Digitized Automation for a Changing World

AX-5 Series Operation Manual

www.deltaww.com

AX-5 Series Operation Manual

Revision History

Version	Revision	Date
1st	The first version was published.	2024/10/15

AX-5 Series Operation Manual

Table of Contents

Chapter 1 Product Introduction

Chapter 2 Specifications and System Configurations

Chapter 3 Installing and Uninstalling DIADesigner-AX

Chapter 4 Basic Operation

Chapter 5 Hardware Configuration

Chapter 6 Network Configuration

Chapter 7 Motion Control Basic Settings and Operation

Chapter 8 Communication

Chapter 9 Convenience Function

Appendix A Troubleshooting

⚠ Caution

- ✓ This is an OPEN TYPE module and therefore should be installed in an enclosure free of airborne dust, humidity, electric shock and vibration. The enclosure should prevent non-maintenance staff from operating the device (e.g. key or specific tools are required for operating the enclosure) in case danger and damage on the device may occur.
- ✓ Please read this manual carefully and follow the instructions in the manual for operations, especially those marked with the safety warnings ⚠ in order to prevent injuries to personnel or damage to products.

Chapter 1 Product Introduction

Table of Contents

1.1 Overview

This manual introduces CPU functions, devices, module tables, electrical specifications, appearances and dimension, basic concept of motion control, basic configurations, troubleshooting, and so forth.

1.1.1 Related Manuals

The related manuals for AX-5 Series PLC are listed below.

- DIADesigner-AX Software Manual

  This introduces the use of the DIADesigner-AX software, programming language (Ladder, SFC, ST and FBD), POUs, tasks and programming for motion controls.
- AX-5 Series Operation Manual

  This introduces CPU functions, devices, I/O arrangements, electrical specifications, appearances, dimensions, motion control concepts, basic settings, troubleshooting, and so forth.
- AX-5 Series Module Manual

  This introduces special I/O modules such as network modules, analog I/O modules, temperature measurement modules, and so forth.
- AX Series - Standard Instructions Manual

  This introduces standard instructions for programming the AX Series Controllers.
- AX Series - Motion Controller Instructions Manual

  This introduces single-axis and multi-axes instructions for programming the AX Series Motion Controllers.

1.1.2 Table of Models and Descriptions

Classification	Model Name	Description
Unit power supply module	AX-502PS11-0A	Unit power supply module - Unit power supply 24 to 5 V, 2 A output
I/O power supply / connector module	AX-510PS12-0A	I/O power supply module - I/O power supply 24 VDC
	AX-516PC10-0A	IO power connector - 16 IOV
	AX-516PC20-0A	IO power connector - 16 IOG
	AX-516PC30-0A	IO power connector - 8 IOV / 8 IOG
	AX-514PC40-0A	IO power connector - 14 PE
AX-5 CPU modules	AX-516EB0MB1T	16-axis motion controller CPU module, 12x DIs (100 kHz, NPN), 4x DOs (80 kHz, NPN), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 64 MB, removable terminal blocks
	AX-516EB0MB1P	16-axis motion controller CPU module, 12x DIs (100 kHz, PNP), 4x DOs (80 kHz, PNP), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 64 MB, removable terminal blocks
	AX-532EB0MB1T	32-axis motion controller CPU module, 12x DIs (100 kHz, NPN), 4x DOs (80 kHz, NPN), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 64 MB, removable terminal blocks
	AX-532EB0MB1P	32-axis motion controller CPU module, 12x DIs (100 kHz, PNP), 4x DOs (80 kHz, PNP), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port,

Classification	Model Name	Description
		1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 64 MB, removable terminal blocks
	AX-564EB0MA1T	64-axis motion controller CPU module, 12x DIs (100 kHz, NPN), 4x DOs (80 kHz, NPN), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 64 MB, removable terminal blocks
	AX-564EB0MA1P	64-axis motion controller CPU module, 12x DIs (100 kHz, PNP), 4x DOs (80 kHz, PNP), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 64 MB, removable terminal blocks
	AX-564ELB0MB1T	64-axis point-to-point motion controller CPU module, 12x DIs (100 kHz, NPN), 4x DOs (80 kHz, NPN), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 32 MB, removable terminal blocks
	AX-564ELB0MB1P	64-axis point-to-point motion controller CPU module, 12x DIs (100 kHz, PNP), 4x DOs (80 kHz, PNP), 2x Ethernet ports (independent IPs), 1x CANopen port, 1x EtherCAT port, 1x RS-485 port, 1x RS-232 port, 1x USB-Type A port, 1x USB-Type C, Micro SD interface, Program capacity: 32 MB, removable terminal blocks
Digital input / output module	AX-516AM10-0A	16 DI, 24 VDC, 2.5 mA,1 ms, Source Input (NPN)
	AX-516AM20-0A	16 DI, 24 VDC, 2.5 mA,1 ms, Sink Input (PNP)
	AX-516AN01-0A	16 DO, 24 VDC, 0.5 A,1 ms, Sink Output (NPN)
	AX-516AN02-0A	16 DO, 24 VDC, 0.5 A,1 ms, Source Output (PNP)
	AX-516AP11-0A	8 DI, 24 VDC, 2.5 mA,1 ms, Source Input (NPN) 8 DO, 24 VDC, 0.5 A,1 ms, Sink Output (NPN)
	AX-516AP22-0A	8 DI, 24 VDC, 2.5 mA,1 ms, Sink Input (PNP) 8 DO, 24 VDC, 0.5 A,1 ms, Source Output (PNP)
Analog input / output module	AX-504AD10-0A	4 AI, ±10V, 16 bits, 100 µs
	AX-504AD20-0A	4 AI, ±20mA, 16 bits, 100 µs
	AX-508AD10-0A	8 AI, ±10V, 16 bits, 100 µs
	AX-508AD20-0A	8 AI, ±20mA, 16 bits , 100 µs
	AX-504DA01-0A	4 AO, ±10V, 16 bits, 100 µs
	AX-504DA02-0A	4 AO, 0–20mA, 16 bits, 100 µs
	AX-508DA01-0A	8 AO, ±10V, 16 bits, 100 µs
Temperature measurement module	AX-502TC10-0B	2 TC, J, K, T, E, L, U, N, R, S, B, C (WRe5-26), PLII, ±130 mV, 24 bits
	AX-504TC10-0B	4 TC, J, K, T, E, L, U, N, R, S, B, C (WRe5-26), PLII, ±130 mV, 24 bits
	AX-502PT10-0B	2 RTD, PT100, PT1000, Ni100, Ni1000, 0 to 300 Ω, 24 bits
	AX-504PT10-0B	4 RTD, PT100, PT1000, Ni100, Ni1000, 0 to 300 Ω, 24 bits
Load cell module	AX-501LC10-0A	1 CH Load cell, 5 V
Positioning module	AX-502HC10-0A	2-channel high-speed counter module, OC, 500 kHz, Sourcing Input (NPN)
	AX-502PU01-0A	2-channel positioning module, OC, 500 kHz, Sinking Output (NPN)
Coupler	AX-500CEC00-0A	EtherCAT coupler
	AX-500CPN00-0A	PROFINET coupler
Accessory	AX-500BC00-0A	Bus cover - End cover

1.2 DIADesigner-AX Software Overview

Conformed to IEC61131-3, DIADesigner-AX is a new programming tool for a new generation Delta PLC. With the abundant applied instructions and an adequate motion function library, DIADesigner-AX provides a friendly and multilingual programming interface for a more convenient and efficient development environment.

1.2.1 Features

DIADesigner-AX is applicable to AX-8, AX-3, and AX-5 series.

- Support all the programming languages that IEC 61131-3 defines, including LD, SFC, ST, and FBD, as well as POU, tasks and other programming language standard.
- Powerful and proven function library for various applications.
- Input assistance for the input and configuration.
- User-friendly programming with mouse and keyboard in IEC 61131-3 supported programming languages.
- 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 various kinds of automation tasks.

Two built-in configuration tools:

- Hardware Configuration: the hardware configurations and parameter managements for the system.
- Network Configuration: the network configurations and data exchange management for the system.

Providing 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 motions
- Support PLCopen POUs for add-on functions, including diagnostics, stop, and CAM controller
- Additional POUs for different tasks including monitoring dynamic data, following error, operating CAMs and CAM controllers
- Integrated graphical CAM editor with loads of configuration options
- Virtual and logical axes are supported.
- Integrated drivers for numerous Modbus and EtherCAT protocols
- Configuration of the drives as standard field devices.

MEMO

Chapter 2 Specifications and System Configurations

Table of Contents

2.1 Technical Specifications

In this section, you can find applicable technical specifications, compliances, and certifications for AX-5 series products.

2.1.1 Certifications and Standards

AX-5 series products have obtained the following certificate of compliances and met the safety, health, environment, and other requirements.

Certification	Listing mark	Standard
CE	CE	2014/35/EU: Low Voltage directive 2014/30/EU: Electromagnetic Compatibility (EMC) directive, including: EN 61131-2, EN 61010-1, EN 61010-2-201, EN IEC 61000-6-1, EN IEC 61000-6-2, EN IEC 61000-6-4, EN IEC 61000-3-2, EN IEC 61000-3-3
UL listed	UL ® CERTIFIED SAFETY US·CA E206327	UL 61010-1 UL 61010-2-201 CAN/CSA-C22.2 No. 61010-1-12 CAN/CSA-C22.2 No. 61010-2-201

2.1.2 Electromagnetic Compatibility

AX-5 series products are in conformity with IEC/EN 61131-2 (Zone B) specified electromagnetic compatibility requirements for Pulse-Shaped Disturbances, Radio-Frequency Sinusoidal Disturbances and Radio-Frequency Emission Interference.

- **Pulse-Shaped Disturbances**

Pulse-Shaped Disturbance	Test Condition
IEC 61000-4-2 Electrostatic Discharge	Air discharge: ±8 kV Contact discharge: ±6 kV
IEC 61000-4-4 Electrical Fast Transient / Burst	Power supply line: ±2 kV Signal line: ±1 kV Communication line: ±1 kV
IEC 61000-4-5 High-energy Surge Immunity	AC power supply cable: ±2 kV CM; ±1 kV DM Tested with external AC input terminals DC power supply cable: ±0.5 kV CM/DM Tested with AX-5 series DC input terminals Data cable: ±1 kV CM, cable length > 30 m Communication cable: ±1 kV CM, cable length > 30 m

- **Radio-Frequency Sinusoidal Disturbances**

Sinusoidal Disturbance	Test Condition
IEC 61000-4-3 Radio-Frequency Electro-magnetic field amplitude modulation	80 to 1000 MHz: 10 V/m 1.4 to 2.0 GHz: 3 V/m 2.0 to 2.7 GHz: 3 V/m 2.7 to 6.0 GHz: 3 V/m 80% AM 1kHz Sinusoidal
IEC 61000-4-6 Radio-Frequency Interference	Power cable: 10 V Signal cable: 10 V Communication cable: 10 V

- **Radio-Frequency Emission Interference**

Emission Interference	Test Condition
IEC 61000-6-4 Radiated Emission	30 to 230 MHz: < 40 dB (μV/m) quasi-peak 230 to 1000 MHz: < 47 dB (μV/m) quasi-peak
IEC 61000-6-4 Conducted Emission	0.15 to 0.5 MHz: < 79 dB (μV) quasi-peak < 66 dB (μV) average 0.5 to 30 MHz: < 73 dB (μV) quasi-peak < 60 dB (μV) average

2.1.3 Transport and Storage

The original packaging of AX-5 series is in compliant with IEC (EN) 61131-2 standard, in terms of transport and storage. The conditions and specifications are listed in the table below.

Condition	Specification
Free Fall	≤ 1 m
Ambient Temperature for Operation	20 to +55°C[*1] -20 to +60°C[*2]
Ambient Temperature for Transport and Storage	-40 to +85°C
Atmospheric Pressure	Operating: 1013 to 795 hPa (altitude: 0–2000 m) Storage: 1013 to 660 hPa (altitude: 0–3500 m)
Relative Humidity	5 to 95% (without condensation)
Vibration Resistance	Tested with: 5 Hz ≦ f ≦ 8.4 Hz, constant amplitude 3.5 mm. 8.4 Hz ≦ f ≦ 150 Hz, constant acceleration 1 g Duration of oscillation: 10 sweep cycles per axis on each direction of the 3 mutually perpendicular axes International Standard IEC 61131-2 & IEC 60068-2-6 (TEST Fc)

Condition	Specification
Shock Resistance	Tested with: Half-sine wave: Strength of shock 15 g peak value, 11 ms duration. Shock direction: The shocks in each in direction per axis, on 3 mutually perpendicular axes (total of 18 shocks) International Standard IEC 61131-2 & IEC 60068-2-27 (TEST Ea)

*1: AX-5 Series controller works with AX-5 Series modules.
*2: AX-5 Series coupler works with AX-5 Series modules.

2.1.4 Protection Class

The following table shows the information related to protection class for AX-5 series.

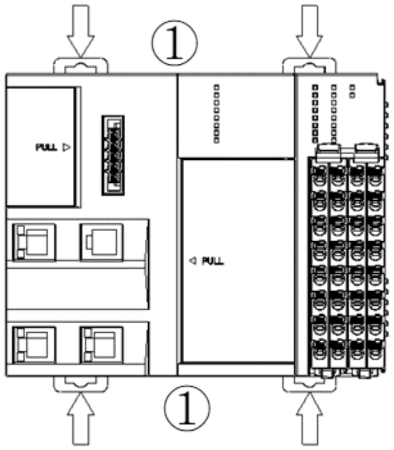
Type	Description
Insulation	Designed in accordance with IEC/EN 61131-2, IEC/EN 61010-1 and IEC/EN 61010-2-201.
DC Input Power Supply	24 VDC (20.4 VDC to 28.8 VDC) (-15 to +20%) (In accordance with IEC/EN 61131-2)
DC Power Source	The 24 VDC SELV or PELV power supply is recommended.
Pollution Degree	2 (In accordance with IEC/EN 61131-2, IEC/EN 61010-2-201)
Overvoltage Category	II (In accordance with IEC/EN 61131-2, IEC/EN 61010-2-201)
Ingress Protection (IP Ratings)	IP20 (IEC 60529) • Protection against contacting hazardous parts by operator’s fingers. • Protection against solid objects of diameter 12.5 mm or greater • No protection against ingress of water
Flammability Rating of Plastics	UL94V-0

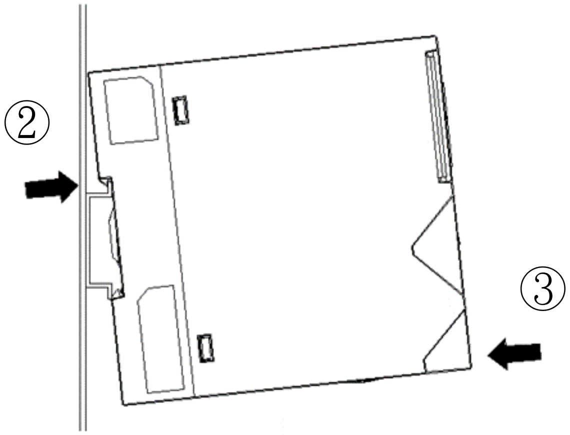
2.2 Hardware Installation

2.2.1 Installing Modules

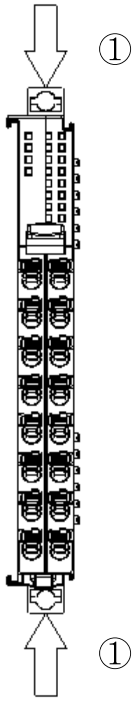
Steps for installing modules are shown below:

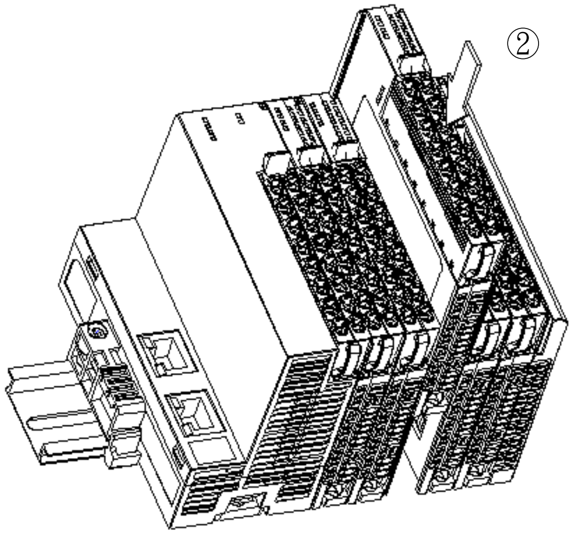
1. Push the two DIN rail clips on the top and bottom of the CPU or coupler towards the middle as shown by the arrows ① in the following figure until you hear a click. Then hang it onto the DIN rail as indicated by the arrow ② and press it towards the DIN rail as indicated by the arrow ③ until another click is heard, which means the module has been secured on the rail.



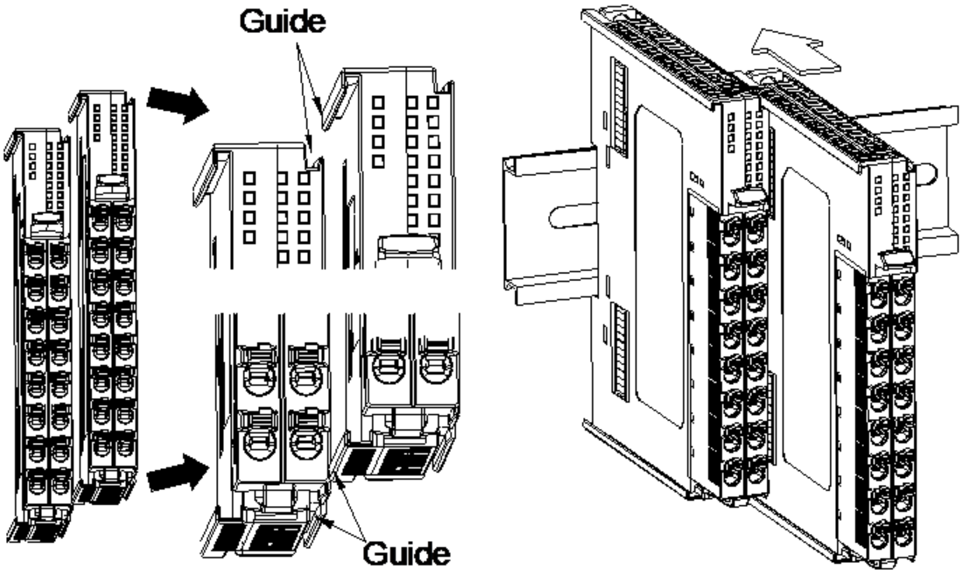


2. Before connecting respective I/O modules to the right side of the CPU or coupler in sequence, first push the upper and lower clips of each module towards the middle as indicated by the arrows ① in the following figure, and then press the module towards the DIN rail as indicated by the arrow ② until you hear a click, which means that the module has been clipped up on the rail and well connected to the CPU or the coupler.

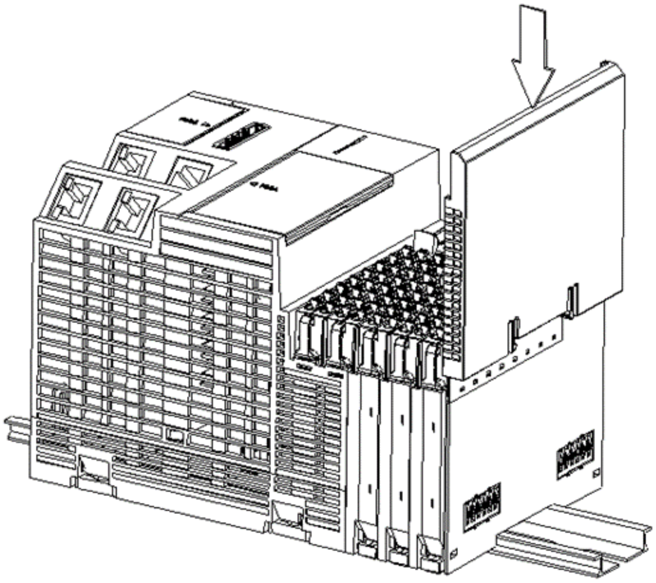




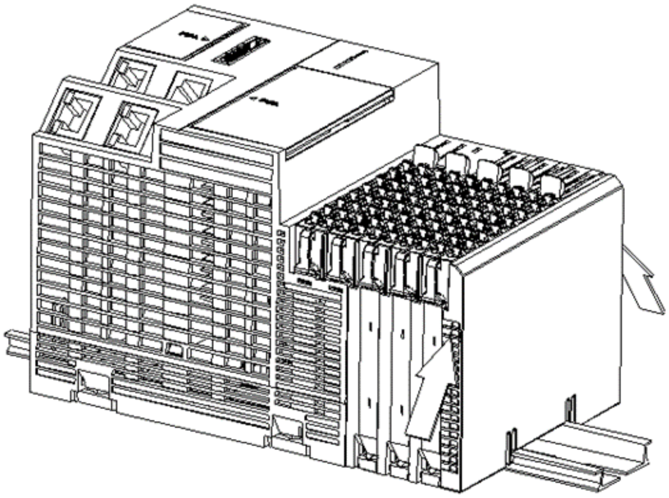
3. First align the upper and lower guides before pushing the module towards the rail when installing an I/O module.



4. Install the protective cover at the end position, just like installing a module, by aligning the cover and then pushing it towards the rail.

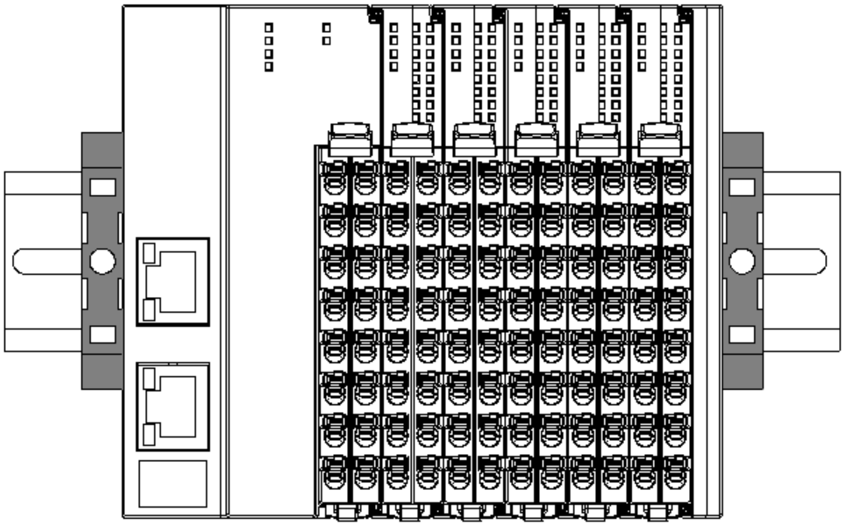


5. Remove the protective cover by taking it out in the upward direction.



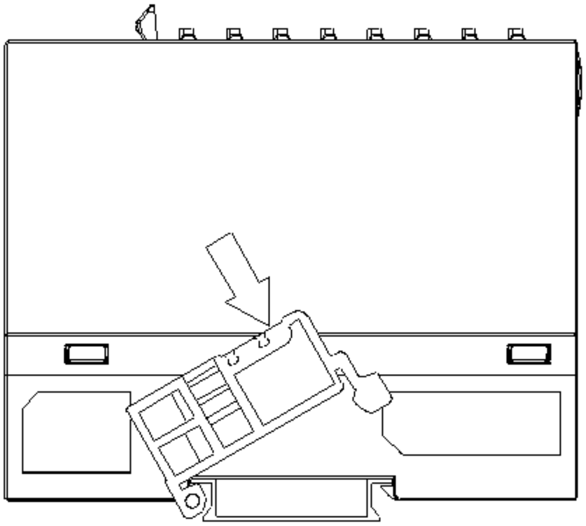
Note:

If there is a vibration source on the installation site, installing anti-vibration baffles on both the leftmost and rightmost sides of the AX-5 series modules is recommended to stabilize all modules. See the gray baffles on the two sides in the following figure.

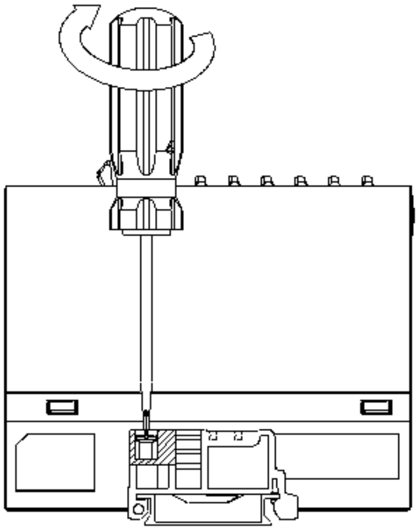


Install the baffles:

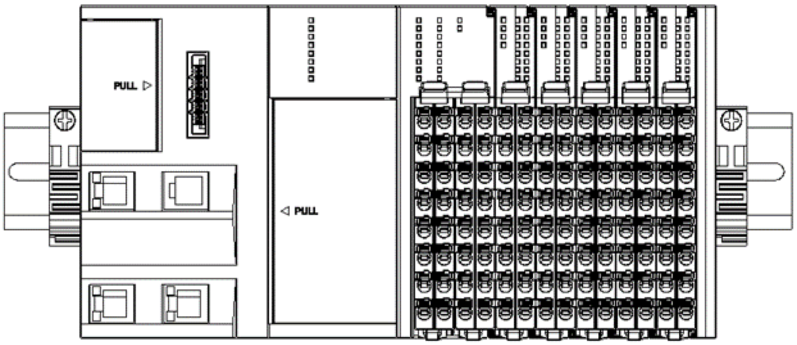
1. Hook the baffle onto the DIN rail and press it down as the directional arrow shows below.



2. Use a screwdriver to tighten the screw of the baffle after the baffle is clipped up on the rail.



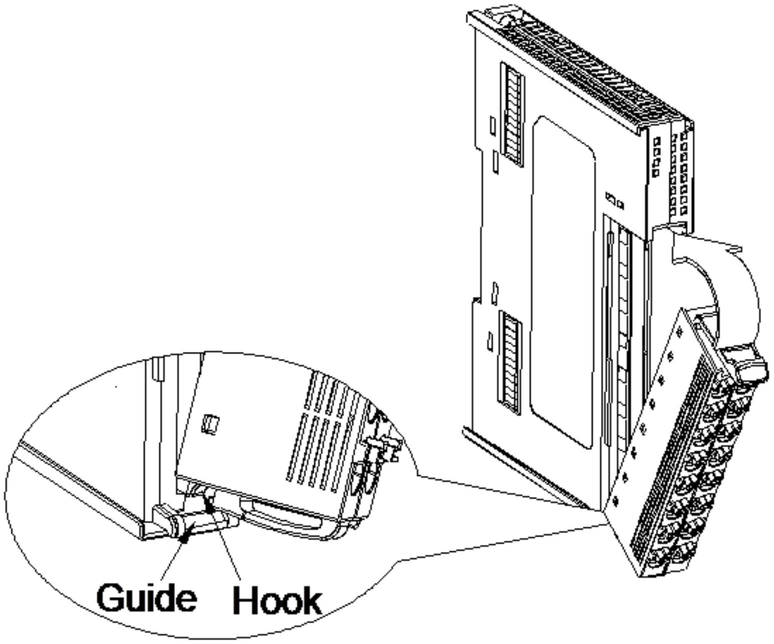
3. The completed baffle installation is as shown below.



2.2.2 Installing and Removing a Terminal Block

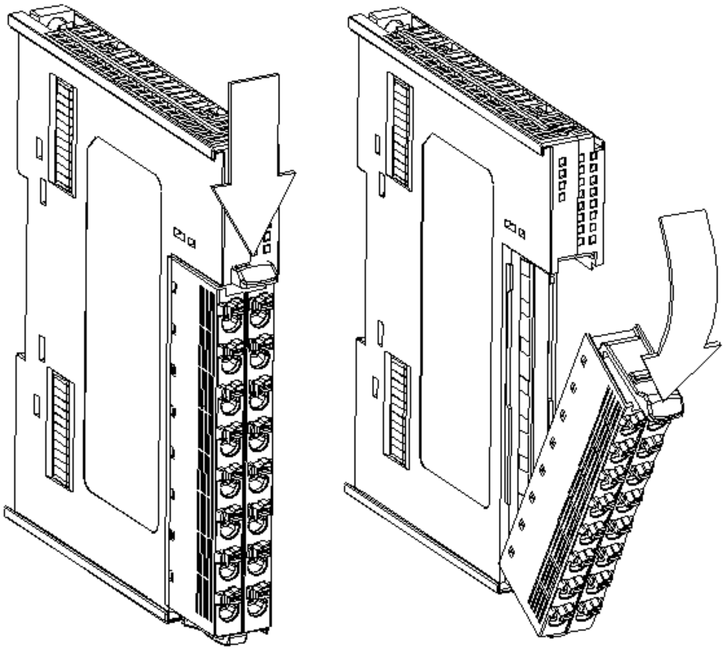
1. Installation

Hook the terminal block onto the guide at the bottom of the module and press it up into the module.



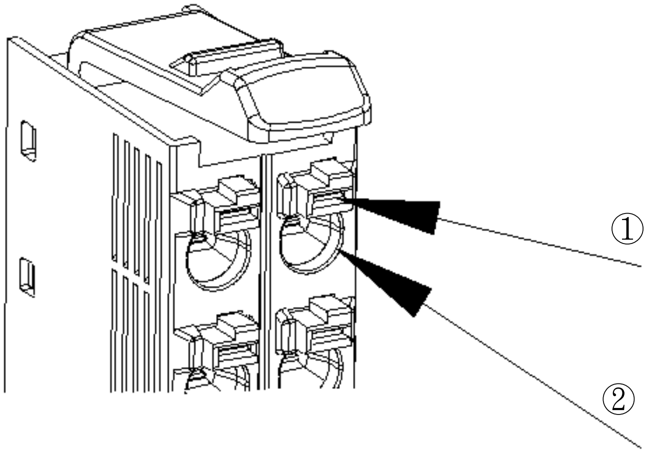
2. **Removal**

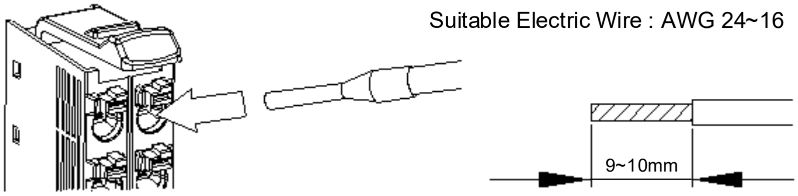
Press down the clip at the top of the removable terminal block and move it outward.



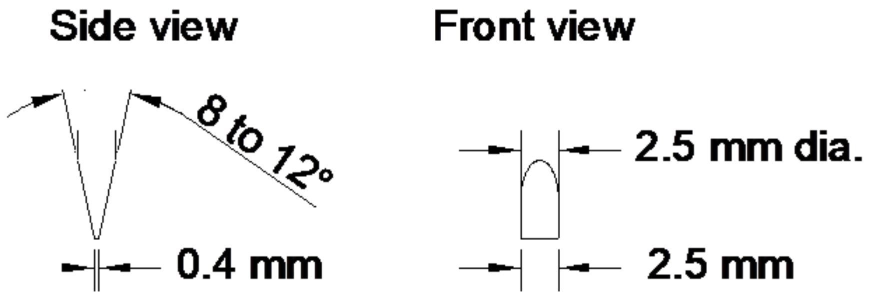
3. **Keeping the cable secured into a terminal hole of the terminal block:**

Press down the part ① above a terminal hole with a screwdriver, and then insert a suitable cable into the hole ② below.



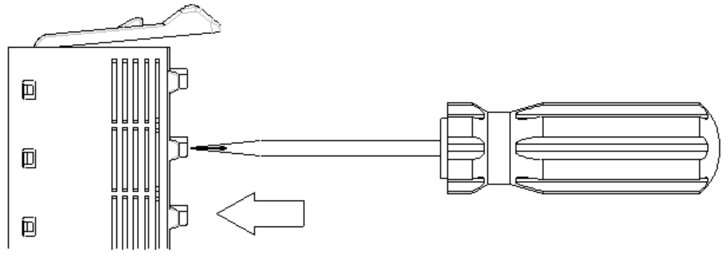


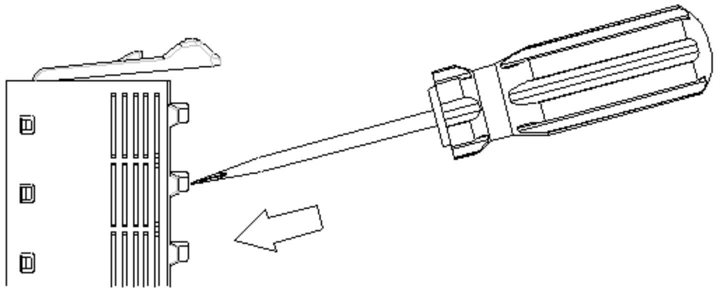
Note 1: Specifications of screwdriver



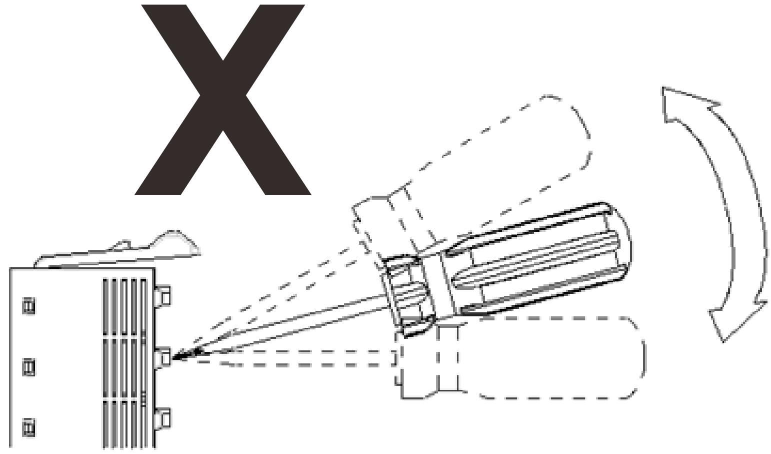
Note 2: Operation of screwdriver:

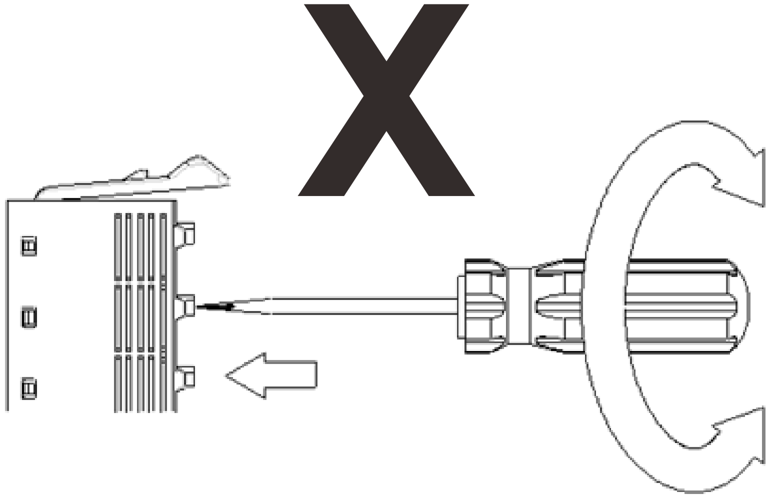
- Correct operation



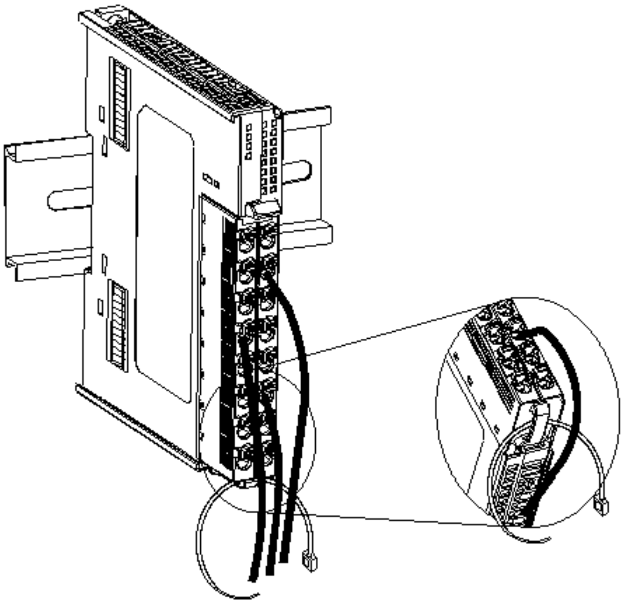


- Incorrect operation

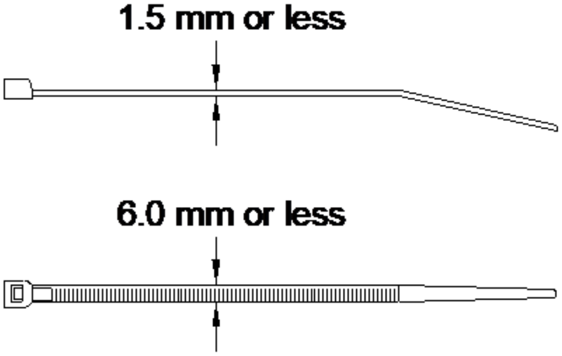




4. After wiring terminal blocks is completed, you can organize the cables by fastening them with a plastic cable tie through the cable tie hole below each terminal block.

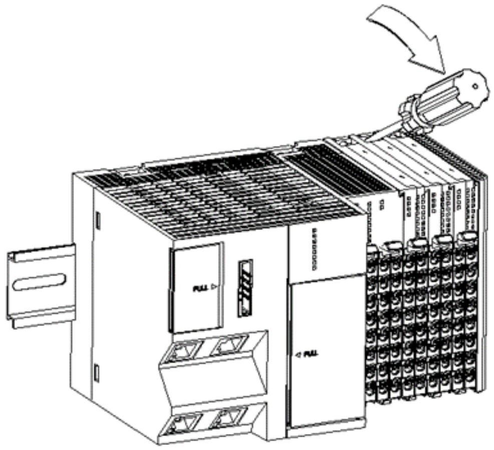


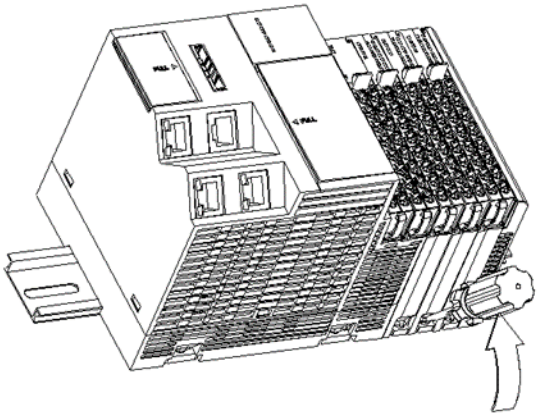
Note: Specifications of cable tie



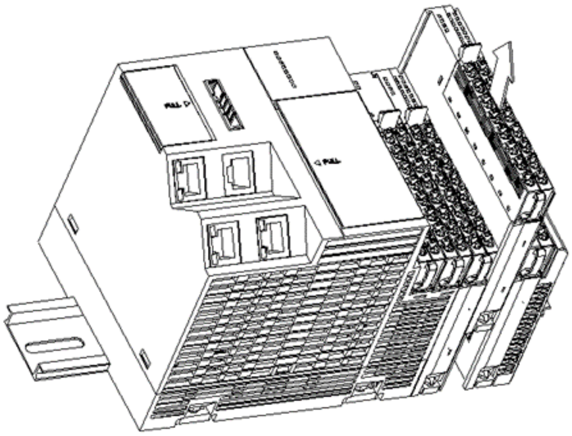
2.2.3 Replacing a Module

1. Remove the terminal block from a module, and then pull out the upper and lower clips of the module as shown below.

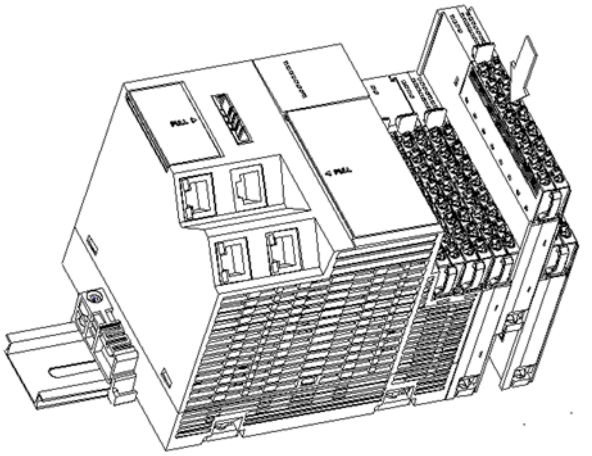




2. Remove the module.



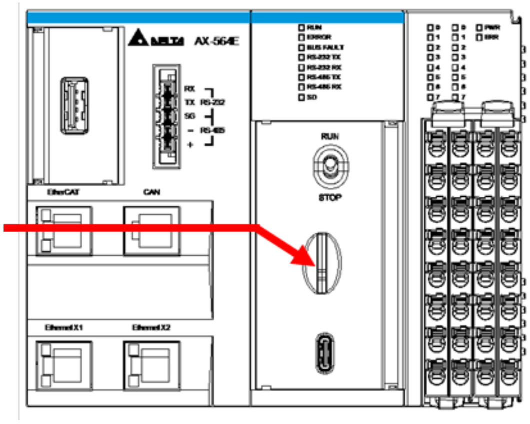
3. Slide a new module in as shown below.



2.2.4 Installing and Removing a SD Card

- **SD card slot of a PLC CPU**

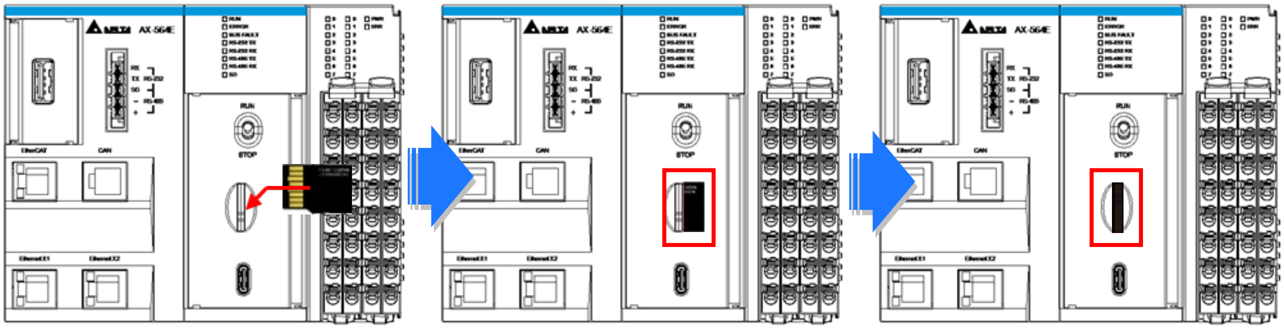
The SD card slot is on the front side of the AX-5 Series PLC CPU, as indicated by the arrows in the diagram below.



SD Card Slot

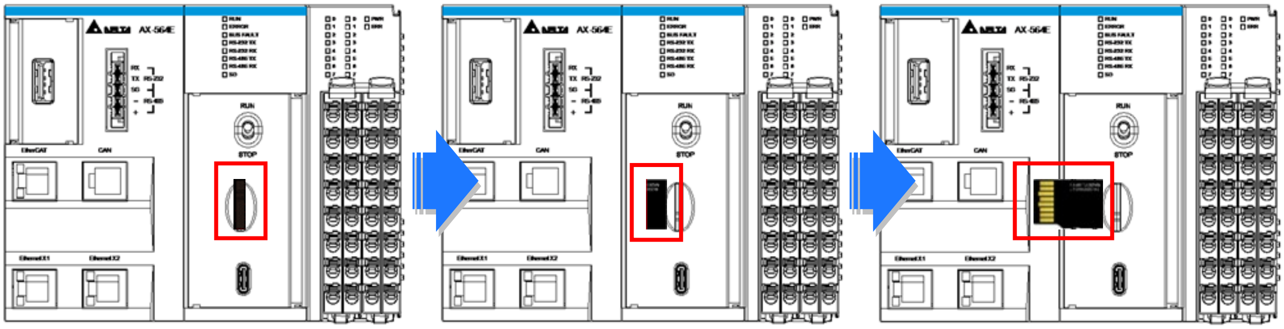
- **Installing a SD card**

Insert a SD card vertically into the SD card slot of the PLC CPU and push it in until it clicks. After successful installation, the SD card should be firmly secured. If it is still loose, it means it has not been installed correctly. Moreover, the SD card has a foolproof design. If the SD card is inserted in the wrong direction, it will not be able to be pushed to the bottom of the slot. Do not force it in to avoid damaging the device. Refer to the diagram below for the correct direction to insert the SD card.

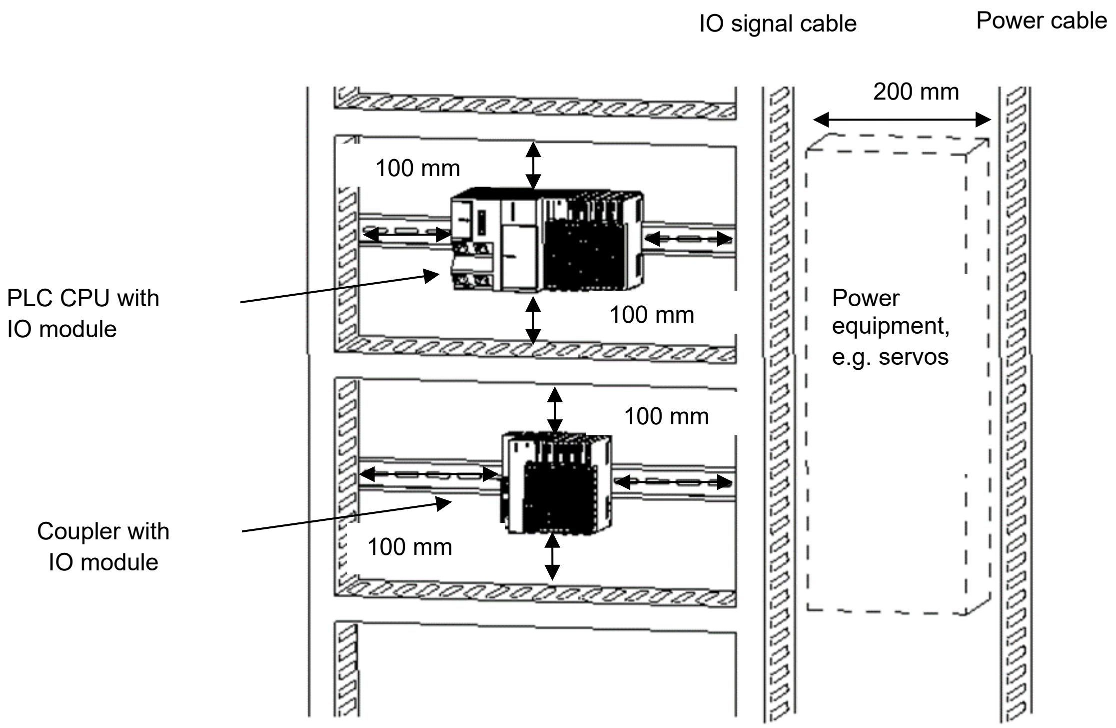


- **Removing a SD Card**

Remove a SD card by pushing it toward the bottom and then card will spring out from the slot.



2.2.5 Suggestion for Installation Inside a Control Box



2.2.6 Precautions ⚠

2.2.6.1 General Notes

- EN: System assemblers should take the responsibility for the system security of installed and assembled equipment.

  FR: Lors de l'installation, la sécurité de tout système incorporant l'équipement est de la responsabilité de l'intégrateur du système.

- EN: This product is an OPEN TYPE module and therefore should be installed in an enclosure free of airborne dust, humidity, electric shock and vibration. The enclosure should prevent non-maintenance staff from operating the device (e.g. key or specific tools are required for operating the enclosure) in case danger and damage on the device may occur.

  FR: Ce produit est un module de TYPE OUVERT et doit donc être installé dans une enceinte exempte de poussière en suspension, d'humidité, de chocs électriques et de vibrations. L'enceinte doit empêcher le personnel non qualifié d'utiliser l'appareil (par exemple, une clé ou des outils spécifiques sont nécessaires pour ouvrir l'enceinte.) afin d'éviter tout danger et dommage à l'appareil.

- EN: Only the manufacturer-specified power modules can be used for power supply. Refer to this manual or contact the manufacturer for details.

  FR: Seuls les d'alimentations spécifiées par le fabricant peuvent être utilisés pour l'alimentation électrique. Consultez ce manuel où contactez le fabricant pour plus de détails.

- EN: To well dissipate the heat produced by modules, please install the product in an open-type control box. If it is installed in a closed control box, louvers or fans should be installed on the sides of the box to ensure enough air convection for cooling.

  FR: Pour bien dissiper la chaleur produite par les modules, veuillez installer le produit dans un boîtier de contrôle ouvert. S’il est installé dans un boîtier de contrôle fermé, des persiennes ou des ventilateurs doivent être installés sur les côtés du boîtier pour assurer une convection d’air suffisante pour le refroidissement.

- EN: For the installation inside the control box, keep at least 10 cm space around the product to ensure sufficient air convection. Otherwise, it may cause the product overheating and failure.

  FR: Pour l’installation à l’intérieur du boîtier de contrôle, laissez au moins 10 cm d’espace autour du produit pour assurer une convection d’air suffisante. Sinon, cela pourrait entraîner une surchauffe et une défaillance du produit.

- EN: To prevent the product from overheating and failure, do NOT block and cover the heat dissipation holes on the top of the product.

  FR: Pour éviter la surchauffe et la défaillance du produit, ne bloquez pas et ne couvrez pas les trous de dissipation de chaleur situés sur le dessus du produit.

- EN: Do NOT place any objects on the top of the product.

  FR: Ne placez aucun objet sur le dessus du produit.

- EN: There is no battery or fuses that need be replaced inside the product.

  FR: Il n’y a ni batterie ni fusibles à remplacer à l’intérieur du produit.

- EN: Users are prohibited from disassembling the product. Only the manufacturer or its distributors can open the outer case for inspection or provide any parts.

  FR: Les utilisateurs ne sont pas autorisés à démonter le produit. Seul le fabricant ou ses distributeurs peuvent ouvrir le boîtier extérieur pour inspection ou fournir des pièces.

- EN: If the device is not used in the manufacturer-specified manner, it may weaken the protection provided by the product.

  FR: Si l'équipement est utilisé d'une manière non spécifiée par le fabricant, la protection fournie par l'équipement pourrait être compromise.

- EN: When installing or wiring, make sure that all the external power is turned off. Otherwise, it may cause an electric shock to personnel or damage to the product.

  FR: Lors de l'installation ou du câblage, il est impératif de s'assurer que toutes les sources d'alimentation externes soient éteintes. Le non-respect de cette consigne pourrait entraîner un risque de choc électrique pour les utilisateurs ou des dommages au produit.

- EN: To ensure correct and safe wiring, please check the rated voltage and terminal configuration based on product specifications for PLC wiring. Connecting to the power supply that does not match the rated value or incorrect product wiring may lead to dangerous situations such as fire or damage.

  FR: Avant d'installer ou de câbler un module, vous devez vérifier que l'alimentation électrique externe est coupée. Si l'alimentation électrique n'est pas coupée, vous risquez de recevoir un choc électrique ou d'endommager le produit.

- EN: Use special tools for folding, crimping, and welding during external wiring. Poor wiring may result in a short circuit, fire, or operational errors.

  FR: Les connexions externes doivent être serties, soudées à la presse au moyen d'outils spécifiques ou soudées correctement. Des connexions incorrectes peuvent entraîner un court-circuit, un incendie ou un dysfonctionnement.

- EN: Make sure that there are no foreign objects such as wiring residues left in each module to avoid the fire, damage, or operational errors.

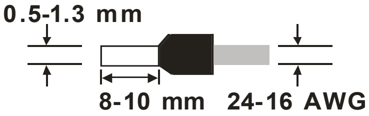
  FR: Il est crucial de s'assurer que chaque module est exempt de corps étrangers tels que des limailles de fer ou des résidus de câblage. Ces objets étrangers peuvent provoquer des incendies, des dommages ou des erreurs opérationnelles.

- EN: Please wire the input line, output line, and power line separately to prevent noise interfering with the I/O signals.

  FR: Gardez les câbles d'entrée, les câbles de sortie et le câble d'alimentation séparés les uns des autres.

- EN: Use single-core or twin-core copper cables with a diameter of 24 to 16 AWG (0.5 mm to 1.3 mm) and minimum temperature rating of 105ºC. The pin-type connector with an insulation tube is recommended as below.

  FR: Utilisez des câbles monobrins ou des câbles en cuivre à deux conducteurs d'un diamètre de 24 à 16 AWG (0,5 à 1,3 mm) et d'une température minimale de 105 °C. Un connecteur de type broche isolé est recommandé comme indiqué ci-dessous.



2.2.6.2 Control Box, Power Cable, and Ground

The PLC must be installed in a control box mainly for safety reasons. In addition, the control box also isolates electromagnetic interference generated by the PLC.

1. Control box

   - Use a conductive control box.
   - Remove the paint on the plate bolts to ensure good contact between the inner plate and the control box.
   - Ensure that the control box is properly grounded, while also avoiding insulation issues caused by paint on the grounding bolts inside the control box.
   - Minimize the size of gaps of the control box to prevent radio waves leaking from the gaps. You can also put EMI gaskets on the painted surface to avoid leaking radio waves.
   - Do NOT install near devices that are high combustion or are easily overheated.
   - Install the control box at a height of 1 to 2 meter for easier installation and operation.
   - Do NOT install any device in the control box when it is powered. The electrical circuit status inside the control cabinet should be OFF before any installation.
   - Ensure that the control box is well earthed even if there is high-frequency noise interference.

2. Connect the power cable for PLC system and the ground as described below.
    - In AX-5 series modules, the coupler module (refer to AX-5 Series Module Manual) and some power modules (refer to AX-5 Series Module Manual) are equipped with terminals for grounding.
    - You can ground any point on the aluminum rail, and the modules with ground terminals mentioned above.
    - Use the 16 AWG wire with the impedance <100 ohm for grounding.
    - You can add a magnetic core for the external AC/DC power cable to suppress radio frequency noise interference.
    - Wire the 24 VDC input and AC input cables separately.
    - Twist the ground and the power wires together; the interference flowing through the power cable is then passed to the ground. The ground and the power wires do not need to be twisted if you install a filter on the power cable.

2.2.6.3 Signal Cable, Communication Cable, and Shielded Cable

1. I/O signal cables for digital modules
    - Supports a cable length of up to 100 m with the impedance less than 2 ohms.
    - Supports shielded cables.
    - The rated current for digital output points is 0.5 A. The cable impedance of 2 ohms will cause the voltage drop of 1 V and power loss of 0.5 W on the cable.
    - Refer to Section 2.3.4 in this manual for the arrangement of terminals, LED indicators and wirings.
2. I/O signal cables for analog modules
    - Supports a cable length of 3 m.
    - Supports shielded cables.
    - Two-wire and three-wire sensors (passive sensors), and four-wire sensors (active sensors) for analog input and output. Refer to Chapter 5 of AX-5 Series Modules for the wiring of analog input/output modules.
    - Four-wire and six-wire sensors for load cell modules. Refer to Chapter 7 of AX-5 Series Modules for the wiring of load cell modules.
    - It is suggested the active sensor should use the power supply alone, not sharing the power with the AX-5 system.
    - The signal cables of the same channel for analog input and output needs to be equal in length to prevent signal interference, but the length of signal cables for different channels does not need to be the same.
3. I/O signal cables for positioning modules
    - Supports a cable length of up to 3 m with the impedance less than 1 ohm.
    - Supports shielded cables.
    - The rated current for digital output points is 30 mA.
    - The high-speed output signal from the PU/HC module should be linked to an external 470 ohm/3 W pull-up resistor.
    - Refer to Chapter 6 of AX-5 Series Module Manual for the wirings.

4. Communication cables

   Supported maximum length:

   - EtherCAT: 100 m
   - Ethernet: 100 m
   - CANopen: 100 m
   - RS-485: 100 m
   - RS-232: 3 m

5. Shielded cables

   - Shielded cables are recommended for connecting digital I/O modules and analog I/O modules including temperature modules.
   - Shielded cables are recommended for connection of communication interfaces of the PLC and the coupler module, such as EtherCAT, Ethernet, CANopen, RS-485, and RS-232 ports.
   - Use the shielded cable to do the single-point grounding.

2.3 PLC CPU Specification

2.3.1 Functional Specifications

Item				AX-516 EB0MB1T/P	AX-532 EB0MB1T/P	AX-564 EB0MB1T/P	AX-564 ELB0MB1T/P
Process time	Execution time	LD instruction		1 ns			
		Arithmetic instructions (LREAL data type)		7 ns			
Program	Program capacity	Capacity		64 MB			32 MB
	Variable memory	Retentive	Retain	1.5 MB (device memory M is included)			
			Persist	512 KB			
		Non-retentive		64 MB			32 MB
	Device memory (%M)	Size		1 MB			
Motion control	Number of controlled axes	Maximum number of controlled axes (Synchronous axis + positioning axis + virtual axis)		32 axes	64 axes	128 axes	80 axes
		Maximum number of synchronous axes (time synchronization)		16 axes (@1 ms)	32 axes (@1 ms)	64 axes (@2 ms)	8 axes (@0.5 ms)
		Maximum number of positioning axes		32 axes	64 axes	128 axes	64 axes
		Maximum number of virtual axes		32 axes	64 axes	128 axes	80 axes
		Pulse out axes		2 axes			
		Maximum number of axes for linear interpolation axis control		6 axes			
		Maximum number of axes for arc interpolation axis control		2 axes			

Item				AX-516 EB0MB1T/P	AX-532 EB0MB1T/P	AX-564 EB0MB1T/P	AX-564 ELB0MB1T/P
		Maximum number of axes for helical interpolation axis control		3 axes			
	Maximum number of axes groups			8 groups	16 groups	32 groups	4 groups
	Motion control period			The same control period as that is used for the process data communications cycle for EtherCAT.			
	CAM	**Number of CAM data points**	**Max. points per CAM table**	32,767 points			
			Max. points for all CAM tables	1,048,560 points			524,280 points
		Maximum number of CAM tables		128			64
Ethernet port	**Number of ports**			2			
	Physical media types			100BASE-TX/1000BASE-T Switch			
	Topology			Star, linear			
	Transmission rate			100/1000 Mbps			
	Cable			Category 5e or later, 100 meters (Max.)			
	Communication Protocols			ARP, IP, TCP, UDP, Modbus TCP, EtherNet/IP, OPC UA, PROFINET RT			
USB port	**Type-A**	**Number of ports**		1			
		Type		Host			
		Function		WiFi (RealTek RTL8188)			
				Bluetooth (RealTek RTL8761B), applicable to Delta Smart Viewer APP			
				Mass Storage			
	Type-C	**Number of ports**		1			
		Type		Slave			

Item			AX-516 EB0MB1T/P	AX-532 EB0MB1T/P	AX-564 EB0MB1T/P	AX-564 ELB0MB1T/P
RS-232 port	**Number of ports**		1			
	Transmission rate		9,600 / 19,200 / 38,400 / 57,600 / 76,800 / 115,200 bps			
	Serial communication format		Stop bit: 1, 2; Parity bit: None, Odd, Even; Data bit: 7, 8			
	Communication protocol		Modbus ASCII/RTU			
RS-485 port	**Number of ports**		1			
	Transmission rate		9,600 / 19,200 / 38,400 / 57,600 / 76,800 / 115,200 bps			
	Serial communication format		Stop bit: 1, 2; Parity bit: None, Odd, Even; Data bit: 7, 8			
	Communication protocol		MODBUS ASCII/RTU			
EtherCAT port	**Number of ports**		1			
	EtherCAT Master		Class B			
	Physical media types		100BASE-TX			
	Number of ports		100 Mbit/s			
	Topology		Line, daisy chain, and branching			
	Cable		Category 5e or later, 100 meters (Max.)			
	Maximum number of Slaves		4096			
	Transmission cycle		500 μs to 1,000,000 μs			
CANopen port	**Number of ports**		1			
	CANopen communication profile		DS301 v4.02			
	PDO	**Maximum number of TPDO**	256 (maximum data size:1000 bytes) *Up to 8 TPDOs for every slave			
		Maximum number of RPDO	256 (maximum data size:1000 bytes) *Up to 8 RPDOs for every slave			
Serial	**Modbus**	**Maximum number of Slaves**	32			
		Maximum data length per Slave	100 words			
	Maximum data length per instruction		1,000 bytes			
TCP	**Modbus TCP**	**Maximum number of the connections**	128 (Server + Client)			
	SOCKET	**Maximum number of the TCP connections**				

Item			AX-516 EB0MB1T/P	AX-532 EB0MB1T/P	AX-564 EB0MB1T/P	AX-564 ELB0MB1T/P
	Modbus TCP	**Maximum data length per connection**	100 words			
	SOCKET	**Maximum data length per instruction**	8 KB			
EtherNet/IP	**CIP IO Connection**	**Maximum number of the Scanner connections**	128			
		Maximum number of the Adapter connections	32 (IO connection and Tag connection included)			
		Requested Packet Interval (RPI)	1to10,000 ms (unit: 1 ms)			
		Maximum Transmission Speed	10,000 pps			
		Maximum data length per connection	Input: 0 to 500 bytes (T -> O) Output: 0 to 500 bytes (O -> T) O: Originator T: Target			
	CIP Explicit Message	**Number of connections**	12			
		UCMM supported	Y			
		Class 3 / UCMM	Get_Attribute_Single (FB) Get_Attributes_All (FB) Set_Attribute_Single (FB) Set_Attributes_All (FB)			
		Supported CIP objects	Identity, Message Router, Assembly, Connection Manager, Port, TCP/IP interface, Ethernet link, Vendor specific			
OPC UA Server	**Supported profiles and models**		PLCopen and OPC Foundation: OPC UA Information Model for IEC 61131-3			
	Endpoints and connecting ports		TCP: 4840 (Reconfigurable via configuration file)			
	Maximum number of sessions (Client)		32			
	Maximum number of monitored items per server		1000			

Item			AX-516 EB0MB1T/P	AX-532 EB0MB1T/P	AX-564 EB0MB1T/P	AX-564 ELB0MB1T/P
	Maximum number of subscriptions per session		20			
	Maximum number of sessions that can be published		20			
	Minimum sampling rate for monitored items		100			
	Conditions for variables that cannot be published		● More than three dimensional arrays ● Array of Array ● The OPC UA Stack will limit messages to about 300 kB. This is the maximum for values too. ● Pointer variables, Interface variables ● Structures containing pointers and interfaces			
	Security mode and policy		● None ● Sign - Basic128Rsa15 ● Sign - Basic256 ● Sign - Basic256Sha256 ● SignAndEncrypt - Basic128Rsa15 ● SignAndEncrypt - Basic256 ● SignAndEncrypt - Basic256Sha256 ● Basic256Rsa15			
	Application authentication	**Authentication**	X.509			
		Number of certificates that can be stored	● Trusted applications: 32 ● Issuer certificates: 32 ● Rejected applications: 32			
	User authentication	**Method of user authentication**	Username / password / Anonymous			
PROFINET	**Conformance Class**		A			
	Max. number of IO Devices Supported		64			
	Min. execution time for data exchange		1 ms			
	Max. Data Length per Transmission		Input: 1440 bytes Output: 1440 bytes			
IO Config.	**Number of IO extension modules supported**		4096 (same as the maximum number of EtherCAT Slave)			
	I/O capacity		IN: 64 KB OUT: 64 KB			

Item			AX-516 EB0MB1T/P	AX-532 EB0MB1T/P	AX-564 EB0MB1T/P	AX-564 ELB0MB1T/P
	Built-in IO	**High speed counter**	● Single phase: 4 groups (80 kHz) ● A/B phase: 4 groups (50 kHz) ● A/B/Z phase: 4 groups (50 kHz) ● A/B phase (Quadruple frequency): 4 groups (50 kHz) ● A/B/Z phase (Quadruple frequency): 4 groups (50 kHz) ● CW/CCW: 4 groups (80 kHz) ● Pulse/Direction: 4 groups (80 kHz)			
		Pulse out	● Open collector: 4 groups (80 kHz) ● A/B phase: 2 groups (50 kHz) ● CW/CCW: 2 groups (50 kHz) ● Pulse/Direction: 2 groups (50 kHz)			
Memory card	**SD card type**		Micro SD (SDXC, 64 GB max / UHS-I)			
Real-time clock			Year, Month, Date, Hour, Minute, Second, Week Data retainable for 30 days (at 25ºC / 77ºF environment)			
Accuracy of the real-time clock			Maximum deviation in seconds per month -20ºC / -4ºF: -117 seconds 25ºC / 77ºF: 52 seconds 55ºC / 131ºF: -127 seconds			
PLC Handler connection number			8			

2.3.2 Electrical Specifications (Power and Weight Included)

Model / Item	AX-516EB0MB1T/P	AX-532EB0MB1T/P	AX-564EB0MB1T/P	AX-564ELB0MB1T/P
Unit power input voltage / power	24 VDC (-15 to 20%) / 27 W			
I/O power input voltage / power	24 VDC (-15 to 20%) / under 240 W, depending on external power output			
Maximum unit power supply	2 A (10 W)			
Maximum I/O power supply	Under 10 A (240 W), depending on external power output			
Current capacity for spring-clamp terminal block	Less than 5A per terminal			
Maximum power consumption (UV/UG)	0.5A (12 W)			
Maximum power consumption (I/O power)	24 VDC / 2A			
Minimum power consumption (I/O power)	10 mA			
Isolation	No isolation between unit power and the external power source. No isolation between I/O power and the external power source. Isolation between unit power and I/O power: 500 VAC (20 MΩ minimum) Isolation among the communication ports Ethernet, EtherCAT, CANopen, RS-232/ RS-485: 500 VAC Isolation between digital signals and internal signals: 500 VAC No isolation between digital channels. No isolation between USB ports.			
Input power protection (UV, UG)	Voltage/Current Protection, Low Voltage Warning			
Unit Power Protection	Voltage/Current Protection			
Weight	560 g			
Dimension (mm)	125 (W) × 100 (H) ×105 (D)			

- **Electrical specifications for the inputs on digital input/output module. The signals passing through the inputs are 24 VDC signals.**

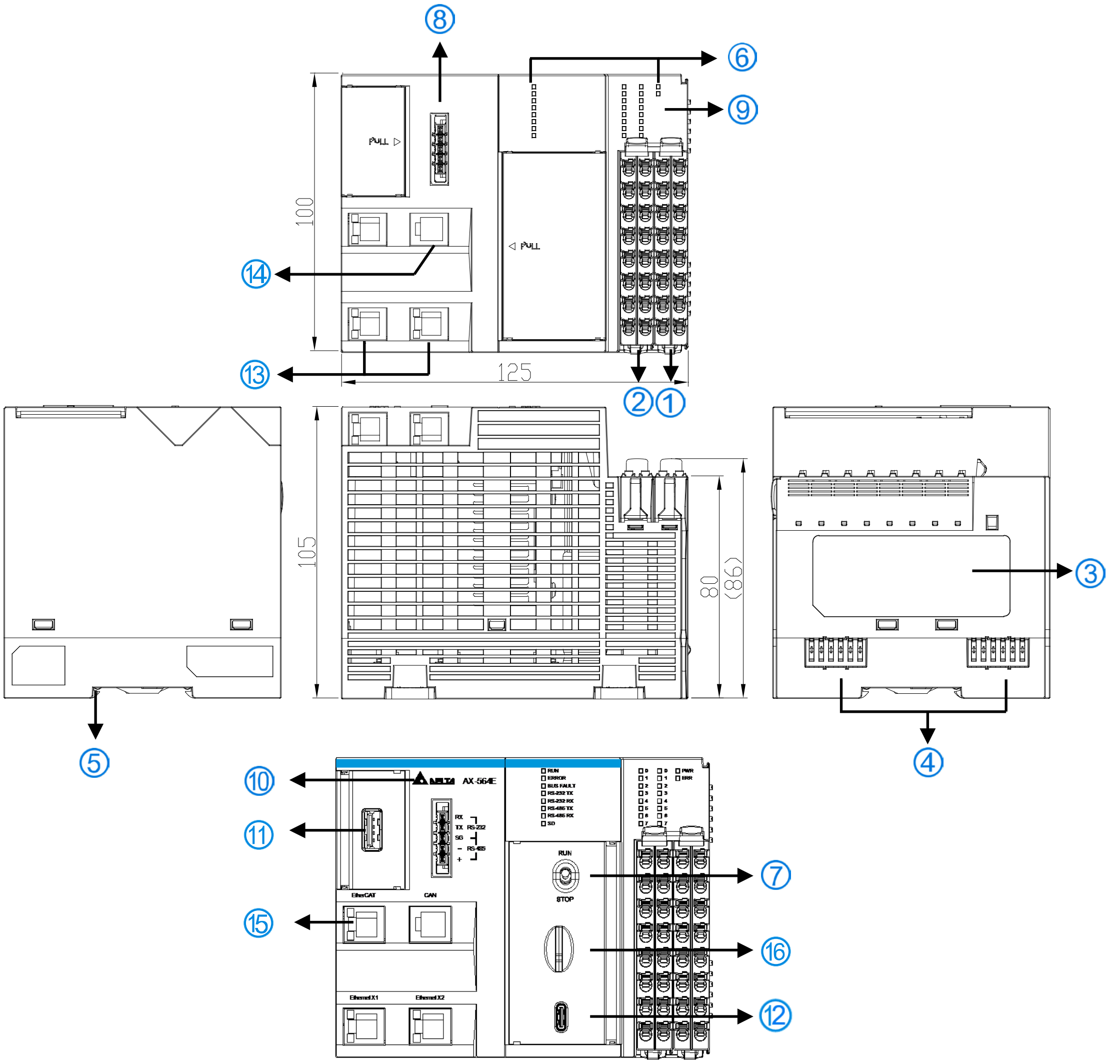
Model Item	AX-516EB0MB1T/P, AX-532EB0MB1T/P, AX-564EB0MB1T/P, AX-564ELB0MB1T/P
Number of inputs	12
Connector type	Screw-free terminal
Input type	Digital input
Input form	T: NPN (Source) P: PNP (Sink)
Input current	2.5 mA TYP (24 VDC), constant current
Voltage (ON)	11 to 30 V (EN 61131-2, type 3)
Voltage (OFF)	-3 to +5 V (EN 61131-2, type 3)
Maximum response time ON/OFF	6.25 μs
Maximum input frequency	80 kHz
Over-voltage/ over-current protection	Input flow control

- **Electrical specifications for the outputs**

Model Item		AX-516EB0MB1T/P, AX-532EB0MB1T/P, AX-564EB0MB1T/P, AX-564ELB0MB1T/P
Number of outputs		4
Connector type		Screw-free terminal
Output form		T: NPN (Sourcing) P: PNP (Sinking)
Voltage		24 VDC (-15 to +20%)
Maximum load	Resistance	0.5A / output
	Inductance	-
	Bulb	-
Leakage current		Less than 0.1 mA
Maximum Response time ON/OFF		6.25 μs
Maximum output frequency		80 kHz

2.3.3 Dimensions and Parts

- **AX-516EB0MB1T/P, AX-532EB0MB1T/P, AX-564EB0MB1T/P, AX-564ELB0MB1T/P**



Unit: mm

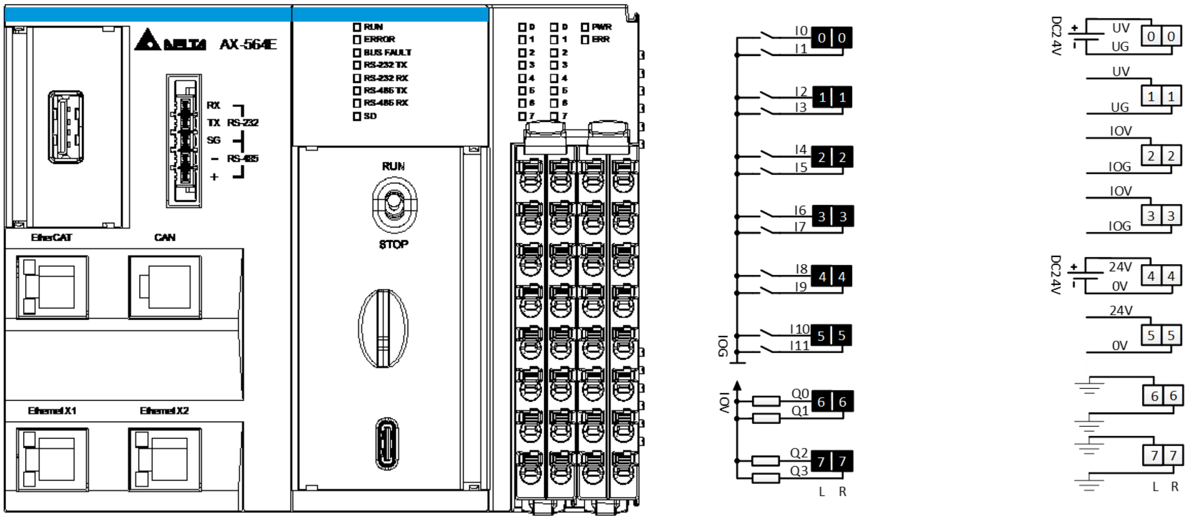
No.	Name	Description
1	Power supply	For power supply
2	Inputs / outputs	For input/output wiring
3	Label	Nameplate
4	Extension module port	For connecting the modules
5	Grounding clip	For grounding
6	PWR LED	Indicates the power status of the PLC CPU
	ERR LED	Indicates the power of the PLC CPU is abnormal
	RUN LED	Indicates the operating status of the PLC CPU
	ERROR LED	Indicates the operating of the PLC CPU is abnormal

No.	Name	Description
	BUS FAULT LED	Indicates the configuration of the PLC CPU is abnormal
	RS-232 TX LED	Indicates the communication status of RS-232
	RS-232 RX LED	
	RS-485 TX LED	Indicates the communication status of RS-485
	RS-485 RX LED	
	SD LED	Indicates if the SD card is correctly installed
7	RUN/STOP	RUN: execute the programs STOP: stop executing the programs
8	COM PORT	For RS-485 and RS-232 communication
9	Input/Output LED indicator	If there is an input signal, the input LED indicator is ON. If there is an output signal, the output LED indicator is ON.
10	Model name	Name of the PLC CPU
11	USB Type A port	USB Type A communication supports the followings: ● WiFi (RealTek RTL8188) ● Bluetooth (RealTek RTL8761B), applicable to Delta Smart Viewer APP ● Mass Storage
12	USB Type C port	USB Type C communication The USB Type C port is capable of supplying power for downloading projects and updating firmware. But, it is insufficient to maintain the AX-5 Series PLC CPU in RUN mode. Note: Ensure that the USB port on your PC can deliver at least 5V/3A.
13	Ethernet port	Ethernet Switch communication port for Ethernet communication LINK indicator (Green): ● LED ON: The network connection is established. ● LED OFF: The network connection is NOT established. ACT indicator (Orange): ● LED blinking: Data transmission (sending/receiving) LED OFF: No data transmission
14	CANopen port	For CANopen communication
15	EtherCAT port	For EtherCAT communication ● LED ON: The network connection is established. ● LED OFF: The network connection is NOT established. ACT indicator (Orange): ● LED blinking: Data transmission (sending/receiving) ● LED OFF: No data transmission
16	SD card slot	For SD card installation

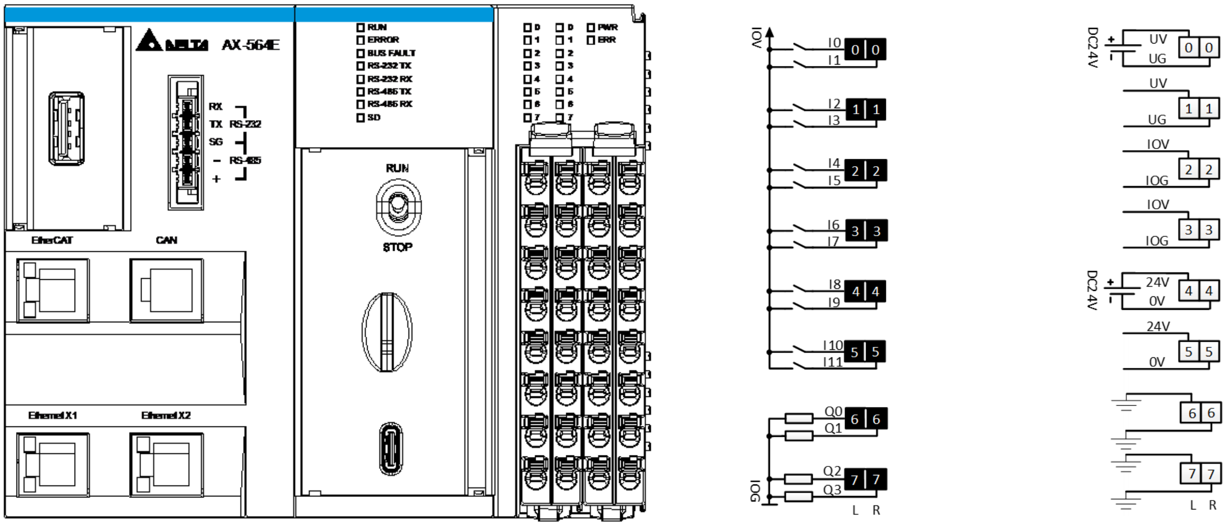
2.3.4 Arrangement of Terminals, LED Indicators and Wiring

2.3.4.1 Arrangement of PLC CPU Terminals

AX-516EB0MB1T, AX-532EB0MB1T, AX-564EB0MB1T, AX-564ELB0MB1T



AX-516EB0MB1P, AX-532EB0MB1P, AX-564EB0MB1P, AX-564ELB0MB1P



Note 1: I0 to I11 are corresponding to input points 0 to 11. Q0 to Q3 are corresponding to output points 0 to 3.

Note2: UV / UG: The power input points for the PLC and its right-side modules;

IOV / IOG: Input points shared by the right-side DIO modules;

24V / 0V: Power input points shared by the right-side DIO modules.

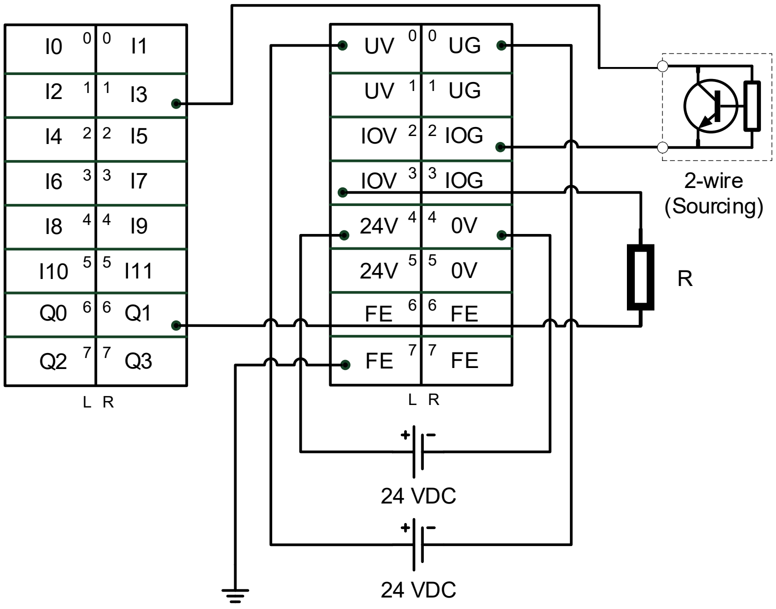
FE ⏚: The point for grounding.

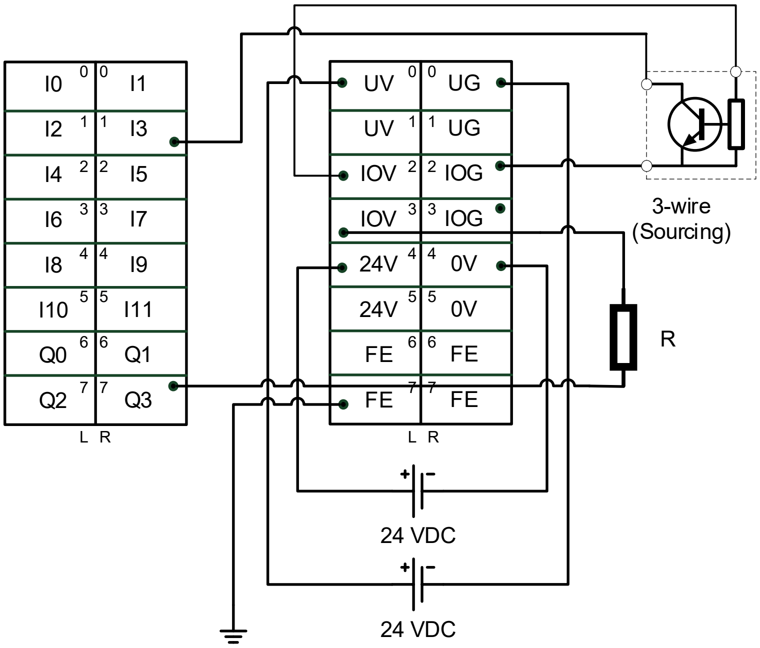
2.3.4.2 LED Indicators

Name	Color	Status		Description
RUN	Green	☐	OFF	PLC CPU is in the STOP mode.
		◩	Blinking	PLC CPU is in the Debug mode.
		🟩	ON	PLC CPU is in the RUN mode.
ERROR	Red	☐	OFF	No errors
		◩	Blinking	A minor error occurs in the system.
		🟥	ON	A severe error occurs in the system.
BUS FAULT		☐	OFF	The system is running normally.
		◩	Blinking	The actual configuration and settings of the module do not match.
RS-232 TX	Orange	☐	ON	PLC CPU does NOT send any RS-232 packets.
		◩	Blinking	PLC CPU is sending some RS-232 packets.
RS-232 RX		☐	OFF	PLC CPU does NOT receive any RS-232 packets.
		◩	Blinking	PLC CPU is receiving RS-232 packets.
RS-485 TX	Orange	☐	OFF	PLC CPU does NOT send any RS-485 packets.
		◩	Blinking	PLC CPU is sending some RS-485 packets.
RS-485 RX		☐	OFF	PLC CPU does NOT receive any RS-485 packets.
		◩	Blinking	PLC CPU is receiving RS-485 packets.
SD	Green	☐	OFF	No SD card is inserted in the PLC CPU.
		🟩	ON	SD card is inserted in the PLC CPU.
PWR	Blue	☐	OFF	No power supply to the PLC PCU
		🟦	ON	Power supply to the PLC CPU
ERR	Red	☐	OFF	Power supply to the PLC CPU is normal.
		🟥	ON	Power supply to the PLC CPU is abnormal.
Channel	Green	☐	OFF	Channel closed
		🟩	ON	Channel opened

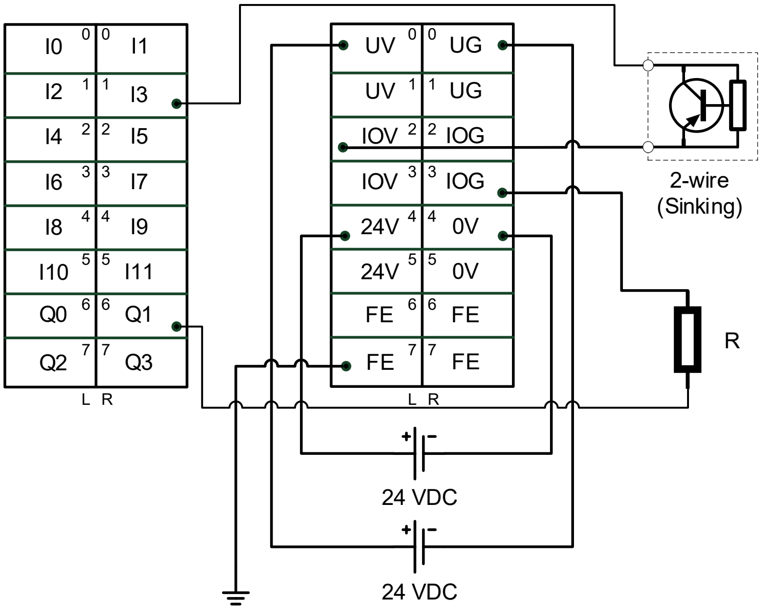
2.3.4.3 Wiring and Loop Configuration

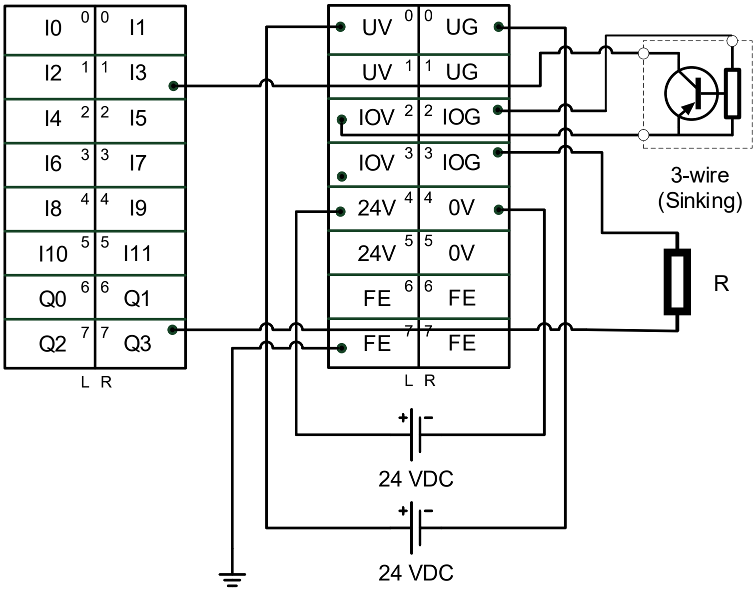
- NPN Wiring (use AX-5xxxxxMB1T as an example)



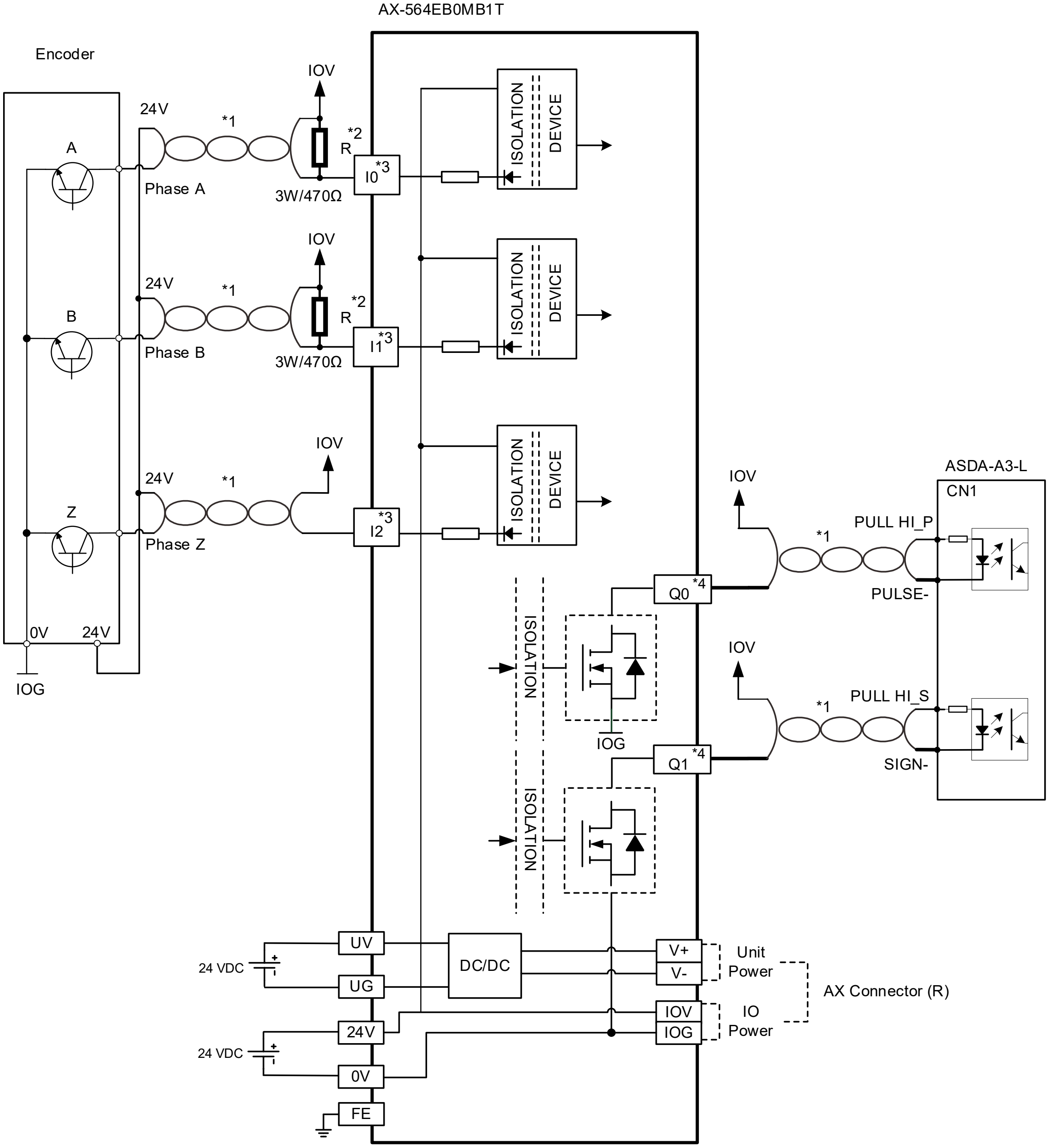


- PNP Wiring (use AX-5xxxxxMB1P as an example)





- Loop Configuration - NPN Wiring (use AX-5xxxxxMB1T as an example)



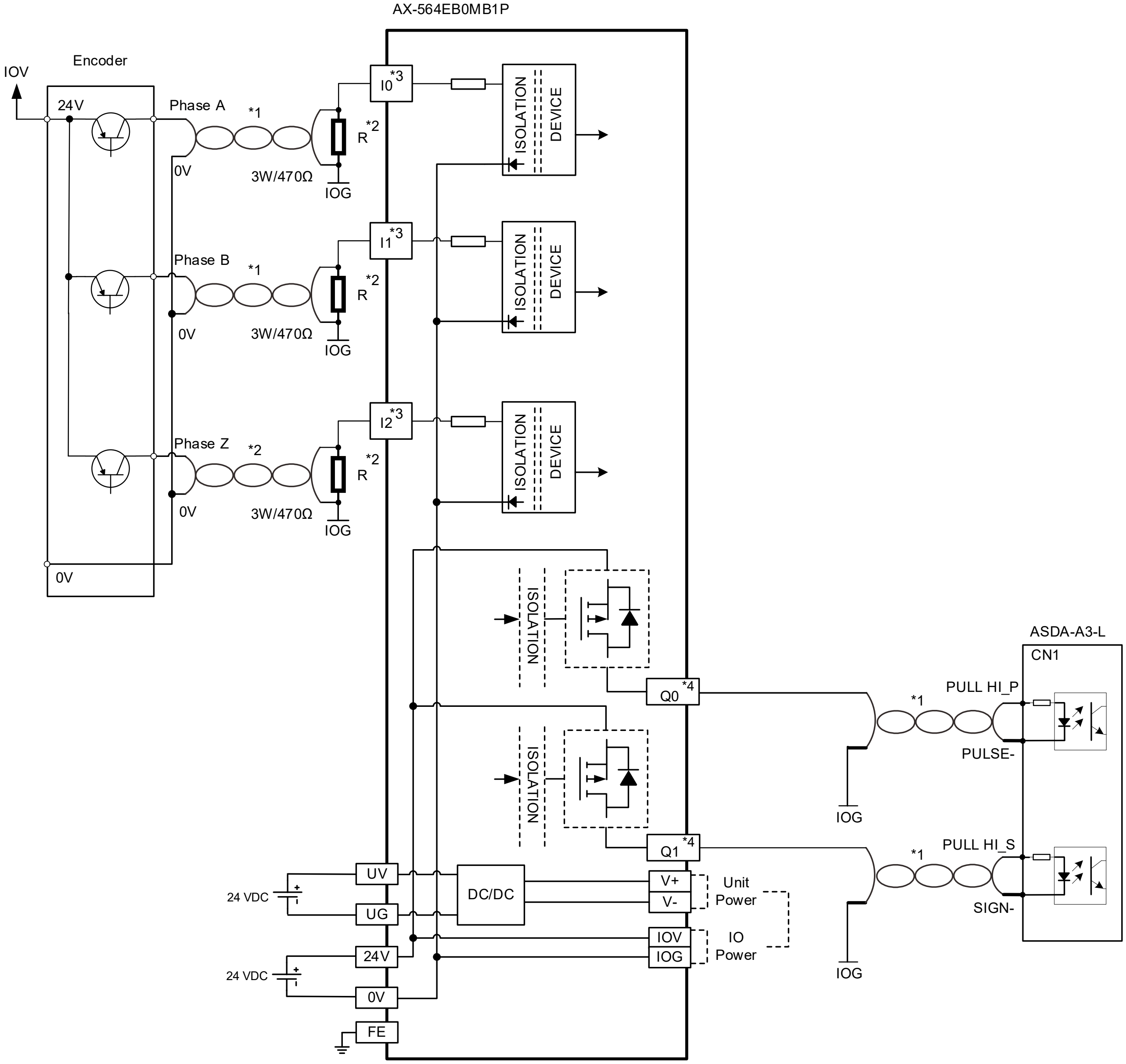
*1. Try to keep the signal cable as far away as possible from the power cable and devices that may cause noise.

*2. It is required to connect the external pull-up resistors with a rating of 3 W / 470 Ω to the input points I0 and I1.

*3. Use I4 to I11, if you need to use the second, third or the fourth groups of high-speed counters.

*4. Use Q2 and Q3 if you need to use the second set of pulse outputs.

- Loop Configuration - PNP Wiring (use AX-5xxxxxMB1P as an example)



*1. Try to keep the signal cable as far away as possible from the power cable and devices that may cause noise.

*2. It is required to connect the external pull-up resistors with a rating of 3 W / 470 Ω to the input points I0 and I1.

*3. Use I4 to I11, if you need to use the second, third or the fourth groups of high-speed counters.

*4. Use Q2 and Q3 if you need to use the second set of pulse outputs.

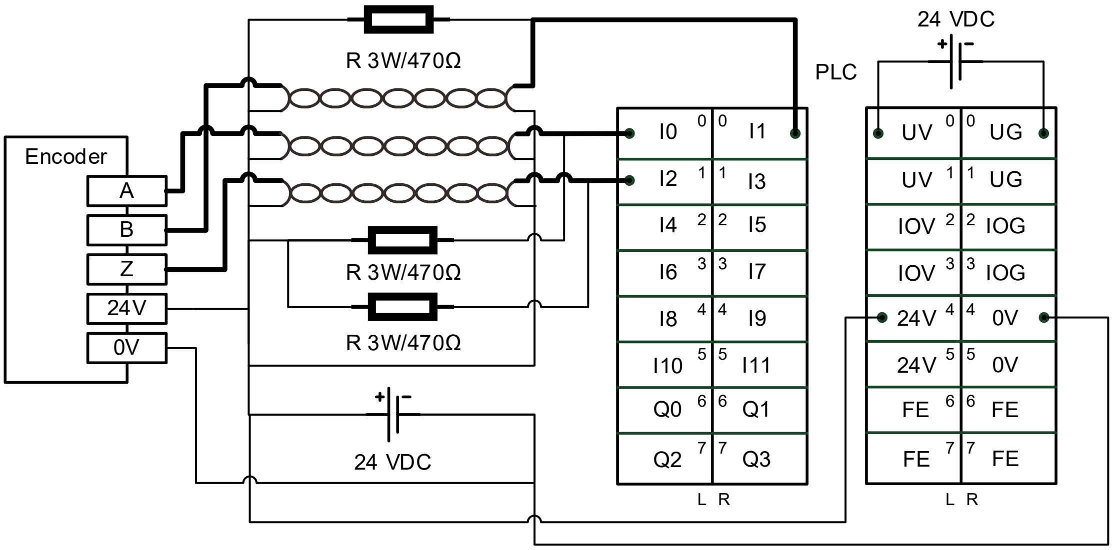
⚠	● This product shall be powered by a certified SELV/PELV power supply unit with operating temperature of minimum 55°C, which meets the requirements of IEC 61010-1:2010 or the latest edition. ● With the SELV/PELV power supply connected to 24V/0V terminals of the removable terminal block, the output rating of the power supply is 24 VDC/10A, and the power is supplied to the right-side module for I/O power use via IOV/IOG of the AX connector. There are no current limiting components on the current paths from 24V/0V to IOV/IOG. ● With the SELV/PELV power supply connected to UV/UG terminals of the removable terminal block, the output rating of the power supply is 24 VDC/1A. After the voltage conversion is made by the DC/DC unit with the current limiting function, the power is supplied to the right-side module for unit power use through V+/V - of the AX connector. ● SELV/PELV power supply units for I/O power and Unit power have different output ratings. Thus, do NOT have them share the same power supply unit. ● The output circuit and internal circuit of this product are separated only by functional insulation. Therefore, to complete the installation of the product in the final system, double or reinforced insulation should be used between the product and hazardous live parts.

2.3.4.4 Wirings of High-speed Counter and Pulse Output

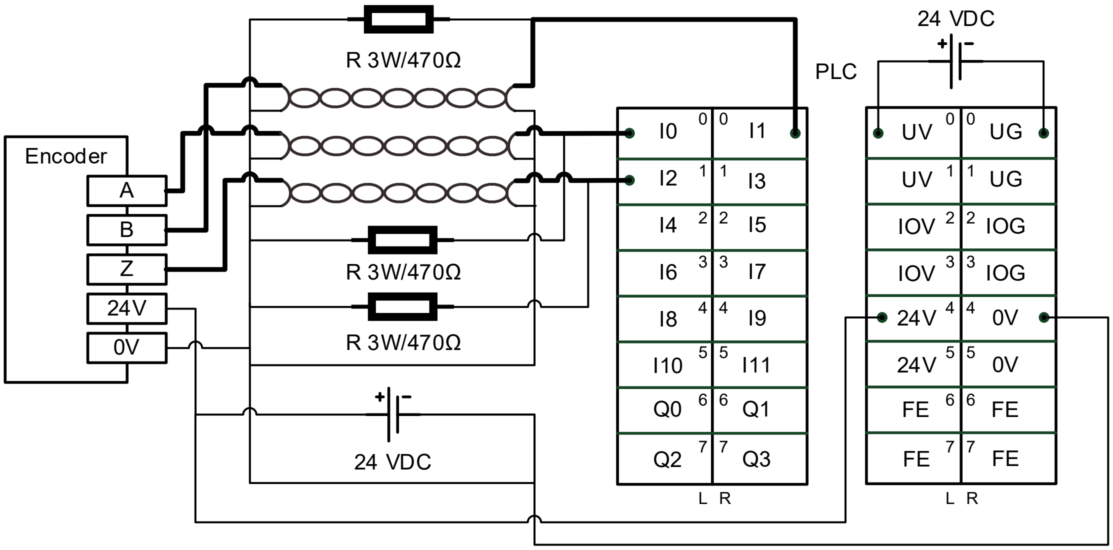
Things to note:

1. When using an open collector (NPN/PNP) output point to connect to the high-speed input of the AX PLC CPU, it is recommended to add a (pull-up/pull-down) parallel resistor group with a rating of 3 W/470 Ω.
2. When using the push-pull output to connect to the high-speed input of the AX PLC CPU, there is no need to connect a resistor in parallel.

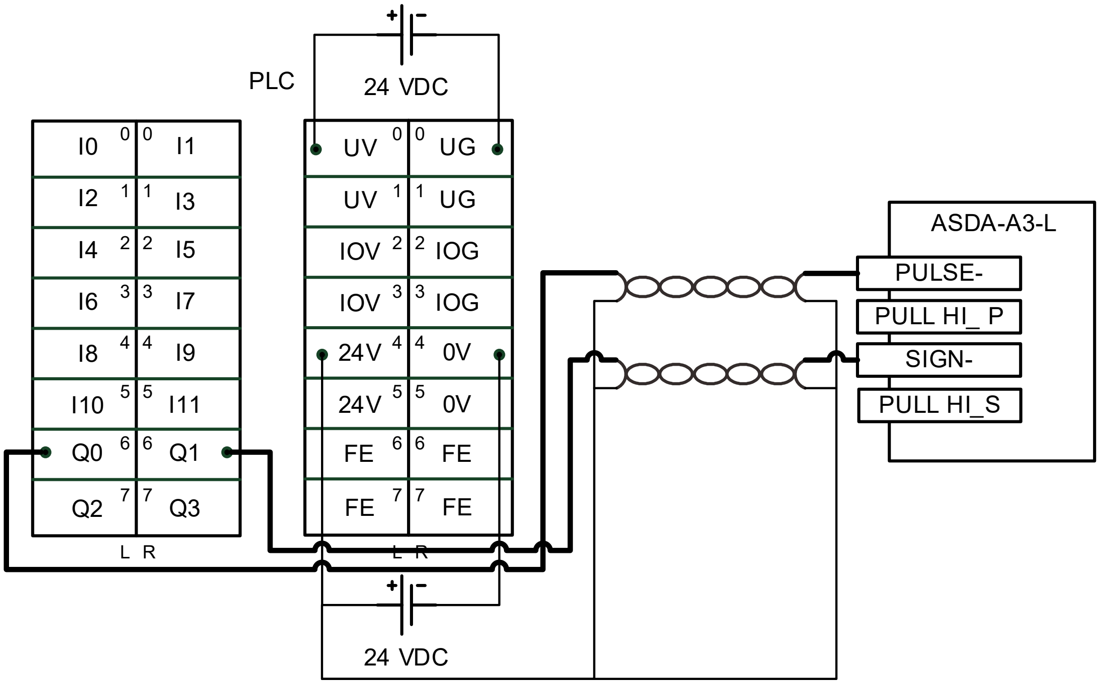
- Wiring for NPN high-speed counter (use AX-5xxxxxMB1T as an example)



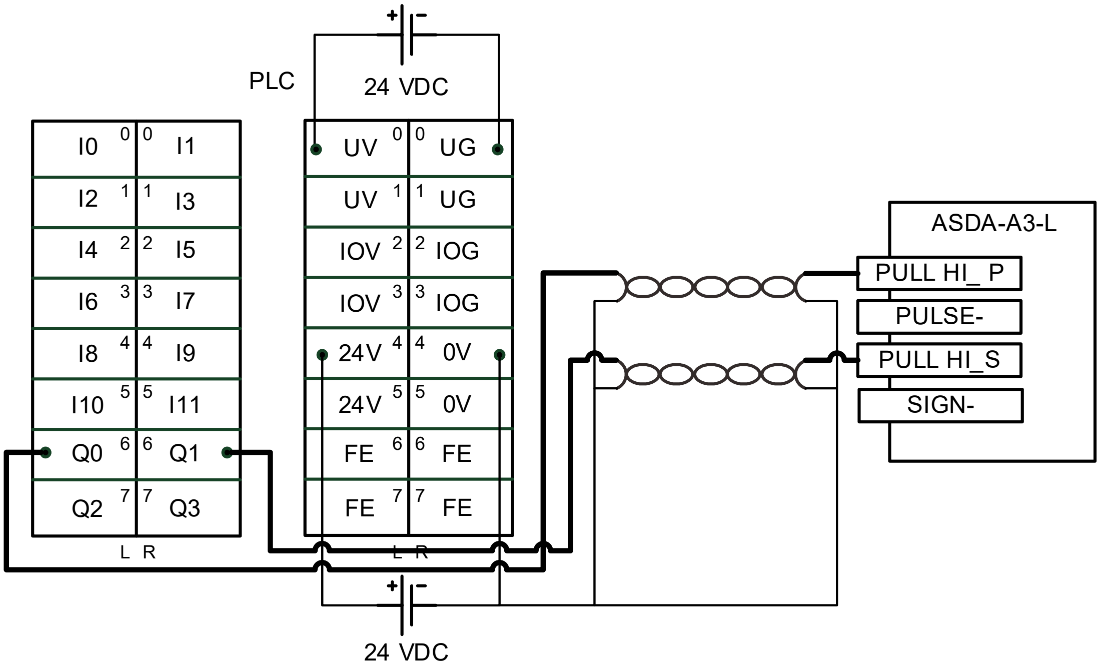
- Wiring for PNP high-speed counter (use AX-5xxxxxMB1P as an example)



- Wiring for NPN pulse output



- Wiring for PNP pulse output



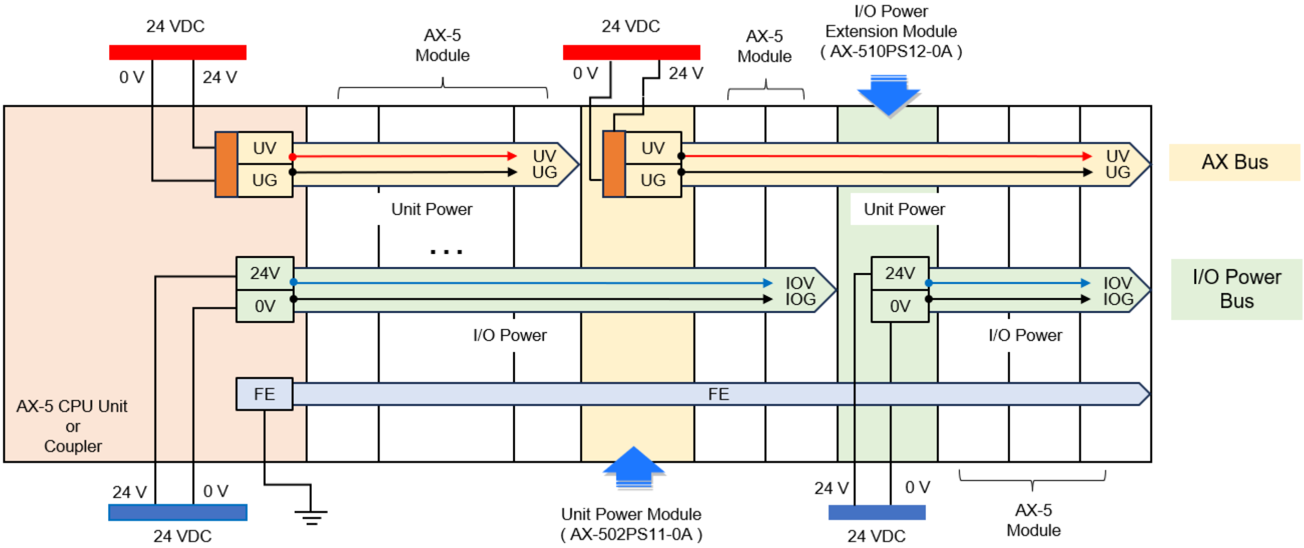
2.4 System Power Management

2.4.1 System Power Architecture

The system power architecture of AX-5 Series includes two power supply loops, AX Bus and I/O Power Bus.

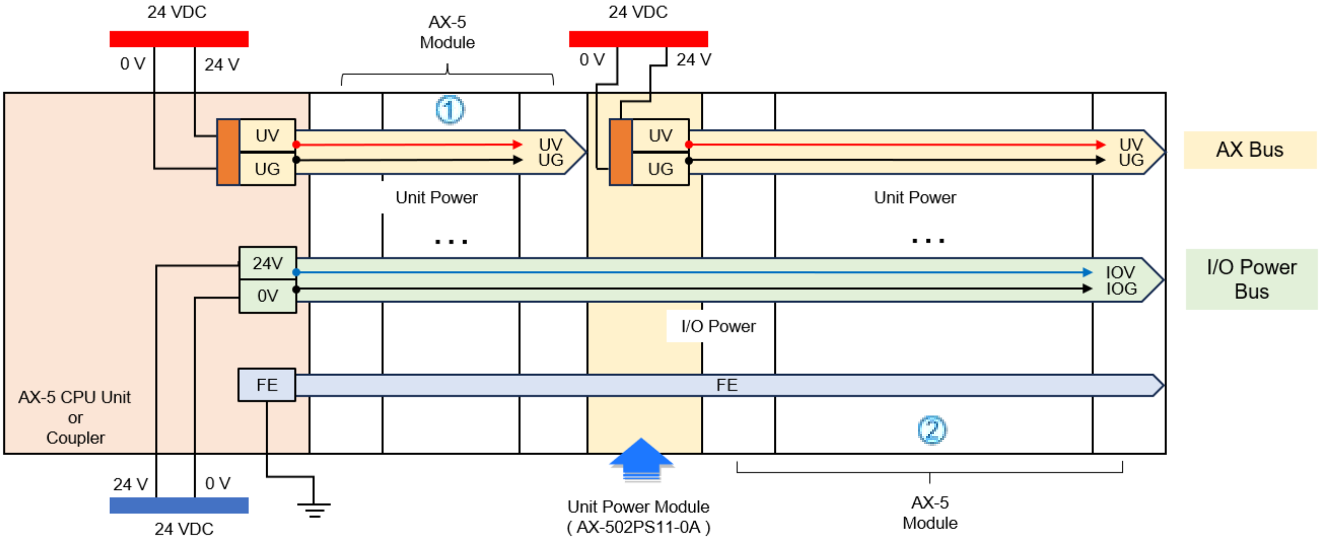
AX Bus is responsible for supplying the power required for the core operation of various modules, while the I/O Power Bus is responsible for supplying power to the I/O drive loops. And the system also includes isolated ground loops.

- Diagram of system power architecture



2.4.1.1 Design for Unit Power

The AX-5 system power architecture allows the AX-5 series PLC CPU or coupler module to deliver up to 2A of current for the AX Bus, supplying power to the AX-5 series modules. You can refer to the "AX-5 Module Current Consumption Chart" below for calculation or check the operating current consumption of the AX-5 module on the Hardware Configuration page of the DIA Designer-AX software. If the power supplied by AX-5 series PLC CPU or coupler is fallen short for the modules on AX Bus, you can add a power supply module (AX-502PS11-0A) to extend the AX Bus power supply loops and then to increase the number of AX-5 extension modules.



Note: ① and ② are supplied by AX-5 PLC CPU and power supply module AX-502PS11-0A.

- Manual calculation example

For instance, the system includes AX-504AD10-0A*4, AX054DA01-0A*4, and AX-516AP11-0A*3.
You can check the AX-5 Module Current Consumption Chart below to calculate and then conclude the maximum power consumption for the system is 4.2 A. (AX-504AD10-0A: 400 mA, AX-504DA01-0A: 500 mA, AX-516AP11-0A: 200mA).
Therefore, two sets of power supply modules (AX-502PS11-0A) are required for the system to run.
Note: The maximum current supply for each power circuit is 2A.

- AX-5 Module Current Consumption Chart

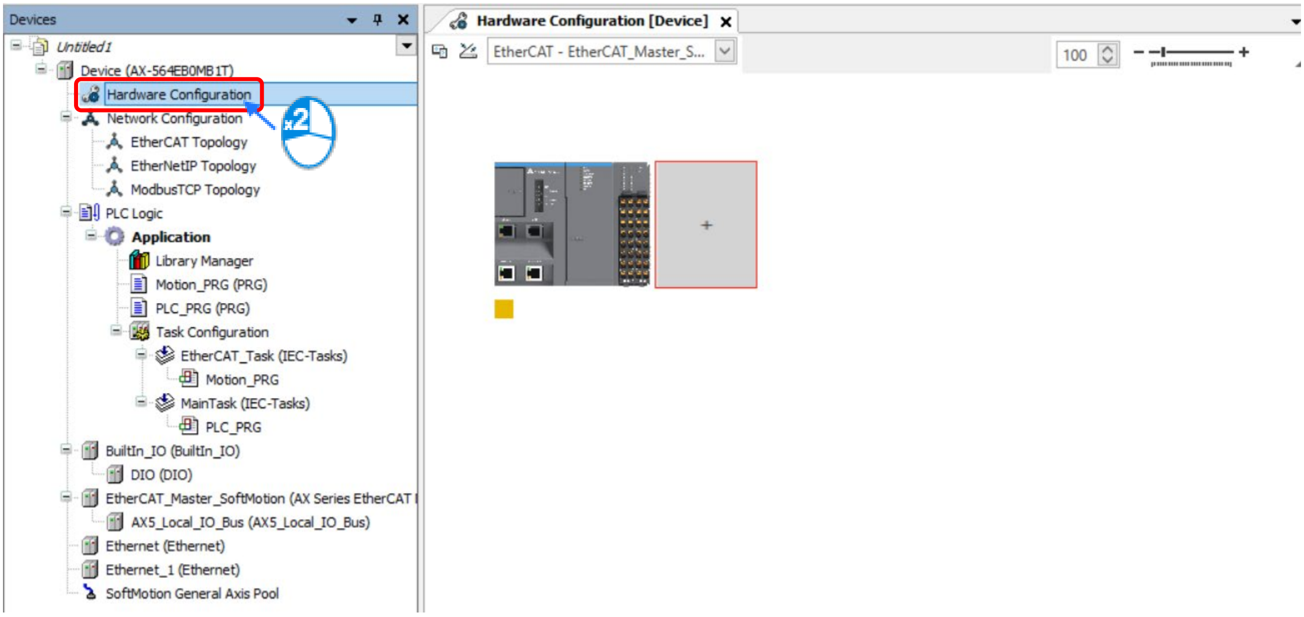
Type	Model Name	Maximum Current Consumption
Digital input / output module	AX-516AM10-0A	Less than 200 mA
	AX-516AM20-0A	
	AX-516AN01-0A	
	AX-516AN02-0A	
	AX-516AP11-0A	
	AX-516AP22-0A	
Analog input / output module	AX-504AD10-0A	Less than 400 mA
	AX-508AD10-0A	Less than 440 mA
	AX-504AD20-0A	Less than 400 mA
	AX-508AD20-0A	Less than 440 mA
	AX-504DA01-0A	Less than 500 mA
	AX-508DA01-0A	Less than 640 mA
	AX-504DA02-0A	Less than 640 mA
Temperature measurement module	AX-502TC10-0B	Less than 300 mA
	AX-504TC10-0B	Less than 360 mA
	AX-502PT10-0B	Less than 360 mA
	AX-504PT10-0B	Less than 400 mA
Loadcell module	AX-501LC10-0A	Less than 400 mA
Positioning module	AX-502HC10-0A	Less than 240 mA
	AX-502PU01-0A	Less than 240 mA

- Checked by DIADesigner-AX

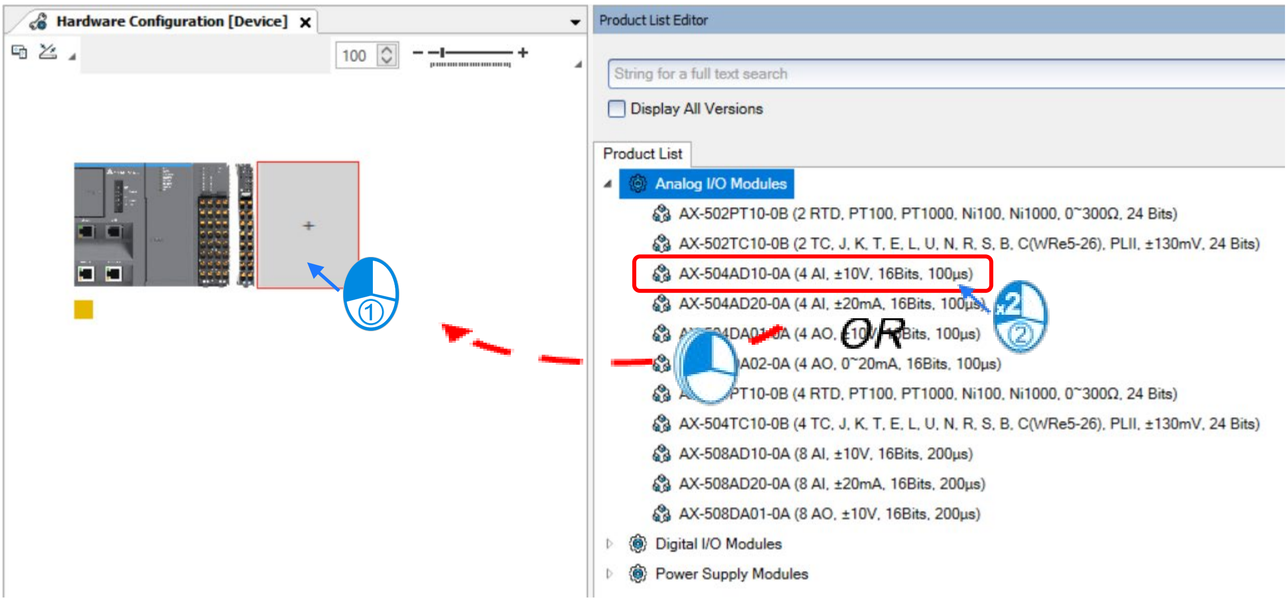
  For instance, the system includes AX-504AD10-0A*4, AX054DA01-0A*4, and AX-516AP11-0A*3.

  You can use DIADesigner-AX to check the maximum power consumption for the system. Follow the steps below:

  - Open Hardware Configuration page



  - Add the modules in: right-click the plus sign to open the Product List Editor.
  - Select the modules that you need to add. Double-click them or drag and drop them to the editing area.



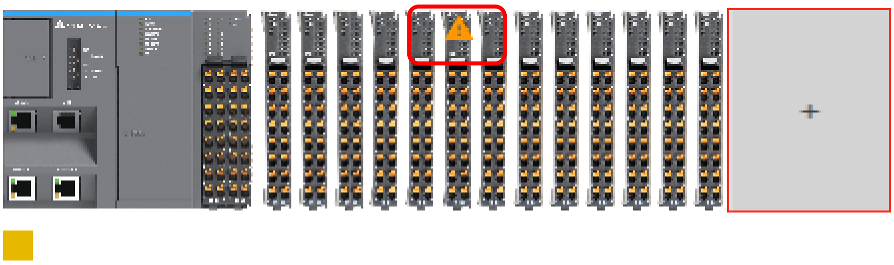
- If the power supply loop is not sufficient to supply the required power for the module, a warning message will appear, reminding users to install a power supply module.

  - ◆ When the AX-5 series PLC CPU fails to provide sufficient current for the connected modules to run.

⚠ The current consumption of installed modules has exceeded the supply of power module.

  - ◆ When the power supply module fails to provide sufficient current for the connected modules to run.

⚠ The current consumption of installed modules has exceeded the supply of power module.

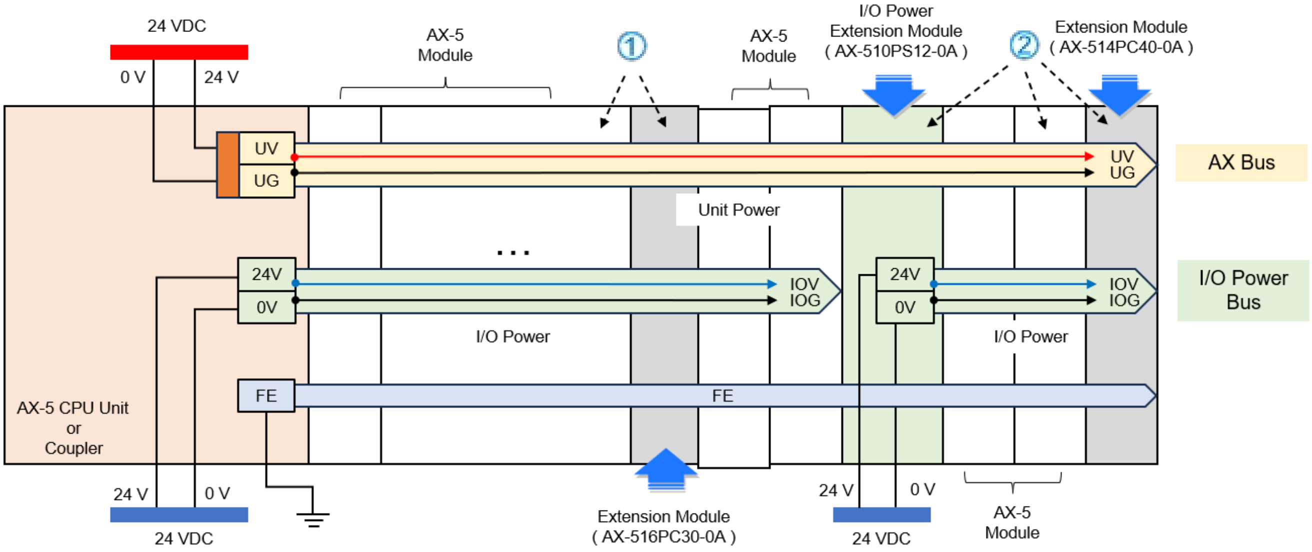


2.4.1.2 I/O Power Supply and Connector Module

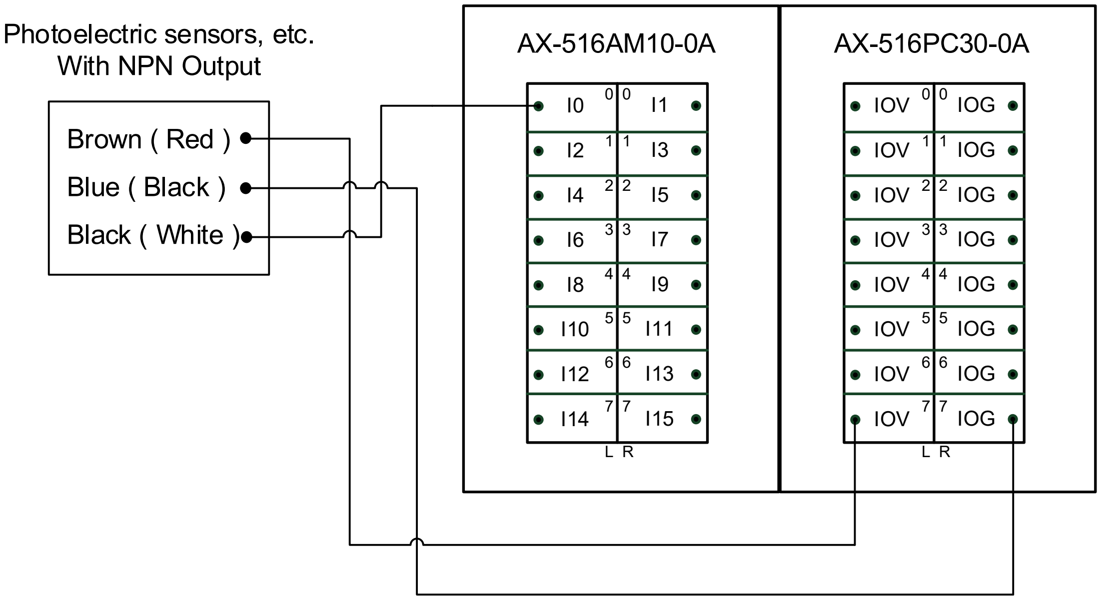
AX-5 series comes with various I/O power supply and connector modules, providing users with easier ways to design system power architecture and saving time and efforts for I/O wiring.

The example below shows using the I/O connector module (AX-516PC30-5A) to extend IOV and IOG, reducing the complexity on wiring. Use the I/O power supply module (AX-510PS12-0A) to create a separate I/O power drive loop and ensure it is isolated from the original I/O power loop to prevent any mutual interference between modules. Use the I/O connector module (AX-514PC40-0A) to provide a signal ground loop to enhance the anti-interference capability even further.

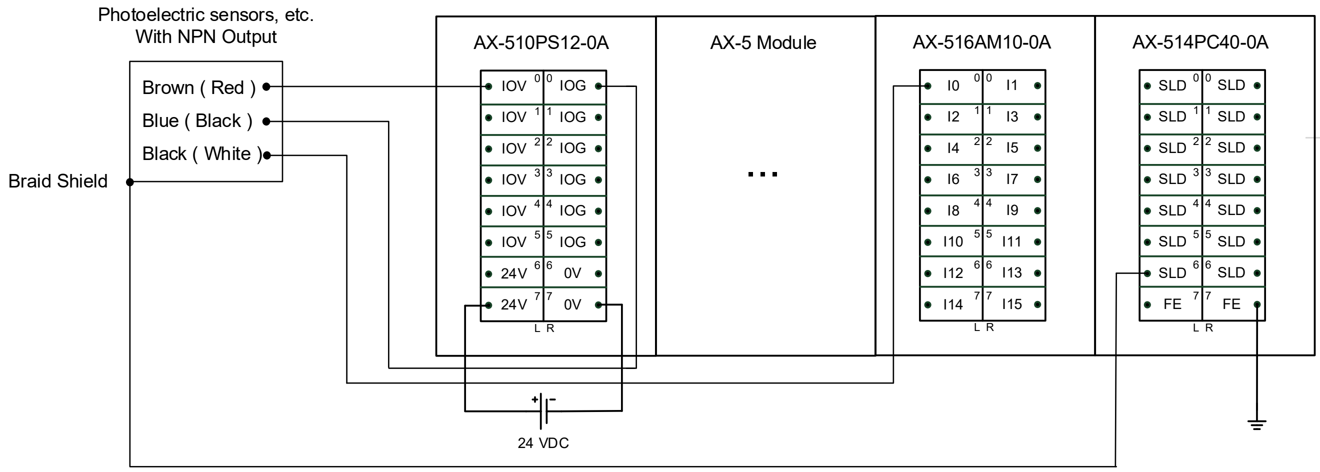
- Power supply module wiring diagram



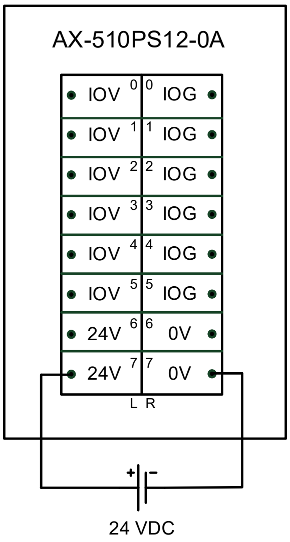
- Example of 3-wire sensor



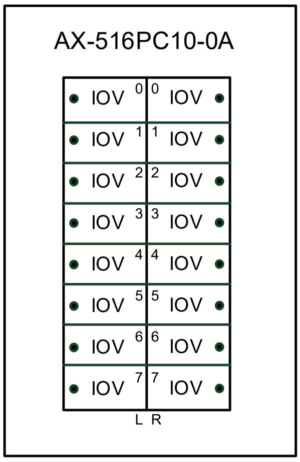
- Example of 3-wire sensor with braid shield



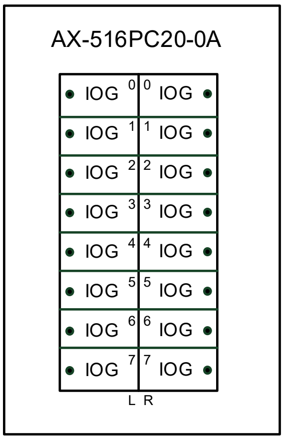
- Explanation of the I/O Power and Connector Modules
  - I/O power supply module (AX-510PS12-0A): can be used to add more IO power drive circuits.



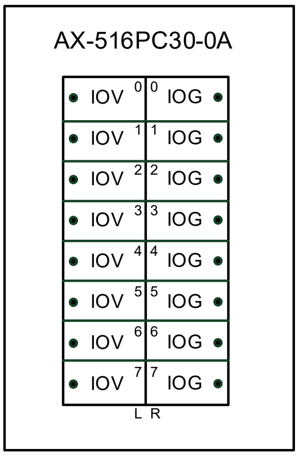
  - I/O connector module (AX-516PC10-0A): can be used to incorporate more IOV pinouts.



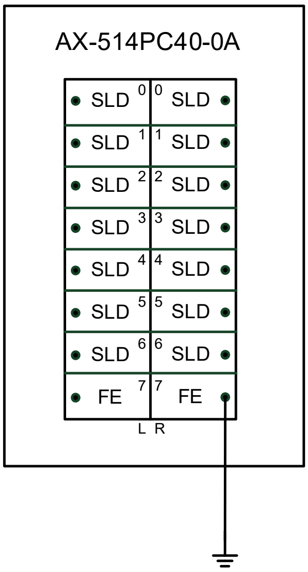
- I/O connector module (AX-516PC20-0A): can be used to incorporate more IOG pinouts.



- I/O connector module (AX-516PC30-0A): can be used to incorporate more IOV and IOG pinouts.



- I/O connector module (AX-514PC40-0A): can be used to increase more FE pinouts and more SLD pinouts for isolated grounding.



Chapter 3 Installing and Uninstalling DIADesigner-AX

Table of Contents

3.1 Installing and Uninstalling DIADesigner-AX

- **System requirements**

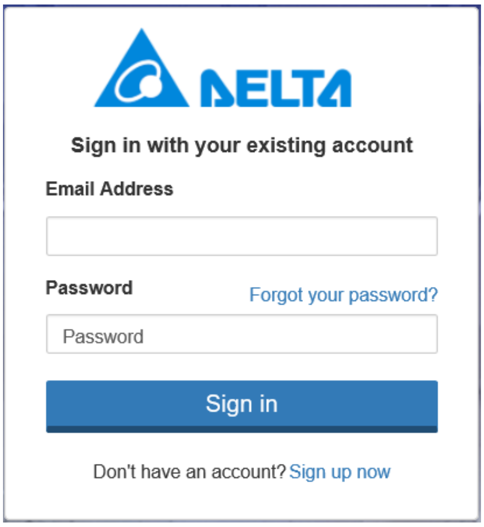
Project	System Requirement
Runtime System	DIADesigner-AX V1.60 or later
Operating System	Windows 7 / 8.1 / 10 (32/64 bits)
CPU	Intel Celeron 540 1.8 GHz (min.), Intel Core i5 M520 2.4 GHz (min.)
Memory	2 GB or above (recommend using 4 GB or more)
Hard Disk Drive	10 GB or more
Monitor	Resolution 1920 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 install .Net Framework 4.6.2

3.1.1 Installing DIADesign-AX

Before installation begins, make sure the computer used for installing DIADesigner-AX meets the minimum system requirements listed in section 3.1.

The **DIAInstaller** is a software installer which assists you to download and install **DIAStudio** software applications. You can download, install, and update products such as **DIASelector, DIADesigner, DIAScreen**, and **COMMGR**. Go to https://diastudio.deltaww.com/home/downloads to download the **DIAStudio** for **DIAInstaller**.

Before entering the download page, you need to sign in or sign up.



Sign in with your existing account

Email Address

Password

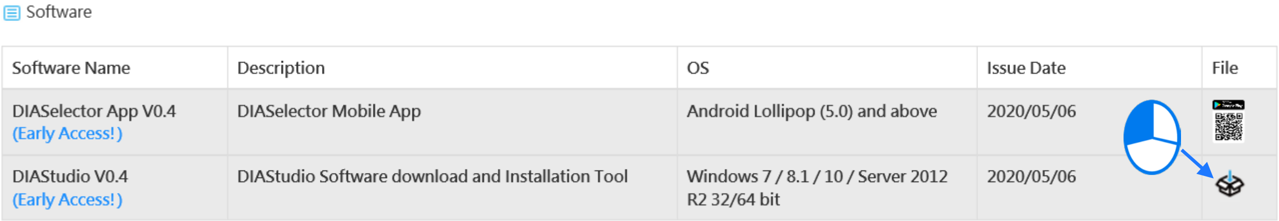
Forgot your password?

Password

Sign in

Don't have an account? Sign up now

After logging-in, click DIAStudio download button to download **DIAInstaller** as the image shown below.



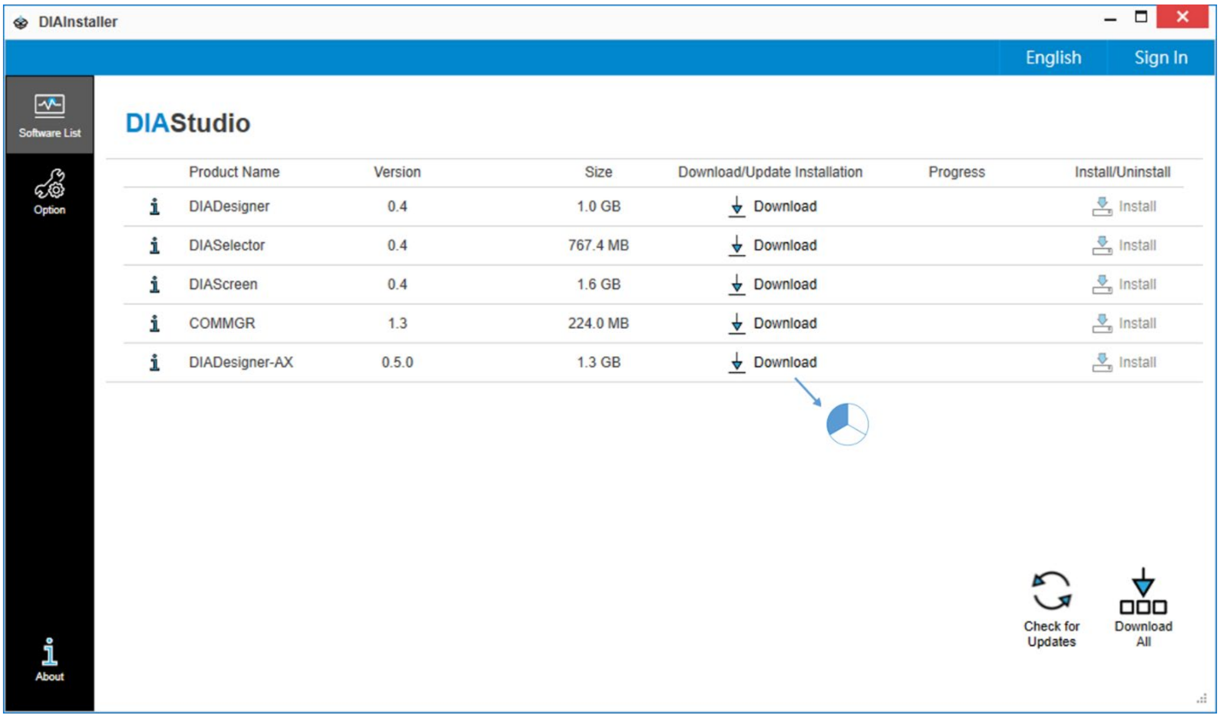
Software

Software Name	Description	OS	Issue Date	File
DIASelector App V0.4 (Early Access!)	DIASelector Mobile App	Android Lollipop (5.0) and above	2020/05/06	
DIAStudio V0.4 (Early Access!)	DIAStudio Software download and Installation Tool	Windows 7 / 8.1 / 10 / Server 2012 R2 32/64 bit	2020/05/06	

Follow the steps below for installing DIADesigner-AX.

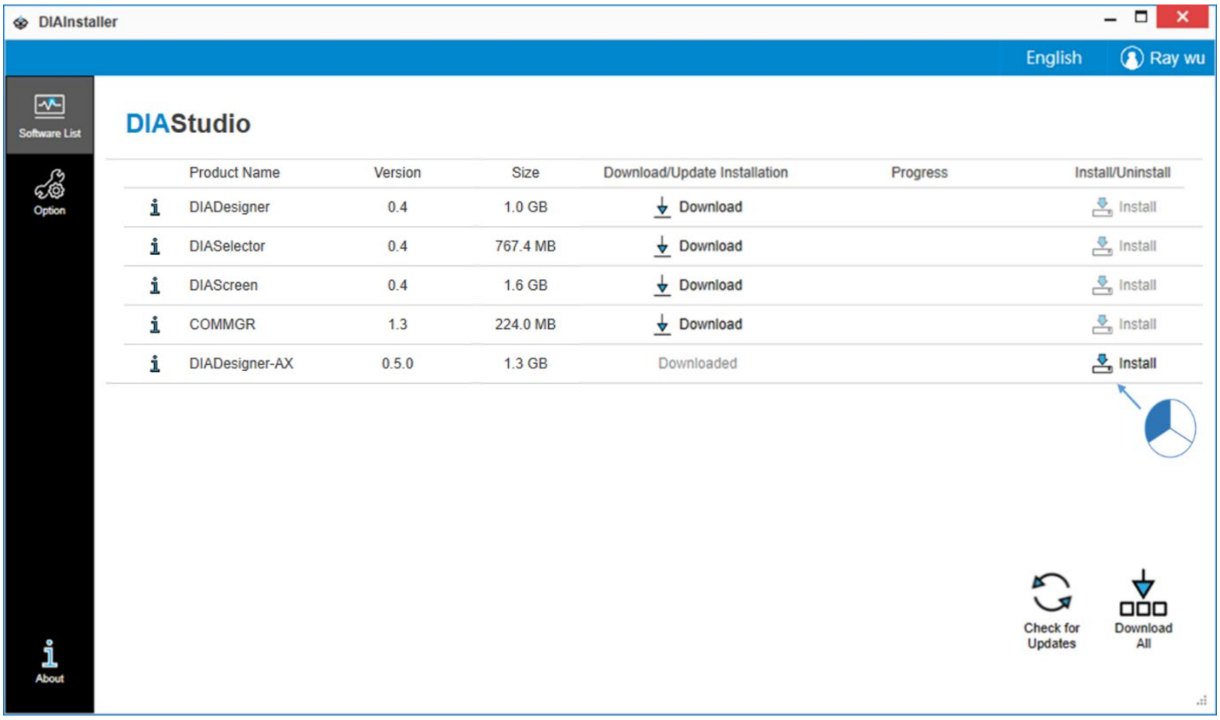
1. Double-click DIAInstaller icon to see the latest version of DIADesigner-AX.

2. Click **Download**.



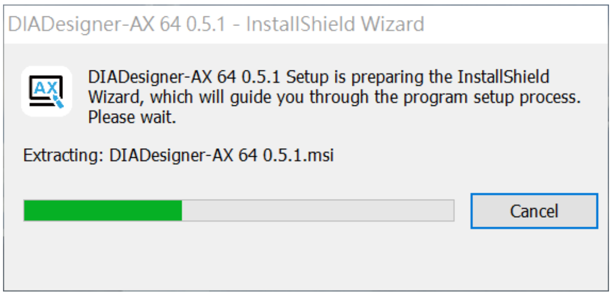
Product Name	Version	Size	Download/Update Installation	Progress	Install/Uninstall
DIADesigner	0.4	1.0 GB	Download		Install
DIASelector	0.4	767.4 MB	Download		Install
DIAScreen	0.4	1.6 GB	Download		Install
COMMGR	1.3	224.0 MB	Download		Install
DIADesigner-AX	0.5.0	1.3 GB	Download		Install

3. After that, you can see DIADesigner-AX is downloaded and grayed out. Click **Install**.



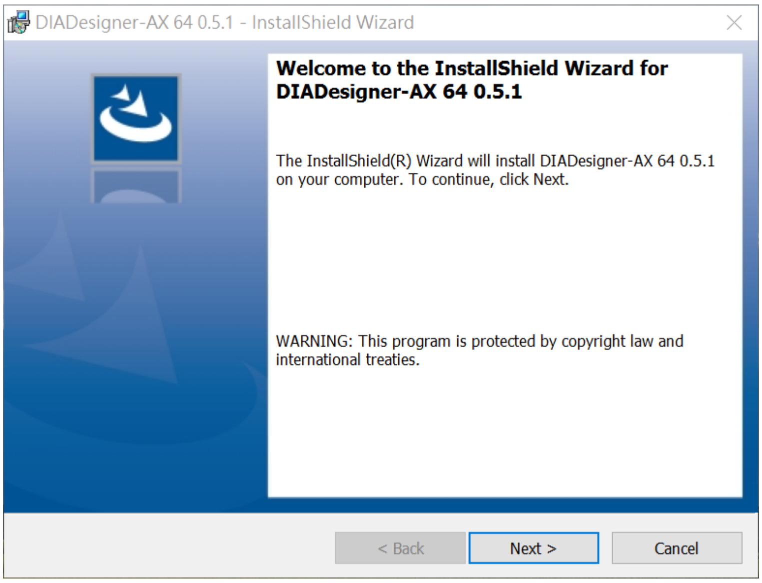
Product Name	Version	Size	Download/Update Installation	Progress	Install/Uninstall
DIADesigner	0.4	1.0 GB	Download		Install
DIASelector	0.4	767.4 MB	Download		Install
DIAScreen	0.4	1.6 GB	Download		Install
COMMGR	1.3	224.0 MB	Download		Install
DIADesigner-AX	0.5.0	1.3 GB	Downloaded		Install

4. An **InstallShied Wizard** shows up and starts installing. Click **Next**.



DIADesigner-AX 64 0.5.1 Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Extracting: DIADesigner-AX 64 0.5.1.msi

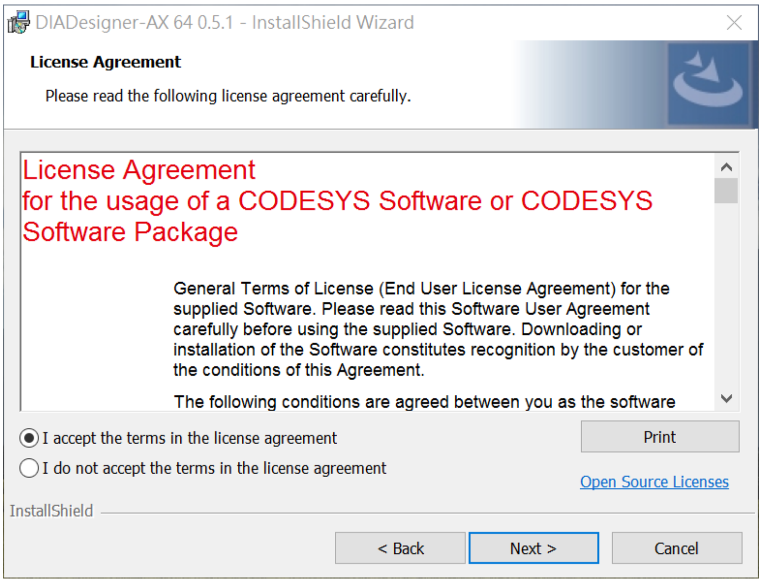


Welcome to the InstallShield Wizard for DIADesigner-AX 64 0.5.1

The InstallShield(R) Wizard will install DIADesigner-AX 64 0.5.1 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

5. The window of License Agreement shows up. Select "I accept the terms in the license agreement" and then click **Next**.



License Agreement

Please read the following license agreement carefully.

License Agreement
for the usage of a CODESYS Software or CODESYS Software Package

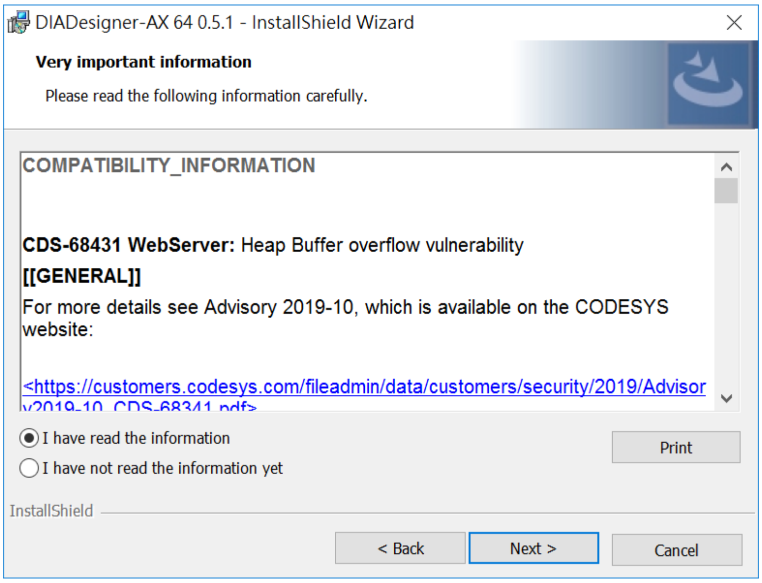
General Terms of License (End User License Agreement) for the supplied Software. Please read this Software User Agreement carefully before using the supplied Software. Downloading or installation of the Software constitutes recognition by the customer of the conditions of this Agreement.

The following conditions are agreed between you as the software

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

6. After that, a window of Very important information shows up. Select “I have read the information.” after reading the information and then click **Next**.



DIADesigner-AX 64 0.5.1 - InstallShield Wizard

Very important information

Please read the following information carefully.

COMPATIBILITY_INFORMATION

CDS-68431 WebServer: Heap Buffer overflow vulnerability

[[GENERAL]]

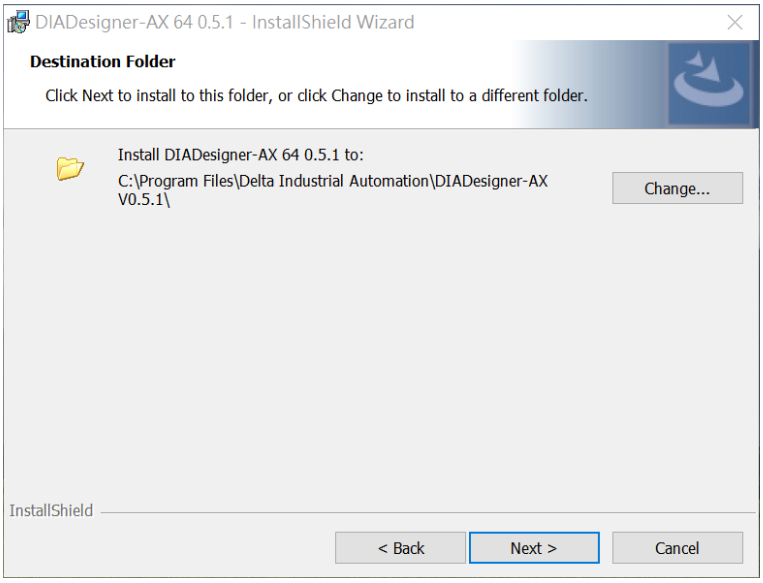
For more details see Advisory 2019-10, which is available on the CODESYS website:

<https://customers.codesys.com/fileadmin/data/customers/security/2019/Advisor

● I have read the information

○ I have not read the information yet

7. Leave the default path unchanged. Click **Next**.



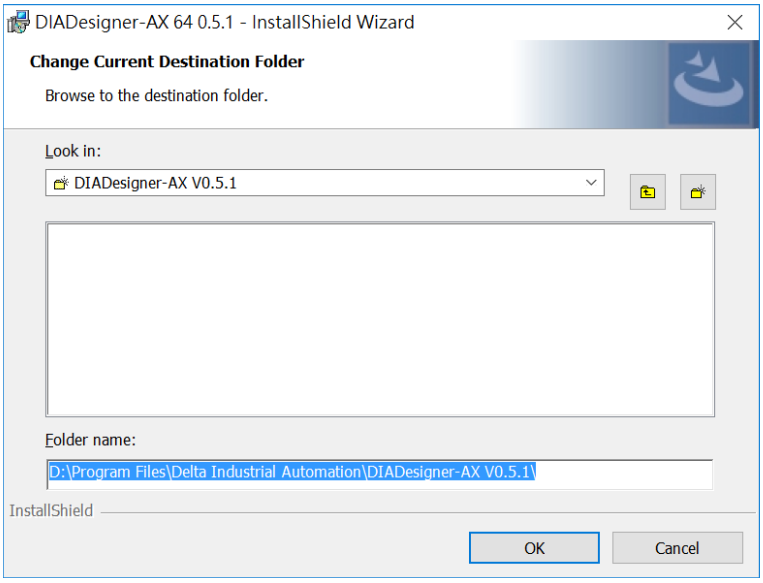
DIADesigner-AX 64 0.5.1 - InstallShield Wizard

Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install DIADesigner-AX 64 0.5.1 to:
C:\Program Files\Delta Industrial Automation\DIADesigner-AX V0.5.1\

Or click **Change…** to change the download path.



DIADesigner-AX 64 0.5.1 - InstallShield Wizard

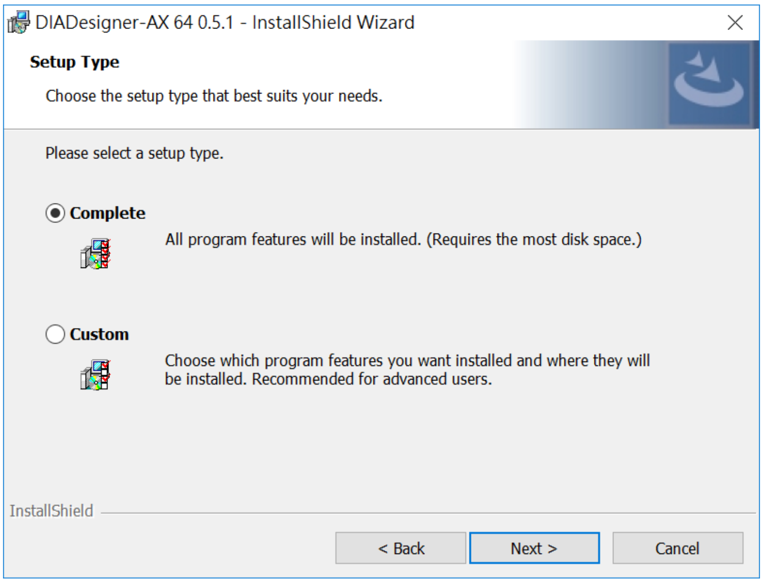
Change Current Destination Folder

Browse to the destination folder.

Look in: DIADesigner-AX V0.5.1

Folder name: D:\Program Files\Delta Industrial Automation\DIADesigner-AX V0.5.1\

8. The window of Setup Type shows up as the image shown below. Select the one you need and then click **Next**.



DIADesigner-AX 64 0.5.1 - InstallShield Wizard

Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

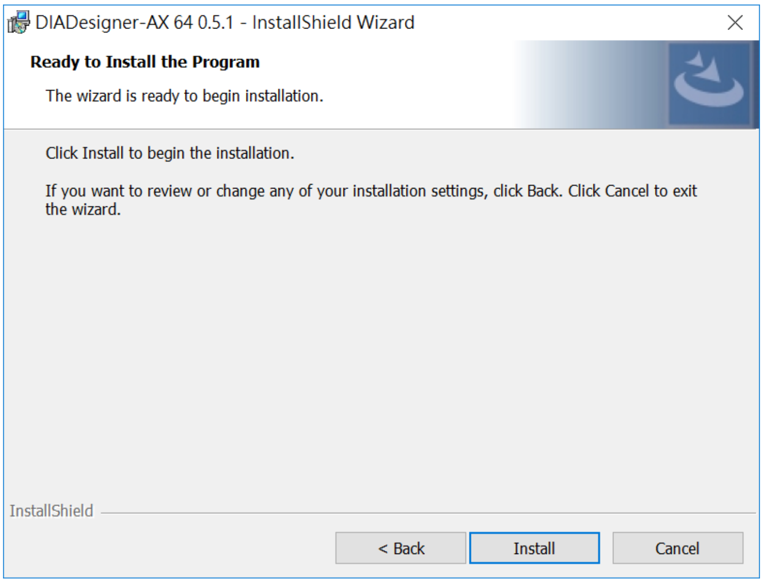
● **Complete**

All program features will be installed. (Requires the most disk space.)

○ **Custom**

Choose which program features you want installed and where they will be installed. Recommended for advanced users.

9. The window of Ready to Install the Program appears as below and then click **Install**.



DIADesigner-AX 64 0.5.1 - InstallShield Wizard

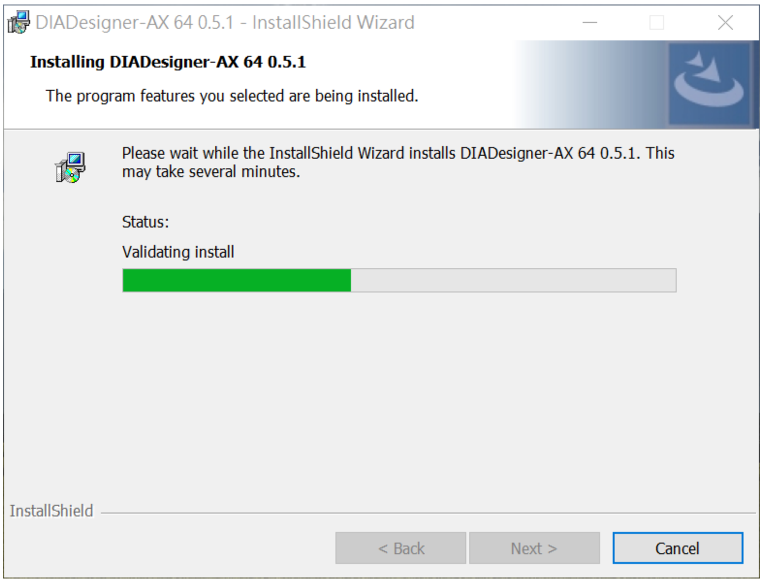
Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

It may take some time to install.



DIADesigner-AX 64 0.5.1 - InstallShield Wizard

Installing DIADesigner-AX 64 0.5.1

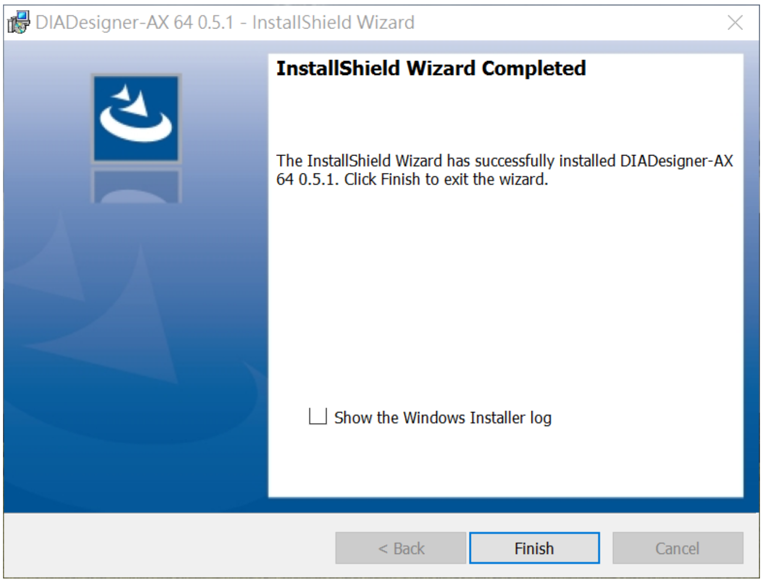
The program features you selected are being installed.

Please wait while the InstallShield Wizard installs DIADesigner-AX 64 0.5.1. This may take several minutes.

Status:

Validating install

10. After installation, the window of InstallShield Wizard Completed appears. Click **Finish** to complete the installation.



InstallShield Wizard Completed

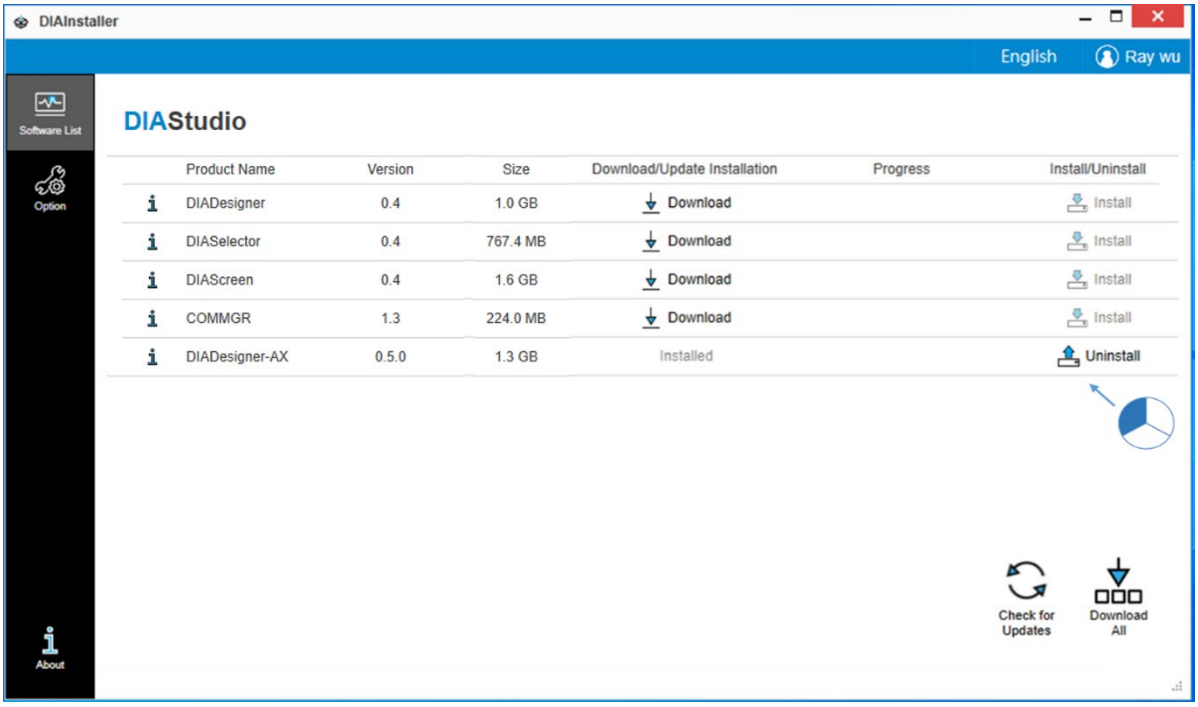
The InstallShield Wizard has successfully installed DIADesigner-AX 64 0.5.1. Click Finish to exit the wizard.

☐ Show the Windows Installer log

3.1.2 Uninstalling DIADesigner-AX

Follow the steps below for uninstalling DIADesigner-AX.

1. Double-click DIAInstaller icon to open and then click **Uninstall**.



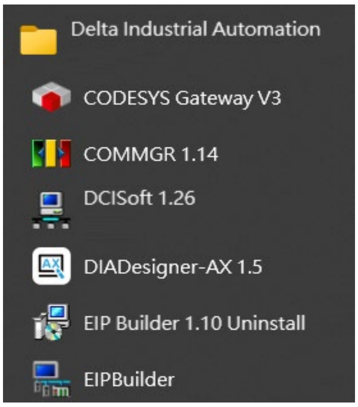
	Product Name	Version	Size	Download/Update Installation	Progress	Install/Uninstall
i	DIADesigner	0.4	1.0 GB	Download		Install
i	DIASelector	0.4	767.4 MB	Download		Install
i	DIAScreen	0.4	1.6 GB	Download		Install
i	COMMGR	1.3	224.0 MB	Download		Install
i	DIADesigner-AX	0.5.0	1.3 GB	Installed		Uninstall

2. The system will remove DIADesigner-AX from your computer in the background.

3.2 Getting Started and Setting up Communication

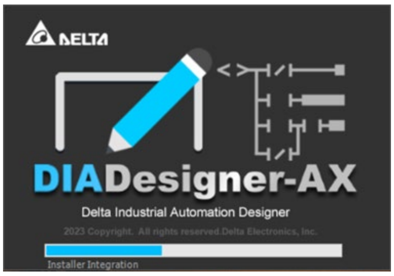
3.2.1 Getting Started

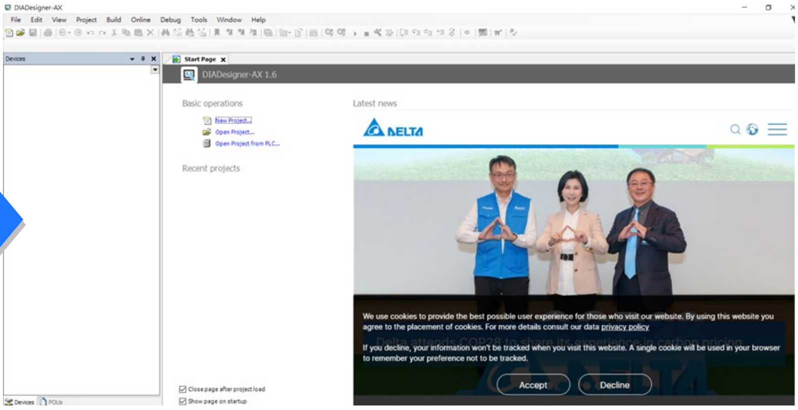
After DIADesigner-AX is successfully installed, click **Start** ⊞, you can find it under the folder of Delta Industrial Automation, and you can also find its short cut on the desktop. Double-click either one to start the software. You can open more than one DIADesigner-AX software to achieve multitasking.



OR

After the loading is done, you can see the start page as below. Refer to Chapter 4 for more details on operation.



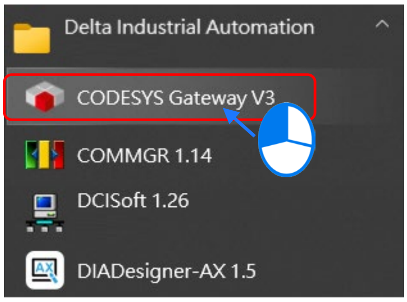


We use cookies to provide the best possible user experience for those who visit our website. By using this website you agree to the placement of cookies. For more details consult our data privacy policy

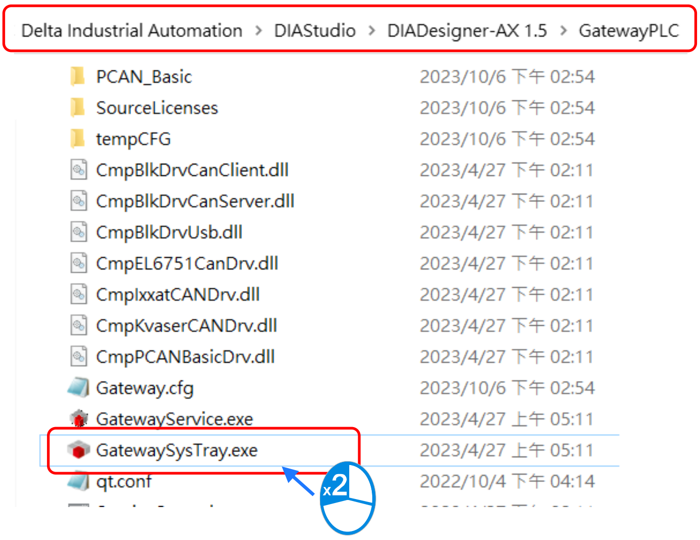
If you decline, your information won't be tracked when you visit this website. A single cookie will be used in your browser to remember your preference not to be tracked.

3.2.2 Setting up Communication

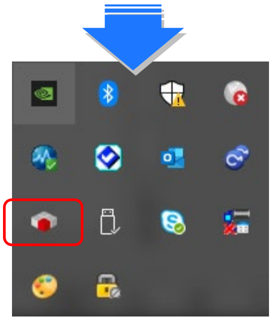
After DIADesigner-AX is successfully installed, the system creates the execution file **CODESYS Gateway V3** under the folder of Delta Industrial Automation and **GatewaySysTray.exe** in the Program Files folder. Double-click either one to start the Gateway. After that, the system starts Gateway automatically whenever you turn your computer on. And its icon will appear on the taskbar. If not, go to the execution file **CODESYS Gateway V3** under the folder of Delta Industrial Automation or **GatewaySysTray.exe** in the Program Files folder to start the Gateway manually.



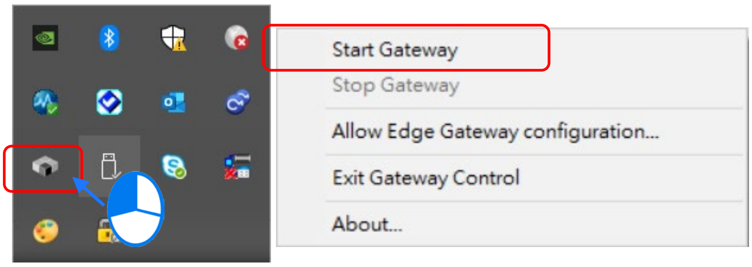
OR

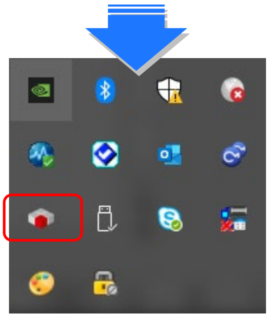

Delta Industrial Automation > DIAStudio > DIADesigner-AX 1.5 > GatewayPLC

PCAN_Basic	2023/10/6 下午 02:54
SourceLicenses	2023/10/6 下午 02:54
tempCFG	2023/10/6 下午 02:54
CmpBlkDrvCanClient.dll	2023/4/27 下午 02:11
CmpBlkDrvCanServer.dll	2023/4/27 下午 02:11
CmpBlkDrvUsb.dll	2023/4/27 下午 02:11
CmpEL6751CanDrv.dll	2023/4/27 下午 02:11
CmpIxxatCANDrv.dll	2023/4/27 下午 02:11
CmpKvaserCANDrv.dll	2023/4/27 下午 02:11
CmpPCANBasicDrv.dll	2023/4/27 下午 02:11
Gateway.cfg	2023/10/6 下午 02:54
GatewayService.exe	2023/4/27 上午 05:11
GatewaySysTray.exe	2023/4/27 上午 05:11
qt.conf	2022/10/4 下午 04:14

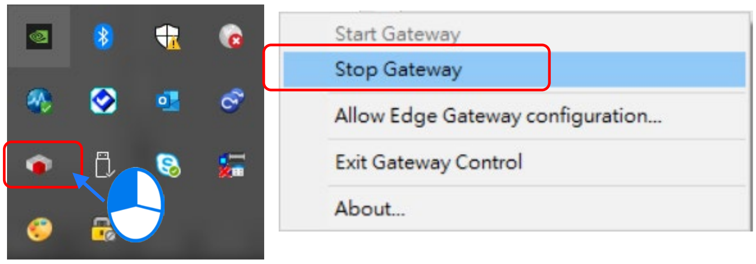


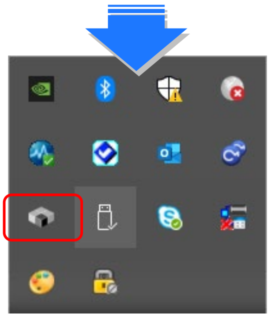
You can click the Gateway icon on the taskbar to see the Gateway status.





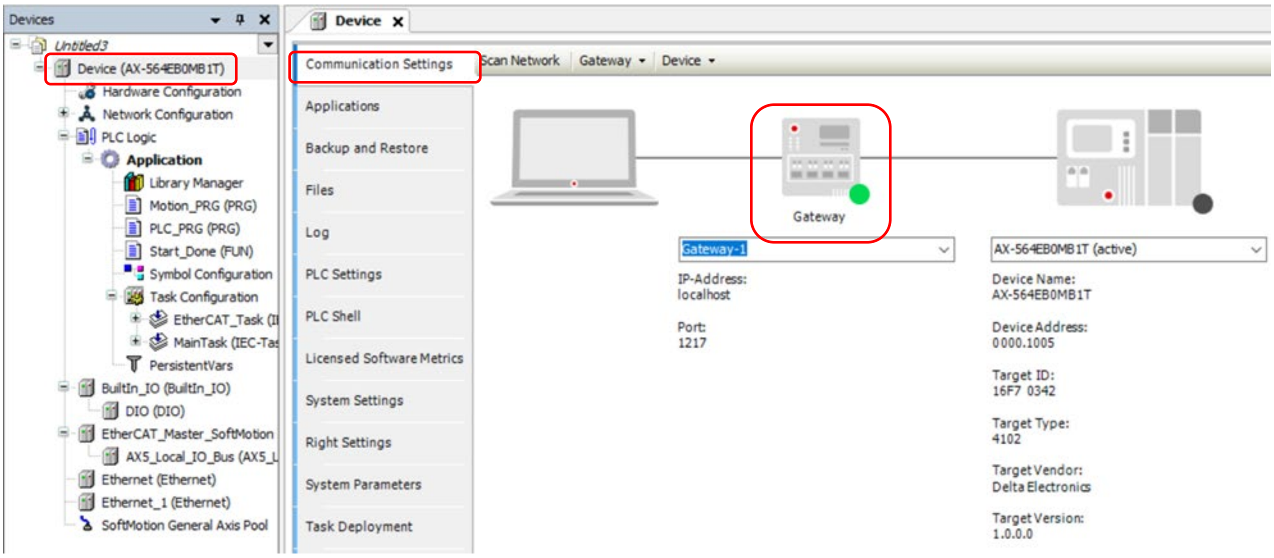
Click **Stop Gateway** if you need to stop gateway working.





If you need to discontinue the execution of GatewaySysTray completely, you can click **Exit Gateway Control** and the icon ◼ will disappear on the taskbar.

Open the software DIADesigner-AX and open/create your project to see the project-setting page. Double-click Device (Product Name) to open the device-setting page. You can find the Gateway status under the Communication Settings tab. If the Gateway is started, its light is green. If the Gateway is stopped, its light is red.



Gateway-1

IP-Address:
localhost

Port:
1217

AX-564EB0MB1T (active)

Device Name:
AX-564EB0MB1T

Device Address:
0000.1005

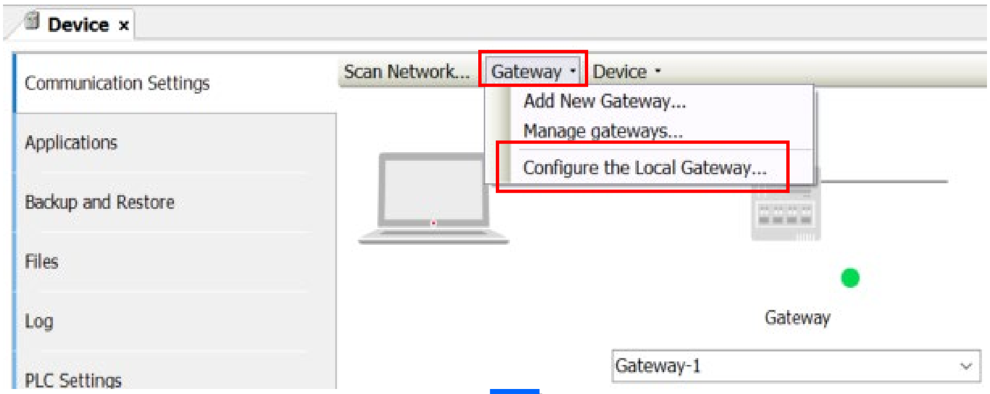
Target ID:
16F7 0342

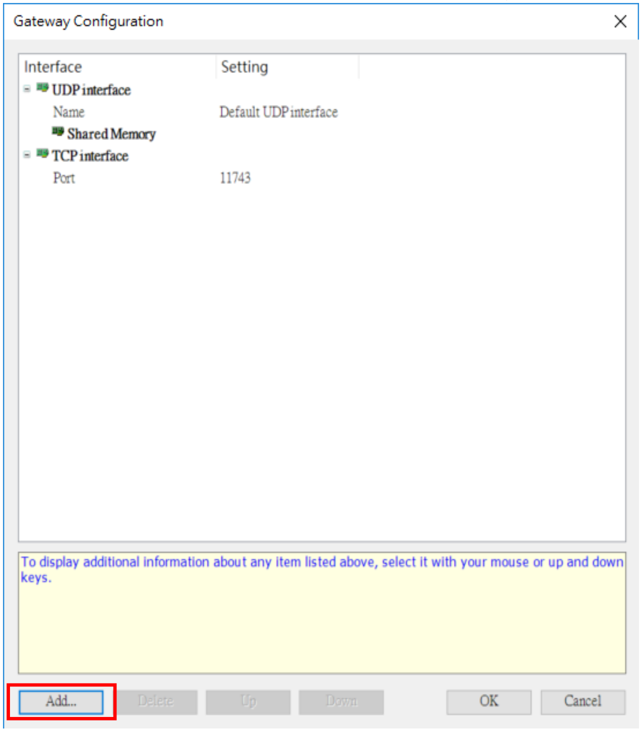
Target Type:
4102

Target Vendor:
Delta Electronics

Target Version:
1.0.0.0

You can configure the Local Gateway. Click **Gateway** and click the option **Configure the Local Gateway** to open the setting page.

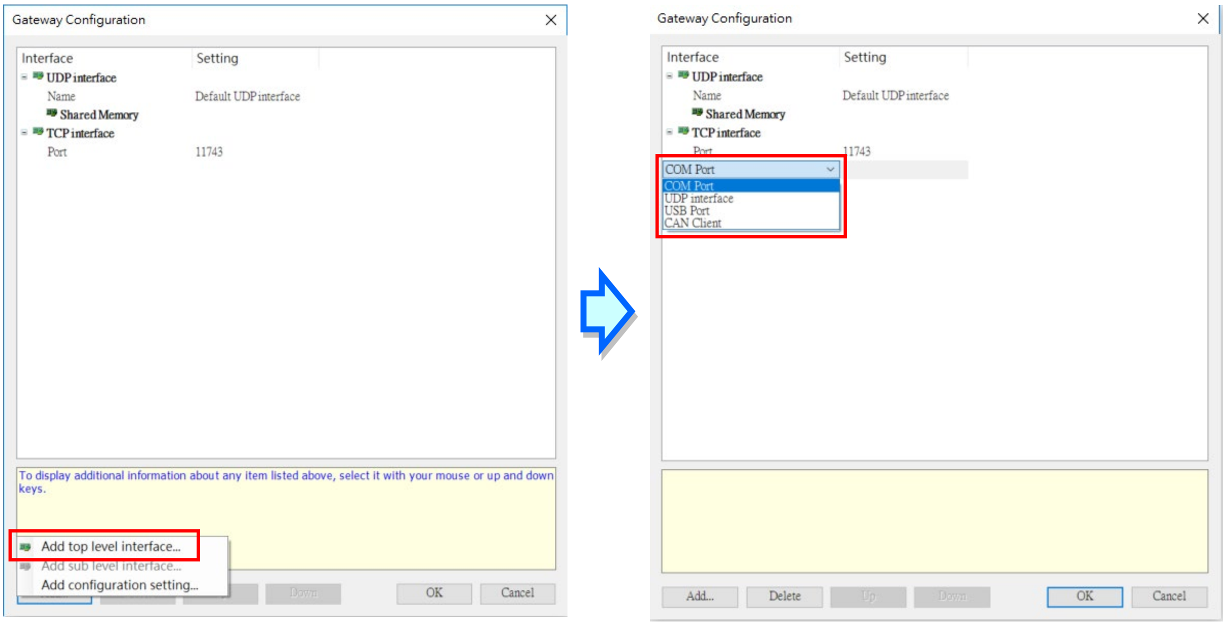




Interface	Setting
UDP interface	
Name	Default UDP interface
Shared Memory	
TCP interface	
Port	11743

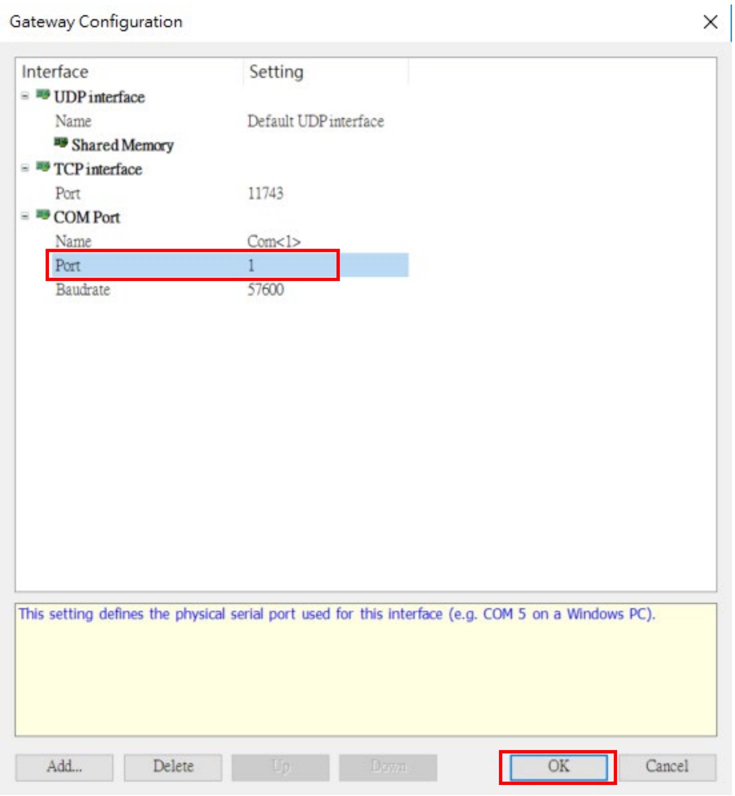
To display additional information about any item listed above, select it with your mouse or up and down keys.

You can find two interfaces under Local Gateway, including UDP interface and TCP interface. You can also create a different port. Click **Add** and select **Add top level interface** and then use the drop-down list to select the port you needed to add. Here we use adding COM Port as an example.



To display additional information about any item listed above, select it with your mouse or up and down keys.

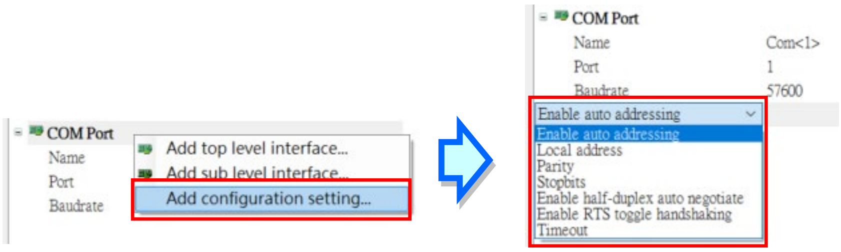
After adding COM Port, you can set up the COM port name, its corresponding port and the baudrate. Once the setting is done, click **OK**. You need to Stop/Start GatewaySysTray again to ensure the following action, such as Scan Network to work properly. Refer to the previous steps to run GatewaySysTray again.



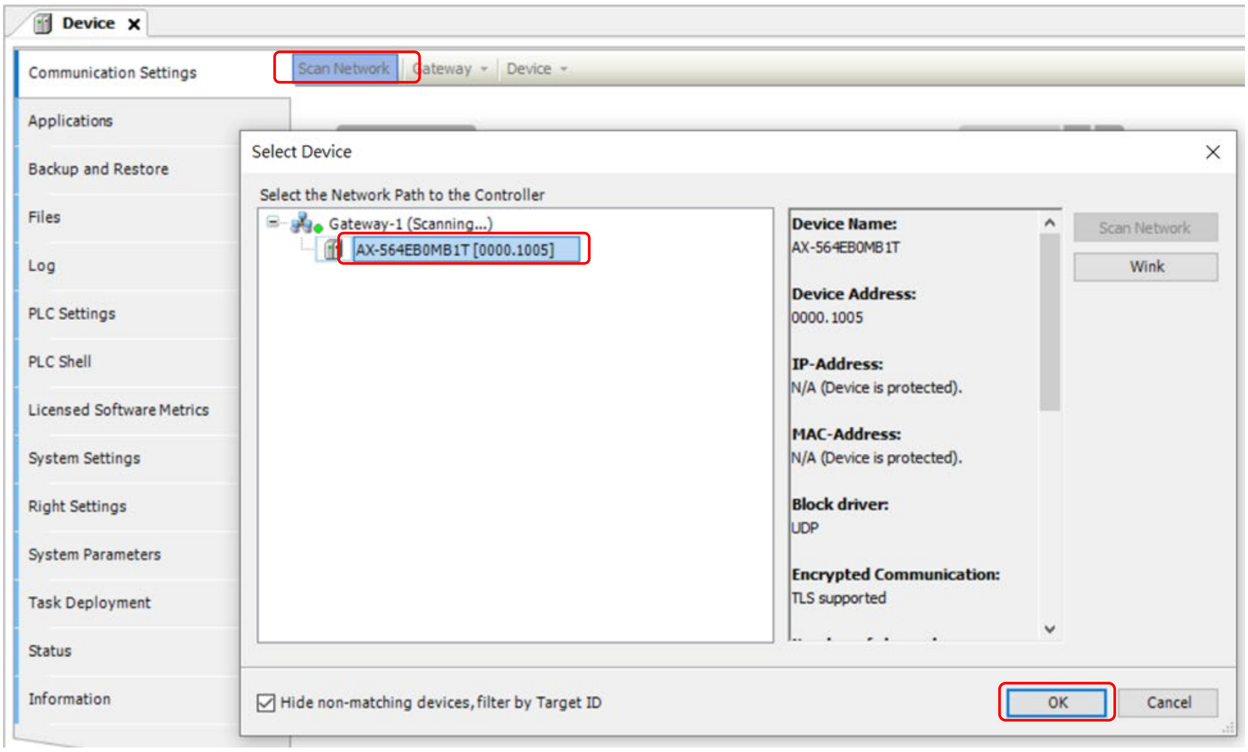
Name Default UDP interface
Port 11743
Name Com<1>
Port 1
Baudrate 57600

This setting defines the physical serial port used for this interface (e.g. COM 5 on a Windows PC).

You can add configuration settings under COM Port. Right-click the COM Port icon COM Port, select **Add configuration setting…..** to add the setting items. After that you can further define the setting values. Once the setting is done, click **OK**.



After the configurations of Local Gateway are set, you can select the **Scan Network** tab to bring out network scanned results on the **Select Device** setting page. Select **AX-308EA0MA1T** and then click **OK.**



Device Name:
AX-564EB0MB1T

Device Address:
0000.1005

IP-Address:
N/A (Device is protected).

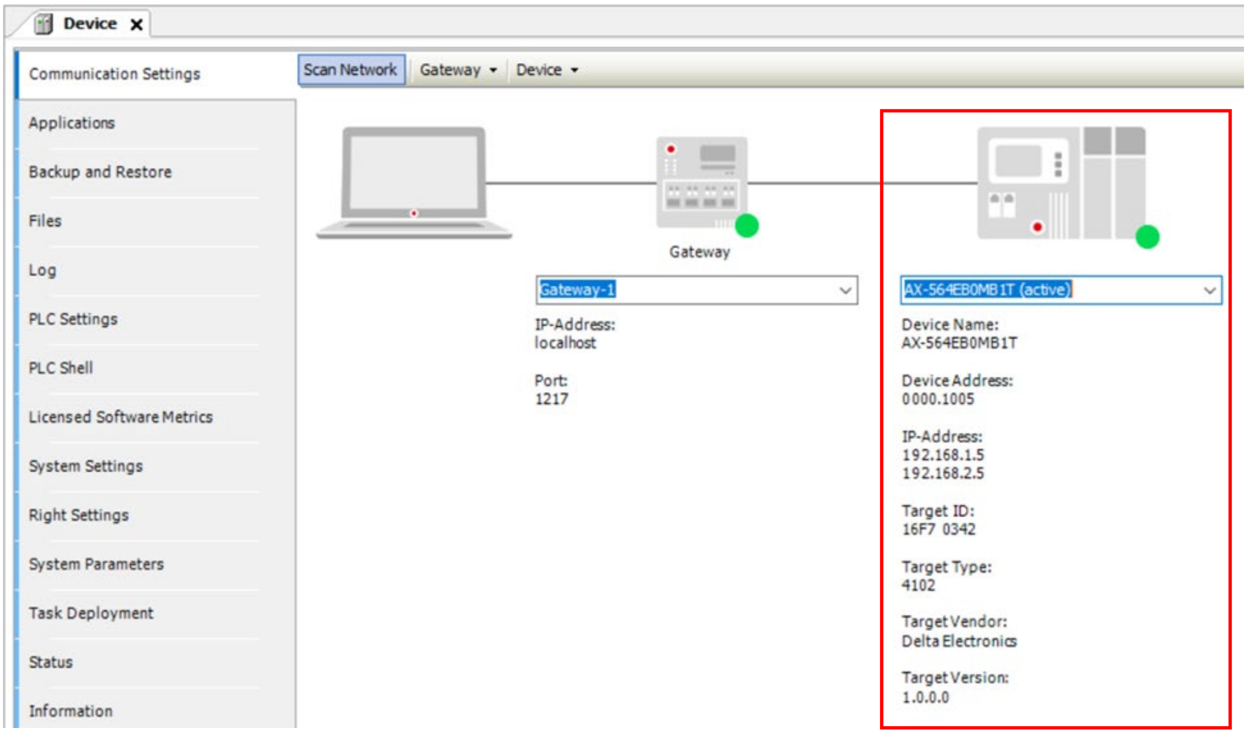
MAC-Address:
N/A (Device is protected).

Block driver:
UDP

Encrypted Communication:
TLS supported

☑ Hide non-matching devices, filter by Target ID

If the connection is established successfully, you can find that the status light is green and the detailed device information under the device image.



Gateway-1

IP-Address:
localhost

Port:
1217

AX-564EB0MB1T (active)

Device Name:
AX-564EB0MB1T

Device Address:
0000.1005

IP-Address:
192.168.1.5
192.168.2.5

Target ID:
16F7 0342

Target Type:
4102

Target Vendor:
Delta Electronics

Target Version:
1.0.0.0

Chapter 4 Basic Operation

Table of Contents

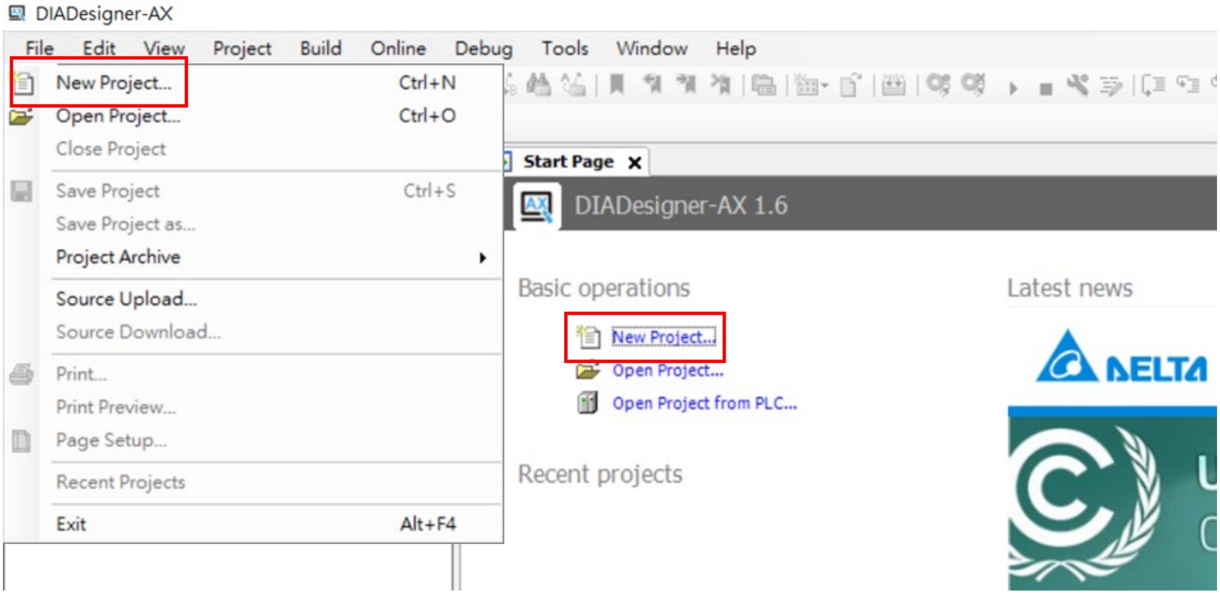
4.1 Introduction on DIADesigner-AX

DIADesigner-AX is an open platform for PLC development system and industrial automation. The adaptable DIADesigner-AX provides an easy way to create professional engineering of IEC 61131-3 automation projects. Based on the IEC 61131-3 data structure and the high-level language programming, DIADesigner-AX is strong in functionality, easy to develop, reliable, extendable and open for development. Integrated with components such as visualization and Safety solution, DIADesigner-AX offers a variety of user-friendly engineering functions for your professional applications in controller development system sectors including PLC and motion control.

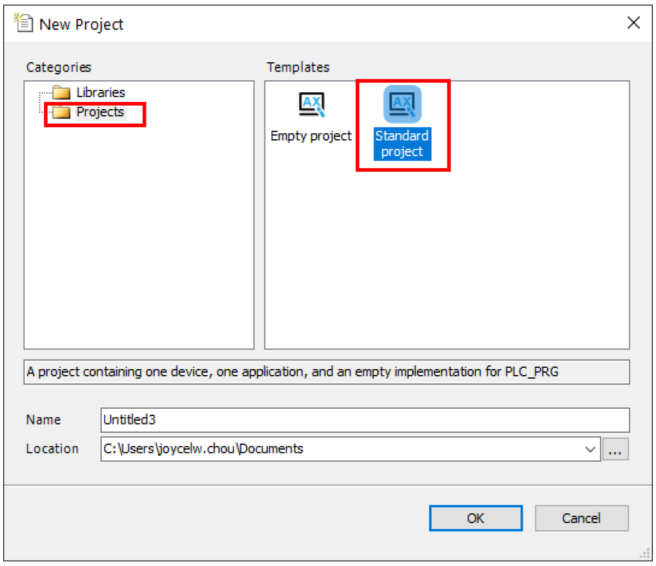
In DIADesigner-AX, you can customize the user interface by arranging the window layout and the appearance of menus, toolbars, and commands according to your requirements.

4.1.1 Creating a New Project

Double-click the DIADesigner-AX icon to open DIADesigner-AX. Click **New Project** on the Start Page or select *File* > *New Project (Ctrl+N)* to create a new project.



Next you will see a window with two sections, Categories and Templates. Click **Projects** in the Categories section and click **Standard project** in the Templates section. After that create a Name and specify a location for the project and then click **OK**.



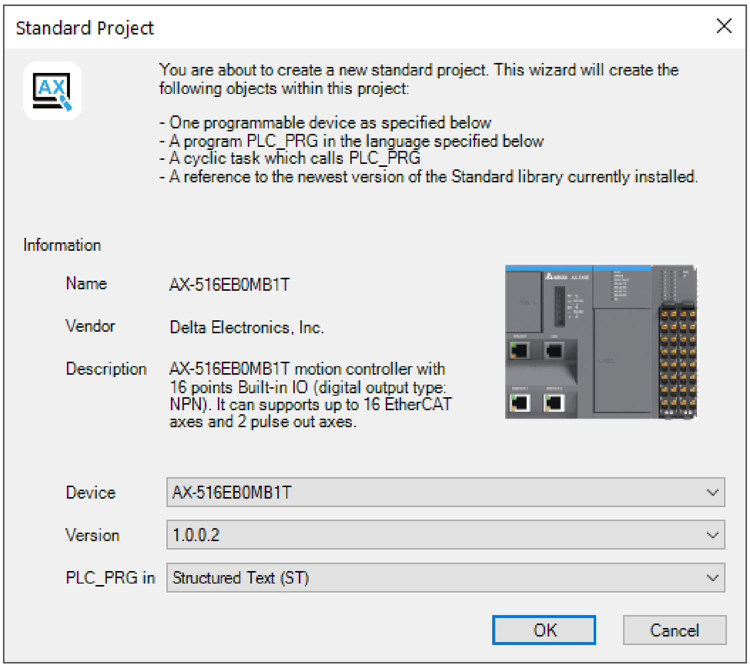
A project containing one device, one application, and an empty implementation for PLC_PRG

Name: Untitled3

Location: C:\Users\joycelw.chou\Documents

And a Standard Project dialog appears. You can select the device and the programming language from the drop-down list. Click **OK**, the system generates a cyclic task with a default PLC_PRG.

Note: It is required to use DIADesigner-AX V1.6 or later for AX-5 Series operation.



You are about to create a new standard project. This wizard will create the following objects within this project:

- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Information

Name: AX-516EB0MB1T

Vendor: Delta Electronics, Inc.

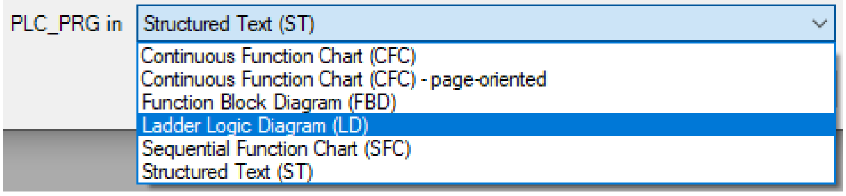
Description: AX-516EB0MB1T motion controller with 16 points Built-in IO (digital output type: NPN). It can supports up to 16 EtherCAT axes and 2 pulse out axes.

Device: AX-516EB0MB1T

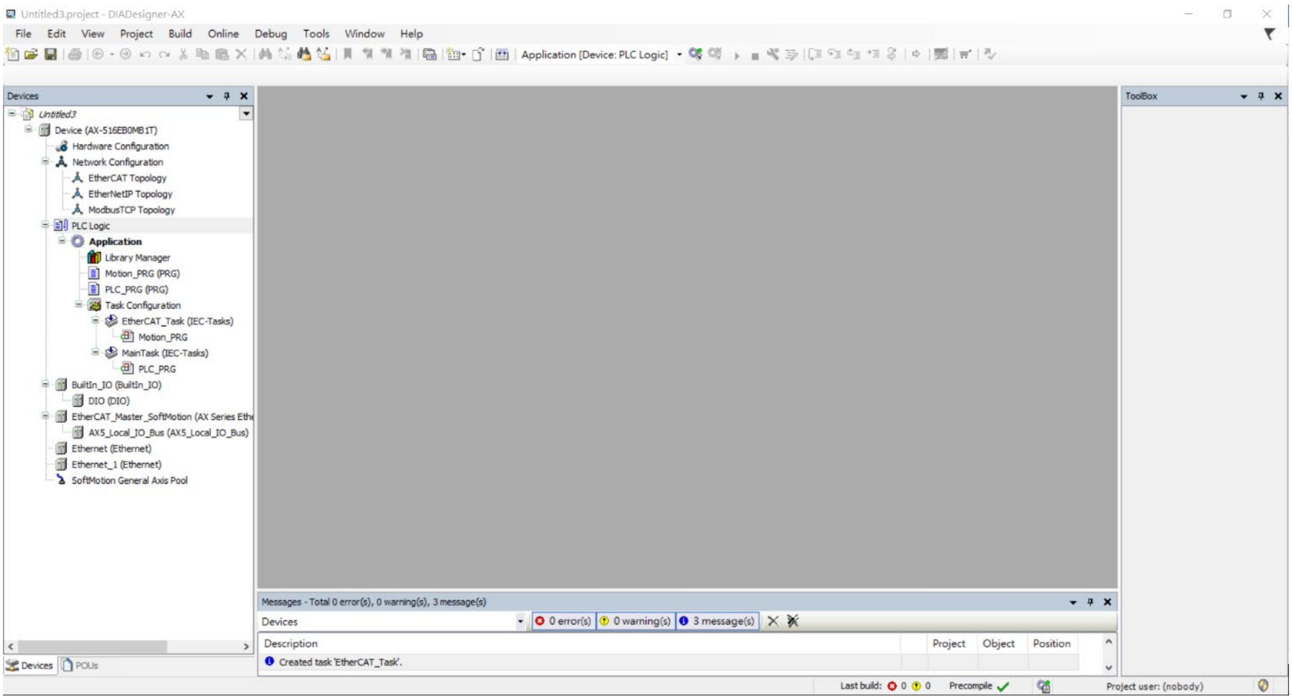
Version: 1.0.0.2

PLC_PRG in: Structured Text (ST)

You can select a POU programming language from the drop-down list.



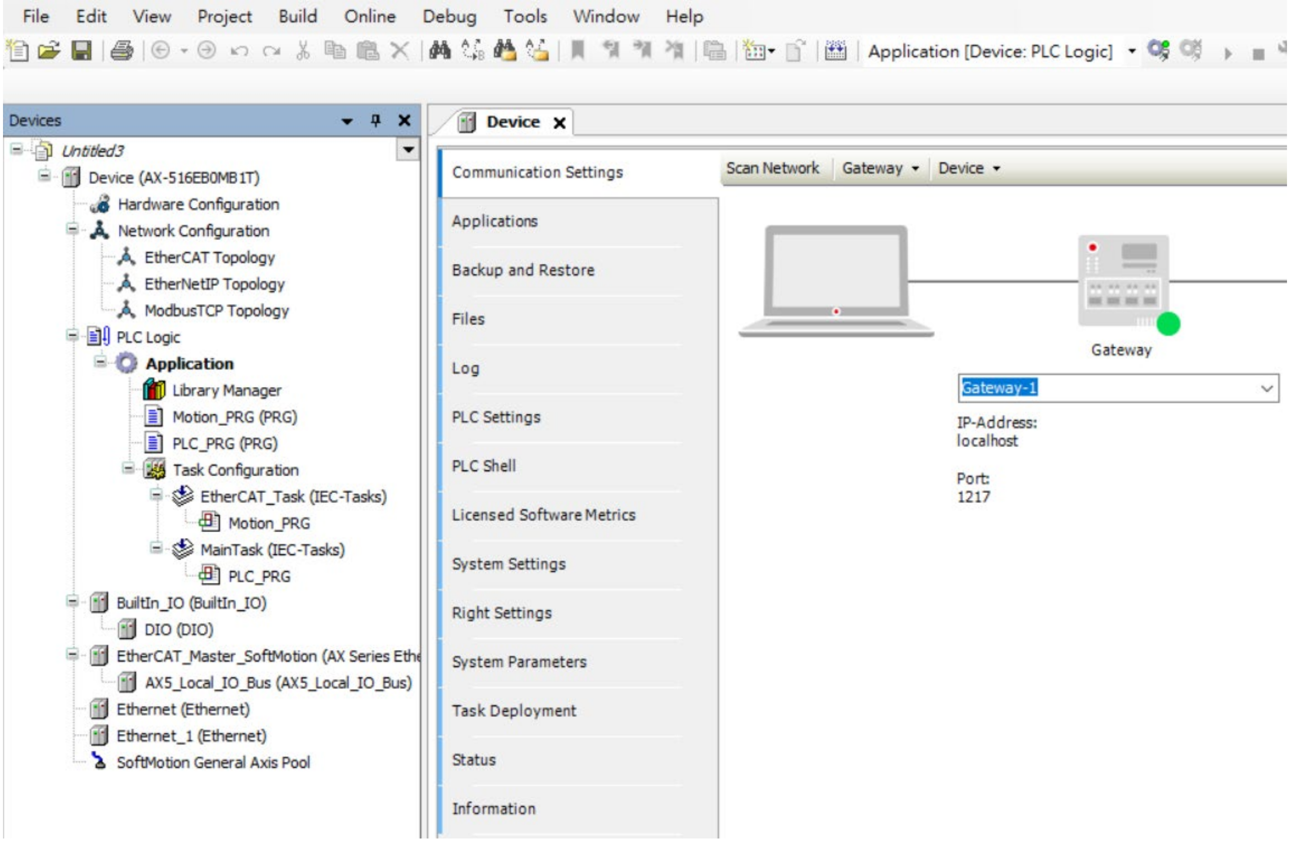
After a new project is successfully created, you can see a project management area in the left side of the window. All the options are listed in nodes. Click *View -> Devices (Alt+0)* on the toolbar if nothing appears in the project management area.



Created task 'EtherCAT_Task'.

4.2 Setting Items on the Device Page

This section introduces all the setting items on the Device Page.



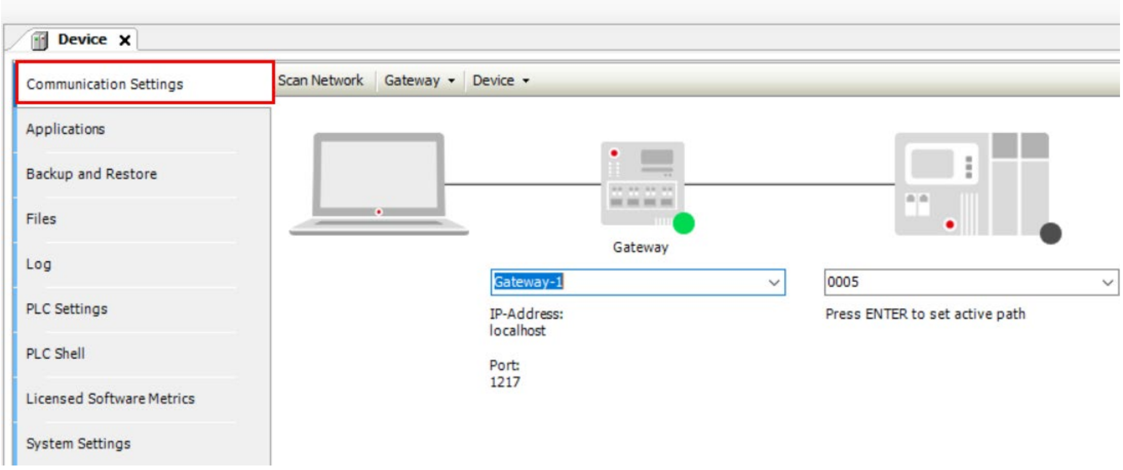
IP-Address:
localhost

Port:
1217

4.2.1 CPU Parameter Settings

4.2.1.1 Communication Settings

On the Communication Settings page, you can define the communication method for DIADesigner-AX and controller. Use the drop-down list of the Gateway tab to add new gateways or manage existing gateways or configure local gateways. You can simply specify an IP address or DNS address for the gateway while adding new gateways. This is useful if you want to connect to a remote gateway running on another PC or device. If you use DNS the address must begin with “dns.” For the setting of PLC, you can enter its IP address (e.g., 192.168.1.5) or its device name (e.g., AX-516EB0MB1T) in the field under the controller image. After that DIADesigner-AX scans to search for the PLC in the network of the gateway.

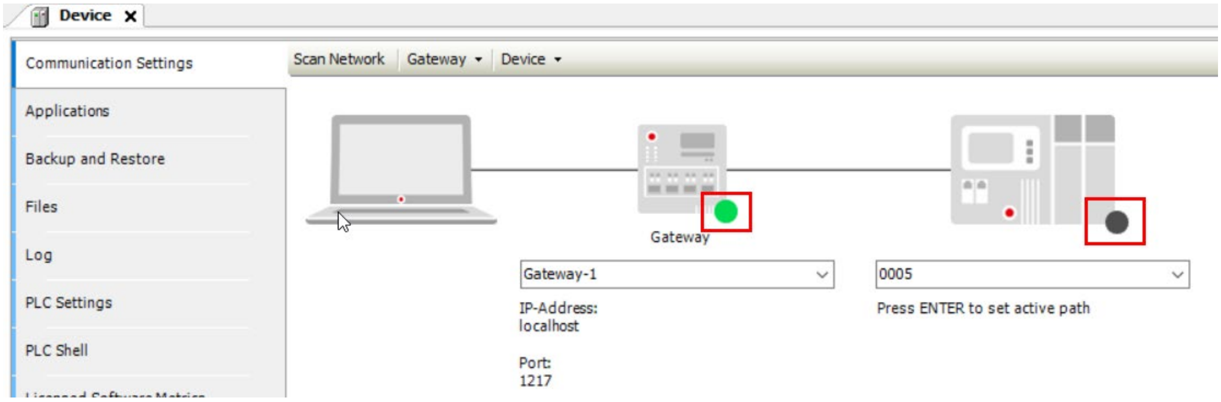


IP-Address:
localhost

Port:
1217

Press ENTER to set active path

- Status of the Connection



Gateway-1

IP-Address:
localhost

Port:
1217

0005

Press ENTER to set active path

The dots under the images of gateway and controller indicate the connection status.

Red: Not be able to establish a connection

Green: A connection is established.

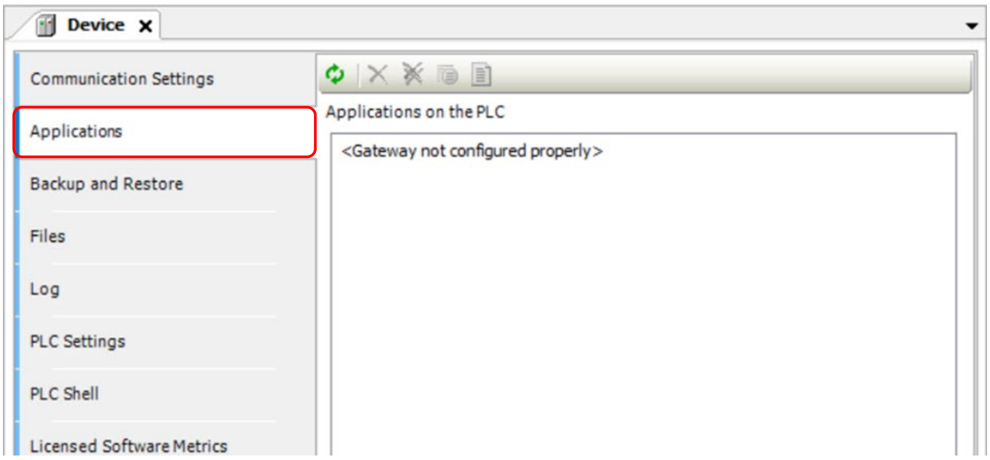
Black: Unknown connection status

Tab	Description
Scan Network	Click **Scan Network** to open the Select Device page. This page lists all configured gateways with the associated devices. You can select one target device from this list.
Gateway	This menu includes the following setting items: ● **Add New Gateway:** You can add and define a new gateway channel here. ● **Manage Gateways:** This page is with an overview of all gateways. You can add or delete entries here or change their order. ● **Configure the Local Gateway:** Select this setting item to open the Gateway Configuration page. You can configure the block drivers for the local gateway.
Device	This menu includes the following setting items: **1. Options:** • **Add Current Device to Favorites:** Adds the currently set device to the list of favorite devices. • **Manage Favorite Devices:** Click this option to open a list of all preferred devices. You can add or delete entries or change their order. The top device is the default. • **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. • **Confirm Online Mode:** ☑ : DIADesigner-AX requires you to confirm the followings when calling the following online commands (for safety purposes): Force values, Write values, Multiple loading, Remove force list, Single cycle, Start, and Stop. • **Store Communication Settings in Project:** ☑ : DIADesigner-AX saves the communication settings in the project for reuse on the same computer. Note: If you use the project on another computer, you need 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 to prevent blocking the device object.

<table>
<tr><th>Tab</th><th colspan="2">Description</th></tr>
<tr><td rowspan="7"></td><td colspan="2">2. Rename Active Device: Click this setting item to open the Change Device Name page.
Wink Current Device: Devices that support this function illuminate a flashing signal.
3. 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.
4. Encrypted Communication:
☑: The communication to this controller is encrypted. A certificate of the controller is required to log in to the controller. If the certificate is not available, then an error message shows up prompting whether the certificate should be displayed and installed.
If the Enforce Encrypted Communication option is selected as Security level in the <i>Security Screen</i> view, then the Encrypted Communication is disabled here.
5. Change Communication Policy: Click this setting item to open the Change Communication Policy page for changing the device setting for the encryption of communication.
If a new communication policy is selected in this dialog, then the configuration on the controller is changed.</td></tr>
<tr><td colspan="2">Communication</td></tr>
<tr><td>Current policy</td><td>The currently selected policy for the encryption of communication</td></tr>
<tr><td>New policy</td><td>Drop-down list for the new policy for encryption
• <i>No encryption</i>: The controller does not support encrypted communication.
• <i>Optional encryption</i>: The controller supports encrypted and unencrypted communication.
• <i>Enforced encryption</i>: The controller supports encrypted communication only.</td></tr>
<tr><td colspan="2">Device User Management</td></tr>
<tr><td>Current policy</td><td>The currently selected policy for user management</td></tr>
<tr><td>New policy</td><td>Drop-down list for the new policy for user management
• <i>Optional user management</i>: It is the responsibility of the user to enable user management on the device or leave the device unprotected.
• <i>Enforced user management</i>: The user management on the device is enabled and cannot be disabled by the user.</td></tr>
</table>

4.2.1.2 Applications

Here you can check and manage the applications on the PLC.

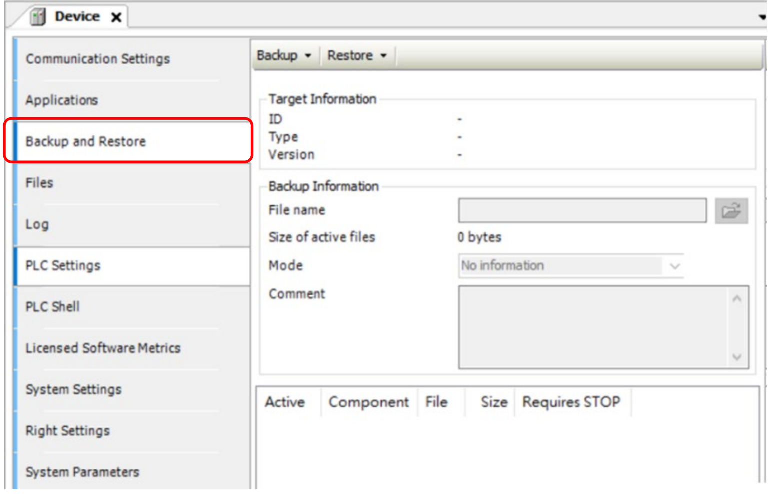


<Gateway not configured properly>

Button	Description
Remove / Remove All	**Remove:** Deletes the application selected in the list. **Remove All:** Deletes all listed applications on the PLC.
Details…	Click **Details** button to see information defined for the application on the Information tab of the dialog box Properties.
Content…	Requirement: Go to *Application > Proprieties > Application Generation Options* to activate the **Download the application info** option. This causes information about the contents of the application to be additionally loaded to the PLC. Click **Content** button to see additional information about the differences between the latest generated code and the application code that exists on the controller. The different modules are displayed in a comparison view.
Refresh List	Click **Refresh List** button to have the controller scanned for applications and the list is refreshed accordingly.

4.2.1.3 Backup and Restore

You can backup and restore the application-specific file on the PLC by saving and reading a zip archive.



ID -
Type -
Version -
File name
Size of active files 0 bytes
Mode No information
Comment

Tab	Description
Backup	Click **Backup** tab to see the followings ● **Read Backup Information from Device:** Use this function to search for application-specific files from the $PlcLogic$ directory of the PLC and lists them on the Backup tab page. ● **Create Backup File and Save to Disk**: Use this function to compress the files in into a backup zip file. The file extension is tbf (=”Target Backup File”). ● **Save Backup File to Device:** Use this function to save the backup file to the TBF directory of the PLC.
Restore	● **Load Backup File from Disk**: After clicking this button, the system generates a list of all backup files found on the disk. Select one of these files to view its contents. ● **Load Backup File from Device:** After clicking this button, the system generates a list of all backup files found on the PLC. Select one of these files to view its contents. ● **Restore on Device:** This function is available if at least one component of the backup file that is currently loaded in the tabbed page is set to active. It prompts for restoring the application status on the device.

- **Target Information**

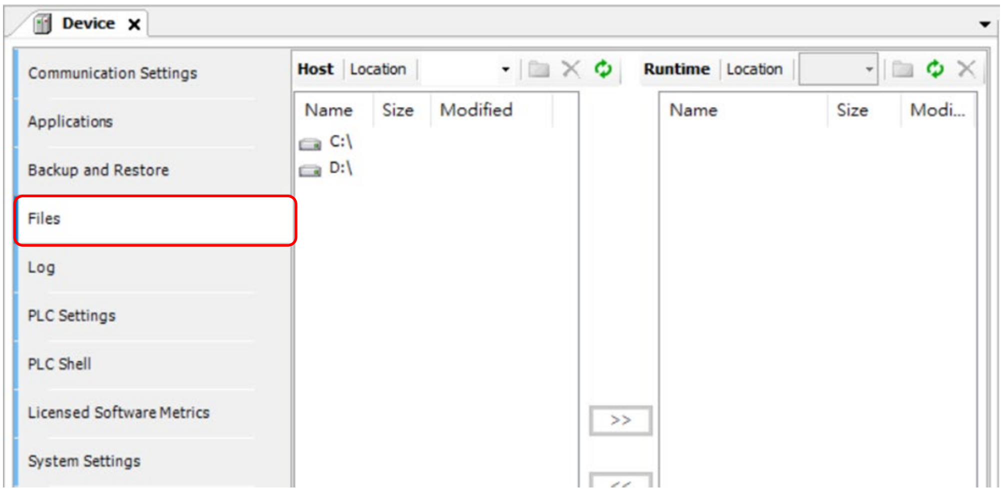
ID	ID of the PLC
Type	Device type
Version	Device version

- **Backup Information**

File name	Storage path of the backup file.
Size of active files	Total size of the files set as active in the table
Mode	Defines the scope of the backup: Application. The application-related files are added to the archive.
Comment	Optional entry for comments to be saved in the meta.info file of the backup and reading when the files are restored.

4.2.1.4 Files

You can transfer files between the computer and the PLC on this page through DIADesigner-AX.

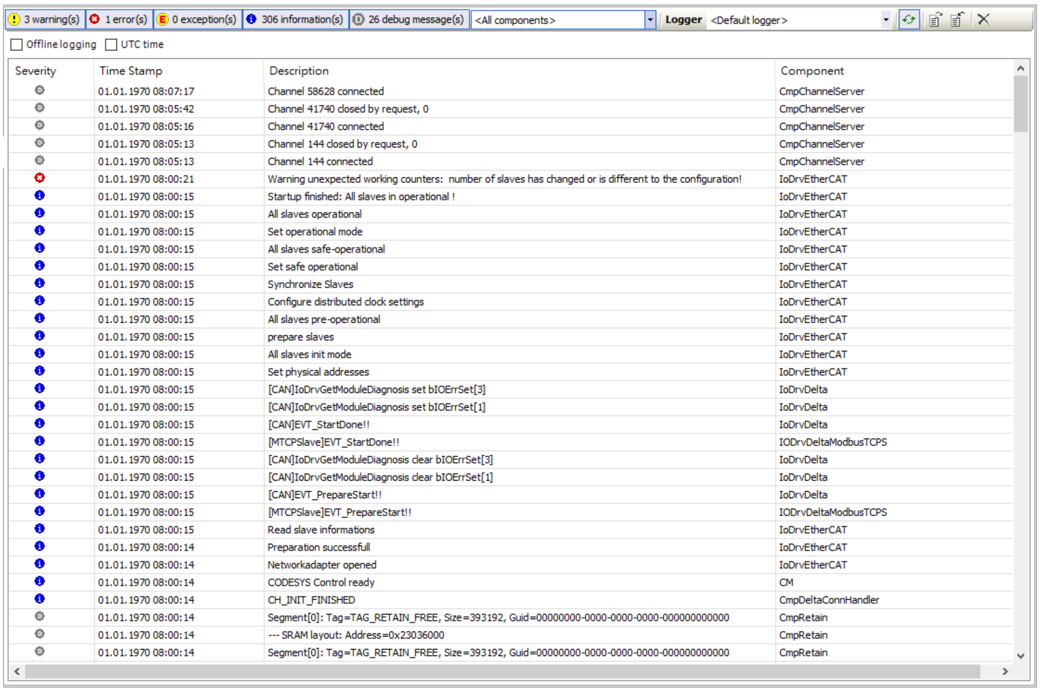


Item	Description
Location	Path in the file system of the computer. Subdirectories and files are shown in the lower part of the view with name, size, and change date.
📁	Click this button to create a new file folder
✕	Deletes the selected files or folders
⟳	Updates the list of files and folders for the set path (location)
>>	Write File to the PLC
<<	Write File from the PLC

4.2.1.5 Log

You can view the PLC log here. It lists the events that were recorded on the target system, including

- Events during the startup and shutdown of the system (components loaded, with version)
- Application download and loading of the boot application
- Custom entries
- Log entries from I/O drivers
- Log entries from data sources



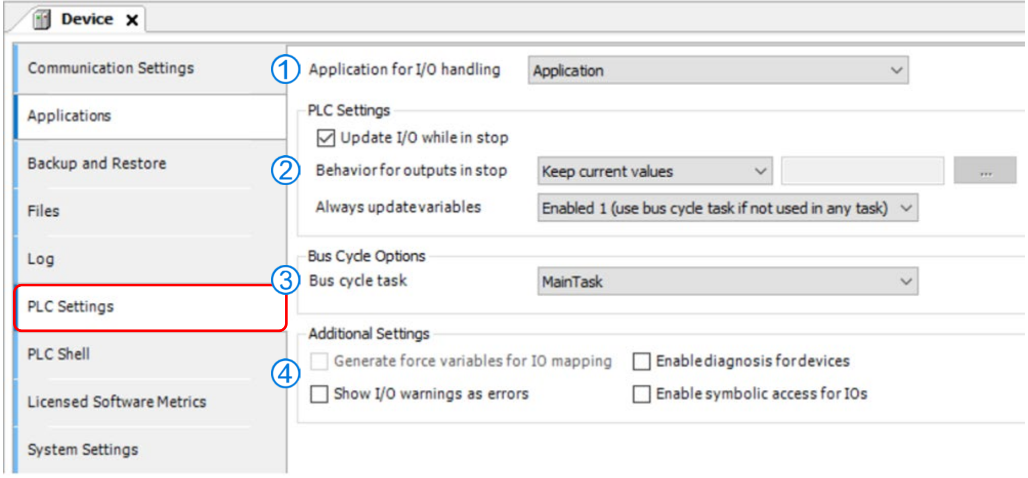
☐ Offline logging ☐ UTC time

Severity	Time Stamp	Description	Component
Ⓓ	01.01.1970 08:07:17	Channel 58628 connected	CmpChannelServer
Ⓓ	01.01.1970 08:05:42	Channel 41740 closed by request, 0	CmpChannelServer
Ⓓ	01.01.1970 08:05:16	Channel 41740 connected	CmpChannelServer
Ⓓ	01.01.1970 08:05:13	Channel 144 closed by request, 0	CmpChannelServer
Ⓓ	01.01.1970 08:05:13	Channel 144 connected	CmpChannelServer
⊗	01.01.1970 08:00:21	Warning unexpected working counters: number of slaves has changed or is different to the configuration!	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	Startup finished: All slaves in operational !	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	All slaves operational	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	Set operational mode	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	All slaves safe-operational	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	Set safe operational	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	Synchronize Slaves	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	Configure distributed clock settings	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	All slaves pre-operational	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	prepare slaves	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	All slaves init mode	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	Set physical addresses	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:15	[CAN]IoDrvGetModuleDiagnosis set bIOErrSet[3]	IoDrvDelta
ⓘ	01.01.1970 08:00:15	[CAN]IoDrvGetModuleDiagnosis set bIOErrSet[1]	IoDrvDelta
ⓘ	01.01.1970 08:00:15	[CAN]EVT_StartDone!!	IoDrvDelta
ⓘ	01.01.1970 08:00:15	[MTCPSlave]EVT_StartDone!!	IODrvDeltaModbusTCPS
ⓘ	01.01.1970 08:00:15	[CAN]IoDrvGetModuleDiagnosis clear bIOErrSet[3]	IoDrvDelta
ⓘ	01.01.1970 08:00:15	[CAN]IoDrvGetModuleDiagnosis clear bIOErrSet[1]	IoDrvDelta
ⓘ	01.01.1970 08:00:15	[CAN]EVT_PrepareStart!!	IoDrvDelta
ⓘ	01.01.1970 08:00:15	[MTCPSlave]EVT_PrepareStart!!	IODrvDeltaModbusTCPS
ⓘ	01.01.1970 08:00:15	Read slave informations	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:14	Preparation successfull	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:14	Networkadapter opened	IoDrvEtherCAT
ⓘ	01.01.1970 08:00:14	CODESYS Control ready	CM
ⓘ	01.01.1970 08:00:14	CH_INIT_FINISHED	CmpDeltaConnHandler
Ⓓ	01.01.1970 08:00:14	Segment[0]: Tag=TAG_RETAIN_FREE, Size=393192, Guid=00000000-0000-0000-0000-000000000000	CmpRetain
Ⓓ	01.01.1970 08:00:14	--- SRAM layout: Address=0x23036000	CmpRetain
Ⓓ	01.01.1970 08:00:14	Segment[0]: Tag=TAG_RETAIN_FREE, Size=393192, Guid=00000000-0000-0000-0000-000000000000	CmpRetain

<table>
<tr><th>Item</th><th colspan="2">Description</th></tr>
<tr><td>Offline logging</td><td colspan="2">☐: Default settings
☑: The PLC also records actions that are not related to the connection with the controller. However, this is currently available only for the safety version of CODESYS.</td></tr>
<tr><td>UTC time</td><td colspan="2">☐: Standard setting; the time stamp is converted to the local time on the computer as indicated by the time zone of the operating system.
☑: The time stamp of the runtime system is displayed.</td></tr>
<tr><td>Severity</td><td colspan="2">Four categories for the severity of the event:
• ⓘ : Message
• ! : Warning
• ⊗ : Error
• Ⓓ : Debugging
You can show or hide each category by clicking corresponding buttons in the bar. Each button shows the number of log entries of the category concerned.</td></tr>
<tr><td>Time stamp</td><td colspan="2">Date and time (example: 08-01-2020 09:48)</td></tr>
<tr><td>Description</td><td colspan="2">Description of the event</td></tr>
<tr><td>Component</td><td colspan="2">Name of the runtime system component concerned, e.g., CmpApp</td></tr>
<tr><td>Drop-down list with component names</td><td colspan="2">The log list displays only events that concern the selected component</td></tr>
<tr><td rowspan="5">Logger</td><td colspan="2">Drop-down list with all available logs. The standard setting is the <Default Logger> specified by the target system; now it is identical to ‘StdLogger for DIADesigner-AX runtime system.</td></tr>
<tr><td>⟳</td><td>Refreshes the log list</td></tr>
<tr><td></td><td>Exports the list contents to an xml file.</td></tr>
<tr><td></td><td>Imports a log list from an xml file.</td></tr>
<tr><td>✕</td><td>Deletes the displayed log list. All entries are deleted.</td></tr>
</table>

4.2.1.6 PLC Settings

You can make the basic settings for the configuration of the PLC here, for example the handling of inputs and outputs and the bus cycle task.



① Application for I/O handling: Application

PLC Settings

☑ Update I/O while in stop

② Behavior for outputs in stop: Keep current values

Always update variables: Enabled 1 (use bus cycle task if not used in any task)

Bus Cycle Options

③ Bus cycle task: MainTask

Additional Settings

④ ☐ Generate force variables for IO mapping ☐ Enable diagnosis for devices

☐ Show I/O warnings as errors ☐ Enable symbolic access for IOs

① Application for I/O handling

Item	Description
Application for I/O handling	Application that is for the I/O handling.

② PLC Settings

Item	Description
Update IO while in stop	☐: DIADesigner-AX does not refresh the values of the input and output channels when the PLC is in the stop state. ☑: DIADesigner-AX refreshes the values of the input and output channels even if the PLC is in the stop state. If the watchdog detects a malfunction, the outputs are set to the predefined default values.
Behavior of the outputs in stop	Handling of the output channels when the controller enters the stop state: ● **Keep current values:** The current values are retained. ● **Set all outputs to default:** The default values resulting from the I/O mapping are assigned. ● **Execute program:** You can control the handling of the output values via a program contained in the project, which DIADesigner-AX executes at “STOP.” Enter the name of the program in the field on the right.
Always update variables	Global setting that defines whether DIADesigner-AX updates the I/O variables in the bus cycle task. This setting is effective for I/O variables of the slaves and modules only if ‘disabled’ is defined in their update settings. ● **Disabled (update only if used in a task):** DIADesigner-AX updates the I/O variables only if they are used in a task. ● **Enabled 1 (use bus cycle task if not used in another task**): DIADesigner-AX updates the I/O variables in the bus cycle task if they are not used in any other task. ● **Enabled 2 (always in bus cycle task**): DIADesigner-AX updates all variables in each cycle of the bus cycle task, regardless of whether they are used and whether they are mapped to an input or output channel.

③ **Bus Cycle Options**

Item	Description
Bus cycle task *1	Task that controls the bus cycle. By default, the task defined by the device description is entered.

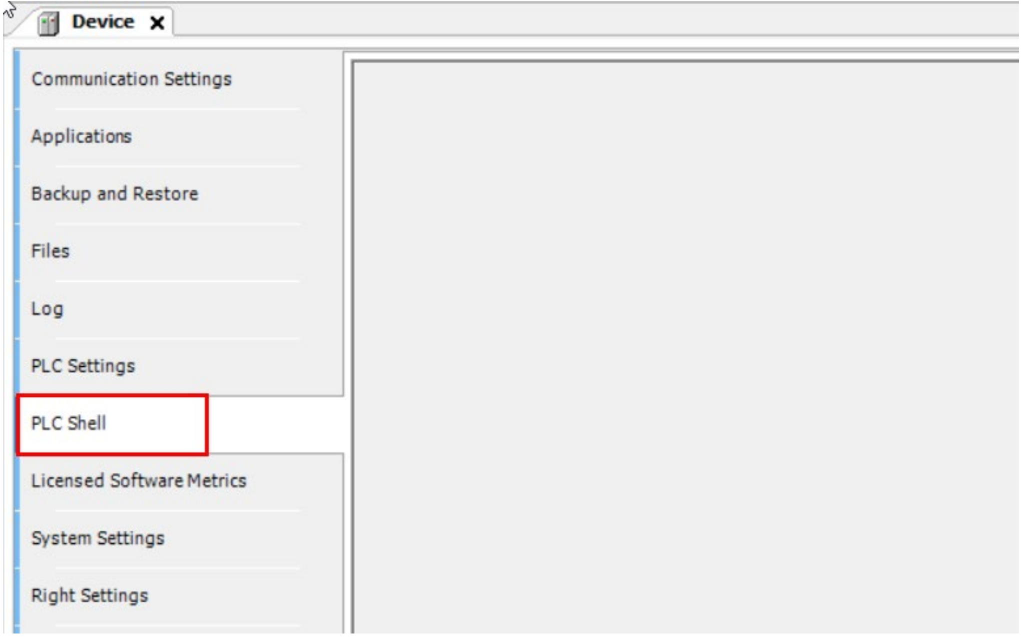
Note 1: Before you select the <unspecified> setting for the bus cycle task, you should be aware that “<unspecified>” means that the default setting given in the device description goes into effects. You should therefore check this description. Use of the task with the shortest cycle time may be defined as the default there, but use of the task with the longest cycle time could equally well be defined!

