Smarter. Greener. Together.

Industrial Automation Headquarters

Taiwan: Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan District,
Taoyuan City 33068, Taiwan
TEL: +886-3-362-6301 / FAX: +886-3-371-6301

Asia

China: 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

Japan: 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

Korea: 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

Singapore: Delta Energy Systems (Singapore) Pte Ltd.
4 Kaki Bukit Avenue 1, #05-04, Singapore 417939
TEL: +65-6747-5155 / FAX: +65-6744-9228

India: Delta Electronics (India) Pvt. Ltd.
Plot No.43, Sector 35, HSIIDC Gurgaon,
PIN 122001, Haryana, India
TEL: +91-124-4874900 / FAX: +91-124-4874945

Thailand: 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

Australia: Delta Electronics (Australia) Pty Ltd.
Unit 20-21/45 Normanby Rd., Notting Hill Vic 3168, Australia
TEL: +61-3-9543-3720

Americas

USA: Delta Electronics (Americas) Ltd.
5101 Davis Drive, Research Triangle Park, NC 27709, U.S.A.
TEL: +1-919-767-3813 / FAX: +1-919-767-3969

Brazil: Delta Electronics Brazil
Rua Itapeva, 26 - 3°, andar Edificio Itapeva,
One - Bela Vista 01332-000 - São Paulo - SP - Brazil
TEL: +55-12-3932-2300 / FAX: +55-12-3932-237

Mexico: 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

EMEA Headquarters: Delta Electronics (Netherlands) B.V.
Sales: Sales.IA.EMEA@deltaww.com
Marketing: Marketing.IA.EMEA@deltaww.com
Technical Support: iatechnicalsupport@deltaww.com
Customer Support: Customer-Support@deltaww.com
Service: Service.IA.emea@deltaww.com
TEL: +31(0)40 800 3900

BENELUX: Delta Electronics (Netherlands) B.V.
Automotive Campus 260, 5708 JZ Helmond, The Netherlands
Mail: Sales.IA.Benelux@deltaww.com
TEL: +31(0)40 800 3900

DACH: Delta Electronics (Netherlands) B.V.
Coesterweg 45, D-59494 Soest, Germany
Mail: Sales.IA.DACH@deltaww.com
TEL: +49(0)2921 987 0

France: Delta Electronics (France) S.A.
ZI du bois Challand 2, 15 rue des Pyrénées,
Lisses, 91090 Evry Cedex, France
Mail: Sales.IA.FR@deltaww.com
TEL: +33(0)1 69 77 82 60

Iberia: Delta Electronics Solutions (Spain) S.L.U
Ctra. De Villaverde a Vallecas, 265 1º Dcha Ed.
Hormigueras – P.I. de Vallecas 28031 Madrid
TEL: +34(0)91 223 74 20

Carrer Llacuna 166, 08018 Barcelona, Spain
Mail: Sales.IA.Iberia@deltaww.com

Italy: Delta Electronics (Italy) S.r.l.
Via Meda 2–22060 Novedrate(CO)
Piazza Grazioli 18 00186 Roma Italy
Mail: Sales.IA.Italy@deltaww.com
TEL: +39 039 8900365

Russia: Delta Energy System LLC
Vereyskaya Plaza II, office 112 Vereyskaya str.
17 121357 Moscow Russia
Mail: Sales.IA.RU@deltaww.com
TEL: +7 495 644 3240

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

MEA: 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-3-0339320-03
2021/10/08

AX-3 Series Quick Start

AX-3 **Series Quick Start**

www.deltaww.com

Smarter. Greener. Together.

AX-3 Series Quick Start

Revision History

Version	Revision	Date
1st	The first version was published.	2020/10/30
2nd	1. Chapter 2: Updated the system requirement table in section 2.1.2. 2. Chapter 3: Added notes in section 3.2.2.1, and updated software images in 3.2.3.2. Updated ASDA-A2-M signal illustration in section 3.3.1. Added notes and updated software images in section 3.3.2.1. Updated the procedures of programming with newly-created POUs and software images in section 3.3.2.4. Updated software images in 3.3.3. Added notes in section 3.5.2.1. Updated software images in sections 3.5.2.1 and 3.5.2.2.	2021/04/26
3rd	1.Chapter 2: Change the model type to AX-324N in the example and update the related screen of the software.	2021/10/8

AX-3 Series Quick Start
Table of Contents

Chapter 1 Introduction

Table of Contents

1.1 Related Manuals

The related manuals for AX-3 Series programmable logic controllers are listed below.

- AX-3 Series Quick Start (this manual)
  This quick start helps you create and use the system in a short time. Besides presenting you with basic system framework, this quick start uses example to demonstrate how to design, write programs, use variables as well as function blocks (FB) and download the PLC program to the PLC. Refer to Appendix A Troubleshooting of AX-3 Series Operation Manual, if any error occurs.
- AX-3 Series Operation Manual
  This manual introduces CPU functions, devices, module tables, electrical specifications, appearances and dimension, basic concept of motion control, basic configurations, troubleshooting, and so forth.
- AS Series Hardware Manual
  This manual introduces electrical specifications, wirings of CPU modules and modules, appearances, dimensions, and so forth.
- AS Series Module Manual
  This manual introduces special I/O modules such as network modules, analog I/O modules, temperature measurement modules, and so forth.
- AX Series Motion Controller Manual
  This introduces single-axis and multi-axes instructions for programming the AX Series Motion Controllers.
- AX Series Standard Instructions Manual
  This introduces standard instructions for programming the AX Series Controllers.
- DIADesigner-AX User Manual
  This manual introduces the use of the software, programming languages, including Ladder Diagram (LD), Sequential Function Chart (SFC), Structured Text (ST), and Function Block Diagram (FBD), as well as Program Organization Unit (POU), tasks and editing techniques for motion control programs.

Graphics used in the manual

Graph	Significance
	Click the left mouse button
	Click the right mouse button
	Double-click the left mouse button
	Press and hold the left mouse button, and then move the mouse without releasing the button
	Use a keyboard
	Operating sequence. This is used when the operating sequence is mentioned. For example,  and ② 
	Number used with a picture

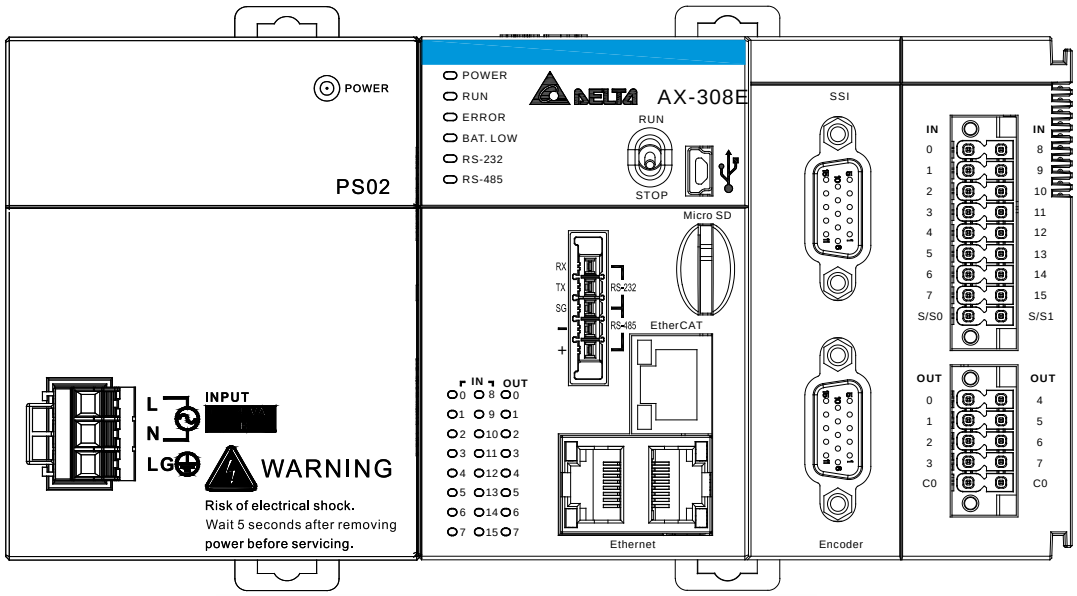
1.2 System Framework

The AX-3 Series PLC is a small-to-medium programmable logic control system. Execution speed and memory capacity are increased. Complete program development using function blocks is also supported. In order to meet more advanced application requirements, the AX-3 programmable logic controllers provide more flexible system extension frameworks. Under such system frameworks, you do not need to use multiple CPU modules to control the system because of too many I/O points or the equipment being too far away. System completeness has been retained, and you can be more efficient in project development.

Minimum framework requirements for the AX-3 system:

To create the AX-3 Series system, you need one CPU module and one power supply module to operate the CPU module.

Power supply module + AX-3 series PLC (AS-PS02 + AS308E-A)



The following are the limits for setting up a common framework for the AX-3 Series PLC system.

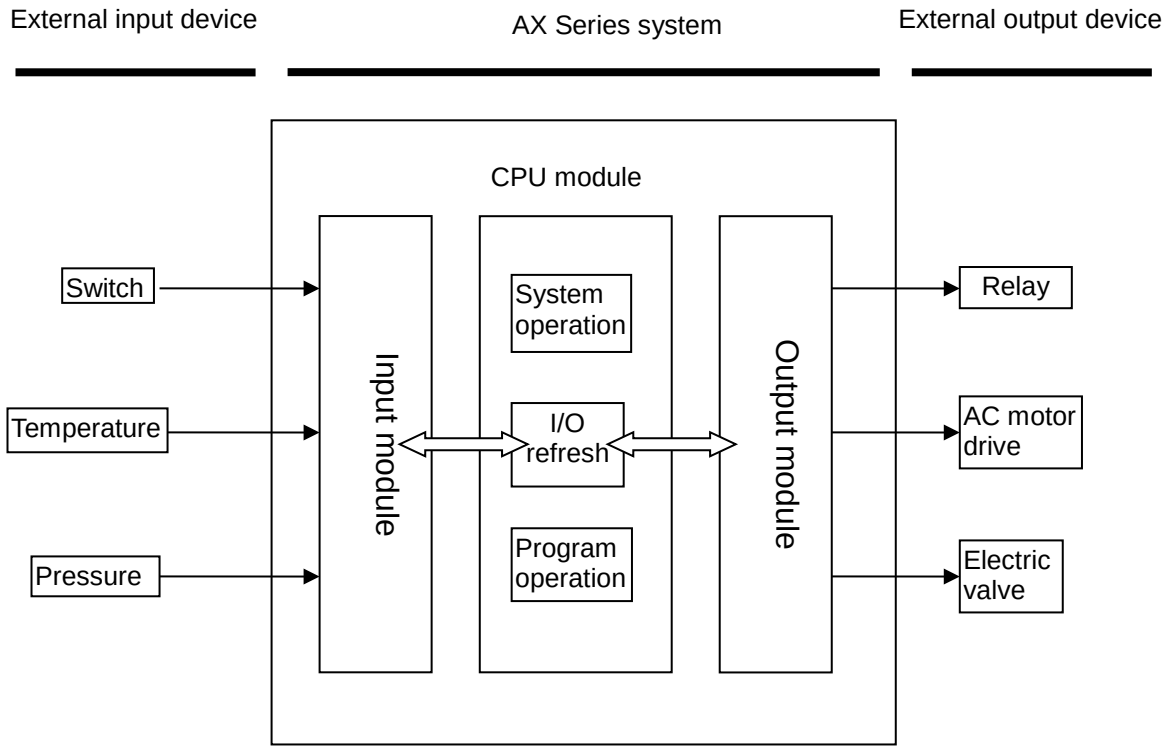
Exceeding any of the following limits triggers an error message:

Limit 1: Up to 32 extension modules can be connected to the PLC. The power module and CPU module are not included.

Limit 2: The maximum data capacity for inputs is 8192 bytes and for outputs is 8192 bytes.

1.3 CPU Operation

The CPU module is the heart of the AX Series system. It is responsible not only for the execution of the logic program, but also the data exchange and processing of communication data. The relation between the AX system and external devices is shown below.



Here shows a simplified system procedures related to initialization, diagnosis, communication, and program procedures related to external interrupts and timed interrupts. Refer to the other manuals for more information. Check the following sections for the task of CPU operation.

1.3.1 Task

You define one or more tasks for controlling and executing the program blocks (POUs) in the PLC. You 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.

1.3.2 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. The variable must be a Boolean value.

- **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.

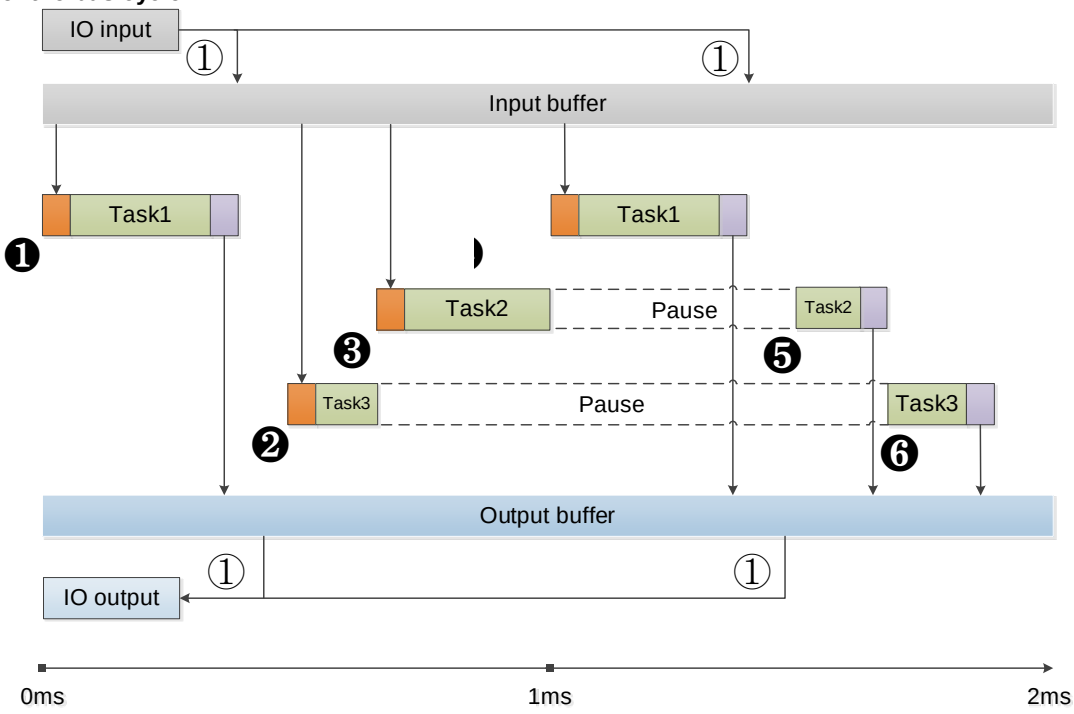
- **Status Task：**

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

1.3.3 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



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

Task 2: Priority = 3, Event Task

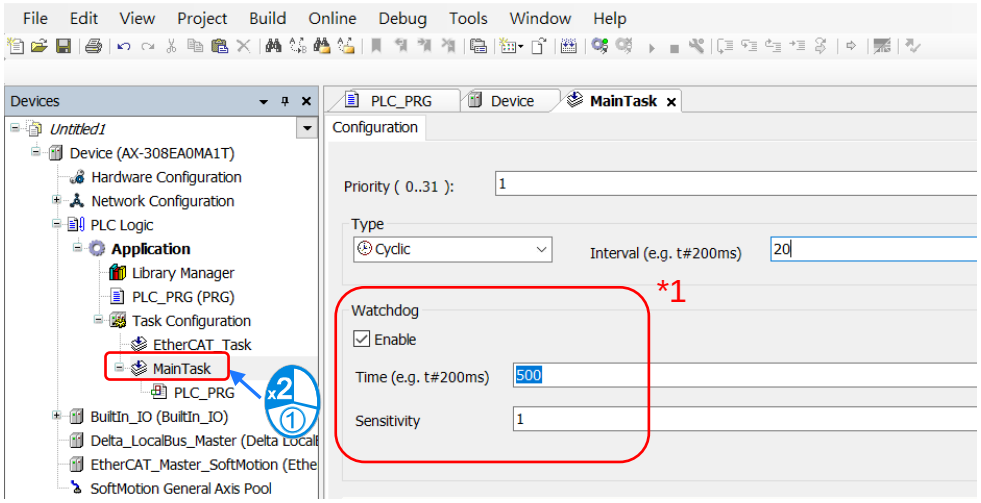
Task 3: Priority = 5, Freewheeling Task

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 ①: The messages are normally sent on the bus in this task. Other tasks copy only the I/O data from an internal buffer that is exchanged only with the physical hardware in the bus cycle task.

1.3.4 Watchdog

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



- Several consecutive timeouts:

  Sensitivity: 0, watchdog timeout = time *1

  Sensitivity: n, watchdog timeout = time *n

MEMO

2

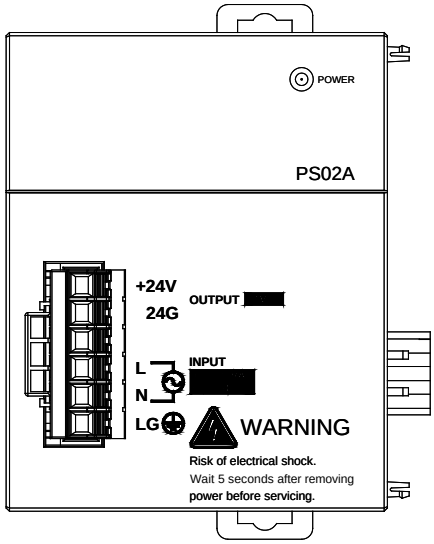
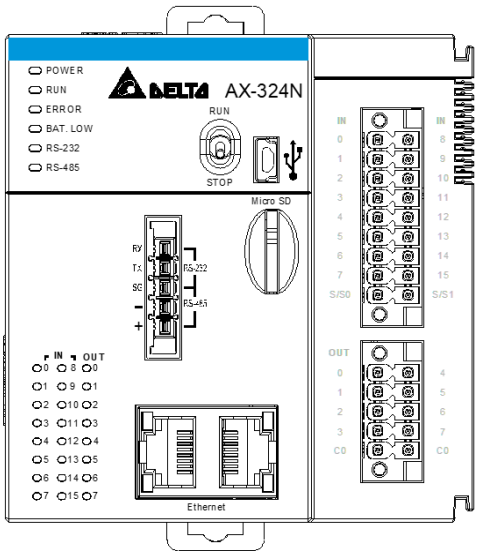
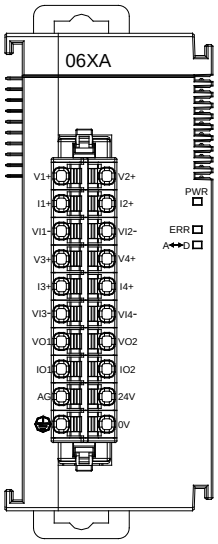
Chapter 2 Creating a Standard PLC Project

Table of Contents

2.1 Preparations

2.1.1 Hardware

The example uses the following hardware.

1. AS Series Power Module AS-PS02A x 1	2. AX-3 Series CPU Module (built-in with Ethernet communication port) AX-324N x 1
	
3. AS Series analog input/output module (4AI+2AO channels) AS06XA-A x 1	
	

2.1.2 Software

- **System requirements**

Project	System Requirement
Runtime System	DIADesigner-AX V1.00 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 more (recommend to use 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

2.1.3 Tools and Materials

The required tools and the materials are listed below.

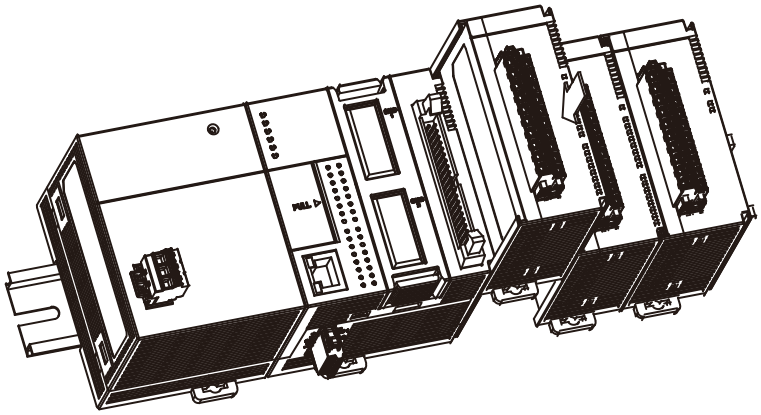
- A personal computer on which the software mentioned above is installed
- A 100-240 VAC and 50/60 Hz power supply socket
- A 24 VDC power supply
- A power cable
- A pin type terminal arranging kit
- At least 20 pin terminals
- A screwdriver
- A USB cable or a network cable.
- Prepare accessories such as a switch and a bulb to simulate the activity of external equipment if necessary.

2.2 Installation

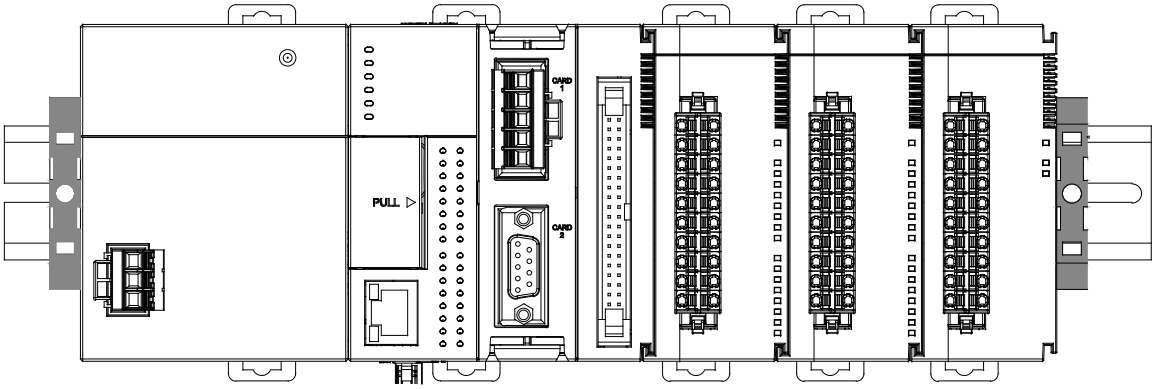
2.2.1 Installing Modules

The illustration below is AS300 CPU NOT AX-308E. It is for illustration purposes only.

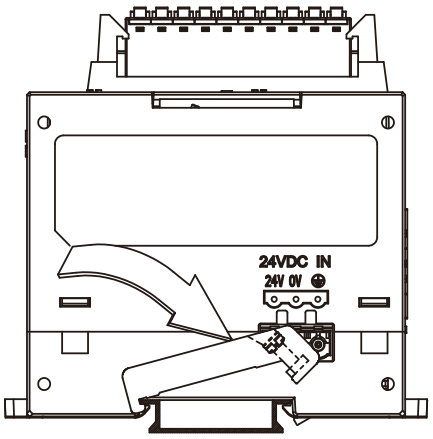
1. Install the PLC onto the power supply module, and then insert the module hooks into the DIN rail mounting slot.
2. Link the I/O modules on the right side of the PLC and make sure they are hooked together. Push the modules into the DIN rail until you hear a clic. (The following illustration demonstrates with AS300 CPU, not AX-324N.)



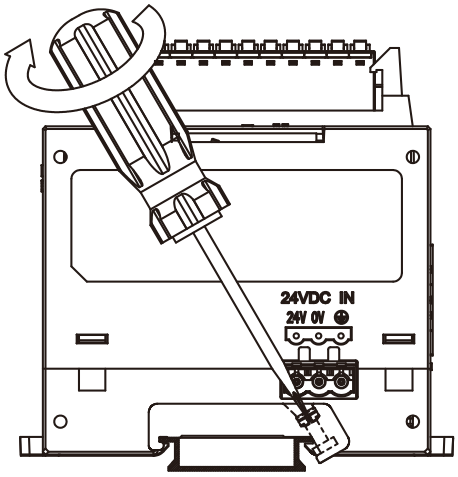
If there is a vibration source near the installation site, install anti-vibration baffles on the sides of the AX Series modules for better stabilization, such as the gray baffles show below.



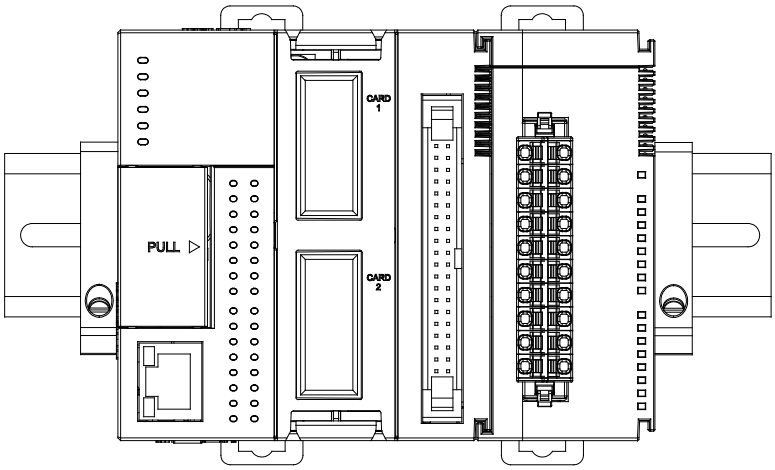
- **Install the baffles:**
  - Hook the baffle onto the DIN rail and press it down as the directional arrow shows below.



  - Use screws to secure the baffle.



  - The completed baffle installation is as shown below.

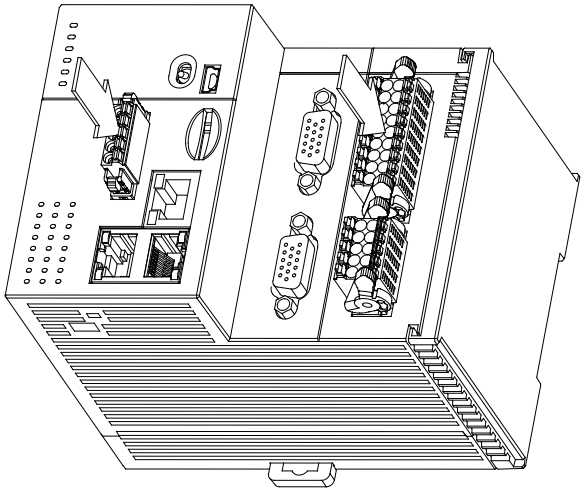


2.2.2 Installing / Removable Terminal Blocks

2.2.2.1 Installing/Removing the Removable Terminal Block on the CPU

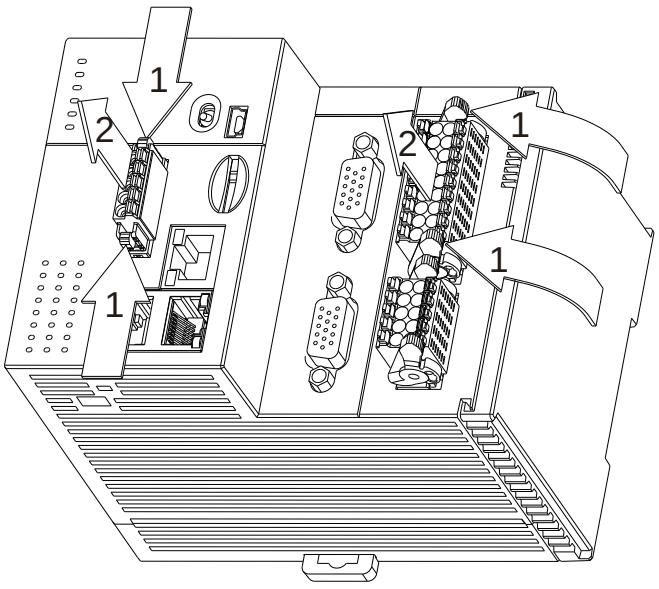
- **Installation**

1. Align the terminal block at the port, and press it into the CPU.



- **Removal**

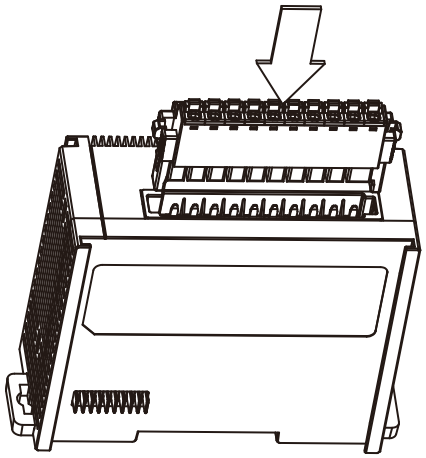
COM port (on the left): Push the clips inward as the arrow 1 shown to release the terminal block and then pull it up as the arrow 2 shown.
I/O port (on the right): Pull the clips backward as the arrow 1 to release the terminal block and then pull it up as the arrow 2 shown.



2.2.2.1 Installing/Removing the Removable Terminal Block on the Module

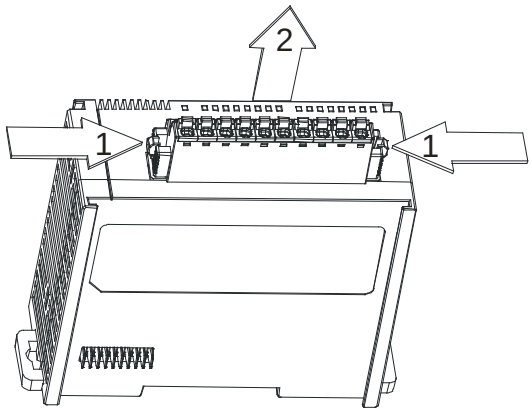
- **Installation**

Align the terminal block at the port, and press it into the module.



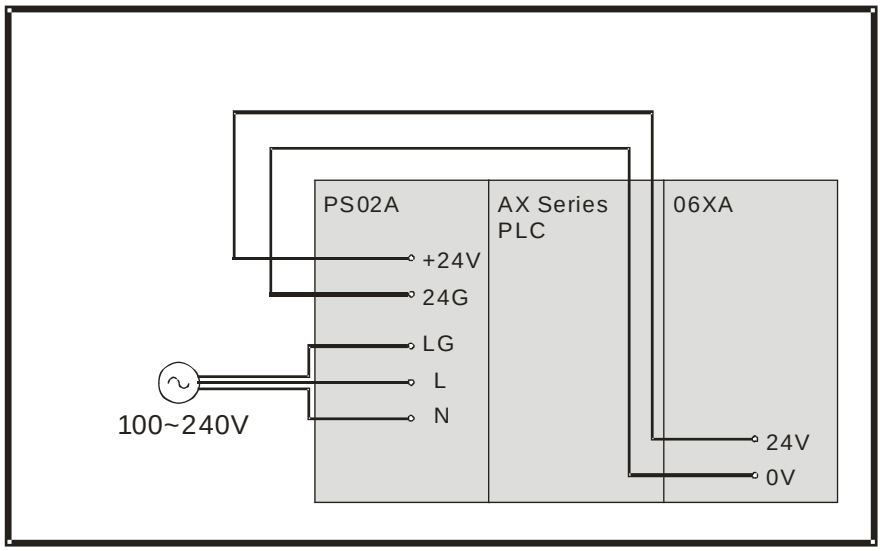
- **Removal**

Push the clips inward as the arrow 1 shown to release the terminal block and then pull it up as the arrow 2 shown.



2.3 Wiring

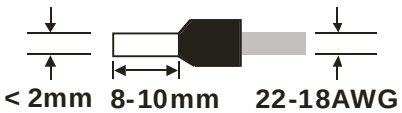
After installation is done, you can start wiring. For your own safety, remember to turn off the power supply before wiring. Connect the power module and analog module first. You can also connect the signal cables to the switches and bulbs for testing. The example framework is as shown below.



Find the detailed wiring descriptions in the following sections. You may also refer to the AS Series Module Manual for more information.

2.3.1 Wiring the Power Supply Module

- The alternating-current input voltage must be between 100–240 VAC. Connect the power supply to terminals L and N. **Please note if you connect the 110 VAC or the 220 VAC power supply to the input terminals +24V and 24G, it will damage the PLC.**
- If power is interrupted for less than 10 milliseconds, the PLC keeps running without being affected. If power is interrupted for more than 10 milliseconds, or if the voltage of the power supply decreases, the PLC stops running, and there is no output. When the power supply returns to normal, the PLC resumes. Notice that there are latched auxiliary relays and registers in the PLC when you write the program.
- For power supply cables (L and N) and ground cable (LG), use single-wire cables or twin-wire cables with a diameter of 22 AWG–18 AWG and with less than 2mm pin terminals. Only use copper conducting wires with a temperature of 60/75°C.



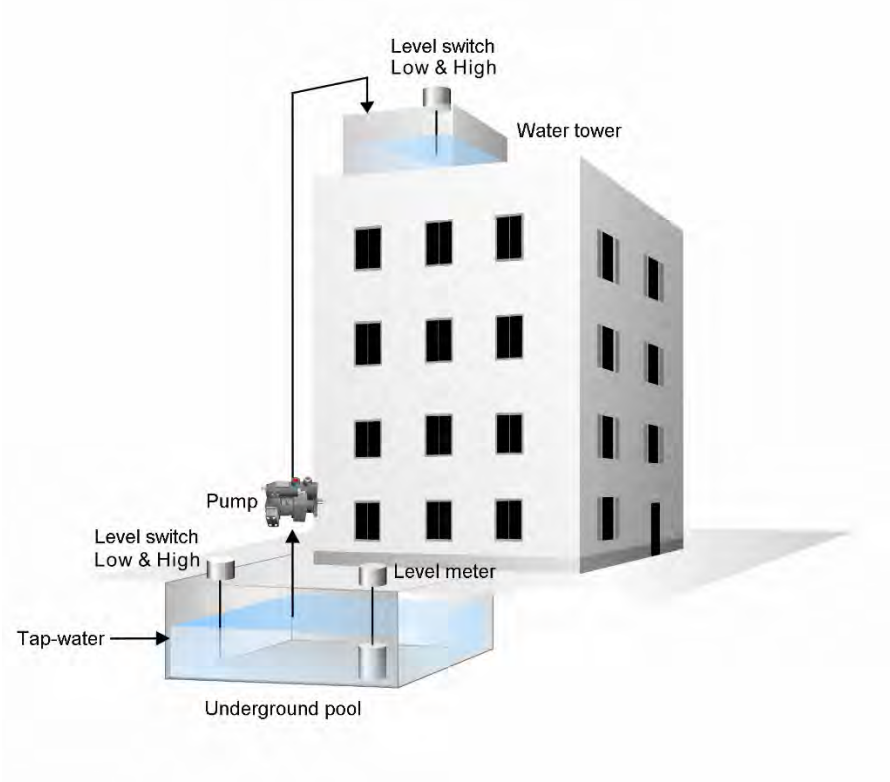
2.3.2 Supplying Power

After you complete the wiring, supply power to the CPU module. Make sure the CPU module is set to STOP before you supply power to the CPU module. The CPU module starts initializing once it is supplied with power. Since there is no program or hardware configuration in the CPU module, only the power indicator will be ON. This is normal.

2.4 Example

After you install the hardware, wire the modules, and supply power to the CPU module, you can then write the program. This manual provides users with an example program to show you how to create a new project and how to download the program to the CPU module step by step.

2.4.1 System Framework



- **Control action**

  This example is the basic design of the water supply of a multi-story building. Tap water is automatically supplied from the underground pool, and the water in the underground pool can be transported to the water tower on top of the building by the pump. The water is distributed by gravity to every story in the building, and the action of the pump is controlled by the level switches in the underground pool and in the water tower. To monitor the water supply, the level meter is installed in the underground pool. The water storage capacity of the underground pool is monitored at all times.

- **Devices connected to the I/O modules:**

1. A single-point level switch (contact A)

   The single-point level switch is installed in the underground pool, and the signal (Low) contact is connected to the digital input module.

2. A two-point level switch (contact A)

   The two-point level switch is installed in the water tower on top of the building, and the signal (Low & High) contact is connected to the digital input module.

3. A pump

   The pump is installed near the underground pool. The device to which the PLC actually connects is the control panel, not the pump. For common uses, there are three digital inputs and one digital output: remote control x 1 (DI) & Run x 1 (DI) & Trip x 1 (DI) & Start x 1 (DO) are connected to the digital input/output module.

4. A level meter

   The level meter is installed in the underground pool, and the signal (Level) contact is connected to the analog input module. 0–10V corresponds to 0–10M. 0 V means that the water is 0.0 meters deep, and 10 V means that the water is 10.0 meters deep.

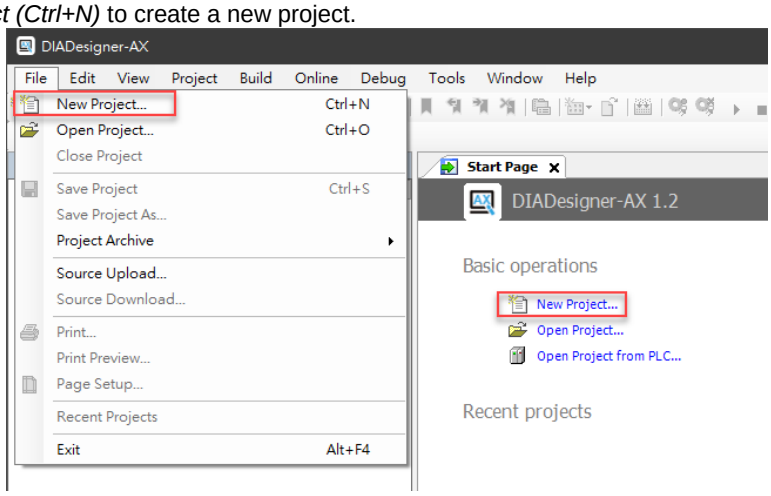
- **The control condition of the pump:**

1. Start: If the water level inside the water tower is low and that inside the underground pool is not low, the pump refills the water tower.
2. Stop: If the water level inside the water tower is high, or if the water level inside the underground pool is low, the pump stops running.

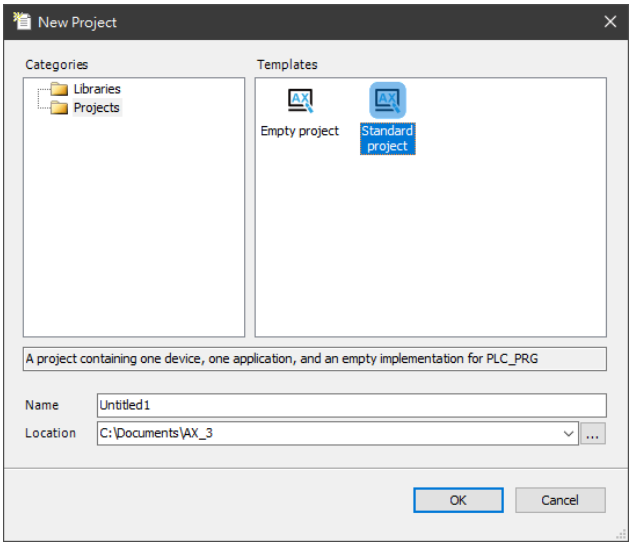
2.4.2 Creating a New Project

After you complete the hardware installation and know a thing or two about how a controller works, you can start creating a new project and writing programs.

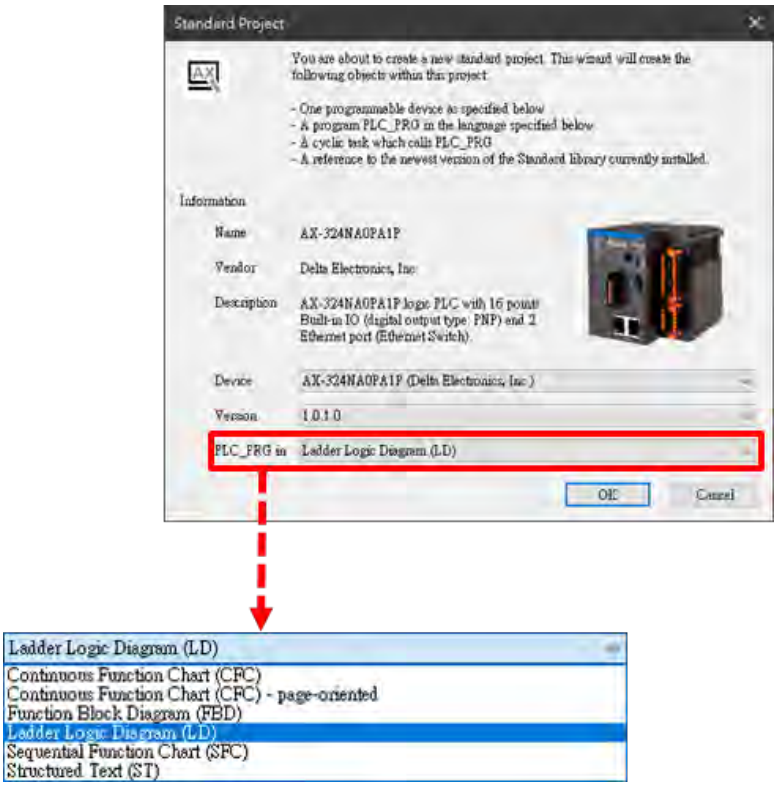
Double-click the DIADesigner-AX icon to open DIADesigner-AX. Click **New Project** on the Start Page or select *File > New Project (Ctrl+N)* to create a new project.



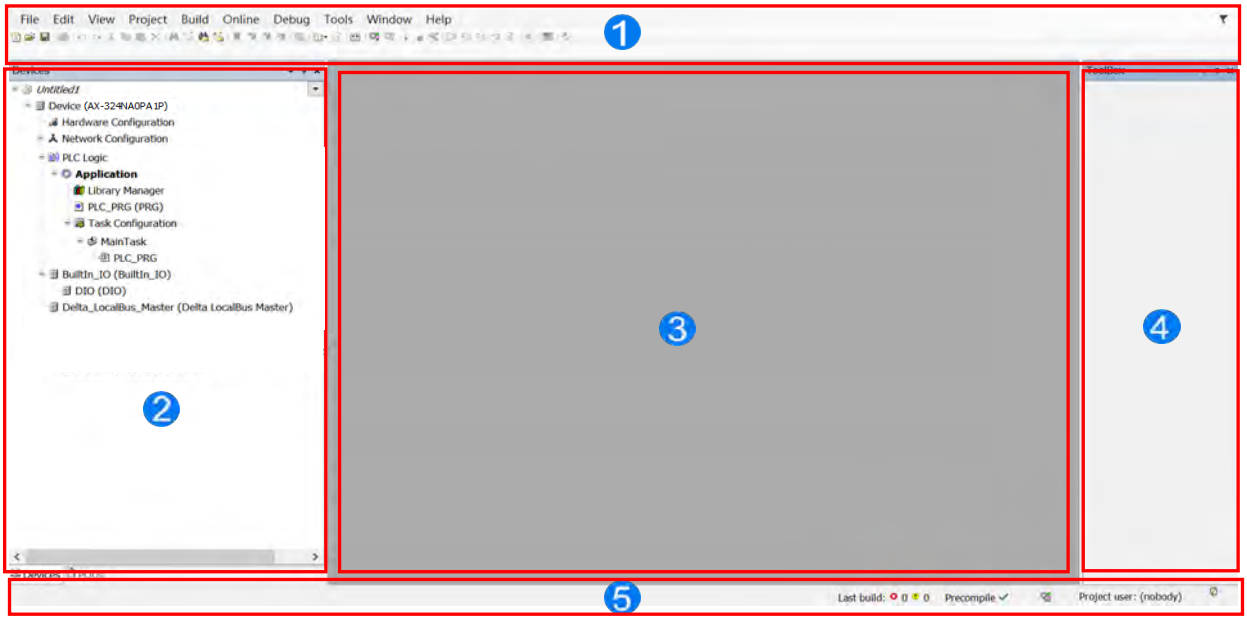
Next you will see a window with two sections, Categories and Templates. Click **Projects** in the Categories section and click **Standard project** in the Templates section. After that create a Name and specify a location for the project and then click **OK**.



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



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 tool bar, if nothing appears in the project management area.



The operation interface of the software:

❶ Function area: The main functions of the software are in this area. Many frequently used functions are placed on the toolbar, and other functions are placed on the menus.

❷ Project management area: The framework of the project is displayed in this area. You can see the relations among the objects in the tree structure, which increases project management efficiency.

❸ Work area: The editing work is in this area.

❹ Output area: Information resulting from the execution of the function is displayed in this area.

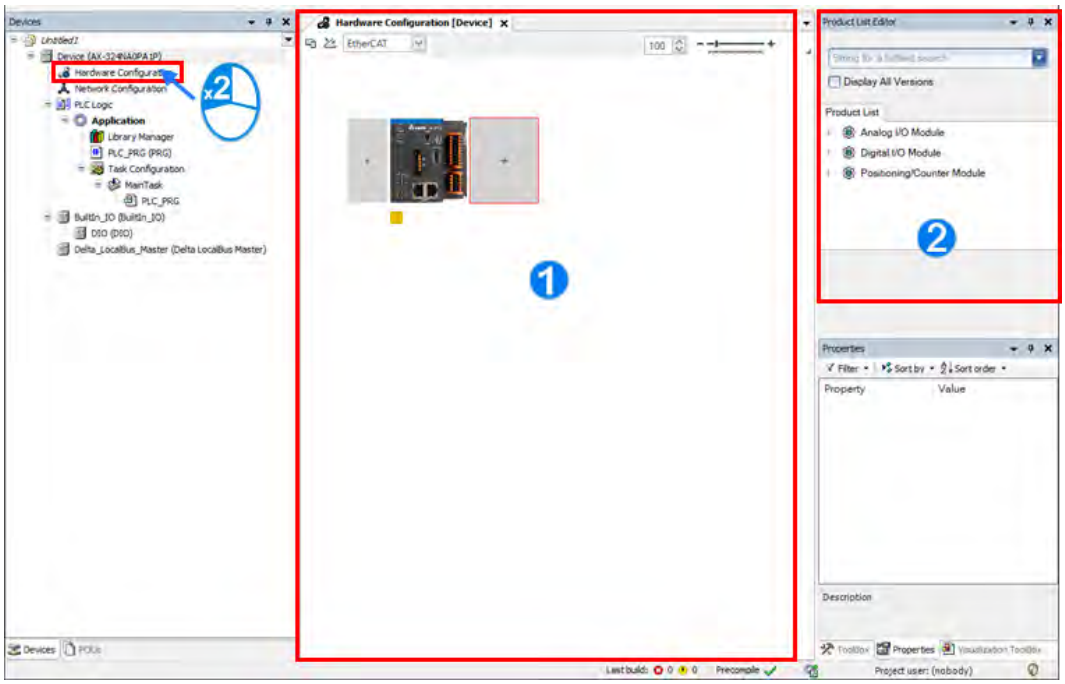
❺ Status area: The project and communication information are displayed in this area.

2.4.3 Hardware Configurations

After the project is created, you can configure the hardward. Suppose you set the following configurations.

- The digital input addresses of the CPU (16 inputs): %IX0.0 ~ %IX0.7 & %IX1.0 ~ %IX1.7
- The digital output addresses of the CPU (8 outputs): %QX0.0 ~ %QX0.7
- The analog input addresses of the 06XA (4 channels): %ID2 ~ %ID5
- The analog output addresses of the 06XA (2 channels): %QD1 ~ %QD2
- The low point of the water level inside the underground pool: %IX0.0
- The low point of the water level inside the water tower: %IX0.1
- The high point of the water level inside the water tower: %IX0.2
- Remote control of the pump: %IX0.3
- The RUN of the pump: %IX0.4
- The TRIP of the pump: %IX0.5
- The START of the pump: %QX0.0
- The value of the watrer level inside the underground pool: %ID2

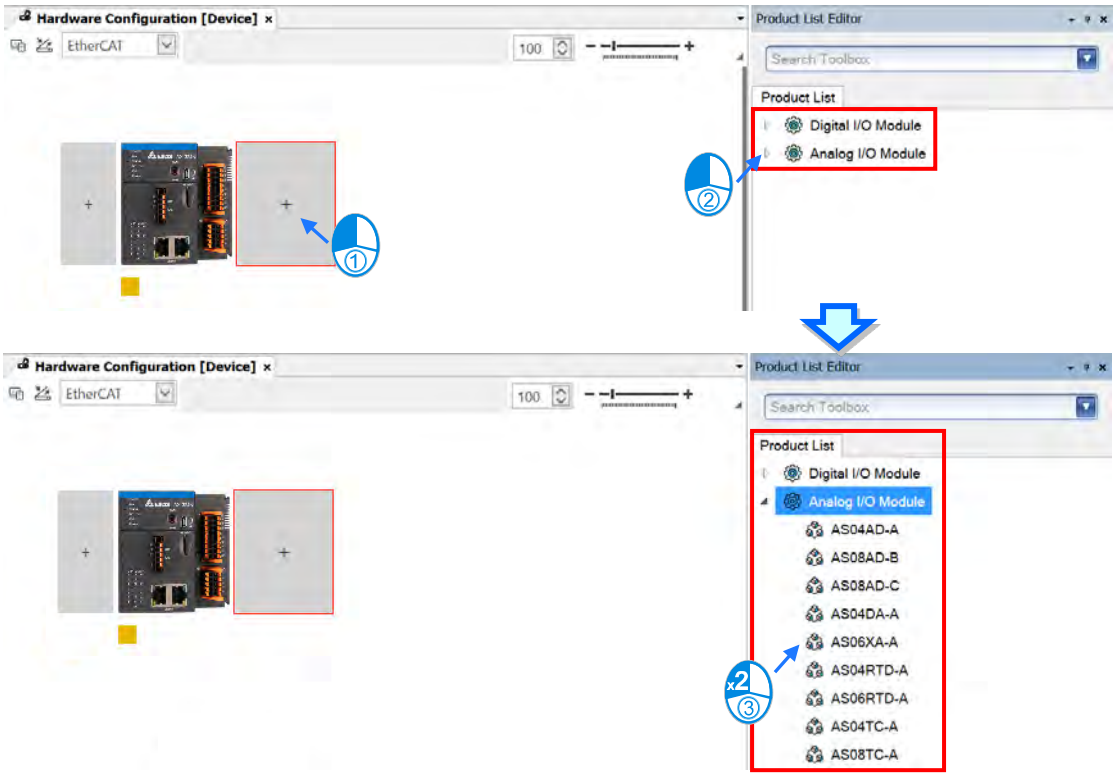
With the information above, you can configure the hardware now. Double-click Hardware Configuration on the Device section to open the Hardware Configurate (Device) window as the image shown below.



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

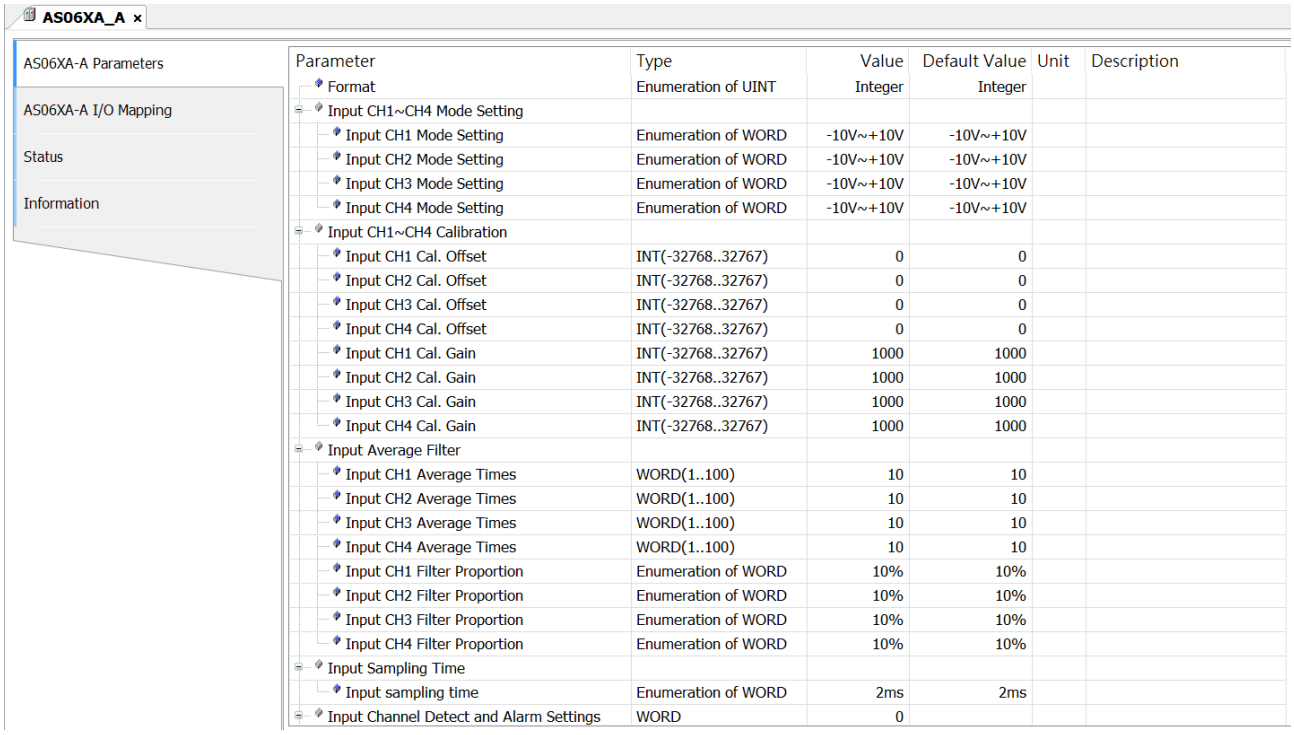
❷ Product List Editor: Here listed out all supported modules for the selected CPU.

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.



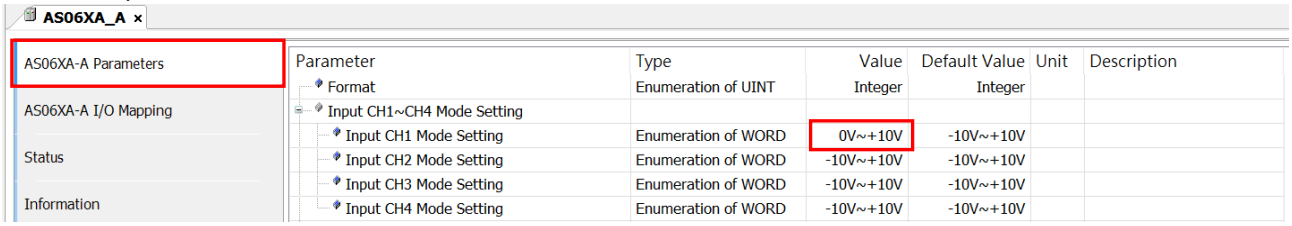
Double-click or drag and drop the extension module (06XA) that you'd like to add from the Product List. Newly added extension modules will apper on the right-side of the AX-3 Series PLC. And the device names will also show up on the left-side under Delta_LocalBus_Master.

In order to complete the basic setting, set the relation between the signals and the conversion values for the analog module. Double-click the module to open the **Parameter** window.



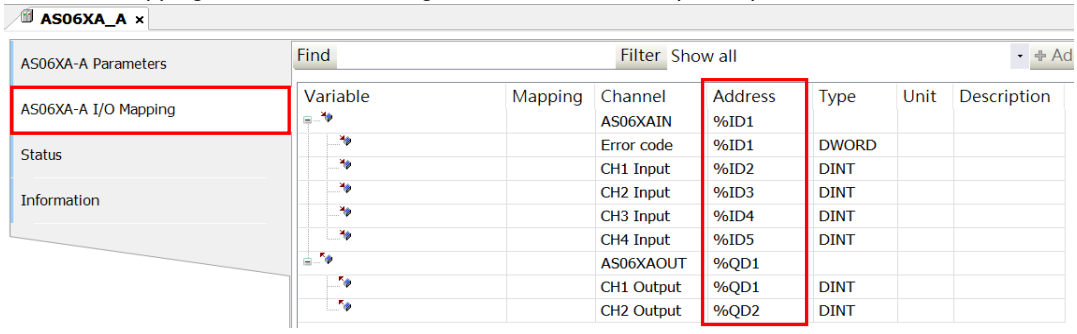
Parameter	Type	Value	Default Value	Unit	Description
Format	Enumeration of UINT	Integer	Integer		
Input CH1~CH4 Mode Setting					
Input CH1 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		
Input CH2 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		
Input CH3 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		
Input CH4 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		
Input CH1~CH4 Calibration					
Input CH1 Cal. Offset	INT(-32768..32767)	0	0		
Input CH2 Cal. Offset	INT(-32768..32767)	0	0		
Input CH3 Cal. Offset	INT(-32768..32767)	0	0		
Input CH4 Cal. Offset	INT(-32768..32767)	0	0		
Input CH1 Cal. Gain	INT(-32768..32767)	1000	1000		
Input CH2 Cal. Gain	INT(-32768..32767)	1000	1000		
Input CH3 Cal. Gain	INT(-32768..32767)	1000	1000		
Input CH4 Cal. Gain	INT(-32768..32767)	1000	1000		
Input Average Filter					
Input CH1 Average Times	WORD(1..100)	10	10		
Input CH2 Average Times	WORD(1..100)	10	10		
Input CH3 Average Times	WORD(1..100)	10	10		
Input CH4 Average Times	WORD(1..100)	10	10		
Input CH1 Filter Proportion	Enumeration of WORD	10%	10%		
Input CH2 Filter Proportion	Enumeration of WORD	10%	10%		
Input CH3 Filter Proportion	Enumeration of WORD	10%	10%		
Input CH4 Filter Proportion	Enumeration of WORD	10%	10%		
Input Sampling Time					
Input sampling time	Enumeration of WORD	2ms	2ms		
Input Channel Detect and Alarm Settings	WORD	0			

Use the drop-down list to select a suitable value.



Parameter	Type	Value	Default Value	Unit	Description
Format	Enumeration of UINT	Integer	Integer		
Input CH1~CH4 Mode Setting					
Input CH1 Mode Setting	Enumeration of WORD	0V~+10V	-10V~+10V		
Input CH2 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		
Input CH3 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		
Input CH4 Mode Setting	Enumeration of WORD	-10V~+10V	-10V~+10V		

Click AS06XA-A I/O Mapping to see the auto-assigned address of each input/output channels.



Variable	Mapping	Channel	Address	Type	Unit	Description
		AS06XAIN	%ID1			
		Error code	%ID1	DWORD		
		CH1 Input	%ID2	DINT		
		CH2 Input	%ID3	DINT		
		CH3 Input	%ID4	DINT		
		CH4 Input	%ID5	DINT		
		AS06XAOUT	%QD1			
		CH1 Output	%QD1	DINT		
		CH2 Output	%QD2	DINT		

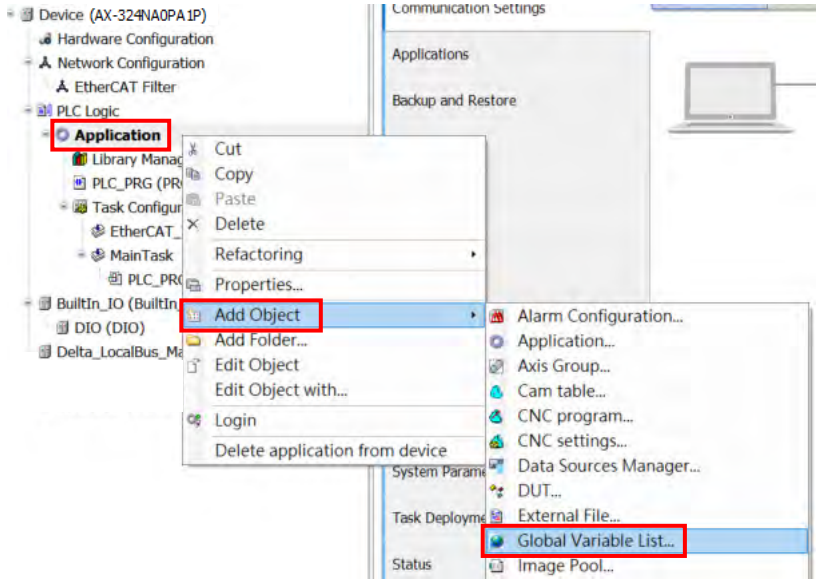
2.4.4 Creating Global Symbols

In order to make the program more readable, the I/O addresses are accompanied by global symbols. Use these symbols when you write the program. The global symbols are as follows.

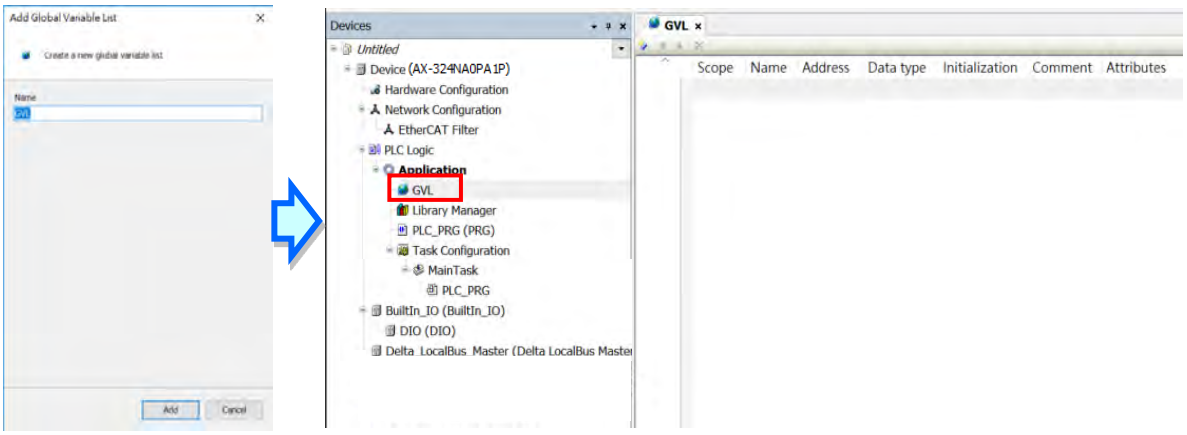
Global symbol table		
Bit (Hardware: I/O on the PLC)		
Identifier	**Address**	**Data type**
Tank_B1F_LSW	%IX0.0	BOOL
Tank_RF_LSW	%IX0.1	BOOL
Tank_RF_HSW	%IX0.2	BOOL
SPP01_Remote	%IX0.3	BOOL
SPP01_Run	%IX0.4	BOOL
SPP01_Trip	%IX0.5	BOOL
SPP01_Start	%QX0.0	BOOL
Tank_B1F_LT	%ID2	DINT
Bit (Software)		
SPP01_Auto		BOOL
SPP01_Man_SW		BOOL

With the information above, now we can create the global symbols.

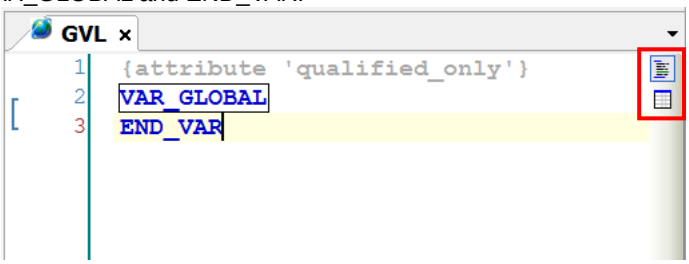
1. Right-click **Application** under the node of **PLC Logic** to open a context menu. Select **Add Object** to see another context menu. Select **Global Variable List** to open the Add Global Variable List window.



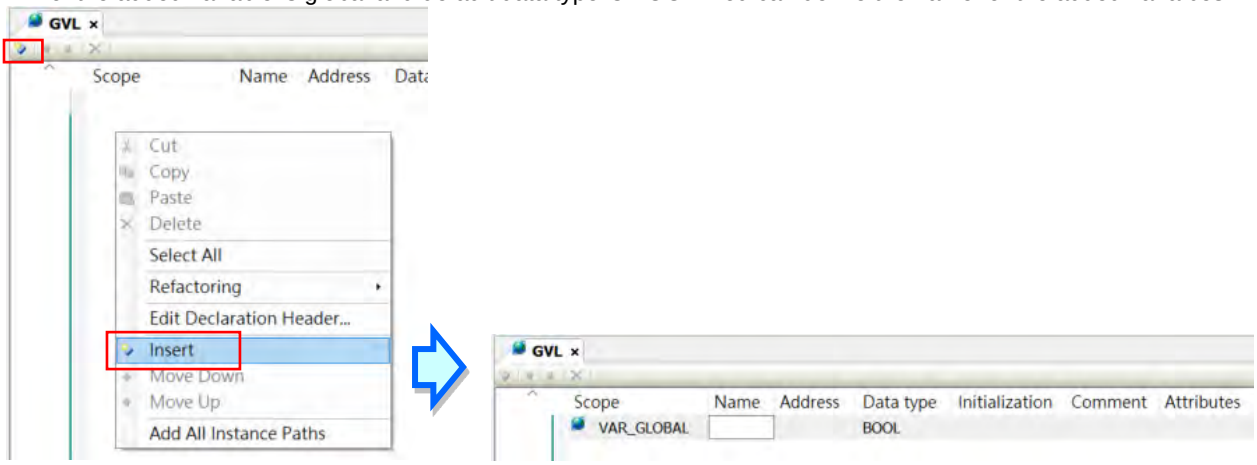
2. Click **Add** on the Add Global Variable List window to add a new global variable list. A default name “GVL” is shown automatically. You can define a new name for the new gloval variable list.



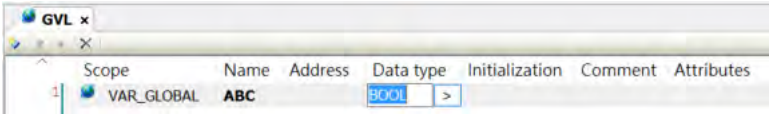
3. When shown in the tabular view, you can see the cursor below the column “Name”. You can click  for the textual view or click  to switch back to the tabular view. When shown in the textual view, you can find two varialbes, including VAR_GLOBAL and END_VAR.



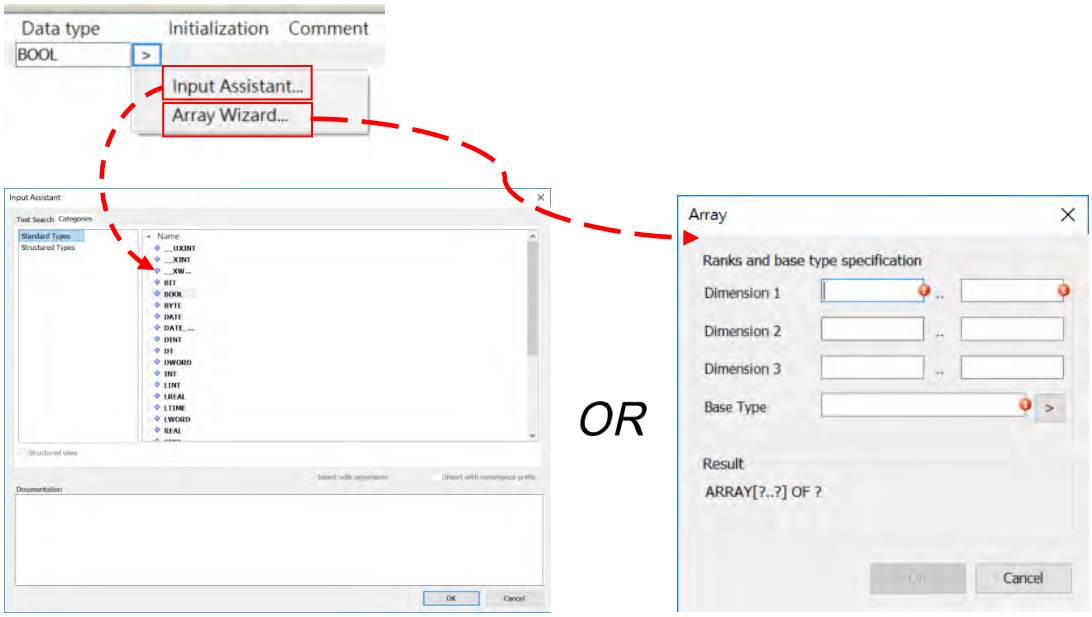
4. Right-click on the blank space of GVL window or click the  button to insert a new variable. The default scope of the added variable is global and default data type is BOOL. You can define the name for the added varialbes.



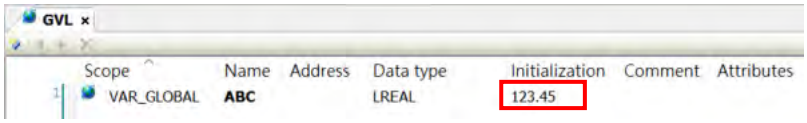
5. Double-click the value in the column of Data Type and the drop-down list button  appears.



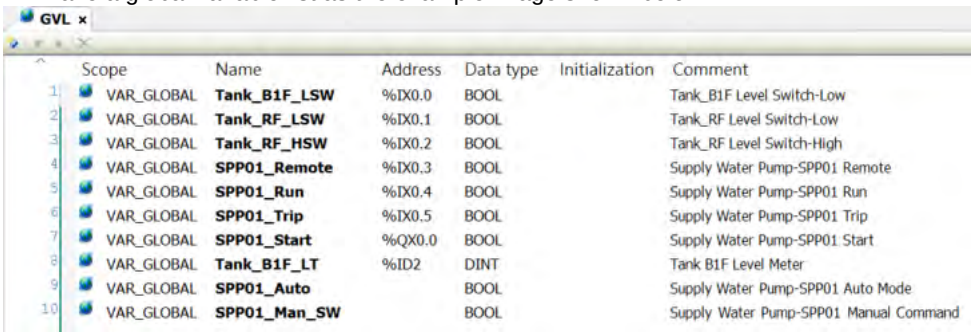
6. Click > to see two options, **Input Assistant** and **Array Wizard**, for configurations: Input Assistant for the data type of the varialbes; Array Wizard for the array configurations.



7. You can enter a value for initialization for the variable.



8. Follow the steps above to add more global variables and set the corresponding IO addresses for the variables. And then you will have a global variable list as the example image shown below.



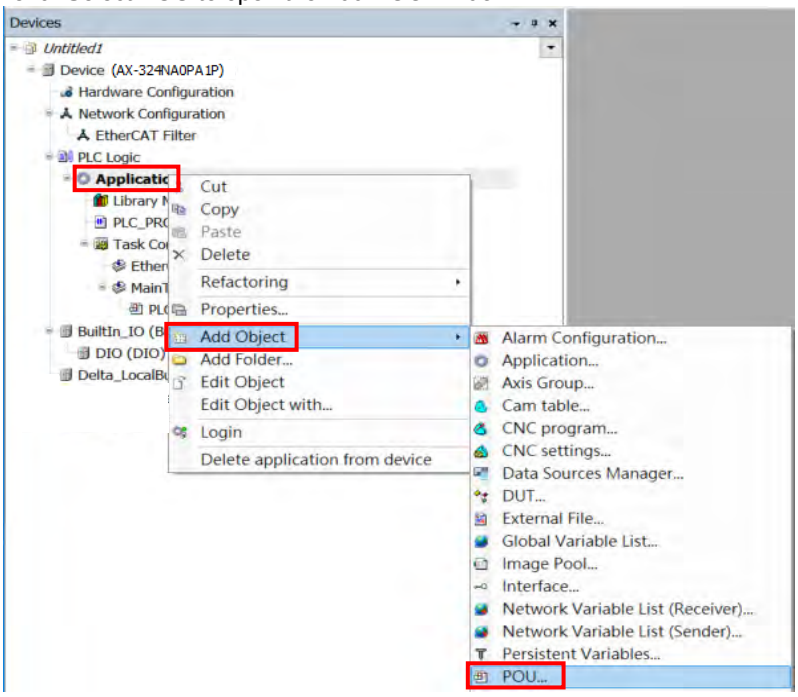
2.4.5 Creating Function Blocks

The procedure in this example is to create a function block first. You can create the main program first if you prefer. There is no relation between the function block and the main program, and they run separately. However, you should create a function first when the function is used repeatedly.

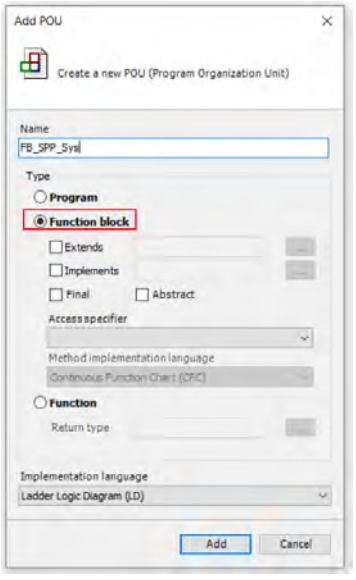
The control relation among the underground pool, the water tower, and the pump can be represented by function blocks. There are usually two water systems in a multi-story building. If you create the function blocks, you only need to change the variables of the input and output pins to complete the second water system.

With the information above, now we can create the function block.

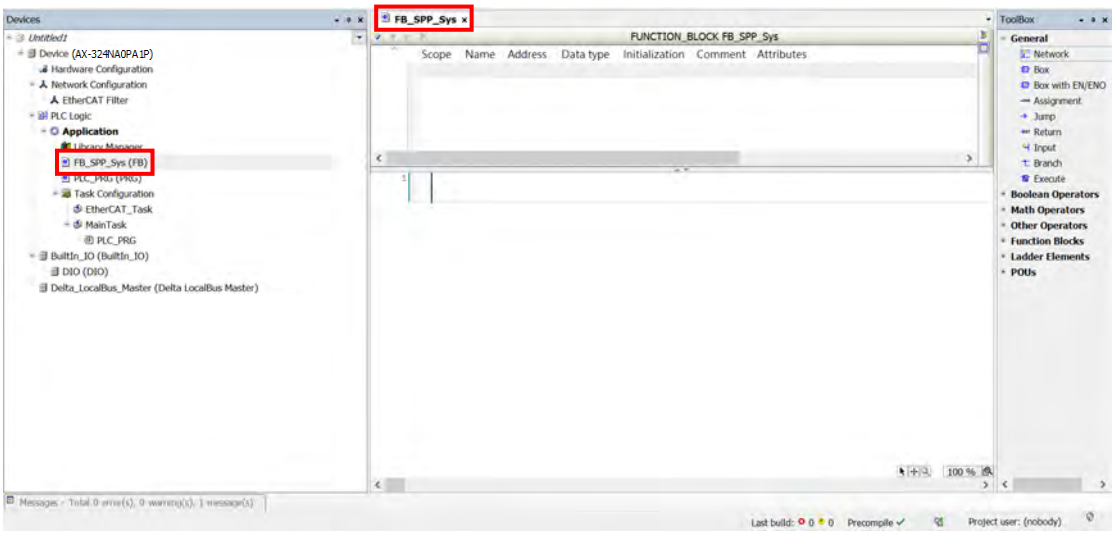
1. Right-click **Application** under the node of **PLC Logic** to open a context menu. Select **Add Object** to see another context menu. Select **POU** to open the Add POU window.



2. You can define a new name for the function block. Select **Function block** in the Type section and click **Add** to add a function block.



3. After that you can find the newly created FB under the node of **Application** on the tree view. And the FB editing window is on the right side.

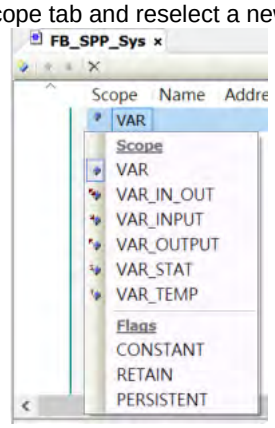


Create the local symbols. The way to create the local symbols is the same as to creat global symbols. You can use the global symbols in the internal program, but by doing so, the protability and convenience of using the function block are decreased. The local symbols and global symbols can bear the same name. But the local symbol has a higher priority.

The local symbols used in this example are as follows.

Declaration type	Identifier	Data type
VAR_INPUT	Tank_B_LSW	BOOL
VAR_INPUT	Tank_R_LSW	BOOL
VAR_INPUT	Tank_R_HSW	BOOL
VAR_INPUT	Pump_Remote	BOOL
VAR_INPUT	Pump_Run	BOOL
VAR_INPUT	Pump_Trip	BOOL
VAR_IN_OUT	Pump_Auto	BOOL
VAR_IN_OUT	Pump_Man_SW	BOOL
VAR_OUTPUT	Pump_Start	BOOL
VAR	Pump_Out	BOOL

You can double-click the variables below the Scope tab and reselect a new scope type.



VAR_INPUT

When you run the program, the value of the external variable is brought into the internal variable. If the value of the corresponding internal variable is changed, it is not transmitted to the external variable. VAR_INPUT is often used if the value of the external variable should not be modified. Most of the inputs in this example are digital inputs, and these should not be modified. The modifying the values of these variables affects the execution of the program or the use of the function block. In order to prevent the values of these variables from being modified in the program, use the declaration type VAR_INPUT.

VAR_IN_OUT

When you run the program, the value of the external variable is brought into the internal variable. After the program comes to an end, the value is transmitted to the external variable. VAR_IN_OUT is often used if the value of the variable should be modified. Generally speaking, Pump_Auto and Pump_Man_SW in this example are used in the SCADA system to set the control mode of the pump. It may seem that this type of variable, VAR_INPUT, meets the requirement. However, you need to switch the control mode of the pump from automatic mode to manual mode to stop the command from being output when the pump trips. Therefore, declare these two variables to the type of VAR_IN_OUT.

VAR_OUTPUT

When you run the program, the value of the external variable is not brought into the internal variable, but the previous stored value is used instead. After the program comes to an end, the value is transmitted to the external variable. Generally speaking, this type of variable appears in the output of the instruction. The variable Pump_Start used in this example belongs to this type of usage.

VAR

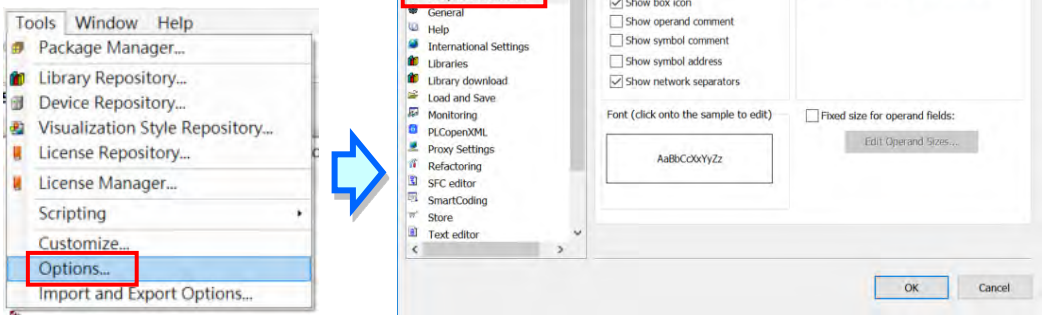
When the program is run, VAR is evaluated as an internal variable. Just like VAR_OUTPUT, the previous stored value is used instead. Generally speaking, with no transmission function, the variable is used as a register when it is used in the program. The variable Pump_Out used in this example belongs to this type of usage.

After the creation of function blocks is done, the table should look like this:

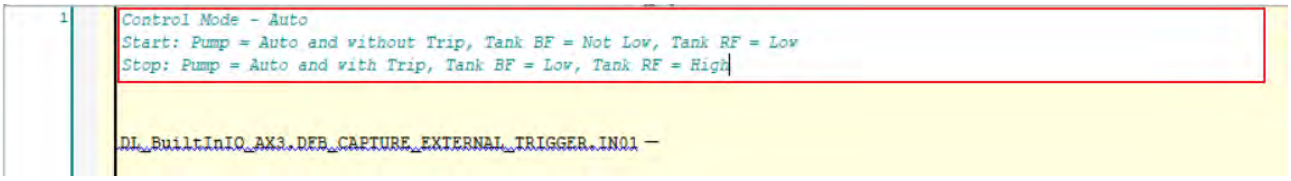
FUNCTION_BLOCK FB_SPP_Sys

	Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR_INPUT	**Tank_B_LSW**		BOOL		Tank BF Level Switch-Low	
2	VAR_INPUT	**Tank_R_LSW**		BOOL		Tank RF Level Switch-Low	
3	VAR_INPUT	**Tank_R_HSW**		BOOL		Tank RF Level Switch-High	
4	VAR_INPUT	**Pump_Remote**		BOOL		Pump-Remote	
5	VAR_INPUT	**Pump_Run**		BOOL		Pump-Run	
6	VAR_INPUT	**Pump_Trip**		BOOL		Pump-Trip	
7	VAR_IN_OUT	**Pump_Auto**		BOOL		Pump-Auto	
8	VAR_IN_OUT	**Pump_Man_SW**		BOOL		Pump-Manual Switch	
9	VAR_OUTPUT	**Pump_Start**		BOOL		Pump-Start	
10	VAR	**Pump_Out**		BOOL		Pump-Out	

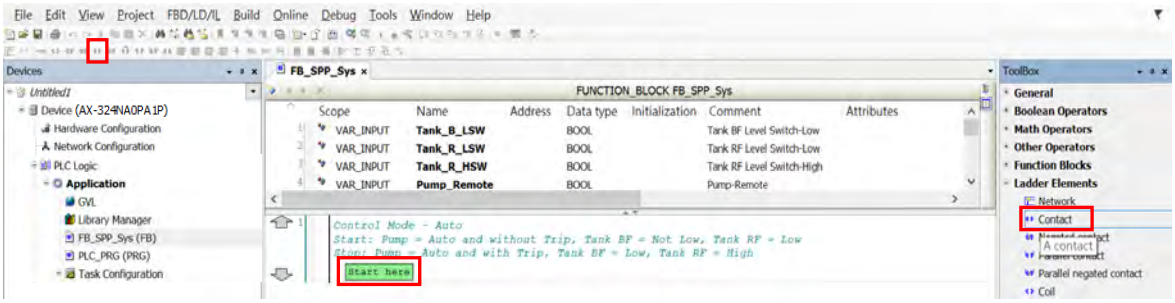
Now you can start editing the contents of the function blocks. If you need to add comments while editing the function block, you need to go to *Tools -> Options -> FBD, LD and IL editor* to select the option **Show network comment** and then click **OK** to confirm the setting.



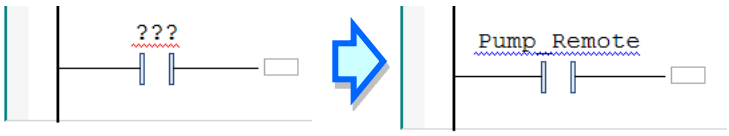
Double-click the upper space of the network section to add comments. Press the Enter key on your keyboard to go to the next line.



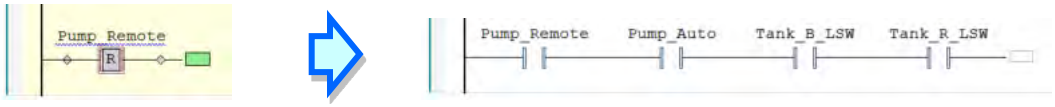
Now you can start programming. Here we use the Ladder Logic Diagram (LD) as the programming language in this example. You can either click where you'd like to add the element and then click the on the tool bar above to add. Or find Contact in the Ladder Elements of the ToolBox on the right side of the window. (ToolBox -> Ladder Elements -> Contact). Select the element and drag it to where you'd like to add the coil. Once you see the green box appears, drop it there.



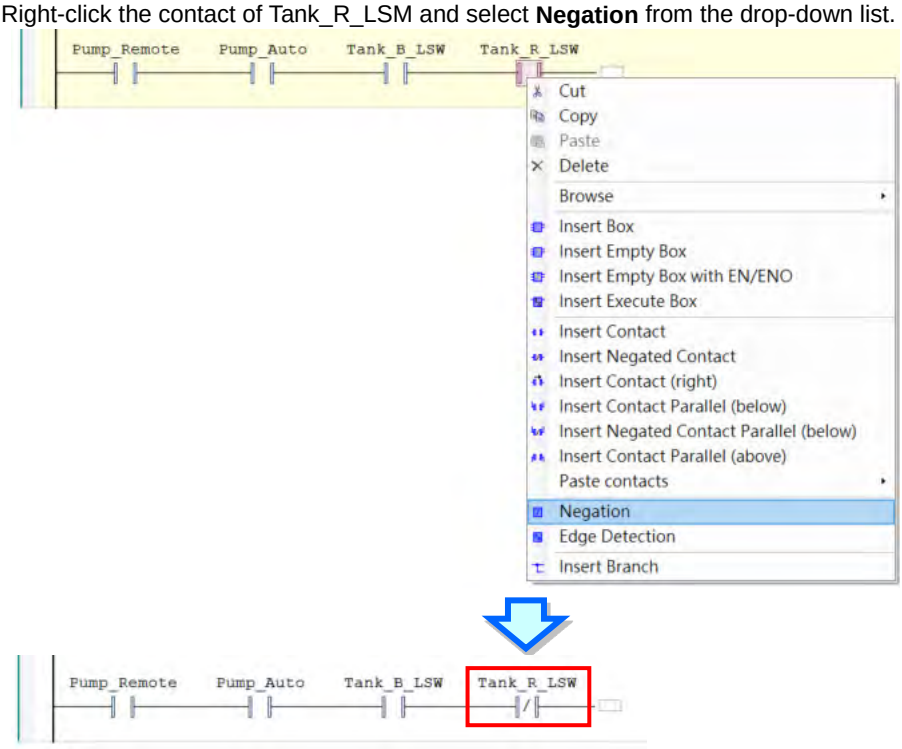
Click the ??? section to type a new name, for example, Pump_Remote and then press the Enter key.



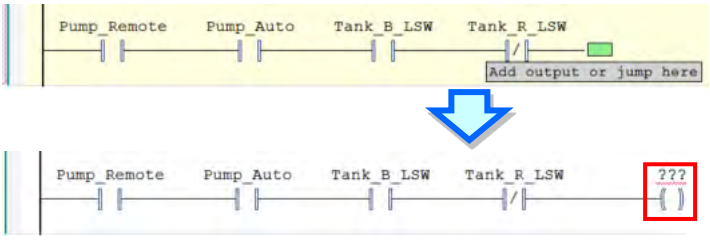
You can use the shortcut **Ctrl+K** to quickly add several in.



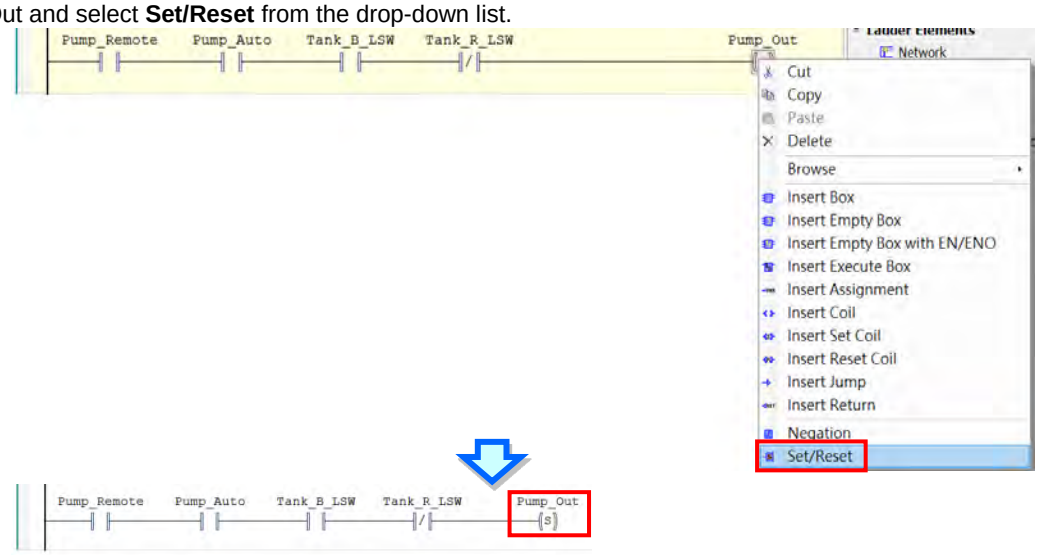
If you need the condition of with water in the underground pool and no water in the water tower, the state of Tank_R_LSM should be False. Right-click the contact of Tank_R_LSM and select **Negation** from the drop-down list.



You can add a coil by either clicking where you'd like to add a coil and then clicking on the tool bar above to add or finding in the Ladder Elements of the ToolBox on the right side of the window (ToolBox -> Ladder Elements ->). Select the element and drag it to where you'd like to add the coil. Once you see the green box appears, drop it there.

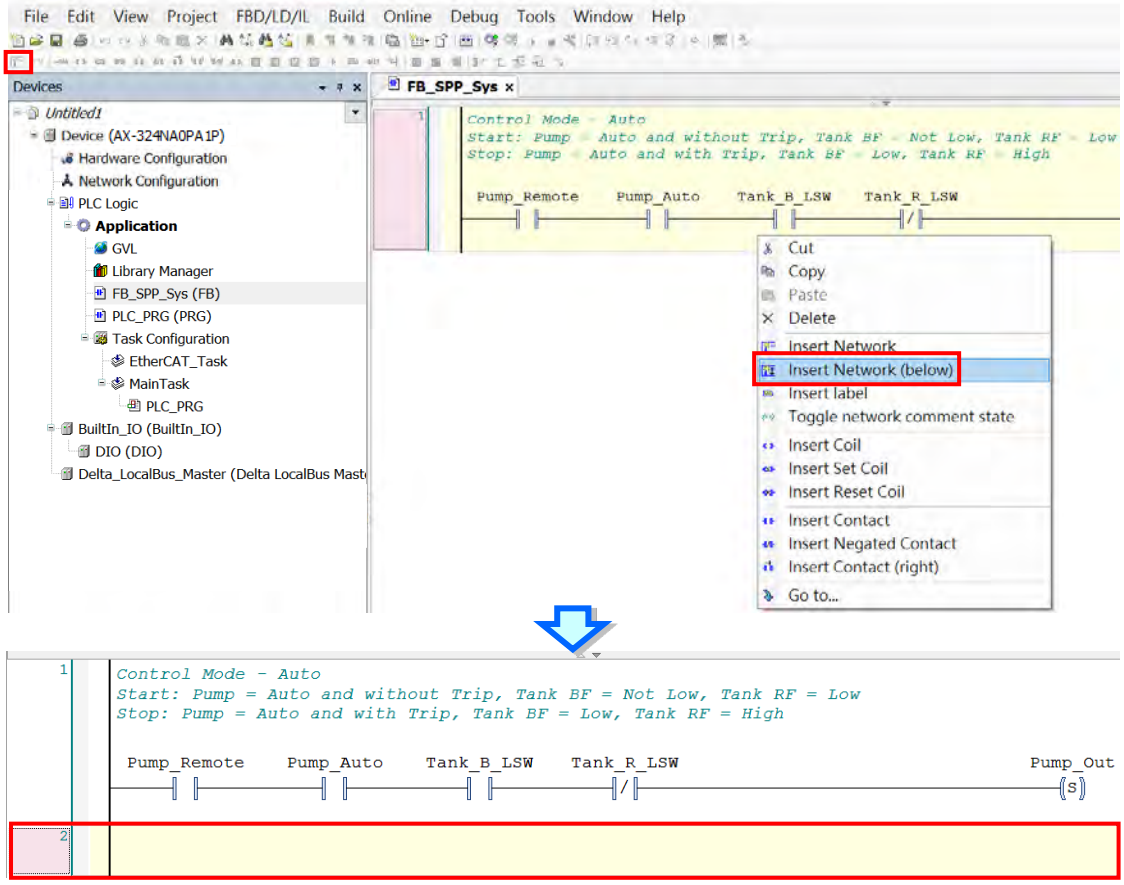


Click the ??? section to type a new name, for example, Pump_Out and then press the Enter key. Right-click the contact of Pump_Out and select **Set/Reset** from the drop-down list.

