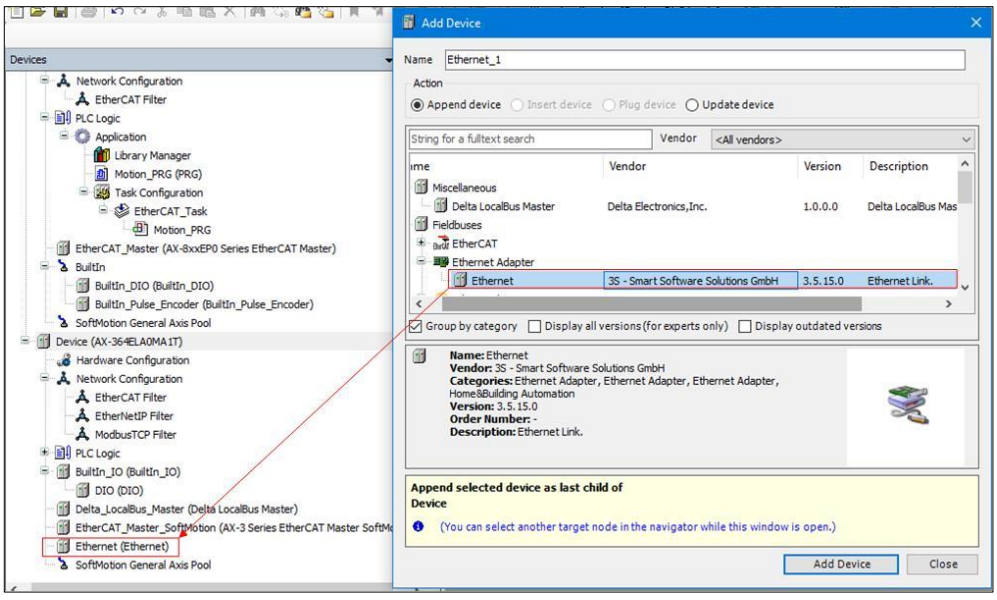


Figure 3- 13: Device added to project tree

3.3.1.3 Communication Settings

In the *Devices* tree, double-click *Device (AX-8xxEP0 window series)*.

The *Communication Settings* tab opens in the *Device* editor.

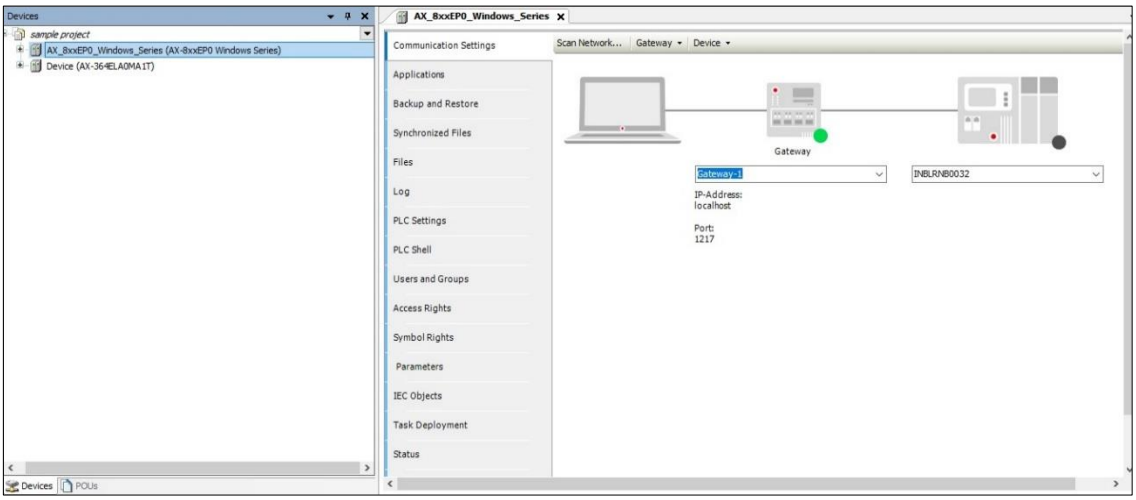


Figure 3- 14: Communication Settings tab

NOTE: *First, the user need to connect communication configuration with DIADesigner-AX, then need to define the local gateway server. If they already defined the gateway server, then it is shown on the Communication Settings tab. In this case, user can continue with Defining the communication channel.*

Now define the communication channel to the device, which is then used via the gateway that user have set. To do this, double-click *Device* in the device tree to open the *Communication Settings* tab of the device editor.

1. Click *Scan Network* to search the local network for all available devices.

   Result: The *Select Device* dialog appears with a list of all devices with which user can establish a connection.

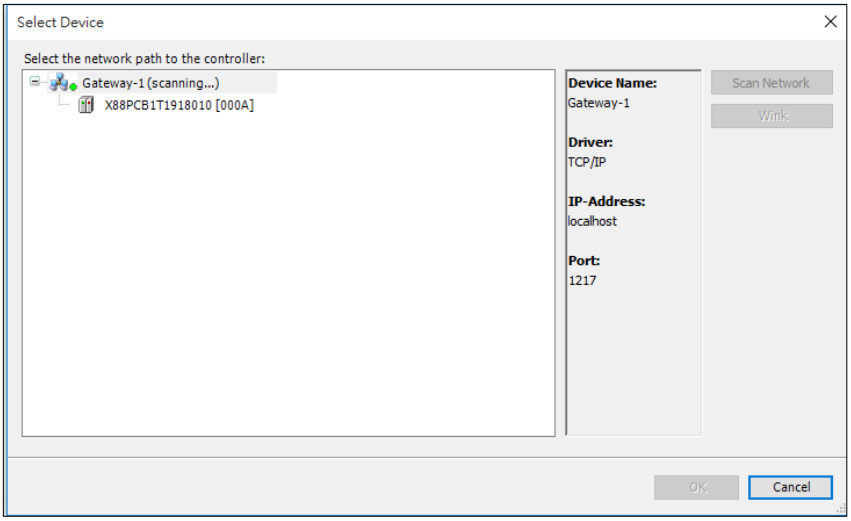


Figure 3- 15: Select network

2. Select the tree entry with the name of user controller.
3. Click *OK* button.

   Result: The channel is now active and the associated information appears below the device graphic on the *Communication Settings* tab.

NOTE: *All communication actions now refer exactly to this channel. Remember this later if user have multiple communication channels in the project.*

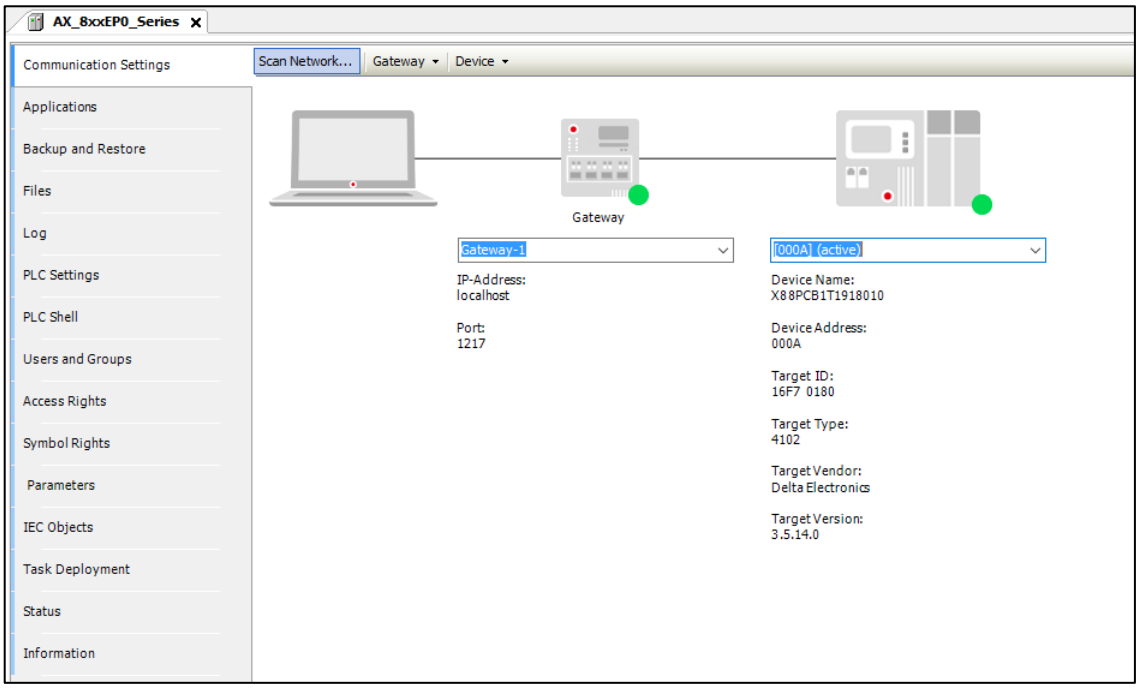


Figure 3- 16: Commnication channel active

3.3.1.4 Network Configuration

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

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

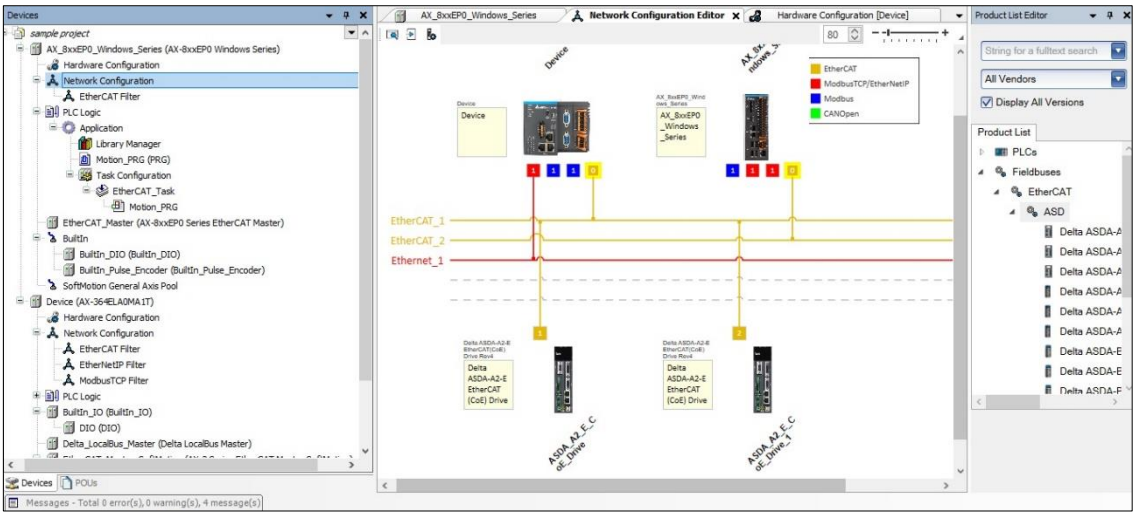


Figure 3- 17: Network Configuration

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.

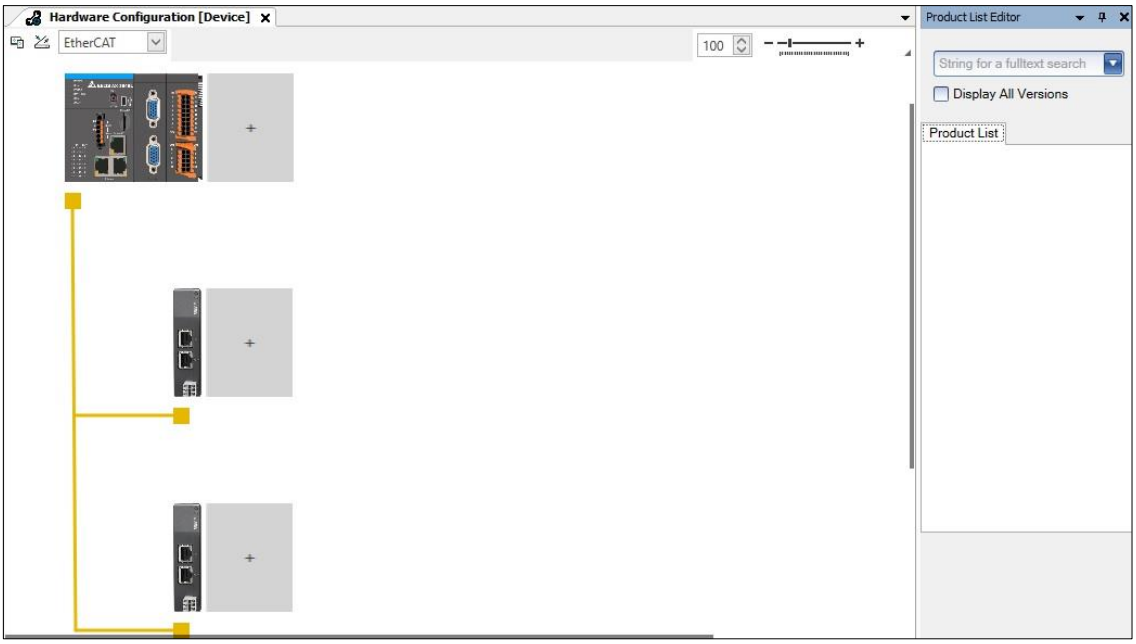


Figure 3- 18: Hardware configuration

3.3.1.6 Variable Declaration

First of all, declare the variables that user wish to use in the entire application. To do this, create a global variable list below *Application:*

1. Right-click *Application* entry and click *Add object* > *Global Variable List*.

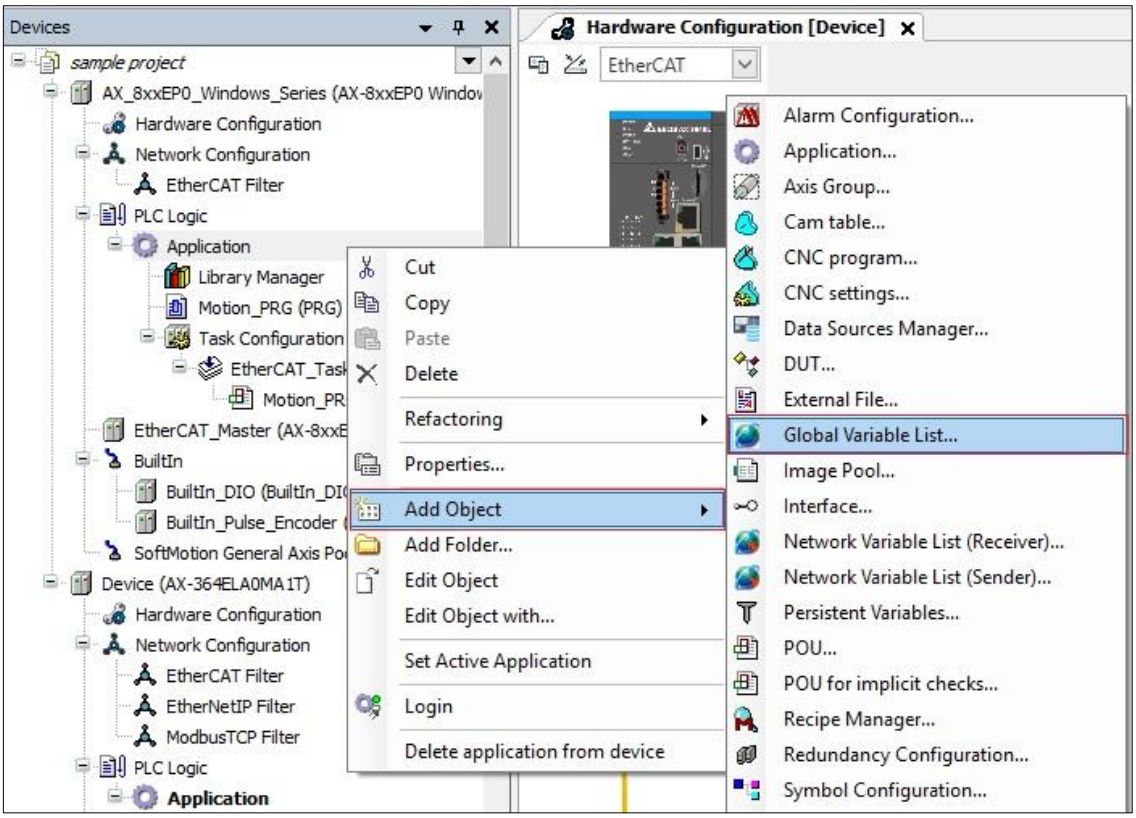


Figure 3- 19: New Global Variable List

2. The default global variable name is *GVL .*Click on *Add Object*> *Global Variable List.*

   Result: The *GVL* object appears below *Application*. The GVL Editor opens to the right of the device tree.

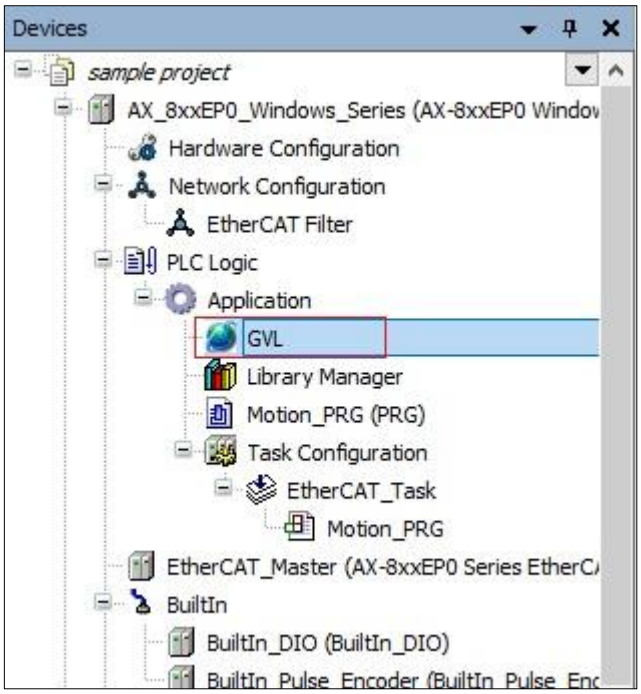


Figure 3- 20: Adding GVL

3. When the textual view appears, it already contains the keywords *VAR_GLOBAL* and *END_VAR*. For our example, activate the tabular view by clicking the  button in the right side bar of the editor. An empty row appears.

Figure 3- 21: GVL textual view

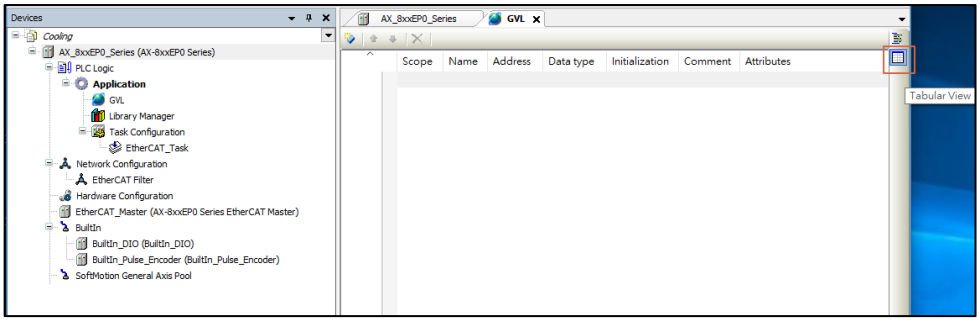


Figure 3- 22: GVL tabular view

4. Select  icon in the context menu.

   Result: An input field opens. At the same time, the Scope *VAR_GLOBAL* and the Data type *BOOL* are automatically entered in the row.

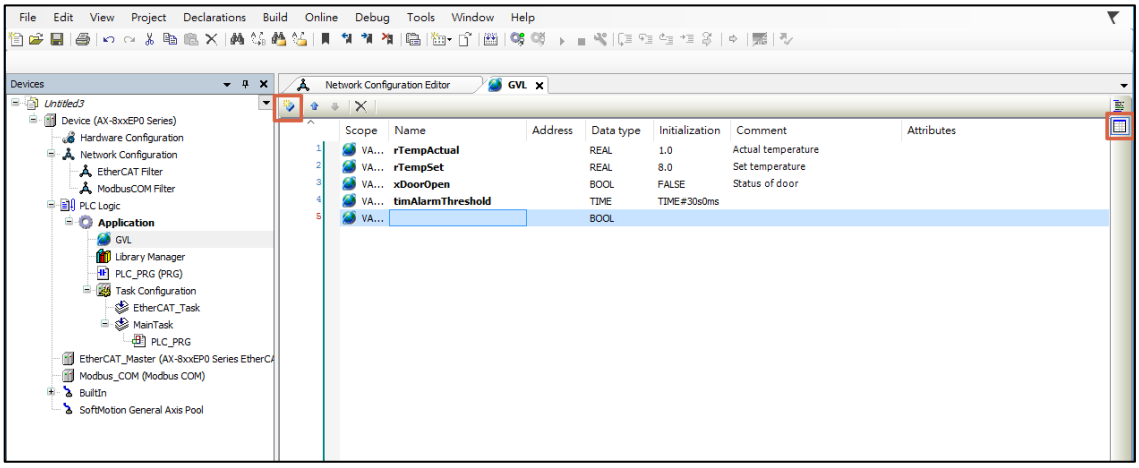


Figure 3- 23: **Inserting variable**

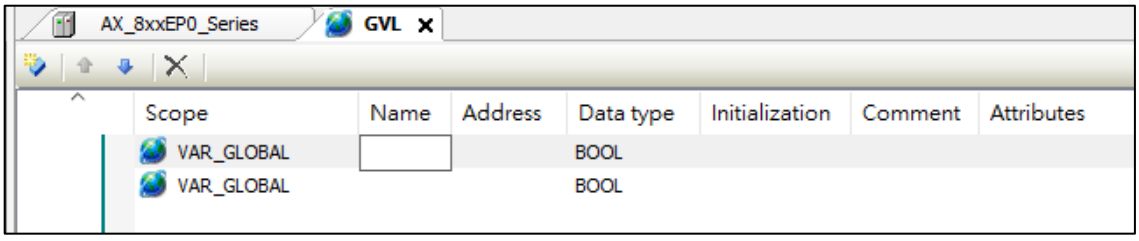


Figure 3- 24: Default GVL display

5. Type *rTempActual*" in the *Name* field.
6. Double-click the field in the *Data type* column (BOOL).

   Result: The field is now editable and the > button appears.

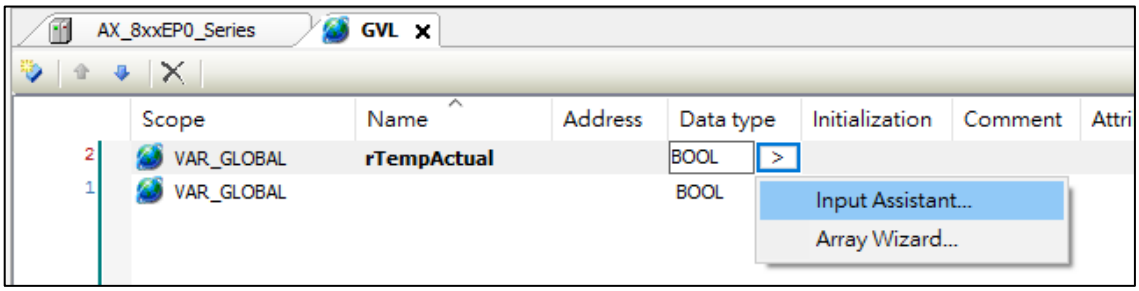


Figure 3- 25: Edit Data Type

7. Click for more settings (>) and then click *Input Assistant*.

   Result: The *Input Assistant* dialog opens.

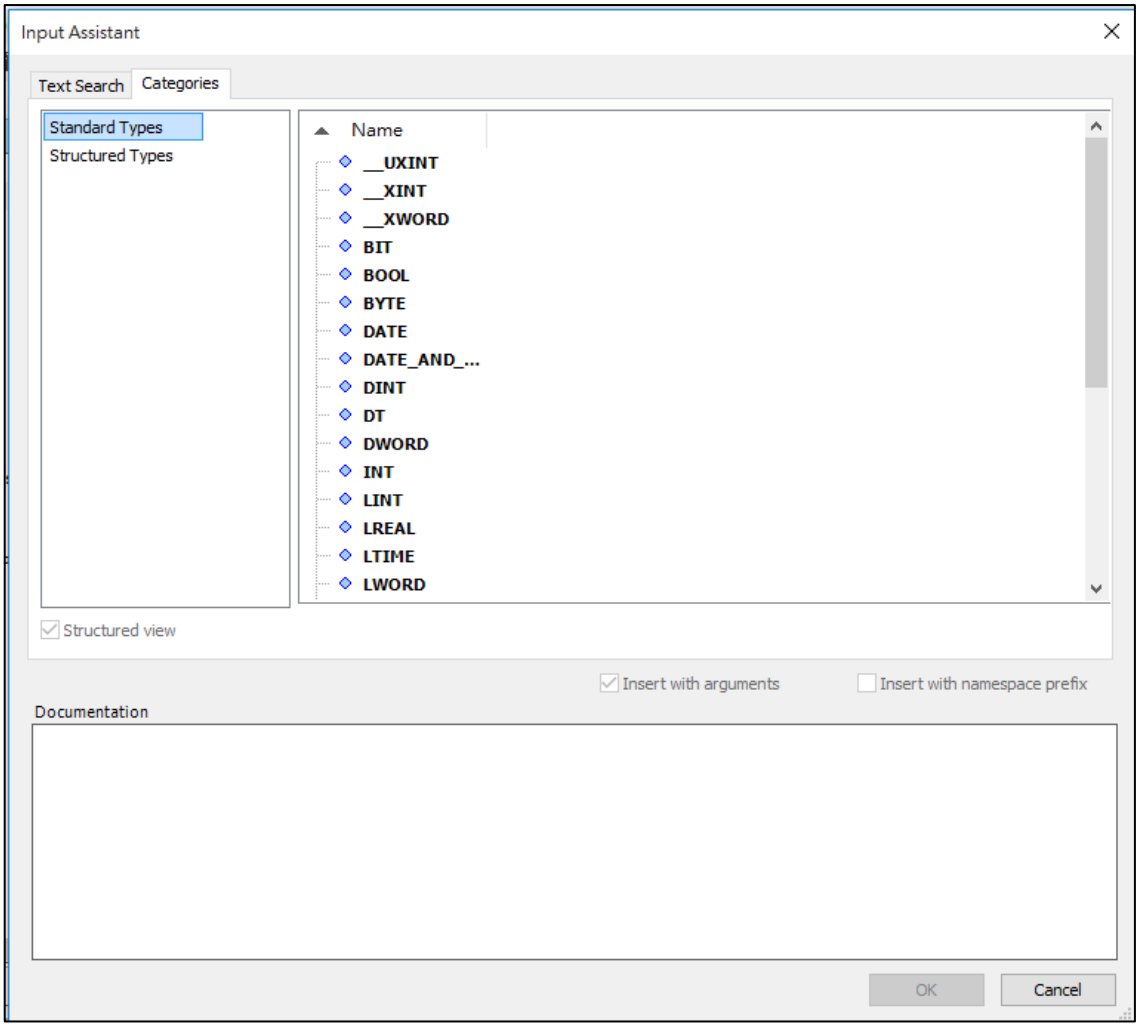


Figure 3- 26: Input Assistant

8. Select the data type REAL and click *OK*.
9. Enter a numerical value in the Initialization column (**Example**: 8.0).

Declare the following variables in the same way:

Name	Data Type	Initialization	Comment
rTempActual	REAL	1.0	Actual temperature
rTempSet	REAL	8.0	Set temperature

Name	Data Type	Initialization	Comment
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

The following figure shows the GVL tab after the settings are complete.

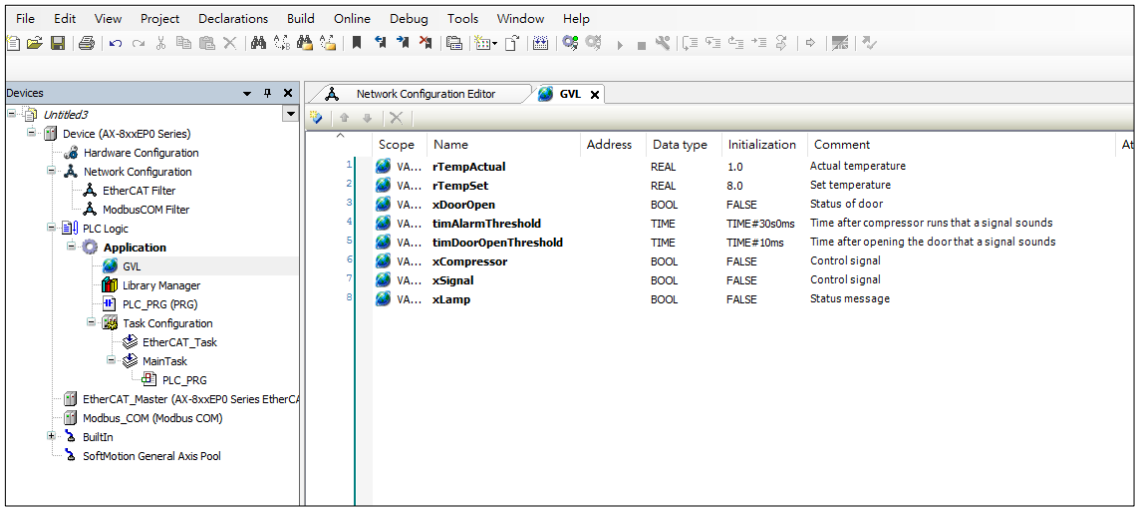


Figure 3- 27: Complete GVL settings

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.

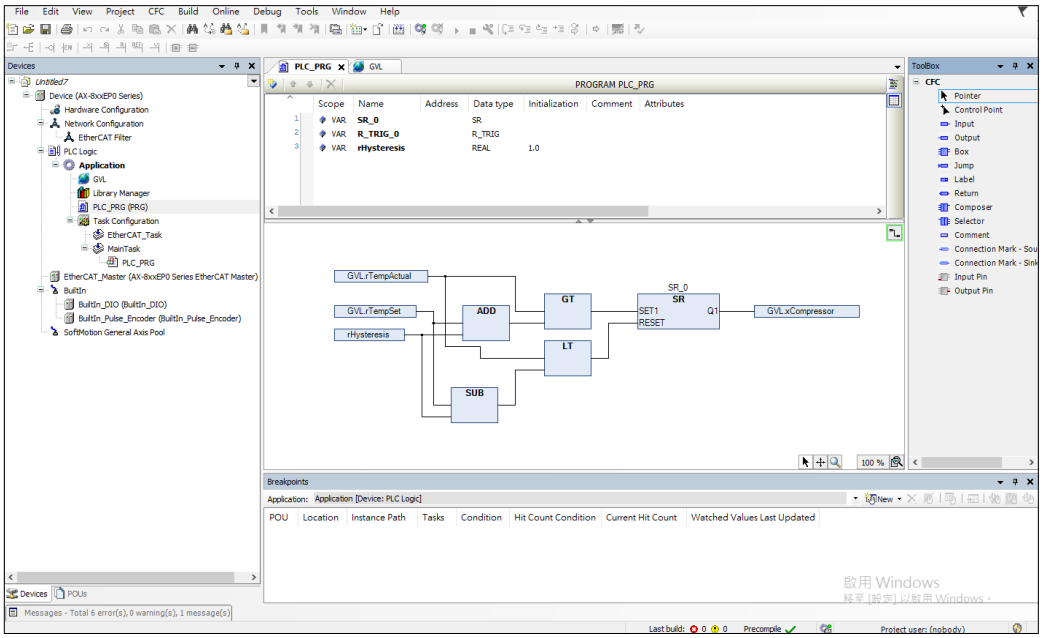


Figure 3- 28: PLC_PRG tab programming window

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 you 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 you 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.
15. 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.
16. 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.
17. 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.
18. 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.

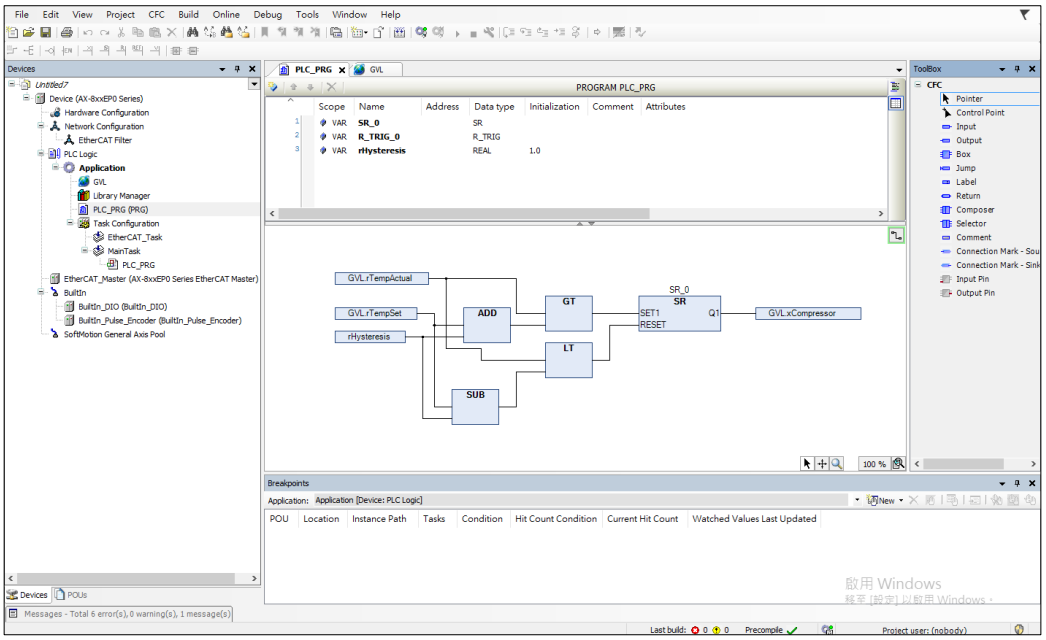


Figure 3- 29: Complete window of CFC implementation language to describe functionality

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.

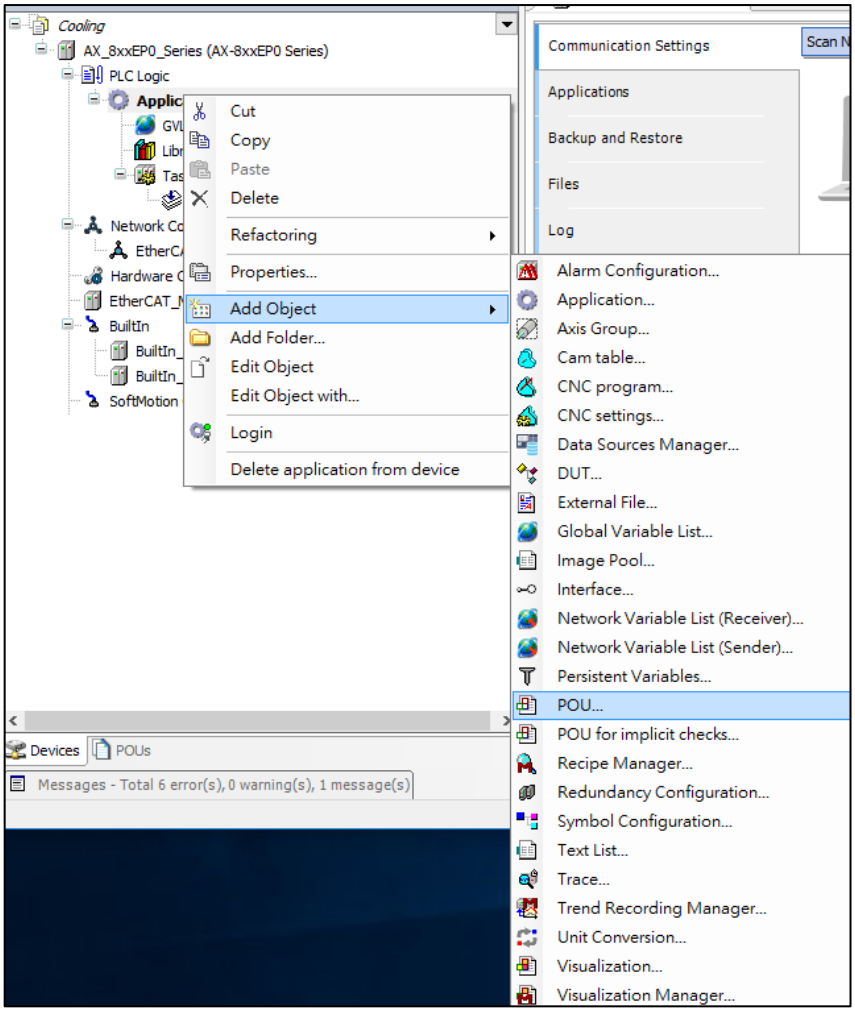


Figure 3- 30: Adding POU

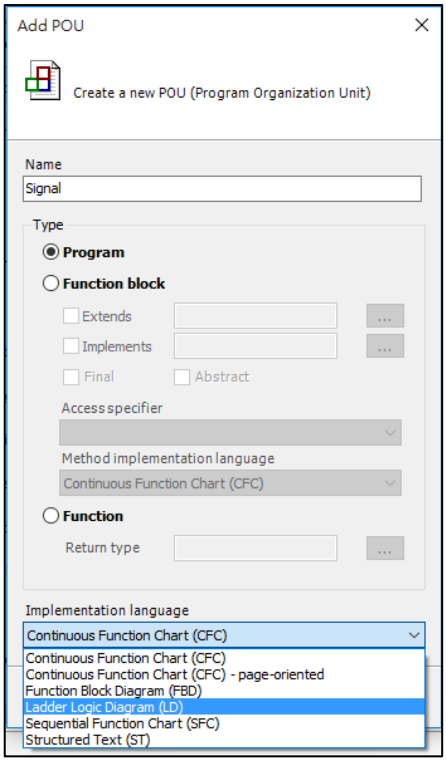


Figure 3- 31: Choosing implementation language

Result: Signals is listed in the device tree next to PLC_PRG**.** 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*).

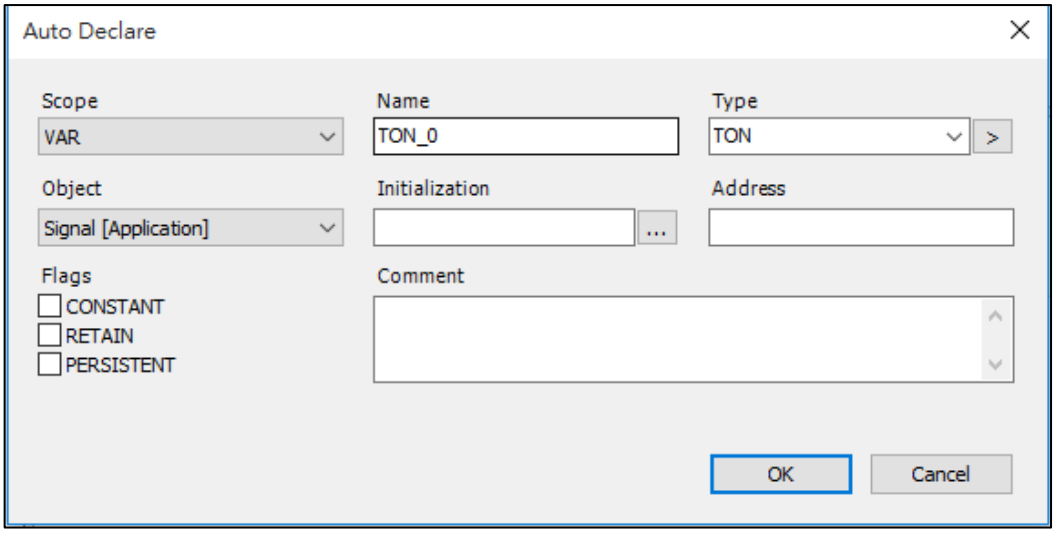


Figure 3- 32: Declare variable

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.

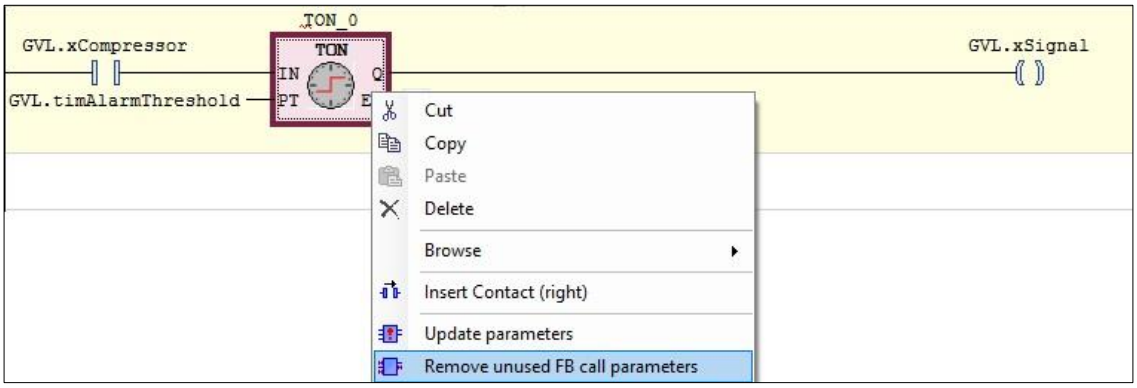


Figure 3- 33: Remove unused FB call parameters

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:

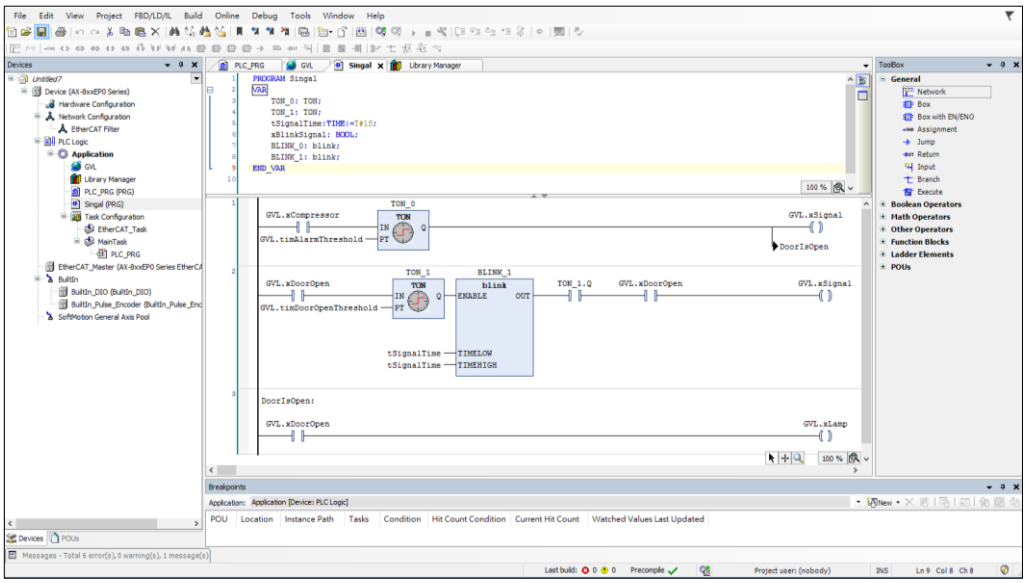


Figure 3- 34: LD program

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.

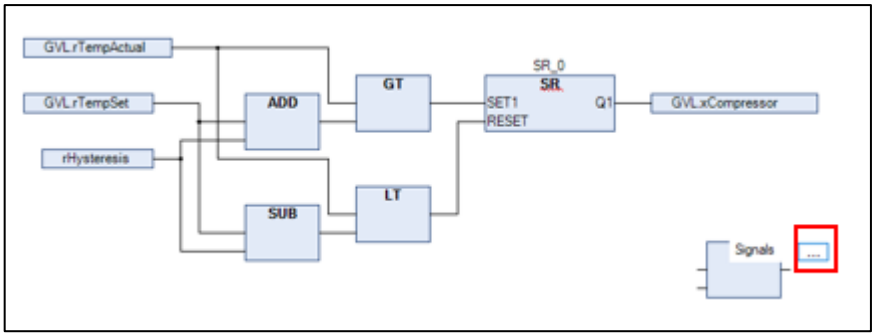


Figure 3- 35: Call signal

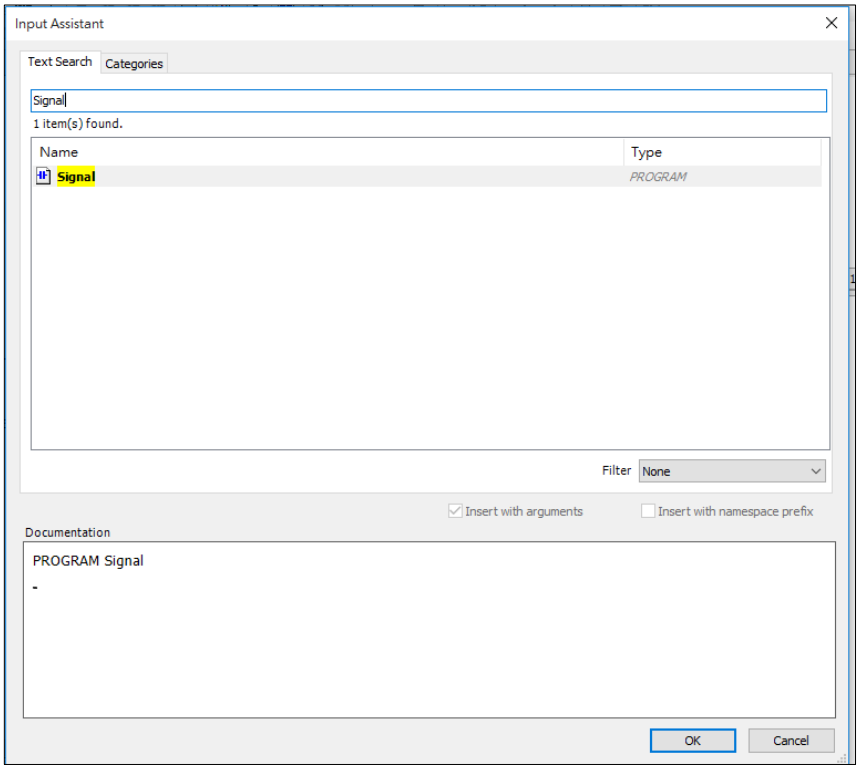


Figure 3- 36: Input assistant

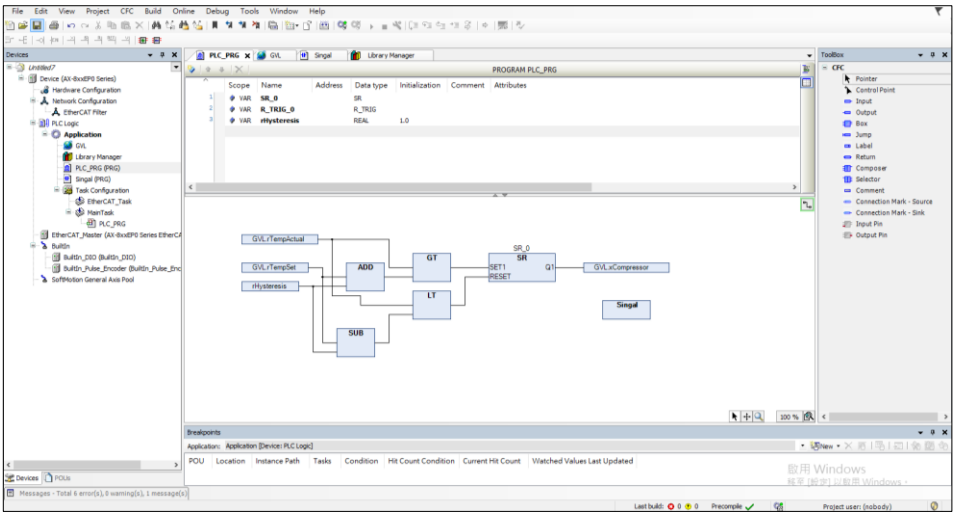


Figure 3- 37: Creating signal to LD

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 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.

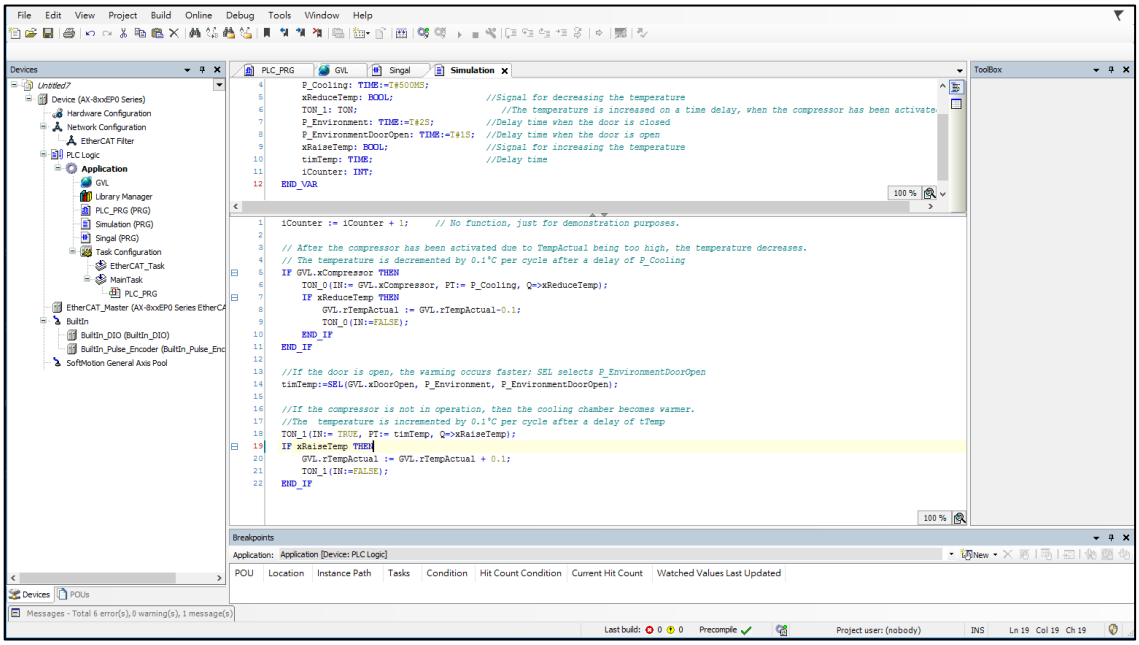


Figure 3- 38: Creating ST program

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.

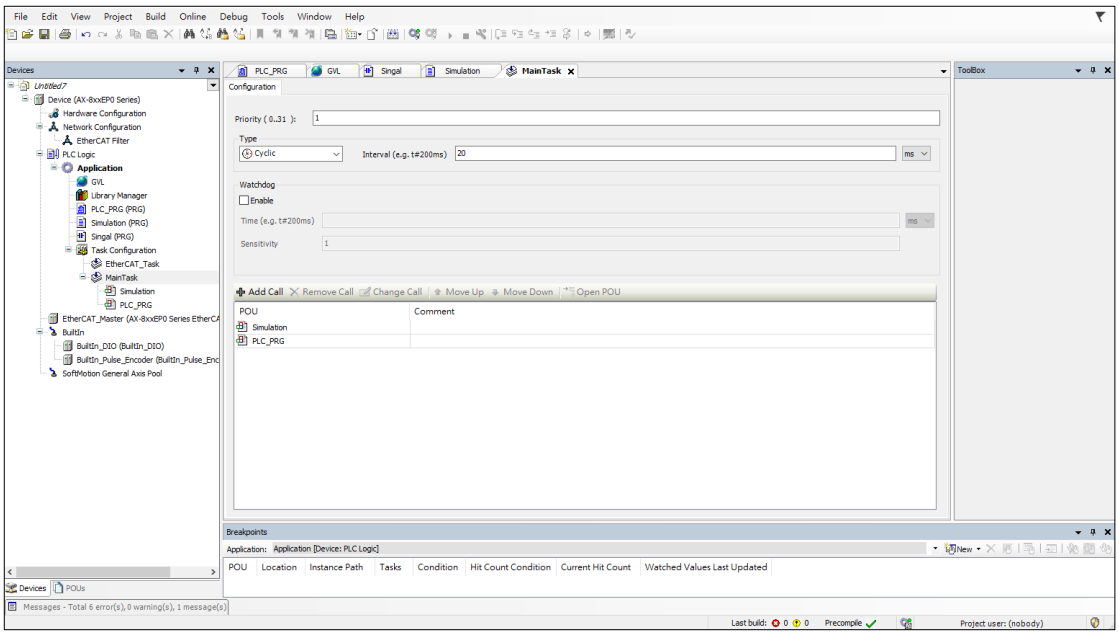


Figure 3- 39: Task configuration main task

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

Downloading the application to the PLC

***NOTE: Refer to** Debugging the application program. 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.

Figure 3- 40: Downloading application to PLC

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 status of the controller appears in the taskbar: STOP.

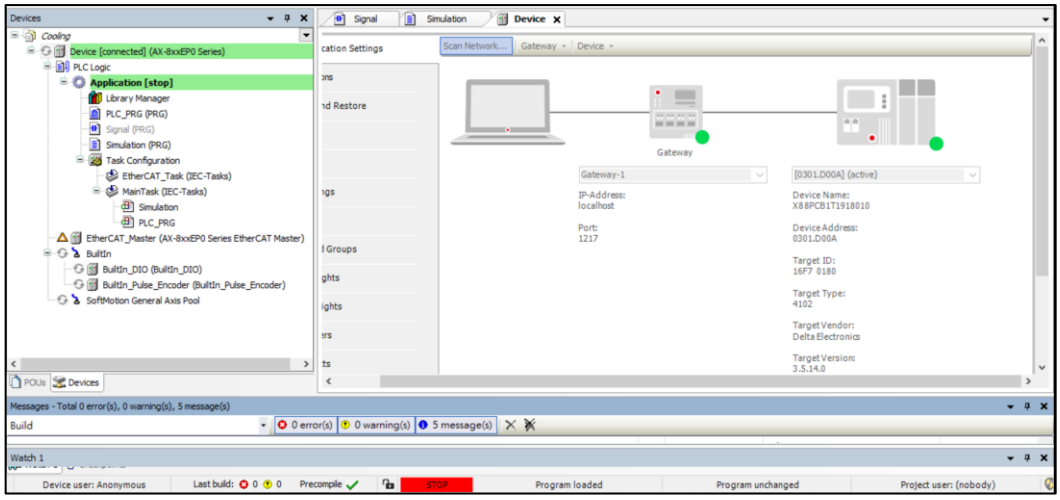


Figure 3- 41: Controller stop status

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.

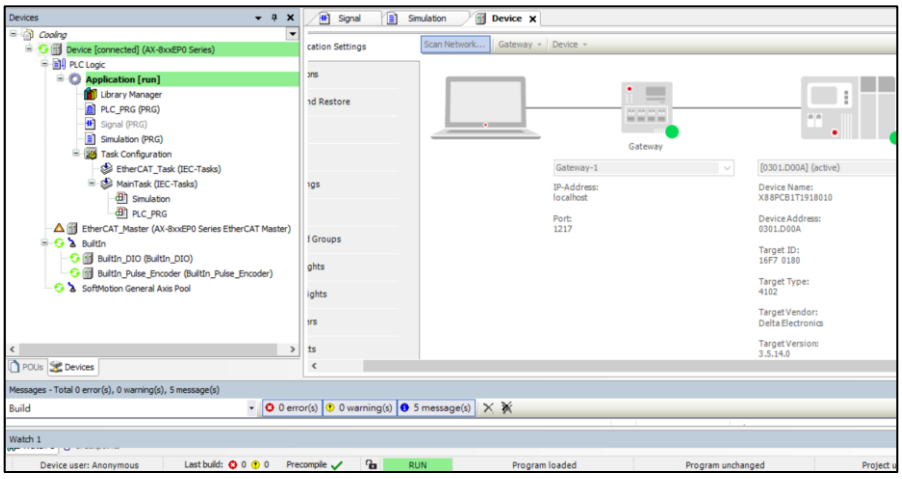


Figure 3- 42: Controller run status

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 *Chapter 6: HMI Interactive* for more information.

Chapter 4: Setting Device Hardware Configuration

4.1 Configuration

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

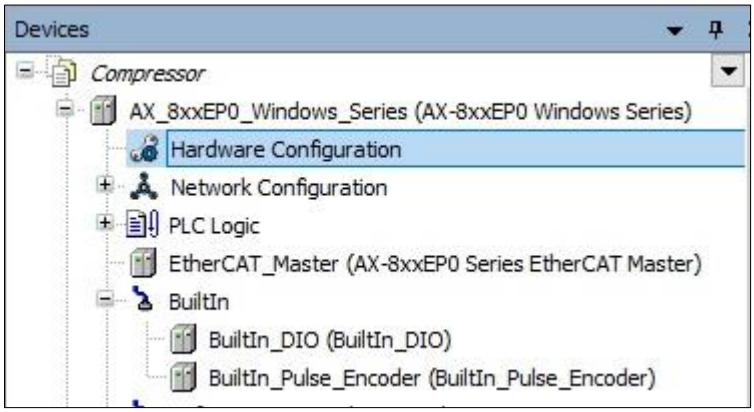


Figure 4 - 1: Hardware Configuration

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

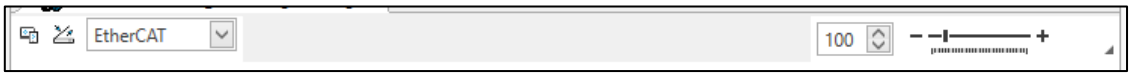


Figure 4 - 2: Hardware Configuration toolbar

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

Function	Description
	Click to display module information.
	Click to perform I/O 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.

4.1.1 Hardware Module Configuration

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

4.1.1.1 AX-3 Series Module Configuration:

This section applies to AX-308EA0MA1T, AX-308EA0MA1P, AX-364ELA0MA1T. Refer the following for example, AX-308EA0MA1T

The AX-308EA0MA1T PLC has a non-backplane design with expansion module placed only on the right side (+ position) of the device in *Hardware Configuration*.

Figure 4 - 3: Hardware Configuration of AX-308EA0MA1T

Follow the steps to create configuration for AX-308EA0MA1T:

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

   Result: The *Hardware Configuration* window displays with

   - A place holder to the right of the device to add I/O modules.
   - A *Product List Editor* pane to the right of the *Hardware Configuration* window.

Figure 4 - 4: Hardware Configuration device

2. Click on the CPU expansion placeholder.

   Result: *Product List* displays the list of I/O modules.

Figure 4 - 5: Product list

3. Expand *Product List* and select the required I/O module. Either double-click on the modules or drag and drop the modules to the placeholder.

   Result: The modules are added to the right side of the device as shown in the following figure.

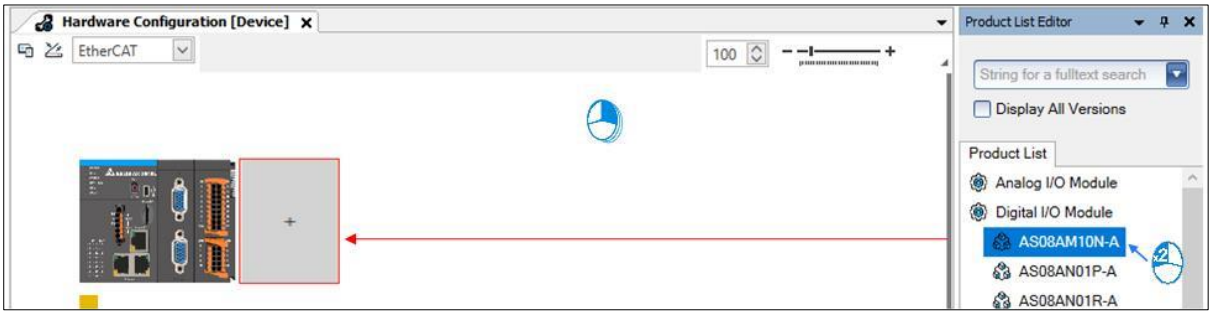


Figure 4 - 6: Adding I/O Module

***NOTE**: Click on the required IO module. The information about the particular module is displayed below the Product List pane.*

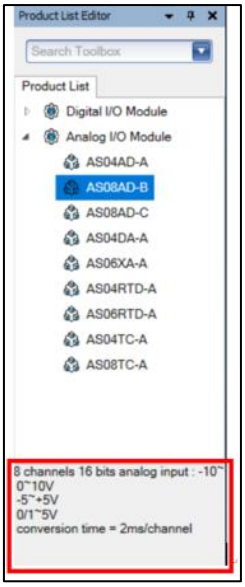


Figure 4 - 7: IO module information

4. Double-click the *Network Configuration* on the device tree to add RIO Module.

   ***NOTE**: AX-300NA0PA1 and AX-324NA0PA1P do not support adding RIO modules.*

   Result: The *Network Configuration Editor* is as shown below.

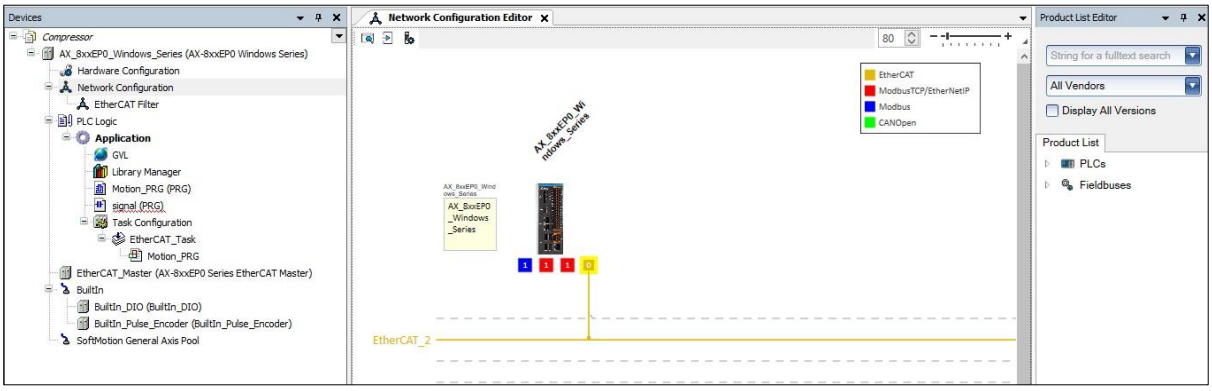


Figure 4 - 8: Add RIO module from network configuration

5. Expand the *Product List* on the right side of the screen. *Fieldbus*> *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*.

Result: Network Configuration Editor is shown in the figure below:

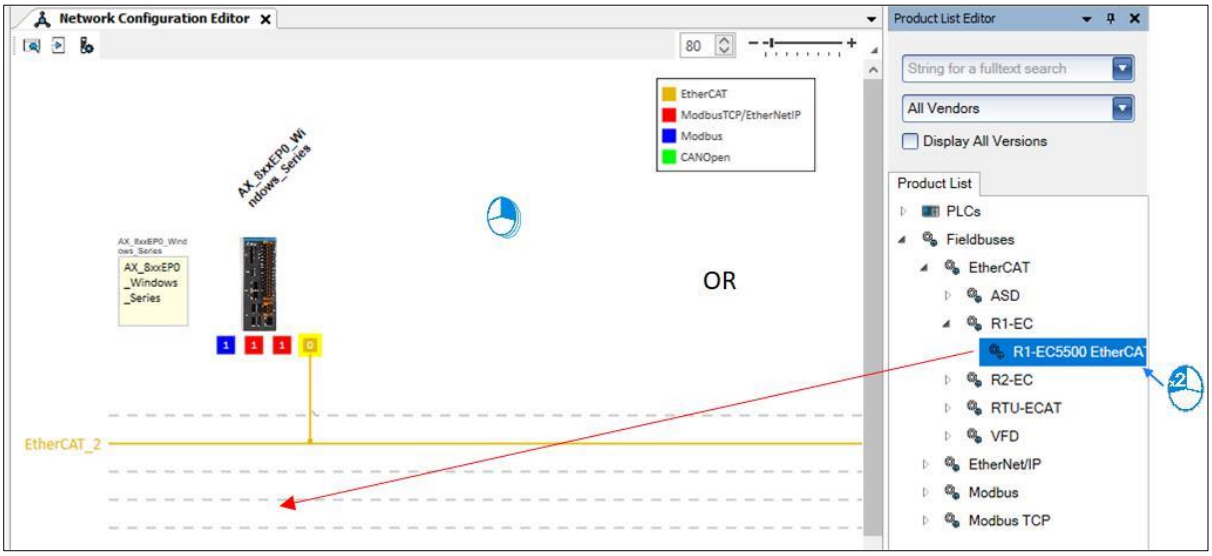


Figure 4 - 9: Add R1-EC EC5500 to the network configuration Editor

6. Connect this R1-EC5500 to the same yellow EtherCAT line as the master station.

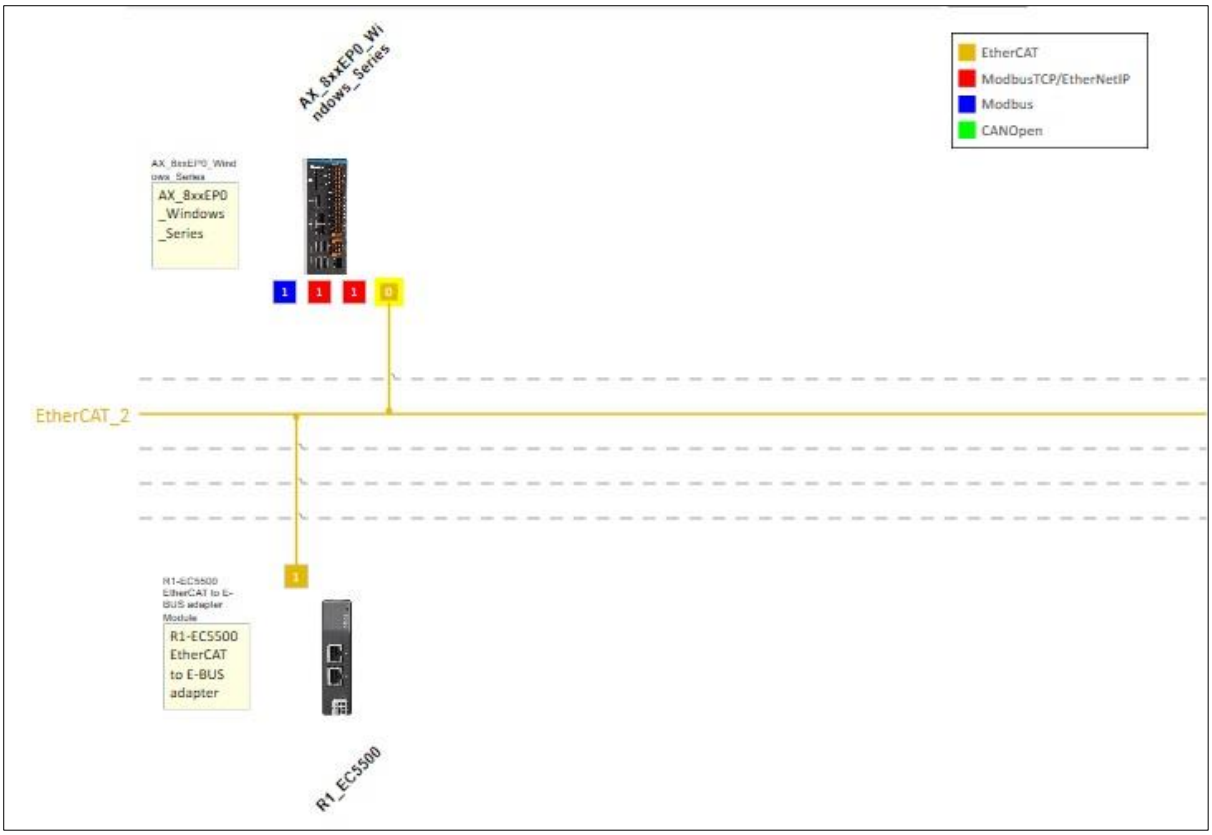


Figure 4 - 10: R1-EC5500 is connected to the main station EtherCAT line

7. Go back to the AX-308EA0MA1T *Hardware Configuration* screen and you can see the RIO Coupler module is displayed under the I/O module. Same as I/O modules, users can click the reserved position (+) on the right to add RIO modules, as shown in the figure below.

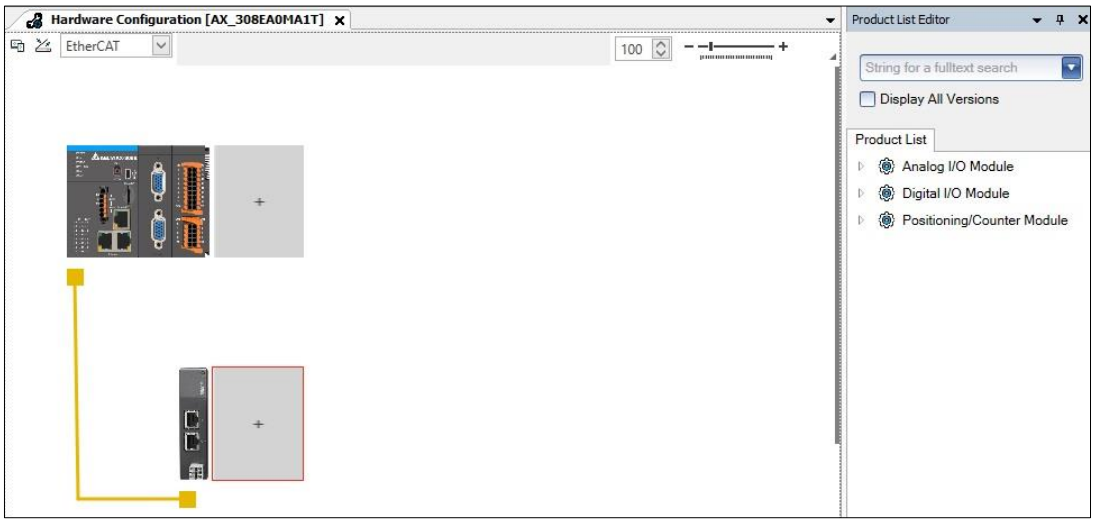


Figure 4 - 11: Add RIO Module

Result: In the AX-3 series hardware configuration, up to 64 RIOs can be added.

Device Group Menu

The device group menu provides the option to cut, copy, paste, and 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

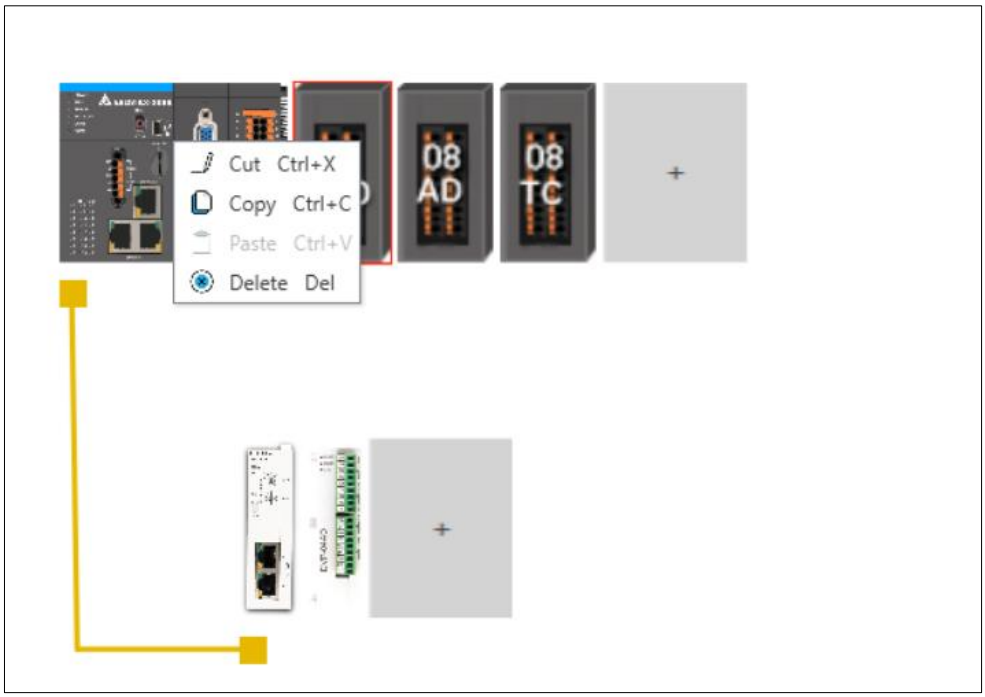


Figure 4 - 12: Device Context Menu - AX-308EA0MA1T

4.1.1.2 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 *Hardware Configuration* in the case of AX-8xxEP0 series.

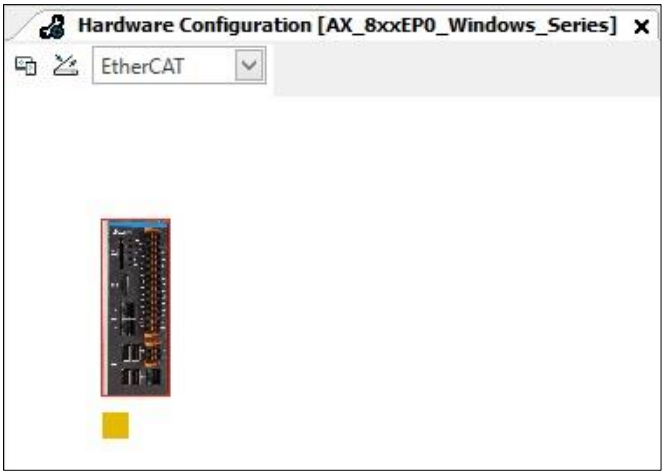


Figure 4 - 13: Device Context Menu - AX-8xxEP0 series

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.

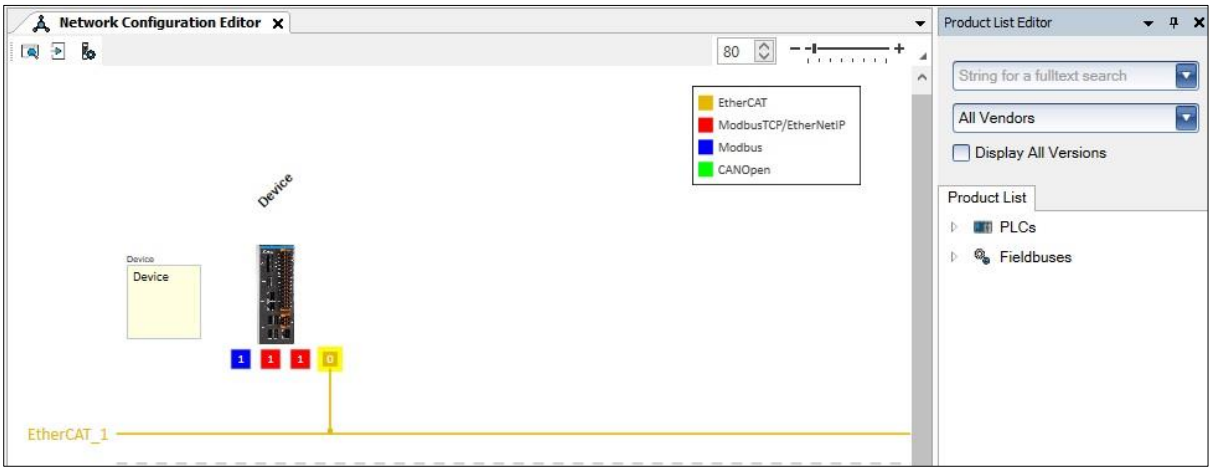


Figure 4 - 14: Network Configuration Window

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.*

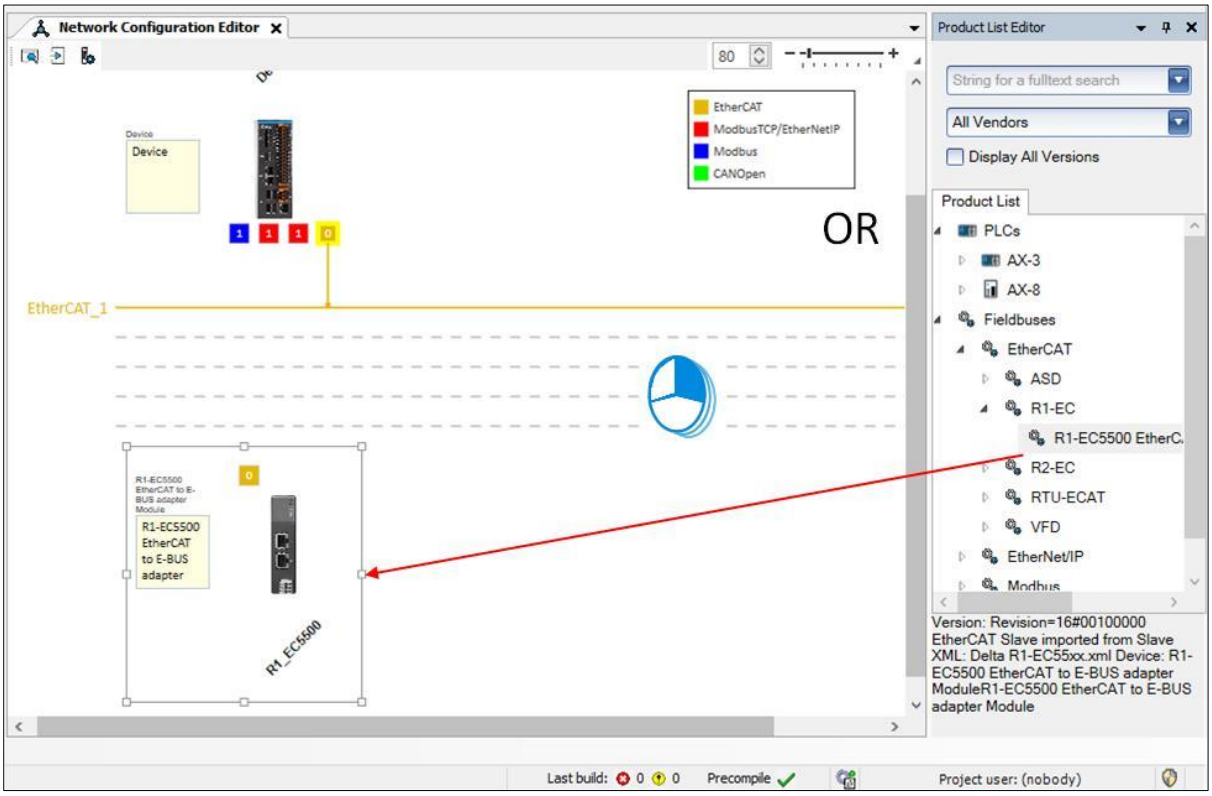


Figure 4 - 15: Add R1-EC5500 to the Network Configuration Window

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

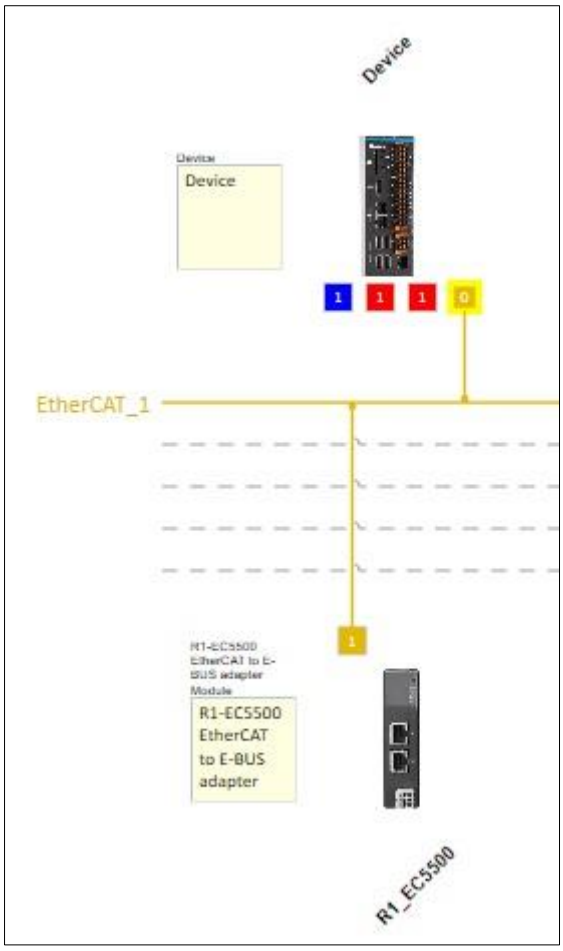


Figure 4 - 16: R1-EC5500 is connected to the yellow EtherCAT line

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

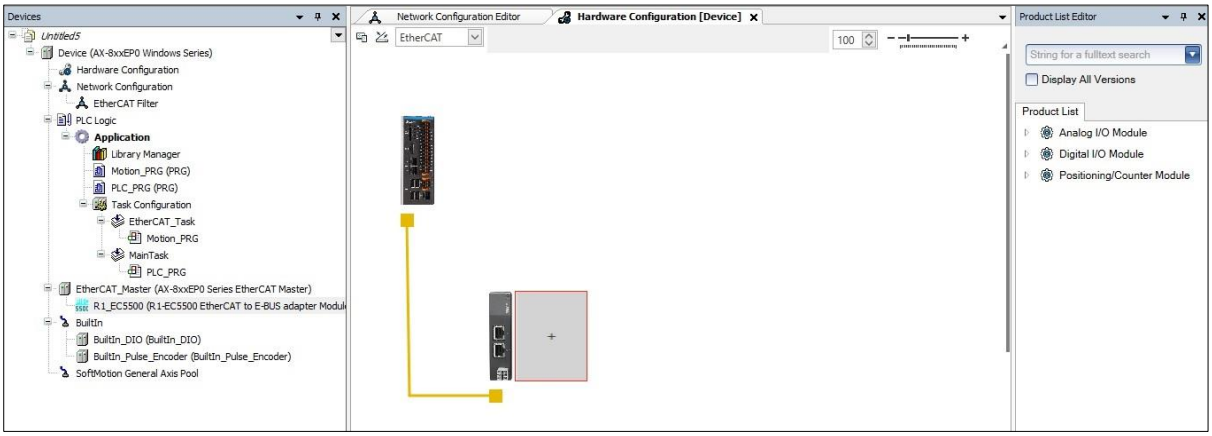


Figure 4 - 17: Add RIO Module

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

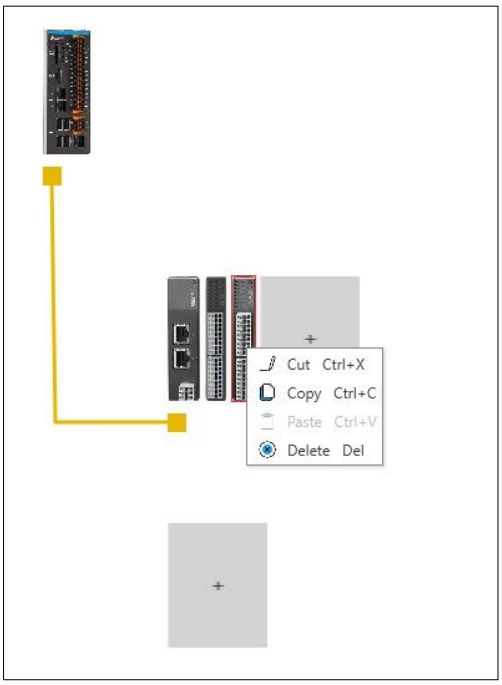


Figure 4 - 18: Device Context Menu - AX-8xxEP0 series

4.1.2 Hardware Firmware Update

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 hardware needs to be updated. The firmware of the hardware can be updated through the firmware update function supported by DIADesigner-AX version.

Hardware series currently supporting firmware update:

- AX-308EA0MA1T
- AX-308EA0MA1P
- AX-364ELA0MA1T
- AX-300NA0PA1
- AX-324NA0PA1P

Please follow the steps below to perform the firmware update:

1. On the menu bar, click *Tools* > *Firmware Update*.

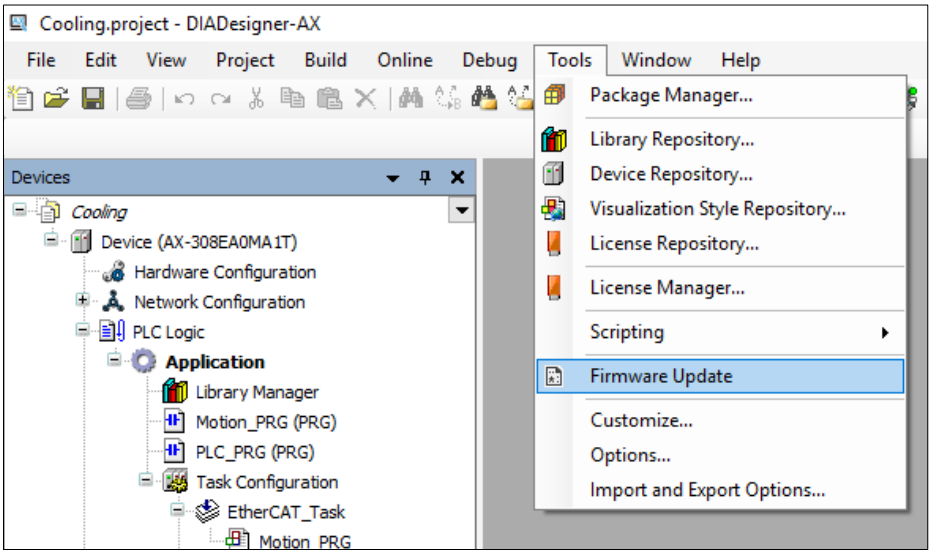


Figure 4 - 19: Firmware update path

Result: The *firmware Package updater* window pops-up.

2. Select the firmware package to be updated in *Path*.
3. Select the updated device from the Online Device.

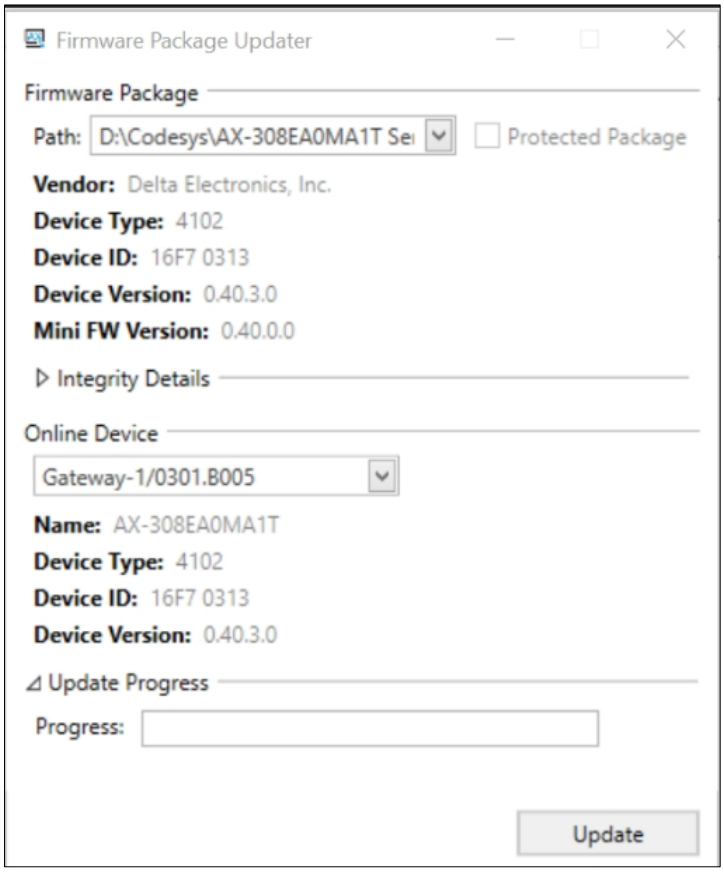


Figure 4 - 20: Firmware update window

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

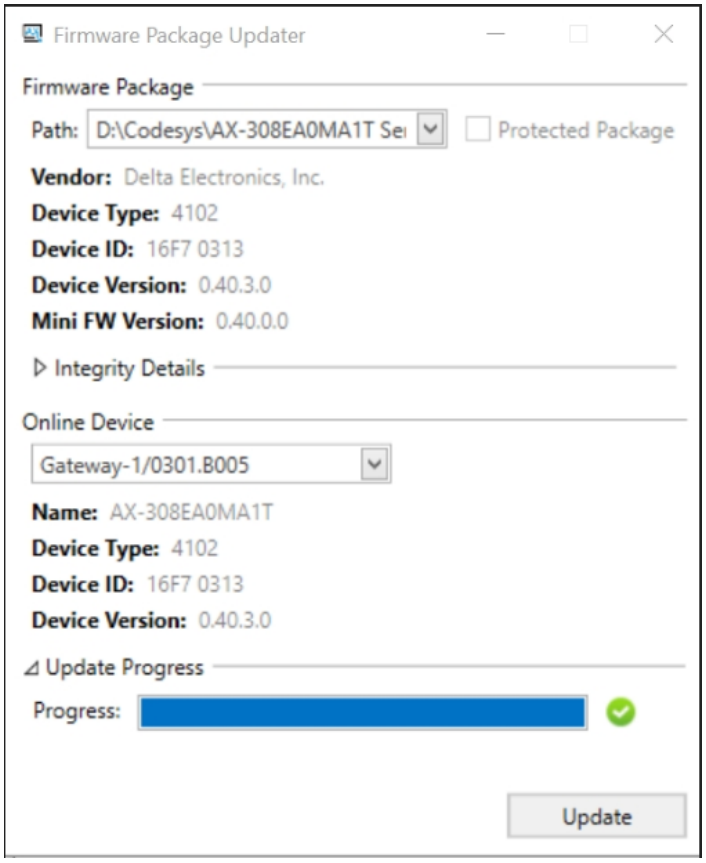


Figure 4 - 21: Firmware update progress

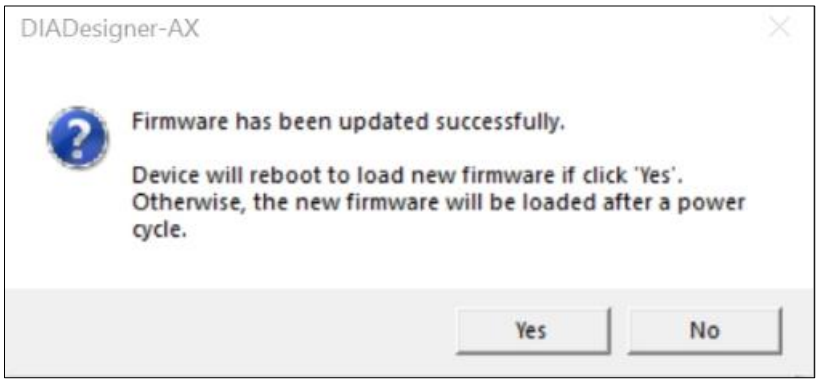


Figure 4 - 22: Successful firmware update

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.
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 select device window
Name	Controller name
Device Type	Controller device type number
Device ID	Controller device ID
Device Version	Controller device version number
Update progress	
Progress	After clicking update, the update progress will be displayed.

4.2 AX-3 Series Parameter 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-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.

4.2.1 System Parameter Tab

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.

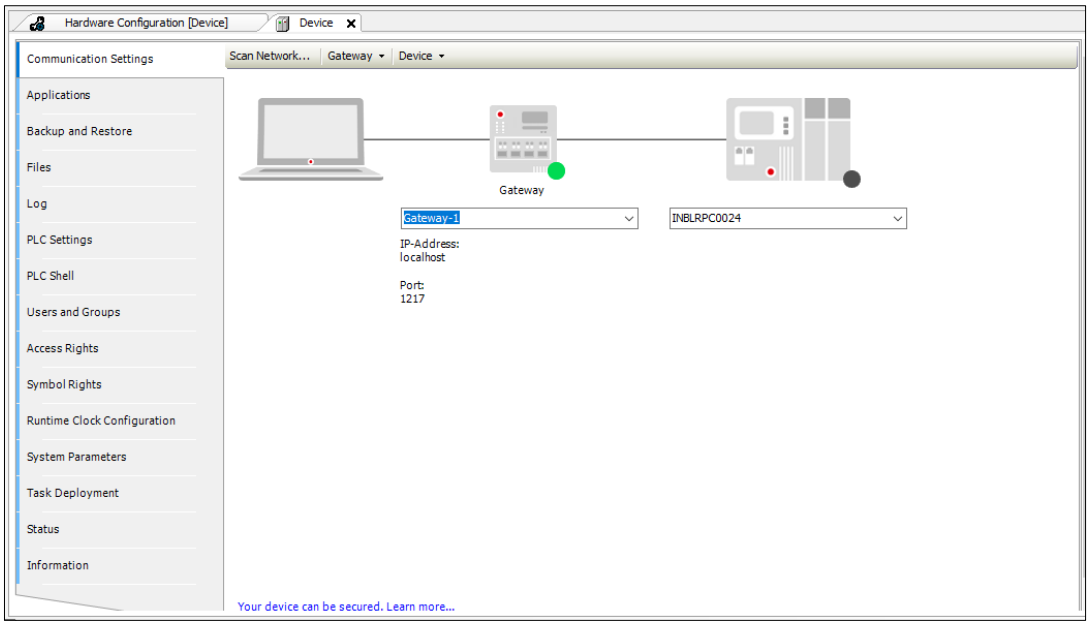


Figure 4 - 23: Communication Setting

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

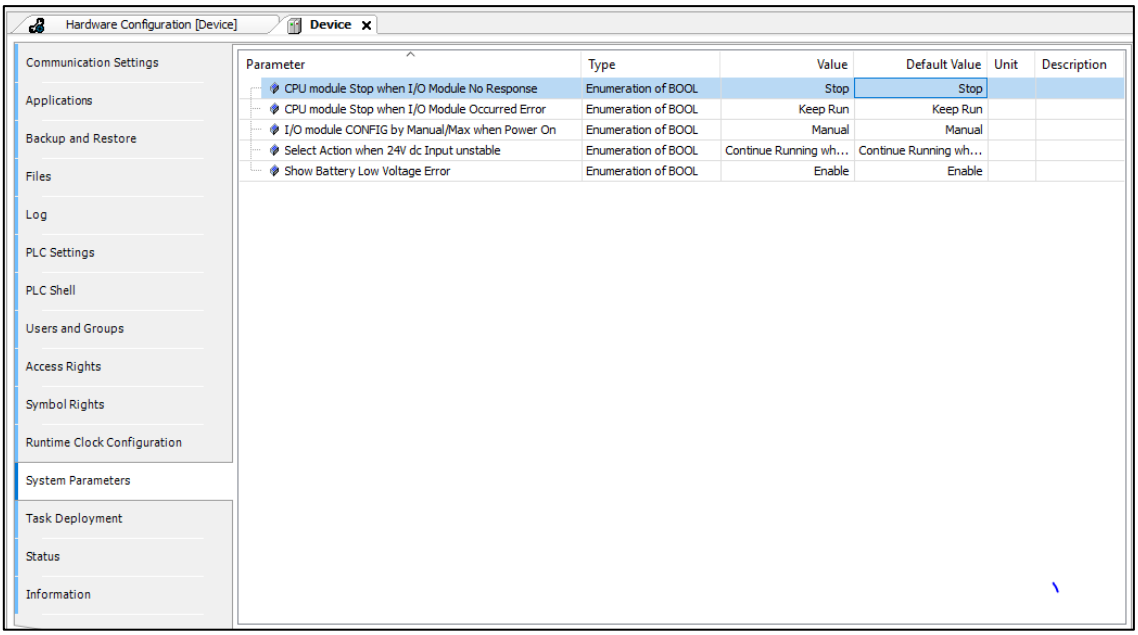


Figure 4 - 24: System Parameters

The *System Parameters* tab displays the following parameters:

Item	Description
Parameter	Parameter name, not editable
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.

4.2.2 System Settings Tab (Runtime Clock & IP Address Setting)

Double-click on the *AX-308EA0MA1T* series to open the System Settings Tab (Runtime Cock & IP Address Setting) to set the controller clock.

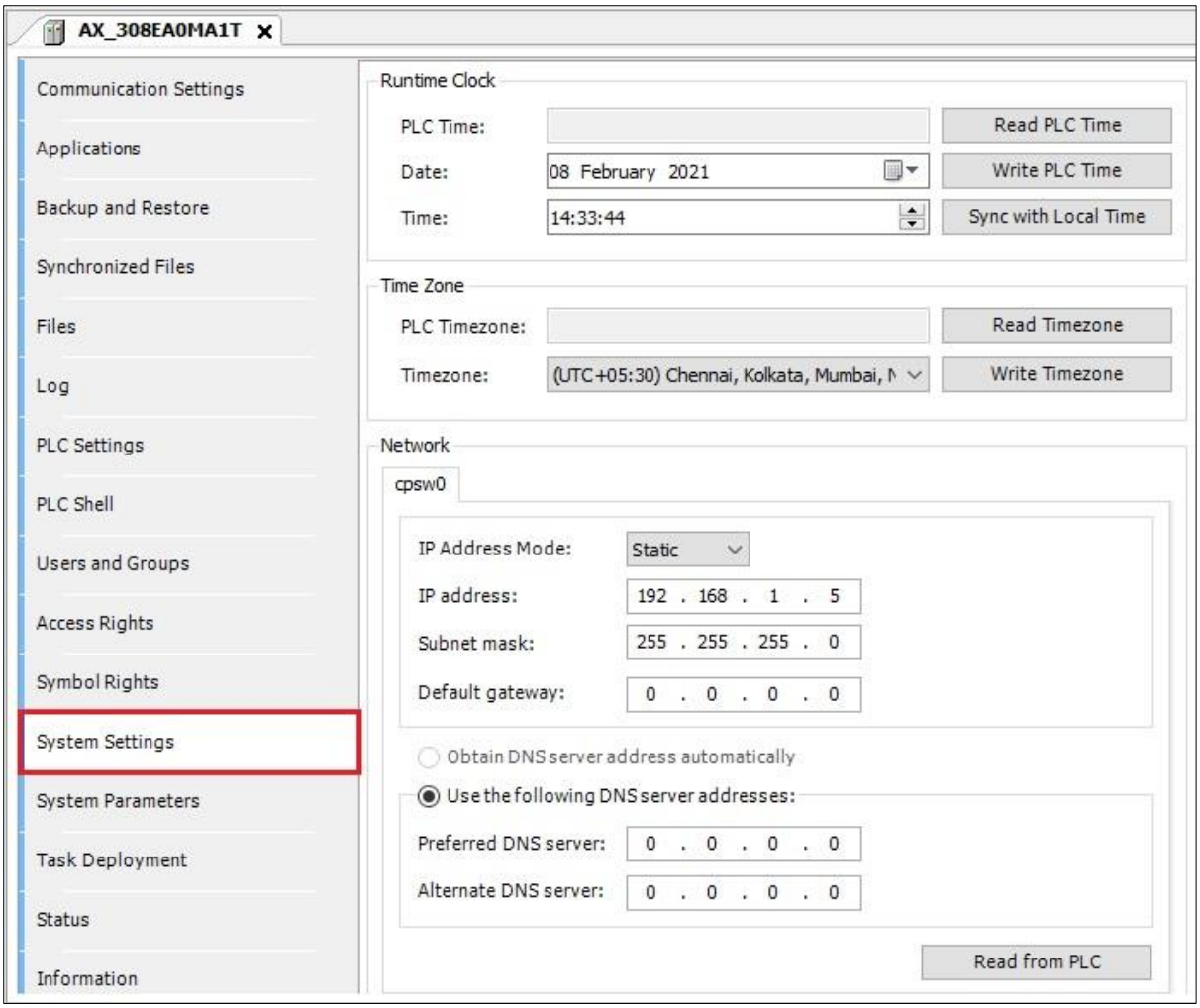


Figure 4 - 25: System Settings Tab

- Runtime Clock:

Item	Description
PLC Time	Click *Read PLC Time* to read time information from PLC
Date	Set Date YYYY MM DD and click *Write PLC Time* to write the Date and time to PLC
Time	Set Time and also can sync with local time

- Time Zone:

Item	Description
PLC Time zone	Read PLC time zone information form PLC
Timezone	Set new time zone and write back to PLC

- Network:

It Contains AX-3 series device network IP address settings, user can also click to read PLC settings to update device IP address information.

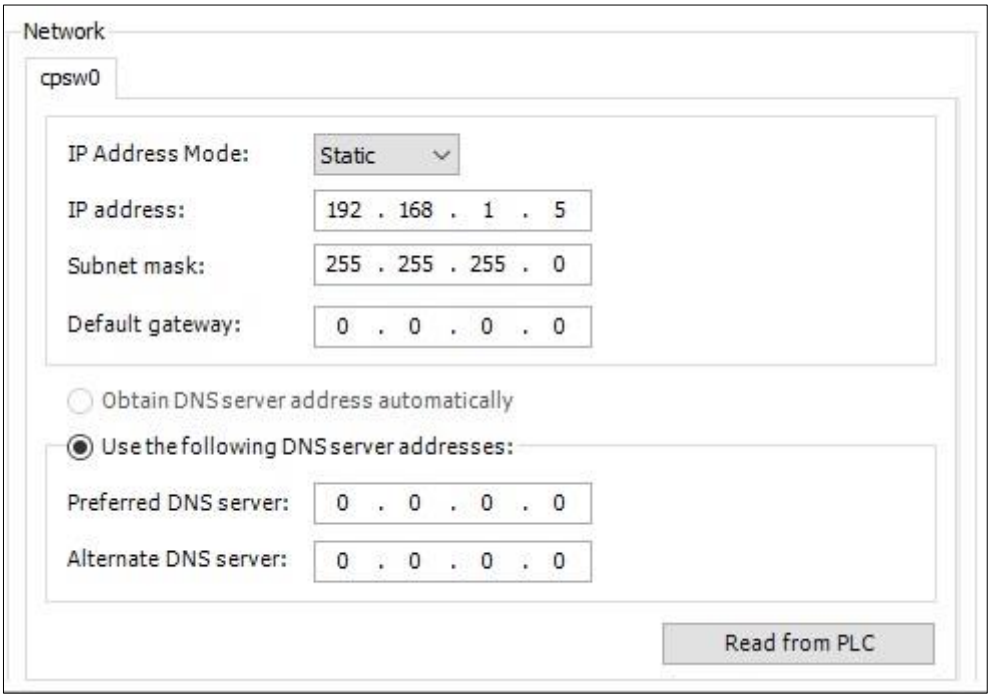


Figure 4 - 26: Netwrok Settings

4.2.3 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.

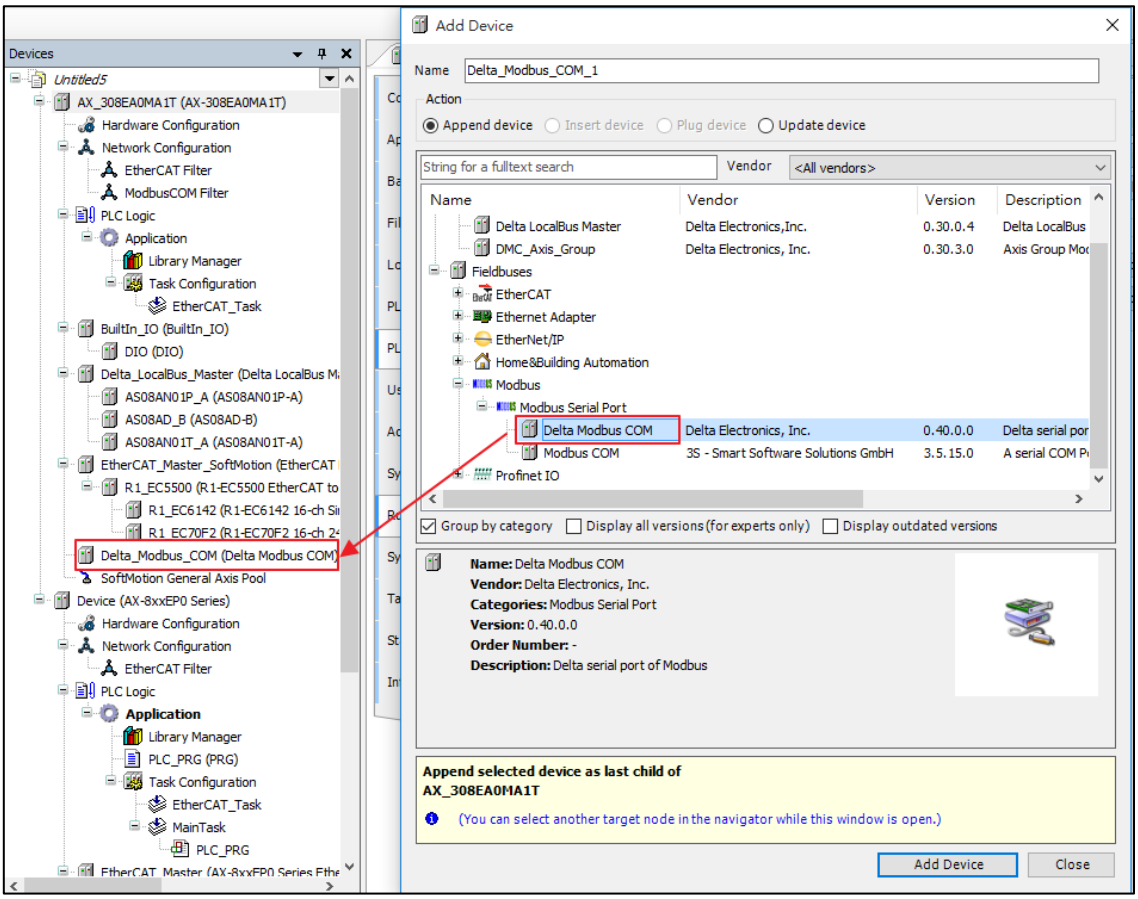


Figure 4 - 27: Setting Delta_MODBUS_COM

4.2.3.1 Tab 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**

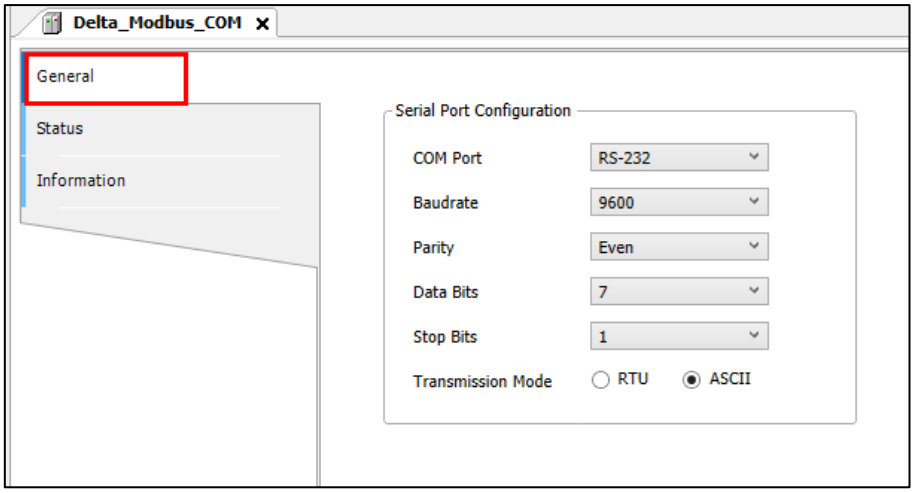


Figure 4 - 28: Delta_MODBUS_COM – General tab

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

Name	Description	
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
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.

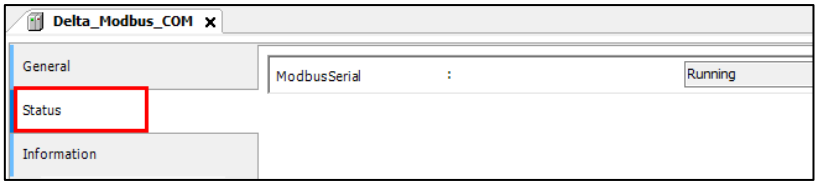


Figure 4 - 29: Delta_MODBUS_COM – Status tab

- **Information**

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

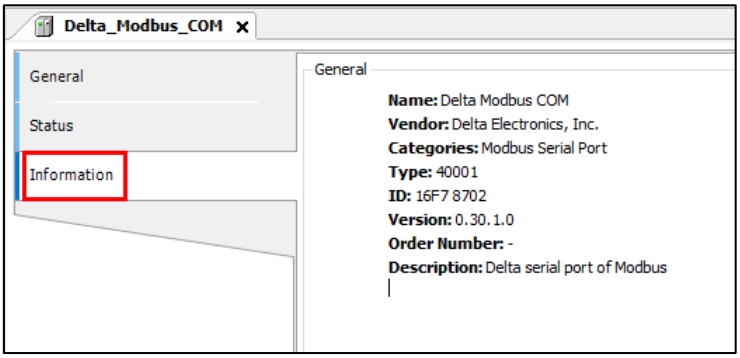


Figure 4 - 30: Delta_MODBUS_COM - Information

4.2.3.2 Tab 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.

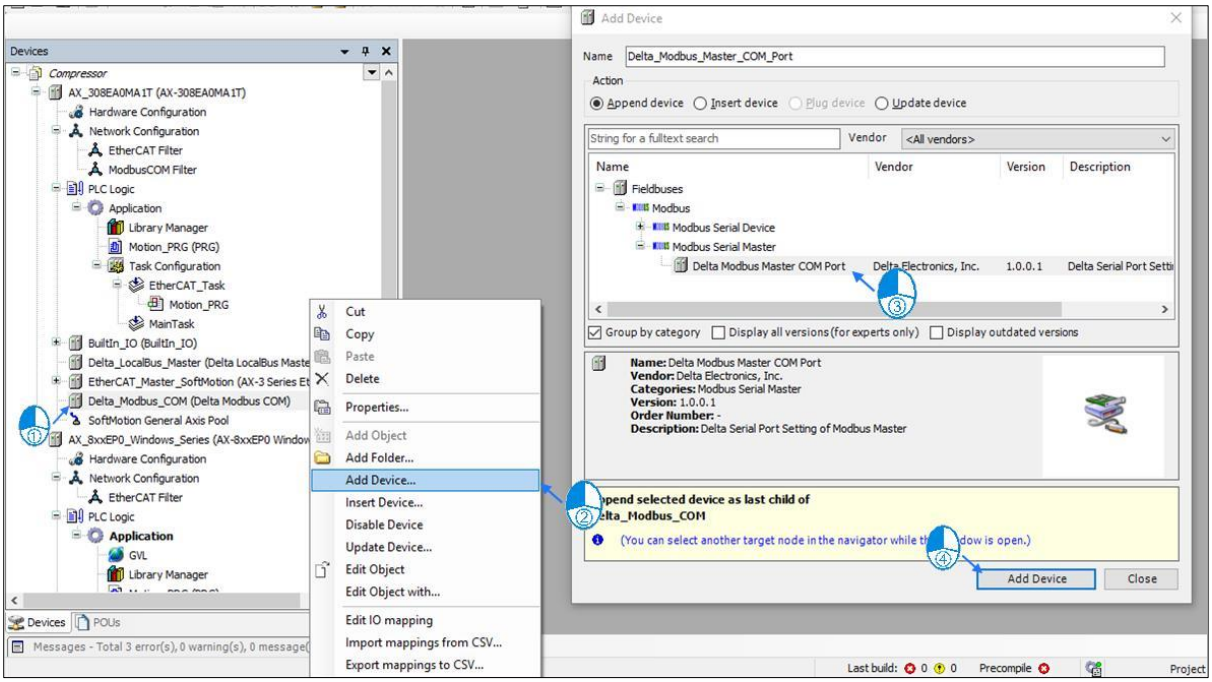


Figure 4 - 31: Adding Delta MODBUS Master COM Port to project tree

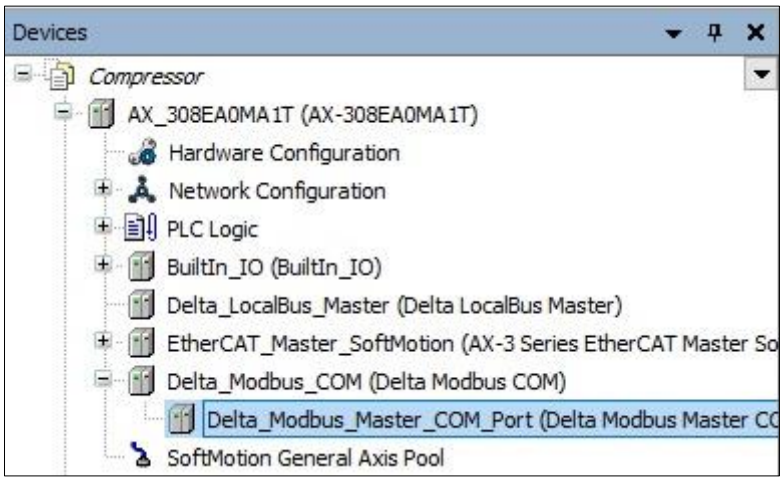


Figure 4 - 32: Adding Device

- **General**:

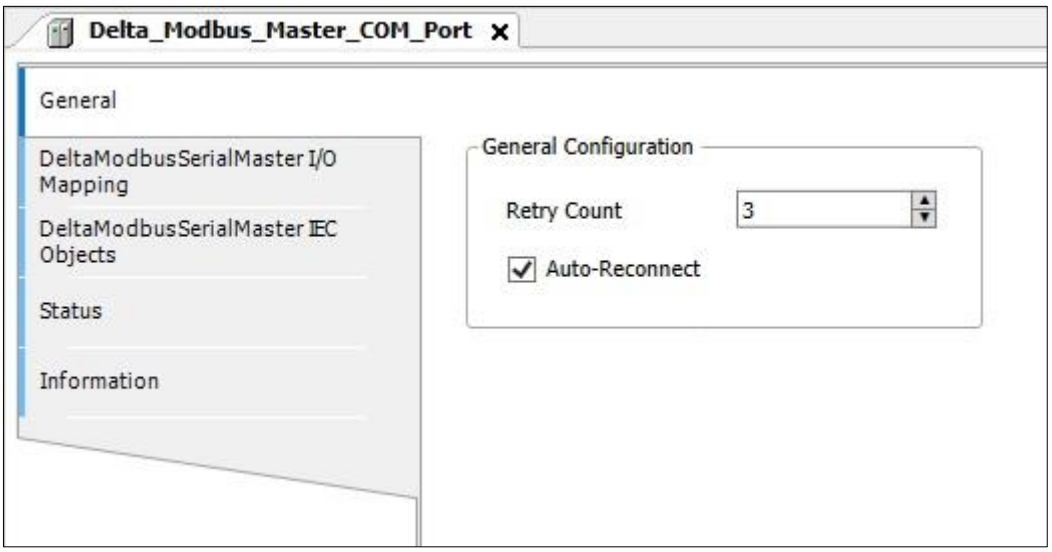


Figure 4 - 33: Delta_MODBUS_Master_COM_Port

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

Figure 4 - 34: MODBUS Serial Master I/O Mapping

Name	Description
MODBUS Serial Master I/O Mapping	User can set Bus Cycle Task of MODBUS Serial Master. Please refer to the Hardware manual with PLC Settings.
Status	MODBUS Serial Master operation status.
Information	Show the current Device information, including Device ID, type, and Vendor…information.

***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.*

4.2.3.3 Tab Delta MODBUS Slave COM port – General

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*.