④ **Additional Settings**

Item	Description
Generate Force variables for I/O mapping	The device does not support this function.
Enable Diagnostics for devices	☑: DIADesigner-AX automatically integrates the library CAA Device Diagnosis in the project and creates an implicit function block for each device. If there is already a function block for the device, then either an extended FB is used (for example with EtherCAT) or a further FB instance is added. This then contains a general implementation of the device diagnostics.
Show I/O warnings as errors	Warnings concerning the I/O configuration are displayed as errors.

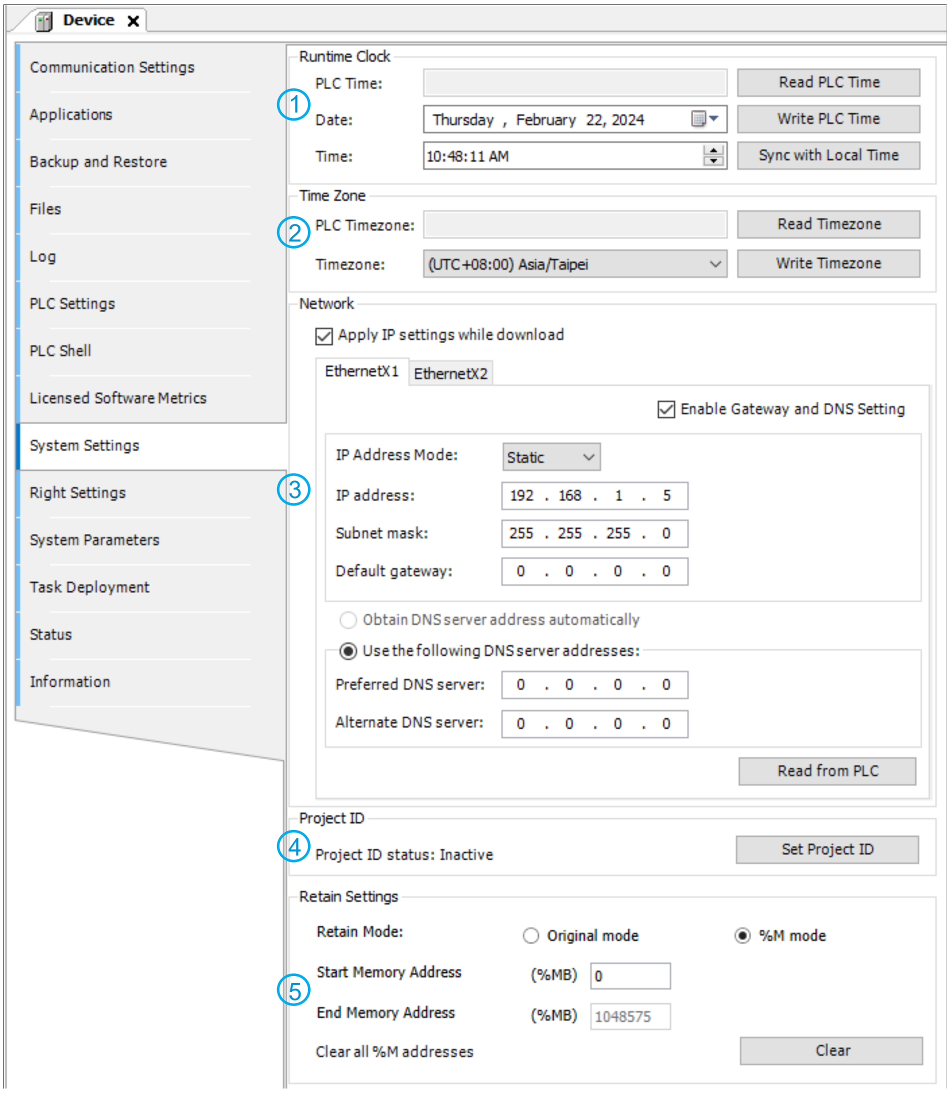
4.2.1.7 PLC Shell

You can use this text-based control monitor for querying specific information from the controller. You can specify device-dependent commands for this and receive the response from the controller in a result window.



4.2.1.8 System Setting

Here you can set up the system settings for the AX-5 Series PLC. Before setting up, make sure that DIADesigner-AX is successfully connected to AX-5 Series PLC. Refer to section 4.2.1.1 for establishing the connection between DIADesigner-AX and AX-5 Series PLC.



PLC Time:

Date: Thursday , February 22, 2024

Time: 10:48:11 AM

PLC Timezone:

Timezone: (UTC+08:00) Asia/Taipei

☑ Apply IP settings while download

☑ Enable Gateway and DNS Setting

IP Address Mode: Static

IP address: 192 . 168 . 1 . 5

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 0 . 0 . 0 . 0

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 0 . 0 . 0 . 0

Alternate DNS server: 0 . 0 . 0 . 0

Project ID status: Inactive

Retain Mode: ○ Original mode ● %M mode

Start Memory Address (%MB) 0

End Memory Address (%MB) 1048575

Clear all %M addresses

① Runtime Clock

Item	Description
PLC Time	Use the button **Read PLC Time** to read the PLC current date and time and the result will be updated here.
Date	Use the button **Write PLC Time** to write the date on DIADesigner-AX (PC) into PLC and the result will be updated here.
Time	Use the button **Sync with Local Time** to write the time on DIADesigner-AX (PC) into PLC and the result will be updated here.

② Time Zone

Item	Description
PLC Timezone	Use the button **Read Timezone** to read the PLC current timezone and the result will be updated here.
Timezone	Use the button **Write Timezone** to write the timezone on DIADesigner-AX (PC) into PLC and the result will be updated here.

③ Network

Item	Description
Apply IP settings while download	The IP address setting is effective once the parameters are downloaded.
Enable Gateway and DNS Settings	You can input your own IP address, Subnet mask, Default gateway and obtain DNS server address automatically or define your own DNS server addresses
IP Address Mode	■ Static: Static IP address ■ DHCP: Dynamic IP address
Read from PLC	Read the network settings from the PLC.

④ Project ID

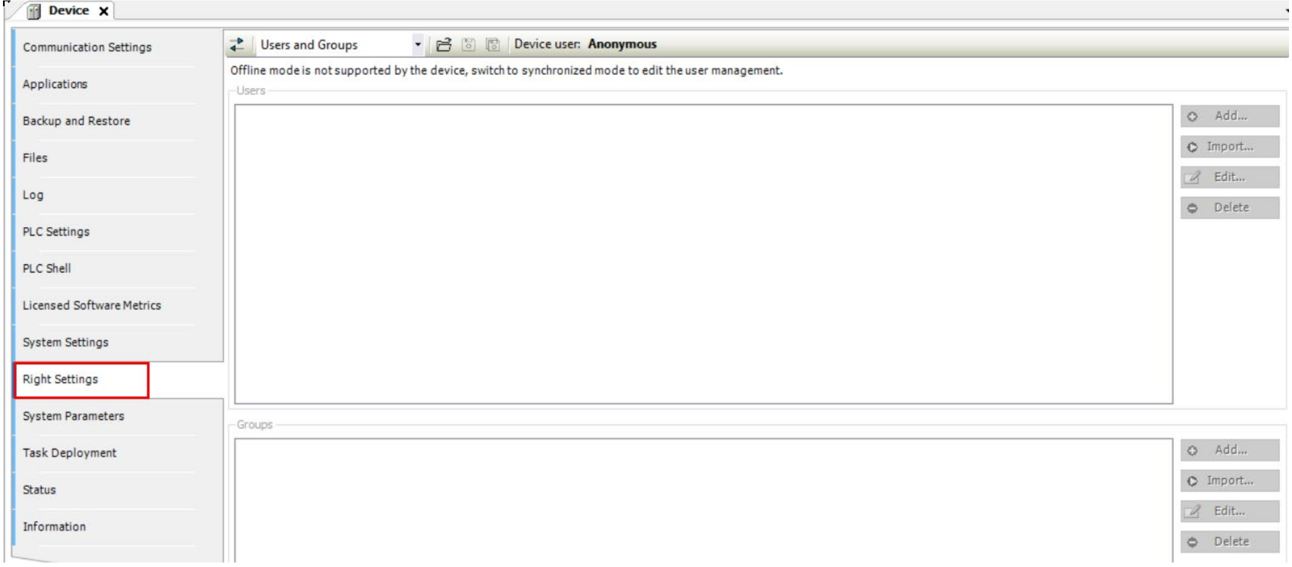
Item	Description
Set Project ID	After enabling Project ID feature, this setting must be complete before downloading projects.

⑤ Retain Settings

Item	Description
Retain Mode	Retain Mode for the variables: ■ **Original mode:** Determined by the retainability of the variable type. ■ **%M mode:** If the variable is in the section of %M, it is retainable. *1 ◆ **Start Memory Address:** If the variable is in the starting address of the %M section (not user-defined). ◆ **End Memory Address:** If the variable is in the ending address of the %M section (not user-defined). ◆ **Clear all %M addresses:** Clear the values in the %M section. *1: Persistent, Persistent Retain, and Retain Persistent variables are directly addressed at the %M variable during compilation when operating in %M mode, ensuring their values persist.

4.2.1.9 Right Settings

The settings include Users and Groups, Access Rights and Symbol Rights.



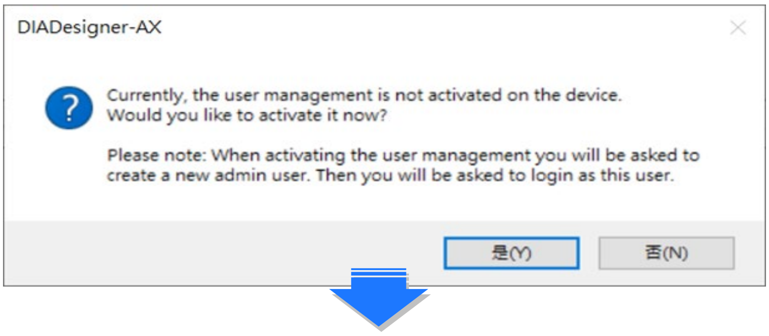
Offline mode is not supported by the device, switch to synchronized mode to edit the user management.

- **Toolbar of the tab**

Item	Description
Synchronization	1. Switches on and off the synchronization between the editor and the user management on the device. 2. If the button is not pressed, then the editor is blank, or it contains a configuration that you loaded from the hard disk. 3. If the button is pressed, then DIADesigner-AX synchronizes the display in the editor continuously with the current user management on the connected device. 4. If you activate the synchronization while the editor contains a user configuration that is not synchronized with the device yet, then you are prompted what should happen to the editor contents. Options: ● Upload from the device and overwrite the editor content: The configuration on the device is loaded into the editor, overwriting the current contents. Download the editor content to the device and overwrite the user management there: The configuration in the editor is transferred to the device and applied there.
Import from disk	Click this button and then to select and import a user management configuration from the file.
Export to disk	Click this button and then to save the current user management configuration as an XML file.
Export all to disk	Click this button and then to save all the user management configurations as an XML file.
Device user	Username of the user currently logged in on the device

- **Users and Groups**

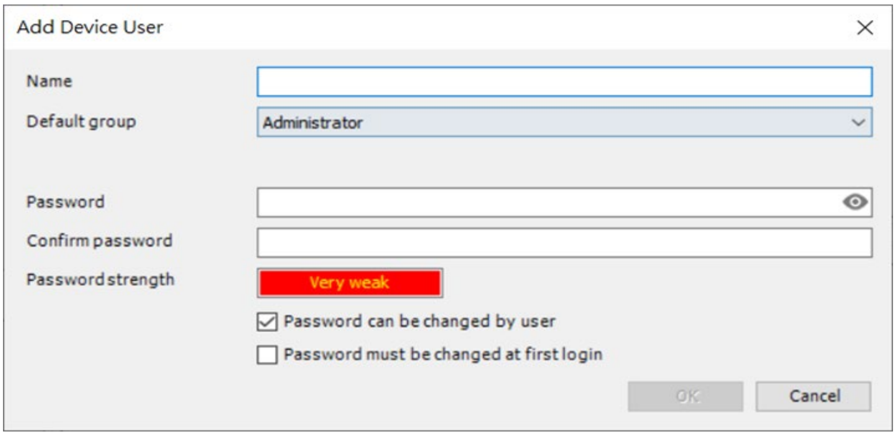
You will need to set up the user account and password for the first time use.



DIADesigner-AX

Currently, the user management is not activated on the device. Would you like to activate it now?

Please note: When activating the user management you will be asked to create a new admin user. Then you will be asked to login as this user.



Add Device User

Name

Default group: Administrator

Password

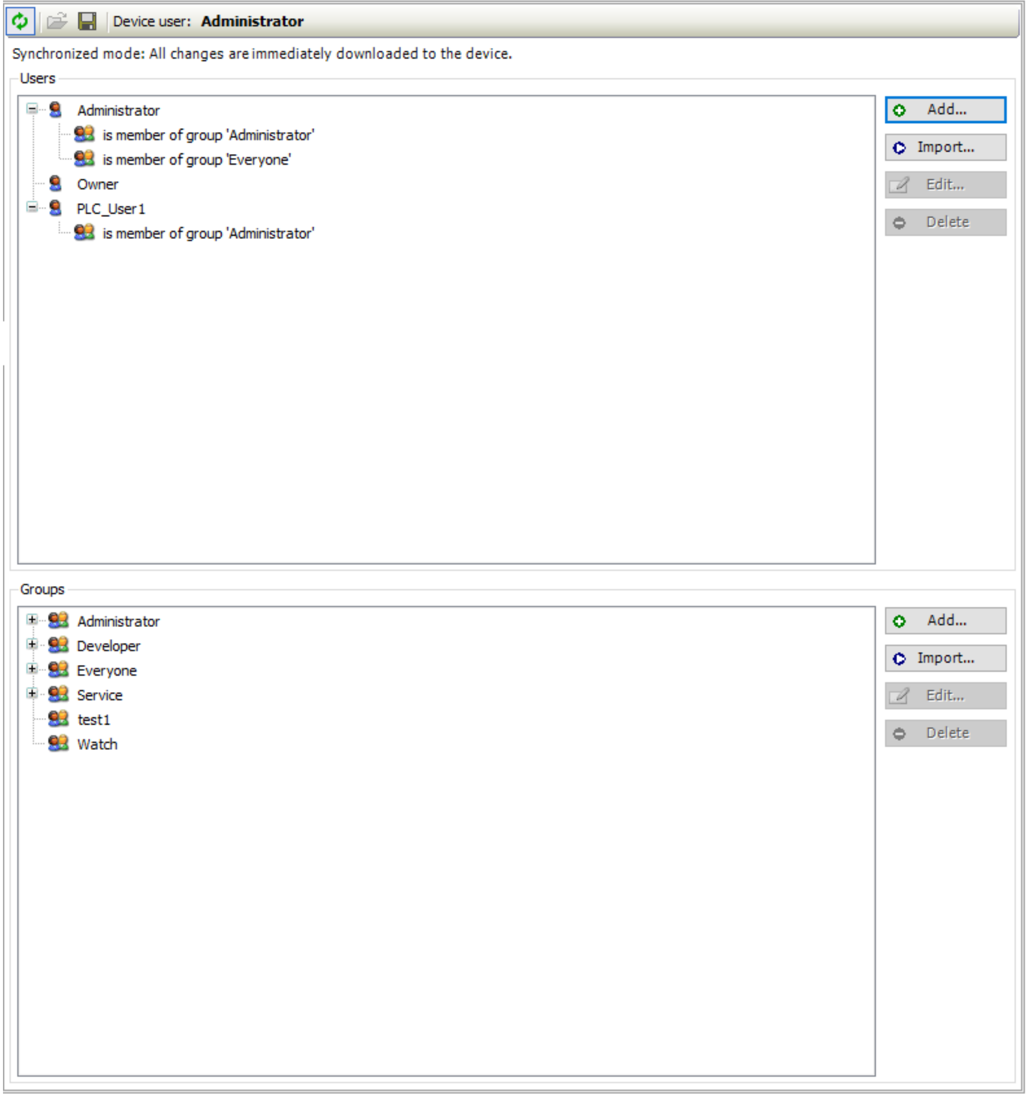
Confirm password

Password strength: Very weak

☑ Password can be changed by user

☐ Password must be changed at first login

- **Account management setting images:**



Device user: **Administrator**

Synchronized mode: All changes are immediately downloaded to the device.

Users

- Administrator
  - is member of group 'Administrator'
  - is member of group 'Everyone'
- Owner
- PLC_User1
  - is member of group 'Administrator'

Groups

- Administrator
- Developer
- Everyone
- Service
- test1
- Watch

- **Toolbar of the tab**

Item	Description
Synchronization	1. Switches on and off the synchronization between the editor and the user management on the device. 2. If the button is not pressed, then the editor is blank, or it contains a configuration that you loaded from the hard disk. 3. If the button is pressed, then DIADesigner-AX synchronizes the display in the editor continuously with the current user management on the connected device. 4. If you activate the synchronization while the editor contains a user configuration that is not synchronized with the device yet, then you are prompted what should happen to the editor contents. Options: • Upload from the device and overwrite the editor content: The configuration on the device is loaded into the editor, overwriting the current contents. • Download the editor content to the device and overwrite the user management there: The configuration in the editor is transferred to the device and applied there.
Import from disk	Click this button and then to select and import a user management configuration from the file.

Item	Description
Export to disk	Click this button and then to save the user management configuration as an XML file.
Device user	Username of the user currently logged in on the device

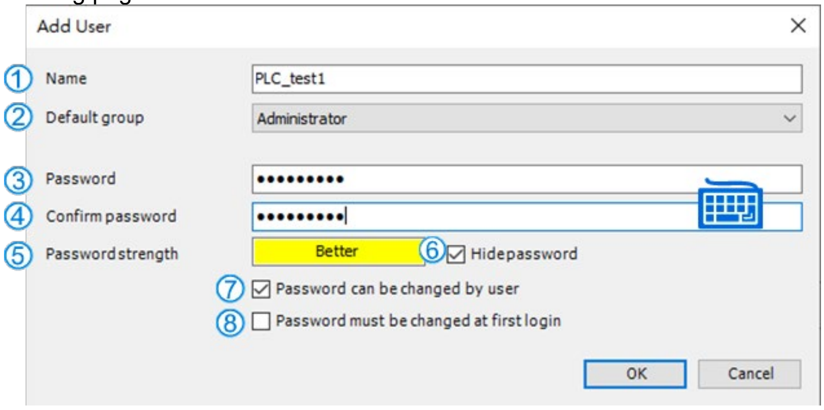
- **Users**

Item	Description
Add	Click this button to create a new user account.*1
Import	Click this button to select the desired entries to import users into the device user management.*2
Edit	Click this button to change the settings of the selected user account.
Delete	Click this button to delete the account of the selected user.

- **Groups**

Item	Description
Add	Click this button to create a new user group.*3
Import	Click this button to select the desired entries to import groups into the device user management.*4
Edit	Click this button to change the settings of the selected group.
Delete	Click this button to delete the selected group.

Note 1: The **Add User** setting page



Add User

① Name: PLC_test1

② Default group: Administrator

③ Password: •••••••••

④ Confirm password: •••••••••

⑤ Password strength: Better

⑥ ☑ Hide password

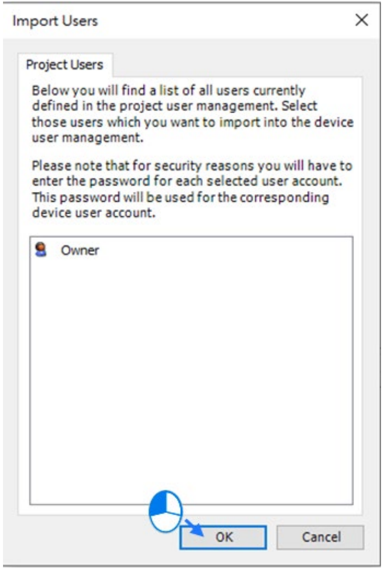
⑦ ☑ Password can be changed by user

⑧ ☐ Password must be changed at first login

	Item	Description
①	**Name**	Username
②	**Default group**	Use the drop-down list to select the default group
③	**Password**	Password
④	**Confirm password**	Confirm password
⑤	**Password strength**	Levels from Very *weak to Very good*
⑥	**Hide password**	☑: The password is shown only with asterisks "*" when it is typed in.
⑦	**Password can be changed by user**	☑: Password can be changed by the user
⑧	**Password must be changed at first login**	☑: Password must be changed at first login

Note 2: The **Import User** setting page

After selected the user from the list, click **OK** to import.

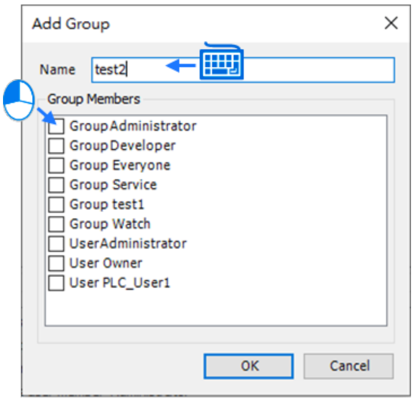


Below you will find a list of all users currently defined in the project user management. Select those users which you want to import into the device user management.

Please note that for security reasons you will have to enter the password for each selected user account. This password will be used for the corresponding device user account.

Note 3: The **Add Group** setting page

Type in the new group name and select the to-be-added group members for this new group and then click **OK**.



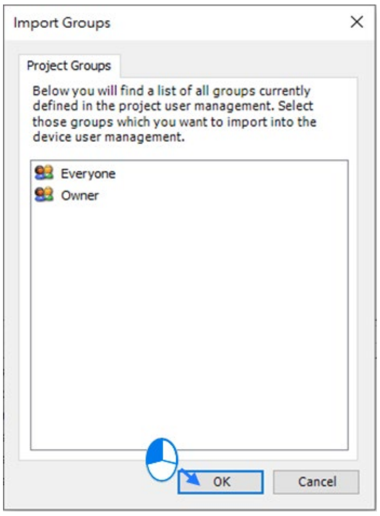
Name test2

Group Members

- ☐ Group Administrator
- ☐ Group Developer
- ☐ Group Everyone
- ☐ Group Service
- ☐ Group test1
- ☐ Group Watch
- ☐ User Administrator
- ☐ User Owner
- ☐ User PLC_User1

Note 4: The **Import Group** setting page

After selected the group from the list, click **OK** to import.



Below you will find a list of all groups currently defined in the project user management. Select those groups which you want to import into the device user management.

- **Access Rights**

Here you can define the device access rights of device users to objects on AX-5 Series PLC. As in the project user management, users must be members of at least one user group and only user groups can be granted certain access rights.

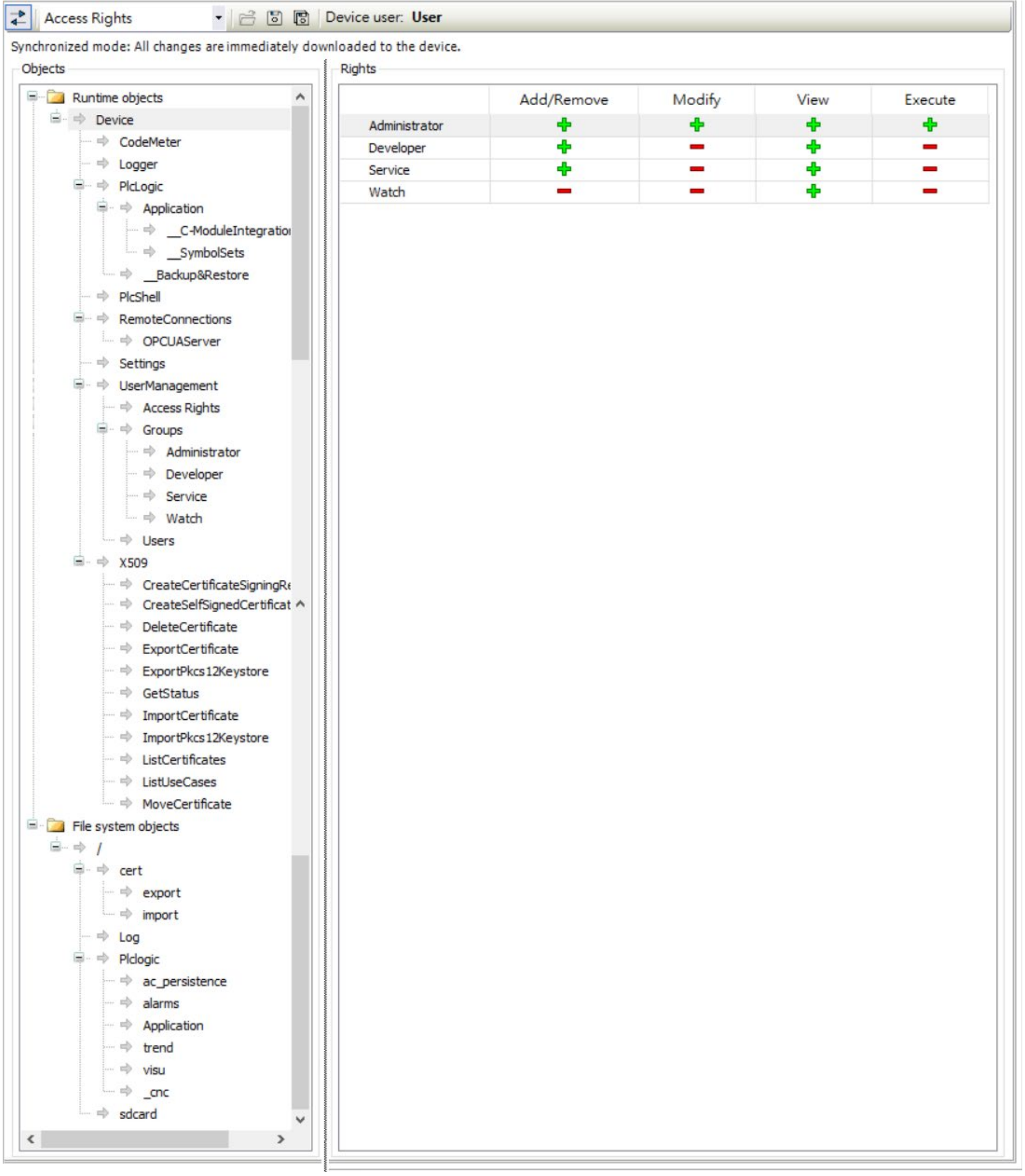
Requirements for the Access Rights tab to be displayed:

- In the DIADesigner-AX options, in the Device editor category, the Show access rights page option must be selected.

Note that this DIADesigner-AX option can be overwritten by the device description.

Requirements for the access rights to be granted to user groups

- A component for the user management must be available on AX-5 Series PLC. That is the primary requirement.
- Users and user groups must be configured on the Users and Groups tab.



Access Rights | Device user: **User**

Synchronized mode: All changes are immediately downloaded to the device.

Rights

	Add/Remove	Modify	View	Execute
Administrator	+	+	+	+
Developer	+	−	+	−
Service	+	−	+	−
Watch	−	−	+	−

- **Objects**

Description

In the tree structure, the objects are listed to which actions can be executed at runtime. The objects are each assigned by their object source and partially sorted in object groups. In the Rights view, you can configure the access options for a user group to a selected object.

Object source (root node)

- File system objects ▸ Device: In these objects, the rights can be granted to folders of the current execution directory of the AX-5 Series PLC.
- Runtime objects ▸ /: In these objects, all objects are managed that have online access in the AX-5 Series PLC and therefore have to control the access rights.

A description of the objects is located in the table. Overview of the objects

Object groups and objects (indented)

Example: Device with child nodes Logger, PlcLogic, Settings, UserManagement.

- **Rights**

Description

In general, the access rights are inherited from the root object (also Device or /) to the sub-objects. This means that if a permission of a user group is denied or explicitly granted to a parent object, then this first affects all child objects.

The table applies for the object that is currently selected in the tree. For every user group, it shows the rights currently configured for the possible actions on this object.

Synchronized mode: All changes are immediately downloaded to the device.

Objects

```
Runtime objects
  Device
    Logger
    PlcLogic
      Application
        __C-ModuleIntegration__
      __Backup&Restore
    PlcShell
```

Rights

	Add/Remove	Modify	View	Execute
Administrator	✚	✚	✚	✚
Developer	✚	▬	✚	▬
Everyone	▬	▬	✚	▬
Service	✚	▬	✚	▬
test_group	✖	✖	✖	✖
Watch	▬	▬	✚	▬

Possible actions on the object:

- **Add/Remove**
- **Modify**
- **View**
- **Execute**

When an object is clicked, a table on the right side shows the access rights of the available user groups for the selected object.

This allows you to quickly see:

- Which access rights are evaluated by an object
- Which user group has which effective rights to which object

Meanings of the symbols

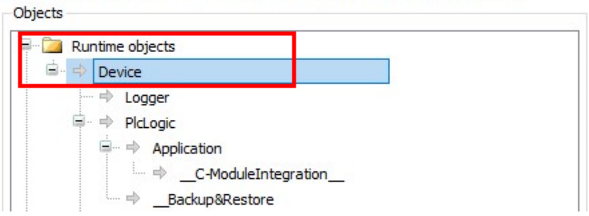
- ✚: Access right granted explicitly
- ▬: Access right denied explicitly
- +: Access right granted through inheritance
- −: Access right denied through inheritance
- ✖: The access right was not granted or denied explicitly and also not inherited by the parent object. Access is not possible.

- No symbol: Multiple objects are selected that have different access rights.

Change the permission by clicking the symbol.

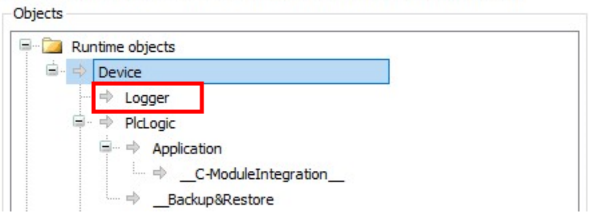
- **Overview**

◆ **Runtime objects > Device**

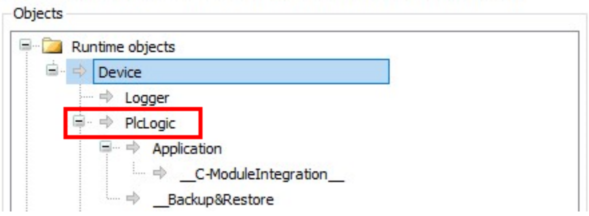


➢ **Device > Logger**

The Logger object on the Access Rights tab was created by the “Logger” component and controls its access rights. The possible access rights for this object can be granted only for the View action.

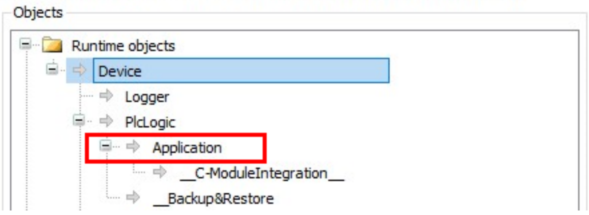


➢ **Device > PlcLogic**



All IEC applications are inserted here automatically as child objects during download. When an application is deleted, it is removed automatically. This allows specific control of online access to the application. Access rights can be assigned centrally over all applications in the PlcLogic. The Administrator and Developer user groups have full access to the IEC applications. The Service and Watch user groups only have read access (for example for read-only monitoring of values).

➢ **PlcLogic > Application**



The following table shows which action is affected in particular when a specific access right is granted for an IEC application.

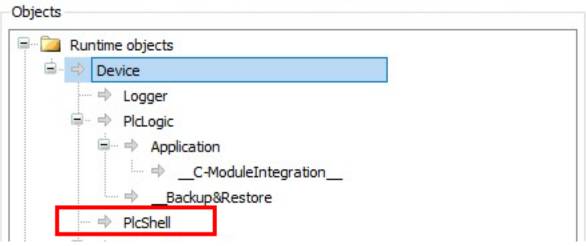
x: Access NOT granted.

●: Access granted.

	Operation	Access rights			
		Add/Remove	**Execute**	**Modify**	**View**
Application	**Login**	●	●	●	x
	Create	x	●	●	●
	Create child object	x	●	●	●
	Delete	x	●	●	●
	Download / online change	x	●	●	●
	Create boot application	x	●	●	●
	Read variable	●	●	●	x
	Write variable	●	●	x	x
	Force variable	●	●	x	x
	Set and delete breakpoint	●	x	x	●
	Set next statement	●	x	x	●
	Read call stack	●	●	●	x
	Single cycle	●	x	●	●
	Switch on flow control	●	x	x	●
	Start / Stop	●	x	●	●
	Reset	●	x	●	●
	Restore retain variables	●	x	●	●
	Save retain variables	●	●	●	x

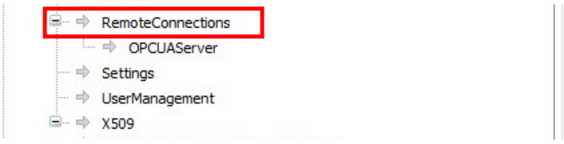
- **PlcShell**

  Only the Modify permission is evaluated at this time. This means that only when the Modify permission has been granted to a user group can PLC shell commands also be evaluated.



- **RemoteConnections**

  Additional external connections to the AX-5 Series PLC can be configured below this node. Currently, access to the OPC UA server can be configured here.

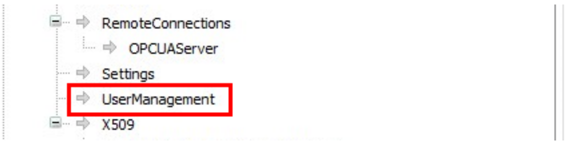


- **Settings**

  This is the online access to the configuration settings of the AX-5 Series PLC. By default, access to Modify is granted only to the administrator.

- **UserManagement**

  This is the online access to the user management of AX-5 Series PLC. By default, read/write access is granted only to the administrator.



- **X509**

  This controls the online access to the X.509 certificates. Two types of access are distinguished here:

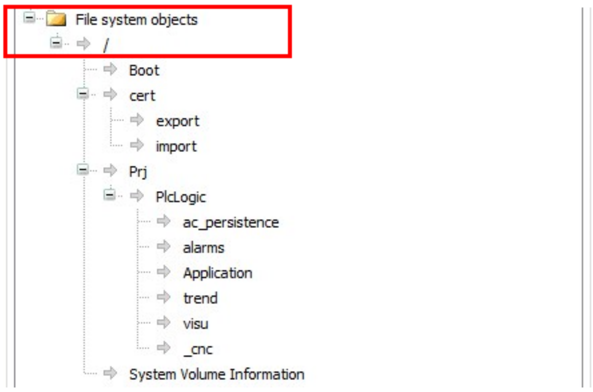
  Read (View)

  Write (Modify)

  Every operation is assigned to one of these two access rights. Each operation is inserted as a child object below X509. Therefore, access per operation can now be fine-tuned even more.

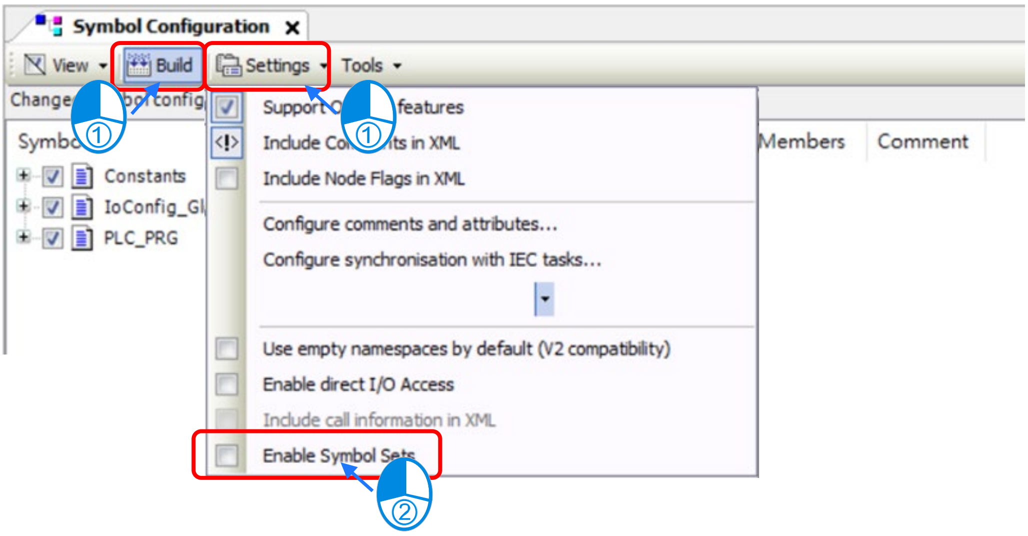
- **File system objects > /**

  All folders from the execution path of the AX-5 Series PLC are inserted below the "/" file system object. This allows you to grant specific rights to each folder of the file system.

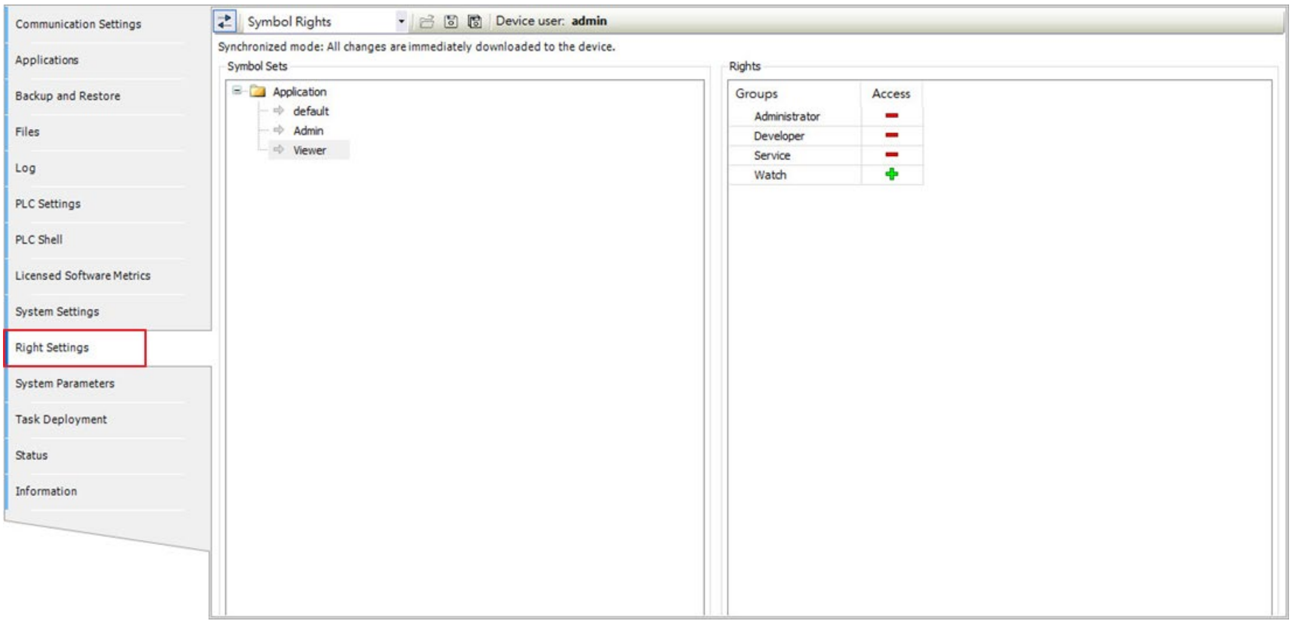


- **Symbol Rights**

Here you can define the access rights of different user groups to the individual symbol sets available on the AX-5 Series PLC. Before that you need to enable the Symbol set under Symbol Configuration and then download the project to the controller. See the following examples for reference.



Use the icons 🔄 or ⇄ for synchronization. Upload the controller current parameters to the DIADesigner-AX3 and then you can set up the access right. You can find the user groups in the column of Rights. And when you select the Application under Symbol Sets you can see the corresponding access right for different user groups in the column of Rights. ✚: Access granted; ▬: Access not granted. You can change the access rights by double-clicking the symbol. Click the 💾 button to save the current access configuration to an XML file. The file type is Device symbol management files (*.dsm). Click the 📂 button to read a file like this from the computer.



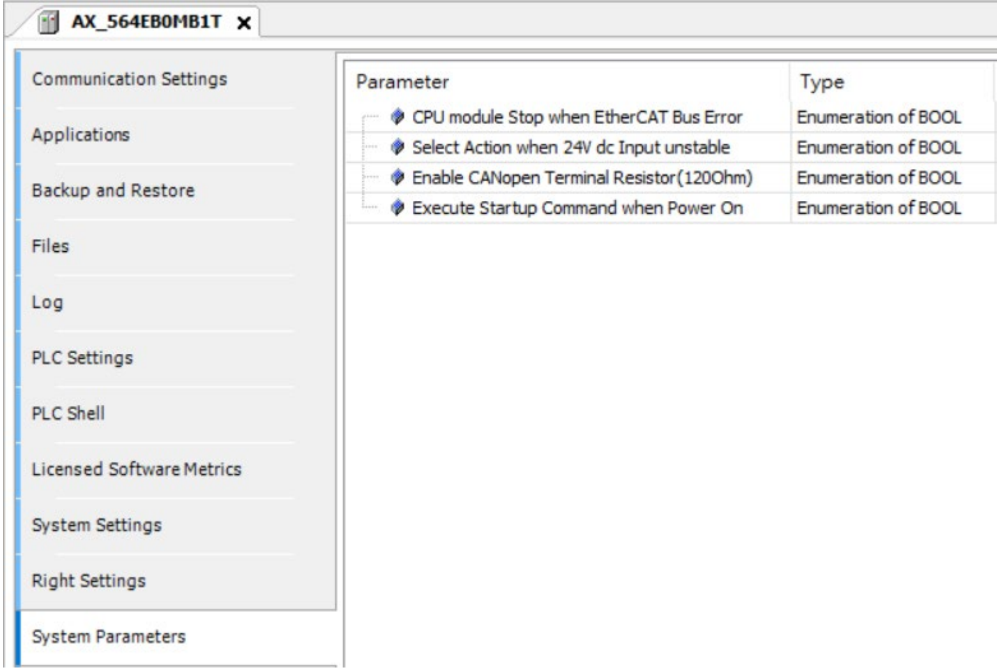
Device user: admin

Synchronized mode: All changes are immediately downloaded to the device.

Groups	Access
Administrator	▬
Developer	▬
Service	▬
Watch	✚

4.2.1.10 System Parameters

Here you can set up the various parameters for the AX-5 Series PLC. Note that settings on this page do NOT support on-line editing.



AX_564EB0MB1T

- Communication Settings
- Applications
- Backup and Restore
- Files
- Log
- PLC Settings
- PLC Shell
- Licensed Software Metrics
- System Settings
- Right Settings
- System Parameters

Parameter	Type
CPU module Stop when EtherCAT Bus Error	Enumeration of BOOL
Select Action when 24V dc Input unstable	Enumeration of BOOL
Enable CANopen Terminal Resistor(120Ohm)	Enumeration of BOOL
Execute Startup Command when Power On	Enumeration of BOOL

- **CPU module Stop when EtherCAT Bus Error**

  The parameter sets whether the CPU and other normal modules can keep running if an error occurs on the extension modules or remote modules.

  ➢ Stop: The CPU module stops running and sends an error. After resolving the problem, the CPU module resumes operation.

  ➢ Keep Run (default): The CPU module and other normal modules keep running but records the warning message.

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

  What to do when the 24 Vdc power is unstable

  ➢ Continue Running when power stable (default): The CPU stops and waits till the power is stable and then the CPU resumes operation.

  ➢ Into Error Status: The CPU stops and ERROR LED blinks; even after the power is stable again, the CPU still stays stop.

- **Enable CAN Terminal Resistor (120 ohm)**

  The built-in CANopen terminal resistor has a resistance of 120 Ω, and it can be activated or deactivated based on the specific operating conditions.

  ➢ Disable: The CANopen terminal resistor is deactivated.

  ➢ Enable (default): The CANopen terminal resistor is activated.

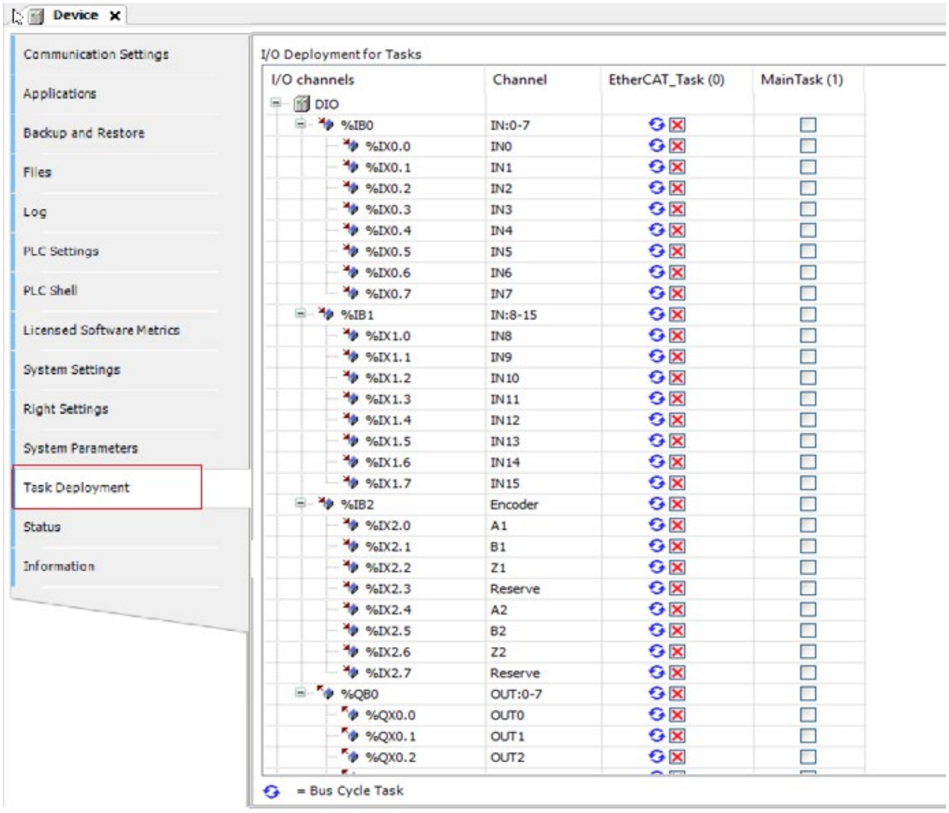
  *To configure the CANopen terminal resistor, you must enable this function and also configure a terminal resistor at the end of the CANopen bus.

- **Execute Startup Command when Power On**

  The parameter sets whether to execute the Startup Command stored on the SD card once the CPU module starts running.

4.2.1.11 Task Deployment

Here displays a table of inputs and outputs and their assignments to the defined tasks and bus cycle task. You can search for the relevant information here. The information is refreshed after the project is compiled and downloaded to the CPU. If the search result is not as expected, you can use the information to troubleshoot.



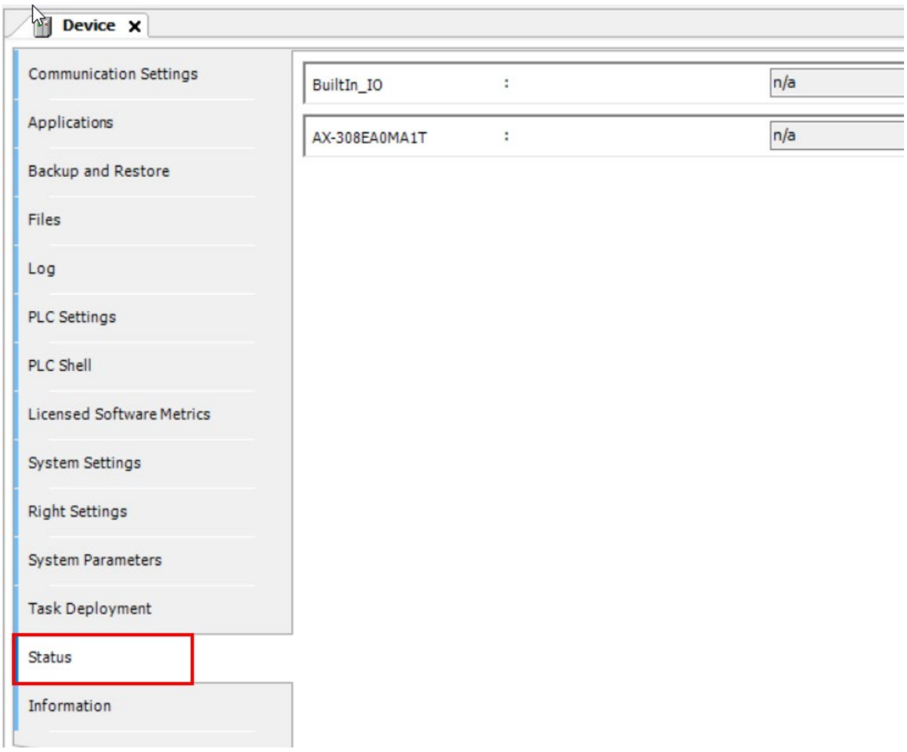
I/O channels	Channel	EtherCAT_Task (0)	MainTask (1)
DIO			
%IB0	IN:0-7	↻☒	☐
%IX0.0	IN0	↻☒	☐
%IX0.1	IN1	↻☒	☐
%IX0.2	IN2	↻☒	☐
%IX0.3	IN3	↻☒	☐
%IX0.4	IN4	↻☒	☐
%IX0.5	IN5	↻☒	☐
%IX0.6	IN6	↻☒	☐
%IX0.7	IN7	↻☒	☐
%IB1	IN:8-15	↻☒	☐
%IX1.0	IN8	↻☒	☐
%IX1.1	IN9	↻☒	☐
%IX1.2	IN10	↻☒	☐
%IX1.3	IN11	↻☒	☐
%IX1.4	IN12	↻☒	☐
%IX1.5	IN13	↻☒	☐
%IX1.6	IN14	↻☒	☐
%IX1.7	IN15	↻☒	☐
%IB2	Encoder	↻☒	☐
%IX2.0	A1	↻☒	☐
%IX2.1	B1	↻☒	☐
%IX2.2	Z1	↻☒	☐
%IX2.3	Reserve	↻☒	☐
%IX2.4	A2	↻☒	☐
%IX2.5	B2	↻☒	☐
%IX2.6	Z2	↻☒	☐
%IX2.7	Reserve	↻☒	☐
%QB0	OUT:0-7	↻☒	☐
%QX0.0	OUT0	↻☒	☐
%QX0.1	OUT1	↻☒	☐
%QX0.2	OUT2	↻☒	☐

↻ = Bus Cycle Task

↻	The task defined as a Bus cycle task in the PLC Settings of the device
☒	For inputs and outputs that are written or read by a task.

4.2.1.12 Status

Here you can find the device status information, for example 'Running' or 'Stopped,' and specific diagnostic messages from the respective device, also information about the card used and the internal bus system.

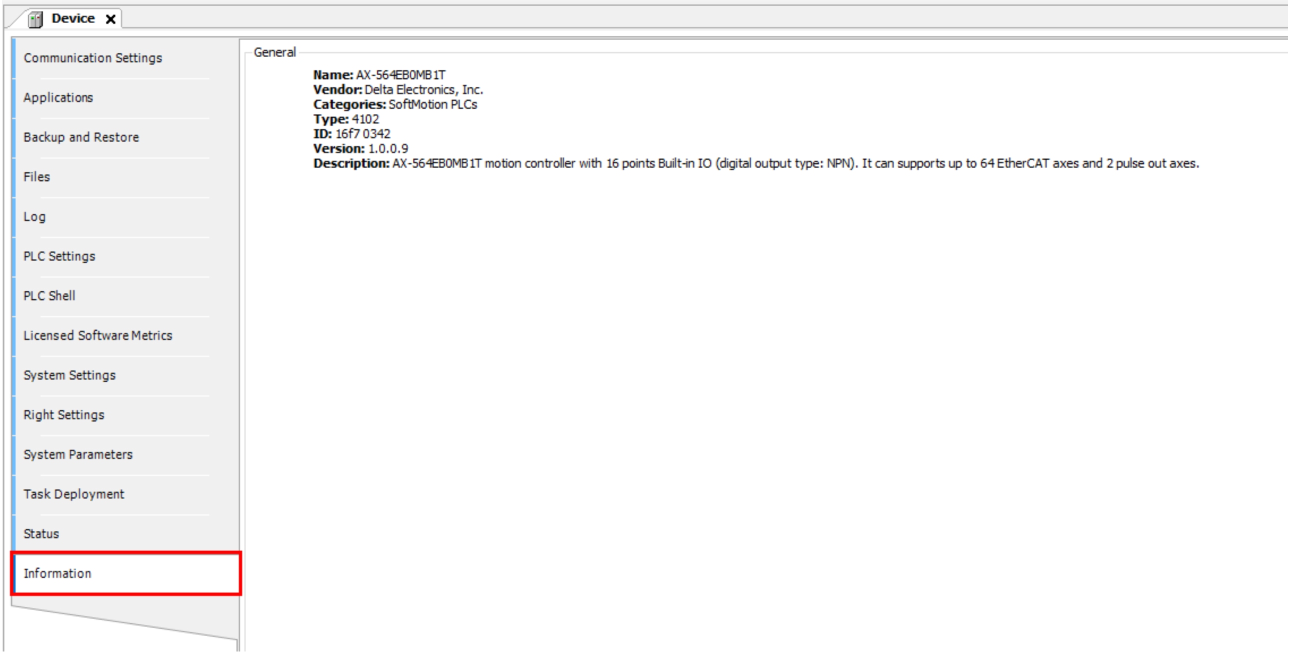


BuiltIn_IO : n/a

AX-308EA0MA1T : n/a

4.2.1.13 Information

Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



Name: AX-564EB0MB1T
Vendor: Delta Electronics, Inc.
Categories: SoftMotion PLCs
Type: 4102
ID: 16f7 0342
Version: 1.0.0.9
Description: AX-564EB0MB1T motion controller with 16 points Built-in IO (digital output type: NPN). It can supports up to 64 EtherCAT axes and 2 pulse out axes.

4.3 Data Type and Variables

4.3.1 Data Type

Data Type	Minimum Value	Maximum Value	Data Width
BOOL	FALSE	TRUE	1 bit
BYTE	0	255	8 bit
WORD	0	65535	16 bit
DWORD	0	4294967295	32 bit
LWORD	0	2^{64}-1	64 bit
SINT	-128	127	8bit
USINT	0	255	8 bit
INT	-32768	32767	16 bit
UINT	0	65565	16 bit
DINT	-2147483648	2147483647	32 bit
UDINT	0	4294967295	32 bit
LINT	-2^{63}	2^{63}-1	64 bit
ULINT	0	2^{64}-1	64 bit
REAL	-3.402823E+38	3.402823E+38	32 bit
LREAL	-1.7976931348623157E+308	1.7976931348623157E+308	64 bit
TIME	T#0ms	T#49d17h2m47s295ms	32 bit
LTIME	LTIME#0ns	LTIME#213503d23h34n33s 709ms551us615ns	64 bit
TIME_OF_DAY (TOD)	TOD#00:00:00.000	TOD#23:59:59.999	32 bit
DATE	D#1970-1-1 (01/01/70)	DATE#2106-2-7 (February 07, 2106)	32 bit
DATE_AND_TIME	DT#1979-1-1-00:00:00 (01/01/1970 00:00:00)	DT#2106-2-7-6:28:15 (February 07, 2106 6:28:15)	32 bit
STRING	ASCII format (8 bit): up to 255 characters		
WSTRING	Unicode format (16 bit): no limit on the length		

4.3.2 Variables

Rules for identifiers of variables:

- No spaces or special characters
- Not case sensitive (For example, Var0 and VAR0 are seen as the same variable)
- No multiple consecutive underscores (For example, b__Var0 is not permitted)

Rules for multiple use of identifiers

- Local variable cannot be declared more than one time.
- If a local variable and a global variable share the same name, the local variable has priority within the POU.
- Variables with the same name can be declared in different global variables list.

  (For example, globe_list1.bvar and globe_list2.bvar can co-exist in two different global variables lists.)

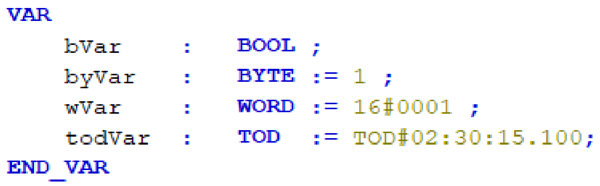
Comments

- Single comment: the symbol // indicates a single comment, for example: // Variable Define
- Multiple comments: the symbol (* XX : XX *) indicates multiples comments from XX to XX, for example (* Variable Define : Variable Define*)

4.3.2.1 Declaration of Variables

In DIADesigner-AX projects you can declare variables in the following methods.

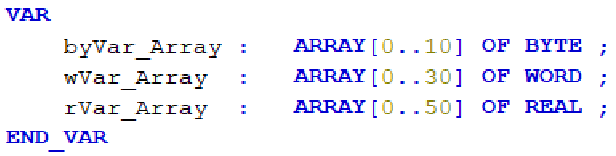
Syntax: <Variable Name> : <Data Type> := <Initialization> ;
Example:



```
VAR
    bVar     :   BOOL ;
    byVar    :   BYTE := 1 ;
    wVar     :   WORD := 16#0001 ;
    todVar   :   TOD  := TOD#02:30:15.100;
END_VAR
```

Array
Syntax : <Variable Name> : ARRAY[0..N] OF <Data Type>

Example:



```
VAR
    byVar_Array :   ARRAY[0..10] OF BYTE ;
    wVar_Array  :   ARRAY[0..30] OF WORD ;
    rVar_Array  :   ARRAY[0..50] OF REAL ;
END_VAR
```

4.3.2.2 Address Assignments

In AX-5 Series, there are three ranges in the memory area, including I (input memory range), Q (output memory range) and M (flag memory range). You can use specific character strings to express memory position and size. For the M flag memory range in AX-5 Series PLC, you cannot manually use the bit operation when in online mode.

Syntax: %<Memory Area Prefix><Size Prefix><Memory Position>

Memory Area	Description	Range
I	Input Memory Range	8 KB
Q	Output Memory Range	8 KB
M	Flag Memory Range	512 KB

Size Prefix	Data Type	Data Width
X	--	1 bit
B	Byte	8 bit
W	Word	16 bit
D	DWord	32 bit
L	LWord	64 bit

- **Memory Area**

The numbering that you use for addressing the memory position depends on the target system. Before specifying the address value in the memory area, you need to know the mapping corresponding relationship of devices to prevent the overlapping memory ranges. See the table below for reference.

<table>
<tr><th colspan="8">Memory Area</th></tr>
<tr><td>X0.63 to X0.56</td><td>X0.55 to X0.48</td><td>X0.47 to X0.40</td><td>X0.39 to X0.32</td><td>X0.31 to X0.17</td><td>X0.23 to X0.16</td><td>X0.15 to X0.8</td><td>X0.7 to X0.0</td></tr>
<tr><td>X7.7 to X7.0</td><td>X6.7 to X6.0</td><td>X5.7 to X5.0</td><td>X4.7 to X4.0</td><td>X3.7 to X3.0</td><td>X2.7 to X2.0</td><td>X1.7 to X1.0</td><td>X0.7 to X0.0</td></tr>
<tr><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr>
<tr><td colspan="2">W3</td><td colspan="2">W2</td><td colspan="2">W1</td><td colspan="2">W0</td></tr>
<tr><td colspan="4">D1</td><td colspan="4">D0</td></tr>
<tr><td colspan="8">L0</td></tr>
</table>

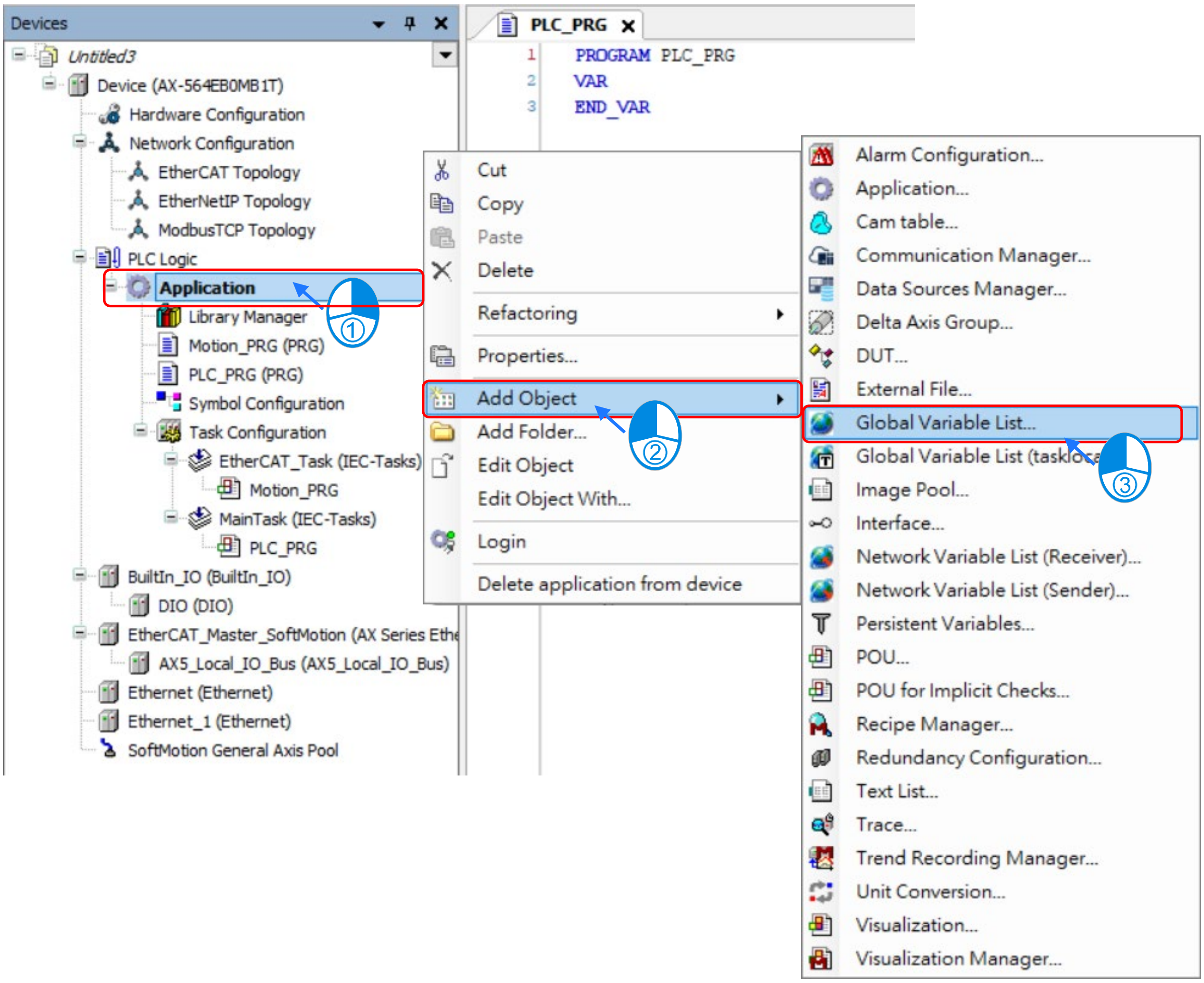
- **Example**

Address	Description
%QX7.5	Single bit address of the output bit 7.5
%IW215	Word address of the input word 215
%QB7	Byte address of the output byte 7
%MD48	Address of a double word at memory position 48 in flag memory
VAR wVar0 AT %IW0 : WORD; END_VAR	Variable declaration with address information of an input word
VAR bVar0 AT IX7.5 : BOOL; END_VAR	Boolean variable declaration with address information of an input bit X7.5.

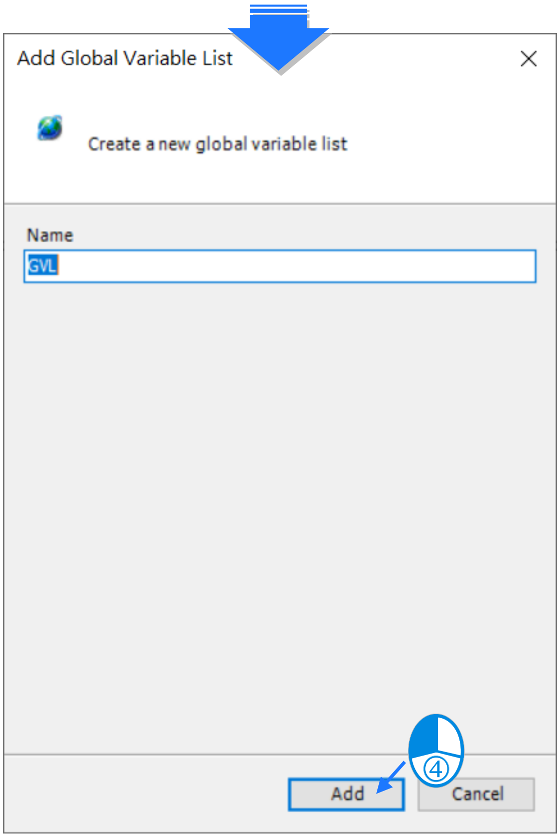
4.3.2.3 Variables

- **Global Variables**

If a variable that is declared in the POU, it is a local variable and it can only be used in the same POU. If a variable that is declared in the global variable list, it is a global variable and it can used in any POU.



```
PROGRAM PLC_PRG
VAR
END_VAR
```

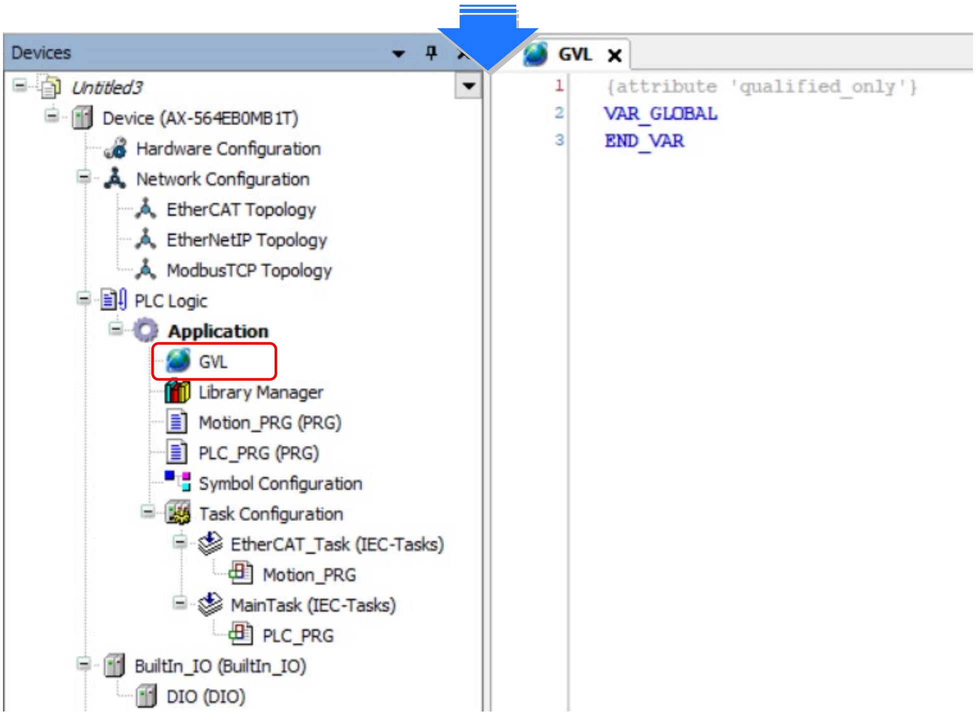


Add Global Variable List

Create a new global variable list

Name

GVL



Devices

- Untitled3
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
      - EtherCAT Topology
      - EtherNetIP Topology
      - ModbusTCP Topology
    - PLC Logic
      - **Application**
        - GVL
        - Library Manager
        - Motion_PRG (PRG)
        - PLC_PRG (PRG)
        - Symbol Configuration
        - Task Configuration
          - EtherCAT_Task (IEC-Tasks)
            - Motion_PRG
          - MainTask (IEC-Tasks)
            - PLC_PRG
    - BuiltIn_IO (BuiltIn_IO)
      - DIO (DIO)

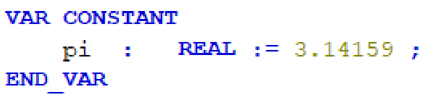
GVL

```
{attribute 'qualified_only'}
VAR_GLOBAL
END_VAR
```

- **Constant Variables**

You can declare a variable as a constant variable. Constant variables can be accessed as read-only and without assigning an initialization value.

Declaration of Constant Variables



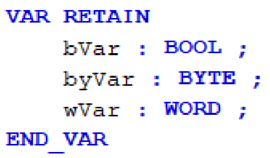
```
VAR CONSTANT
    pi  :   REAL := 3.14159 ;
END_VAR
```

- **Retain Variables**

You can declare a variable as retentive or use retain / persistent variable directly. Refer to the table below for differences among variable, retain variable and persistent variable.

	Initialize				
	Reboot PLC	**Reset warm**	**Reset cold**	**Download**	**Reset Origin**
Variable	O	O	O	O	O
Retain Variable	X	X	O	O	O
Persistent Variable	X	X	X	X	O

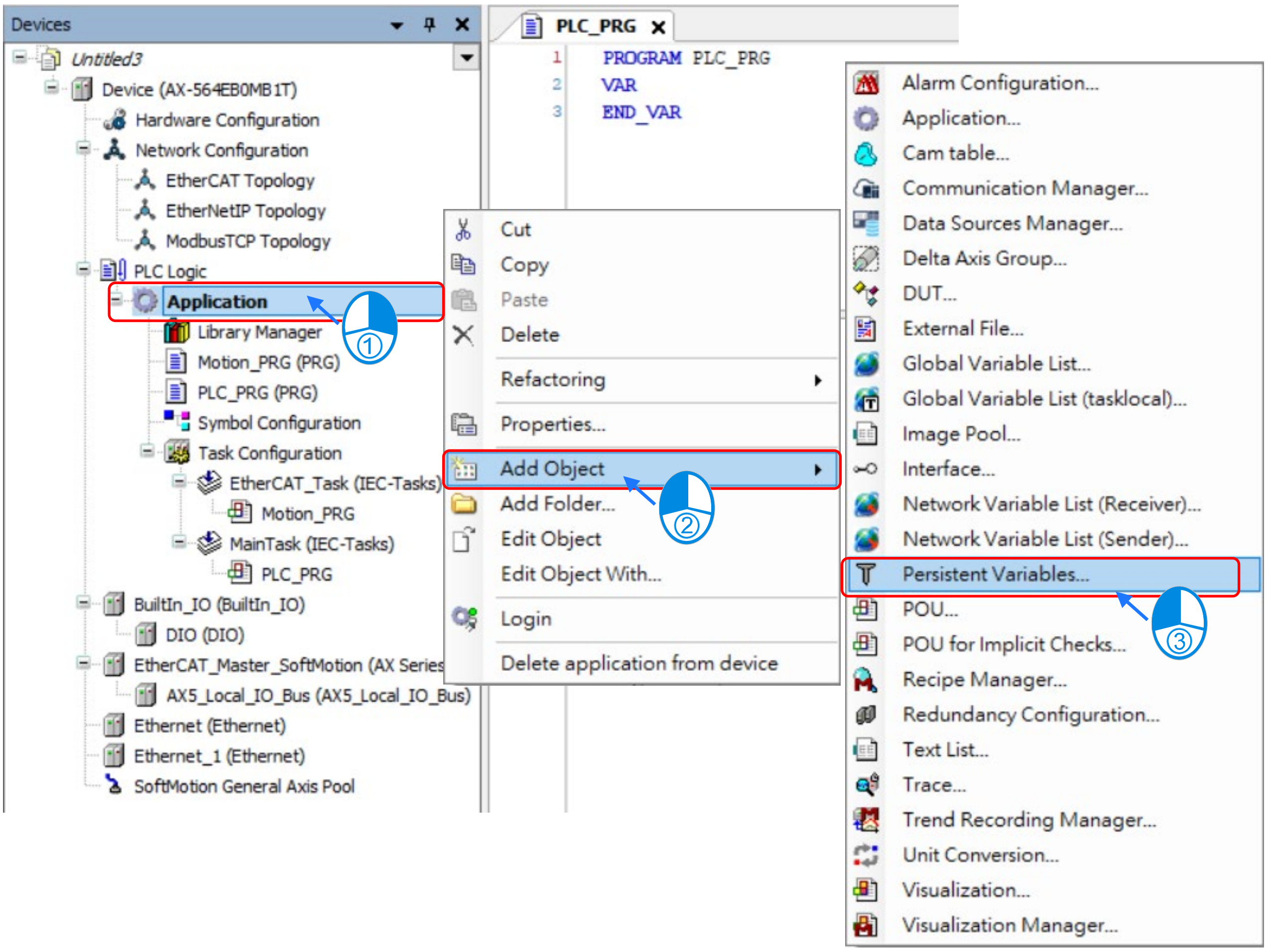
Declaration of Retain Variables



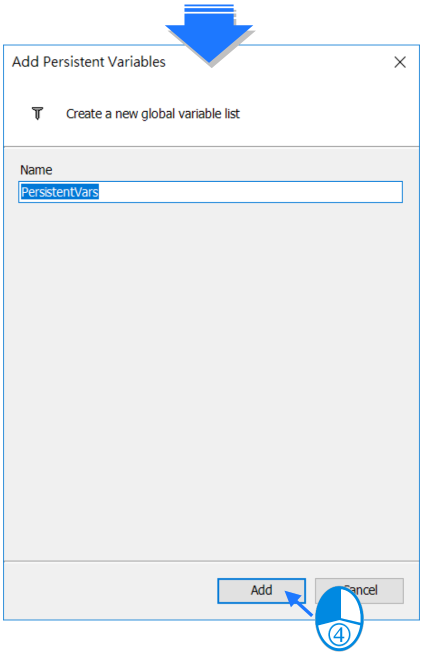
```
VAR RETAIN
    bVar : BOOL ;
    byVar : BYTE ;
    wVar : WORD ;
END_VAR
```

You can declare the Persistent Variable / Retain Persistent Variable / Persistent Retain Variable in the Persistent Variable Object and the results are the same.

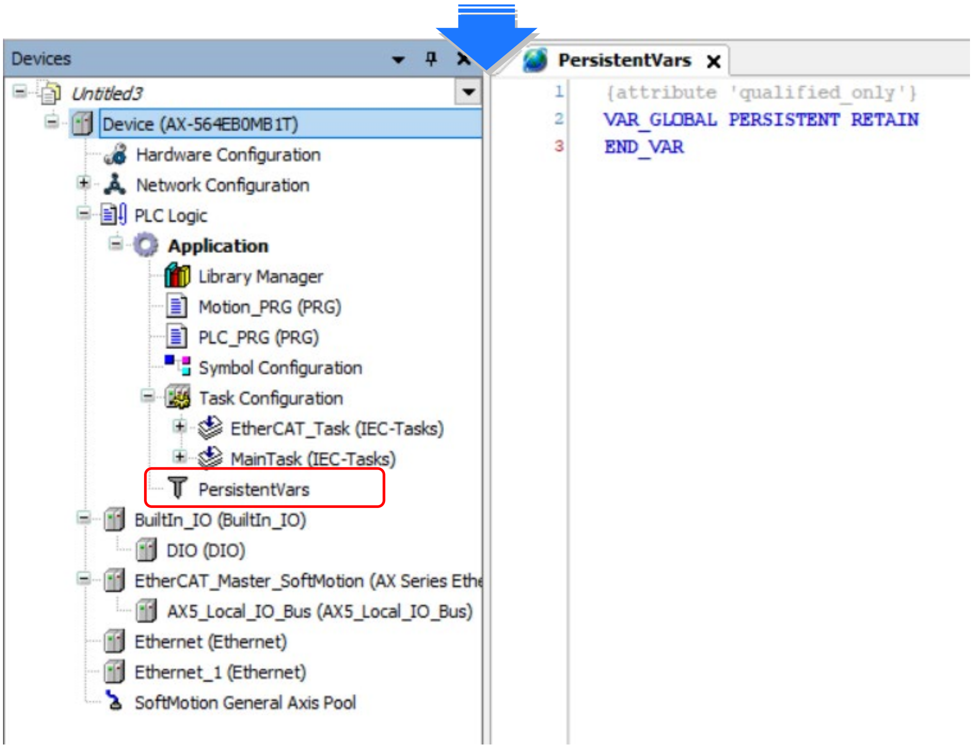
Persistent Variable List:



```
PROGRAM PLC_PRG
VAR
END_VAR
```

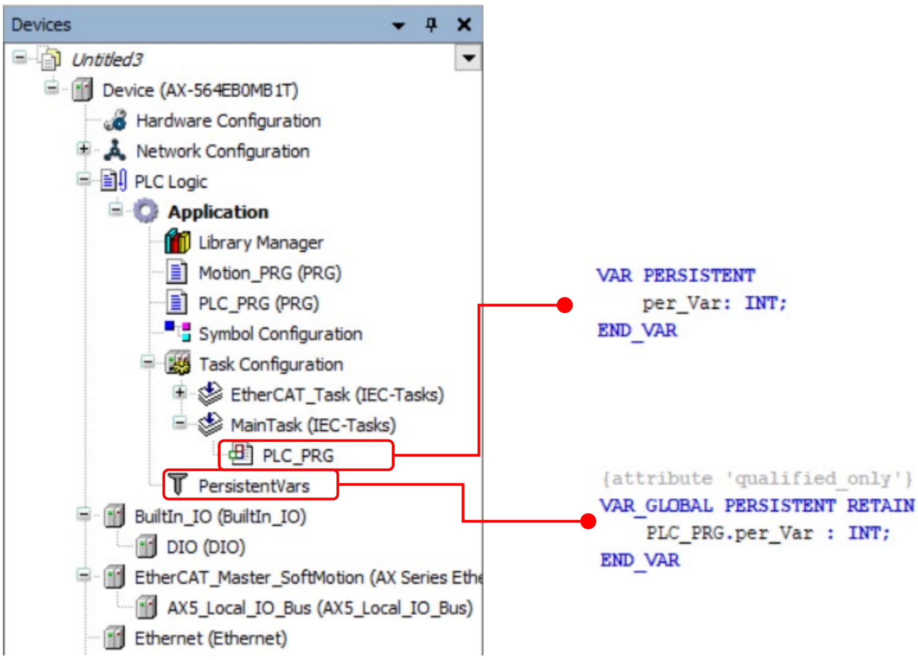


Name: PersistentVars



```
{attribute 'qualified_only'}
VAR_GLOBAL PERSISTENT RETAIN
END_VAR
```

If you need to declare a local variable as persistent, you need to add the variable instance path in the persistent variable list.



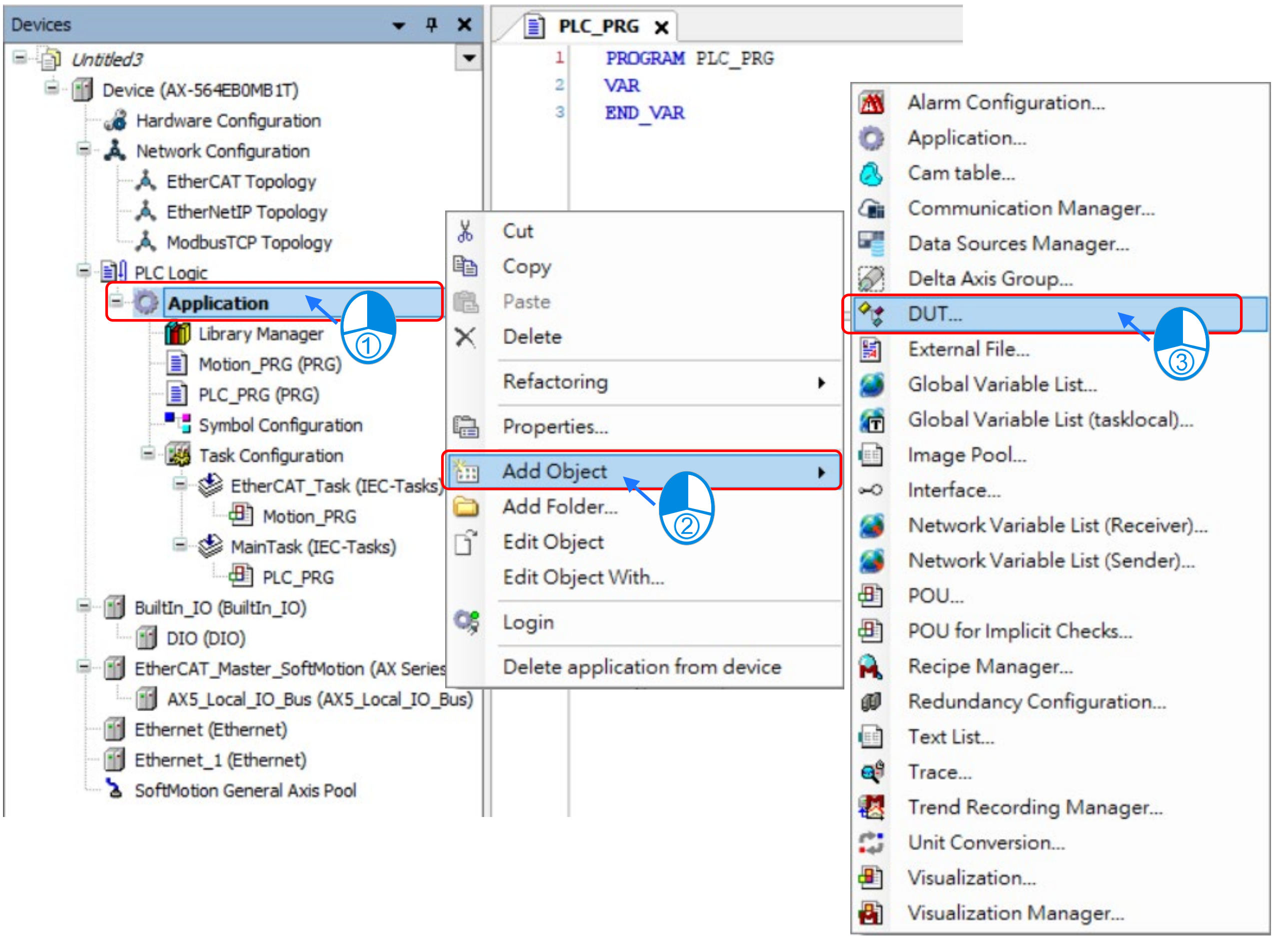
```
VAR PERSISTENT
    per_Var: INT;
END_VAR
```

```
{attribute 'qualified_only'}
VAR_GLOBAL PERSISTENT RETAIN
    PLC_PRG.per_Var : INT;
END_VAR
```

4.3.2.4 User-defined Data Types

You can create your own data type, DUT (Data Type Unit) or UDT (User-defined Data Type), by clicking ADD Object and selecting DUT. Four data types can be created, including Structure, Enumeration, Alias, and Union.

DUT:



```
PROGRAM PLC_PRG
VAR
END_VAR
```

- **Structure**：

A structure is a compound data type used for grouping simple data types or other compound data types.
Syntax:

TYPE <Structure Name>:

STRUCT

<Variable Declaration 1>

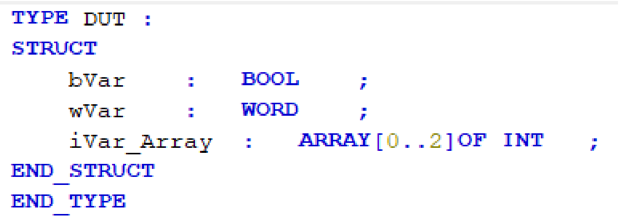
...

<Variable Declaration n>

END_STRUCT

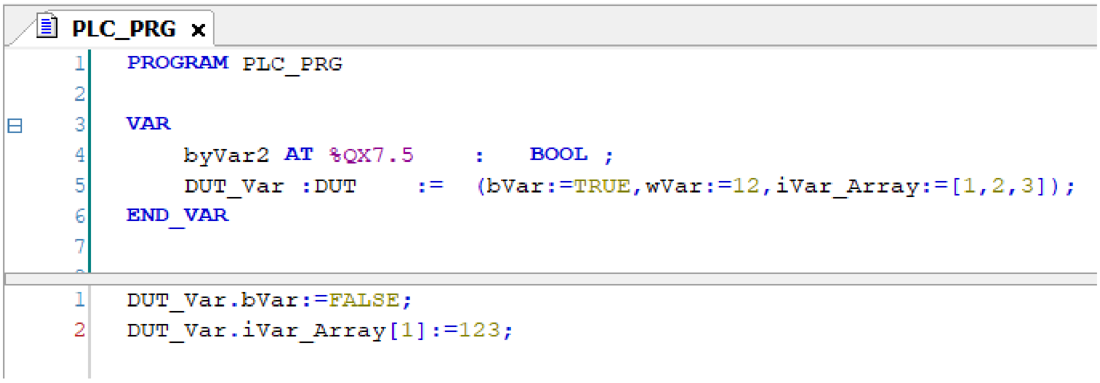
END_TYPE

Example:



```
TYPE DUT :
STRUCT
    bVar        :   BOOL     ;
    wVar        :   WORD     ;
    iVar_Array  :   ARRAY[0..2]OF INT    ;
END_STRUCT
END_TYPE
```

Applications:



```
PROGRAM PLC_PRG

VAR
    byVar2 AT %QX7.5     :   BOOL ;
    DUT_Var :DUT     :=  (bVar:=TRUE,wVar:=12,iVar_Array:=[1,2,3]);
END_VAR

```

```
DUT_Var.bVar:=FALSE;
DUT_Var.iVar_Array[1]:=123;
```

- **Enumeration**：

An enumeration is used to map a set of names to numeric values. Enumerated data types help make the code more self-documenting and make program listing more readable.

Syntax:

TYPE <Enumeration Name> :

(

<First Component Declaration>:= Component Declaration,

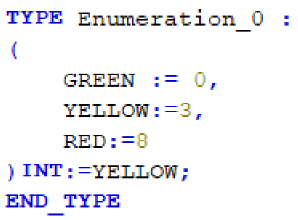
… ,

< Last Component Declaration >:= Component Declaration

) <Basic Data Type> := Default Variable Initialization;

END_TYPE

Example:



```
TYPE Enumeration_0 :
(
    GREEN  := 0,
    YELLOW:=3,
    RED:=8
) INT:=YELLOW;
END_TYPE
```

- **Alias：**

Alias is a scalar data type for a variable that can save a single value and self-define the data type.

Example:

TYPE <Alias Name> : STRING(20); END_TYPE

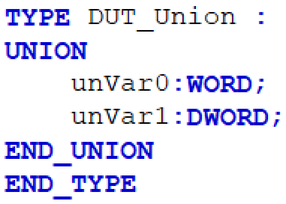
- **Union：**

Union is a data structure that contains different data types. All components have the same amount of memory.

Syntax:

```
TYPE <Union Name>:
UNION
    <Variable Declaration 1>
    ...
    <Variable Declaration n>
END_UNION
END_TYPE
```

Example:



```
TYPE DUT_Union :
UNION
    unVar0:WORD;
    unVar1:DWORD;
END_UNION
END_TYPE
```

4.3.2.5 Timing for the Variable to be Cleared to Zero

For different types of variables, the timing to clear the variables to zero is various. Find the various timings below for the variables to be cleared to zero under various occasions.

- **Retain Mode：Original mode**

Action	VAR	VAR Retain	VAR Retain Persistent
Online Change	●	●	●
Reboot PLC	○	●	●
Reset Warm	○	●	●
Reset Cold	○	○	●
Download	○	○	●
Reset Origin	○	○	○

- **Retain Mode：%M mode**

Action	VAR	VAR Retain	VAR Retain Persistent
	The variable address is in the retainable area.		
Online Change	●	●	●
Reboot PLC	●	●	●
Reset Warm	●	●	●
Reset Cold	●	●	●
Download	●	●	●
Reset Origin	○	○	○

Action	VAR	VAR Retain	VAR Retain Persistent
	The variable address is out of the retainable area.		
Online Change	●	●	●
Reboot PLC	○	●	●
Reset Warm	○	●	●
Reset Cold	○	○	●
Download	○	○	●
Reset Origin	○	○	○

● = Value retained

○ = Clear to zero

*Note: If there is no function of retained values, default values would be effective.

4.3.2.6 Timing for the Default Value to be Effective

- **Retain Mode：Original mode**

Action	VAR	VAR Retain	VAR Retain Persistent
Online Change	●	●	●
Reboot PLC	○	●	●
Reset Warm	○	●	●
Reset Cold	○	○	●
Download	○	○	●
Reset Origin	○	○	○

- **Retain Mode：%M mode**

Action	VAR	VAR Retain	VAR Retain Persistent
	The variable address is in the retainable area.		
Online Change	●	●	●
Reboot PLC	●	●	●
Reset Warm	●	●	●
Reset Cold	●	●	●
Download	●	●	●
Reset Origin	○	○	○

Action	VAR	VAR Retain	VAR Retain Persistent
	The variable address is out of the retainable area.		
Online Change	●	●	●
Reboot PLC	○	●	●
Reset Warm	○	●	●
Reset Cold	○	○	●
Download	○	○	●
Reset Origin	○	○	○

● = Invalid

○ = Valid

4.4 Task

4.4.1 Task Configuration

You define one or more tasks for controlling and executing the program blocks (POUs) in the PLC. 0You define a task with a name, a priority, and a type, which determines which condition triggers the start of the task. You can define this condition either by time (cyclic-interval, freewheeling) or by the occurrence of an internal or external event to process the task.

A task calls one or more program blocks (POUs). With the combination of priority and condition, you define the order in which the tasks are processed. You can configure a watchdog for each task.

Rules for the processing order of the defined tasks:

- If the task condition is satisfied, then the system processes the task.
- If several tasks satisfy the condition for processing at the same time, then the system processes the tasks with the highest priority first.
- If several tasks with the same priority level satisfy the condition for processing at the same time, then the system processes the longest waiting task first.
- The program calls are processed in the order they appear in the configuration dialog of the task.
- If a called program has the same name in the device tree of the application and in a library or project-global in the POU window, then the application program is used.

Note: Set the priority level from 0 to 31. If the set number is closer to 0, it has higher priority.

4.4.1.1 Task Types

There are five types of task types:

- **Cyclic Task**：

  The system processes the task in cycles. The cycle time of the task is defined in the input field Interval.

- **Event Task**：

  The system starts processing the Event Task as soon as the global variable defined in the input field Event contains a rising edge.

- **Freewheeling Task**：

  The system starts processing the Freewheeling Task again automatically in a continuous loop at program start and at the end of a complete pass.

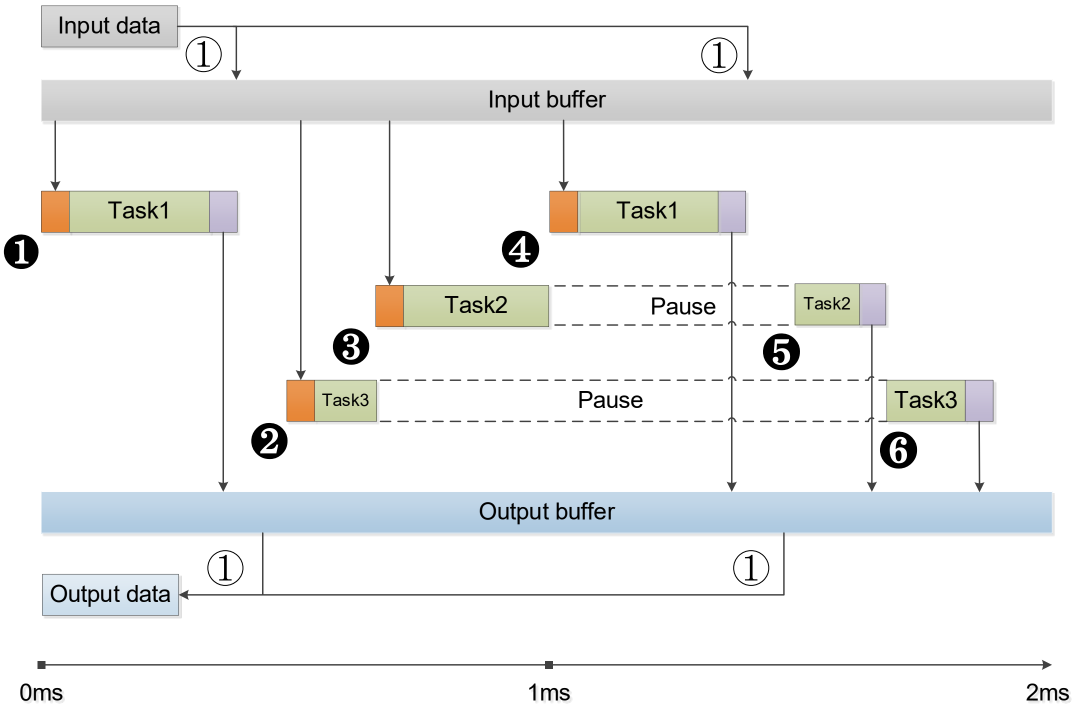
- **Status Task**：

  The system starts Status Task processing as soon as the variable defined in the Event input field yields the Boolean value TRUE.

4.4.1.2 Bus Cycle Task

If the task condition is satisfied, then the system processes the task.
Set the priority level from 0 to 31. If the set number is closer to 0, it has higher priority.
The system processes the task in the order of Task Group in Task Configuration.

Behavior of the bus cycle



① Bus cycle

Task 1: Priority = 1, Bus cycle Task, Cyclic Task

Task 2: Priority = 3, Event Task

Task 3: Priority = 5, Freewheeling Task

❶ The condition for starting Task 1 is met; Task 1 starts.

❷ Task 1 completes and the I/O data from buffer is exchanged with the I/O channel (physical hardware.) Task 3 starts.

❸ The condition for starting Task 2 is met and Task 2 has higher priority than Task 3 does. Thus Task 2 starts and Task 3 halts.

❹ The condition for starting Task 1 is met and Task 1 has higher priority than Task 2 does. Thus Task 3 starts and Task 4 halts.

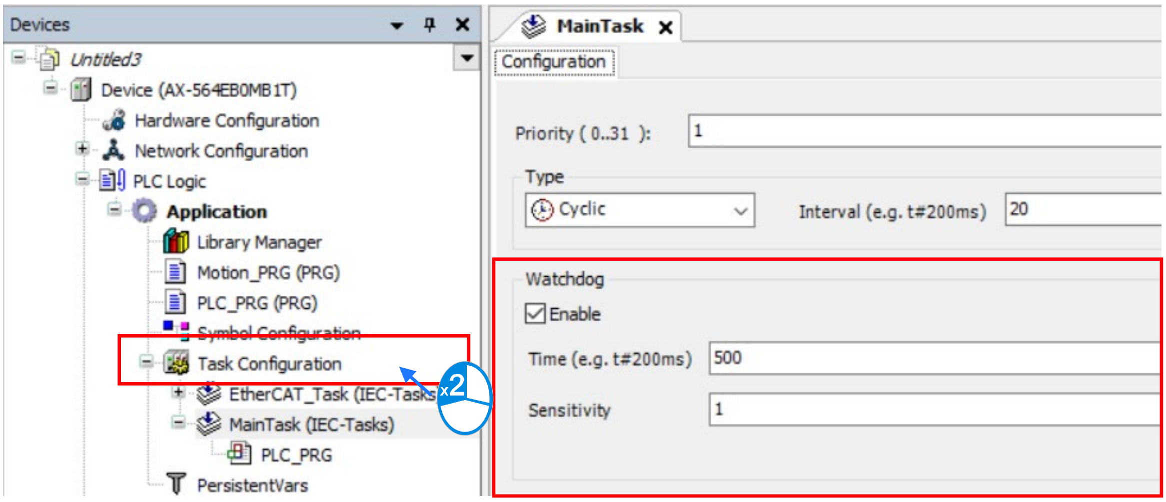
❺ Task 1 completes and the I/O data from buffer is exchanged with the I/O channel (physical hardware.) Task 2 starts again.

❻ Task 2 completes and the Task 3 starts again.

Note: The connected extension modules are refreshed according to the set refreshing cycle time to update input data (actual input) to the input buffer and update data from output buffer to output data (actual output).
Components: The components e.g., ethernet and EtheCAT are refreshed according to the set bus cycle task. After the bus cycle task is executed, update input data (actual input) to the input buffer and update data from output buffer to output data (actual output).

4.4.1.3 Watchdog

If the task exceeds the time set for the watchdog, then the task is halted with an error status.



Priority (0..31): 1

Type: Cyclic

Interval (e.g. t#200ms): 20

Watchdog

☑ Enable

Time (e.g. t#200ms): 500

Sensitivity: 1

- Several consecutive timeouts:

  Sensitivity: 0, watchdog timeout = time *1

  Sensitivity: n, watchdog timeout = time *n

4.4.1.4 Motion Instructions for Types of Tasks

Here is the table of motion instructions for different task types. “V” means the motion instruction can be executed for the task type.

- **Synchronization axes**

Classification	Instruction Name	Task Type		
		Cyclic	Freewheeling	Bus Cycle EtherCAT
Motion Control Function Blocks	MC_Home			V
	MC_Stop			V
	MC_Halt			V
	MC_MoveAbsolute			V
	MC_MoveRelative			V
	MC_MoveAdditive			V
	MC_MoveSuperImposed			V
	MC_CamIn			V
	MC_CamOut			V
	MC_MoveVelocity			V
	MC_PositionProfile			V
	MC_VelocityProfile			V
	MC_AccelerationProfile			V
	MC_Jog			V
	MC_GearIn			V
	MC_GearOut			V
	MC_GearInPos			V
	MC_Phasing			V
	DMC_TorqueControl			V
	DMC_VelocityControl			V
	DMC_MoveLinearAbsolute			V
	DMC_MoveLinearRelative			V
	DMC_MoveCircularAbsolute			V
	DMC_MoveCircularRelative			V
	DMC_GroupStop			V
	DMC_GroupHalt			V
	DMC_Home_P			V
	DMC_GroupInterrupt			V
	DMC_GroupContinue			V
	DMC_ImmediateStop_P			V
Instructions for Management	MC_Power	V	V	V
	MC_SetPosition	V	V	V
	MC_ReadParameter	V	V	V
	MC_WriteParameter	V	V	V
	MC_ReadBoolParameter	V	V	V
	MC_WriteBoolParameter	V	V	V
	MC_ReadActualPosition	V	V	V
	MC_ReadActualVelocity	V	V	V
	MC_ReadActualTorque	V	V	V

Classification	Instruction Name	Task Type		
		Cyclic	Freewheeling	Bus Cycle EtherCAT
	MC_Reset	V	V	V
	MC_ReadStatus	V	V	V
	MC_ReadAxisError	V	V	V
	MC_CamTableSelect	V	V	V
	MC_TouchProbe	V	V	V
	MC_AbortTrigger	V	V	V
	MC_DigitalCamSwitch	V	V	V
	DMC_GroupEnable	V	V	V
	DMC_GroupDisable	V	V	V
	DMC_GroupReadStatus	V	V	V
	DMC_GroupReadError	V	V	V
	DMC_GroupReset	V	V	V
	DMC_CamReadTappetStatus	V	V	V
	DMC_CamReadTappetValue	V	V	V
	DMC_CamWriteTappetValue	V	V	V
	DMC_CamAddTappet	V	V	V
	DMC_CamDeleteTappet	V	V	V
	DMC_CamReadPoint	V	V	V
	DMC_CamWritePoint	V	V	V
	DMC_ChangeMechanismGearRation	V	V	V
	DMC_ReadMotionState	V	V	V
	DMC_GroupReadParameter	V	V	V
	DMC_GroupWriteParameter	V	V	V

Note: it is suggested a motion function block should be created within a bus cycle EtherCAT to avoid inconsistent movement.

- **Positioning axes**

Classification	Instruction Name	Task Type		
		Cyclic	Freewheeling	Bus Cycle EtherCAT
Motion Control Function Blocks	MC_Halt_DML	V	V	V
	MC_Home_DML	V	V	V
	MC_MoveAbsolute_DML	V	V	V
	MC_MoveRelative_DML	V	V	V
	MC_MoveVelocity_DML	V	V	V
	MC_Stop_DML	V	V	V
Instructions for Management	MC_Power_DML	V	V	V
	MC_ReadBoolParameter_DML	V	V	V
	MC_ReadParameter_DML	V	V	V
	MC_ReadStatus_DML	V	V	V
	MC_Reset_DML	V	V	V
	MC_WriteBoolParameter_DML	V	V	V
	MC_WriteBoolParameter_DML	V	V	V
	MC_ChangeAxisConfig_DML	V	V	V
	MC_ReinitDrive_DML	V	V	V
	MC_SetOpmode_DML	V	V	V
	MC_StartupDrive_DML	V	V	V

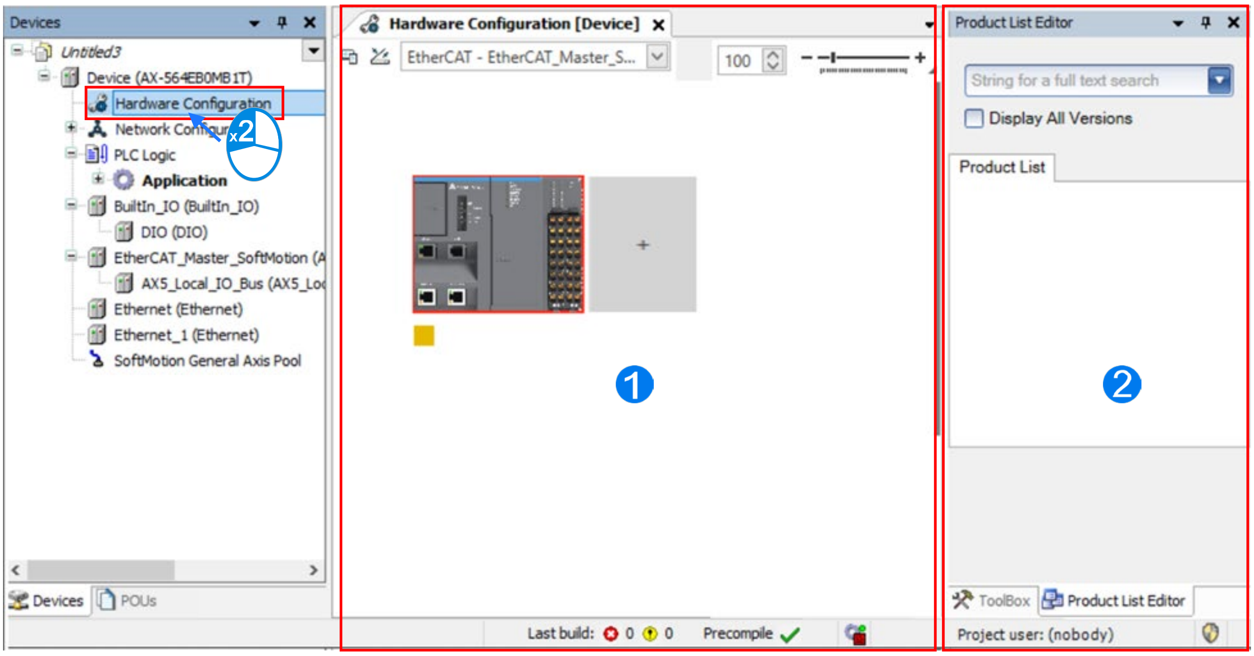
Chapter 5 Hardware Configuration

Table of Contents

Hardware Configuration is the tools in DIADesign-AX for hardware configuration. Its functions include setting parameters for CPU and modules. This chapter will introduce the abovementioned functions.

5.1 Environment of Hardware Configuration

Double-click Hardware Configuration on the Device section to open the Hardware Configuration (Device) window as the image shown below.

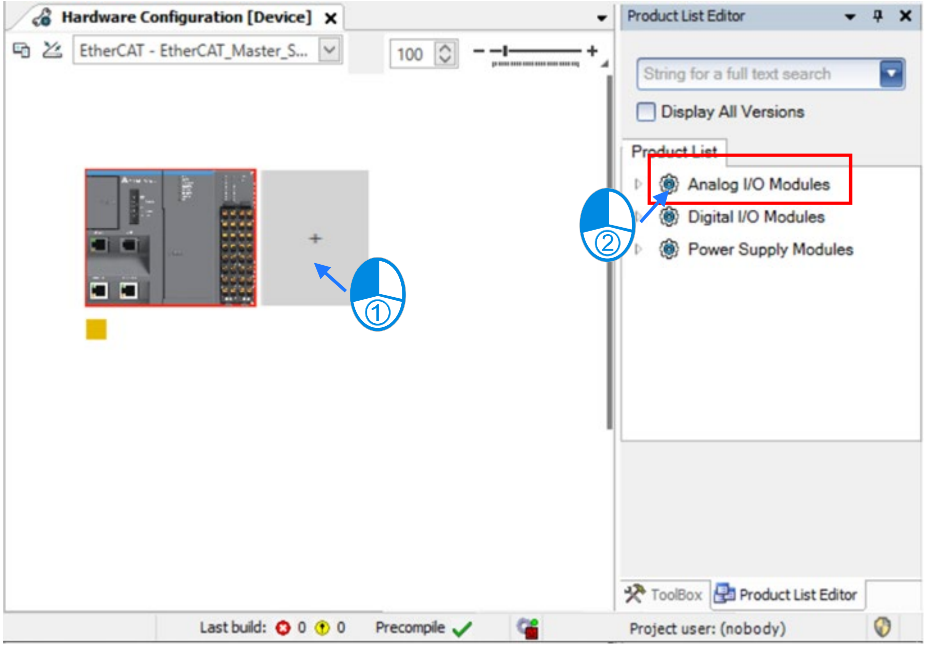


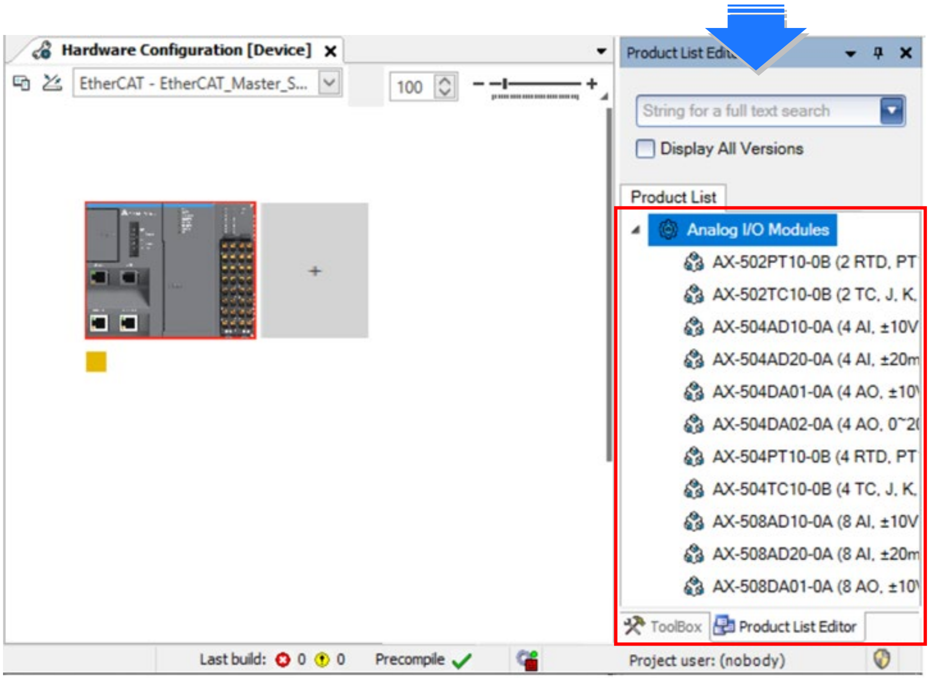
❶ Hardware Configuration (Device): This is the main work area for system configuration and settings.

❷ Product List Editor: Here listed out all supported modules for the selected CPU.

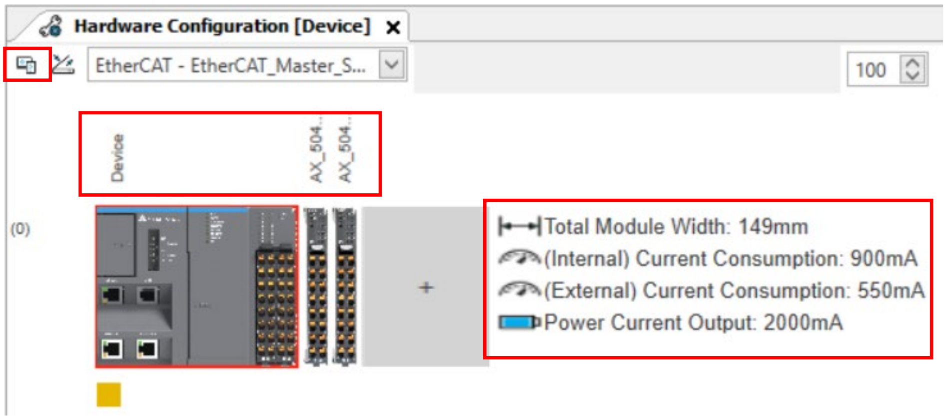
Click + to see all the supported modules on the right window (Product List Editor).

Click ▷ to unfold the list. Click the module name to see a short module description.



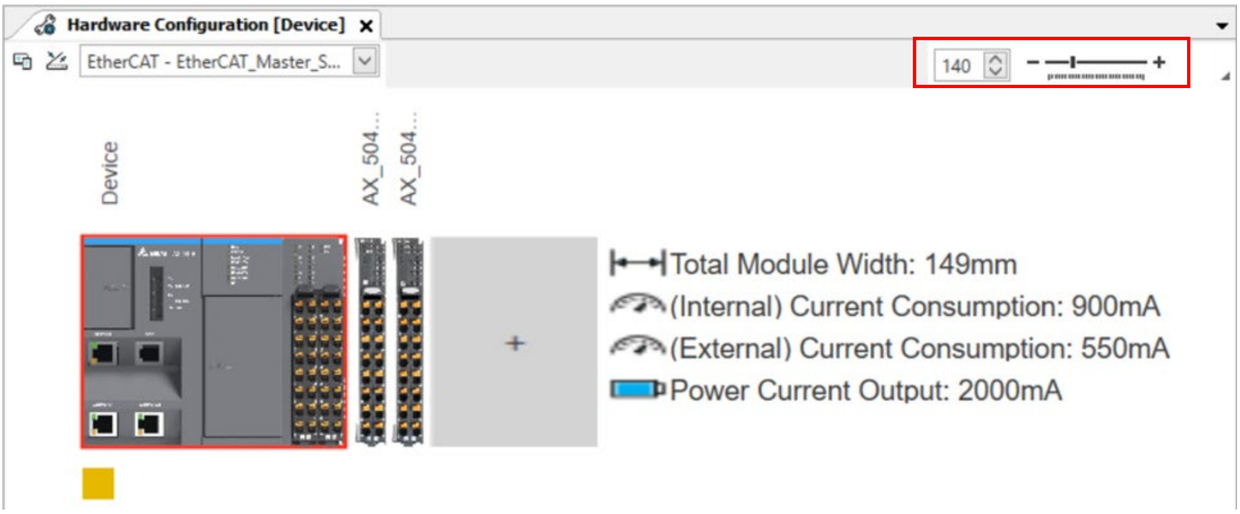


Click on the upper-left corner to see the current configurations. For example, the width of the total connected module, the current consumption and power current output.



Total Module Width: 149mm
(Internal) Current Consumption: 900mA
(External) Current Consumption: 550mA
Power Current Output: 2000mA

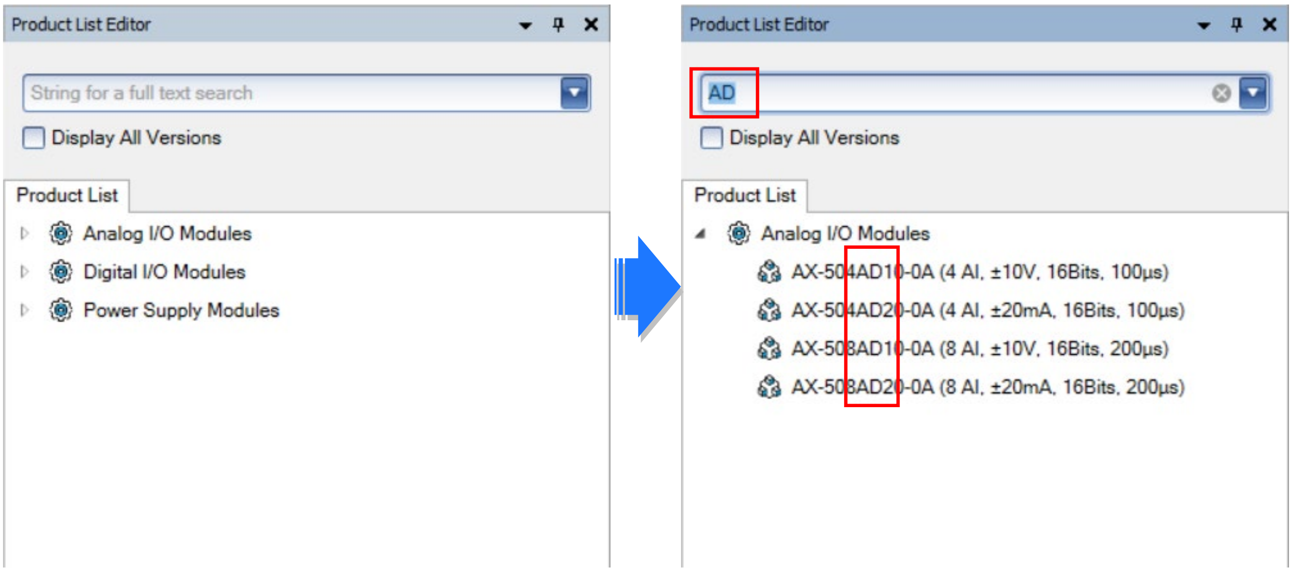
Use  on the upper-right corner to rearrange the device image for better viewing experience and easier operation.



Total Module Width: 149mm
(Internal) Current Consumption: 900mA
(External) Current Consumption: 550mA
Power Current Output: 2000mA

Total Module Width: 149mm
(Internal) Current Consumption: 900mA
(External) Current Consumption: 550mA
Power Current Output: 2000mA

Enter a key word in the **Search Toolbox** on the right-side window and press “Enter” button on your keyboard to search for the matched modules.

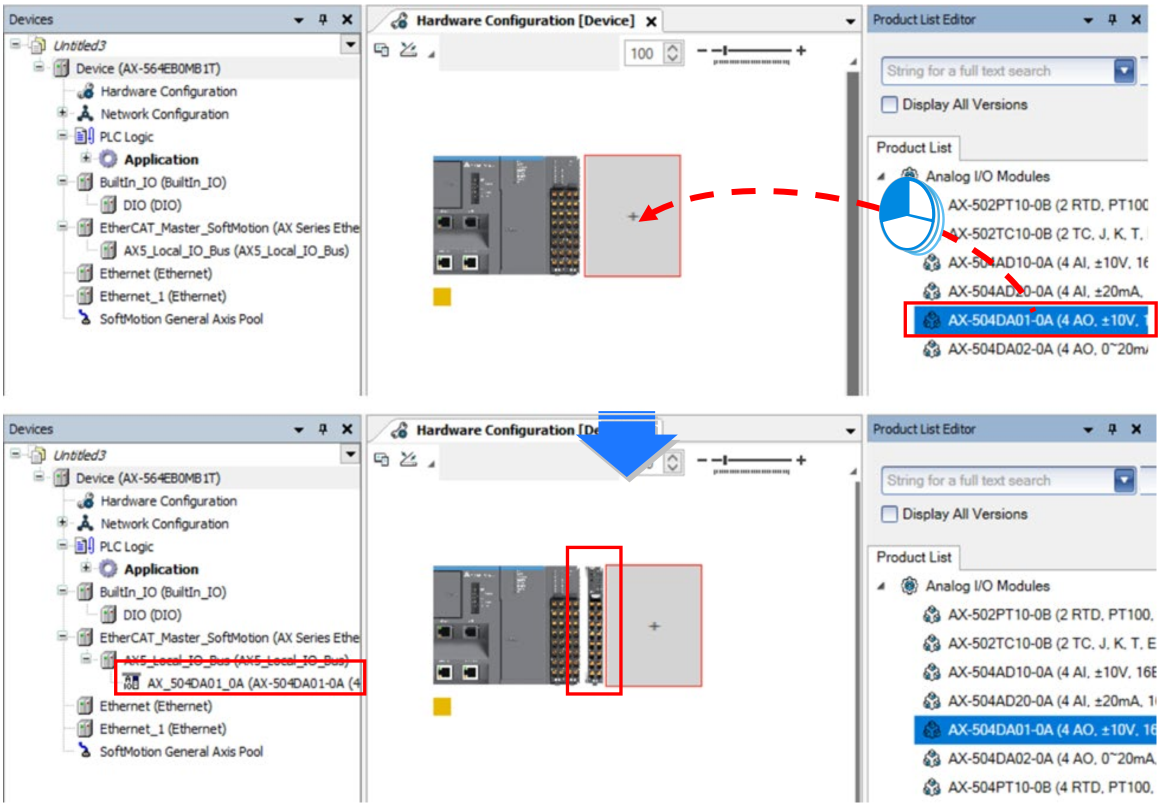


AX-504AD10-0A (4 AI, ±10V, 16Bits, 100µs)
AX-504AD20-0A (4 AI, ±20mA, 16Bits, 100µs)
AX-508AD10-0A (8 AI, ±10V, 16Bits, 200µs)
AX-508AD20-0A (8 AI, ±20mA, 16Bits, 200µs)

5.2 Adding a Module

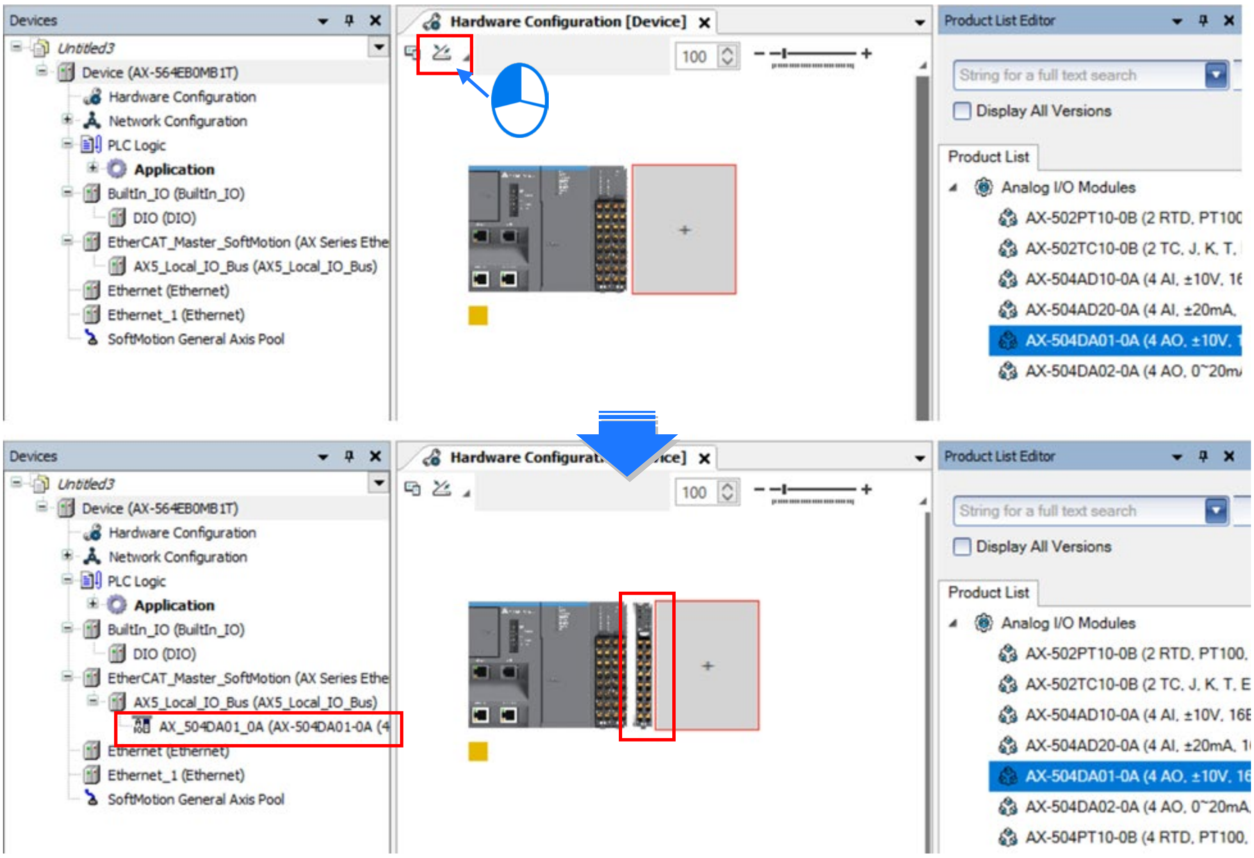
- **Method 1**

With AX-5 Series PLC backplaneless design, the extension module can install on the right-side of AX-5 Series PLC directly. Double-click or drag and drop the extension module that you'd like to add from the Product List. Newly added extension modules will appear on the right-side of the AX-5 Series PLC. And the device names will also show up on the left-side under AX5_Local_IO_BUS.



- **Method 2**

If the AX-5 Series PLC and its connected extension module are powered on and the gateway is correctly set, you can use the icon to scan and add the modules in. Newly added extension modules will appear on the right-side of the AX-5 Series PLC. And the device names will also show up on the left-side under AX5_Local_IO_BUS.

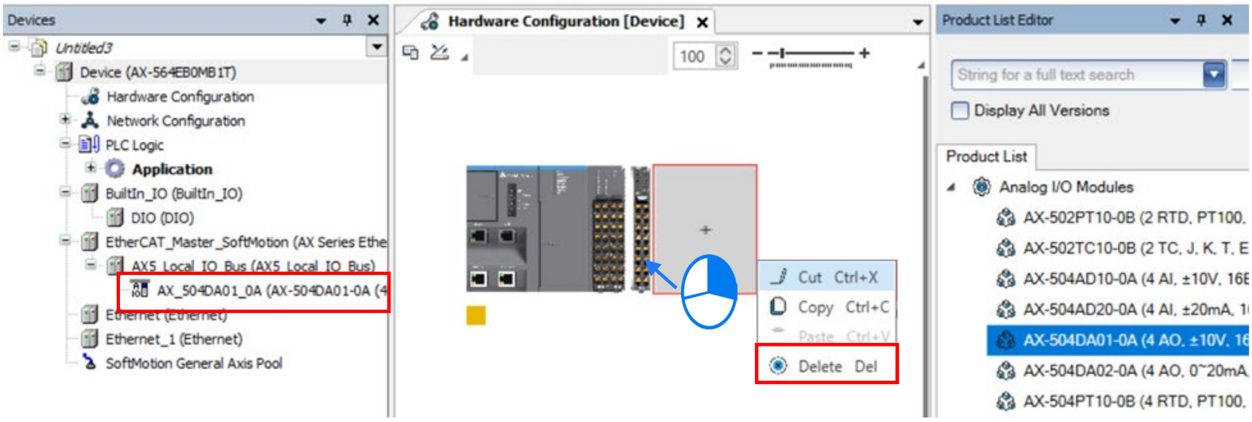


5.3 Removing a Module

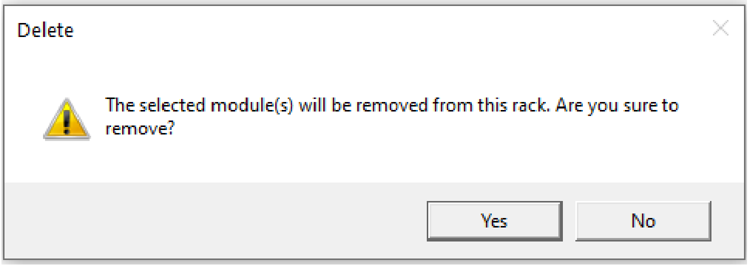
You cannot remove a CPU. You can only delete extension modules.

- **Method 1**

Right-click the module image that you’d like to remove to open the context menu and click the option **Delete** or use the Delete Button on your keyboard to remove the module.

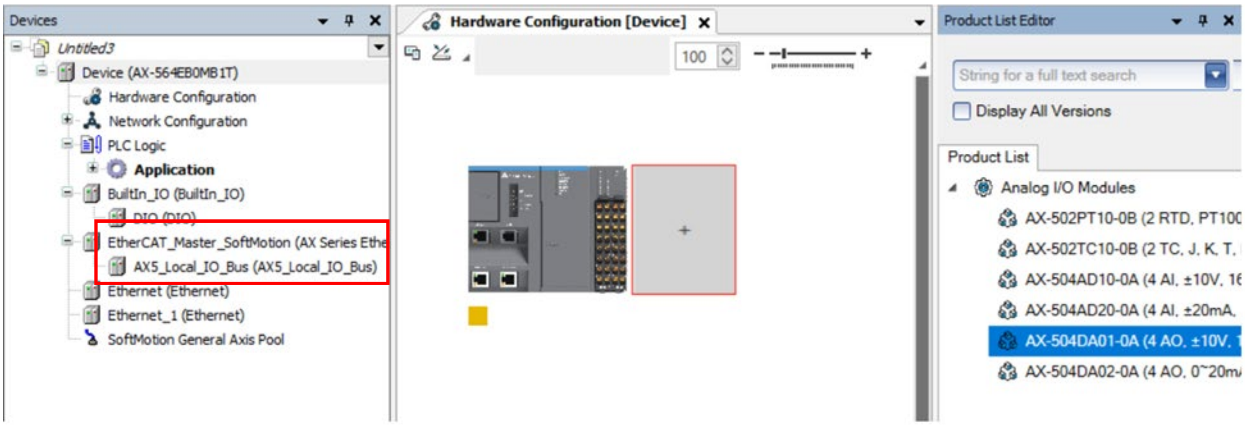


After you click **Delete**, a confirmation shows up. Click **Yes** to delete the module.


Delete

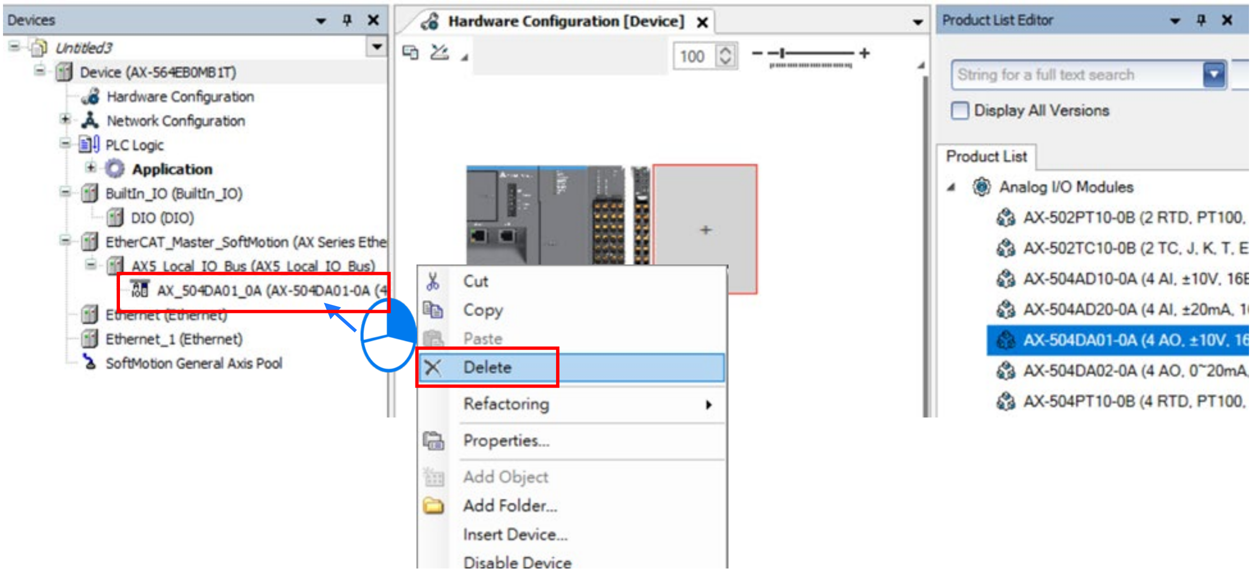
The selected module(s) will be removed from this rack. Are you sure to remove?

And the device names will also be removed from the left-side under AX5_Local_IO_BUS.



- **Method 2**

Right-click the device name under AX5_Local_IO_BUS that you'd like to remove to open the context menu and click the option **Delete** or use the Delete Button on your keyboard to remove the module. After that the device image will also be removed from the editing area.



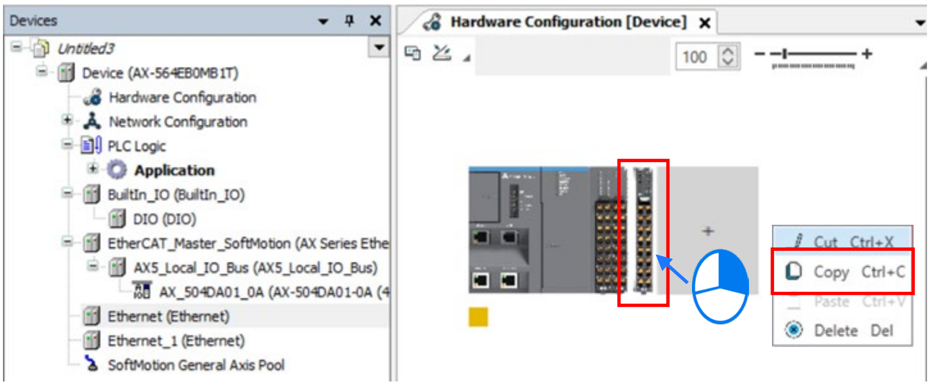
5.4 Using Copy and Paste to Add a Module

You cannot use copy and paste on a CPU. You can only use copy and paste on extension modules.

5.4.1 Using Copy to Add a Module

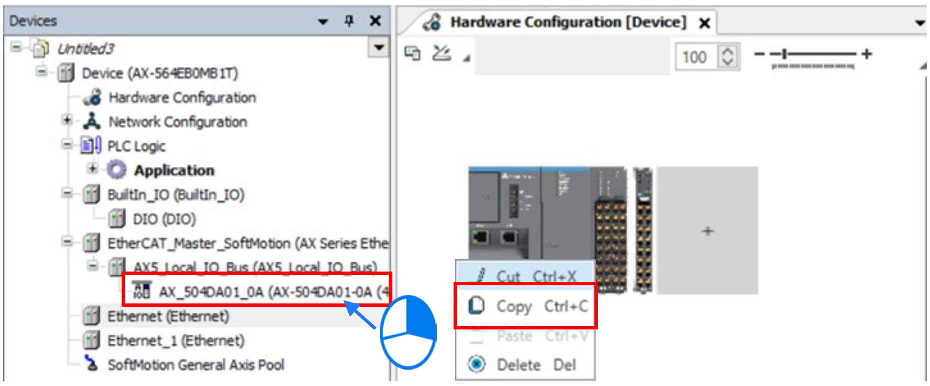
- **Method 1**

Right-click the module image that you'd like to copy to open the context menu and click the option **Copy** to duplicate the module.



- **Method 2**

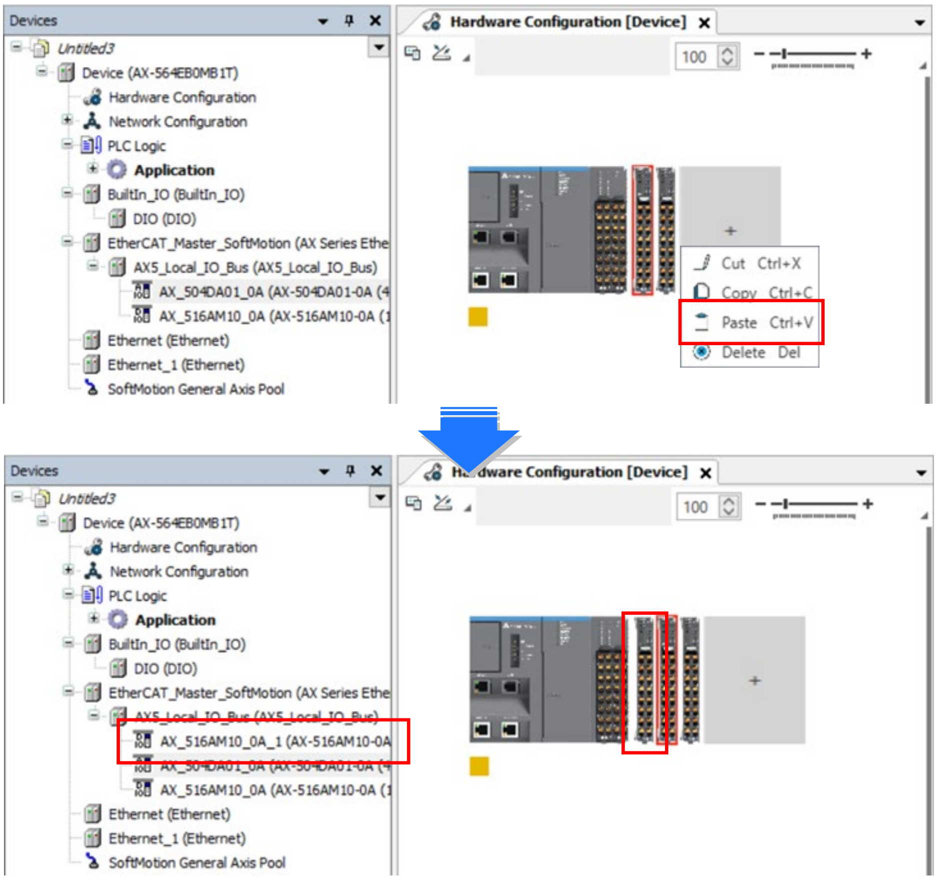
Right-click the device name under AX5_Local_IO_BUS that you'd like to copy to open the context menu and click the option **Copy** to copy the module.



5.4.2 Using Paste to Add a Module

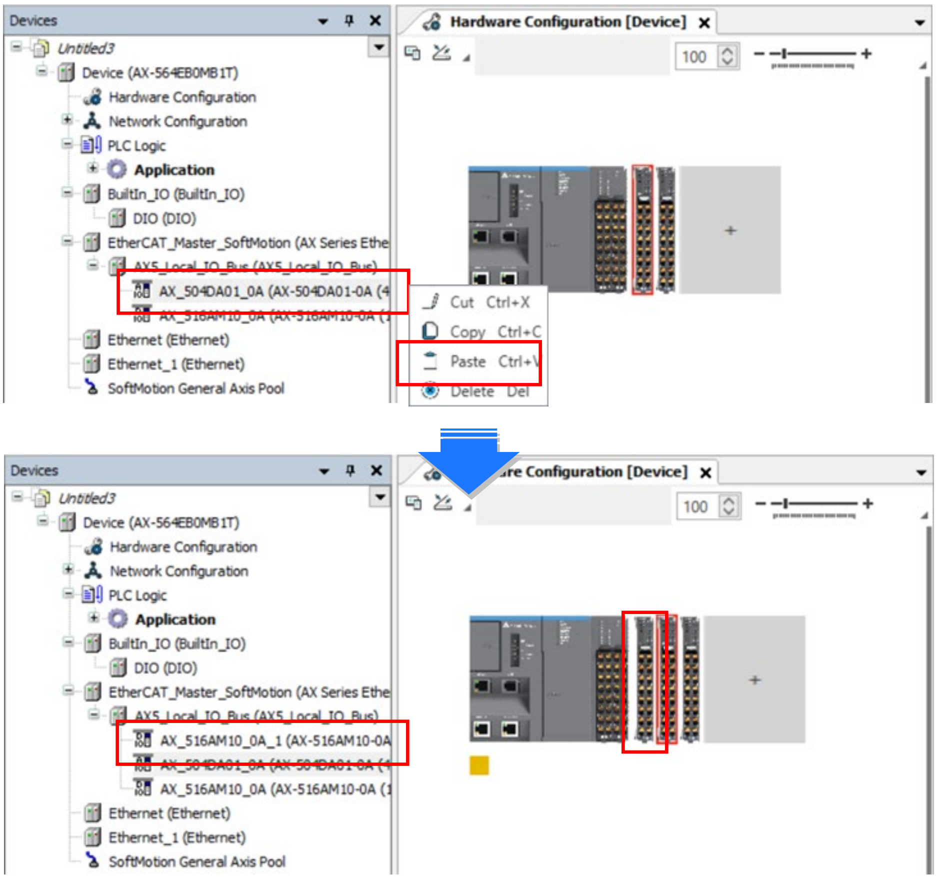
- **Method 1**

You can place the module between modules. Right-click where you'd like to paste the module to open the context menu and click the option **Paste** to place the module on the left of the module you had clicked. Or you can place the module at the end by right clicking the + to paste the copied module there. And the device names will also be updated on the left-side under AX5_Local_IO_BUS.



- **Method 2**

You can place the module between modules. Right-click where you'd like to paste the module under AX5_Local_IO_BUS to open the context menu and click the option **Paste** to place the module above the module you had clicked. Or you can place the module at the end by right-clicking AX5_Local_IO_BUS to paste the copied module. And the module image will also be updated on the editing area.



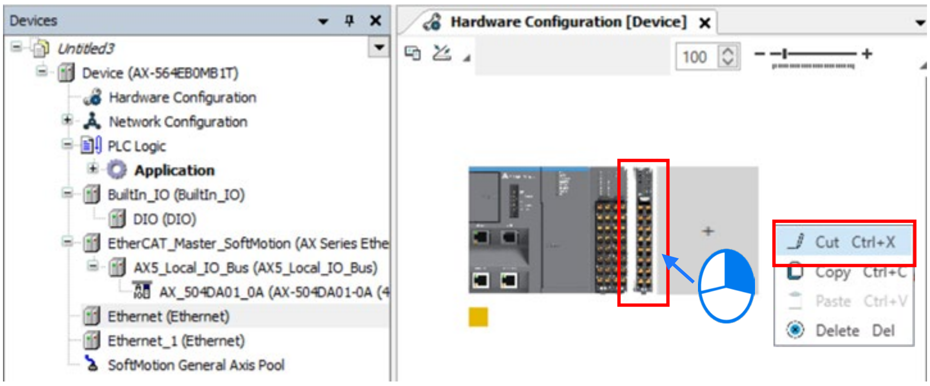
5.5 Using Cut and Paste to Remove or Add a Module

You cannot use cut and paste on a CPU. You can only use cut and paste on extension modules.

5.5.1 Using Cut to Remove a Module

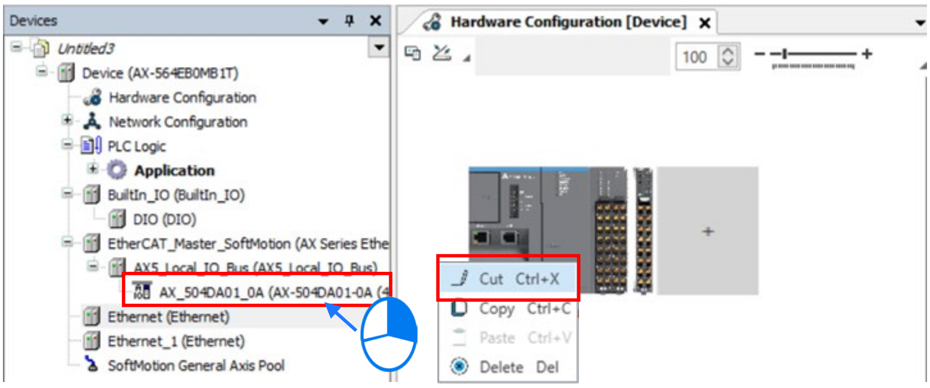
- **Method 1**

Right-click the module image that you'd like to cut to open the context menu and click the option **Cut** to take out the module.



- **Method 2**

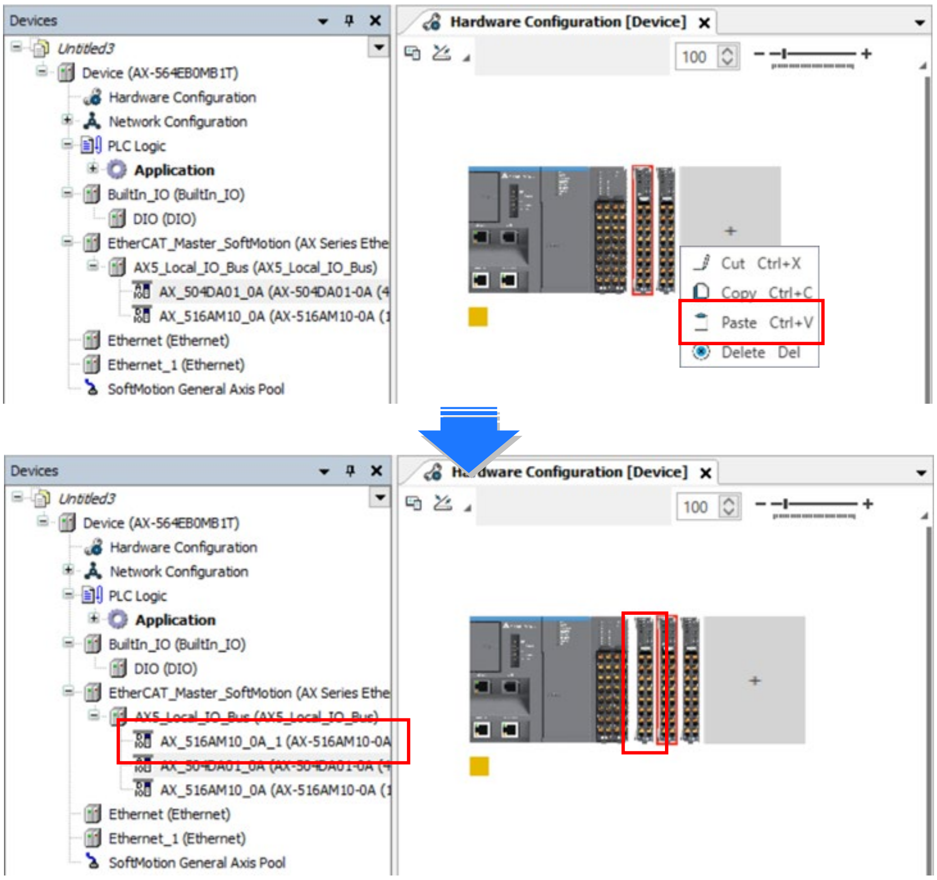
Right-click the device name under AX5_Local_IO_BUS that you'd like to cut to open the context menu and click the option **Cut** to take out the module.



5.5.2 Using Paste to Add a Module

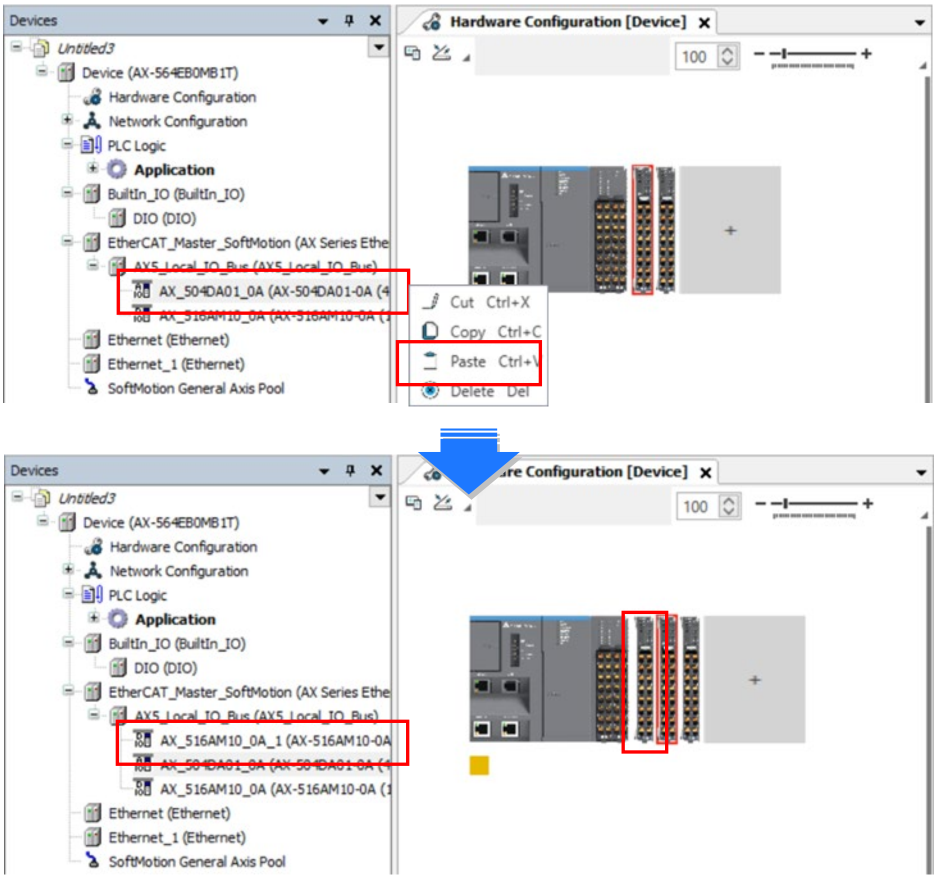
- **Method 1**

You can place the module between modules. Right-click where you'd like to paste the module to open the context menu and click the option **Paste** to place the module on the left of the module you had clicked. Or you can place the module at the end by right clicking the + to paste the copied module there. And the device names will also be updated on the left-side under AX5_Local_IO_BUS.



- **Method 2**

You can place the module between modules. Right-click where you'd like to paste the module under AX5_Local_IO_BUS to open the context menu and click the option **Paste** to place the module above the module you had clicked. Or you can place the module at the end by right-clicking AX5_Local_IO_BUS to paste the copied module. And the module image will also be updated on the editing area.



Chapter 6 Network Configuration

Table of Contents

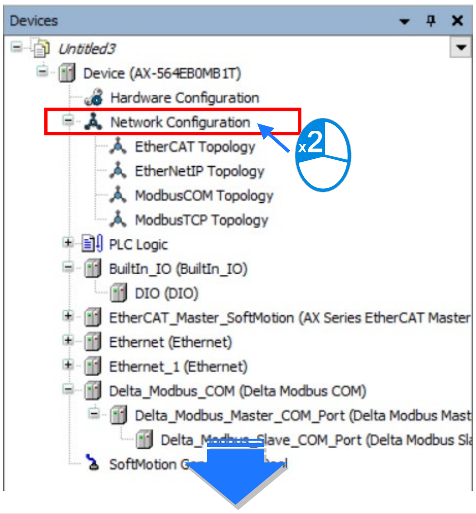
6.1 Network Configuration

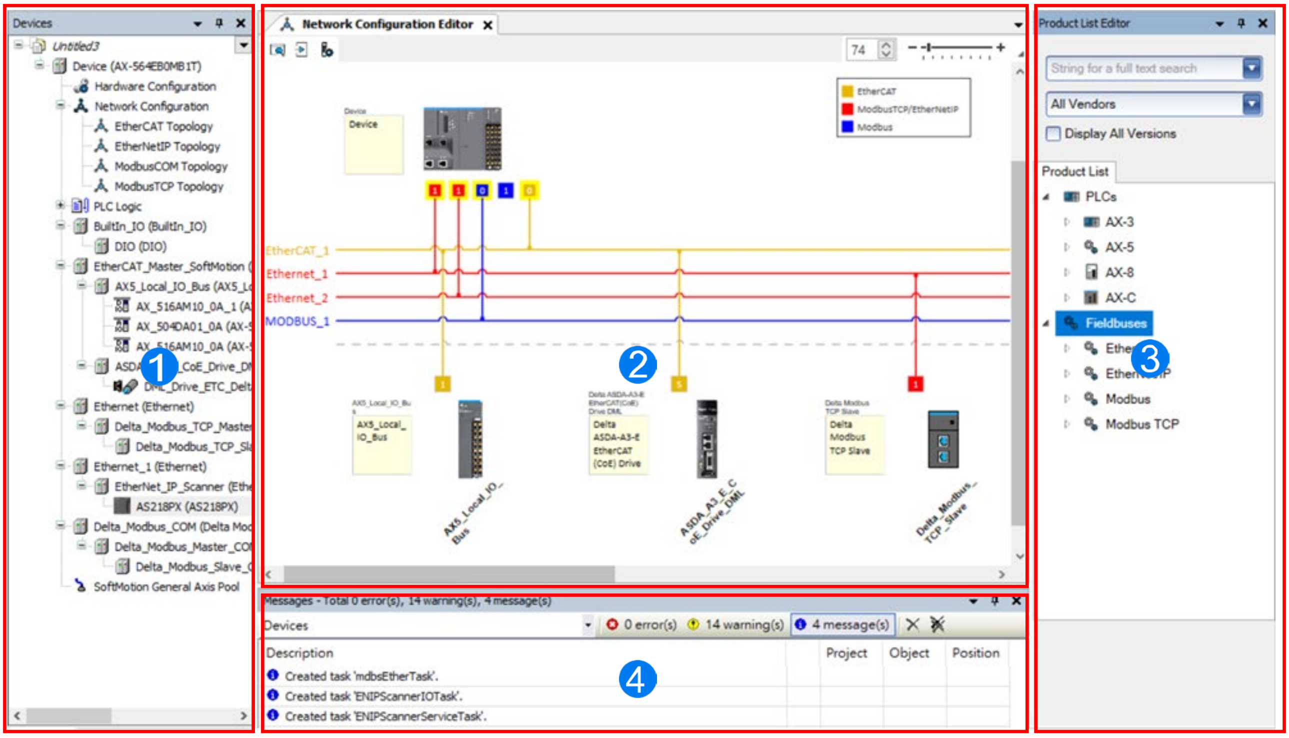
DIADesigner-AX provides a Network Configuration tool for users to configure the network in a project. Detailed network setting information will be covered in the following sections.

6.1.1 Introduction

You can use Network Configuration to create networks such as EtherCAT, ModbusTCP, EtherNet/IP, and Modbus in a project and set up file sending paths and (b) set up EtherCAT Master, Modbus COM port, and Ethernet IP settings

Network Configuration is under the Device tree. You can double-click Network Configuration to open its setting page and start planning a network framework for the project.





Messages - Total 0 error(s), 14 warning(s), 4 message(s)

Created task 'mdbsEtherTask'.
Created task 'ENIPScannerIOTask'.
Created task 'ENIPScannerServiceTask'.

❶ **Device:** Here shows all the configured devices in a tree view.

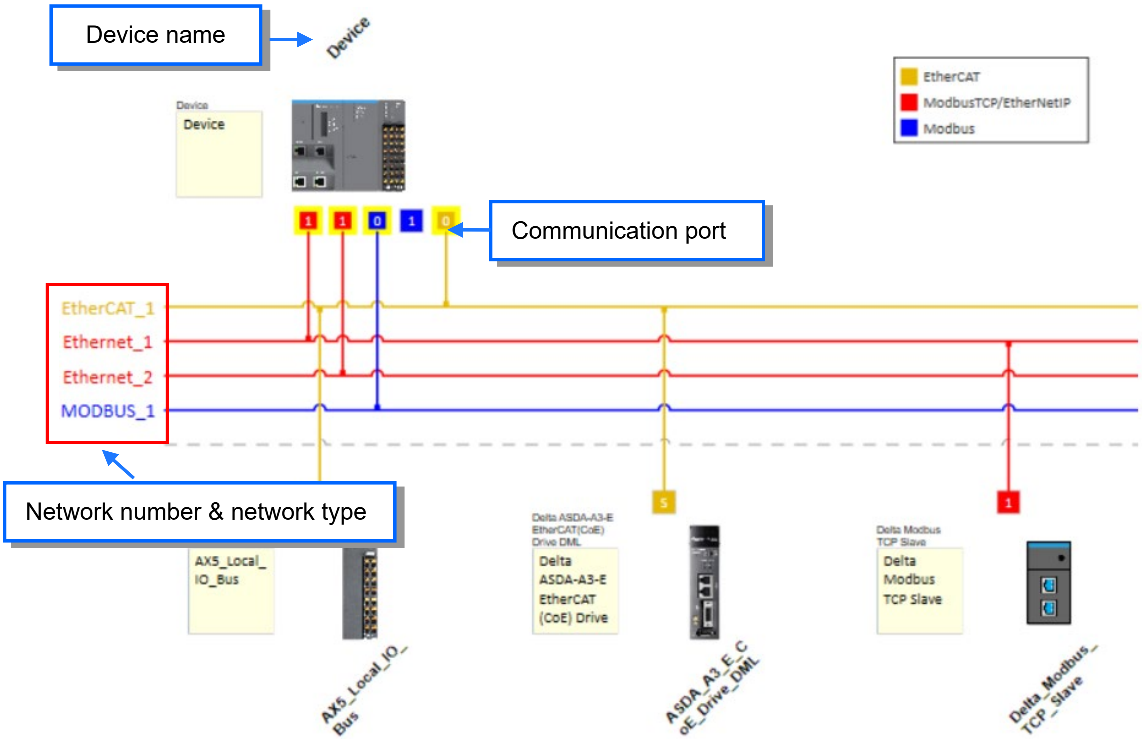
❷ **Working area:** Here is the main working area for you to create a network framework.

❸ **Device list**: Here lists all the available devices in a tree view.

❹ **Message display area:** Here displays operational messages.

6.1.2 Basic Knowledge

Before creating networks, you need to have some basic knowledge. Here we provide some basic knowledge in the following sections for you.



- **Device and Network**

A device is the most basic element in a network. It can be a PLC, a servo, a drive or any device that you defined. Here a network is a collection of devices which are interconnected. Every communication port should be assigned with a network type, such as Modbus, Ethernet or EtherCAT. A physical interface that a device uses to connect to a network is a communication port of the device. If there are more than two ports on a device, the device can connect to different networks.

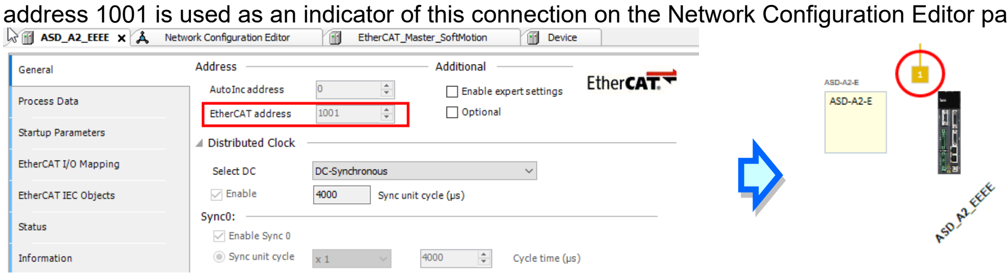
- **Device Name**

A device name is the identity of the device. You can identify a device in the Device Tree by its name. However, it bears little significance on operation.

- **Network Type and Communication Port**

■ **EtherCAT**

The orange yellow line indicates the EtherCAT communication. Double-click the Master station node to open the EtherCAT setting page of the Master. The number of Master Station is 0 and that cannot be changed. Double-click the connection of Slave to open the EtherCAT setting page of the Slave. The last digit appeared in the EtherCAT address 1001 is used as an indicator of this connection on the Network Configuration Editor page.



Address

AutoInc address: 0

EtherCAT address: 1001

Additional

☐ Enable expert settings

☐ Optional

Distributed Clock

Select DC: DC-Synchronous

☑ Enable 4000 Sync unit cycle (µs)

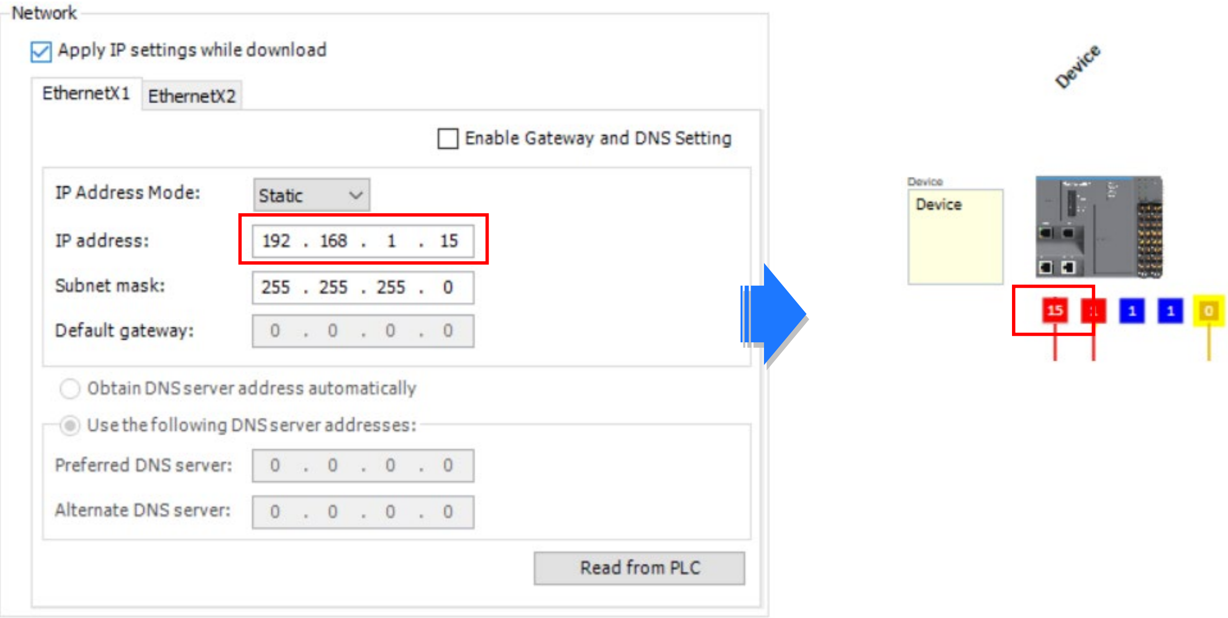
Sync0:

☑ Enable Sync 0

● Sync unit cycle x 1 4000 Cycle time (µs)

■ Modbus TCP or EtherNETIP

The blue line indicates the Modbus TCP or EtherNetIP communication. Double-click this line to open its setting page to edit IP addresses. The last digit appeared in the last section of the IP address is used as an indicator of this connection on the Network Configuration Editor page.



Network

☑ Apply IP settings while download

EthernetX1 | EthernetX2

☐ Enable Gateway and DNS Setting

IP Address Mode: Static

IP address: 192 . 168 . 1 . 15

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 0 . 0 . 0 . 0

○ Obtain DNS server address automatically

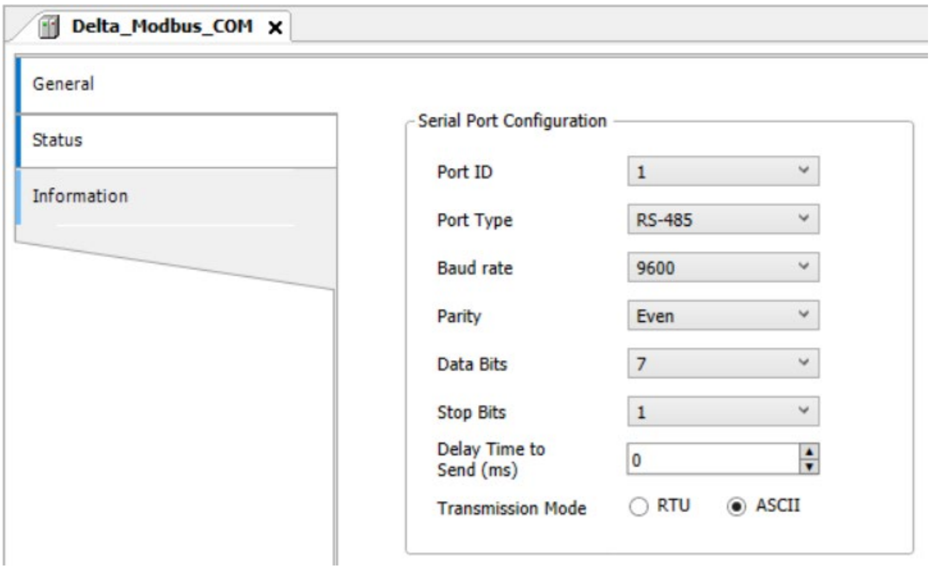
● Use the following DNS server addresses:

Preferred DNS server: 0 . 0 . 0 . 0

Alternate DNS server: 0 . 0 . 0 . 0

■ Modbus

The blue line indicates the Modbus communication (RS-232 / RS-485). Double-click this line to open the Modbus communication port setting page.



Serial Port Configuration

Port ID: 1

Port Type: RS-485

Baud rate: 9600

Parity: Even

Data Bits: 7

Stop Bits: 1

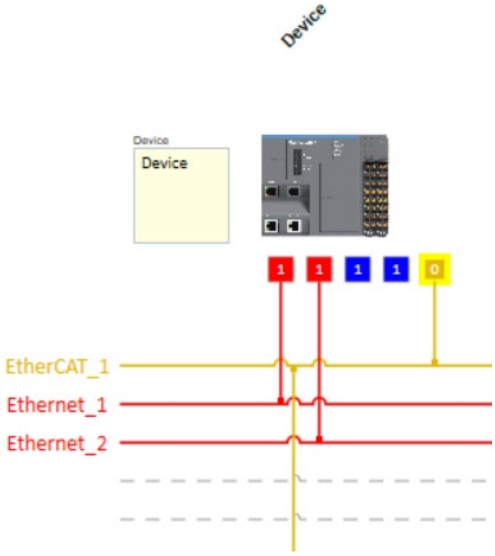
Delay Time to Send (ms): 0

Transmission Mode: ○ RTU ● ASCII

6.1.3 Creating a Network Topology

6.1.3.1 Station Nodes

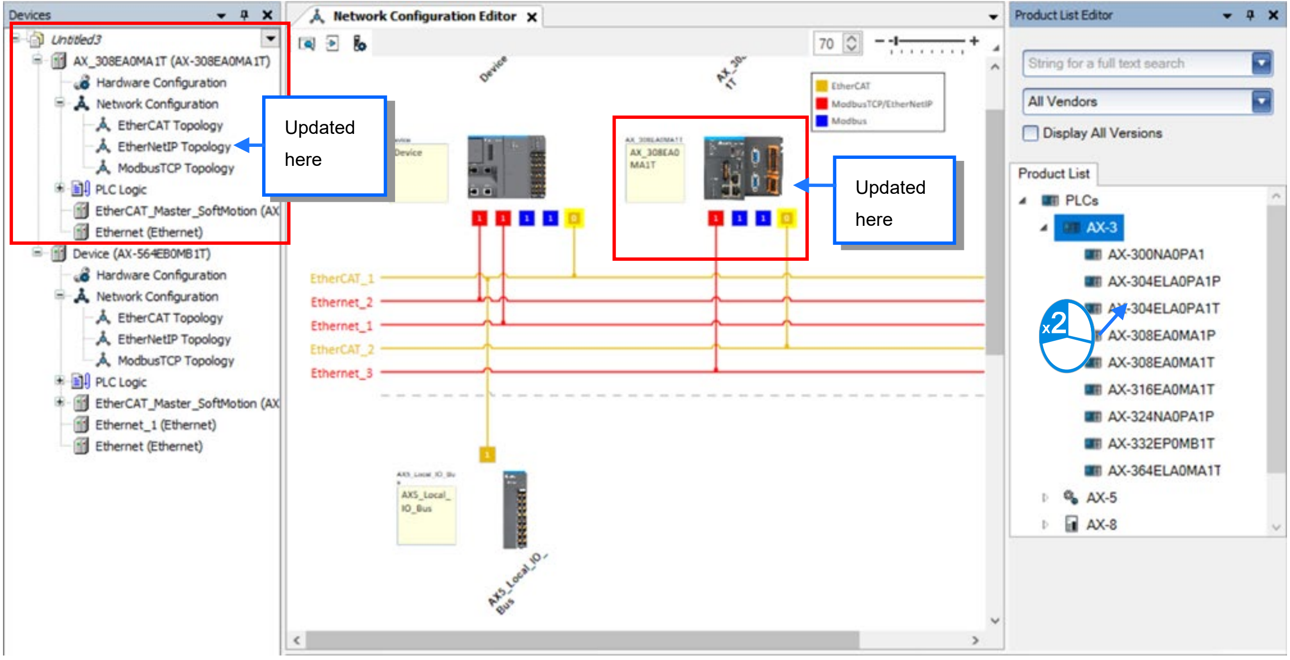
When you open the Network Configuration for the first time, the system creates a graphical representation automatically.



You can use the following methods to add devices including PLCs, servo motors, and drives in the network topology.

- **Method 1**

  Double-click the device that you want to add from the **Product List** on the right. After that you can see the added device is updated in the graphical representation and also on the Device Tree.

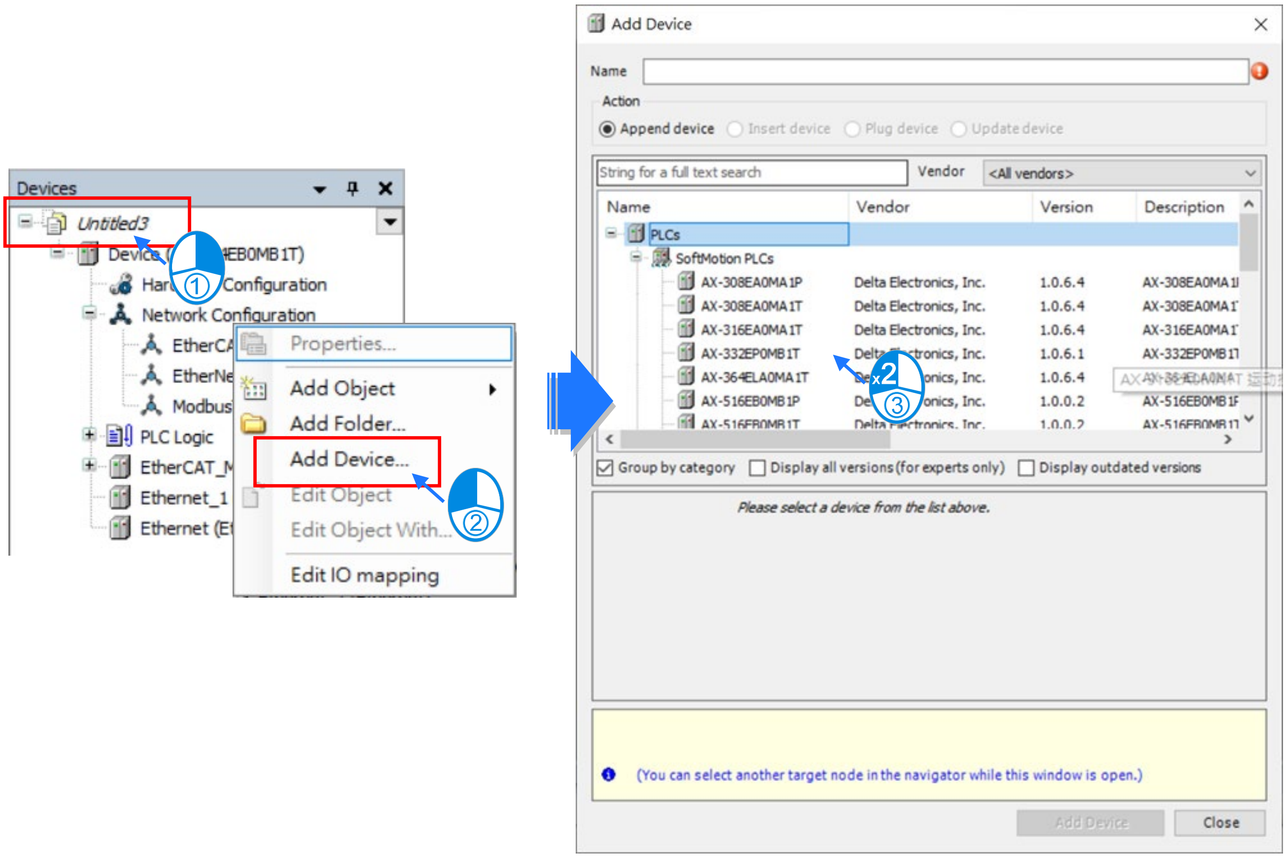


Updated here

Updated here

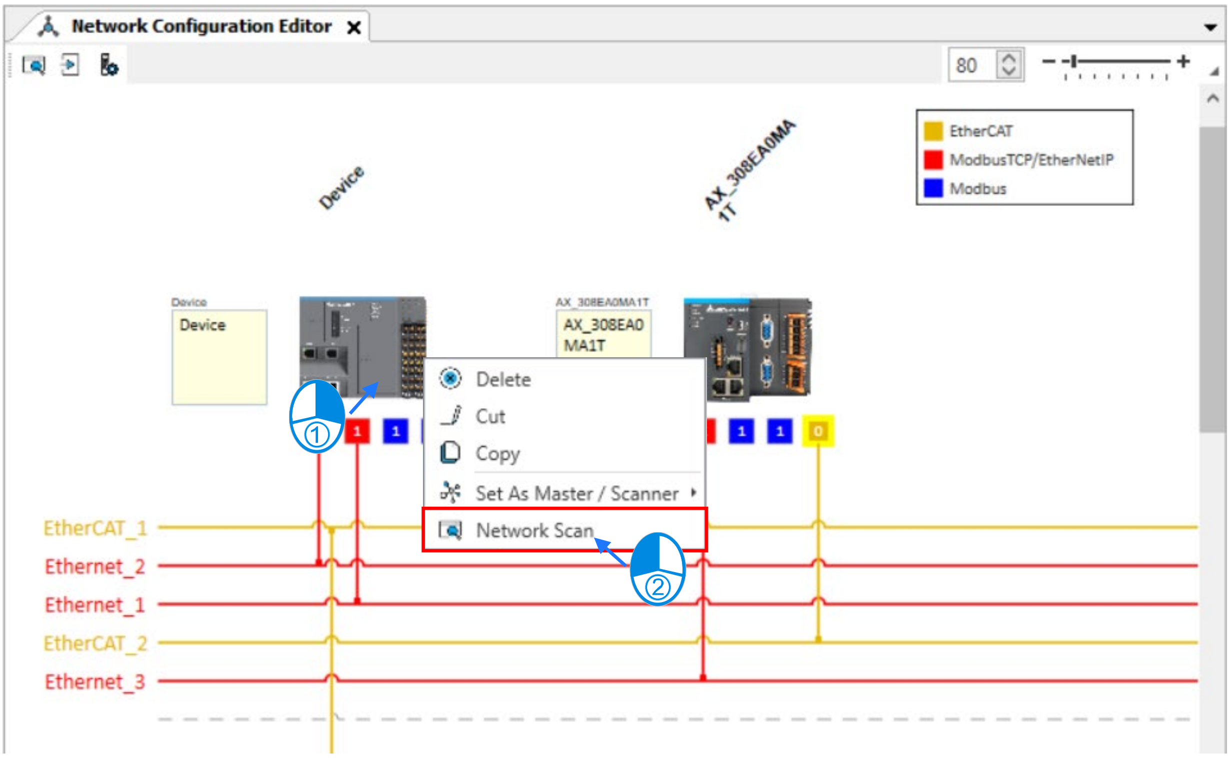
- **Method 2**

  Right-click the project name on the Device Tree to bring out the context menu. Double-click **Add Device** on the context menu to open a setting page for adding devices. Double-click the device you'd like to add or click **Add Device** to add the device in.



- **Method 3**

Right-click the device to bring out the context menu and click **Network Scan** or click the icon for Network Scan to automatically scan and then add the connected configured devices and network in the project.

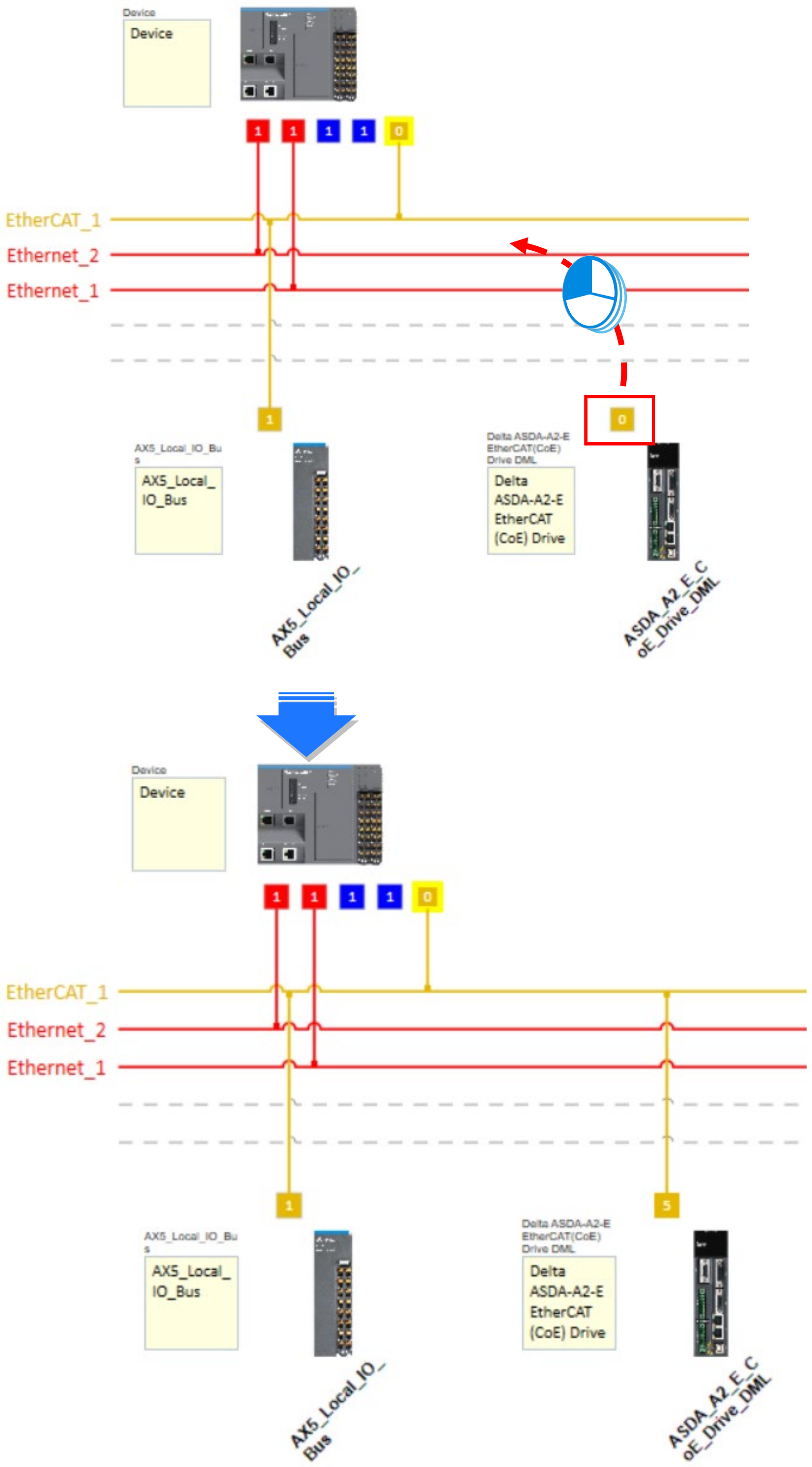


6.1.3.2 Creating a Connection

After creating the station nodes, you can start to crate connections. The network types include Modbus, Ethernet, and EtherCAT. Refer to 6.1.2 for more information. You can use the following methods to add created network connections.

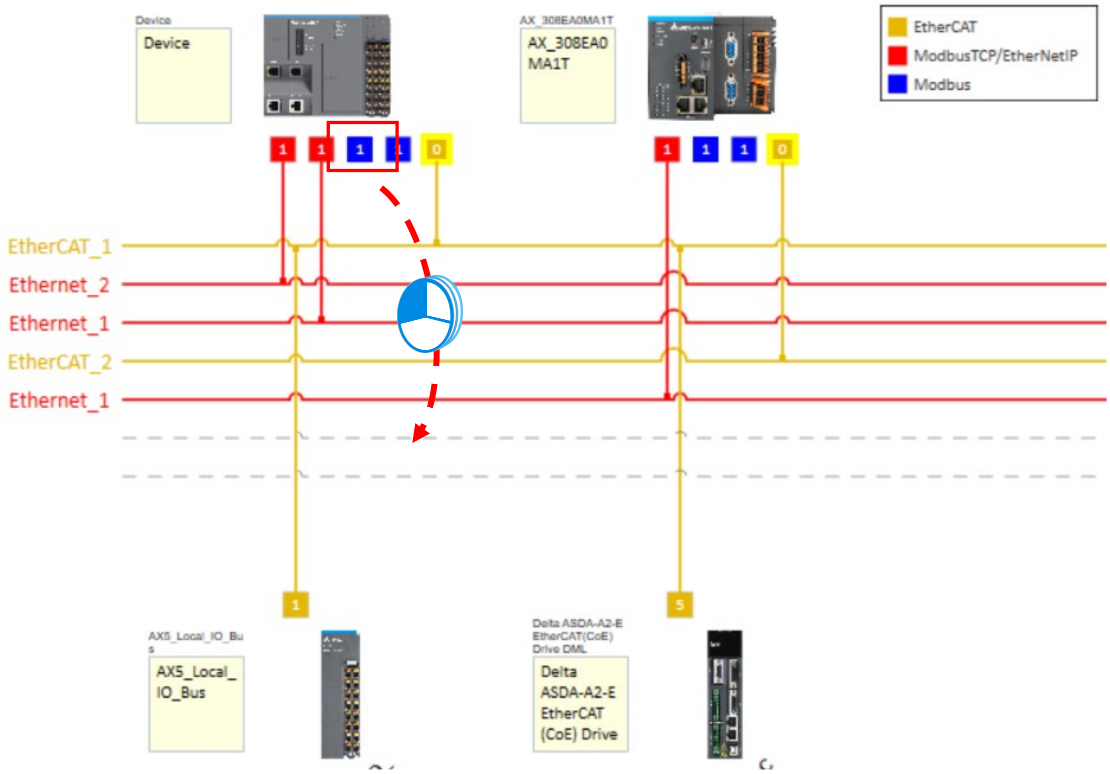
- **Method 1**

Drag and drop the communication port to the corresponding network type shown in line to create a connection between devices.



- **Method 2**

Hold the communication port and drag it to the unused dotted line to create a network connection that is the same as the selected network communication type and then a new gray unused dotted line will also be created.



MEMO

Chapter 7 Motion Control Setup & Operation

Table of Contents

7.1 Introduction on Motion Control Instructions

7.1.1 Motion Control Instructions

This manual introduces the elements for motion control programming including devices, symbols and motion control instructions.

Motion control instructions are defined as function blocks (FB) and are used in the program for performing a variety of motion control purposes. The motion control (MC) instructions are developed based on the specifications of PLCopen* motion control function blocks.

This section gives an overview of the motion control instructions for both PLCopen-based function blocks and Delta-defined function blocks. PLCopen defines the program and function block interfaces so as to achieve a standardized motion control programming environment for the languages specified in IEC61131-3. Using PLCopen-based instructions together with Delta-defined instructions reduces the costs for training and support.

Before using the instructions, please be sure that you understand the devices, symbols and the function of instructions sufficiently.

You can also refer to the **Appendices** for a quick reference of the motion control instruction list and error codes.

***Note:**
PLCopen is an organization promoting industrial control based on IEC61131-3, which is an international standard widely adopted for PLC programming. For more information regarding PLCopen, check the official website at: http://www.plcopen.org/

7.1.2 Application Notes on Motion Control Instructions

This section explains important specifications and limitations when applying motion control instructions. For detailed information of each instruction in this manual, refer to section 7.6.3 Motion Control Programming.

- **Programming languages for motion control instructions**

You can use all programming languages provided by DIADesigner-AX to create, edit, or maintain the program. The supported languages include Ladder Diagram (LD), Sequential Function Chart (SFC), Continuous Function Chart (CFC), Structured Text (ST) and Function Block Diagram (FBD).

For detailed information about the programming languages, refer to **DIADesigner-AX Software Manual.**

7.1.3 Categories of Motion Control Instructions

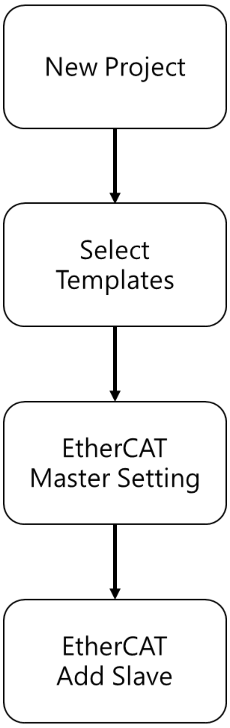
This section explains the categories of motion control instructions. The related instructions can be found in the libraries of SM3_Basic, DL_MotionControl and DL_MotionControlLight. For details, refer to **AX Series Motion Controller Manual**.

Categories	Type	Function Group	Description
Single-axis motion control instructions	Motion	Single axis positioning	"SMC": Motion instructions "MC_": PLCopen motion control instructions "DMC_": Delta motion control instructions "MC_XXX_DML": Delta motion control instructions, used with positioning axis.
		Velocity control on single axis	
		Torque control on single axis	
		Synchronized control on single axis	
	Administrative	Administrative functions on single axis	
Multiple-axis motion control instructions	Motion	Axis group movement functions	Multiple-axes motion
	Administrative	Administrative functions on multiple axes	Multiple-axes configuration, monitoring and reset function.

7.2 Creating Motion Control Project

7.2.1 Process Flowchart

The following flowchart shows the process of creating motion control project and positioning axis.

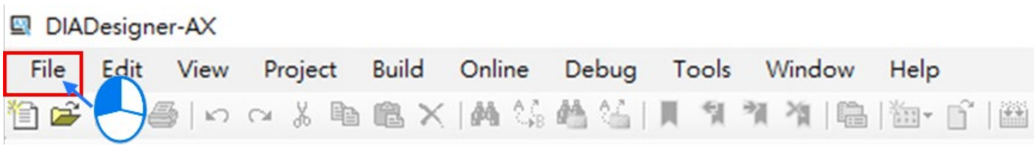


7.2.2 Process for Creating a Project

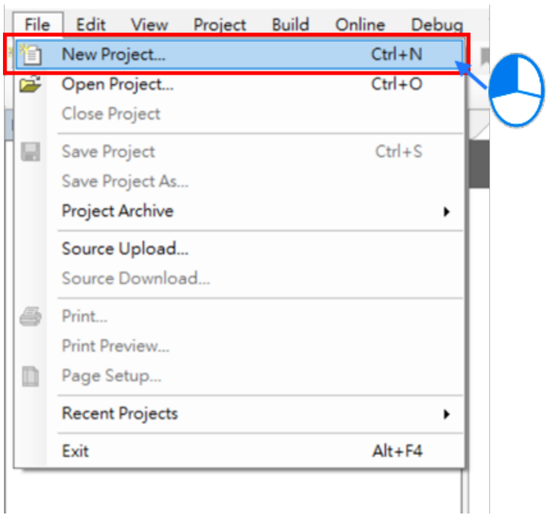
- **Create a new project**

■ Double-click on the DIADesign-AX icon to open the software.

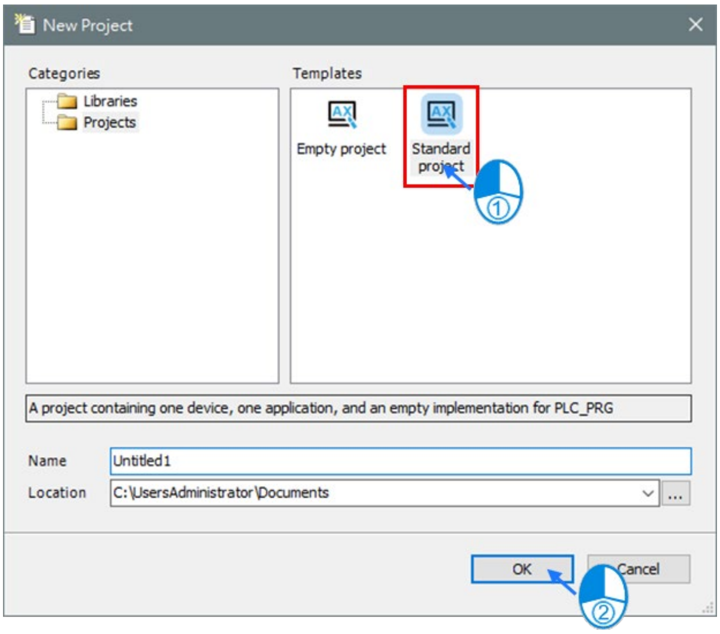
■ Click **File.**



■ Choose **New Project**



■ Type in the fields of **Name** and **Location** in the New Project window. Select **Standard project** and then click **OK**.

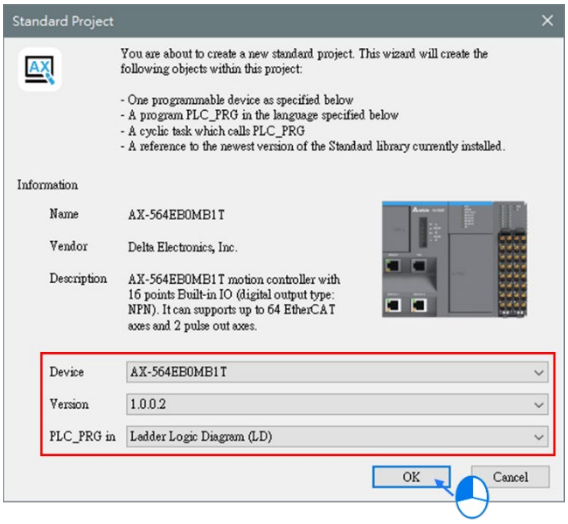


A project containing one device, one application, and an empty implementation for PLC_PRG

Name Untitled1

Location C:\UsersAdministrator\Documents

- Select AX-564EB0MB1T from the drop-down list of **Device**. And select its corresponding version in **Version** and a programming language in **PLC_PRG in**. After that, click OK.



Standard Project

You are about to create a new standard project. This wizard will create the following objects within this project:

- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Information

Name AX-564EB0MB1T

Vendor Delta Electronics, Inc.

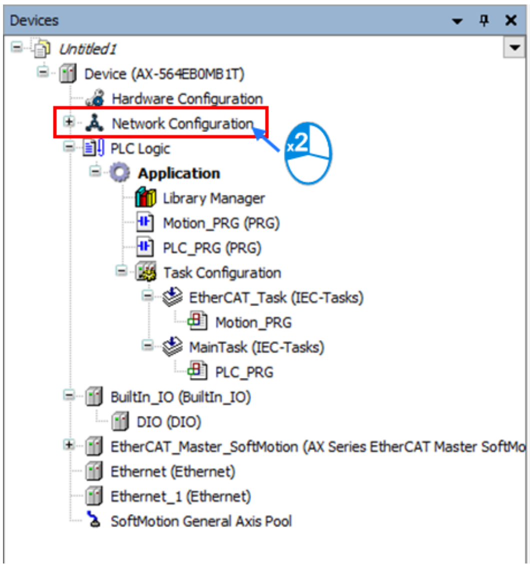
Description AX-564EB0MB1T motion controller with 16 points Built-in IO (digital output type: NPN). It can supports up to 64 EtherCAT axes and 2 pulse out axes.

Device AX-564EB0MB1T

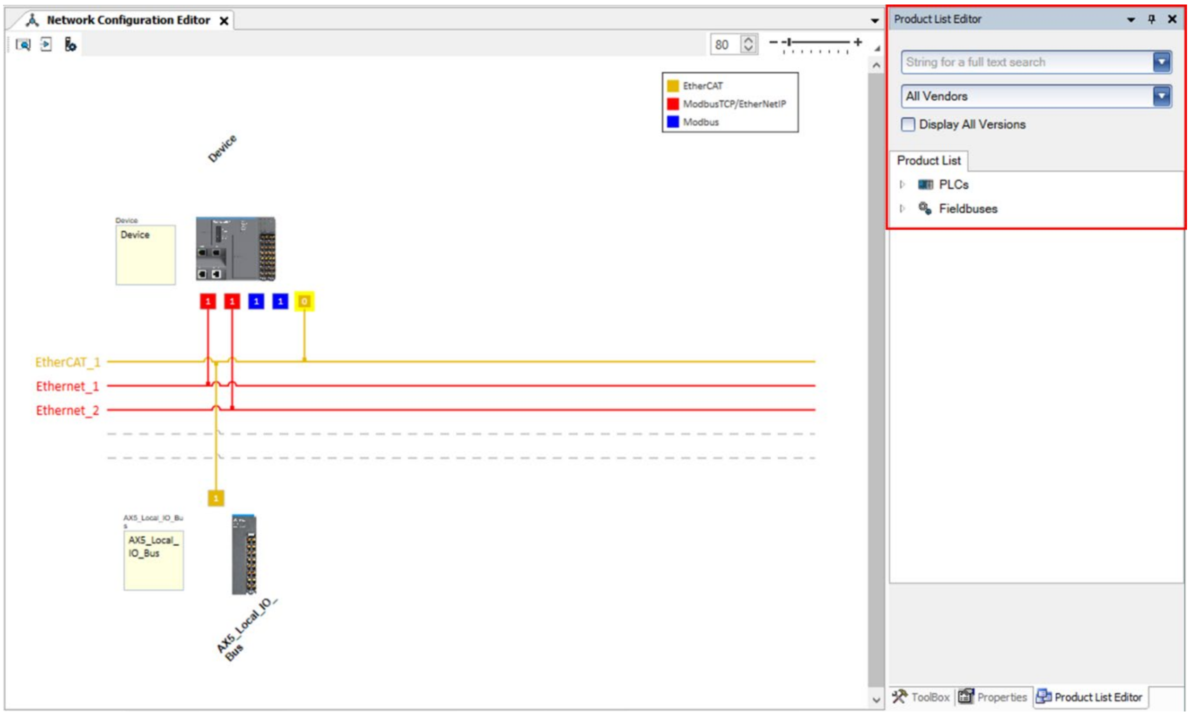
Version 1.0.0.2

PLC_PRG in Ladder Logic Diagram (LD)

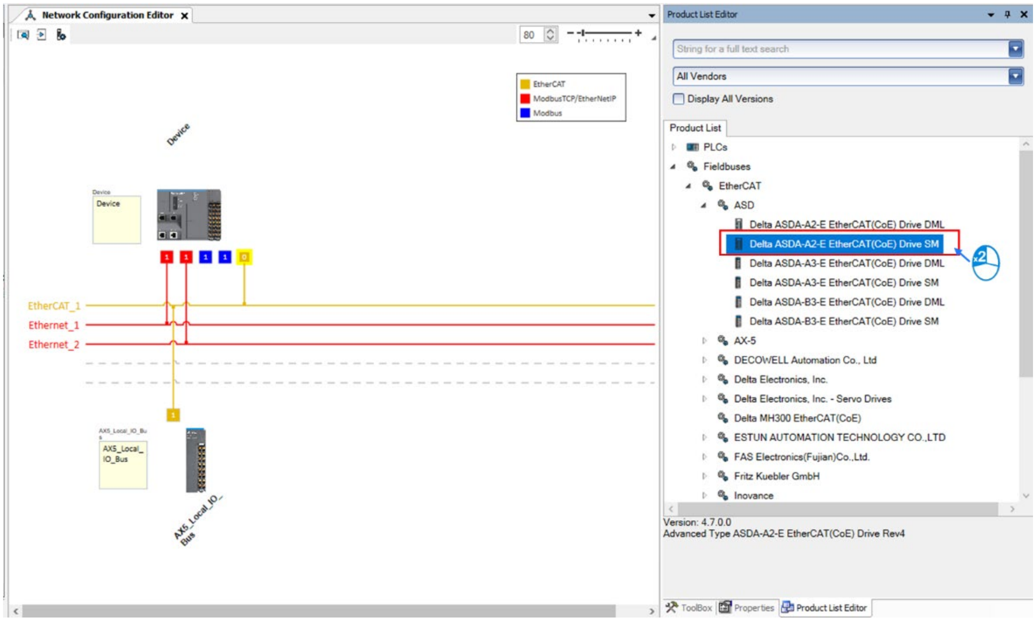
- Double-click on "Network Configuration" to continue with EtherCAT settings.



- “Network Configuration Editor” window will pop up after double-clicking. Find the target slave devices from “Product List Editor” on the right.

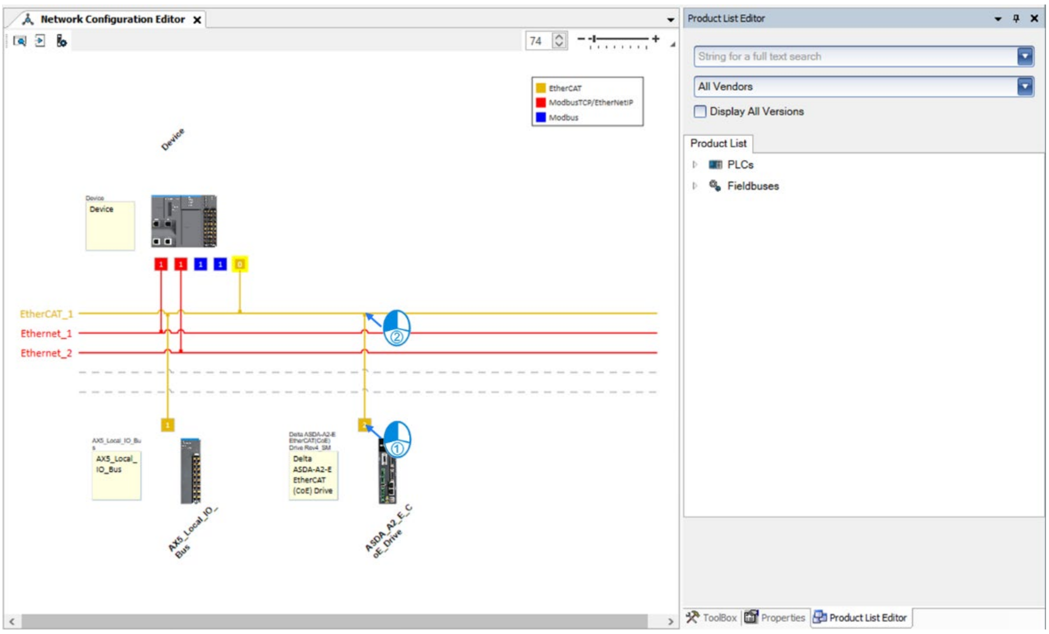


- Choose “**Servo Device**” → “**ASD**” → “**Delta ASD-A2-E EtherCAT(CoE) Drive SM**” from the product list. Then, the device will be automatically added to “**Network Configuration Editor**” after a double-click.

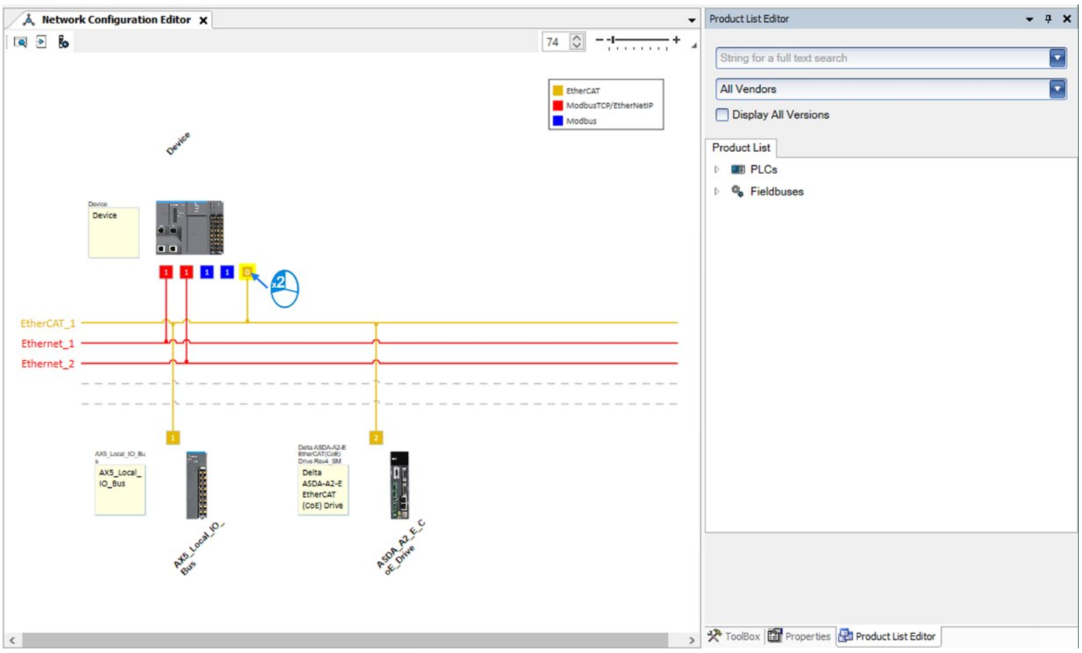


Version: 4.7.0.0
Advanced Type ASDA-A2-E EtherCAT(CoE) Drive Rev4

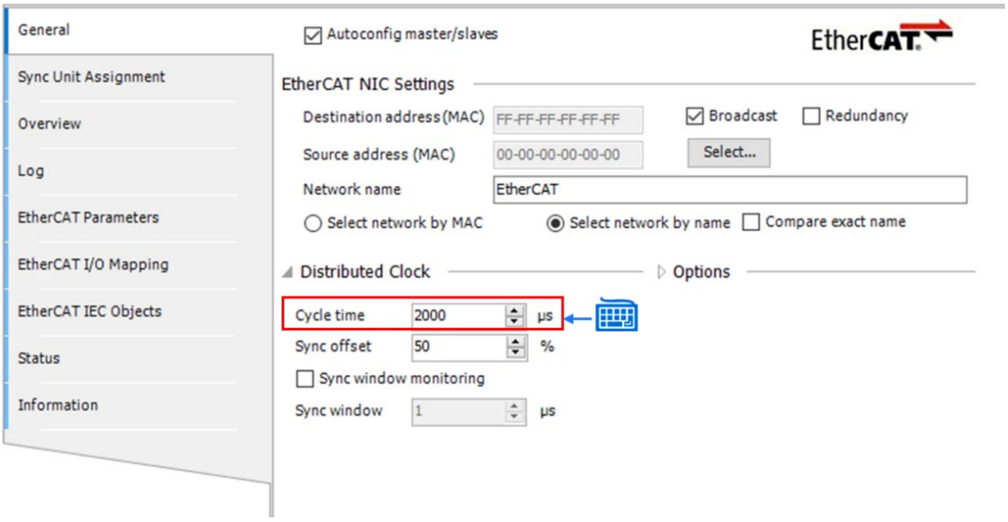
- Click and hold the left mouse button on the yellow box of slave device and drag it towards the EtherCAT main line to complete the configuration of master-slave connection.



- Double-click on the yellow box of master device to continue parameter settings for EtherCAT master device.



- EtherCAT distributed clock can be configured within master device settings.



☑ Autoconfig master/slaves

EtherCAT NIC Settings

Destination address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast ☐ Redundancy

Source address (MAC) 00-00-00-00-00-00

Network name EtherCAT

○ Select network by MAC ● Select network by name ☐ Compare exact name

Distributed Clock

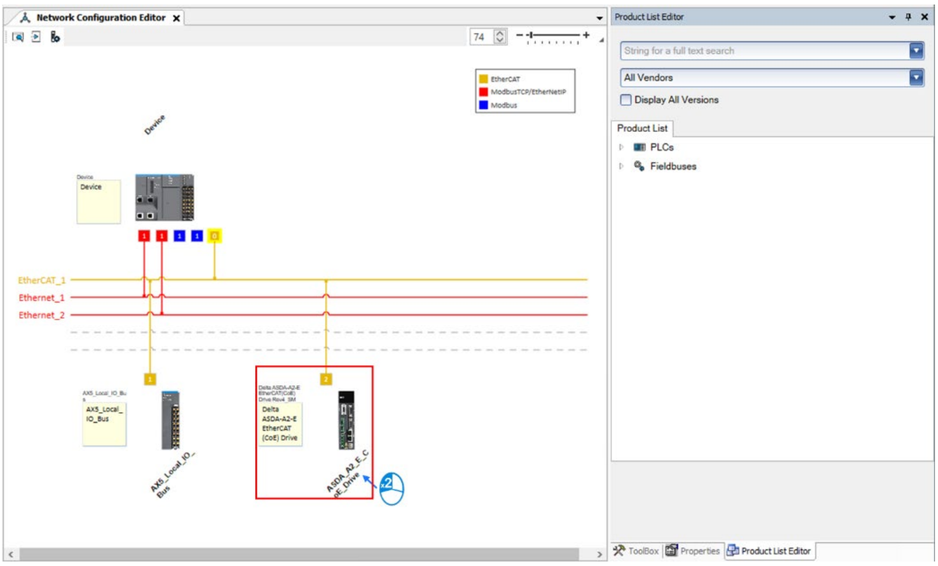
Cycle time 2000 µs

Sync offset 50 %

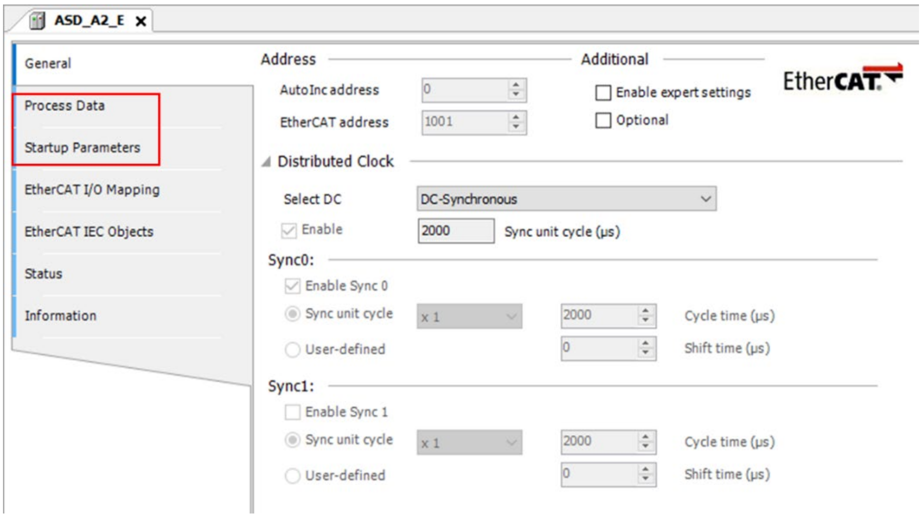
☐ Sync window monitoring

Sync window 1 µs

- Double-click on the slave device to continue EtherCAT slave device settings.



- With a double-click on the slave device, the window for setting slave configurations will be displayed, such as station address setting, and setups in "Process Data" and "Startup Parameters."



Address

AutoInc address 0

EtherCAT address 1001

Additional

☐ Enable expert settings

☐ Optional

Distributed Clock

Select DC DC-Synchronous

☑ Enable 2000 Sync unit cycle (µs)

Sync0:

☑ Enable Sync 0

● Sync unit cycle x 1 2000 Cycle time (µs)

○ User-defined 0 Shift time (µs)

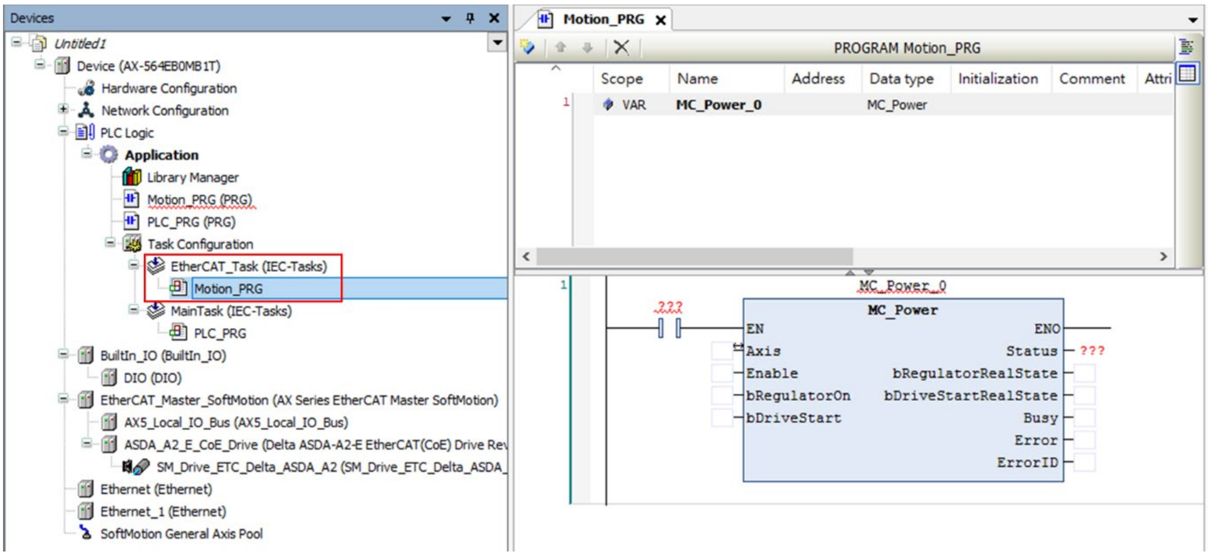
Sync1:

☐ Enable Sync 1

● Sync unit cycle x 1 2000 Cycle time (µs)

○ User-defined 0 Shift time (µs)

- Afterwards, you can start writing programs. Please use the motion function blocks in POUs under “EtherCAT+Task”, to ensure normal operation of function blocks.

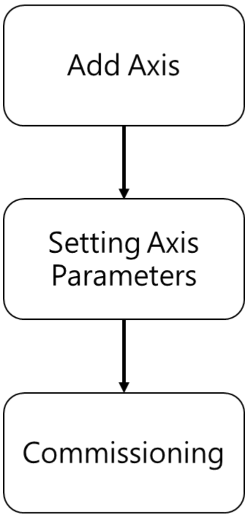


	Scope	Name	Address	Data type	Initialization	Comment	Attri
1	VAR	MC_Power_0		MC_Power			

7.3 Commissioning

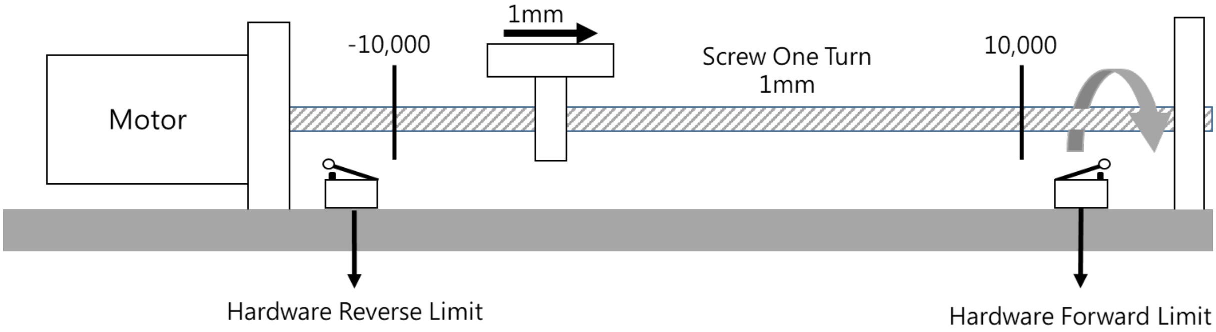
7.3.1 Procedure for Commissioning

The chart below shows the steps to build a commissioning process:

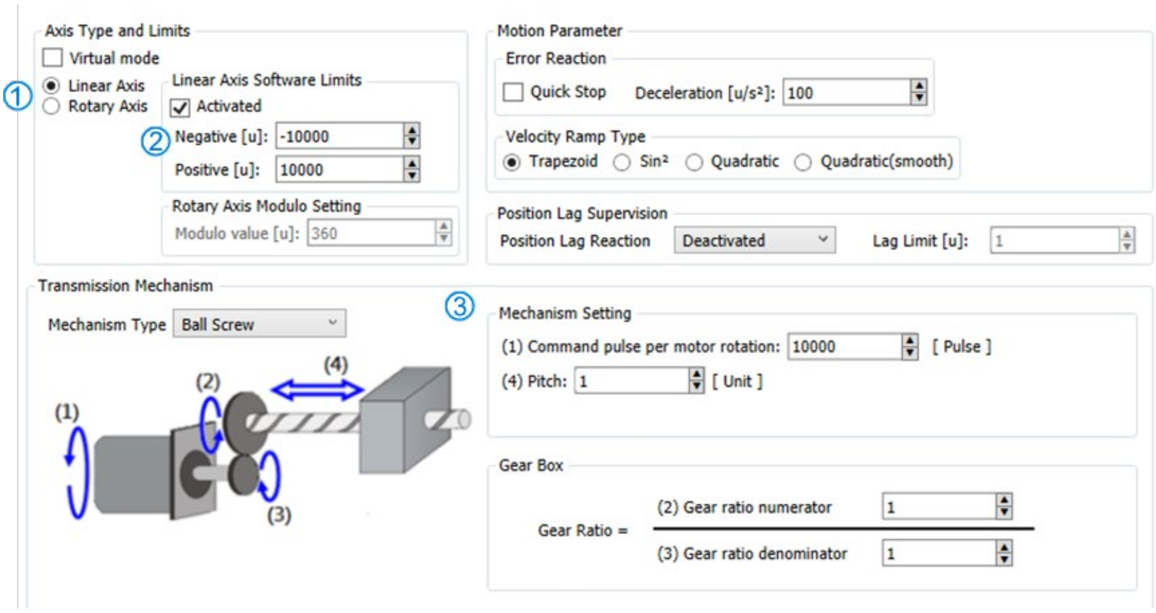


7.3.2 Example of Axis Parameter Settings

Before using software to perform commissioning, axis parameters must be set first. The figure below illustrates the setting method.



- Axis configuration screen



Axis Type and Limits

☐ Virtual mode

● Linear Axis

○ Rotary Axis

Linear Axis Software Limits

☑ Activated

Negative [u]: -10000

Positive [u]: 10000

Rotary Axis Modulo Setting

Modulo value [u]: 360

Motion Parameter

Error Reaction

☐ Quick Stop Deceleration [u/s^2]: 100

Velocity Ramp Type

● Trapezoid ○ Sin^2 ○ Quadratic ○ Quadratic(smooth)

Position Lag Supervision

Position Lag Reaction: Deactivated Lag Limit [u]: 1

Transmission Mechanism

Mechanism Type: Ball Screw

Mechanism Setting

(1) Command pulse per motor rotation: 10000 [Pulse]

(4) Pitch: 1 [Unit]

Gear Box

Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

- Parameters setting

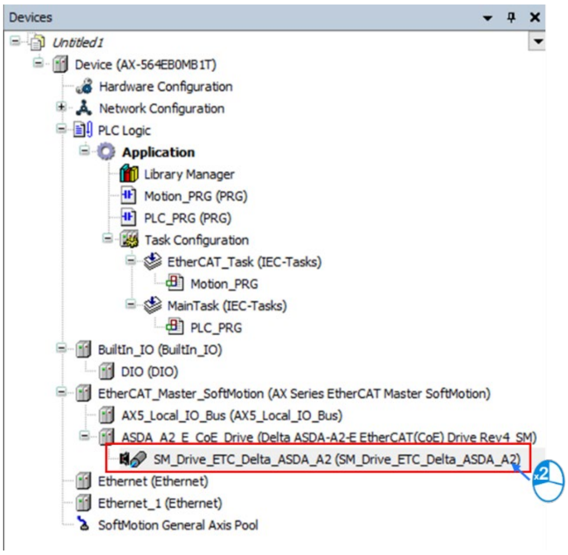
Name	Setting
Axis Type①	Linear Axis
Command pulse per motor rotation③	10,000
Pitch③ [Unit]	1^{*1}
Gear ratio denominator	128^{*2}
Gear ratio numerator	1^{*2}
Software limit_Positive②	10,000
Software limit_Negative②	-10,000

***Note:**

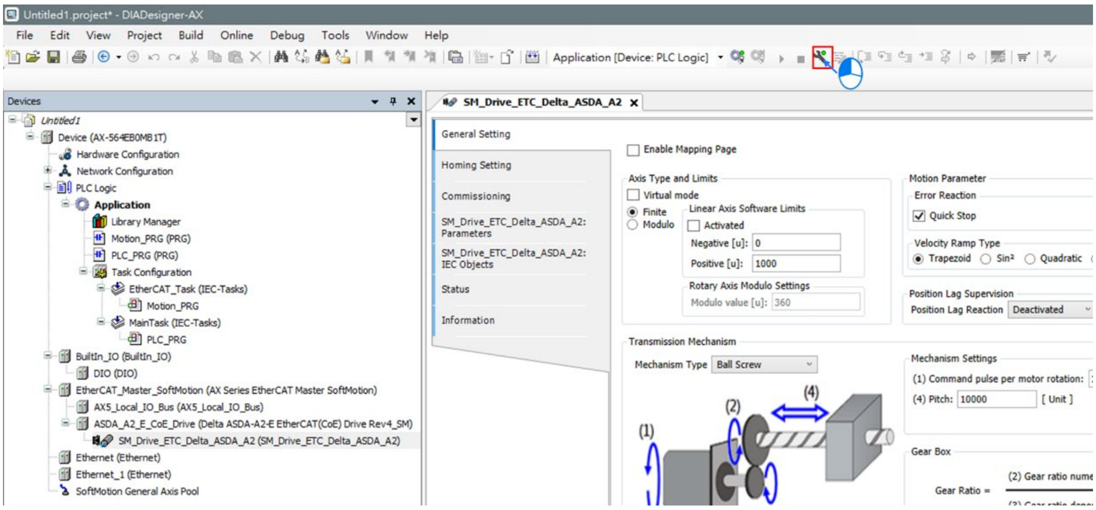
1. In case of the Unit [mm], the input parameter should be 0.001 for moving 1um.
2. It’s a must to set P1-44 and P1-45 of the servo drive.

7.3.3 Perform Axes Commissioning

- Select “SM_Drive_ETC_Delta_ASDA_A2” and double-click on it.



- Click on the “Online Config Mode” icon.



☐ Enable Mapping Page

☐ Virtual mode

● Finite ○ Modulo

☐ Activated

Negative [u]: 0

Positive [u]: 1000

Modulo value [u]: 360

☑ Quick Stop

● Trapezoid ○ Sin² ○ Quadratic

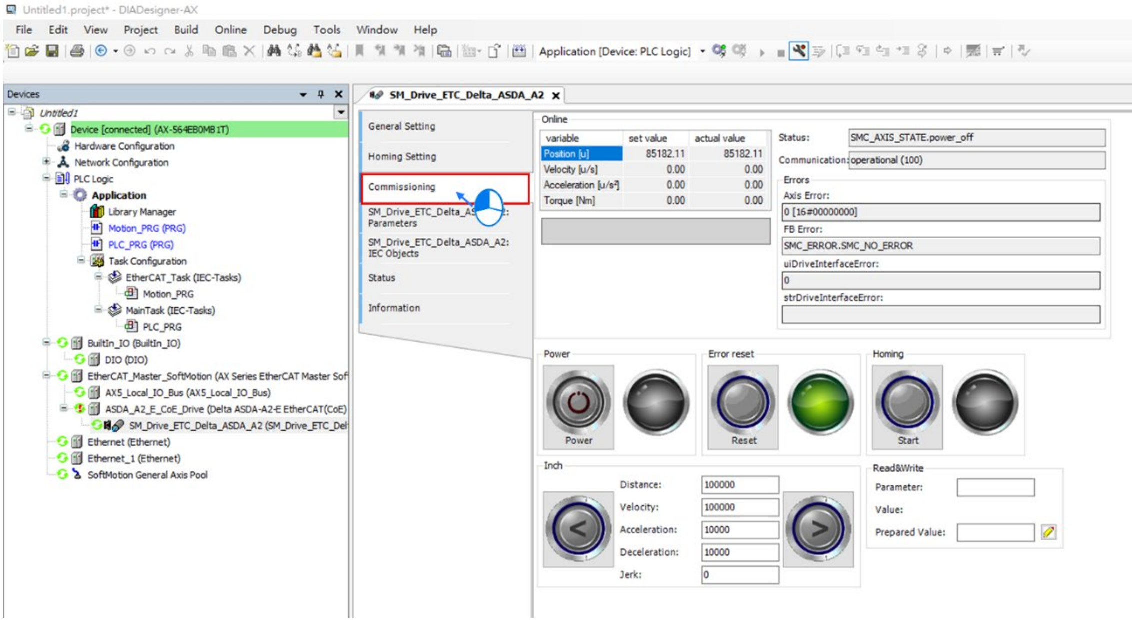
Position Lag Reaction: Deactivated

Mechanism Type: Ball Screw

(1) Command pulse per motor rotation:

(4) Pitch: 10000 [Unit]

- Open “Commissioning” tab after entering the Online Config Mode.



variable	set value	actual value
Position [u]	85182.11	85182.11
Velocity [u/s]	0.00	0.00
Acceleration [u/s²]	0.00	0.00
Torque [Nm]	0.00	0.00

Status: SMC_AXIS_STATE.power_off

Communication: operational (100)

Errors

Axis Error: 0 [16#00000000]

FB Error: SMC_ERROR.SMC_NO_ERROR

uiDriveInterfaceError: 0

strDriveInterfaceError:

Distance: 100000

Velocity: 100000

Acceleration: 10000

Deceleration: 10000

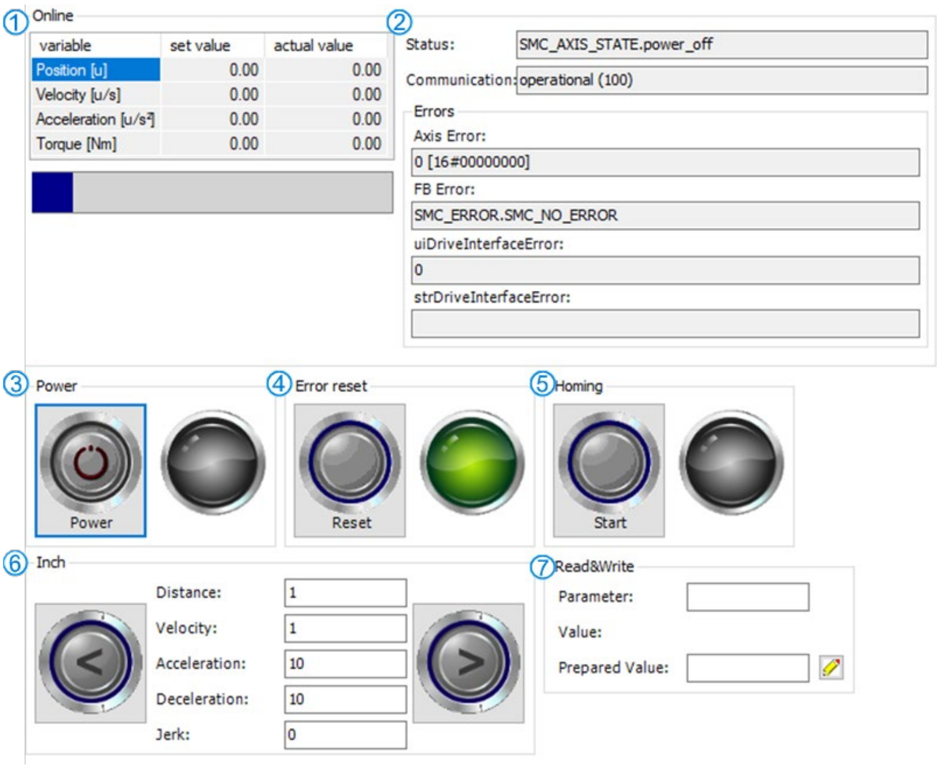
Jerk: 0

Parameter:

Value:

Prepared Value:

- See the introduction of the monitoring and commissioning screen.



variable	set value	actual value
Position [u]	0.00	0.00
Velocity [u/s]	0.00	0.00
Acceleration [u/s²]	0.00	0.00
Torque [Nm]	0.00	0.00

Status: SMC_AXIS_STATE.power_off

Communication: operational (100)

Errors

Axis Error: 0 [16#00000000]

FB Error: SMC_ERROR.SMC_NO_ERROR

uiDriveInterfaceError: 0

strDriveInterfaceError:

Distance: 1

Velocity: 1

Acceleration: 10

Deceleration: 10

Jerk: 0

Parameter:

Value:

Prepared Value:

- ① Information of axis commands

Name	Function
Position[u]	Command position and actual position
Velocity[u/s]	Command velocity and actual velocity
Acceleration[u/s^2]	Command acceleration and actual acceleration
Torque[Nm]	Command torque and actual torque

- ② Axis status and communication status

Name	Function
Status	Axis status
Communication	Communication status

- ③ Power: Enable/ disable the axis by setting the power to ON/ OFF.
- ④ Error reset: Clear error messages of the servo axis.
- ⑤ Homing: Make the axis back to the home position (origin).
- ⑥ Inch

Name	Function
Distance	Moving distance
Velocity	Moving velocity
Acceleration	Acceleration rate
Deceleration	Deceleration rate
Jerk	Command value of jerk

- ⑦ Read&Write: Read/write upper axis parameters. You can read and modify the parameters in Object Dictionary in the way as follows.
  Read and write the parameter 0x6098 in the object dictionary

$$16\#\boxed{1}\boxed{6098}\boxed{00}$$

1 = fixed number

6098 =the parameter to be read and written

00 = subindex of the parameter

1. Convert 0x1609800 to demical number as 23,107,584
2. Change 23,107,584 to -23,107,584
3. Enter -23,107,584 in the “Parameter” field to read the parameter “0x6098”.

7.4 Motion Control Device

7.4.1 Overview

Motion control devices are mainly used for configuring parameters for motion axes. In most applications, you can set up axis parameters in DIADesigner-AX software, a convenient environment for you, where axis parameters required for configuring motion control on axis are defined as Structure. A Structure is a data type applicable to group the data elements together.

7.4.2 Introduction to Axis

The axis is used to perform motion control in the system and includes real servo drives, encoders and virtual servo drives. The following table shows the axis types:

Type	Description
Positioning axis*3	Achieve basic positioning control via EtherCAT, such as functions of absolute positioning, relative positioning, and etc.
Velocity axis*3	Achieve velocity control and torque control via EtherCAT. (as seen in CiA 402 Velocity Mode)
Synchronous axis*3	Achieve servo motor control and basic positioning control via EtherCAT, as well as synchronous motion control like electronic cam function.
Pulse-type axis	Achieve real servo motor control with pulses.
Virtual axis	Execute motion control commands without using a real servo motor.
Encoder axis	Use real encoder as feedback signals.
Virtual encoder axis	Can only be used in the program without encoders.