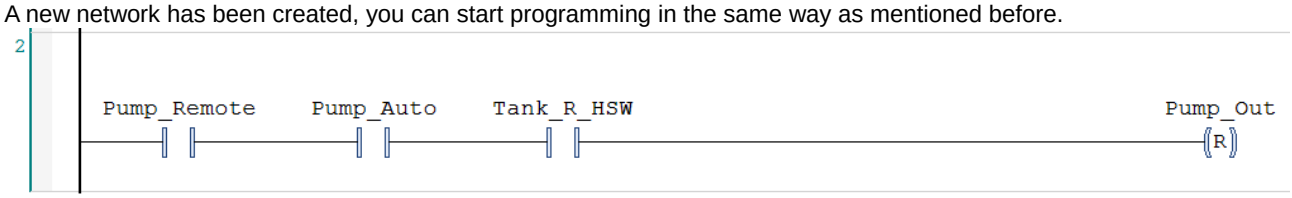


Three methods for you to insert a new network.

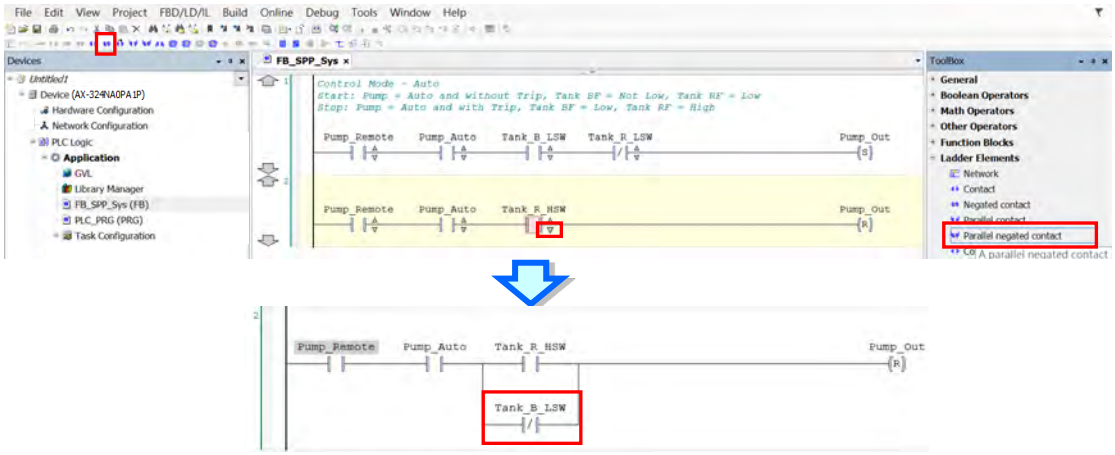
1. Click where you'd like to add a network above and then click on the tool bar above to add a new network above the network you have selected.
2. Find in the Ladder Elements of the ToolBox on the right side of the window. (ToolBox -> Ladder Elements ->); select the element and drag it to where you'd like to add the network, once you see the green box appears, drop it there.
3. Right-click the created network and select **Insert Network or Insert Network (below)** from the drop-down list.



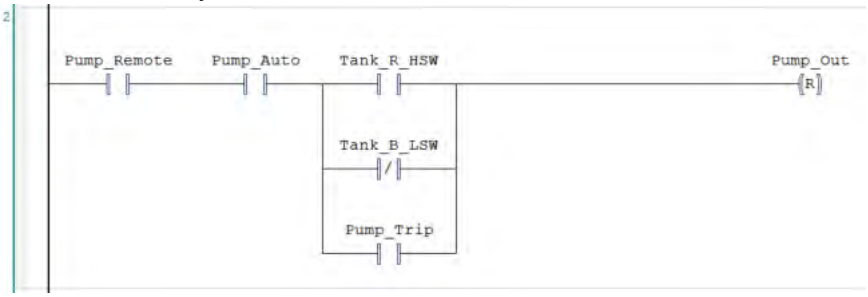
A new network has been created, you can start programming in the same way as mentioned before.



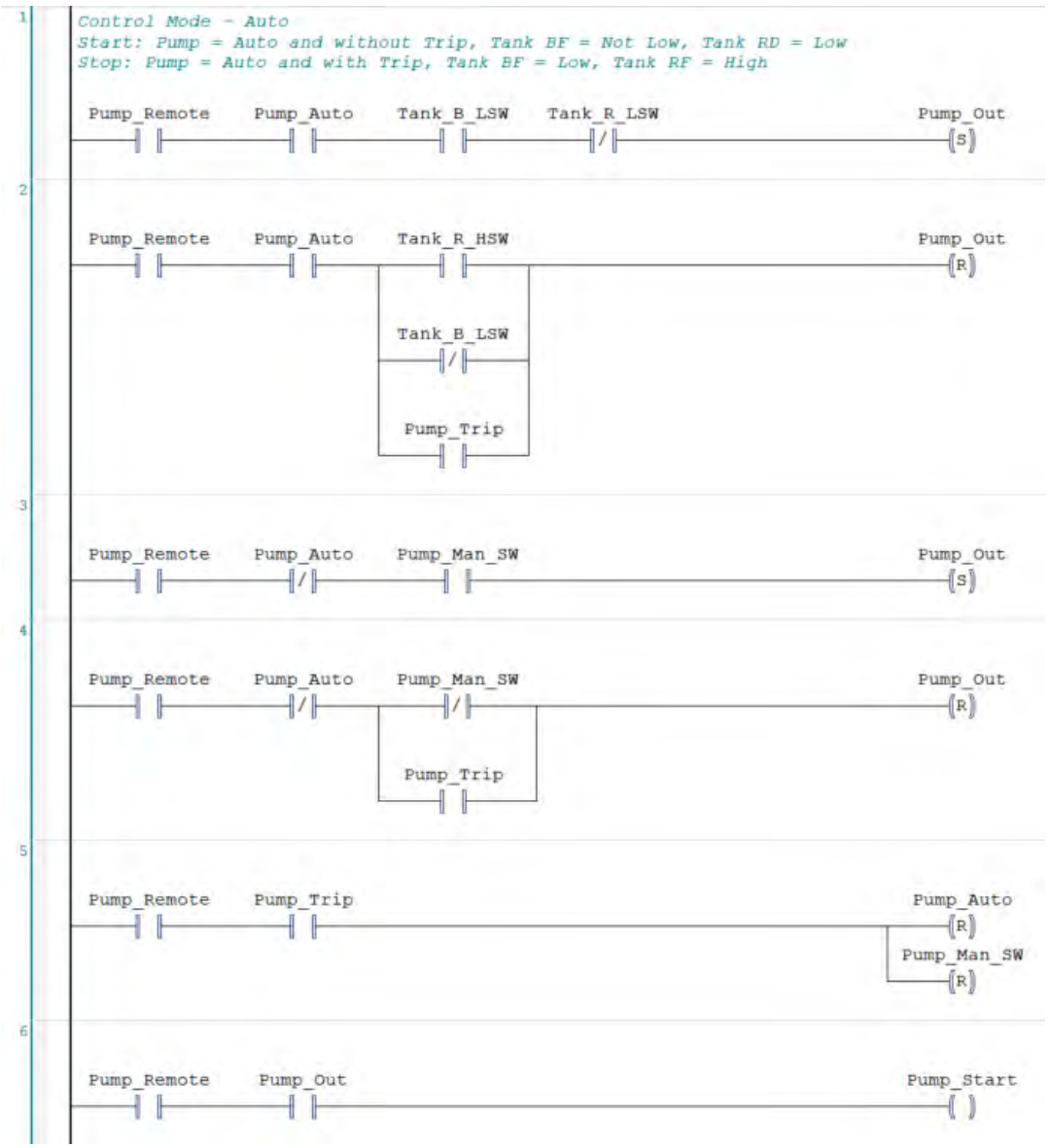
If you need the condition of no water in the underground pool or the pump trips, you can click where you'd like to add a parallel negated contact and then click Parallel negated contact on the tool bar above to add. Or find Parallel negated contact in the Ladder Elements of the ToolBox on the right side of the window (ToolBox -> Ladder Elements -> Parallel negated contact), select the element and drag it to where you'd like to add the parallel negated contact. Once you see the green obtrangular form appears, drop it there. Click the ??? section to type a new name, for example, Tank_B_LSW and then press the Enter key.



Write the following program in the same way.



Now you have complete an automatic control program. Write the following program in the same way.

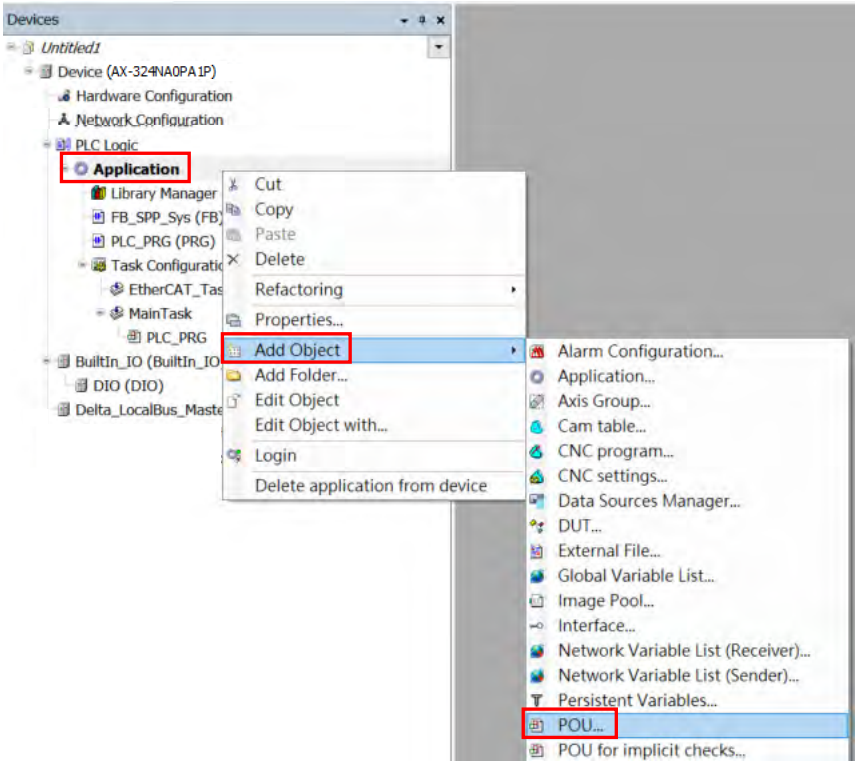


You have created the function block POU. Save the project.

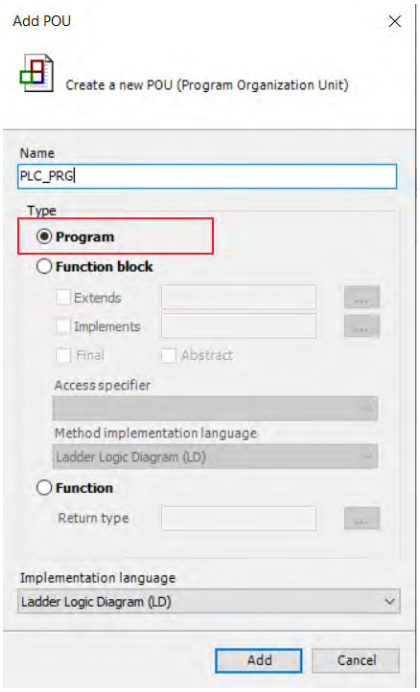
2.4.6 Creating Main Programs

After we created the function block, we can now create main programs.

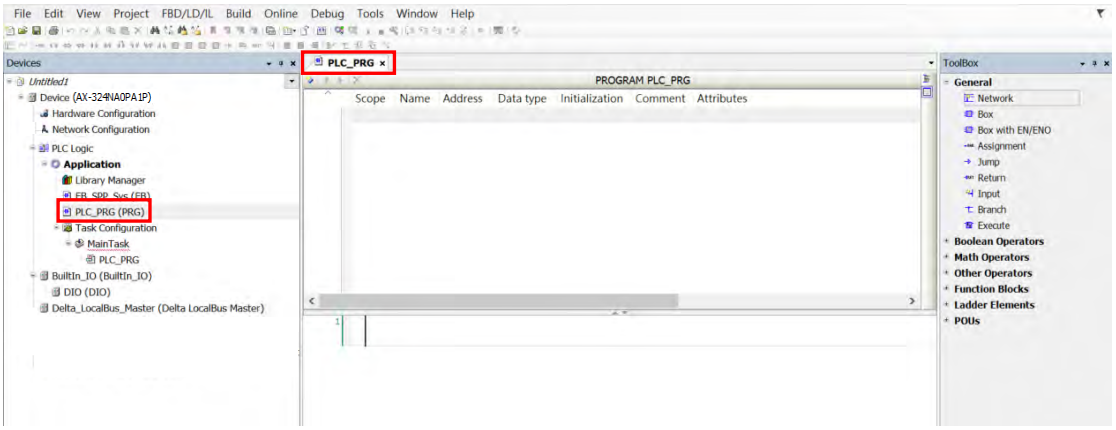
1. Right-click **Application** under the node of **PLC Logic** to open a context menu. Select **Add Object** to see another context menu. Select **POU** to open the Add POU window.



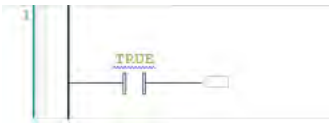
2. You can define a new name for the function block. Select **Program** in the Type section and click **Add** to add a function block.



3. After that you can find the newly created PLC_PRG (PRG) under the node of **Application** on the tree view. And the main program editing window is on the right side.



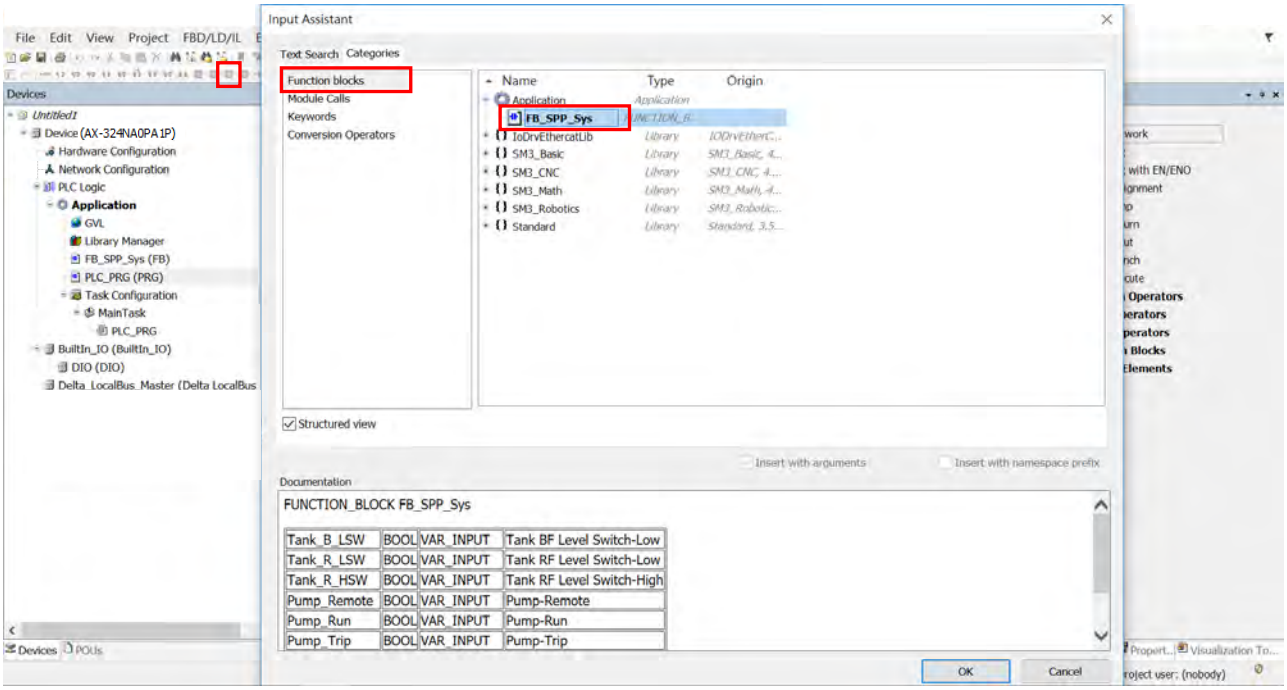
The difference between the function block and the main program is that the function block has to be called before it runs, whereas the main program runs when it is added in the Task. This example omits creating local symbols for the main program. Use what we have learned from the function block section to write the following program.



Two methods to call the function block.

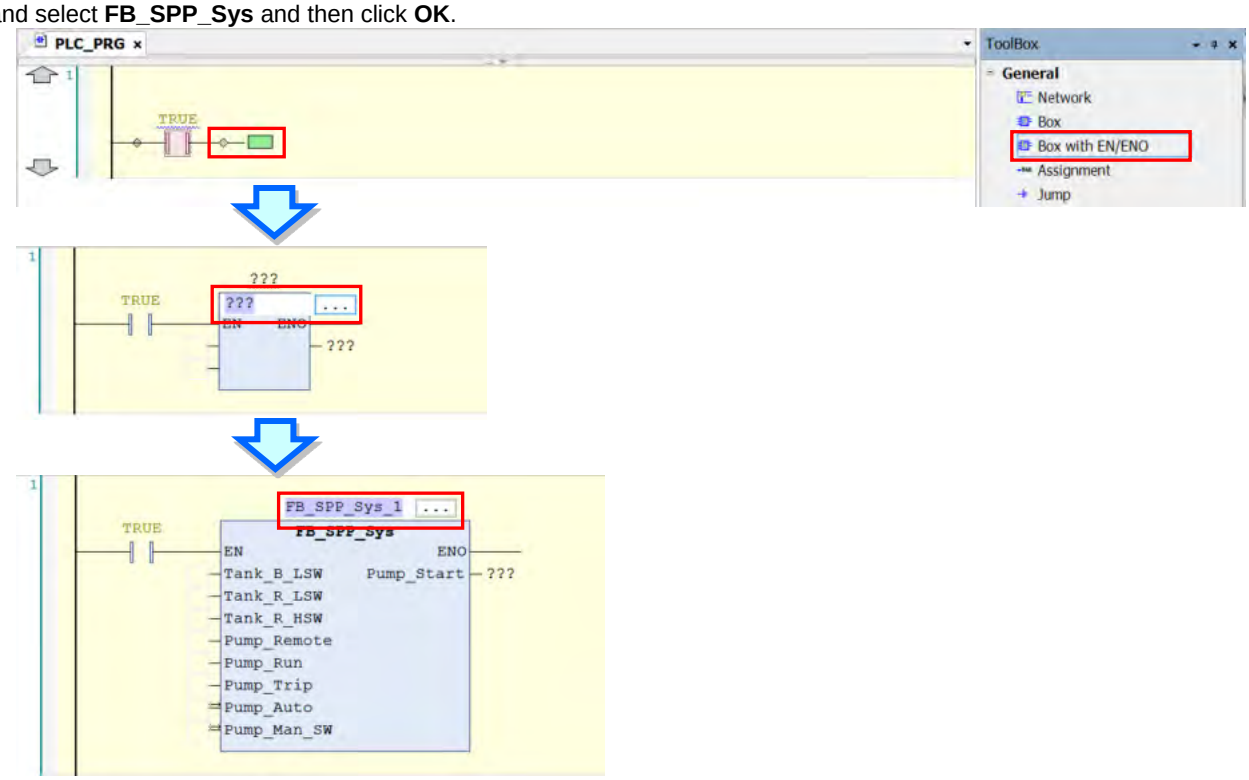
Method 1:

Click any blank space in the network to have the functional buttons on the toolbar activated. Click the functional button **Insert Box with EN/ENO** on the toolbar to open the **Input Assistant** window. Select **Funciton blocks** under the **Categories** tab. Unfold the node of **Application** and select **FB_SPP_Sys** and then click **OK**.



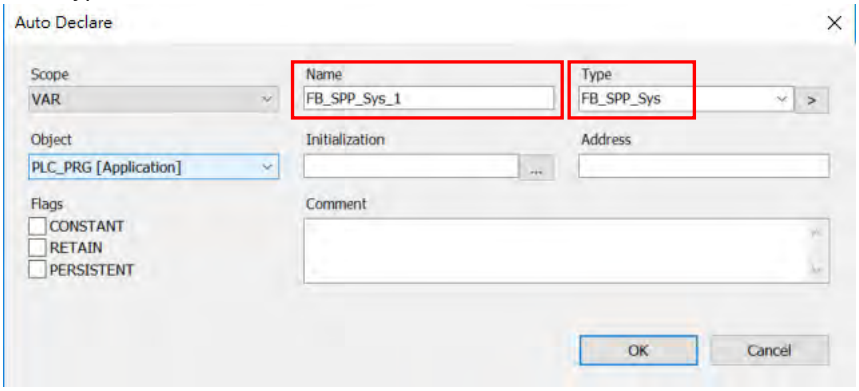
Method 2:

Click Box with EN/ENO in the General of the ToolBox on the right side of the window (ToolBox -> General -> Box with EN/ENO). Select and drag it to where you'd like to add the FB_SPP_Sys. Once you see the green obtrangular form appears, drop it there. Click the ??? section to type the name of the function block. You can also click ... after clicking the ??? section to select **Funciton blocks** under the **Categories** tab. Unfold the node of **Application** and select **FB_SPP_Sys** and then click **OK**.

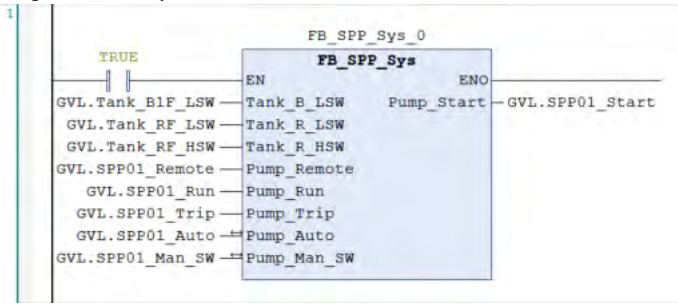


After that the system will declare a data block for this function block automatically. Its POU name is the same as the name of the function block. If you need to change the POU name, click the function block name and then press the Enter key once or twice on your keyboard to open an **Auto Declare** window. Redefine the POU name and click **OK**.

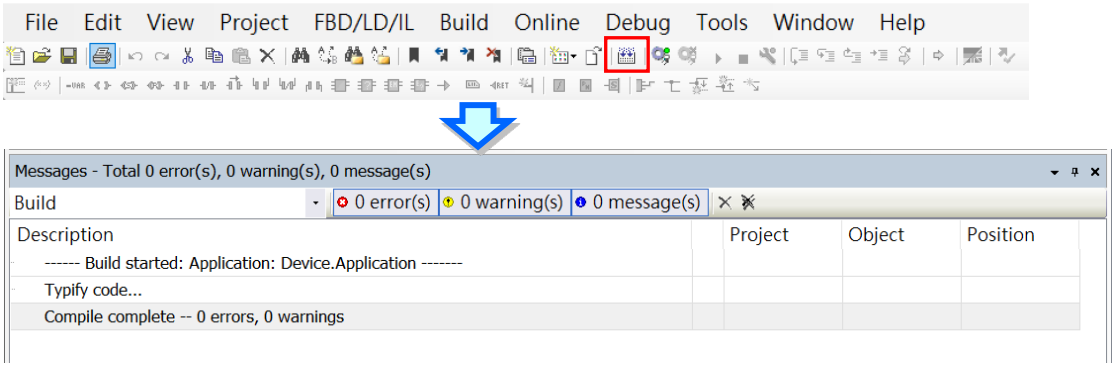
Here the data block is created as a local symbol in the main program. The way to create a data block is the same as the way you create a local symbol. The only difference is that you do not need to specify the data type. The system will auto fill the name of the data type to be the same as the name of the function block.



Refer to the followings and type the corresponding name of the global symbol in the contacts. After you save the file, the program is complete.

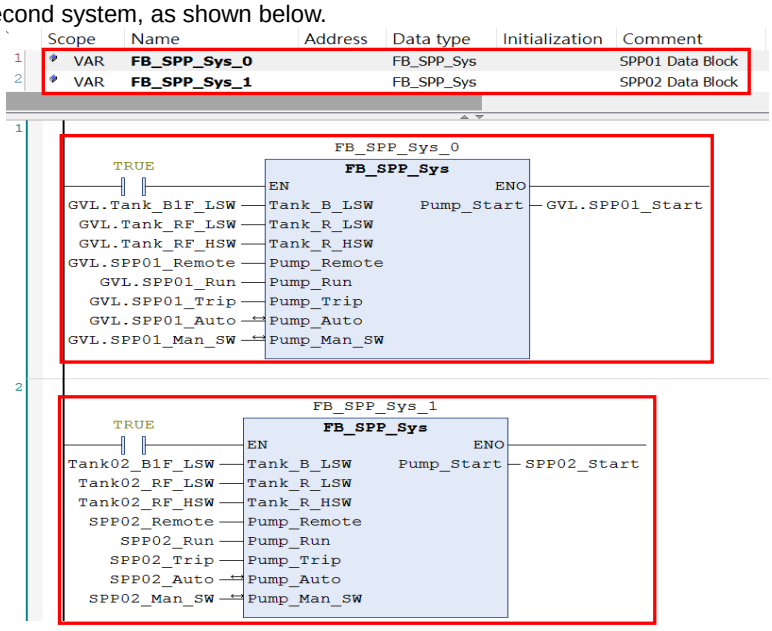


You can click  to compile the project to check if the syntax of the current program is correct. Compiling the project involves not only checking the syntax, but also compiling all programs as well as the contents of the function block.



Note: In order to help you understand the control program, the example provided here is simple. The actual control program is much more complex because the functions related to checking for start failure and conversion from the local control to remote control must be taken into account. You can make use of the tools in the ToolBox on the right side of the window and instructions in the **Input Assistant** to write programs.

If you need to create a second system, you only need to call “FB_SPP_Sys” again. Type the name of the function block to have the same function block as the one used in the first system. Note that do NOT use the same function block in two systems if the values of VAR_OUTPUT and VAR of the two systems can affects each other. In that case, create a new function block for the second system, as shown below.



MEMO

3

Chapter 3 Creating Motion Control Projects

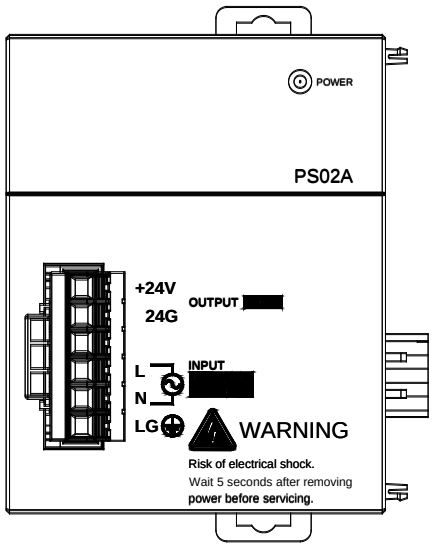
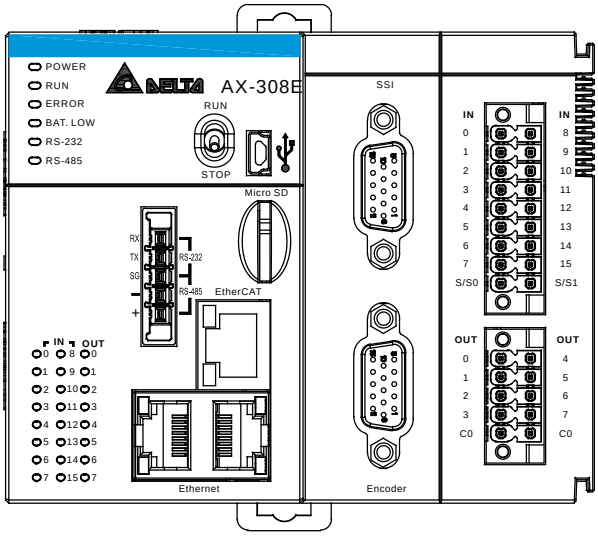
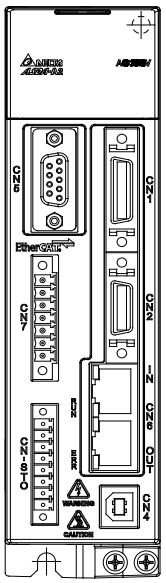
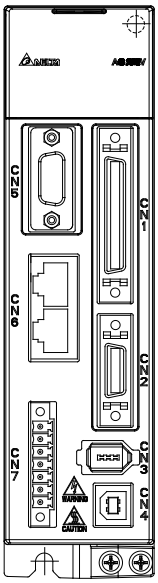
Table of Contents

3.1 Preparation

Chapter 3 provides users with the example, which is the most commonly used with AX series products, to create basic projects of motion control, as well as the required software, hardware, and tools explained in this section.

3.1.1 Hardware

The hardware needed in the example is as follows.

1. AS Series Power Module (AS-PS02A)	2. AX-3 Series CPU Module (AX-308E)
	
3. ASDA-A2-0421-E Servo Drive (EtherCAT)	**4. ASDA-A2-0421-M Servo Drive (CANOpen)**
	

3.1.2 Software

The software needed in the example is as follows.

- DIADesigner-AX V1.0.0 and above.

3.1.3 Tools and Materials

The tools and the materials needed in the example are as follows:

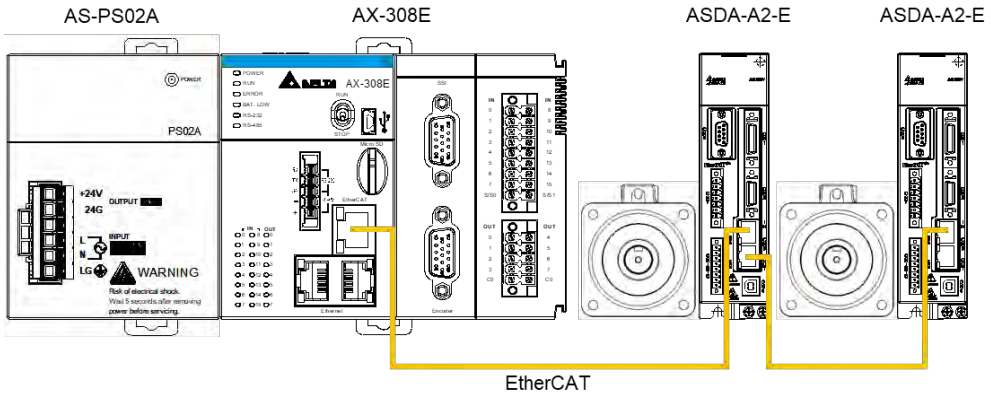
- Personal computer (with software DIADesigner-AX installed)
- A 100~240VAC 50/60Hz power supply socket
- A 24VDC power supply
- A cable
- A pin type terminal arranging kit
- At least 20 pieces of pin type terminals
- A screwdriver
- An USB cable or a network cable

3.2 Setup Servo Axis Projects via EtherCAT

The example is a dual-axis servo drive project which controls moving back and forth.

3.2.1 System Hardware Configuration

AX-308 connects to two servo drivers (ASDA-A2-E) directly through EtherCAT communication ports to control the positioning of dual-axis servo motors.



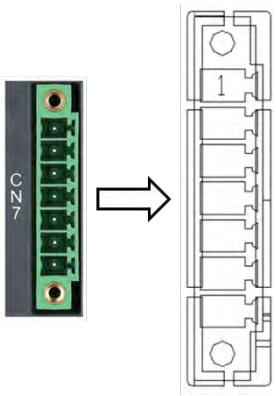
- Tools and materials

Numbering	Item	Model Number	Q'ty
1	AS Series Power Module	AS-PS02A	1
2	AX-3 Series CPU Module	AX-308EA0MA1T	1
3	Power supply	DVP-PS02	1
4	Servo drive (EtherCAT)	ASDA-A2-0421-E	2
5	Servo motor	ECMA-C10604ES	2
6	Servo power cable	ASD-ABPW0003	2
7	Servo encoder cable	ASD-ABEN0003	2
8	EtherCAT communication cable (1 meter)	UC-EMC010-02A	2

- Configuring hardware limits

Configure the hardware limit on CN1 or CN7 which CN7 connector is used in the following example.

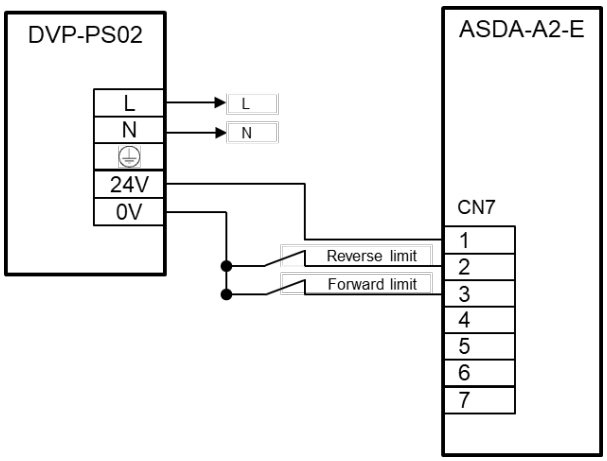
Use external power DC24V and set Reverse/ Forward limit switches. Then configure DI9~DI10 pin functions of ASDA-A2-E Servo with same parameter setting.



- CN7 Extension DI

Pin No.	Signal Name	Terminal Symbol	Function and Description
1	VDD 24 V power	COM+	VDD (24 V) power is the same as the voltage of Pin11 in CN1.
2	Extension DI9	EDI9-	Digital input pin 9-
3	Extension DI10	EDI10-	Digital input pin 10-
4	Extension DI11	EDI11-	Digital input pin 11-
5	Extension DI12	EDI12-	Digital input pin 12-
6	Extension DI13	EDI13-	Digital input pin 13-
7	Extension DI14	EDI14-	Digital input pin 14-

■ Limit switch in ASDA-A2-E



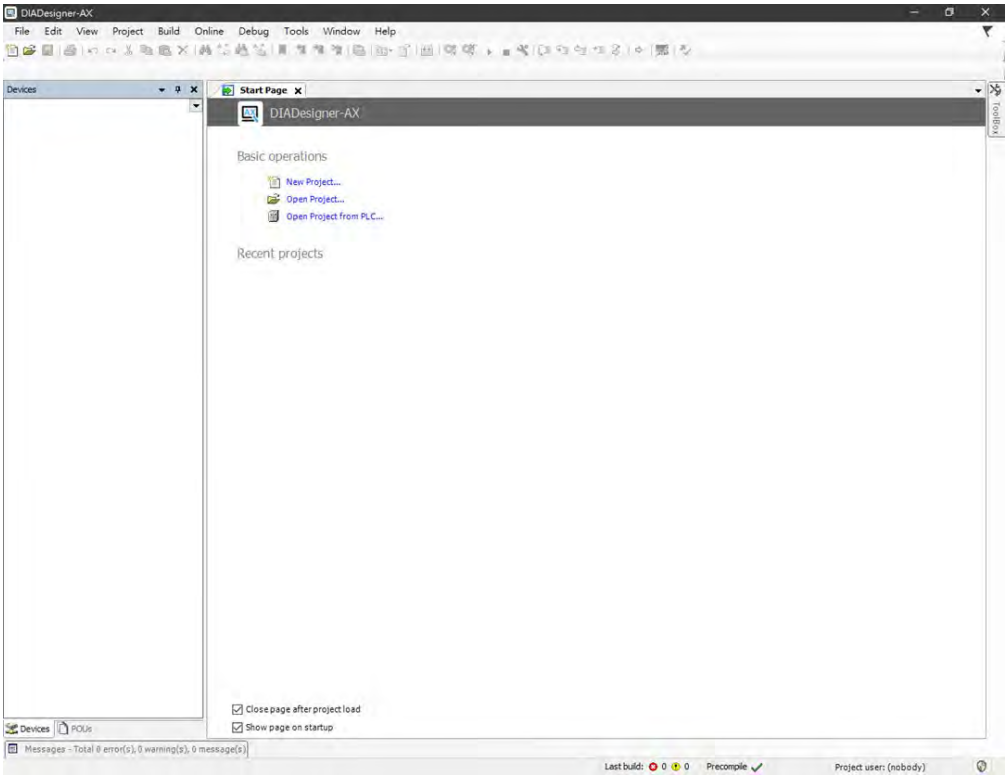
***Note:** For more information regarding signal configuration in ASDA-A2-E, please refer to the user manuals of Delta ASDA-A2 Series.

3.2.2 Create New Projects and POUs

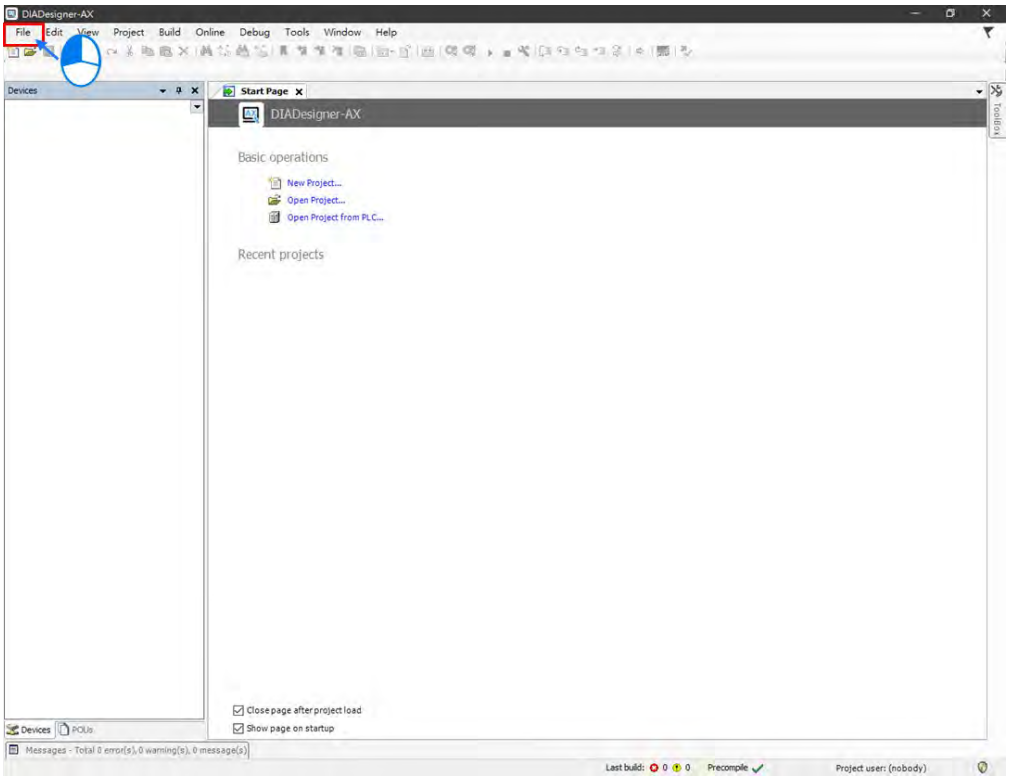
3.2.2.1 Create a New Project

1. First, open the programming software DIADesigner-AX.

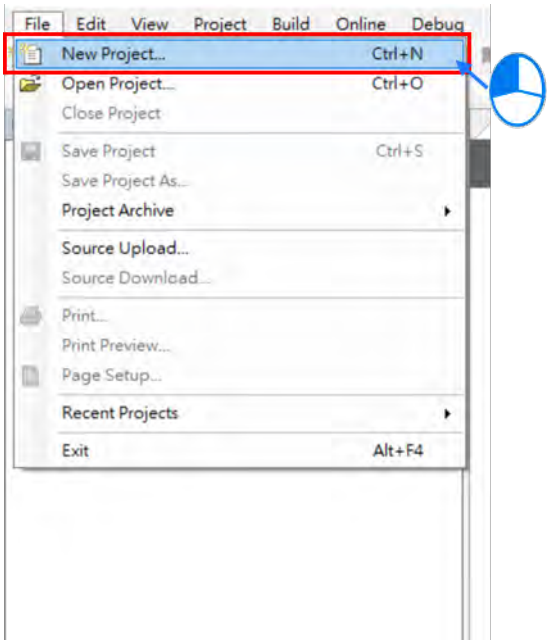
2. The display of DIADesigner-AX is shown as below.



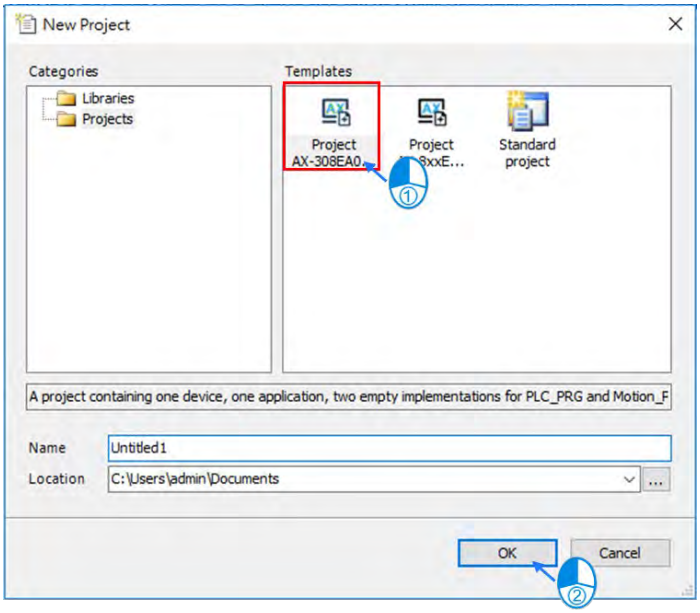
3. Click on “File”.



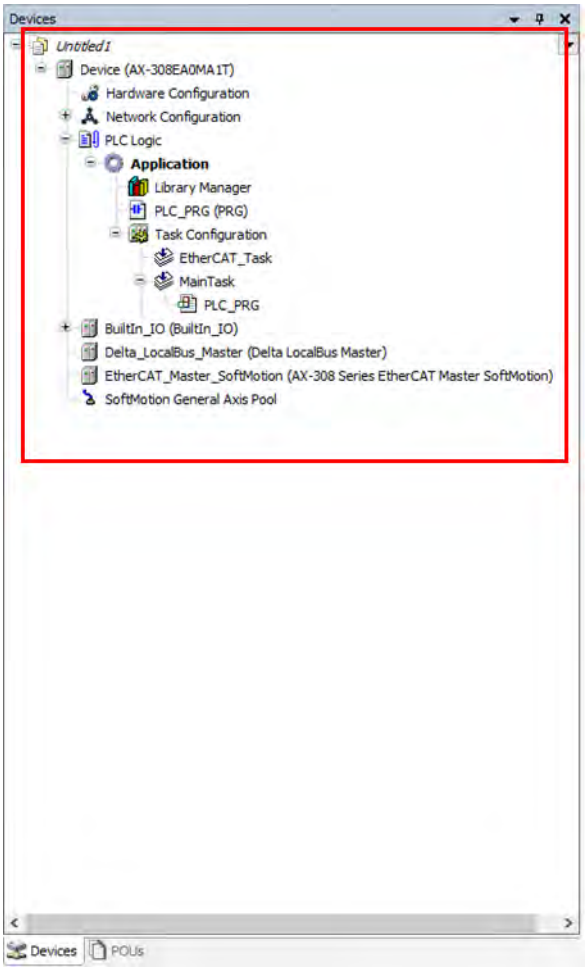
4. To create a new project, select "New Project" from the drop-down menu for DIADesigner-AX V1.0.0 or select “Standard Project” from the drop-down menu for DIADesigner-AX V1.1.0 or later.



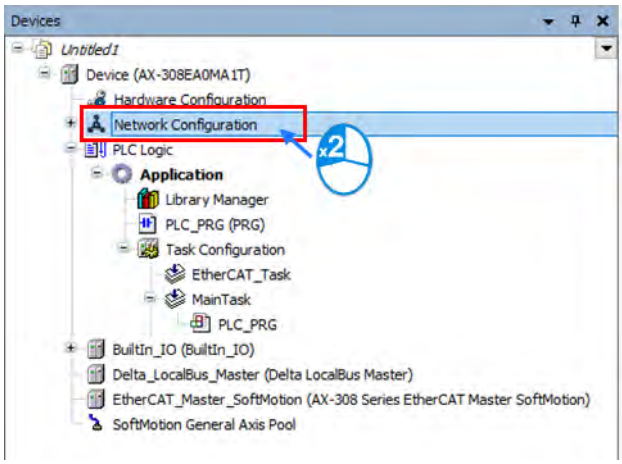
5. Select “Project AX-308EA0MA1T” after entering New Project page. Name the project in the field of Name and choose a path to the archive location, then click OK to move to the next page.



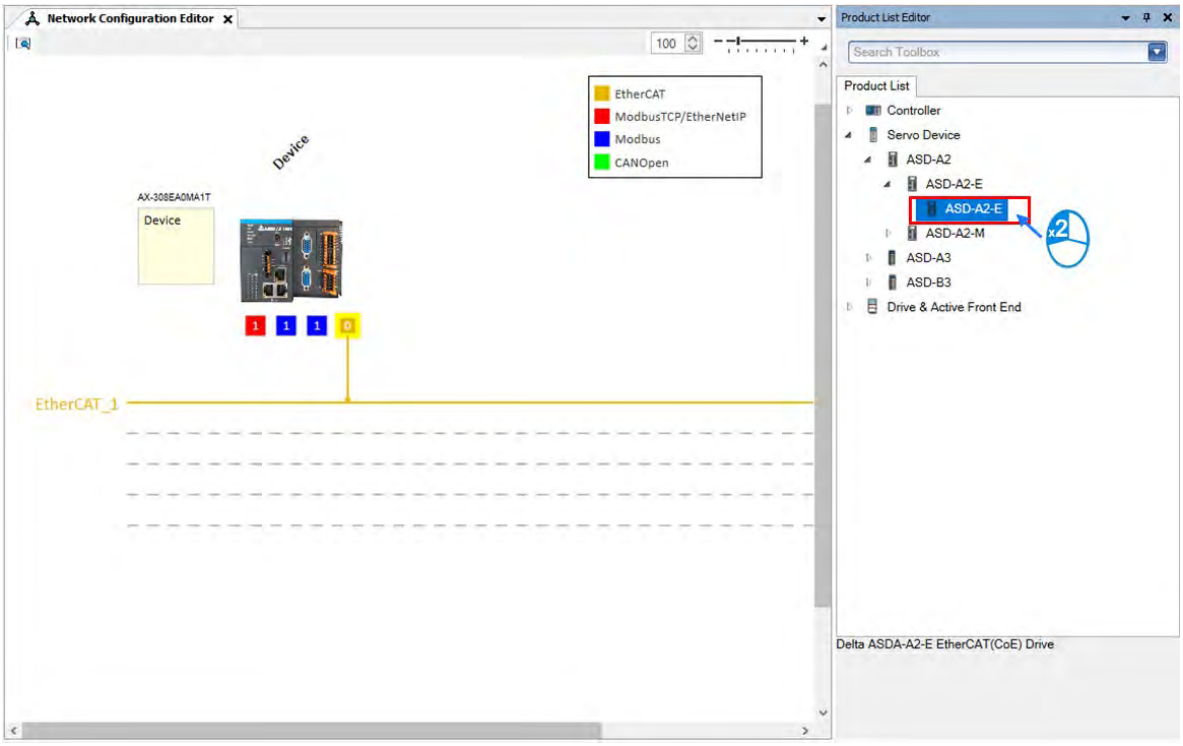
6. After entering the project page, double click the target object in the project tree on the left side of the page to open a certain program or configure settings of relating modules.



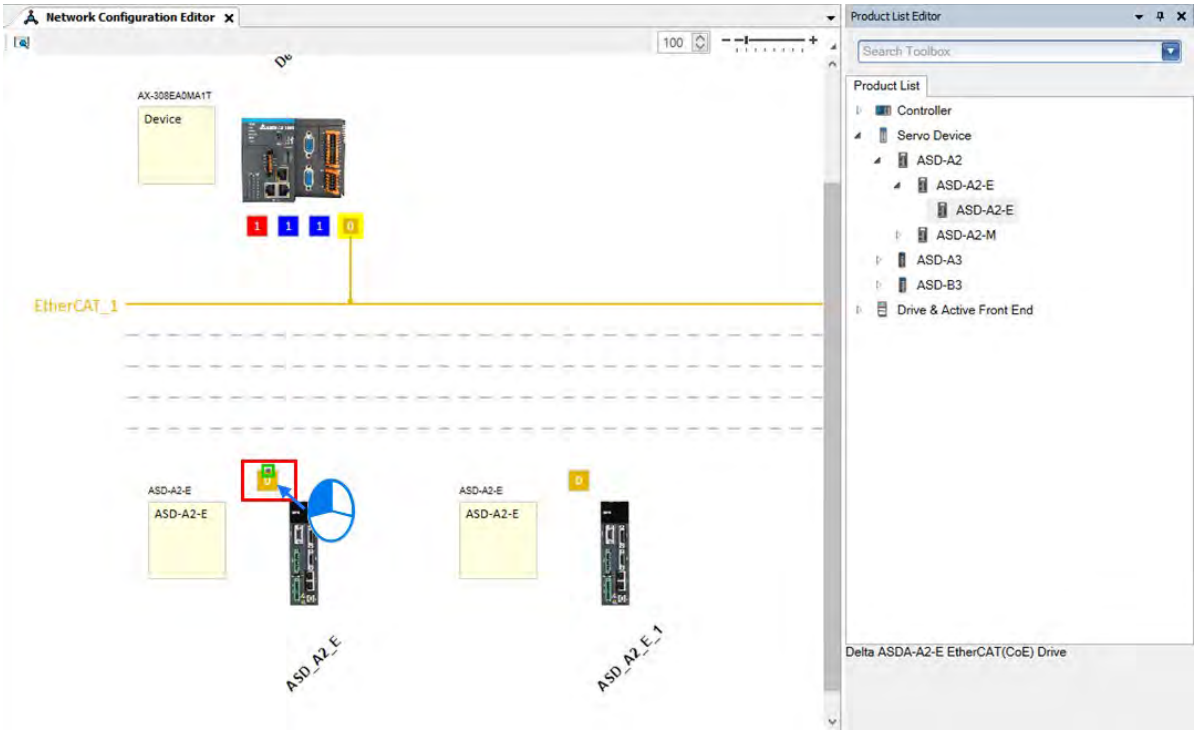
7. Double click on “Network Configuration” to create servo axes in the system.



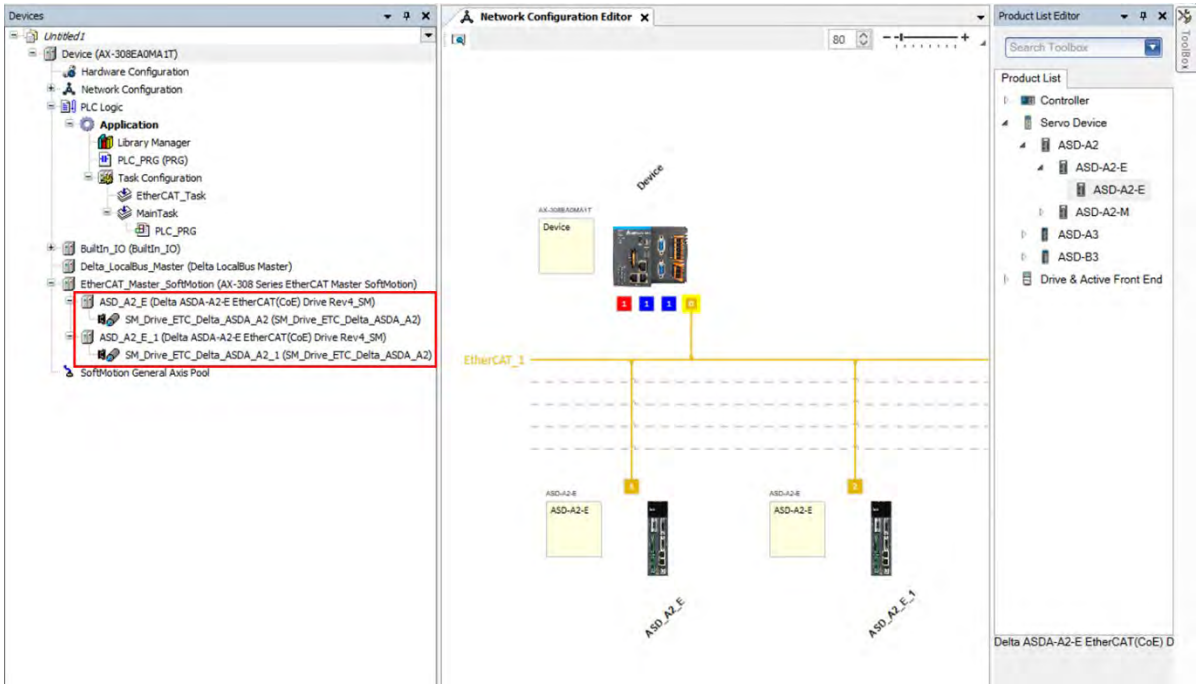
8. After entering “Network Configuration” page, select the servo drive slave based on EtherCAT and double click on it to add new axes.



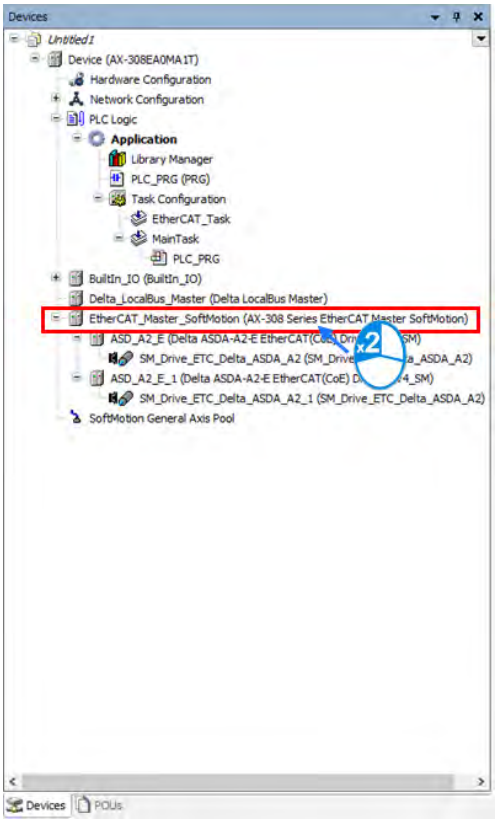
9. Move the cursor onto 0, then click and hold the left mouse button and drag the yellow line to connect with the line of EtherCAT_1.



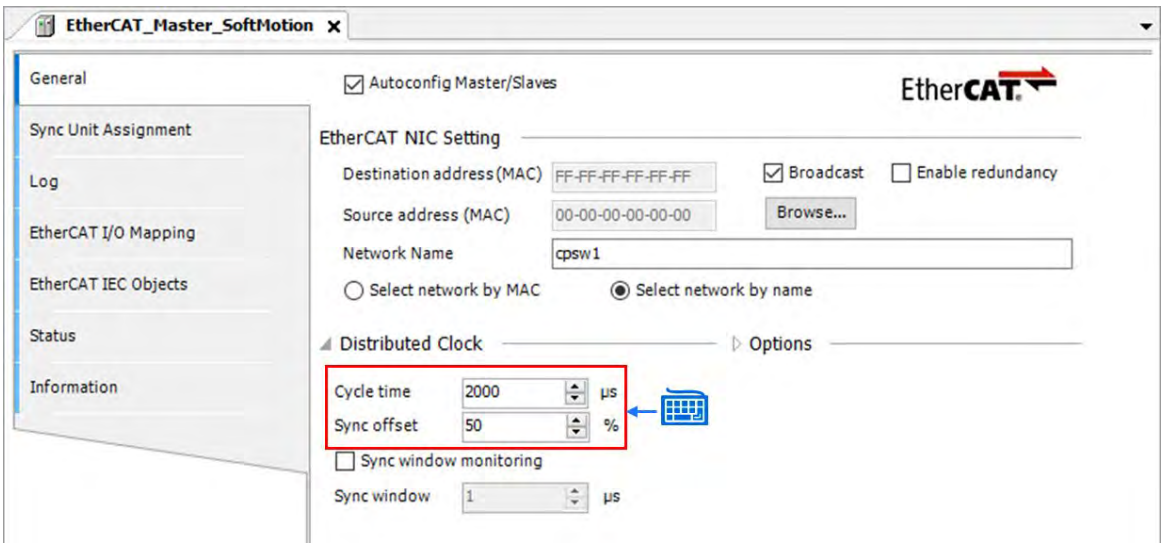
10. After connecting with EtherCAT_1, the slave device is displayed in the project tree (fields with red borders) which also means it is successfully configured with EtherCAT communication.



11. Double click on “EtherCAT_Master_SoftMotion”.

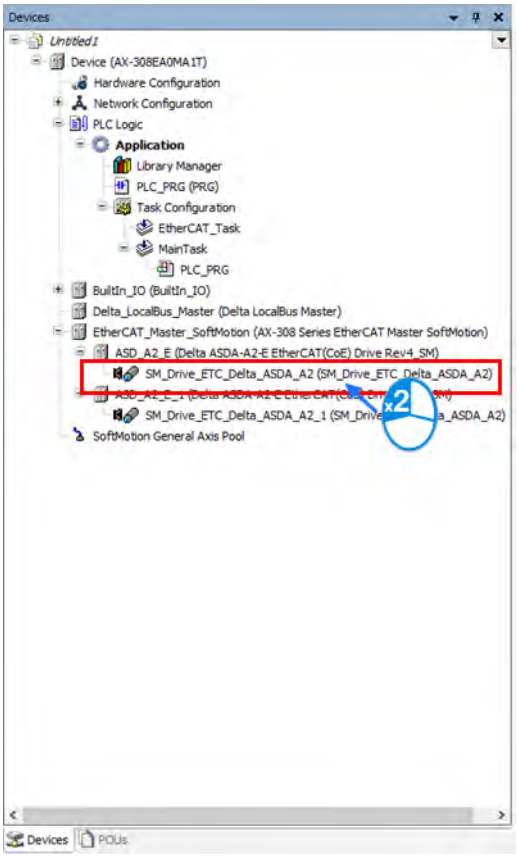


12. On the EtherCAT_Master_SoftMotion page, set “Cycle time” and “Sync offset” as 2ms and 50 respectively.

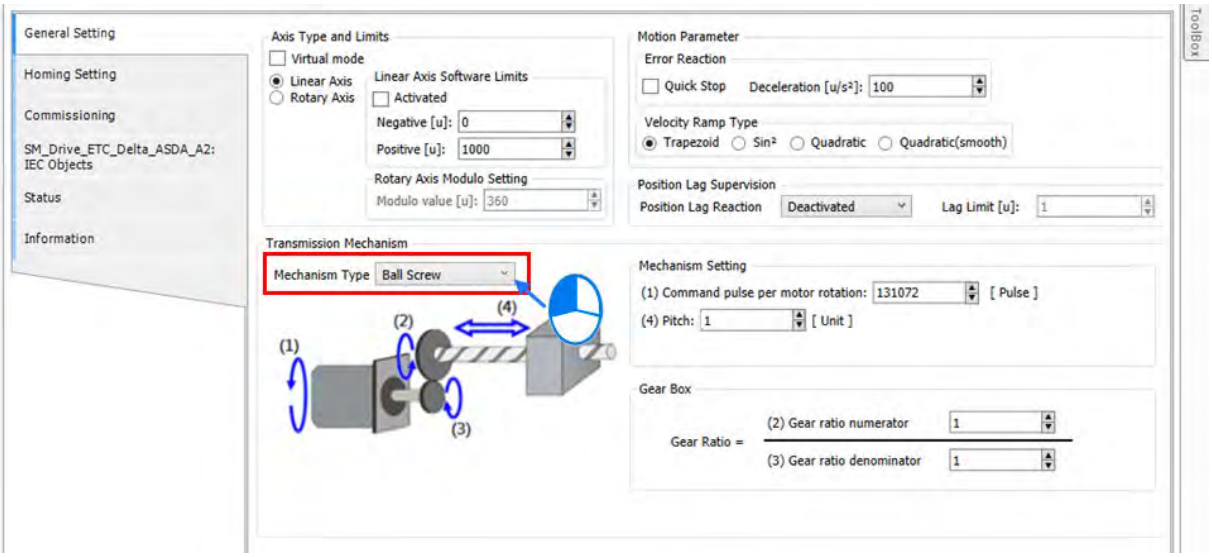


3.2.2.2 Axis Parameter Settings

1. Select the first servo axis “SM_Drive_ETC_Delta_ASDA_A2” and double click on it.

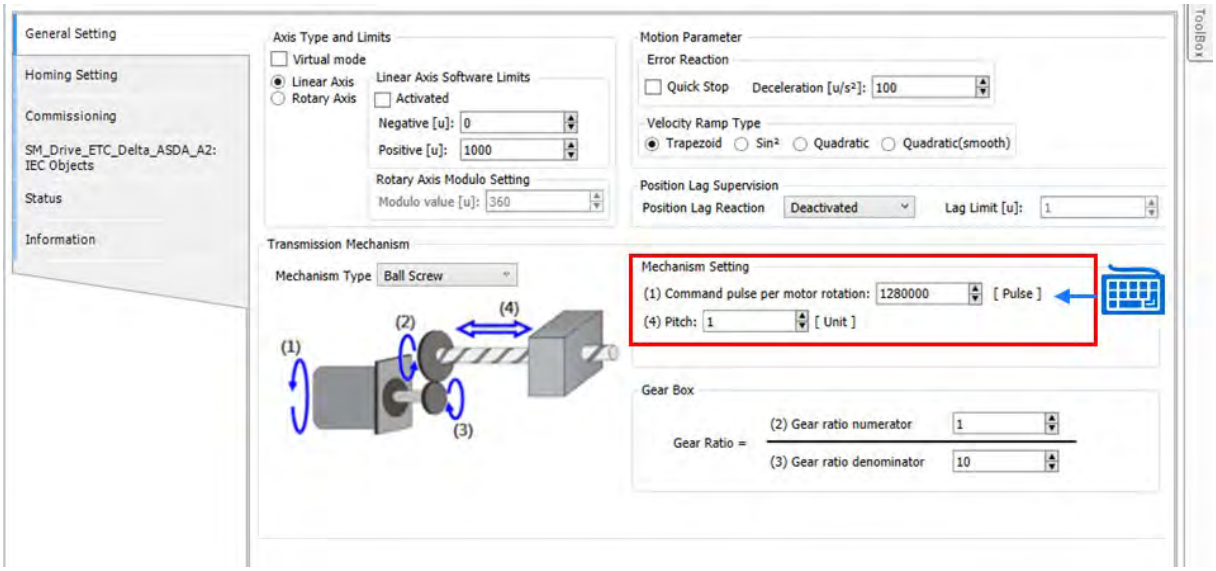


2. In this example, choose “Ball Screw” as the mechanism type after entering “SM_Drive_ETC_Delta_ASDA_A2” page.

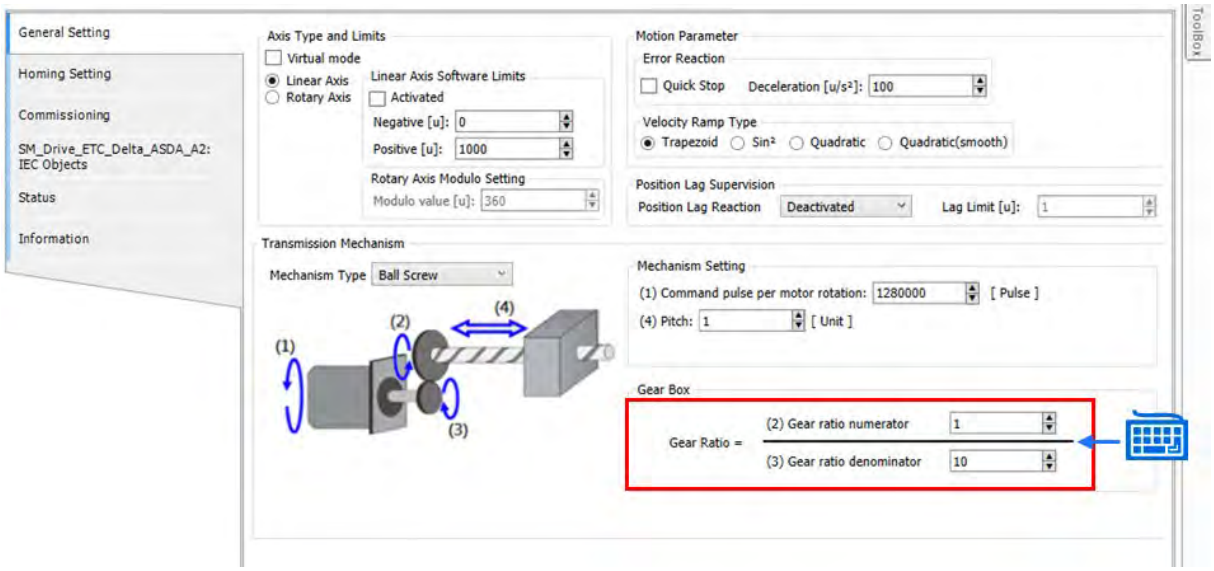


3. Then configure the mechanism setting, which is related to the result of gear rato, with input of 1,280,000 for “Command pulse per motor rotation” and 1mm for “Pitch”.

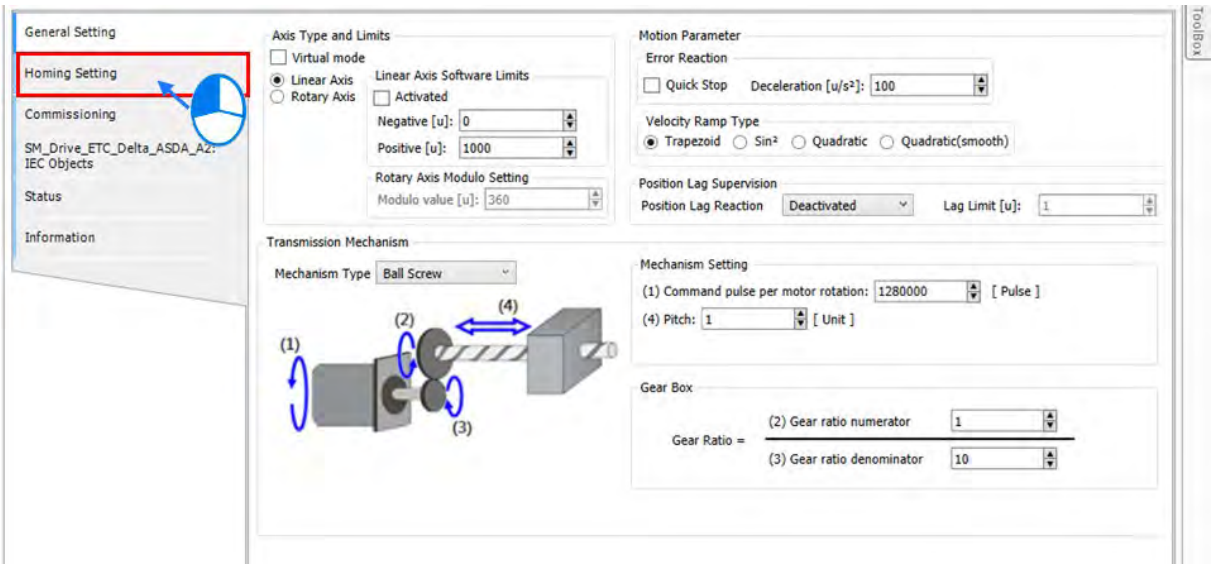
(Servo gear ratio $\frac{\text{P1-44}}{\text{P1-45}} = \frac{\text{Motor Pulse per revolution}}{\text{Command Pulse per rotation}}$)



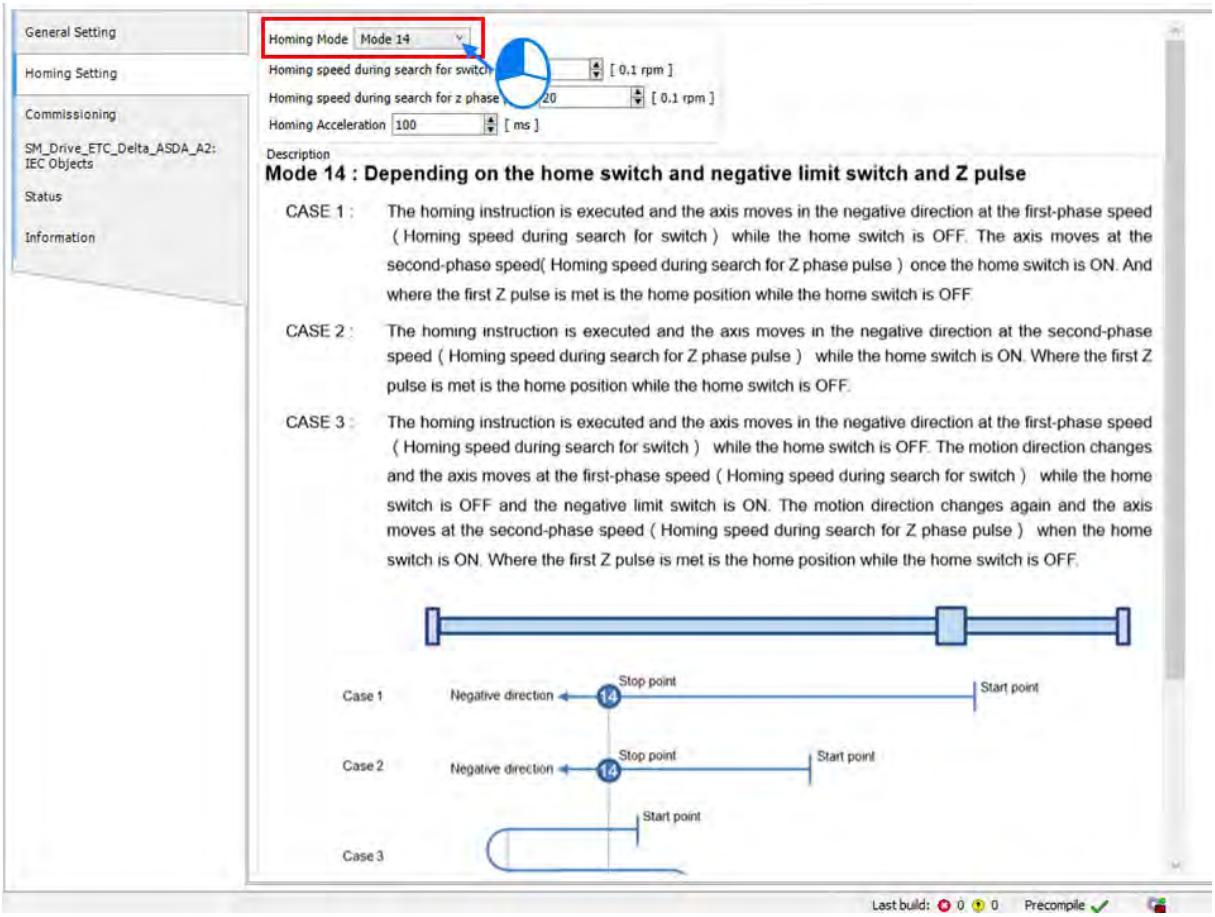
4. Since the gear reducer ratio is 10:1, input 1 for the numerator and 10 for the denominator.



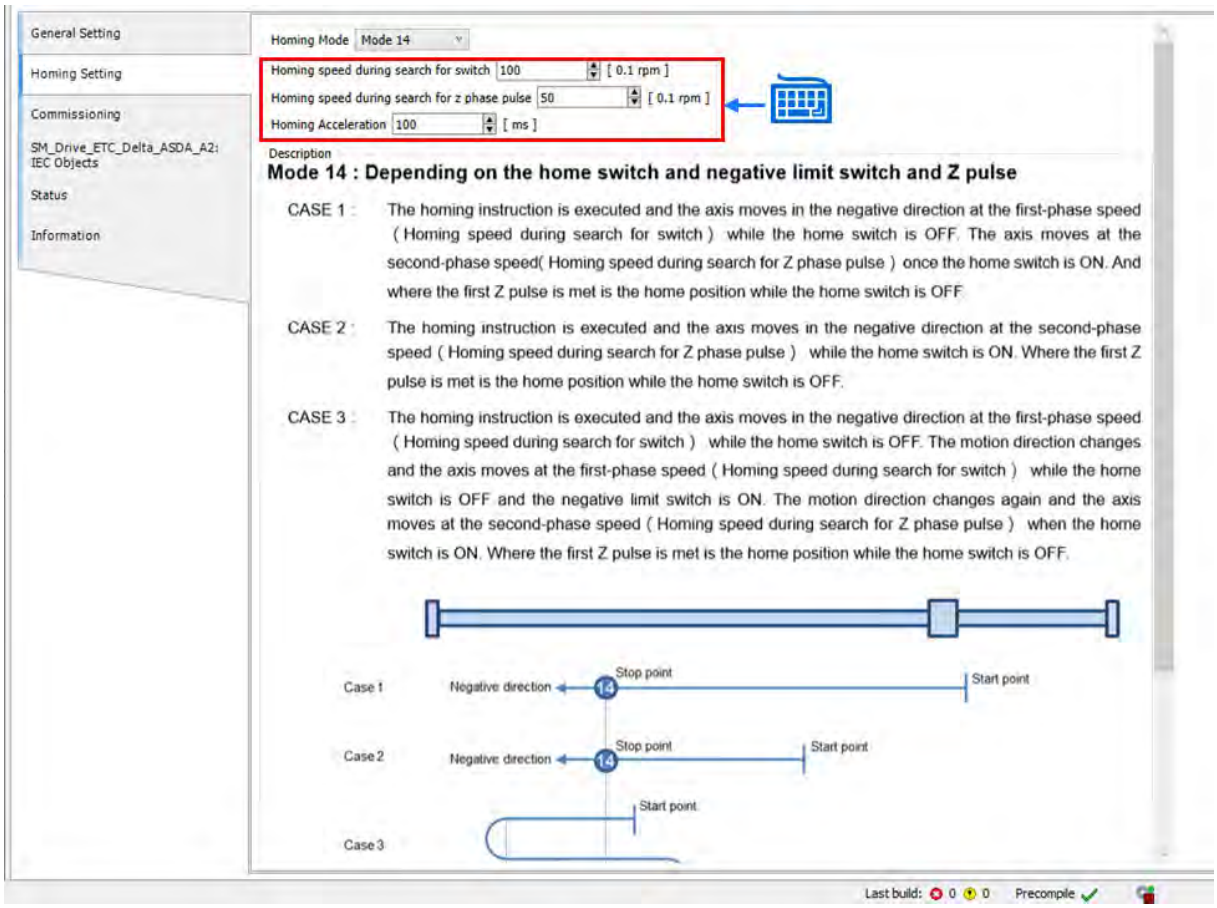
5. When finished the configuration of gear ratio, click “Homing Setting” to enter the page.。



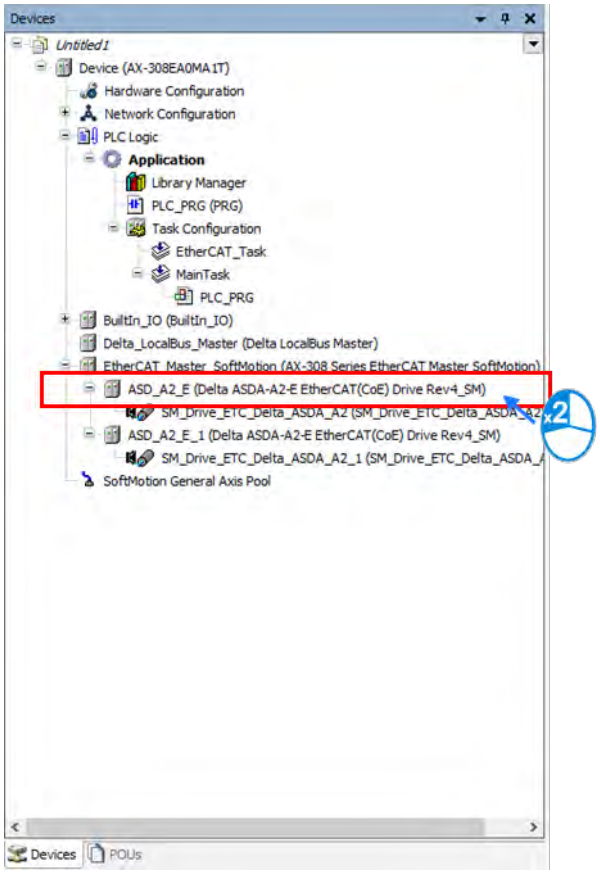
6. Choose the “Homing mode” corresponded to the hardware limit configuratuon, which the detailed descriptions of all homing modes are shown on the lower half of the page.



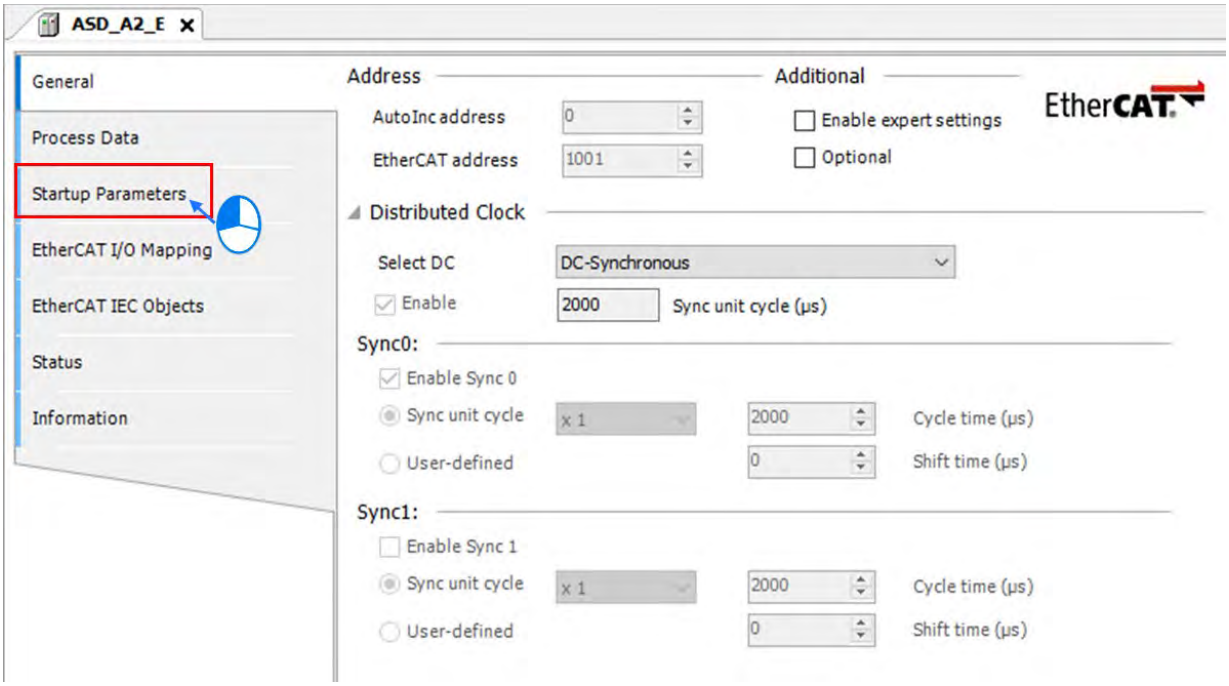
7. Set the proper homing speed which should match with the mechanism. If the speed is too fast, the hardware limit would possibly be exceeded.



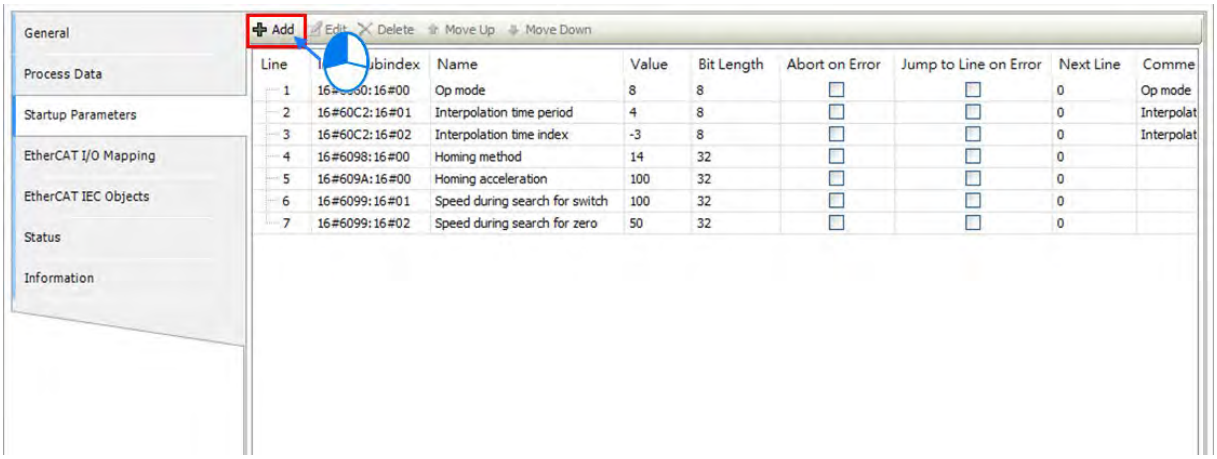
8. After finish the settings of gear ratio and homing mode, the parameters on the servo drive need to be tested for DI9 and DI10 inputs placed on the servo as a result of hardware limit. To configure the pin functions of DI9 and DI10, double click on “ASD_A2_E”.



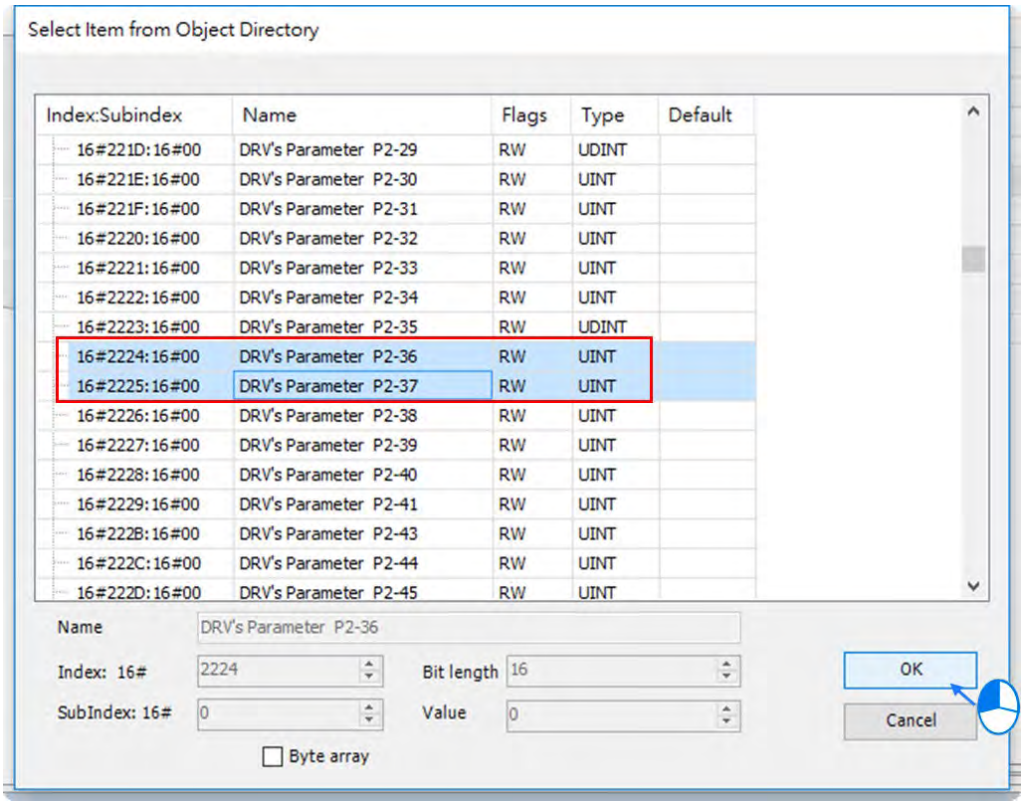
9. Click “Startup Parameters” tab on the ASD_A2_E page.



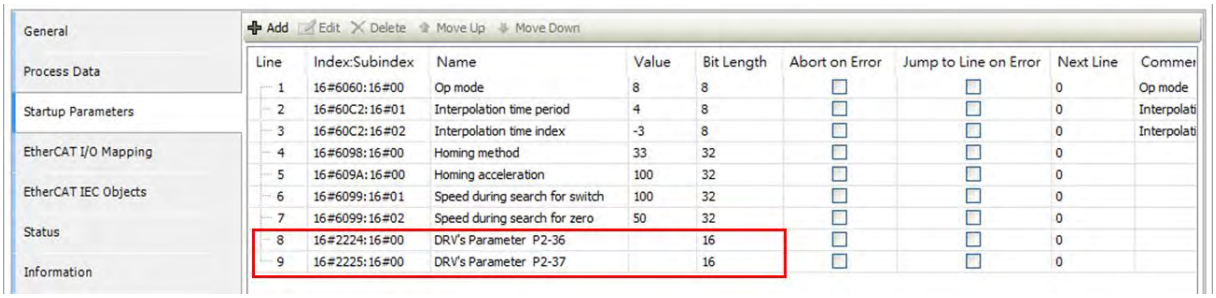
10. Click “Add” to add new parameters on “Startup parameters” page.



11. Select “P2-36” and hold the Shift button to select multiple parametres. Then click OK to add the selected parameters.

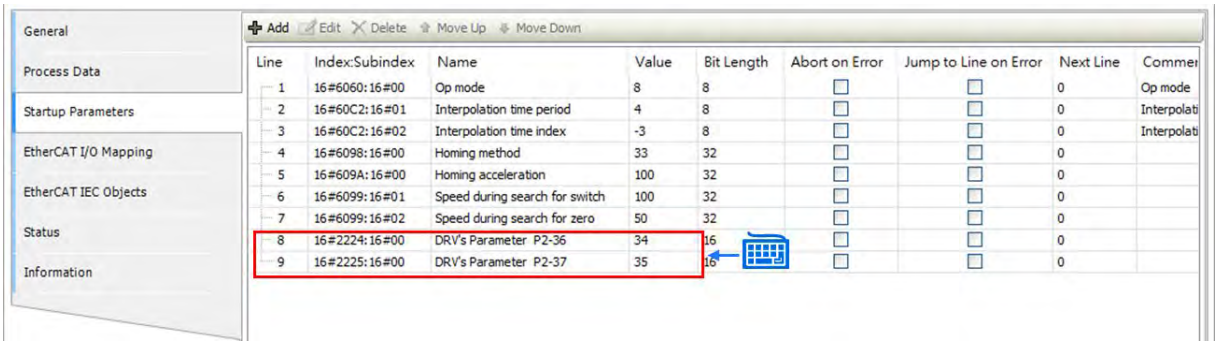


12. The selected parameters you've just added will be displayed on the list.



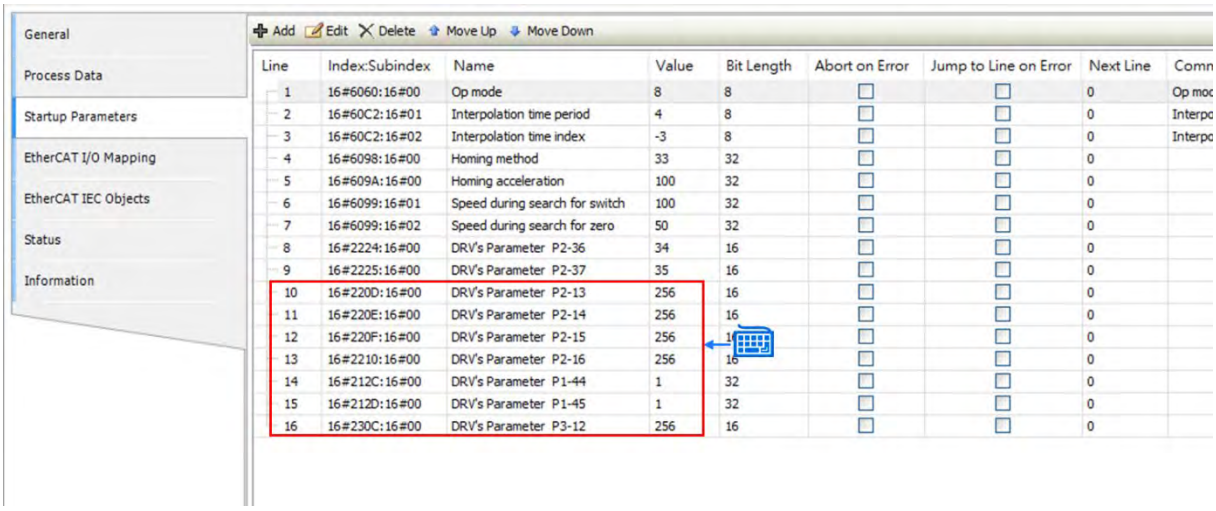
Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Error	Next Line	Commer
1	16#6060:16#00	Op mode	8	8	☐	☐	0	Op mode
2	16#60C2:16#01	Interpolation time period	4	8	☐	☐	0	Interpolati
3	16#60C2:16#02	Interpolation time index	-3	8	☐	☐	0	Interpolati
4	16#6098:16#00	Homing method	33	32	☐	☐	0	
5	16#609A:16#00	Homing acceleration	100	32	☐	☐	0	
6	16#6099:16#01	Speed during search for switch	100	32	☐	☐	0	
7	16#6099:16#02	Speed during search for zero	50	32	☐	☐	0	
8	16#2224:16#00	DRV's Parameter P2-36		16	☐	☐	0	
9	16#2225:16#00	DRV's Parameter P2-37		16	☐	☐	0	

13. Set the value of P2-36 to 34 and P2-37 to 35 in deximal mode. (DI9 is set to negative limit (B pin) ; DI10 is set to positive limit (Normally Closed pin) (All the setting values on this parameter list will be downloaded to the servo drive once the EtherCAT communication is established.)



Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Error	Next Line	Commer
1	16#6060:16#00	Op mode	8	8	☐	☐	0	Op mode
2	16#60C2:16#01	Interpolation time period	4	8	☐	☐	0	Interpolati
3	16#60C2:16#02	Interpolation time index	-3	8	☐	☐	0	Interpolati
4	16#6098:16#00	Homing method	33	32	☐	☐	0	
5	16#609A:16#00	Homing acceleration	100	32	☐	☐	0	
6	16#6099:16#01	Speed during search for switch	100	32	☐	☐	0	
7	16#6099:16#02	Speed during search for zero	50	32	☐	☐	0	
8	16#2224:16#00	DRV's Parameter P2-36	34	16	☐	☐	0	
9	16#2225:16#00	DRV's Parameter P2-37	35	16	☐	☐	0	

14. Use the same operation to configure the following parameters as well as setting the DI4~DI7 input status to be OFF and the gear ratio on the servo drive to be 1:1.



Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Error	Next Line	Comn
1	16#6060:16#00	Op mode	8	8	☐	☐	0	Op mo
2	16#60C2:16#01	Interpolation time period	4	8	☐	☐	0	Interpo
3	16#60C2:16#02	Interpolation time index	-3	8	☐	☐	0	Interpo
4	16#6098:16#00	Homing method	33	32	☐	☐	0	
5	16#609A:16#00	Homing acceleration	100	32	☐	☐	0	
6	16#6099:16#01	Speed during search for switch	100	32	☐	☐	0	
7	16#6099:16#02	Speed during search for zero	50	32	☐	☐	0	
8	16#2224:16#00	DRV's Parameter P2-36	34	16	☐	☐	0	
9	16#2225:16#00	DRV's Parameter P2-37	35	16	☐	☐	0	
10	16#220D:16#00	DRV's Parameter P2-13	256	16	☐	☐	0	
11	16#220E:16#00	DRV's Parameter P2-14	256	16	☐	☐	0	
12	16#220F:16#00	DRV's Parameter P2-15	256	1	☐	☐	0	
13	16#2210:16#00	DRV's Parameter P2-16	256	16	☐	☐	0	
14	16#212C:16#00	DRV's Parameter P1-44	1	32	☐	☐	0	
15	16#212D:16#00	DRV's Parameter P1-45	1	32	☐	☐	0	
16	16#230C:16#00	DRV's Parameter P3-12	256	16	☐	☐	0	

15. Continue to configure the second servo axis with the same setting operation as the first axis.

- Servo parameters

  The following parameters and values are used for this example on the Startup Parameter page.

Parameter	Function	Setting Value
P1-44	Gear ratio numerator	1
P1-45	Gear ratio denominator	1
P2-13	DI4 digital input pin function	256
P2-14	DI5 digital input pin function	256
P2-15	DI6 digital input pin function	256
P2-16	DI7 digital input pin function	256
P2-36	DI9 digital input pin function	34
P2-37	DI10 digital input pin function	35
P3-12	Remain unchanged with the parameters before the drive power being cut off.	256

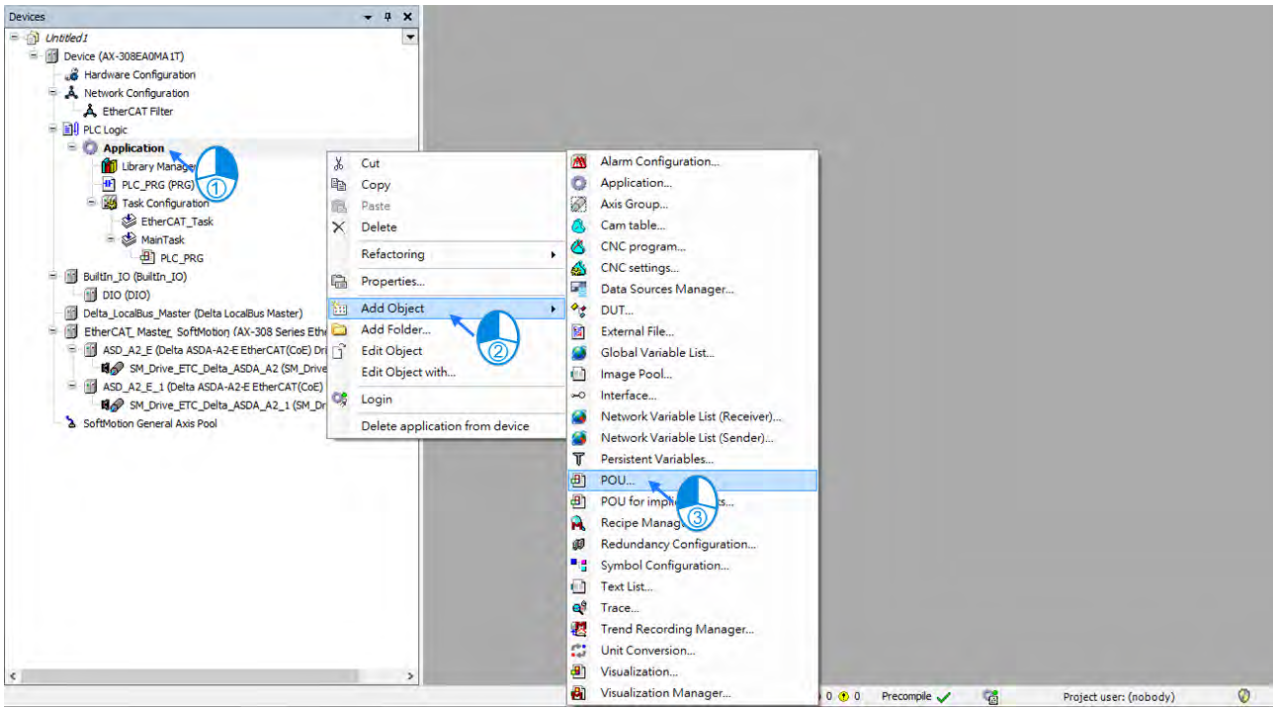
***Note:** Please refer to the user manuals of Delta ASDA-A2 Series for more detailed information of parameters.

3.2.2.3 Programming for New Projects

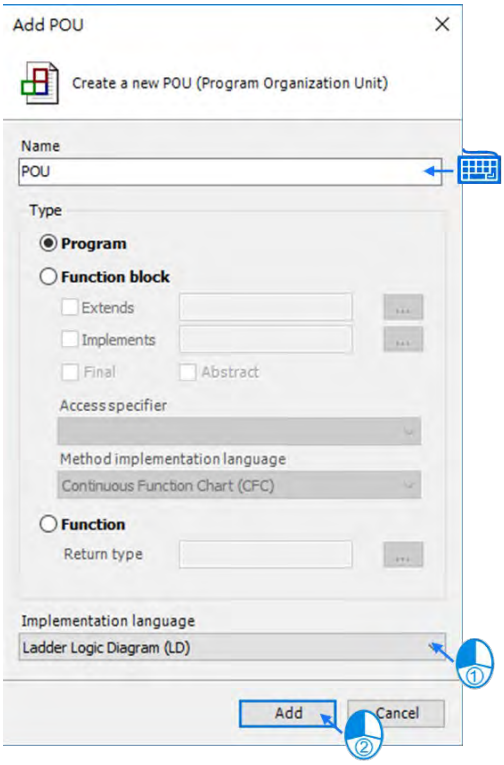
- Operating process in the example
  - "MC_Power" will be triggered and enable two servo axes (Servo ON) after executing "Step 1".
  - "MC_Home" will be triggered after executing "Step 2" to command both axes back to the home position.
  - Execute "Step 3" to trigger "MC_MoveRelative" so as to automate the action of two axes performing once for both positive and negative rotation, which means that the second axis will be triggered to rotate in positive direction after the first axis finishes its positive rotary. When the second axis finishes doing so, the first axis is triggered to perform negative direction of travel. Once the first axis finishes, the second axis starts moving in the negative direction to the end.) Finally, turn the "Step 3" to OFF.
  - The value of Counter_1 will be increased by 1 for each time positive and negative rotary completed.

- Programming with newly-created POUs

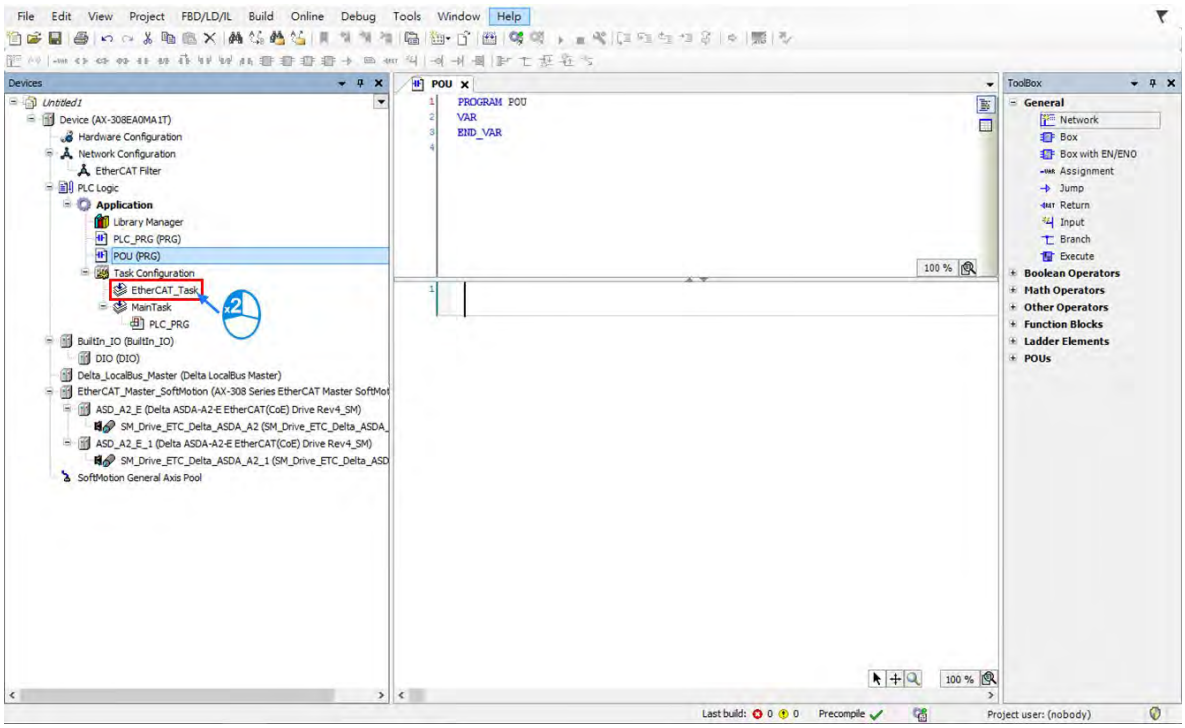
1. First, create a new POU by right-clicking “Application” to choose “Add Object” and select “POU”.



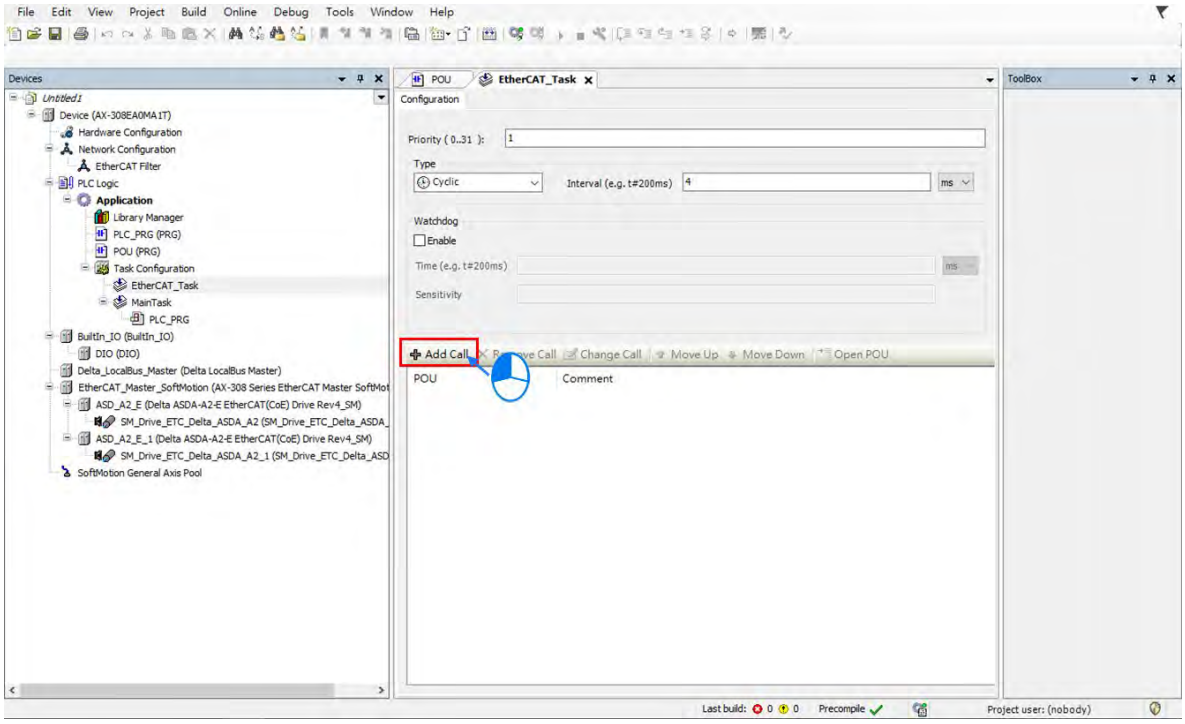
2. Input the name of the new POU in the “Name” field and choose LD language for “Implementation language”. When finished, click “Add” to add the new POU.



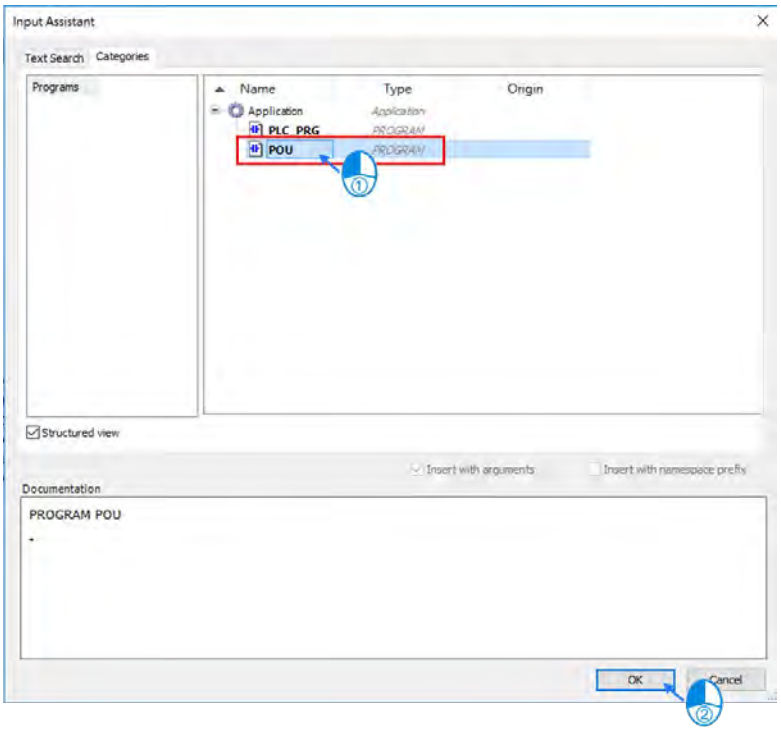
3. The newly added POU would be shown in the project tree, which needs to be added in Task by double-clikcing “EtherCAT_Task”. (Function blocks related to axis motion needs to be established in EtherCAT_Task to ensure normal operation of axes.)



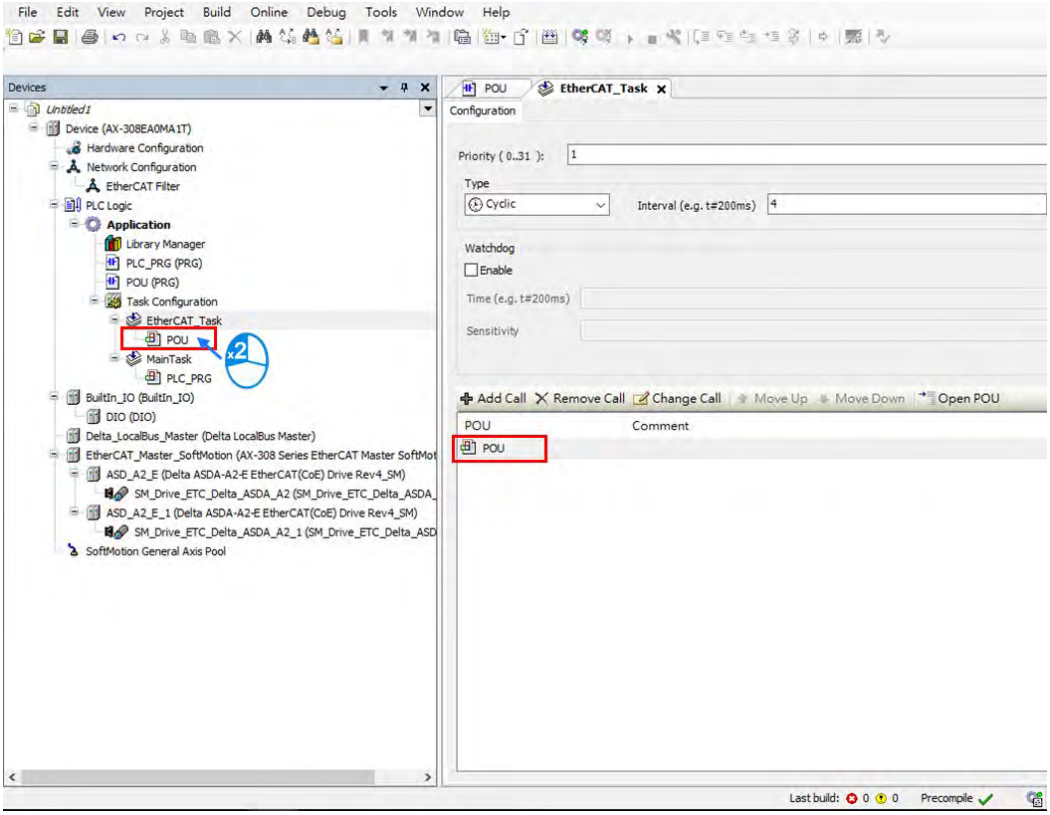
4. Click “Add Call” on the EtherCAT_Task page.



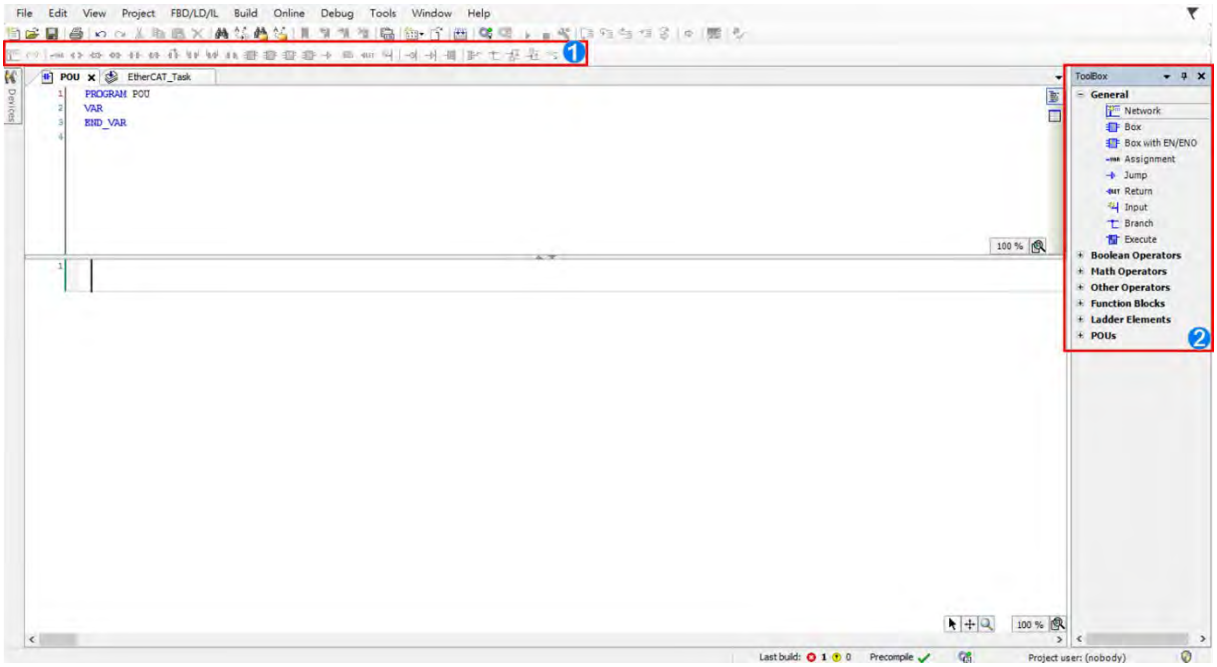
5. Click “OK” after choosing the newly-created POU.



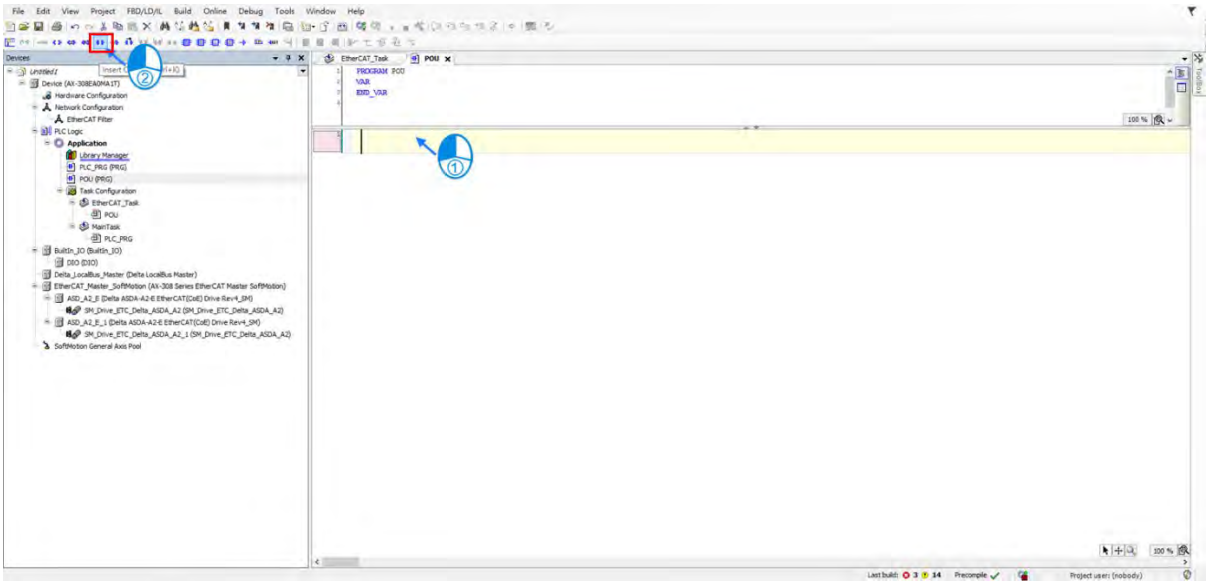
6. After adding the POU in EtherCAT_Task, “POU” will be shown on the lower half of the EtherCAT_Task page as well as in the project tree. Then click the POU from the tree, which is under the category of EtherCAT_Task.



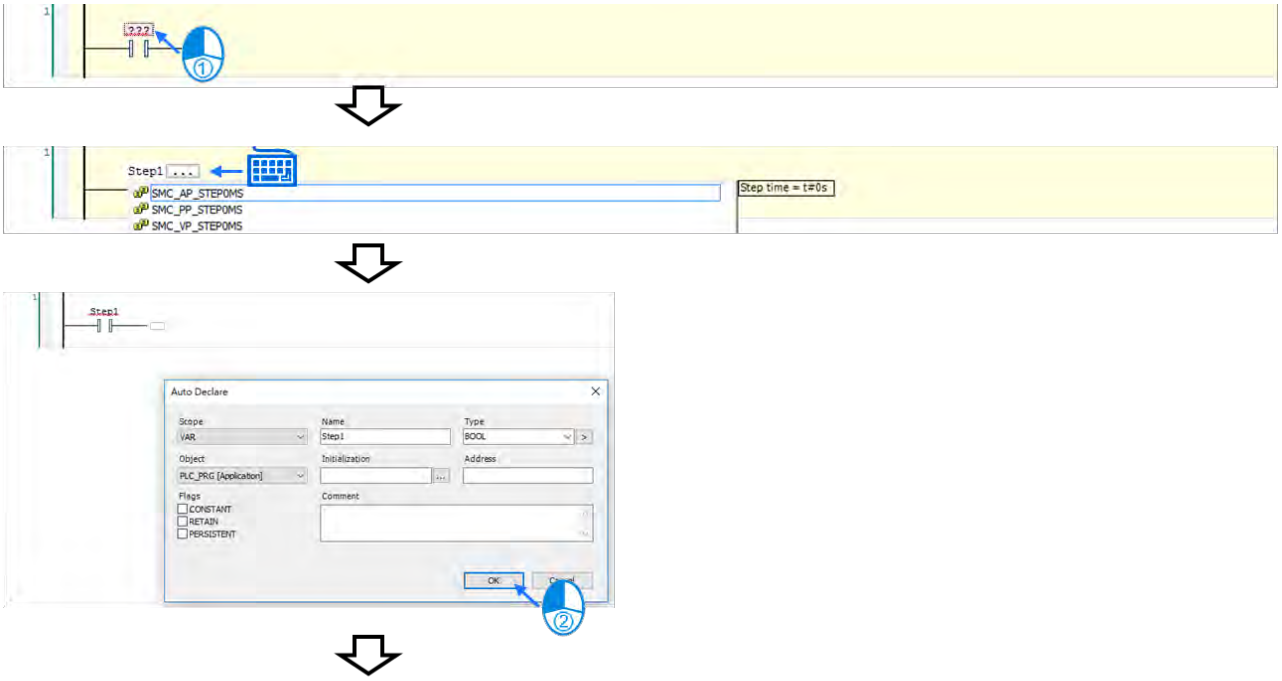
7. With ladder logic programming language, add new commands by using the red-circled field marked with 1, while field 2 is for adding required Function or Function Block.



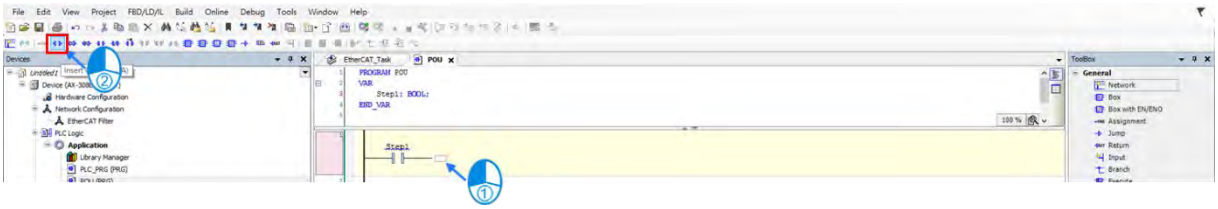
8. Click on the desired place to add new commands on POU page, then select .



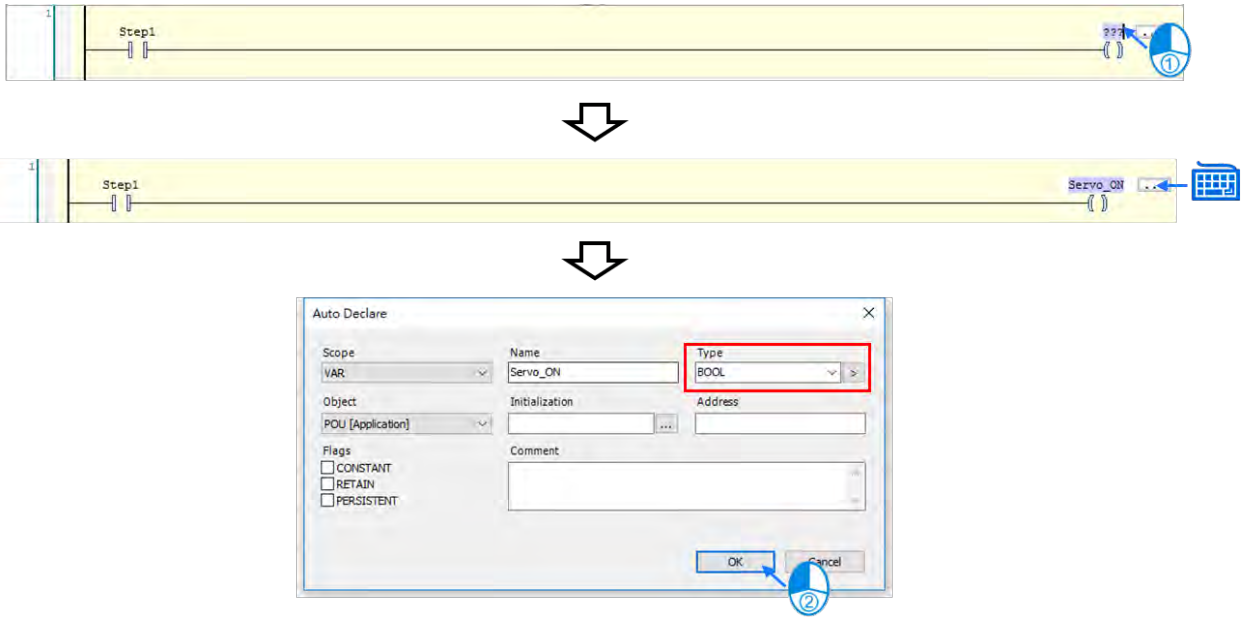
9. After the selected command is shown in the POU, click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.



10. Then add the ouput instruction by clicking on the desired place and choosing  .

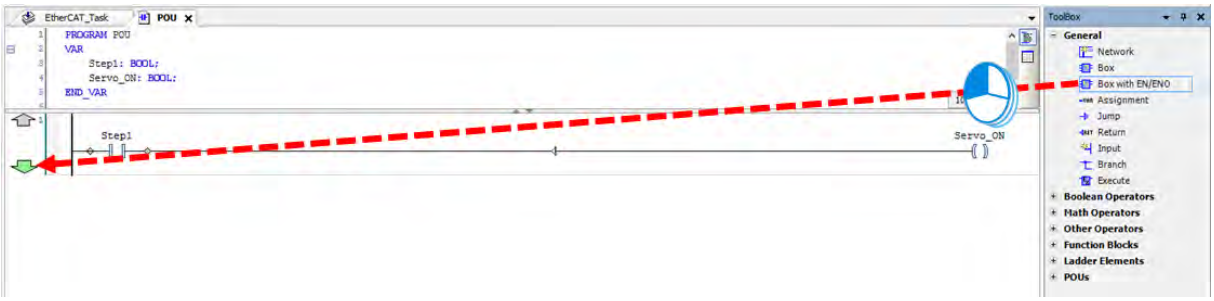


11. Click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.

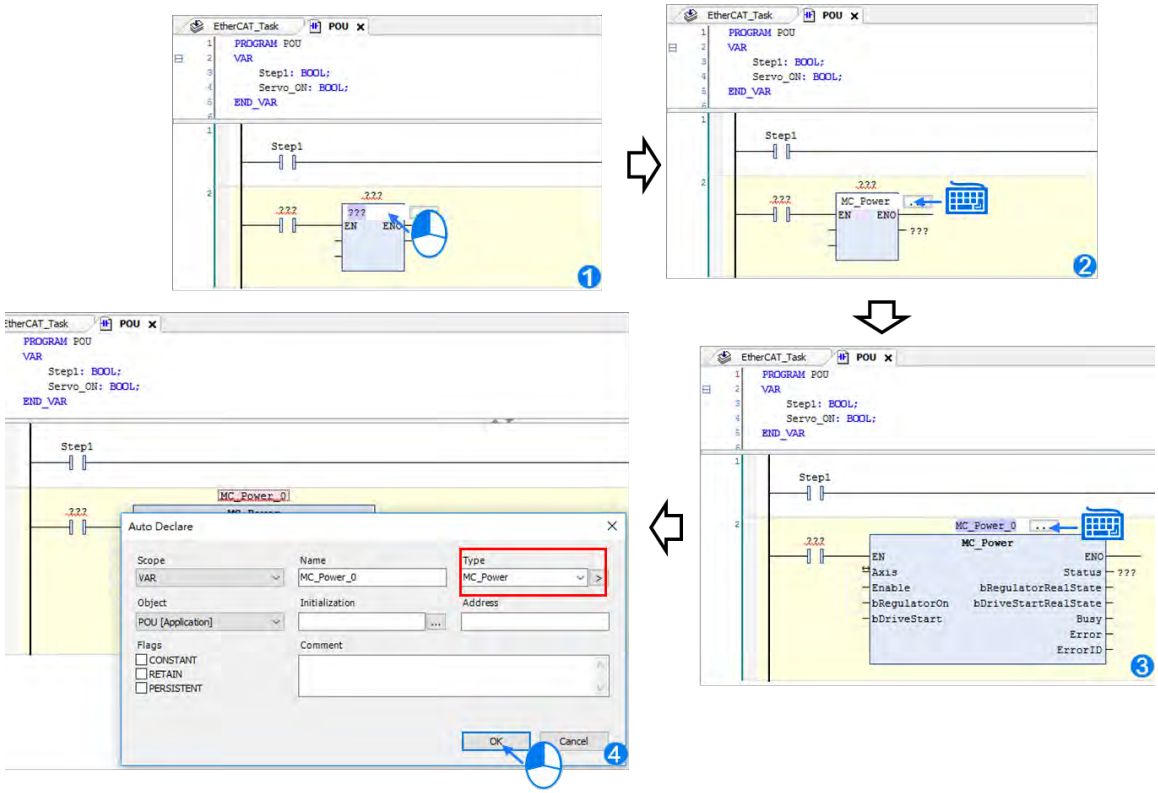


12. The instruction names you have added previously will be displayed as well as the instruction types, where the name and type of the instruction can also be added directly. (Same operation for the examples in later discussion.)

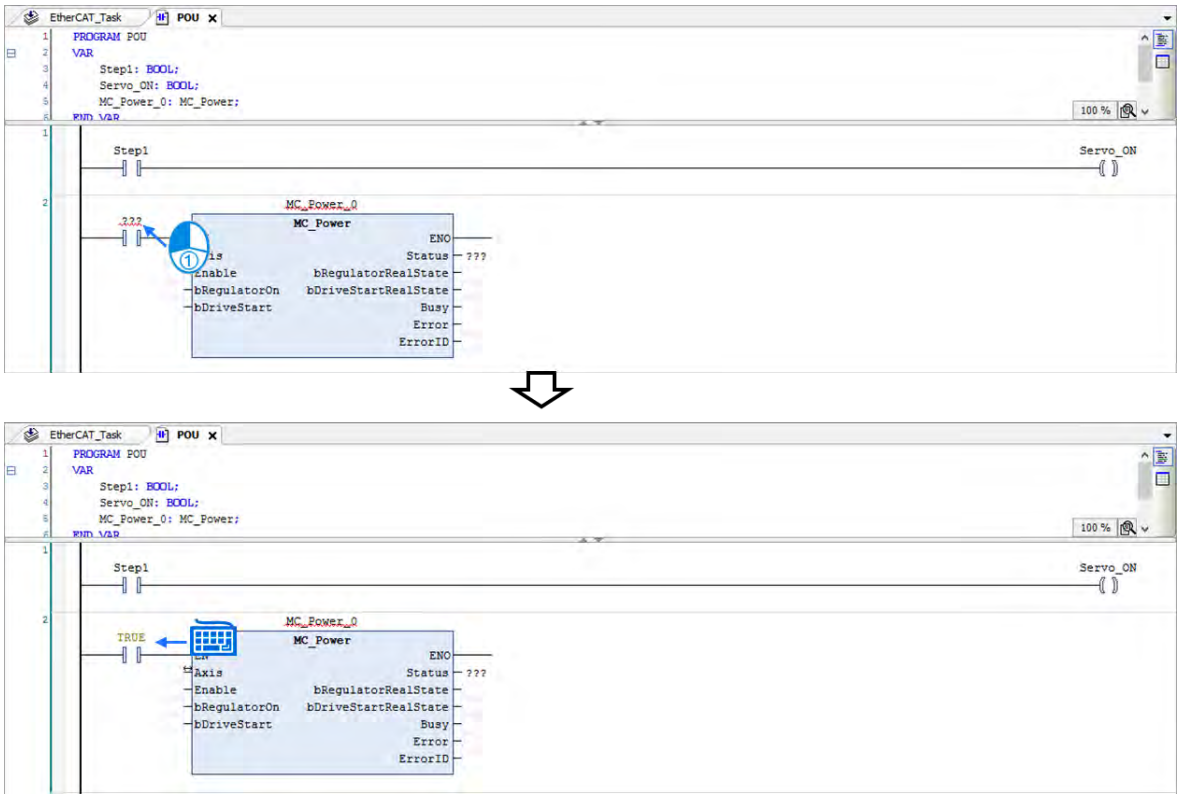
13. To add new function blocks, click and hold “Box With EN / ENO” then drag to  .

14. 

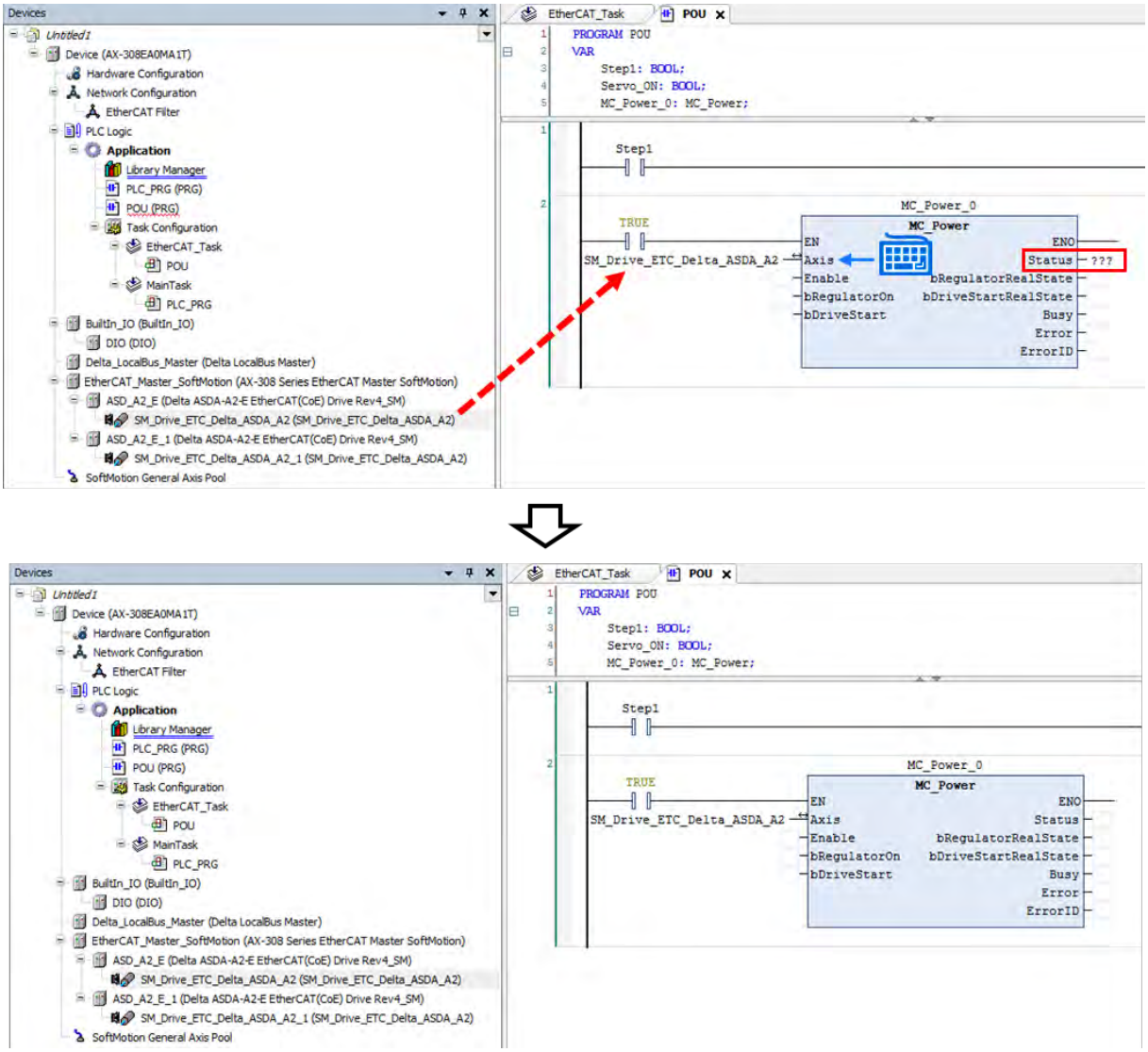
Click “???” and insert the name of function block then press ENTER to display the function block. Insert the name and press ENTER key again, then “Auto Declare” window will pop up. Click “OK” after confirming the “Type” of function block is correct.



15. Continue to click “???” and insert “TRUE”, then press the ENTER key.

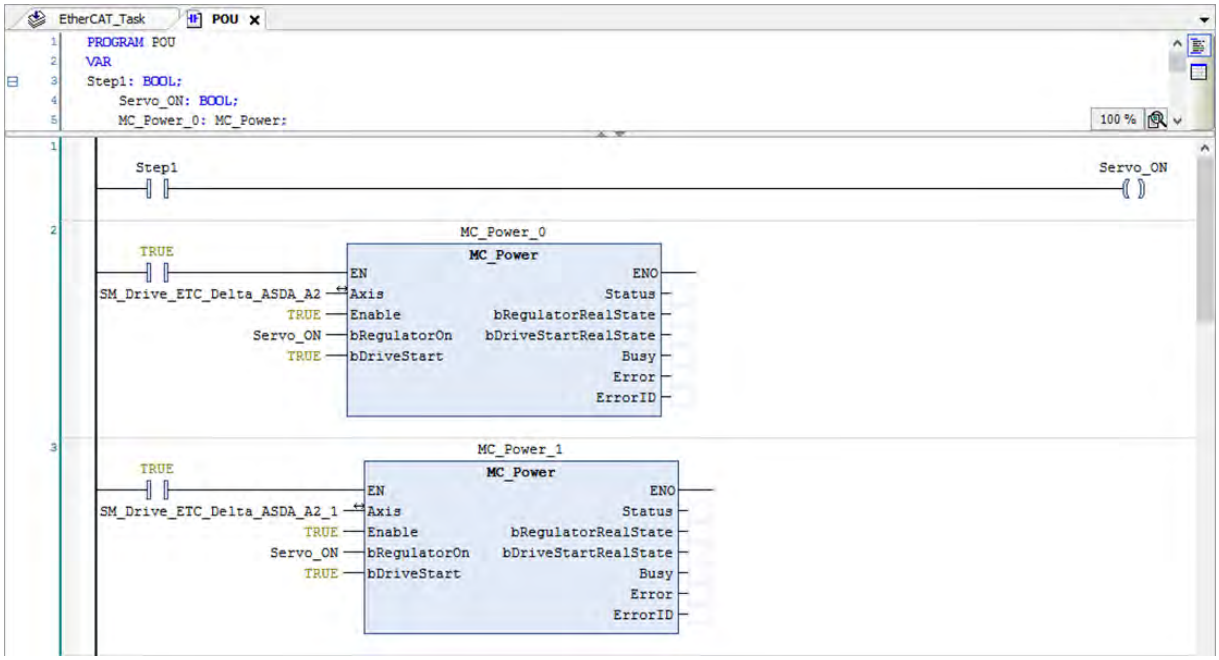


16. Insert the name of servo drive to the Axis input pin of the function block and delete “???” of other pins, since the mark “???” does not represent any variables and errors may occur while parsing.

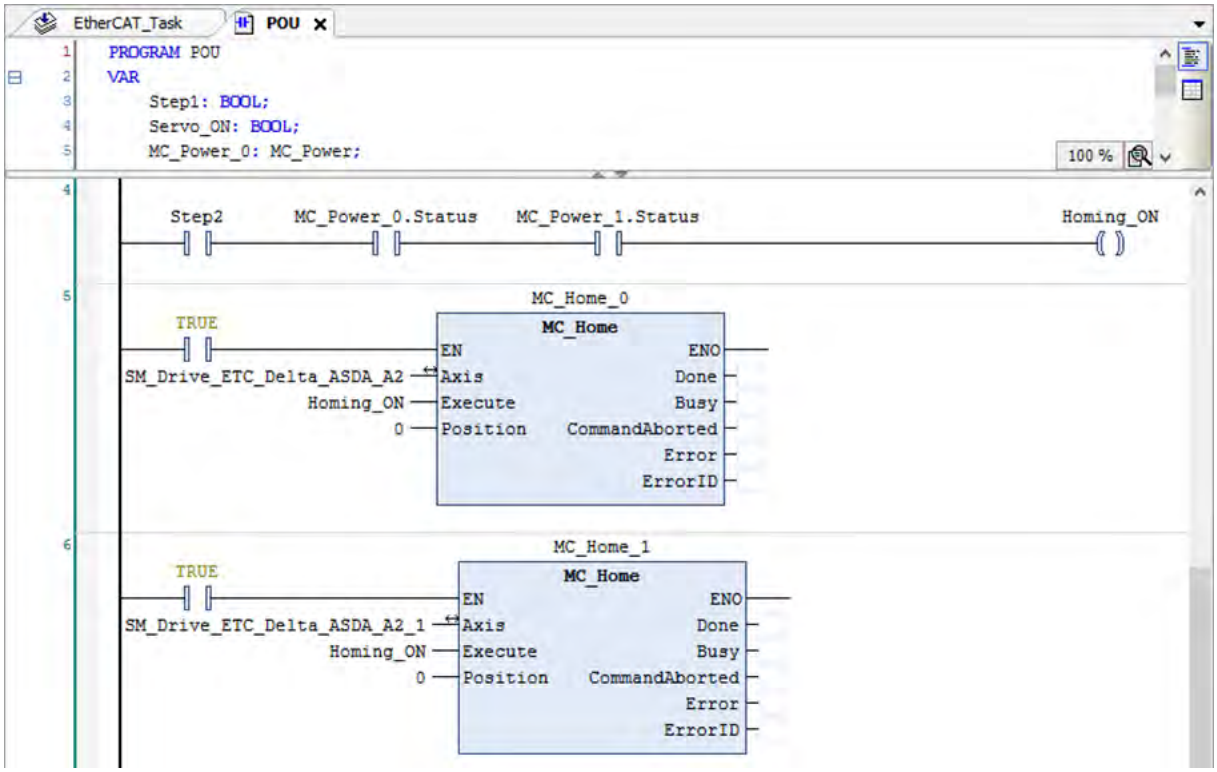


17. Complete the following programming example with the same operation process.

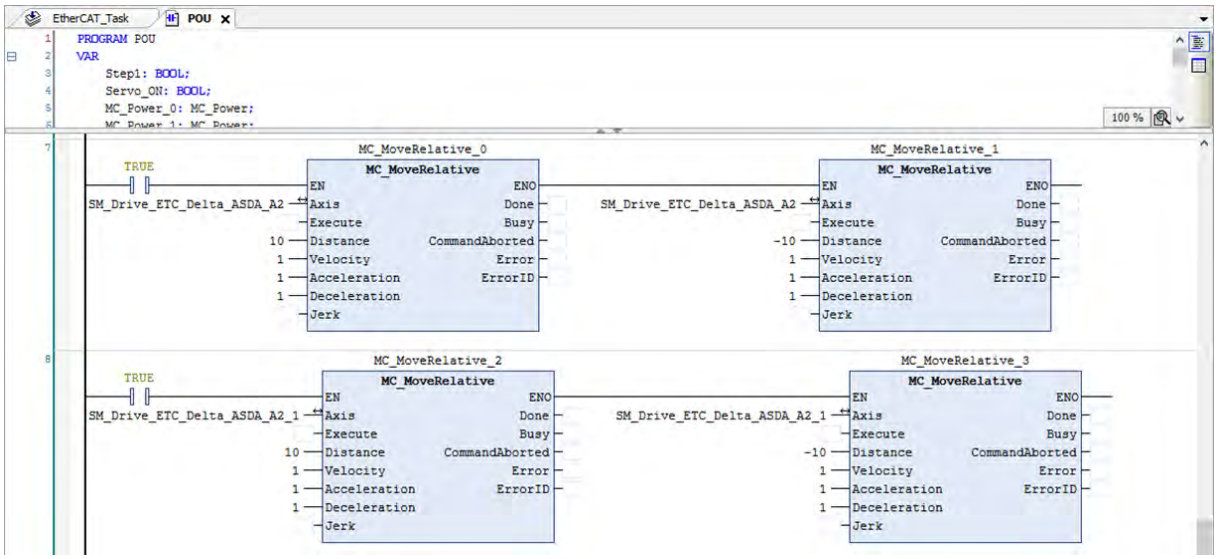
- Create function blocks of MC_Power for both servo axis.



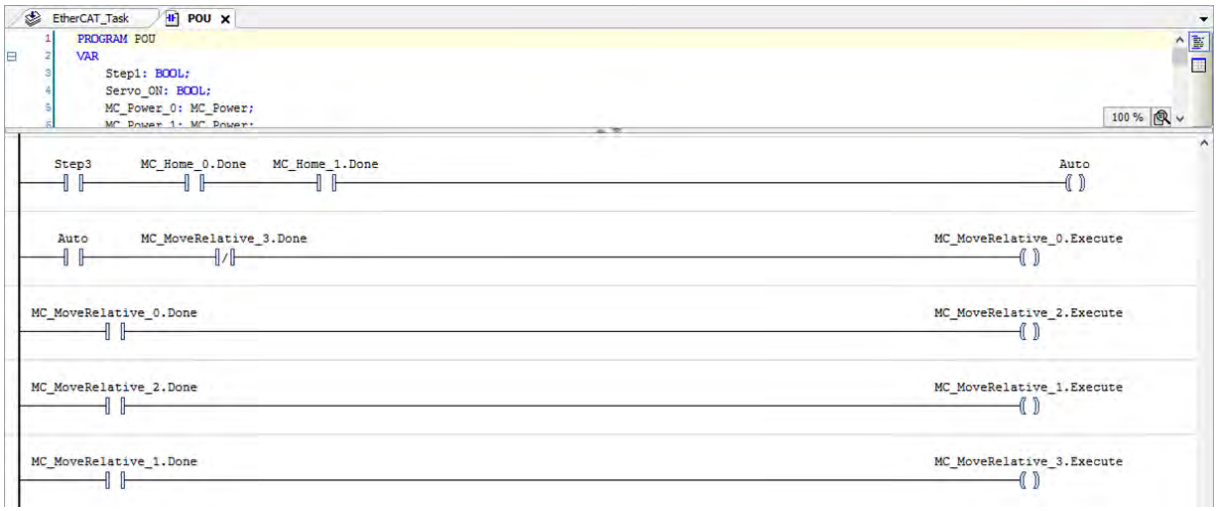
- Continue to create MC_Home function blocks with additional condition for Execute, which MC_Home can be executed only if the axis status is Servo ON.



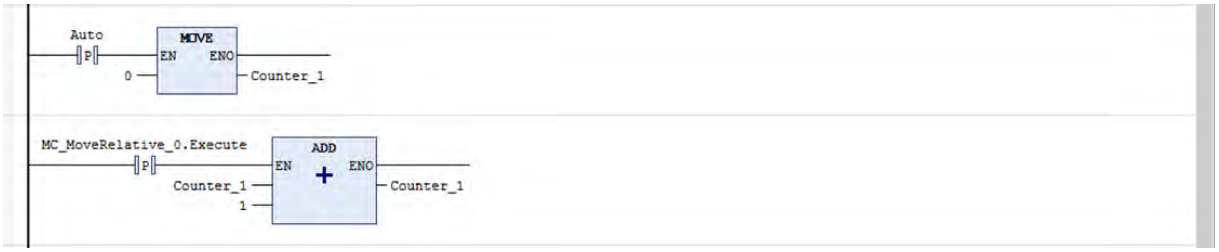
- Then create MC_MoveRelative function blocks to let both axes able to be switched between the left and right positions.



- Create program to perform positive rotation of axis 1 & 2 then change to negative direction by means of the function blocks (MC_MoveRelative) created earlier.



18. The value of Counter_1 will be returned to zero when execute AUTO. As soon as the positive and negative rotation being performed once, increase the value by one automatically.

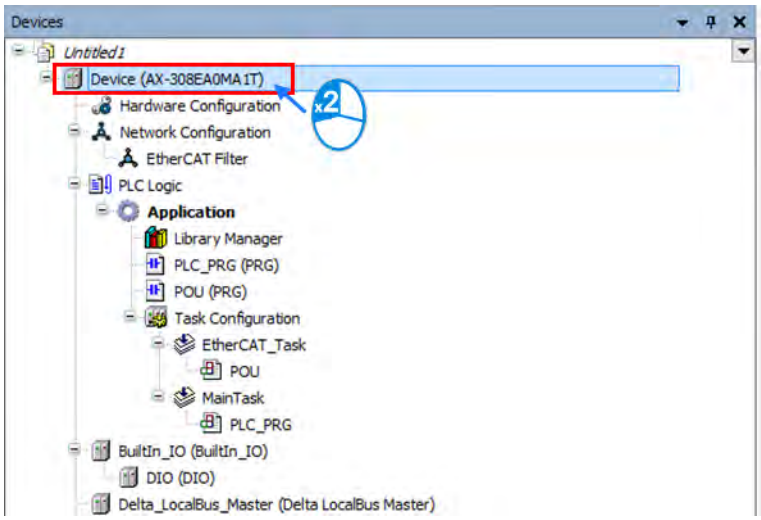


3.2.3 Program Monitoring

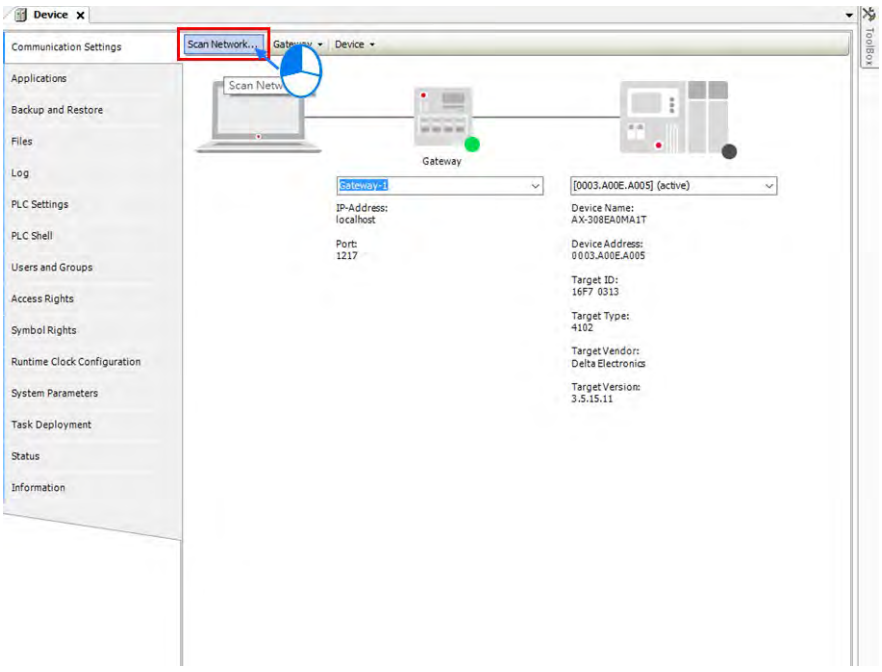
This section is concerned with the monitoring operation via the software DIADesigner-AX. When running a program, the current control status of system can be monitored and part of the device values are allowed to be modified for system testing as well.

3.2.3.1 Setup Connection between Devices

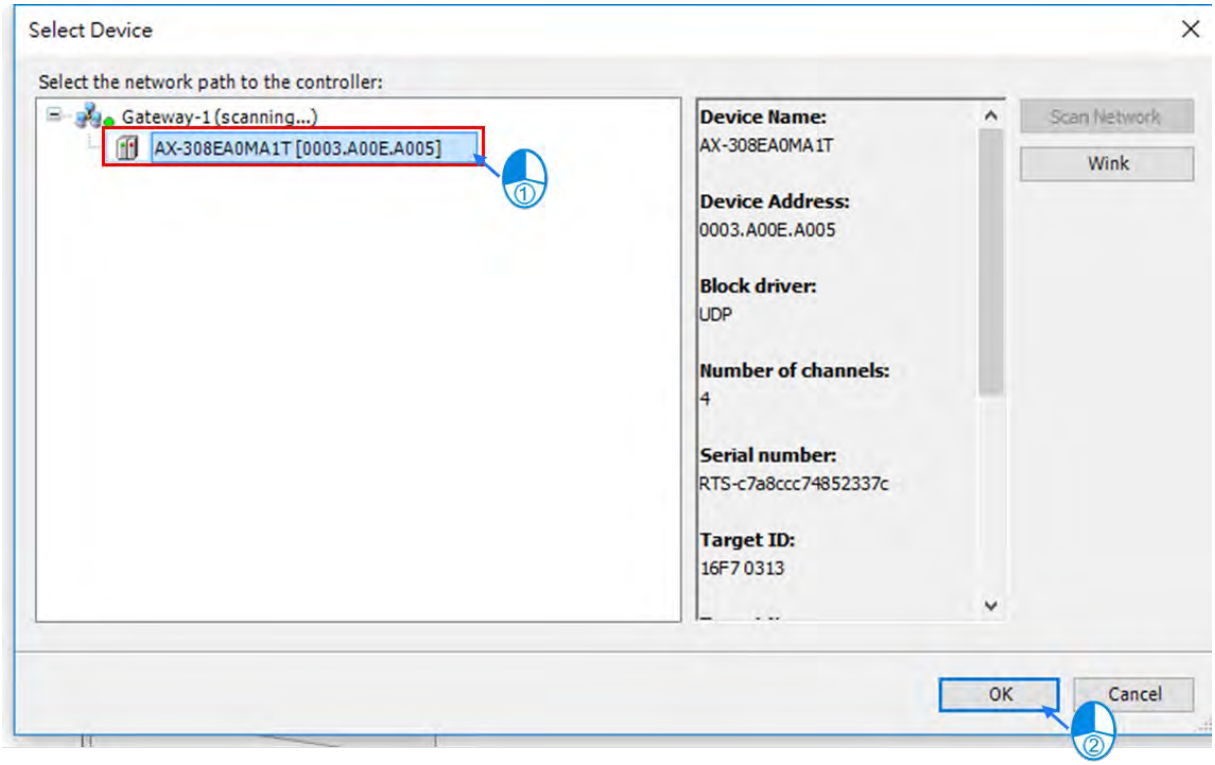
1. Double click on “Device”. (The default IP address for AX-308E is 192.168.1.5)



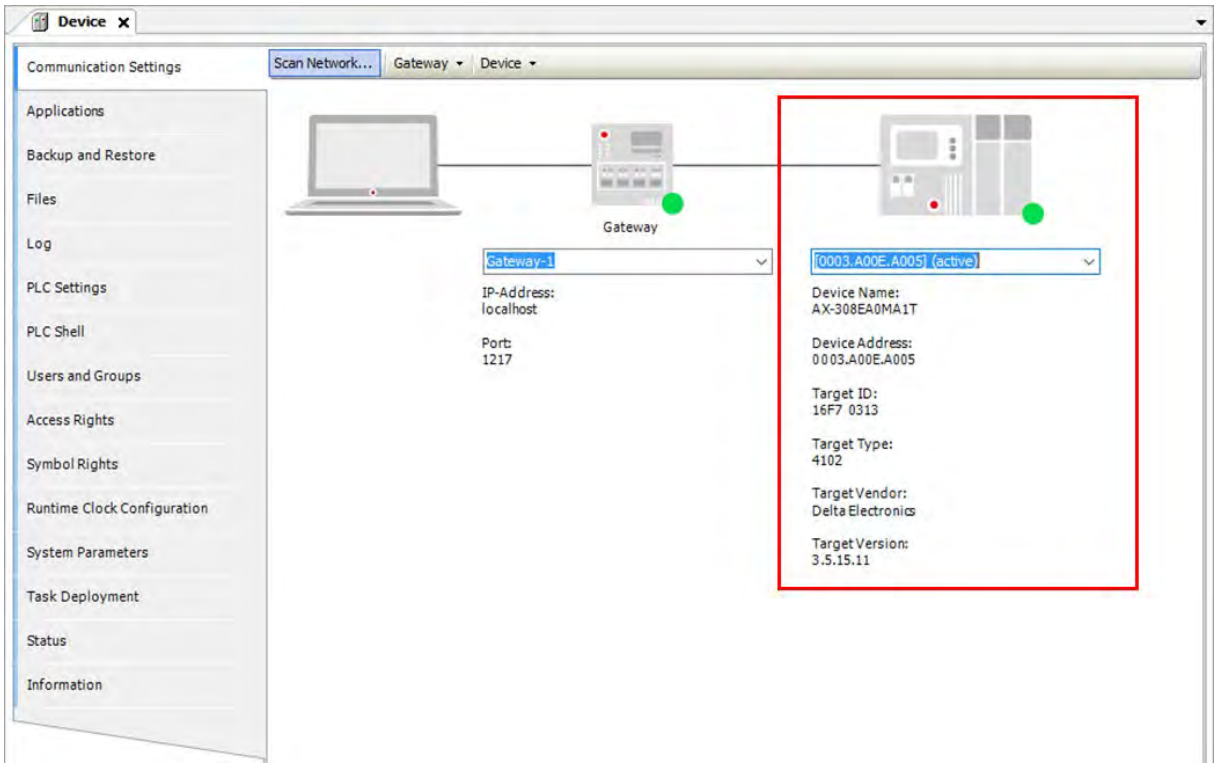
2. Choose “Scan Network” after entering the “Device” page.



3. After the “Select Device” window pops up, choose “AX-308E” and click “OK”.

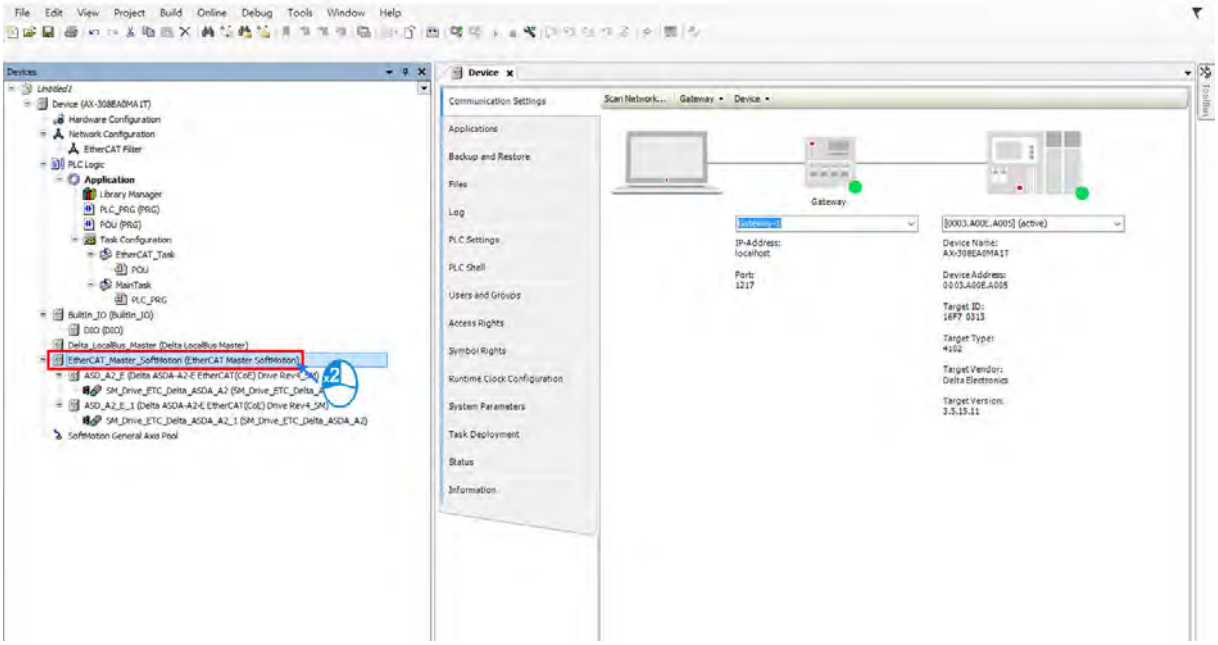


4. Once the connection between the PC and the device is established, information of the device will be displayed as shown in the following figure marked by the red box.

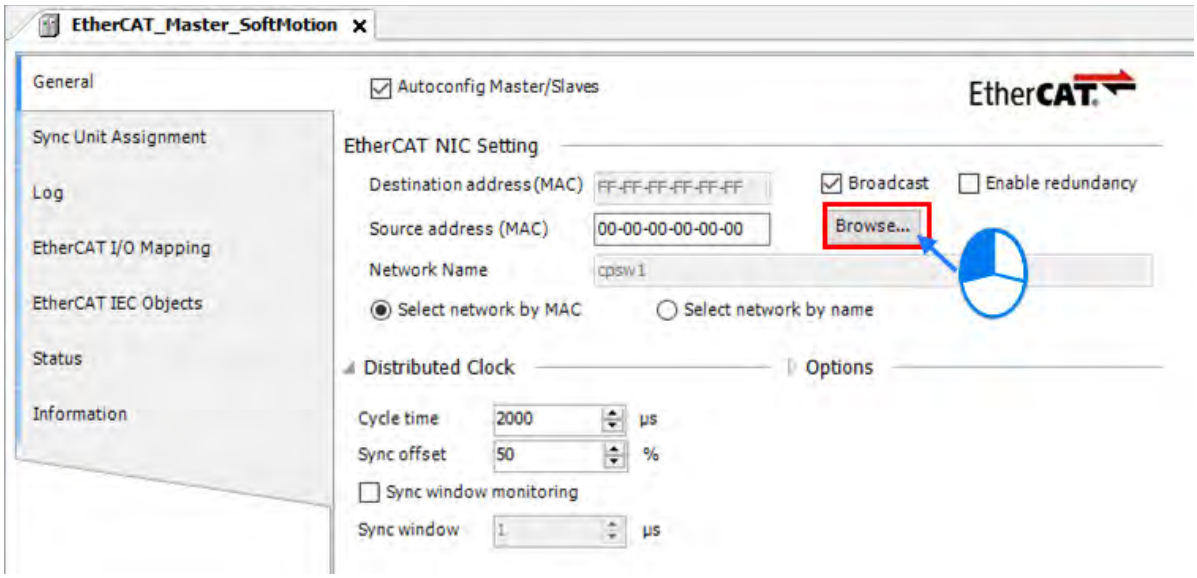


3.2.3.2 Process Monitoring and Control

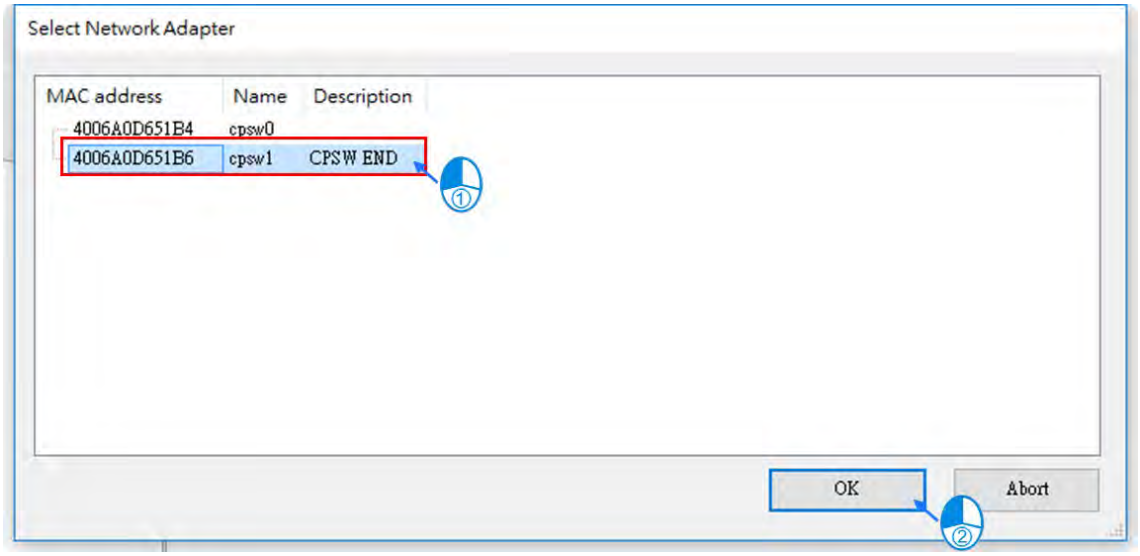
1. The correctness of EtherCAT communication should be checked before program monitoring starts. After confirmation, double click on “EtherCAT_Master_SoftMotion”.



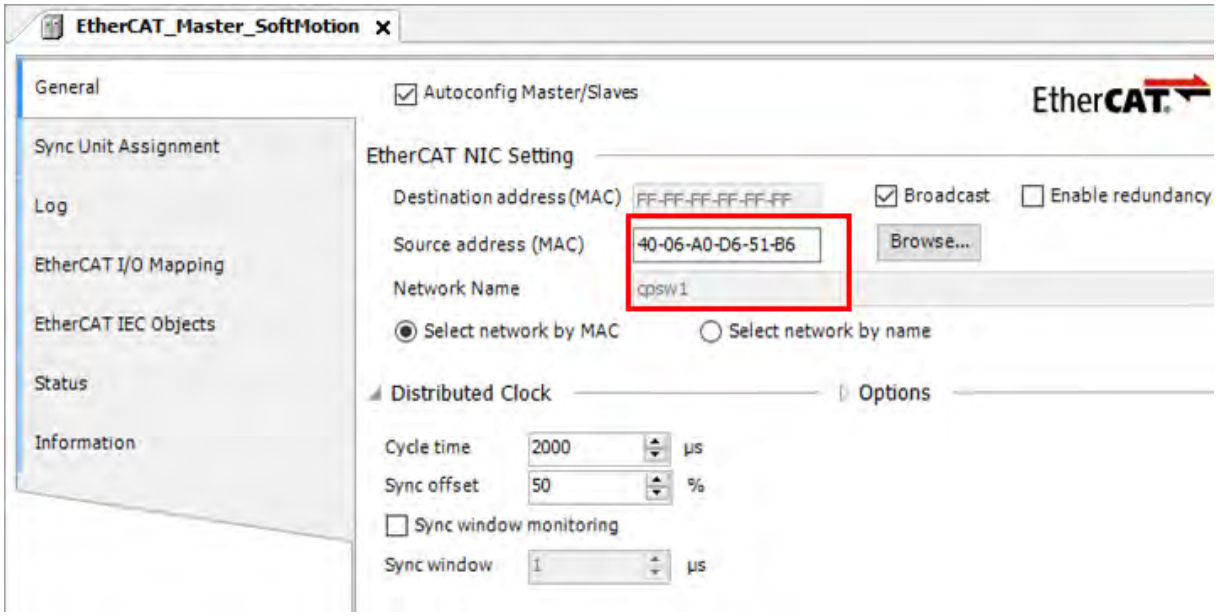
2. Choose “Browse” after entering the EtherCAT_Master_SoftMotion page.



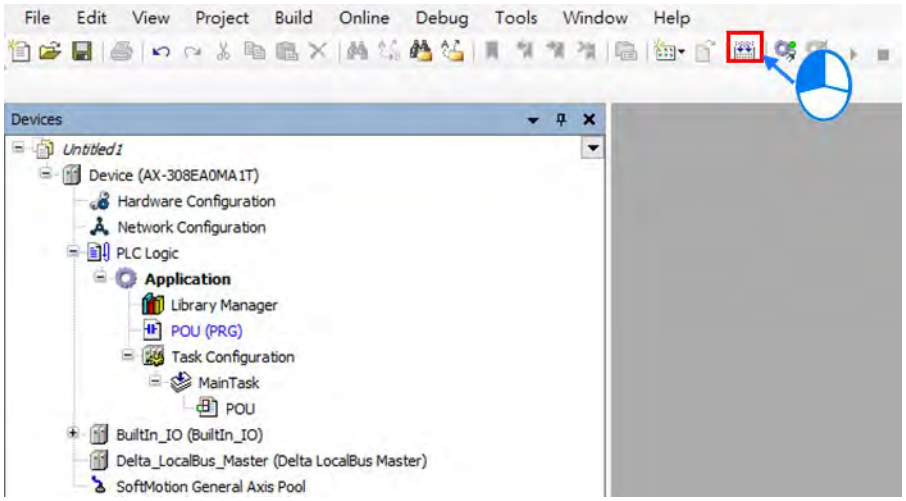
3. Select EtherCAT port “cpsw1” and click “OK”.



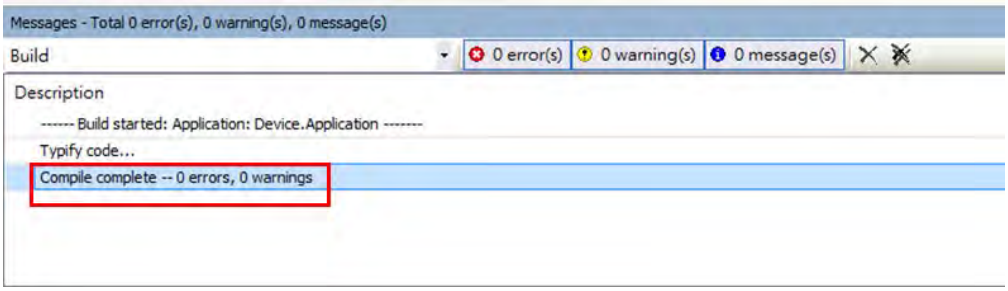
4. After finish choosing network adaptor, the setting value of Source address (MAC) is displayed.



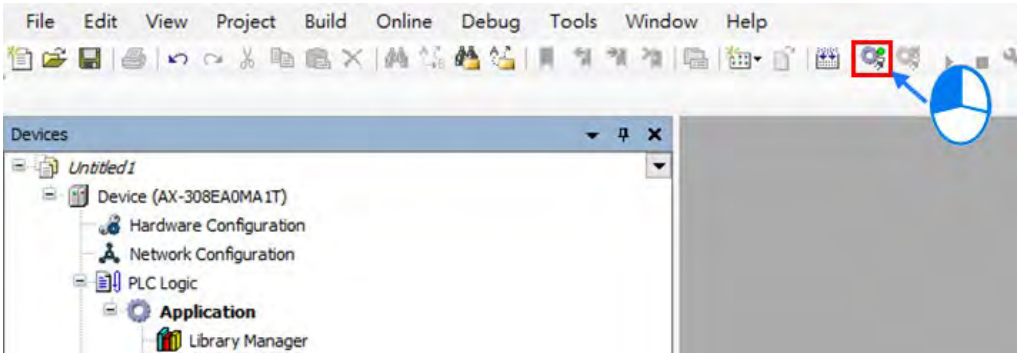
5. Click on the Compile button to verify the correctness of program.



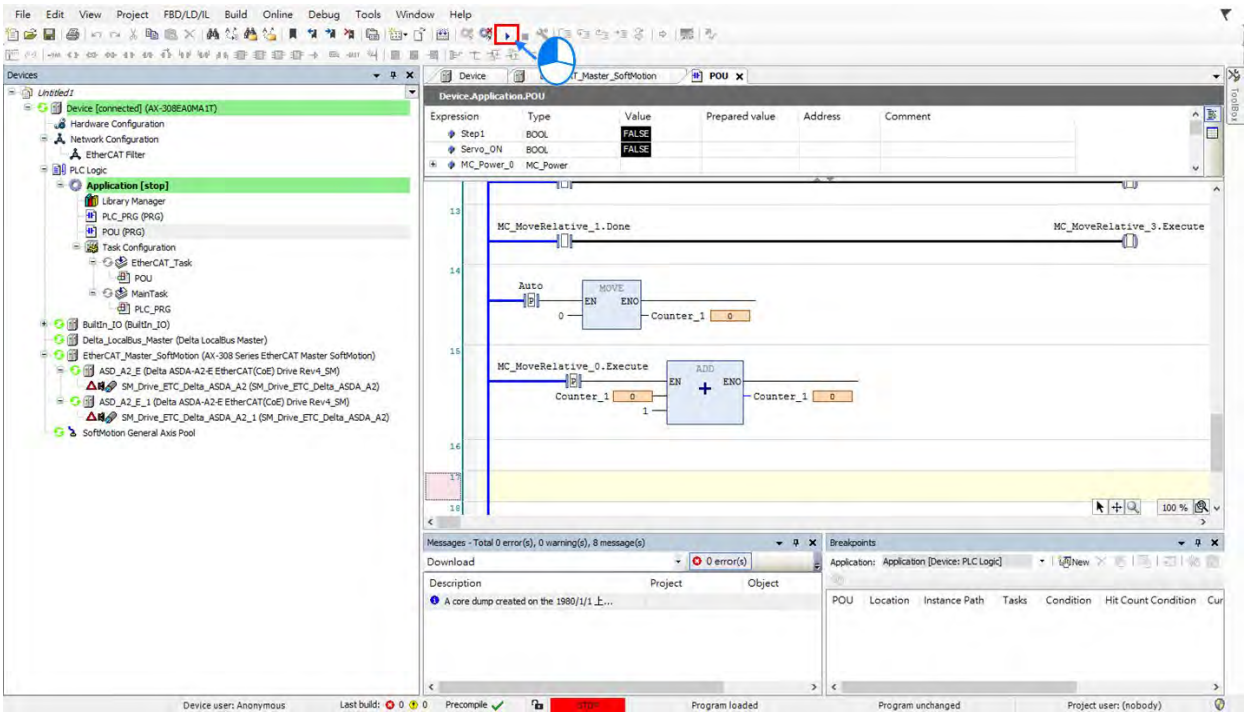
6. Upon completion of compiling, a report message of programming errors and warnings is displayed.



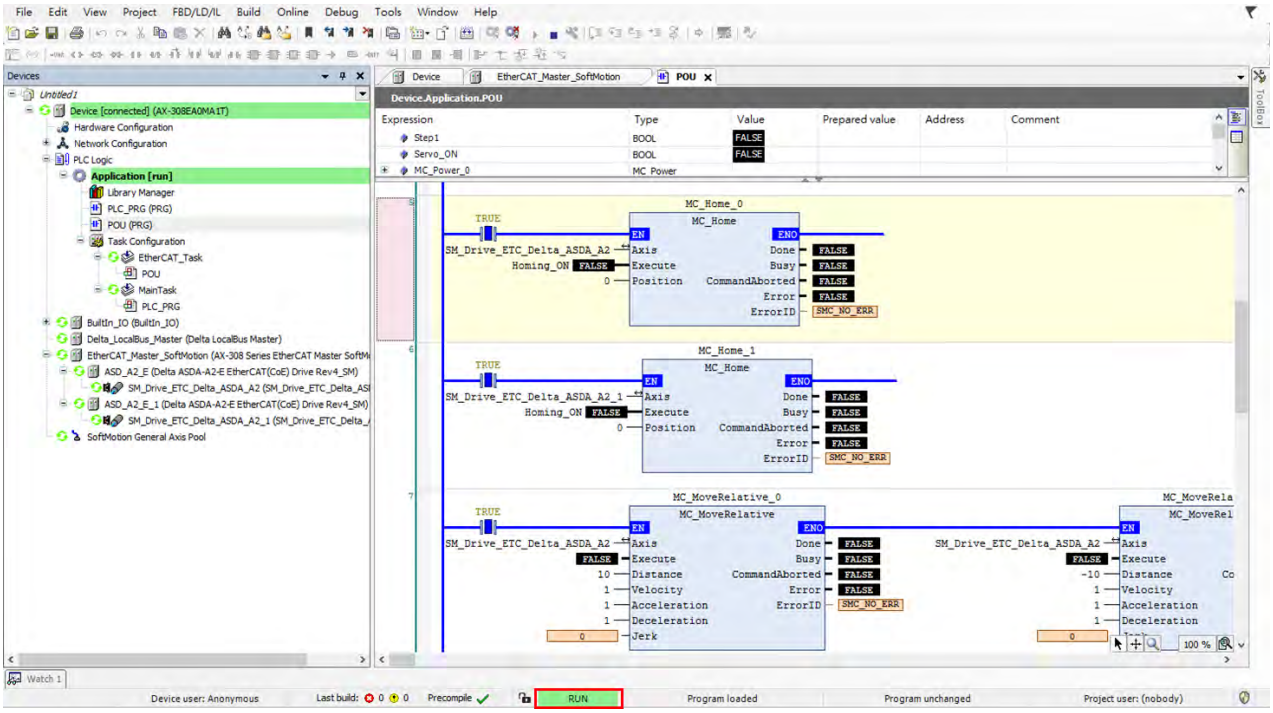
7. And then click on to perform online monitoring.



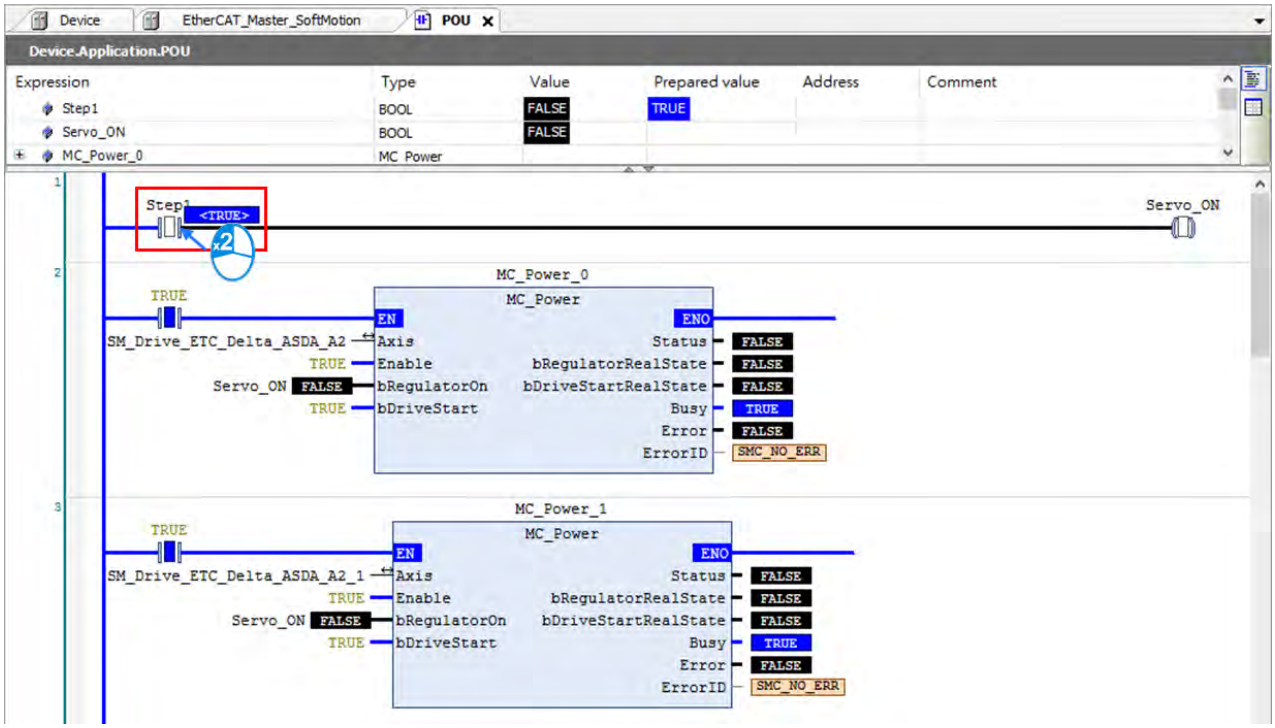
8. The program needs to be downloaded after monitoring action is performed. When the download is completed, the status of PLC shows STOP until you click on and the status would shift to RUN.



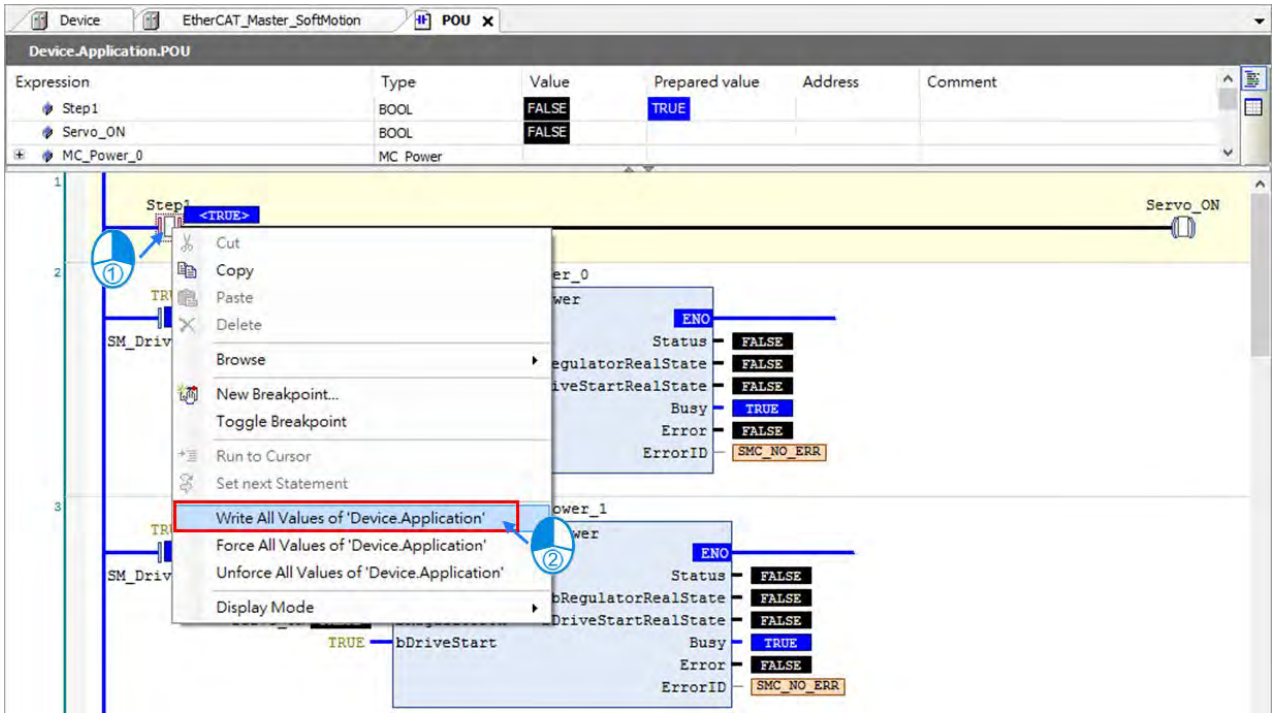
9. When the PLC is in the RUN mode, the status would be displayed as RUN at the bottom of the page and the program is ready to be executed.



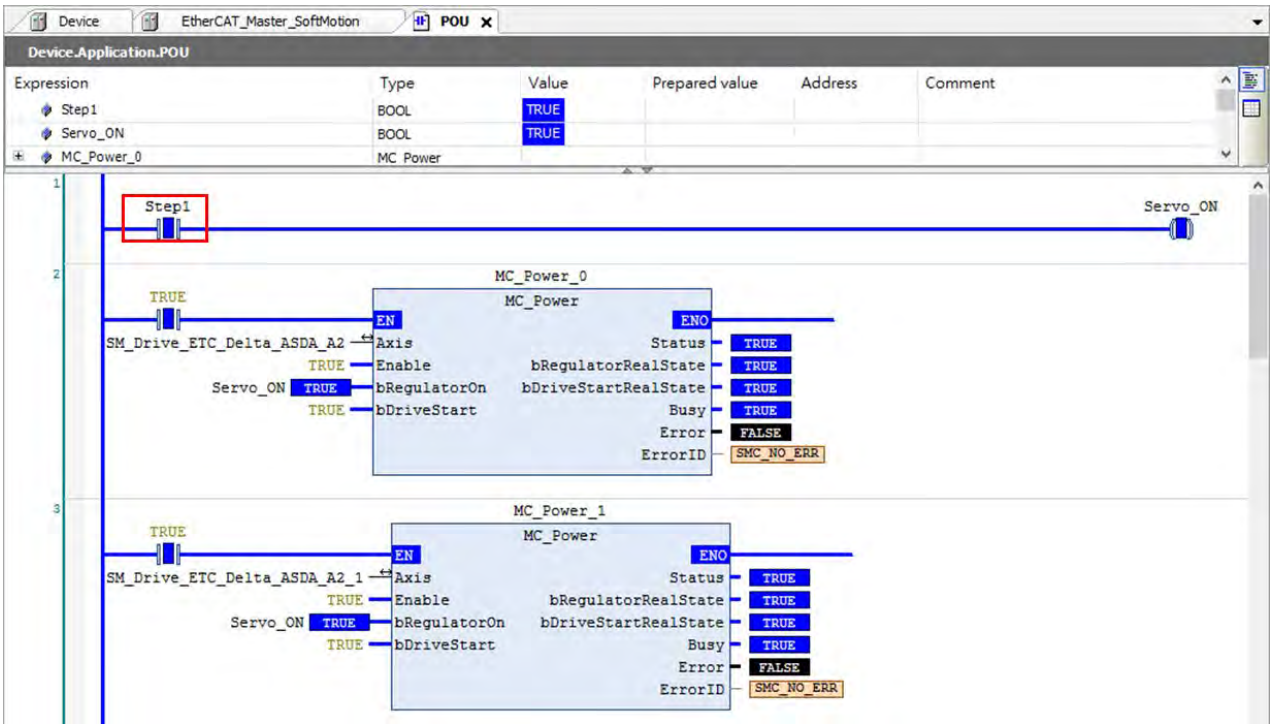
10. Double click on “Step 1” and the icon <TRUE> will be shown.