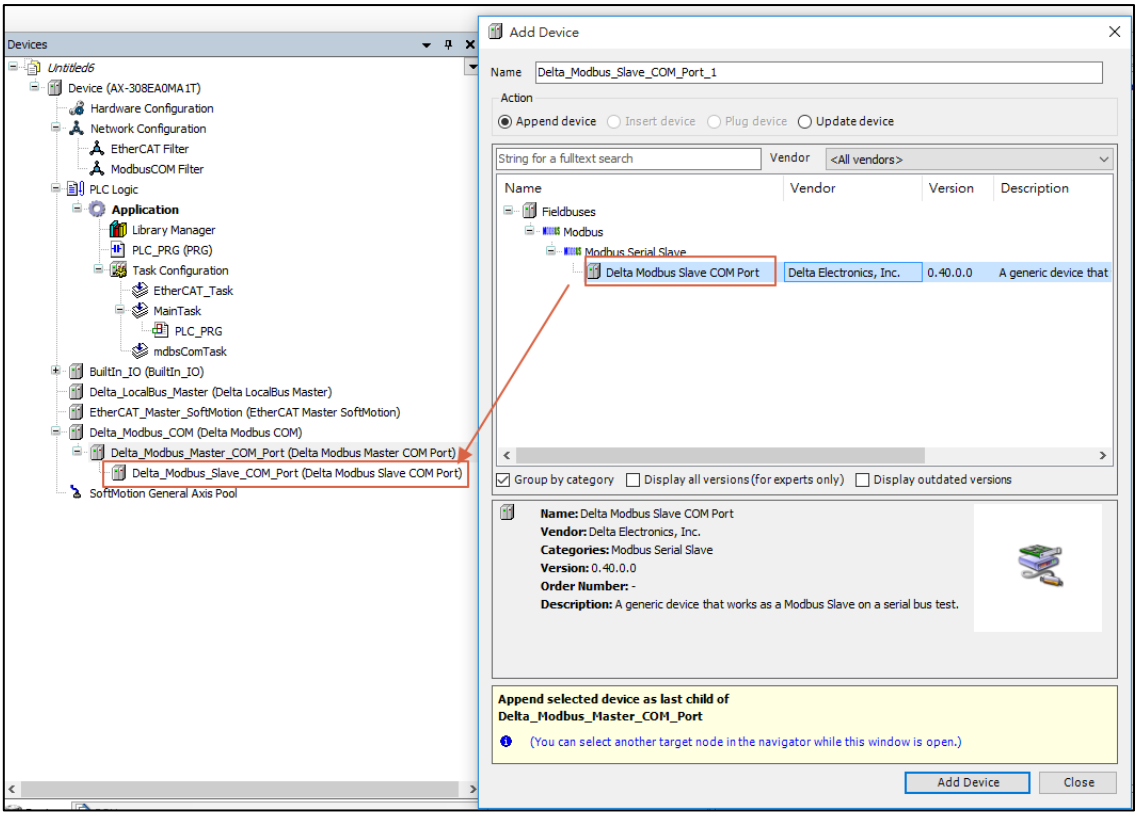


Figure 4 - 35: Adding Delta MODBUS Slave COM Port

- **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.

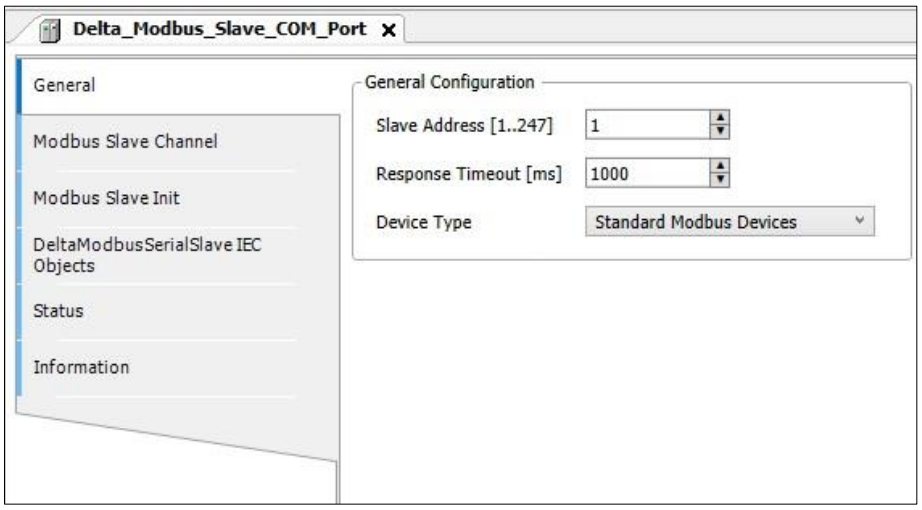


Figure 4 - 36: Delta_MODBUS_Slave_COM_Port – General Tab

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).

4.2.3.4 Tab Delta MODBUS COM Slave – Channel

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.

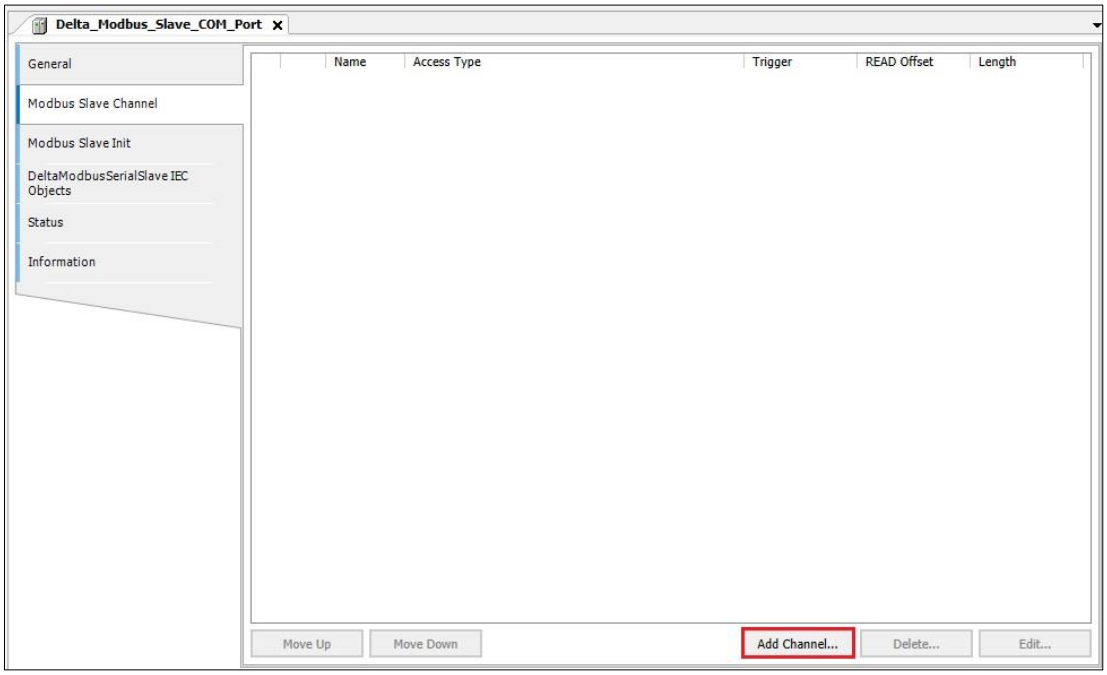


Figure 4 - 37: MODBUS slave channel tab

As shown above, there is an option to *Add Channel* 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.

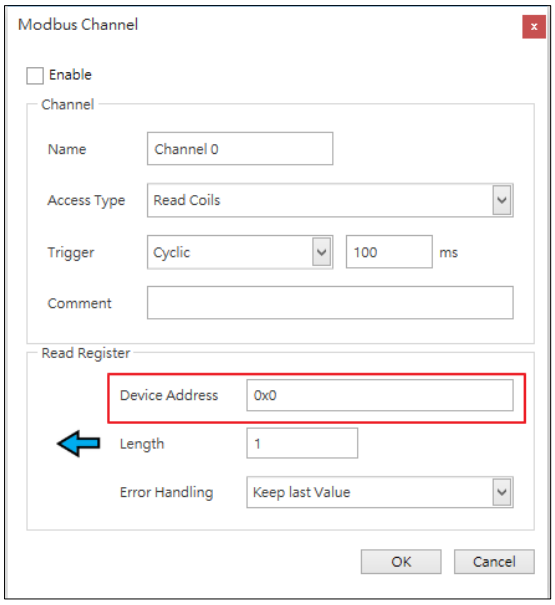


Figure 4 - 38: Standard MODBUS Device

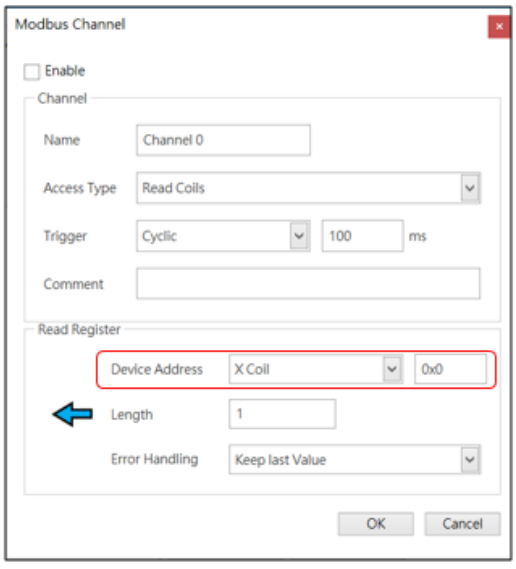


Figure 4 - 39: AH Series

MODBUS Channel Description:

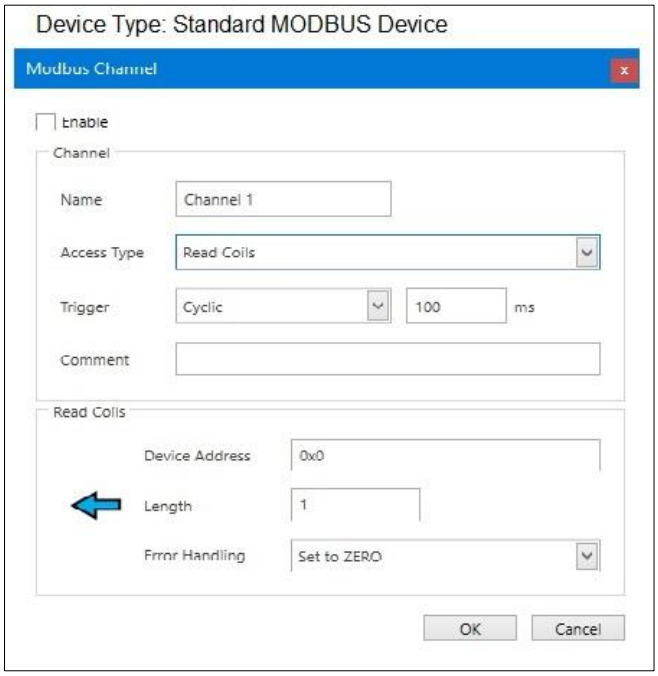


Figure 4 - 40: Standard MODBUS Device

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 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	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

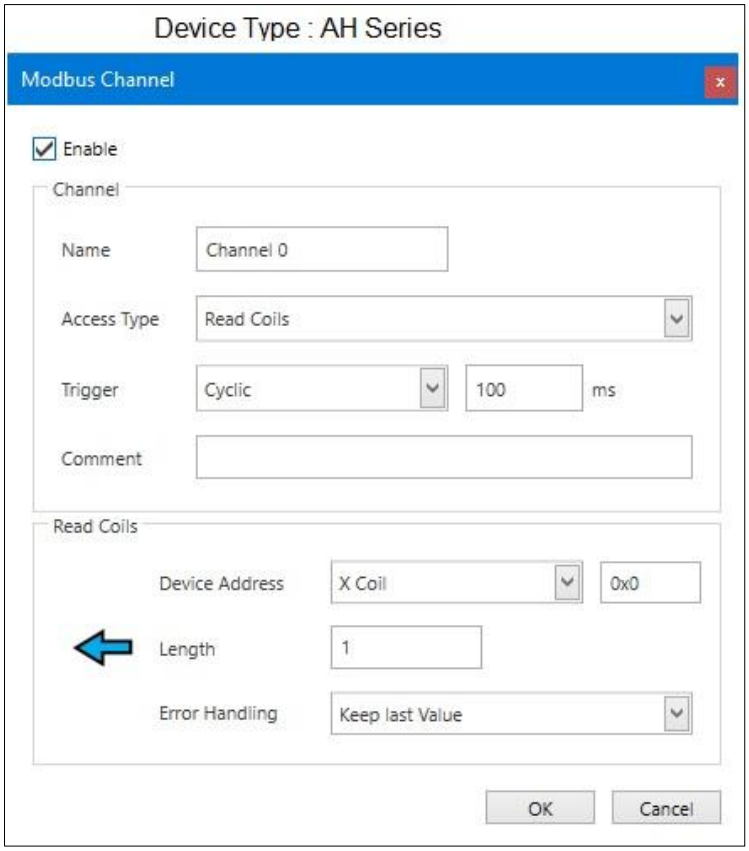


Figure 4 - 41: AH Series

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

Name	Description
	• 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. 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

4.2.3.5 Tab Delta MODBUS COM Slave – Init

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.

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.

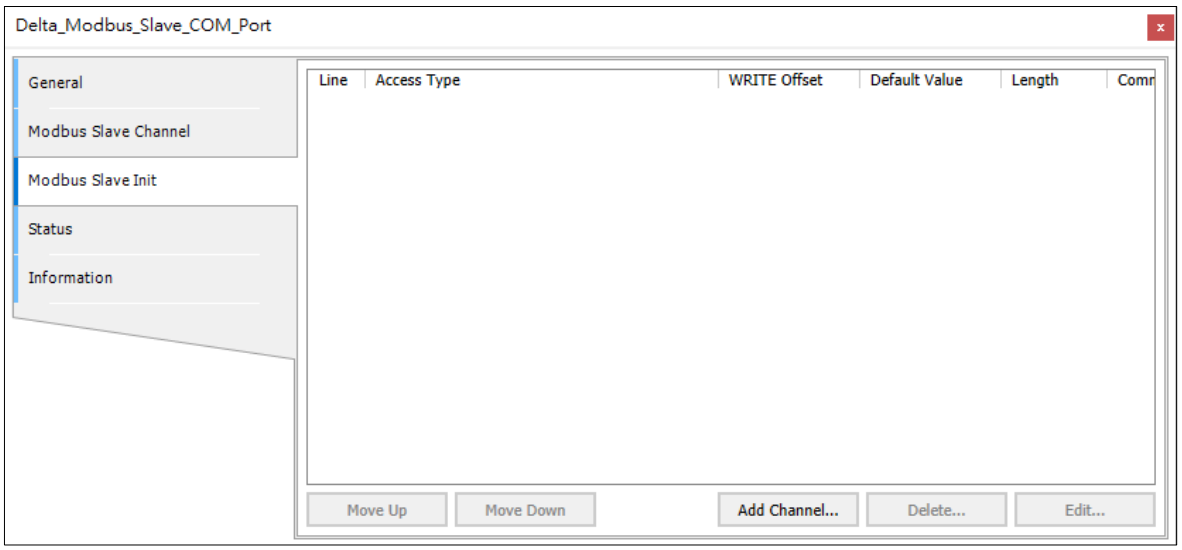


Figure 4 - 42: MODBUS slave Init Tab

4.2.3.6 Tab Delta MODBUS Serial Device Slave

After completing the procedure *4.2.3.2 Tab 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.

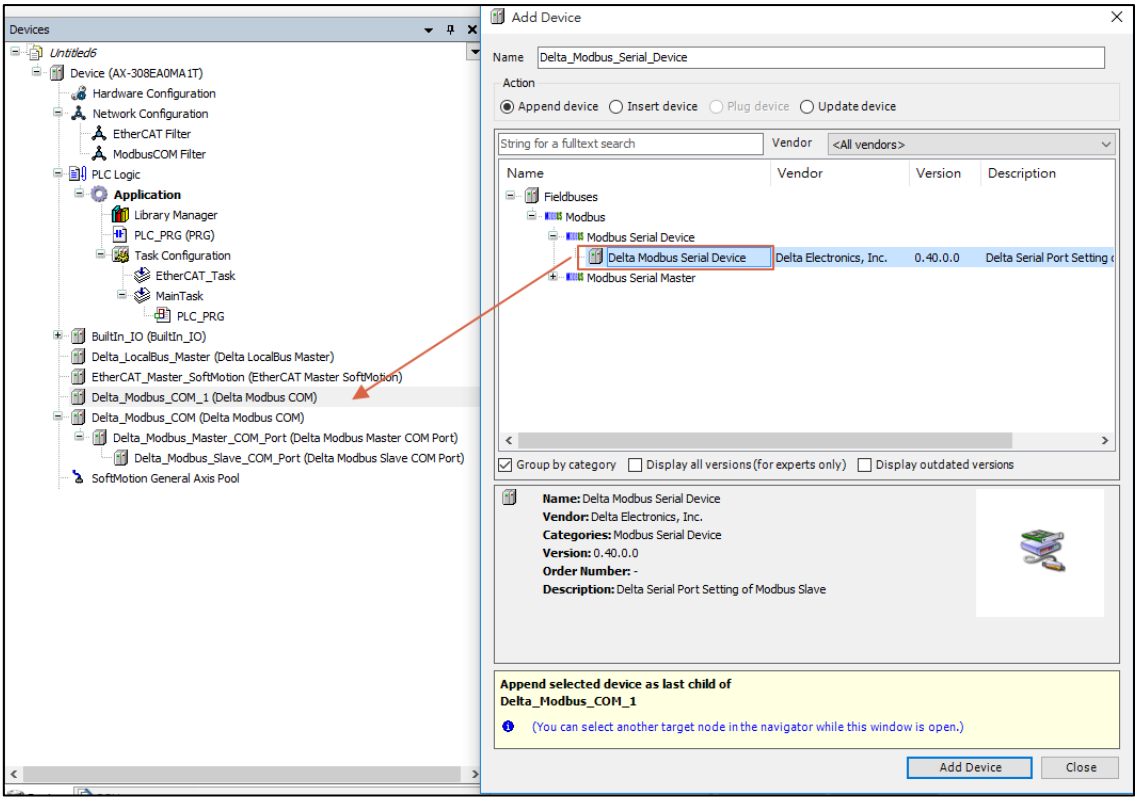


Figure 4 - 43: Adding Delta MODBUS Serial Device

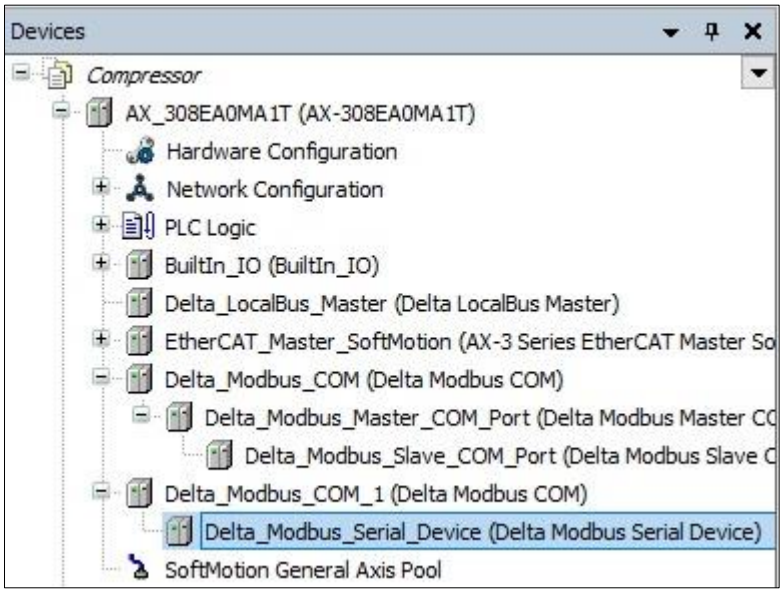


Figure 4 - 44: Devices - Delta MODBUS Serial Device

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.

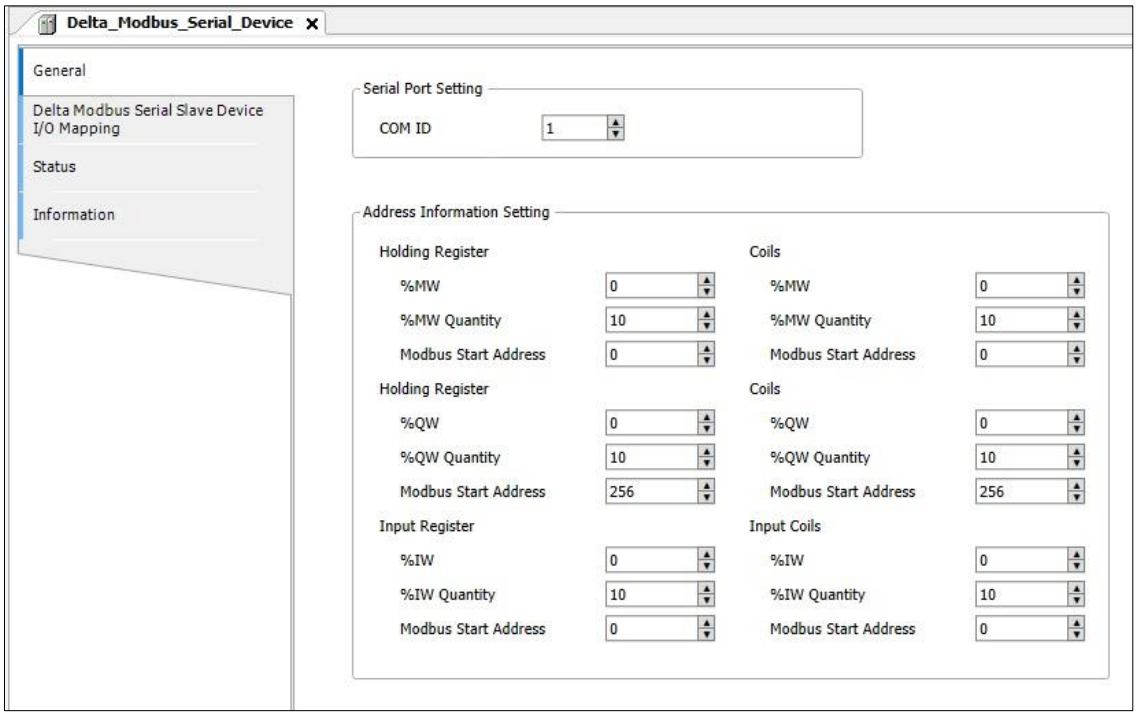


Figure 4 - 45: General Window

- MODBUSSerialSlave 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”.

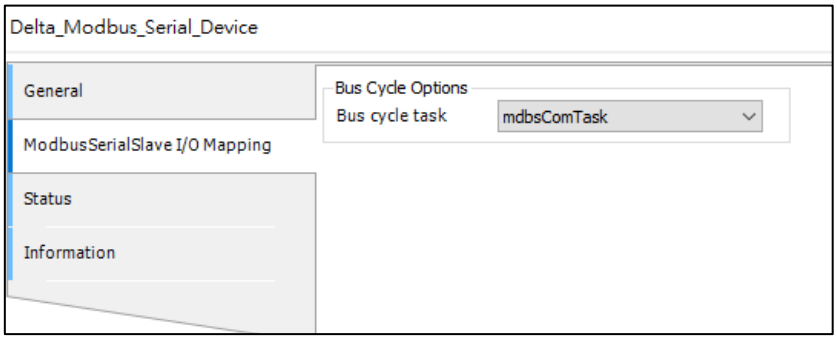


Figure 4 - 46: MODBUS Serial Slave I/O Mapping

4.2.4 EtherCAT Setting

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

4.2.4.1 EtherCAT Master Setting

Click on the *device tree*> *EtherCAT_Master_SoftMotion*.

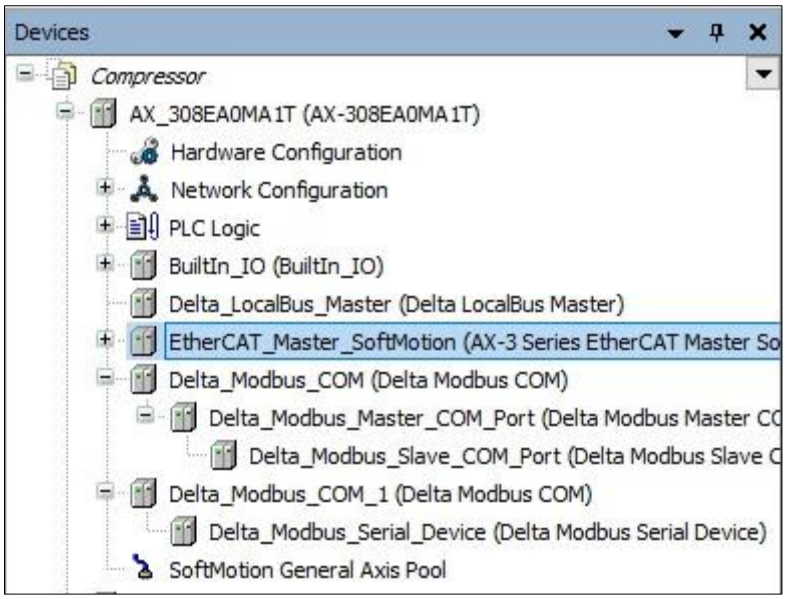


Figure 4 - 47: EtherCAT Master Setting

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

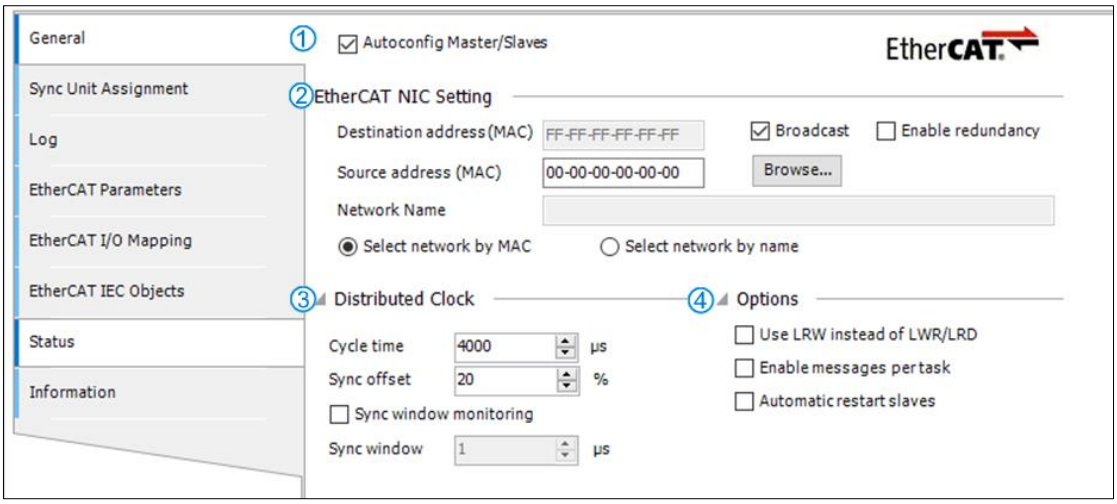


Figure 4 - 48: General Tab

- 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

- 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.

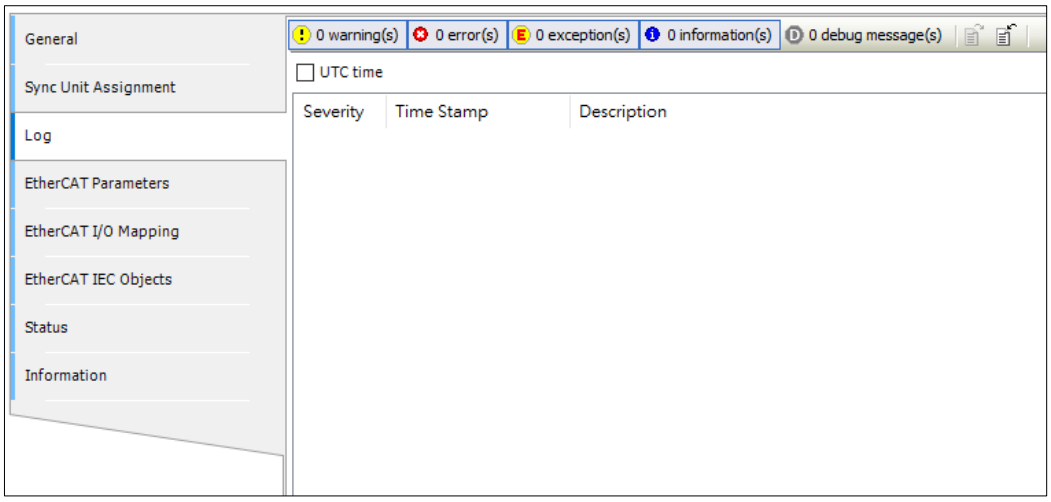


Figure 4 - 49: Log window

- 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.

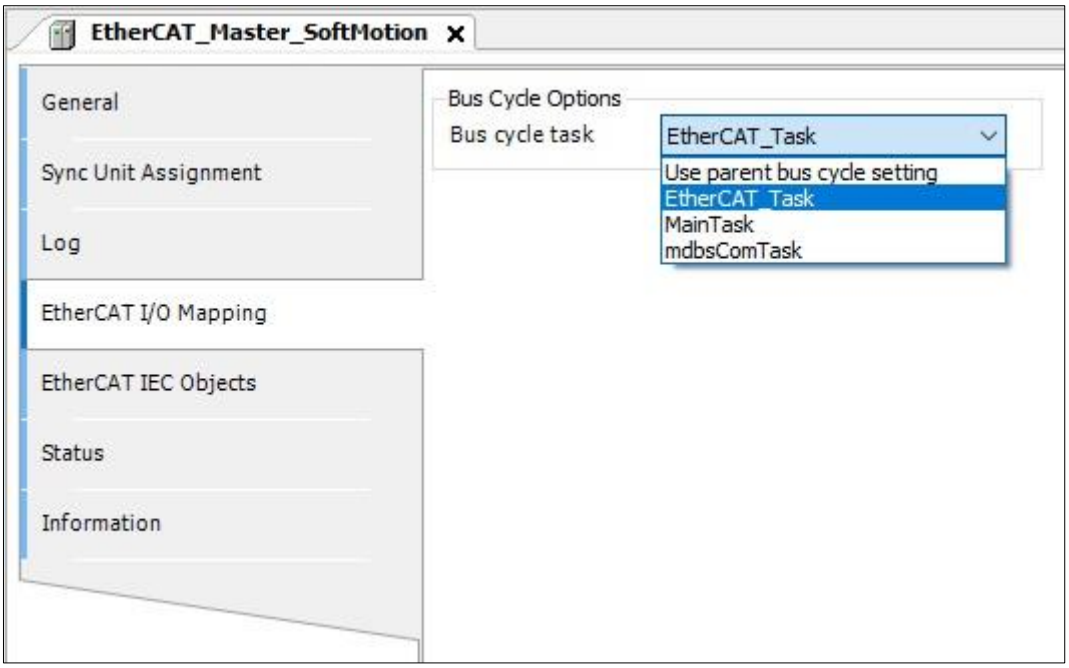


Figure 4 - 50: EtherCAT I/O Mapping

4.2.4.2 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.

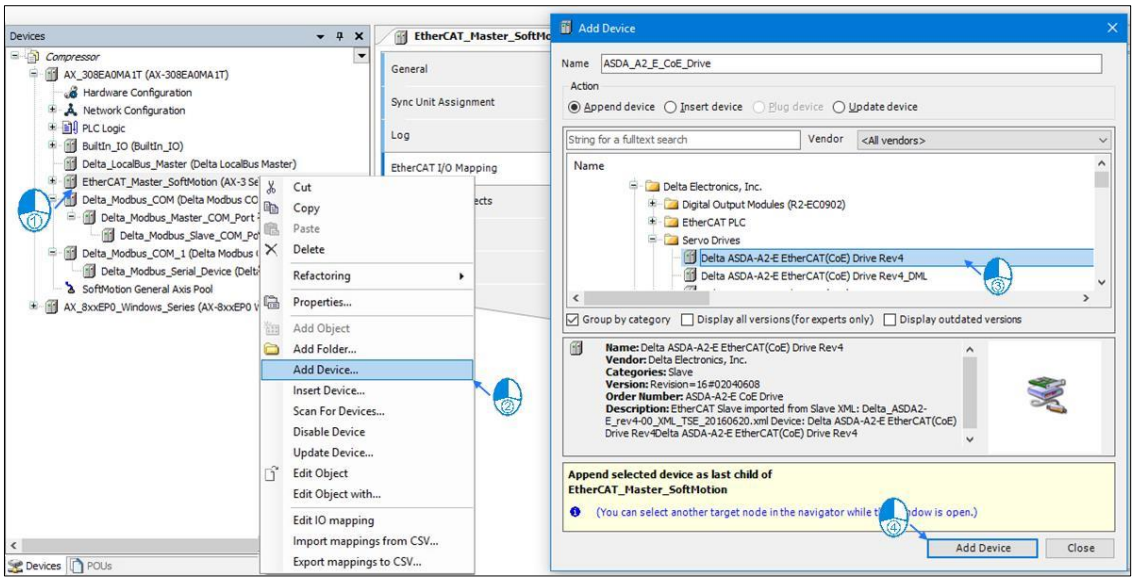


Figure 4 - 51: Add EtherCAT Slave

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.

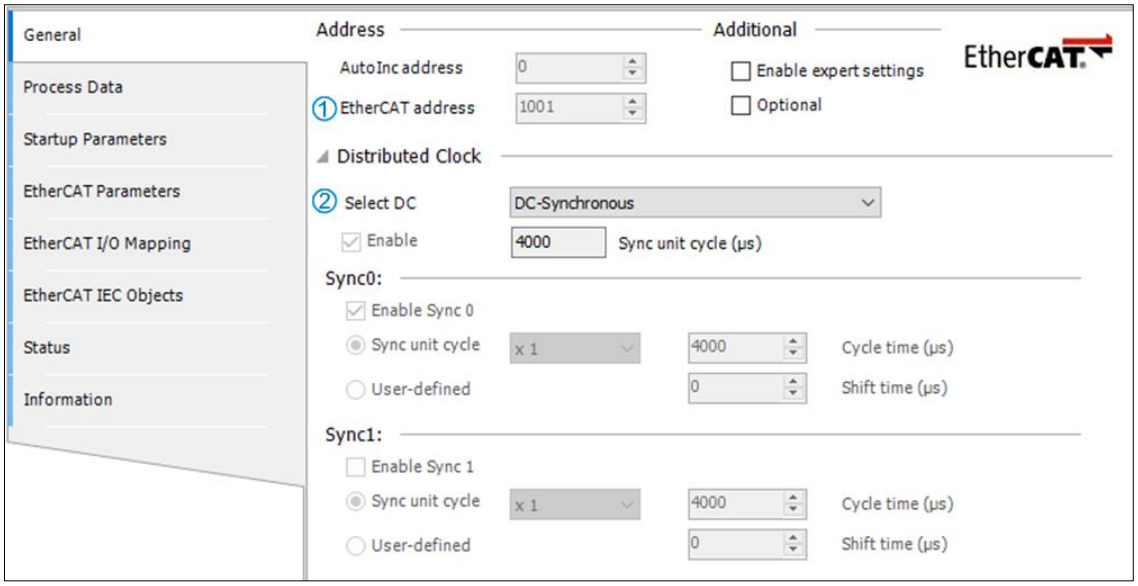


Figure 4 - 52: General Window

- 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.

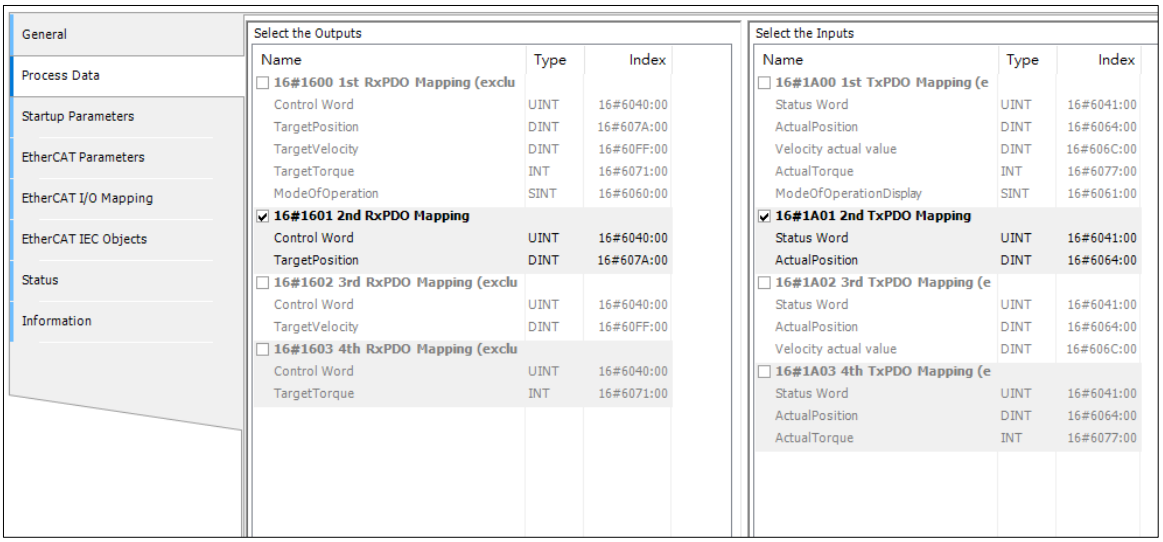


Figure 4 - 53: Process Data Window

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

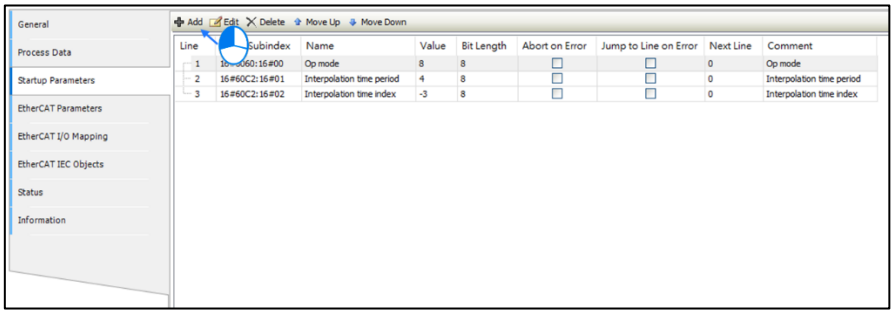


Figure 4 - 54: Startup Parameter

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.

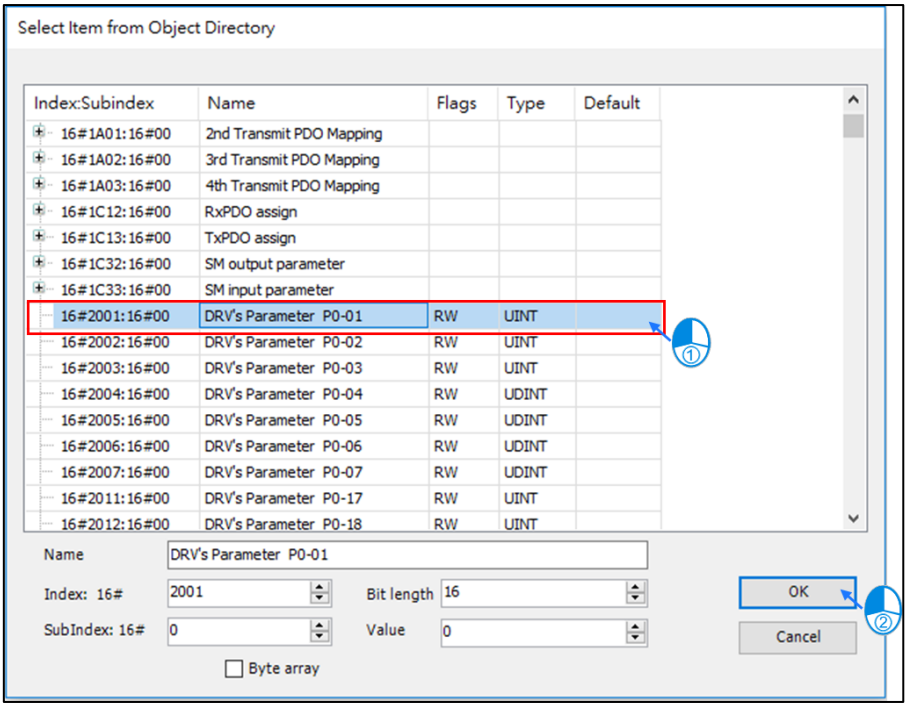


Figure 4 - 55: Selecting the Object Directory

4.2.5 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.

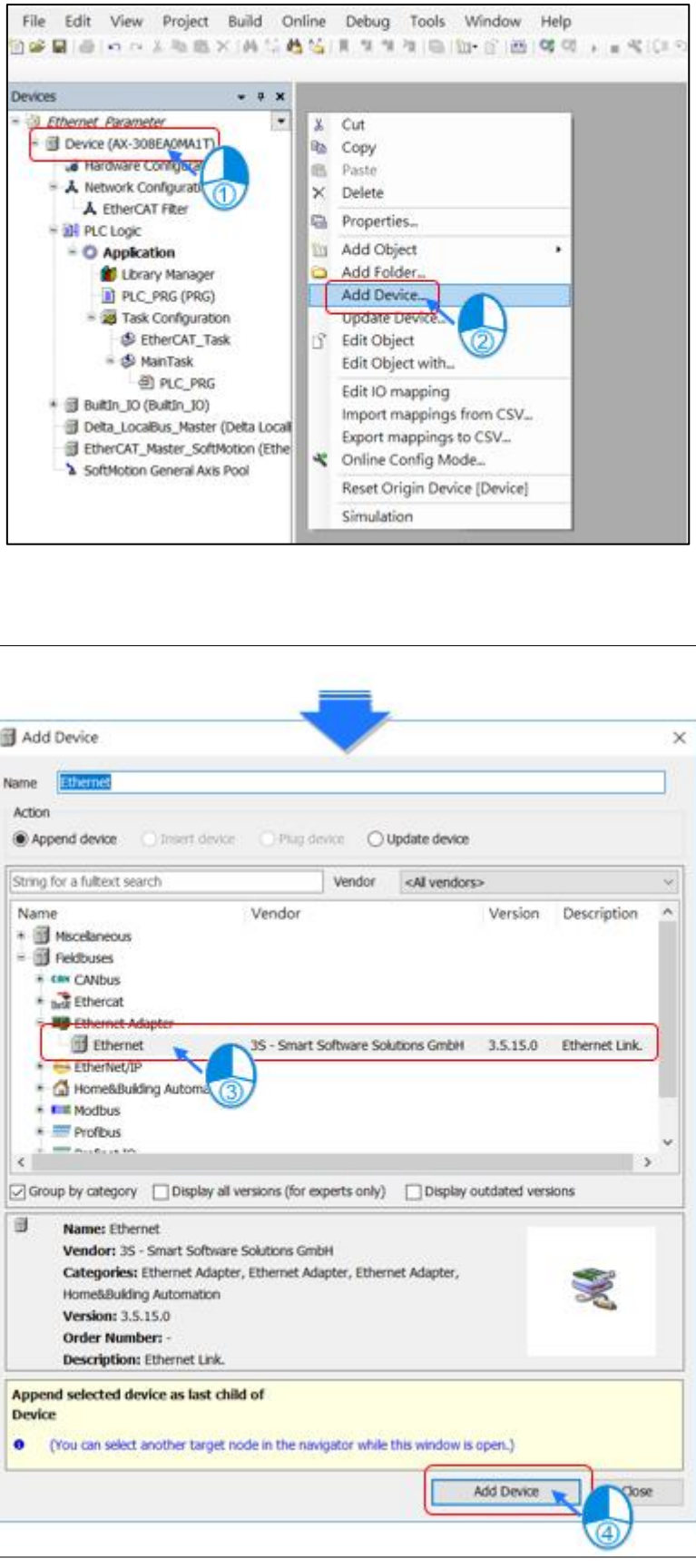


Figure 4 - 56: Configuring Ethernet

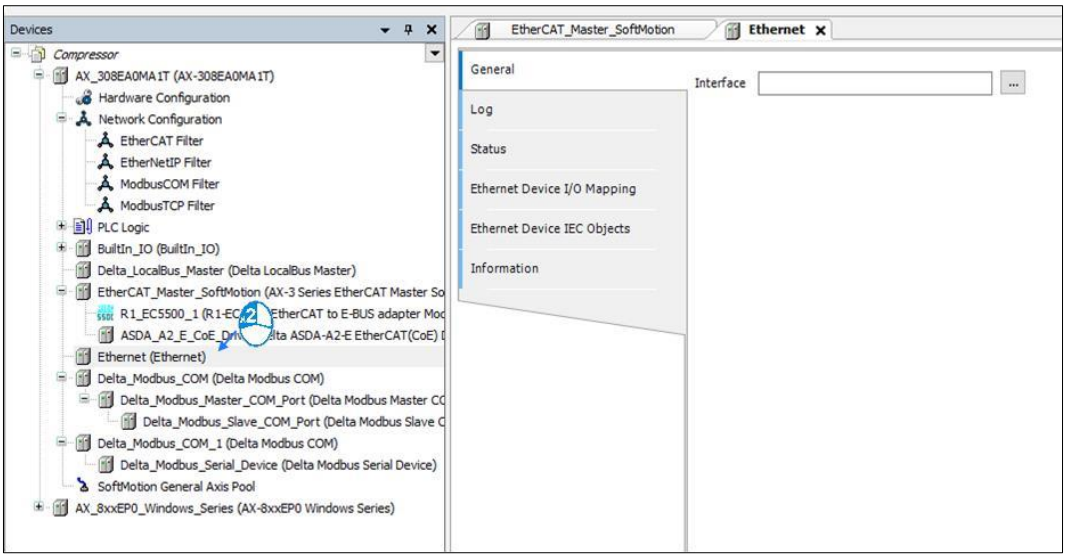


Figure 4 - 57: Ethernet added to Application

- EtherNet Device – General

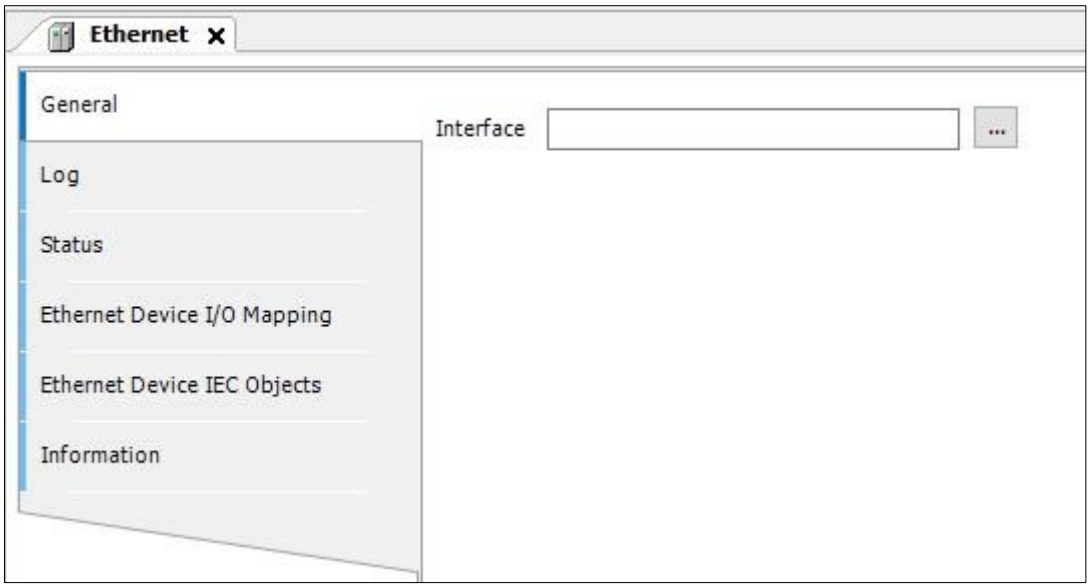


Figure 4 - 58: General Tab

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.

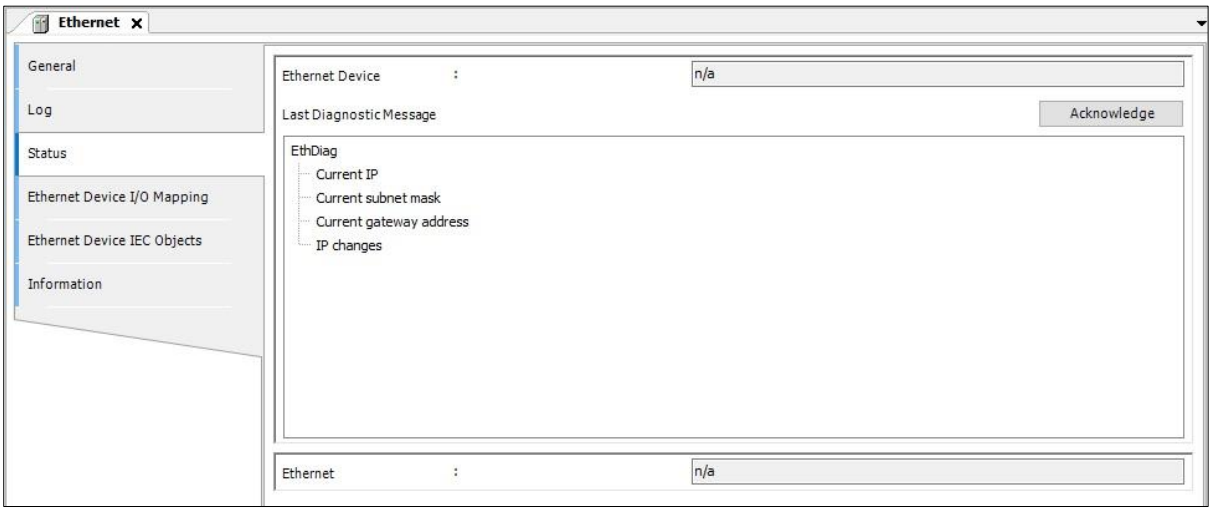


Figure 4 - 59: Status Tab

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”.

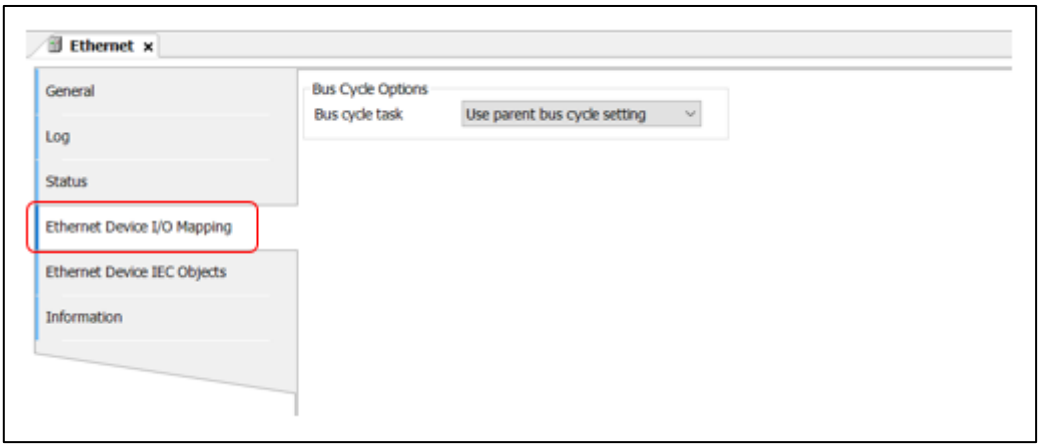


Figure 4 - 60: Ethernet Device I/O Mapping Tab

- **Ethernet Device IEC Objects**

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

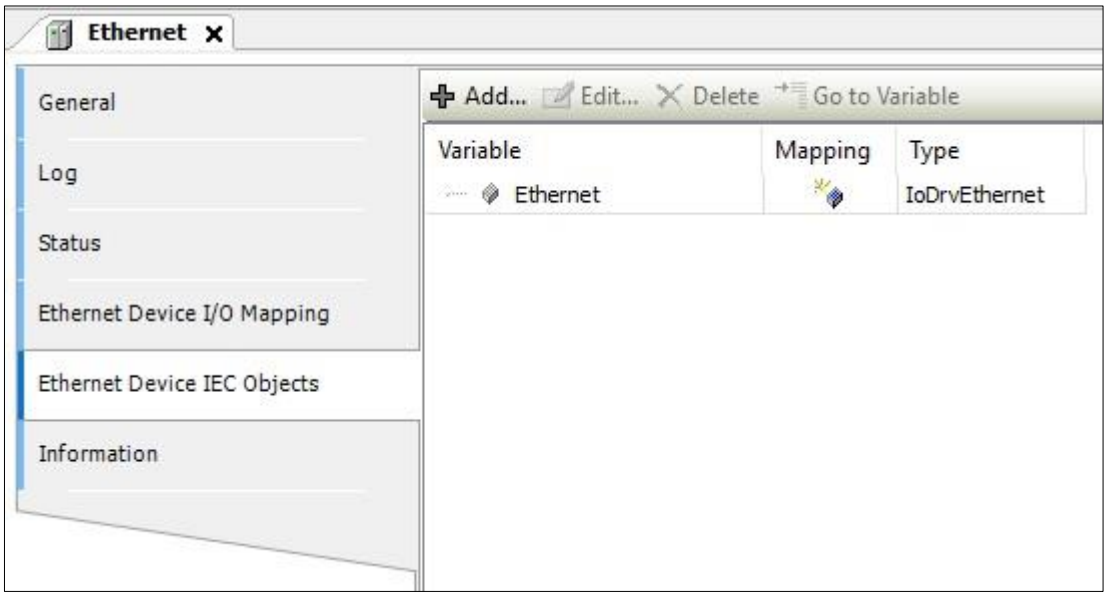


Figure 4 - 61: Ethernet Device IEC Objects Tab

4.2.5.1 Tab 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*.

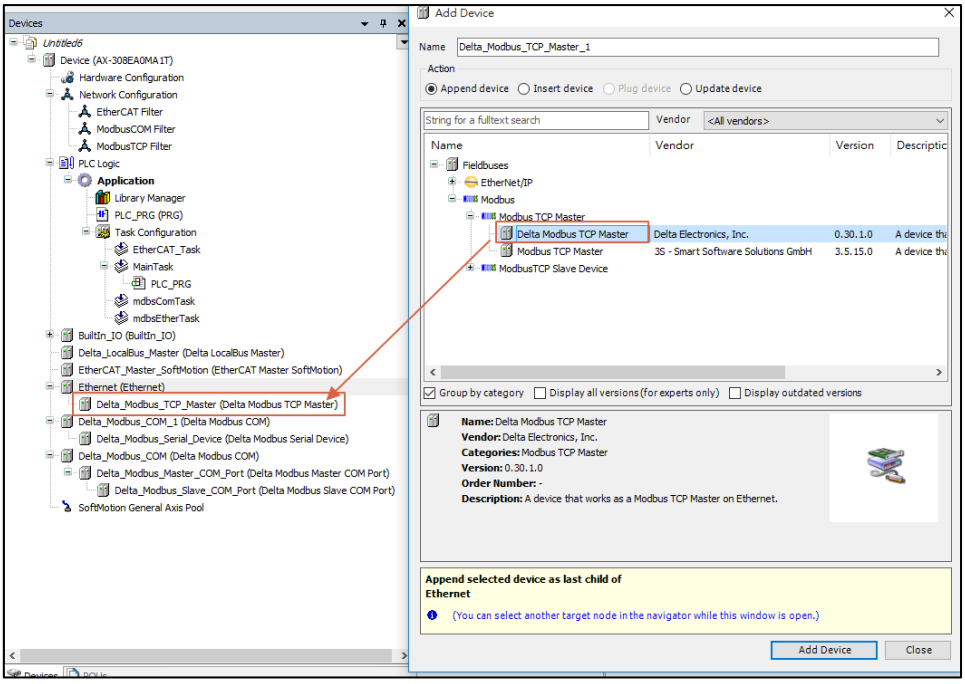


Figure 4 - 62: Adding Delta MODBUS TCP Master

- Delta MODBUS TCP Master I/O Mapping:

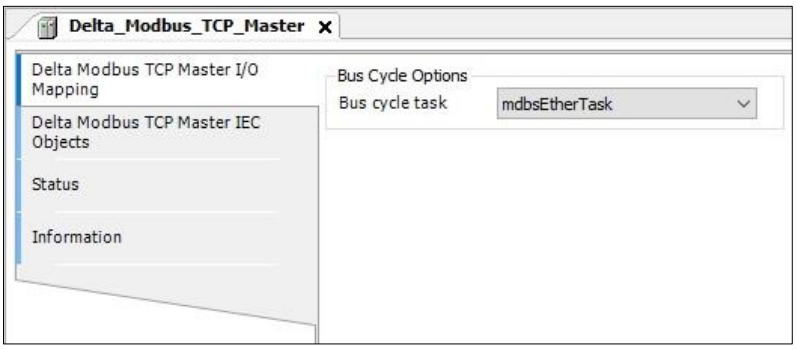


Figure 4 - 63: Delta MODBUS TCP Master I/O Mapping

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

4.2.5.2 Tab Delta MODBUS TCP Slave - General

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*.

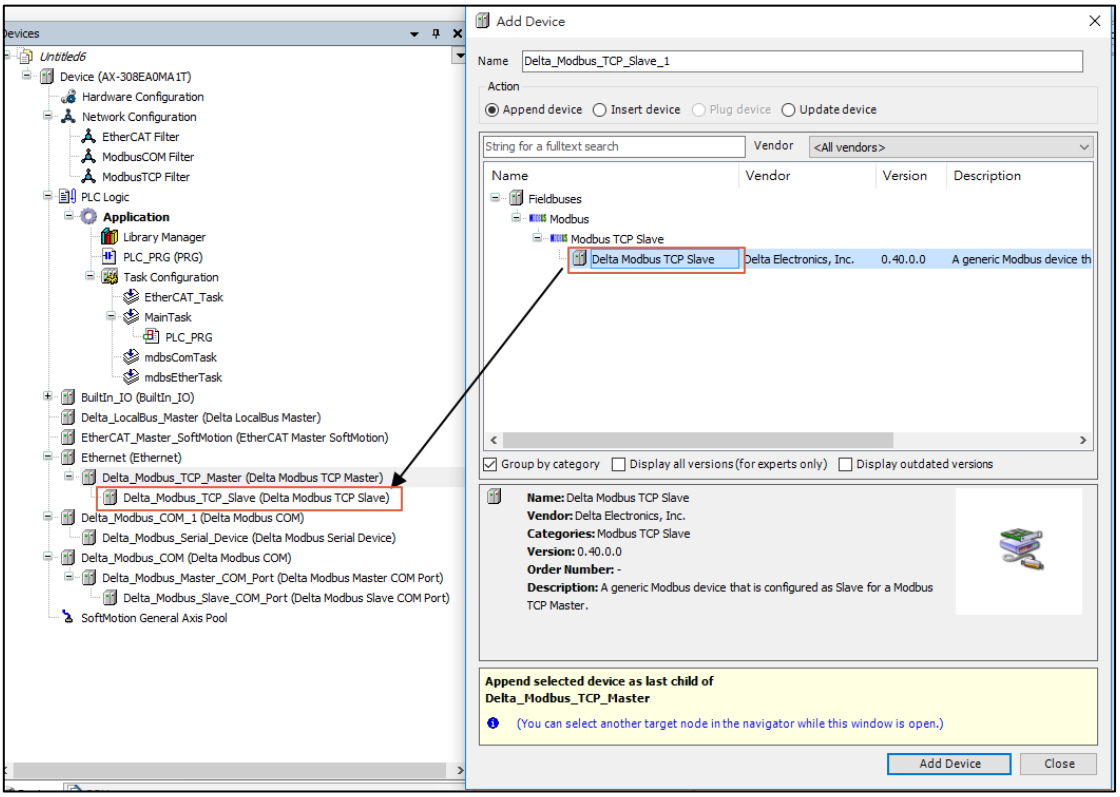


Figure 4 - 64: Delta MODBUS TCP Slave

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.

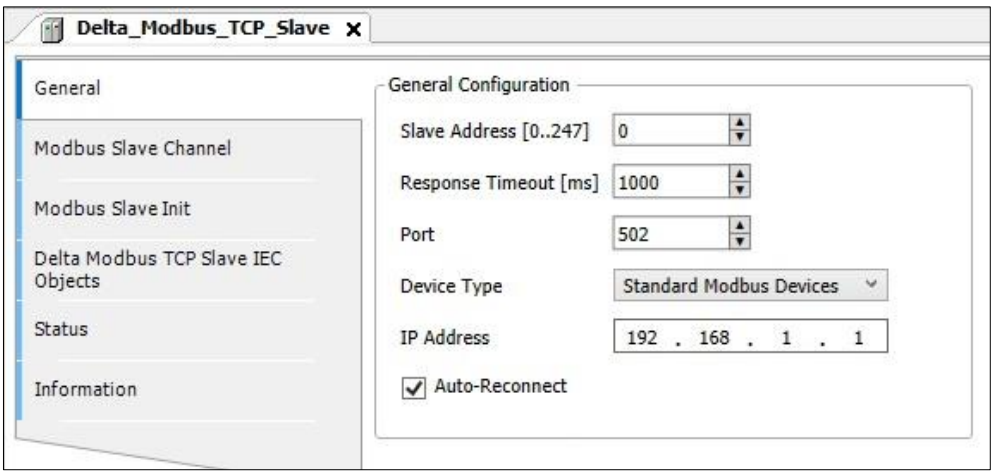


Figure 4 - 65: Delta MODBUS TCP Slave – General Tab

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.

4.2.5.3 Tab Delta MODBUS TCP Slave – Channel

- 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.

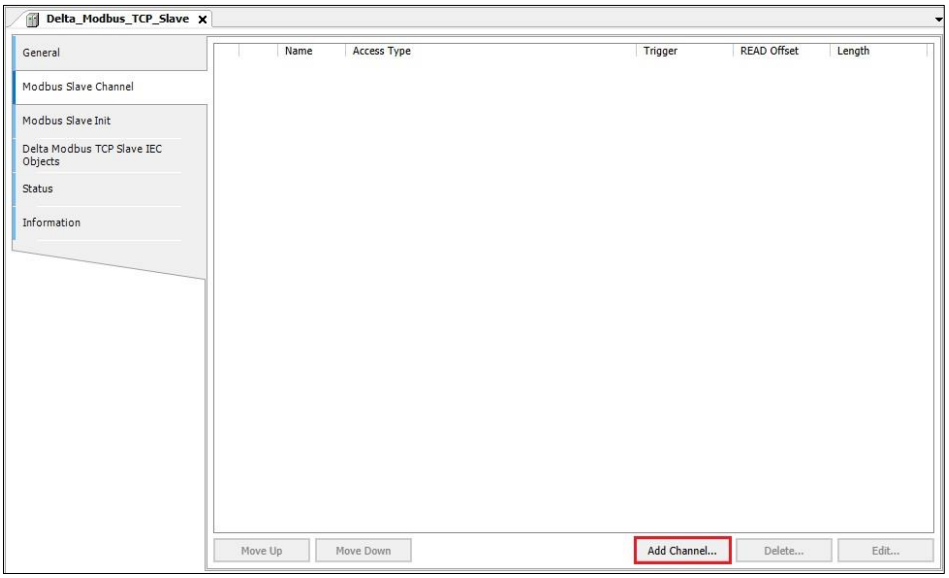


Figure 4 - 66: Delta MODBUS TCP Slave – MODBUS slave channel Tab

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

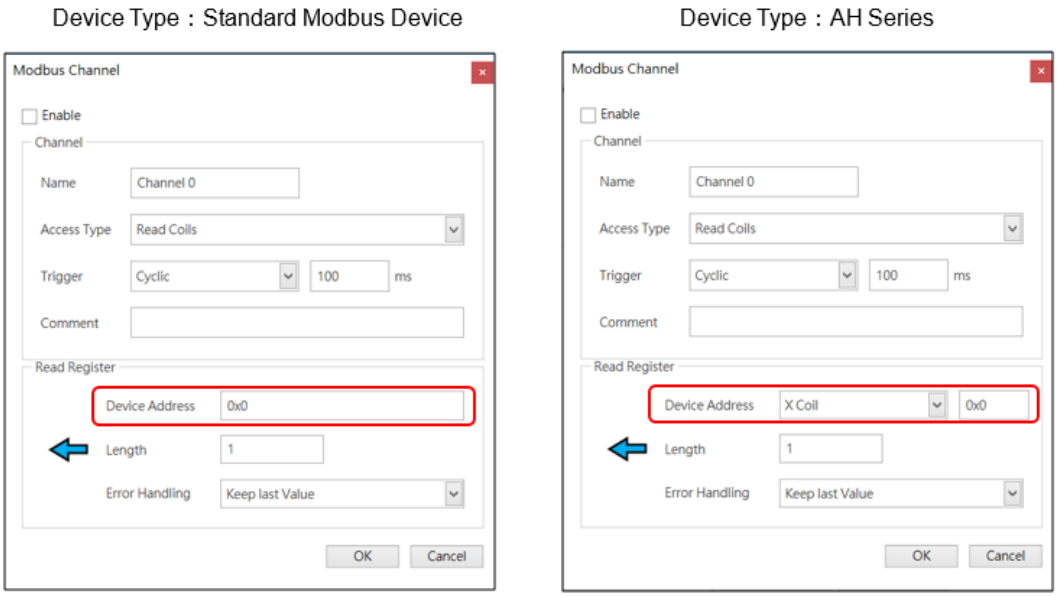


Figure 4 - 67: Standard MODBUS Device and AH Series

MODBUS Channel Description:

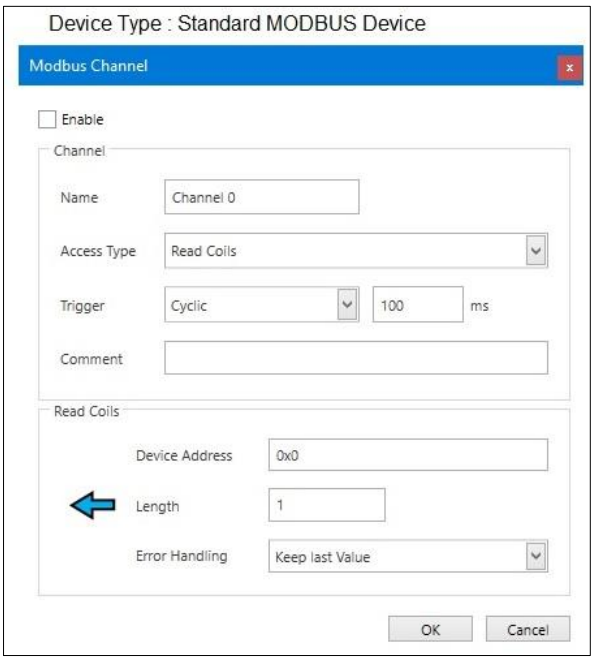


Figure 4 - 68: MODBUS Channel of Standard MODBUS Device

Name	Description
Enable	Decide whether this channel is effective.

Name	Description
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. 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.

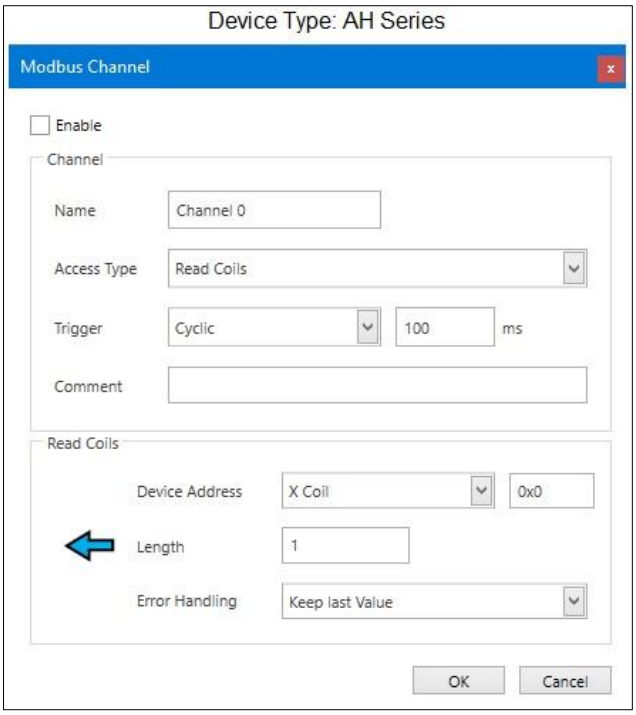


Figure 4 - 69: AH Series

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. 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).

Name	Description
Error Handing	When a communication error occurs, the setting operation of the Registers data is performed. • Set To ZERO. • Keep last value.

4.2.5.4 Tab Delta MODBUS TCP Slave – Init

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.

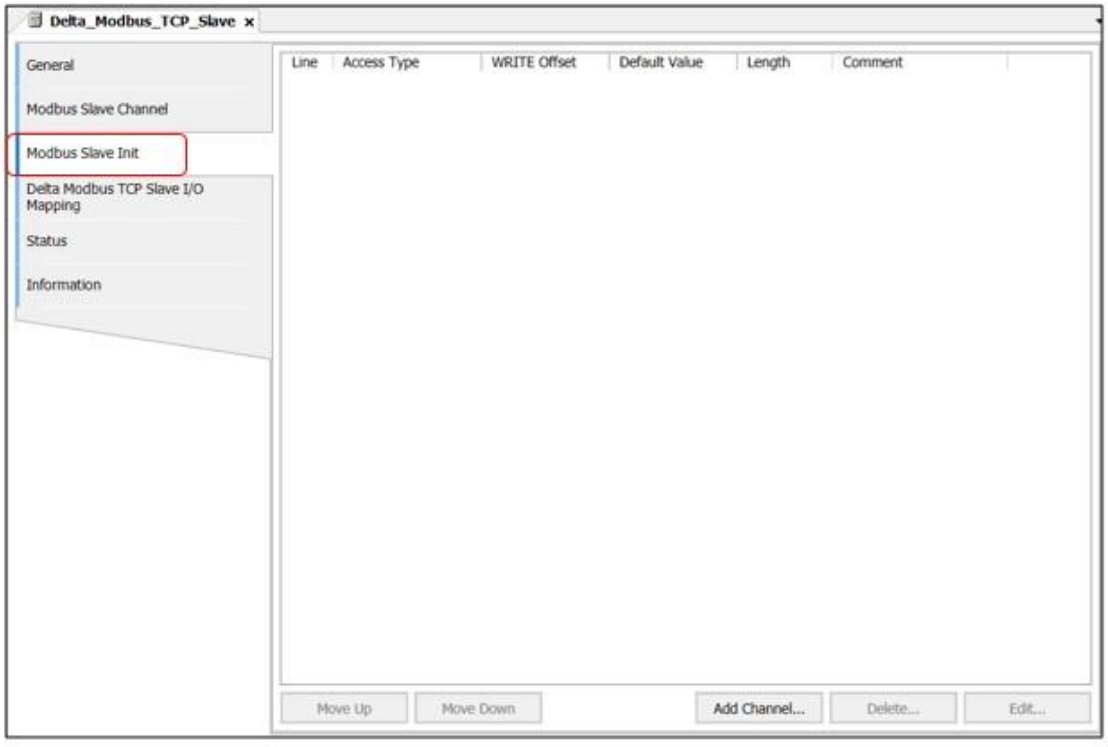
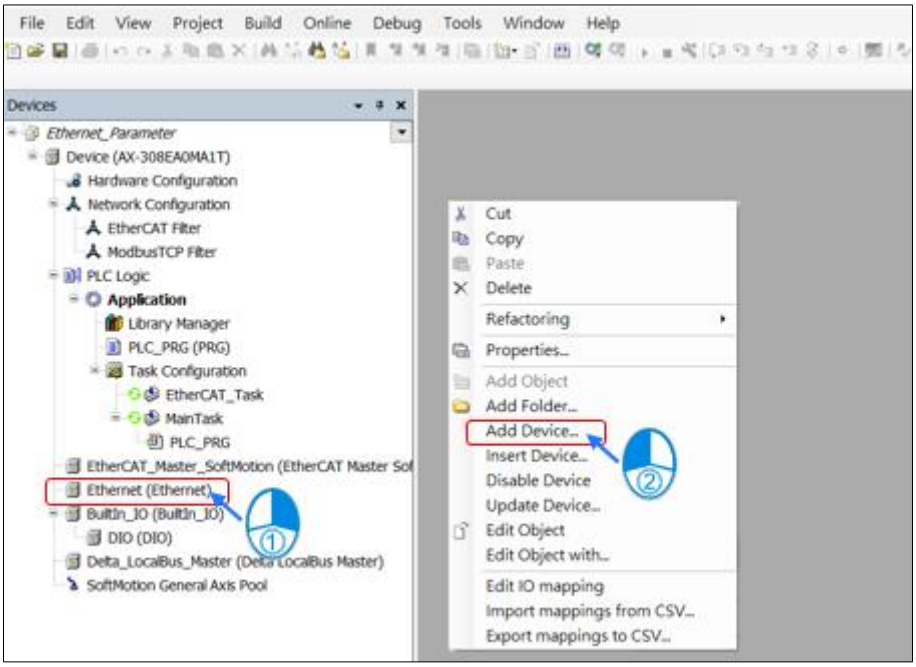


Figure 4 - 70: Delta_MODBUS_TCP_Slave – MODBUS slave Init

4.2.5.5 Tab 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.*



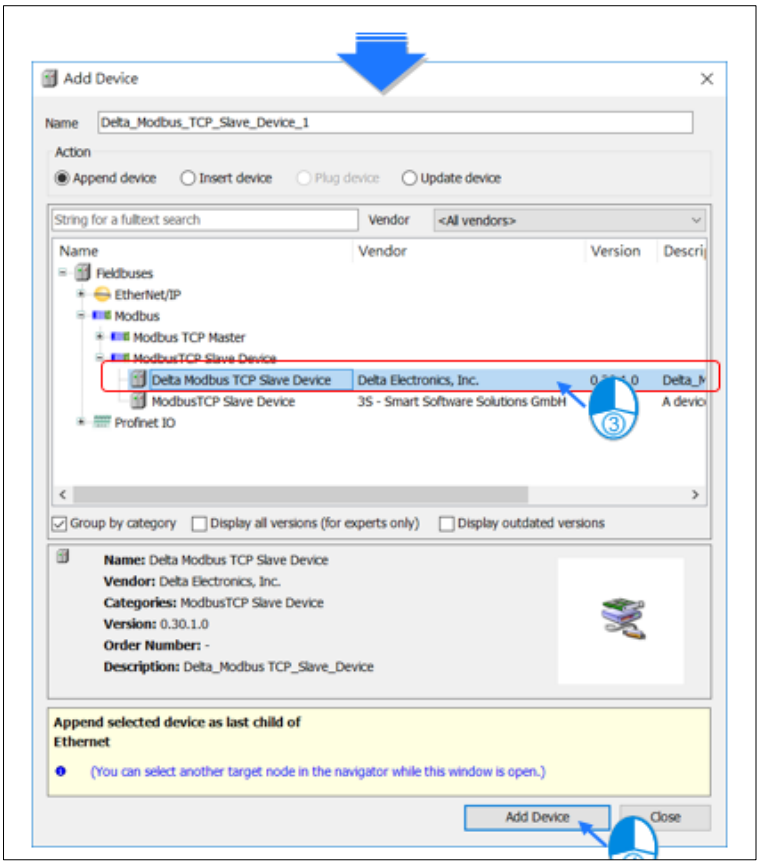


Figure 4 - 71: Adding EtherNet to Device

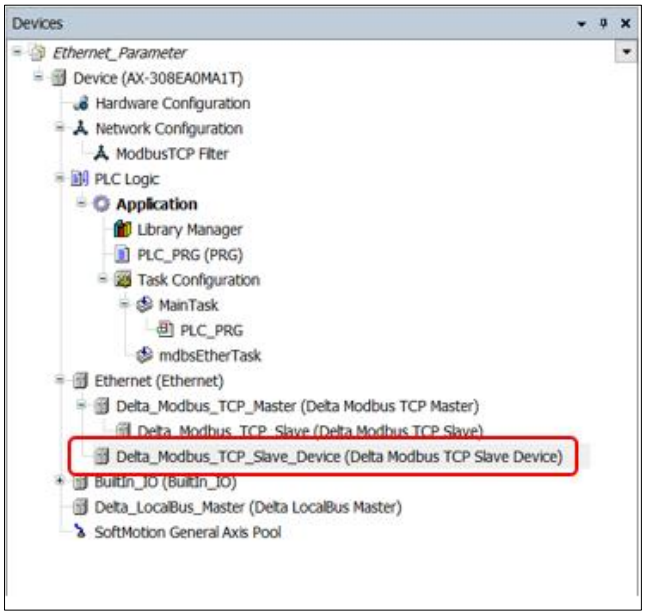


Figure 4 - 72: Adding Delta_MODBUS_TCP_Slave Device

- General:

  User can set the controlled port, station ID, and the register area in this page. If the MODBUSTCP Master is using Delta specific communication, the register is not limited.

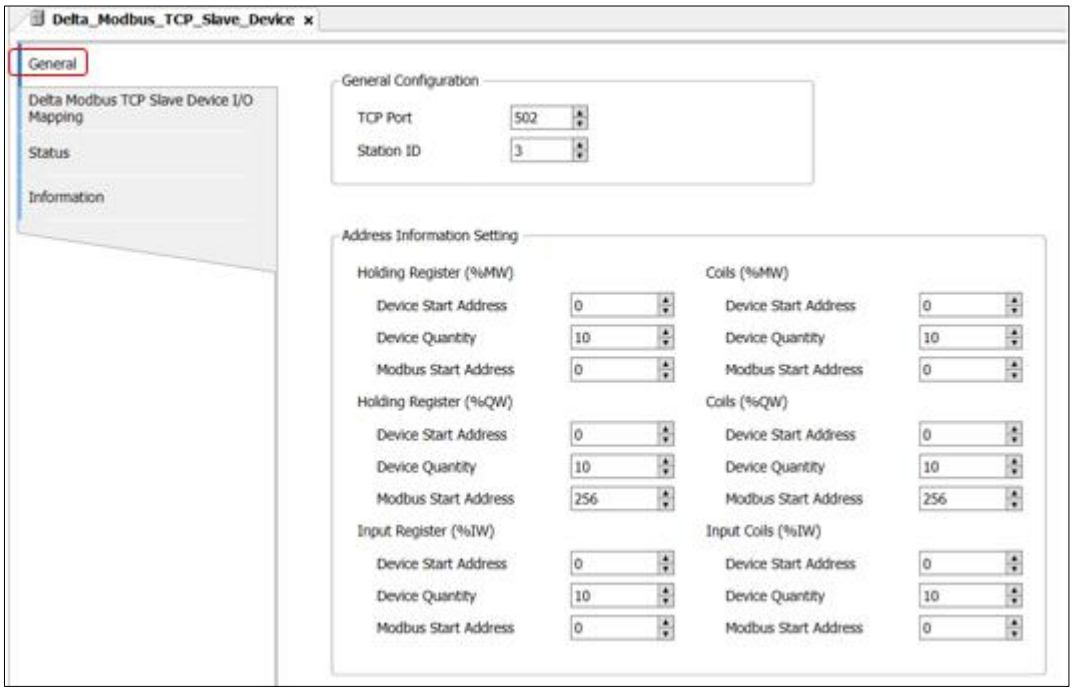


Figure 4 - 73: Delta_MODBUS_TCP_Slave Device – General Tab

- 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*.

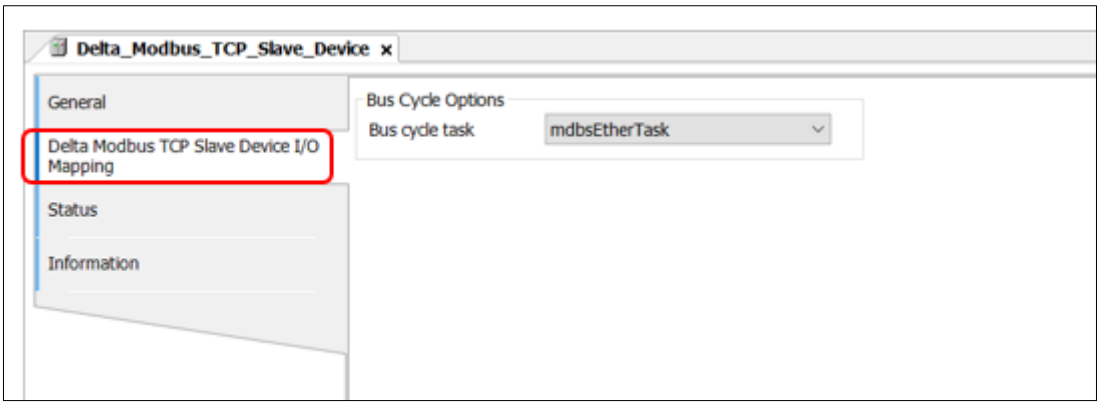


Figure 4 - 74: Delta_MODBUS_TCP_Slave Device I/O Mapping Tab

4.2.6 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.

.

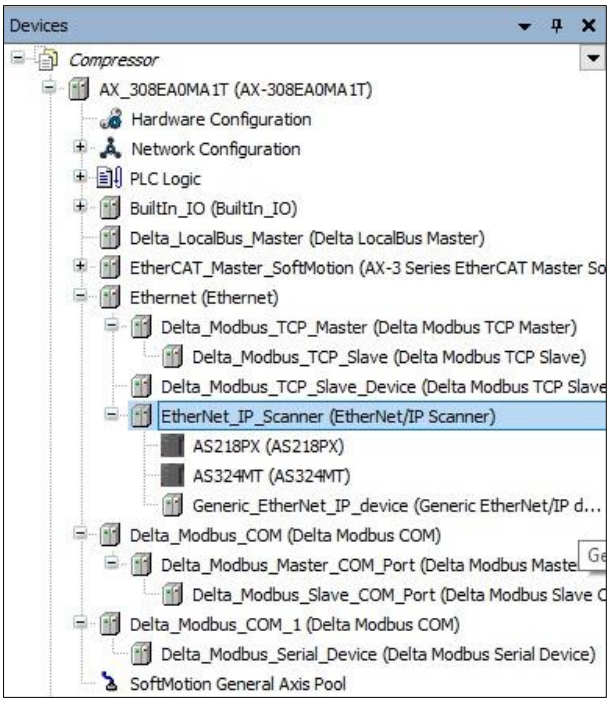


Figure 4 - 75: EtherNet/IP Configuration

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.

4.2.6.1 EtherNet/IP

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

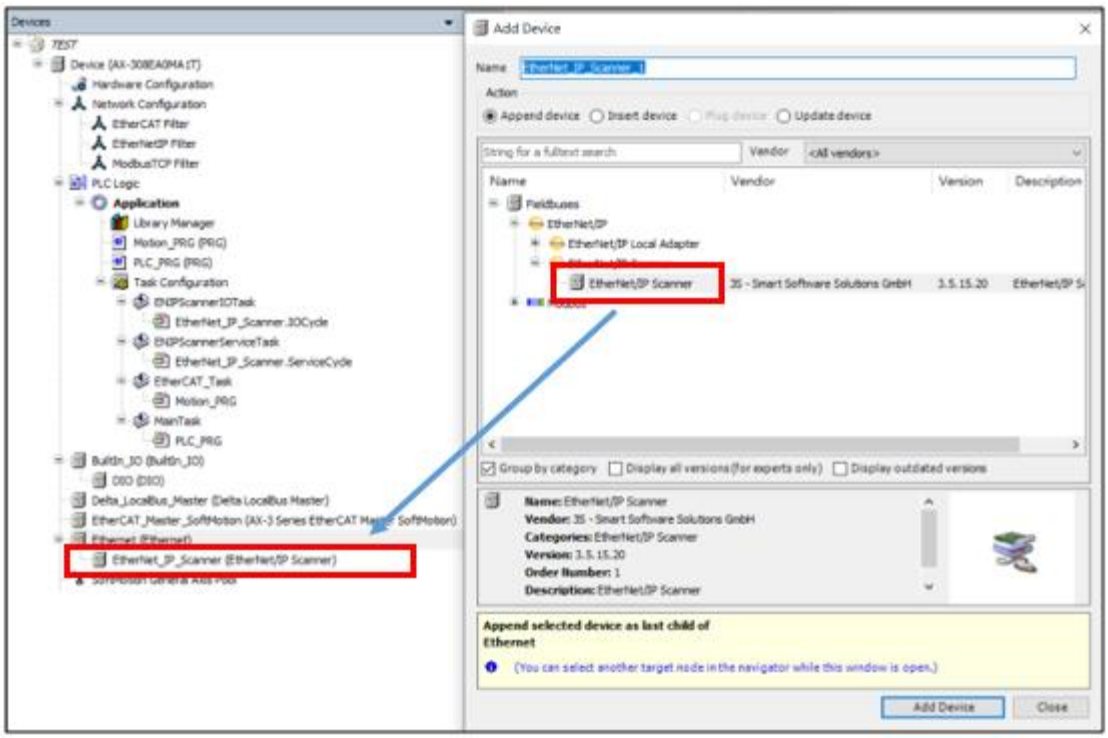


Figure 4 - 76: Increase EtherNet/IP Scanner

EtherNet/IP Scanner General tab

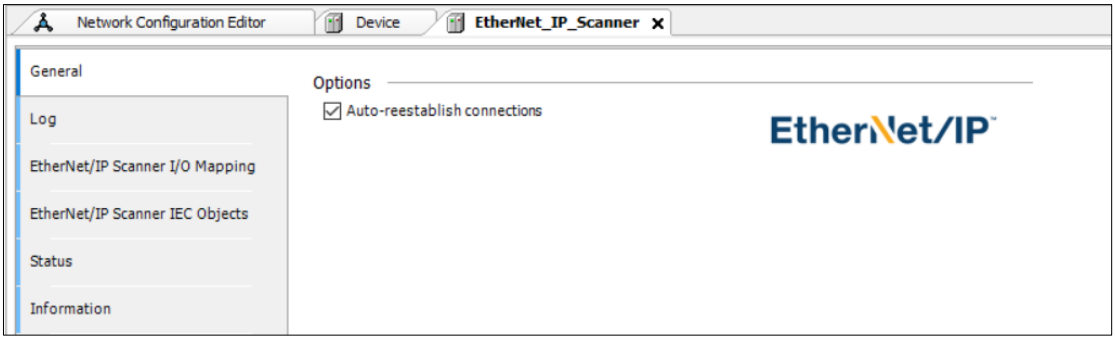


Figure 4 - 77: EtherNet/IP Scanner General tab

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

EtherNet/IP Scanner I/O Map tab

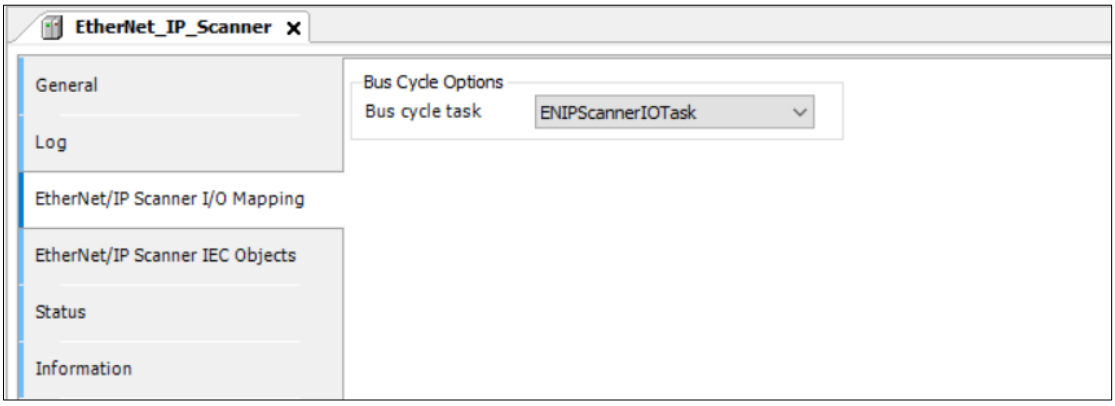


Figure 4 - 78: EtherNet/IP Scanner I/O Map tab

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".

EtherNet/IP Scanner IEC Object tab

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

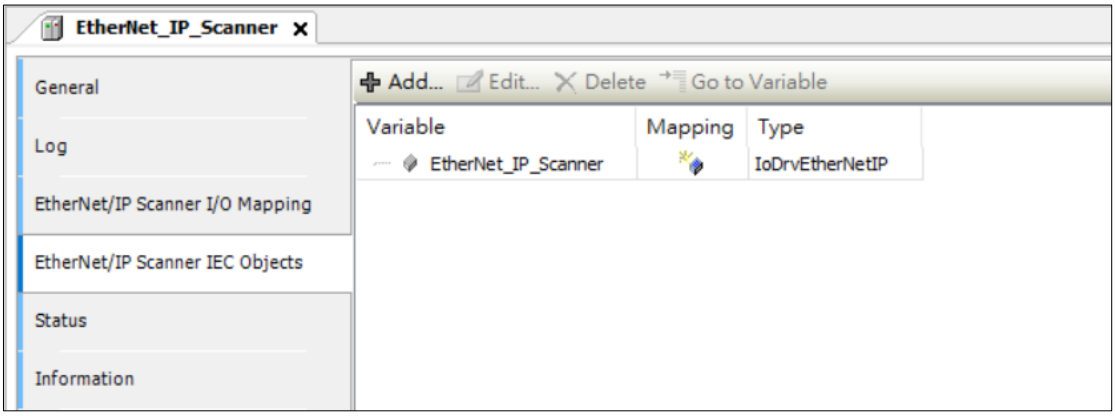


Figure 4 - 79: EtherNet/IP Scanner IEC Object tab

4.2.6.2 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*.

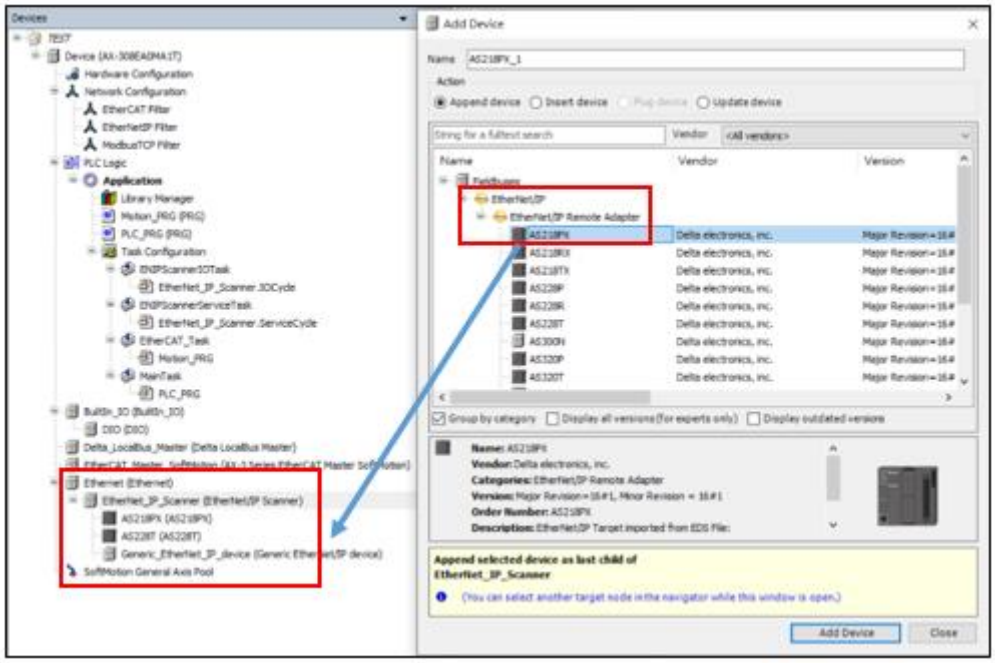


Figure 4 - 80: Add EtherNet/IP Remote Adapter

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

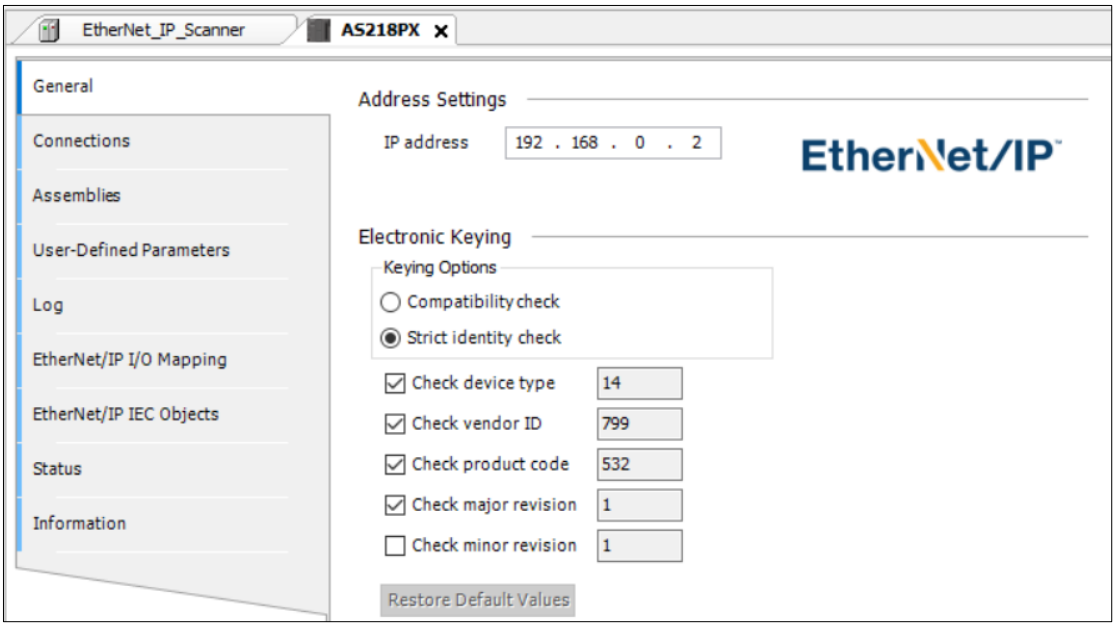


Figure 4 - 81: EtherNet/IP Remote Adapter General Tab

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.

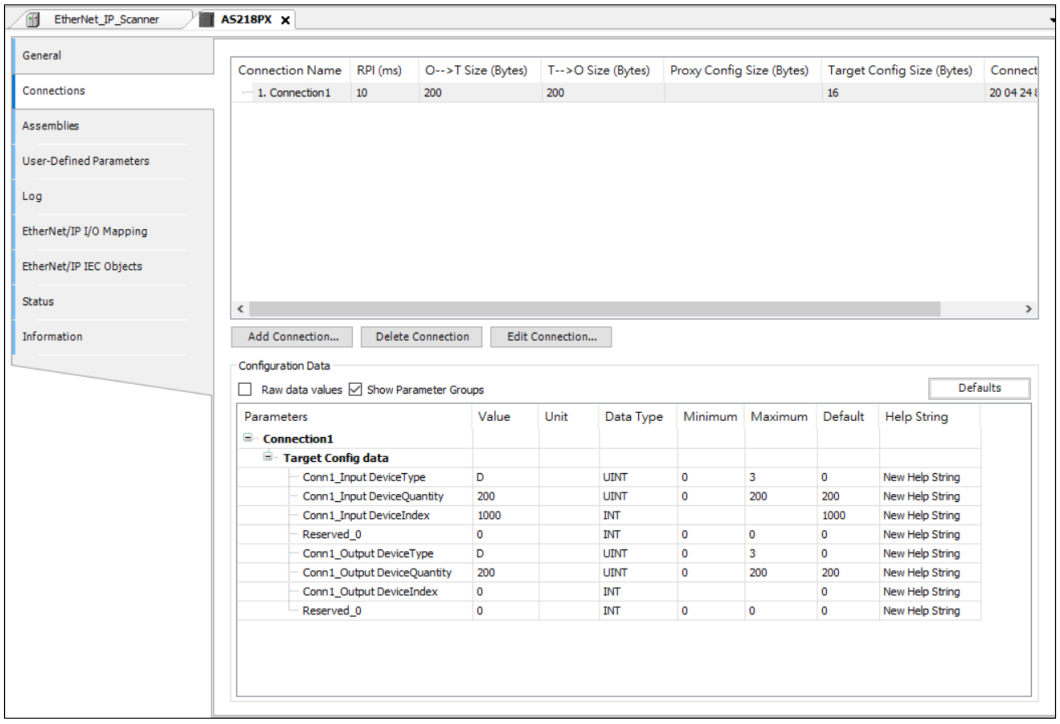


Figure 4 - 82: EtherNet/IP Remote Adapter Connection Tab

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

Project	Description
	（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
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.

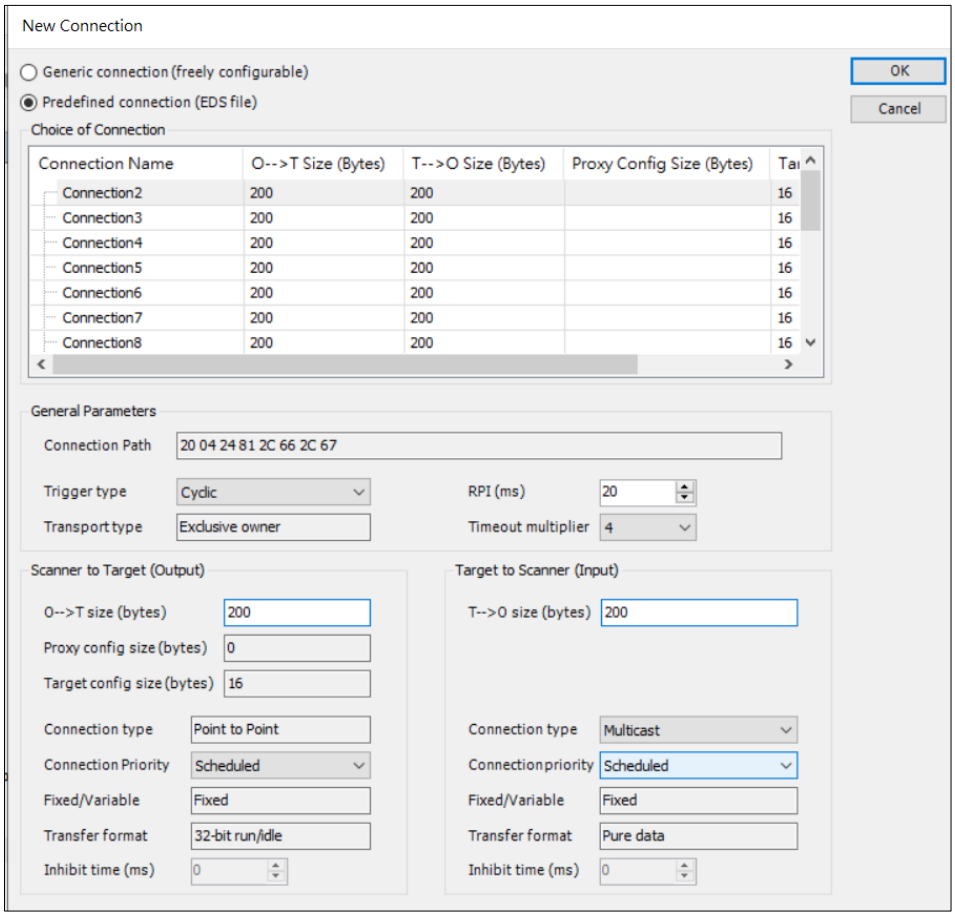


Figure 4 - 83: Add Connection Screen

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

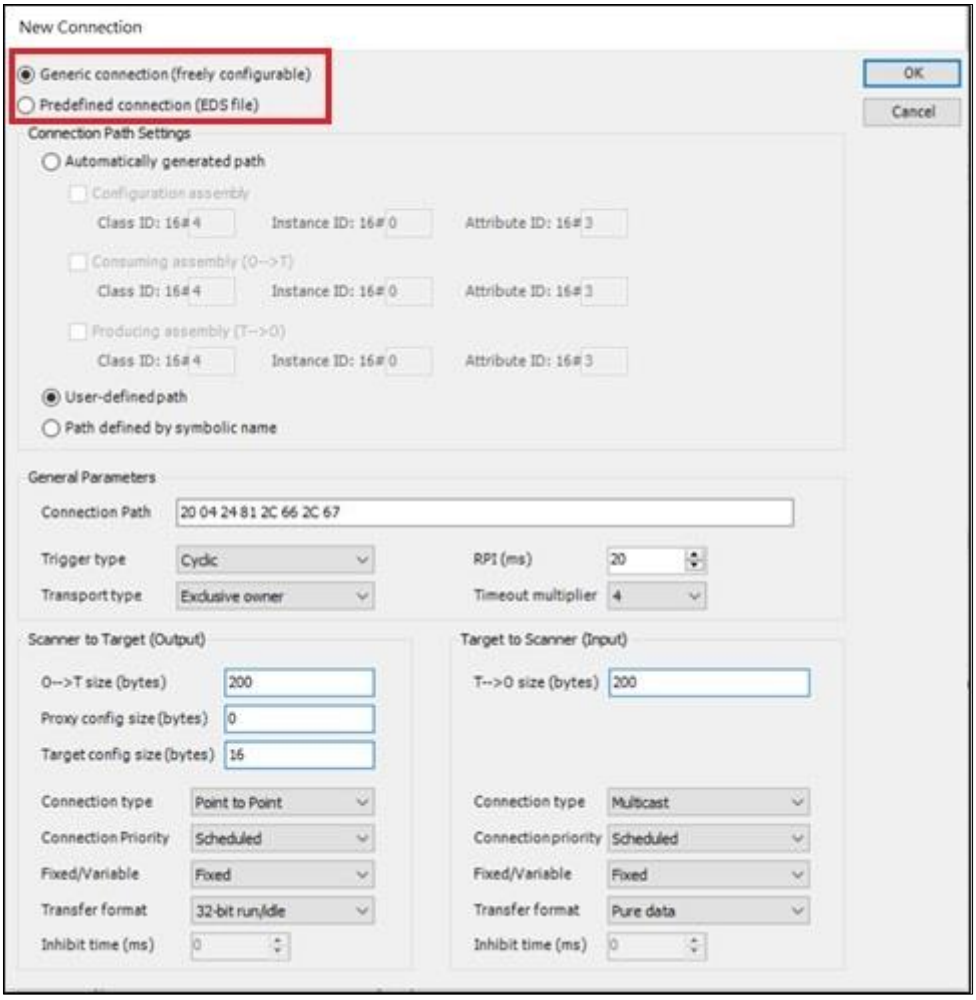


Figure 4 - 84: Universal Connection (Free configuration)

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

Project	Description
General parameters	
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.

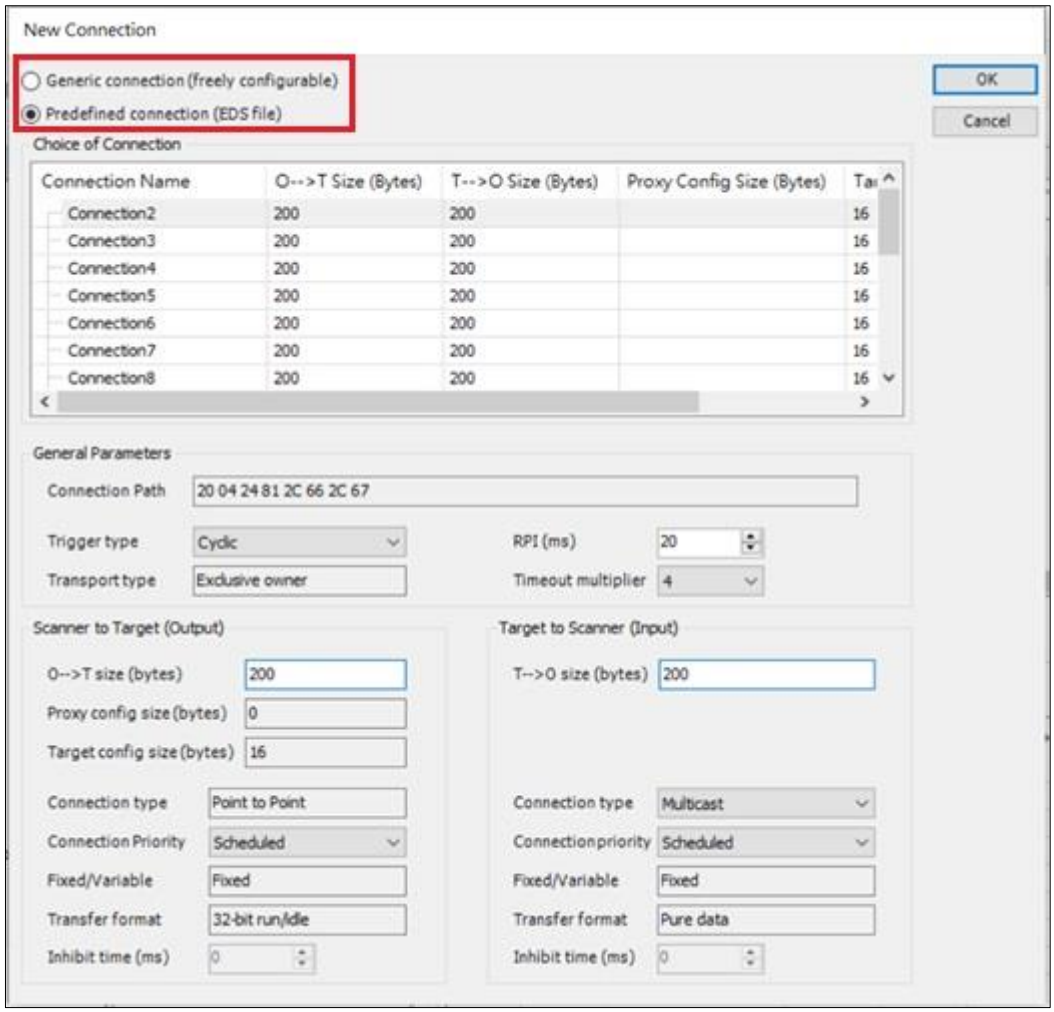


Figure 4 - 85: Predefined Connection (EDS file)

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. • 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	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.
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.

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.

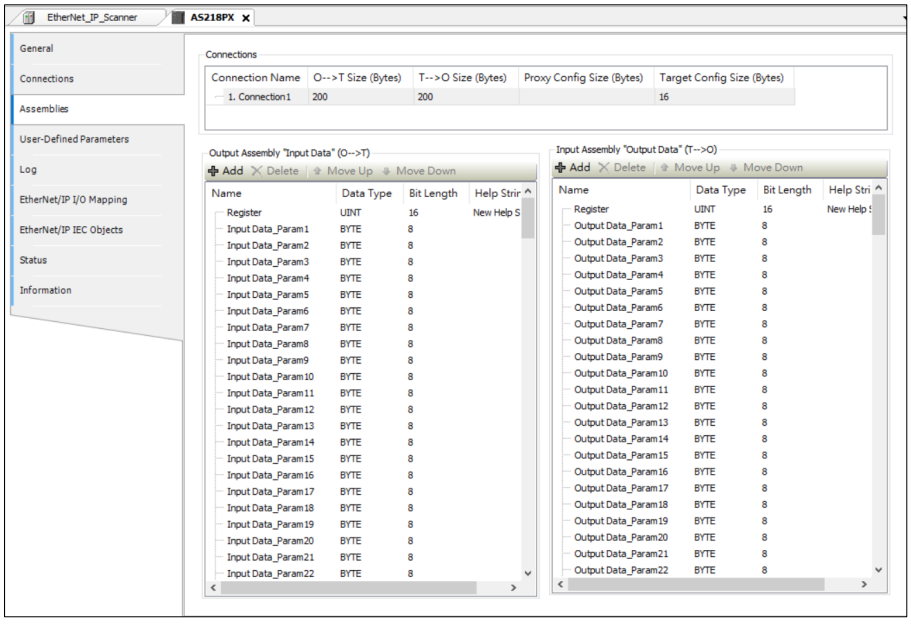


Figure 4 - 86: EtherNet/IP Remote Adapter Component Tab

- **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**

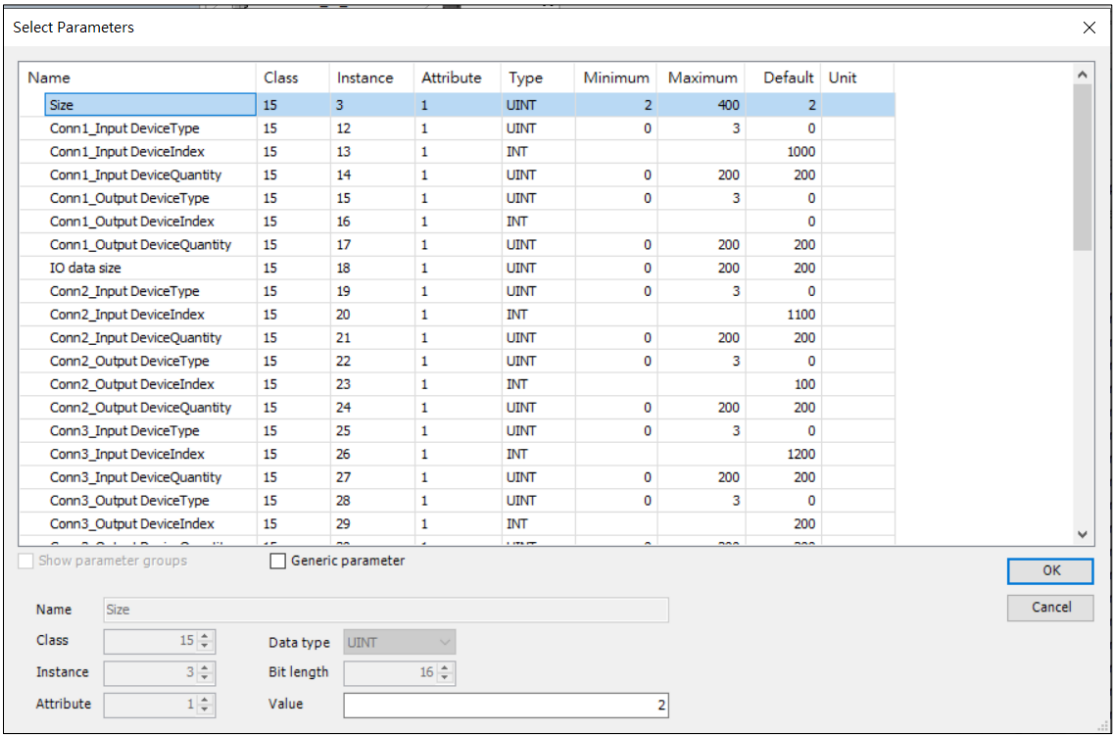


Figure 4 - 87: Select Parameters

Project	Description
Show parameter groups	• Start: This dialog box displays all parameters in the EDS file according to the group.

Project	Description
	• 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 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.

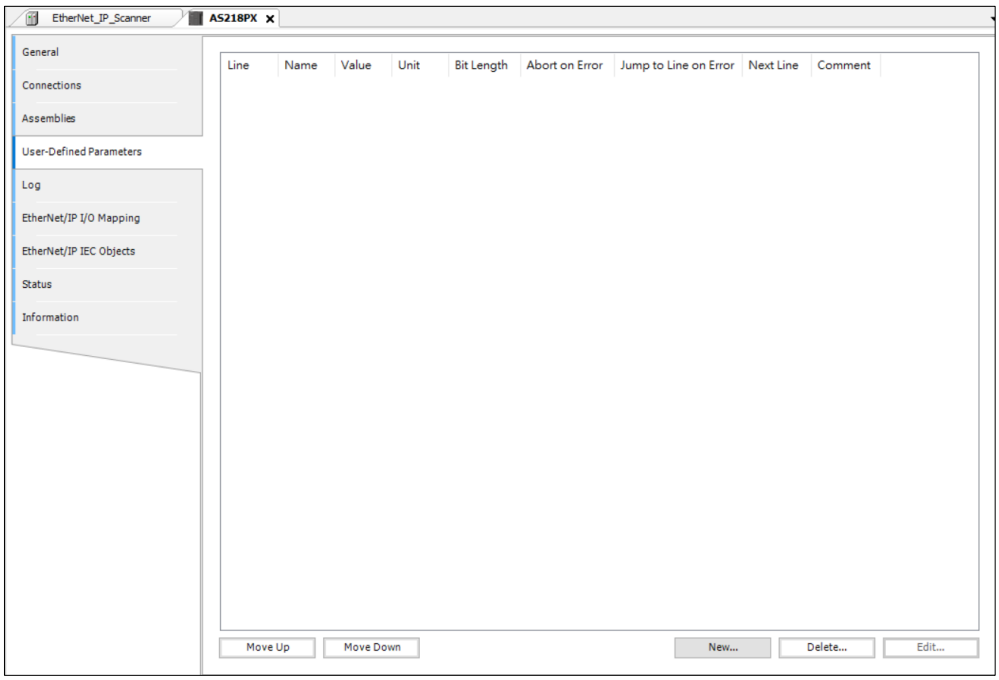


Figure 4 - 88: EtherNet/IP Remote Adapter user Parameters Tab

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

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

4.2.6.3 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*.

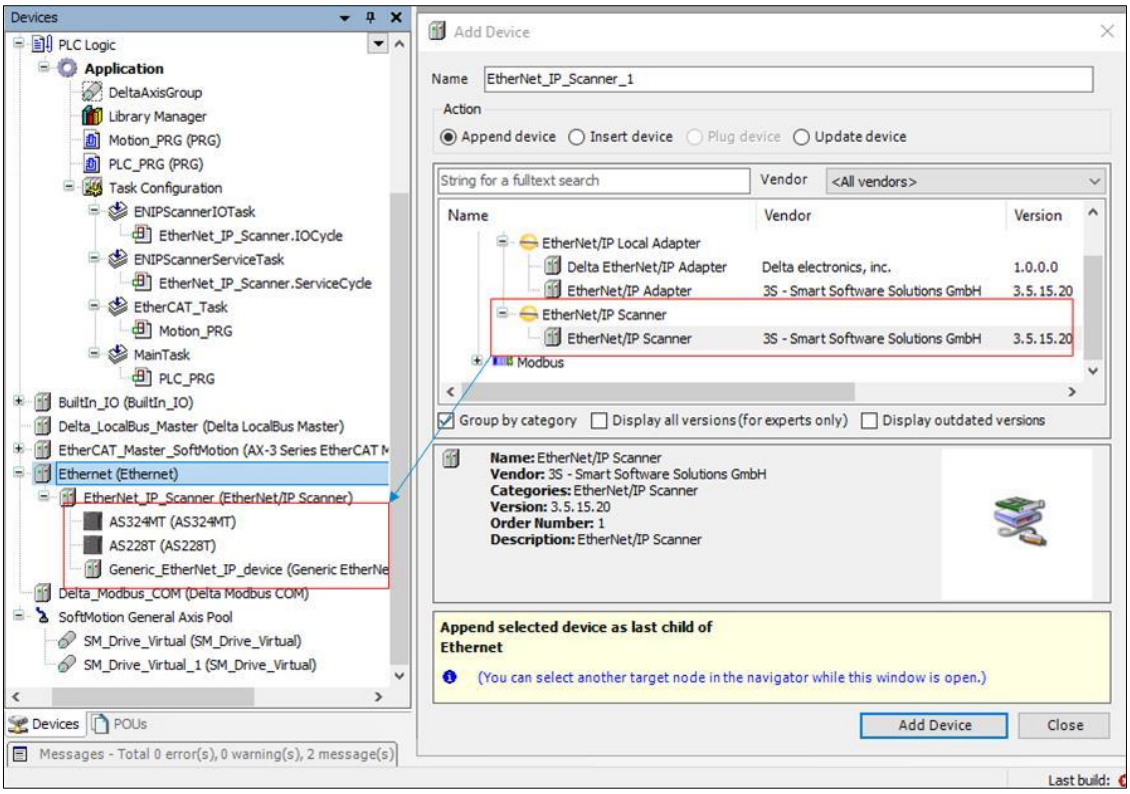


Figure 4 - 89: EtherNet/IP Local adapter

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

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

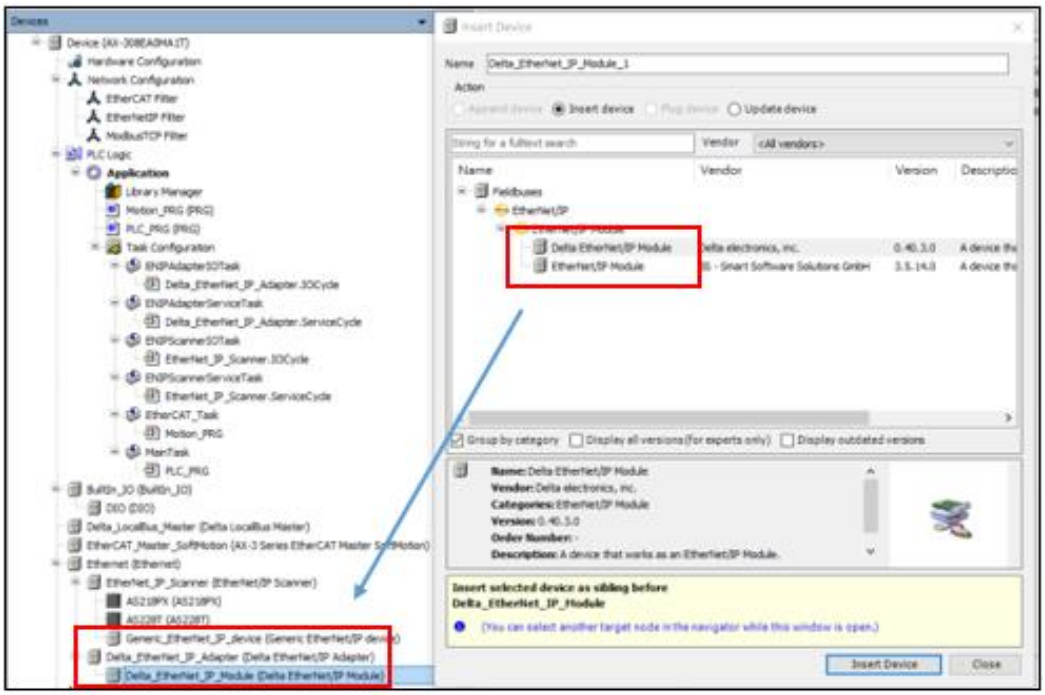


Figure 4 - 90: **AX-308E Adapter settings**

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*).