***Note 1:** Positioning and velocity axes must work with the function library of DL_MotionControlLight.

***Note 2:** Synchronous axes must work with DL_MotionControl and the function library of SM3_Basic.

***Note 3: Modes of CiA 402 are as below.**

- Profile Position Mode (PP)
- Velocity Mode (VL)
- Profile Velocity Mode (PV)
- Profile Torque Mode (PT)
- Homing Mode (HM)
- Cyclic Synchronous Position Mode (CSP)
- Cyclic Synchronous Velocity Mode (CSV)
- Cyclic Synchronous Torque Mode (CST)

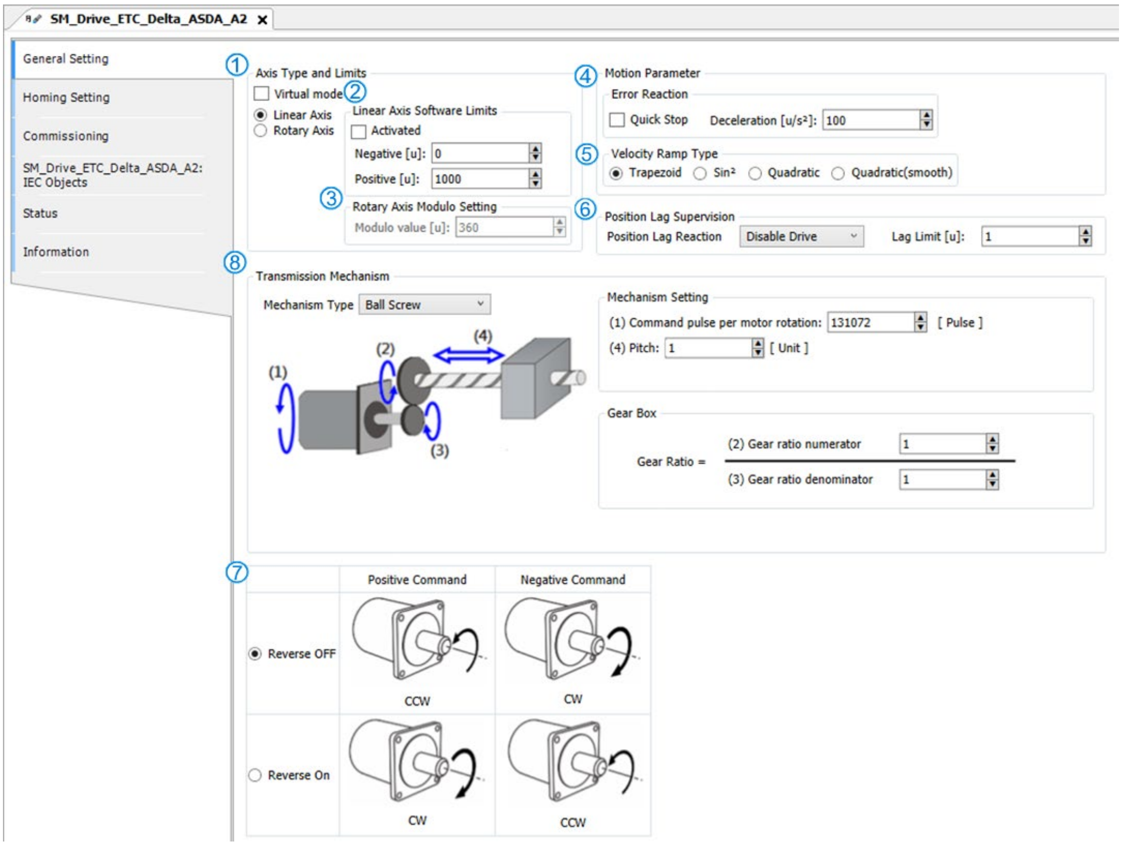
Available modes of CiA 402 for the different EtherCAT axis types:

Type	Modes of CiA 402
Positioning axis*3	PP, PV, PT, HM, VL
Velocity axis*3	VL, PT
Synchronous axis*3	CSP, CSV, CST, HM, PT

7.4.2.1 About Axis Parameters

After a servo axis is created, the corresponding axis parameters will be generated as well. The following table details the relevant description.

- Synchronous Axis



Axis Type and Limits
☐ Virtual mode
● Linear Axis
○ Rotary Axis

Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 1000

Rotary Axis Modulo Setting
Modulo value [u]: 360

Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 100

Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

Position Lag Supervision
Position Lag Reaction: Disable Drive Lag Limit [u]: 1

Transmission Mechanism
Mechanism Type: Ball Screw

Mechanism Setting
(1) Command pulse per motor rotation: 131072 [Pulse]
(4) Pitch: 1 [Unit]

Gear Box
Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

● Reverse OFF
○ Reverse On

■ ① Axis Type and Limits

Name	Function
Virtual	Activate virtual axes.
Linear Axis / Rotary Axis	Set to be linear axis or rotary axis.

■ ② Linear Axis Software Limits

Name	Function
Activated	Activate software limits (only supports linear axes)
Negative[u]	Reverse software limit.
Positive[u]	Forward software limit.

■ ③ Rotary Axis Modulo Setting

Name	Function
Modulo Value[u]	Set the range of rotation for a turn. (only supports rotary axes)

- ④ Error Reaction

Name	Function
Quick Stop	Emergency stop for axes
Deceleration[u/s2]	Decelerate to stop for axes (effective when Quick Stop is inactive)

- ⑤ Velocity Ramp Type

Name	Function
Trapezoid/Sin2/Quadratic/ Quadratic(Smooth)	Motion curves setting for axes

- ⑥ Position Lag Supervision

Name	Function
Position Lag Reaction	Set the reaction for position lag.
Lag Limit [u]	Set the deviation range between command and feedback

- ⑦ Positive / Negative Command

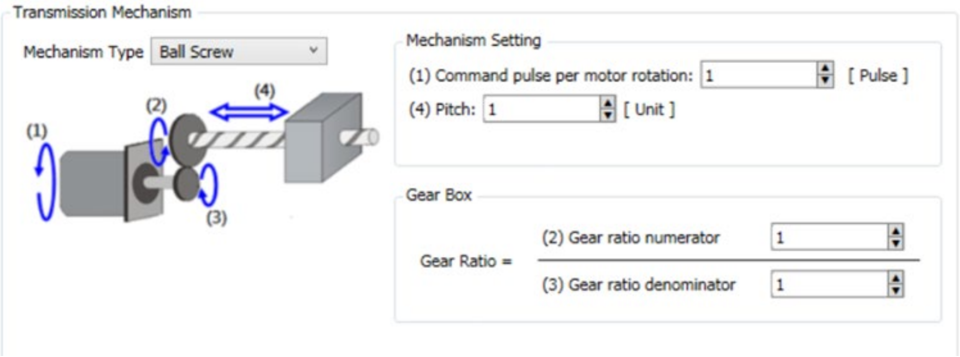
Name	Function
Reverse OFF / ON	Enable or disable reverse function for positive/negative command setting.

- ⑧ Transmission Mechanism
- Servo Gear Ratio Setting

Name	Function
Unit Numerator	Numerator of the electronic gear ratio
Unit Denominator	Denominator of the electronic gear ratio

Descriptions of different mechanism types are as follows:

- Ball Screw



Transmission Mechanism

Mechanism Type: Ball Screw

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

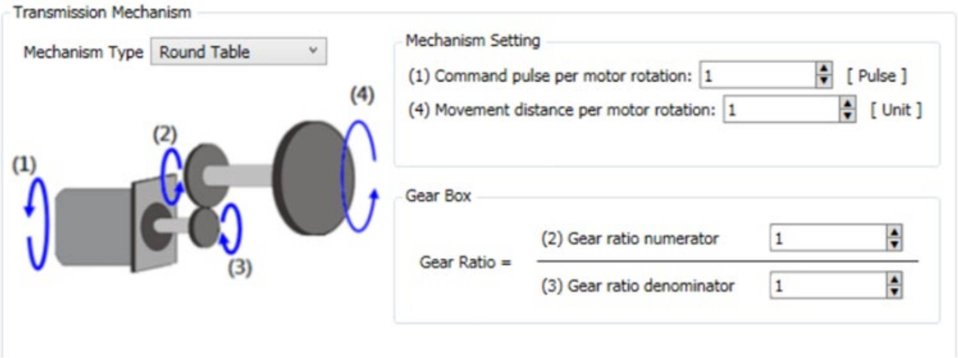
(4) Pitch: 1 [Unit]

Gear Box

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Pitch	The distance between screw threads

- Round Table



Transmission Mechanism

Mechanism Type: Round Table

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

(4) Movement distance per motor rotation: 1 [Unit]

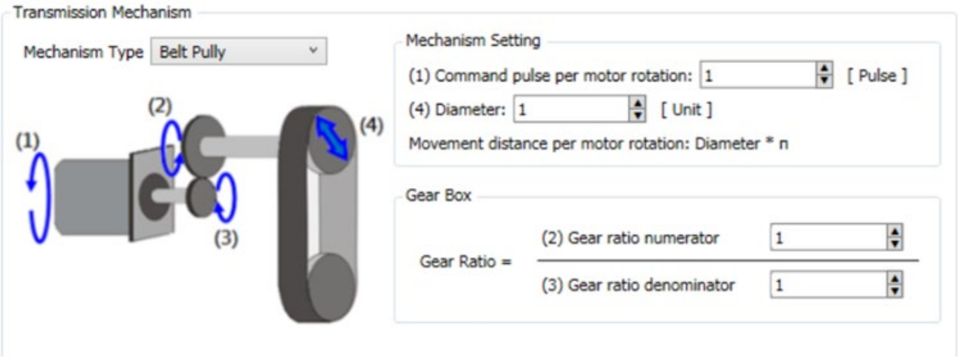
Gear Box

(2) Gear ratio numerator: 1

(3) Gear ratio denominator: 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Movement distance per motor rotation	Movement distance for one full motor rotation

- Belt Pully



Transmission Mechanism

Mechanism Type: Belt Pully

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

(4) Diameter: 1 [Unit]

Movement distance per motor rotation: Diameter * π

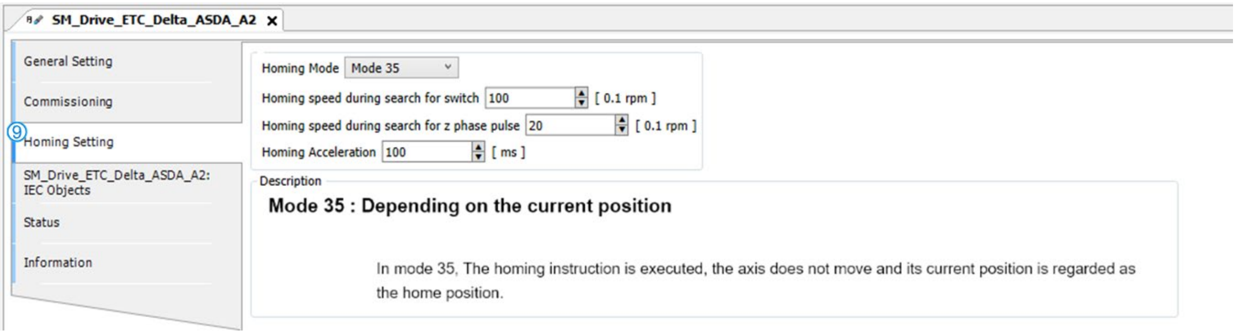
Gear Box

(2) Gear ratio numerator: 1

(3) Gear ratio denominator: 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Diameter* (Movement distance per motor rotation: Diameter X π)	Diameter (Movement distance per motor rotation: Diameter X π)

■ ⑨ Homing Setting



SM_Drive_ETC_Delta_ASDA_A2

- General Setting
- Commissioning
- ⑨ Homing Setting
- SM_Drive_ETC_Delta_ASDA_A2: IEC Objects
- Status
- Information

Homing Mode: Mode 35

Homing speed during search for switch: 100 [0.1 rpm]

Homing speed during search for z phase pulse: 20 [0.1 rpm]

Homing Acceleration: 100 [ms]

Description

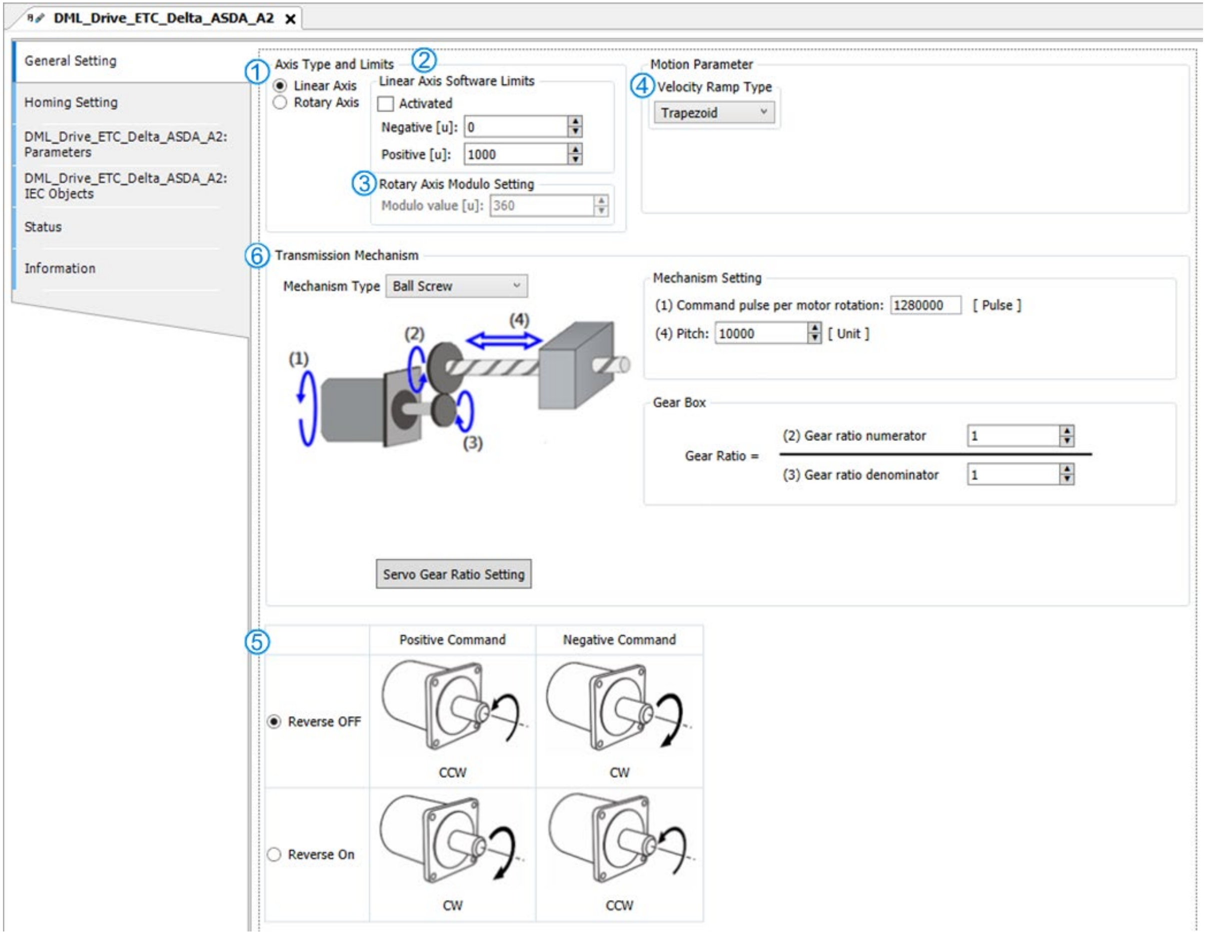
Mode 35 : Depending on the current position

In mode 35, The homing instruction is executed, the axis does not move and its current position is regarded as the home position.

Name	Function
Homing Mode	Select a homing mode.
Homing Speed during search for switch	Set the homing speed during search for the home switch.
Homing Speed during search for z phase pulse	Set the homing speed during search for Z phase pulse.
Homing Acceleration	Set the acceleration rate of homing.

- **Positioning Axis**

■ Positioning axis – Delta servos



DML_Drive_ETC_Delta_ASDA_A2

- General Setting
- Homing Setting
- DML_Drive_ETC_Delta_ASDA_A2: Parameters
- DML_Drive_ETC_Delta_ASDA_A2: IEC Objects
- Status
- Information

① Axis Type and Limits

● Linear Axis

○ Rotary Axis

② Linear Axis Software Limits

☐ Activated

Negative [u]: 0

Positive [u]: 1000

③ Rotary Axis Modulo Setting

Modulo value [u]: 360

Motion Parameter

④ Velocity Ramp Type: Trapezoid

⑥ Transmission Mechanism

Mechanism Type: Ball Screw

Mechanism Setting

(1) Command pulse per motor rotation: 1280000 [Pulse]

(4) Pitch: 10000 [Unit]

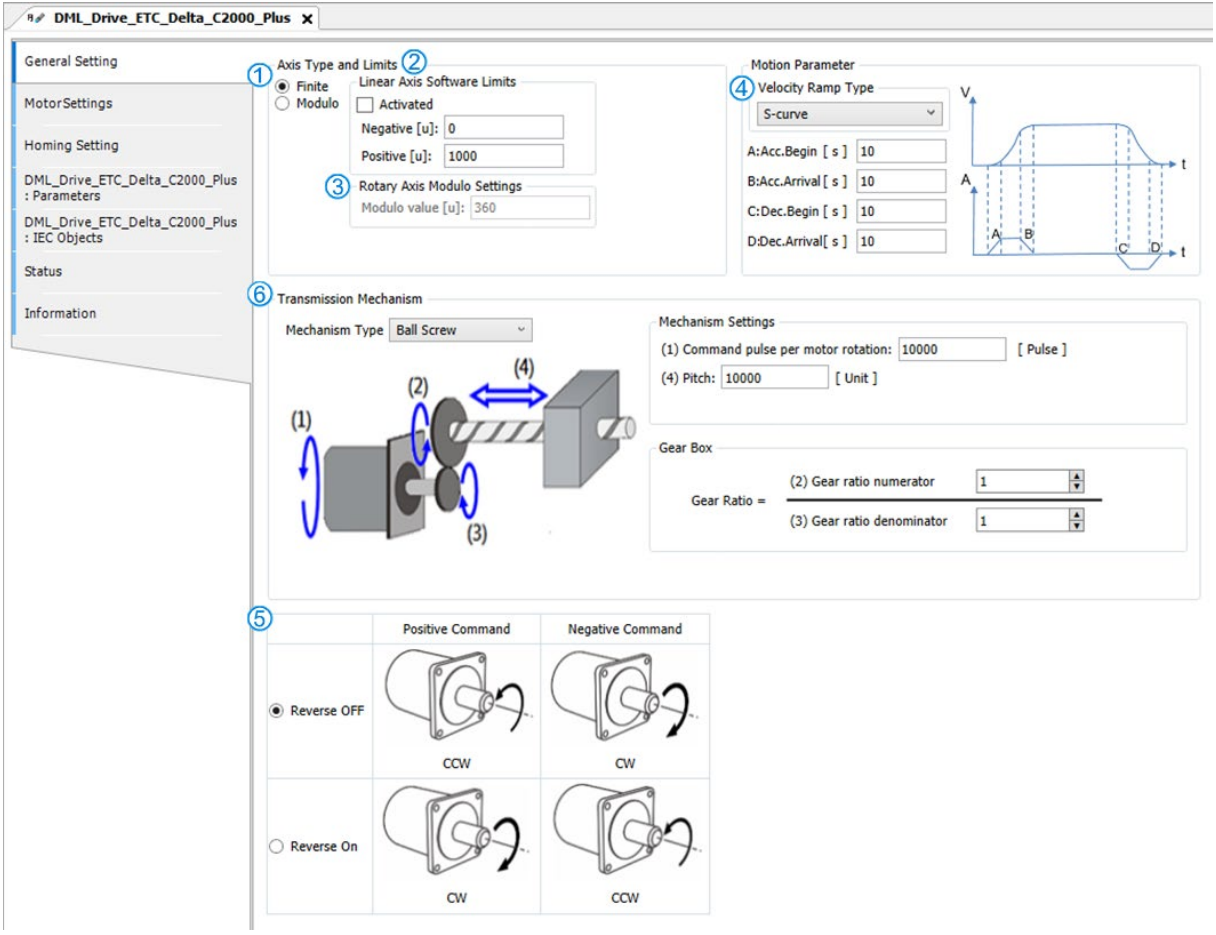
Gear Box

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

⑤

	Positive Command	Negative Command
● Reverse OFF	CCW	CW
○ Reverse On	CW	CCW

- Positioning axis – Delta inverters



Axis Type and Limits
● Finite
○ Modulo

Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 1000

Rotary Axis Modulo Settings
Modulo value [u]: 360

Motion Parameter
Velocity Ramp Type: S-curve
A:Acc.Begin [s] 10
B:Acc.Arrival [s] 10
C:Dec.Begin [s] 10
D:Dec.Arrival[s] 10

Transmission Mechanism
Mechanism Type: Ball Screw

Mechanism Settings
(1) Command pulse per motor rotation: 10000 [Pulse]
(4) Pitch: 10000 [Unit]

Gear Box
Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

	Positive Command	Negative Command
● Reverse OFF	CCW	CW
○ Reverse On	CW	CCW

- ① Axis Type and Limits

Name	Function
Linear Axis / Rotary Axis	Set to be linear axis or rotary axis.

- ② Linear Axis Software Limits

Name	Function
Activated	Activate software limits (only supports linear axes)
Negative[u]	Reverse software limit.
Positive[u]	Forward software limit.

- ③ Rotary Axis Modulo Setting

Name	Function
Modulo Value[u]	Set the range of rotation for a turn. (only supports rotary axes)

- ④ Velocity Ramp Type

For Delta servos

Name	Function
Trapezoid/Sin2	Motion curves setting for axes

For Delta inverters

Name	Function
Trapezoid/S-Curve	Motion curves setting for axes
A：Acc.Begin	Time setting for the S-Curve to begin the acceleration rate 1 (s)
B：Acc.Arrival	Time setting for the S-Curve to end the acceleration 2 (s)
C：Dec.Begin	Time setting for the S-Curve to begin the deceleration rate 1 (s)
D：Dec.Arrival	Time setting for the S-Curve to end the deceleration 2 (s)

- ⑤ Positive / Negative Command

Name	Function
Reverse OFF / On	Enable or disable reverse function for positive/negative command setting.

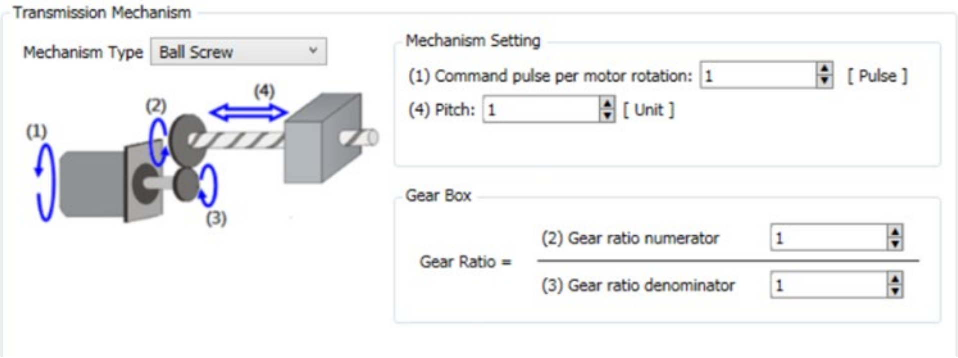
- ⑥ Transmission Mechanism

◆ Servo Gear Ratio Setting

Name	Function
Unit Numerator	Numerator factor of the electronic gear unit
Unit Denominator	Denominator factor of the electronic gear unit

Descriptions of different machanism types are as follows:

◆ Ball Screw



Transmission Mechanism

Mechanism Type: Ball Screw

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

(4) Pitch: 1 [Unit]

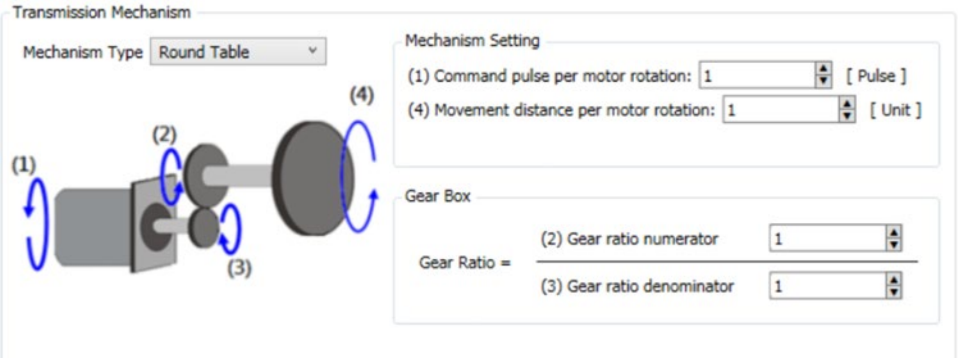
Gear Box

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio

Name	Function
(3) Gear ratio denominator	Denominator of gear ratio
(4) Pitch	The distance between screw threads

◆ Round Table



Transmission Mechanism

Mechanism Type: Round Table

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

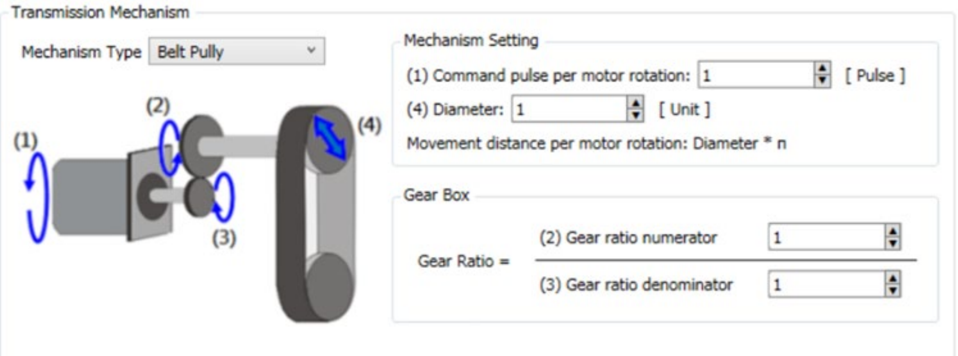
(4) Movement distance per motor rotation: 1 [Unit]

Gear Box

Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Movement distance per motor rotation	Movement distance for one full motor rotation

◆ Belt Pully



Transmission Mechanism

Mechanism Type: Belt Pully

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

(4) Diameter: 1 [Unit]

Movement distance per motor rotation: Diameter * π

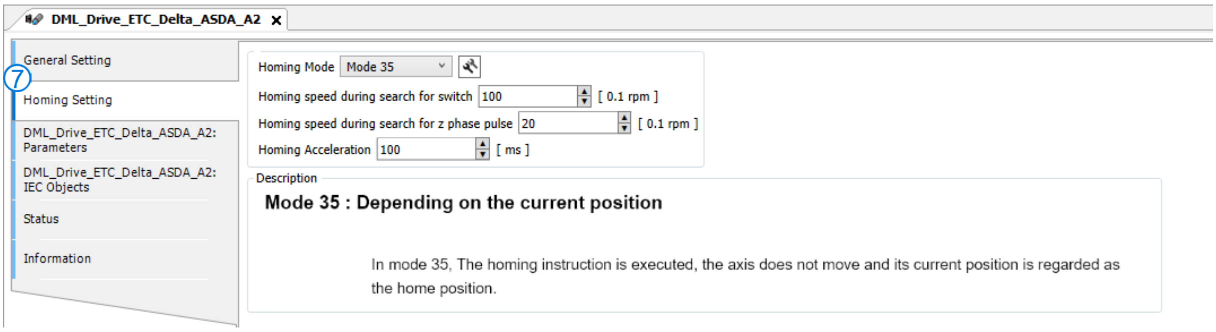
Gear Box

Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Diameter (Movement distance per motor rotation：Diameter X π)	Diameter (Movement distance per motor rotation: Diameter X π)

- ⑦ Homing Setting

  For Delta servos



Homing Mode: Mode 35

Homing speed during search for switch: 100 [0.1 rpm]

Homing speed during search for z phase pulse: 20 [0.1 rpm]

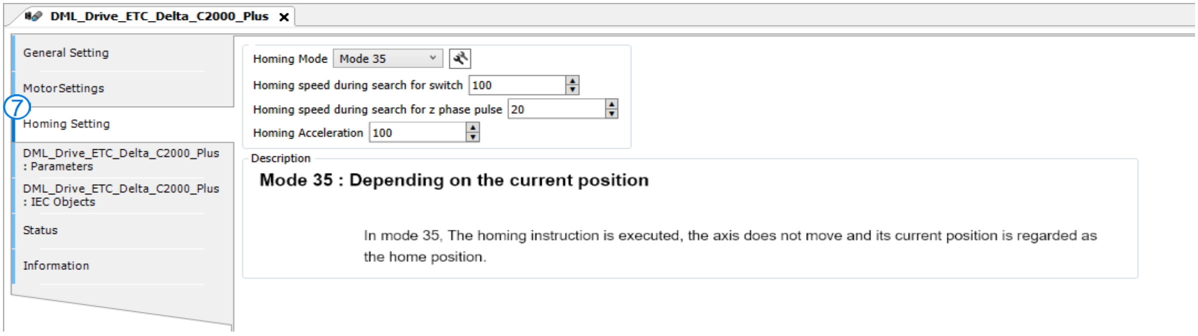
Homing Acceleration: 100 [ms]

Description

Mode 35 : Depending on the current position

In mode 35, The homing instruction is executed, the axis does not move and its current position is regarded as the home position.

For Delta inverters (the following series supported: C2000Plus, CH2000, MH300, DDF V1.0.10 or later)



Homing Mode: Mode 35

Homing speed during search for switch: 100

Homing speed during search for z phase pulse: 20

Homing Acceleration: 100

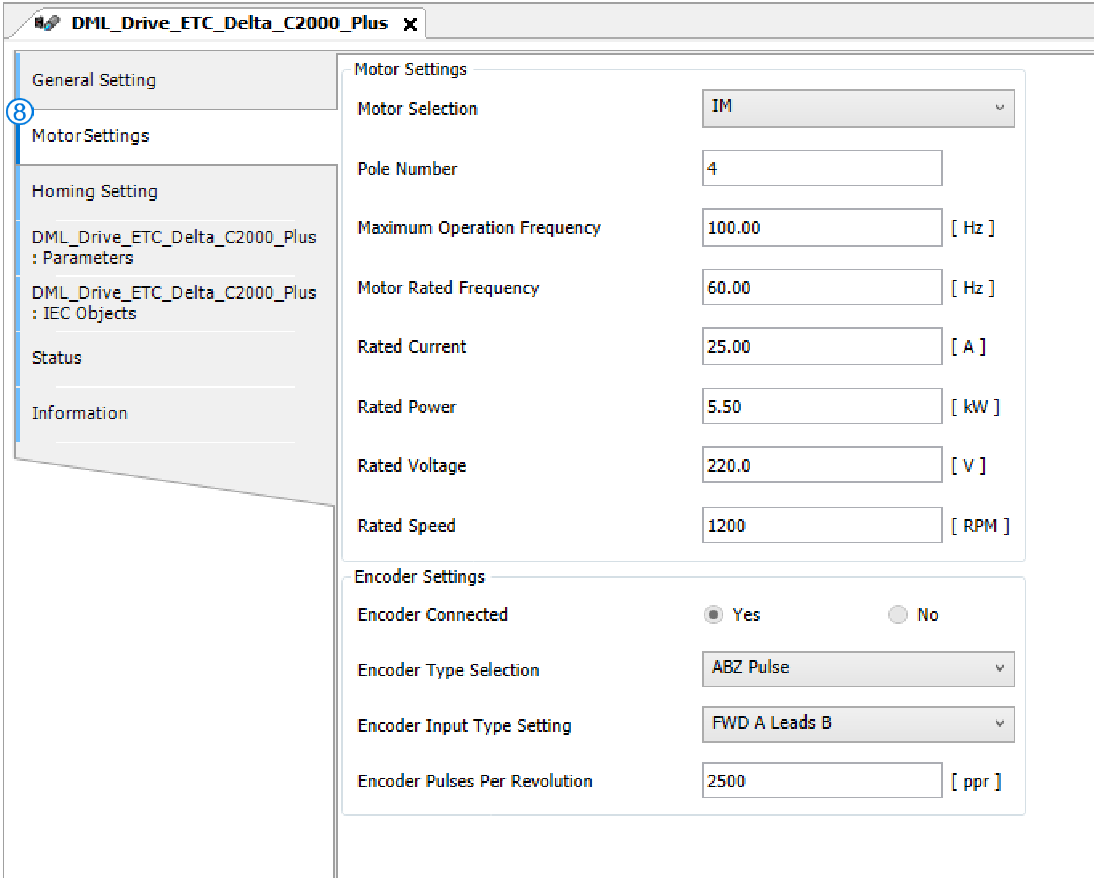
Description

Mode 35 : Depending on the current position

In mode 35, The homing instruction is executed, the axis does not move and its current position is regarded as the home position.

Name	Function
Homing Mode	Configure homing mode setting.
Homing Speed during search for switch	Set the homing speed during search for switch.
Homing Speed during search for z phase pulse	Set the homing speed during search for Z phase pulse.
Homing Acceleration	Set the homing acceleration rate.

- ⑧Motor Settings – Delta inverters



Motor Settings

Motor Selection: IM

Pole Number: 4

Maximum Operation Frequency: 100.00 [Hz]

Motor Rated Frequency: 60.00 [Hz]

Rated Current: 25.00 [A]

Rated Power: 5.50 [kW]

Rated Voltage: 220.0 [V]

Rated Speed: 1200 [RPM]

Encoder Settings

Encoder Connected: ● Yes ○ No

Encoder Type Selection: ABZ Pulse

Encoder Input Type Setting: FWD A Leads B

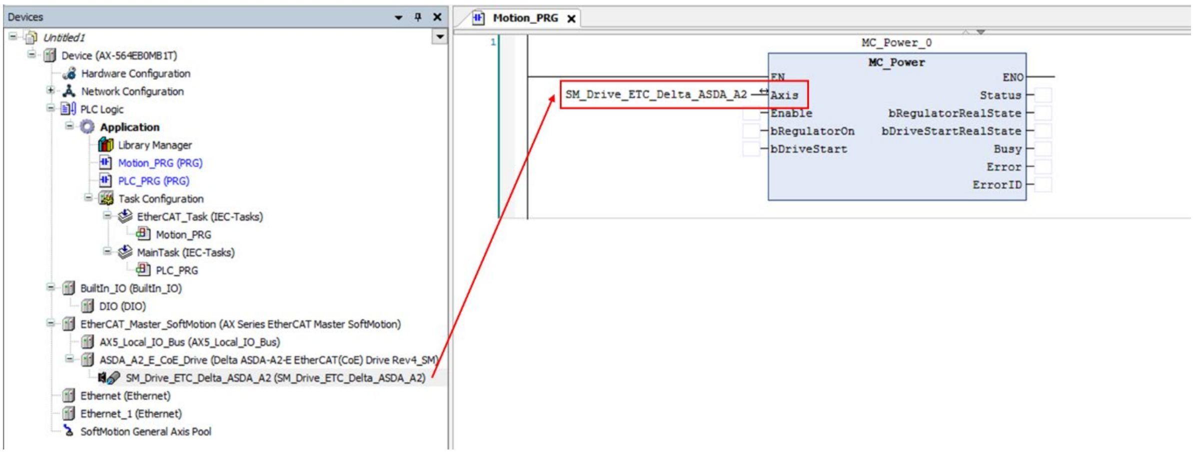
Encoder Pulses Per Revolution: 2500 [ppr]

Name	Function
Motor Selection	Select a motor type.
Pole Number	Set up pole number.
Maximum Operation Frequency	Set up a maximum operation frequency.
Motor Rated Frequency	Set up a rated motor frequency.
Rated Current	Set up rated current.
Rated Power	Set up rated power.
Rated Voltage	Set up rated voltage.
Rated Speed	Set up rated speed.
Encoder Connected	An encoder is connected.
Encoder Type Selection	Select an encoder type.
Encoder Input Type Setting	Set up an encoder input type.
Encoder Pulses Per Revolution	Set up the number of pulses per revolution for an encoder

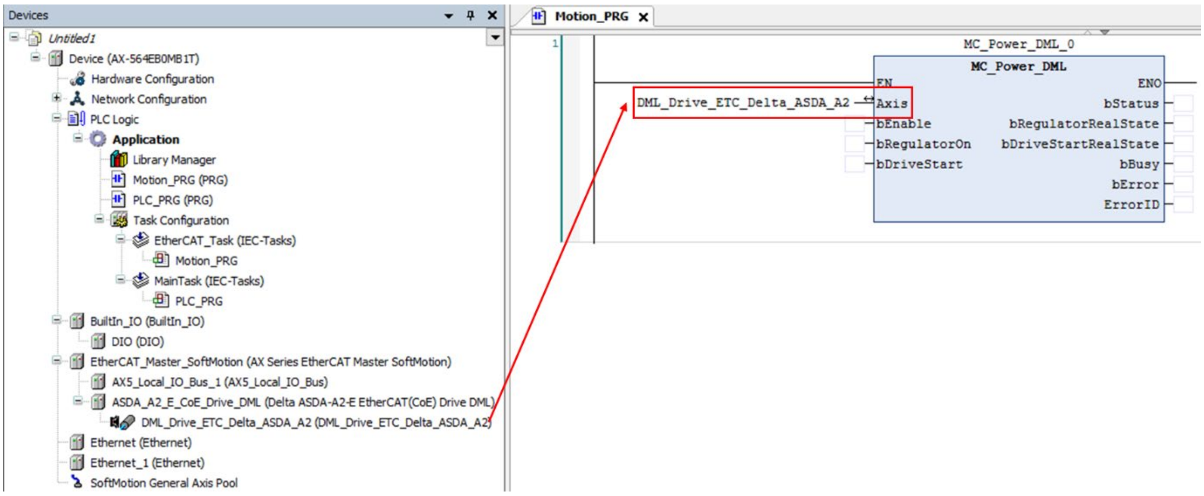
7.4.2.2 Axis Application in Program

After a servo axis is newly added in the project, the name of the servo axis will be generated automatically (you are allowed to change the name) and input to the function block.

- **Synchronous Axis**

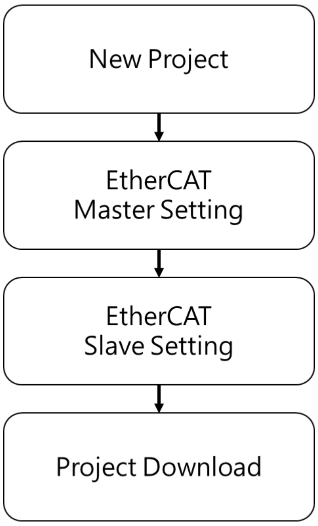


- **Positioning Axis**

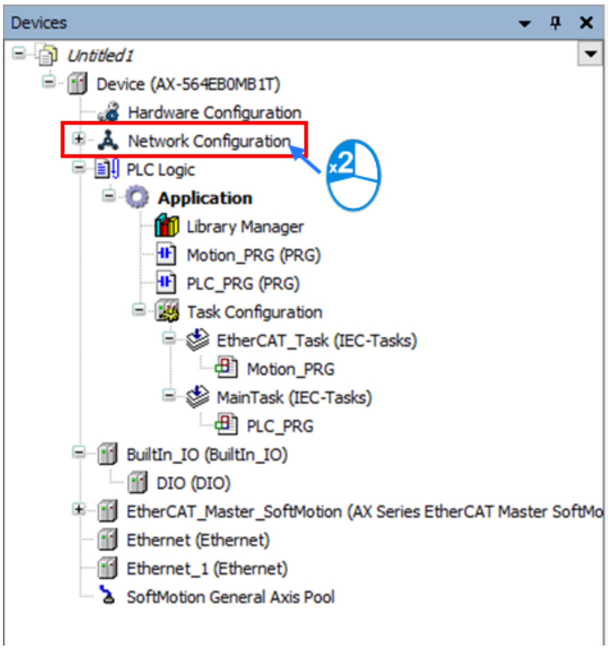


7.4.3 Procedure for Single-axis Configuration

- The procedure for axis settings is shown as follows. For more details of creating new projects, please find section 7.2.

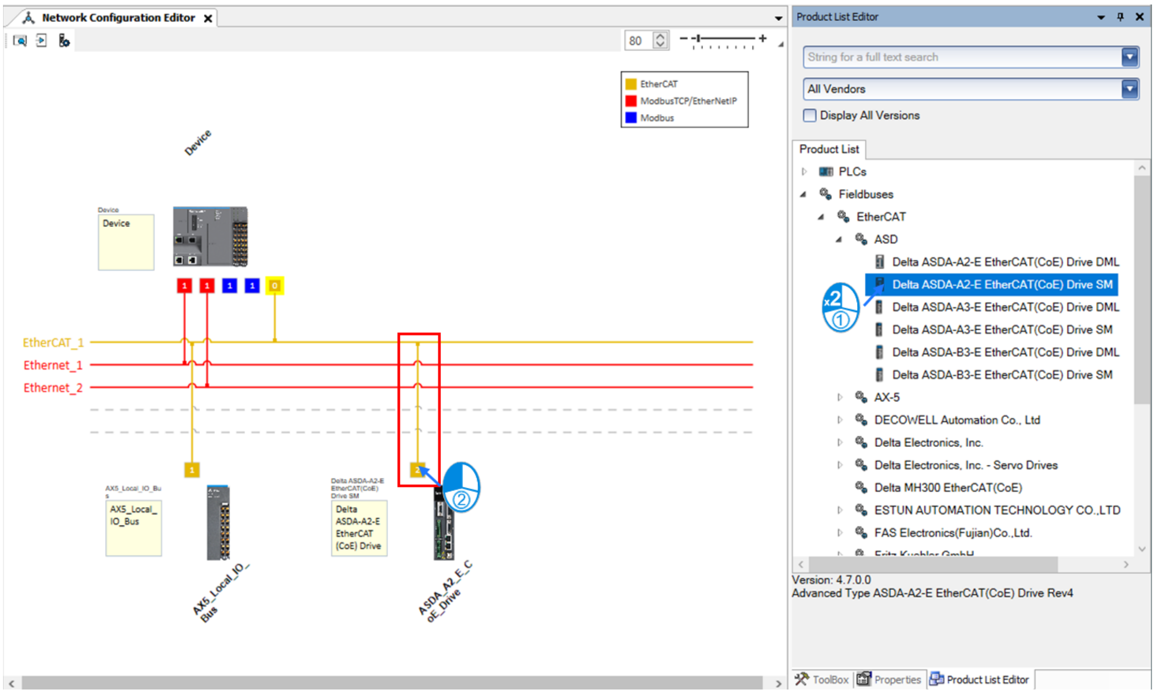


- Configure EtherCAT settings after opening the project. First, click “Network Configuration.”



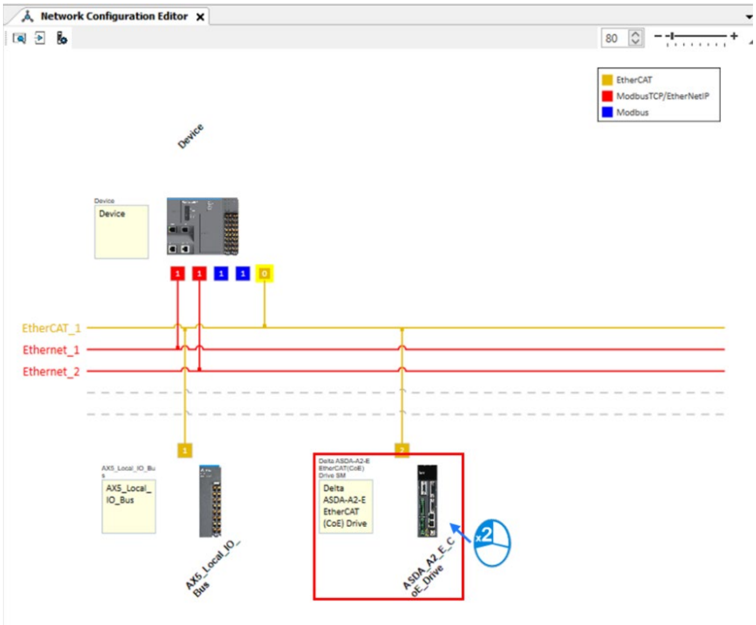
Untitled1
Device (AX-564EB0MB1T)
Hardware Configuration
Network Configuration
PLC Logic
Application
Library Manager
Motion_PRG (PRG)
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task (IEC-Tasks)
Motion_PRG
MainTask (IEC-Tasks)
PLC_PRG
BuiltIn_IO (BuiltIn_IO)
DIO (DIO)
EtherCAT_Master_SoftMotion (AX Series EtherCAT Master SoftMo
Ethernet (Ethernet)
Ethernet_1 (Ethernet)
SoftMotion General Axis Pool

- Click “Delta ASDA-A2-E EtherCAT(CoE) Drive_SM” *1 after entering Network Configuration page and connect 1 to the line above.

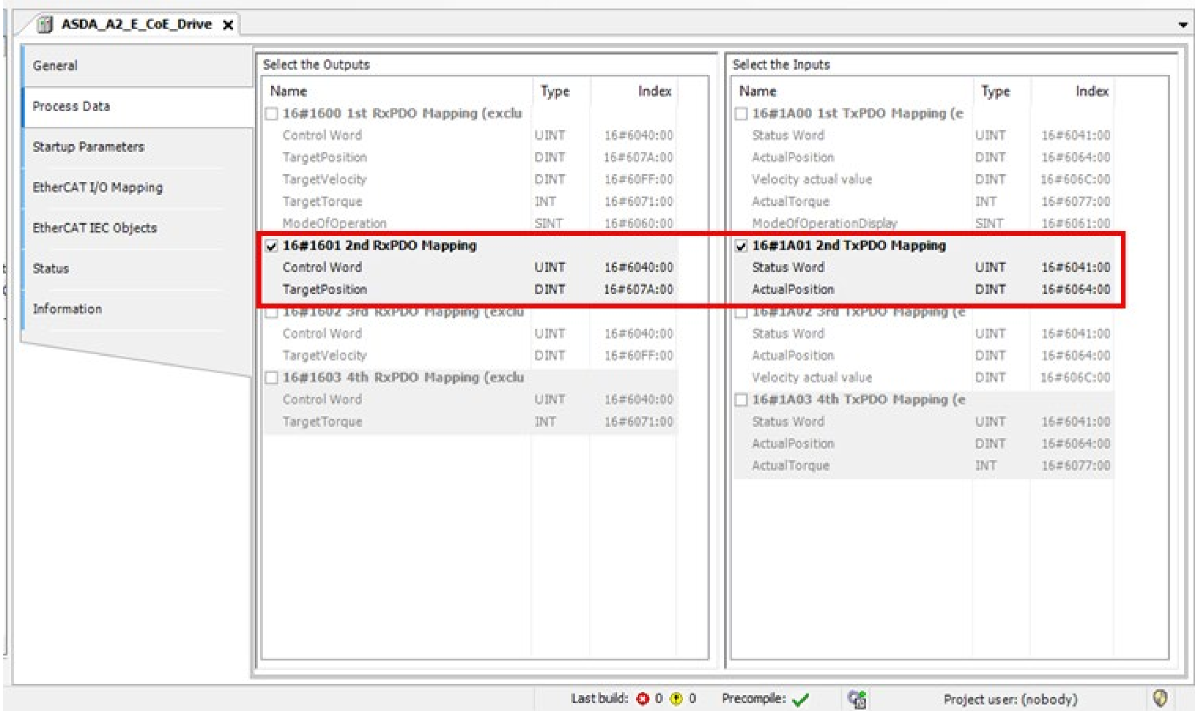


Note 1: *1 Delta ASDA-A2-E EtherCAT(CoE) Drive_SM is a synchronous axis. If a positioning axis is what you need, select Delta ASDA-A2-E EtherCAT(CoE) Drive_DML instead. After that, the operational procedures are the same for the synchronous axis and positioning axis.

- Double-click on the slave device after finishing the connection.



- Switch to “Process Data” page to configure mapping groups of PDO. The default setting for ASDA-A2 is the second group, which can operate normally with most function blocks. If additional groups or parameters of PDO need to be selected and added, please refer to content concerning function blocks description in **AX Series Motion Controller Manual**.



Select the Outputs

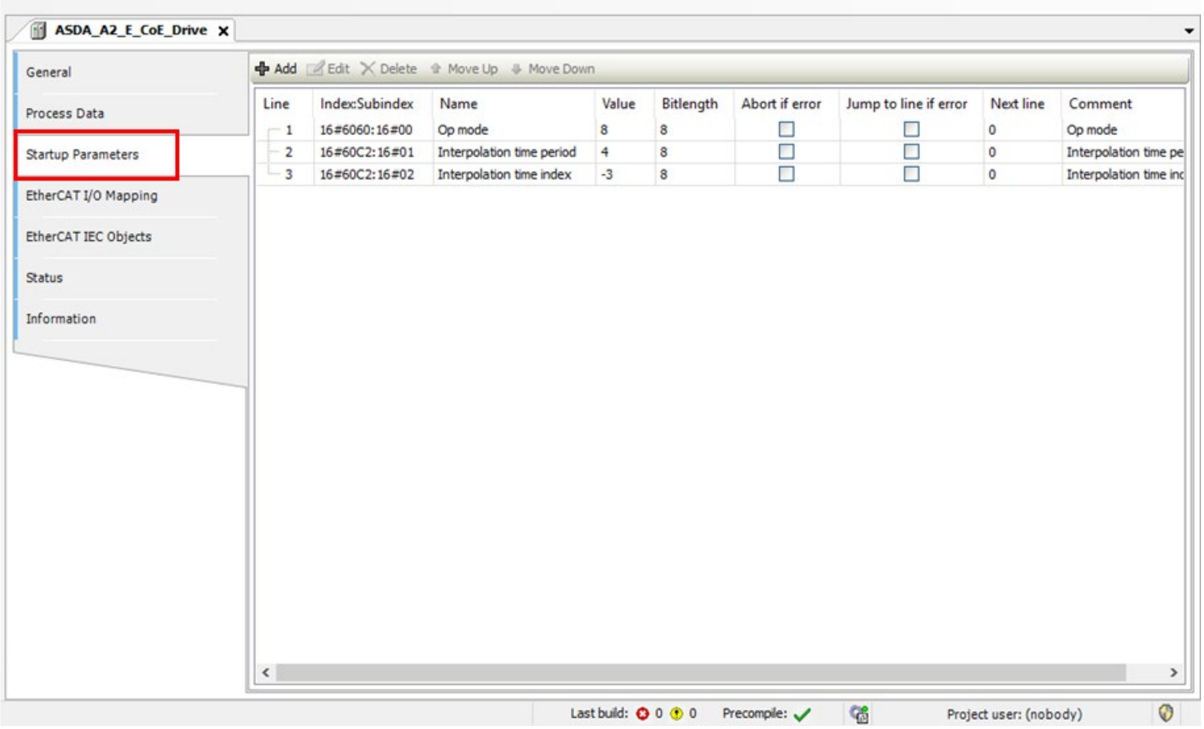
Name	Type	Index
☐ 16#1600 1st RxPDO Mapping (exclu		
Control Word	UINT	16#6040:00
TargetPosition	DINT	16#607A:00
TargetVelocity	DINT	16#60FF:00
TargetTorque	INT	16#6071:00
ModeOfOperation	SINT	16#6060:00
☑ 16#1601 2nd RxPDO Mapping		
Control Word	UINT	16#6040:00
TargetPosition	DINT	16#607A:00
☐ 16#1602 3rd RxPDO Mapping (exclu		
Control Word	UINT	16#6040:00
TargetVelocity	DINT	16#60FF:00
☐ 16#1603 4th RxPDO Mapping (exclu		
Control Word	UINT	16#6040:00
TargetTorque	INT	16#6071:00

Select the Inputs

Name	Type	Index
☐ 16#1A00 1st TxPDO Mapping (e		
Status Word	UINT	16#6041:00
ActualPosition	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
ActualTorque	INT	16#6077:00
ModeOfOperationDisplay	SINT	16#6061:00
☑ 16#1A01 2nd TxPDO Mapping		
Status Word	UINT	16#6041:00
ActualPosition	DINT	16#6064:00
☐ 16#1A02 3rd TxPDO Mapping (e		
Status Word	UINT	16#6041:00
ActualPosition	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
☐ 16#1A03 4th TxPDO Mapping (e		
Status Word	UINT	16#6041:00
ActualPosition	DINT	16#6064:00
ActualTorque	INT	16#6077:00

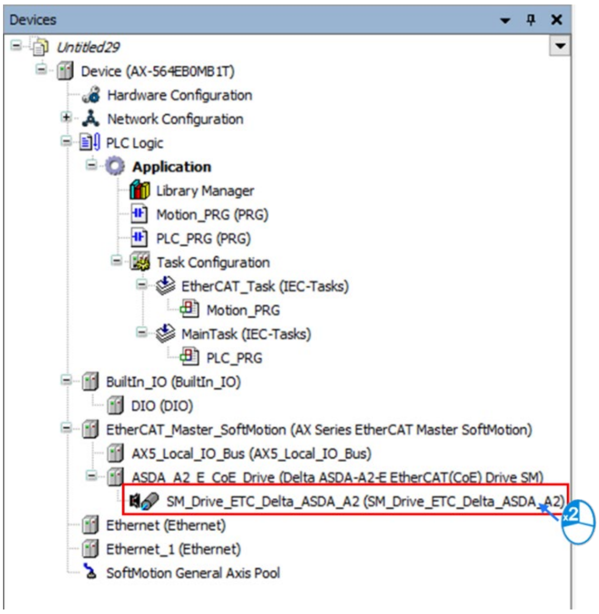
- **Initialize EtherCAT communication**

After initialization is completed, you need to input fixed values for the required Object Dictionary which can be configured on “Startup Parameters” page.



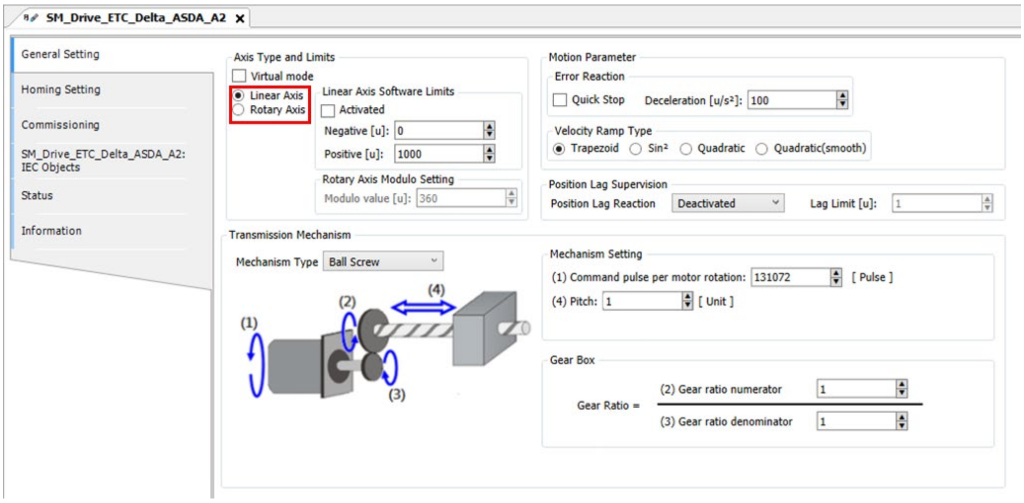
Line	Index:Subindex	Name	Value	Bitlength	Abort if error	Jump to line if error	Next line	Comment
1	16#6060:16#00	Op mode	8	8	☐	☐	0	Op mode
2	16#60C2:16#01	Interpolation time period	4	8	☐	☐	0	Interpolation time pe
3	16#60C2:16#02	Interpolation time index	-3	8	☐	☐	0	Interpolation time inc

- After finishing the settings of axis communication, double-click on “SM_Drive_ETC_Delta_ASDA_A2”.



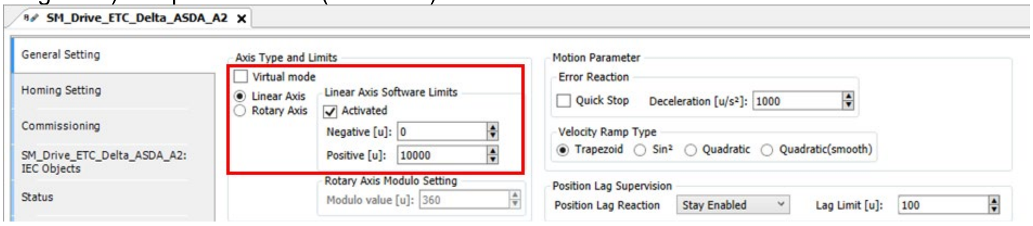
- Axis settings page

Options of axis type: “Rotary Axis” and “Linear Axis”



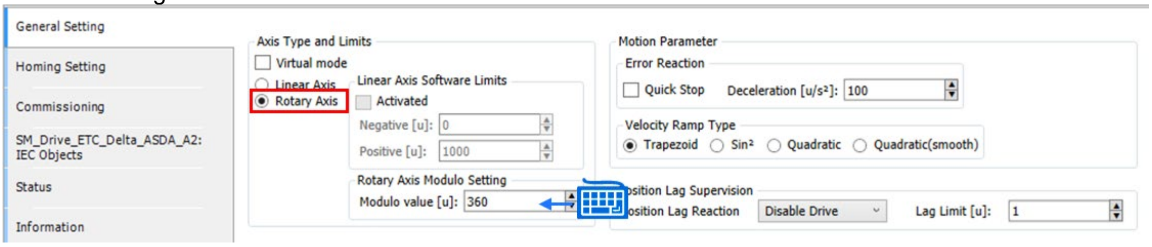
☐ Virtual mode
● Linear Axis
○ Rotary Axis
☐ Activated
Negative [u]: 0
Positive [u]: 1000
Modulo value [u]: 360
☐ Quick Stop
Deceleration [u/s²]: 100
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)
Position Lag Reaction: Deactivated
Lag Limit [u]: 1
Mechanism Type: Ball Screw
(1) Command pulse per motor rotation: 131072 [Pulse]
(4) Pitch: 1 [Unit]
(2) Gear ratio numerator: 1
(3) Gear ratio denominator: 1

- Setup Software Limits for linear axis. Click Activated to start software limit that contains negative limits ("Negative") and positive limits ("Positive").



Axis Type and Limits
☐ Virtual mode
● Linear Axis
○ Rotary Axis
Linear Axis Software Limits
☑ Activated
Negative [u]: 0
Positive [u]: 10000
Rotary Axis Modulo Setting
Modulo value [u]: 360
Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 1000
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)
Position Lag Supervision
Position Lag Reaction: Stay Enabled Lag Limit [u]: 100

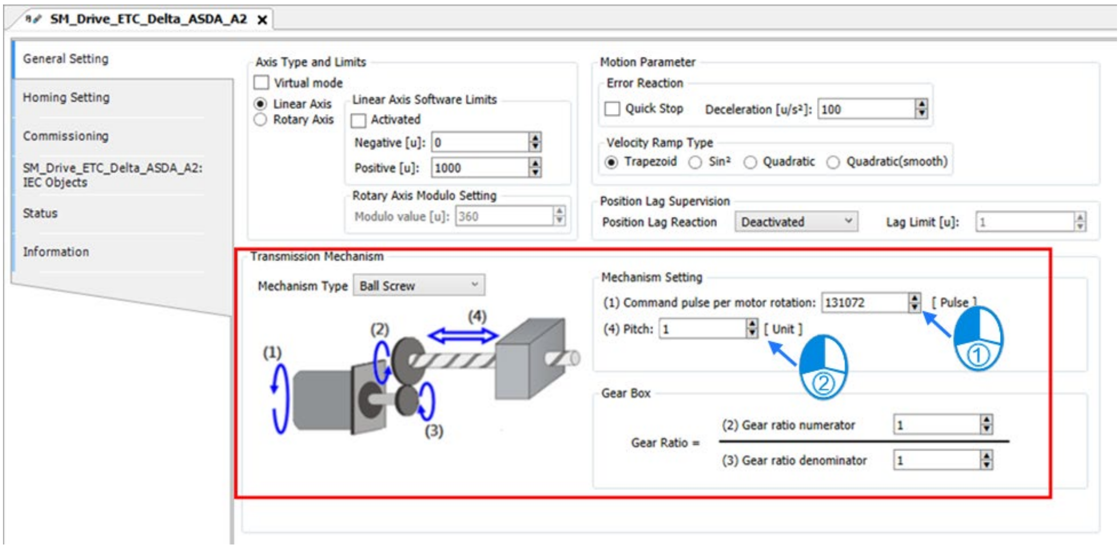
- The rotation range must be defined after finishing rotary axis settings. Please set up "Modulo value" in "Modulo settings."



Axis Type and Limits
☐ Virtual mode
○ Linear Axis
● Rotary Axis
Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 1000
Rotary Axis Modulo Setting
Modulo value [u]: 360
Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 100
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)
osition Lag Supervision
osition Lag Reaction: Disable Drive Lag Limit [u]: 1

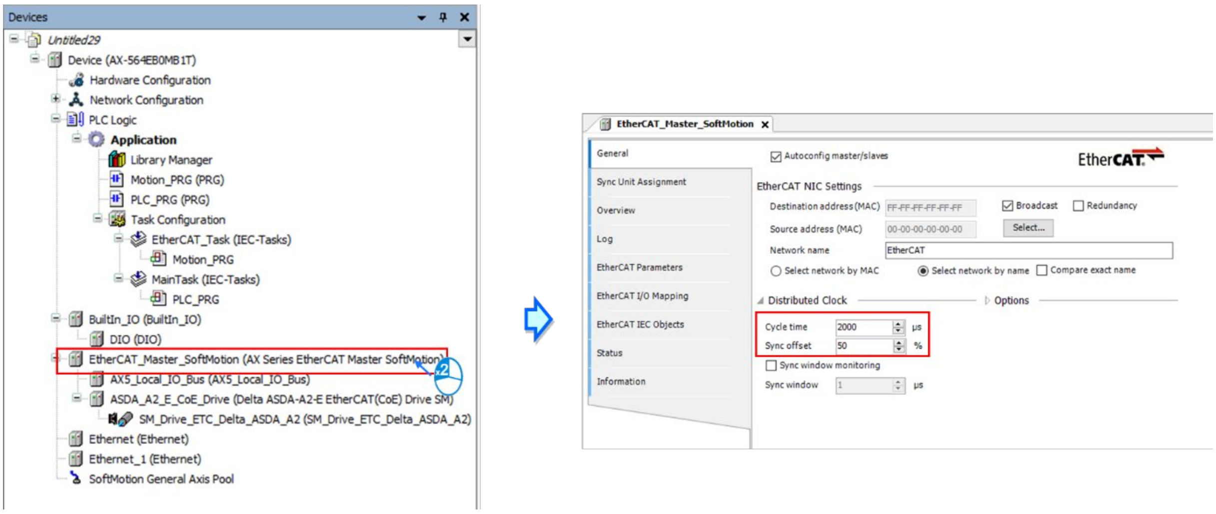
- Scaling/ Mapping page

Set the pulse number for "Command pulse per motor rotation." Set the movement distance within one full motor rotation for "Pitch."



Axis Type and Limits
☐ Virtual mode
● Linear Axis
○ Rotary Axis
Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 1000
Rotary Axis Modulo Setting
Modulo value [u]: 360
Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 100
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)
Position Lag Supervision
Position Lag Reaction: Deactivated Lag Limit [u]: 1
Transmission Mechanism
Mechanism Type: Ball Screw
Mechanism Setting
(1) Command pulse per motor rotation: 131072 [Pulse]
(4) Pitch: 1 [Unit]
Gear Box
Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

- To configure the Ethernet communication cycle time, click “EtherCAT_Master_SoftMotion,” and then set the value of “Cycle time” as 2000 and “Sync offset” as 50. (Cycle time = 2000 and Sync offset = 50 are suggested here.)



☑ Autoconfig master/slaves

Destination address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast ☐ Redundancy

Source address (MAC) 00-00-00-00-00-00

Network name EtherCAT

○ Select network by MAC ● Select network by name ☐ Compare exact name

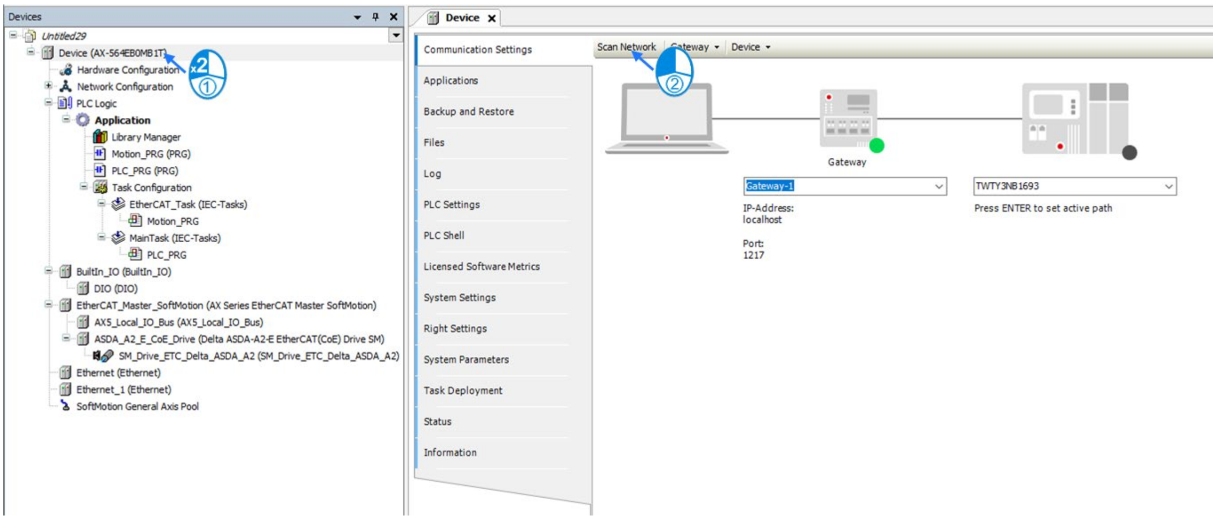
Cycle time 2000 µs

Sync offset 50 %

☐ Sync window monitoring

Sync window 1 µs

- Scan PLC controller



Gateway-1

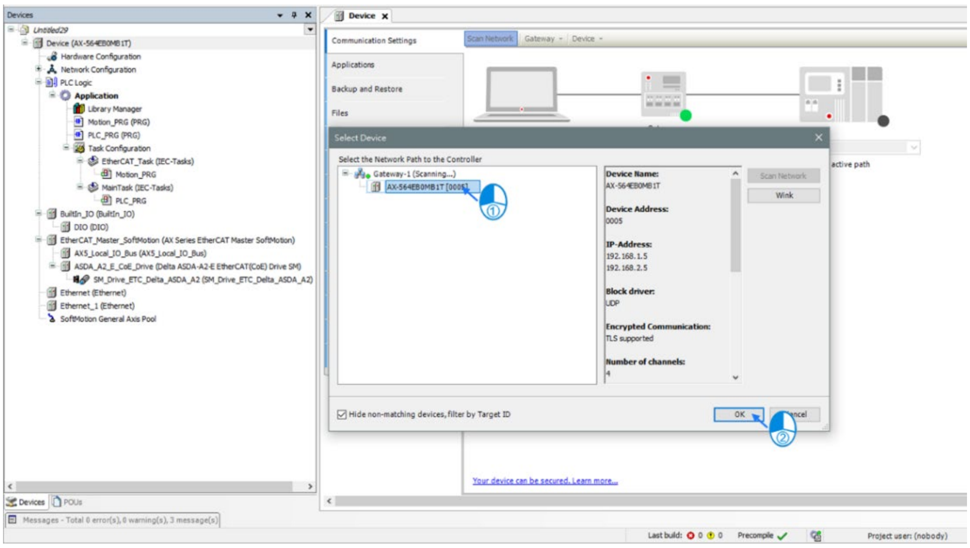
IP-Address:
localhost

Port:
1217

TWTY3NB1693

Press ENTER to set active path

- Add the newly scanned PLC controller and click “OK.”



Device Name:
AX-564EB0MB1T

Device Address:
0005

IP-Address:
192.168.1.5
192.168.2.5

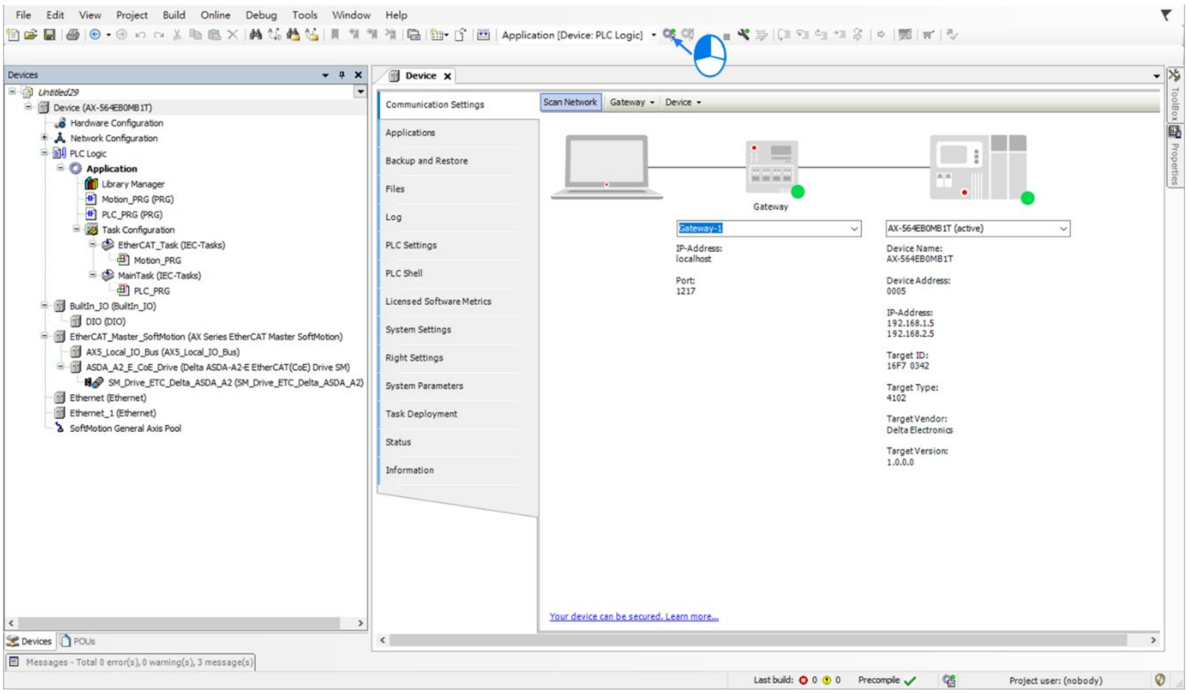
Block driver:
UDP

Encrypted Communication:
TLS supported

Number of channels:
4

☑ Hide non-matching devices, filter by Target ID

- A green light icon will be shown if the connection is successful. Then click "Login."



Gateway-1

IP-Address:
localhost

Port:
1217

AX-564EB0MB1T (active)

Device Name:
AX-564EB0MB1T

Device Address:
0005

IP-Address:
192.168.1.5
192.168.2.5

Target ID:
16F7 0342

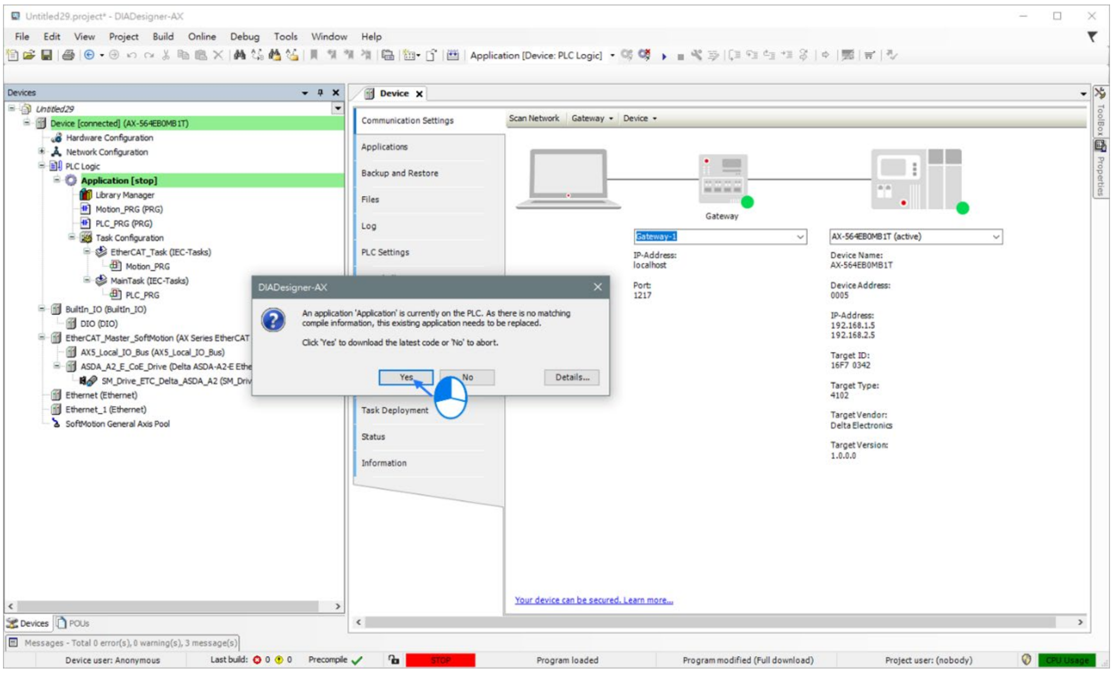
Target Type:
4102

Target Vendor:
Delta Electronics

Target Version:
1.0.0.0

Messages - Total 0 error(s), 0 warning(s), 3 message(s)

- A prompt box will pop out to remind you if you want to perform a download. Click "Yes" to continue.



An application 'Application' is currently on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

Gateway-1

IP-Address:
localhost

Port:
1217

AX-564EB0MB1T (active)

Device Name:
AX-564EB0MB1T

Device Address:
0005

IP-Address:
192.168.1.5
192.168.2.5

Target ID:
16F7 0342

Target Type:
4102

Target Vendor:
Delta Electronics

Target Version:
1.0.0.0

Messages - Total 0 error(s), 0 warning(s), 3 message(s)

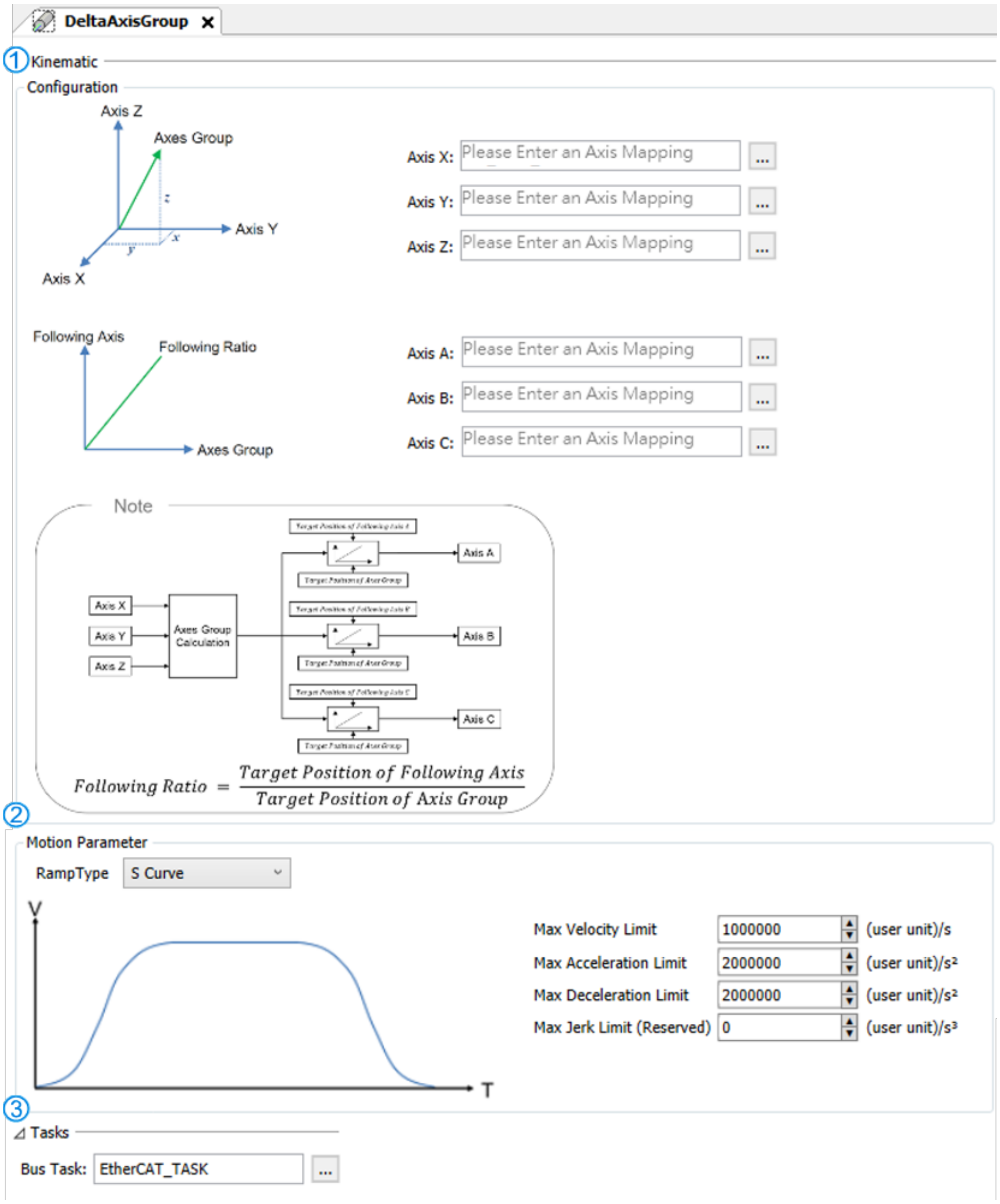
7.4.4 Axis Group Settings

Axis group movement need be applied for the linear interpolation and circular interpolation with multiple axes. DIADesigner-AX is required for grouping axes.

Maximum control axes	**Linear interpolation**	6 axes
	Circular interpolation	6 axes (3 follower axes)

7.4.4.1 Parameters for Axis Group

The parameters used for axis group movement are as follows.



Axis X: Please Enter an Axis Mapping ...

Axis Y: Please Enter an Axis Mapping ...

Axis Z: Please Enter an Axis Mapping ...

Axis A: Please Enter an Axis Mapping ...

Axis B: Please Enter an Axis Mapping ...

Axis C: Please Enter an Axis Mapping ...

$$Following\ Ratio = \frac{Target\ Position\ of\ Following\ Axis}{Target\ Position\ of\ Axis\ Group}$$

RampType: S Curve

Max Velocity Limit	1000000	(user unit)/s
Max Acceleration Limit	2000000	(user unit)/s²
Max Deceleration Limit	2000000	(user unit)/s²
Max Jerk Limit (Reserved)	0	(user unit)/s³

Bus Task: EtherCAT_TASK ...

① Kinematic

Name	Function
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 axis group
Axis B^{*1}	B axis in axis group
Axis C^{*1}	C axis in axis group

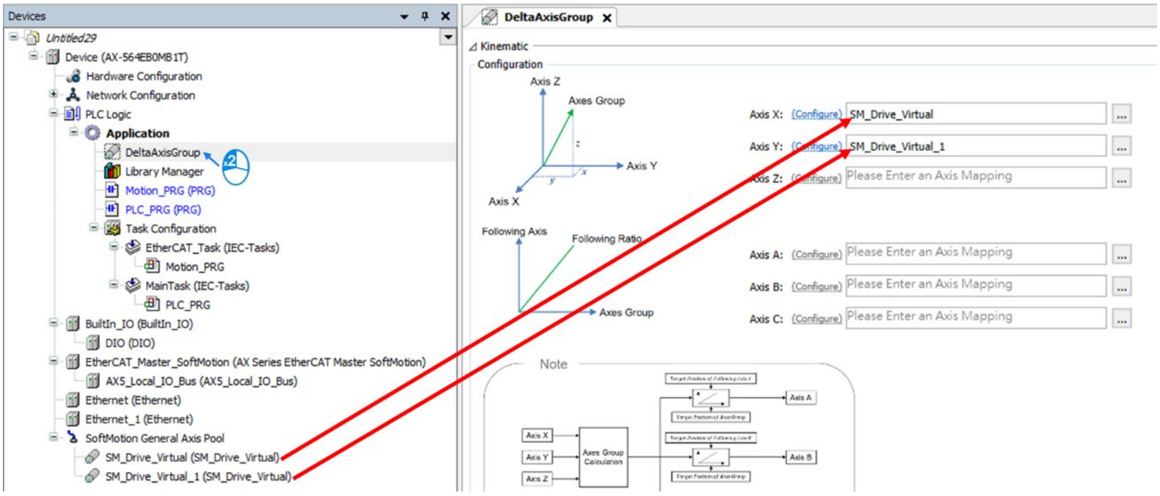
② Motion Parameter

Name	Function
Ramp Type*2	Velocity ramp type
Max Velocity Limit*3	The max velocity of axis group
Max Acceleration Limit*3	The max acceleration of axis group
Max Deceleration Limit*3	The max deceleration of axis group
Max Jerk Limit(Reserved)*3	The max jerk rate of axis group (Reserved)

③ Tasks

Name	Function
Bus Task	Show the updating task for axis groups.

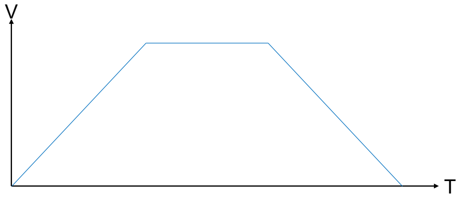
Note 1: Axis X to Axis C: Enter the names of axes individually.



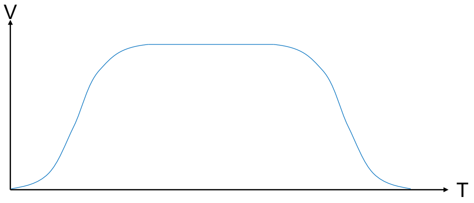
Axis X: (Configure) SM_Drive_Virtual
Axis Y: (Configure) SM_Drive_Virtual_1
Axis Z: (Configure) Please Enter an Axis Mapping
Axis A: (Configure) Please Enter an Axis Mapping
Axis B: (Configure) Please Enter an Axis Mapping
Axis C: (Configure) Please Enter an Axis Mapping

Note 2: There are two Ramp Types: Trapezoid and S-curve types, which are shown in the following figures.

- **Trapezoid**



- **S Curve**

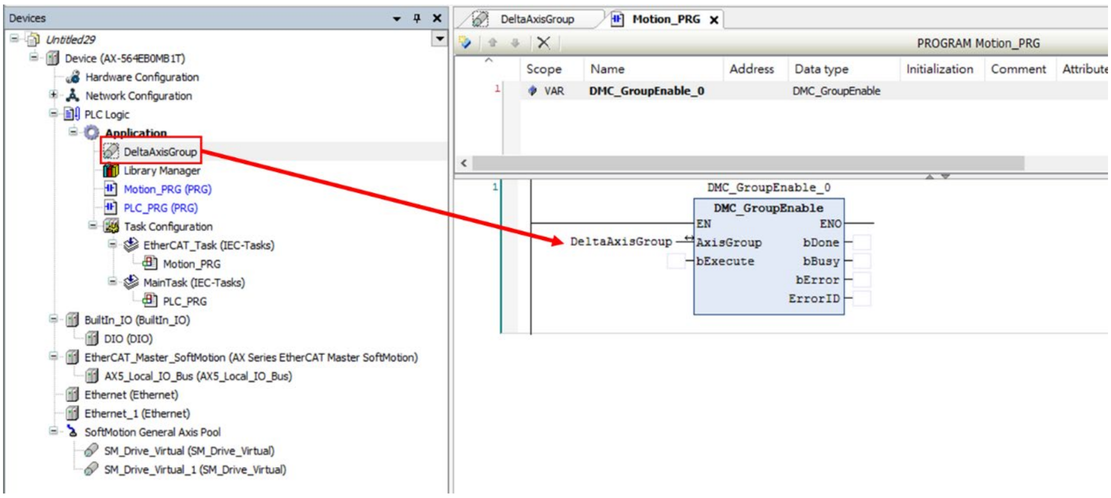


Note 3:

- Max Velocity Limit：An error occurs when the velocity exceeds the setting value.
- Max Acceleration Limit：An error occurs when the acceleration exceeds the setting value.
- Max Deceleration Limit：An error occurs when the deceleration exceeds the setting value.

7.4.4.2 Using Axis Groups in Program

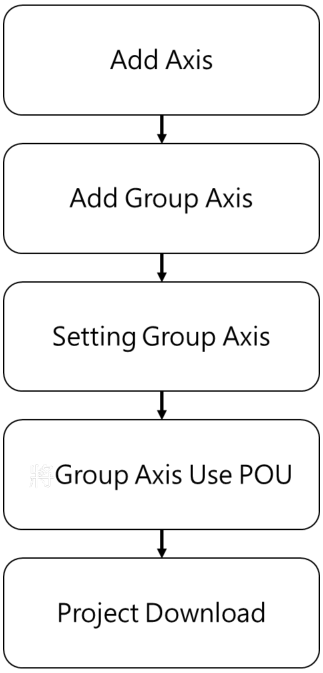
To follow the procedure, you must add the node of an axis group to the project tree and set the names of the required axes in the group individually before using axis-group function blocks. After finishing the settings, please connect the node name of the axis group to AxisGroup input of each function block.



Scope	Name	Address	Data type	Initialization	Comment	Attribute
VAR	DMC_GroupEnable_0		DMC_GroupEnable			

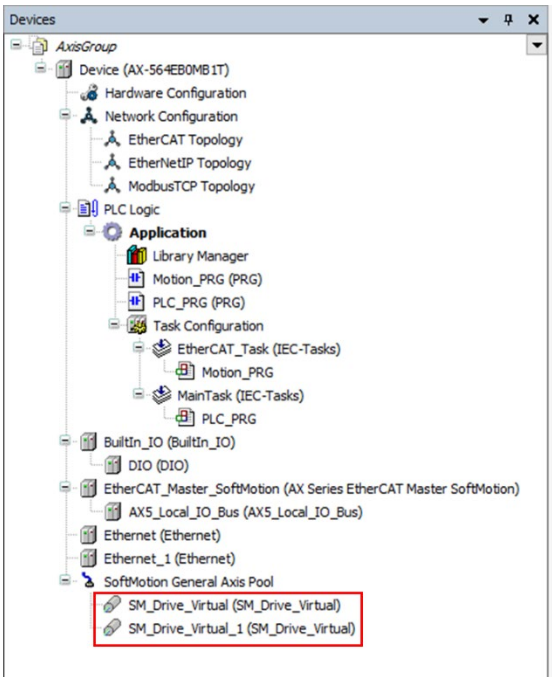
7.4.5 Procedure for Axis Group Configuration

- To use the axis group movement function, you must name the axis group and set the corresponding individual axes with DIADesigner-AX. The process flowchart of creating an axis group is shown below.



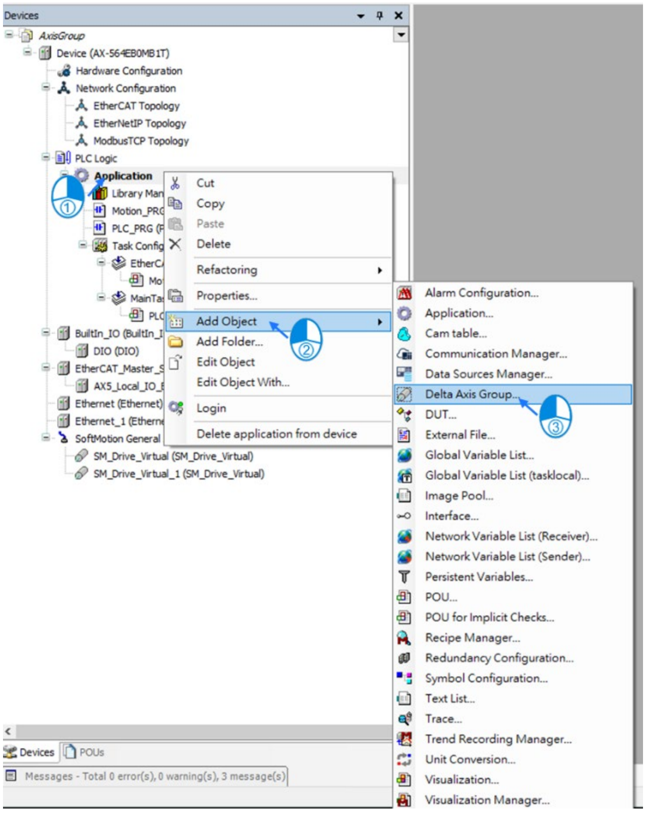
- **Procedure of creating axis groups in program**

(1) Add single axes. The following example starts from creating two virtual axes.

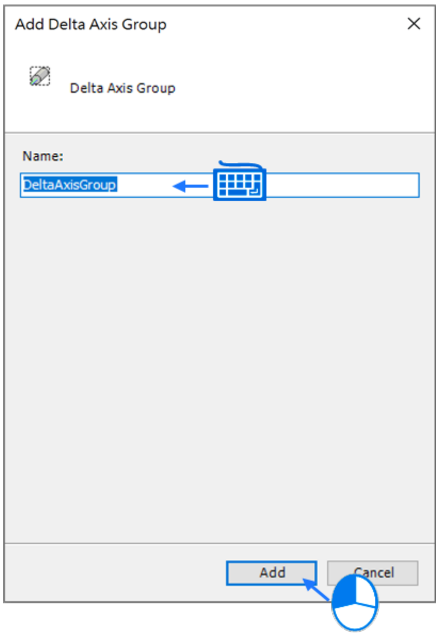


- *AxisGroup*
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
      - EtherCAT Topology
      - EtherNetIP Topology
      - ModbusTCP Topology
    - PLC Logic
      - **Application**
        - Library Manager
        - Motion_PRG (PRG)
        - PLC_PRG (PRG)
        - Task Configuration
          - EtherCAT_Task (IEC-Tasks)
            - Motion_PRG
          - MainTask (IEC-Tasks)
            - PLC_PRG
    - BuiltIn_IO (BuiltIn_IO)
      - DIO (DIO)
    - EtherCAT_Master_SoftMotion (AX Series EtherCAT Master SoftMotion)
      - AX5_Local_IO_Bus (AX5_Local_IO_Bus)
    - Ethernet (Ethernet)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool
      - SM_Drive_Virtual (SM_Drive_Virtual)
      - SM_Drive_Virtual_1 (SM_Drive_Virtual)

(2) After finishing creating axes, select “Application” and right-click “Add Object” → “Delta Axis Group”

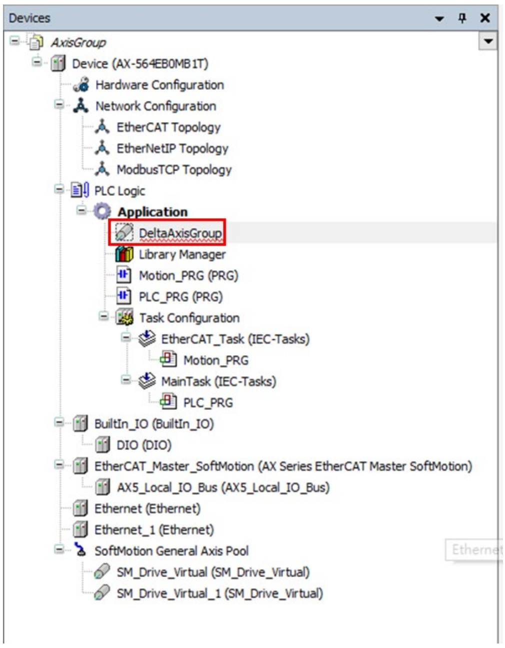


(3) Set the name for the axis group on the “Add Delta Axis Group” page, and then click “Add.”

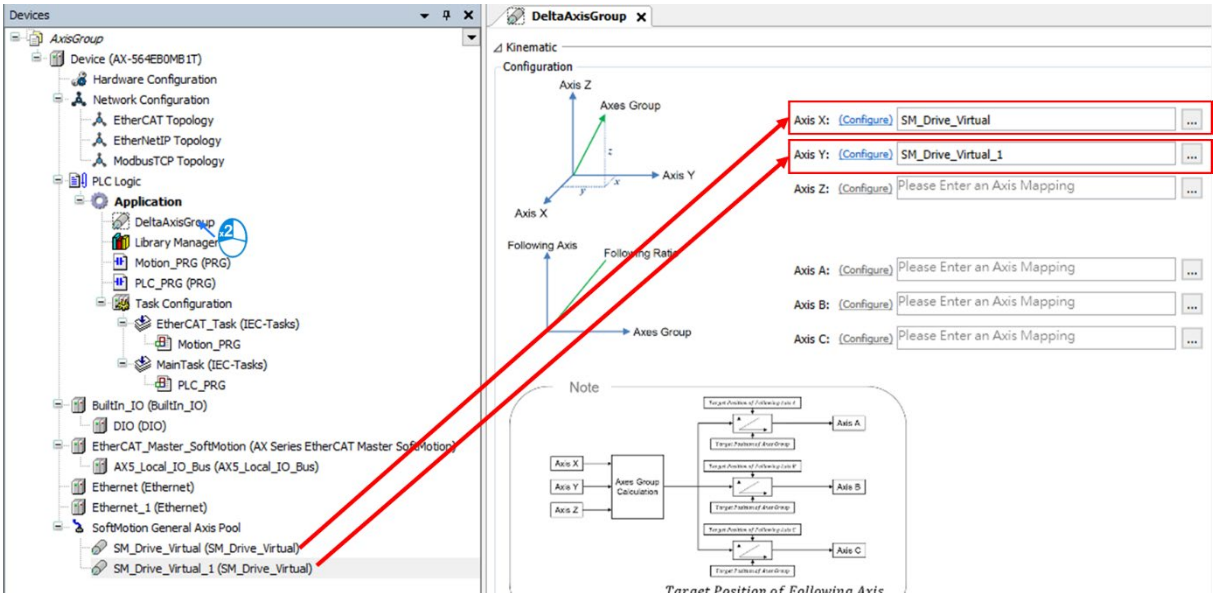


Name: DeltaAxisGroup

(4) Afterwards, “DMC_Axis_Group” will be shown on the Project tree.



(5) Click “DeltaAxisGroup,” and then enter the names of two virtual axes into the fields of “Axis X” and “Axis Y.”


Axis X: (Configure) SM_Drive_Virtual

Axis Y: (Configure) SM_Drive_Virtual_1

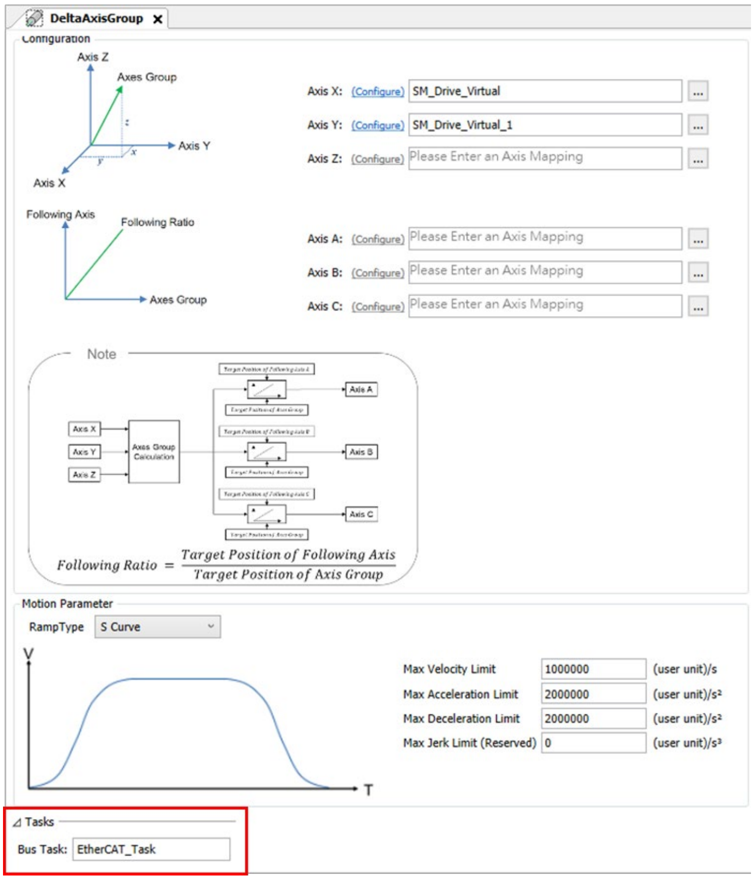
Axis Z: (Configure) Please Enter an Axis Mapping

Axis A: (Configure) Please Enter an Axis Mapping

Axis B: (Configure) Please Enter an Axis Mapping

Axis C: (Configure) Please Enter an Axis Mapping

(6) After you enter the names for the axes, the tasks in the Bus Task will be set automatically according to the tasks followed by axes.



Axis X: SM_Drive_Virtual

Axis Y: SM_Drive_Virtual_1

Axis Z: Please Enter an Axis Mapping

Axis A: Please Enter an Axis Mapping

Axis B: Please Enter an Axis Mapping

Axis C: Please Enter an Axis Mapping

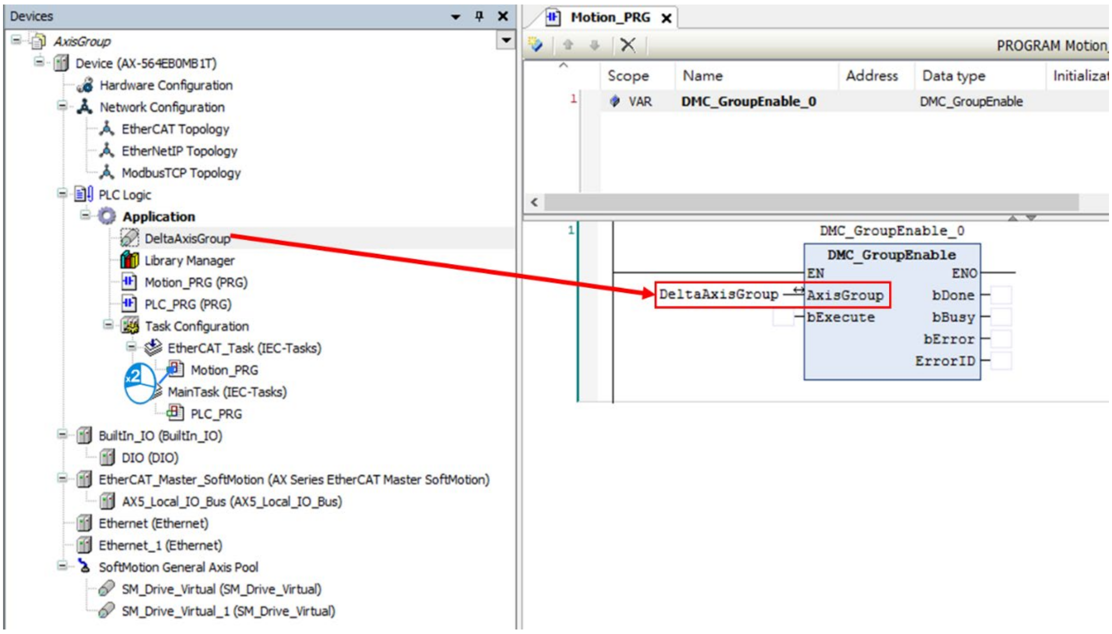
$$\text{Following Ratio} = \frac{\text{Target Position of Following Axis}}{\text{Target Position of Axis Group}}$$

RampType: S Curve

Max Velocity Limit	1000000	(user unit)/s
Max Acceleration Limit	2000000	(user unit)/s²
Max Deceleration Limit	2000000	(user unit)/s²
Max Jerk Limit (Reserved)	0	(user unit)/s³

Bus Task: EtherCAT_Task

(7) Add “DMC_GroupEnable” function block in Motion_PRG and type the axis group name in the AxisGroup input of the function block.



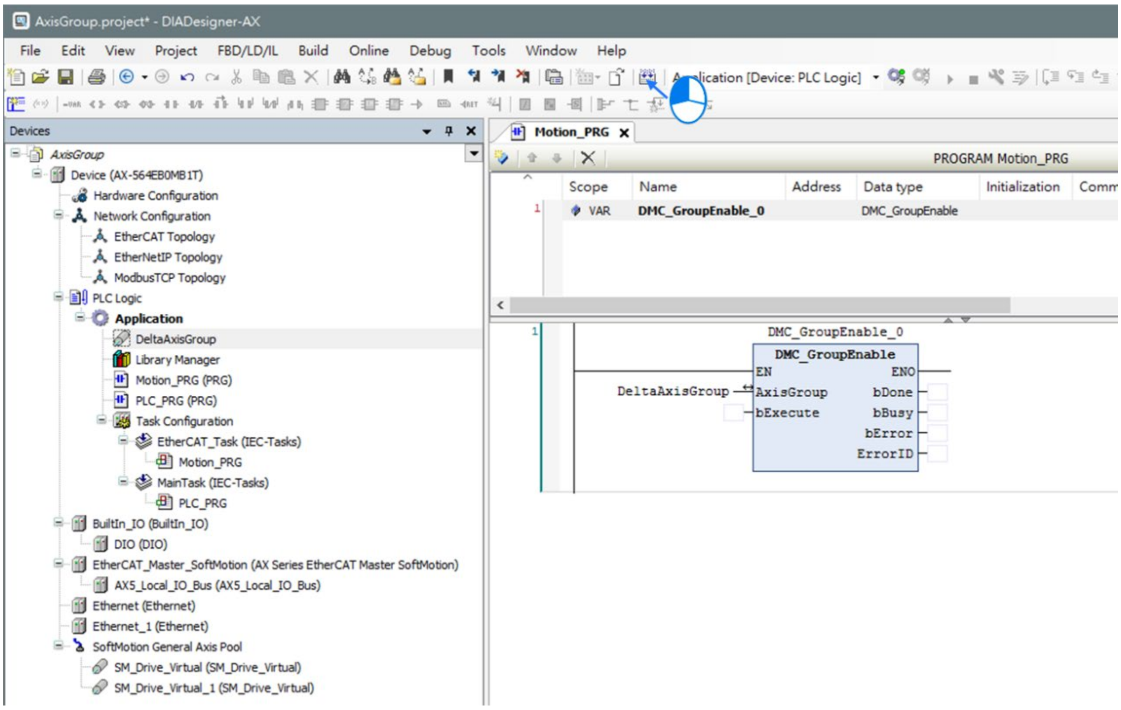
	Scope	Name	Address	Data type	Initializat
1	VAR	DMC_GroupEnable_0		DMC_GroupEnable	

DMC_GroupEnable_0

DMC_GroupEnable

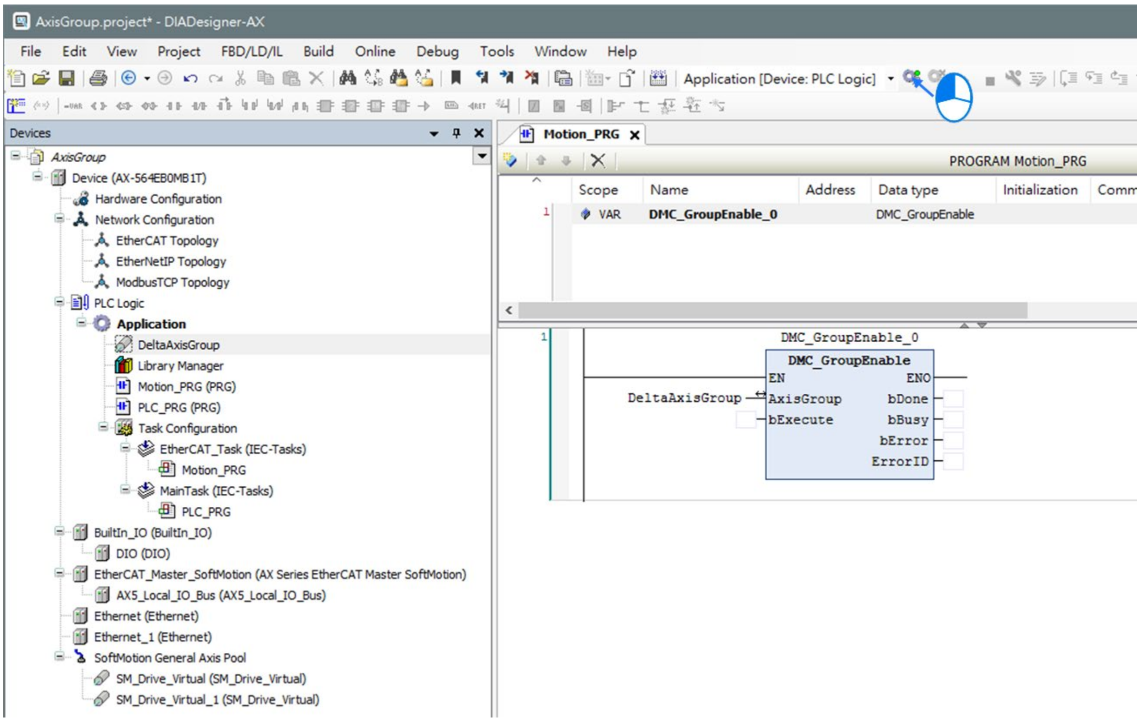
DeltaAxisGroup — AxisGroup

(8) After the program writing is completed, click the Compile button to confirm the validity.



	Scope	Name	Address	Data type	Initialization	Comm
1	VAR	DMC_GroupEnable_0		DMC_GroupEnable		

(9) After compilation, click Online Monitoring button to download the program.



	Scope	Name	Address	Data type	Initialization	Comm
1	VAR	DMC_GroupEnable_0		DMC_GroupEnable		

7.5 Motion Axis Variables

7.5.1 Variables for Single Axis

After you have created axes in the Project tree with DIADesigner_AX, the corresponding axis parameters (read-only) will be generated automatically. Axes are categorized into two types: synchronous axis (Axis_REF_SM3) and positioning axis (Axis_REF_DML). See the axis parameters described in the following table.

- **Synchronous axis (Axis_REF_SM3)**

Number	Name	Data type	Default value	Description
1000	nAxisState	SMC_AXIS_ STATE(INT)	Standstill (3)	Operating state of the current axis according to MC_ReadStatus
1012	bCommunication	BOOL	FALSE	When communication is normal, it sets to True; when communication is disconnected, it sets to False.
1014	uiDriveInterfaceError	UINT	0	When Drive Interface detects an error, Error Handling occurs.
1021	wDriveId	WORD	Driver	The number in drive nodes on the Field bus
1025	fTaskCycle	LREAL	Driver	EtherCAT cycle time of task
1031	dwErrorID	DWORD	0	Error code of Slave (Add the parameter 0x603F in PDO and use the function block MC_ReadAxisError to read the error code.)
1035	fbeFBError	ARRAY [0..g_SMC_NUMBER_FB_ERRORS] OF SMC_FBERROR	0	Axis-related error table
1040	bVirtual	BOOL	FALSE	True: virtual axis ; false: real axis
1051	iRatioTechUnitsNum	DINT	1	Change gear ratio in axis setting (denominator)
1052	dwRatioTechUnits Denom	DWORD	1	Change gear ratio in axis setting (numerator)
1060	iMovementType	INT	1	0 = Modulo 1 = Finite
1061	fPositionPeriod	LREAL	1000	Max movement distance of the rotary axis
1062	eRampType	SMC_RAMP	Trapez	Velocity ramp type: ■ Trapezoid

Number	Name	Data type	Default value	Description
		TYPE		■ sin^2 ■ Quadratic ■ Quadratic (smooth)
1100/1	fSetPosition	LREAL	0	Commanded position (User-defined unit)
1101	fActPosition	LREAL	0	Feedback position (User-defined unit)
1110,11	fSetVelocity	LREAL	0	Commanded velocity (User-defined unit /s)
1111,10	fActVelocity	LREAL	0	Feedback velocity (User-defined unit /s)
1115	bConstantVelocity	BOOL	FALSE	True: the axis is driving with a constant velocity
1120	fSetAcceleration	LREAL	0	Commanded acceleration (Unit: User-defined unit /s^2)
1125	bAccelerating	BOOL	FALSE	True when Axis is accelerating
1135	bDecelerating	BOOL	FALSE	True when Axis is decelerating
1140	fSetJerk	LREAL	0	Commanded jerk value
1160	fSetTorque	LREAL	0	Commanded torque (Nm)
1161	fActTorque*	LREAL	0	Actual torque (Nm)
1200,2	fSWLimitPositive	LREAL	0	Setting the range of the positive software limit
1201,3	fSWLimitNegative	LREAL	0	Setting the range of the positive software limit
1204	bSWEndSwitchActive	BOOL	FALSE	True when the software limit switch activated The state machine changes to ErrorStop.
1205	bSWLimitEnable	BOOL	FALSE	Software limit switches: True (Enable) /False (Disable)
-	strDriveInterfaceError	STRING	‘’	Axis error

***Note:** You need to configure parameter 0x6077 in PDO to have this parameter shown correctly.

- **Positioning Axis (Axis_REF_DML)**

Numbering	Name	Data Type	Default value	Description
1000	nAxisState	SML_AXIS_STATE	SML_AS_PowerOff(0)	Operating state of the current axis according to MC_ReadStatus
1012	bCommunication	BOOL	FALSE	When communication is normal, it sets to True; when communication is disconnected, it sets to False.
1014	uiDriveInterfaceError	UINT	0	When Driver Interface detects an error, Error Handling occurs
1051	iRatioTechUnitsNum	DINT	1	Change gear ratio in axis setting (denominator)
1052	dwRatioTechUnitsDenom	DWORD	1	Change gear ratio in axis setting (numerator)
1060	iMovementType	SML_MovementType	SML_MT_MODULO	Axis types SML_MT_MODULO = Rotary axis SML_MT_FINITE = Linear axis
1062	eRampType[*1]	SMC_RAMPTYPE	Trapez	Setting Ramp type: ■ Trapezoid ■ sin^2
1101	fActPosition	LREAL	0	Feedback position (User-defined unit)
-	strDriveInterfaceError	STRING	‘’	Axis error

***Note 1:** Only support Trapezoid and sin^2

7.5.2 Variables for Axis Group

After you have created an axis group in the project tree in DIADesigner-AX, the corresponding axis variables will be generated automatically, which are described in the following table.

Name	Data Type	Setting Value (Default Value)	Function
GroupState	DMC_ GROUP_ STATE	GroupDisabled / GroupStandby / GroupMoving / GroupHoming / GroupStopping / GroupErrorstop (GroupDisabled)	Commands for axis group status.
bError	BOOL	TRUE / FALSE (FALSE)	TRUE when an error occurs in the axis group
dwErrorId	DMC_ ERROR	DMC_ERROR (DMC_GM_NO_ ERROR)	Detailed error description
lrVelocity	LREAL	0 to 1.798E+308 (0)	Current velocity of the axis group
lrAcceleration	LREAL	Positive number, negative number or zero (0)	Current acceleration of the axis group
lrJerk	LREAL	Positive number, negative number or zero (0)	Current jerk of the axis group
bAccelerating	BOOL	TRUE / FALSE (FALSE)	TRUE when accelerating
bDecelerating	BOOL	TRUE / FALSE (FALSE)	TRUE when decelerating
bConstantVelocity	BOOL	TRUE / FALSE (FALSE)	TRUE when moving at a constant velocity (including zero velocity)
bInPosition	BOOL	TRUE / FALSE (FALSE)	TRUE when positioning is done.
bContinueDataWriten	BOOL	TRUE / FALSE (FALSE)	TRUE when the axis group is forced to stop, and the relevant data can be used by DMC_GroupContinue.
ContinuePos	ARRAY [0..5] OF LREAL	[0,0,0,0,0,0]	When the execution of DMC_GroupInterrupt is done, the position of the current axis group is recorded.
AxisX_Name*	String		Display the Axis_X name for the current axis group
AxisY_Name*	String		Display the Axis_Y name for the current axis group
AxisZ_Name*	String		Display the Axis_Z name for the current axis group
AxisA_Name*	String		Display the Axis_A name for the current axis group
AxisB_Name*	String		Display the Axis_B name for the

Name	Data Type	Setting Value (Default Value)	Function
			current axis group
AxisC_Name[*1]	String		Display the Axis_C name for the current axis group
RampType	DMC_GROUP_RAMP_TYPE	Trapezoid / S Curve (S Curve)	Ramp type of current S-curve
lrMaxVelocityLimit	LREAL	Positive number or zero (1000000)	The maximum velocity of the axis group
lrMaxAcceleration Limit	LREAL	Positive number or zero (2000000)	The maximum acceleration of the axis group
lrMaxDecelerationLimit	LREAL	Positive number or zero (2000000)	The maximum deceleration of the axis group
lrMaxJerkLimit (Reserved)	LREAL	Positive number or zero (0)	The maximum jerk of the axis group (Reserved)
PlanningPriority[*2, 3]	DMC_GROUP_PLANNING_PRIORITY	Velocity / Acceleration (Acceleration)	Planning the path: Velocity: Acceleration or deceleration can be ignored in order to meet the condition of velocity. Acceleration: Velocity can be ignored in order to meet the condition of acceleration / deceleration.
bVelocityWarning	BOOL	TRUE / FALSE (FALSE)	When the velocity of the axis group exceeds the proportionality, the state is TRUE.
bAccelerationWarning	BOOL	TRUE / FALSE (FALSE)	When the acceleration of the axis group exceeds the proportionality, the state is TRUE.
bDecelerationWarning	BOOL	TRUE / FALSE (FALSE)	When the deceleration of the axis group exceeds the proportionality, the state is TRUE.
bJerkWarning (Reserved)	BOOL	TRUE / FALSE (FALSE)	When the jerk of the axis group exceeds the proportionality, the state is TRUE.
StopMethod	Enum of BYTE	Immediate Stop / MaxGroupDecStop / MaxAxisDecStop (Immediate Stop)	Set the stop method when an error occurs to the axis group or in the middle of motions
lrVelocityWarning Percentage	LREAL	0 to 1 (0)	Set the proportionality of the maximum velocity of the axis group for the warning to start. Once the set percentage is reached, the warning starts. Set the value to 0 to stop the warning.
lrAccelerationWarning Percentage	LREAL	0 to 1 (0)	Set the proportionality of the maximum acceleration of the axis group for the warning to start. Once the set percentage is reached, the warning starts. Set the value to 0 to stop the warning.

Name	Data Type	Setting Value (Default Value)	Function
lrDecelerationWarning Percentage	LREAL	0 to 1 (0)	Set the proportionality of the maximum deceleration of the axis group for the warning to start. Once the set percentage is reached, the warning starts. Set the value to 0 to stop the warning.
lrJerkWarning Percentage (Reserved)	LREAL	0 to 1 (0)	Set the proportionality of the maximum jerk of the axis group for the warning to start. Once the set percentage is reached, the warning starts. Set the value to 0 to stop the warning.
Radius Correction	LREAL	0 to 100 (0, 1)	This is to set the tolerance for setting the radius when circular interpolation is selected in the function block of DMC_MoveCircularRelative.AuxPoint. Tolerance % = the distance between the center point and the bisection of the starting and ending points to be divided by the radius.
bVelocityWarning	BOOL	TRUE / FALSE (FALSE)	TRUE when the velocity of the axis group exceeds the value set in the lrVelocityWarning Percentage.
bAccelerationWarning	BOOL	TRUE / FALSE (FALSE)	TRUE when the acceleration of the axis group exceeds the value set in the lrAccelerationWarningPercentage.
bDecelerationWarning	BOOL	TRUE / FALSE (FALSE)	TRUE when the deceleration of the axis group exceeds the value set in the lrDecelerationWarningPercentage.
bJerkWarning (Reserved)	BOOL	TRUE / FALSE (FALSE)	TRUE when the jerk of the axis group exceeds the value set in the lrDecelerationWarningPercentage.
StopMethod	Enum of BYTE	Immediate Stop / MaxGroupDecStop / MaxAxisDecStop (Immediate Stop)	Set the stop method for the axis group when errors occur or when it is time to stop the movement.

Notes:

1. When the rotary axis type is selected, the range of motion can NOT exceed the value set in modulo, otherwise, an error message “Axis limit violated” appears.
2. For the DL_MotionControl library of V1.3.3.0 or later versions, the display of PlanningPriority is supported.
3. For the DL_MotionControl library of V1.3.5.0 or later versions, the default value is Acceleration.

7.6 Motion Control Programming

7.6.1 Motion Control Program

Before programming in DIADesigner-AX, please take the following descriptions as reference.

7.6.1.1 Program Architecture and Types in DIADesigner-AX

In the classic architecture, a source code for a PLC is composed of procedures including subroutines. When the size of a program becomes larger, maintenance and debugging also become a huge burden. Under the IEC 61131-3 architecture, a program is divided into several units according to the functions or characteristics, which makes developing and maintaining much easier. Since POUs are modularized, different POUs can be developed by different designers to enhance distribution of professional manpower and project execution.

There are three types of POUs: program (PROG), function block (FB) and function (FC).

■ Program (PROG):

The program type plays a major process role in a PLC program. The execution is assigned by Task which includes specific scan cycle or interrupt subroutines and provides scan order arrangement for programs in the Task list. Besides, a POU of the program type can call a function block (FB).

■ Function block (FB):

A static symbol can be declared in a function block (FB). As a result, the value of the symbol after an operation can be retained. Because the operation is performed on the value memorized in the function block and an input value, the output values may be different even if the input values are the same.

Besides, a function block can call another function block. The function block (FB) type is similar to subroutines. The FB process requires suitable parameters and can only execute once called by a program.

■ Function (FC):

Function (FC) is used to return operation results. Contrary to FBs, it has no memory and can only return a single value. Since a Function (FC) does not have any memory of its own, it cannot call a function block but a function.

➨ Tasks

Each program POU needs to assign a Task that determines the order for program execution or start.

The programming structure characteristic of IEC 61131-3 is that a program can be divided into several independent POUs. When POUs are compiled, they are rearranged and combined into an execution code for scanning. The new combination order of POUs is based on the assigned Tasks.

Below are types of tasks:

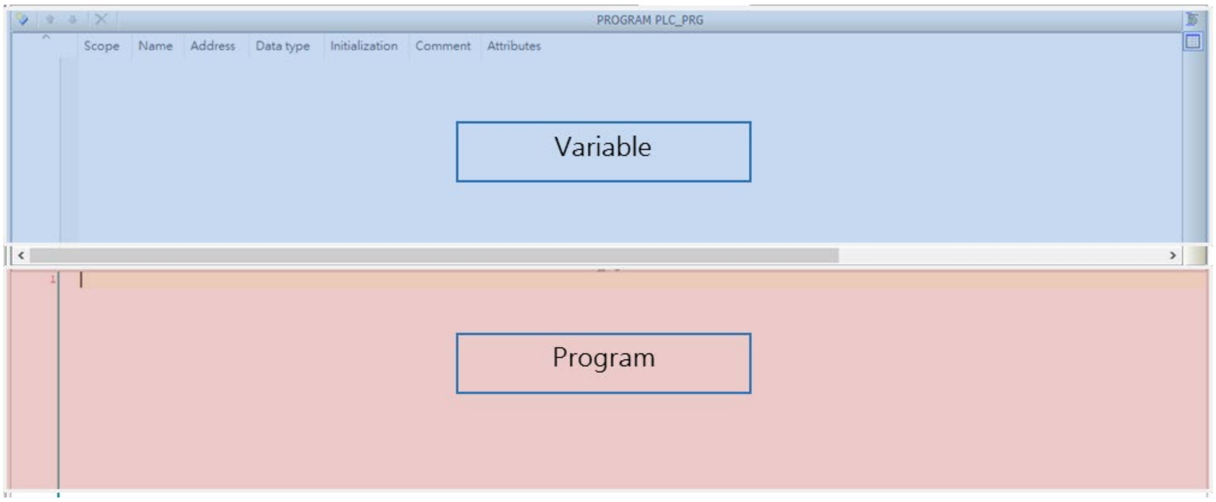
- Cyclic: Assigned POU sets interval time for per scan.
- Event: When Bool variable is set from False to True, a scan execution is performed.
- External: When external triggers to send a signal, a corresponding POU is executed.
- Freewheeling: Assigned POU performs scan automatically in a continuous loop when the previous scan has been completed.
- Status: When Bool variable is set from False to True, the scan is executed for a cycle.

Please refer to section 4.4.1 for the details of task operating process.

7.6.1.2 POU in DIADesigner-AX

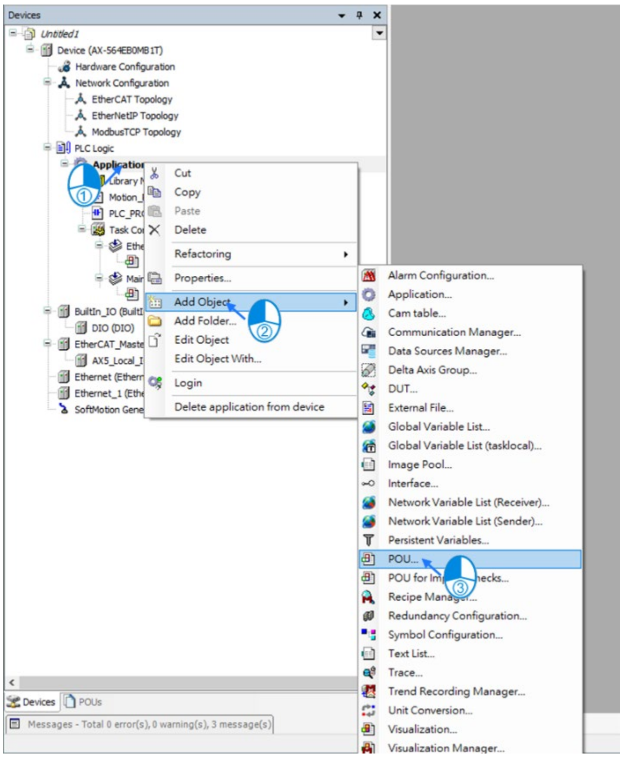
All POUs created by you are listed in the project management area with programs and function blocks managed separately. In addition, the icon of POU may vary based on different program and function block programming languages which also includes information beside the POU name.

Double-click the POU in the project management area for editing. The POU editing section is composed of two parts. The upper part of the editing section is the symbol table of local variables, while the lower part is the main part of the program. Also, the editing environment at the lower part of the editing section is different for different programming languages. For more information on symbol tables and programming, please refer to the following sections.

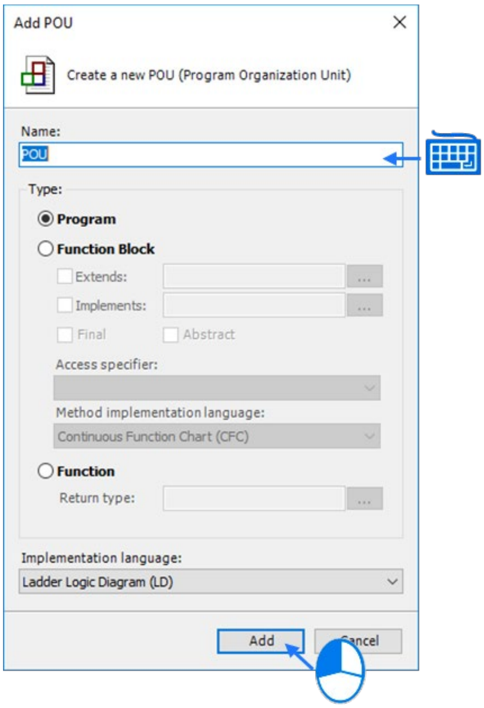


7.6.1.3 Adding POU in DIADesigner-AX

- Open an existing project in DIADesigner-AX, right-click “Application” to select “Add Object,” and then choose “POU.”



- Type a POU name in Name field. For the Implementation language, select a programming language, and then click “Add.”



Add POU

Create a new POU (Program Organization Unit)

Name: POU

Type:

● Program

○ Function Block

☐ Extends:

☐ Implements:

☐ Final ☐ Abstract

Access specifier:

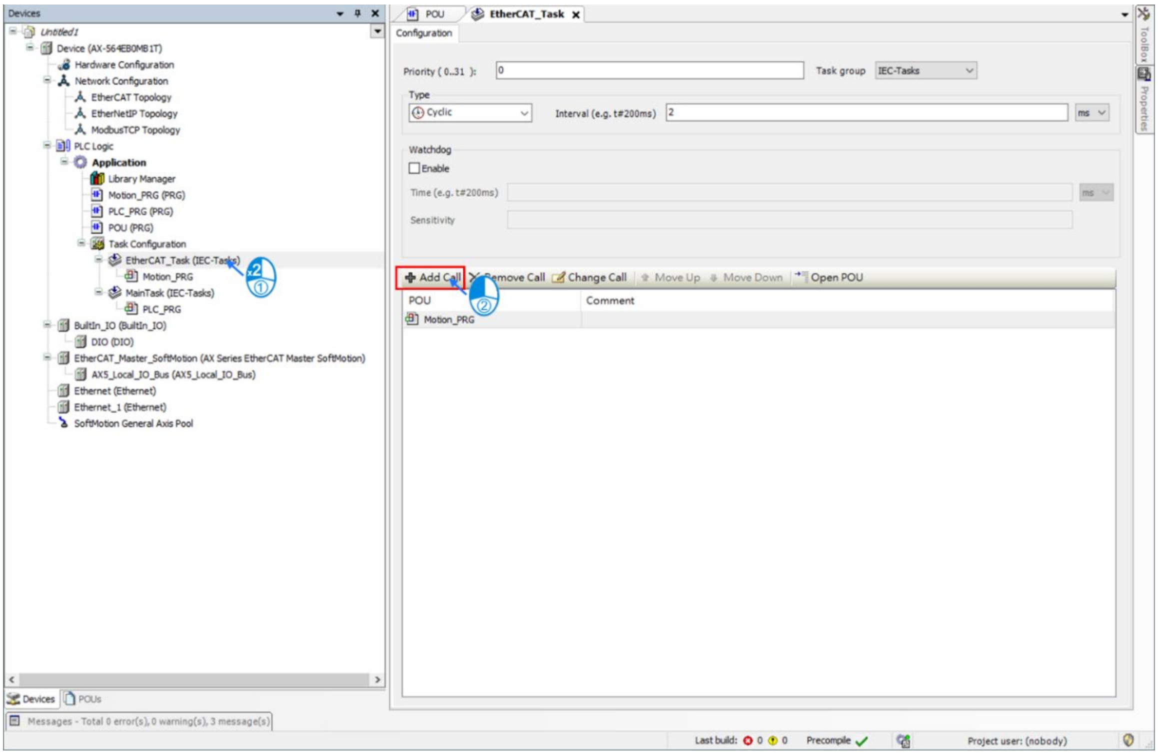
Method implementation language: Continuous Function Chart (CFC)

○ Function

Return type:

Implementation language: Ladder Logic Diagram (LD)

- The POU appears in the left column. Double-click on “EtherCAT_Task” and choose “Add Call.”



Priority (0..31): 0

Task group: IEC-Tasks

Type: Cyclic

Interval (e.g. t#200ms): 2 ms

Watchdog

☐ Enable

Time (e.g. t#200ms)

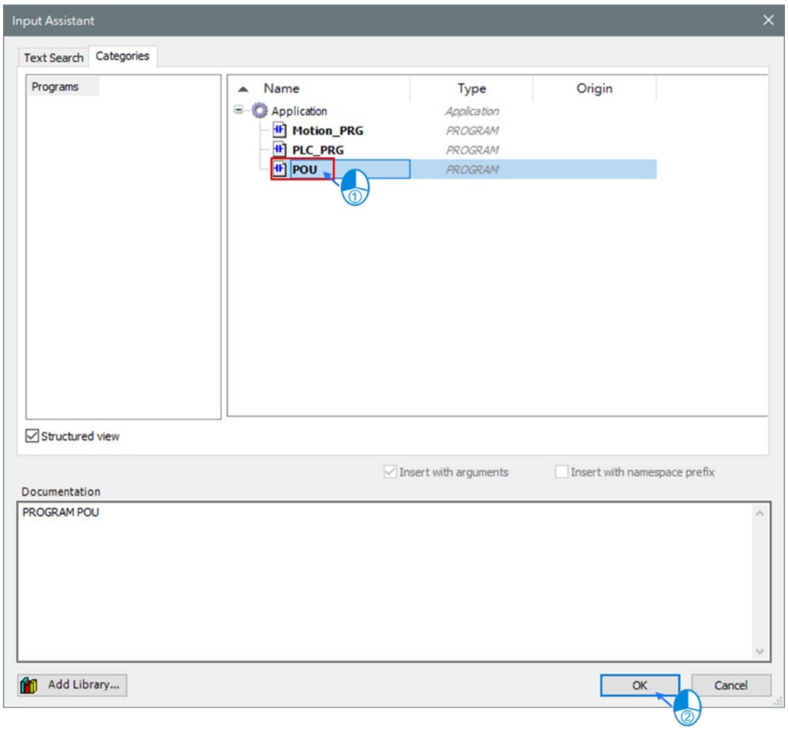
Sensitivity

POU	Comment
Motion_PRG	

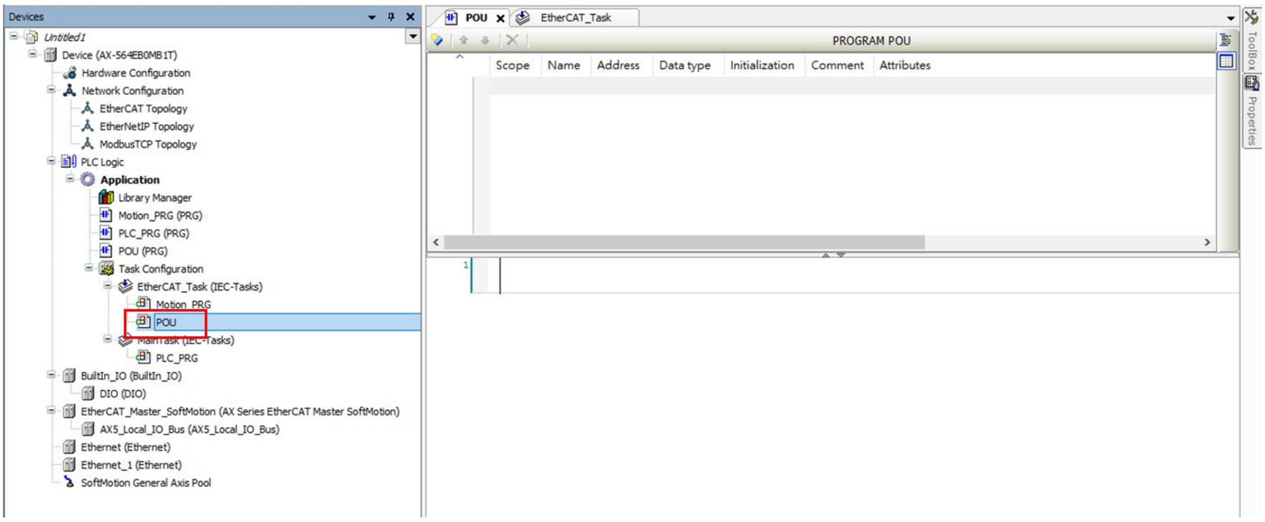
Messages - Total 0 error(s), 0 warning(s), 3 message(s)

Last build: 0 0 Precompile ✓ Project user: (nobody)

- Select the created POU and click "OK."



- Choose the POU under EtherCAT_Task item to edit the program.



7.6.1.4 PDO Mapping

Before using motion control instructions, the communication of PDO (Process Data Objects) mapping between the software DIADesigner-AX and AX motion CPU must be set up first.

PDO mapping settings:

RxPDO (1600 hex)	Control Word (6040 hex), TargetPosition (607A hex)
TxPDO (1A00 hex)	Status Word (6041 hex), ActualPosition (6064 hex)

The table above shows the default PDO mapping parameters for ASDA-A2-E.

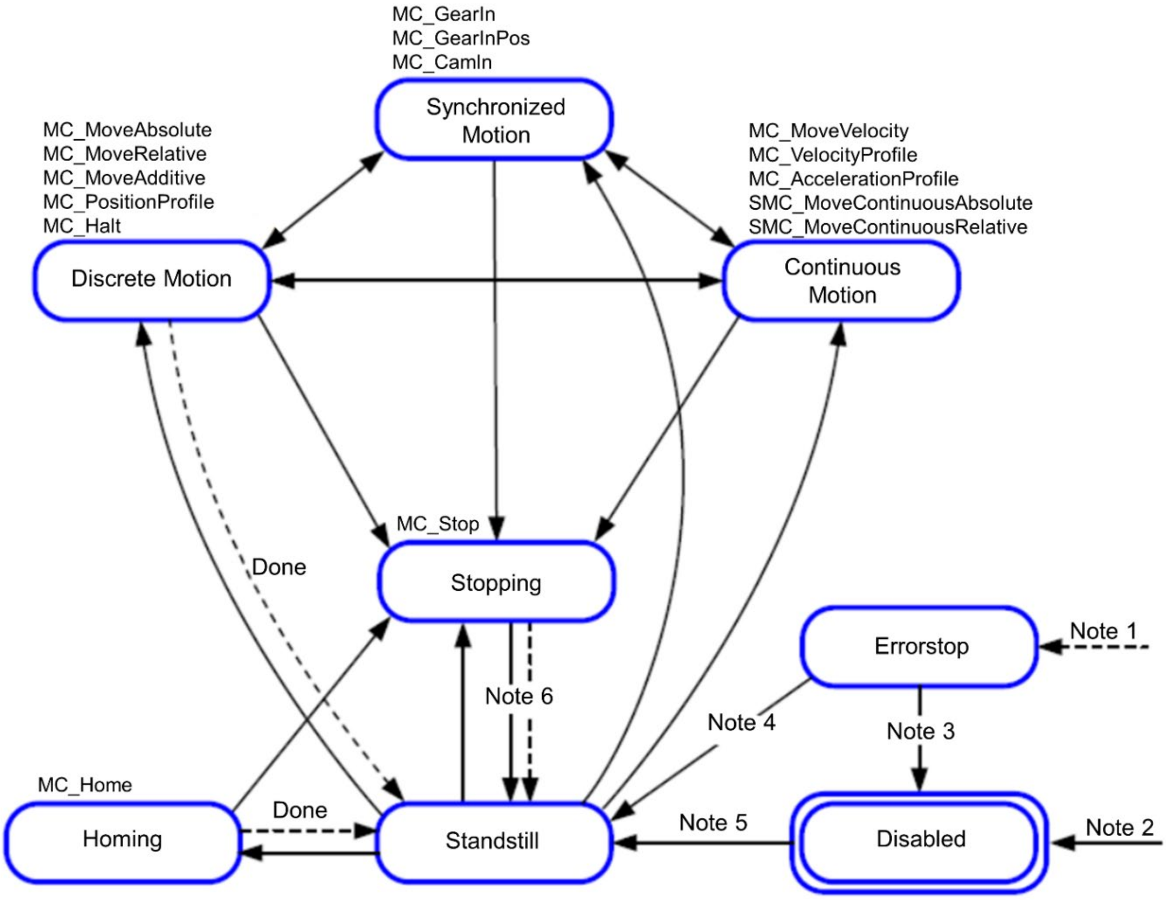
Please refer to **AX Series Motion Controller Manual** for the PDO parameters required by the related motion function blocks.

7.6.2 Axis State Transitions

This section introduces single axis state transitions and multi-axis state transitions in axis groups for multiple function block use. The transition rules fulfill PLCopen motion control standard.

7.6.2.1 Axis State

- Synchronous Axis



Note 1: When any error occurs in the axis, it enters “ErrorStop”.

Note 2: MC_Power.Enable = FALSE and there is no error in the axis

Note 3: MC_Reset and MC_Power.Status = FALSE

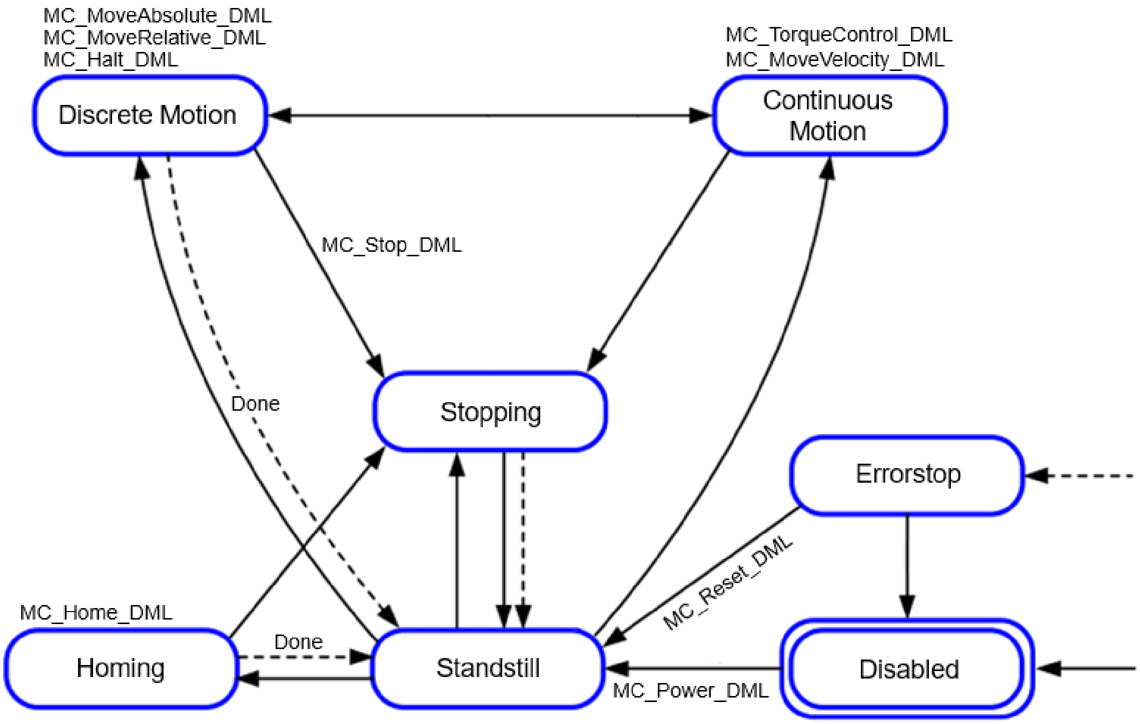
Note 4: MC_Reset and MC_Power.Status = TRUE and MC_Power.Enable = TRUE

Note 5: MC_Power.Enable = TRUE and MC_Power.Status = TRUE

Note 6: MC_Stop.Done = TRUE and MC_Stop.Execute = FALSE

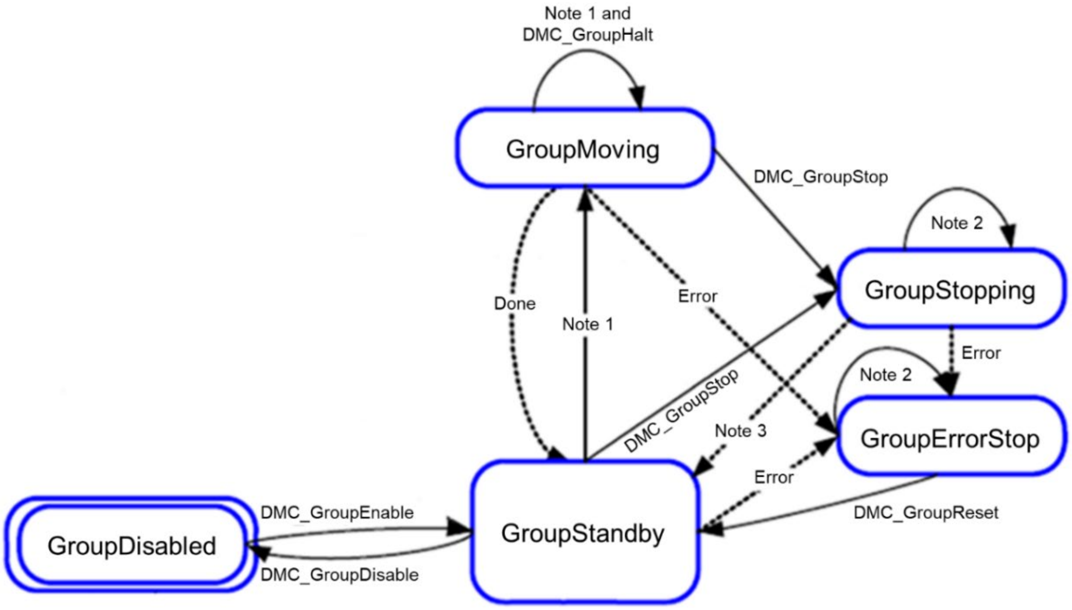
State	Meaning
Disabled	The axis is in servo OFF state.
Standstill	The axis is in servo ON state.
Discrete Motion	The state would be Discrete Motion while single-axis motion instructions are being executed.
Continuous Motion	The state would be Continuous Motion while single-axis continuous motion instructions are being executed.
Synchronized Motion	Execute the synchronization instructions to have the axis in the state of Synchronized Motion; this includes waiting for the execution of synchronization instructions.
Stopping	After the execution of MC_Stop instruction is done, the Active switches to True. No single-axis motion instuctions are allowed in this state.
ErrorStop	An error occurs in the single axis motion and the state of the axis is ErrorStop. No single-axis motion instuctions are allowed in this state.
Homing	The state would be Homing while MC_Home or MC_HomeWithParameter instruction is being executed for a single axis.

- Positioning Axis



State	Meaning
Disabled	Axis during servo OFF.
Standstill	Axis during servo ON.
Discrete Motion	The state would be Discrete Motion while single-axis motion instructions are being executed.
Continuous Motion	The state would be Continuous Motion while single-axis continuous motion instructions are being executed.
Stopping	After the execution of MC_Stop instruction is done, the Active switches to True. No single-axis motion instuctions are allowed in this state.
ErrorStop	An error occurs in the single-axis motion and the state of the axis is ErrorStop. No single-axis motion instuctions are allowed in this state.
Homing	The state would be Homing while MC_Home or MC_HomeWithParameter instruction is being executed for a single axis.

7.6.2.2 Axis Group State



Note 1: Applicable to all function blocks of group moving, non-administrative.

Note 2: All motion function blocks are able to be executed when the group state is GroupErrorStop or GroupStopping, but the axis group will not act.

Note 3: When DMC_GroupStop is Done or MC_GroupStop is not executed.

Note 4: The state GroupDisabled can only be switched to from GroupStandby state , or an error will occur when DMC_GroupDisable is executed in the non-GroupStandby state.

Status	Definition
GroupDisabled	Execute MC_GroupDisable and switch the axis group to GroupDisabled.
GroupStandby	No motion instructions have been executed and the state of the axis group is GroupStandby.
GroupMoving	A axis-group positioning instruction is being executed and the state of the axis group is GroupMoving.Moving.
GroupStopping	When Active of MC_GroupSto is True, the state of the axis group is GroupStopping. No motion instructions can be executed under this state.
GroupErrorStop	The axis group will enter GroupErrorStop state once an error occurs.

- Interaction between single-axis state and axis-group state

  (1) If one of the axes in the group is in ErrorStop and the axis group is not in GroupDisabled, the group would be in GroupErrorStop status.

  (2) When state GroupMoving/GroupStopping/GroupHoming disconnect the power of an axis, the axis group would be in GroupErrorStop state.

  (3) If all axes are in Standstill, the axis group can be in state GroupStandby, GroupDisabled or GroupErrorStop.

  (4) If the motion of a single axis interrupts the motion of an axis group, the other axes in the group should be stopped and enter state Stopping, while the state of the axis group enters state GroupStandby.

  (5) In case that the axis group is in GroupStandby, there’s no need for all the single axes being in state SynchronizedMotion.

  (6) For axis group motion instructions (including MC_GroupStop), all single axes in the axis group should be in state Synchronized Motion.

  (7) When an error occurs during the movement of an axis group, all axis in the group should stop immediately till the axis group enters state GroupErrorStop. Those single axes with no errors will enter state Standstill.

  (8) When the state of an axis group is GroupErrorStop, the state of single axes will not be affected.

7.6.3 Execution and Status Indication for Motion Control Instructions

The motion function blocks are grouped under two main categories with AX series motion controllers:

Category	Description
MC_	PLCopen motion control function blocks
DMC_	Delta self-defined function blocks*

***Note:** Delta self-defined function blocks (DMC) include motion control type and other administrative/ non-administrative type applicable for AX series motion CPU.

General pins for motion control function blocks include input, output and in-out. The section explains the meanings and behaviors of these pins. For more details concerning motion function blocks, please refer to **AX Series Motion Controller Manual.**

7.6.3.1 Basic Rules of Executing Instructions

● Defining input and output pins

Common inputs and outputs in motion control function blocks are listed below. Usually, a function block consists of at least one or a part of the input/output pins listed below. For example, a function block contains either Execute or Enable input pin based on the properties of the motion control function block.

Inputs			
Name	Description	Date Type	Setting value (Default)
En	Receiving the logic status in front of the instruction	BOOL	True/False (False)
Enable	Enabling a motion control function block	BOOL	True/False (False)
Execute	Executing a motion control function block	BOOL	True/False (False)
Outputs			
Name	Description	Date Type	Setting value(Default)
Eno	Transferring the input logic state of the *En* to the next serial instruction	BOOL	True/False (False)
Done	The execution of the function block is completed	BOOL	True/False (False)
Valid	The output pin value is valid	BOOL	True/False (False)
Busy	The motion control function block is listed for execution	BOOL	True/False (False)
Active	Axes are being controlled by function blocks	BOOL	True/False (False)
CommandAborted(Aborted)	Aborts execution for motion control function blocks	BOOL	True/False (False)
Error	Error occurs in function blocks	BOOL	True/False (False)

A motion control function block usually consists of Execute or Enable input pin and is used to either execute or enable a motion control function block. In addition, a motion control function block has Busy and Done output pins. The Busy and Done outputs refer to the status of motion control function blocks. When execution of motion control function blocks can be aborted by another motion control function block, the CommandAborted/Aborted output pin appears in the function block. Nevertheless, when Error output pin is True, this indicates an error during function block execution.

A motion control function block not only has Execute/Enable input, but also include the input value/state. The characteristics are described below.

- Use input value
  - ■ When a function block contains Execute input, each input value is used once Execute input signal changes from False to True. However, when Execute is re-triggered, input values are not updated as a result.
  - ■ When a function block contains Enable input, each input value is used once Enable input signal changes from False to True. Compared to Execute input, function blocks of Enable input usually to have more input values which need to be continuously updated. (Refer to each function block for more details).
- Input value exceeds range

When a motion control function block is enabled, the system restricts you to input values that exceeds the permitted range. Nevertheless, an error occurs during execution of motion control function blocks and results in motion axes errors. You should avoid inputting incorrect values in programs.

- Output pins are mutually exclusive.
  - ■ When a function block contains Execute input, Busy output, Done output, CommandAborted output or Error output, only one state is set to True during the same time. When Execute input is set to True, one output (Busy, Done, CommandAborted or Error) must set to True.
  - ■ When a function block contains Enable input, while Valid output and Error output are mutually exclusive, this indicates only one output is set to True.

- Valid time for output data/status value
  - ■ When a function block contains Execute input and the input signal changes from True to False, the current Done output, Error output, CommandAborted output of current True and output pin data are reset or cleared. However, when a function block is Busy, despite that the Execute input signal changes from True to False, execution of the function block will not stop. The expected output state (Done output, Error output, CommandAborted output) will generate to True and retain for one week.
  - ■ When a function block contains Enable input and input signal changes from True to False, Valid output, Busy output and Error output are reset. (For input and output description not mentioned, please refer to MC_Power instruction for more details.)
- Characteristic of Done output

When execution of a motion control function block is completed, Done output is set to True.

- Characteristic of Busy output

  - When a function block contains Execute input and uses Buy output to indicate incomplete execution, new output state (value) is to be generated. When Execute input signal changes from False to True, then Busy output is set to True. When Done output, CommandAborted output or Error output is set to True, then Busy output is reset.
  - When a function block contains Enable input and uses Buy output to indicate incomplete execution, new output state (value) is to be generated. When Enable input signal changes from False to True and as long as Busy output is set True, changes in input state (value) can be expected.

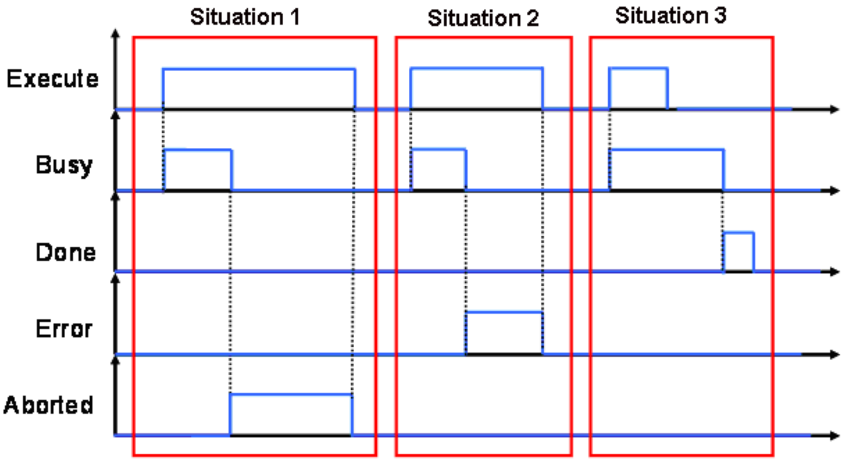
- Characteristic of CommandAborted/Aborted output

When execution of a motion control function block is aborted, CommandAborted/Aborted output is set to True.

- Relation between Enable input and Valid output

A function block contains Enable input and uses Valid output to indicate validity of output data/status. Only when Enable input is set to True and output data/status is valid, then Valid output is set to True; when an errors occur in function blocks, then output data/status is invalid and Valid output is set to False; when errors are cleared in motion control function blocks and output data/status changes to valid, then Valid output is set to True.

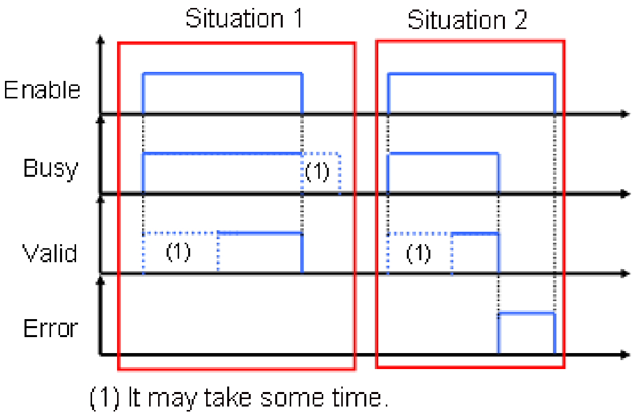
7.6.3.2 Timing Diagram for Input/Outputs



Situation 1: The execution of a motion control function block is aborted.

Situation 2: Errors occur in a motion control function block.

Situation 3: The execution of a motion control function block is completed.



Situation 1: The execution of a motion control function block is normal.

Situation 2: An error occurs in a motion control function block.

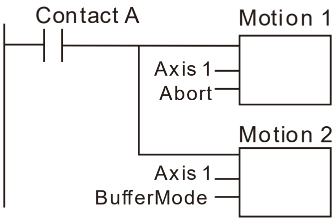
7.6.3.3 Repeated Execution Behavior of Single Axis Motion Instructions

When single axis motion function blocks are executing (Busy state), variables for input pins can be modified and function block pins can be re-triggered on the rising edge. Meanwhile, the state of function block output pins remains the same (remain Busy), while the system is executing which means it is aborting the previous rising edge-triggered instruction under buffer mode. For similar mode of behavior, refer to section 7.6.3.5 Single Axis Buffer Mode (Aborting) for more details.

7.6.3.4 Multi-execution of Motion Control Instructions

This section describes executing multiple motion control instructions for the same axis or axis group within the same scan period.

- In the following programming, instruction instances Move1 and Move2 start in the same task period when contact A turns ON.
- According to the ladder logic, instructions in a program are executed from the top. Therefore, Motion1 starts first, and then Motion 2 will be executed once Motion 1 is finished.
- This is considered multi-execution of motion control instructions. Since the motion combination is determined by input variables of BufferMode, BufferMode setting in Motion 2 is used to execute Motion 2 in relation to Motion 1.



7.6.3.5 Synchronous Execution Behavior of Motion Instructions

SoftMotion V4.10.0.0 * or later with SM3_Basic V4.10.0.0 or later supports Buffer Mode function of single-axis motion instructions, and DL-MotionControl V1.2.0.0 or later supports SoftMotion V4.10.0.0.

***Note:** SoftMotion V4.10.0.0 does not allow you to modify the parameters within Axis.REF. If any parameter within Axis.REF is modified, the error message "SMC-MOVING-WithOUT-ACTIVE-MOVEMENT" will appear.

- **Buffer Mode**

You can execute another motion control instruction while an axis is moving. A total of six types of BufferMode can be chosen to conduct multi-execution of two instructions, where you can set the Buffer Mode input variable of the later motion control instruction by selecting one of the six Buffer Modes.

The meanings of terms relating to Buffer Mode shown as follows:

1. Current instruction: The motion control instruction that was in operation just before executing the multi-execution instruction.
2. Buffered instruction: A motion control instruction that was executed during an axis motion and is waiting to be executed
3. Transit velocity: The velocity to use by the current instruction to transfer to the buffered instruction.
4. Target Velocity: The Velocity parameters of the instruction.
5. Target position: the Position or Distance parameters of related move instructions.

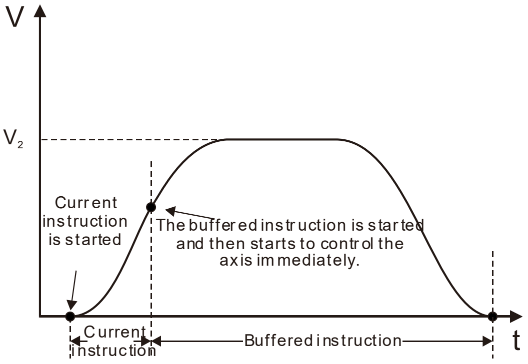
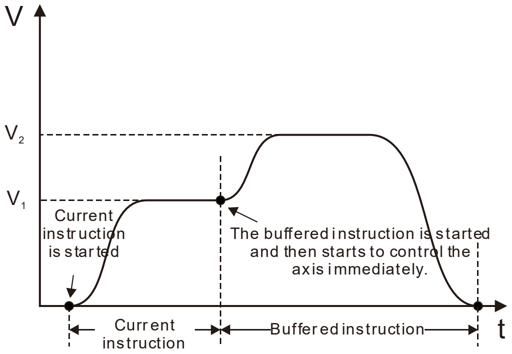
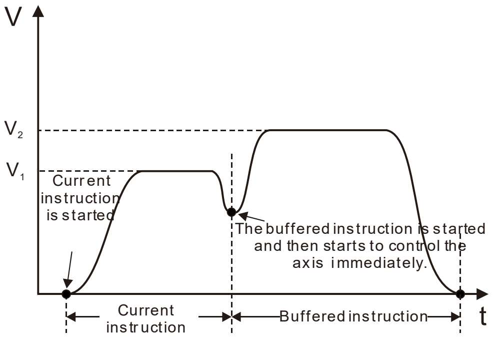
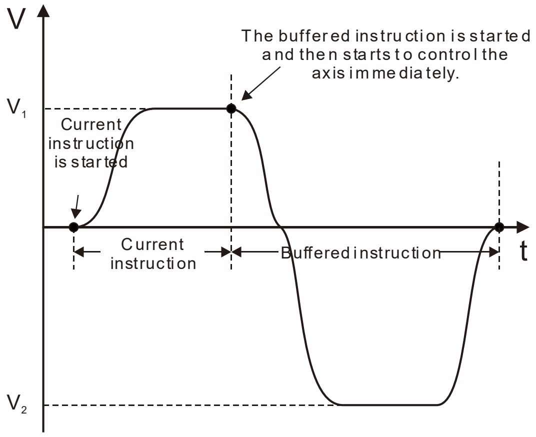
BufferMode	Description of Operation
0：mcAborting (Aborting)	The current instruction is aborted, and the multi-executed instruction is executed.
1：mcBuffered (Buffered)	The buffered instruction is executed after the operation for the current instruction is normally finished.
2：mcBlendingLow (Low velocity)	The buffered instruction is executed after the target position of the current instruction is reached. The transit velocity is set to the target velocity of the current instruction or the buffered instruction, whichever is lowest.
3：mcBlendingPrevious (Previous velocity)	The buffered instruction is executed after the target position of the current instruction is reached. The target velocity of the current instruction is used as the transit velocity.
4：mcBlendingNext (Next velocity)	The buffered instruction is executed after the target position of the current instruction is reached. The target velocity of the buffered instruction is used as the transit velocity.
5：mcBlendingHigh (High velocity)	The buffered instruction is executed after the target position of the current instruction is reached. The transit velocity is set to the target velocity of the current instruction the buffered instruction, whichever is highest.

- **Example:** Briefly explain with two MoveRelative instructions

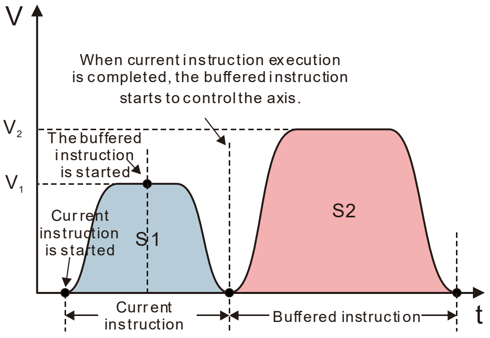
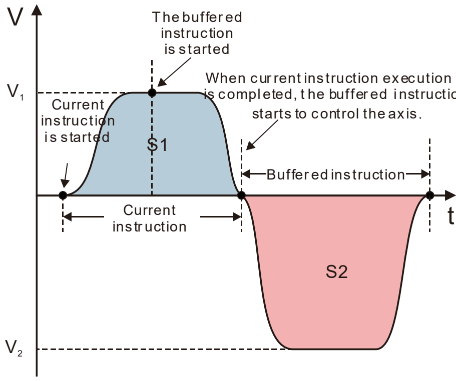
  The max velocity and the displacement of the first and second instructions are respectively V_1, S_1 and V_2, S_2.

  Different types of BufferModes set for the second instruction result in various transmitting situation shown as follows.

- **Buffermode=mcAborting**

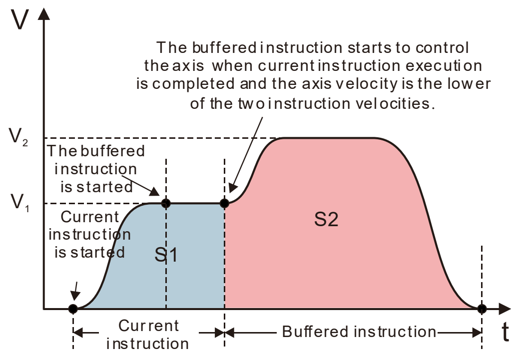
1. Present instruction aborted during acceleration.	2. Present instruction aborted at constant speed.
	
3. Present instruction aborted during decelerating.	4. The instruction transferred and the present instruction are in opposite directions.
	

- **Buffermode=mcBuffered**

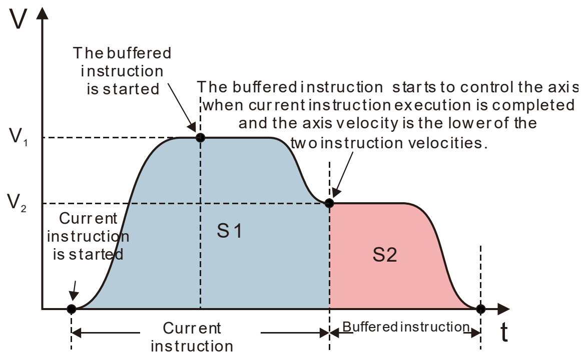
1. The instruction transferred and the present instruction are in the same direction.	2. The instruction transferred and the present instruction are in opposite directions.
	

- **Buffermode=mcBlendingLow**

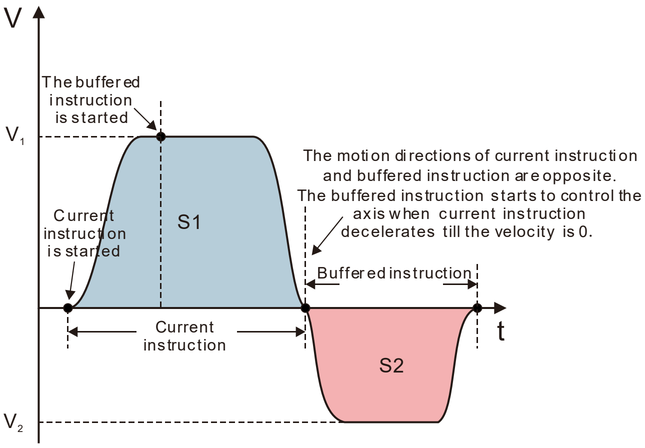
1. The velocity of the current instruction is slower



2. The velocity of the current instruction is greater

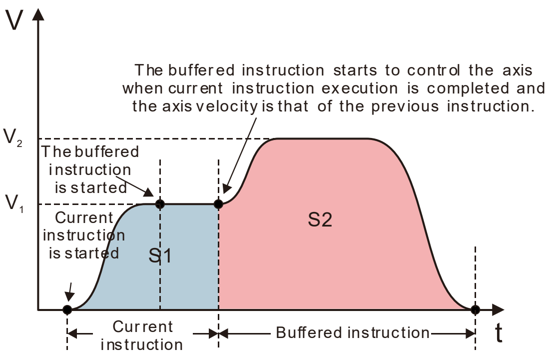


3. The instruction transferred and the present instruction are in opposite directions.

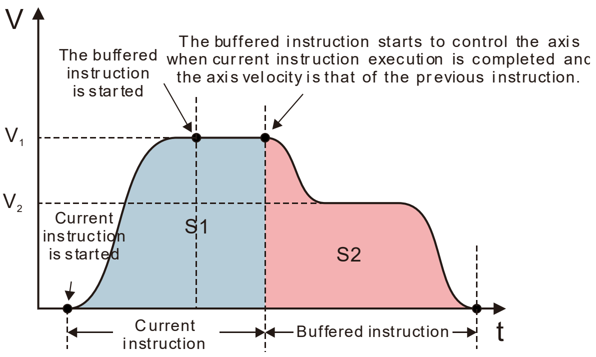


- **Buffermode=mcBlendingPrevious**

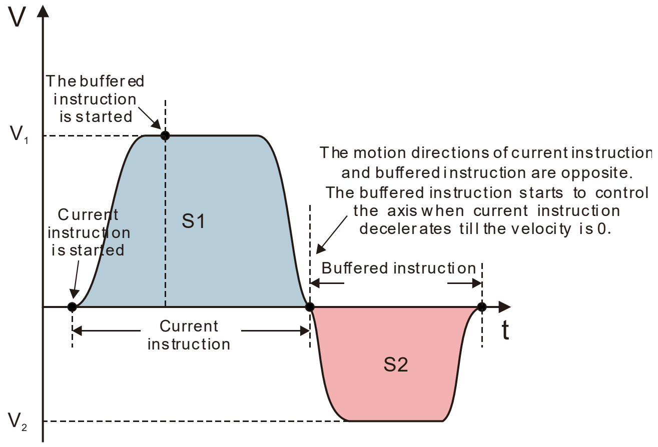
1. The velocity of the current instruction is slower



2. The velocity of the current instruction is greater

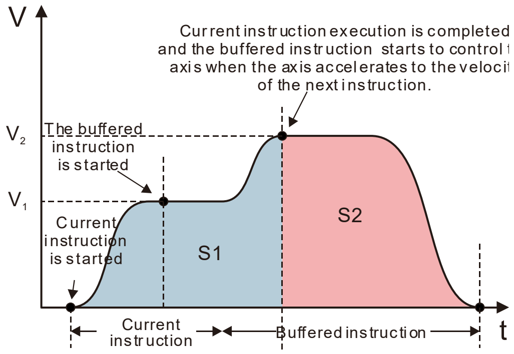


3. The instruction transferred and the present instruction are in opposite directions.

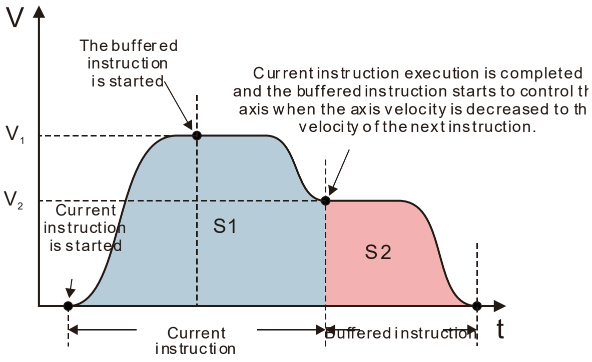


- **Buffermode=mcBlendingNext**

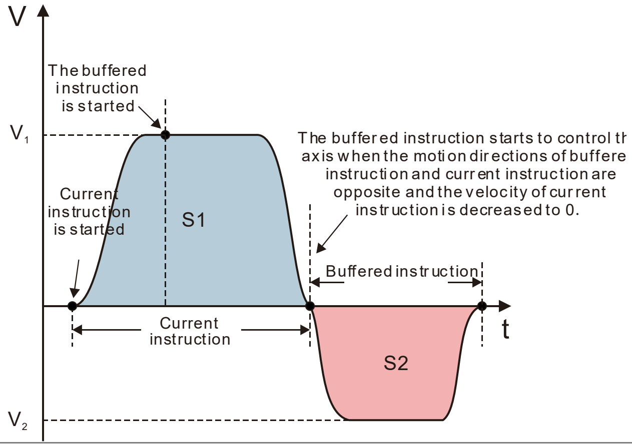
1. The velocity of current instruction is slower



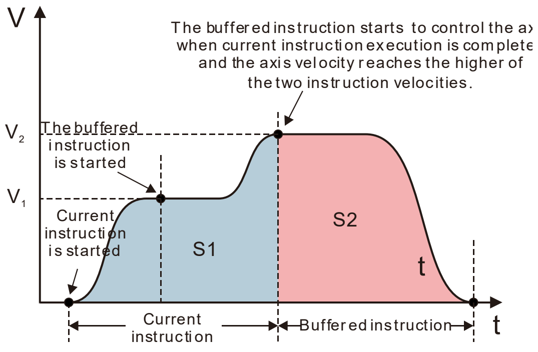
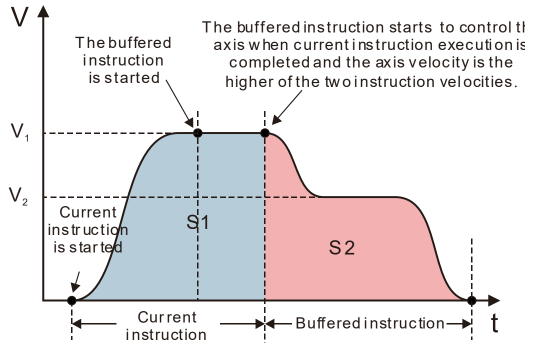
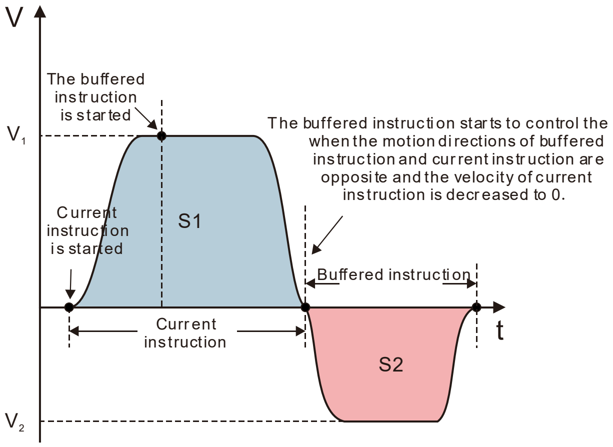
2. The velocity of current instruction is greater



3. The instruction transferred and the present instruction are in opposite directions.



- **Buffermode=mcBlendingHigh**

<table>
<tr><th>1. The velocity of current instruction is slower</th><th>2. The velocity of current instruction is greater</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><th colspan="2">3. The instruction transferred and the present instruction are in opposite direction.</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

***Note:** Single-axis motion instructions MC support only Buffermode=mcAborting while motion instructions for axis group support all of the above BufferMode.

7.6.4 Position

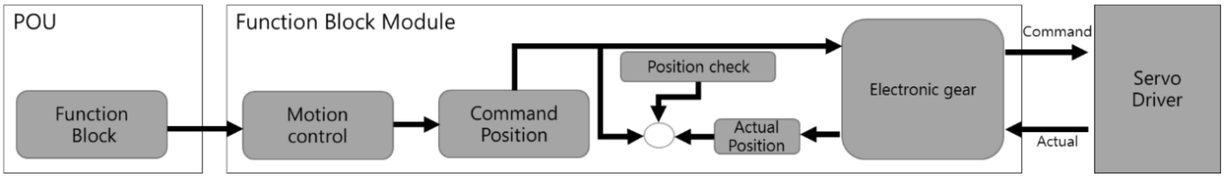
This section describes the position processes of motion control programming.

7.6.4.1 Types of Positions

MC function blocks are formed by the following two types of positions.

- Command position: MC function block provides the command position.
- Actual position: The actual feedback position from servo drives.

The following figure indicates the relationship between the command position and the actual position.



Command position and actual position:

Position Type	Description
Command position	This is the position that the motion controller outputs to the servo drive
Actual (feedback) position	This is the position feedback from the servo drive or encoder

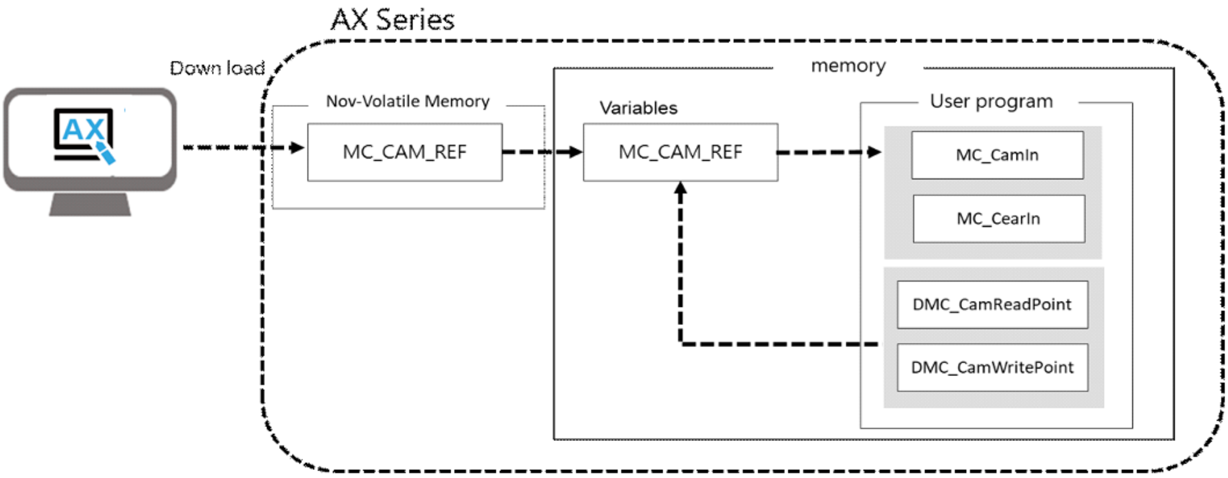
***Note:** For axes configured as Virtual, the actual position is equal to the command position.

7.6.5 CAM Tables and Framework

This section introduces electronic cam (E-CAM) operation and how to use DIADesigner-AX to generate a CAM table, as well as E-CAM applications. For details regarding instructions, please refer to **AX Series Motion Controller Manual**.

7.6.5.1 E-CAM Framework

Adopt CAM Editor function in the software DIADesigner-AX for planning CAM curves, download the data to PLC via communication and then use MC function blocks to control CAM.



7.6.5.2 Creating E-CAM

The data that defines the relationship between master/slave (CAM axis) is called E-CAM data.

When using CAM Editor of DIADesigner-AX , it is crucial to know the relationship between master and slave axis position through the two methods described below:

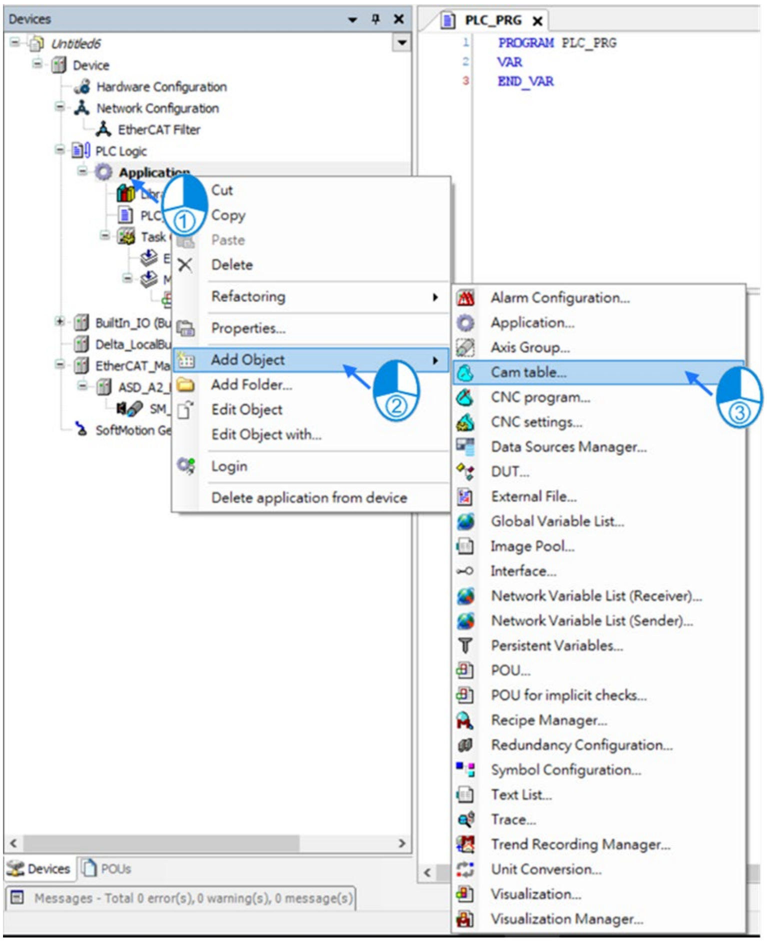
Method 1: Obtains the relationship between master and slave axis position based on E-CAM data setting.

Method 2: Measures the corresponding relationship between master and slave axis position through real tasks.

When the CAM master-slave relationship is confirmed, the slave position can be obtained based on the master axis position.

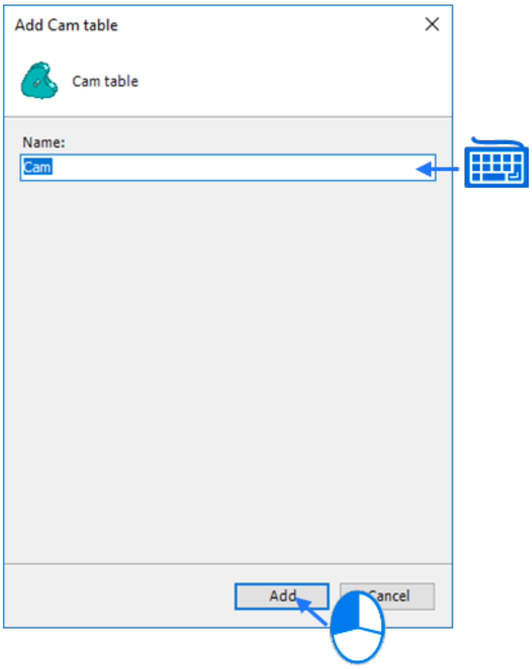
- **Create DIADesigner-AX CAM tables**

(1) Right-click “Application,” choose “Add Object” and then select “CAM Table.”



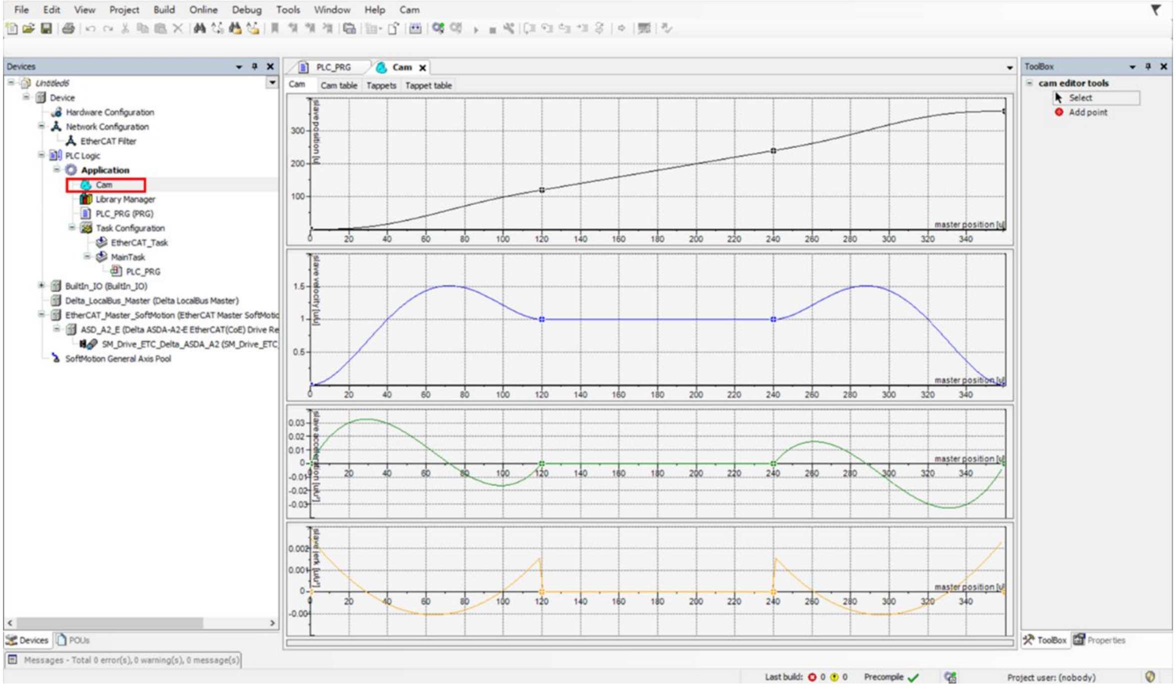
```
PROGRAM PLC_PRG
VAR
END_VAR
```

(2) Type the name for the CAM table.



Name: Cam

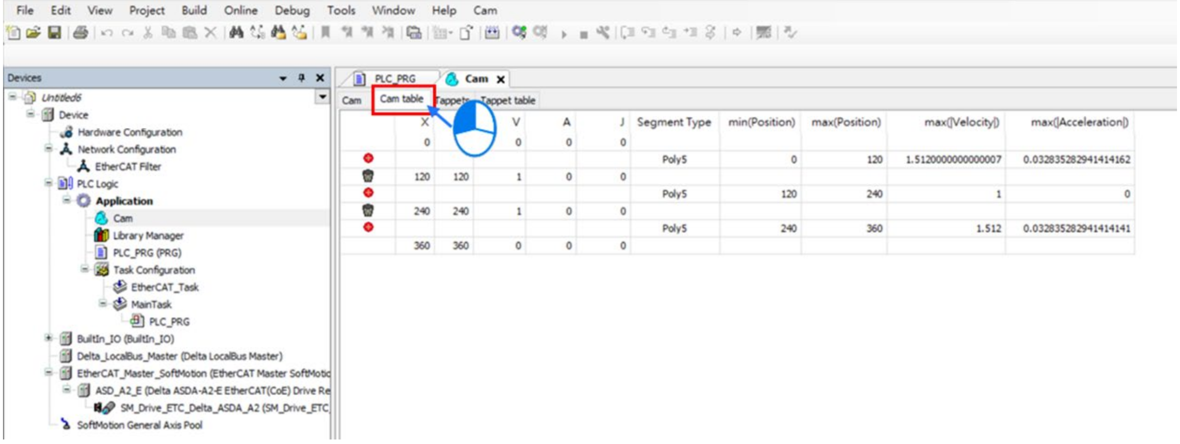
(3) After clicking “Add,” CAM icon is shown on the left item box.



Messages - Total 0 error(s), 0 warning(s), 0 message(s)

Last build: 0 0 Precompile ✓ Project user: (nobody)

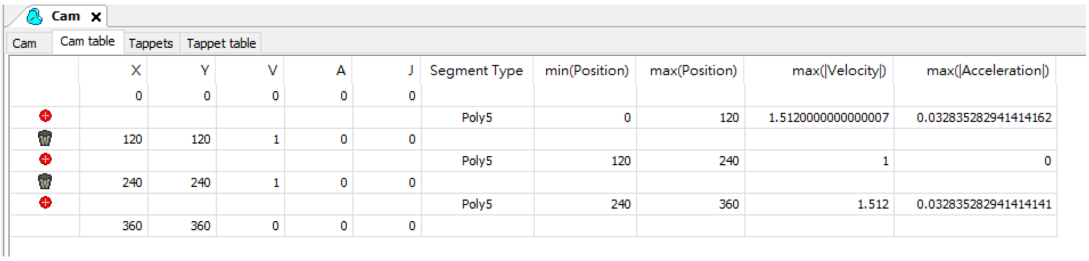
(4) Click “Cam Table” on the CAM page.



	X		V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0		0	0	0					
⊕						Poly5	0	120	1.5120000000000007	0.032835282941414162
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.032835282941414141
	360	360	0	0	0					

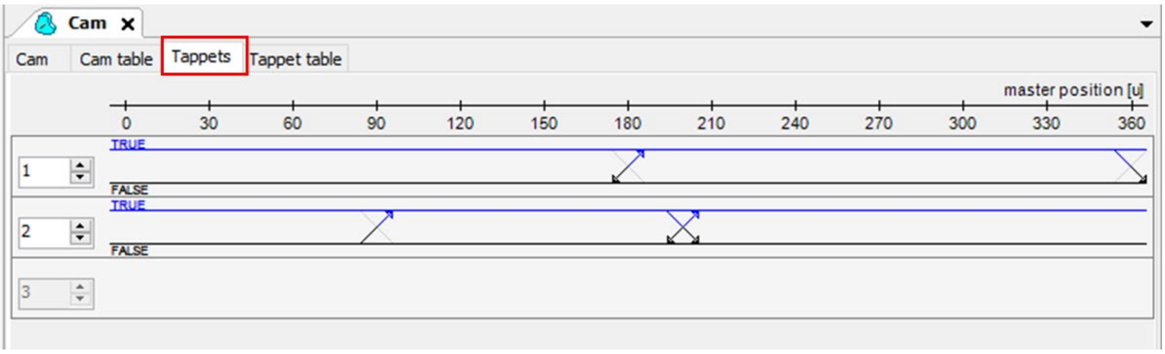
(5) Add or delete CAM data on the CAM Table screen

- Click ⊕ to add new CAM data
- Click 🗑 to delete CAM data
- X: Position data of master axis
- Y: Position data of slave axis
- A: Acceleration of slave axis
- J: Jerk of slave axis
- Segment Type: Curve type



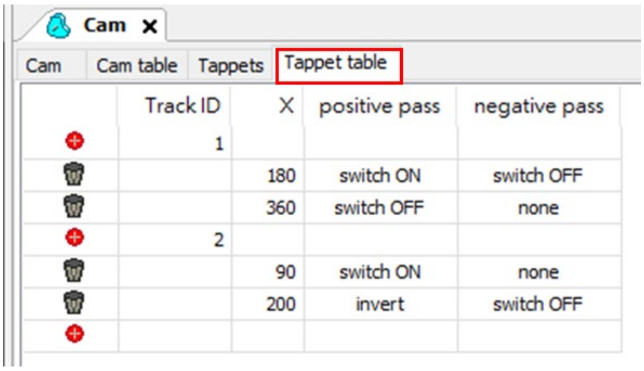
	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	120	1.5120000000000007	0.032835282941414162
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.032835282941414141
	360	360	0	0	0					

(6) You can configur multiple tappets on “Tappets” page and several tappets can be set for each tappet ID. After you finish setting “Tappet table”, a diagram which illustrates the relation between tappets and master axes would be shown on “Tappets “ page. While you are moving the points on Tappets page, the setting parameters on Tappet table page would be changed simultaneously.



(7) You can configure tappets on “Tappet table” page and read the status of tappets with SMC_GetTappetValue, which can also be modified according to the settings in “Tappet table” and the direction when CAM master passes the tappets.

- Click ⊕ to add new Track ID.
- Click 🗑 to delete TrackID.
- Track ID: Tappet ID
- X: Master position
- Positive pass: Axis passes tappets in positive direction, which the setting is as below:
    - None: No action
    - Switch to ON: TRUE
    - Switch to OFF: FALSE
    - Invert: Opposite direction
- Negative pass: Axis passes tappets in negative direction, which the setting is as below:
    - None: No action
    - Switch to ON: TRUE
    - Switch to OFF: FALSE
    - Invert: Opposite direction



	Track ID	X	positive pass	negative pass
⊕	1			
🗑		180	switch ON	switch OFF
🗑		360	switch OFF	none
⊕	2			
🗑		90	switch ON	none
🗑		200	invert	switch OFF
⊕				

7.7 Motion Control Functions

7.7.1 System Structure

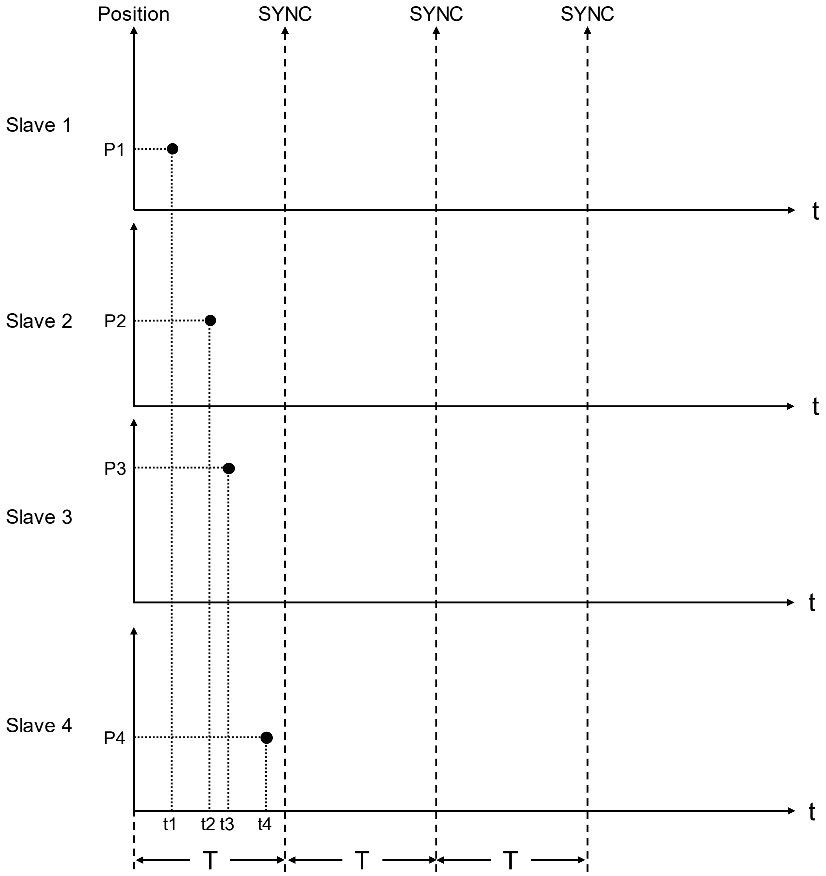
The single axis motion instructions of MC function blocks can generate specified motion paths for axes based on user-defined parameters under three control modes including position control, velocity control, and torque control.

AX series controllers support the CANopen over EtherCAT (CoE) protocol. In the CiA 402 specification, the supported motion modes include Cyclic Synchronous Position Mode, Cyclic Synchronous Velocity Mode and Cyclic Synchronous Torque Mode, which are explained in the following sections.

7.7.2 Single-axis Control

7.7.2.1 Cyclic Synchronous Position Mode

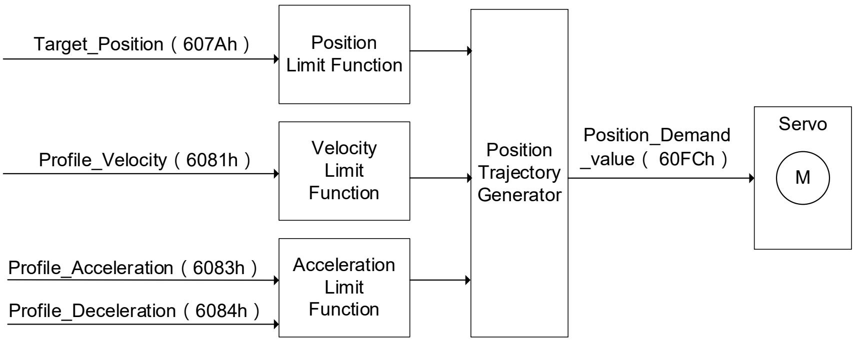
The synchronization between AX series controllers and servo drives is implemented via sync signal transmission sent by controllers. These incoming data would not be valid until the Distributed Clocks (DC)* in servo drives are synchronized. In the following figure, four servo drives receive control data at different timing (t1, t2, t3, t4) within a synchronous cyclic time (T). However, the data is valid after all servo drives are synchronized with the SYNC event of the distributed clock system.



***Note:** Cyclic synchronous position mode is used only for synchronous axes.

7.7.2.2 Profile Position Mode

After the servo drive receives position demands from the master device, the drive controls the motor to reach the target position. Under profile position mode*, at first the master device only informs the drive about configuration relating to target position, velocity command, acceleration, and deceleration. All motion plannings are executed by the trajectory generator inside the servo drive, from triggering demand to reaching target position.



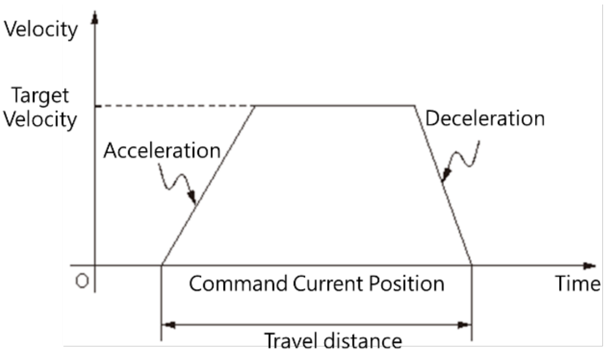
* Profile position mode is only used for positioning axes.

7.7.2.3 Positioning

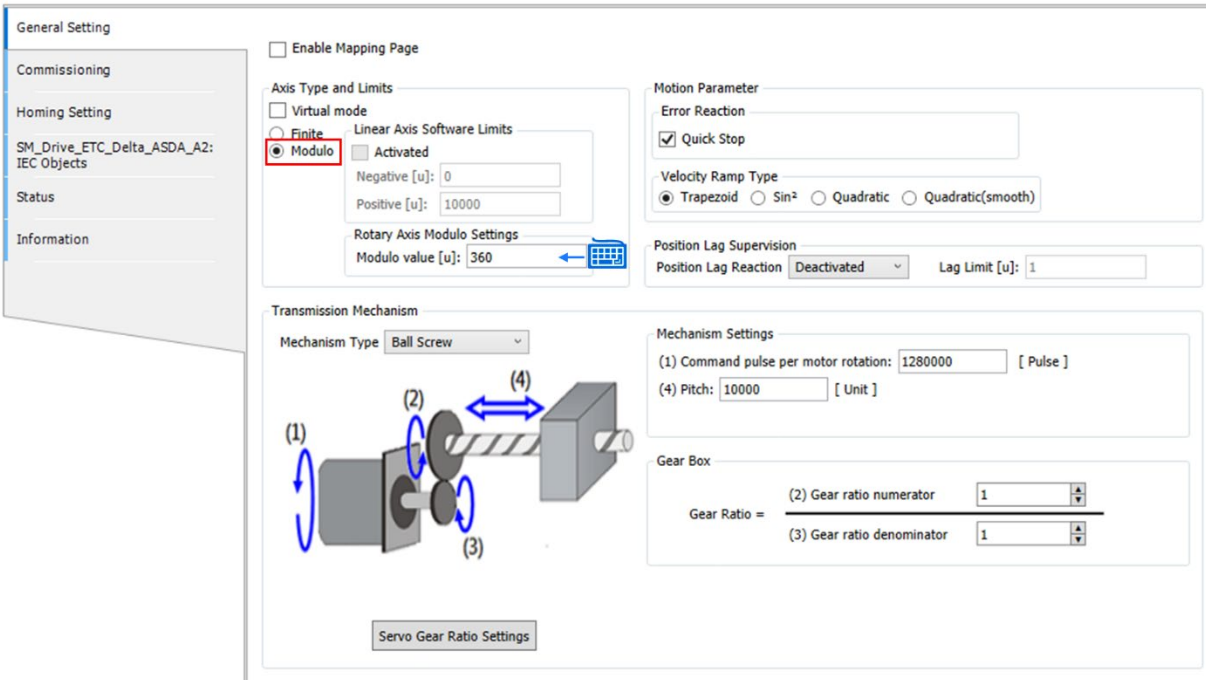
- **Absolute positioning**

The curves for motion planning allows an axis to move to the absolute coordinates of the target position in relation to home. In addition, the absolute positioning range for modulo axis is limited to the range of its cyclic rotation. Please refer to MC_MoveAbsolute function block for more information.

The following figure shows the motion trajectory for absolute positioning.



- **Rotary axes setting**
  - After choosing “Rotary Axis” for axis type, set the angle range of rotation for the rotary axis in “Modulo value” area.



☐ Enable Mapping Page

Axis Type and Limits
☐ Virtual mode
○ Finite
● Modulo
Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 10000
Rotary Axis Modulo Settings
Modulo value [u]: 360

Motion Parameter
Error Reaction
☑ Quick Stop
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

Position Lag Supervision
Position Lag Reaction: Deactivated
Lag Limit [u]: 1

Transmission Mechanism
Mechanism Type: Ball Screw

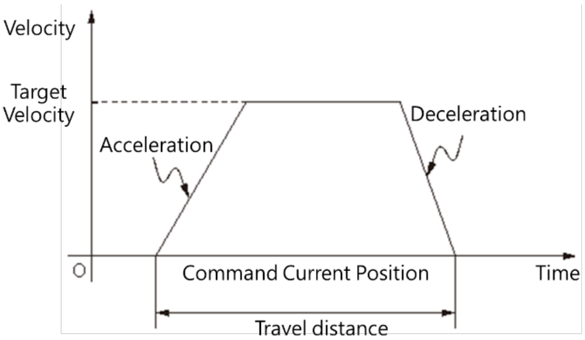
Mechanism Settings
(1) Command pulse per motor rotation: 1280000 [Pulse]
(4) Pitch: 10000 [Unit]

Gear Box
Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

- Relative positioning

The curves for motion planning allows an axis to move to the relative coordinates of the target position in relation to the actual position. Please refer to MC_MoveRelative function block for more information.

The following figure shows the motion trajectory for relative positioning.



7.7.2.4 Stop Method

The stopping state includes using motion instructions or enabled limit input as well as error stop input to stop axis operation. The stop behavior regarding clear error and limit input differs depending on the servo drives.

- **Using motion instructions to stop**

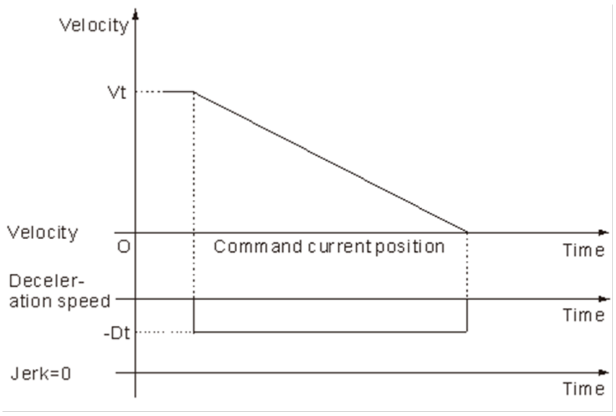
  To stop single-axis movement, use MC_Stop or MC_Halt instruction.

- **MC_Stop**

- MC_Stop stops an axis in motion based on the specified method and changes the state to “Stopping.”
- The instruction aborts any instructions in execution. When the axis state is “Stopping,” no instructions can be executed.
- The state of “Stopping” continues until velocity reaches 0 or Execute becomes False. When velocity is 0, Done changes to True.
- When Done becomes True and Execute is False, the axis changes to “Standstill” state.

The following diagram shows MC_Stop motion trajectory.

Velocity is determined by specified deceleration (DT).



Vt：Velocity before the deceleration slope starts Dt：The specified deceleration rate

- **MC_Halt**

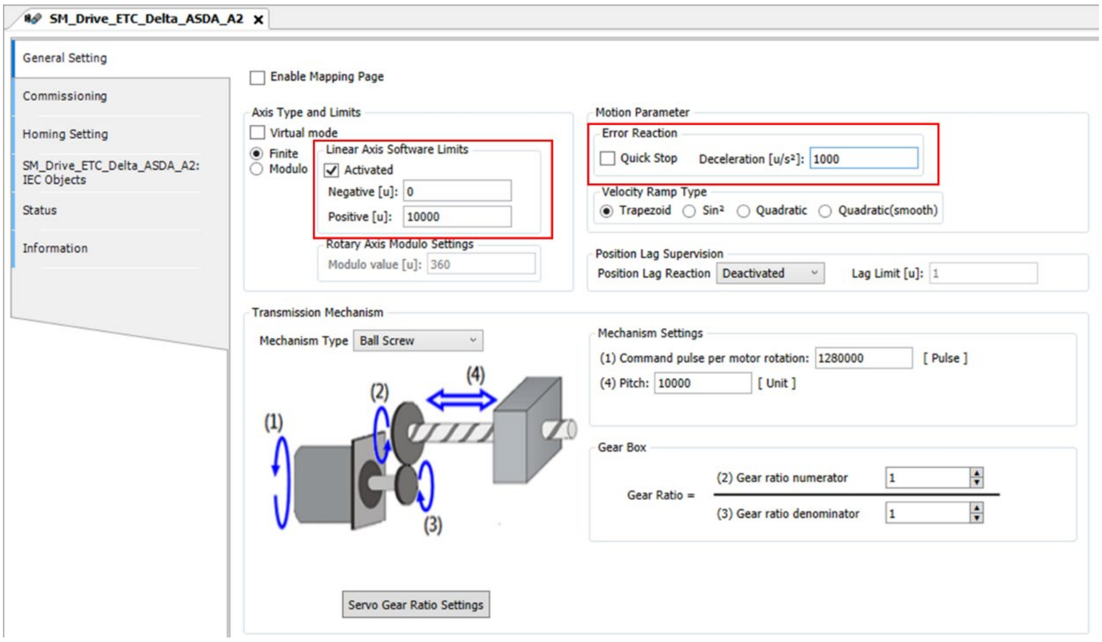
- MC_Halt temporarily stops an axis in motion and changes the axis state to “DsicreteMotion” until the axis velocity reaches 0. When the axis stops, the axis state changes to “Standstill.”
- During axis deceleration, other motion instructions can be executed to immediately abort MC_Halt operation.

- **Limit input stop**

Software limit: You can activate/ inactivate the software limit and configure its parameter settings on axis parameter setting page. When the axis is close to the software limit during the movement, it will start the deceleration stop based on the axis parameters and stop under the software limit.

The example is shown as below:

- The positive and negative limits are respectively set as 10000 and 0 with “Activated” being selected. Then set 1000 for Deceleration.



☐ Enable Mapping Page

Axis Type and Limits
☐ Virtual mode
● Finite
○ Modulo

Linear Axis Software Limits
☑ Activated
Negative [u]: 0
Positive [u]: 10000

Rotary Axis Modulo Settings
Modulo value [u]: 360

Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 1000

Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

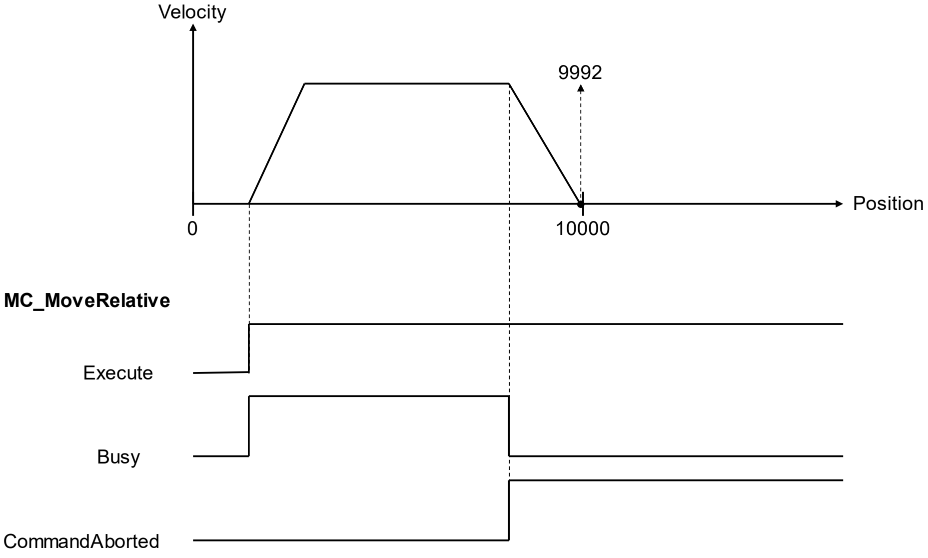
Position Lag Supervision
Position Lag Reaction: Deactivated Lag Limit [u]: 1

Mechanism Type: Ball Screw

Mechanism Settings
(1) Command pulse per motor rotation: 1280000 [Pulse]
(4) Pitch: 10000 [Unit]

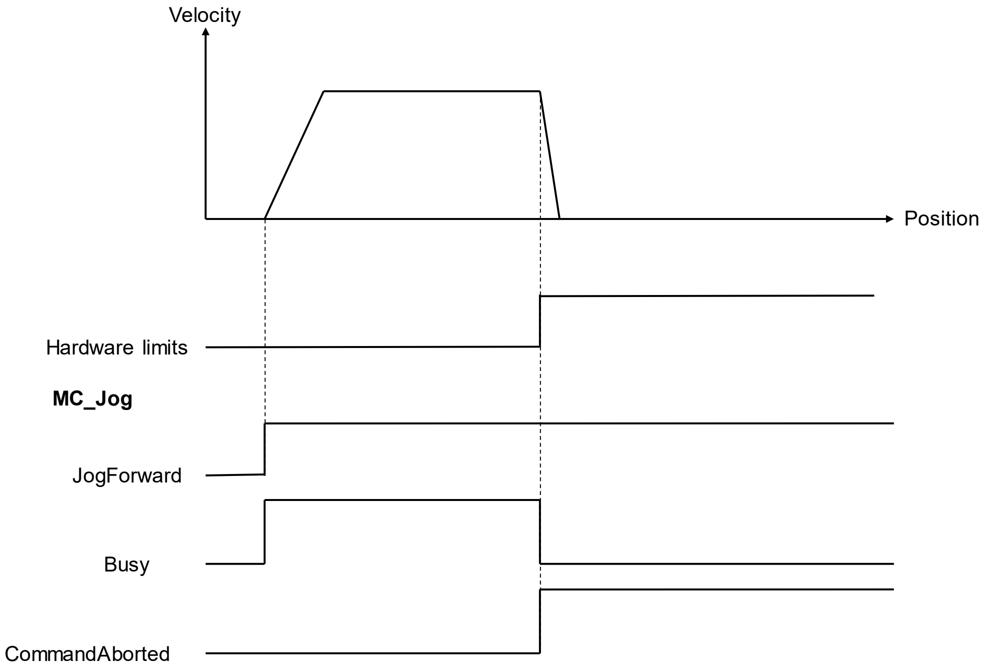
Gear Box
Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

- Use function block MC_MoveRelative and activate the function block when the target position is set to 11,000. After the axis reaches about 8,000, Busy of the function block will shift from TRUE to FALSE, while CommandAborted shifts from FALSE to TRUE. The axis then starts to decelerate and stop at the position inside the software limit.



Hardware limit: Since the EtherCAT servo wires carry the hardware limit signals, the stop method for hardware limit may be different among vendors and brands. The following description takes Delta ASDA-A2-E servo drive as example:

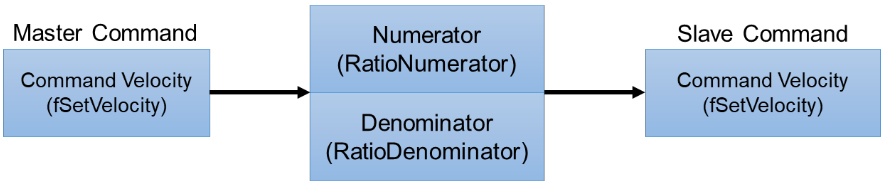
- Use MC_Jog function block to perform the rotation of the servo axis in the positive direction. Once the hardware limit is reached during the rotation, ASDA-A2-E servo drive will be stopped and report error messages via communication.



After using MC_Reset to clear errors for reaching the software/ hardware limit, the system synchronizes the command position based on the feedback position value automatically, and then the servo moves away from the direction of limit to operate properly again.

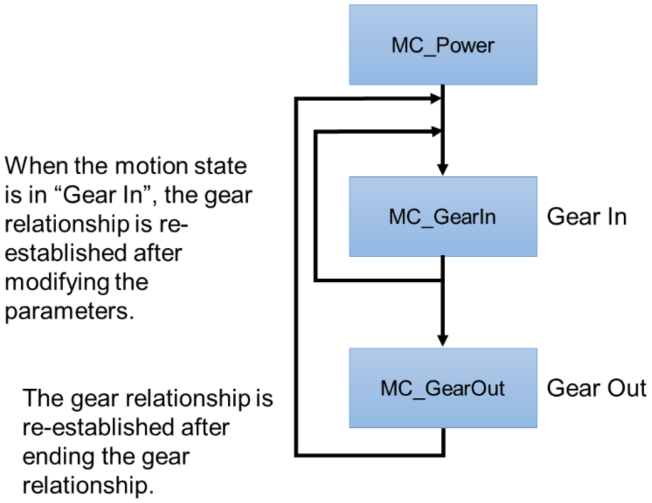
7.7.2.5 MC_GearIn

Use MC_GearIn instruction to perform the gear movement of master and slave axes at a gear ratio. The slave axis will no longer proceed with the synchronous motion via MC_Gear Out instruction and will move at a constant velocity. For details, refer to **AX Series Motion Controller Manual**.



In MC_GearIn, the master and slave axes, gear ratio numerator and gear ratio denominator, acceleration, deceleration as well as jerk are specified.

- **The following diagram shows the execution steps of instructions for electronic gears:**

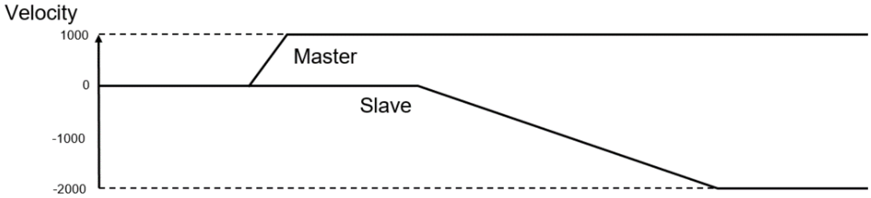


- When executing MC_GearIn, the slave axis enters the state of synchronized motion, while for MC_GearOut execution, the slave axis shifts always from sync state and maintains instant velocity to continue the movement and enters the state of continuous motion.
- During synchronized motion, when executing MC_Stop on the slave axis, MC_GearIn is aborted while the master axis maintains the state of continuous motion and the slave axis enters to stopping state that will return to standstill once MC_Stop is Done.
- When the slave axis is in synchronized motion state, its velocity may alter according to the master axis velocity and gear ratio.
- When both master and slave axes enter state of synchronization, use MC_SetPosition to prevent motors from generating accidents due to high speed operation.

- **Using RatioNumerator, RatioDenominator in MC_GearIn to setup the gear ratio between master and slave axes.**
  - When gear ratio is positive, the master and slave axes are moving in the same direction.

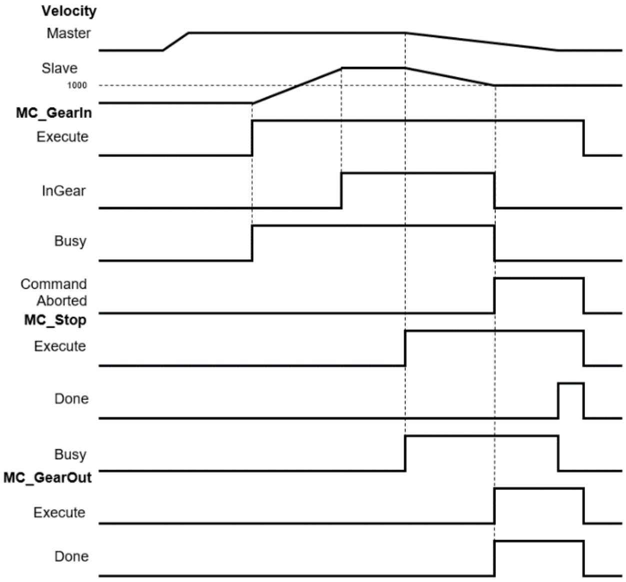
Ratio Numerator: Ratio Denominator = 2:1

  - When gear ratio is negative, the master and slave axes are moving in the opposite direction.



Ratio Numerator: Ratio Denominator = -2:1

  - Synchronization of master and slave axes is completed once slave velocity reaches the setting in the instruction.



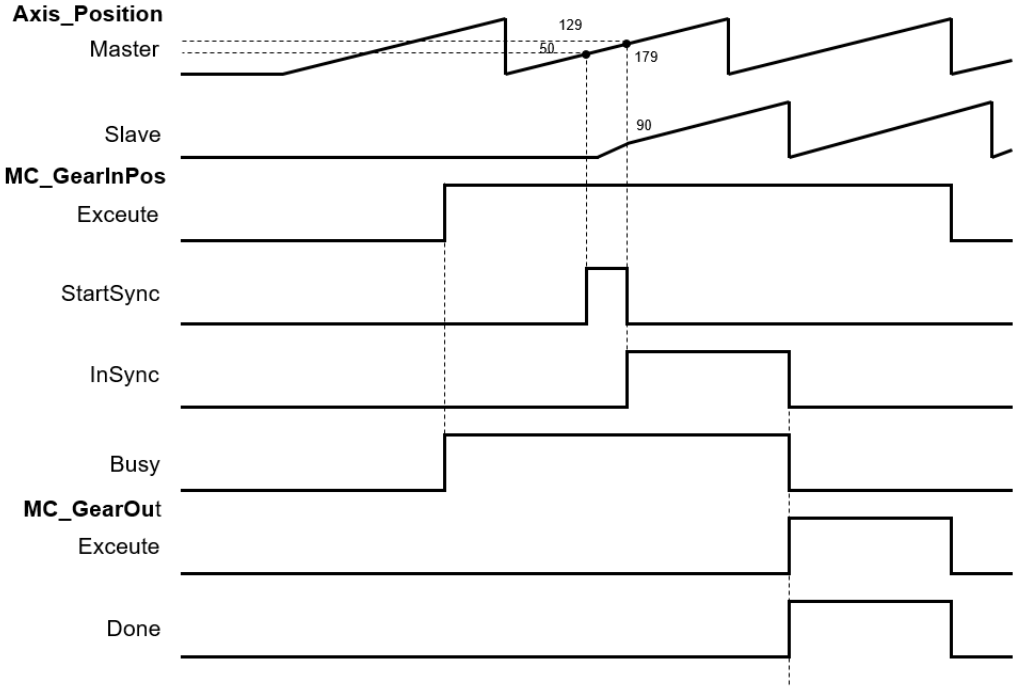
1. When MC_GearIn is enabled, the slave starts to engage with the master axis and the slave velocity is twice the speed of the master velocity (RatioNumerator : RatioDenominator = 2:1).
2. When InGear is True, synchronization of master and slave axes are completed and slave axis is in synchronized motion state.
3. When MC_Stop is enabled, the master axis starts decelerating and the slave axis in sync also decelerates based on the gear ratio.
4. When MC_Stop is operating and MC_GearOut is enabled, the sync between master and slave axes is aborted but the slave axis maintains that velocity and is in continous motion state.

7.7.2.6 MC_GearInPos

You can adopt MC_GearInPos to specify the synchronous starting positions of master and slave axes.

- MC_GearInPos sequence

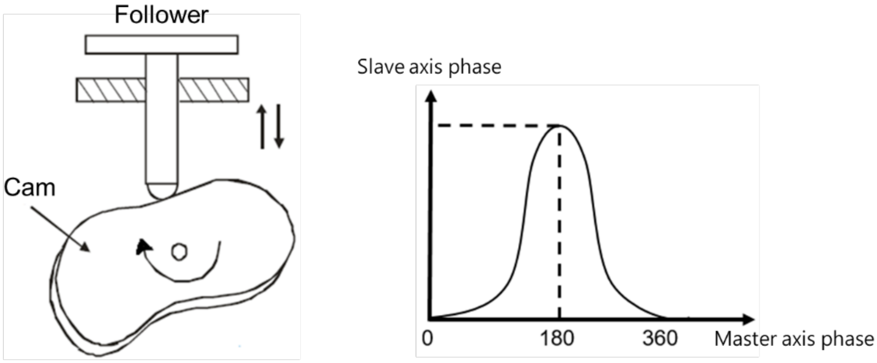
The specified master and slave axes, gear ratio numerator and denominator, synchronous starting positions of master and slave axes in MC_GearInPos executes the master start distance in sync as well as whether or not to permit reversal. The function block engages both master and slave axes in the specified position based on the planned curve for the slave axis.



- The master axis starts to execute sync position as MasterSyncPosition(180) – MasterStartDistance(50). When the axis reaches that position, StartSync is True.
- The slave axis generates a motion curve based on other parameters. When the master reaches MasterSyncPosition(180) and the slave axis also reaches SlaveSyncPosition(90), the StartSync is False and InSync is True.
- When MasterStartDistance ≦ 0, the function block executes and synchronization is completed; Meanwhile, the slave axis position will move up and down to the assigned sync position.
- When slave reversal is not permitted, you need to set AvoidReversal to True.

7.7.2.7 MC_CamIn

The slave axis follows the master axis for the synchronized motion based on a CAM table. Select a cam table for an e-cam motion via MC_CamTableSelect. Use MC_CamIn for CAM engagement, and MC_CamOut to remove cam engagement.



After the engagement, synchronization between master and slave axes is completed successfully and the state of the slave axis is Synchronized Motion. The following is the information about creating E-CAM:

- **Initial setting**

\- Create E-CAM data

The following two methods can create E-CAM curve data:

Method 1: Master and slave positions are determined based on standard functions.

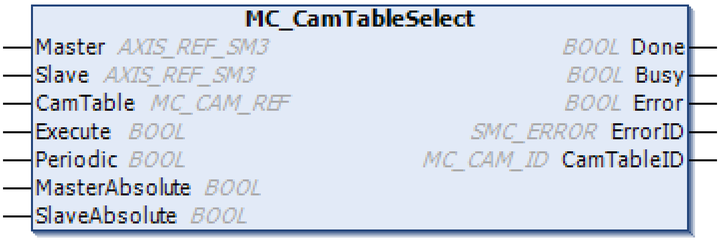
Method 2: The corresponding relationship between master and slave is based on actual measurement.

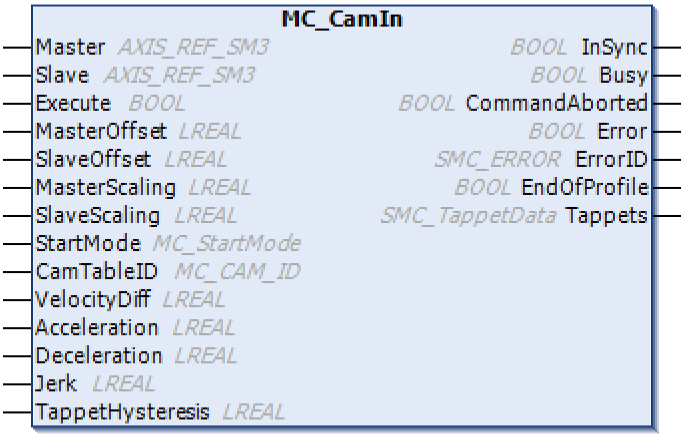
- **E-CAM master and slave setting and operation**

By using MC_CamIn and MC_CamTableSelect, E-CAM slave and master as well as basic operation setups can be completed.

\- Master and slave sources setting

In MC_CamTableSelect and MC_CamIn function blocks, the master input pins determine the master source while slave input pins determine the slave source.



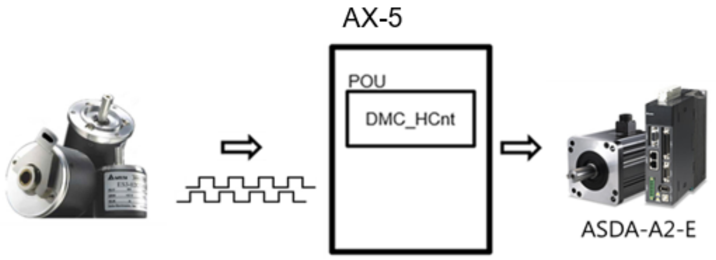


***Note:** For more details of pins definition, please refer to **AX Series Motion Controller Manual**.

- Master as external pulse counter

The sources of E-CAM master include real and virtual axes as well as the counter. When using the external counter as master's source, use DMC_HCnt function block.

- System structure and DMC_HCnt



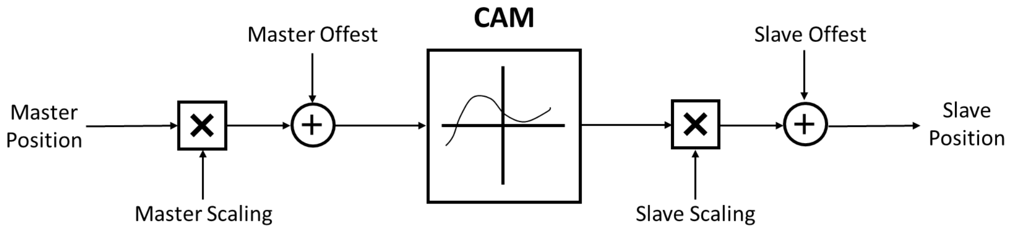
- Relationship between master and slave positions

By using the software to pre-plane the relationship between CAM master and slave positions, the positions in the CAM table rather than actual axis positions define the phase of the master and slave axes. When the pre-planned CAM mechanism is defined as CAM function, the input is the CAM master phase, and the output is the CAM slave phase. For example:

x: CAM master phase ; y: CAM slave phase

$$y = CAM(x)$$

The CAM phase derives from the axis position and conversion may take place. The conversion between axis position and CAM phase is related to parameters including MasterAbsolute, SlaveAbsolute, MasterOffset, SlaveOffset, MasterScaling and SlaveScaling. The slave follows the master axis to perform synchronized motion under MC_CamIn instruction. The relationship between master and slave positions should be based on the pre-planned CAM relationship (relation curve or CAM table). The process of calculating the slave position from the master position is shown below:



The above diagram results in the following calculation method:

Position_Slave = SlaveScaling×CAM (MasterScaling×MasterPosition + MasterOffset) + SlaveOffset

When master is in absolute mode, the current master position is the arithmetic result of the rotating axis; when in relative mode, the master position is the starting point (usually 0) for CAM motion.

- Relationship between Startmode and MasterAbsolute, SlaveAbsolute in CamTableSelect

- Absolute mode (StartMode=0): When E-CAM synchronization starts, the CAM calculation and current slave position is irrelevant. When current slave position is different from the starting position that is calculated, then Jump is generated.
- Relative mode (StartMode=1): CAM changes based on current slave positions; the slave positions are added from its current position. When the engaging position of the slave is different from the starting position plus the current position that is calculated, then Jump is generated.
- Ramp mode (StartMode = 2, 3, 4): Add a curve of motion compensation based on VelocityDiff, Acceleration, Deceleration, Jerk to prevent the Jump during CAM engagement.

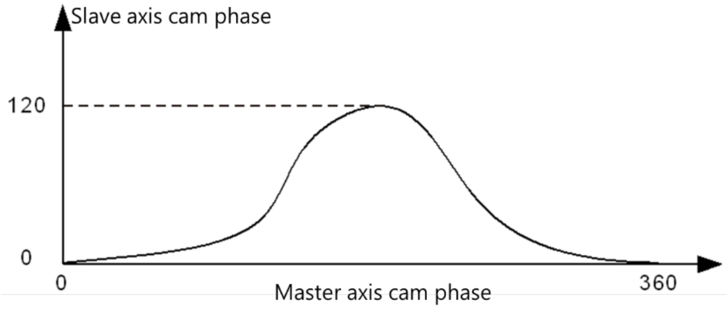
MC_CamTableSelect.MasterAbsolute	Master mode
absolute	Absolute mode
relative	Relative mode

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	Slave mode
absolute	True	Absolute mode
absolute	False	Relative mode
relative	True	Relative mode
relative	False	Relative mode
ramp_in	True	Ramp in absolute mode
ramp_in	False	Ramp in relative mode
ramp_in_pos	True	Positive ramp in absolute mode
ramp_in_pos	False	Positive ramp in relative mode
ramp_in_neg	True	Negative ramp in absolute mode
ramp_in_neg	False	Negative ramp in relative mode

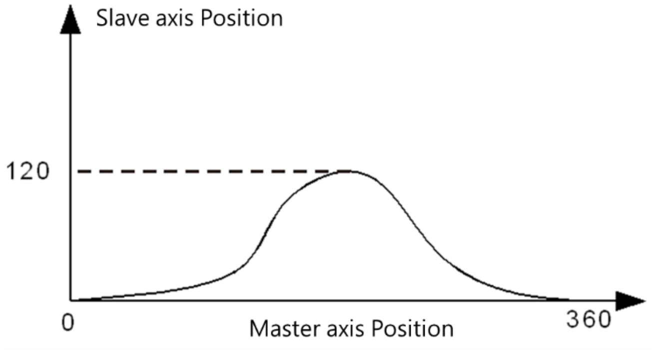
- Offset and scaling (MasterOffset/MasterScaling/SlaveOffset/Slavescaling)

Since the CAM mechanism between master and slave is pre-planned, when executing CAM, you can adopt Offset and Scaling parameters to pre-plane position offset or scaling. For example, the processing product has different dimensions, but only one CAM mechanism is required for programming, therefore, by changing offset and scaling parameters, the switching of processing products amongst different dimensions can be adjusted. You can input specific scaling values for master scaling of CAM and slave offset. The master and slave can set up offset and scaling values accordingly.

The master and slave offset and scaling both determine the actual CAM in relation to the effect that is described in the following example. The diagram below demonstrates pre-planned CAM mechanism:



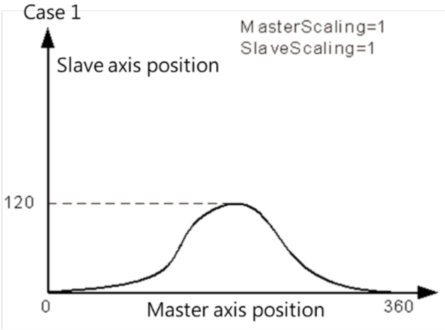
When master and slave are both in absolute mode and executes engagement, both master and slave positions are 0; when not using offset and scaling (default value), the following diagram shows the actual corresponding relationship between master and slave during the process of executing CAM:

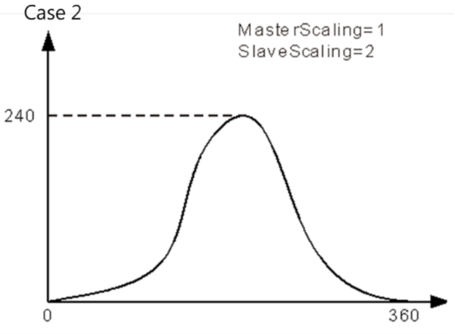


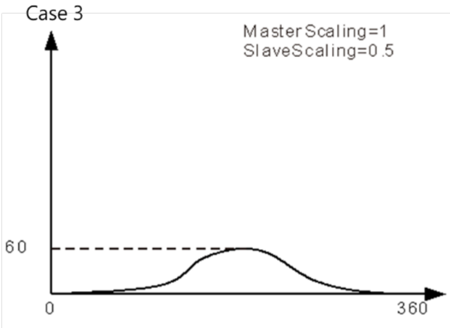
When position offset or scaling is not in default value, the following diagrams show the effects of the corresponding relationship between master and slave actual positions during CAM execution:

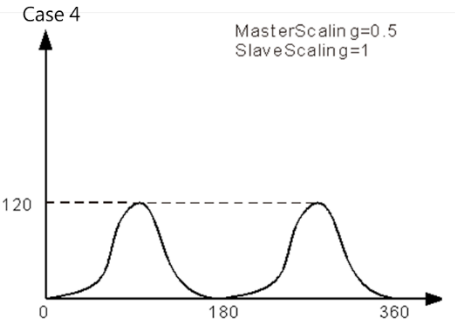
With master and slave offsets as 0, the effects from scaling of master and slave for actual CAM execution

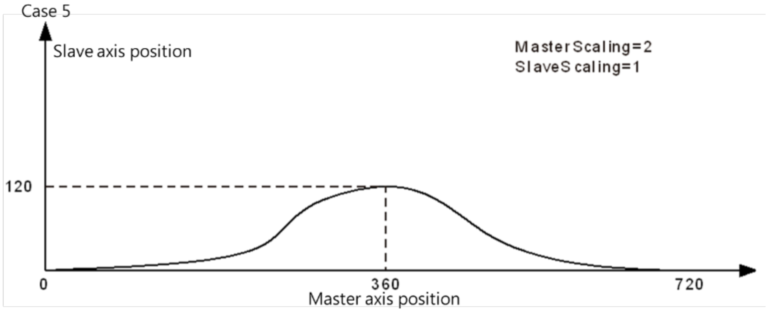
Situations:











Situation 1: When scaling ratio for master and slave is 1 and offset is 0, the actual CAM mechanism is the same as pre-planned.