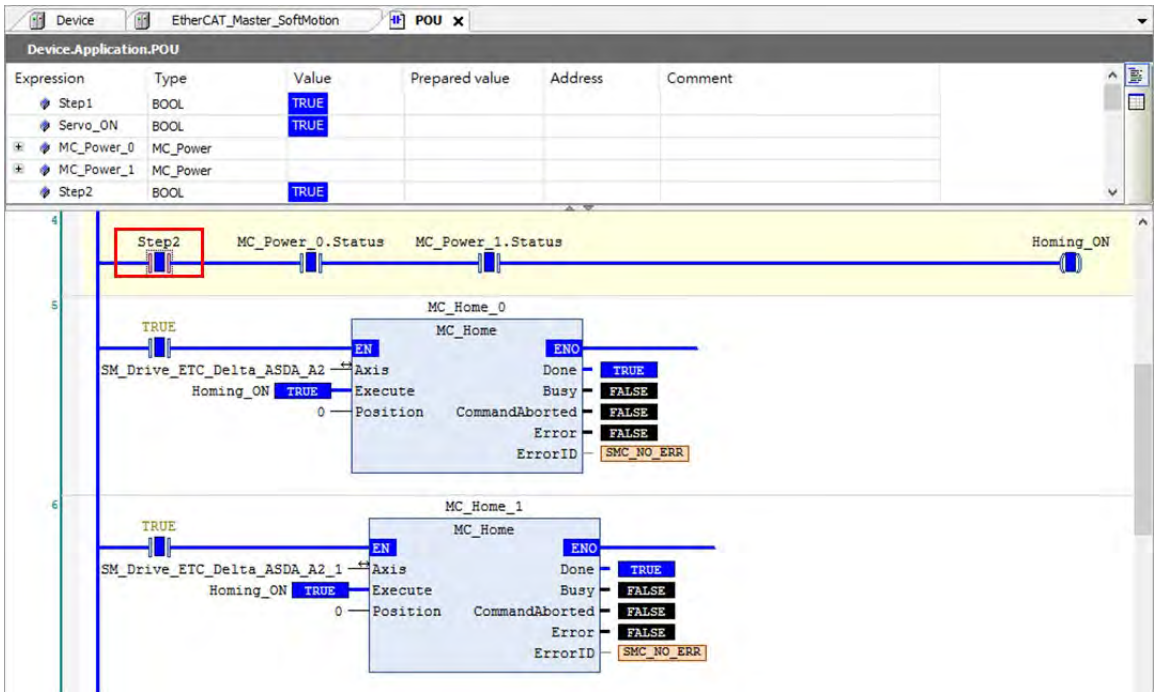
11. Right click on “Step 1” to select “Write All Values of ‘Device.Application” and all the corresponding status will be written in, which can also be performed by using the shortcut key **Ctrl + F7**.



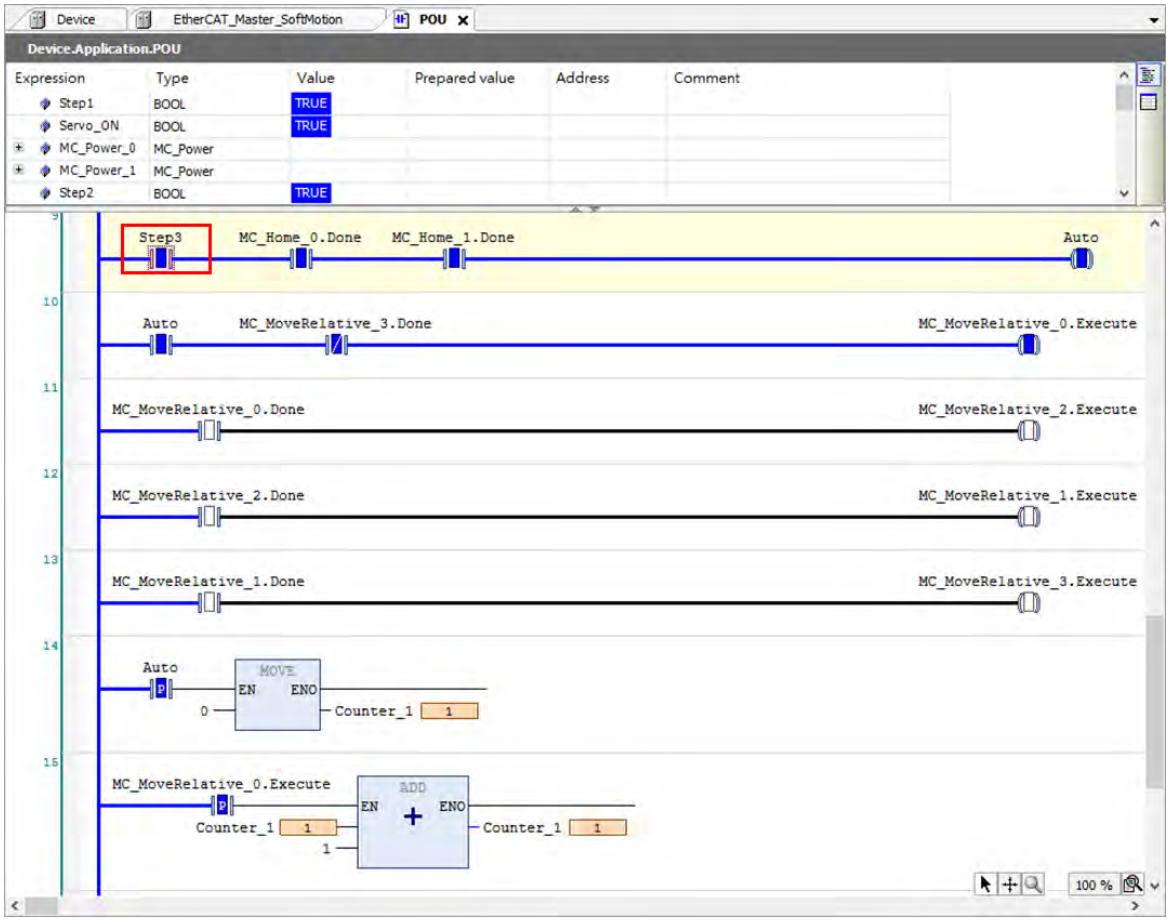
12. After activating “Step 1”, axis 1 and 2 are under Servo ON mode. (Same operation to execute on the following programs.)



13. Start homing for axis 1 and 2 by activating “Step 2”.

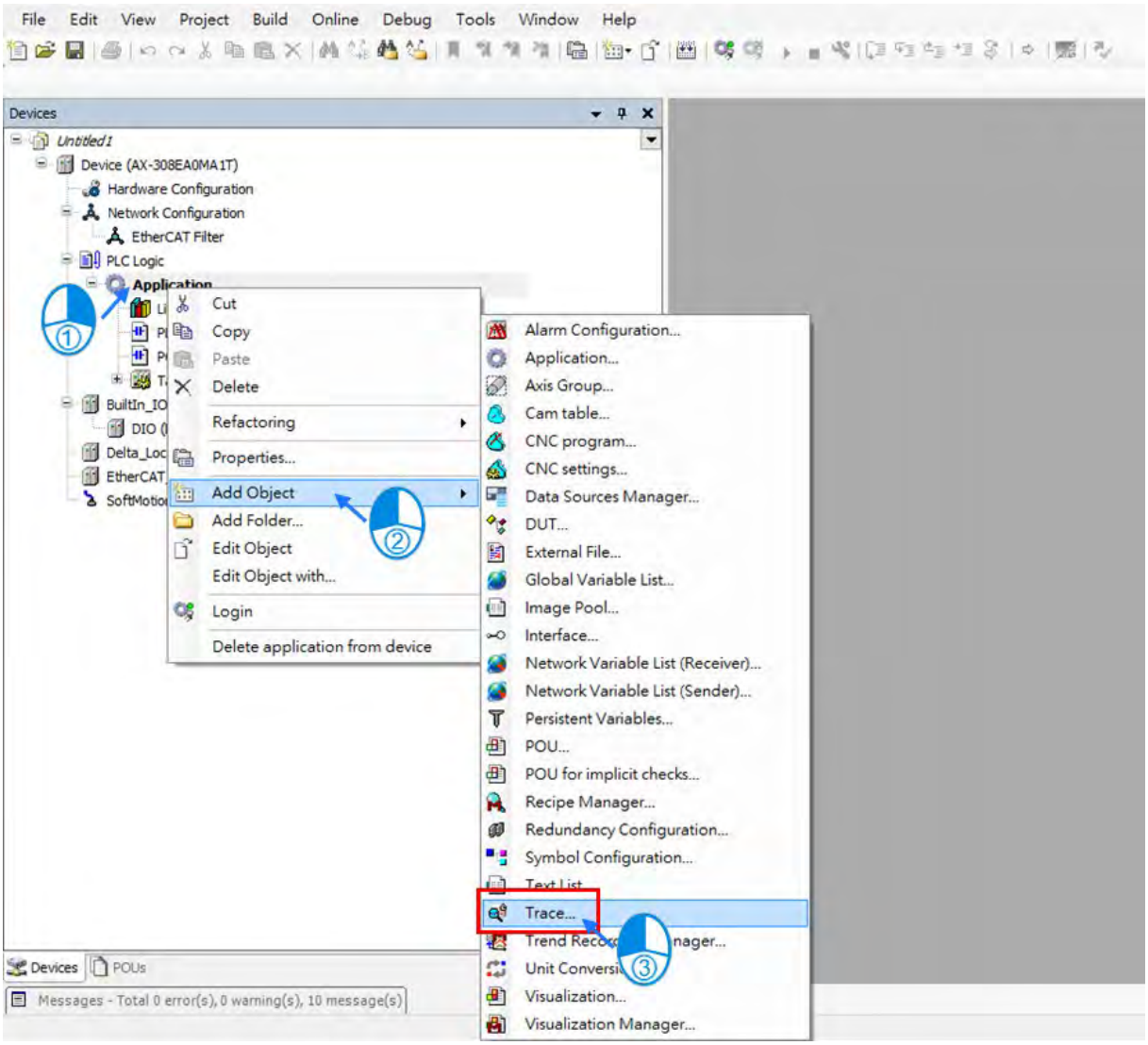


14. After starting "Step 3", axes rotate in positive direction then reverse, axis by axis. In addition, the value of Counter_1 will be increased by 1 for each time positive and negative rotary completed, until “Step 3” being turned off.

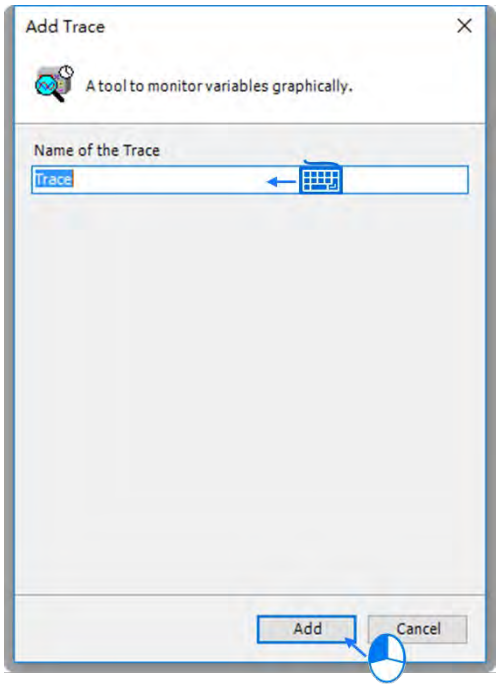


3.2.3.3 Monitor with Trace Tool

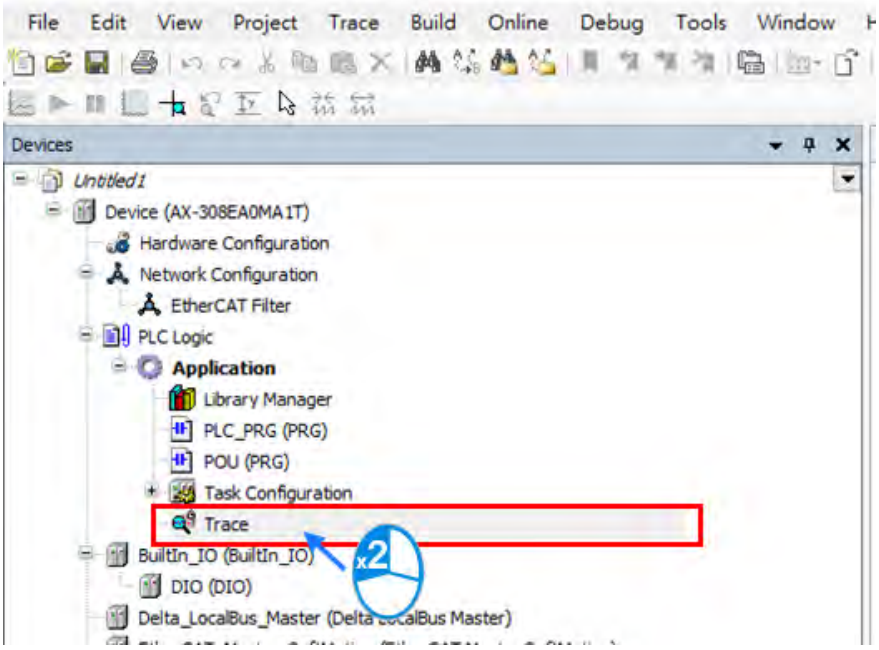
1. A Trace makes it possible to record the value history of variables on the PLC, just like a digital sampling oscilloscope. To monitor the status of input/ output pins and axes with Trace function while executing the program, navigate to “Application” > “Add Object” > “Trace”. Then you’ll open the “Add Trace” window.



2. Insert the name of Trace then click “Add” on “Add Trace” window.

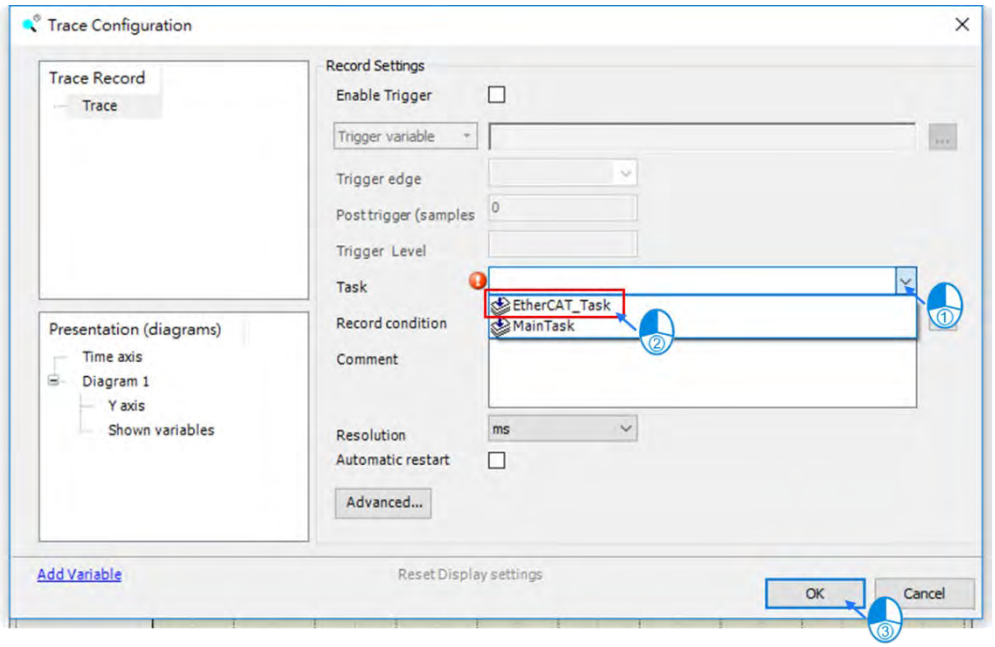


3. After the Trace is successfully added, it will be shown in the project tree on the left side of the screen. Then double click on “Trace” to open the Trace page.



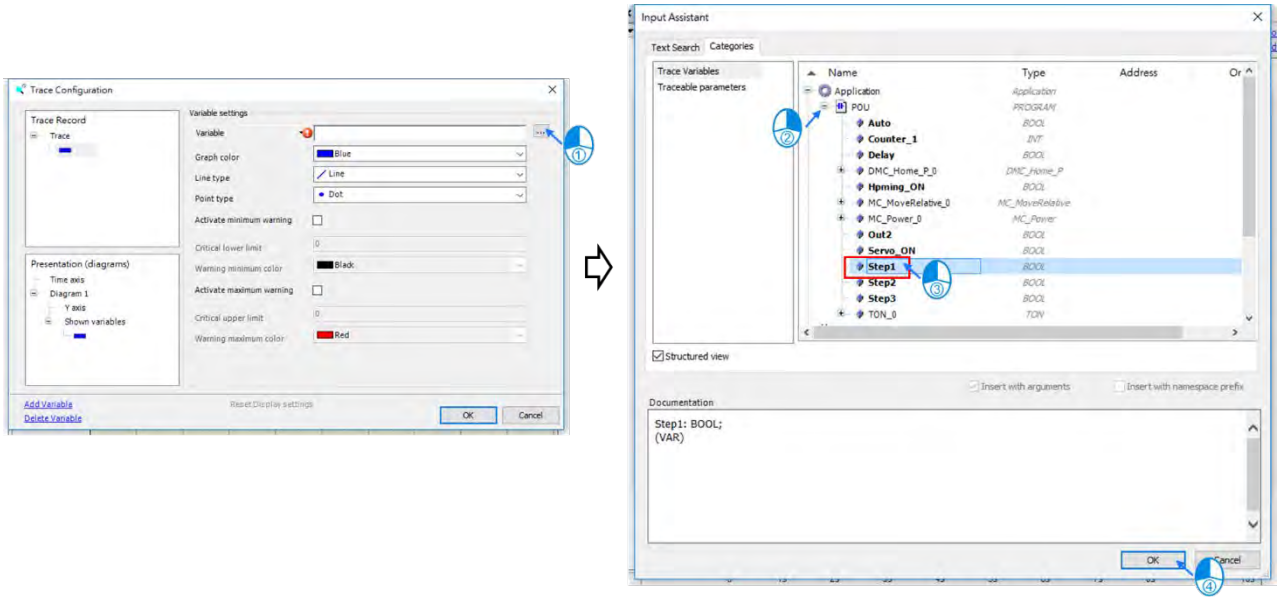
4. Click on “Configuration” on the Trace page to open the Trace Configuration window.

5. Select “EtherCAT_Task” from the drop-down list of Task on the Trace Configuration window.

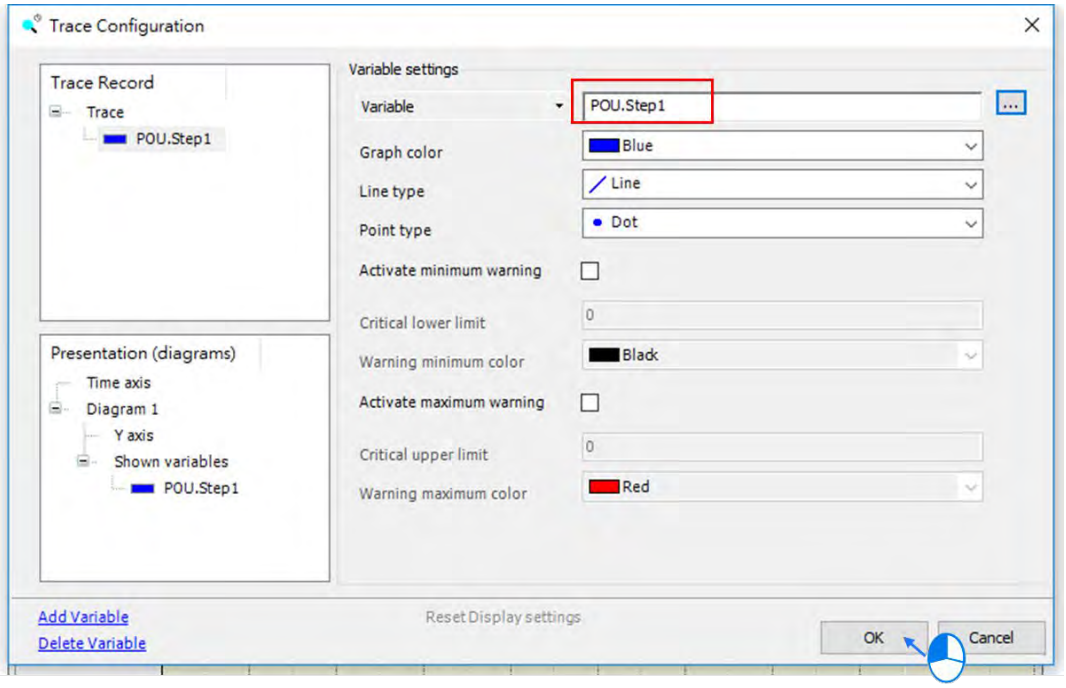


6. After finishing configuration, click “Add Variable” on the Trace page to open the Trace Configuration window.

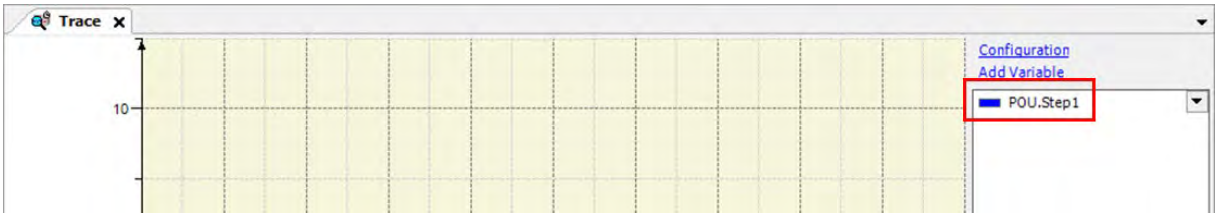
7. Click the button ... on the Trace Configuration page to add the required trace variables or traceable parameters. After the Input Assistant window is opened, expand POU on the tree and select “Step 1” then click on “OK”.



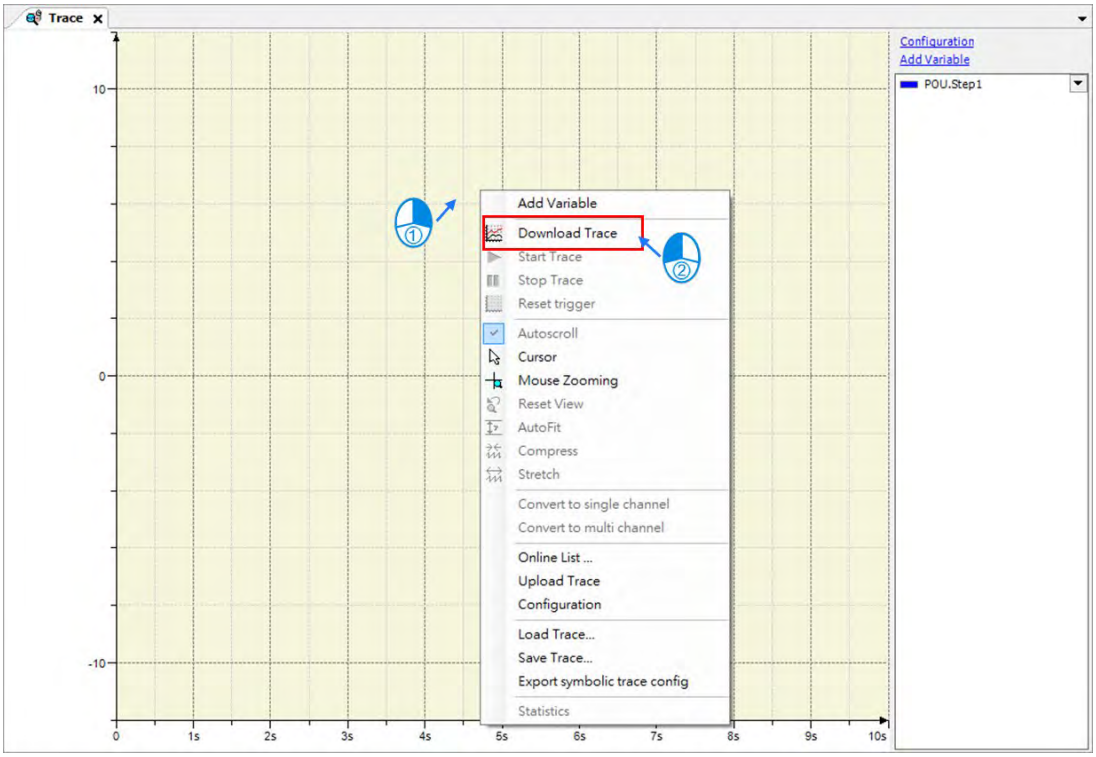
8. Make sure the chosen variable is shown in the variable field of variable settings and then click “OK”.



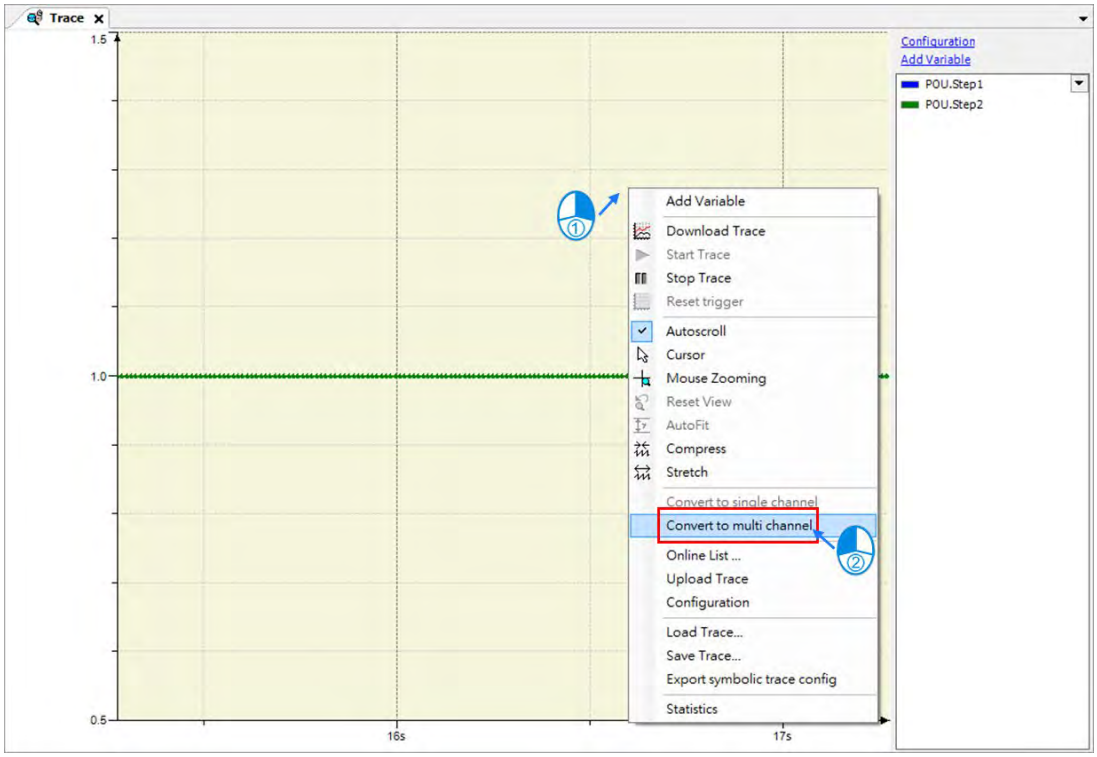
9. Afterwards, the added variable will be shown on the right side of the Trace page.



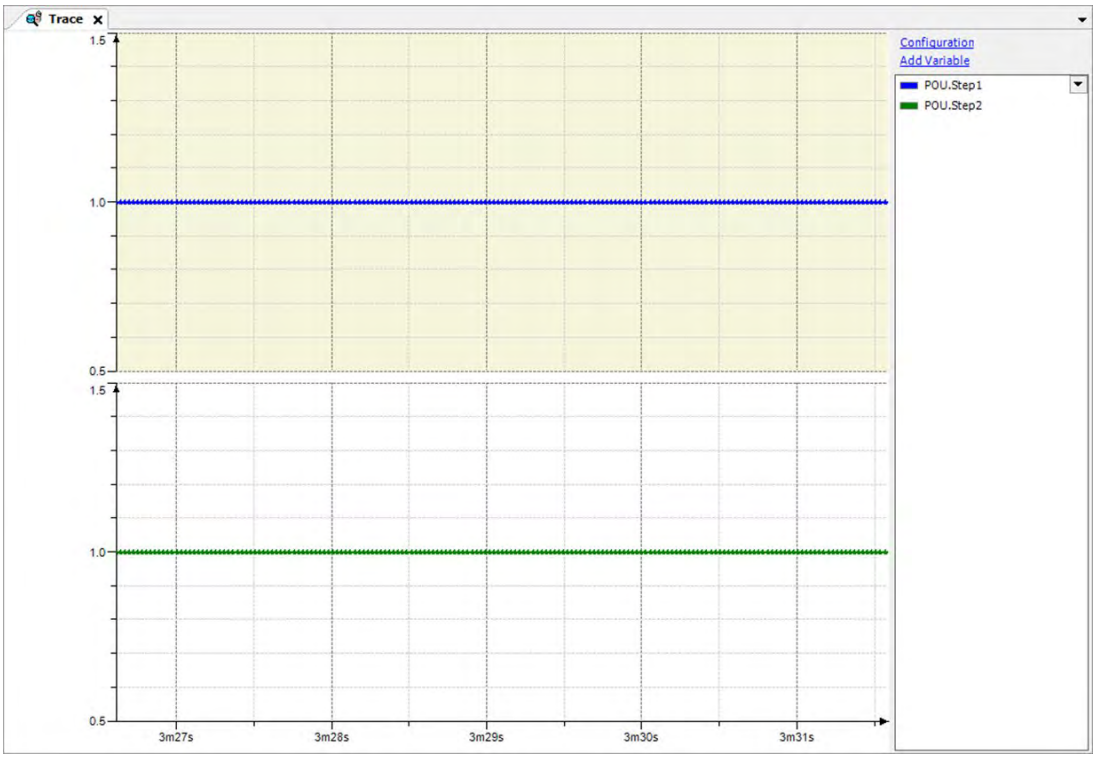
10. Right click on the Trace and select “Download Trace” to start monitoring.



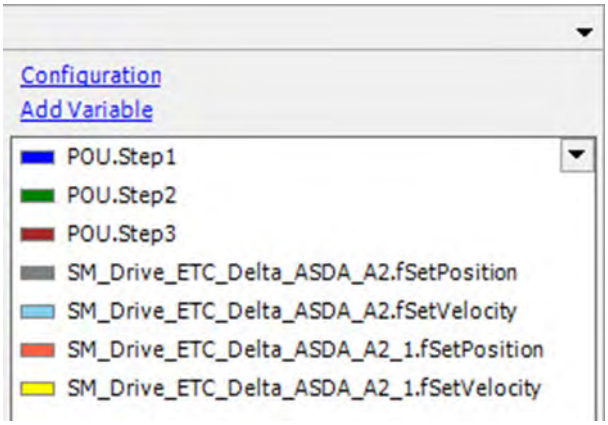
11. In case that there’re more than one variable or parameter need to be monitored, you can just right click on the scope and select “Convert to multi channel”.



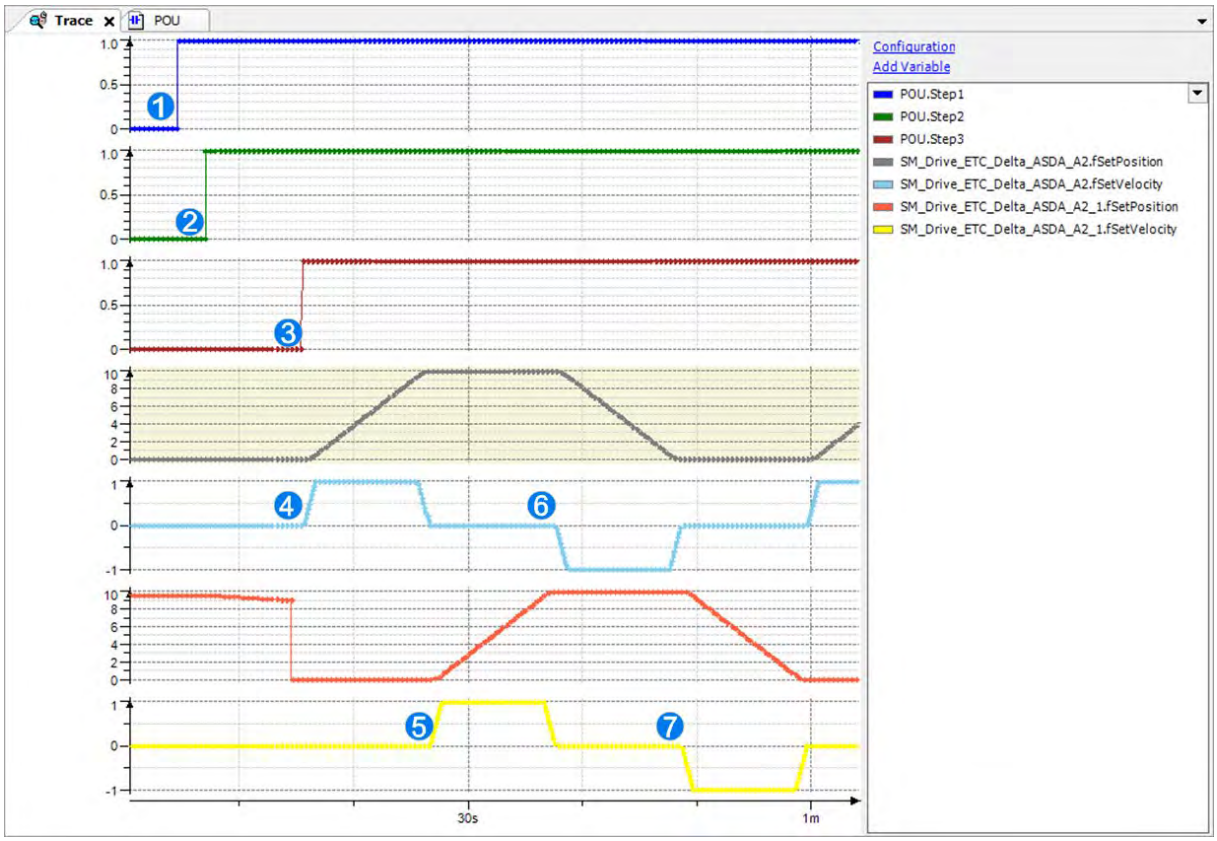
12. After choosing “Convert to multi channel”, two data-recording oscilloscope will be displayed on the page for the two chosen variables added on the right.



13. Repeat the above steps to add more required trace variables and traceable parameters.



14. While a program is running, you can observe trigger events and the current position of axis as well as the velocity via the data-recording oscilloscope.



Introduction of the process shown on the scope:

① Execute “Step1” to activate the servo axes via EtherCAT.

② Execute “Step2” to make axes perform homing.

③ Execute “Step3” to trigger “MC_MoveRelative” to activate the axis 1 & 2 for rotating in positive and negative directions.

④ Axis 1 starts moving in the positive direction.

⑤ Axis 2 is activated for rotating in the positive direction right after axis 1 stopped.

⑥ Axis 1 is activated for rotating in the negative direction right after axis 2 stopped.

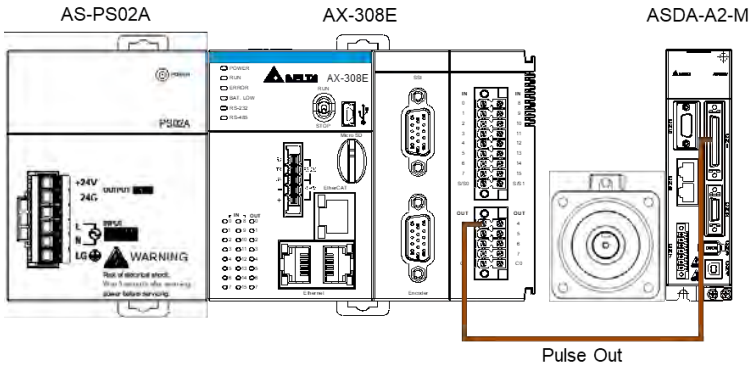
⑦ Axis 2 is activated for rotating in the negative direction right after axis 1 stopped.

3.3 Setup Projects with Pulse Output Features

The example here demonstrates using positioning function to abort the previous instruction with single-axis servo drive after the servo motor finishes rotating at constant velocity.

3.3.1 System Hardware Configuration

The servo motor (ASDA-A2-M) is controlled with a servo drive in between the PLC and the motor with the emphasis on the pulse train signals output from the PLC controller.

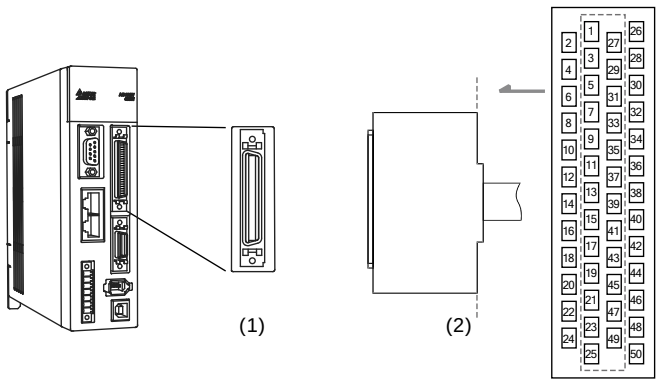


- Tools and materials

Numbering	Item	Model Number	Q'ty
1	AS Series Power Module	AS-PS02A	1
2	AX-3 Series CPU Module	AX-308EA0MA1T	1
3	Power supply	DVP-PS02	1
4	Pulse output type Servo drive	ASDA-A2-0421-M	1
5	Servo motor	ECMA-C10604ES	1
6	Servo power cable	ASD-ABPW0003	1
7	Servo encoder cable	ASD-ABEN0003	1
8	CN1 Terminal board t	ASD-BM-50A	1

- PLC Controller combining ASDA-A2-M hardware configuration

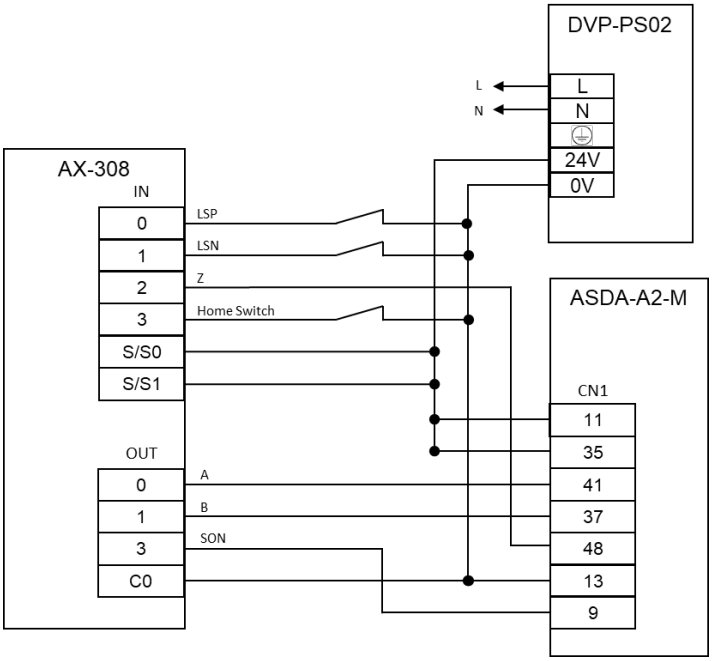
  The wiring and hardware limit need to be configured on the controller so as to feature pulse outputs.



(1) CN1 pin connector ; (2) CN1 connector pin diagram.

◆ Introduction to CN1 signals

Signals		Pin No	Features
Position pulse (Input)	PULSE /PULSE SIGN /SIGN PULL HI_P PULL HI_S	43 41 36 37 39 35	Position pulse can be sent by the open collector (single-phase max. frequency 200 kHz). Three command types, which are CW/CCW pulse, pulse and direction, and A/B pulse, can be selected with the parameter P1-00. If open collector type is used when sending position pulses, CN1 should be connected to an external power supply for pull high.
Position pulse (Output)	OCZ	48	Encoder Z output (Open collector).
Power	VDD	17	VDD is the +24 V power provided by the drive and is for Digital Input (DI) and Digital Output (DO) signal. The maximum current is 500 mA.
	COM+ COM-	11 45 47 49	COM+ is the common input of Digital Input (DI) and Digital Output (DO) voltage. When using VDD, VDD should be connected to COM+. If not using, it needs to apply the external power (+12 V ~ + 24 V). Its positive end should connect to COM+ and the negative end should connect to COM-.
	VCC	20	VCC is the +12V power provided by the drive. It is used for providing the simple analog command (speed or torque command). The maximum current is 100 mA.
	GND	12 ,13 ,19 , 44	VCC voltage is based on GND.
DI1	SON	9	When the DI.SON is ON and the motor servo circuit can operate smoothly, this DO is ON.



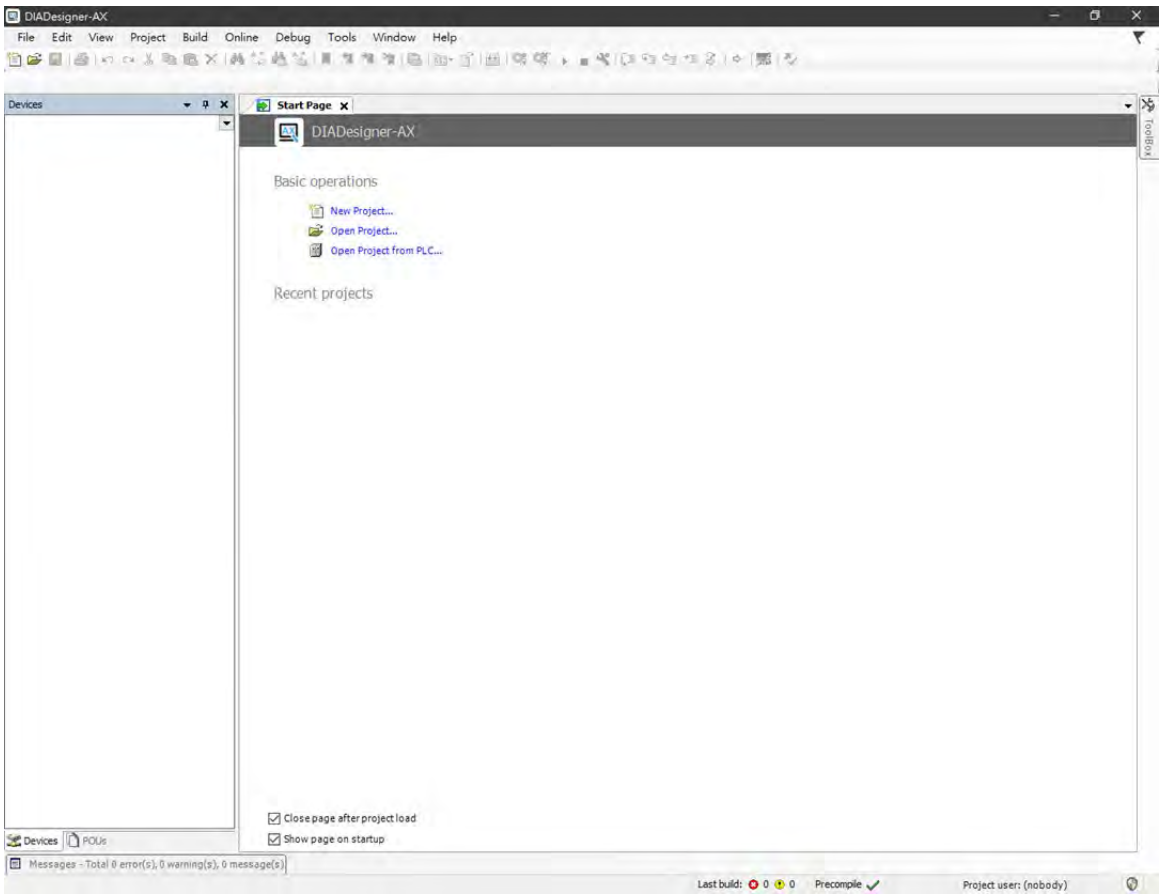
***Note:** Please find **ASDA-A2 Series User Manual** for more detailed information about the signal configuration of ASDA-A2-M**.**

3.3.2 Create New Projects and POUs

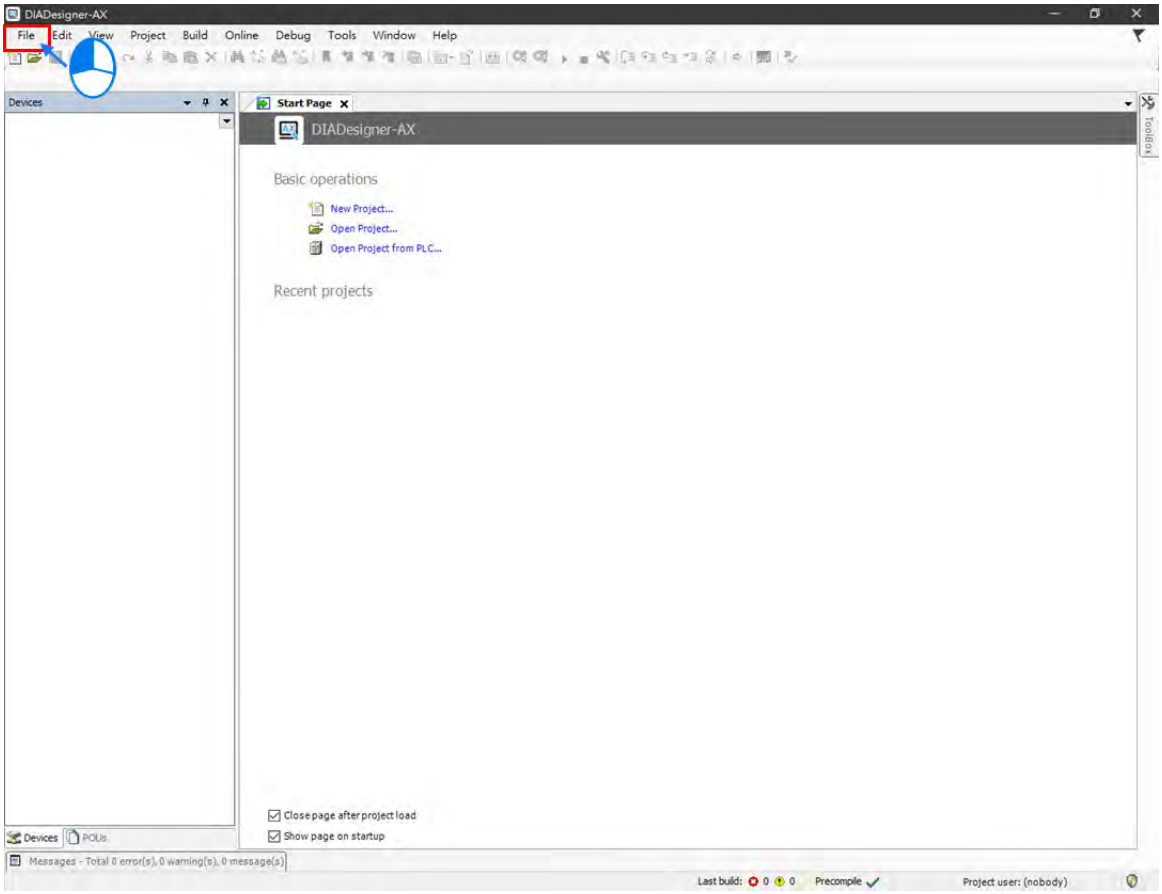
3.3.2.1 Create a New Project

1. First, open the programming software DIADesigner-AX.

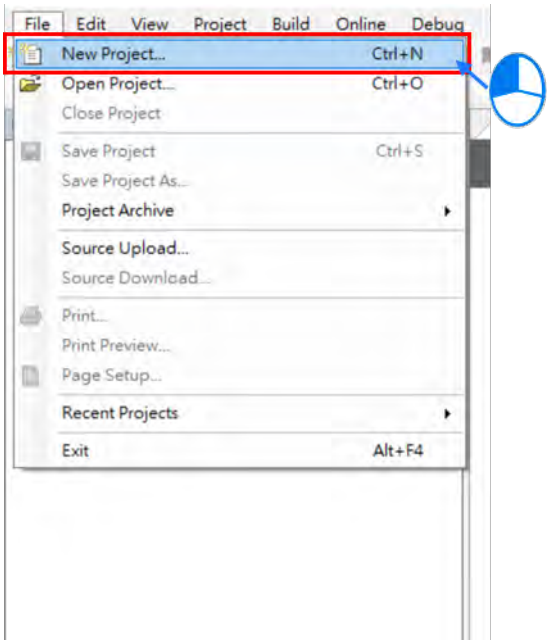
2. The display of DIADesigner-AX is shown as below.



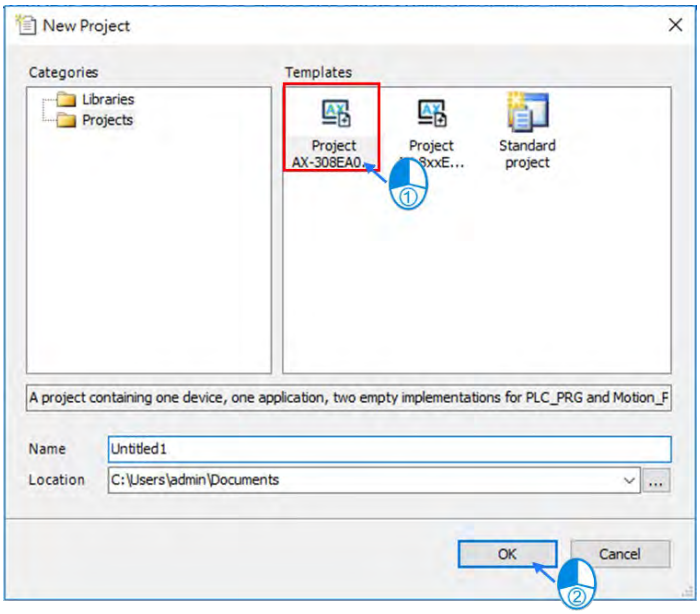
3. Click on “File”.



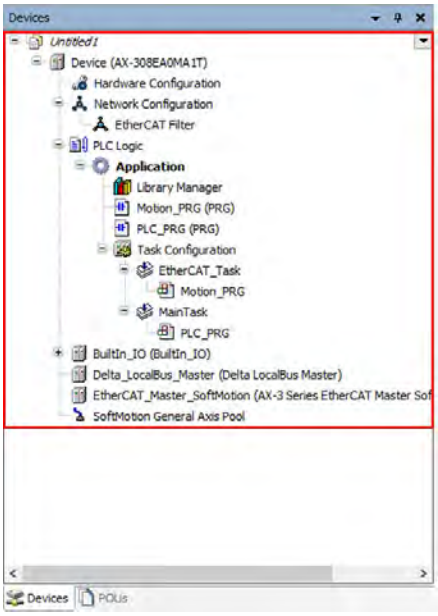
4. To create a new project, select "New Project" from the drop-down menu for DIADesigner-AX V1.0.0 or select “Standard Project” from the drop-down menu for DIADesigner-AX V1.1.0 or later.



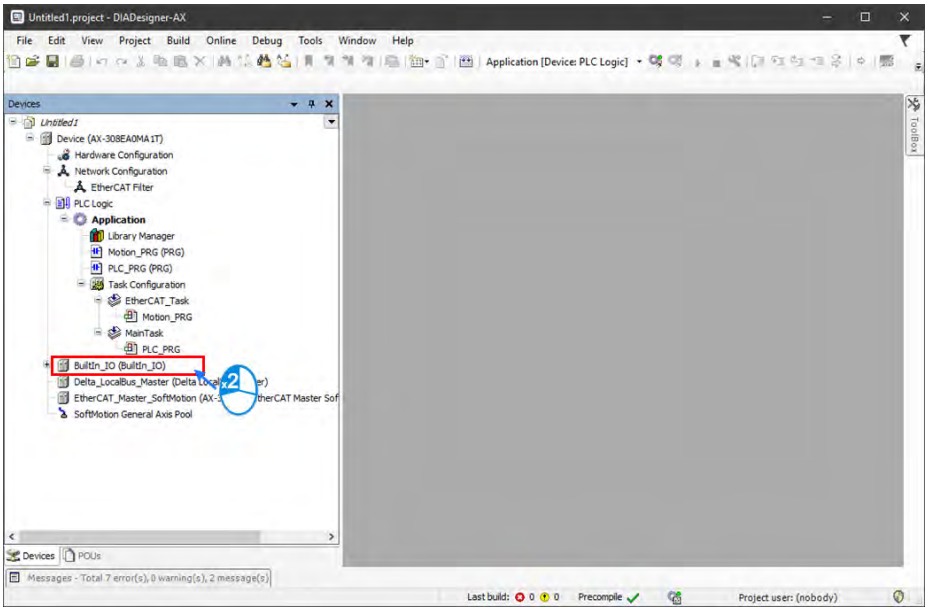
5. Select “Project AX-308EA0MA1T” after entering New Project page. Name the project in the field of Name and choose a path to the archive location, then click OK to move to the next page.



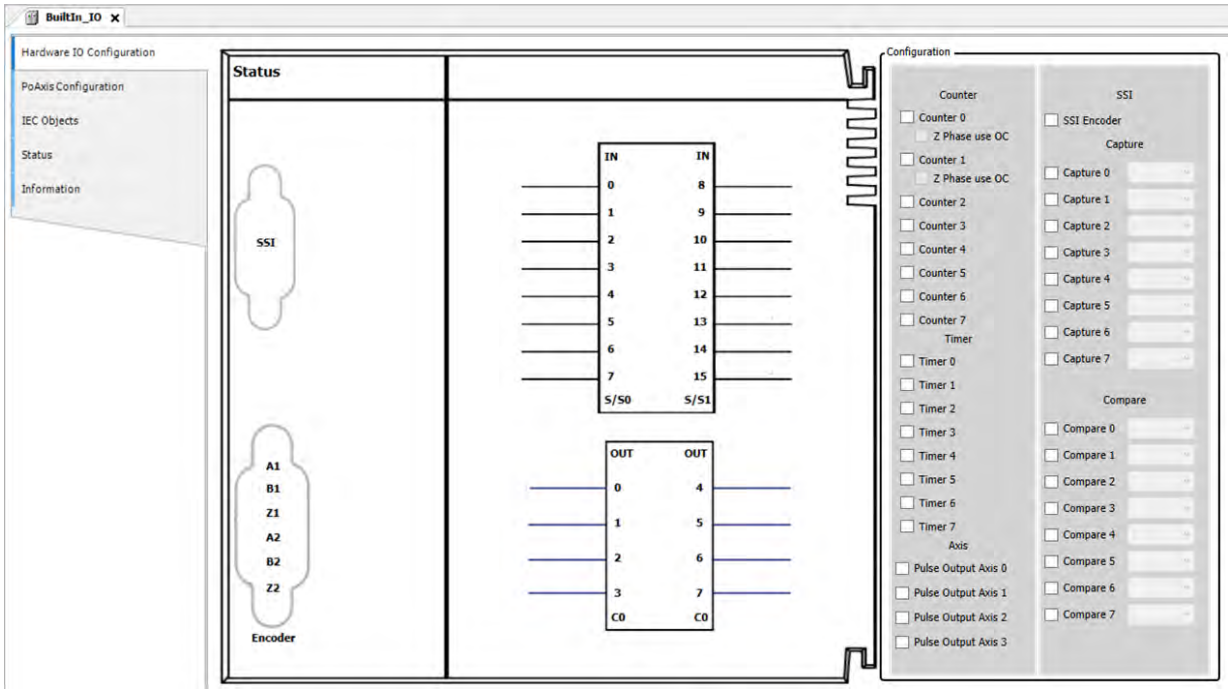
6. After entering the project page, double click the target object in the project tree on the left side of the page to open a certain program or configure settings of relating modules.



7. Double click on “BuiltIn_IO” to enter the I/O configuration page.

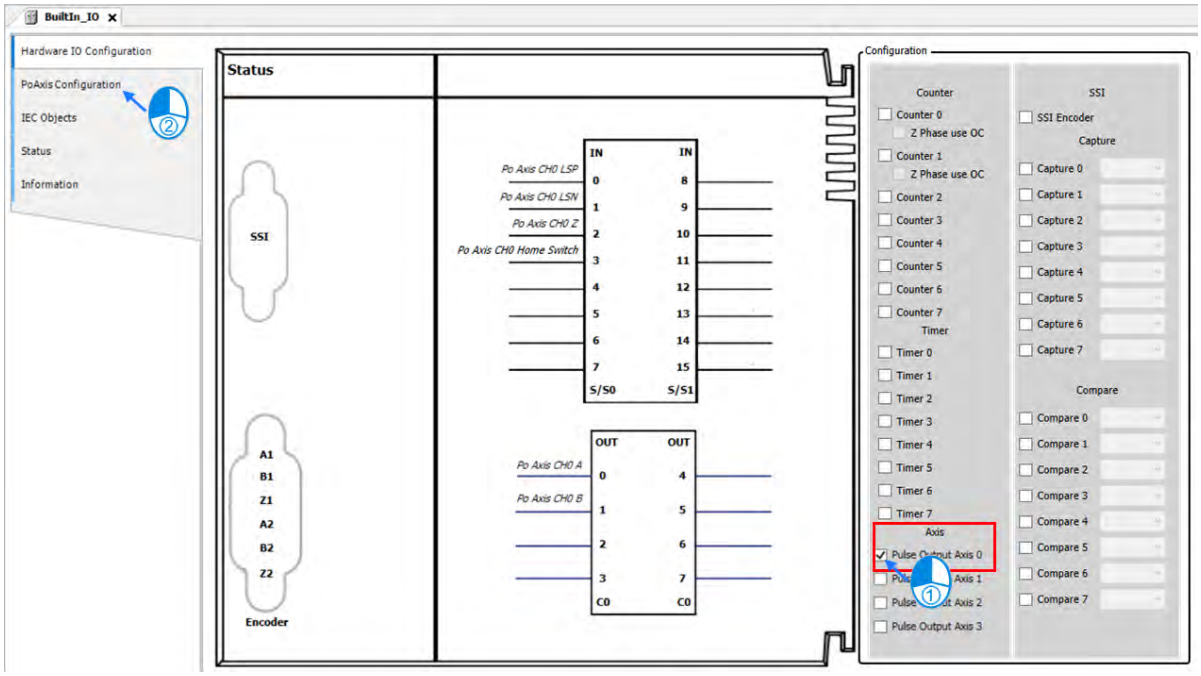


8. Start configuring the I/O devices settings in AX-308EA0MA1T on the BuiltIn_IO page.

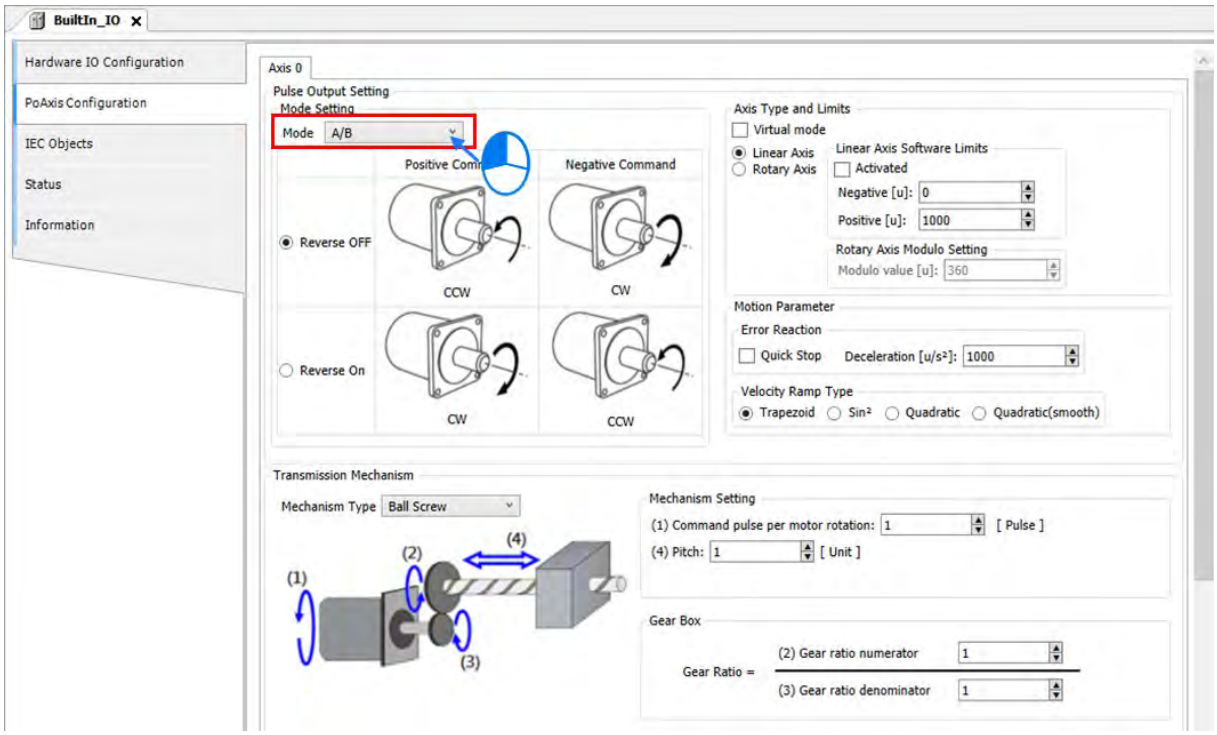


3.3.2.2 Axis Parameter Settings

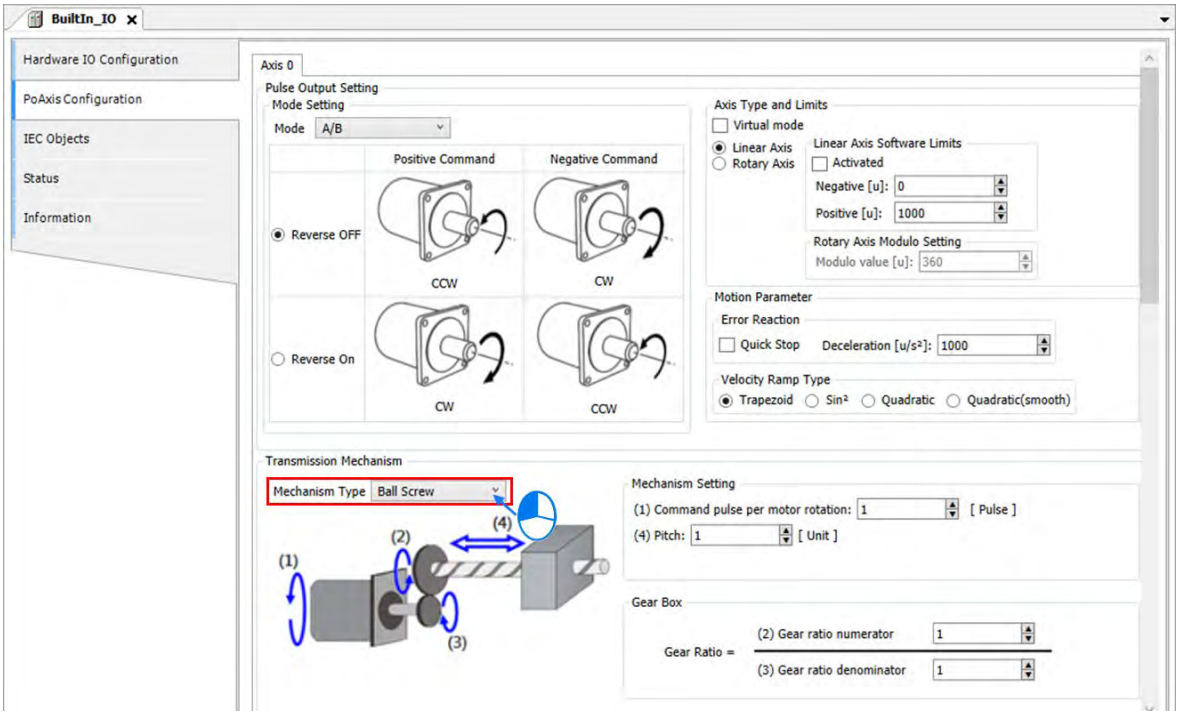
1. Select the checkbox of Pulse Output Axis 0 on the lower left side of the Configuration area. Then click “PoAxis Configuration” tab on the left to enter the page.



2. Select the pulse output mode from the drop down list of “Mode” on PoAxis Configuration page.

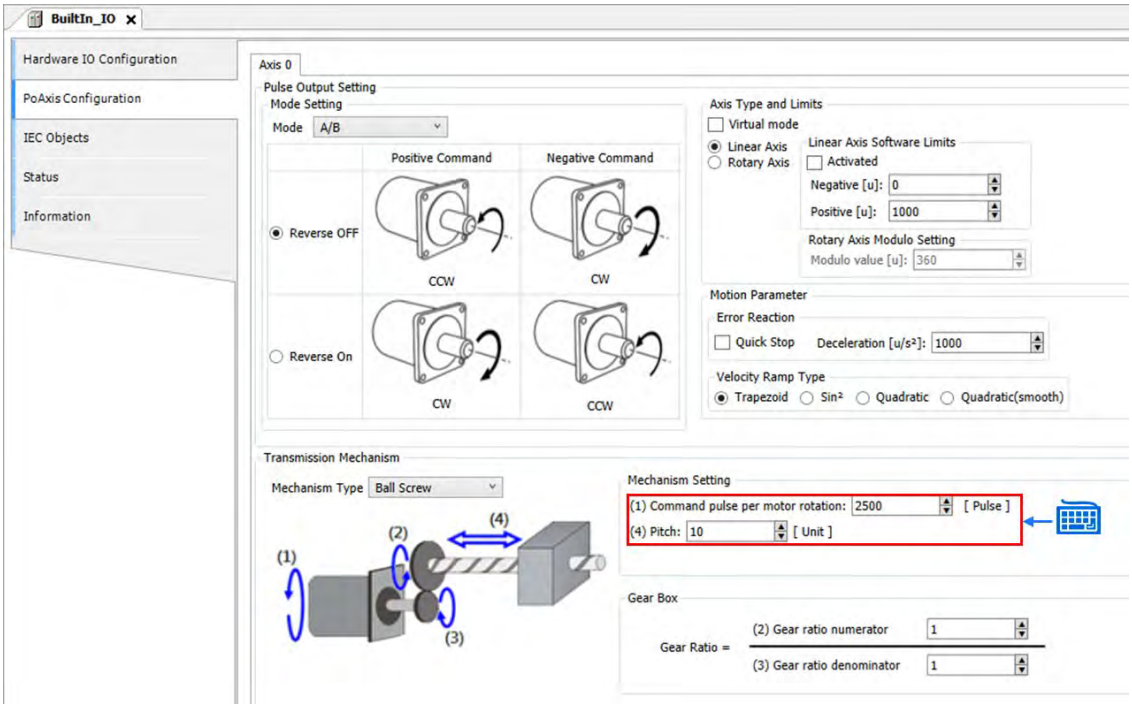


3. In this example, choose “Ball Screw” as the mechanism type.

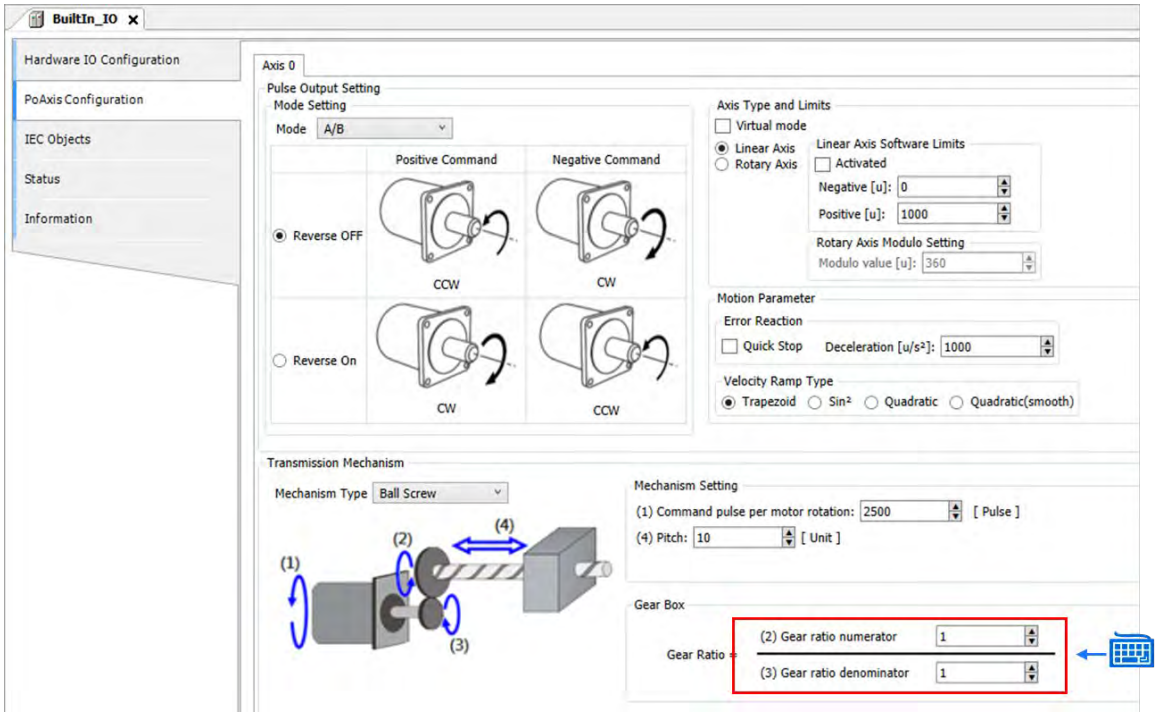


4. Continue with gear setting, set 2,500 for “Command pulse per motor rotation” (This parameter should be 10000, which is four times the setting value 2500, as a result of the phase AB on the servo drive.) Set 10 for the screw pitch (Gear ratio set on the servo motor with P1-44 = 128, P1-45 = 1) and the user unit in this example is “mm”.

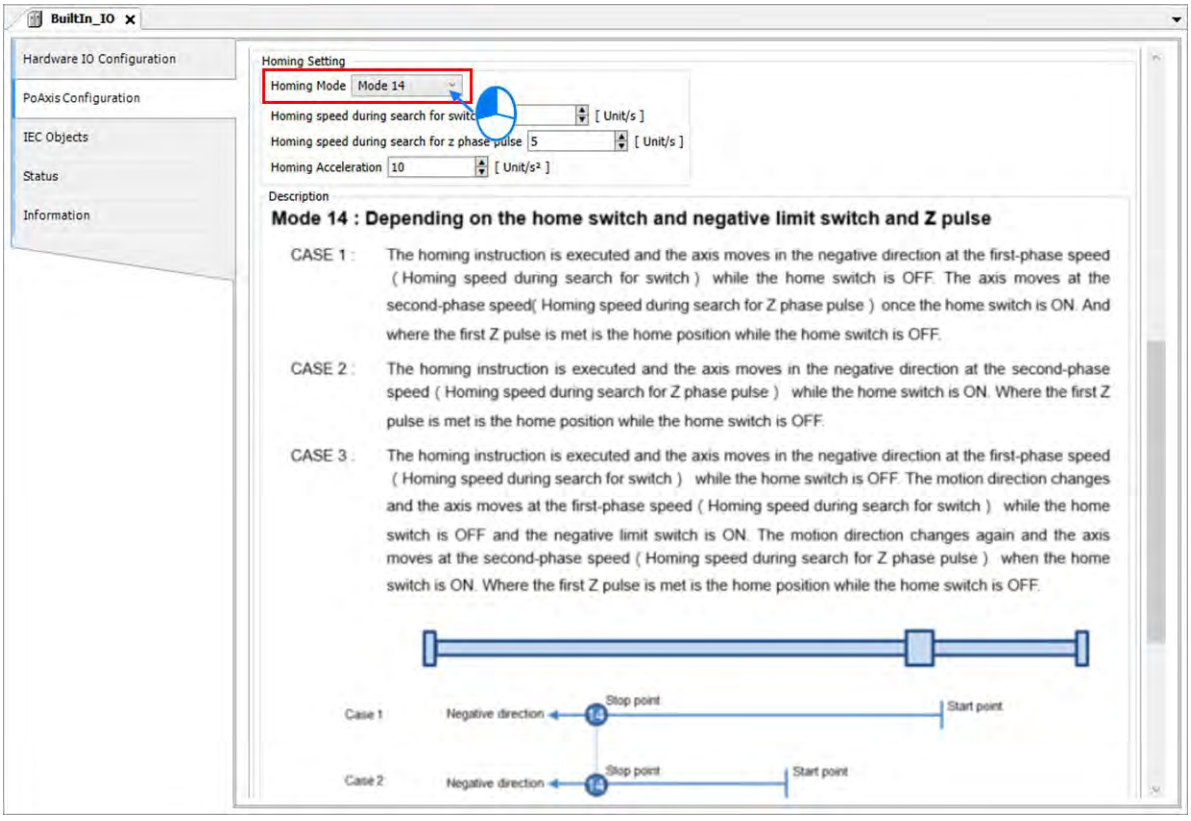
(Gear ratio $\frac{P1-44}{P1-45} = \frac{\text{Motor Pulse per rotation}}{\text{Command Pulse per rotation}}$)



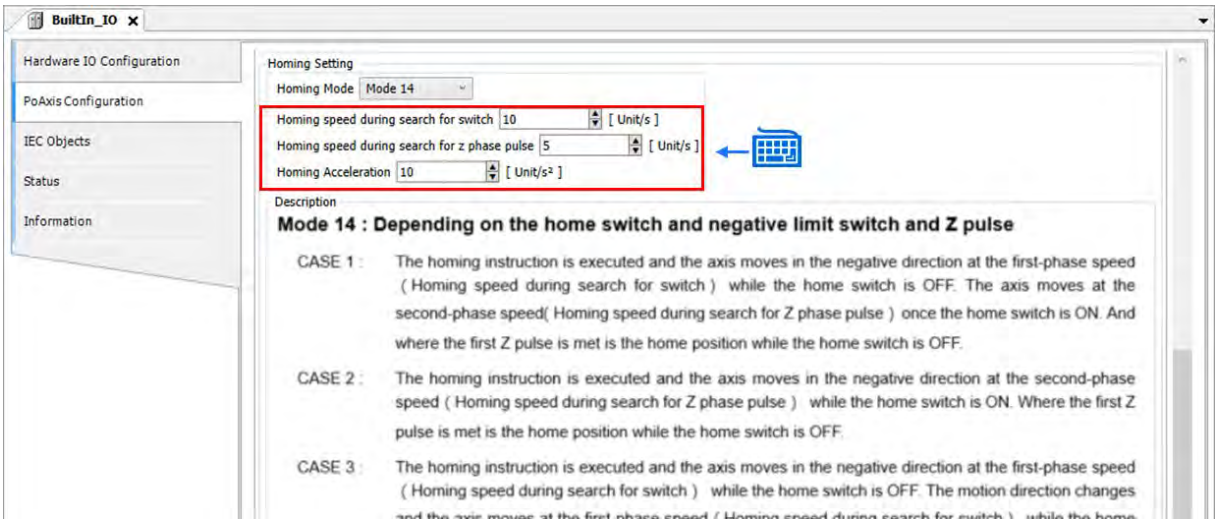
5. Since the mechanism is not equipped with a gearbox, the gear ratio for gearbox would be 1:1.



6. After finish configuring gear ratio parameters, set the homing mode in homing setting. The homing mode should match the hardware configuration, which the descriptions of each mode in different cases are listed below.



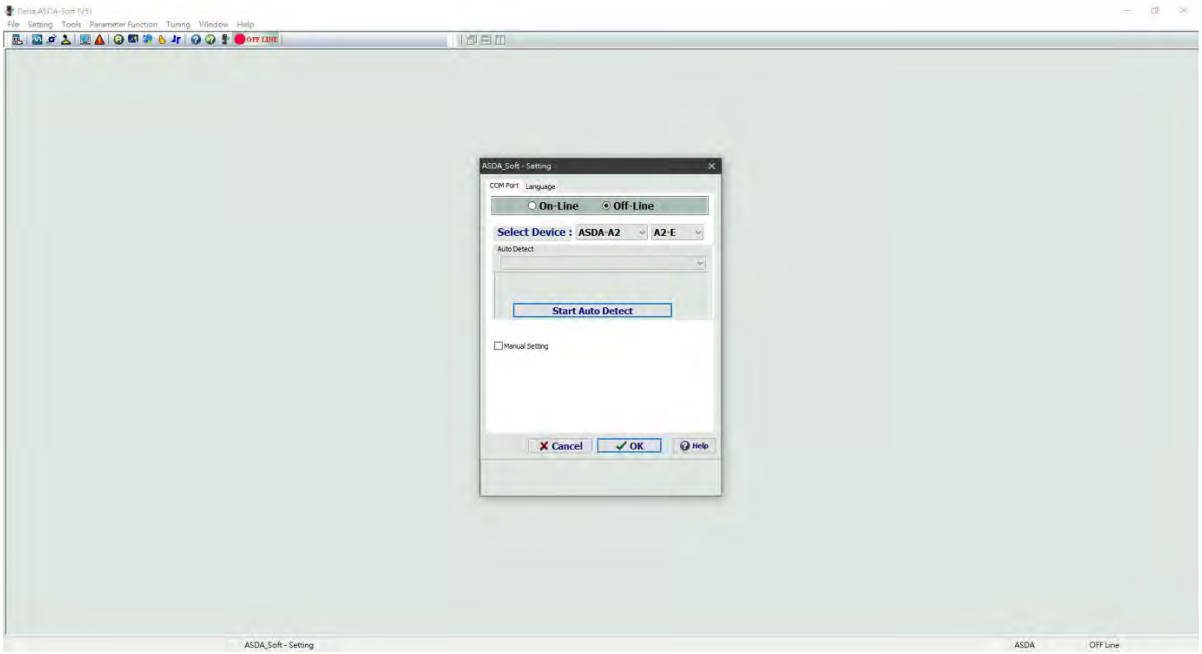
7. For homing speed, set 10 for the speed during search for switch and set the homing speed to 5 mm/s during search for z phase pulse, then adjust the homing acceleration to 10 mm/s. (User unit: mm)



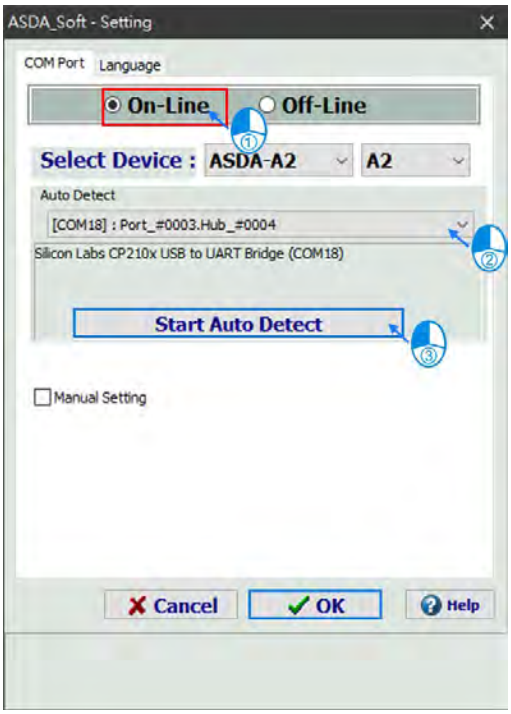
3.3.2.3 Configure Parameters for Servo Drive

There're two ways to set the servo parameters. First is via the control panel on the servo drive while the second is to use the software called "ASDA_SOFT".

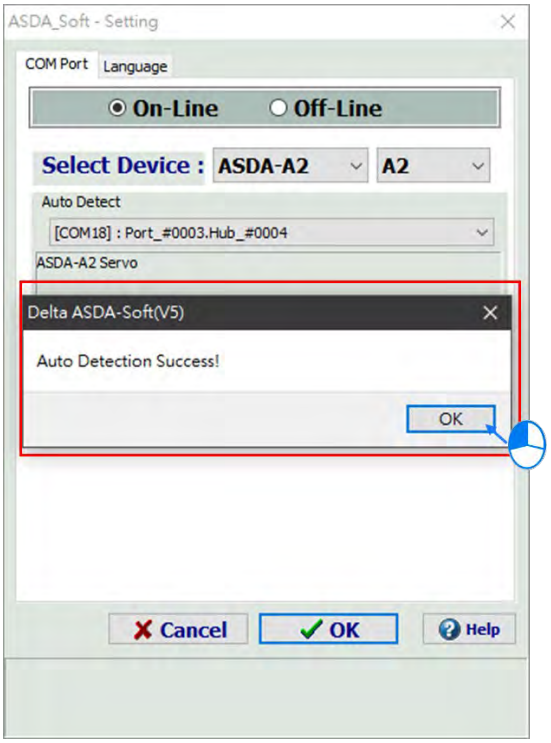
The display after activating the software ASDA_SOFT is shown as below.



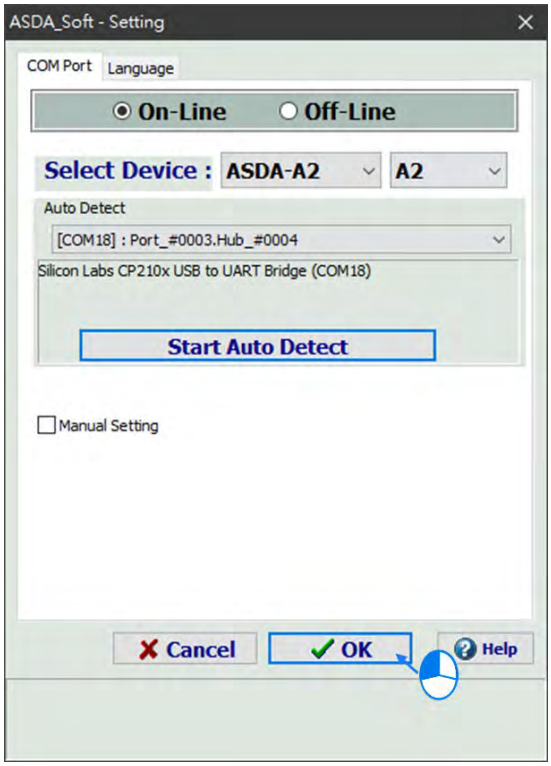
1. After choosing the target COM port used to connect the servo drive under “On-Line” mode, click on “Start Auto Detect”.



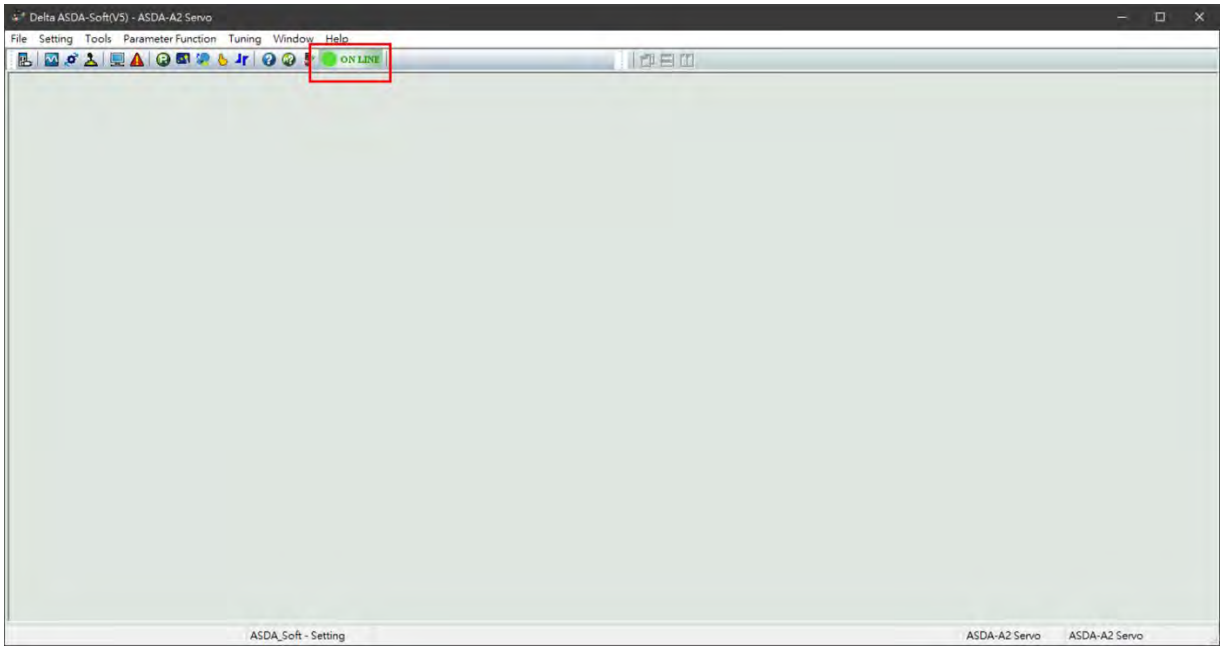
2. A message window will pop up if the detection is successful.



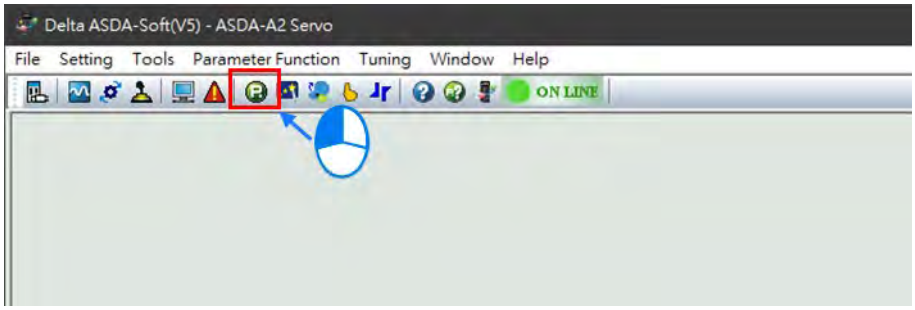
3. Click “OK” after confirming the communication is normal.



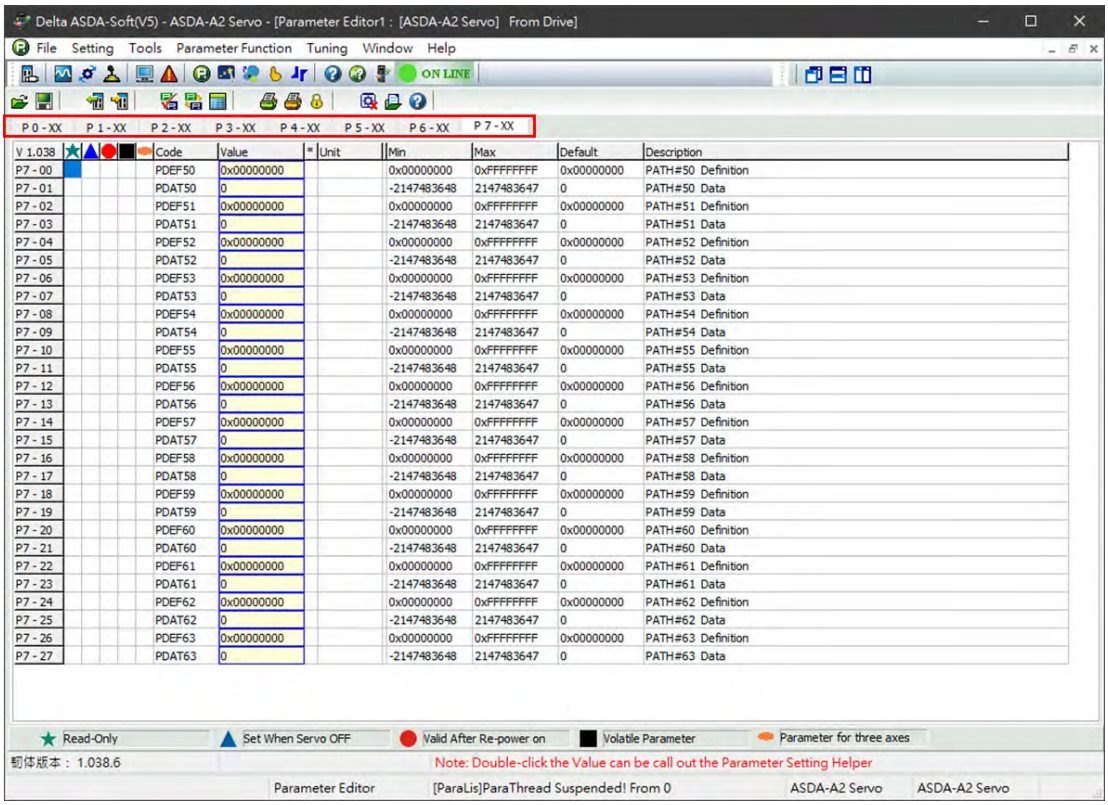
4. The status is shown as ON LINE which means the PC is connecting to the servo drive’s com port.