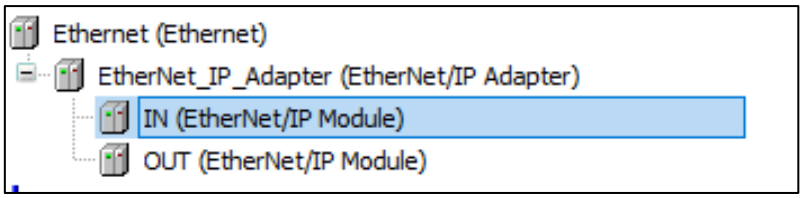


Figure 4 - 91: Add Delta EtherNet/IP Module

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

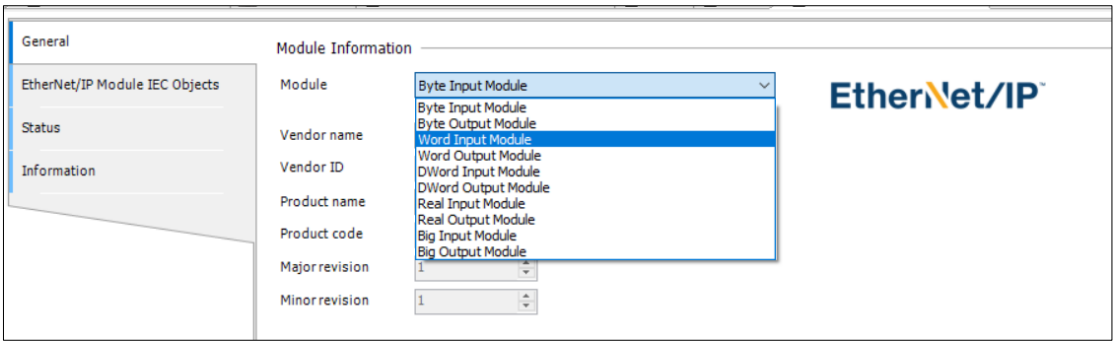


Figure 4 - 92: Word Input Module

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

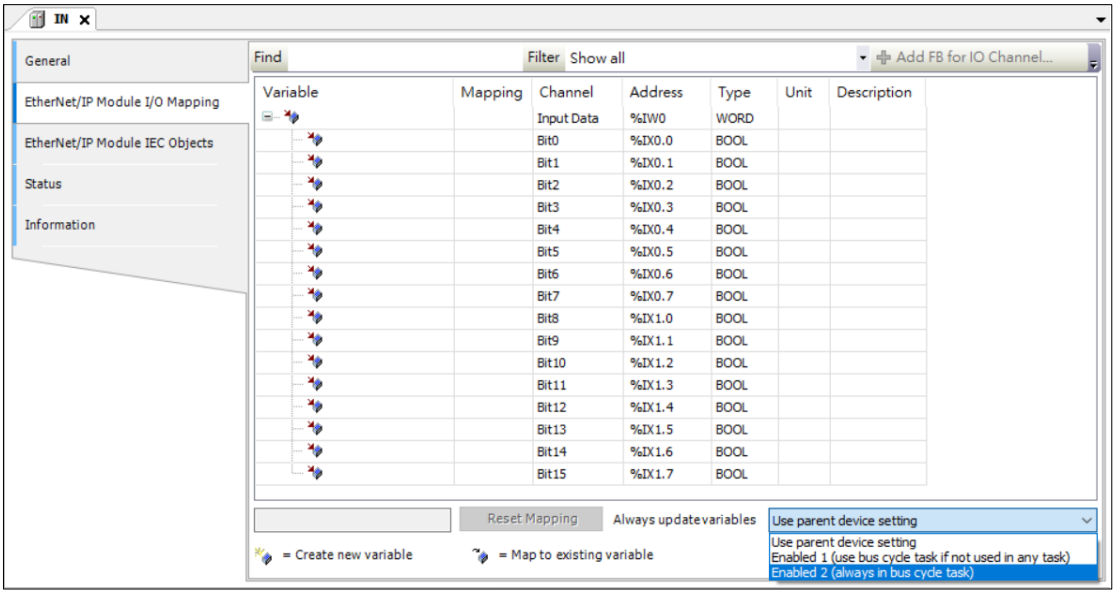


Figure 4 - 93: EtherNet/IP Module I/O Map tab

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

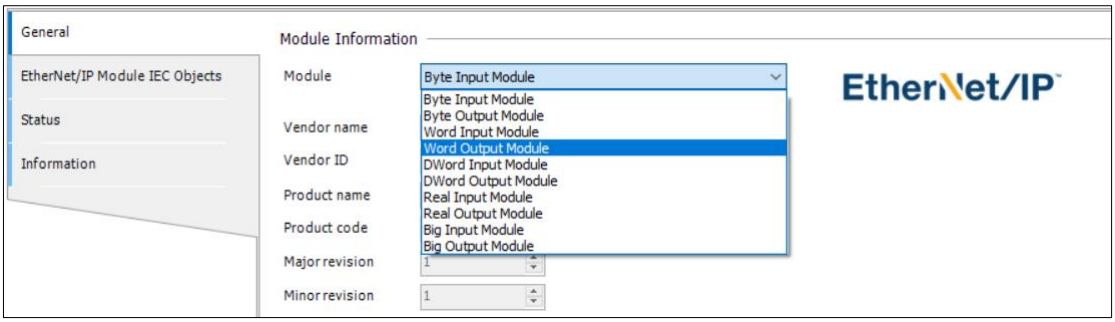


Figure 4 - 94: Word Output Module

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

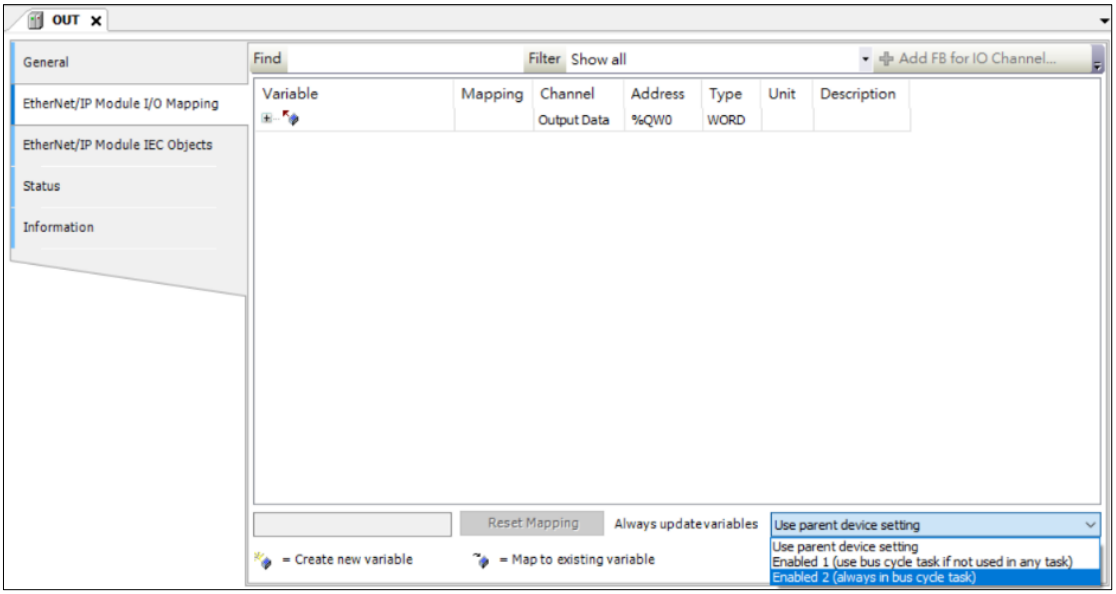


Figure 4 - 95: EtherNet/IP Module I/O Map tab

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.

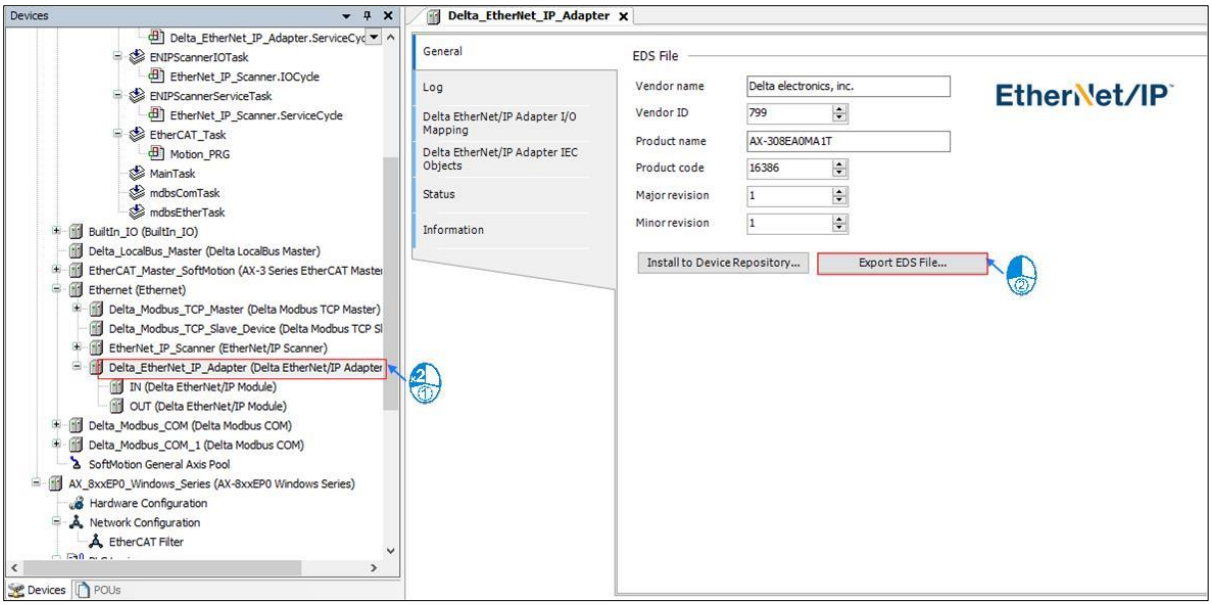


Figure 4 - 96: Export EDS file

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.

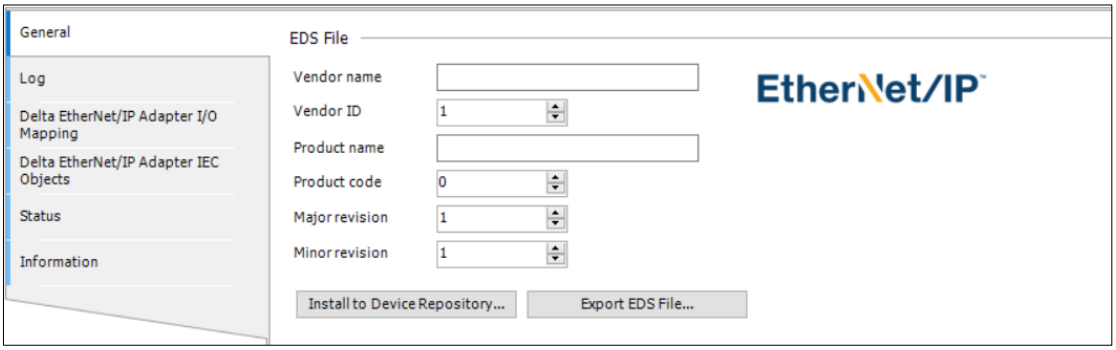


Figure 4 - 97: EtherNet/IP Local adapter general tab

Project	Description
Vendor name	Supplier name
Vendor ID	Provided by ODVA Association
Product name	EDS file name
Product Code	Product Code

Project	Description
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.

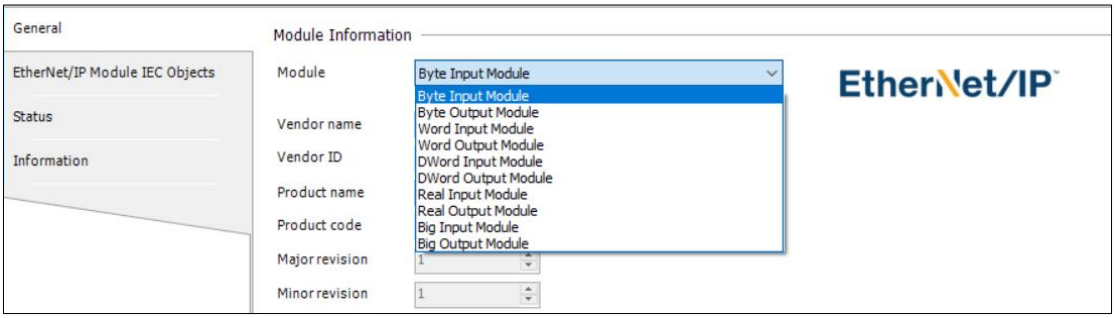


Figure 4 - 98: EtherNet/IP Module General Tab

4.2.7 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-364EL/AX-308	Description
DIO Configuration	✓	✓	AX-324 lacks a differential encoder.
SSI	-	✓	-
Counter/Timer	6 groups	8 groups	AX-324 lacks a differential encoder.
Capturer/Comparator	✓	✓	-
Pulse output shaft	-	✓	-
Interrupt	✓	✓	AX-324 lacks Channels 16 & 17.

4.2.7.1 High-speed IO setting (AX-364EL/AX-308)

4.2.7.1.1 BuiltIn IO: Hardware IO configuration

Click *BuiltIn_IO* from the device tree to configure hardware IO.

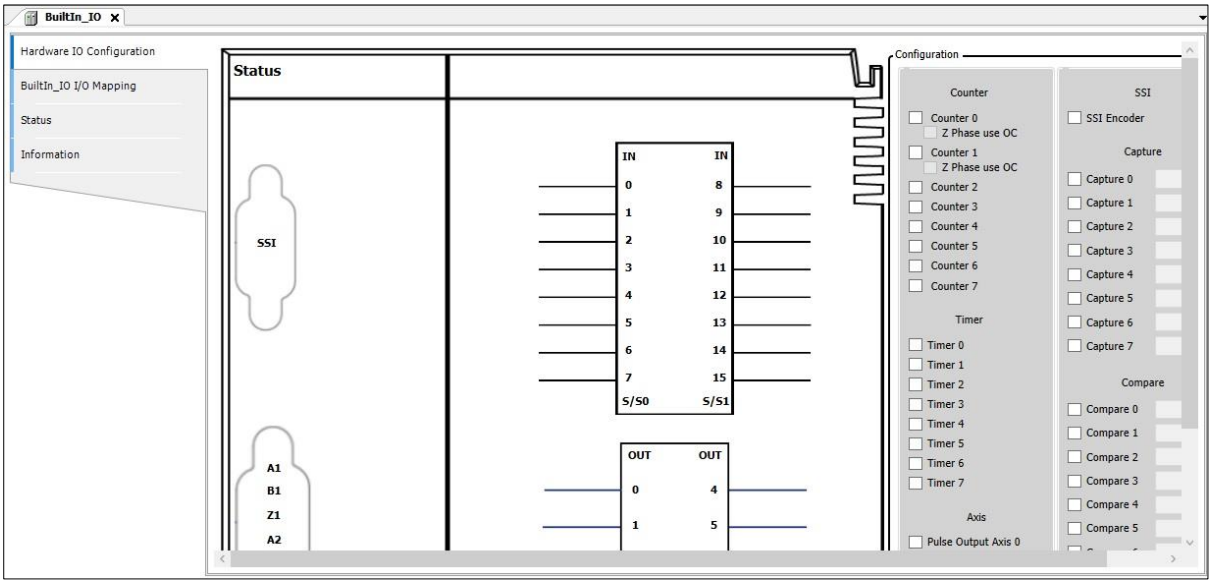


Figure 4 - 99: Hardware I/O Configuration

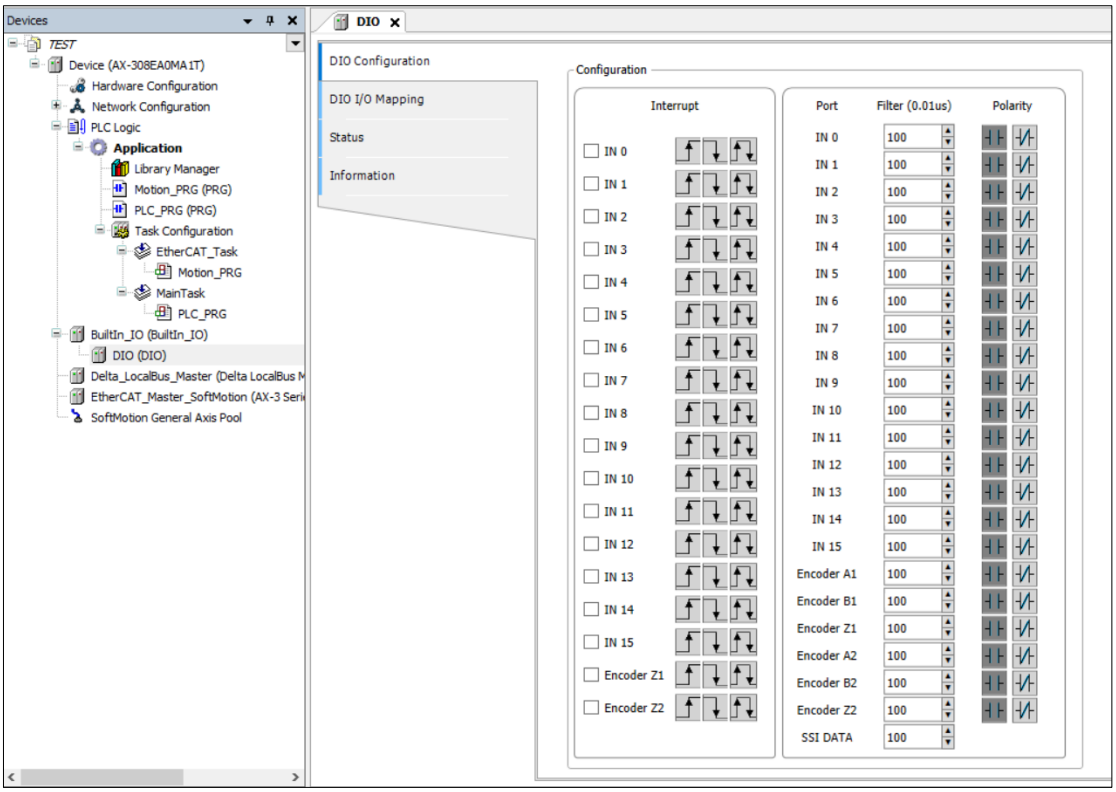


Figure 4 - 100: DIO Bookmark

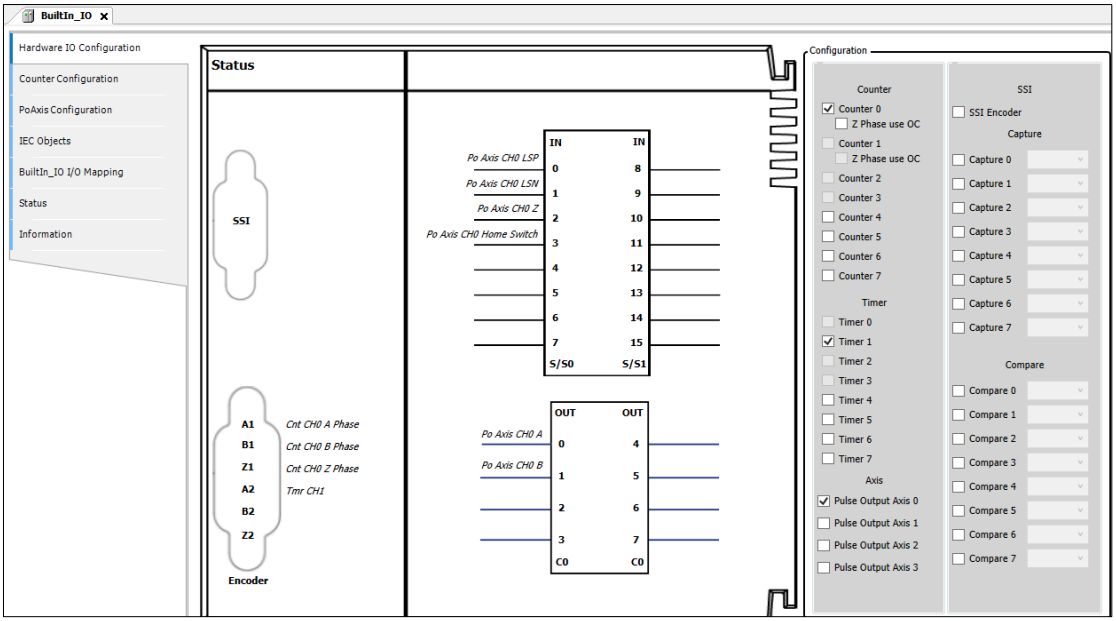


Figure 4 - 101: Counter

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.

4.2.7.1.2 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.

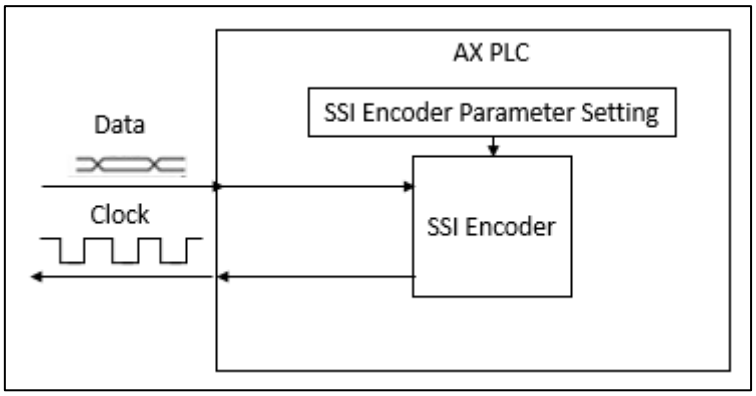


Figure 4 - 102: SSI Encoder Architecture

2. SSI Encoder function activated

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

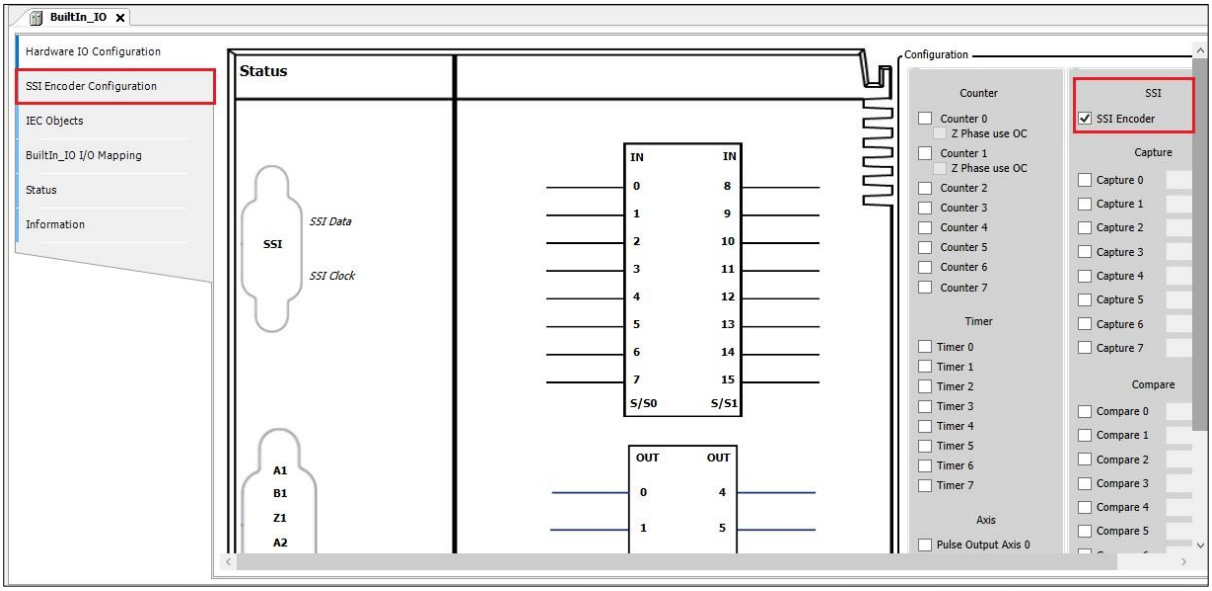


Figure 4 - 103: SSI Encoder Configuration

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

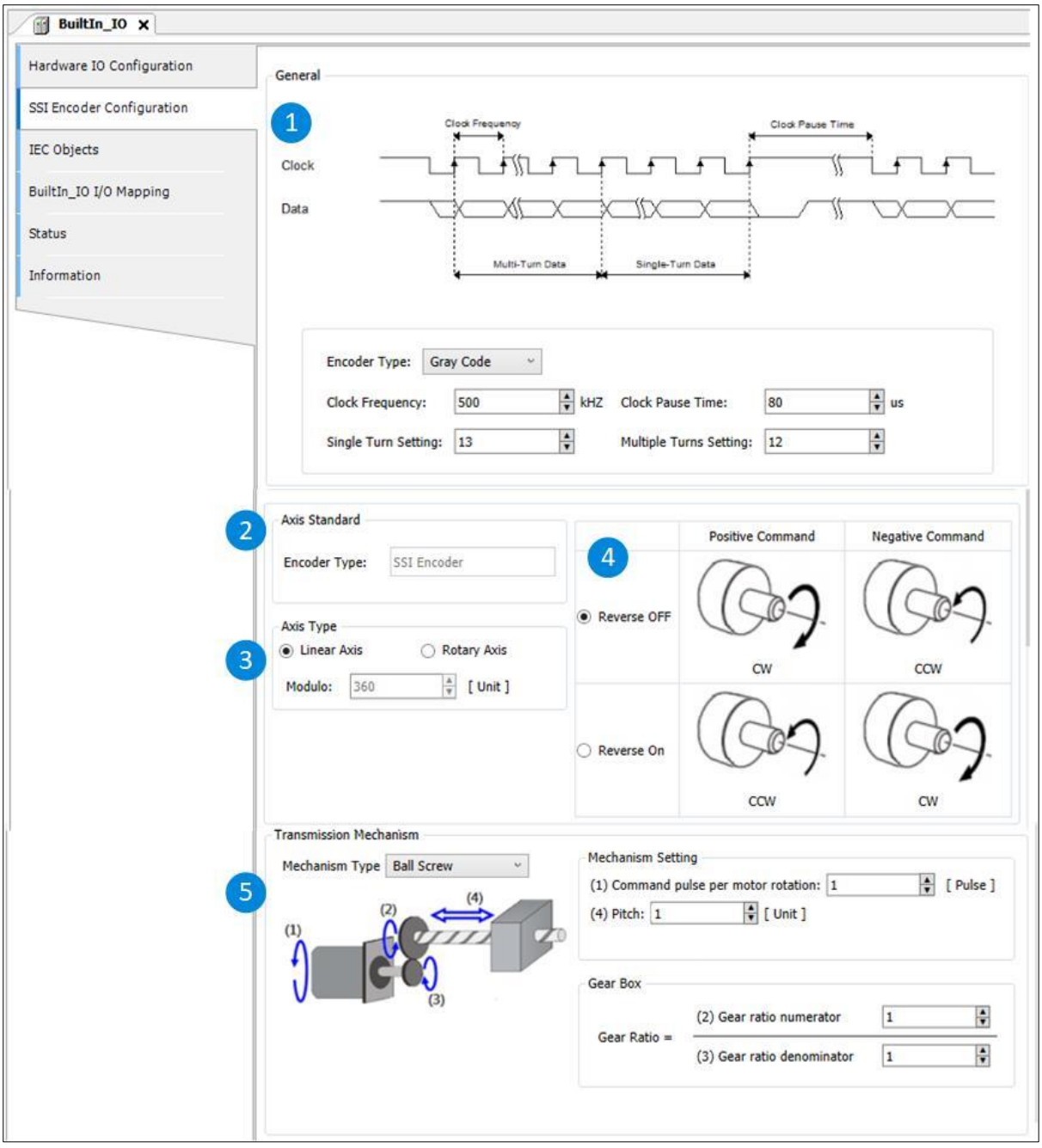


Figure 4 - 104: SSI Encoder Configuration Tab

① General

Name	Features	Set value (default value)
Encoder Type	SSI Encoder type	Gray code/ Binary code (Gray code)

Name	Features	Set value (default value)
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 (Need to use the SSI encoder data sheet as a reference)	(12)
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

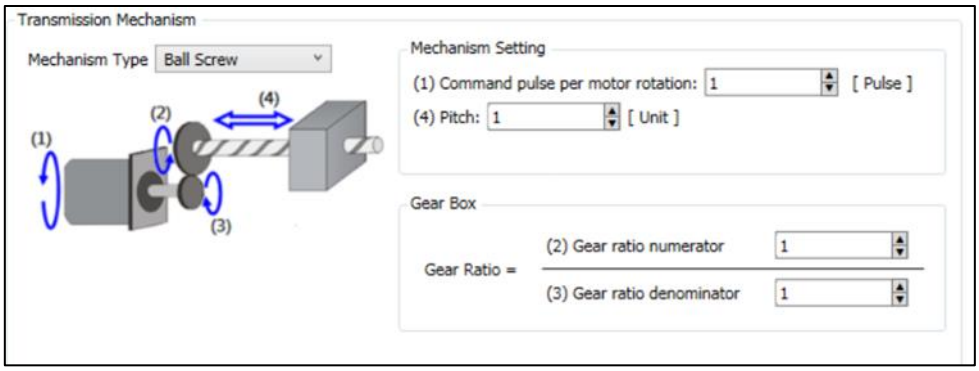


Figure 4 - 105: Transmission Mechanism – Ball screw Type

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

- Round Table

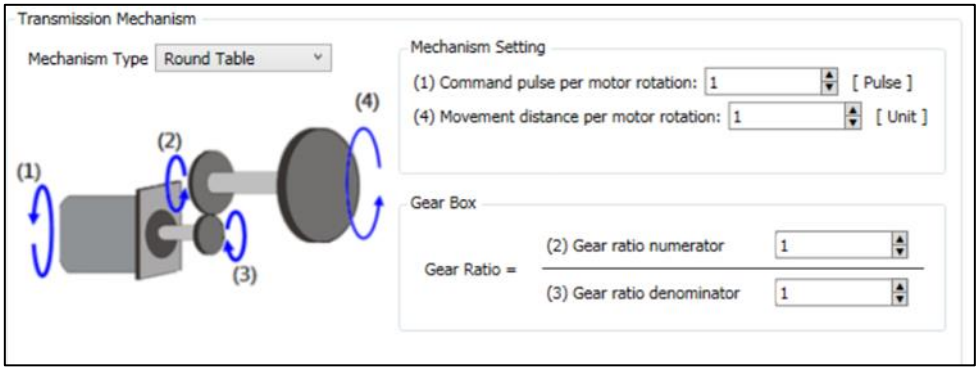


Figure 4 - 106: Transmission Mechanism – Round Table Type

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

- Belt Pully

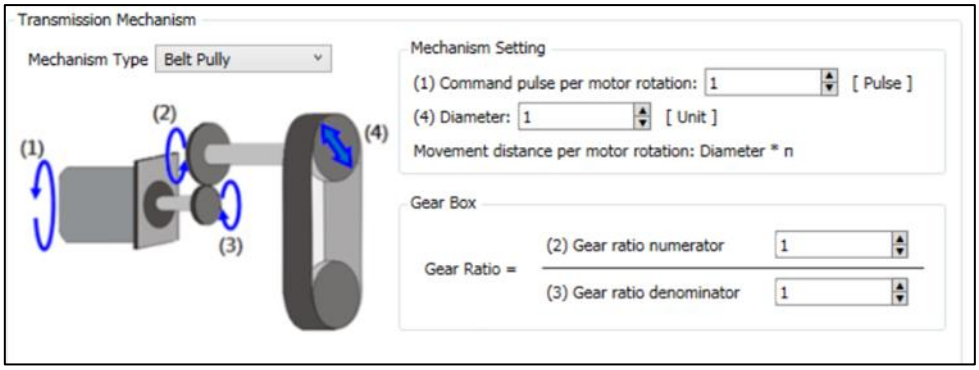


Figure 4 - 107: Transmission Mechanism – Belt Pully Type

Legend	Features
(1)	Command pulse per motor rotation
(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.*

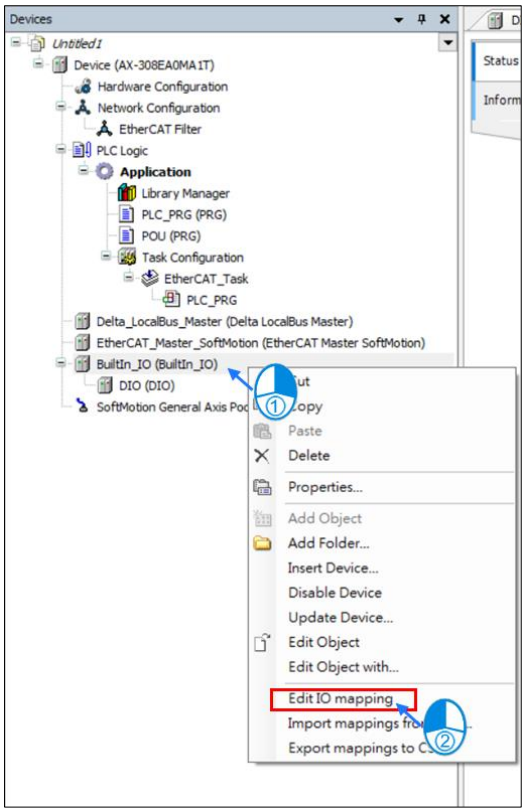


Figure 4 - 108: Edit I/O Mapping

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

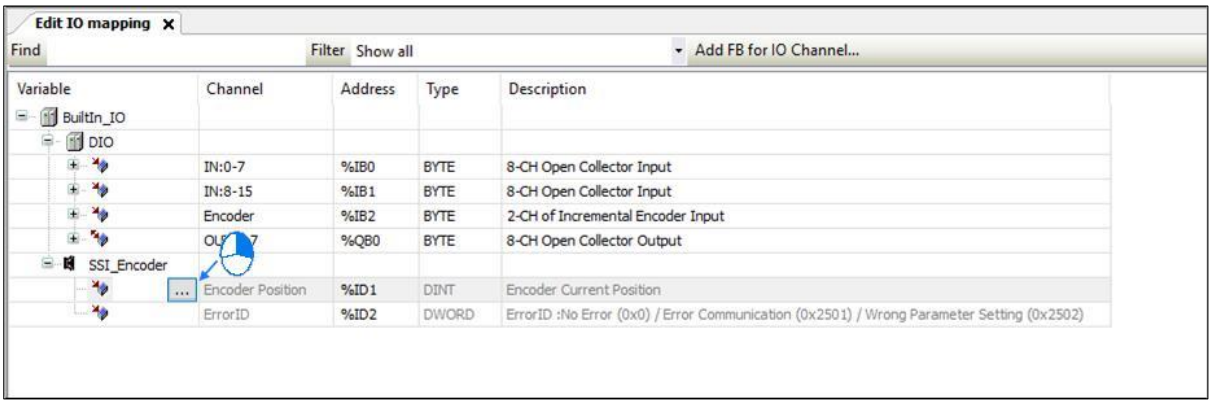


Figure 4 - 109: Adding Variable

Name	Content
Encoder Position	SSI Actual location

Name	Content
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. *BuiltIn_IO* sceen, click the *IEC Objects* tab.

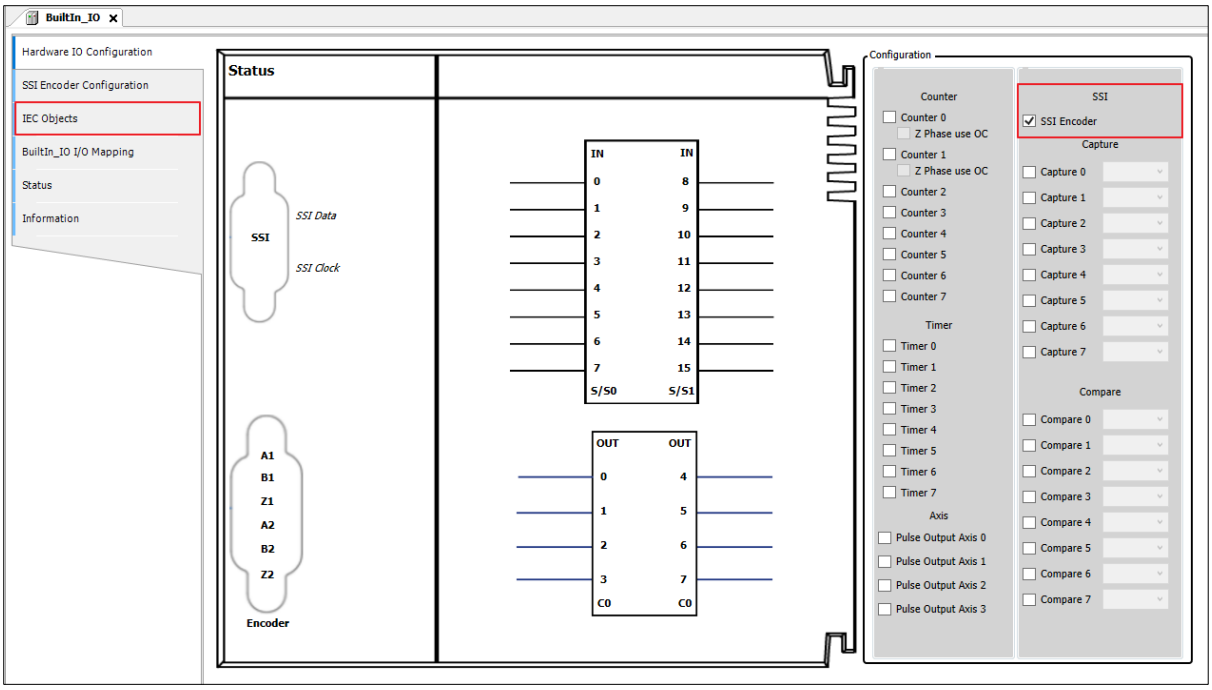


Figure 4 - 110: Hardware Configuration Window

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.

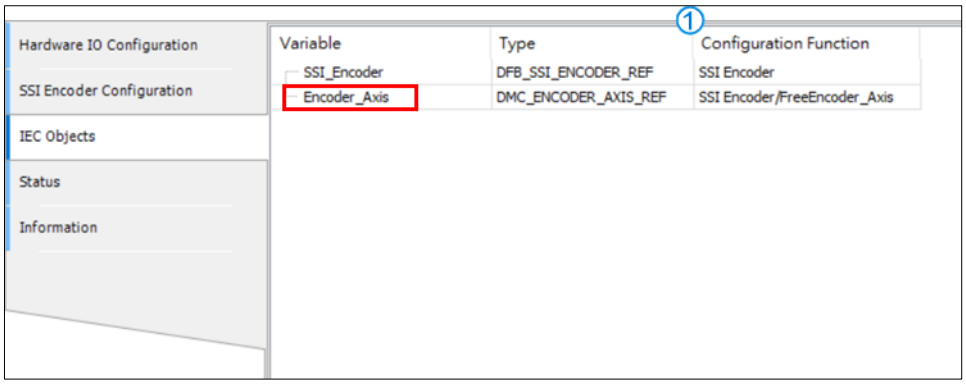


Figure 4 - 111: IEC Objects Window

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.

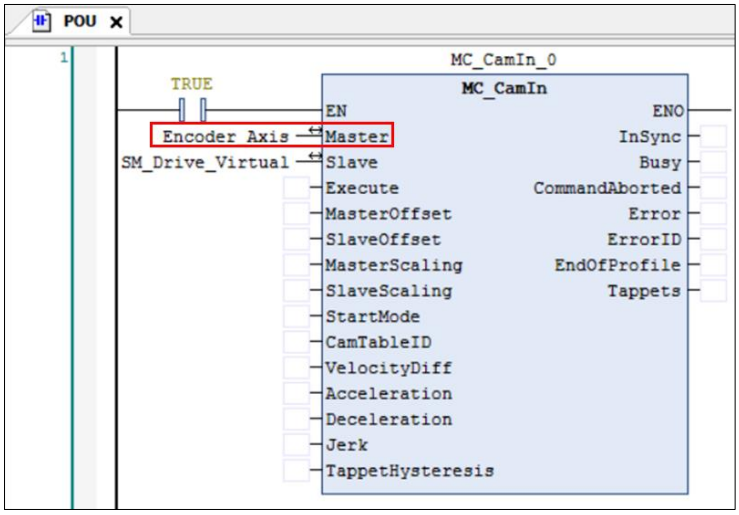


Figure 4 - 112: Encoder Axis

4.2.7.1.3 BuiltIn IO: Counter/timer configuration

The connection between AX-3 series 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.

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)

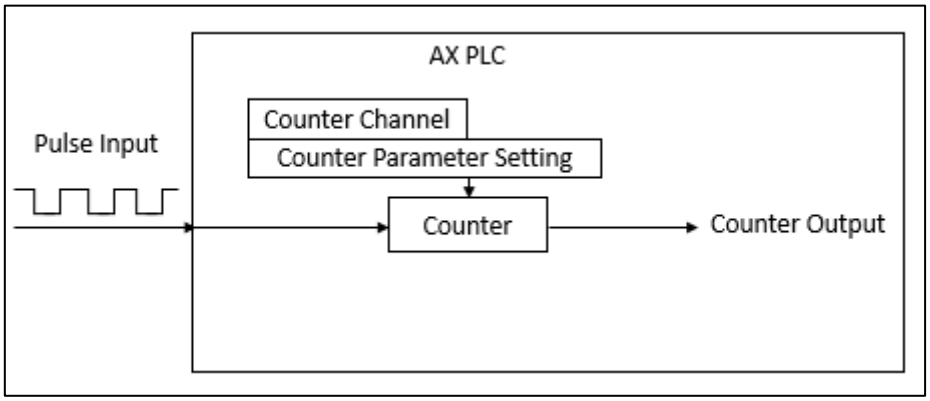


Figure 4 - 113: High-Speed Counter

- 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*.

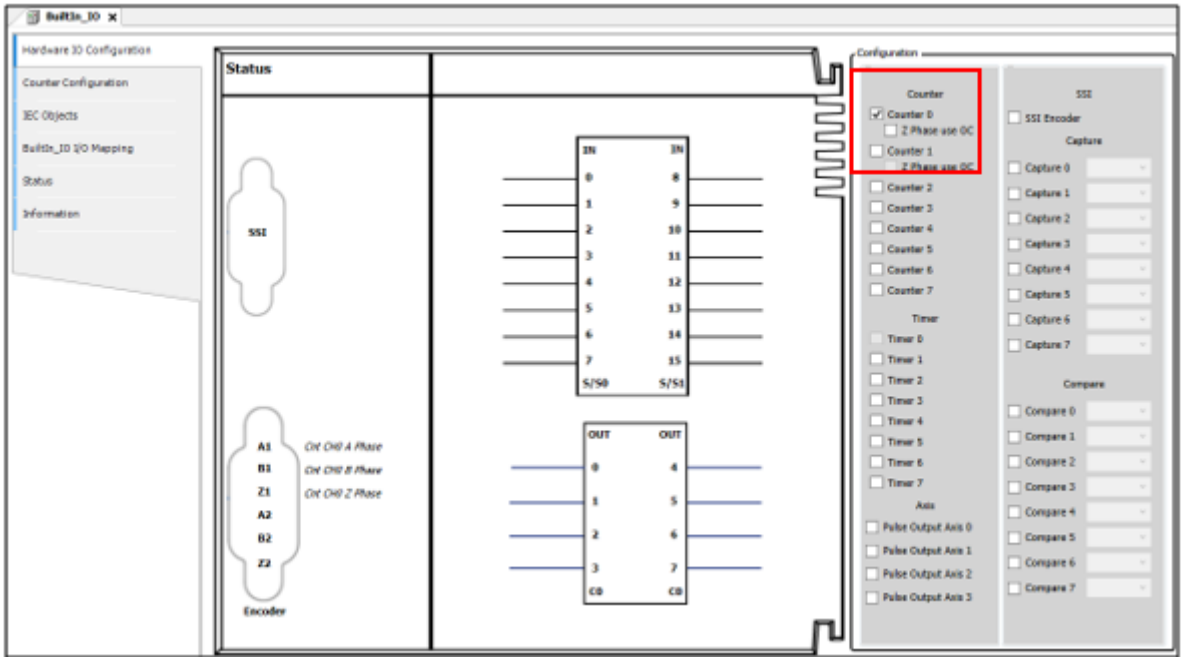


Figure 4 - 114: Activating Counter Configuration

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

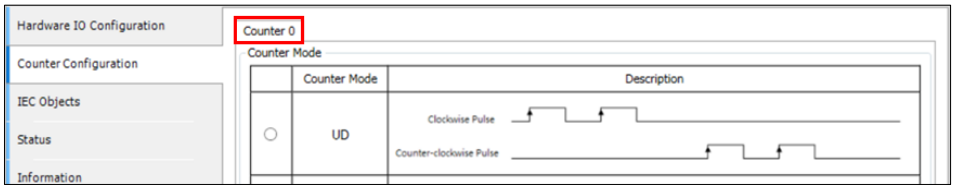


Figure 4 - 115: Counter Configuration Window

3. Counter related settings can be set in the *Counter Configuration* screen. The following explains the settings:

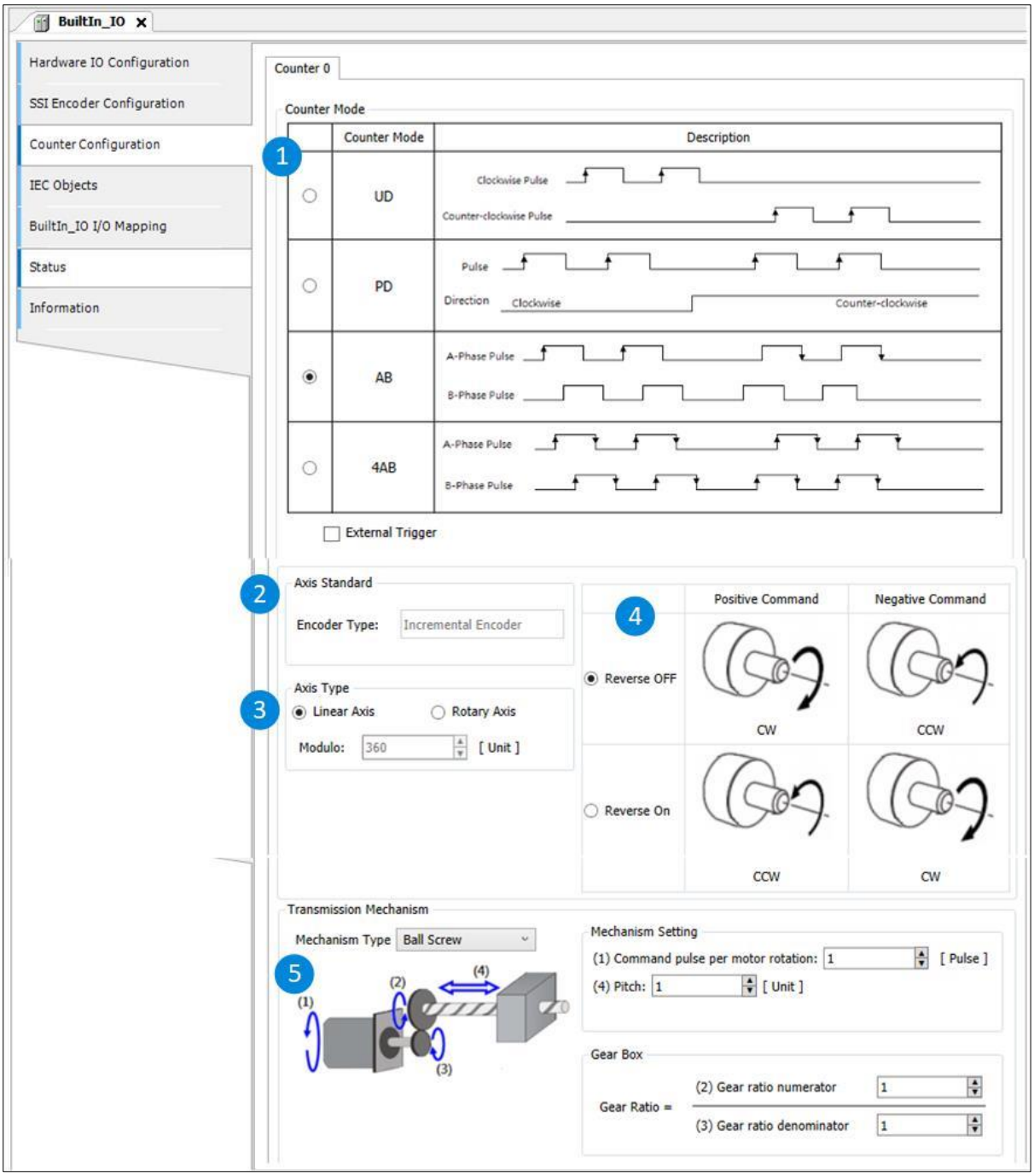


Figure 4 - 116: Counter Configuration Setting

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.*

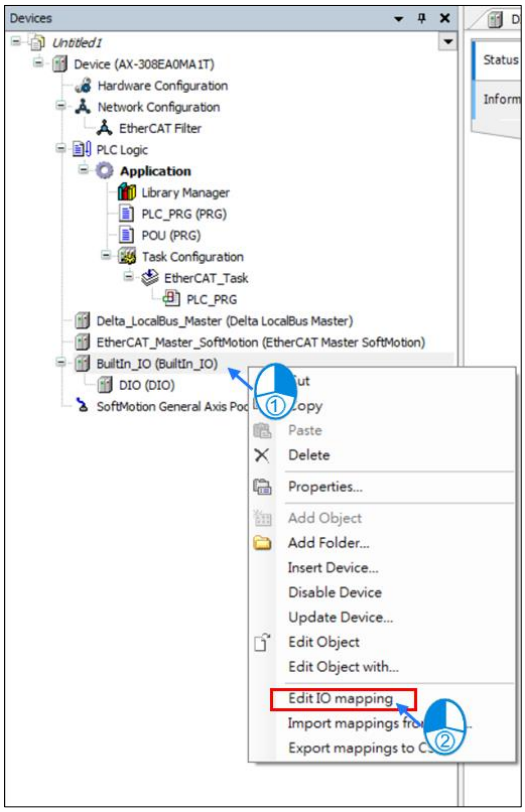


Figure 4 - 117: Setting Counter Mapping Variable

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

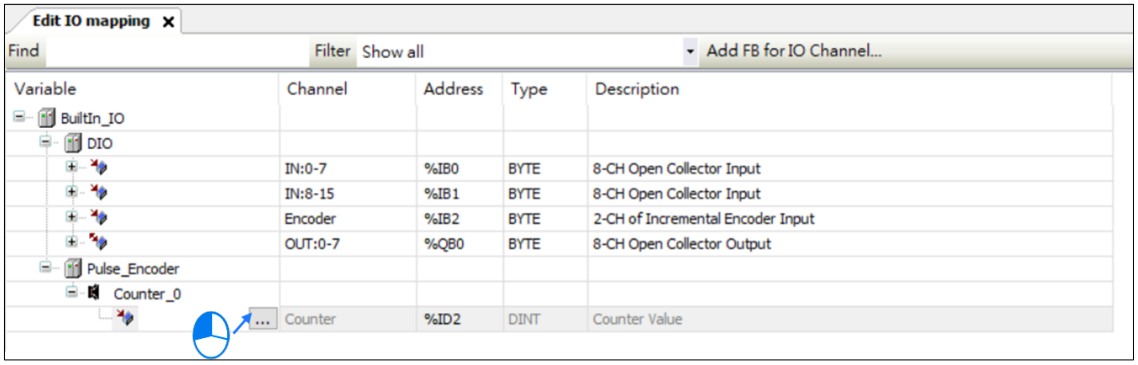


Figure 4 - 118: Edit I/O Mapping

- 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.

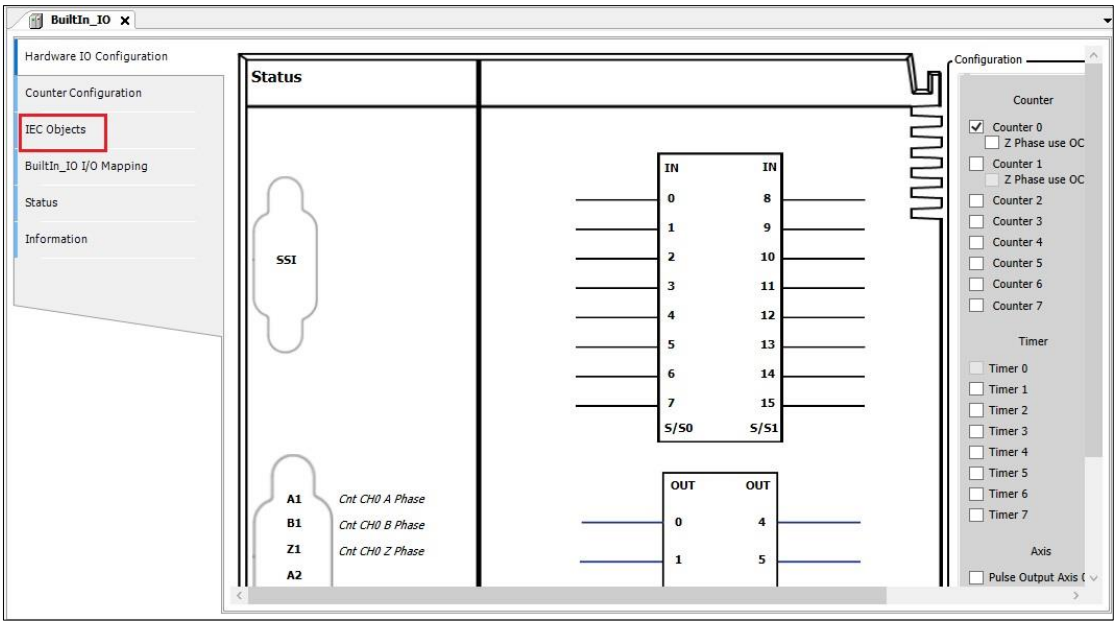


Figure 4 - 119: IEC Window

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.

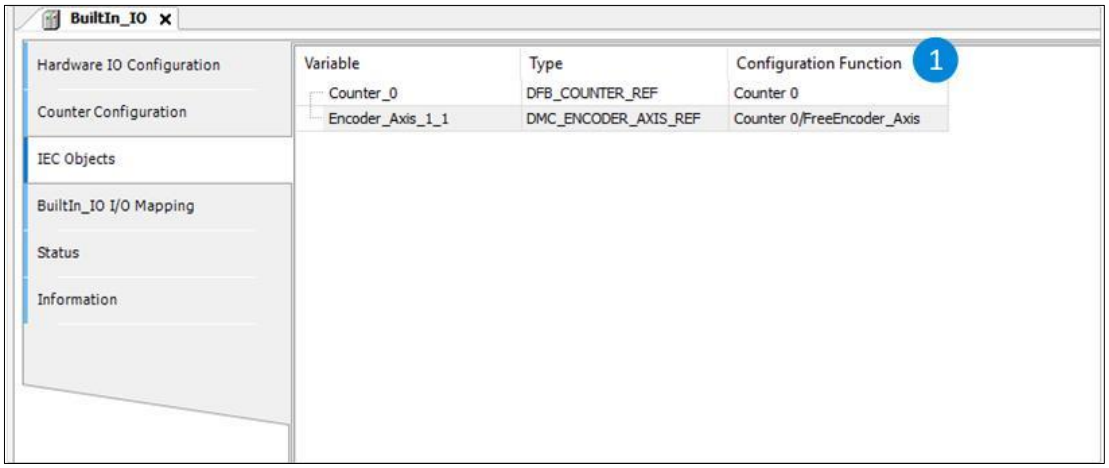


Figure 4 - 120: BuiltIn_IO – IEC Objects Tab

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.

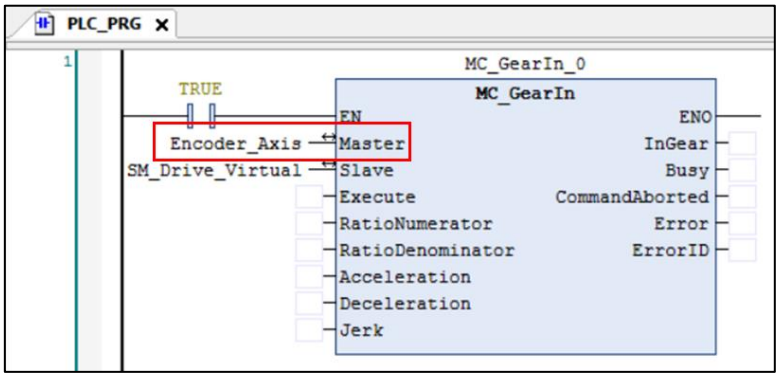


Figure 4 - 121: Encoder Axis

- High-speed timing (Tmr)

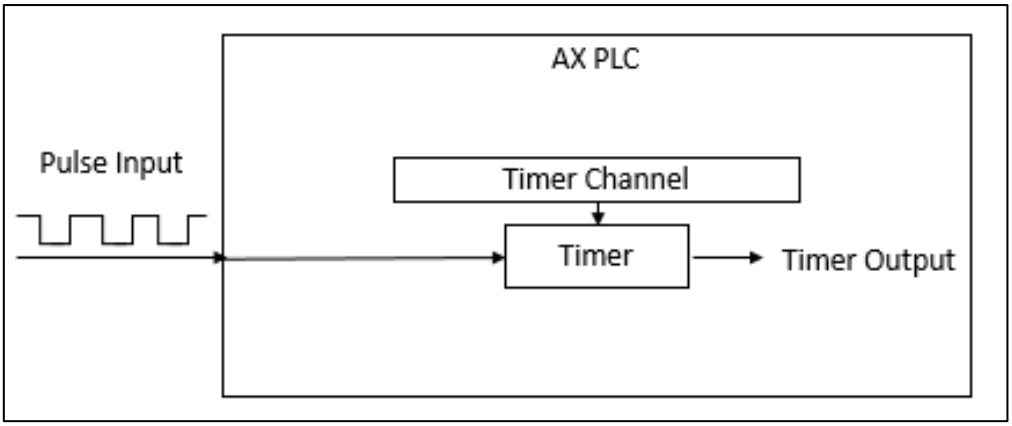


Figure 4 - 122: High-speed Timing

- 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 0.1μs.

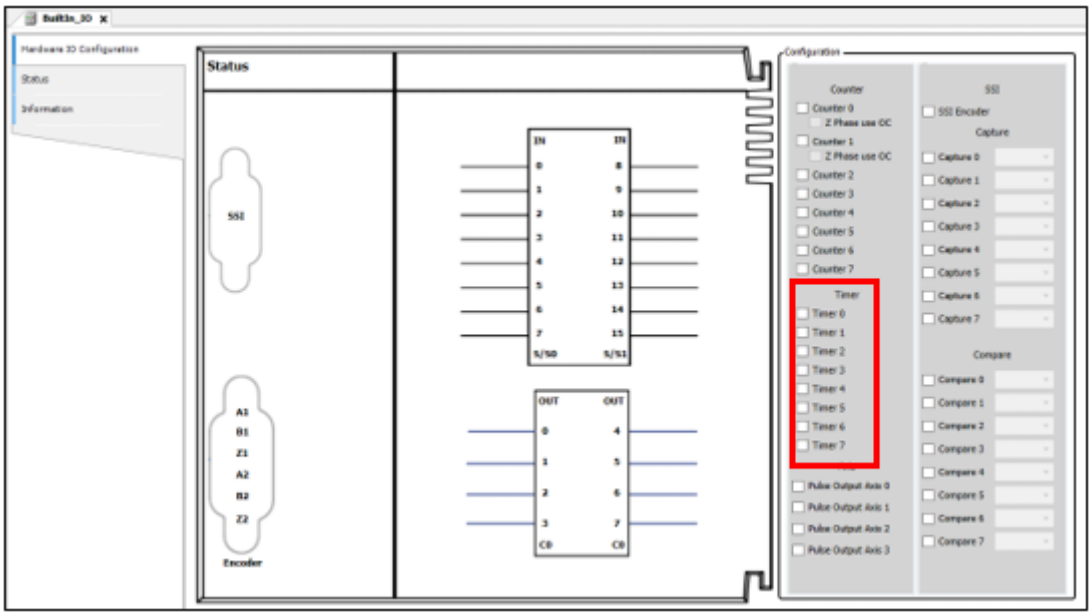


Figure 4 - 123: Timer

- Set Timer mapping variables

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

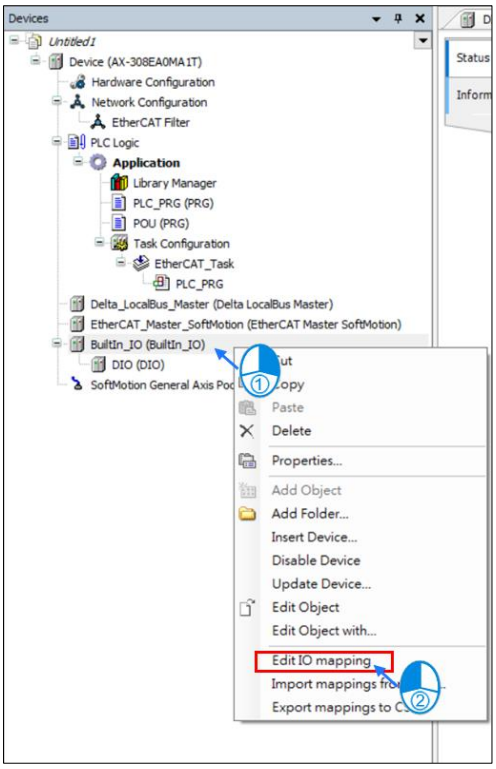


Figure 4 - 124: Edit I/O Mapping

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

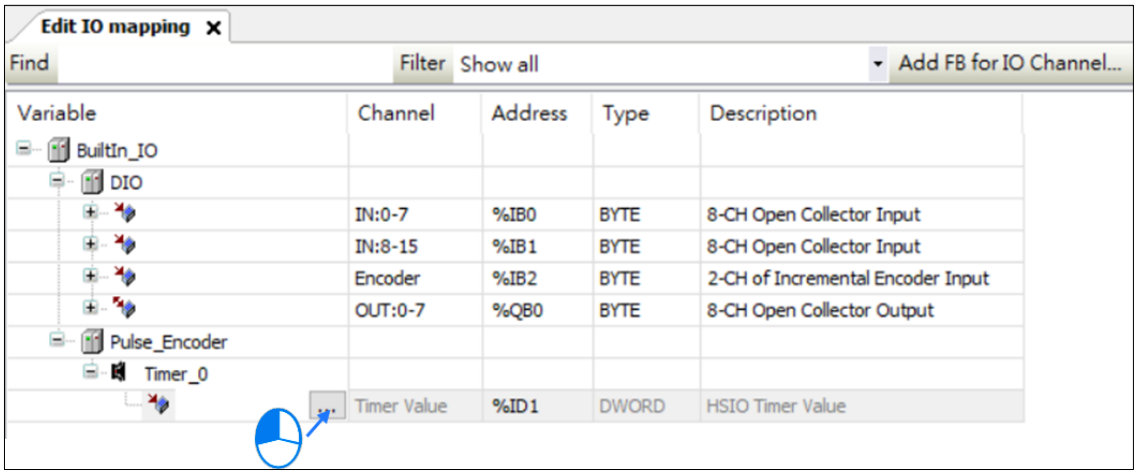


Figure 4 - 125: Set timer mapping variables

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.

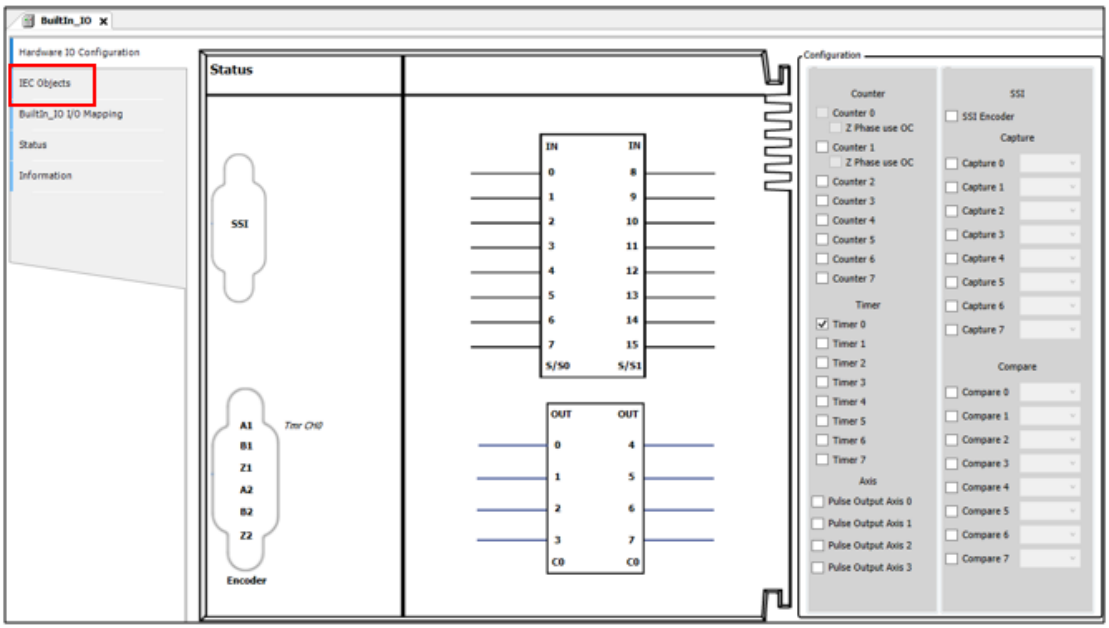


Figure 4 - 126: IEC Object window

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.

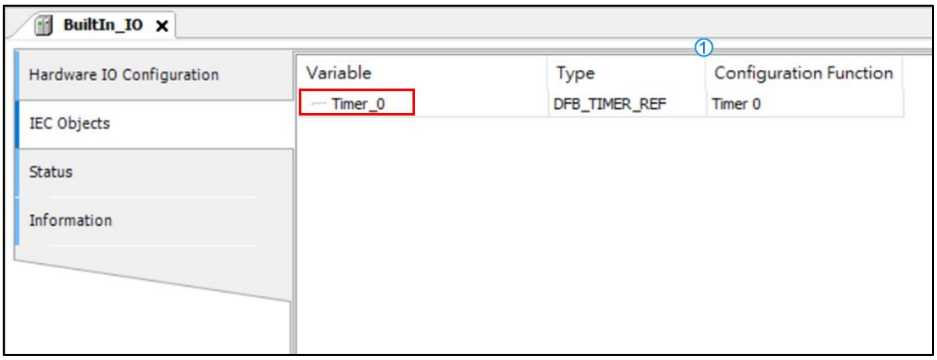


Figure 4 - 127: Setting timer

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

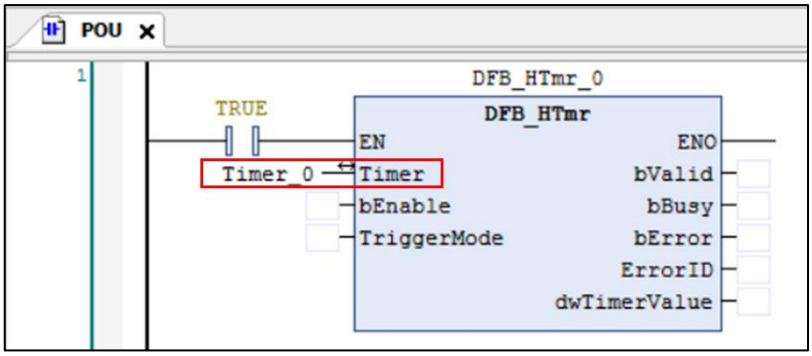


Figure 4 - 128: Entering Axis name

4.2.7.1.4 BuiltIn IO: Po axis configuration

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

- Axis Function start

1. In the *BuiltIn_IO* screen, *Axis* has 4 axis pulse wave outputs, click one of them to start.

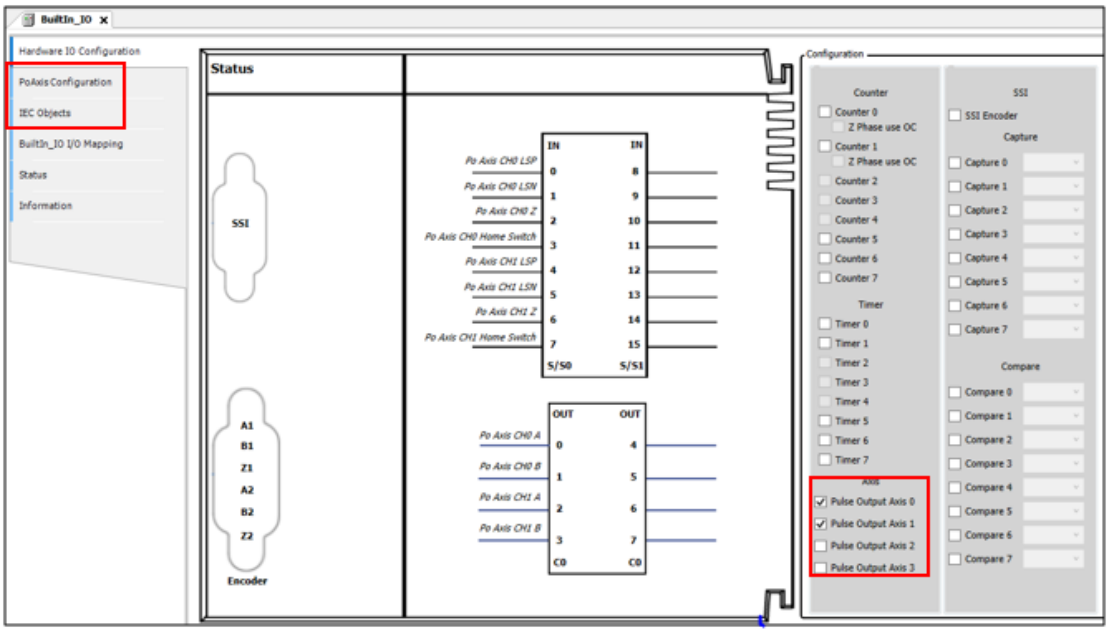


Figure 4 - 129: Selecting Pulse Output Axis

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

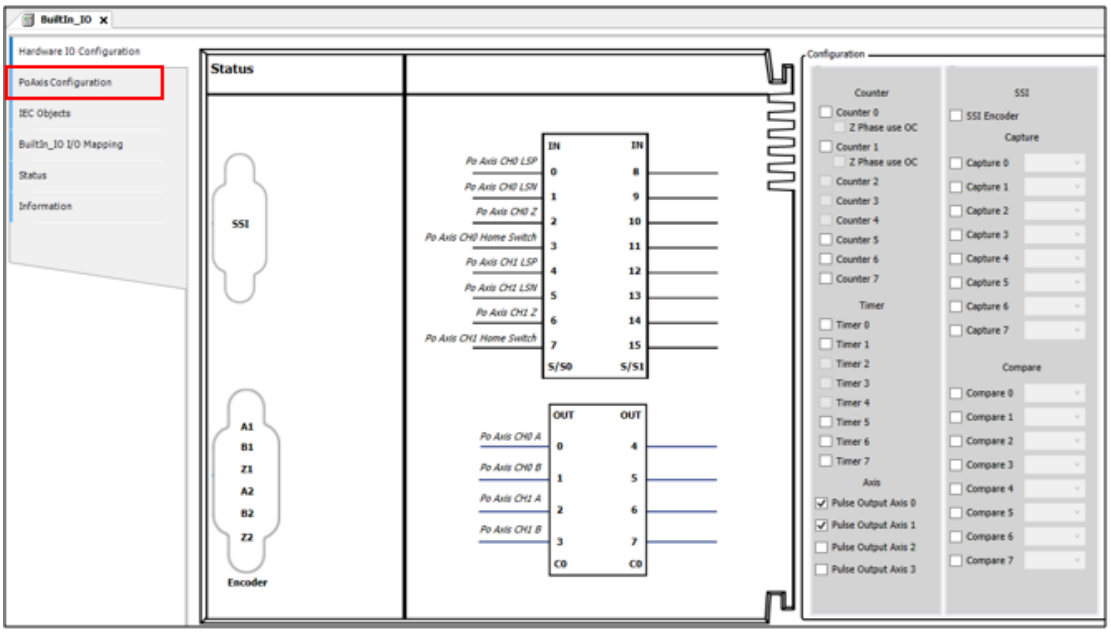


Figure 4 - 130: PoAxis Configuration

3. Click the *Axis 0* tab after the *PoAxis Configuration* setting.

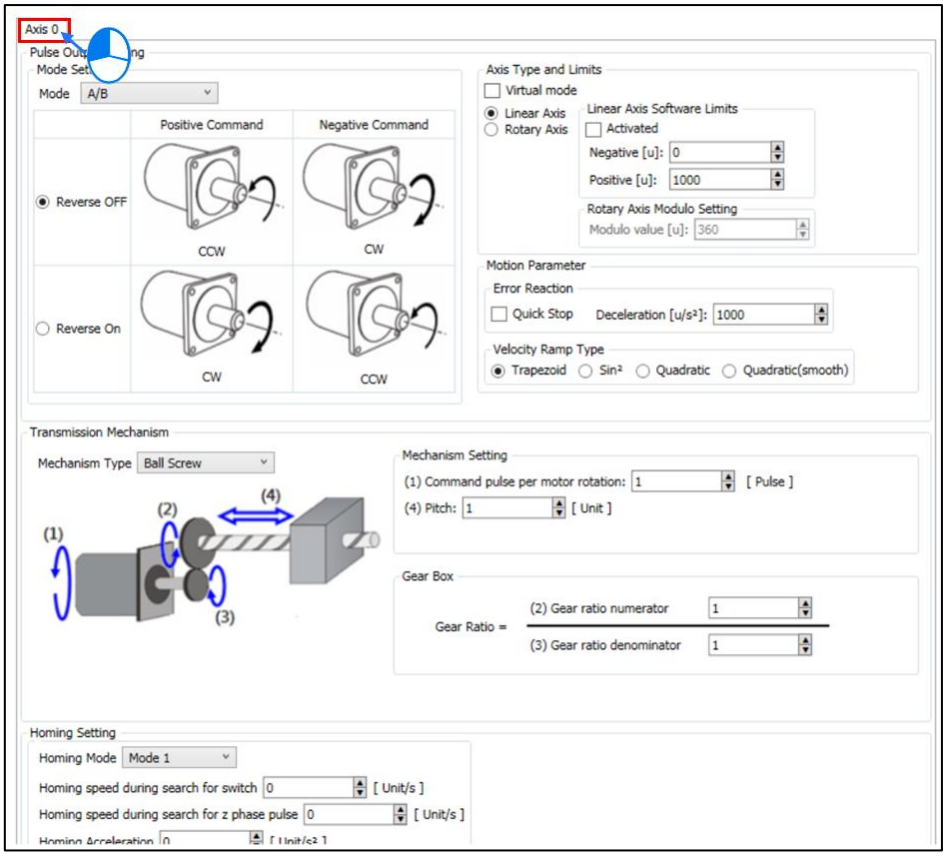


Figure 4 - 131: Axis0 Tab

4. The related settings of Axis can be set in the *Pulse Output Setting* screen:

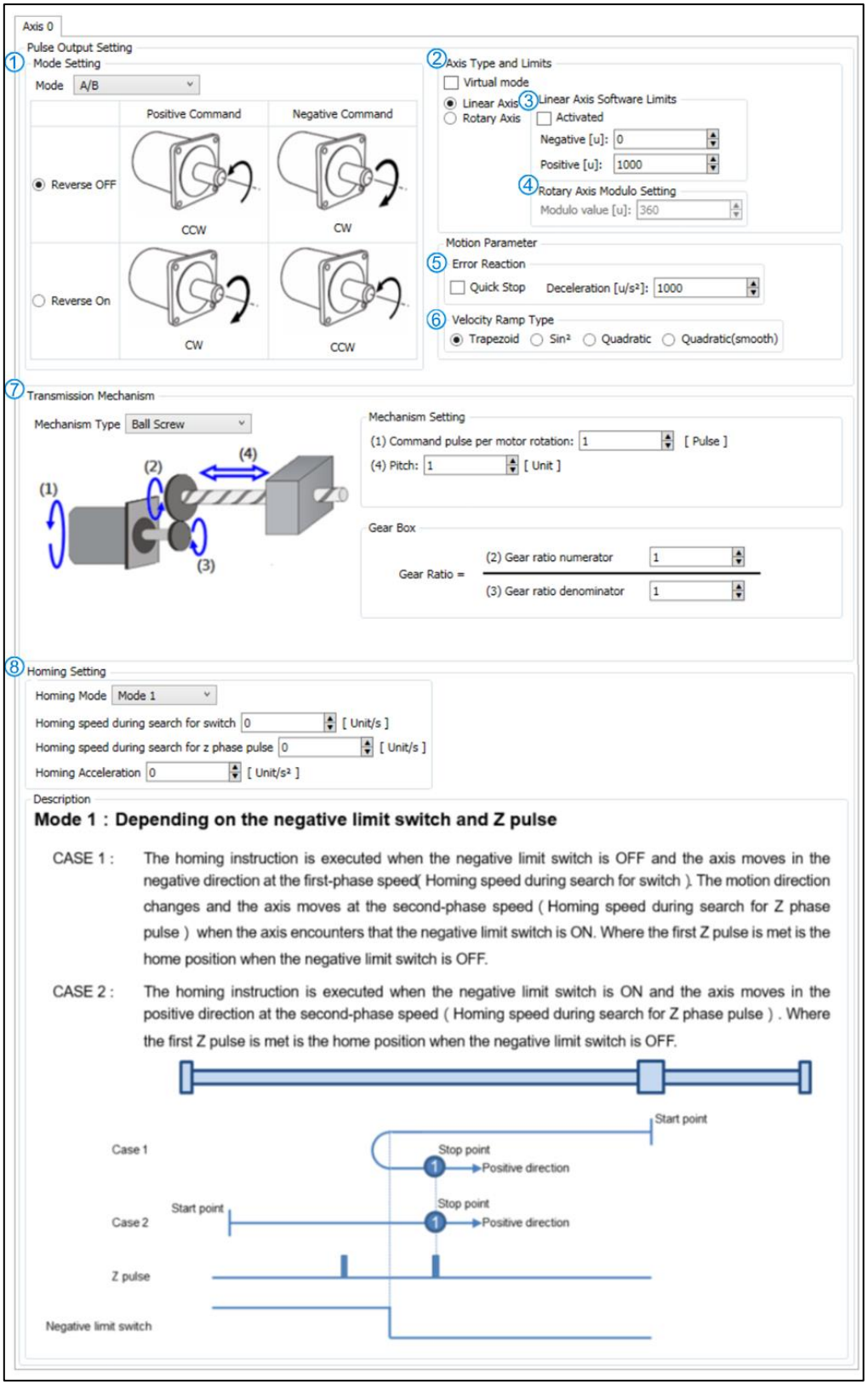


Figure 4 - 132: Mode Setting

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)
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.

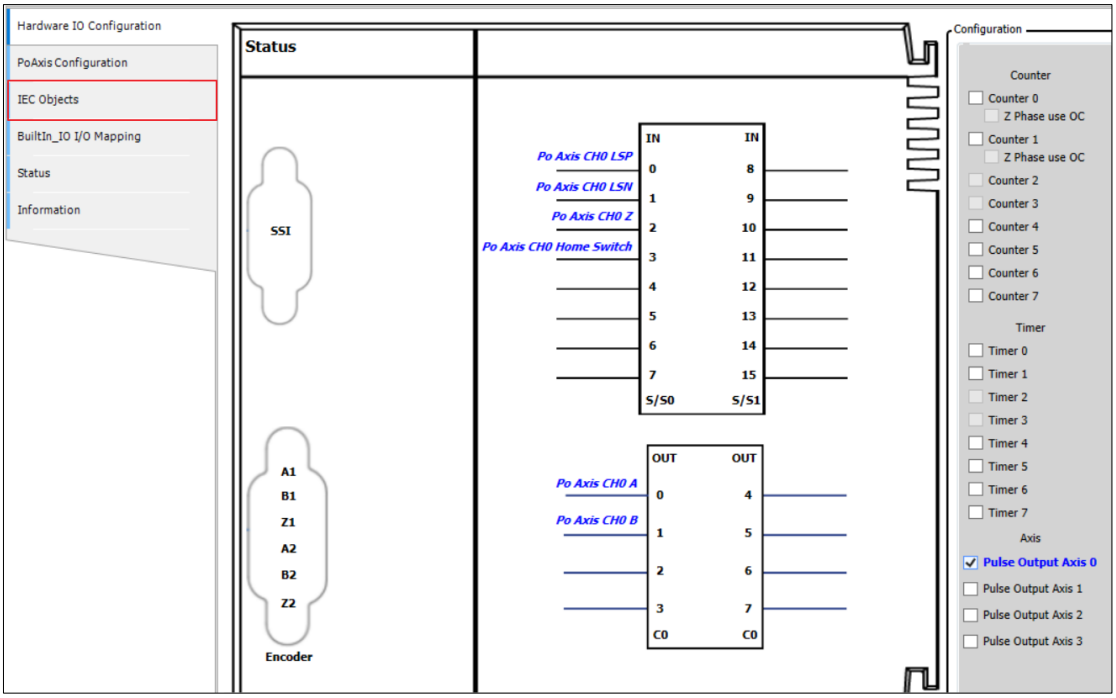


Figure 4 - 133: IEC Object window

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.

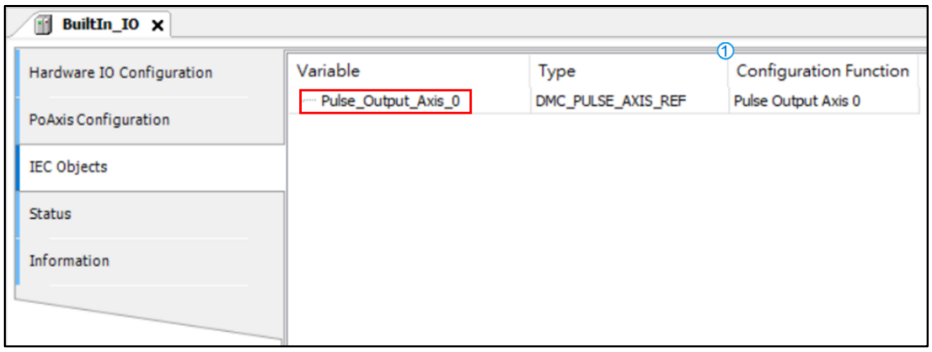


Figure 4 - 134: Selecting Pulse_Output_Axis_0

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

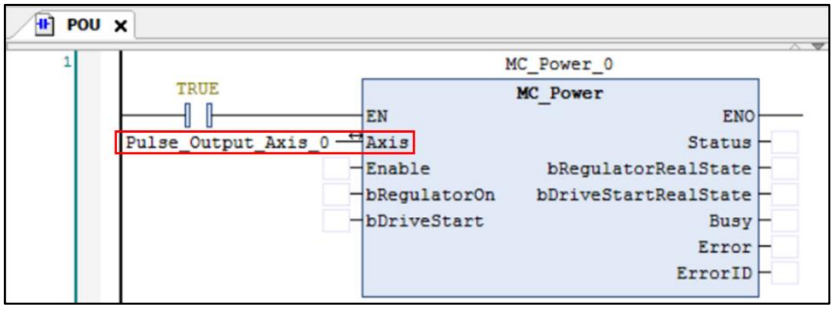


Figure 4 - 135: Entering axis name

4.2.7.1.5 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

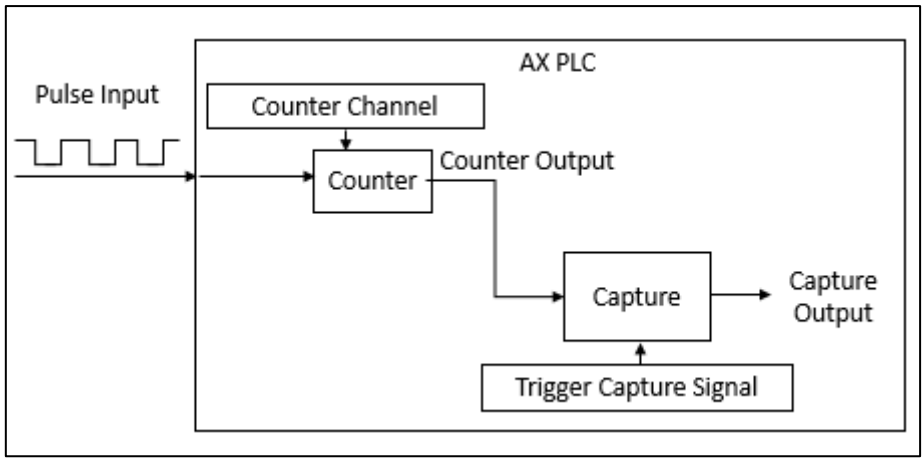


Figure 4 - 136: High-Speed Counter Capture

- Capture function is activated.

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

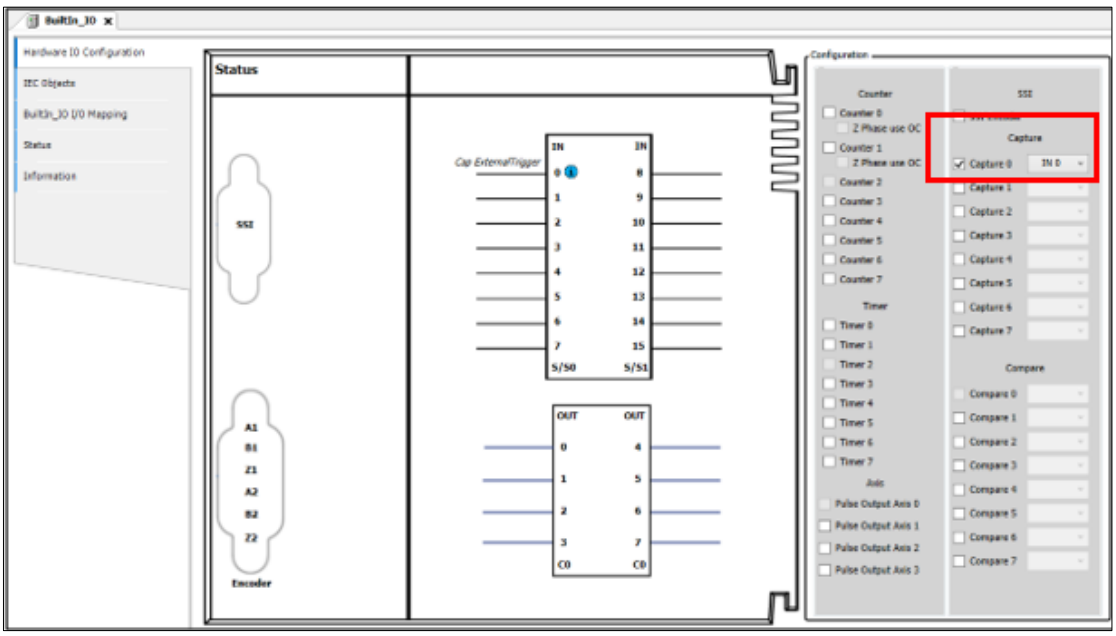


Figure 4 - 137: Capture

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

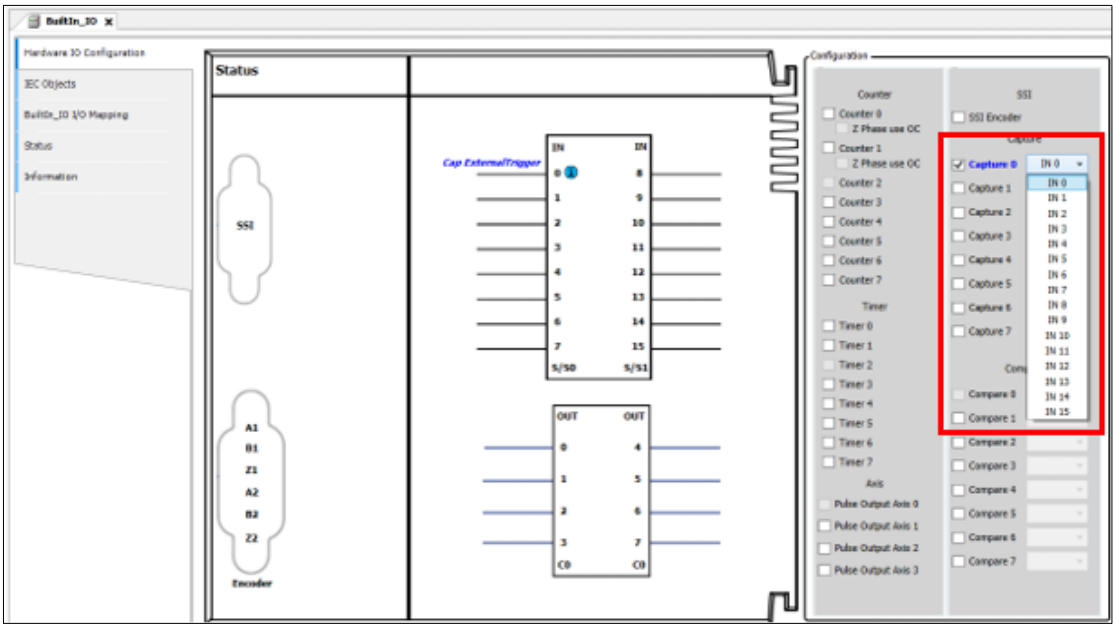


Figure 4 - 138: Capture drop-down menu

- *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.

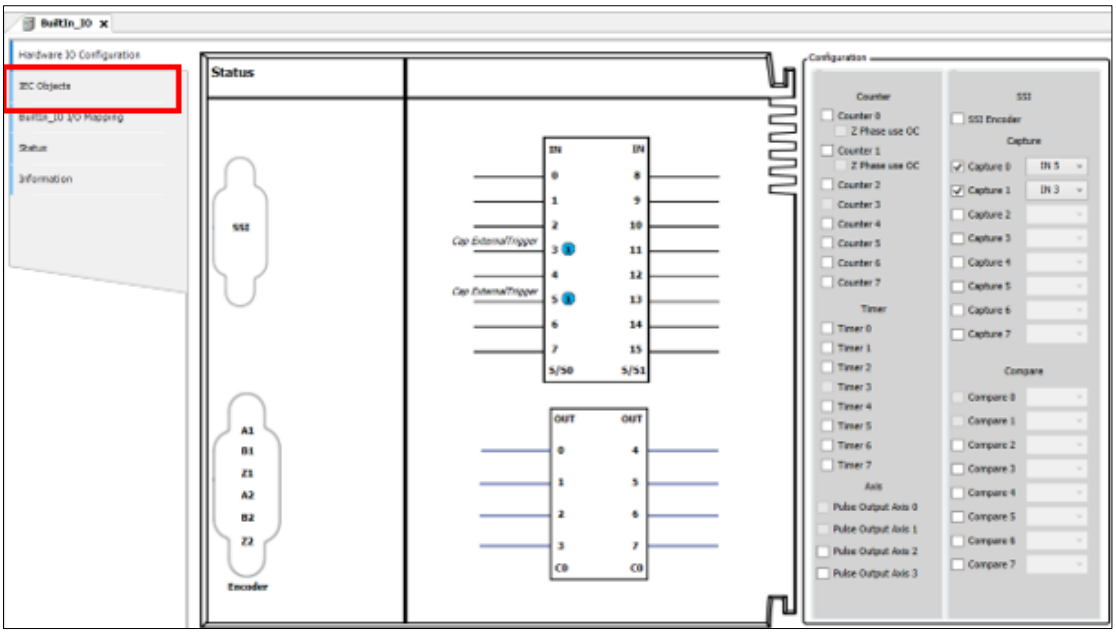


Figure 4 - 139: IEC Object

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.

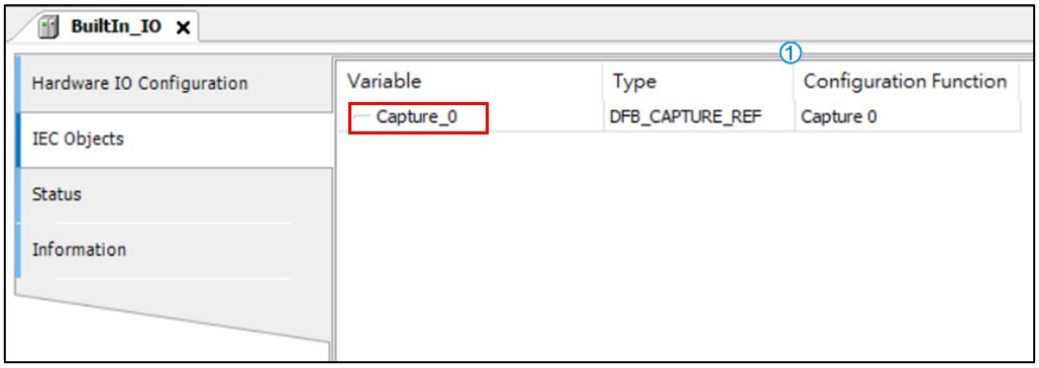


Figure 4 - 140: Using Capture_0 Axis Name

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

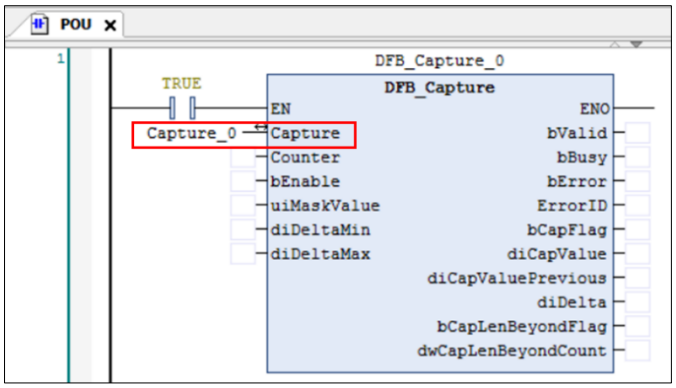


Figure 4 - 141: Entered Axis Name

- High-Speed Counting Comparator (Compare)

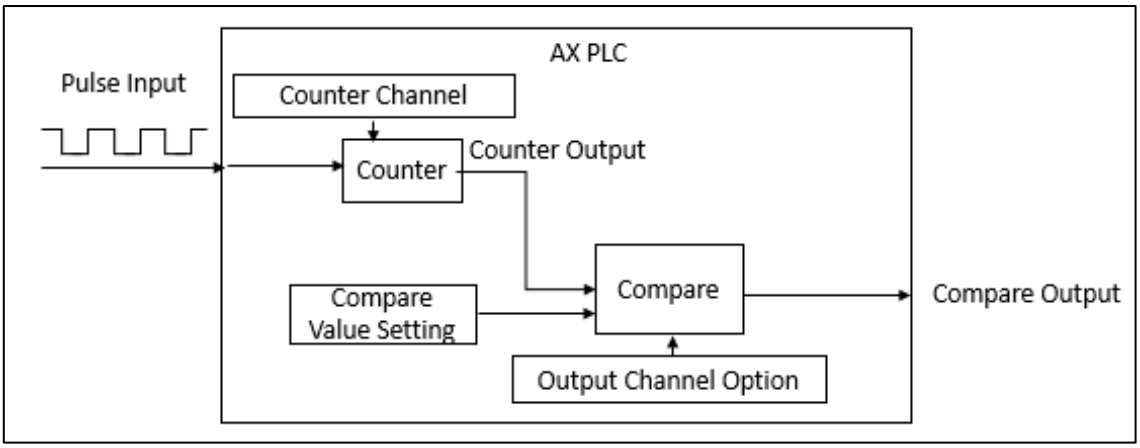


Figure 4 - 142: High speed Counting comparator

- Compare Function start

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

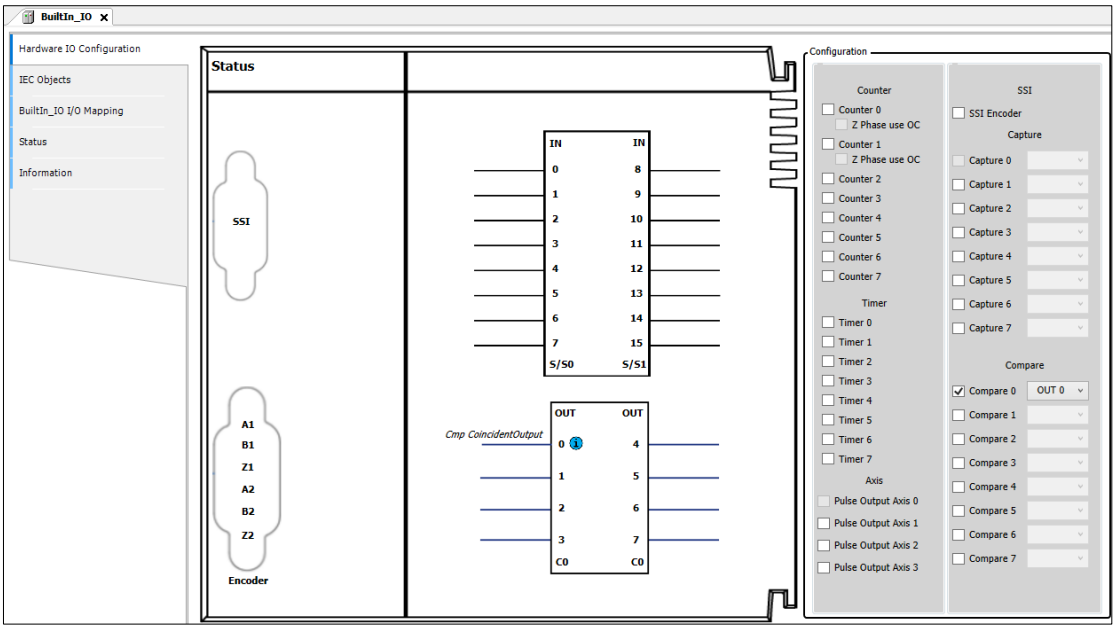


Figure 4 - 143: Compare

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

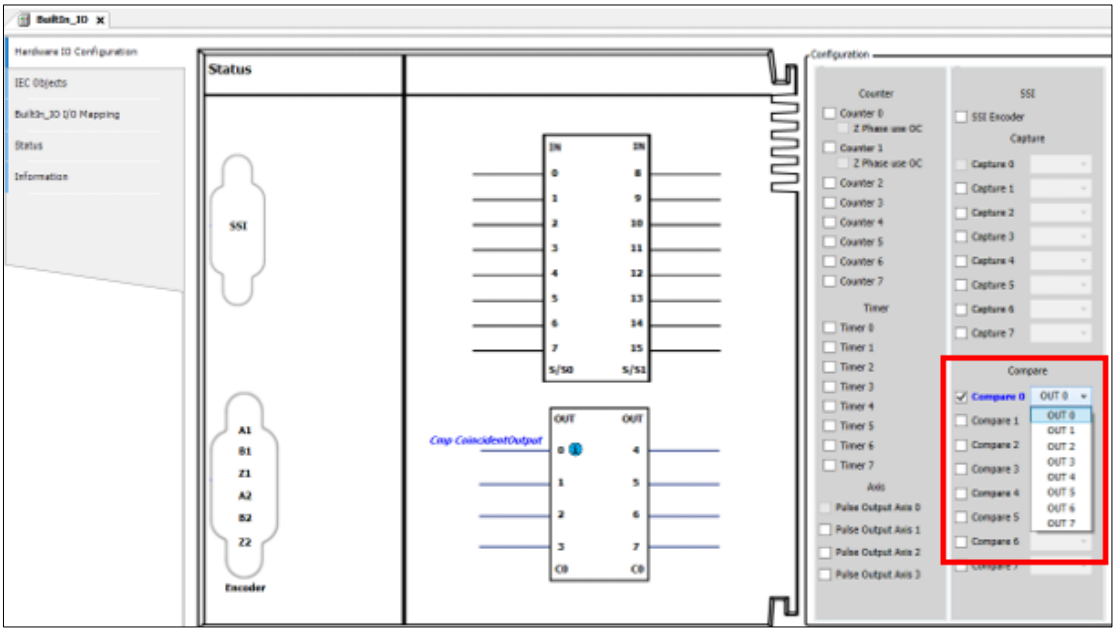


Figure 4 - 144: Compare pull-down menu

- *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.

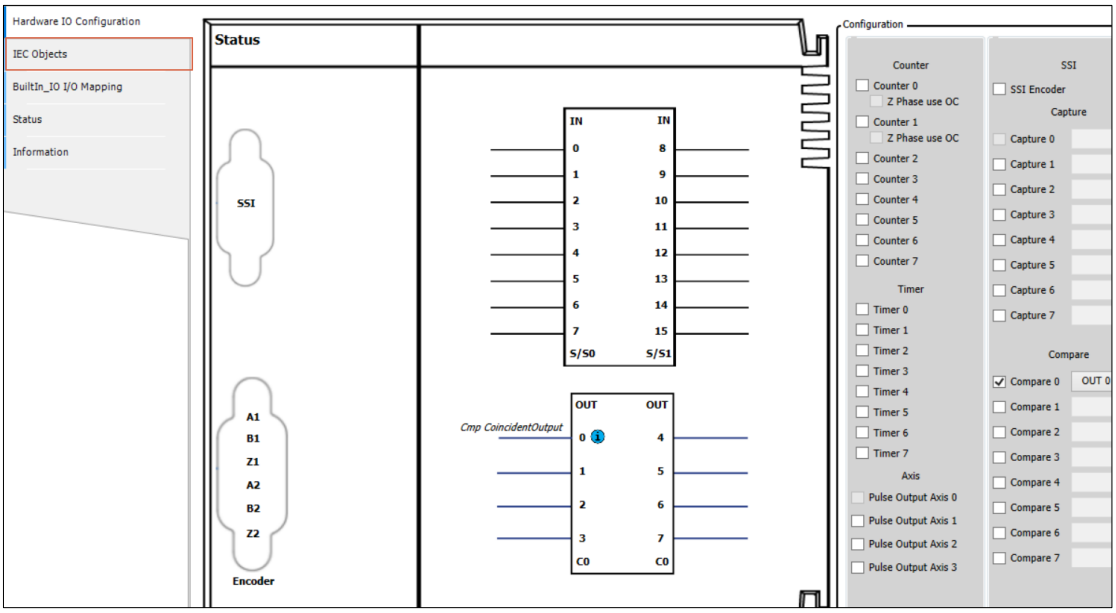


Figure 4 - 145: Select IEC Objects

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.

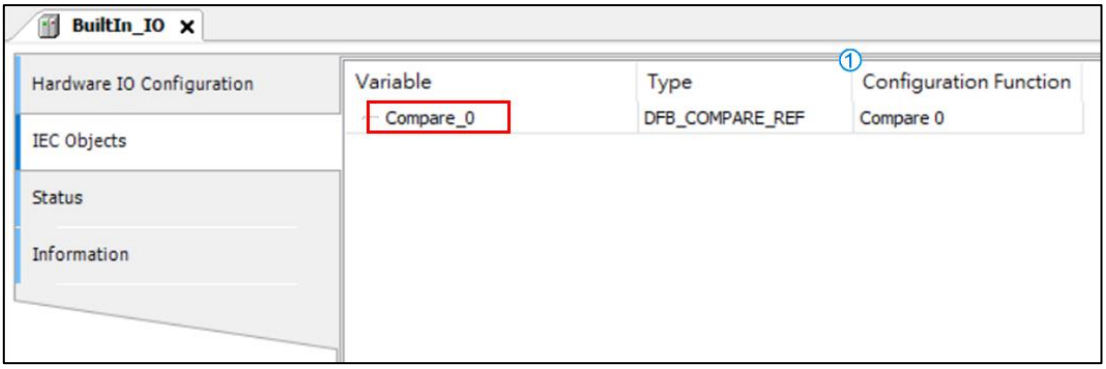


Figure 4 - 146: Enter the Axis name

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

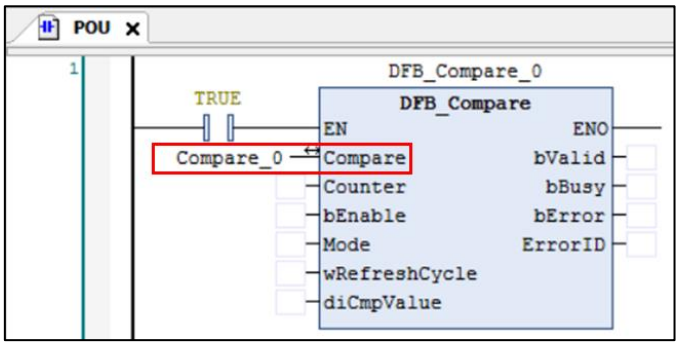


Figure 4 - 147: Axis name compare_0

4.2.7.1.6 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.

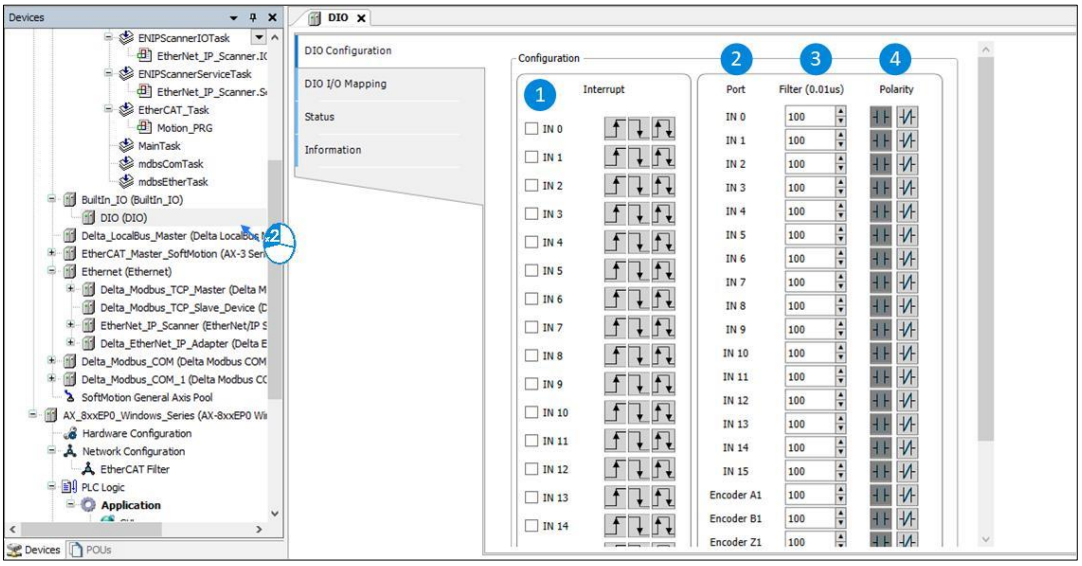


Figure 4 - 148: DIO Configuration

- 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.
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.

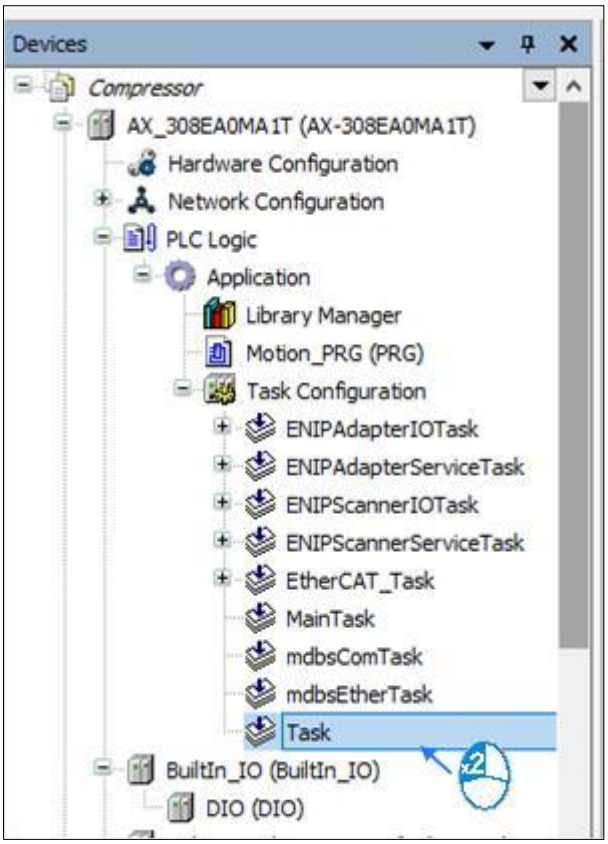


Figure 4 - 149: Add Task to Interrupt function

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

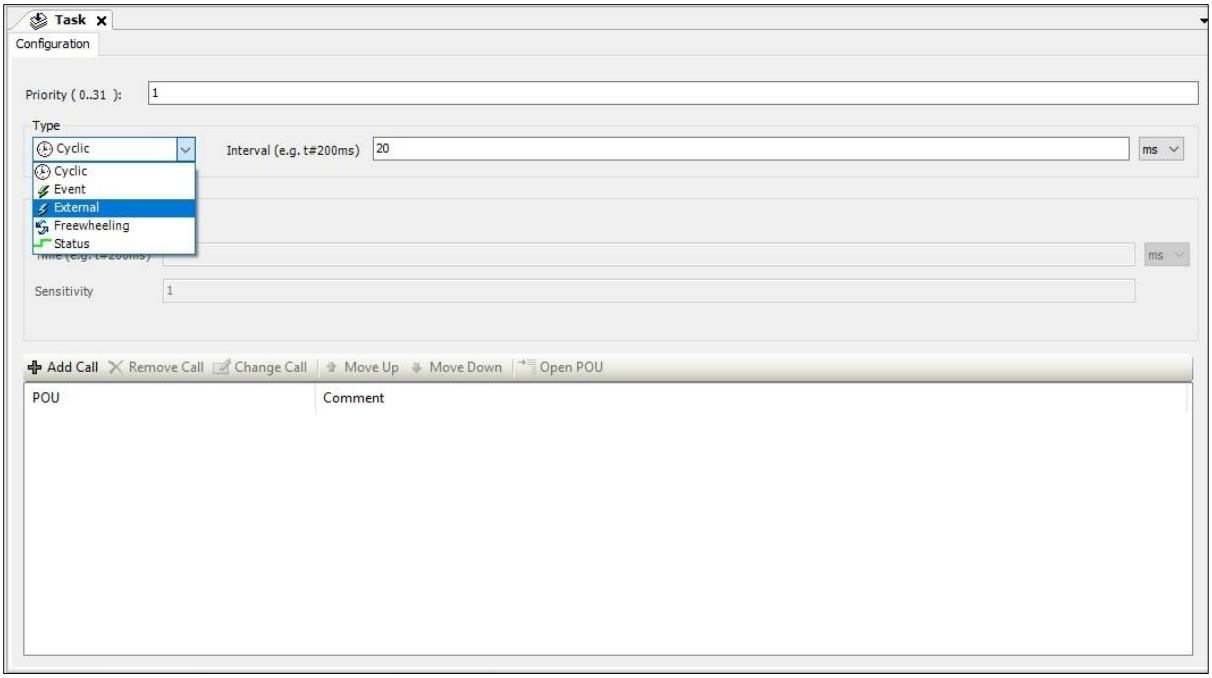


Figure 4 - 150: Selecting external

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

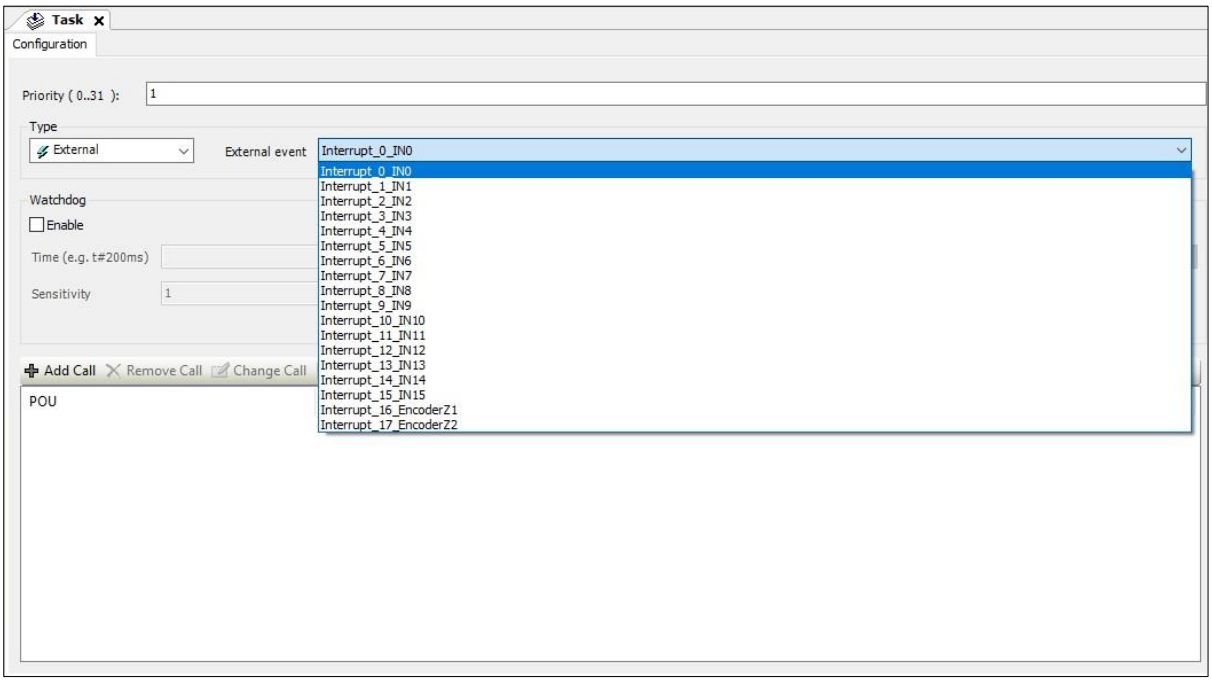


Figure 4 - 151: Selecting external

- 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 figure below:

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

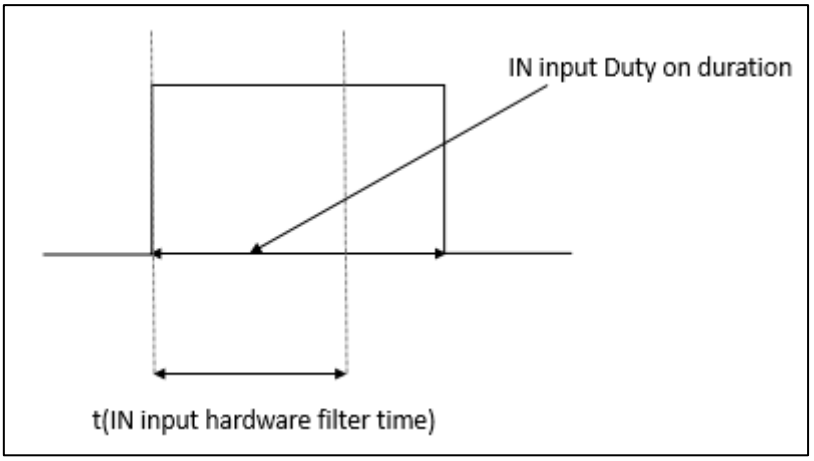


Figure 4 - 152: Hardware filtering time less than inputting Duty On at IN point time

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.