Situation 2: When master scaling ratio is 1, slave scaling ratio is 2 and offset for both axes is 0, the slave position that corresponds to the master position is twice the amount of pre-planned measurement.

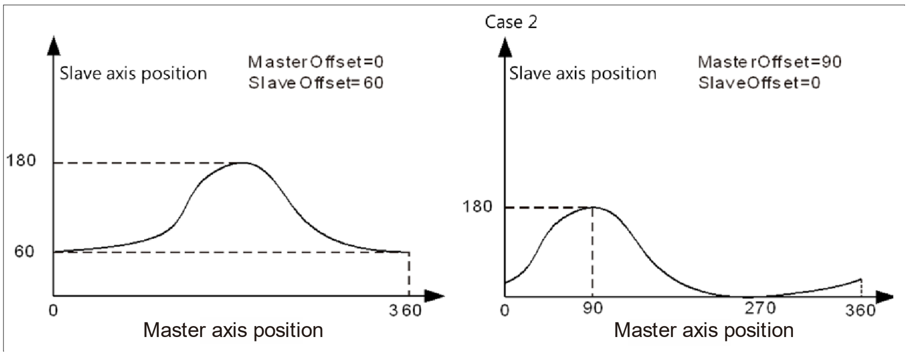
Situation 3: When master scaling ratio is 1, slave scaling ratio is 0.5 and offset for both axes is 0, the slave position that corresponds to the master position is half the amount of pre-planned measurement.

Situation 4: When master scaling ratio is 2, slave scaling ratio is 1 and offset for both axes is 0, the master position that corresponds to the slave position is twice the amount of pre-planned measurement. From CAM phase perspective, the Master CAM is twice the amount of pre-planned measurement, meaning the Master CAM changes from 360 to 180, while Slave CAM phase remains the same.

Situation 5: When master scaling ratio is 0.5, slave scaling ratio is 1 and offset for both axes is 0, the master position that corresponds to the slave position is half the amount of pre-planned measurement. From CAM phase perspective, the Master CAM is half the amount of pre-planned measurement, meaning the Master CAM changes from 360 to 720, while Slave CAM phase remains the same.

The scaling ratio for master and slave is 1 and the CAM effect when executing actual master and slave offset. The master offset means that the position curve of actual axis positions moves horizontally during CAM execution; the slave offset means that the position curve moves vertically during CAM execution.

Situations:

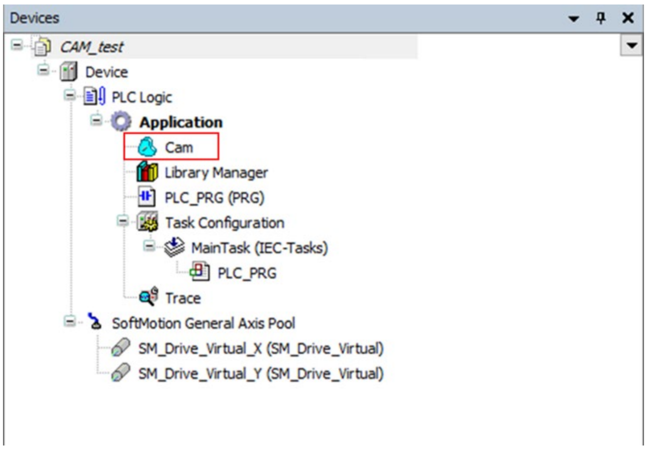


Situation 1: When the scaling ratio of master and slave is 1, the master offset is 0 and the slave offset is 60, the slave position that corresponds to the master position need to add 60 based on the pre-planned measurement. For instance, the master position is 180 and corresponds to the slave position that is 180 in CAM mechanism, but the slave position is 240 (240=180+60) during actual execution.

Situation 2: When the scaling ratio of master and slave is 1, the master offset is 90 and the slave offset is 0, the master position that corresponds to the slave position offsets by 90 (adding offset value) based on the pre-planned measurement. For instance, the master position is 180 and corresponds to the slave position that is 180 in CAM mechanism. However, during actual execution, the master position is 90 and corresponds to the slave position of 180, meaning the slave position corresponds to the master position that is 180 (180=90+90) in pre-planned CAM mechanism.

- **CAM table**

By selecting CAM in **DIADesigner-AX** project tree, you can edit the CAM curve that determines the operating characteristics of CAM.



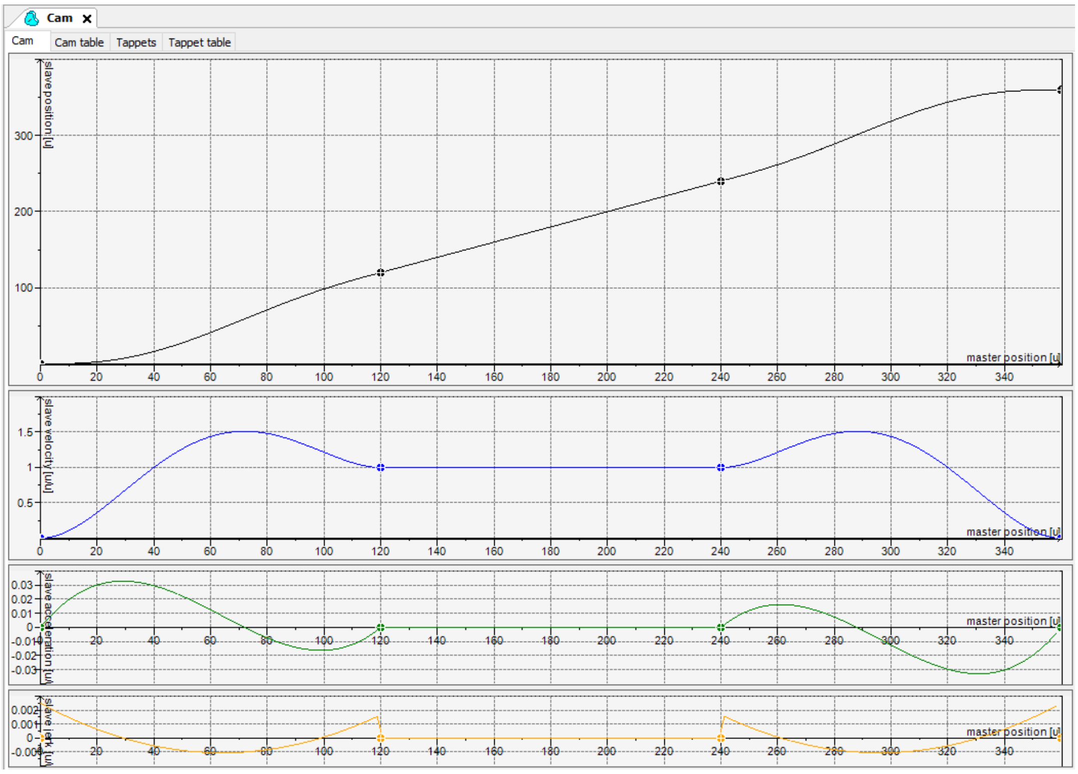
- **Features of CAM table**

◆ Direct observation on the changes of CAM curves corresponds to the slave motion range, velocity, acceleration, and jerk at any time.

◆ The master starting coordinate by default begins from 0 and ends at 360. You can make modifications based on the real physical range.

- **Editing method for CAM curves**

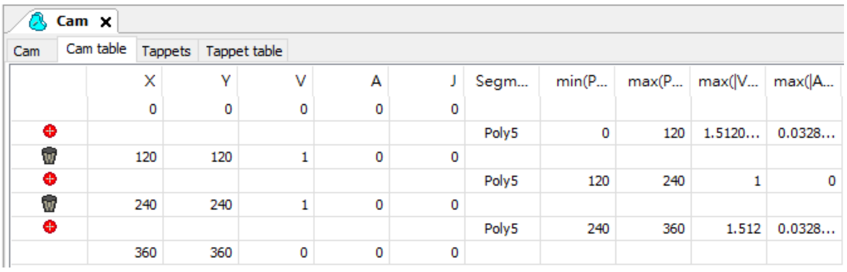
◆ **Graph editing on DIADesigner-AX**



You adopt graphs to edit CAM table, horizontal coordinates as master position and master axis length to determine CAM operating range. The four kinds of curves shown in the page (see below) represent position, speed, acceleration and jerk. When designing CAM, position and speed curves can be used to make motion range adjustment, while adjusting acceleration curve allows stabilization in movement.

◆ **CAM table editing on DIADesigner-AX**

Besides using graphs for editing, the CAM table is also used to modify any increase or decrease on critical points and positions directly on the CAM table page.



	X	Y	V	A	J	Segm...	min(P...	max(P...	max(|V...	max(|A...
	0	0	0	0	0					
⊕						Poly5	0	120	1.5120...	0.0328...
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.0328...
	360	360	0	0	0					

◆ **Programming editing**

You can also adopt programming to make modifications regarding critical points on the CAM table. To modify a program (see below), the starting position (master, slave) of CAM table moves from (0,0) to (0, 30), but the image displayed in the software will not be changed.

For using DMC_CamWritePoint function block to modify CAM table in programming, descriptions are as follows:

⓪Specify a CAM table

①Execute function blocks

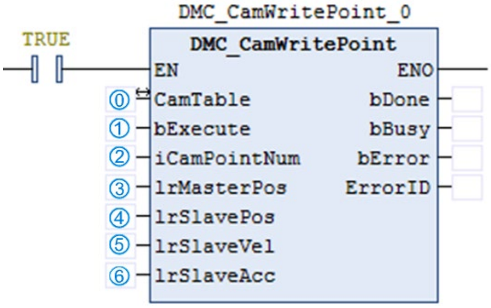
②Choose the CAM point number to read

③Position of the CAM master axis

④ Position of the CAM slave axis

⑤Velocity of the CAM slave axis

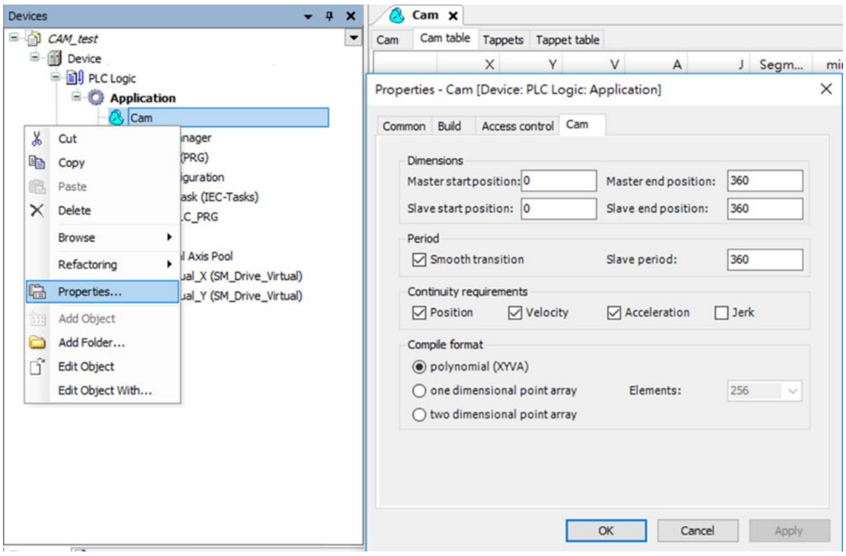
⑥Acceleration of the CAM slave axis



***Note:** For more details on function blocks, please refer to **AX Series Motion Controller Manual**.

- **CAM table properties:**

In Properties window, you can adjust the properties regarding CAM table, for example, the starting and ending position of master and slave, periodic parameters setups, required curve continuation and editing formats.



Properties - Cam [Device: PLC Logic: Application]

Dimensions

Master start position: 0 Master end position: 360

Slave start position: 0 Slave end position: 360

Period

☑ Smooth transition Slave period: 360

Continuity requirements

☑ Position ☑ Velocity ☑ Acceleration ☐ Jerk

Compile format

● polynomial (XYVA)

○ one dimensional point array Elements: 256

○ two dimensional point array

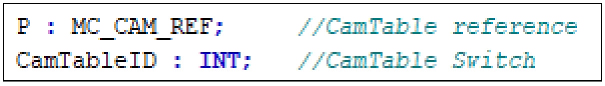
- **Steps on using E-CAM:**

1. CAM table configuration: setup master range, slave range, create starting point, ending point and other critical points as well as curve type adjustments.
2. Use instruction MC_CamTableSelect to connect configured CAM table with the actual one and receive CAM ID to be used for later instructions.
3. After receiving CAM ID, use instruction MC_CamIn to execute engagement for specified master and slave.
4. Use instruction MC_Camout for the master and slave relationship disengagement. For synchronous movement, use instruction MC_Stop and MC_Halt on slave axis for disengaging synchronous relation between master and slave.

- **Switching of CAM tables:**

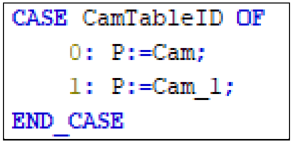
When CAM table is operating, please refer to MC_CAM_REF for switching the CAM table of MC_CamTableSelect.

- Declaring variables



```
P : MC_CAM_REF;        //CamTable reference
CamTableID : INT;      //CamTable Switch
```

- Switching of CAM tables

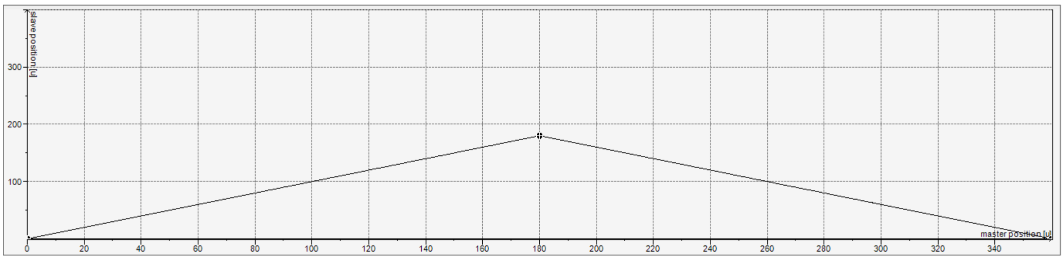


```
CASE CamTableID OF
    0: P:=Cam;
    1: P:=Cam_1;
END_CASE
```

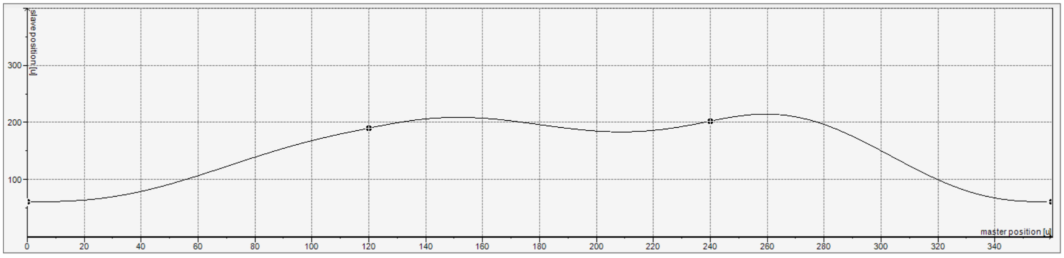
In the programming examples shown above, use the switching of CamTableID to change MC_CAM_REF to achieve switching of multiple CAM tables.

Below are the two CAM tables:

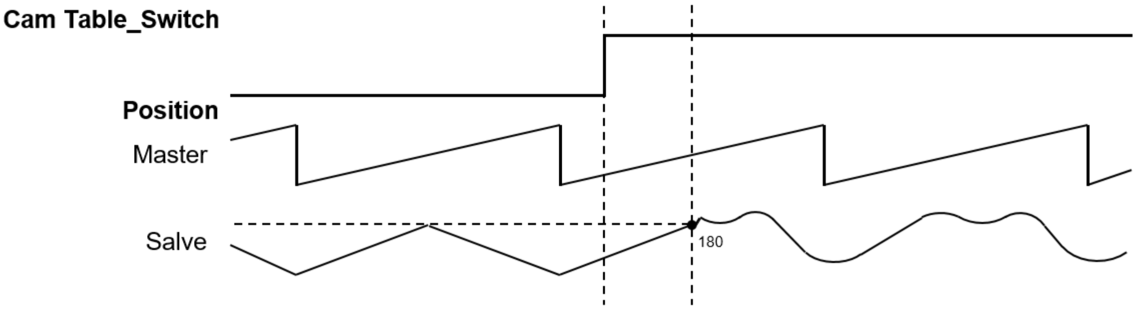
➨ The first Cam table



➨ The second Cam table (Cam_1)



- Timing diagram for switching of Cam table.



When switching Cam tables, the slave moves along the motion path based on the first CAM table until the master position reaches to the next critical point and then starts to follow the motion path based on the second.

7.7.3 Velocity Control

There are three kinds of motion control modes, the Cyclic Synchronous Position (CSP), the Cyclic Synchronous Velocity mode (CSV), and Profile Velocity mode (PV).

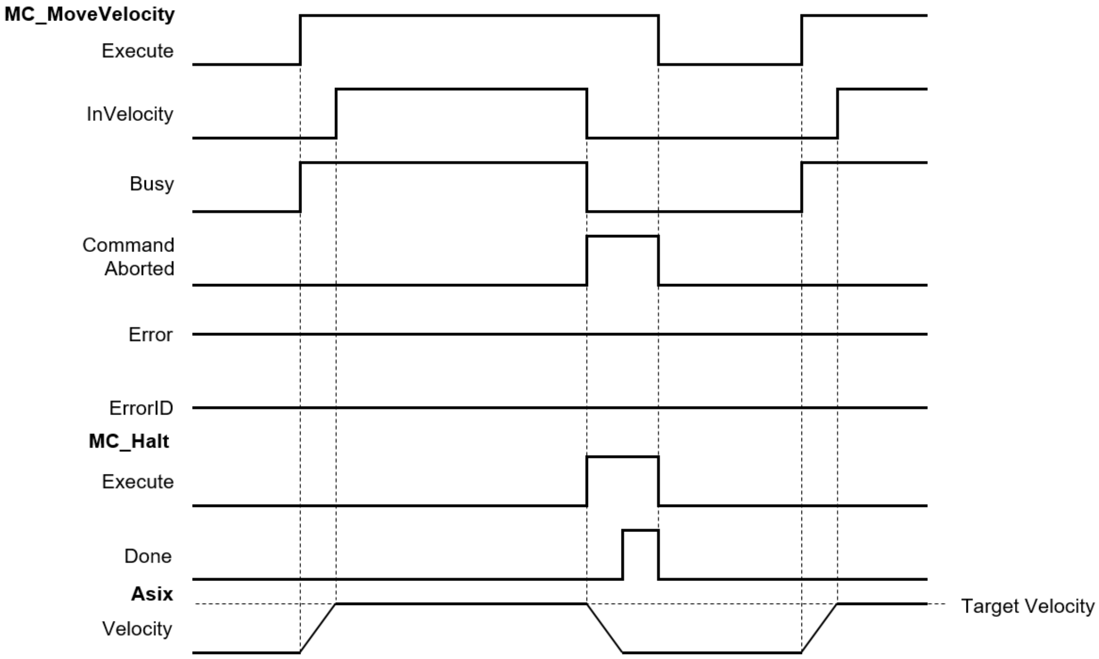
7.7.3.1 CSP Mode

The CSP mode is described as cyclic synchronous position in section 7.7.2.1. Under this mode, the controller can calculate the position of a command per cycle based on assigned velocity (including acceleration, deceleration and jerk), and then send this command to the servo for execution.

In CSP mode, when external interference causes the current servo position to lag behind the position command of the controller, as a result vibrations may appear to compensate these position errors.

The use of motion instruction MC_MoveVelocity can execute velocity and motion control in CSP mode. When executing, the axis state enters continuous_motion state. The specified acceleration, deceleration and jerk can be set during velocity adjustment (before reaching the specified velocity or during buffering). MC_Stop and MC_Halt or other motion instructions can be used to stop the control mode when needed.

The following diagram uses MC_MoveVelocity to proceed with velocity and motion control, as well as MC_Halt for discontinuity in the timing diagram:

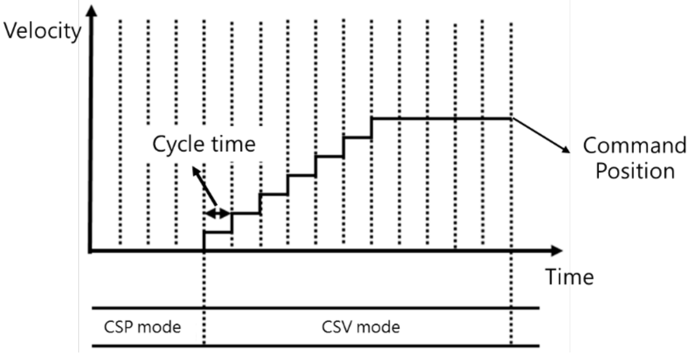


Set the velocity to 0, and though the current movement is static, the system will be in continuous_motion status.

In AX series, use instruction MC_MoveVelocity to execute velocity control for single axes in CSP mode. Please refer to **AX Series Motion Controller Manual** for more function block details.

7.7.3.2 CSV Mode

The CSV mode is the cyclic synchronous velocity mode (CSV). Under this mode, the controller can calculate the velocity for per cycle based on the specified velocity (including acceleration, deceleration and jerk) then send this command to the servo for execution.

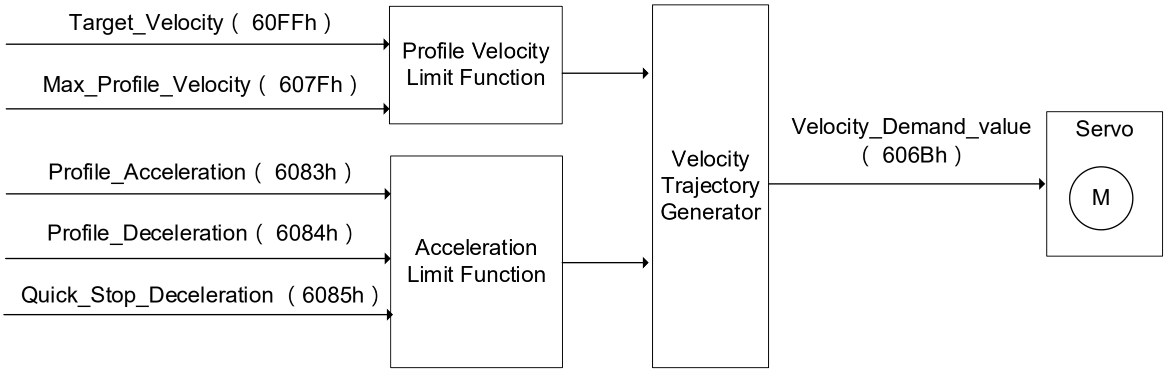


Despite external interference, cyclic velocity commands in CSV mode are sent to servos that are unlikely to cause vibrations due to compensating positions found in CSP mode.

In AX series, use instruction MC_ VelocityControl to execute velocity control for single axes in CSV mode. Please refer to **AX Series Motion Controller Manual** for more function block details.

7.7.3.3 Profile Velocity Mode

Under this mode, velocity trajectory generator performs motion path planning based on conditions assigned by master devices, such as velocity command and acceleration as well as deceleration.



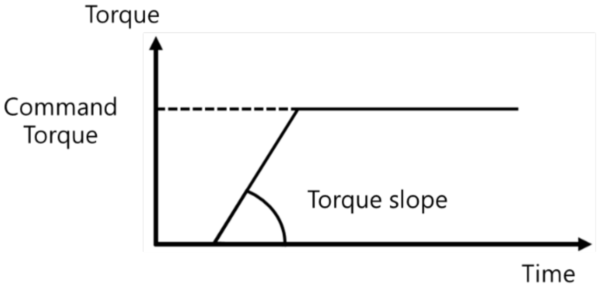
***Note:** Profile Velocity mode is used for positioning axes.

7.7.4 Torque control

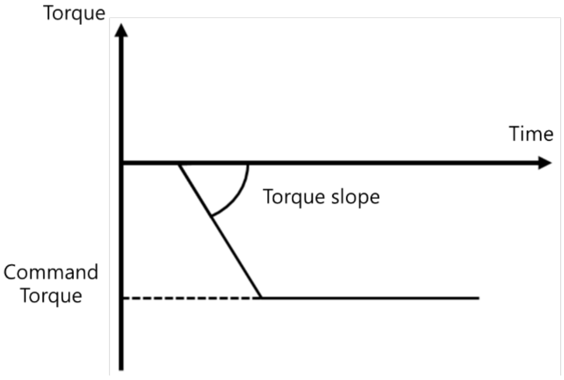
Torque control can be categorized into Cyclic Synchronous Torque mode (CST) and Profile Torque mode (PT).

- Profile Torque mode* (PT)

- Use DMC_TorqueControl to generate assigned torque output continuously through single axes.
- Notification
  - When using DMC_TorqueControl, switch the control mode to torque mode.
  - When using DMC_TorqueControl, the control mode switches to torque mode and cannot use function blocks regarding shifts or velocity. Use DMC_TorqueControl Enable instead of MC_Stop to stop motors.
  - Use the velocity of DMC_TorqueControl to set the maximum velocity limit for servo motors, which avoids high speed rotation as motor load declines in torque mode.
  - Adopt TorqueRamp to achieve the target torque value.
  - When Torque is bigger than 0 (Torque > 0), the motor operates in the positive direction.



- When Torque is smaller than 0 (Torque < 0), the motor operates in the negative direction.



Note:

*1: ASDA-A3-E Series V1.1165 or later supports Profile Torque Mode.

*2: ASDA-B3-E Series V1.0665 or later supports Profile Torque Mode.

7.7.5 Common Functions for Single-axis Control

The common functions for single-axis control are described in the following section.

7.7.5.1 Command Position

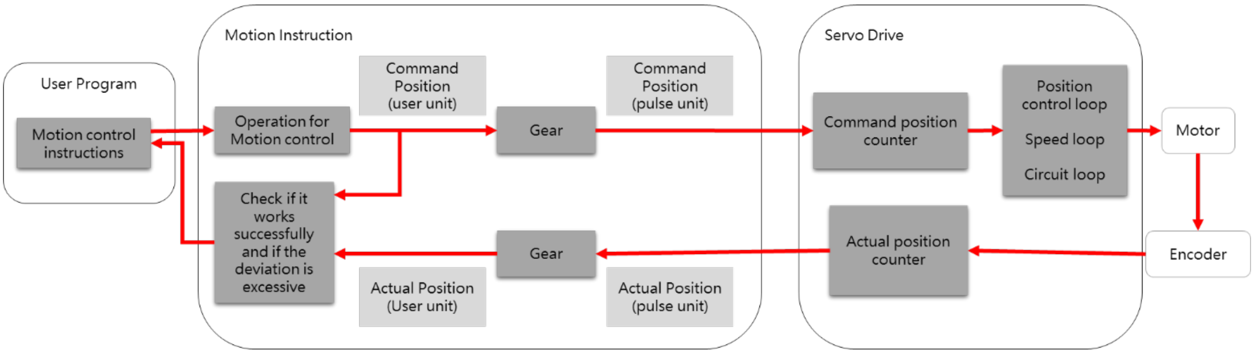
- Types of positions

The axis motion function blocks adopt the following two types of positions.

Type of position	Meaning
Command position	The position that MC function block outputs to control an axis.
Actual position	The position as feedback from the servo drive*

***Note:** For a virtual axis, there is no position feedback from the servo drive, so the command position will replace the actual position.

The following figure shows the relationship between the command position and actual position:



A comparison between the command position and actual position:

Item	Command position	Actual position
Count mode	Linear axis / rotary axis	The same count mode setting as in command position
Command unit	Length unit (m, mm, inch…) / angle unit (degree) / …	The same unit setting as in command position
Software limits	Set the range limit for MC function blocks	The same range limit setting as in command position
Positioning	Change to any desire position within the range limit	The same position setting as in command position, but position lag may appear*

***Note:** Due to the settings of servo mechanism, the so-called position lag may be generated between command and actual positions. As the motion velocity increases, position lag also increases slightly. When limiting the lag, you can adjust axis setting to monitor the position lag and set operation for position lag being too large. For virtual axis, the actual position equals to the command position and position lag does not exist.

Descriptions for the relevant parameters are as follows:

- **Position unit**

The unit refers to "command unit."

- **Position lag**

Setting	Value	Meaning
Position lag supervision	Deactivated	Position lag not checked
	Disable drive	When position lag exceeds the limit, the axis is in servo off.
	Do quick stop	When position lag exceeds the limit, the axis is in quick stop.
	Stay enabled	When position lag exceeds the limit, the axis maintains servo on.
Lag limit [u]	LREAL	Allowable lag limit

Besides deactivated setting value, when other settings exceed the lag limit, the axis reports the error SMC_ERROR.SMC_DI_POSITIONLAGERROR.

- **Software limits**

Setting	Value	Meaning
Software limits Activated	Checked / Unchecked	Whether or not software limits are activated.
Negative [u]	LREAL	Negative software limit
Positive [u]	LREAL	Positive software limit

- **Description of positions in MC function blocks**

Please take note of the following input variables with two different interpretations that are related to positions in MC function blocks:

Item	Meaning
Position	Target position (absolute position)
Distance	Moving distance (relative position)

- **Monitoring positions**

To observe change in position, you can focus on the following two axis variables (AXIS_REF_SM3 type) for monitoring:

Variable name	Position type	Data type
.fSetPosition	Command position	LREAL
.fActPosition	Actual position	LREAL

7.7.5.2 Velocity Command

- **Types of velocity**

The following two types of velocity are used in MC function blocks.

Position type	Meaning
Command velocity	The velocity in which MC function block outputs for axis control
Actual velocity	The velocity based on the actual feedback position of servo drives at each point in time*

***Note:** For virtual axis, there is no position feedback from the servo drive, so the command position will replace the actual position.

- **Velocity unit**

The velocity unit is “command unit/s.”

- **Velocity ramp type**

Setting	Value	Meaning
Velocity ramp type	Trapezoid	A trapezoidal velocity ramp (Each section is constant acceleration)
	Sin^2	The velocity ramp equals to sin^2 function (acceleration ramp is fixed)
	Quadratic	Acceleration ramp with trapezoidal profile (jerk limited)
	Quadratic (smooth)	Adopts the same meaning as in Quadratic, but with continuous S-curve velocity (jerk limited).

- **Description of velocity in MC function blocks**

The following input variable that is related to velocity in MC function blocks:

Item	Meaning
Velocity	Target velocity*

***Note:** Due to inadequate trajectory length, small acceleration and jerk as well as other factors, it is not possible to obtain the target velocity.

- **Monitoring velocity**

To observe change in velocity, you can focus on the following two axis variables (AXIS_REF_SM3 type) for monitoring:

Variable name	Position type	Data type
.fSetVelocity	Command velocity	LREAL
.fActVelocity	Actual velocity	LREAL

7.7.5.3 Acceleration and Deceleration Command

- **Types of acceleration**

The following two types of acceleration are used in the MC function blocks.

Position type	Meaning
Acceleration command	The outputs of MC function blocks to control axis acceleration
Actual acceleration	The acceleration calculated based on actual velocity

- **Acceleration unit**

The acceleration rates are in “command units/ s^2 “.

- **Axis settings related to acceleration**

(1) Types of acceleration waveform

Please refer to “7.7.5.2 Velocity Command- Velocity ramp type” for more information.

- **Description of acceleration in MC function blocks**

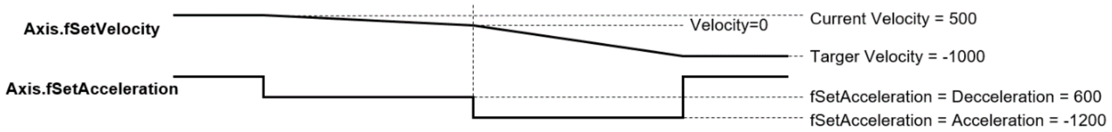
The following input variables that are related to acceleration/deceleration in MC function blocks:

Item	Meaning
Acceleration	Target acceleration*
Deceleration	Target deceleration*

***Note:** Due to inadequate trajectory length, small jerk and other factors, it is not possible to obtain the target acceleration or target deceleration.

According to standard acceleration and deceleration rates, if the demand for absolute value of current velocity decreases, deceleration rate is performed; if the demand for absolute value of current velocity increases, acceleration rate is performed.

For instance, when the current axis velocity is 500, the motion control instructions during execution is in the reverse direction (Velocity = 1000, Acceleration = 1200, Deceleration = 600). The following diagram shows the velocity and acceleration waveform:



- **Monitoring acceleration**

To observe change in acceleration, you can focus on the following two axis variables (AXIS_REF_SM3 type) for monitoring:

Variable name	Position type	Data type
.fSetAcceleration	Command acceleration	LREAL
.fActAcceleration	Actual acceleration	LREAL

7.7.5.4 Jerk Command

The jerk assigns the changes in acceleration or deceleration rate. When the jerk is specified, the velocity waveform is in S-curve (the ramp of acceleration increases or decreases, no jerk), which can reduce the shock on machines.

- **Types of jerk**

The following two types of jerk are used in the MC function blocks.

Position type	Meaning
Command jerk	The outputs of MC function blocks to control the axis
Actual jerk	The jerk that is calculated based on actual acceleration

- **Jerk unit**

The jerk is in “command units/s^3”.

- **Axis settings related to jerk**

(1) Types of jerk waveform

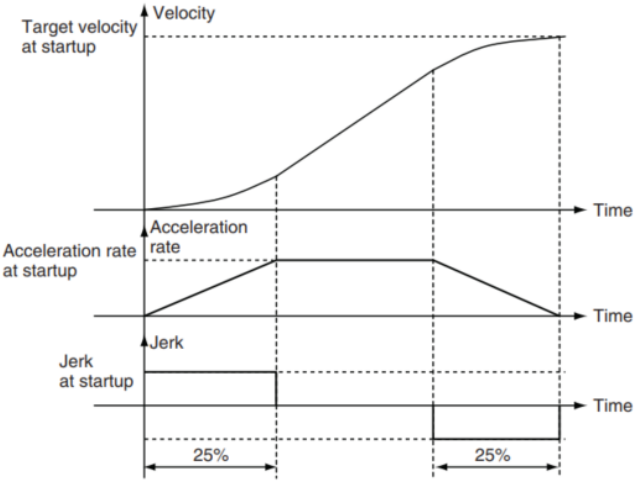
Please refer to “7.7.5.2 Velocity Command- Velocity ramp type” for more information.

- **Description of jerk in MC function blocks**

The following input variable that is related to jerk in MC function blocks:

Item	Meaning
Jerk	Target jerk*

***Note:** When velocity ramp type is trapezoid or in Sin^2, the setting values of jerk are not applied in the movement; when velocity ramp type is quadratic or quadratic (smooth), the jerk does affect the velocity ramp.



- **Monitoring jerk**

To observe change in jerk, you can focus on the following two axis variables (AXIS_REF_SM3 type) for monitoring:

Variable name	Position type	Data type
.fSetJerk	Command jerk	LREAL
.fActJerk	Actual jerk	LREAL

7.7.5.5 Axis Direction

The following situation requires specified operation directions:

- When the input value of Velocity is absolute for a constant-velocity motion, specifying the direction by Direction is required.
- When the axis is set to the rotary axis, the movement towards either positive or negative direction can reach the target position, therefore, specifying the direction is required.

See the following input variables that are related to the direction in MC function blocks:

Item	Setting	Meaning
Direction	negative	Motion operates in a negative direction
	shortest	Motion operates the shortest way (Only for the rotary axis)*
	positive	Motion operates in a positive direction
	current	Motion operates based on the current direction (Only for the rotary axis)
	fastest	Motion operates in the fastest way (Only for the rotary axis)*

***Note:** The concepts of being the shortest (moving distance) and fastest (moving time) are similar but not completely the same, please refer to the following example:

- Setup:

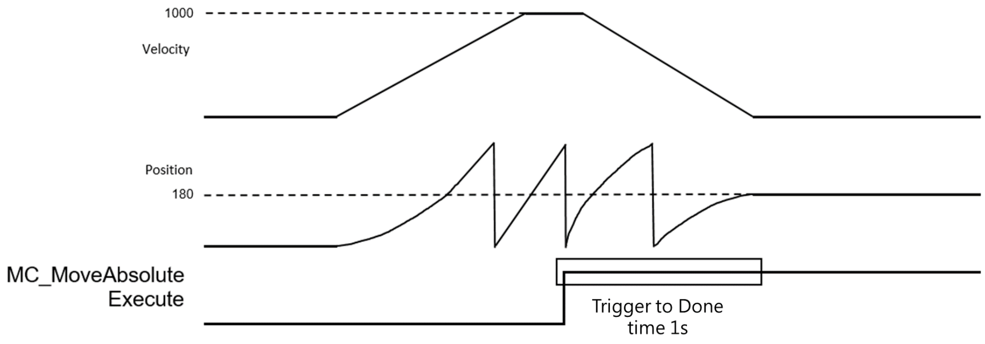
  Set axis as rotary axis, range 360。
  Set velocity ramp type of axis as Trapezoid.

- Procedure:

  Use MC_MoveVelocity to execute a constant velocity motion. (Velocity=1000)

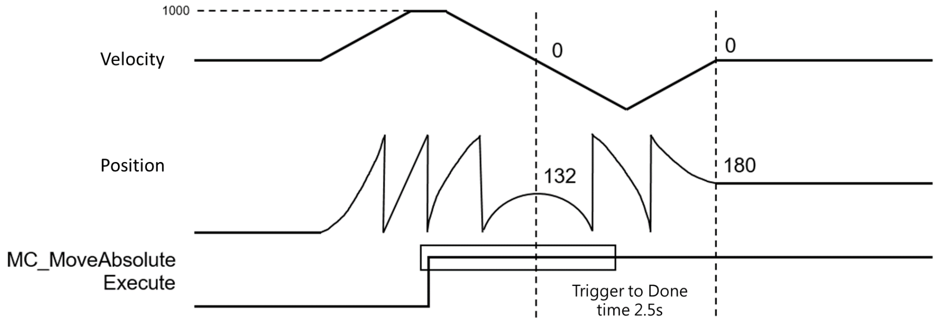
  When the motor reaches 350 and velocity reaches 1000, execute MC_MoveAbsolute with 2 different direction settings

  (1) Execute MC_MoveAbsolute (Position=180, Velocity = Acceleration = Deceleration = 1000, Direction = fastest)



When MC_MoveAbsolute.Execute triggers, the system determines the shortest way to reach position 180 is to move in the positive direction and decrease velocity to 0. The process takes about 1 sec.

(2) Execute MC_MoveAbsolute (Position = 180, Velocity = Acceleration = Deceleration = 1000, Direction = shortest)



When MC_MoveAbsolute.Execute triggers, the system determines the shortest way to reach position 180 is to move in the negative direction (350 – 180 = 170). However, since the process requires velocity to be in reverse, therefore, more turns are included. The process takes about 2.5 sec.

7.7.6 Axis Group Control

An axis group consists of more than one axis configured via DIADesigner-AX. Up to six axes can be supported for linear axes, while for rotary axes, three axes are supported with three extra axes as the follower axes.

7.7.6.1 Linear Interpolation

TransitionMode: The resulting noises and vibration of machines may occur if the trajectory of interpolation changes while in motion. By using the input variable "TransitionMode," the chances of the above situation will be minimized.

- **Available transition modes**

Mode	Description
None	No effects (default)
Overlap	Continued by combining the deceleration of the previous motion and the acceleration of the current motion.
SingleAxis	Continues according to the settings in Blending Mode, specifically for the single axis in a group

- **Supported buffer modes**

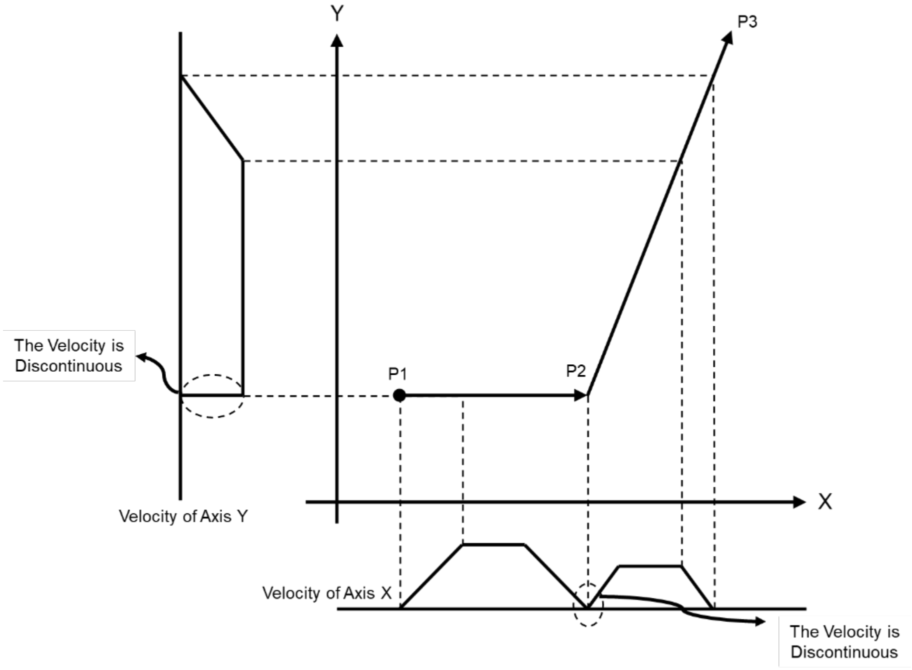
Mode	Aborting	Buffered	Blending Low	Blending Previous	Blending Next	Blending High
None	A	A	N	N	N	N
Overlap	A	A	D	D	D	D
SingleAxis	A	A	A	A	A	A

A = Supported

N = Not supported

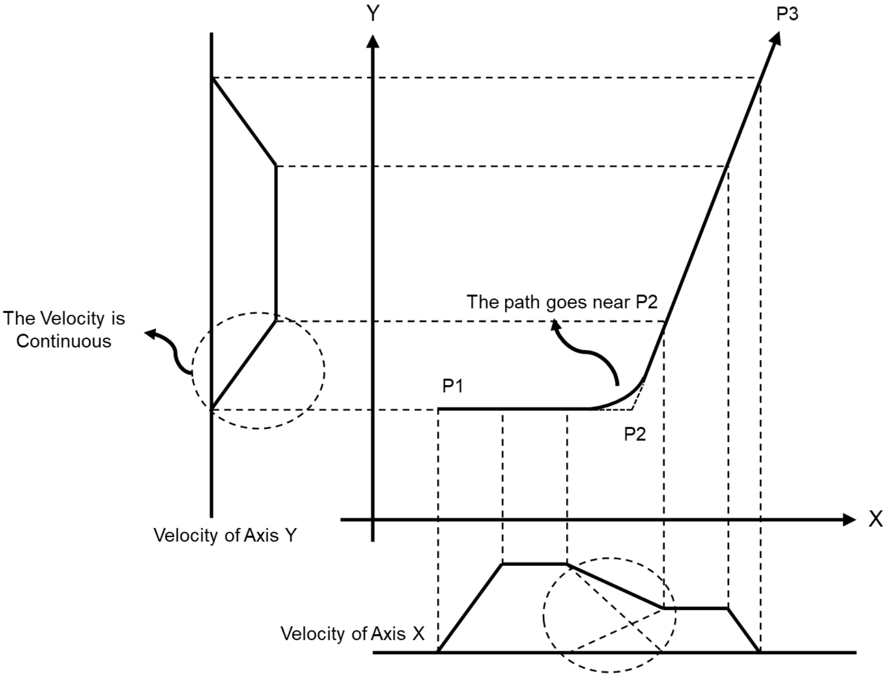
D = Continued but different from the effects of settings set in Blending Mode for the single axes in a group.

- **TransitionMode:** For the below situation, set the mode to None or Overlap, and then choose buffered.

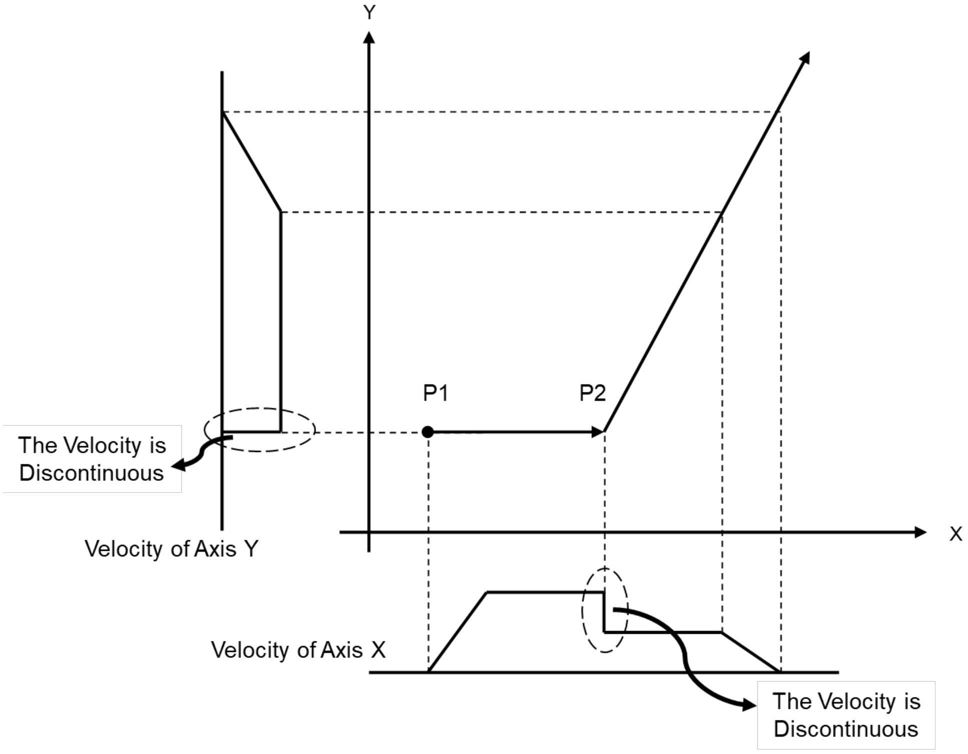


- **TransitionMode:** For the below situation, set the mode to Overlap, and then choose Blending.

Plan with reference to acceleration and deceleration given to the motion function block of each axis group.

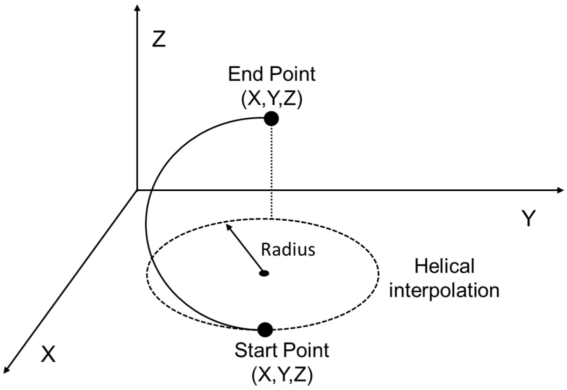


- **TransitionMode:** For the below situation, set the mode to SingleAxis, and then choose Blending.



7.7.6.2 Circular Interpolation

Circular movements can run in the three main planes of the spatial coordinate system, only using X, Y, Z axis and three additional follower axes.

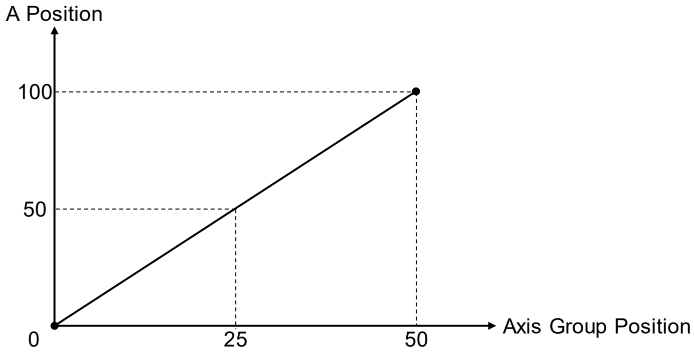


- **Concept of follower axes:**

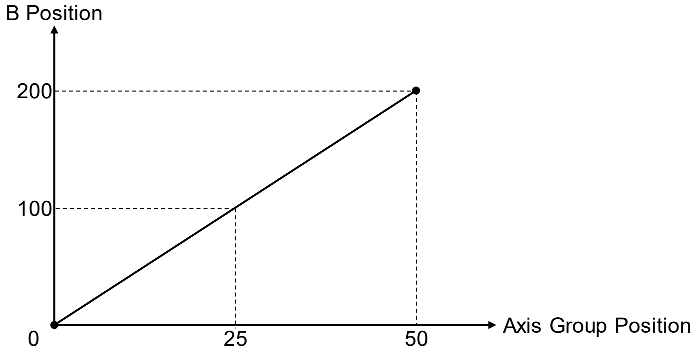
  Follower axes A, B, C move in a proportional and synchronized motion as axes X, Y, Z are moving.

  The axis group moves to position (30, 40, 0) with the start point of 0, and the combined moving distance is 50 while follower axes move to position (100, 200, 300). The synchronized movement between the axis group and follower axes is shown as following figures.

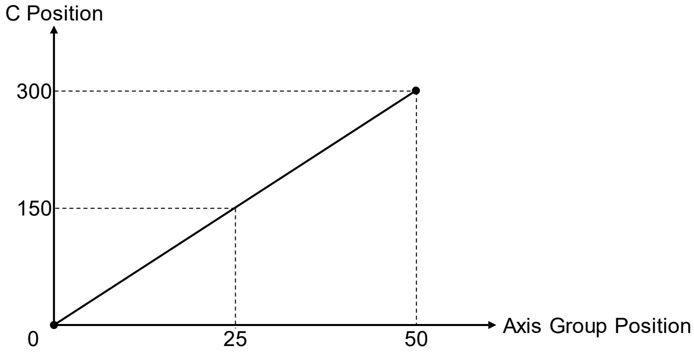
  - Follower A



  - Follower B



  - Follower C



***Note:** When the axis group is not in motion, the input velocity given to the axis group function block is used for the follower axis with the longest distance. At the same time, other follower axes move in synchronized motion based on the proportion of distances.

7.7.6.3 Group Stop Command

There're two different ways to stop an axis group motion:

- Programming stop

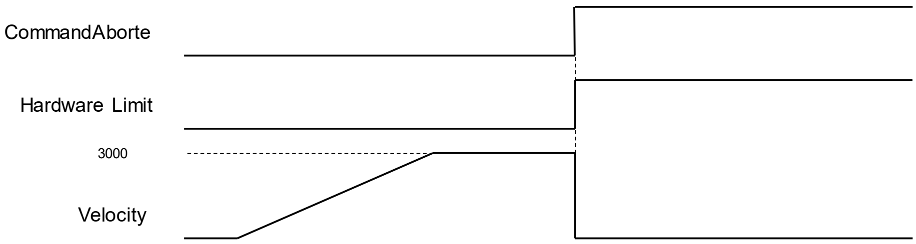
  Use DMC_GroupStop in the programming to decelerate the moving axis group to a stop. Then the group state switches to GroupStopping, and no motion instruction can be executed under this status.

  The velocity for a deceleration stop must be set to the lrDeceleration pin.

- Error stop

  As soon as an error occurs in a group motion, the axis group stops operating.

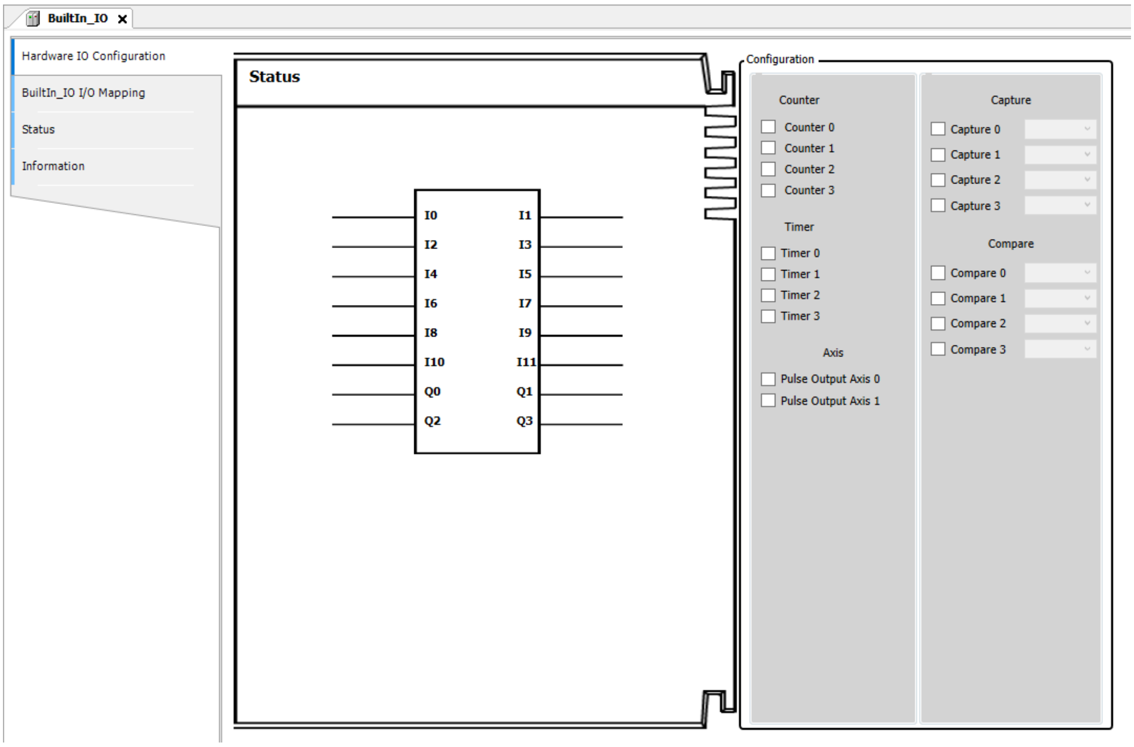
  For example, Hardware Limit is reached while the axis group is moving. The velocity drops to zero as a result of the output CommandAborted.



7.7.7 High-speed IO

The chapter contains information regarding CPU with IOs for configuration and parameter settings.

7.7.7.1 IO Configurations

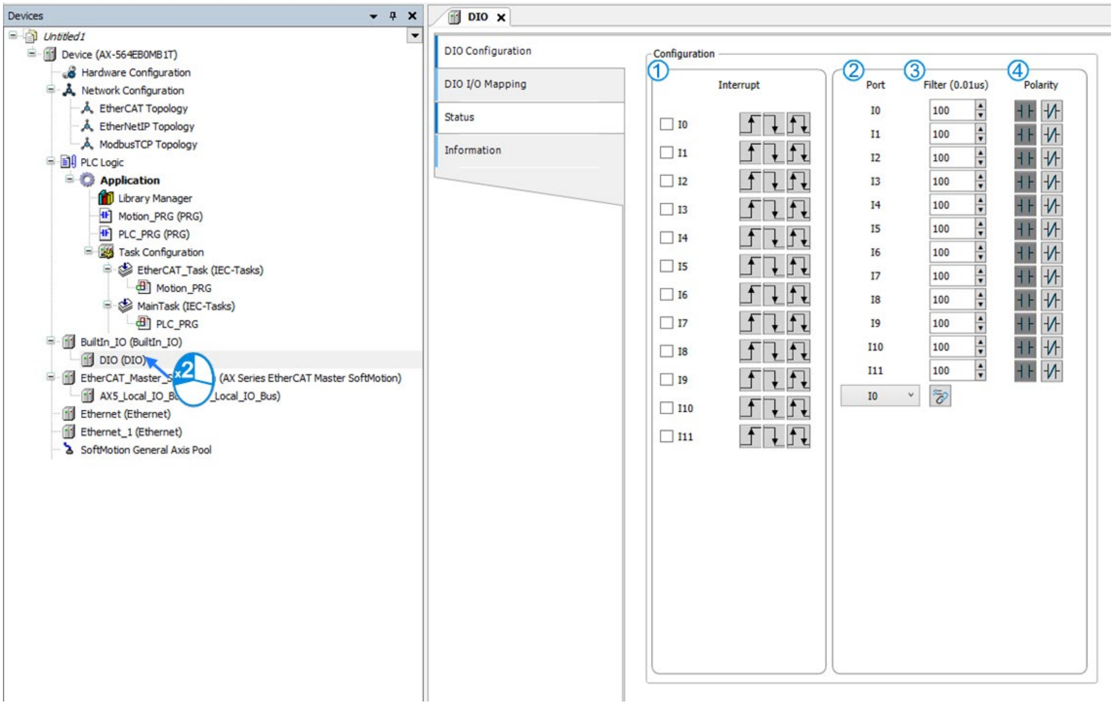


- **DIO:** Set functions including interrupt, filter and polarity. Refer to section 7.7.7.2 for more information.
- **Pulse Encoder:** Set functions including high speed counter variables, count modes, enable or disable Z phase signal as well as declare high speed timer variables. Refer to section 7.7.7.3 for more information.
- **Capture/ Compare:** Declares variables regarding high speed capture and compare. Refer to section 7.7.7.4 for more information.
- **Pulse Output:** Set functions including pulse output, direction and homing mode. Refer to section 7.7.7.5 for more information.

7.7.7.2 DIO Settings

The section describes setting functions including interrupt, filter and polarity of IOs in DIO device.

Double-click on “DIO” to enter the configuration page.

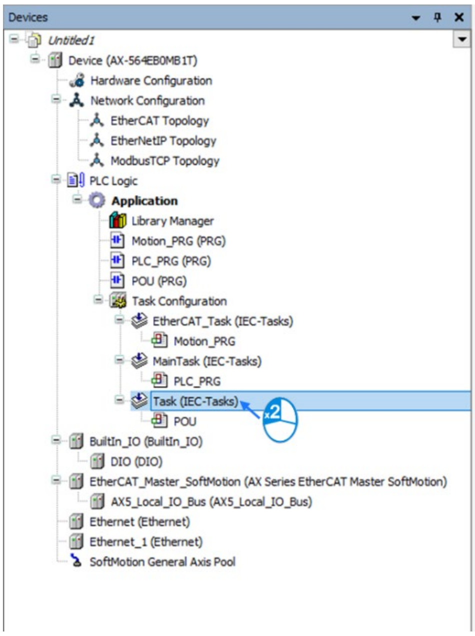


- **Configuration**

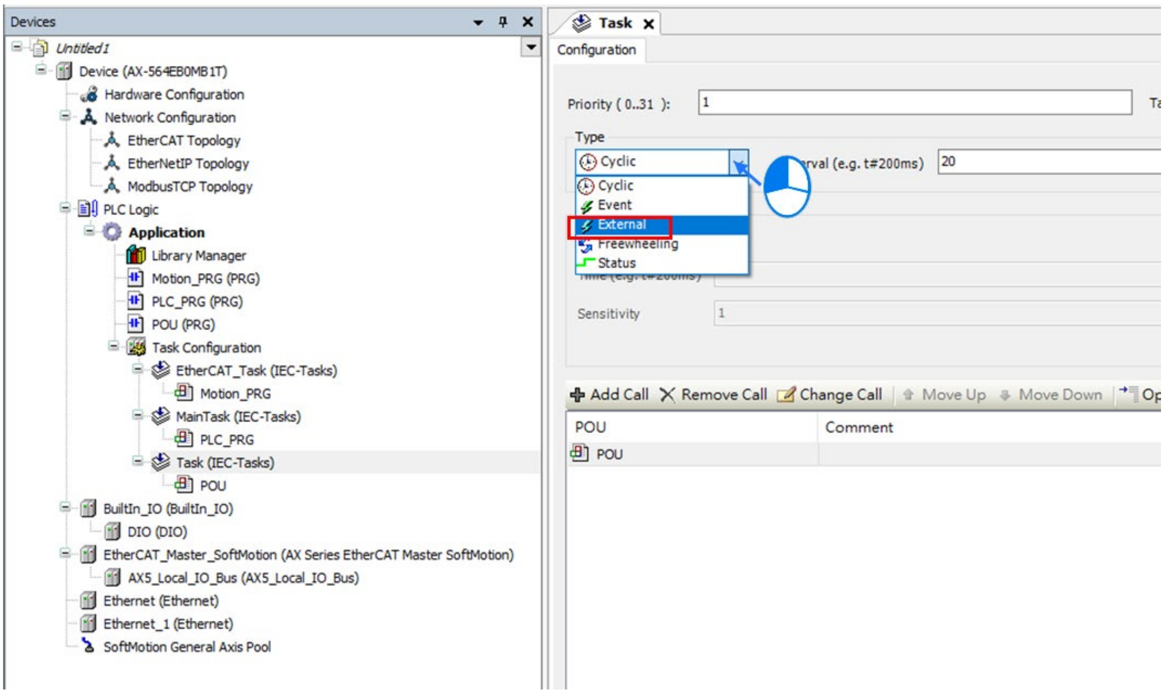
Function	Description
① External Interrupt Setting*	☐ Default value
	☑ Activate external interrupt
	↑ When external interrupt is activated, set input signals as rising edge.
	↓ When external interrupt is activated, set input signals as falling edge.
	↑↓ When external interrupt is activated, set input signals as rising and falling edge.
② Port	Port number
③ Filter	100 Set filter time (us), setting range is from 0 to 100000000. The default is 100us.
④ Polarity	┤├ Set input polarity. The default is contact A .
	┤/├ Set input polarity, The default is contact B.

- **IO interrupt mode setting**

  - After activating the interrupt function on DIO setting page, click on “Task” to proceed.



  - Enter Task configuration page and choose “External” from the drop down list for Type.

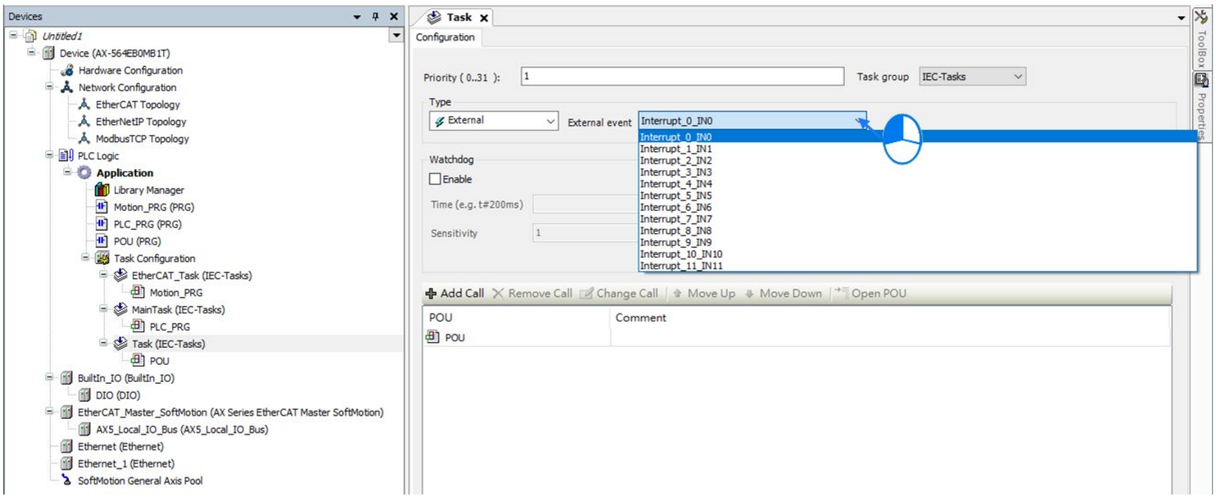


Priority (0..31): 1

rval (e.g. t#200ms) 20

Sensitivity 1

- Then choose the corresponding interrupt contact from the drop down list of External event.



Priority (0..31): 1

Task group: IEC-Tasks

Type: External

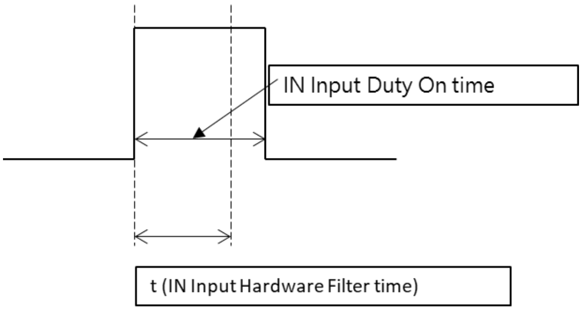
External event: Interrupt_0_IN0

Watchdog: ☐ Enable

Time (e.g. t#200ms)

Sensitivity: 1

- The setting value for hardware filter time is smaller than IN input duty-on time as shown below:



For AX-5 Series PLC CPU: The input range for hardawre filter is from 1 to 50,000,000, unit as 0.1 μs.

- The relation between filter frequency and filter time:

Filter frequency*^{1} (Hz): Filter frequency= 1 / (2*t); t is the filter time setting value (unit: 0.01μs). When the input frequency is higher than the filter frequency range, signals are filtered.

The function focuses on the X input point used in DFB_Capture, DFB_Hcnt, DFB_HTmr, DFB_Compare and IO interrupt.

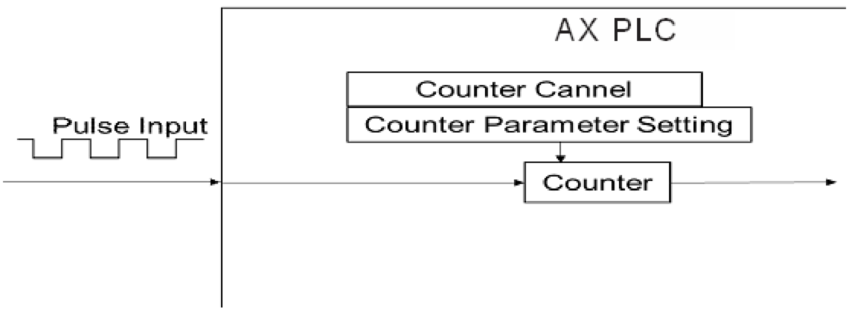
7.7.7.3 Pulse Encoder Settings

- **AX-5 Series PLC CPU provides following connecting interfaces for the pulse encoder**
  - AX-5 Series PLC CPU supports four sets of open-collector pulse inputs:
    Connecting through the euroblock (European-style terminal block) installed on the IO board; up to 4 sets of high-speed counters / timers can be used to count or time the pulse numbers or the frequencies of the encorder.

You need to select to enable pulse-type encoder function and set up the required parameters, and then through the configured hardawre channel to receive the encoder data.
This section below describes the pulse-type encoder (see below), the maximum total amount of high speed counters and high speed timers supported for AX-5 is 4 sets.

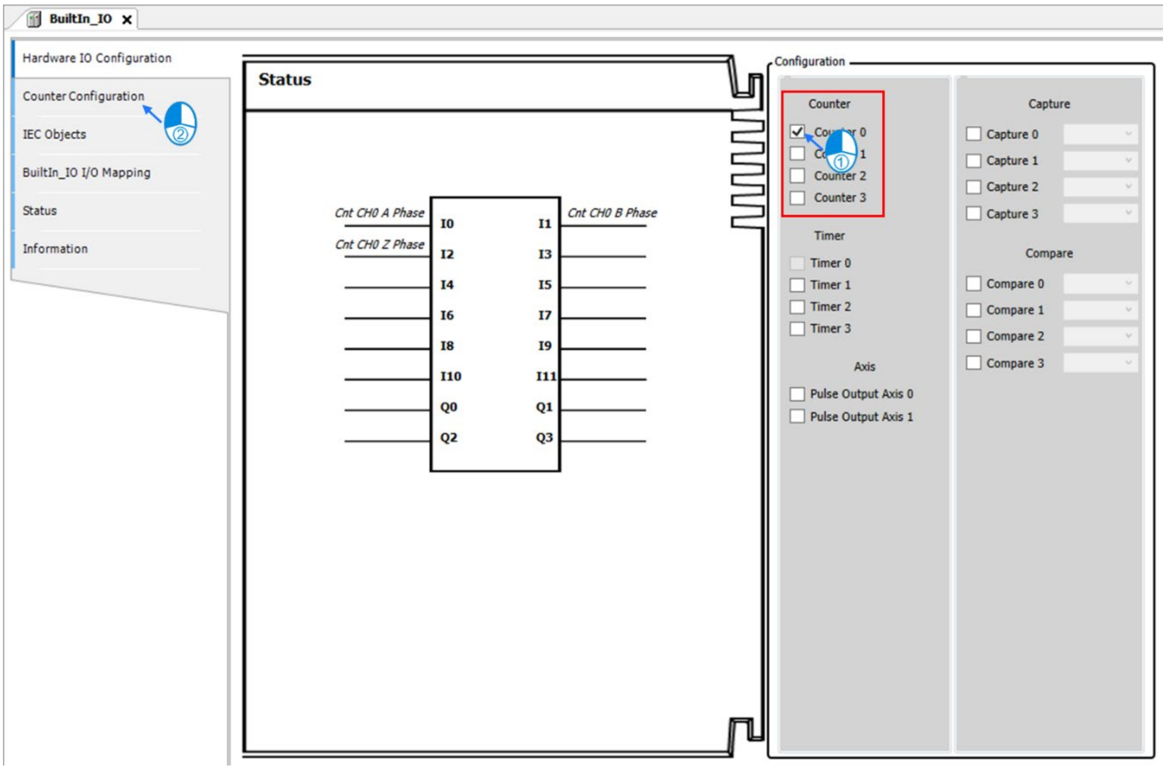
- **High speed counter (Cnt)**

  When selecting Cnt function in Hardware IO Configuration, you can also set up the high speed counter and encoder sections.

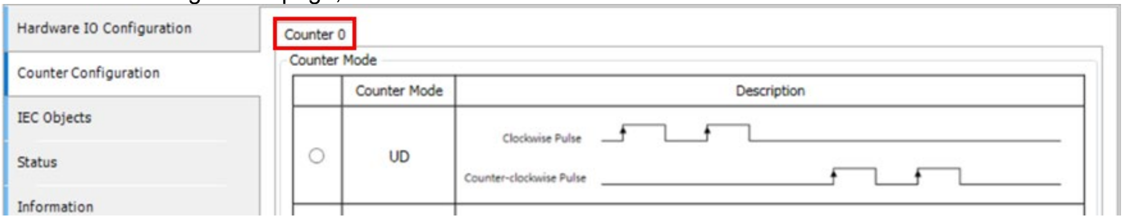


- **Enable hign-speed counter function**

  - A total of 4 counters are displayed on BuiltIn_IO page. Select Counter 0, and then click “Counter Configuration” tab.

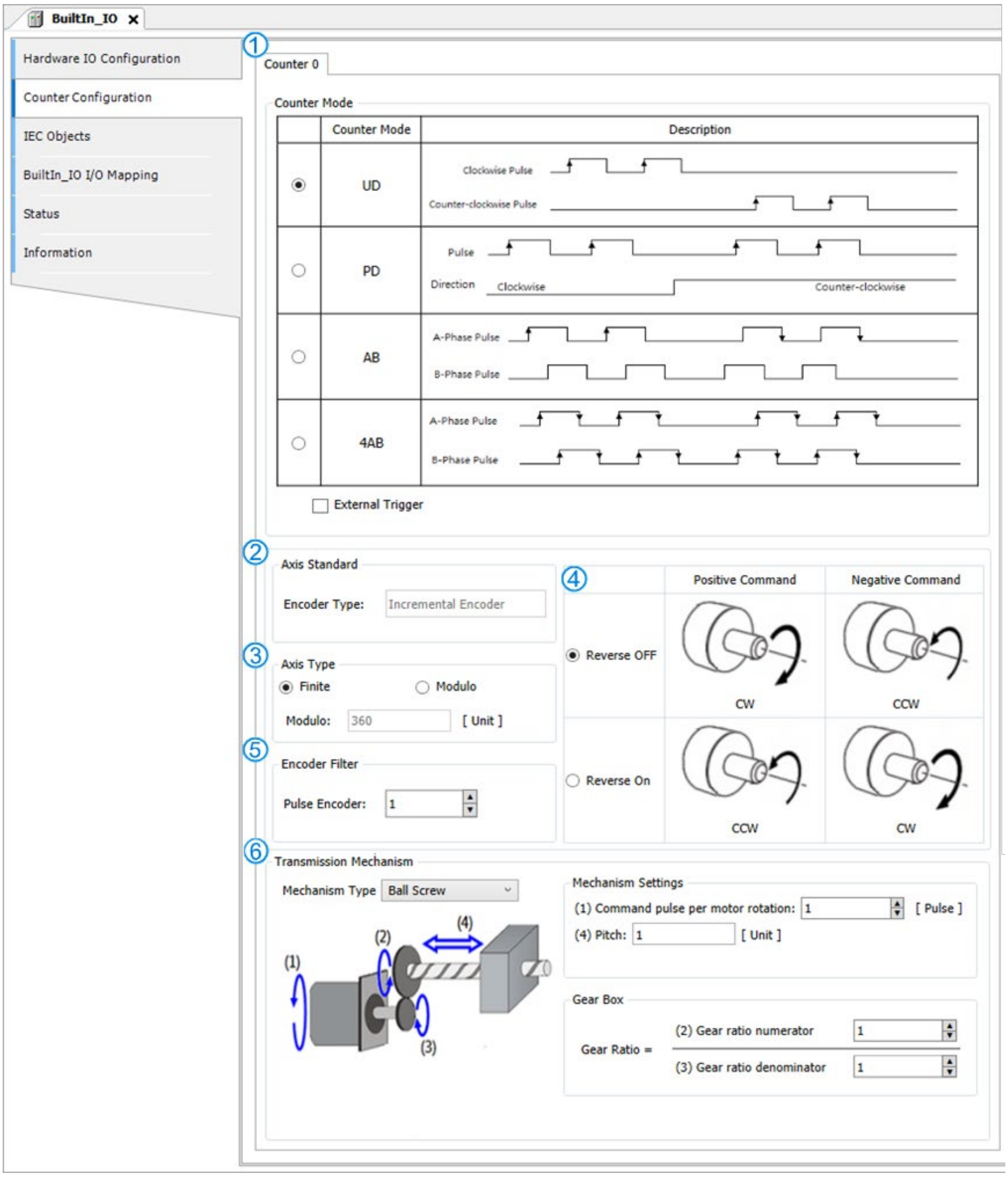


- On Counter Configuration page, choose Counter 0.



	Counter Mode	Description
○	UD	Clockwise Pulse Counter-clockwise Pulse

- Configure Counter-related settings on Counter Configuration page. Descriptions are as follows.



● UD
○ PD
○ AB
○ 4AB
☐ External Trigger

Encoder Type: Incremental Encoder

● Finite ○ Modulo

Modulo: 360 [Unit]

Pulse Encoder: 1

Mechanism Type: Ball Screw

(1) Command pulse per motor rotation: 1 [Pulse]

(4) Pitch: 1 [Unit]

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

● Reverse OFF
○ Reverse On

① Counter Mode

Pulse Counter Mode	Description
UD	Forward rotation pulse train and reverse rotation pulse train
PD	Pulse and direction
AB	A-phase and B-phase pulse
4AB	A-phase and B-phase pulse (4x)
External Trigger*	Activate Z-phase signals

Note: Refer to section 3.5 DFB_PresetValue from AX Series Standard Instruction Manual for more information on function blocks.

- ② Axis Standard

Name	Function	Setting value (default)
Encoder Type	Display the encoder type	-

- ③ Axis Type

Name	Function	Setting value (default)
Linear Axis / Rotary Axis	Set to be linear axis or rotary axis.	Linear Axis Rotary Axis (Linear Axis)
Modulo	Set the range of rotation for a turn	(360)

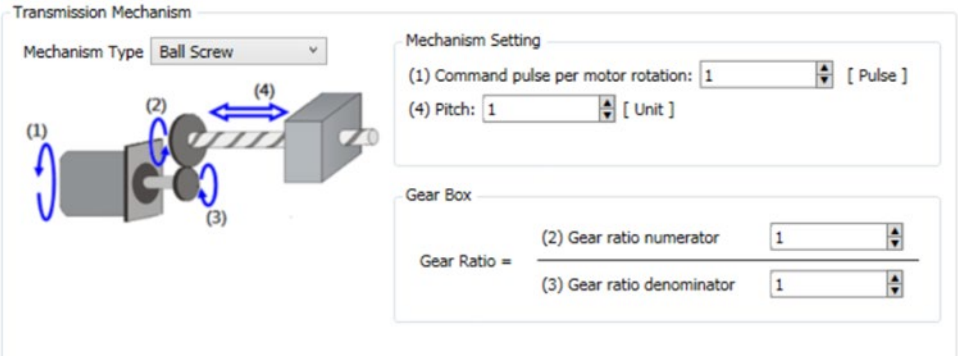
- ④ Positive / Negative Command

Name	Function
Reverse OFF / ON	Enable or disable reverse function for positive/negative command setting.

- ⑤ Encoder Fiter

Name	Function
Pulse Encoder	The setting value is 1 to 100.

- ⑥ Transmission Mechanism
  - ◆ Ball Screw



Transmission Mechanism

Mechanism Type: Ball Screw

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

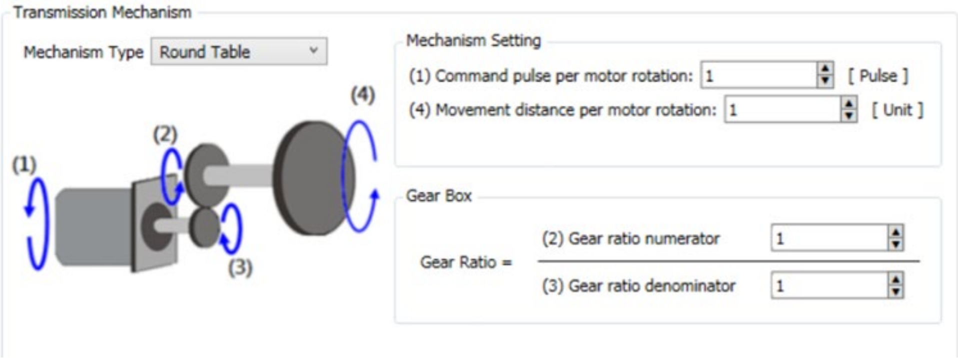
(4) Pitch: 1 [Unit]

Gear Box

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Pitch	The distance between screw threads

◆ Round Table



Transmission Mechanism

Mechanism Type: Round Table

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

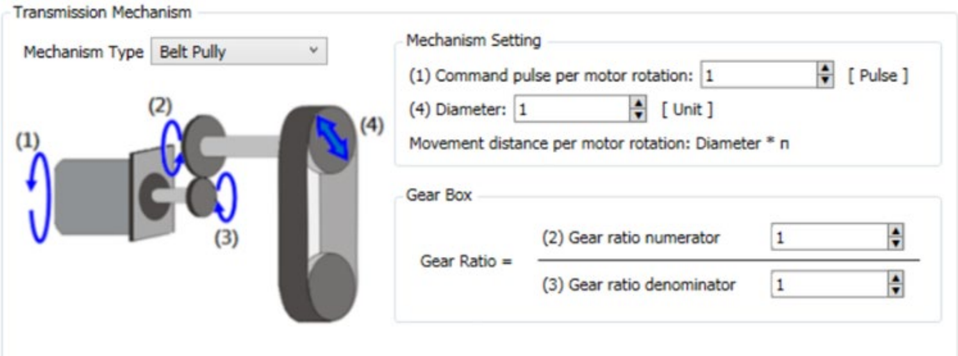
(4) Movement distance per motor rotation: 1 [Unit]

Gear Box

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Movement distance per motor rotation	Movement distance for one full motor rotation

◆ Belt Pully



Transmission Mechanism

Mechanism Type: Belt Pully

Mechanism Setting

(1) Command pulse per motor rotation: 1 [Pulse]

(4) Diameter: 1 [Unit]

Movement distance per motor rotation: Diameter * π

Gear Box

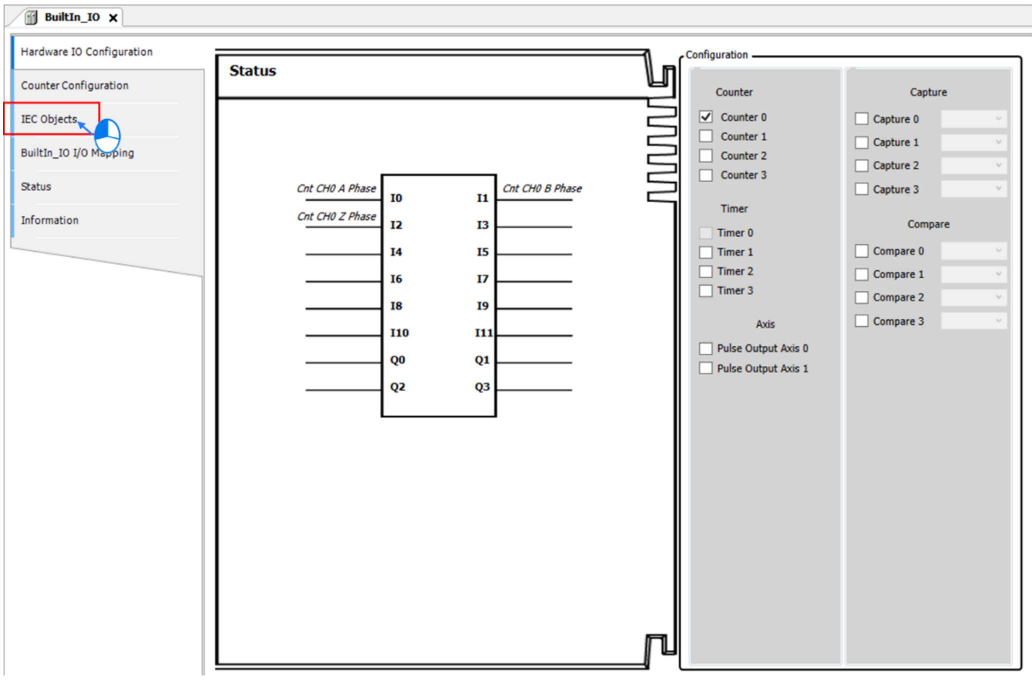
Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

Name	Function
(1) Command Pulse per motor rotation	The command pulse number for per motor rotation
(2) Gear ratio numerator	Numerator of gear ratio
(3) Gear ratio denominator	Denominator of gear ratio
(4) Diameter* (Movement distance per motor rotation: Diameter X π)	Diameter (Movement distance per motor rotation: Diameter X π)

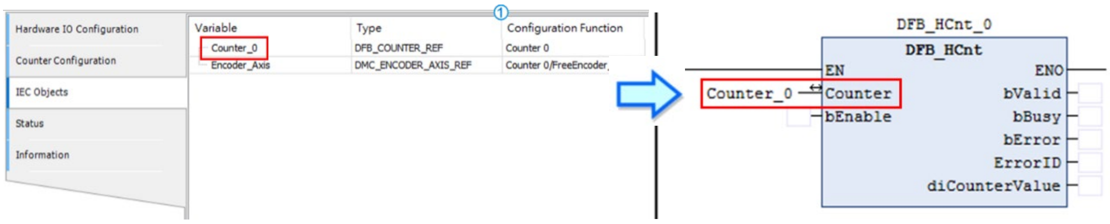
- **Use Counter in program**

The high-speed counter contains encoder axis variables that can be used for MC function blocks in POU.

- ■ Click on “IEC Objects” tab on BuiltIn_IO page.

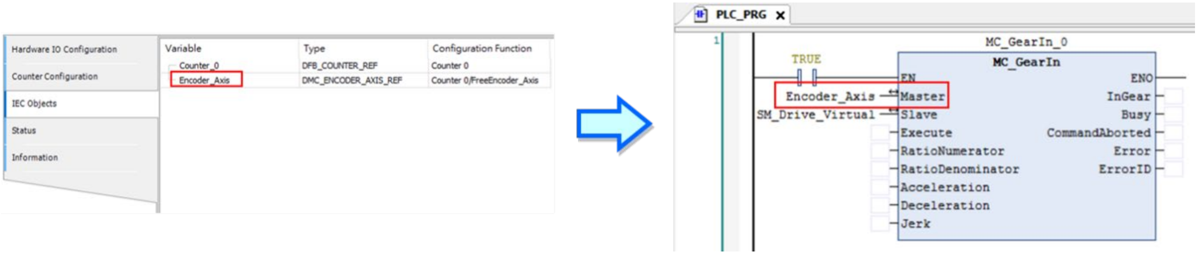


- ■ The column marked ① on the IEC Objects tab is the description of the configuration function of each variable. To enable counter function, the variable name Counter_0 needs to be input to the Counter pin of DFB_HCnt.



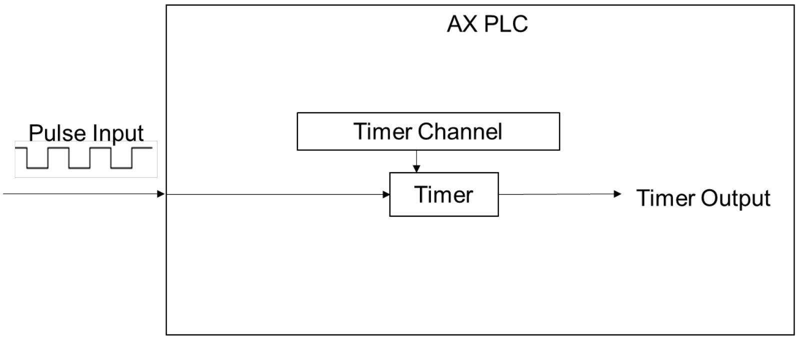
Variable	Type	Configuration Function
Counter_0	DFB_COUNTER_REF	Counter 0
Encoder_Axis	DMC_ENCODER_AXIS_REF	Counter 0/FreeEncoder_

- ■ For MC_GearIn function block in POU, the input variable corresponding to Master should be Encoder_Axis if the variable Counter_0 is used as the source of the master axis.

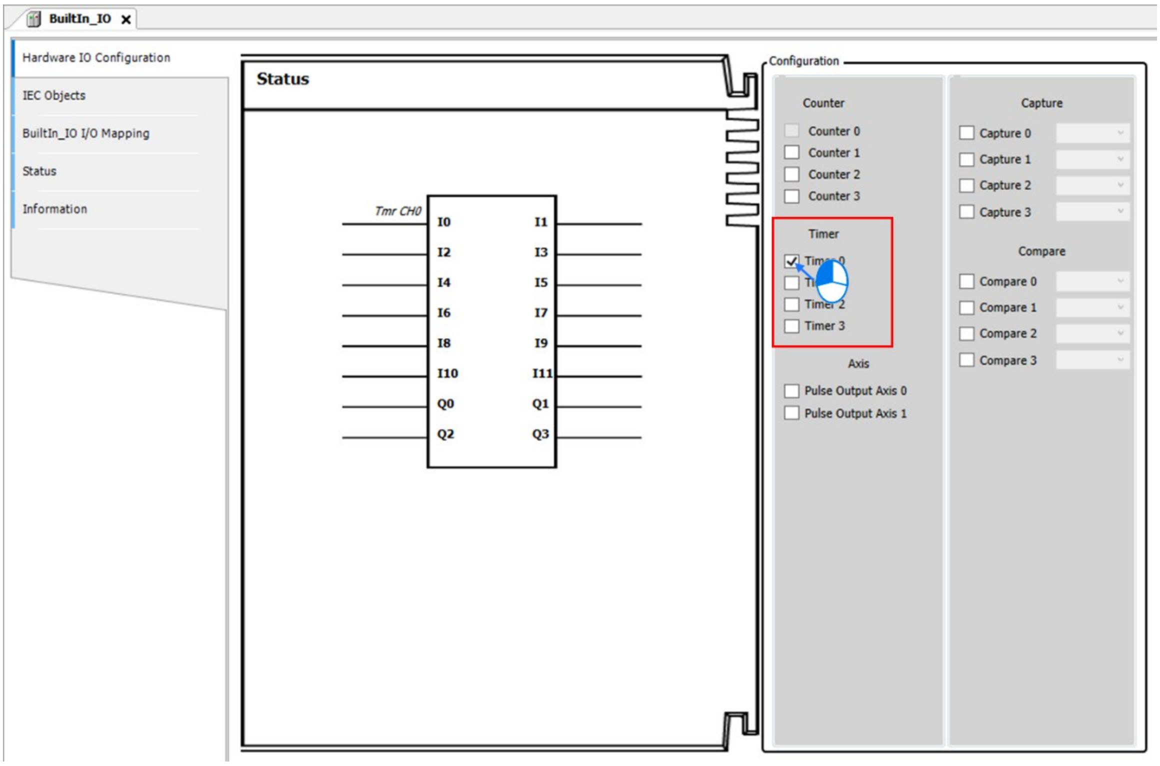


Variable	Type	Configuration Function
Counter_0	DFB_COUNTER_REF	Counter 0
Encoder_Axis	DMC_ENCODER_AXIS_REF	Counter 0/FreeEncoder_Axis

- **High-speed timer (Tmr)**



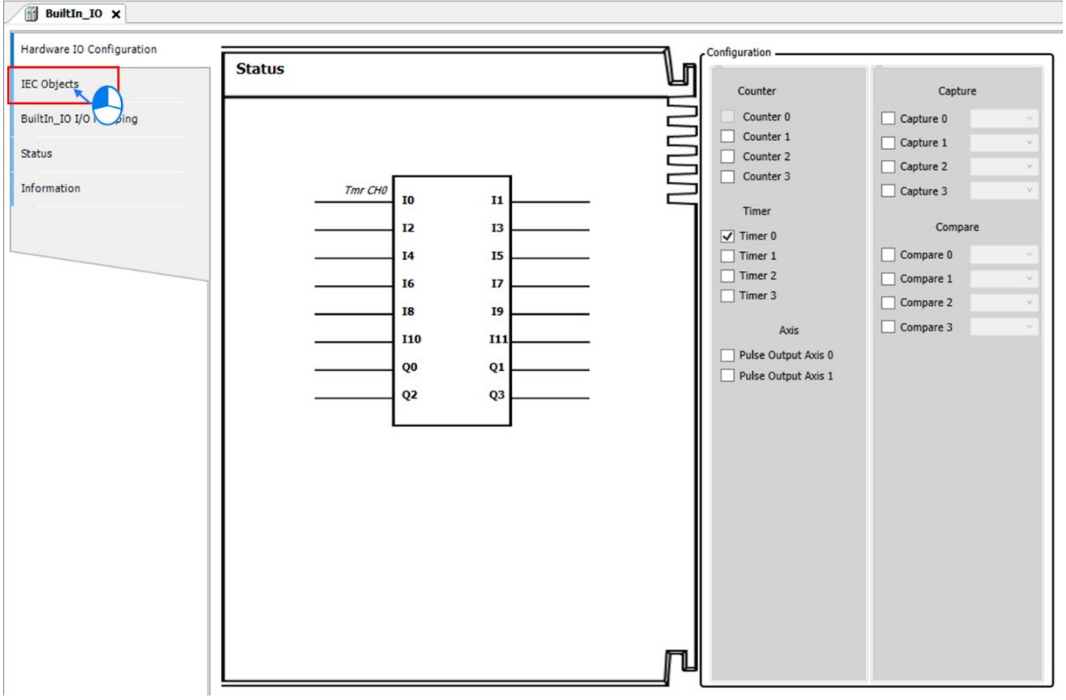
- **Enable high-speed timer function**
  - When selecting Timer function in Hardware IO Configuration, the high-speed timer in AX series is set as a 0.1μs timer. To enable timer function, select a timer from Timer 0 to 3 on BuiltIn_IO page to activate.



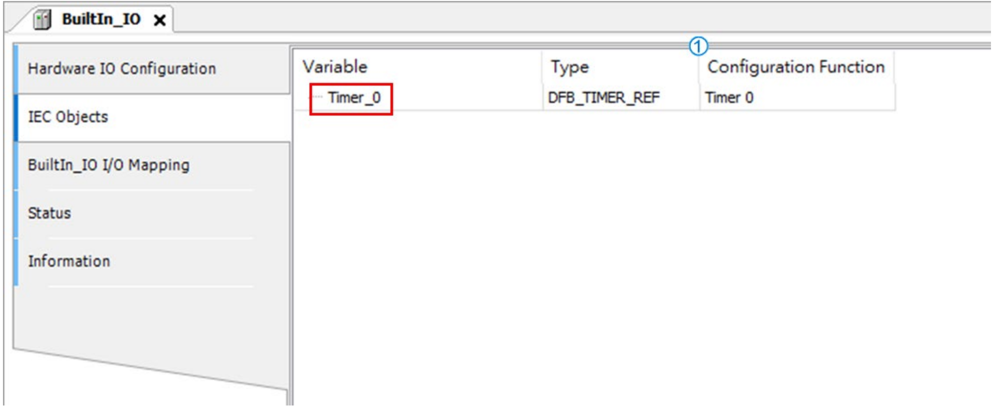
- **Use Timer in program**

The Timer variables can be used for MC function blocks in POU.

■ Click “IEC Objects” on BuiltIn_IO page.

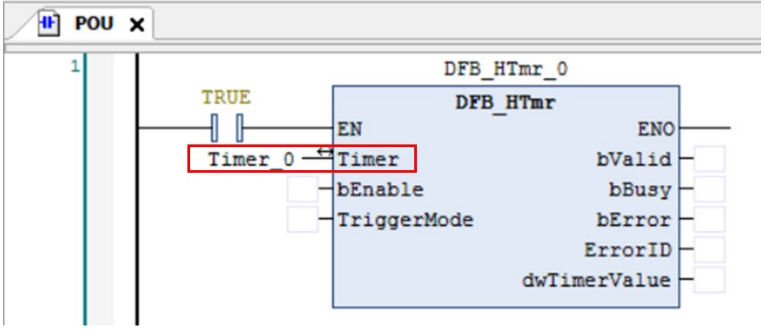


■ The column marked ① on the IEC Objects tab is the description of the configuration function of each variable. For the axis used in POU, the variable name should be set as Timer_0.



Variable	Type	Configuration Function
Timer_0	DFB_TIMER_REF	Timer 0

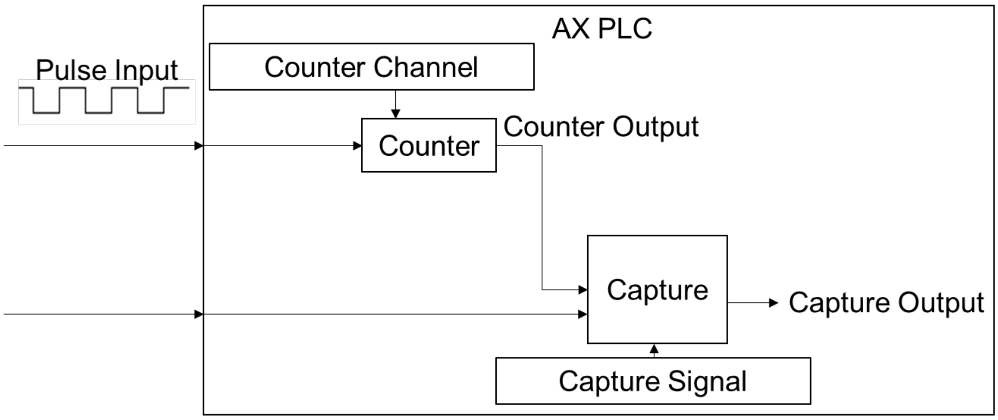
■ Enter Timer_0 as the variable name of high speed timer for DFB_HTmr_0 function block in POU.



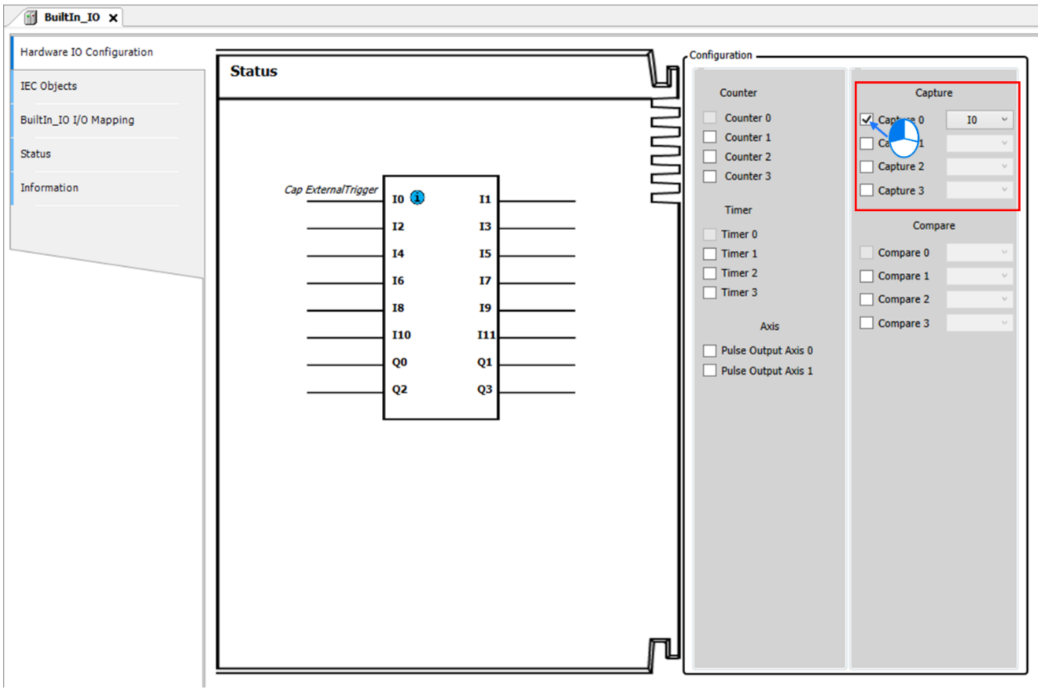
7.7.7.4 Capture/Compare Function Settings

This section introduces the Capture and Compare function blocks with built-in high-speed counters. A maximum of 8 groups of high-speed captures and compares can be supported by AX series motion controllers.

- **Capture**

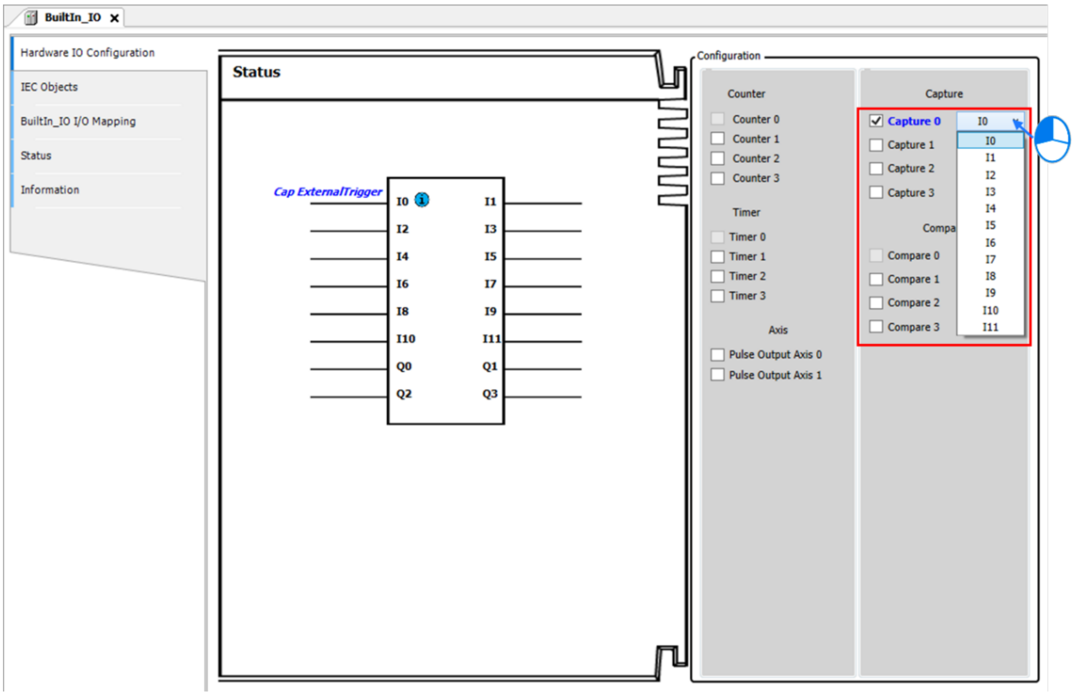


- **Enable Capture function**
  - Select one of the 4 Capture points to activate on the BuiltIn_IO page.



☑ Capture 0 I0

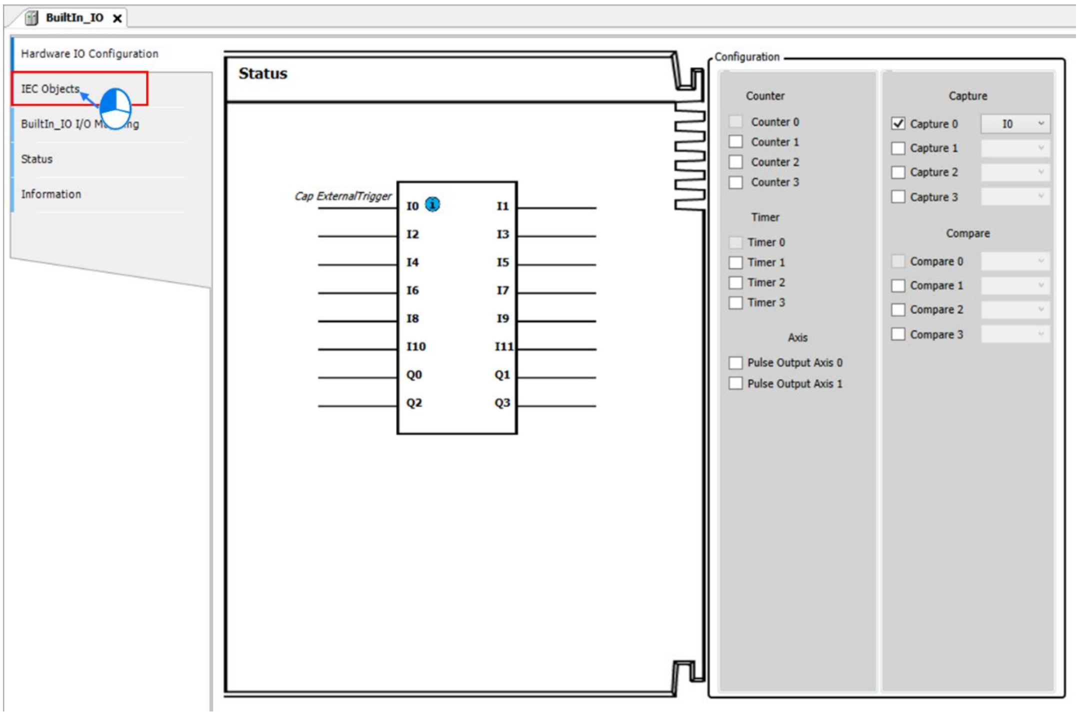
- Then choose an external trigger input from the drop-down list after selecting one Capture.



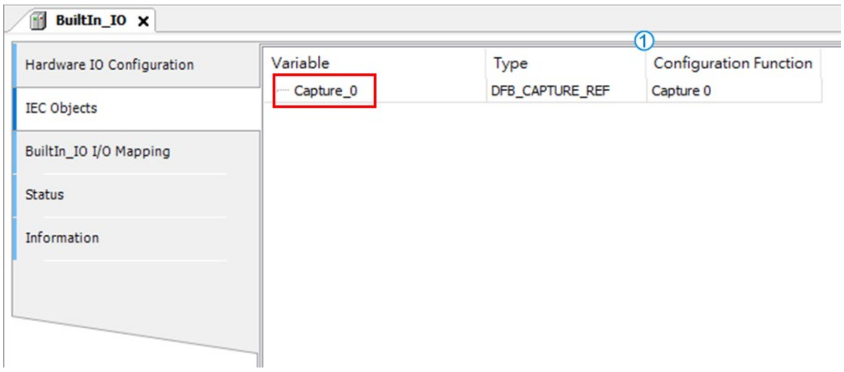
- **Use Capture in program**

The Capture variables can be used for MC function blocks in POU.

- Click “IEC Objects” on BuiltIn_IO page.

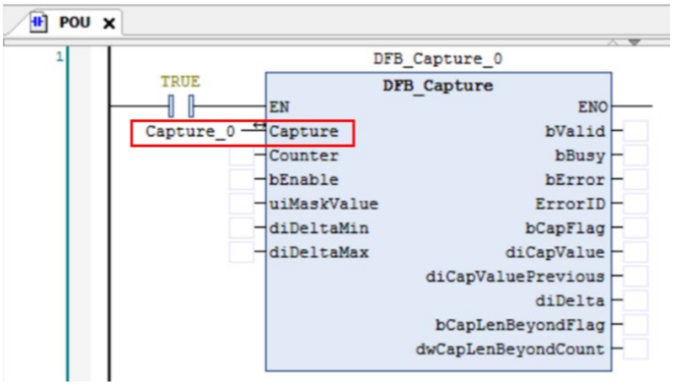


- The column marked ① on the IEC Objects tab is the configuration function of each variable. For the axis used in POU, the variable name should be set as Capture _0.

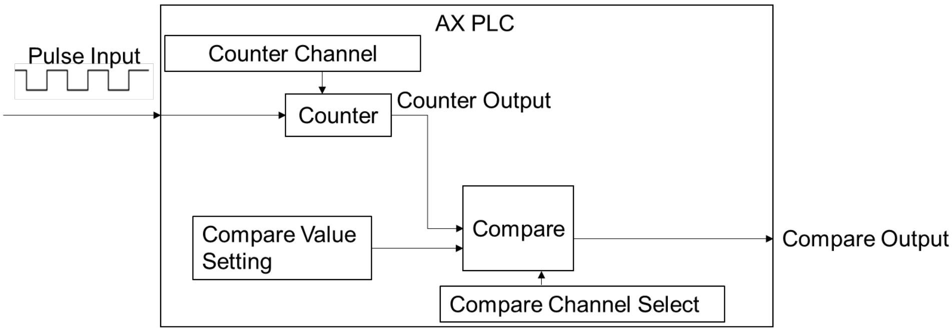


Variable	Type	Configuration Function
Capture_0	DFB_CAPTURE_REF	Capture 0

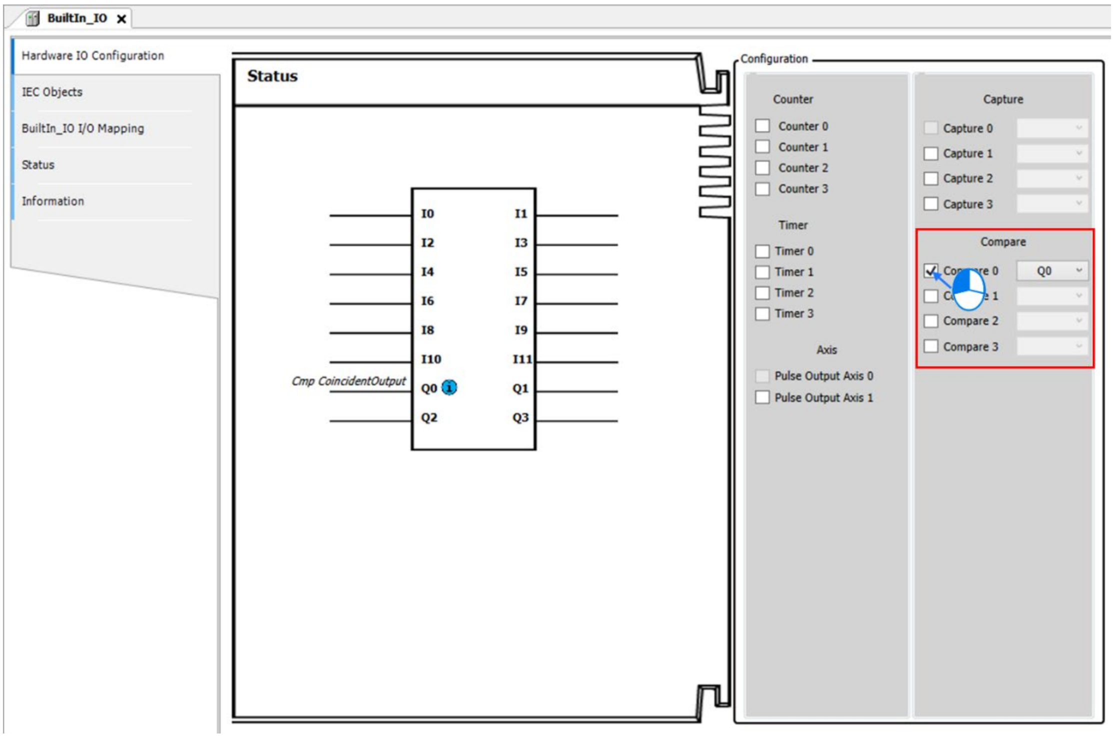
- For DFB_Capture function block in POU, enter Capture _0 as the variable name.



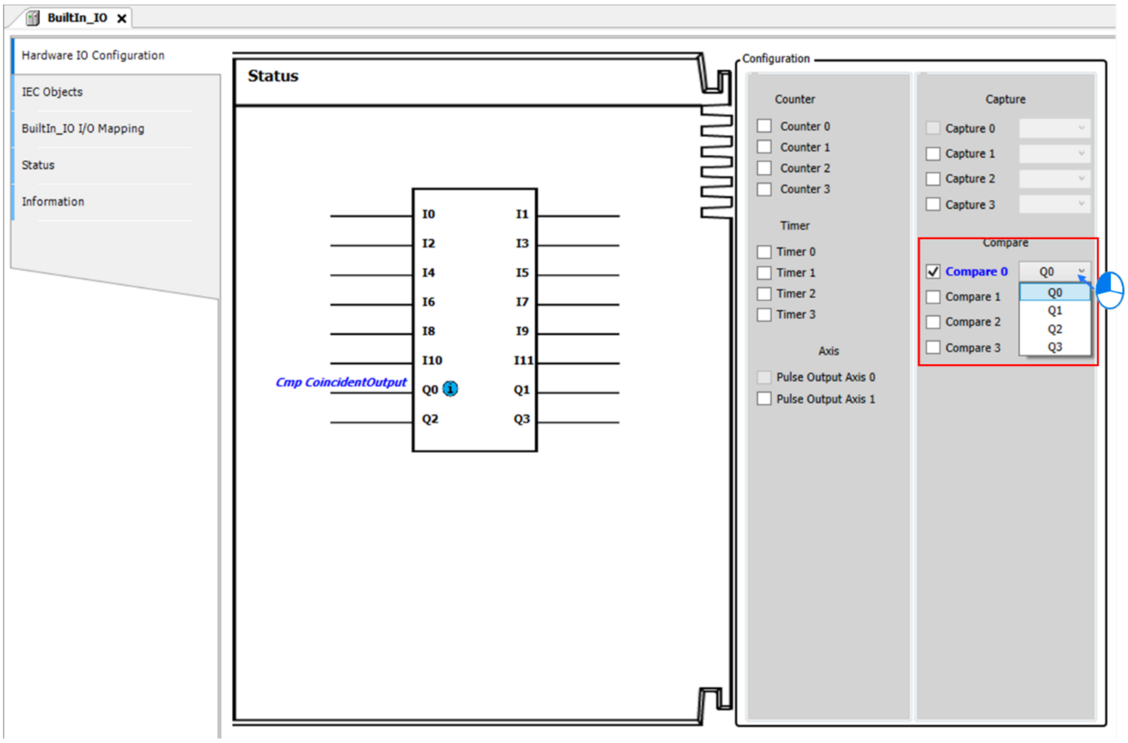
- **Compare**



- **Enable Compare function**
  - Select one of the 4 Compare points to activate on the BuiltIn_IO page.



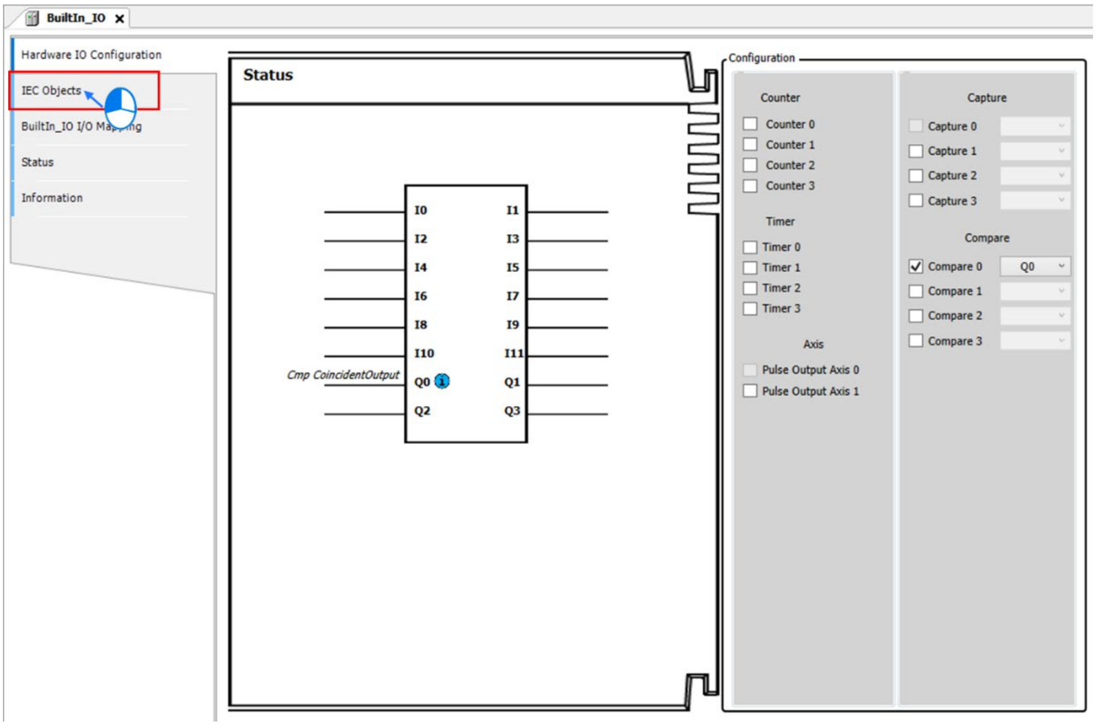
  - Then choose an external trigger output from the drop-down list after selecting one Compare.



- **Use Compare in program**

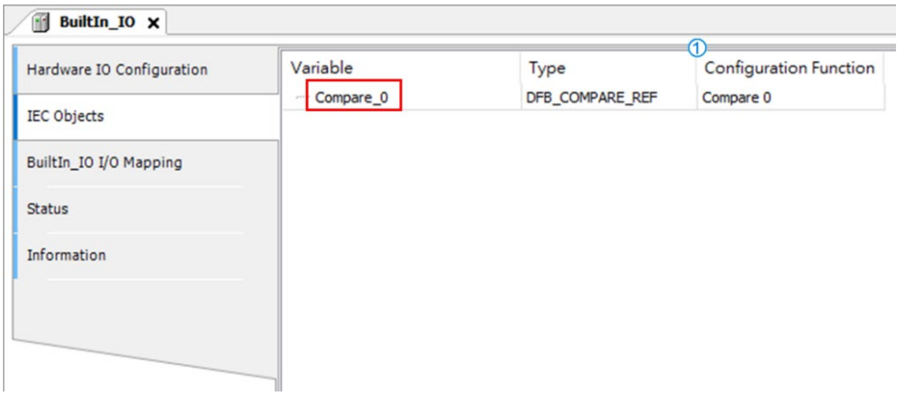
The Compare variables can be used for MC function blocks in POU.

- ■ Click “IEC Objects” on BuiltIn_IO page.



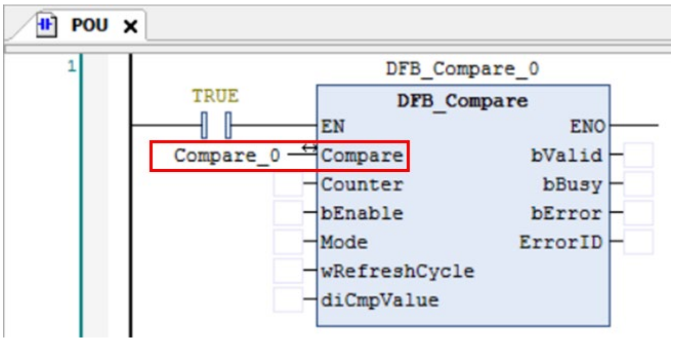
☑ Compare 0 — Q0

- ■ The column marked ① on the IEC Objects tab is the configuration function of each variable. For the name used in POU, the variable name should be set as Compare _0.



Variable	Type	Configuration Function
Compare_0	DFB_COMPARE_REF	Compare 0

- ■ For DFB_Compare function block in POU, enter Compare _0 as the variable name.

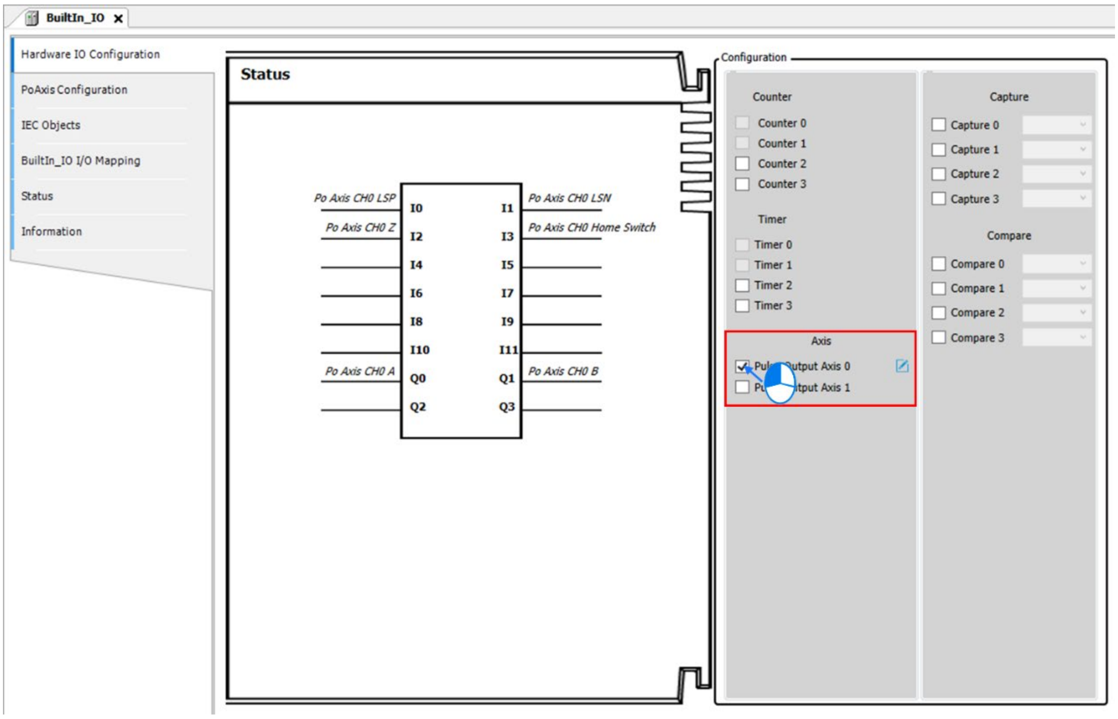


7.7.7.5 Pulse Output Function Settings

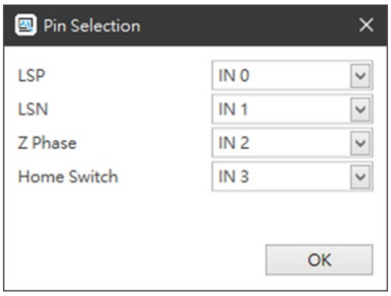
This section introduces pulse output function blocks with built-in IO shown as follows.

For AX-5 Series PLC CPU, a maximum of 2 groups of pulse-output units are for option.

- **Activate axis function**
  - Choose one of the two pulse output axes to activate on BuiltIn_IO page.

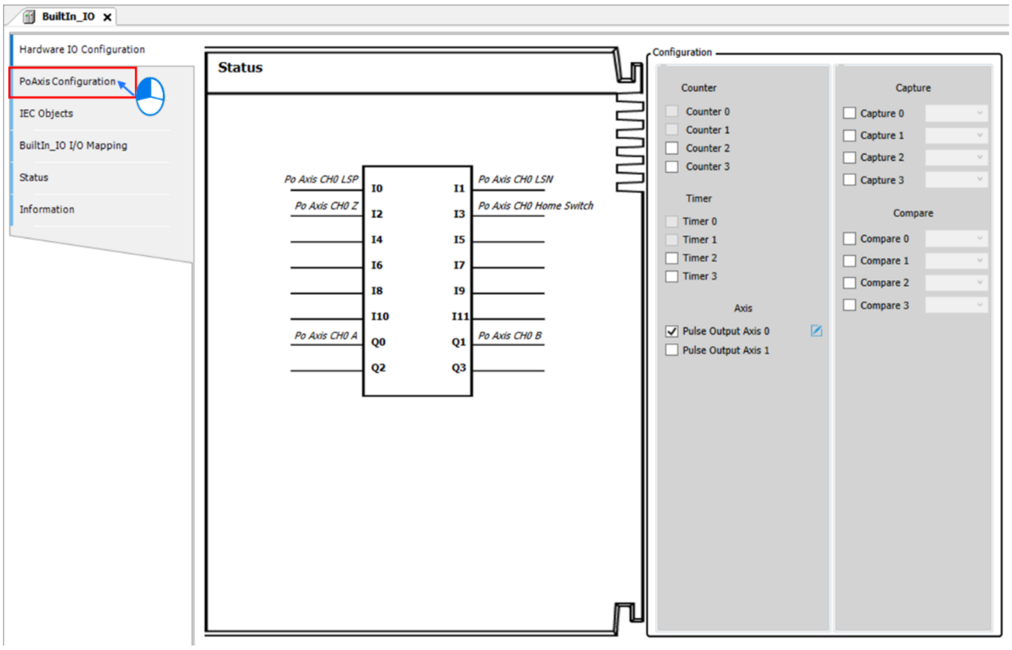


Clicking ✎, you can further set up the LSP, LSN, Z phase and home swich.



LSP: IN 0
LSN: IN 1
Z Phase: IN 2
Home Switch: IN 3

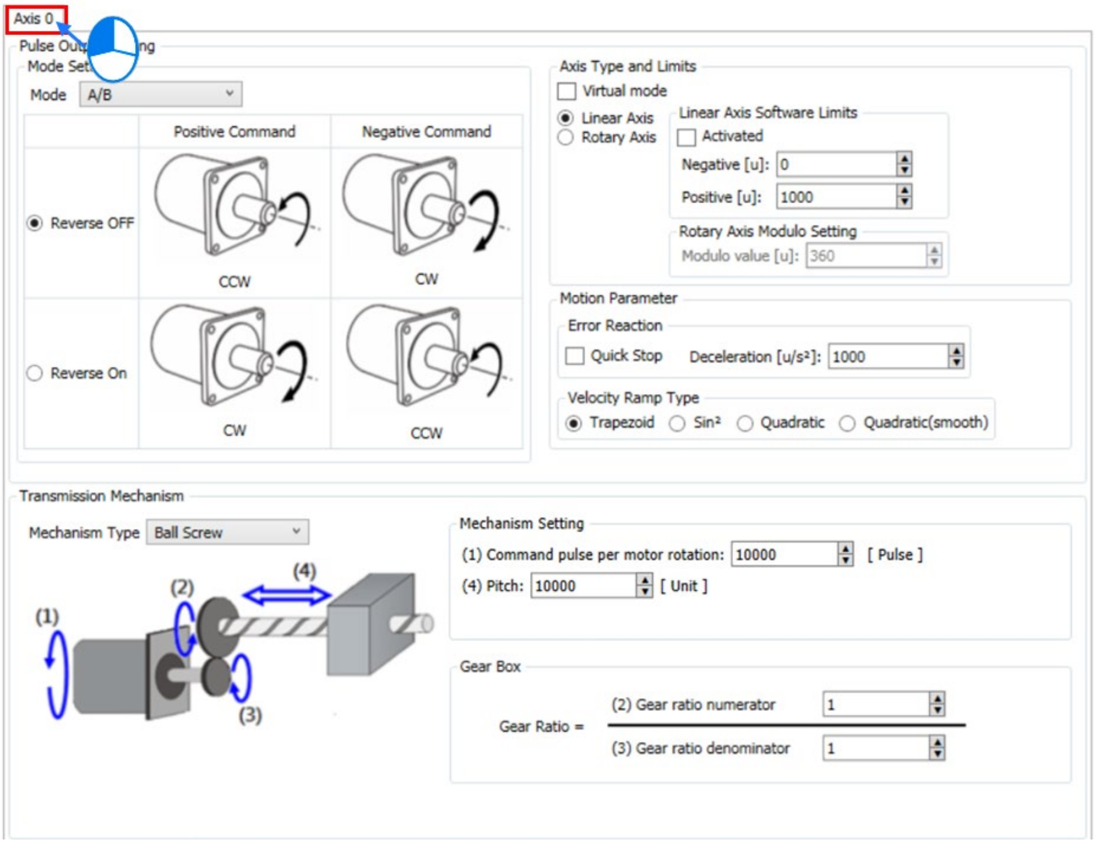
- Click “PoAxis Configuration” after activating Axis.



☑ Pulse Output Axis 0

☐ Pulse Output Axis 1

- Click to enter Axis 0 tab on PoAxis Configuration page.



Mode: A/B

● Reverse OFF

○ Reverse On

☐ Virtual mode

● Linear Axis

○ Rotary Axis

☐ Activated

Negative [u]: 0

Positive [u]: 1000

Modulo value [u]: 360

☐ Quick Stop

Deceleration [u/s²]: 1000

● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

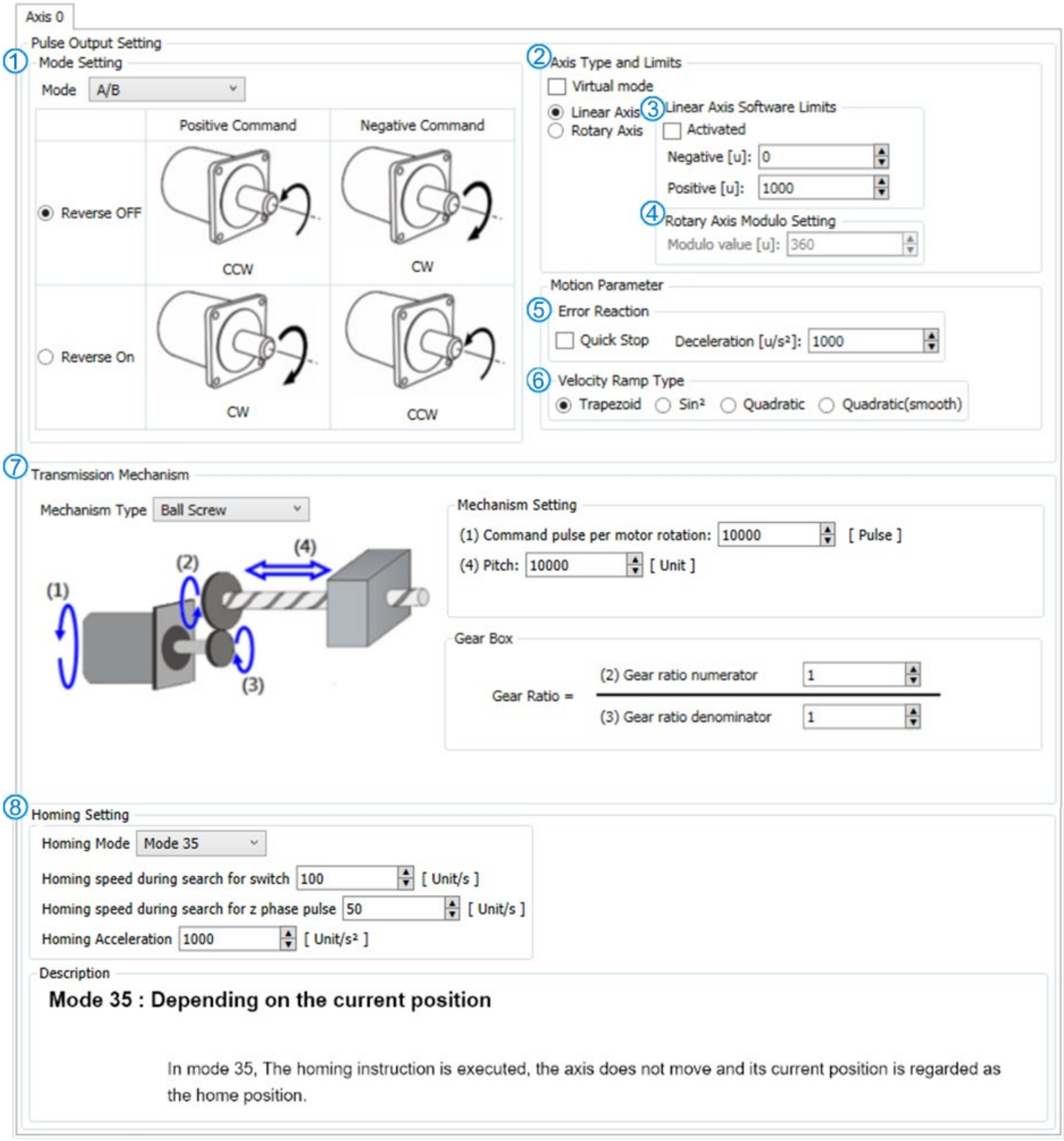
Mechanism Type: Ball Screw

(1) Command pulse per motor rotation: 10000 [Pulse]

(4) Pitch: 10000 [Unit]

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

- Axis-related settings can be configured on Pulse Output Setting page, which is described in the following information.



Pulse Output Setting

① Mode Setting

Mode: A/B

	Positive Command	Negative Command
● Reverse OFF	CCW	CW
○ Reverse On	CW	CCW

② Axis Type and Limits

☐ Virtual mode

● Linear Axis

○ Rotary Axis

③ Linear Axis Software Limits

☐ Activated

Negative [u]: 0

Positive [u]: 1000

④ Rotary Axis Modulo Setting

Modulo value [u]: 360

Motion Parameter

⑤ Error Reaction

☐ Quick Stop　Deceleration [u/s²]: 1000

⑥ Velocity Ramp Type

● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

⑦ Transmission Mechanism

Mechanism Type: Ball Screw

Mechanism Setting

(1) Command pulse per motor rotation: 10000 [Pulse]

(4) Pitch: 10000 [Unit]

Gear Box

Gear Ratio = (2) Gear ratio numerator 1 / (3) Gear ratio denominator 1

⑧ Homing Setting

Homing Mode: Mode 35

Homing speed during search for switch: 100 [Unit/s]

Homing speed during search for z phase pulse: 50 [Unit/s]

Homing Acceleration: 1000 [Unit/s²]

Description

Mode 35 : Depending on the current position

In mode 35, The homing instruction is executed, the axis does not move and its current position is regarded as the home position.

① Mode setting

Item	Function	Setting Value (Default)
Mode	Set the type of pulse output.	CW/CCW Pulse and Direction (A/B)
Reverse ON／Reverse OFF	Set the pulse axis to rotate in the positive or negative direction.	Reverse ON Reverse OFF (Reverse OFF)

② Axis Type and Limits

Item	Function	Setting Value (Default)
Virtual	Activate virtual axes.	TRUE FALSE (FALSE)
Linear Axis／Rotary Axis	Set the axis type to linear axis or rotary axis.	Linear Axis Rotary Axis (Linear Axis)

③ Linear Axis Software Limits

Item	Function	Setting Value (Default)
Activated	Activate software limit (only supports linear axes)	TRUE/FALSE (FALSE)
Negative[u]	Set the negative software limit.	(0)
Positive[u]	Set the positive software limit.	(10000)

④ Rotary Axis Modulo Setting

Item	Function	Setting Value (Default)
Modulo Value[u]	Set the range of rotation for a turn. (only supports rotary axes)	(360)

⑤ Error Reaction

Item	Function	Setting Value (Default)
Quick Stop	Stop the axis immediately.	(360)
Deceleration[u/s2]	The axis will perform a deceleration stop. (functional only when Quick Stop is not activated)	(10000)

⑥ Velocity Ramp Type

Item	Function	Setting Value (Default)
Trapezoid/Sin^2/Quadratic/ Quadratic (Smooth)	Set the ramp type for axis motion.	(Trapezoid)

⑦ Software Configuration Page: Please refer to **7.7.7.3 Pulse Encoder Settings.**

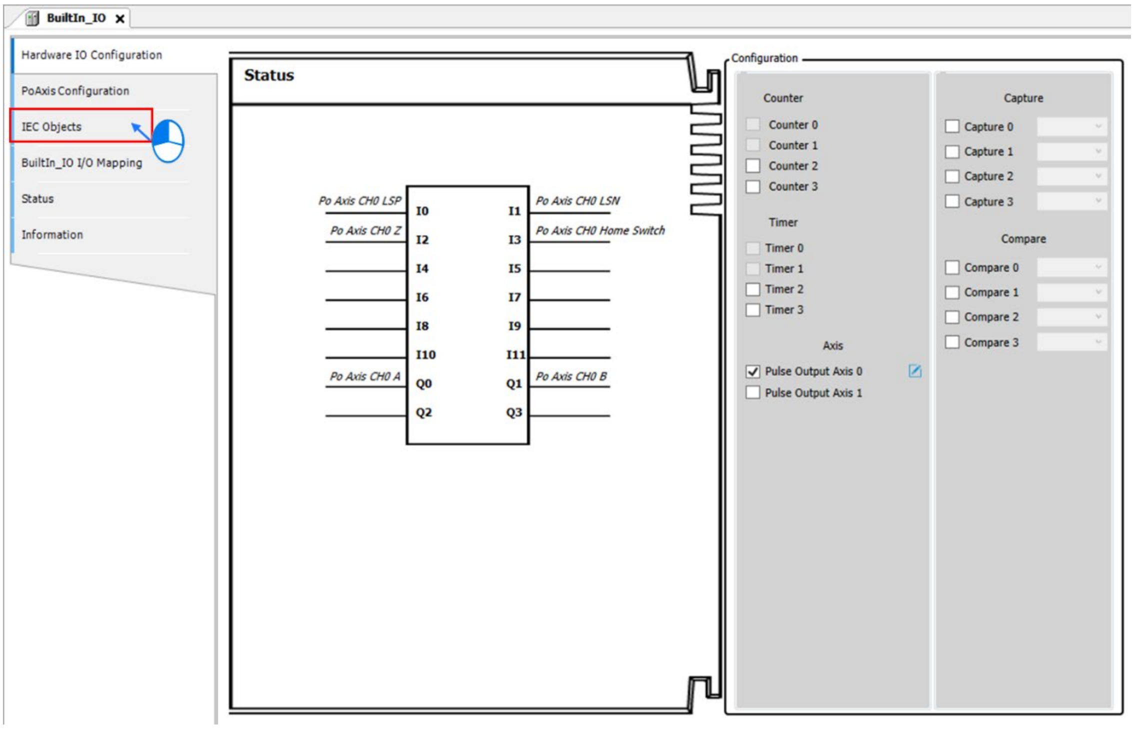
⑧ Homing Setting

Item	Function	Setting Value (Default)
Homing Mode	Set the homing mode.	(Mode 35)
Homing speed during search for switch	Set the homing speed during search for switch.	(100)
Homing speed during search for z phase pulse	Set the homing speed during search for z phase pulse.	(50)
Homing Acceleration	Set the acceleration of homing.	(1000)

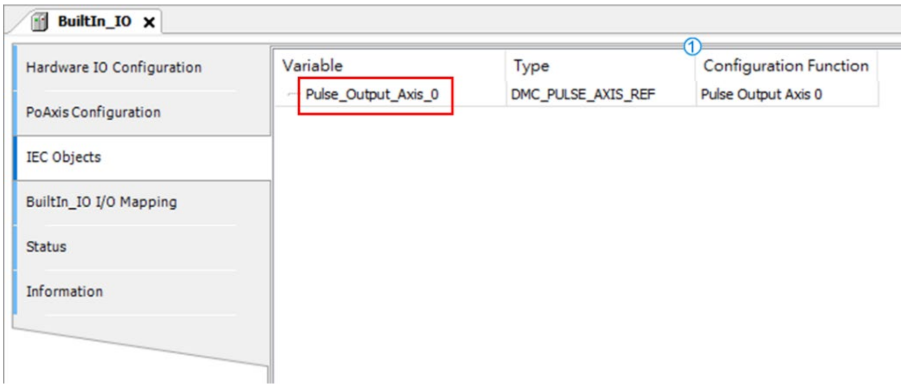
- **Use Pulse Axis in program**

To use Pulse Axis in POU, Pulse Output Axis variables are required for MC function blocks in POU.

- ■ Click “IEC Objects” on BuiltIn_IO page.

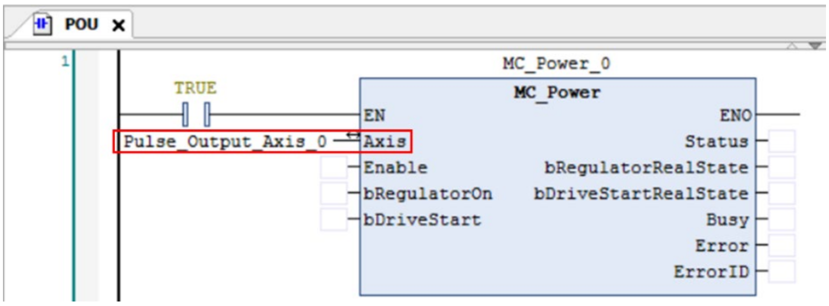


- ■ The column marked ① on the IEC Objects tab is the configuration function of each variable. For the axis used in POU, the axis name should be set as Pulse_Output_Axis_0.



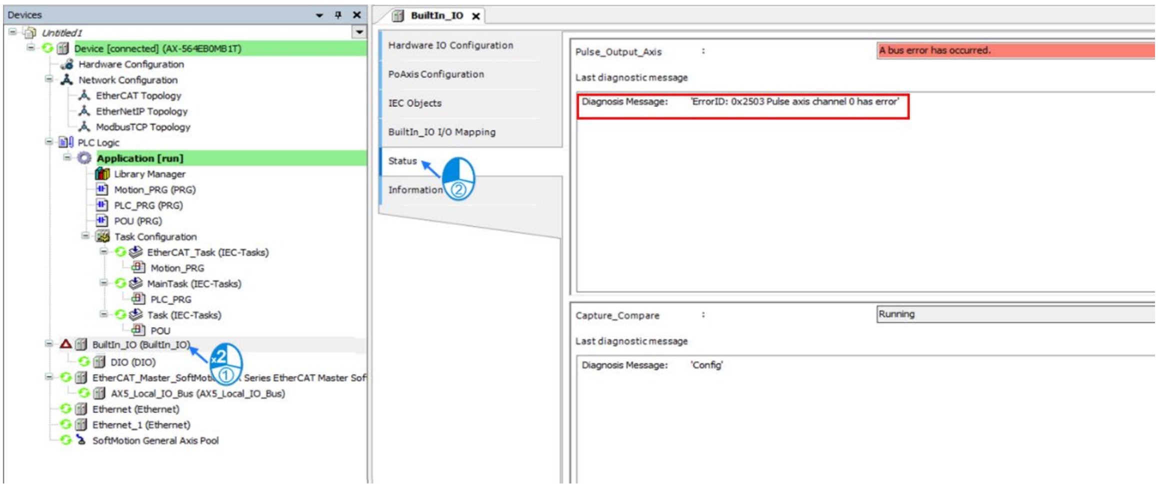
Variable	Type	Configuration Function
Pulse_Output_Axis_0	DMC_PULSE_AXIS_REF	Pulse Output Axis 0

- For MC_Power function block in POU, enter Pulse_Output_Axis_0 as the axis name.



7.7.7.6 Confirm High-Speed IO Errors

Errors in Pulse Output Axis are displayed on Status tab under BuiltIn_IO page with messages notifying you of which pulse axis has an error.



Pulse_Output_Axis : A bus error has occurred.

Last diagnostic message

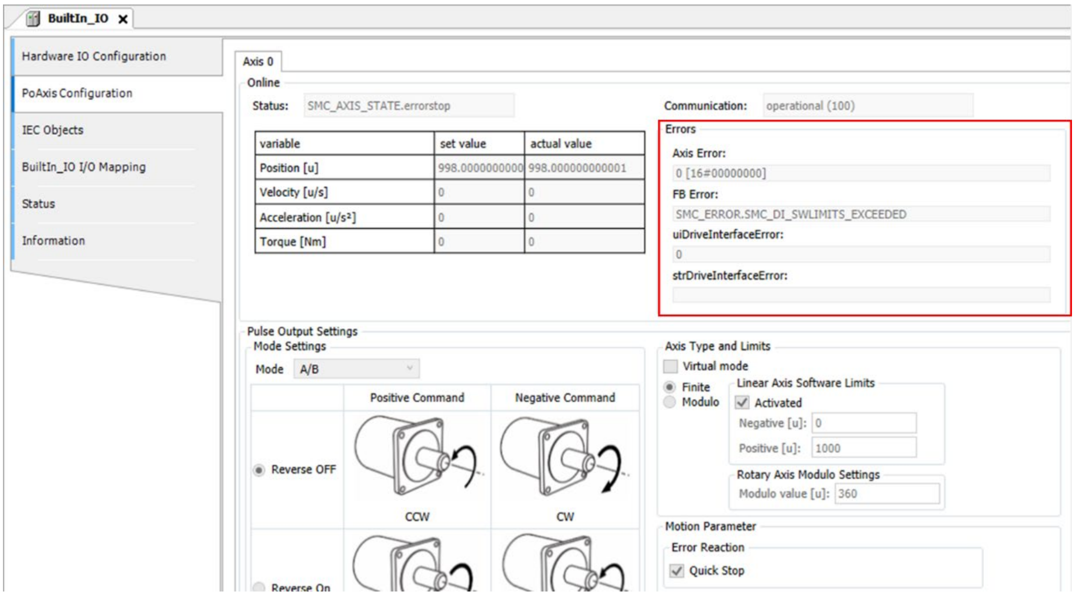
Diagnosis Message: 'ErrorID: 0x2503 Pulse axis channel 0 has error'

Capture_Compare : Running

Last diagnostic message

Diagnosis Message: 'Config'

You can continue to check and monitor the error information on PoAxis Configuration tab page.



Status: SMC_AXIS_STATE.errorstop

Communication: operational (100)

variable	set value	actual value
Position [u]	998.0000000000	998.000000000001
Velocity [u/s]	0	0
Acceleration [u/s²]	0	0
Torque [Nm]	0	0

Axis Error: 0 [16#00000000]

FB Error: SMC_ERROR.SMC_DI_SWLIMITS_EXCEEDED

uiDriveInterfaceError: 0

strDriveInterfaceError:

Mode A/B

● Reverse OFF

○ Reverse On

☐ Virtual mode

● Finite

○ Modulo

☑ Activated

Negative [u]: 0

Positive [u]: 1000

Modulo value [u]: 360

Error Reaction

☑ Quick Stop

7.7.8 Other Features

7.7.8.1 Change Current Position

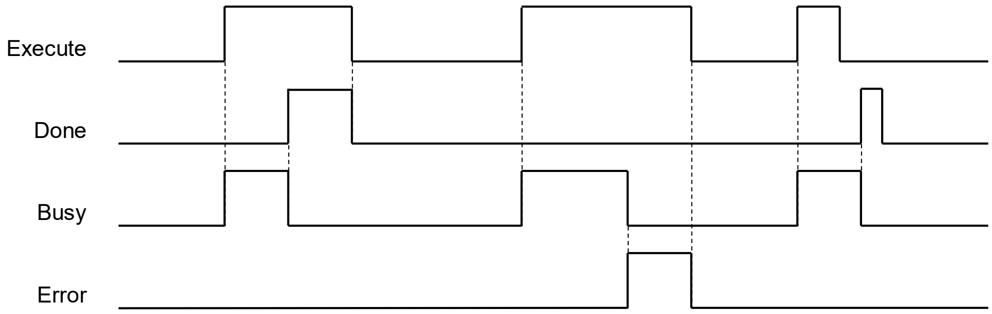
- **MC_SetPosition**

This function block is to change the current command position of the servo axis to the specified target position. If MC_SetPosition is executed on the external encoder axis, the current feedback position will change to the specified target position after instruction execution is finished.

While the current position of the instruction (command position) is changed, the actual position from the feedback signals changes accordingly. The following error between command position and actual position remains the same value.

The function block is used to change the coordinate system and does not lead to servo drive and motor movement. And the current position of the encoder axis can be edited by this function block.

- **Timing diagram**



7.7.8.2 Software Limit

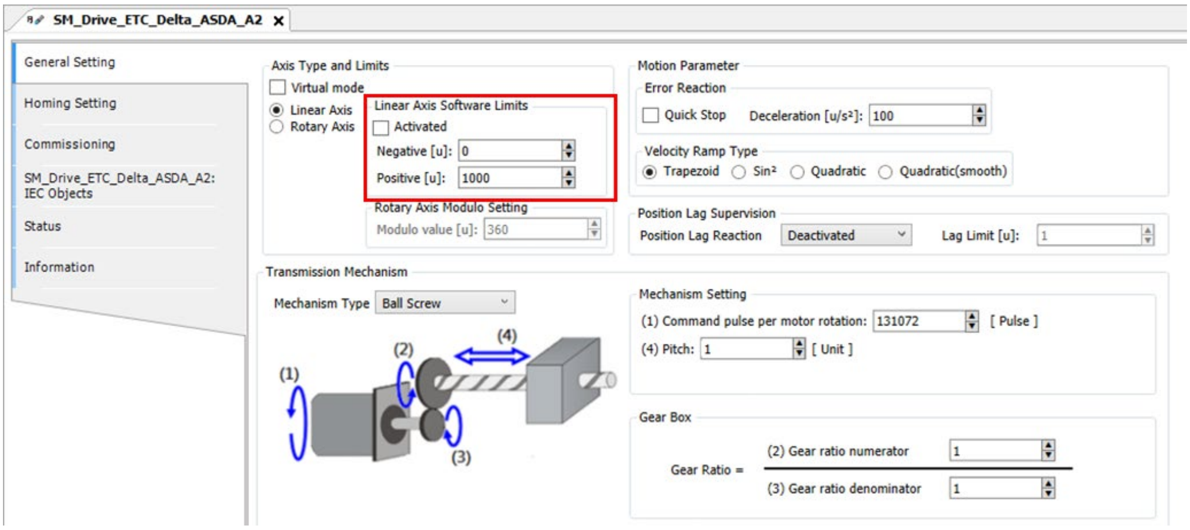
In addition to hardware limits, the range of axis motion can also be limited by software limits.

Values for forward and reverse limit ranges need to be set before activating software limits. Software limits are set to be not activated as default so as to prevent any damage to the device when an operator error occurs.

Note: Refer to section 7.7.2.4 for example on Stop Method.

- **Software display**

Can be configured via DIADesigner-AX software.



Axis Type and Limits
☐ Virtual mode
● Linear Axis
○ Rotary Axis
Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 1000
Rotary Axis Modulo Setting
Modulo value [u]: 360

Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 100
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

Position Lag Supervision
Position Lag Reaction: Deactivated Lag Limit [u]: 1

Transmission Mechanism
Mechanism Type: Ball Screw

Mechanism Setting
(1) Command pulse per motor rotation: 131072 [Pulse]
(4) Pitch: 1 [Unit]

Gear Box
Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

The positive and negative position are able to be resized on the configuration page:

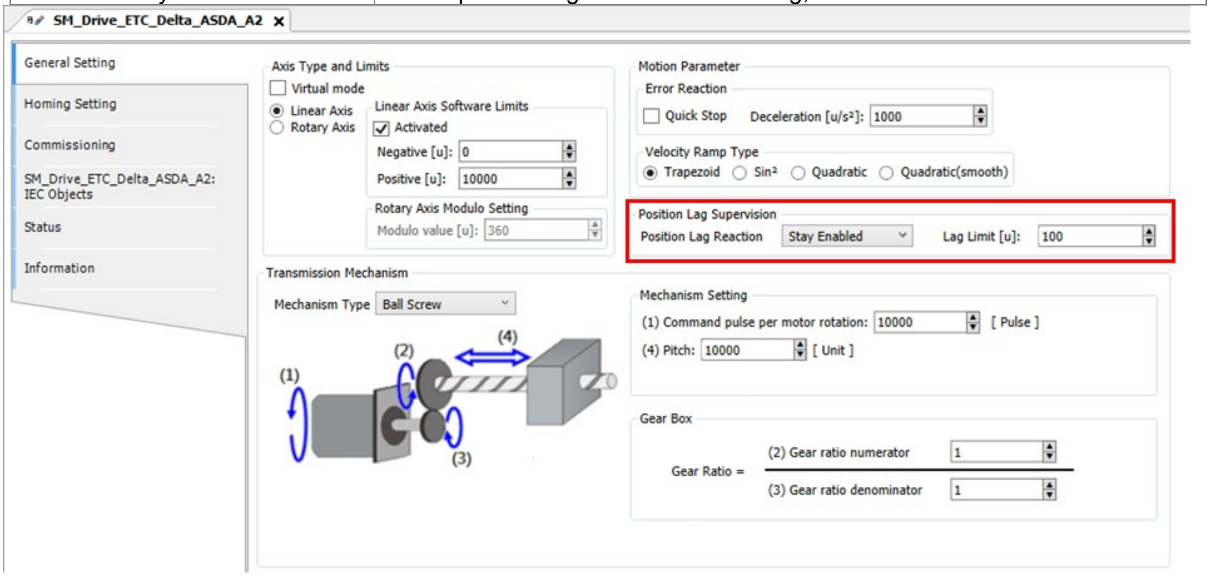
Item	Data Type	Default Setting
Negative	LREAL	0.0
Position	LREAL	10000.0

7.7.8.3 Position Lag Setting

The command position as well as feedback position are located at zero while the axis is in motion. If there's a great difference between command position and feedback position, an error will be reported.

The position lag reaction is set to "Deactivated" as default.

Setting mode	Function
Deactivated	Not activated.
Disable drive	When position lag exceeds limit setting, axis will shift to servo off.
Do quickstop	When position lag exceeds limit setting, axis will shift to quick stop.
Stay enabled	When position lag exceeds limit setting, axis will remain as servo on.



Axis Type and Limits
☐ Virtual mode
● Linear Axis
○ Rotary Axis
Linear Axis Software Limits
☑ Activated
Negative [u]: 0
Positive [u]: 10000
Rotary Axis Modulo Setting
Modulo value [u]: 360

Motion Parameter
Error Reaction
☐ Quick Stop Deceleration [u/s²]: 1000
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)

Position Lag Supervision
Position Lag Reaction: Stay Enabled Lag Limit [u]: 100

Transmission Mechanism
Mechanism Type: Ball Screw

Mechanism Setting
(1) Command pulse per motor rotation: 10000 [Pulse]
(4) Pitch: 10000 [Unit]

Gear Box
Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

7.7.8.4 Cam Switch Function

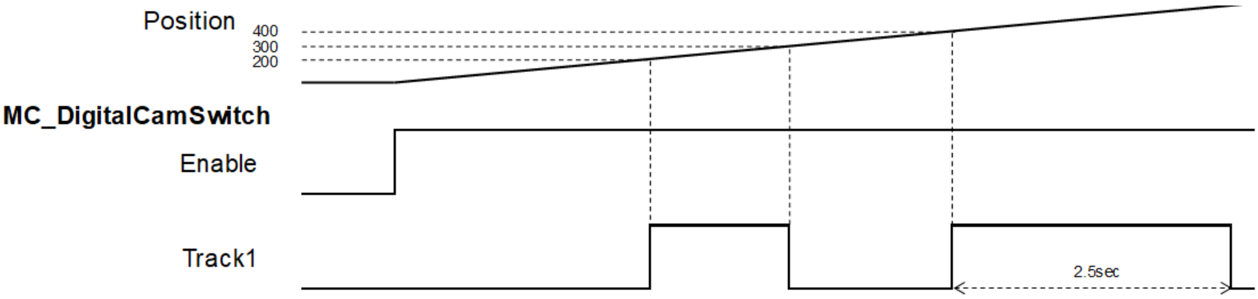
MC_DigitalCamSwitch

Specify the tappet position. True when the moving axis reaches the specified position, and then False when the axis leaves it. The following example is about configuration settings.

- Example: Use two switches in the same track with MC_DigitalCamSwitch instruction.
  - Parameter setting

Parameter	Type	Switch1	Switch2
TrackNumber	INT	1	1
FirstOnPosition [u]	REAL	200	400
LastOnPosition [u]	REAL	300	-
AxisDirection	INT	0=Both	0=Both
CamSwitchMode	INT	0=Position	1=TIME
Duration	TIME	-	2500ms

  - Trigger and timing



- Switch 1 on Track 1 is ON when the position reaches 200 and turns to OFF once the axis position reaches 300.
- When the position reaches 400, Switch 1 turns to ON again for 2500ms, and then shifts to OFF.

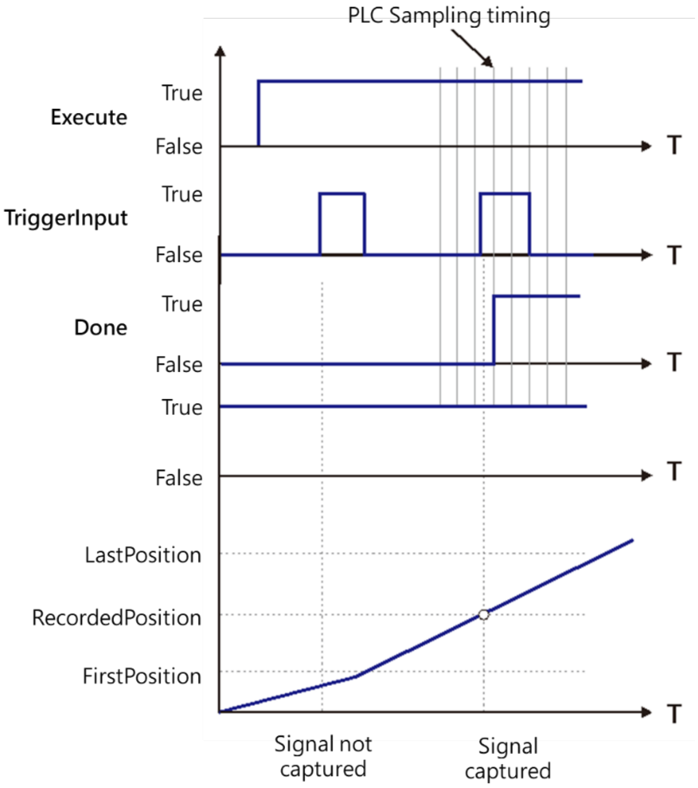
7.7.8.5 Position Capture

Using sensor signals as trigger signals, MC_TouchProbe captures and records the current position of the servo axis once a signal trigger occurs after signals are input to the servo drive.

A total of two trigger signals can be configured for each axis. MC_AbortTrigger is used to abort capture function.

Function description:

- The touch probe operation activates for only one time for recording the very first trigger signal after Execute is set as True. When a valid position is captured and recorded, the following trigger signals will be ignored.
- One function block instance corresponds to only one MC_TouchProbe instruction.
- If there were multiple function block instances on the same capture and axis, MC_TRIGGER_REF should use different TouchProbeID, which identifies different TouchProbe actions. The TouchProbeID corresponds to MC_AbortTrigger as well.
- The operation of MC_TouhcProbe with window mask function is demonstrated as below:



- When the first trigger signal is input, the signal is not accepted because the axis position hasn't reached the specified window mask section.
- When the axis position enters the window mask section, the second trigger input signal is accepted, and Done changes to True in the next period after the signal capture is completed.

7.8 Programming Example

The following section explains on the basis of the programming example.

7.8.1 Device Framework

The following devices are used in the example.

Device	Model Name
CPU	AX-5
Power	DVP-PS02
Servo drive	Delta ASDA-A2-E
Servo motor	Delta ECMA-C

7.8.1.1 Utilization

Please refer to the following manuals for information regarding device configuration and wiring.

Device	Reference
CPU and Power	Chapter 2 in this manual
Servo drive	Related configuration description in Delta servo drive user manuals
Wiring for EtherCAT slave device	Delta ASDA A2-E EtherCAT Interface Servo Drive User Manual

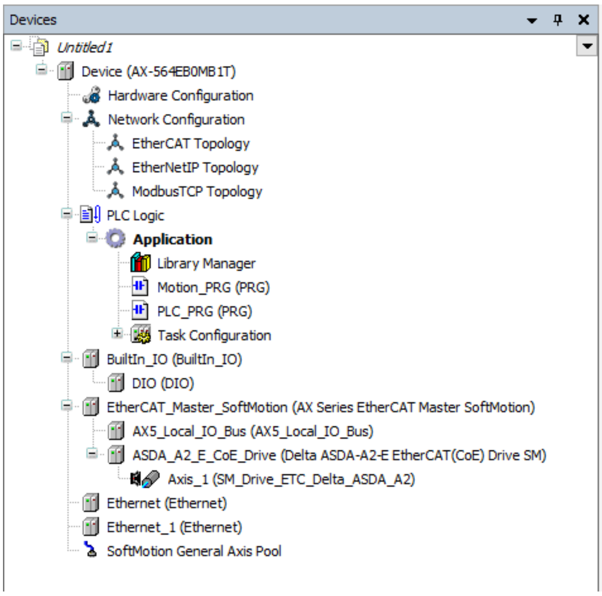
7.8.1.2 Configurations

The following configuration is applied in the example in the next section.

Device	Configuration setting
Controller	Chapter 2 in this manual
Motion control settings	Chapter 7 in this manual
Servo parameters	Use the default settings of ASDA-A2-E slave, gear ratio=10000：10000

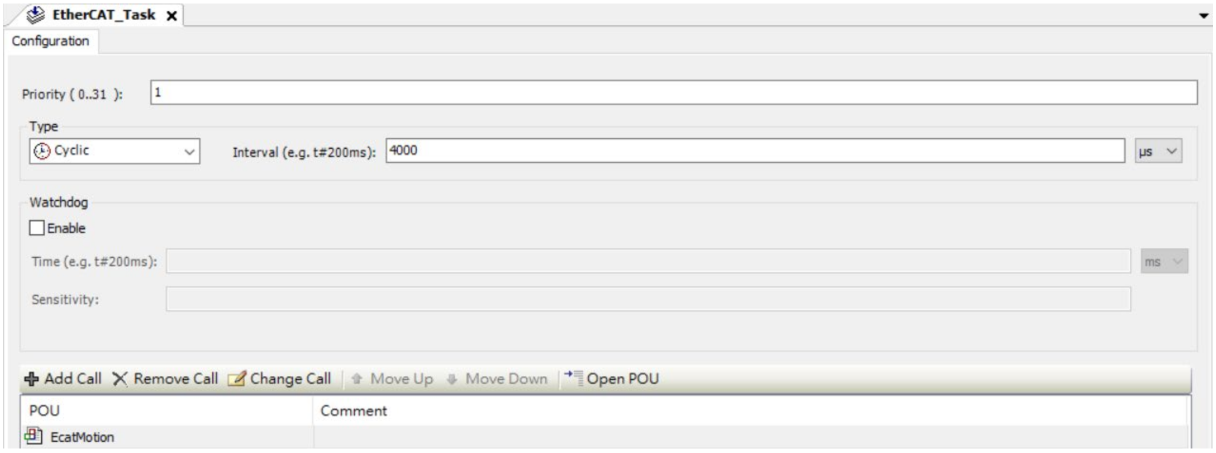
7.8.2 Examples

The following example uses the same POU in EtherCAT task to explain. Also, the required variables will be declared and used in this POU Task. (The POU naming in LD and ST languages will be different for illustration purpose.)



- *Untitled1*
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
      - EtherCAT Topology
      - EtherNetIP Topology
      - ModbusTCP Topology
    - PLC Logic
      - **Application**
        - Library Manager
        - Motion_PRG (PRG)
        - PLC_PRG (PRG)
        - Task Configuration
    - BuiltIn_IO (BuiltIn_IO)
      - DIO (DIO)
    - EtherCAT_Master_SoftMotion (AX Series EtherCAT Master SoftMotion)
      - AX5_Local_IO_Bus (AX5_Local_IO_Bus)
      - ASDA_A2_E_CoE_Drive (Delta ASDA-A2-E EtherCAT(CoE) Drive SM)
        - Axis_1 (SM_Drive_ETC_Delta_ASDA_A2)
    - Ethernet (Ethernet)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

The Interval time for ECAT synchronization is set to 4 ms.



Priority (0..31): 1

Type: Cyclic

Interval (e.g. t#200ms): 4000 µs

Watchdog

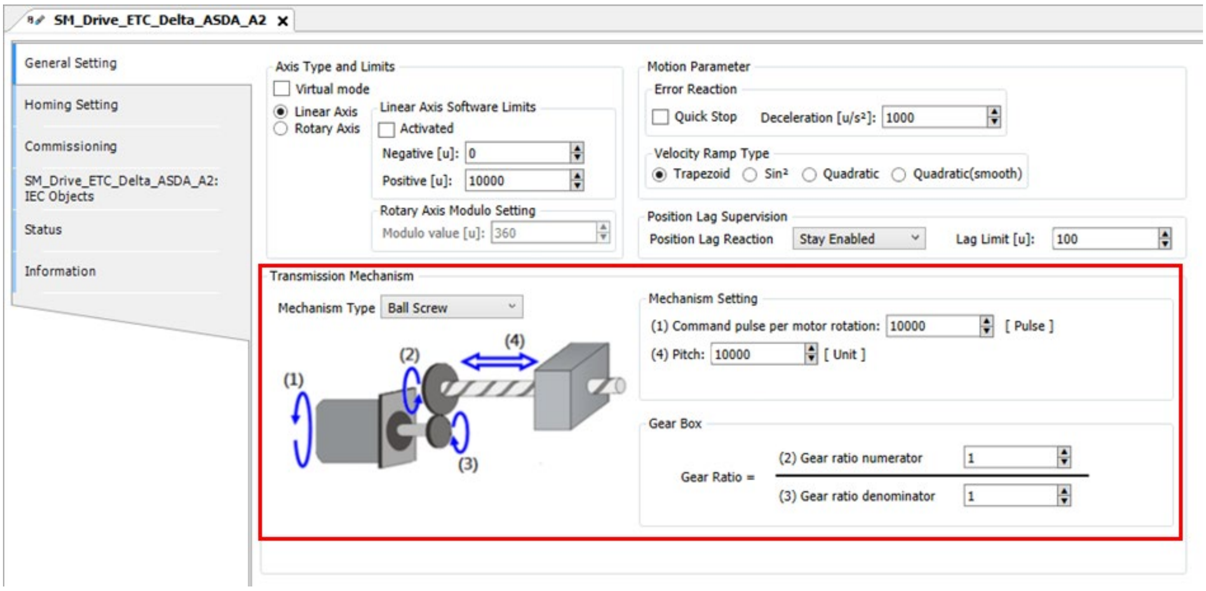
☐ Enable

Time (e.g. t#200ms): ms

Sensitivity:

POU	Comment
EcatMotion	

Set the gear ratio as 10000:10000 for mechanism setting.



SM_Drive_ETC_Delta_ASDA_A2

Axis Type and Limits
☐ Virtual mode
● Linear Axis
○ Rotary Axis
Linear Axis Software Limits
☐ Activated
Negative [u]: 0
Positive [u]: 10000
Rotary Axis Modulo Setting
Modulo value [u]: 360

Motion Parameter
Error Reaction
☐ Quick Stop
Deceleration [u/s²]: 1000
Velocity Ramp Type
● Trapezoid ○ Sin² ○ Quadratic ○ Quadratic(smooth)
Position Lag Supervision
Position Lag Reaction: Stay Enabled
Lag Limit [u]: 100

Transmission Mechanism
Mechanism Type: Ball Screw
Mechanism Setting
(1) Command pulse per motor rotation: 10000 [Pulse]
(4) Pitch: 10000 [Unit]
Gear Box
Gear Ratio = (2) Gear ratio numerator: 1 / (3) Gear ratio denominator: 1

7.8.2.1 Servo On

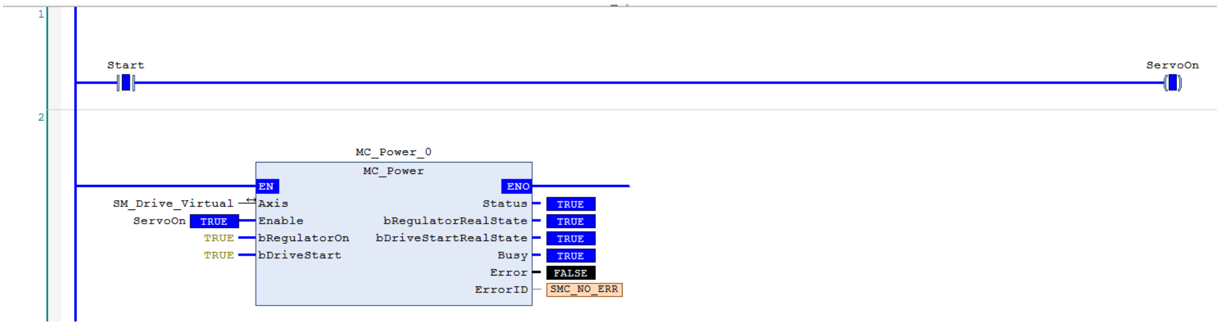
Execute MC_Power (Servo on) instruction to activate the servo drive after the EtherCAT communication is built in the following example with LD and ST programming languages supported.

- **Main variables used in programming**

Variable	Data Type	Default	Note
SM_Drive_Virtual	AXIS_REF_SM3	-	Virtual axis variable
Start	BOOL	FALSE	Shift to True to enable ServoOn when starting the servo.

- **LD language**

Check for the successful EtherCAT communication when Start is True so as to enable MC_Power instruction via ServoOn output. When Status changes to True, the servo is ON.



- **ST language**

Check for the successful EtherCAT communication when Start is True so as to enable MC_Power via ServoOn output. When Status changes to True, the servo is ON.

Monitoring window can also be used to observe the variable output status with no need for naming the output variables.

```
IF Start THEN
        ServoOn :=TRUE;
        ELSE
        ServoOn :=FALSE;
END_IF

//MC_Power
MC_Power_0(
        Axis:= SM_Drive_Virtual,
        Enable:= ServoOn,
        bRegulatorOn:= TRUE,
        bDriveStart:= TRUE,
        Status=> ,
        bRegulatorRealState=> ,
        bDriveStartRealState=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );
```

7.8.2.2 Reset and Control Single-axis Error

You can view the error information of variable status through Watch table. Take MC_MoveVelocity input as an example, when the acceleration value is set as 0 and Execute is True, an error will occur in the function block and the ErrorID displays Row Data 301. You can find the complete error message in the Watch table, which is SMC_MV_INVALID_ACCDEC_VALUES. After you make the troubleshooting with the manual’s help, MC_MoveVelocity can function normally by shifting the Execute status from False to True. As for MC_Reset, it is used for clearing servo errors.

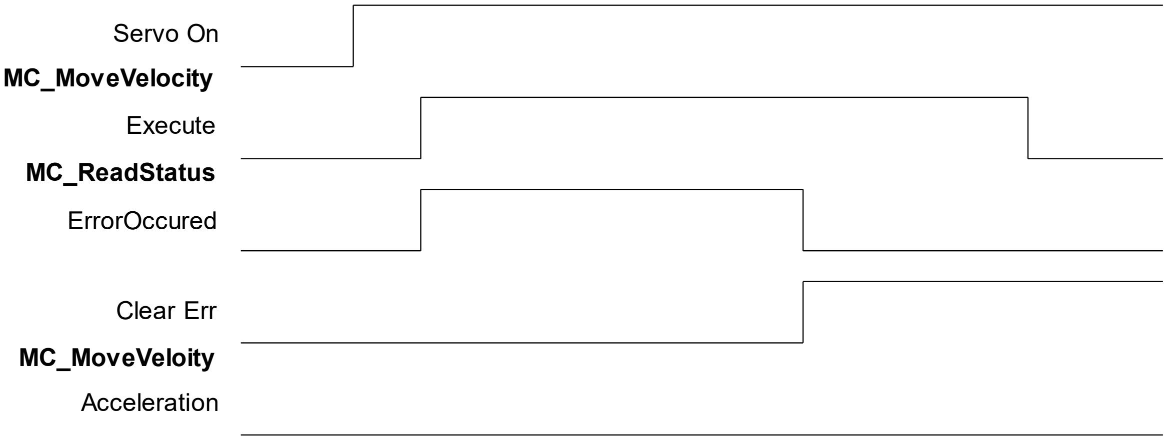
The following examples illustrate the program in LD and ST programming languages respectively.

- **Main variables used in programming**

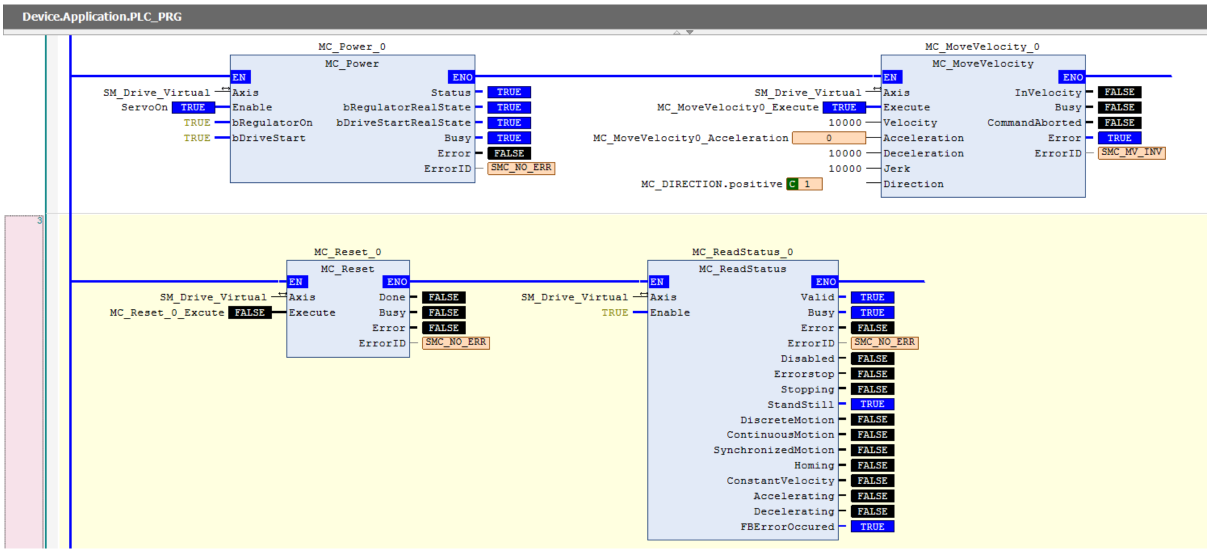
Variable	Data Type	Default	Description
SM_Drive_Virtual	AXIS_REF_SM3	-	Virtual axis variables
ServoOn	BOOL	FALSE	To enable MC_Power
MC_MoveVelocity0_Execute	BOOL	FALSE	Execute input of the velocity instruction
MC_MoveVelocity0_Acceleration	LREAL	0	Acceleration input of the velocity instruction, for setting acceleration.
MC_DIRECTION.positive	MC_Direction	-	Specify the moving direction - positive
FBErrorOccured	MC_ReadStatus	FALSE	True when an error occurs in the function block

Variable	Data Type	Default	Description
ClearErr	BOOL	FALSE	When FBErrorOccured is True, FB errors can be cleared by triggering SMC_ClearFBError

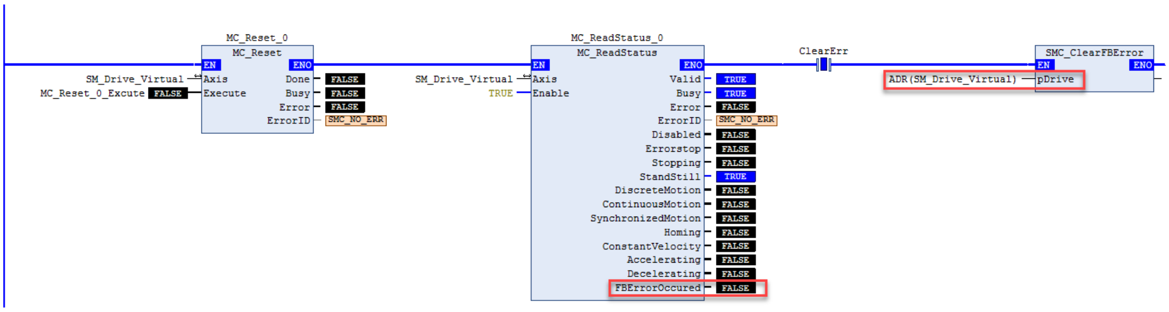
- **Timing Diagram**

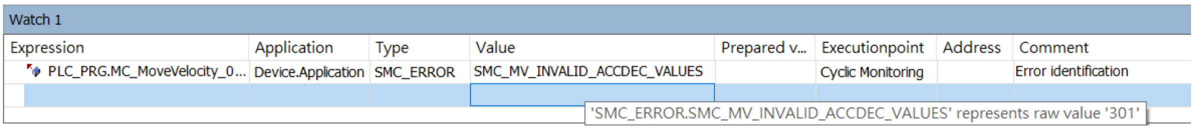


- **LD Language**



Via function block SMC_ClearFBError, the error can be deleted and output FBErrorOccured of MC_ReadStatus would shift to False once an error occurs in the function block. In addition, since the input of SMC_ClearFBError needs to be transferred via a pointer, ADR (input variable) must be added, and use bool to clear the FB error flag.



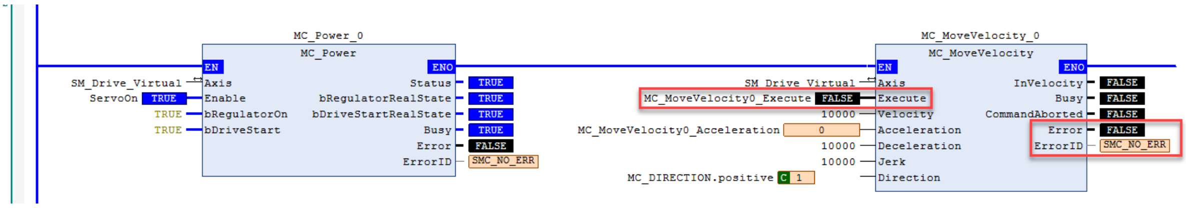


Watch 1

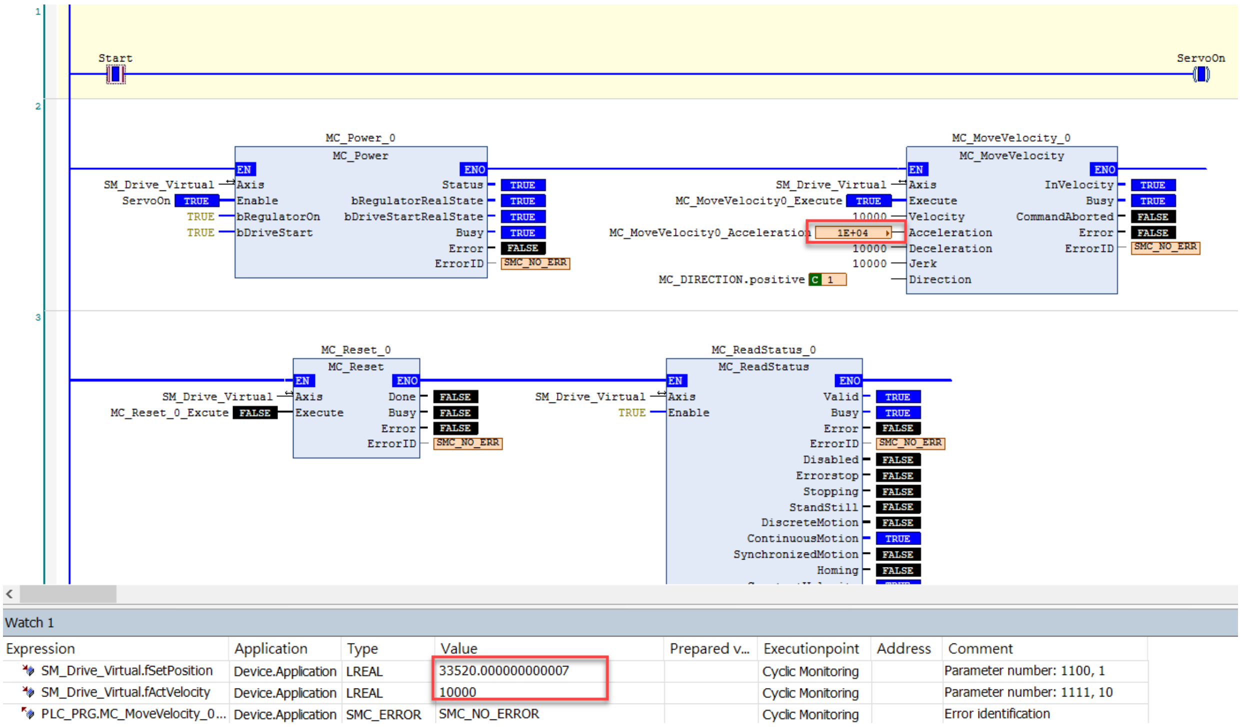
Expression	Application	Type	Value	Prepared v...	Executionpoint	Address	Comment
PLC_PRG.MC_MoveVelocity_0...	Device.Application	SMC_ERROR	SMC_MV_INVALID_ACCDEC_VALUES		Cyclic Monitoring		Error identification

'SMC_ERROR.SMC_MV_INVALID_ACCDEC_VALUES' represents raw value '301'

Disable Execute input of MC_MoveVelocity to update the status of Error output.



Set acceleration of MC_MoveVelocity to 10000 and restart (Execute is True). The output of MC_MoveVelocity would be Busy with values of fSetVelocity and fSetPosition shown on the Watch table under normal operation.



Watch 1

Expression	Application	Type	Value	Prepared v...	Executionpoint	Address	Comment
SM_Drive_Virtual.fSetPosition	Device.Application	LREAL	33520.000000000007		Cyclic Monitoring		Parameter number: 1100, 1
SM_Drive_Virtual.fActVelocity	Device.Application	LREAL	10000		Cyclic Monitoring		Parameter number: 1111, 10
PLC_PRG.MC_MoveVelocity_0...	Device.Application	SMC_ERROR	SMC_NO_ERROR		Cyclic Monitoring		Error identification

- **ST Language**

```
MC_MoveVelocity_0(
    Axis:= SM_Drive_Virtual,
    Execute:= MC_MoveVelocity0_Execute,
    Velocity:= 10000,
    Acceleration:= MC_MoveVelocity0_Acceleration,
    Deceleration:= 10000,
    Jerk:= 10000,
    Direction:= MC_DIRECTION.positive,
    InVelocity=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=>);

MC_ReadStatus_0(
    Axis:= SM_Drive_Virtual,
```

```
    Enable:= TRUE);
```

Set acceleration of MC_MoveVelocity to be 10000 and restart (Execute is True). The output of MC_MoveVelocity would be Busy with values of fSetVelocity and fSetPosition shown on the Watch table under normal operation.

```
 MC_MoveVelocity_0(
    Axis:= SM_Drive_Virtual,
    Execute:= MC_MoveVelocity0_Execute,
    Velocity:= 10000,
    Acceleration:= MC_MoveVelocity0_Acceleration := 10000,
    Deceleration:= 10000,
    Jerk:= 10000,
    Direction:= MC_DIRECTION.positive,
    InVelocity=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );

MC_ReadStatus_0(
    Axis:= SM_Drive_Virtual,
    Enable:= TRUE   );
```

7.8.2.3 Control on Instruction Errors

If an error occurs while executing instruction MC_Power (Servo On), no further action will be taken, while ProgNext indicates whether execution can be moved on. The following examples shows the program in LD and ST programming languages.

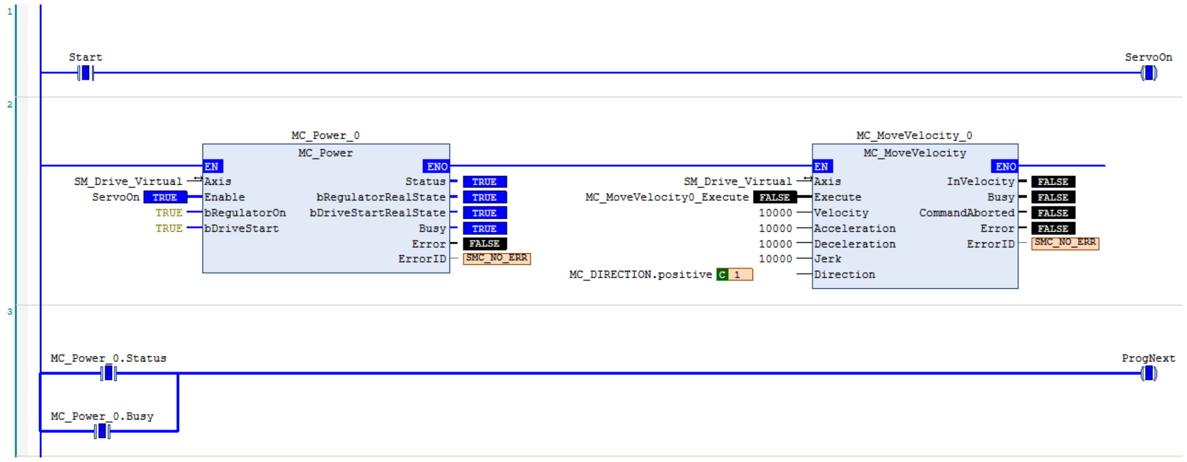
- **Main variables used in programming**

Variable	Data Type	Default	Note
SM_Drive_Virtual	AXIS_REF_SM3	-	Virtual axis variables
ServoOn	BOOL	FALSE	To enable MC_Power
ProgNext	BOOL	FALSE	ProgNext indicator shows whether to take further action
MC_Power_0.Status	BOOL	FALSE	The axis is ready to move when the status is True.
MC_Power_0.Busy	BOOL	FALSE	Execution of the FB has not been completed when the status is True.

- **Timing Diagram**

- **LD Language**

Check if any errors have occurred in MC_Power before the program execution moves onto the next step.



- **ST Language**

```
IF Start THEN
        ServoOn :=TRUE;
        ELSE
        ServoOn :=FALSE;
END_IF

IF (MC_Power_0.Status=TRUE) OR (MC_Power_0.Busy=TRUE) THEN
        ProgNext :=TRUE;
        ELSE
        ProgNext :=FALSE;
END_IF

//MC_Power
MC_Power_0(
        Axis:= SM_Drive_Virtual,
        Enable:= ServoOn,
        bRegulatorOn:= TRUE,
        bDriveStart:= TRUE,
        Status=> ,
        bRegulatorRealState=> ,
        bDriveStartRealState=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );
```

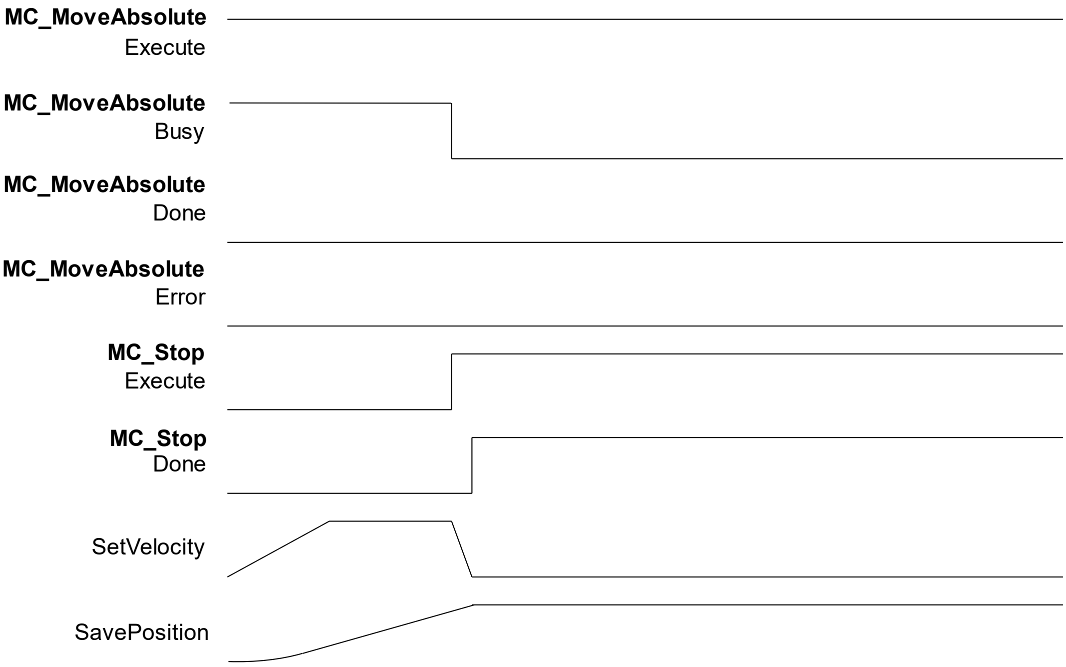
7.8.2.4 Quick Stop for Single Axes

MC_Stop can be used to stop the moving axis when an error occurs during execution of MC_MoveAbsolute instruction. The following examples illustrate the program in LD and ST programming languages respectively.

- **Main variables used in programming**

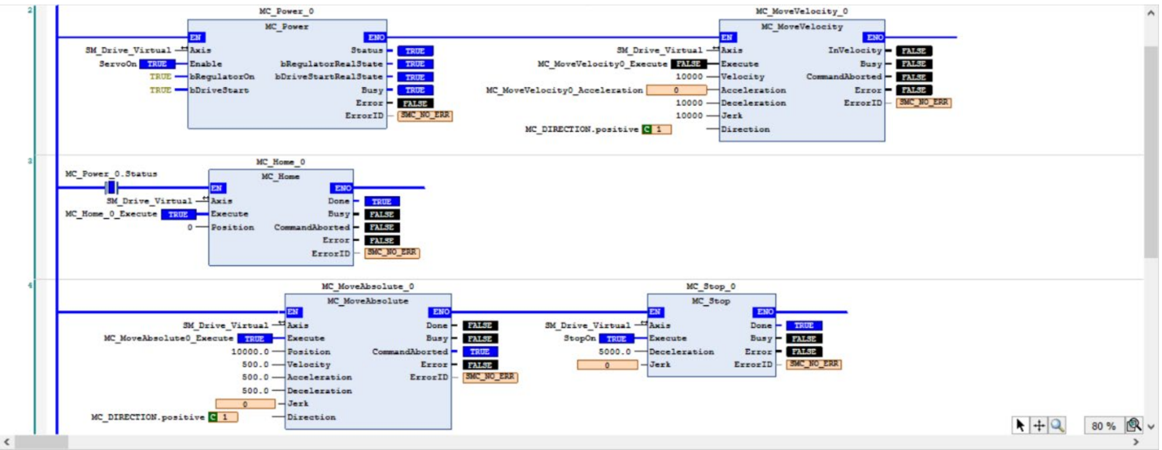
Variable	Data Type	Default	Description
SM_Drive_Virtual	AXIS_REF_SM3	-	Virtual axis variables
ServoOn	BOOL	FALSE	To enable MC_Power
MC_MoveAbsolute0_Execute	BOOL	FALSE	Execute input of MC_MoveAbsolute
MC_DIRECTION.positive	MC_Direction	-	Specify the moving direction-positive (valid for rotary axes)
StopOn	BOOL	FALSE	Activate MC_Stop when the status is True
MC_Stop_0.Done	BOOL	FALSE	Execution of MC_Stop is done when the status is True

- **Timing Diagram**



- **LD Language**

Execute homing under normal output status of MC_Power. Once homing is completed, execute MC_MoveAbsolute. At the same time, MC_Stop can be executed for a quick stop if needed, which would abort MC_MoveAbsolute with state True of CommandAborted output so as to perform a deceleration stop for the axis based on the setting of deceleration, then the Done output of MC_Stop shifts to True after the stop command is completed.



- **ST Language**

The process is same as LD. After MC_Home is done, the state would be Standstill.

```
//MC_Power
MC_Power_0(
    Axis:= SM_Drive_Virtual,
    Enable:= ServoOn,
    bRegulatorOn:= TRUE,
    bDriveStart:= TRUE,
    Status=> ,
    bRegulatorRealState=> ,
    bDriveStartRealState=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );

//MC_Home
IF MC_Power_0.Status THEN
    MC_Home_0(
    Axis:= SM_Drive_Virtual,
    Execute:= MC_Home_0_Execute,
    Position:= 0,
    Done=> ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=>);
END_IF
```

If a quick stop is performed by MC_Stop during execution of MC_MoveAbsolute, MC_MoveAbsolute would be aborted, and the axis would be in Stopping state.

```
//MC_MoveAbsolute & MC_Stop
MC_MoveAbsolute_0(
        Axis:= SM_Drive_Virtual,
        Execute:= MC_MoveAbsolute0_Execute,
        Position:= 10000.0,
        Velocity:= 500.0,
        Acceleration:= 500.0,
        Deceleration:= 500.0,
        Jerk:= ,
        Direction:= MC_DIRECTION.positive,
        Done=> ,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=>);

        MC_Stop_0(
        Axis:= SM_Drive_Virtual,
        Execute:= StopOn,
        Deceleration:= 5000.0,
        Jerk:= ,
        Done=> ,
        Busy=> ,
        Error=> ,
        ErrorID=>);
```

7.8.2.5 Homing

Use homing instruction in the following example to let you understand how to perform the homing operation. Currently, a total of 36 homing modes (0 to 35) are supported and the OD is 6098 (Homing method) /6099sub1(Speed during search for switch) /6099sub2(Speed during search for zero). For more details, please refer to Delta High Resolution AC Servo Drive ASDA-A2 Series User Manual.

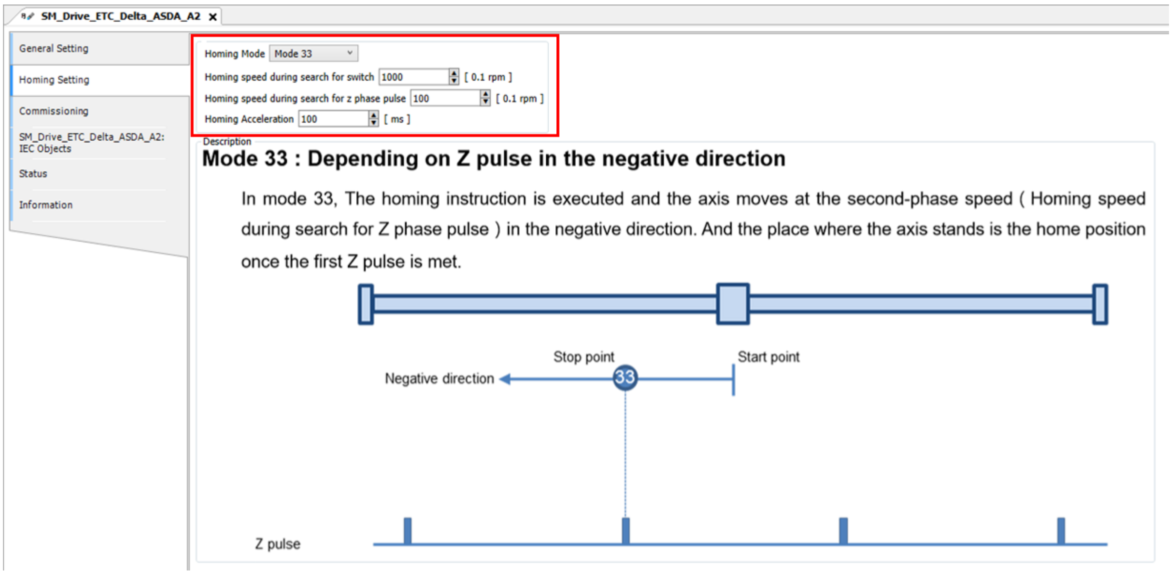
For the following example, specify the parameters of OD as mentioned above after adding A2-E servo in EtherCAT Slave.

Choose mode 33 for Homing Method (Perform homing operation once meeting the first Z pulse.)

Speed during search for switch =1000 (Unit: 0.1rpm) (Search for the limit switch at the speed of 100rmp.)

Speed during search for zero =100 (Unit: 0.1rpm) (Search for zero at the speed of 10rmp.)

After settings are completed, the homing method for executing MC_Home with LD/ ST language would be corresponding to the one specified as above.



Homing Mode: Mode 33

Homing speed during search for switch: 1000 [0.1 rpm]

Homing speed during search for z phase pulse: 100 [0.1 rpm]

Homing Acceleration: 100 [ms]

Description

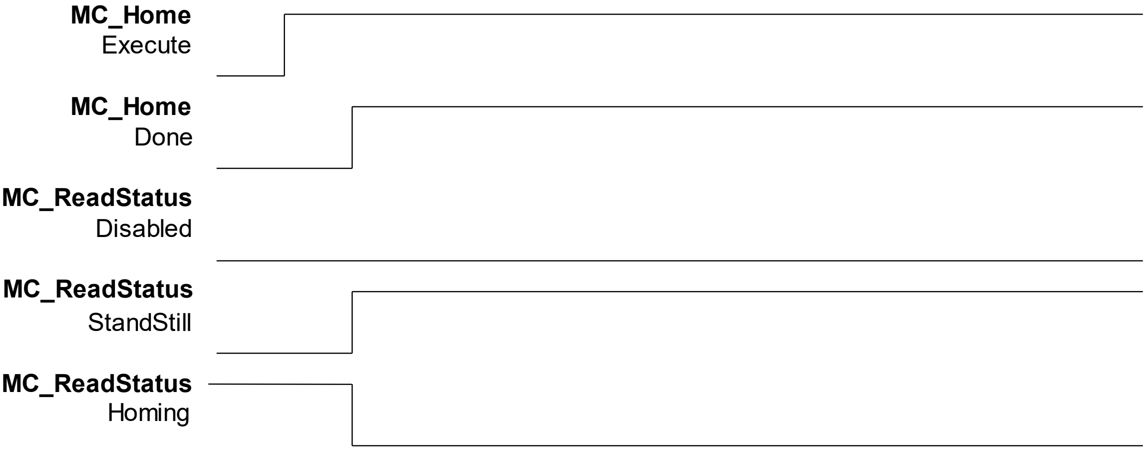
Mode 33 : Depending on Z pulse in the negative direction

In mode 33, The homing instruction is executed and the axis moves at the second-phase speed (Homing speed during search for Z phase pulse) in the negative direction. And the place where the axis stands is the home position once the first Z pulse is met.

- **Main variables used in programming**

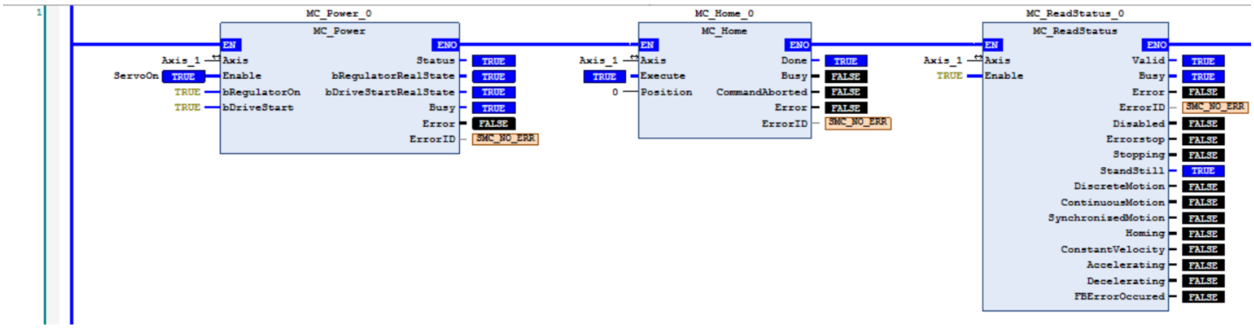
Variable	Data Type	Default	Note
Axis_1	AXIS_REF_SM3	-	Real axis variables
ServoOn	BOOL	FALSE	To enable MC_Power

- **Timing diagram**



- **LD language**

The state is Standstill when the outputs of MC_Power are under normal status. The state is switched to Homing when executing MC_Home, and then goes back to Standstill after home positioning is completed.



- **ST language**

The process is same as LD. The state is Standstill after execution of MC_Home is completed, and the output status can be checked via variables and Watch table.

```
MC_Home_0(
        Axis:= Axis_1,
        Execute:= ,
        Position:= 0,
        Done=> ,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );

MC_ReadStatus_0(
        Axis:= Axis_1,
        Enable:= TRUE,
        Valid=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> ,
        Disabled=> ,
        Errorstop=> ,
        Stopping=> ,
        StandStill=> ,
        DiscreteMotion=> ,
        ContinuousMotion=> ,
        SynchronizedMotion=> ,
        Homing=> ,
        ConstantVelocity=> ,
        Accelerating=> ,
        Decelerating=> ,
        FBErrorOccured=>);
```

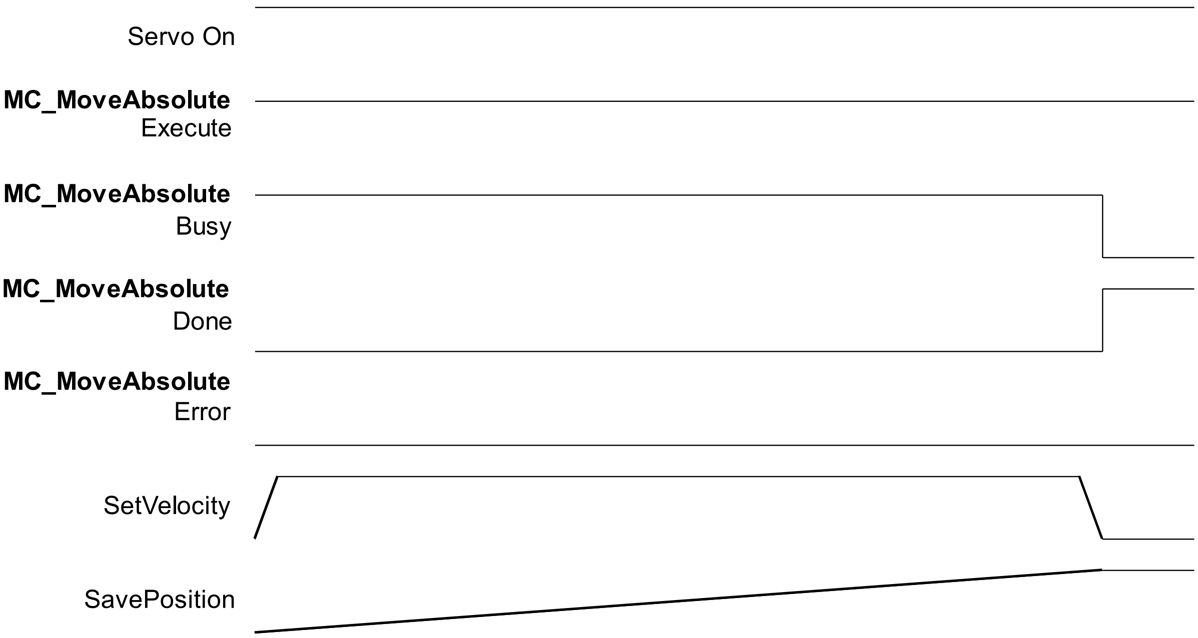
7.8.2.6 Absolute Positioning

Via MC_MoveAbsolute instruction used in the following example, you are able to understand how to perform displacement at one single speed. The following example shows the program in LD and ST programming languages respectively.

- **Main variables used in programming**

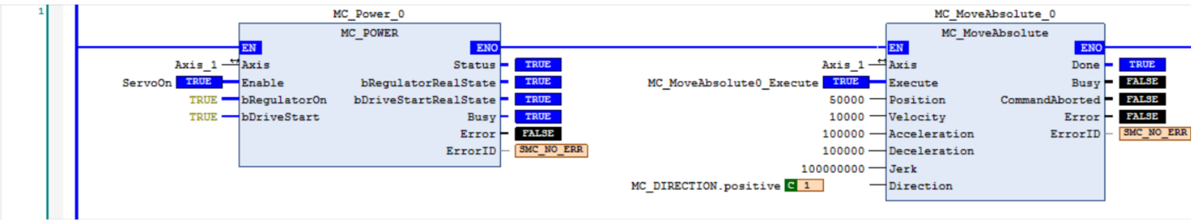
Variable	Data Type	Default	Note
Axis_1	AXIS_REF_SM3	-	Real axis variables
ServoOn	BOOL	FALSE	To enable MC_Power
MC_MoveAbsolute0_Execute	BOOL	FALSE	Execute input of MC_MoveAbsolute
MC_DIRECTION.positive	MC_Direction	-	Specify the moving direction-positive (valid for rotary axes)

- **Timing diagram**



- **LD language**

Check if the output Status of MC_Power is normal, and then execute MC_MoveAbsolute to move from the start position 0 to the target position 50000.



- **ST language**

```
MC_Home_0(
        Axis:= Axis_1,
        Execute:= ,
        Position:= 0,
        Done=> ,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );

MC_MoveAbsolute_0(
        Axis:= Axis_1,
        Execute:= MC_MoveAbsolute0_Execute,
        Position:= 50000,
        Velocity:= 10000,
        Acceleration:= 100000,
        Deceleration:= 100000,
        Jerk:= 100000,
        Direction:= SM3_Basic.MC_DIRECTION.positive,
        Done=> ,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );
```

7.8.2.7 Switch CAM Table during CAM Operation

The following examples illustrate that CAM table can be switched while executing MC_CamIn with the LD and ST programs.

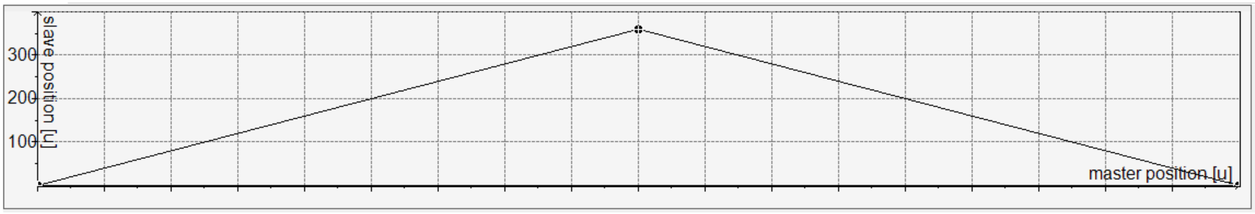
Perform switching between two CAM tables configured with different output parameters by adding master and slave axes as well as using two MC_CamIn instructions. Use CamTable 1 when the command position of master axis is below 3000. Once its position is over 3000, the cam table will be switched to CamTable 2.

- Main variables used in programming

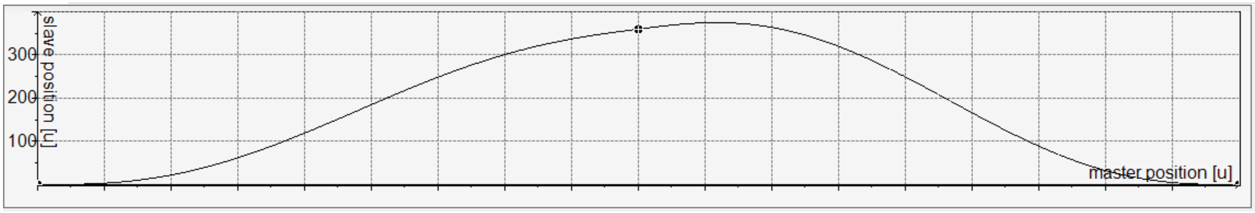
Variable	Data Type	Default	Note
Axis_Master	AXIS_REF_VIRTUAL_SM3	-	Master-related axis variables
Axis_Slave	AXIS_REF_VIRTUAL_SM3	-	Slave-related axis variables
CamTable1	MC_CAM_REF	-	Related variables for Cam table1
CamTable2	MC_CAM_REF	-	Related variables for Cam table2
StartFlag	BOOL	FALSE	If this variable is TRUE and the communication with axes is normal, Servo ON will be activated and continue with further actions.
MC_Power0_Status	BOOL	FALSE	Status output variable of MC_Power for master, TRUE when Servo On
MC_Power1_Status	BOOL	FALSE	Status output variable of MC_Power for slave, TRUE when Servo On

Variable	Data Type	Default	Note
MC_Home0_Done	BOOL	FALSE	Output Done variable of MC_Home for master, TRUE when homing completed.
MC_Home1_Done	BOOL	FALSE	Output Done variable of MC_Home for slave, TRUE when homing completed.
MC_MoveAbs_Busy	BOOL	FALSE	Output Busy variable of MC_MoveAbsolute for master, TRUE when the FB is executed.
CamTableSelect	MC_CAM_REF	-	Specify the corresponding Cam table.
CamTable1_En	BOOL	FALSE	TRUE when CamTable1 is chosen to be used.
CamTable2_En	BOOL	FALSE	TRUE when CamTable2 is chosen to be used.
CamTableID	MC_CAM_ID	-	The internal data structure of the selected Cam table, which is from MC_CamTableSelect and used as input of MC_CamIn.
MC_CamIn1_InSync	BOOL	FALSE	Output InSync variable of CamTable1, TRUE when master and slave axes are synchronized in the e-cam motion.
MC_CamIn2_InSync	BOOL	FALSE	Output InSync variable of CamTable2, TRUE when master and slave axes are synchronized in the e-cam motion.

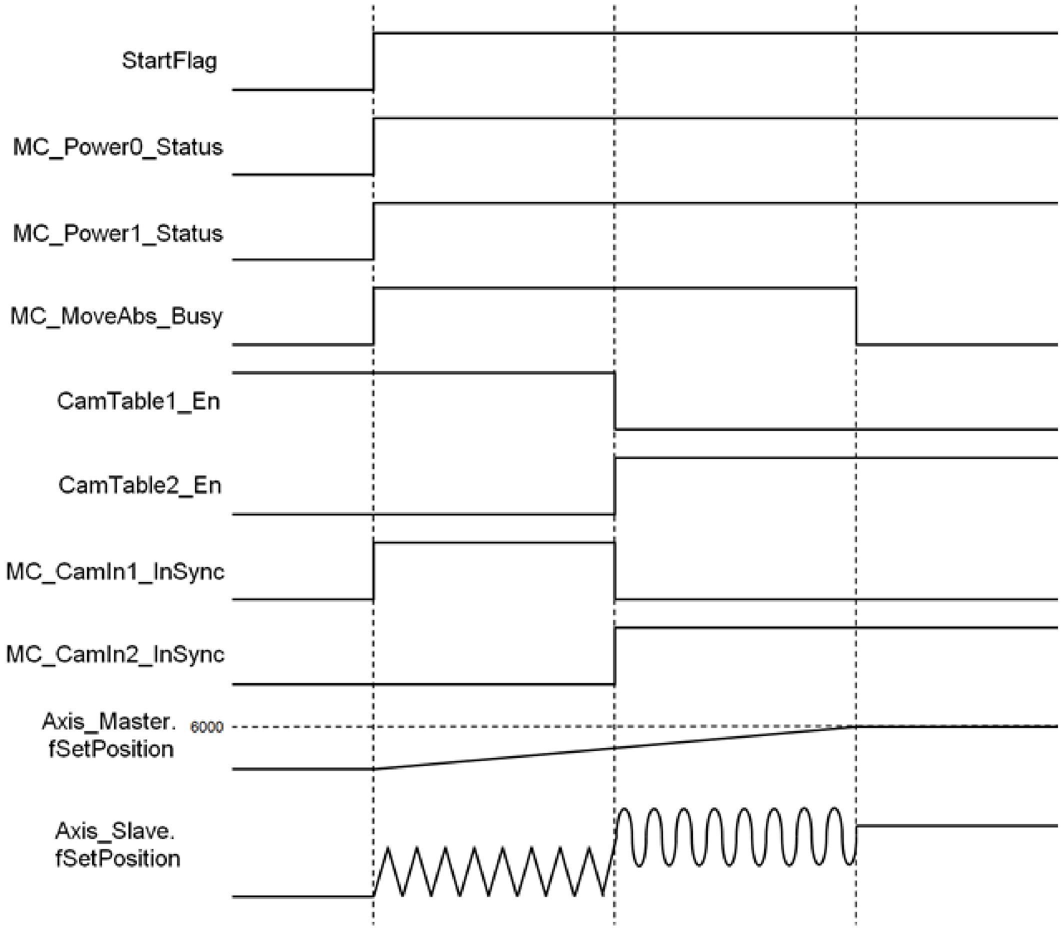
CamTable1 :



CamTable2 :

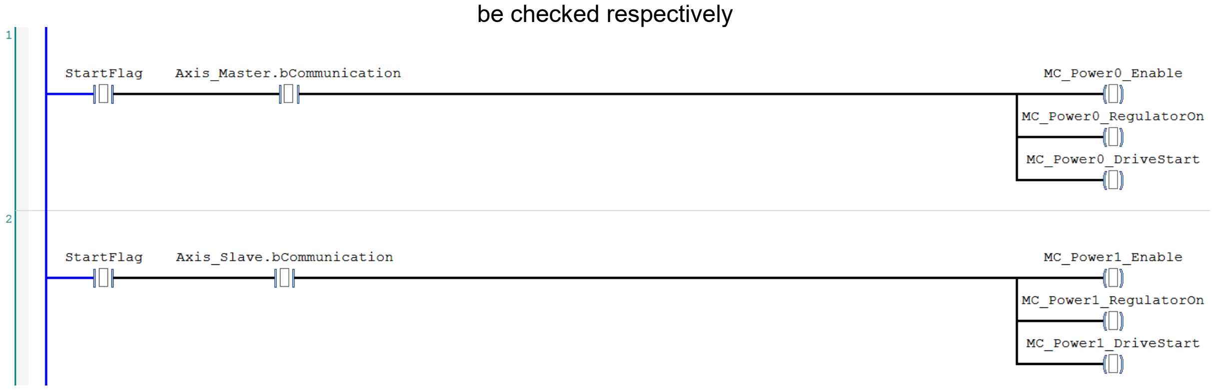


- **Timing diagram**

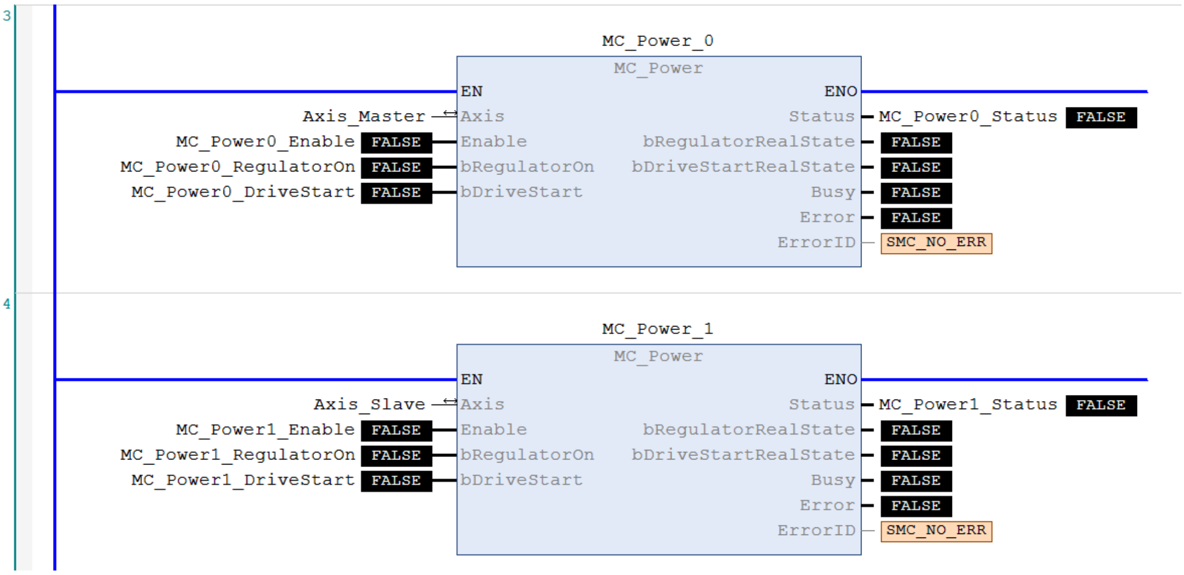


- **LD language**

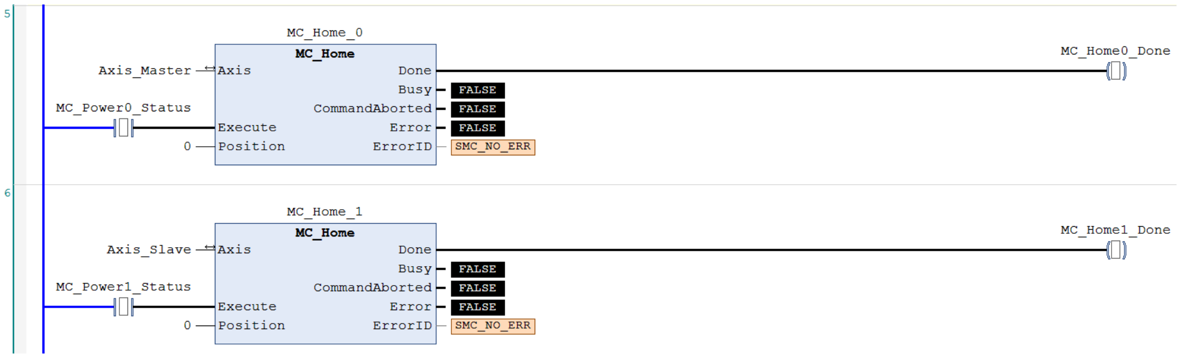
Set StartFlag to TRUE, and then the normal operation of communications for both master and slave axes would be checked respectively



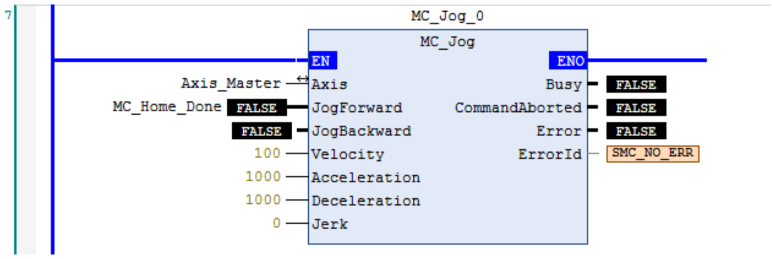
Under normal condition, master and slave axes will be set to Servo ON state.



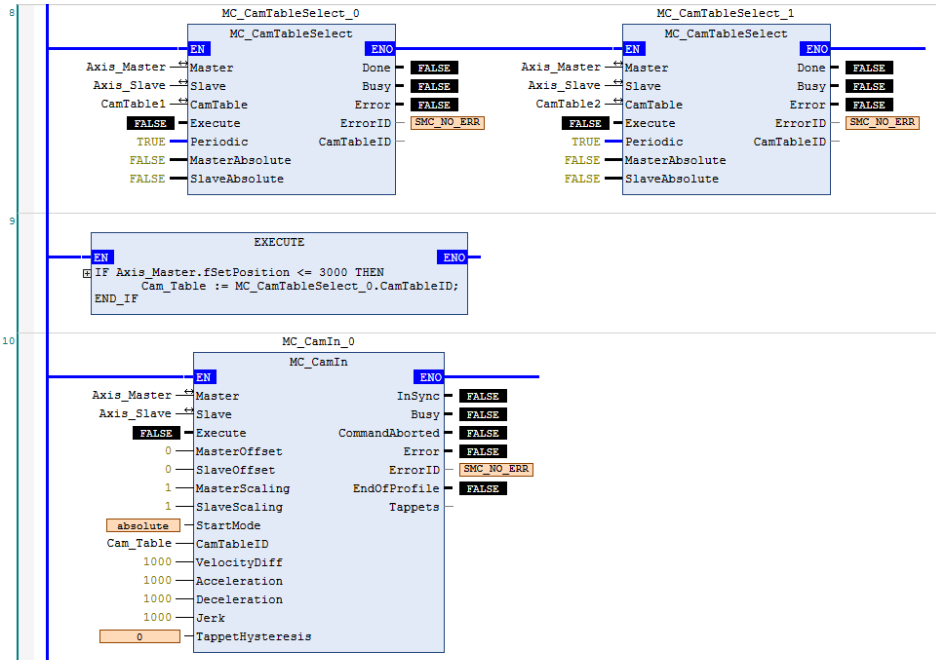
The master and slave axes are under Servo On state and the homes are unsure yet, so homing motions will be performed first.



After the homing of master axis is completed, execute MC_Jog instruction.

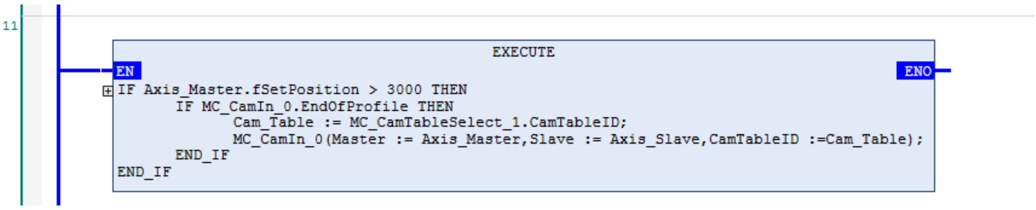


You can use two MC_CamTableSelect to two corresponding CamTables. Set MC_CamTableSelect_0.Execute and MC_CamTableSelect_2.Execute = True and execute MC_CamIn_0.Excute. When the instruction position of master axis is below 3000, use CamTable1 (CamTable1_En=True, CamTable2_En=False).



```
IF Axis_Master.fSetPosition <= 3000 THEN
      Cam_Table := MC_CamTableSelect_0.CamTableID;
END_IF
```

Conversely, when position is over 3000 and used with MC_CamIn.EndofProfile, use CamTable2 (CamTable1_En=False, CamTable2_En=True).



```
IF Axis_Master.fSetPosition > 3000 THEN
        IF MC_CamIn_0.EndOfProfile THEN
                Cam_Table := MC_CamTableSelect_1.CamTableID;
                MC_CamIn_0(Master := Axis_Master,Slave := Axis_Slave,CamTableID :=Cam_Table);
        END_IF
END_IF
```

Make sure that the MC_CamIn_0 function block is executed after the execution of CamIn is completed to prevent missing one cycle of following.

- **ST language**

// Set StartFlag to TRUE, and then the normal operation of communications for both master and slave axes would be checked respectively

```
IF StartFlag = TRUE THEN
        IF Axis_Master.bCommunication = TRUE THEN
                MC_Power0_Enable := TRUE;
                MC_Power0_RegulatorOn := TRUE;
                MC_Power0_DriveStart := TRUE;
        END_IF

        IF Axis_Slave.bCommunication = TRUE THEN
                MC_Power1_Enable := TRUE;
                MC_Power1_RegulatorOn := TRUE;
                MC_Power1_DriveStart := TRUE;
        END_IF
END_IF
```

```
//Under normal condition, master and slave axes will be set to Servo ON state.
MC_Power_0(
        Axis:= Axis_Master,
        Enable:= MC_Power0_Enable,
        bRegulatorOn:= MC_Power0_RegulatorOn,
        bDriveStart:= MC_Power0_DriveStart,
        Status=> MC_Power0_Status,
        bRegulatorRealState=> ,
        bDriveStartRealState=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );

MC_Power_1(
        Axis:= Axis_Slave,
        Enable:= MC_Power1_Enable,
        bRegulatorOn:= MC_Power1_RegulatorOn,
        bDriveStart:= MC_Power1_DriveStart,
        Status=> MC_Power1_Status,
        bRegulatorRealState=> ,
        bDriveStartRealState=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );


// Master and slave axes are in Servo On state and the homes are unsure, so homing motions will be operated first.
IF MC_Power0_Status = TRUE THEN
        MC_Home0_Execute := TRUE;
END_IF

IF MC_Power1_Status = TRUE THEN
        MC_Home1_Execute := TRUE;
END_IF

MC_Home_0(
        Axis:= Axis_Master,
        Execute:= MC_Home0_Execute,
        Position:= 0,
        Done=> MC_Home0_Done,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=>);

MC_Home_1(
        Axis:= Axis_Slave,
        Execute:= MC_Home1_Execute,
        Position:= 0,
        Done=> MC_Home1_Done,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );
```

```
// After the homing operation of master axis is completed, execute MC_Jog instruction.

//MC_Jog(

        Axis:= Axis_Master,
        JogForward:= MC_Home1_Done,

        JogBackward:= ,

        Velocity:= 100,

        Acceleration:= 1000,

        Deceleration:= 1000,

        Jerk:= ,

        Busy=> ,

        CommandAborted=> ,

        Error=> ,

        ErrorId=> );
```

MC_CamTableSelect_0.Execute and MC_CamTableSelect_2.Execute = True, and execute MC_CamIn_0.Excute. When the position of master axis is below 3000, use CamTable1 (CamTable1_En=True, CamTable2_En=False). Set MC_CamTableSelect to two corresponding CamTables

```
MC_CamTableSelect_0(

        Master:= Axis_Master,

        Slave:= Axis_Slave,

        CamTable:= CamTable1

        Execute:= ,

        Periodic:= TRUE,

        MasterAbsolute:= FALSE,

        SlaveAbsolute:= FALSE,

        Done=> MC_CamTableSelect_Done,

        Busy=> ,

        Error=> ,

        ErrorID=> ,

        CamTableID=> CamTableID);


MC_CamTableSelect_1(

        Master:= Axis_Master,

        Slave:= Axis_Slave,

        CamTable:= CamTable2,
```

```
    Execute:= ,
    Periodic:= TRUE,
    MasterAbsolute:= FALSE,
    SlaveAbsolute:= FALSE,
    Done=> MC_CamTableSelect_Done,
    Busy=> ,
    Error=> ,
    ErrorID=> ,
    CamTableID=> CamTableID);

//When the position of master axis is below 3000, use CamTable1.
IF Axis_Master.fSetPosition <= 3000 THEN
    Cam_Table := MC_CamTableSelect_0.CamTableID;
END_IF
```

When the executions of MC_CamTableSelect_0 and MC_CamTableSelect_1 are done, execute MC_CamIn_0.Execute again.

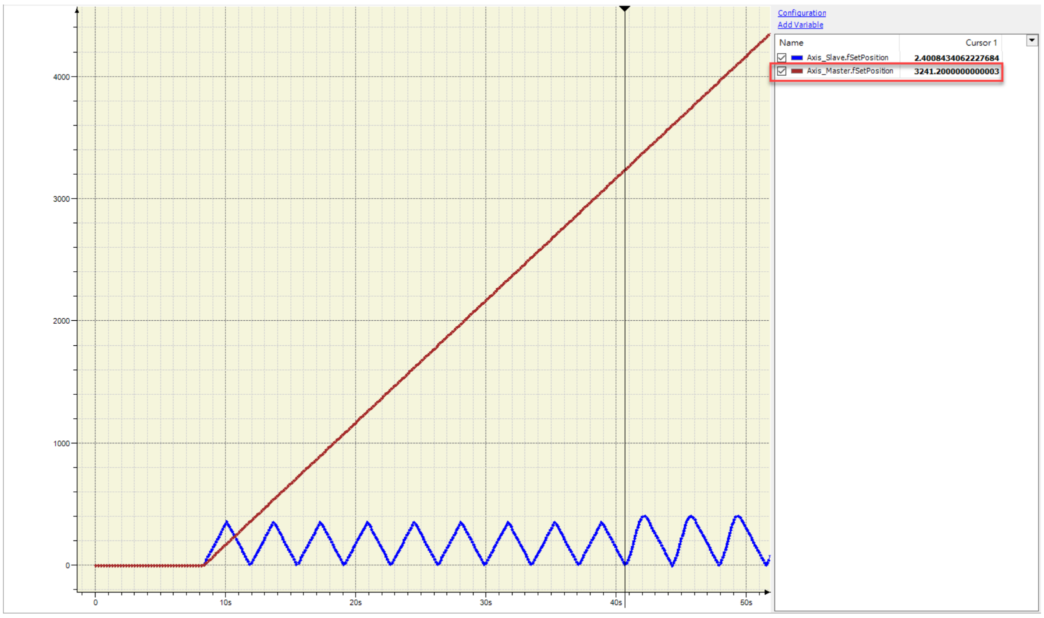
```
MC_CamIn_0(
    Master:= Axis_Master,
    Slave:= Axis_Slave,
    Execute:= TRUE,
    MasterOffset:= 0,
    SlaveOffset:= 0,
    MasterScaling:= 1,
    SlaveScaling:= 1,
    StartMode:= relative,
    CamTableID:= Cam_Table,
    VelocityDiff:= 1000,
    Acceleration:= 1000,
    Deceleration:= 1000,
    Jerk:= ,
    TappetHysteresis:= ,
    InSync=> MC_CamIn1_Insync,
    Busy=> ,
```

```
        CommandAborted=> ,
        Error=> ,
        ErrorID=> ,
        EndOfProfile=> ,
        Tappets=> );
END_IF
```

When the position of master axis is over 3000 and works with MC_CamIN.EndofProfile, change the source to CamTable 2.

```
IF Axis_Master.fSetPosition > 3000 THEN
        IF MC_CamIn_0.EndOfProfile THEN
                Cam_Table := MC_CamTableSelect_1.CamTableID;
                MC_CamIn_0(Master := Axis_Master,Slave := Axis_Slave,CamTableID :=Cam_Table);
        END_IF
END_IF
```

Note: Make sure that the MC_CamIn_0 function block is executed after the execution of CamIn is completed to prevent missing one cycle of following. Switch the Cam tables when the position of master axis is over 3000 accordingly.