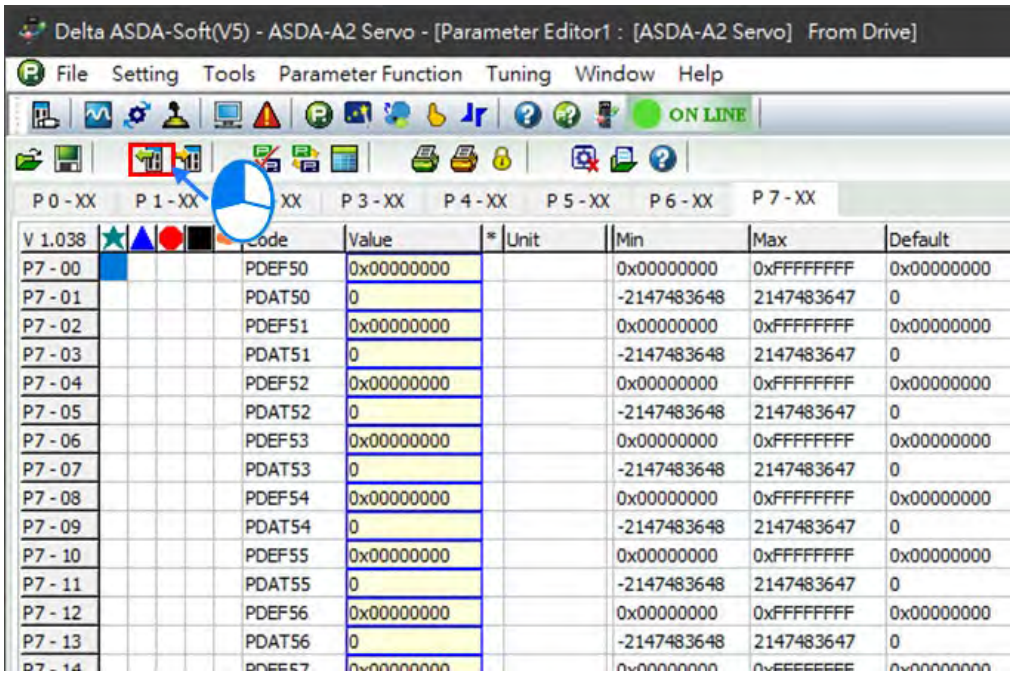
5. Click  in the tool bar to enter the servo parameter page.



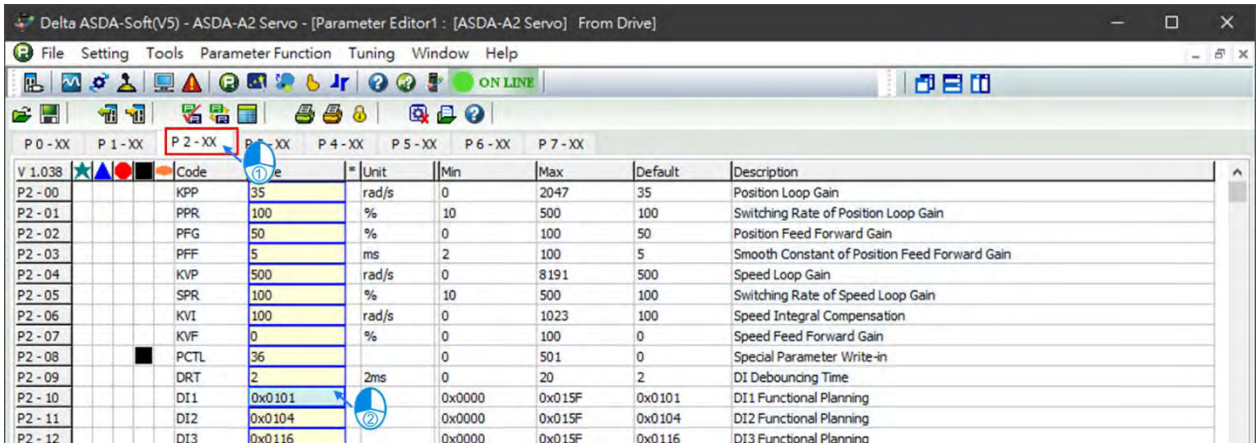
6. As shown in the following figure, the field being circled by the red box on the upper part of the page displays all the servo parameters in groups.



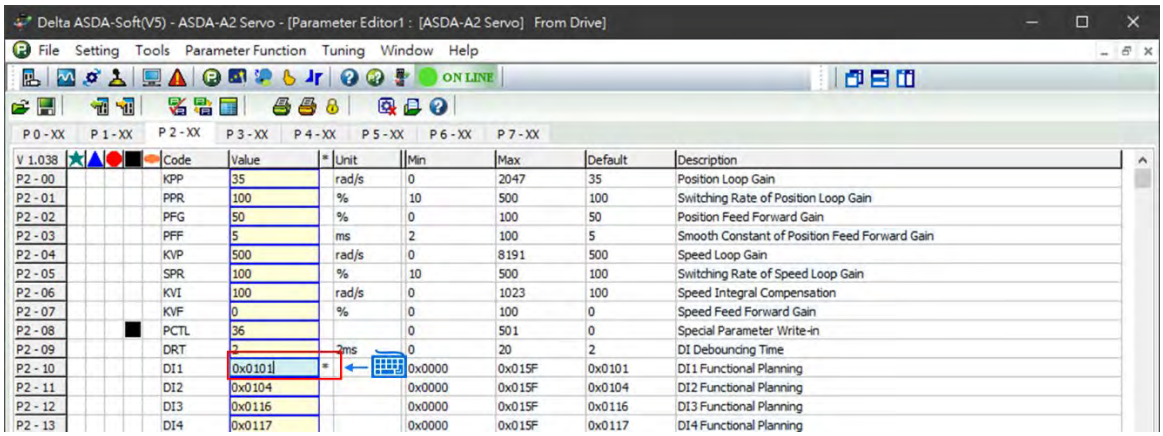
7. Click  to upload all the servo drive's parameters.



8. Click "P2-XX" after the upload is completed, then click on the value of P2-XX parameter.

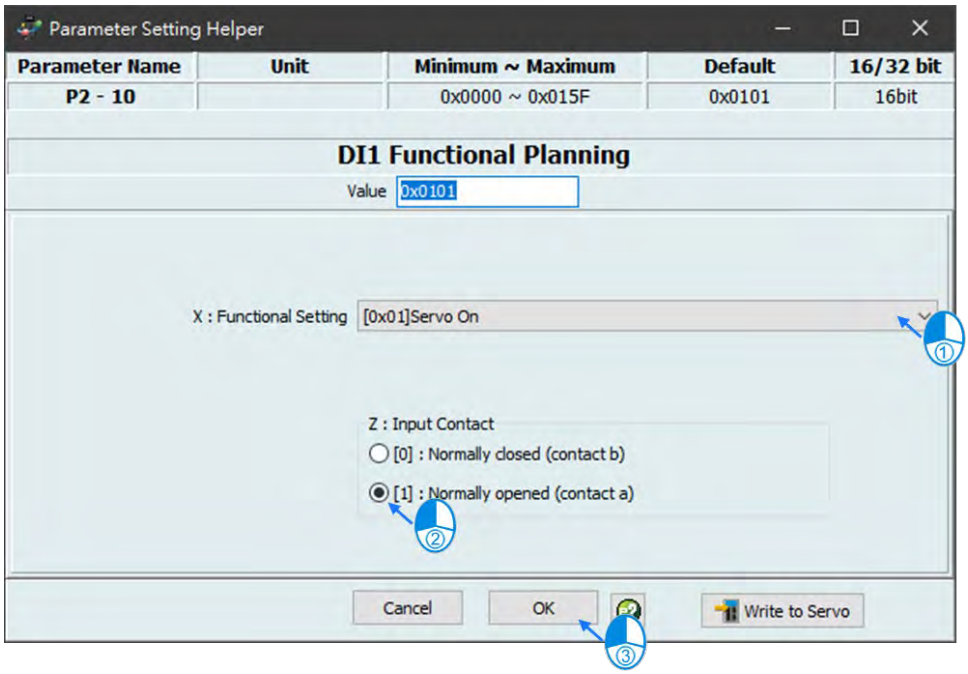


9. The setting value here can be modified by inputting the value directly or double-clicking on it to open the "Parameter Setting Helper" window.

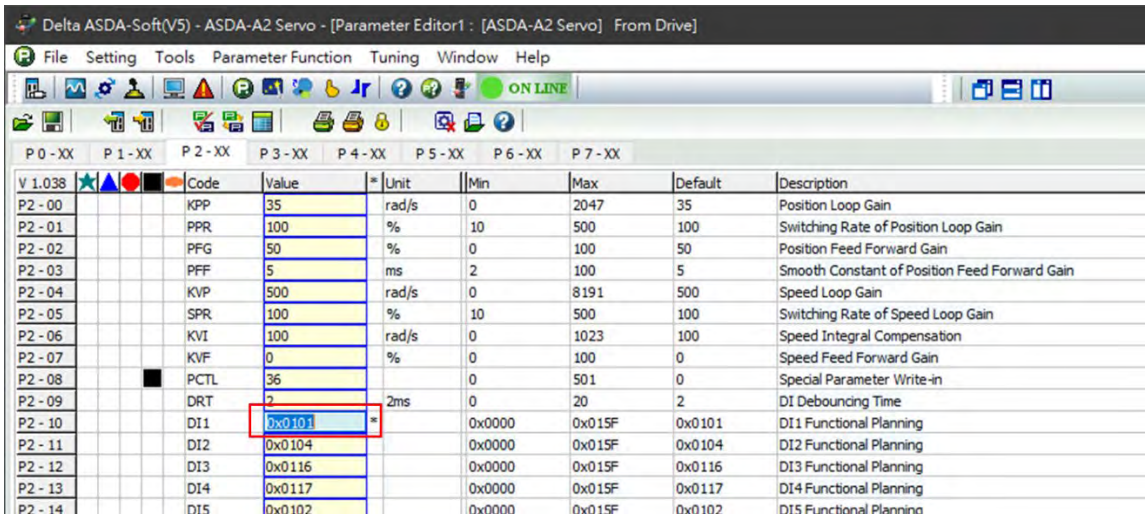


V 1.038	Code	Value	*	Unit	Min	Max	Default	Description
P2-00	KPP	35		rad/s	0	2047	35	Position Loop Gain
P2-01	PPR	100		%	10	500	100	Switching Rate of Position Loop Gain
P2-02	PFG	50		%	0	100	50	Position Feed Forward Gain
P2-03	PFF	5		ms	2	100	5	Smooth Constant of Position Feed Forward Gain
P2-04	KVP	500		rad/s	0	8191	500	Speed Loop Gain
P2-05	SPR	100		%	10	500	100	Switching Rate of Speed Loop Gain
P2-06	KVI	100		rad/s	0	1023	100	Speed Integral Compensation
P2-07	KVF	0		%	0	100	0	Speed Feed Forward Gain
P2-08	PCTL	36			0	501	0	Special Parameter Write-in
P2-09	DRT	2		2ms	0	20	2	DI Debouncing Time
P2-10	DI1	0x0101	*		0x0000	0x015F	0x0101	DI1 Functional Planning
P2-11	DI2	0x0104			0x0000	0x015F	0x0104	DI2 Functional Planning
P2-12	DI3	0x0116			0x0000	0x015F	0x0116	DI3 Functional Planning
P2-13	DI4	0x0117			0x0000	0x015F	0x0117	DI4 Functional Planning

10. Choose the mode for "Functional Setting" and select for the "Input Contact" on the Parameter Setting Helper window, then click "OK" to finish.

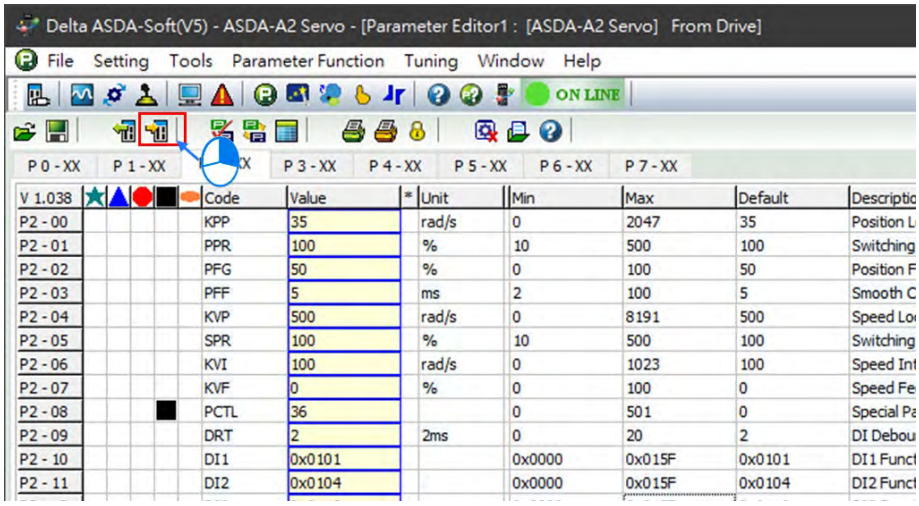


11. A mark of * would appear next to the setting value when the value is changed. To import the changed value to the servo drive, click on it and press Enter key.



V 1.038		Code	Value	*	Unit	Min	Max	Default	Description
P2 - 00		KPP	35		rad/s	0	2047	35	Position Loop Gain
P2 - 01		PPR	100		%	10	500	100	Switching Rate of Position Loop Gain
P2 - 02		PFG	50		%	0	100	50	Position Feed Forward Gain
P2 - 03		PFF	5		ms	2	100	5	Smooth Constant of Position Feed Forward Gain
P2 - 04		KVP	500		rad/s	0	8191	500	Speed Loop Gain
P2 - 05		SPR	100		%	10	500	100	Switching Rate of Speed Loop Gain
P2 - 06		KVI	100		rad/s	0	1023	100	Speed Integral Compensation
P2 - 07		KVF	0		%	0	100	0	Speed Feed Forward Gain
P2 - 08		PCTL	36			0	501	0	Special Parameter Write-in
P2 - 09		DRT	2		2ms	0	20	2	DI Debouncing Time
P2 - 10		DI1	0x0101	*		0x0000	0x015F	0x0101	DI1 Functional Planning
P2 - 11		DI2	0x0104			0x0000	0x015F	0x0104	DI2 Functional Planning
P2 - 12		DI3	0x0116			0x0000	0x015F	0x0116	DI3 Functional Planning
P2 - 13		DI4	0x0117			0x0000	0x015F	0x0117	DI4 Functional Planning
P2 - 14		DI5	0x0102			0x0000	0x015F	0x0102	DI5 Functional Planning

12. After the value being imported, the mark * would disappear. In case that a large number of parameter values have been modified, you are allowed to download all the changed parameters to the servo drive in one time by clicking on  .



V 1.038		Code	Value	*	Unit	Min	Max	Default	Descriptio
P2 - 00		KPP	35		rad/s	0	2047	35	Position L
P2 - 01		PPR	100		%	10	500	100	Switching
P2 - 02		PFG	50		%	0	100	50	Position F
P2 - 03		PFF	5		ms	2	100	5	Smooth C
P2 - 04		KVP	500		rad/s	0	8191	500	Speed Lo
P2 - 05		SPR	100		%	10	500	100	Switching
P2 - 06		KVI	100		rad/s	0	1023	100	Speed Int
P2 - 07		KVF	0		%	0	100	0	Speed Fe
P2 - 08		PCTL	36			0	501	0	Special Pa
P2 - 09		DRT	2		2ms	0	20	2	DI Debou
P2 - 10		DI1	0x0101			0x0000	0x015F	0x0101	DI1 Funct
P2 - 11		DI2	0x0104			0x0000	0x015F	0x0104	DI2 Funct

■ Introduction of servo parameters

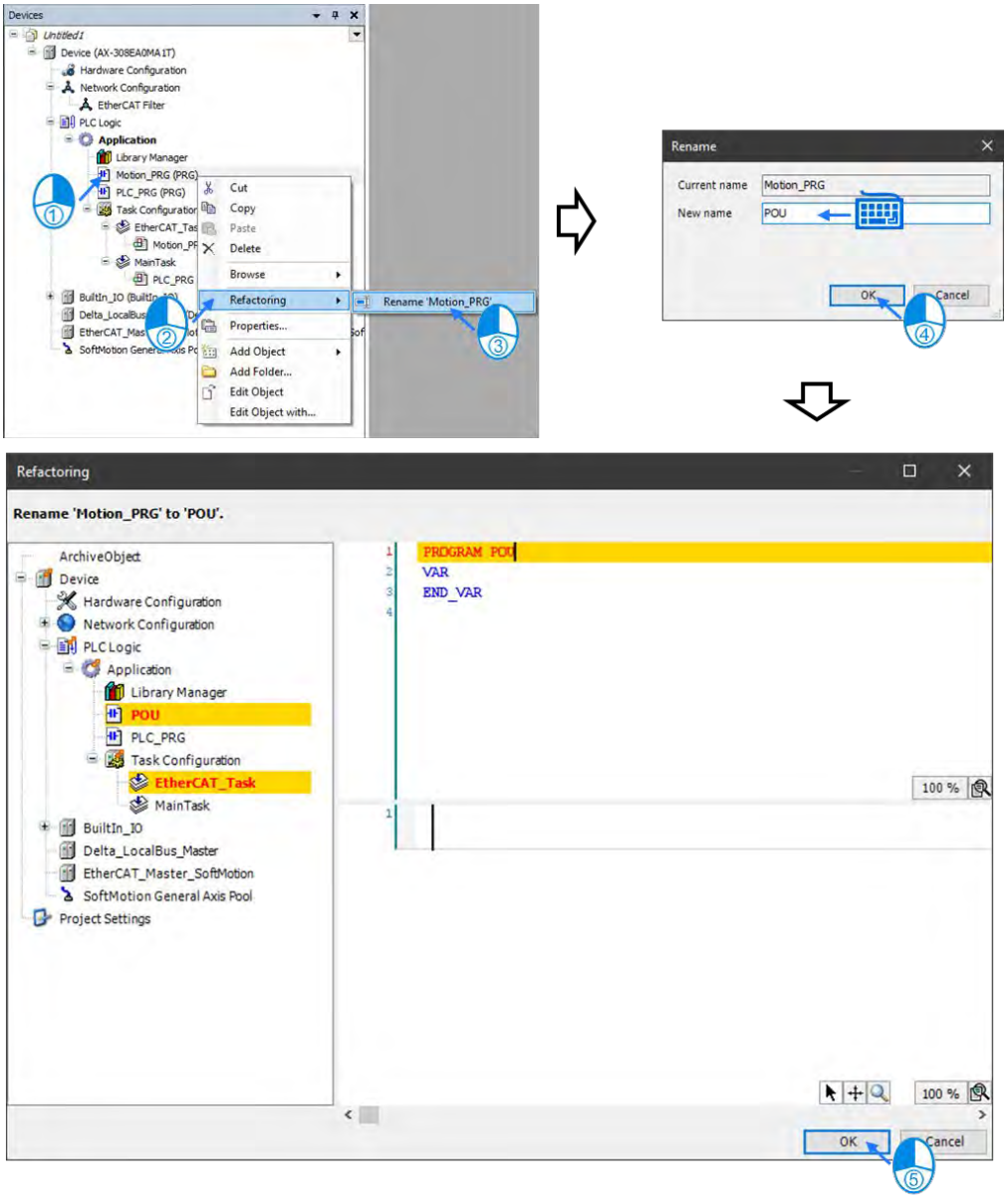
Parameter	Function	Setting value
P1-00	External pulse train input type	0x0000
P1-44	Gear Ratio (Numerator)	128
P1-45	Gear Ratio (Denominator)	1
P2-10	DI1 Functional Planning	0x0101
P2-15	DI6 Functional Planning	0x0100
P2-16	DI7 Functional Planning	0x0100
P2-17	DI8 Functional Planning	0x0100

***Note:** Please find **ASDA-A2 Series User Manual** for more detailed information about the parameters.

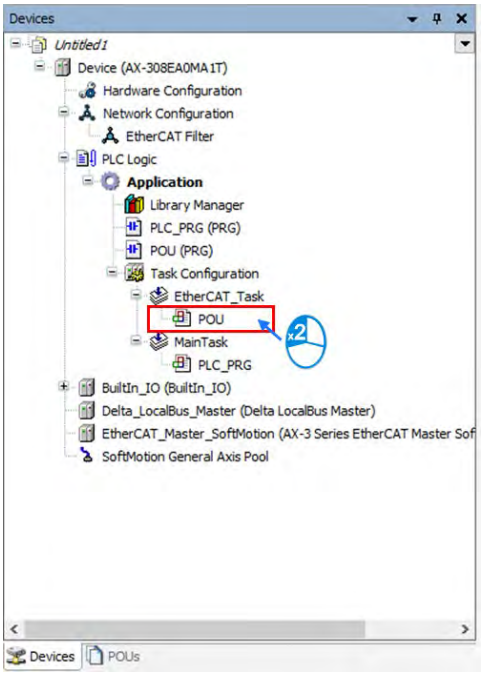
3.3.2.4 Programming for New Project

- Operating process in the example
  - "MC_Power" will be triggered and enable the servo axis (Servo ON) after executing "Step 1".
  - "MC_Home" will be triggered after executing "Step 2" to command the axis back to the home position 0.
  - Execute "Step 3" to trigger "MC_MoveVelocity". Then "MC_MoveAbsolute" will be triggered once the position of 300 is reached. At the same time, the axis is commanded to move back to position 0. The cycle will be repeatly executed until "Step 3" being turned off.
  - The value of Counter_1 will be increased by 1 for each time the cycle is completed.
- Programming with newly-created POUs

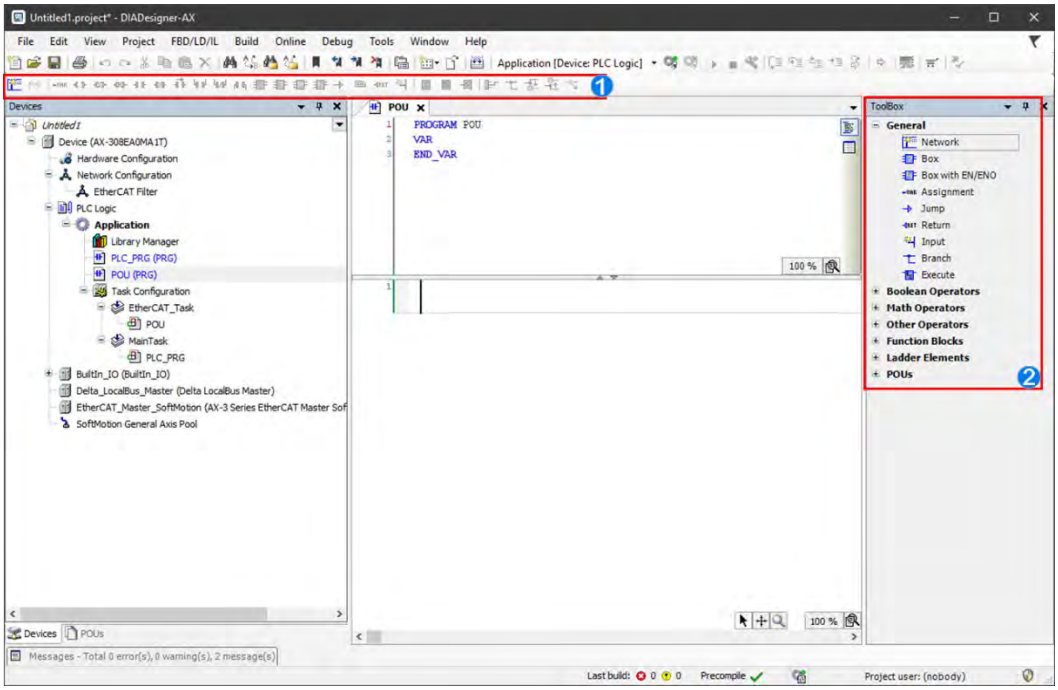
1. Change the name of Motion_PRG. Choose "Motion_PRG" and then "Refactoring", to change the name of Motion_PRG to POU, then click on "OK". Meanwhile, a confirmation window will pop up to confirm if you want to automaticallly adapt all references within the project. Click on "Yes" to complete.



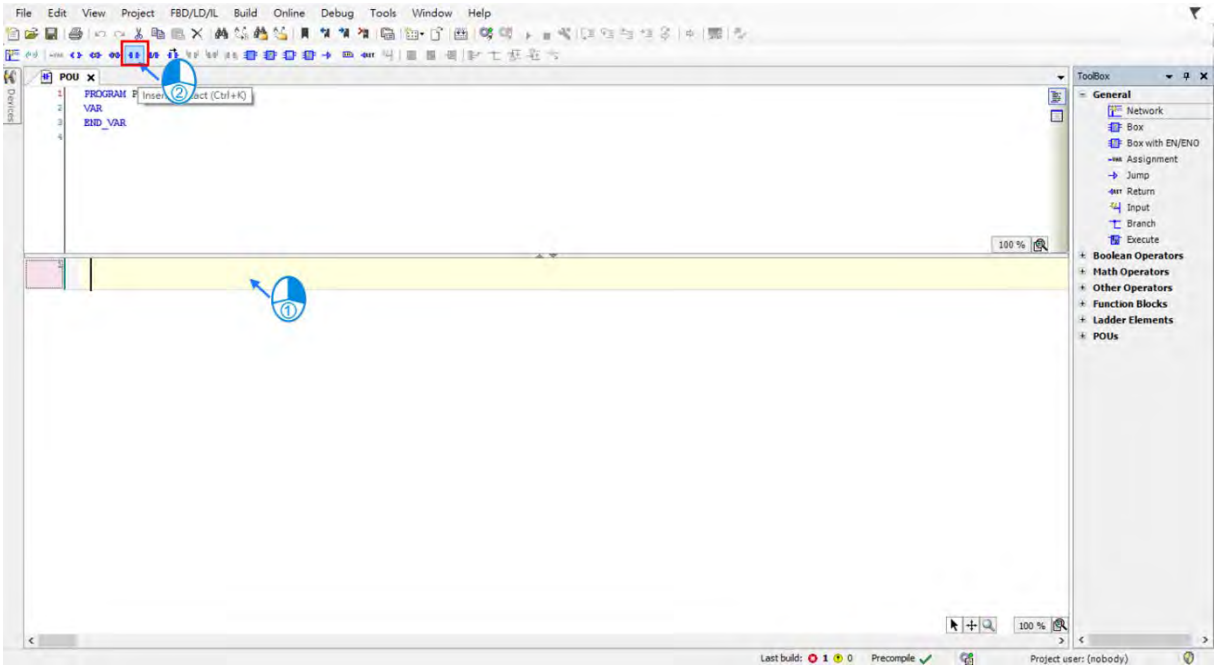
2. Double click on “POU” after name changed.



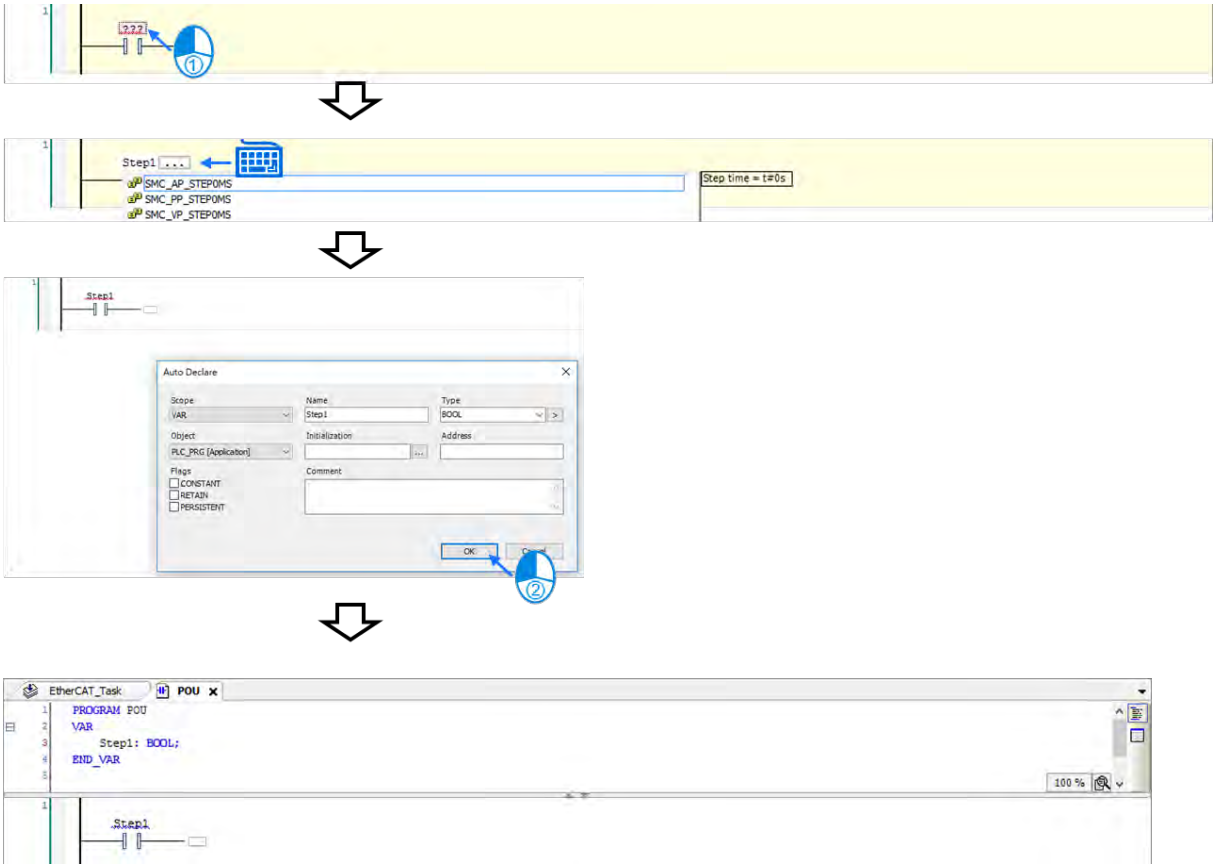
3. With Ladder Diagram programming, add new commands by using the red-circled field marked with 1 , while field 2 is for adding required Function or Function Block.



4. Click on the desired place to add new commands on POU page, then select  .

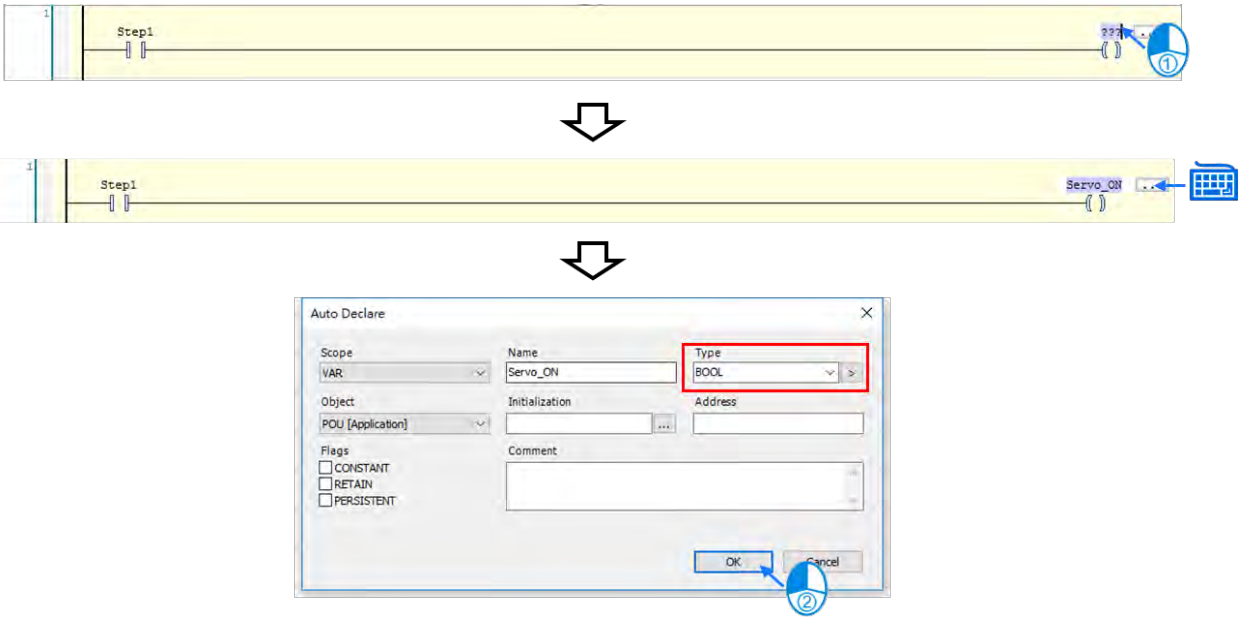


5. After the selected command being shown in the POU, click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.



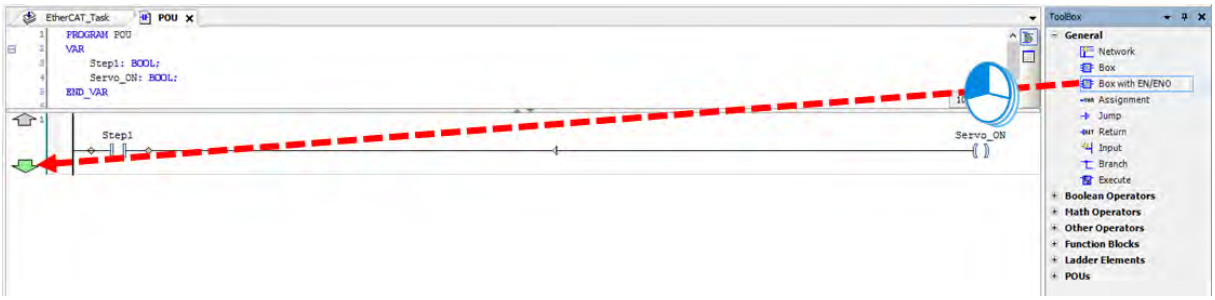
6. Then add the ouput instruction by clicking on the desired place and choosing  .

7. Click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.

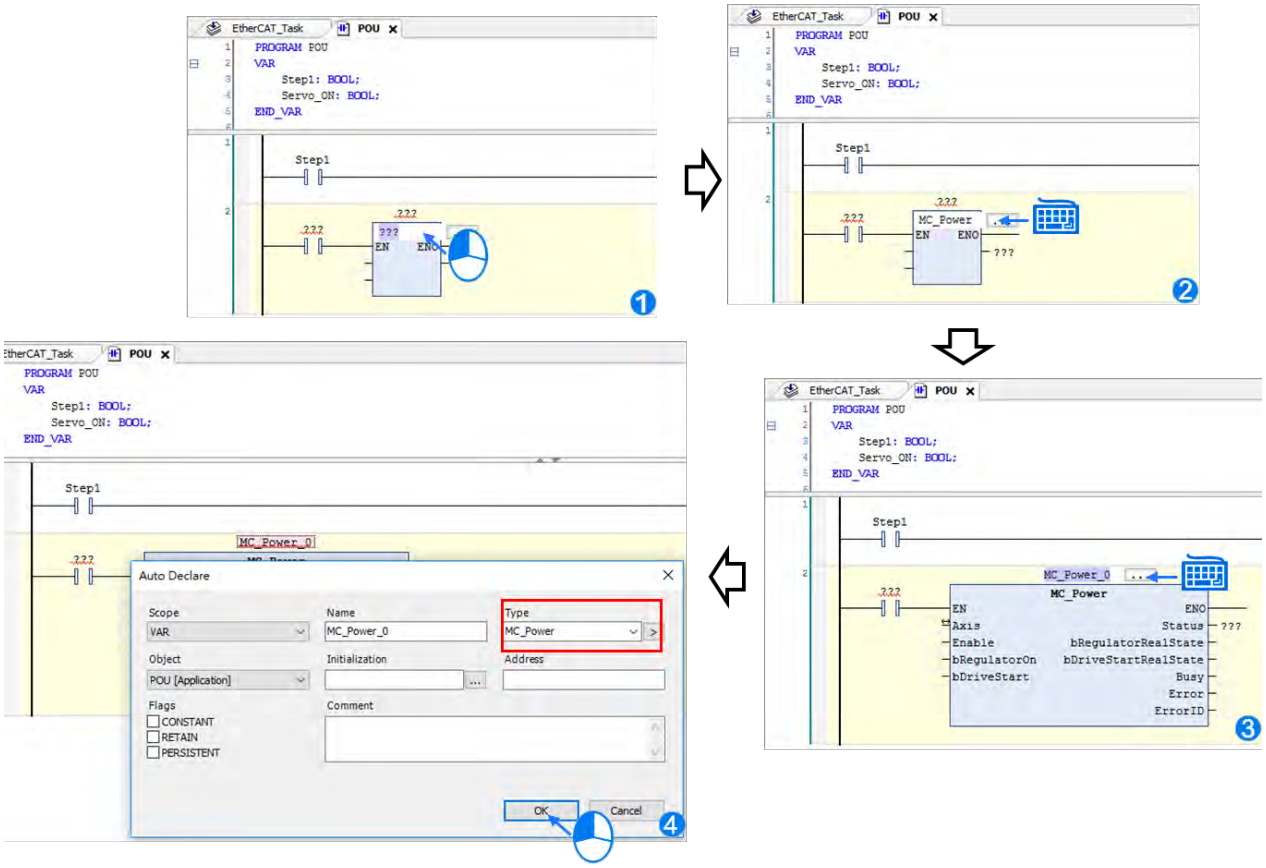


8. The instruction names you have added previously will be displayed as well as the instruction types, where the name and type of the instruction can also be added directly. (Same operation for the examples in later discussion.)

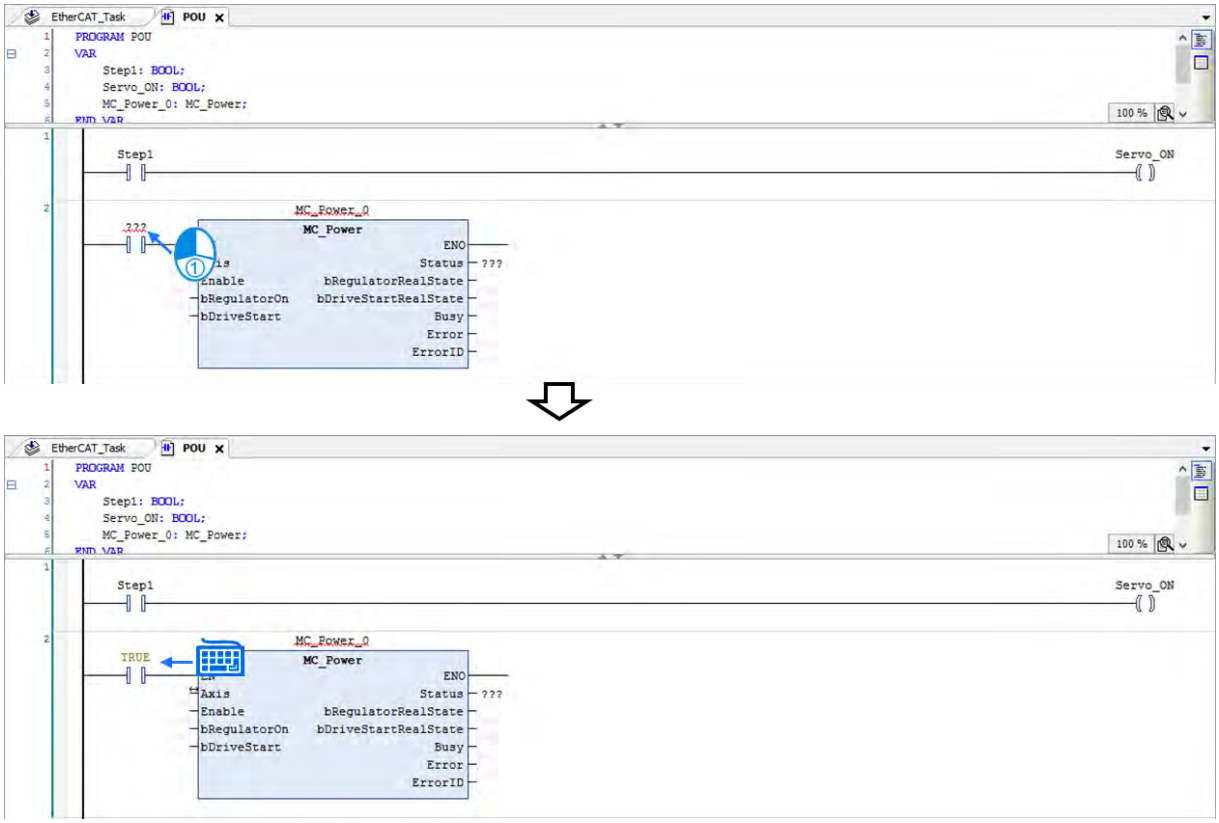
9. To add new function blocks, click and hold “Box With EN / ENO” then drag to .



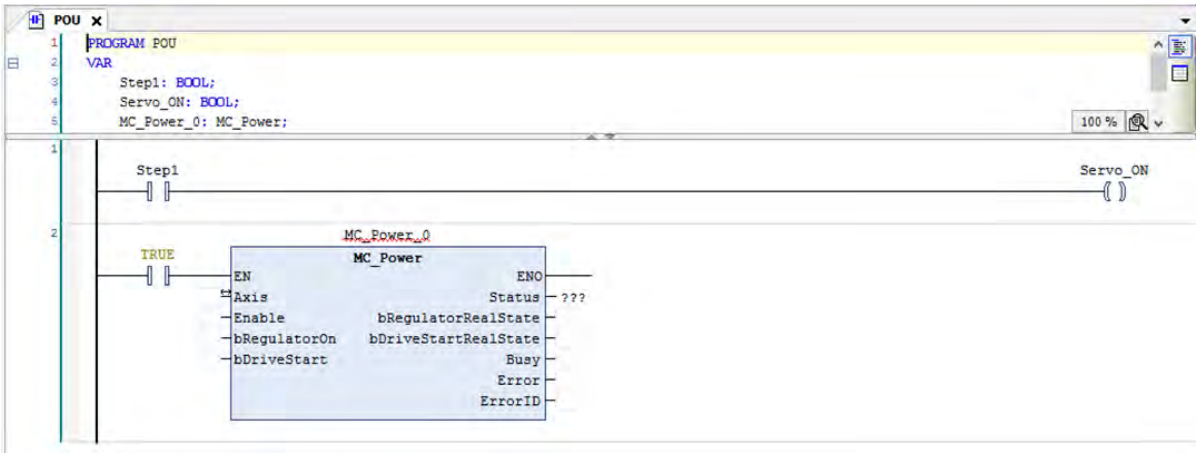
10. Click “???” and insert the name of function block then press ENTER to display the function block. Insert the name and press ENTER key again, then “Auto Declare” window will pop up. Click “OK” after confirming the “Type” of function block is correct.



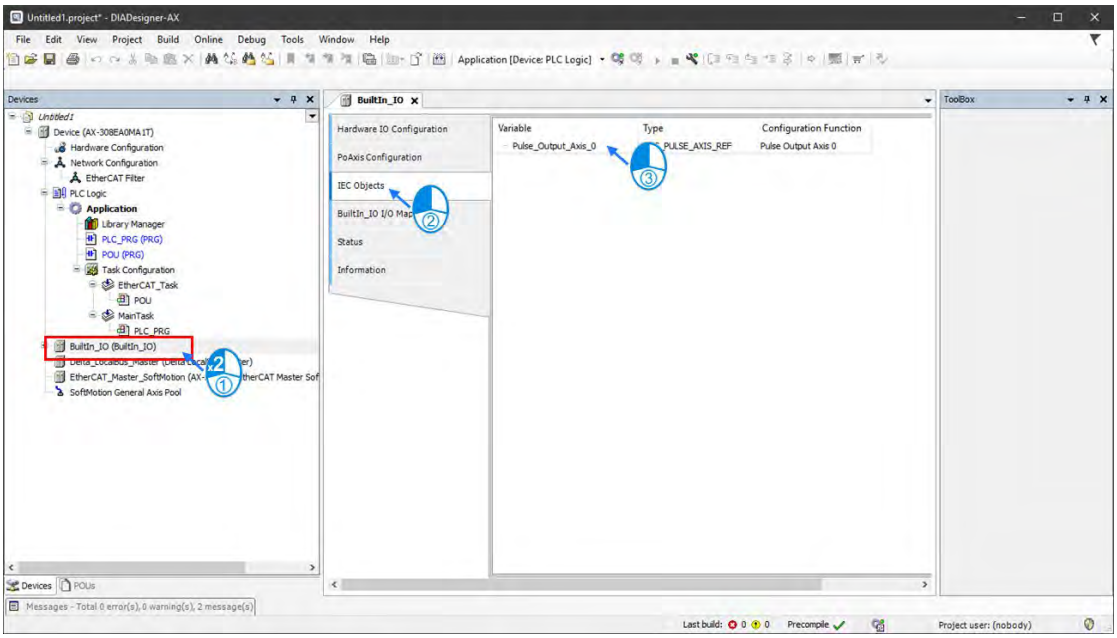
11. Continue to click “???” and insert “True”, then press Enter key with the newly-added function block.



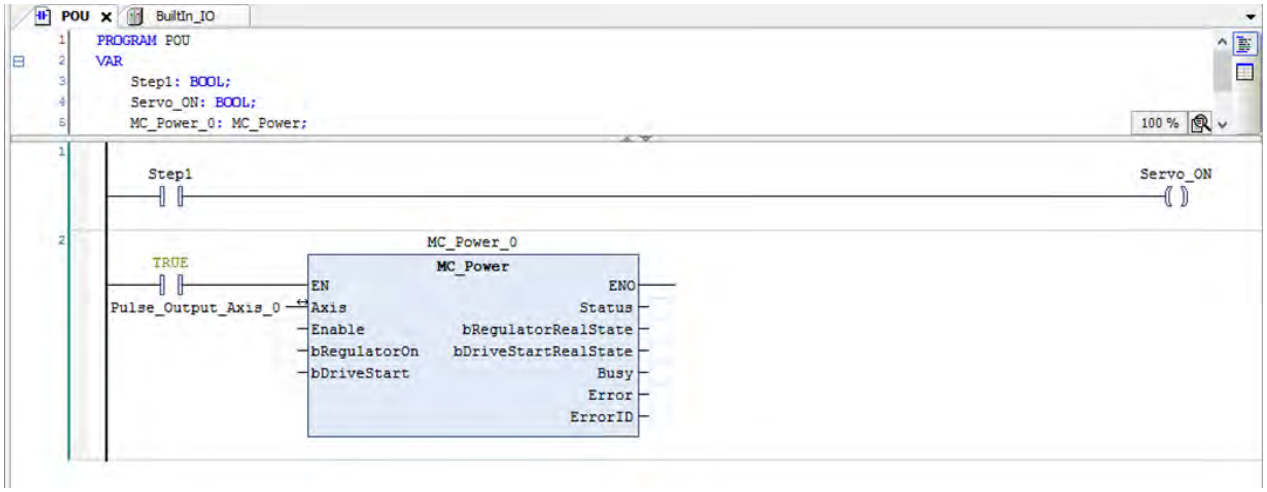
12. Insert the name of pulse train to the Axis input pin of the function block.



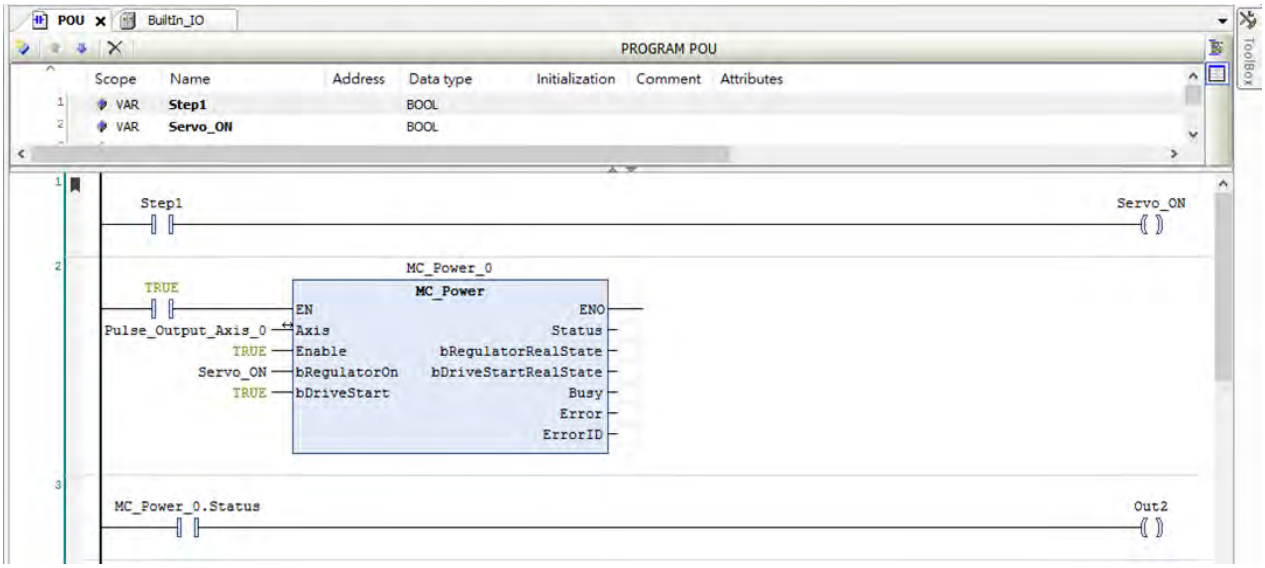
13. Navigate to: “BuiltIn_IO” > “IEC Objects” > “Variable”, then enter “Pulse_Output_Axis_0” as the name of pulse output type axis. In addition, all the newly-added axes will be displayed on this page.



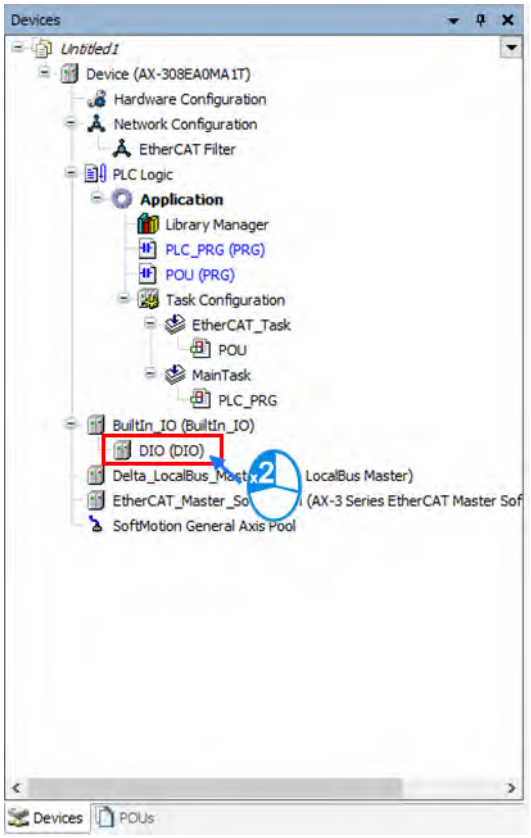
14. Insert the name of servo axis to the Axis input pin of the function block and delete “???” of other pins, since the mark “???” does not represent any variables and errors may occur while parsing.



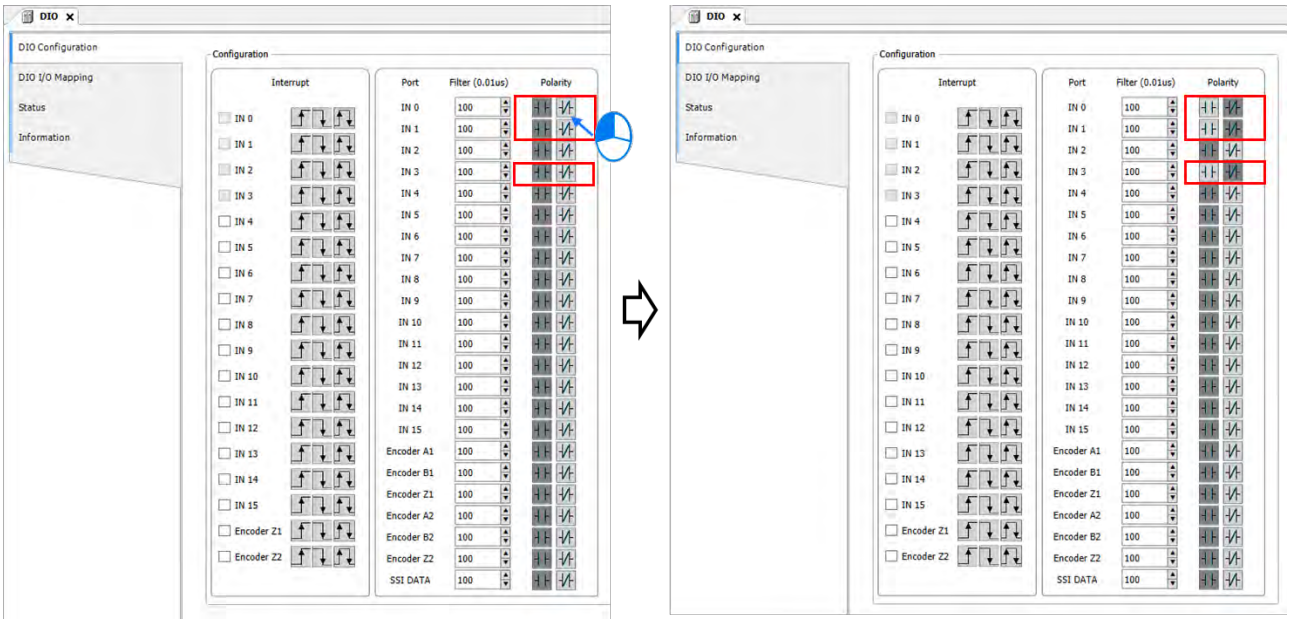
15. Set the status output of the function block to be Out2. (Later, Out2 would need to be configured to connect with a certain position of the external hardware.)



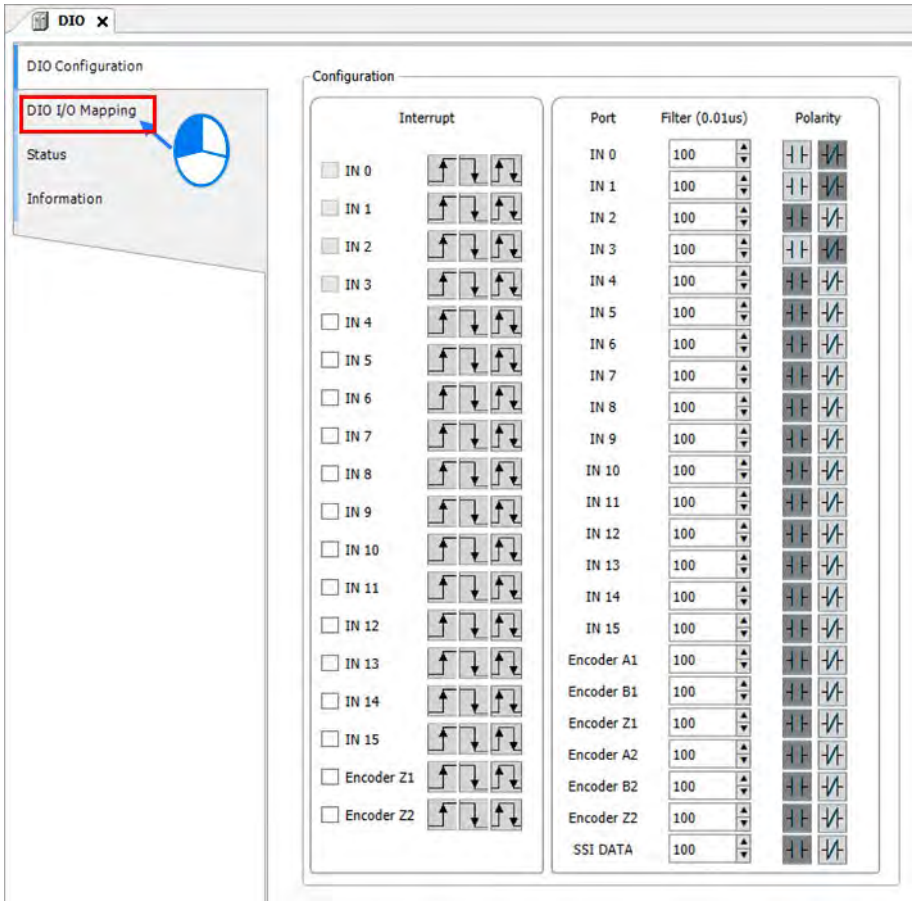
16. Double click on “DIO” in the project tree.



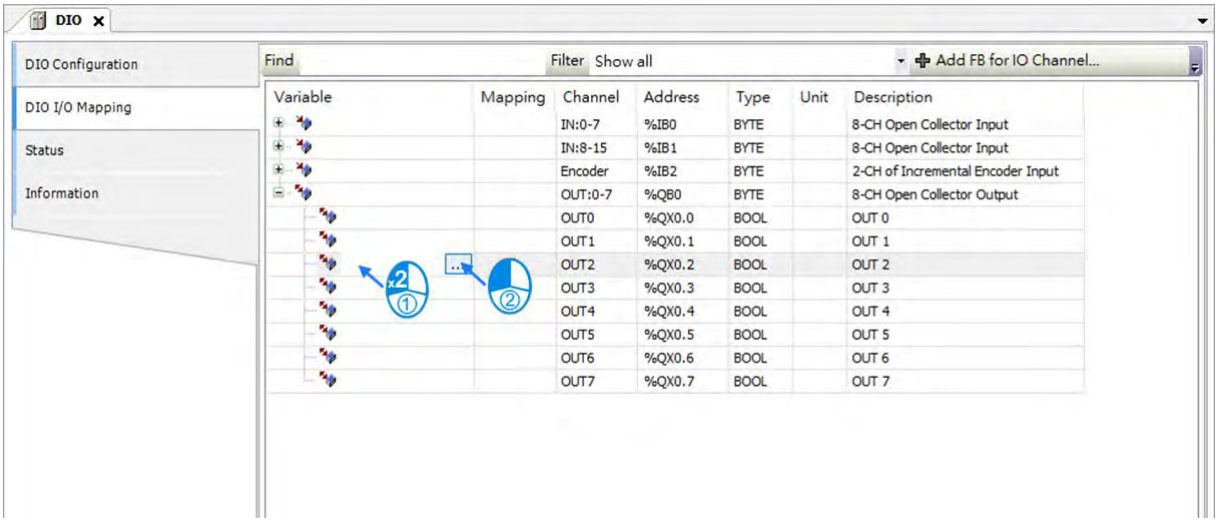
17. Since the contact pins of external hardware are normally-closed, the polarity of IN0, IN1 and IN3 need to be changed to  on the DIO page.



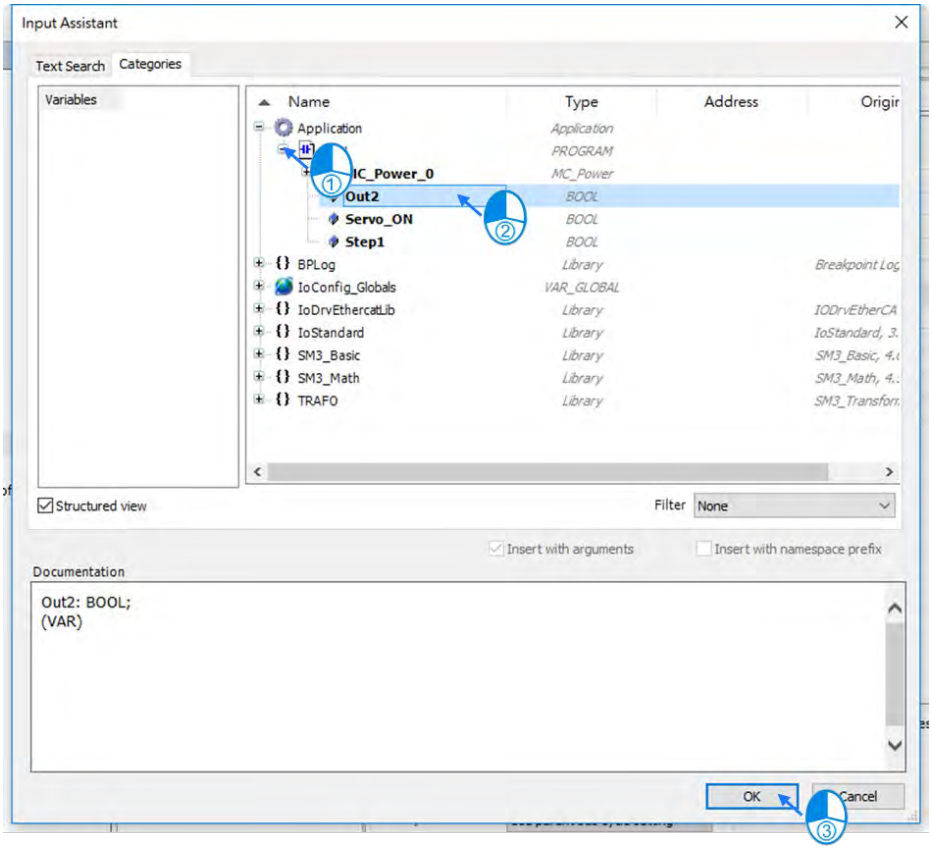
18. Click on “DIO I / O Mapping” tab after the polarity setting is completed.



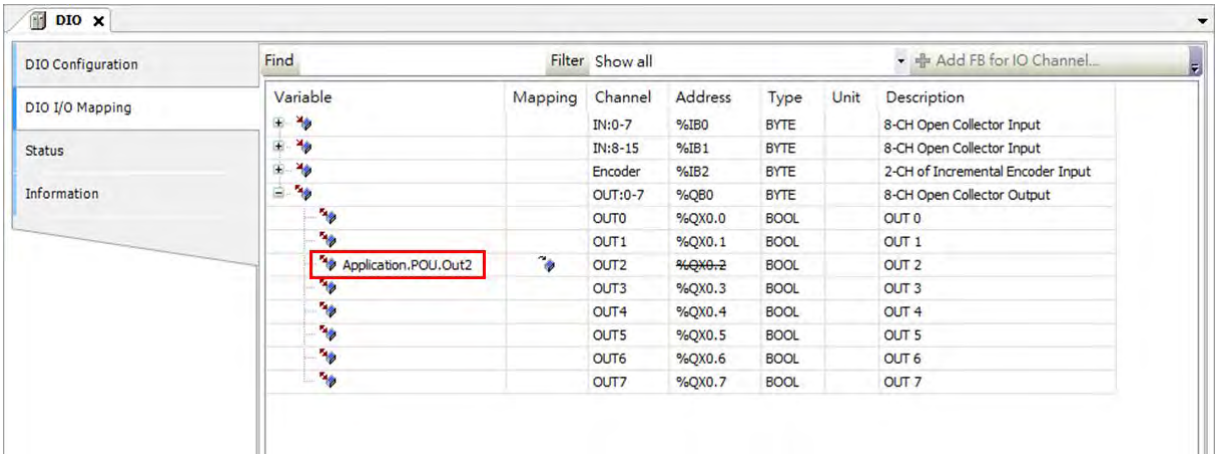
19. Double click on the variable field of OUT2 then click [...] on DIO I / O Mapping page.



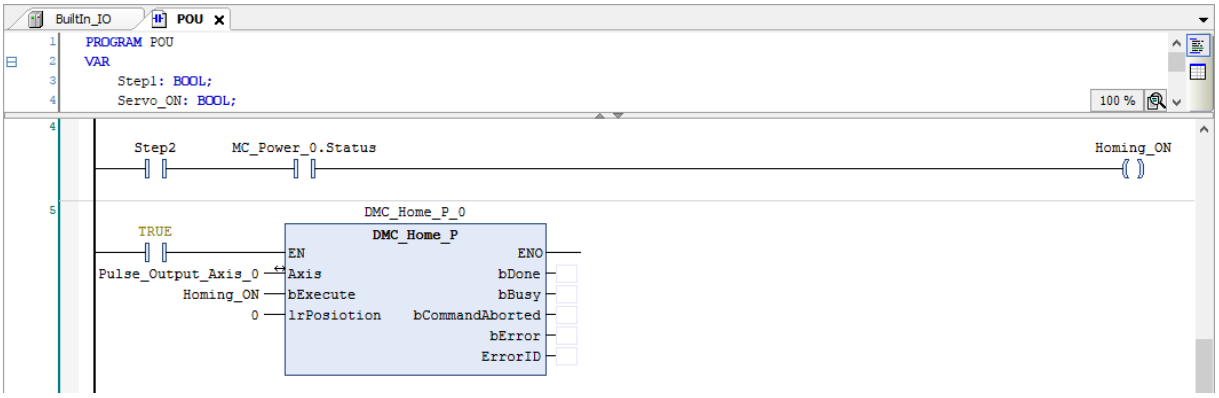
20. Meanwhile, the output pin Out2 of POU needs to be mapped into a physical port (hardware) by expanding POU on the tree and selecting “Out2” then click “OK”.



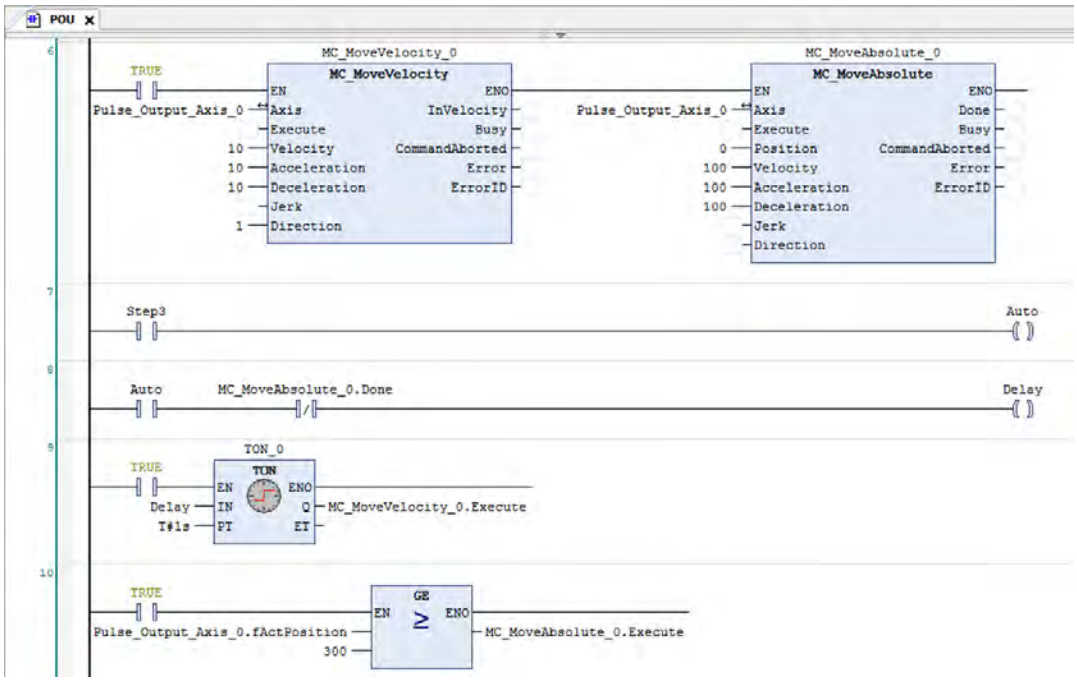
21. After the configuration is completed, the words “Application.POU.Out2” will be shown in the variable field. Out2 will connect to the contactor OUT2 of the external hardware if there’s an output of POU. (The settings of hardware signals need to be configured here, if required.)



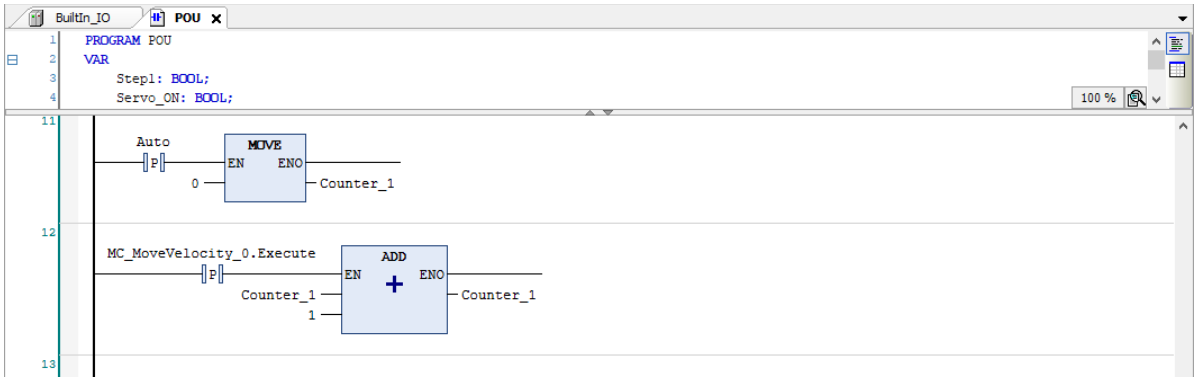
22. Create the function block “DMC_Home_P”, mainly used for axes executing homing, with operations according to previous steps.



23. Create the function blocks “MC_MoveVelocity” and “MC_MoveAbsolute” according to previous steps and set one second delay time for activation.



24. When “Auto” is executed, the value of “ Counter_1” will be cleared and then increased by 1 for each trigger event to MC_MoveRelative.

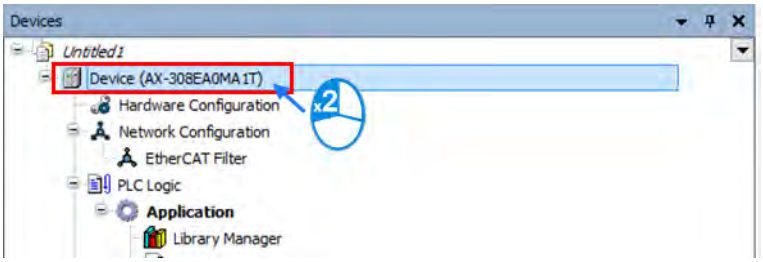


3.3.3 Program Monitoring

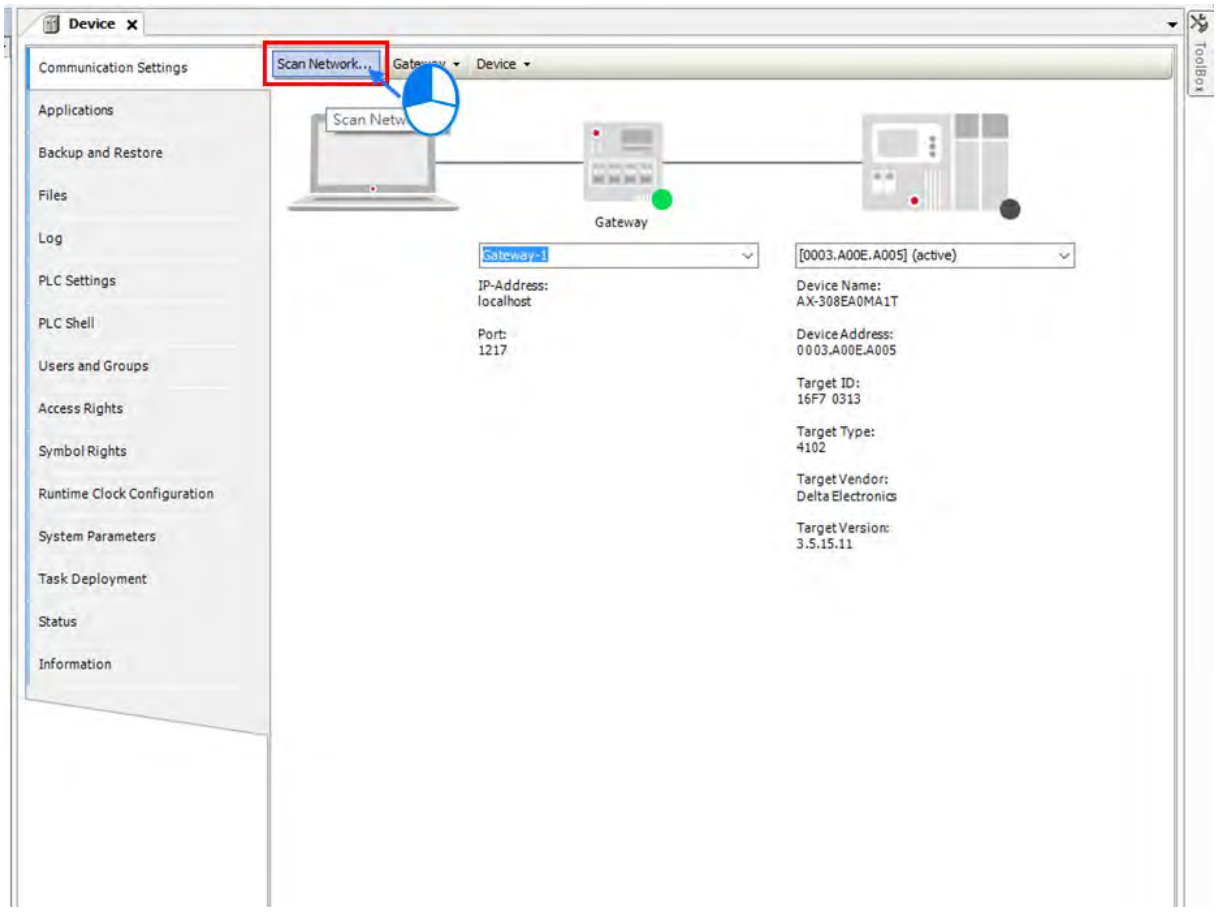
When running a program, the current control status of system can be monitored and part of the device values are allowed to be modified for system testing as well.

3.3.3.1 Setup Connection between Devices

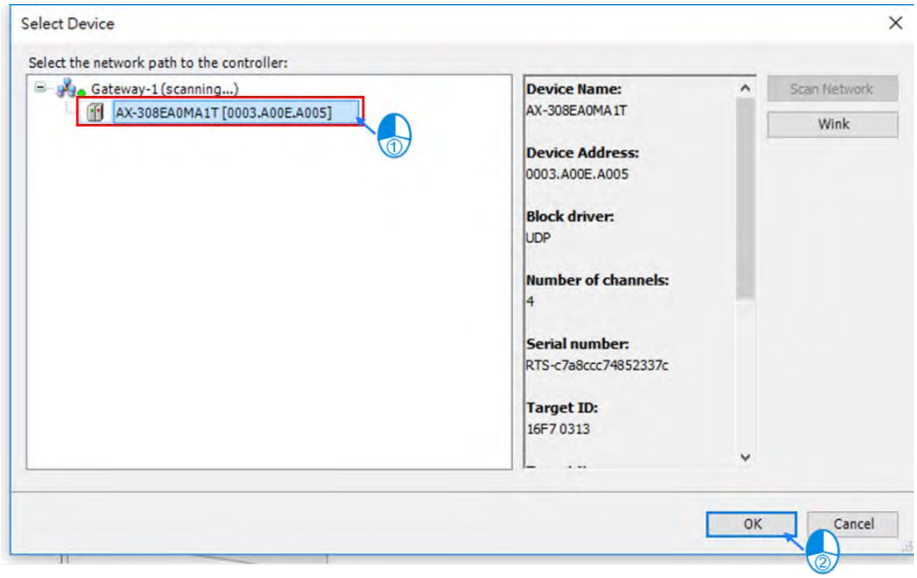
1. Double click on “Device”. (The default IP address for AX-308E is 192.168.1.5)



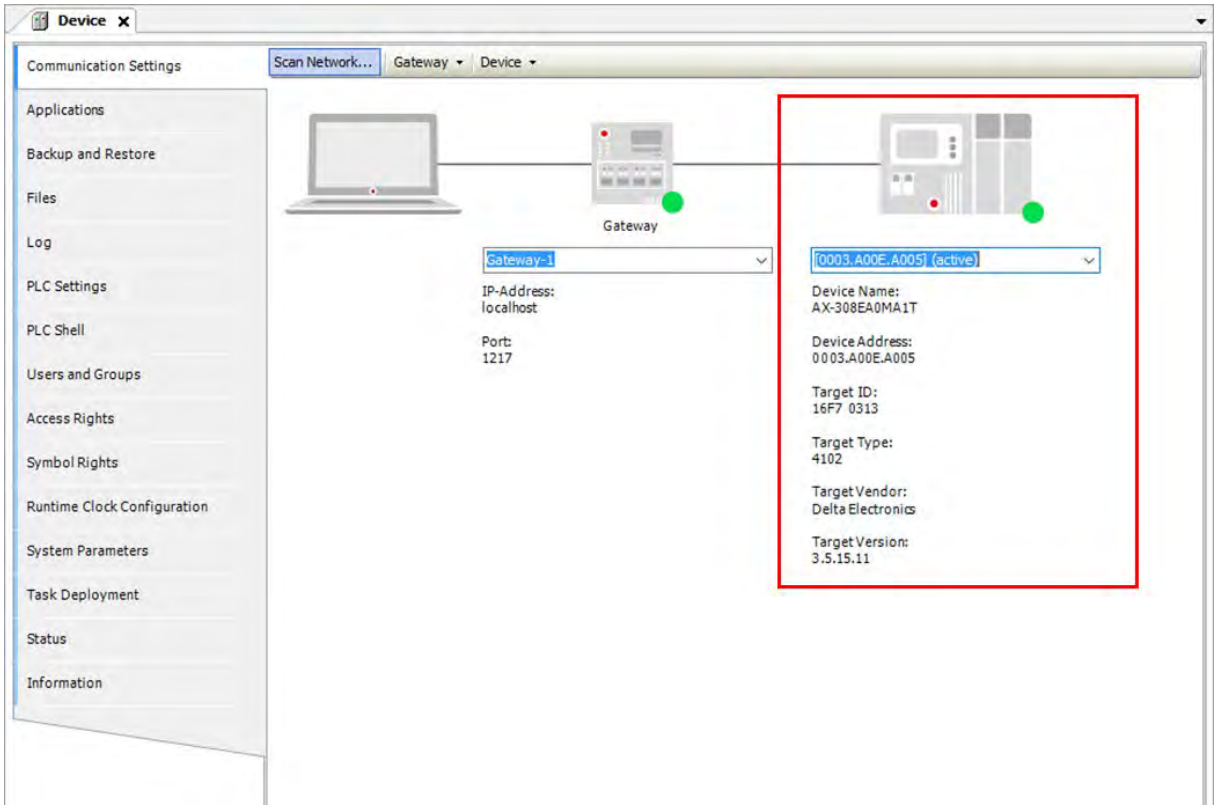
2. Choose “Scan Network” on the Device page.



3. After the “Select Device” window pops up, select the “AX-308E” device and click on “OK”.

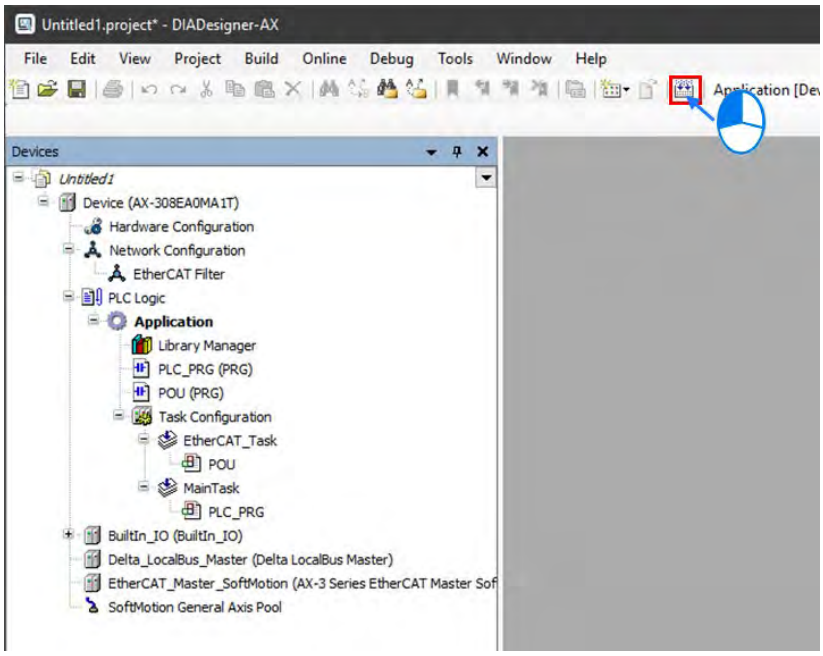


4. When the device is connected with your PC, information of the device will be displayed as shown in the following figure marked by the red box.

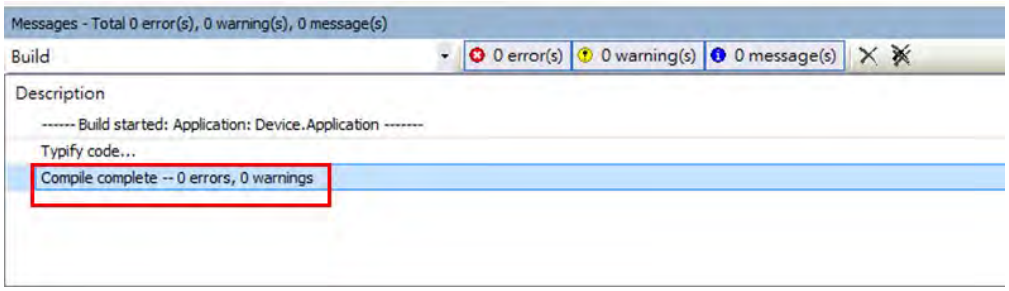


3.3.3.2 Process Monitoring and Control

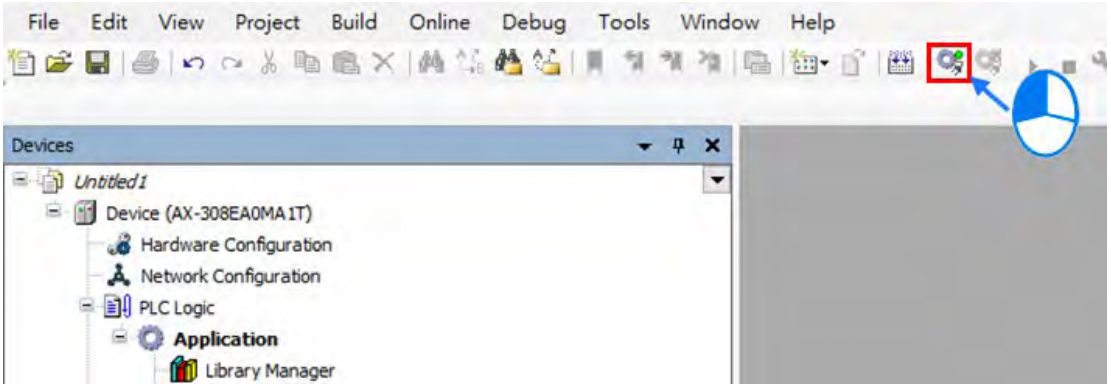
1. Click on the Compile button  to verify the correctness of program.



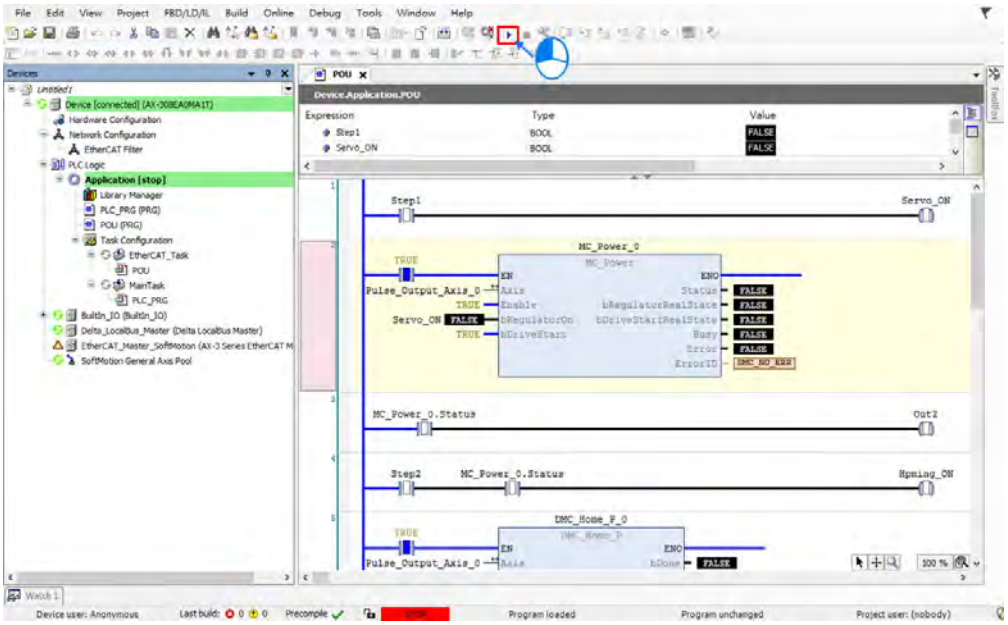
2. Upon completion of compiling, a report message of programming errors and warnings is displayed.



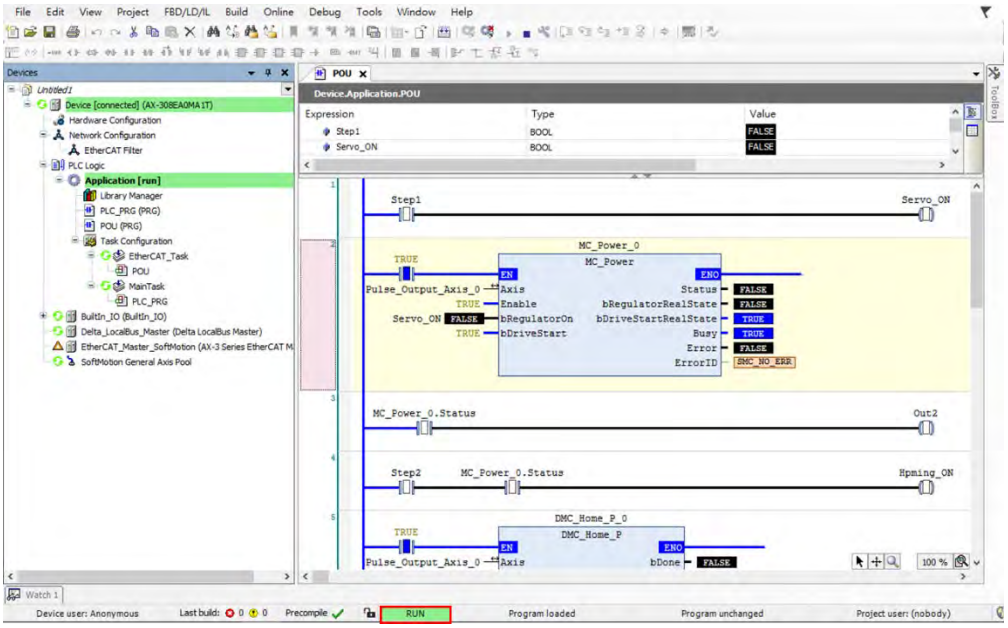
3. And then click on  to perform online monitoring



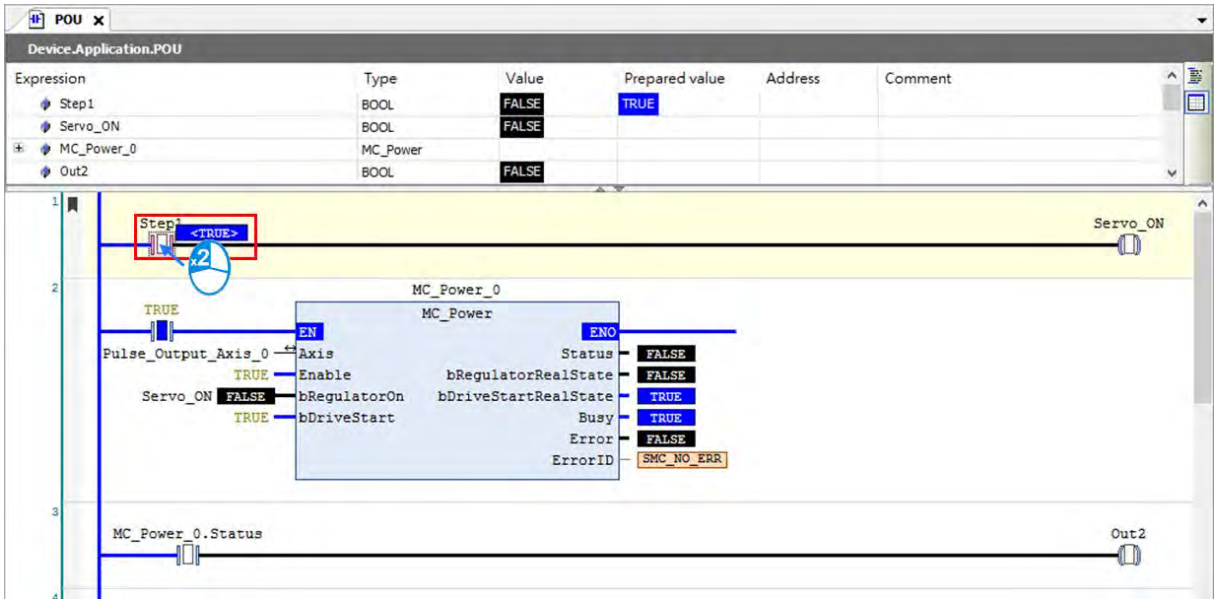
4. The program needs to be downloaded after monitoring action is performed. When the download is completed, the status of PLC shows STOP until you click on and the status would shift to RUN.