4.2.7.2 High-speed IO setting (AX-324)

4.2.7.2.1 BuiltIn IO: Hardware IO configuration

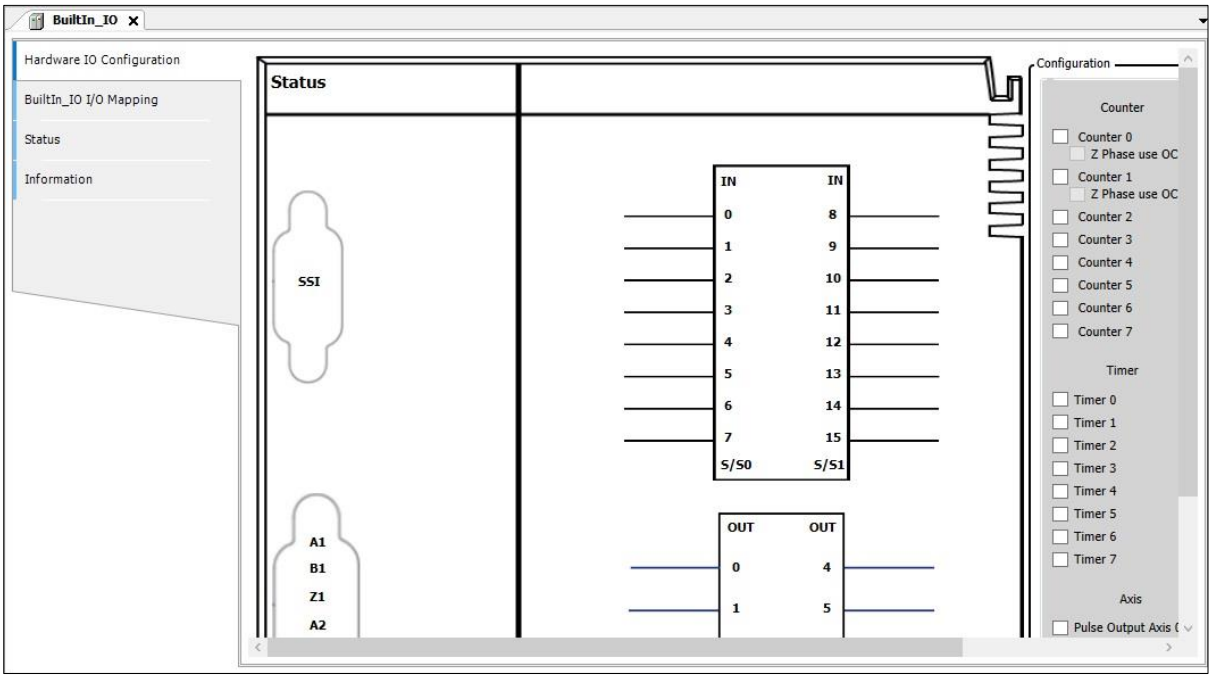


Figure 4 - 153: Hardware IO Configuration Tab

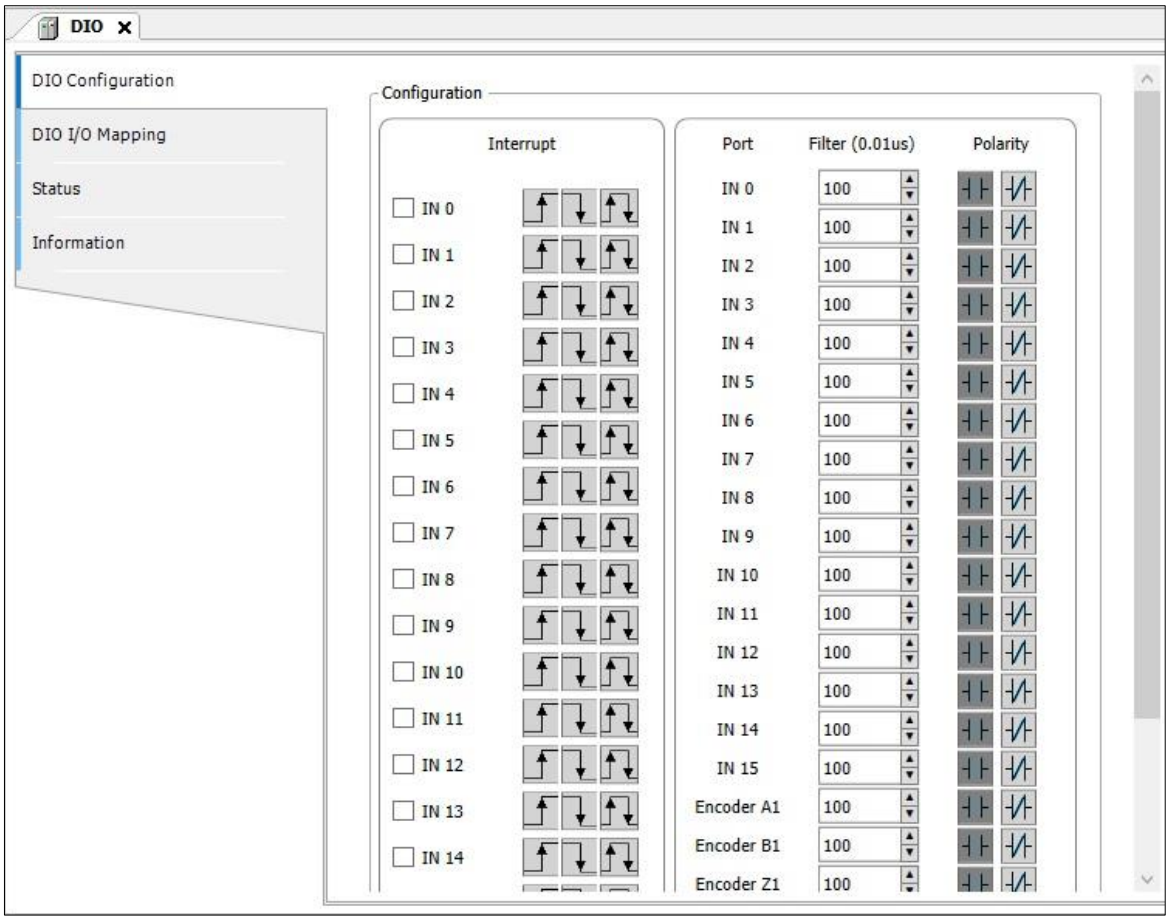


Figure 4 - 154: DIO Tab

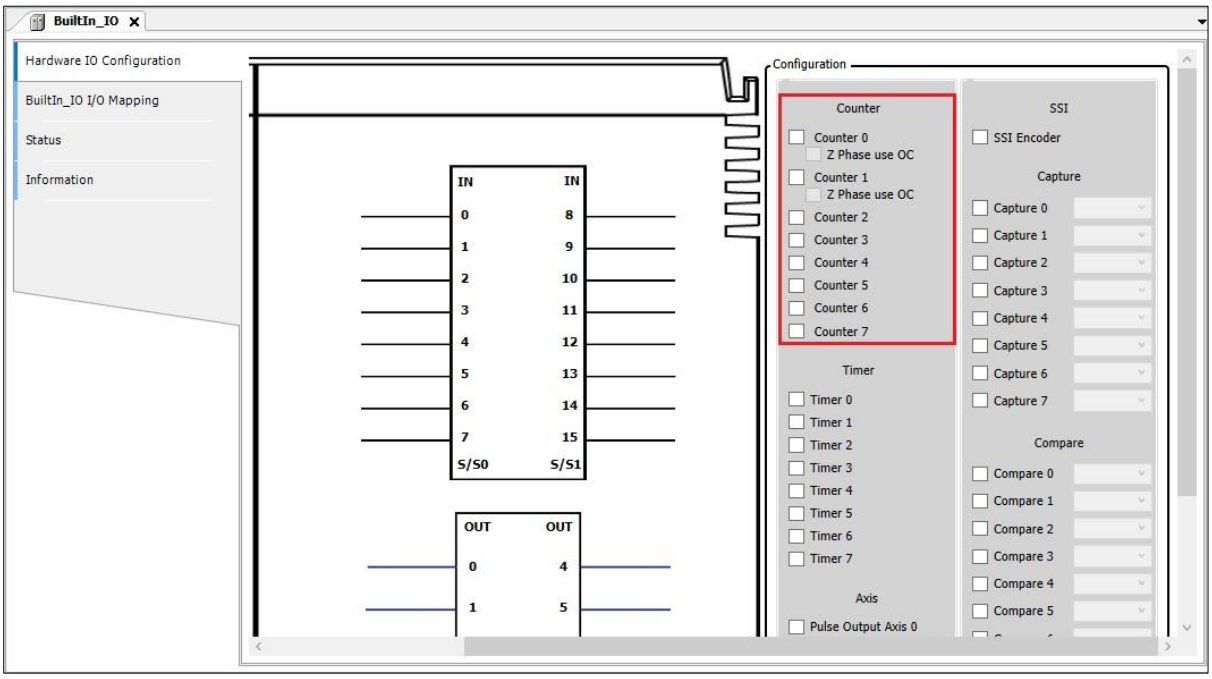


Figure 4 - 155: Counter

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

- DIO: interrupt, filter, polarity and other functions can be set.
- Counter: You 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: You can declare the variables of the high-speed capture or high-speed comparator.

4.2.7.2.2 BuiltIn IO: Counter/timer configuration

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

High-speed counter (Cnt)

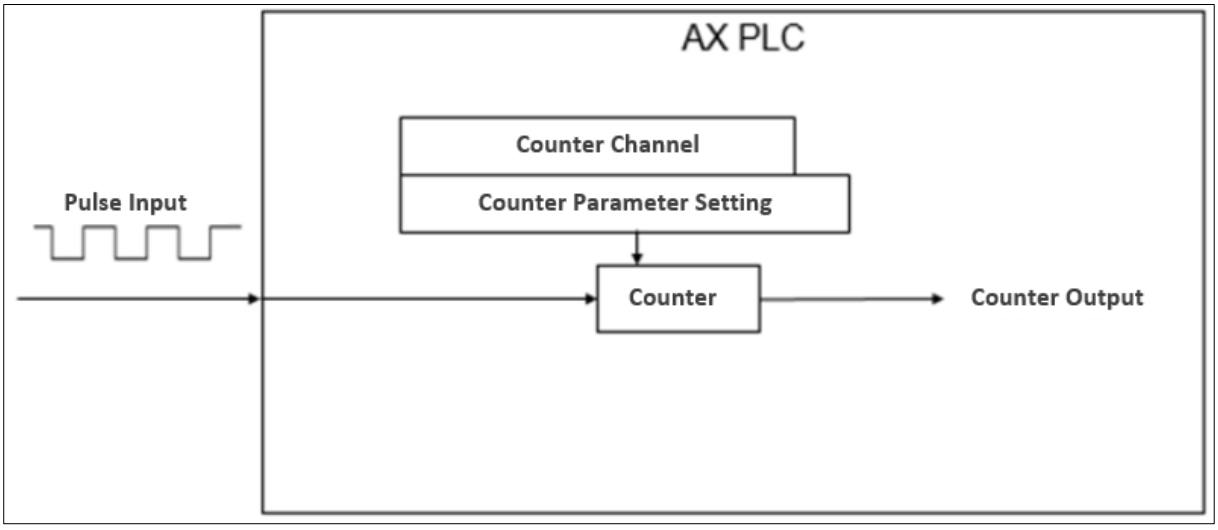


Figure 4 - 156: High-speed Counter

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.

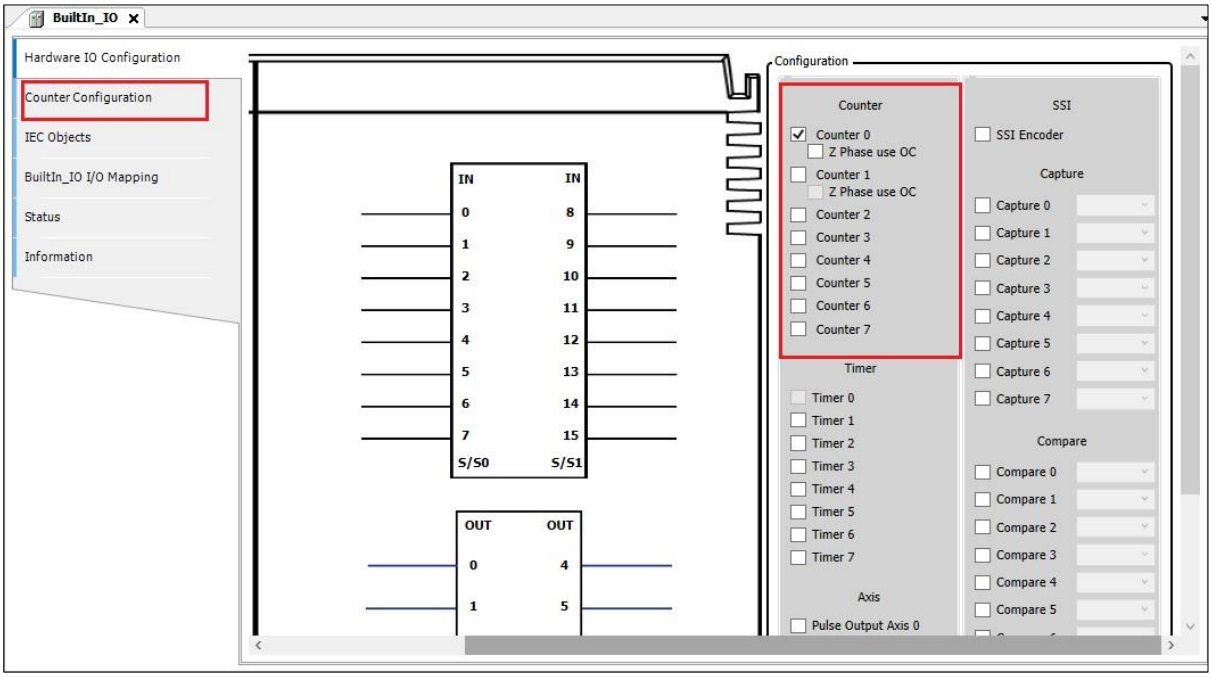


Figure 4 - 157: Counter

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

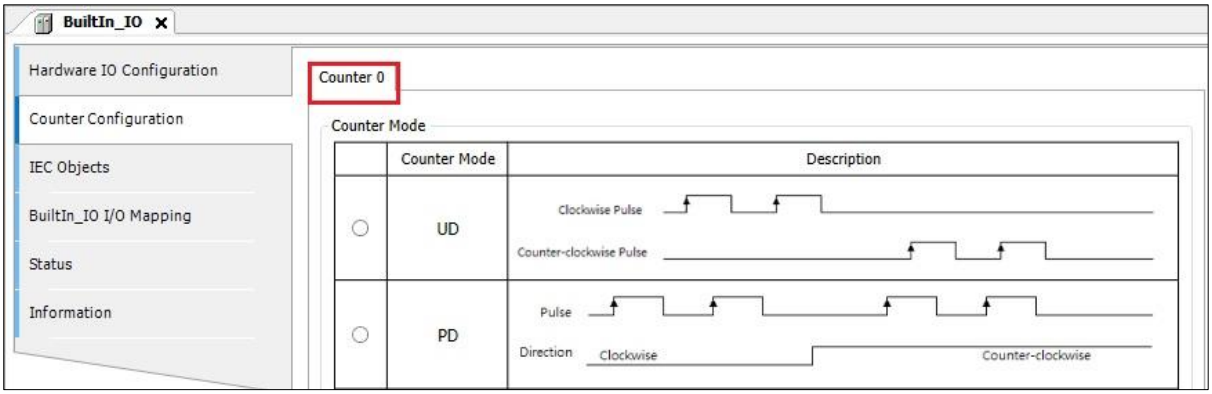


Figure 4 - 158: Counter configuration tab

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

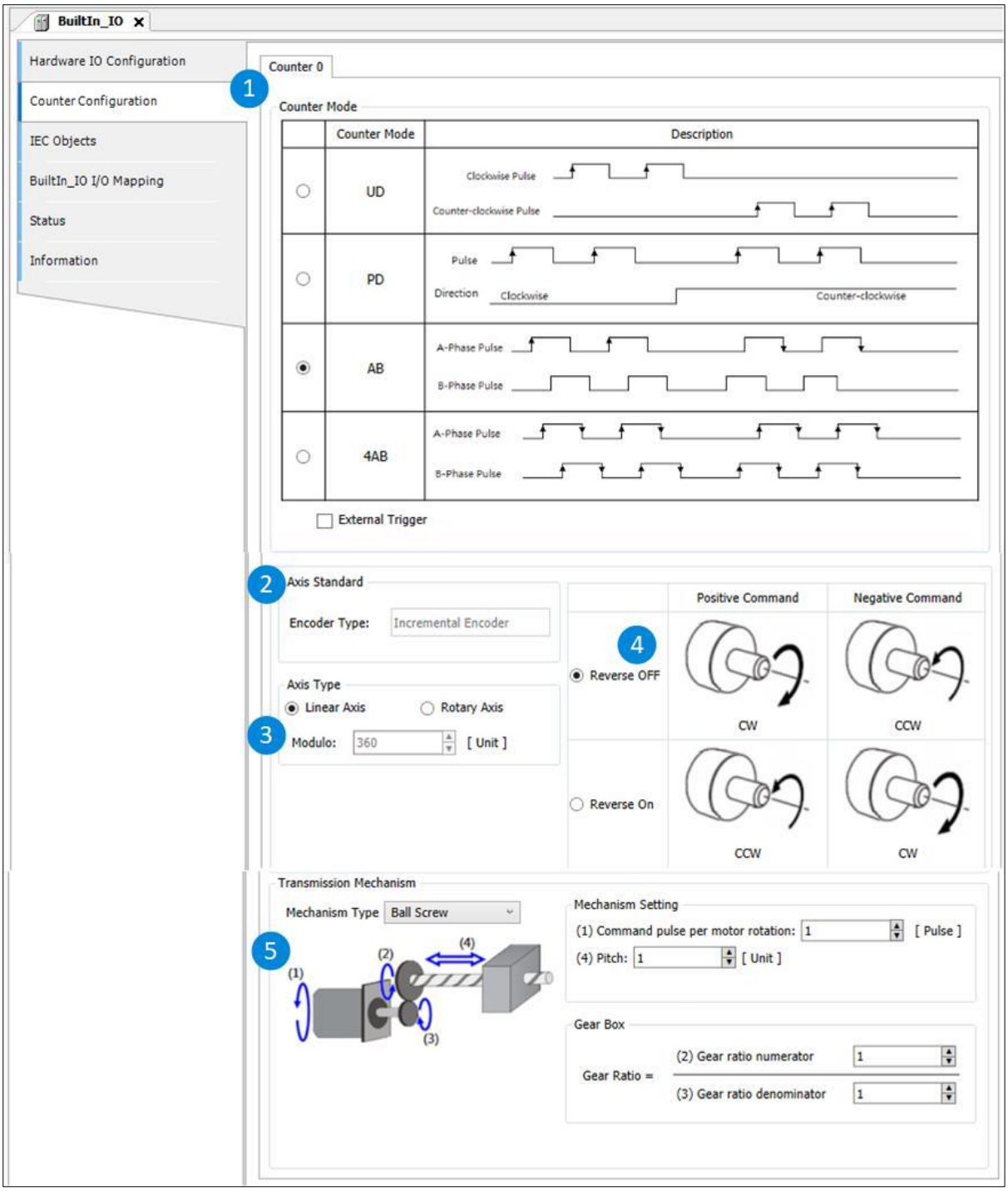


Figure 4 - 159: Counter configuration screen

① Counting Mode

Name	Features
UD	Forward pulse train and reverse pulse train

Name	Features
PD	Pulse train + direction
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

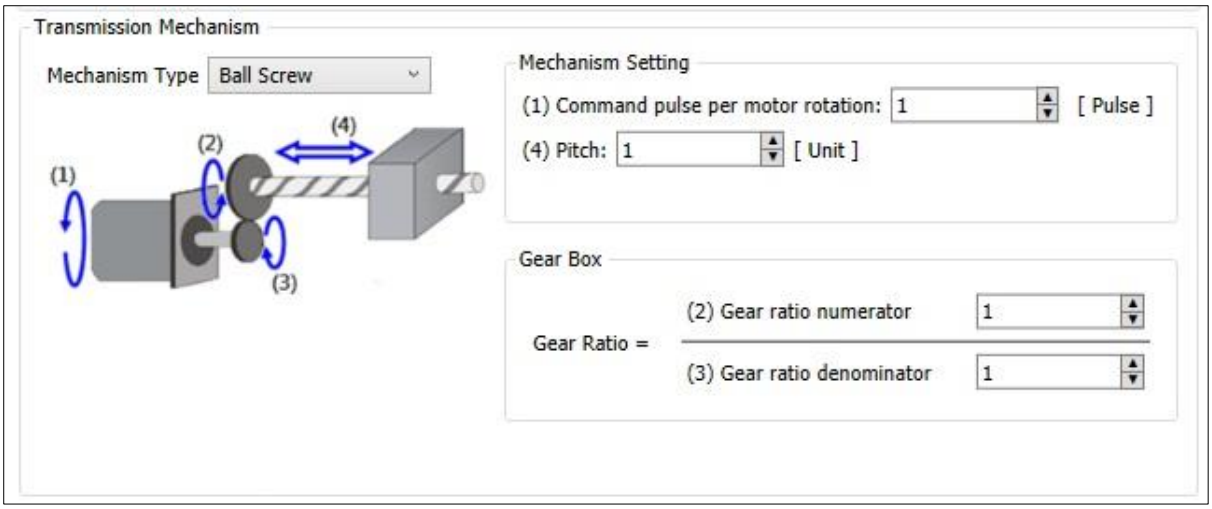


Figure 4 - 160:Ball Screw Drive

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

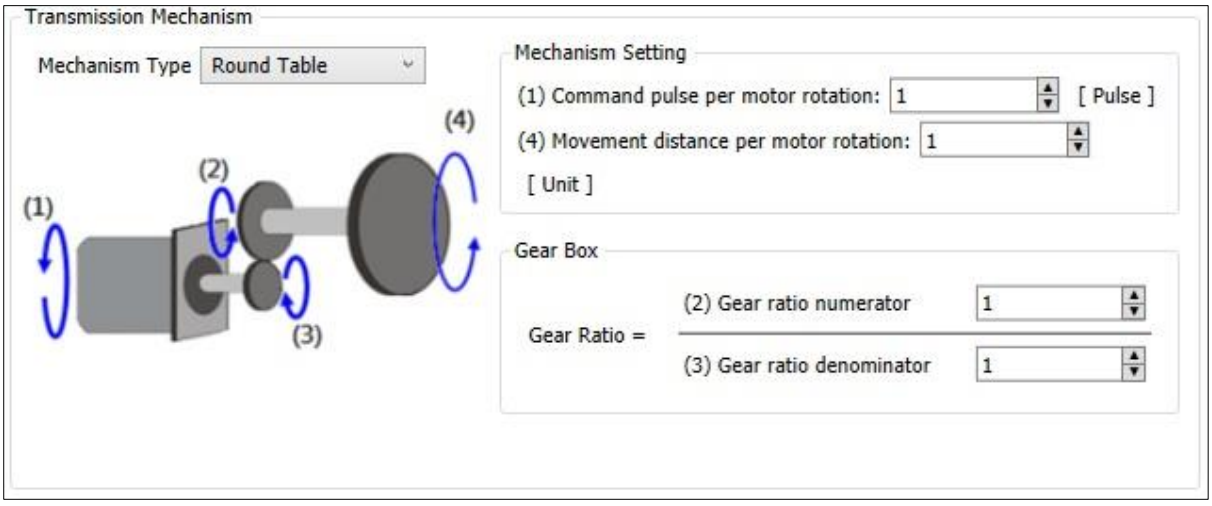


Figure 4 - 161: Round Table Transmission

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

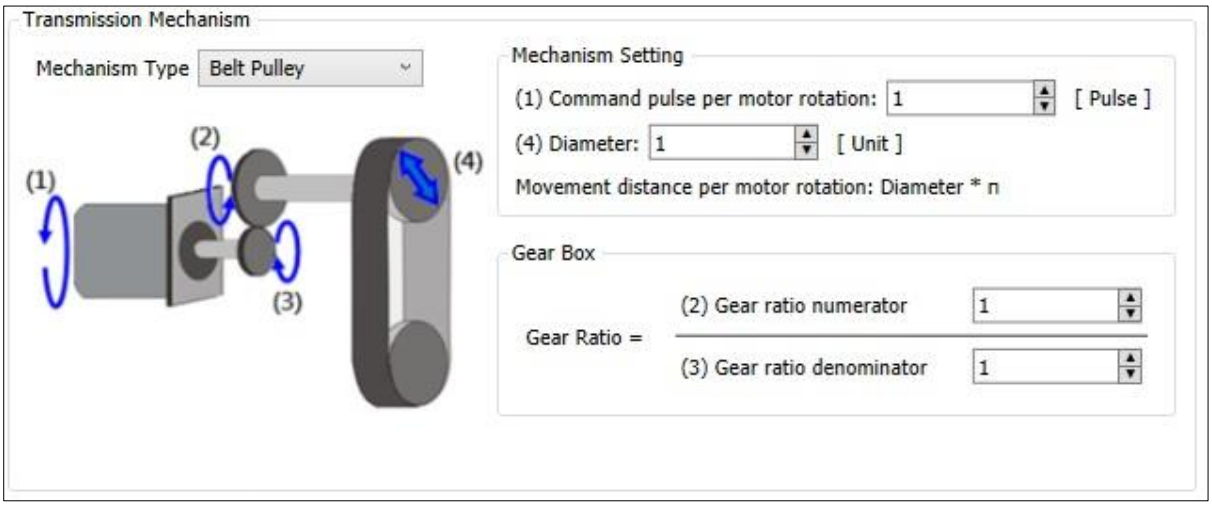


Figure 4 - 162: Belt Drive

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*.

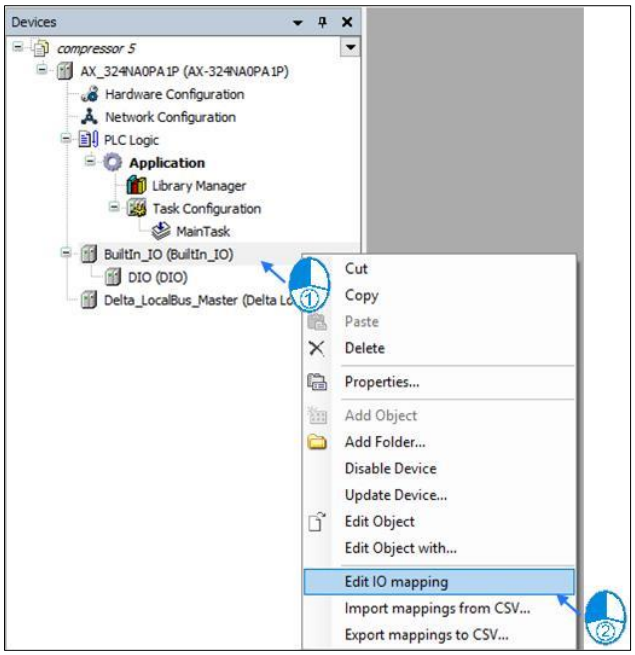


Figure 4 - 163: Edit IO mapping

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

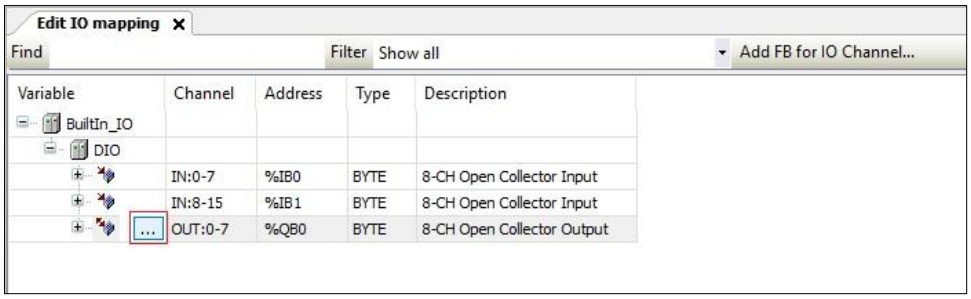


Figure 4 - 164: Edit IO mapping to add new variables

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*.

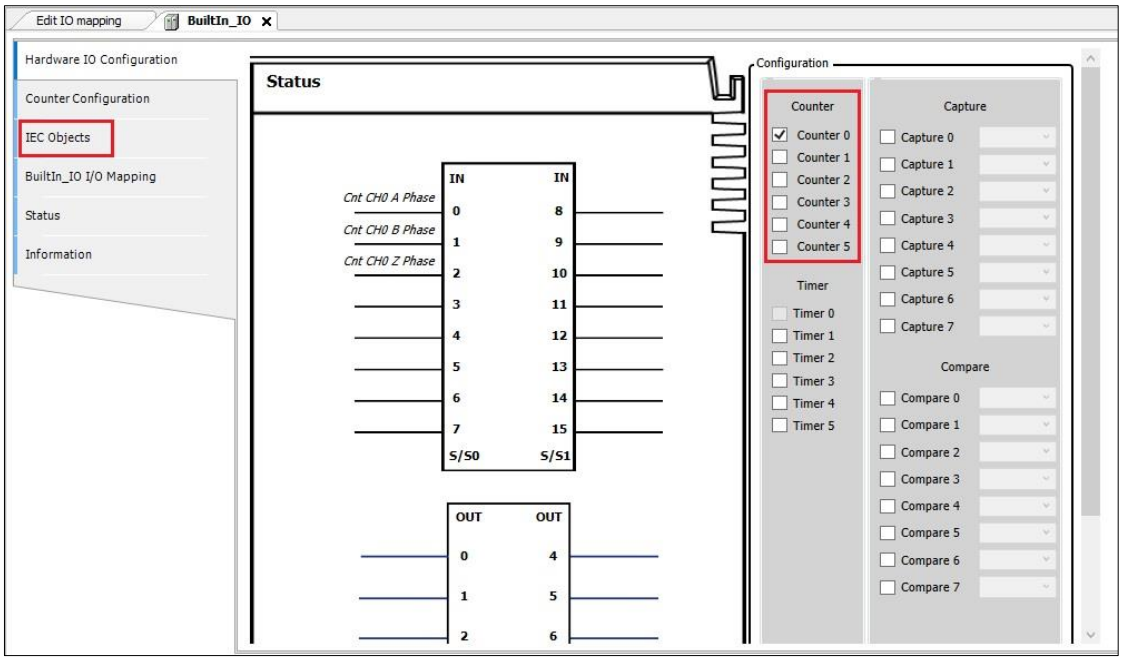


Figure 4 - 165: IEC Object Tab

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

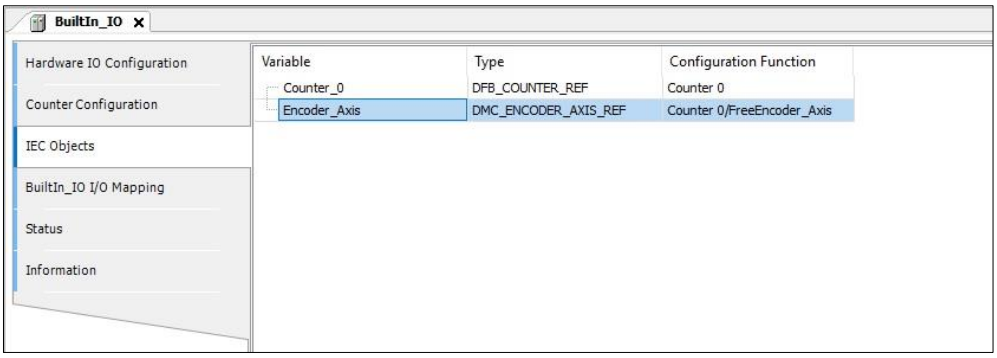


Figure 4 - 166: IEC object tab Encoder_Axis

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.

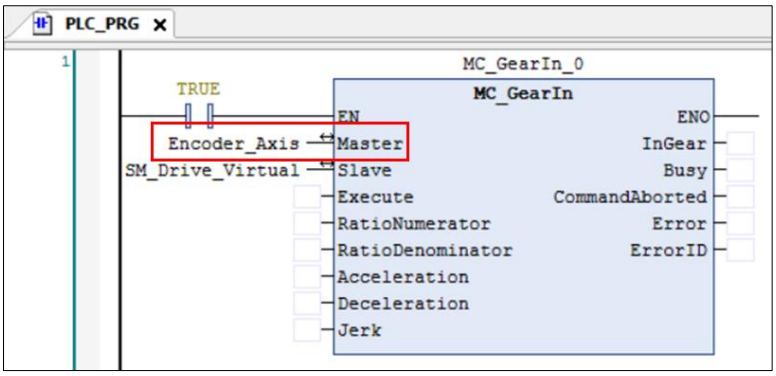


Figure 4 - 167: Encoder_Axis

High-speed timing (Tmr)

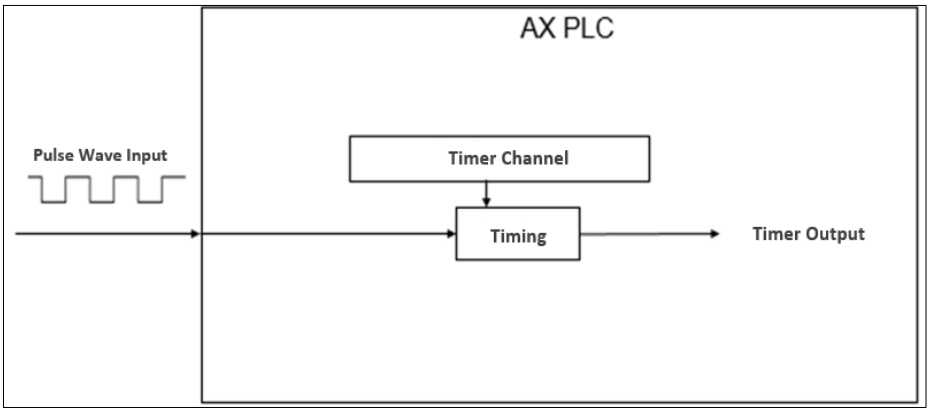


Figure 4 - 168: High-speed timing

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 0.1μs.

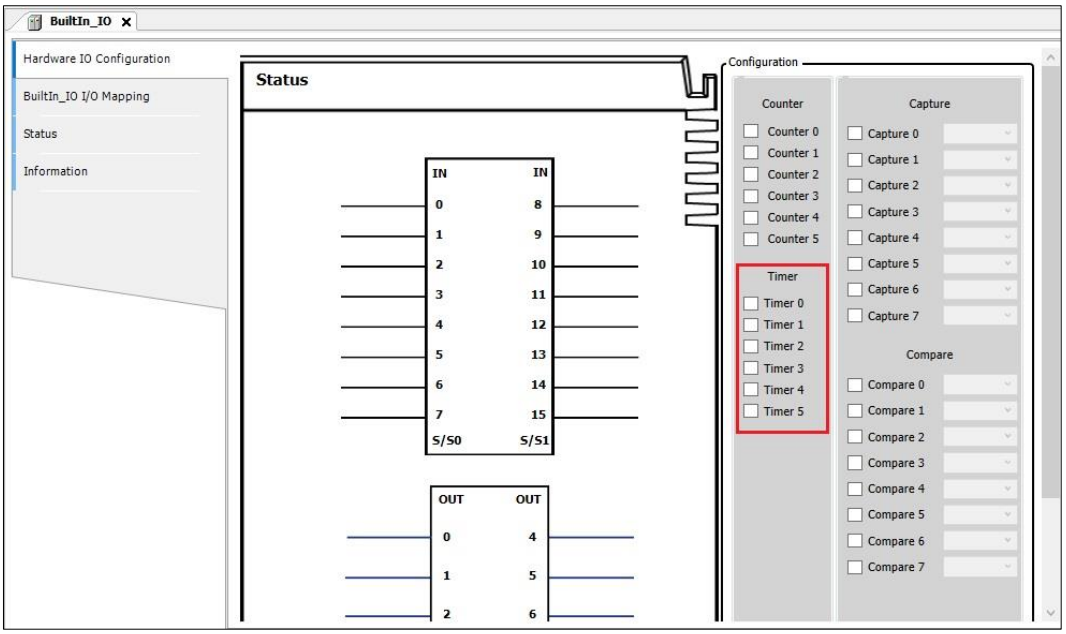


Figure 4 - 169: Timer

Set timer mapping variable

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

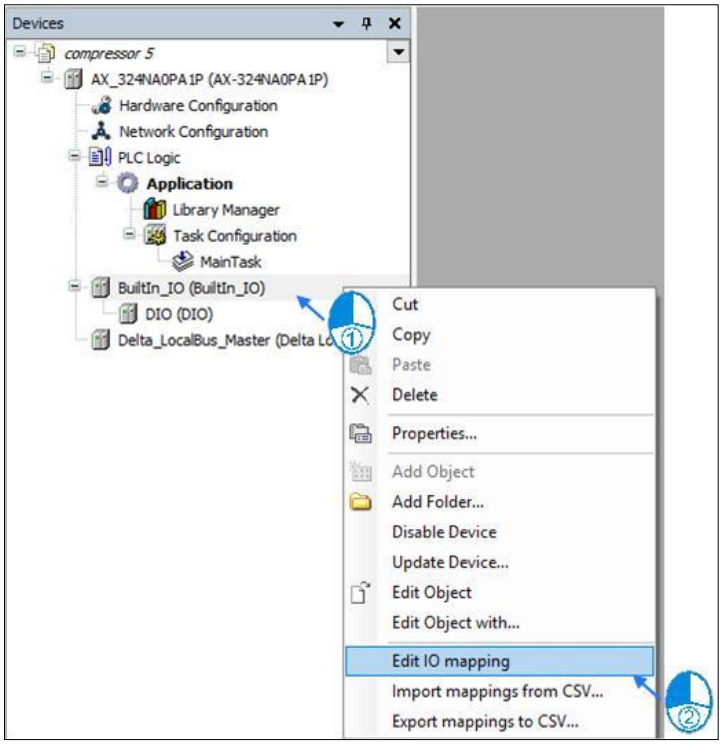


Figure 4 - 170: Edit IO mapping

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

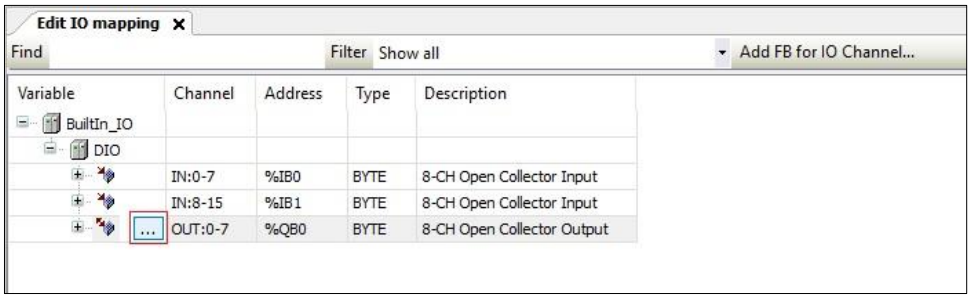


Figure 4 - 171: Edit IO mapping to add new variables

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 object* on the *BuiltIn_IO screen*.

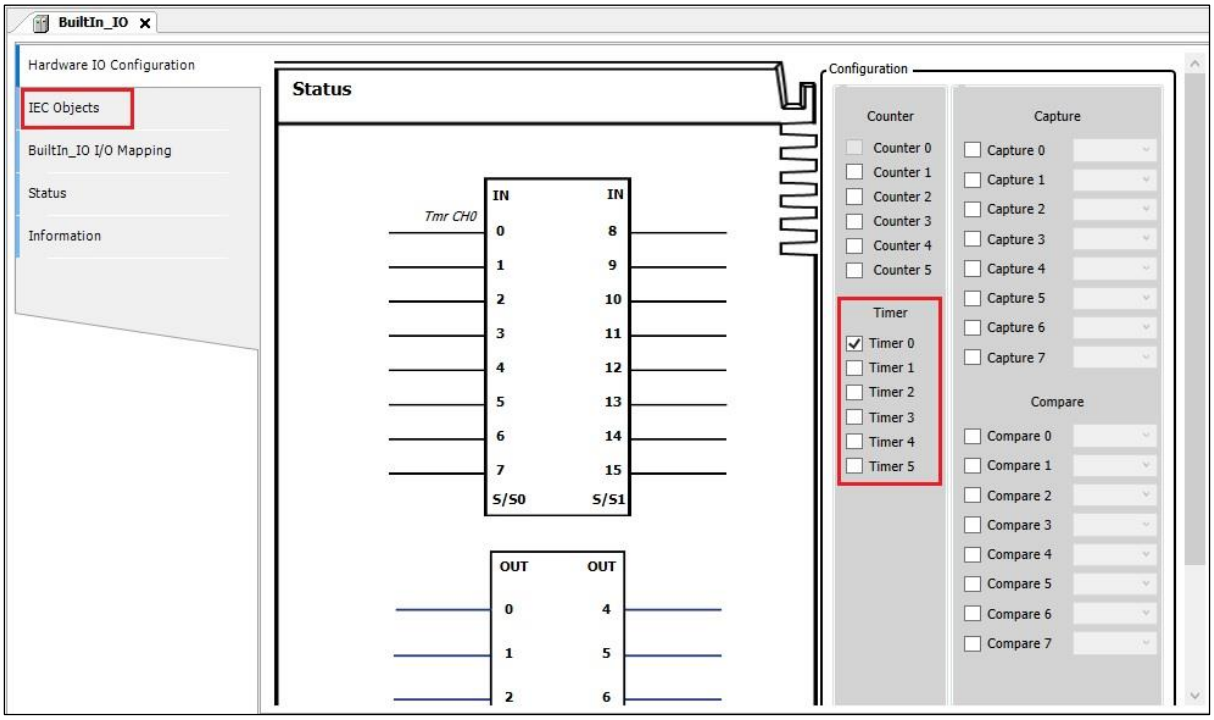


Figure 4 - 172: IEC Objects Tab

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

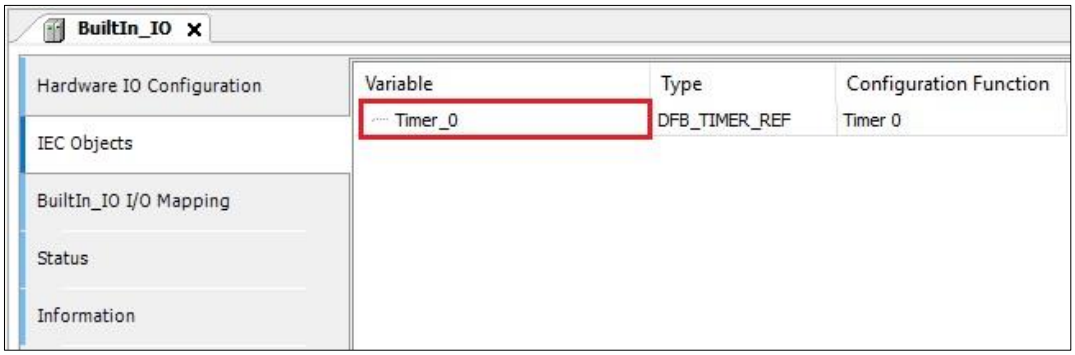


Figure 4 - 173: IEC Object Timer_0

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

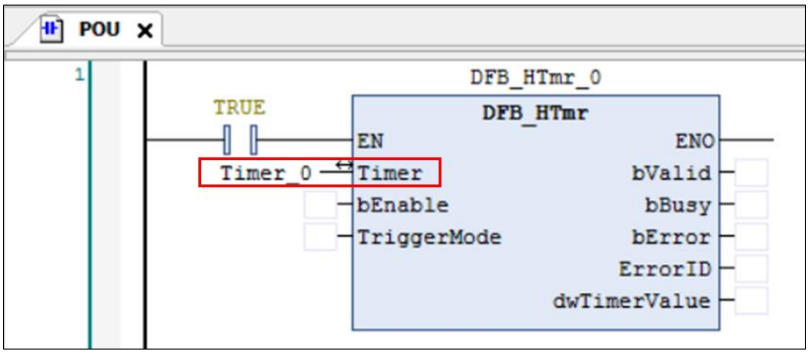


Figure 4 - 174: Timer 0

4.2.7.2.3 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 supports up to 8 groups of high-speed catcher and high-speed counter comparator.

High-speed counter capture

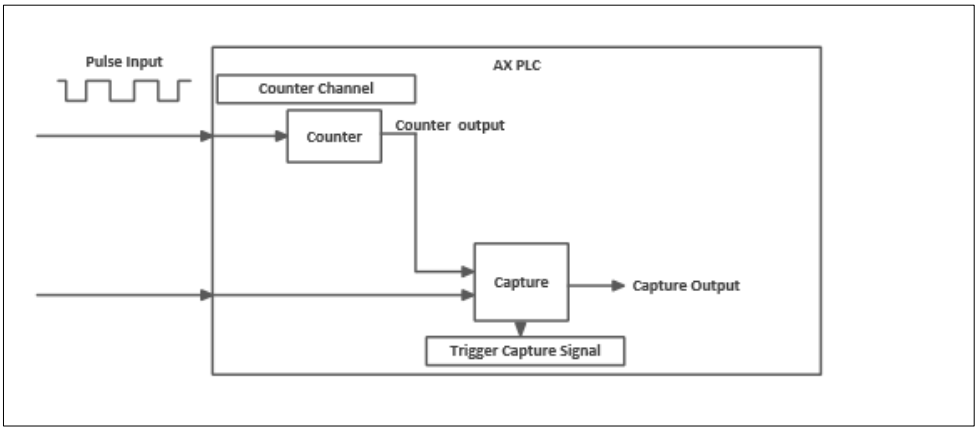


Figure 4 - 175: High-speed counter capture

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.

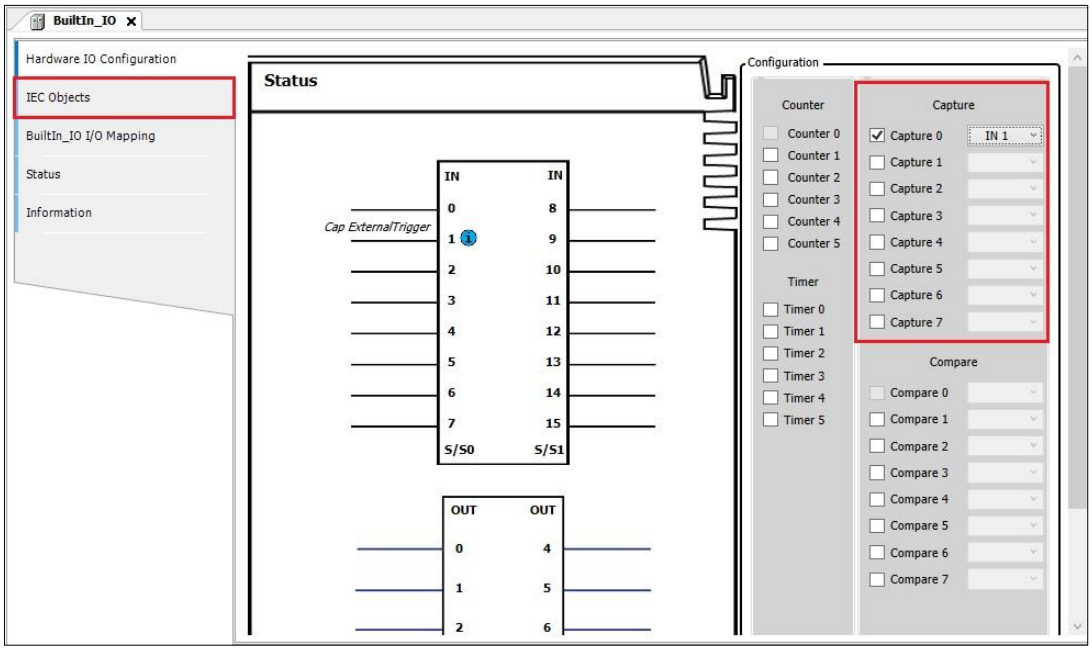


Figure 4 - 176: Capture

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

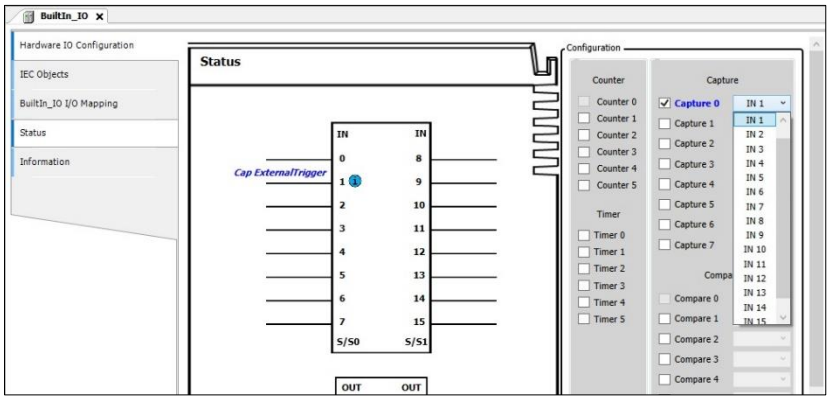


Figure 4 - 177: Capture drop-down menu

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 object on the BuiltIn_IO screen below.

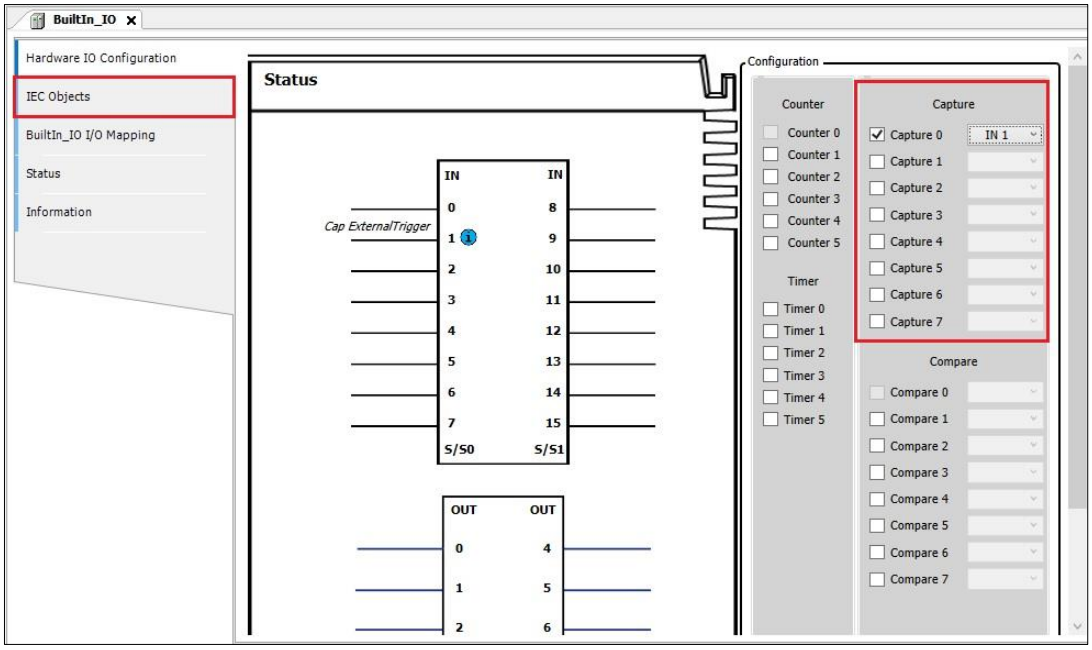


Figure 4 - 178: IEC objects

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

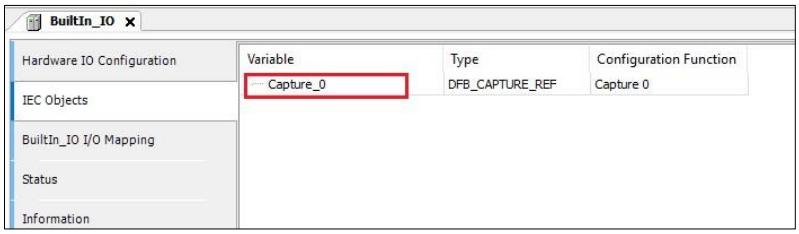


Figure 4 - 179: IEC object Capture_0

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

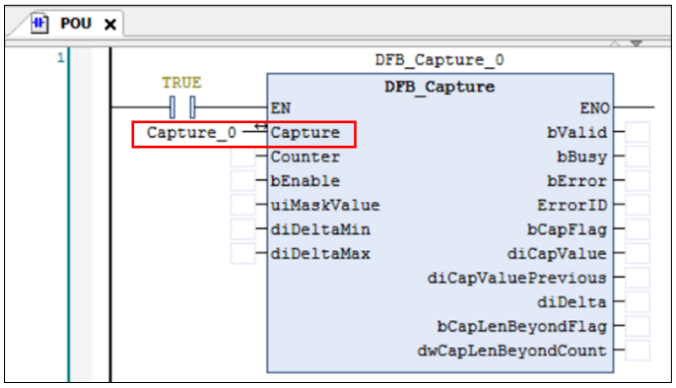


Figure 4 - 180: Capture_0

High-speed counting comparator

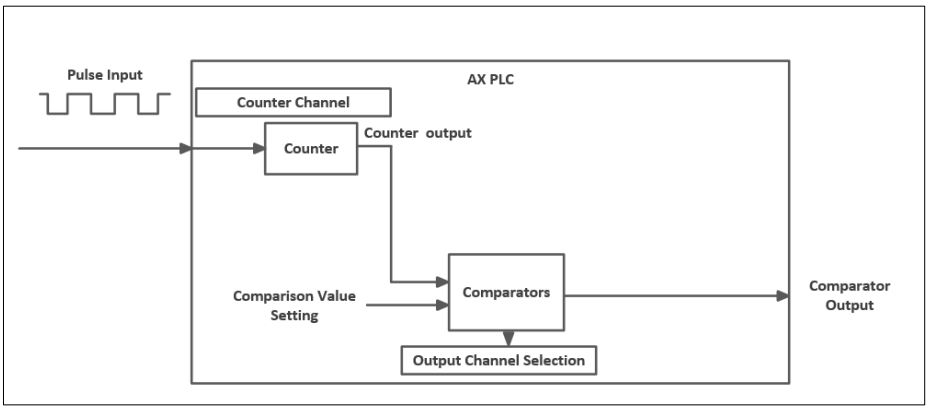


Figure 4 - 181: High-speed counting comparator

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

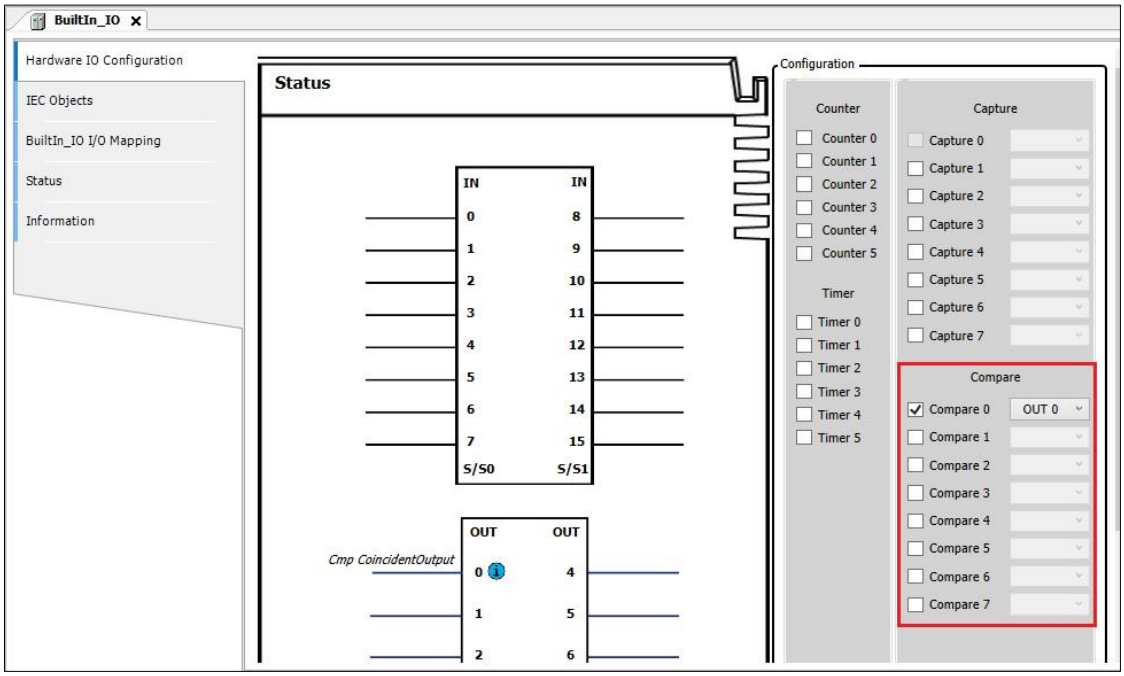


Figure 4 - 182: Comparators

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

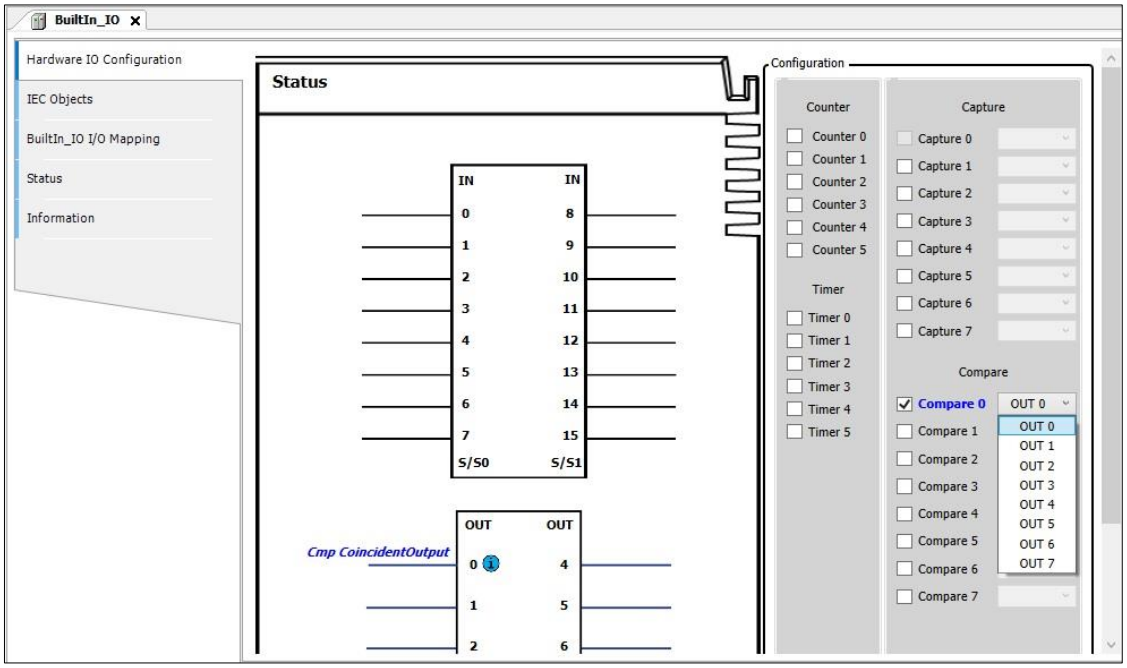


Figure 4 - 183: Comparator drop-down menu

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 object on the BuiltIn_IO screen below.

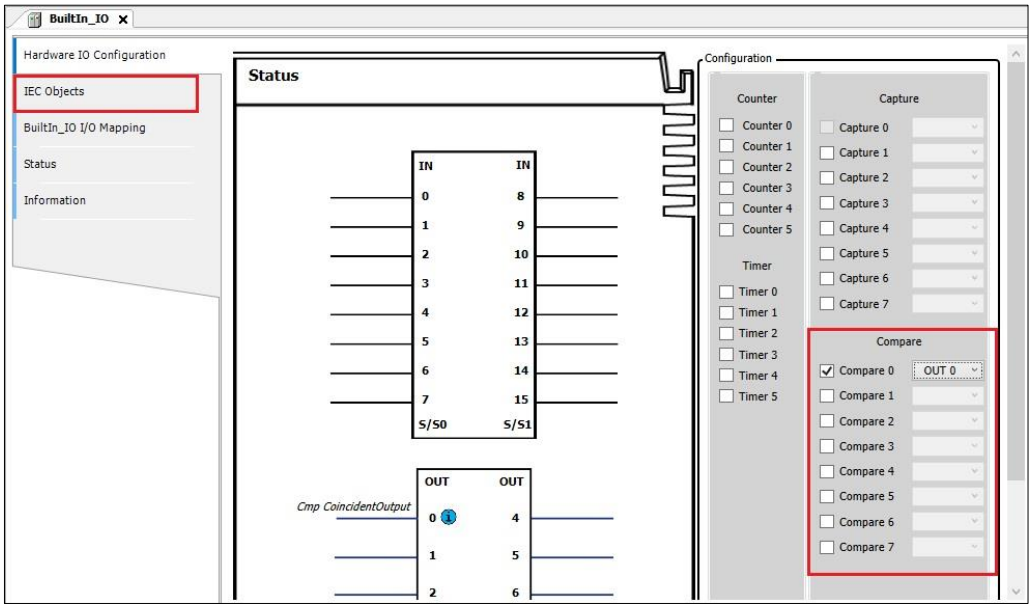


Figure 4 - 184: IEC objects

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

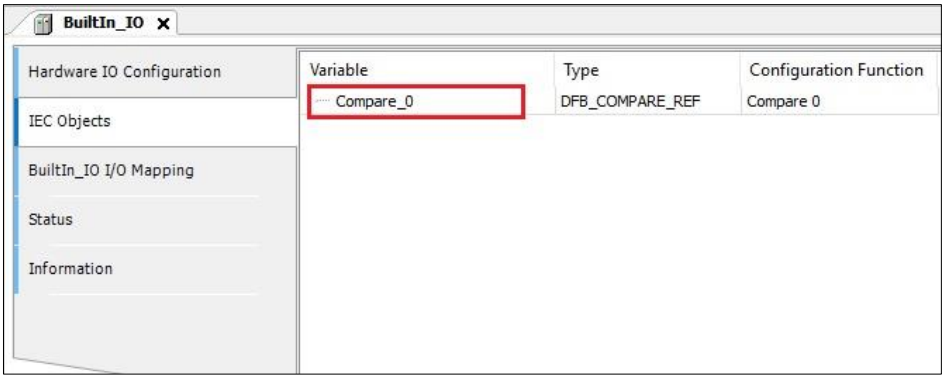


Figure 4 - 185: IEC Object Compare_0

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

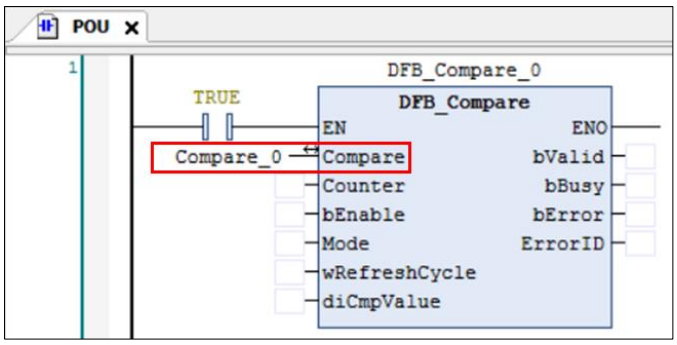


Figure 4 - 186: Compare_0

4.2.7.2.4 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.

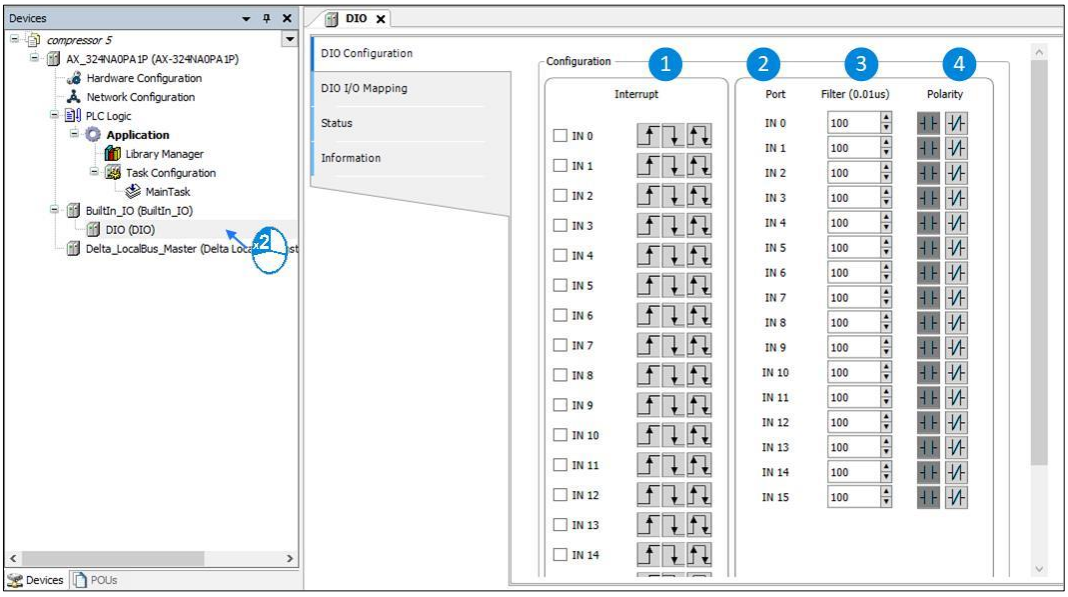


Figure 4 - 187: Enter DIO setting screen

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.
④ polarity	Set the polarity of the input contact, the default value is A contact.

Features	Description
	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.

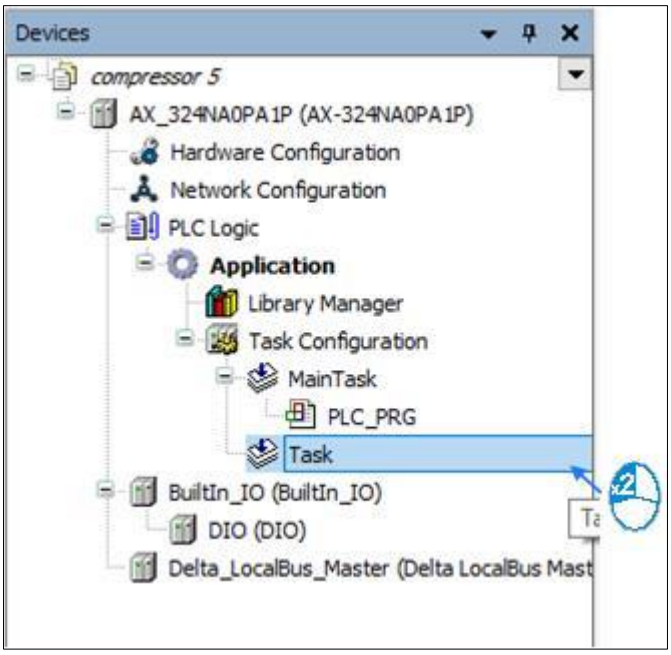


Figure 4 - 188: I/O interrupt method

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

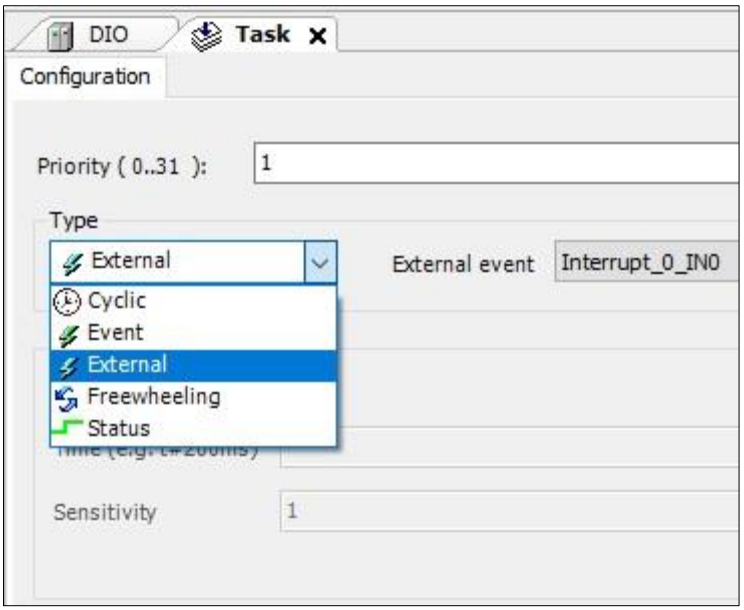


Figure 4 - 189: Task setting screen

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

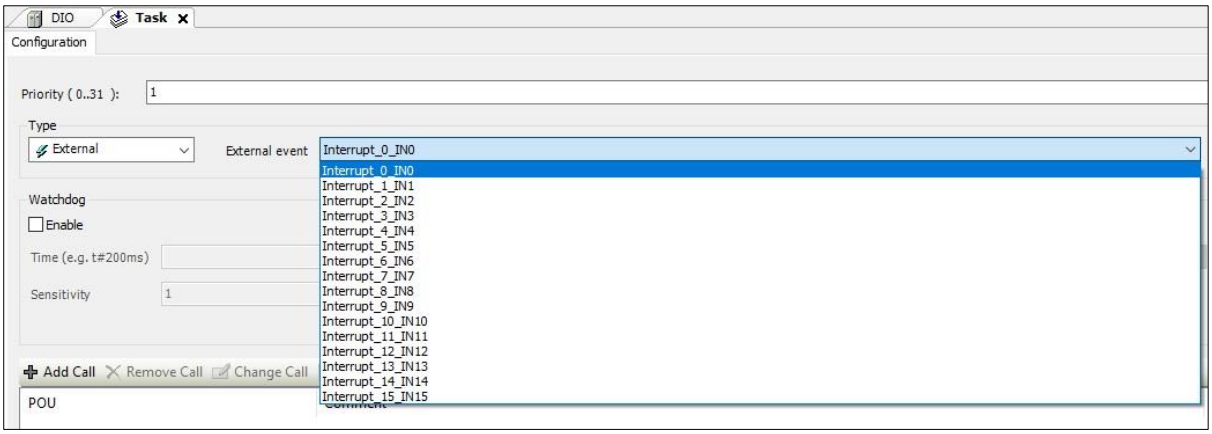


Figure 4 - 190: External event drop-down menu

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

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

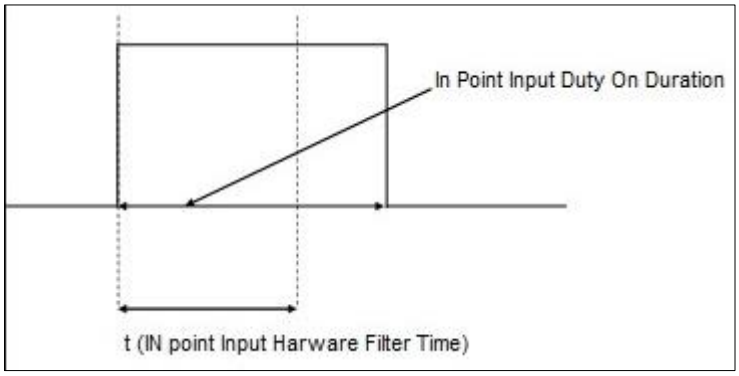


Figure 4 - 191: Hardware filter time setting value

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.

4.3 AX-8xxEP0 Series Parameter 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.

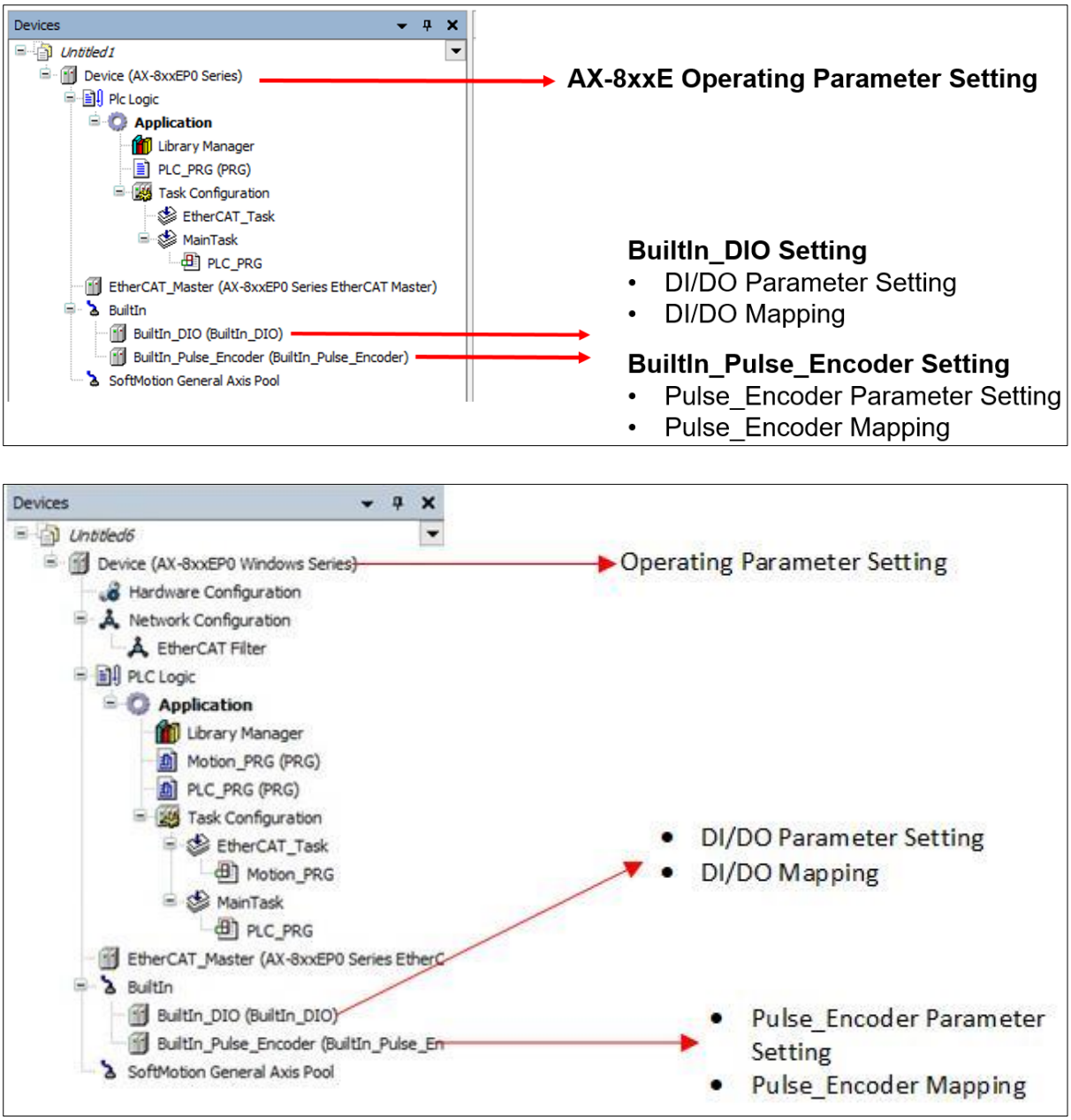


Figure 4 - 192: AX-8 Parameter setting

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.

4.3.1 System Setting Page

System setting supports the communication and controller basic parameters. Double-click the AX-8xxEP0 series device in the device tree to open the parameter tab.

- Parameters Tab

Item	Description
Application Run Mode	Set RTE and PLC “Run” status when controller active • Setting Value True, False • Default Value False ***NOTE**: Only can change the setting when offline. After Download, the status will be Run.*
Serial Communication Mode	Set Serial port communication mode • Setting Value: RS-485, RS-422, SSI • Default Value: RS-485 ***NOTE**: Only can change the setting when offline. After Download, the status will be Run.*

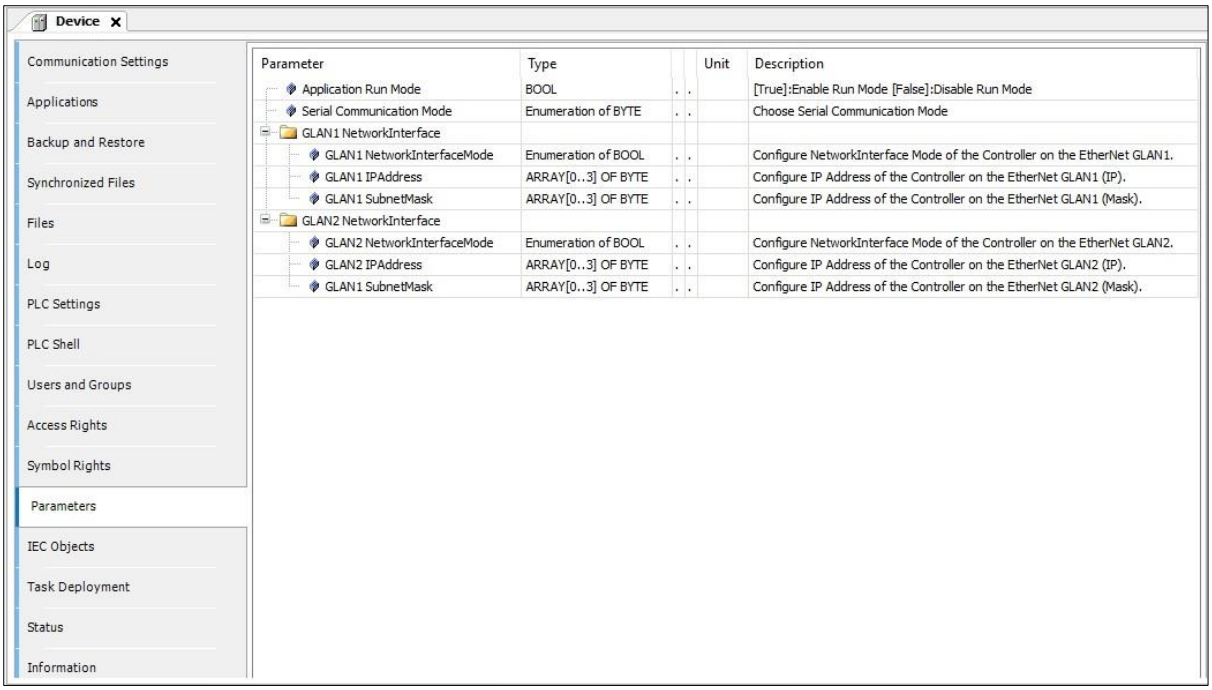


Figure 4 - 193: AX-8xxEP0 series device parameter tab

Item	Description
Parameter	Parameter name, not editable
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

4.3.2 COM Port Settings

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

4.3.2.1 MODBUS COM Port

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

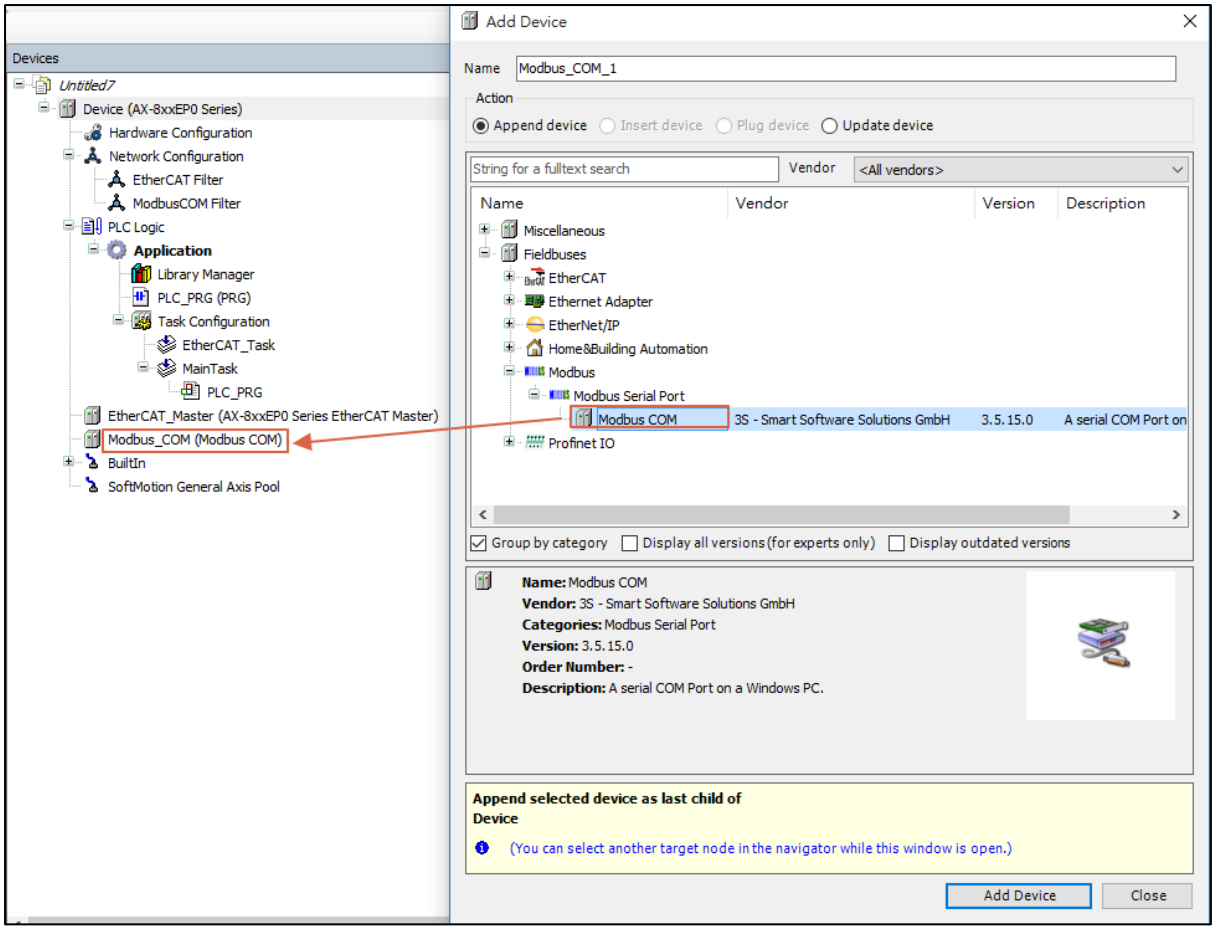


Figure 4 - 194: Adding MODBUS COM

- 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

4.3.2.2 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.

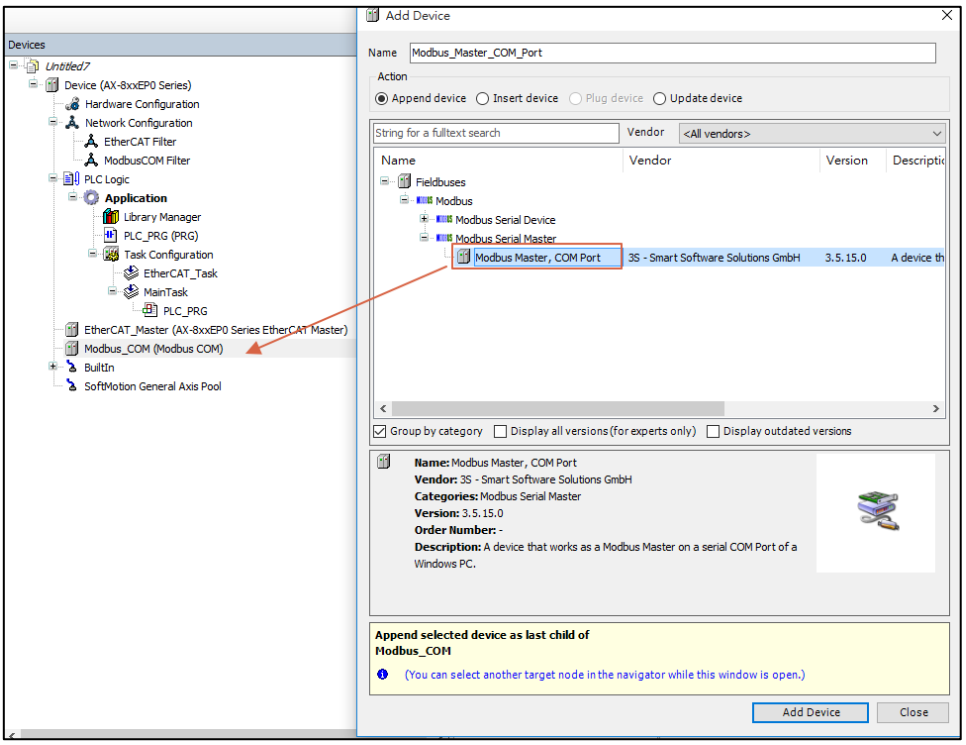


Figure 4 - 195: Adding MODBUS Master, COM Port

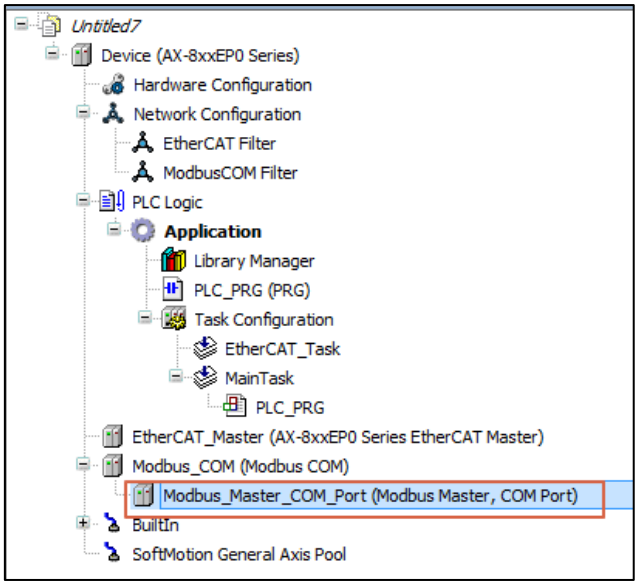


Figure 4 - 196: Adding MODBUS Master COM

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.

4.3.2.3 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*.

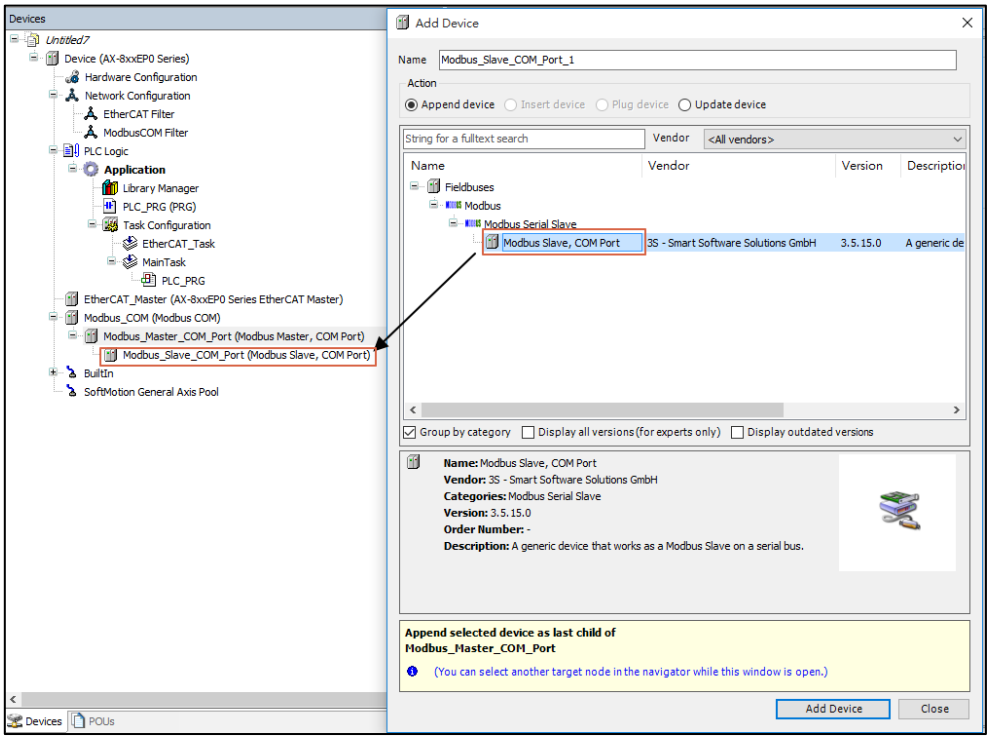


Figure 4 - 197: Adding MODBUS Slave COM port

- General

Item	Description
Slave address [1..247]	Address of a serial MODBUS device (value between 1 and 247)

Item	Description
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.

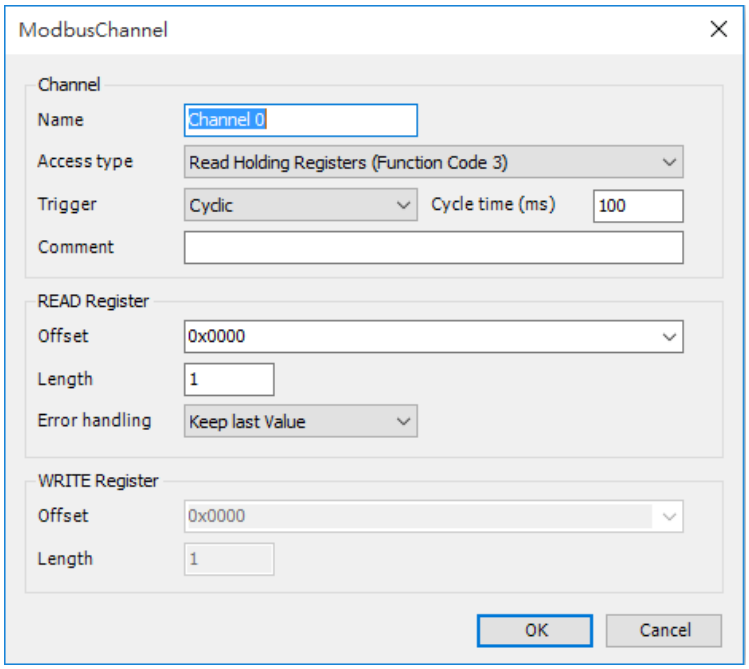


Figure 4 - 198: MODBUS Slave Channel window

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* • *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.

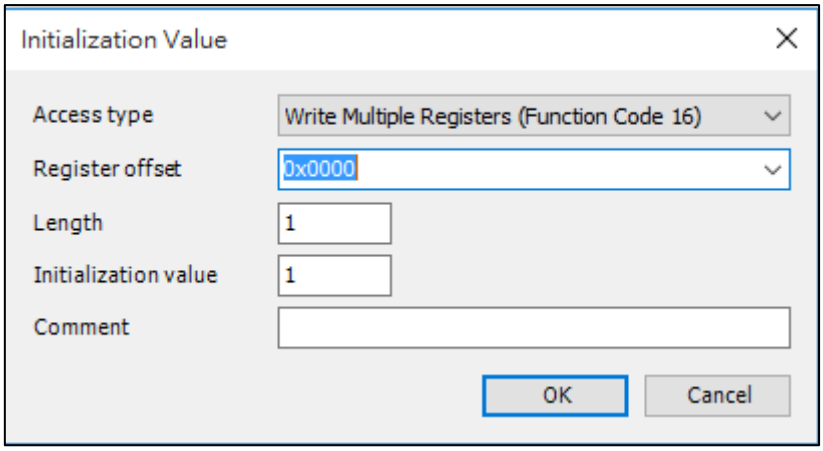


Figure 4 - 199: Initialization value

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

4.3.2.4 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.

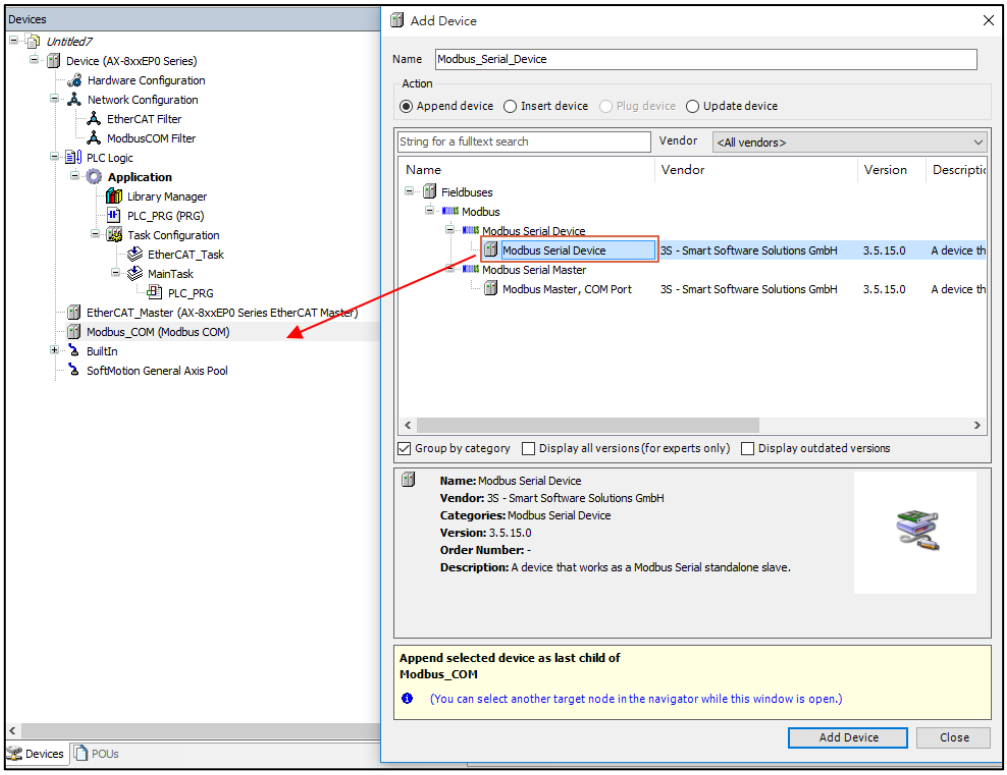


Figure 4 - 200: Adding MODBUS Serial Device

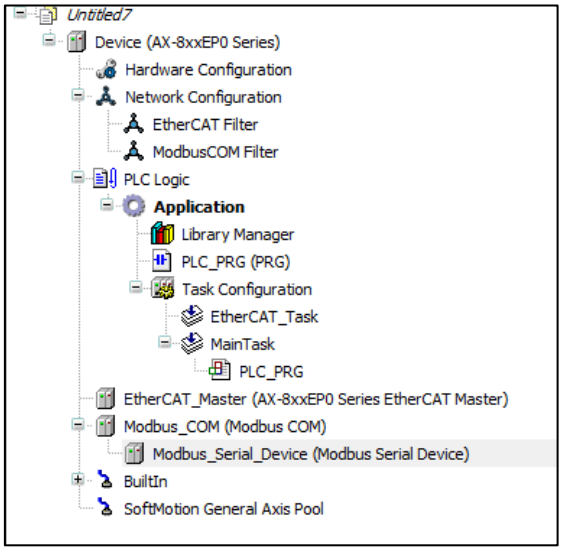


Figure 4 - 201: Project tree

- 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.

4.3.3 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.

4.3.3.1 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.

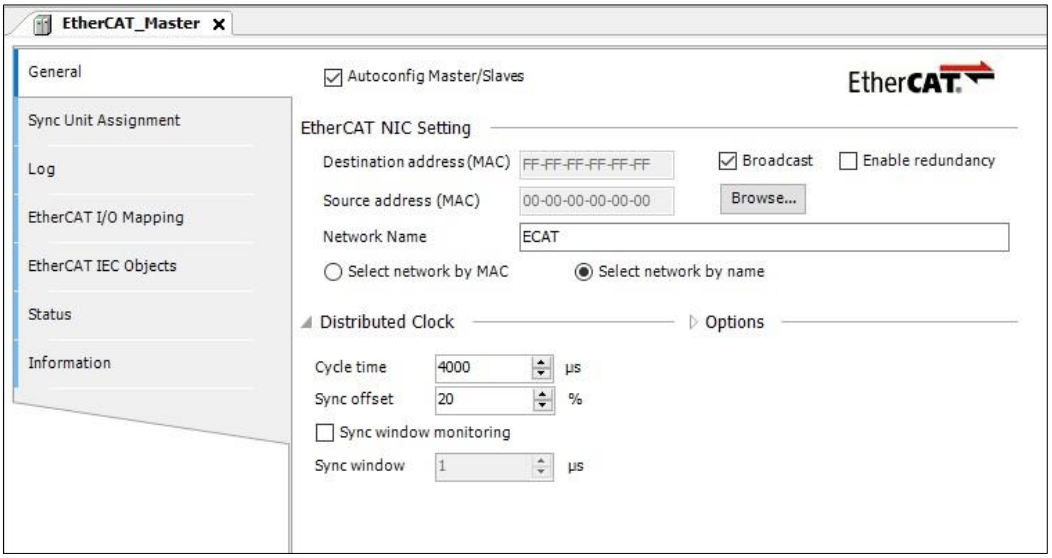


Figure 4 - 202: EtherCAT Master Window

- 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 triangle* 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 you 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.

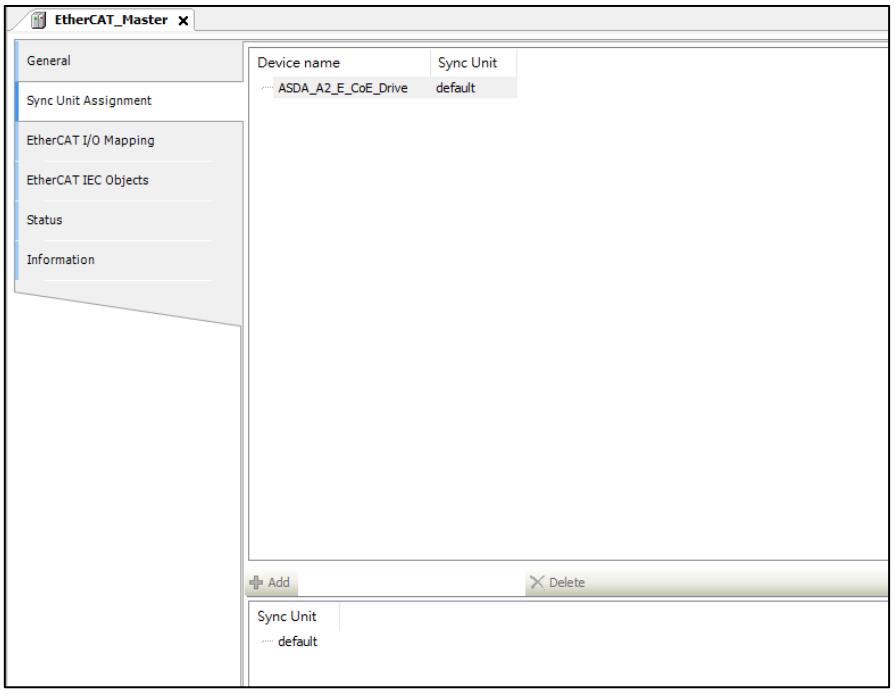


Figure 4 - 203: EtherCAT Master – Sync Unit Assignment Tab

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.

4.3.3.2 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.*

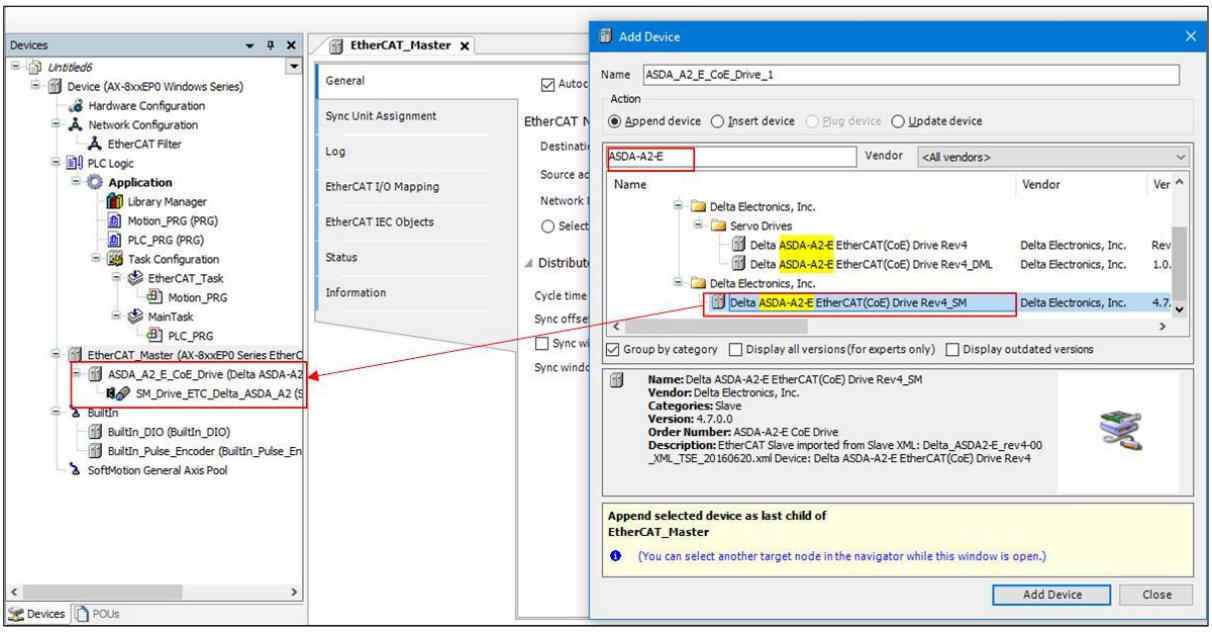


Figure 4 - 204: Adding ASDAA-A2-E-SM to device

- **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.

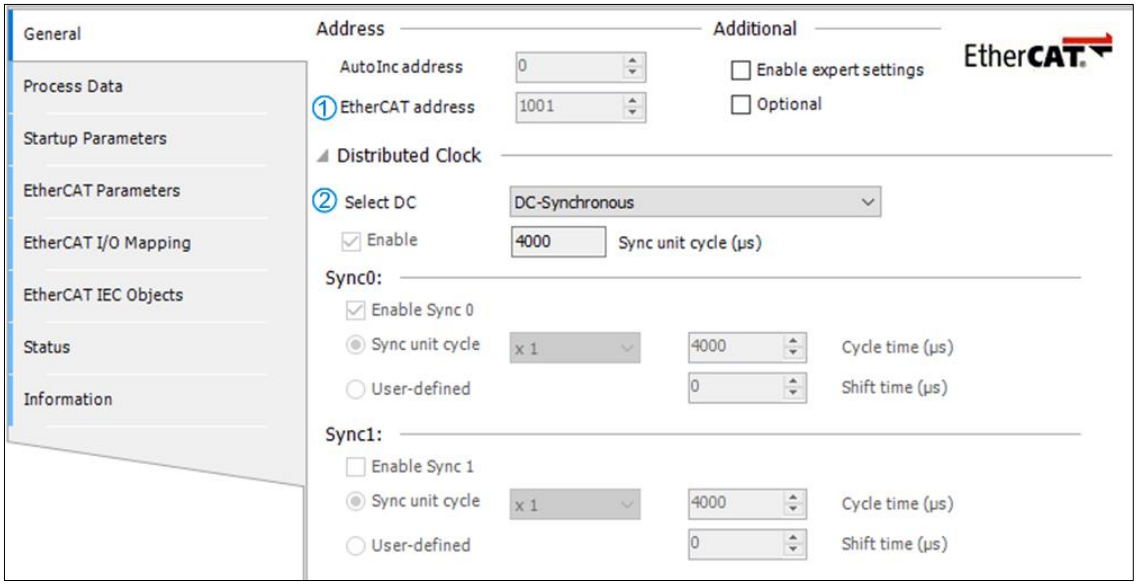


Figure 4 - 205: General Tab

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.

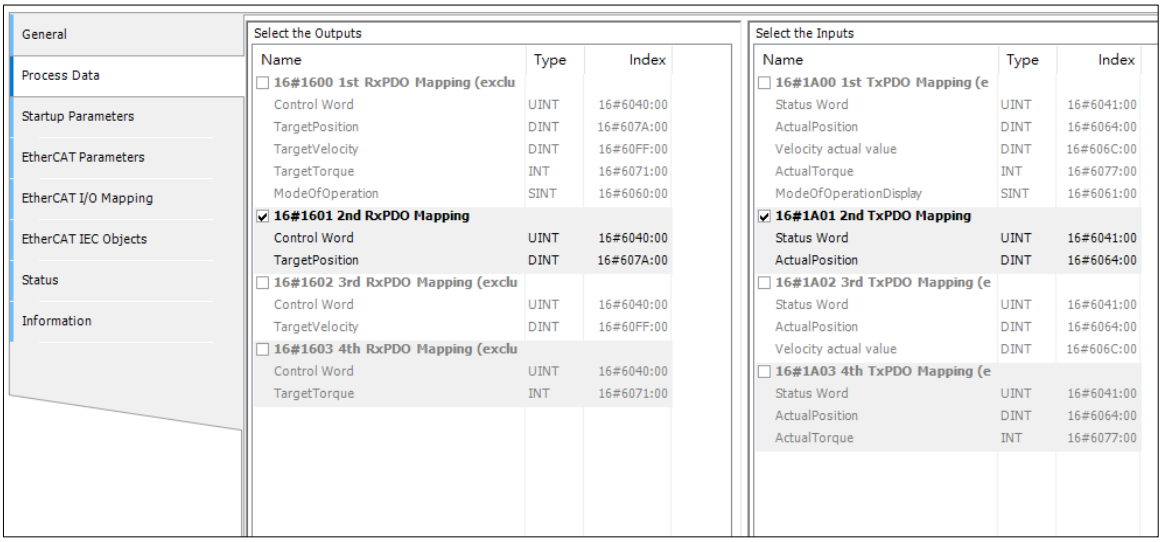


Figure 4 - 206: Process data Tab

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.

Item	Description
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.

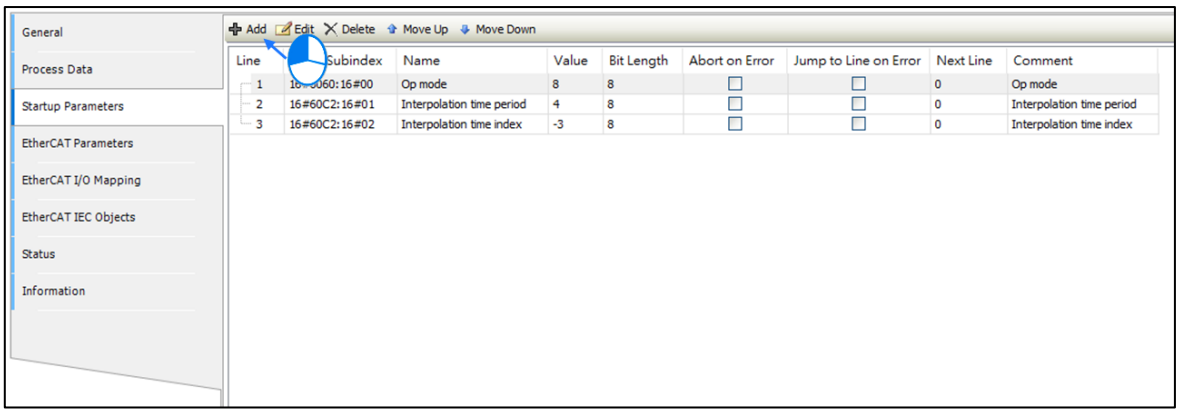


Figure 4 - 207: Startup parameter Tab

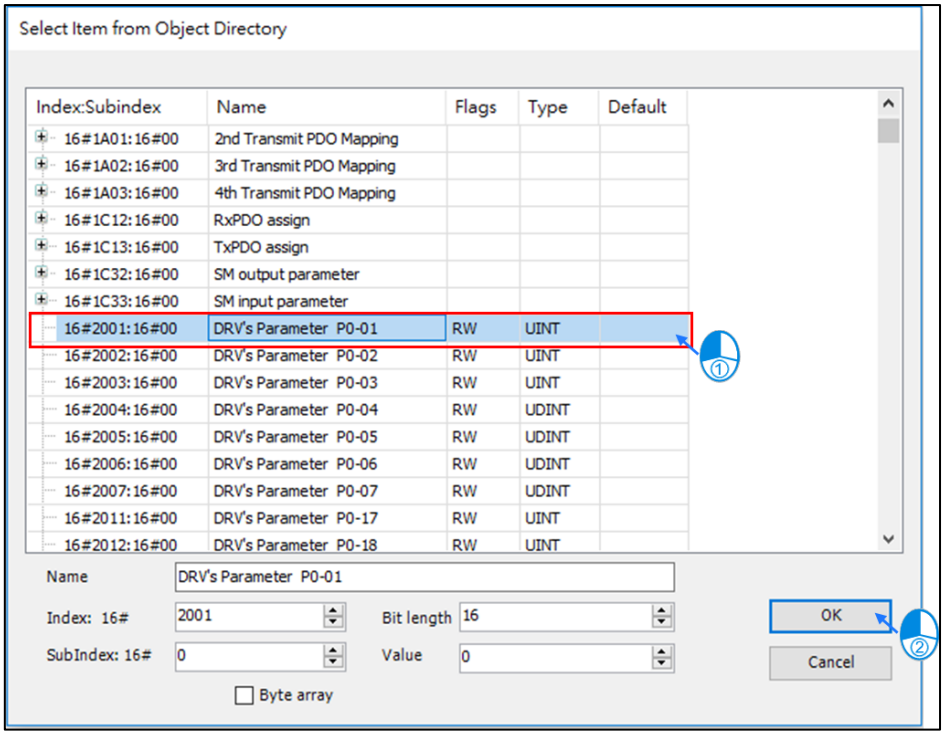


Figure 4 - 208: Parameter added

4.3.4 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*.

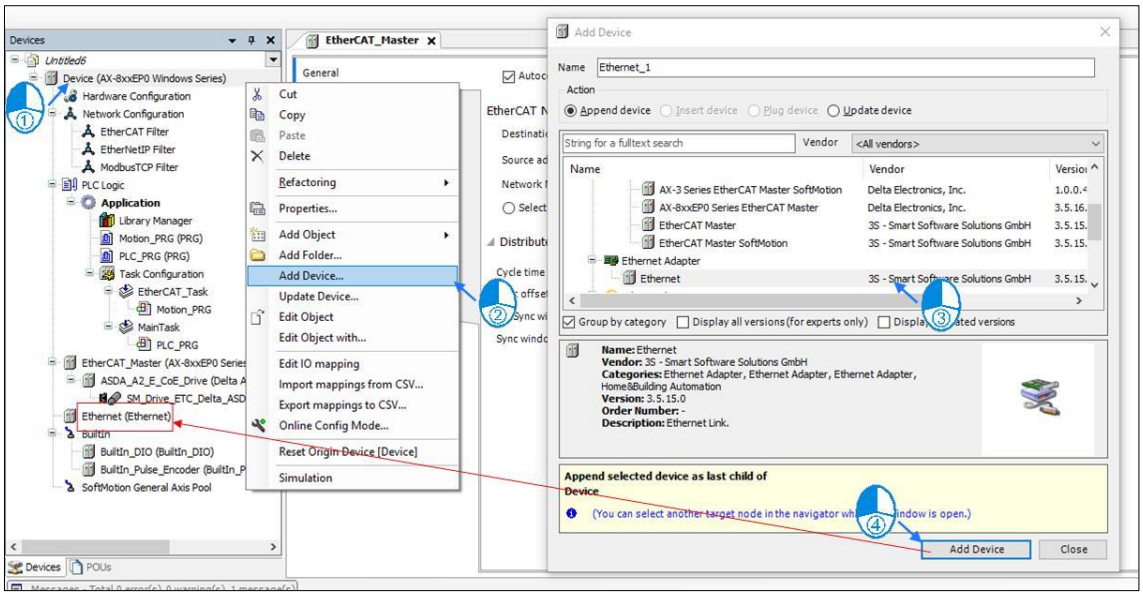


Figure 4 - 209: Adding Ethernet Adapter device

- EtherNet Device – General

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

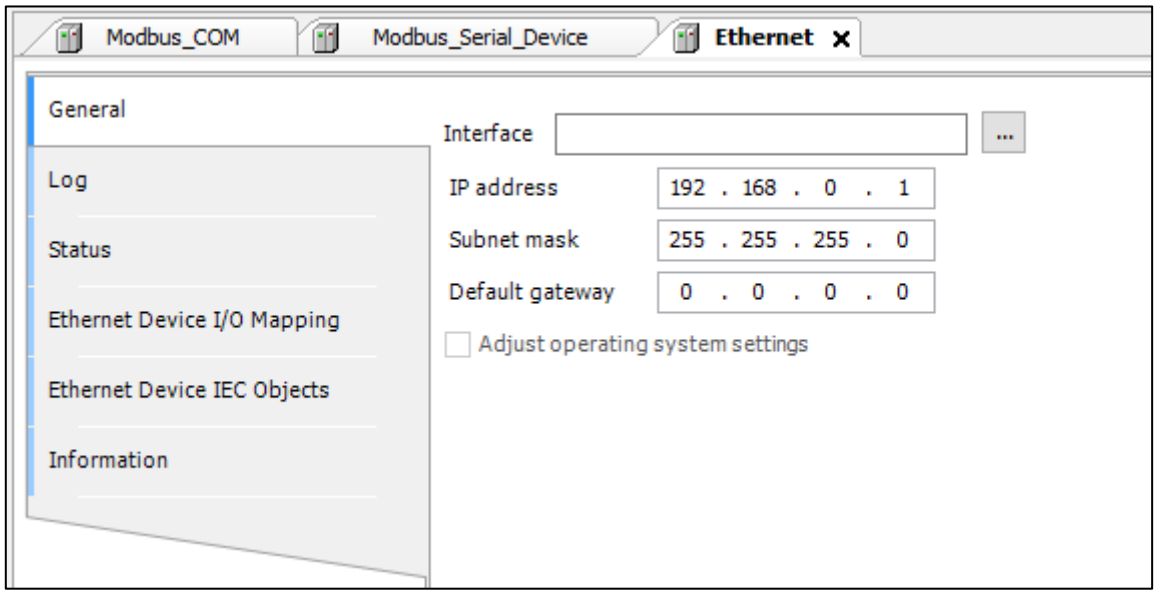


Figure 4 - 210: General Tab

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

4.3.4.1 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*.

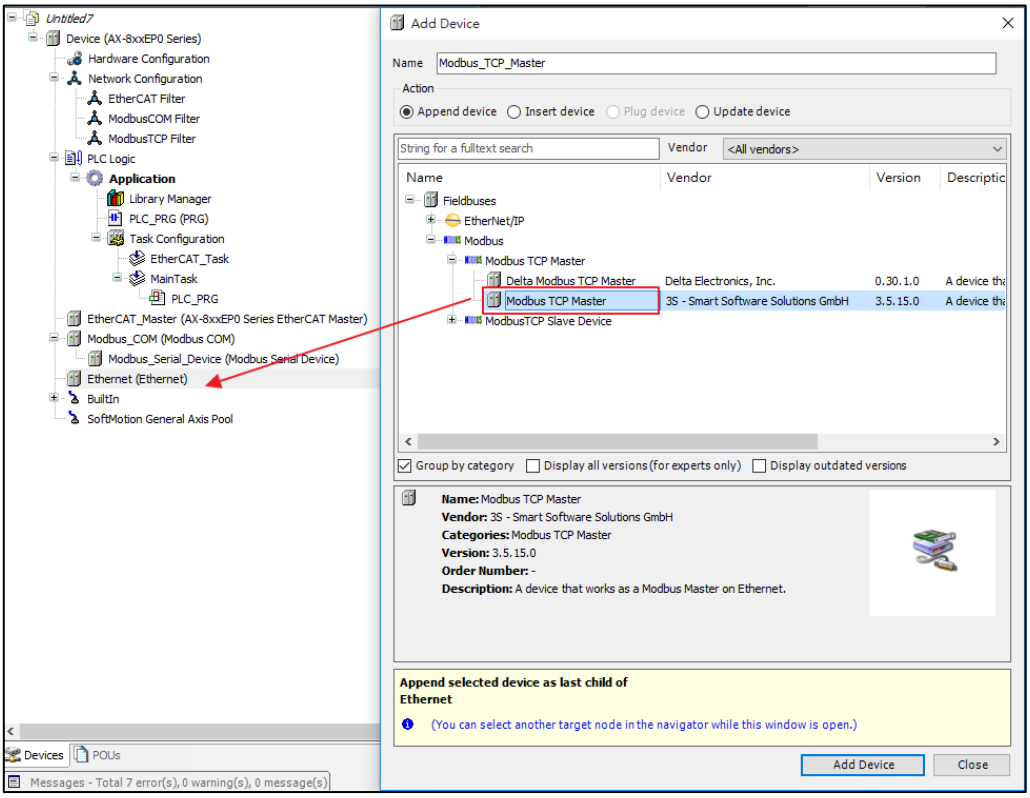


Figure 4 - 211: Adding MODBUS TCP Master

- General

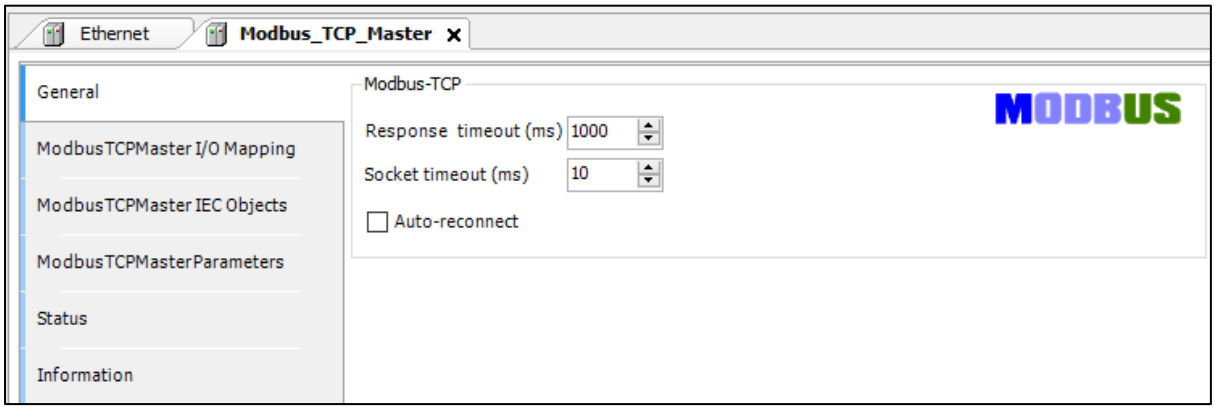


Figure 4 - 212: MODBUS TCP Master – General Tab

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.