Name	Cursor 1
☑ 🟦 Axis_Slave.fSetPosition	2.4008434062227684
☑ 🟥 Axis_Master.fSetPosition	3241.2000000000003

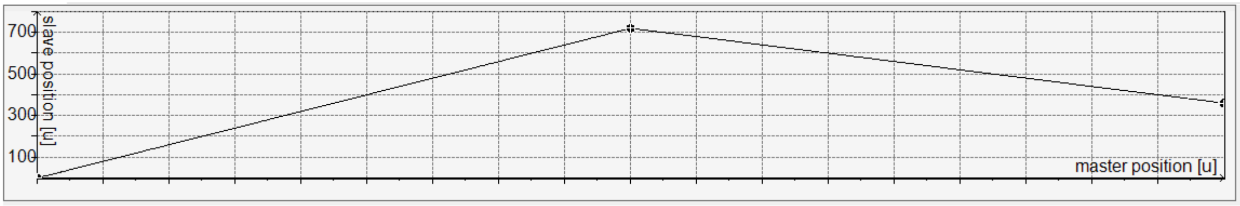
7.8.2.8 Perform Master PhaseOffset for CAM

After the slave axis is aborted during original CAM motion, it starts to synchronize with the controlled master axis. Phase offset of the master axis is operated by executing MC_Phasing when PhasingActive is TRUE, and the slave axis synchronizes with the phase after offset is completed. The following examples illustrate the program in the LD and ST programming languages respectively.

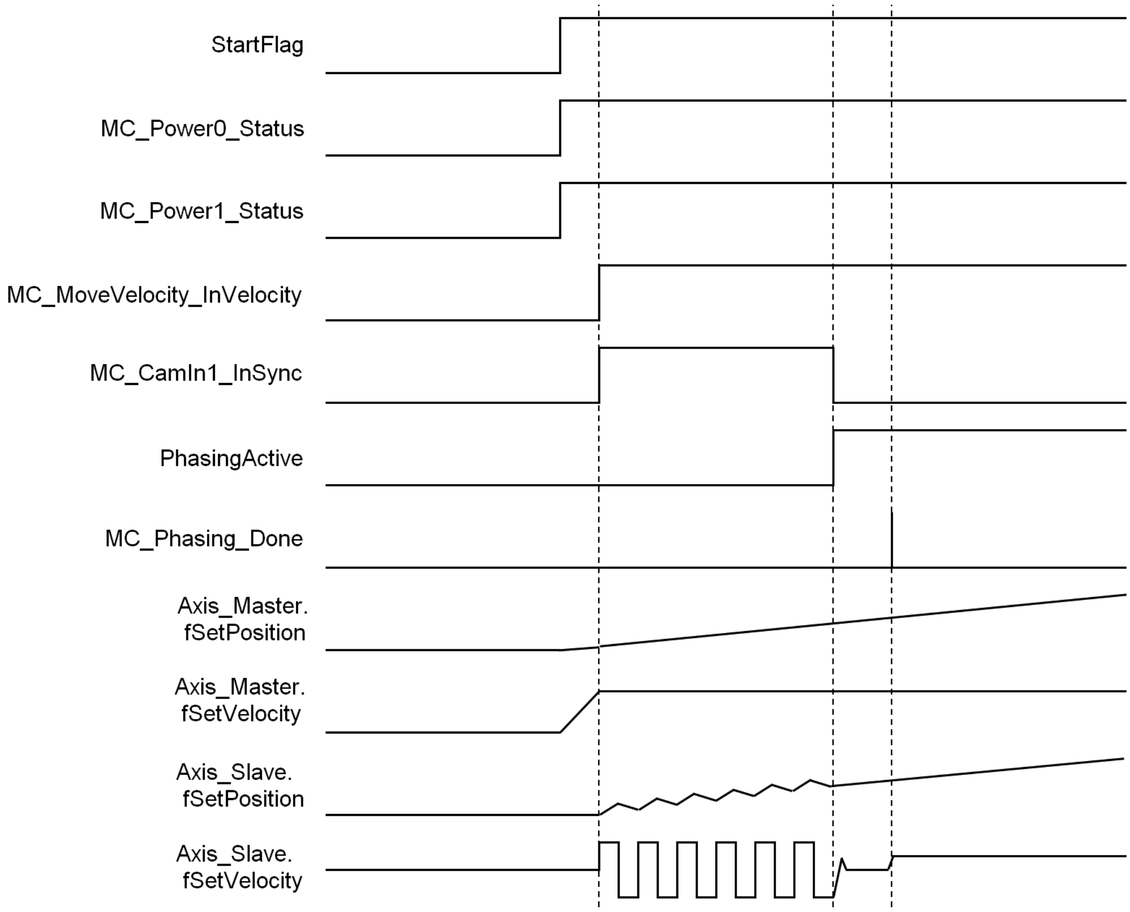
- **Main variables used in programming**

Variable	Data Type	Default	Note
Axis_Master	AXIS_REF_ VIRTUAL_SM3	-	Master-related axis variables.
Axis_Slave	AXIS_REF_ VIRTUAL_SM3	-	Slave-related axis variables.
CamTable	MC_CAM_REF	-	Variables related to Cam table.
StartFlag	BOOL	FALSE	If this variable is TRUE and the communication with axes is normal, Servo ON will be activated and continue with further actions.
MC_Power0_Status	BOOL	FALSE	Status output variable of MC_Power for master, TRUE when Servo On.
MC_Power1_Status	BOOL	FALSE	Status output variable of MC_Power for slave, TRUE when Servo On.
MC_Home0_Done	BOOL	FALSE	Output Done variable of MC_Home for master, TRUE when homing is completed.
MC_Home1_Done	BOOL	FALSE	Output Done variable of MC_Home for slave, TRUE when homing is completed.
MC_MoveVelocity_ Velocity	LREAL	500	The target velocity for master axis to move in constant velocity motion.
MC_MoveVelocity_ InVelocity	BOOL	FALSE	The InVelocity output variable of MC_MoveVelocity, TRUE when the target velocity is reached.
CamTableID	MC_CAM_ID	-	The internal data structure of the selectedCam table, which is from MC_CamTableSelect and used as input of MC_CamIn.
MC_CamIn1_InSync	BOOL	FALSE	Output InSync variable of CamTable1, TRUE when master and slave axes are synchronized in the cam motion.
PhasingActive	BOOL	FALSE	If the variable is TRUE and Cam is InSync, MC_Phasing will start to be executed.
MC_Phasing_PhaseShift	LREAL	500	Specify the phase shift values for the master and slave axes.
MC_Phasing_Velocity	LREAL	300	Specify the relative velocity for phasing operating between the master and slave axis.
MC_Phasing_Done	BOOL	FALSE	The Done output variable of MC_Phasing, TRUE when phase offset is completed.

CamTable：



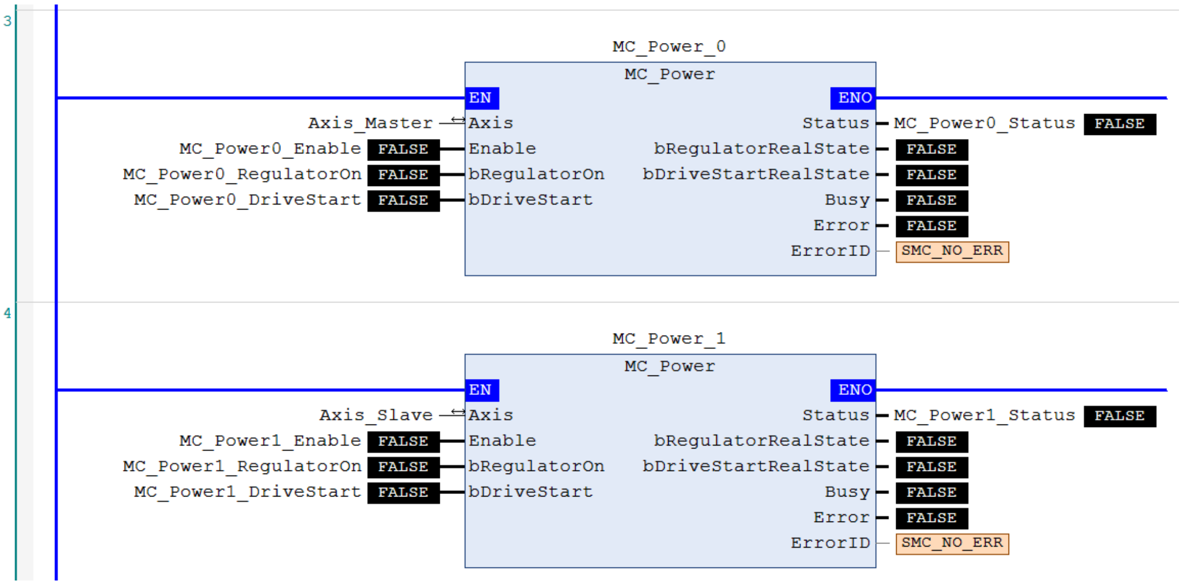
- **Timing diagram**



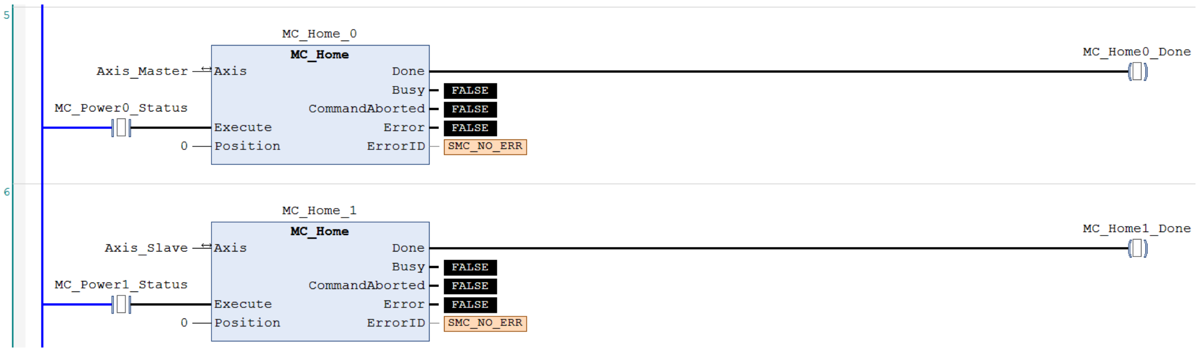
- **LD language**

Set StartFlag to TRUE, and then the normal operation of communications for both master and slave axes would be checked respectively.

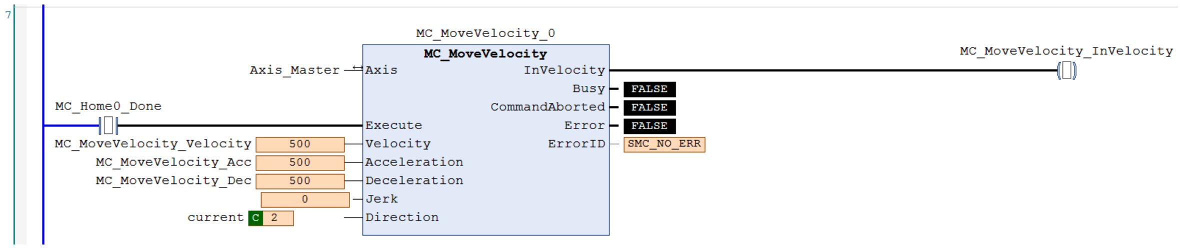
Under normal condition, master and slave axes will be set to Servo ON state.



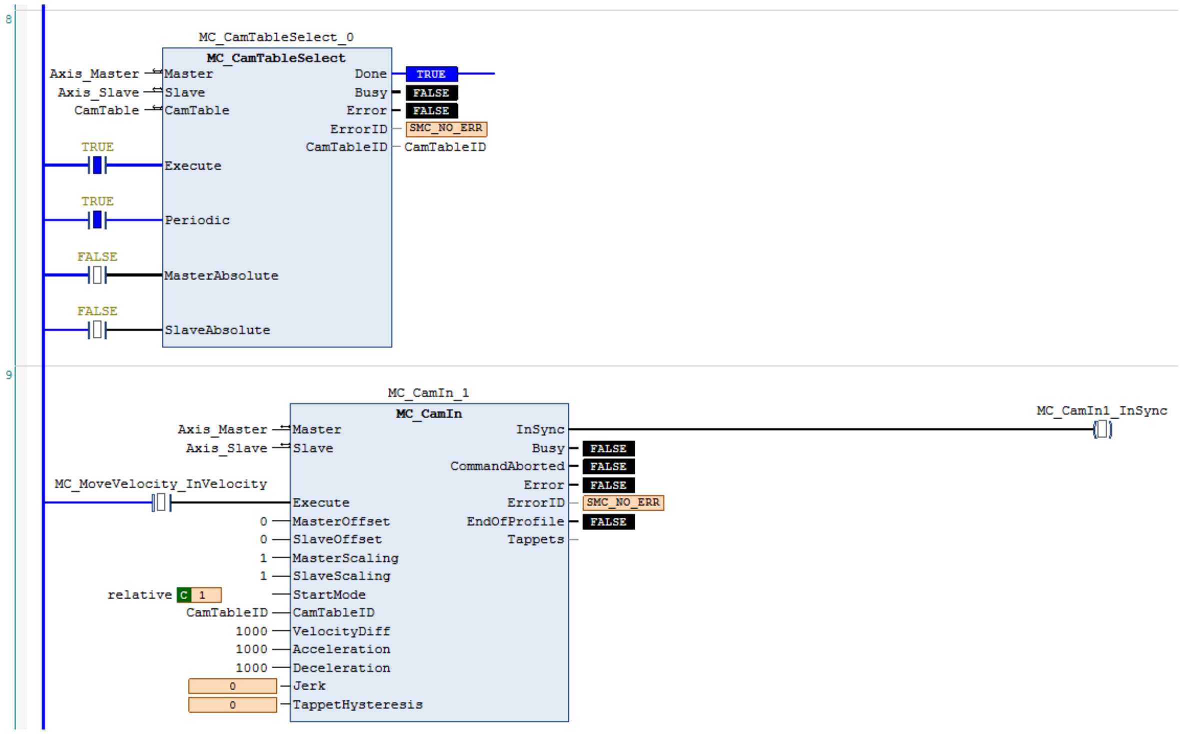
Master and slave axes are under Servo On state and homes are unsure, so homing motions will be operated first.



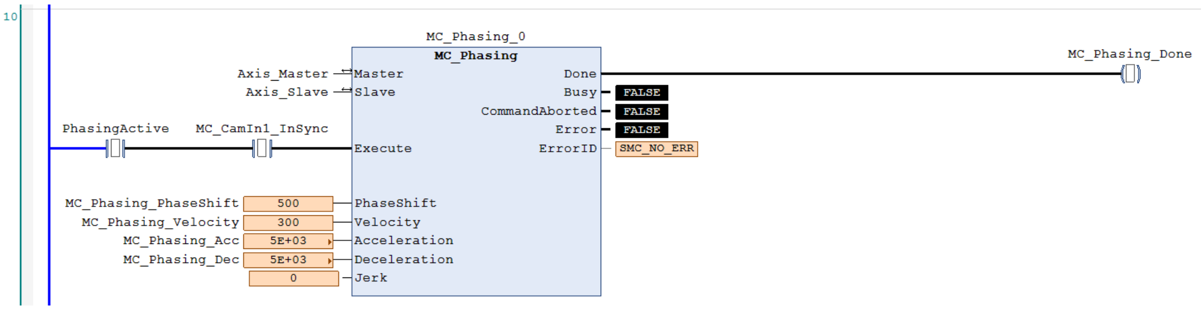
After the homing operation of master axis is completed, execute MC_MoveVelocity.



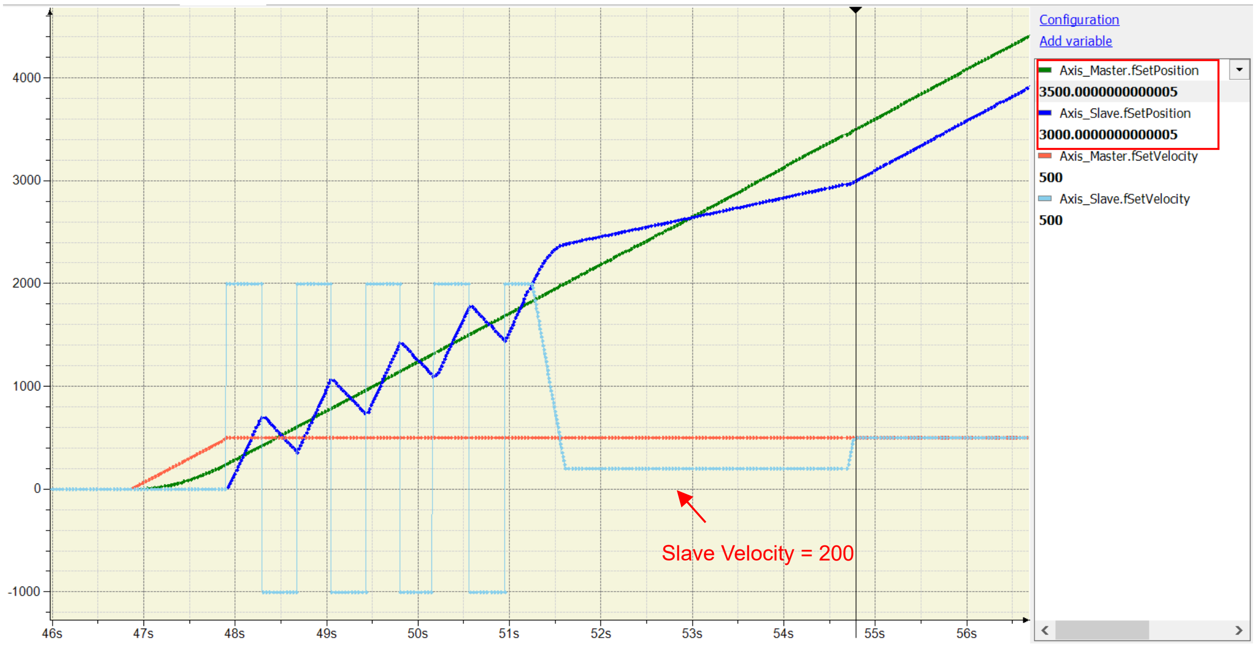
After the master axis reaches the target velocity, execute MC_CamIn with the Cam table specified by MC_CamTableSelect.



If PhasingActive is TRUE and the slave axis is synchronized with the master axis based on the setting of MC_Phasing, master and slave axes starts performing phase offset, which breaks the original master-slave relationship in Cam motion.



According to above setting to perform phase offset of the master axis, the slave axis synchronizes with the phase after offset is completed, and the Phase Shift would be fixed. The Phase Shift (phase difference) between master and slave would be 500, taking the cursor timing 3500 to 3000 as an example, and the velocity of slave axis would be 200 while performing phase offset (that is the velocity of master axis 500 minus velocity 300).



Axis_Master.fSetPosition
3500.0000000000005
Axis_Slave.fSetPosition
3000.0000000000005
Axis_Master.fSetVelocity
500
Axis_Slave.fSetVelocity
500

Slave Velocity = 200

- **ST language**

//Set StartFlag to TRUE, and then the normal operation of communication for both master and slave axes would be checked respectively.r

```
IF StartFlag = TRUE THEN
    IF Axis_Master.bCommunication = TRUE THEN
        MC_Power0_Enable := TRUE;
        MC_Power0_RegulatorOn := TRUE;
        MC_Power0_DriveStart := TRUE;
    END_IF

    IF Axis_Slave.bCommunication = TRUE THEN
        MC_Power1_Enable := TRUE;
        MC_Power1_RegulatorOn := TRUE;
        MC_Power1_DriveStart := TRUE;
    END_IF
END_IF
```

//Under normal condition, master and slave axes will be set to Servo ON state.

```
MC_Power_0(
Axis:= Axis_Master,
Enable:= MC_Power0_Enable,
bRegulatorOn:= MC_Power0_RegulatorOn,
bDriveStart:= MC_Power0_DriveStart,
Status=> MC_Power0_Status,
bRegulatorRealState=> ,
```

```
bDriveStartRealState=> ,
Busy=> ,
Error=> ,
ErrorID=> );

MC_Power_1(
Axis:= Axis_Slave,
Enable:= MC_Power1_Enable,
bRegulatorOn:= MC_Power1_RegulatorOn,
bDriveStart:= MC_Power1_DriveStart,
Status=> MC_Power1_Status,
bRegulatorRealState=> ,
bDriveStartRealState=> ,
Busy=> ,
Error=> ,
ErrorID=> );

//Under Servo On state and with the home positions unsure, homing motions will be operated first
IF MC_Power0_Status = TRUE THEN
MC_Home0_Execute := TRUE;
END_IF

IF MC_Power1_Status = TRUE THEN
MC_Home1_Execute := TRUE;
END_IF

MC_Home_0(
Axis:= Axis_Master,
Execute:= MC_Home0_Execute,
Position:= 0,
Done=> MC_Home0_Done,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );

MC_Home_1(
Axis:= Axis_Slave,
Execute:= MC_Home1_Execute,
Position:= 0,
Done=> MC_Home1_Done,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );


//After the homing operation of master axis is completed, execute MC_MoveVelocity.
MC_MoveVelocity(
        Axis:= Axis_Master,
        Execute:= MC_Home0_Done,
        Velocity:= MC_MoveVelocity_Velocity,
        Acceleration:= MC_MoveVelocity_Acc,
        Deceleration:= MC_MoveVelocity_Dec,
        Jerk:= ,
        Direction:= current,
```

```
        InVelocity=> MC_MoveVelocity_InVelocity,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );
```

// After the master axis reaches the target velocity, execute MC_CamIn with the Cam table specified by MC_CamTableSelect.

```
MC_CamTableSelect(
        Master:= Axis_Master,
        Slave:= Axis_Slave,
        CamTable:= CamTable,
        Execute:= TRUE,
        Periodic:= TRUE,
        MasterAbsolute:= FALSE,
        SlaveAbsolute:= FALSE,
        Done=> MC_CamTableSelect_Done,
        Busy=> ,
        Error=> ,
        ErrorID=> ,
        CamTableID=> CamTableID);

IF MC_MoveVelocity_InVelocity = TRUE THEN
        MC_CamIn_1(
        Master:= Axis_Master,
        Slave:= Axis_Slave,
        Execute:= TRUE,
        MasterOffset:= 0,
        SlaveOffset:= 0,
        MasterScaling:= 1,
        SlaveScaling:= 1,
        StartMode:= relative,
        CamTableID:= CamTableID,
        VelocityDiff:= 1000,
        Acceleration:= 1000,
        Deceleration:= 1000,
        Jerk:= ,
        TappetHysteresis:= ,
        InSync=> MC_CamIn1_Insync,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> ,
        EndOfProfile=> ,
        Tappets=> );
END_IF
```

//If PhasingActive is TRUE and the slave axis is synchronized with the master axis based on the setting of MC_Phasing, master and slave axes start performing phase offset, which breaks the original master-slave relationship in Cam motion.

```
IF (PhasingActive = TRUE) AND (MC_CamIn1_Insync = TRUE) THEN
        MC_Phasing_Execute := TRUE;
END_IF
```

```
MC_Phasing(
        Master:= Axis_Master,
        Slave:= Axis_Slave,
        Execute:= MC_Phasing_Execute,
        PhaseShift:= MC_Phasing_PhaseShift,
        Velocity:= MC_Phasing_Velocity,
        Acceleration:= MC_Phasing_Acc,
        Deceleration:= MC_Phasing_Dec,
        Jerk:= ,
        Done=> MC_Phasing_Done,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );
```

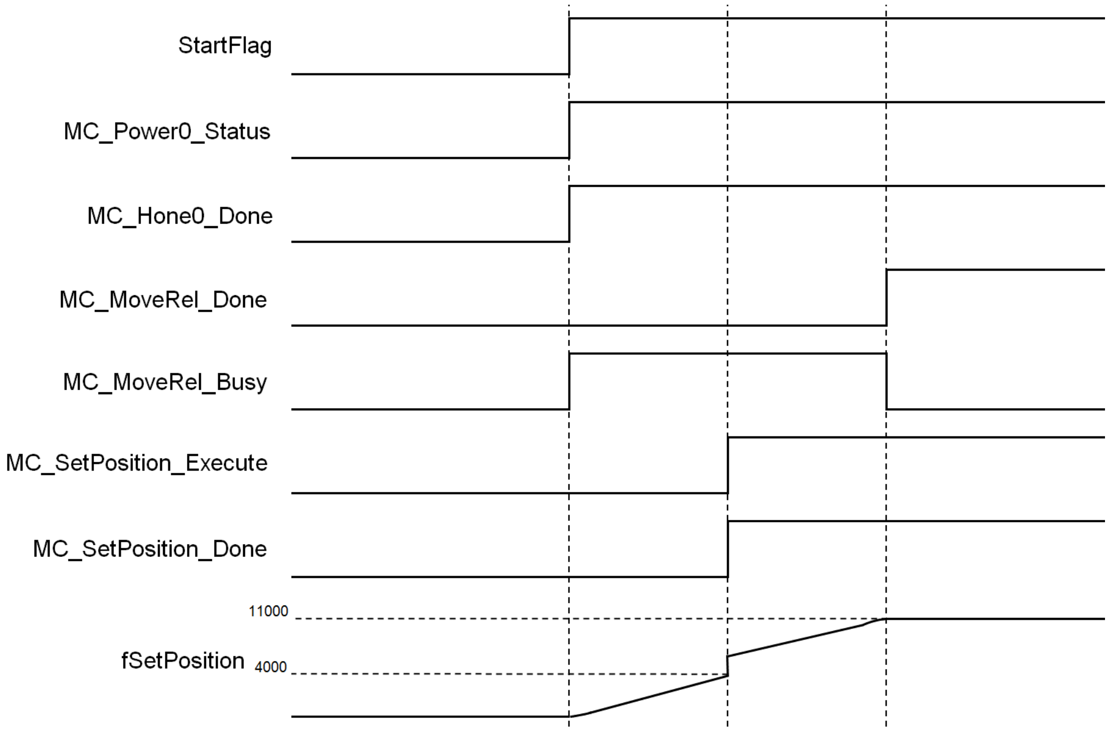
7.8.2.9 Change Current Position in Movement

During motion control, you can change the current position of an axis to the target position in the coordinate system by changing the current feedback position into the specified target position. The interacting effects between MC_MoveRelative and MC_SetPosition are explained in the below example. The following examples illustrate the program in both LD and ST programming languages.

- **Main variables used in programming**

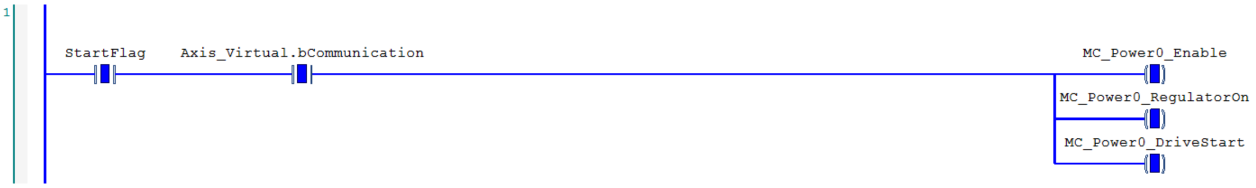
Variable	Data Type	Default	Note
Axis_Virtual	AXIS_REF_ VIRTUAL_ SM3	-	Axis-related variables
StartFlag	BOOL	FALSE	If this variable is TRUE and the communication with the axis is normal, Servo ON will be activated and continue with further actions.
MC_Power0_Status	BOOL	FALSE	Status output variable of MC_Power for master, TRUE when Servo On.
MC_Home0_Done	BOOL	FALSE	Output Done variable of MC_Home for master, TRUE when homing completed.
MC_MoveRel_Distance	LREAL	8000	The target relative position of MC_MoveRelative.
MC_MoveRel_Done	BOOL	FALSE	The output Done variable of MC_MoveRelative. TRUE when the relative positioning is completed.
MC_MoveRel_Busy	BOOL	FALSE	The output Busy variable of MC_MoveRelative, TRUE when the instruction is triggered and executed.
MC_SetPosition_Execute	BOOL	FALSE	If TRUE, MC_SetPosition starts to be executed.
MC_SetPosition_Position	LREAL	3000	The absolute position and relative distance changed by MC_SetPosition.
MC_SetPosition_Mode	BOOL	TRUE	MC_SetPosition is to set the axis position to absolute position or relative position.
MC_SetPosition_Done	BOOL	FALSE	The output Done variable of MC_SetPosition, TRUE when the position is changed.

- **Timing diagram**

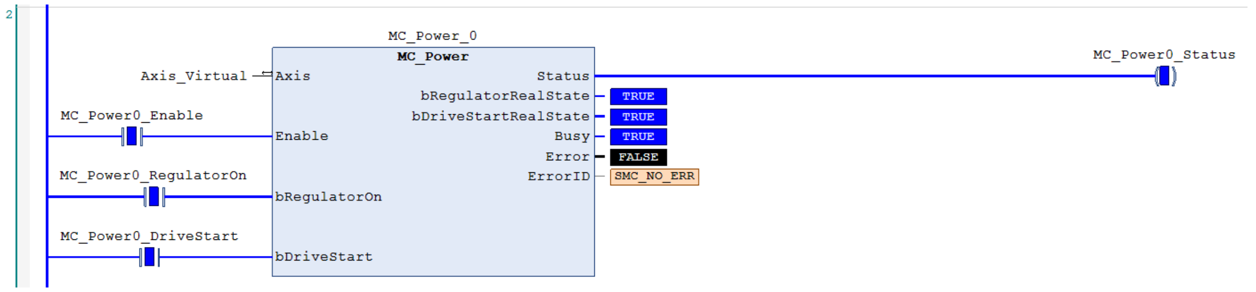


- **LD language**

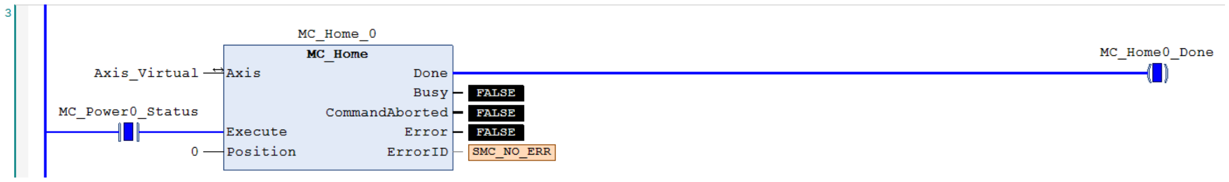
Set StartFlag to TRUE, and then the normal operation of communication for axis would be checked.



Under normal condition, set the axis to be in state Servo On.

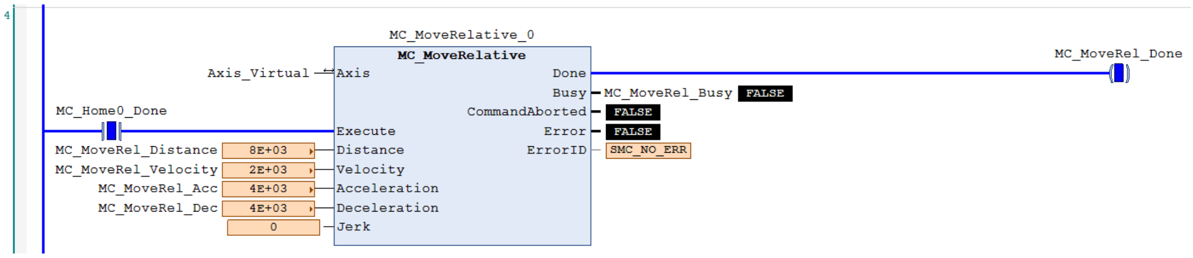


Under Servo On state and with the home position unsure, homing operation will be required.

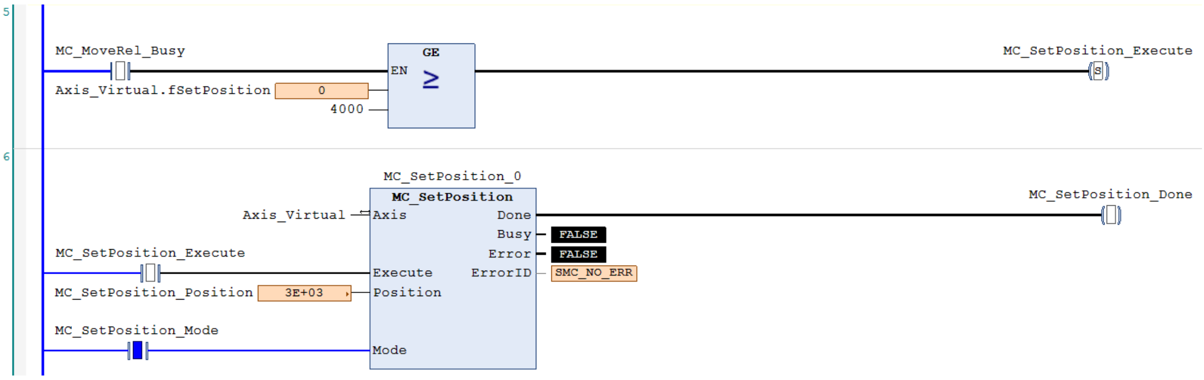


After the homing operation of the axis is completed, execute MC_MoveRelative.

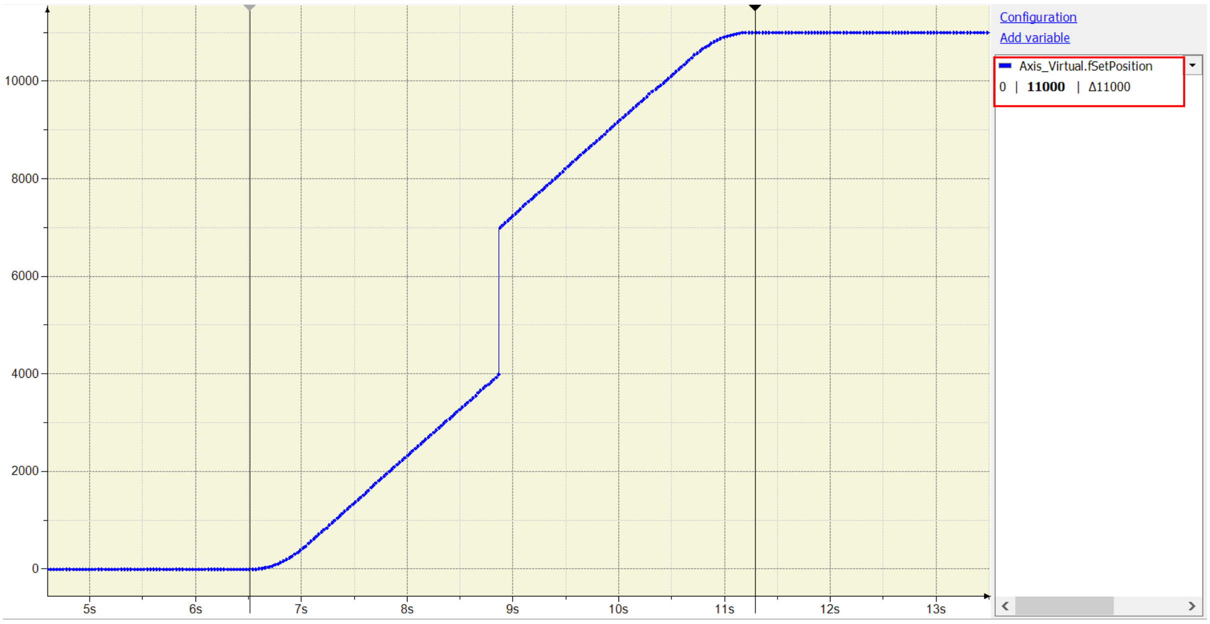
The target position of relative displacement = 8000



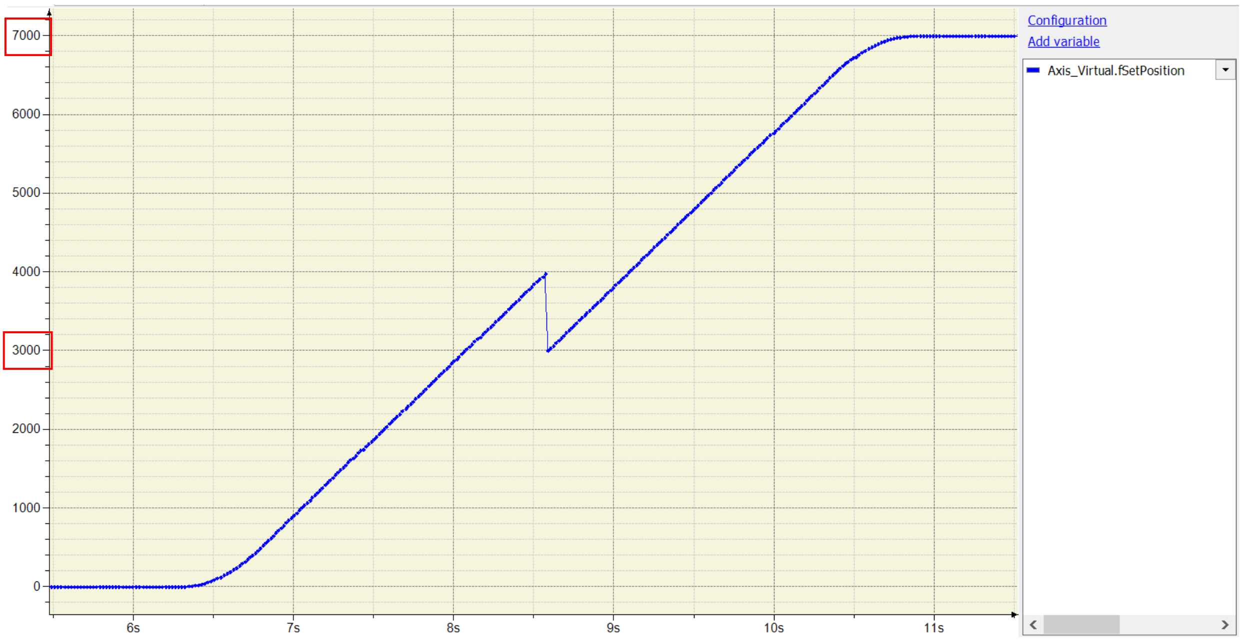
When the current axis position passes 4000, execute MC_SetPosition (Mode = Relative, Distance = 3000) so as to change the current position to the specified target position.



Start a relative positioning procedure based on the current set position in coordinate system according to the above settings, which the position would finally reach 11000 (11000 = 4000 + 3000 + (8000 – 4000)) without influencing the displacement of motion body controlled by MC_MoveRelative. The displacement is 8000 (8000 =(4000 – 0)+(11000 – 7000)) same as the original setting.



The difference between the above and the picture below is that the mode of MC_SetPoition is changed to Absolute (Position = 3000). The actual position is set to the parameterized absolute target Position value, and the position would finally reach 7000 (7000 = 3000+(8000 – 4000)) without influencing the displacement of motion body controlled by MC_MoveRelative. The displacement would be 8000 (8000=(4000 – 0)+(7000 – 3000)) same as the original setting.



- **ST language**

Set StartFlag to TRUE, and then the normal operation of communication for the axis would be checked.

```
IF StartFlag = TRUE THEN
IF Axis_Virtual.bCommunication = TRUE THEN
        MC_Power0_Enable := TRUE;
        MC_Power0_RegulatorOn := TRUE;
        MC_Power0_DriveStart := TRUE;
   END_IF
END_IF
```

```
// Under normal condition, set the axis to Servo On state.

MC_Power_0(
   Axis:= Axis_Virtual,
   Enable:= MC_Power0_Enable,
   bRegulatorOn:= MC_Power0_RegulatorOn,
   bDriveStart:= MC_Power0_DriveStart,
   Status=> MC_Power0_Status,
   bRegulatorRealState=> ,
   bDriveStartRealState=> ,
   Busy=> ,
   Error=> ,
   ErrorID=> );
```

```
//Under Servo On state and with the home position unsure, homing operation will be required.

IF MC_Power0_Status = TRUE THEN
    MC_Home0_Execute := TRUE;
END_IF

MC_Home_0(
    Axis:= Axis_Virtual,
    Execute:= MC_Home0_Execute,
    Position:= 0,
    Done=> MC_Home0_Done,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );


//After the homing operation of axis is completed, execute MC_MoveRelative.

//The target position of relative displacement = 8000

MC_MoveRelative(
    Axis:= Axis_Virtual,
    Execute:= MC_Home0_Done,
    Distance:= MC_MoveRel_Distance,
    Velocity:= MC_MoveRel_Velocity,
    Acceleration:= MC_MoveRel_Acc,
    Deceleration:= MC_MoveRel_Dec,
    Jerk:= ,
    Done=> MC_MoveRel_Done,
    Busy=> MC_MoveRel_Busy,
    CommandAborted=> ,
    Error=> ,
    ErrorID=> );


//When the current position of the axis passes 4000, execute MC_SetPosition (Mode = Relative, Distance = 3000) so
as to change the current position into the specified target position.

IF (MC_MoveRel_Busy = TRUE) AND (Axis_Virtual.fSetPosition >= 4000) THEN
    MC_SetPosition_Execute := TRUE;
END_IF

MC_SetPosition(
    Axis:= Axis_Virtual,
    Execute:= MC_SetPosition_Execute,
    Position:= MC_SetPosition_Position,
    Mode:= MC_SetPosition_Mode,
    Done=> MC_SetPosition_Done,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

7.8.2.10 Perform Superimposed during Gear Engagement

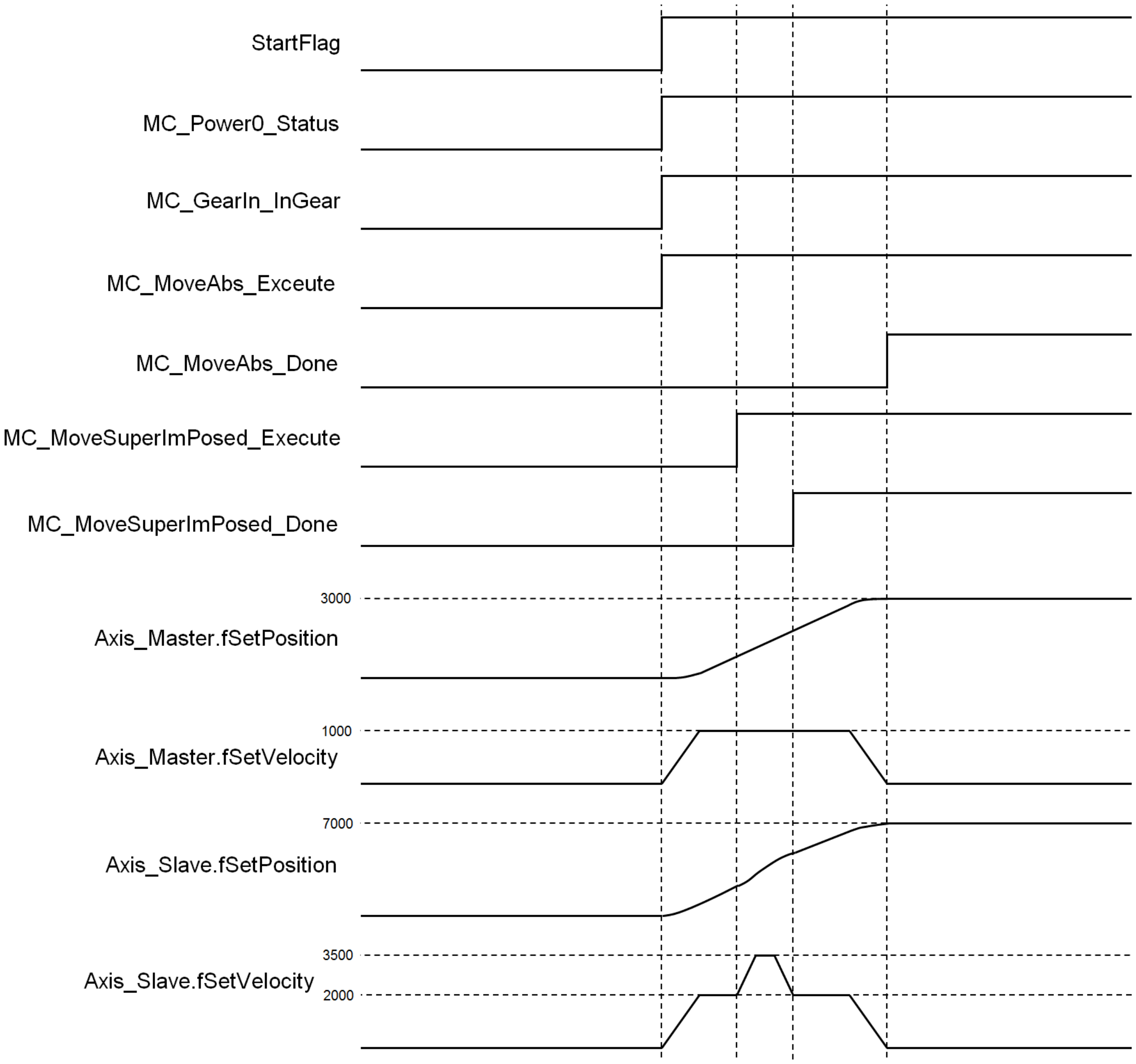
Perform MC_MoveSuperImposed on the particular slave axis while the gear has been engaged in the following example. The final position of slave axis would be the displacement of gear ratio relative to master axis, plus the specific distance superimposed in motion. The following examples illustrate the program in both LD and ST programming languages respectively.

- **Main variables used in programming**

Variable	Data Type	Default	Note
Axis_Master	AXIS_REF_ VIRTUAL_SM3	-	Master-related axis variables
Axis_Slave	AXIS_REF_ VIRTUAL_SM3	-	Slave-related axis variables
StartFlag	BOOL	FALSE	If this variable is TRUE and the communication with axes is normal, Servo ON will be activated and continue with further actions.
MC_Power0_Status	BOOL	FALSE	Status output variable of MC_Power for master, TRUE when Servo On.
MC_Power1_Status	BOOL	FALSE	Status output variable of MC_Power for slave, TRUE when Servo On.
MC_Home0_Done	BOOL	FALSE	Output Done variable of MC_Home for master, TRUE when homing operation is completed.
MC_Home1_Done	BOOL	FALSE	Output Done variable of MC_Home for slave, TRUE when homing operation is completed.
MC_GearIn_InGear	BOOL	FALSE	Output InGear variable of MC_GearIn. TRUE when the engage operation is completed.
MC_GearIn_RatioNumerator	DINT	2	Numerator of the gear ratio between master and slave axes.
MC_GearIn_RatioDenominator	UDINT	1	Denominator of the gear ratio between master and slave axes.
MC_MoveAbs_Execute	BOOL	FALSE	When the variable is TRUE, MC_MoveAbsolute is executed.
MC_MoveAbs_Position	LREAL	3000	Specify the absolute target position of master axis.
MC_MoveAbs_Velocity	LREAL	1000	Specify the target velocity of master axis.
MC_MoveAbs_Done	BOOL	FALSE	Output Done variable of MC_MoveAbsolute for master, TRUE when absolute positioning is completed.
MC_MoveAbs_Busy	BOOL	FALSE	Output Busy variable of MC_MoveAbsolute for master axis. TRUE when the instruction is executed.
MC_MoveSuperImposed_ Execute	BOOL	FALSE	When the variable is TRUE, MC_MoveSuperImposed is executed.

Variable	Data Type	Default	Note
MC_MoveSuperImposed_Done	BOOL	FALSE	Output Done variable of MC_MoveSuperImposed for slave axis. TRUE when the superimposed movement is completed.
MC_MoveSuperImposed_Distance	LREAL	1000	Specify the superimposed displacement of slave axis.
MC_MoveSuperImposed_VelocityDiff	LREAL	1500	Specify the relative velocity to the master axis while the superimposed movement is operating on the slave axis.

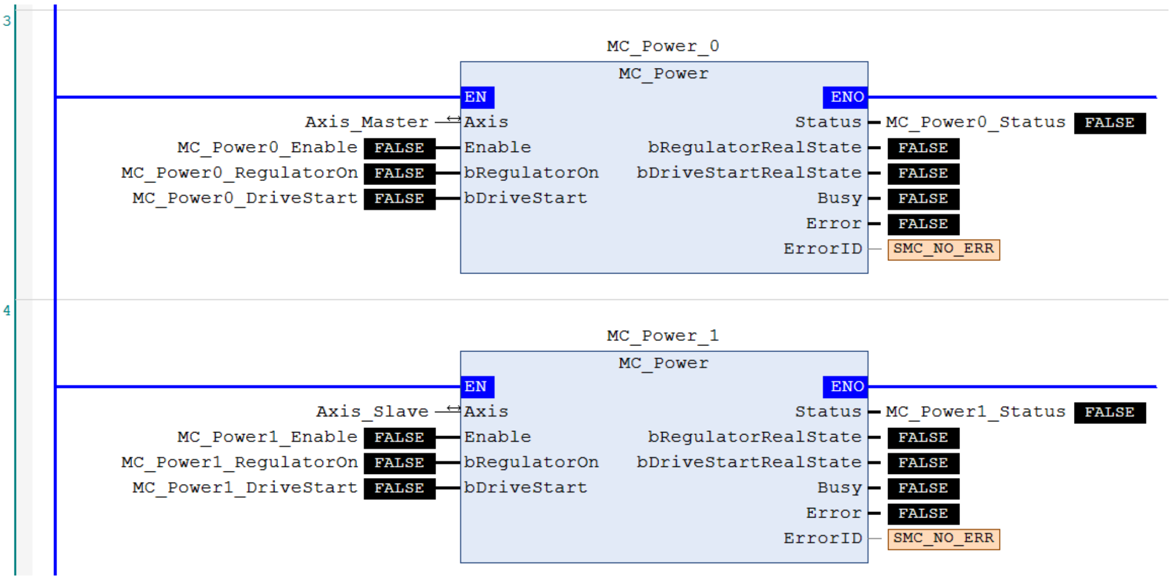
- **Timing diagram**



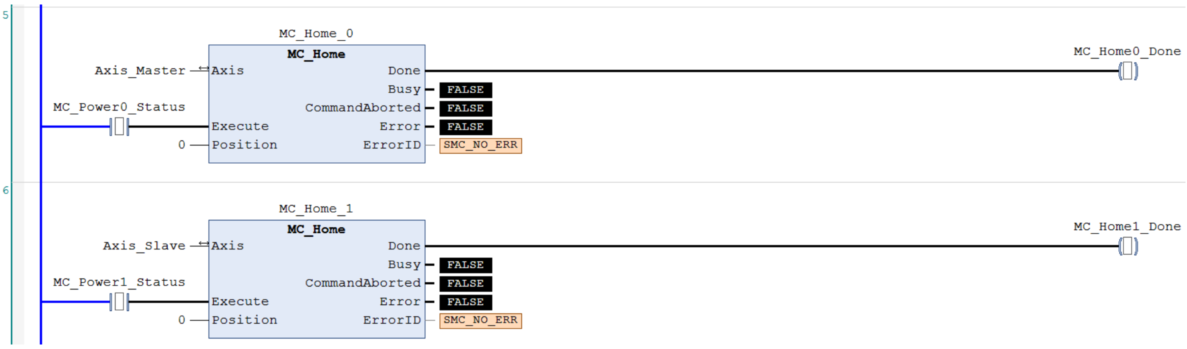
- **LD language**

Set StartFlag to TRUE, and then the normal operation of communications for both master and slave axes would be checked respectively.

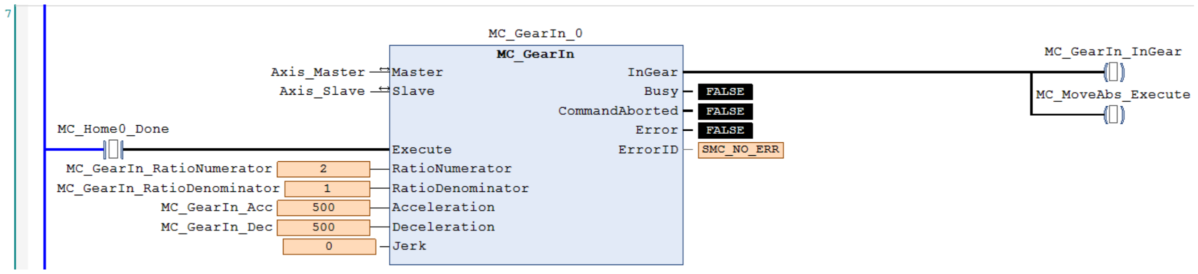
Under normal condition, master and slave axes will be set to Servo ON state.



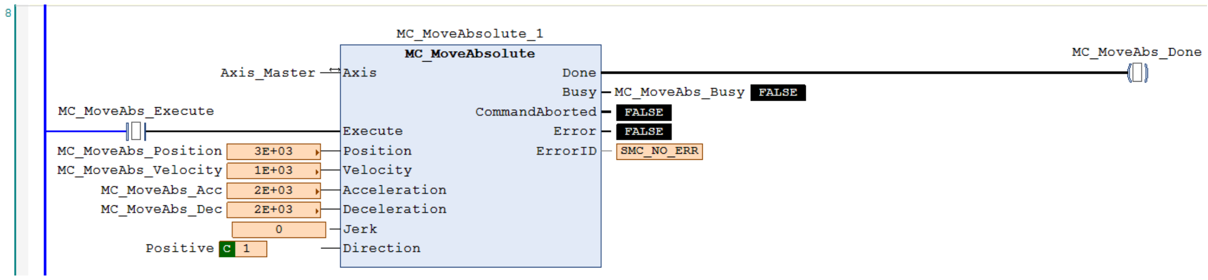
Master and slave axes are in Servo On state and the home positions are unsure, so homing operations will be required.



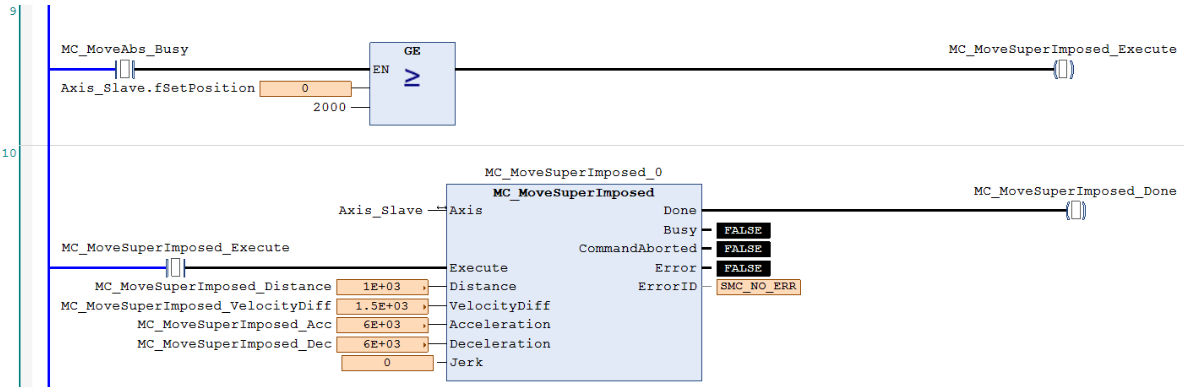
After the homing operation is completed, execute MC_GearIn to activate a master-slave coupling (gear coupling).



Right after the engage action is completed with output InGear done, execute MC_MoveAbsolute to the master axis.



At the same time, when the slave axis moves to the preset triggering position=2000 based on the coupling relationship, MC_MoveSuperImposed would be executed- the slave axis would further move a specified superimposed distance after reaching the original preset target position.



According to the above settings, the slave axis would move a displacement based on the gear ratio relative to the master axis, and then further move a specified superimposed distance with the specified velocity for superimposed motion to reach the final target position.

The moving distance of master axis is 3000 and the original target position of slave axis would be 6000 calculated with the gear ratio 1:2. Therefore, the final target position of slave axis will change to be 7000 (6000+1000) with an extra superimposed distance=1000. While coupling, the velocities of master and slave axes are respectively 1000 and 2000. Yet the velocity of slave axis changes to 3500 while superimposing (the original velocity 2000+ VelocityDiff 1500).

- **ST language**

Set StartFlag to be TRUE, and then the normal operation of communications for both master and slave axes would be checked respectively.

```
IF StartFlag = TRUE THEN
    IF Axis_Master.bCommunication = TRUE THEN
        MC_Power0_Enable := TRUE;
        MC_Power0_RegulatorOn := TRUE;
        MC_Power0_DriveStart := TRUE;
    END_IF

    IF Axis_Slave.bCommunication = TRUE THEN
        MC_Power1_Enable := TRUE;
        MC_Power1_RegulatorOn := TRUE;
        MC_Power1_DriveStart := TRUE;
    END_IF
END_IF
```

Under normal condition, master and slave axes will be set to Servo ON state.

```
 MC_Power_0(
    Axis:= Axis_Master,
    Enable:= MC_Power0_Enable,
    bRegulatorOn:= MC_Power0_RegulatorOn,
    bDriveStart:= MC_Power0_DriveStart,
    Status=> MC_Power0_Status,
    bRegulatorRealState=> ,
    bDriveStartRealState=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

```
MC_Power_1(
        Axis:= Axis_Slave,
        Enable:= MC_Power1_Enable,
        bRegulatorOn:= MC_Power1_RegulatorOn,
        bDriveStart:= MC_Power1_DriveStart,
        Status=> MC_Power1_Status,
        bRegulatorRealState=> ,
        bDriveStartRealState=> ,
        Busy=> ,
        Error=> ,
        ErrorID=> );
```

Master and slave axes are in Servo On state and the home positions are unsure, so homing motions will be performed.

```
IF MC_Power0_Status = TRUE THEN
        MC_Home0_Execute := TRUE;
END_IF

IF MC_Power1_Status = TRUE THEN
        MC_Home1_Execute := TRUE;
END_IF

MC_Home_0(
        Axis:= Axis_Master,
        Execute:= MC_Home0_Execute,
        Position:= 0,
        Done=> MC_Home0_Done,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );

MC_Home_1(
        Axis:= Axis_Slave,
        Execute:= MC_Home1_Execute,
        Position:= 0,
        Done=> MC_Home1_Done,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );
```

After the homing operations are completed, execute MC_GearIn to activate a master-slave coupling (gear coupling).

```
MC_GearIn(
        Master:= Axis_Master,
        Slave:= Axis_Slave,
```

```
        Execute:= MC_Home0_Done,
        RatioNumerator:= MC_GearIn_RatioNumerator,
        RatioDenominator:= MC_GearIn_RatioDenominator,
        Acceleration:= MC_GearIn_Acc,
        Deceleration:= MC_GearIn_Dec,
        Jerk:= ,
        InGear=> MC_GearIn_InGear,
        Busy=> ,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );

IF MC_GearIn_InGear = TRUE THEN
        MC_MoveAbs_Execute := TRUE;
END_IF
```

Right after the engage action is completed with output InGear done, execute MC_MoveAbsolute to the master axis.

```
MC_MoveAbsolute(
        Axis:= Axis_Master,
        Execute:= MC_MoveAbs_Execute,
        Position:= MC_MoveAbs_Position,
        Velocity:= MC_MoveAbs_Velocity,
        Acceleration:= MC_MoveAbs_Acc,
        Deceleration:= MC_MoveAbs_Dec,
        Jerk:= ,
        Direction:= Positive,
        Done=> MC_MoveAbs_Done,
        Busy=> MC_MoveAbs_Busy,
        CommandAborted=> ,
        Error=> ,
        ErrorID=> );
```

At the same time, when the slave axis reaches the preset triggering position=2000 based on the coupling relationship, MC_MoveSuperImposed would be executed - the slave axis would move a specified superimposed distance after the original preset target position.

```
IF MC_MoveAbs_Busy = TRUE THEN
        IF Axis_Slave.fSetPosition >= 2000 THEN
                MC_MoveSuperImposed_Execute := TRUE;
        END_IF
END_IF

MC_MoveSuperImposed(
        Axis:= Axis_Slave,
        Execute:= MC_MoveSuperImposed_Execute,
        Distance:= MC_MoveSuperImposed_Distance,
        VelocityDiff:= MC_MoveSuperImposed_VelocityDiff,
        Acceleration:= MC_MoveSuperImposed_Acc,
```

```
Deceleration:= MC_MoveSuperImposed_Dec,
Jerk:= ,
Done=> MC_MoveSuperImposed_Done,
Busy=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

MEMO

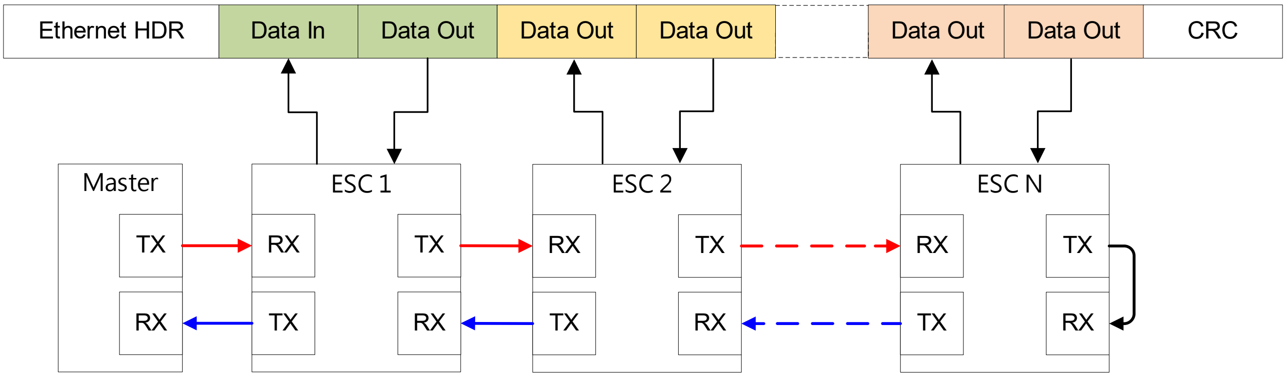
Chapter 8 Communication

Table of Contents

8.1 Introduction on EtherCAT Communication

8.1.1 Features of EtherCAT Fieldbus

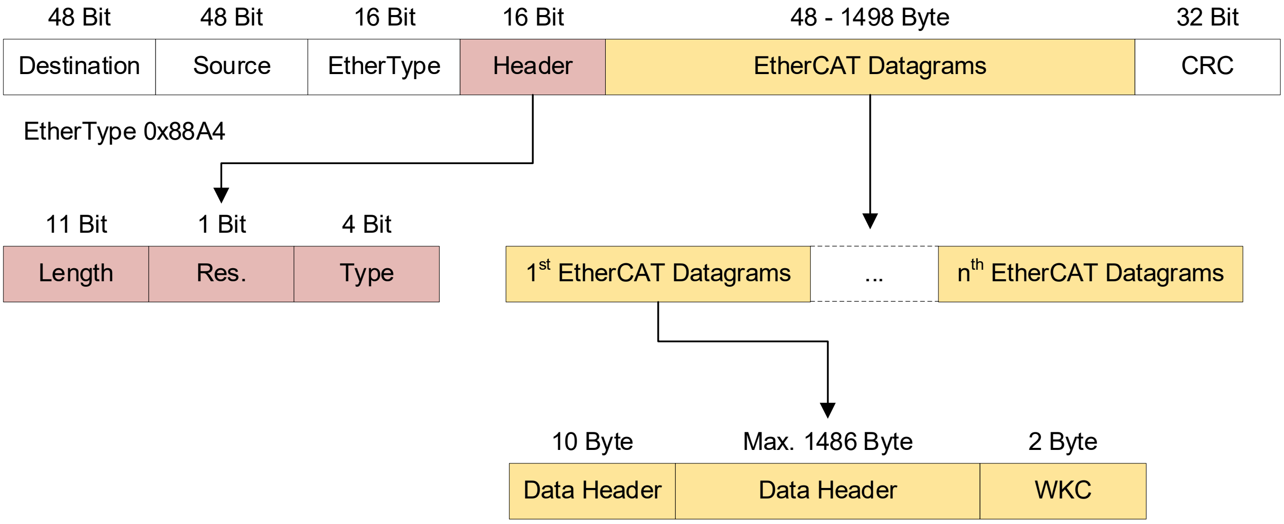
The EtherCAT bus is the Ethernet-based fieldbus. The communication rate of the EtherCAT network is 100Mbps and the distance between two adjacent nodes is within 50 meters. The EtherCAT network is noticeably very different from the general Ethernet network. One EtherCAT network has just one EtherCAT master and EtherCAT slaves contain ESC chips (EtherCAT Slave Controller) specially used for processing EtherCAT communication data and inserting the data which slaves need to transmit to the master into the EtherCAT frame. The last EtherCAT slave in the network will return the data which have been handled to the master in chronological order. See the illustration of data transmission shown below. Thanks to the ESC chips in slaves, the master can make a communication with all slaves in an EtherCAT data frame and thus the communication efficiency is enhanced.



- **EtherCAT Communication between the Controller and Slaves**

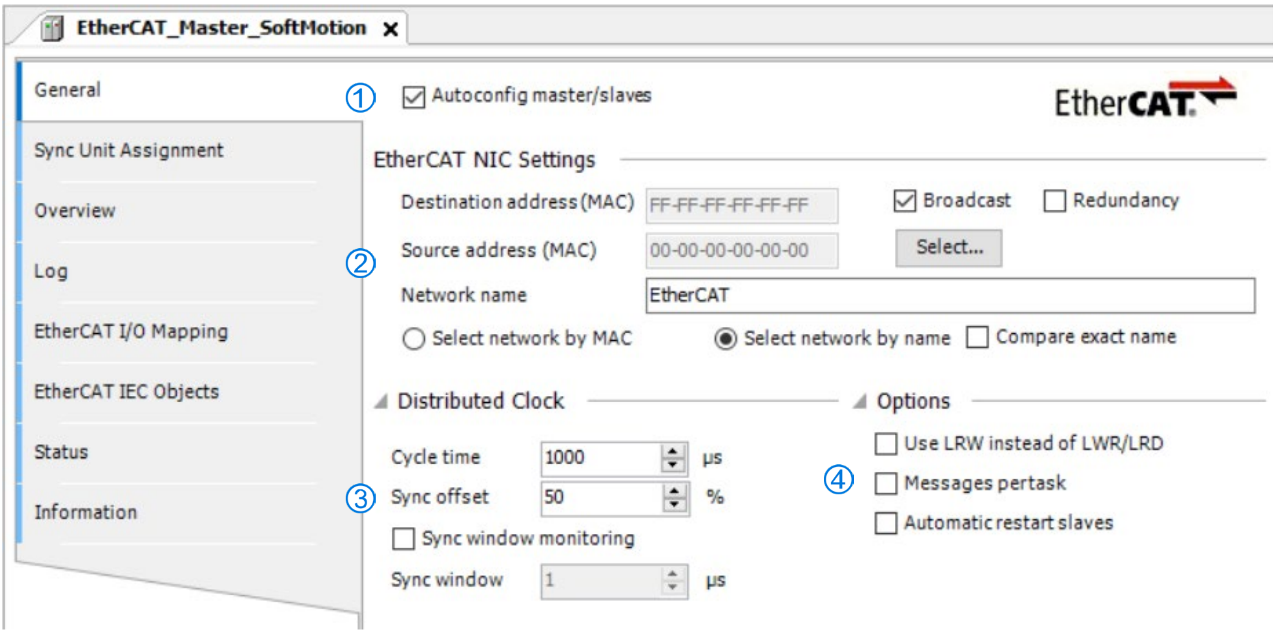
  Since the EtherCAT bus is the EtherNet-based fieldbus, the EtherCAT data frame still adopts the UDP/IP Ethernet data frame structure. EtherCAT data field includes 2 bytes of EtherCAT data header and 44 to1498 bytes of EtherCAT data. EtherCAT Data field consists of one or more EtherCAT datagrams. EtherCAT Data can be defined and analyzed in a protocol as long as the master and slaves comply with the protocol. Currently the mostly used two protocols are COE (CANopen Over EtherCAT) and FOE (Filetransfer Over EtherCAT).

  EtherCAT data frame structure is as displayed below.



8.1.2 Settings up EtherCAT Master

This section introduces functions in the tab of EtherCAT_Master_SoftMotion.



☑ Autoconfig master/slaves

EtherCAT NIC Settings

Destination address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast ☐ Redundancy

Source address (MAC) 00-00-00-00-00-00

Network name EtherCAT

○ Select network by MAC ● Select network by name ☐ Compare exact name

Distributed Clock

Cycle time 1000 µs

Sync offset 50 %

☐ Sync window monitoring

Sync window 1 µs

Options

☐ Use LRW instead of LWR/LRD

☐ Messages pertask

☐ Automatic restart slaves

- **General**

① Autoconfig Master/Slaves: Enable this option to have basic configurations done. Suggested to use this option.

② EtherCAT NIC Setting

- Destination address (MAC): MAC address of the device in the EtherCAT network that is to receive the telegrams.
- Source address (MAC): MAC address of the controller (Select CPSW1 when you use Browse… to find Slave)
- Network Name: Name or MAC of the network, depending on which of the following options is activated:
- Select network by Name: Network is identified by the network name and the project is device-independent.

Note: The EtherCAT communication port of AX-5 series PLC is identified as EtherCAT.

③ Distributed Clock

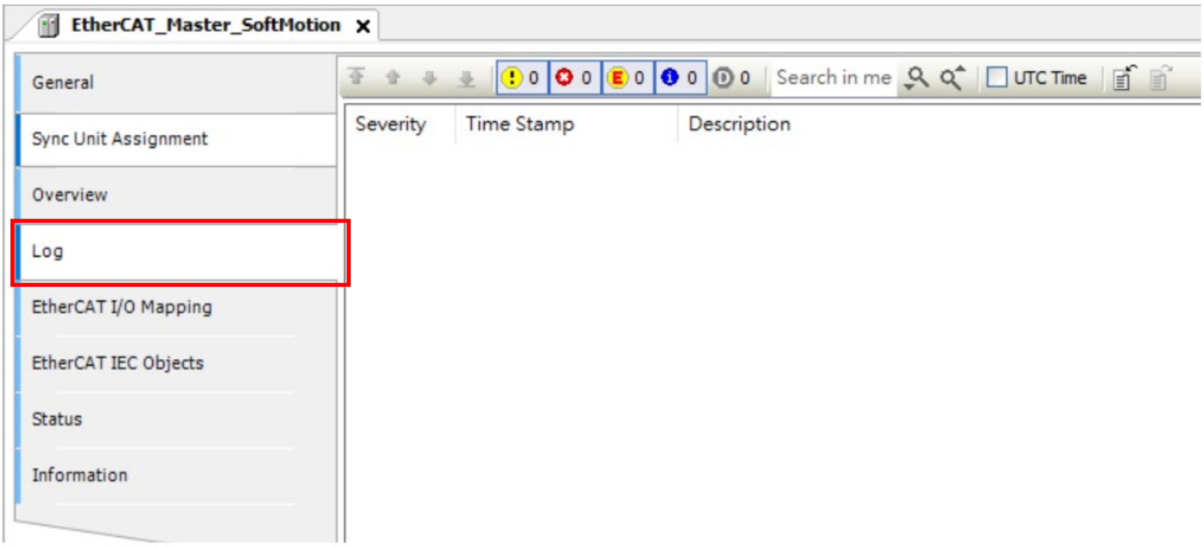
- Cycle time: Master sends out corresponding data to the Slaves in a cycle time specified here. The setting of the cycle time also applies to the interval setting of EtherCAT_Task.
- Sync offset: Setting the delay time between the react time for PLC cycle time and time to stop SYNC for EtherCAT slaves. For instance, if the sync offset is configured to 20%, then the PLC cycle is postponed by 20% of the cycle time in relation to the EtherCAT slave's SYNC interrupt. This ensures that the PLC cycle retains 80% of the cycle time margin, and as long as the PLC cycle is delayed within this margin, it will not result in any loss of synchronized data.
- Sync window monitoring: Enabled to monitor the synchronization of the slaves.
- Sync window: Time for Sync window monitoring.

④ Options

- Use LRW instead of LWR/LRD: Use grouped (LRW) to read/write PDOs, instead of reading (LRD) or writing (LWR) PDOs separately. Default: Reading PDOs or writing PDOs separately.
- Automatic restart slaves: Once started, if the EtherCAT communication of the Master is disconnected, the slave will immediately restart.

- **Log**

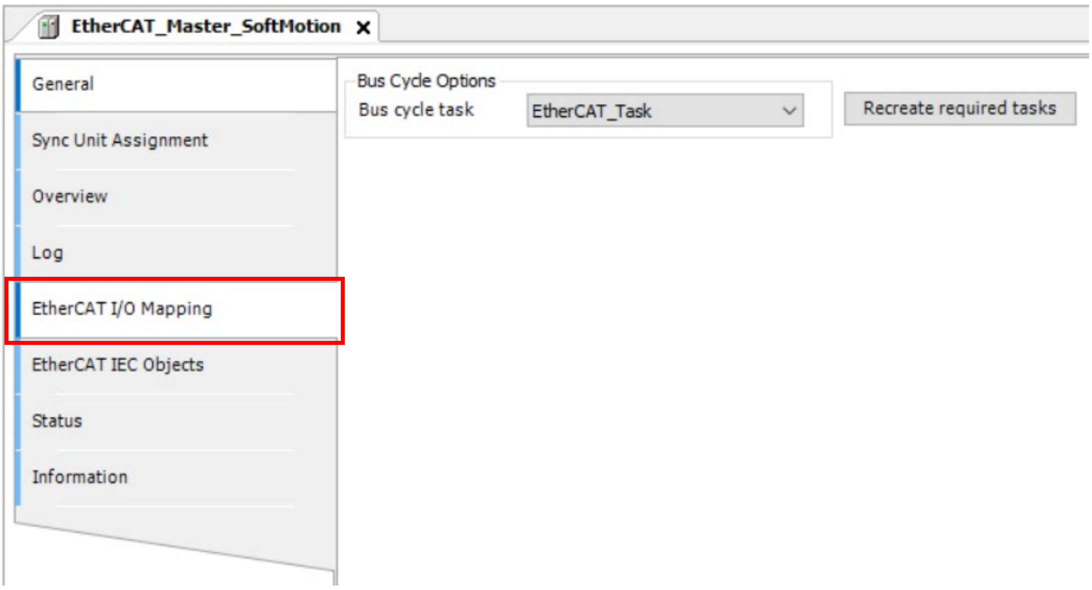
Here you can view the PLC log. It lists the events that were recorded on the target system. Refer to section 4.2.1.5 Log for more information.



- **EtherCAT I/O Mapping**

■ **Bus cycle task:** Select EtherCAT_Task for Bus cycle task. If selecting other task, it is possible to cause EtherCAT communication error. Default: EtherCAT_Task.

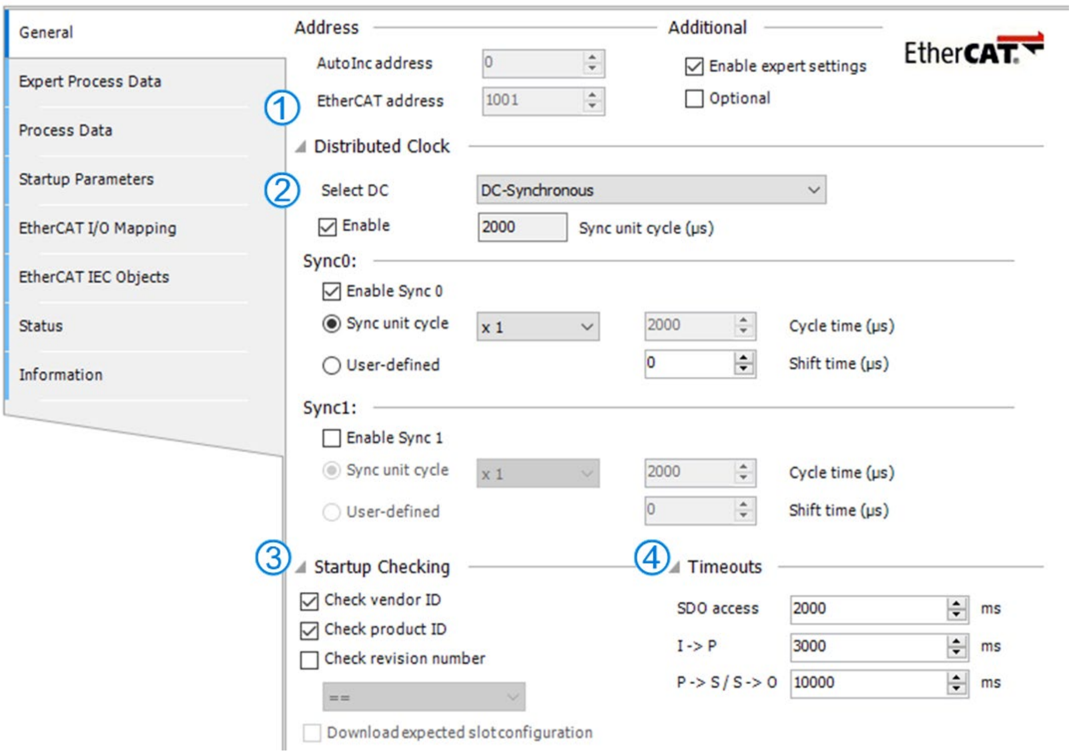
Note: When the option “Use parent bus cycle setting” is selected, the system uses the bus cycle setting of the PLC Setting as the EtherCAT bus cycle time.



Bus cycle task: EtherCAT_Task

8.1.3 Setting up the EtherCAT Slave

This section introduces functions in the tab of Slaves.



Address

AutoInc address: 0

EtherCAT address: 1001

Additional

☑ Enable expert settings

☐ Optional

Distributed Clock

Select DC: DC-Synchronous

☑ Enable 2000 Sync unit cycle (µs)

Sync0:

☑ Enable Sync 0

● Sync unit cycle x 1 2000 Cycle time (µs)

○ User-defined 0 Shift time (µs)

Sync1:

☐ Enable Sync 1

○ Sync unit cycle x 1 2000 Cycle time (µs)

○ User-defined 0 Shift time (µs)

Startup Checking

☑ Check vendor ID

☑ Check product ID

☐ Check revision number

==

☐ Download expected slot configuration

Timeouts

SDO access 2000 ms

I -> P 3000 ms

P -> S / S -> O 10000 ms

- **General**
  - **Address**

    ① **EtherCAT address**: Final address of the slaves, assigned by the master during bootup.

  - **Distributed Clocks**

    ② **Select DC**: Cycle time for the data exchange.

  - ③ **Startup Checking**

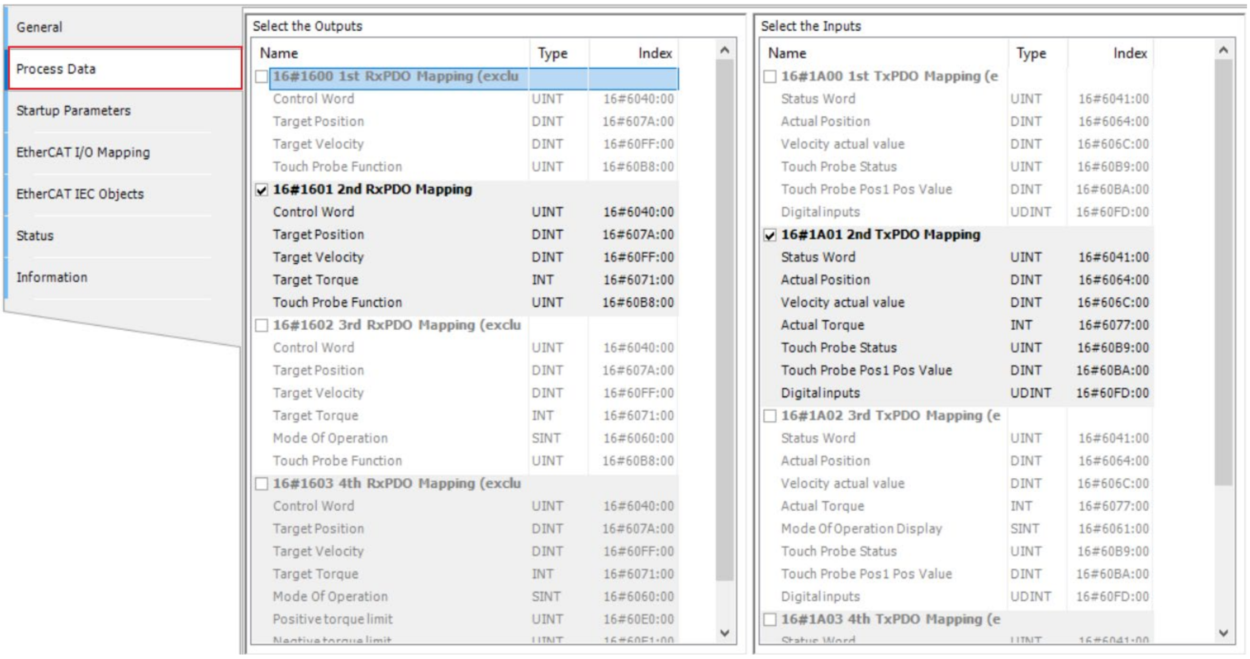
Function	Description
Check vendor ID	Once the system starts, it checks if the vendor ID and product ID are the same as configured. If not, the system stops without any further operation.
Check product ID	
Check revision number	Once the system starts, it checks if the revision number is the same as the drop-down list showed.

  - ④ **Timeouts**

Function	Description
SDO access	Once the system starts, the SDO also starts transmitting. Unit: ms
I -> P	Switching form Init mode to Pre operational mode. Unit: ms
P -> S / S -> O	Switching from Pre operational mode to Safe Operational mode. Or switching from Safe-Op mode to Operational mode. Unit: ms

- **Process Data**

The data mapping of the EtherCAT network is a cyclic data exchange between the master and slave tthrough the CoE-based PDO mapping. The data that a slave sends to the master are packed in TxPDO and the data that the slave reads from the master are packed in RxPDO. The inputs and outputs on the pages of Select the Outputs and Select the Inputs contain the lists of PDOs which are available for data exchange and can be edited. For ESI file of a device, the PDOs and PDO contents for option have been defined and some PDO contents are allowed to be edited by users themselves as defined in ESI.



Select the Outputs

Name	Type	Index
☐ 16#1600 1st RxPDO Mapping (exclu		
Control Word	UINT	16#6040:00
Target Position	DINT	16#607A:00
Target Velocity	DINT	16#60FF:00
Touch Probe Function	UINT	16#60B8:00
☑ 16#1601 2nd RxPDO Mapping		
Control Word	UINT	16#6040:00
Target Position	DINT	16#607A:00
Target Velocity	DINT	16#60FF:00
Target Torque	INT	16#6071:00
Touch Probe Function	UINT	16#60B8:00
☐ 16#1602 3rd RxPDO Mapping (exclu		
Control Word	UINT	16#6040:00
Target Position	DINT	16#607A:00
Target Velocity	DINT	16#60FF:00
Target Torque	INT	16#6071:00
Mode Of Operation	SINT	16#6060:00
Touch Probe Function	UINT	16#60B8:00
☐ 16#1603 4th RxPDO Mapping (exclu		
Control Word	UINT	16#6040:00
Target Position	DINT	16#607A:00
Target Velocity	DINT	16#60FF:00
Target Torque	INT	16#6071:00
Mode Of Operation	SINT	16#6060:00
Positive torque limit	UINT	16#60E0:00

Select the Inputs

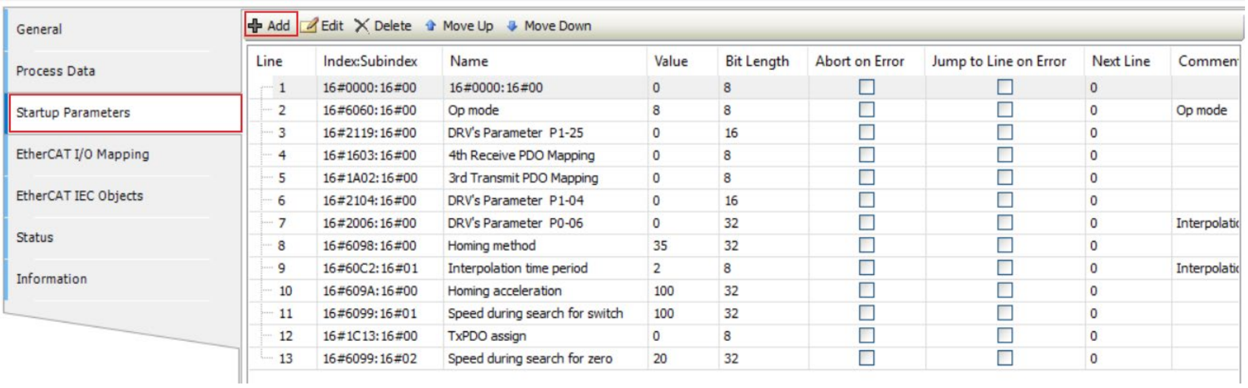
Name	Type	Index
☐ 16#1A00 1st TxPDO Mapping (e		
Status Word	UINT	16#6041:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Touch Probe Status	UINT	16#60B9:00
Touch Probe Pos1 Pos Value	DINT	16#60BA:00
Digital inputs	UDINT	16#60FD:00
☑ 16#1A01 2nd TxPDO Mapping		
Status Word	UINT	16#6041:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Actual Torque	INT	16#6077:00
Touch Probe Status	UINT	16#60B9:00
Touch Probe Pos1 Pos Value	DINT	16#60BA:00
Digital inputs	UDINT	16#60FD:00
☐ 16#1A02 3rd TxPDO Mapping (e		
Status Word	UINT	16#6041:00
Actual Position	DINT	16#6064:00
Velocity actual value	DINT	16#606C:00
Actual Torque	INT	16#6077:00
Mode Of Operation Display	SINT	16#6061:00
Touch Probe Status	UINT	16#60B9:00
Touch Probe Pos1 Pos Value	DINT	16#60BA:00
Digital inputs	UDINT	16#60FD:00
☐ 16#1A03 4th TxPDO Mapping (e		

If outputs of the device are activated here (for writing), these outputs can be assigned to project variables in the EtherCAT I/O Mapping window. And if inputs of the device are activated here (for reading), these inputs can be assigned to project variables in the EtherCAT I/O Mapping window. It takes more PLC system resources if you use more PDOs.

- **Startup Parameters**

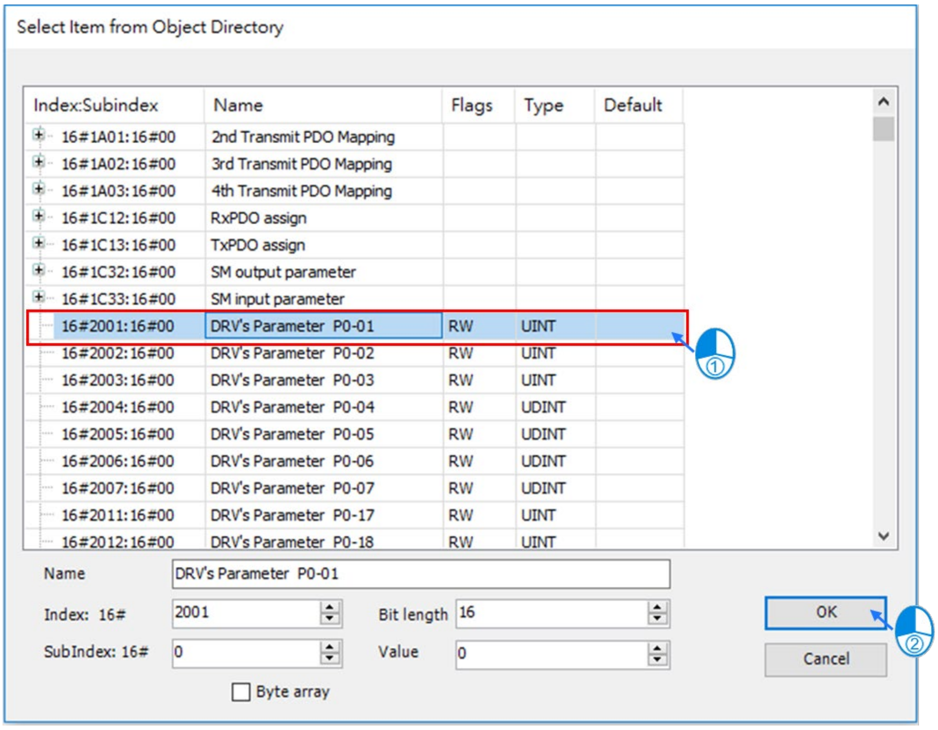
After EtherCAT communication is established, the master will download all the parameters from the table of the slave. The table shows the instructions which are defined in the ESI file by default and you can add or edit the instructions in the table.

Function Button	Description
Add	By specifying new index/subindex 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.
Edit	In this window you can change the parameters of the SDO before the SDO is added to the configuration.
Move Up	Moves the selected line upwards by one line
Move Down	Moves the selected line downwards by one line



Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Error	Next Line	Commen
1	16#0000:16#00	16#0000:16#00	0	8	☐	☐	0	
2	16#6060:16#00	Op mode	8	8	☐	☐	0	Op mode
3	16#2119:16#00	DRV's Parameter P1-25	0	16	☐	☐	0	
4	16#1603:16#00	4th Receive PDO Mapping	0	8	☐	☐	0	
5	16#1A02:16#00	3rd Transmit PDO Mapping	0	8	☐	☐	0	
6	16#2104:16#00	DRV's Parameter P1-04	0	16	☐	☐	0	
7	16#2006:16#00	DRV's Parameter P0-06	0	32	☐	☐	0	Interpolatio
8	16#6098:16#00	Homing method	35	32	☐	☐	0	
9	16#60C2:16#01	Interpolation time period	2	8	☐	☐	0	Interpolatio
10	16#609A:16#00	Homing acceleration	100	32	☐	☐	0	
11	16#6099:16#01	Speed during search for switch	100	32	☐	☐	0	
12	16#1C13:16#00	TxPDO assign	0	8	☐	☐	0	
13	16#6099:16#02	Speed during search for zero	20	32	☐	☐	0	

Click **Add** button to open the **Select Item Object Directory** window. And select the parameter that you'd like to add and then click **OK** to add the item in.



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

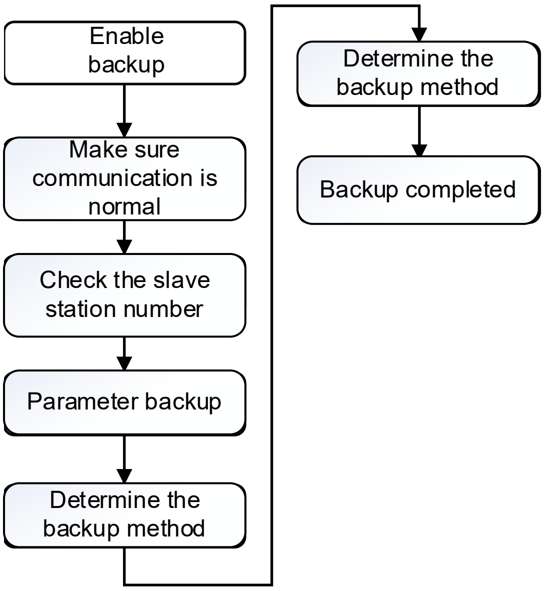
Name: DRV's Parameter P0-01
Index: 16# 2001
SubIndex: 16# 0
Bit length: 16
Value: 0
☐ Byte array

8.1.4 Backup Parameters for EtherCAT Slaves

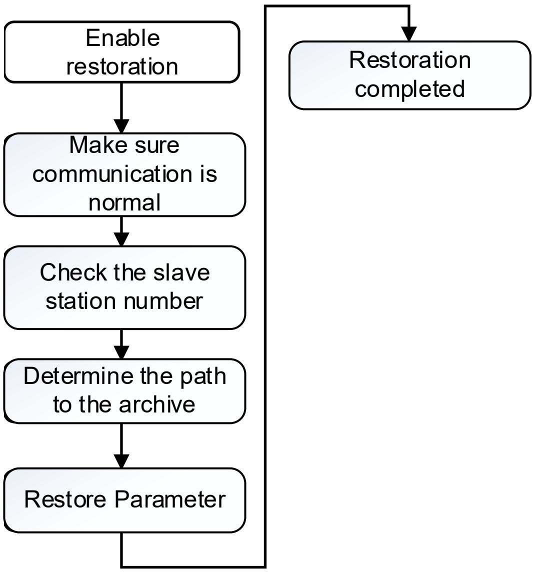
While using EtherCAT communication, we provide custom parameter storage feature for ASDA series servo drives with the backup feature to backup and recover parameters of all slave stations.

8.1.4.1 Data Backup Procedure

- Backup procedure



- Restore procedure

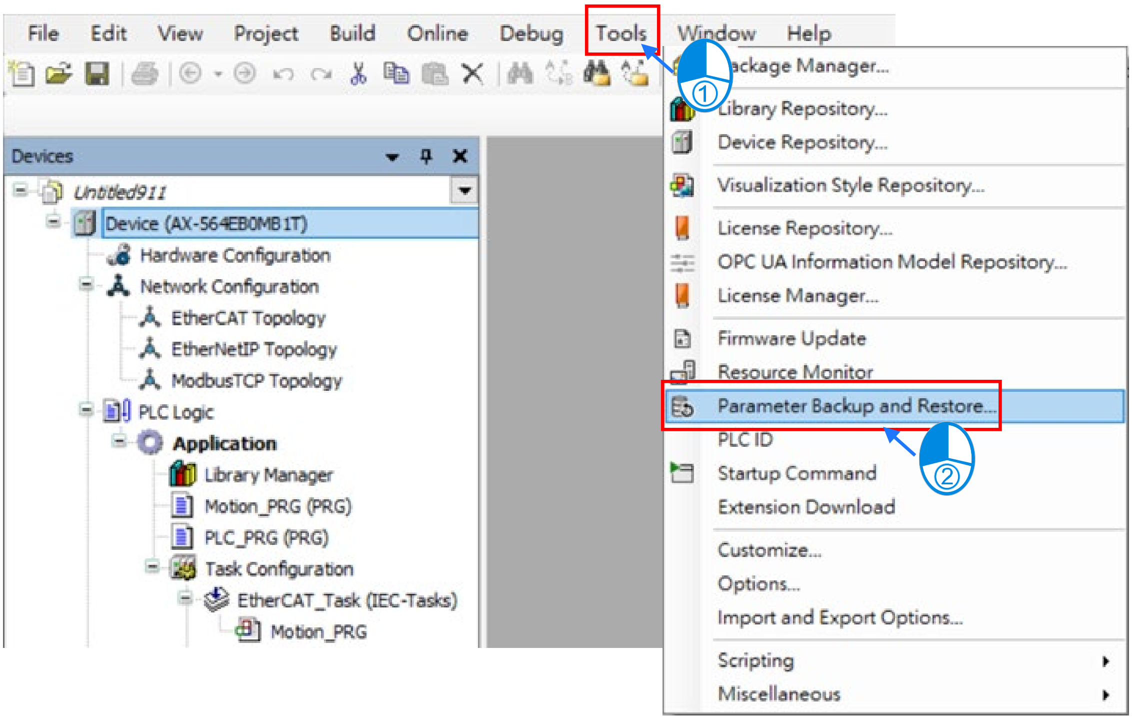


***Note:** If there's any error occurs in the axis error during the execution of parameter backup or restoration, the slave will skipped the problematic axis and keep the procedures of backup and restoration for the others. After the backup / restoration of the rest of slave stations are completed, the related messages of problematic axis will be displayed.

8.1.4.2 Introduction to Backup and Restore

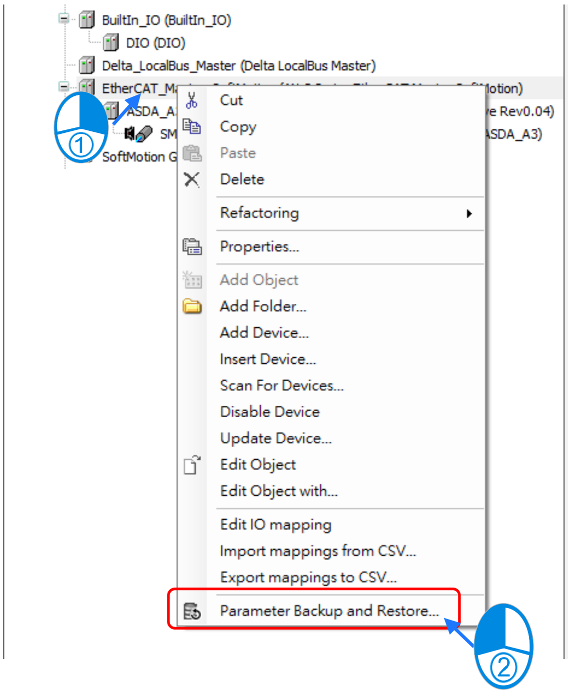
- Supported version for backup and restoration.
  - DIADesigner-AX version: V1.2 and above
  - Only models ASDA-A3-E and ASDA-B3-E are supported for parameter backup and restoration.
    - ASDA-A3-E firmware version: V11165 sub 92 and above
    - ASDA-B3-E firmware version: V10665 sub 75 and above
- Data to be back up from a slave servo.

  Servo parameters P0 to P4 (except P0.001 and P4.000), P5.0003, P5.0008 to P5.0009, P5.0020 to P5.0030 and P6.0000 to P6.0001.
- Two ways to open the parameter backup and restore page.
  - Go to Tools and double-click on the option Parameter Backup and Restore.

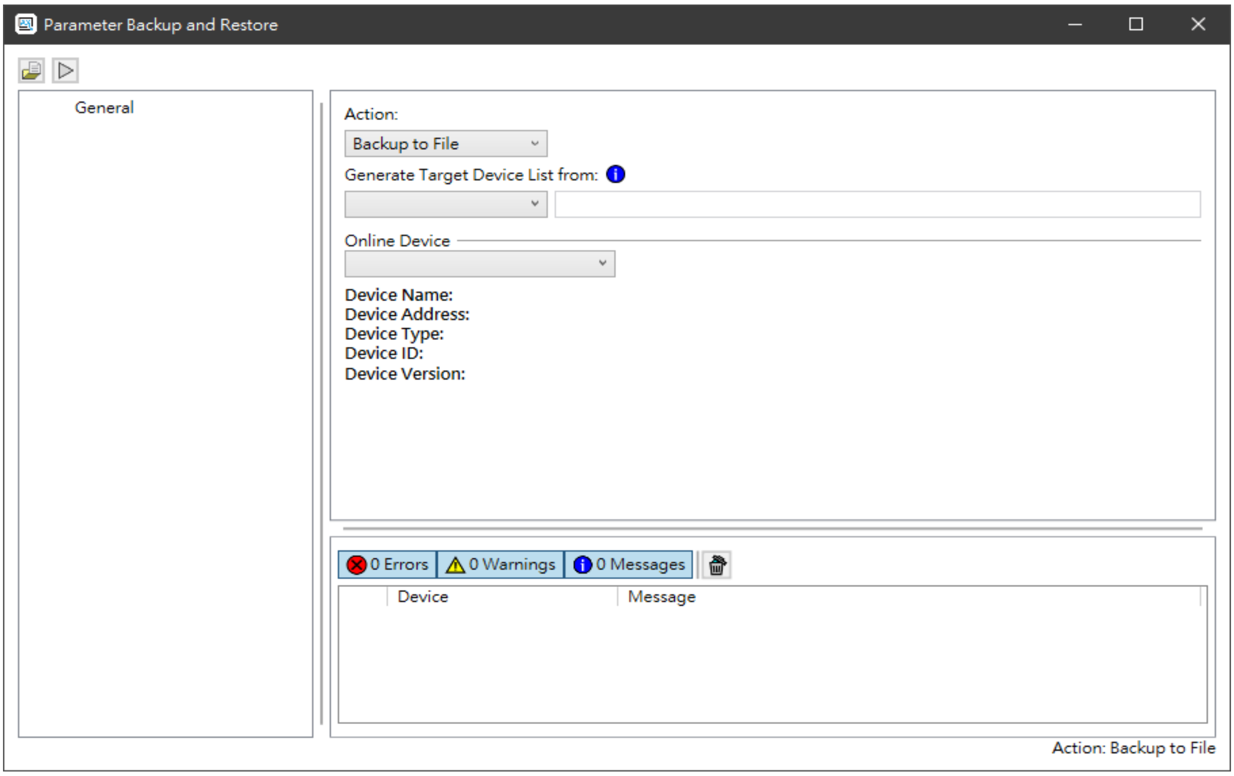


***Note:** When using this method, the option “Current Project” will not be available from “Generate Target Device List from”.

- ■ Right-click the option EtherCAT_Master_SoftMotion (AX Series EtherCAT Master SoftMotion).



- ● Parameter backup and restore page.



Action:
Backup to File

Generate Target Device List from:

Online Device

Device Name:
Device Address:
Device Type:
Device ID:
Device Version:

Action: Backup to File

Name	Function
Online Device	Select the target device to connect.
Generate Target Device List from	Select EtherCAT project tree - Archive File → EtherCAT topology file - Current Project → EtherCATtopology in the current project - Online Topology → Online EtherCAT topology
Action	Select the target action - Backup to File → Backup parameters to files. - Backup to SD Card → Backup parameters to external SD cards. - Restore from File → Restore parameters from files. - Restore from SD Card → Restore parameters from SD cards.
📂	Save the current EtherCAT topology (Archive File)
▷	Execute the backup/ restore feature.

- External SD card backup path

External SD card path: /PLC CARD/AX_/SysDup/ECAT/BackupRestore/ (The "_" in the path represents model types. For example, model AX-5 would be AX5 here.)

8.1.4.2.1 Operation for the Backup Function

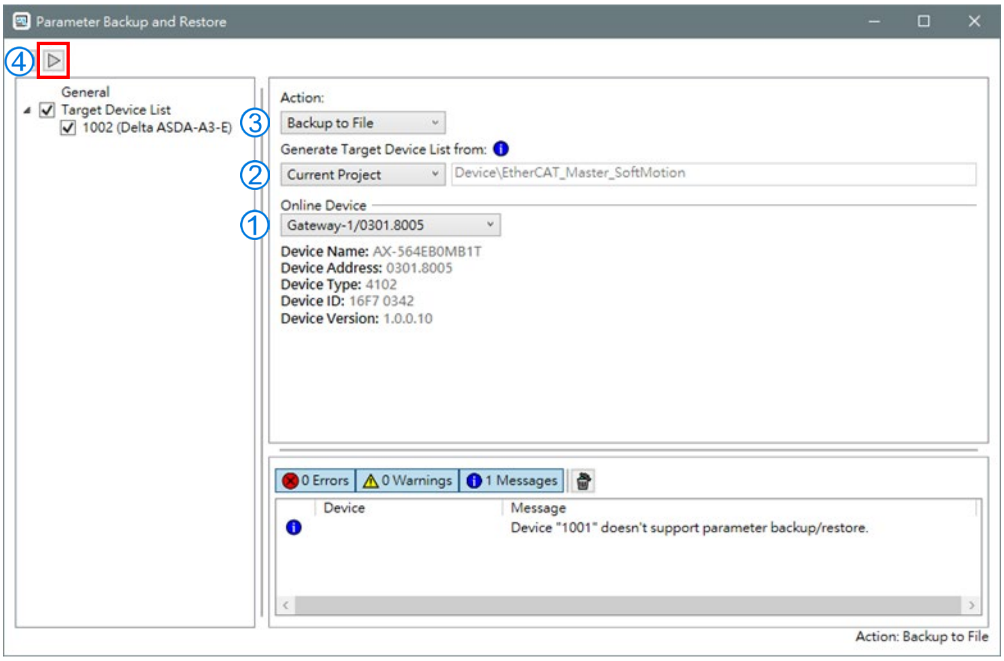
- Parameter backup

  ① Select Device.

  ② Configure EtherCAT topology for the current project.

  ③ Set Action to "Backup to File".

  ④ Execute backup.



General
Target Device List
☑ 1002 (Delta ASDA-A3-E)

Action: Backup to File

Generate Target Device List from: Current Project Device\EtherCAT_Master_SoftMotion

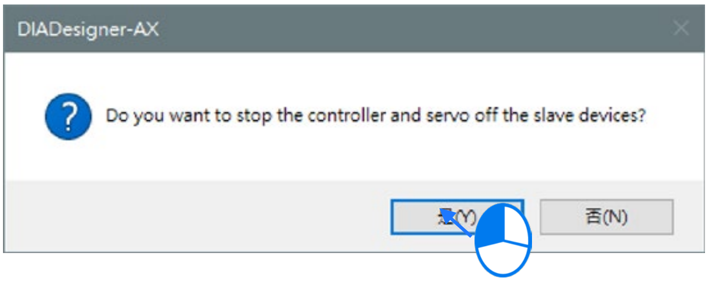
Online Device: Gateway-1/0301.8005

Device Name: AX-564EB0MB1T
Device Address: 0301.8005
Device Type: 4102
Device ID: 16F7 0342
Device Version: 1.0.0.10

Device	Message
	Device "1001" doesn't support parameter backup/restore.

Action: Backup to File

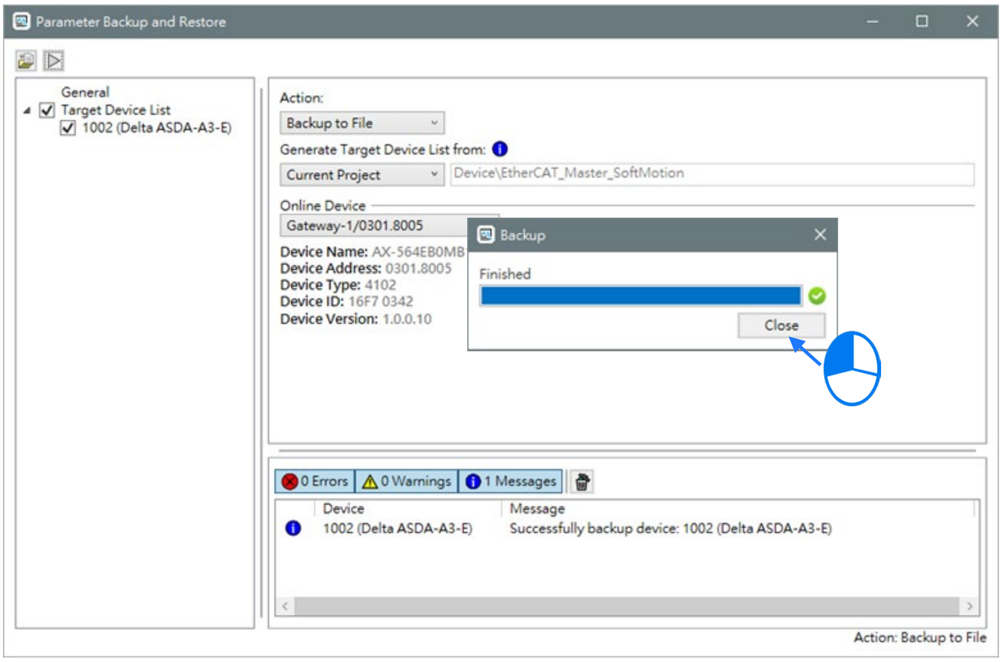
- Change the PLC state to Stop.



DIADesigner-AX

Do you want to stop the controller and servo off the slave devices?

- Click “Close” after the parameter backup is complete.



Action: Backup to File

Generate Target Device List from: Current Project Device\EtherCAT_Master_SoftMotion

Online Device: Gateway-1/0301.8005

Device Name: AX-564EB0MB
Device Address: 0301.8005
Device Type: 4102
Device ID: 16F7 0342
Device Version: 1.0.0.10

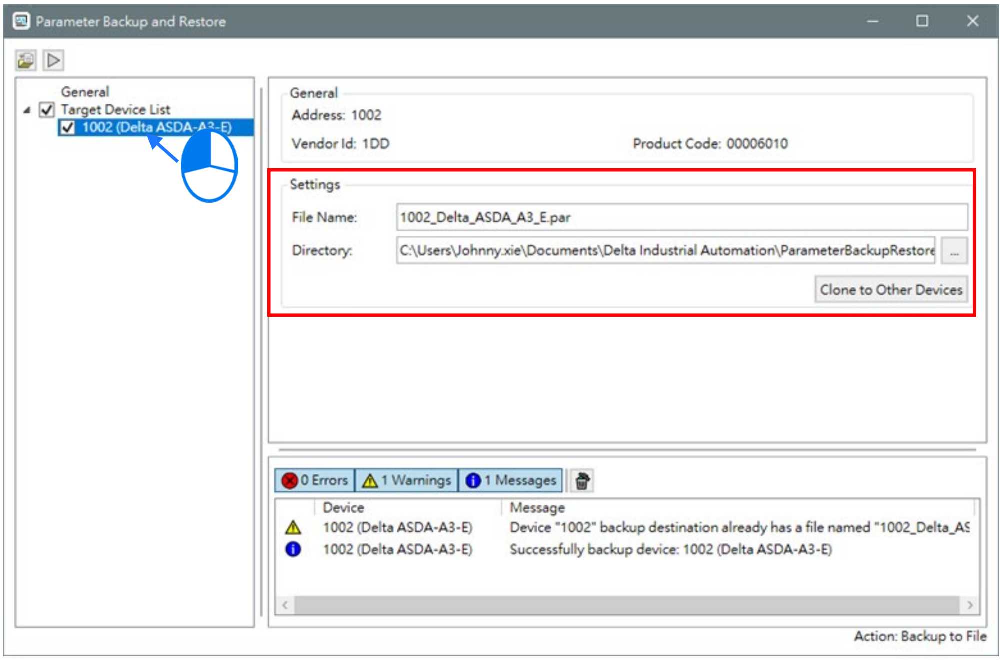
Backup: Finished

0 Errors | 0 Warnings | 1 Messages

Device	Message
1002 (Delta ASDA-A3-E)	Successfully backup device: 1002 (Delta ASDA-A3-E)

Action: Backup to File

- Backup directory



General
Address: 1002
Vendor Id: 1DD
Product Code: 00006010

Settings
File Name: 1002_Delta_ASDA_A3_E.par
Directory: C:\Users\Johnny.xie\Documents\Delta Industrial Automation\ParameterBackupRestore

0 Errors | 1 Warnings | 1 Messages

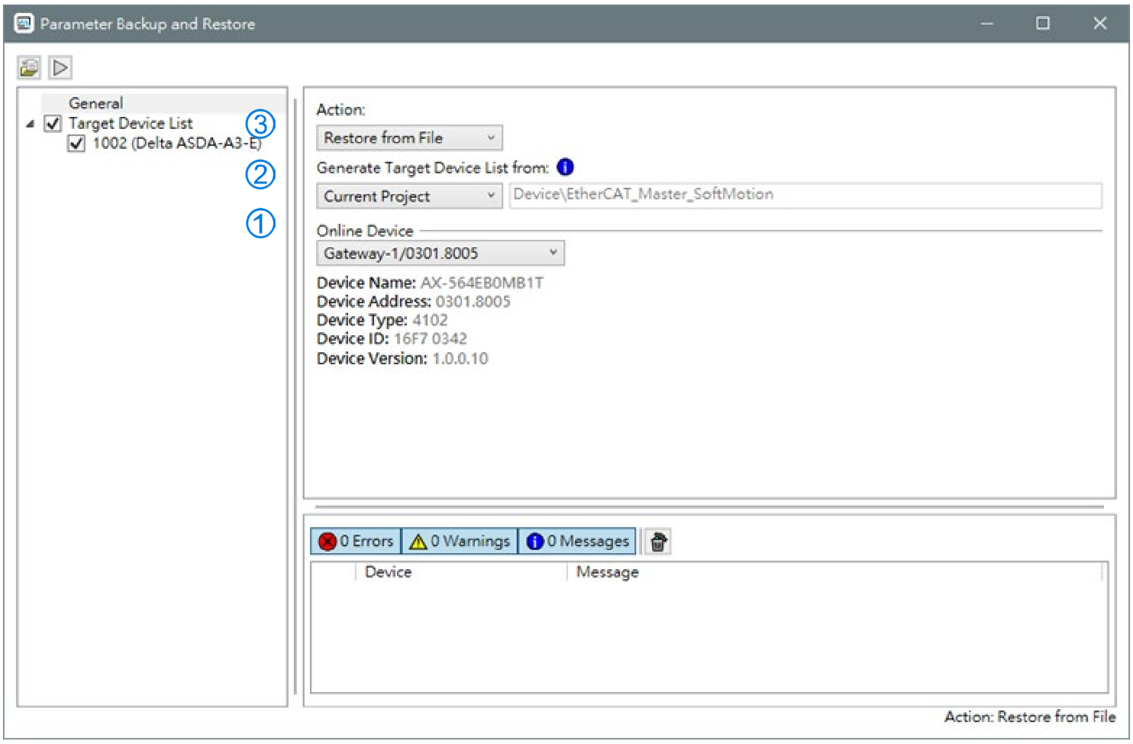
Device	Message
1002 (Delta ASDA-A3-E)	Device "1002" backup destination already has a file named "1002_Delta_AS
1002 (Delta ASDA-A3-E)	Successfully backup device: 1002 (Delta ASDA-A3-E)

Action: Backup to File

Name	Function
File Name	Set the name for parameter backup file.
Directory	Set the backup directory.
Clone to Other Devices	Change all the backup directory of other devices.

8.1.4.2.2 Operation for the Restore Function

- Parameter restoration
  ① Select a Device.
  ② Configure EtherCAT topology for the current project.
  ③ Set an Action: “Backup to File”, “Backup to SD Card”, “Restore from File” or “Restore from SD Card”.



Action: Restore from File

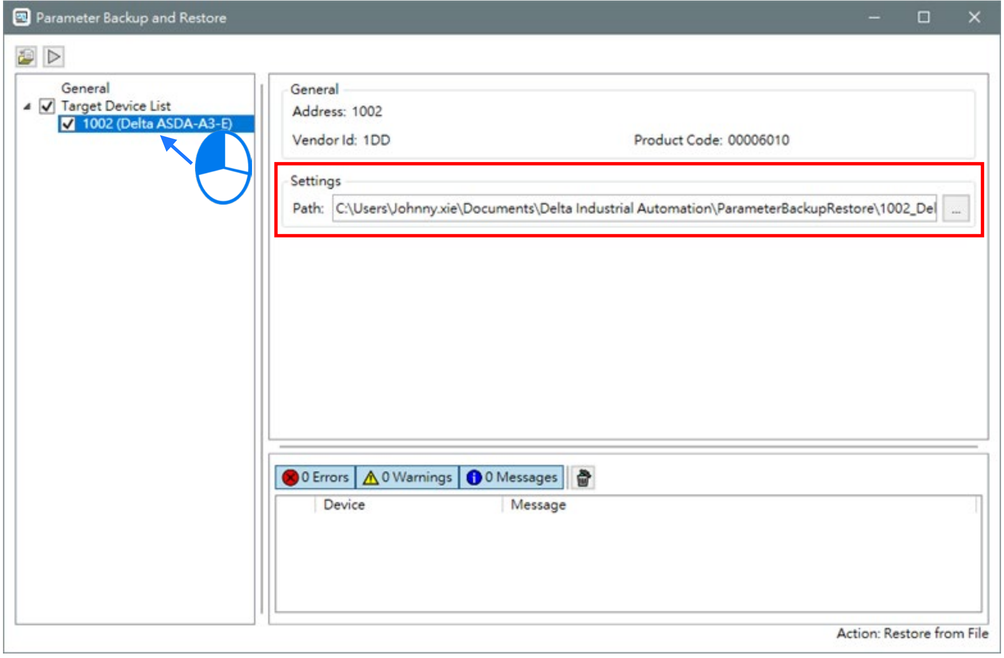
Generate Target Device List from: Current Project Device\EtherCAT_Master_SoftMotion

Online Device: Gateway-1/0301.8005

Device Name: AX-564EB0MB1T
Device Address: 0301.8005
Device Type: 4102
Device ID: 16F7 0342
Device Version: 1.0.0.10

Action: Restore from File

- Click on the target device and set the path to the file to restore.

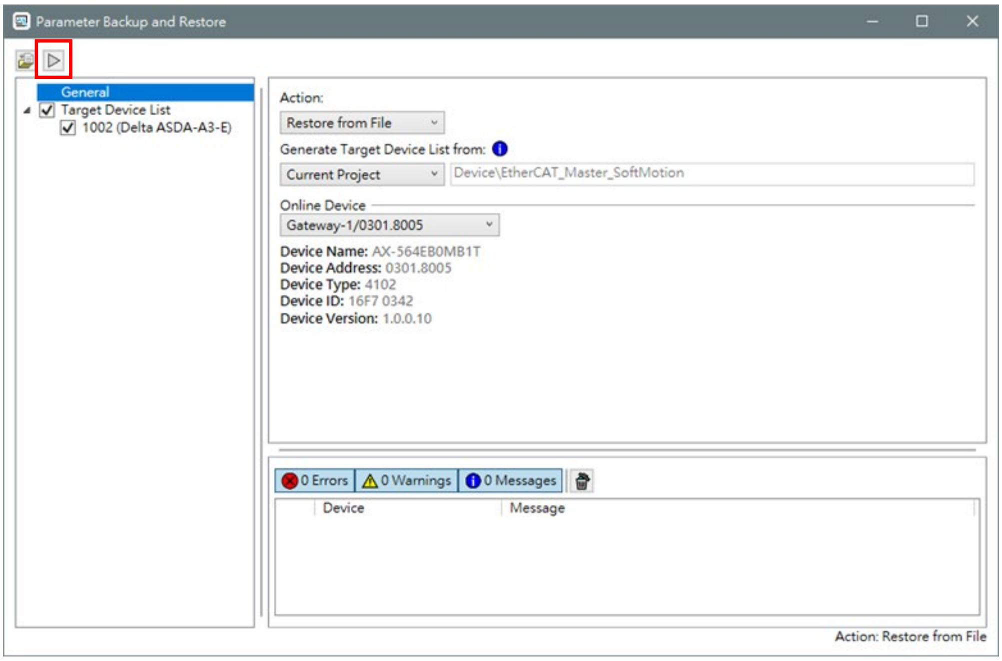


General
Address: 1002
Vendor Id: 1DD
Product Code: 00006010

Settings
Path: C:\Users\Johnny.xie\Documents\Delta Industrial Automation\ParameterBackupRestore\1002_Del

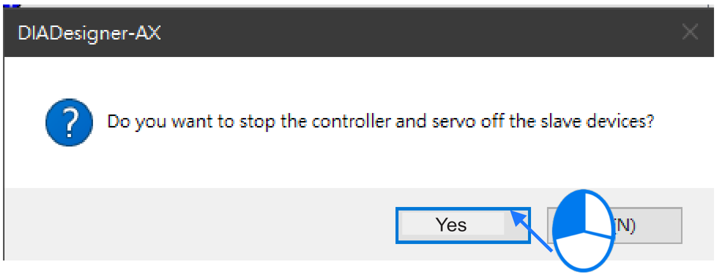
Action: Restore from File

- After you click Execute ▷ and click yes to stop the controller and servo for the moment.



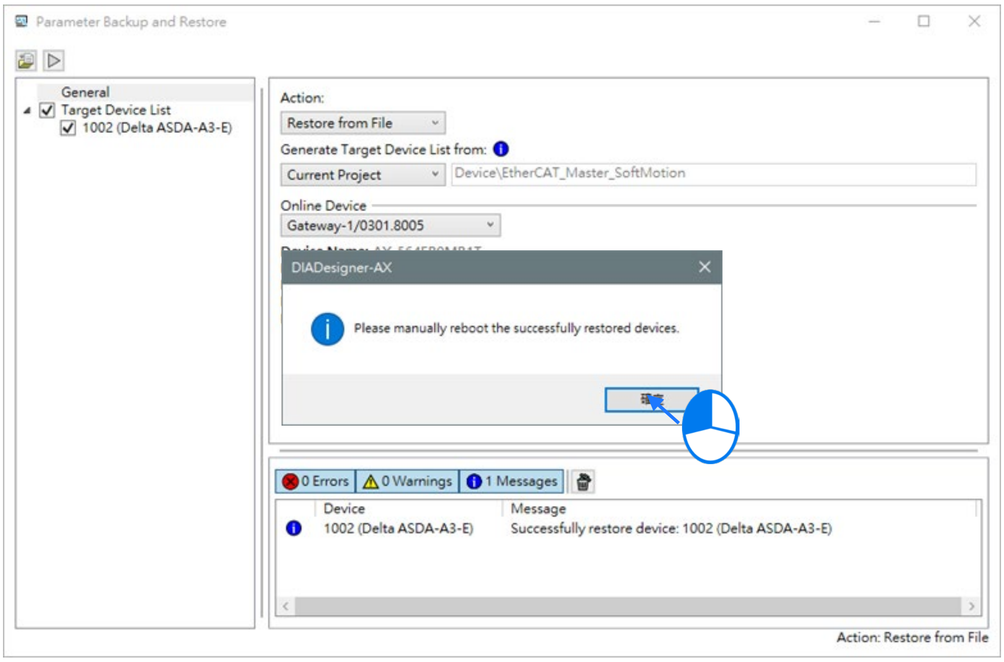
Action:
Restore from File
Generate Target Device List from:
Current Project
Device\EtherCAT_Master_SoftMotion
Online Device
Gateway-1/0301.8005
Device Name: AX-564EB0MB1T
Device Address: 0301.8005
Device Type: 4102
Device ID: 16F7 0342
Device Version: 1.0.0.10

Action: Restore from File



Do you want to stop the controller and servo off the slave devices?

- After the backup / restoration is completed, click close as the image shown below.



Action: Restore from File

Generate Target Device List from: Current Project Device\EtherCAT_Master_SoftMotion

Online Device: Gateway-1/0301.8005

Please manually reboot the successfully restored devices.

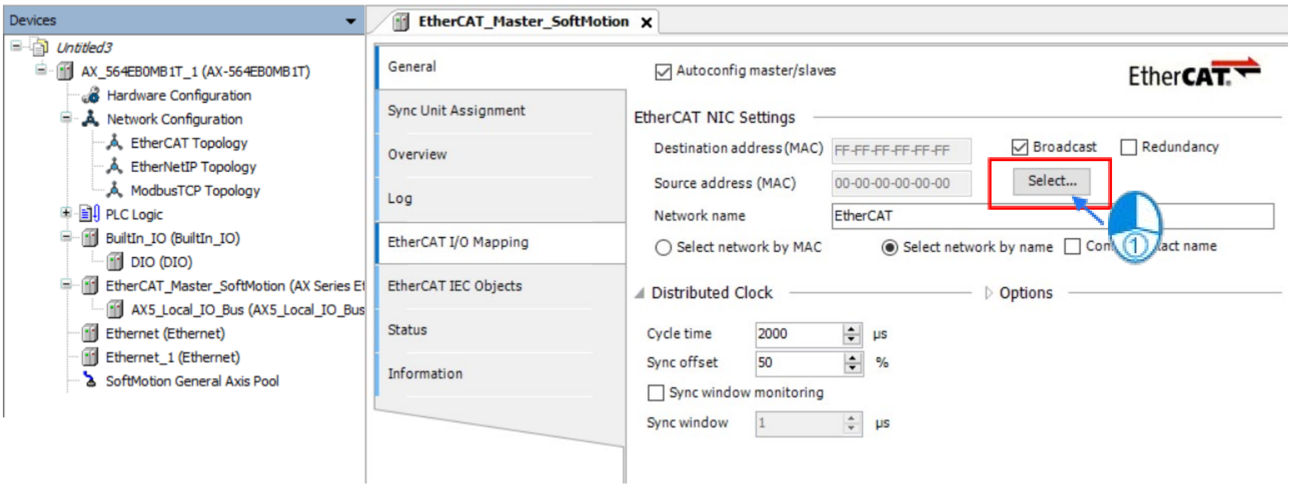
Device	Message
1002 (Delta ASDA-A3-E)	Successfully restore device: 1002 (Delta ASDA-A3-E)

Action: Restore from File

8.1.5 Operational Example of EtherCAT Master

This section uses coupler module AX-500CEC00-0A EtherCAT and DA module AX-504DA01-0A to demonstrate.

- Select EtherCAT as the communication port for AX-5 Series PLC CPU.



☑ Autoconfig master/slaves

EtherCAT NIC Settings

Destination address (MAC) FF-FF-FF-FF-FF-FF ☑ Broadcast ☐ Redundancy

Source address (MAC) 00-00-00-00-00-00

Network name EtherCAT

○ Select network by MAC ● Select network by name ☐ Com…act name

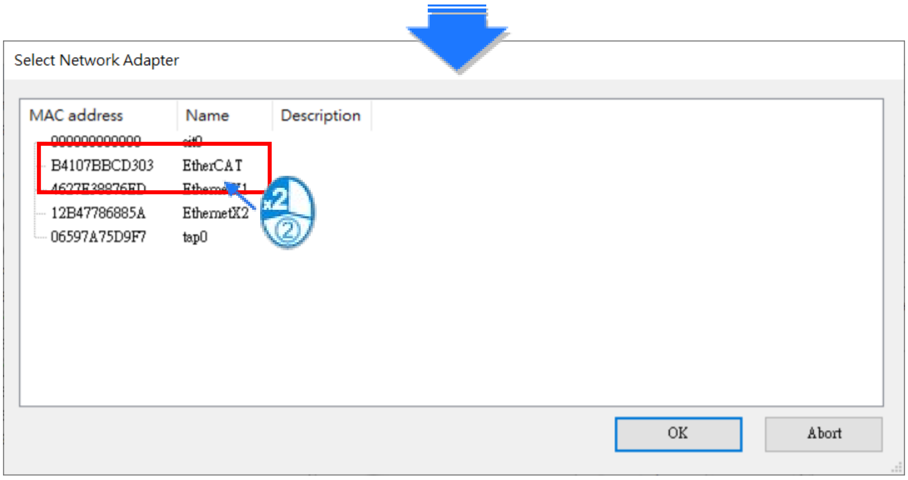
Distributed Clock

Cycle time 2000 µs

Sync offset 50 %

☐ Sync window monitoring

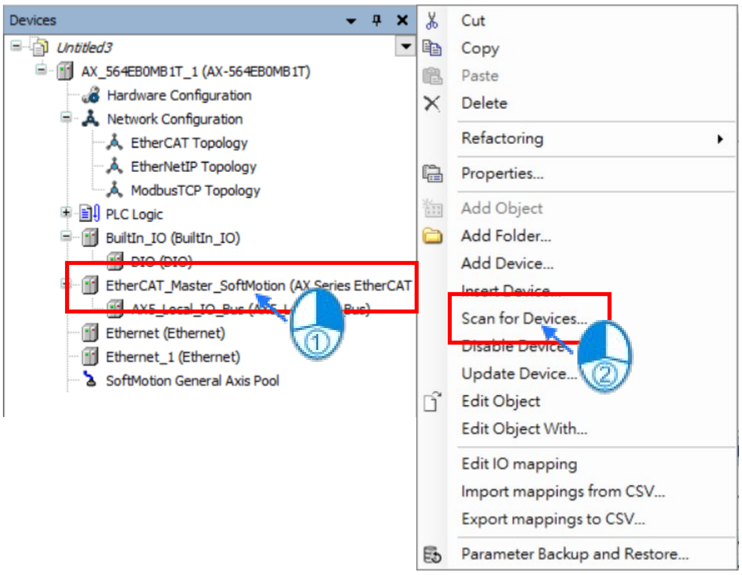
Sync window 1 µs

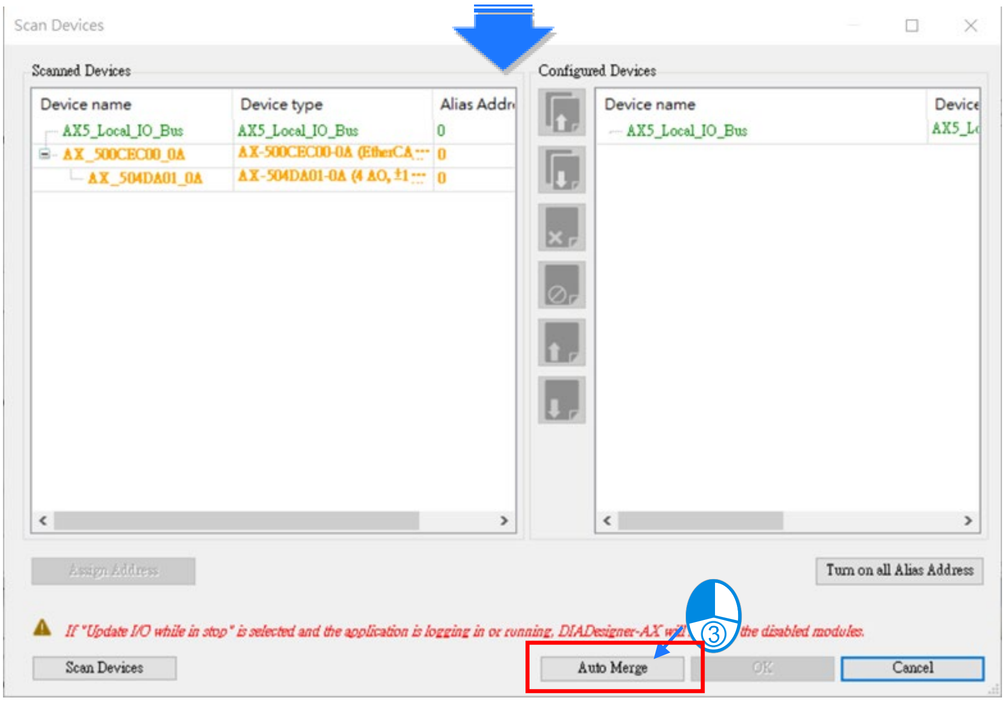


MAC address	Name	Description
000000000000	eth0	
B4107BBCD303	EtherCAT	
4627E38876ED	Ethernet1	
12B47786885A	EthernetX2	
06597A75D9F7	tap0	

- Add an EtherCAT Slave (the couple module AX-500CEC00-0A and DA module AX-504DA01-0A DA):

Adding an EtherCAT Slave by using the function *Scan for Devices* or *Add Device*. This example uses Scan for Devices to add an EtherCAT Slave.





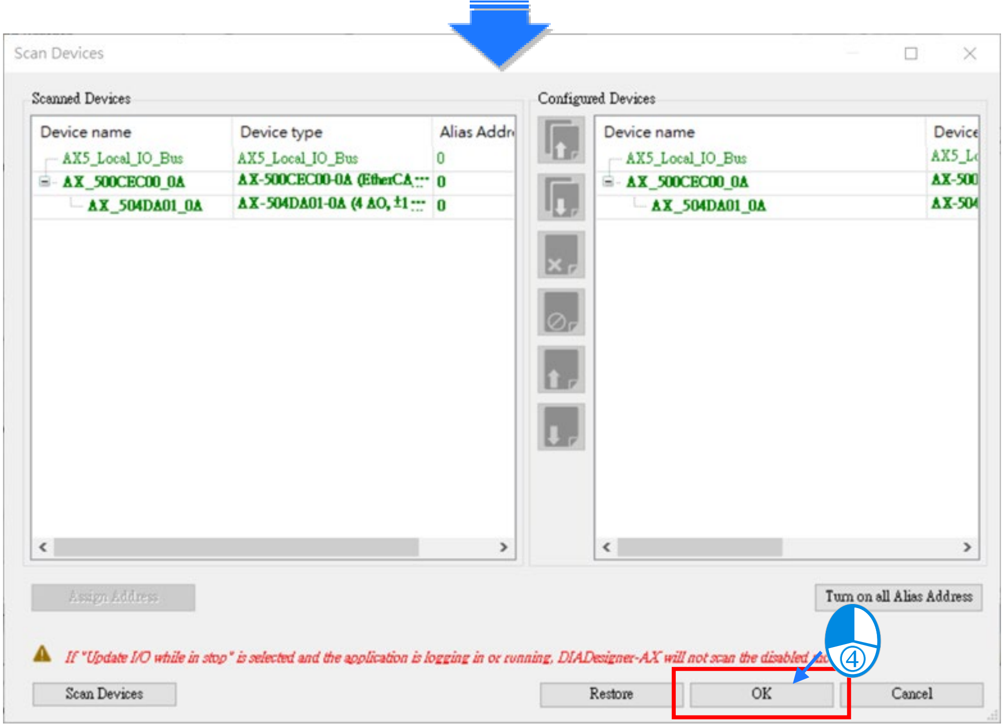
Scanned Devices

Device name	Device type	Alias Addr
AX5_Local_IO_Bus	AX5_Local_IO_Bus	0
AX_500CEC00_0A	AX-500CEC00-0A (EtherCA···	0
AX_504DA01_0A	AX-504DA01-0A (4 AO, ±1···	0

Configured Devices

Device name	Device
AX5_Local_IO_Bus	AX5_L

If "Update I/O while in stop" is selected and the application is logging in or running, DIADesigner-AX wi the disabled modules.



Scan Devices

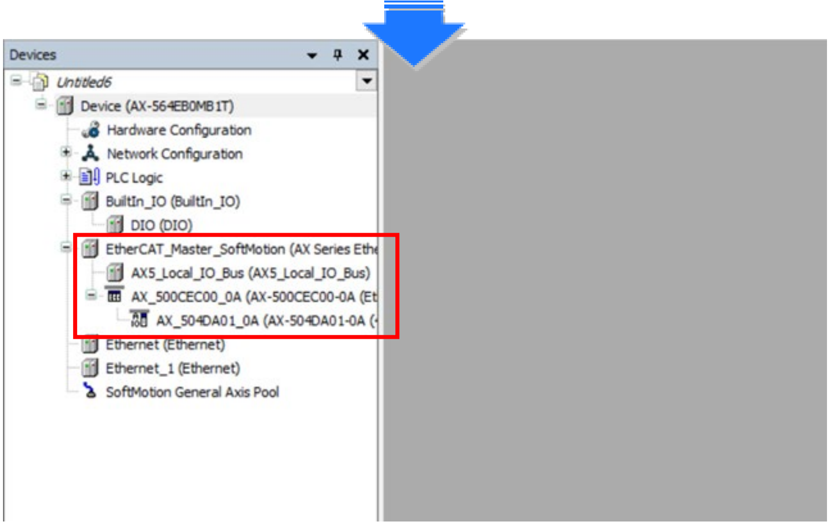
Scanned Devices

Device name	Device type	Alias Addr
AX5_Local_IO_Bus	AX5_Local_IO_Bus	0
AX_500CEC00_0A	AX-500CEC00-0A (EtherCA…	0
AX_504DA01_0A	AX-504DA01-0A (4 AO, ±1…	0

Configured Devices

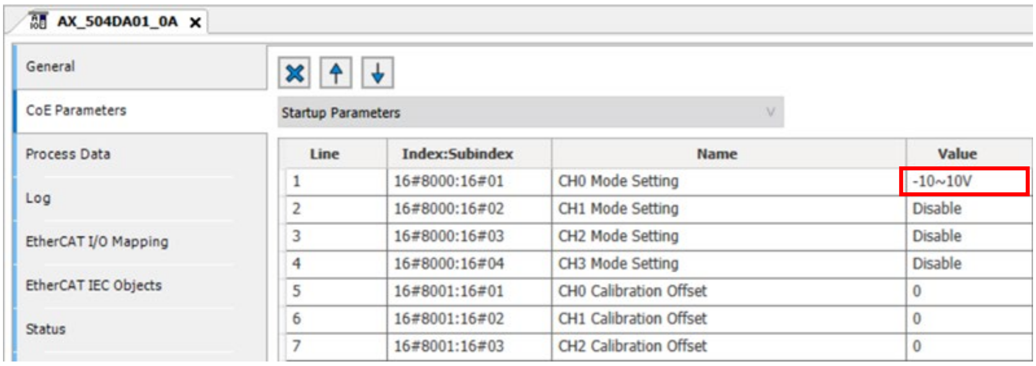
Device name	Device
AX5_Local_IO_Bus	AX5_Lo
AX_500CEC00_0A	AX-500
AX_504DA01_0A	AX-504

If "Update I/O while in stop" is selected and the application is logging in or running, DIADesigner-AX will not scan the disabled

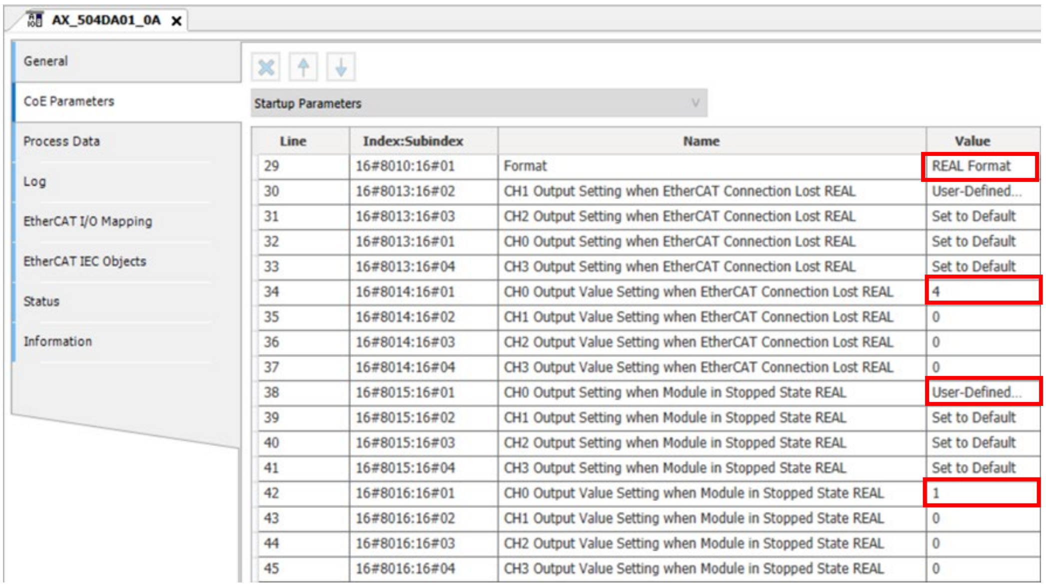


Devices
- Untitled6
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
    - BuiltIn_IO (BuiltIn_IO)
      - DIO (DIO)
    - EtherCAT_Master_SoftMotion (AX Series Ethe
      - AX5_Local_IO_Bus (AX5_Local_IO_Bus)
      - AX_500CEC00_0A (AX-500CEC00-0A (Et
        - AX_504DA01_0A (AX-504DA01-0A (
    - Ethernet (Ethernet)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

- **Set up the channel mode as below. (Format: REAL)**
  - Channel Mode Setting Value: -10 to 10 V
  - Format value: REAL
  - CH0 Output Value Setting when EtherCAT Connection Closed: 4 V
  - CH0 Output Setting when Module in Stopped State: 1 V

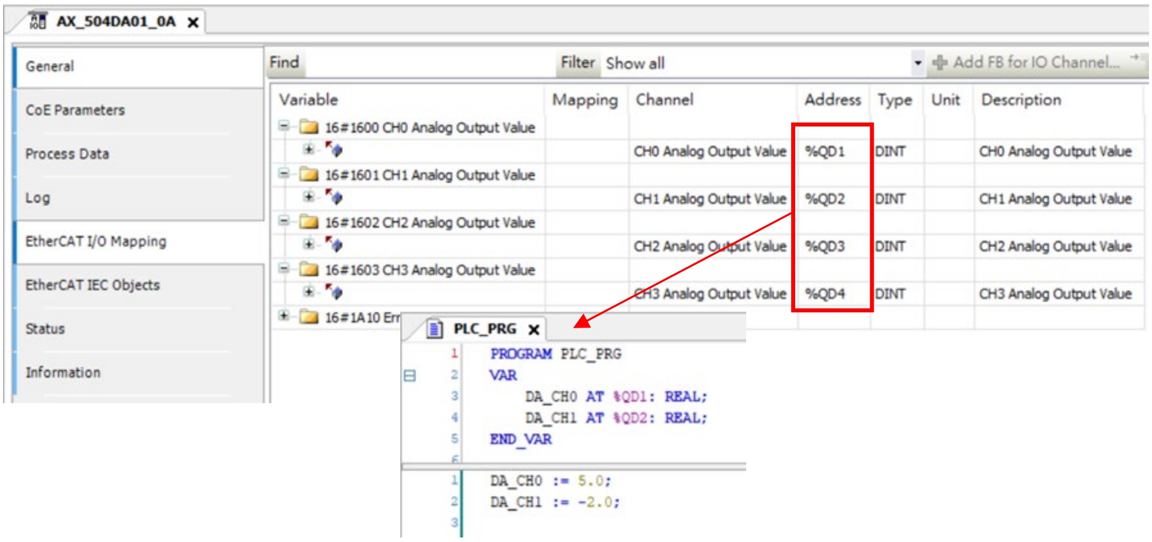


Line	Index:Subindex	Name	Value
1	16#8000:16#01	CH0 Mode Setting	-10~10V
2	16#8000:16#02	CH1 Mode Setting	Disable
3	16#8000:16#03	CH2 Mode Setting	Disable
4	16#8000:16#04	CH3 Mode Setting	Disable
5	16#8001:16#01	CH0 Calibration Offset	0
6	16#8001:16#02	CH1 Calibration Offset	0
7	16#8001:16#03	CH2 Calibration Offset	0



Line	Index:Subindex	Name	Value
29	16#8010:16#01	Format	REAL Format
30	16#8013:16#02	CH1 Output Setting when EtherCAT Connection Lost REAL	User-Defined...
31	16#8013:16#03	CH2 Output Setting when EtherCAT Connection Lost REAL	Set to Default
32	16#8013:16#01	CH0 Output Setting when EtherCAT Connection Lost REAL	Set to Default
33	16#8013:16#04	CH3 Output Setting when EtherCAT Connection Lost REAL	Set to Default
34	16#8014:16#01	CH0 Output Value Setting when EtherCAT Connection Lost REAL	4
35	16#8014:16#02	CH1 Output Value Setting when EtherCAT Connection Lost REAL	0
36	16#8014:16#03	CH2 Output Value Setting when EtherCAT Connection Lost REAL	0
37	16#8014:16#04	CH3 Output Value Setting when EtherCAT Connection Lost REAL	0
38	16#8015:16#01	CH0 Output Setting when Module in Stopped State REAL	User-Defined...
39	16#8015:16#02	CH1 Output Setting when Module in Stopped State REAL	Set to Default
40	16#8015:16#03	CH2 Output Setting when Module in Stopped State REAL	Set to Default
41	16#8015:16#04	CH3 Output Setting when Module in Stopped State REAL	Set to Default
42	16#8016:16#01	CH0 Output Value Setting when Module in Stopped State REAL	1
43	16#8016:16#02	CH1 Output Value Setting when Module in Stopped State REAL	0
44	16#8016:16#03	CH2 Output Value Setting when Module in Stopped State REAL	0
45	16#8016:16#04	CH3 Output Value Setting when Module in Stopped State REAL	0

Before programming, it is important to make sure the memory locations are assigned to the channels and then declare the REAL type variables and assigned them to the memory locations.

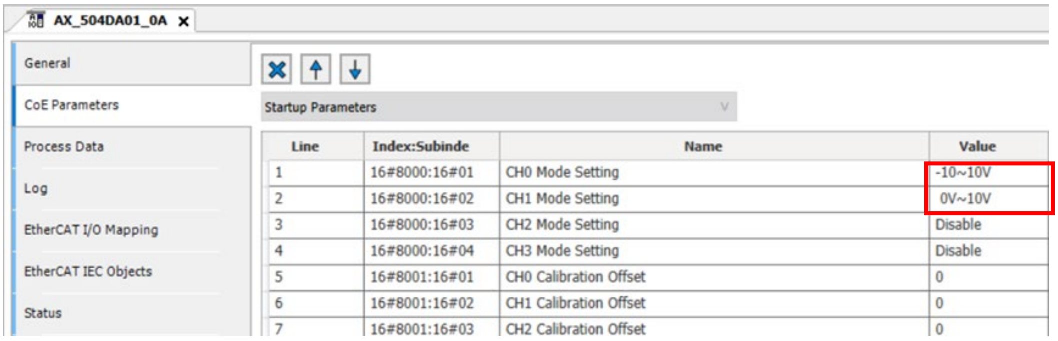


Variable	Mapping	Channel	Address	Type	Unit	Description
16#1600 CH0 Analog Output Value						
		CH0 Analog Output Value	%QD1	DINT		CH0 Analog Output Value
16#1601 CH1 Analog Output Value						
		CH1 Analog Output Value	%QD2	DINT		CH1 Analog Output Value
16#1602 CH2 Analog Output Value						
		CH2 Analog Output Value	%QD3	DINT		CH2 Analog Output Value
16#1603 CH3 Analog Output Value						
		CH3 Analog Output Value	%QD4	DINT		CH3 Analog Output Value
16#1A10 Err						

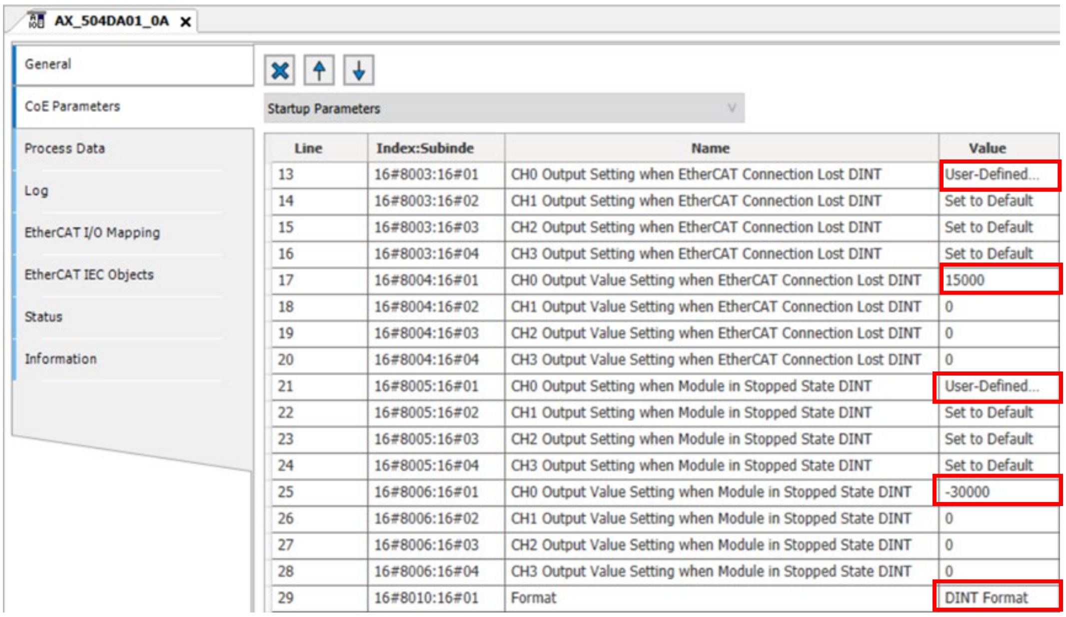
```
PROGRAM PLC_PRG
VAR
    DA_CH0 AT %QD1: REAL;
    DA_CH1 AT %QD2: REAL;
END_VAR
```

```
DA_CH0 := 5.0;
DA_CH1 := -2.0;
```

- **Set up the channel mode as below. (Format: DINT)**
  - Channel Mode Setting Value: -10 to 10 V (CH0), 0 to 10V (CH1)
  - Format value: DINT
  - CH0 Output Value Setting when EtherCAT Connection Closed: 5 V
  - CH0 Output Setting when Module in Stopped State: -10 V

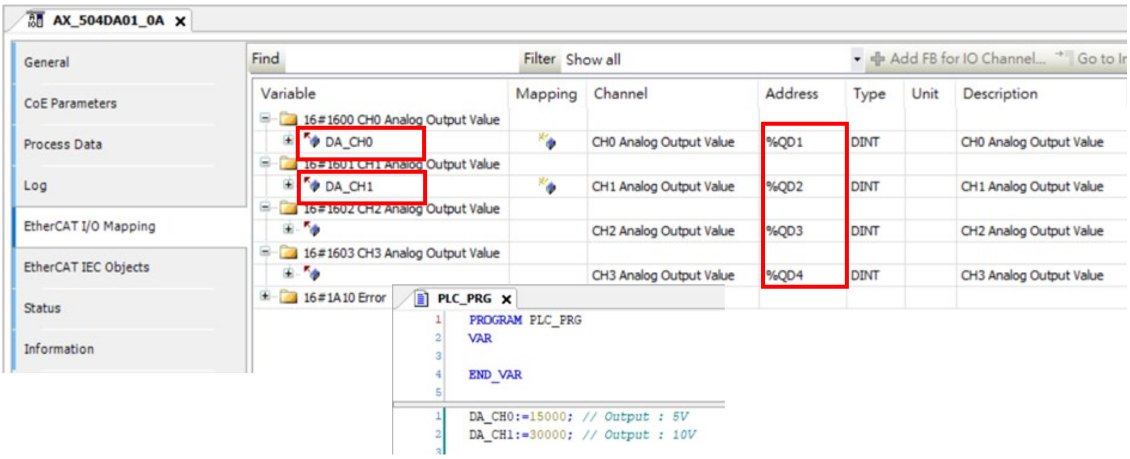


Line	Index:Subinde	Name	Value
1	16#8000:16#01	CH0 Mode Setting	-10~10V
2	16#8000:16#02	CH1 Mode Setting	0V~10V
3	16#8000:16#03	CH2 Mode Setting	Disable
4	16#8000:16#04	CH3 Mode Setting	Disable
5	16#8001:16#01	CH0 Calibration Offset	0
6	16#8001:16#02	CH1 Calibration Offset	0
7	16#8001:16#03	CH2 Calibration Offset	0



Line	Index:Subinde	Name	Value
13	16#8003:16#01	CH0 Output Setting when EtherCAT Connection Lost DINT	User-Defined...
14	16#8003:16#02	CH1 Output Setting when EtherCAT Connection Lost DINT	Set to Default
15	16#8003:16#03	CH2 Output Setting when EtherCAT Connection Lost DINT	Set to Default
16	16#8003:16#04	CH3 Output Setting when EtherCAT Connection Lost DINT	Set to Default
17	16#8004:16#01	CH0 Output Value Setting when EtherCAT Connection Lost DINT	15000
18	16#8004:16#02	CH1 Output Value Setting when EtherCAT Connection Lost DINT	0
19	16#8004:16#03	CH2 Output Value Setting when EtherCAT Connection Lost DINT	0
20	16#8004:16#04	CH3 Output Value Setting when EtherCAT Connection Lost DINT	0
21	16#8005:16#01	CH0 Output Setting when Module in Stopped State DINT	User-Defined...
22	16#8005:16#02	CH1 Output Setting when Module in Stopped State DINT	Set to Default
23	16#8005:16#03	CH2 Output Setting when Module in Stopped State DINT	Set to Default
24	16#8005:16#04	CH3 Output Setting when Module in Stopped State DINT	Set to Default
25	16#8006:16#01	CH0 Output Value Setting when Module in Stopped State DINT	-30000
26	16#8006:16#02	CH1 Output Value Setting when Module in Stopped State DINT	0
27	16#8006:16#03	CH2 Output Value Setting when Module in Stopped State DINT	0
28	16#8006:16#04	CH3 Output Value Setting when Module in Stopped State DINT	0
29	16#8010:16#01	Format	DINT Format

Before programming, it is important to make sure the memory locations are assigned to the channels and then declare the variables and assigned them to the memory locations. Refer to AX-5 Series Module Manual to check the setting range for each channel. -10 to 10 V: -30,000 to 30,000; 0 to 10 V: 0 to 30,000



Variable	Mapping	Channel	Address	Type	Unit	Description
16#1600 CH0 Analog Output Value						
DA_CH0		CH0 Analog Output Value	%QD1	DINT		CH0 Analog Output Value
16#1601 CH1 Analog Output Value						
DA_CH1		CH1 Analog Output Value	%QD2	DINT		CH1 Analog Output Value
16#1602 CH2 Analog Output Value						
		CH2 Analog Output Value	%QD3	DINT		CH2 Analog Output Value
16#1603 CH3 Analog Output Value						
		CH3 Analog Output Value	%QD4	DINT		CH3 Analog Output Value
16#1A10 Error						