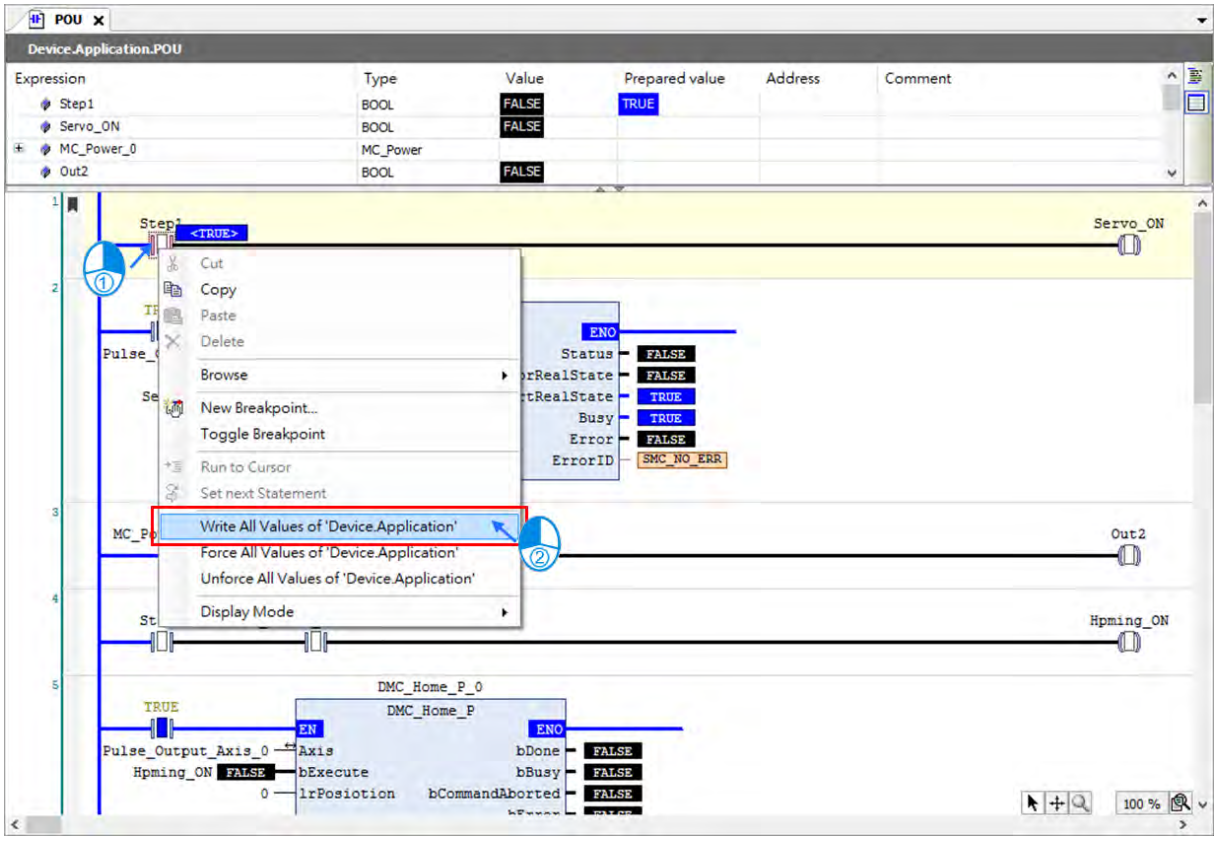
5. When the PLC is in the RUN mode, the status would be displayed as RUN at the bottom of the page and the program is ready to be executed.



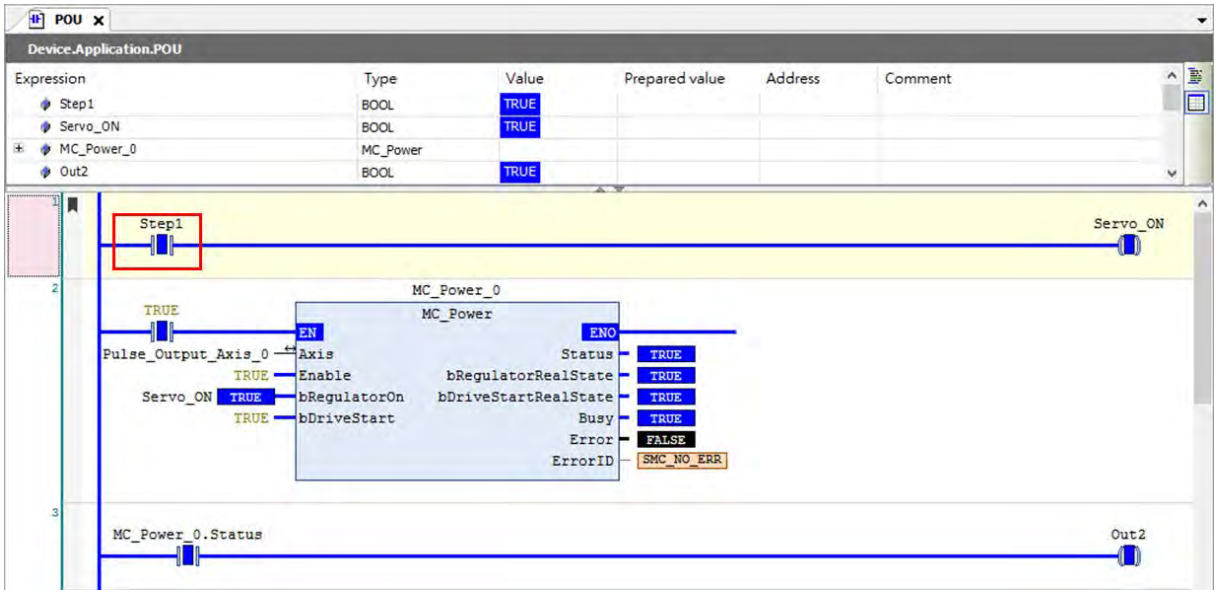
6. Double click on “Step 1” and the icon <TRUE> will be shown.



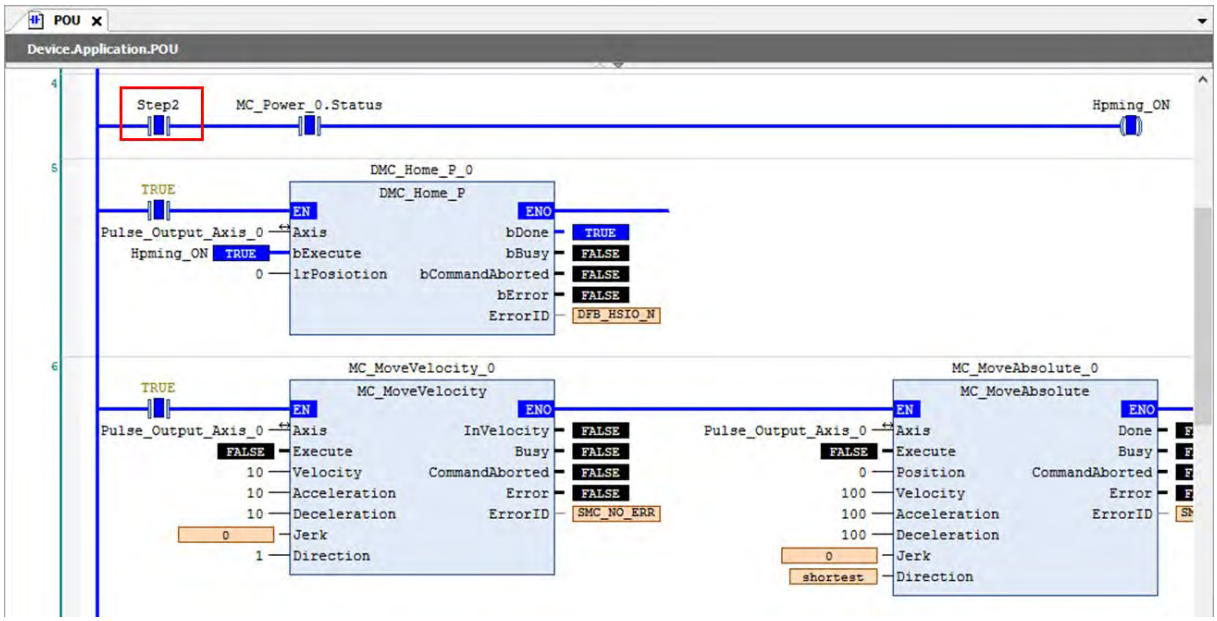
7. Right click on “Step 1” to select “Write All Values of ‘Device.Application” and all the corresponding status will be filled in, which can also be performed by using the shortcut key Ctrl + F7.



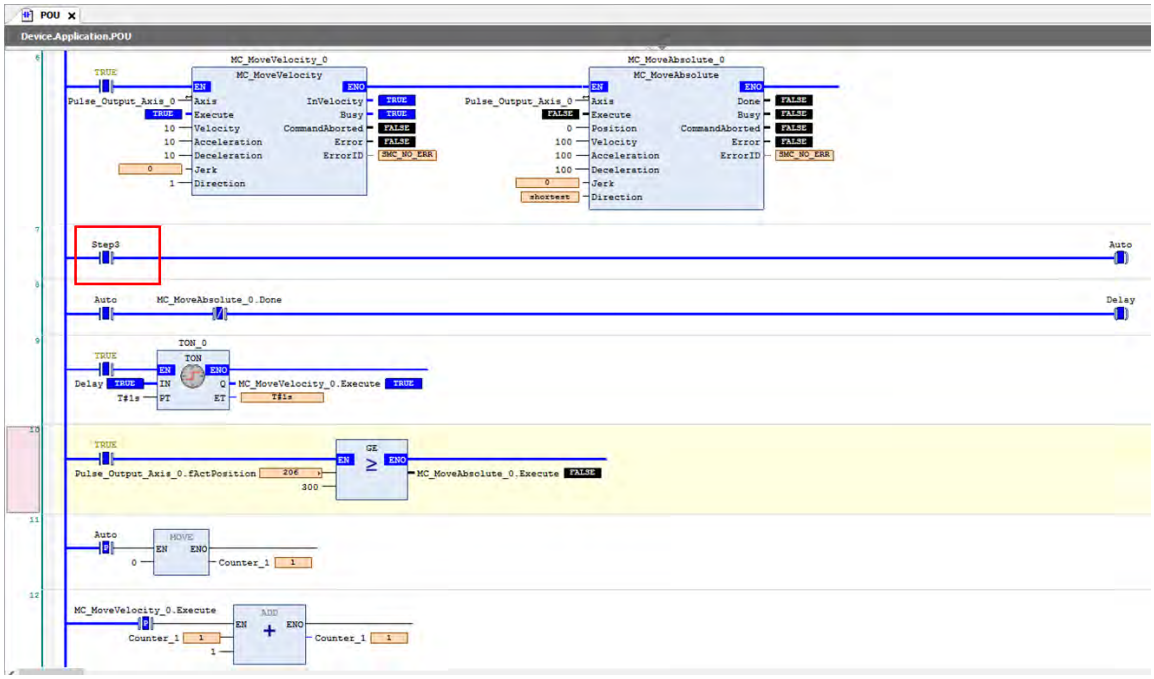
8. After activating “Step 1”, Servo ON will be TRUE. (Same operation method to execute the following programs.)



9. Start homing for the axis by activating “Step 2”

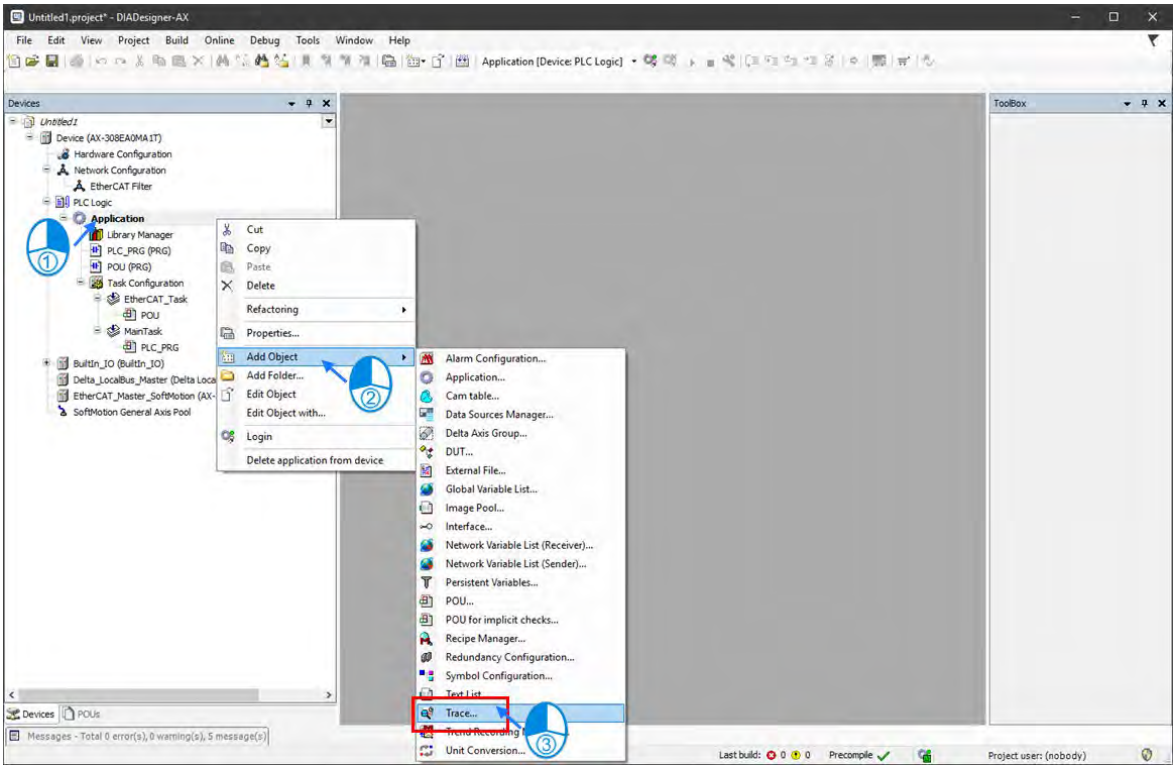


10. After starting "Step 3", execute “MC_MoveVelocity “ to drive the axis moving to position 300mm at a constant speed, then execute “MC_MoveAbsolute” to reverse the axis back to position 0. In addition, the value of Counter_1 will be increased by 1 for each rotary completed, until “Step 3” being turned off.

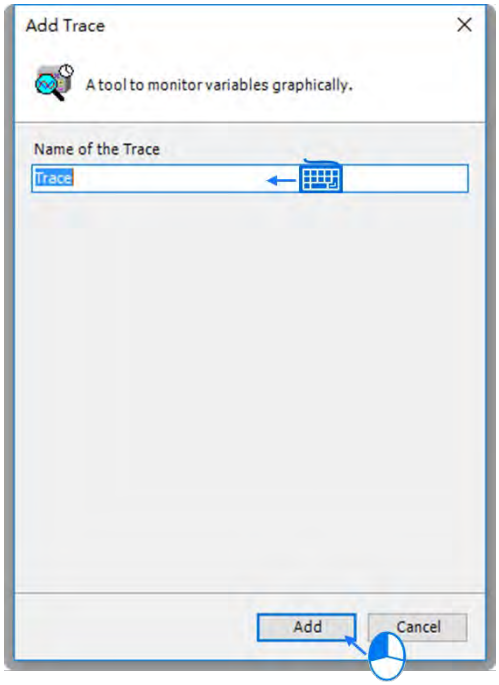


3.3.3.3 Monitor with Trace Tool

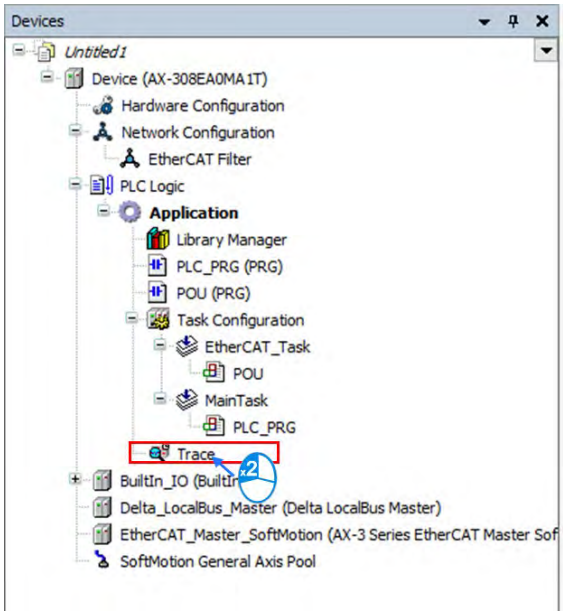
A Trace makes it possible to record the value history of variables on the PLC, just like a digital sampling oscilloscope. To monitor the status of input/ output pins and axes with Trace function while executing the program, navigate to “Application” > “Add Object” > “Trace”. Then you’ll open the “Add Trace” window.



1. Insert the name of Trace then click “Add” on “Add Trace” window.

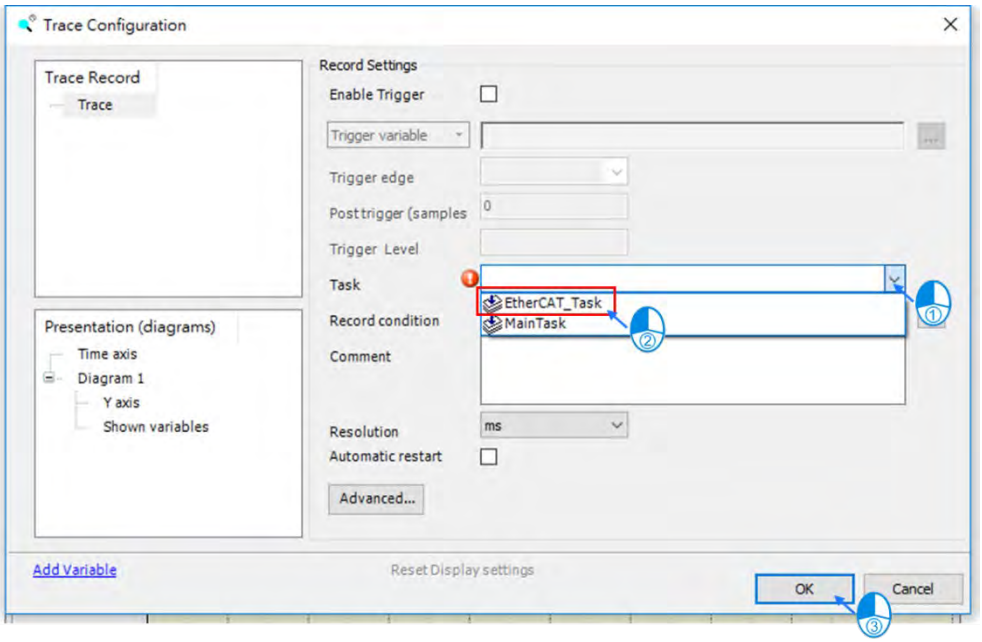


2. After the Trace is successfully added, it will be shown on the project tree on the left side of the screen. Then double click on “Trace” to open the Trace page.



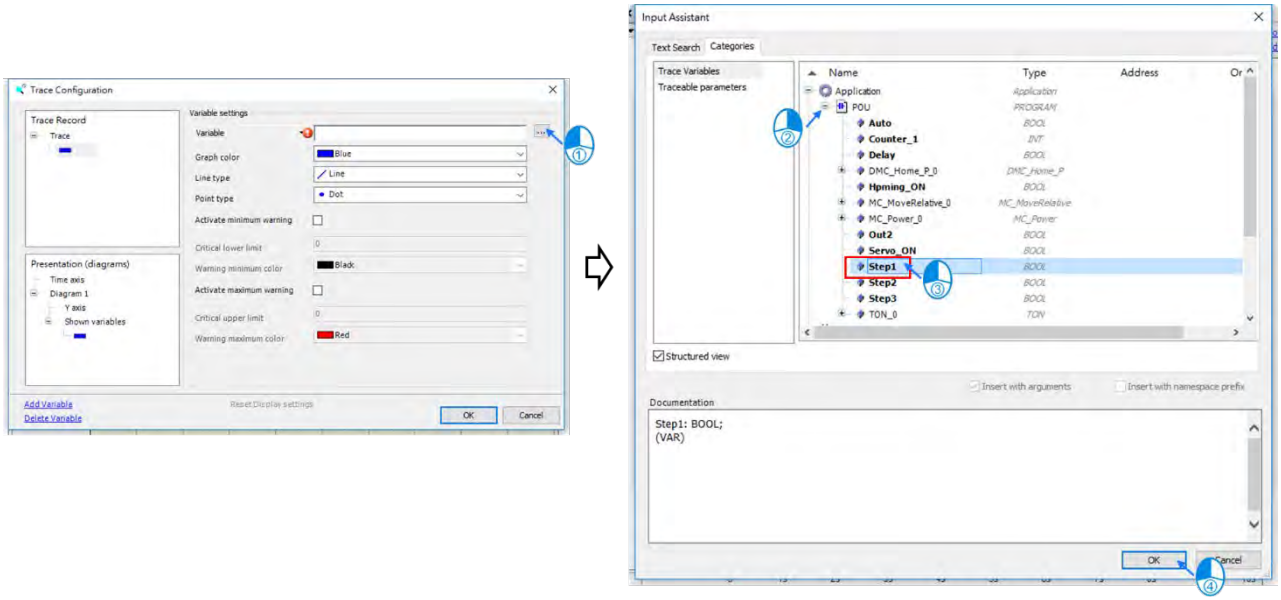
3. Click “Configuration” on the Trace page to open the configuration window.

4. Select “EtherCAT_Task” from the drop-down list of Task on the Trace Configuration window.

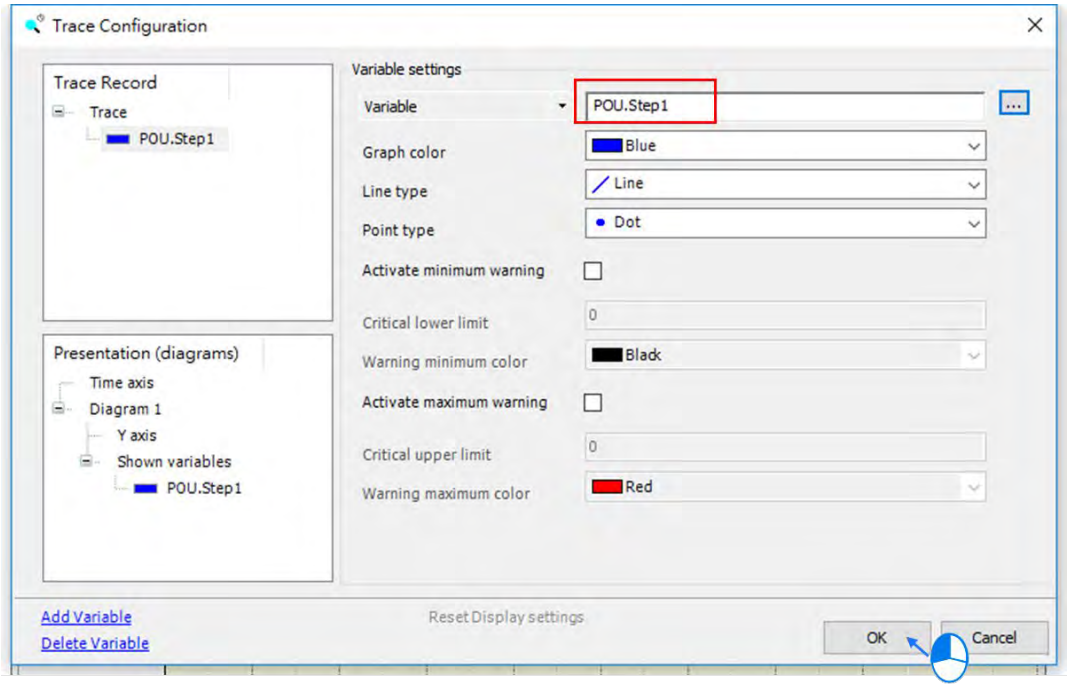


5. After finishing configuration, click “Add Variable” on the Trace page to open the Trace Configuration window.

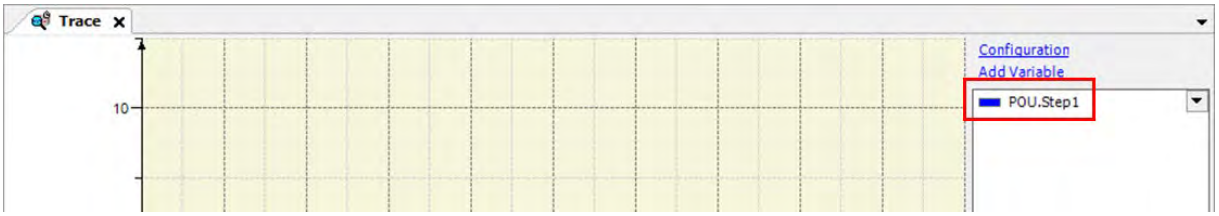
6. Click the button ... on the Trace Configuration page to add the required trace variables or traceable parameters. After the Input Assistant window is opened, expand POU on the tree and select “Step 1” then click on “OK”.



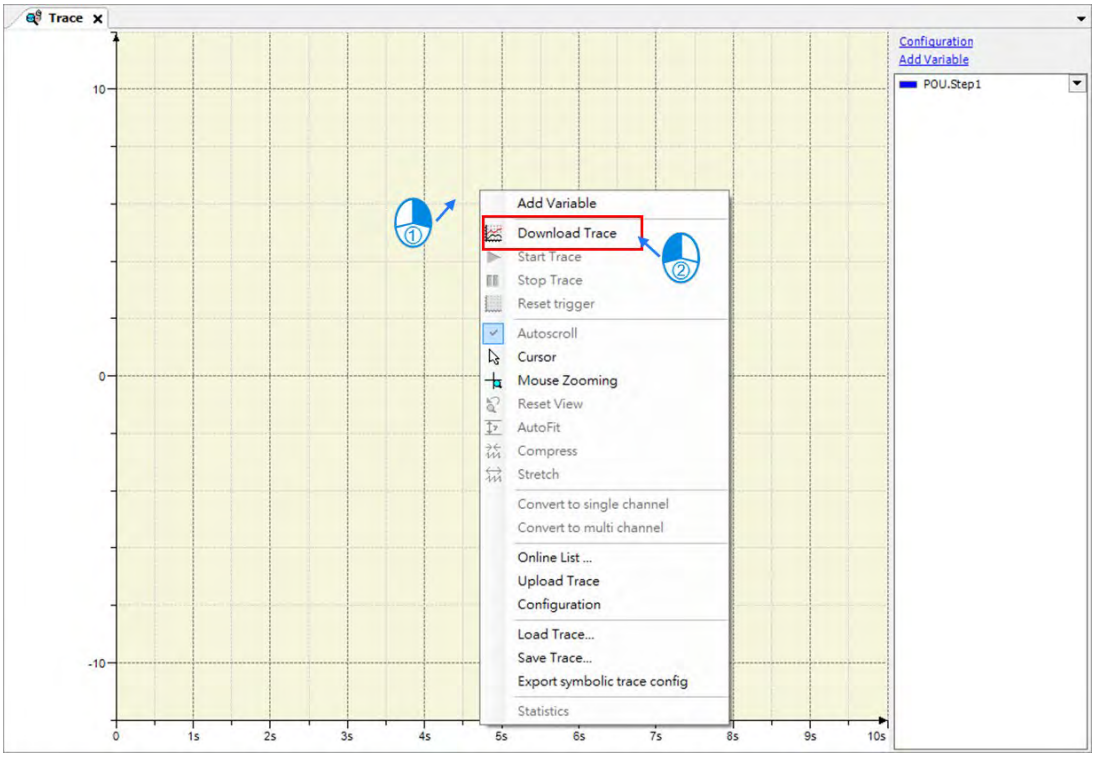
7. Make sure the chosen variable is shown in the variable field of variable settings and then click “OK”.



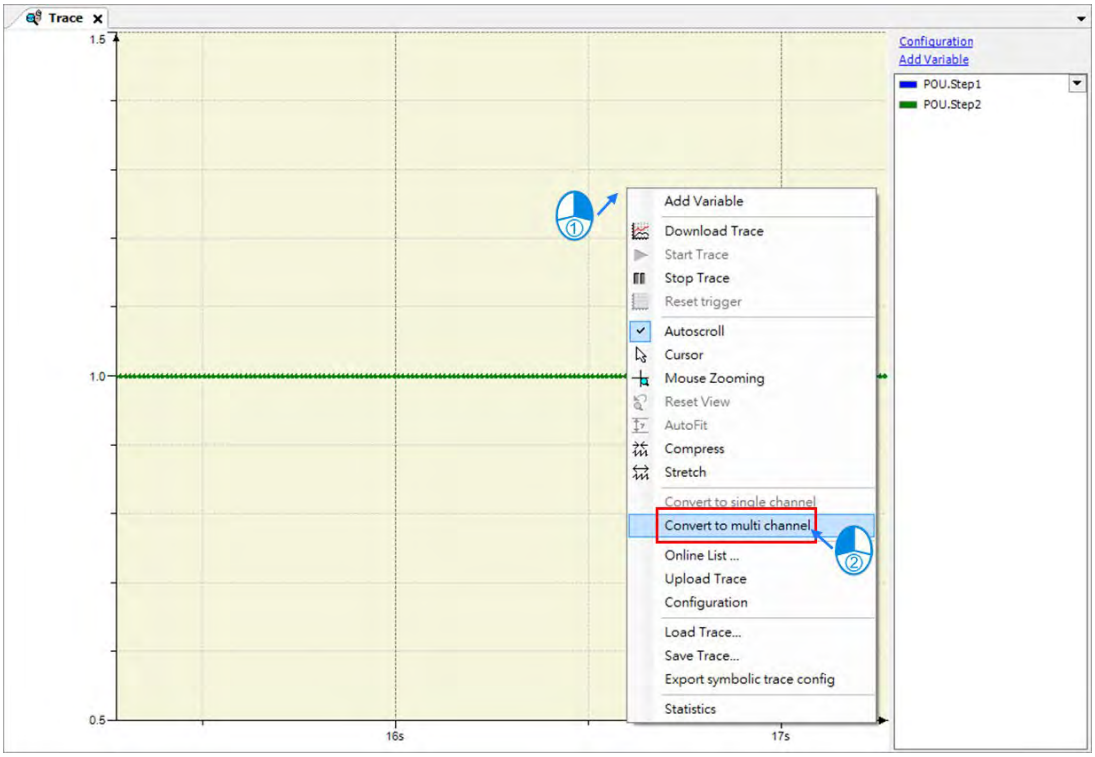
8. Afterwards, the added variable will be shown on the right side of the Trace page.



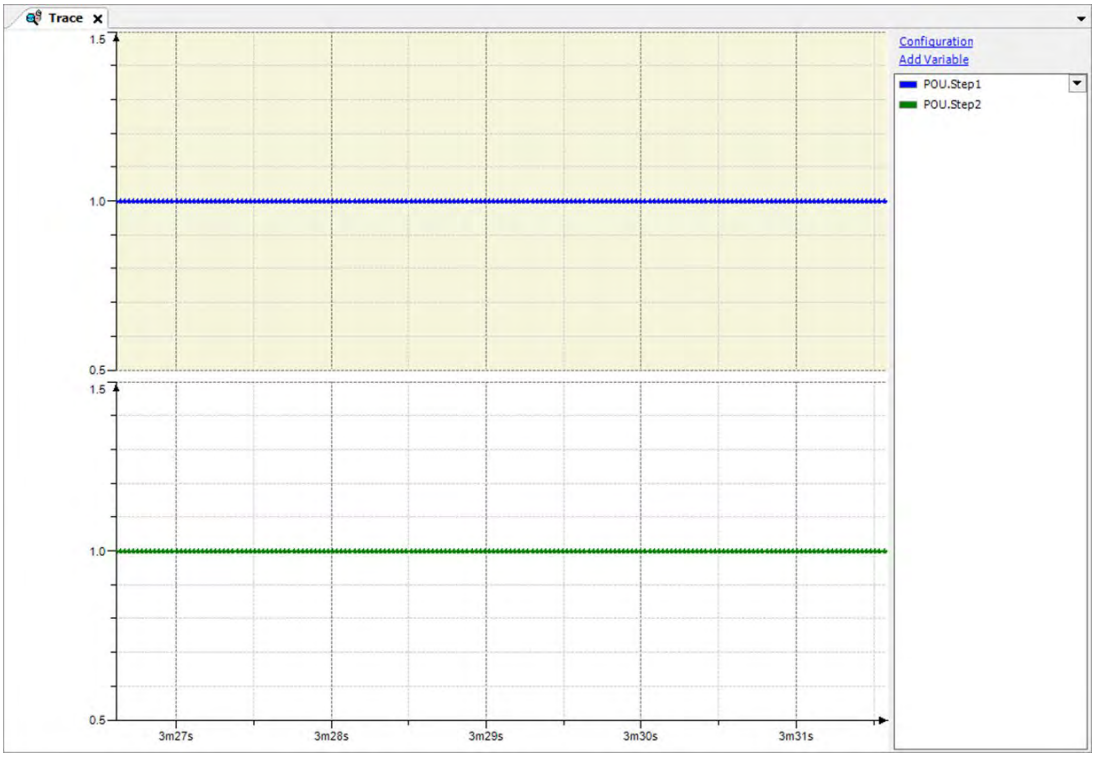
9. Right click on the Trace and select “Download Trace” to start monitoring.



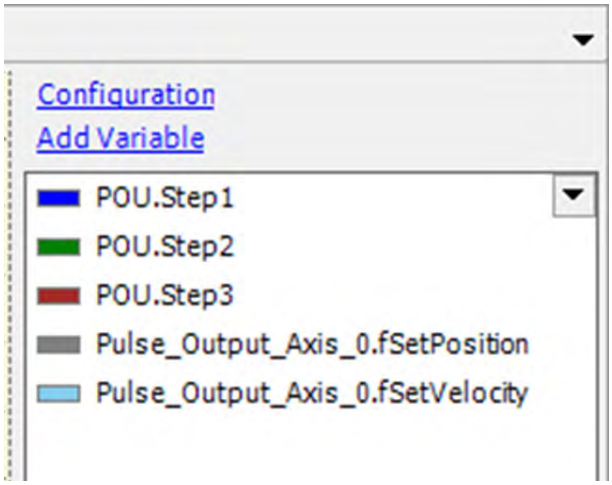
10. In case that there’re more than one variable or parameter need to be monitored, you can just right click on the scope and select “Convert to multi channel”



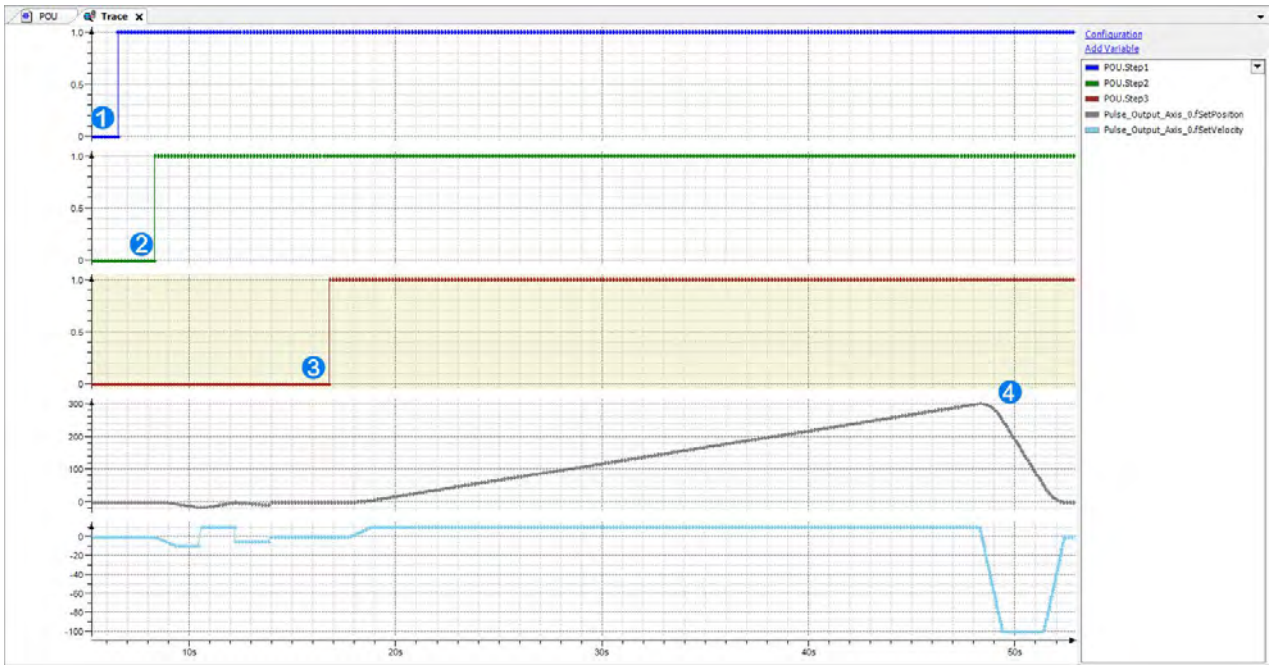
11. After choosing “Convert to multi channel”, two data-recording oscilloscope of the two chosen variables added on the right will be displayed on the page.



12. Repeat the above steps to add more required trace variables and traceable parameters.



13. While a program is running, you can observe trigger events and the current position of axis as well as the velocity via the data-recording oscilloscope



Introduction of the process shown on the scope:

① Execute “Step1” to activate the pulse type axis

② Execute “Step2” to make the pulse type axis perform homing.

③ Execute “Step3” and wait for one second to trigger “MC_MoveVelocity” and perform positive rotation.

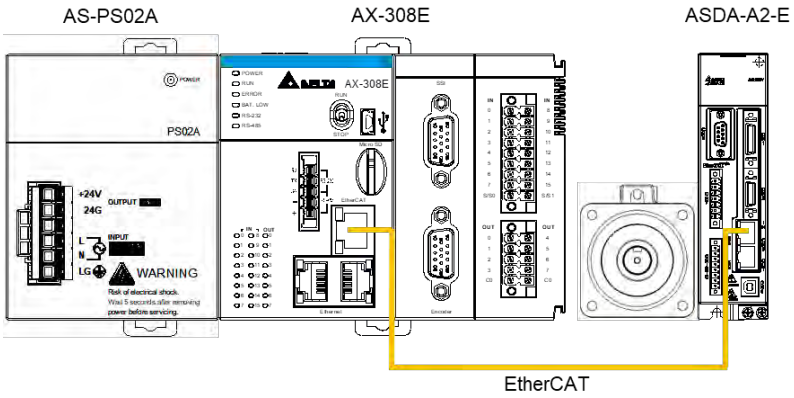
④ Once the axis passes the position 300, MC_MoveAbsolute will be triggered to reverse the axis back to position 0 as well as aborting the previous function block.

3.4 Setup Projects with Virtual Master Axis

In this example, a virtual axis is presented to be the master axis with the servo motor adopting the master-slave control function.

3.4.1 System Hardware Configuration

AX-308 connects to a servo driver (ASDA-A2-E) directly through EtherCAT communication port to control servo motor positioning which follows the virtual master axis.



- Tools and materials

Numbering	Item	Model Number	Q'ty
1	AS Series Power Module	AS-PS02A	1
2	AX-3 Series CPU Module	AX-308EA0MA1T	1
3	Power supply	DVP-PS02	1
4	Servo drive (EtherCAT)	ASDA-A2-0421-E	1
5	Servo motor	ECMA-C10604ES	1
6	Servo power cable	ASD-ABPW0003	1
7	Servo encoder cable	ASD-ABEN0003	1
8	EtherCAT communication cable (1 meter)	UC-EMC010-02A	1

- Configuring hardware limits for ASDA-A2-E

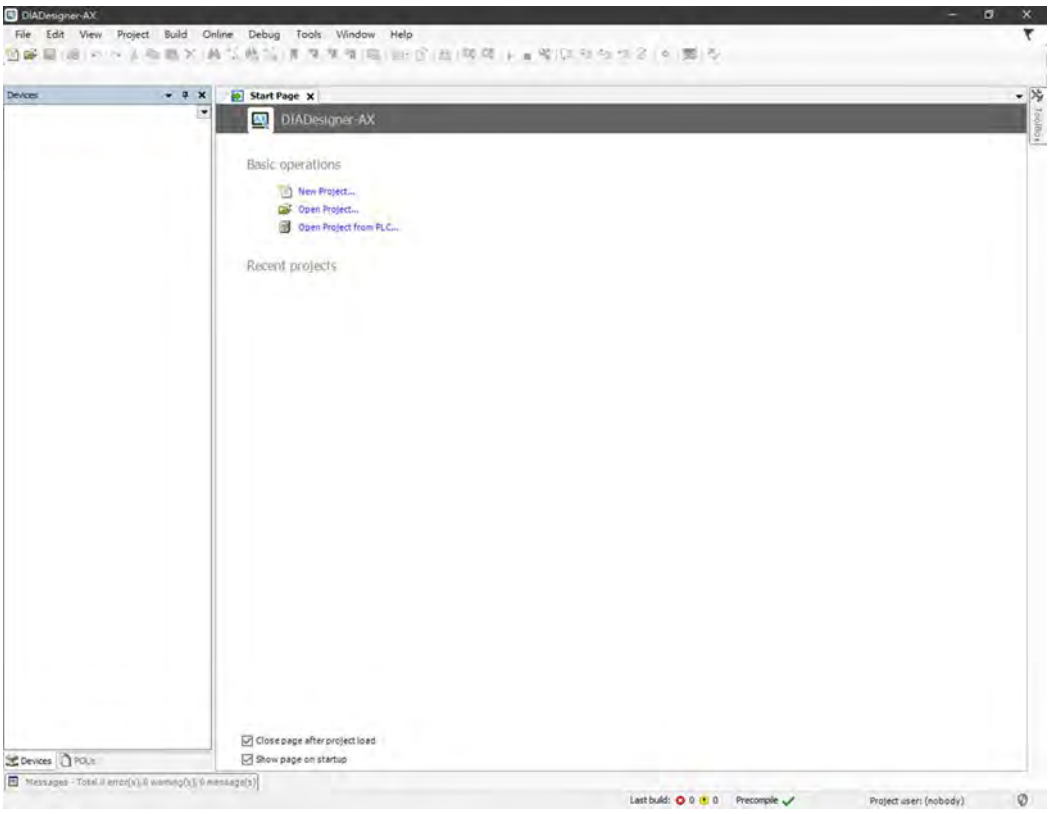
Hardware limit is not required in this example, which features rotary axes.

3.4.2 Create New Projects and POUs

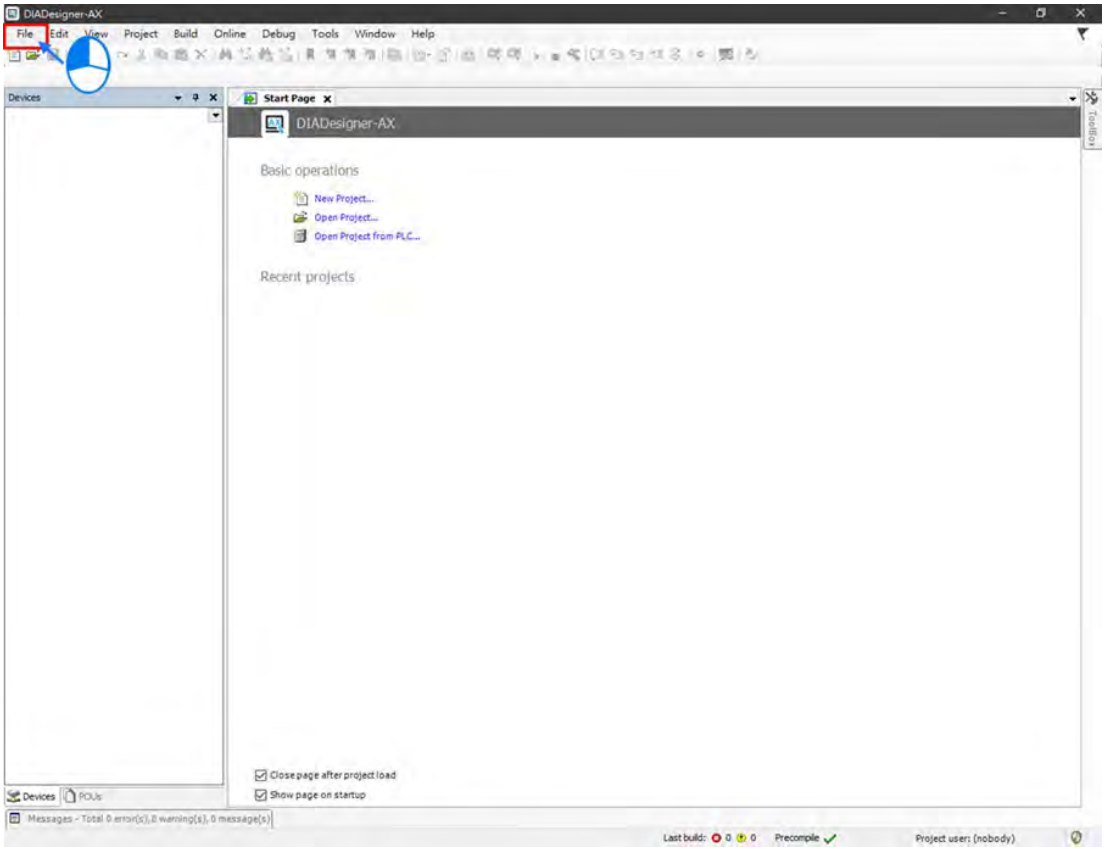
3.4.2.1 Create a New Project

1. First, open the programming software DIADesigner-AX.

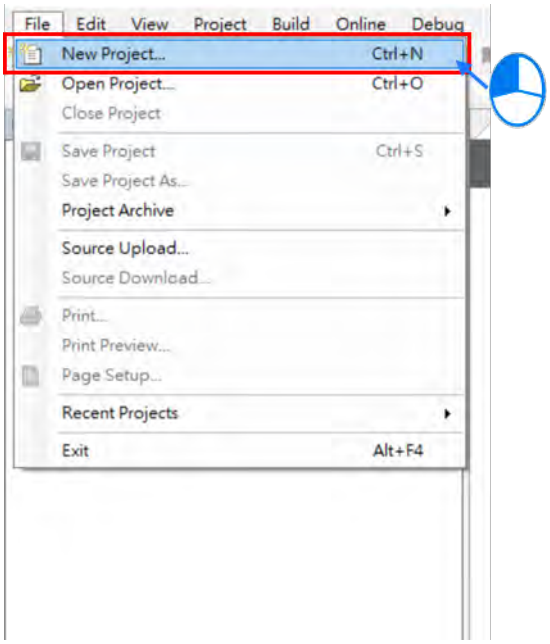
2. The display of DIADesigner-AX is shown as below.



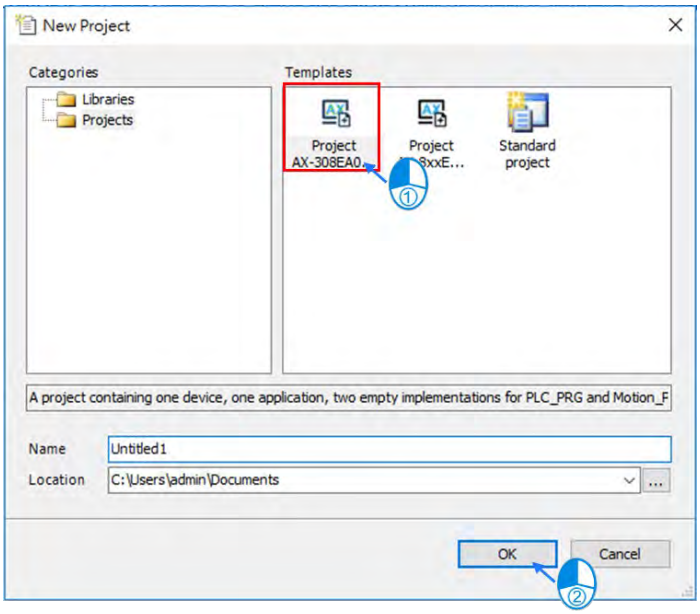
3. Click on “File”.



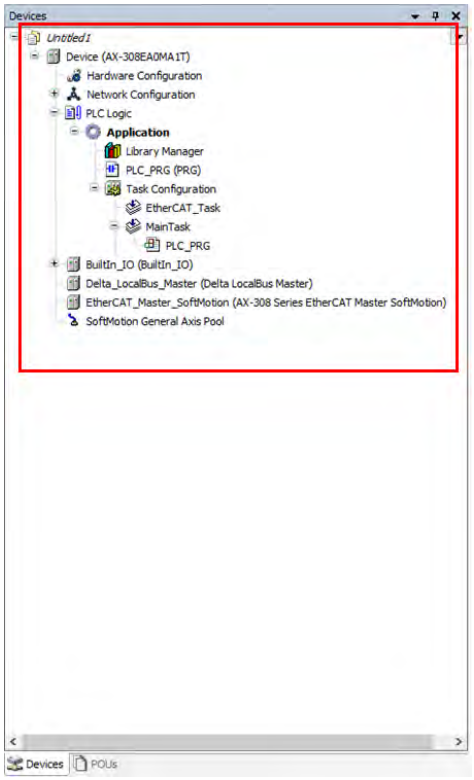
4. To create a new project, select "New Project" from the drop-down menu for DIADesigner-AX V1.0.0 or select “Standard Project” from the drop-down menu for DIADesigner-AX V1.1.0 or later.



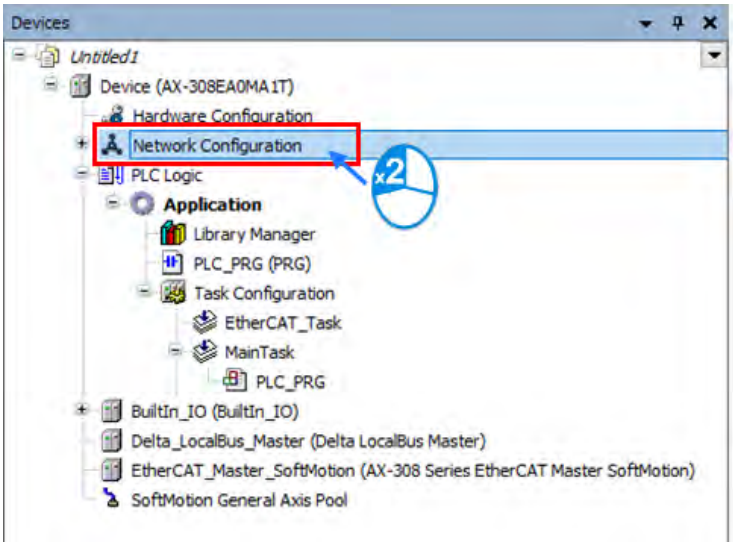
5. Select “Project AX-308EA0MA1T” after entering New Project page. Name the project in the field of Name and choose a path to the archive location, then click OK to move to the next page.



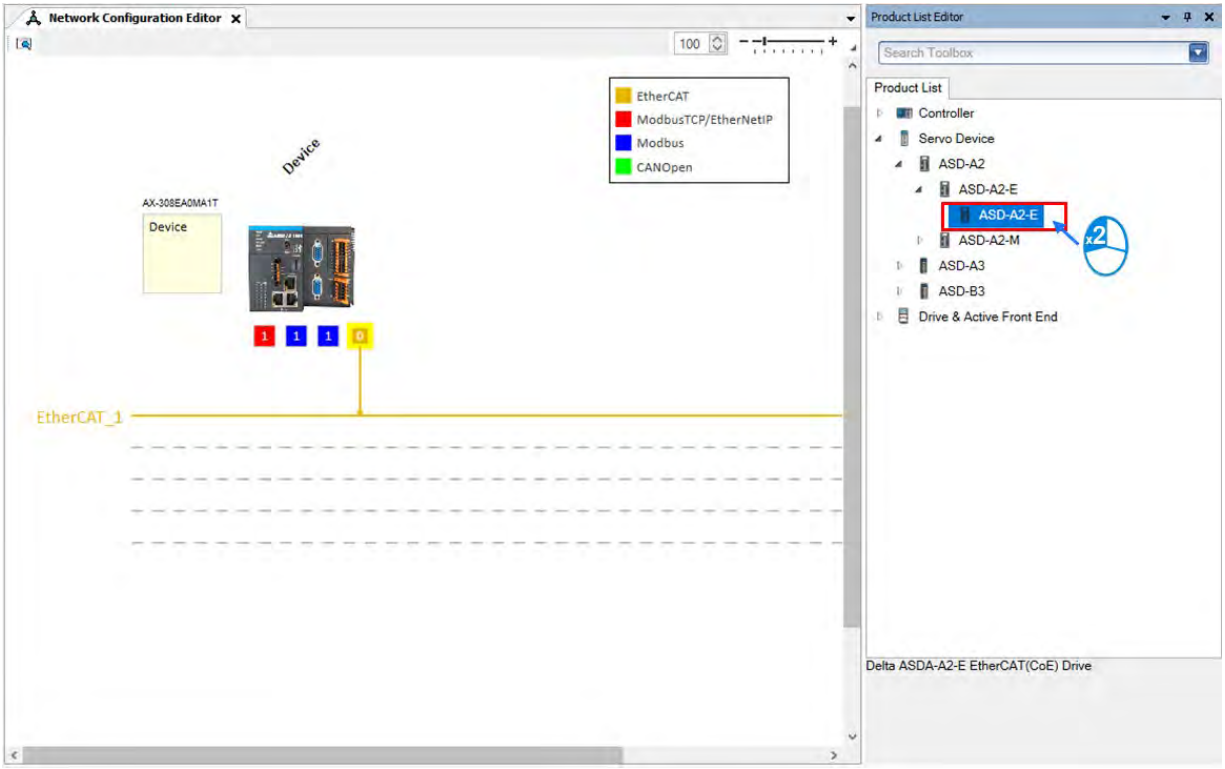
6. After entering the project page, double click the target object in the project tree on the left side of the page to open a certain program or configure settings of relating modules



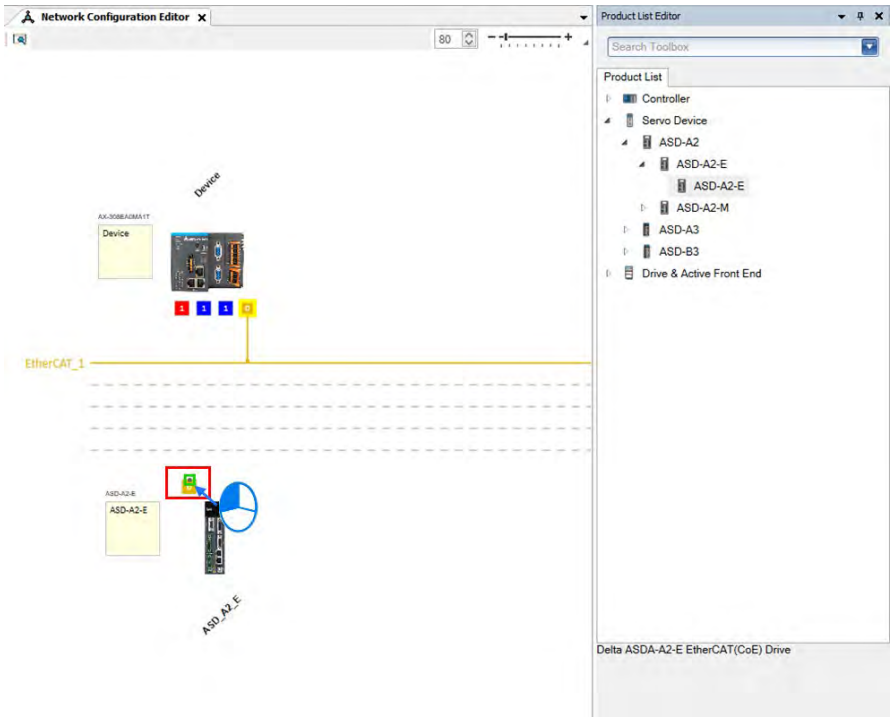
7. Double click on “Network Configuration” to create servo axes in the system.



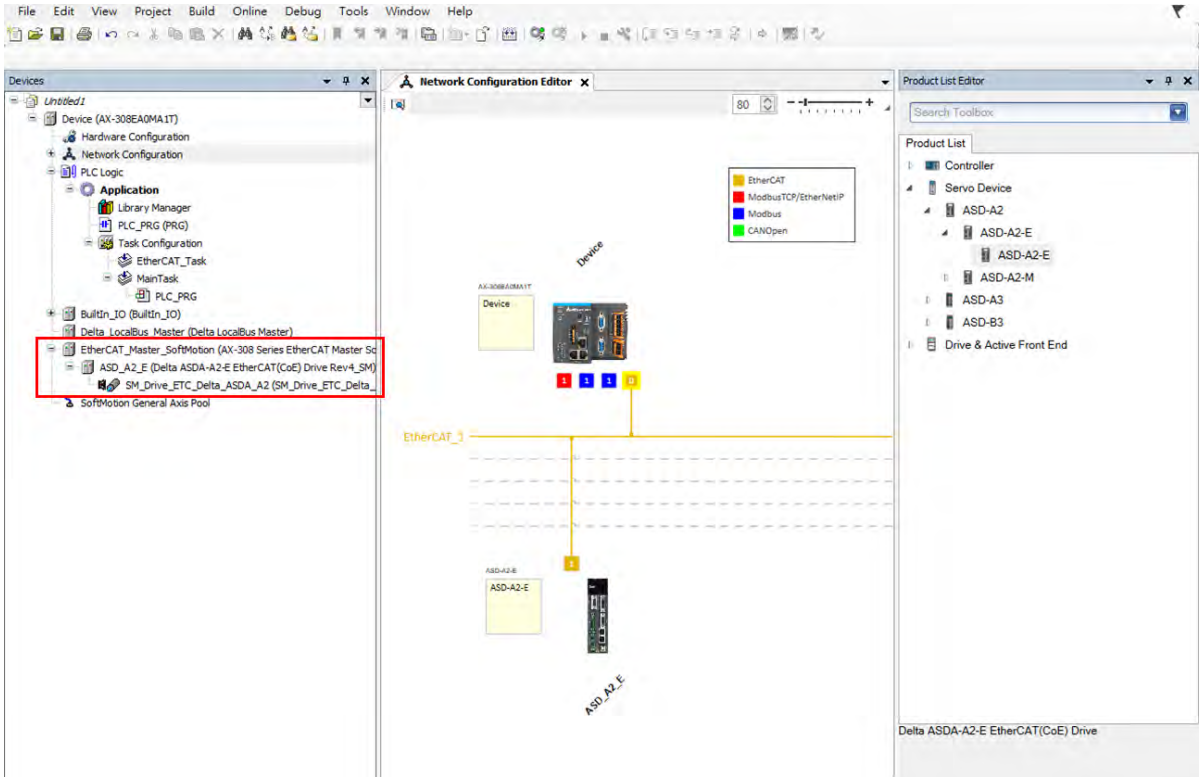
8. After entering “Network Configuration” page, select the servo drive slave based on EtherCAT and double click on it to add new axes



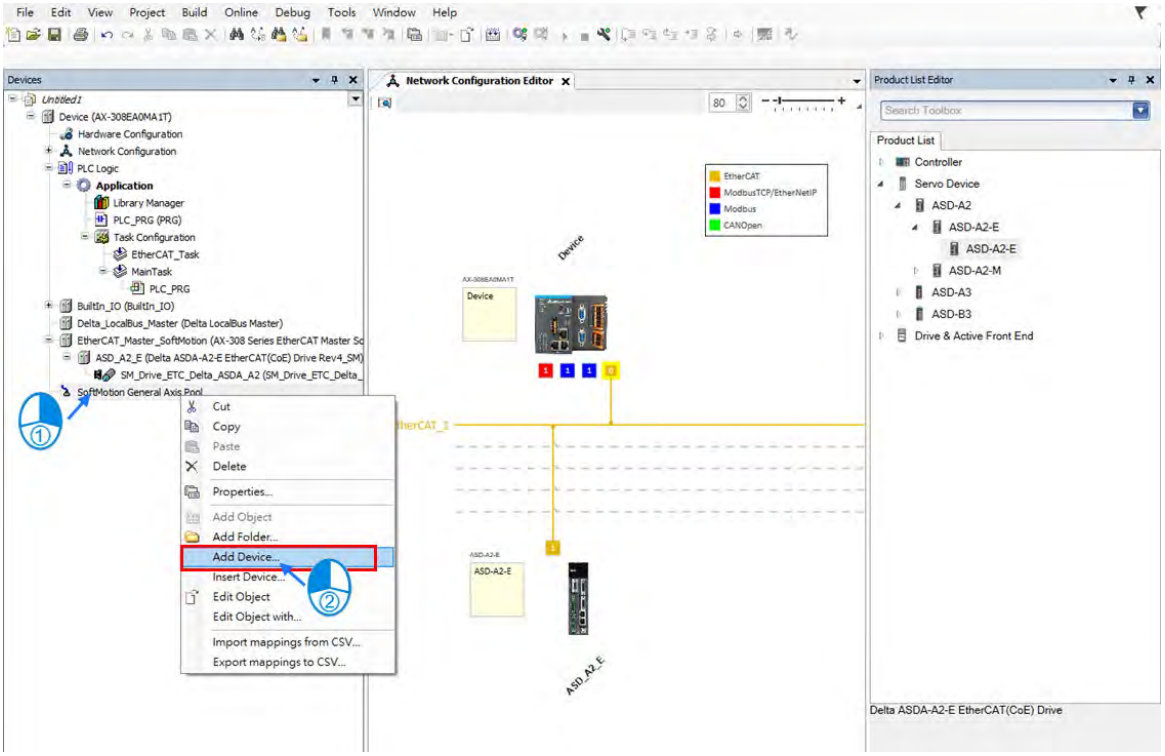
9. Move the cursor onto 0, then click and hold the left mouse button and drag the yellow line to connect with the line of EtherCAT_1



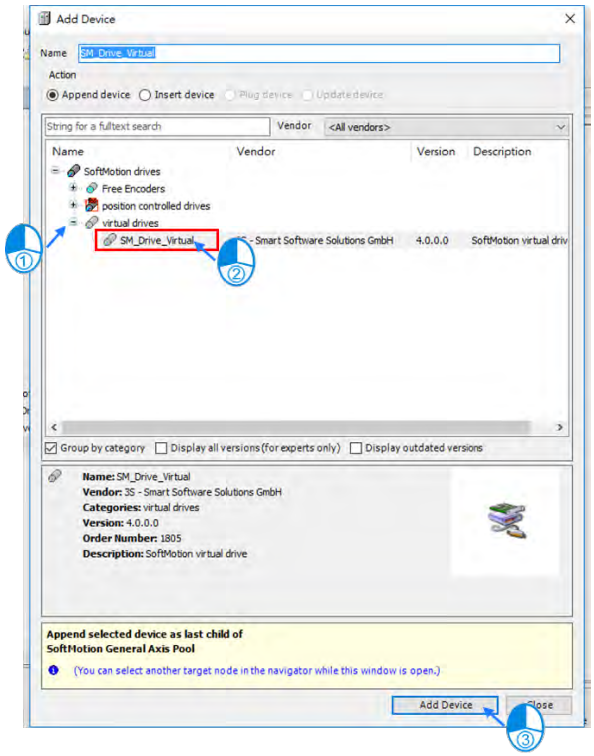
10. After connecting with EtherCAT_1, the slave device is displayed in the project tree (fields with red borders) which also means it is successfully configured with EtherCAT communication.



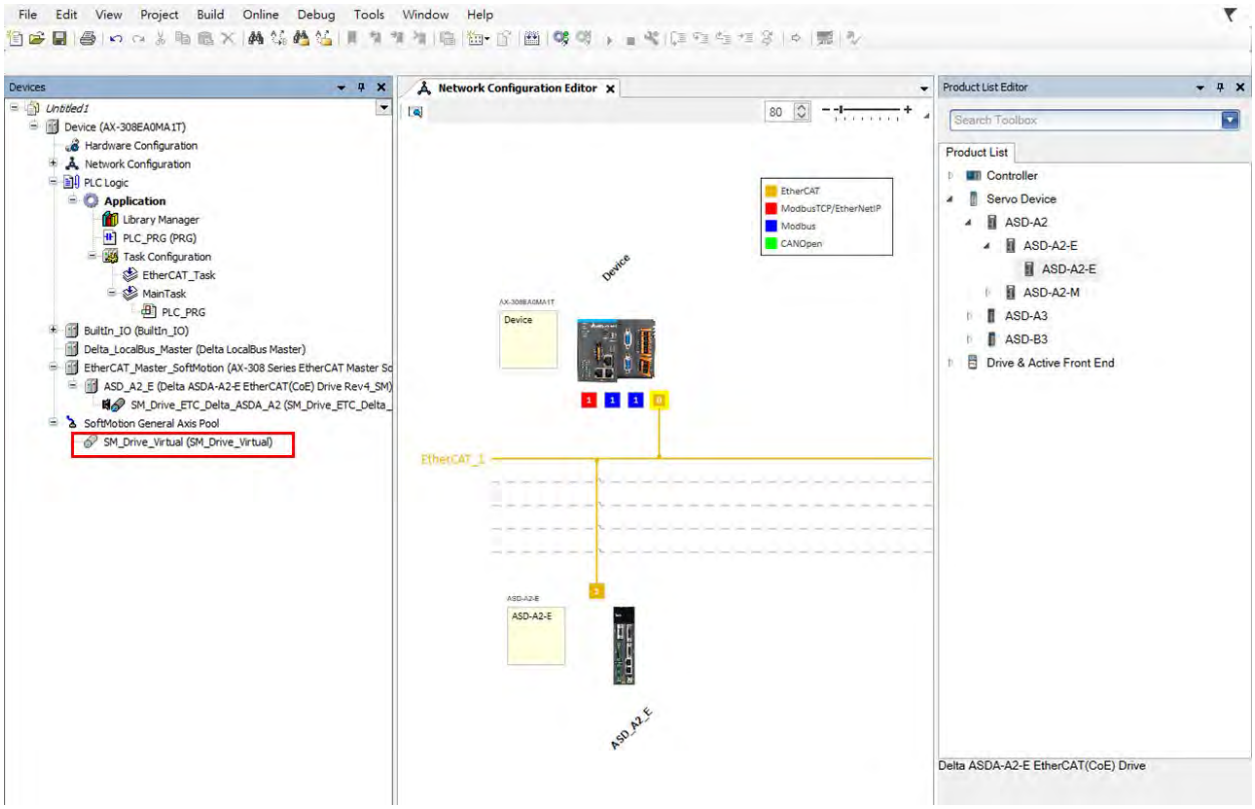
11. After created the real axis, continue to build the virtual axis by right-clicking “SoftMotion General Axis Pool” and selecting “Add Device”.



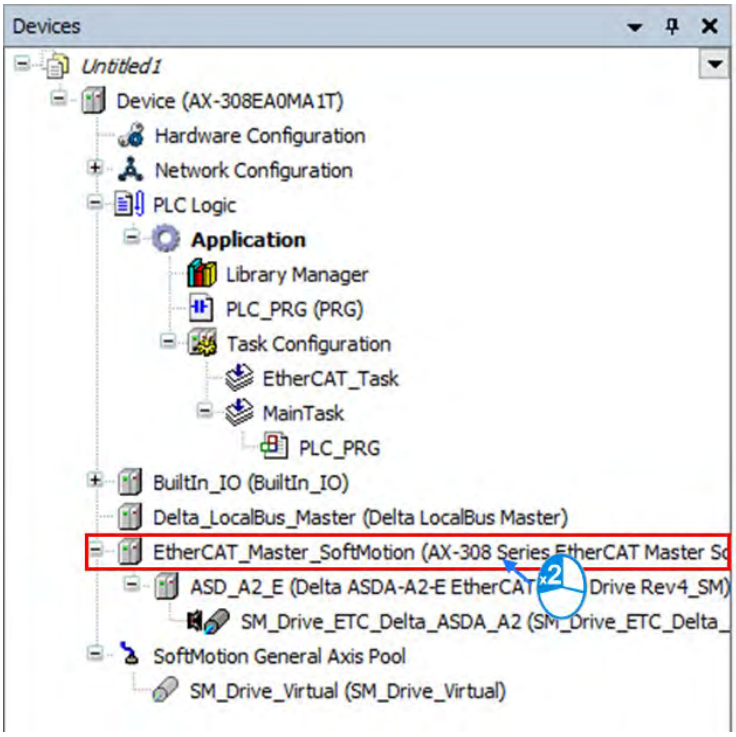
12. Expand “Virtual drives” object and choose “SM_Drive_Virtual” then select “Add Device” on the Add Device page.



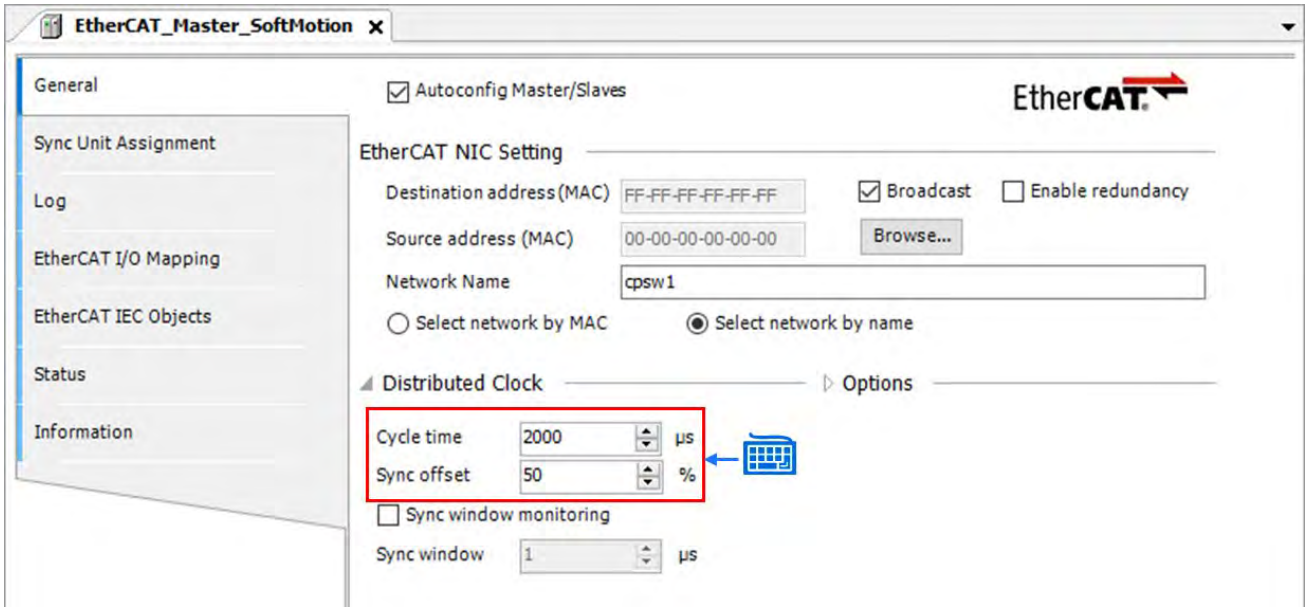
13. “SM_Drive_Virtual” will be shown in the project tree after the virtual axis is created.



14. Double-click on “EtherCAT_Master_SoftMotion”.

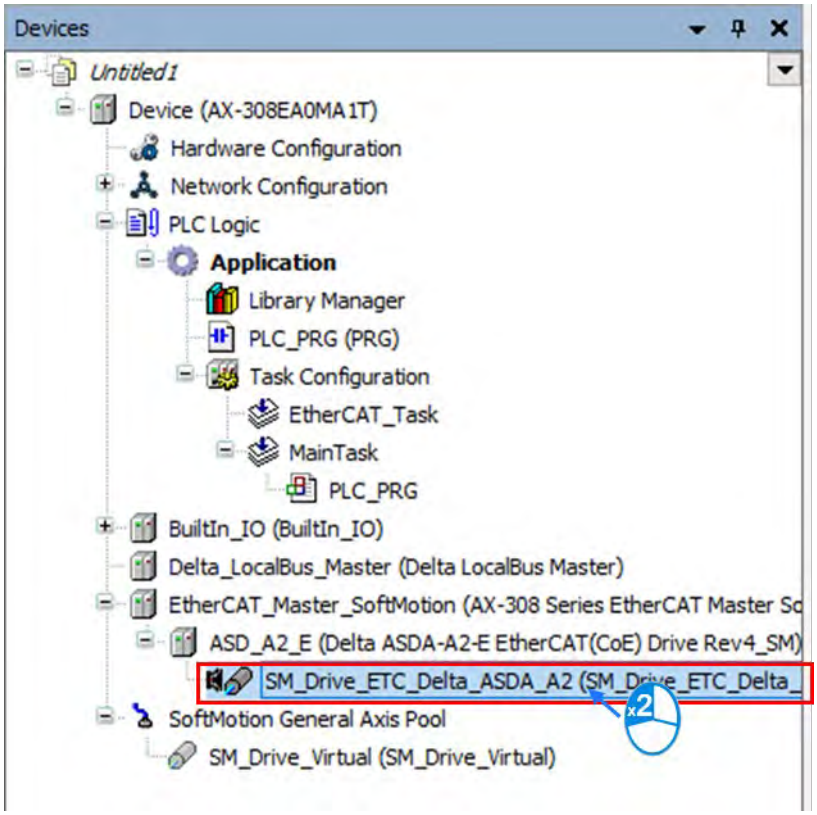


15. Set “Cycle time” to 2ms and “Sync offset” to 50 on the “EtherCAT_Master_SoftMotion” page.

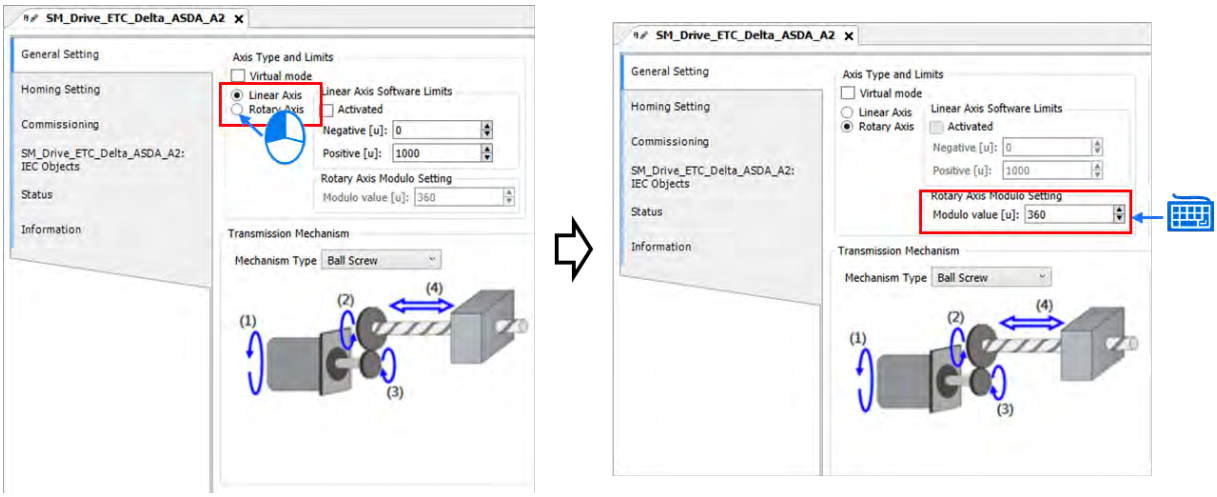


3.4.2.2 Axis Parameter Settings

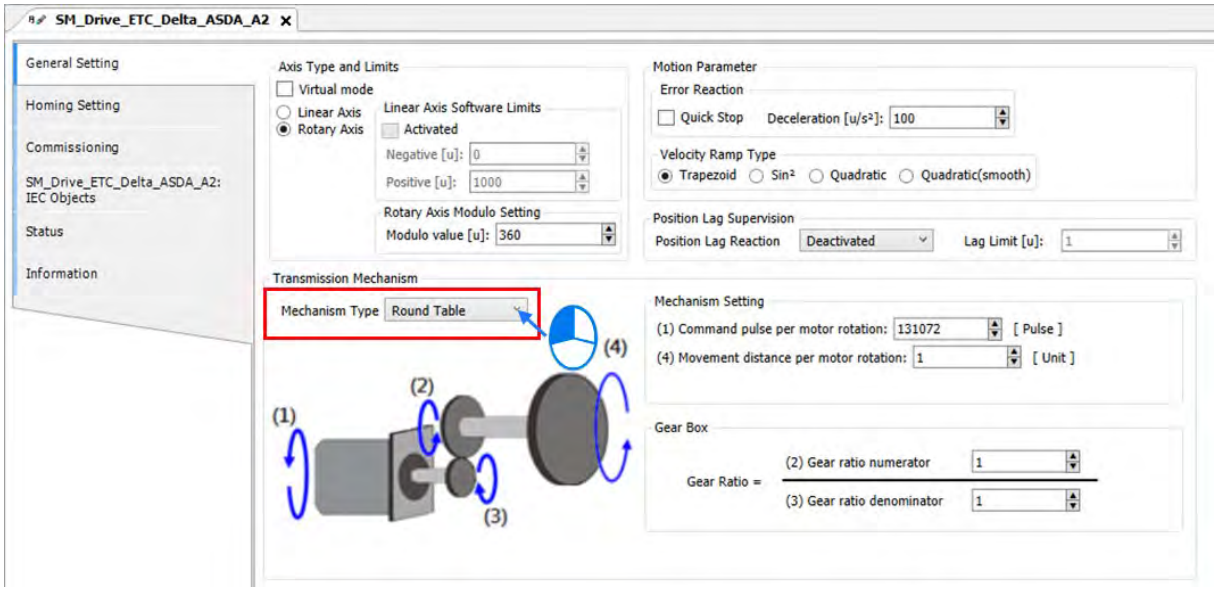
1. Select the first servo axis “SM_Drive_ETC_Delta_ASDA_A2” and double click on it to open the “SM_Drive_ETC_Delta_ASDA_A2” page.



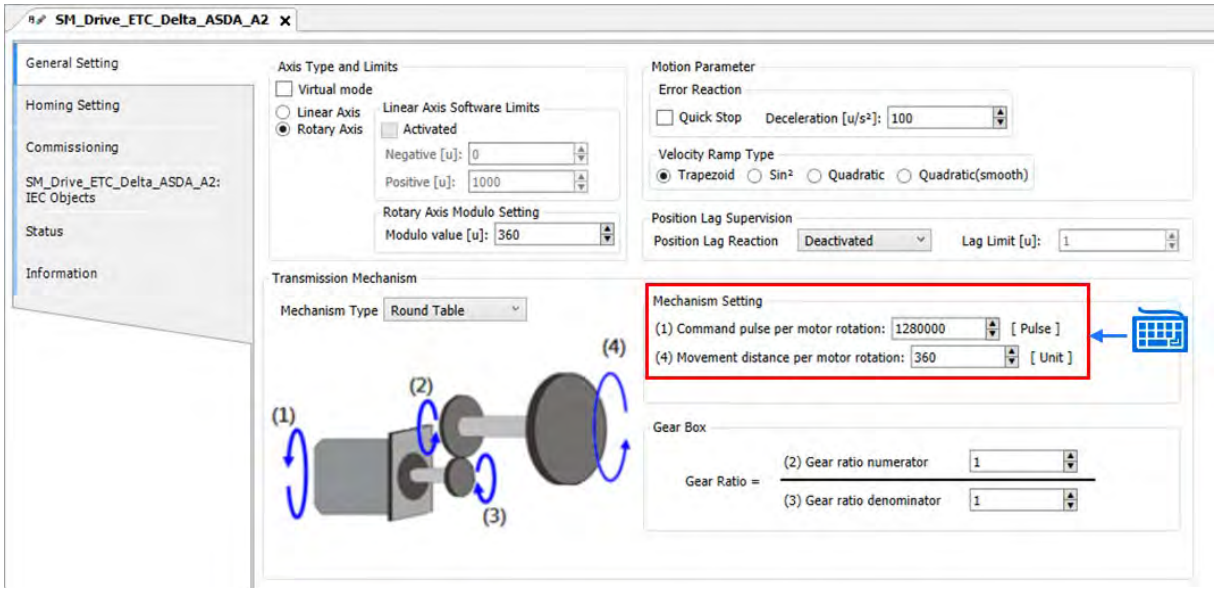
2. Set the axis type to “Rotary Axis” and you will be able to configure the “Rotary Axis Modulo Setting” with a maximum setting value of 360 degrees.



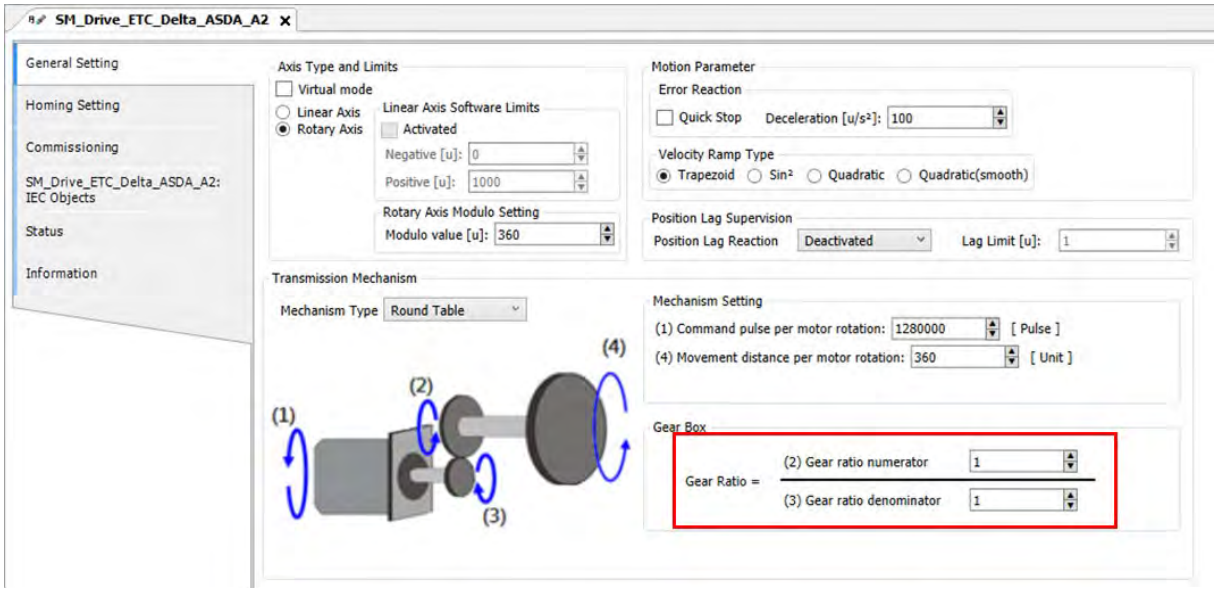
3. Choose “Round Table” to be the mechanism type as a result of rotary axis setting.



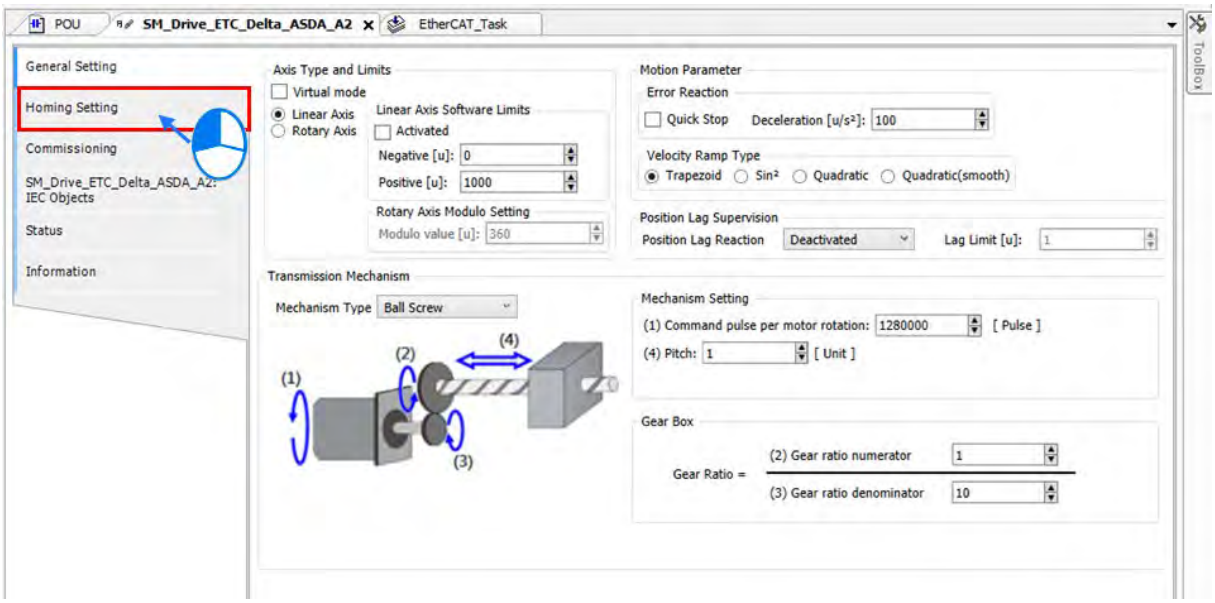
4. Then determine the proper gerar ratio by setting “Command pulse per motor rotation” to 1,280,000 and the user unit to 360 degrees. (Servo’s electronic gear ratio $\frac{P1-44}{P1-45} = \frac{\text{Motor Pulse per revolution}}{\text{Command Pulse per rotation}}$)



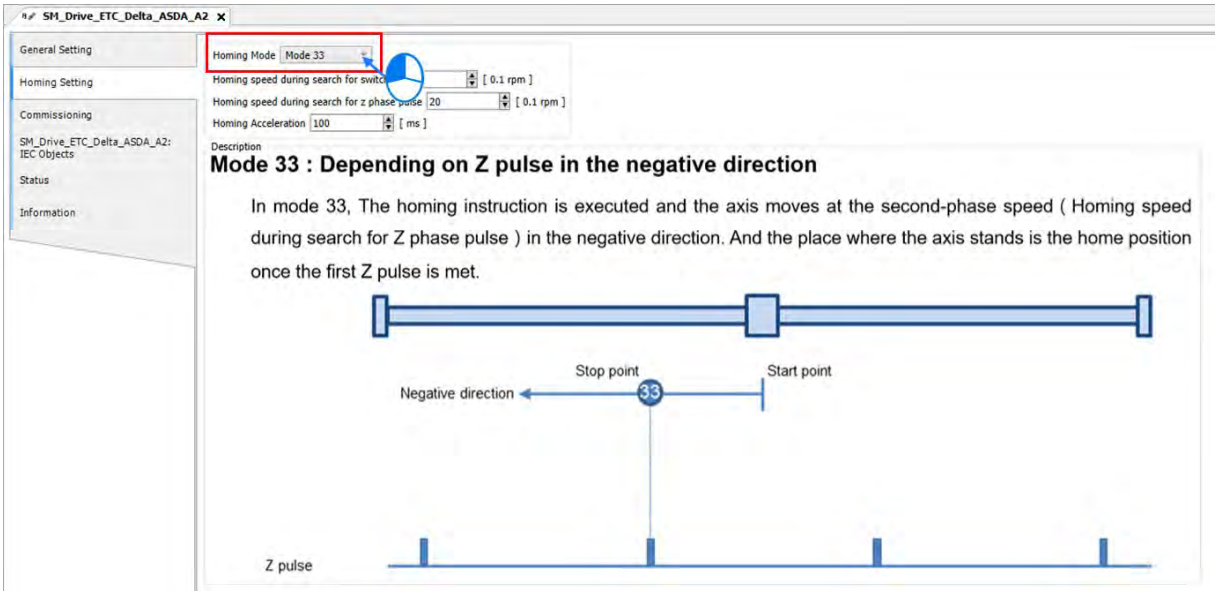
5. Since the mechanism is not equipped with a gearbox, the gear ratio for gearbox would be 1:1.



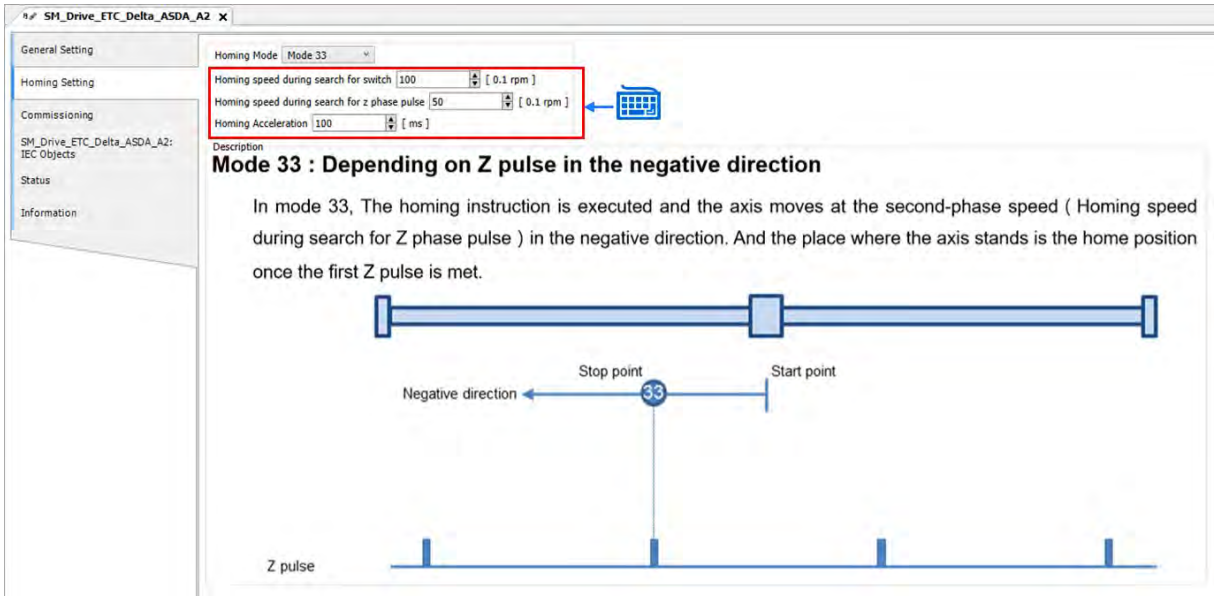
6. When finished the configuration of gear ratio, click “Homing Setting” to enter the page.



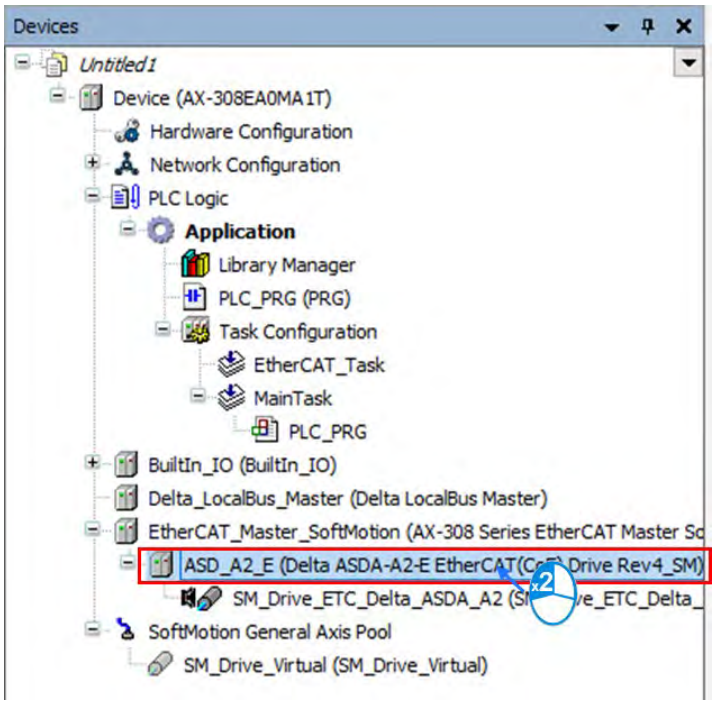
7. Choose Mode 33 for rotary axis type mechanism so as to perform homing depending on Z puls. The detailed descriptions of all homing modes are shown below.