4.3.4.2 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*.

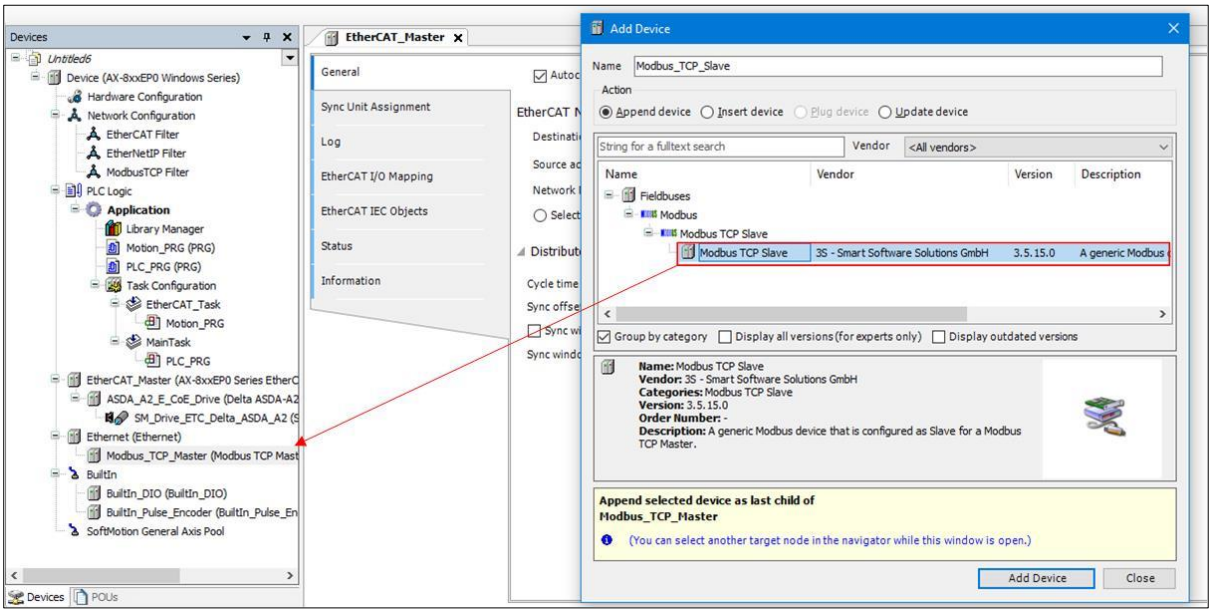


Figure 4 - 213: Adding MODBUS TCP Slave

- General

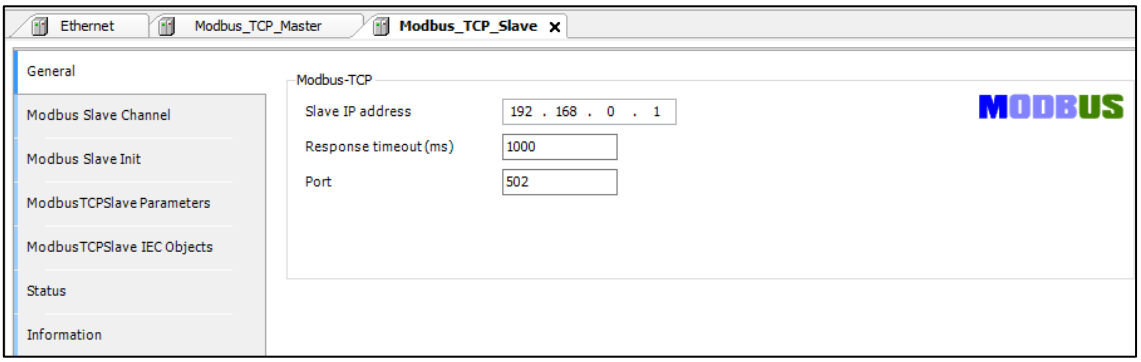


Figure 4 - 214: MODBUS TCP Slave – General Tab

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, you can predefine individual MODBUS registers or the description of the entire channel.

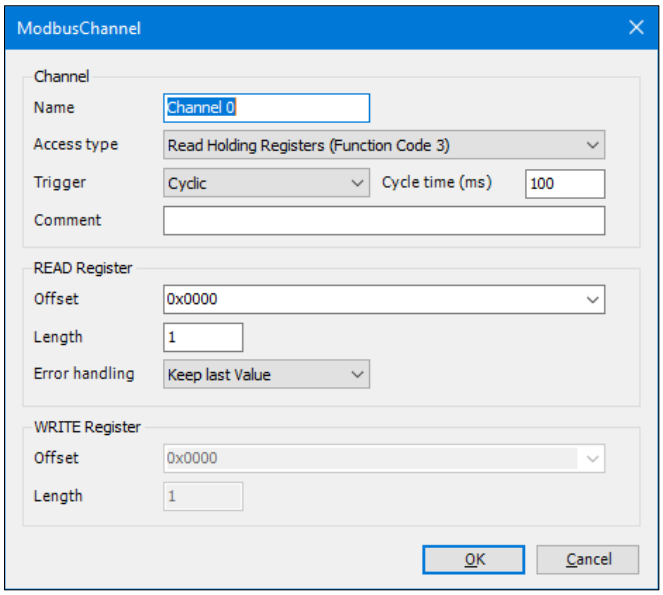


Figure 4 - 215: MODBUS Channel Window

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 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.

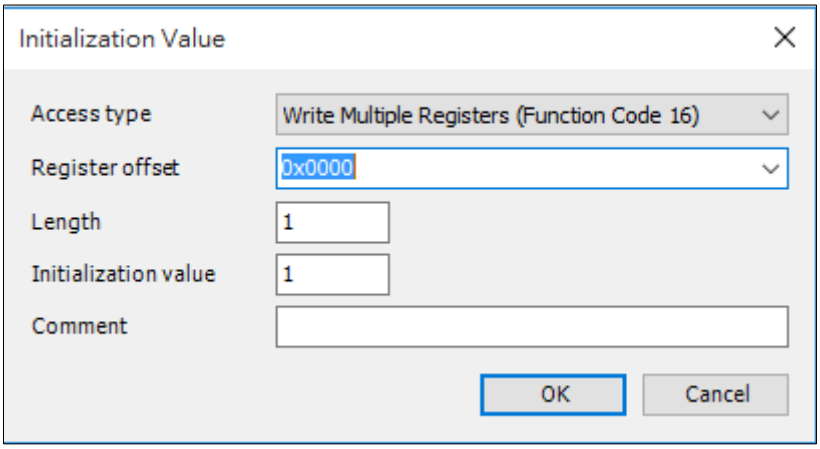


Figure 4 - 216: Initialization Value

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

4.3.4.3 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.*

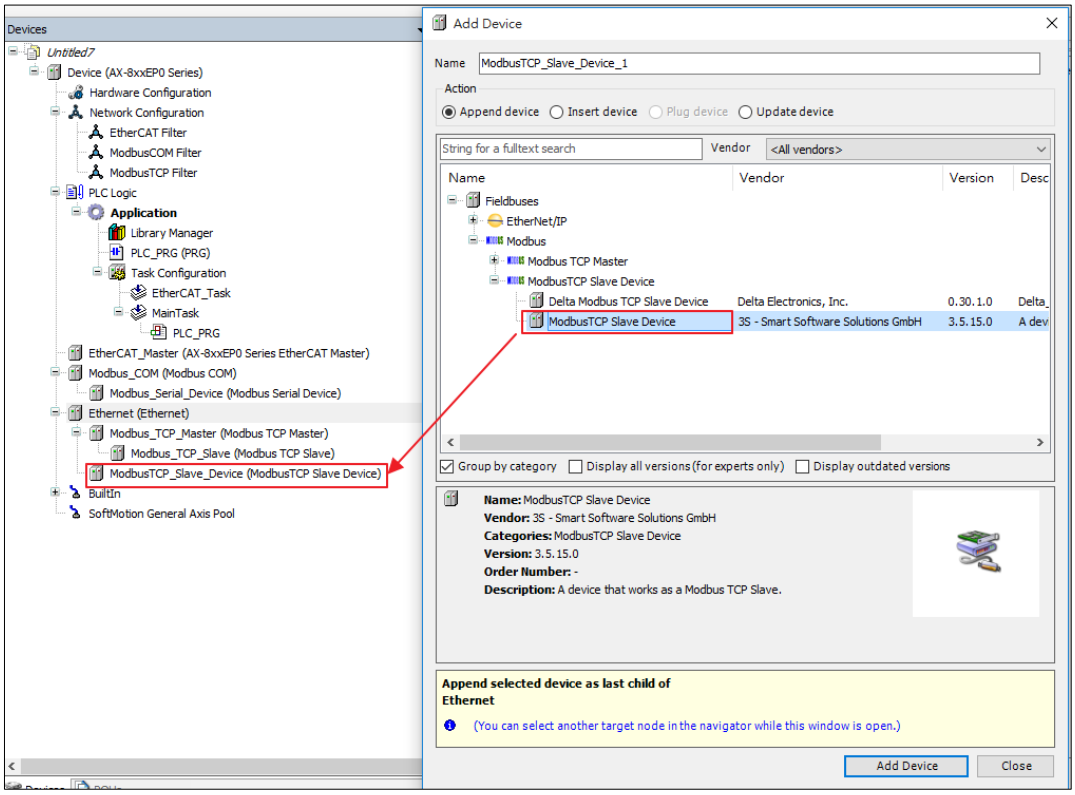


Figure 4 - 217: Adding MODBUS TCP Slave Device

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

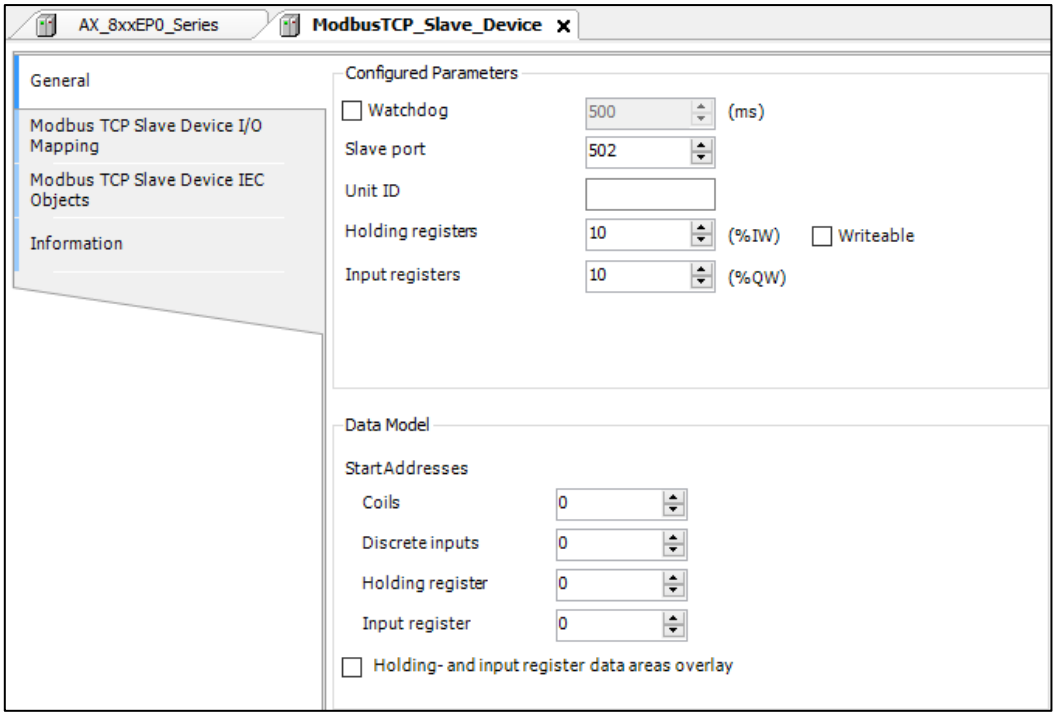


Figure 4 - 218: General Tab

- **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.
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.

4.3.5 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.

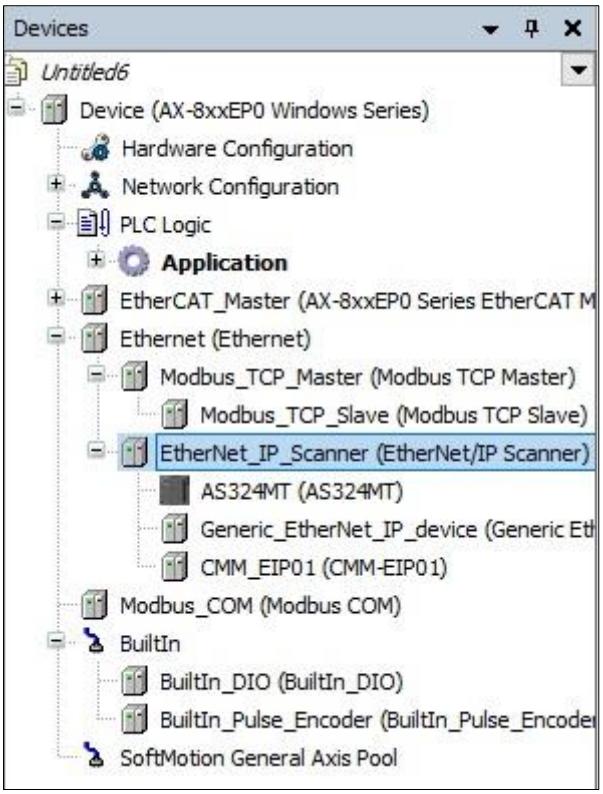


Figure 4 - 219: EtherNet/IP Configuration

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.

4.3.5.1 EtherNet/IP Scanner

AX-8xxEP0 Support EtherNet/IP scanner function. Right-click on the *Ethernet node* > *Add Device* > *Fieldbus* > *Ethernet IP* > *EthernetIP Scanner* > *EtherNet/IP Scanner.*

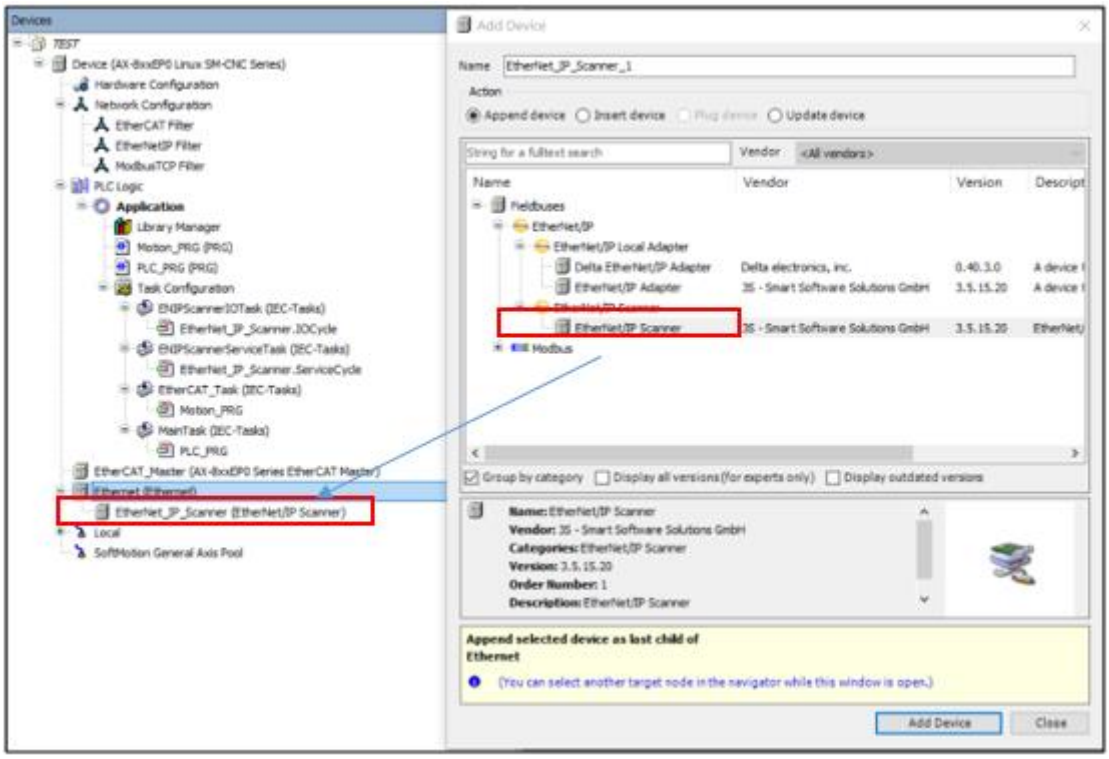


Figure 4 - 220: IncreaseEtherNet/IP Scanner

EtherNet/IP Scanner General tab

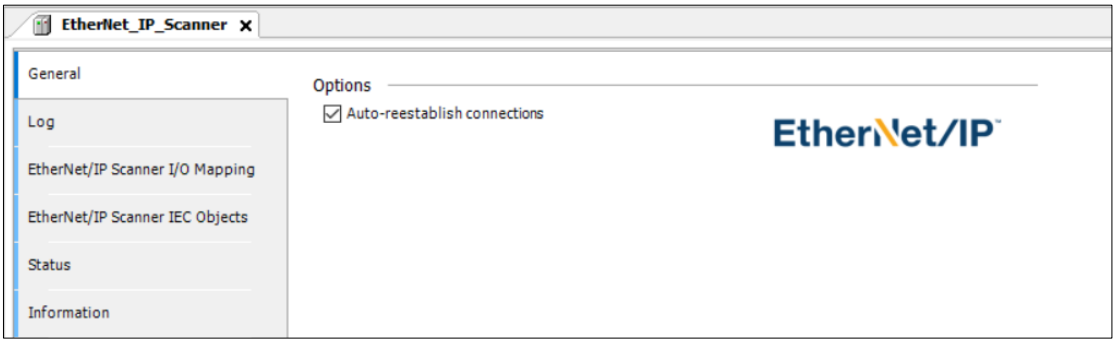


Figure 4 - 221: EtherNet/IP Scanner General tab

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

EtherNet/IP Scanner I/O Map tab

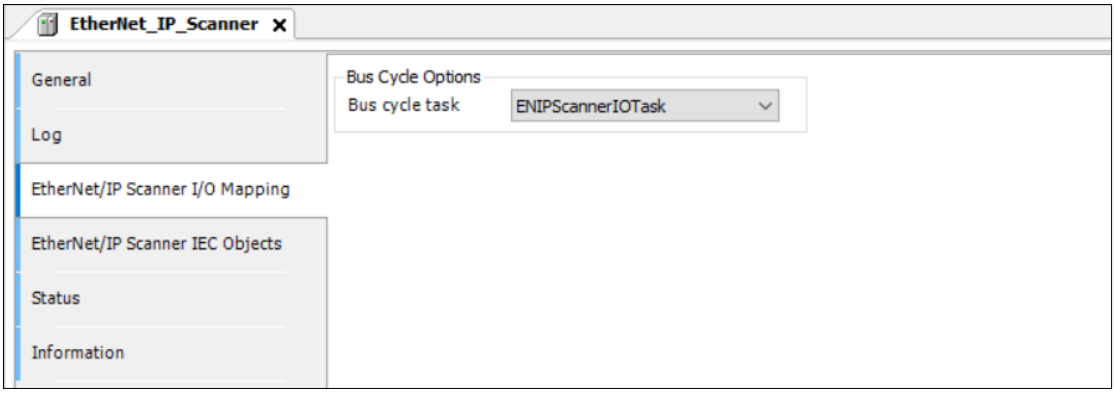


Figure 4 - 222: EtherNet/IP Scanner I/O Map tab

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

EtherNet/IP Scanner IEC Object tab

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

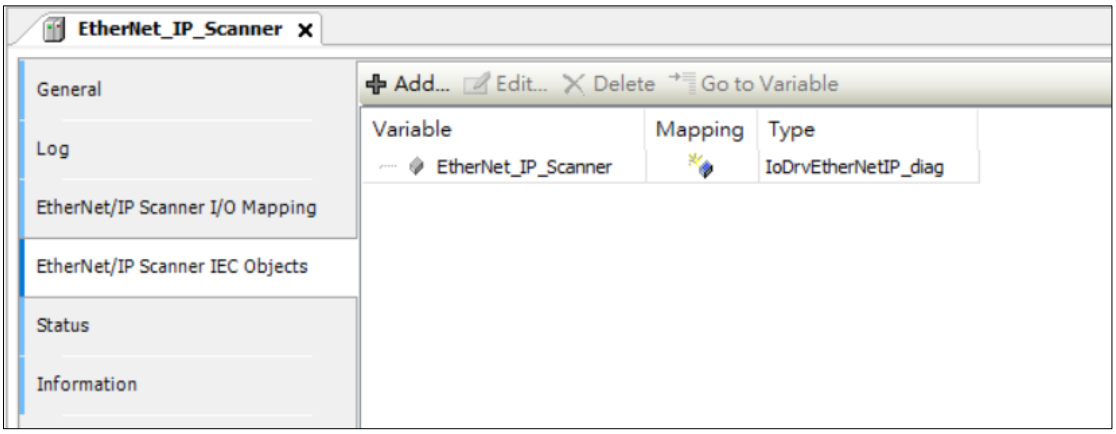


Figure 4 - 223: EtherNet/IP Scanner IEC Object tab

4.3.5.2 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*.

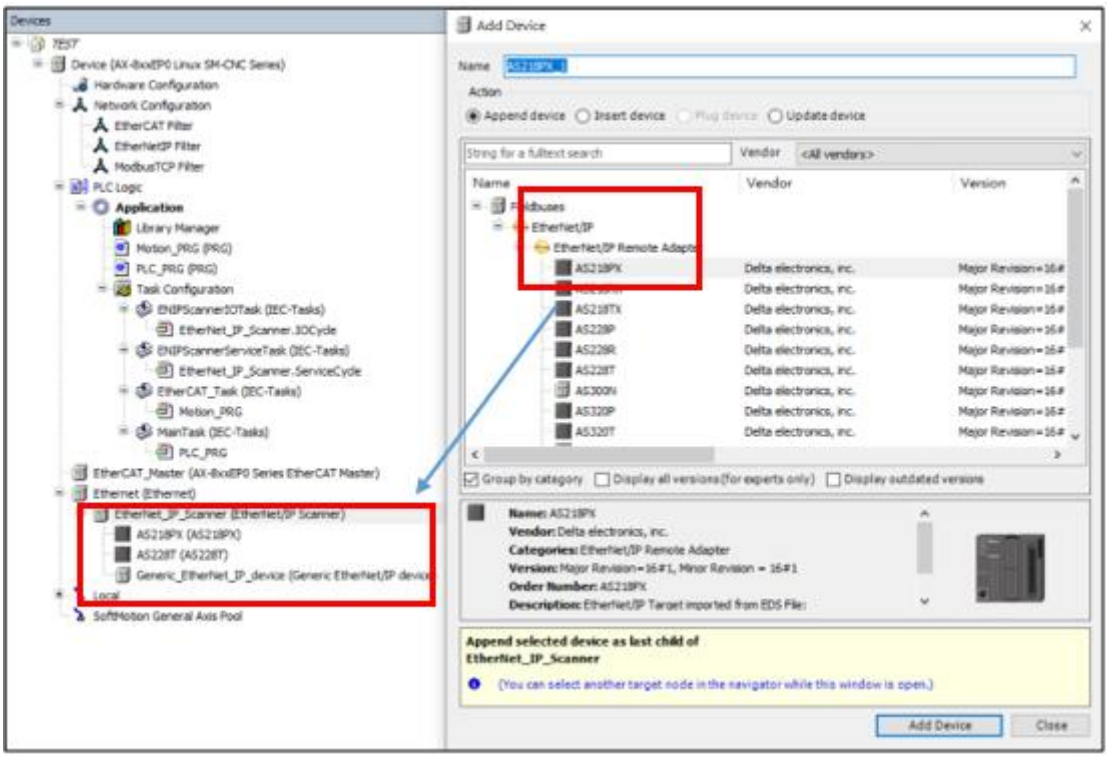


Figure 4 - 224: Add EtherNet/IP Remote Adapter

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

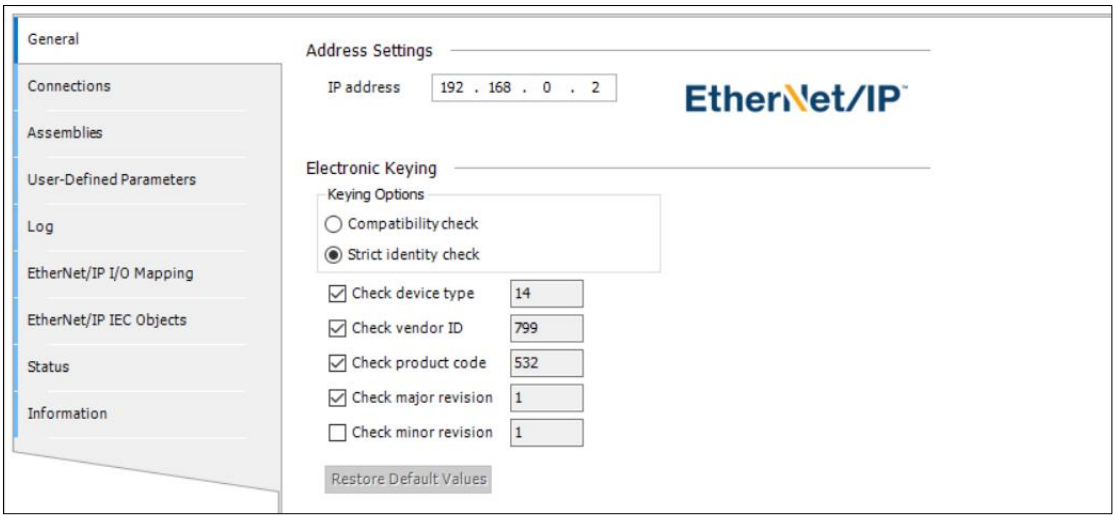


Figure 4 - 225: EtherNet/IP Remote Adapter General Tab

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.

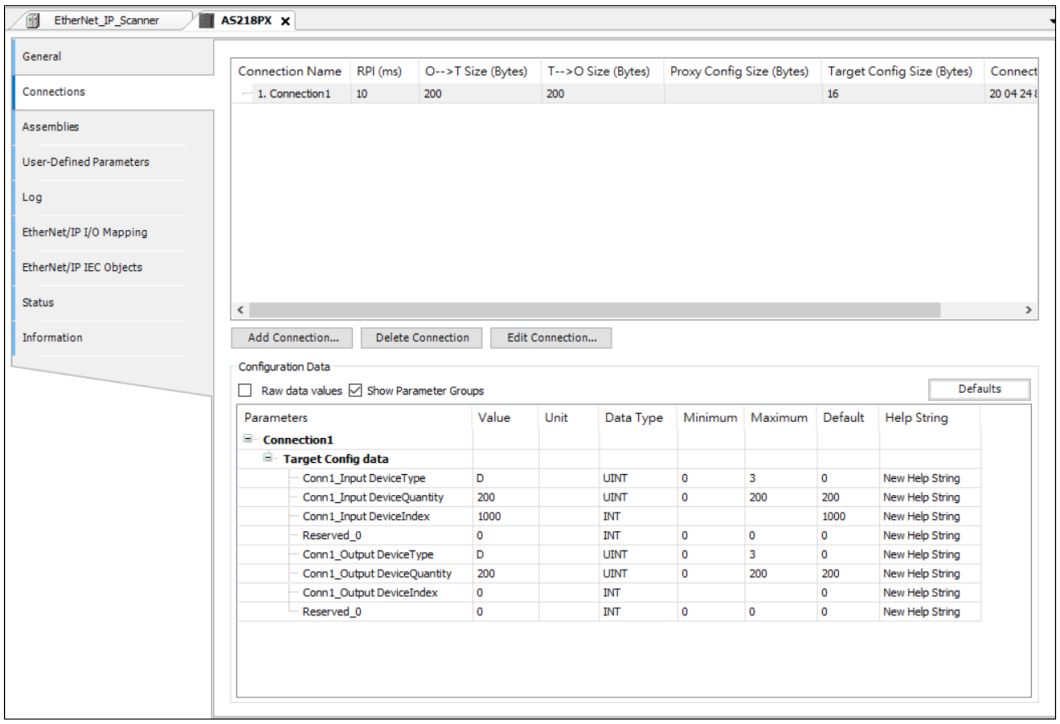


Figure 4 - 226: EtherNet/IP Remote Adapter Connection Tab

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
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.

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

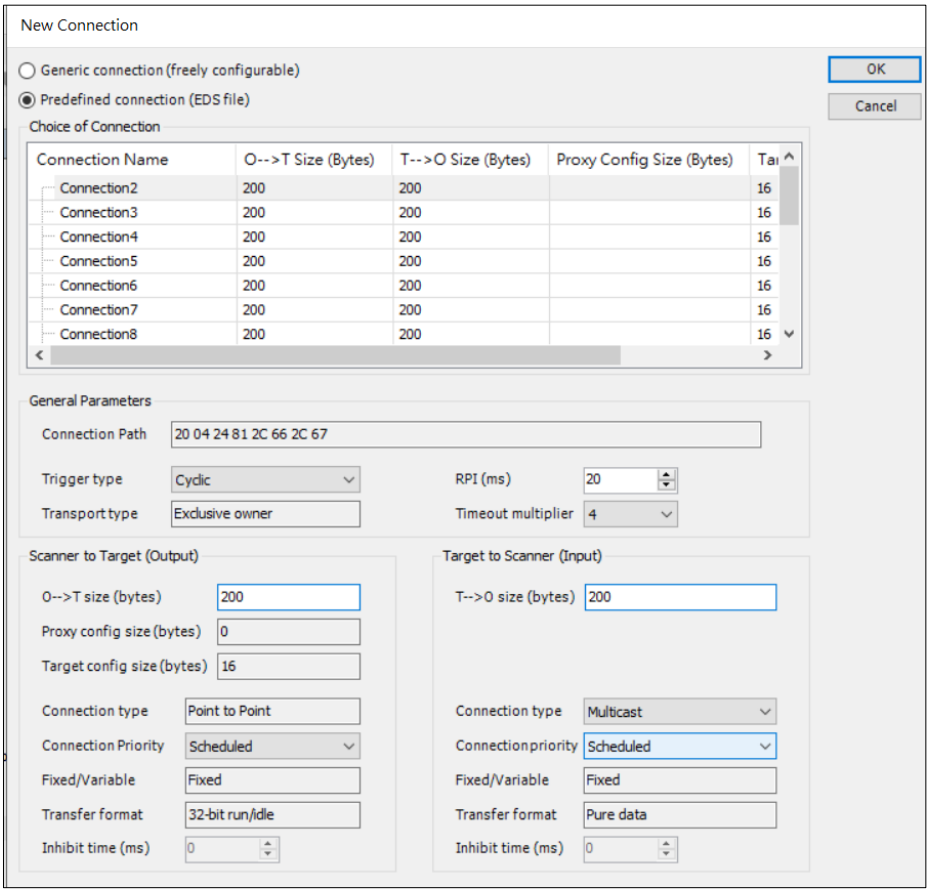


Figure 4 - 227: Add connection screen

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

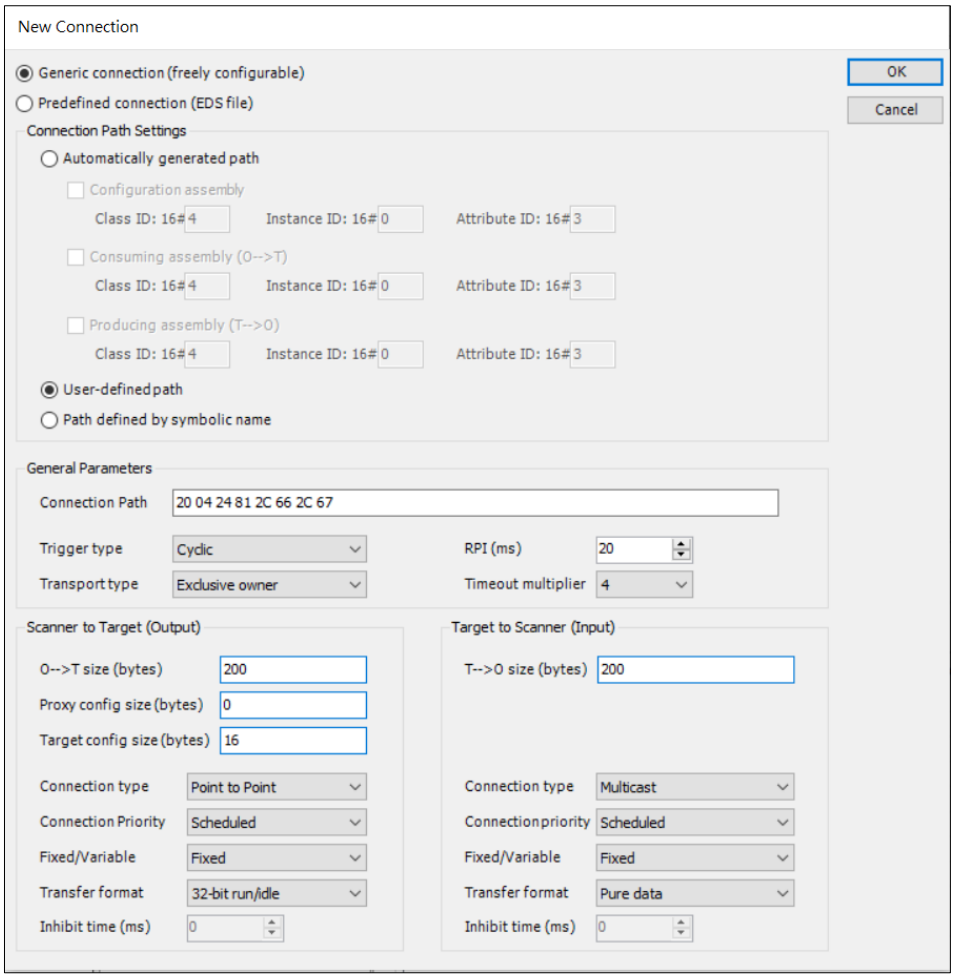


Figure 4 - 228: Universal connection (free configuration)

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	
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.

Project	Description
	***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.

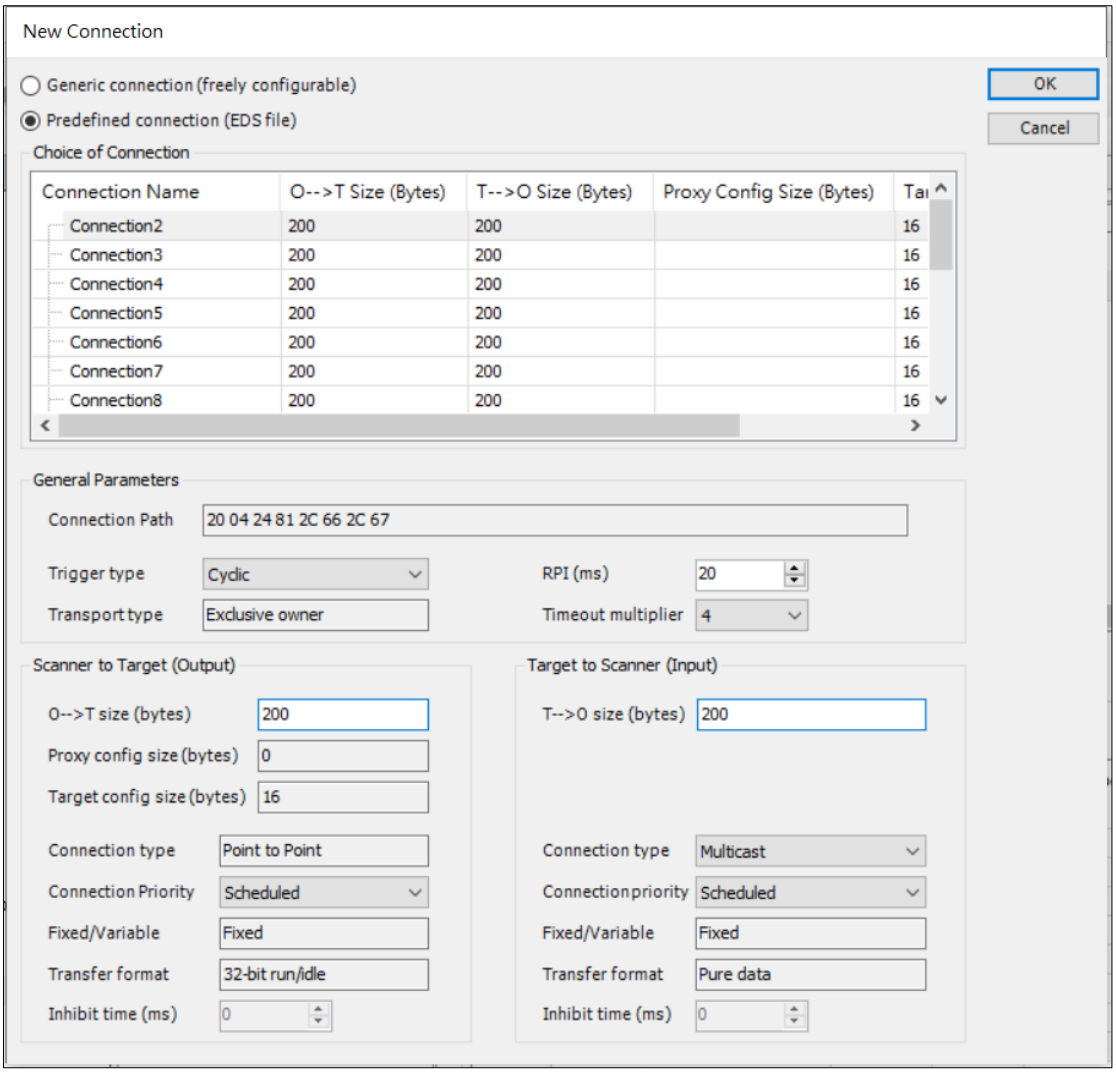


Figure 4 - 229: Predefined connection (EDS file)

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.
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

- **Configuration Data**

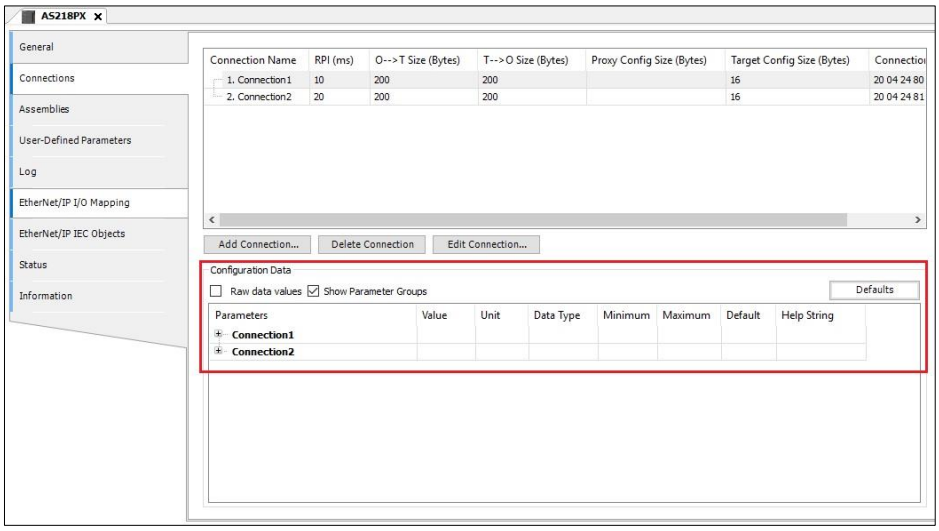


Figure 4 - 230: Configuration Data

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. • 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.

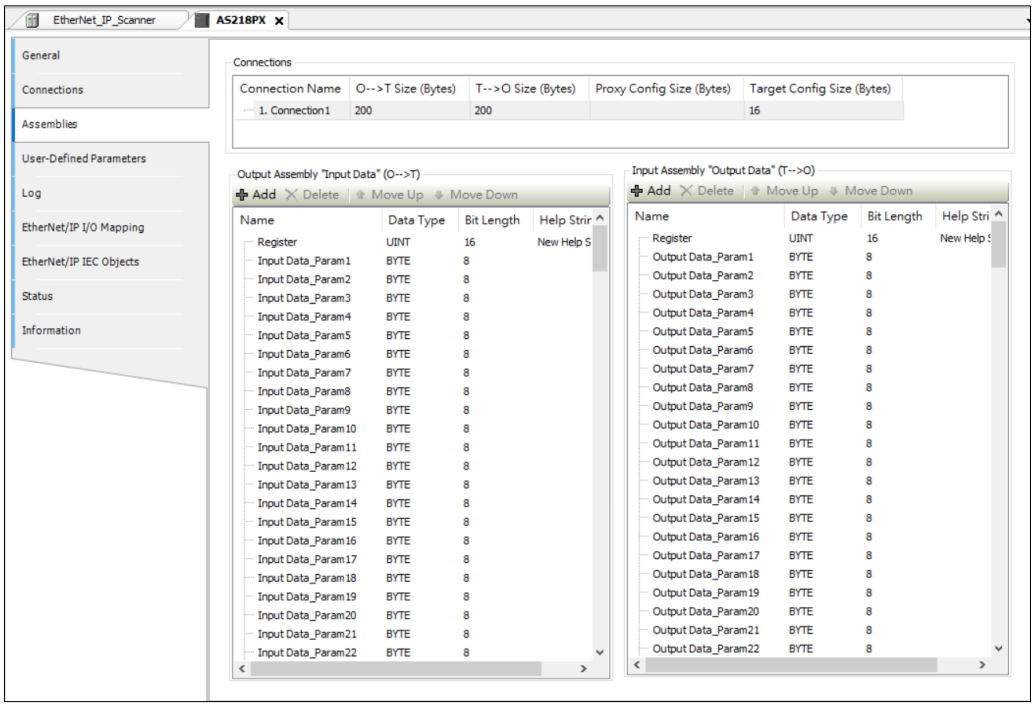


Figure 4 - 231: EtherNet/IP

- **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**

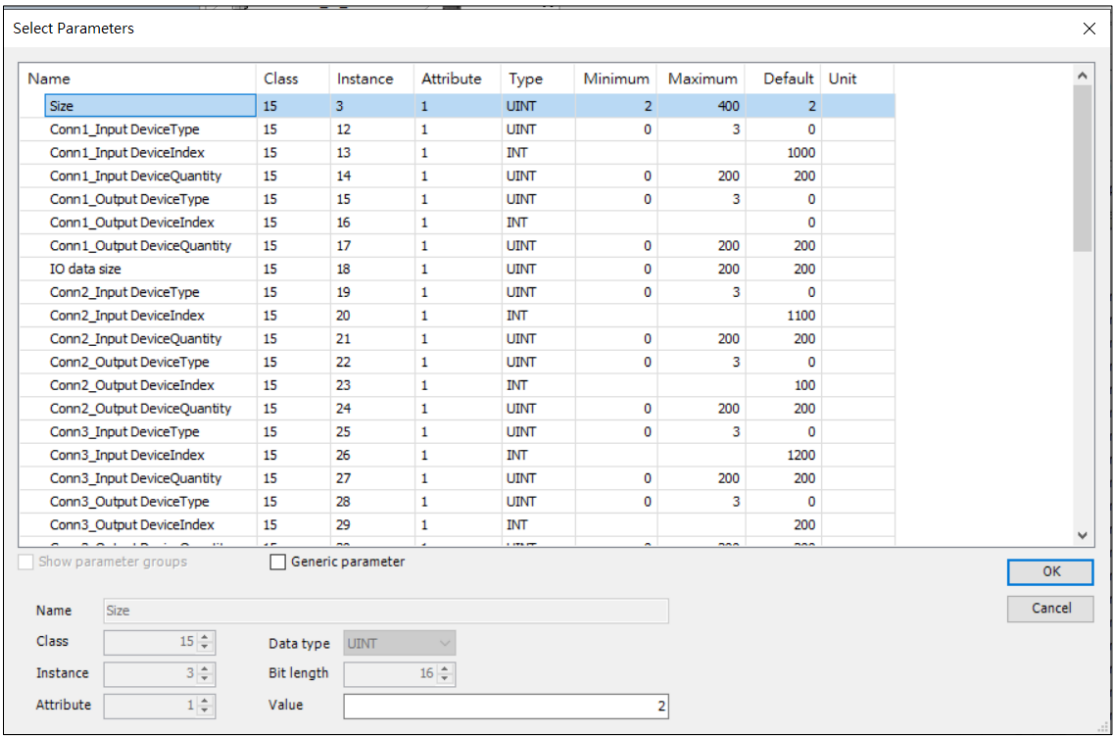


Figure 4 - 232: Select Parameters

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 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.

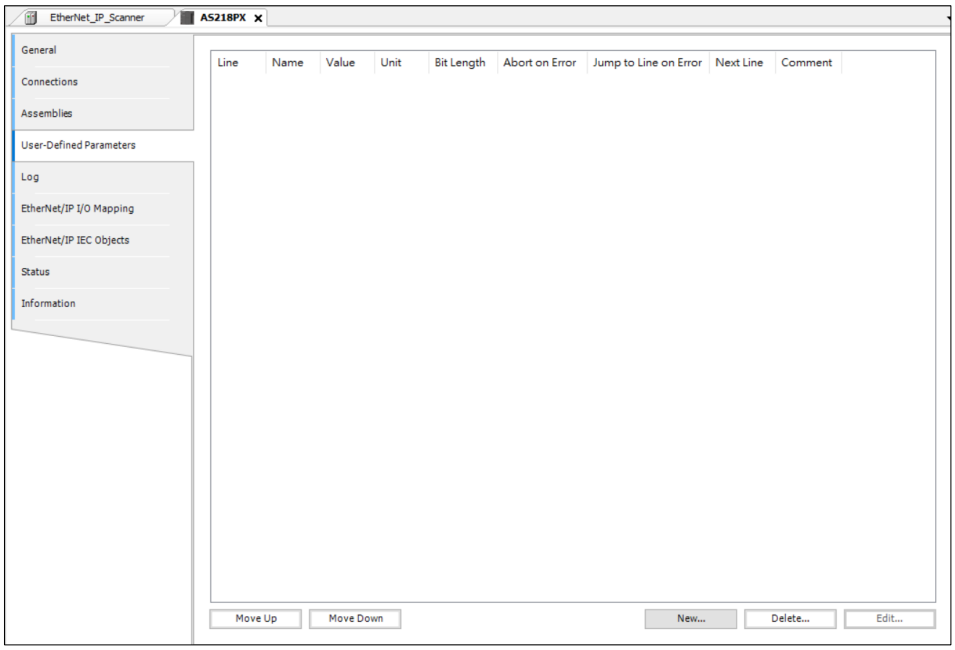


Figure 4 - 233: EtherNet/IP Remote adapter user parameters tab

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 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.*

4.3.5.3 EtherNet/IP Local Adapter

The AX-8xxEP0 series supports EtherNet/IP local adapter function. Right-click on the *Ethernet node*> *Add Device*> *Fieldbus*> *EtherNet/IP*> *EtherNet/IP Module* > *EtherNet/IP Module*.

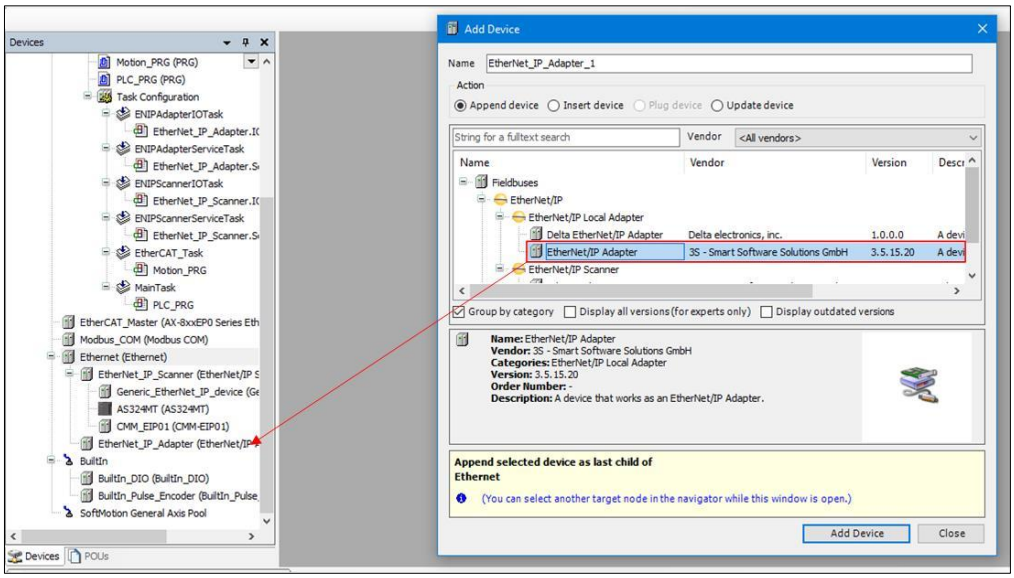


Figure 4 - 234: EtherNet/IP Local adapter

AX-8xxEP0 Set as the EtherNet IP Module as follows:

1. Right-click on *EtherNet_IP_Adapter (EtherNet_IP_Adapter)* > *Add Device* > *Fieldbus* > *EtherNet/IP* > *EtherNet/IP Local Adapter* > *EtherNet/IP Module*.

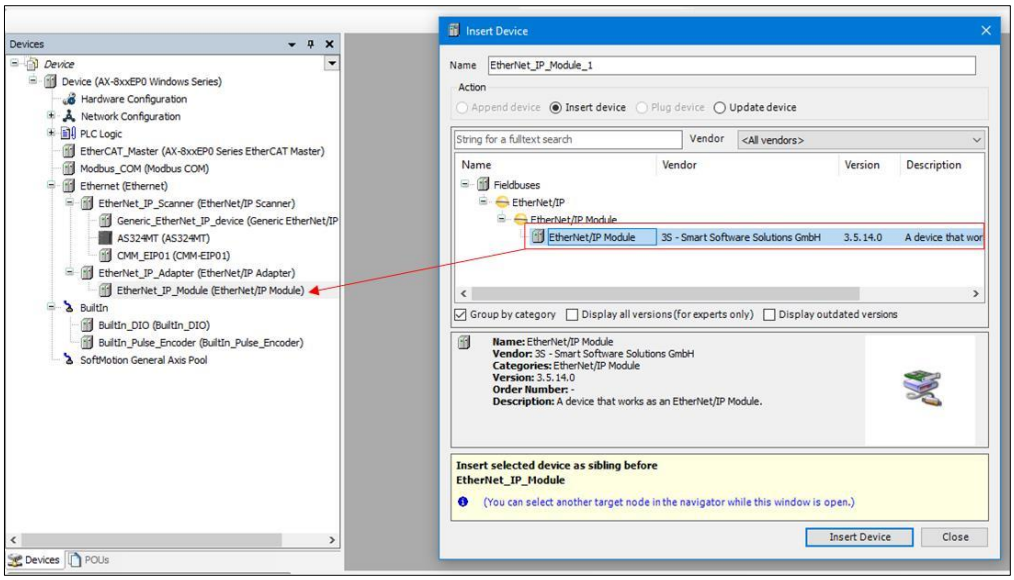


Figure 4 - 235: AX-8xxEP0 Adapter settings

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> General tab and change the name).

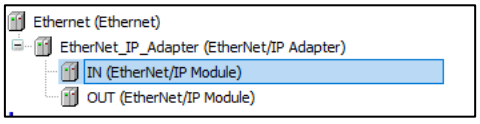


Figure 4 - 236: Add Delta EtherNet/IP Module

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

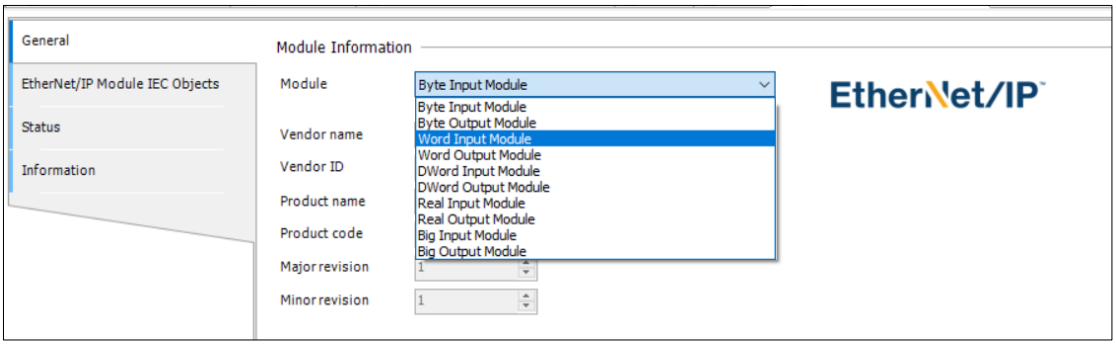


Figure 4 - 237: Word Input Module

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

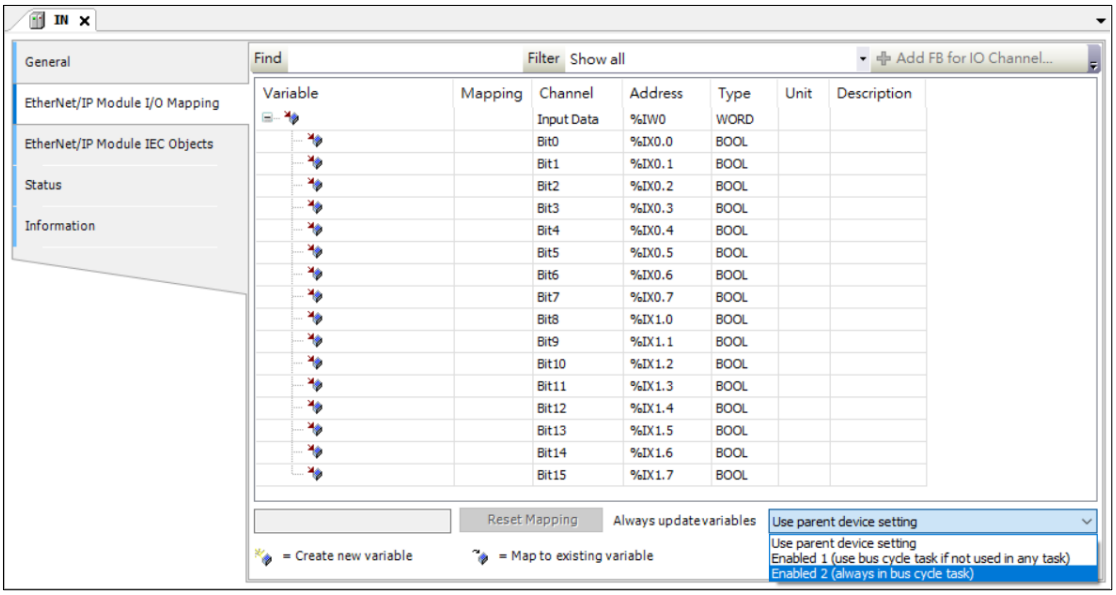


Figure 4 - 238: EtherNet/IP Module I/O Map tab

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

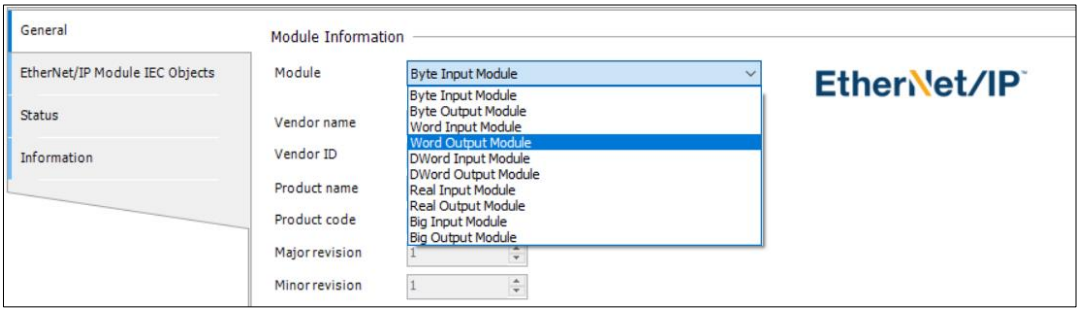


Figure 4 - 239: Word Output Module

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

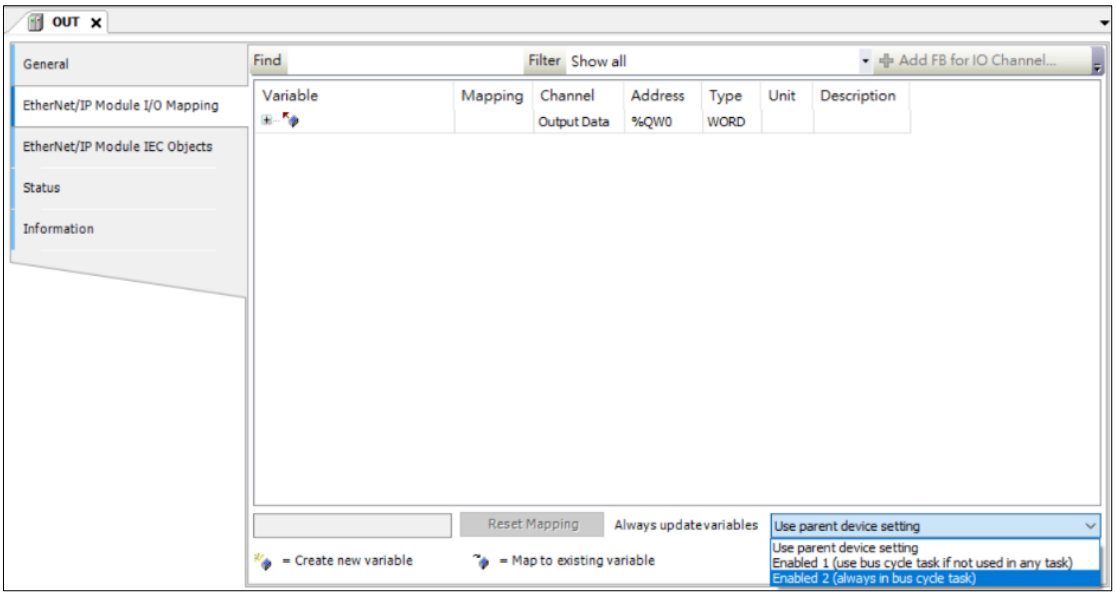


Figure 4 - 240: EtherNet/IP Module I/O Map tab

7. Double-click on *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 the third-party configuration environment.

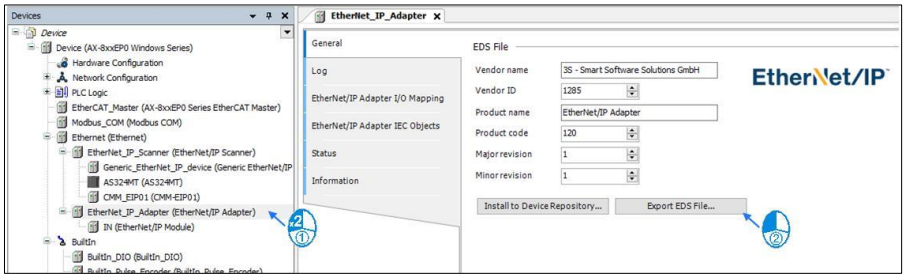


Figure 4 - 241: Export EDS file

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.

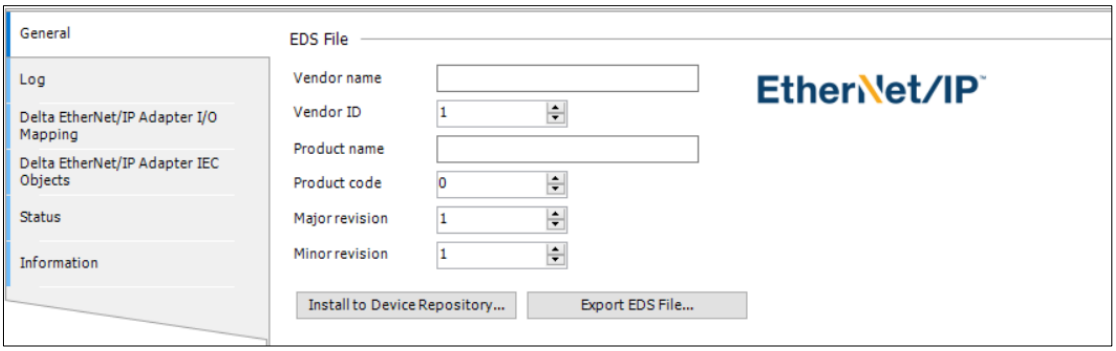


Figure 4 - 242: EtherNet/IP Local adapter general tab

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.

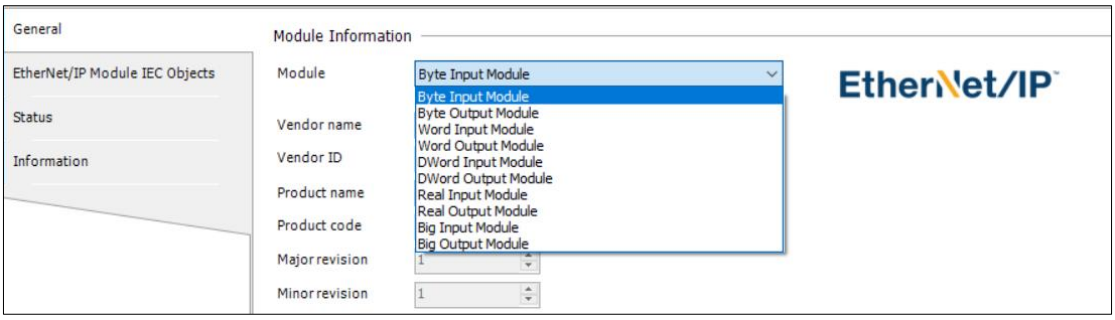


Figure 4 - 243: EtherNet/IP Module General Tab

4.3.6 High-Speed IO Setting

4.3.6.1 BuiltIn_DIO Parameter

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.*

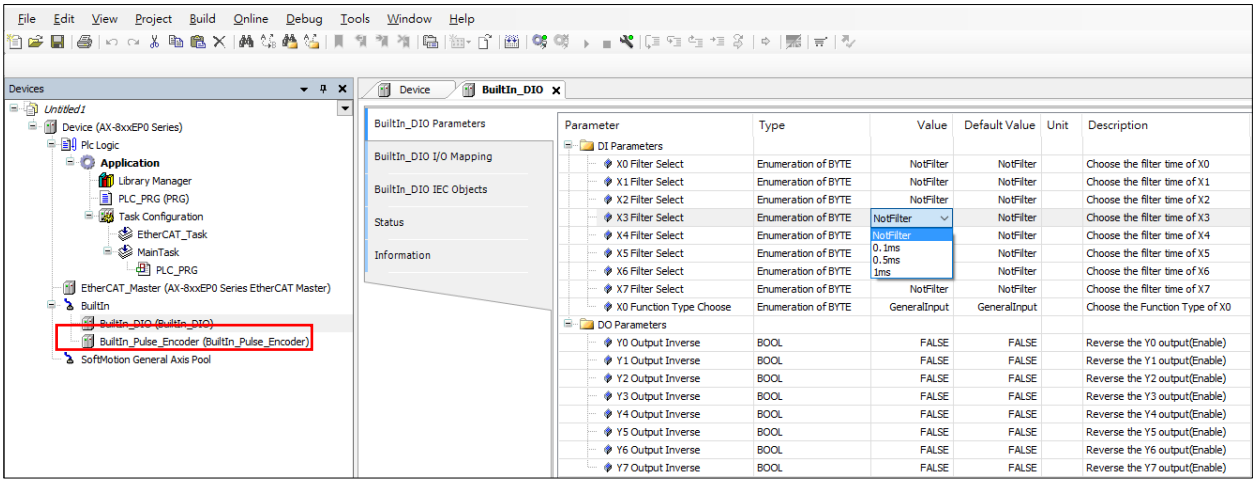


Figure 4 - 244: BuiltIn DIO Parameter

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.*

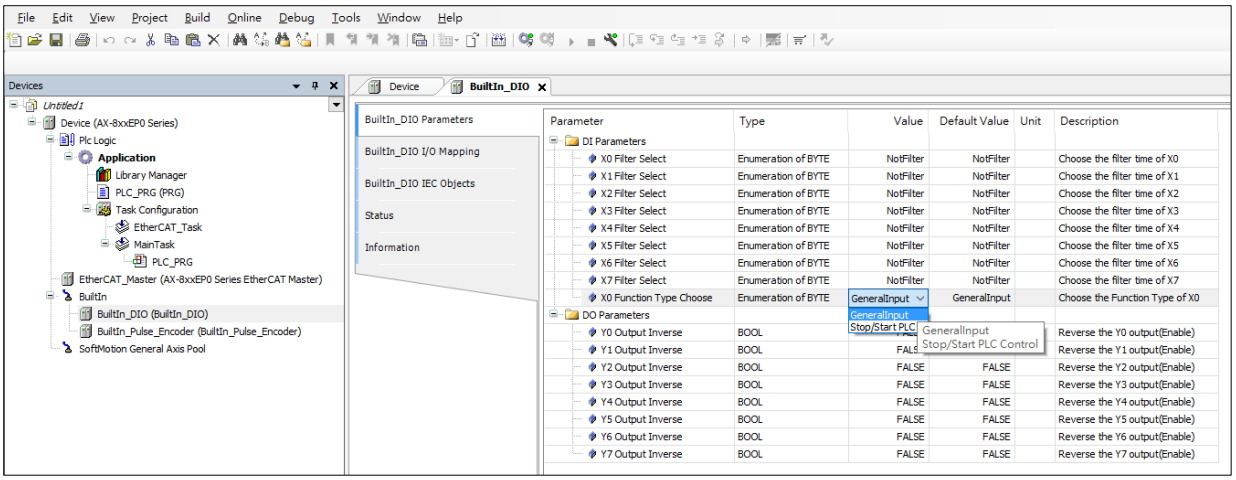


Figure 4 - 245: Entering Input

***NOTE**: Stop/Start PLC Control Timing diagram.*

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

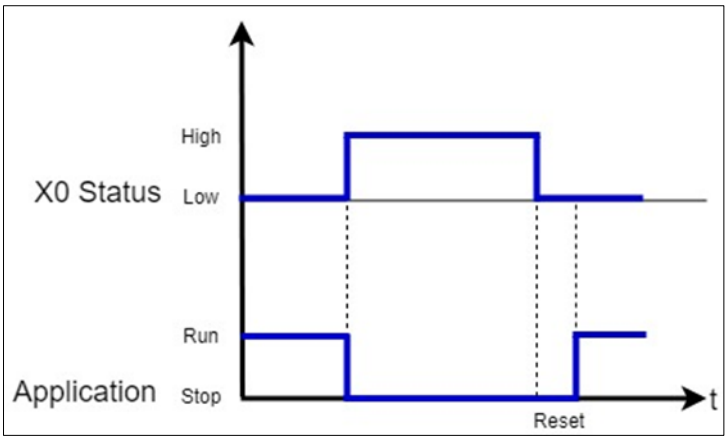


Figure 4 - 246: PLC Control Timing diagram

4.3.6.2 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)

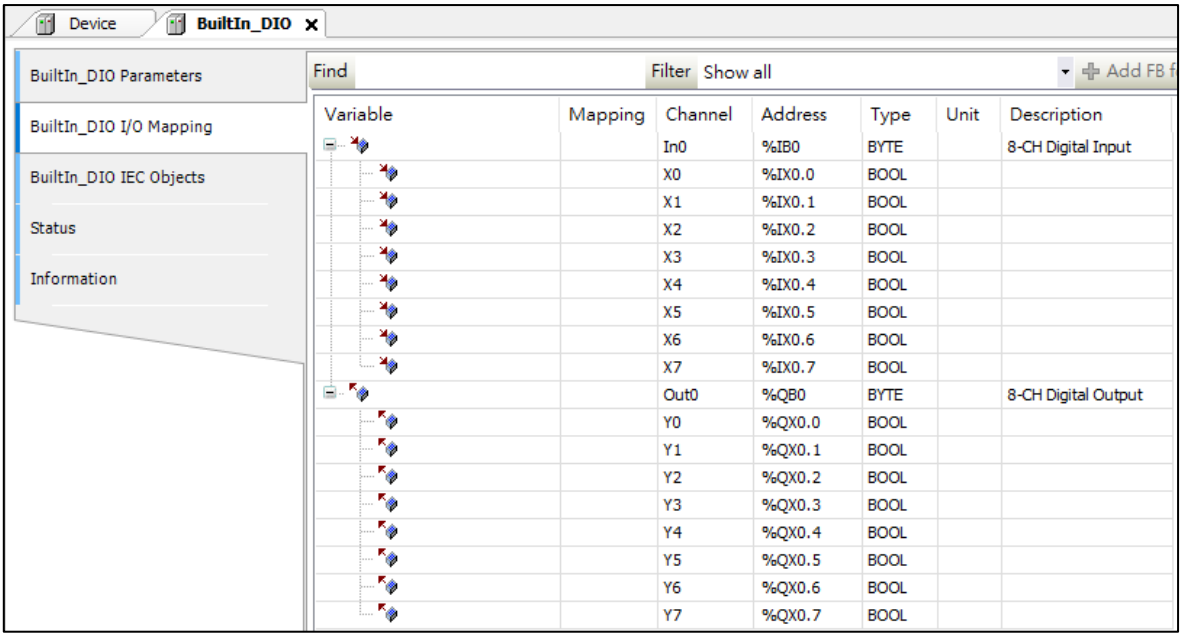


Figure 4 - 247: BuiltIn DIO I/O Mapping

4.3.6.3 BuiltIn_Pulse_Encoder Parameter

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

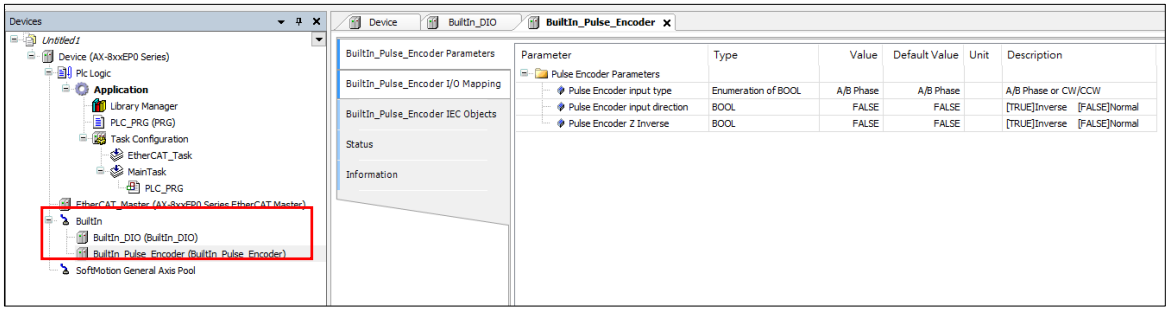


Figure 4 - 248: BuiltIn Pulse Encoder Parameter

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

4.3.6.4 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)

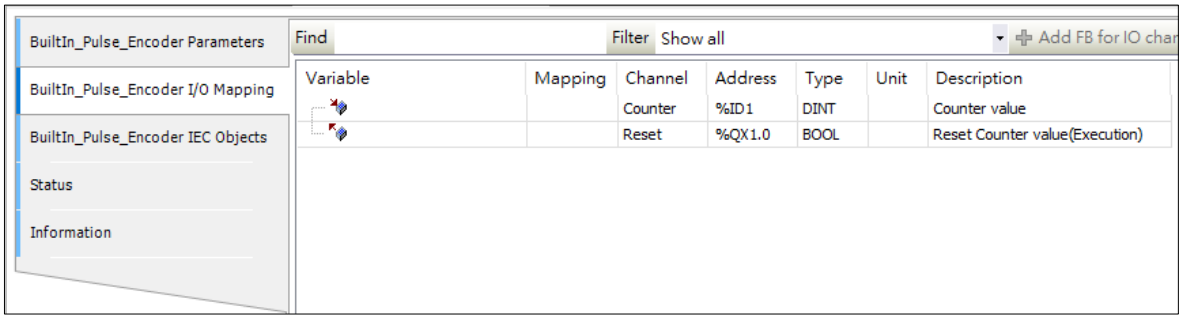


Figure 4 - 249: BuiltIn Pulse Encoder I/O Mapping

- Reset Timing diagram

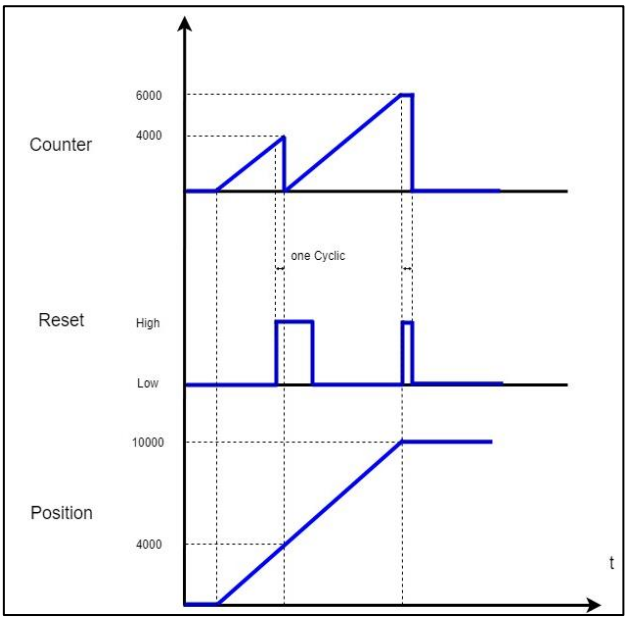


Figure 4 - 250: Reset Timing Diagram

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.

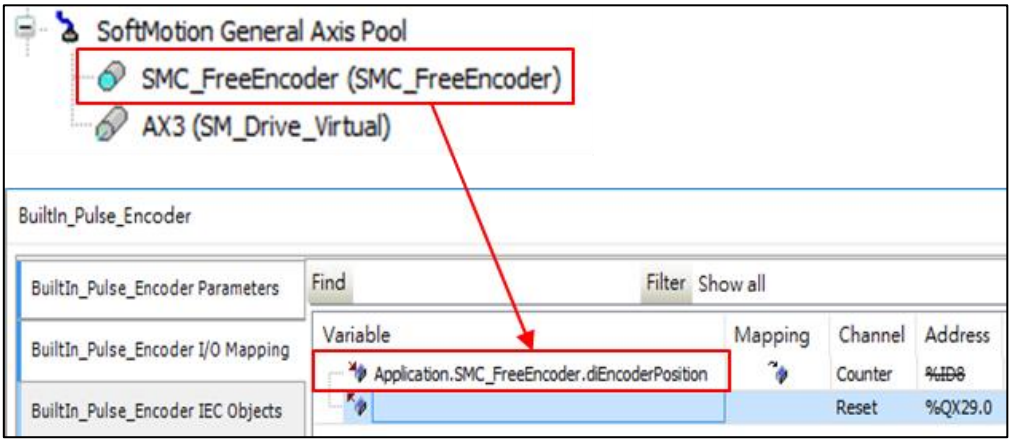


Figure 4 - 251: Pulse encoder mapped with counter

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

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

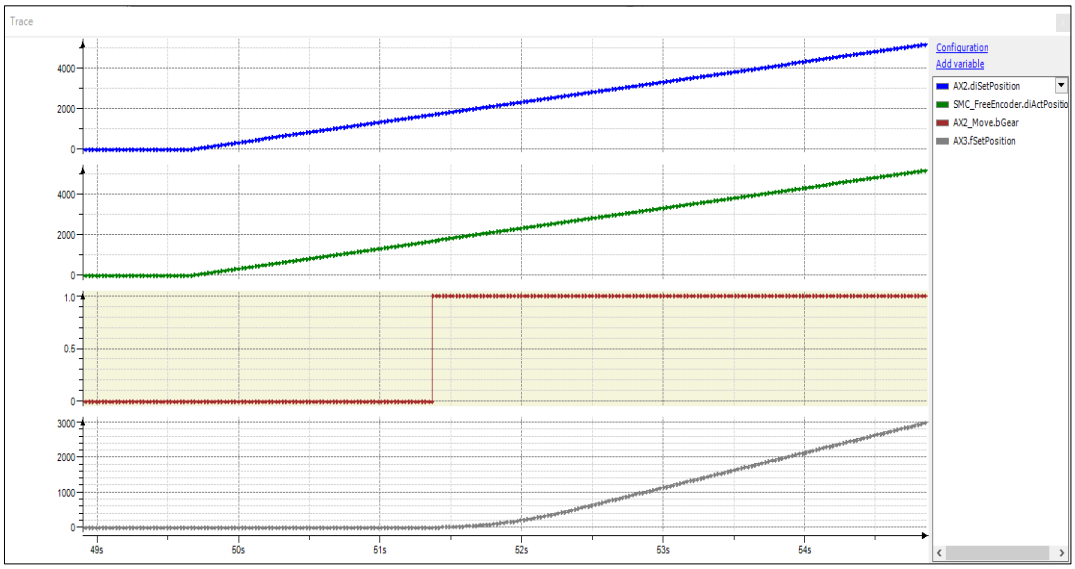


Figure 4 - 252: SMC_FreeEncoder as master axis and do GearIn with virtual axis

4.4 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.

4.4.1 Synchronous Axis

Support slave device:

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

Please follow the steps below to add a synchronous axis: Right-click EtherCAT_Master_SoftMotion in the device tree> Add Device> Search ASDA-A2> ASDA-A2-E_SM.

After selecting the SM 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), as shown in the figure below. The servo axis name generated below is SM_Drive_ETC_Delta_ASDA_A2.

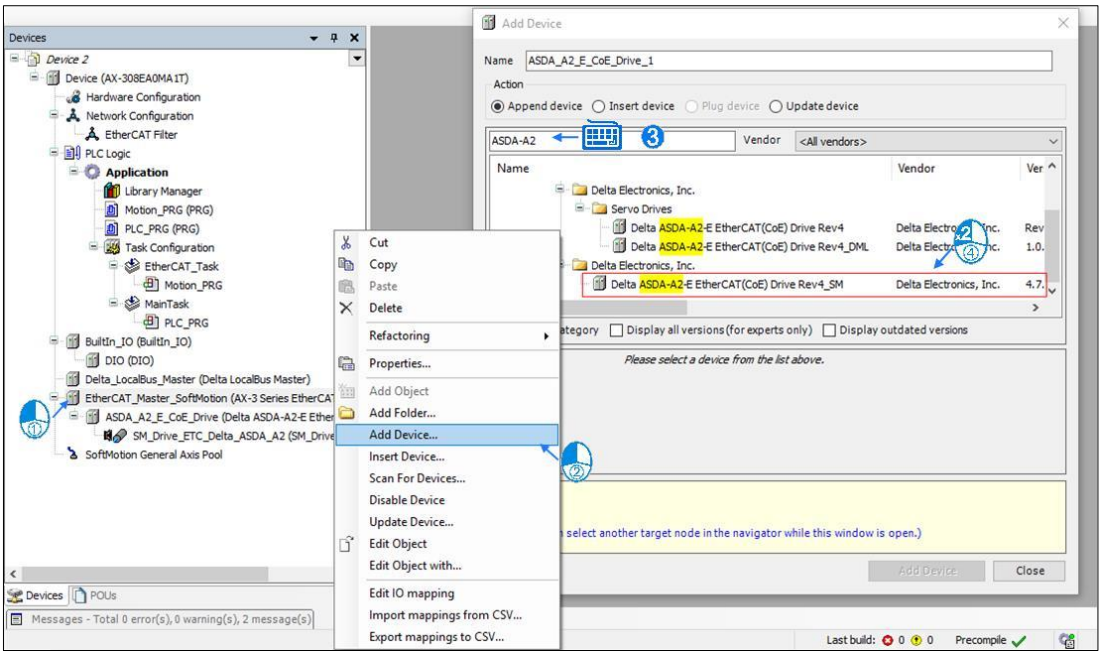


Figure 4 - 253: Added ASDA-A2-E synchronous axis

Perform basic positioning control such as absolute positioning, relative positioning... and other functions through EtherCAT communication. Synchronous axis needs to match DL_MotionControlLight library.

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.

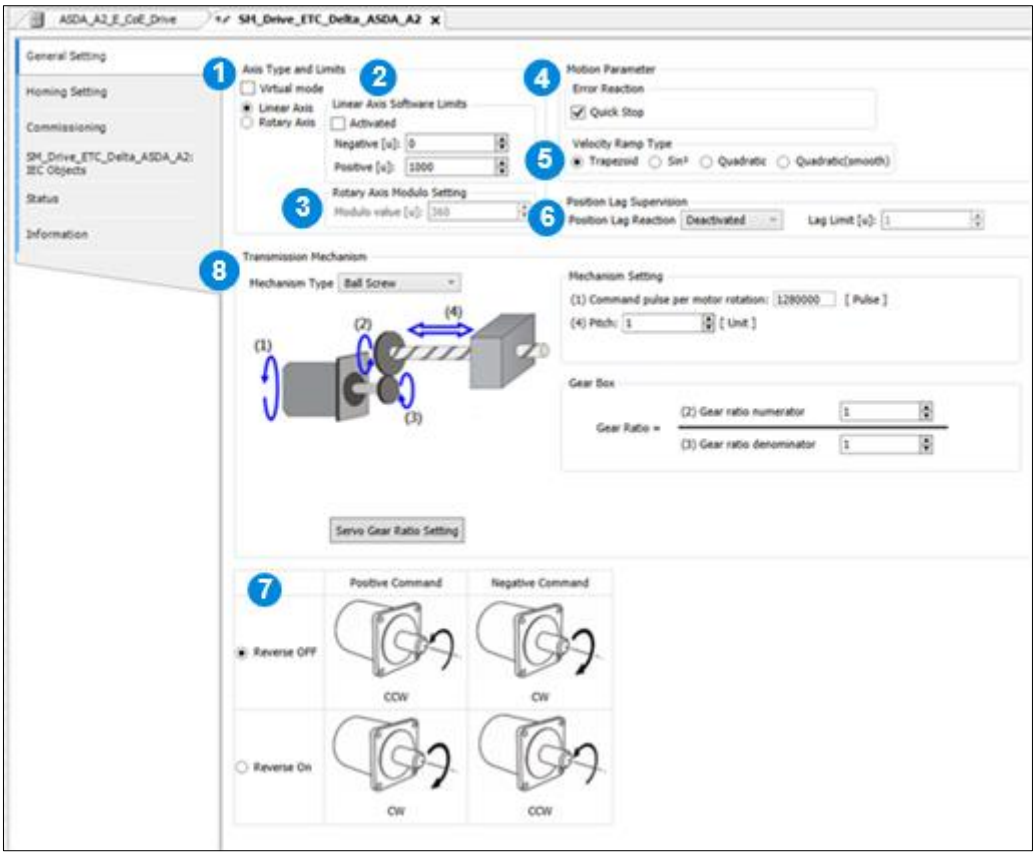


Figure 4 - 254: General Settings Tab

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**

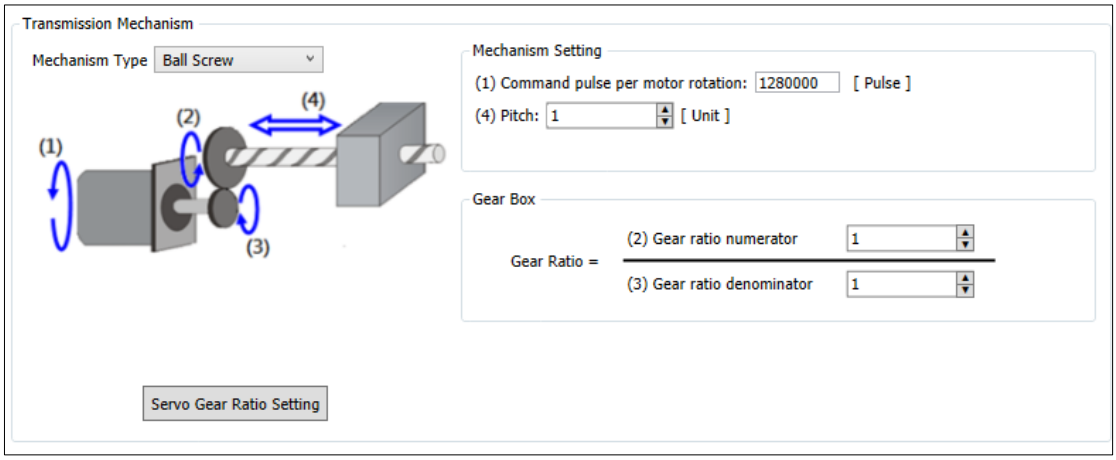


Figure 4 - 255: Ball Screw Drive

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**

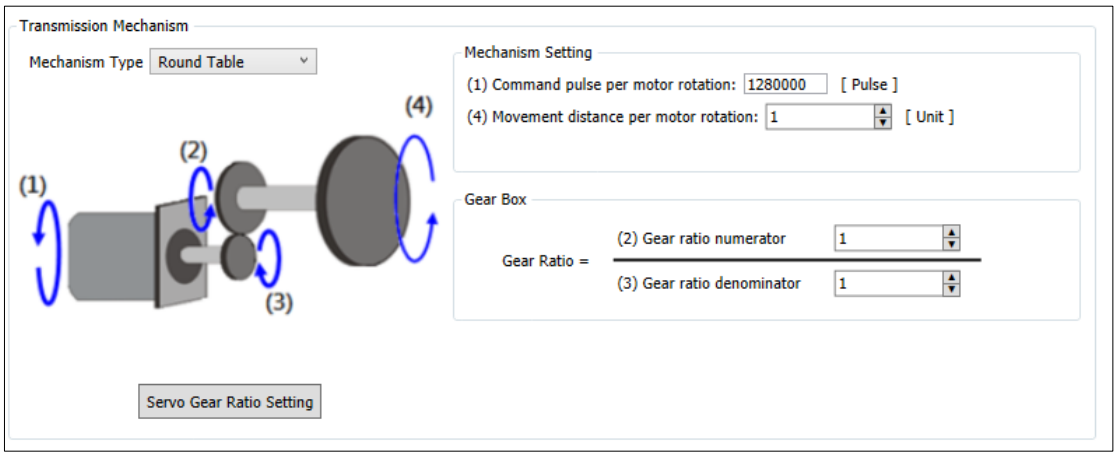


Figure 4 - 256: Round table transmission

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**

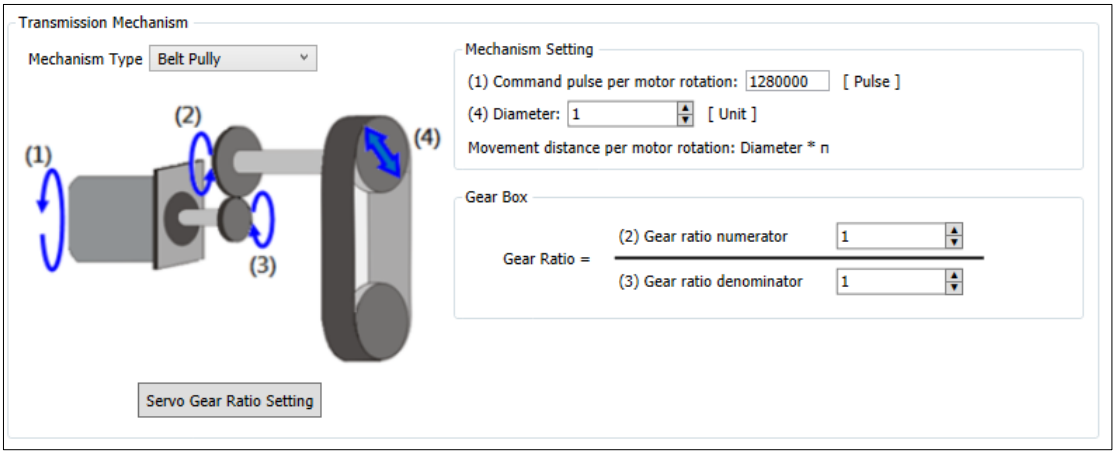


Figure 4 - 257: Belt drive

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)

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

Use 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 figure below. The name of the servo axis generated by DIADesigner-AX is *SM_Drive_ETC_Delta_ASDA_A2*.

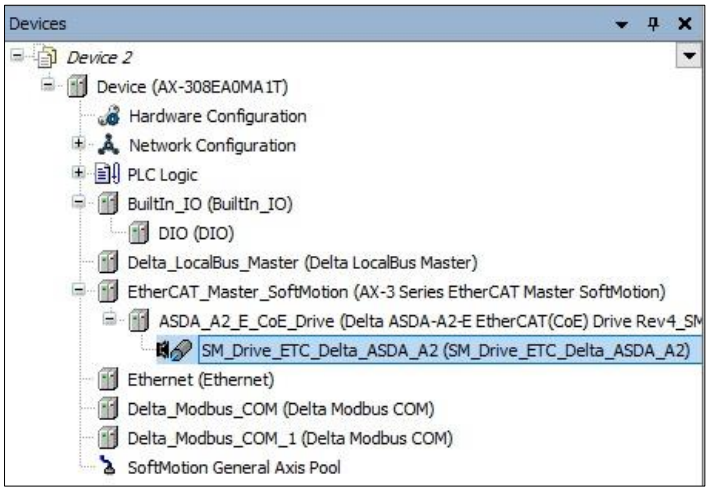


Figure 4 - 258: Add servo axis

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

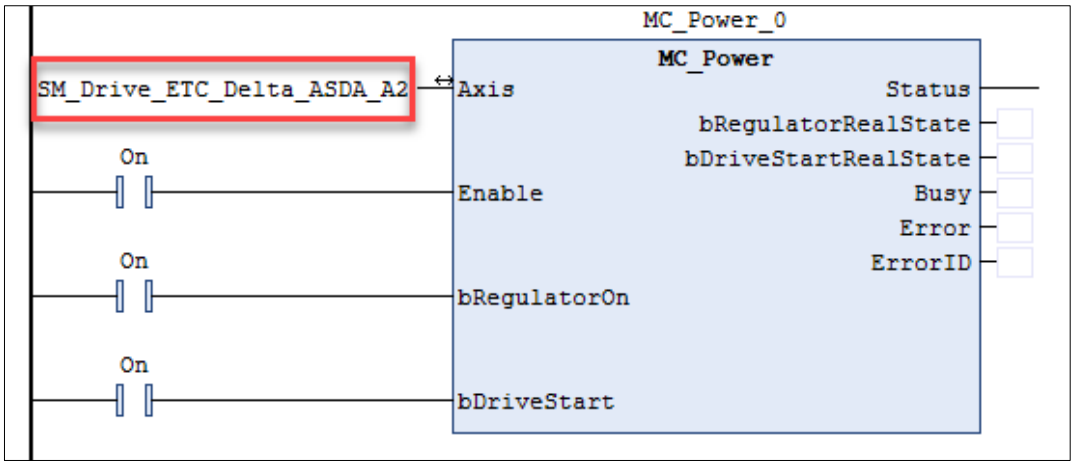


Figure 4 - 259: Axis pin SM_Drive_ETC_Delta_ASDA_A2

4.4.2 Positioning Axis

This section applies to the following devices:

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

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	

Category	Type	Function group	Description
Multi-axis motion control finger	Motion control type	Axis group movement	Multi-axis motion
	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.*

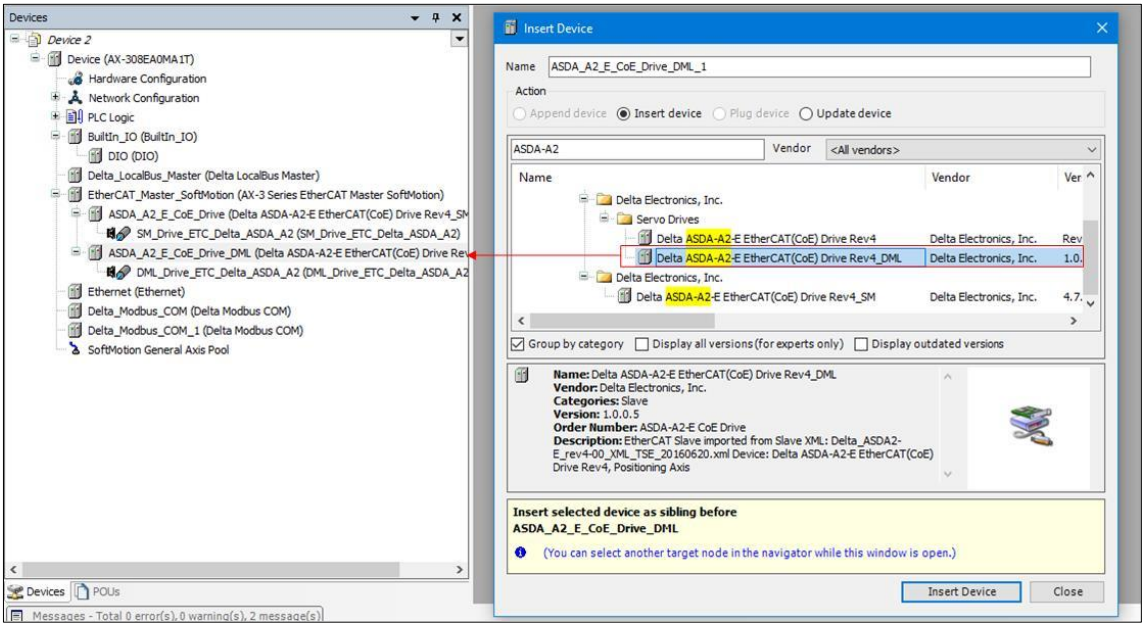


Figure 4 - 260: Added ASDA-A2-E positioning axis

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

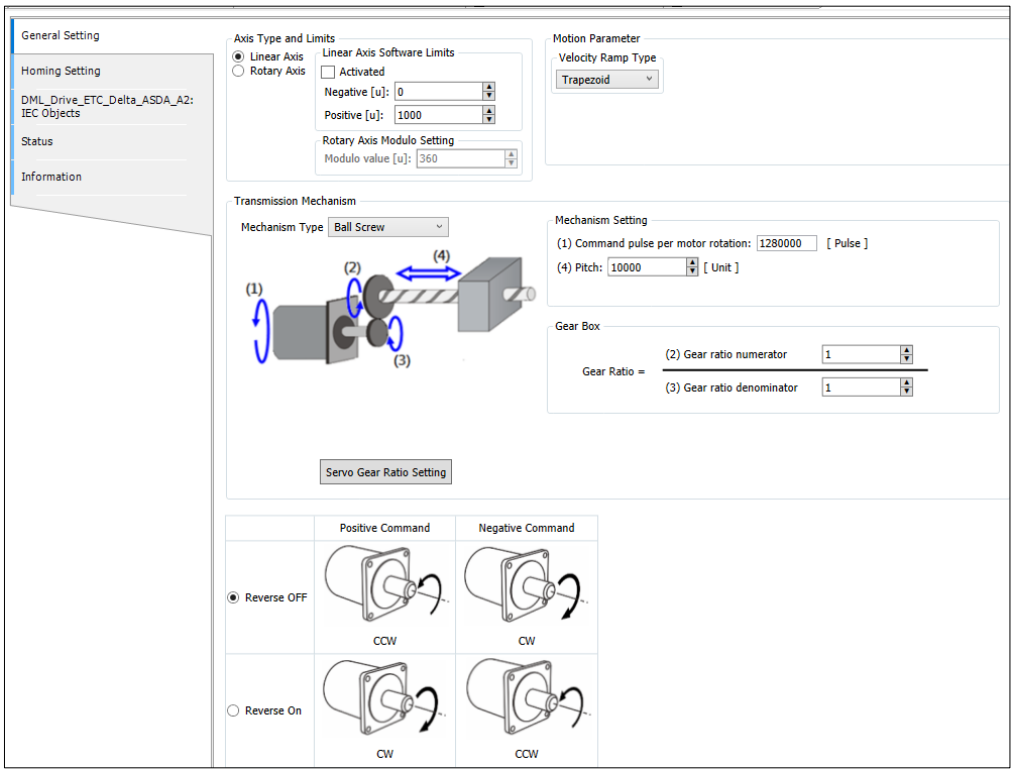


Figure 4 - 261: Positioning Axis General Setting Page

- **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**

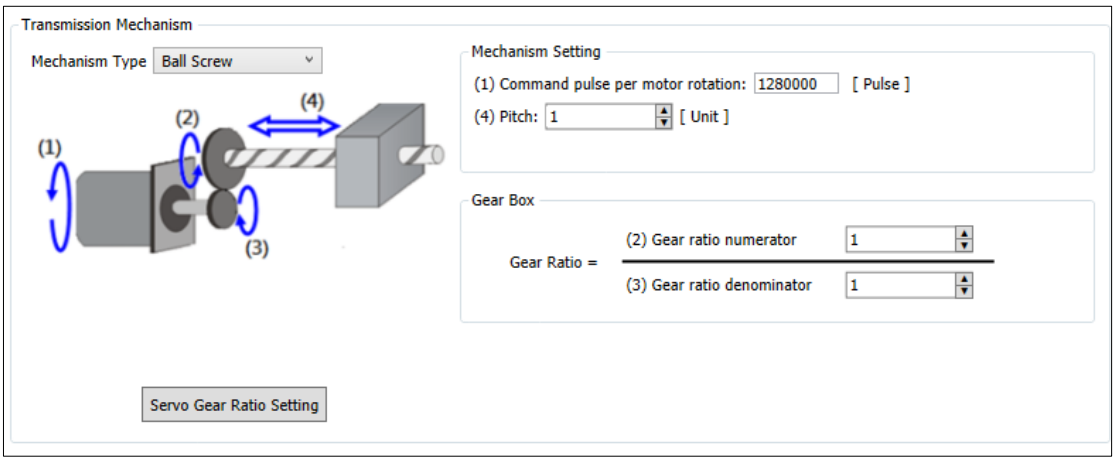


Figure 4 - 262: Ball screw drive

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**

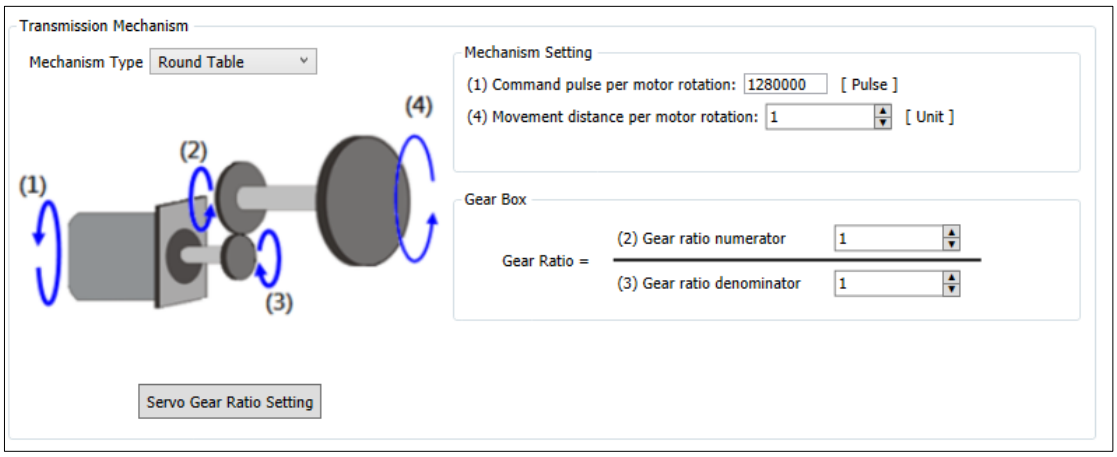


Figure 4 - 263: Round table transmission

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**

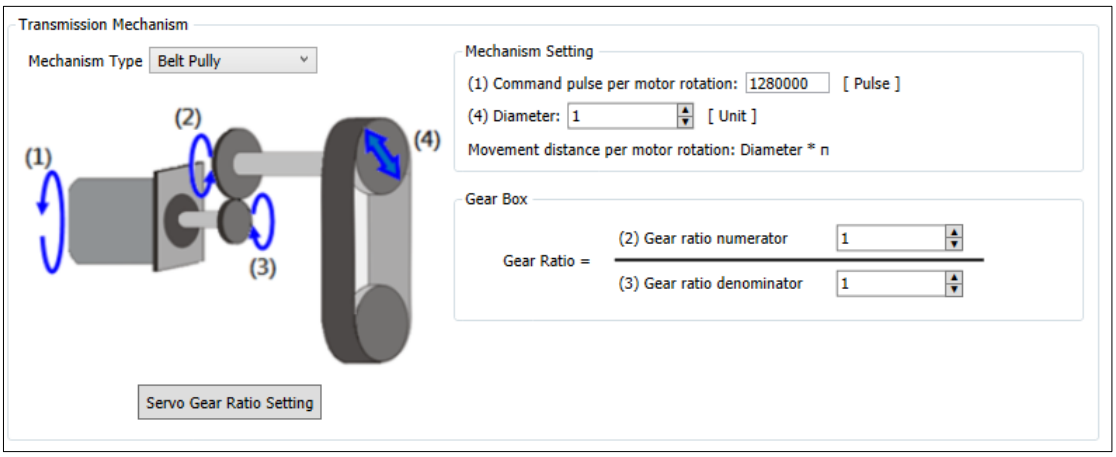


Figure 4 - 264: Belt drive

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
(4) Diameter (Movement distance per motor rotation: Diameter *n)	The Movement distance per motor rotation: Diameter *n

- **Servo Gear Ratio**

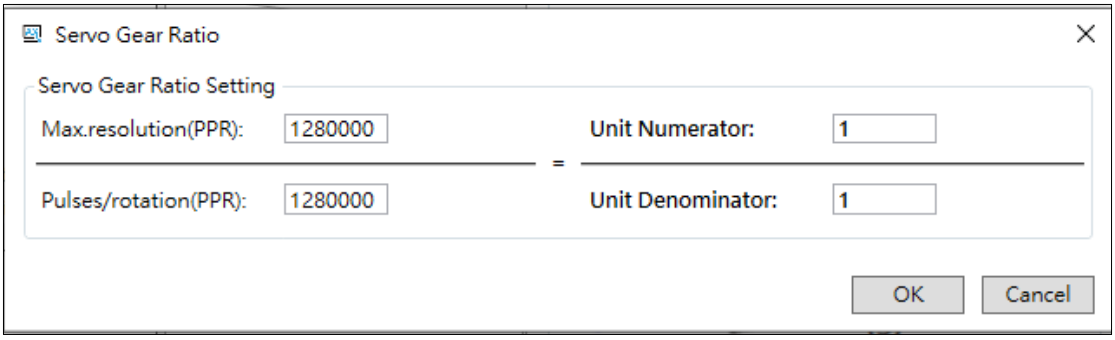


Figure 4 - 265: Servo gear ratio

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.

Return to origin setting page

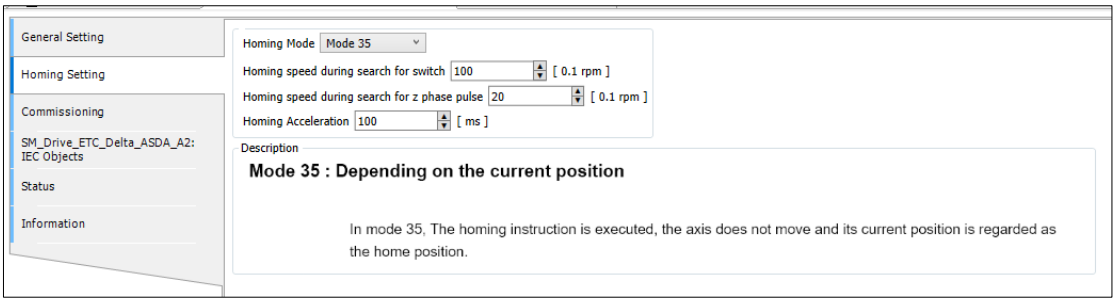


Figure 4 - 266: Origin return setting

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, you 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.

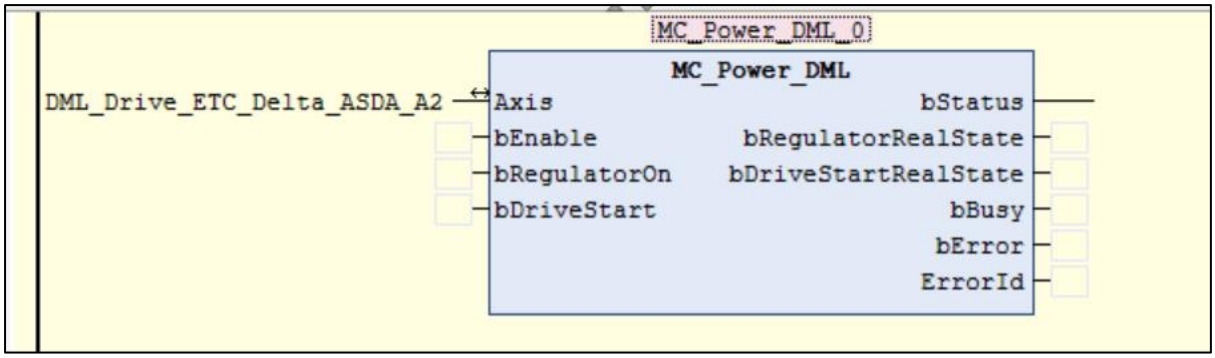


Figure 4 - 267: Shaft pin DML_Drive_ETC_Delta_ASDA_A2

4.4.3 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*.

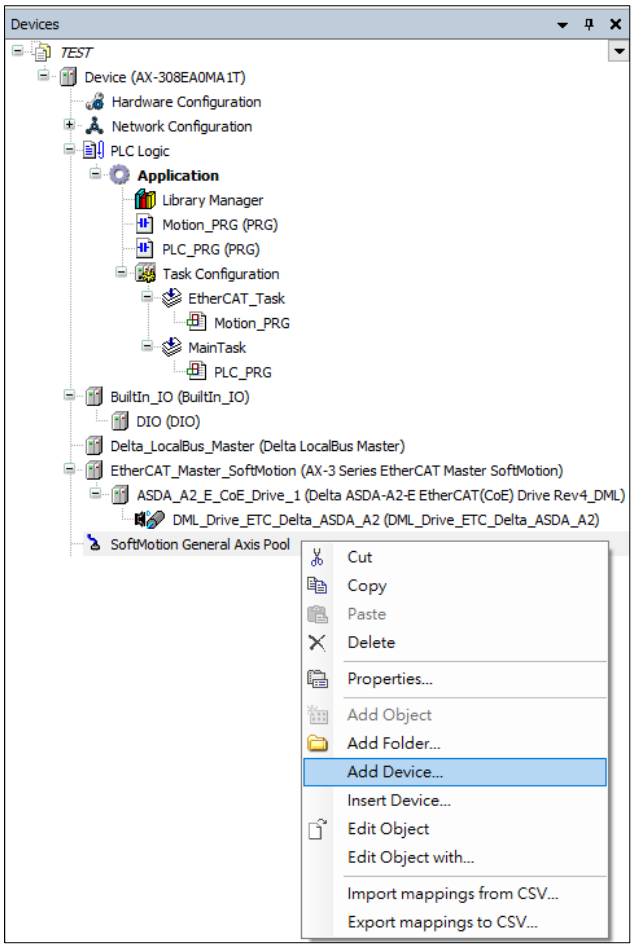


Figure 4 - 268: Add a virtual drive (1)

Virtual Axis Page

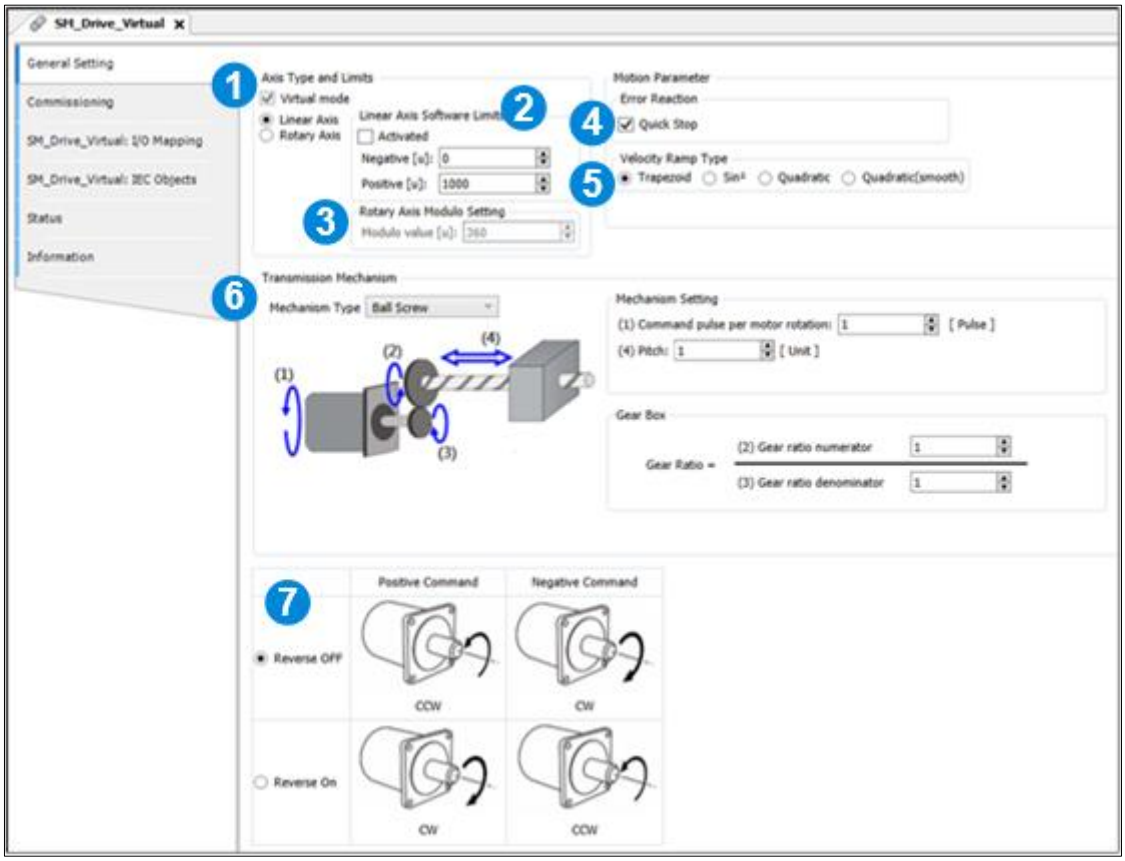


Figure 4 - 269: Virtual axis

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)

Name	Description
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

3. **Velocity Ramp Type**

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

4. **Transmission mechanism**

Ball screw drive

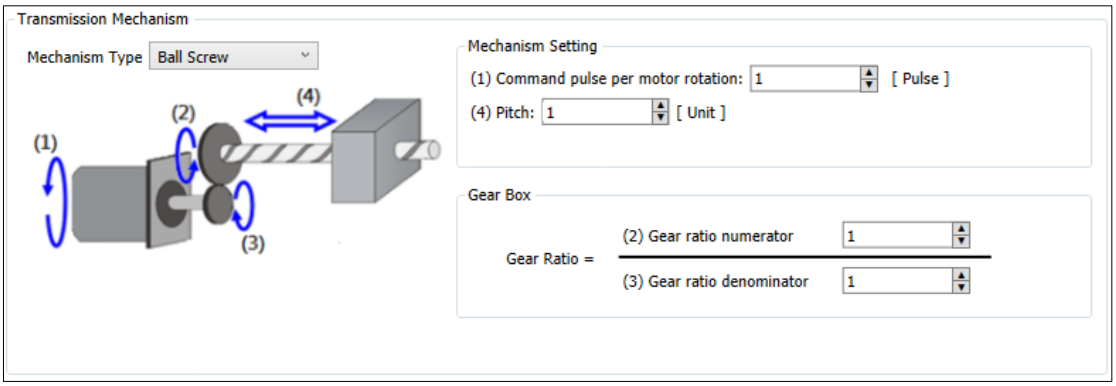


Figure 4 - 270: Ball screw drive

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

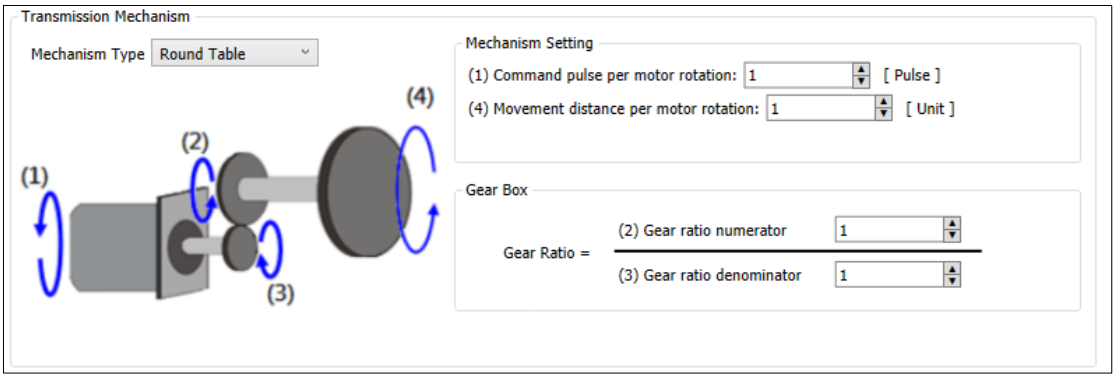


Figure 4 - 271: Round table transmission

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	Movement distance per motor rotation = diameter*n

Belt drive

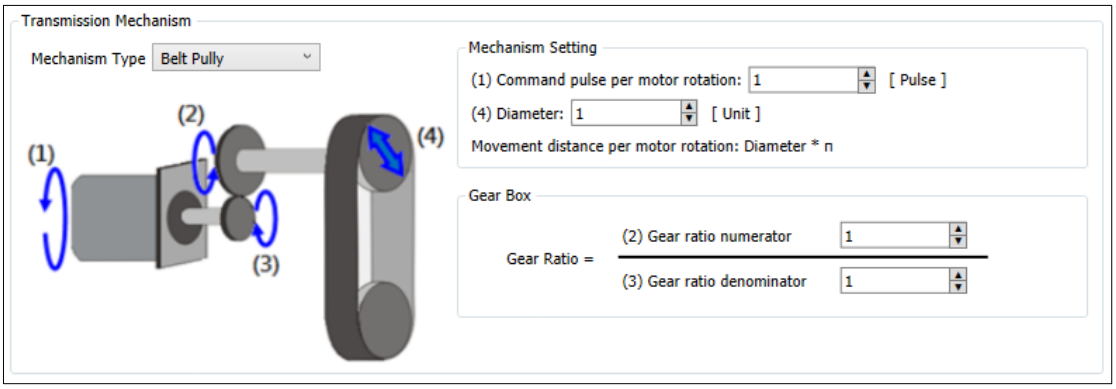


Figure 4 - 272: Belt Drive

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

5. **Positive/Negative Commands**

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

4.4.4 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.*

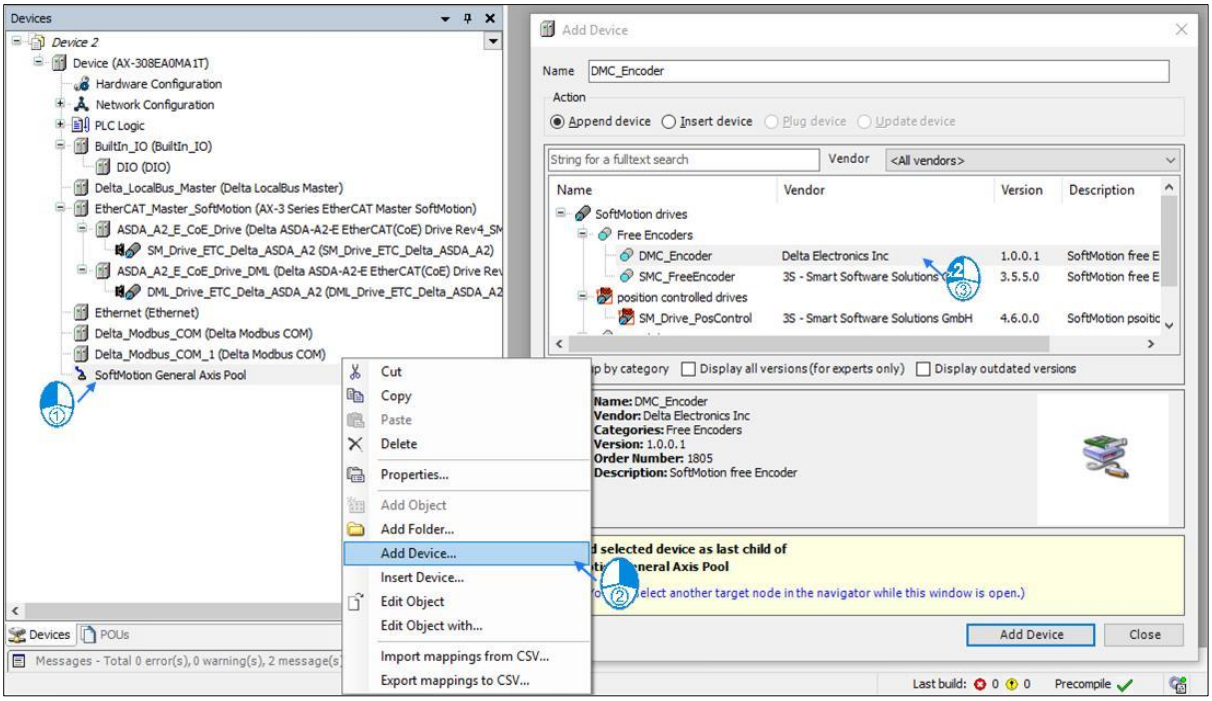


Figure 4 - 273: Add DMC_Encoder

The following figure shows the default display screen after double-clicking DMC_Encoder in the project tree:

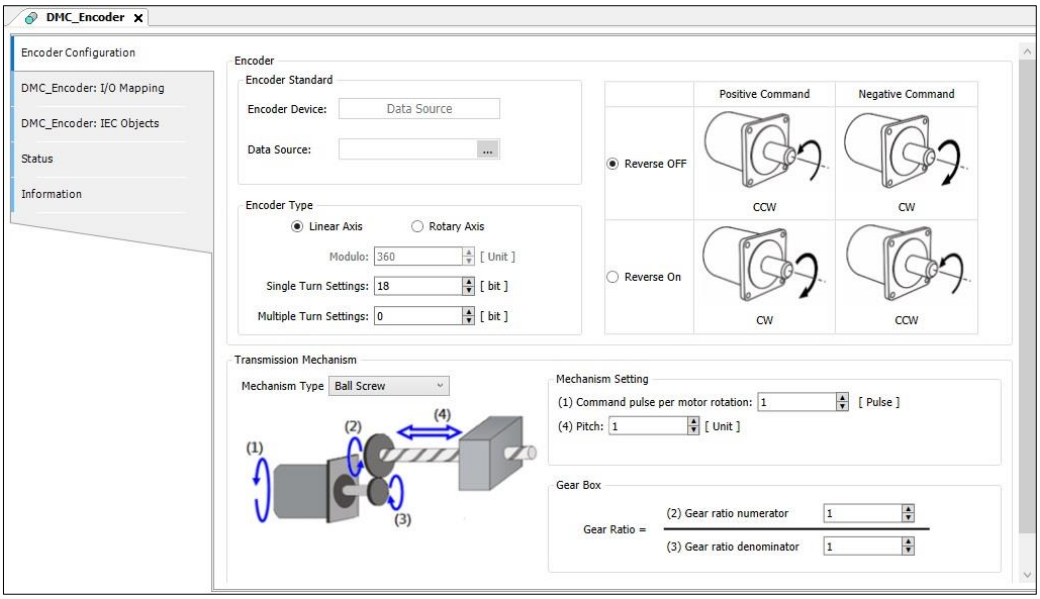


Figure 4 - 274: DMC_Encoder preset screen

Parameters Tab:

To add the DMC_Encoder parameter setting tab, please go to the main menu *Tools*> *Options*> *Device Editor*> tick *Display Generic Device Configuration Views*.