```
PROGRAM PLC_PRG
VAR

END_VAR
```

```
DA_CH0:=15000; // Output : 5V
DA_CH1:=30000; // Output : 10V
```

8.2 Introduction on Modbus Serial Communication

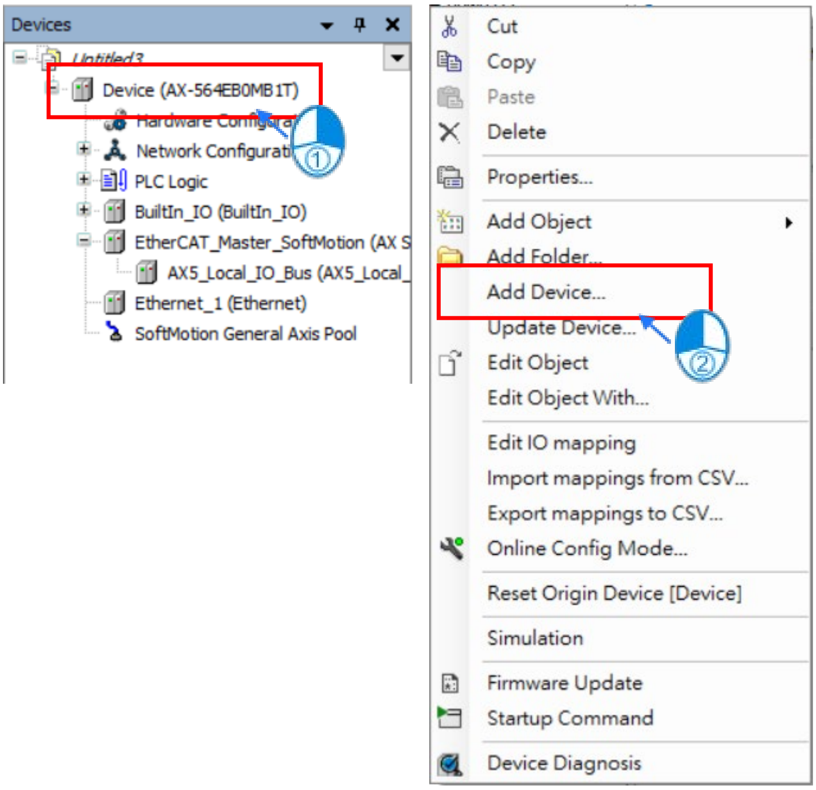
8.2.1 Modbus Serial Port

AX-5 Series PLC supports one RS-232 and one RS-485. Each Modbus Serial Port allows one master. A maximum of 32 Modbus Slave COM ports can be added to a Modbus Serial Master. But since RS-232 has no multipoint capability, only point-to-point connection is possible. And only the FIRST Modbus Slave COM Port can communicate with the Modbus Serial Master; the rest of the 31 Modbus Slave COM Port are not working. RS-485 has multipoint capability and therefore RS-485 does NOT have such limitations. Follow the below section to set up the basic settings for communication via the serial port for the Modbus serial port.

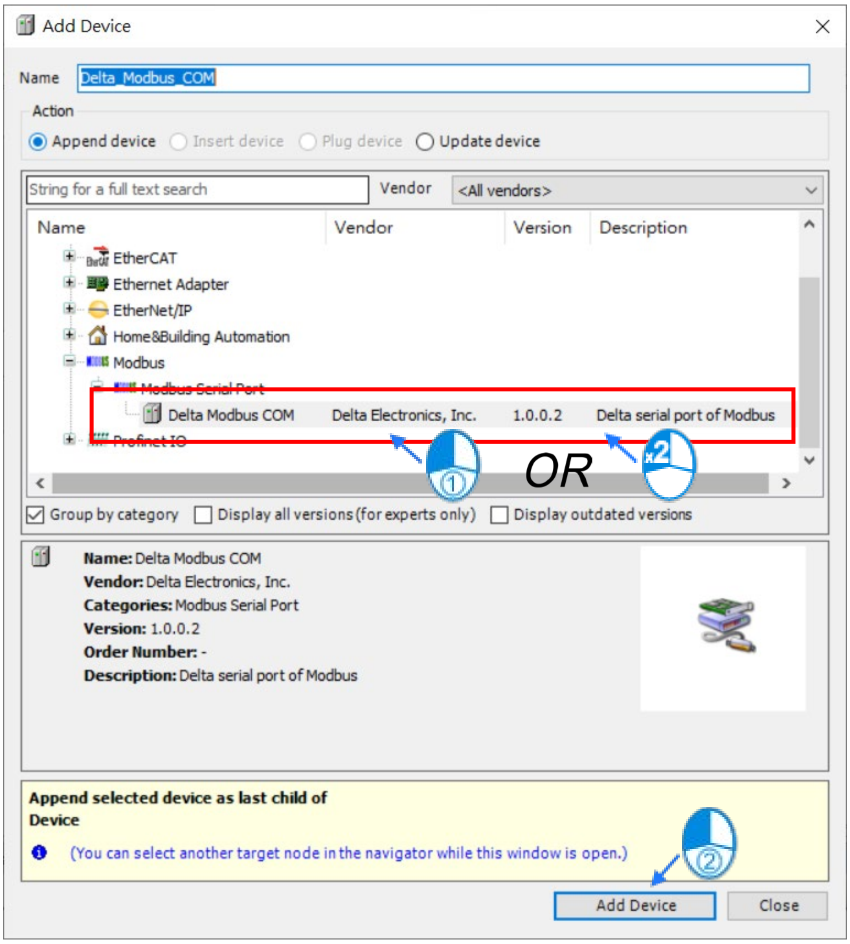
Follow the below section to set up the basic settings for communication through the Modbus serial port.

8.2.1.1 Adding Delta Modbus COM

1. Right-click the PLC in the tree view to open a context menu. And click **Add Device…**to open the Add Device setting page.



2. Find **Delta Modbus COM** (Modbus -> Modbus Serial Port -> Delta Modbus COM) and then double-click it or click **Add Device** to add this port in.



Name: Delta_Modbus_COM

● Append device

Vendor: <All vendors>

Name	Vendor	Version	Description
Delta Modbus COM	Delta Electronics, Inc.	1.0.0.2	Delta serial port of Modbus

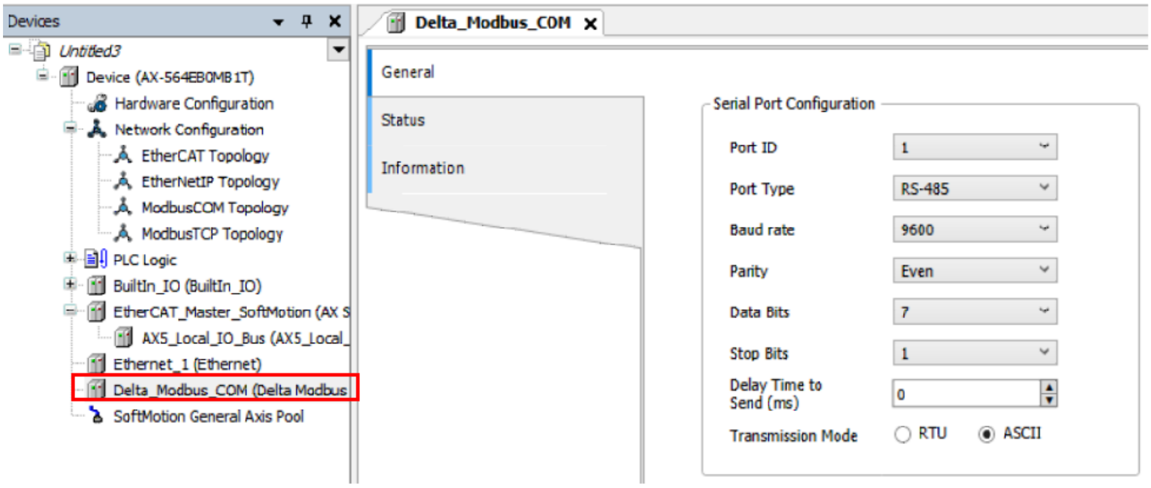
☑ Group by category

Name: Delta Modbus COM
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Port
Version: 1.0.0.2
Order Number: -
Description: Delta serial port of Modbus

Append selected device as last child of
Device

(You can select another target node in the navigator while this window is open.)

3. Find the added port **Delta_Modbus_COM (Delta Modbus COM)** in the tree view and double-click it to open the setting page to set up.

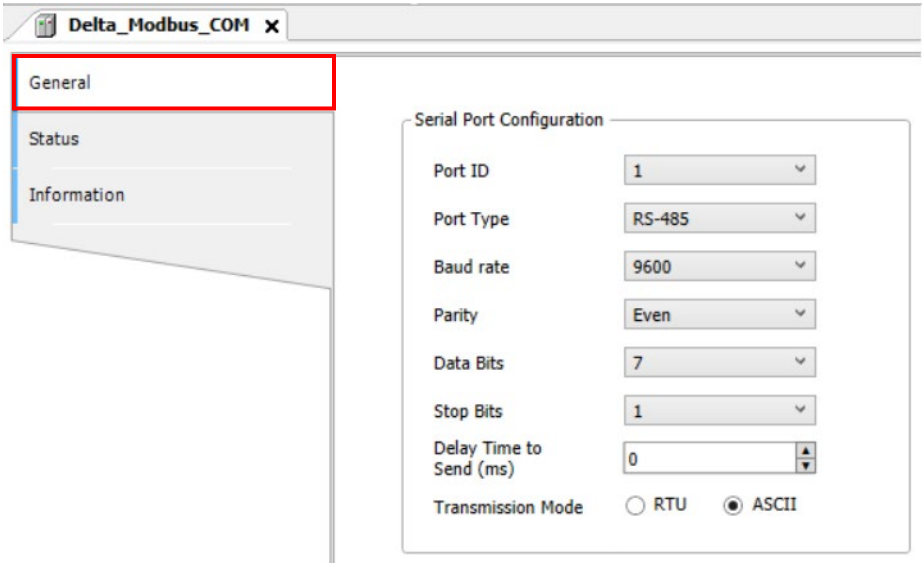


Port ID: 1
Port Type: RS-485
Baud rate: 9600
Parity: Even
Data Bits: 7
Stop Bits: 1
Delay Time to Send (ms): 0
Transmission Mode: ○ RTU ● ASCII

8.2.1.2 Setting up Delta Modbus COM

- **General**

  Here you can configure Serial Port Parameters. Settings include COM Port (RS-232 /RS-485), Baud rate, Parity, Data Bits, Stop Bits and Transmission Mode can be set here.



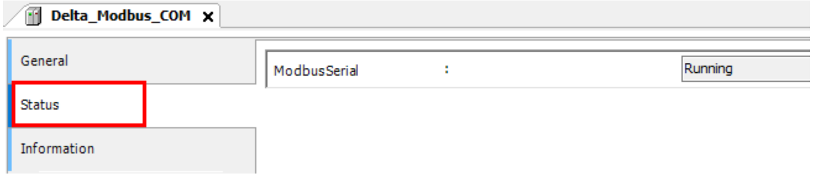
Serial Port Configuration

Port ID: 1
Port Type: RS-485
Baud rate: 9600
Parity: Even
Data Bits: 7
Stop Bits: 1
Delay Time to Send (ms): 0
Transmission Mode: ○ RTU ● ASCII

Item	Description
Port ID	Com port number: 0 (RS-232), 1 (RS-485)
Port Type	Communication interface: RS-232, RS-485
Baud rate	9600, 19200, 38400, 57600, 115200
Parity	None, Odd, Event
Data Bits	7, 8 (when setting the transmission mode to RTU, you need to set the data bits to 8)
Stop Bits	1 bit, 2bits
Delay Time to Send (ms)	0 to 3000 ms
Transmission Mode	RTU, ASCII

- **Status**

  Here you can find the operational status, e.g. ‘Running’ or ‘Stopped’ for the Delta Modbus COM.

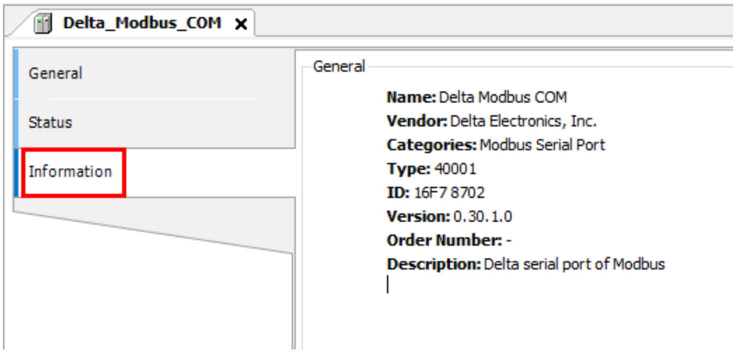


ModbusSerial : Running

Item	Description
Modbus Serial	The status of Modbus Serial Communication

- **Information**

  Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



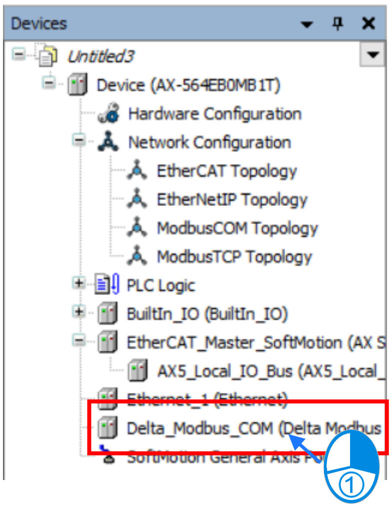
Name: Delta Modbus COM
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Port
Type: 40001
ID: 16F7 8702
Version: 0.30.1.0
Order Number: -
Description: Delta serial port of Modbus

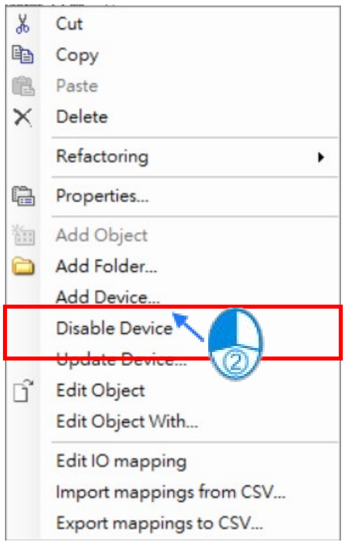
8.2.2 Modbus Serial Master

The AX-5 series controller not only supports the standard Modbus communication protocol but also includes internal device conversion for Delta PLCs (X, M, D devices), no need to check for conversion anymore. Once you created Modbus Master COM port and then Modbus Slave COM port AX-5 Series, PLC can act as a Modbus Serial Master. Follow the below section to set up the Modbus Serial Master.

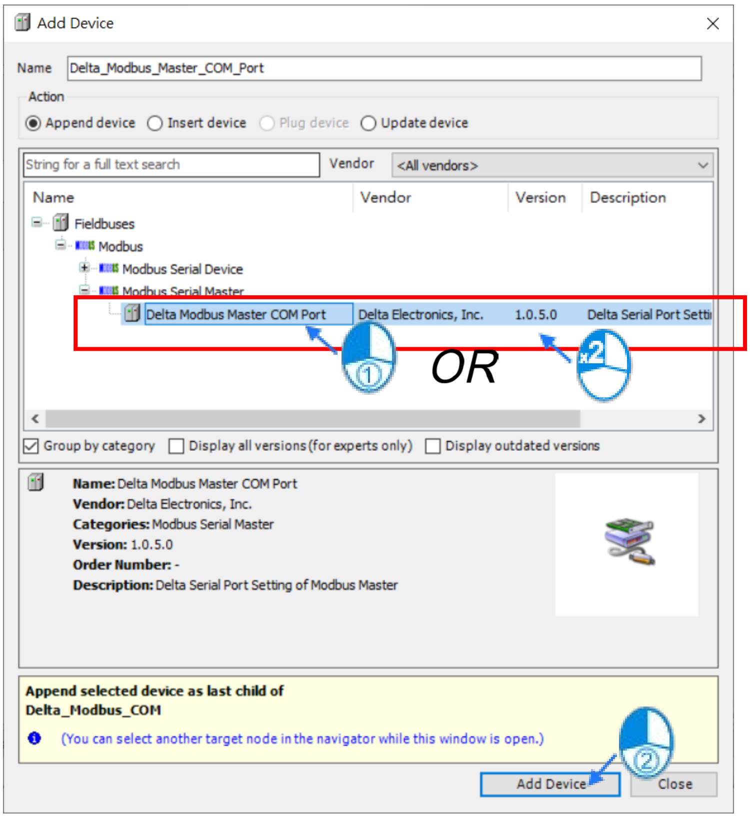
8.2.2.1 Adding Delta Modbus Master/Slave COM

1. Right-click the created Delta_Modbus_COM (Delta Modbus COM) in the tree view to open a context menu. And click **Add Device…**to open the Add Device setting page.





2. Find and double-click **Delta Modbus Master COM Port** (Fieldbuses -> Modbus -> Modbus Serial Master -> Delta Modbus Master COM Port) or click **Add Device** to add this port in.



Name Delta_Modbus_Master_COM_Port

● Append device

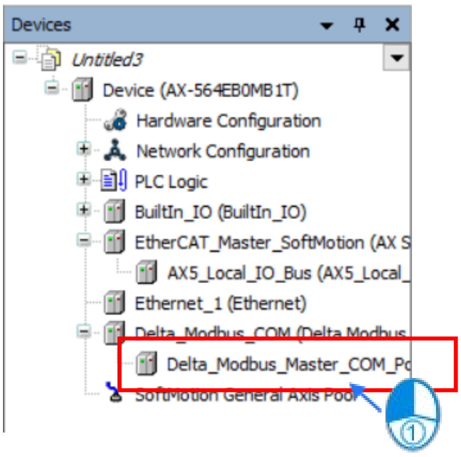
Name	Vendor	Version	Description
Delta Modbus Master COM Port	Delta Electronics, Inc.	1.0.5.0	Delta Serial Port Setti

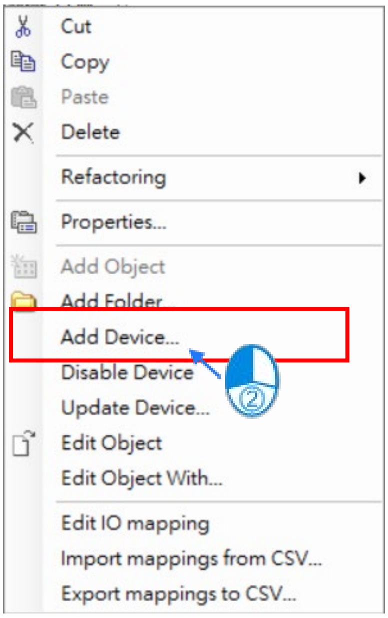
☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: Delta Modbus Master COM Port
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Master
Version: 1.0.5.0
Order Number: -
Description: Delta Serial Port Setting of Modbus Master

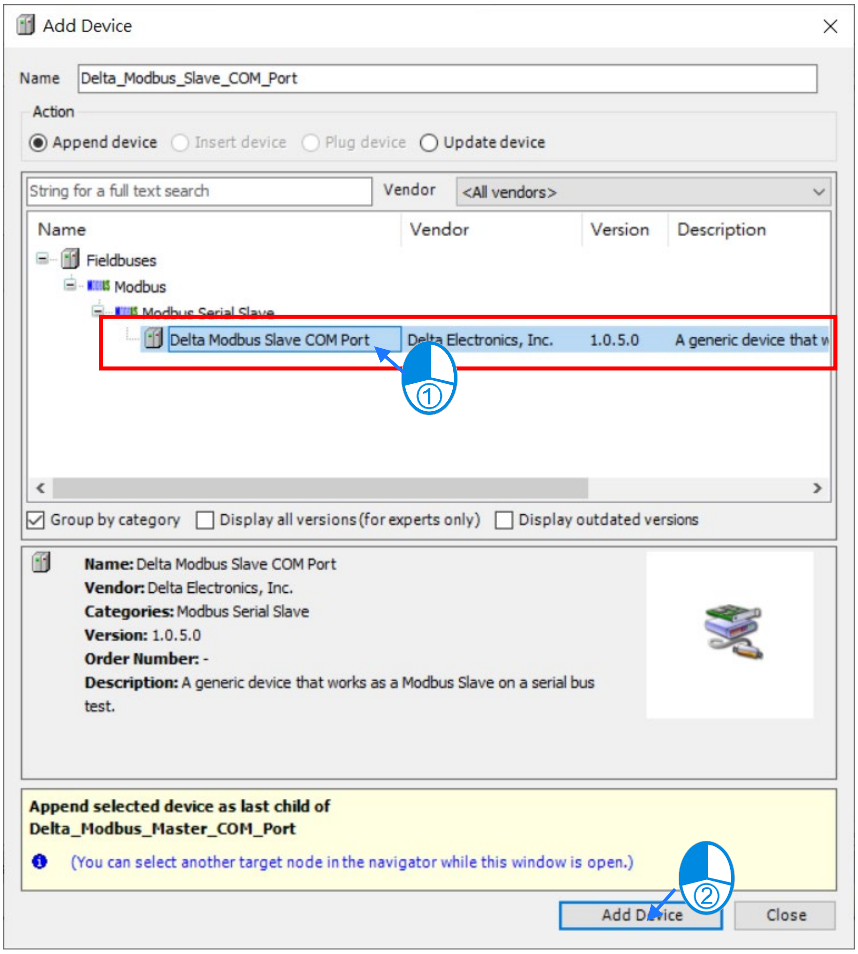
Append selected device as last child of
Delta_Modbus_COM
(You can select another target node in the navigator while this window is open.)

3. Right-click the created Delta_Modbus_Master_COM_Port in the tree view to open a context menu. And click **Add Device…**to open the Add Device setting page.





4. Find and double-click **Delta Modbus Slave COM Port** (Fieldbuses -> Modbus -> Modbus Serial Slave -> Delta Modbus Slave COM Port) or click **Add Device** to add this port in.



Name: Delta_Modbus_Slave_COM_Port

Name	Vendor	Version	Description
Delta Modbus Slave COM Port	Delta Electronics, Inc.	1.0.5.0	A generic device that w

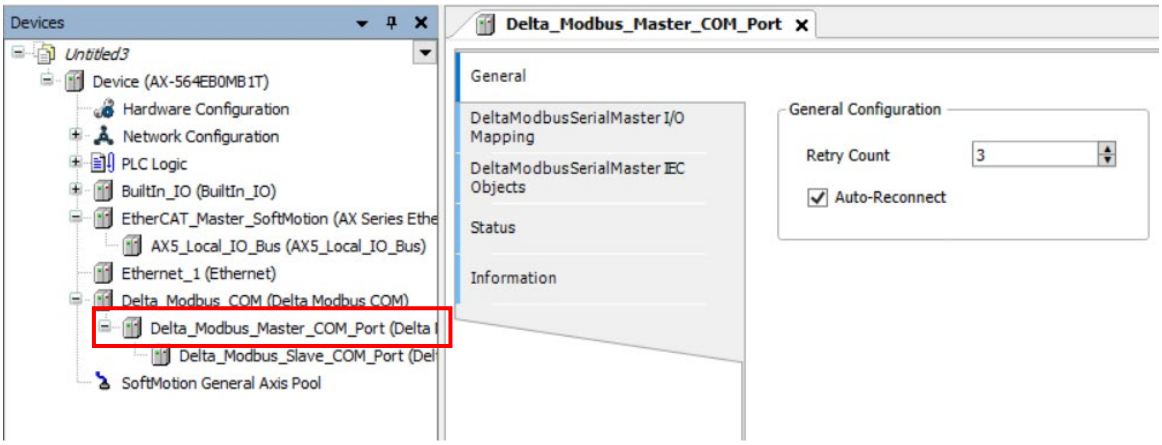
Name: Delta Modbus Slave COM Port
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Slave
Version: 1.0.5.0
Order Number: -
Description: A generic device that works as a Modbus Slave on a serial bus test.

Append selected device as last child of
Delta_Modbus_Master_COM_Port

(You can select another target node in the navigator while this window is open.)

8.2.2.2 Setting up Delta Modbus Master COM

1. Find the added port **Delta_Modbus_Master_COM (Delta Modbus COM)** in the tree view and double-click it to open the setting page to set up.

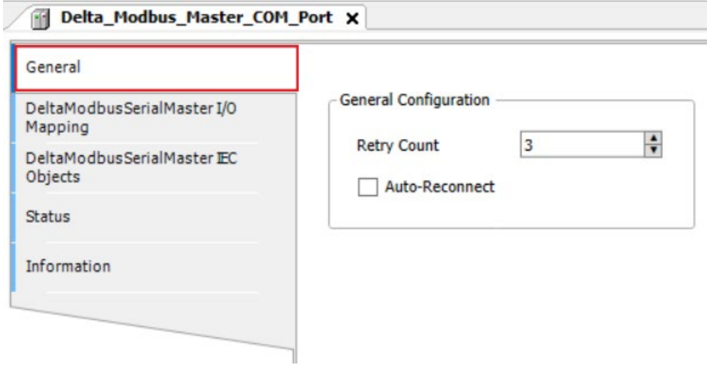


Retry Count 3

☑ Auto-Reconnect

- **General**

  Here you can configure the basic settings for Modbus Serial Master COM Port.



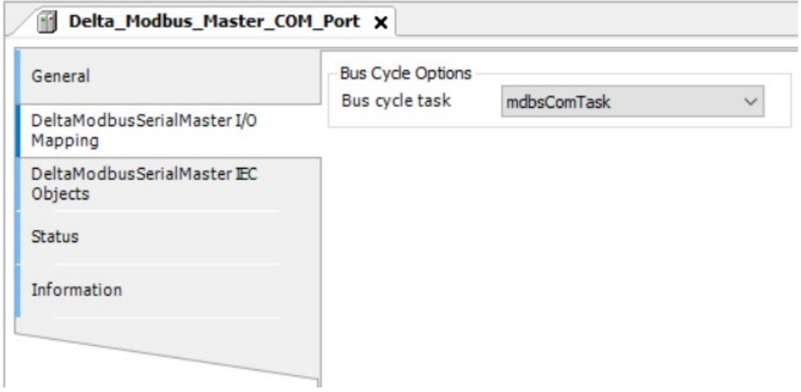
Retry Count 3

☐ Auto-Reconnect

Item	Description
Retry Count	Set up the number of times for the COM port to reconnect if the connection is lost. (Default: 3)
Auto-Reconnect	Enable this option to have this port to reconnect automatically if an error occurs or connection timeout occurs.

- **Delta Modbus Serial Master I/O Mapping**

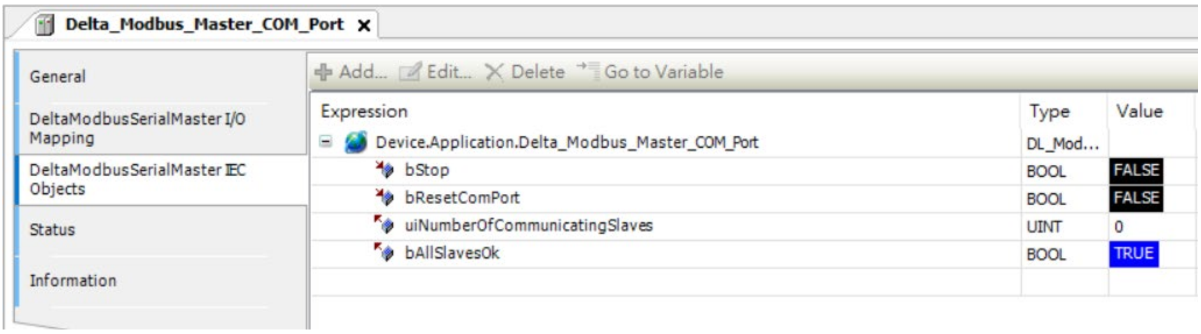
  Bus cycle task: Select a bus cycle task to synchronize with the Modbus communication time. When the option “Use parent bus cycle setting is selected”, the system uses the shortest cycle time as the bus cycle time. Refer to section 4.2.1.6 PLC Settings for more information.



Bus cycle task mdbsComTask

- **Delta Modbus Serial Master IEC Objects**

  Here is the corresponding of the DFB_ModbusCOMMaster function block. You can check the status of Modbus Serial Master under this tab.

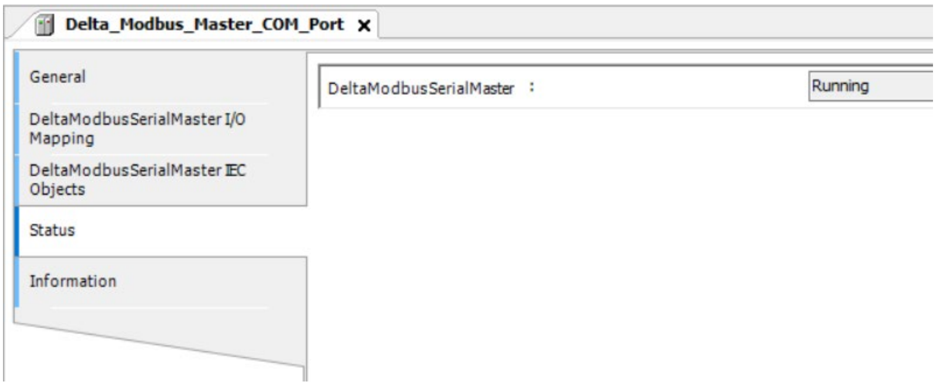


Expression	Type	Value
Device.Application.Delta_Modbus_Master_COM_Port	DL_Mod...	
bStop	BOOL	FALSE
bResetComPort	BOOL	FALSE
uiNumberOfCommunicatingSlaves	UINT	0
bAllSlavesOk	BOOL	TRUE

Item	Description
bStop	Stop sending the Slave any new request
bResetComPort	Reset the COM port
uiNumberOfCommunicatingSlaves	Number of the Slaves that are in communication
bAllSlavesOk	The communication status of the Slave

- **Status**

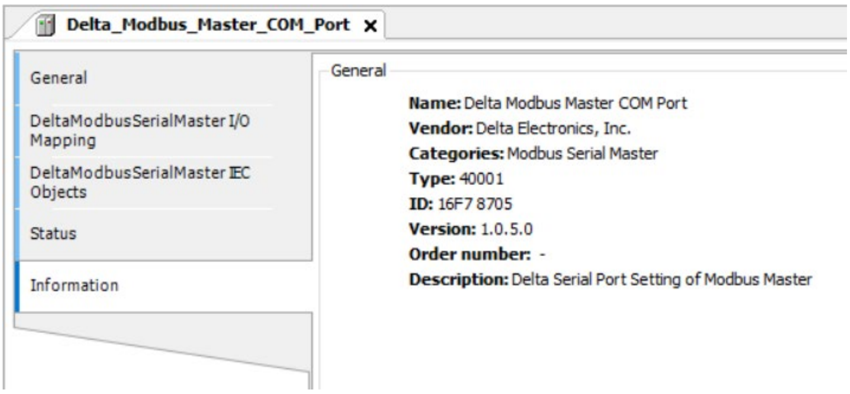
  Here you can find the device status information, for example ‘Running’ or ‘Stopped’, and “specific diagnostic” messages from the respective device.



DeltaModbusSerialMaster : Running

- **Information**

  Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.

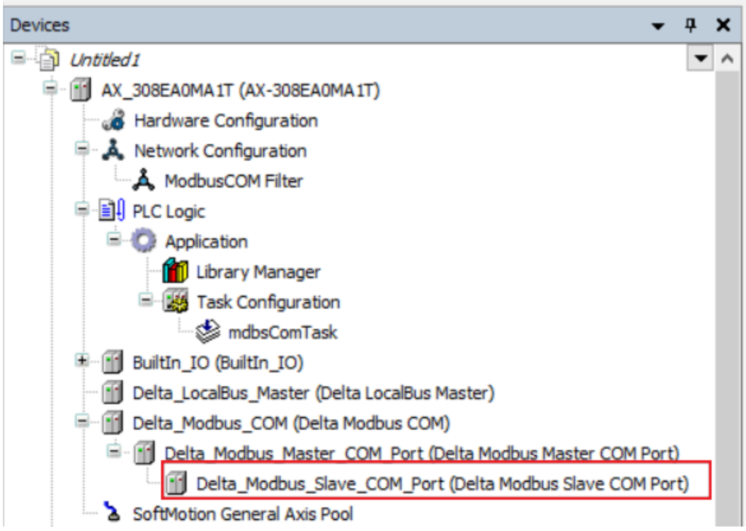


General

Name: Delta Modbus Master COM Port
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Master
Type: 40001
ID: 16F7 8705
Version: 1.0.5.0
Order number: -
Description: Delta Serial Port Setting of Modbus Master

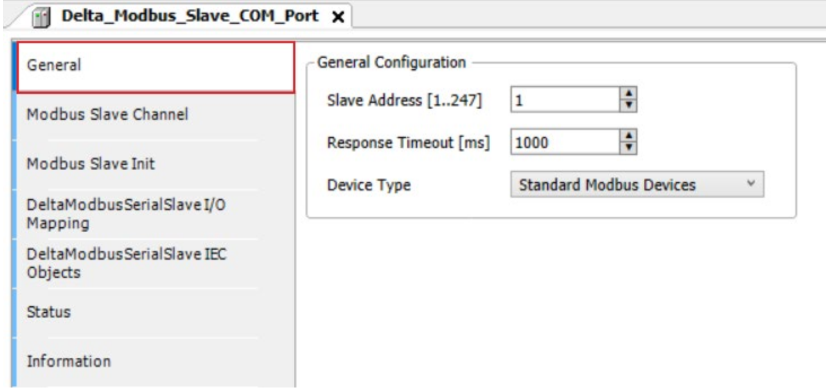
8.2.2.3 Setting up Delta Modbus Slave COM

In the tree view, find the added port **Delta_Modbus_Slave_COM_Port (Delta Modbus Slave COM Port).** Double-click it to open the setting page.



- **General**

Here you can configure the basic settings for Modbus Serial Slave COM Port, such as Slave Address, Response Timeout and Device Type.



General Configuration

Slave Address [1..247]: 1

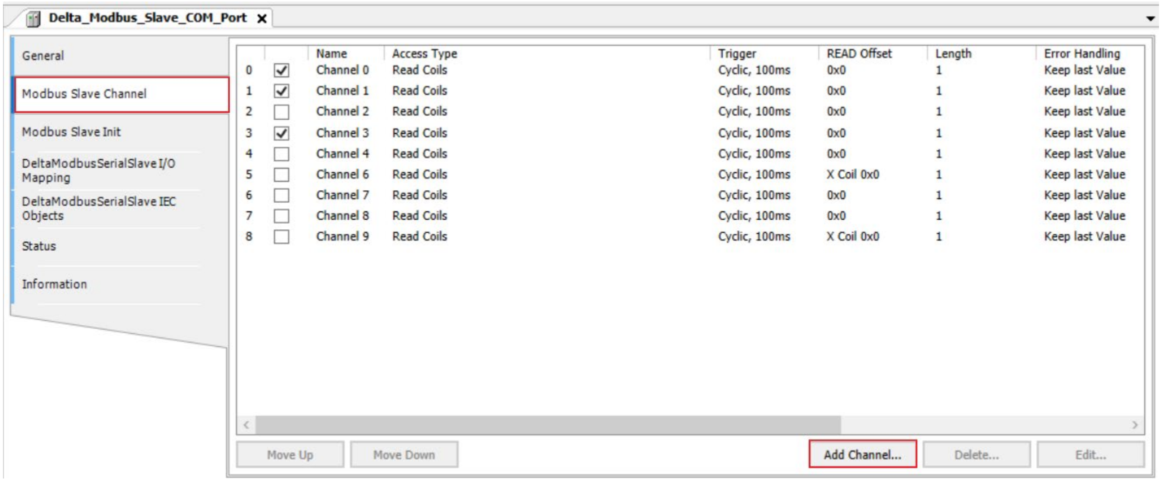
Response Timeout [ms]: 1000

Device Type: Standard Modbus Devices

Item	Description
Slave Address	Address of a serial Modbus device (between 1 and 247)
Response Timeout	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.
Device Type	You can select standard Modbus devices or Delta devices. If you select Delta devices, the system converts the protocol used into Modbus protocol automatically so that you do NOT need to refer to the register map for the conversion.

- **Modbus Slave Channel**

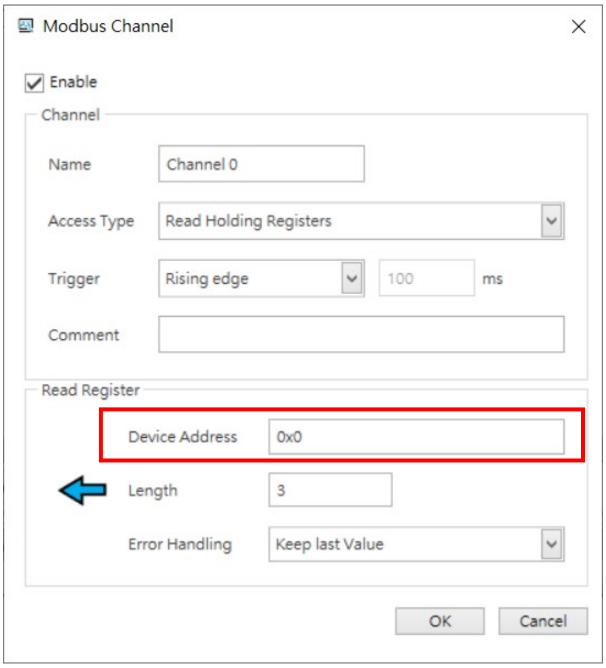
Here you can define slave channels. Each channel represents a single Modbus request. You can create up to 10 channels for each slave. AX-5 Series PLC will send out Modbus request packets in chronological order. All channels share the same Modbus connection.



		Name	Access Type	Trigger	READ Offset	Length	Error Handling
0	☑	Channel 0	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
1	☑	Channel 1	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
2	☐	Channel 2	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
3	☑	Channel 3	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
4	☐	Channel 4	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
5	☐	Channel 6	Read Coils	Cyclic, 100ms	X Coil 0x0	1	Keep last Value
6	☐	Channel 7	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
7	☐	Channel 8	Read Coils	Cyclic, 100ms	0x0	1	Keep last Value
8	☐	Channel 9	Read Coils	Cyclic, 100ms	X Coil 0x0	1	Keep last Value

Click **Add Channel** to open the setting page; you can edit the channel before adding it in. The **Device Address** shows the Modbus protocol address you selected whether the device type is **Standard Modbus Device** or **Delta Devices** (D register, M coil, X coil, etc.) under the **General** tap. Since the system converts the protocol used into Modbus protocol automatically, you do NOT need to refer to the register map for the conversion.

Device Type：Standard Modbus Device



☑ Enable

Channel

Name: Channel 0

Access Type: Read Holding Registers

Trigger: Rising edge 100 ms

Comment:

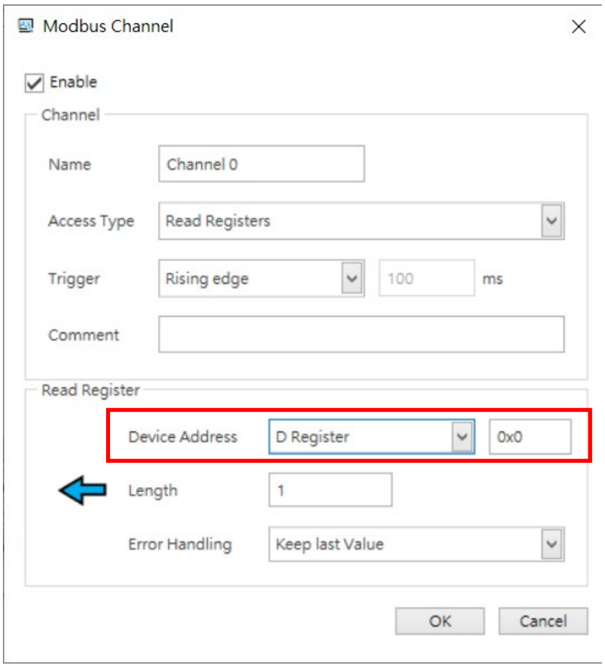
Read Register

Device Address: 0x0

Length: 3

Error Handling: Keep last Value

Device Type：AH Series



☑ Enable

Channel

Name: Channel 0

Access Type: Read Registers

Trigger: Rising edge 100 ms

Comment:

Read Register

Device Address: D Register 0x0

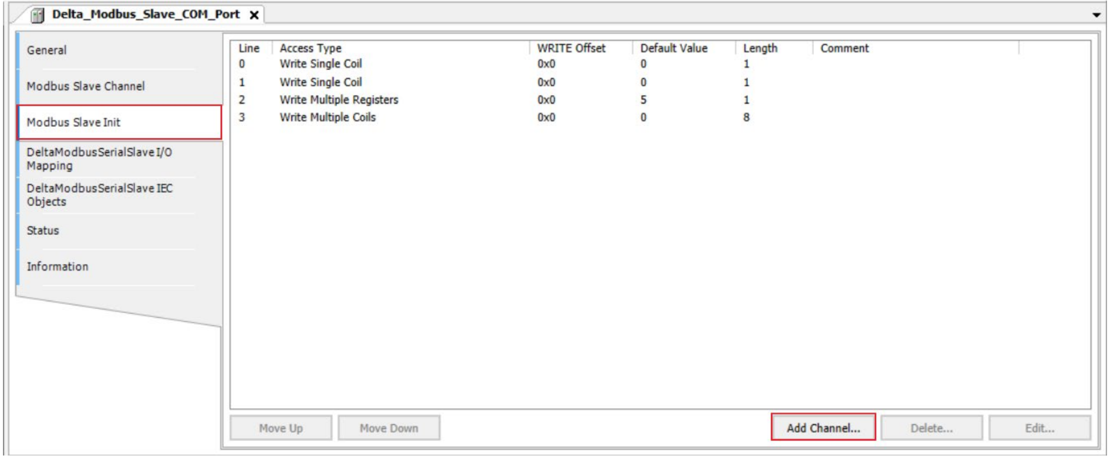
Length: 1

Error Handling: Keep last Value

Item	**Description**	
Device Type	**Standard Modbus Device**	**Delta Series Device**
Enable	Activate this channel	
Name	Define this channel name	
Access Type	Modbus 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)	Read/Write Registers ● Read coils ● Read registers ● Write coils ● Write registers Note: PLC uses the corresponding Modbus function code according to the read/write register of the device type.
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 **DFB_ModbusComChannel**	● **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 **DFB_ModbusComChannel**
Comment	Description of the channel	
Device Address	Modbus protocol address	Delta register address (will be converted into Modbus protocl in the background)
Length	Number of the register to be read/written to. (up to 100 coils and 100 registers)	Number of the register to be read/written to. (up to 256 coils and 100 registers)
Error Handling	What to do with the data in case of a communication error: ● Set To ZERO ● Keep last value	

■ **Modbus Slave Init**

After the Modbus connection between AX-5 Series PLC and the slaves is established, you can use **Add Channel** button to edit the Initialization Value of the Coil/Register.



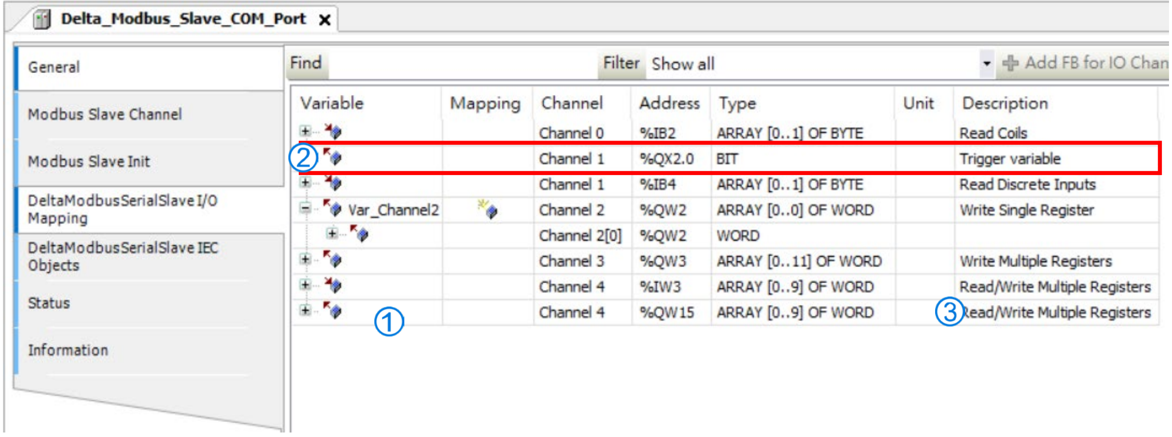
Line	Access Type	WRITE Offset	Default Value	Length	Comment
0	Write Single Coil	0x0	0	1	
1	Write Single Coil	0x0	0	1	
2	Write Multiple Registers	0x0	5	1	
3	Write Multiple Coils	0x0	0	8	

Click **Add Channel**, you can edit the Access Type, Device Address, Length, Initialization Value and Comment. Click OK to confirm the settings.

Access Type: Write Multiple Registers
Device Address: 0x0
Length: 1
Initialization Value: 5
Comment:

Delta Modbus Generic Serial Slave I/O Mapping

After you have added channels under the tab of Modbus Slave Channel, you can find the variables and the set access types under this tab and to define the variables for mapping.



Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%IB2	ARRAY [0..1] OF BYTE		Read Coils
		Channel 1	%QX2.0	BIT		Trigger variable
		Channel 1	%IB4	ARRAY [0..1] OF BYTE		Read Discrete Inputs
Var_Channel2		Channel 2	%QW2	ARRAY [0..0] OF WORD		Write Single Register
		Channel 2[0]	%QW2	WORD		
		Channel 3	%QW3	ARRAY [0..11] OF WORD		Write Multiple Registers
		Channel 4	%IW3	ARRAY [0..9] OF WORD		Read/Write Multiple Registers
		Channel 4	%QW15	ARRAY [0..9] OF WORD		Read/Write Multiple Registers

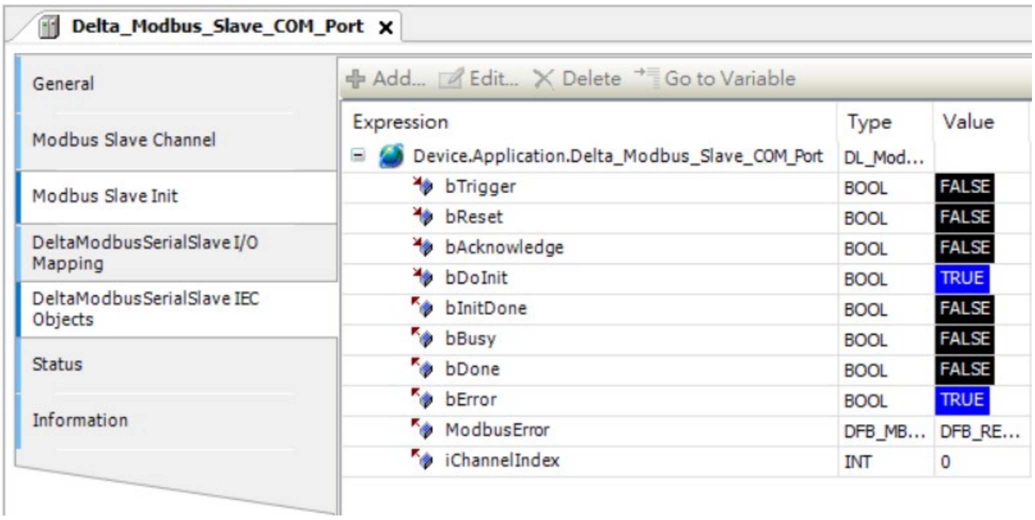
① The descriptions here reflect what you have set for the **Access Type.**

② When the Boolean variable **Trigger type** for this channel is set to **Rising edge**, the description here generates **Trigger variable**.

③ The device type for this channel is **Delta Devices** (coils/registers).

Delta Modbus Serial Slave IEC Objects

You can check the status of Modbus Serial Slave under this tab.

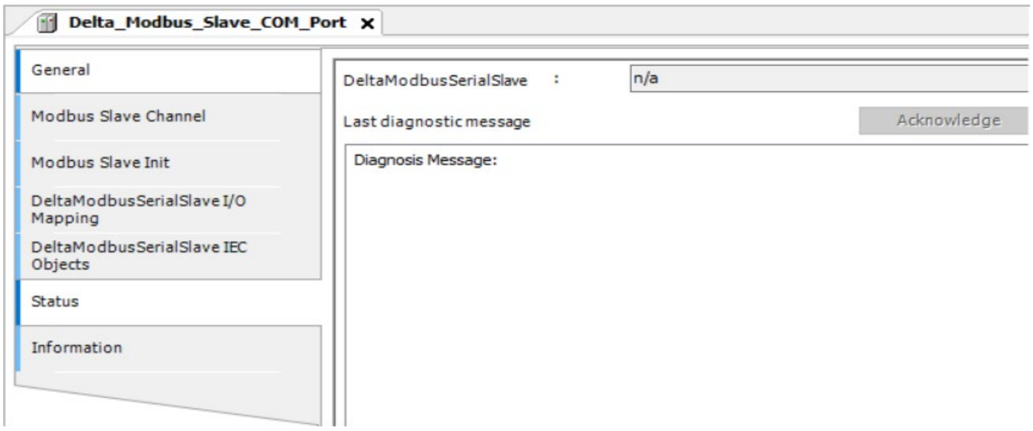


Expression	Type	Value
Device.Application.Delta_Modbus_Slave_COM_Port	DL_Mod...	
bTrigger	BOOL	FALSE
bReset	BOOL	FALSE
bAcknowledge	BOOL	FALSE
bDoInit	BOOL	TRUE
bInitDone	BOOL	FALSE
bBusy	BOOL	FALSE
bDone	BOOL	FALSE
bError	BOOL	TRUE
ModbusError	DFB_MB...	DFB_RE...
iChannelIndex	INT	0

Item	Description
bTrigger	Activate all Modbus channels simultaneously.
bReset	Re-establish the connection and reset bError and ModbusError when the connection status shows error. And this function is only available when the option “Auto-Reconnect” is NOT enabled.
bAcknowledge	Re-establish the connection. The data exchange resumes from the Modbus channel where the previous error occurred. And this function is only available when the option “Auto-Reconnect” is NOT enabled.
bDoInit	Initialized the Slave
bInitDone	The initialization of the Slave is complete.
bBusy	This channel is executing data exchange.
bDone	The data exchange via this channel is complete.
bError	Error occurs when this channel is executing data exchange.
ModbusError	Record Modbus errors occurred.
iChannelIndex	The channel number that is in execution.

- **Status**

  Here you can find the Modbus Slave COM Port status information, for example ‘Running’ or ‘Stopped’, and specific diagnostic messages from the connected device, also information about the card used and the internal bus system.



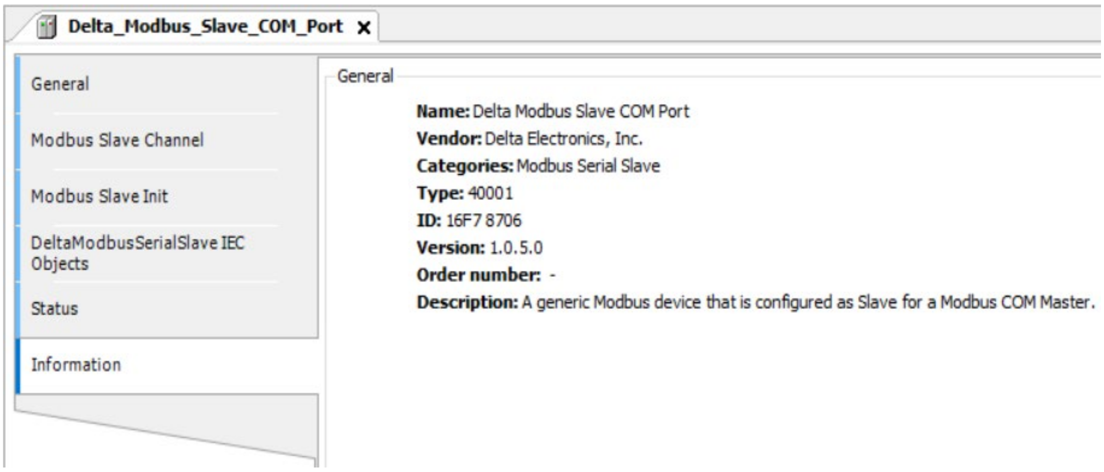
DeltaModbusSerialSlave : n/a

Last diagnostic message

Diagnosis Message:

- **Information**

  Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



General

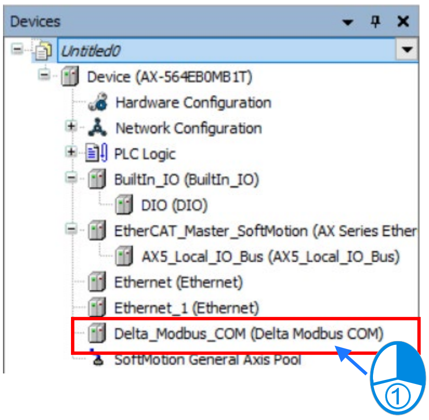
Name: Delta Modbus Slave COM Port
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Slave
Type: 40001
ID: 16F7 8706
Version: 1.0.5.0
Order number: -
Description: A generic Modbus device that is configured as Slave for a Modbus COM Master.

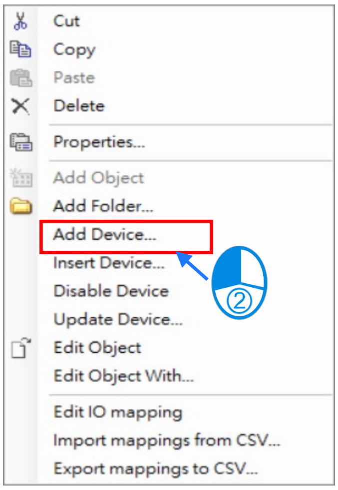
8.2.3 Modbus Serial Slave

AX-5 Series PLC can act as a Modbus Serial Slave, after you add Modbus Serial Device in and set up the allowable areas for Coils/Register. If Modbus Serial Master uses Delta device communication protocol, there is no access restrictions. Follow the below section to set up the Modbus Serial Slave.

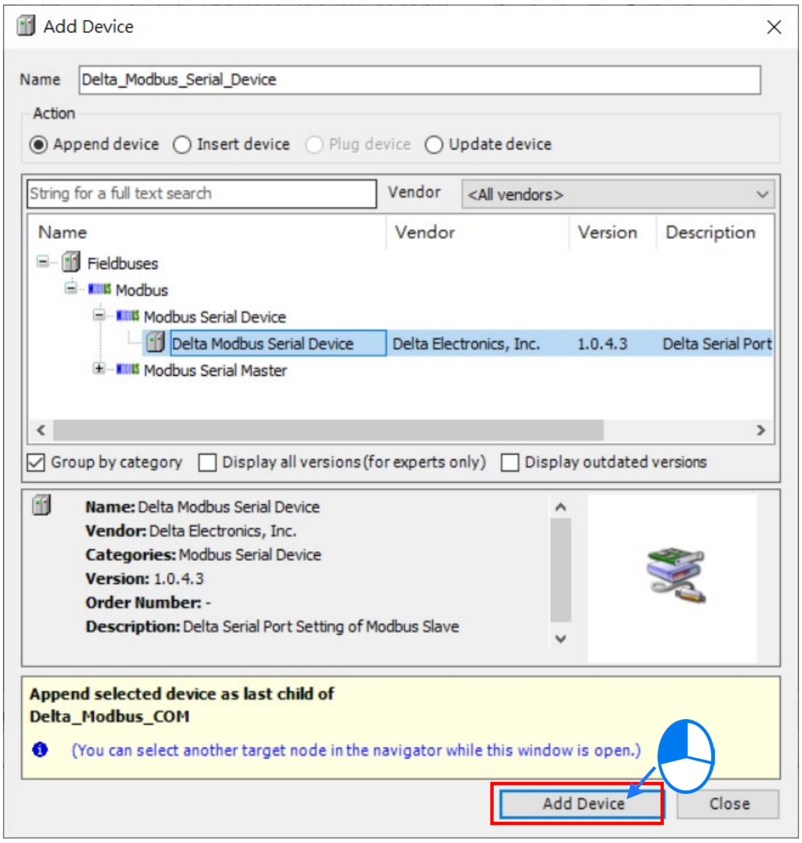
8.2.3.1 Adding a Modbus Serial Slave Device

1. Right-click the created Delta_Modbus_COM (Delta Modbus COM) in the tree view to open up a context menu. And click **Add Device…**to open the Add Device setting window.





2. Find and double-click **Delta Modbus Serial Device** (Fieldbuses -> Modbus -> Modbus Serial Master -> Delta Modbus Serial Device) or click **Add Device** to add this device in.



Name: Delta_Modbus_Serial_Device

Name: Delta Modbus Serial Device
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Device
Version: 1.0.4.3
Order Number: -
Description: Delta Serial Port Setting of Modbus Slave

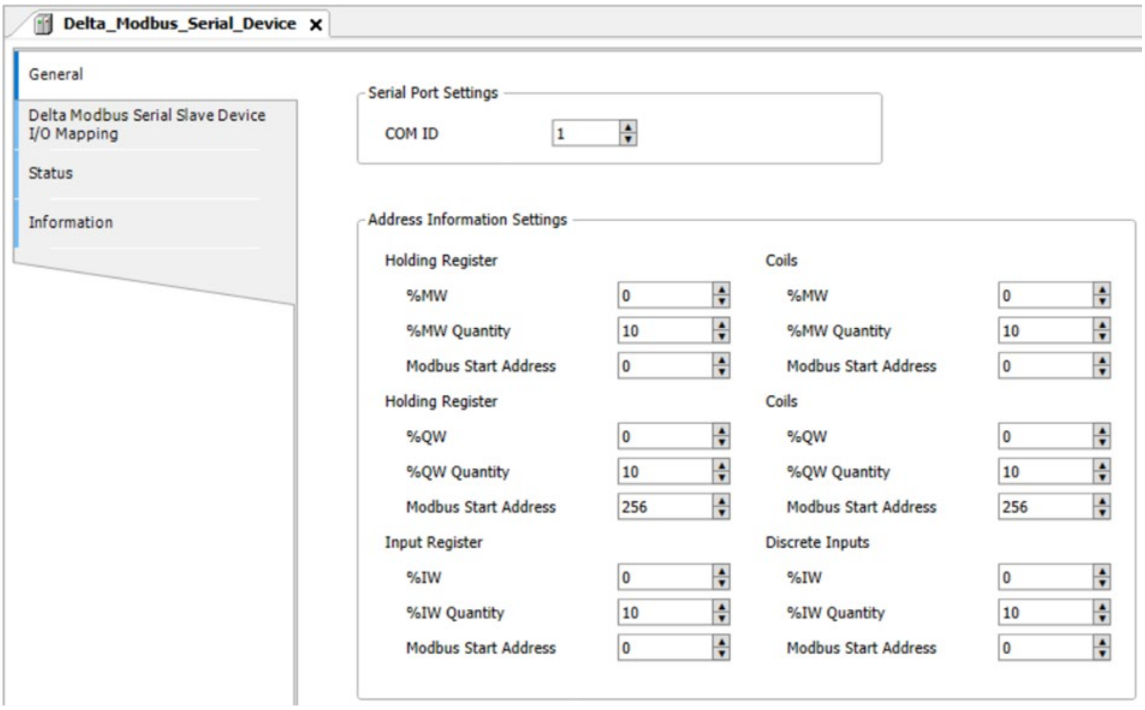
Append selected device as last child of
Delta_Modbus_COM

(You can select another target node in the navigator while this window is open.)

8.2.3.2 Setting up the Modbus Serial Slave

- **General**

  Here you can configure the basic settings for Modbus Serial Device. Set up the allowable areas for Coils/Register. If Modbus Serial Master uses Delta device communication protocol, there is no access restrictions.



Serial Port Settings

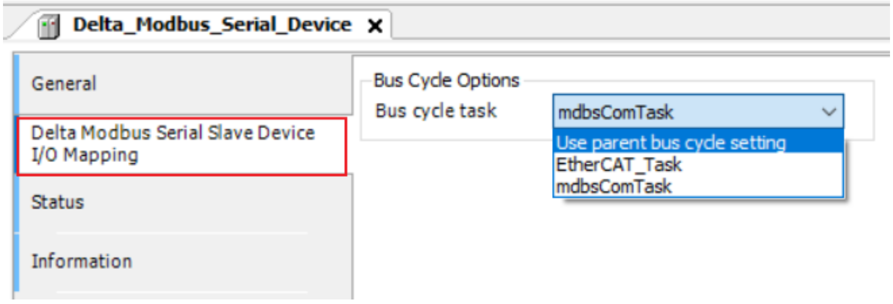
COM ID: 1

Address Information Settings

Holding Register		Coils	
%MW	0	%MW	0
%MW Quantity	10	%MW Quantity	10
Modbus Start Address	0	Modbus Start Address	0
Holding Register		Coils	
%QW	0	%QW	0
%QW Quantity	10	%QW Quantity	10
Modbus Start Address	256	Modbus Start Address	256
Input Register		Discrete Inputs	
%IW	0	%IW	0
%IW Quantity	10	%IW Quantity	10
Modbus Start Address	0	Modbus Start Address	0

- **Delta Modbus Serial Slave Device I/O Mapping**

  Bus cycle task: Select a bus cycle task to synchronize with the Modbus communication time. When the option "Use parent bus cycle setting is selected", the system uses the shortest cycle time as the bus cycle time. Refer to section 4.2.1.6 PLC Settings for more information.

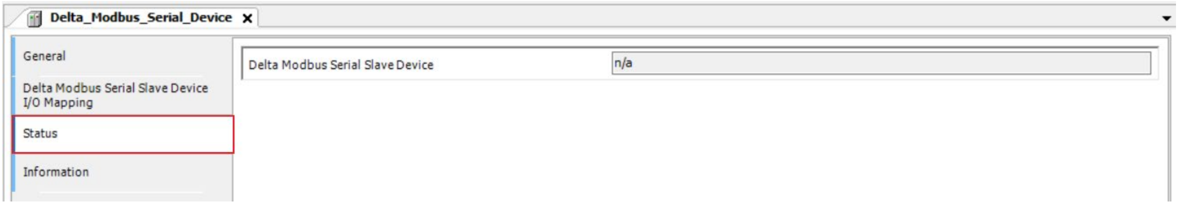


Bus Cycle Options

Bus cycle task: mdbsComTask

- **Status**

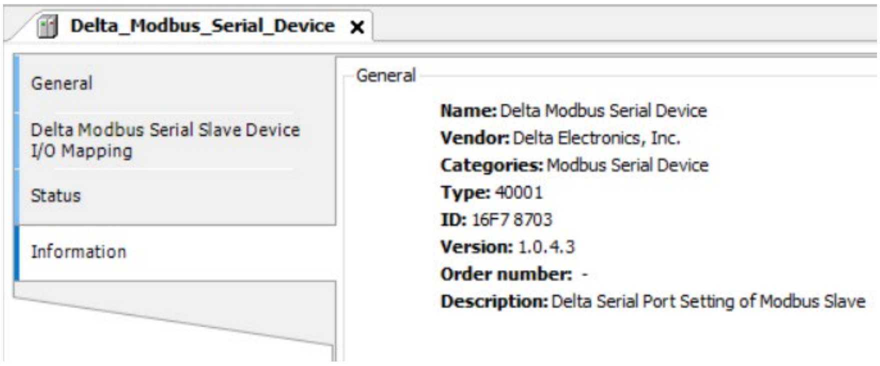
Here you can find the Modbus Serial Slave Device status information, for example 'Running' or 'Stopped', and specific diagnostic messages from the respective device, also information about the card used and the internal bus system.



Delta Modbus Serial Slave Device n/a

- **Information**

Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



General

Name: Delta Modbus Serial Device
Vendor: Delta Electronics, Inc.
Categories: Modbus Serial Device
Type: 40001
ID: 16F7 8703
Version: 1.0.4.3
Order number: -
Description: Delta Serial Port Setting of Modbus Slave

8.3 Introduction on Ethernet Communication

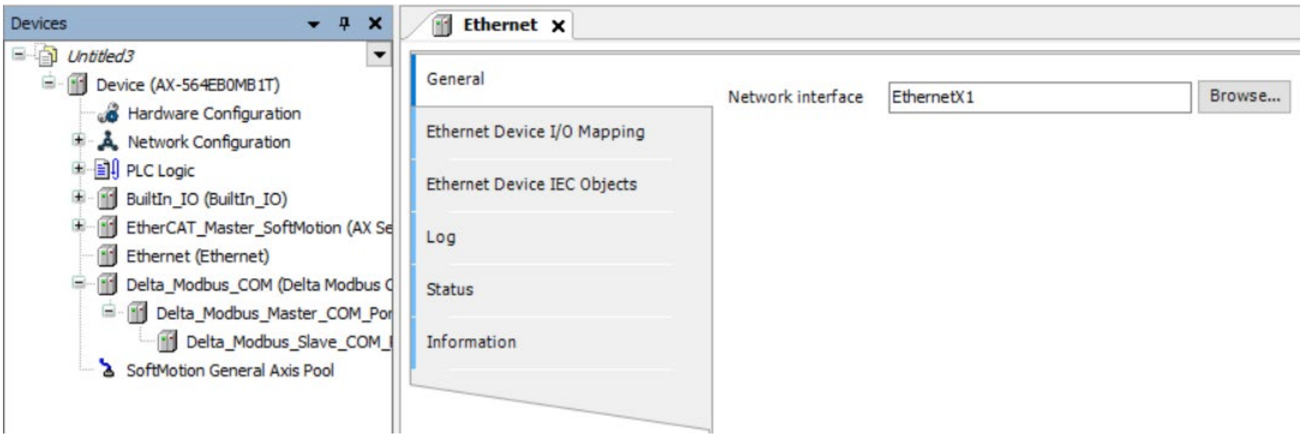
DIADesigner-AX supports the following Modbus network types, including Modbus TCP and EtherNet/IP. Follow the below section to set up the basic settings for communication through the Ethernet Adapter.

8.3.1 Network Security

We suggest you use closed network or use local network with a firewall to secure and prevent the Ethernet network as well as our products from any unwanted attack.

8.3.2 Ethernet

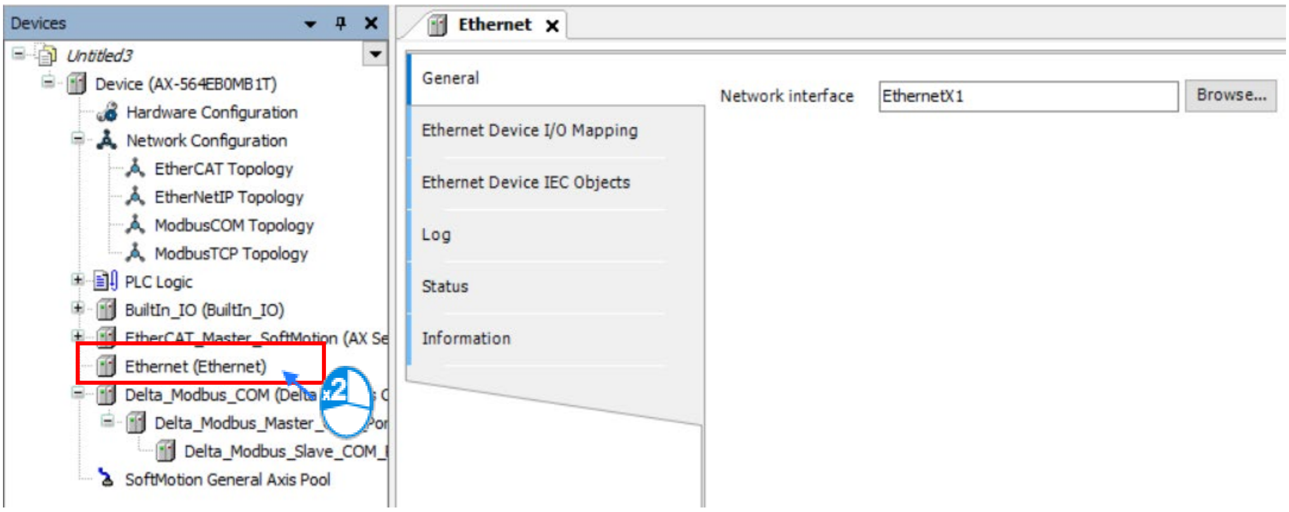
Two Ethernet ports are available for AX-5 Series CPU. You can set up two different IP addresses for the ports and then they can run individually.



Network interface EthernetX1

8.3.2.1 Setting up the Ethernet

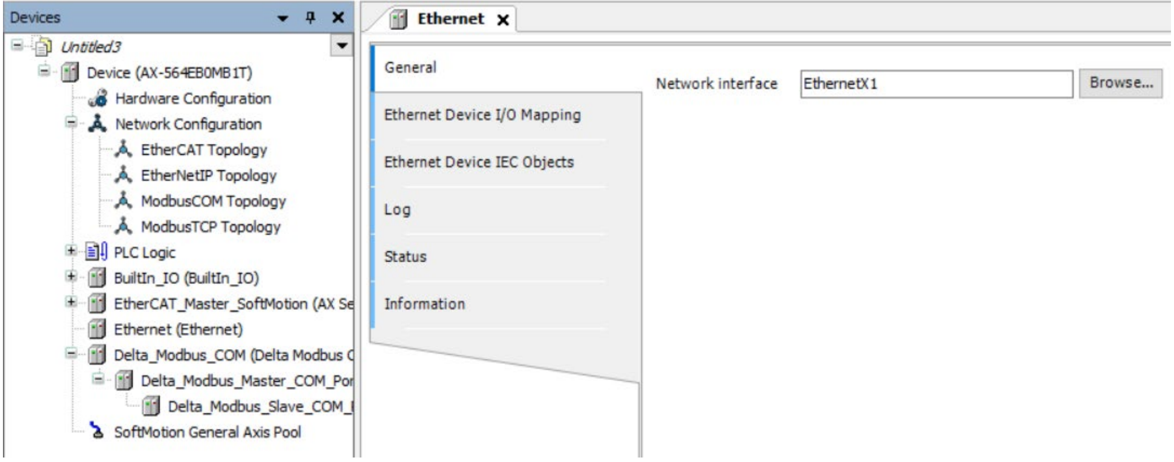
Find the added **Ethernet (Ethernet)** in the tree view and double-click it to open the setting page for setup.



Network interface EthernetX1

- **General**

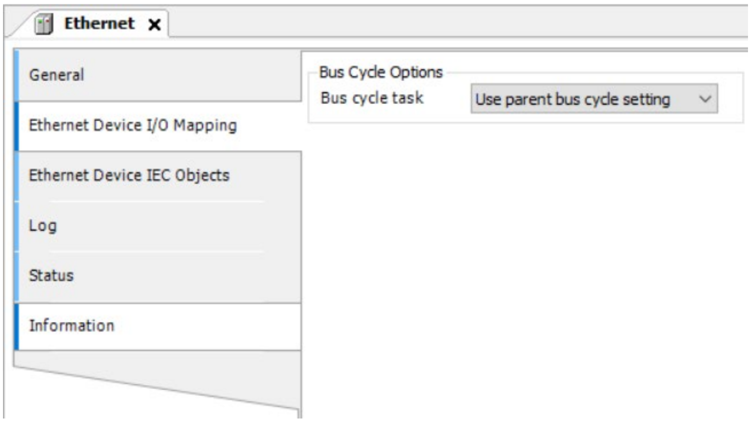
  Here you can configure Ethernet Parameters. Settings include Interface, IP address, Subnet mask, Default gateway and Adjust operating system settings can be set here.



Network interface: EthernetX1

- **Ethernet Device I/O Mapping**

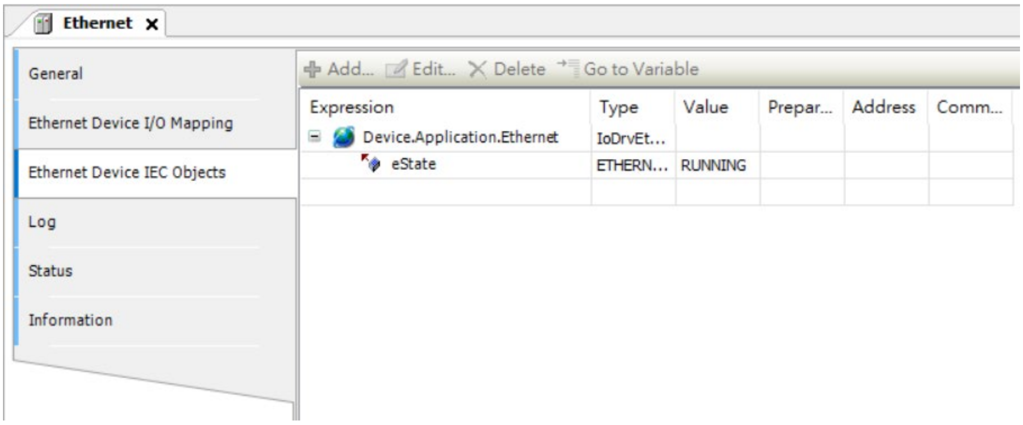
  Bus cycle task: Select a bus cycle task to synchronize with the communication time. When the option "Use parent bus cycle setting is selected", the system uses the shortest cycle time as the bus cycle time. Refer to section 4.2.1.6 PLC Settings for more information.



Bus Cycle Options
Bus cycle task: Use parent bus cycle setting

- **Ethernet Device IEC Objects**

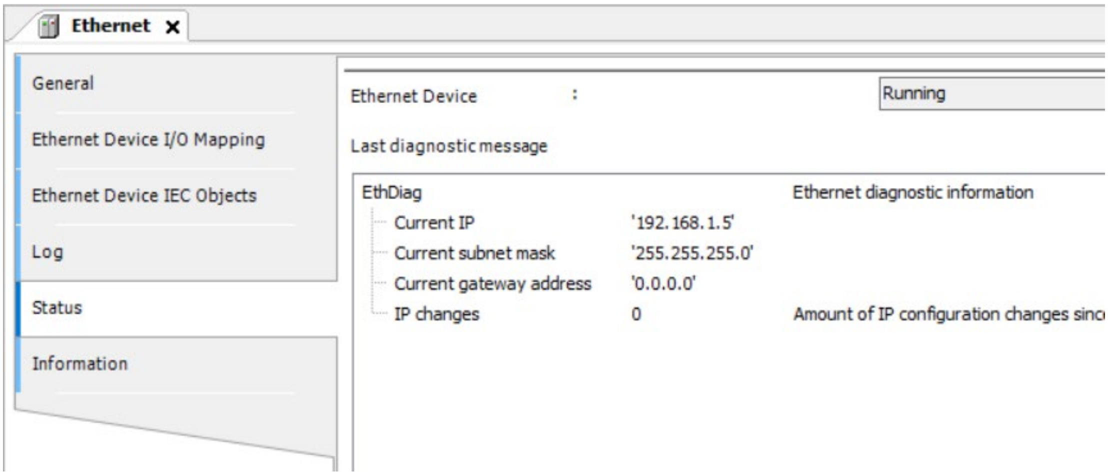
  Here you can find the objects defined by Ethernet Adapter Device. "Objects" are listed that allow for access to the device from the IEC application. In online mode, you can use the table of IEC objects as a monitoring view.



Expression	Type	Value	Prepar...	Address	Comm...
Device.Application.Ethernet	IoDrvEt...				
eState	ETHERN...	RUNNING			

- **Status**

Here you can find the device status information, for example 'Running' or 'Stopped', and specific diagnostic messages from the respective device.



Ethernet Device : Running

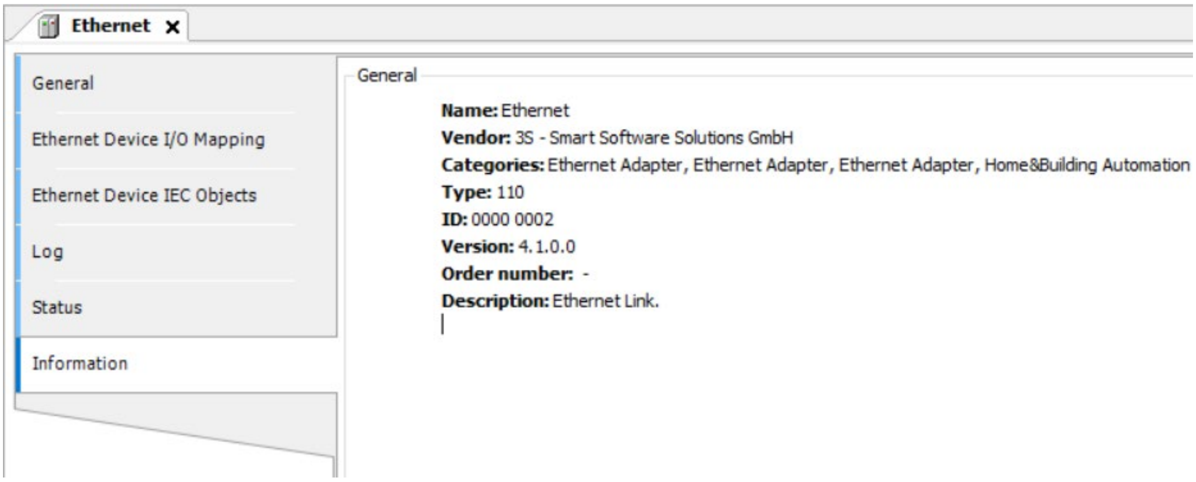
Last diagnostic message

EthDiag		Ethernet diagnostic information
Current IP	'192.168.1.5'	
Current subnet mask	'255.255.255.0'	
Current gateway address	'0.0.0.0'	
IP changes	0	Amount of IP configuration changes sinc

Item	Description
Ethernet Device	The status of Ethernet Communication
Last Diagnostic Message	Network diagnosis

- **Information**

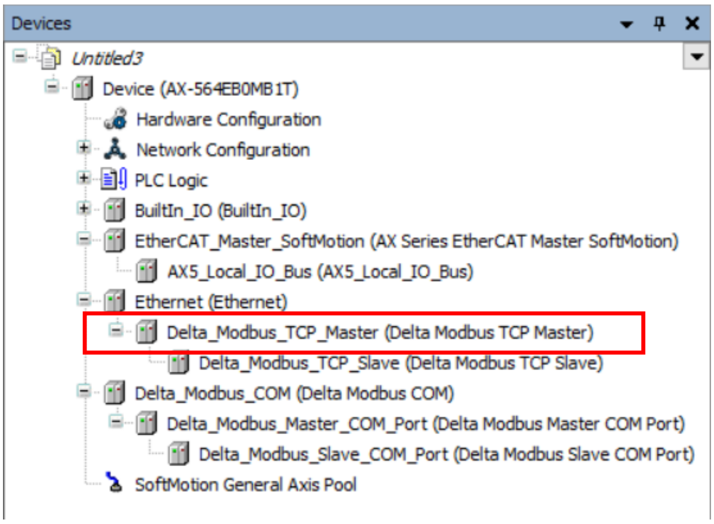
Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



Name: Ethernet
Vendor: 3S - Smart Software Solutions GmbH
Categories: Ethernet Adapter, Ethernet Adapter, Ethernet Adapter, Home&Building Automation
Type: 110
ID: 0000 0002
Version: 4.1.0.0
Order number: -
Description: Ethernet Link.

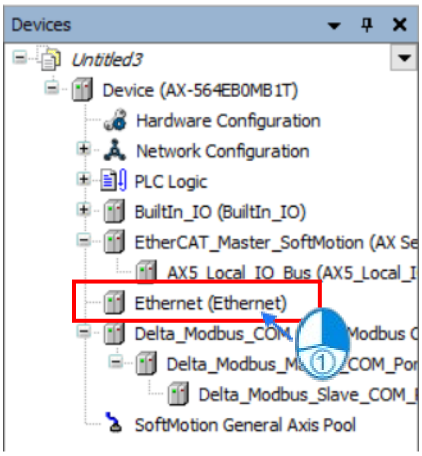
8.3.3 Modbus TCP Master (Client)

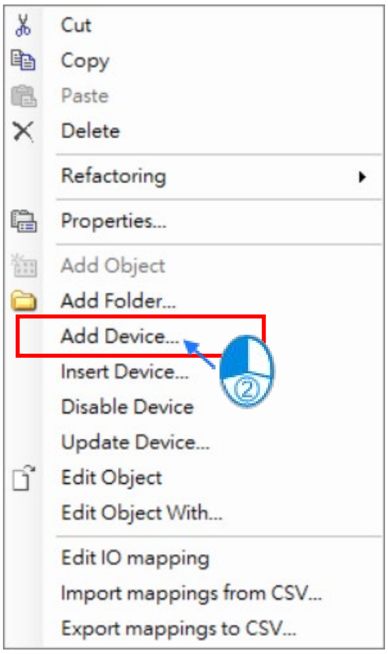
In addition to providing the standard Modbus communication protocol, the AX-5 Series PLC further executes the Delta controller internal device conversion (X, M, D devices, etc.), no need to check the conversion table. AX-5 Series PLC can act as a Modbus TCP Master, after you have created Modbus TCP Master and Modbus TCP Slave. Follow the below section to set up the Modbus TCP Master.



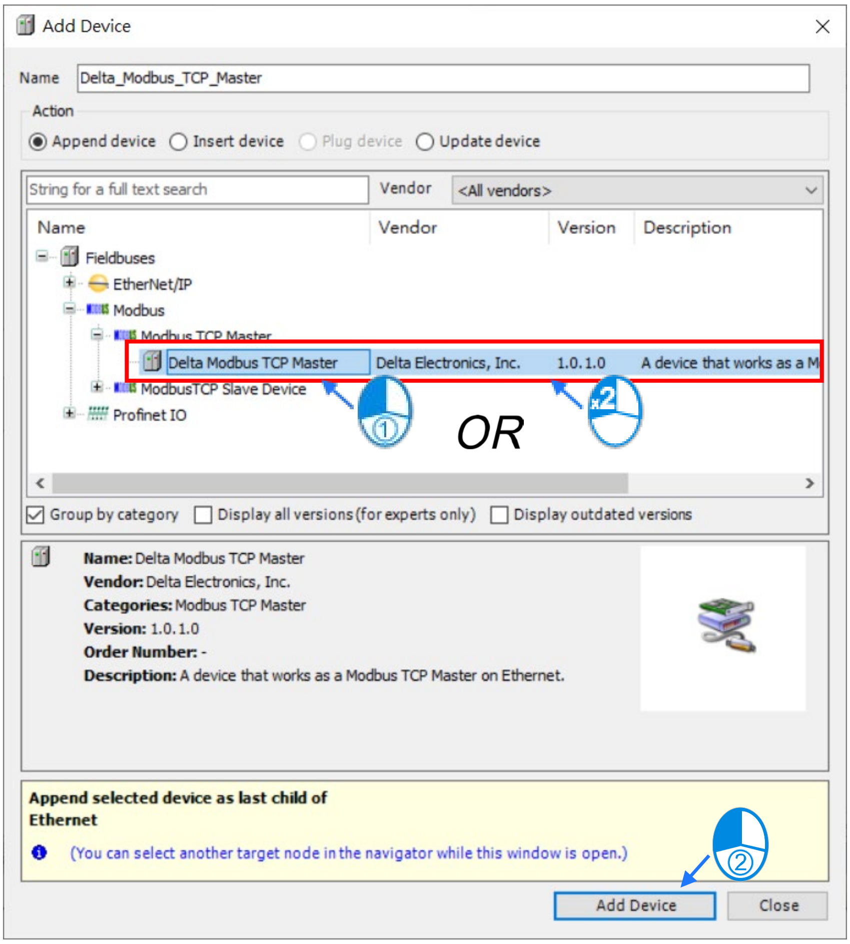
8.3.3.1 Adding a Modbus TCP Master/Slave

1. Right-click the **Ethernet (Ethernet)** node in the tree view to open a context menu. And click **Add Device…**to open the Add Device setting page.





2. Find and double-click **Delta Modbus TCP Master** (Fieldbuses -> Modbus -> Modbus TCP Master -> Delta Modbus TCP Master) or click **Add Device** to add this port in. After that you can find **Delta_Modbus_TCP_Master** in the tree view.



Name: Delta_Modbus_TCP_Master

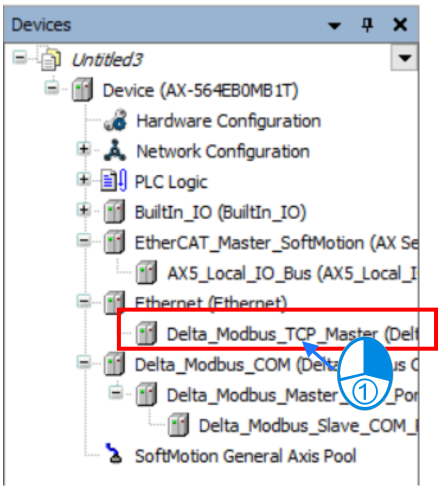
Action: ● Append device ○ Insert device ○ Plug device ○ Update device

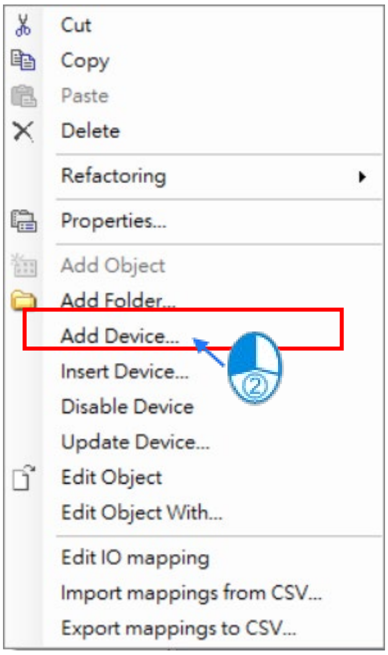
Name: Delta Modbus TCP Master
Vendor: Delta Electronics, Inc.
Categories: Modbus TCP Master
Version: 1.0.1.0
Order Number: -
Description: A device that works as a Modbus TCP Master on Ethernet.

Append selected device as last child of Ethernet

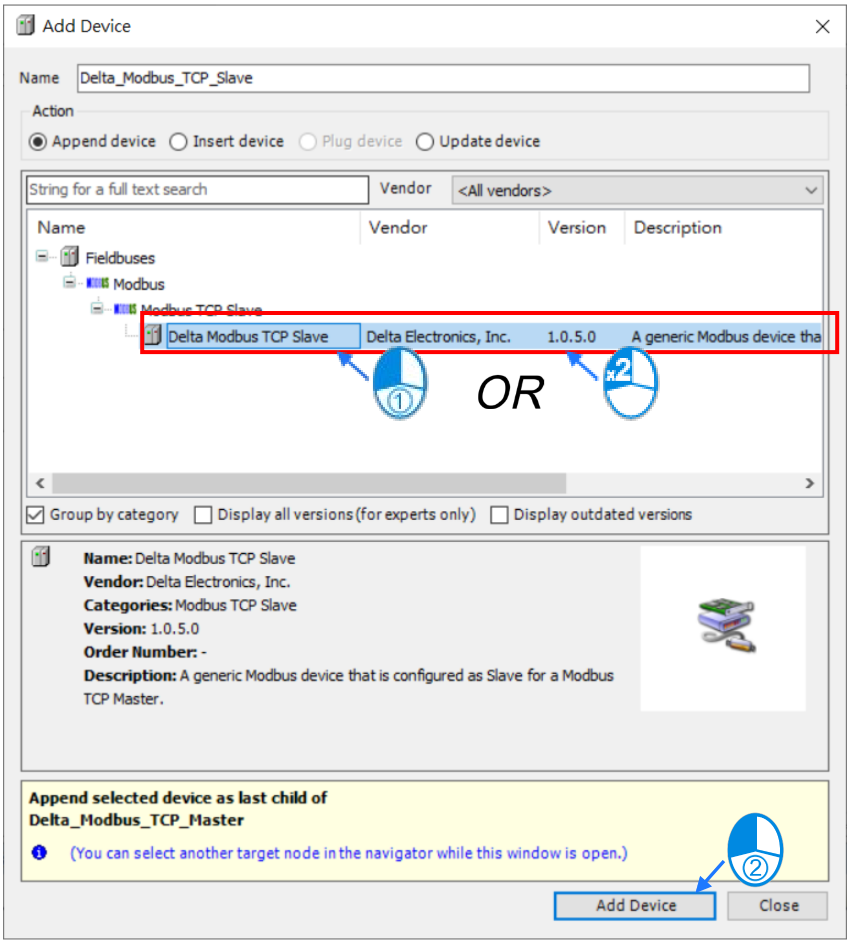
(You can select another target node in the navigator while this window is open.)

3. Right-click **Delta_Modbus_TCP_Master** under the **Ethernet** node in the tree view to open a context menu. And click **Add Device…**to open the Add Device setting page.





4. Find and double-click **Delta Modbus TCP Slave** (Fieldbuses -> Modbus -> Modbus TCP Slave -> Delta Modbus TCP Slave) or click **Add Device** to add it in.



Name: Delta_Modbus_TCP_Slave

Name	Vendor	Version	Description
Delta Modbus TCP Slave	Delta Electronics, Inc.	1.0.5.0	A generic Modbus device tha

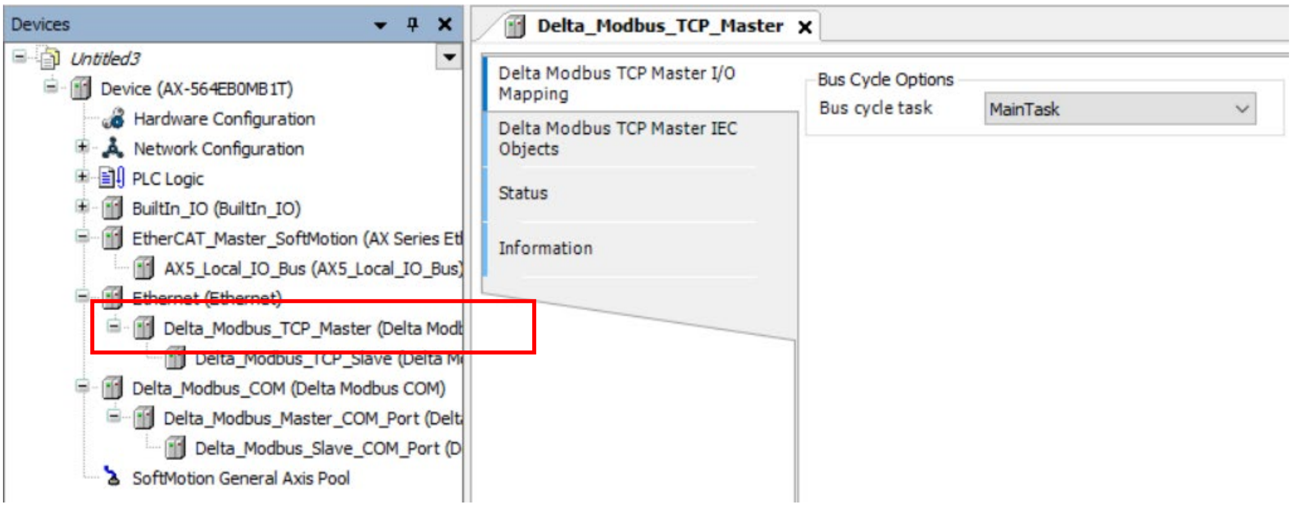
Name: Delta Modbus TCP Slave
Vendor: Delta Electronics, Inc.
Categories: Modbus TCP Slave
Version: 1.0.5.0
Order Number: -
Description: A generic Modbus device that is configured as Slave for a Modbus TCP Master.

Append selected device as last child of
Delta_Modbus_TCP_Master

(You can select another target node in the navigator while this window is open.)

After that you can find **Delta_Modbus_TCP_Slave** under the **Delta_Modbus_TCP_Master** node in the tree view.

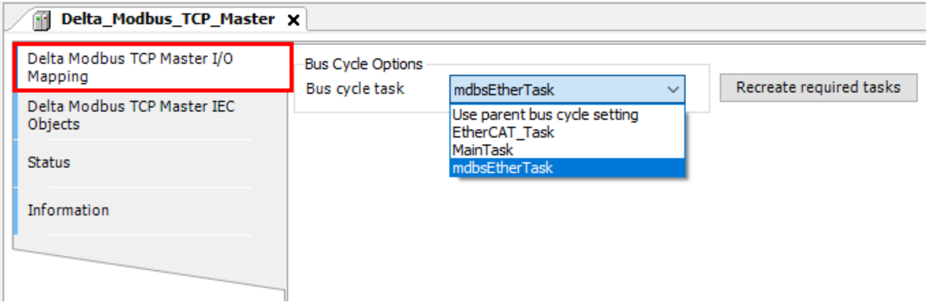
8.3.3.2 Setting up the Modbus TCP Master



Bus cycle task: MainTask

- **Delta Modbus TCP Master I/O Mapping**

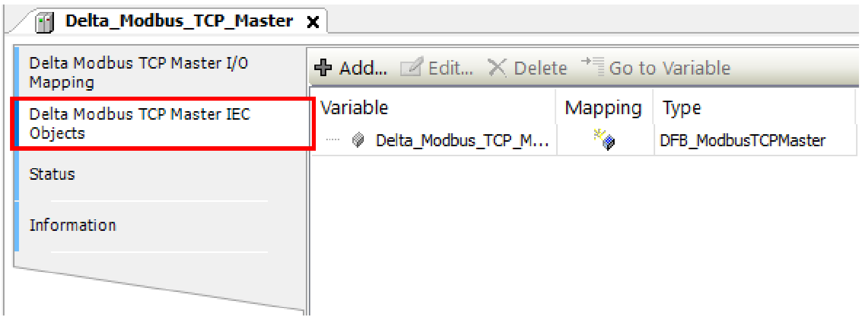
  Bus cycle task: Select a bus cycle task to synchronize with the Modbus communication time. When the option "Use parent bus cycle setting is selected", the system uses the shortest cycle time as the bus cycle time. Refer to 4.2.1.6 section "PLC Settings" for more information.



Bus cycle task: mdbsEtherTask

- **Delta Modbus TCP Master IEC Objects**

You can check the status of Modbus TCP Master under this tab.



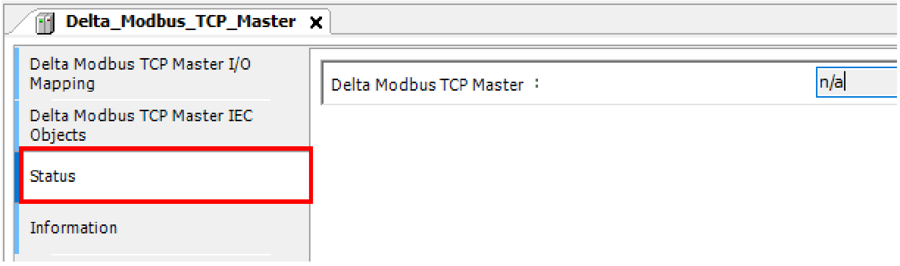
Variable	Mapping	Type
Delta_Modbus_TCP_M...		DFB_ModbusTCPMaster

  - bStop: TRUE => Stop sending Modbus TCP packets.
  - bSlaveError: TRUE => connection/communication with the Slave is abnormal
  - uiConnectedSlaves: the number of the connected Slaves

  EX: (ST programming language): Delta Modbus TCP Master.bStop:= TRUE;

- **Status**

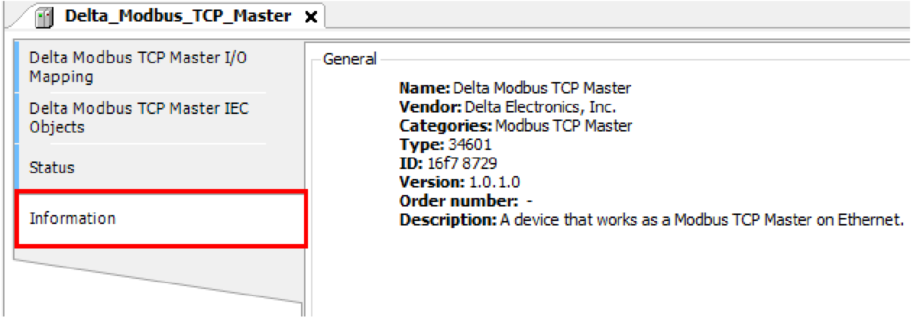
  Here you can find the device status information, for example 'Running' or 'Stopped', and specific diagnostic messages from the respective device, also information about the card used and the internal bus system.



Delta Modbus TCP Master : n/a

- **Information**

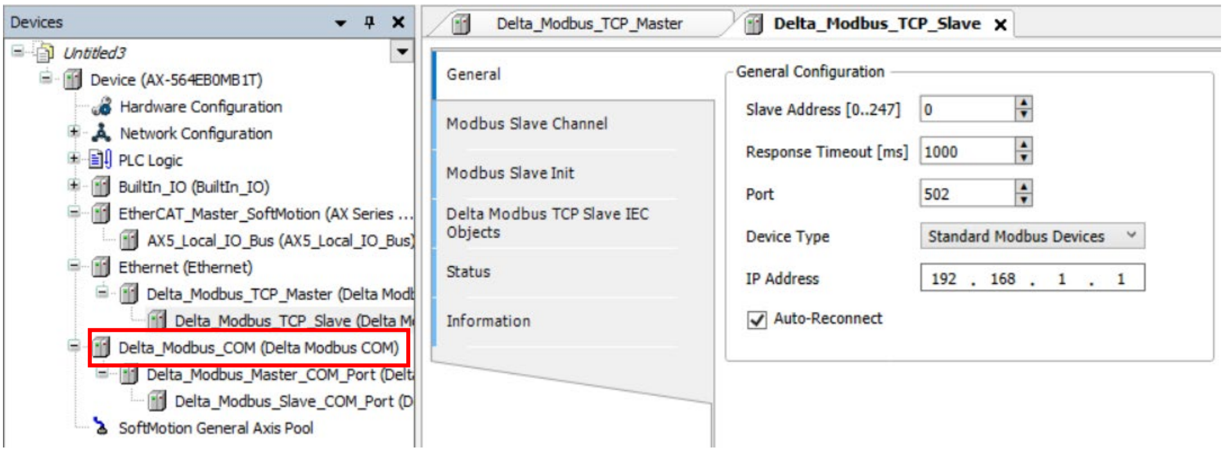
  Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



Name: Delta Modbus TCP Master
Vendor: Delta Electronics, Inc.
Categories: Modbus TCP Master
Type: 34601
ID: 16f7 8729
Version: 1.0.1.0
Order number: -
Description: A device that works as a Modbus TCP Master on Ethernet.

8.3.3.3 Setting up the Modbus TCP Slave

1. In the tree view, find the **Delta_Modbus_TCP_Slave (Delta Modbus TCP Slave)** and double-click it to open the setting page to set up.



General Configuration

Slave Address [0..247]: 0

Response Timeout [ms]: 1000

Port: 502

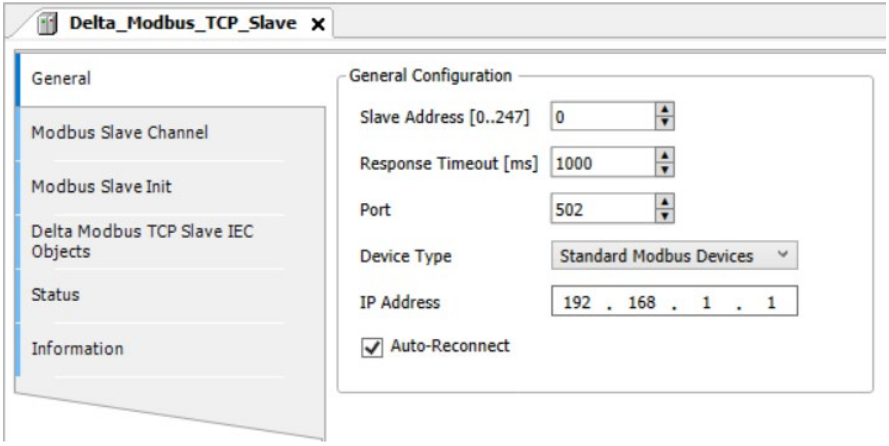
Device Type: Standard Modbus Devices

IP Address: 192 . 168 . 1 . 1

☑ Auto-Reconnect

- **General**

Here you can configure the basic settings for Modbus TCP Slave, such as Slave Address, Response Timeout and Device Type.



General Configuration

Slave Address [0..247]: 0

Response Timeout [ms]: 1000

Port: 502

Device Type: Standard Modbus Devices

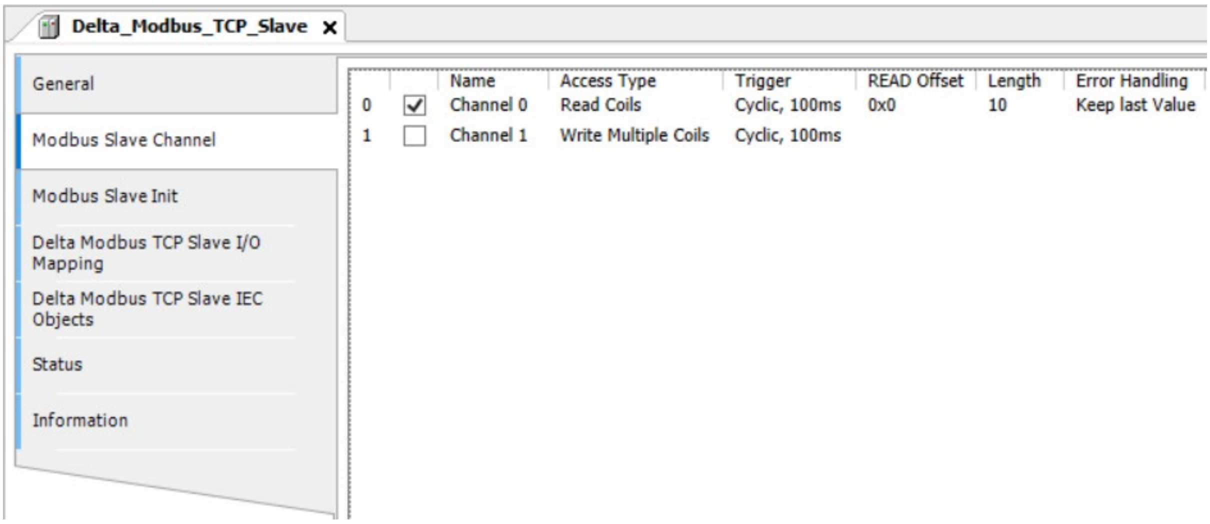
IP Address: 192 . 168 . 1 . 1

☑ Auto-Reconnect

Item	Description
Slave Address	Address of a serial Modbus device
Response Timeout	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
Device Type	You can select standard Modbus devices or Delta devices. If you select Delta devices, the system converts the protocol used in Modbus protocol automatically so that you do NOT need to refer to the register map for the conversion.
IP Address	Slave IP address
Auto-Reconnect	Enable this option to have this port reconnect automatically if an error or connection timeout occurs.

- **Modbus Slave Channel**

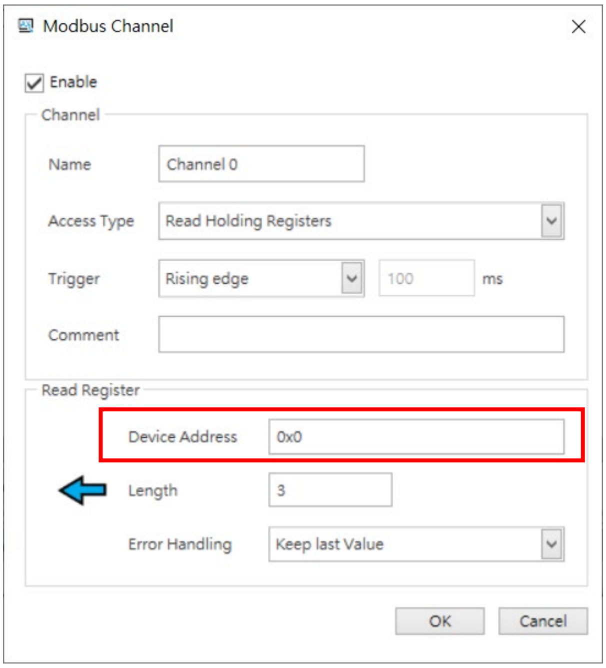
Here you can define slave channels. Each channel represents a single Modbus request. You can create up to 100 channels for each slave. AX-5 Series PLC will send out Modbus request packets in chronological order. All channels share the same one Modbus TCP connection.



		Name	Access Type	Trigger	READ Offset	Length	Error Handling
0	☑	Channel 0	Read Coils	Cyclic, 100ms	0x0	10	Keep last Value
1	☐	Channel 1	Write Multiple Coils	Cyclic, 100ms			

Click **Add Channel** to open the setting page; you can edit the channel before adding it in. The **Device Address** shows the Modbus protocol address you selected whether the device type is **Standard Modbus Device** or **Delta Devices** (D register, M coil, X coil, etc.) under the **General** tap. Since the system converts the protocol used into Modbus protocol automatically, you do NOT need to refer to the register map for the conversion.

Device Type：Standard Modbus Device



☑ Enable

Name: Channel 0

Access Type: Read Holding Registers

Trigger: Rising edge 100 ms

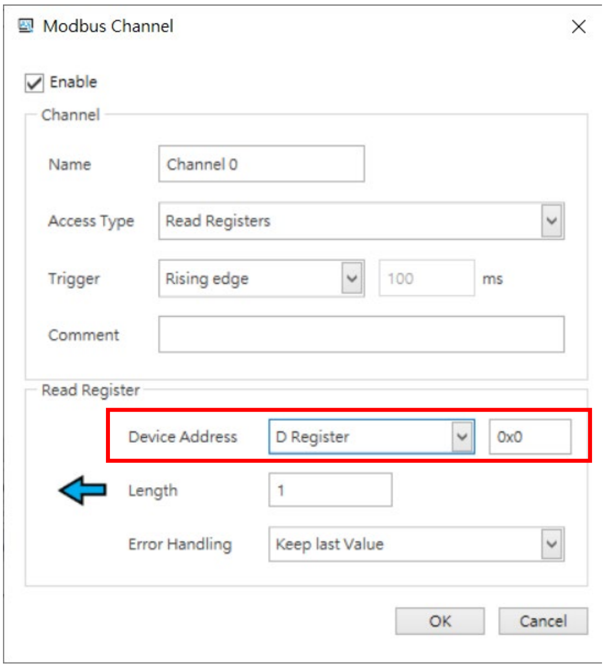
Comment:

Device Address: 0x0

Length: 3

Error Handling: Keep last Value

Device Type：AH Series



☑ Enable

Name: Channel 0

Access Type: Read Registers

Trigger: Rising edge 100 ms

Comment:

Device Address: D Register 0x0

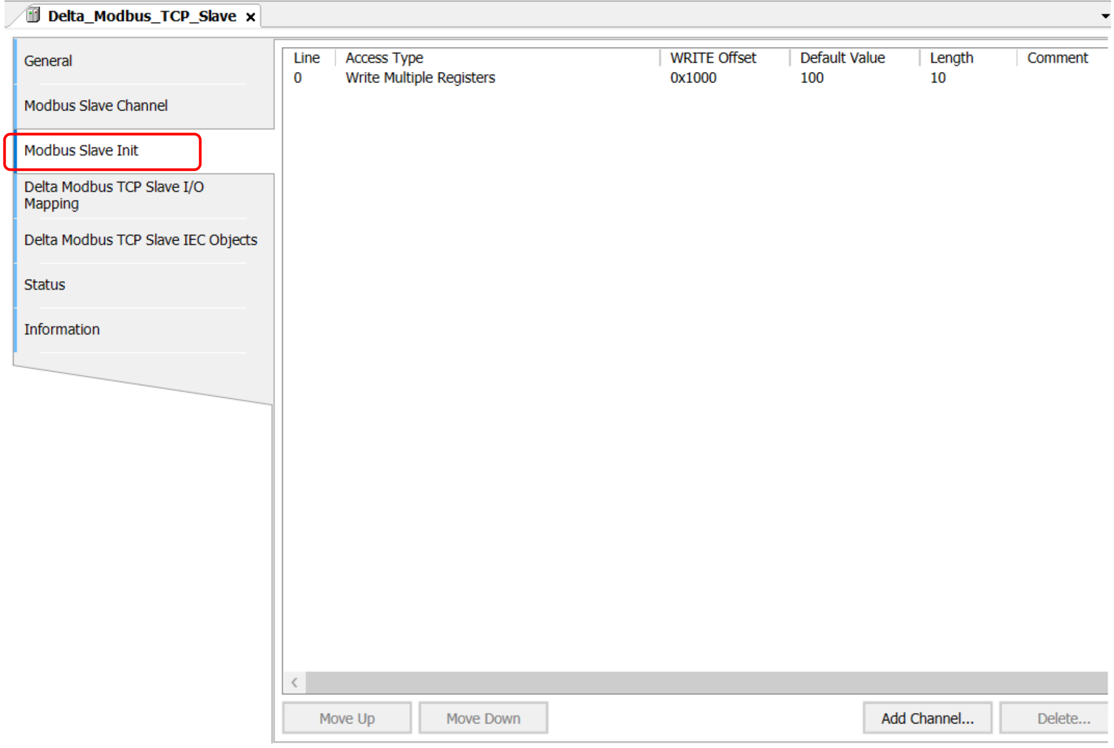
Length: 1

Error Handling: Keep last Value

Item	Description	
Device Type	**Standard Modbus Device**	**Delta Series Device**
Enable	Activates this channel	
Name	Defines this channel name	
Access Type	Modbus 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)	Read/Write Registers ● Read coils ● Read registers ● Write coils ● Write registers Note: PLC uses the corresponding Modbus function code according to the read/write register of the device type.
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 **DFB_ModbusTCPChannel**	● **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 **DFB_ModbusTCPChannel**
Comment	Description of the channel	
Device Address	Modbus protocol address	Delta register address (will be converted into Modbus protocl in the background)
Length	Number of the register to be read/written to.	Number of the register to be read/written to. (up to 256 coils and 100 registers)
Error Handling	What to do with the data in case of a communication error: ● Set To ZERO ● Keep last value	

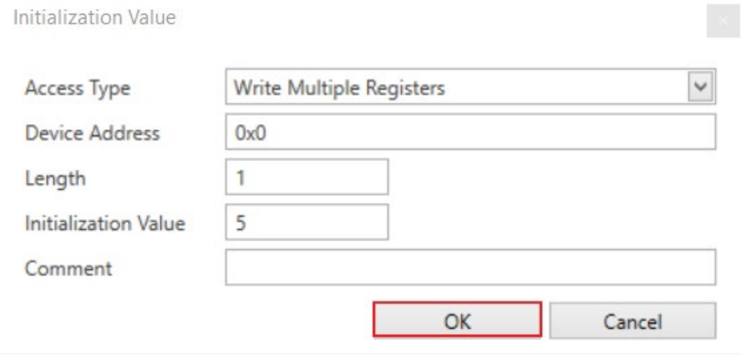
- **Modbus Slave Init**

After the Modbus connection between AX-5 Series PLC and the slaves is established, you can use **Add Channel** button to edit the Initialization Value of the Coil/Register.



Line	Access Type	WRITE Offset	Default Value	Length	Comment
0	Write Multiple Registers	0x1000	100	10	

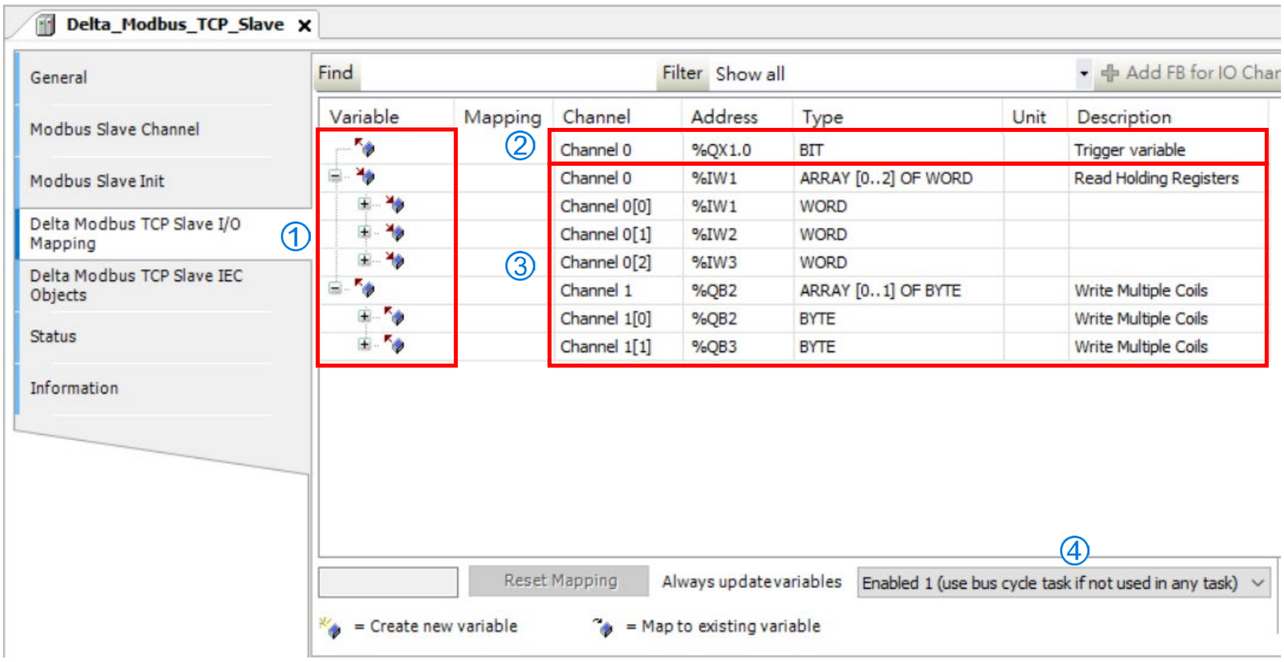
Click **Add Channel**, you can edit the Access Type, Device Address, Length, Initialization Value and Comment. Click OK to confirm the settings.



Access Type: Write Multiple Registers
Device Address: 0x0
Length: 1
Initialization Value: 5
Comment:

Delta Modbus TCP Slave I/O Mapping

After you have added channels under the tab of Modbus TCP Slave Channel, you can find the variables and the set access types under this tab. Here you can define the variables for mapping. The descriptions here reflect what you have set for the **Access Type** in Modbus Slave Channel tab. When the **Trigger type** is set to **Rising edge** in Modbus Slave Channel, the description here adds one more condition, **Trigger variable**.



Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%QX1.0	BIT		Trigger variable
		Channel 0	%IW1	ARRAY [0..2] OF WORD		Read Holding Registers
		Channel 0[0]	%IW1	WORD		
		Channel 0[1]	%IW2	WORD		
		Channel 0[2]	%IW3	WORD		
		Channel 1	%QB2	ARRAY [0..1] OF BYTE		Write Multiple Coils
		Channel 1[0]	%QB2	BYTE		Write Multiple Coils
		Channel 1[1]	%QB3	BYTE		Write Multiple Coils

Always update variables: Enabled 1 (use bus cycle task if not used in any task)

① The descriptions here reflect what you have set for the **Access Type** in Modbus Slave Channel tab.

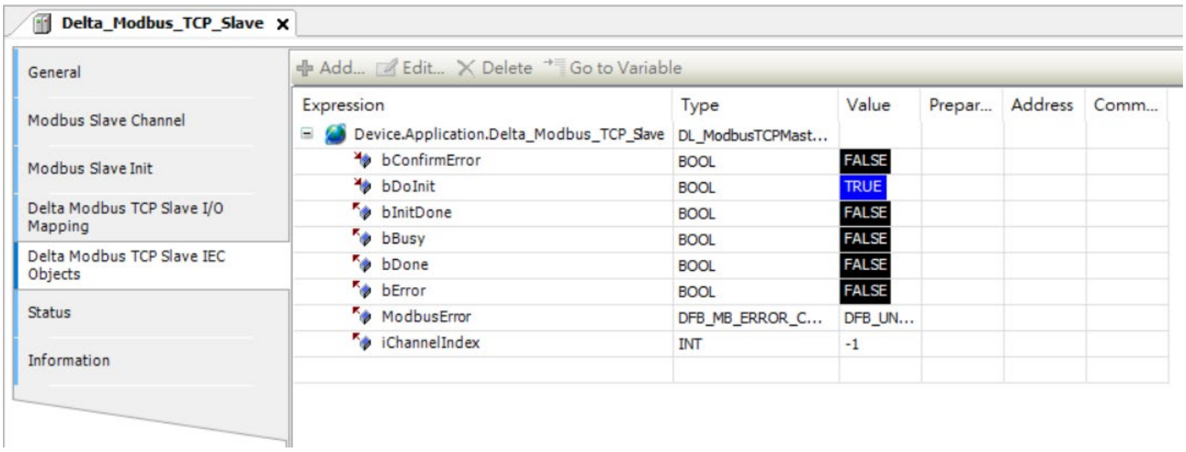
② The triggered Boolean variable for this channel.

③ The controller registers that are read/written by this channel.

④ Timing for the data refreshing; refer to section 4.2.1.6 PLC Settings for more information.

Delta Modbus TCP Slave IEC Objects

You can check the status of Modbus TCP Slave under this tab.

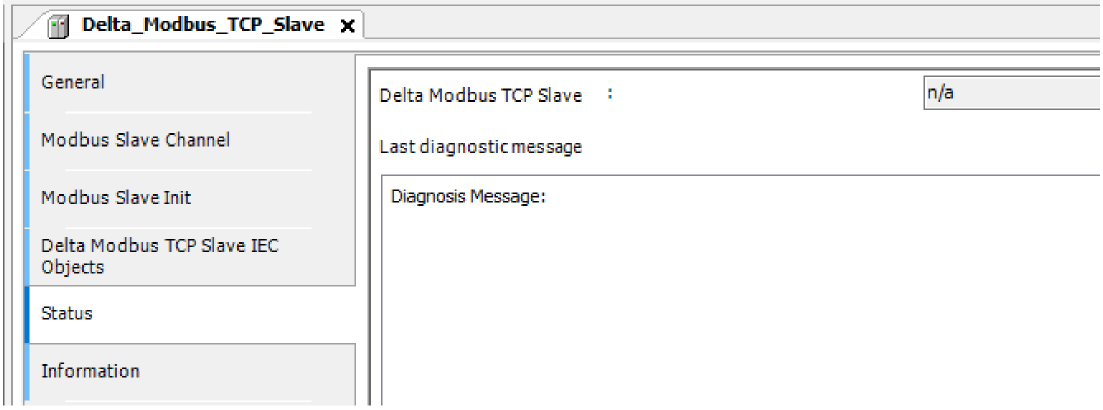


Expression	Type	Value	Prepar...	Address	Comm...
Device.Application.Delta_Modbus_TCP_Slave	DL_ModbusTCPMast...				
bConfirmError	BOOL	FALSE			
bDoInit	BOOL	TRUE			
bInitDone	BOOL	FALSE			
bBusy	BOOL	FALSE			
bDone	BOOL	FALSE			
bError	BOOL	FALSE			
ModbusError	DFB_MB_ERROR_C...	DFB_UN...			
iChannelIndex	INT	-1			

Item	Description
bConfirmError	If the option “Auto-Reconnect” is NOT enabled, during the data transmission, any channel that showed error stops. After the bConfirmError shows “TRUE”, the channel that showed error previously continues to execute.
bDoInit	Initialized the Slave
bInitDone	The initialization of the Slave is complete.
bBusy	This channel is executing data exchange.
bDone	The data exchange via this channel is complete.
bError	Error occurs when this channel is executing data exchange.
ModbusError	Record Modbus errors occurred.
iChannelIndex	The channel number that is in execution.

- **Status**

Here you can find the Modbus TCP Slave status information, for example ‘Running’ or ‘Stopped’, and specific diagnostic messages from the respective device, also information about the card used and the internal bus system.



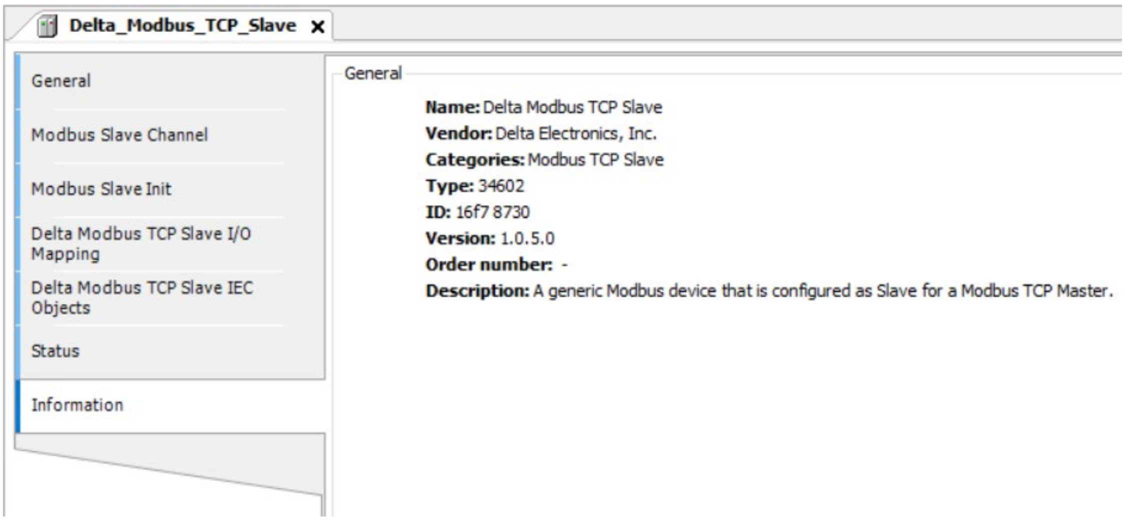
Delta Modbus TCP Slave : n/a

Last diagnostic message

Diagnosis Message:

- **Information**

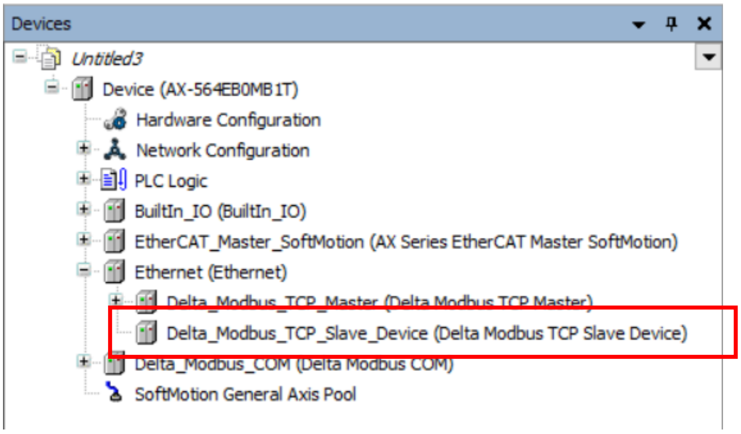
Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



Name: Delta Modbus TCP Slave
Vendor: Delta Electronics, Inc.
Categories: Modbus TCP Slave
Type: 34602
ID: 16f7 8730
Version: 1.0.5.0
Order number: -
Description: A generic Modbus device that is configured as Slave for a Modbus TCP Master.

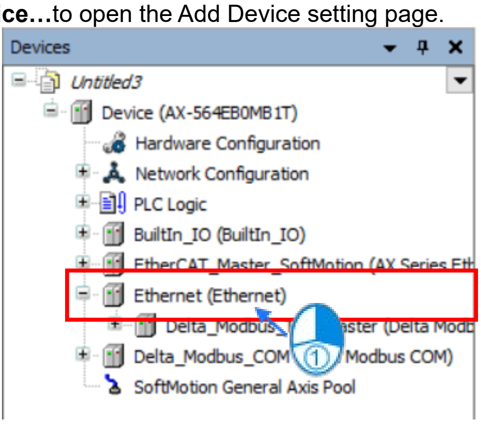
8.3.4 Modbus TCP Slave (Server)

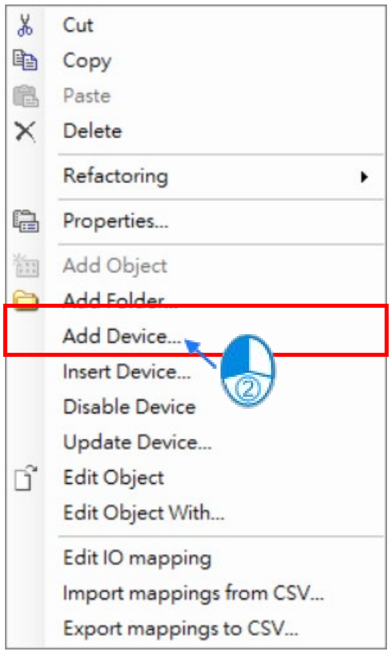
AX-5 Series PLC can act as a Modbus TCP Slave, after you add Modbus TCP Slave Device in and set up the allowable areas for Coils/Register. If Modbus TCP Master uses Delta device communication protocol, there is no access restrictions. Follow the below section to set up the Modbus TCP Slave.



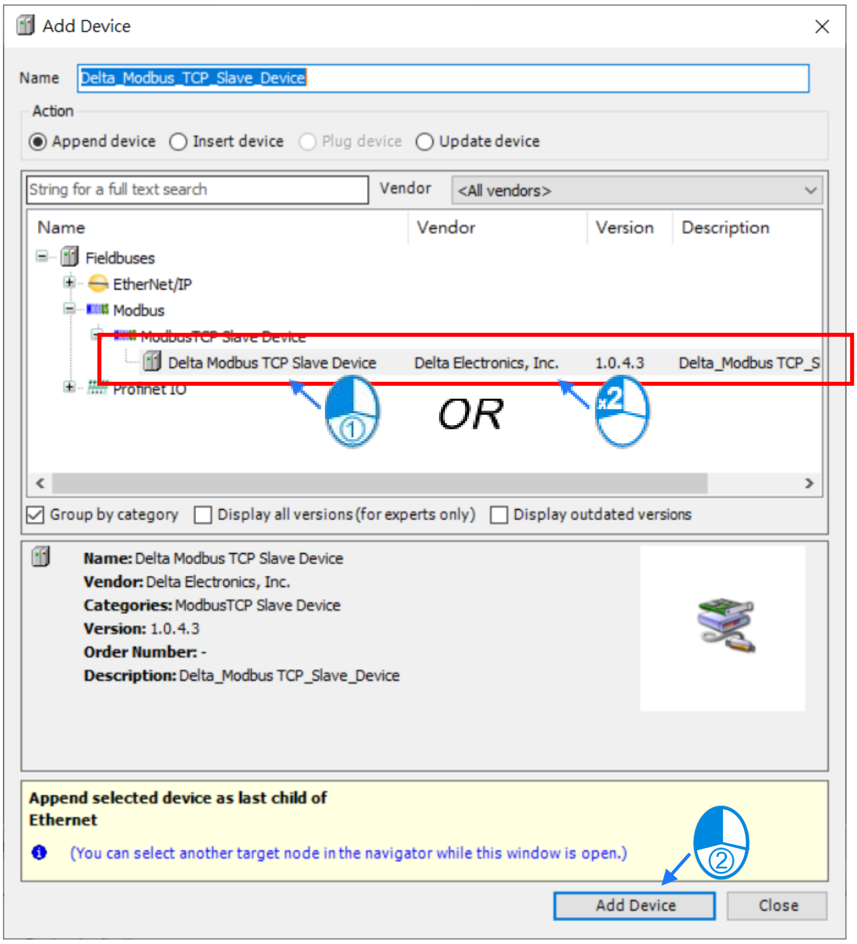
8.3.4.1 Adding a Modbus TCP Slave Device

1. Right-click the **Ethernet (Ethernet)** node in the tree view to open up a context menu. And click **Add Device…**to open the Add Device setting page.





2. Find and double-click **Delta Modbus TCP Slave Device** (Fieldbuses -> Modbus -> Modbus TCP Slave Device -> Delta Modbus TCP Slave Device) or click **Add Device** to add this port in.



Name: Delta_Modbus_TCP_Slave_Device

Action: ● Append device ○ Insert device ○ Plug device ○ Update device

String for a full text search

Vendor <All vendors>

Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
Modbus			
ModbusTCP Slave Device			
Delta Modbus TCP Slave Device	Delta Electronics, Inc.	1.0.4.3	Delta_Modbus TCP_S
Profinet IO			

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: Delta Modbus TCP Slave Device
Vendor: Delta Electronics, Inc.
Categories: ModbusTCP Slave Device
Version: 1.0.4.3
Order Number: -
Description: Delta_Modbus TCP_Slave_Device

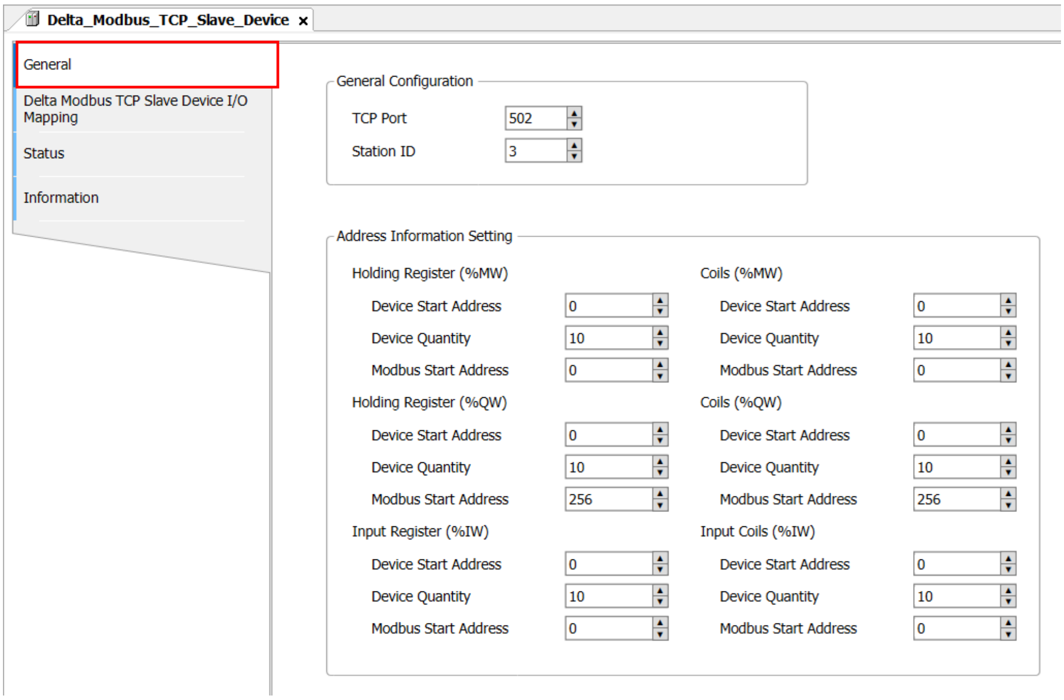
Append selected device as last child of Ethernet

(You can select another target node in the navigator while this window is open.)

8.3.4.2 Setting up the Modbus TCP Slave Device

- **General**

  Here you can configure the basic settings for Modbus TCP Slave Device. Set up the allowable areas for Coils/Register. If Modbus TCP Slave uses Delta device communication protocol, there is no access restrictions.



General Configuration

TCP Port 502

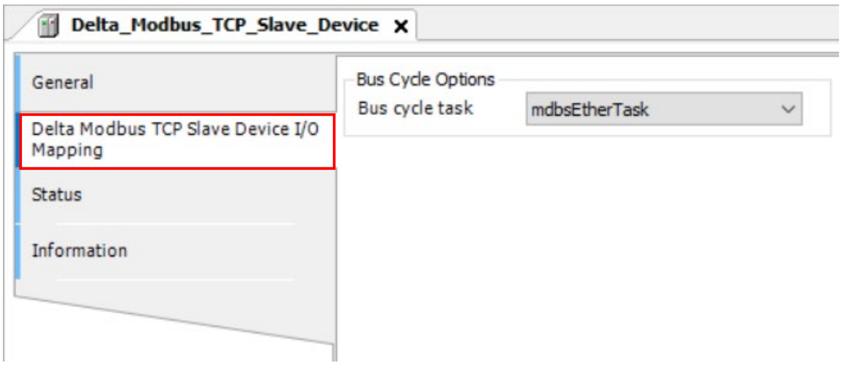
Station ID 3

Address Information Setting

Holding Register (%MW)		**Coils (%MW)**	
Device Start Address	0	Device Start Address	0
Device Quantity	10	Device Quantity	10
Modbus Start Address	0	Modbus Start Address	0
Holding Register (%QW)		**Coils (%QW)**	
Device Start Address	0	Device Start Address	0
Device Quantity	10	Device Quantity	10
Modbus Start Address	256	Modbus Start Address	256
Input Register (%IW)		**Input Coils (%IW)**	
Device Start Address	0	Device Start Address	0
Device Quantity	10	Device Quantity	10
Modbus Start Address	0	Modbus Start Address	0

- **Delta Modbus TCP Slave Device I/O Mapping**

  Bus cycle task: Select a bus cycle task to synchronize with the Modbus communication time. When the option "Use parent bus cycle setting is selected", the system use the shortest cycle time as the bus cycle time. Refer to section 4.2.1.6 PLC Settings for more information.

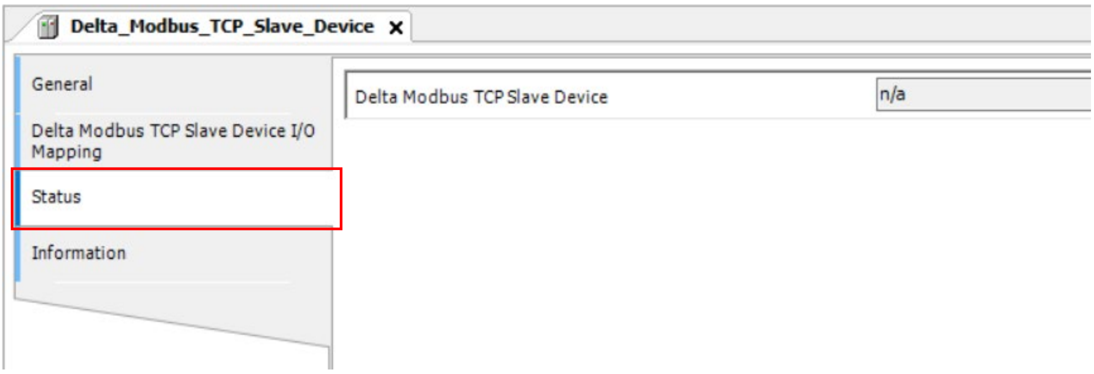


Bus Cycle Options

Bus cycle task: mdbsEtherTask

- **Status**

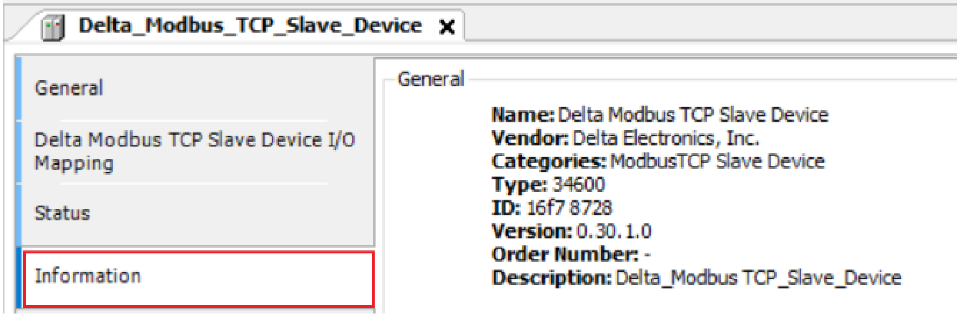
  Here you can find the Modbus TCP Slave Device status information, for example ‘Running’ or ‘Stopped’, and specific diagnostic messages from the respective device, also information about the card used and the internal bus system.



Delta Modbus TCP Slave Device n/a

- **Information**

  Here you can find general information that originates from the device description file: name, vendor, categories, version, order number, description, and other relevant information.



Name: Delta Modbus TCP Slave Device
Vendor: Delta Electronics, Inc.
Categories: ModbusTCP Slave Device
Type: 34600
ID: 16f7 8728
Version: 0.30.1.0
Order Number: -
Description: Delta_Modbus TCP_Slave_Device

8.4 EtherNet/IP

DIADesigner-AX supports the following Modbus network types, including Modbus TCP and EtherNet/IP. Follow the below section to set up the basic settings for communication through EtherNet/IP.

8.4.1 Introduction on EtherNet/IP

Ethernet Industrial Protocol (EtherNet/IP) is an open industrial networking standard, managed by ODVA (Open DeviceNet Vendors Association).

EtherNet/IP works on a TCP/UDP/IP based Ethernet network and uses most widely deployed collections of Ethernet standards to provide a broad range of applications in different industries that require high-speed and stability including Factory Automation (FA), Building Automation (BA), Process Automation (PA) and many more.

Delta covers a full range of controller and drive products supported by EtherNet/IP, including Programmable Logic Controllers (PLC), inverters, Human Machine Interfaces (HMI) and so on. Refer to section 8.4.1.4 for a full product list supported by EtherNet/IP. In addition, users can also use the EDS file to connect to the EtherNet/IP devices of other brands.

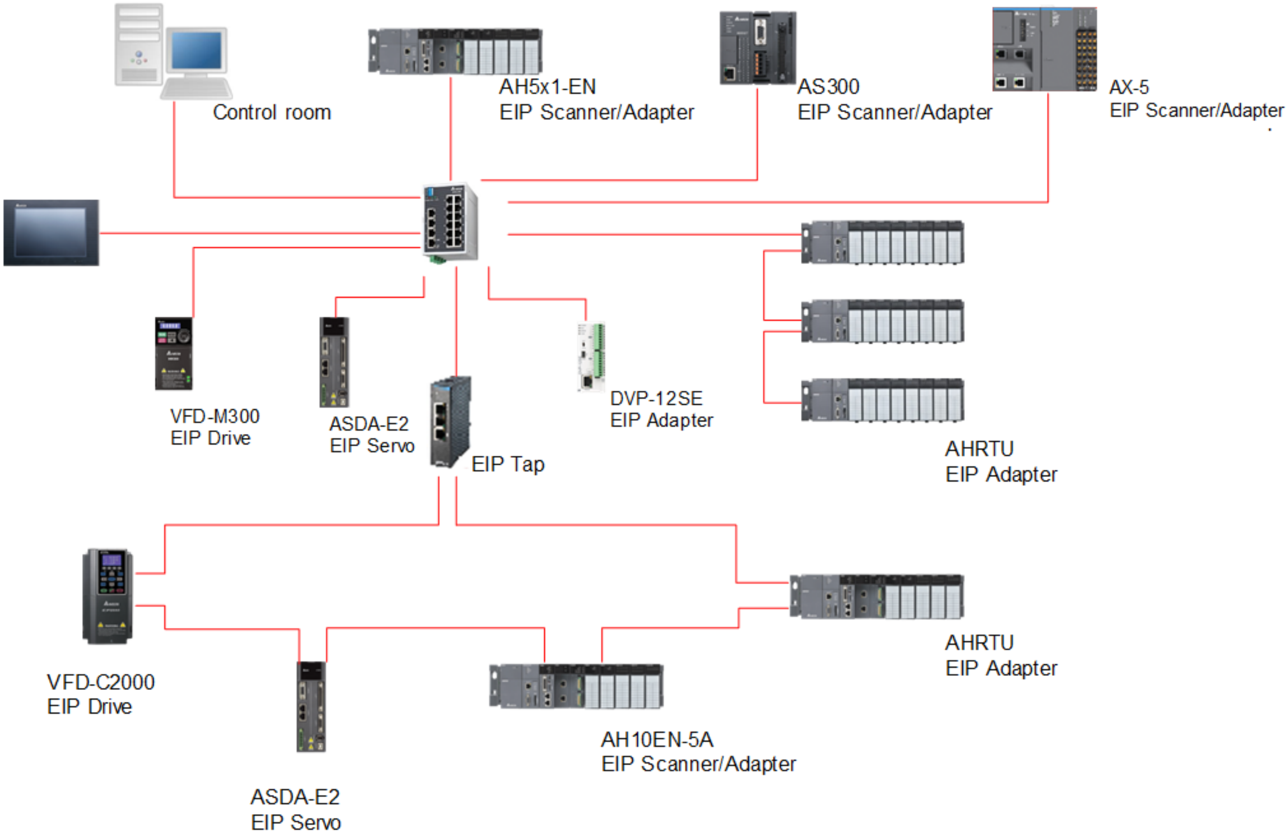
8.4.1.1 Definition

Term	Definition
ODVA	Open DeviceNet Vendor Association for EtherNet/IP
EIP	EtherNet/IP, an industrial Ethernet network, provides interoperability for system providers. IP stands for Industrial Protocol. The term “EIP” (EtherNet/IP) will be used throughout this manual.
I/O Connection	Via the I/O connection to connect to EtherNet/IP and to exchange data cyclically
Explicit Message	Connect to EtherNet/IP and to exchange data non-cyclically. Data will be exchanged piece by piece via instructions.
RPI	Requested Packet Interval, via the I/O connection to connect to EtherNet/IP to exchange data at regular time intervals
ACD	Address Conflict Detection to detect IP address duplications.
P/C TAG	Produced / Consumed TAG. A produced TAG sends its data to consumed TAGs (consumers) without using logic. TAGs are the methods used for assigning and referencing memory locations for Rockwell PLCs, the same as the registers for Delta PLCs.
EDS	Electronic Data Sheets; EDS files are simple text files used by EtherNet/IP network configuration tools to help you identify EtherNet/IP products and easily commission them on a network.
Data Mapping	Exchange data between devices.
EIP Scanner	The master station is called Scanner in EtherNet/IP.
EIP Adapter	The slave station is called Adapter in EtherNet/IP.
Modbus TCP	Modbus TCP is a Modbus communication protocol, widely used on Ethernet.

8.4.1.2 Features of Ethernet

8.4.1.2.1 Delta EIP Architecture

This typical Delta EIP architecture includes EIP Scanner and Adapter; data mapping can be achieved between devices via an I/O connection and explicit message.



8.4.1.2.2 Features of EIP

- Flexibility
  - Flexible topology: EIP devices may include an Ethernet single port as well as Ethernet dual port, and provide applicable networks such as linear topology, ring topology and ring topology for faster expansion and easier management.
  - Network compatible: IT specialists are not required for Internet connection setup, while the Wi-Fi connection is provided.

- Simplicity
  - Via a connector: Delta provides a full range of product line, including human machine interfaces (HMI), programmable logic controllers (PLC) and inverter drives, for application in an industrial operation. Simply via a RJ-45 connector, a network can be built up, saving costs on cables and other connecting tools.
  - Single network: In replace with the 3-tier industrial architecture, single network architecture provides 100Mbps high-speed cyclical and non-cyclical data mapping function, ensuring a complete network diagnosis, and effectively shortening debugging time.

8.4.1.2.3 Delta EIP Product List

- **Delta EIP Products, Adapter supported**

Positioning	Product	Firmware Version
Mid-range PLC	AHCPU501-EN, AHCPU511-EN, AHCPU521-EN, AHCPU531-EN	V2.00
	AHCPU560-EN2	V1.00
	AH10EN-5A	V2.00
	AHRTU-ETHN-5A	V1.00
	AH10EMC-5A	V1.00
	AS300 Series	V1.00
	AS200 Series	V1.00
	AS100 Series	V1.10
	AS-FEN02 communication card (V1.06) for AS300 Series (V1.00)	V1.06
	AS-FEN02 communication card (V2.02) for AS00SCM-A Series (V1.00)	V2.02
	AX-3 Series	V1.01
	AX-5 Series	V1.00
Small PLC	DVP-ES2-E Series	V3.60
	DVP26SE Series	V1.00
	DVP-ES3 Series	V1.00
Inverter	CMM-EIP01/02 Communication Card for VFD-MS300 Series	V1.00
	CMM-EIP03 Communication Card for VFD-MS300 Series	V1.00
	CMC-EIP01 Communication Card for VFD-C2000 Series	V1.06
	CMC-EIP02 Communication Card for VFD-C2000 Series	V1.00

- **Delta EIP Products, Scanner supported**

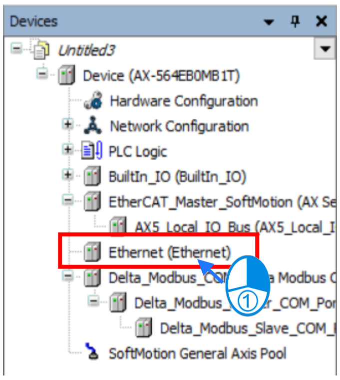
Positioning	Product	Firmware Version
Mid-range PLC	AHCPU501-EN, AHCPU511-EN, AHCPU521-EN, AHCPU531-EN	V2.00
	AHCPU560-EN2	V1.00
	AH10EN-5A	V2.00
	AS300 Series, AS200 Series	V1.00
	AS100 Series	V1.10
	AX-3 Series	V1.01
	AX-5 Series	V1.00
Small PLC	DVP-ES3 Series	V1.00

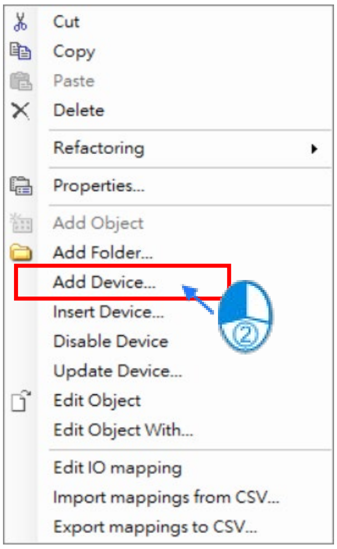
8.4.2 EtherNet/IP Scanner Function

Before utilizing the AX-5 Series PLC as an EtherNet/IP Scanner (master), you need to add Modbus TCP Master first and then Modbus TCP Slave. After completing the configurations, AX-5 Series PLC can act as an EtherNet/IP Scanner. The following steps demonstrate how to use AX-5 as an EtherNet/IP Scanner. If you cannot find any EtherNet/IP Adapter to add, you will need to import the EDS file of the EtherNet/IP Adapter.

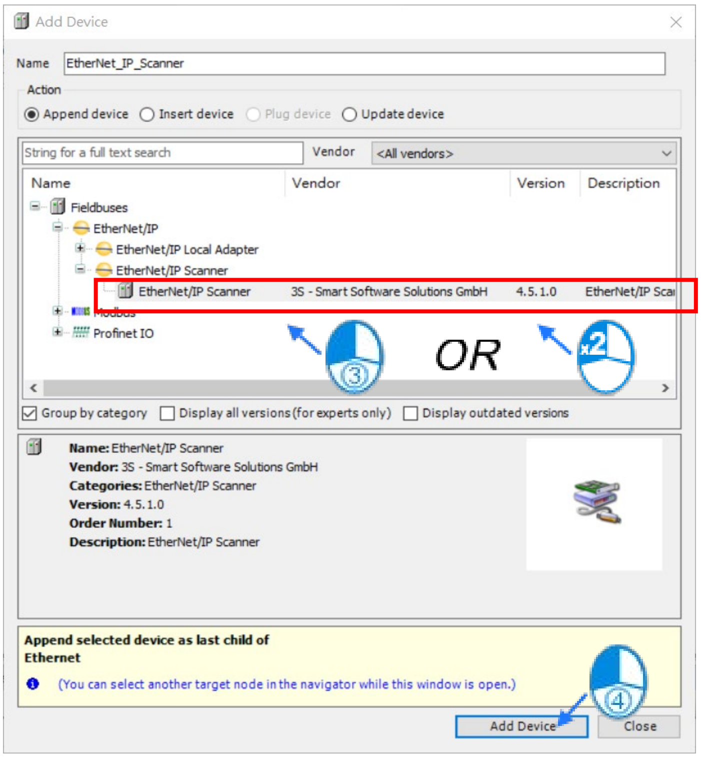
8.4.2.1 Adding EtherNet/IP Scanner and EtherNet/IP Adapter

1. Right-click the **Ethernet (Ethernet)** in the tree view to open a context menu. And click **Add Device…**to open the setting page.





2. Find and double-click **EtherNet/IP Scanner** (Fieldbuses -> EtherNet/IP -> EtherNet/IP Scanner -> EtherNet/IP Scanner) or click **Add Device** to add this device in.


Name EtherNet_IP_Scanner

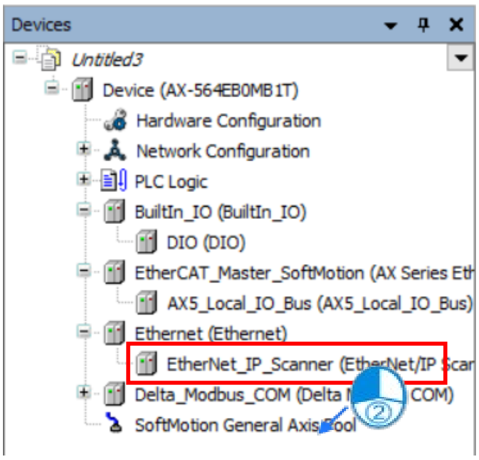
Action: ● Append device ○ Insert device ○ Plug device ○ Update device

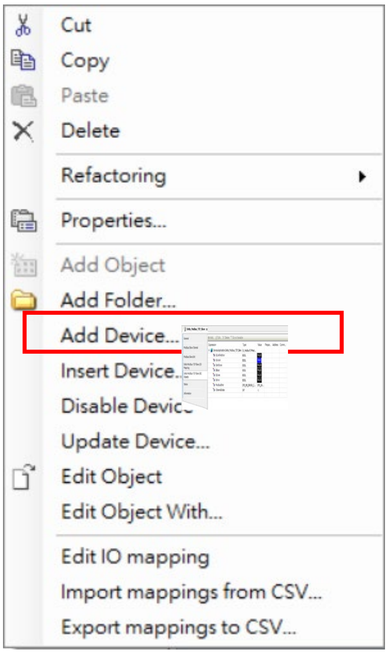
Name: EtherNet/IP Scanner
Vendor: 3S - Smart Software Solutions GmbH
Categories: EtherNet/IP Scanner
Version: 4.5.1.0
Order Number: 1
Description: EtherNet/IP Scanner

Append selected device as last child of
Ethernet

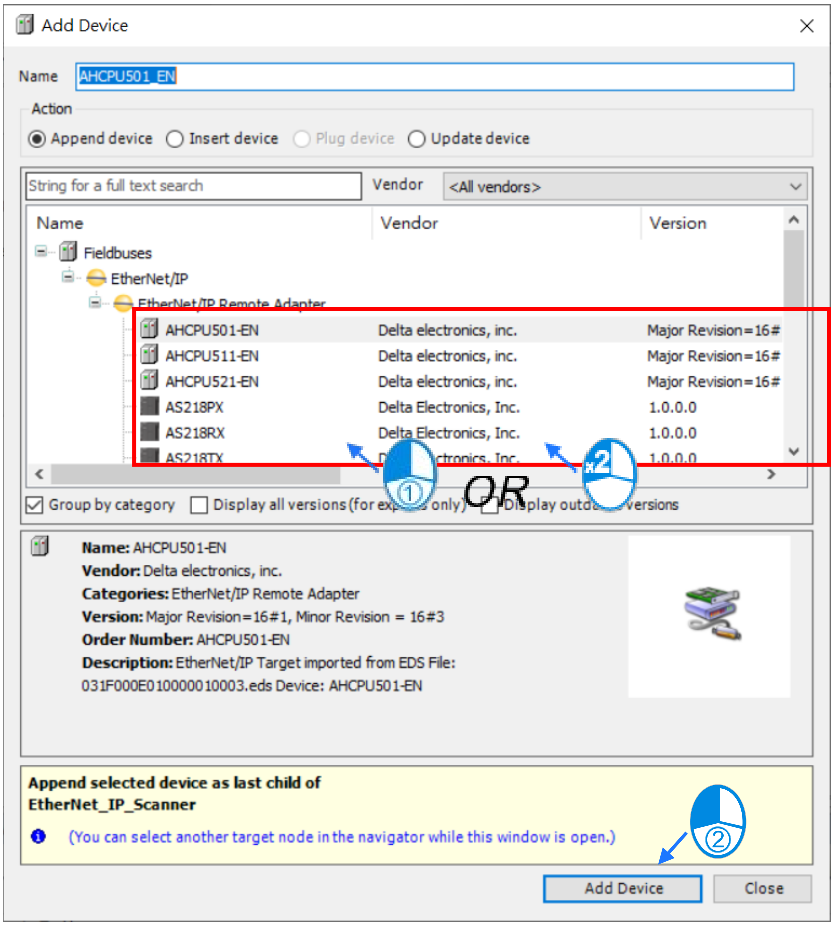
(You can select another target node in the navigator while this window is open.)

3. To add an EtherNet/IP Adapter: Right-click the **Ethernet_IP_Scanner (EtherNet/IP Scanner)** in the tree view to open a context menu. And click **Add Device…**to open the setting page.





4. Find and double-click the model under **EtherNet/IP Adapter** (Fieldbuses -> EtherNet/IP -> EtherNet/IP Remote Adapter -> any model you'd like to add as an EtherNet/IP Adapter and click **Add Device** to add this device in.



Name: AHCPU501_EN

● Append device ○ Insert device ○ Plug device ○ Update device

Name	Vendor	Version
Fieldbuses		
EtherNet/IP		
EtherNet/IP Remote Adapter		
AHCPU501-EN	Delta electronics, inc.	Major Revision=16#
AHCPU511-EN	Delta electronics, inc.	Major Revision=16#
AHCPU521-EN	Delta electronics, inc.	Major Revision=16#
AS218PX	Delta Electronics, Inc.	1.0.0.0
AS218RX	Delta Electronics, Inc.	1.0.0.0
AS218TX	Delta Electronics, Inc.	1.0.0.0

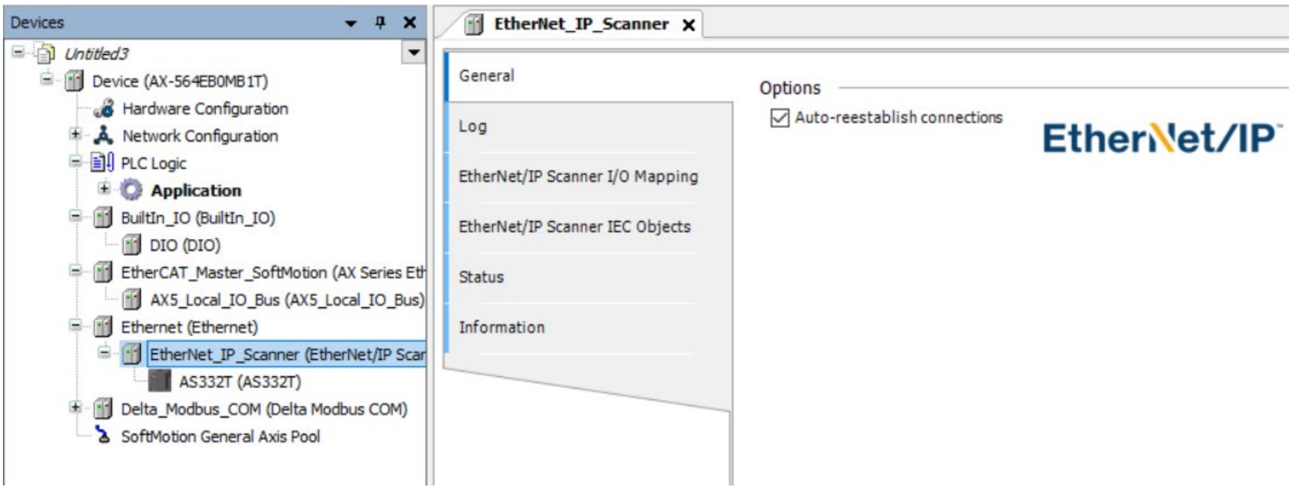
☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: AHCPU501-EN
Vendor: Delta electronics, inc.
Categories: EtherNet/IP Remote Adapter
Version: Major Revision=16#1, Minor Revision = 16#3
Order Number: AHCPU501-EN
Description: EtherNet/IP Target imported from EDS File: 031F000E010000010003.eds Device: AHCPU501-EN

Append selected device as last child of EtherNet_IP_Scanner

(You can select another target node in the navigator while this window is open.)

8.4.2.2 Setting up EtherNet/IP Scanner

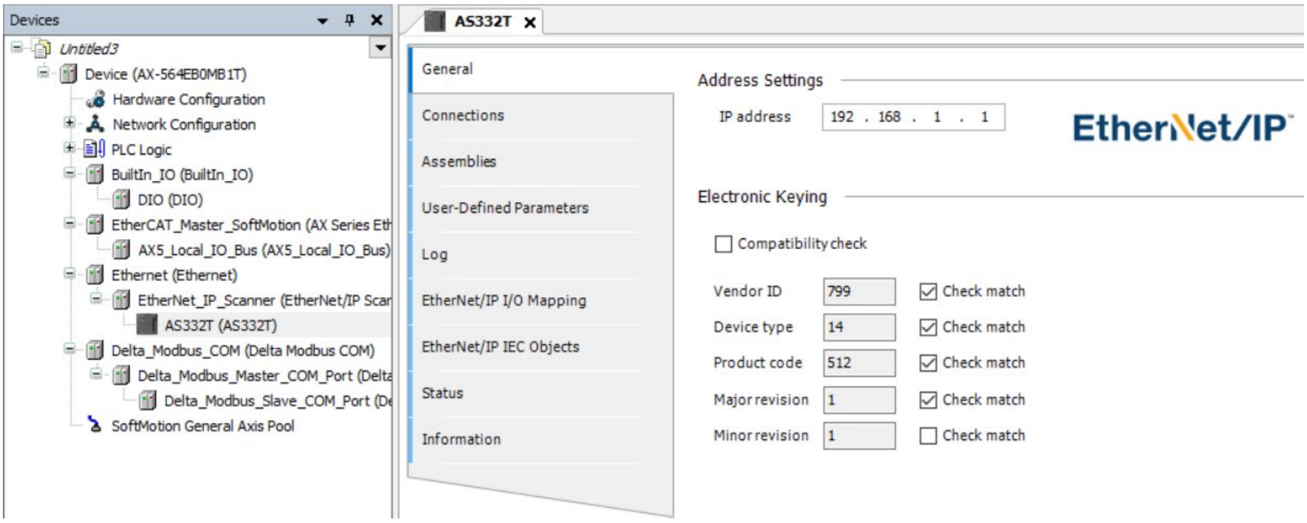


☑ Auto-reestablish connections

Tab	Description
General	Auto-reestablish connections: Enable this option to have this connection established automatically if an error occurs or connection lost between the EtherNet/IP Adapter.
Log	Operational record of the EtherNet/IP Scanner
EtherNet/IP Scanner I/O Mapping	You can set up the Bus Cycle Task for the EtherNet/IP Scanner; Refer to section 4.2.1.6 PLC Settings for more information.
EtherNet/IP Scanner IEC Objects	xReset: ON: Re-establish all connections (right-edge triggered) eState: State of EtherNet/IP Scanner eError: Error code
Status	Current running status of EtherNet/IP Scanner
Information	Here you can find general information of the current EtherNet/IP Scanner, including vendor, firmware version and other relevant information.

8.4.2.3 Setting up EtherNet/IP Adapter

This section will explain how to set up an EtherNet/IP adapter (using AS332T). Refer to the followings for configuration.



Address Settings

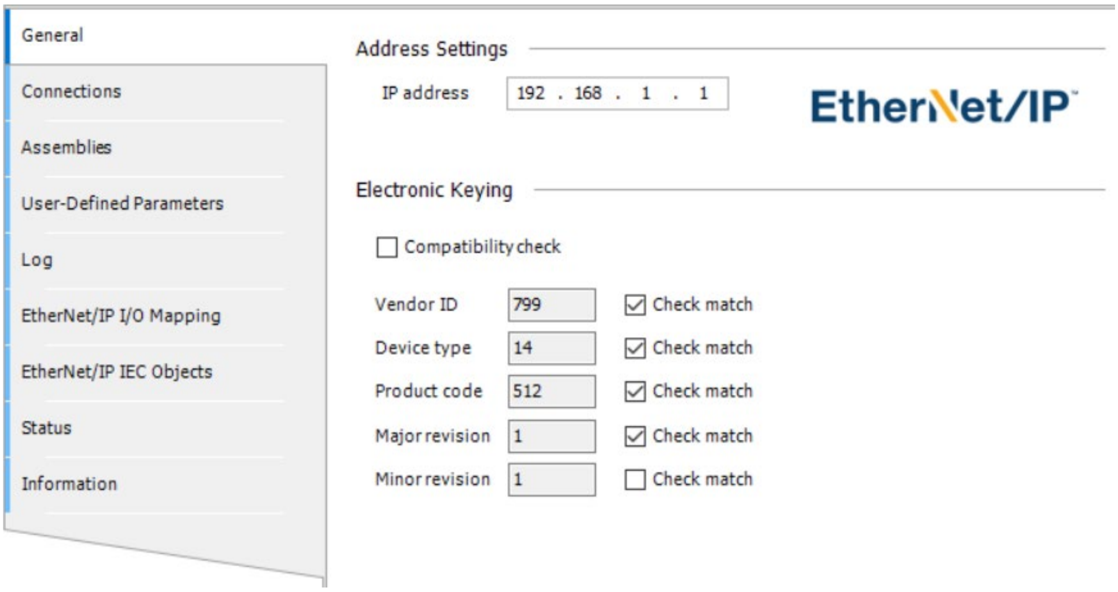
IP address 192 . 168 . 1 . 1

Electronic Keying

☐ Compatibility check

Vendor ID	799	☑ Check match
Device type	14	☑ Check match
Product code	512	☑ Check match
Major revision	1	☑ Check match
Minor revision	1	☐ Check match

- General: You can set up the IP address as a condition to check if the EtherNet/IP Adapter to be connected is the right one. If the IP address is not matched, the connection cannot be established.



Address Settings

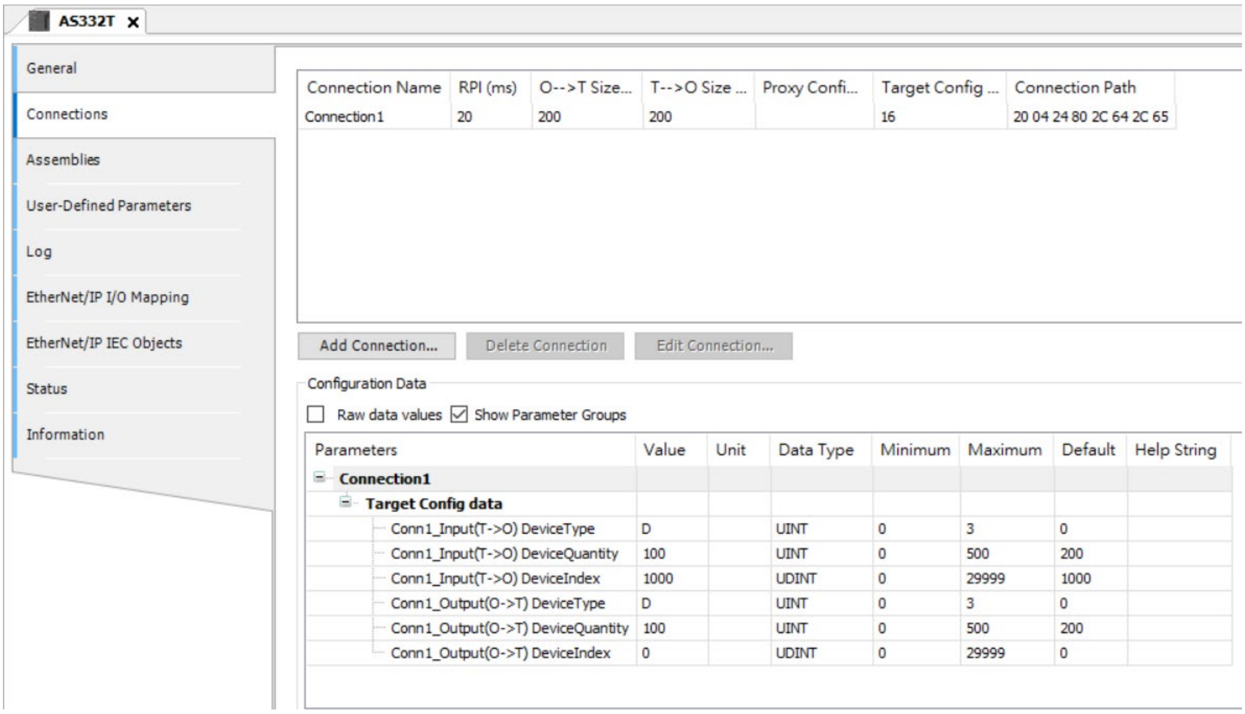
IP address 192 . 168 . 1 . 1

Electronic Keying

☐ Compatibility check

Vendor ID	799	☑ Check match
Device type	14	☑ Check match
Product code	512	☑ Check match
Major revision	1	☑ Check match
Minor revision	1	☐ Check match

- Connections: This setting page is for data exchange between EtherNet/IP Scanner and EtherNet/IP Adapter. Here you can find the settings for connection names, PRI times, size between T -> O (Target -> Originator) or O - >T (Originator -> Target) and more.



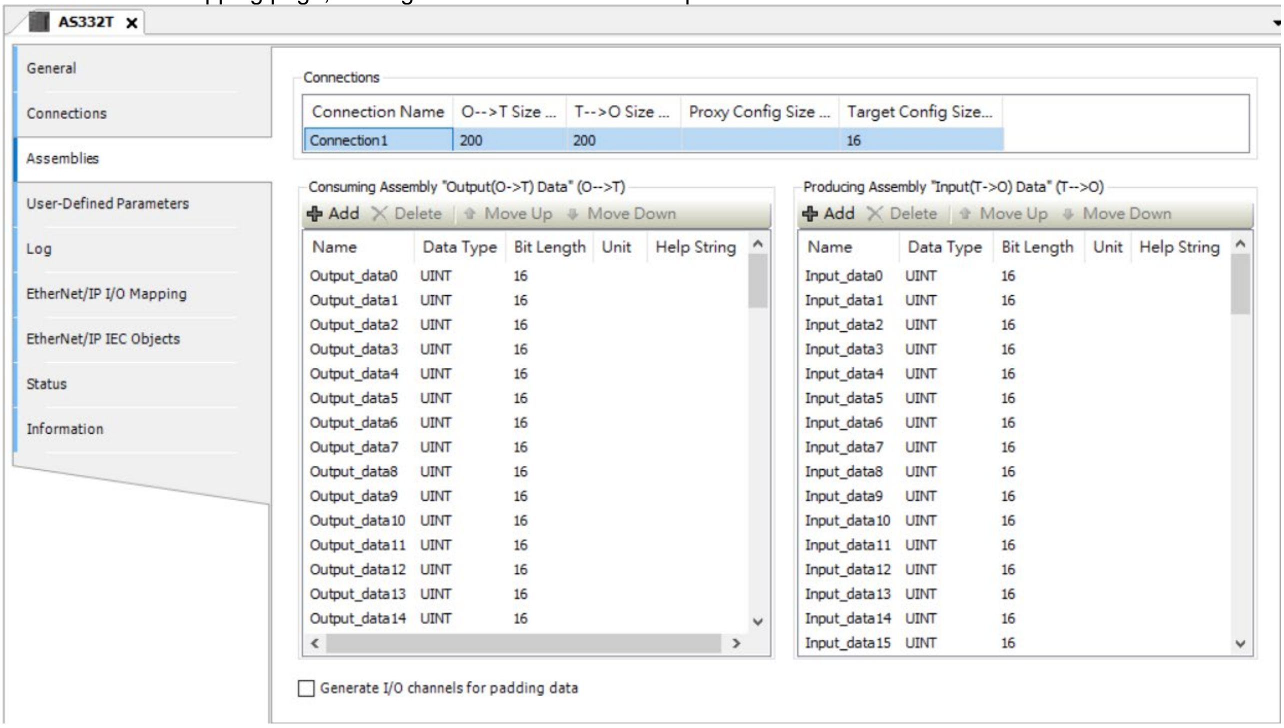
Connection Name	RPI (ms)	O-->T Size...	T-->O Size ...	Proxy Confi...	Target Config ...	Connection Path
Connection1	20	200	200		16	20 04 24 80 2C 64 2C 65

Configuration Data

☐ Raw data values ☑ Show Parameter Groups

Parameters	Value	Unit	Data Type	Minimum	Maximum	Default	Help String
Connection1							
Target Config data							
Conn1_Input(T->O) DeviceType	D		UINT	0	3	0	
Conn1_Input(T->O) DeviceQuantity	100		UINT	0	500	200	
Conn1_Input(T->O) DeviceIndex	1000		UDINT	0	29999	1000	
Conn1_Output(O->T) DeviceType	D		UINT	0	3	0	
Conn1_Output(O->T) DeviceQuantity	100		UINT	0	500	200	
Conn1_Output(O->T) DeviceIndex	0		UDINT	0	29999	0	

- Assembles: This setting page is for defining the data structure of the connection which will be synchronized on the EtherNet/IP I/O Mapping page, making it easier for users to interpret the data.



Connections

Connection Name	O-->T Size ...	T-->O Size ...	Proxy Config Size ...	Target Config Size...
Connection1	200	200		16

Consuming Assembly "Output(O->T) Data" (O-->T)

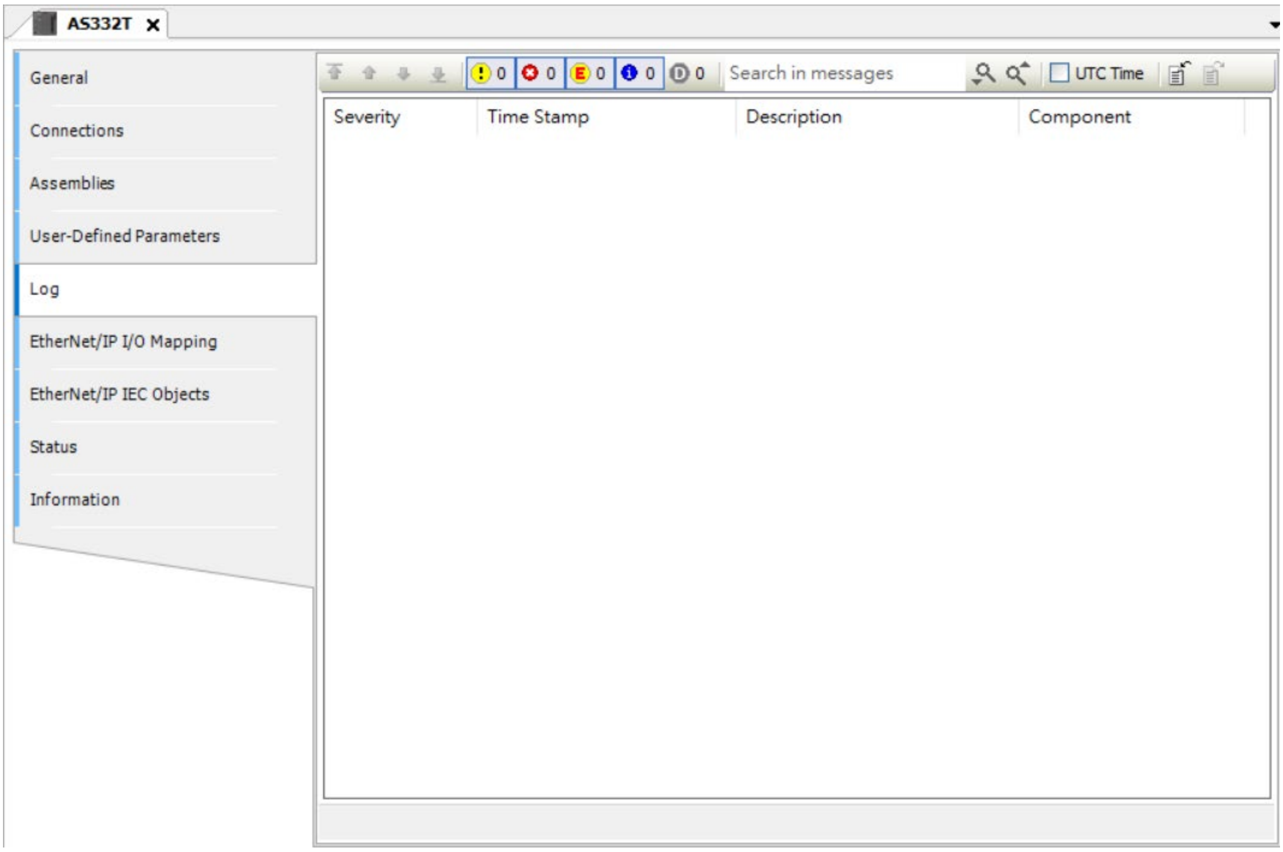
Name	Data Type	Bit Length	Unit	Help String
Output_data0	UINT	16		
Output_data1	UINT	16		
Output_data2	UINT	16		
Output_data3	UINT	16		
Output_data4	UINT	16		
Output_data5	UINT	16		
Output_data6	UINT	16		
Output_data7	UINT	16		
Output_data8	UINT	16		
Output_data9	UINT	16		
Output_data10	UINT	16		
Output_data11	UINT	16		
Output_data12	UINT	16		
Output_data13	UINT	16		
Output_data14	UINT	16		

Producing Assembly "Input(T->O) Data" (T-->O)

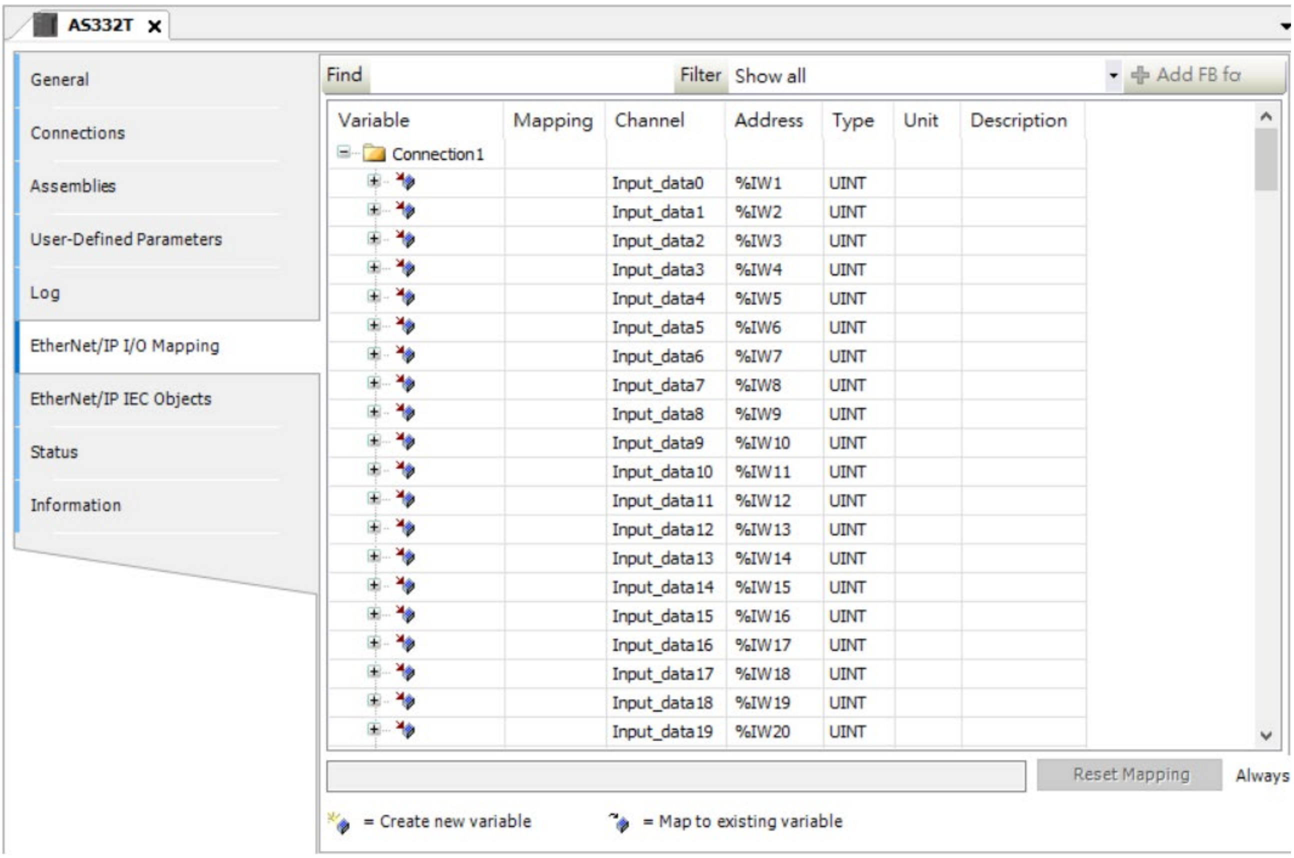
Name	Data Type	Bit Length	Unit	Help String
Input_data0	UINT	16		
Input_data1	UINT	16		
Input_data2	UINT	16		
Input_data3	UINT	16		
Input_data4	UINT	16		
Input_data5	UINT	16		
Input_data6	UINT	16		
Input_data7	UINT	16		
Input_data8	UINT	16		
Input_data9	UINT	16		
Input_data10	UINT	16		
Input_data11	UINT	16		
Input_data12	UINT	16		
Input_data13	UINT	16		
Input_data14	UINT	16		
Input_data15	UINT	16		

☐ Generate I/O channels for padding data

- Log: This page shows the operational record of the EtherNet/IP adapter.



- EtherNet/IP I/O Mapping: This page displays the corresponding memory location to the data, and the variable data type is determined by what you have defined on the Assembles setting page.

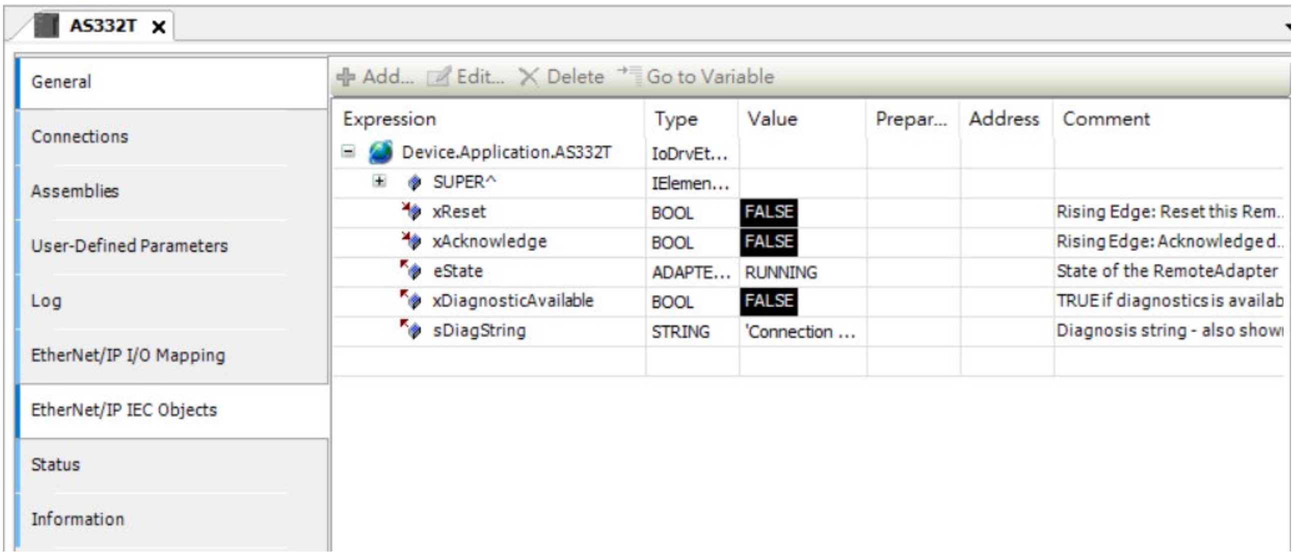


Variable	Mapping	Channel	Address	Type	Unit	Description
Connection1						
		Input_data0	%IW1	UINT		
		Input_data1	%IW2	UINT		
		Input_data2	%IW3	UINT		
		Input_data3	%IW4	UINT		
		Input_data4	%IW5	UINT		
		Input_data5	%IW6	UINT		
		Input_data6	%IW7	UINT		
		Input_data7	%IW8	UINT		
		Input_data8	%IW9	UINT		
		Input_data9	%IW10	UINT		
		Input_data10	%IW11	UINT		
		Input_data11	%IW12	UINT		
		Input_data12	%IW13	UINT		
		Input_data13	%IW14	UINT		
		Input_data14	%IW15	UINT		
		Input_data15	%IW16	UINT		
		Input_data16	%IW17	UINT		
		Input_data17	%IW18	UINT		
		Input_data18	%IW19	UINT		
		Input_data19	%IW20	UINT		

= Create new variable

= Map to existing variable

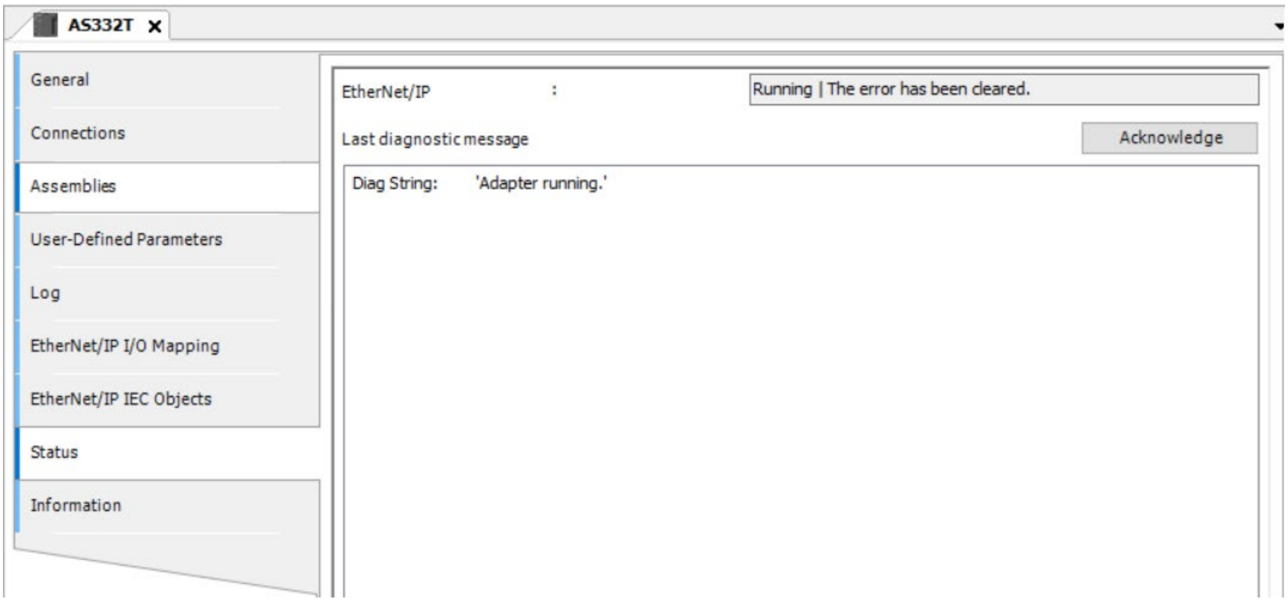
- EtherNet/IP IEC Objects：



Expression	Type	Value	Prepar...	Address	Comment
Device.Application.AS332T	IoDrvEt...				
SUPER^	IElemen...				
xReset	BOOL	FALSE			Rising Edge: Reset this Rem..
xAcknowledge	BOOL	FALSE			Rising Edge: Acknowledge d..
eState	ADAPTE...	RUNNING			State of the RemoteAdapter
xDiagnosticAvailable	BOOL	FALSE			TRUE if diagnostics is availab
sDiagString	STRING	'Connection ...			Diagnosis string - also show

Item	Description
xReset	ON: Close all the EtherNet/IP connections and re-establish them (right-edge triggered).
xAcknowledge	ON: Set the xDiagnosticAvaliable to OFF (right-edge triggered).
eState	State of EtherNet/IP Adapter
xDiagnosticAvaliable	ON: An error occurs.
sDiagString	Message of diagnosis for the EtherNet/IP Adapter; this information will also be shown under the Status tab.

- Status: Status of the EtherNet/IP Adapter

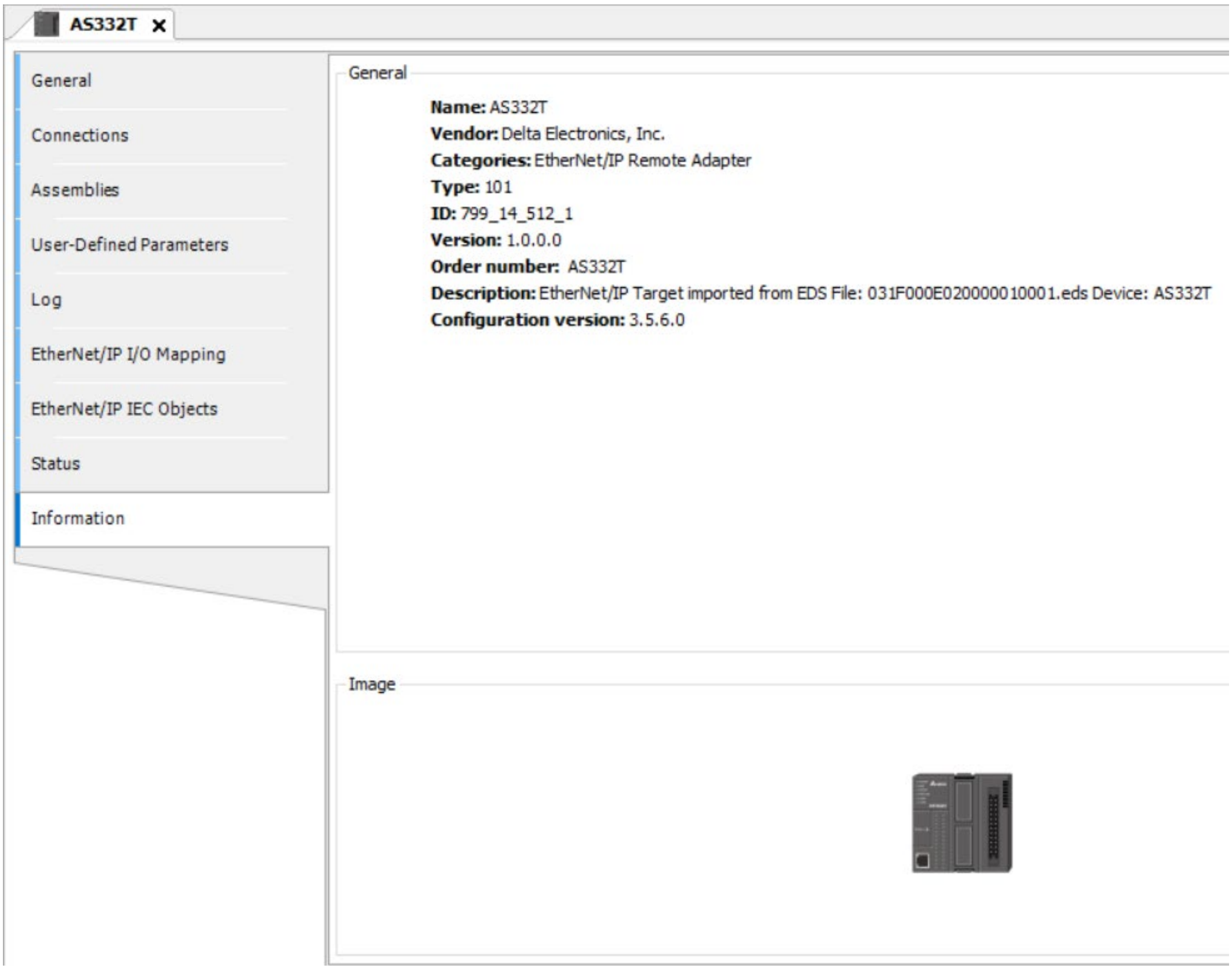


EtherNet/IP : Running | The error has been cleared.

Last diagnostic message

Diag String: 'Adapter running.'

- Information: Information of the EtherNet/IP Adapter



Name: AS332T
Vendor: Delta Electronics, Inc.
Categories: EtherNet/IP Remote Adapter
Type: 101
ID: 799_14_512_1
Version: 1.0.0.0
Order number: AS332T
Description: EtherNet/IP Target imported from EDS File: 031F000E020000010001.eds Device: AS332T
Configuration version: 3.5.6.0

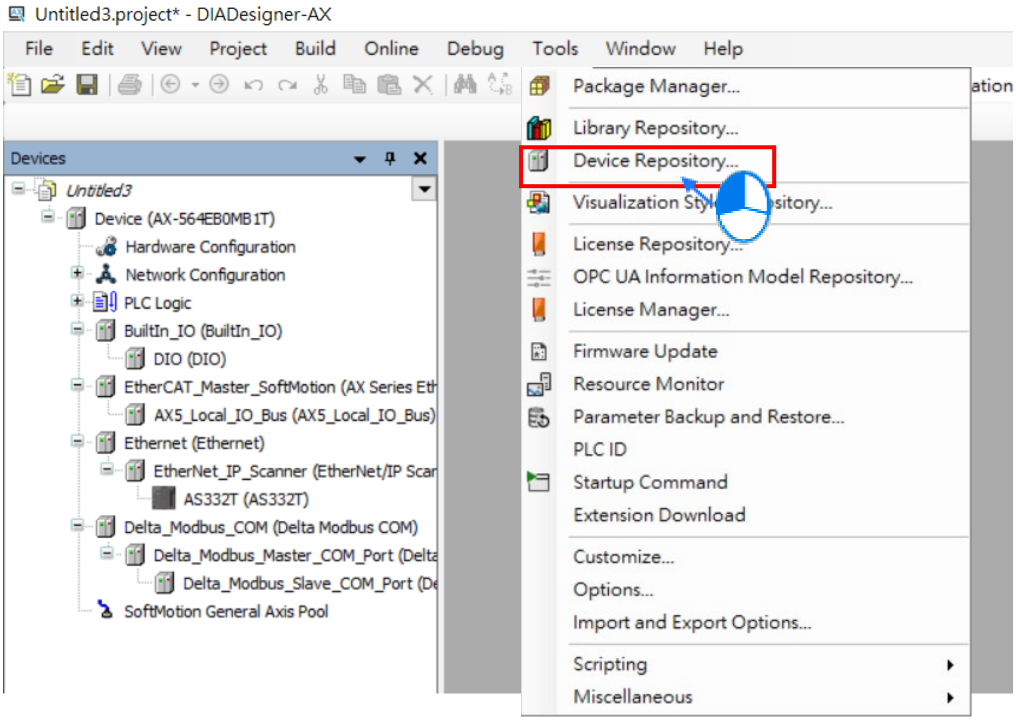
8.4.2.4 Tag Connection

This section demonstrates how AX-5, as an EtherNet/IP Scanner, creates consumer tags to read data from the produced tags of the EIP Adapter.

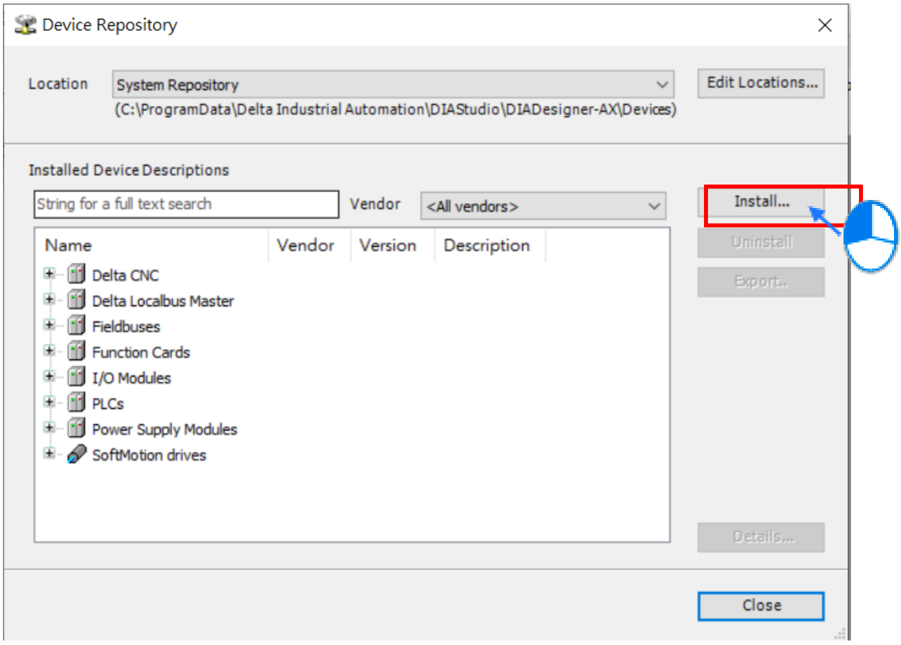
8.4.2.4.1 Installing EDS file of the EtherNet/IP Adapter

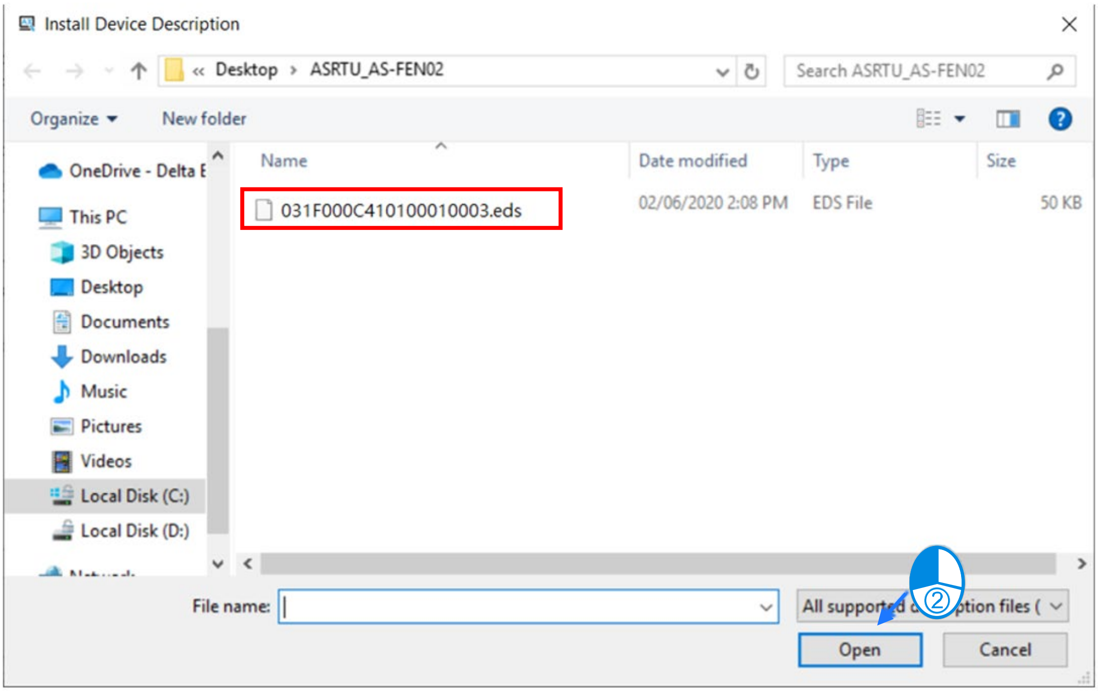
- **Install the EDS file of the EtherNet/IP Adapter**

1. Open Device Repository (Tools -> Device Repository).



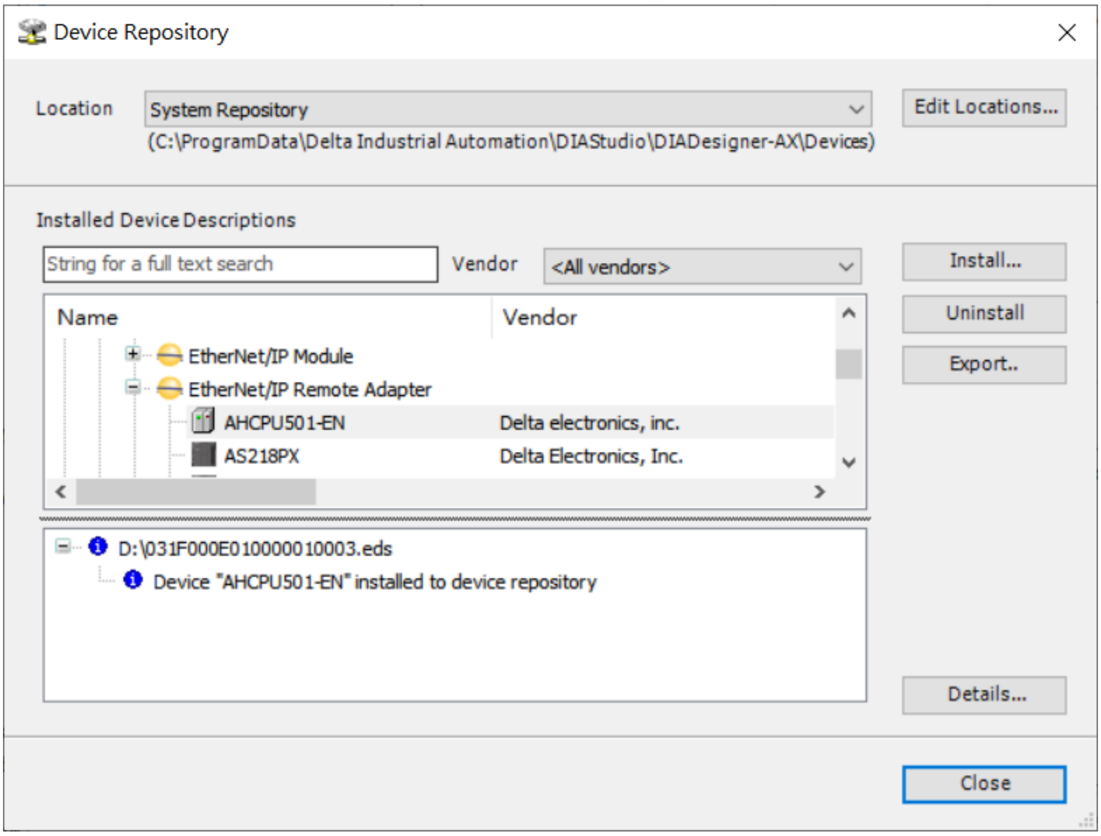
2. Click **Install** to browse and select the EDS file you'd like to install.





Name	Date modified	Type	Size
031F000C410100010003.eds	02/06/2020 2:08 PM	EDS File	50 KB

3. After that the EDS file is installed.



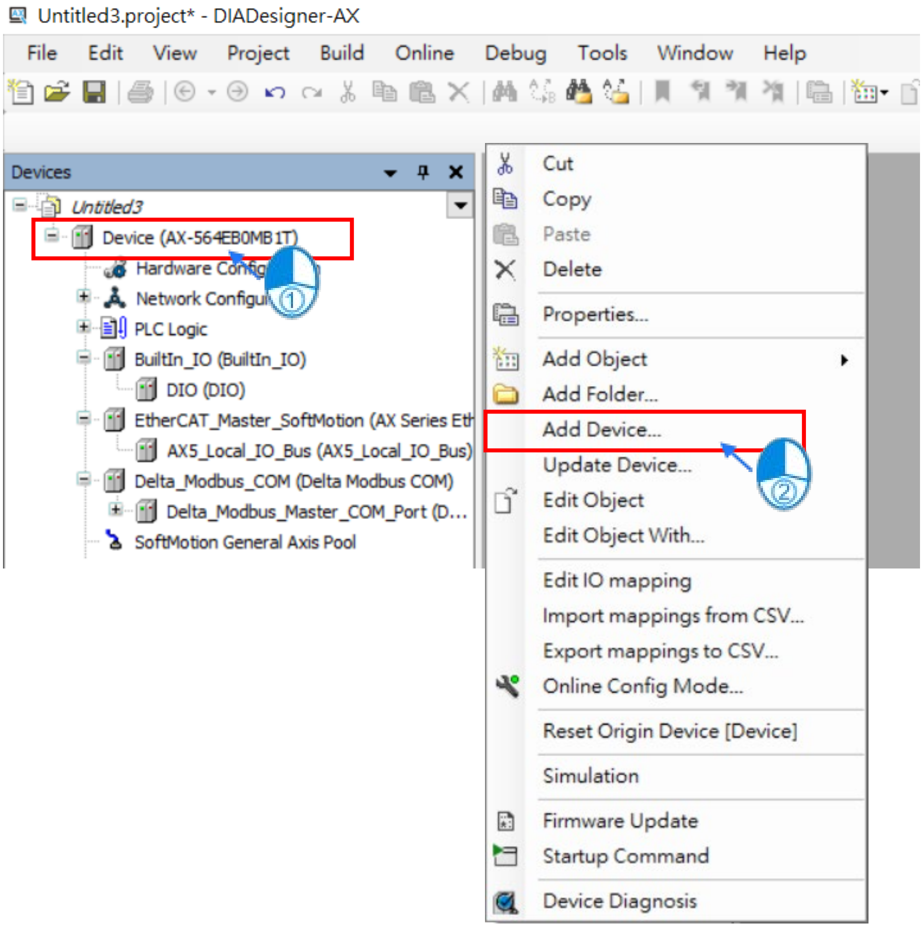
Location: System Repository
(C:\ProgramData\Delta Industrial Automation\DIAStudio\DIADesigner-AX\Devices)

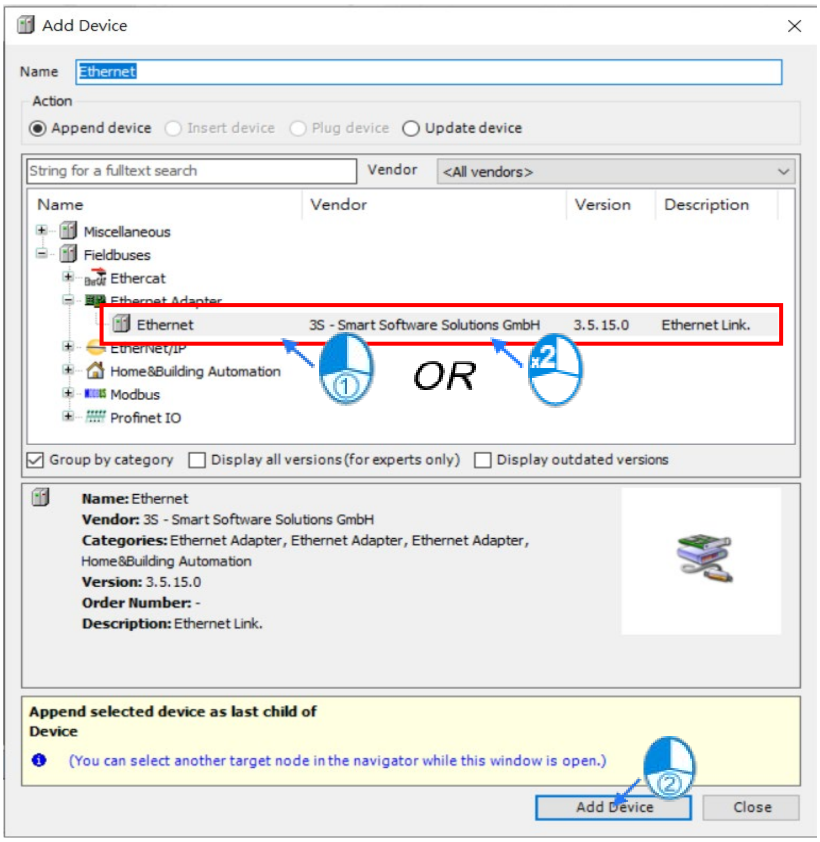
Name	Vendor
EtherNet/IP Module	
EtherNet/IP Remote Adapter	
AHCPU501-EN	Delta electronics, inc.
AS218PX	Delta Electronics, Inc.

D:\031F000E010000010003.eds
Device "AHCPU501-EN" installed to device repository

- **Adding EtherNet/IP Adapter (using AHCPU501-EN as an example) after installing its EDS file**

1. Add Ethernet device.





Add Device

Name: Ethernet

Action

● Append device ○ Insert device ○ Plug device ○ Update device

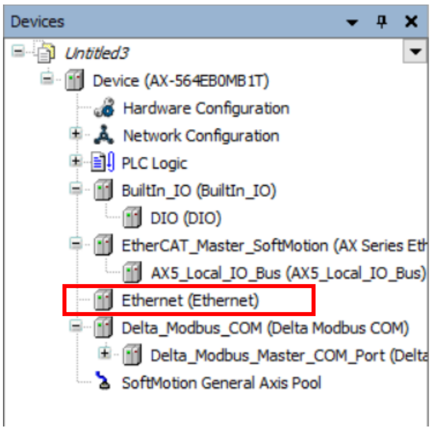
Name	Vendor	Version	Description
Ethernet	3S - Smart Software Solutions GmbH	3.5.15.0	Ethernet Link.

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

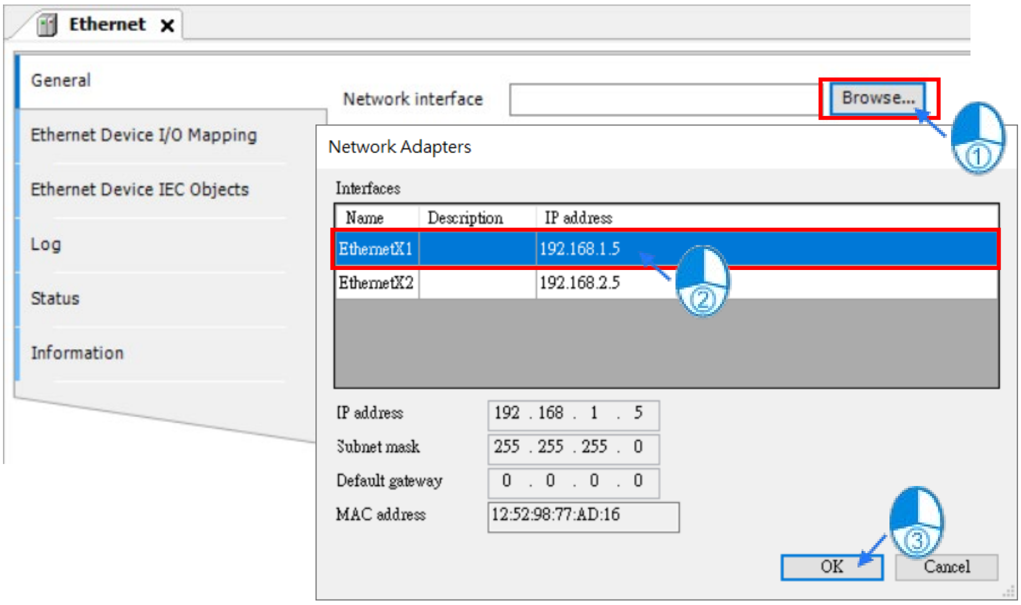
Name: Ethernet
Vendor: 3S - Smart Software Solutions GmbH
Categories: Ethernet Adapter, Ethernet Adapter, Ethernet Adapter, Home&Building Automation
Version: 3.5.15.0
Order Number: -
Description: Ethernet Link.

Append selected device as last child of
Device

(You can select another target node in the navigator while this window is open.)



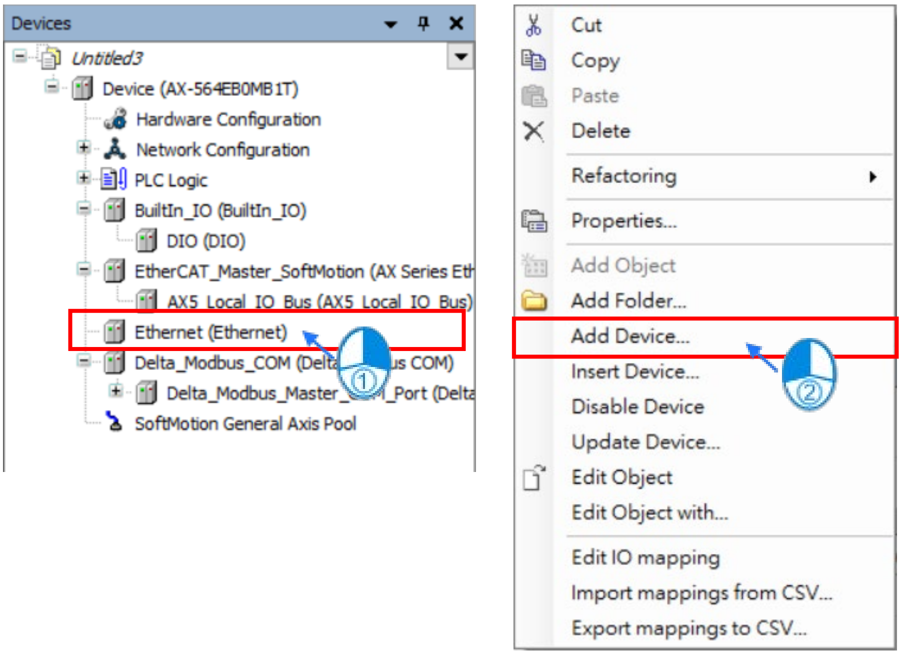
2. Select the network interface.

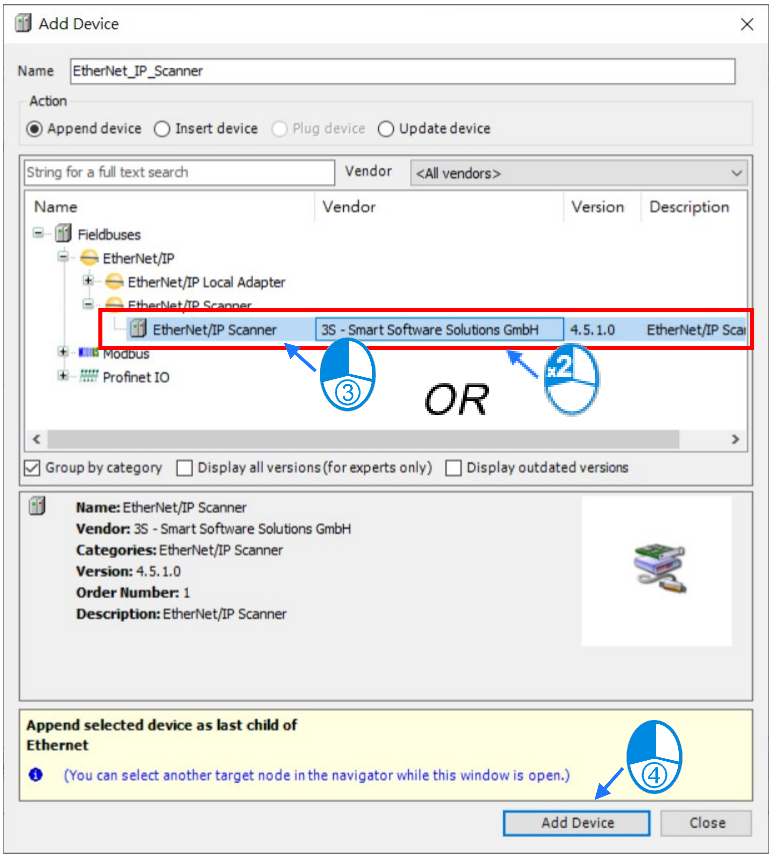


Name	Description	IP address
EthernetX1		192.168.1.5
EthernetX2		192.168.2.5

IP address 192 . 168 . 1 . 5
Subnet mask 255 . 255 . 255 . 0
Default gateway 0 . 0 . 0 . 0
MAC address 12:52:98:77:AD:16

3. Add EtherNet/IP Scanner.





Add Device

Name: EtherNet_IP_Scanner

Action

● Append device ○ Insert device ○ Plug device ○ Update device

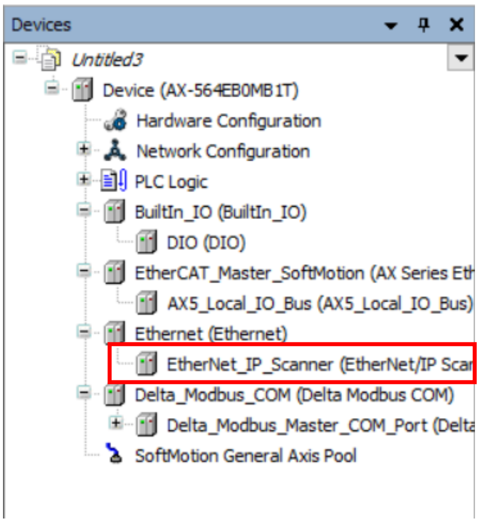
Name	Vendor	Version	Description
EtherNet/IP Scanner	3S - Smart Software Solutions GmbH	4.5.1.0	EtherNet/IP Sca

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

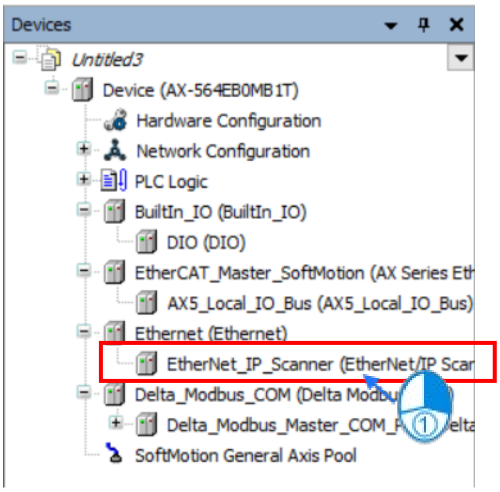
Name: EtherNet/IP Scanner
Vendor: 3S - Smart Software Solutions GmbH
Categories: EtherNet/IP Scanner
Version: 4.5.1.0
Order Number: 1
Description: EtherNet/IP Scanner

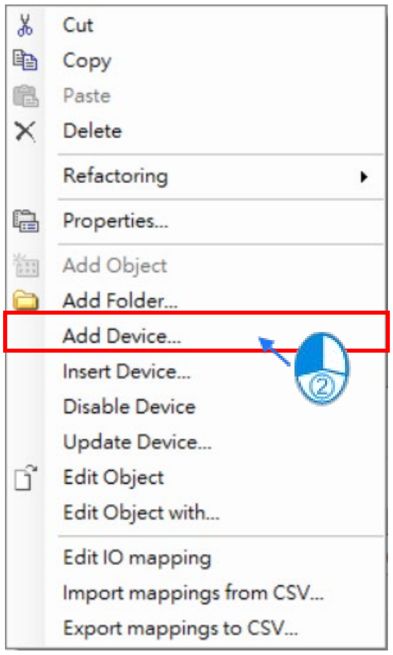
Append selected device as last child of Ethernet

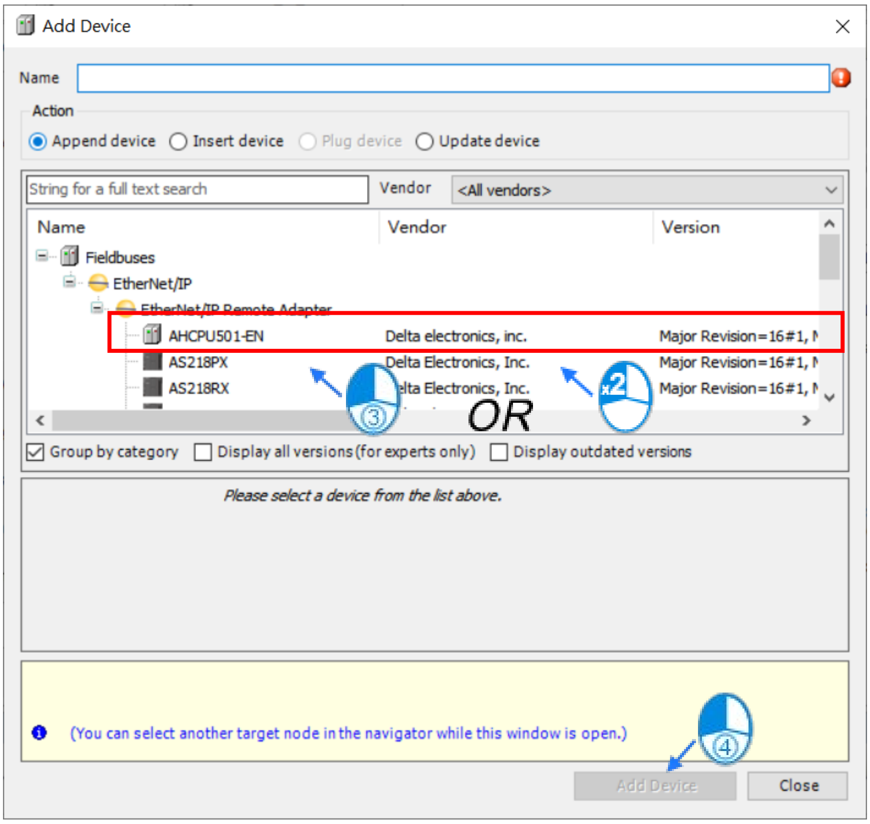
(You can select another target node in the navigator while this window is open.)



4. Add AHCPU501-EN as an EtherNet/IP Adapter.

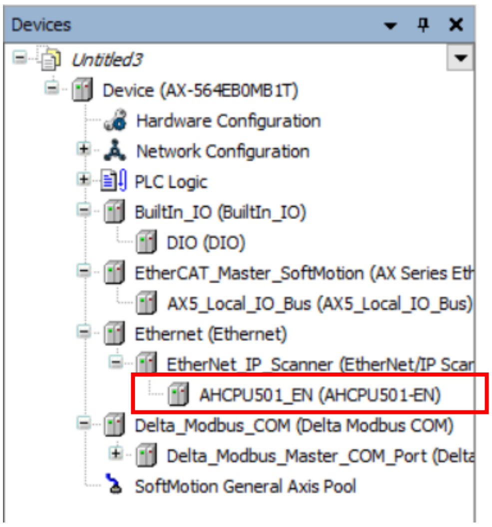






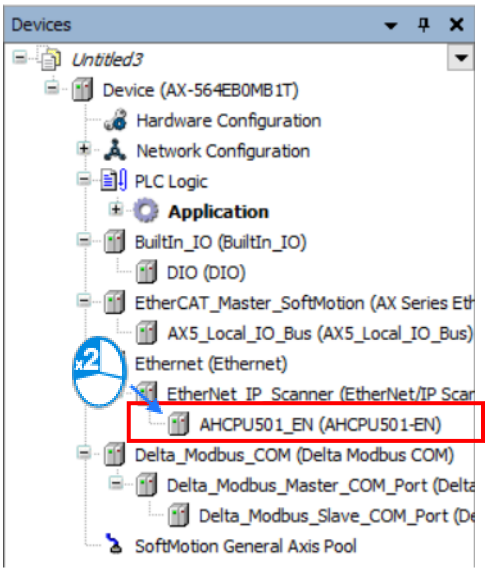
Please select a device from the list above.

(You can select another target node in the navigator while this window is open.)

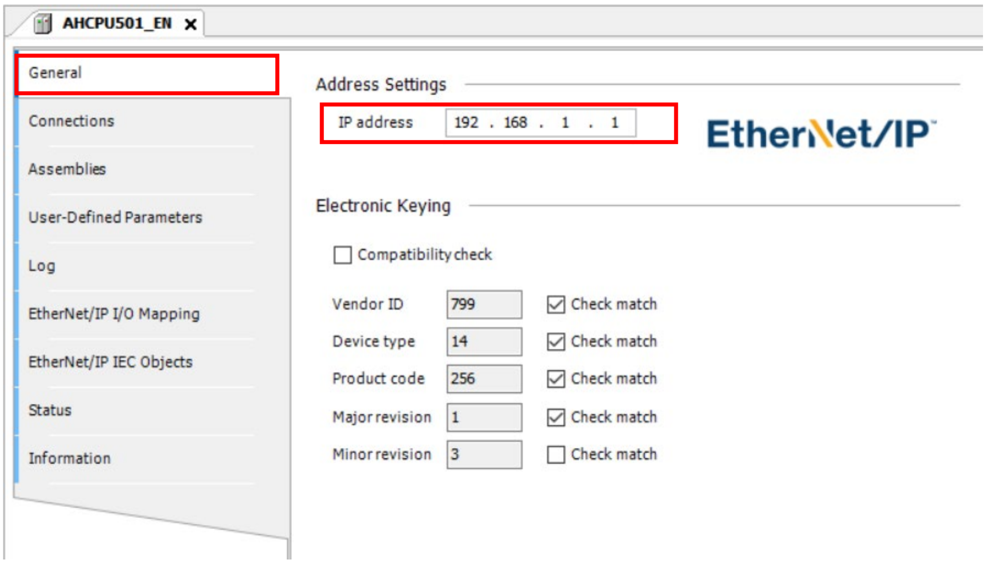


8.4.2.4.2 Setting up EtherNet/IP Adapter (AHCPU501-EN)

You can double click AHCPU501-EN (AHCPU501-EN) under Ethernet (Ethernet) to open the setting page.



- Define the IP address for EtherNet/IP Adapter.



Address Settings

IP address 192 . 168 . 1 . 1

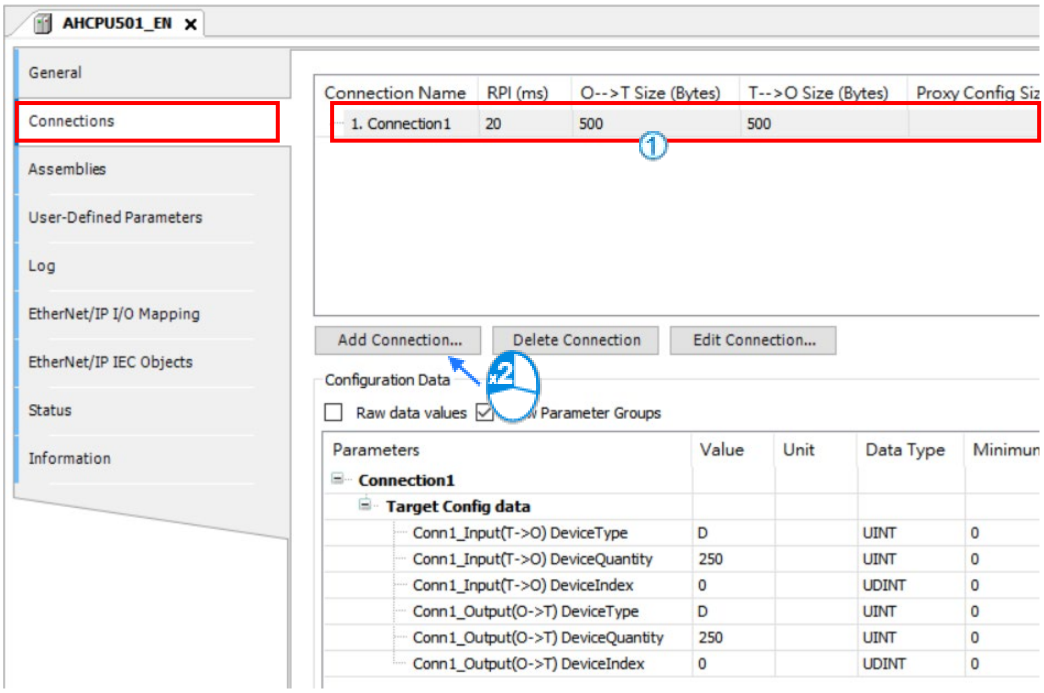
Electronic Keying

☐ Compatibility check

Vendor ID	799	☑ Check match
Device type	14	☑ Check match
Product code	256	☑ Check match
Major revision	1	☑ Check match
Minor revision	3	☐ Check match

- Create EtherNet/IP Adapter Produced Tag

1. Add an EtherNet/IP Adapter Tag Connection



Connection Name	RPI (ms)	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Siz
1. Connection1	20	500	500	

Configuration Data

☐ Raw data values ☑ Parameter Groups

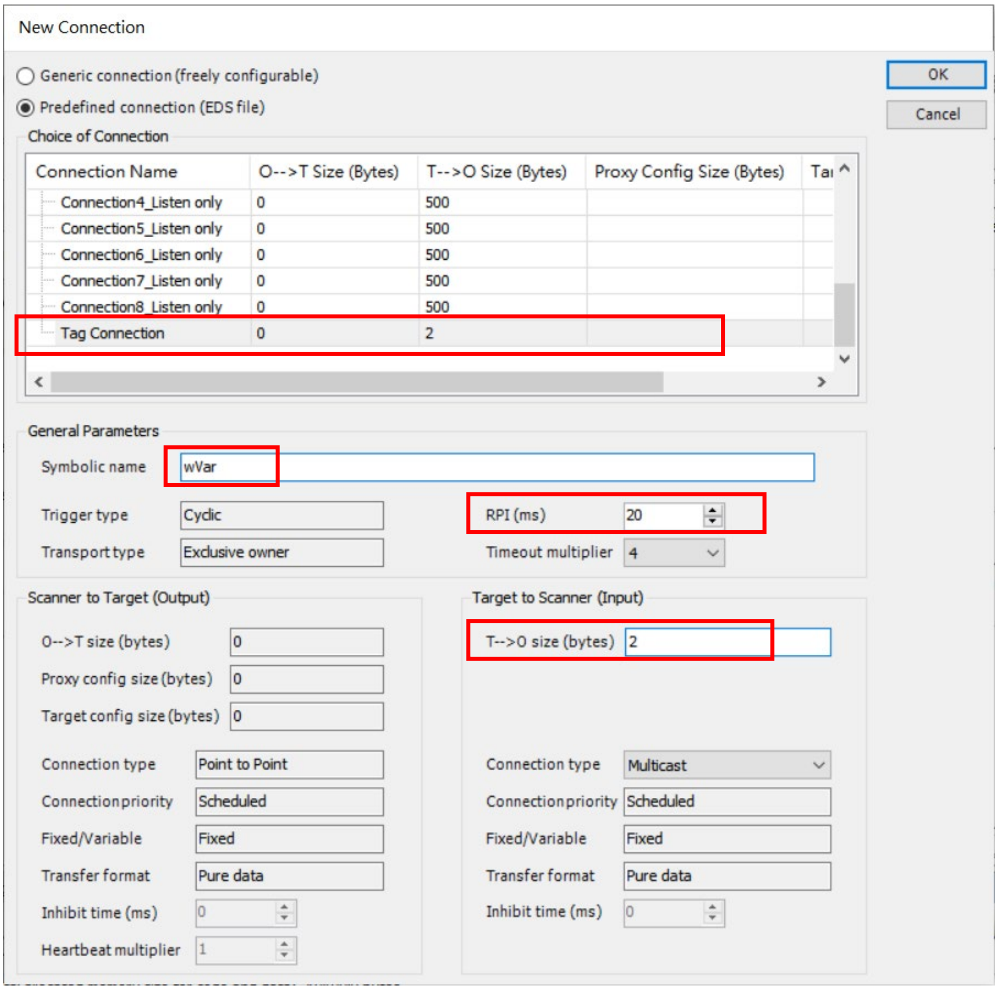
Parameters	Value	Unit	Data Type	Minimun
Connection1				
Target Config data				
Conn1_Input(T->O) DeviceType	D		UINT	0
Conn1_Input(T->O) DeviceQuantity	250		UINT	0
Conn1_Input(T->O) DeviceIndex	0		UDINT	0
Conn1_Output(O->T) DeviceType	D		UINT	0
Conn1_Output(O->T) DeviceQuantity	250		UINT	0
Conn1_Output(O->T) DeviceIndex	0		UDINT	0

Note: ① comes with the imported EDS file for IO Connection, you can delete it if it’s not needed.

2. Set up EtherNet/IP Adapter Produced Tag

You need to add a tag connection and set up the tag name, tag PRI, and the tag data bytes. As you can find in the example, the tag name is “wVar”, PRI time is set to 20 ms, and the tag data is 2 bytes.

Note: You need to define the Produced Tag for the EtherNet/IP Adapter device.



New Connection

○ Generic connection (freely configurable)

● Predefined connection (EDS file)

Choice of Connection

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Tai
Connection4_Listen only	0	500		
Connection5_Listen only	0	500		
Connection6_Listen only	0	500		
Connection7_Listen only	0	500		
Connection8_Listen only	0	500		
Tag Connection	0	2		

General Parameters

Symbolic name: wVar

Trigger type: Cyclic

Transport type: Exclusive owner

RPI (ms): 20

Timeout multiplier: 4

Scanner to Target (Output)

O-->T size (bytes): 0

Proxy config size (bytes): 0

Target config size (bytes): 0

Connection type: Point to Point

Connection priority: Scheduled

Fixed/Variable: Fixed

Transfer format: Pure data

Inhibit time (ms): 0

Heartbeat multiplier: 1

Target to Scanner (Input)

T-->O size (bytes): 2

Connection type: Multicast

Connection priority: Scheduled

Fixed/Variable: Fixed

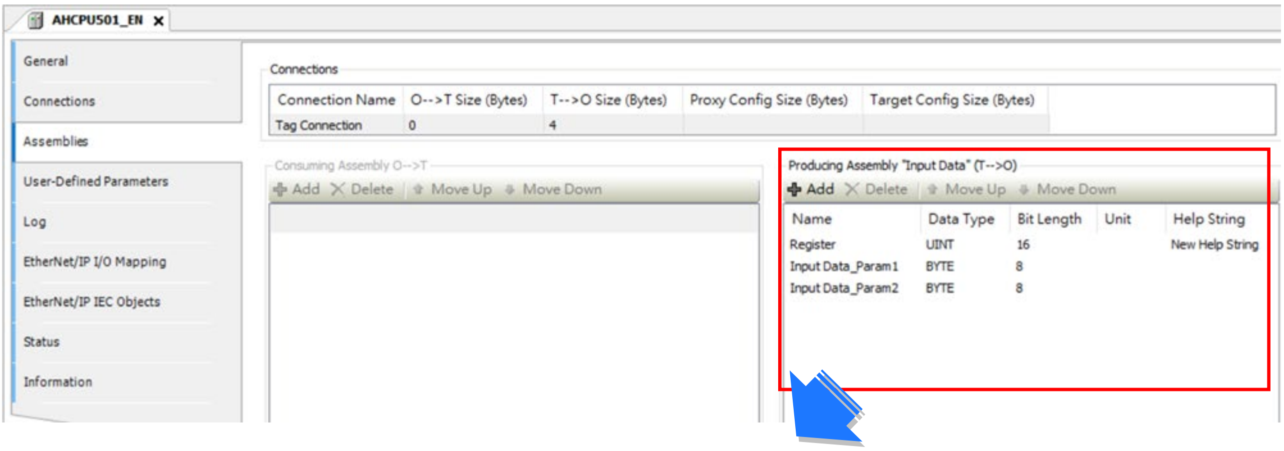
Transfer format: Pure data

Inhibit time (ms): 0

8.4.2.5 Assemblies

DIADesigner-AX may generate a data structure that does not align with your expectations. When this happens, you can reconfigure the data structure through Assemblies. For instance, if the Tag Connection data length is 4 bytes, but DIADesigner-AX automatically organizes it as one UINT and two bytes, the EtherNet/IP I/O Mapping page will display the Tag Connection data as three separate records, which can create difficulty in interpreting the data.

- **Before assembling**

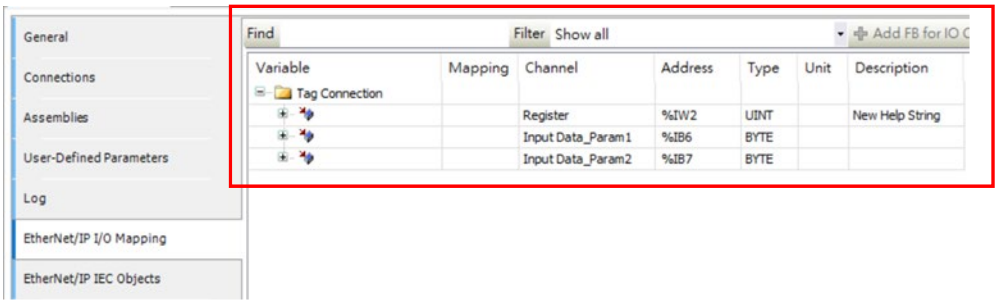


Connections

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)
Tag Connection	0	4		

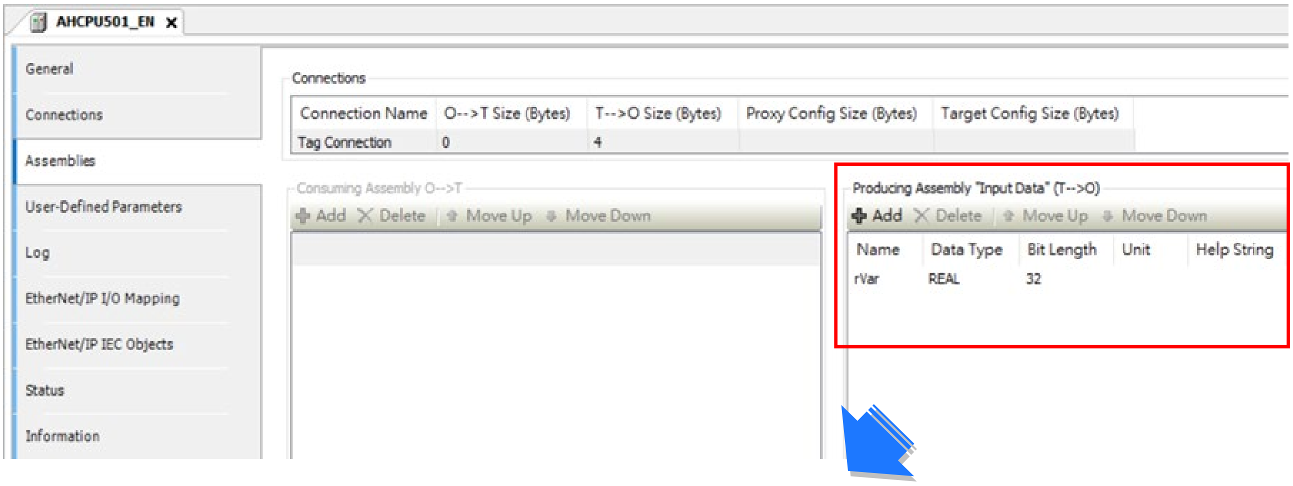
Producing Assembly "Input Data" (T-->O)

Name	Data Type	Bit Length	Unit	Help String
Register	UINT	16		New Help String
Input Data_Param1	BYTE	8		
Input Data_Param2	BYTE	8		



Variable	Mapping	Channel	Address	Type	Unit	Description
Tag Connection						
		Register	%IW2	UINT		New Help String
		Input Data_Param1	%IB6	BYTE		
		Input Data_Param2	%IB7	BYTE		

- **After assembling**

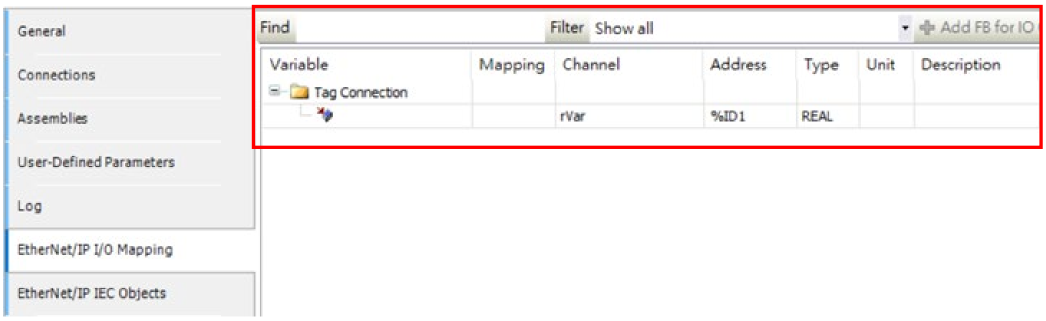


Connections

Connection Name	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)
Tag Connection	0	4		

Producing Assembly "Input Data" (T-->O)

Name	Data Type	Bit Length	Unit	Help String
rVar	REAL	32		



Filter Show all

Variable	Mapping	Channel	Address	Type	Unit	Description
Tag Connection						
		rVar	%ID1	REAL		

8.4.3 EtherNet/IP Adapter Function

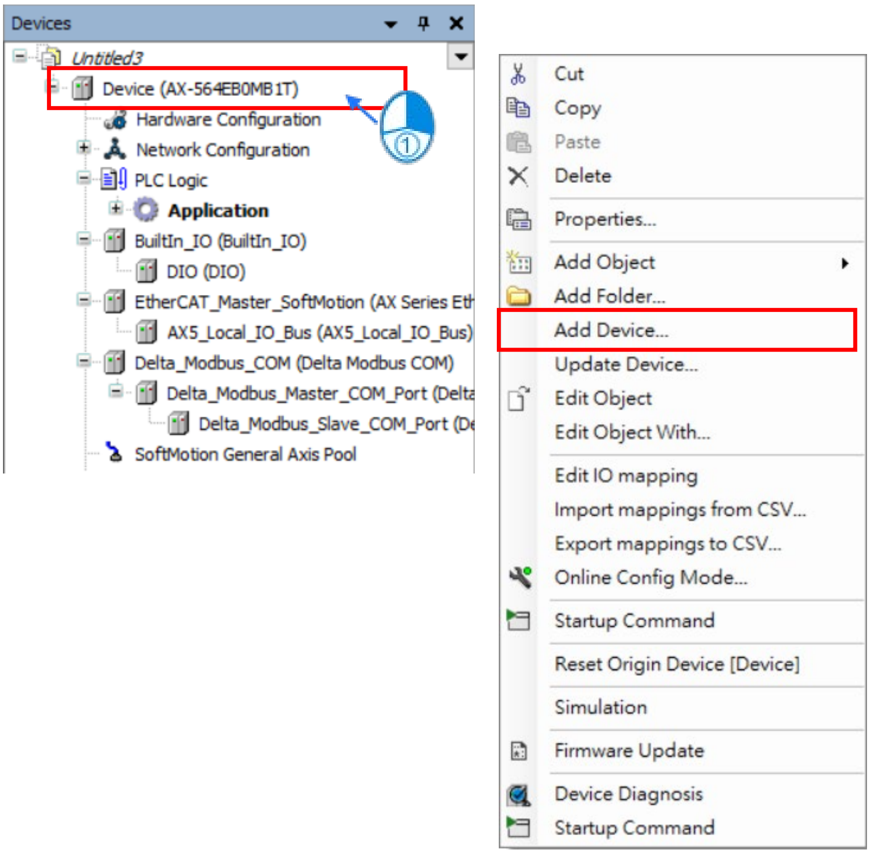
When the Delta_EtherNet_IP_Adapter is added, DIADesigner-AX will automatically generate the Delta_EtherNet_IP_Adapter.IOCycle and Delta_EtherNet_IP_Adapter.ServiceCycle POU and assign them to ENIPAdapterIOTask and ENIPAdapterServiceTask respectively. Do not adjust the allocation of POU and task settings unless necessary to avoid related functional abnormalities.

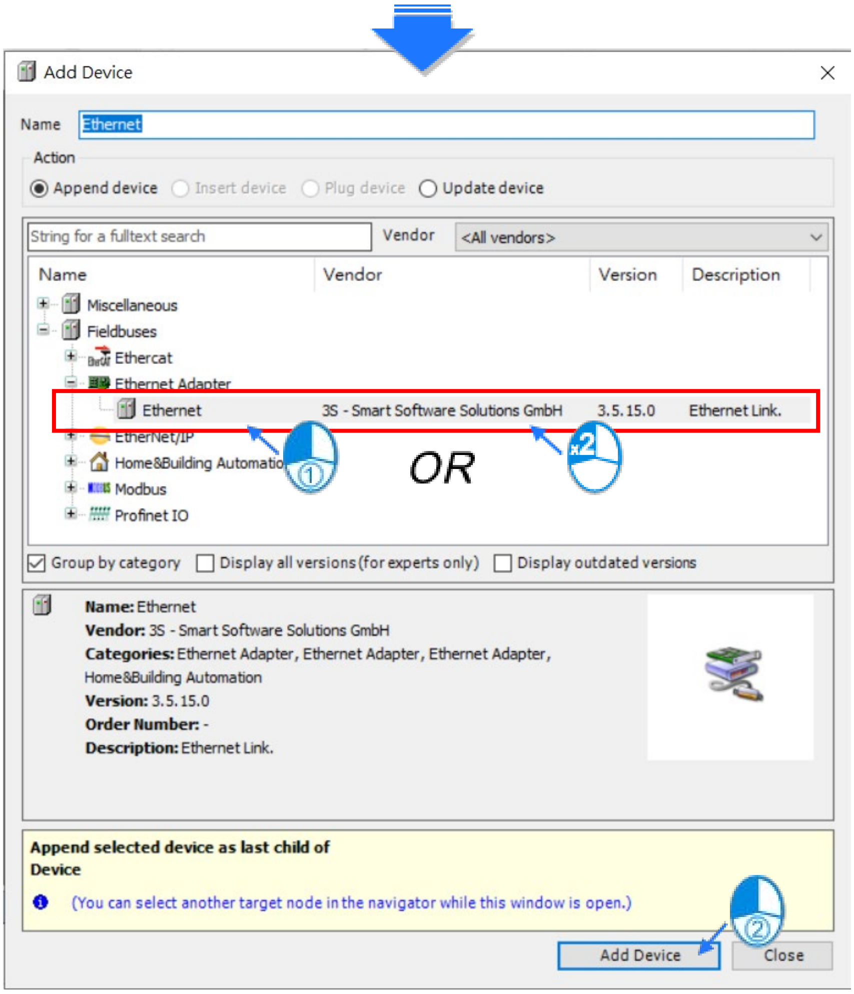
8.4.3.1 EtherNet/IP Adapter

This section will explain how to add and set up an EtherNet/IP Adapter. Refer to the followings for configuration.

8.4.3.1.1 Setting up EtherNet/IP Adapter

- Right-click the **Device (AX-564EB0MB1T))** in the tree view to open a context menu. And click **Add Device…**to open the setting page and add the EtherNet in.





Add Device

Name: Ethernet

Action

● Append device ○ Insert device ○ Plug device ○ Update device

String for a fulltext search Vendor <All vendors>

Name	Vendor	Version	Description
Miscellaneous			
Fieldbuses			
Ethercat			
Ethernet Adapter			
Ethernet	3S - Smart Software Solutions GmbH	3.5.15.0	Ethernet Link.
EtherNet/IP			
Home&Building Automatio			
Modbus			
Profinet IO			

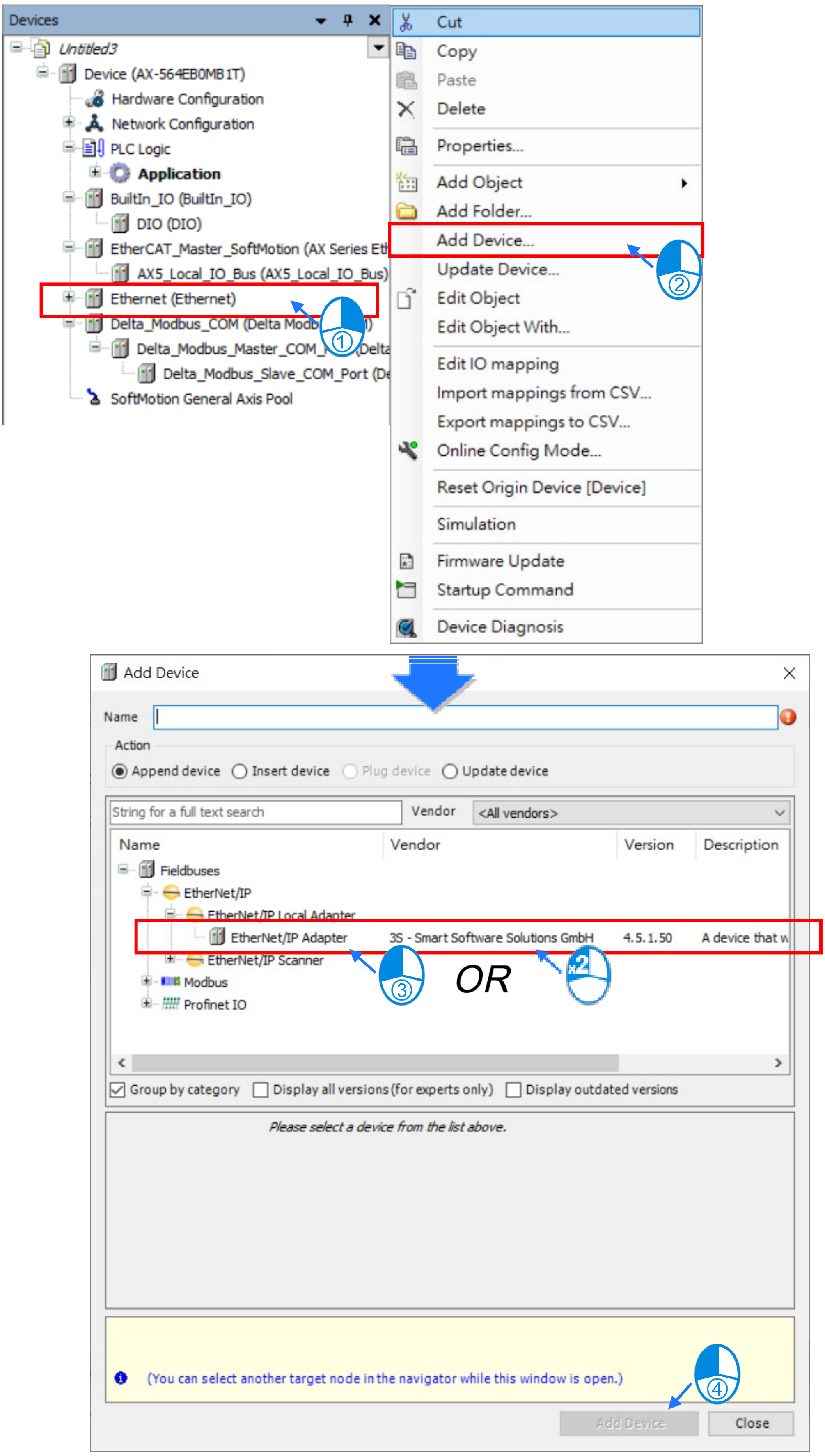
☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: Ethernet
Vendor: 3S - Smart Software Solutions GmbH
Categories: Ethernet Adapter, Ethernet Adapter, Ethernet Adapter, Home&Building Automation
Version: 3.5.15.0
Order Number: -
Description: Ethernet Link.

Append selected device as last child of
Device

(You can select another target node in the navigator while this window is open.)

- **Adding Delta_EtherNet_IP_Adapter Device**



Action: ● Append device ○ Insert device ○ Plug device ○ Update device

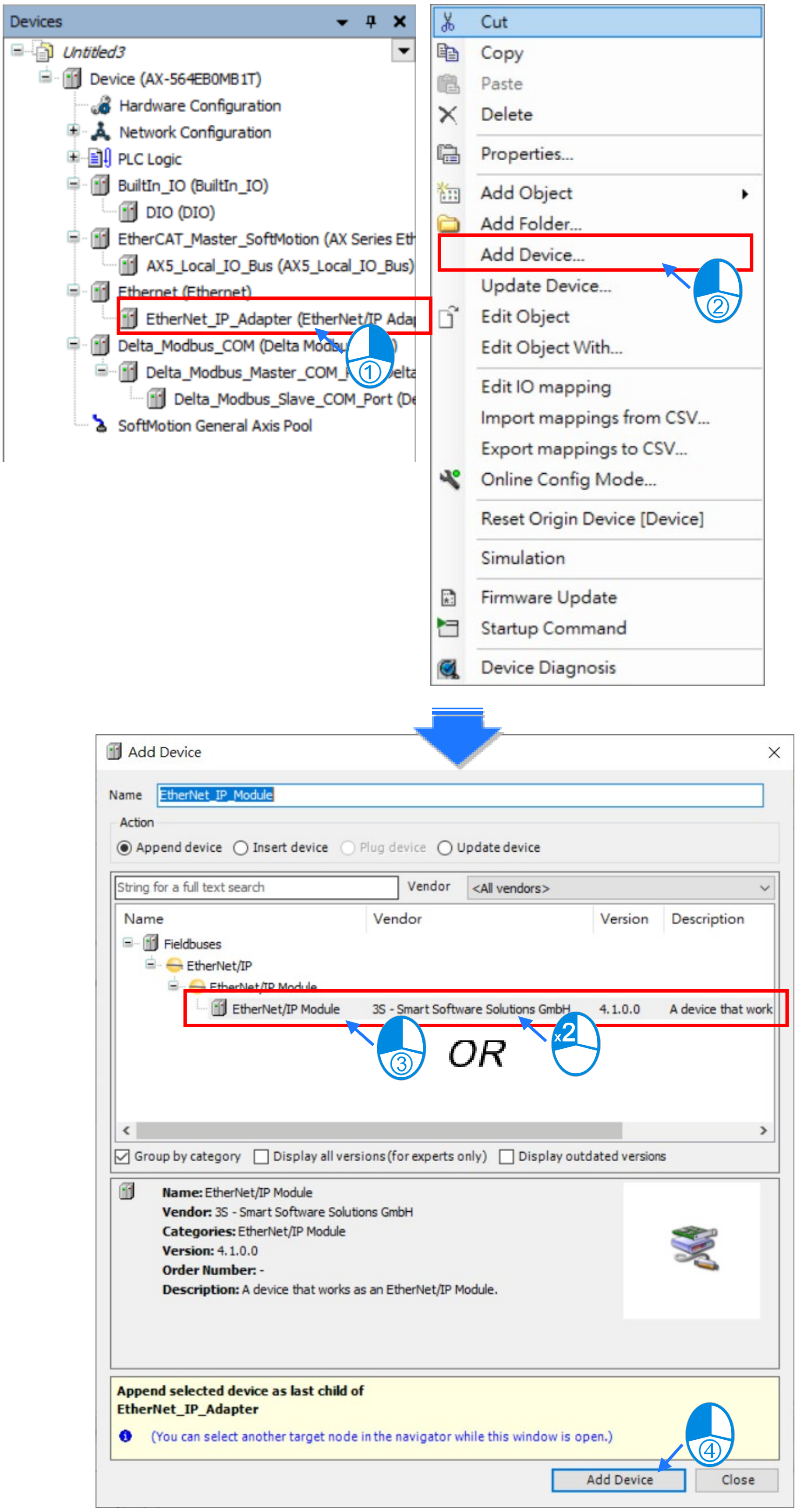
Name	Vendor	Version	Description
EtherNet/IP Adapter	3S - Smart Software Solutions GmbH	4.5.1.50	A device that w

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Please select a device from the list above.

(You can select another target node in the navigator while this window is open.)

- **Adding EtherNet/IP Adapter Data in.**



Name: EtherNet_IP_Module

Action: ● Append device ○ Insert device ○ Plug device ○ Update device

String for a full text search
Vendor <All vendors>

Name	Vendor	Version	Description
Fieldbuses			
EtherNet/IP			
EtherNet/IP Module			
EtherNet/IP Module	3S - Smart Software Solutions GmbH	4.1.0.0	A device that work

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

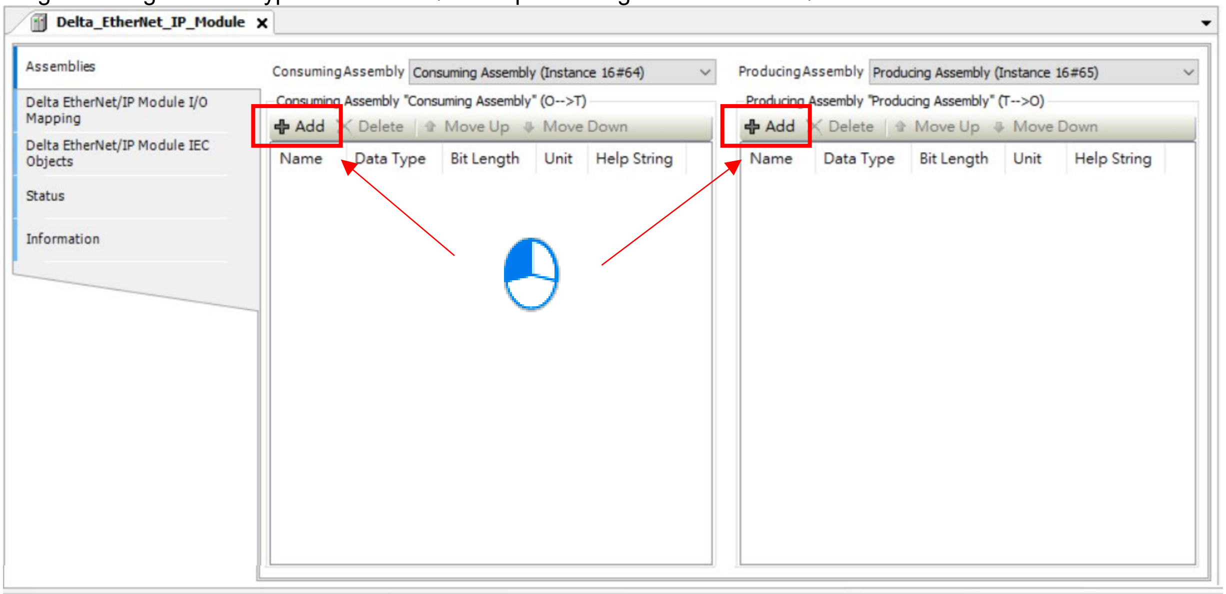
Name: EtherNet/IP Module
Vendor: 3S - Smart Software Solutions GmbH
Categories: EtherNet/IP Module
Version: 4.1.0.0
Order Number: -
Description: A device that works as an EtherNet/IP Module.

Append selected device as last child of
EtherNet_IP_Adapter

(You can select another target node in the navigator while this window is open.)

- Delta EtherNet/IP Module

  Adding or setting the data type for EtherNet/IP Adapter through Delta EtherNet/IP Module.

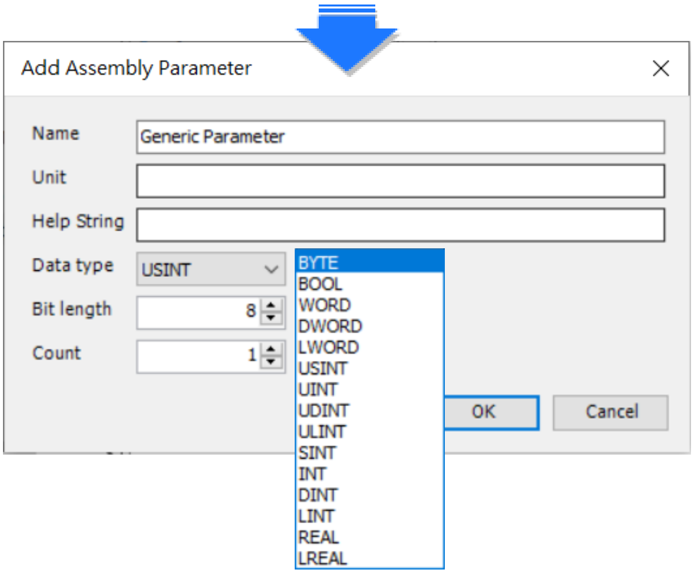


Consuming Assembly: Consuming Assembly (Instance 16#64)

Consuming Assembly "Consuming Assembly" (O-->T)

Producing Assembly: Producing Assembly (Instance 16#65)

Producing Assembly "Producing Assembly" (T-->O)



Name: Generic Parameter

Unit:

Help String:

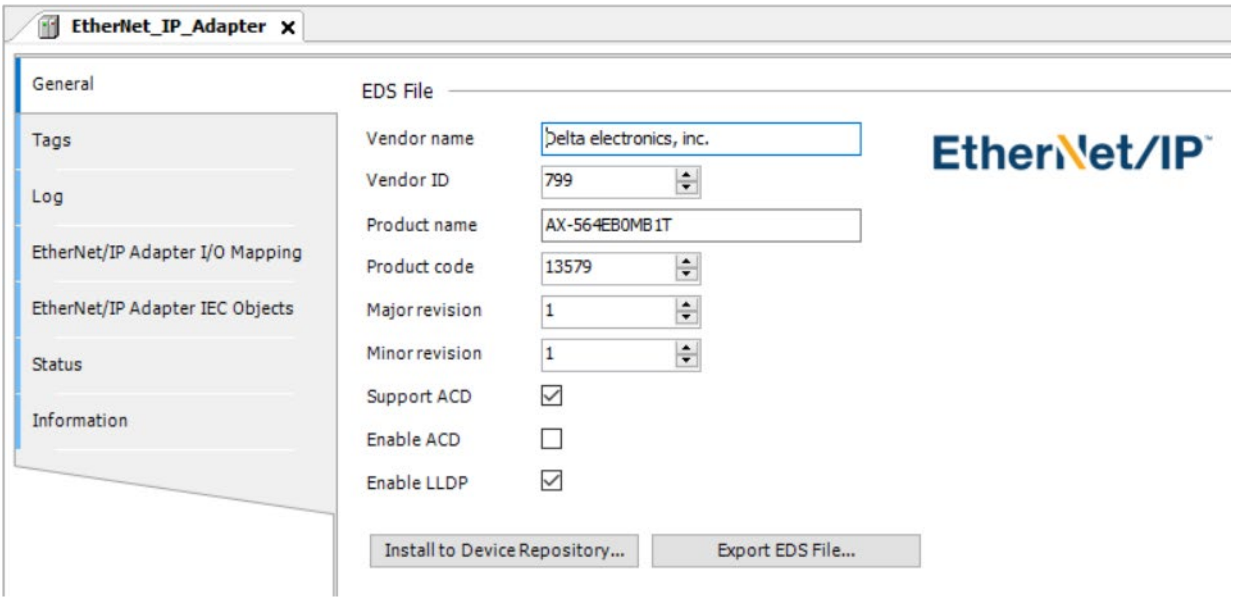
Data type: USINT

Bit length: 8

Count: 1

8.4.3.1.2 Setting up EtherNet/IP Adapter

- Delta EtherNet/IP Adapter
  - General – Setup EDS File

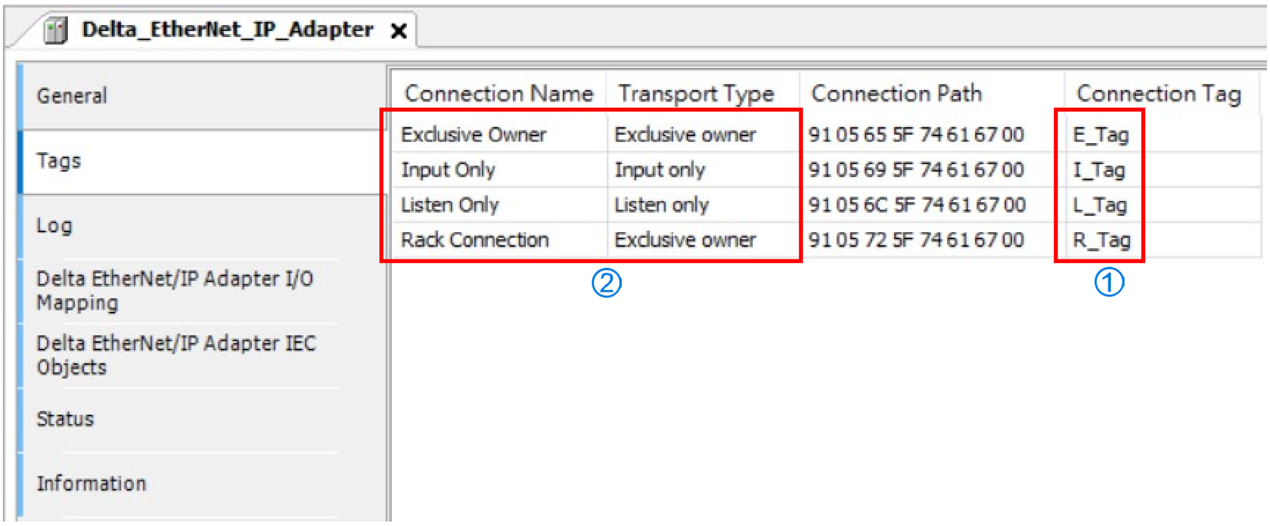


Vendor name: Delta electronics, inc.
Vendor ID: 799
Product name: AX-564EB0MB1T
Product code: 13579
Major revision: 1
Minor revision: 1
Support ACD: ☑
Enable ACD: ☐
Enable LLDP: ☑

Item	Description	Default
Vendor name	The name of the supplier	Delta electronics, inc.
Vendor ID	Supplier ID	799
Product name	The name of the product	AX-564EB0MB1T
Product code	Product code	16386
Major revision	Major revision	1
Minor revision	Minor revision	1
Enable ACD	EtherNet/IP Adapter IP conflict detection.	Disable
Install to Device Repository	In case that a device with the same device identification has already been installed, you would be asked whether the device should be overwritten. If the device is taken as the remote adapter inserted directly below the EtherNet/IP scanner, you would be asked to update the device automatically.	
Export EDS File	The EDS file is created and stored on the local computer. In this way, the EDS file can be used in an external configuration file.	

8.4.3.1.3. Tag Connection

You can define the tags under Tags tab. And then EtherNet/IP Scanner can read or set up tags through Tag Connection.



Connection Name	Transport Type	Connection Path	Connection Tag
Exclusive Owner	Exclusive owner	91 05 65 5F 74 61 67 00	E_Tag
Input Only	Input only	91 05 69 5F 74 61 67 00	I_Tag
Listen Only	Listen only	91 05 6C 5F 74 61 67 00	L_Tag
Rack Connection	Exclusive owner	91 05 72 5F 74 61 67 00	R_Tag

① Connection Tag Name: EtherNet/IP Scanner uses the tag names defined here to perform data exchange. If this field is left blank, the data exchange will be performed by IO Connection.

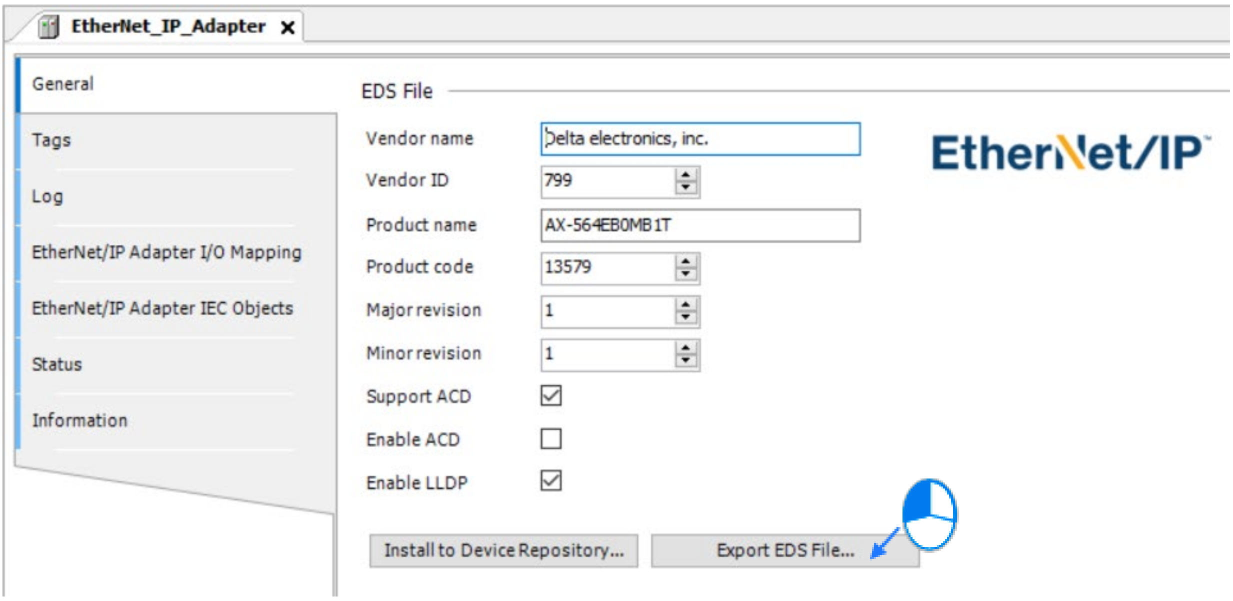
② Transport Type

Transport Type		Description
1	Exclusive owner	This type of connection allows for the simultaneous transmission of both input and output data.
2	Input only	This type of connection only accepts data of the Input type.
3	Listen only	This connection type is designed to monitor input data from an established EtherNet/IP connection. If the primary EtherNet/IP connection is disrupted, this connection will also become inactive.
4	Rack Connection	This type of connection involves the simultaneous transfer of both input and output data.

Please be aware that when importing the EDS parameter file into an Allen-Bradley device, only one Tag Connection and one Tag data can be defined.

8.4.3.1.4 Exporting an EDS File

After the configuration is complete, export the EDS file and store the EDS file – AX-564EB0MB1T.eds in the PC.



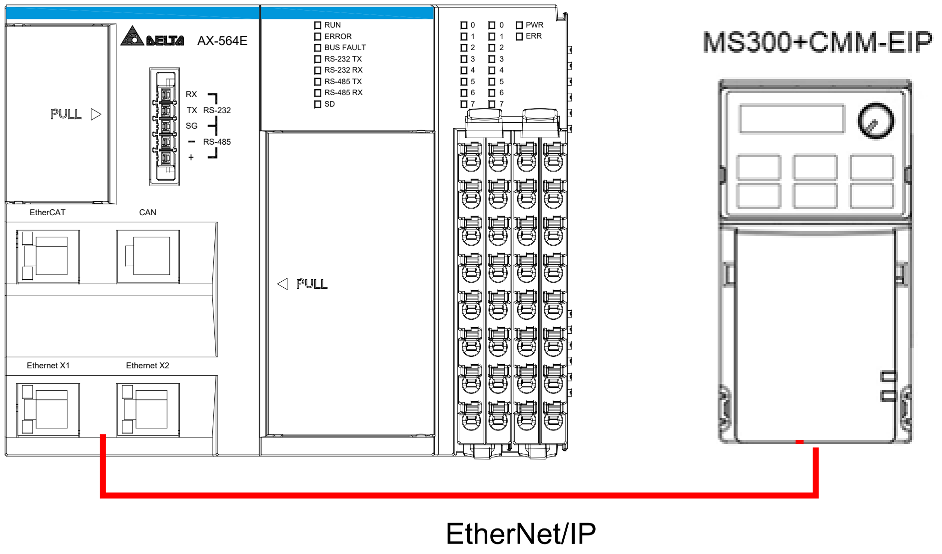
Vendor name: Delta electronics, inc.
Vendor ID: 799
Product name: AX-564EB0MB1T
Product code: 13579
Major revision: 1
Minor revision: 1
Support ACD: ☑
Enable ACD: ☐
Enable LLDP: ☑

8.4.4 Operational Example of EtherNet/IP Scanner

8.4.4.1 Setting up Compact Inverter VFD-MS300

8.4.4.1.1 Hardware Configuration

This application example is to connect AX-5 series to compact inverter VFD-MS300 with CMM-EIP communication card through Ethernet port.



Note: The version of CMM-EIP communication card should be V2.04.01 or above.

8.4.4.1.2 Read-Write Setting for Implicit Messages

EtherNet/IP implicit message is for periodic data exchange, utilizing the mapping registers of the CMM-EIP communication card to establish read/write data addresses via the EtherNet/IP scanner. Subsequently, it utilizes the mapping registers to conduct one-time data read/write.

- Before using VFD-MS300 with CMM-EIP communication card

- Settings for VFD-MS300

  To control VFD-MS300 with the CMM-EIP communication card through the network, the control authority of VFD-MS300 should be initially set to the CMM-EIP communication card. Refer to the steps below for setup.

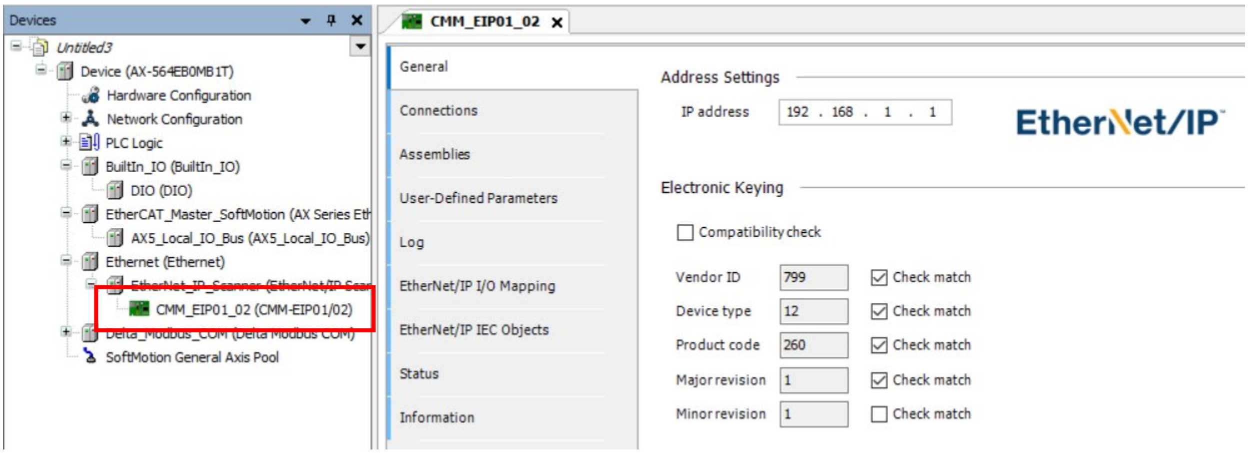
1. When the CMM-EIP communication card is connected, check if the value in parameters 09 to 60 is 5 (EtherNet/IP).
2. Set parameter 09-75=0 (static IP) and the IP address is user-defined.
3. Change the IP address of the CMM-EIP communication card to 192.168.1.30 (default is 192.168.1.5) from parameter 09-76 to parameter 09-79. Then set parameter 09-91 to 2.
4. Set parameter 00-20 to 8: setting the source for AUTO frequency command to the CMM-EIP communication card.
5. Set parameter 00-21 to 5: setting the source for AUTO control to the CMM-EIP communication card.
6. Set parameter 09-30 to 1: setting the communication decoding method to 60xx or 20xx. Refer to section 4.2 from the EtherNet/IP Control Method Standard of VFD EtherNet/IP Application Manual for more details.)

- Example of creating an EIP operation

The IP address of the devices applied in this example are shown as follows:

Devices	**AX-54EB0MB1T**	192.168.1.5 (default)
	MS300 (CMM-EIP02)	192.168.1.30

1. Create a n EtherNet/IP Scanner and CMM-EIP02 (acts as EtherNet/I Adapter) under Ethernet. Refer to sections 8.4.1.1 and 8.4.2.1 for more details.



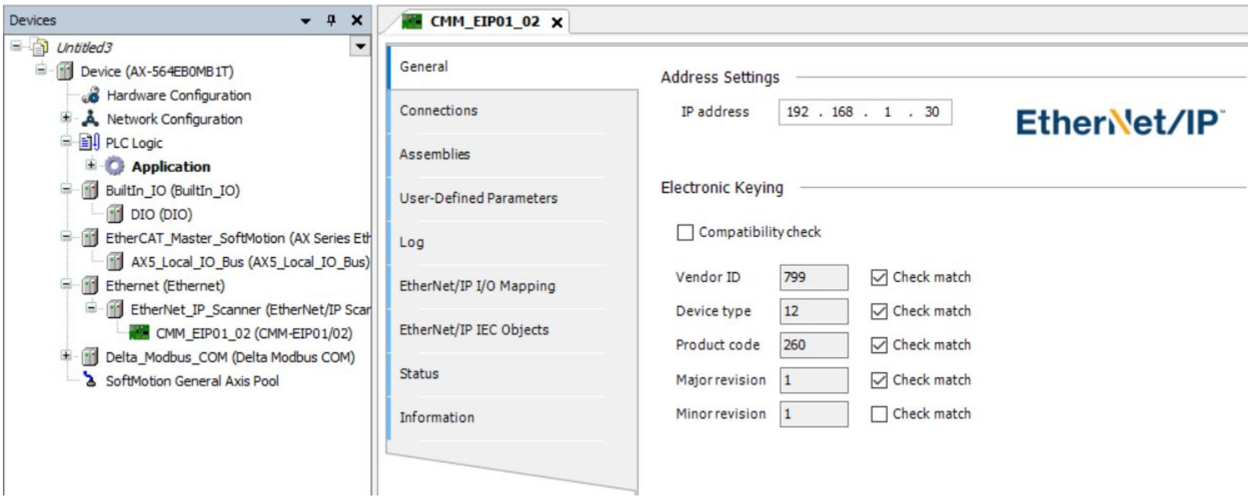
IP address 192 . 168 . 1 . 1

☐ Compatibility check

Vendor ID	799	☑ Check match
Device type	12	☑ Check match
Product code	260	☑ Check match
Major revision	1	☑ Check match
Minor revision	1	☐ Check match

*Note: Adapters can be created via “Scan For Device”.

2. Double-click **CMM_EIP0102** and go to the General tab. Set IP Address to 192.168.1.30.



IP address 192 . 168 . 1 . 30

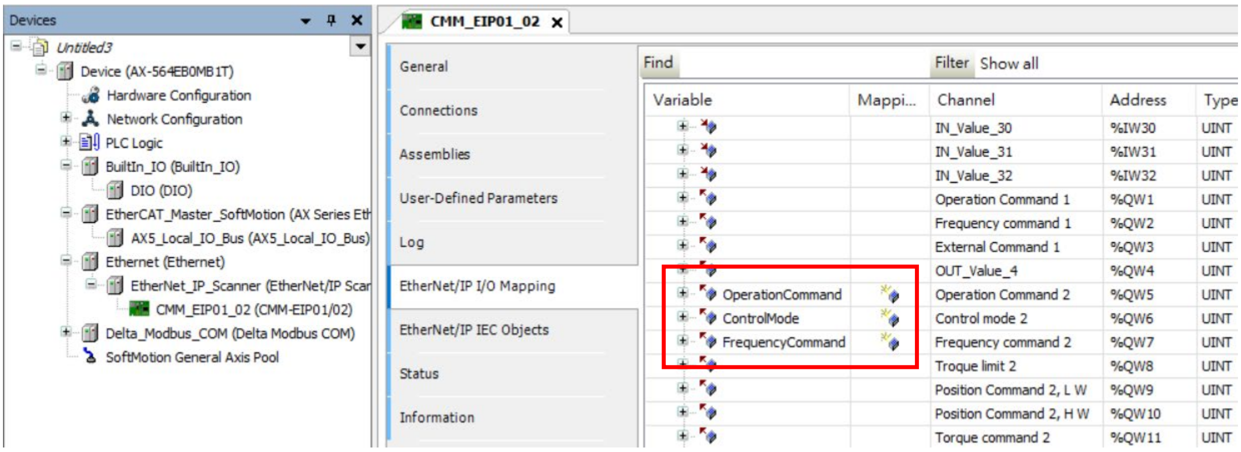
☐ Compatibility check

Vendor ID	799	☑ Check match
Device type	12	☑ Check match
Product code	260	☑ Check match
Major revision	1	☑ Check match
Minor revision	1	☐ Check match

Item	Description
IP address	The IP address of the target device.
Compatibility check	Check the compatibility between the target device and information of EDS files.
Strict identity check	Strictly check the information of the target device and EDS files. Inspection information is user-defined.
Check Device type	Check the device type.
Check Vendor ID	Vendor ID
Check Product code*	Product code*
Check Major revision	Major revision
Check Minor revision	Minor revision

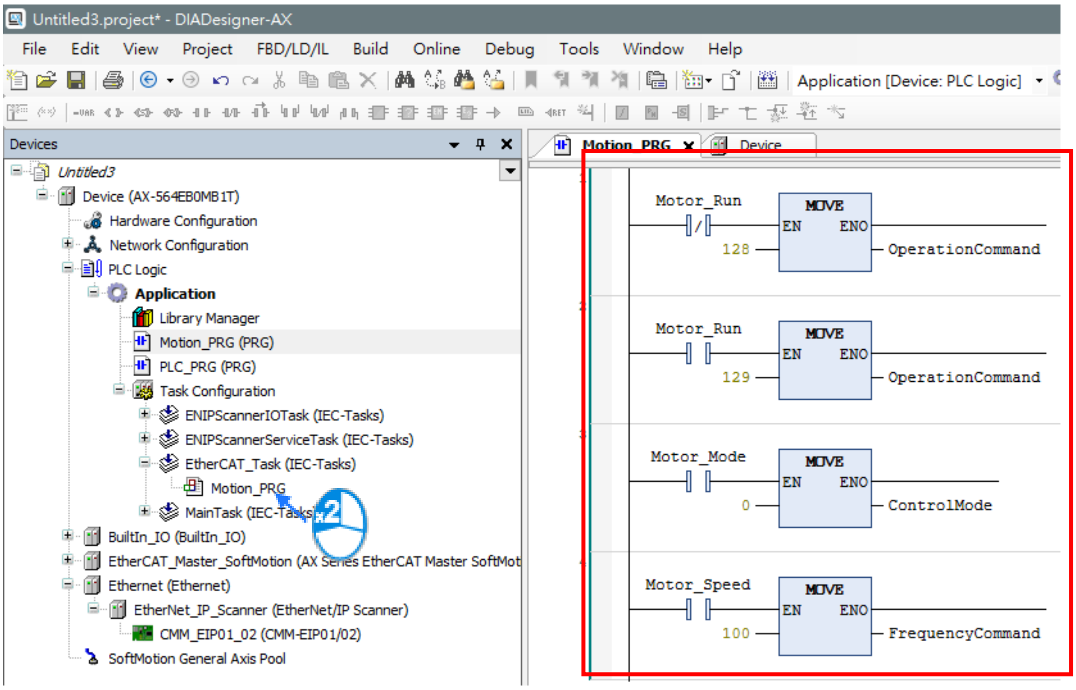
*Note: If Adapter and Scanner are required at the same time, please deselect Check Product code.

3. Go to EtherNet/IP I/O Mapping and add variable name for Operation Command, Control Mode and Frequency Command of channels.

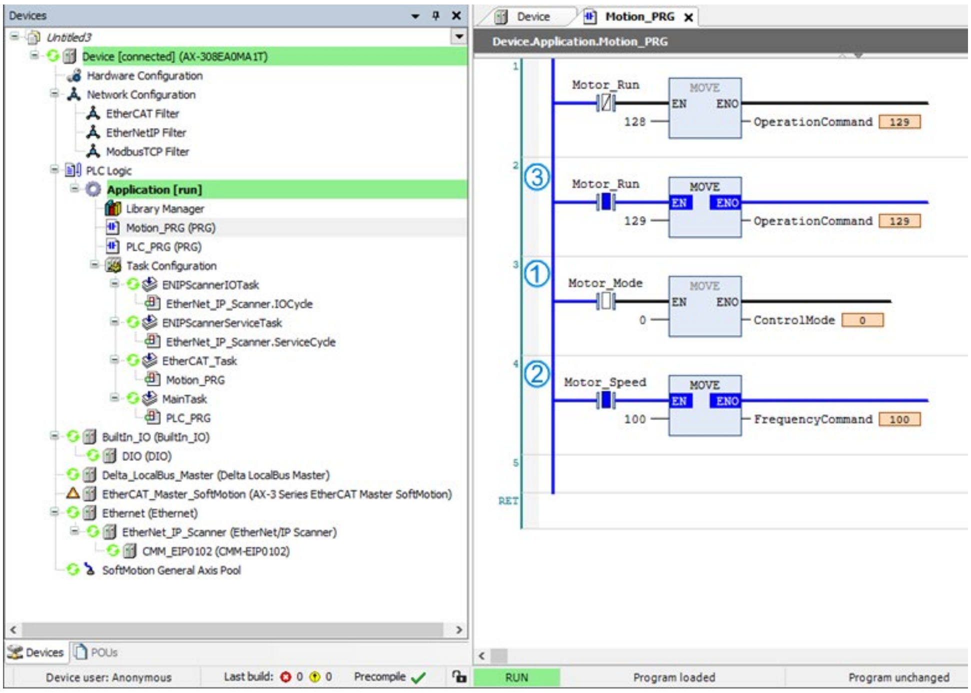


Variable	Mappi...	Channel	Address	Type
		IN_Value_30	%IW30	UINT
		IN_Value_31	%IW31	UINT
		IN_Value_32	%IW32	UINT
		Operation Command 1	%QW1	UINT
		Frequency command 1	%QW2	UINT
		External Command 1	%QW3	UINT
		OUT_Value_4	%QW4	UINT
OperationCommand		Operation Command 2	%QW5	UINT
ControlMode		Control mode 2	%QW6	UINT
FrequencyCommand		Frequency command 2	%QW7	UINT
		Troque limit 2	%QW8	UINT
		Position Command 2, L W	%QW9	UINT
		Position Command 2, H W	%QW10	UINT
		Torque command 2	%QW11	UINT

4. Double-click **Motion_PRG** to add programs as shown below.



5. Implement the following procedure with online monitoring:
   ① Turn on Motor_Mode.
   ② Execute Motor_Speed and write 100 to the speed. (The unit is Hz; value is in two decimal places. For example, write 100 to get 1.00 Hz.)
   ③ Write in 129 to execute Motor_Run, while value 128 is for excitation.



*Note: Information concerning CMM-EIP parameters are detailed in VFD EtherNet/IP Application Manual.

8.4.4.1.3 CIP Object Read-Write Setting for Explicit Messages

Refer to Appendix A <EtherNet/IP Service and Object> in VFD EtherNet/IP Application Manual to check the objects supported by the CMM-EIP communication card and make sure to learn read-write methods for explicit messages before using this function. The master is allowed to configure the setting values of drives directly with the relevant Object Class address. The object class code is 0x300 for drives and the address is formatted as the following shown.

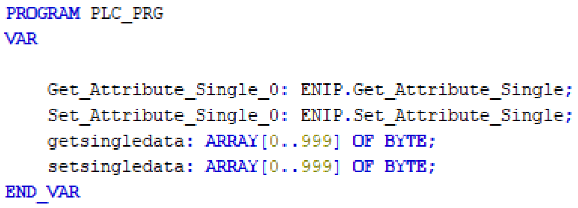
- **EIP communication data format**

Object class		Instance		Attribute
0x300	+	Pr. Group	+	Pr. Number

- **Read-write example**

  To read and write parameter 09-30 (Decoding with EtherNet/IP)

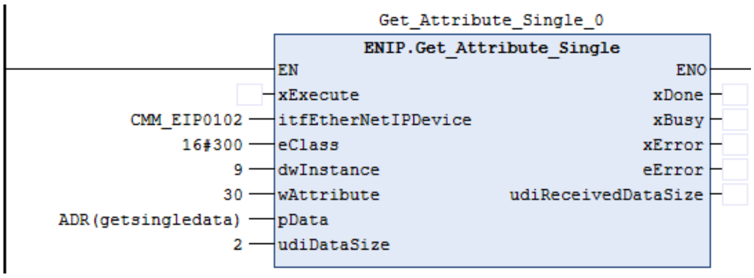
  ■ Declare function blocks and variables.



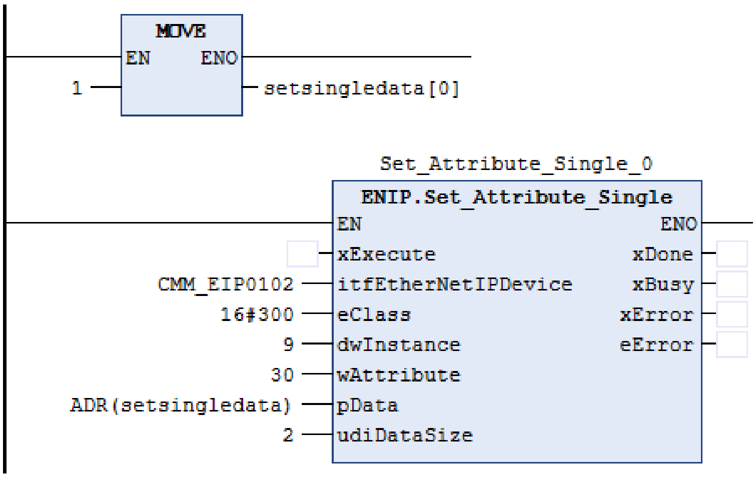
```
PROGRAM PLC_PRG
VAR