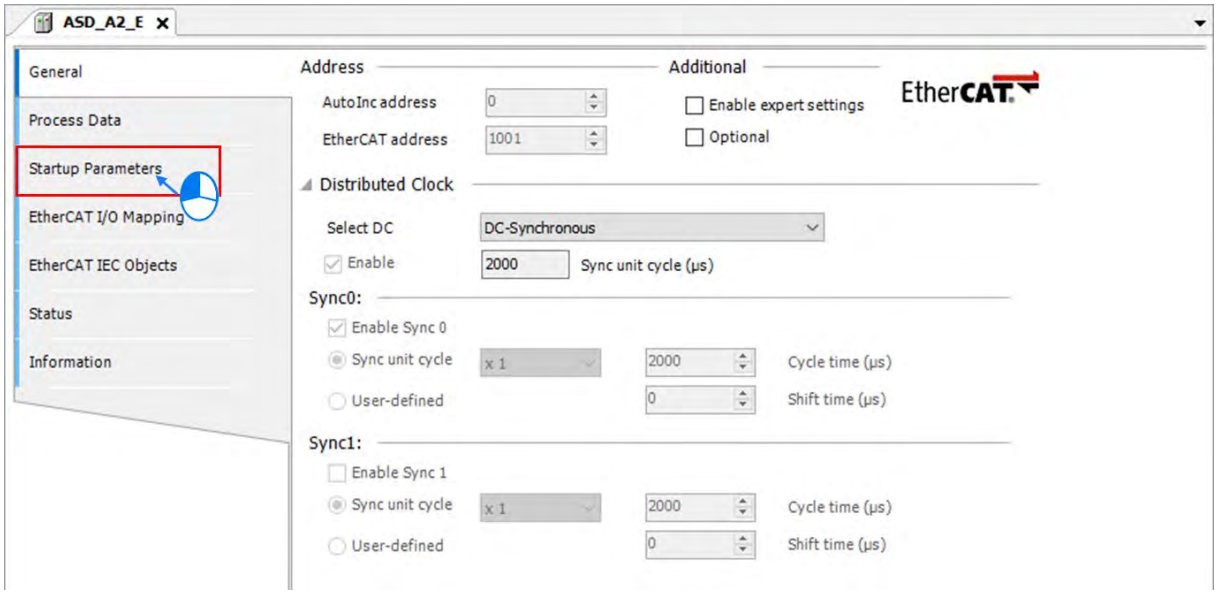
8. Then set the proper homing speed which should match with the mechanism.



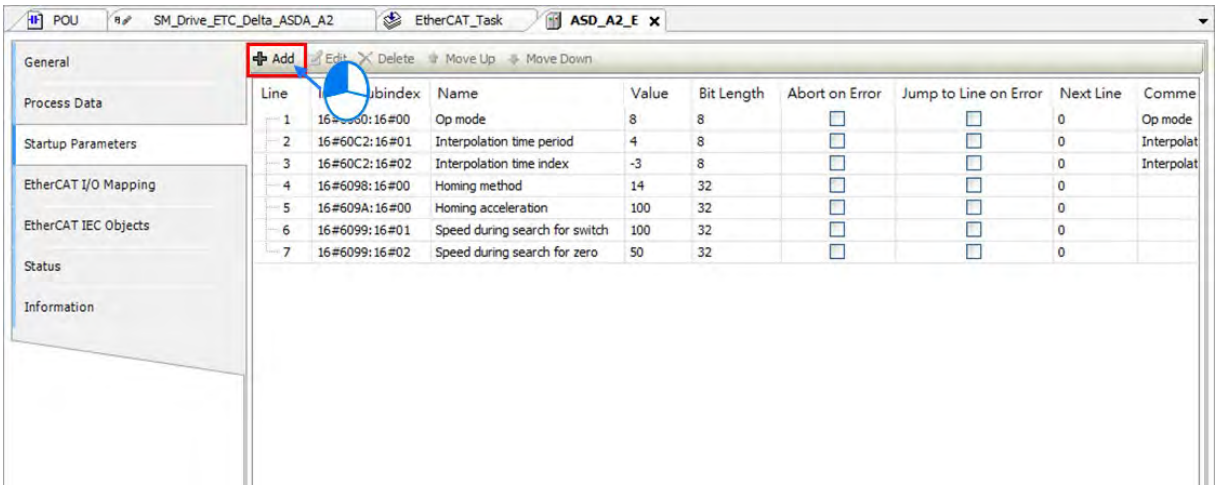
9. After finish the settings of gear ratio and homing mode, the parameters on the servo drive need to be tested for DI9 and DI10 inputs placed on the servo as a result of hardware limit. To configure the pin functions of DI9 and DI10, double click on “ASD_A2_E”.



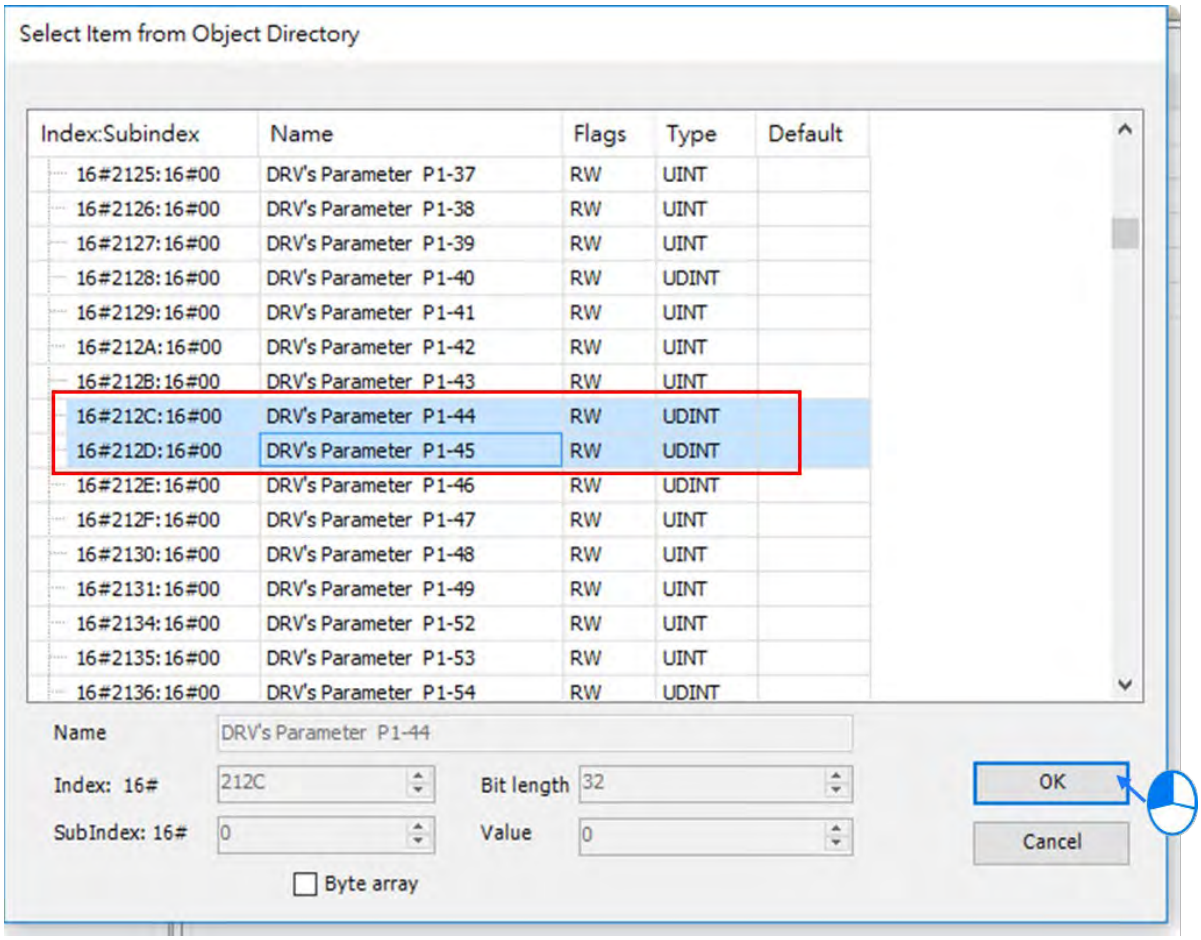
10. Click “Startup Parameters” tab on the ASD_A2_E page.



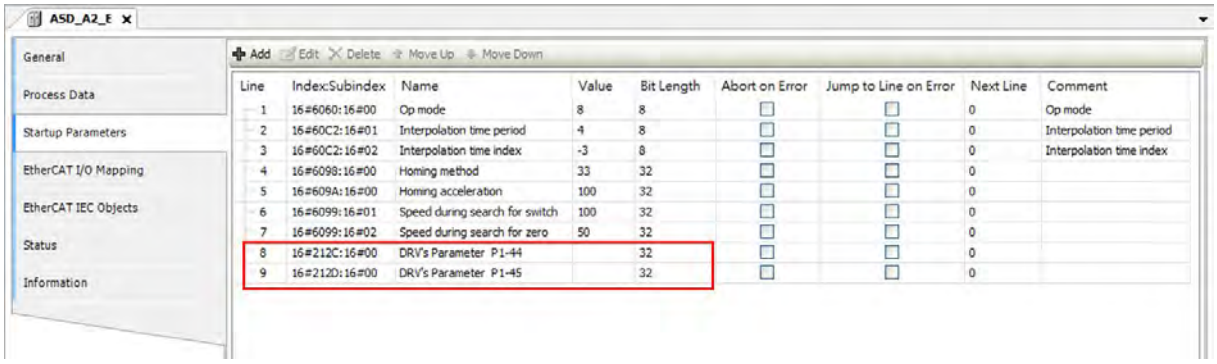
11. Click “Add” to add new parameters on “Startup parameters” page.



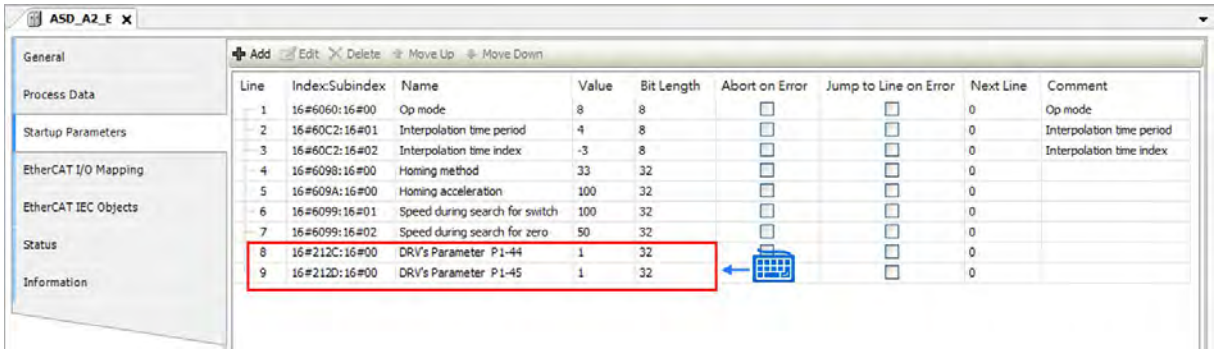
12. Select “P1-44” and hold the Shift button to select multiple parametres. Then click OK to add the selected parameters.



13. The selected parameters you’ve just added will be displayed on the list.

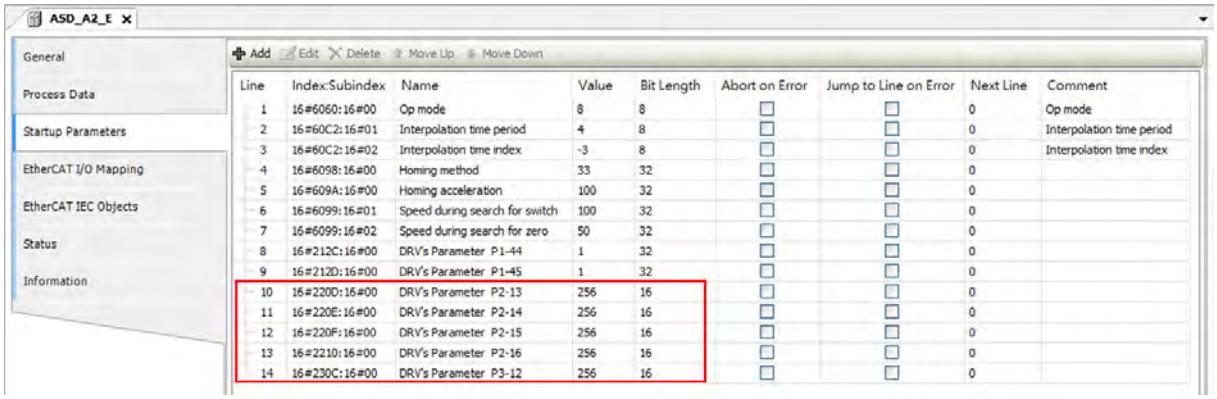


14. Set the values of servo's gear ratio P1-44 and P1-45 to 1, which should match the gear ratio in axis parameters. (All parametres on this page will be downloaded to the servo after EtherCAT communication is established.)



Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on 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 period
3	16#60C2:16#02	Interpolation time index	-3	8	☐	☐	0	Interpolation time index
4	16#6098:16#00	Homing method	33	32	☐	☐	0	
5	16#609A:16#00	Homing acceleration	100	32	☐	☐	0	
6	16#6099:16#01	Speed during search for switch	100	32	☐	☐	0	
7	16#6099:16#02	Speed during search for zero	50	32	☐	☐	0	
8	16#212C:16#00	DRV's Parameter P1-44	1	32	☐	☐	0	
9	16#212D:16#00	DRV's Parameter P1-45	1	32	☐	☐	0	

15. Use the same operation to configure the following parameters, which the DI4~DI7 should be turned OFF and the retentive parameters should be configured.



Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on 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 period
3	16#60C2:16#02	Interpolation time index	-3	8	☐	☐	0	Interpolation time index
4	16#6098:16#00	Homing method	33	32	☐	☐	0	
5	16#609A:16#00	Homing acceleration	100	32	☐	☐	0	
6	16#6099:16#01	Speed during search for switch	100	32	☐	☐	0	
7	16#6099:16#02	Speed during search for zero	50	32	☐	☐	0	
8	16#212C:16#00	DRV's Parameter P1-44	1	32	☐	☐	0	
9	16#212D:16#00	DRV's Parameter P1-45	1	32	☐	☐	0	
10	16#220D:16#00	DRV's Parameter P2-13	256	16	☐	☐	0	
11	16#220E:16#00	DRV's Parameter P2-14	256	16	☐	☐	0	
12	16#220F:16#00	DRV's Parameter P2-15	256	16	☐	☐	0	
13	16#2210:16#00	DRV's Parameter P2-16	256	16	☐	☐	0	
14	16#230C:16#00	DRV's Parameter P3-12	256	16	☐	☐	0	

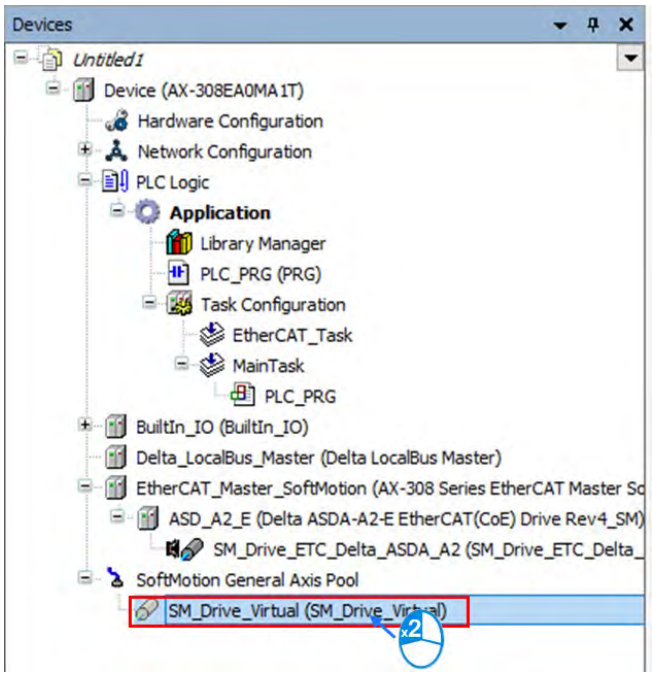
■ Servo parameters

The following parameters and values are used for this example on the Startup Parameter page.

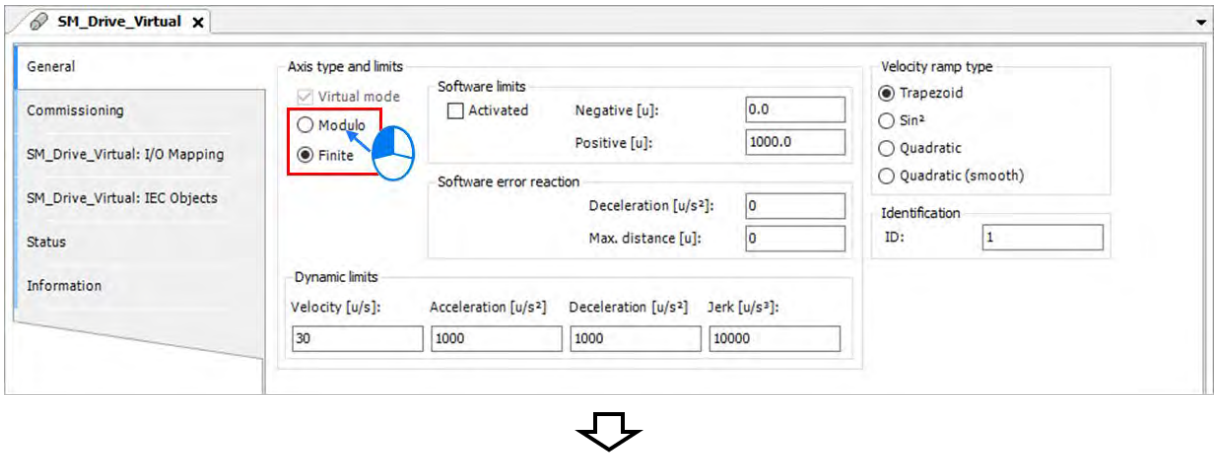
Parameter	Function	Setting Value
P1-44	Gear ratio numerator	1
P1-45	Gear ratio denominator	1
P2-13	DI4 digital input pin function	256
P2-14	DI5 digital input pin function	256
P2-15	DI6 digital input pin function	256
P2-16	DI7 digital input pin function	256
P3-12	Remain unchanged with the parameters before the drive power being cut off.	256

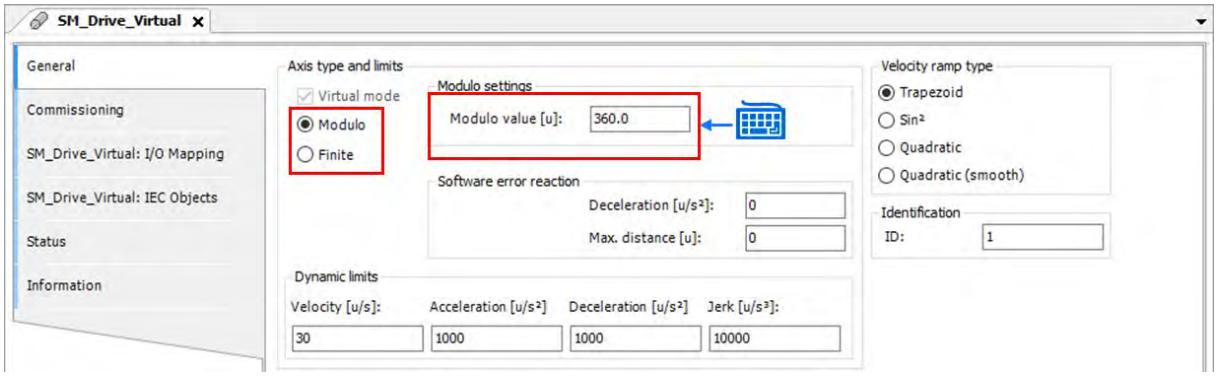
***Note:** Please refer to the user manuals of Delta ASDA-A2 Series for more detailed information of parameters.

16. After the configuration of real axes is completed, continue to set the virtual axes by double-clicking on “SM_Drive_Virtual”.



17. Change the axis type to “Modulo” and the maximum value of modulo setting is 360.



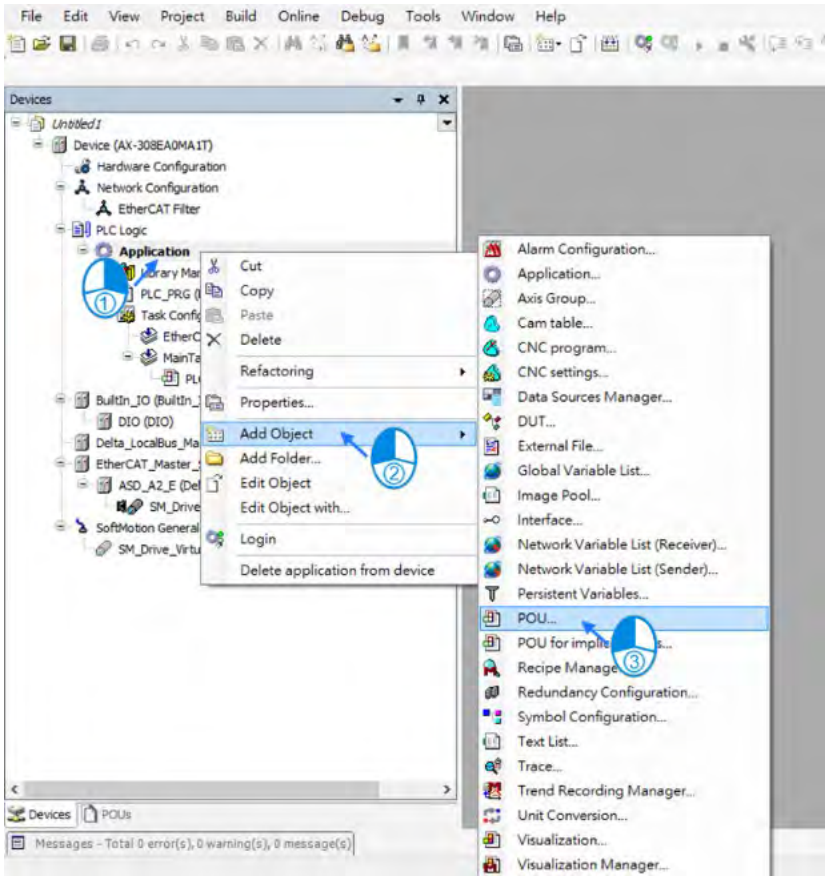


3.4.2.3 Programming for New Projects

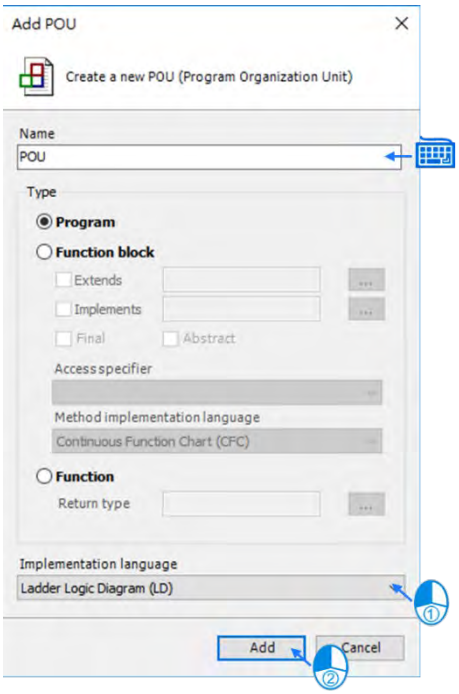
- Operating process in the example
  - "MC_Power" will be triggered and enable two servo axes (Servo ON) after executing "Step 1".
  - "MC_Home" will be triggered after executing "Step 2" to command the real axis back to the home position. While "MC_SetPosition" is required for returning the virtual axis to zero position.
  - Execute "Step 3" to trigger "MC_Jog" and the virtual axis starts to move. At the same time, "MC_GernIn" would be executed and the slave axis starts to follow. Once "Step 3" is aborted, "MC_Jog" will be stopped and "MC_GearOut" will be triggered. After the Done output of MC_GearOut becomes TRUE, "MC_Stop" is triggered to stop the slave axis. The whole process will be performed repeatly till the contact singnal of "Step3" is OFF.
  - The value of Counter_1 will be increased by 1 for each time the operation being executed automatically.

- Programming with newly-created POUs

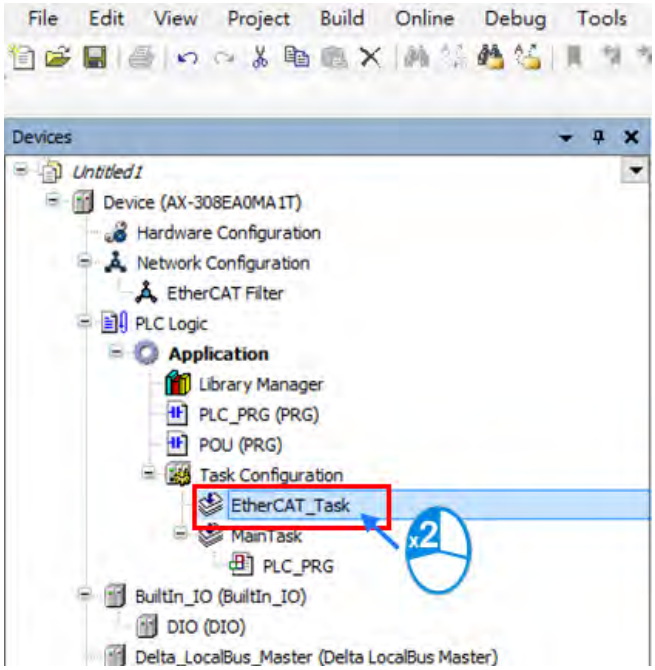
1. First, create a new POU by right-clicking "Application" to choose "Add Object" and select "POU".



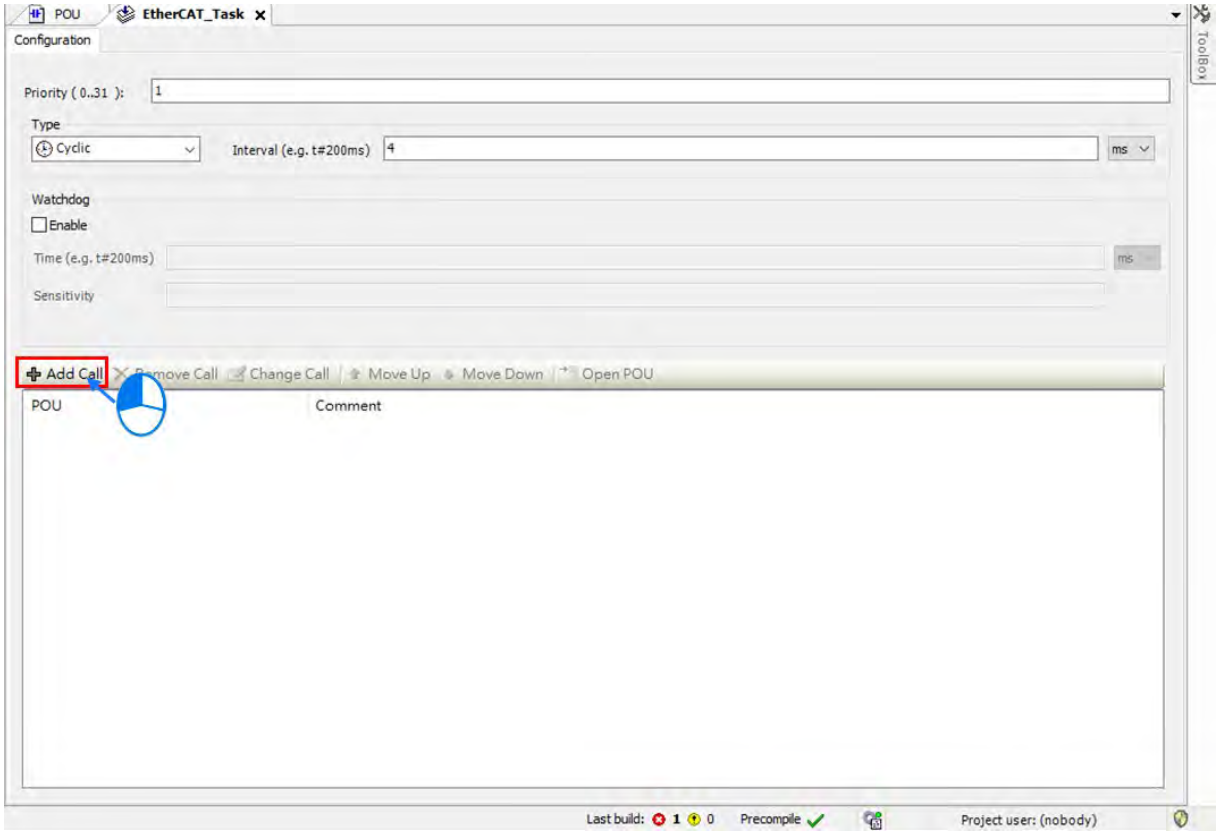
2. Input the name of the new POU in the “Name” field and choose LD language for “Implementation language”. When finished, click “Add” to add the new POU.



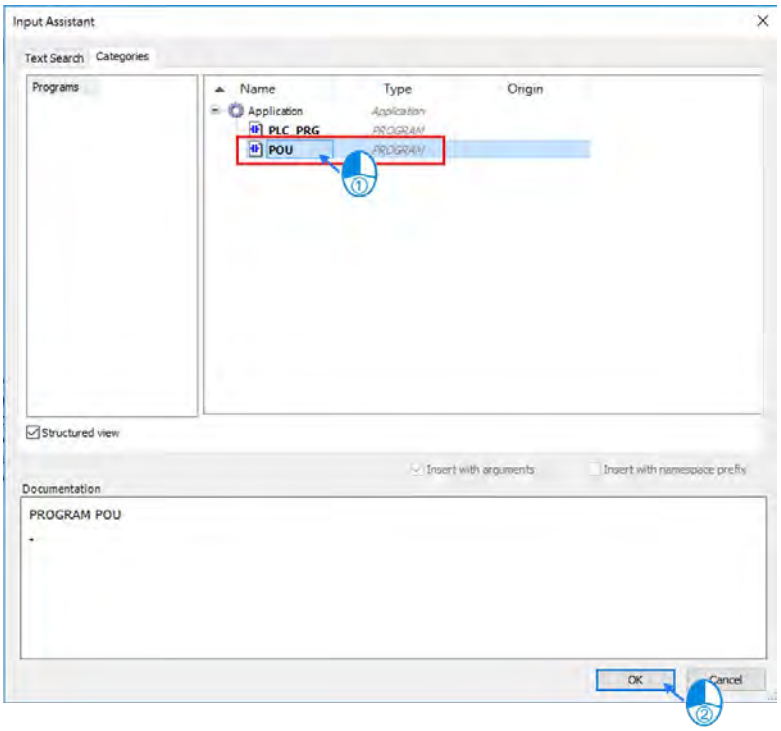
3. The newly added POU would be shown in the project tree, which needs to be added in Task by double-clicking “EtherCAT_Task”. (Function blocks related to axis motion needs to be established in EtherCAT_Task to ensure normal operation of axes.)



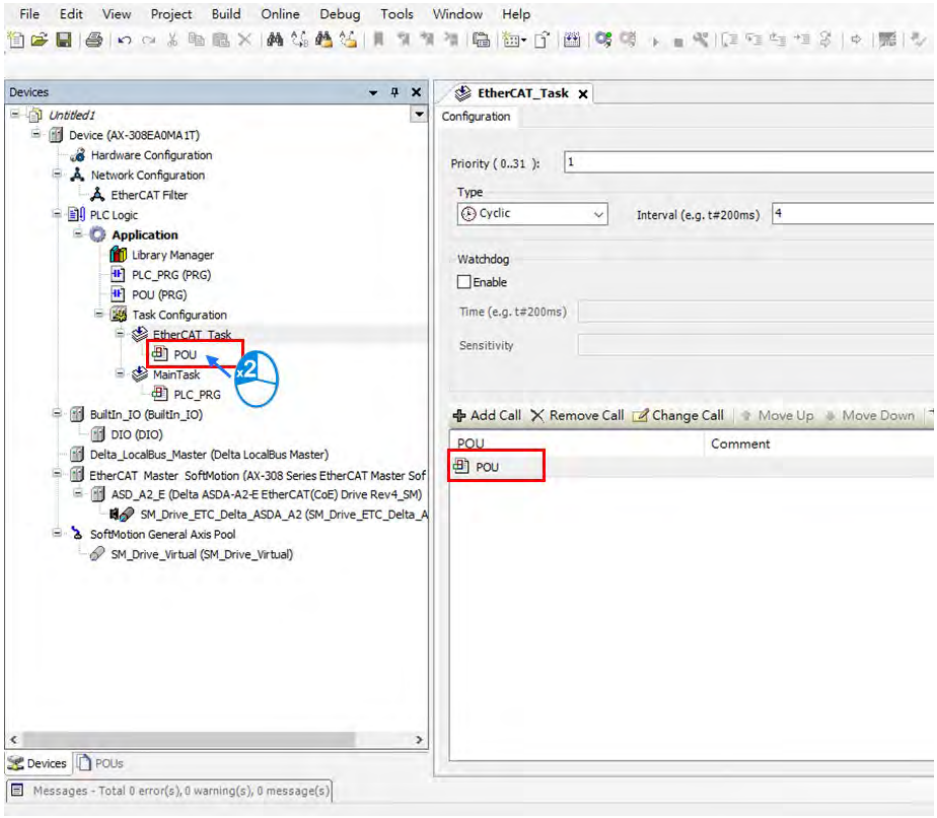
4. Click “Add Call” on the EtherCAT_Task page.



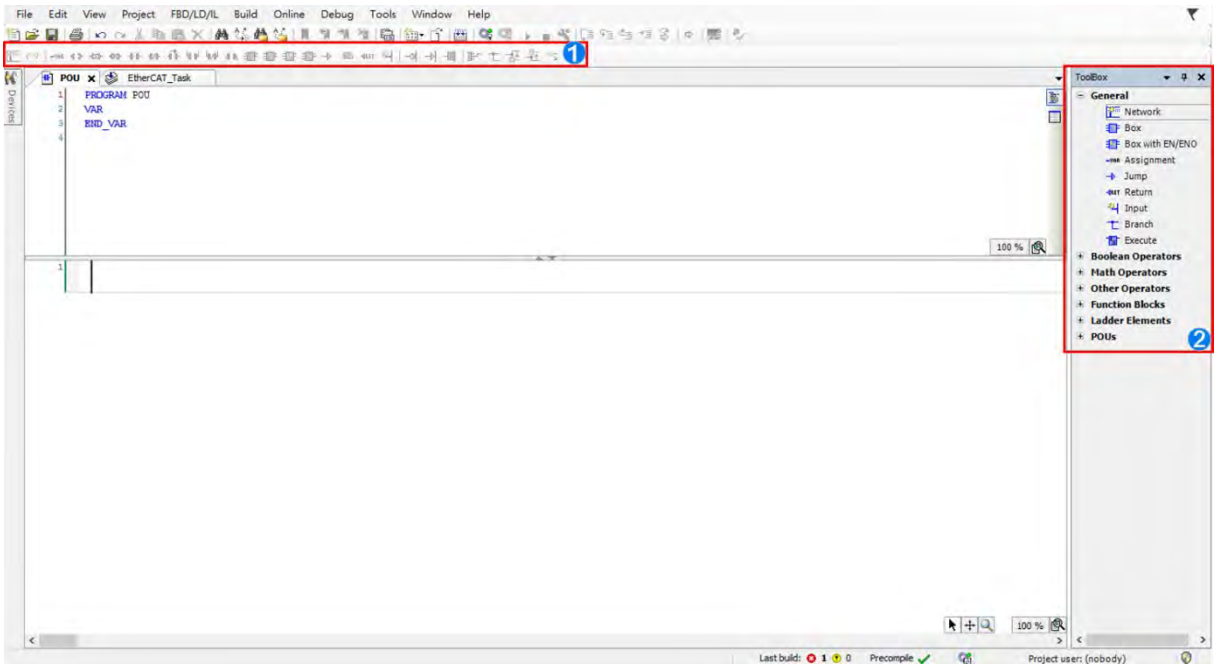
5. Click “OK” after choosing the newly-created POU.



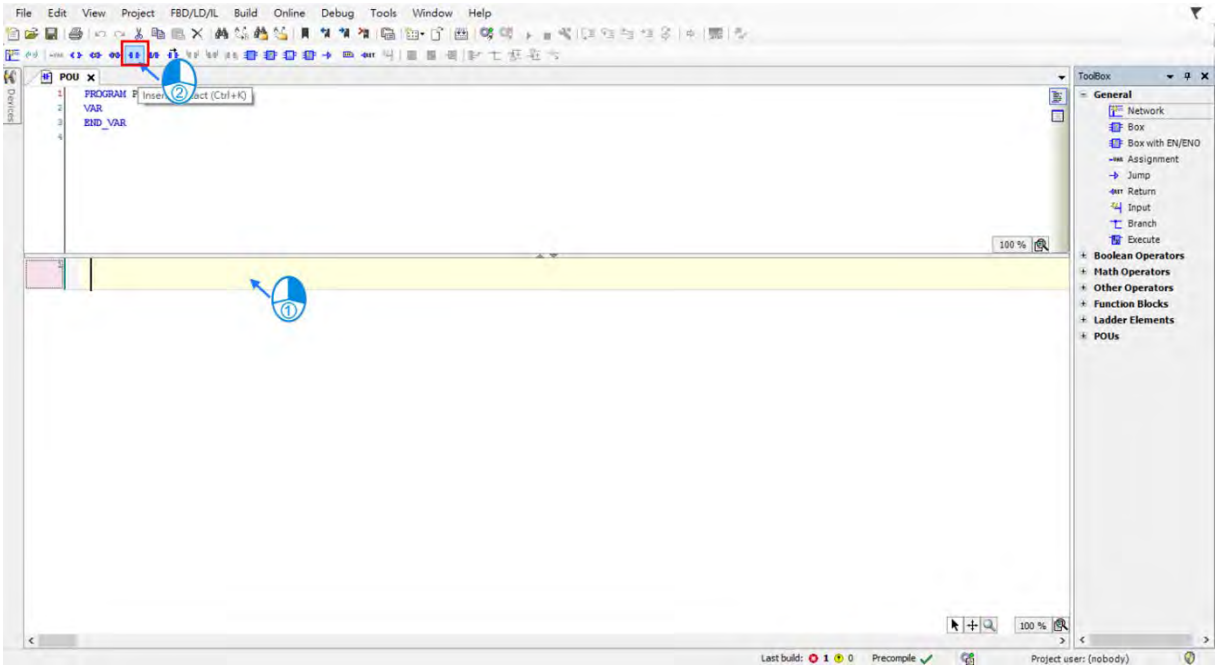
6. After adding the POU in EtherCAT_Task, “POU” will be shown on the lower half of the EtherCAT_Task page as well as in the project tree. Then click the POU from the tree, which is under the category of EtherCAT_Task.



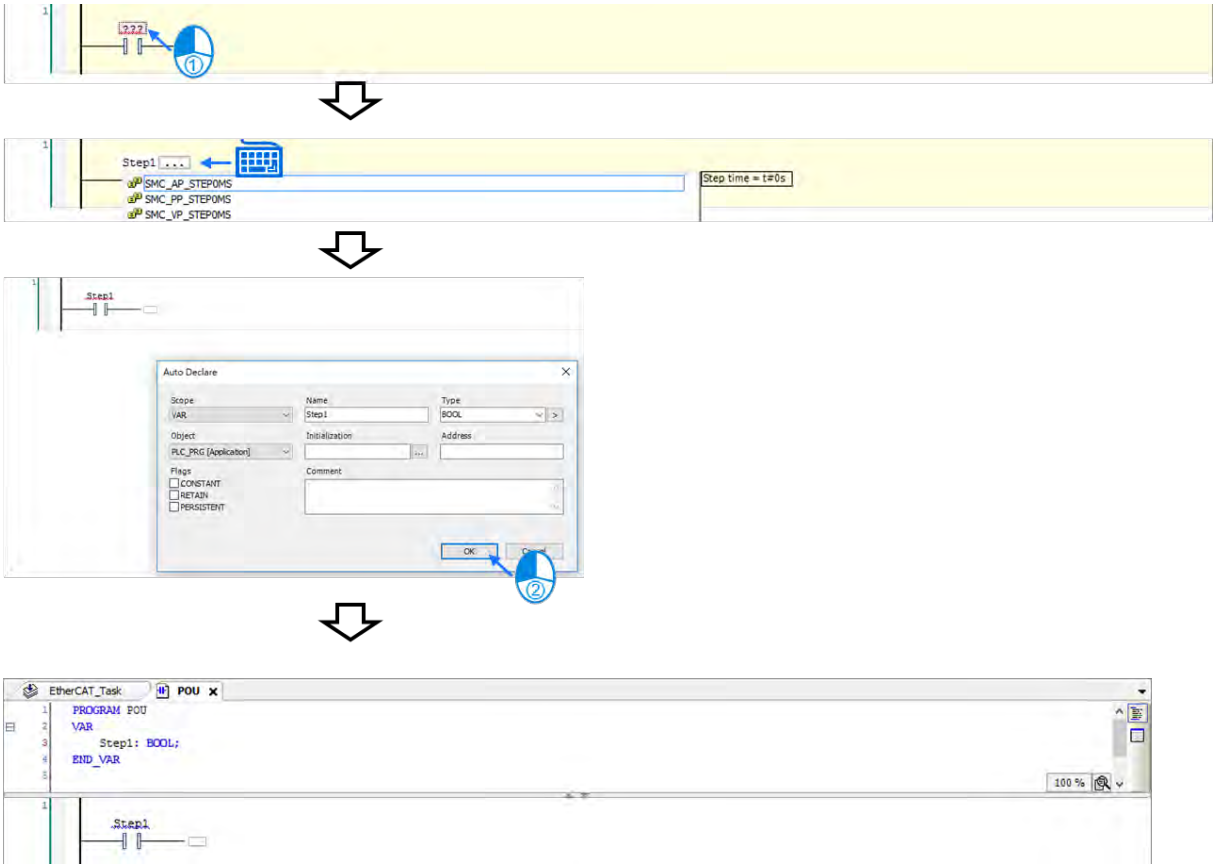
7. With ladder logic programming language, add new commands by using the red-circled field marked with 1, while field 2 is for adding required Function or Function Block.



8. Click on the desired place to add new commands on POU page, then select  .

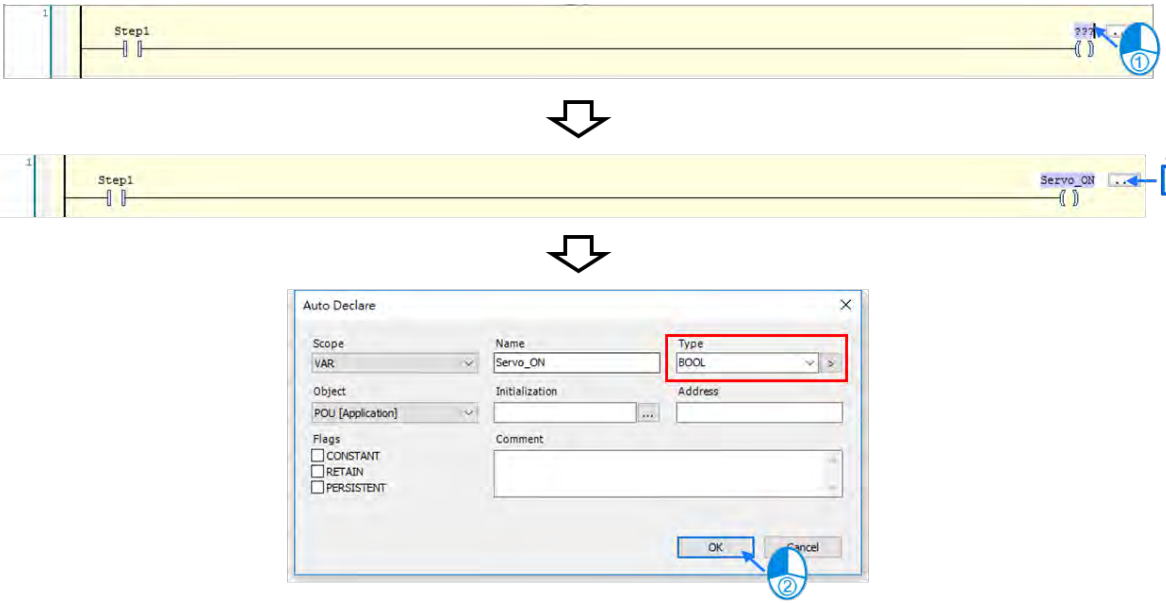


9. After the selected command is shown in the POU, click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.



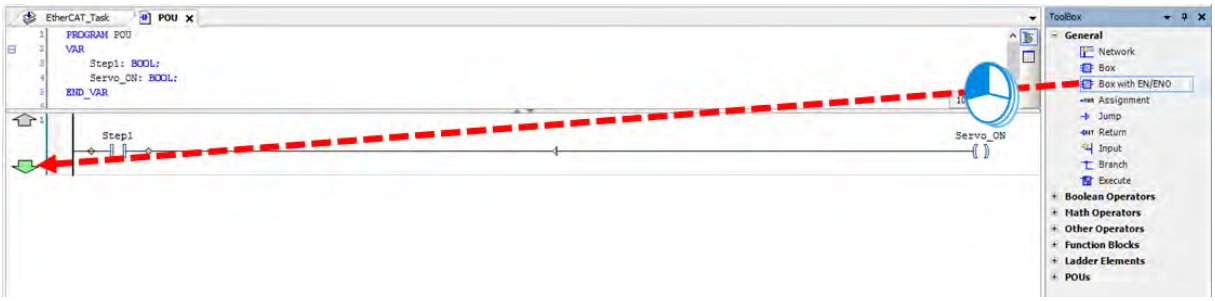
10. Then add the ouput instruction by clicking on the desired place and choosing  .

11. Click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.

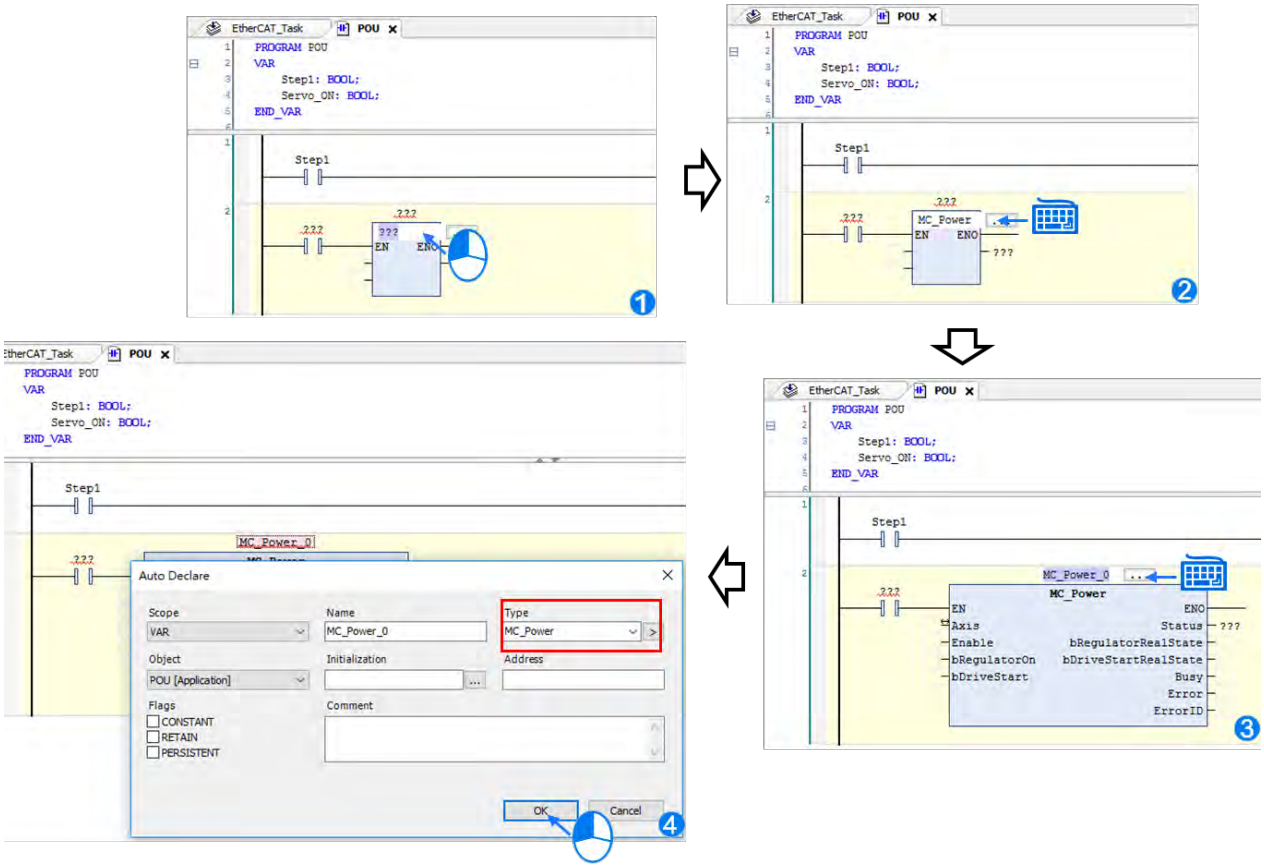


12. The instruction names you have added will be displayed as well as the instruction types, where the name and type of the instruction can also be added directly. (Same operation for the examples in later discussion.)

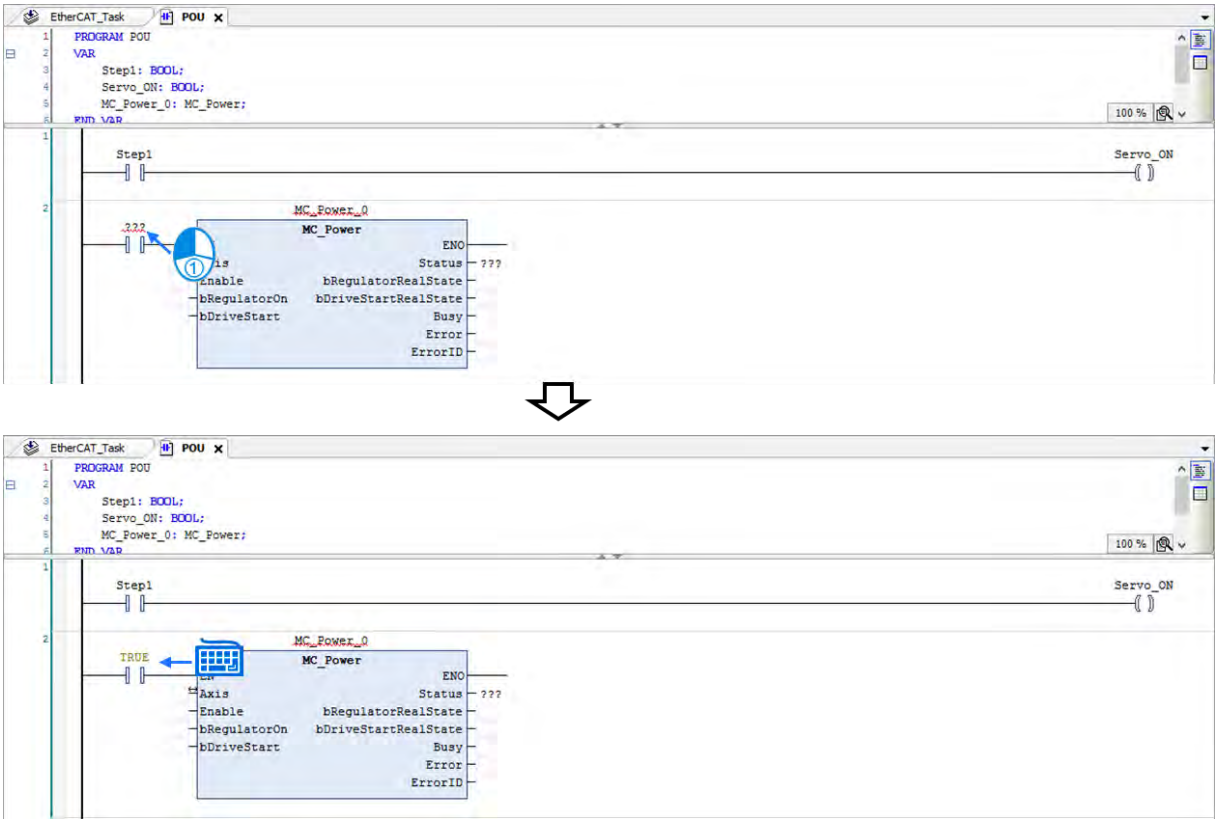
13. To add new function blocks, click and hold “Box With EN / ENO” then drag to  .



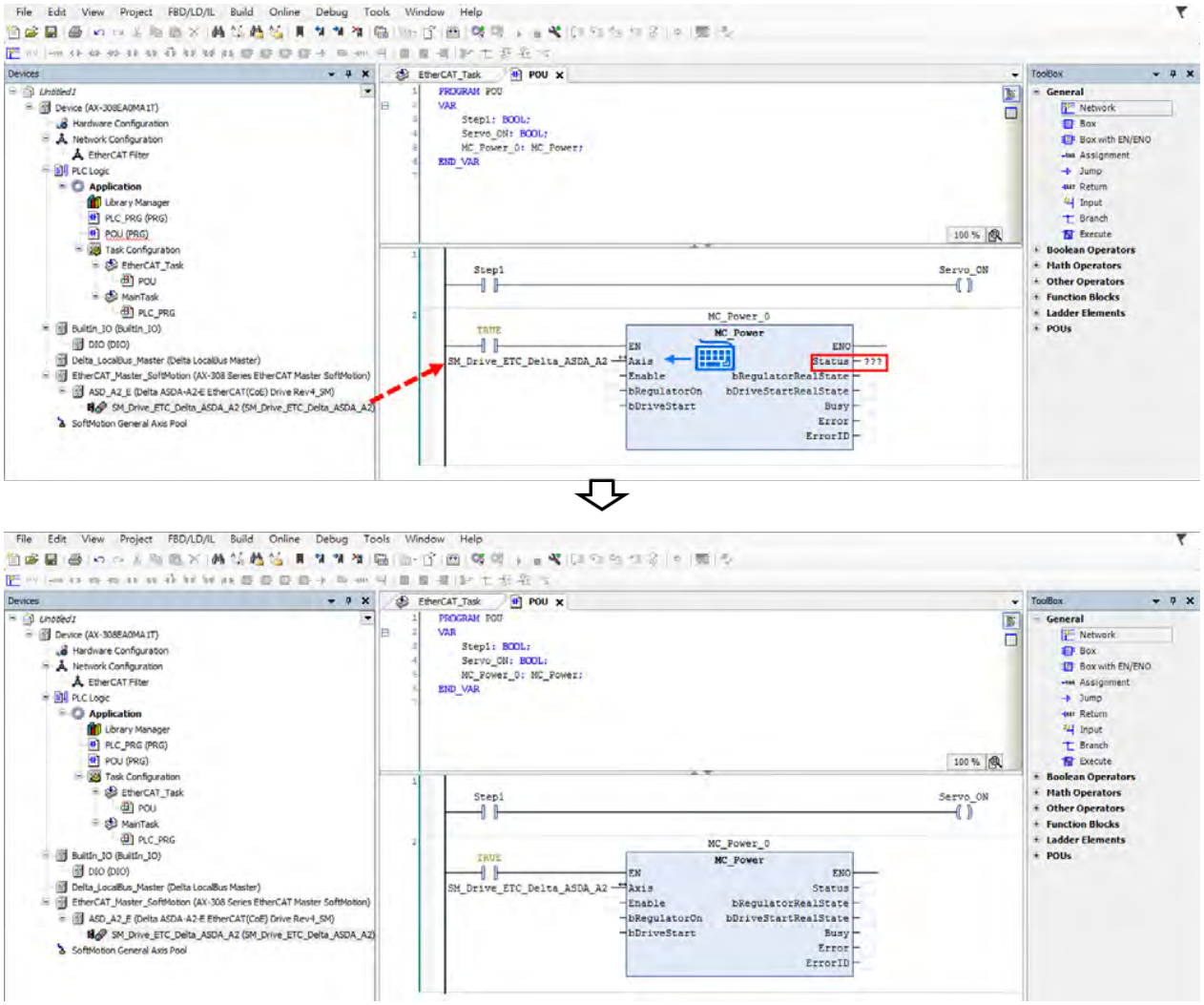
14. Click “???” and insert the name of function block then press ENTER to display the function block. Insert the name and press ENTER key again, then “Auto Declare” window will pop up. Click “OK” after confirming the “Type” of function block is correct.



15. Continue to click “???” and insert “TRUE”, then press the ENTER key.

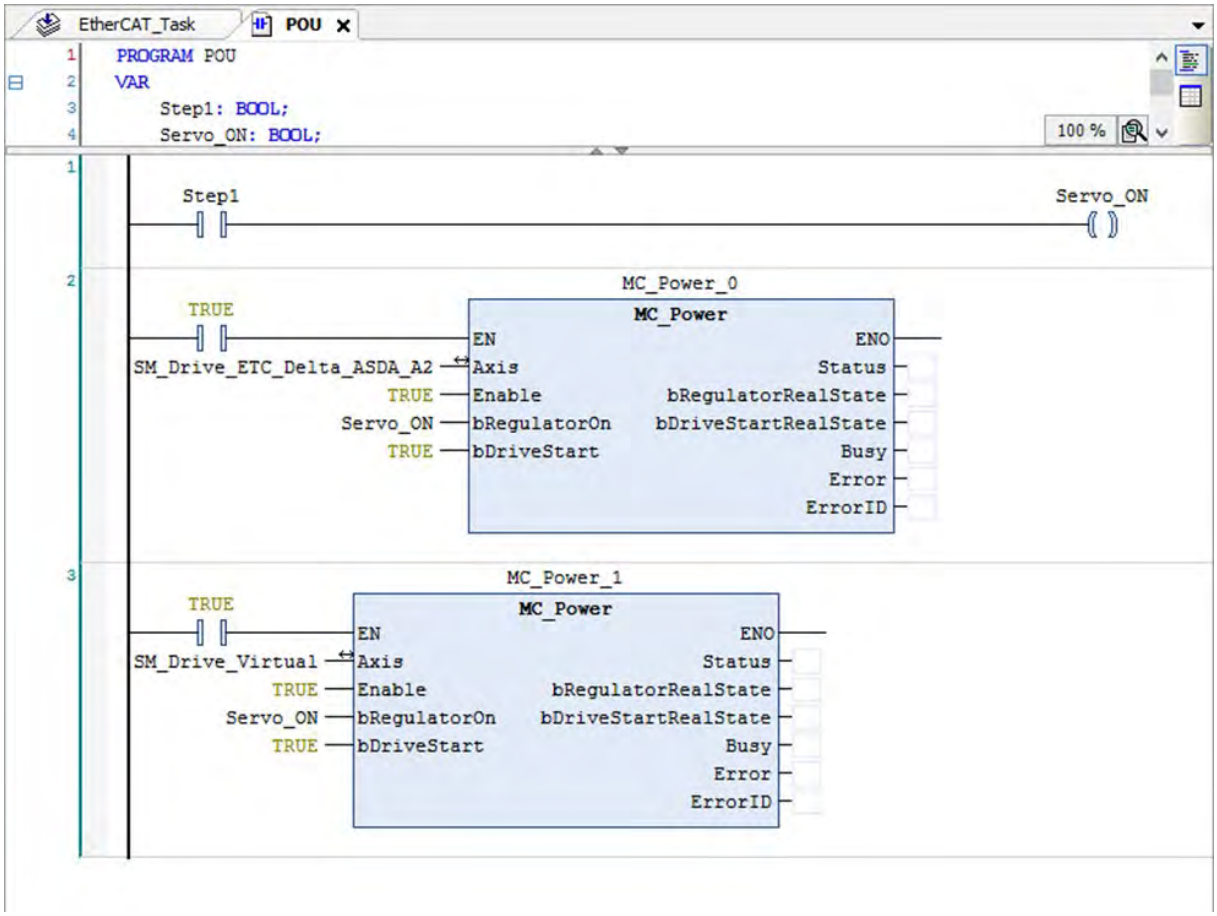


16. Insert the name of servo drive to the Axis input pin of the function block and delete “???” of other pins, since the mark “???” does not represent any variables and errors may occur while parsing.

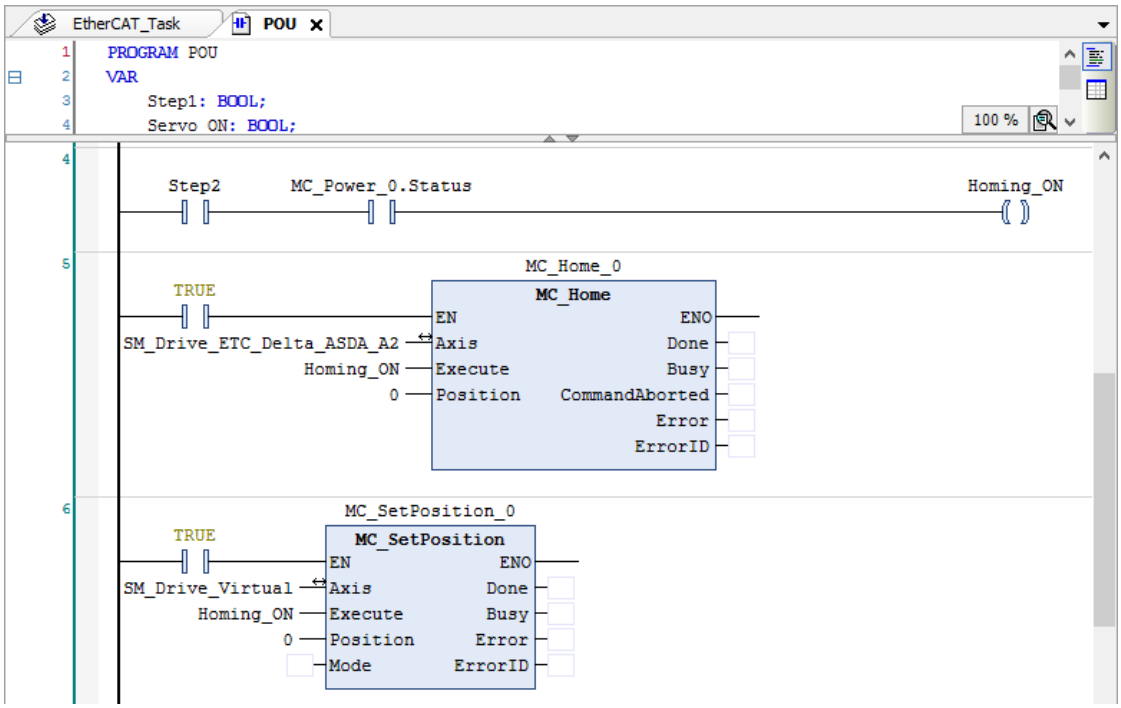


17. Complete the following programming example with the same operation process.

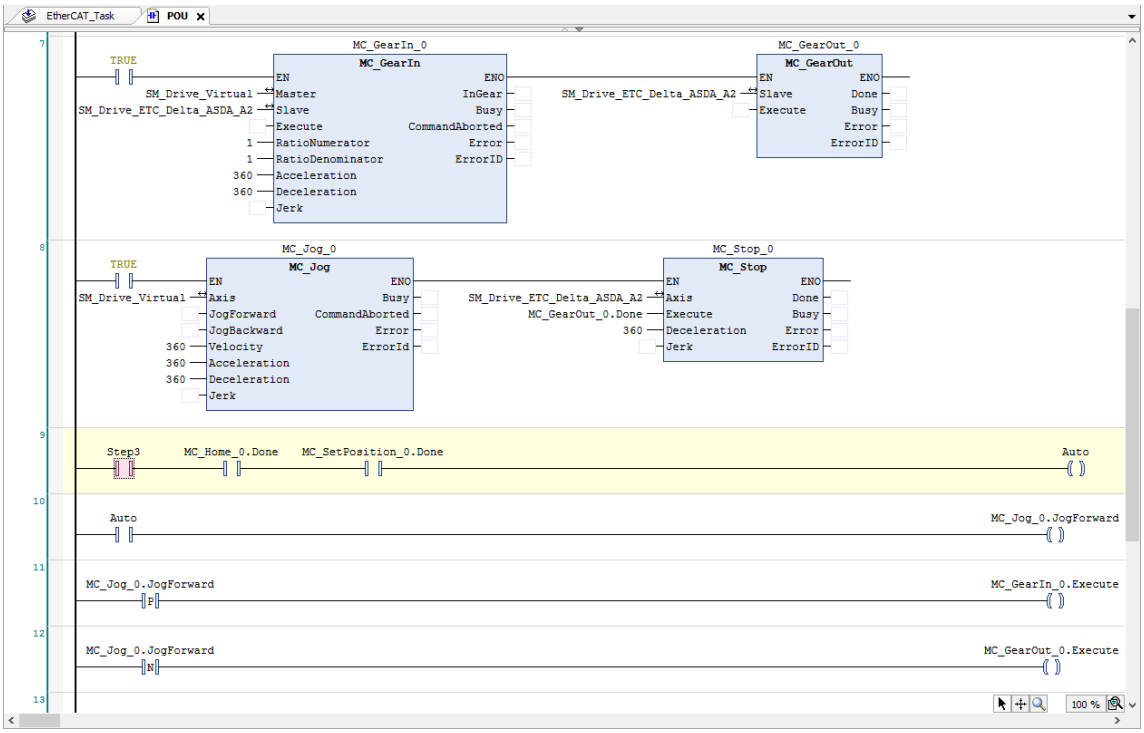
- Create function blocks of MC_Power for both axes.



- The function block “MC_Home” cannot be used with a virtual axis. Instead, MC_SetPosition can command the virtual axis to perform homing function.



- Then create “MC_GearIn” to activate a master-slave coupling with MC_Jog to trigger the virtual axis to move. “MC_GearOut” will also be executed by stopping the virtual master axis. Once the slave is decoupled, it will retain its velocity and move at a constant speed, which would need MC_Stop to stop the movement.

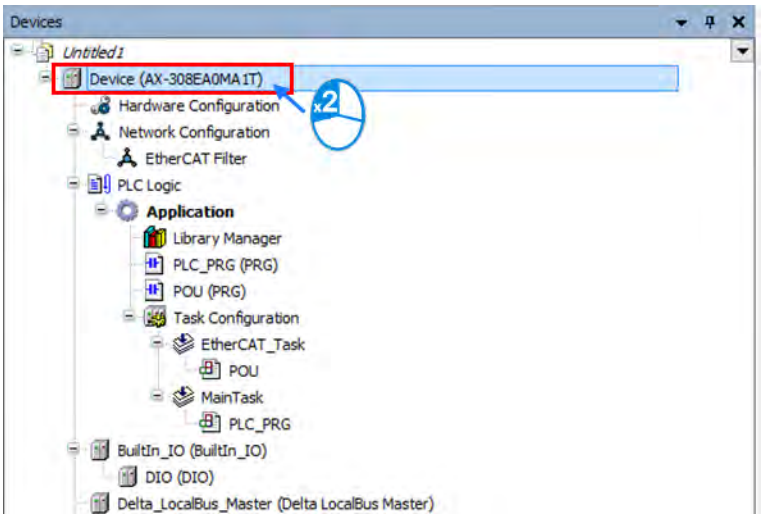


3.4.3 Program Monitoring

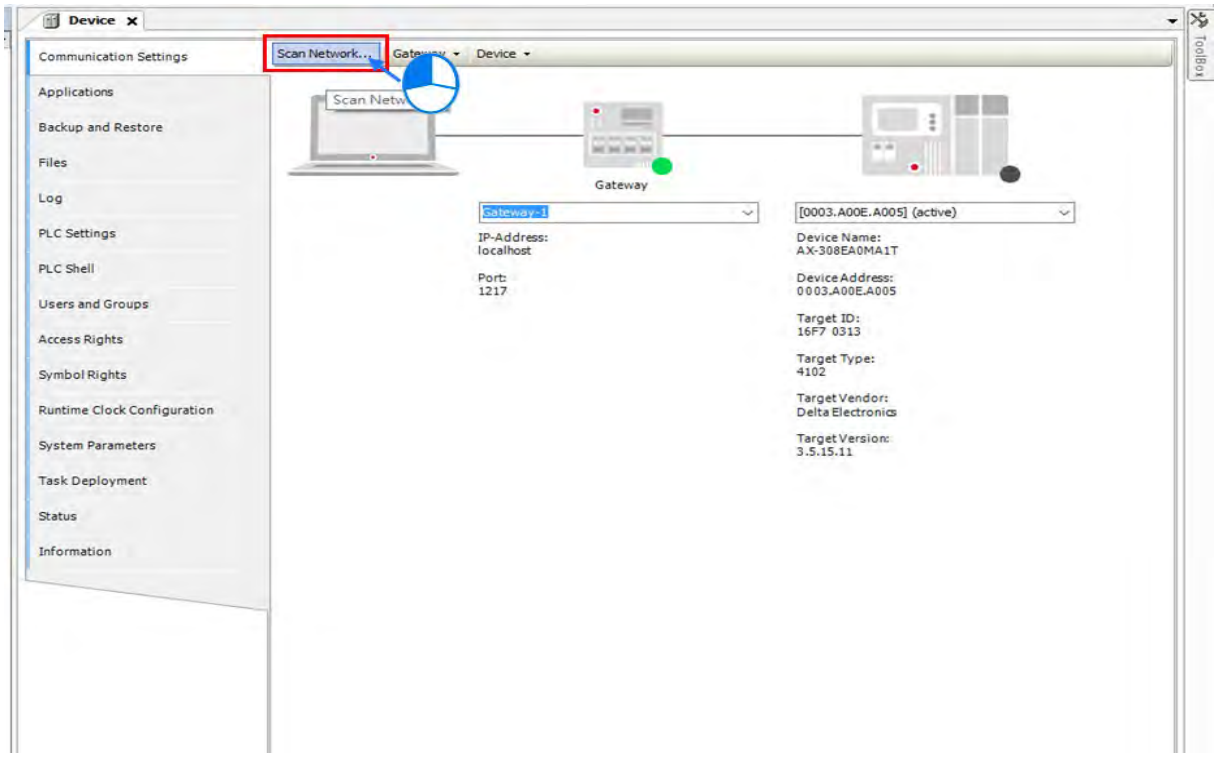
When running a program, the current control status of system can be monitored and part of the device values are allowed to be modified for system testing as well.

3.4.3.1 Setup Connection between Devices

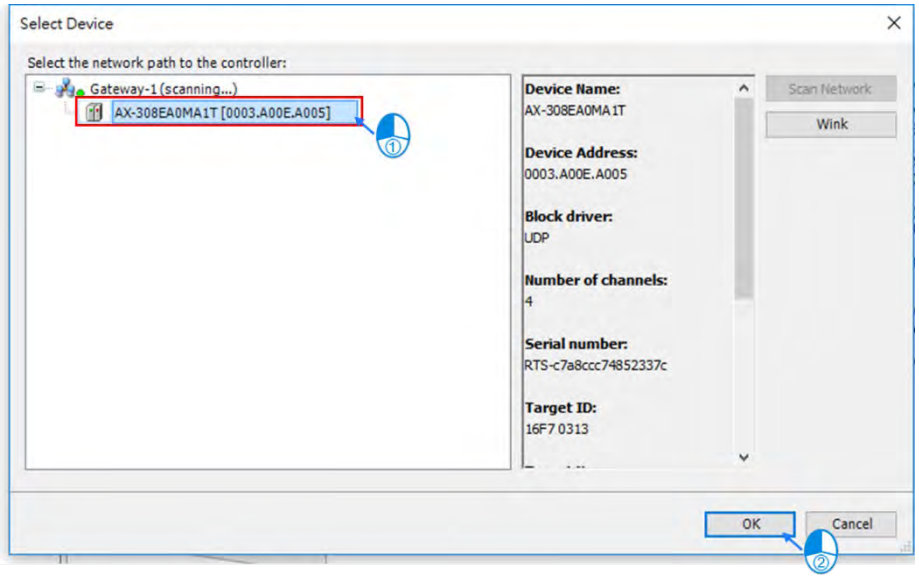
1. Double click on “Device”. (The default IP address for AX-308E is 192.168.1.5)



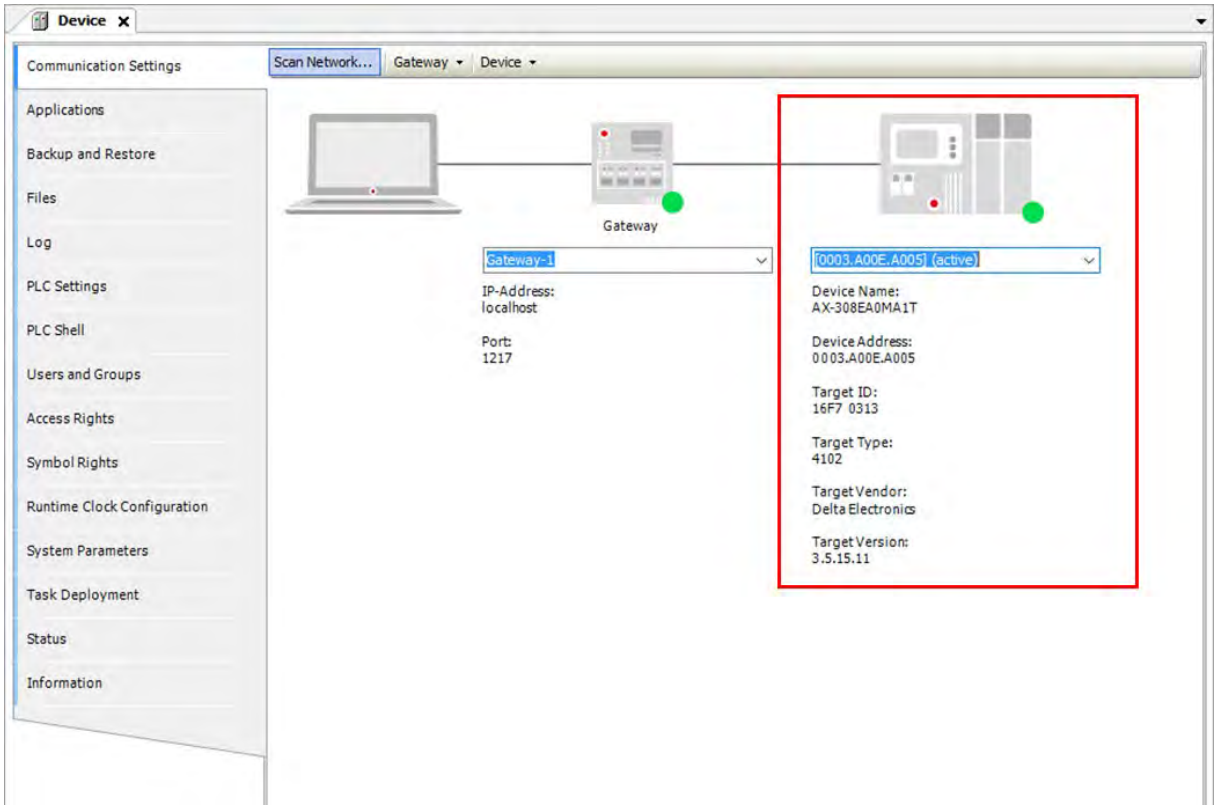
2. Choose “Scan Network” on the Device page.



3. After the “Select Device” window pops up, select the “AX-308E” device and click on “OK”.

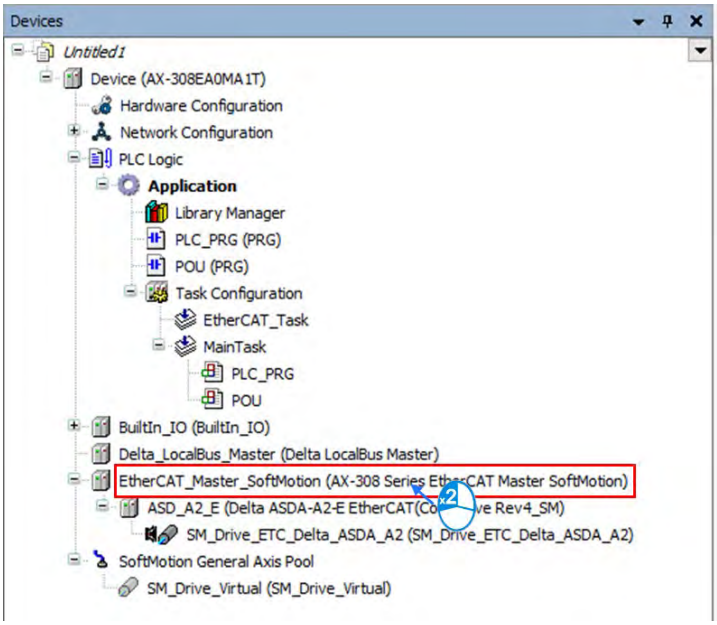


4. When the device is connected with your PC, information of the device will be displayed as shown in the following figure marked by the red box.

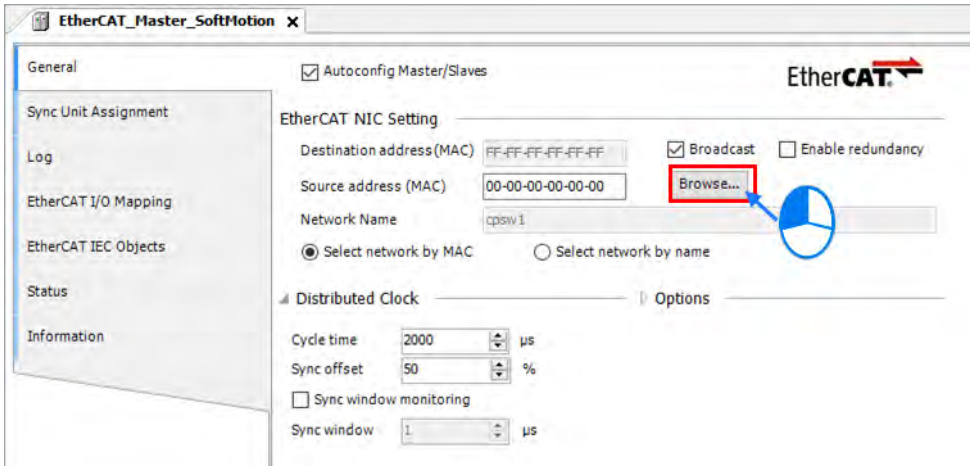


3.4.3.2 Process Monitoring and Control

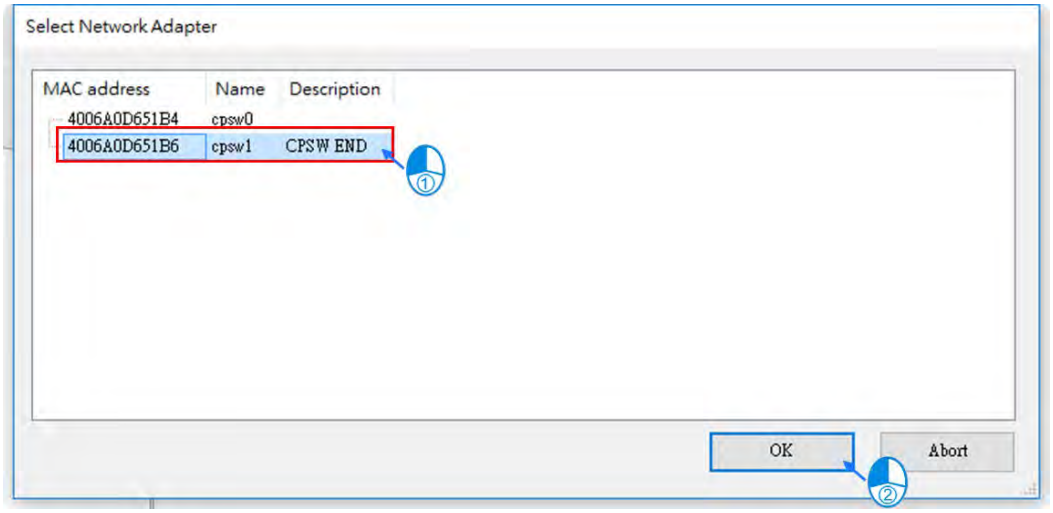
1. The correctness of EtherCAT communication should be checked before program monitoring starts. After confirmation, double click on “EtherCAT_Master_SoftMotion”.



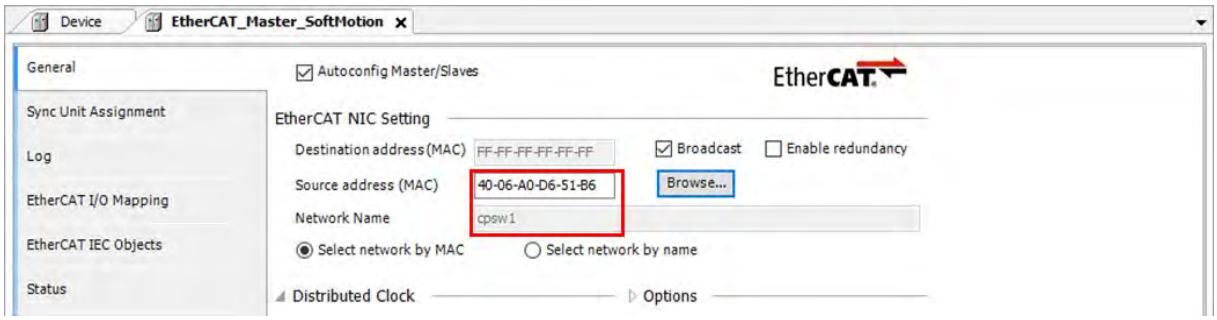
2. Choose “Browse” after entering the EtherCAT_Master_SoftMotion page.



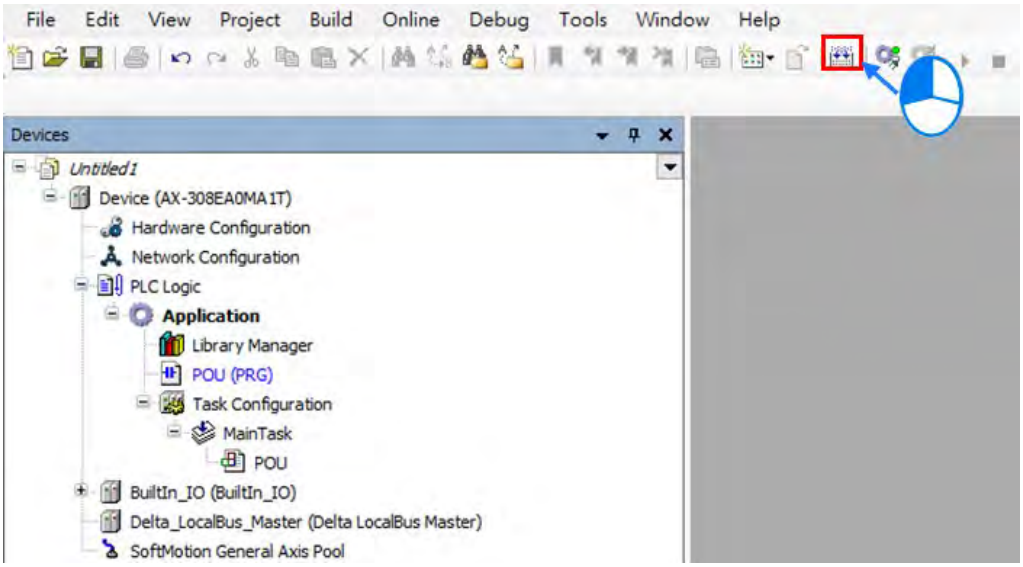
3. Select EtherCAT port “cpsw1” and click “OK”.



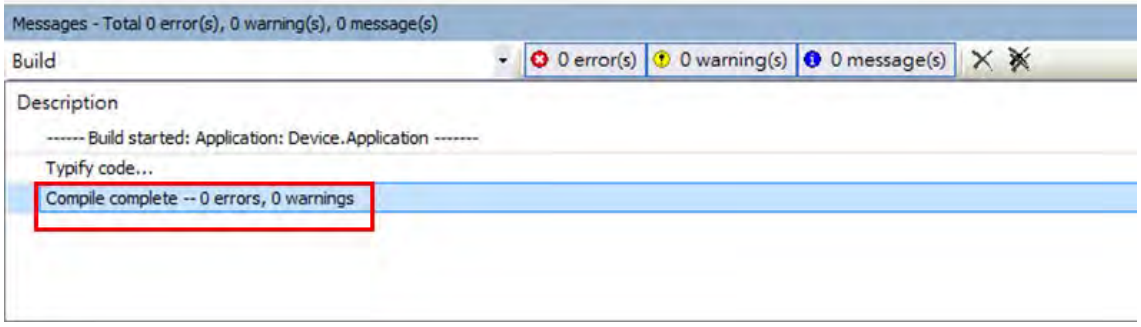
4. After finish choosing network adaptor, the setting value of Source address (MAC) is displayed.



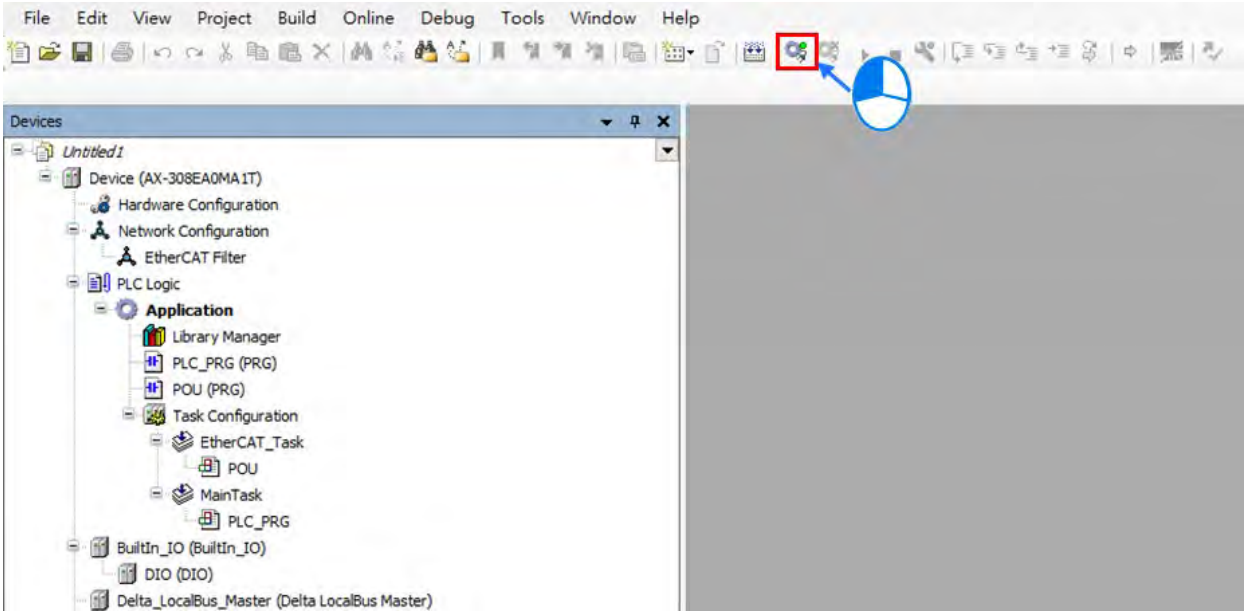
5. Click on the Compile button  to verify the correctness of program.



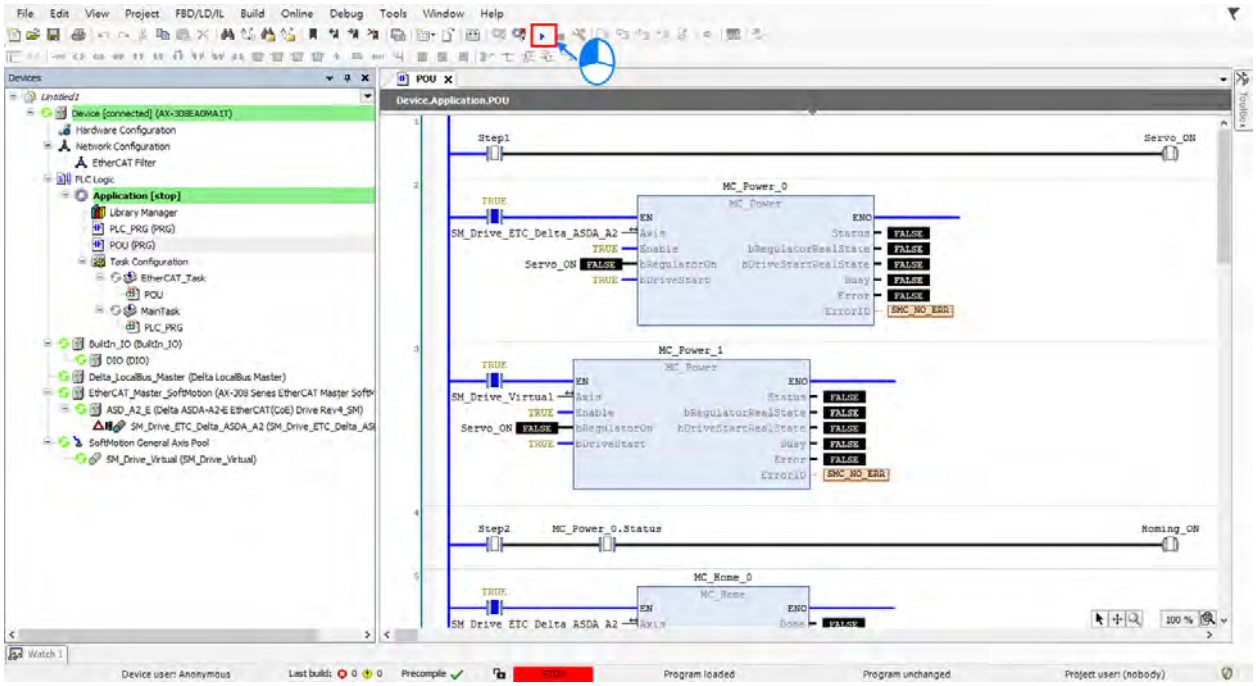
6. Upon completion of compiling, a report message of programming errors and warnings is displayed.