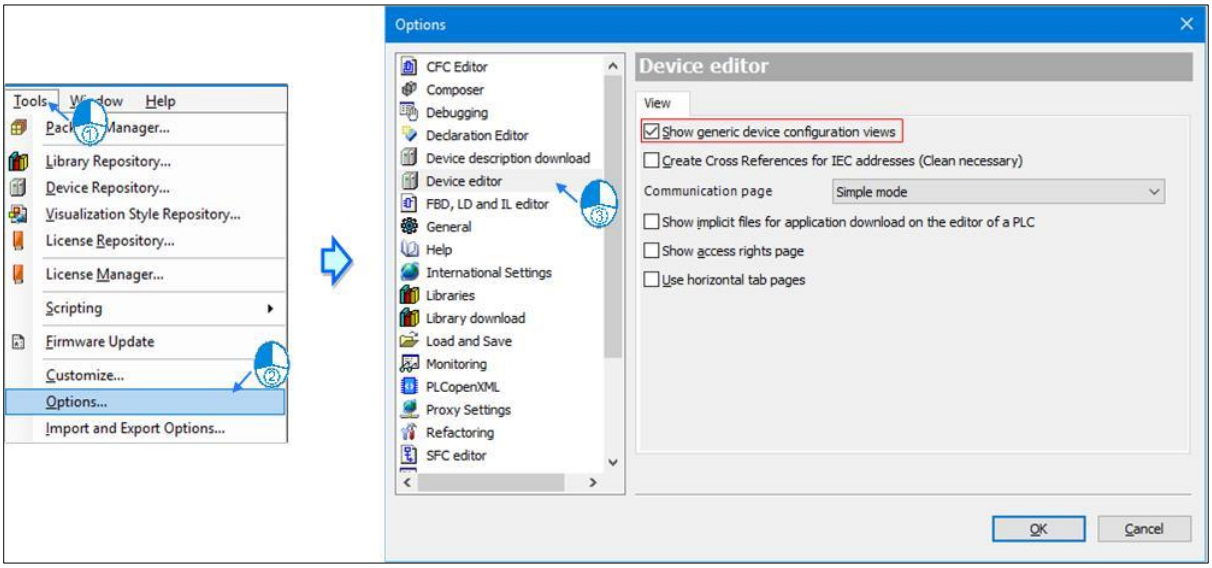


Figure 4 - 275: Add DMC_Encoder parameter setting screen steps

The user can set relevant parameters in the DMC_Encoder parameter tab.

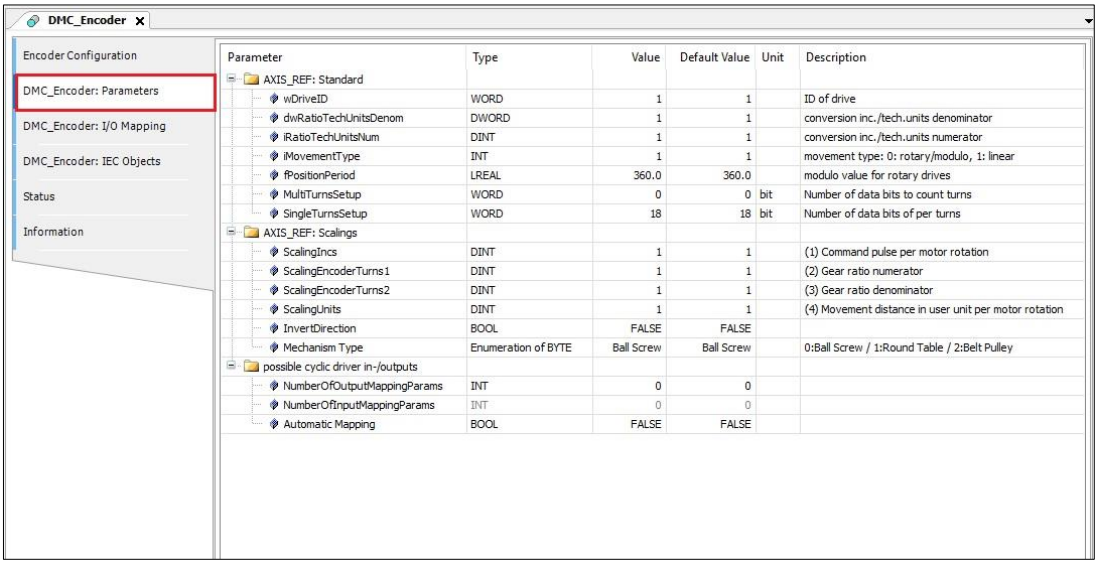


Figure 4 - 276: DMC_Encoder parameter tab

Encoder configuration tab

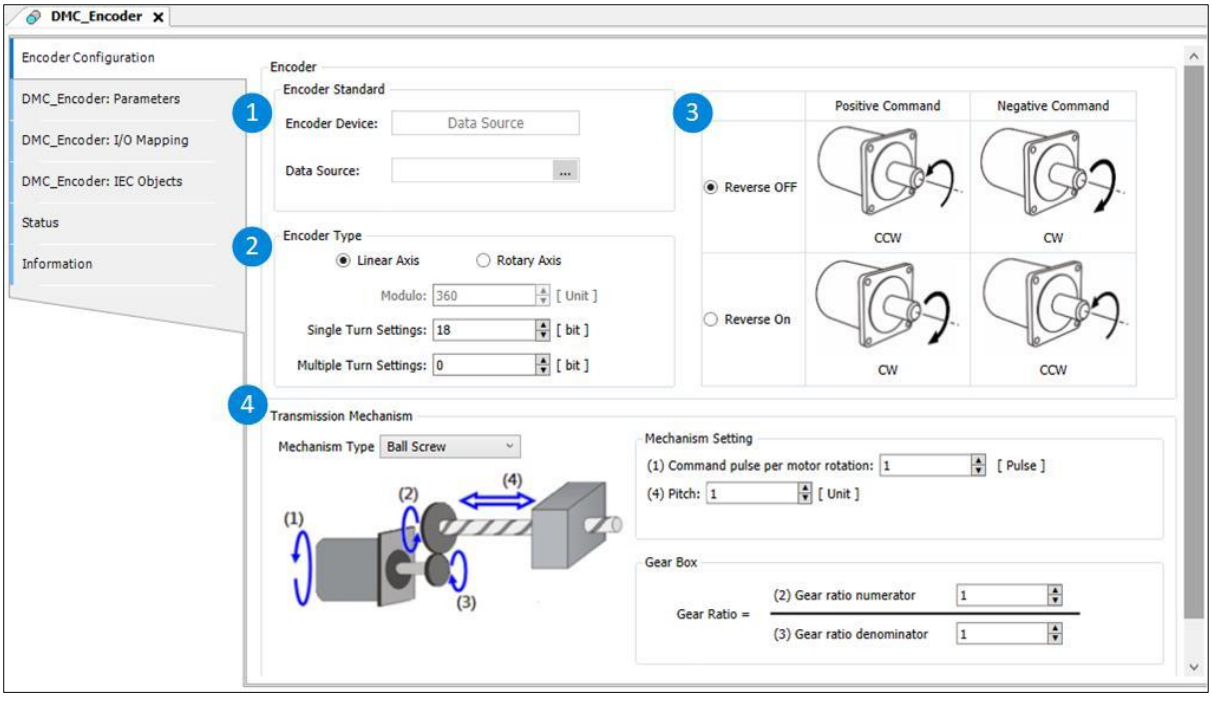


Figure 4 - 277: Virtual encoder configuration screen

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

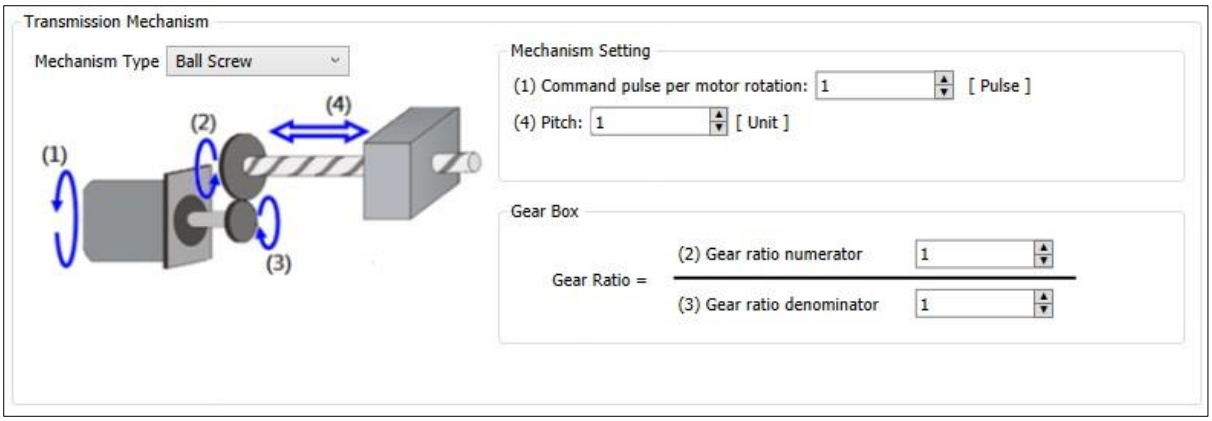


Figure 4 - 278: Ball screw drive

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

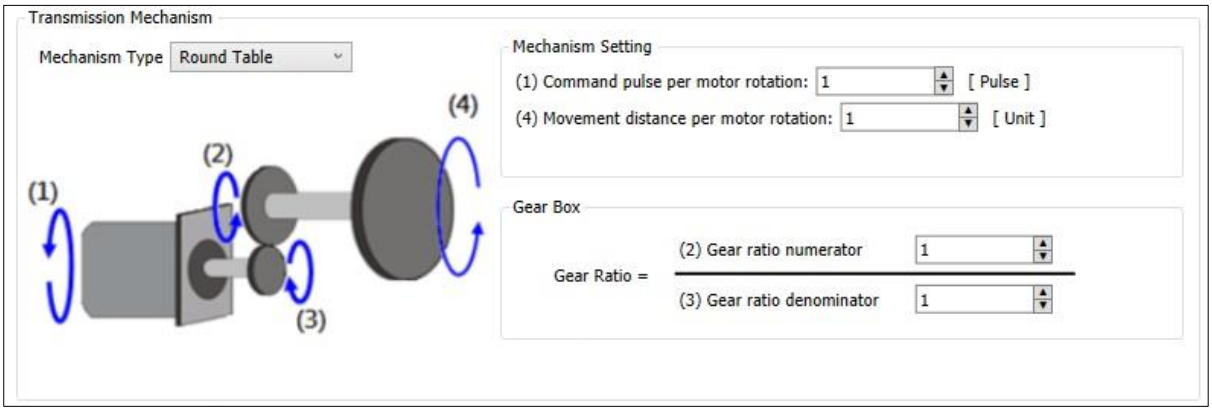


Figure 4 - 279: Round table transmission

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

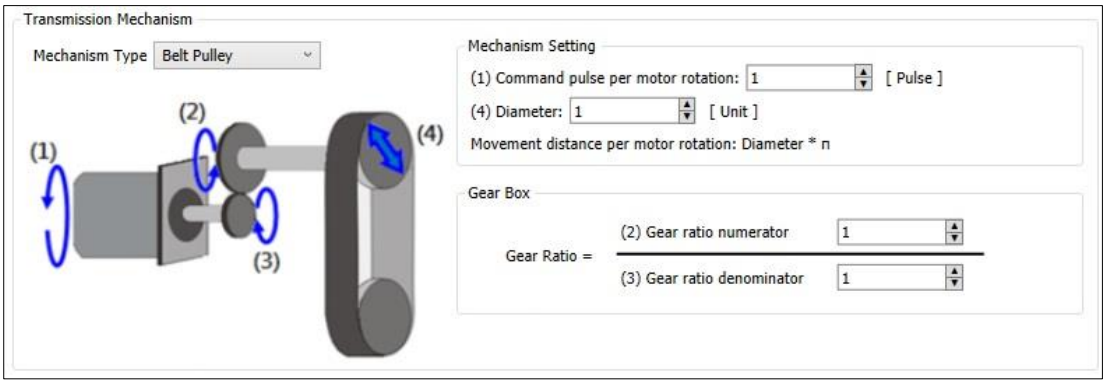


Figure 4 - 280: Belt Drive

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 figure below.

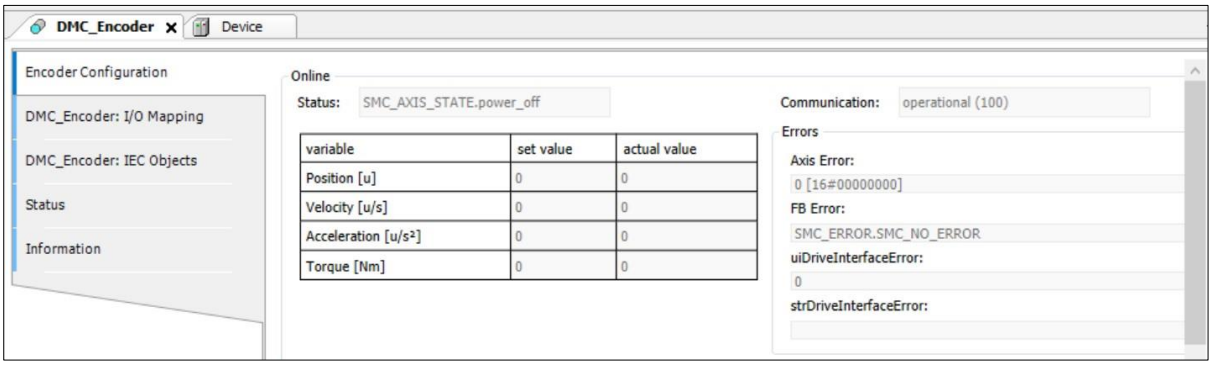


Figure 4 - 281: Online Monitoring Information

4.4.5 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 equipment: AX-308E, AX-364E.

4.4.5.1 Axis Group Configuration Process And Setting Steps

The parameters used for axis group motion are as follows:

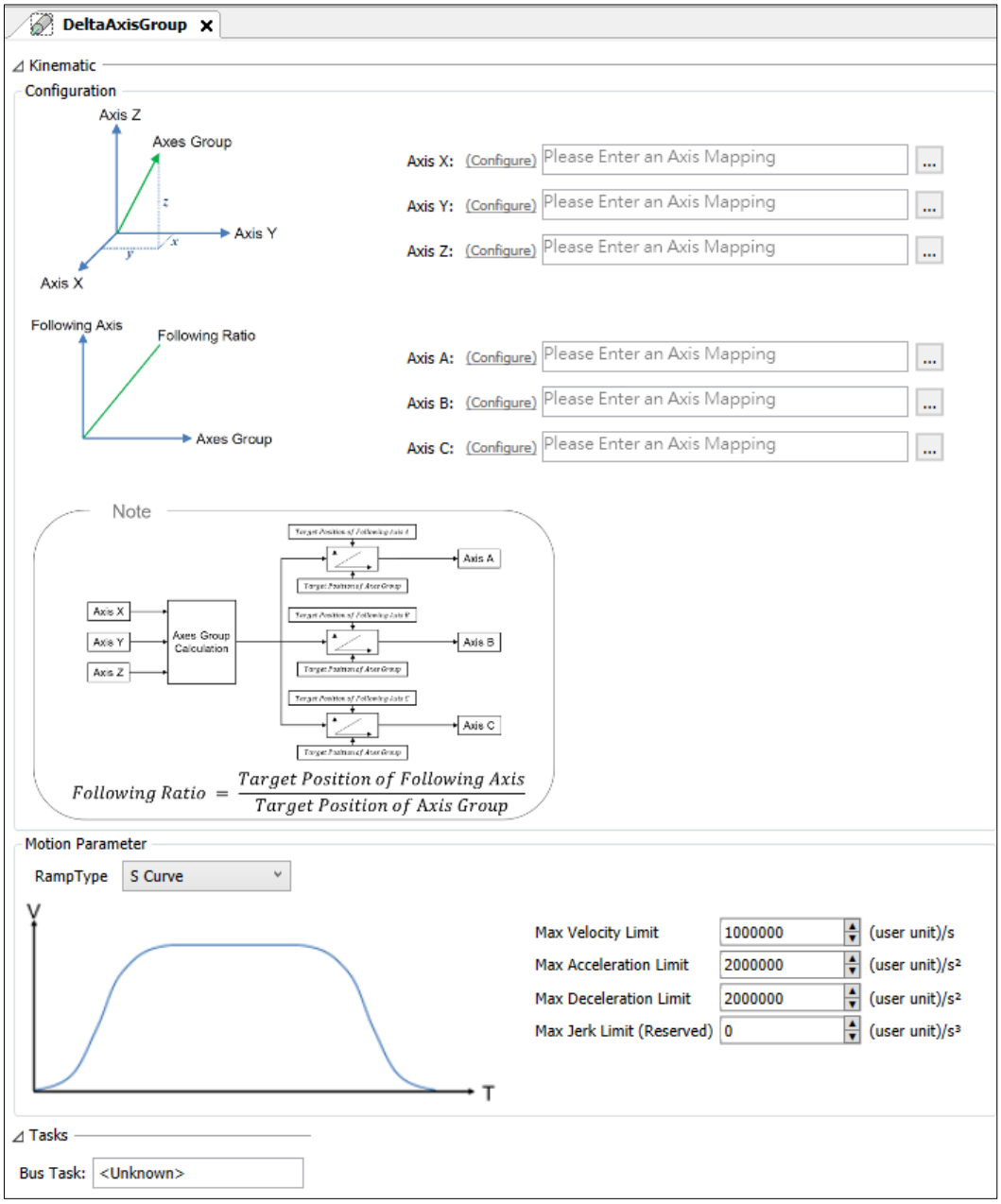


Figure 4 - 282: Axis group parameters

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

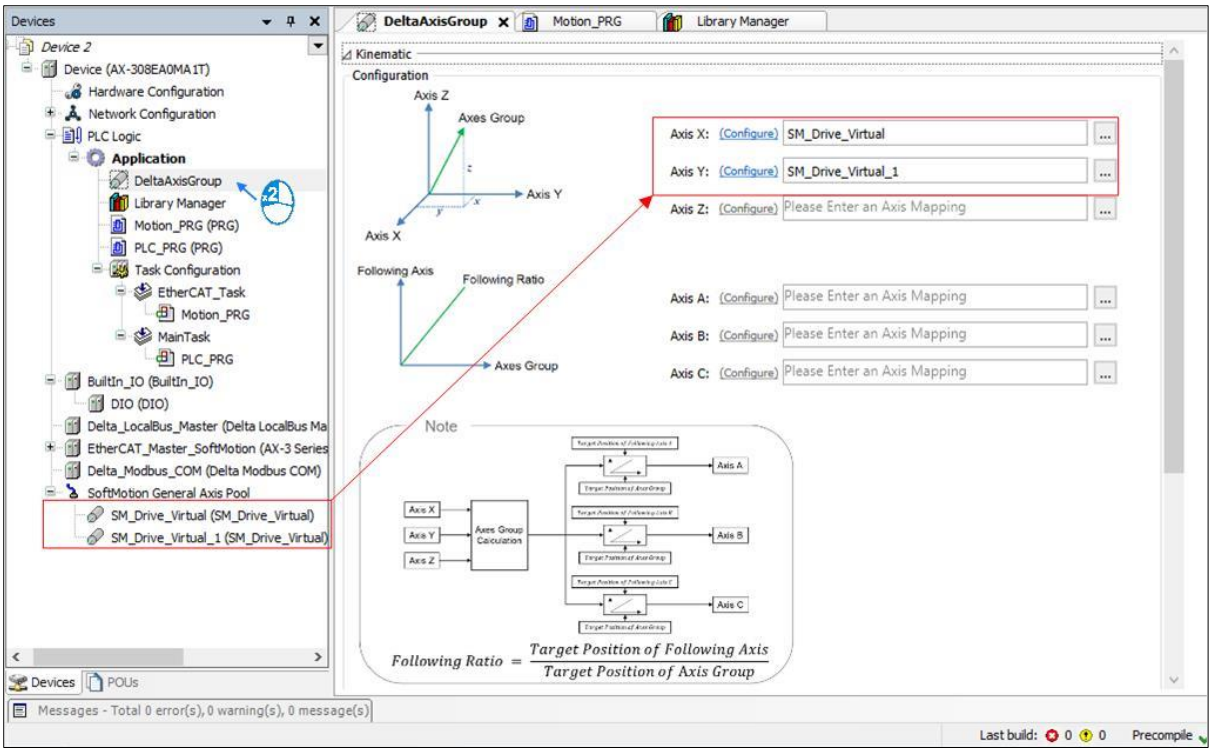


Figure 4 - 283: Kinematic model

2. *Slope type: There are two types of curves, trapezoid and S-curve, and the curves are shown as the following two types.*

Trapezoid

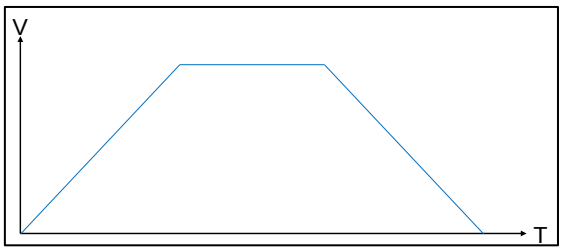


Figure 4 - 284: Trapezoidal curve

S curve

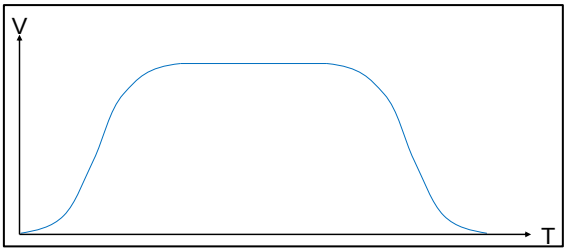


Figure 4 - 285: S curve

3.

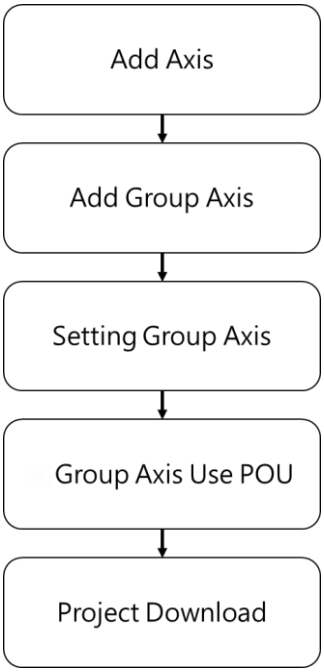
- *Maximum Velocity Limit: an error will be reported when the axis group movement velocity exceeds the set parameter.*

- *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.*

4.4.4.2 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 a axis group:

1. Add a single axis, this process takes the creation of 2 virtual axes as an example.

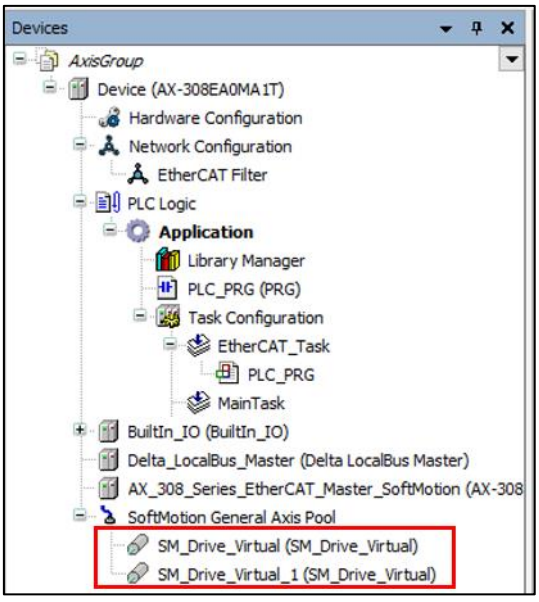


Figure 4 - 286: **New Single Axis**

2. After *the creation is complete,* right-click on the *Application* and select *Add*

Object > *Delta Axis Group* *as shown in the following figure*.

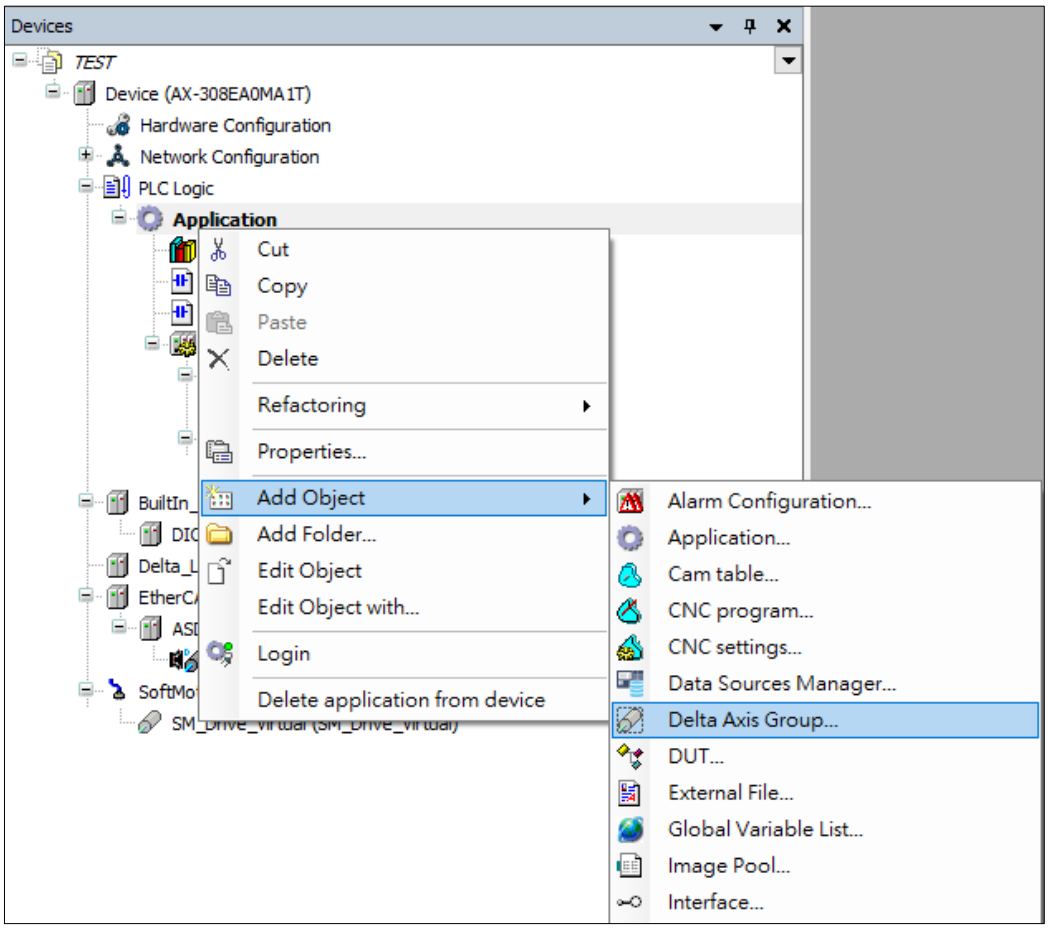


Figure 4 - 287: Delta Shaft group

Result: Opens *Add Delta Axis Group* window.

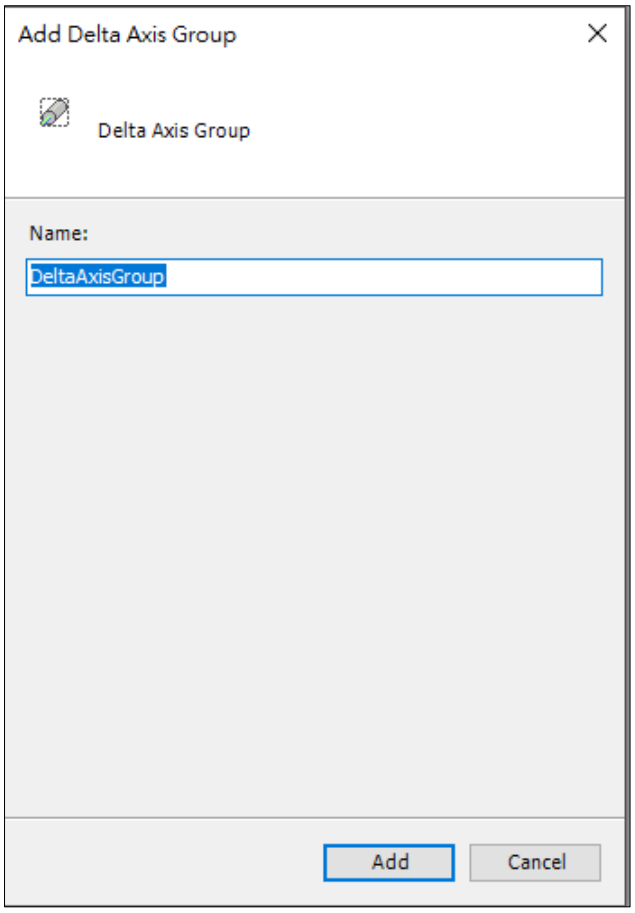


Figure 4 - 288: Add Delta axis group

3. Add name in the *Name* field and click on *Add* button.

   Result: The *DeltaAxisGroup* appears on the project tree.

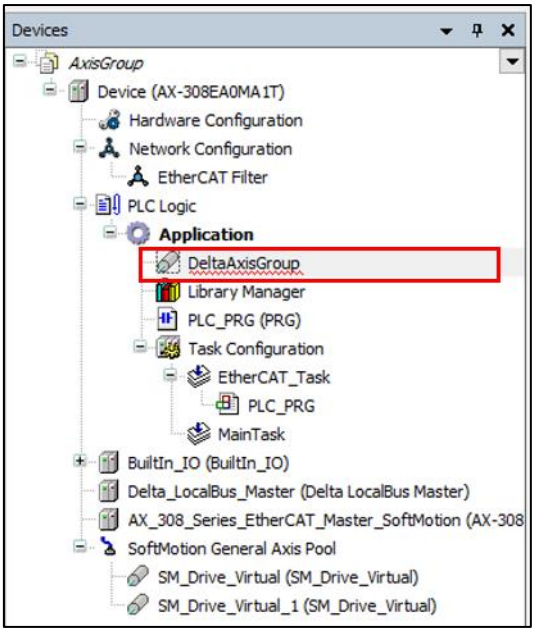


Figure 4 - 289: DeltaAxisGroup

4. Click on *DeltaAxisGroup*, enter the *input assistant* and enter the names of the two-axis virtual axis into *Axis X* and *Axis Y*.

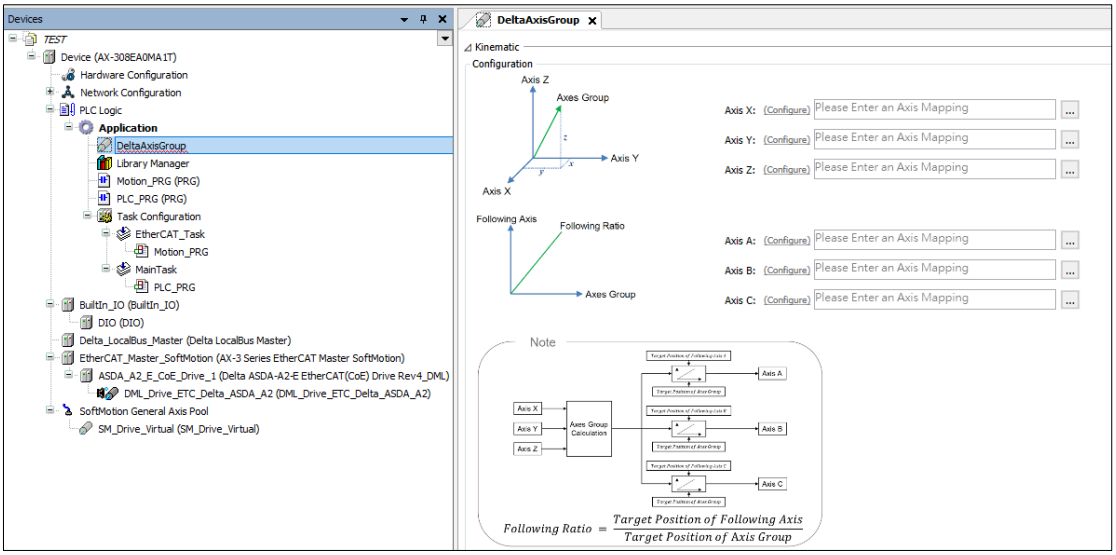


Figure 4 - 290: DeltaAxisGroup Picture

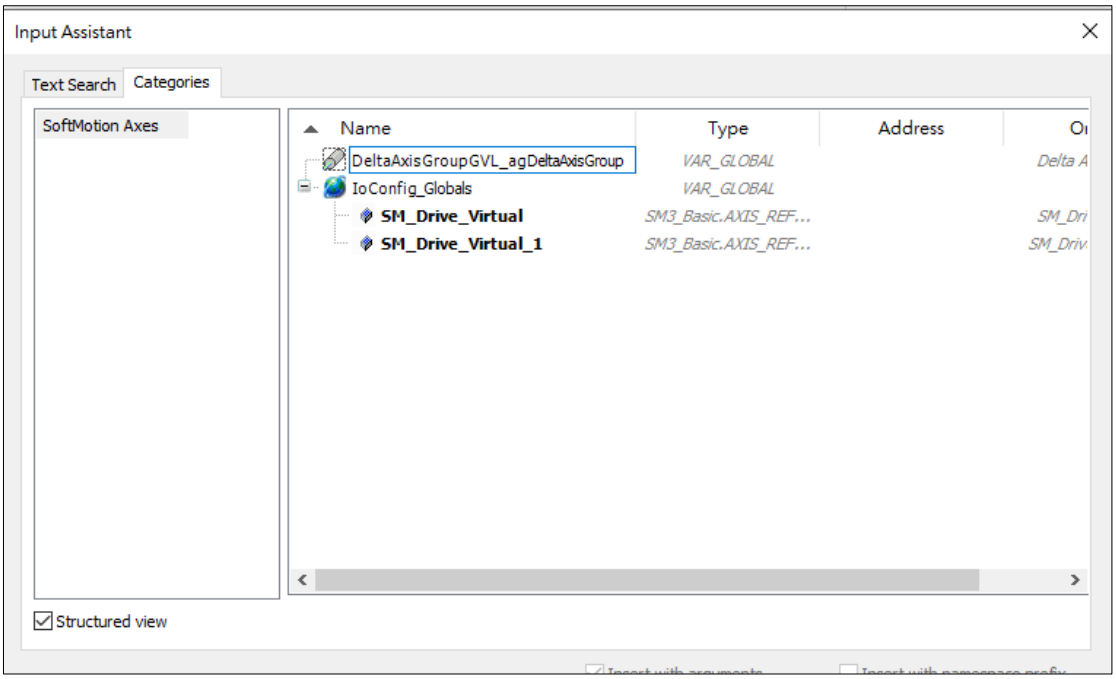


Figure 4 - 291: Input assistant

5. Add the *DMC_GroupEnable* function block in Motion_PRG and fill in the axis group name in the AxisGroup pin.

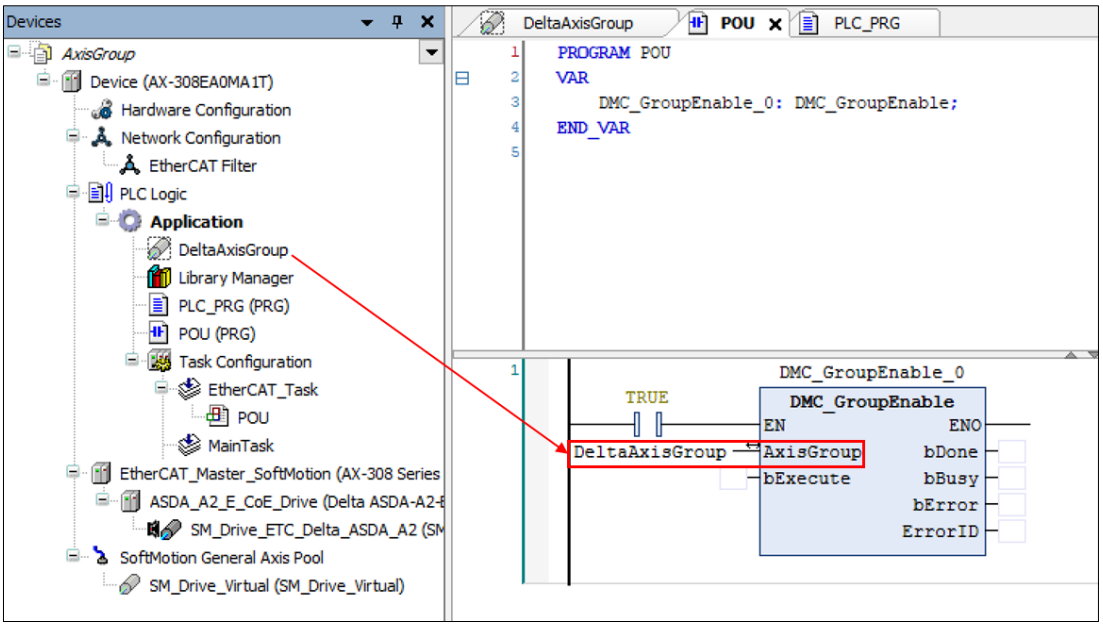


Figure 4 - 292: Added DMC_GroupEnable function block

6. When the program is complete, click compile and confirm whether there are

errors.

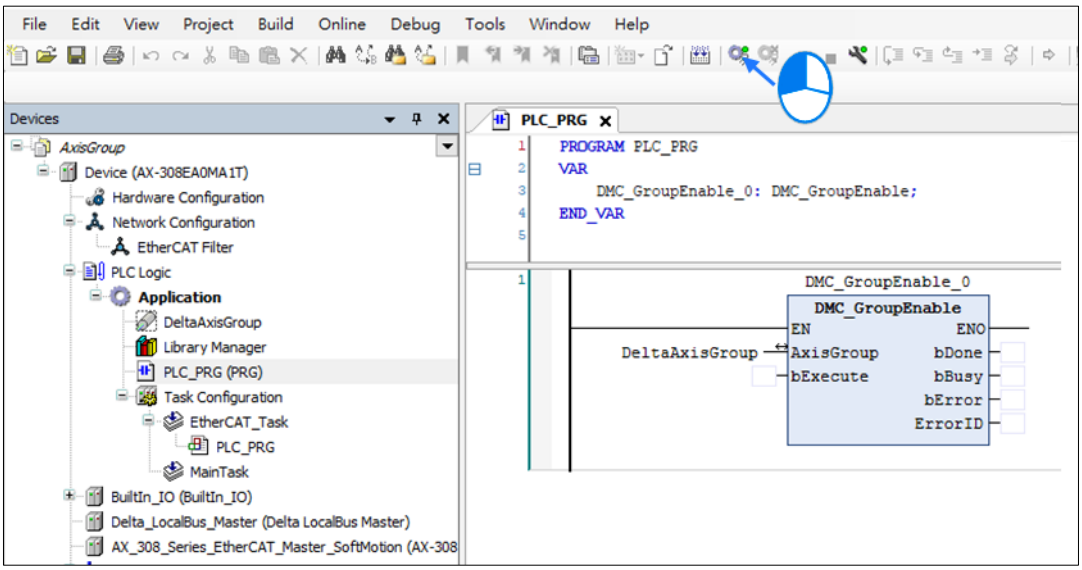


Figure 4 - 293: Compile

7. After confirming that there are no errors after compiling, click Login icon to download the program.

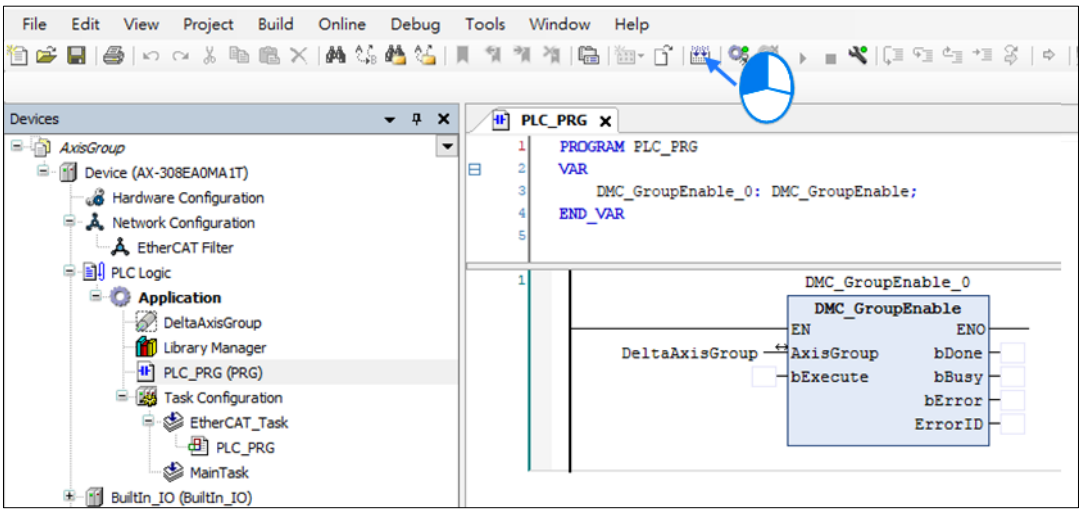


Figure 4 - 294: Compile

8. After confirming that there are no errors after compiling, click Login to download the program.

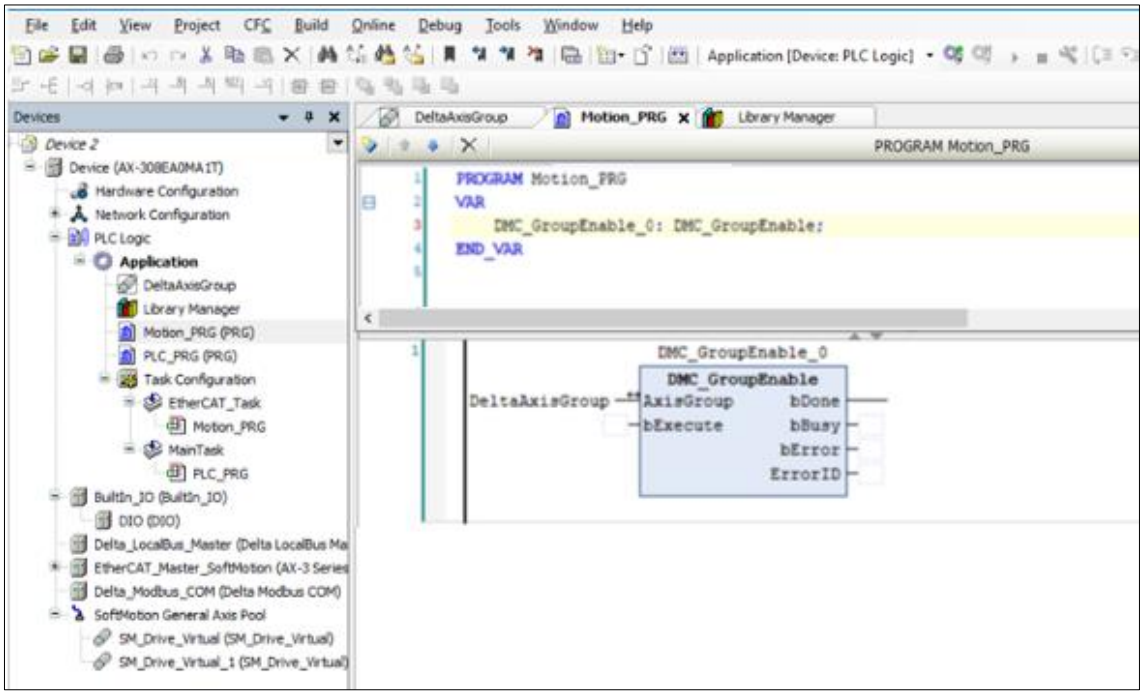


Figure 4 - 295: Log In

Chapter 5: Network Configuration

5.1 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.

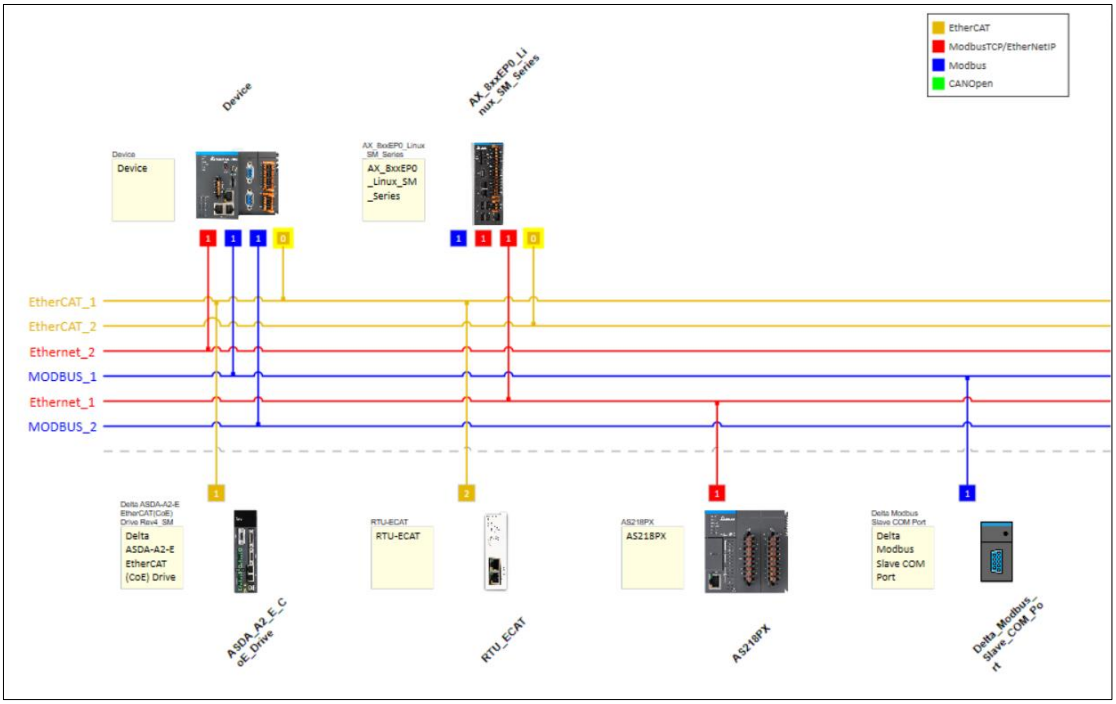


Figure 5 - 1: Network Configuration Editor

Network View Description

Network Configuration window displays two types of information:

- Device information
- Network information

Device information includes the following:

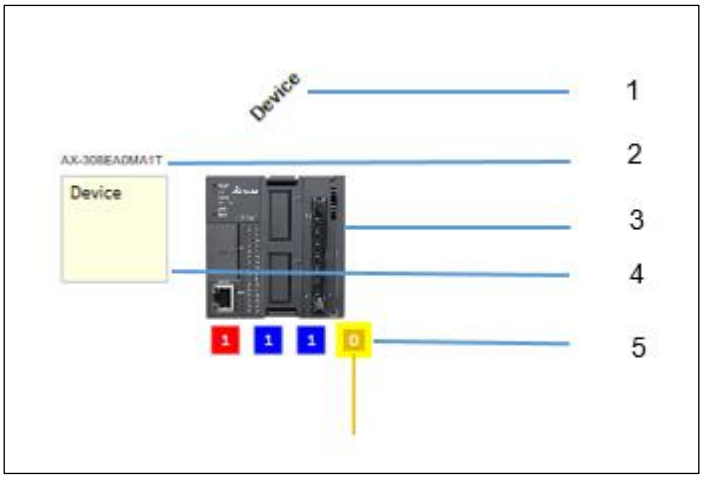


Figure 5 - 2: Device Information

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:

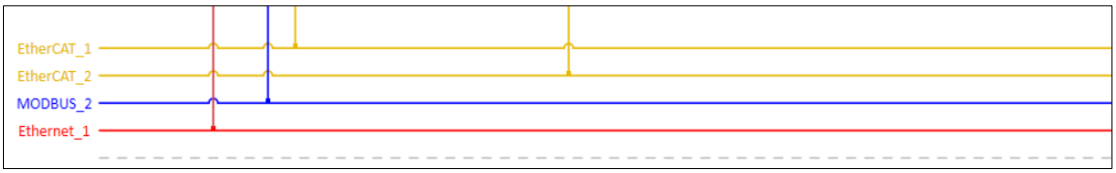


Figure 5 - 3:Network Information

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:

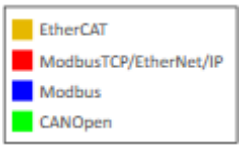


Figure 5 - 4: Color Description

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.

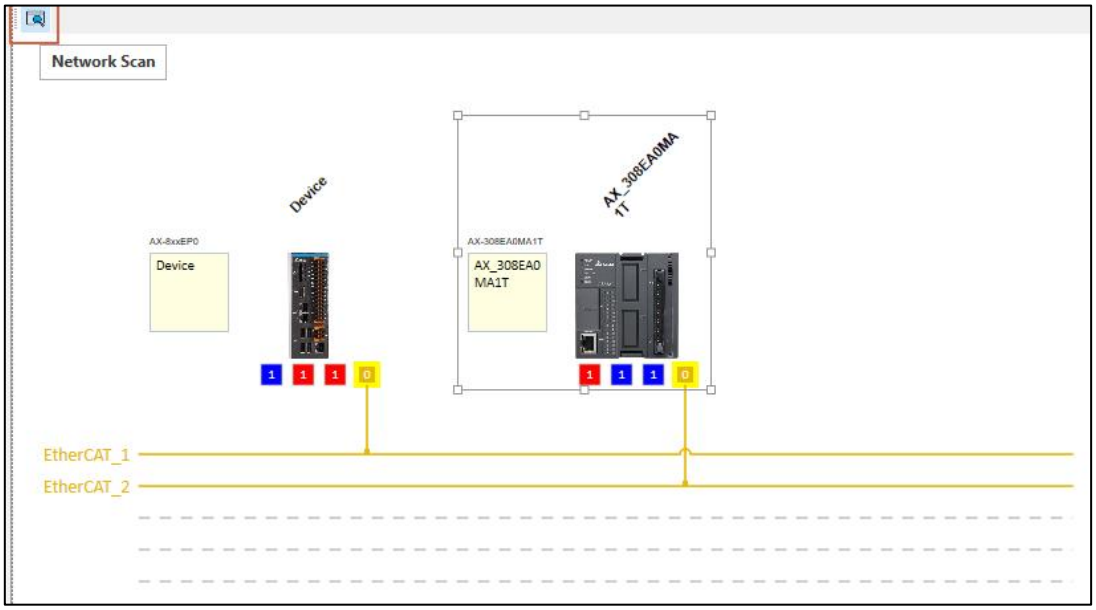


Figure 5 - 5: Network Scan

2. Right-click the EtherCAT node and choose *Scan For Devices* to scan the device under fieldbus.

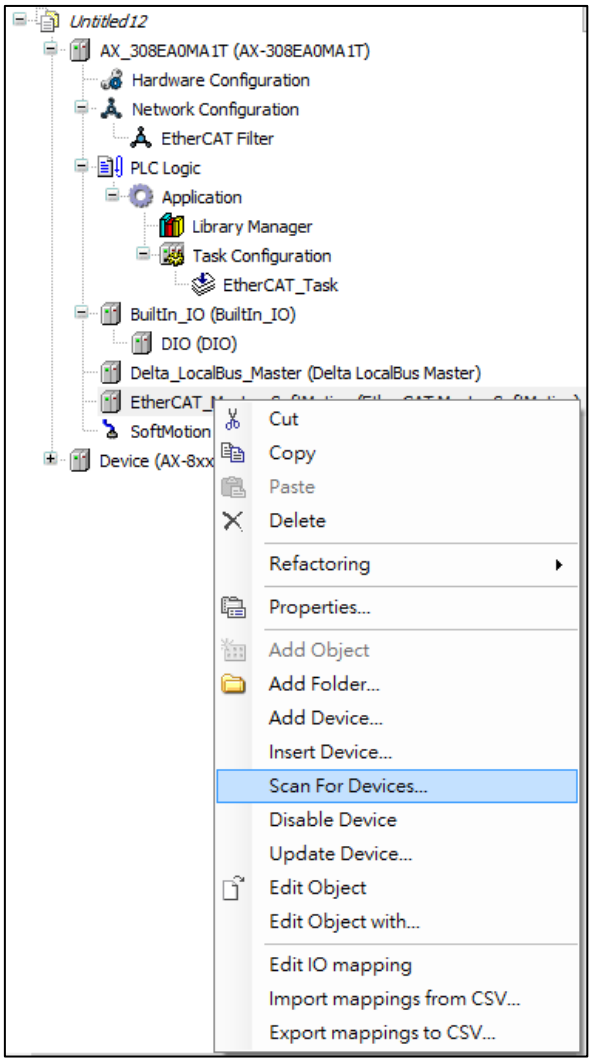


Figure 5 - 6: Scan For Device

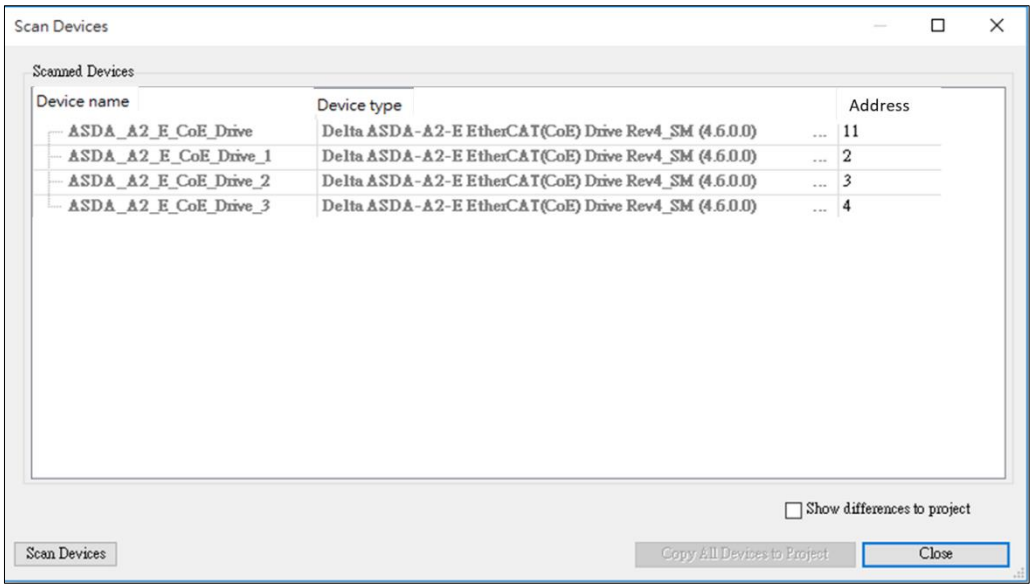


Figure 5 - 7: Network Scan Window

NOTE: *Please LOGIN first to build the map DIADesigner-AX fieldbus scan.*

5.2 Configuration

5.2.1 Create a Network Configuration Diagram

Follow the procedure to create a Network Topology:

1. Create a Project (refer to *3.3.1.1 Create 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.

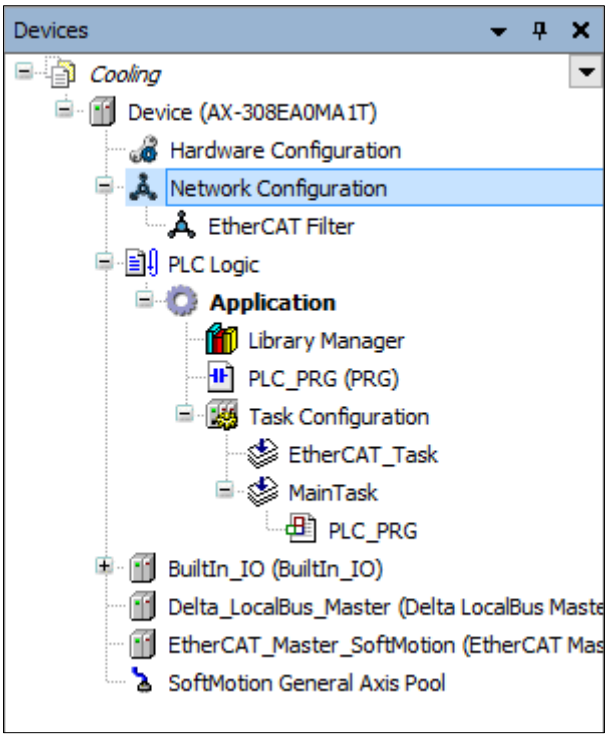


Figure 5 - 8: Creating Network Topology

Result: The *Network Configuration Editor* window displays, including the Topology and Product list.

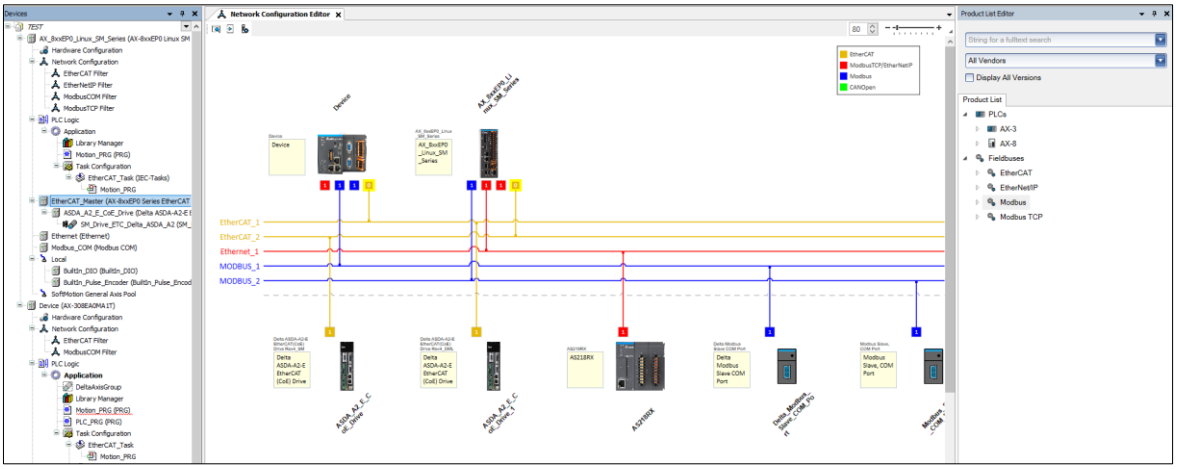


Figure 5 - 9: Network Configuration window

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.

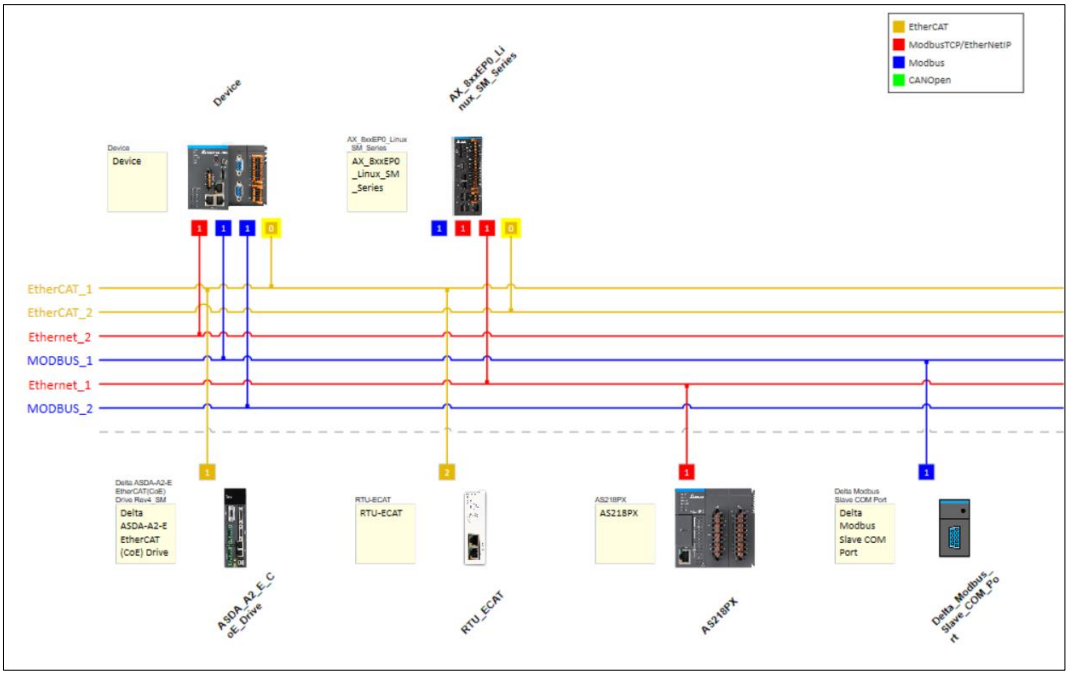


Figure 5 - 10: Network Configuration

6. Set as Master to connect the slave device: Right-click controller and choose *Set as Communication Master/Scanner* to set the major protocol.

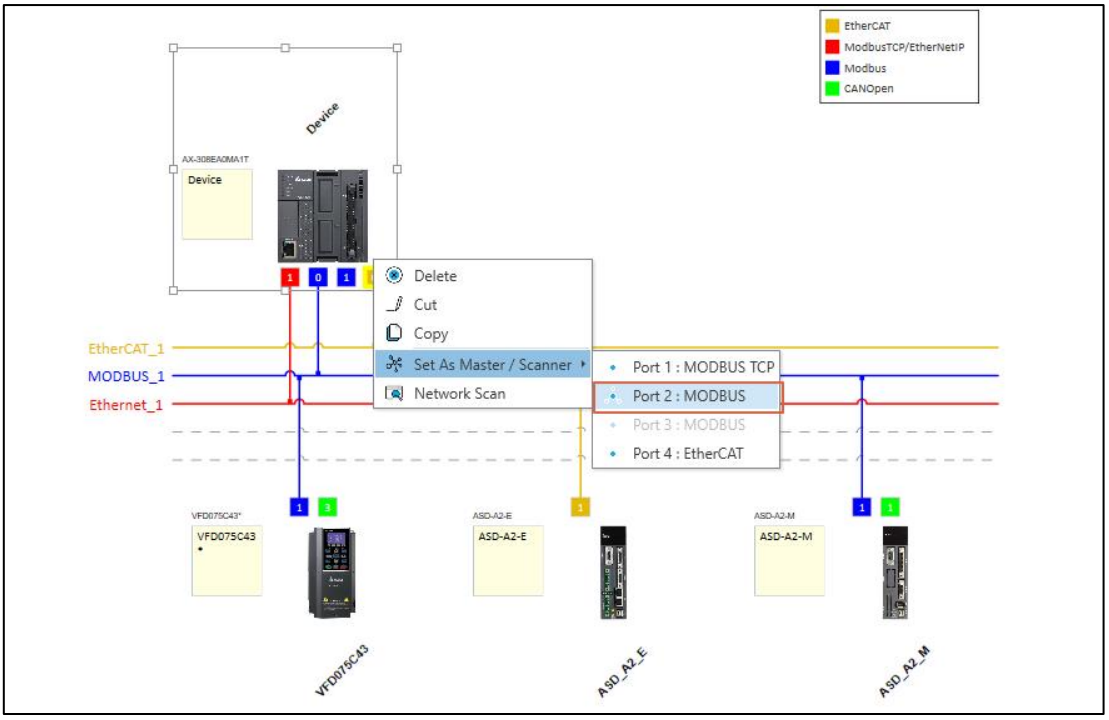


Figure 5 - 11: Setting Major Protocol

Then, project tree will show the related slave device. For example, set MODBUS as Master and the MODBUS slave will be on project tree.

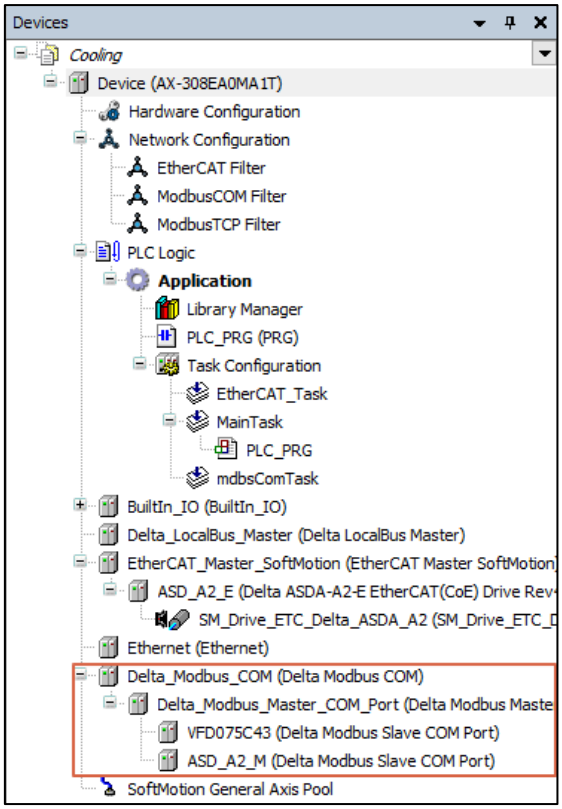


Figure 5 - 12: Project Tree

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.*

5.2.2 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.

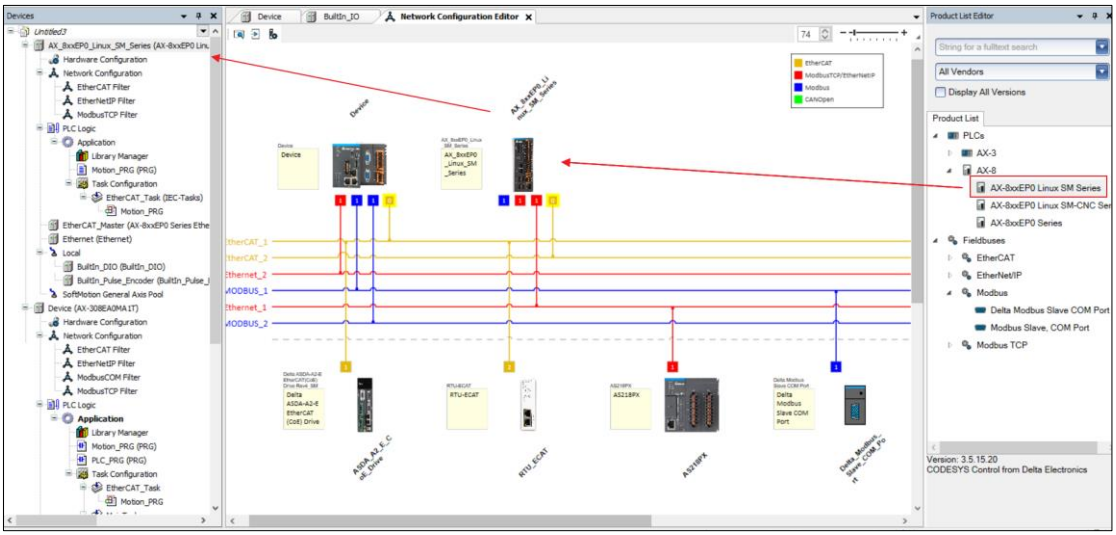


Figure 5 - 13: Network configuration diagram and project tree show controller name at the same time

2. Connect the correspond protocol to build project network topology

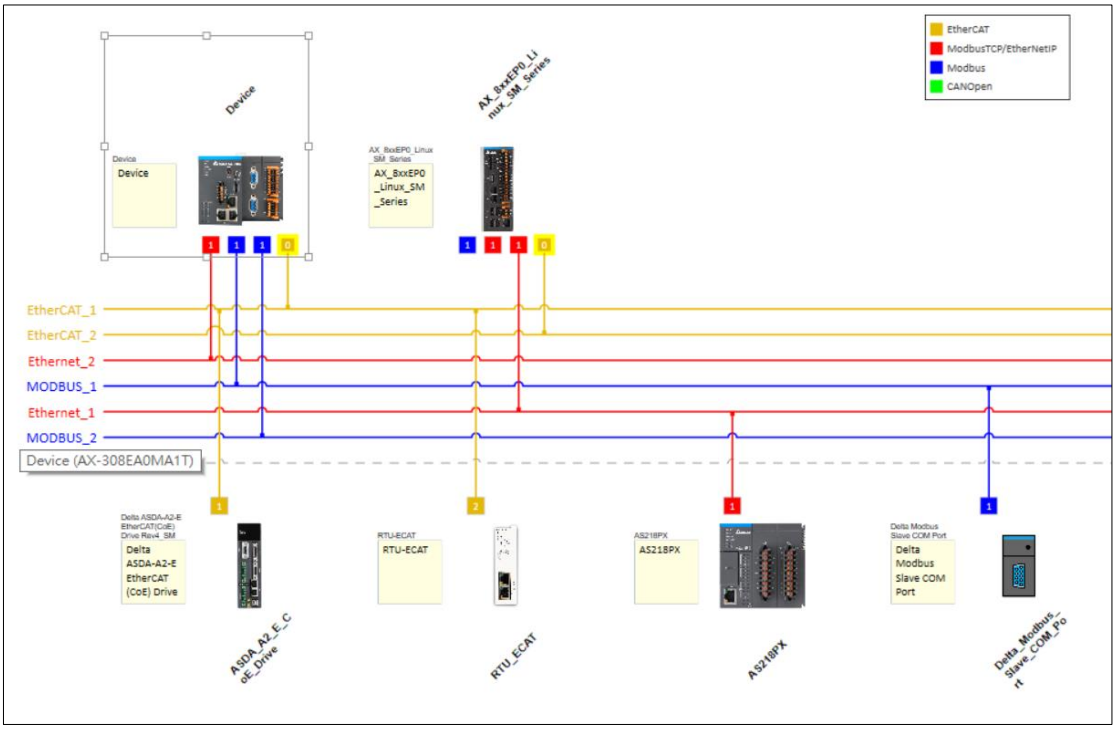


Figure 5 - 14: Connecting Protocol

3. Set as Master to connct the slave device: Right-click controller and choose *Set as Master/Scanner* to set the major protocol.

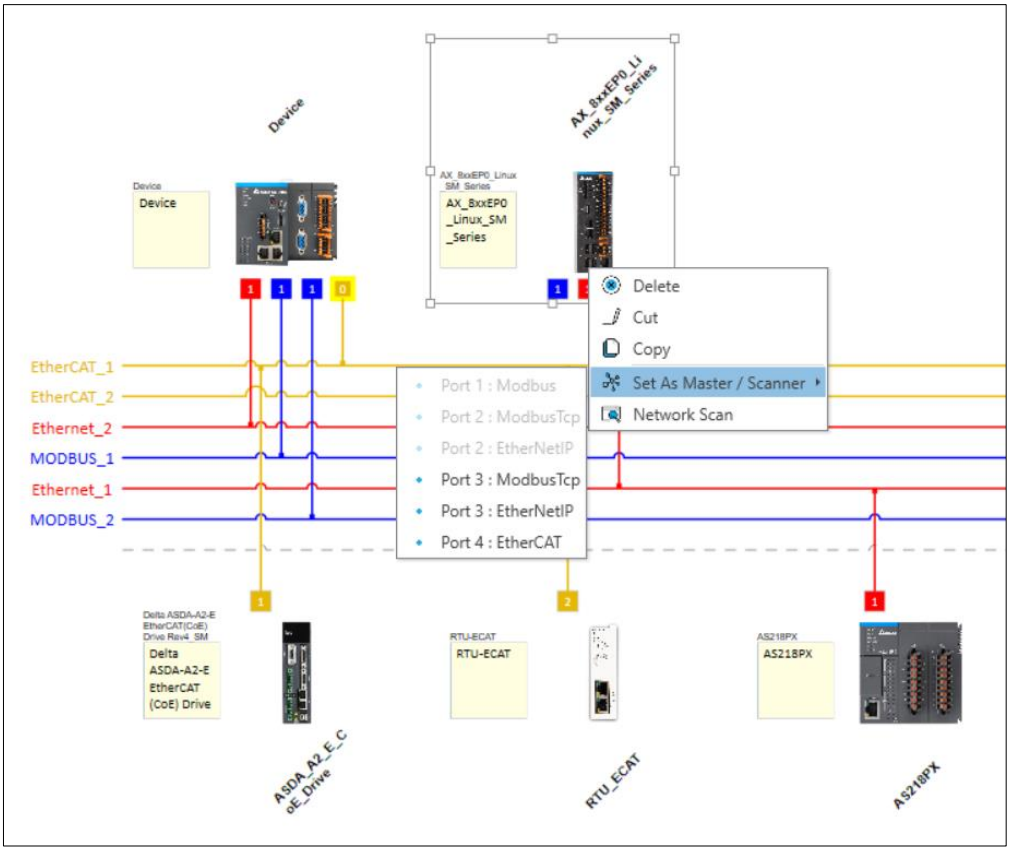


Figure 5 - 15: Setting Major Protocol

Project will show the related slave devices.

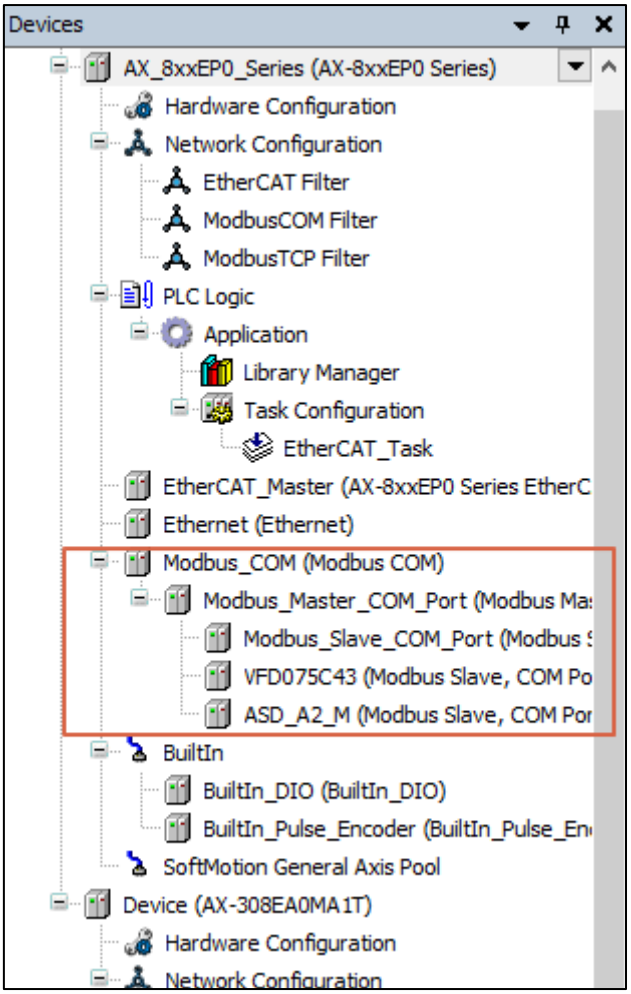


Figure 5 - 16: Slave devices added in Project Tree

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

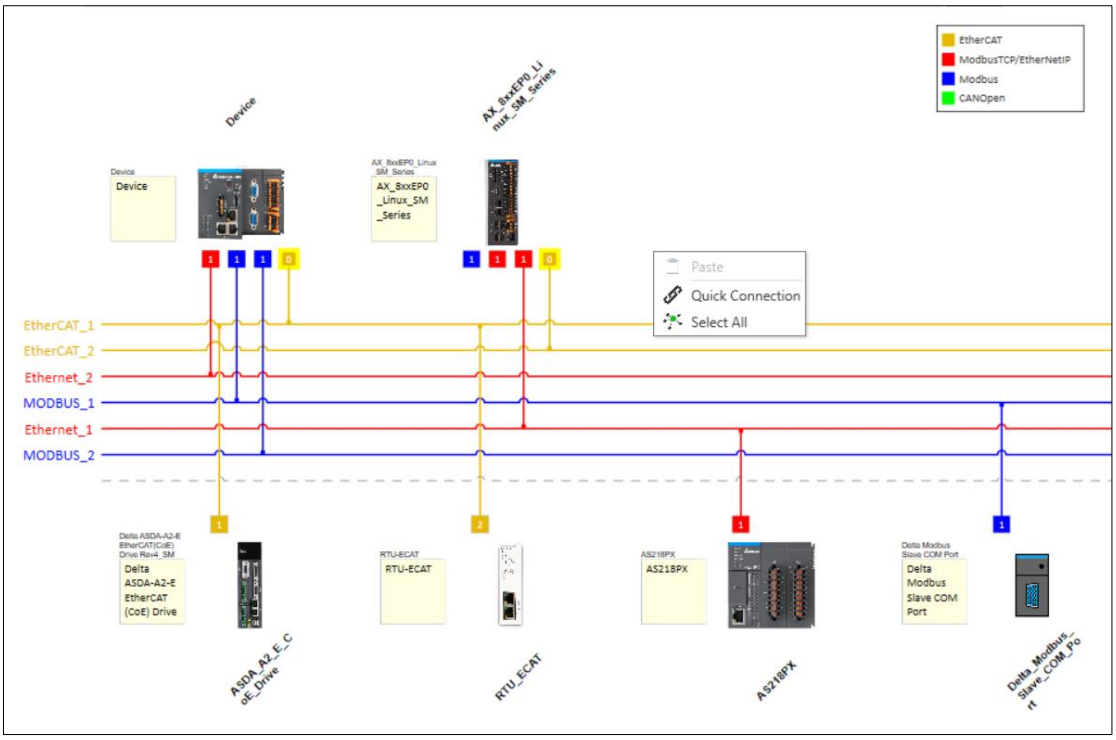


Figure 5 - 17: Quick connection

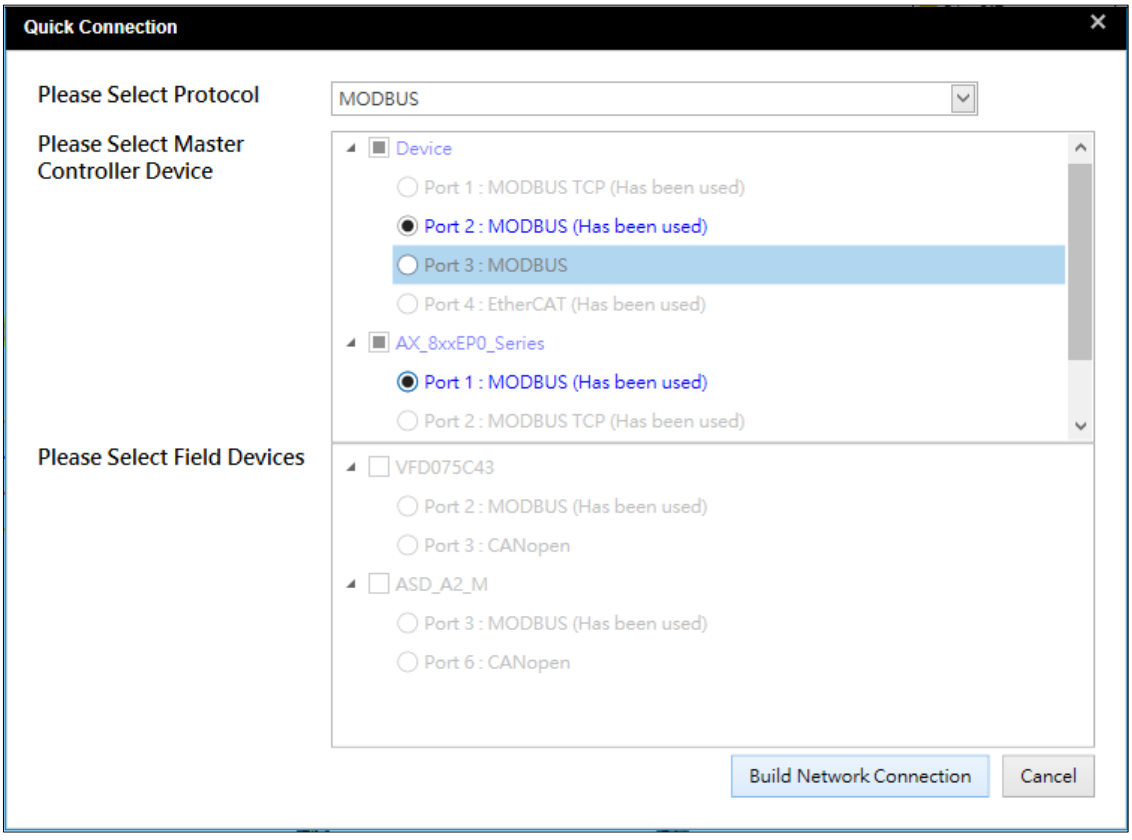


Figure 5 - 18: Quick Connection Window

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.
Build Network Connection	Click the button, the selected fieldbus and device will build and connect

5.3 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.

Figure 5 - 19: Communication Setting

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.

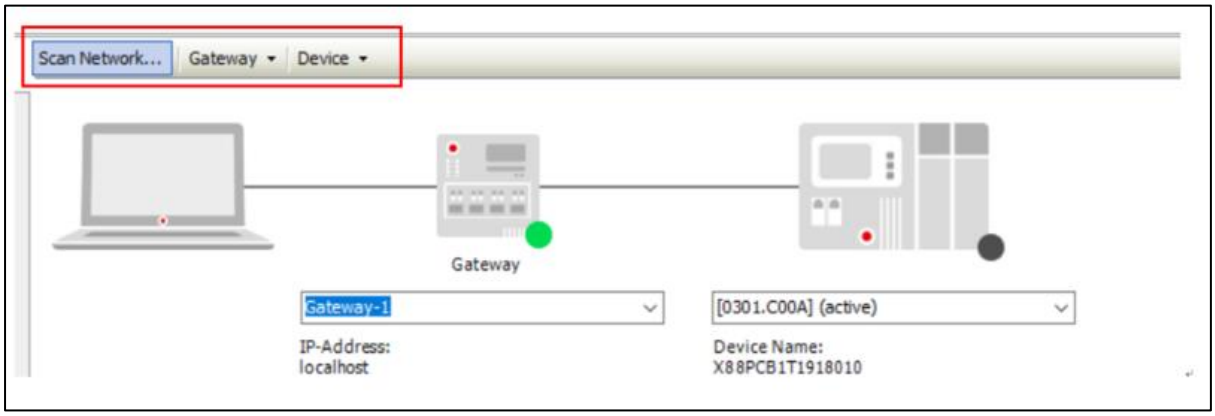
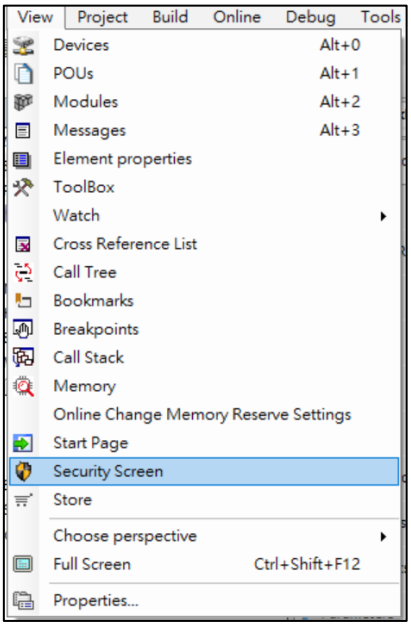


Figure 5 - 20: Scan Network

Function	Description
Scan Network	This button opens the *Select Device* dialog. It lists all configured gateways with the associated devices. User can select one target device from this list. Refer to the description of the classic view below for details about this dialog.
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: • *Add Current Device to Favorites*: Adds the currently set device to the list of favorite devices.

Function	Description
	• *Manage Favorite Devices*: Opens the favorites dialog with a list of all preferred devices. In this dialog, user can add or delete entries or change their order. The top device is the default. • *Rename Active Device*: Opens the *Change Device Name* dialog. • *Wink Current Device*: Devices that support this function illuminate a flashing signal. • *Send Echo Service*: DIADesigner-AX sends five echo services to the PLC. These are used to test the network connection, similar to the ping function. The services are sent first without data packets and then with data packets. The scope of the data packets depends on the communication buffer of the PLC. A message box opens with information about the average echo service delay and the scope of the sent data packets. • *Store Communication Settings in Project*: ☑ : DIADesigner-AX saves the communication settings in the project for reuse on the same computer. ***NOTE:*** *If user use the project on another computer, then user have to reset the active path.* ☐ : DIADesigner-AX saves the communication settings in the options of the local installation for reuse on the same computer. ***NOTE****: When using DIADesigner-AX SVN, the option should be cleared in order to prevent blocking the device object.* • *Confirmed Online Mode*: ☑ : DIADesigner-AX requires user to confirm the following when calling the following online commands (for safety purposes): Force values, write values, multiple loading, Remove force list, Single cycle, Start, Stop. • *Filter Network Scans by Target ID*: ☑ : The display is limited on the devices that have the same target ID as the current device configured in the project. • *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

Function	Description
	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*  • In *Security Screen,* if choosing *Enforce encrypted communication* in Security Level, *Encrypted Communication* will be ineffective.  • *Change Communication Policy* Opens the *Change Communication Policy* dialog for changing the device setting for the encryption of communication.

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*: 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*: 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.

5.3.1 EtherCAT

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 figure below.

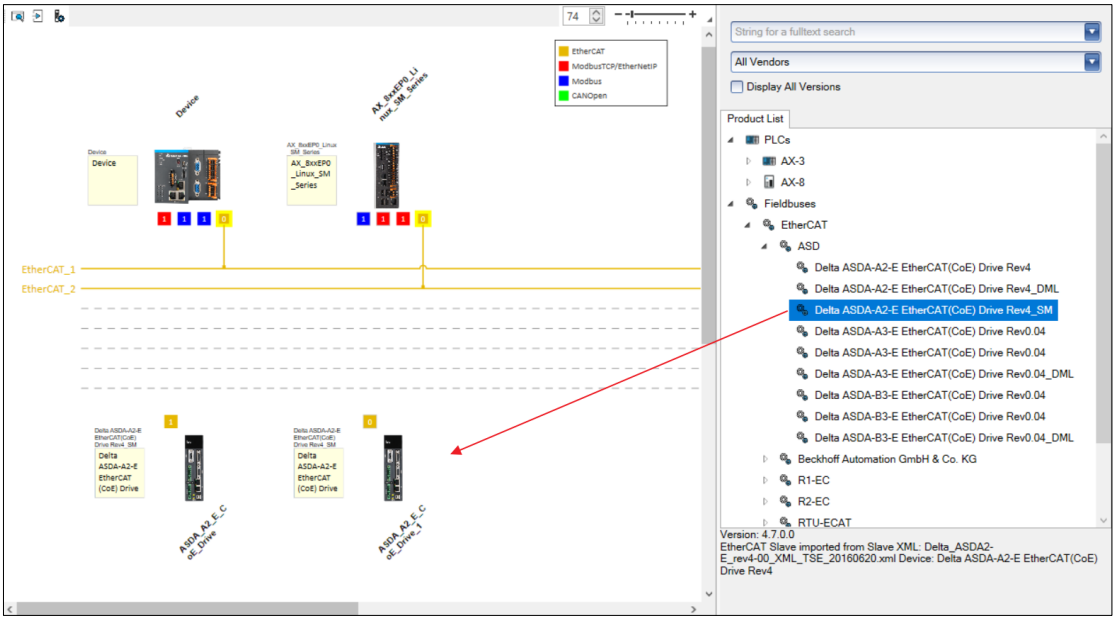


Figure 5 - 21: Adding device to network configuration

4. Connect the device to the network as shown in the following figure.

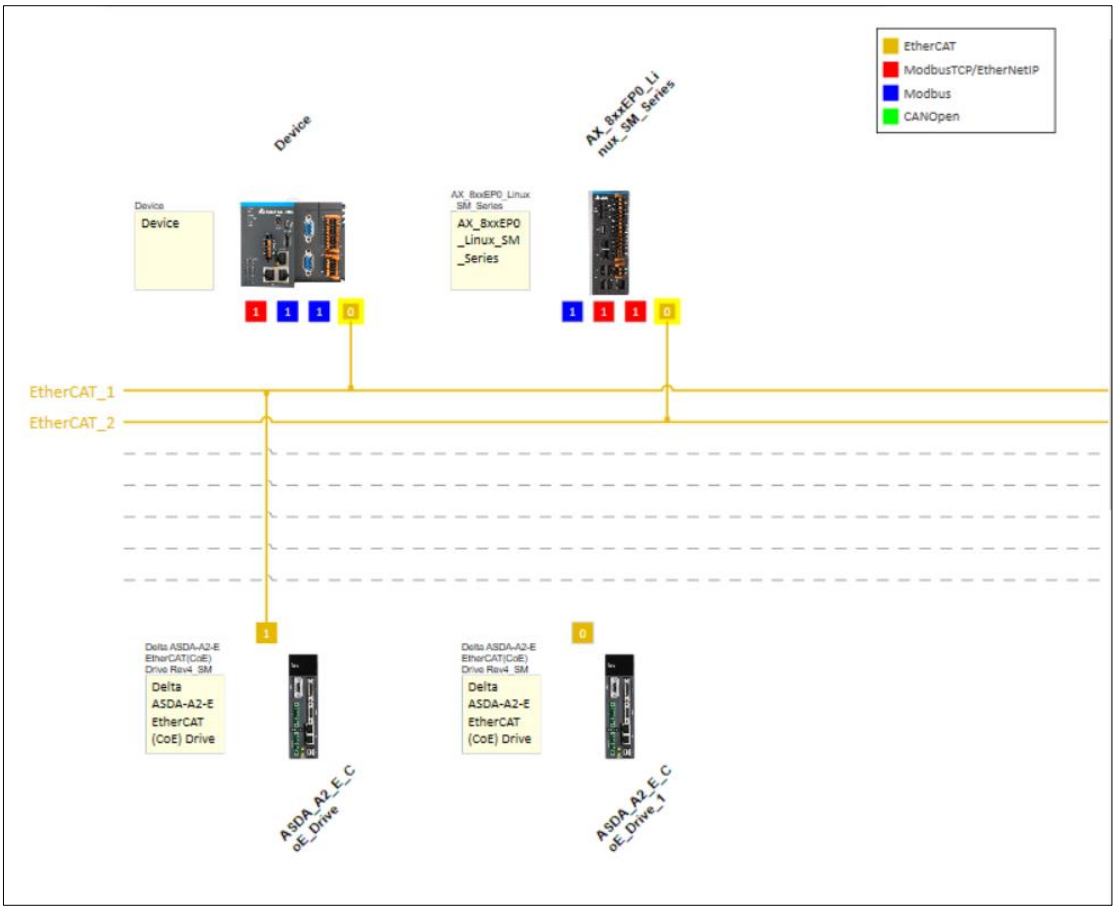


Figure 5 - 22: Connecting device to network configuration

5. Double-click on the master station node as shown in the following figure.

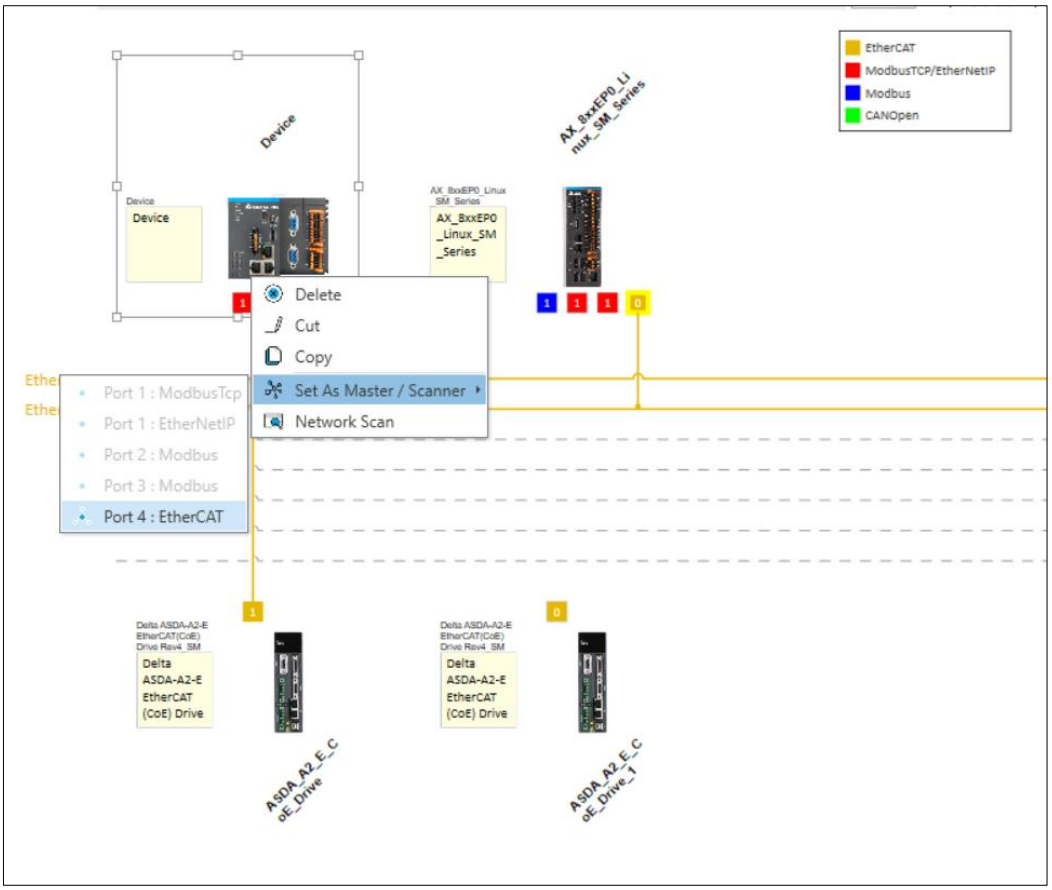


Figure 5 - 23: Clicking on Master station

Result: Displays the *EtherCAT* master station.

6. Set the EtherCAT synchronization time and other required settings as shown in the following figure.

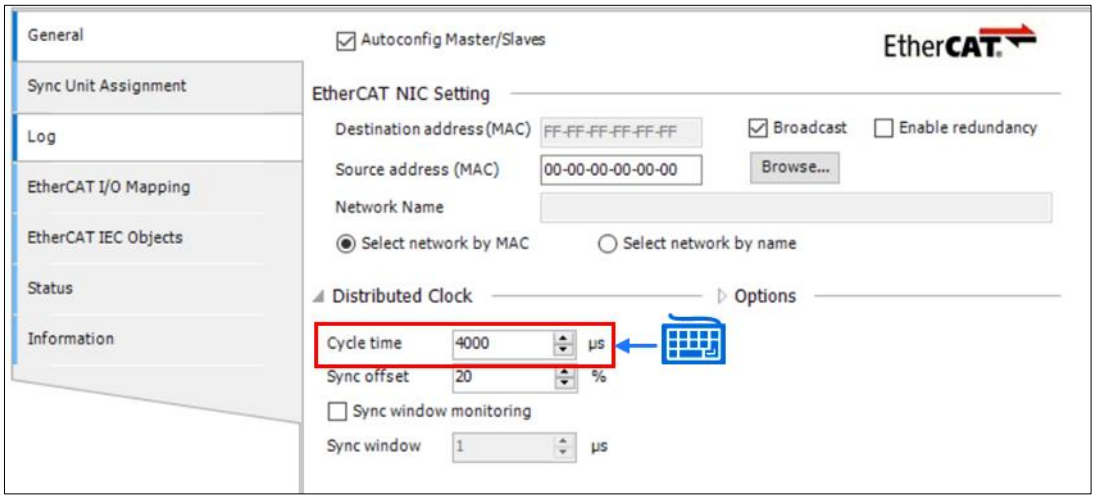


Figure 5 - 24: Set EtherCAT synchronization time

7. Double-click on the slave device to set the EtherCAT settings as shown in the following figure.

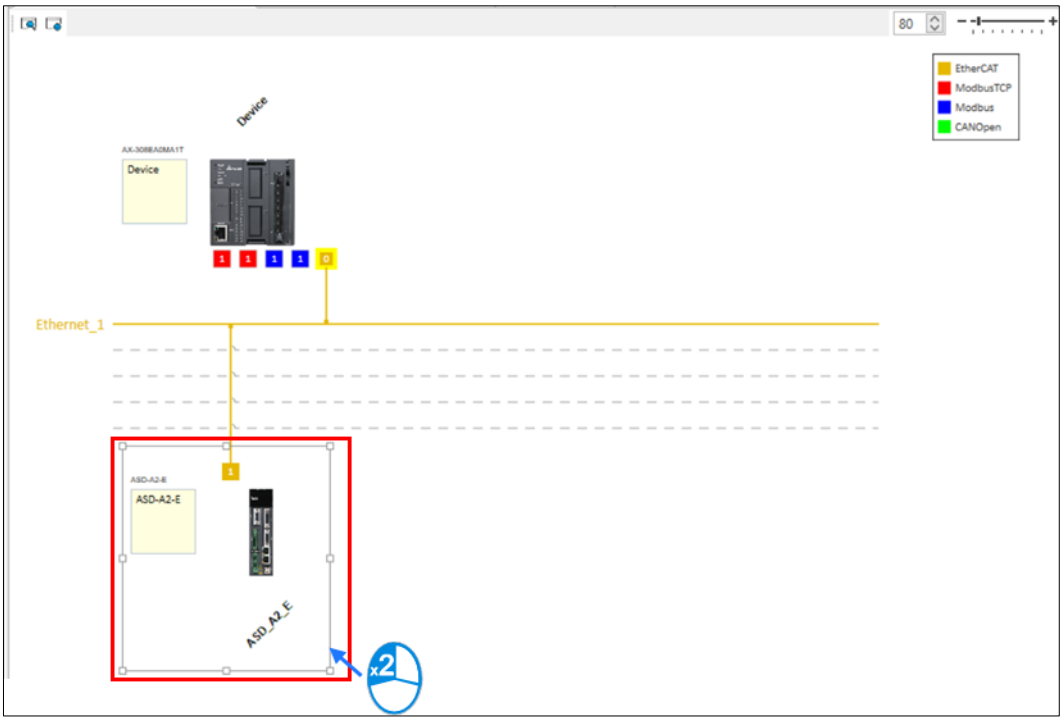


Figure 5 - 25: Click on slave device

8. Set the required EtherCAT settings as show in the following figure.

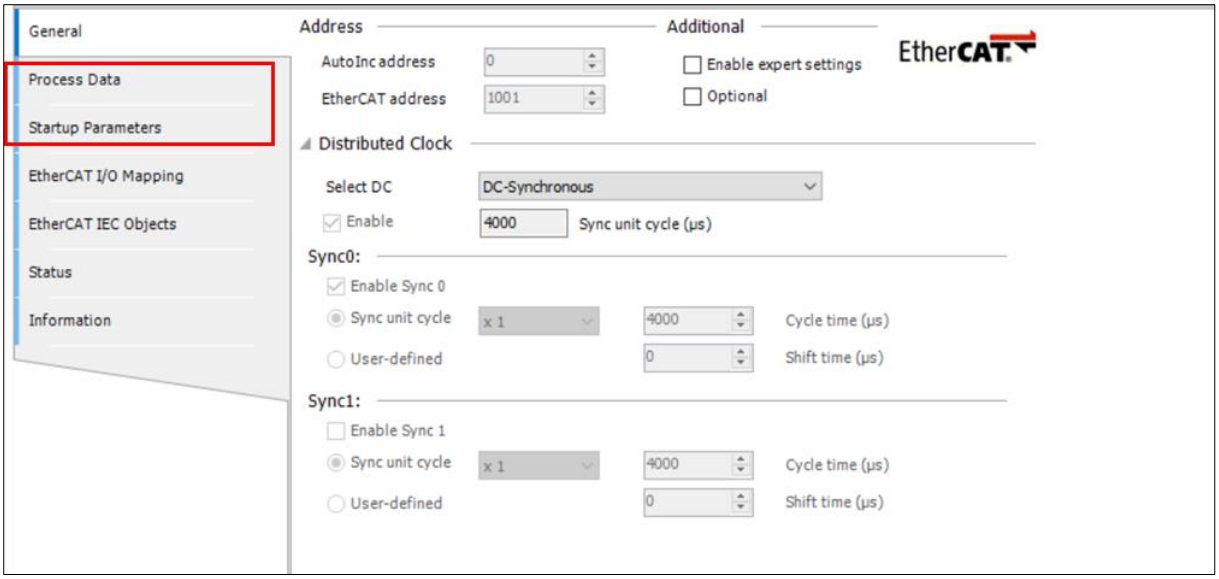


Figure 5 - 26: EtherCAT Setting

***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.*

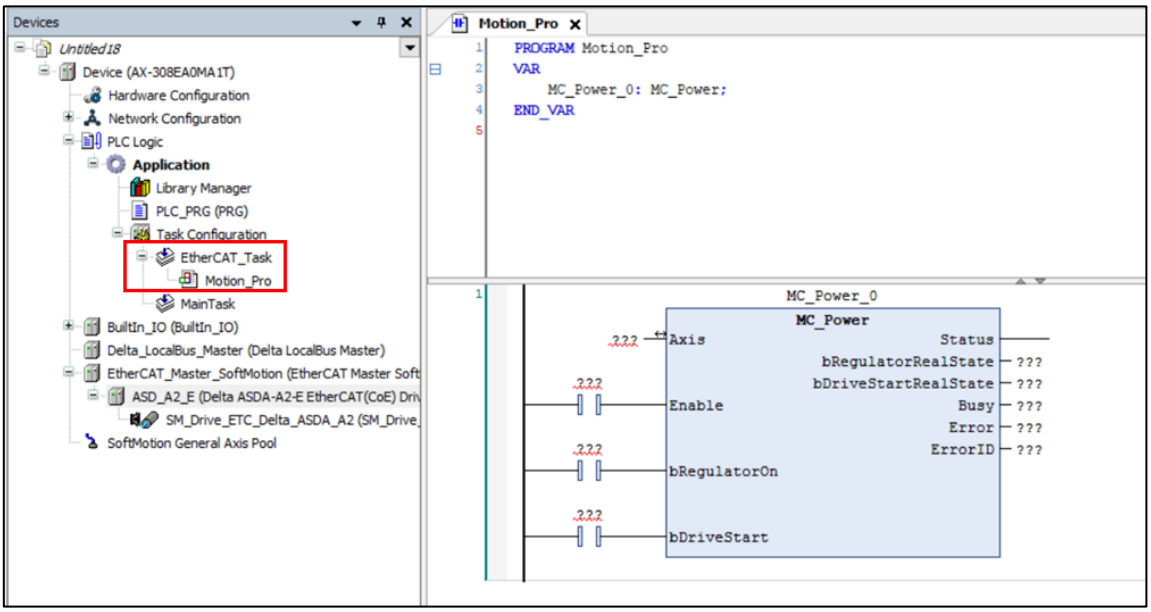


Figure 5 - 27: EtherCAT added to device

5.3.2 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 figure below.

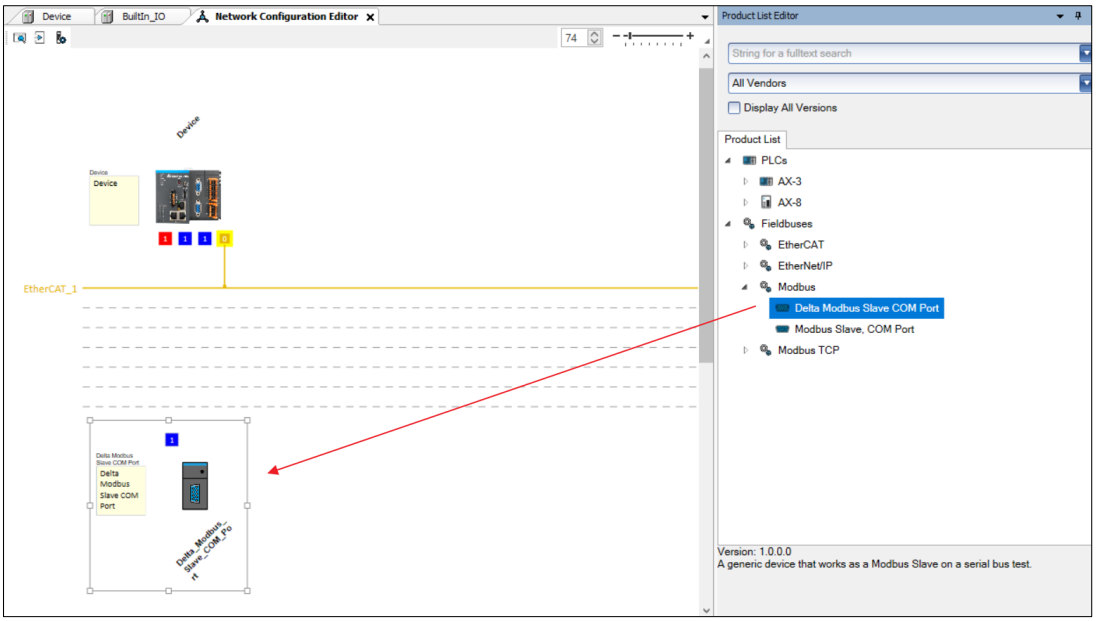


Figure 5 - 28: Adding MODBUS COM

2. Connect to the MODBUS line, Set controller as *MODBUS Master*, and double-click field device to open the parameter page.

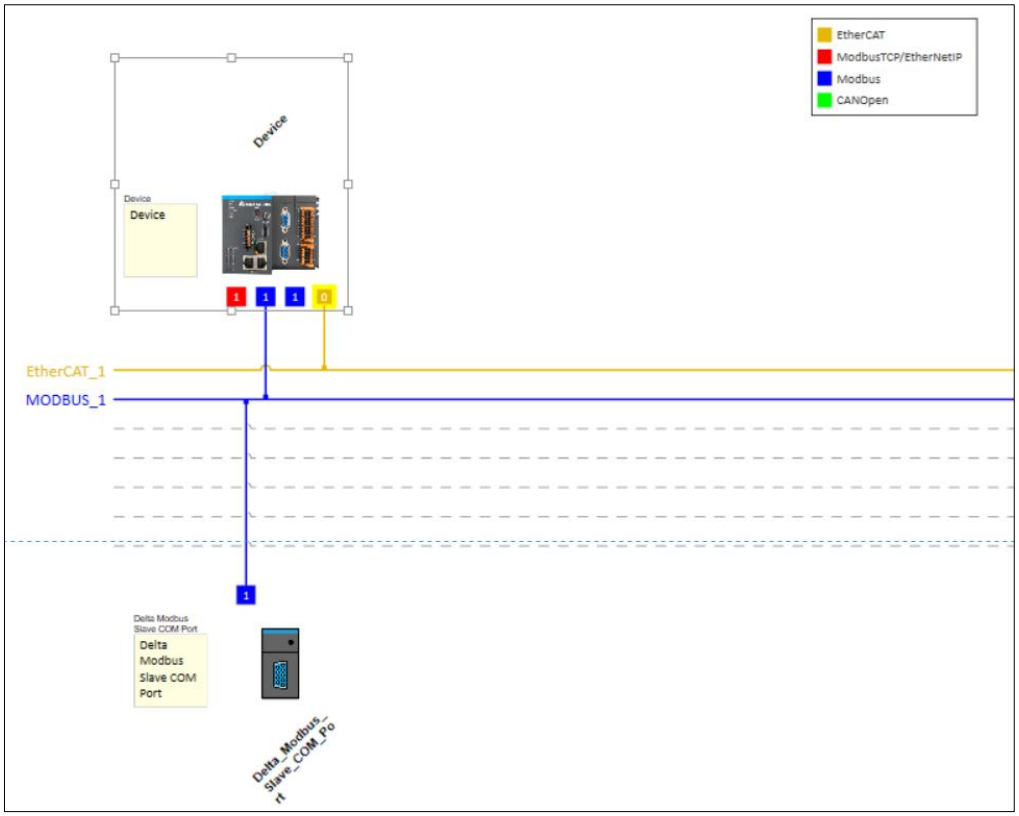


Figure 5 - 29: Setting Controller

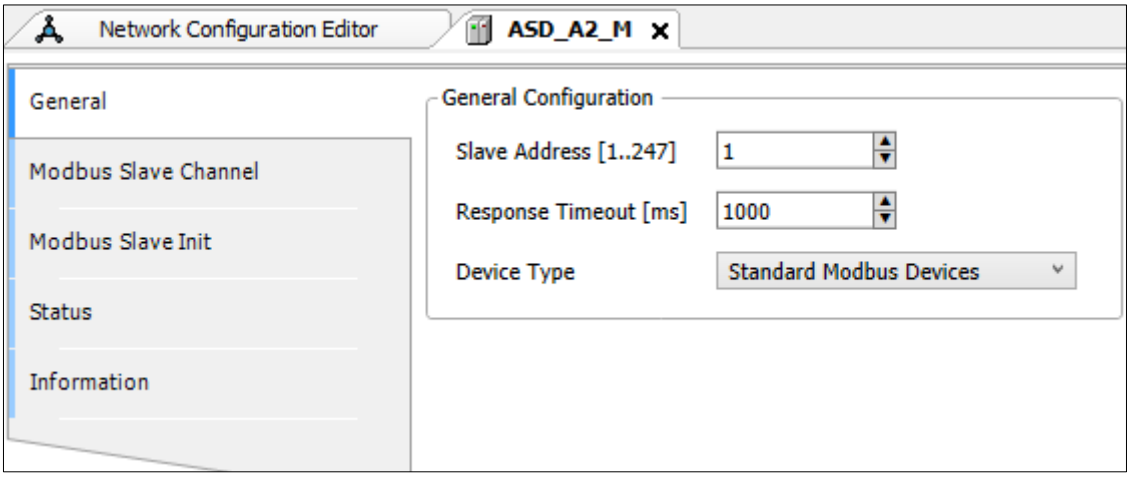


Figure 5 - 30: ASD_A2_M – General Configuration

For more information, please refer to *4.2.3 COM Port Setting* and *4.3.2 COM Port Setting* setting.

5.3.3 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.

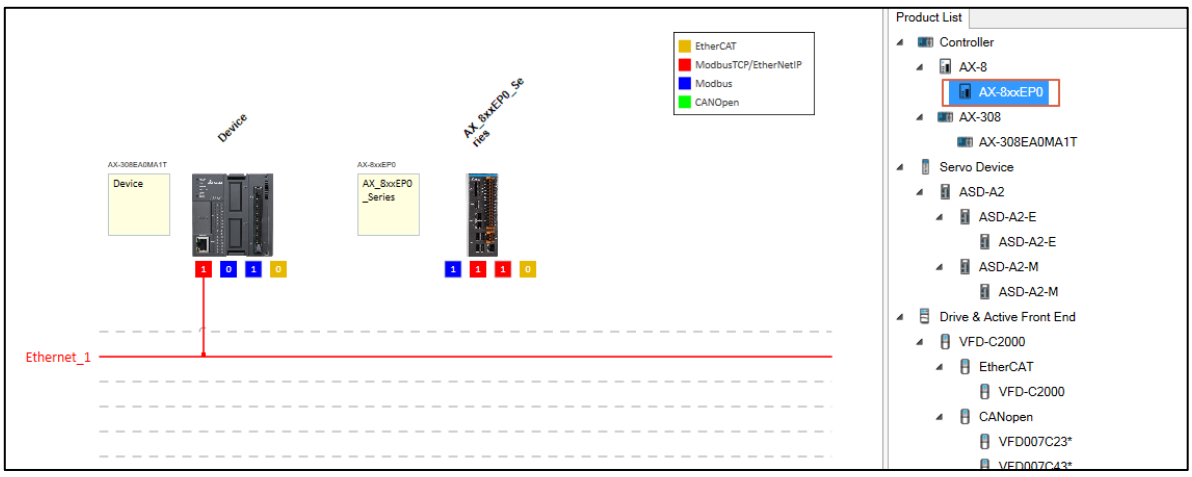


Figure 5 - 31: Adding Ethernet Device

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.