    Get_Attribute_Single_0: ENIP.Get_Attribute_Single;
    Set_Attribute_Single_0: ENIP.Set_Attribute_Single;
    getsingledata: ARRAY[0..999] OF BYTE;
    setsingledata: ARRAY[0..999] OF BYTE;
END_VAR
```

  ■ Read parameter 9-30 via the function block as shown below.



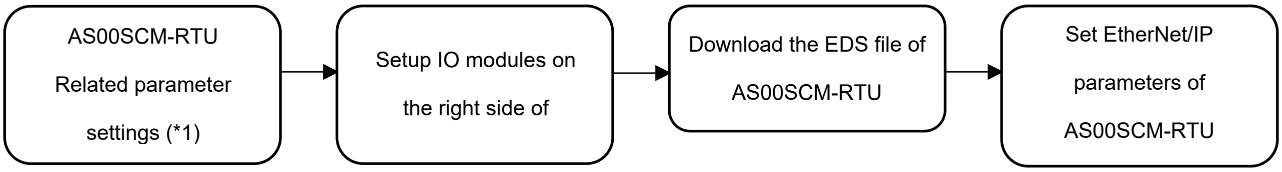
  ■ Write 1 to parameter 9-30 via the function block as shown below.



8.4.4.2 Read-Write to AS00SCM-A (AS-FEN02 Communication Card)

The way to connect AS00SCM-RTU (AS-FEN02) via EtherNet/IP would be explained in this section. Please do read chapter 9 “Serial Communication Module AS00SCM” in AS Series Module Manual to understand the related settings and application of this module before actual operation.

Setup Steps:

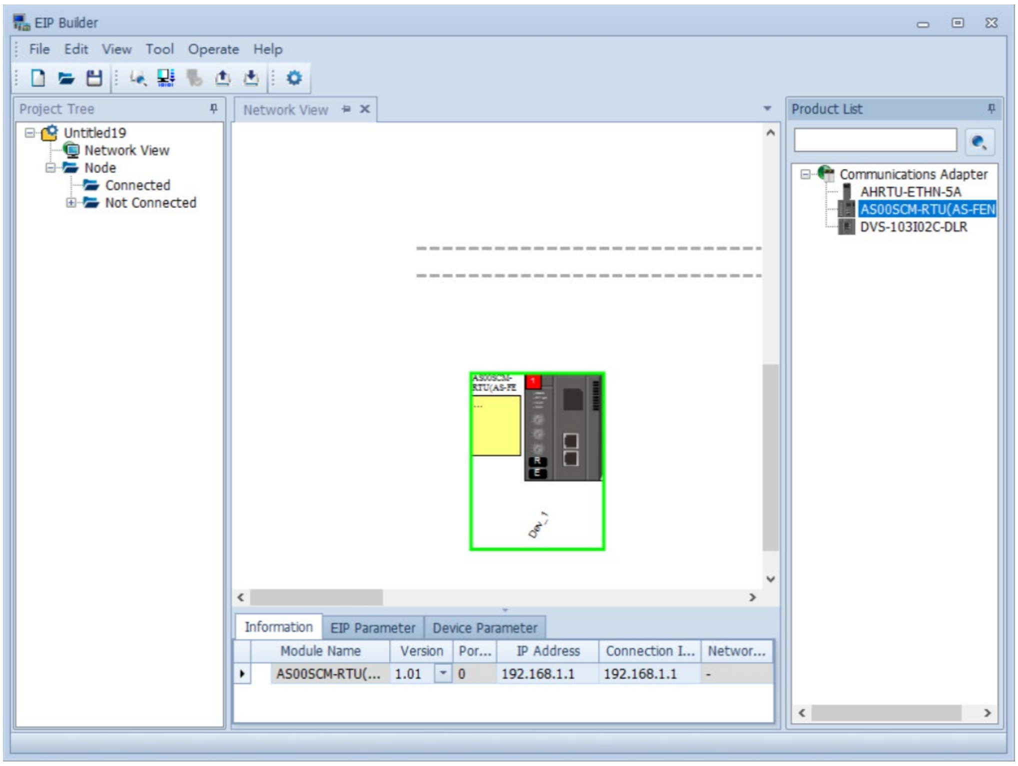


*1: Refer to chapter 9 “Serial Communication Module AS00SCM” in AS Series Module Manual for more details concerning setups of AS00SCM-A IP address and RTU mode.

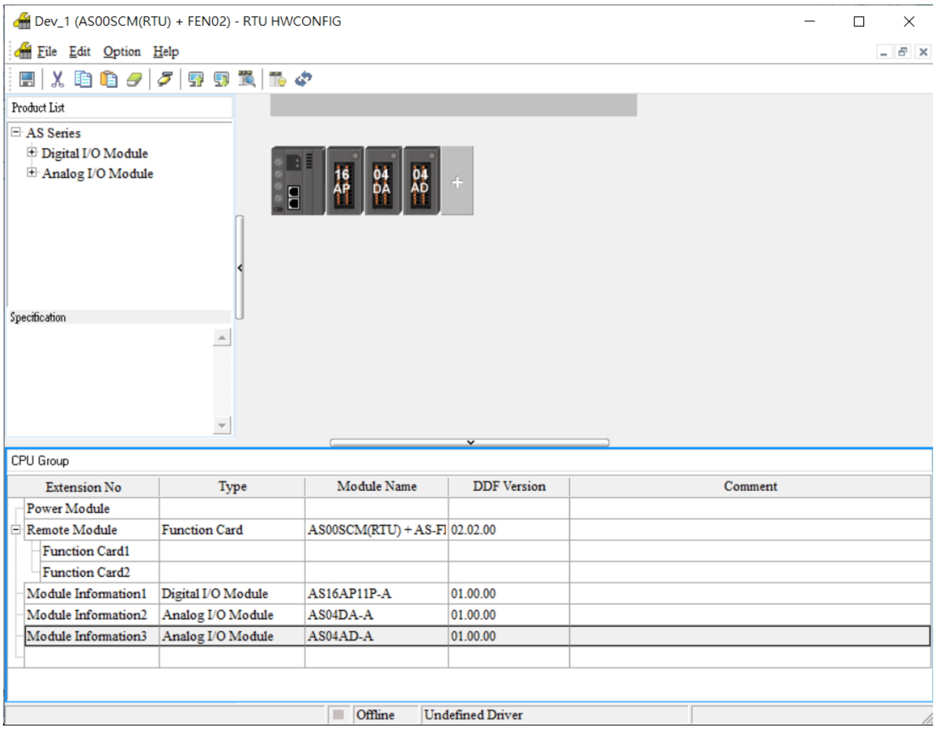
8.4.4.2.1 Setup IO modules on AS00SCM-RTU

Before connecting to AS00SCM-RTU (AS-FEN02), it is necessary to set up the IO modules on the right side of AS00SCM-RTU (AS-FEN02) by using EIP Builder software on your PC.

- Steps to operate the software EIP Builder are shown below.
  1. Add the remote module to the hardware configuration manually or via Scan for Devices. Click on the remote module to open HWCONFIG to scan and download the IO module on the right side.

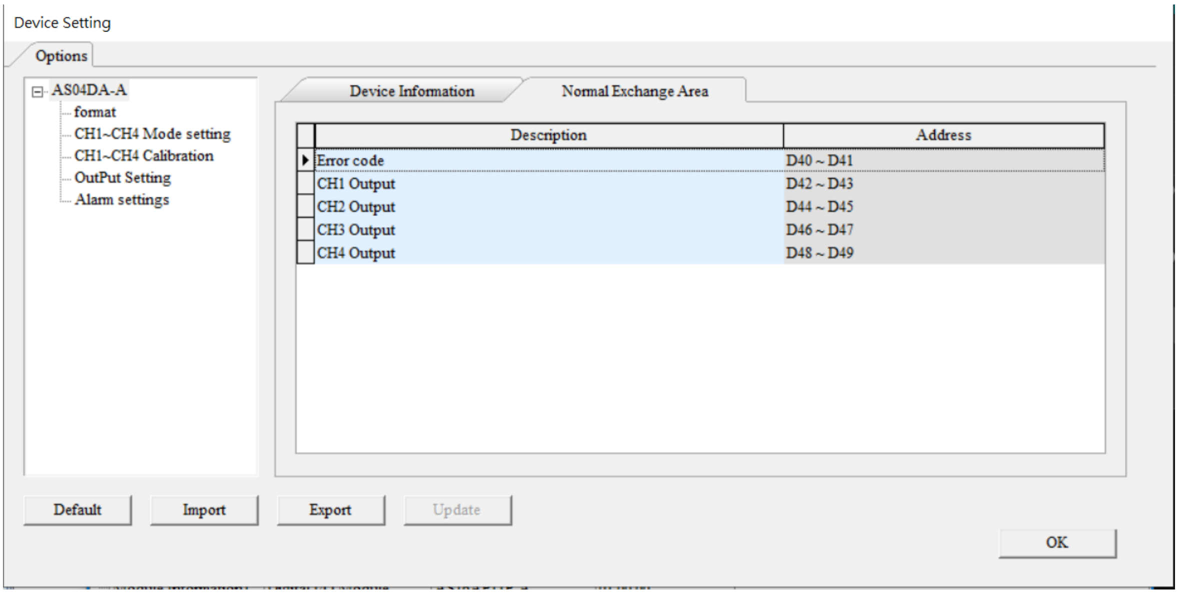


Module Name	Version	Por...	IP Address	Connection I...	Networ...
AS00SCM-RTU(...	1.01	0	192.168.1.1	192.168.1.1	-



Extension No	Type	Module Name	DDF Version	Comment
Power Module				
Remote Module	Function Card	AS00SCM(RTU)+AS-FI	02.02.00	
Function Card1				
Function Card2				
Module Information1	Digital I/O Module	AS16AP11P-A	01.00.00	
Module Information2	Analog I/O Module	AS04DA-A	01.00.00	
Module Information3	Analog I/O Module	AS04AD-A	01.00.00	

2. Data would be exchanged according to the sequence in the Normal Exchange Area on the third-party device. Take AS04DA-A for example, the first input value is an error code (All the error codes of the module are input values, which are defined to be tranmitted from the remote module to the scanner). The data type of the first to the fourth value output from channel 1 to 4 are REAL.



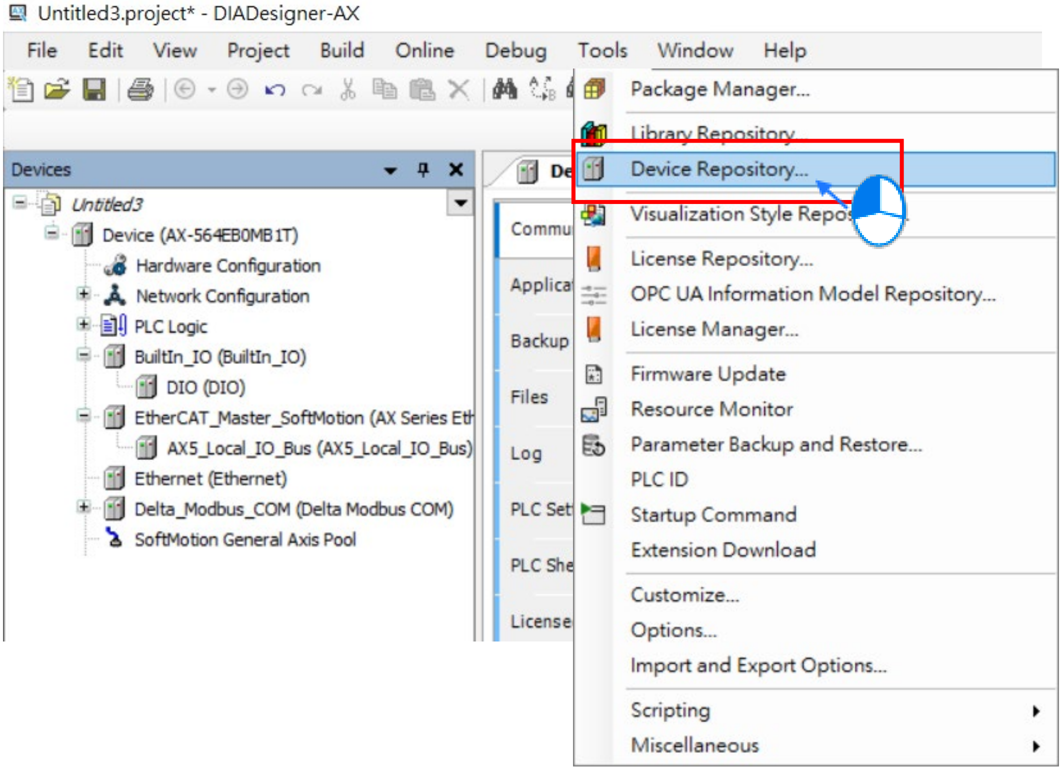
Description	Address
Error code	D40 ~ D41
CH1 Output	D42 ~ D43
CH2 Output	D44 ~ D45
CH3 Output	D46 ~ D47
CH4 Output	D48 ~ D49

8.4.4.2.2 Download the EDS File of ASFEN02 (installed on AS00SCM-RTU)

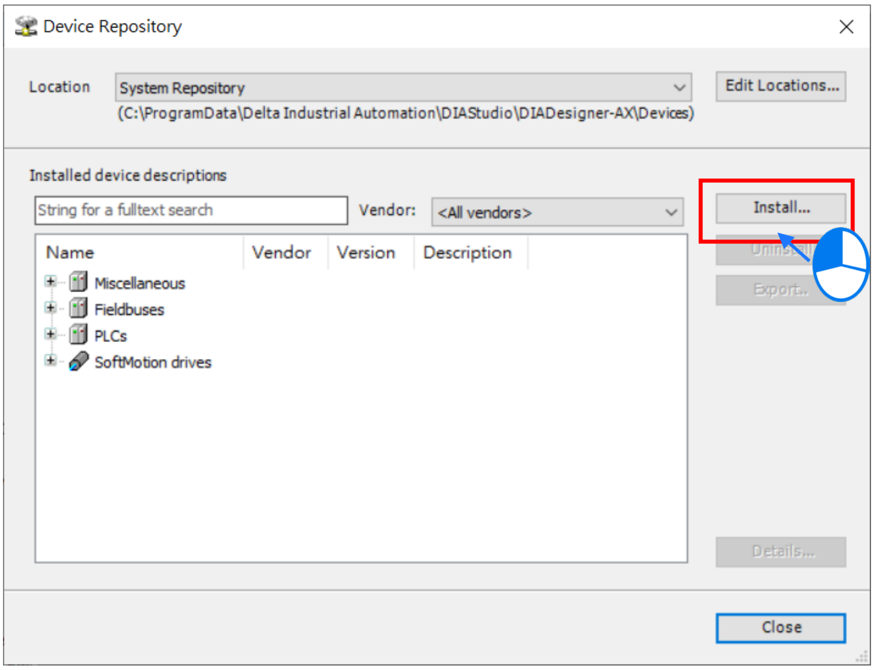
Download the EDS file of ASFEN02 (installed on AS00SCM-RTU) module from Delta's official website.

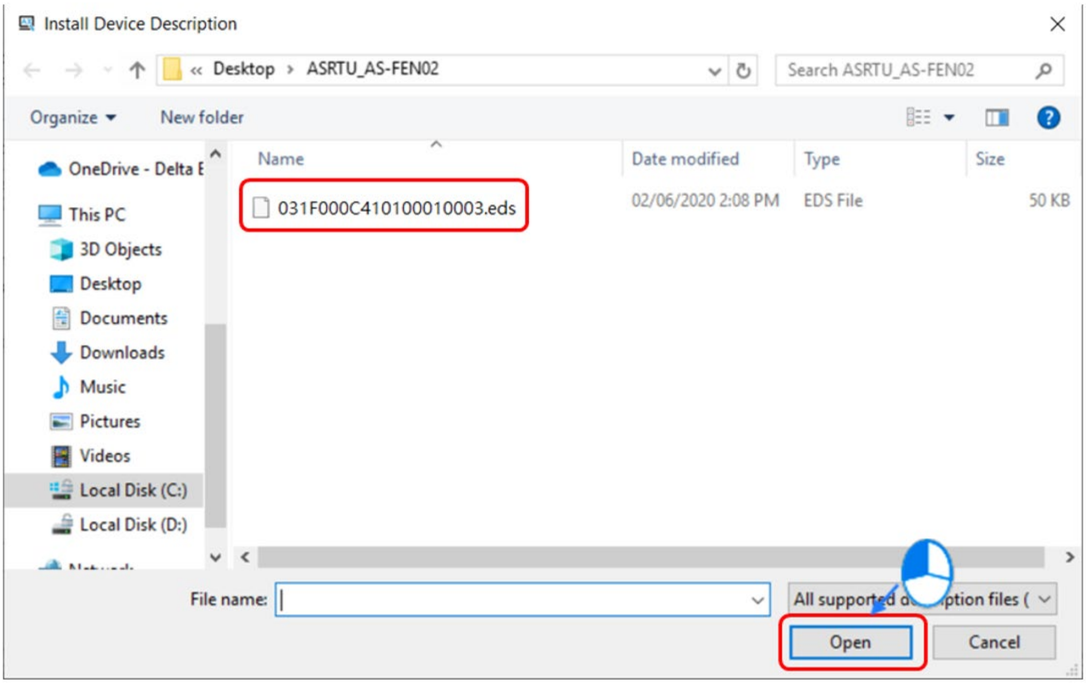
1. Download the EDS file.

1.1 Open Device Repository.



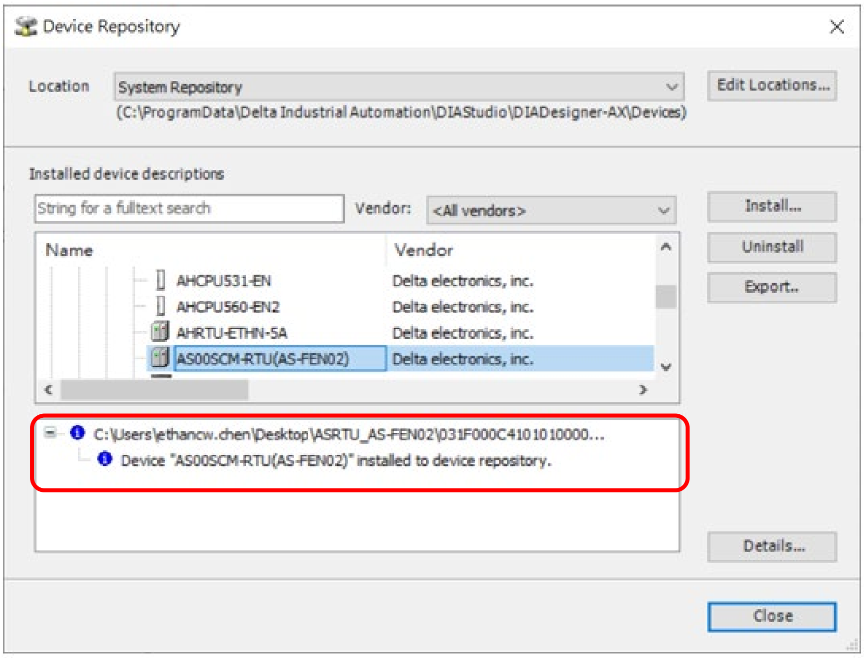
1.2 Choose the target EDS file.





Name	Date modified	Type	Size
031F000C410100010003.eds	02/06/2020 2:08 PM	EDS File	50 KB

1.3 The download is complete.



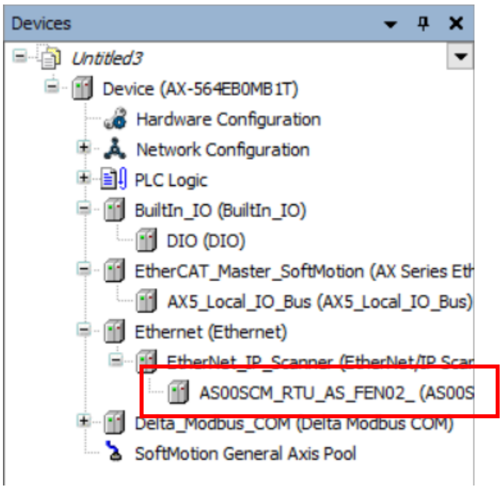
Location: System Repository
(C:\ProgramData\Delta Industrial Automation\DIAStudio\DIADesigner-AX\Devices)

Name	Vendor
AHCPU531-EN	Delta electronics, inc.
AHCPU560-EN2	Delta electronics, inc.
AHRTU-ETHN-5A	Delta electronics, inc.
AS00SCM-RTU(AS-FEN02)	Delta electronics, inc.

C:\Users\ethancw.chen\Desktop\ASRTU_AS-FEN02\031F000C41010010000...

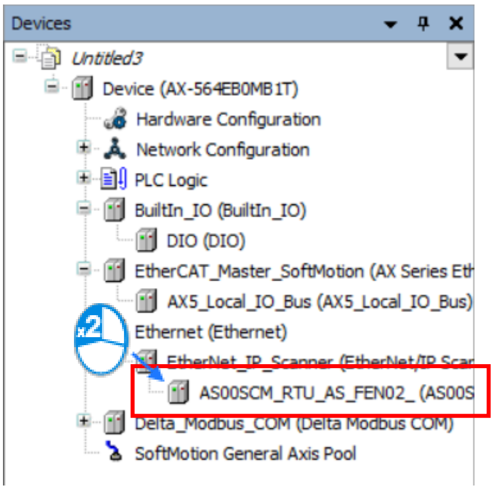
Device "AS00SCM-RTU(AS-FEN02)" installed to device repository.

2. After the download is complete, you can add the AS00SCM-RTU device. Refer to section 8.4.1.1 and 8.4.2.1 for more information on how to establish an Ethernet connection for EtherNet/IP Scanner and EtherNet/IP Adapter (AS00SCM-RTU)

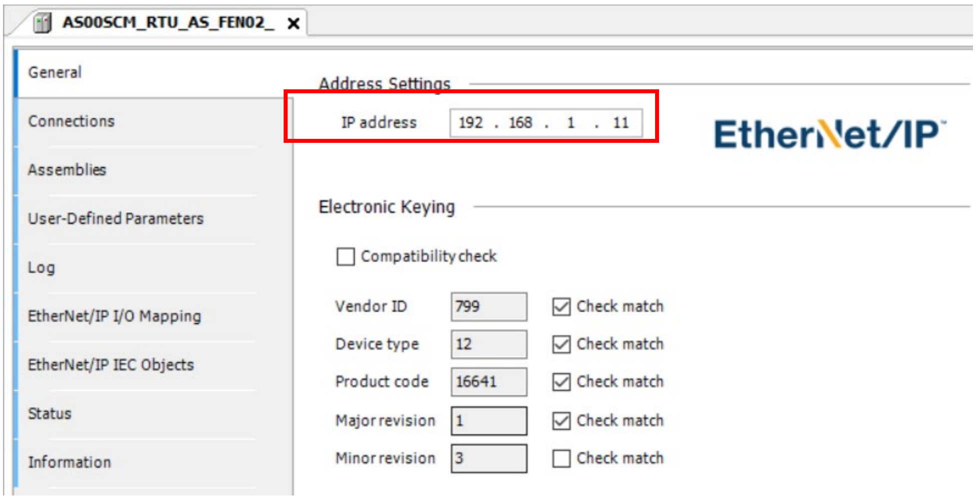


8.4.4.2.3 Configure EtherNet/IP Parameters of AS00SCM-RTU

You are allowed to open the parameter setting page or download the settings from AS00SCM-RTU device so as to start the operation with the IO module.



1. Set the IP address of AS00SCM-RTU.



Address Settings

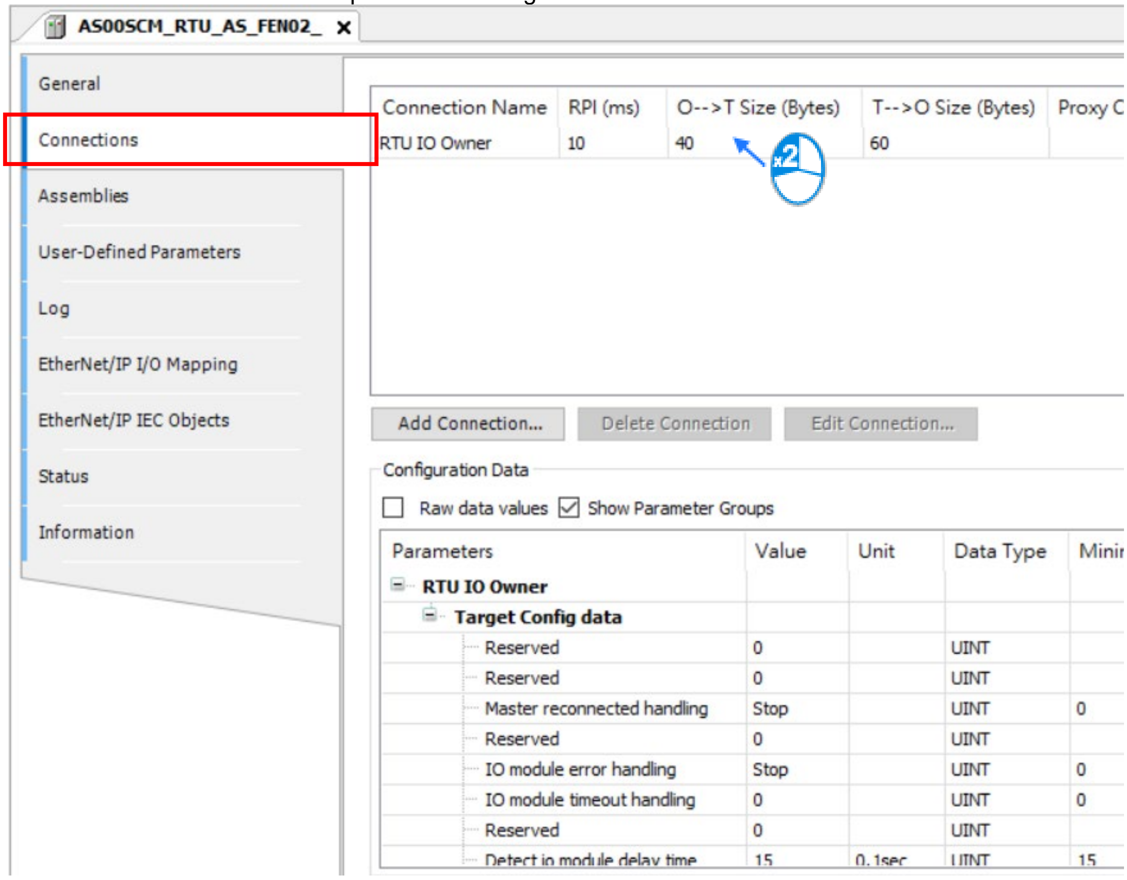
IP address 192 . 168 . 1 . 11

Electronic Keying

☐ Compatibility check

Vendor ID	799	☑ Check match
Device type	12	☑ Check match
Product code	16641	☑ Check match
Major revision	1	☑ Check match
Minor revision	3	☐ Check match

2. Set Connection parameters for EtherNet/IP, which should be configured according to the actual IO module. (Refer to section 8.4.2.2.1 for more details of parameter settings.

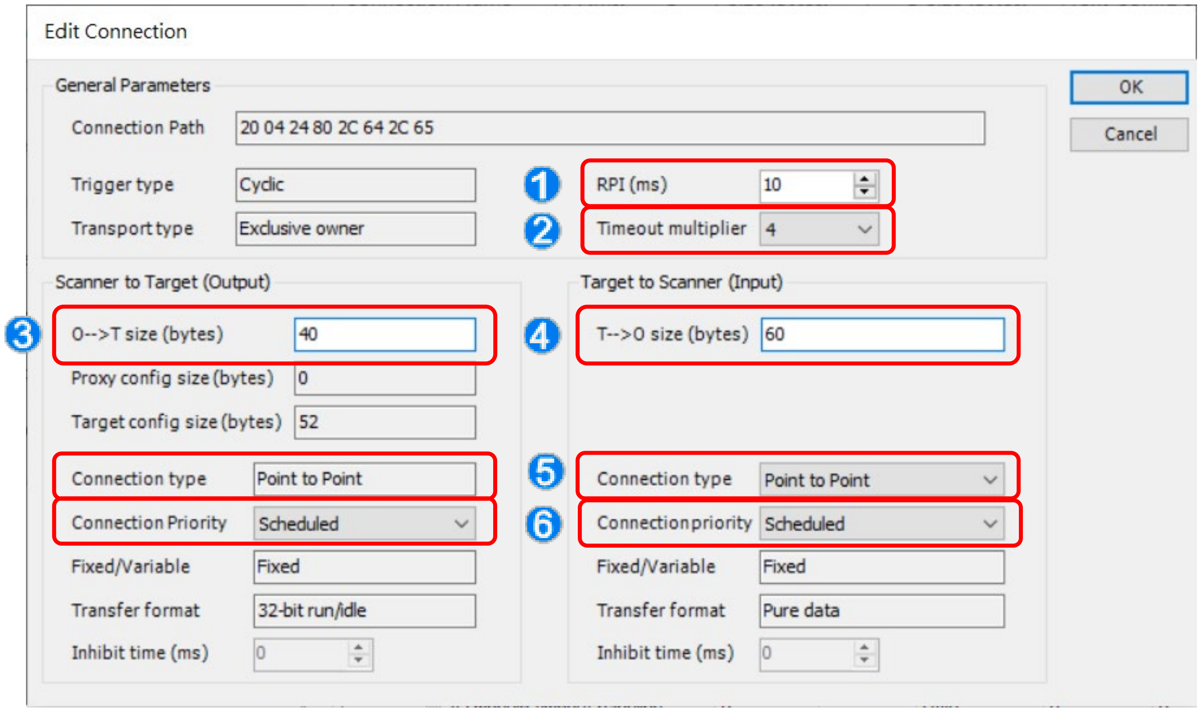


Connection Name	RPI (ms)	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy C
RTU IO Owner	10	40	60	

Configuration Data

☐ Raw data values ☑ Show Parameter Groups

Parameters	Value	Unit	Data Type	Minir
⊟ **RTU IO Owner**				
⊟ **Target Config data**				
Reserved	0		UINT	
Reserved	0		UINT	
Master reconnected handling	Stop		UINT	0
Reserved	0		UINT	
IO module error handling	Stop		UINT	0
IO module timeout handling	0		UINT	0
Reserved	0		UINT	
Detect io module delay time	15	0.1sec	UINT	15



General Parameters

Connection Path: 20 04 24 80 2C 64 2C 65

Trigger type: Cyclic

Transport type: Exclusive owner

RPI (ms): 10

Timeout multiplier: 4

Scanner to Target (Output)

O-->T size (bytes): 40

Proxy config size (bytes): 0

Target config size (bytes): 52

Connection type: Point to Point

Connection Priority: Scheduled

Fixed/Variable: Fixed

Transfer format: 32-bit run/idle

Inhibit time (ms): 0

Target to Scanner (Input)

T-->O size (bytes): 60

Connection type: Point to Point

Connection priority: Scheduled

Fixed/Variable: Fixed

Transfer format: Pure data

Inhibit time (ms): 0

①: RPI: Requested Packet Interval. Connect to EtherNet/IP to exchange data at regular time intervals via the IO connection.

②: Timeout multiplier: Set up the timeout time according to the RPI or the multiple of RPI.

③: O → T size (bytes): The length of the data transmitted from the scanner to the adapter, which is considered to be the output data for the scanner.

④: T → O size (bytes): The length of the data transmitted from the adapter to the scanner, which is considered to be the input data for the scanner.

⑤: Connection type: There are “Point to Point” and “Multicast” modes.

⑥: Connection Priority: The priority of connection. AS00SCM-RTU only supports “Scheduled” mode”.

Note 1: Configure settings of T → O size and O → T size according to the IO module configured in section 8.4.2.2.1. The following table shows the relevant data length of each model type of modules.

- The input/output data length of different DIO modules

Digital I/O Module	T → O size bytes (Input)	O → T size bytes (Output)
AS08AM10N-A	2	0
AS08AN01T-A	0	2
AS08AN01P-A	0	2
AS08AN01R-A	0	2
AS16AM10N-A	2	0
AS16AP11T-A	2	2
AS16AP11P-A	2	2

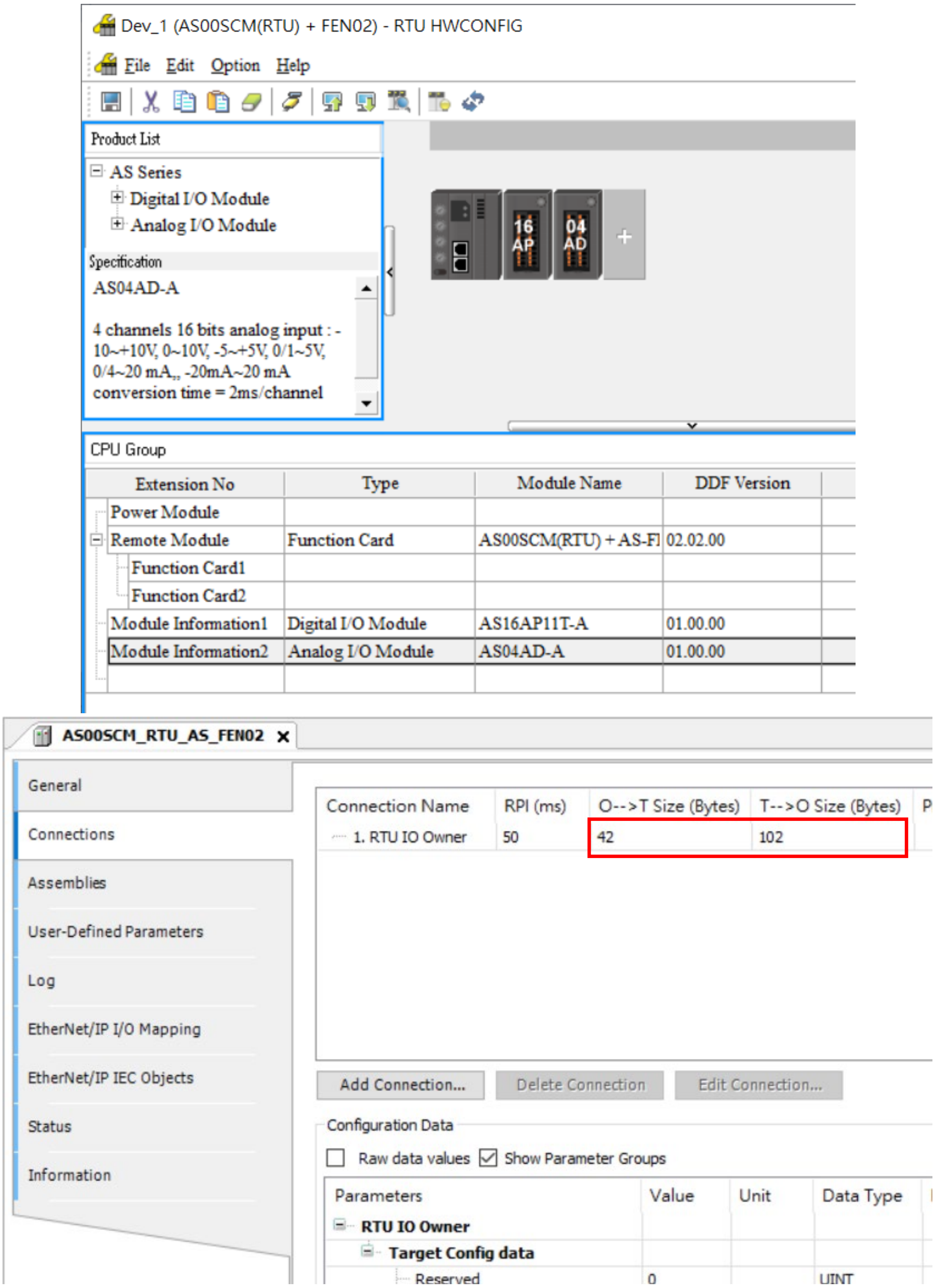
Digital I/O Module	T → O size bytes (Input)	O → T size bytes (Output)
AS16AP11R-A	2	2
AS16AN01T-A	0	2
AS16AN01P-A	0	2
AS16AN01R-A	0	2
AS32AM10N-A	4	0
AS32AN02T-A	0	4
AS64AM10N-A	8	0
AS64AN02T-A	0	8

- The input/output data length of different AIO modules

Analog I/O Module	T → O size bytes (Input)	O → T size bytes (Output)
AS02ADH-A	40	0
AS04AD-A	40	0
AS08AD-B	40	0
AS08AD-C	40	0
AS04DA-A	4	36
AS06XA-A	20	20
AS04RTD-A	40	0
AS06RTD-A	40	0
AS04TC-A	40	0
AS08TC-A	40	0

8.4.4.2.4 Operation of the Right-side IO Modules of AS00SCM-RTU

After the EtherNet/IP connection setting is complete, input and output data can be found on EtherNet/IP IO Mapping tab. Then you would be allowed to operate the IO module on the right side of AS00SCM-RTU. The following configuration shows that AS16AP11T-A (T → O: 2 bytes; O → T: 2 bytes) and AS04AD-A module (T → O: 40 bytes; O → T: 0 bytes) are connected to the right side of AS00SCM-RTU, which the total data length of T → O and O → T respectively are 102 bytes and 42 bytes.



AS04AD-A

4 channels 16 bits analog input : -10~+10V, 0~10V, -5~+5V, 0/1~5V, 0/4~20 mA,, -20mA~20 mA
conversion time = 2ms/channel

Extension No	Type	Module Name	DDF Version
Power Module			
Remote Module	Function Card	AS00SCM(RTU) + AS-F	02.02.00
Function Card1			
Function Card2			
Module Information1	Digital I/O Module	AS16AP11T-A	01.00.00
Module Information2	Analog I/O Module	AS04AD-A	01.00.00

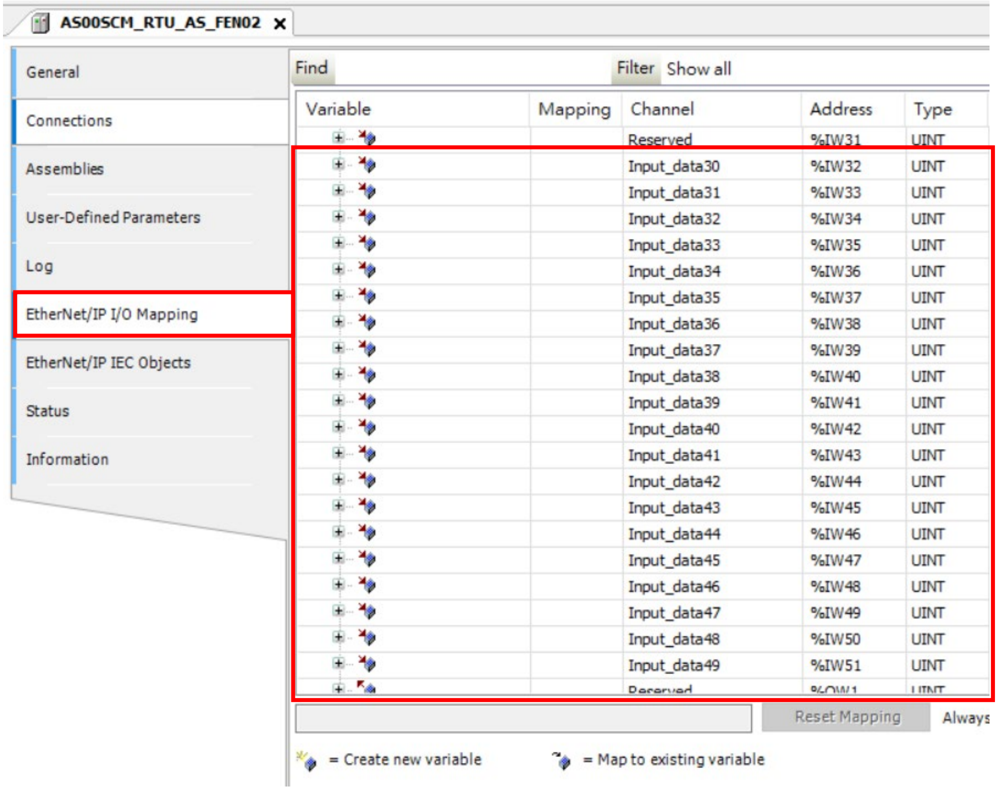
Connection Name	RPI (ms)	O-->T Size (Bytes)	T-->O Size (Bytes)
1. RTU IO Owner	50	42	102

☐ Raw data values ☑ Show Parameter Groups

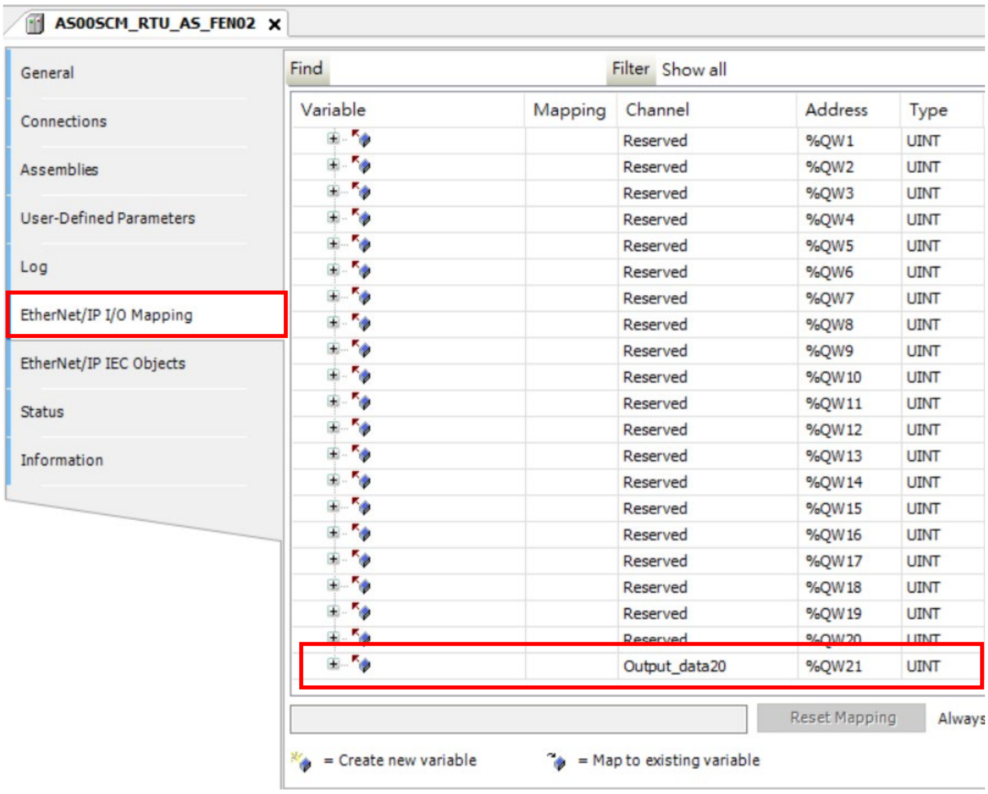
Parameters	Value	Unit	Data Type
RTU IO Owner			
Target Config data			
Reserved	0		UINT

Note 1: Please be noticed that channel mode and other related parameters of AIO modules should be configured first as detailed in section 8.4.2.2.1. Only reading and operating with IO channels would be explained in this section.

Note 2: If the data type of values to read or write is floating point, you would need to exchange the high word and low word so as to display the correct values.



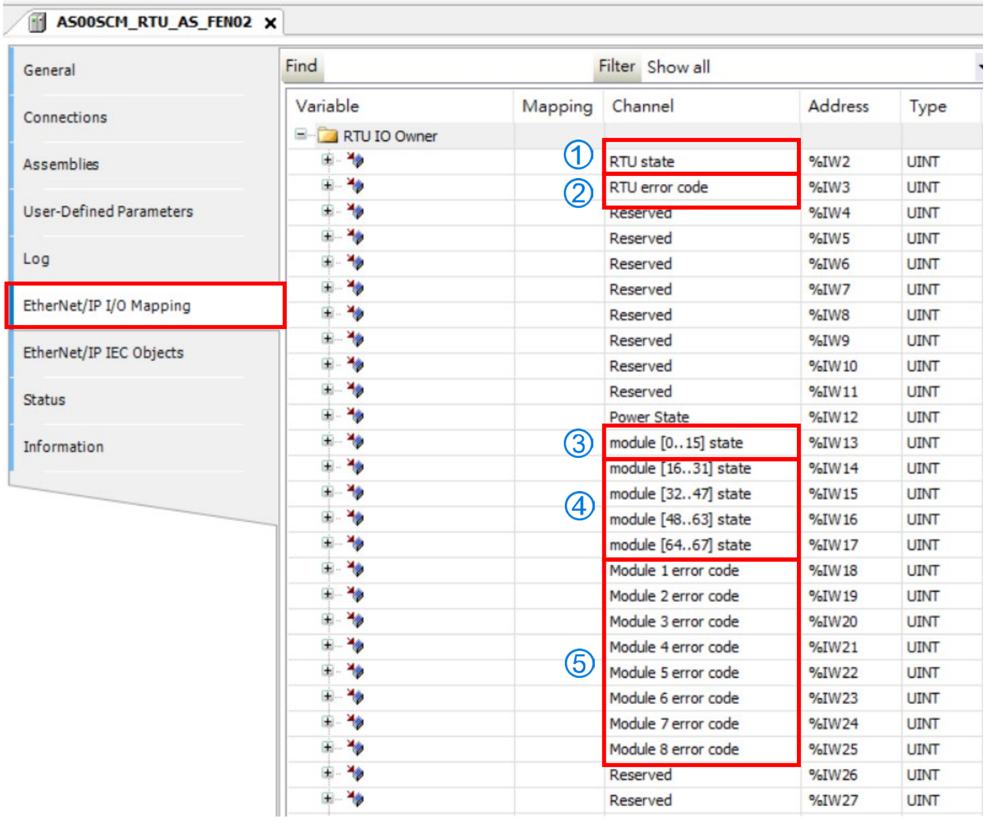
Variable	Mapping	Channel	Address	Type
		Reserved	%IW31	UINT
		Input_data30	%IW32	UINT
		Input_data31	%IW33	UINT
		Input_data32	%IW34	UINT
		Input_data33	%IW35	UINT
		Input_data34	%IW36	UINT
		Input_data35	%IW37	UINT
		Input_data36	%IW38	UINT
		Input_data37	%IW39	UINT
		Input_data38	%IW40	UINT
		Input_data39	%IW41	UINT
		Input_data40	%IW42	UINT
		Input_data41	%IW43	UINT
		Input_data42	%IW44	UINT
		Input_data43	%IW45	UINT
		Input_data44	%IW46	UINT
		Input_data45	%IW47	UINT
		Input_data46	%IW48	UINT
		Input_data47	%IW49	UINT
		Input_data48	%IW50	UINT
		Input_data49	%IW51	UINT
		Reserved	%QW1	UINT



Variable	Mapping	Channel	Address	Type
		Reserved	%QW1	UINT
		Reserved	%QW2	UINT
		Reserved	%QW3	UINT
		Reserved	%QW4	UINT
		Reserved	%QW5	UINT
		Reserved	%QW6	UINT
		Reserved	%QW7	UINT
		Reserved	%QW8	UINT
		Reserved	%QW9	UINT
		Reserved	%QW10	UINT
		Reserved	%QW11	UINT
		Reserved	%QW12	UINT
		Reserved	%QW13	UINT
		Reserved	%QW14	UINT
		Reserved	%QW15	UINT
		Reserved	%QW16	UINT
		Reserved	%QW17	UINT
		Reserved	%QW18	UINT
		Reserved	%QW19	UINT
		Reserved	%QW20	UINT
		Output_data20	%QW21	UINT

8.4.4.2.5 Parameter Information of AS00SCM-RTU Module

The AS00SCM-RTU status can be diagnosed via the parameter information displayed on EtherNet/IP IO Mapping tab.



Variable	Mapping	Channel	Address	Type
RTU IO Owner				
	①	RTU state	%IW2	UINT
	②	RTU error code	%IW3	UINT
		Reserved	%IW4	UINT
		Reserved	%IW5	UINT
		Reserved	%IW6	UINT
		Reserved	%IW7	UINT
		Reserved	%IW8	UINT
		Reserved	%IW9	UINT
		Reserved	%IW10	UINT
		Reserved	%IW11	UINT
		Power State	%IW12	UINT
	③	module [0..15] state	%IW13	UINT
		module [16..31] state	%IW14	UINT
	④	module [32..47] state	%IW15	UINT
		module [48..63] state	%IW16	UINT
		module [64..67] state	%IW17	UINT
		Module 1 error code	%IW18	UINT
		Module 2 error code	%IW19	UINT
		Module 3 error code	%IW20	UINT
		Module 4 error code	%IW21	UINT
	⑤	Module 5 error code	%IW22	UINT
		Module 6 error code	%IW23	UINT
		Module 7 error code	%IW24	UINT
		Module 8 error code	%IW25	UINT
		Reserved	%IW26	UINT
		Reserved	%IW27	UINT

①: RTU state: Communication module status (0 = Normal; 1 = Error)

②: RTU error code: Please refer to section 9.7 Error Codes in AS Series Module Manual.

③: Power State: The power status of communication module. (0 = Normal; 1 = Error)

④: Module state [0..67]: I/O module status, expressed with bits. (0 = Operate normally; 1 = Operate improperly)

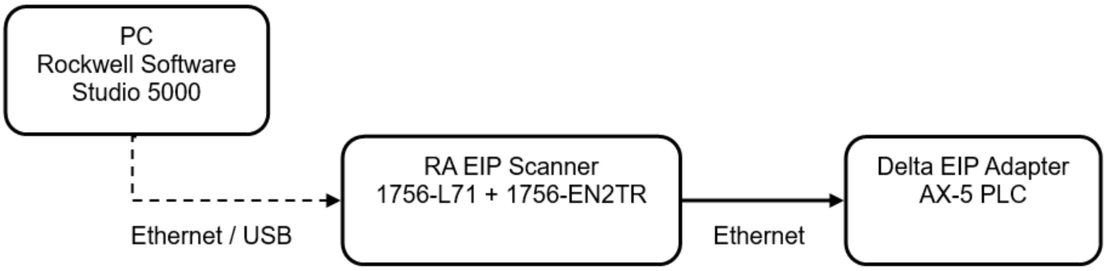
⑤: Module error code: I/O module error codes. For more details of error codes, please refer to the manual of each module.

8.4.5 Example of Connecting to a Third Party (Allen Bradley Controllogix 1756-L71)

This section introduces how to connect Delta's EtherNet/IP adapter via EtherNet/IP by using other brands' software. The Rockwell's software is used as an example in the following section.

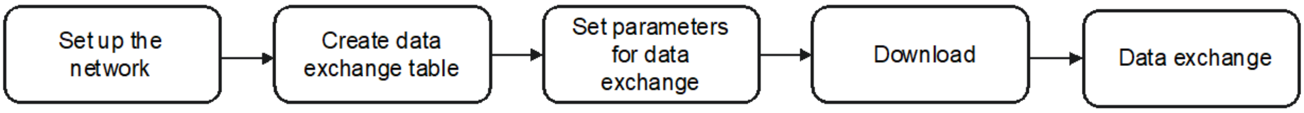
8.4.5.1 Structure

RA EIP scanner connects to Delta's adapter via Ethernet, while connecting to PC via Ethernet or USB.



※ Rockwell Software Studio 5000, ControlLogix, RSLogix are the trademark of Rockwell Automation.

The operation process is shown as follows:



8.4.5.2 Create a Project

- Open Studio 5000 and click "New Project" from "Create".
- Select the model type of PLC. Model 1756-L71 is used in the following example.
- Click "Finish" to finish creating projects.
- The configuration page would be opened automatically after the project has been successfully added.

8.4.5.3 Create a Scanner

After the project being created, add the EtherNet/IP module (1756-EN2TR) on the PLC backplane, then setup the EtherNet/IP device to connect via the EtherNet/IP module.

8.4.5.4 Create a New Module

- Right click on 1756 Backplane 1756-A7 and select "New Module".
- Enter "1756-EN2TR" in the Filter field and select "Create".
- Enter the information of Name and IP address, then click "OK" to complete the task of creating EtherNet/IP modules.
- Expand project tree on the 1756-EN2TR module.

8.4.5.5 Import an EDS File

- Choose EDS Hardware Installation Tool from Tools
- Select “Register an EDS file (s)”.
- Select Browse from Register a single file and find the target EDS file to download: AX-564EB0MB1T.eds。
- Follow the instructions to click “Next” until the EDS file is successfully created.

8.4.5.6 Create a New Adapter

- Right click “Ethernet” and select “New Module” under EtherNet/IP Scanner module in the project tree.
- Enter the module number of the imported EDS file and select the target model type (such as AX-564EB0MB1T), then click “Create”.
- Enter the product name and IP address, which should be same as the information shown in the Module Definition section.
- To change Connections information, click “Change” in Module Definition to open the modification page.
- Change Connections information
  (1) Name: Tap the arrow next to Name to list all the available connections supported by the device.

  (2) Size: the value indicates the length of the input/ output data for data exchange.

※ For general purposes, there is no need to change the parameters from the imported EDS files which often can be used directly for connection.

- On Connection tab page, settings of RPI and input type can be modified, which the former is set as the interval time of periodic data exchange with scanners (unit: ms). Select the input type between Unicast and Multicast according to the feature supported by each product.
- Click OK after the Delta adapter has been successfully added and the model name would be displayed in the project tree.

8.4.5.7 Projects Download

After the creation of the Delta Adapter device is done, download the project to the PLC and go online.

- Click the “Communications” tab to and then select the option “Who Active”. For establishing a connection, select the PC connected Scanner model number and then go to Communications > Download.
- After the connection is successfully established, the I/O status will show OK.

8.4.5.8 Data Mapping

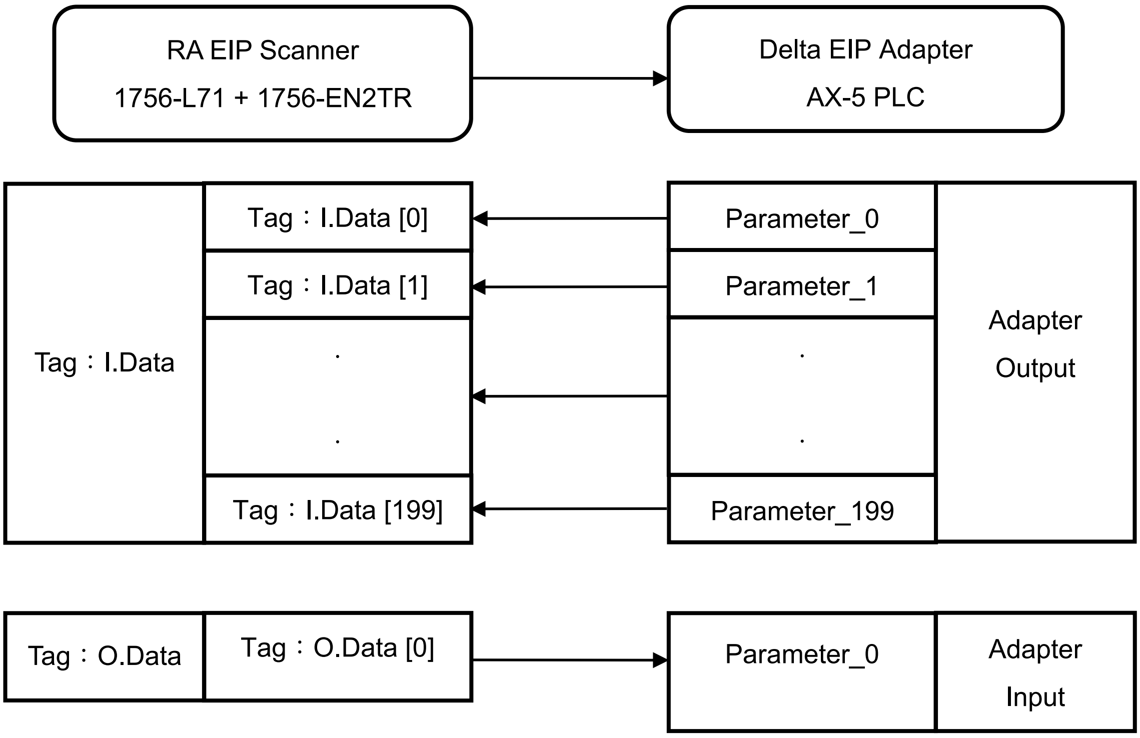
Click the “Program TAGs” under the “Tasks” node for data mapping setups, including Configure, Input and Output. After the device is created in the I/O Configuration, the TAG will be added automatically.

- Click the “Program TAGs”.
- You will see the tags corresponding to each product name on the right-hand side of the window.

TAG: C contains information from Adapter EDS file, including Input and Output parameters. Users can edit the parameters of Input and Output here.

TAG: I1, the mapping starts from TAG: I1[0] and will be mapped to the first parameters of the Adapter Output. The length is the output length provided by the Adapter.

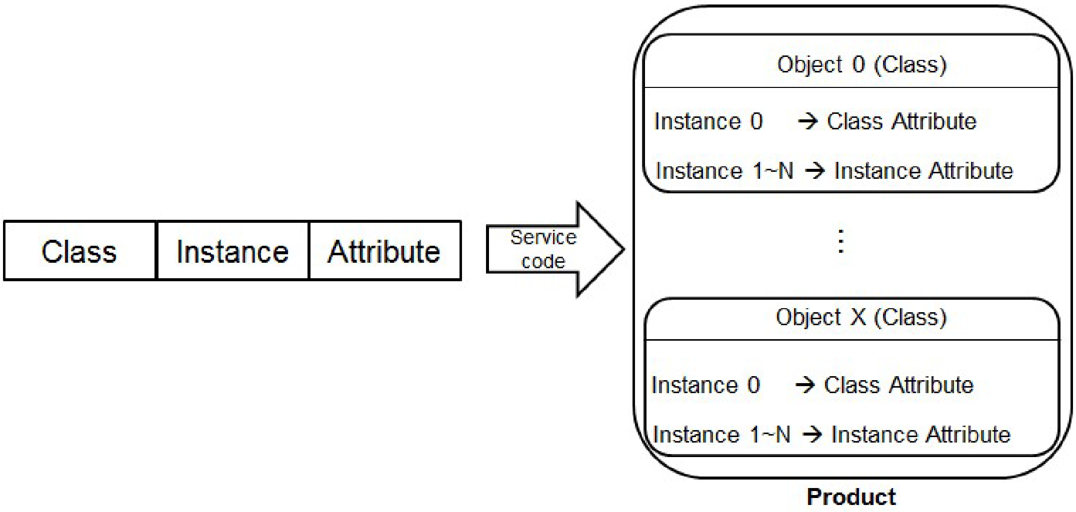
TAG: O1, the mapping starts from TAG: O1[0] and will be mapped to the first parameters of the Adapter Input. The length is the input length provided by the Adapter.


RA EIP Scanner
1756-L71 + 1756-EN2TR
Delta EIP Adapter
AX-5 PLC
Tag：I.Data
Tag：I.Data [0]
Tag：I.Data [1]
.
.
Tag：I.Data [199]
Parameter_0
Parameter_1
.
.
Parameter_199
Adapter
Output
Tag：O.Data
Tag：O.Data [0]
Parameter_0
Adapter
Input

8.4.6 CIP Object

8.4.6.1 Object List

In EtherNet/IP, object is referred to as a set of parameters which are structured accordingly by Class, Instance and Attribute. For example, Instance 0 contains basic information of every object, e.g., version and length. While Instance from 1 to N creates connection or status of required parameters for each product. Users can obtain product parameters from the supported service code via objects (see diagram below).



Read or write objects by using EtherNetIP Services.library or explicit message tool. The supported EtherNet/IP objects are listed below. Refer to the following sections for more information on data type definition and object contents.

Object Name	Function	Class ID
Identity Object	Provides information including manufacturer, device types and versions.	1 (H’01)
Assembly Object	Defines parameter of I/O connection data exchange	4 (H’04)
TCP/IP Interface Object	Displays methods of IP configuration and interface	245 (H’F5)
Ethernet Link Object	Shows the connection status of each Ethernet port on the device.	246 (H’F6)

8.4.6.2 Data Type

This section will provide an overview of the supported data types by objects.

<table>
<tr><th>Data Type</th><th>Description</th></tr>
<tr><td>BOOL</td><td>False (H’00) or True (H’01)</td></tr>
<tr><td>SIGNED INTEGER</td><td>SINT (1 byte), INT (2 bytes), DINT (4 bytes), LINT (8 bytes)
<table>
<tr><th>Number</th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th><th>5th</th><th>6th</th><th>7th</th><th>8th</th></tr>
<tr><td>SINT</td><td>0LSB</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>INT</td><td>0LSB</td><td>1LSB</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>DINT</td><td>0LSB</td><td>1LSB</td><td>2LSB</td><td>3LSB</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>LINT</td><td>0LSB</td><td>1LSB</td><td>2LSB</td><td>3LSB</td><td>4LSB</td><td>5LSB</td><td>6LSB</td><td>7LSB</td></tr>
</table>
Ex: DINT value = H’12345678
<table>
<tr><th>Number</th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th></tr>
<tr><td>DINT</td><td>78</td><td>56</td><td>34</td><td>12</td></tr>
</table>
</td></tr>
<tr><td>UNSIGNED INTEGER</td><td>USINT(1 byte), UINT(2 bytes), UDINT(4 bytes), ULINT(8 bytes)
Ex: UDINT value = H’AABBCCDD
<table>
<tr><th>Number</th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th></tr>
<tr><td>UDINT</td><td>DD</td><td>CC</td><td>BB</td><td>AA</td></tr>
</table>
</td></tr>
<tr><td>STRING</td><td>ASCII, 1 or 2 bytes
STRING: 2 bytes character count + 1 byte character
<table>
<tr><th></th><th colspan="2">Contents (Charcount)</th><th colspan="4">Contents (String contents)</th></tr>
<tr><td>STRING</td><td>04</td><td>00</td><td>4D</td><td>69</td><td>6C</td><td>6C</td></tr>
</table>
STRING2: 2 bytes character count + 2 byte character
<table>
<tr><th></th><th colspan="2">Contents (Charcount)</th><th colspan="8">Contents (String contents)</th></tr>
<tr><td>STRING2</td><td>04</td><td>00</td><td>4D</td><td>00</td><td>69</td><td>00</td><td>6C</td><td>00</td><td>6C</td><td>00</td></tr>
</table>
SHORT_STRING: 1 bytes character count + 1 byte character
<table>
<tr><th></th><th>Contents (Charcount)</th><th colspan="4">Contents (String contents)</th></tr>
<tr><td>STRING</td><td>04</td><td>4D</td><td>69</td><td>6C</td><td>6C</td></tr>
</table>
</td></tr>
<tr><td>Fixed LENGTH BIT STRING</td><td>BYTE(1 byte), WORD(2 bytes), DWORD(4 bytes), LWORD(8 bytes)
<table>
<tr><th></th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th><th>5th</th><th>6th</th><th>7th</th><th>8th</th></tr>
<tr><td>Byte</td><td>7…0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>WORD</td><td>7…0</td><td>15…8</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>DWORD</td><td>7…0</td><td>15…8</td><td>23…16</td><td>31…24</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>LWORD</td><td>7…0</td><td>15…8</td><td>23…16</td><td>31…24</td><td>39…32</td><td>47…40</td><td>55…48</td><td>63…56</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Data Type</th><th>Description</th></tr>
<tr><td>STRING</td><td>A single string consists multiple language representation.
<table>
<tr><th>Name</th><th>Data Type</th><th>Meaning</th></tr>
<tr><td>Number</td><td>USINT</td><td>The number of internationalized character strings</td></tr>
<tr><td>Strings</td><td>Array of:
Struct of:</td><td>Array of individual internationalized character strings</td></tr>
<tr><td>LanguageChar1</td><td>USINT</td><td>The first ASCII character of the ISO 639-2/T language</td></tr>
<tr><td>LanguageChar2</td><td>USINT</td><td>The second ASCII character of the ISO 639-2/T language</td></tr>
<tr><td>LanguageChar3</td><td>USINT</td><td>The third ASCII character of the ISO 639-2/T language</td></tr>
<tr><td>CharStringStruct</td><td>USINT</td><td>The structure of the character string, limited to the Elementary Data type value 0xD0(STRING), 0xD5(STRING2), 0xD9(STRINGN)and 0xDA(SHORT_STRING)</td></tr>
<tr><td>CharSet</td><td>UINT</td><td>The character set which the character string is based on which comes from IANA MIB Printer Code (RFC 1759).</td></tr>
<tr><td>InternationalString</td><td>Defined in CharStringStruct</td><td>An array of 8-bit octet elements which is the actual international character string.</td></tr>
</table>
ISO 639-2/T language:
<table>
<tr><th>Language</th><th>First Character</th><th>Second Character</th><th>Third Character</th></tr>
<tr><td>English</td><td>e</td><td>n</td><td>G</td></tr>
<tr><td>French</td><td>f</td><td>r</td><td>e</td></tr>
<tr><td>Spanish</td><td>s</td><td>p</td><td>a</td></tr>
<tr><td>Italian</td><td>i</td><td>t</td><td>a</td></tr>
</table></td></tr>
<tr><td>STRUCT</td><td>STRUCT of: Any Data Type composes the structure.
Ex.: STRUCT of { BOOL, UINT, DINT } = { TRUE, H’1234, H’56789ABC }
<table>
<tr><th></th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th><th>5th</th><th>6th</th><th>7th</th></tr>
<tr><td>Byte</td><td>01</td><td>34</td><td>12</td><td>BC</td><td>9A</td><td>78</td><td>56</td></tr>
</table></td></tr>
<tr><td>ARRAY</td><td>Array of: Any Data Type composes the array.
Ex.: ARRAY of UINTs = { 1, 2, 3 }
<table>
<tr><th>Number</th><th>1st</th><th>2nd</th><th>3rd</th><th>4th</th><th>5th</th><th>6th</th></tr>
<tr><td>Array</td><td>01</td><td>00</td><td>02</td><td>00</td><td>03</td><td>00</td></tr>
</table></td></tr>
<tr><td>EPATH</td><td>It’s a path that consists of multiple segments and references the class, instance, and attribute of another object.
Ex.: Identity Object, Instance attribute 5 = “ 20 01 24 01 30 05 “</td></tr>
</table>

8.4.6.3 Identity Object (Class ID: 01 Hex)

Identity information is stored in the Identity Object and consists of the Vendor ID, Device Type, Product Code and Major Revision for your device.

- Service Code

Service code	Service Name	Attribute		Description
		Class Attribute	Instance Attribute	
H’01	Get_Attributes_All	X	V	Read all attributes.
H’0E	Get_Attribute_Single	X	V	Read one attribute.

- Class
  - Class ID: H’01
- Instance
  - H’01: Instance Attribute
  - When Instance =1, the Instance attributes are listed below:

Instance Attribute	Name	Access Rule	Data Type	Values	Description
H’01	Vendor ID	Get	UINT	H’31F	Delta Electronics, inc.
H’02	Device Type	Get	UINT	H’0C	Data Type: Communication Adapter
H’03	Product Code	Get	UINT	H’4002	Product code
H’04	Revision	Get	STRUCT	--	Revision of this device: Major. Minor
	Major Revision		USINT	H’01	Major Revision Range: H’01 to H’7F
	Minor Revision		USINT	H’01	Minor Revision Range: H’01 to H’FF
H’05	Status	Get	WORD	H’64	Status, refer to the following※1
H’06	Serial Number	Get	UDINT	H’2374F75C	The last 8 characters of the MAC address 23: 74: f7: 5C
H’07	Product Name	Get	SHORT_STRING	The maximum number of a product name is 32 words. (Data length + Product Name) (H’0D) AX-564EB0MB1T	

※1 Status Description (H’05)

Bit (s)	Name	Description
0	Owned	Display if the device has an owner connection. 0: No 1: Yes
1	Reserved	0: Always OFF
2	Configured	Display if the device is configured or not. 0: No 1: Yes
3	Reserved	0: Always OFF
4 to 7	Extended Device Status	0: Self-Testing 1: Firmware Update 2: At least one faulted I/O connection 3: No I/O connections established 4: Non-Volatile Configuration bad 5: Major Fault 6: At least one I/O connection in run mode 7: At least one I/O connection established, all in idle mode 8 to 15: Reserved
8	Minor Recoverable Fault	0: No minor recoverable fault detected 1: Minor recoverable fault detected
9	Minor Unrecoverable Fault	0: No minor unrecoverable fault detected 1: Minor unrecoverable fault detected
10	Major Recoverable Fault	0: No major recoverable fault detected 1: Major recoverable fault detected
11	Major Unrecoverable Fault	0: No major unrecoverable fault detected 1: Major unrecoverable fault detected

8.4.6.4 Assembly Object (Class ID: 04 Hex)

Assembly Objects are used to aggregate data for the input data and output data associated with I/O connections.

- Service Code

Service Code	Service Name	Support		Description
		Class Attribute	Instance Attribute	
H’0E	Get_Attribute_Single	X	V	Read a single attribute

- Class
  - Class ID: H’04

- Instance
  - H’64：Connection 1 Output assembly
  - H’65：Connection 1 Input assembly
  - ……
  - H’A3：Connection 32 Output assembly
  - H’A4：Connection 33 Input assembly
  - When Instance = 64 to A4, the Instance Attributes are listed below:

Instance Attribute	Name	Access Rule	Data Type	Values	Description
H’03	Data	Get	ARRAY of BYTE	H’2	IO Connection Data

- Examples of reading and writing objects

(1) To read output assembly data, write the data as shown below:

Service code: H’ 0E

Class ID: H’ 04

Instance ID: H’ 64

Attribute ID: H’ 03

(2) To read input assembly data, write the data as shown below:

Service code: H’ 0E

Class ID: H’ 04

Instance ID: H’ 65

Attribute ID: H’ 03

8.4.6.5 TCP/IP Interface Object (Class ID: F5 Hex)

- Service Code

Service Code	Service Name	Support		Description
		Class Attribute	Instance Attribute	
H’0E	Get_Attribute_Single	V	V	Read a single attribute
H’10	Set_Attribute_Single	X	V	Set values of a single attribute

- Class
  - Class ID = H’F5
- Instance
  - H’00：Class Attribute
  - H’01：Instance Attribute
  - When Instance = 0, the class attributes are listed below:

Class Attribute	Name	Access Rule	Data Type	Values	Description
H’01	Revision	Get	UINT	H’4	Object revision

  - When Instance =1, the Instance attributes are listed below:

Instance Attribute	Name	Access Rule	Data Type	Values	Description
H’01	Status	Get	DWORD	H’2	IP status ※1
H’02	Configuration Capability	Get	DWORD	H’20	Configuration capability, refer to the following ※2
H’03	Configuration Control	Get/Set	DWORD	H’0	Configuration Control, refer to the following ※3
H’04	Physical Link Object：	Get	STRUCT of	--	Path to physical link object
	Path Size		UINT	H’0	Size of Path
	Path		EPATH	--	Logical segments identifying the physical link object
H’05	Interface Configuration：	Get/Set	STRUCT of	--	TCP/IP network interface configuration.
	IP Address		UDINT	192.168.1.5	The device’s IP address
	Network Mask		UDINT	255.255.255.0	The device’s network mask:
	Gateway Address		UDINT	0	Default gateway address
	Name Server		UDINT	0	Primary name server
	Name Server 2		UDINT	0	Secondary name server
	Domain Name		STRING	00 00	Default domain name
H’06	Host Name	Get	STRING	AX-564EB0MB1T	Device name
H’0A	SelectAcd	Get/Set	BOOL	Enable	Activates the use of ACD

H'0B	LastConflictDetected	Get	STRUCT of	0 (NoConflictDetected)	Structure containing information related to the last conflict detected
	AcdActivity	Get	USINT	0	State of ACD activity when last conflict detected
	RemoteMAC	Get	Array of 6 USIN	0	MAC address of remote node from the ARP PDU in which a conflict was detected
	ArpPdu	Get	ARRAY of 28 USINT	0	Copy of the raw ARP PDU in which a conflict was detected

※ When the master is communicating, the instance attribute H'03 and H'05 cannot be written.

- Examples of reading objects

(1) To read Instance Attribute H'03, write the data as shown below:

Service code：H'0E

Class ID：H'F5

Instance ID：H'01

Attribute ID：H'03

※1 Interface status

Status	Description
0	Interface Configuration attribute has not been configured.
1	The Interface Configuration attribute contains valid configuration obtained from BOOTP, DHCP or non-volatile memory.
2	The Interface Configuration attribute contains valid configuration obtained from hardware.

※2 Interface capability flags

Bit	Description
0	BOOTP Client
1	DNS Client
2	DHCP Client
3	DHCP-DNS Update
4	Configuration Settable
5	Hardware Configurable
6	Interface Configuration Change Requires Reset

※3 Interface Configuration Control

Status	Description
0	The device shall use the interface configuration values previously stored (for example, in non-volatile memory or via hardware witches).
1	The device shall obtain its interface configuration values via BOOTP.
2	The device shall obtain its interface configuration values via DHCP upon start-up.

8.4.6.6 Ethernet Link Object (Class ID: F6 Hex）

- Service Code

Service Code	Service Name	Support		Description
		Class Attribute	Instance Attribute	
H’01	Get_Attributes_All	V	X	Read multiple attributes
H’0E	Get_Attribute_Single	V	V	Read a single attribute
H’4C	Get_and_Clear	X	V	Read the attribute and then reset it.

- Class
  - Class ID：H’F6
- Instance
  - H’00：Class Attribute
  - H’01：Instance Attribute

  - When Instance =0, the Instance attributes are listed below:

Class Attribute	Name	Access Rule	Data Type	Values	Description
H’01	Revision	Get	UINT	H’04	Object revision

  - When Instance =1, the Instance attributes are listed below:

Instance Attribute	Name	Access Rule	Data Type	Values	Description
H’01	Interface Speed	Get	DWORD	0	Interface speed (indeterminate)
H’02	Interface Flags	Get	DWORD	H’0F	Ethernet port status, refer to the following※1
H’03	Physical Address	Get	ARRAY of 6 USINTs	By Product	MAC address
H’04	Interface Counters	Get	STRUCT of:	0	Interface Counters
	In Octets		UDINT	0	Octets received on the interface
	In Ucast Packets		UDINT	0	Unicast packets received on the interface
	In NUcast Packets		UDINT	0	Non-unicast packets received on the interface
	In Discards		UDINT	0	Inbound packets received on the interface but discarded
	In Errors		UDINT	0	Inbound packets that contain errors (does not include In

Instance Attribute	Name	Access Rule	Data Type	Values	Description
					Discards)
	In Unknown Protos		UDINT	0	Inbound packets with unknown protocol
	Out Octets		UDINT	0	Octets sent on the interface
	Out Ucast Packets		UDINT	0	Unicast packets sent on the interface
	Out NUcast Packets		UDINT	0	Non-unicast packets sent on the interface
	Out Discards		UDINT	0	Outbound packets discarded
	Out Errors		UDINT	0	Outbound packets that contain errors
H’05	Media Counters	Get	STRUCT of:	0	Media-specific counters
	Alignment Errors		UDINT	0	Frames received that are not an integral number of octets in length.
	FCS Errors		UDINT	0	Frames received that do not pass the FCS check.
	Single Collisions		UDINT	0	Successfully transmitted frames which experienced exactly one collision.
	Multiple Collisions		UDINT	0	Successfully transmitted frames which experienced more than one collision.
	SQE Test Errors		UDINT	0	Number of times SQE test error message is generated.
	Deferred Transmissions		UDINT	0	Frames for which first transmission attempt is delayed because the medium is busy.
	Late Collisions		UDINT	0	Number of times a collision is detected later than 512 bit-times into the transmission of a packet.
	Excessive Collisions		UDINT	0	Frames for which transmission fails due to excessive collisions.
	MAC Transmit Errors		UDINT	0	Frames for which transmission fails due to an internal MAC sublayer transmit error.

Instance Attribute	Name	Access Rule	Data Type	Values	Description
	Carrier Sense Errors		UDINT	0	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame.
	Frame Too Long		UDINT	0	Frames received that exceed the maximum permitted frame size MAC Receive Errors.
H’0B	Interface Capability	Get	STRUCT of :	--	Capabilities of Ethernet interface ※2
	Capability Bits		DWORD	H’02000000	The definition of Ethernet interface capability.
	Speed/Duplex Options		STRUCT of :	--	The definition of speed and duplex options of Ethernet interface.
	Speed/Duplex Array Count		USINT	H’00	The count of speed/ duplex options.
	Speed/Duplex Array		ARRAY of STRUCT of :	--	Speed and duplex settings
	Interface Speed		UINT	H’00	Ethernet interface speed. For example, 10 bps and 100 bps would be H’0A and H’64 accordingly.
	Interface Duplex Mode		USINT	H’00	Duplex mode capability of Ethernet interface. For example, half and full duplex would be H’00 and H’01 accordingly.

※1 Interface Flag Table

Bit (s)	Name	Description
0	Link Status	0 indicates an inactive link 1 indicates an active link
1	Half/Full Duplex	0 indicates half duplex 1 indicates full duplex
2 to 4	Negotiation Status	0：Auto-negotiation in progress 1：Auto-negotiation and speed detection failed 2：Auto negotiation failed but detected speed 3：Successfully negotiated speed and duplex 4：Auto-negotiation not attempted. Forced speed and duplex.
5	Manual Setting Requires Reset	shall be set zero
6	Local Hardware Fault	0 indicates the interface detects no local hardware fault 1 indicates a local hardware fault is detected

Bit (s)	Name	Description
7-31	Reserved	0

※2 Interface Capability Bits

Bit (s)	Name	Description
0	Manual Setting Requires Reset	Indicates whether or not the device requires a reset when instance attribute #6 (Interface Control attribute) changes. 0 indicates the device does not require a reset. 1 indicates the device requires a rest.
1	Auto-negotiate	0 indicates the interface does not support auto-negotiation. 1 indicates the interface supports auto-negotiation.
2	Auto-MDIX	0 indicates the interface does not support auto MDIX operation. 1 indicates the interface supports auto MDIX operation.
3	Manual Speed/Duplex	0 indicates the interface does not support to set speed/duplex. (Instance attribute #6, Interface Control attribute) 1 indicates the interface supports to set speed/duplex.
4 to 31	Reserved	shall be set to 0.

8.5 PROFINET IO

PROFINET IO is a fieldbus protocol that enables communication between programmable controllers and distributed field devices in Ethernet. With specific service sets, devices are classified into IO Controllers, IO Supervisors, and IO Devices. PROFINET IO uses three different communication channels to exchange data: Standard UDP/IP and TCP/IP channels, Real-time (RT) channels and Instant Real-time channels.

The standard UDP/IP and TCP/IP channels are used for parameterization and configuration of devices and non-cyclical operations; the Real-Time (RT) channel is used for cyclic data transmission and alarms, while the Instant Real-Time (IRT) channel is used for motion control applications.

Note: This function is only available for the following devices and software versions.

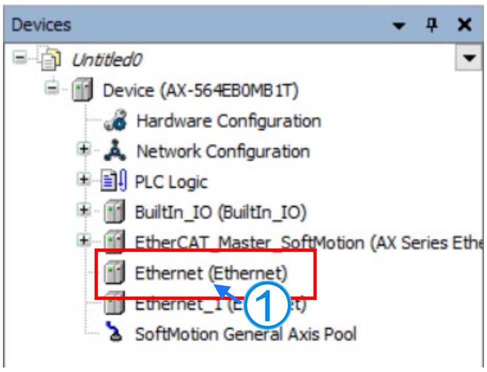
- DIADeigner-AX: V1.4.0 or later
- AX-3 Series PLC CPU: V1.0.5.0 or later
- AX-5 Series PLC CPU: V1.0.0.0 or later
- Library “IoDrvProfinetDevice”: V4.3.0.0 or later

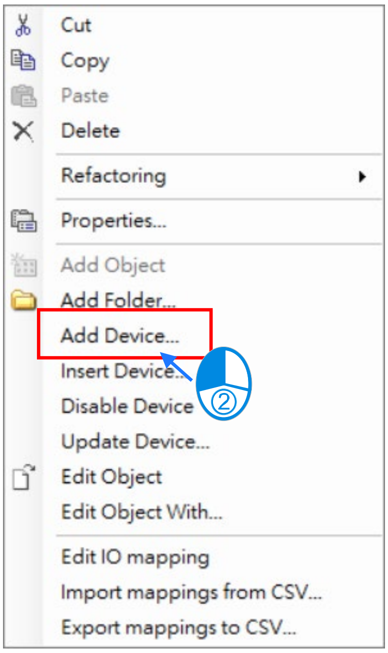
8.5.1 PROFINET IO Controller Function

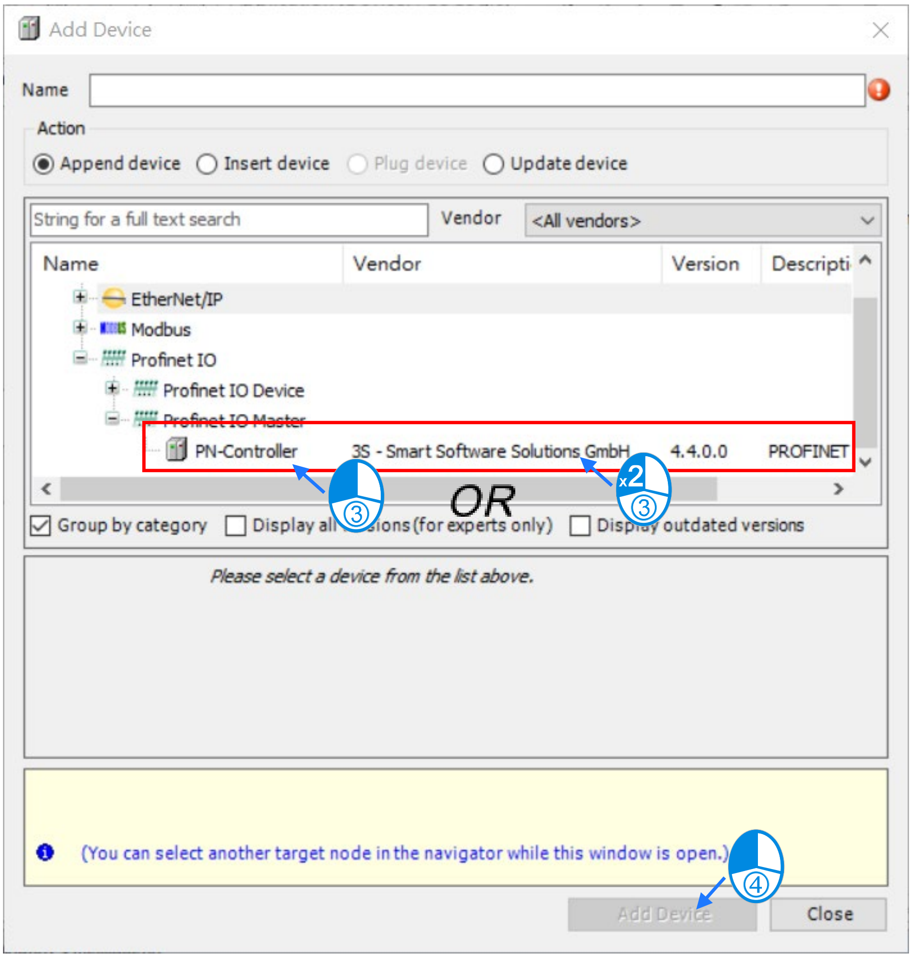
8.5.1.1 Add a PROFINET IO Controller

This section will explain how to use AX-5 Series PLC as a PROFINET IO Controller. Follow the steps below to set up.

- **Adding a PROFINET IO Controller**



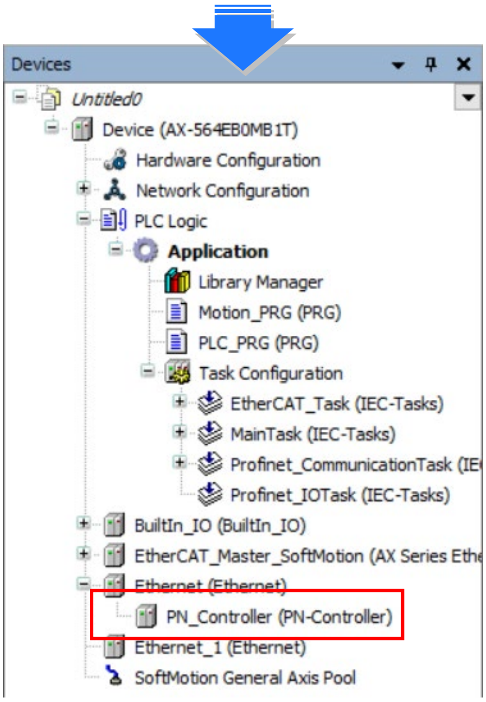




Name	Vendor	Version	Descripti
EtherNet/IP			
Modbus			
Profinet IO			
Profinet IO Device			
Profinet IO Master			
PN-Controller	3S - Smart Software Solutions GmbH	4.4.0.0	PROFINET

Please select a device from the list above.

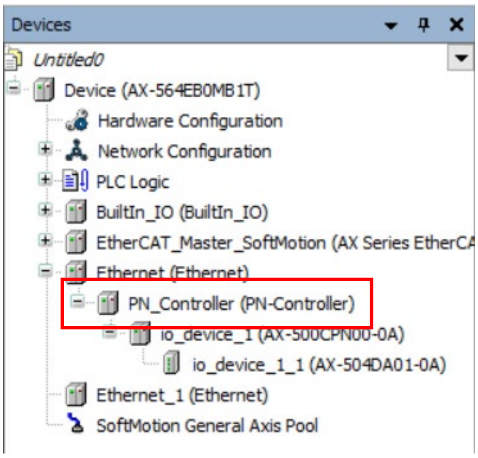
(You can select another target node in the navigator while this window is open.)



- *Untitled0*
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
      - **Application**
        - Library Manager
        - Motion_PRG (PRG)
        - PLC_PRG (PRG)
        - Task Configuration
          - EtherCAT_Task (IEC-Tasks)
          - MainTask (IEC-Tasks)
          - Profinet_CommunicationTask (IE
          - Profinet_IOTask (IEC-Tasks)
    - BuiltIn_IO (BuiltIn_IO)
    - EtherCAT_Master_SoftMotion (AX Series Ethe
    - Ethernet (Ethernet)
      - PN_Controller (PN-Controller)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

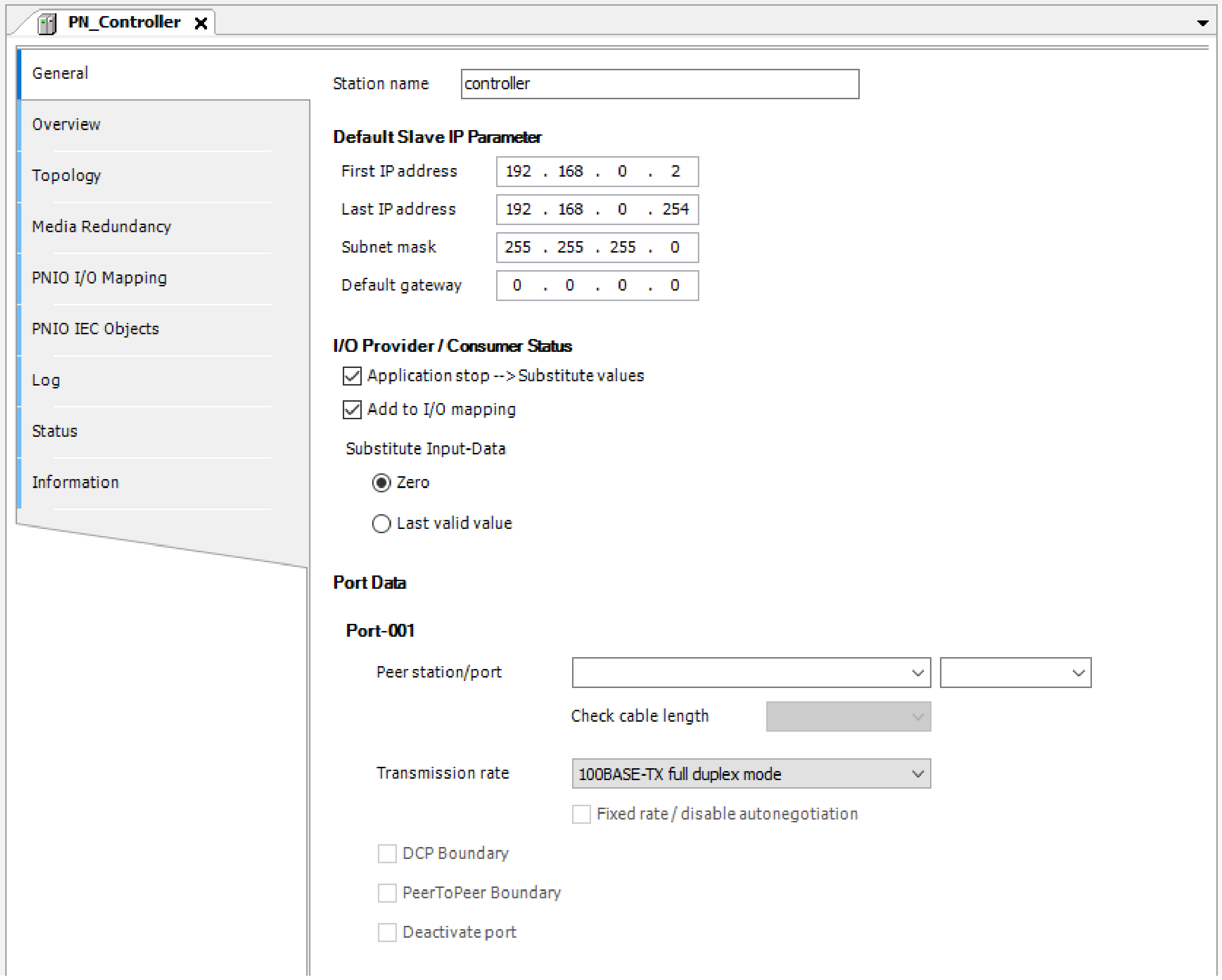
8.5.1.2 Settings of PROFINET IO Controller

This section introduces the PROFINET Controller setting pages.



- **General**

Define the station name and IP address and for the PROFINET Device here.



Station name: controller

Default Slave IP Parameter

First IP address: 192 . 168 . 0 . 2
Last IP address: 192 . 168 . 0 . 254
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 0 . 0 . 0 . 0

I/O Provider / Consumer Status

☑ Application stop --> Substitute values
☑ Add to I/O mapping

Substitute Input-Data
● Zero
○ Last valid value

Port Data

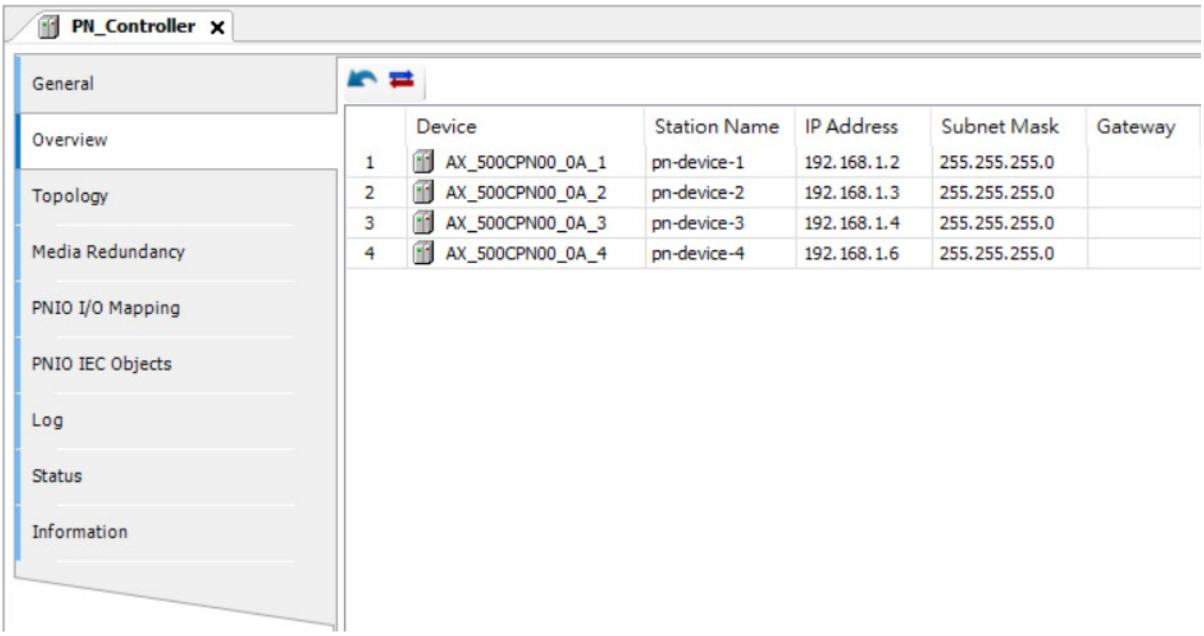
Port-001

Peer station/port:
Check cable length:
Transmission rate: 100BASE-TX full duplex mode
☐ Fixed rate / disable autonegotiation
☐ DCP Boundary
☐ PeerToPeer Boundary
☐ Deactivate port

Item	Description
Station Name	Station name of the Profinet controller
Default Slave IP Parameter	The Profinet IO Device will be assigned an IP address based on the specified IP address range when the "Auto-IP" function is used in the "Scan for Devices" feature.
I/O Provider / Consumer Status	Application stop → Substitute values: when the PLC CPU is in Stop state, the input data is zero. Substitute Input-Data: zero or last valid value
Port Data	Network topology of the Profinet IO controller

- **Overview**

Define the station name and IP address for the slaves of the PROFINET Device here.

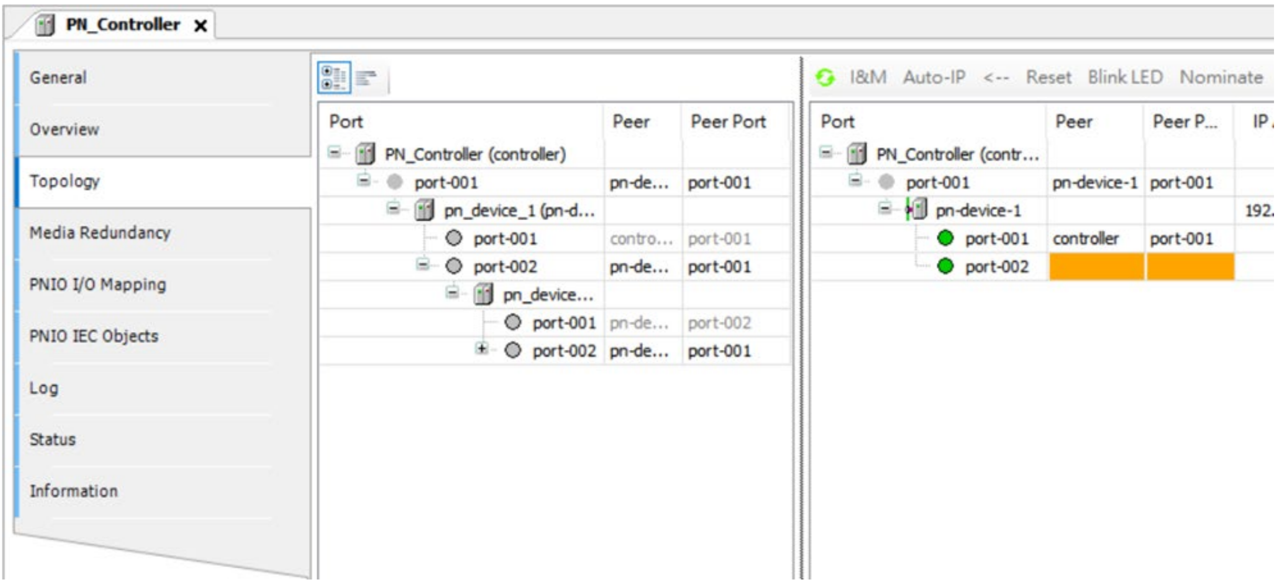


	Device	Station Name	IP Address	Subnet Mask	Gateway
1	AX_500CPN00_0A_1	pn-device-1	192.168.1.2	255.255.255.0	
2	AX_500CPN00_0A_2	pn-device-2	192.168.1.3	255.255.255.0	
3	AX_500CPN00_0A_3	pn-device-3	192.168.1.4	255.255.255.0	
4	AX_500CPN00_0A_4	pn-device-4	192.168.1.6	255.255.255.0	

Button	Description
↶	Modify the frequency of data updates for the chosen Profinet IO Device.
⇄	Automatically allocate the IP address for the chosen Profinet IO Device based on the IP address range specified in the 'Default Slave IP Parameter'. (If there are multiple IO Devices requiring configuration, select all of them at once and then utilize this function key for configuration).

- **Topology**

This page presents the Profinet network architecture and provides details about each adjacent device. On the left side, there is a depiction of the Profinet network architecture intended for the ongoing project, while the right side shows the real connection information. Any disparities will be emphasized in orange.

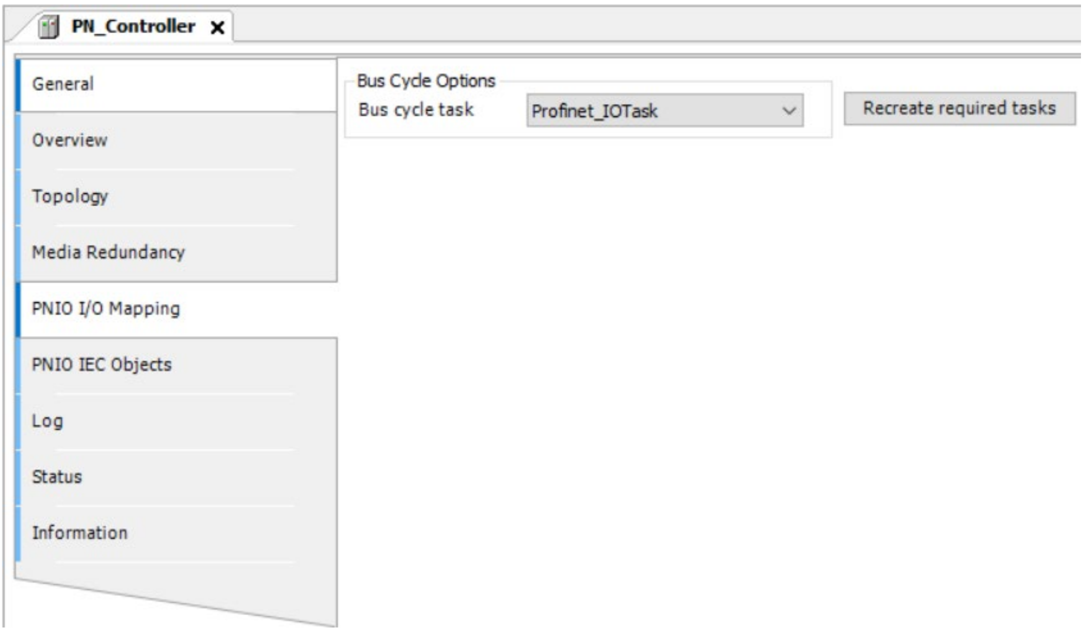


Port	Peer	Peer Port
PN_Controller (controller)		
port-001	pn-de...	port-001
pn_device_1 (pn-d...		
port-001	contro...	port-001
port-002	pn-de...	port-001
pn_device...		
port-001	pn-de...	port-002
port-002	pn-de...	port-001

Port	Peer	Peer P...	IP
PN_Controller (contr...			
port-001	pn-device-1	port-001	
pn-device-1			192.
port-001	controller	port-001	
port-002	🟧	🟧	

- **PNIO I/O Mapping**

Set up the Bus Cycle Task for Profinet IO Controller. Refer to section 4.2.1.6 PLC Settings for more details.

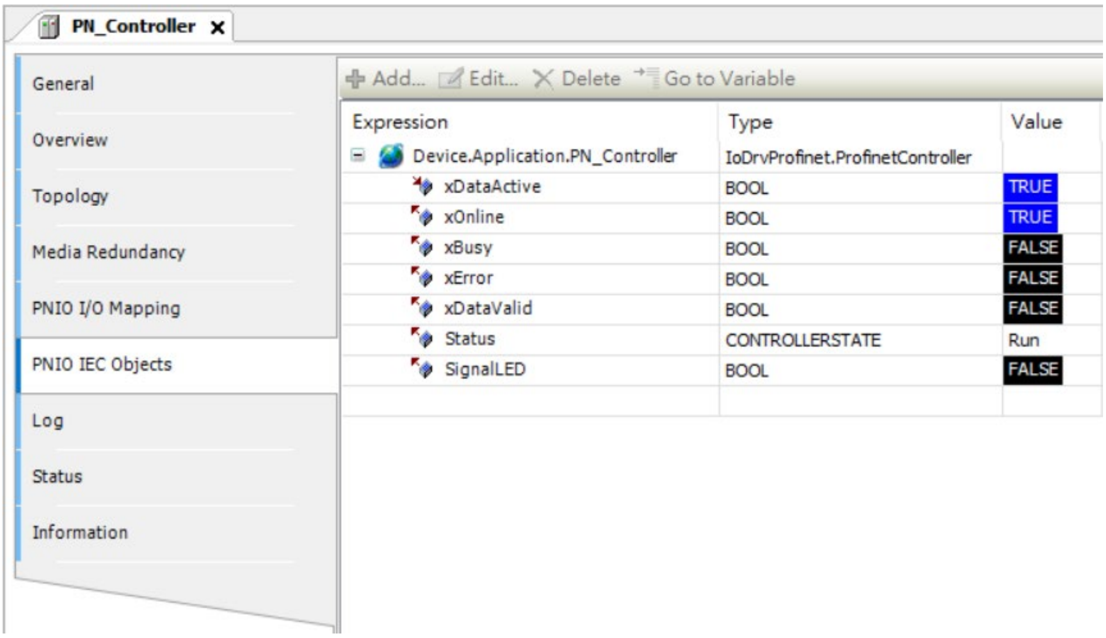


Bus Cycle Options

Bus cycle task: Profinet_IOTask

- **PNIO IEC Objects**

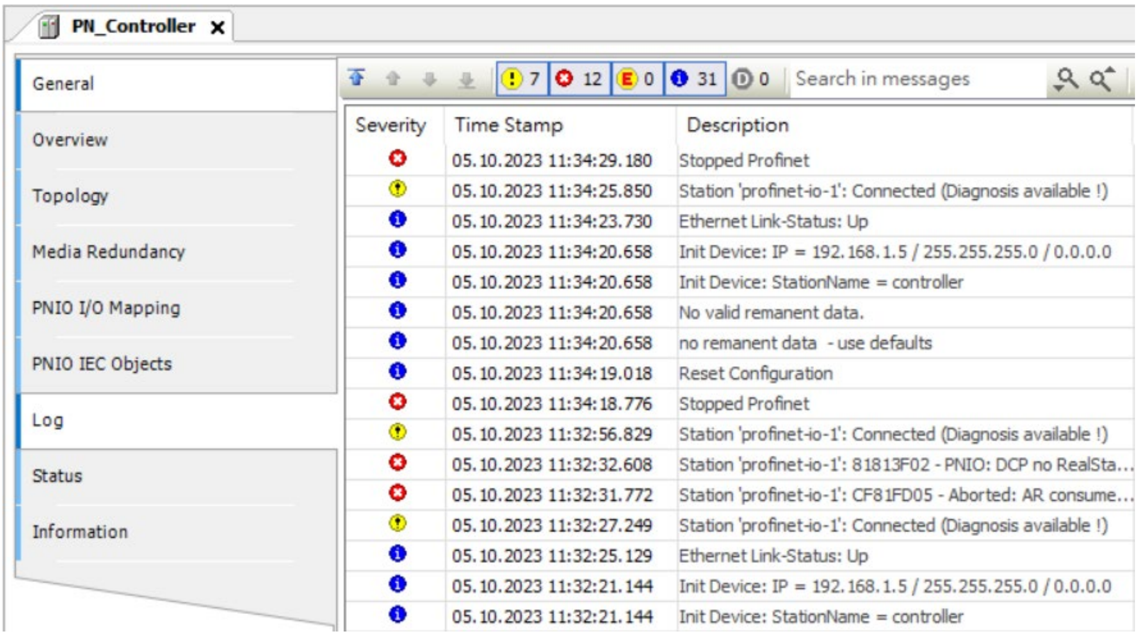
Here you can find the status of the Profinet IO Controller.



Expression	Type	Value
Device.Application.PN_Controller	IoDrvProfinet.ProfinetController	
xDataActive	BOOL	TRUE
xOnline	BOOL	TRUE
xBusy	BOOL	FALSE
xError	BOOL	FALSE
xDataValid	BOOL	FALSE
Status	CONTROLLERSTATE	Run
SignalLED	BOOL	FALSE

- **Log**

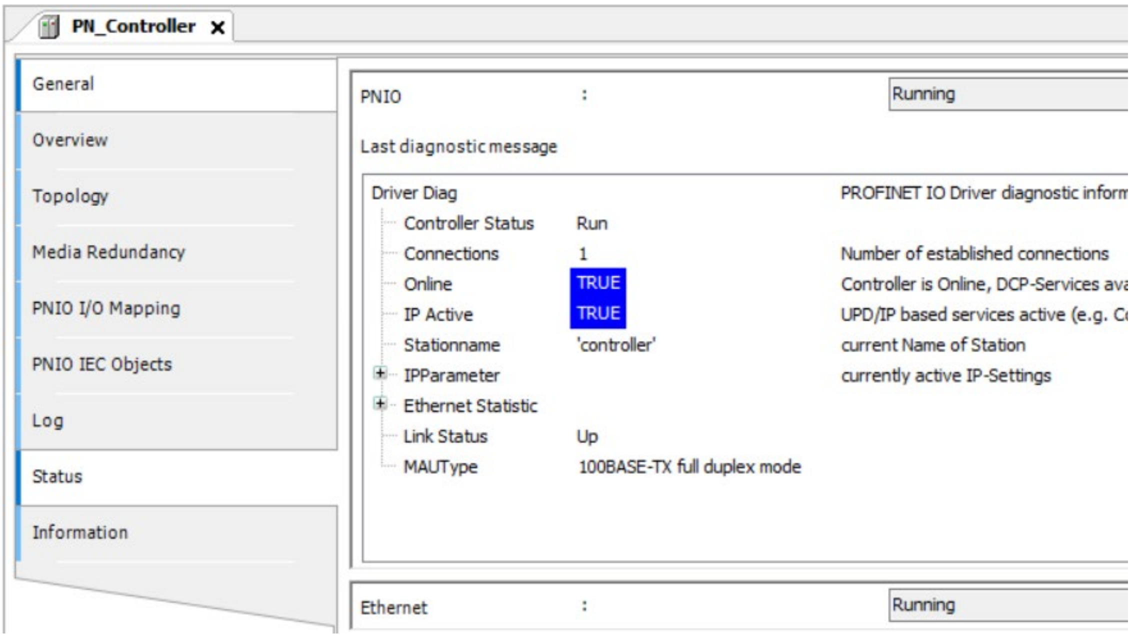
Here you can find the log of the Profinet IO Controller.



Severity	Time Stamp	Description
⊗	05.10.2023 11:34:29.180	Stopped Profinet
!	05.10.2023 11:34:25.850	Station 'profinet-io-1': Connected (Diagnosis available !)
ℹ	05.10.2023 11:34:23.730	Ethernet Link-Status: Up
ℹ	05.10.2023 11:34:20.658	Init Device: IP = 192.168.1.5 / 255.255.255.0 / 0.0.0.0
ℹ	05.10.2023 11:34:20.658	Init Device: StationName = controller
ℹ	05.10.2023 11:34:20.658	No valid remanent data.
ℹ	05.10.2023 11:34:20.658	no remanent data - use defaults
ℹ	05.10.2023 11:34:19.018	Reset Configuration
⊗	05.10.2023 11:34:18.776	Stopped Profinet
!	05.10.2023 11:32:56.829	Station 'profinet-io-1': Connected (Diagnosis available !)
⊗	05.10.2023 11:32:32.608	Station 'profinet-io-1': 81813F02 - PNIO: DCP no RealSta...
⊗	05.10.2023 11:32:31.772	Station 'profinet-io-1': CF81FD05 - Aborted: AR consume...
!	05.10.2023 11:32:27.249	Station 'profinet-io-1': Connected (Diagnosis available !)
ℹ	05.10.2023 11:32:25.129	Ethernet Link-Status: Up
ℹ	05.10.2023 11:32:21.144	Init Device: IP = 192.168.1.5 / 255.255.255.0 / 0.0.0.0
ℹ	05.10.2023 11:32:21.144	Init Device: StationName = controller

- **Status**

Here you can find the information of the Profinet IO Controller, including the operational status, the station name, IP parameters and so forth.



PNIO : Running

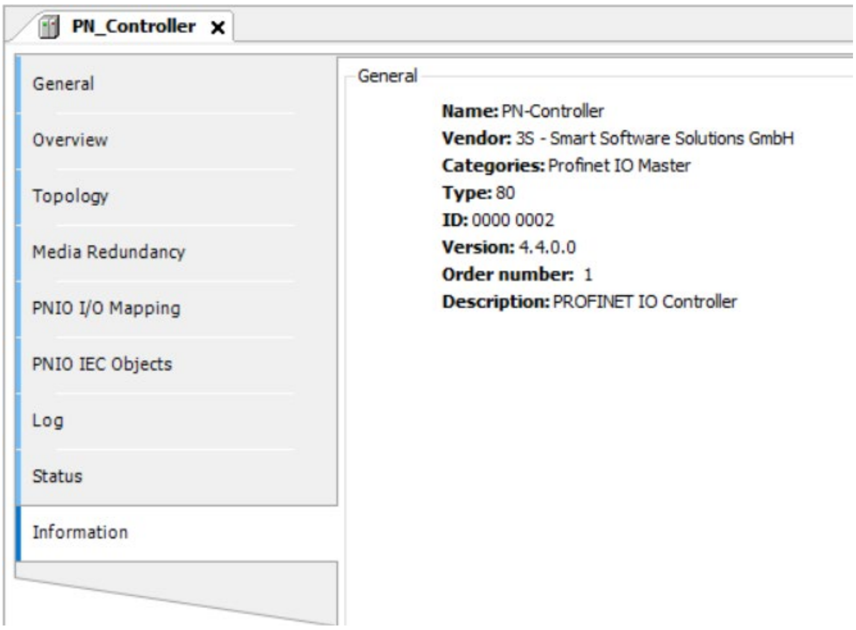
Last diagnostic message

Driver Diag		PROFINET IO Driver diagnostic inforr
Controller Status	Run	
Connections	1	Number of established connections
Online	TRUE	Controller is Online, DCP-Services ava
IP Active	TRUE	UPD/IP based services active (e.g. C
Stationname	'controller'	current Name of Station
IPParameter		currently active IP-Settings
Ethernet Statistic		
Link Status	Up	
MAUType	100BASE-TX full duplex mode	

Ethernet : Running

- **Information**

Here you can find general information of the Profinet IO Controller, including vendor, firmware version and other relevant information.



General

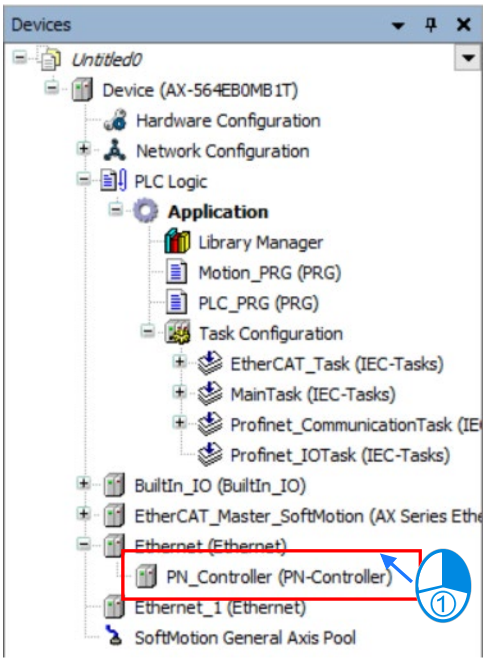
Name: PN-Controller
Vendor: 3S - Smart Software Solutions GmbH
Categories: Profinet IO Master
Type: 80
ID: 0000 0002
Version: 4.4.0.0
Order number: 1
Description: PROFINET IO Controller

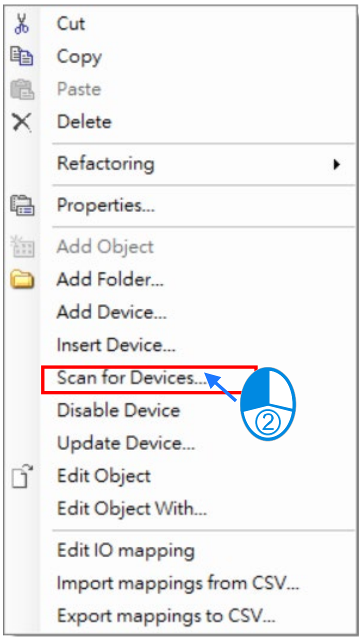
8.5.1.3 Operational Example of Profinet IO Controller

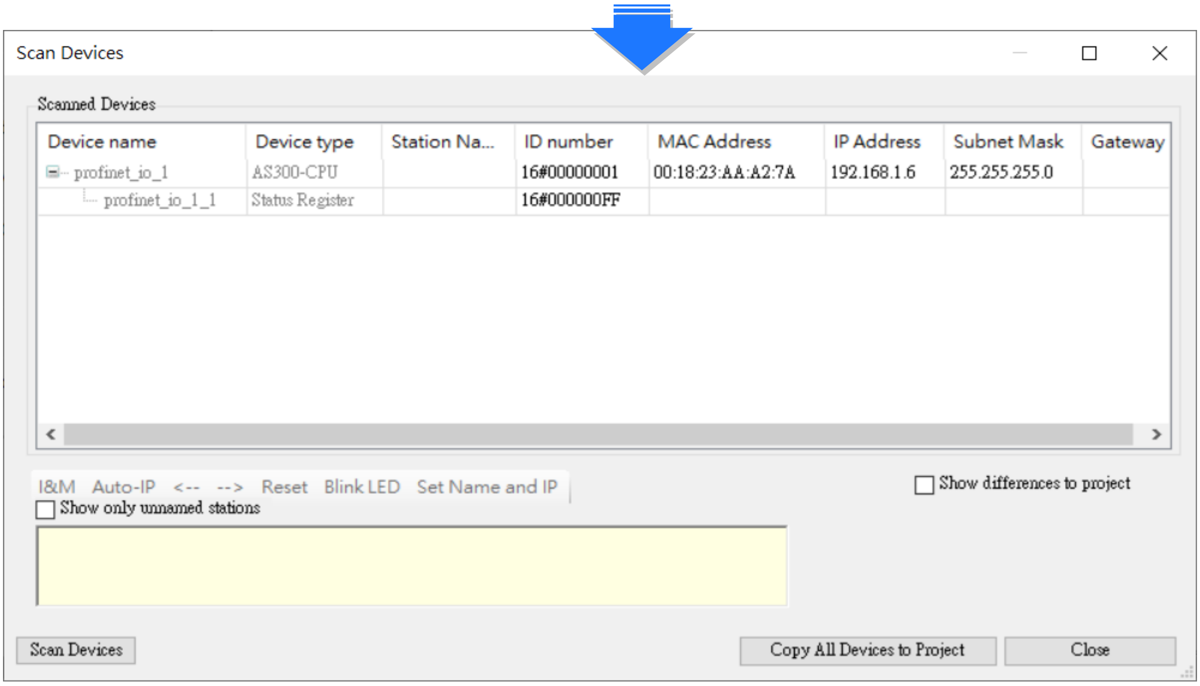
8.5.1.3.1 Example of Connecting AX-5 Series to AS300 and AS-FPFN02

- **Add a Profinet IO Device**

Adding a Profinet IO Device by using the function *Scan for Devices* or *Add Device*. This example uses Scan for Devices to add a Profinet IO Device. Prior to using the Scan for Devices function, you need to download a project that contains the PN-Controller component.







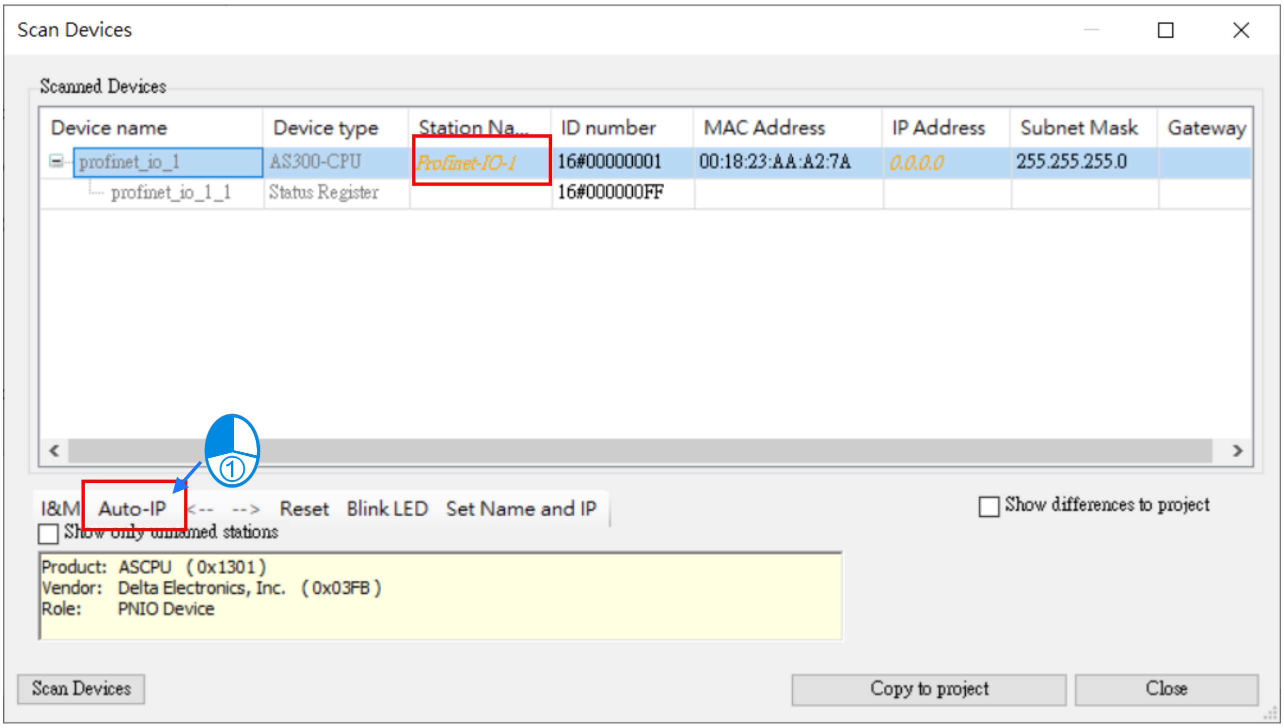
Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gateway
profinet_io_1	AS300-CPU		16#00000001	00:18:23:AA:A2:7A	192.168.1.6	255.255.255.0	
profinet_io_1_1	Status Register		16#000000FF				

Prior to setting up the IP address of the Profinet IO Device, it is necessary to define the Station Name. The IP address can be set either by utilizing automatic assignment (Auto IP) or by manually inputting the IP address (Set Name and IP address).

- Scan Devices

Function Key	Description
I & M	Display module identification information.
Auto-IP	Assign the Profinet IO Device IP address automatically. Note: Define the Station Name first before using this function.
Reset	Reset the Profinet IO Device to the defaults.
Blink LED	The Profinet IO Device's LED will flash to indicate its status.
Set Name and IP	Set up the Station Name and IP address manually.

- Auto-IP

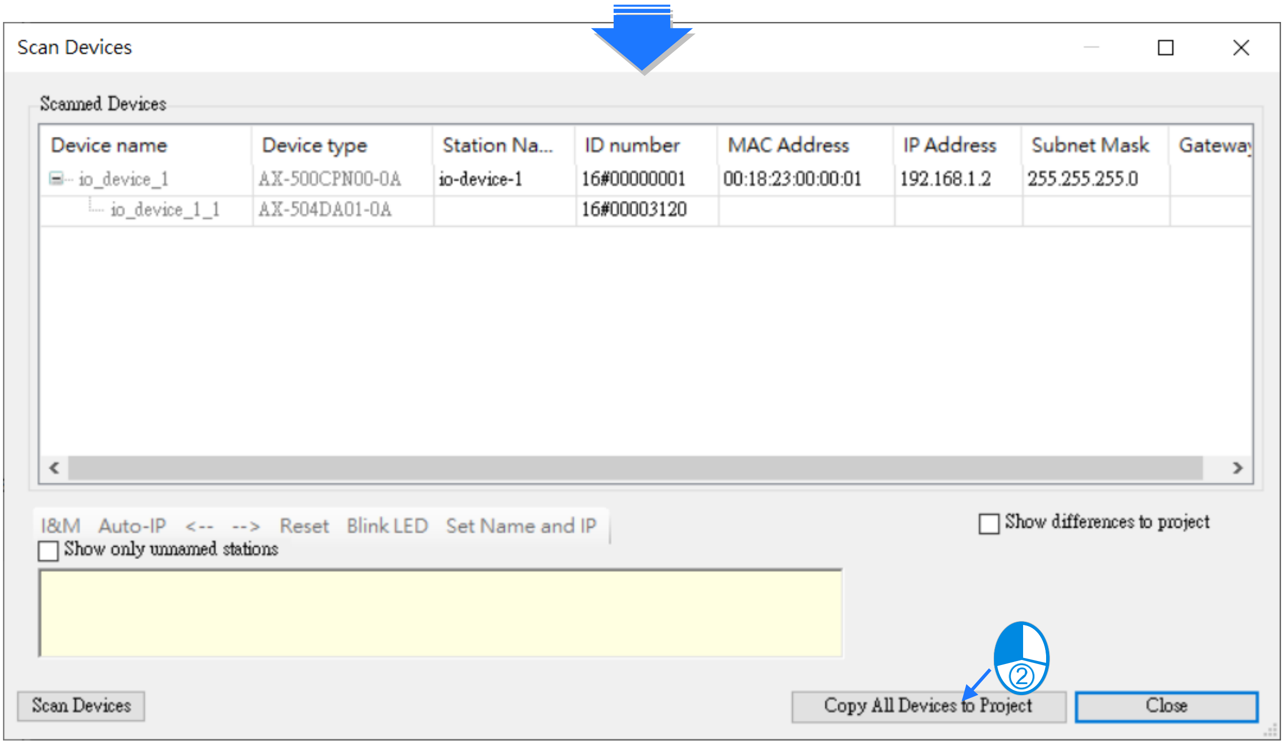


Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gateway
profinet_io_1	AS300-CPU	Profinet-IO-1	16#00000001	00:18:23:AA:A2:7A	0.0.0.0	255.255.255.0	
profinet_io_1_1	Status Register		16#000000FF				

☐ Show only unnamed stations

☐ Show differences to project

Product: ASCPU (0x1301)
Vendor: Delta Electronics, Inc. (0x03FB)
Role: PNIO Device

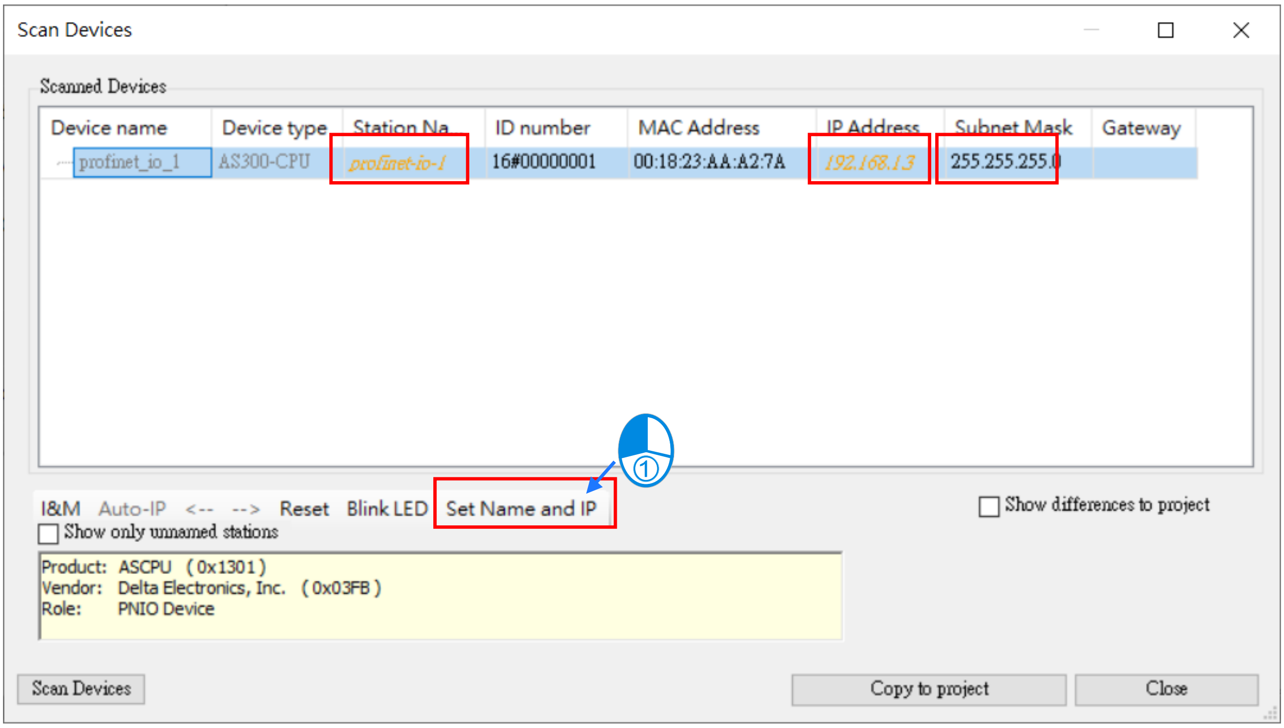


Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gateway
io_device_1	AX-500CPN00-0A	io-device-1	16#00000001	00:18:23:00:00:01	192.168.1.2	255.255.255.0	
io_device_1_1	AX-504DA01-0A		16#00003120				

☐ Show only unnamed stations

☐ Show differences to project

- Set up the Station Name and IP address manually.

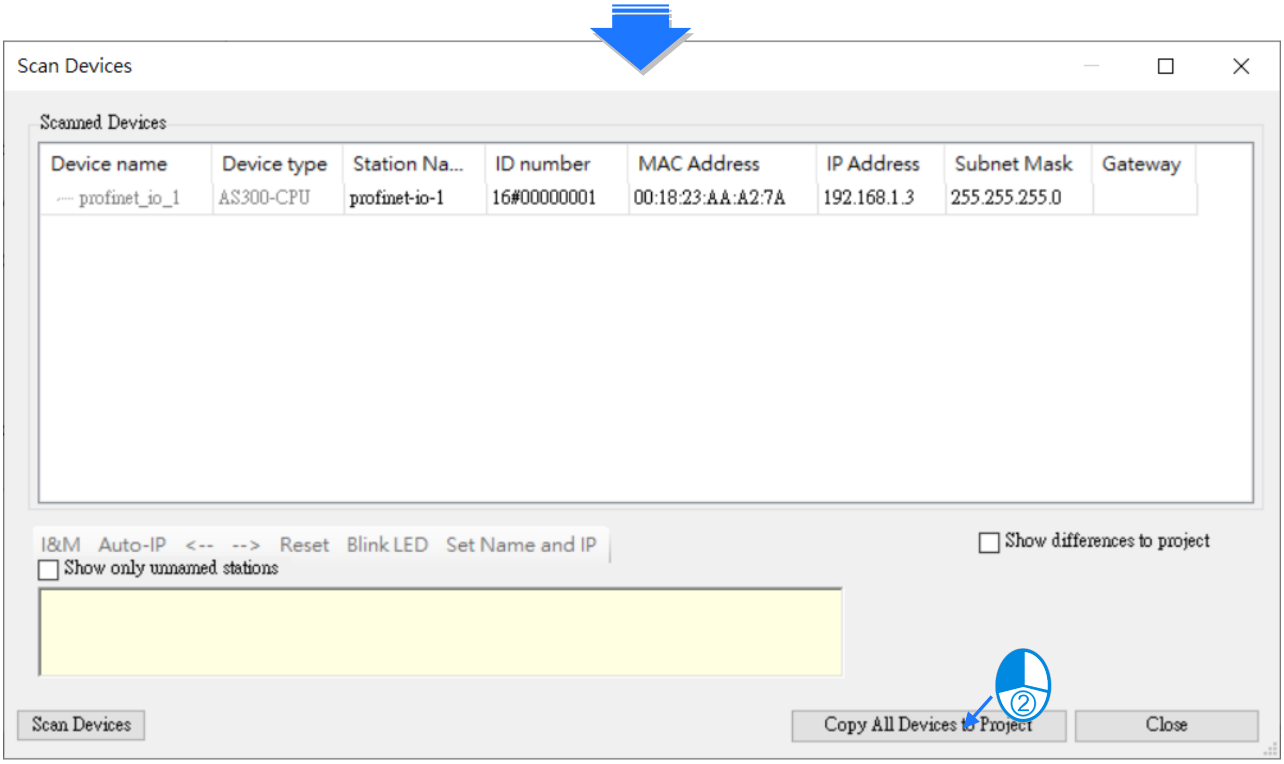


Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gateway
profinet_io_1	AS300-CPU	profinet-io-1	16#00000001	00:18:23:AA:A2:7A	192.168.1.3	255.255.255.0	

☐ Show only unnamed stations

☐ Show differences to project

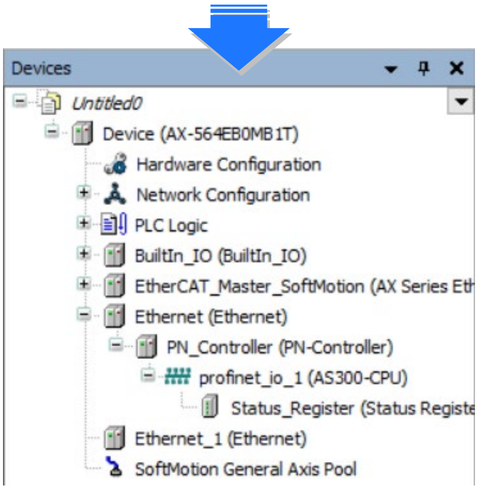
Product: ASCPU (0x1301)
Vendor: Delta Electronics, Inc. (0x03FB)
Role: PNIO Device



Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gateway
profinet_io_1	AS300-CPU	profinet-io-1	16#00000001	00:18:23:AA:A2:7A	192.168.1.3	255.255.255.0	

☐ Show only unnamed stations

☐ Show differences to project

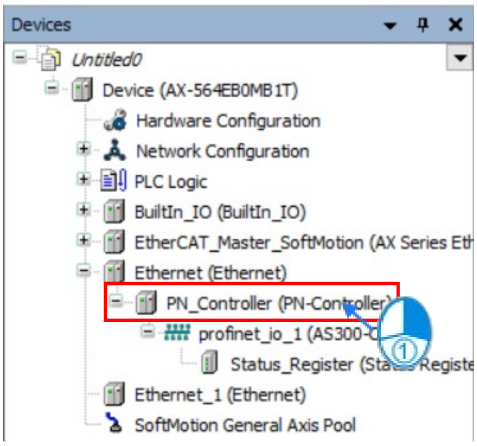


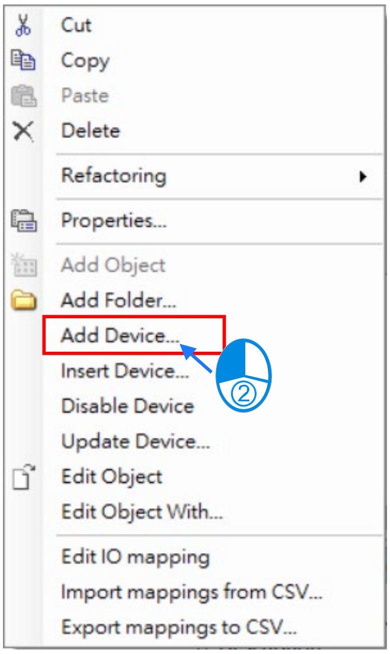
- Untitled0
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
    - BuiltIn_IO (BuiltIn_IO)
    - EtherCAT_Master_SoftMotion (AX Series Eth
    - Ethernet (Ethernet)
      - PN_Controller (PN-Controller)
        - profinet_io_1 (AS300-CPU)
          - Status_Register (Status Registe
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

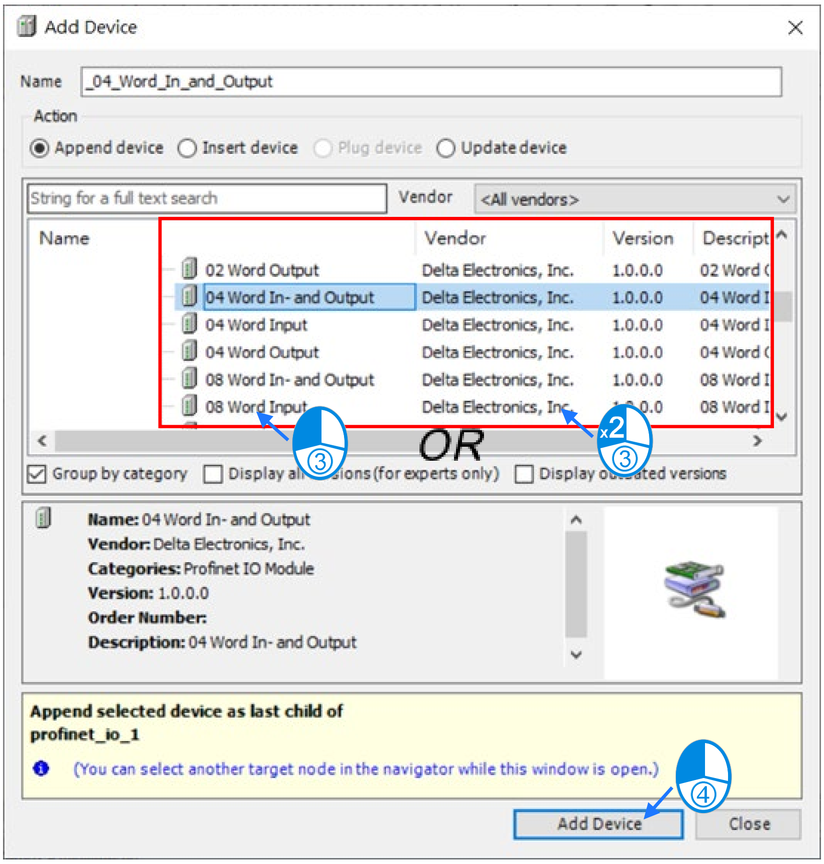
- **Add device and set up the data exchange configurations.**

  You need to manually add a device and set up the data type, data size and device addresses for reading and writing.

  - Establish a connection for data exchange.







Name: _04_Word_In_and_Output

◉ Append device ○ Insert device ○ Plug device ○ Update device

String for a full text search Vendor <All vendors>

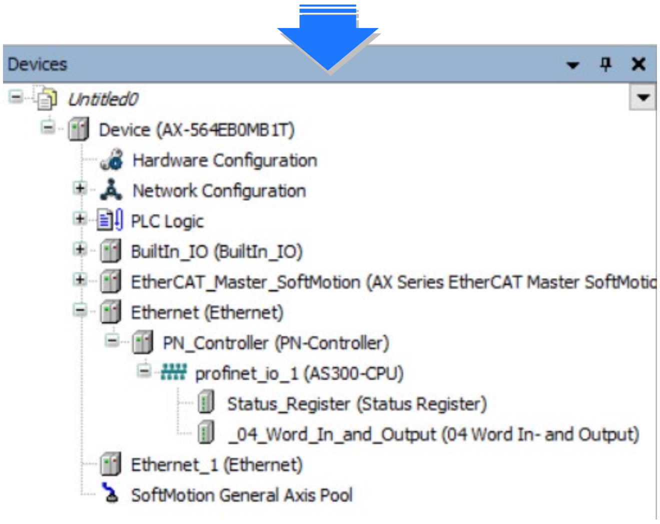
Name	Vendor	Version	Descript
02 Word Output	Delta Electronics, Inc.	1.0.0.0	02 Word (
04 Word In- and Output	Delta Electronics, Inc.	1.0.0.0	04 Word I
04 Word Input	Delta Electronics, Inc.	1.0.0.0	04 Word I
04 Word Output	Delta Electronics, Inc.	1.0.0.0	04 Word (
08 Word In- and Output	Delta Electronics, Inc.	1.0.0.0	08 Word I
08 Word Input	Delta Electronics, Inc.	1.0.0.0	08 Word I

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: 04 Word In- and Output
Vendor: Delta Electronics, Inc.
Categories: Profinet IO Module
Version: 1.0.0.0
Order Number:
Description: 04 Word In- and Output

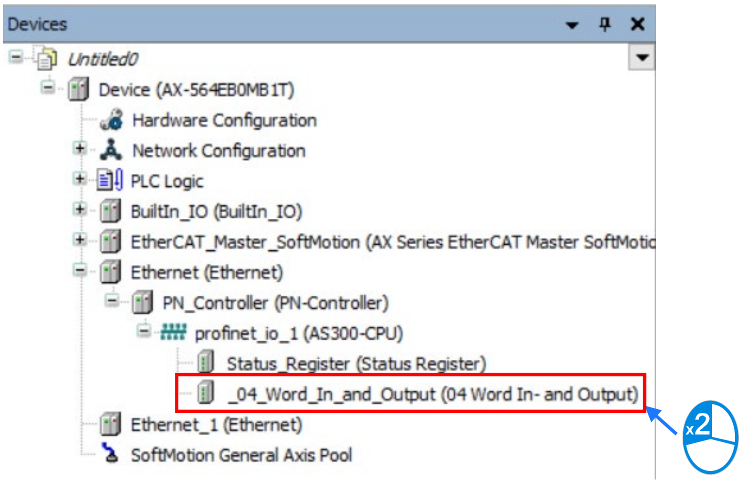
Append selected device as last child of profinet_io_1

(You can select another target node in the navigator while this window is open.)

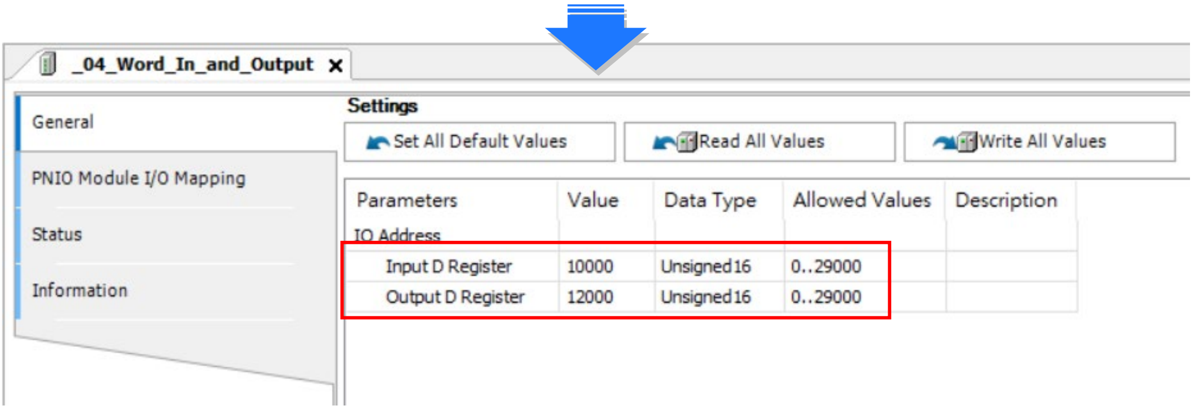


- Untitled0
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
    - BuiltIn_IO (BuiltIn_IO)
    - EtherCAT_Master_SoftMotion (AX Series EtherCAT Master SoftMotic
    - Ethernet (Ethernet)
      - PN_Controller (PN-Controller)
        - profinet_io_1 (AS300-CPU)
          - Status_Register (Status Register)
          - _04_Word_In_and_Output (04 Word In- and Output)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

- Specify the device addresses for data exchange (reading and writing).
  You can specify the device address of the AS Series PLC CPU for reading and writing on the General page of the data exchange.



- Untitled0
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
    - BuiltIn_IO (BuiltIn_IO)
    - EtherCAT_Master_SoftMotion (AX Series EtherCAT Master SoftMotic
    - Ethernet (Ethernet)
      - PN_Controller (PN-Controller)
        - profinet_io_1 (AS300-CPU)
          - Status_Register (Status Register)
          - _04_Word_In_and_Output (04 Word In- and Output)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

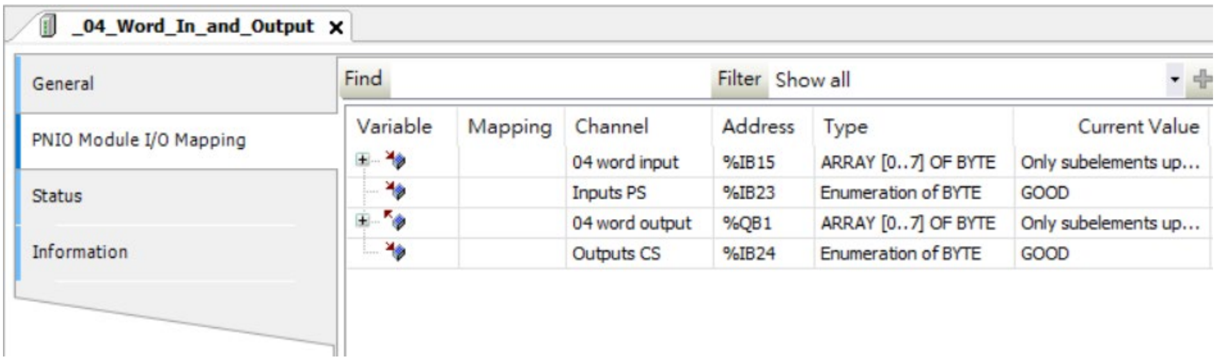


Parameters	Value	Data Type	Allowed Values	Description
IO Address				
Input D Register	10000	Unsigned16	0..29000	
Output D Register	12000	Unsigned16	0..29000	

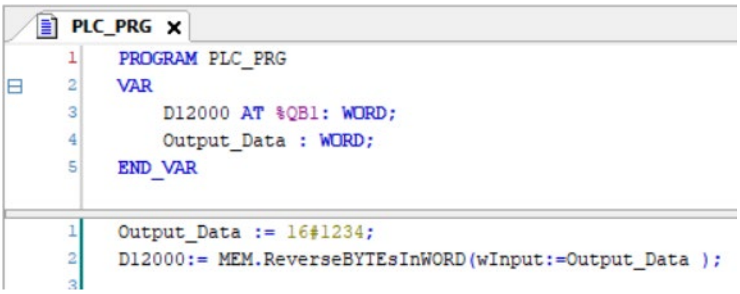
- Programming

  Check the memory location corresponding to the data on the PNIO Module I/O Mapping page and make any necessary adjustments to the data format.

Note: Since the byte order definition for AX Series PLC is different from that of AH and AS Series PLC, you must manually swap the High Byte and Low Byte. The MEM.ReverseBYTEsInWORD instruction can be used to perform this swap.



Variable	Mapping	Channel	Address	Type	Current Value
		04 word input	%IB15	ARRAY [0..7] OF BYTE	Only subelements up...
		Inputs PS	%IB23	Enumeration of BYTE	GOOD
		04 word output	%QB1	ARRAY [0..7] OF BYTE	Only subelements up...
		Outputs CS	%IB24	Enumeration of BYTE	GOOD



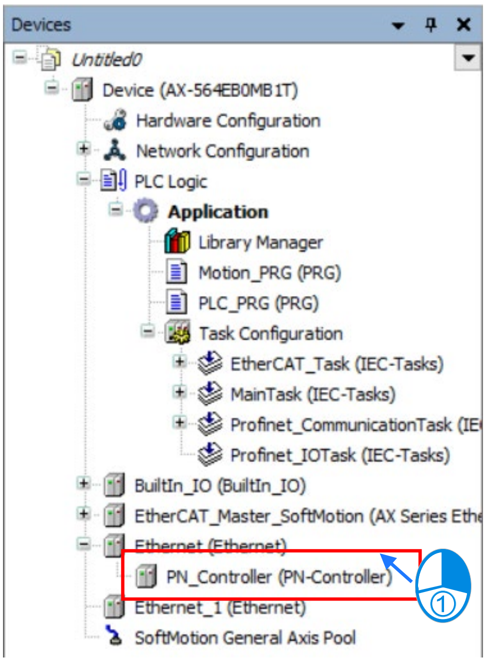
```
PROGRAM PLC_PRG
VAR
    D12000 AT %QB1: WORD;
    Output_Data : WORD;
END_VAR
```

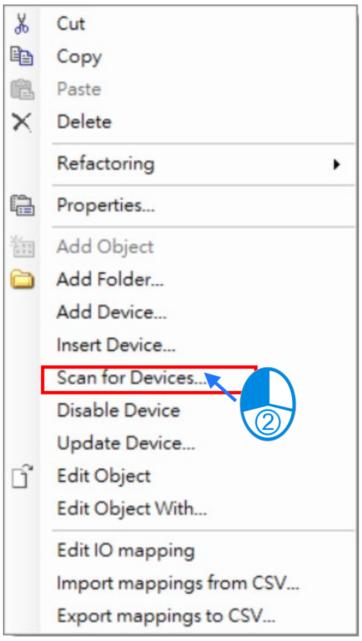
```
Output_Data := 16#1234;
D12000:= MEM.ReverseBYTEsInWORD(wInput:=Output_Data );
```

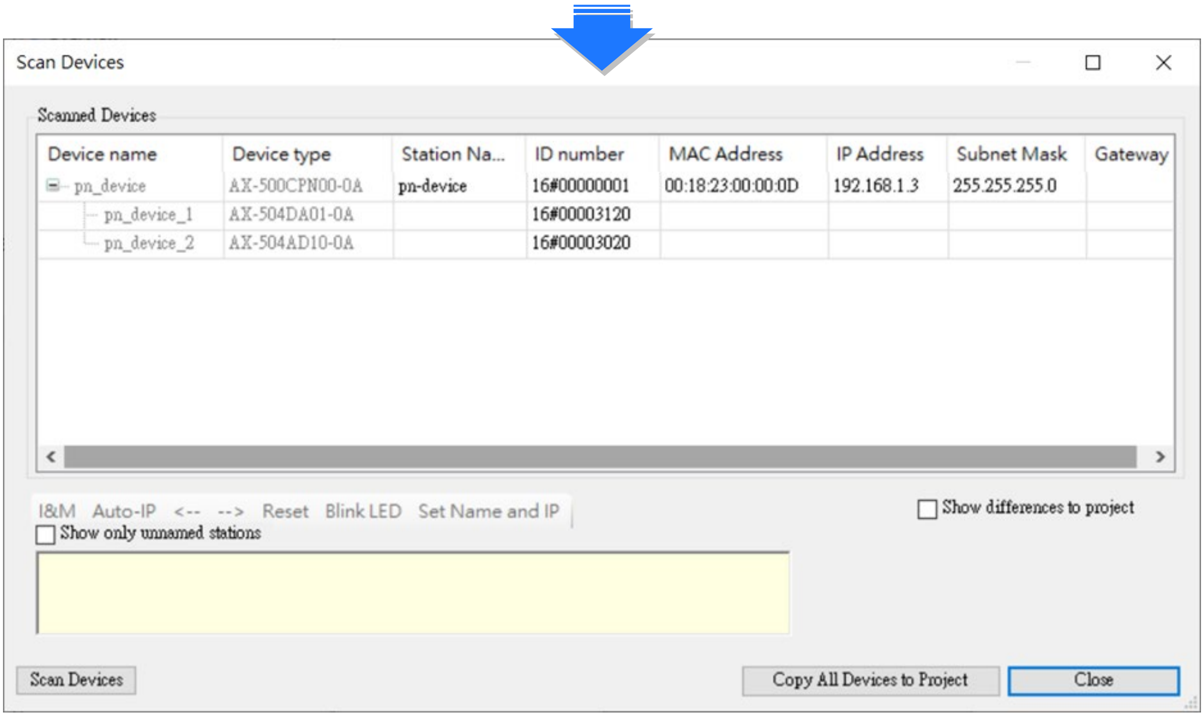
8.5.1.3.2 Example of Connecting AX-5 Series to AX-500CPN00-0A

- **Add a Profinet IO Device**

Adding a Profinet IO Device by using the function *Scan for Devices* or *Add Device*. This example uses Scan for Devices to add a Profinet IO Device. Prior to using the Scan for Devices function, you need to download a project that contains the PN-Controller component.







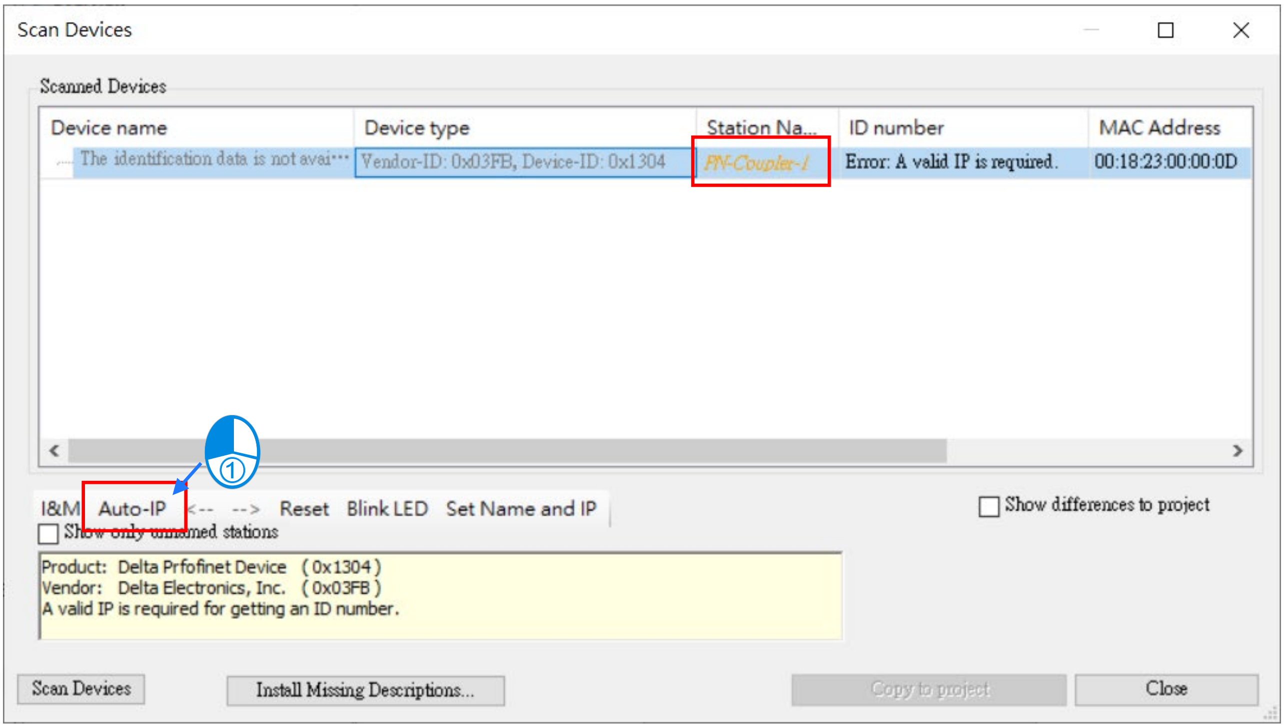
Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gateway
pn_device	AX-500CPN00-0A	pn-device	16#00000001	00:18:23:00:00:0D	192.168.1.3	255.255.255.0	
pn_device_1	AX-504DA01-0A		16#00003120				
pn_device_2	AX-504AD10-0A		16#00003020				

Prior to setting up the IP address of the Profinet IO Device, it is necessary to define the Station Name. The IP address can be set either by utilizing automatic assignment (Auto IP) or by manually inputting the IP address (Set Name and IP address).

- Scan Devices

Function Key	Description
I & M	Display module identification information.
Auto-IP	Assign the Profinet IO Device IP address automatically. Note: Define the Station Name first before using this function.
Reset	Reset the Profinet IO Device to the defaults.
Blink LED	The Profinet IO Device's LED will flash to indicate its status.
Set Name and IP	Set up the Station Name and IP address manually.

- Auto-IP



Scan Devices

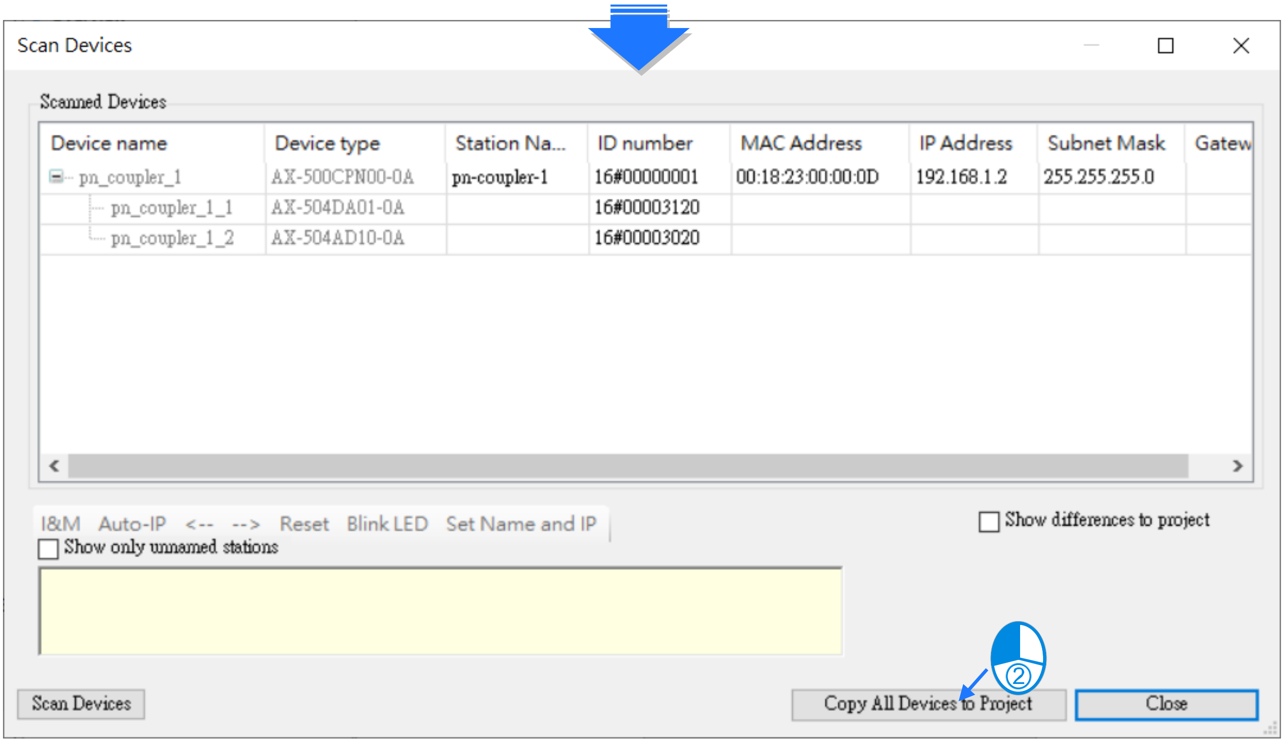
Scanned Devices

Device name	Device type	Station Na...	ID number	MAC Address
The identification data is not avai···	Vendor-ID: 0x03FB, Device-ID: 0x1304	PN-Coupler-1	Error: A valid IP is required.	00:18:23:00:00:0D

☐ Show only unnamed stations

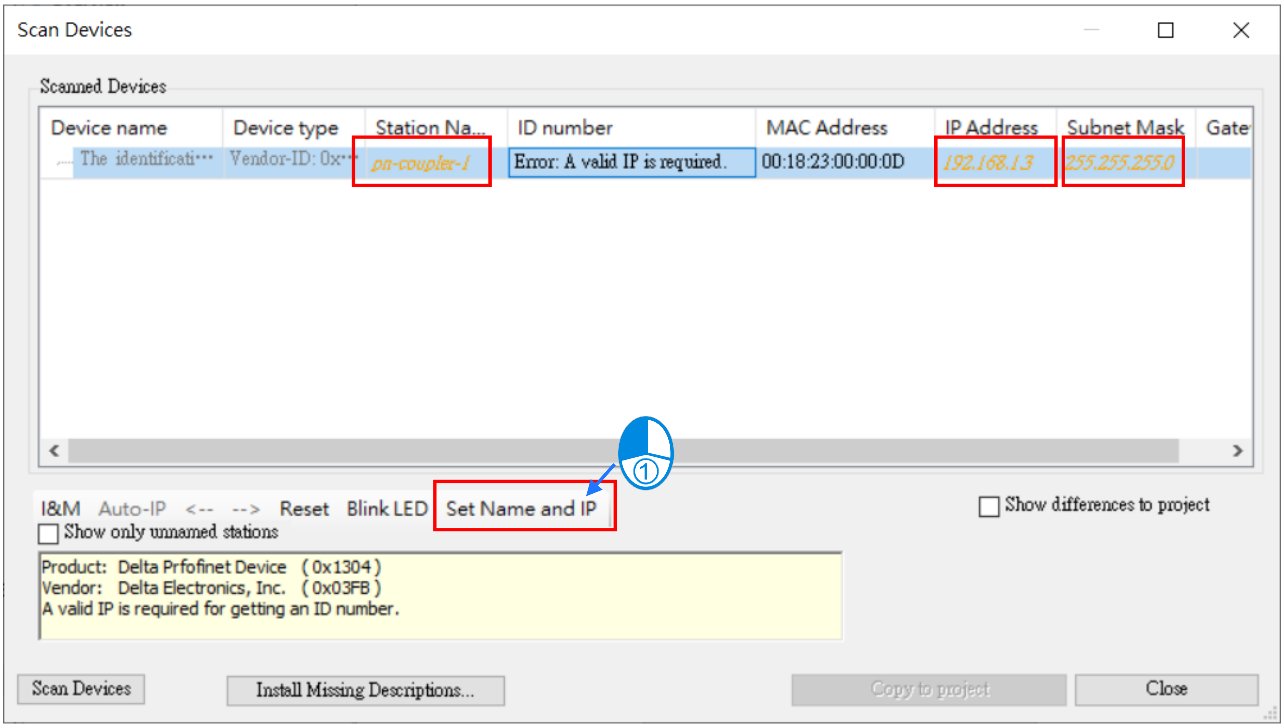
☐ Show differences to project

Product: Delta Prfofinet Device (0x1304)
Vendor: Delta Electronics, Inc. (0x03FB)
A valid IP is required for getting an ID number.



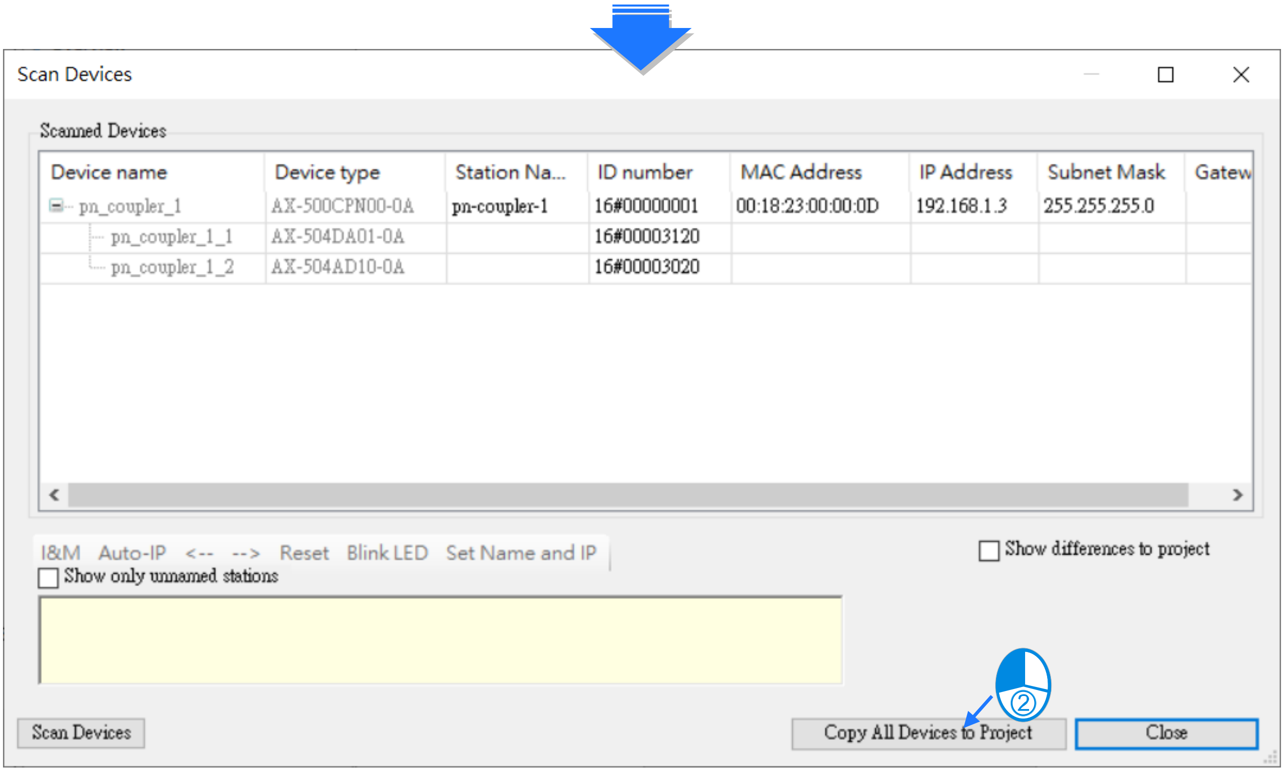
Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gatew
pn_coupler_1	AX-500CPN00-0A	pn-coupler-1	16#00000001	00:18:23:00:00:0D	192.168.1.2	255.255.255.0	
pn_coupler_1_1	AX-504DA01-0A		16#00003120				
pn_coupler_1_2	AX-504AD10-0A		16#00003020				

- Set up the Station Name and IP address manually.



Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gate
The identificati···	Vendor-ID: 0x···	pn-coupler-1	Error: A valid IP is required.	00:18:23:00:00:0D	192.168.1.3	255.255.255.0	

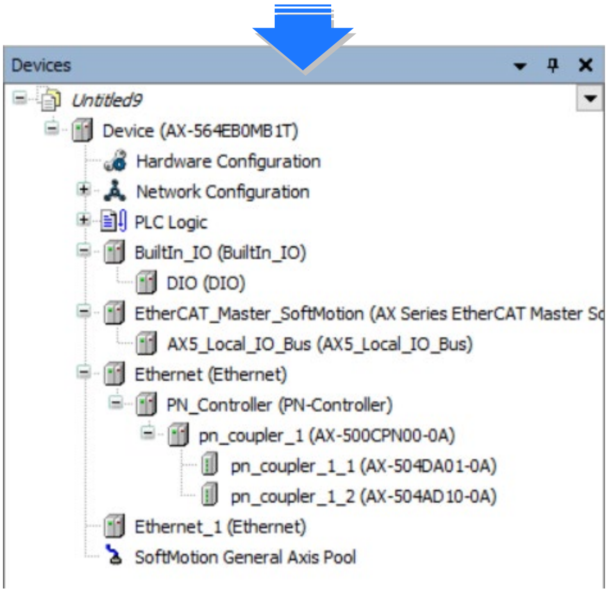
Product: Delta Prfofinet Device (0x1304)
Vendor: Delta Electronics, Inc. (0x03FB)
A valid IP is required for getting an ID number.



Scan Devices

Scanned Devices

Device name	Device type	Station Na...	ID number	MAC Address	IP Address	Subnet Mask	Gatew
pn_coupler_1	AX-500CPN00-0A	pn-coupler-1	16#00000001	00:18:23:00:00:0D	192.168.1.3	255.255.255.0	
pn_coupler_1_1	AX-504DA01-0A		16#00003120				
pn_coupler_1_2	AX-504AD10-0A		16#00003020				

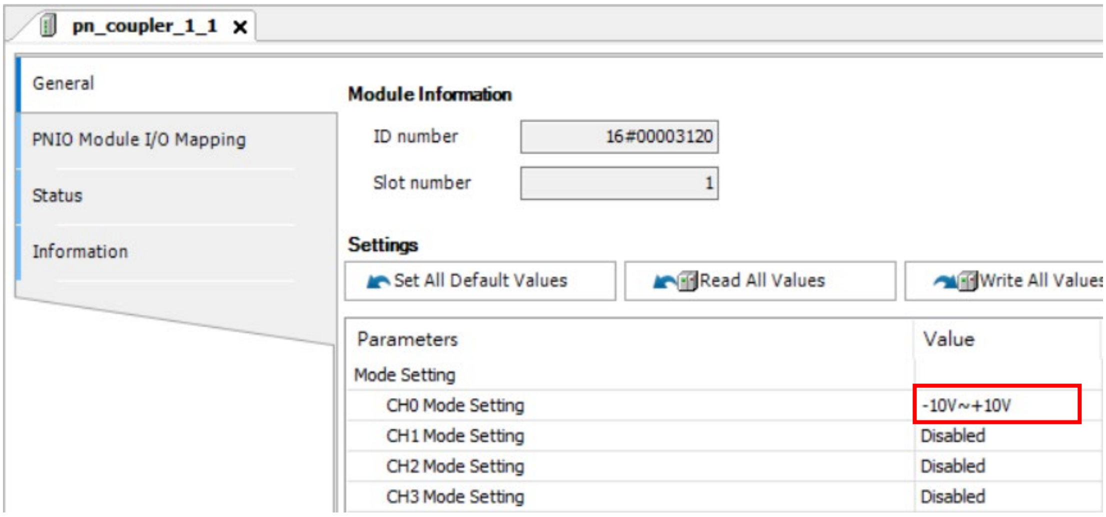


Devices

- Untitled9
  - Device (AX-564EB0MB1T)
    - Hardware Configuration
    - Network Configuration
    - PLC Logic
    - BuiltIn_IO (BuiltIn_IO)
      - DIO (DIO)
    - EtherCAT_Master_SoftMotion (AX Series EtherCAT Master Sc
      - AX5_Local_IO_Bus (AX5_Local_IO_Bus)
    - Ethernet (Ethernet)
      - PN_Controller (PN-Controller)
        - pn_coupler_1 (AX-500CPN00-0A)
          - pn_coupler_1_1 (AX-504DA01-0A)
          - pn_coupler_1_2 (AX-504AD10-0A)
    - Ethernet_1 (Ethernet)
    - SoftMotion General Axis Pool

- Example of mode setting and program writing for DA module (REAL Format)
  - CH0 Mode Setting: -10 V to +10 V
  - Format Setting: REAL
  - CH0 Output Value Setting when EtherCAT Connection Lost REAL: 4 V
  - CH0 Output Value Setting when Module in Stopped State REAL: 1 V

Note: You can customize the module name in the tree structure. Here the pn_coupler_1_1 is AX-504DA01-0A.

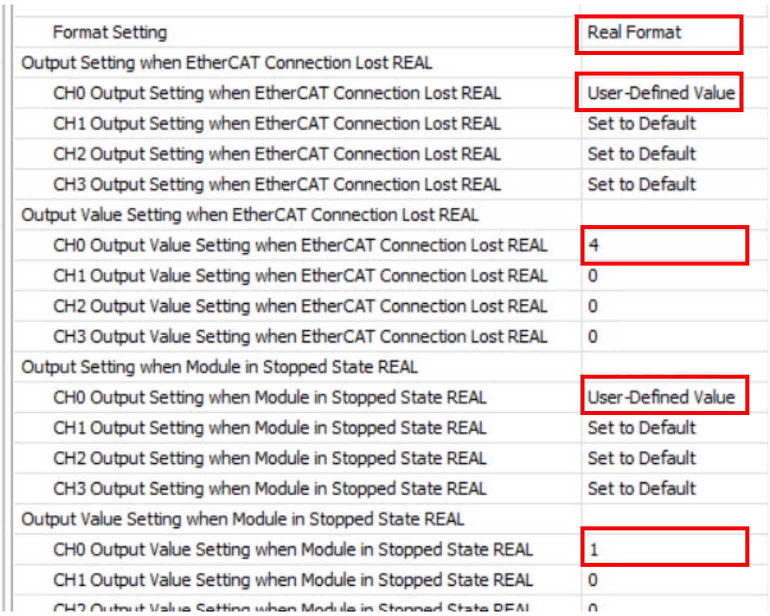


Module Information

ID number: 16#00003120

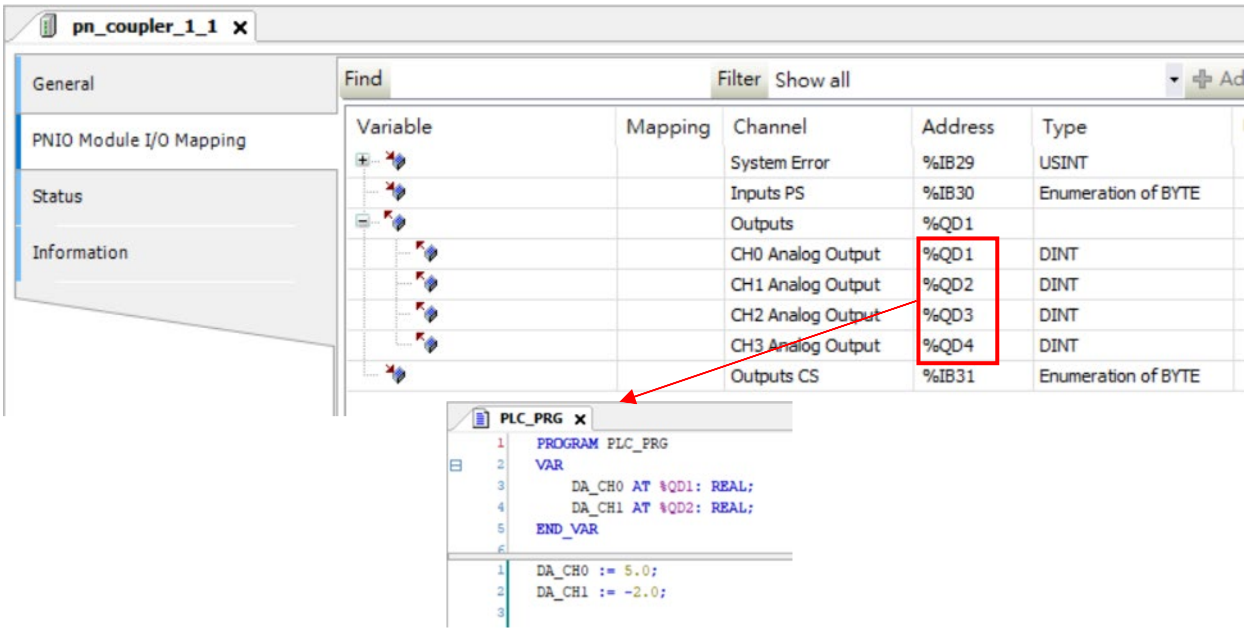
Slot number: 1

Settings



Parameters	Value
Mode Setting	
CH0 Mode Setting	-10V~+10V
CH1 Mode Setting	Disabled
CH2 Mode Setting	Disabled
CH3 Mode Setting	Disabled
Format Setting	Real Format
Output Setting when EtherCAT Connection Lost REAL	
CH0 Output Setting when EtherCAT Connection Lost REAL	User-Defined Value
CH1 Output Setting when EtherCAT Connection Lost REAL	Set to Default
CH2 Output Setting when EtherCAT Connection Lost REAL	Set to Default
CH3 Output Setting when EtherCAT Connection Lost REAL	Set to Default
Output Value Setting when EtherCAT Connection Lost REAL	
CH0 Output Value Setting when EtherCAT Connection Lost REAL	4
CH1 Output Value Setting when EtherCAT Connection Lost REAL	0
CH2 Output Value Setting when EtherCAT Connection Lost REAL	0
CH3 Output Value Setting when EtherCAT Connection Lost REAL	0
Output Setting when Module in Stopped State REAL	
CH0 Output Setting when Module in Stopped State REAL	User-Defined Value
CH1 Output Setting when Module in Stopped State REAL	Set to Default
CH2 Output Setting when Module in Stopped State REAL	Set to Default
CH3 Output Setting when Module in Stopped State REAL	Set to Default
Output Value Setting when Module in Stopped State REAL	
CH0 Output Value Setting when Module in Stopped State REAL	1
CH1 Output Value Setting when Module in Stopped State REAL	0
CH2 Output Value Setting when Module in Stopped State REAL	0

Prior to writing the program, it is necessary to identify the memory location assigned to the channel and to declare a REAL-type variable that is directly addressed to that memory location.



Variable	Mapping	Channel	Address	Type
		System Error	%IB29	USINT
		Inputs PS	%IB30	Enumeration of BYTE
		Outputs	%QD1	
		CH0 Analog Output	%QD1	DINT
		CH1 Analog Output	%QD2	DINT
		CH2 Analog Output	%QD3	DINT
		CH3 Analog Output	%QD4	DINT
		Outputs CS	%IB31	Enumeration of BYTE

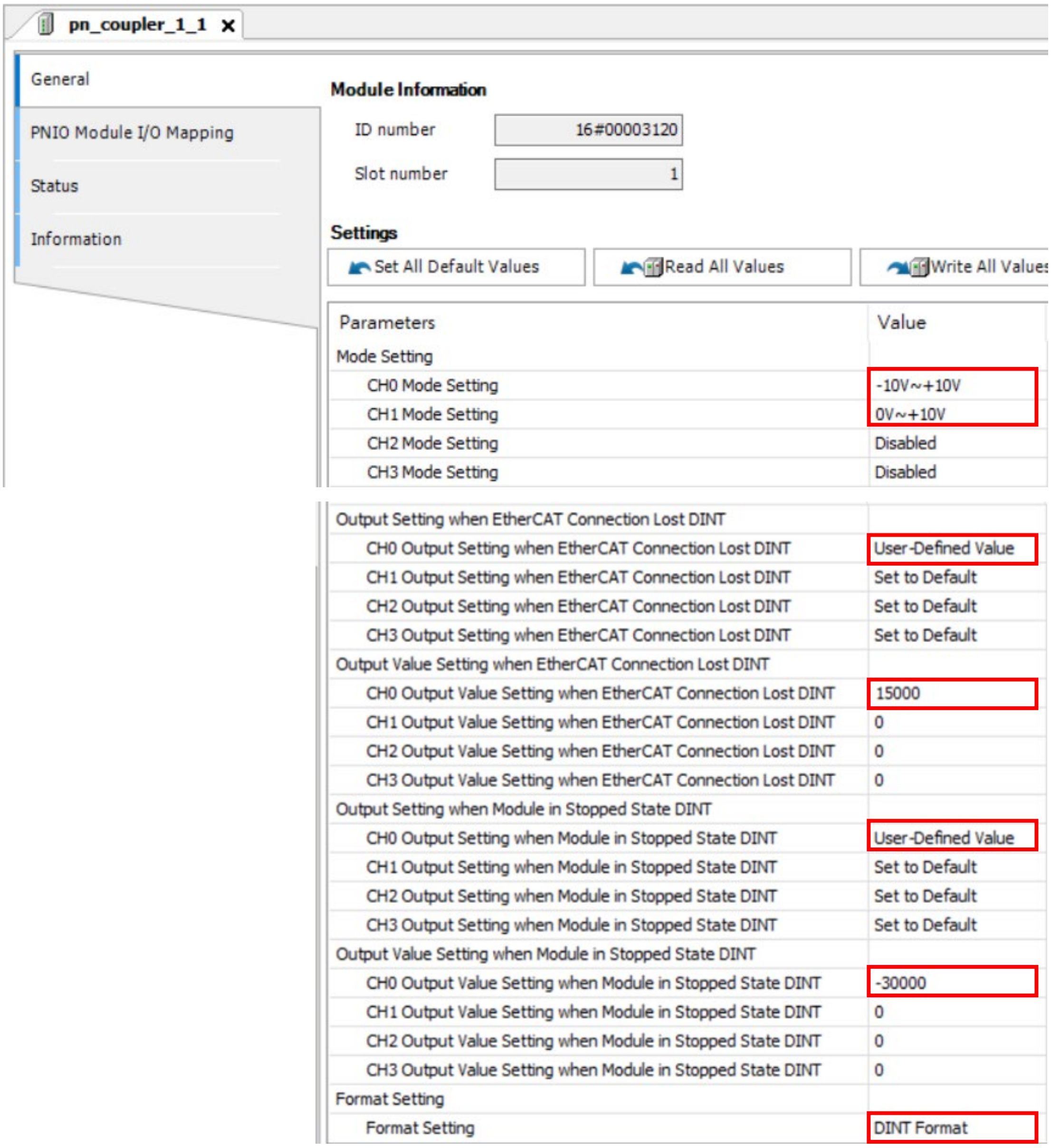
```
PROGRAM PLC_PRG
VAR
    DA_CH0 AT %QD1: REAL;
    DA_CH1 AT %QD2: REAL;
END_VAR
```

```
DA_CH0 := 5.0;
DA_CH1 := -2.0;
```

- Example of mode setting and program writing for DA module (DINT Format)
    - CH0 Mode Setting: -10 V to +10 V
    - CH1 Mode Setting: 0 V to +10 V
    - Format Setting: DINT
    - CH0 Output Value Setting when EtherCAT Connection Lost DINT: 15000 (5 V)
    - CH0 Output Value Setting when Module in Stopped State DINT: -30000 (-10 V)

Channel setting range: -10 V to +10 V: -30,000 to 30,000; 0 V to +10 V: 0 to 30,000.
Refer to AX-5 Series Module Manual for more channel setting ranges.

Note: You can customize the module name in the tree structure. Here the pn_coupler_1_1 is AX-504DA01-0A.



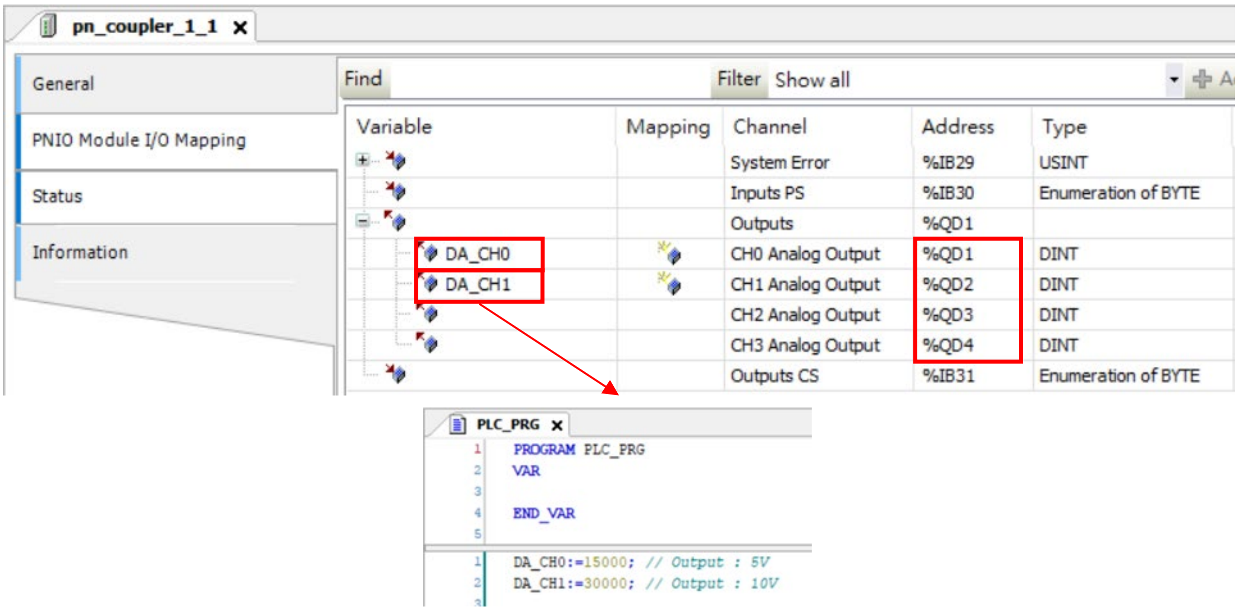
Module Information

ID number: 16#00003120
Slot number: 1

Settings

Parameters	Value
Mode Setting	
CH0 Mode Setting	-10V~+10V
CH1 Mode Setting	0V~+10V
CH2 Mode Setting	Disabled
CH3 Mode Setting	Disabled
Output Setting when EtherCAT Connection Lost DINT	
CH0 Output Setting when EtherCAT Connection Lost DINT	User-Defined Value
CH1 Output Setting when EtherCAT Connection Lost DINT	Set to Default
CH2 Output Setting when EtherCAT Connection Lost DINT	Set to Default
CH3 Output Setting when EtherCAT Connection Lost DINT	Set to Default
Output Value Setting when EtherCAT Connection Lost DINT	
CH0 Output Value Setting when EtherCAT Connection Lost DINT	15000
CH1 Output Value Setting when EtherCAT Connection Lost DINT	0
CH2 Output Value Setting when EtherCAT Connection Lost DINT	0
CH3 Output Value Setting when EtherCAT Connection Lost DINT	0
Output Setting when Module in Stopped State DINT	
CH0 Output Setting when Module in Stopped State DINT	User-Defined Value
CH1 Output Setting when Module in Stopped State DINT	Set to Default
CH2 Output Setting when Module in Stopped State DINT	Set to Default
CH3 Output Setting when Module in Stopped State DINT	Set to Default
Output Value Setting when Module in Stopped State DINT	
CH0 Output Value Setting when Module in Stopped State DINT	-30000
CH1 Output Value Setting when Module in Stopped State DINT	0
CH2 Output Value Setting when Module in Stopped State DINT	0
CH3 Output Value Setting when Module in Stopped State DINT	0
Format Setting	
Format Setting	DINT Format

Prior to writing the program, it is necessary to identify the memory location assigned to the channel and to declare a DINT-type variable that is directly addressed to that memory location.



Variable	Mapping	Channel	Address	Type
		System Error	%IB29	USINT
		Inputs PS	%IB30	Enumeration of BYTE
		Outputs	%QD1	
DA_CH0		CH0 Analog Output	%QD1	DINT
DA_CH1		CH1 Analog Output	%QD2	DINT
		CH2 Analog Output	%QD3	DINT
		CH3 Analog Output	%QD4	DINT
		Outputs CS	%IB31	Enumeration of BYTE

```
PROGRAM PLC_PRG
VAR

END_VAR

```

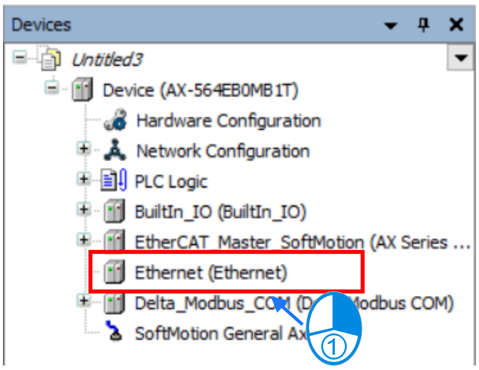
```
DA_CH0:=15000; // Output : 5V
DA_CH1:=30000; // Output : 10V
```

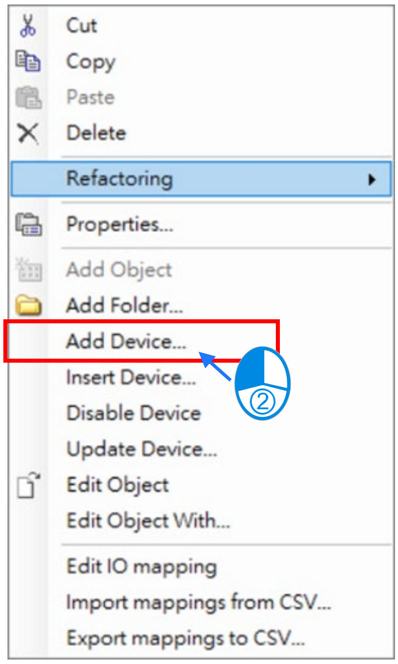
8.5.2 PROFINET IO Device Function

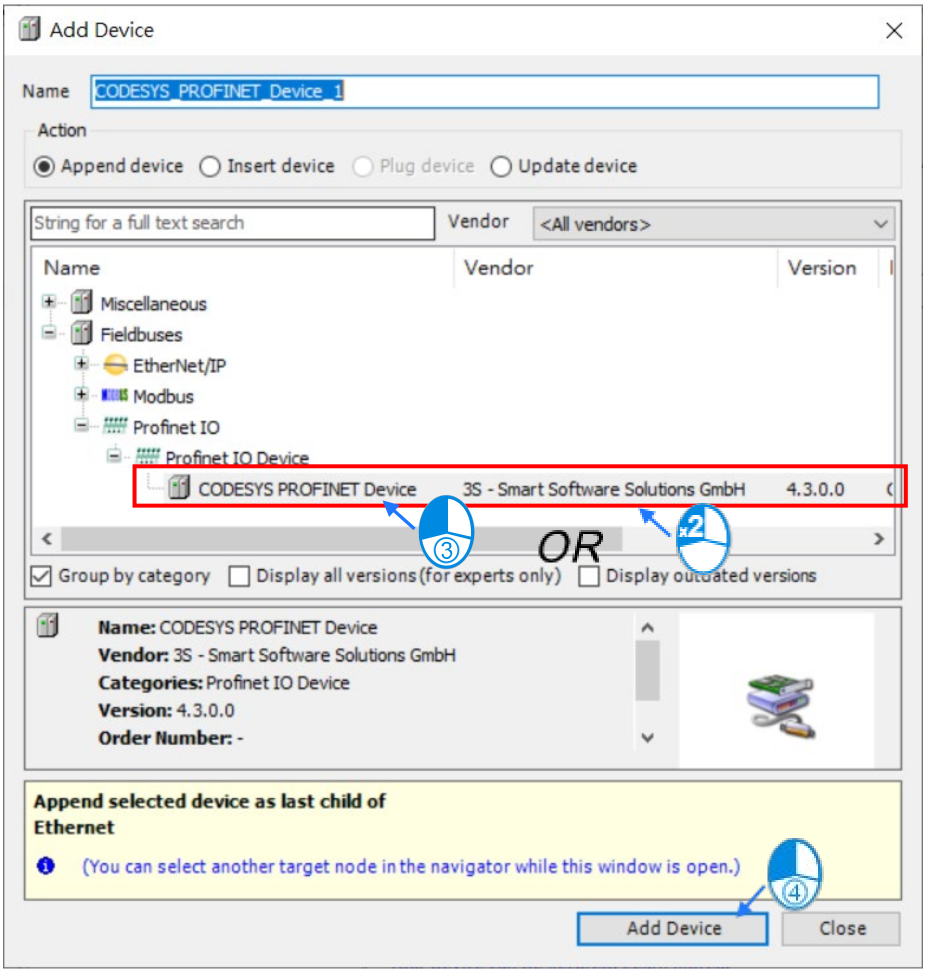
8.5.2.1 Add a PROFINET IO Device

This section will explain how to use AX-5 Series PLC as a PROFINET IO Device. Follow the steps below to set up.

- **Adding a PROFINET IO Device**







Add Device

Name: CODESYS_PROFINET_Device_1

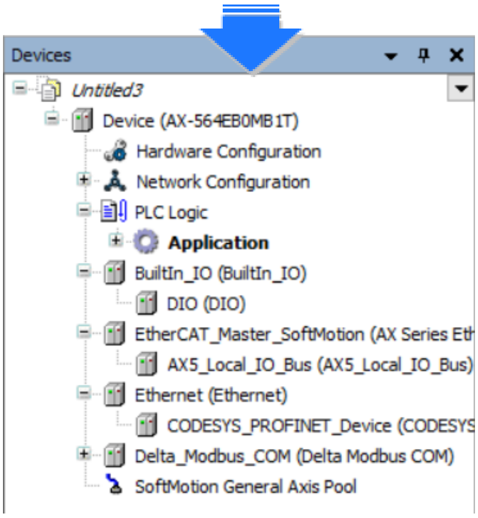
Action

● Append device ○ Insert device ○ Plug device ○ Update device

Name: CODESYS PROFINET Device
Vendor: 3S - Smart Software Solutions GmbH
Categories: Profinet IO Device
Version: 4.3.0.0
Order Number: -

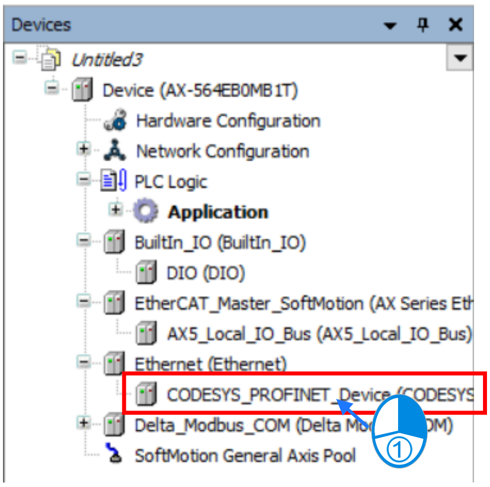
Append selected device as last child of Ethernet

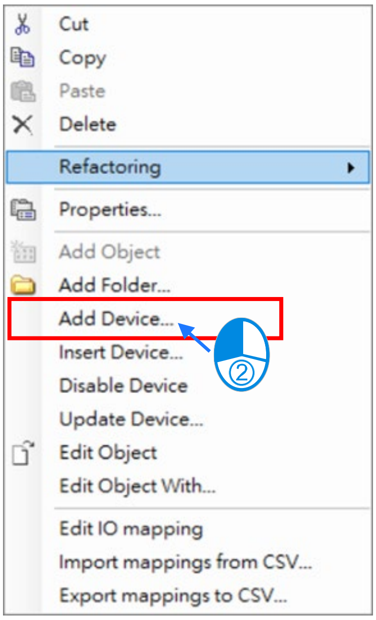
(You can select another target node in the navigator while this window is open.)

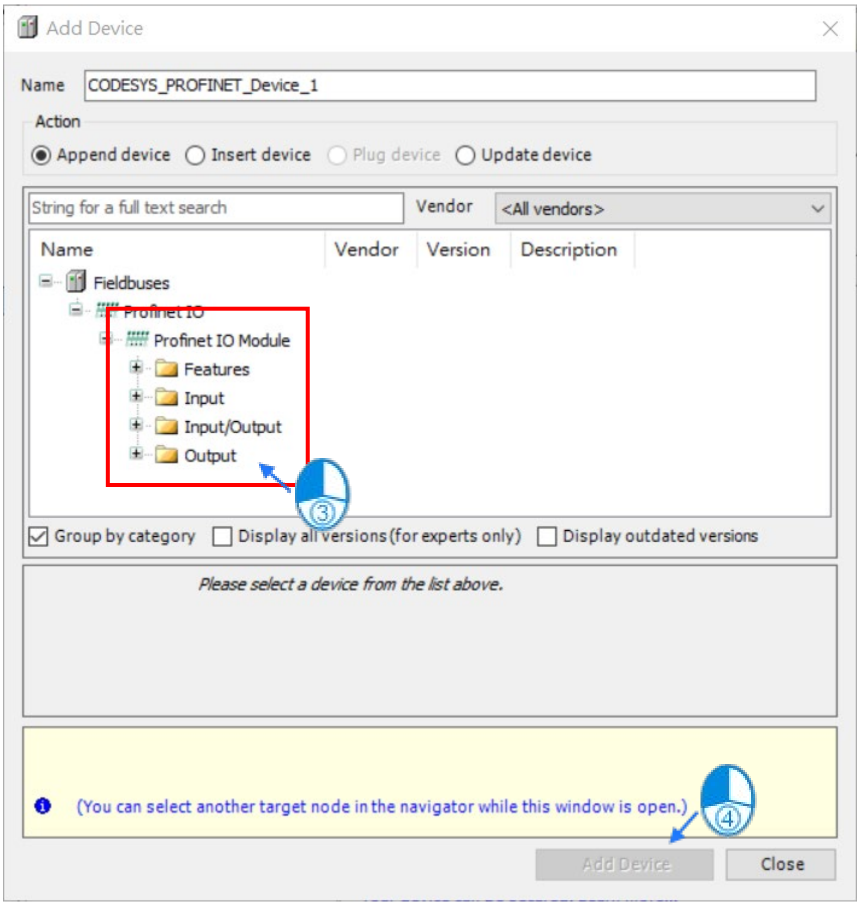


- **Adding a PROFINET IO Device**

You can customize the data structure according to your specific requirements.







Name: CODESYS_PROFINET_Device_1

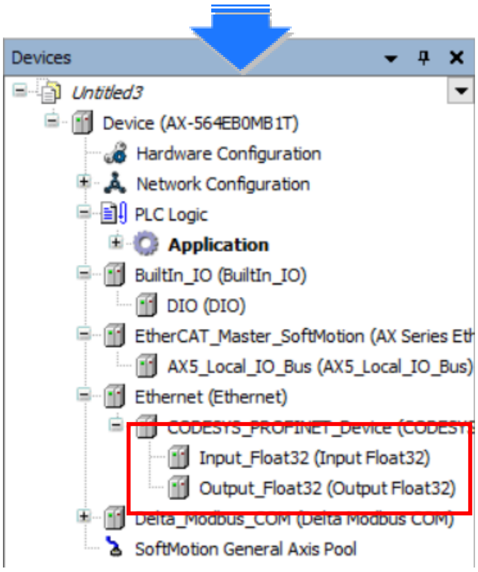
● Append device

Vendor: <All vendors>

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Please select a device from the list above.

(You can select another target node in the navigator while this window is open.)



Devices

Untitled3

Device (AX-564EB0MB1T)

Hardware Configuration

Network Configuration

PLC Logic

Application

BuiltIn_IO (BuiltIn_IO)

DIO (DIO)

EtherCAT_Master_SoftMotion (AX Series Eth

AX5_Local_IO_Bus (AX5_Local_IO_Bus)

Ethernet (Ethernet)

CODESYS_PROFINET_Device (CODESYS

Input_Float32 (Input Float32)

Output_Float32 (Output Float32)

Delta_Modbus_COM (Delta Modbus COM)

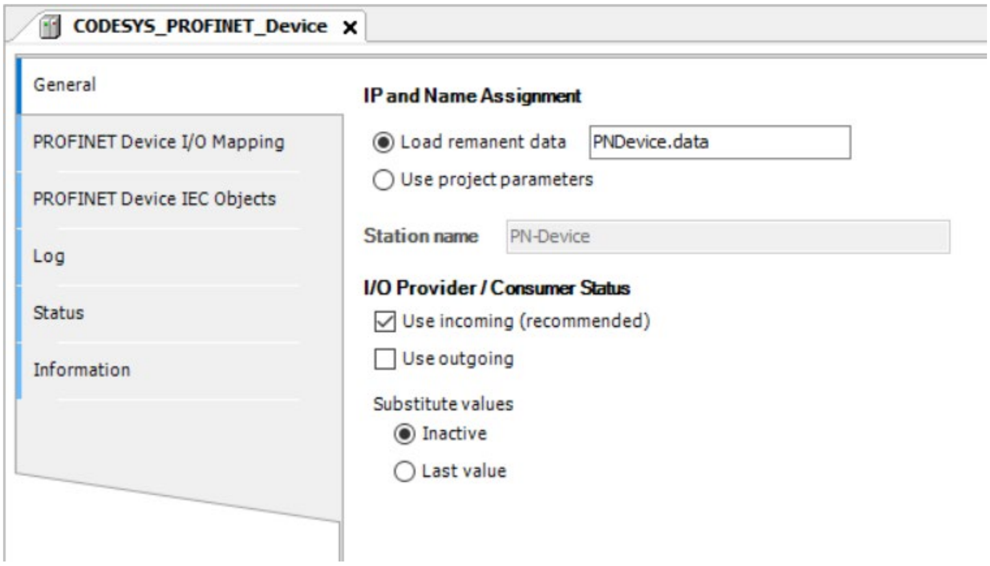
SoftMotion General Axis Pool

8.5.2.2 Settings of PROFINET Device

This section introduces the PROFINET Device setting pages.

- **General**：

Define the IP address and station name for the PROFINET Device here.



IP and Name Assignment

● Load remanent data PNDevice.data

○ Use project parameters

Station name PN-Device

I/O Provider / Consumer Status

☑ Use incoming (recommended)

☐ Use outgoing

Substitute values

● Inactive

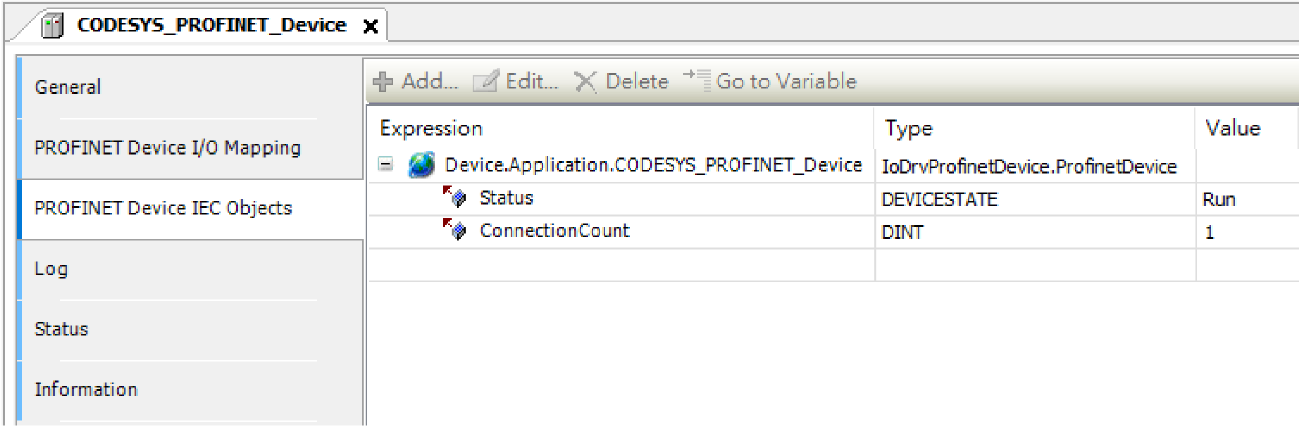
○ Last value

■ IP and Name Assignment

Setting Item	Description
Load remanent data	The IP Address and Station Name of PROFINET Device are determined by the “PNDDevice.data” from the controller. Defaults: ● IP: 0.0.0.0 ● Station name: Leave it blank.
Use project parameters	The IP address of the PROFINET Device is the same as the controller's IP address, and the Station Name can be set through this page. (Do not use this option.)

- **PROFINET Device IEC Objects**

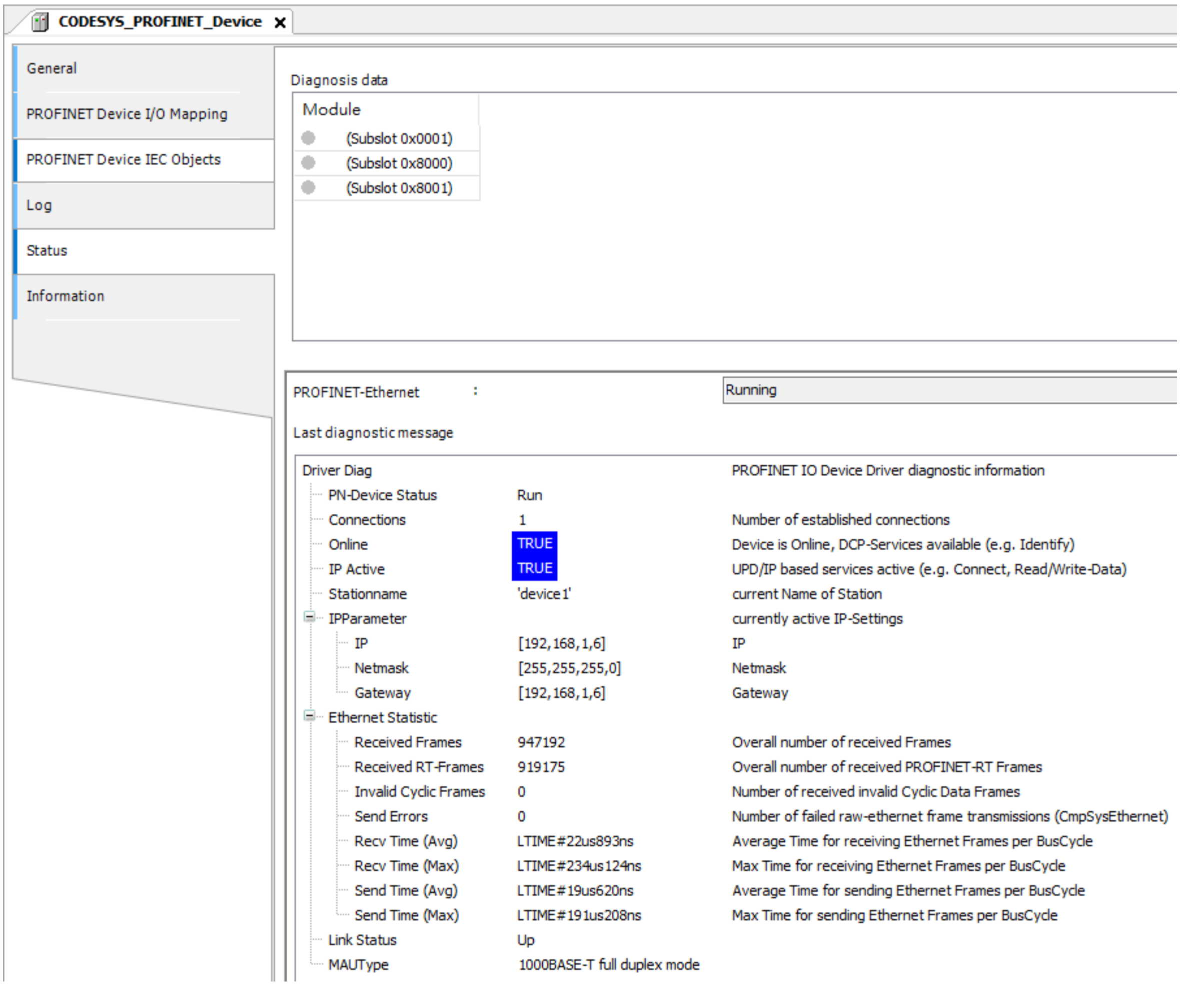
This page displays the status and number of connections of the PROFINET Device.



Expression	Type	Value
Device.Application.CODESYS_PROFINET_Device	IoDrvProfinetDevice.ProfinetDevice	
Status	DEVICESTATE	Run
ConnectionCount	DINT	1

- **Status**

You can check the running status of the PROFINET Device on this page, including IP address, Station Name, time intervals for packet transmission, and many more to help you get to know the current operation.



Diagnosis data

Module
(Subslot 0x0001)
(Subslot 0x8000)
(Subslot 0x8001)

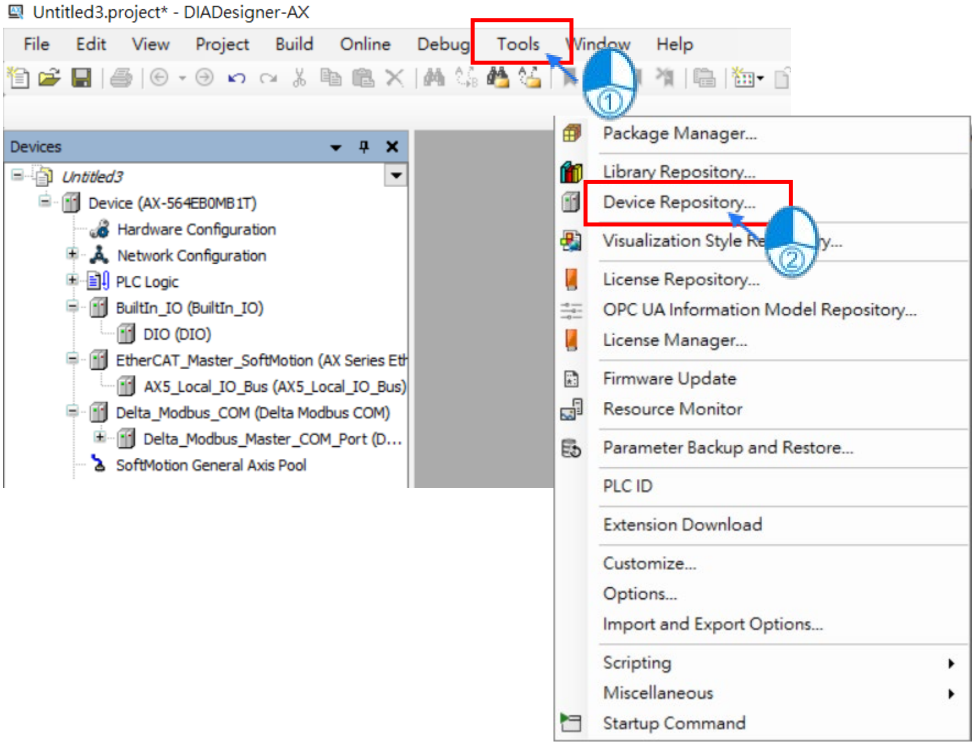
PROFINET-Ethernet : Running

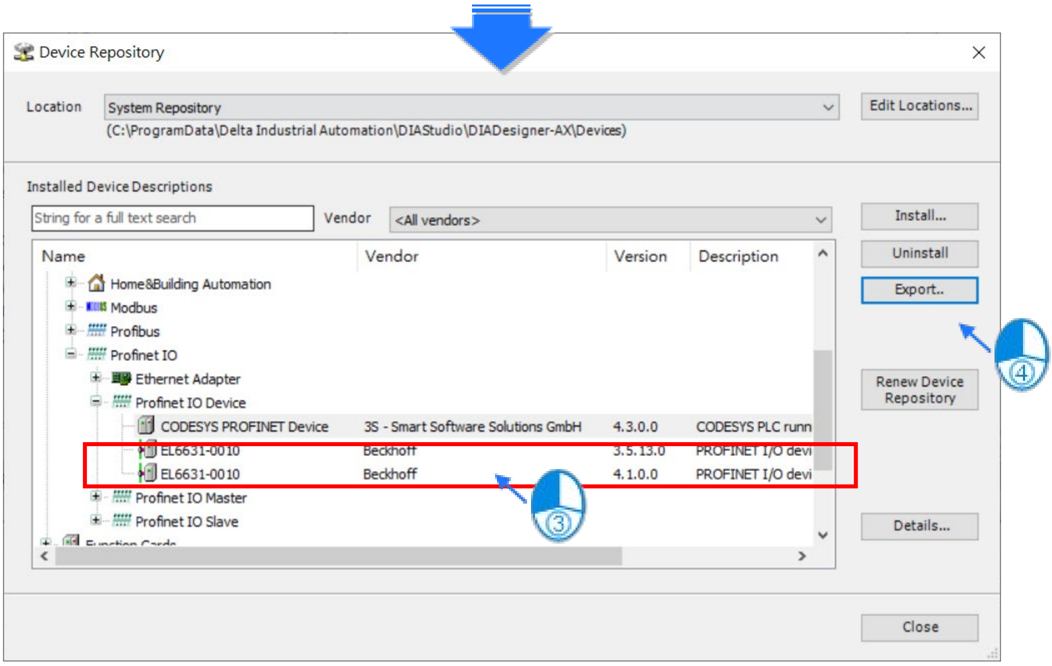
Last diagnostic message

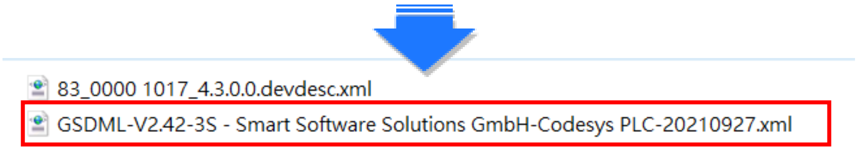
Driver Diag		PROFINET IO Device Driver diagnostic information
PN-Device Status	Run	
Connections	1	Number of established connections
Online	TRUE	Device is Online, DCP-Services available (e.g. Identify)
IP Active	TRUE	UPD/IP based services active (e.g. Connect, Read/Write-Data)
Stationname	'device1'	current Name of Station
IPParameter		currently active IP-Settings
IP	[192,168,1,6]	IP
Netmask	[255,255,255,0]	Netmask
Gateway	[192,168,1,6]	Gateway
Ethernet Statistic		
Received Frames	947192	Overall number of received Frames
Received RT-Frames	919175	Overall number of received PROFINET-RT Frames
Invalid Cyclic Frames	0	Number of received invalid Cyclic Data Frames
Send Errors	0	Number of failed raw-ethernet frame transmissions (CmpSysEthernet)
Recv Time (Avg)	LTIME#22us893ns	Average Time for receiving Ethernet Frames per BusCycle
Recv Time (Max)	LTIME#234us124ns	Max Time for receiving Ethernet Frames per BusCycle
Send Time (Avg)	LTIME#19us620ns	Average Time for sending Ethernet Frames per BusCycle
Send Time (Max)	LTIME#191us208ns	Max Time for sending Ethernet Frames per BusCycle
Link Status	Up	
MAUType	1000BASE-T full duplex mode	

8.5.2.3 Export GSDML File

This section introduces how to export a GSDML file.







83_0000 1017_4.3.0.0.devdesc.xml

GSDML-V2.42-3S - Smart Software Solutions GmbH-Codesys PLC-20210927.xml

8.6 OPC UA Server

The standard installation of DIADesigner-AX includes an OPC UA server. You can use it to access the variable interface of the controller via a client. The OPC UA server communicates with connected OPC UA clients over a separate TCP connection. Therefore, these connections must be examined again separately with regard to security.

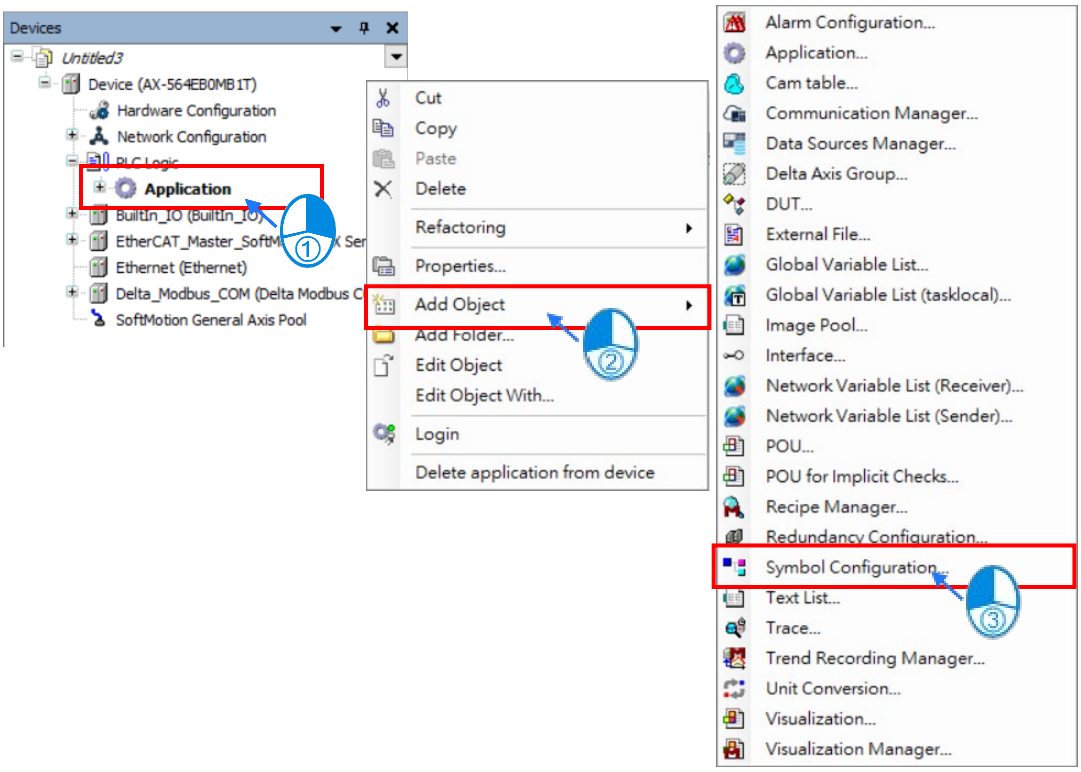
The OPC UA server can now be safeguarded by using encrypted communication to the client and OPC UA user management. See the following sections for these settings.

- Browsing of data types and variables
- Standard read/write services
- Notification for value changes: subscription and monitored item services.
- Encrypted communication according to OPC UA standard, Aes128_Sha256_RsaOaep, Basic256SHA256, and Aes256_Sha256_RsaOaep supported.

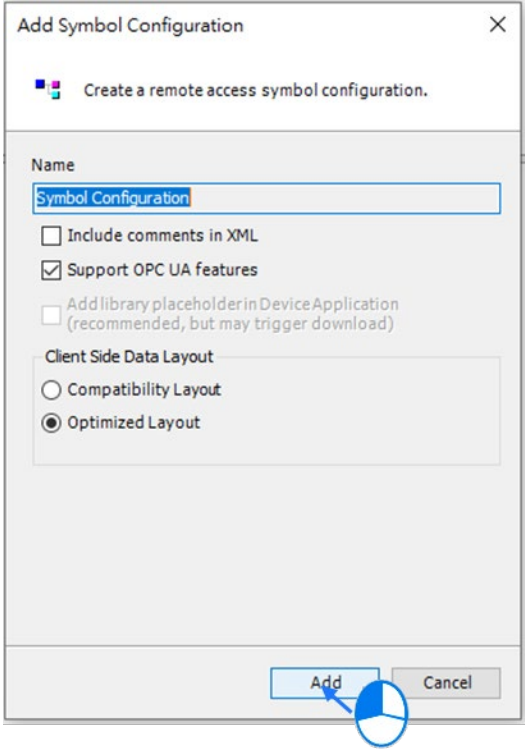
8.6.1 Setting up OPC UA Server

You need to create a project for OPC UA access before using OPC UA Server. Follow the steps below.

1. Go to *Application -> Add Object -> Symbol Configuration* to add a Symbol Configuration object.



2. Select **Support OPC UA feature** and click **Add** on the setting page of Add Symbol Configuration. After that Symbol Configuration setting page shows up automatically.



Add Symbol Configuration

Create a remote access symbol configuration.

Name: Symbol Configuration

☐ Include comments in XML

☑ Support OPC UA features

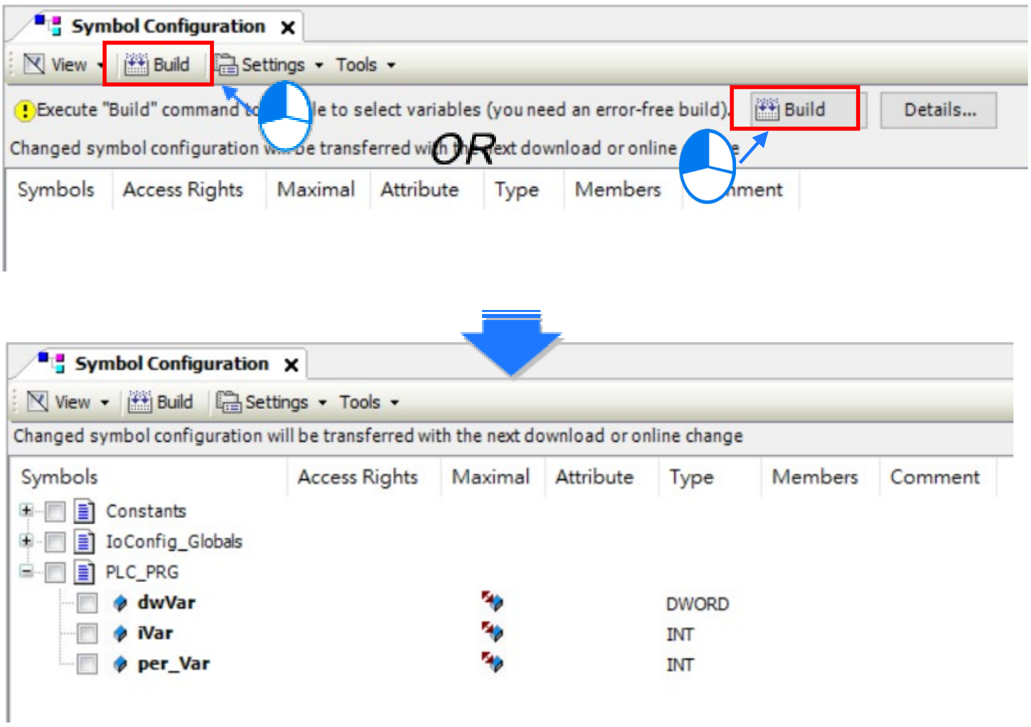
☐ Add library placeholder in DeviceApplication (recommended, but may trigger download)

Client Side Data Layout

○ Compatibility Layout

● Optimized Layout

3. Click **Build** on the Symbol Configuration setting page. The variables are shown in a tree structure.



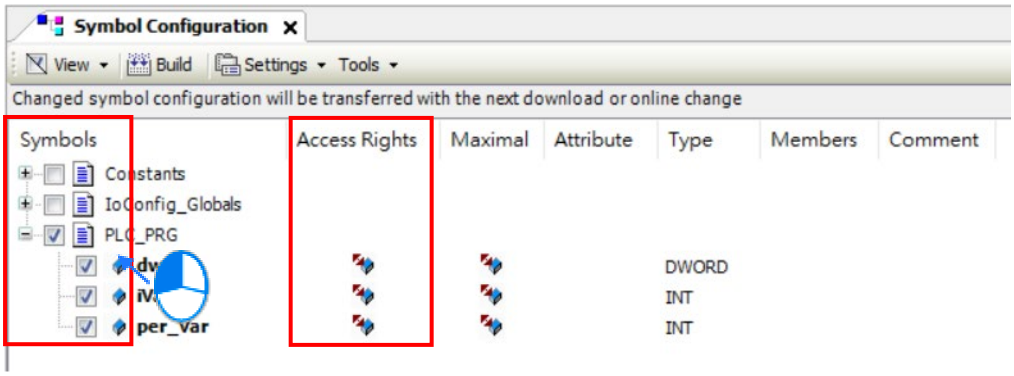
Execute "Build" command to

le to select variables (you need an error-free build).

Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
☐ Constants						
☐ IoConfig_Globals						
☐ PLC_PRG						
☐ **dwVar**				DWORD		
☐ **iVar**				INT		
☐ **per_Var**				INT		

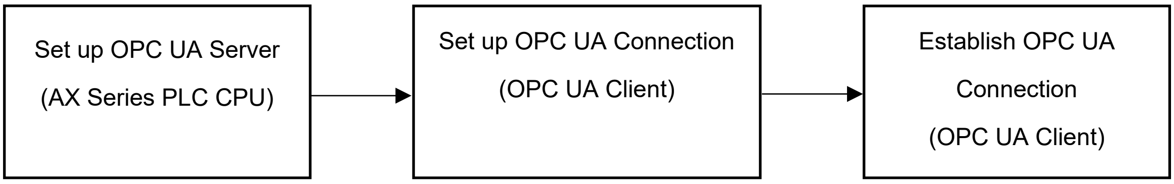
4. Select the variables that you want to change with an OPC UA client. Specify the access rights. After setting, click **Build** again. Download the project to the AX-5 Series PLC.



Changed symbol configuration will be transferred with the next download or online change

8.6.2 Setting up an Unencrypted Connection with the “UaExpert” Client

The OPC UA client “UaExpert (V1.7.1.540)” is freely accessible software. You can download the software here: https://www.unified-automation.com/downloads/opc-ua-clients.html Using this client, you can connect to the OPC UA server. After download UAExpert, follow the following steps to set up a connection. AX Series PLC acts as an OPC UA Server.



(1) Double-click the UaExpert to start the UaExpert.

(2) Right-click **Servers** and then click **Add** to open Add Server window.

(3) Go to **Custom Discovery** -> **Double click to Add Server...>** and then type in “**opc.tcp://192.168.1.5**” in the Enter URL dialog. The default IP address for AX Series PLC CPU is 192.168.1.5.

(4) After that you can find **AX-564EB0MB1T** under the **opc.tcp://192.168.1.5.** Select **OPCUAServer@AX-564EB0MB1T** and click **OK** to close the window. If the connection type is NOT an encrypted one, the node None-None appears under the added server.

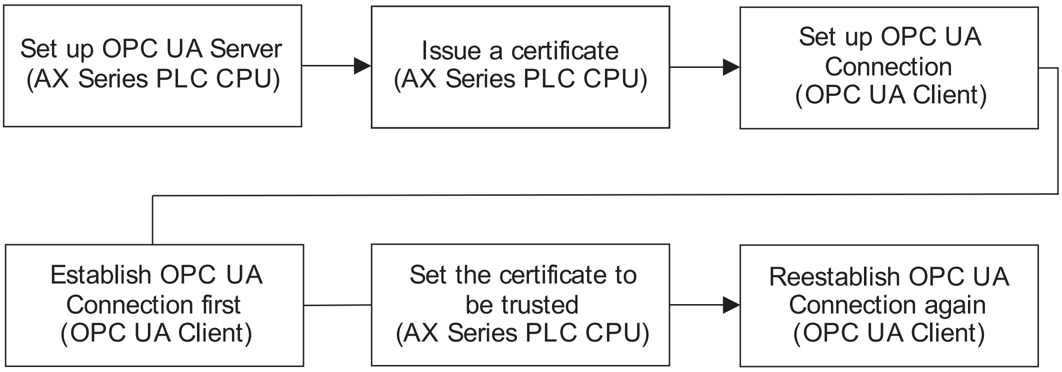
(5) If you need to edit the server properties, go back to the starting window. Expand the option **Servers** under **Project** and then right-click **AX-564EB0MB1T** to open a context menu. Click **Properties** to open the Server Settings page.

(6) Change the Endpoint URL to **opc.tcp://192.168.1.5:4840** and click **OK** to close the window.

(7) Right-click **OPCUAServer@AX-564EB0MB1T** (**opc.tcp://192.168.1.5**) to open a context menu. Click **Connect** to establish a connection to OPC UA Server.

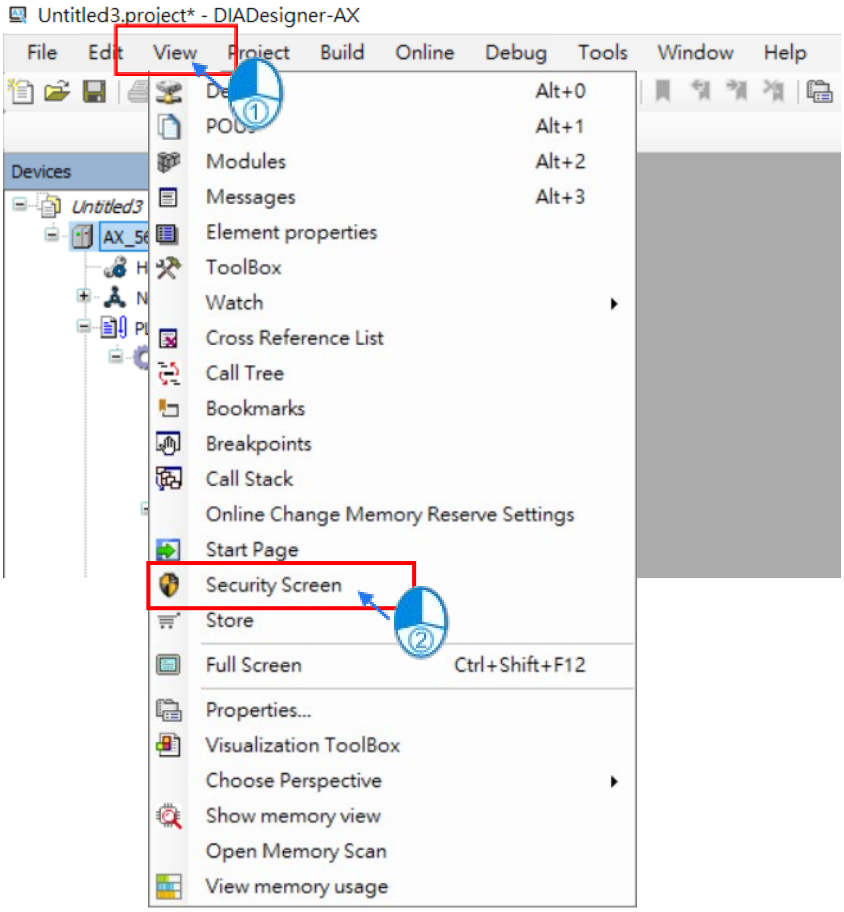
8.6.3 Setting up an Encrypted Connection with the “UaExpert” Client

The OPC UA client “UaExpert (V1.7.1.540)” is freely accessible software. You can download the software here: https://www.unified-automation.com/downloads/opc-ua-clients.html Using this client, you can connect to the OPC UA server. After download UAExpert, follow the following steps to set up a connection. AX Series PLC acts as an OPC UA Server.

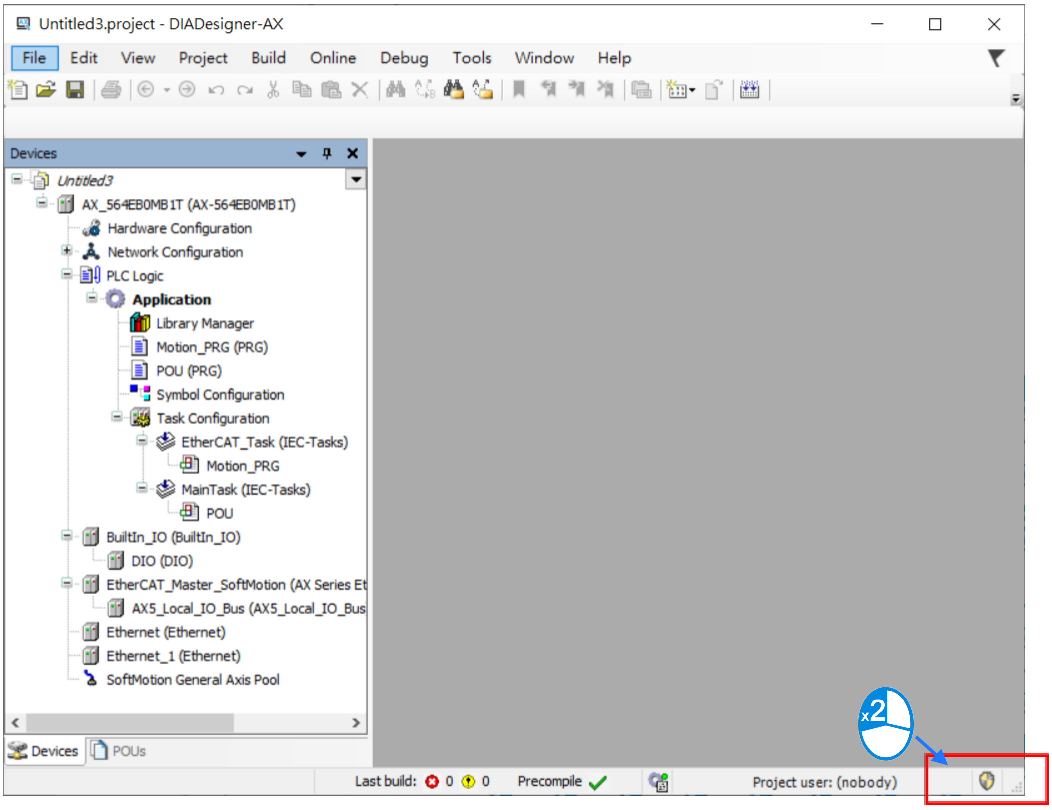


8.6.3.1 Issue a Certificate

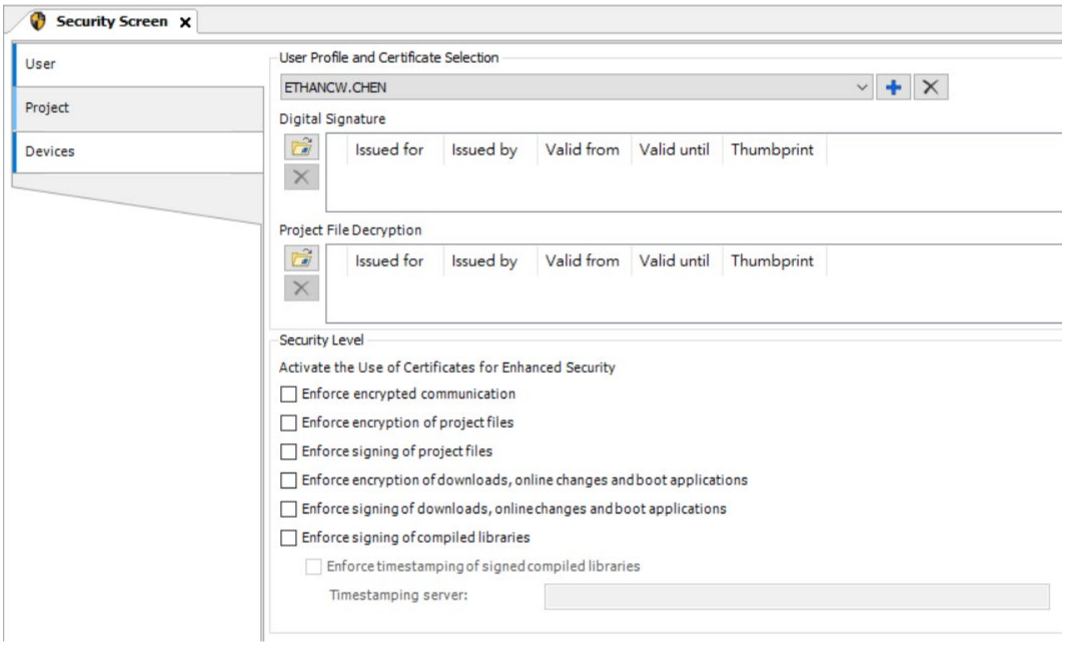
(1) Open DIADesigner-AX to create a project. Click **View** on the toolbar and then click the option **Security Screen** to open the setting page.



Or you can double-click the icon to open the **Security Screen** setting page.

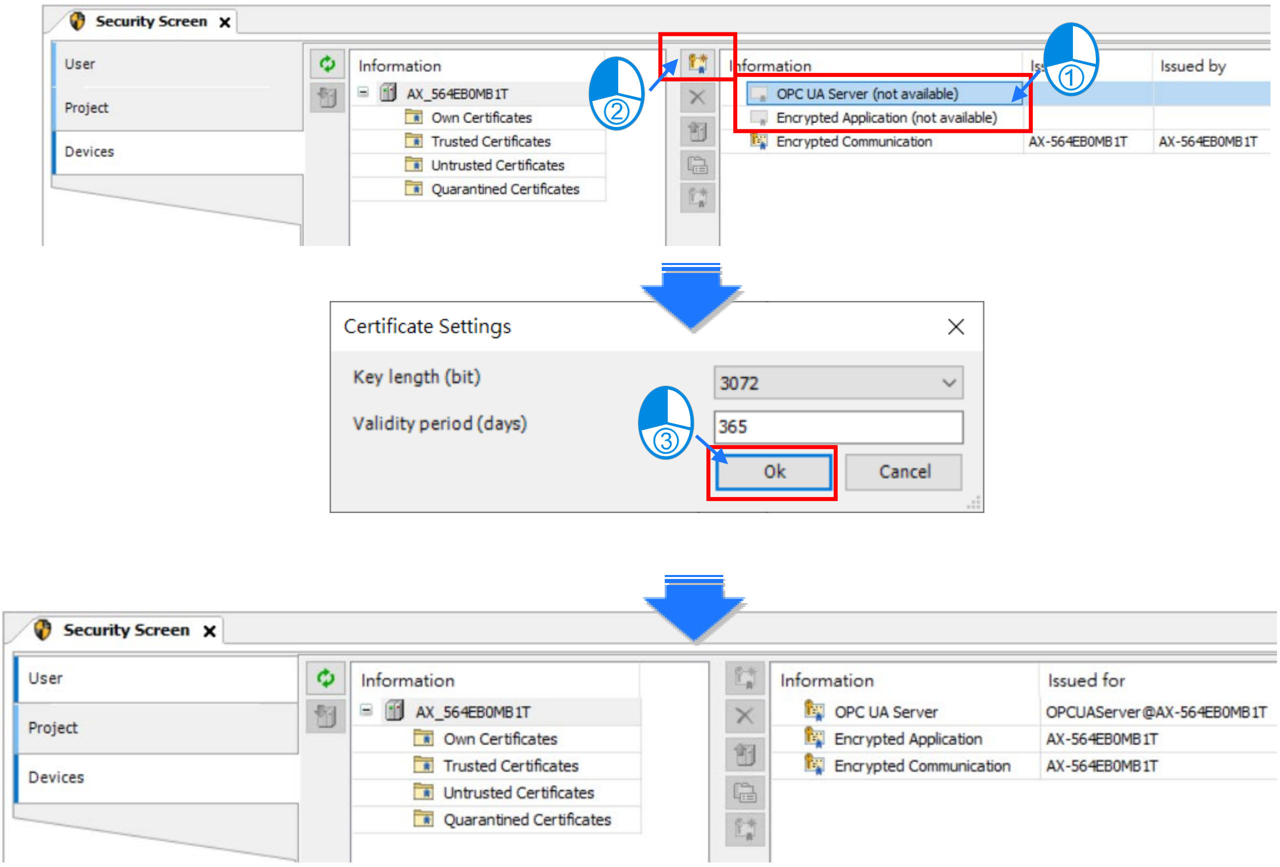


(2) Select the **Devices** tab.



(3) To have the OPC UA Server and Encrypted Application take effect, select the service **OPC UA Server** and then click to open the **Certificate Settings** page for the creation of a new certificate for the device. After setting up the certificate parameters, click **OK**. And the certificate is created on the controller. Once the settings have been configured, proceed to restart the PLC CPU. The certificate feature will only be fully activated after the PLC CPU has been restarted.

Note: Since the certificate comes with a specified validity period, it is important to ensure that the PLC CPU's system clock is accurate to prevent certificate expiration leading to the inability to establish an OPC UA connection.



Key length (bit): 3072

Validity period (days): 365

8.6.3.2 Example of Setting up an OPC UA Client (UaExpert)

(1) Double-click the UaExpert to start the UaExpert.

(2) Right-click **Servers** and then click **Add** to open Add Server window.

(3) Go to **Custom Discovery** -> **Double click to Add Server...>** and then type in “**opc.tcp://192.168.1.5**” in the Enter URL dialog. The default IP address for AX Series PLC CPU is 192.168.1.5.

(4) After that you can find **OPCUAServer@AX-564EB0MB1T(opc.tcp://192.168.1.5** under the **opc.tcp://192.168.1.5.** Select it and click **OK** to close the window. If the connection type is NOT an encrypted one, the node None-None appears under the added server.

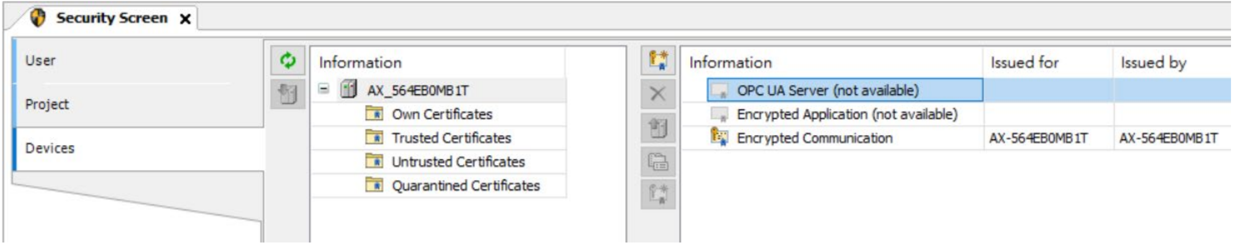
(5) If you need to edit the server properties, go back to the starting window. Expand the option **Servers** under **Project** and then right-click **AX-564EB0MB1T** to open a context menu. Click **Properties** to open the Server Settings page.

(6) Change the Endpoint URL to **opc.tcp://192.168.1.5:4840** and click **OK** to close the window.

(7) Right-click **OPCUAServer@AX-564EB0MB1T** (**opc.tcp://192.168.1.5**) to open a context menu. Click **Connect** to establish a connection to OPC UA Server.

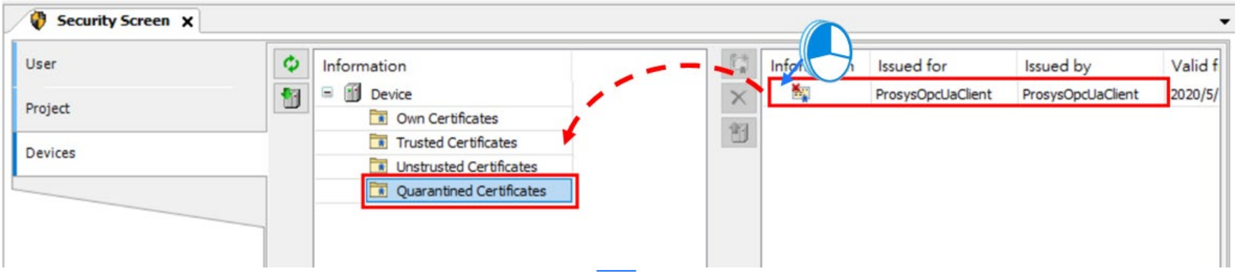
8.6.3.3 Set the Certificate to be Trusted

(1) Go back to DIADesigner-AX. Click **View** on the toolbar and then click the option **Security Screen** to open the setting page. Select the **Devices** tab.

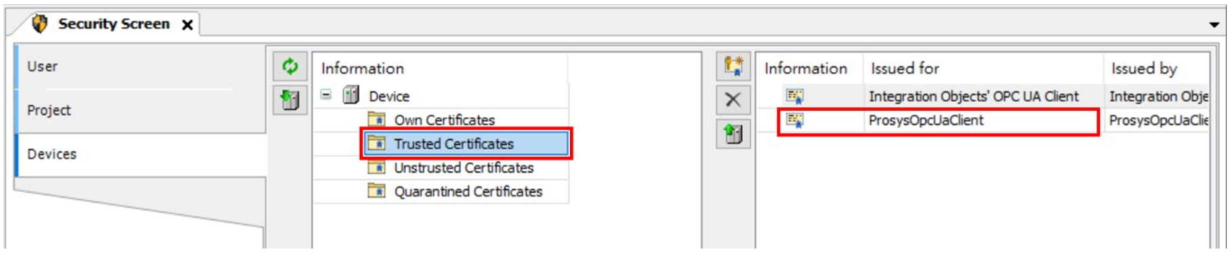


Information	Issued for	Issued by
OPC UA Server (not available)		
Encrypted Application (not available)		
Encrypted Communication	AX-564EB0MB1T	AX-564EB0MB1T

(2) Click ⟳ to refresh and all services of the controller that require a certificate are displayed in the right view. Find **ProsysOpcUaClient** in the folder of **Quarantined Certificates**. Drag it to the folder of **Trusted Certificates**.



Infor	Issued for	Issued by	Valid f
	ProsysOpcUaClient	ProsysOpcUaClient	2020/5/



Information	Issued for	Issued by
	Integration Objects' OPC UA Client	Integration Obje
	ProsysOpcUaClient	ProsysOpcUaClie

8.7 CANopen

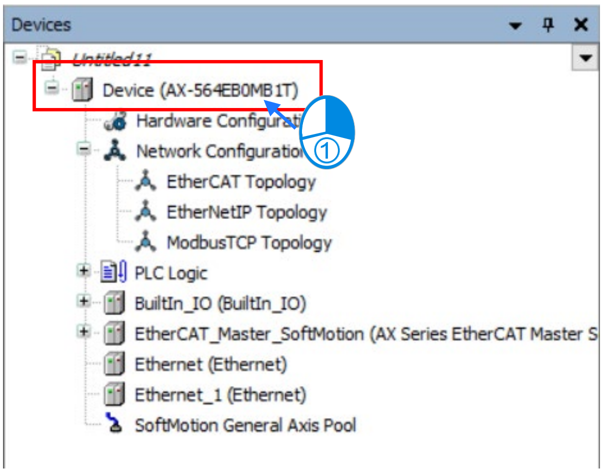
8.7.1 Introduction on CANopen

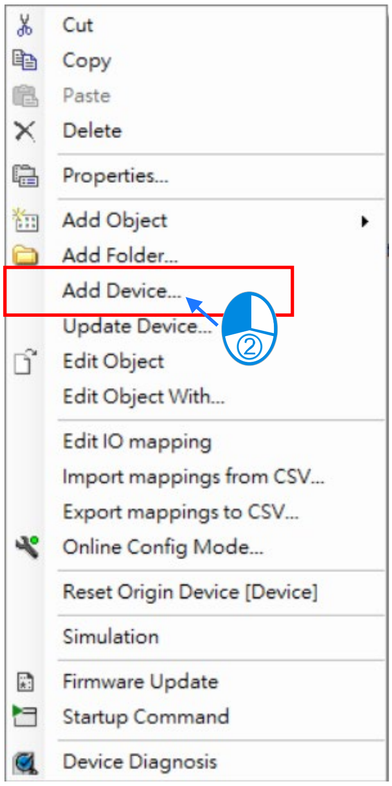
CANopen is an open-source industrial protocol built upon the CAN bus, designed for efficient inter-device communication. Its real-time performance, reliability, and cost-effectiveness make it a popular choice for a wide range of applications, including industrial automation, automotive systems, and medical devices.

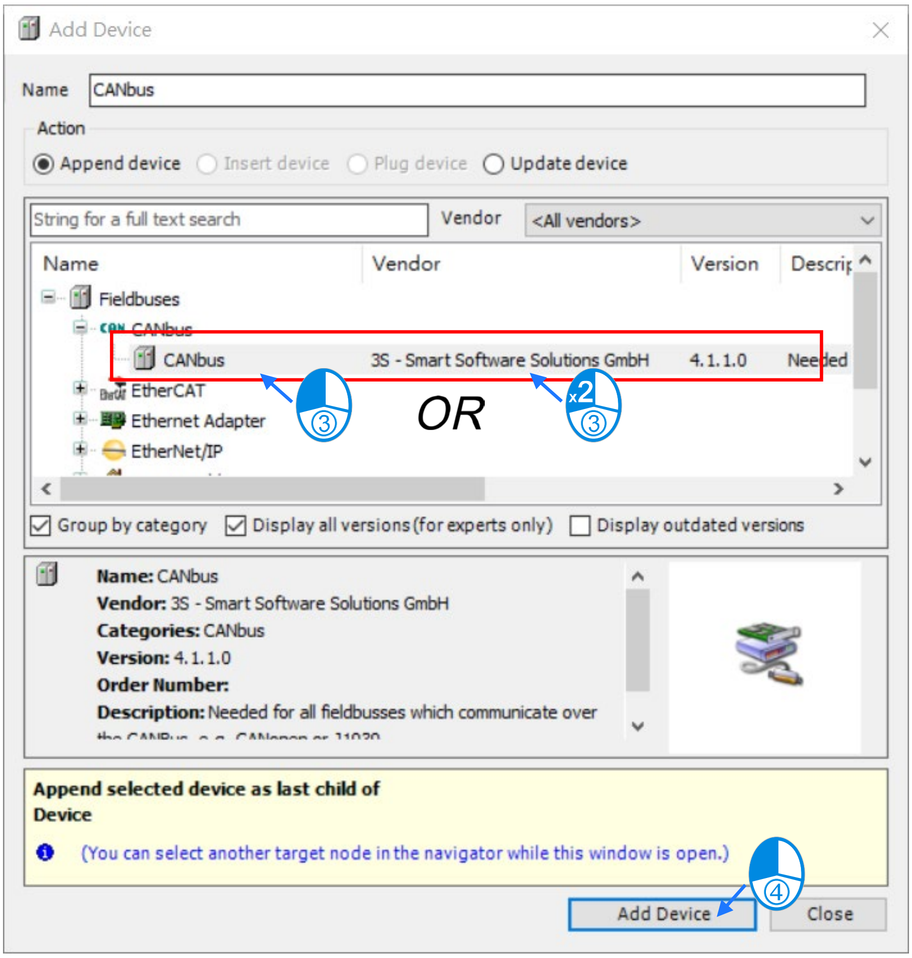
8.7.2 Creating a CANbus Component

To enable CANopen features, a CANbus component must be added to the device tree. The node ID and baud rate should be configured based on the specific needs of the application.

- Adding a CANbus component







Name: CANbus

● Append device ○ Insert device ○ Plug device ○ Update device

String for a full text search   Vendor <All vendors>

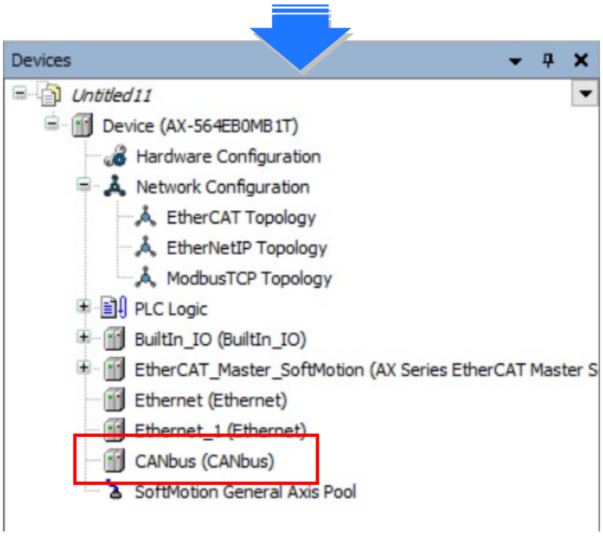
Name	Vendor	Version	Descrip
CANbus	3S - Smart Software Solutions GmbH	4.1.1.0	Needed

☑ Group by category ☑ Display all versions (for experts only) ☐ Display outdated versions

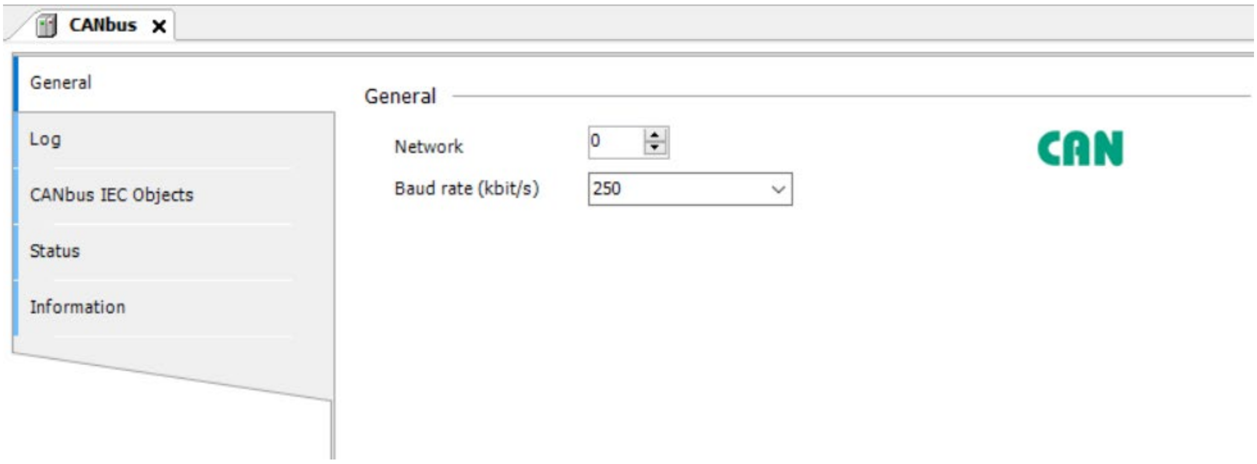
Name: CANbus
Vendor: 3S - Smart Software Solutions GmbH
Categories: CANbus
Version: 4.1.1.0
Order Number:
Description: Needed for all fieldbusses which communicate over

Append selected device as last child of
Device

(You can select another target node in the navigator while this window is open.)



- Setting up CANbus



Network 0

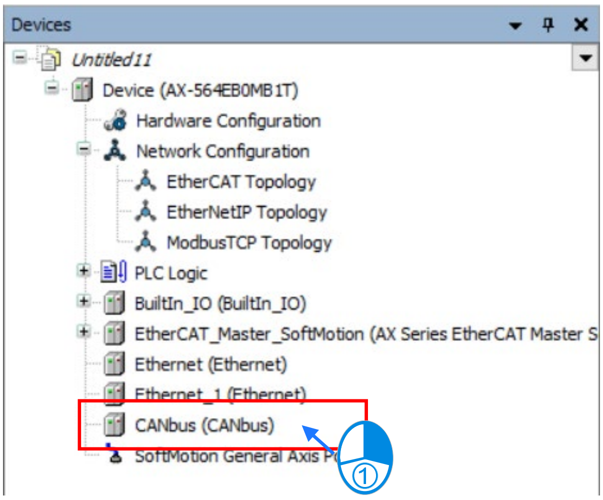
Baud rate (kbit/s) 250

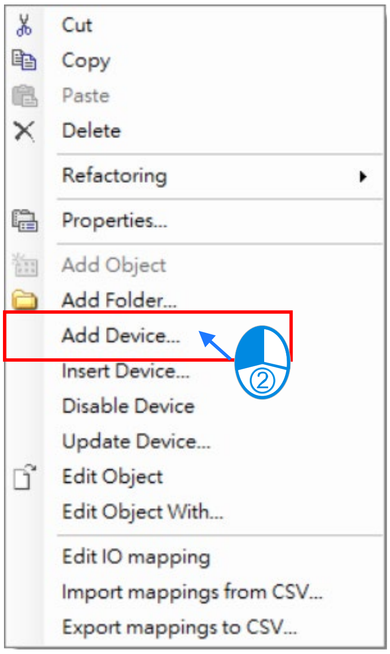
Tab	Description
General	Network: Code for the CAN bus interface. (set to 0) Baud rate: CAN communication baud rate
Log	CANbus component operation log
Status	CANbus component current operation status
Information	The current CANbus component information includes manufacturer, firmware version, and more.

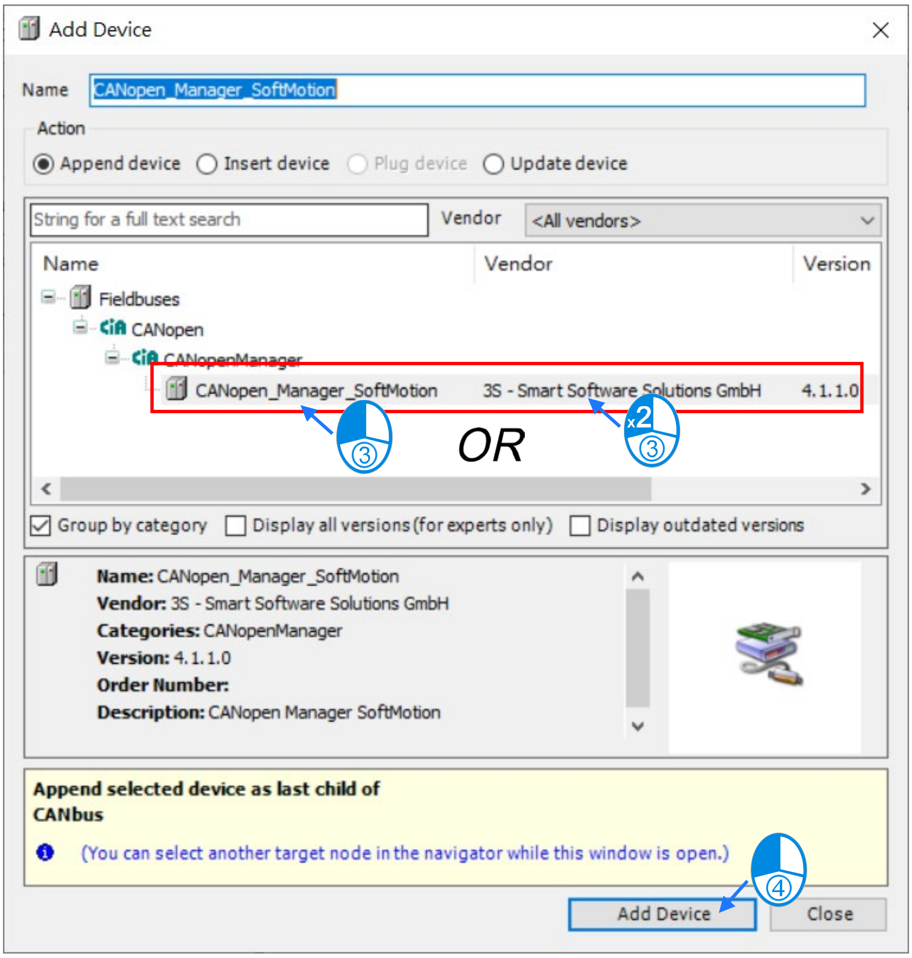
8.7.3 Creating a CANopen Manager

This section shows how to set up the AX-5 Series PLC to function as a CANopen Manager.

- Adding a CANopen Manager







Name: CANopen_Manager_SoftMotion

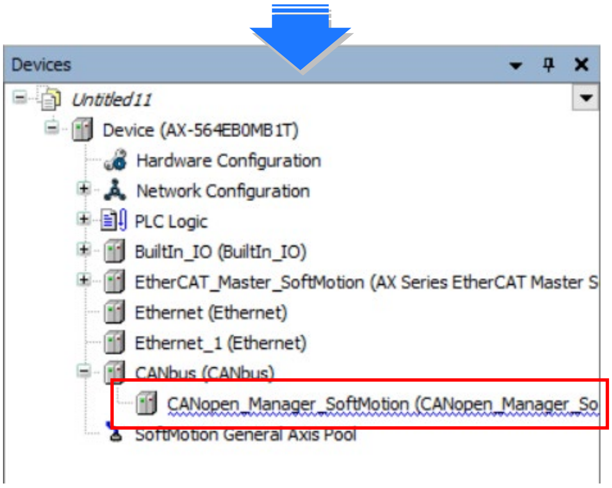
Name	Vendor	Version
CANopen_Manager_SoftMotion	3S - Smart Software Solutions GmbH	4.1.1.0

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

Name: CANopen_Manager_SoftMotion
Vendor: 3S - Smart Software Solutions GmbH
Categories: CANopenManager
Version: 4.1.1.0
Order Number:
Description: CANopen Manager SoftMotion

Append selected device as last child of
CANbus

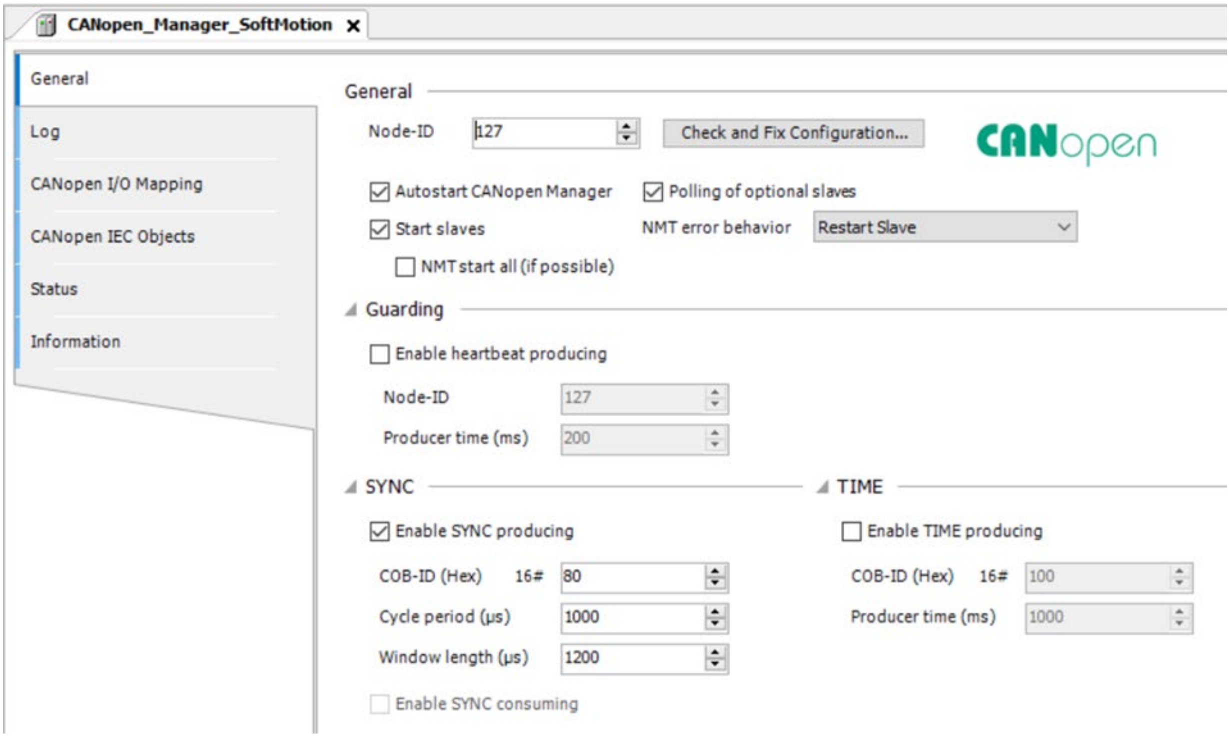
(You can select another target node in the navigator while this window is open.)



Devices

Untitled11
- Device (AX-564EB0MB1T)
  - Hardware Configuration
  - Network Configuration
  - PLC Logic
  - BuiltIn_IO (BuiltIn_IO)
  - EtherCAT_Master_SoftMotion (AX Series EtherCAT Master S
  - Ethernet (Ethernet)
  - Ethernet_1 (Ethernet)
  - CANbus (CANbus)
    - CANopen_Manager_SoftMotion (CANopen_Manager_So
  - SoftMotion General Axis Pool

8.7.4 Setting up CANopen Manager



General

Node-ID 127

☑ Autostart CANopen Manager

☑ Polling of optional slaves

☑ Start slaves

NMT error behavior: Restart Slave

☐ NMT start all (if possible)

Guarding

☐ Enable heartbeat producing

Node-ID 127

Producer time (ms) 200

SYNC

☑ Enable SYNC producing

COB-ID (Hex) 16# 80

Cycle period (μs) 1000

Window length (μs) 1200

☐ Enable SYNC consuming

TIME

☐ Enable TIME producing

COB-ID (Hex) 16# 100

Producer time (ms) 1000

Tab	Category	Description
General	General	• Node-ID: CANopen Manager Node ID (1 to 127) • Check and Fix Configuration: Validate and correct the settings. • Autostart CANopen Manager: ☑ : CANopen Manager: The CANopen Manager starts automatically (switches to OPERATIONAL mode) after all required slaves are ready. ☐ : The CANopen Manager starts from the function block CiA405 NMT. • Polling of optional slaves: When a slave does not respond during the boot sequence, the CANopen Manager interrogates it every second until it does respond. • Start slaves： ☑ : CANopen Manager is responsible for starting the slaves. ☐ : Start the slaves from the function block CiA405 NMT. • NMT start all (if possible)： ☑ : If the Start slaves option is activated, then the CANopen Manager starts all slaves with an "NMT Start All" command. • NMT error behavior： Restart Slave: If an error occurs during slave monitoring, then the slave is restarted automatically. Stop Slave: If an error occurs during slave monitoring, then the slave is stopped.
	Guarding	• Enable heartbeat producing: Working with heartbeat message is an alternative method of monitoring. It can be executed from both master and slave nodes. • Node-ID: Unique identification (1 to 127) of the heartbeat producer on the bus. • Producer time (mms): Interval length between successive heartbeats.
	SYNC	• Enable SYNC producing: The CANopen Manager sends SYNC telegrams.

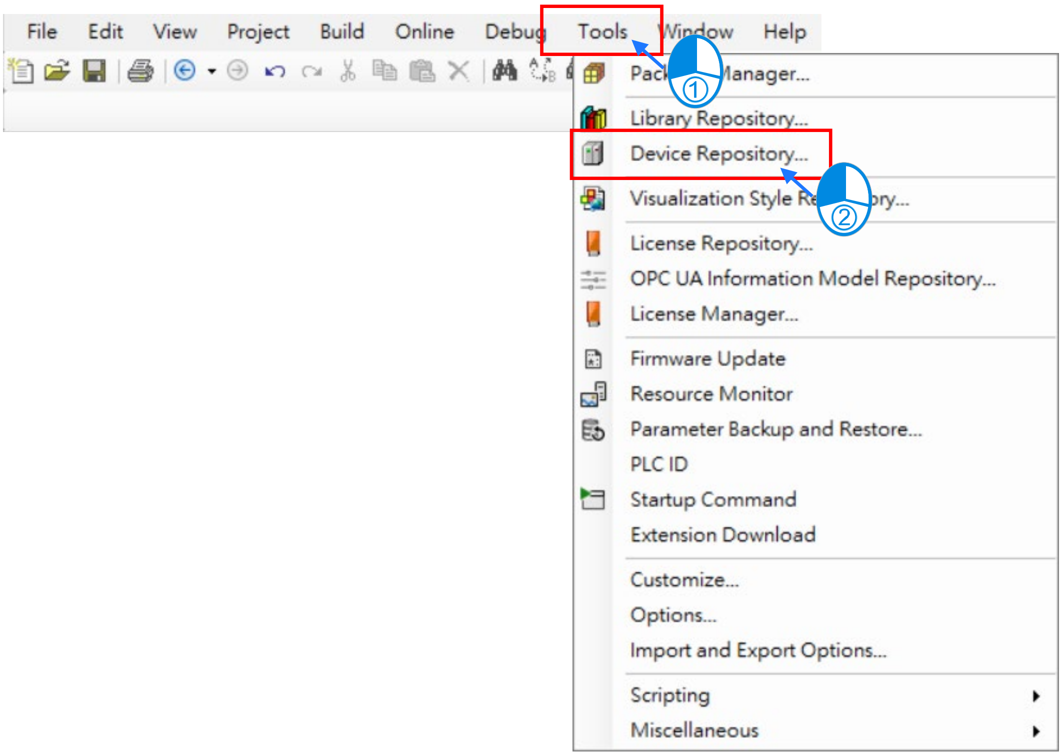
		• COB-ID (Hex): CAN-ID of the SYNC telegram. Range: 1 to 2047. • Cycle period (µs): Interval length in microseconds after which the SYNC telegram is sent. • Window length (µs): Length of the time frame for synchronous PDOs. • Enable SYNC consuming: Another device must produce the SYNC telegrams that are received by the CANopen Manager.
	TIME	• Enable TIME producing: CANopen Manager sends TIME messages. • COB-ID (Hex): It identifies the time stamp of the message. Range: 0 to 2047. * COB：Communication Object Identifier。 • Producer time (ms): Interval when the time stamp is sent. This value has to be a multiple of the task cycle time.

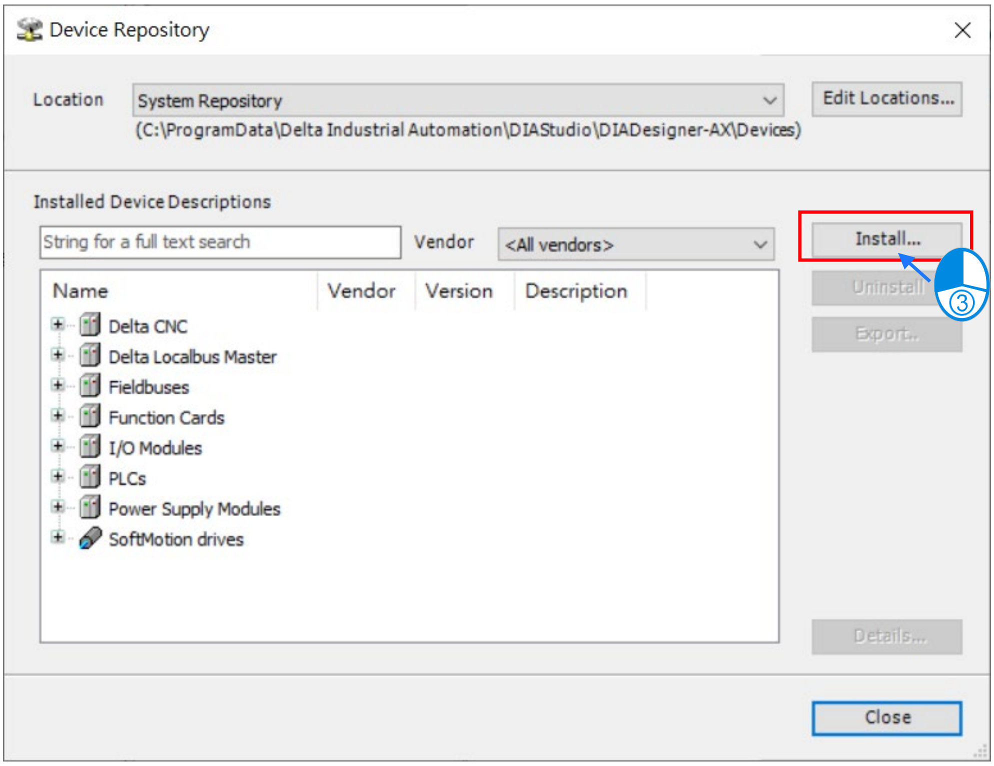
8.7.5 Demonstration of CANopen Manager Feature

8.7.5.1 Example of AS200 Series PLC CPU

- Import the EDS file of AS200 Series PLC CPU.

The EDS file of AS200 Series PLC CPU can be obtained from the Download Center of the Delta official website.





Device Repository

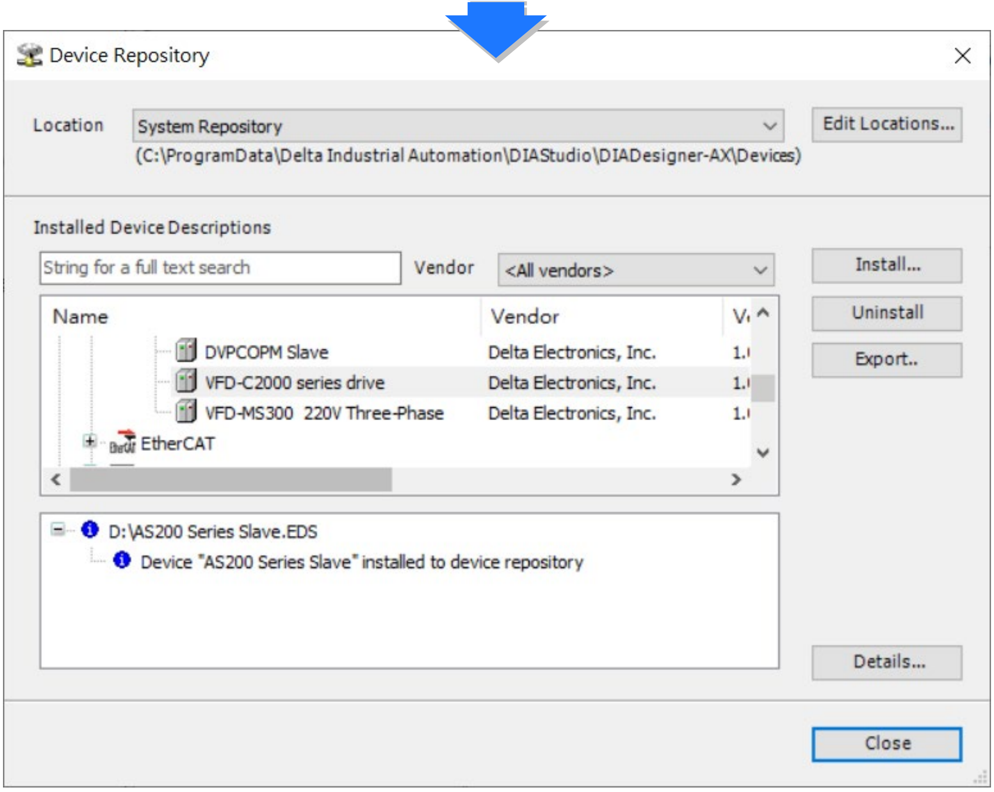
Location: System Repository
(C:\ProgramData\Delta Industrial Automation\DIAStudio\DIADesigner-AX\Devices)

Installed Device Descriptions

String for a full text search

Vendor: <All vendors>

Name	Vendor	Version	Description
Delta CNC			
Delta Localbus Master			
Fieldbuses			
Function Cards			
I/O Modules			
PLCs			
Power Supply Modules			
SoftMotion drives			



Device Repository

Location: System Repository
(C:\ProgramData\Delta Industrial Automation\DIAStudio\DIADesigner-AX\Devices)

Installed Device Descriptions

String for a full text search

Vendor: <All vendors>

Name	Vendor	V
DVPCOPM Slave	Delta Electronics, Inc.	1.
VFD-C2000 series drive	Delta Electronics, Inc.	1.
VFD-MS300 220V Three-Phase	Delta Electronics, Inc.	1.
EtherCAT		

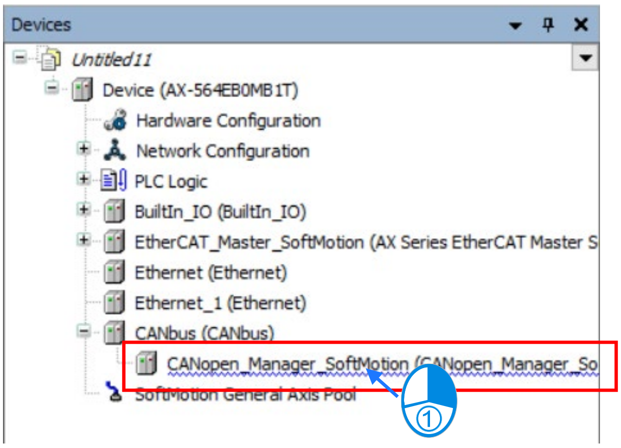
D:\AS200 Series Slave.EDS
Device "AS200 Series Slave" installed to device repository

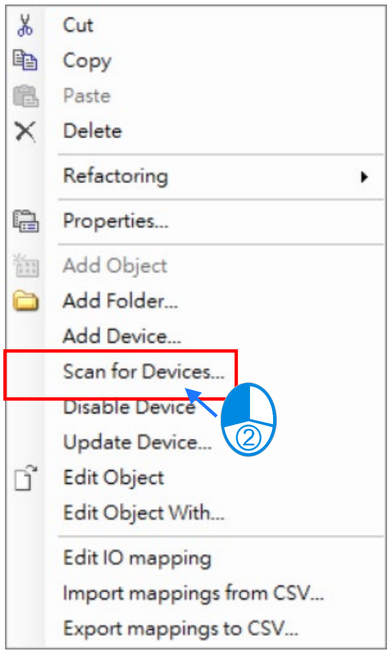
- Add a CANopen Slave

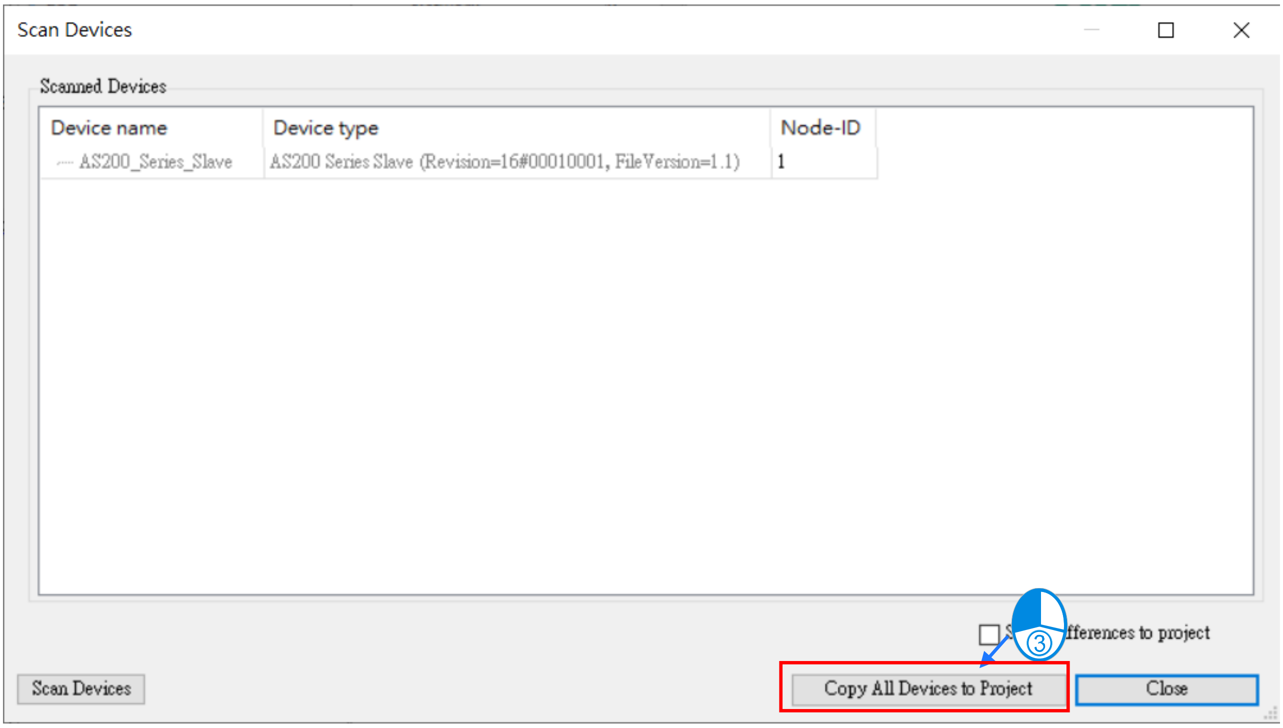
There are two ways to add CANopen devices: “Scan for Devices” and “Add Device”. This example will demonstrate the commonly used method “Scan for Devices”. Please note that before using the Scan for Devices method, you must download a project that includes the CANopen Manager component.

Note 1: Refer to Chapter 10 CANopen Function and Operation in the AS Series Hardware and Operation Manual for guidance on configuring the AS200 series PLC CPU as a CANopen slave.

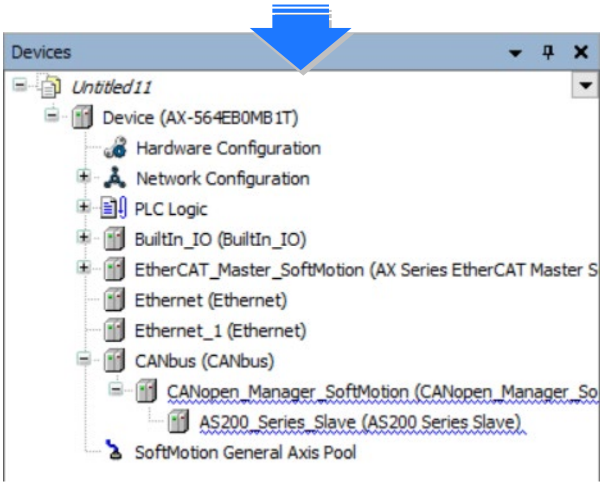
Note 2: In order for the CANopen Master to scan the CANopen Slave, both the Slave and Master must operate at the same baud rate.







Device name	Device type	Node-ID
AS200_Series_Slave	AS200 Series Slave (Revision=16#00010001, FileVersion=1.1)	1

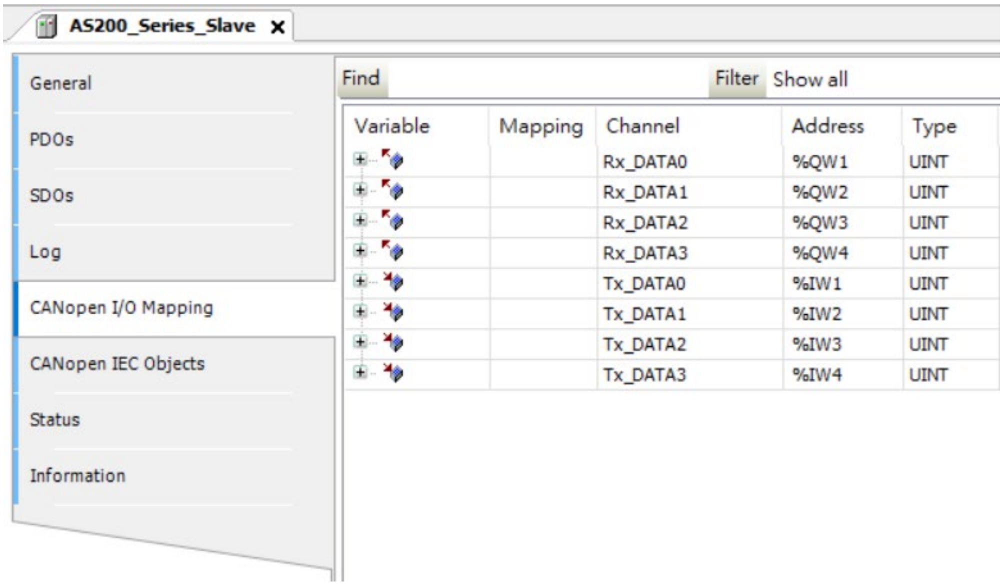


- Data mapping between Master and Slave in CANopen

When the AS Series PLC acts as a Slave, the output mapping areas are D25032–25063, and the input mapping areas are D24032–24063 as the following table shows.

Device in the PLC	Mapping area	Mapping length
D24032 to D24063	RxPDO (Master -> Slave)	64 Words
D25032 to D25063	TxPDO (Slave -> Master)	64 Words

Data mapping can be executed in CANopen I/O Mapping page of AS200_Series_Slave. Here we use the AS200 Series PLC CPU as a demonstration example.



Variable	Mapping	Channel	Address	Type
		Rx_DATA0	%QW1	UINT
		Rx_DATA1	%QW2	UINT
		Rx_DATA2	%QW3	UINT
		Rx_DATA3	%QW4	UINT
		Tx_DATA0	%IW1	UINT
		Tx_DATA1	%IW2	UINT
		Tx_DATA2	%IW3	UINT
		Tx_DATA3	%IW4	UINT

MEMO

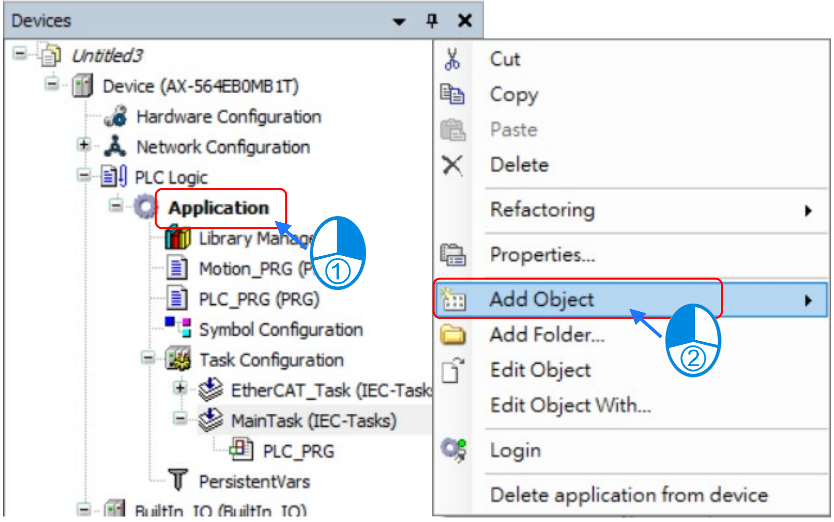
Chapter 9 Convenience Functions

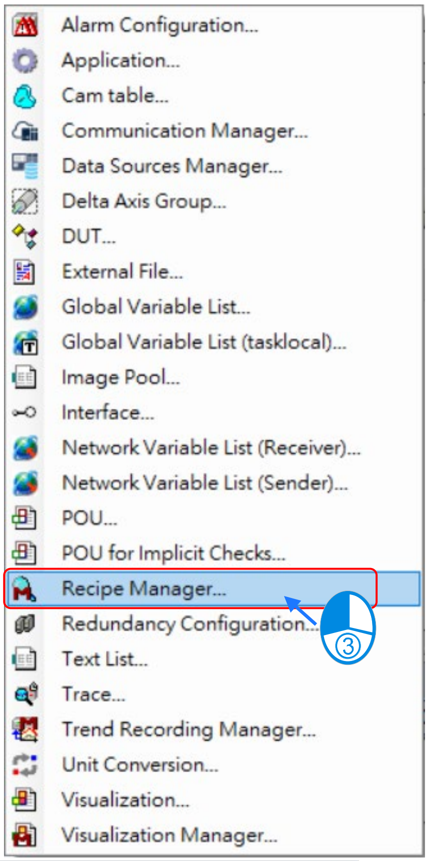
Table of Contents

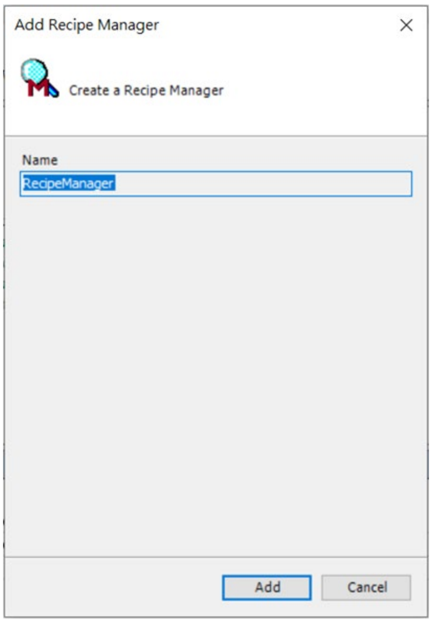
9.1 Recipe Manager

The recipe manager allows you to import recipe files into specific variables in the controller, or export specific variables from the controller to recipe files. You can use the function block "RecipeManCommands" of the "Recipe_Management.library" to perform such tasks.

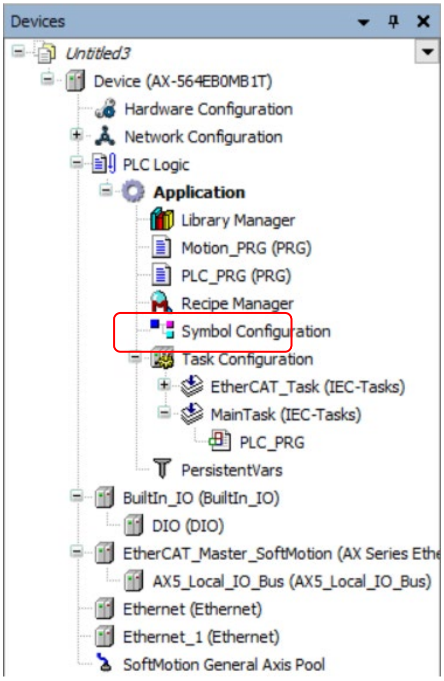
- **Add Recipe Manager**



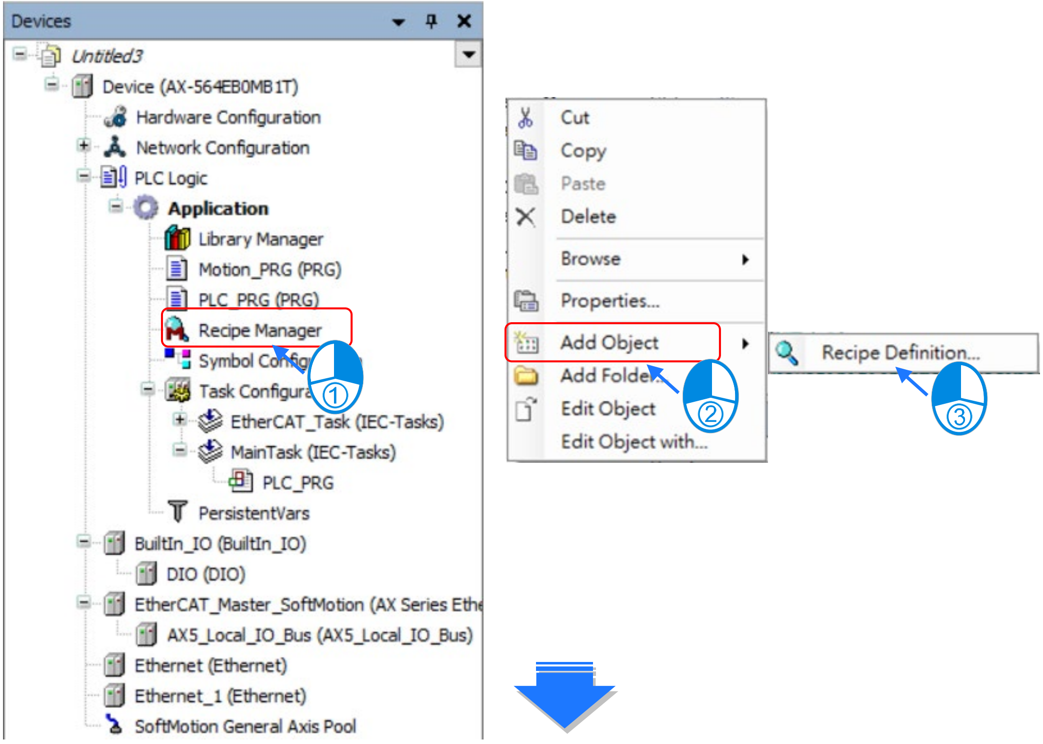


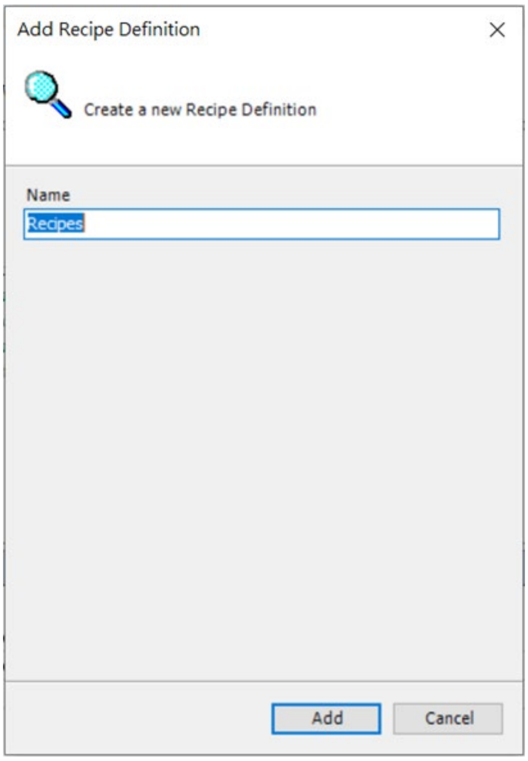


Name: RecipeManager

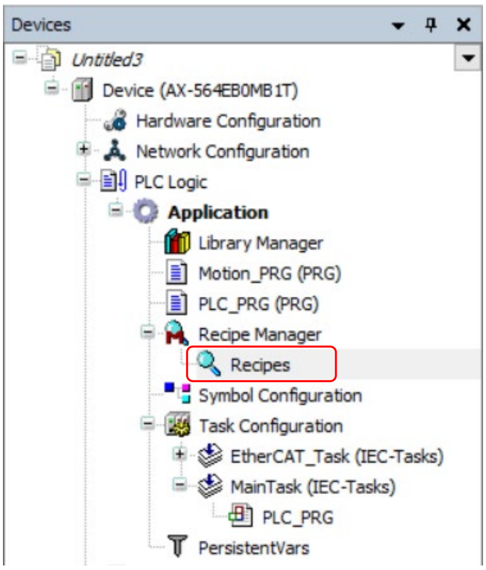


- **Add Recipe Definition**

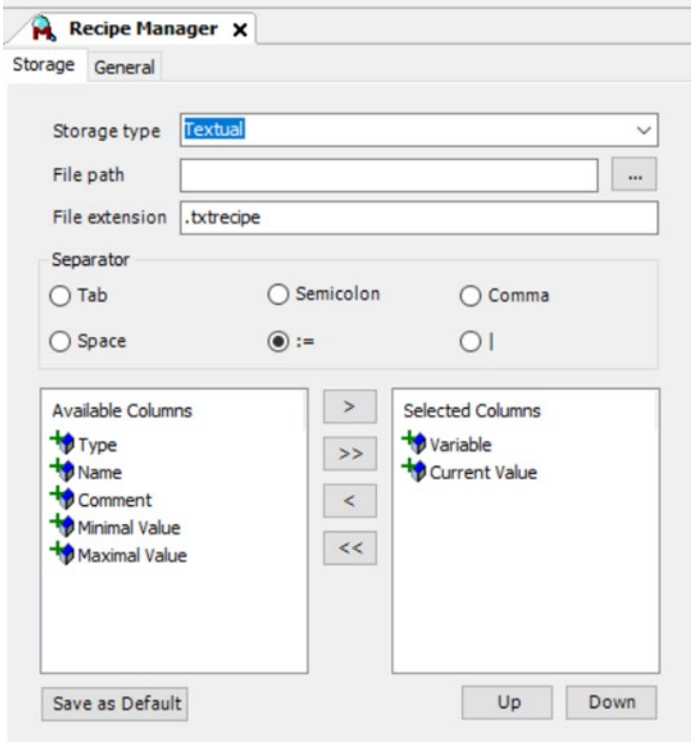




Name: Recipes



9.1.1 Configurations on the Recipe Management Page



Storage type: Textual

File path:

File extension: .txtrecipe

Separator

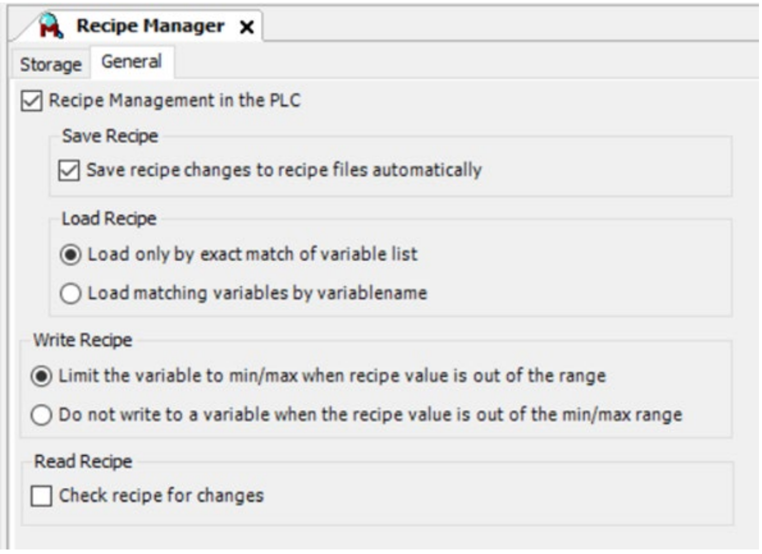
○ Tab ○ Semicolon ○ Comma

○ Space ● := ○ |

Available Columns
- Type
- Name
- Comment
- Minimal Value
- Maximal Value

Selected Columns
- Variable
- Current Value

Item	Description
Storage type	Formats textual and binary are available to store the recipe files.
File path	Saving path for the recipe files. Example: the saving path for the file “AllRecipes” is PlcLogic/AllRecipes.
File extension	File extension of the file name: .<file extension> Naming of the file name: <recipe>.<recipe definition>.<file extension>.
Separator	Each value in the recipe file should be separated by a delimiter.
Available Columns Selected Columns	Definition of the content and order of a recipe file.
Save as Default	Apply this setting to all other recipe managers in the project.



☑ Recipe Management in the PLC

Save Recipe

☑ Save recipe changes to recipe files automatically

Load Recipe

● Load only by exact match of variable list

○ Load matching variables by variablename

Write Recipe

● Limit the variable to min/max when recipe value is out of the range

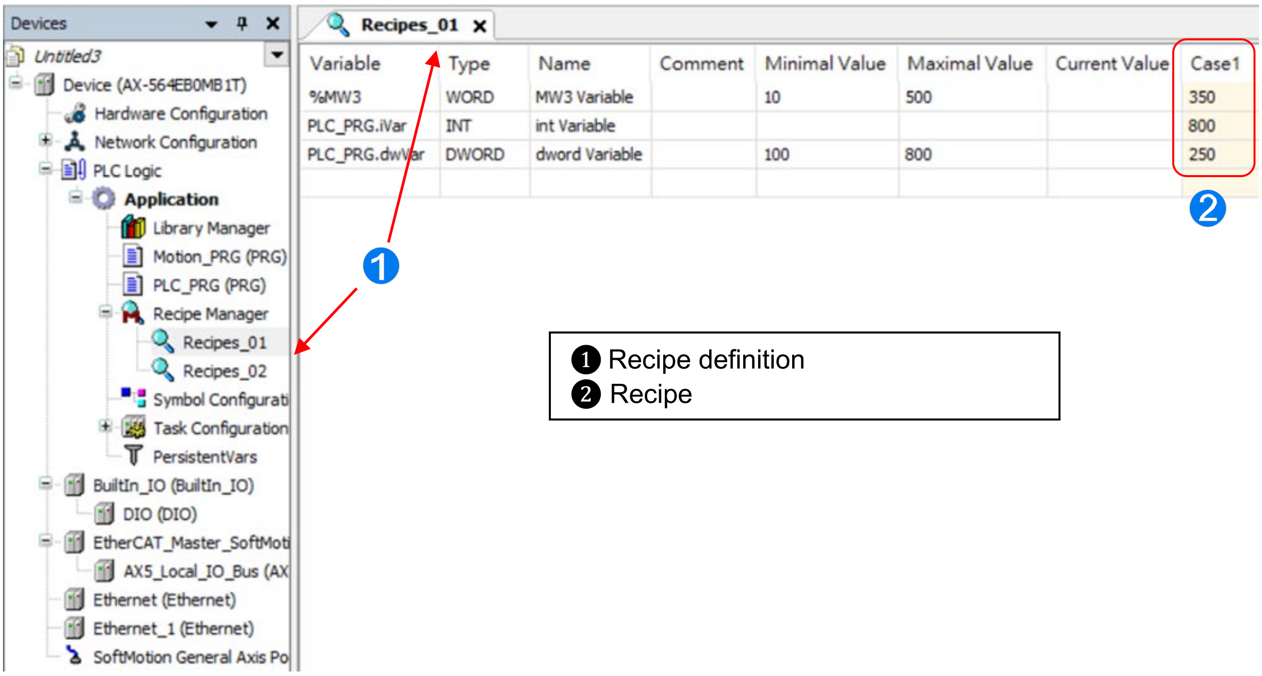
○ Do not write to a variable when the recipe value is out of the min/max range

Read Recipe

☐ Check recipe for changes

Item	Description
Recipe management in the PLC	Select this option to have this function activated.
Save Recipe	
Save recipe changes to recipe files automatically	Select this option to enable automatic updating of the recipe file during project downloads, ensuring that any modifications to the recipe are automatically saved to the file.
Load Recipe	
Load only by exact match of variable list	If this option is selected, the recipe file must contain the same variables and variable order, as well as the content in the Name field, in order to successfully import the recipe file into the controller variable when loading the recipe. (If there is extra variable information included at the end of the recipe file, and the variable order, content of the Name field, and the variables set in the controller recipe are identical, then the additional variable information at the end of the recipe file will be automatically disregarded and not imported.)
Load matching variables by variable name	If this option is selected, the controller imports variables with matched names from the recipe file, even if the order or the contents of the variable are different.
Write Recipe	
Limit the variable to min/max when recipe value is out of the range	If the recipe value goes beyond the specified maximum or minimum range, the corresponding variable will be assigned the maximum or minimum value.
Do not write to a variable when the recipe value is out of the min/max range	If the value of the recipe goes beyond the specified maximum or minimum range, it will not be imported to the corresponding variable.

9.1.2 Recipe Definition



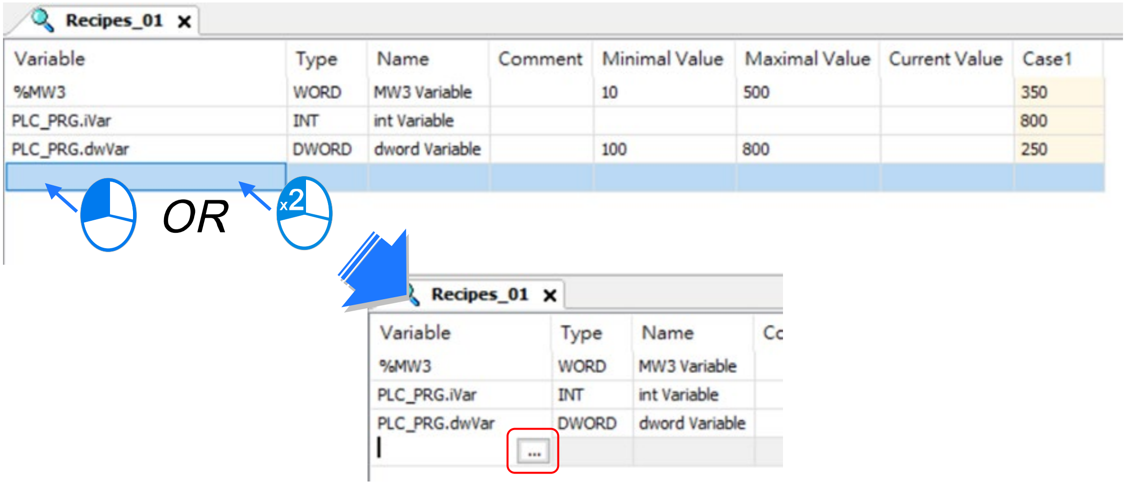
Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	Case1
%MW3	WORD	MW3 Variable		10	500		350
PLC_PRG.iVar	INT	int Variable					800
PLC_PRG.dwVar	DWORD	dword Variable		100	800		250

❶ Recipe definition
❷ Recipe

Item	Description
Variable	You can choose any variable from the table, including those that are defined within the POU.
Type	This field is filled automatically with the data type locked in the variable.
Name	You can define the variable names which can be used in verifying while loading the recipes.
Comment	Additional information can be added here.
Minimal Value Maximal Value	You can set the upper and lower limits for importing variables. If the to-be-imported recipe value goes beyond this range, the controller decides whether to import the recipe value into the variable according to the settings in the recipe manager.
Current Value	Current variable value during online mode

- Adding variables

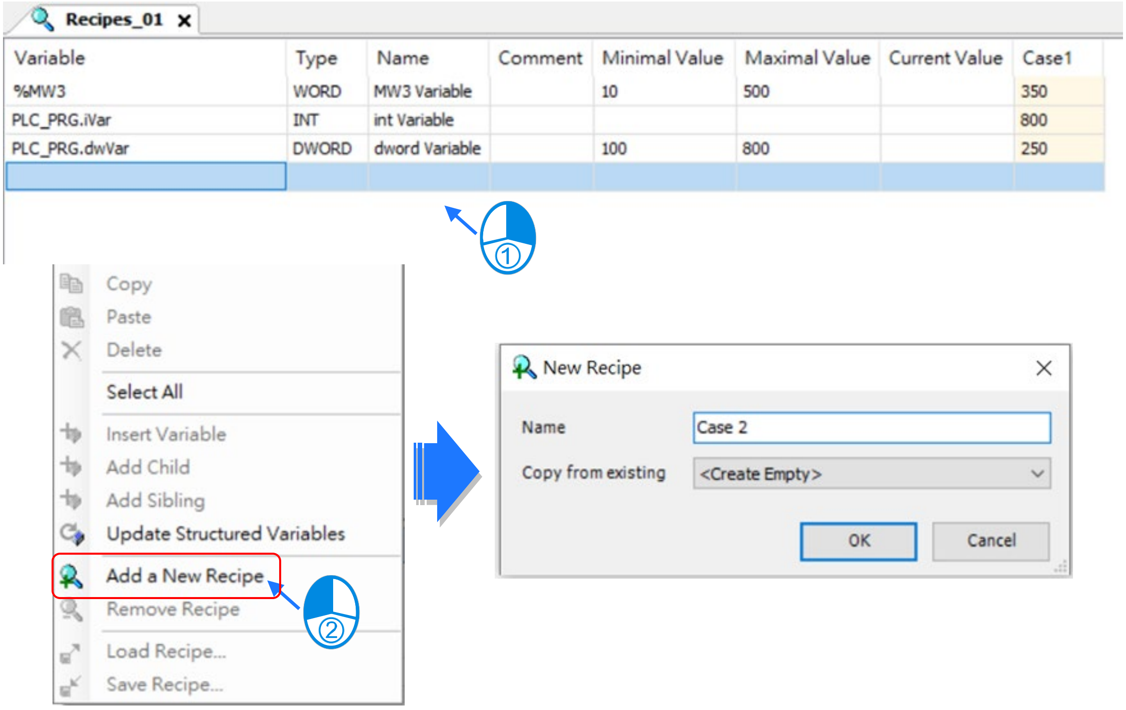
You can add variables by entering the name of the variable or double click the blank row to open the **Input Assistant** and select the variable.



Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	Case1
%MW3	WORD	MW3 Variable		10	500		350
PLC_PRG.iVar	INT	int Variable					800
PLC_PRG.dwVar	DWORD	dword Variable		100	800		250

- Adding recipes

You can add recipes by right clicking the blank area and select **Add a New Recipe**.



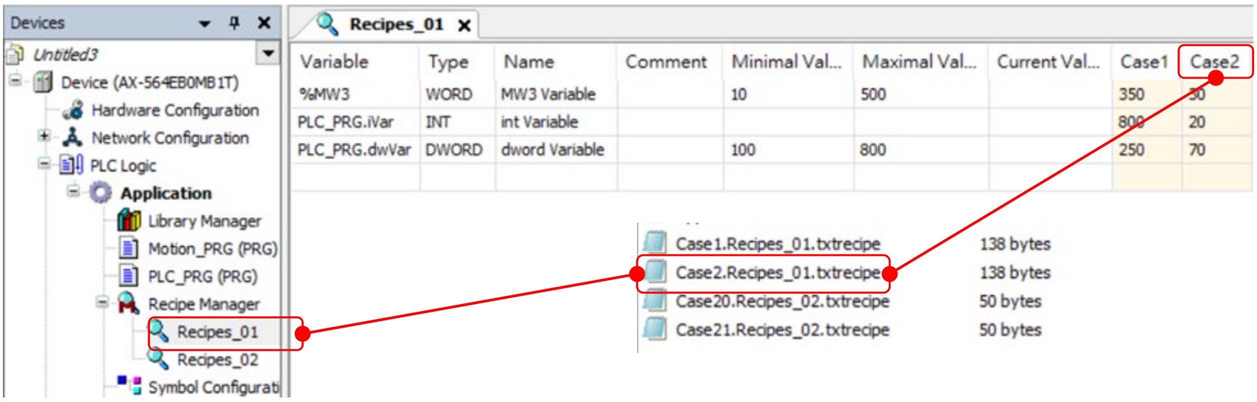
Variable	Type	Name	Comment	Minimal Value	Maximal Value	Current Value	Case1
%MW3	WORD	MW3 Variable		10	500		350
PLC_PRG.iVar	INT	int Variable					800
PLC_PRG.dwVar	DWORD	dword Variable		100	800		250

New Recipe

Name: Case 2

Copy from existing: <Create Empty>

- Generating recipe files from the controller.



Variable	Type	Name	Comment	Minimal Val...	Maximal Val...	Current Val...	Case1	Case2
%MW3	WORD	MW3 Variable		10	500		350	30
PLC_PRG.iVar	INT	int Variable					800	20
PLC_PRG.dwVar	DWORD	dword Variable		100	800		250	70

Case1.Recipes_01.txtrecipe	138 bytes
Case2.Recipes_01.txtrecipe	138 bytes
Case20.Recipes_02.txtrecipe	50 bytes
Case21.Recipes_02.txtrecipe	50 bytes

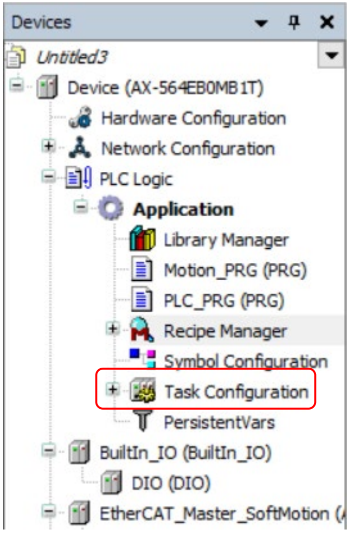
9.1.3 Recipe ManCommands

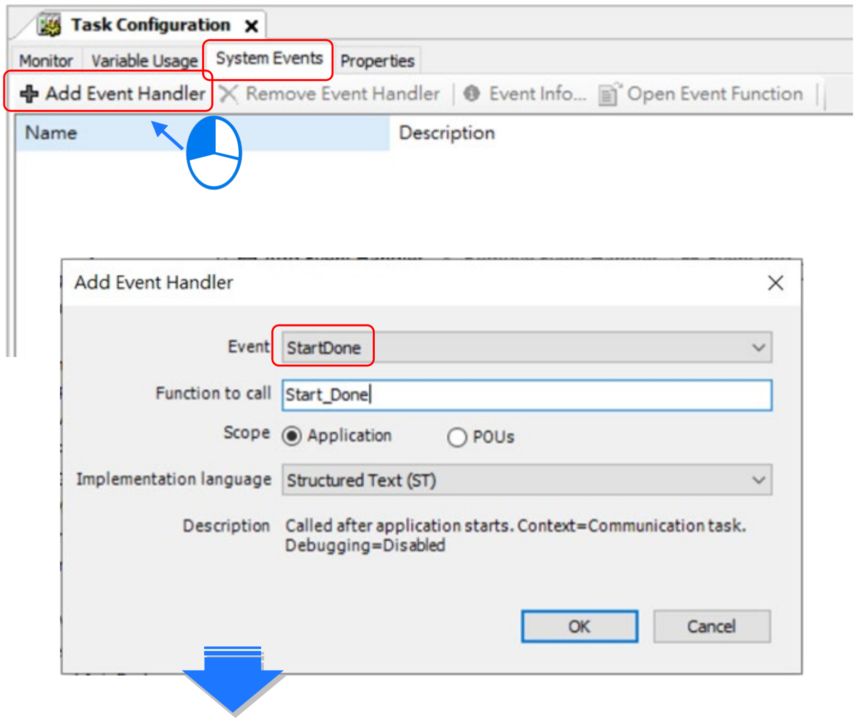
The "RecipeManCommands" of the "Recipe_Management.library" provides several methods for users to import recipe files into variables, or export controller variables to recipe files.

RecipeManCommands	Description
LoadAndWriteRecipe	Import the recipe from the default recipe file into the controller variable.
LoadFromAndWriteRecipe	Import the recipe from the specified recipe file into the controller variable.
ReadAndSaveAS	Save the controller variable to a specified file.
ReadAndSaveRecipe	Export the controller variables to the default configuration file.
ReadAndSaveRecipeAS	Export the controller variables to the default configuration file and save it to a specified file.

- **Example 1**

In this example, “StartDone” event is added by using “Add Event Handler” with "LoadAndWriteRecipe" to change the state of the controller from STOP to RUN and import the recipe “Case 1” from Recipes_01 to the corresponding variable of the controller automatically.





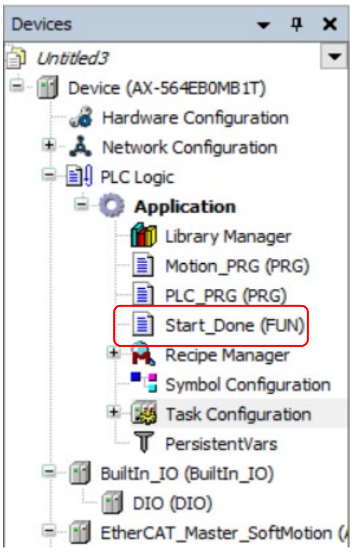
Event: StartDone

Function to call: Start_Done

Scope: ● Application ○ POUs

Implementation language: Structured Text (ST)

Description: Called after application starts. Context=Communication task. Debugging=Disabled



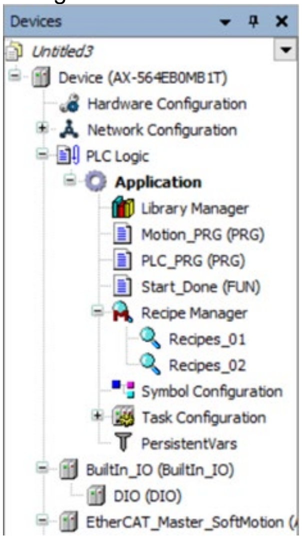

POU: Start_Done

```
FUNCTION Start_Done : DWORD
VAR_IN_OUT
    EventPrm : CmpApp . EVTPARAM_CmpApp ;
END_VAR
VAR
    FB0 : RecipeManCommands ;
END_VAR

FB0 . LoadAndWriteRecipe ( RecipeDefinitionName := 'Recipes_01' , RecipeName :=
'Case1' ) ;
```

- **Example 2**

In this example, “ReadAndSaveRecipe” and “ReadAndSaveAs” are used to export the controller variables to the default configuration file and to the specified file.

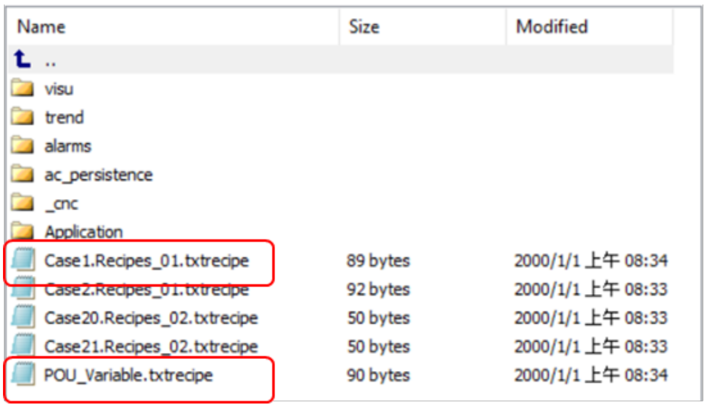


```
PROGRAM  PLC_PRG
VAR
     iVar  : INT ;
     dwVar , dw_Retur : DWORD ;
     udi_Return  : UDINT ;
     bVar0 , bVar1 : BOOL ;
     FB1 :  RecipeManCommands ;
END_VAR
```

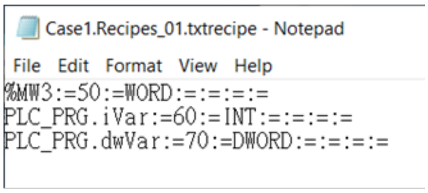
```
IF  bVar0  THEN
     %MW3 := 50 ;
     iVar := 60 ;
     dwVar := 70 ;
     udi_Return  := FB1 . ReadAndSaveRecipe ( RecipeDefinitionName :=
'Recipes_01' , RecipeName := 'Case1' ) ;
     bVar0 := FALSE ;
END_IF

IF  bVar1  THEN
     %MW3 := %MW3 + 10 ;
     iVar := iVar + 20 ;
     dwVar := dwVar + 30 ;
     udi_Return  := FB1 . ReadAndSaveAS ( RecipeDefinitionName := 'Recipes_01' ,
FileName := 'POU_Variable.txtrecipe' ) ;
     bVar1 := FALSE ;
END_IF
```

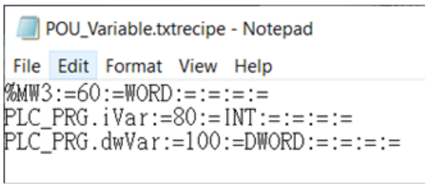
- Set the variables “bVar0” and “bVar1” to ON and generate files and contents from the controller.



Name	Size	Modified
..		
visu		
trend		
alarms		
ac_persistence		
_cnc		
Application		
Case1.Recipes_01.txtrecipe	89 bytes	2000/1/1 上午 08:34
Case2.Recipes_01.txtrecipe	92 bytes	2000/1/1 上午 08:33
Case20.Recipes_02.txtrecipe	50 bytes	2000/1/1 上午 08:33
Case21.Recipes_02.txtrecipe	50 bytes	2000/1/1 上午 08:33
POU_Variable.txtrecipe	90 bytes	2000/1/1 上午 08:34



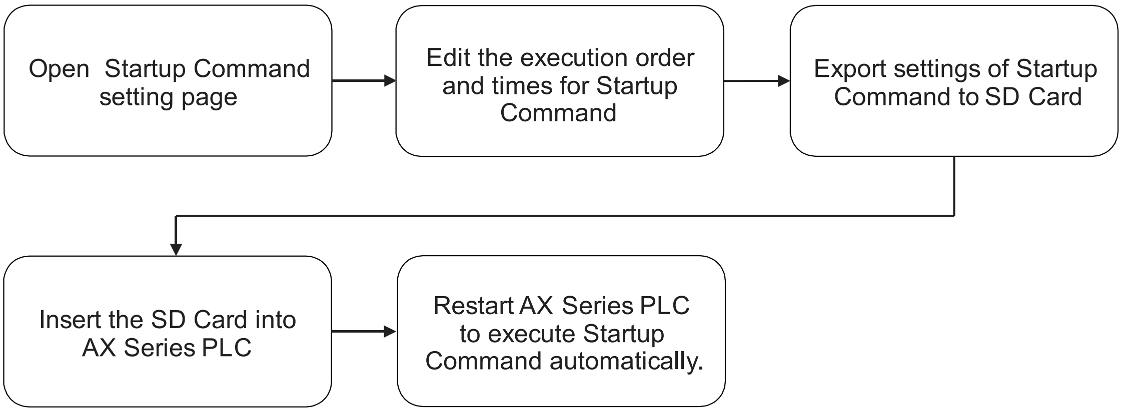
```
%MW3:=50:=WORD:=:=:=:=
PLC_PRG.iVar:=60:=INT:=:=:=:=
PLC_PRG.dwVar:=70:=DWORD:=:=:=:=
```



```
%MW3:=60:=WORD:=:=:=:=
PLC_PRG.iVar:=80:=INT:=:=:=:=
PLC_PRG.dwVar:=100:=DWORD:=:=:=:=
```

9.2 Startup Command

The Startup Command provides several different commands for users to execute functions, including project backup, project restore, and firmware update. With these commands, a quick mass replication can apply on controllers, saving time on individual controller setup.



- **Startup Commands**
  - Restore Application from Selected Device
  - Restore Application from Portable Device
  - Restore Source from Current Project
  - Restore Source from Portable Device
  - Restore to Factory Settings
  - Backup Application
  - Backup Source
  - Firmware Update

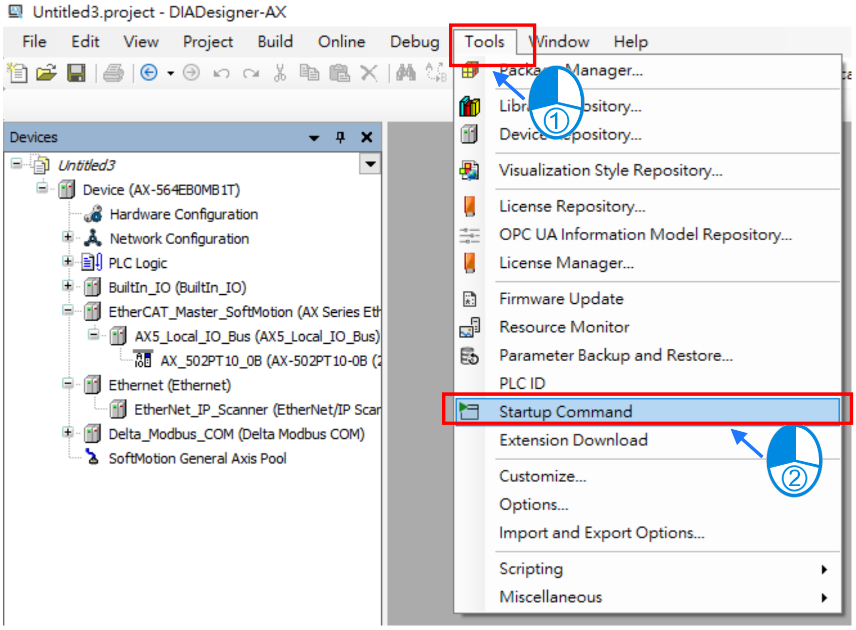
- **The followings support Startup Commands**
  - AX-5 Series PLC with firmware version V1.0.0.0 or later
  - DIADesigner-AX version V1.4 or later

9.2.1 Operation of Startup Command

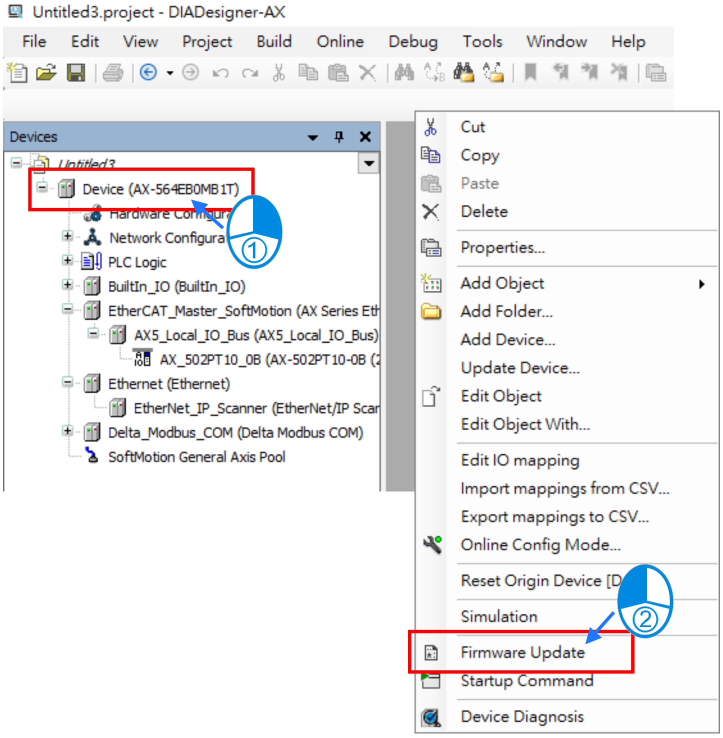
- **Open Startup Command Setting Page**

You can open the setting page of Startup Command with a specific controller or without. If you open Startup Command Setting Page without selecting a specific controller, you can edit the startup commands, but the applications or project archive cannot be generated.

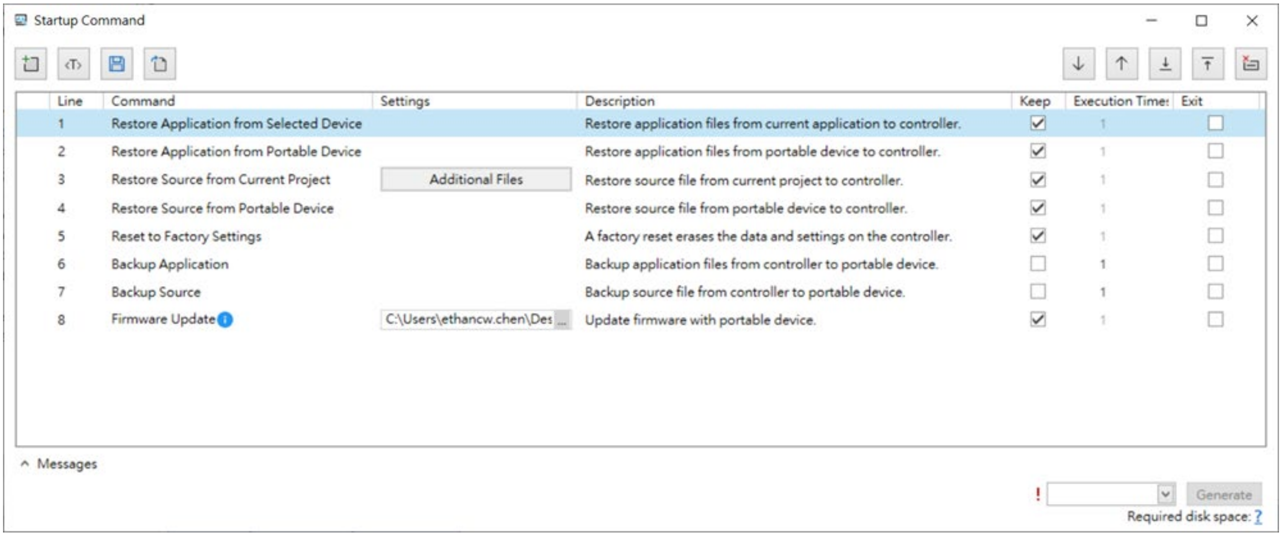
- Open Startup Command setting page without selecting a specific controller.



- Open Startup Command setting page with a specific controller.



- **Startup Command Setting Page**



Line	Command	Settings	Description	Keep	Execution Times	Exit
1	Restore Application from Selected Device		Restore application files from current application to controller.	☑	1	☐
2	Restore Application from Portable Device		Restore application files from portable device to controller.	☑	1	☐
3	Restore Source from Current Project	Additional Files	Restore source file from current project to controller.	☑	1	☐
4	Restore Source from Portable Device		Restore source file from portable device to controller.	☑	1	☐
5	Reset to Factory Settings		A factory reset erases the data and settings on the controller.	☑	1	☐
6	Backup Application		Backup application files from controller to portable device.	☐	1	☐
7	Backup Source		Backup source file from controller to portable device.	☐	1	☐
8	Firmware Update	C:\Users\ethancw.chen\Des ...	Update firmware with portable device.	☑	1	☐

Function Button	Description
+☐	Add or insert Startup Command
‹T›	Startup Command template
💾	Save Startup Comman setting file
⮌☐	Import Startup Command setting file
↓	Move the selected command downward
↑	Move the selected command upward
⤓	Move the selected command to the last row
⤒	Move the selected command to the first row
✗	Delete the selected command
Keep	Ticked: always keep the command Unticked: unselected, after the set execution time has met, the unselected command will be deleted.
Execution Times	The number of times that a command to be executed.
Exit	After this selected command is executed, exit from the startup command. And the rest commands are not executed.

- **Startup Commands**

Startup Command	Description
Restore Application from Selected Device (*1)	Use the restored applications (Project Archive) from the selected device and save them as the applications (Project Archive) of the current controller in a designed folder of a SD card.
Restore Application from Portable Device	Restore the applications from a portable device.
Restore Source from Current Project (*1) (*2)	Use the restored applications (Project Archive) from the current project and save them as the applications (Project Archive) of the current controller in a designed folder of a SD card.
Restore Source from Portable Device	The project archive of configuration files is stored in a portable device for restoring.
Restore to Factory Settings	Restore to factory settings
Backup Application	Backup the applications
Backup Source	Project archive for backup
Firmware Update (*3) (*4)	Firmware update

Note:

*1: Open the Startup Command settings page from the Device tree view to use this command.

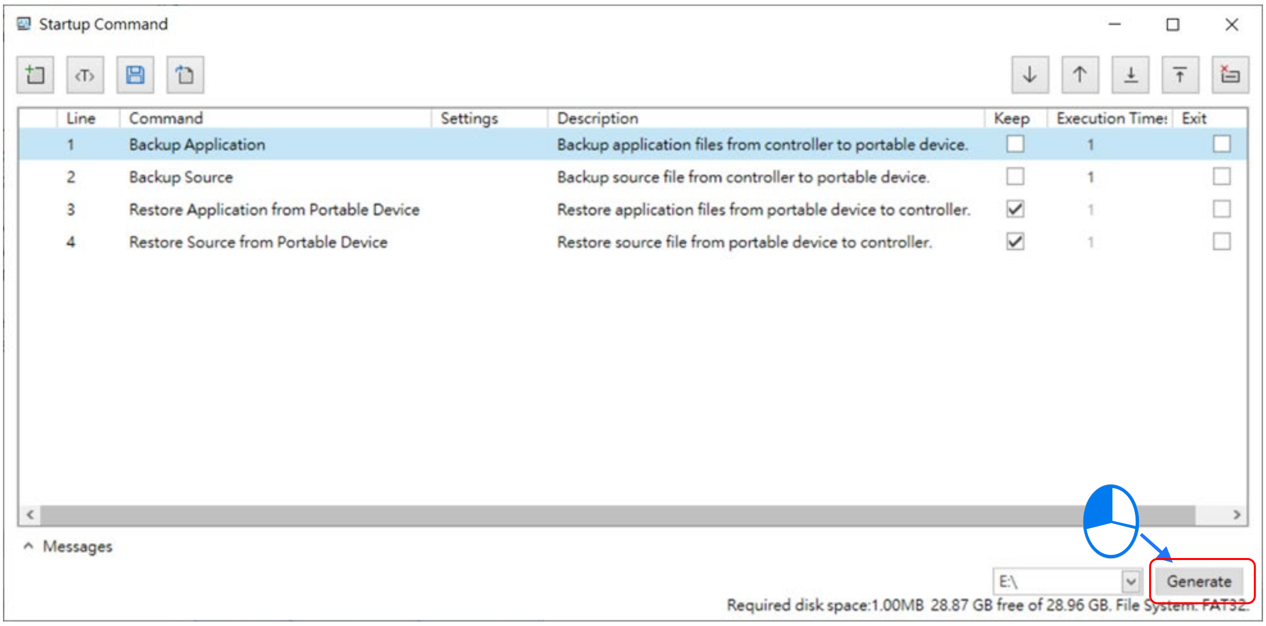
*2: This command allows you to select which types of files need to be imported from the Project Archive.

*3: To use this command, you need to have the firmware file ready and be able to provide the file path.

*4: After the firmware update is complete, you need to restart the AX Series PLC CPU and after that the controller runs on the new version of the firmware.

- **Export Startup Command**

After adding and arranging the startup commands on DIADesigner-AX, click **Generate** to export the commands to the SD card. After that you can insert this SD card to AX Series PLC and after restart the power, the generated Startup Commands will be executed automatically.



Line	Command	Settings	Description	Keep	Execution Time:	Exit
1	Backup Application		Backup application files from controller to portable device.	☐	1	☐
2	Backup Source		Backup source file from controller to portable device.	☐	1	☐
3	Restore Application from Portable Device		Restore application files from portable device to controller.	☑	1	☐
4	Restore Source from Portable Device		Restore source file from portable device to controller.	☑	1	☐

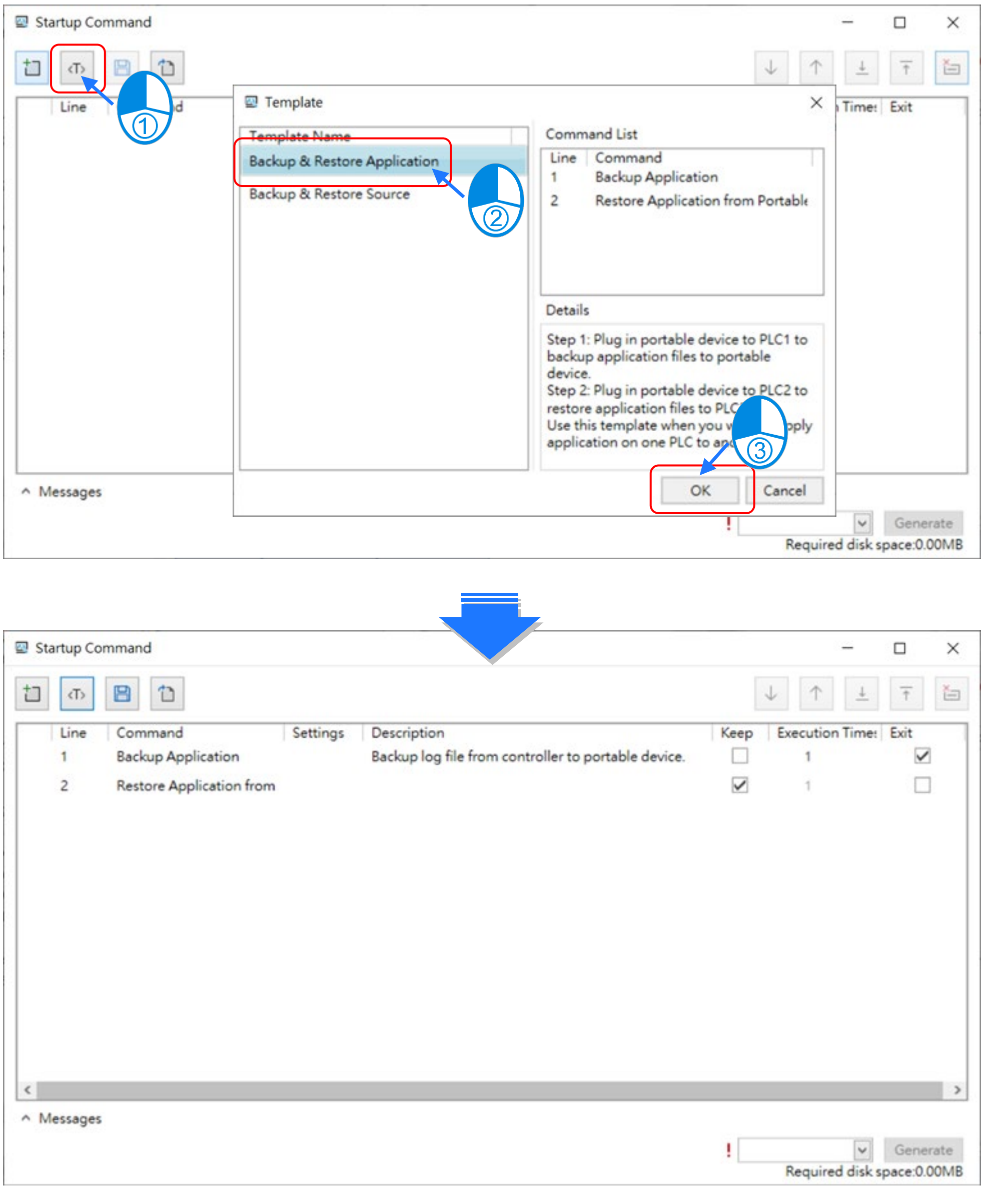
Required disk space:1.00MB 28.87 GB free of 28.96 GB. File System: FAT32.

9.2.2 Example

- **Example 1: Copying the startup commands of one AX-5 Series PLC CPU (A) to another AX-5 Series PLC CPU (B).**

■ **Operation steps:**

1. Click to open **Startup Command Template**. Select “Backup & Restore Application” to add it in your startup commands. Click **OK** to export the command to the SD card inserted on your PC.
2. Insert the SD card to the AX-5 Series PLC (A) and then restart the power for backup. Wait for the RUN LED stops blinking to complete the task. And then take the SD card out.
3. Insert this SD card to the other AX-5 Series PLC (B). Restart the power on this AX-5 Series PLC (B) for the backup startup commands to be copied to this AX-5 Series PLC (B). Wait for the RUN LED stops blinking to complete the task. And then AX-5 Series PLC (B) is loaded with applications from AX-5 Series PLC (A).



Line	Command
1	Backup Application
2	Restore Application from Portable

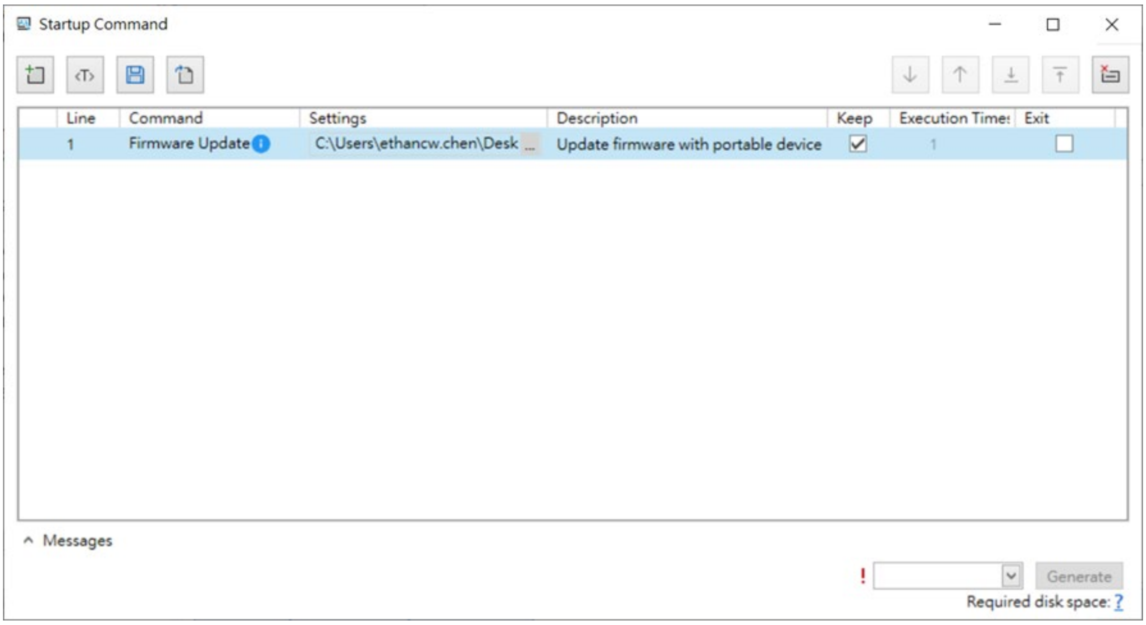
Step 1: Plug in portable device to PLC1 to backup application files to portable device.
Step 2: Plug in portable device to PLC2 to restore application files to PLC
Use this template when you w pply application on one PLC to an

Line	Command	Settings	Description	Keep	Execution Time:	Exit
1	Backup Application		Backup log file from controller to portable device.	☐	1	☑
2	Restore Application from			☑	1	☐

- **Example 2: Firmware update on AX-5 Series PLC.**

■ **Operation steps:**

1. Click and add “Firmware Update” for your startup commands. Click **Generate** to export the command to the SD card inserted on your PC.
2. Insert the SD card to the AX-3 Series PLC and then restart the power to update firmware. Wait for the RUN LED stops blinking to complete the task.
3. Take out this SD card and restart the power. After that the controller runs on the new version of the firmware.

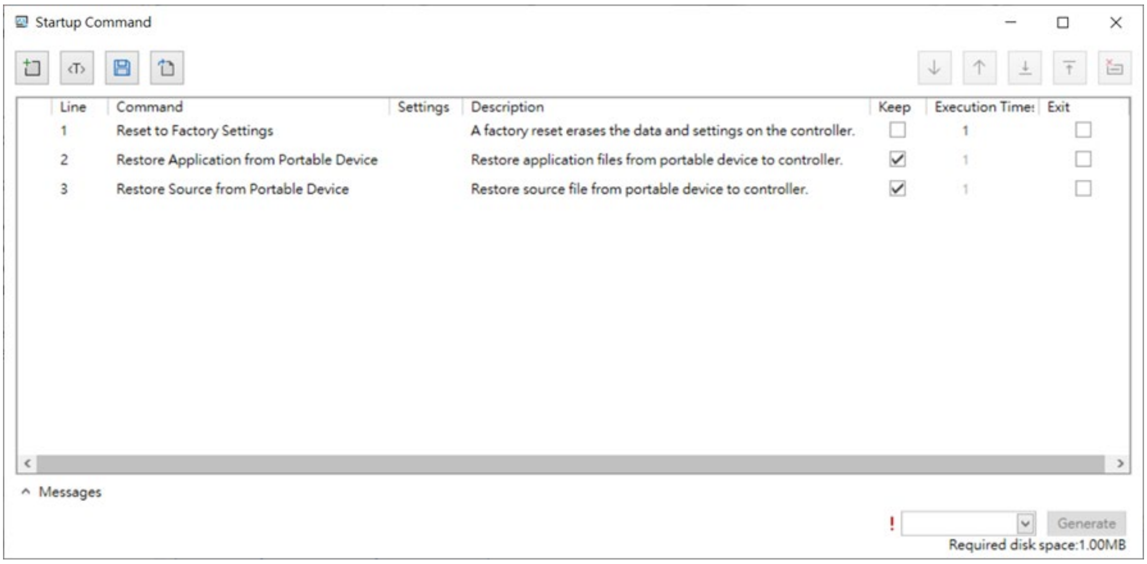


Line	Command	Settings	Description	Keep	Execution Time:	Exit
1	Firmware Update	C:\Users\ethancw.chen\Desk ...	Update firmware with portable device	☑	1	☐

- **Example 3: If the login account and password of the AX-5 Series PLC are lost, you need to restore AX-5 Series PLC to its factory settings and restore the applications as well as the project archive.**

■ **Operation steps:**

1. Click and add “Reset to Factory Settings”, “Restore Application from Portable Device” and “Restore Source from Portable Device” for your startup commands. Click **Generate** to export the commands to the SD card inserted on your PC.
2. Insert the SD card to the AX-5 Series PLC and then turn restart the power to execute the startup commands. Wait for the RUN LED stops blinking to complete the task.
3. Take out this SD card. The AX-5 Series PLC is now reset to factory settings and with applications and project archive from the SD card.



Line	Command	Settings	Description	Keep	Execution Time:	Exit
1	Reset to Factory Settings		A factory reset erases the data and settings on the controller.	☐	1	☐
2	Restore Application from Portable Device		Restore application files from portable device to controller.	☑	1	☐
3	Restore Source from Portable Device		Restore source file from portable device to controller.	☑	1	☐

9.3 Protection Mechanisms

The AX series controller provides different types of protection mechanisms to prevent users' projects from being opened or directly copied to unprotected controllers. The protection mechanisms can be divided into three types, project encryption, account permissions, as well as Project ID and PLC ID.

Protection Mechanism	Description
Project encryption	Project encryptions can be set by users to ensure that the project remains protected from being accessed by unauthorized individuals.
Account permission	Project accounts within the project can have different permission groups set up by users, which restrict the operations that each project account group can perform.
Project ID and PLC ID	When this feature is enabled, it will verify if the Project ID configured in DIA Designer-AX matches the PLC ID of the controller during downloading. If there is no match, the project cannot be downloaded. This method guarantees that only authorized controllers are utilized.

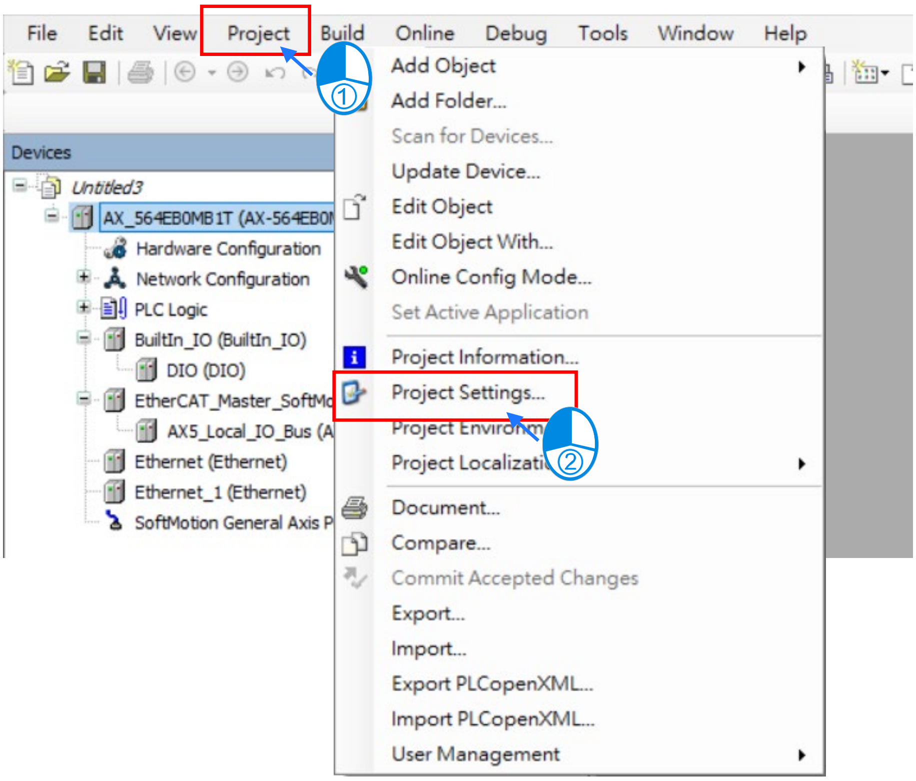
9.3.1 Project Encryption

You can encrypt a project. After a project is encrypted, you will need a password to open it.

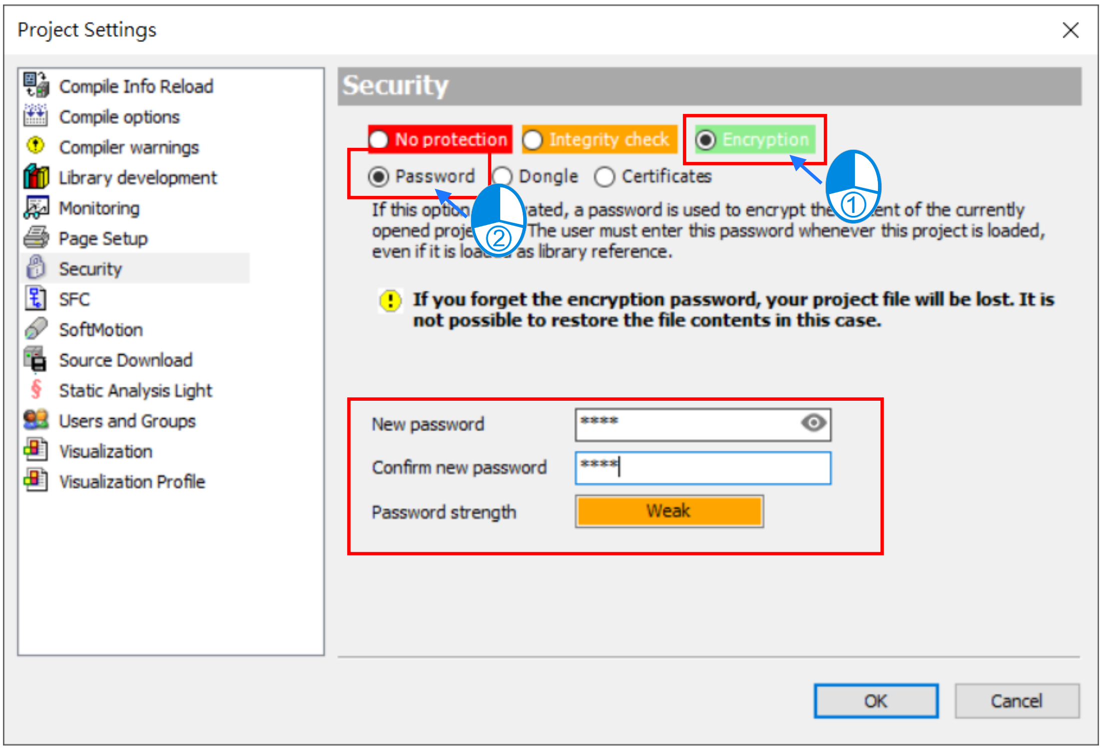
Caution: If the project password is lost, the project will be irretrievable.

9.3.1.1 Set up a Project Password

- **Open the Project Settings page.**



- **Set up a password for the project.**



If this option ... ated, a password is used to encrypt the ... ent of the currently opened proje... The user must enter this password whenever this project is loaded, even if it is loa... as library reference.

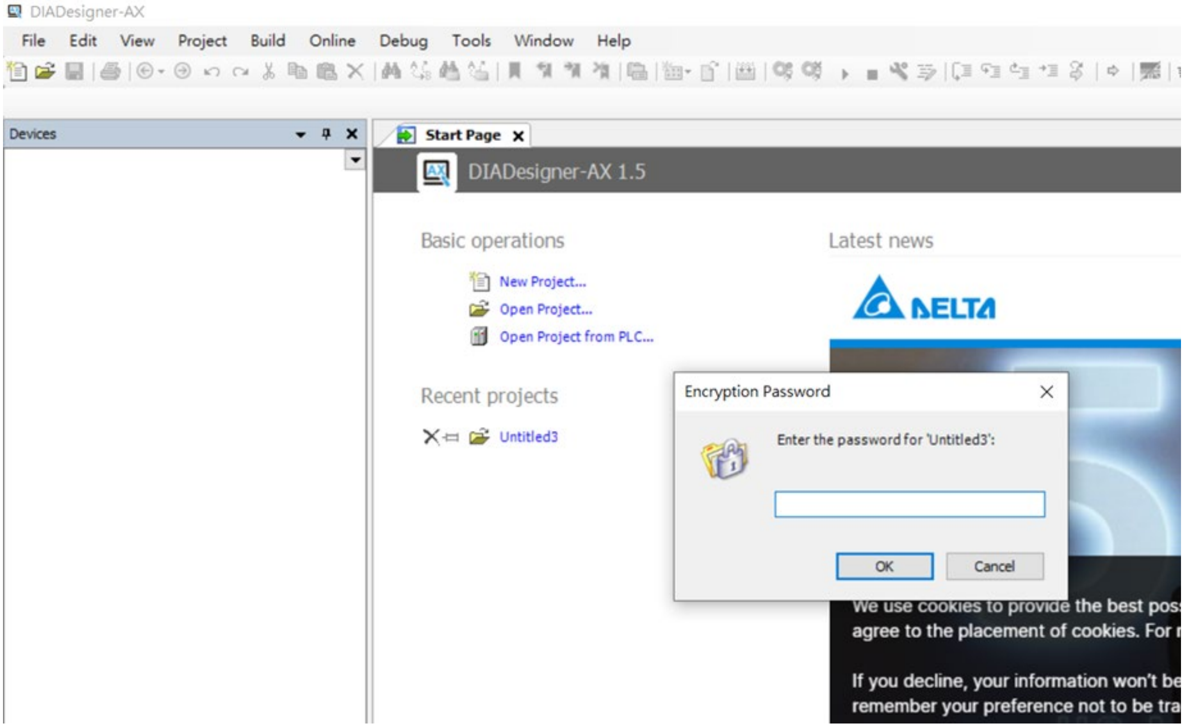
If you forget the encryption password, your project file will be lost. It is not possible to restore the file contents in this case.

New password: ****

Confirm new password: ****

Password strength: Weak

- **After setting up a password for the project, a window will prompt you to enter the project password to open the project.**



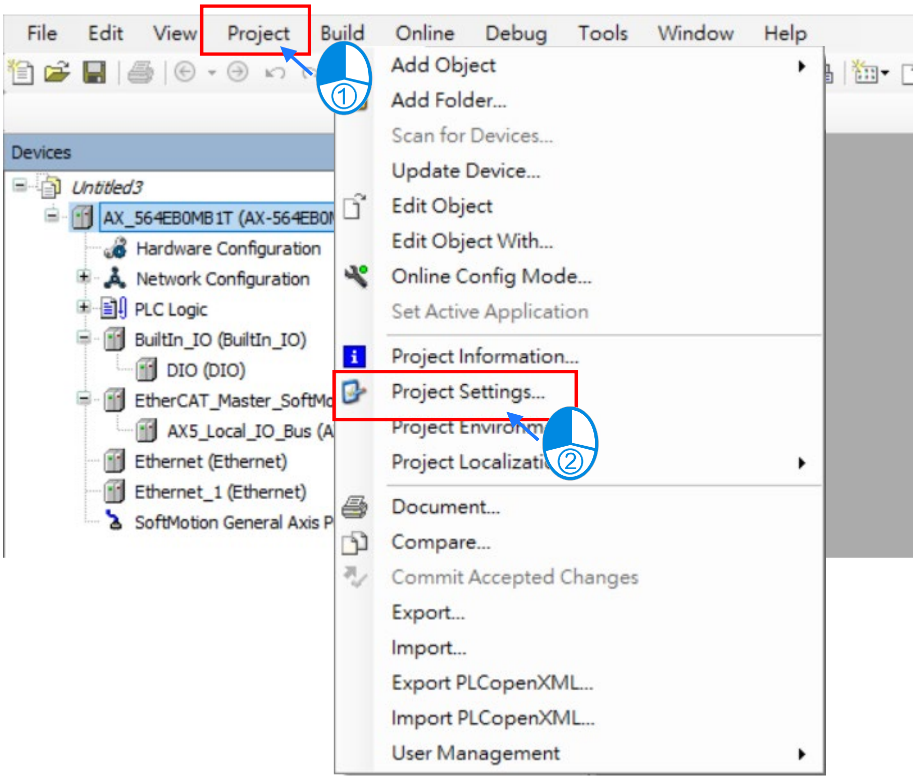
Enter the password for 'Untitled3':

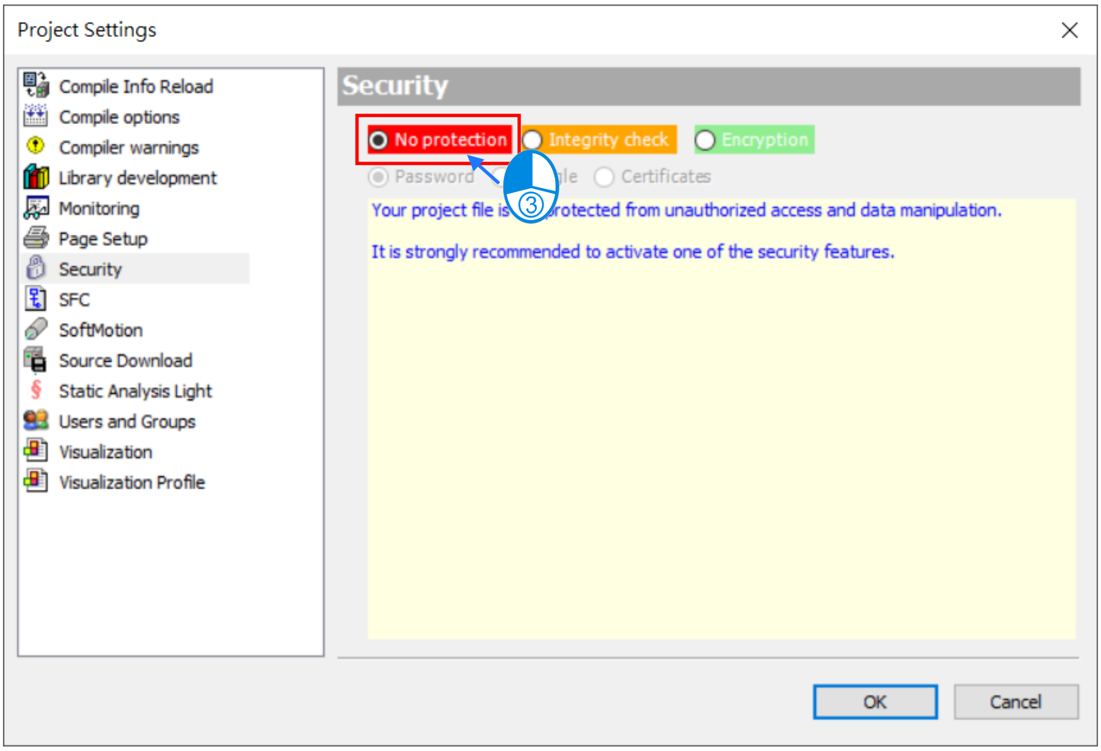
We use cookies to provide the best pos
agree to the placement of cookies. For r

If you decline, your information won't be
remember your preference not to be tra

9.3.1.2 Remote the Project Password

- **Open the Project Settings page.**





Your project file is ③ protected from unauthorized access and data manipulation.

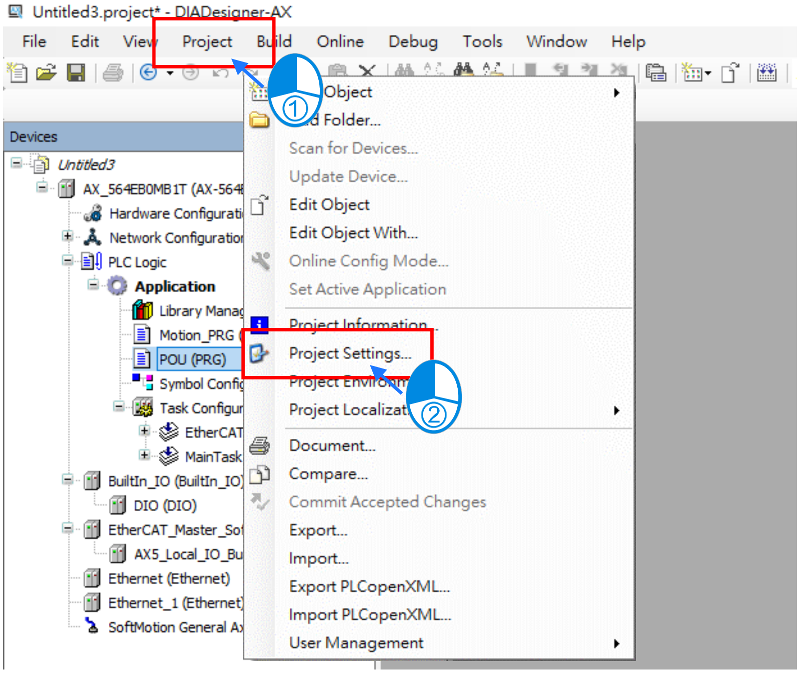
It is strongly recommended to activate one of the security features.

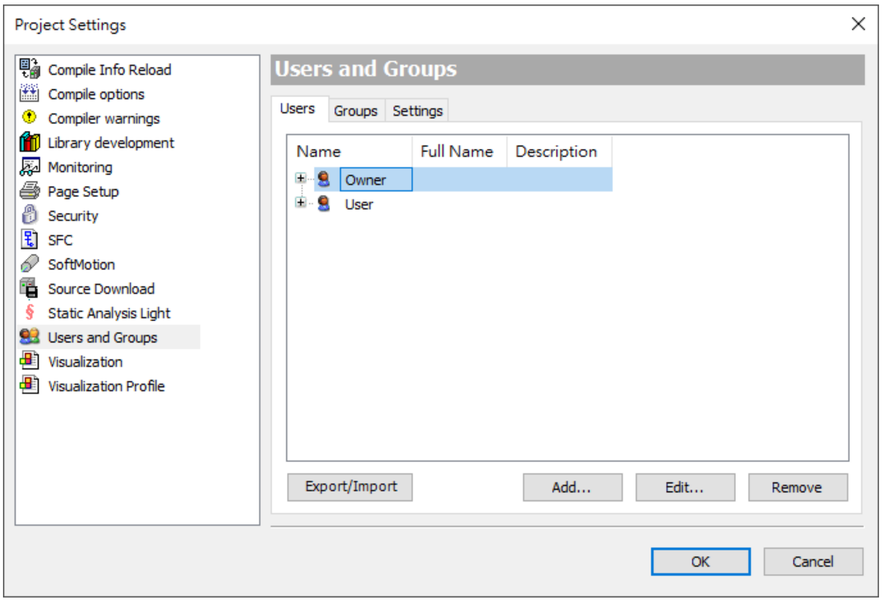
9.3.2 Account Permission

Project account permissions are used to manage editing, modifying, and other operations related to project contents. A default account will be created for all projects. Ensure that the default account password is either changed or deleted to prevent any misuse of the default project account. This account permission is different from Right Settings stated in section 4.2.19. The account permission here is set for the project and the right settings is created for the user.
Default project account and password are: Account Name: Owner; Password: blank (no need to input anything).

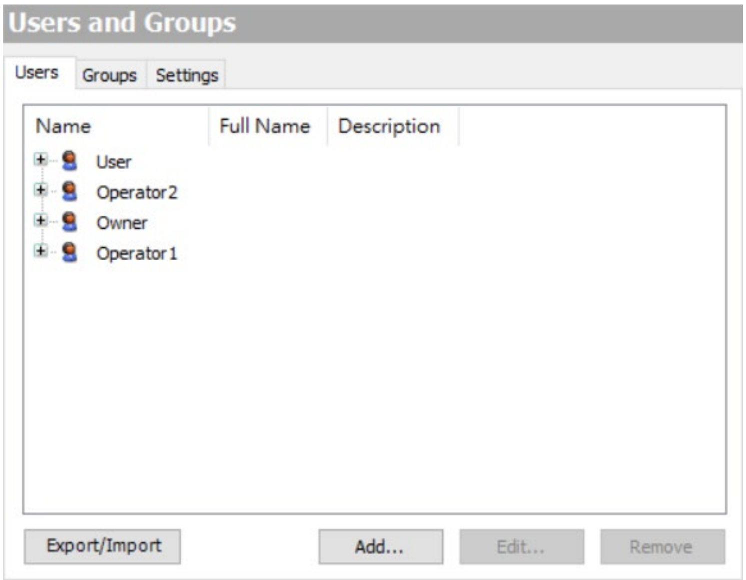
9.3.2.1 Set up a Project Account

- **Open the Project Settings page.**



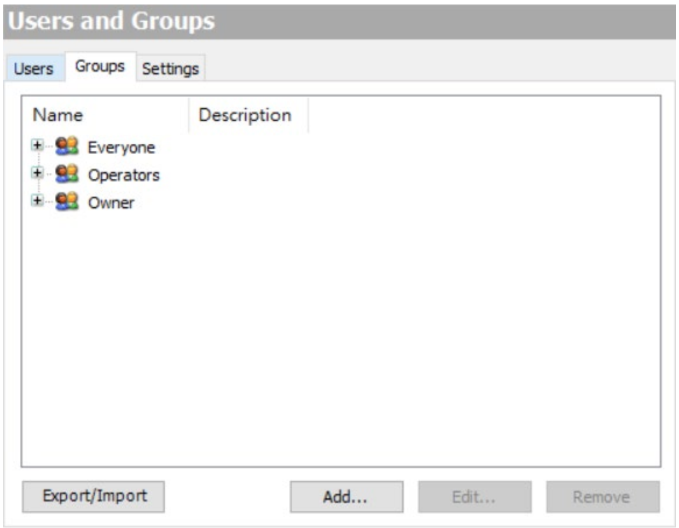


- **Set up a project account.**
  Users tab: All currently defined users, and below them their memberships of user groups, are listed in a tree structure.



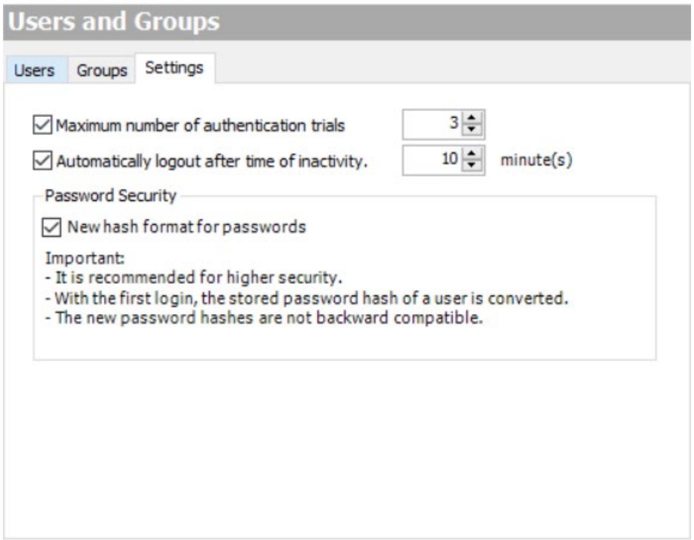
Item	Description
Export/Import	Export / import project accounts and group accounts.
Add User	Add a project account.
Edit User	Edit a project account.
Remove	Delete a project account.

Groups tab: All currently defined groups, and below them the users assigned to them, are listed in a tree structure.



Item	Description
Export/Import	Export / import project accounts and group accounts.
Add Group	Add a project group account.
Edit Group	Edit a project group account.
Remove	Delete a project group account.

Settings tab: Configure the related behaviors for project account login.



Users and Groups

☑ Maximum number of authentication trials 3

☑ Automatically logout after time of inactivity. 10 minute(s)

Password Security

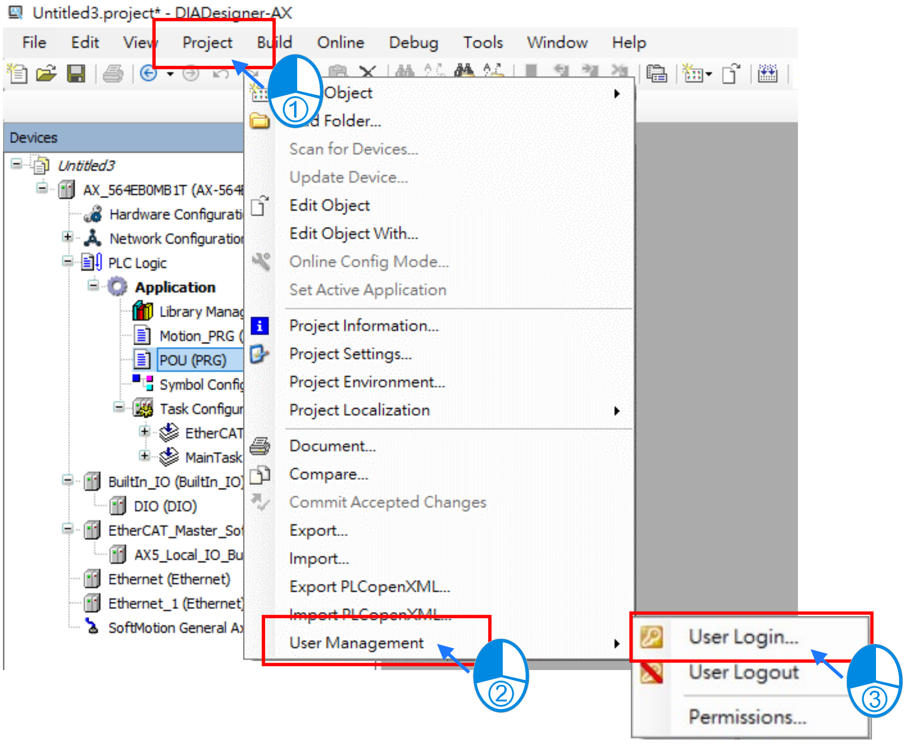
☑ New hash format for passwords

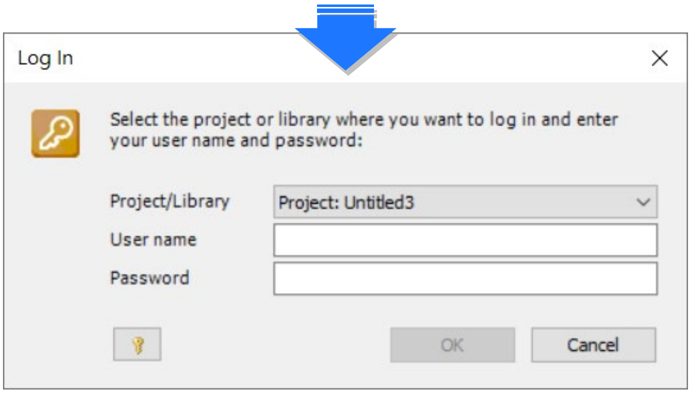
Important:
- It is recommended for higher security.
- With the first login, the stored password hash of a user is converted.
- The new password hashes are not backward compatible.

Item	Description
Maximum number of authentication trials	Maximum number of login attempt. Note: If you have tried to log in with an incorrect password the number of times specified here, then the user account will be disabled. And that will lead to no permission to open the project.
Automatically logout after time of inactivity	Automatic logout time for project accounts You will be automatically logged out if no action is detected during the time span (in minutes) as specified here.

9.3.2.2 Project - User Log in

- User Login Method 1





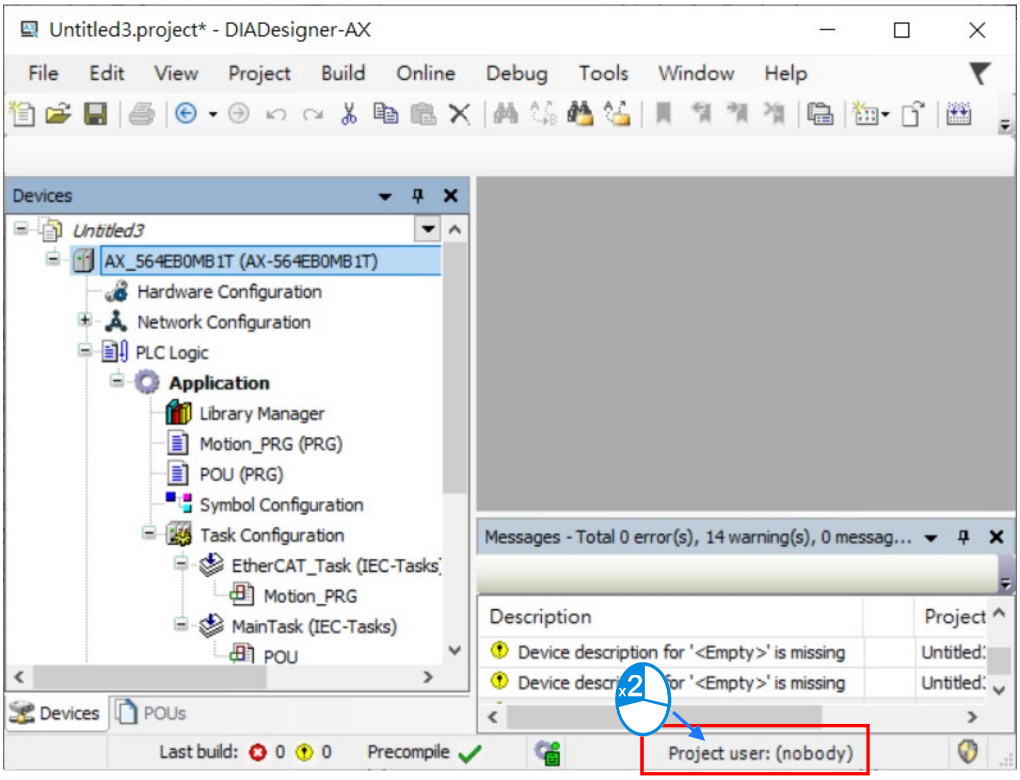
Select the project or library where you want to log in and enter your user name and password:

Project/Library: Project: Untitled3

User name:

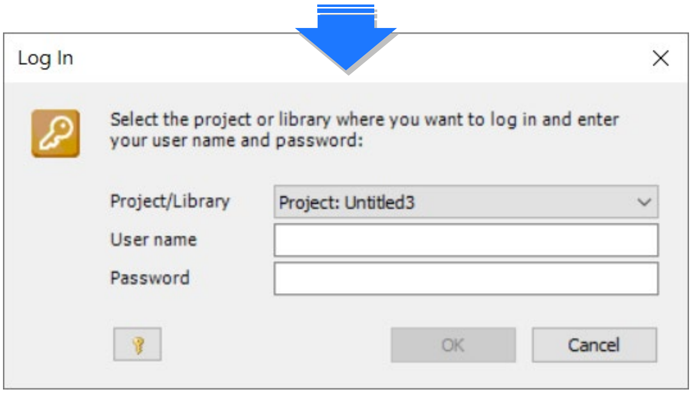
Password:

- User Login Method 2



Messages - Total 0 error(s), 14 warning(s), 0 messag...

Description	Project
Device description for '<Empty>' is missing	Untitled
Device description for '<Empty>' is missing	Untitled



Select the project or library where you want to log in and enter your user name and password:

Project/Library: Project: Untitled3

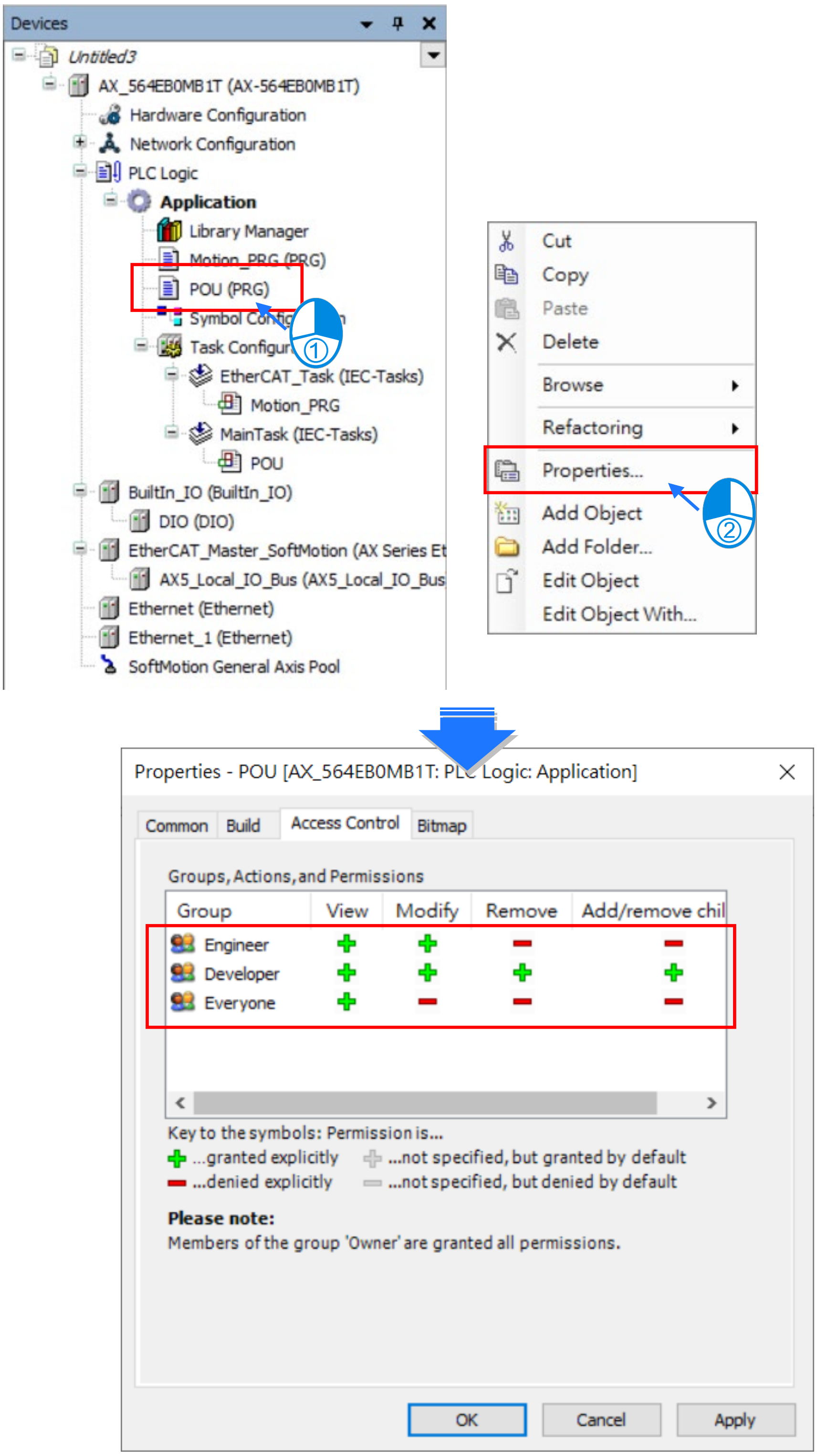
User name:

Password:

9.3.2.3 Properties

All objects within the project tree structure can be assigned corresponding user permissions. The following example uses POU to illustrate this concept.

- **Open the Properties Settings page.**



Groups, Actions, and Permissions

Group	View	Modify	Remove	Add/remove chil
Engineer	✚	✚	▬	▬
Developer	✚	✚	✚	✚
Everyone	✚	▬	▬	▬

Key to the symbols: Permission is...

✚ ...granted explicitly ✚ ...not specified, but granted by default

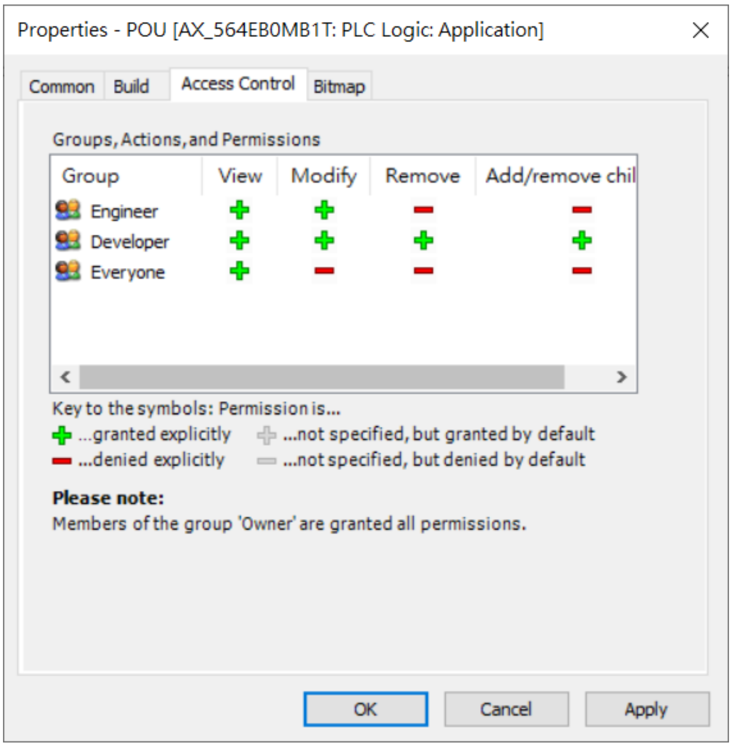
▬ ...denied explicitly ▬ ...not specified, but denied by default

Please note:
Members of the group 'Owner' are granted all permissions.

Note: You will need to set up the project account before setting up the properties.

- **Access Control**

Access Control defines which user groups are permitted to execute which actions on the object. For Everyone group: they are users without login; they only have permission to view but no permission to change or modify anything. When this user group try to edit or delete the contents in a POU, DIADesigner-AX will prompt a message to notify users which project account groups have permissions to carry out the relevant operations.



Groups, Actions, and Permissions

Group	View	Modify	Remove	Add/remove chil
Engineer	+	+	−	−
Developer	+	+	+	+
Everyone	+	−	−	−

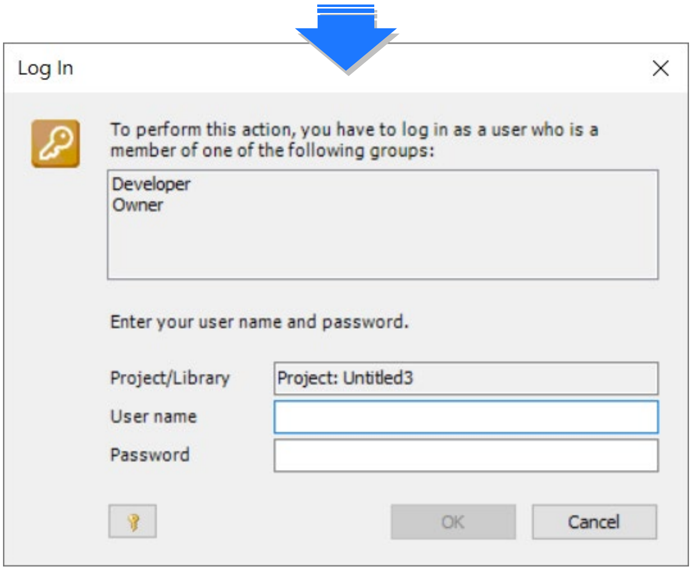
Key to the symbols: Permission is...

+ ...granted explicitly   + ...not specified, but granted by default

− ...denied explicitly   − ...not specified, but denied by default

Please note:

Members of the group 'Owner' are granted all permissions.



To perform this action, you have to log in as a user who is a member of one of the following groups:

Developer
Owner

Enter your user name and password.

Project/Library: Project: Untitled3

User name:

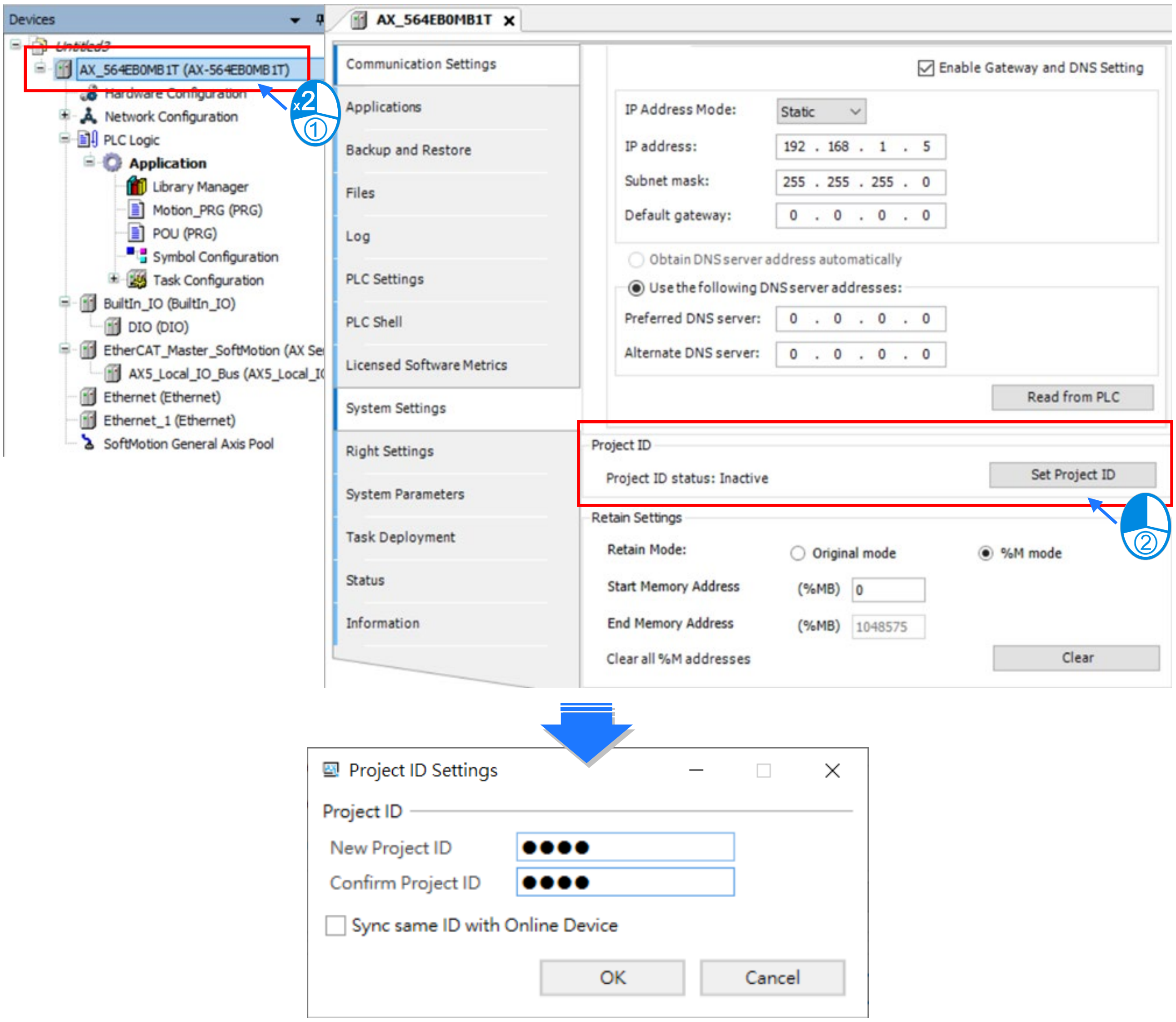
Password:

9.3.3 Project ID and PLC ID

When the PLC ID is activated on the AX series controller, the software "DIA Designer-AX" will verify the Project ID of the project with the PLC ID of the controller before allowing the project download process to proceed. If the IDs do not match, the download process will not be permitted.

9.3.3.1 Set up a Project ID

You can set the Project ID in the "System Settings" page of the AX series controller. If the "Sync same ID with Online Device" option is checked, the Project ID will be synchronized to the selected AX series controller's PLC ID.



☑ Enable Gateway and DNS Setting

IP Address Mode: Static

IP address: 192 . 168 . 1 . 5

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 0 . 0 . 0 . 0

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 0 . 0 . 0 . 0

Alternate DNS server: 0 . 0 . 0 . 0

Project ID

Project ID status: Inactive

Retain Settings

Retain Mode: ○ Original mode ● %M mode

Start Memory Address (%MB) 0

End Memory Address (%MB) 1048575

Clear all %M addresses

Project ID

New Project ID ●●●●

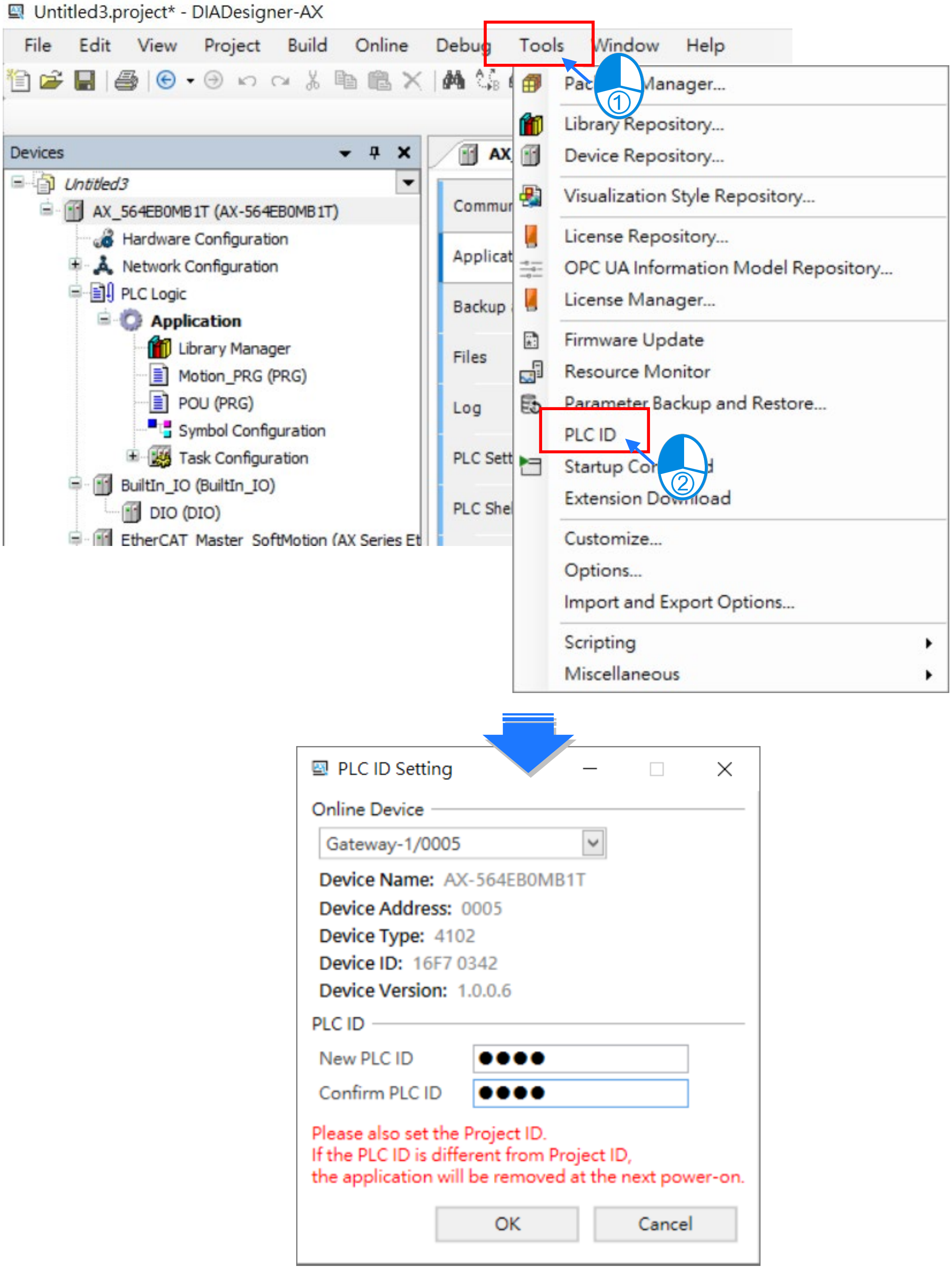
Confirm Project ID ●●●●

☐ Sync same ID with Online Device

9.3.3.2 Set up a PLC ID

You can set up PLC ID in the PLC ID Setting page. Find Tools tab on the menu bar and click PLC ID to open its setting page.

Note: If the "Project ID" configured in the current controller does not match the "PLC ID" of the controller, the AX series controller will delete the project (Application) automatically upon the next power-on.



Online Device

Gateway-1/0005

Device Name: AX-564EB0MB1T

Device Address: 0005

Device Type: 4102

Device ID: 16F7 0342

Device Version: 1.0.0.6

PLC ID

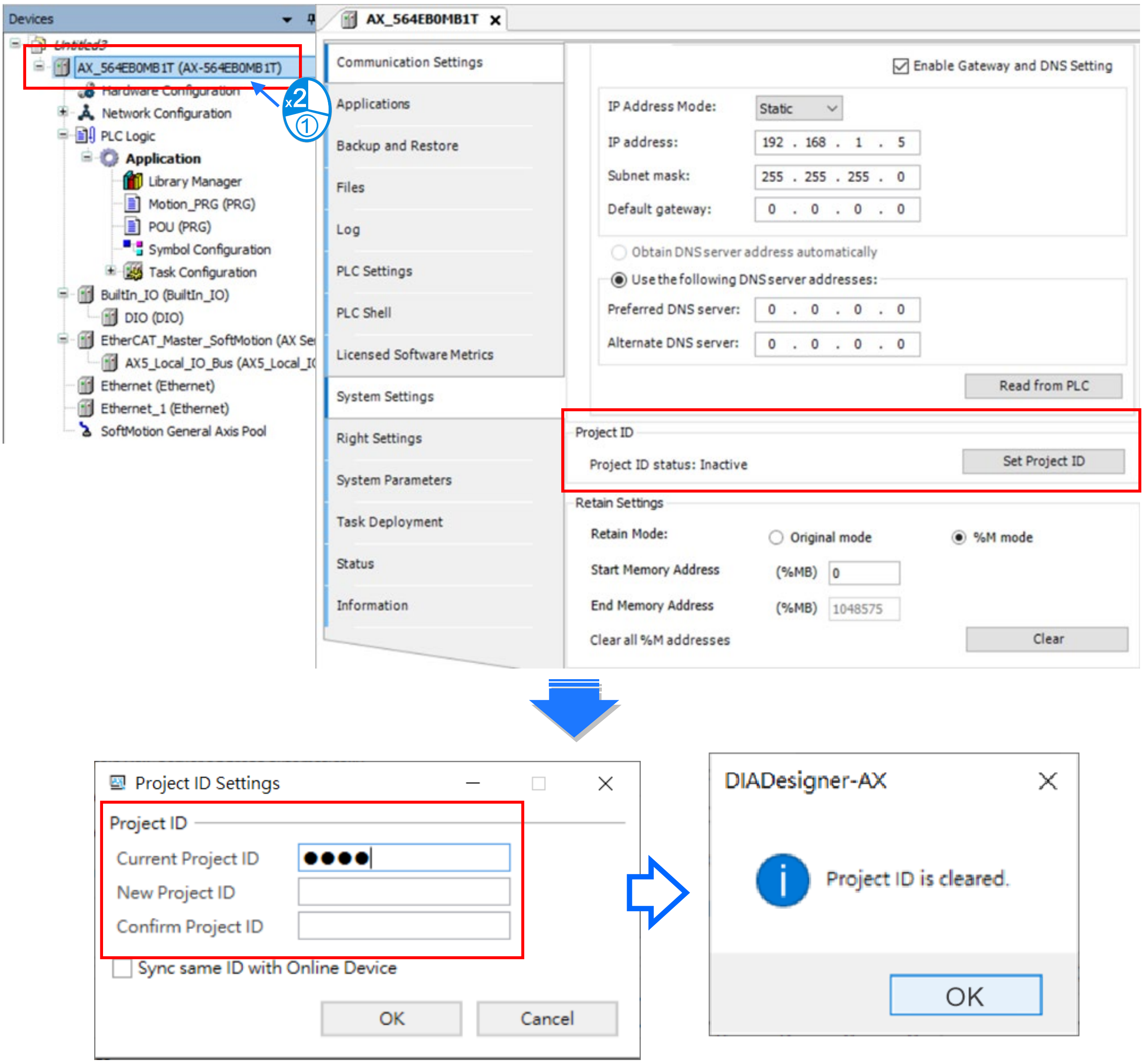
New PLC ID ●●●●

Confirm PLC ID ●●●●

Please also set the Project ID.
If the PLC ID is different from Project ID,
the application will be removed at the next power-on.

9.3.3.3 Delete a Project ID

You can delete a Project ID by resetting it and then leaving the new Project ID blank.
Note: If the original Project ID is lost, it cannot be deleted or modified.



☑ Enable Gateway and DNS Setting
IP Address Mode: Static
IP address: 192 . 168 . 1 . 5
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 0 . 0 . 0 . 0
○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 0 . 0 . 0 . 0
Alternate DNS server: 0 . 0 . 0 . 0

Project ID
Project ID status: Inactive

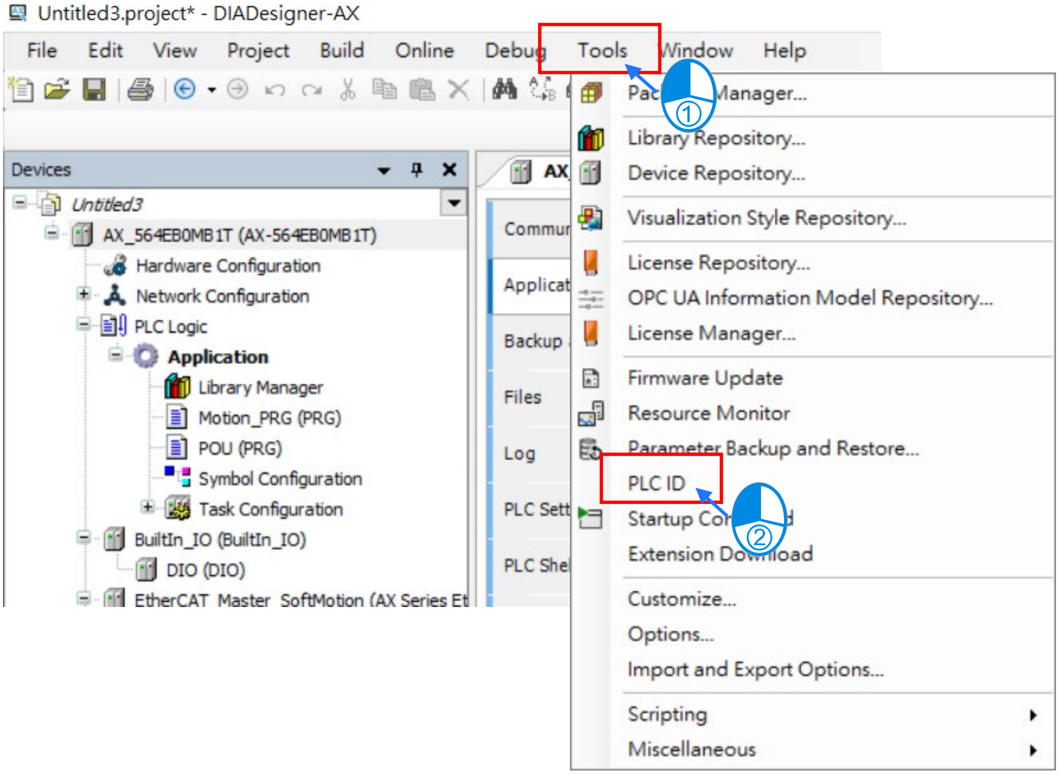
Retain Settings
Retain Mode: ○ Original mode ● %M mode
Start Memory Address (%MB) 0
End Memory Address (%MB) 1048575
Clear all %M addresses

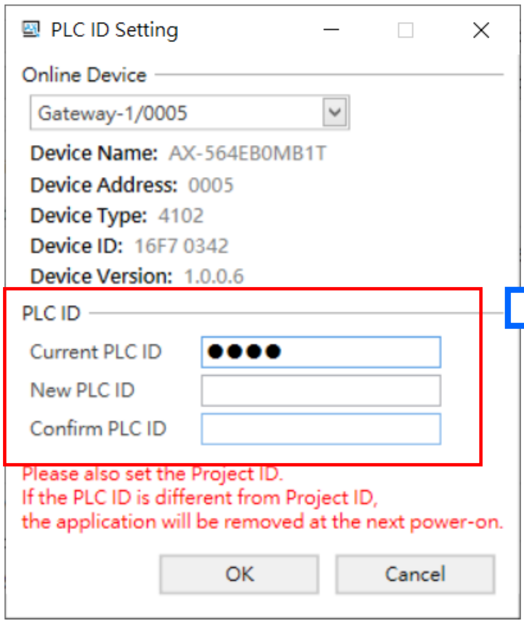
Project ID Settings
Project ID
Current Project ID ●●●●
New Project ID
Confirm Project ID
☐ Sync same ID with Online Device

DIADesigner-AX
Project ID is cleared.

9.3.3.4 Delete a PLC ID

You can delete a PLC ID by resetting it and then leaving the new PLC ID blank.
Note: If the original PLC ID is lost, you need to rest origin the PLC.



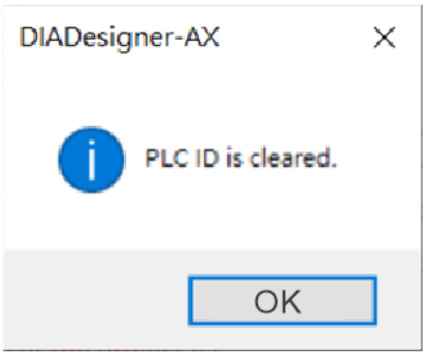


Gateway-1/0005

Device Name: AX-564EB0MB1T
Device Address: 0005
Device Type: 4102
Device ID: 16F7 0342
Device Version: 1.0.0.6

PLC ID
Current PLC ID ●●●●
New PLC ID
Confirm PLC ID

Please also set the Project ID.
If the PLC ID is different from Project ID,
the application will be removed at the next power-on.

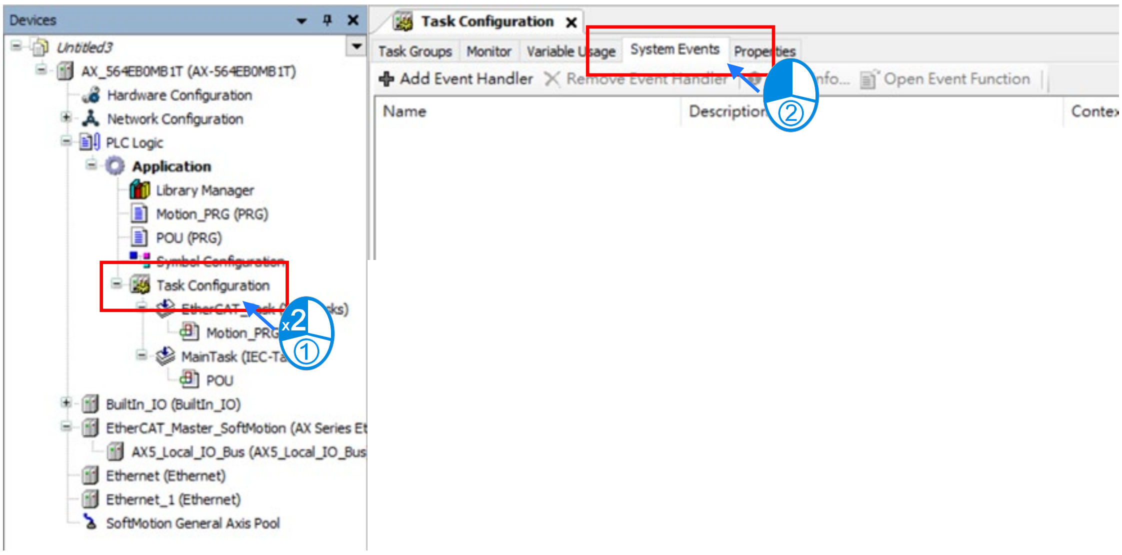


PLC ID is cleared.

9.4 System Event

From the setting page of “System Events” under the “Task Configuraiton”, you can set up Event Handlers and trigger events after the event occurs.

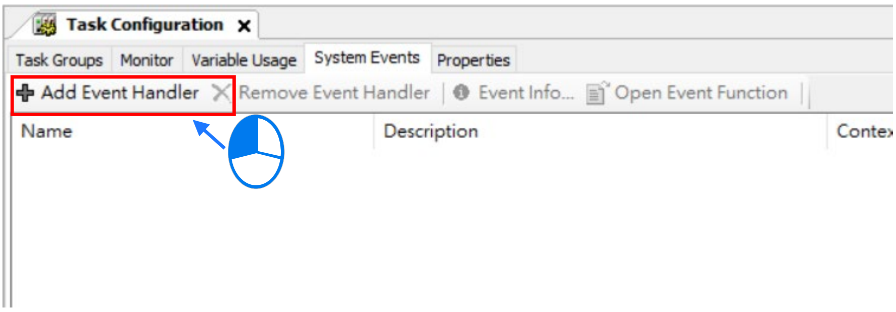
- System Events

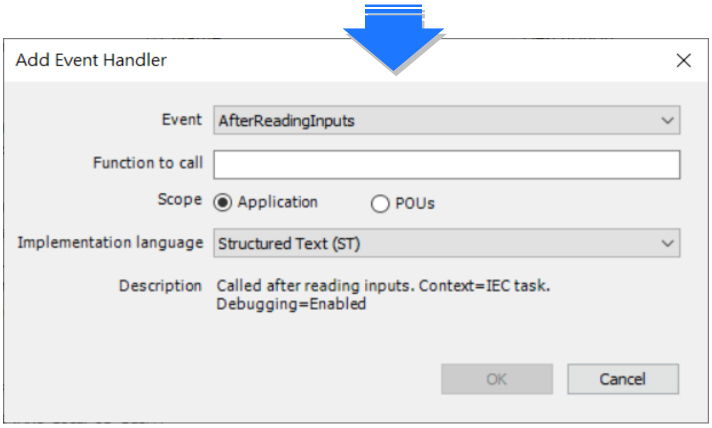


Function button	Description
Add Event Handler	Open the Add Event Handler dialog
Remove Event Handler	Delete the selected event
Event Info	Shows information from the corresponding event library
Open Event Function	Open and edit the selected event

9.4.1 Event Handler

- Add Event Handler




Add Event Handler

Event: AfterReadingInputs

Function to call:

Scope: ● Application ○ POUs

Implementation language: Structured Text (ST)

Description: Called after reading inputs. Context=IEC task. Debugging=Enabled

- On the setting page of Add Event Handler

Item	Description
Event	Event type
Function to Call	Function name
Scope	Application: The function is available to the application. POUs: The function is available to the entire project.
Implementation language	Programming language for the new function
Description	A brief description of the selected event

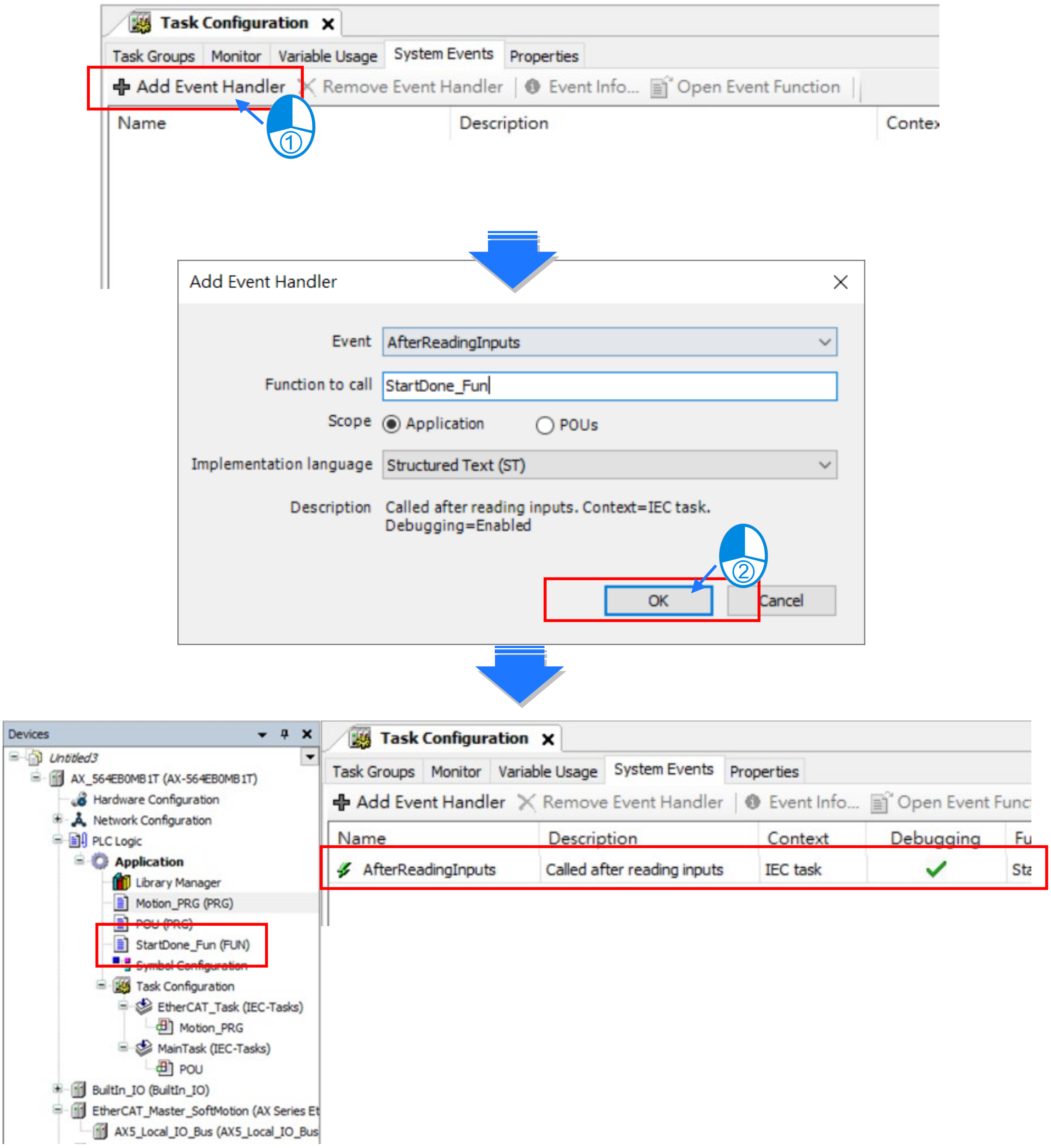
- Possible System Events

Event	Description
PrepareStart	Call before starting the application
StartDone	Call after starting the application
PrepareStop	Call before stopping the application
StopDone	Call after stopping the application
PrepareReset	Call before resetting the application
ResetDone	Call after resetting the application
PrepareOnlineChange	Call before online change of the application
OnlineChangeDone	Call after online change of the application
PrepareDownload	Call before downloading the application
DownloadDone	Call after downloading the application
PrepareDelete	Call before deleting the application
DeleteDone	Call after deleting the application
PrepareExit	Call before exiting the application
ExitDone	Call after exiting the application
CodeInitDone	Event is sent after Code Init. Called within the Task Safe Section and only with an online change. (For example, the copy code for online change is executed here)
Exception	The event is sent if an exception has occurred in the context of an application.
Login	Login of a client to this application
Logout	Logout of a client from this application
BeforeReadingInputs	Call before reading the inputs
AfterReadingInputs	Call after reading the inputs
BeforeWritingOutputs	Call before writing the outputs
AfterWritingOutputs	Call after writing the outputs
DebugLoop	Event is sent in cycles to the debug loop if the IEC task stops at a breakpoint.
PrepareShutdown	Event is sent immediately before the runtime is downloaded.
PrepareExitComm	Event is sent during download before exiting the communication server.
PrepareExitTasks	Event is sent during download before exiting (Exit) all tasks.

9.4.2 Operational Example

This section uses the event “StartDone” as an example. When the state of AX-Series controller changes from Stop to RUN, call event “StartDone” and write the value %MW0 in.

Step 1: Add Event “StartDone”.



Event: AfterReadingInputs

Function to call: StartDone_Fun

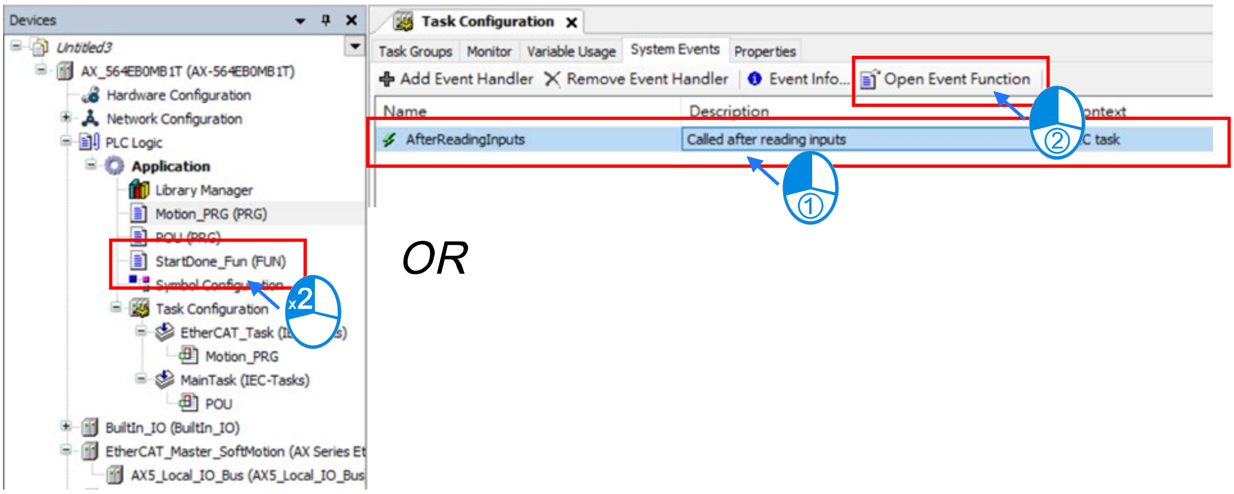
Scope: ● Application ○ POUs

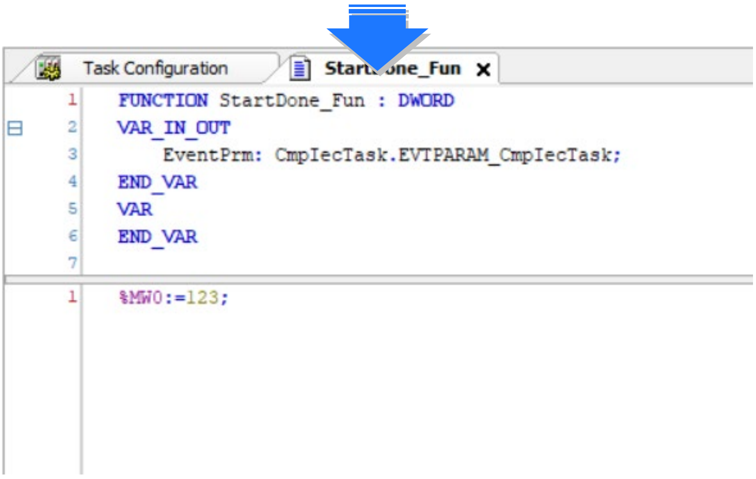
Implementation language: Structured Text (ST)

Description: Called after reading inputs. Context=IEC task. Debugging=Enabled

Name	Description	Context	Debugging	Fu
AfterReadingInputs	Called after reading inputs	IEC task	✓	Sta

Step 2: Set up the event and start programming.





```
FUNCTION StartDone_Fun : DWORD
VAR_IN_OUT
    EventPrm: CmpIecTask.EVTPARAM_CmpIecTask;
END_VAR
VAR
END_VAR
```

```
%MW0:=123;
```

MEMO

Appendix A Troubleshooting

Table of Contents

A.1 Troubleshooting

A.1.1 Basic Troubleshooting Steps

This chapter includes the possible errors that can occur during operation, their causes, and corrective actions.

(1) Check the following:

- The PLC should be operated in a safe environment (consider environmental, electronic, and vibration safeties) .
- Connect power supply correctly to the PLC.
- The installations of the module, terminal, and cable are secured.
- The states shown by LED indicators are normal.
- All switches and dials are set correctly.

(2) Check the following operational functions:

- Toggle the RUN/STOP switch.
- Check the set conditions for the PLC CPU to RUN/Stop.
- Check and mitigate any potential disruptions caused by external devices.
- Use the System Log function in DIADesigner-AX to check system operation and logs.

(3) Identify possible causes of errors:

- AX-5 Series PLC or other external devices
- PLC CPU or extension modules
- Parameters or program settings

A.1.2 Clear the Error States

If the system is in an error state, you can attempt the following methods to resolve the system's error state. However, if the error source persists, the system will return to the error state.

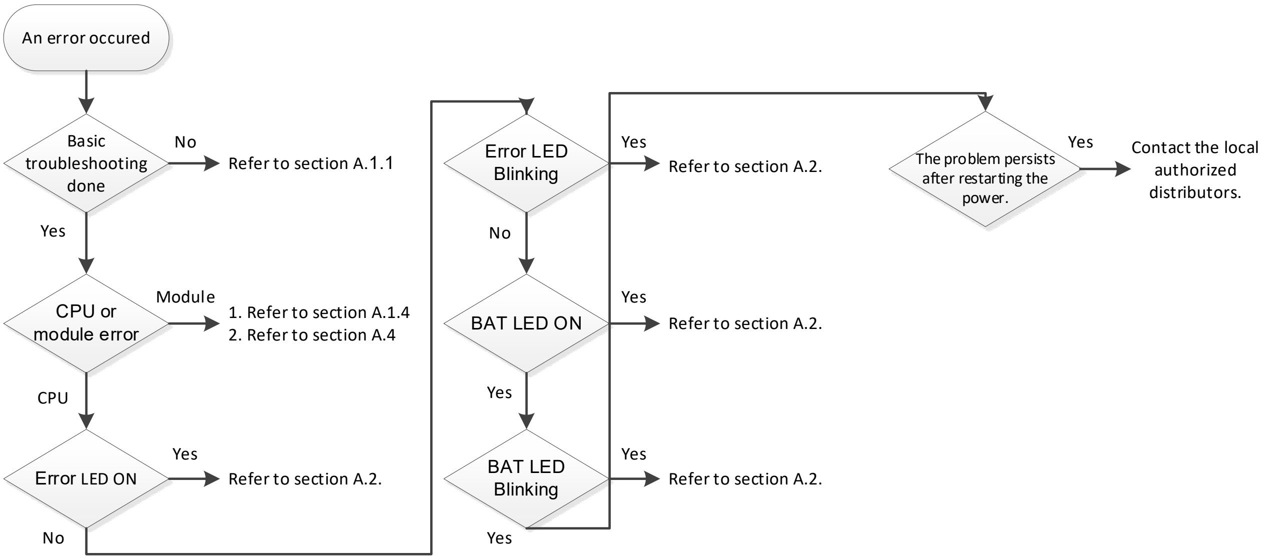
(1) Toggle the PLC CPU model state to STOP and then to RUN.

(2) Restart the PLC CPU.

(3) Use DIADesigner-AX to execute **Reset Warn** to clear the error logs.

(4) Use DIADesigner-AX to execute **Reset Origin** to reset the PLC CPU to default settings and then download the project and open the project to start again.

A.1.3 Troubleshooting SOP

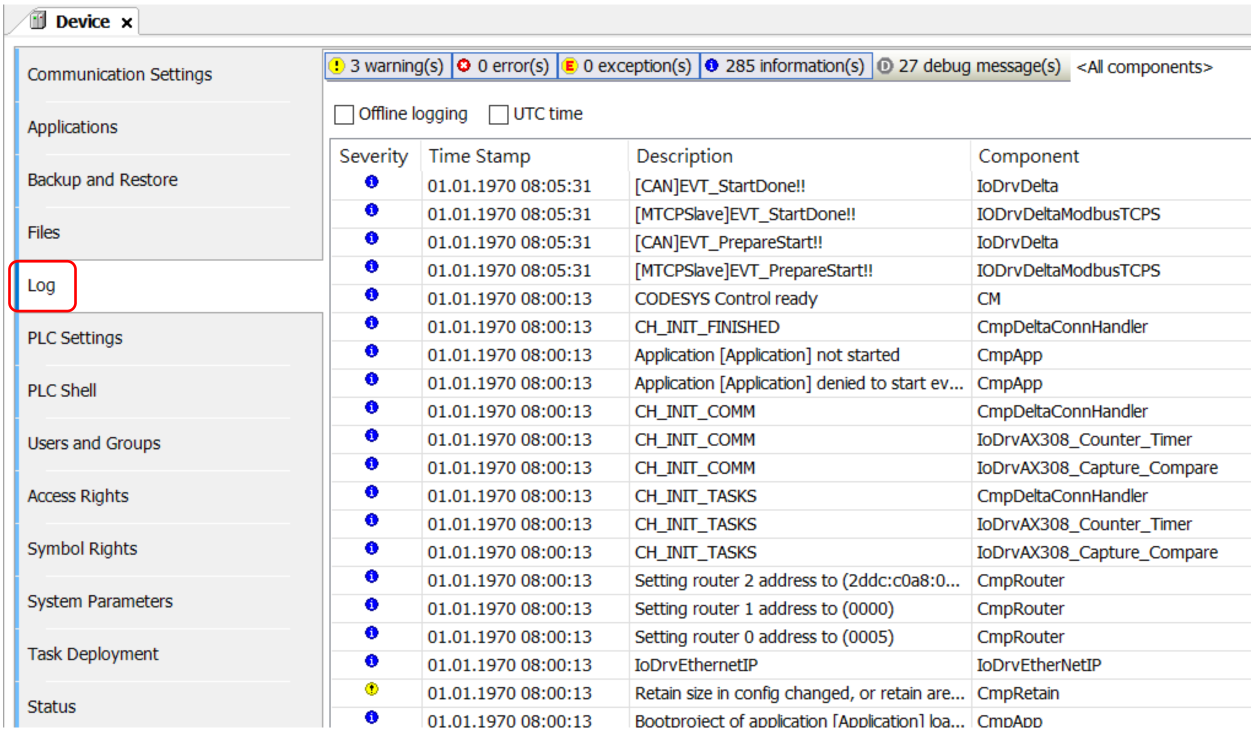


A.1.4 Viewing Log

When an error occurs, the system generates corresponding error codes and stores the error messages in the PLC. You can find events during the startup and shutdown of the system, application download and loading of the boot application, custom entries, log entries from I/O drivers, and log entries from data sources under the Log tab of the Device setting page. Refer to section 4.2.1.5 for more information on Log.

1. Log Tab

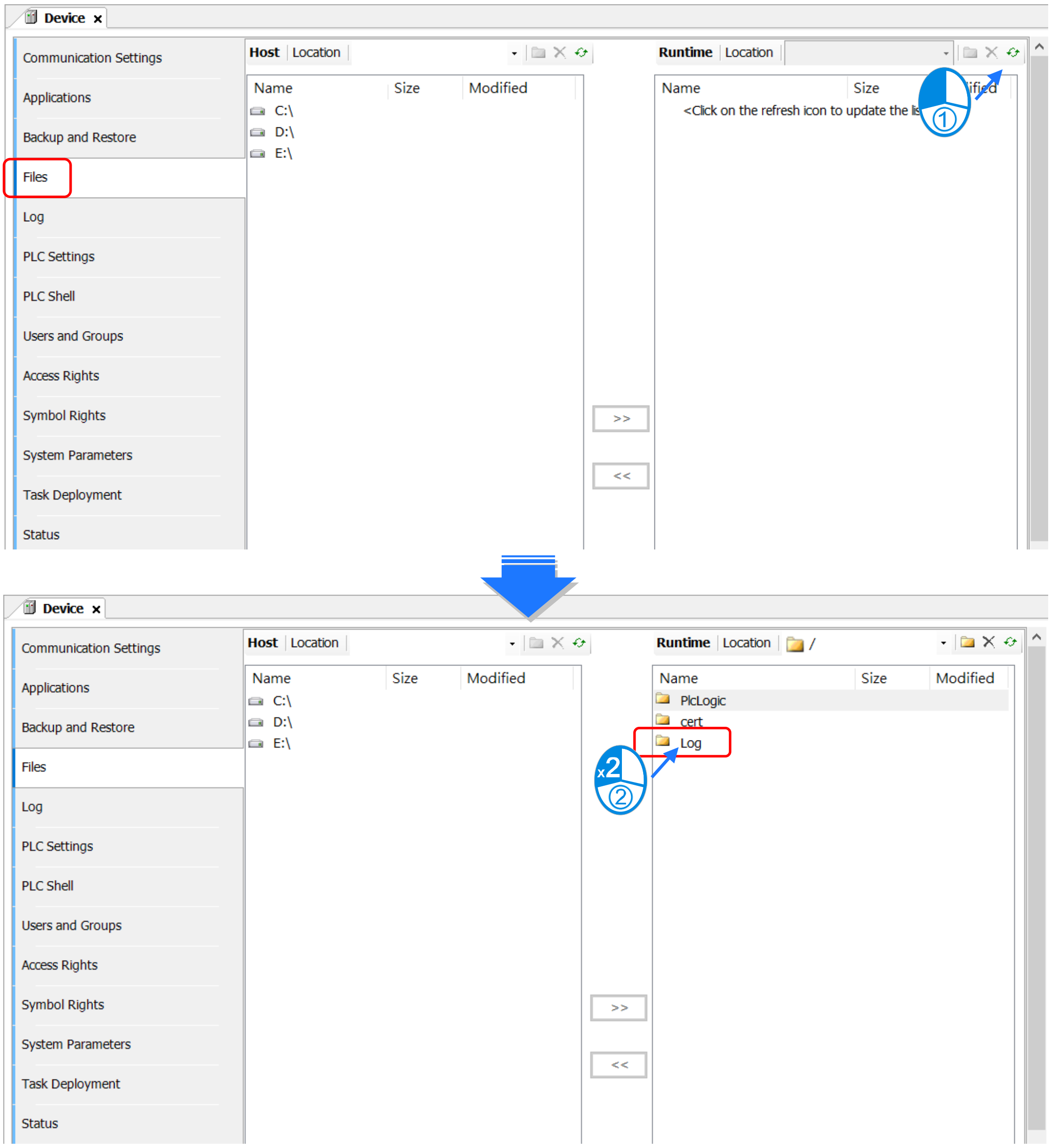
Double-click the **Device** in the tree view to open the Device setting page and then you can find Log tab on the left section.



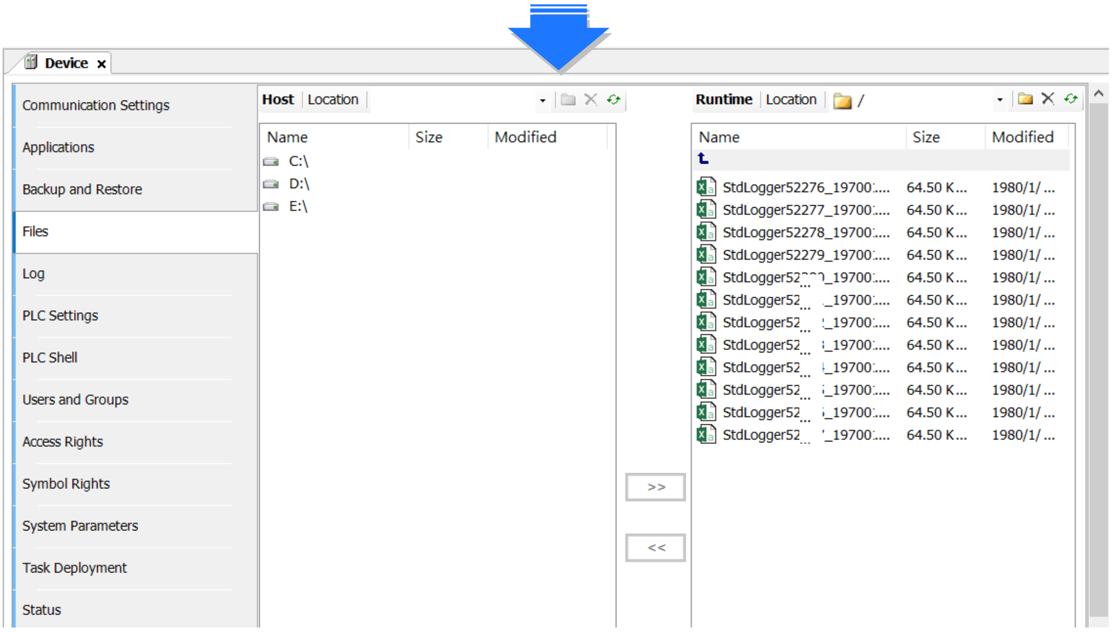
Severity	Time Stamp	Description	Component
i	01.01.1970 08:05:31	[CAN]EVT_StartDone!!	IoDrvDelta
i	01.01.1970 08:05:31	[MTCPSlave]EVT_StartDone!!	IODrvDeltaModbusTCPS
i	01.01.1970 08:05:31	[CAN]EVT_PrepareStart!!	IoDrvDelta
i	01.01.1970 08:05:31	[MTCPSlave]EVT_PrepareStart!!	IODrvDeltaModbusTCPS
i	01.01.1970 08:00:13	CODESYS Control ready	CM
i	01.01.1970 08:00:13	CH_INIT_FINISHED	CmpDeltaConnHandler
i	01.01.1970 08:00:13	Application [Application] not started	CmpApp
i	01.01.1970 08:00:13	Application [Application] denied to start ev...	CmpApp
i	01.01.1970 08:00:13	CH_INIT_COMM	CmpDeltaConnHandler
i	01.01.1970 08:00:13	CH_INIT_COMM	IoDrvAX308_Counter_Timer
i	01.01.1970 08:00:13	CH_INIT_COMM	IoDrvAX308_Capture_Compare
i	01.01.1970 08:00:13	CH_INIT_TASKS	CmpDeltaConnHandler
i	01.01.1970 08:00:13	CH_INIT_TASKS	IoDrvAX308_Counter_Timer
i	01.01.1970 08:00:13	CH_INIT_TASKS	IoDrvAX308_Capture_Compare
i	01.01.1970 08:00:13	Setting router 2 address to (2ddc:c0a8:0...	CmpRouter
i	01.01.1970 08:00:13	Setting router 1 address to (0000)	CmpRouter
i	01.01.1970 08:00:13	Setting router 0 address to (0005)	CmpRouter
i	01.01.1970 08:00:13	IoDrvEthernetIP	IoDrvEtherNetIP
!	01.01.1970 08:00:13	Retain size in config changed, or retain are...	CmpRetain
i	01.01.1970 08:00:13	Bootproject of application [Application] loa...	CmpApp

2. Files

The system generates log files (.csv) when the PLC is power-off or the log exceeds 64 KB. You can read the log file from the Files tab of the Device setting page.



<Click on the refresh icon to update the lis



Host Location

Name	Size	Modified
C:\		
D:\		
E:\		

Runtime Location /

Name	Size	Modified
StdLogger52276_19700:....	64.50 K...	1980/1/ ...
StdLogger52277_19700:....	64.50 K...	1980/1/ ...
StdLogger52278_19700:....	64.50 K...	1980/1/ ...
StdLogger52279_19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...
StdLogger52..._19700:....	64.50 K...	1980/1/ ...

A.2 Troubleshooting of CPU Modules

Check the LED indicators and the error codes from the CPU module and refer to the following table for troubleshooting.

A.2.1 ERROR LED Indicators Blinking Every 0.5 Seconds

- **CPU ERROR**

Error Code (16#)	Description	Solution
2002	File containing corrupted O.S. firmware.	1. Upgrade firmware. 2. After firmware is upgraded, if the problem persists, contact the local authorized distributors.
2005	Failed to initialize system GPIO.	1. Restart the PLC CPU or upgrade firmware. 2. If the problem persists, contact the local authorized distributors.
2006	Failed to create system Semaphore.	1. Restart the PLC CPU or upgrade firmware. 2. If the problem persists, contact the local authorized distributors.
2007	Failed to create system Task.	1. Restart the PLC CPU or upgrade firmware. 2. If the problem persists, contact the local authorized distributors.
2010	Failed to initialize ESC.	1. Restart the PLC CPU or upgrade firmware. 2. If the problem persists, contact the local authorized distributors.
2011	Failed to initialize firmware PRU.	1. Restart the PLC CPU or upgrade firmware. 2. If the problem persists, contact the local authorized distributors.
2014	Failed to initialize system RTC.	Restart the PLC CPU. If the problem persists, contact the local authorized distributors.
2100	The number of MODBUS TCP connections exceeds the range.	Check if the number of Modbus TCP connection (Server+Client) exceeds the maximum number 32.

A.2.2 ERROR LED Indicators Blinking Rapidly Every 0.2 Seconds

The blinking happens when the power supply 24 VDC of the CPU module is disconnected, or the power supply is not sufficient, not stable or abnormal.

Error Code (16#)	Description	Solution
2004	The external voltage is abnormal.	Check whether the external 24 V power supply to the module is normal.

A.2.3 ERROR LED Indicators Slow Blinking Every 0.1 Seconds

Error Code (16#)	Description	Solution
2013	System power error	1. Ensure the power supply is 24 VDC. Do NOT use Type C connector. 2. If the problem persists, contact the local authorized distributors.

A.2.4 ERROR LED Indicators Are ON

The CPU is experiencing a boot sector error.

Error Code (16#)	**Description**	**Solution**
2012	Boot sector failure	1. Upgrade firmware. 2. After firmware is upgraded, if the problem persists, contact the local authorized distributors.
2015	Scan timeout	1. Check if the scan time is set too short. 2. Check if the program has any design flaws that cause long scan times?

A.2.5 BUS FAULT LED Indicators Blinking Every 0.5 Seconds

EtherCAT ERROR (EtherCAT error LED is defined by the Library IODrvEtherCAT.)

Error Code (16#)	**Description**	**Solution**
1	EtherCAT communication lost	Make sure the terminal and cable are properly connected to the CPU module. Execute the function block, DFB_ResetECATMaster, to reset the EtherCAT Master.
2	EtherCAT data mapping failed	Make sure the terminal and cable are properly connected to the CPU module. Execute the function block, DFB_ResetECATMaster, to reset the EtherCAT Master.
4	Incorrect EtherCAT network name	Make sure the Network Name/address is correctly set on the setting page of the EtherCAT Master.
7	EtherCAT Slave failed to initialize	Make sure the actual placement is the same as the settings in the Network Configuration.
8	Vendor ID of the Slave does NOT match.	● Make sure the actual placement is the same as the settings in the Network Configuration. ● Make sure the ESI file of the Slave is matched. ● Disable the Startup Checking item to cancel checking Vendor ID on the EtherCAT Master setting page.
9	Product ID of the Slave does NOT match.	● Make sure the actual placement is the same as the settings in the Network Configuration. ● Make sure the ESI file of the Slave is matched. ● Disable the Startup Checking item to cancel checking Product ID on the EtherCAT Master setting page.

A.2.6 Others

Error Code (16#)	**Description**	**Solution**
2500	The firmware version of the PLC is not in accordance with what stated on the DDF (Device Description File).	Check the firmware version of the PLC and the requirement on the DDF.
2503	An error occurs when the pulse outputs.	Check the log of the corresponding pulse on the ON-LINE monitoring page.
2504	The configuration of the CMP output point does not match.	Check if the output point and CMP device configuration are correct.

A.3 Troubleshooting of the Function Blocks

A.3.1 DL_BuiltInIO

The following errors are specified as warnings and will be stated in the log of the PLC CPU; however, no error LED indicators will show and the AX-5 Series CPU runs without interruptions.

Error Code (16#)	Item Name	Description	Solution
0	DFB_HSIO_NO_ERR	No error on the high speed IO function block	-
186A0	DMC_HP_INVALID_ HOME_SPEED	The speed set in the homing motion on the pulse axis is invalid.	The setting value in the fields of Search for Switch and Search for Z Phase Pulse on the setting page of Pulse Axis cannot not be set to 0. Set a non-zero value.
186A1	DMC_HP_INVALID_ HOME_ACC_DEC	The acceleration set or the deceleration set in the homing motion is invalid.	The setting value in the fields of acceleration and deceleration in the homing motion on the setting page of Pulse Axis cannot not be set to 0. Set a non-zero value.
186A2	DMC_HP_INVALID_ HOME_POSITION	The position set in the homing motion is invalid.	Set the function block pin, lrPosiotion, in the range of [0 to PulseAxis.Modulo Value].
186A3	DMC_HP_AXIS_NOT_PULSEAXIS	The variable of the function block pin is NOT a PulseAxis_REF type.	Make sure to select **Pulse Axis** on the IO Configuration setting page and import IEC Object to the pin “Axis” of the function block DMC_Home_P.
186A4	DMC_HP_HOMING_METHOD_RESERVED	This version does NOT support this type of homing mode.	Check if this type of homing mode is supported in this version. Refer to the specification and then change the mode accordingly.
186A5	DMC_HP_HOMING_MOVEMENT_HW_LIMIT	If the positive/negative limit is activated, the axis cannot move in this homing mode.	Make sure the hardware limit used is supported by this homing mode. Refer to the specification and then change the mode or the setting accordingly.
186A6	DMC_HP_HOMING_ AXIS_STATE_NOT_ STANDSTILL	The state of the pulse axis is not at standstill.	Make sure the function block DMC_Home_P is executed when the axis state is at standstill.
186AC	DFB_CAP_INVALID_CAPTURE_REF	The variable of the function block pin is NOT a Capture_REF type.	Make sure to select **Capture** on the IO Configuration setting page and import IEC Object to the pin “Capture” of the function block DMC_Capture.
186AD	DFB_CAP_INVALID_COUNTER_REF	The variable of the function block pin is NOT a Counter_REF type.	Make sure to select **Counter** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DMC_Capture.
186AE	DFB_CAP_INVALID_ VALUE_SETTING	The mask setting value (uiMaskValue) in DFB_Capture exceeds the range of rotary axis.	Set the pin "uiMaskValue" of the function block DFB_Capture in the range of [0 to EncoderAxis.Modulo Value].
186AF	DFB_CAP_INVALID_DELTARANGE	When the encoder of high-speed counter is a rotary axis and the pin of "diDeltaMax" or "diDeltaMin" exceeds the range of rotary axis.	Set the pin "diDeltaMax" or "diDeltaMin" of the function block DFB_Capture in the range of [0 to EncoderAxis.Modulo Value].
186B0	DFB_CAP_CAPTURE_ALREADY_ENABLE	The device for high-speed capture is already enabled.	Check if the device for high-speed capture is already enabled by other DFB_Capture.

Error Code (16#)	Item Name	Description	Solution
186B6	DFB_CMP_INVALID_COMPARE_REF	The variable of the function block pin is NOT a Compare_REF type.	Make sure to select **Compare** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DMC_Compare.
186B7	DFB_CMP_INVALID_COUNTER_REF	The variable of the function block pin is NOT a Counter_REF type.	Make sure to select **Counter** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DMC_Compare.
186B8	DFB_CMP_INVALID_CMPVALUE	When the encoder of high-speed counter is a rotary axis and the pin of "diCompareValue" exceeds the range.	Set the pin "diCompareValue" of the function block DFB_Compare in the range of [0 to EncoderAxis.Modulo Value].
186B9	DFB_CMP_INVALID_REFRESHCYCLE	The setting value of input pin "wRefreshCycle" exceeds the range of [0-30000], unit 0.1us.	Set the pin "wRefreshCycle" of the function block DFB_Compare in the range of [0 to 30000].
186BA	DFB_CMP_ COMPARE_ALREADY_ENABLE	The device for high-speed compare is already enabled.	Check if the device for high-speed compare is already enabled by other DFB_Compare.
186C0	DFB_HC_INVALID_ COUNTER_REF	The variable of the function block pin is NOT a Counter_REF type.	Make sure to select **Counter** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DMC_HCnt.
186C1	DFB_HC_COUNTER_ALREADY_ENABLE	The device for high-speed counter is already enabled.	Check if the device for high-speed counter is already enabled by other DFB_HCnt.
186C2	DFB_HC_COUNTER_REF_CHANGED_ DURING_ OPERATION	The input pin "Counter" has been changed during the execution of the function block.	Check if the variable of the pin “Counter” has been changed after the execution of the DFB_HCnt.
186C8	DFB_HT_INVALID_ TIMER_REF	The variable of the function block pin is NOT a Timer_REF type.	Make sure to select **Timer** on the IO Configuration setting page and import IEC Object to the pin “Timerr” of the function block DFB_HTmr.
186C9	DFB_HT_TIMER_ ALREADY_ENABLE	The device for high-speed timer is already enabled.	Check if the device for high-speed timer is already enabled by other DFB_HTmr.
186CA	DFB_HT_TIMER_REF_ CHANGED_DURING_OPERATION	The input pin "Timer" has been changed during the execution of the function block.	Check if the variable of the pin “Timer” has been changed after the execution of the DFB_HTmr.
186D0	DFB_PV_INVALID_ COUNTER_REF	The variable of the function block pin is NOT a Counter_REF type.	Make sure to select **Counter** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DFB_PresetValue.
186D1	DFB_PV_NOT_ ENABLE_EXTERNAL_TRIGGER	The counter is not set as triggered externally but the mode of DFB_PresetValue is set to "EXTERNAL_TRIGGER".	Make sure to select **External Trigger** on the Counter Configuration page.
186D2	DFB_PV_PREVIOUS_PRESET_NOT_DONE	The preset counting function of the counter has been enabled by other function block	Execute this function block after the execution of DFB_PresetValue of this counter completes.

Error Code (16#)	Item Name	Description	Solution
		DMC_PresetValue and is not done yet.	
186D3	DFB_PV_CANNOT_ PRESET_WHEN_SAMPLING	The counter is executing DFB_Sample.	Disable the sample function of this counter. Disable DFB_Sample of this counter.
186D4	DFB_PV_SETRING_ NOT_DONE	The counter is executing DFB_SetRing and is not done yet.	Execute this function block after the execution of DFB_SetRing of this counter completes.
186D5	DFB_PV_INVALID_ PRESET_VALUE	When the encoder of high-speed counter is a rotary axis and the pin of "diPresetValue" exceeds the range.	Set the pin "diPresetValue" of the function block in the range of [0 to EncoderAxis.Modulo Value].
186D6	DFB_PV_COUNTER_REF_CHANGED_ DURING_ OPERATION	The input pin "Counter" has been changed during the execution of the function block.	Check if the variable of the pin “Counter” has been changed after the execution of the DFB_PresetValue.
186DC	DFB_SP_INVALID_ COUNTER_REF	The variable of the function block pin is NOT a Counter_REF type.	Make sure to select **Counter** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DMC_Sample.
186DD	DFB_SP_COUNTER_NOT_ENABLE	The function block DFB_Counter is not enabled yet.	Execute DFB_Sample after making sure this counter is enabled by DFB_HCnt.
186DE	DFB_SP_ALREADY_SAMPLING	The counter is executing DFB_Sample.	Check if this counter is enabled by other DFB_Sample.
186DF	DFB_SP_PRESET_ NOT_DONE	The counter is executing DFB_PresetValue and is not done yet.	Execute this function block after the execution of DFB_PresetValue of this counter completes.
186E0	DFB_SP_INVALID_ SAMPLE_TIME	The setting value of input pin “wSampleTime” of the function block DFB_Sample exceeds the range of [10-65535].	Set the pin “wSampleTime” of the function block DFB_Sample in the range of [10 to 65535].
186E1	DFB_SP_COUNTER_REF_CHANGED_ DURING_ OPERATION	The input pin "Counter" has been changed during the execution of the function block.	Check if the variable of the pin “Counter” has been changed after the execution of the DFB_Sample.
186E7	DFB_SR_INVALID_ COUNTER_REF	The variable of the function block pin is NOT a Counter_REF type.	Make sure to select **Counter** on the IO Configuration setting page and import IEC Object to the pin “Counter” of the function block DFB_SetRing.
186E8	DFB_SR_COUNTER_HAS_NO_CHILD_ ENCODER_AXIS	No child node of the high-speed counter is connected to the encoder.	Insert EncoderAxis into the counter and set the encoder type to rotary axis and re-execute the function block.
186E9	DFB_SR_COUNTER_NOT_RING	The encoder of the high-speed counter is not a rotary axis type.	Select the encoder type to rotary axis on the Counter Configuration page.
186EA	DFB_SR_PREVIOUS_SETRING_NOT_ DONE	The preset counting function of the counter has been enabled by other function block DMC_SetRing and is not done yet.	Execute this function block after the execution of DFB_SetRing of this counter completes.

Error Code (16#)	Item Name	Description	Solution
186EB	DFB_SR_PRESET_NOT_DONE	The counter is executing DFB_PresetValue and is not done yet.	Execute this function block after the execution of DFB_PresetValue of this counter completes.
186EC	DFB_SR_INVALID_RING_RANGE	When the encoder of high-speed counter is a rotary axis and the pin of "diPositionPeriod" is less than 0 and bigger than the setting value of bSetDown.	Set the pin "diPositionPeriod" of the function block bigger than 0 and less than the setting value of bSetDown.
186ED	DFB_SR_COUNTER_REF_CHANGED_DURING_OPERATION	The input pin "Counter" has been changed during the execution of the function block.	Check if the variable of the pin “Counter” has been changed after the execution of the DFB_SetRing.

A.3.2 Motion Control Related Instructions

The errors occurred in DL_MotionControl or DL_MotionControlLight are specified as warnings; however no error indicators will appear and the AX Series CPU can still run. Refer to AX Series Motion Controller Manual for the troubleshooting of DL_MotionControl.

A.3.3 DL_ModbusComMaster

The errors occurred in DL_ModbusComMaster are specified as warnings; however, no error indicators will appear, and the AX Series CPU can still run. Refer to AX Series Motion Controller Manual for the troubleshooting.

Error Code (16#)	Item Name	Description	Solution
19E12	DFB_ILLEGAL_DATA_ADDRESS	The device’s address is not supported by the slave.	Ensure the device address is correct.
19E1D	DFB_RESPONSE_CRC_ERROR	The response to the communication command from the slave contains an invalid checksum.	Check if the response received from the slave is accurate.
19E1E	DFB_RESPONSE_WRONG_SLAVE	The response to the station number from the slave is incorrect.	Check if the response received from the slave is accurate.
19E1F	DFB_RESPONSE_WRONG_FUNCTIONCODE	The response to the function code from the slave is incorrect.	Check if the response received from the slave is accurate.
19E20	DFB_REQUEST_FAILED_TO_SEND	Failed to send requests.	Restart the PLC CPU. If the problem persists, contact the local authorized distributors.
19E21	DFB_RESPONSE_INVALID_PROTOCOL	The function code response to the communication command from the slave is invalid.	Check if the response received from the slave is accurate.
19E22	DFB_RESPONSE_INVALID_HEAD	The response to communication format from the slave is incorrect.	Check if the response received from the slave is accurate.
19E23	DFB_INVALID_CHANNEL_INDEX	The setting of Function Block Channel Index is invalid.	Ensure the settings of Function Block Channel Index are correct and the channel index is already set.

Error Code (16#)	Item Name	Description	Solution
19E24	DFB_CHANNEL_SETTING_NOT_SUPPORT	The DFB_ModbusComChannel is not supported by the Channel. Use an alternative trigger method.	● Check if the channel is enabled. ● Check if the Channel Trigger mode is Application.
19E25	DFB_INVALID_COMPORT	Setting error in the Function Block byComPort	Check if the setting value of byComPort is within the acceptable range.
19E26	DFB_INVALID_BUFFER	No buffer is set for the Function Block to send or receive data.	Set a buffer for the Function Block to send and receive data to and from the available device or variable address.
19E27	DFB_INVALID_LENGTH	Data length setting error in the Function Block	Check if the setting value of the data length is within the acceptable range.
19E28	DFB_INVALID_SLAVE_ADDRESS	Slave number setting error in the Function Block	Check if the slave number setting is correct.
19E29	DFB_INVALID_FUNCTION_CODE	Function code setting error in the Function Block	Check if the function code setting is correct.
19E2A	DFB_NO_MASTER_CONFIG	Delta_Modbus_Master_COM_Port is NOT added.	Ensure to add Delta_Modbus_Master_COM_Port.
19E2B	DFB_MEMORY_NOT_ENOUGH	Insufficient memory for Function Blocks	Minimize the amount of function blocks.

A.3.4 DL_ModbusTCPMaster

The errors occurred in DL_ModbusTCPMaster are specified as warnings; however, no error indicators will appear, and the AX Series CPU can still run. Refer to AX Series Motion Controller Manual for the troubleshooting.

Error Code (16#)	Item Name	Description	Solution
19E11	DFB_ILLEGAL_FUNCTION	The response to the Modbus Function Code from the slave is not supported.	Ensure the slave is correctly set.
19E12	DFB_ILLEGAL_DATA_ADDRESS	The device's address is not supported by the slave.	Ensure the device address is correct.
19E13	DFB_ILLEGAL_DATA_VALUE	The response from the slave contains incorrect information.	Check if the slave settings are correct and if there has been any disruption in the transmission process.
19E1C	DFB_RESPONSE_TIMEOUT	The slave did not respond within the expected time.	Check if there has been a response from the slave and if the wiring is done correctly.
19E1F	DFB_RESPONSE_WRONG_FUNCTIONCODE	The response to the function code from the slave is incorrect.	Check if the response received from the slave is accurate.
19E20	DFB_REQUEST_FAILED_TO_SEND	Failed to send requests.	Restart the PLC CPU. If the problem persists, contact the local authorized distributors.
19E21	DFB_RESPONSE_INVALID_PROTOCOL	The function code response to the communication command from the slave is invalid.	Check if the response received from the slave is accurate.
19E22	DFB_RESPONSE_INVALID_HEAD	The response to communication format from the slave is incorrect.	Check if the response received from the slave is accurate.
19E23	DFB_INVALID_CHANNEL_INDEX	The setting of Function Block Channel Index is invalid.	Ensure the settings of Function Block Channel Index are correct and the channel index is already set.

Error Code (16#)	Item Name	Description	Solution
19E24	DFB_CHANNEL_SETTING_NOT_SUPPORT	The DFB_ModbusComChannel is not supported by the Channel. Use an alternative trigger method.	● Check if the channel is enabled. ● Check if the Channel Trigger mode is Application.
19E25	DFB_INVALID_SLAVE	The pin configuration for the slave is incorrect.	Check if the slave has been set up in the project.
19E26	DFB_INVALID_BUFFER	No buffer is set for the Function Block to send or receive data.	Set a buffer for the Function Block to send and receive data to and from the available device or variable address.
19E27	DFB_INVALID_LENGTH	Data length setting error in the Function Block	Check if the setting value of the data length is within the acceptable range.
19E29	DFB_INVALID_FUNCTION_CODE	Function code setting error in the Function Block	Check if the function code setting is correct.
19E2C	DFB_MEMORY_NOT_ENOUGH	Insufficient memory for Function Blocks	Minimize the amount of function blocks.
19E2D	DFB_CONNECTION_TIMEOUT	Timeout occurs when establishing a Modbus TCP connection	Ensure that the network cable is connected securely, and that the client station is operating.
19E2E	DFB_CONNECTION_FAILED	The Modbus TCP connection establishment is being declined.	Ensure the slave is correctly set and the Modbus TCP function is enabled.

A.3.5 IoDrvEtherCATLib

A.3.5.1 AL Status

The errors occurred in AL_Status are specified as warnings; however, no error indicators will appear, and the AX Series CPU can still run. Refer to AX Series Motion Controller Manual for the troubleshooting.

Error Code (16#)	Item Name	Description
0	No errors	
1	Unspecified error	
2	No memory	
11	Invalid requested state change	
12	Unknown requested state	
13	Bootstrap not supported	
14	No valid firmware	
15	Invalid Mailbox Configuration	The device cannot be configured. Maybe due to the data from the ESI file does not match the device.
16	Invalid Mailbox Configuration	
17	Invalid sync manager configuration	
18	No valid inputs	
19	No valid outputs	
1A	Synchronization error	The DC settings may be incorrect.
1B	Sync manager watchdog	The connection was interrupted, possibly due to a temporary cable disconnection or the PLC being stopped.
1C	Invalid sync manager types	
1D	Invalid output configuration	The device cannot be configured. Maybe due to the data from the ESI file does not match the device.
1E	Invalid input configuration	The device cannot be configured. Maybe due to the data from the ESI file does not match the device.
1F	Invalid watchdog configuration	
20	Slave needs cold start.	Turn the device off and then turn it back on.
21	Slave needs INIT	
22	Slave needs PREOP	
23	Slave needs SAFEOP	
24	Invalid input mapping	
25	Invalid output mapping	
26	Inconsistent settings	
27	Free-Run not supported	The device must be configured with “Distributed Clock”.
28	Synchronization NOT supported.	The device does not support “Distributed Clock”.
29	Free-Run needs 3 buffer mode.	
2A	Background watchdog	
2B	No valid inputs and outputs	

Error Code (16#)	Item Name	Description
2C	Fatal Sync error	
2D	No Sync error	The synchronization with DC is not successful, possibly because the jitter of the runtime is too large.
30	Invalid DC SYNCH configurations	
31	Invalid DC latch configurations	
32	PLL error	The slave synchronization is unsuccessful.
33	Invalid DC IO error	
34	Invalid DC timeout error	
35	DC invalid sync cycle time	The configuration of “Distributed Clock” does not match the device.
36	DC Sync0 cycle time	The configuration of “Distributed Clock” does not match the device.
37	DC Sync1 cycle time	The configuration of “Distributed Clock” does not match the device.
41	MBX_AOE	
42	MBX_EOE	
43	MBX_COE	
44	MBX_FOE	
45	MBX_SOE	
4F	MBX_VOE	
50	Cannot visit EEPROM	
51	EEPROM error	
60	Restart the slave locally.	

A.4 Troubleshooting of I/O Modules

- **Introduction to troubleshooting modules**

The following AX-5 series modules can be installed in an AX-5 Series system. There are 2 types of error codes; error and warning. The CPU module and its modules stop operating when errors occur. The CPU modules and its modules do not stop operating when warnings are triggered.

A.4.1 Troubleshooting of Analog Modules (AD/DA/XA) and Temperature Modules (PT/TC)

A.4.1.1 ERROR LED Indicators Are ON

You can set up the option to be **True** in **Module Alarm Setting** to have the following errors appear as warnings when they occur. Otherwise, when an error occurs, only an error message appears.

Index	Subindex	Description	LED		Solution
			Error LED	Channel LED	
16#A000	16#01	(16#FF01) Unit Power Error	ON	ON	Check the power supply.
	16#02	(16#FF02) EtherCAT Connection Lost	ON	ON	1. Check if the connections for the modules are secured. 2. After checking the connection, reset the module's ECAT status through CPU module.
	16#03	(16#FF03) ESC or EEPROM Error	ON	ON	If the problem persists, contact the local authorized distributors.
	16#04	(16#FF04) Flash Error	ON	ON	If the problem persists, contact the local authorized distributors.
	16#05	(16#FF05) Analog IC Error	ON	ON	If the problem persists, contact the local authorized distributors.
	16#06	(16#FF06) Analog Power Error	ON	ON	If the problem persists, contact the local authorized distributors.

A.4.1.2 ERROR LED Indicators Blinking Every 1 / 0.2 Seconds

The following errors are specified as warnings to ensure that the AX-3 Series CPU can still run even when the warnings are triggered.

Index	Subindex	Description	LED		Solution
			Error LED	Channel LED	
16#A00**n**[*1]	16#01	(16#FF11) CH (**n**-1) Upper Limit Alarm	Every 1 second blinking	ON	Check if the channel signal meets the upper limit setting.
	16#02	(16#FF12) CH (**n**-1) Lower Limit Alarm	Every 1 second blinking	ON	Check if the channel signal meets the lower limit setting.
	16#03	(16#FF13) CH (**n**-1) Over Hardware range	Every 0.2 seconds blinking	OFF	Check if the input signal of the channel meets the product specifications.
	16#04	(16#FF14) CH (**n**-1) Under Hardware range	Every 0.2 seconds blinking	OFF	Check if the input signal of the channel meets the product specifications.
	16#05	(16#FF15) CH (**n**-1) Wire Break	Every 0.2 seconds blinking	OFF	Check if the sensor connection is secure.

*1. n = 1 to 4

A.4.2 Troubleshooting of Positioning Module (Pulse Unit)

A.4.2.1 ERROR LED Indicators Are ON

- **Error Message**

Index	Subindex	Error Code	Description	Error LED	Solution
16#A000	16#01	16#FF01	Unit Power Error	ON	Check the power supply.
	16#02	16#FF02	EtherCAT Connection Lost	ON	1. Check if the connections for the modules are secured. 2. After checking the connection, reset the module's ECAT status through CPU module.
	16#03	16#FF03	ESC or EEPROM Error	ON	If the problem persists, contact the local authorized distributors.
	16#04	16#FF04	Flash Error	ON	If the problem persists, contact the local authorized distributors.

A.4.2.2 ERROR LED Blinking Every 2/0.2 Seconds

- **Warning message**

Index	Subindex	Error Code	Description	Error LED	Solution
16#A001	16#01	16#FF81	Channel Hardware Alarm	Every 2 seconds blinking	1. Check the I/O power supply. 2. Check if the output circuit of the module is short-circuited. 3. If the problem persists, contact the local authorized distributors.

- **Error Message**

Index	Subindex	Error Code	Description	Error LED	Solution
16#603F 16#683F	16#00	16#6320	PARAMETER_ERROR	Every 0.2 seconds blinking	Check if the parameters required for the activation of this axis are correct.
		16#7120	MOTOR_ERROR	Every 0.2 seconds blinking	Check if Check if the axis motor driver is normal. (ALM bit triggered?)
		16#7500	COMMUNICATION	Every 0.2 seconds blinking	1. Check if the connections for the modules are secured. 2. After checking the connection, reset the module's ECAT status through CPU module.
		16#FF20	OverMaxFreq	Every 0.2 seconds blinking	Check the configured value for the axis pulse frequency.
		16#FF21	LSPOccur	Every 0.2 seconds blinking	Check the travel limits of the drive system for the specified axis.
		16#FF22	LSNOccur	Every 0.2 seconds blinking	Check the travel limits of the drive system for the specified axis.
		16#FF23	IMSTOPOccur	Every 0.2 seconds blinking	Determine if the axis causes an abrupt halt.
		16#FF24	ServoOnAtFactoryMode	Every 0.2 seconds blinking	Check if the device is in factory mode.

A.4.3 Troubleshooting of Positioning Module (High-speed Counter)

A.4.3.1 ERROR LED Indicators Are ON

- **Error Message**

Index	Subindex	Error Code	Description	Error LED	Solution
16#A000	16#01	16#FF01	Unit Power Error	ON	Check the power supply.
	16#02	16#FF02	EtherCAT Connection Lost	ON	1. Check if the connections for the modules are secured. 2. After checking the connection, reset the module's ECAT status through CPU module.
	16#03	16#FF03	ESC or EEPROM Error	ON	If the problem persists, contact the local authorized distributors.
	16#04	16#FF04	Flash Error	ON	If the problem persists, contact the local authorized distributors.
16#200A 16#201A	16#01	16#FF02	EtherCAT Connection Lost	ON	1. Check if the connections for the modules are secured. 2. After checking the connection, reset the module's ECAT status through CPU module.

A.4.3.2 ERROR LED Blinking Every 2/0.2 Seconds

- **Warning message**

Index	Subindex	Error Code	Description	Error LED	Solution
16#A001	16#01	16#FF81	Channel Hardware Alarm	Every 2 seconds blinking	1. Check the I/O power supply. 2. Check if the output circuit of the module is short-circuited. 3. If the problem persists, contact the local authorized distributors.

- **Error Message**

Index	Subindex	Error Code	Description	Error LED	Solution
16#200A	16#02	16#FF20	CH0 Counter is not enabled	Every 0.2 seconds blinking	1. Check if CH0 Counter is enabled before using this function. 2. After Step 1 is done, set the CH0 Counter Operation Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.
		16#FF23	CH0 DI Function is not set to CLR	Every 0.2 seconds blinking	1. Check if CH0 DI Function DI0 has been set to CLR0 or if CH0 DI Function DI2 has been set to CLR1 before using the function of CH0 CounterExternalReset. 2. After Step 1 is done, set the CH Counter Operation Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.

Index	Subindex	Error Code	Description	Error LED	Solution
	16#03	16#FF21	CH0 DI Function is not set to LATCH0	Every 0.2 seconds blinking	1. Check if CH0 DI Function DI0 has been set to Latch0 before using DI0 as "Latch". 2. After Step 1 is done, set the CH Latch0 Input Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.
		16#FF22	CH0 DI Function is not set to LATCH1	Every 0.2 seconds blinking	1 Check if CH0 DI Function DI2 has been set to Latch1 before using DI2 as "Latch1". 2 After Step 1 is done, set the CH Latch1 Input Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.
16#201A	16#02	16#FF20	CH1 Counter is not enabled	Every 0.2 seconds blinking	1. Check if CH1 Counter is enabled before using this function. 2. After Step 1 is done, set the CH Counter Operation Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.
		16#FF23	CH1 DI Function is not set to CLR	Every 0.2 seconds blinking	1. Check if CH1 DI Function DI0 has been set to CLR0 or if CH1 DI Function DI2 has been set to CLR1 before using the function of CH1 CounterExternalReset. 2. After Step 1 is done, set the CH Counter Operation Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.
	16#03	16#FF21	CH1 DI Function is not set to LATCH0	Every 0.2 seconds blinking	1. Check if CH1 DI Function DI1 has been set to Latch0 before using DI as "Latch0". 2. After Step 1 is done, set the CH Latch0 Input Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.
		16#FF22	CH1 DI Function is not set to LATCH1	Every 0.2 seconds blinking	1. Check if CH1 DI Function DI3 has been set to Latch1 before using DI3 as "Latch1". 2. After Step 1 is done, set the CH Latch1 Input Control to 0, and then execute ErrorReset. Or reset the module's ECAT status through CPU module.

A.4.4 Troubleshooting of Load Cell Modules

A.4.4.1 ERROR LED Indicators Are ON

Index	Subindex	Description	LED		Solution
			Error LED	Channel LED	
16#A000	16#01	(16#FF01) Unit Power Error	ON	OFF	Check the power supply.
	16#02	(16#FF02) EtherCAT Connection Lost	ON	OFF	1. Check if the connections for the modules are secured. 2. After checking the connection, reset the module's ECAT status through CPU module.
	16#03	(16#FF03) ESC or EEPROM Error	ON	OFF	If the problem persists, contact the local authorized distributors.
	16#04	(16#FF04) Flash Error	ON	OFF	If the problem persists, contact the local authorized distributors.
	16#05	(16#FF05) Analog IC Error	ON	OFF	If the problem persists, contact the local authorized distributors.
	16#06	(16#FF06) Analog Power Error	ON	OFF	If the problem persists, contact the local authorized distributors.

A.4.4.2 ERROR LED Blinking Every 0.2/1 Seconds

Index	Subindex	Description	LED		Solution
			Error LED	Channel LED	
16#6100	16#02	(16#FF11) Input Voltage Error	Every 0.2 seconds blinking	OFF	Check if the channel signal meets the specification.
	16#03	(16#FF12) Weight Limit Exceeded	Every 1 second blinking	ON	Check if the weight setting meets the upper limit setting.
	16#04	(16#FF13) Calibration Error	Every 1 second blinking	ON	Recalibrate

A.5 Error Codes and LED Indicators for CPU Modules

A. Columns

a. Error code: If an error occurs in the system, an error code is generated.

b. Description: The description of the error

c. CPU status: If the error occurs, the status of the CPU.

- Stop: The CPU stops running when the error occurs.
- Continue: The CPU keeps running when the error occurs.

d. LED indicator status: If the error occurs, the LED indicator is ON, OFF, or blinks.

- ERROR: System error

- **Descriptions**

Module Type	LED indicator	Descriptions
CPU	Error LED	There are five types of indicators for of the CPU module errors, including LED indicator ON, OFF, blinking fast, blinking normally, and blinking slowly. When the LED indicator is ON, blinking fast/normally, clear the problems first for the CPU module to keep on running. When the LED indicator is blinking slowly, indicating it is a warning type of error codes; it does not require immediate action. Clear the problems when the module is not running. Error type: ON: A severe error occurs in the module. Blinking fast (every 0.2 seconds): Unstable power supply or hardware failure. Blinking normally (every 0.5 second): System program errors or system cannot run. Warning type: Blinking slowly (every 1 second): A warning is triggered, but the system can still run. OFF: This warning is shown in DIADesigner-AX; You can modify the rules in DIADesigner-AX and use LED indicators to indicate the triggered errors.

A.5.1 Error Codes and LED Indicators for CPU Modules

- **CPU ERROR**

Error Code (16#)	Description	CPU status	ERROR LED indicator				
			ON	Blinking fast	Blinking normally	Blinking slowly	OFF
2002	File containing corrupted O.S. firmware.	Stop			V		
2004	24 vdc power supply is not sufficient and then is recovered from low-voltage.	Stop		V			
2005	Failed to initialize system GPIO.	Stop			V		
2006	Failed to create system Semaphore.	Stop			V		
2007	Failed to create system Task.	Stop			V		
2010	Failed to initialize ESC.	Stop			V		
2011	Failed to initialize firmware PRU.	Stop			V		
2012	Boot sector failure	Stop	V				
2013	System power error	Stop				V	
2014	Failed to initialize system RTC.	Stop			V		
2015	Scan timeout	Stop	V				
2100	The number of Modbus TCP connections exceeds the range.	Stop			V		
2500	The firmware version of the PLC is not in accordance with what stated on the DDF (Device Description File).	Continue					
2503	An error occurs when the pulse outputs.	Continue					
2504	The configuration of the CMP output point does not match.	Stop			V		

- **EtherCAT ERROR**

Error Code (16#)	Description	CPU status	ERROR LED indicator*2				
			ON	Blinking fast	Blinking normally	Blinking slowly	OFF
1	EtherCAT communication lost	User-defined*1			V		
2	EtherCAT data mapping failed	User-defined*1			V		
4	Incorrect EtherCAT network name	User-defined*1			V		
7	EtherCAT Slave failed to initialize	User-defined*1			V		
8	Vendor ID of the Slave does NOT match.	User-defined*1			V		
9	Product ID of the Slave does NOT match.	User-defined*1			V		

Note 1: When an error occurs, how the PLC CPU works is according to what you have set in “CPU Module Stop when EtherCAT Bus Error”.

Note 2: Blinking fast: blinks every 0.2 seconds;
Blinking normally: blinks every 0.5 seconds;
Blinking slowly: blinks every 1 second.

A.5.2 Error Codes and LED Indicators for Analog and Temperature Modules

Index	Subindex	Description	LED	
			Error LED	Channel LED
16#A000	16#01	(16#FF01) Unit Power Error	ON	OFF
	16#02	(16#FF02) EtherCAT Connection Lost	ON	OFF
	16#03	(16#FF03) ESC or EEPROM Error	ON	OFF
	16#04	(16#FF04) Flash Error	ON	OFF
	16#05	(16#FF05) Analog IC Error	ON	OFF
	16#06	(16#FF06) Analog Power Error	ON	OFF
16#A00**n**[*1]	16#01	(16#FF11) CH (**n**-1) Upper Limit Alarm	Every 1 second blinking	ON
	16#02	(16#FF12) CH (**n**-1) Lower Limit Alarm	Every 1 second blinking	ON
	16#03	(16#FF13) CH(**n**-1) Over Hardware range	Every 0.2 seconds blinking	OFF
	16#04	(16#FF14) CH(**n**-1) Under Hardware range	Every 0.2 seconds blinking	OFF
	16#05	(16#FF15) CH(**n**-1) Wire Break	Every 0.2 seconds blinking	OFF

[*1]: The errors are specified as warnings to ensure that the AX-5 Series CPU can still run even when the warnings are triggered by its modules. If you need the CPU STOP running immediately when the errors occur, you need to set them as errors in Module Hardware Configuration.

A.5.3 Error Codes and LED Indicators for Positioning Module (Pulse Unit)

Index	Subindex	Error Code	Description	Error LED
16#A001	16#01	16#FF81	Channel Hardware Alarm	2 seconds
16#A000	16#01	16#FF01	Unit Power Error	ON
		16#FF02	EtherCAT Connection Lost	
		16#FF03	ESC or EEPROM Error	
		16#FF04	Flash Error	
16#603F 16#683F	16#00	16#6320	PARAMETER_ERROR	Every 0.2 seconds blinking
		16#7120	MOTOR_ERROR	
		16#7500	COMMUNICATION	
		16#FF20	OverMaxFreq	
		16#FF21	LSP Occur	
		16#FF22	LSN Occur	
		16#FF23	IMSTOPOccur	
		16#FF24	ServoOnAtFactoryMode	

A.5.4 Error Codes and LED Indicators for Positioning Module (High-speed Counter)

Index	Subindex	Error Code	Description	Error LED
16#A001	16#00	16#FF81	Channel Hardware Alarm	Every 2 seconds blinking
16#A000	16#00	16#FF01	Unit Power Error	ON
		16#FF02	EtherCAT Connection Lost	
		16#FF03	ESC or EEPROM Error	
		16#FF04	Flash Error	
16#200A	16#01	16#FF02	EtherCAT Connection Lost	
16#200A 16#201A	16#02	16#FF20	Counter is NOT enabled before using this function.	Every 0.2 seconds blinking
		16#FF23	DI Function did NOT set to CLR0 or CLR1 before using the function of CounterExternalReset.	
	16#03	16#FF21	DI Function did NOT set to Latch0 before using DI as “Latch0”.	
		16#FF22	DI Function did NOT set to Latch1 before using DI as “Latch1”.	

A.5.5 Error Codes and LED Indicators for Load Cell (HC) Module

Index	Subindex	Description	LED	
			Error LED	Channel LED
16#A000	16#01	(16#FF01) Unit Power Error	ON	OFF
	16#02	(16#FF02) EtherCAT Disconnection		
	16#03	(16#FF03) ESC or EEPROM Error		
	16#04	(16#FF04) Flash Error		
	16#05	(16#FF05) Analog IC Error		
	16#06	(16#FF06) Analog Power Error		
16#6100*1	16#02	(16#FF11) Input Voltage Error	Every 0.2 seconds blinking	OFF
	16#03	(16#FF12) Weight Limit Exceeded	Every 1 second blinking	ON
	16#04	(16#FF13) Calibration Error		

*1: You can also find this information under the tab of EtherCAT I/O Mapping on DIADesigner-AX.

MEMO

Smarter. Greener. Together.

Industrial Automation Headquarters

Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan District,
Taoyuan City 330477, 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.
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.
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: +66-2709-2827

Delta Electronics (Australia) Pty Ltd.
Unit 2, Building A, 18-24 Ricketts Road,
Mount Waverley, Victoria 3149 Australia
Mail: IA.au@deltaww.com
TEL: +61-1300-335-823 / +61-3-9543-3720

Americas

Delta Electronics (Americas) Ltd.
5101 Davis Drive, Research Triangle Park, NC 27709, U.S.A.
TEL: +1-919-767-3813

Delta Electronics Brazil Ltd.
Estrada Velha Rio-São Paulo, 5300 Eugênio de
Melo - São José dos Campos CEP: 12247-004 - SP - Brazil
TEL: +55-12-3932-2300 / FAX: +55-12-3932-237

Delta Electronics International Mexico S.A. de C.V.
Gustavo Baz No. 309 Edificio E PB 103
Colonia La Loma, CP 54060
Tlalnepantla, Estado de México
TEL: +52-55-3603-9200

EMEA

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

Delta Electronics (Netherlands) B.V.
Automotive Campus 260, 5708 JZ Helmond, The Netherlands
Mail: Sales.IA.Benelux@deltaww.com
TEL: +31(0)40 800 3900

Delta Electronics (Netherlands) B.V.
Coesterweg 45, D-59494 Soest, Germany
Mail: Sales.IA.DACH@deltaww.com
TEL: +49 2921 987 238

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

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

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

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

Eltek Dubai (Eltek MEA DMCC)
OFFICE 2504, 25th Floor, Saba Tower 1,
Jumeirah Lakes Towers, Dubai, UAE
Mail: Sales.IA.MEA@deltaww.com
TEL: +971(0)4 2690148

AX-5-0379420-01
2024/05/31