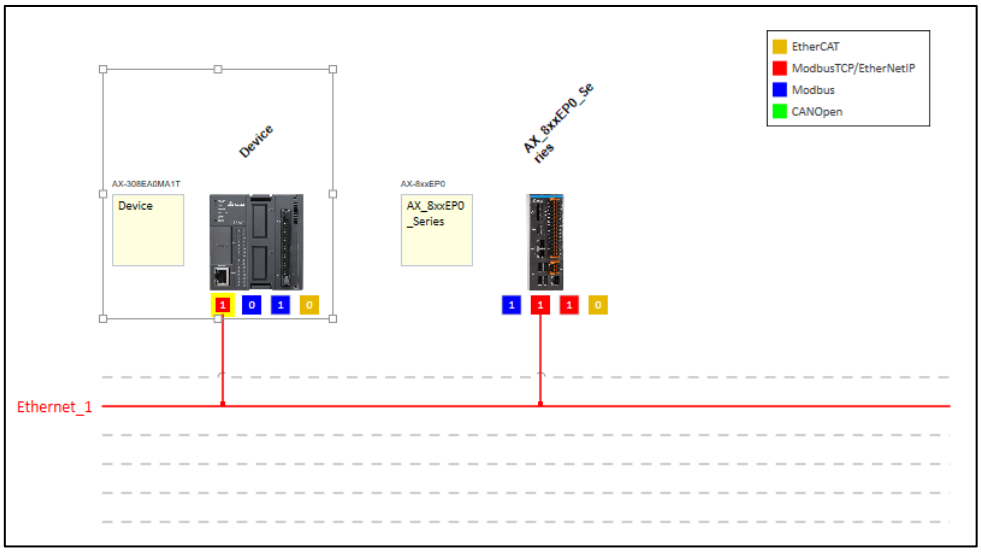


Figure 5 - 32: Setting AX- 3 Controller

User can go to project tree to open the MODBUS TCP slave parameter.

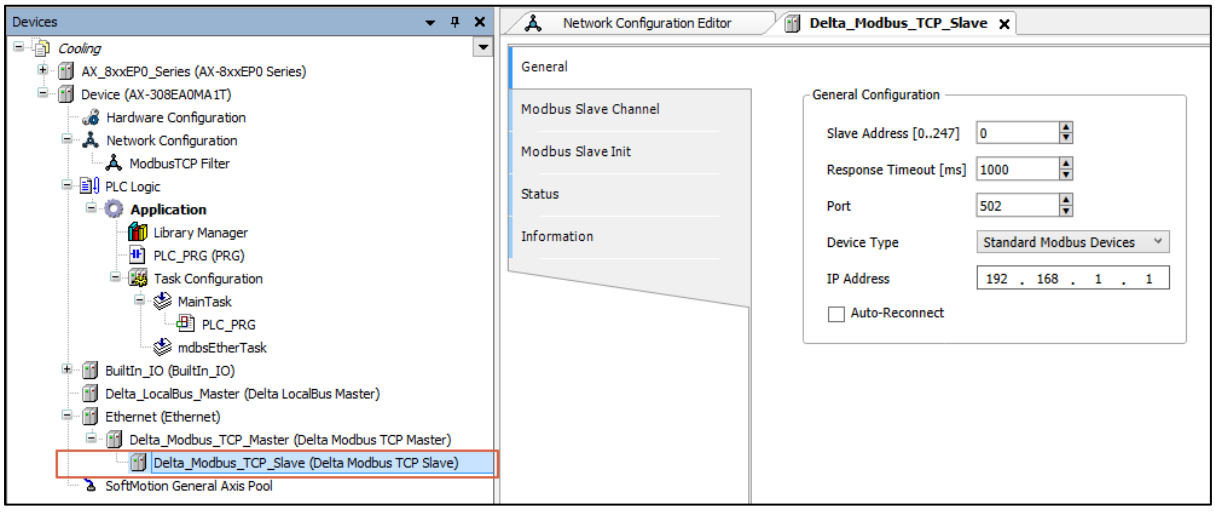


Figure 5 - 33: MODBUS TCP Slave parameter

For more information, please refer to *4.2.5 EtherNet Setting* for AX-3 TCP port setting and *4.3.4 EtherNet Setting* for AX-8 TCP port setting.

5.3.4 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.

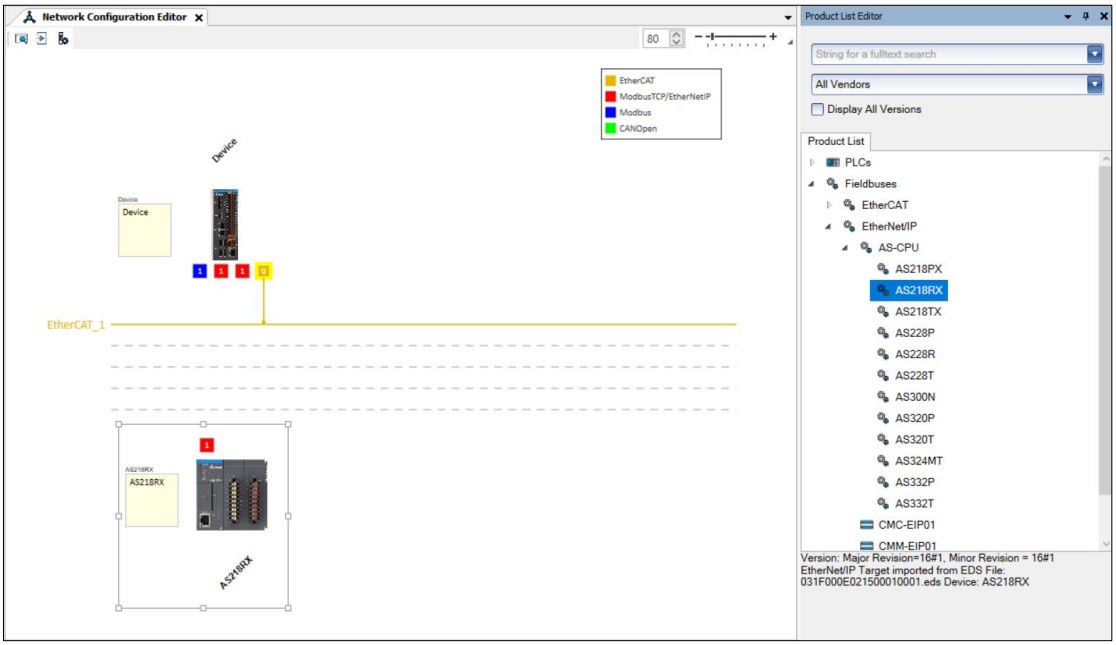


Figure 5 - 34: Add MODBUS COM device

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.

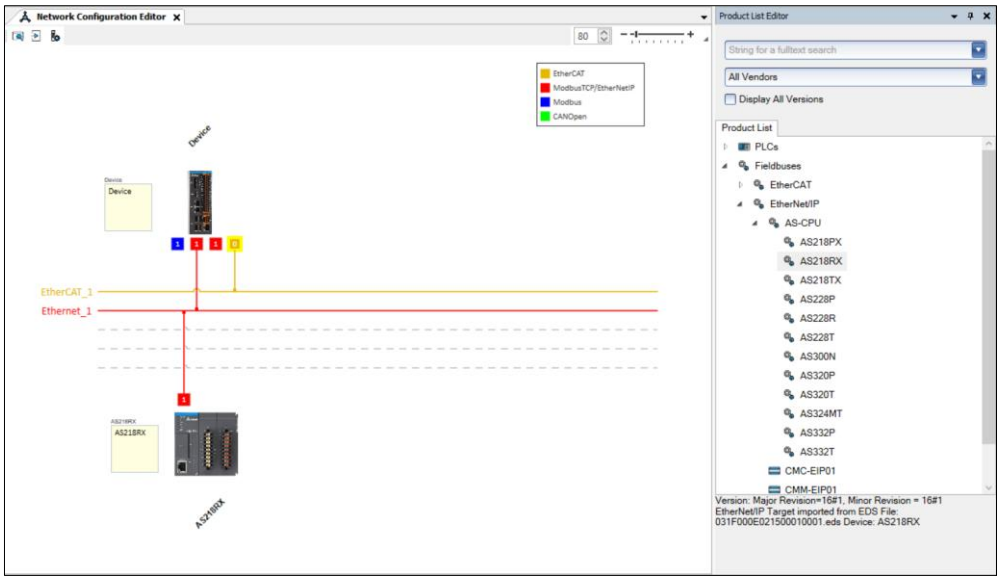


Figure 5 - 35: Connect the Ethernet line

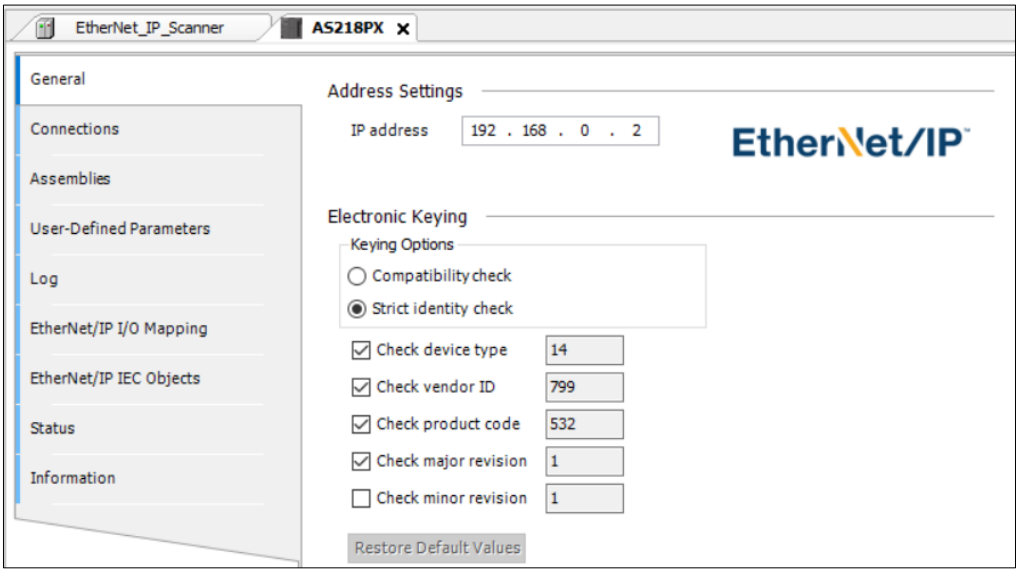


Figure 5 - 36: General tab

For more information on AX-308E EtherNet/IP settings, please refer to *4.2.6 EtherNet/IP Settings.* For more information on AX-8 EtherNet/IP settings, please refer to *4.3.5 EtherNet/IP Settings*.

5.3.5 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.

5.3.5.1 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.

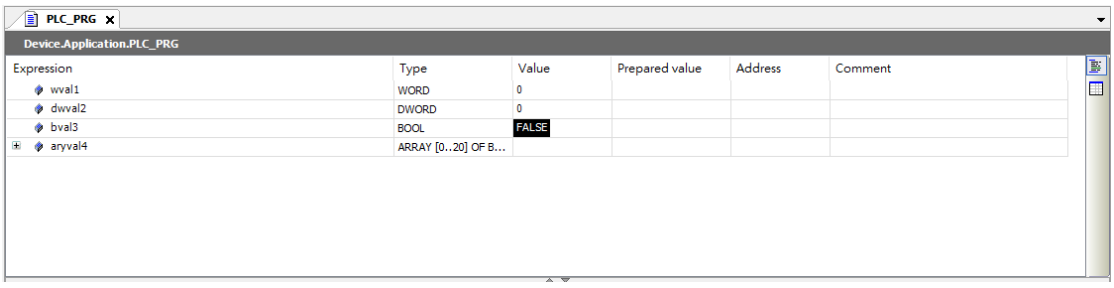


Figure 5 - 37: **PLC_PRG Create variables in**

3. From Application adds a symbol configuration, *Application* > *Add Object* > *Symbol Configuration*.

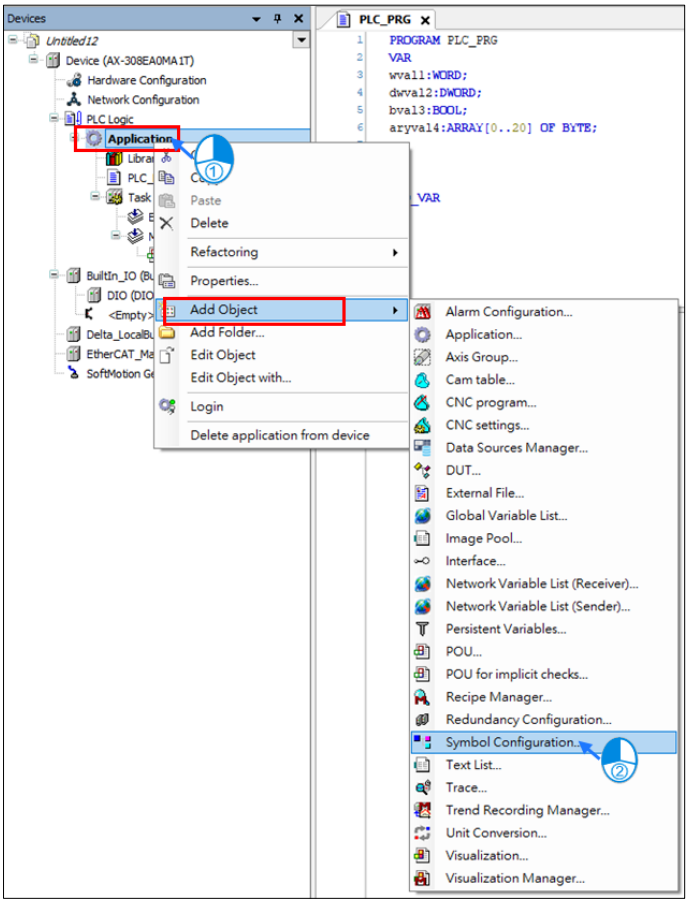


Figure 5 - 38: **Add symbol configuration**

4. In the *Add Symbol Configuration* dialog box, select, *Support OPC UA features* and click on *Add* button.

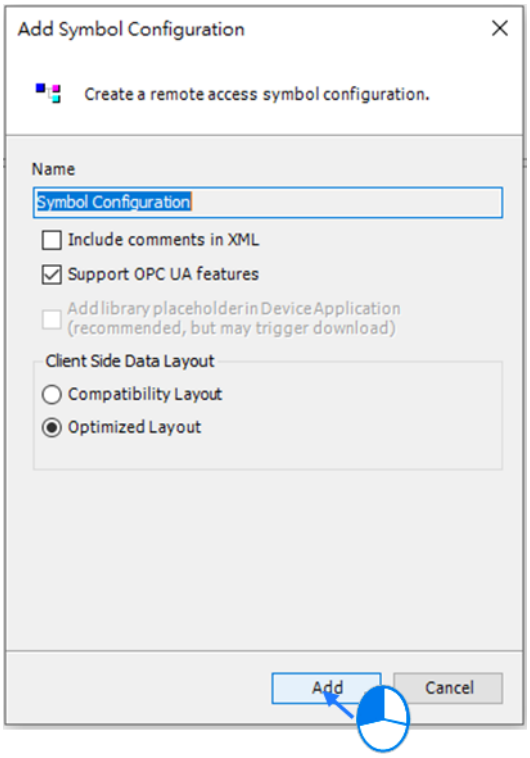


Figure 5 - 39: **Add symbol configuration dialog**

5. Open the symbol configuration setting interface and click compile: The created projects and variables are displayed in the tree structure table below.

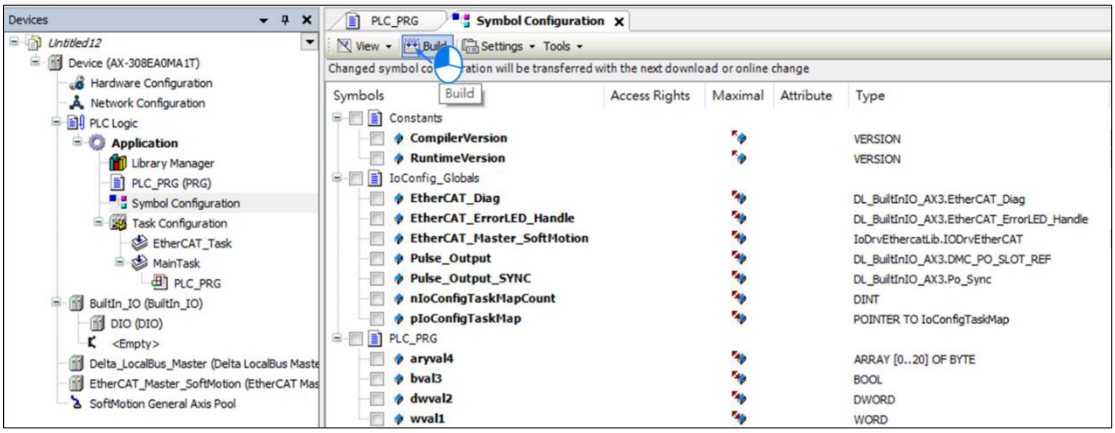


Figure 5 - 40: **Symbol configuration setting interface**

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.

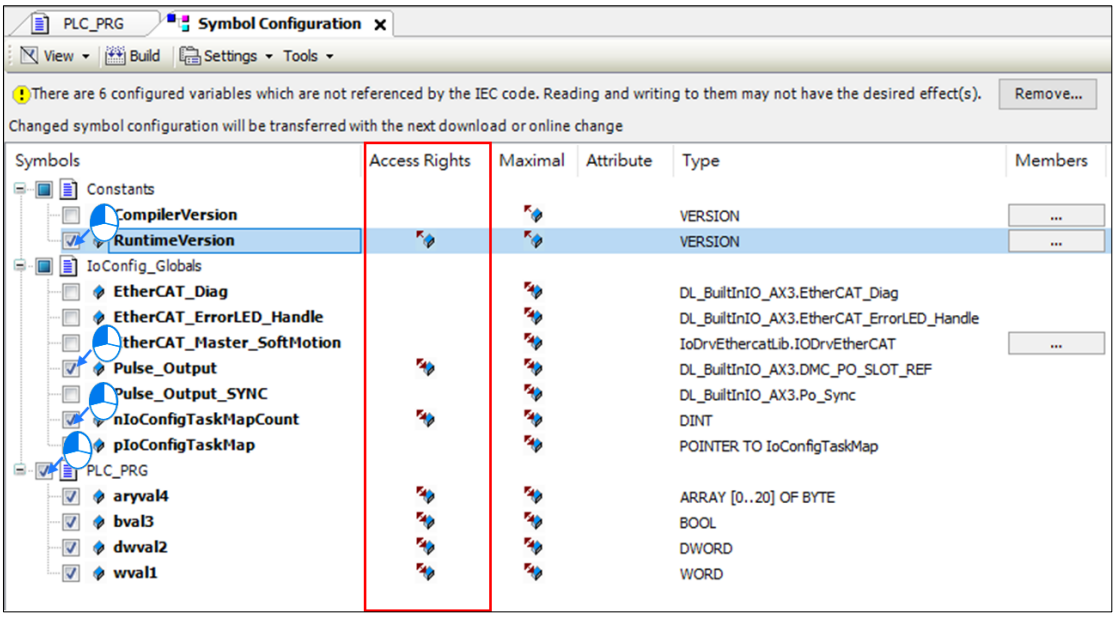


Figure 5 - 41: **Change symbol configuration settings**

7. Finally, download the project to the host.

5.3.5.2 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*.

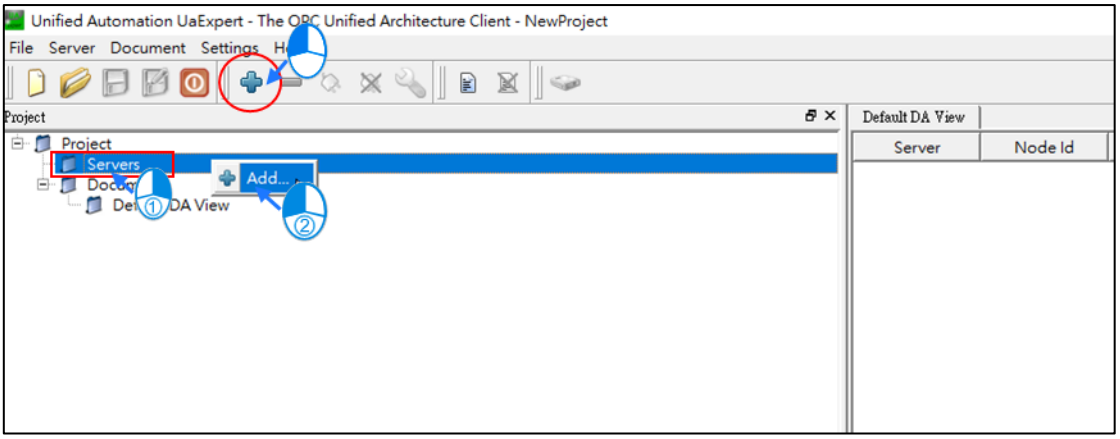


Figure 5 - 42: **Add Server**

3. Click *Custom Discovery* > *Double click to Add Server* > to pop up the *Enter Url* dialog box, enter opc.tcp://192.168.1.5.

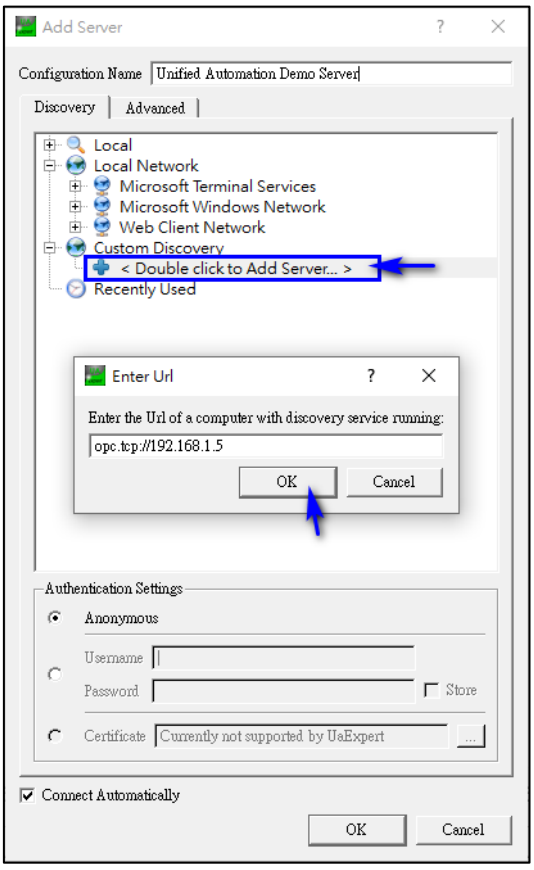


Figure 5 - 43: **Enter Url Dialog box**

4. 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).

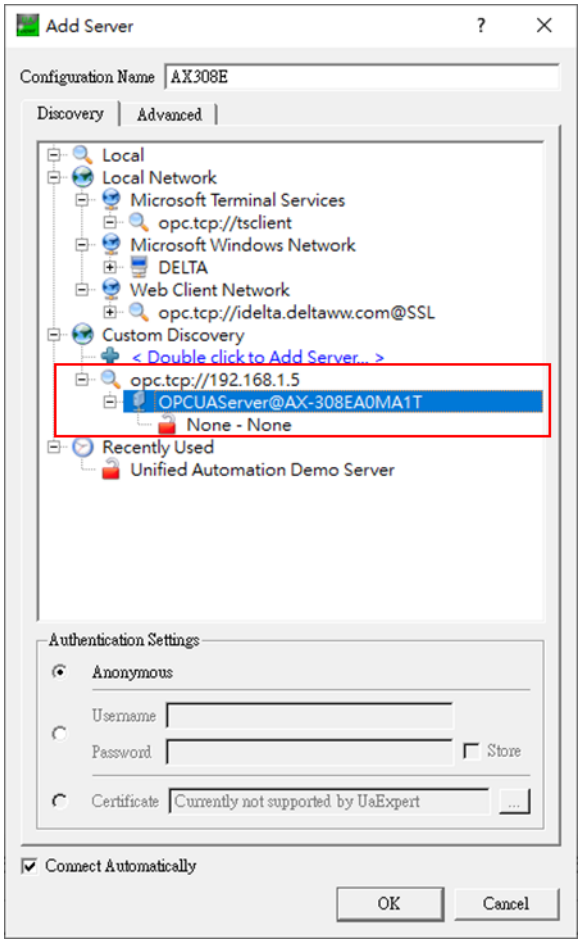


Figure 5 - 44: Connection display

5. Modify the connection content: *Project* > *AX308E* and click the right mouse button > *Properties*.

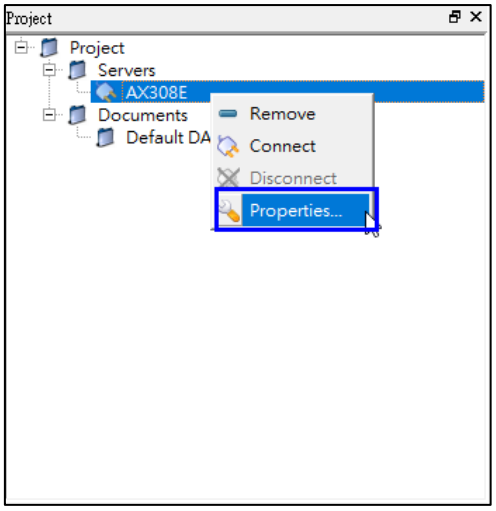
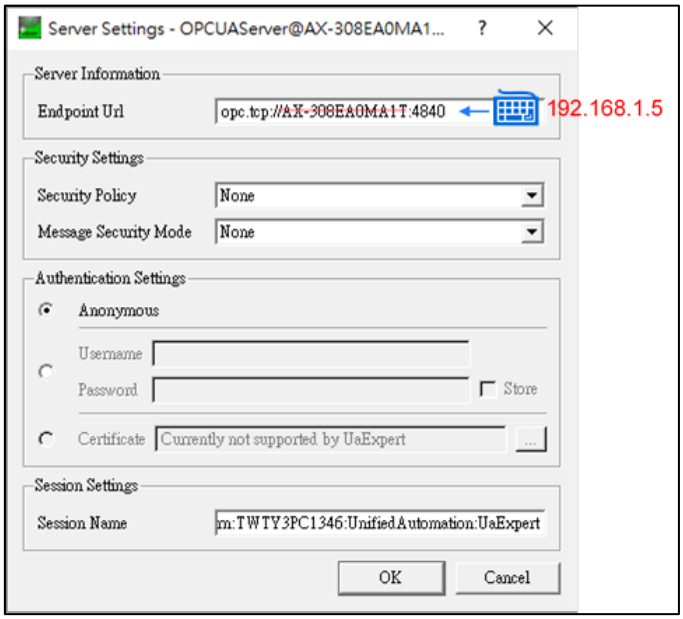


Figure 5 - 45: Modify connection content

6. Enter in *Endpoint Url* in the Server Settings dialog box *opc.tcp://192.168.1.5:4840*.



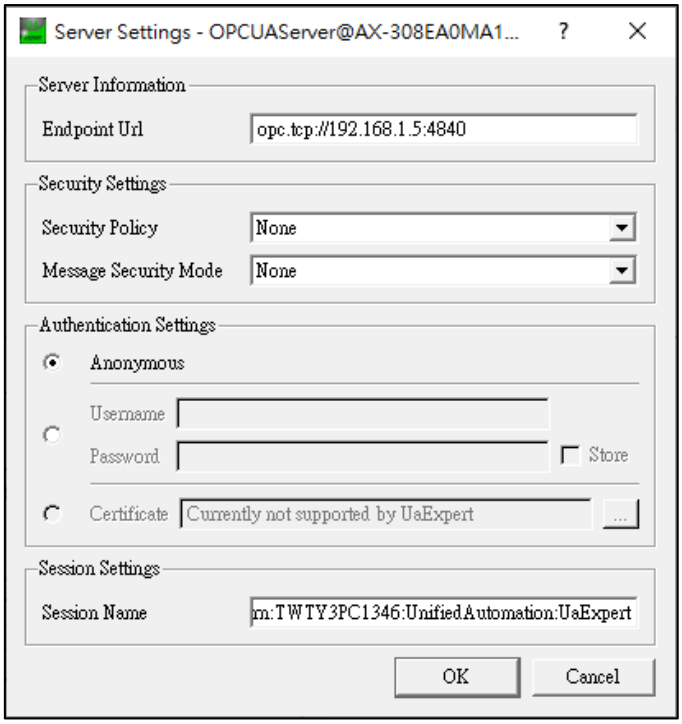


Figure 5 - 46: change Endpoint Url

7. Click *Project* > *AX308E,* then, right-click > *Connect.*

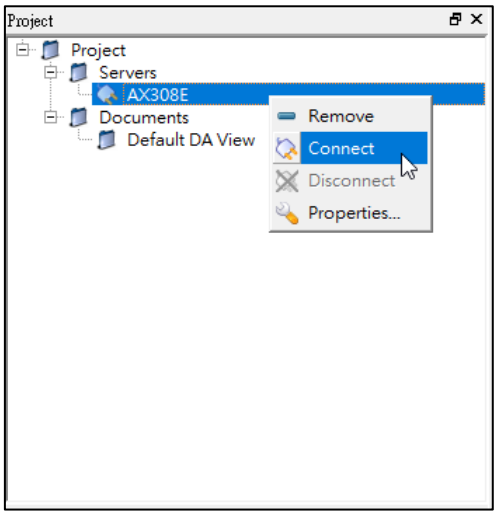


Figure 5 - 47: AX308E Connect

8. 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.

5.3.5.3 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.

- **OPC UA Server Start certification**

  To set the encryption method of OPC UA Server, user need to use Matrikon ® FLEX™ OPC UA Editor to configure the settings (DIADesigner-AX has been installed in advance).

1. After the project is created, right-click *Device* > *Add Object* > *Matrikon FLEX OPC UA Server*. In Matrikon FLEX OPC UA Server dialog box, Add *Matrikon FLEX OPC UA Server* and click on *Add* button.

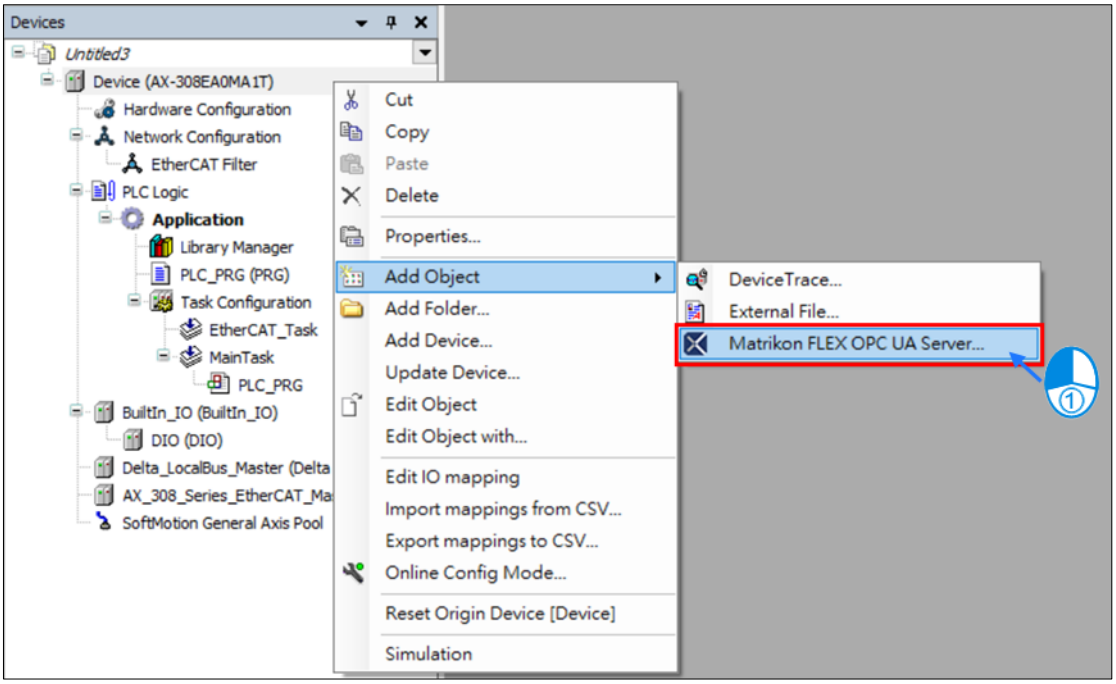


Figure 5 - 48: Add Object

Result: The Project tree displays *Matrikon FLEX OPC UA Server* in the project list.

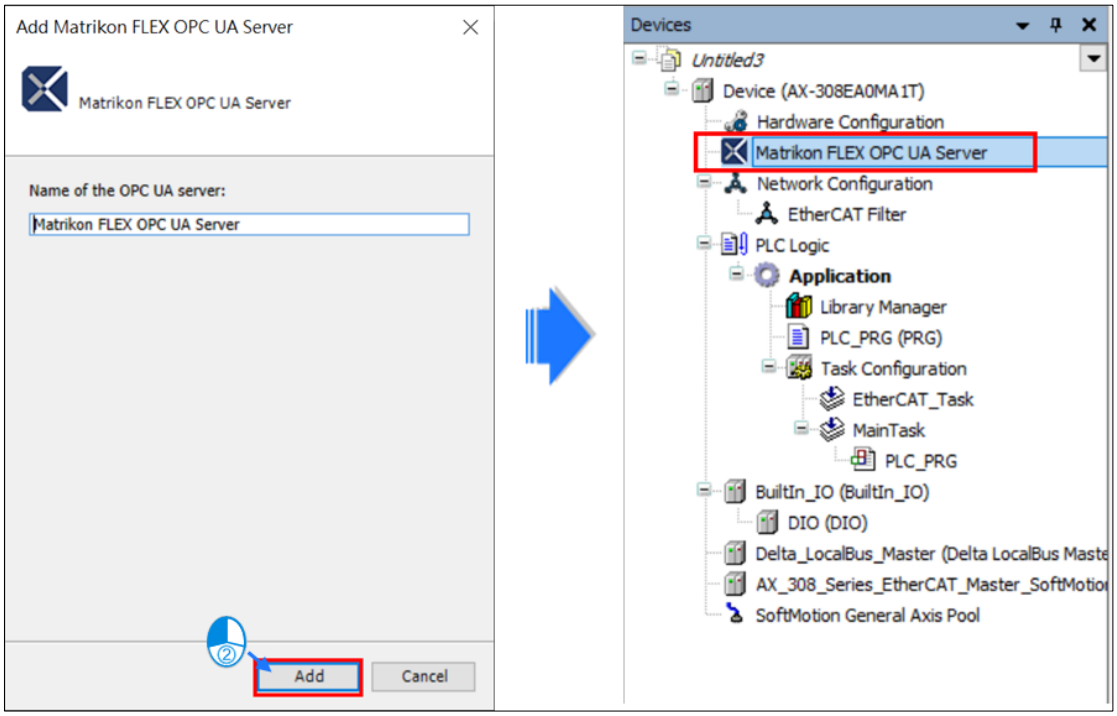


Figure 5 - 49: **Project tree display Matrikon FLEX OPC UA Server**

2. Matrikon FLEX OPC UA Server Setting interface.

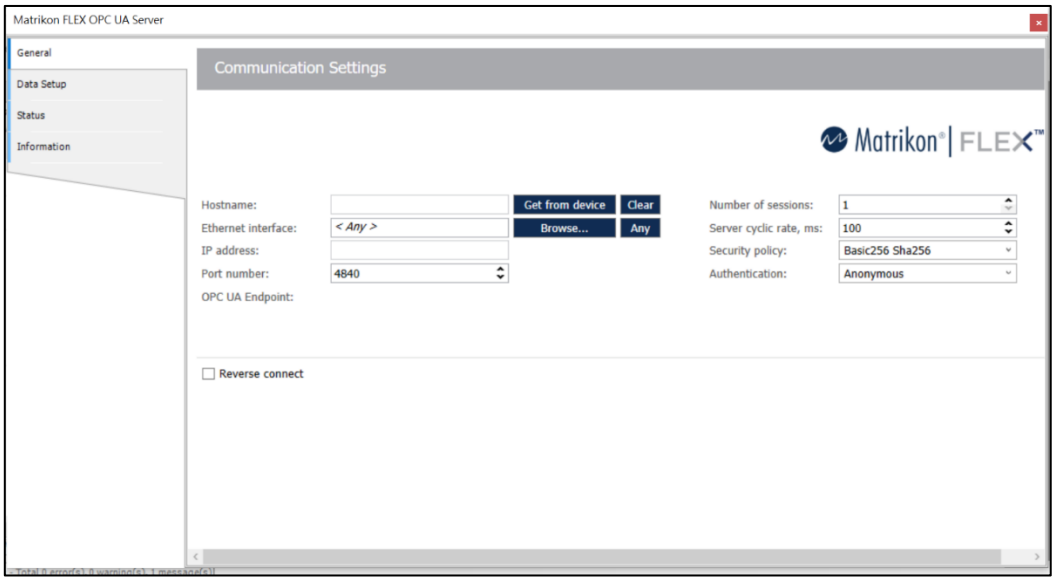


Figure 5 - 50: **Matrikon FLEX OPC UA Server Setting interface**

3. Set up the network communication interface.

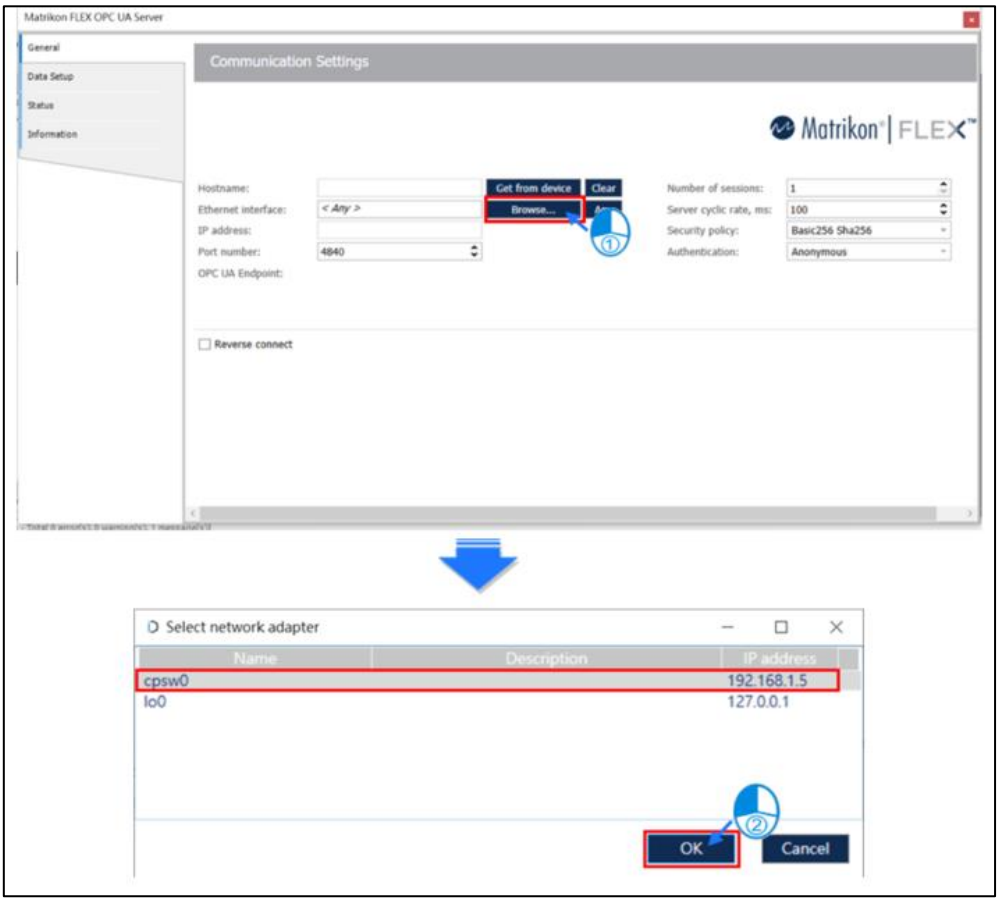


Figure 5 - 51: **Matrikon FLEX OPC UA Server Set up the network communication interface**

4. Setting complete screen. (Other fields do not need to modify settings).

Figure 5 - 52: **Matrikon FLEX OPC UA Server Setup complete screen**

5. After setting, please download to the PLC host.

- **Set user account password**

The OPC UA Server account password used is the same as the account password of the host encryption.

Set up two accounts:

1. Default account：Administrator
2. New account：guest

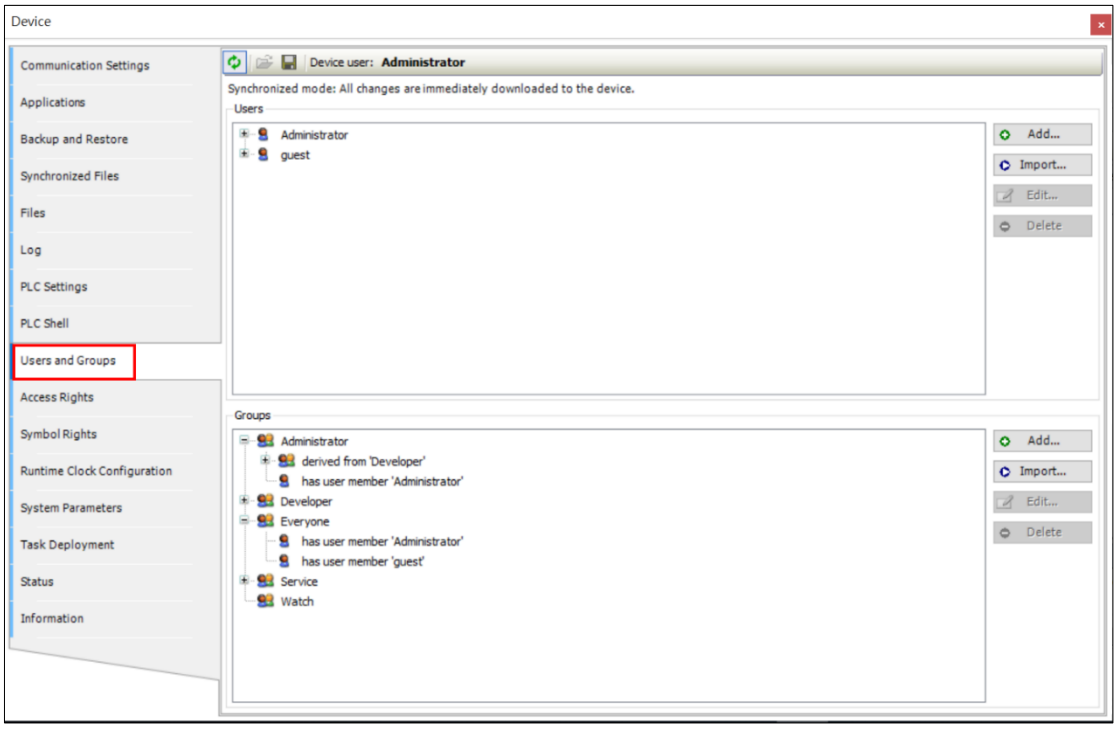


Figure 5 - 53: Set user account password screen

- **Trust Certificate Settings**

  Install the CODESYS Security Agent add-on. DIADesigner-AX is already installed.

1. Click *View* > *Security Screen*.

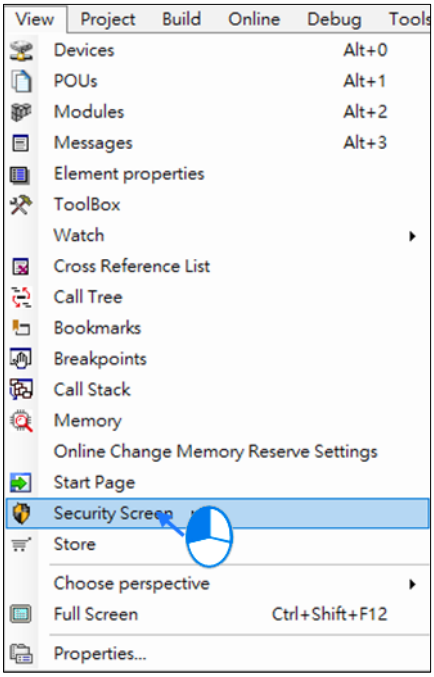


Figure 5 - 54: Trust certificate setting path

2. Select the *Devices* tab.

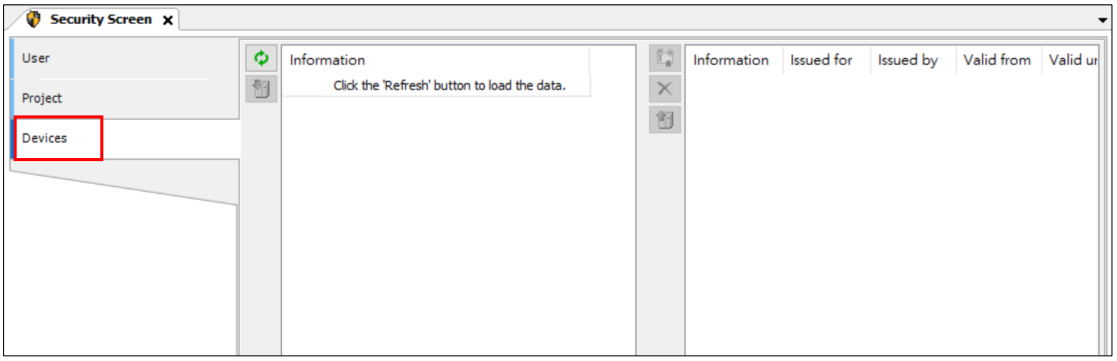


Figure 5 - 55: Safety fence Devices tab

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.

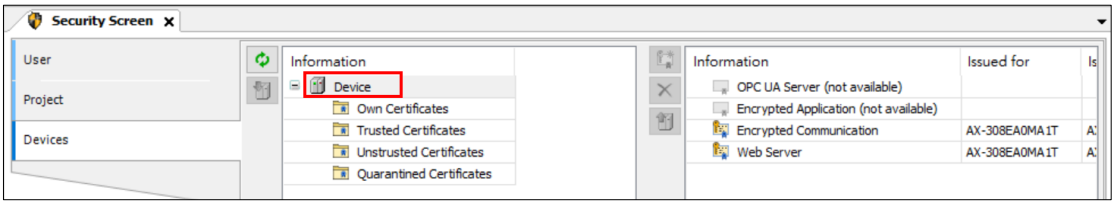


Figure 5 - 56: **Show all credentials**

4. Click to create a new certificate for the controller: make settings in the following *Certificate Settings* dialog box.

Select *OPC UA Server*.

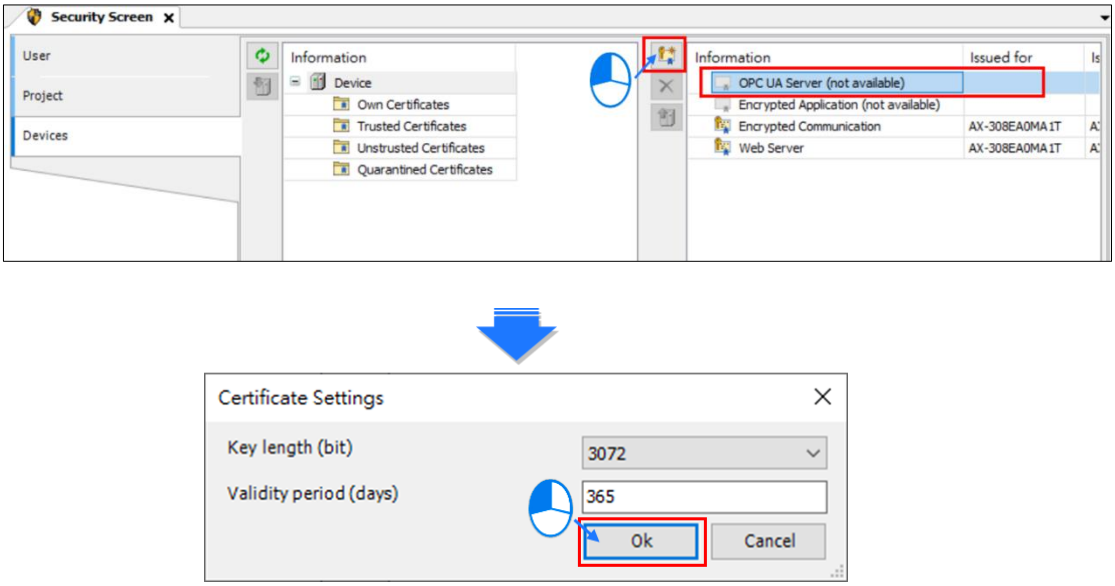


Figure 5 - 57: **Create a new certificate for the controller**

Select *Encrypted Application*

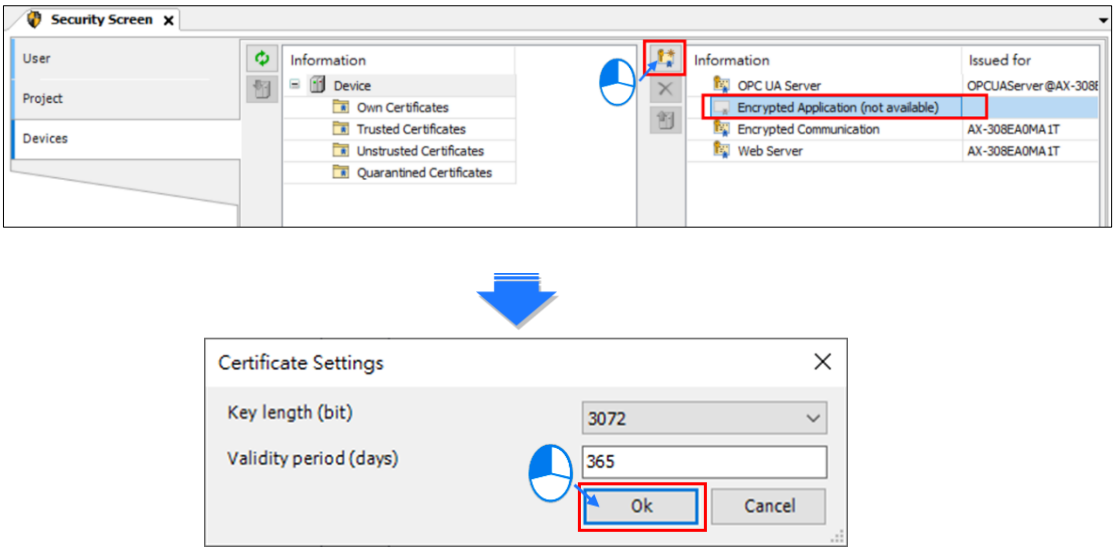


Figure 5 - 58: **Select Encrypted Application**

5. Complete the *OPC UA Server* and *Encrypted Application* certificate.

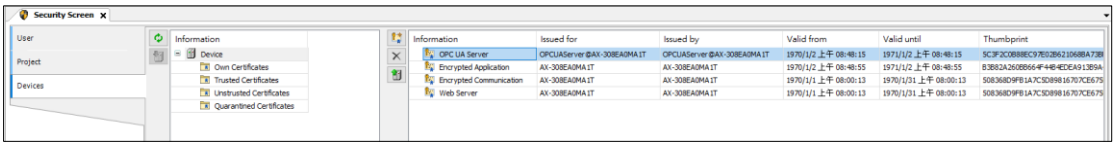


Figure 5 - 59: **Complete OPC UA Server and Encrypted Application certificate**

- **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*

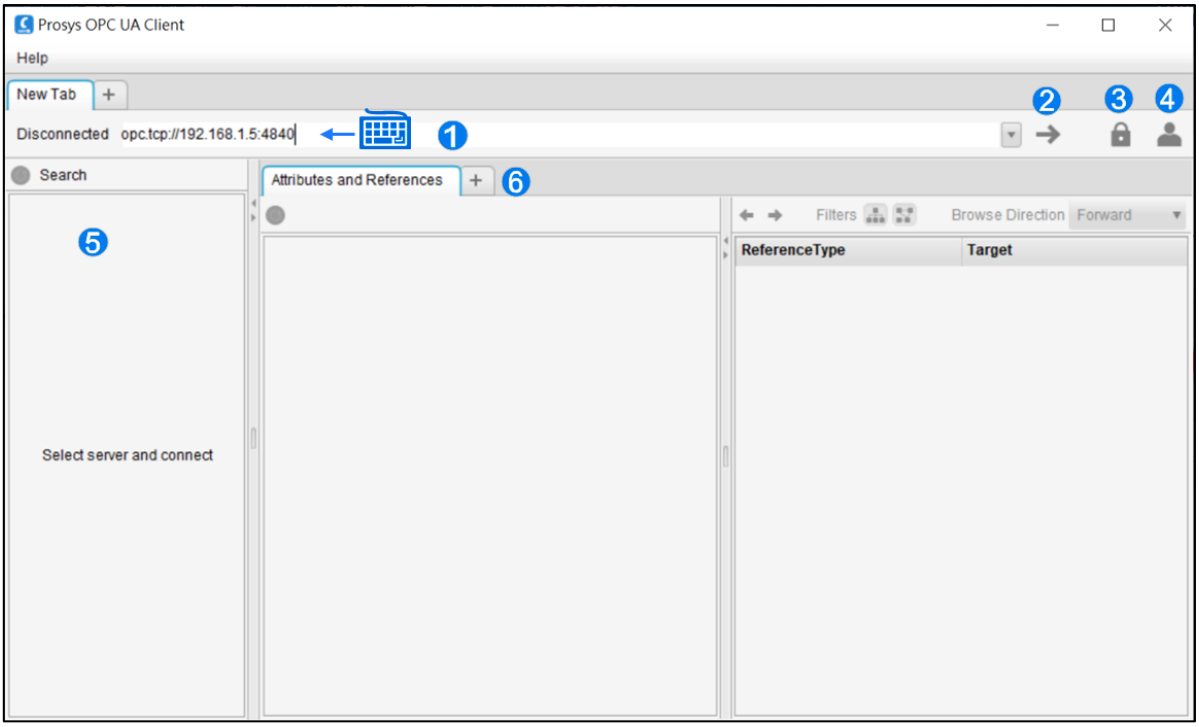


Figure 5 - 60: Enter the OPC UA Server IP address

3. Select the connection encryption method ③  rear, *Security Settings* Window (only supports Basic256SHA256) check *Show only modes that are supported by the server.*

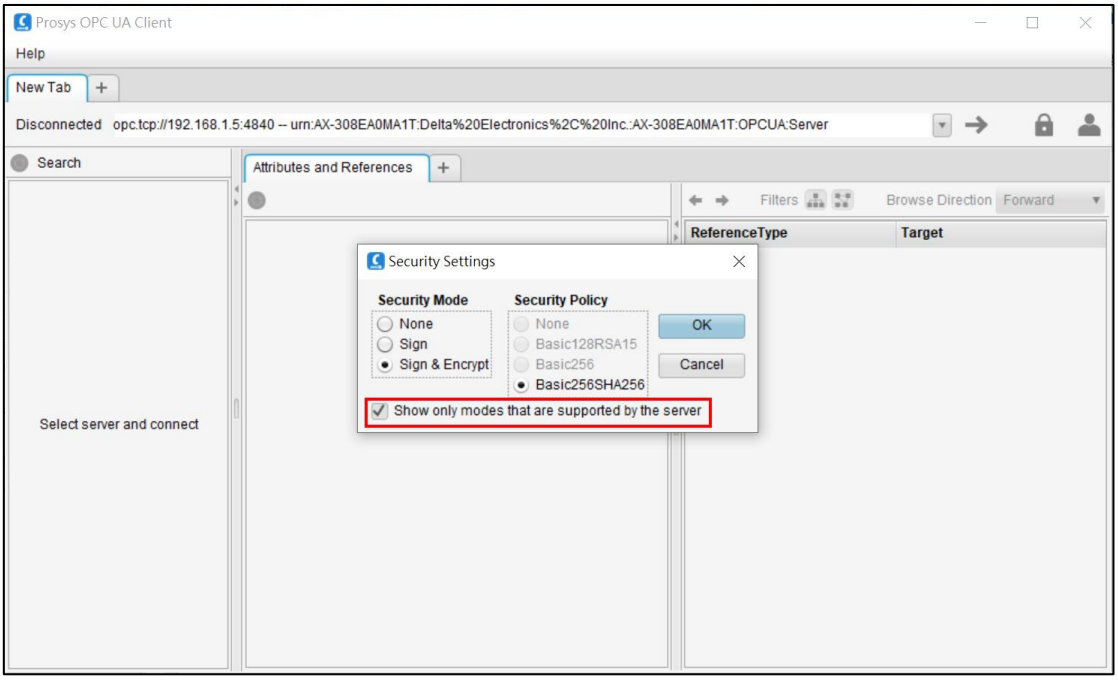


Figure 5 - 61: Security Settings Windows

4. Click on ④  After, set the *user account password* > *Apply,* After setting, click ②  Connect.

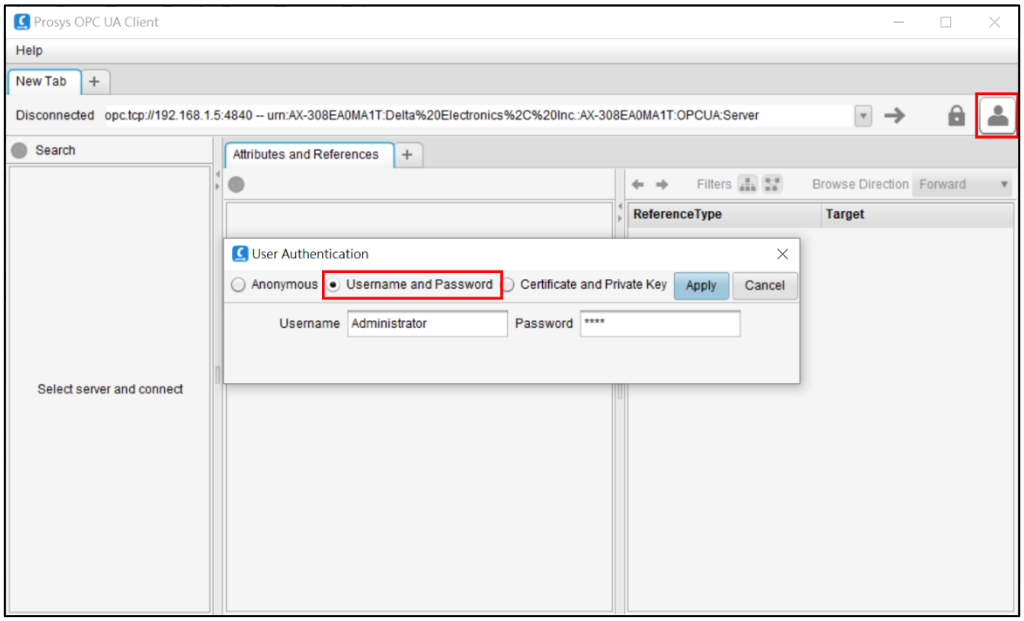


Figure 5 - 62: **Set user account password**

5. Pop up the prompt window: no Server Allowed App Certificate.

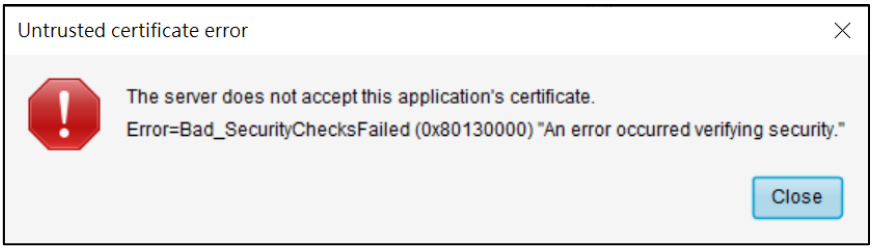


Figure 5 - 63: **Application certificate not allowed by server**

6. Back to the Security Screen page, there is an untrusted certificate ProsysOpcUaClient in Quarantined Certificates, drag this certificate to Trusted Certificates.

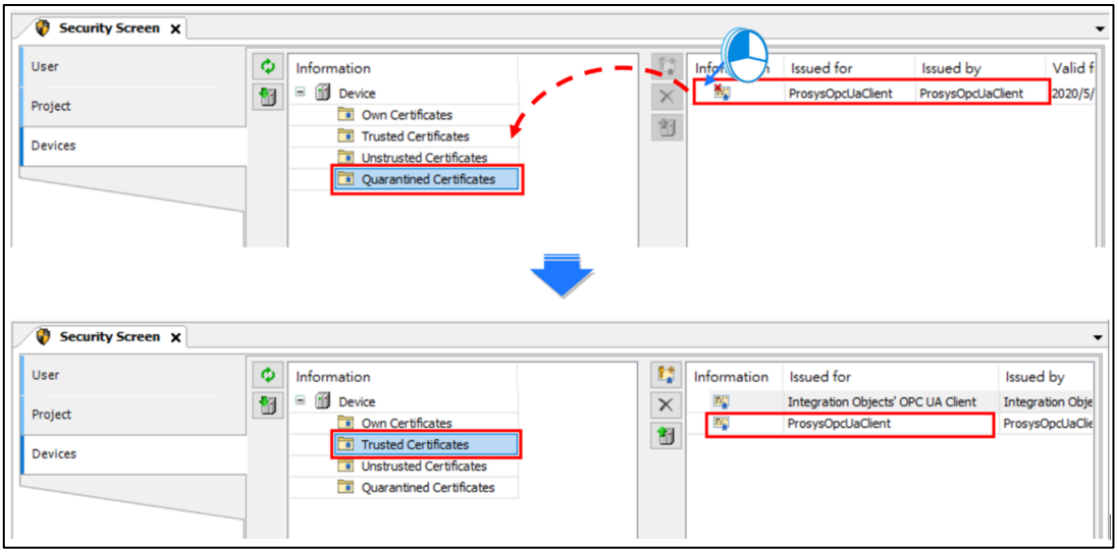


Figure 5 - 64: **Drag untrusted certificates to Trusted Certificates**

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

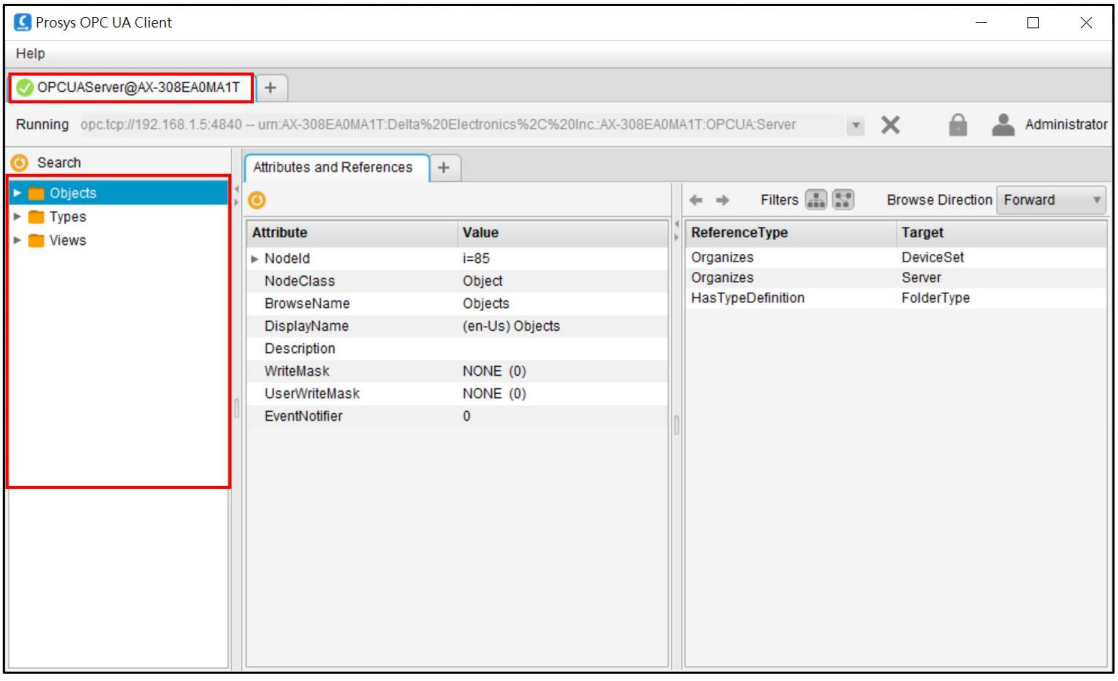


Figure 5 - 65: Administrator

User：guest（No write permission）

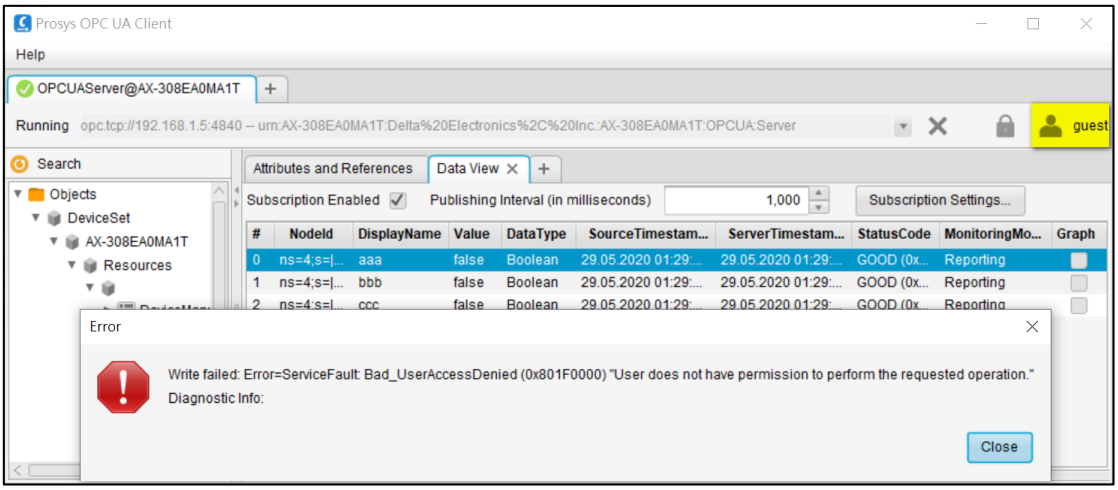


Figure 5 - 66: Guest

Chapter 6: HMI Interactive

6.1 Overview

DIADesigner-AX supports export of variables to DIAScreen. User can define the variables in DIADesigner-AX and export the xml file of *Symbol configuration.* Import the xml to DIAScreen to communicate between controller and Delta HMI functions.

***NOTE**: HMI Tag sharing is supported only for DOP-100 series.*

6.2 Export Variable

1. Open *DIADesigner-AX* > *File* > *New project.*

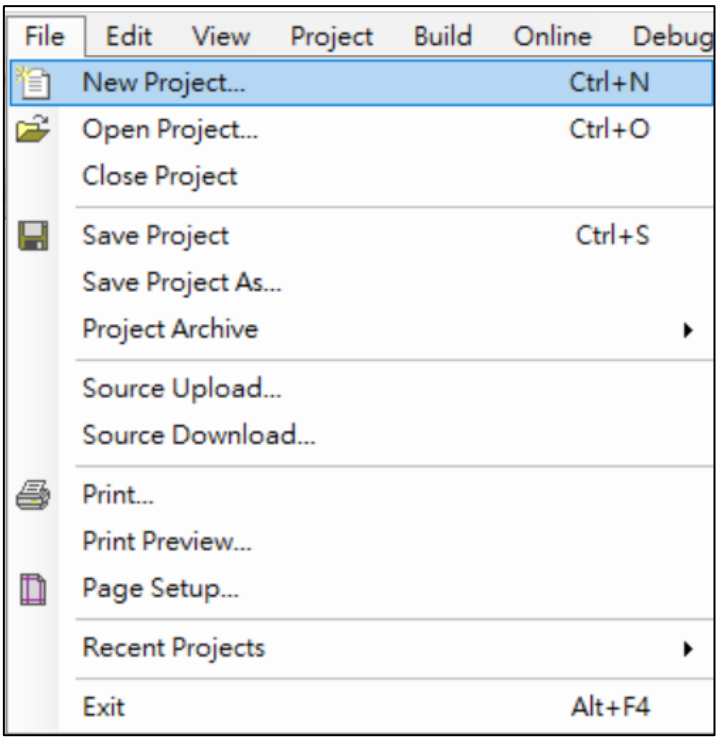


Figure 6 - 1: New project

2. Right-click on *Application > Add device >* Select *Global variable list.*

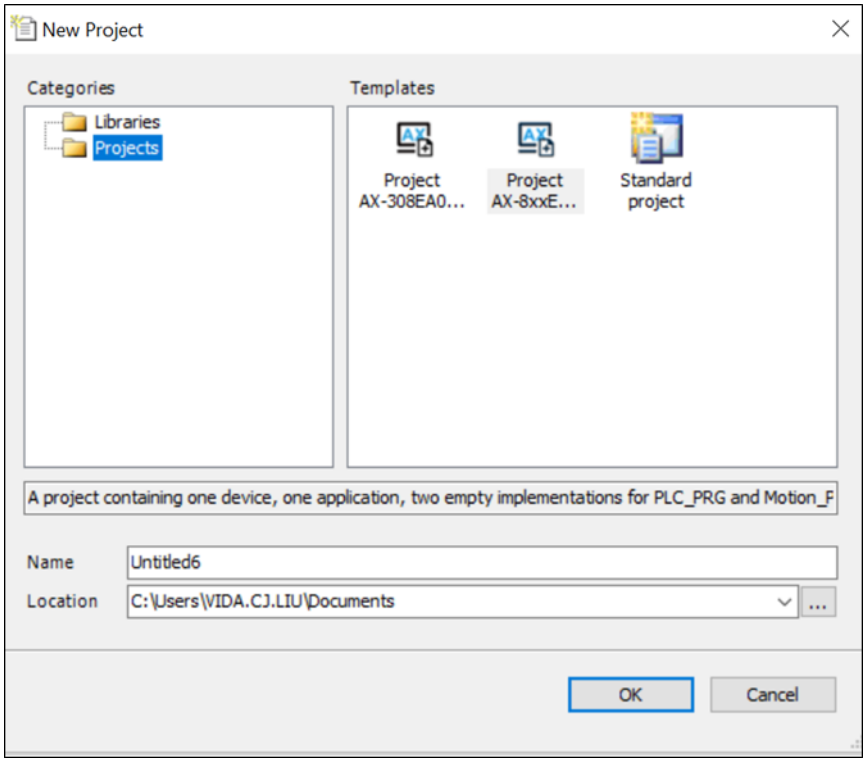


Figure 6 - 2: Creating New Project

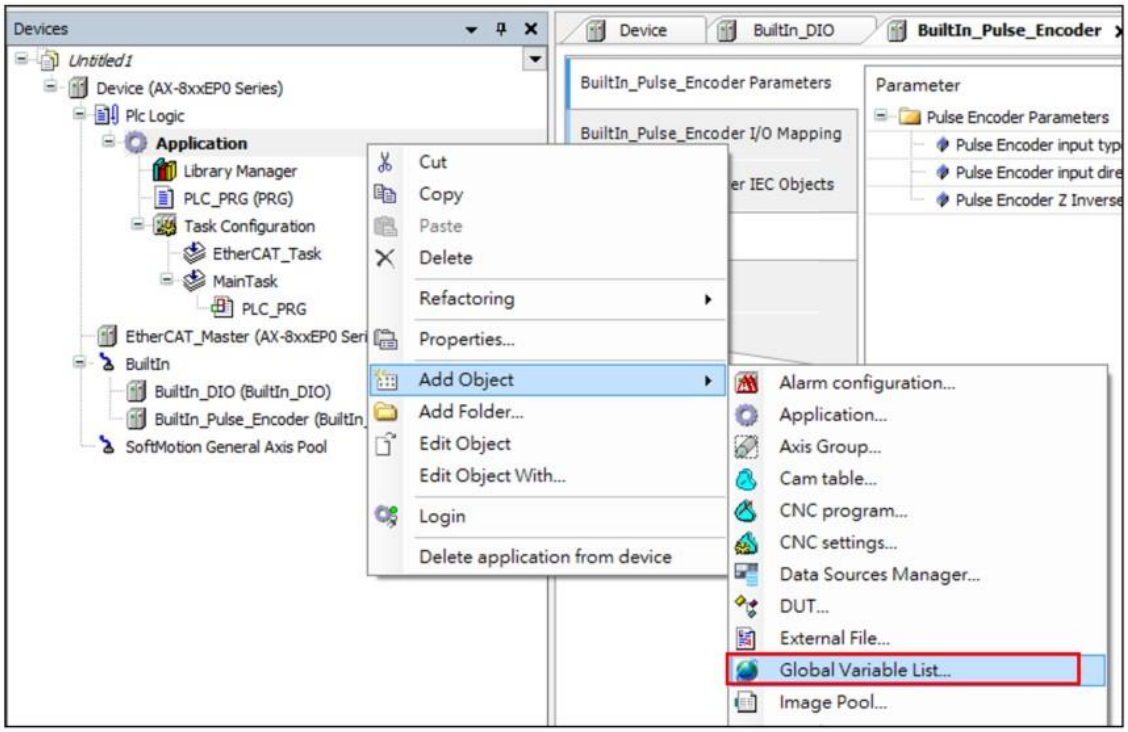


Figure 6 - 3: Adding Global Variable List

Result: *Add Global Variable List* displays.

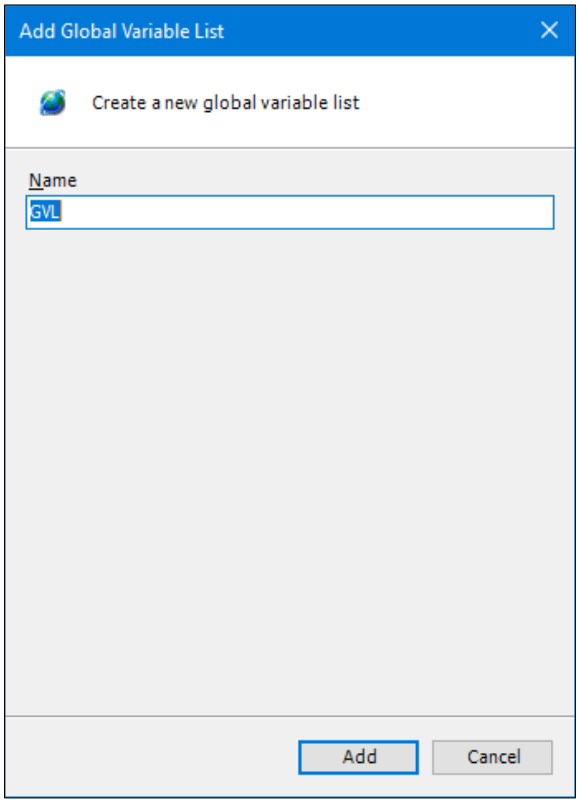


Figure 6 - 4: Creating New Global Variable List

3. Enter the required name and click *Add.*

   Result: *GVL* is added in *Application*.

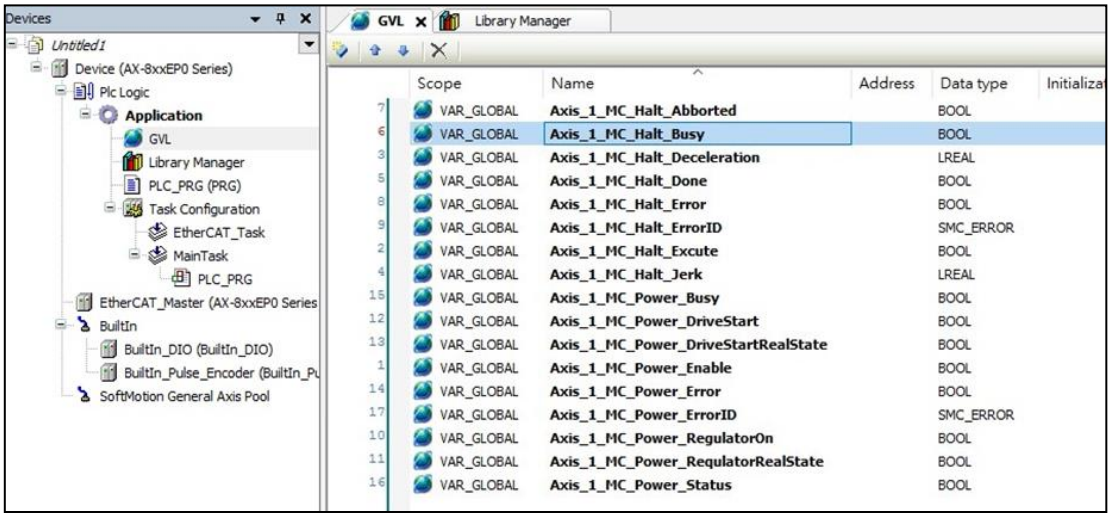


Figure 6 - 5: GVL added to Application

4. Right-click on *Application* and select *Add Object > Symbol Configuration.*

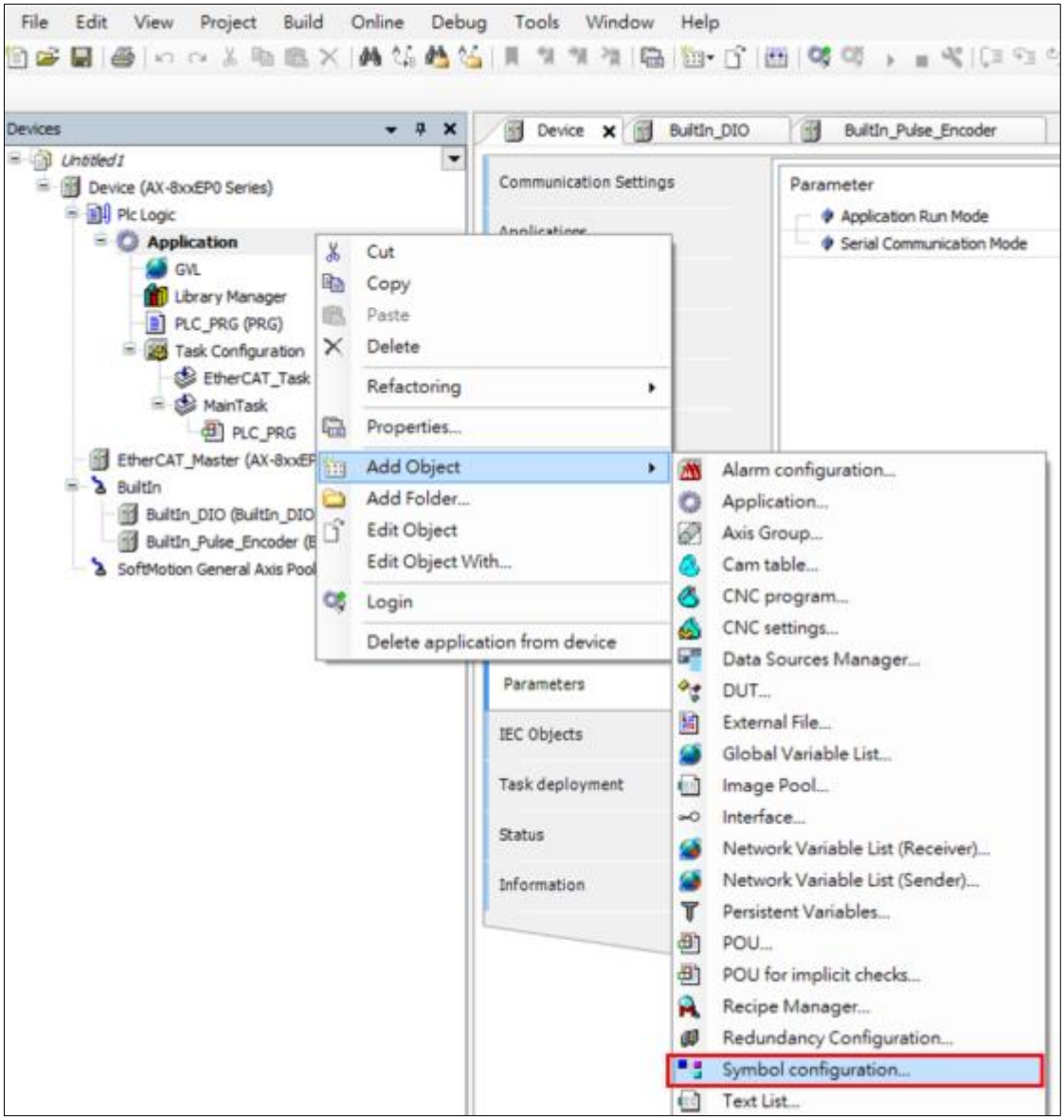


Figure 6 - 6: Symbol Configuration Window

Result: The *Add Symbol Configuration* window displays.

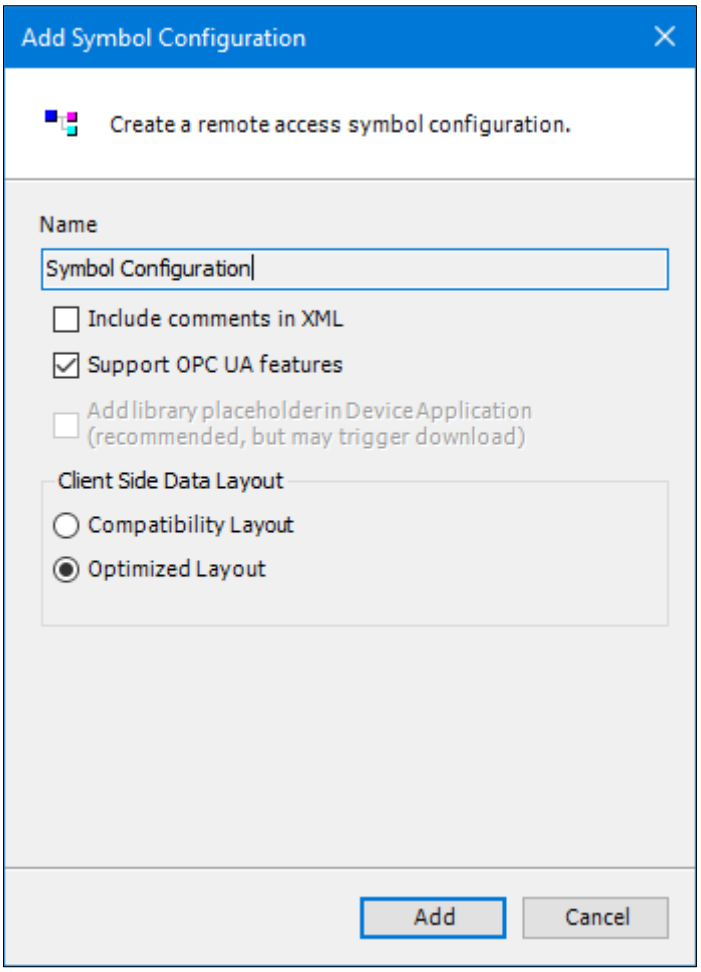


Figure 6 - 7: Adding Symbol Configuration

5. Select *Include comments in XML* and click *Add.*

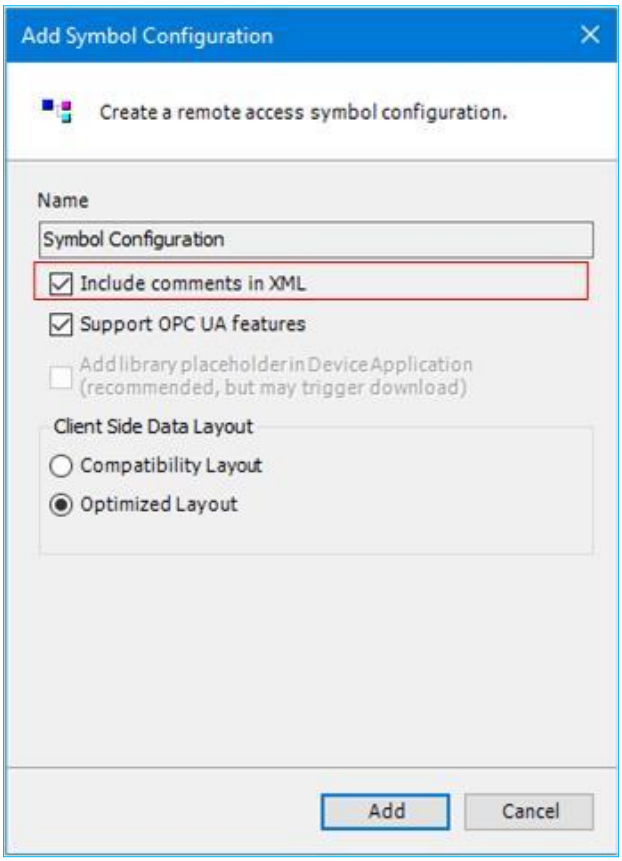


Figure 6 - 8: Adding XML

Result: A Symbol configuration is added to the application.

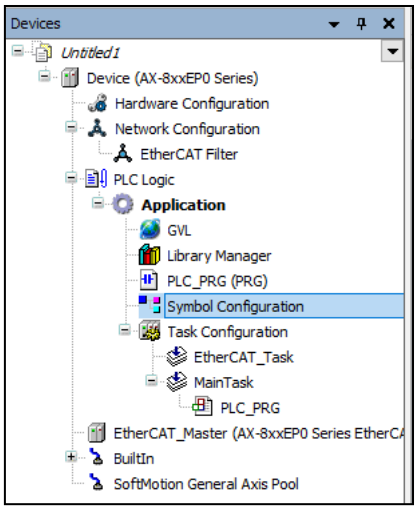


Figure 6 - 9: Adding Symbol Configuration to Application

6. Click *Build* in *Menu bar* or

Press *F11* and select the respective *GVL variable* in the symbol list.

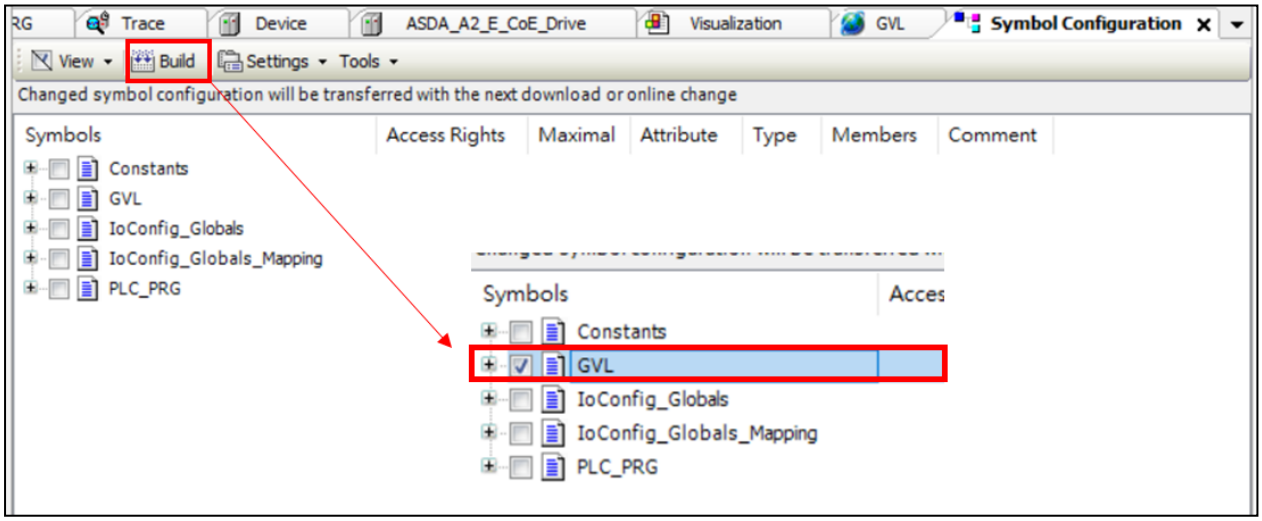


Figure 6 - 10: Build

7. Click *Build* in Toolbar > *Generate code.*

   Result: It generates xml file in the assigned folder.

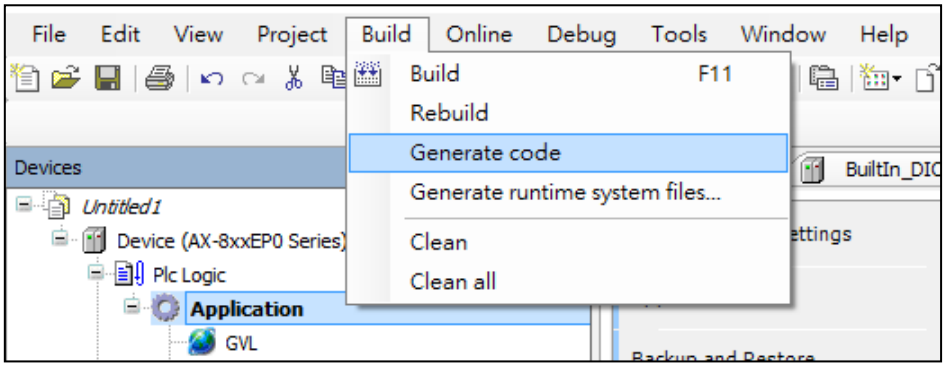


Figure 6 - 11: Generating Code

8. The XML file will be stored in the designated folder.

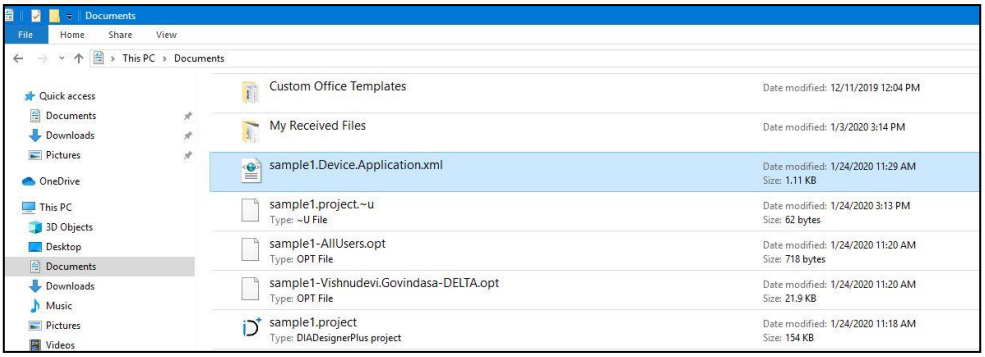


Figure 6 - 12: Sample Output

6.3 DIAScreen with PAC AX-8 Series

1. Launch DIAScreen.

   Result: DIAScreen displays the home screen as shown in the following figure.

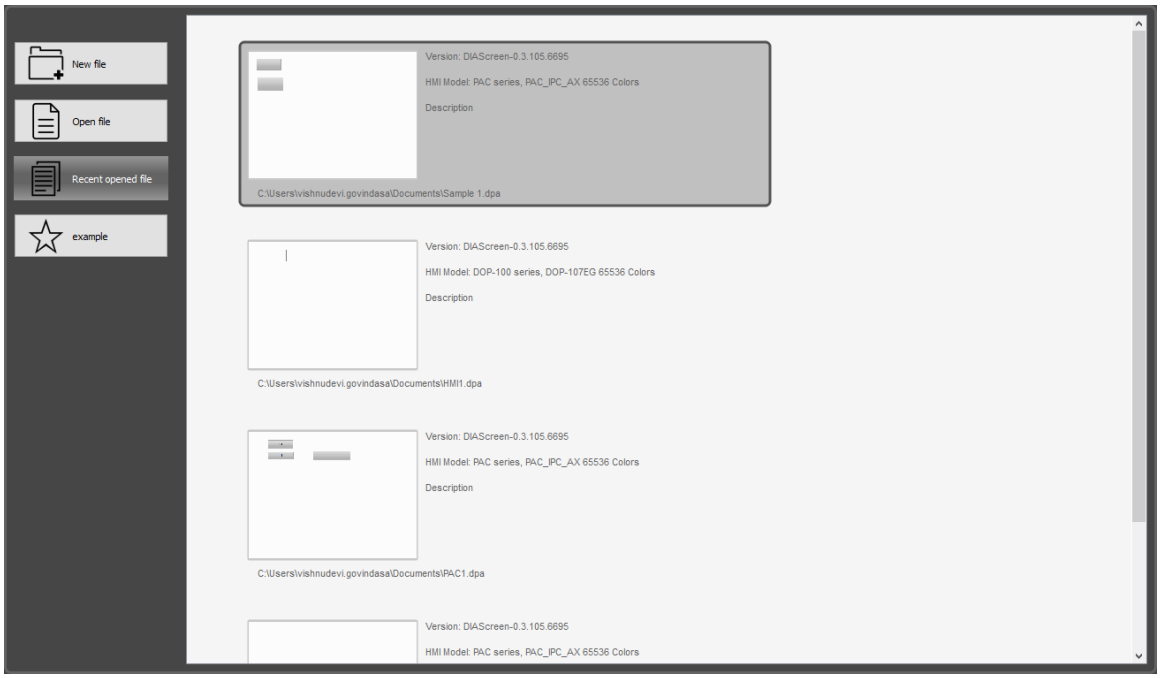


Figure 6 - 13: DIAScreen Window

2. Click *New file.*

   Result: A *Project Wizard* window displays as shown in the following figure.

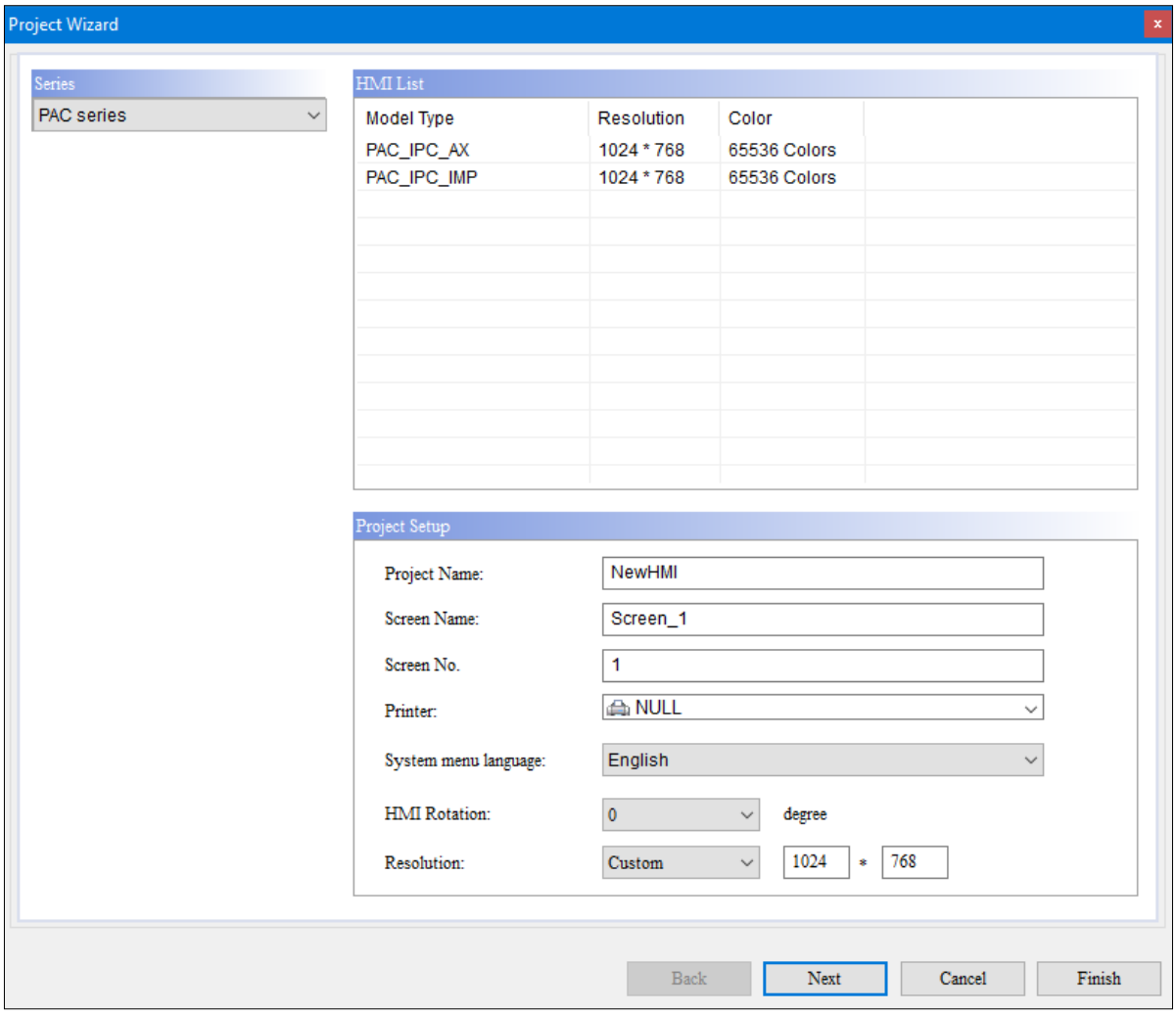


Figure 6 - 14: Project Wizard

3. Select *PAC series* in *Series* and *PAC_IPC_AX Model* in *Model Type.*
4. Enter *Project Name*, *Screen Name*, *Screen No.*, *Printer*, *System menu language*, *HMI Rotation* and *Resolution* in the *Project Setup* box.Click *Next.*

   Result: A *Project Wizard* window displays as shown in the following figure.

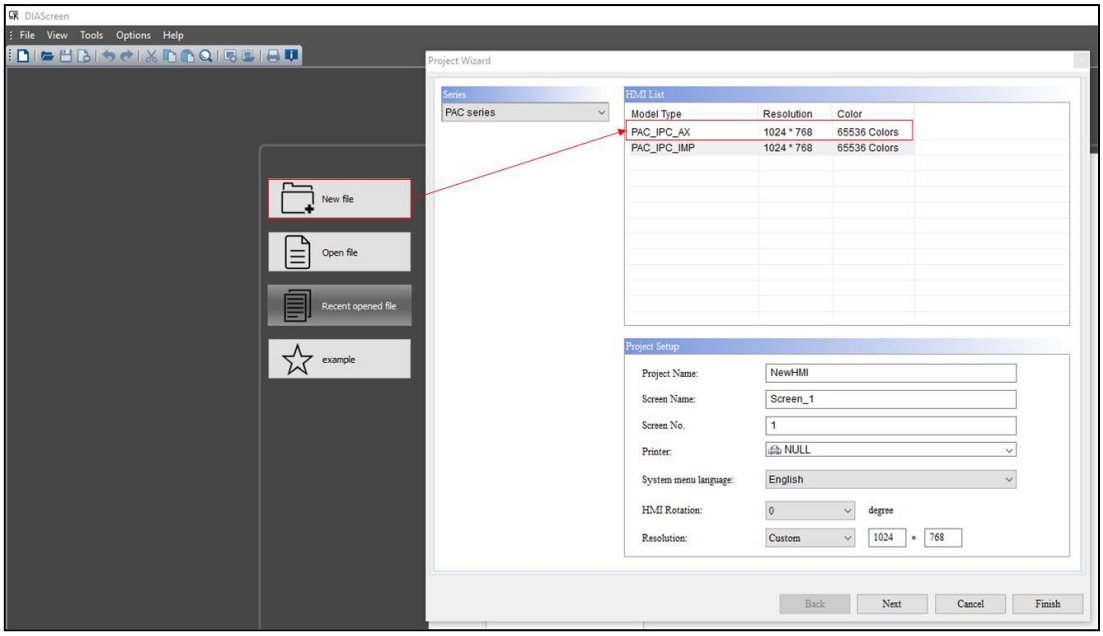


Figure 6 - 15: Creating New Project

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.*

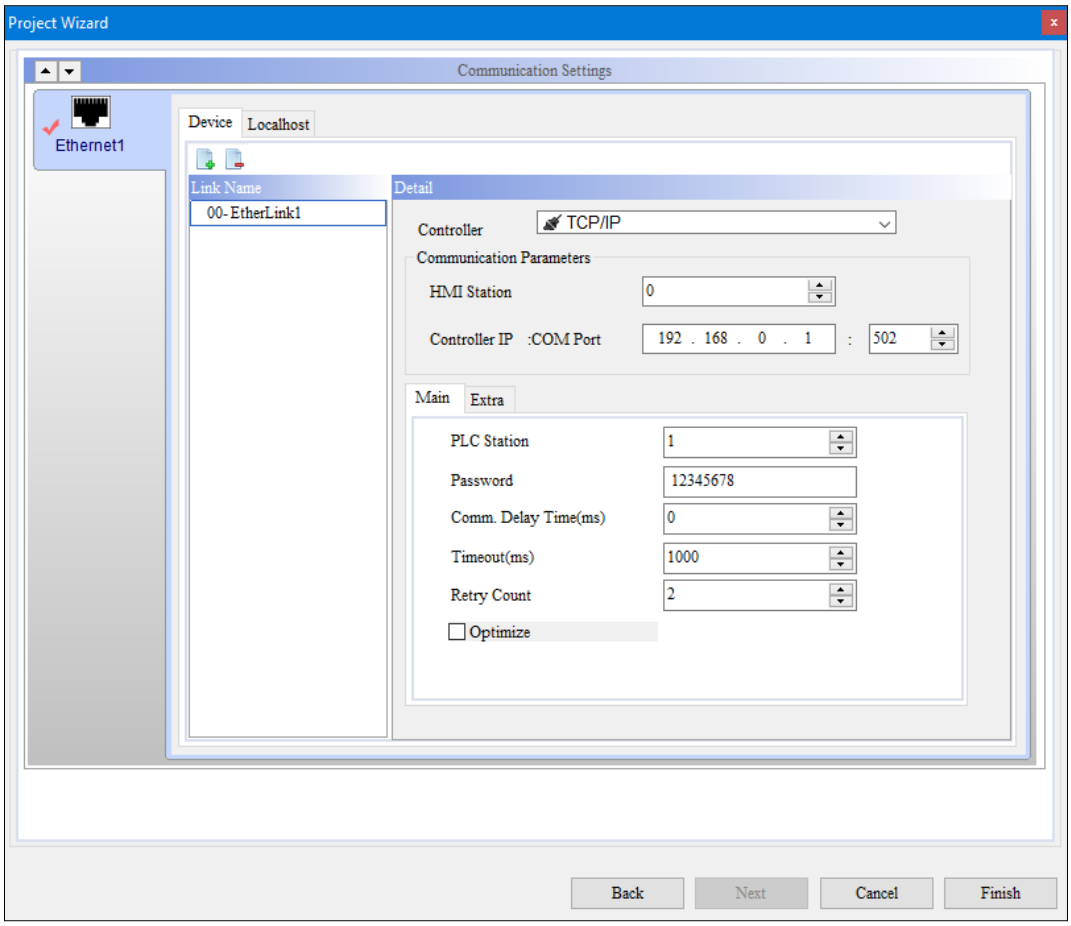


Figure 6 - 16: Communication Setting

6. Click *New HMI* in the screen.

   Result: User can view *CODESYS* in the project tree.

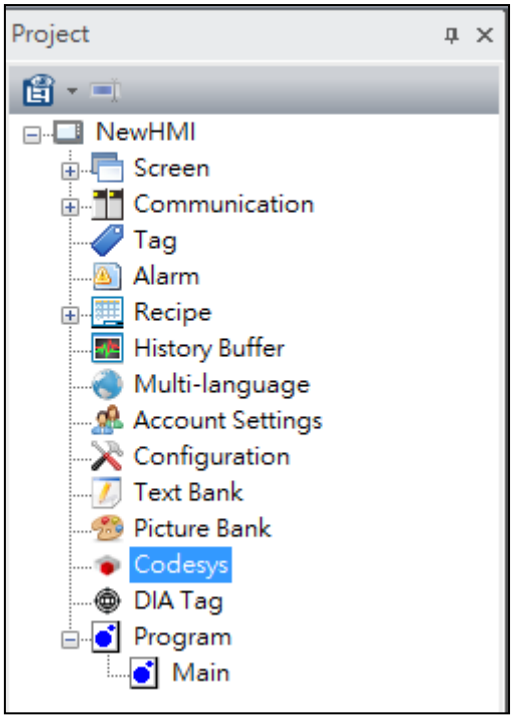


Figure 6 - 17: Project Tree

7. Click on *CODESYS* > *Import icon* > select the respective *xml file* > *open*.

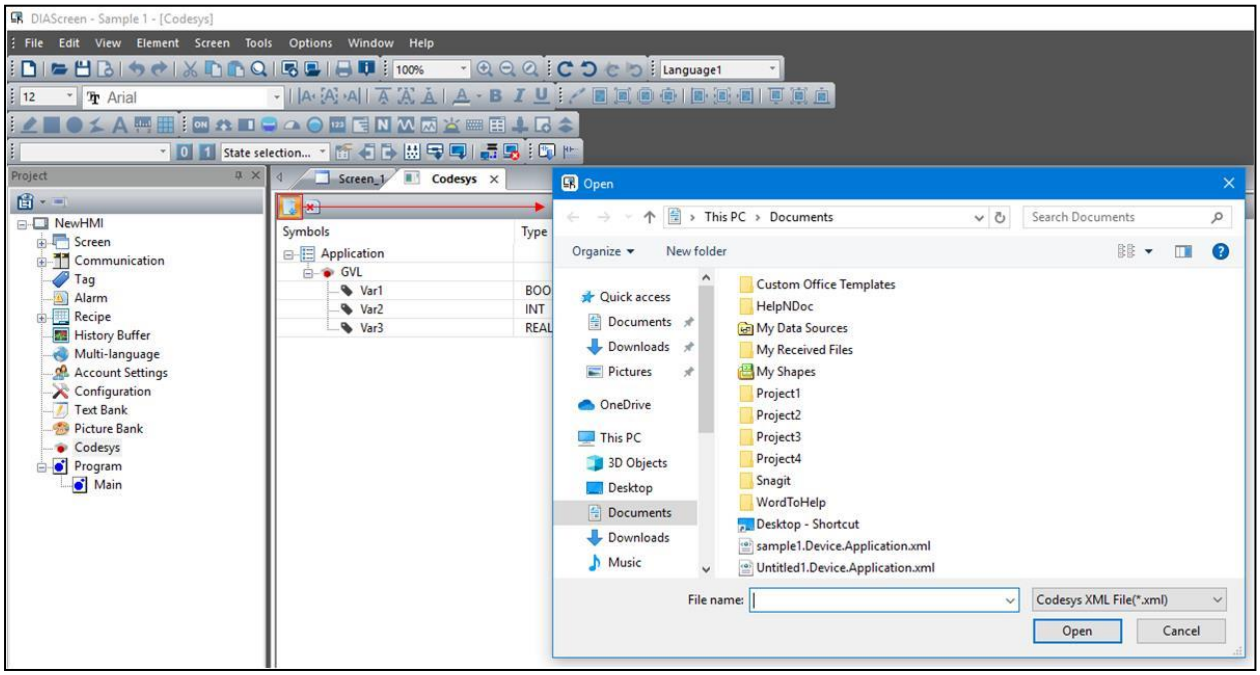


Figure 6 - 18: Importing XML file

Result: The Imported symbols appears on DIAScreen.

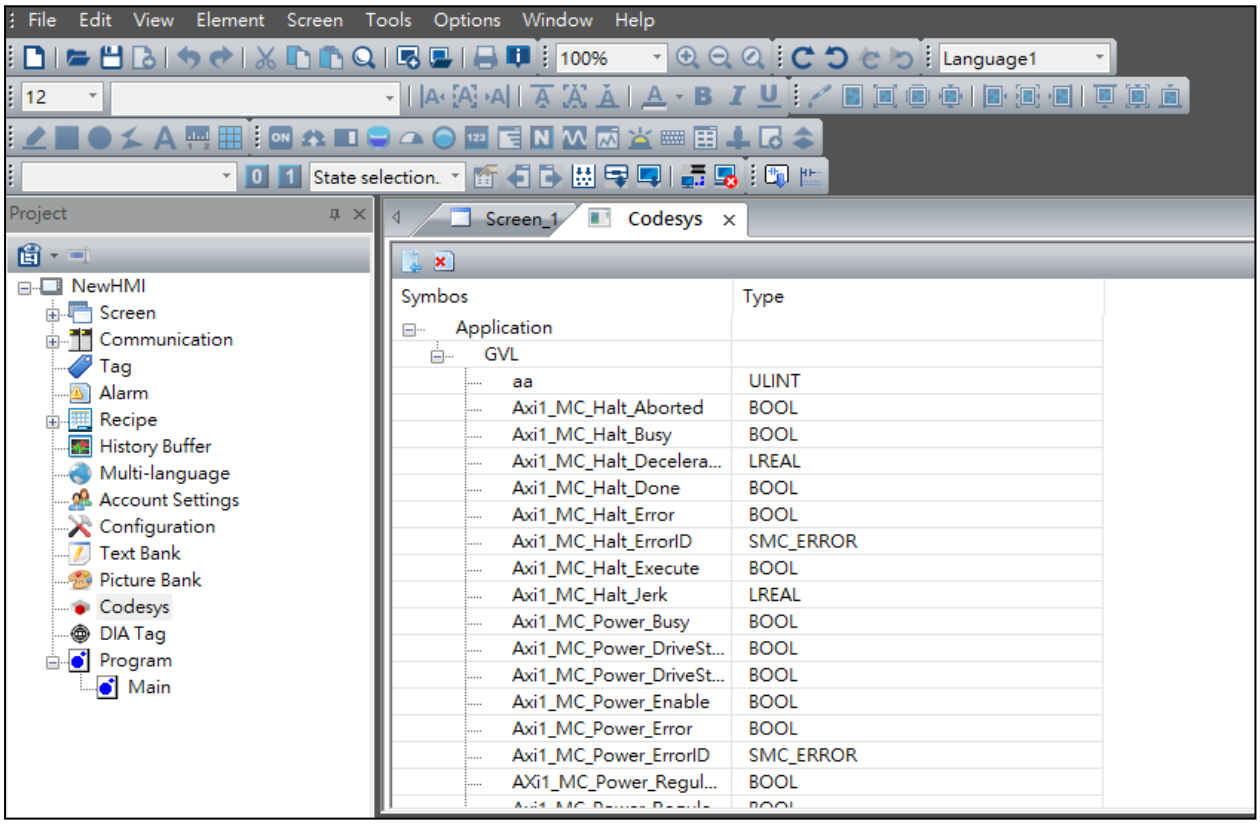


Figure 6 - 19: GVL added to Application

8. Use any of the buttons on the element and double-click the element.

   Example: *Set to on* button shown in the following figure.

   ***NOTE**: User can choose the CODESYS parameter to connect with the element.*

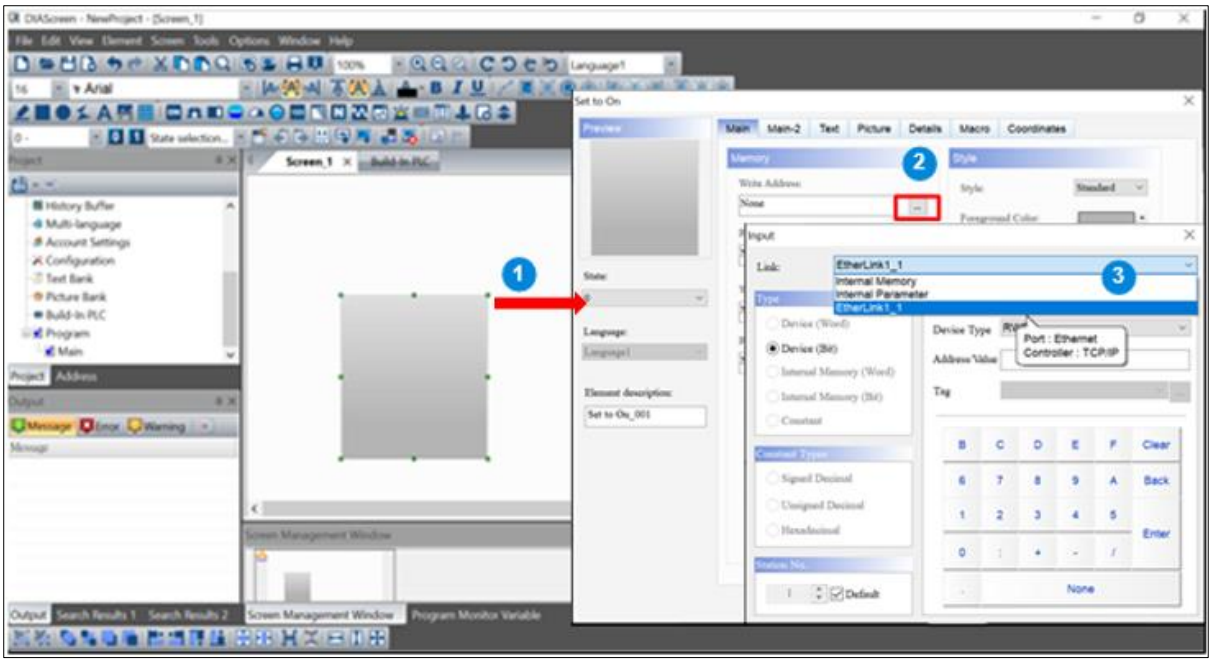


Figure 6 - 20: Creating Set to On Button

9. Select *CODESYS* in Link.

   Result: CODESYS is linked with the element.

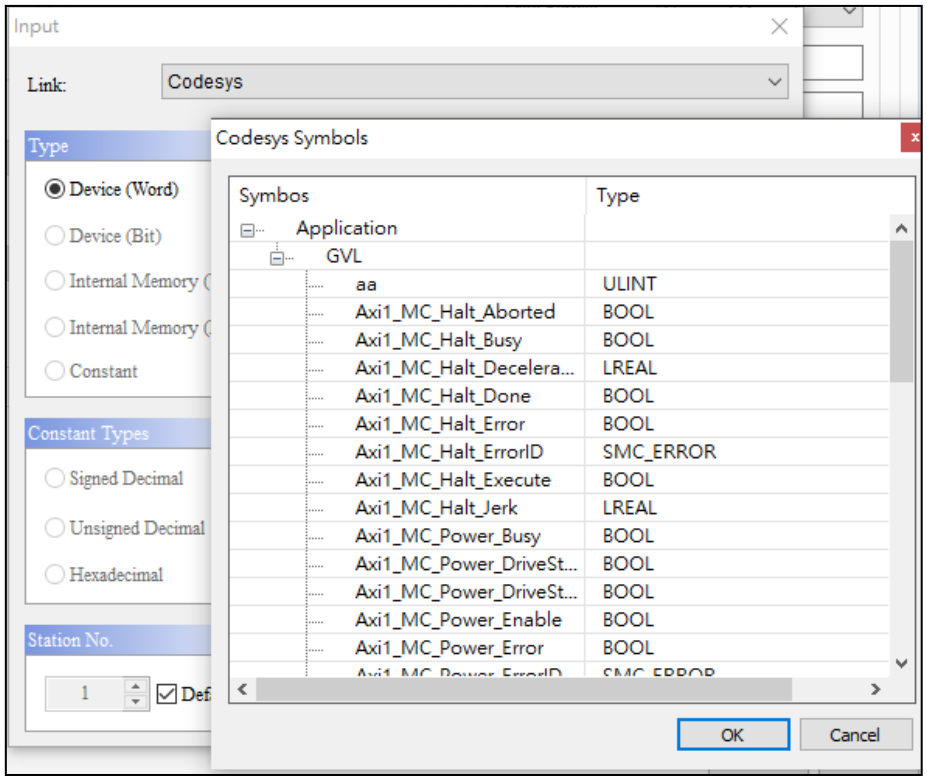


Figure 6 - 21: Linking CODESYS

10. After configuration screen, click *Download* and choose the model to download the screen.

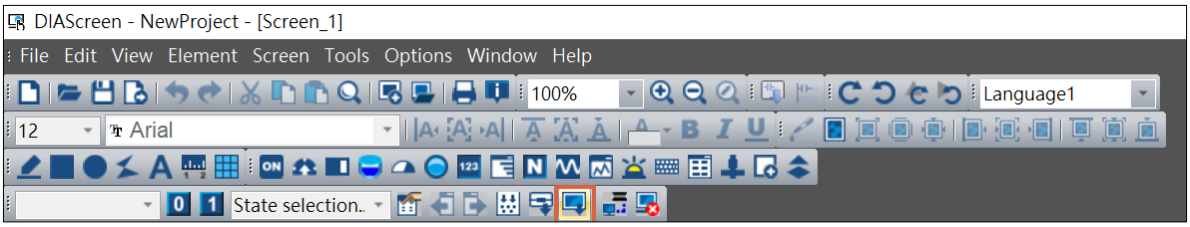


Figure 6 - 22: Download

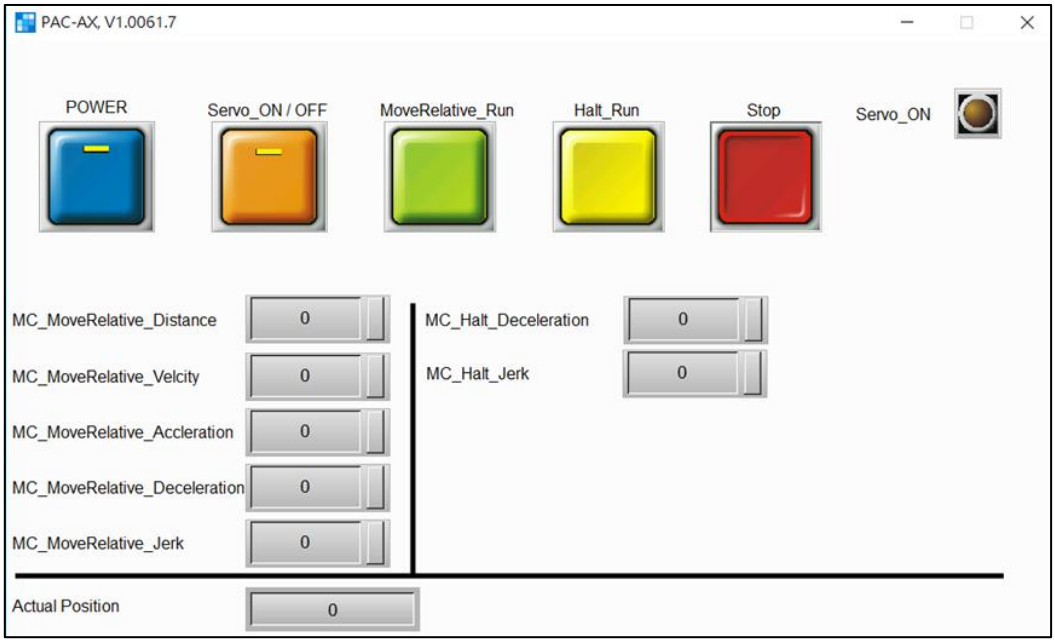


Figure 6 - 23: Downloading Application

6.4 DIAScreen with PLC AX-308E Series

1. Launch DIAScreen.

   Result: DIAScreen displays the home screen as shown in the following figure.

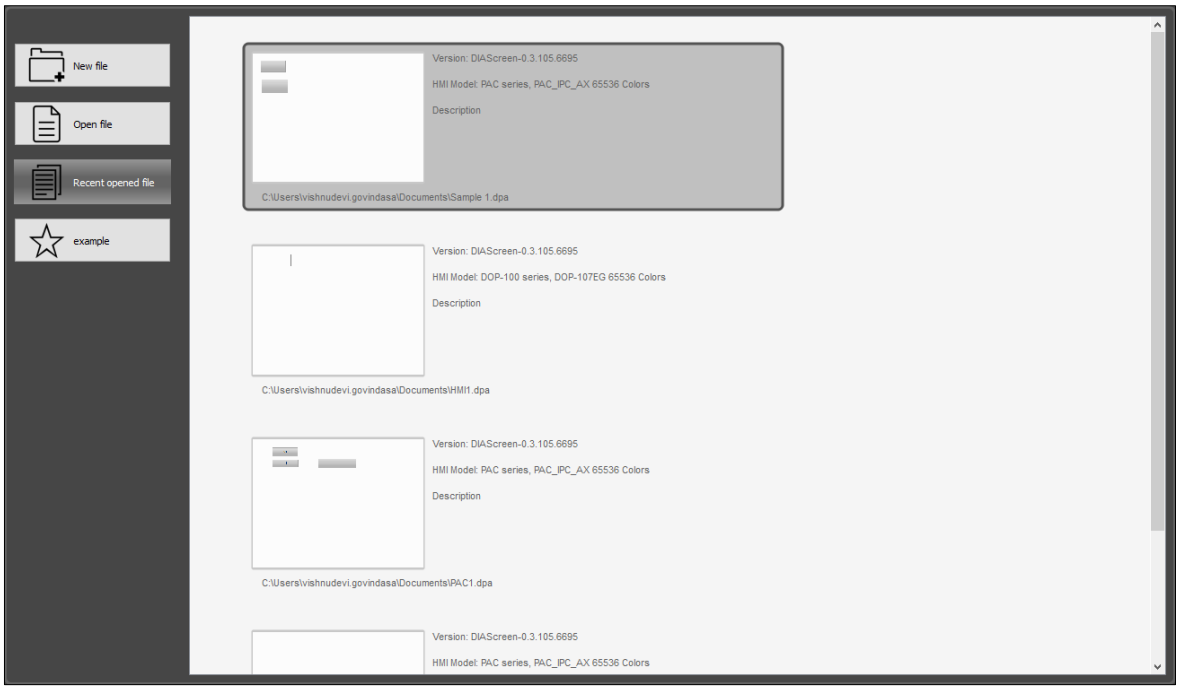


Figure 6 - 24: DIAScreen

2. Click *New file* in the screen.

   Result: A *Project Wizard* window displays as shown in the following figure.

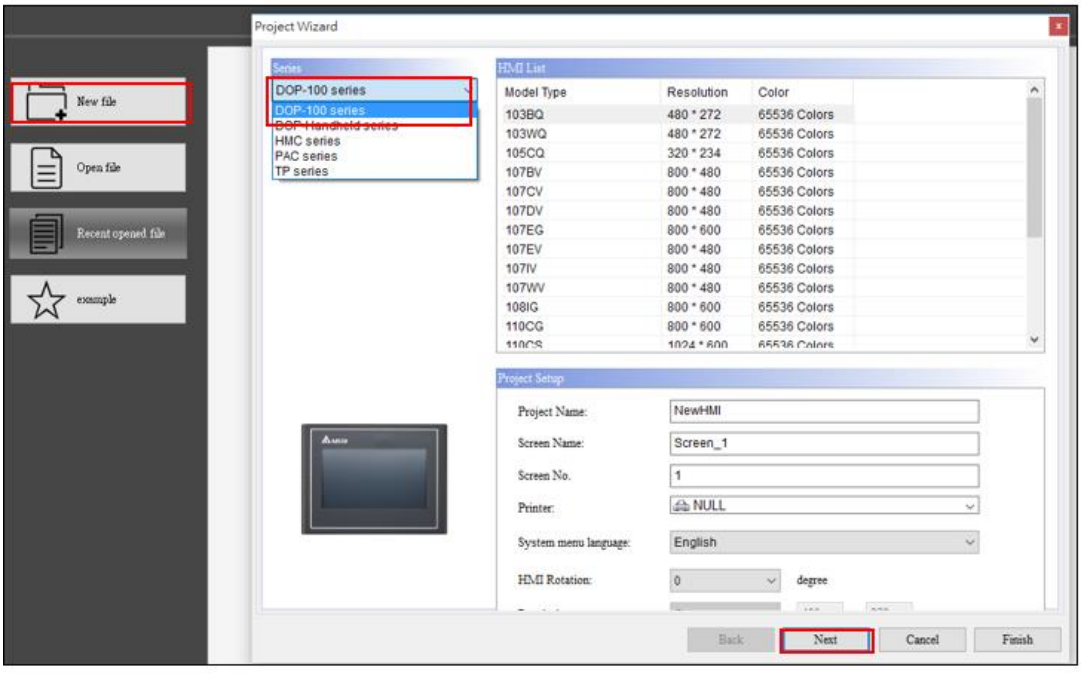


Figure 6 - 25: Creating New file

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.

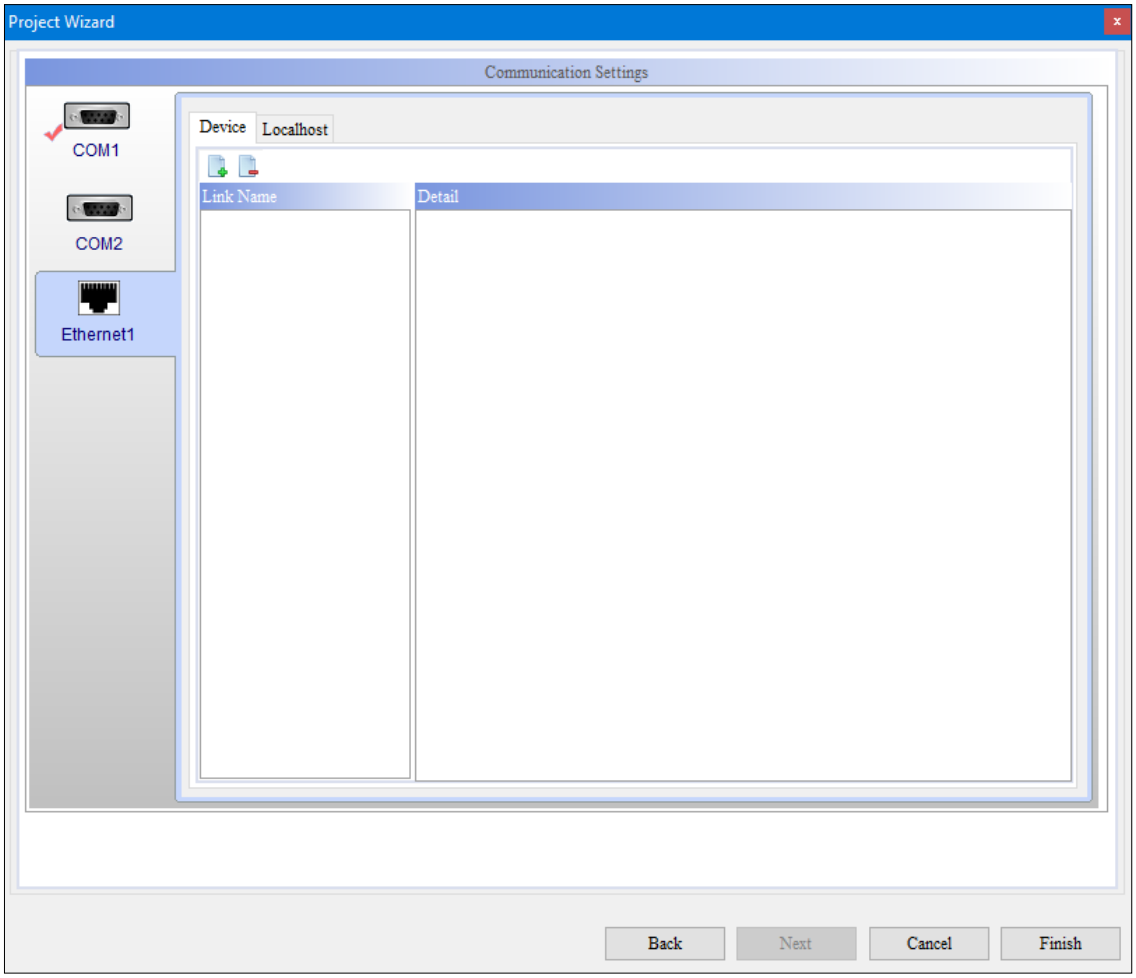


Figure 6 - 26: Communication setting

4. Click *Ethernet1* > *icon.* Select *CODESYS* in controller. Click *Import.*

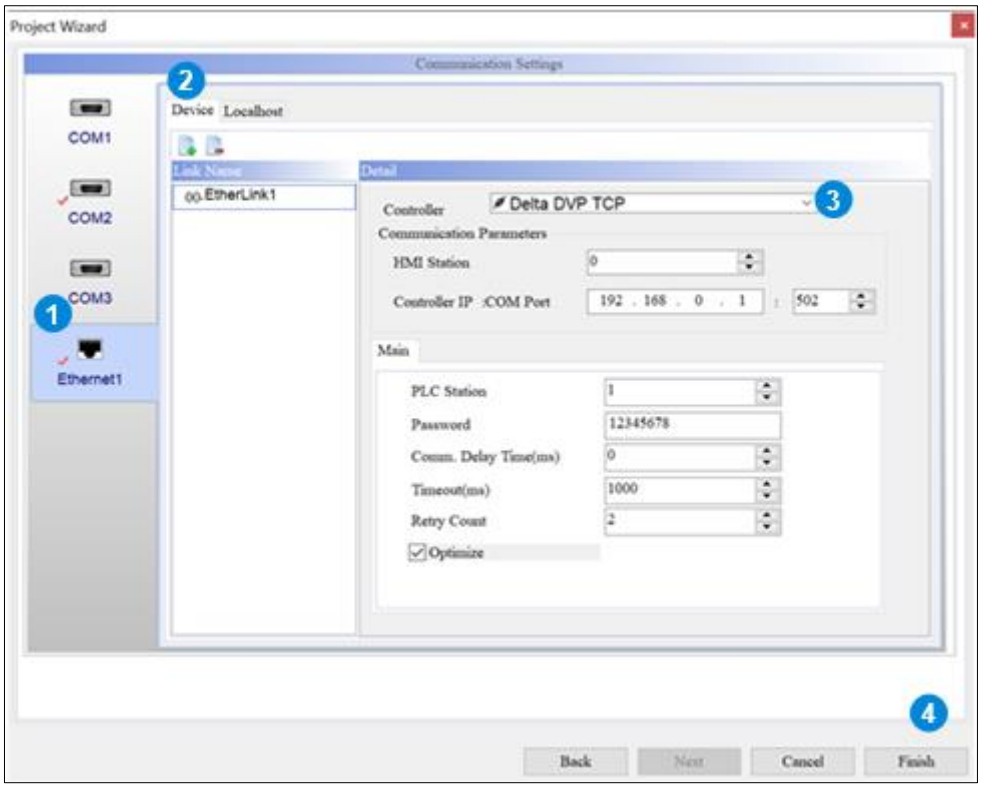


Figure 6 - 27: Selecting CODESYS

5. Click *Controller* drop-down menu and click *Open.*

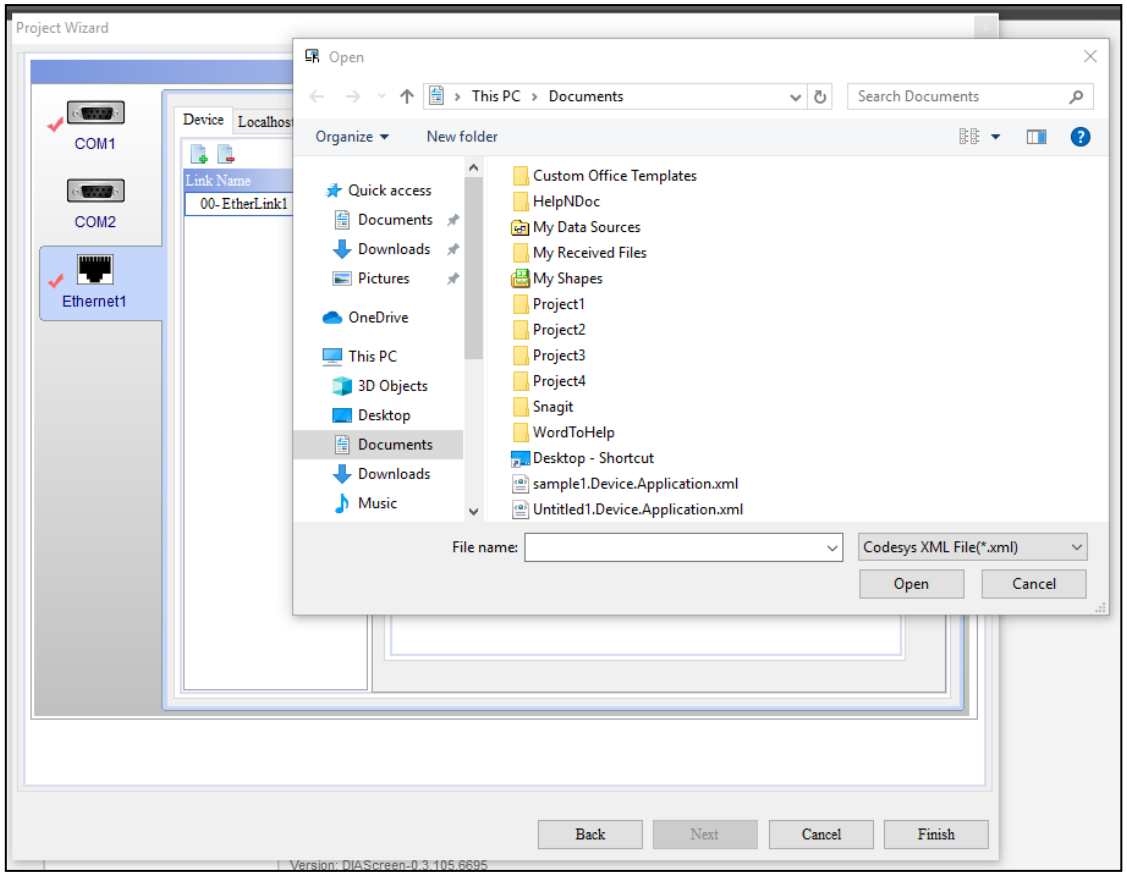


Figure 6 - 28: Importing CODESYS

6. Use any button to set the input.

   Example: *Set to On* is used in the following figure. Enter the required details.

   Click *OK.*

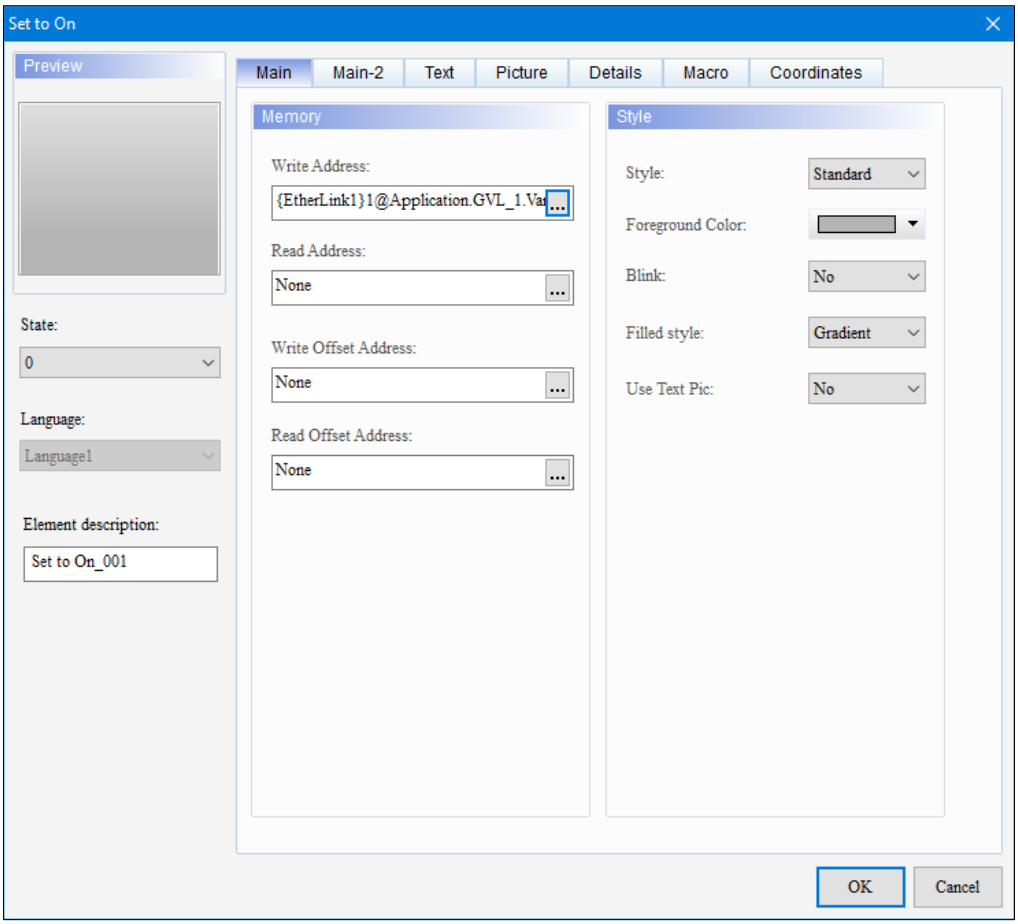


Figure 6 - 29: Set To On Button

Result: CODESYS is linked with the element.

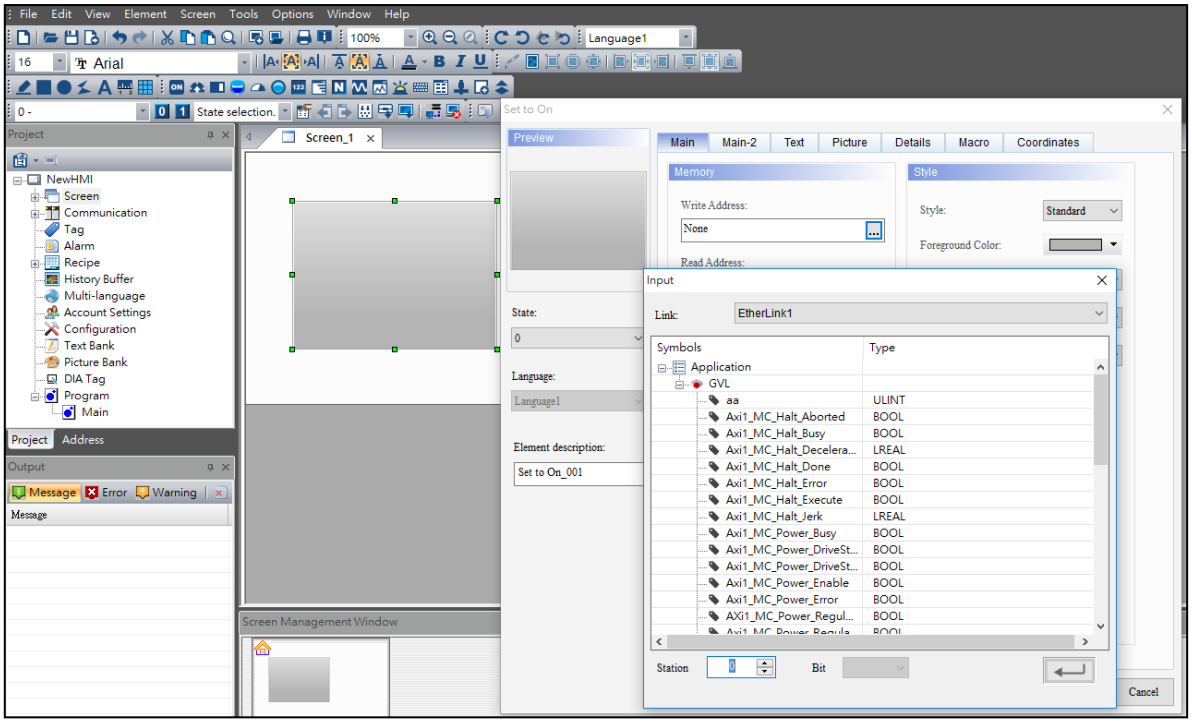


Figure 6 - 30: Linked CODESYS

7. After configuration screen, click *Download* and choose the model to download the screen.

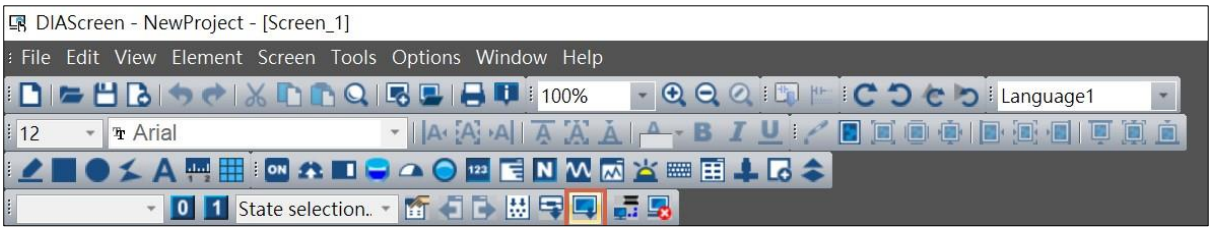


Figure 6 - 31: Download

Figure 6 - 32: Download Screen

6.5 Upload 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

6.5.1 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.

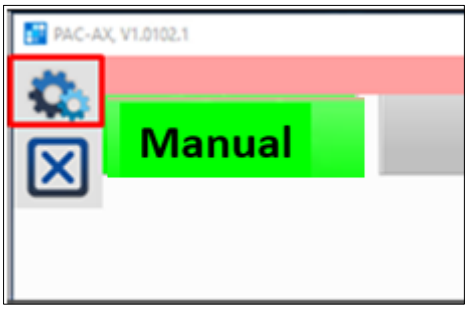


Figure 6 - 33: Enter the system screen

2. Click on *Upload/Download*.

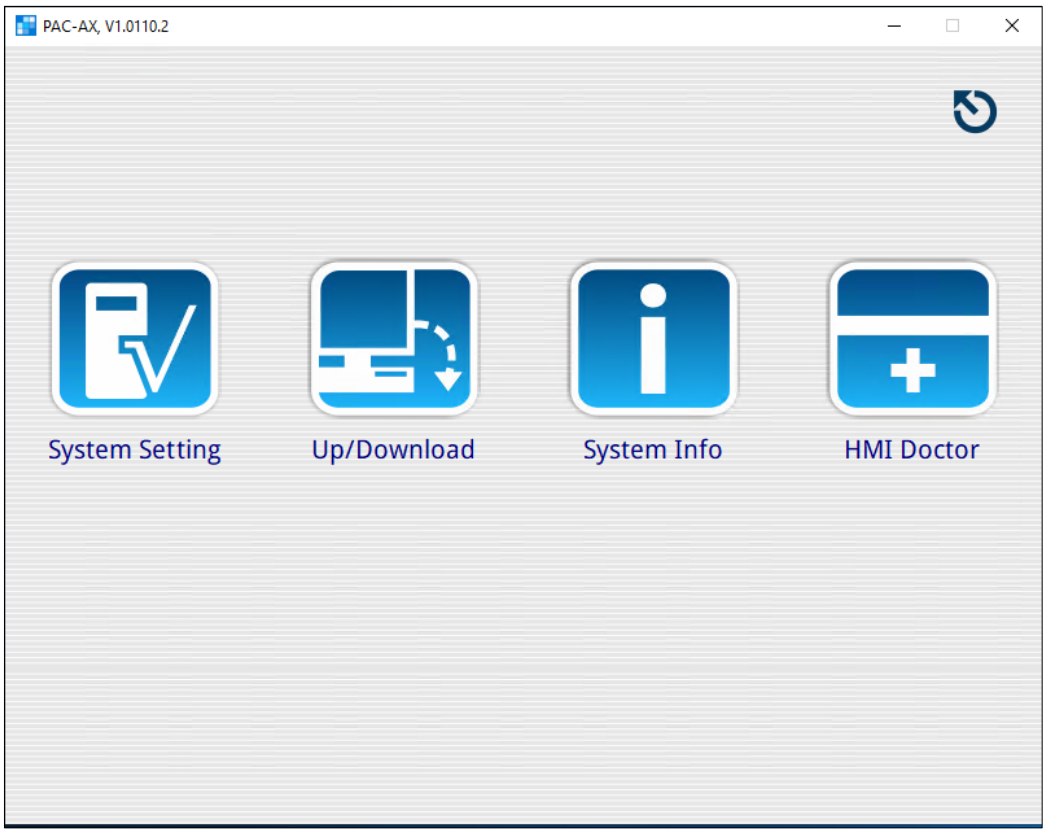


Figure 6 - 34: Upload and download

3. Click on *Transfer Mode*: This item is for the Application file in the PLC used. CODESYS Application file only supports *Transfer Mode*.

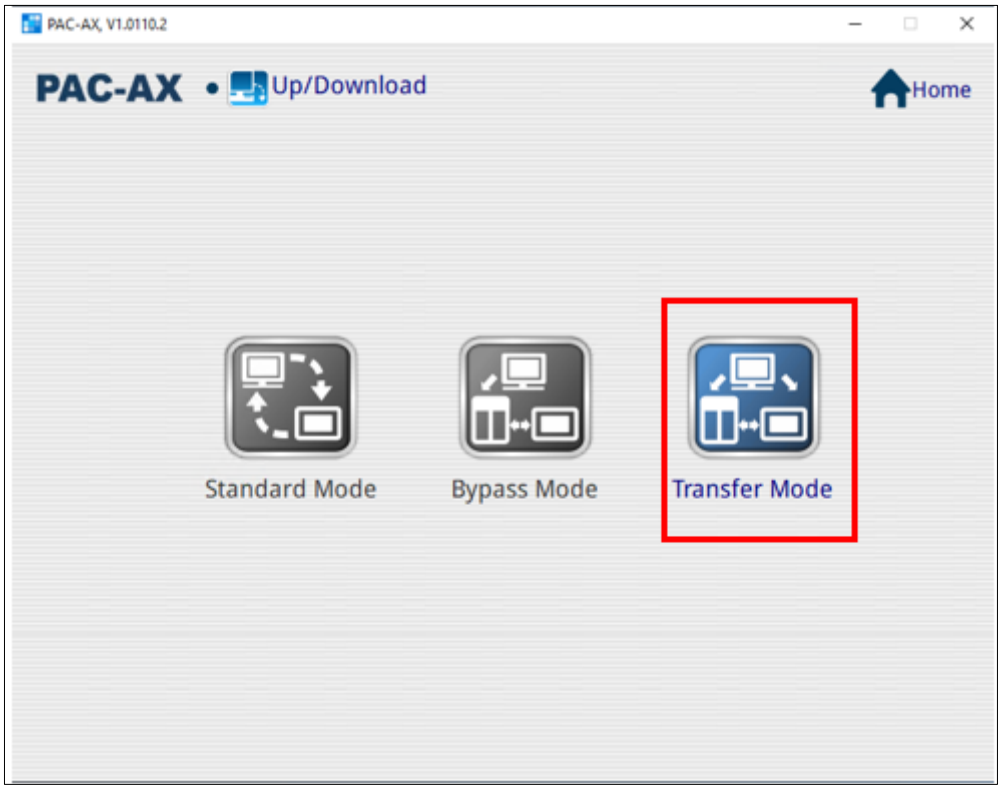


Figure 6 - 35: Transfer Mode

4. Click on the *CODESYS* tab.

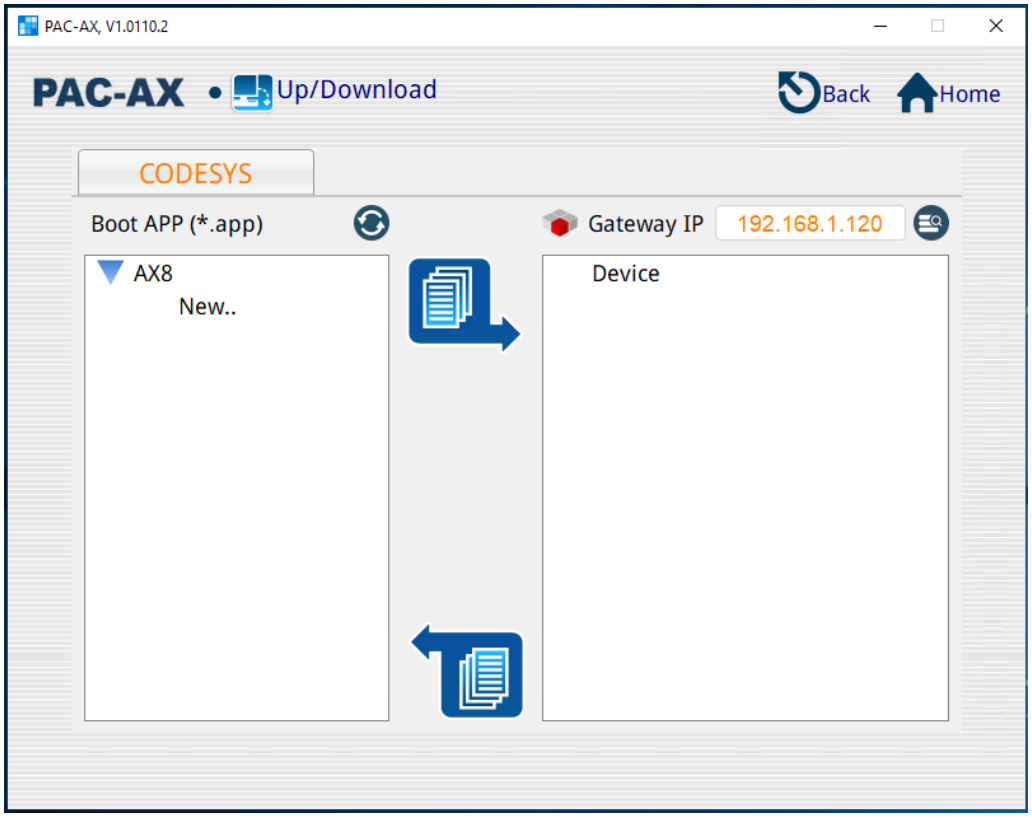


Figure 6 - 36: CODESYS tab

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*.

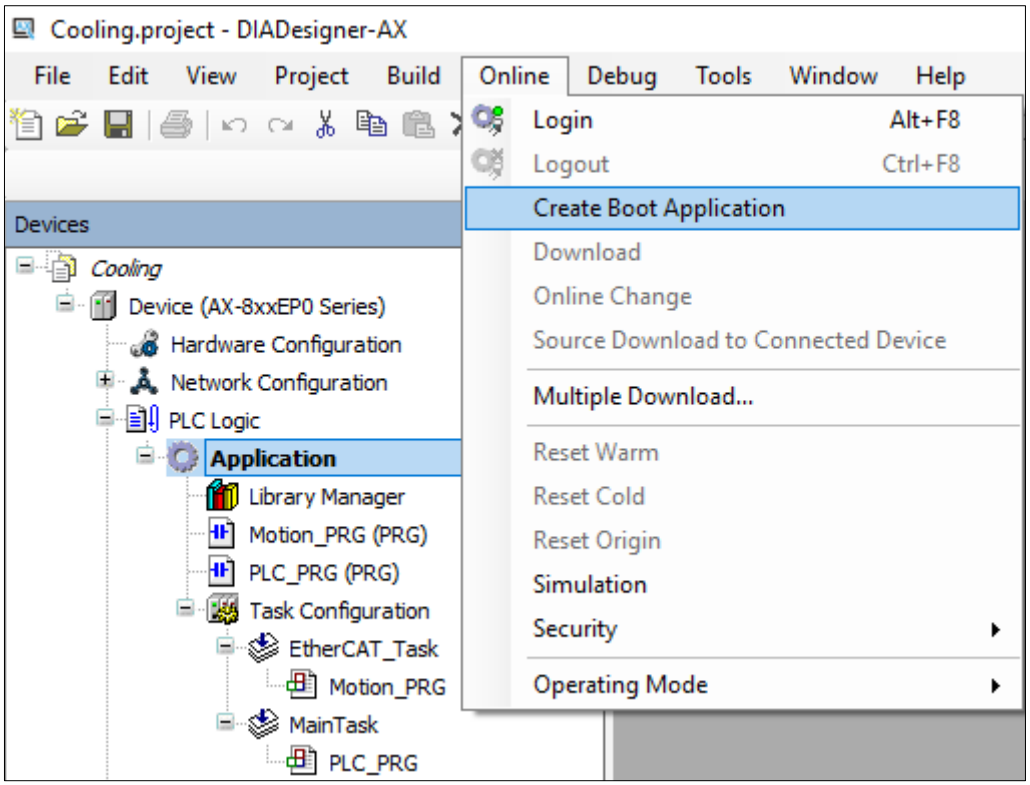
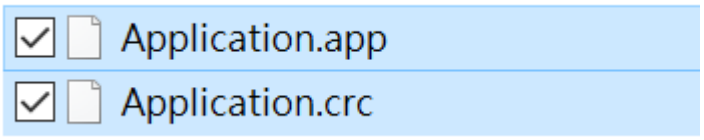


Figure 6 - 37: Create Boot Application path

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.

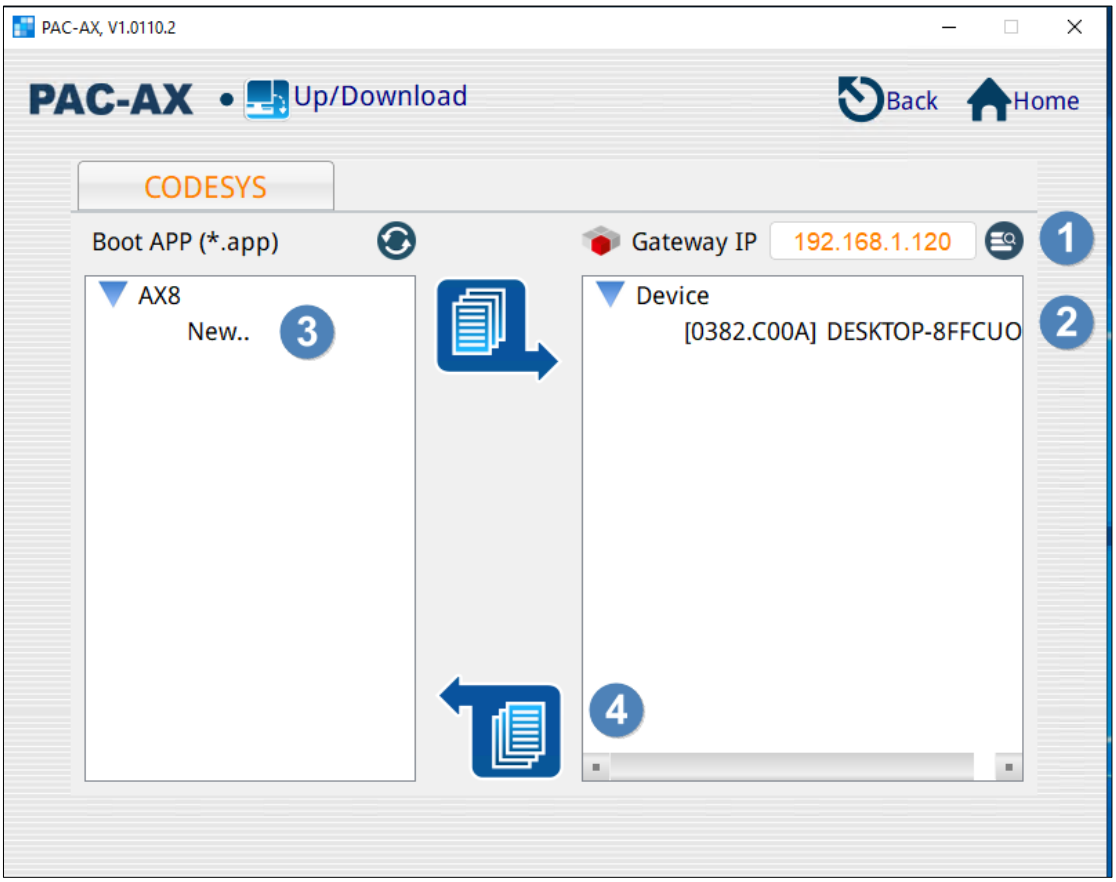


Figure 6 - 38: Upload

Result: Displays file upload success message.

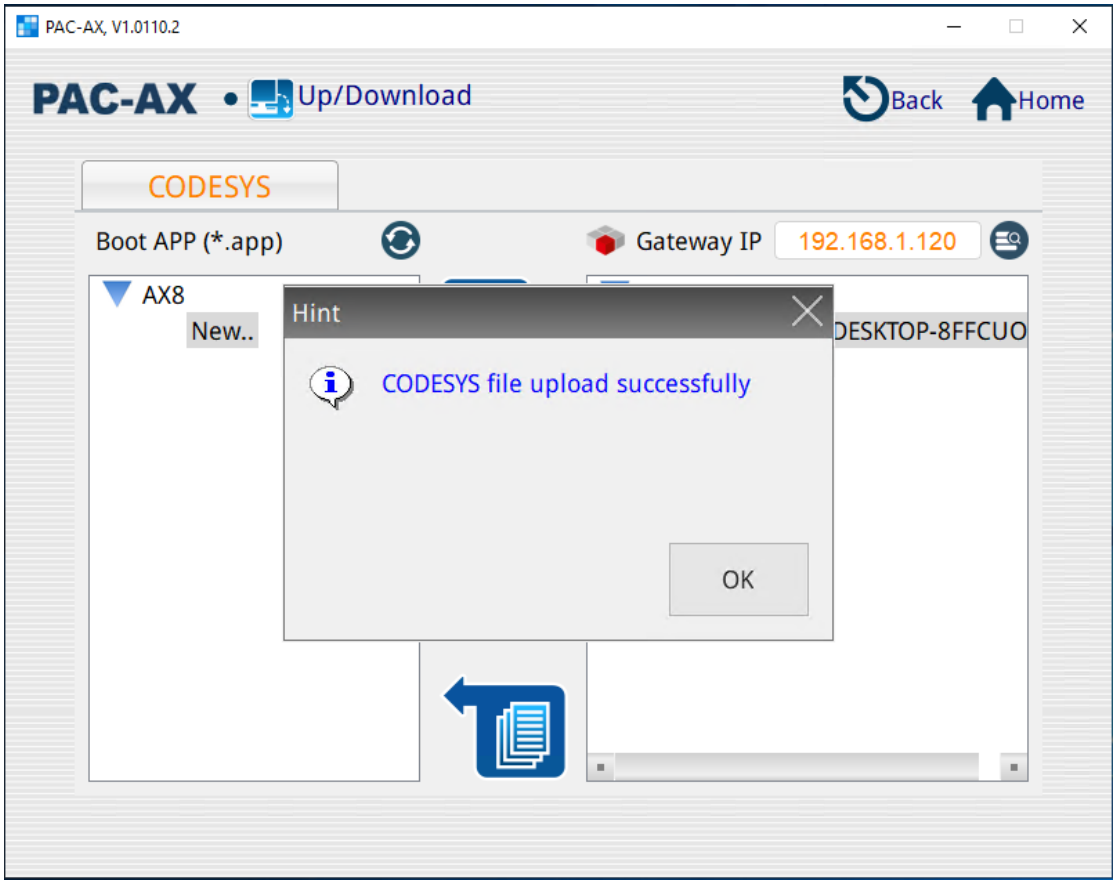


Figure 6 - 39: CODESYS file upload successfully

***NOTE**: After the upload is complete, there will be more .app files in the path.*

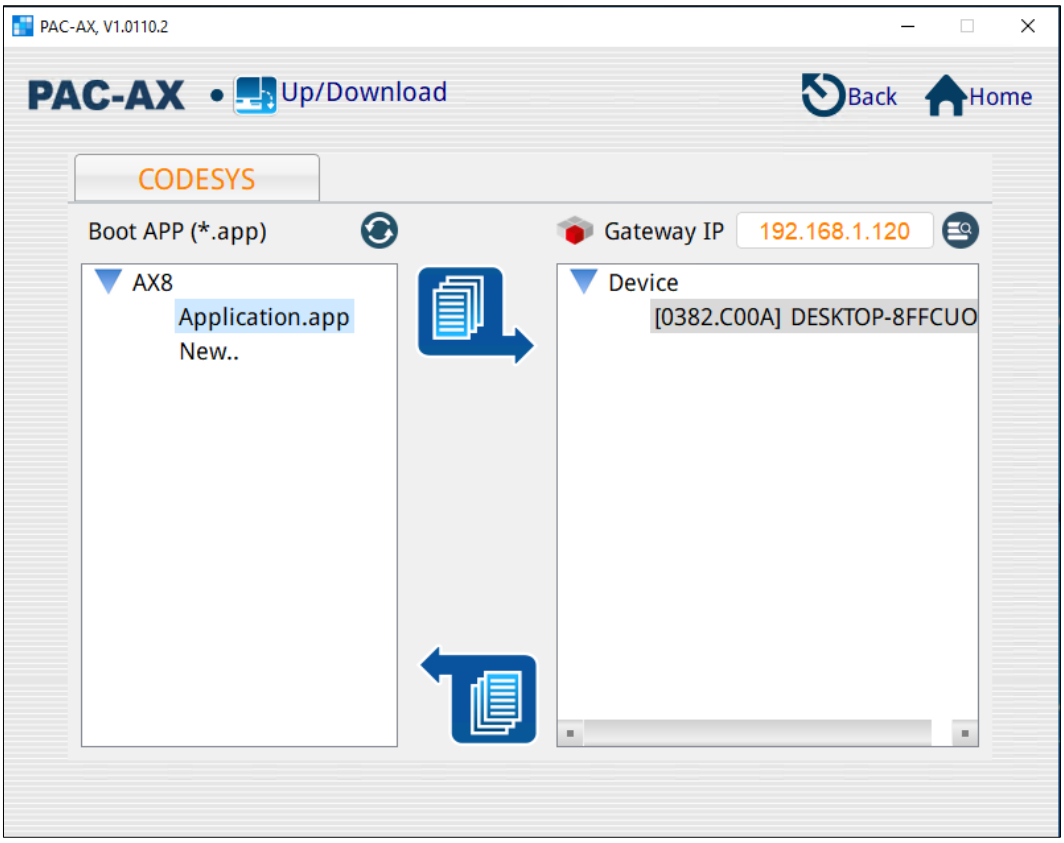


Figure 6 - 40: After Successful Upload

6.5.2 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  .

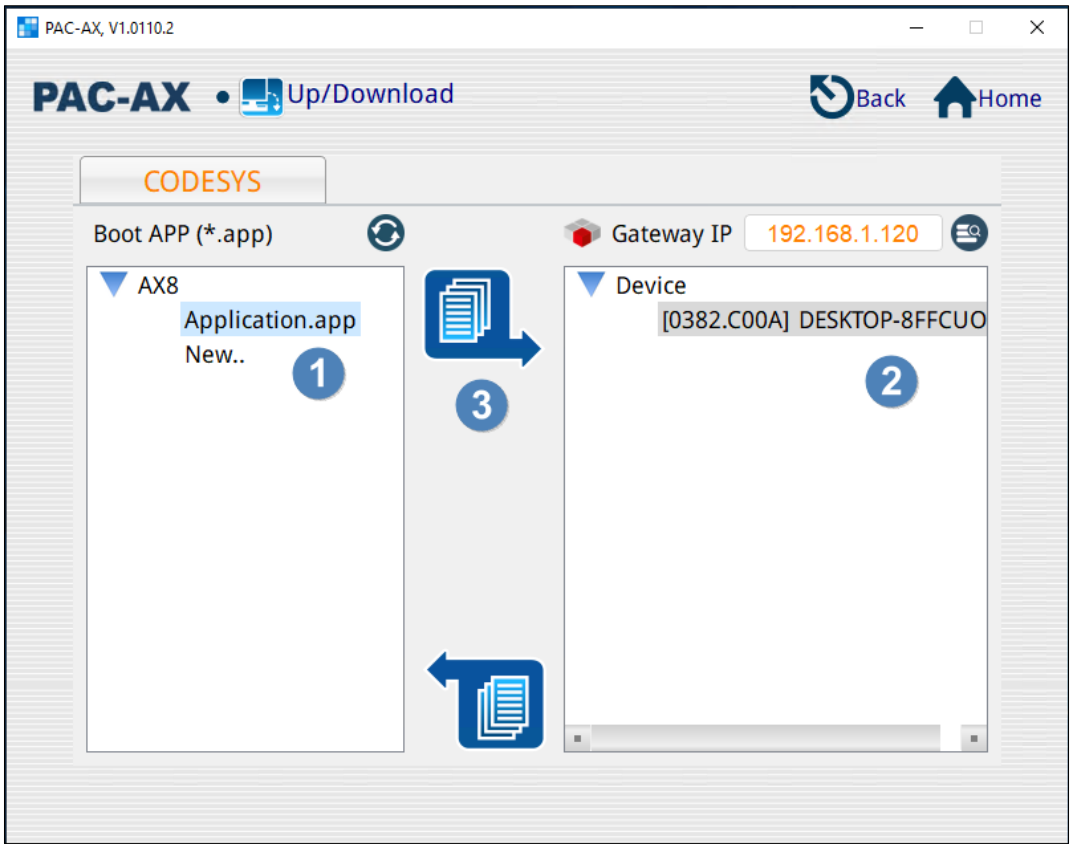


Figure 6 - 41: **Download**

4. A message after the download is successful, the newly downloaded Application program will run on the PLC.

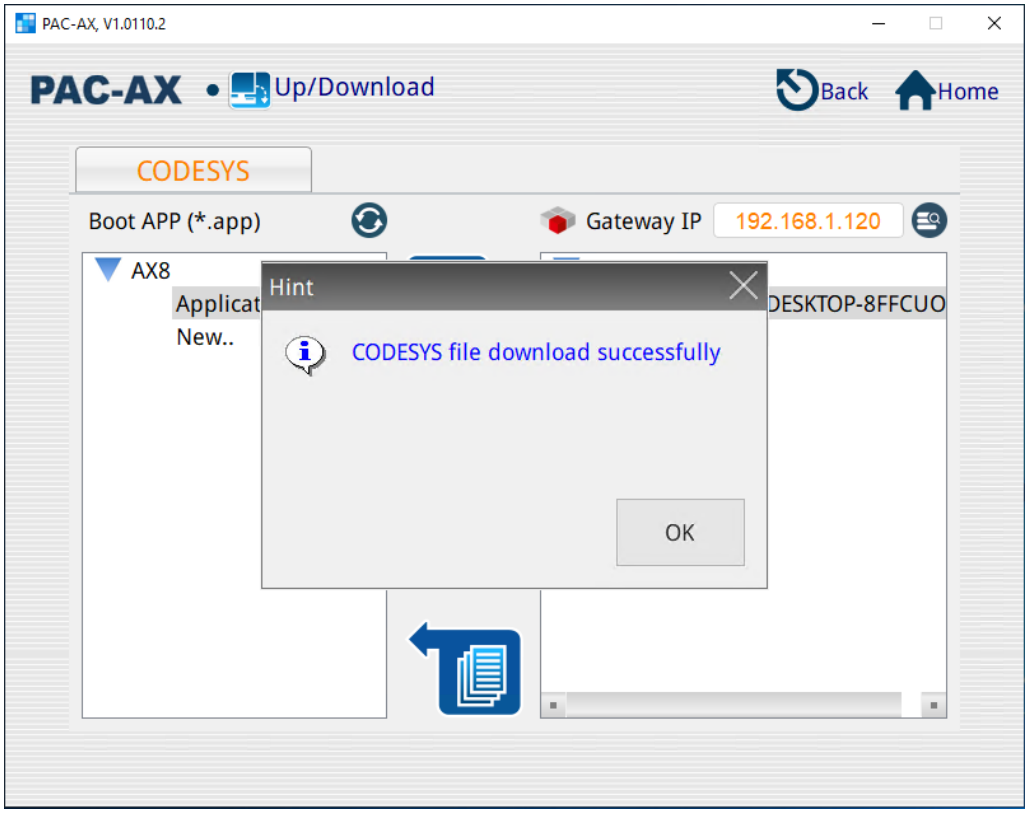


Figure 6 - 42: CODESYS File downloaded successfully

Chapter 7: 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, there is a version manager to update the latest library.

In addition to the pre-installed system and motion control libraries, user can also create and edit more libraries.

7.1 Device Repository

Users can follow the steps below to install third-party device xml files.

1. Click on *Tools*> *Device Repository.*

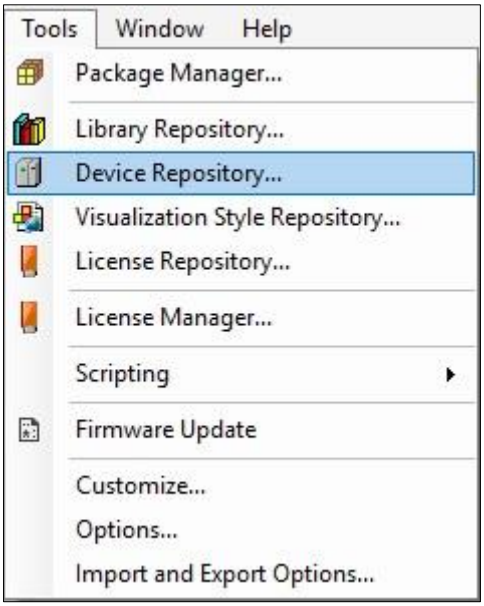


Figure 7 - 1: Device repository

Result: The device storage library window pops up.

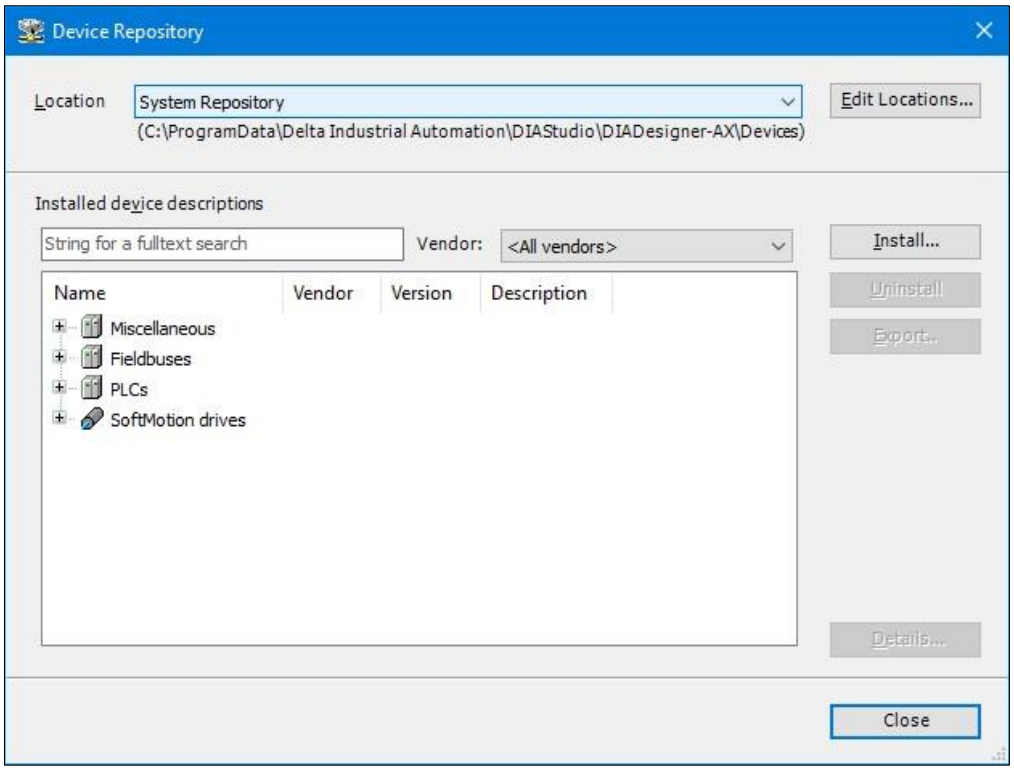


Figure 7 - 2: Device repository window

2. Click *Install* in the *device repository*.

   Result: The installation device description window pops up.

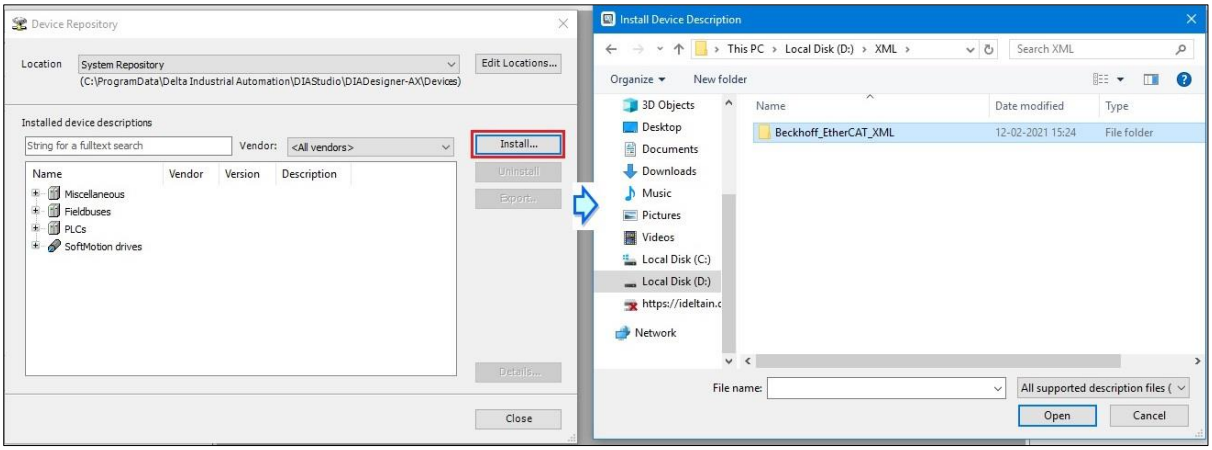


Figure 7 - 3: Install device profile

3. Select the *xml file* of the third-party device to be installed> click *Open*.

Result: The selected xml file has been installed in the device repository.

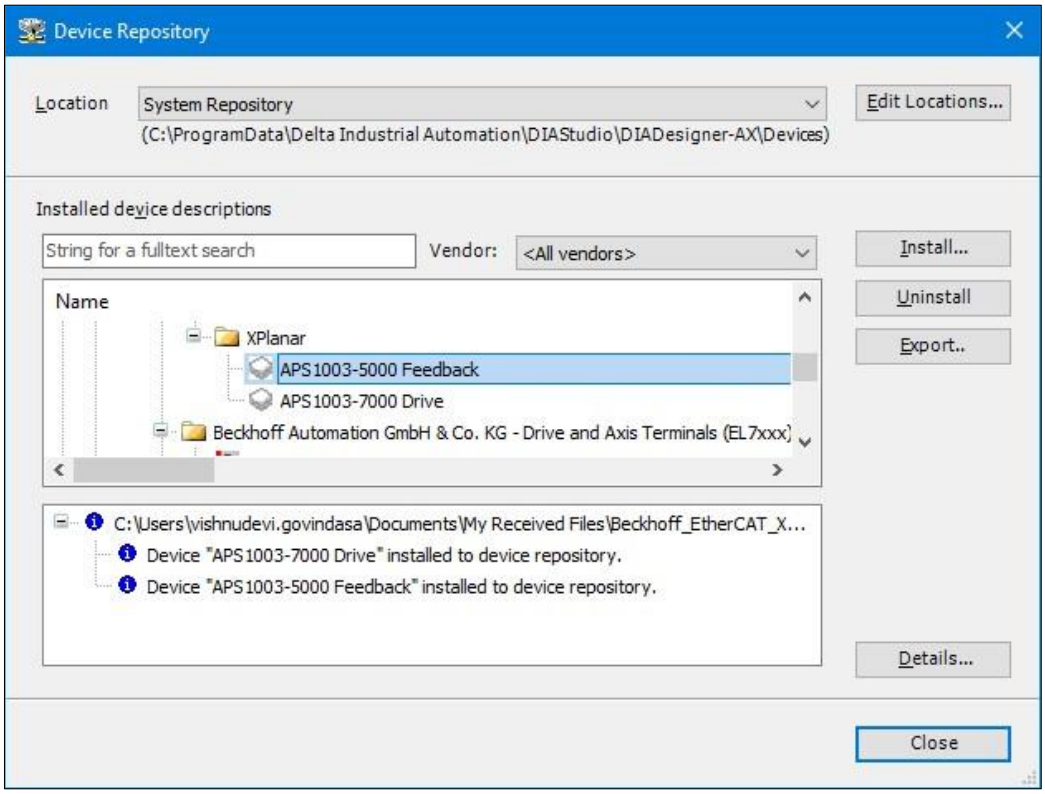


Figure 7 - 4: The xml file has been installed in the device repository

7.2 Library Manager

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.

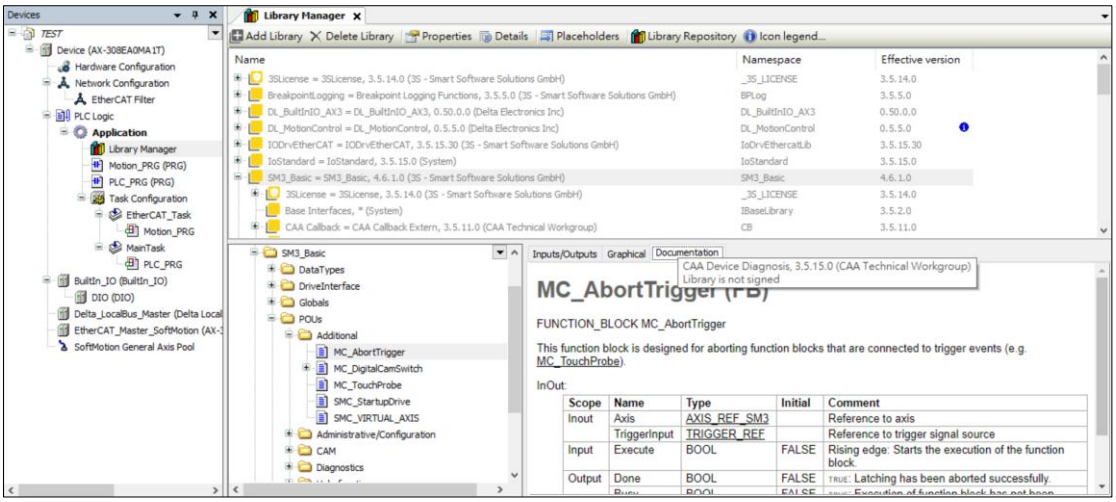


Figure 7 - 5: Library manager

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 document 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.

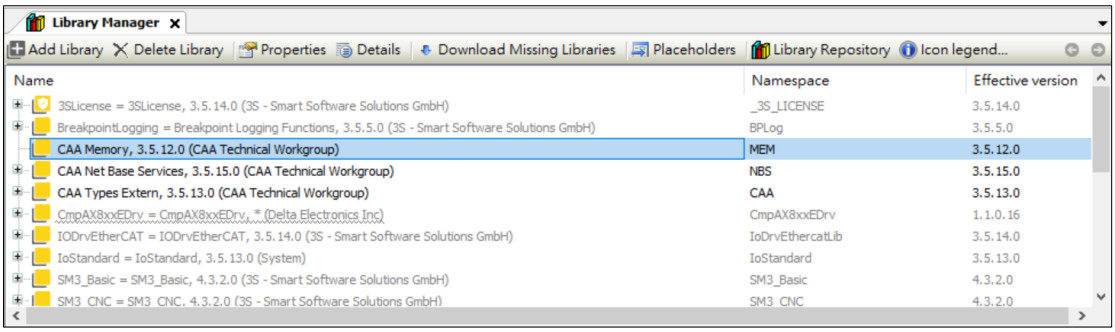


Figure 7 - 6: Library manager

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
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.

Project	Description
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.

7.2.1 Add Library

The following example shows how to add the Util library to userr application. User can add various libraries to use in the device in this way.

1. Select the *Library Repository*.

2. Select *Add Library* to open the dialog box for adding library.

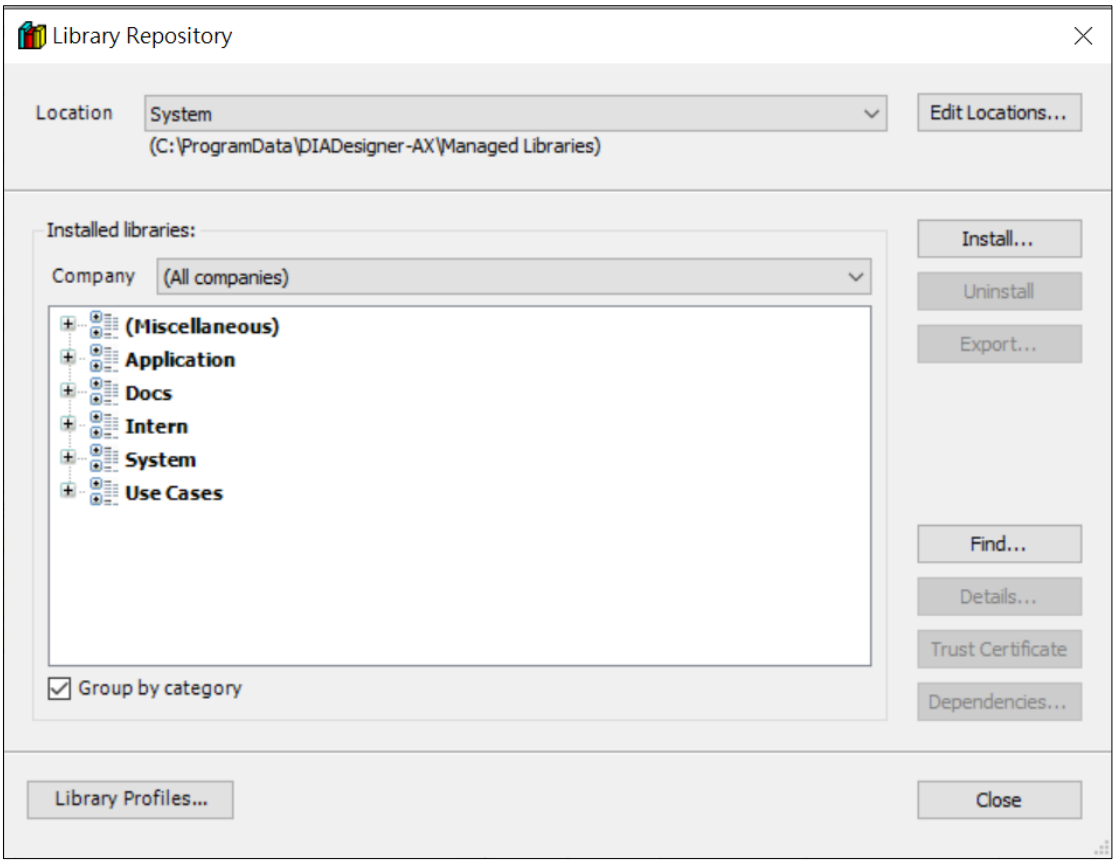


Figure 7 - 7: Add library

Project	Description
String for a fulltext search..	In the row above the library list, user can search for the library name or library module by entering the appropriate string.
Library	Appropriate library installed in the inventory repository. For example, the choice of library is in the device description or defined by the OEM. By default, the displayed libraries are divided into several categories.
Company	Creator of the library.
Library Repository	Open the Advanced Add Library dialog box
	The displayed libraries are divided into several categories.
	The displayed libraries are listed in alphabetical order.
	Show all available libraries.

- **Function library**

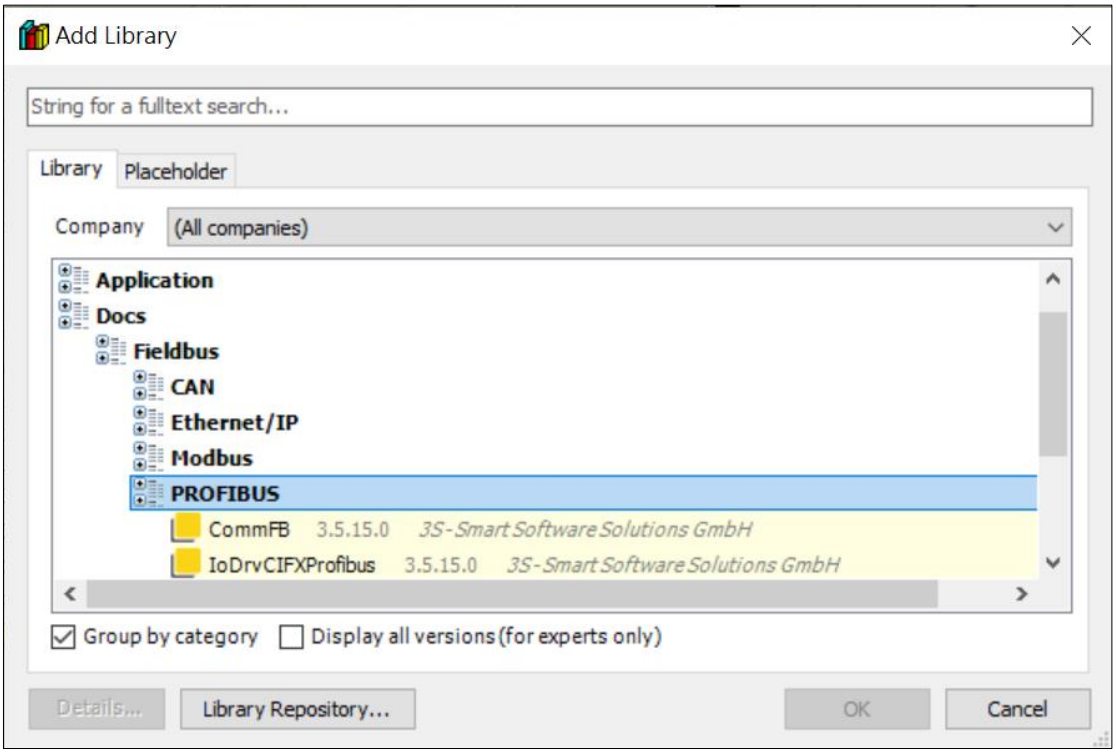


Figure 7 - 8: **Function Library**

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.
Display all versions	Checked: Show all versions of libraries. Version specification '*' means the latest version available in the repository. Unchecked: Only display the latest version of the library. Multiple libraries can be selected in this display. To do this, hold down the [Shift] key and select the entry.
Details	Details of library modules.
Library Repository	Inventory storage dialog box. More libraries can be installed to the local system.

- **Placeholder**

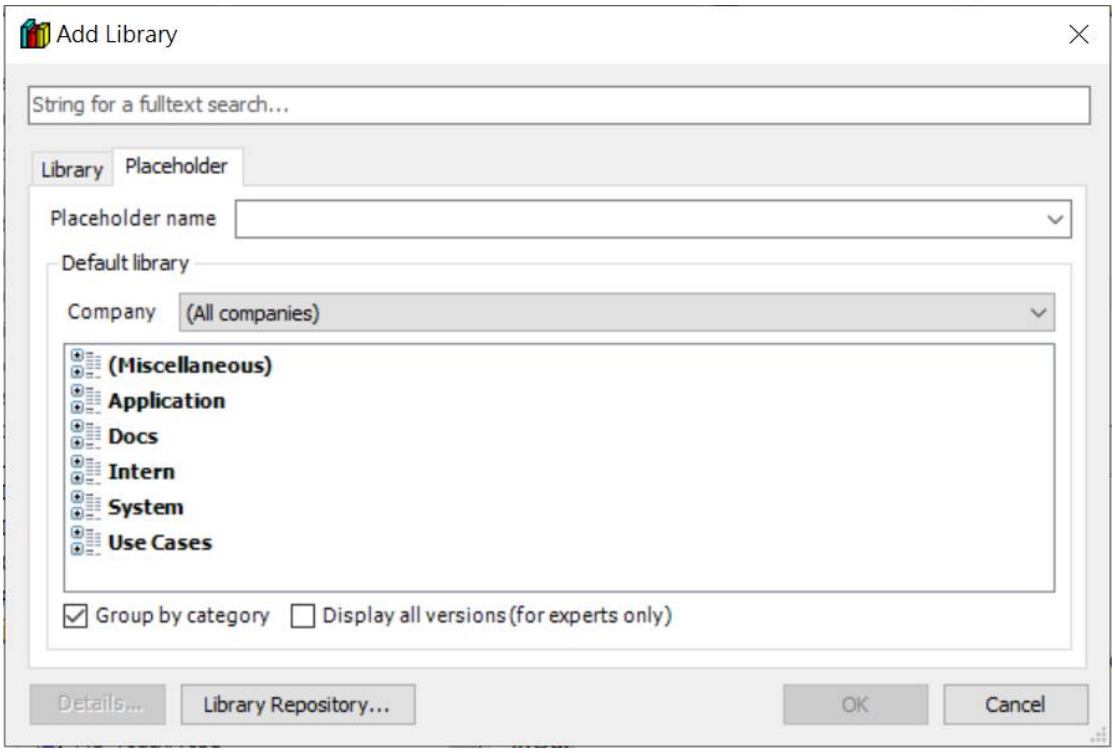


Figure 7 - 9: Placeholder

Project	Description
Placeholder name	The input field provides a drop-down list box for inputting the valid placeholder name read from the currently accessible device description.
the 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.
Show all versions	Checked: Show all versions of libraries. Version specification '*' means the latest version available in the repository. Unchecked: Only display the latest version of the library. Multiple libraries can be selected in this display. To do this, hold down the [Shift] key and select the entry.
Details	Details of library modules.
Library	For any reason, when the device defined by the resolution is not available, CODESYS uses this library. In this way, the current project can be compiled correctly.

3. Enter the string *Util* in the search box to browse the library, select the Util library, and click *OK*.

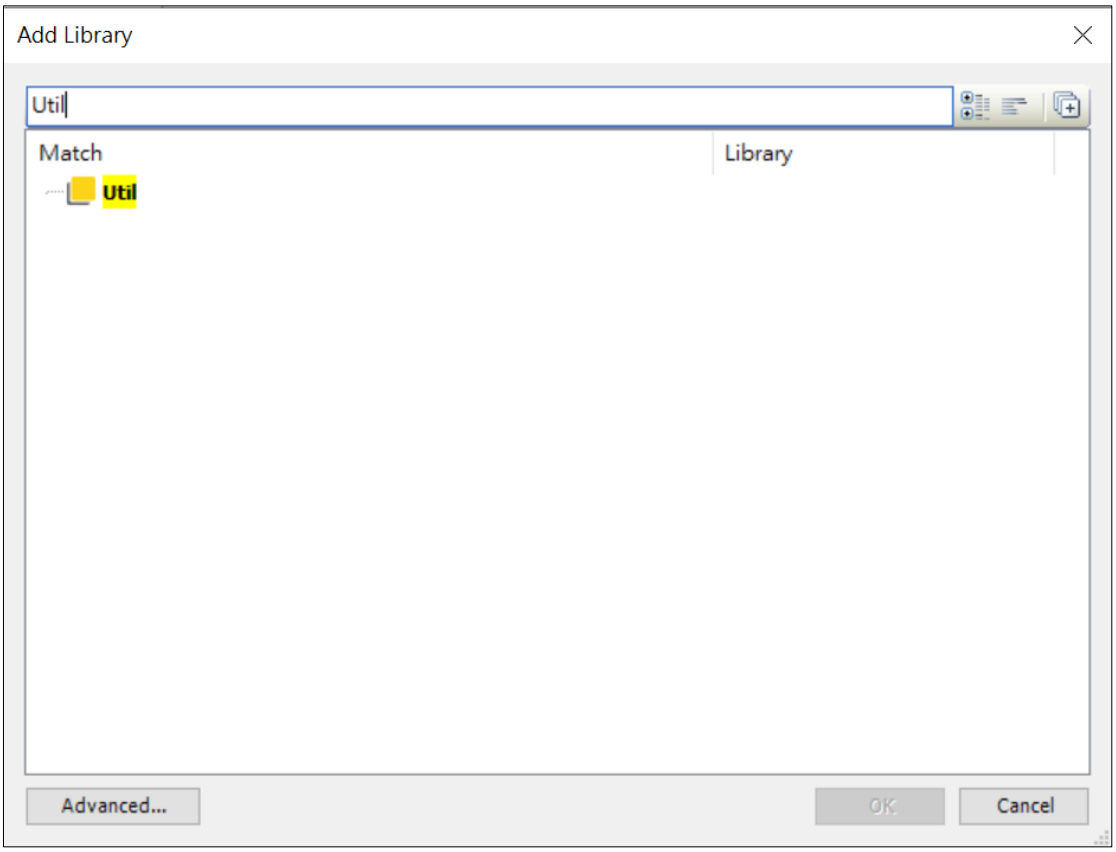


Figure 7 - 10: **Util Library**

4. The Util library has been added to the library manager and can be used in the library manager.

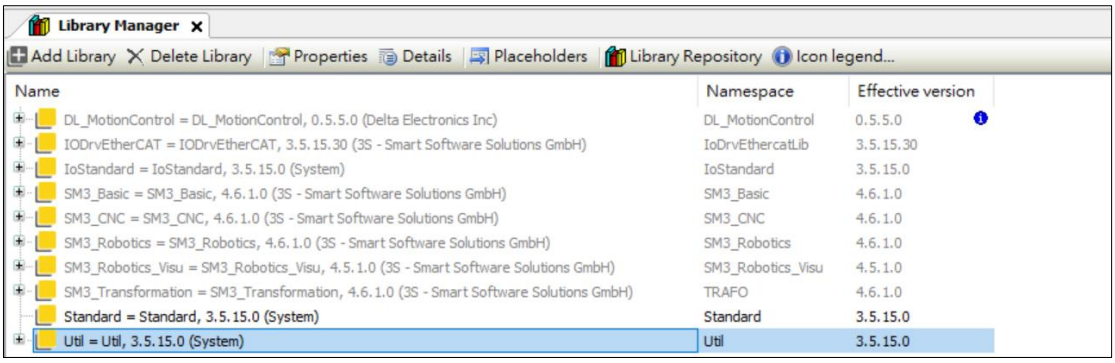


Figure 7 - 11: Added the Util library to the library manager

7.2.2 Properties

Properties dialog box of the selected library is available in the library manager.

**NOTE: It is recommended for people with professional knowledge related to library development to use.*

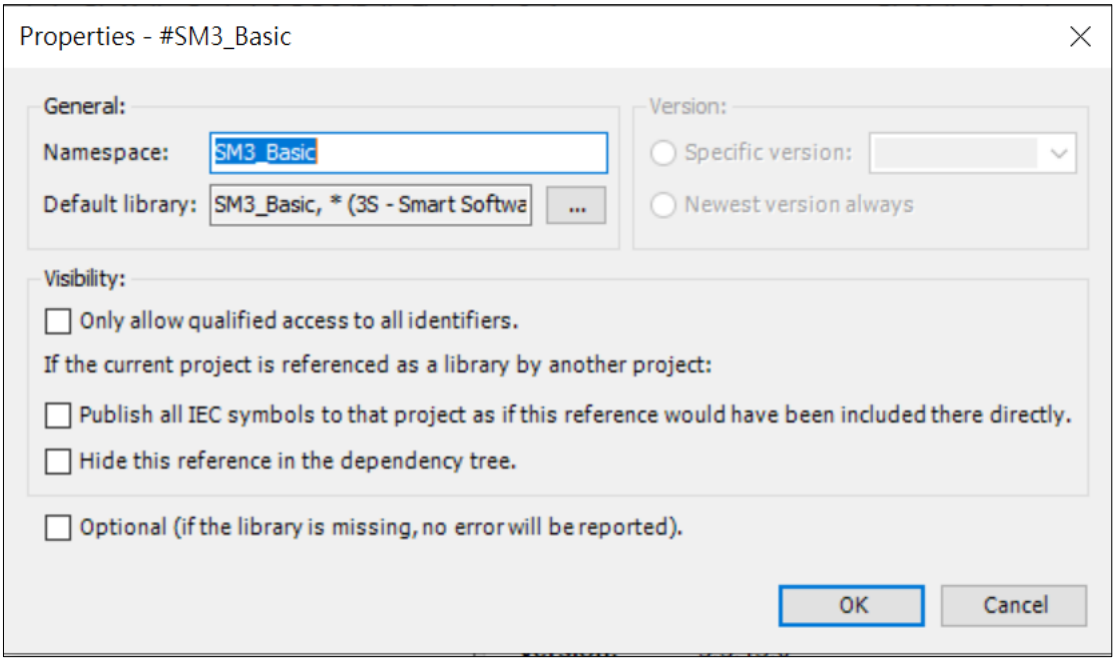


Figure 7 - 12: Library Properties

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	
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.
Optional	The selected library is considered an optional library. When downloading a project that references the library, even if

Project	Description
	the library is not available in the library repository, no errors will be reported.

7.2.3 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.

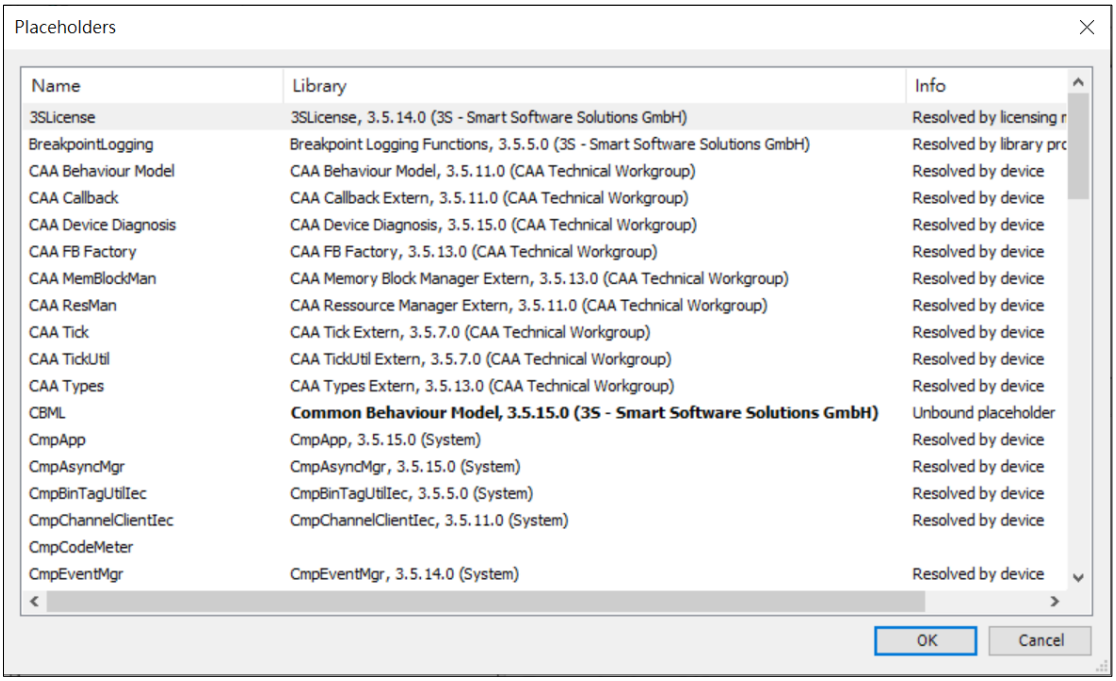
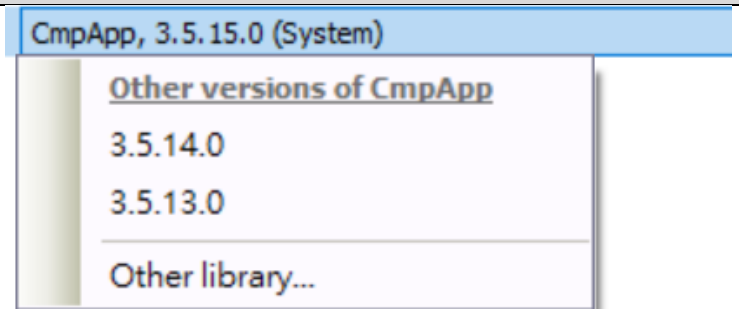


Figure 7 - 13: Placeholder

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.

Project	Description
	
Info	Placeholder device type: • Device analysis: Solve by device 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.

7.2.4 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.

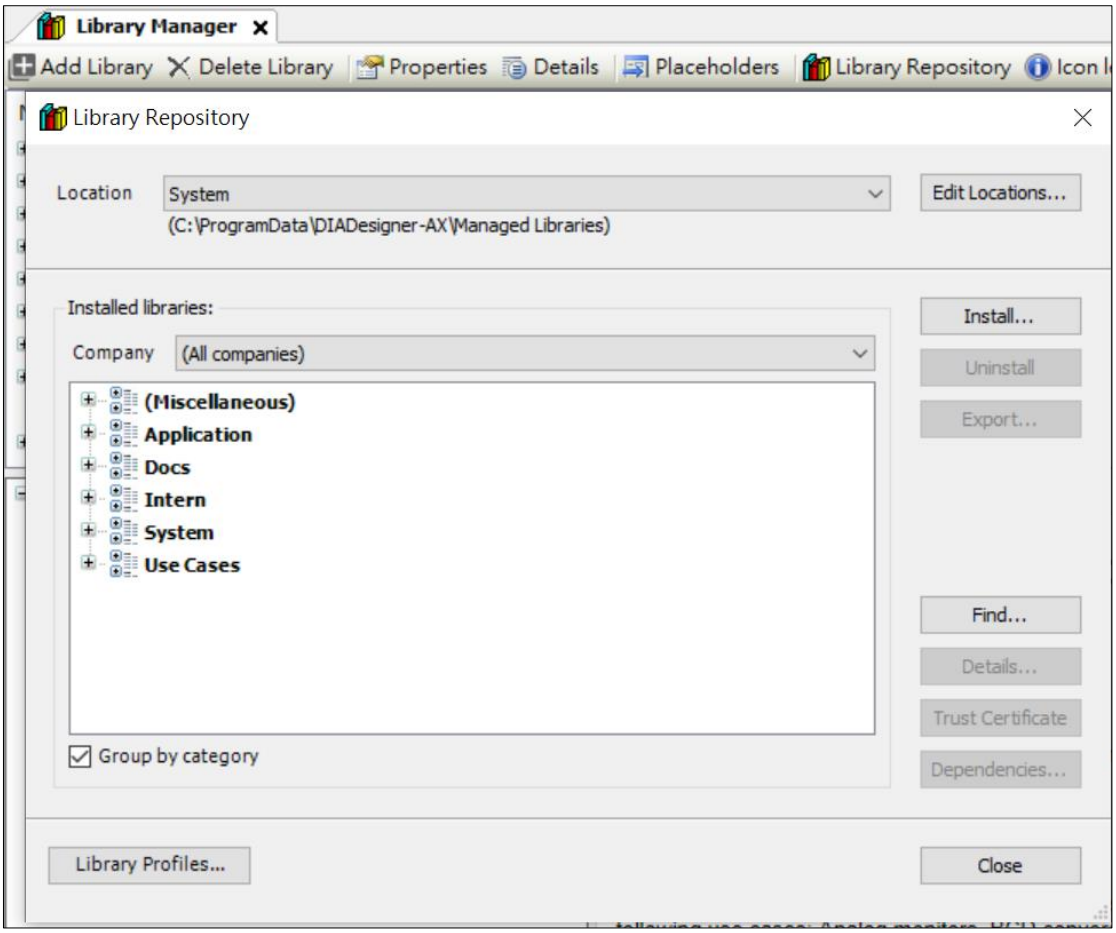


Figure 7 - 14: Resource Library

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). • 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

Project	Description
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 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.

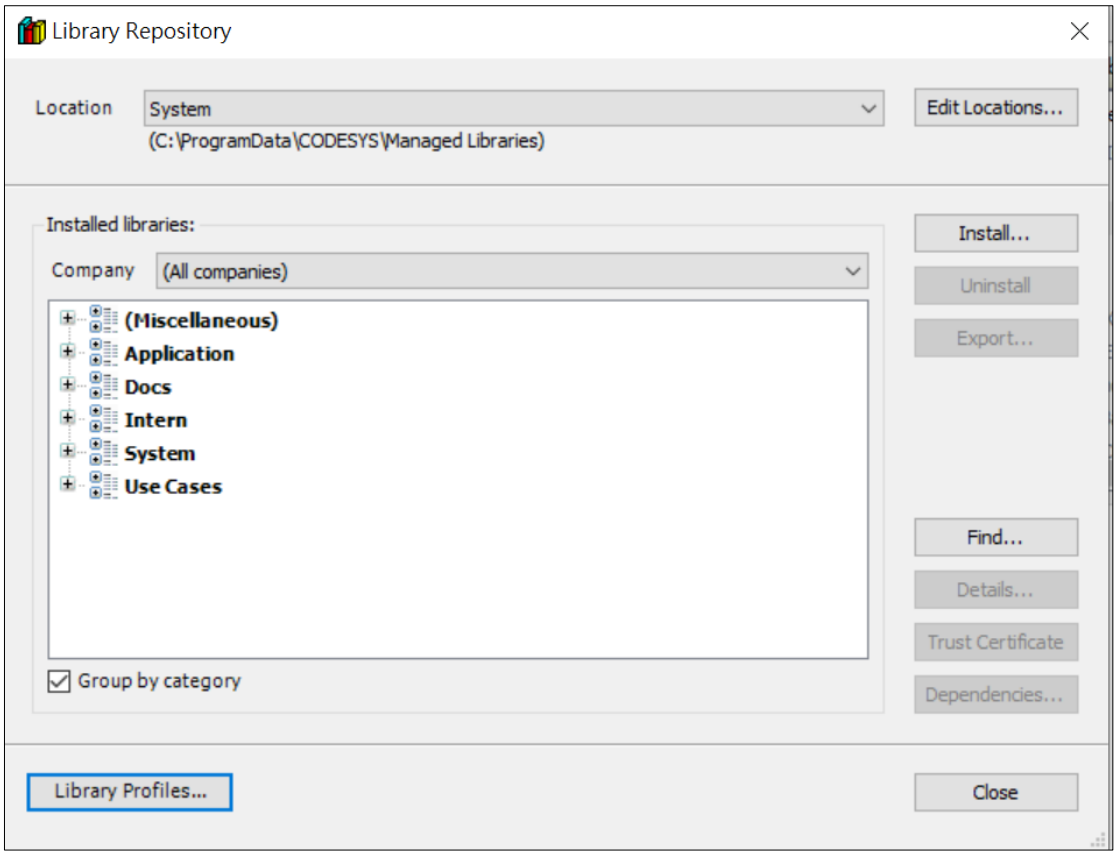


Figure 7 - 15: **Install library**

After the installation is complete, the new installation library will appear in the resource library.

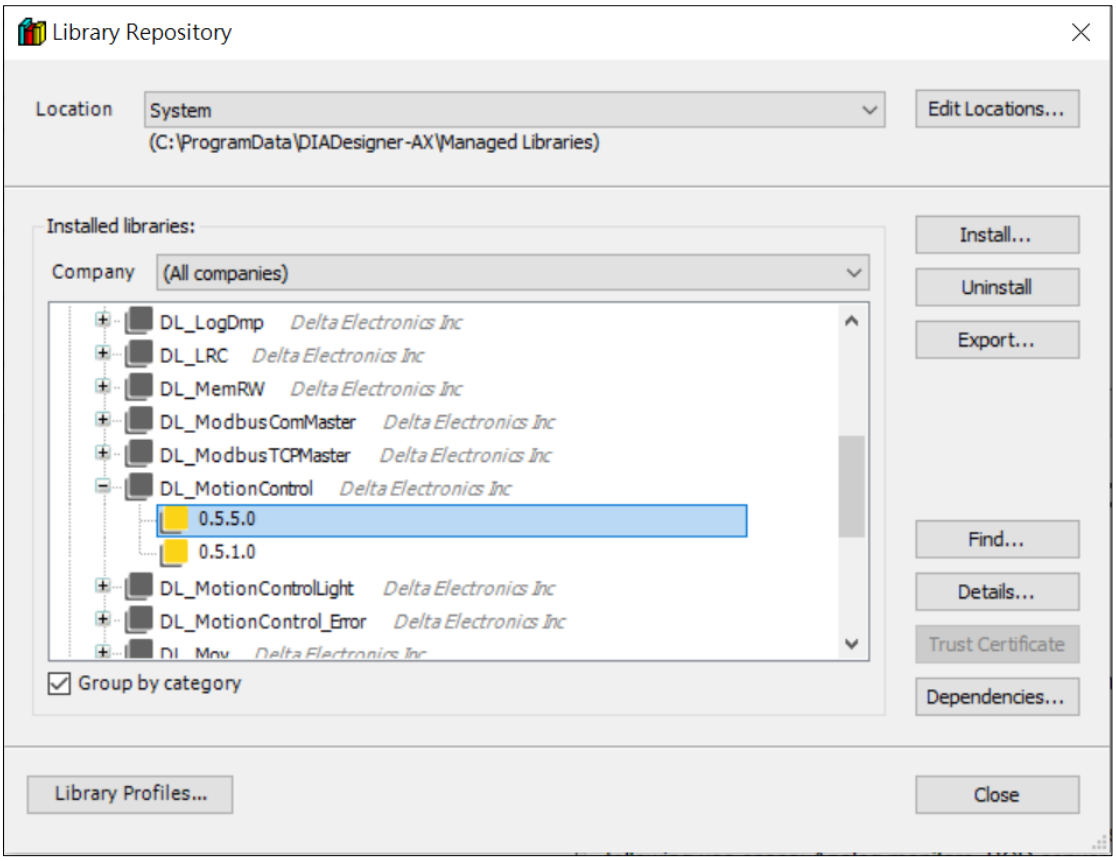


Figure 7 - 16: **New installation library**

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.

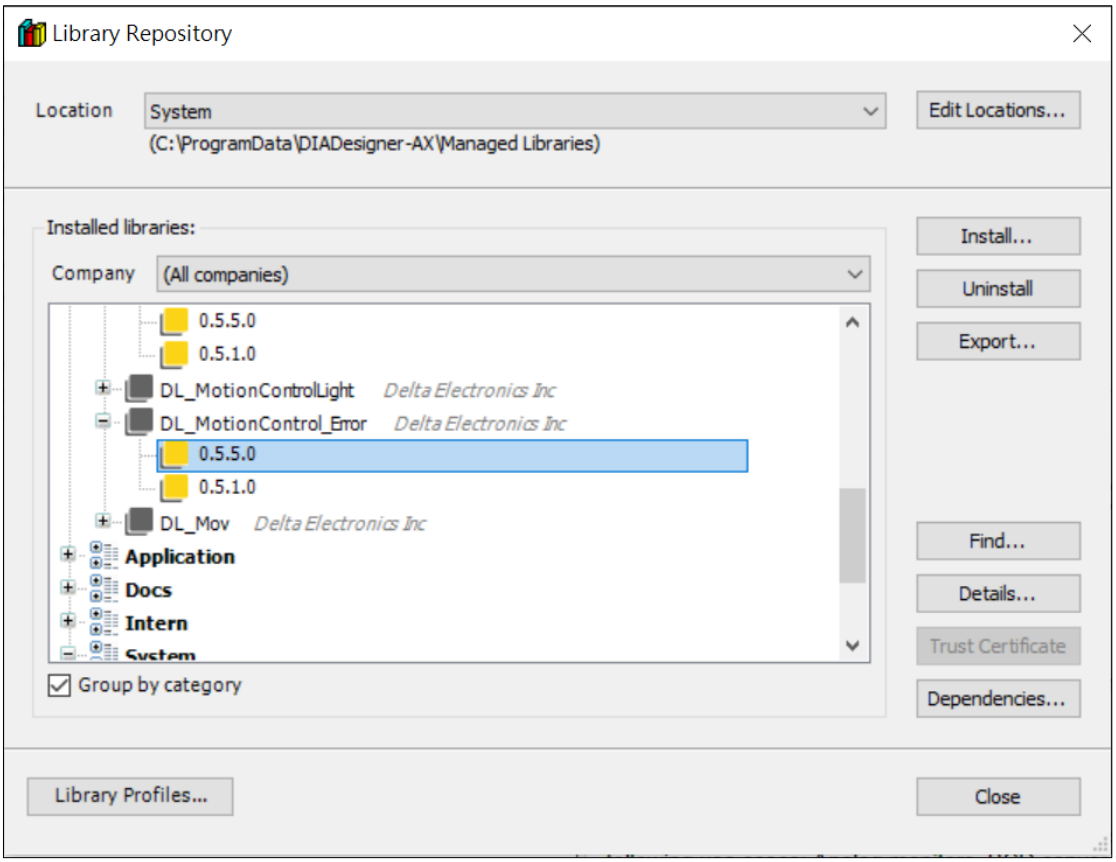


Figure 7 - 17: Library selection

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.

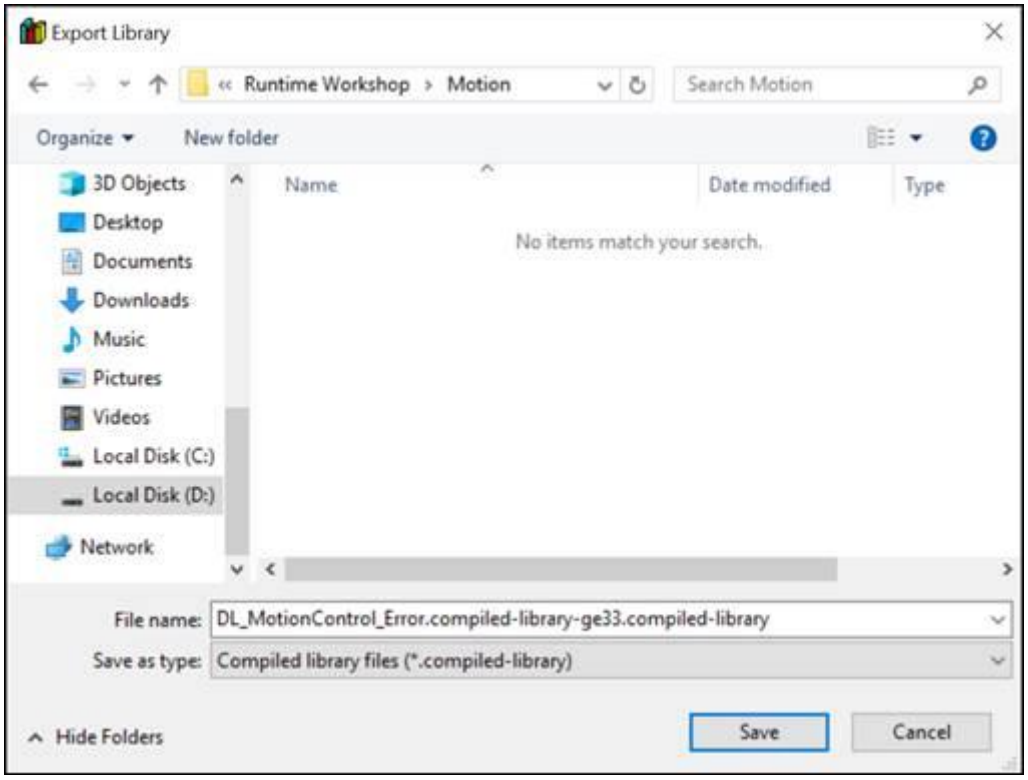


Figure 7 - 18: **Export Library**

7.3 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.

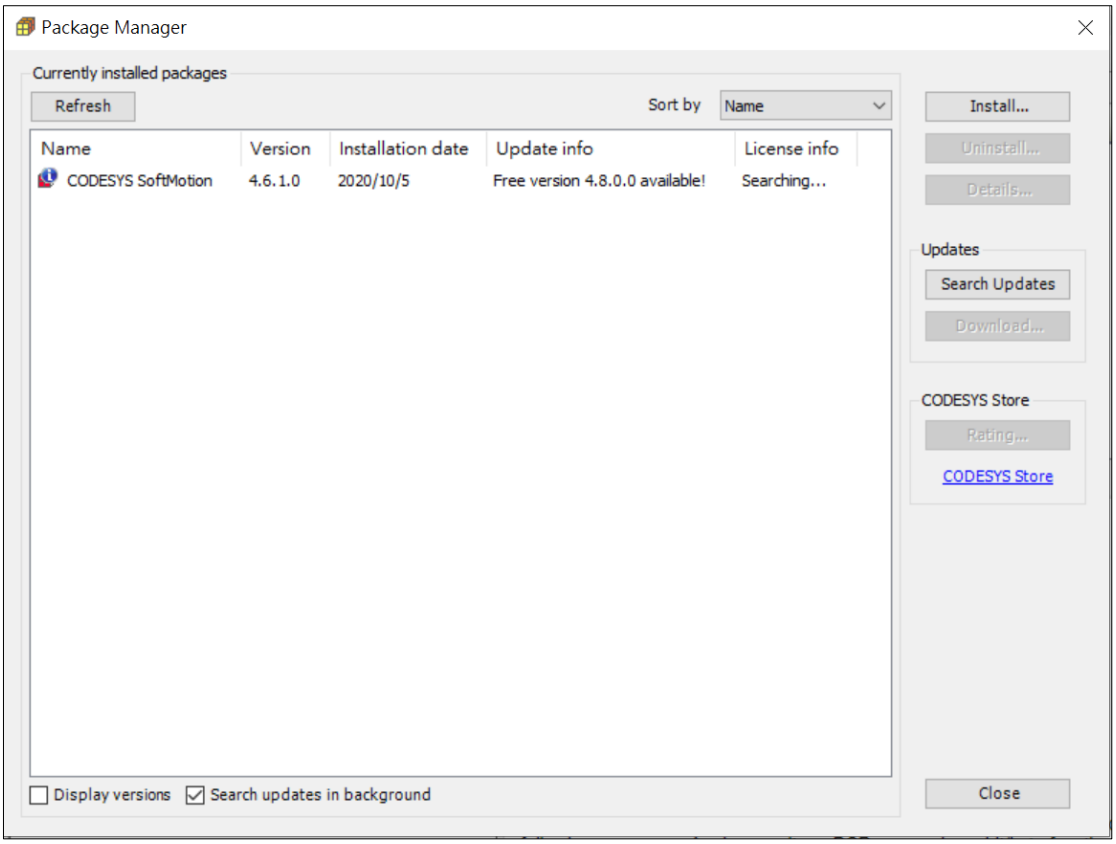


Figure 7 - 19: Package manager

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

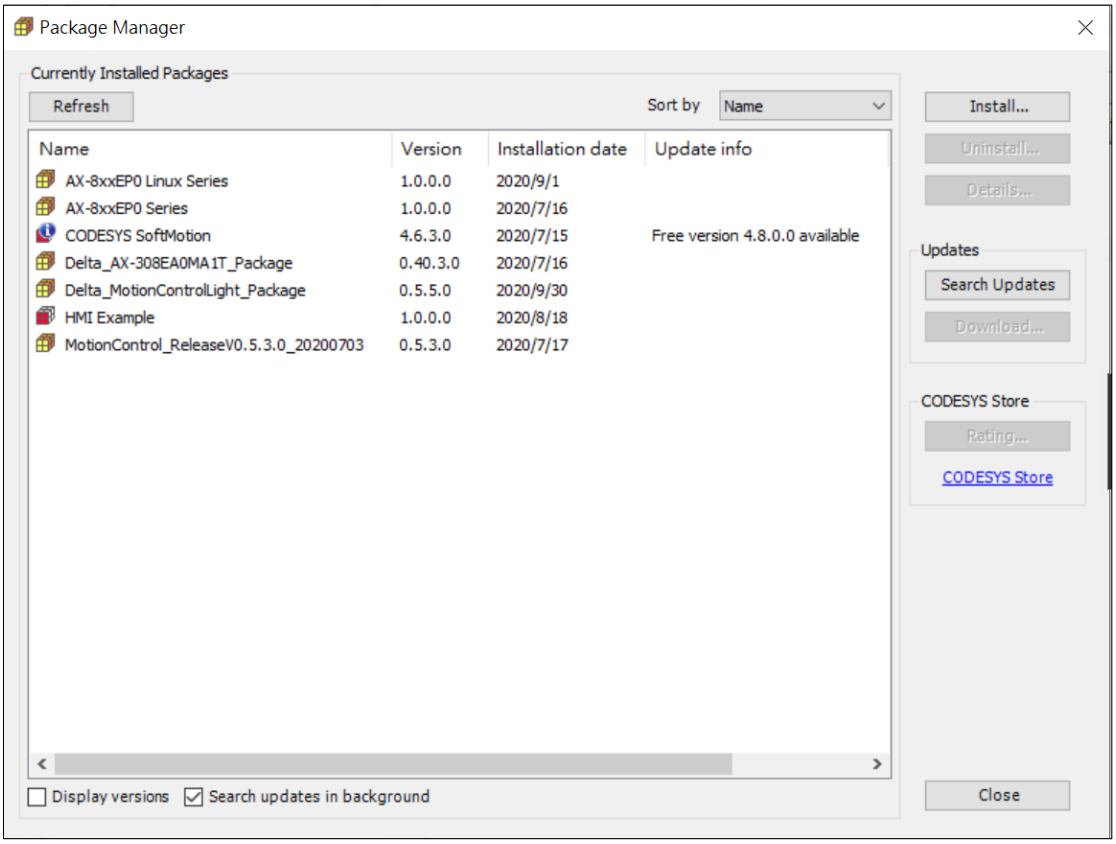


Figure 7 - 20: **Package manager**

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. After selecting the package, the installation steps are shown as follows: • Installation-License Agreement

Project	Description

	Show package checksum in installation protocol dialog.
	Only displayed if the package contains a license agreement.
	• Choose installation type
	  ○ Full installation: install all components
	  ○ 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
	  ○ Name: The name of the package
	  ○ Version
	  ○ Checksum: packaged SHA-1 CRC
	  ○ 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.

Project	Description
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. *Tools* > *Package Manager*.

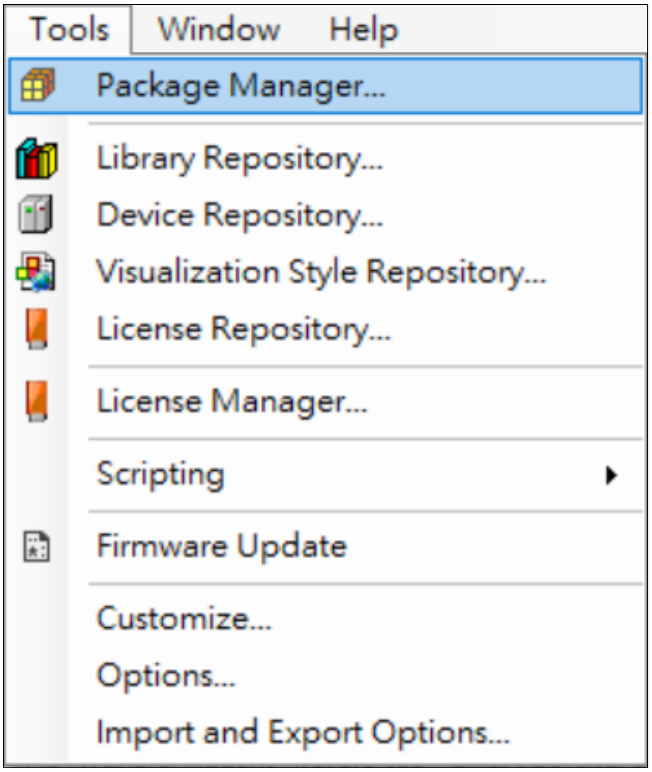


Figure 7 - 21: **Install a package path**

2. Click *Install*, select the package user want to install and open it to install.

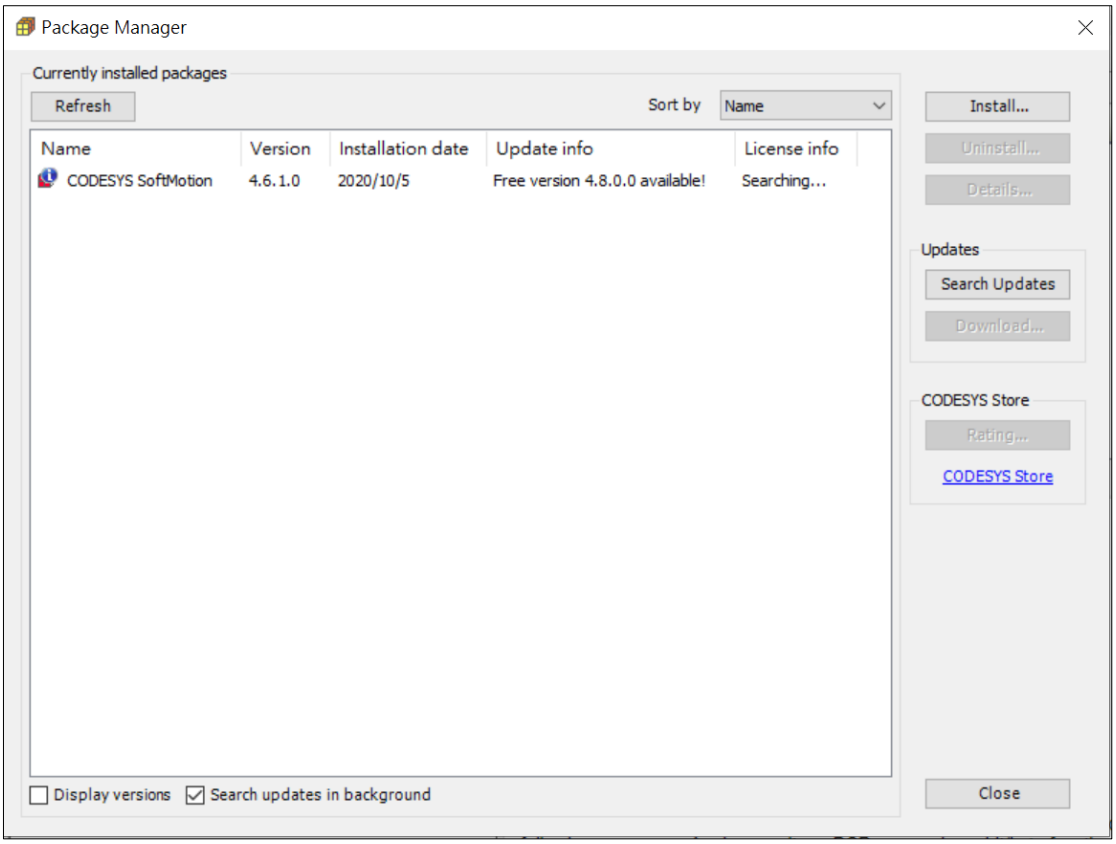


Figure 7 - 22: **Installation Package**

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*.

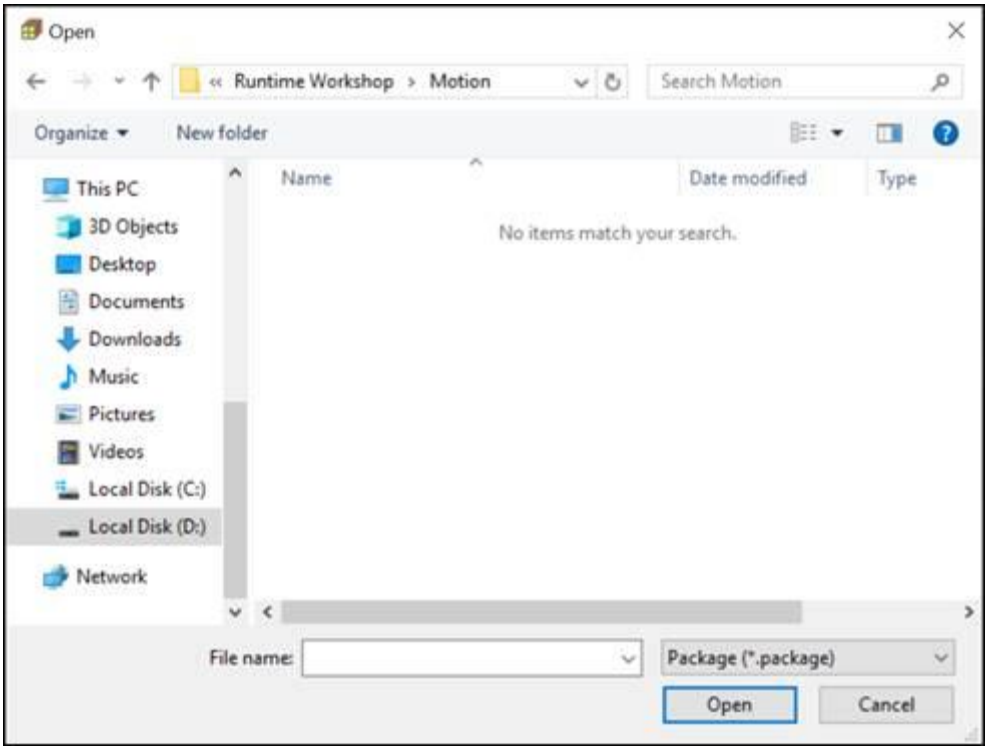


Figure 7 - 23: Open Window

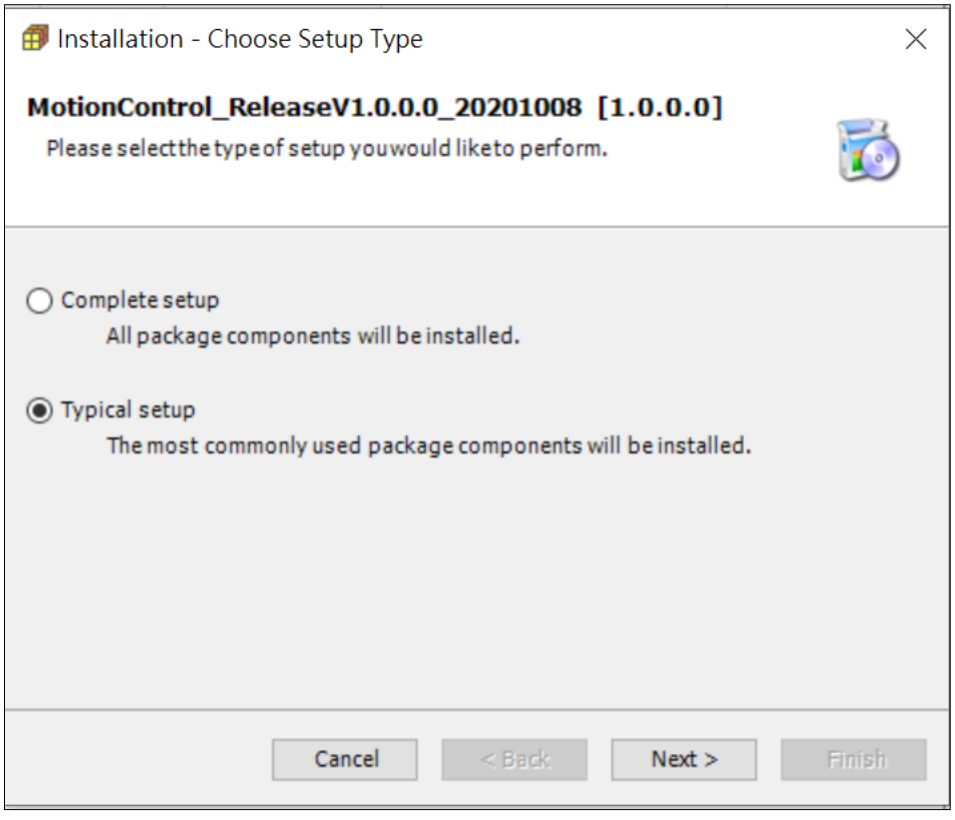


Figure 7 - 24: Software package successfully installed

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.

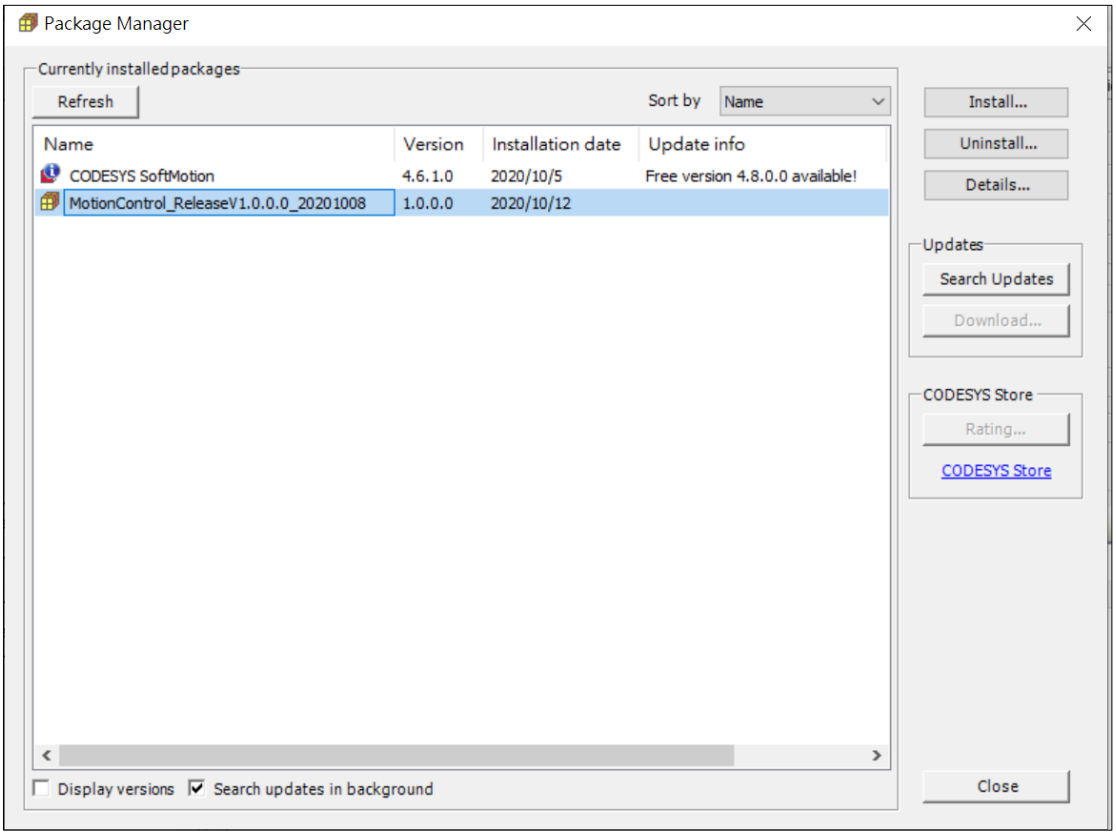


Figure 7 - 25: **Remove a package**

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