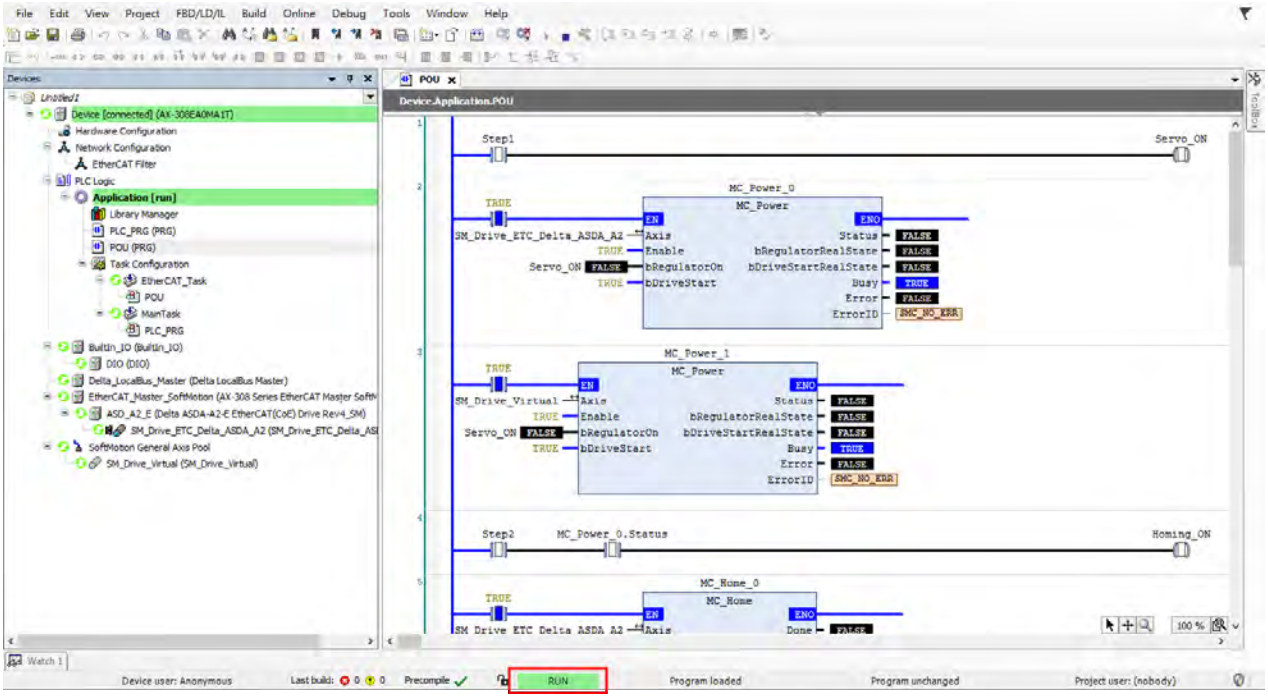
7. And then click on  to perform online monitoring.



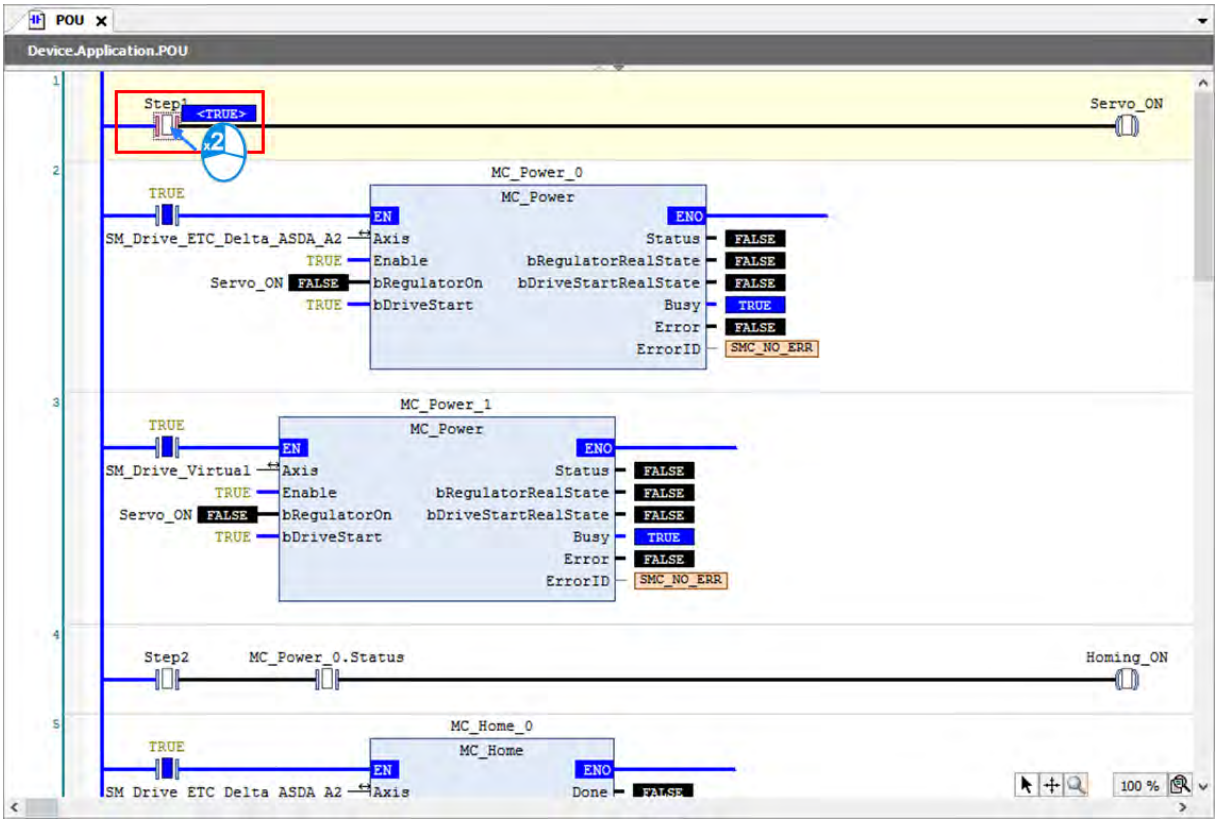
8. The program needs to be downloaded after monitoring action is performed. When the download is completed, the status of PLC shows STOP until you click on and the status would shift to RUN.



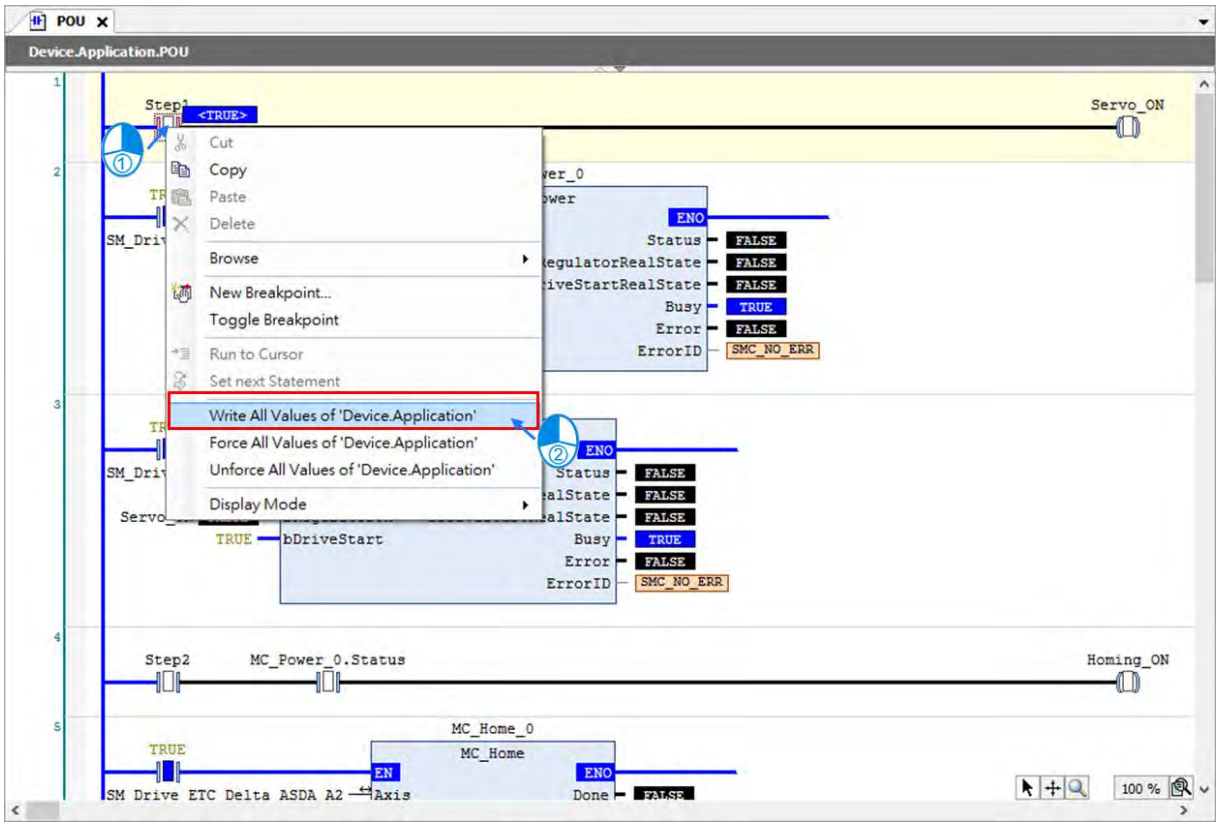
9. When the PLC is in the RUN mode, the status would be displayed as RUN at the bottom of the page and the program is ready to be executed.



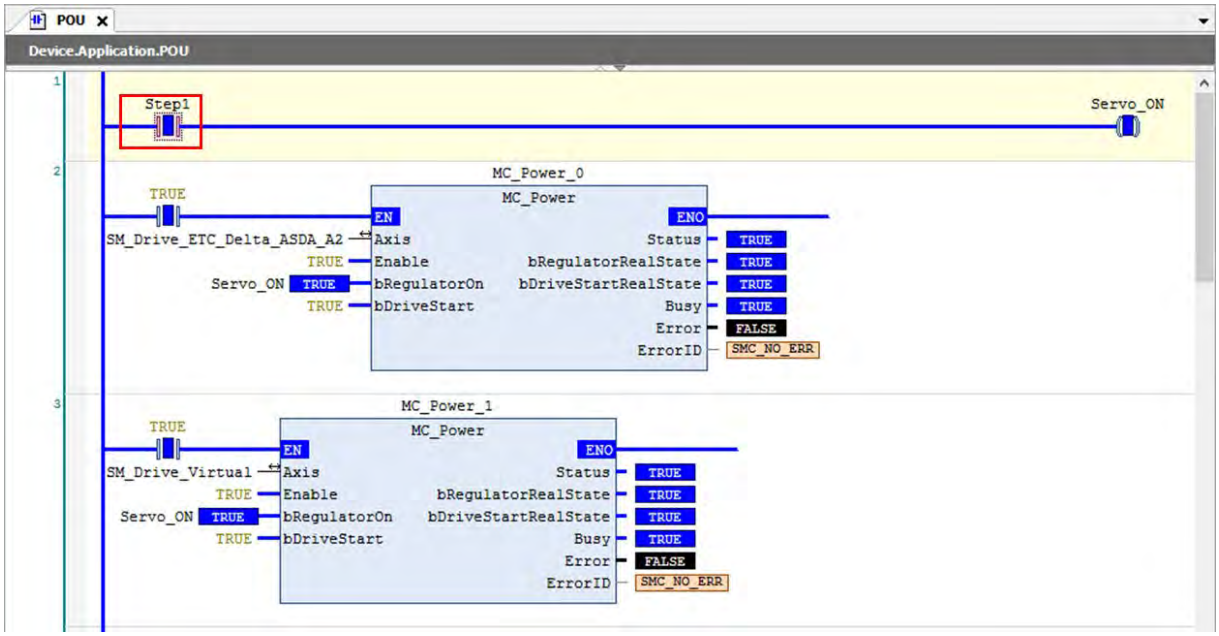
10. Double click on “Step 1” and the icon <TRUE> will be shown.



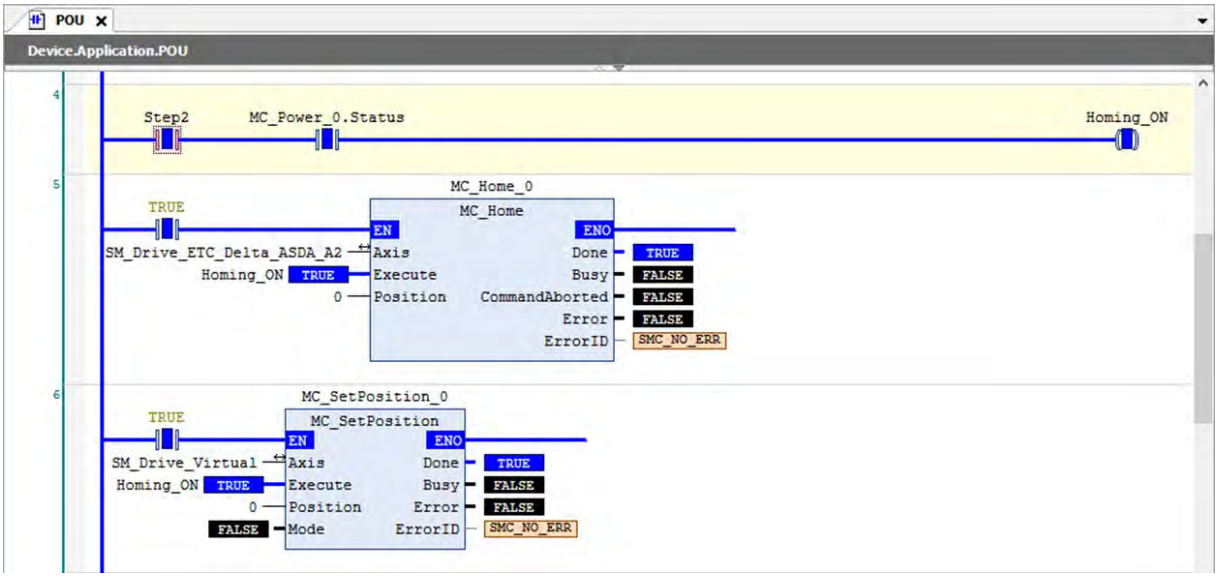
11. Right click on “Step 1” to select “Write All Values of ‘Device.Application” and all the corresponding status will be written in, which can also be performed by using the shortcut key Ctrl + F7.



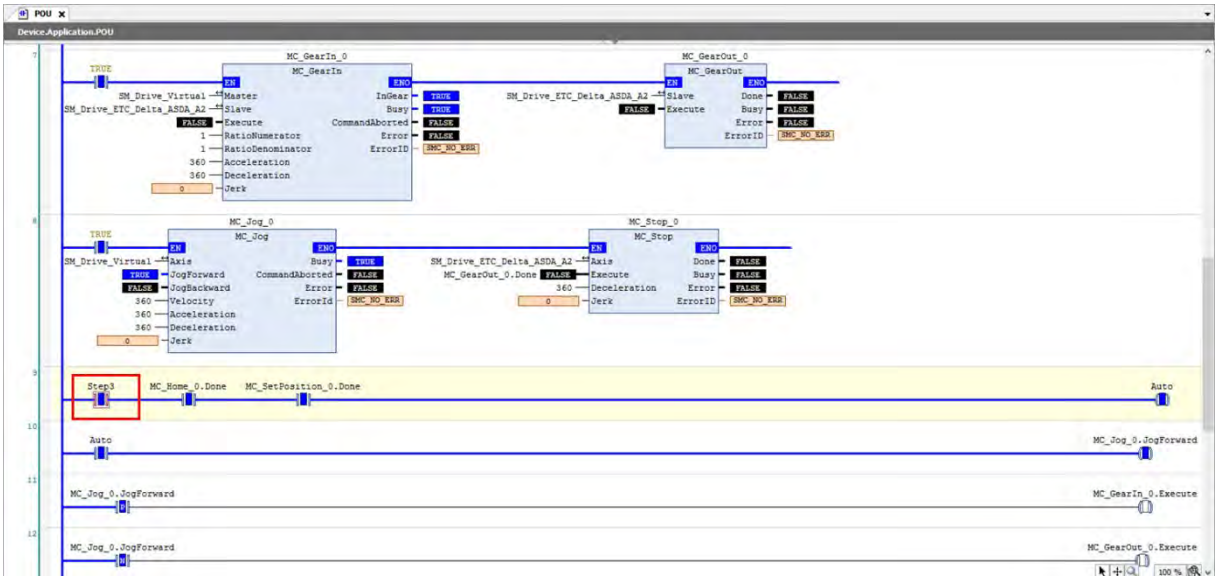
12. After activating “Step 1”, both axes are under Servo ON mode. (Same operation to execute on the following programs.)



13. Start homing for the real and virtual axis by activating “Step 2”.

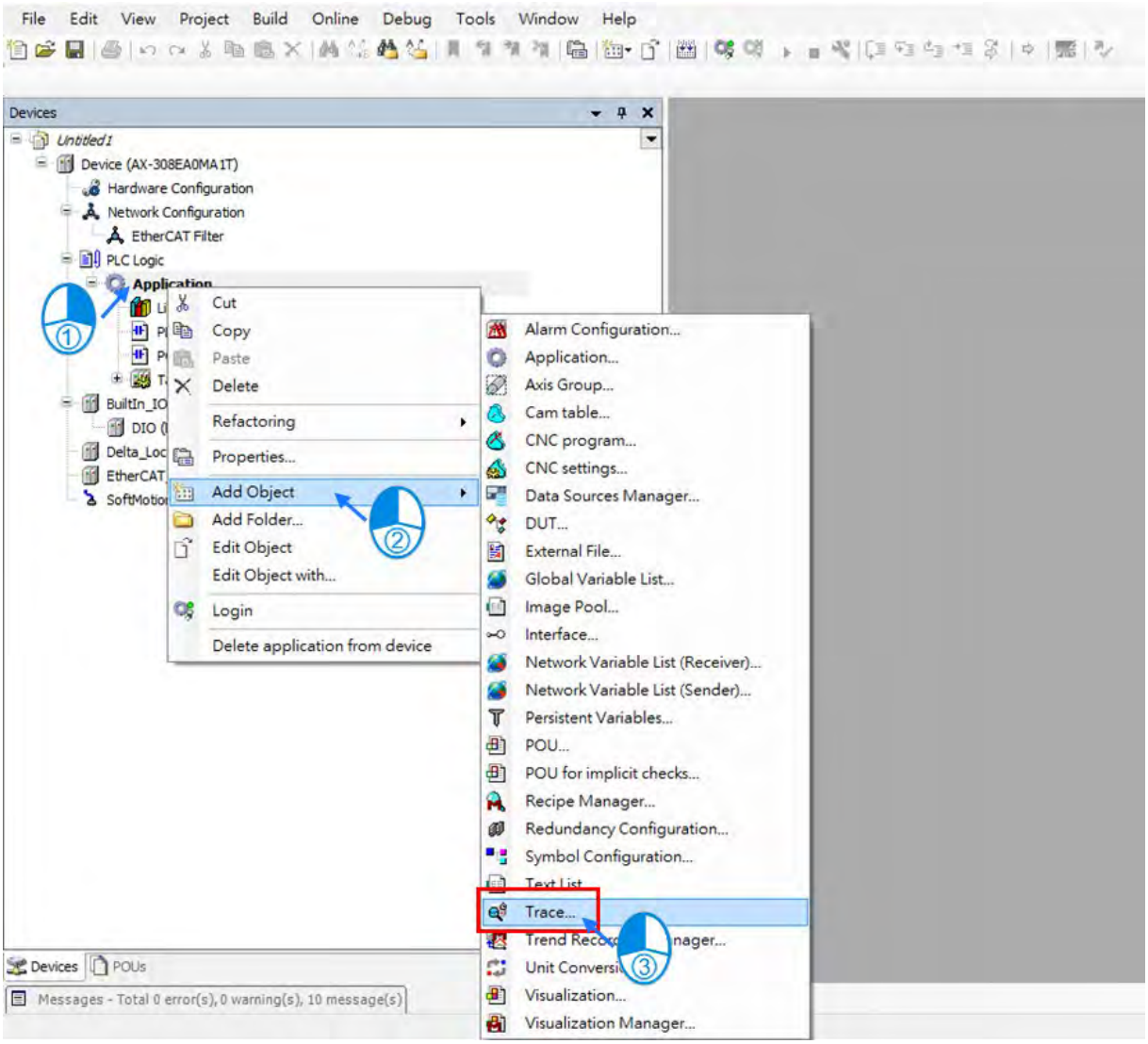


14. After activating "Step 3", “MC_Jog” is triggered to enable the virtual axis to move, which also triggers “MC_GearIn” to perform the following move of slave axis. When “Step3” turns False, “MC_Jog” and “MC_GearOut” are triggered at the same time and the virtual axis stops to move. The slave axis moves at a constant speed after being decoupled, then the function block “MC_Stop” will be executed to command a motion stop to the axis.

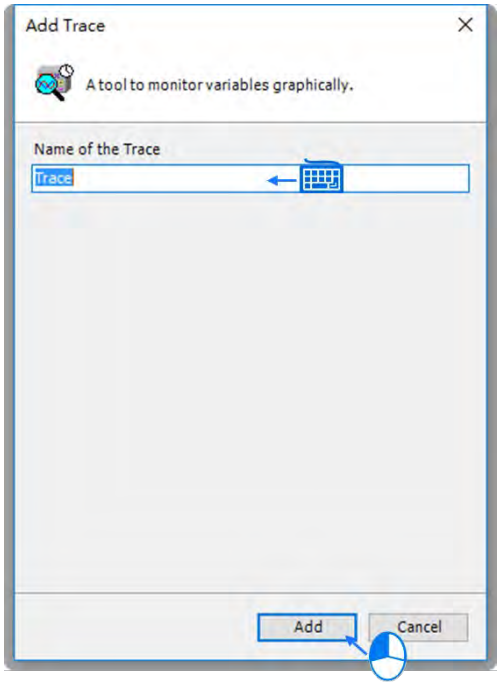


3.4.3.3 Monitor with Trace Tool

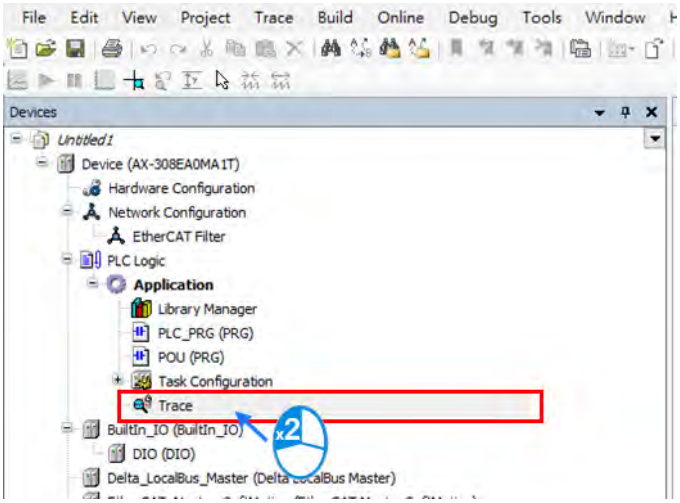
1. To monitor the status of input/ output pins and axes with Trace function while executing the program, navigate to “Application” > “Add Object” > “Trace”. Then you’ll open the “Add Trace” window.



2. Insert the name of Trace then click “Add” on “Add Trace” window.

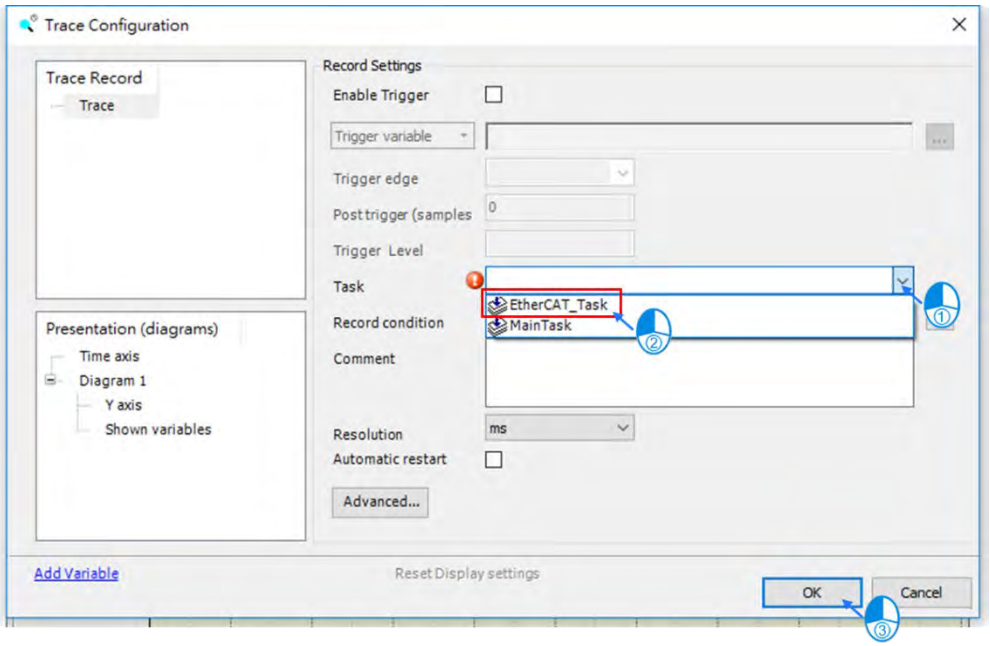


3. After the Trace is successfully added, it will be shown in the project tree on the left side of the screen. Then double click on “Trace” to open the Trace page.



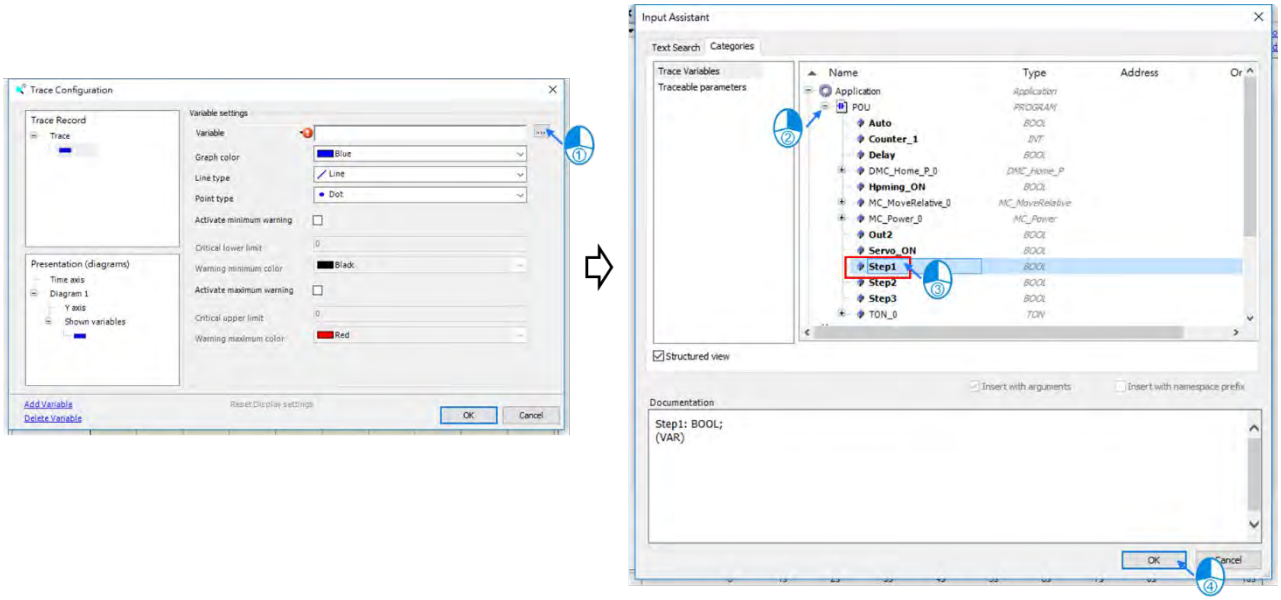
4. Click on “Configuration” on the Trace page to open the Trace Configuration window.

5. Select “EtherCAT_Task” from the drop-down list of Task on the Trace Configuration window.

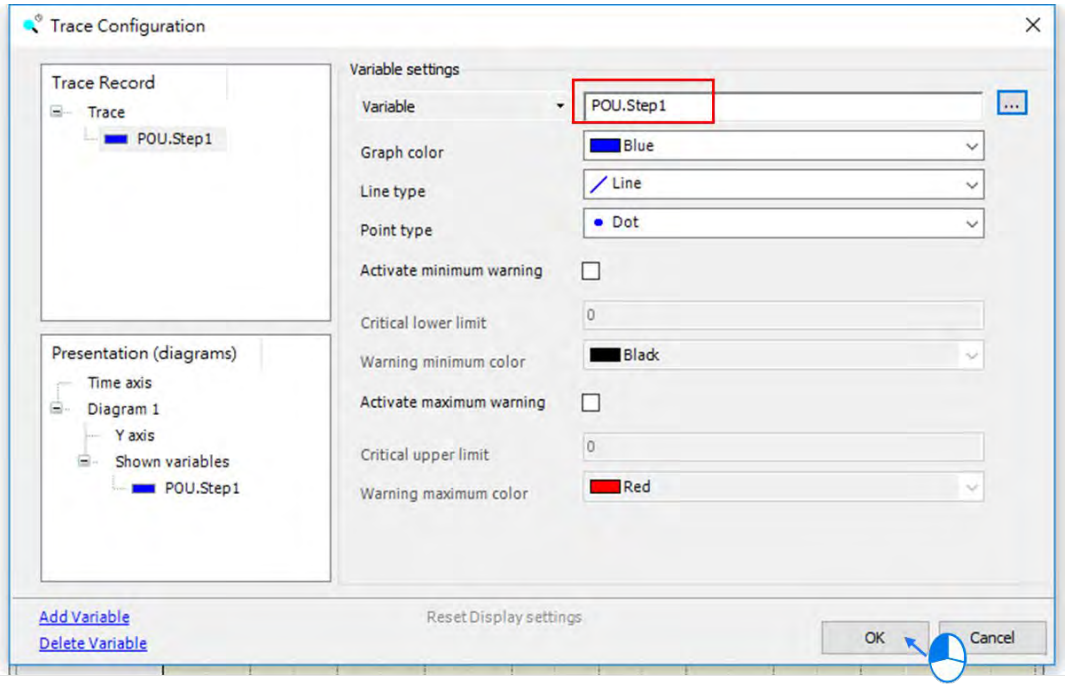


6. After finishing configuration, click “Add Variable” on the Trace page to open the Trace Configuration window.

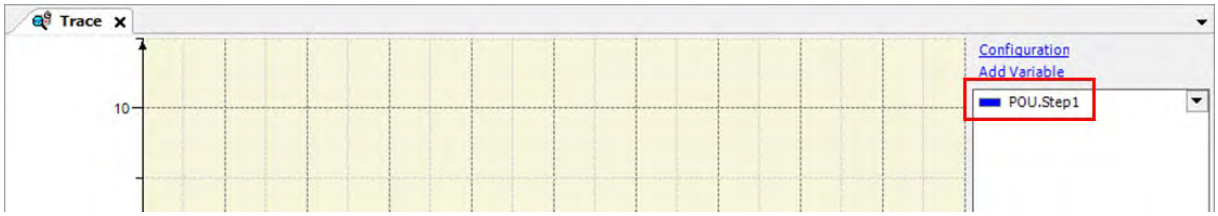
7. Click the button ... on the Trace Configuration page to add the required trace variables or traceable parameters. After the Input Assistant window is opened, expand POU on the tree and select “Step 1” then click on “OK”.



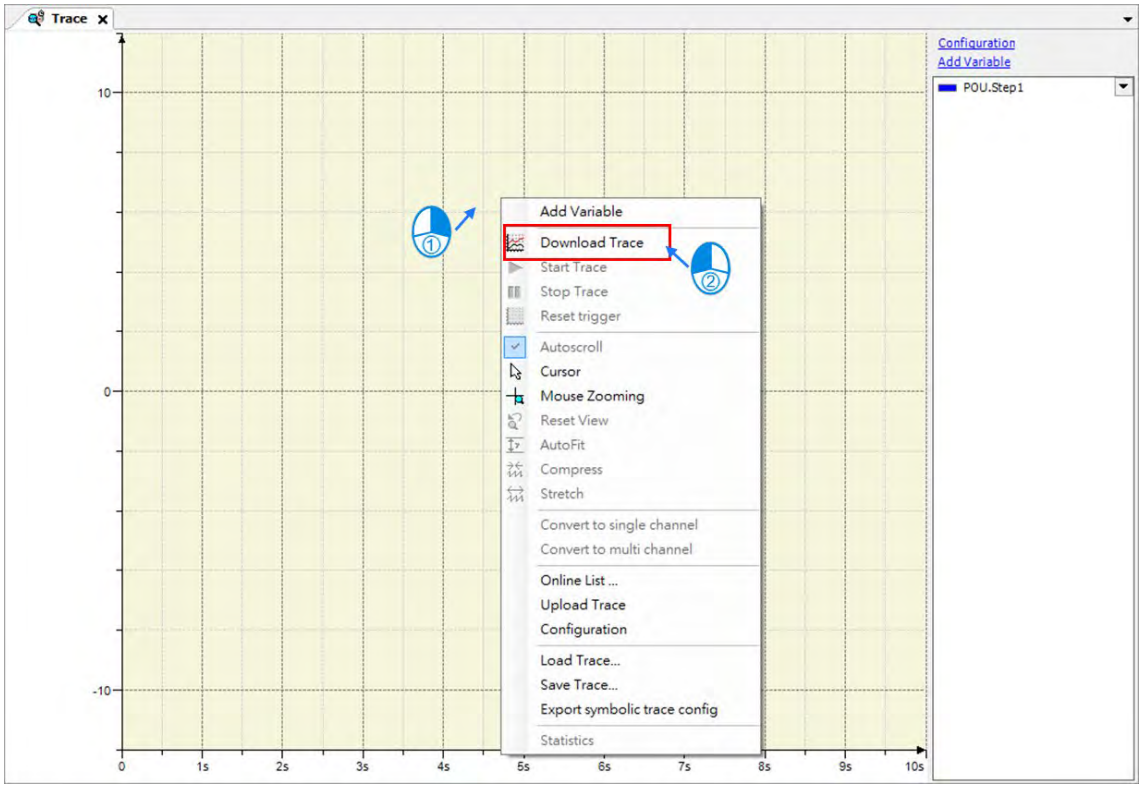
8. Make sure the chosen variable is shown in the variable field of variable settings and then click “OK”.



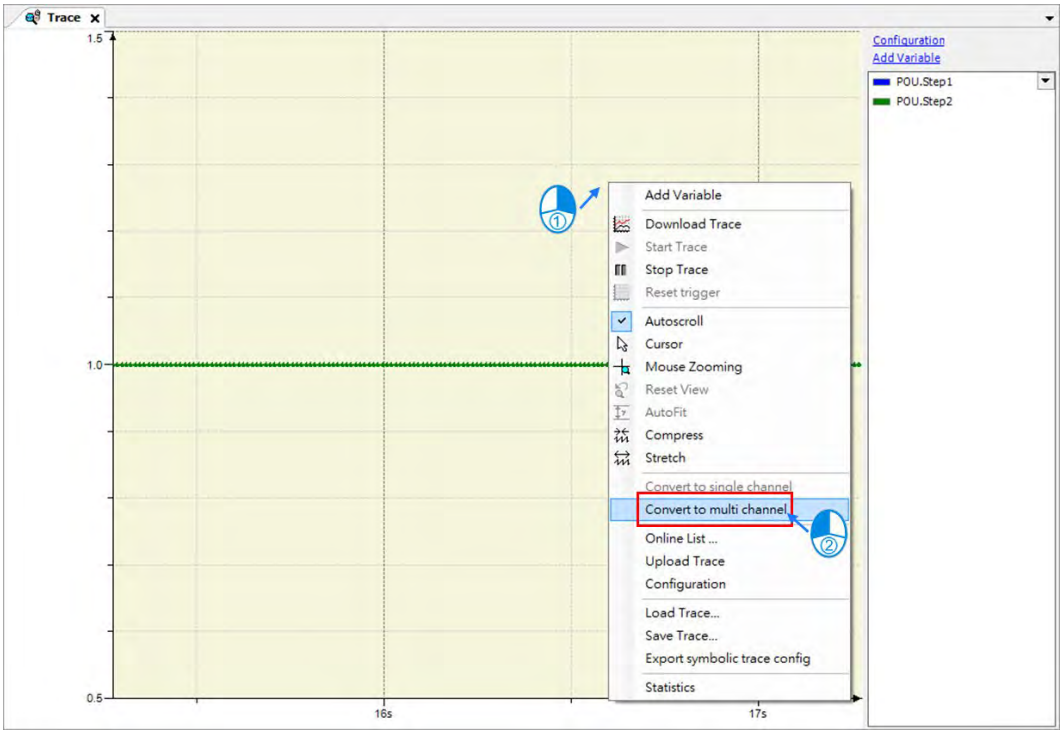
9. Afterwards, the added variable will be shown on the right side of the Trace page.



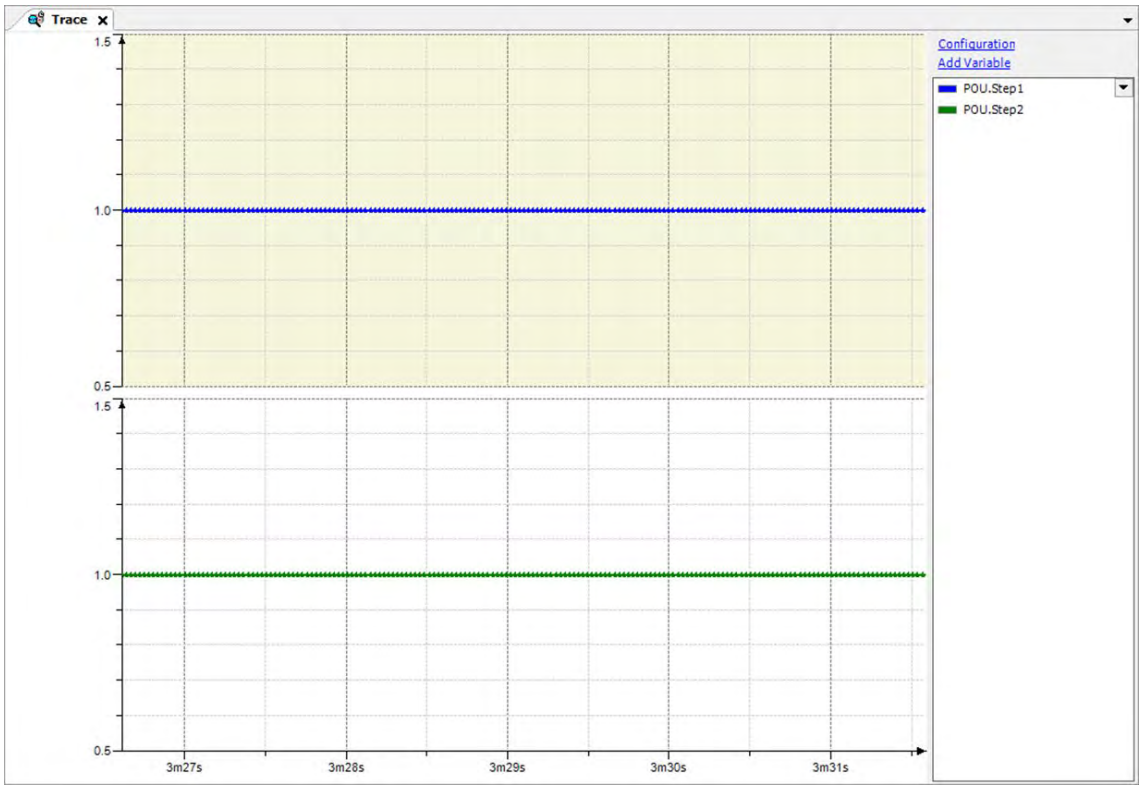
10. Right click on the Trace and select “Download Trace” to start monitoring.



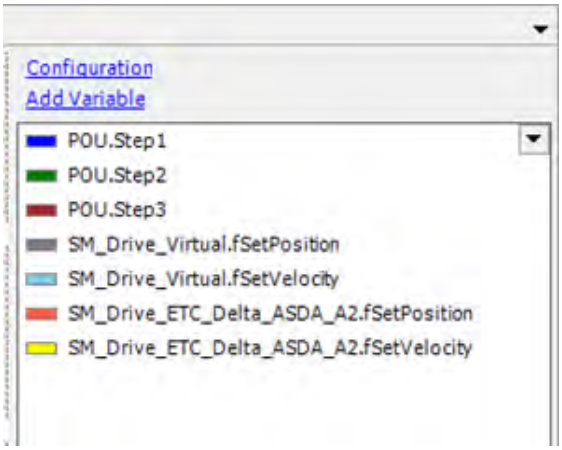
11. In case that there’re more than one variable or parameter need to be monitored, you can just right click on the scope and select “Convert to multi channel”.



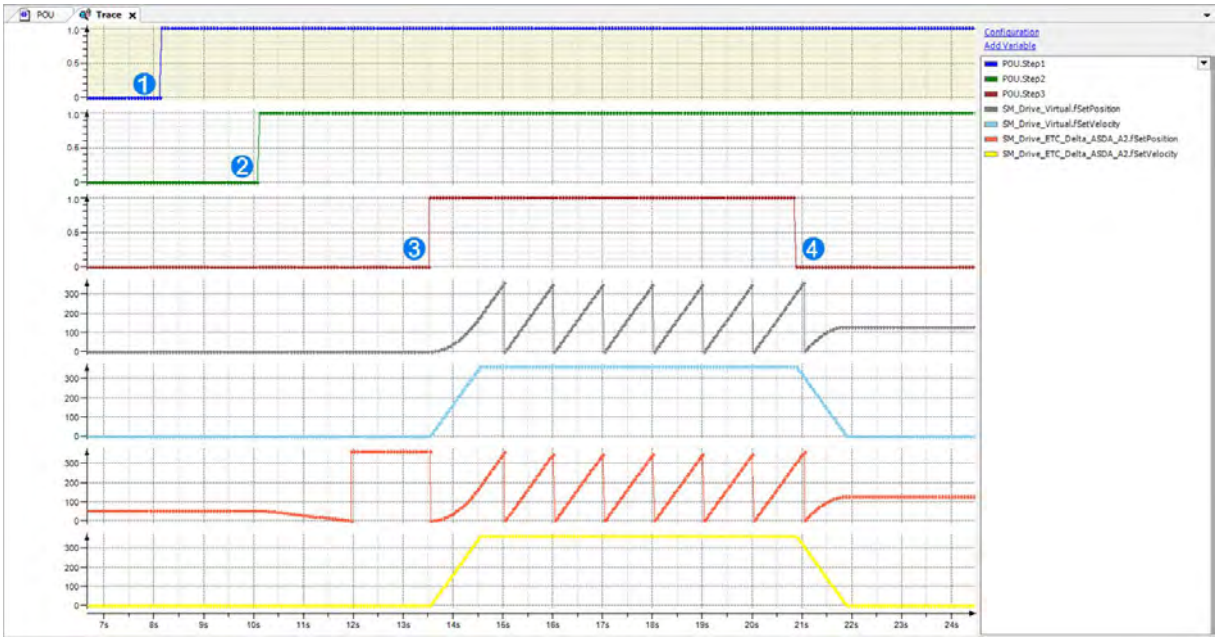
12. After choosing “Convert to multi channel”, two data-recording oscilloscope will be displayed on the page for the two chosen variables added on the right.



13. Repeat the above steps to add more required trace variables and traceable parameters.



14. While a program is running, you can observe trigger events and the current position of axis as well as the velocity via the data-recording oscilloscope.



Introduction of the process shown on the scope:

① Execute “Step1” to activate the real axis and also the virtual axis.

② Execute “Step2” to make both axes perform homing.

③ Execute “Step3” to trigger “MC_Jog” to activate the master axis. Meanwhile, “MC_GearIn” is also executed to command the slave axis move which follows the master axis.

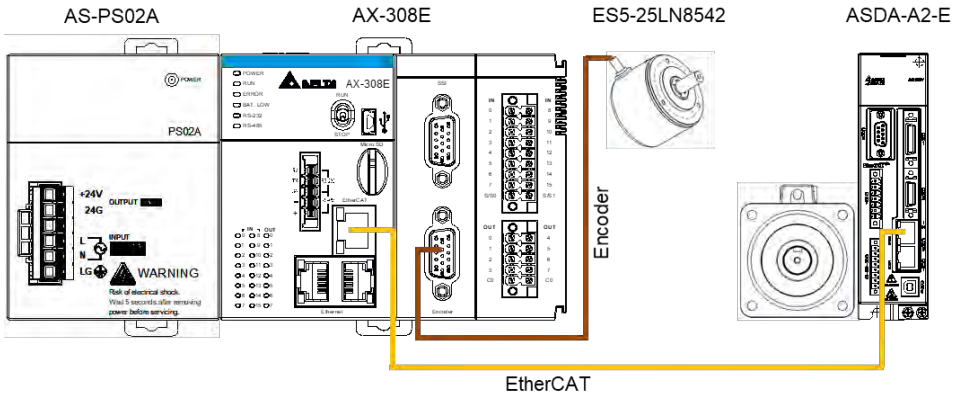
④ After “Step3” turns False, the master axis will be stopped by “MC_Jog” and “MC_GearOut” is triggered at the same time to stop the virtual axis. Then the slave axis moves at a constant speed after being decoupled. Finally, the function block “MC_Stop” is executed to command a motion stop to the axis.

3.5 Setup Projects with Built-in Encoder as Master

This example presents a servo motor with a master encoder to perform axis coupling.

3.5.1 System Hardware Configuration

AX-308 connects to a servo driver (ASDA-A2-E) directly through EtherCAT communication port to control the positioning of servo motors which follows the virtual axis.



- Tools and materials

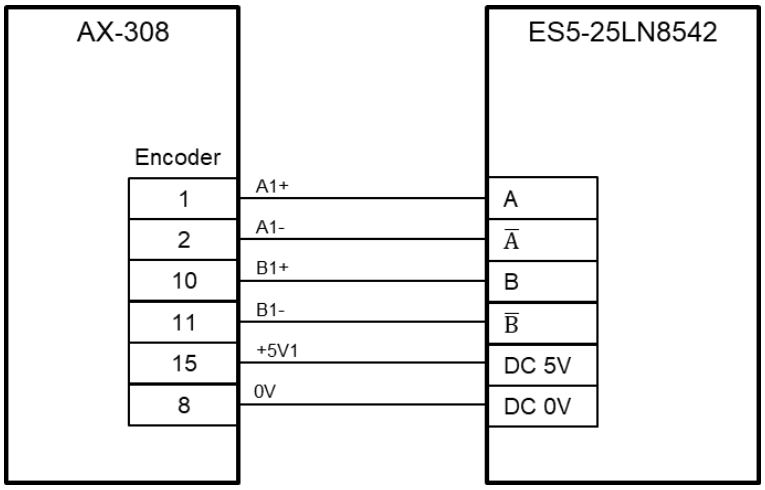
Numbering	Item	Model Number	Q'ty
1	AS Series Power Module	AS-PS02A	1
2	AX-3 Series CPU Module	AX-308EA0MA1T	1
3	Encoder	ES5-25LN8542	1
4	Power supply	DVP-PS02	1
5	Servo drive (EtherCAT)	ASDA-A2-0421-E	1
6	Servo motor	ECMA-C10604ES	1
7	Servo power cable	ASD-ABPW0003	1
8	Servo encoder cable	ASD-ABEN0003	1
9	EtherCAT communication cable (1 meter)	UC-EMC010-02A	1

- Configuration between encoders and motion controllers

  ◆ Introduction to encoder's function

Numbering	Color	Function
1	Brown	5V
2	Blue	0V
3	Black	A
4	Black/Red	$\overline{A}$
5	White	B
6	White/Red	$\overline{B}$
7	Orange	Z
8	Orange/Red	$\overline{Z}$

  ◆ Connect the encoder to the controllers

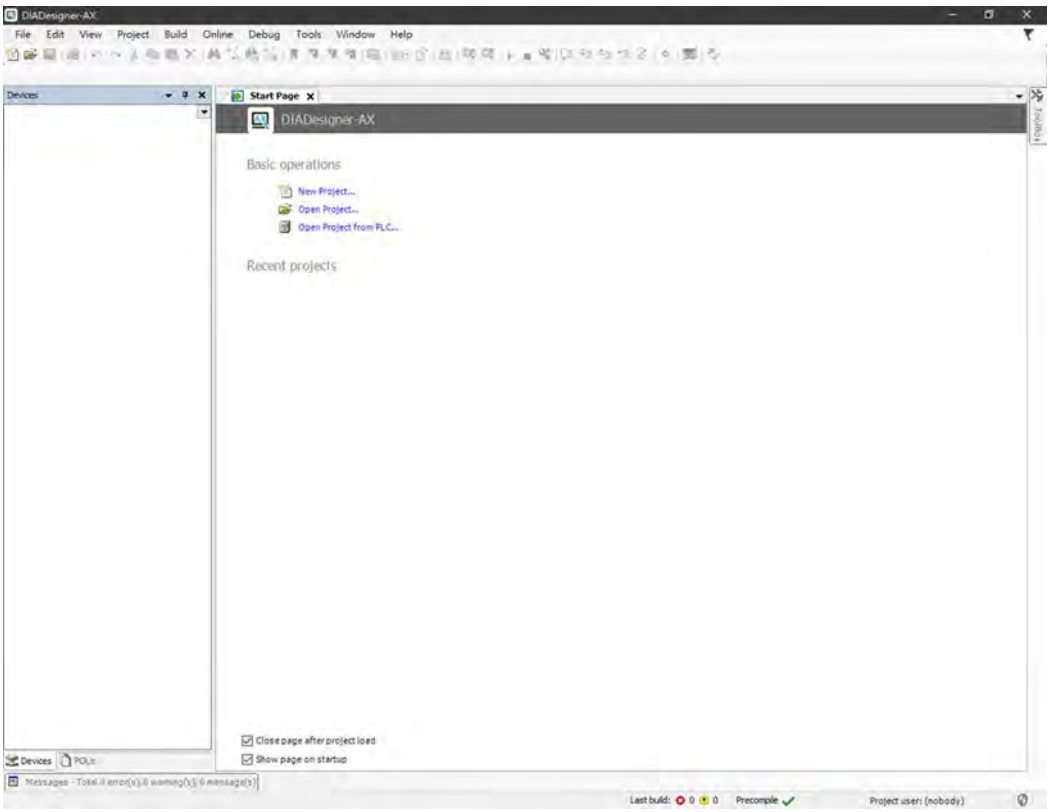


3.5.2 Create New Projects and POUs

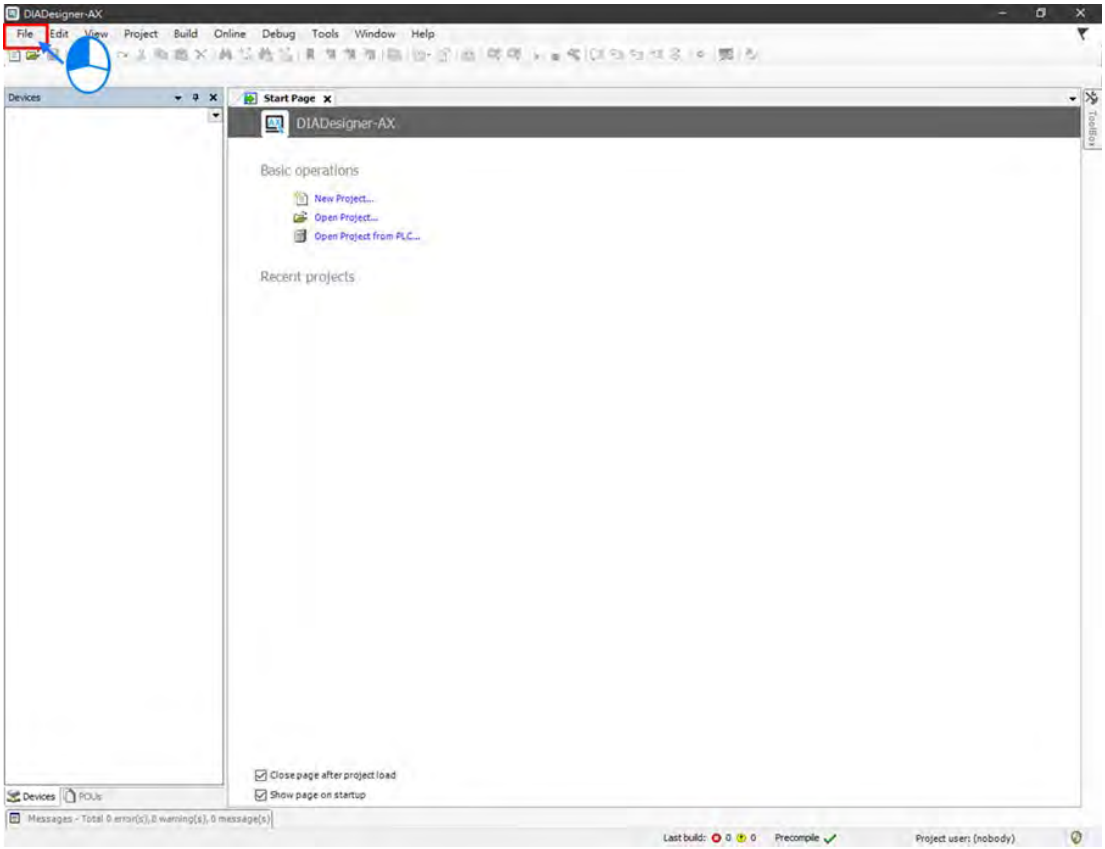
3.5.2.1 Create a New Project

1. First, open the programming software DIADesigner-AX.

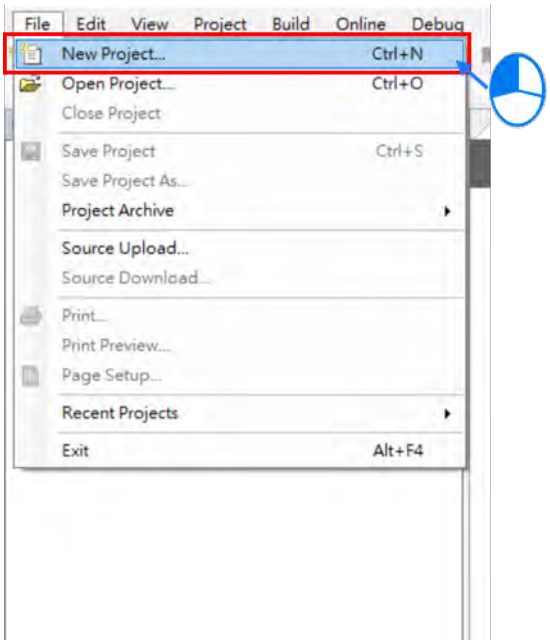
2. The display of DIADesigner-AX is shown as below.



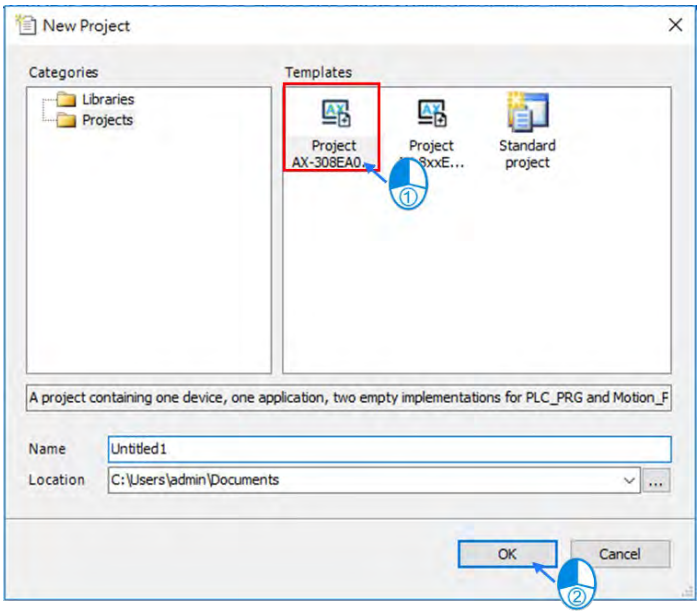
3. Click on “File”.



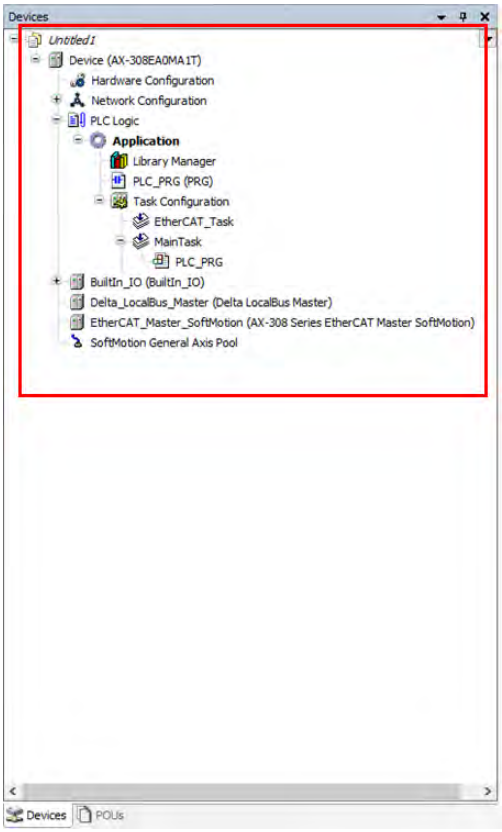
4. To create a new project, select "New Project" from the drop-down menu for DIADesigner-AX V1.0.0 or select “Standard Project” from the drop-down menu for DIADesigner-AX V1.1.0 or later.



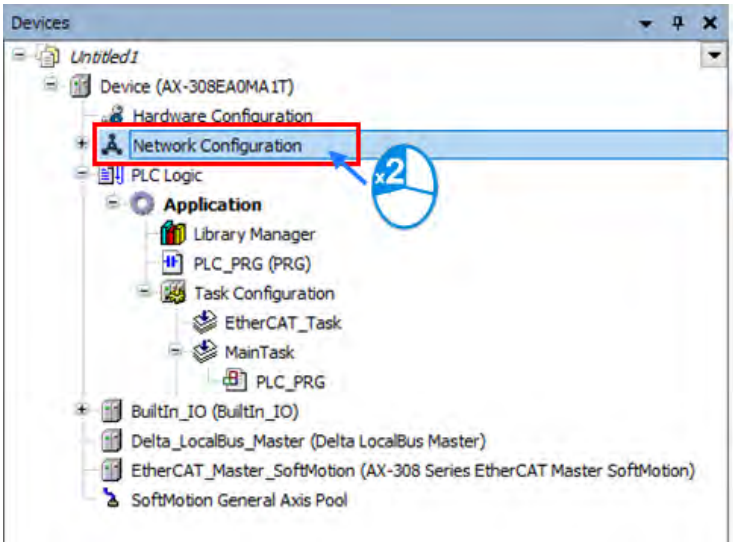
5. Select “Project AX-308EA0MA1T” after entering New Project page. Name the project in the field of Name and choose a path to the archive location, then click OK to move to the next page.



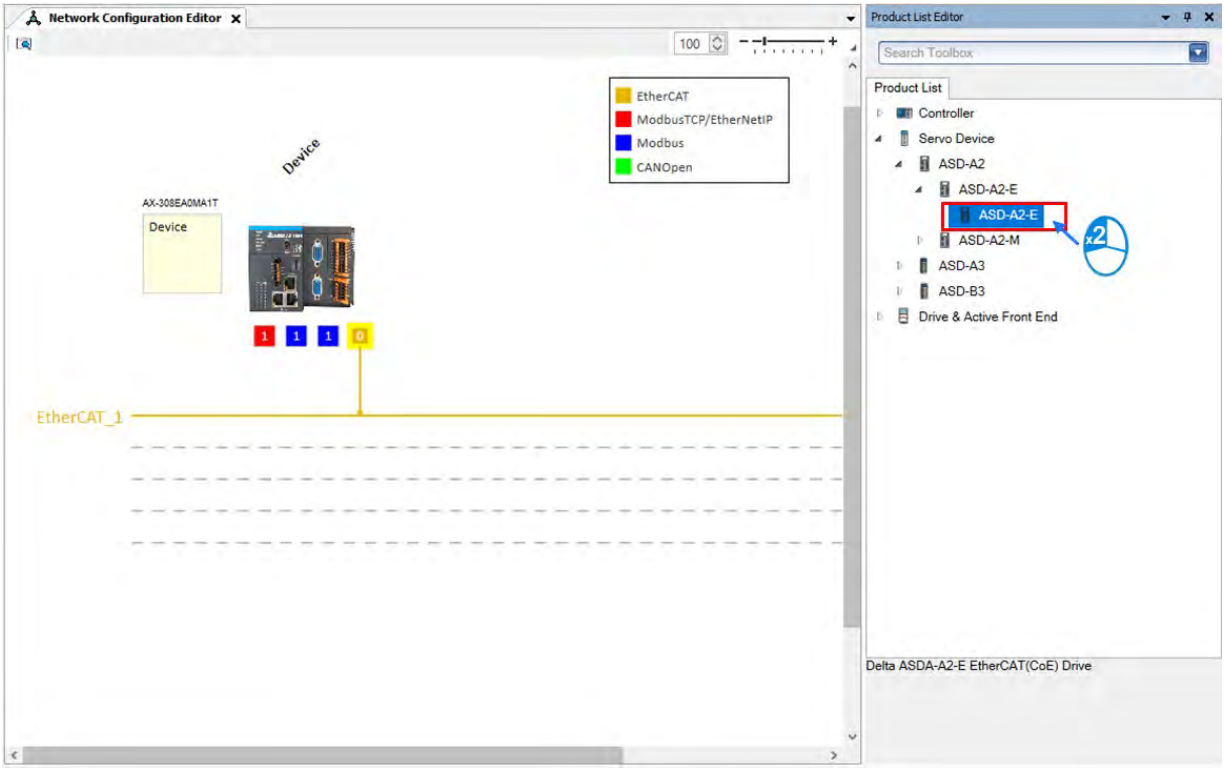
6. After entering the project page, double click the target object in the project tree on the left side of the page to open a certain program or configure settings of relating modules.



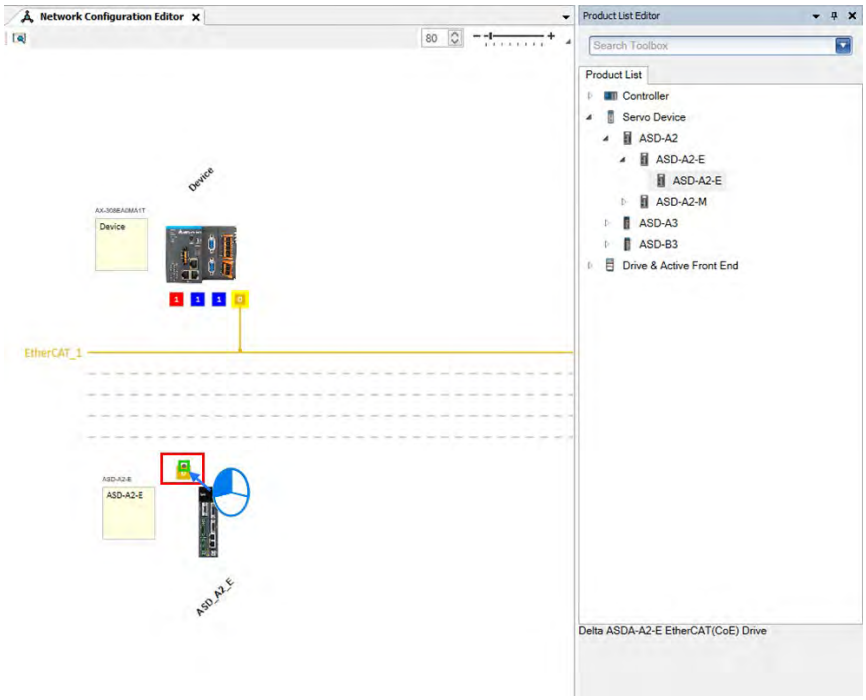
7. Double click on “Network Configuration” to create servo axes in the system.



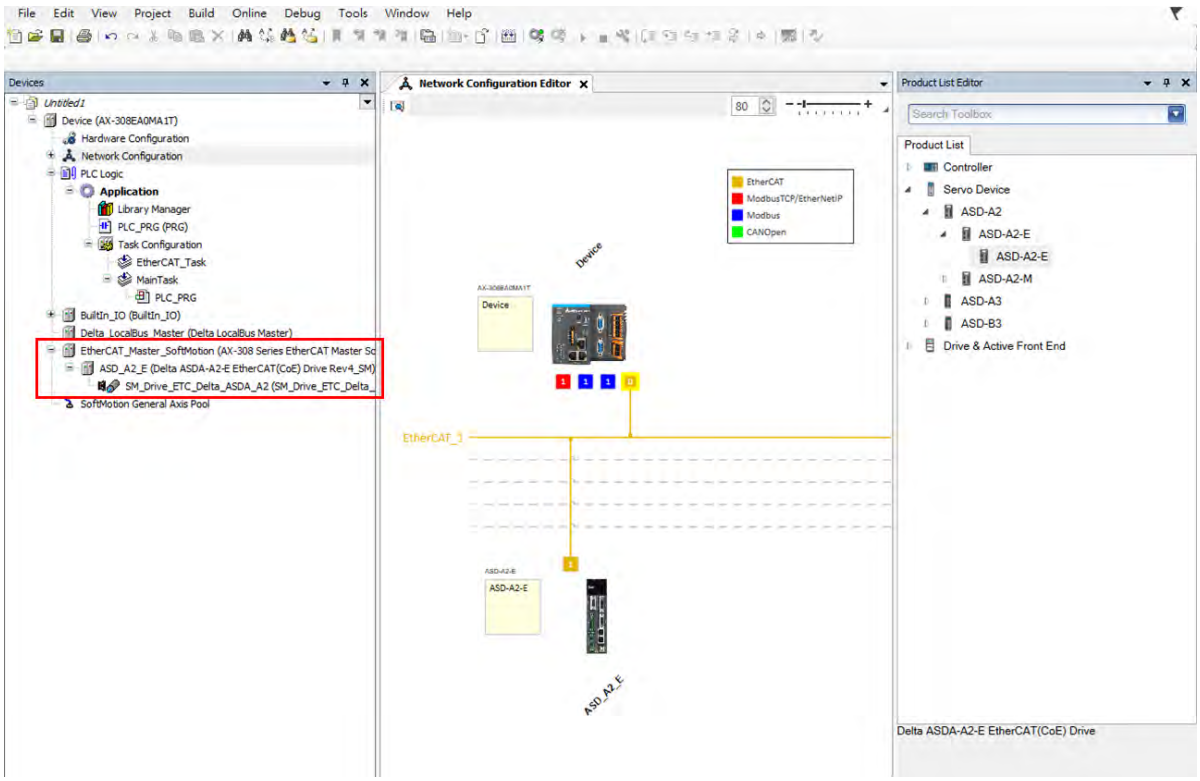
8. After entering “Network Configuration” page, select the servo drive slave based on EtherCAT and double click on it to add new axes.



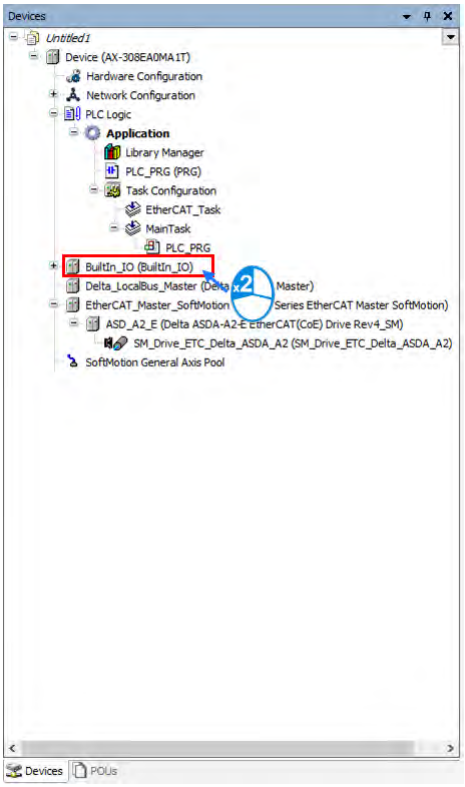
9. Move the cursor onto 0, then click and hold the left mouse button and drag the yellow line to connect with the line of EtherCAT_1.



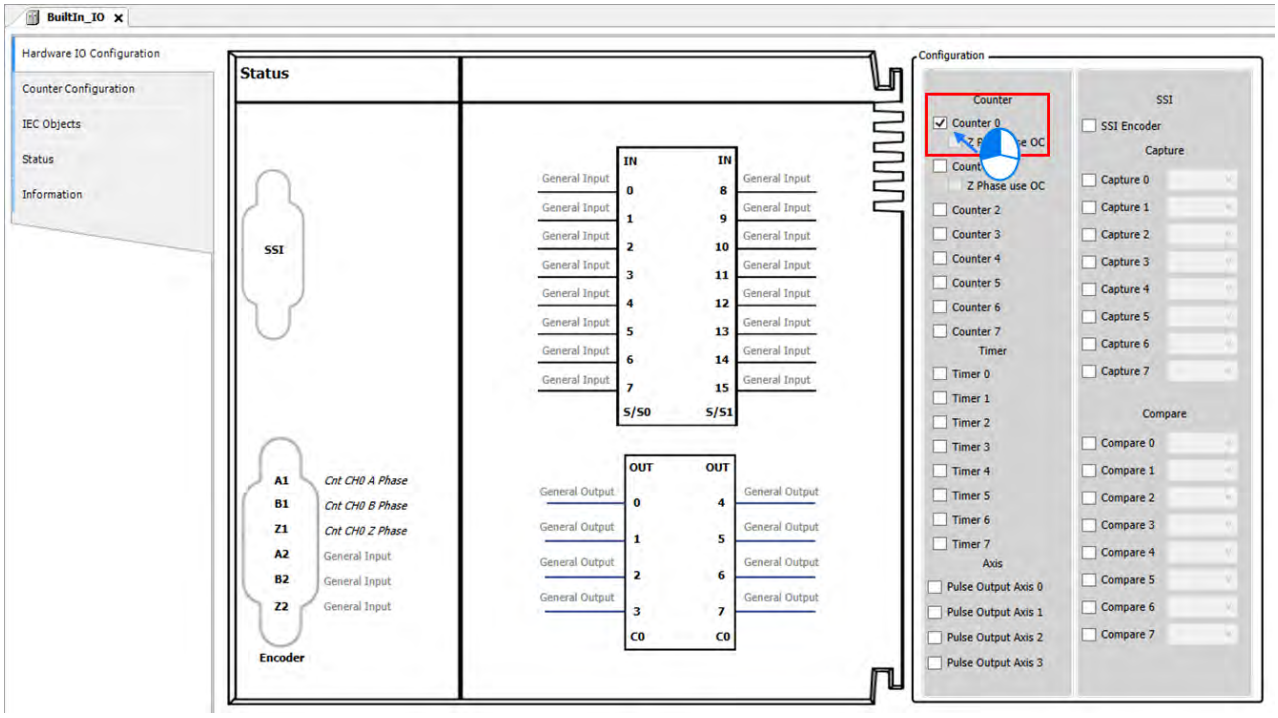
10. After connecting with EtherCAT_1, the slave device is displayed in the project tree (fields with red borders) which also means it is successfully configured with EtherCAT communication.



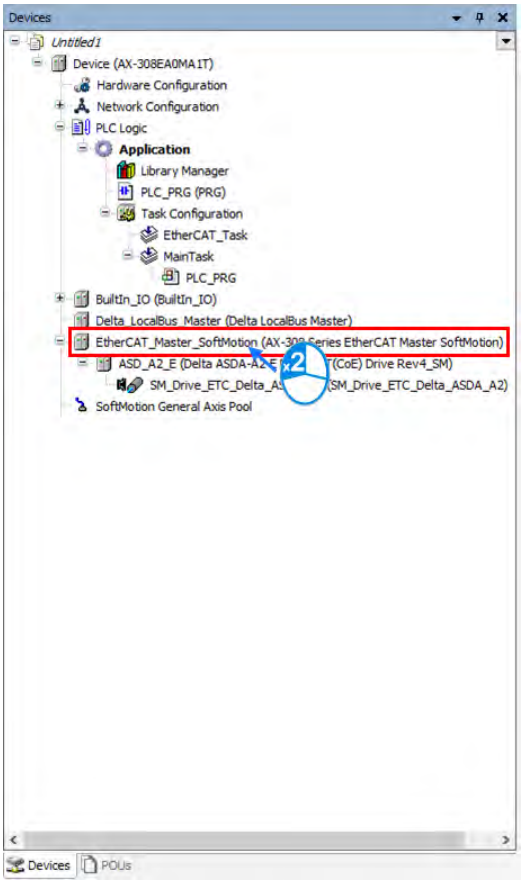
11. Then click “BuiltIn_IO” in the project tree to open the BuiltIn_IO page.



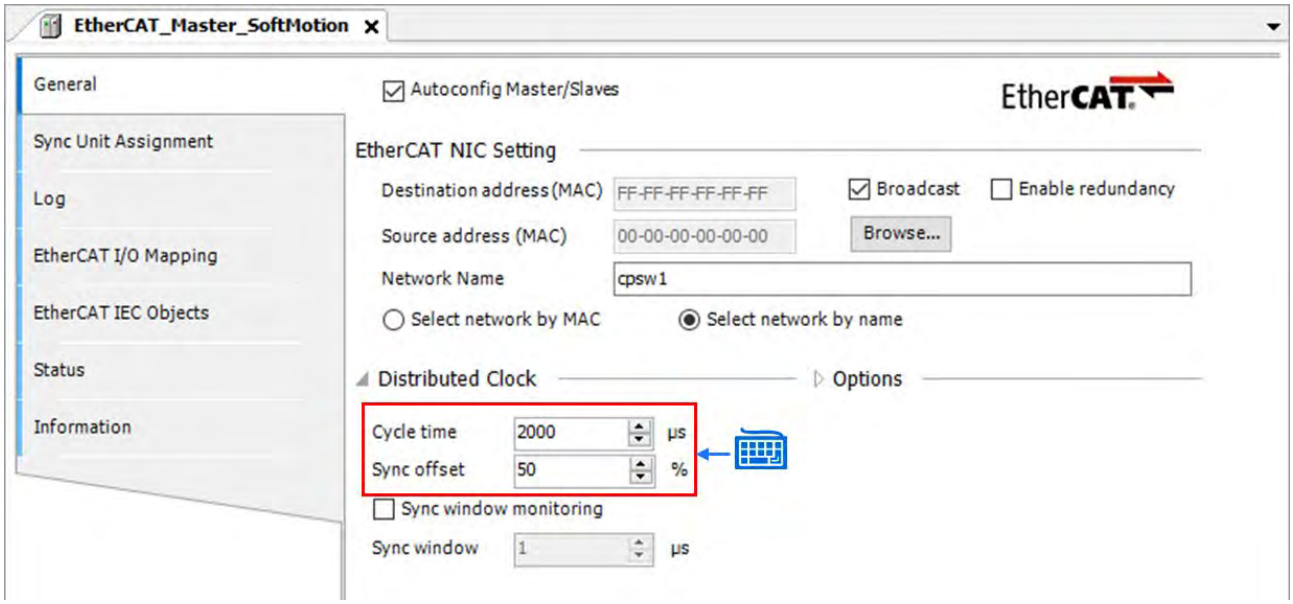
12. Select the checkbox of “Counter 0” since the first set is Counter in this example. To start the encoder, it must be configured with an external harware device.



13. Double click on “EtherCAT_Master_SoftMotion”.

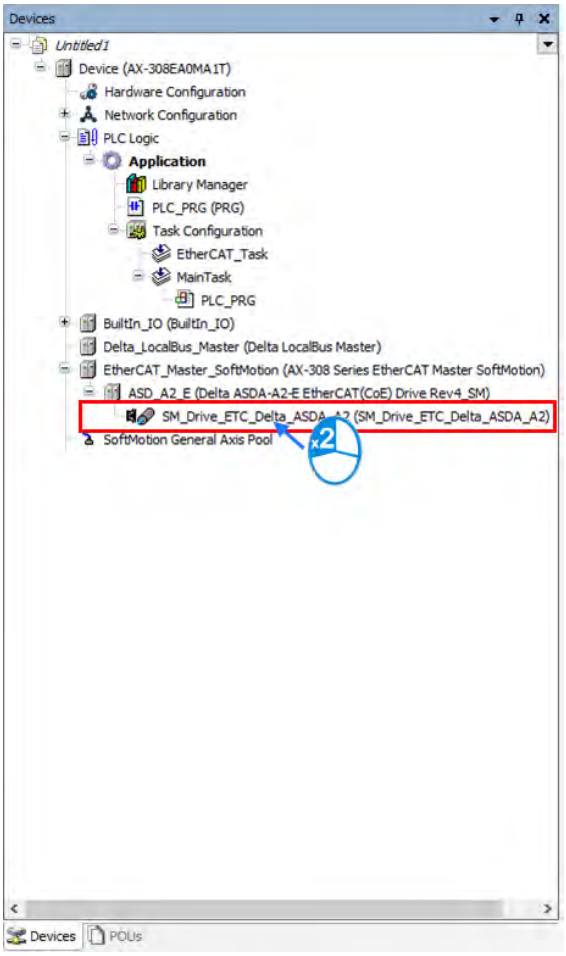


14. Set “Cycle time” to 2ms and “Sync offset” to 50 on the EtherCAT_Master_SoftMotion page.

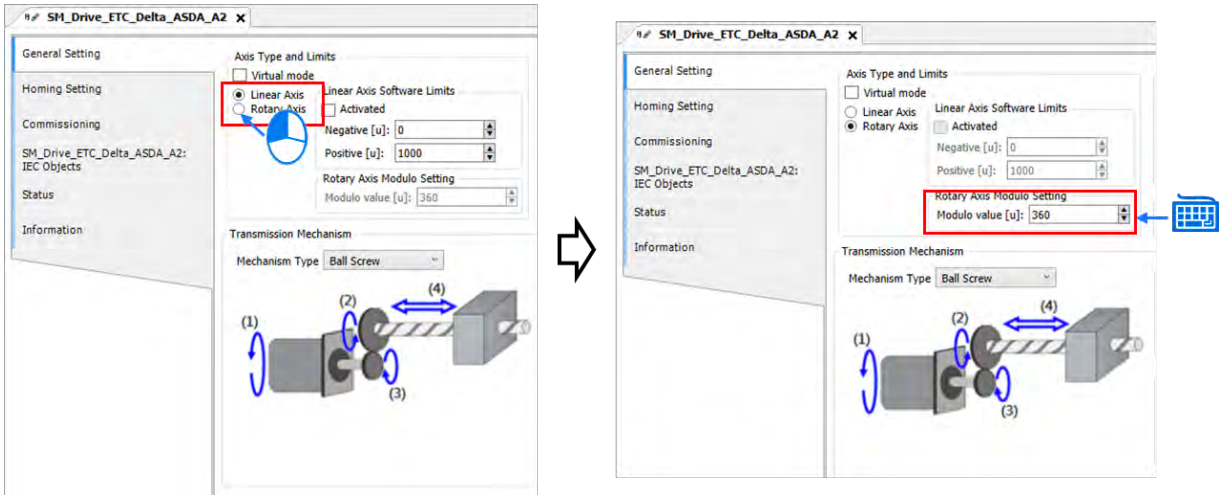


3.5.2.2 Axis Parameter Setting

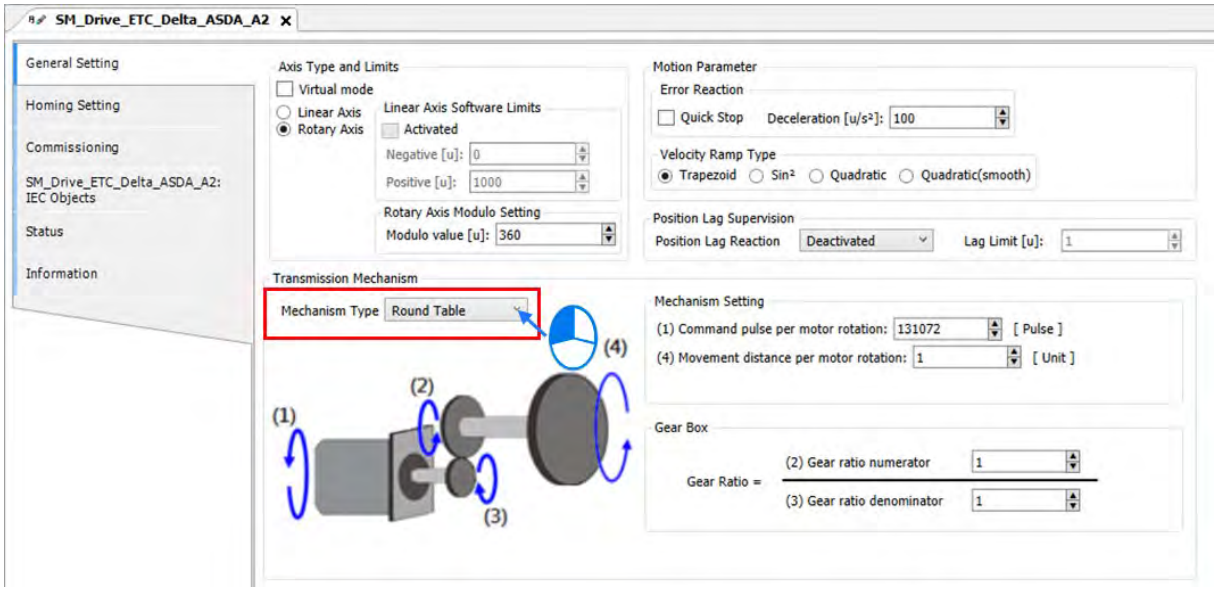
1. Select the first servo axis “SM_Drive_ETC_Delta_ASDA_A2” and double click on it.



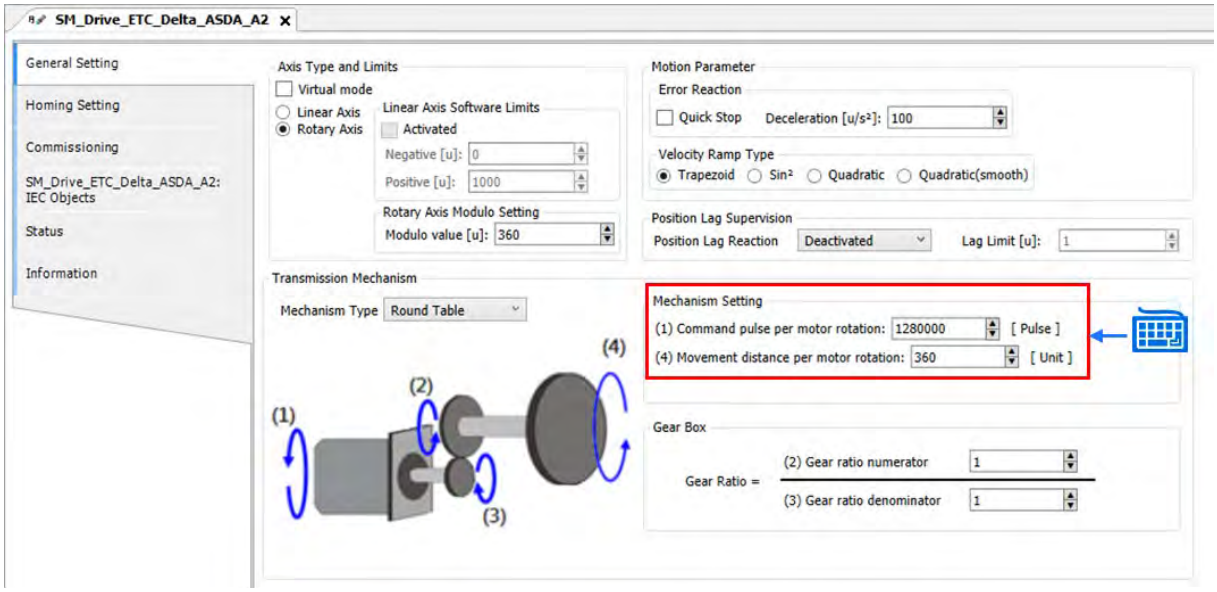
2. Set the axis type to “Rotary Axis” and you will be able to configure the max modulo value for “Rotary Axis Modulo Setting” with a default value of 360 degrees.



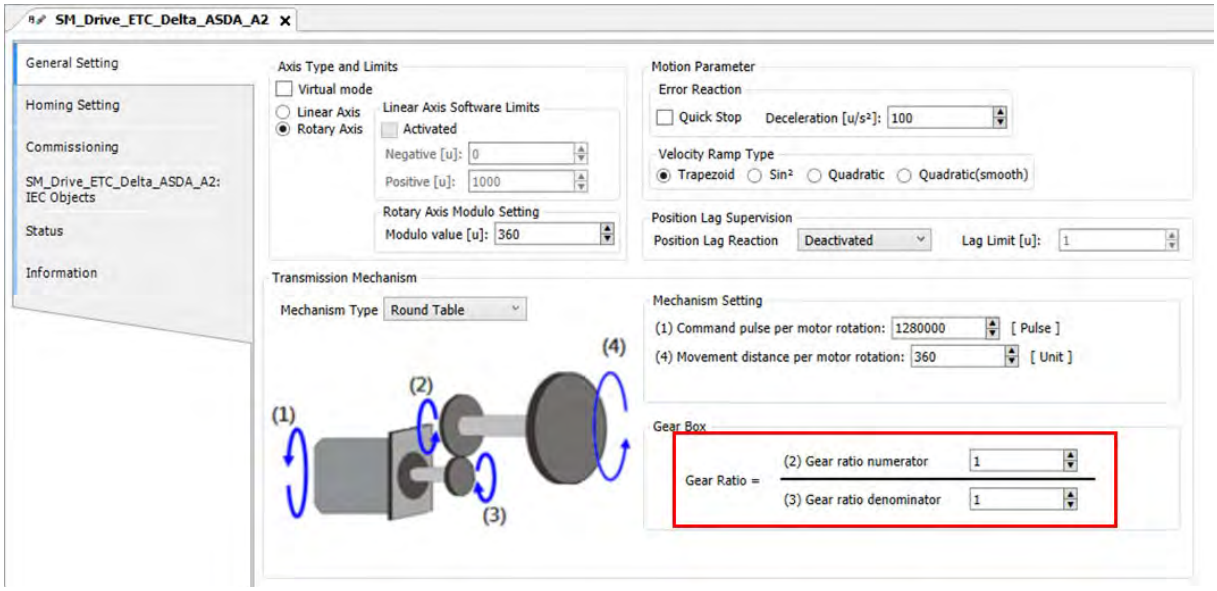
3. Choose “Round Table” for the machanism type as a reault of rotary axis type used in this example.



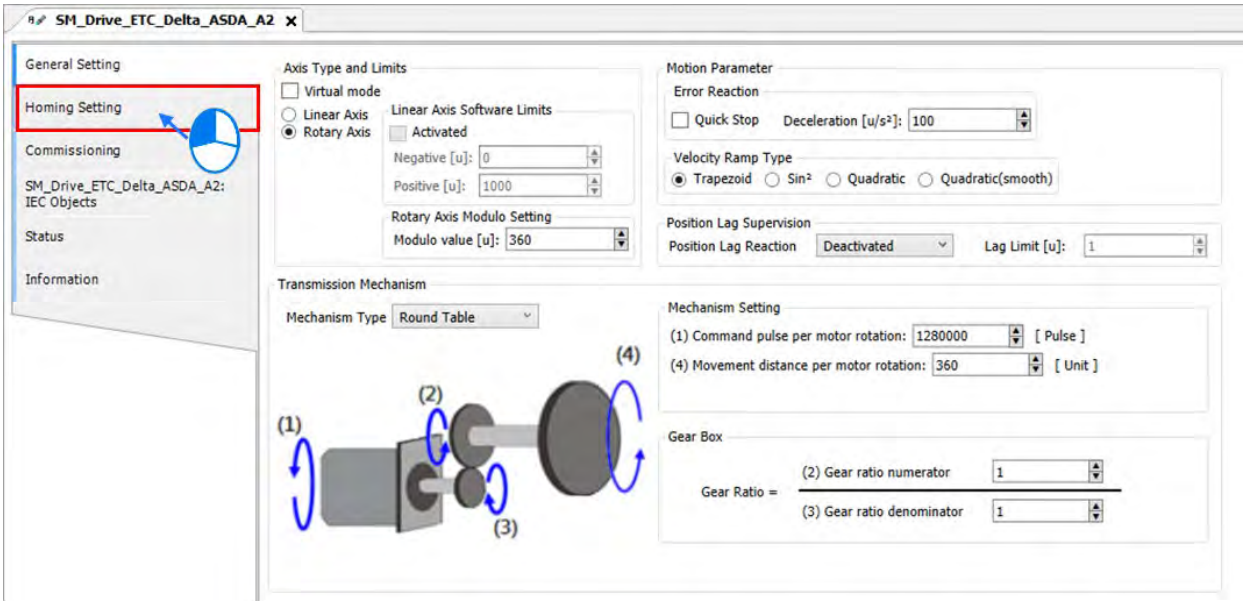
4. Continue with gear setting, set “Command pulse per motor rotation” to 1,280,000 and the user unit to 360 degrees. (Gear ratio $\frac{P1-44}{P1-45} = \frac{\text{Motor Pulse per revolution}}{\text{Command Pulse per rotation}}$)



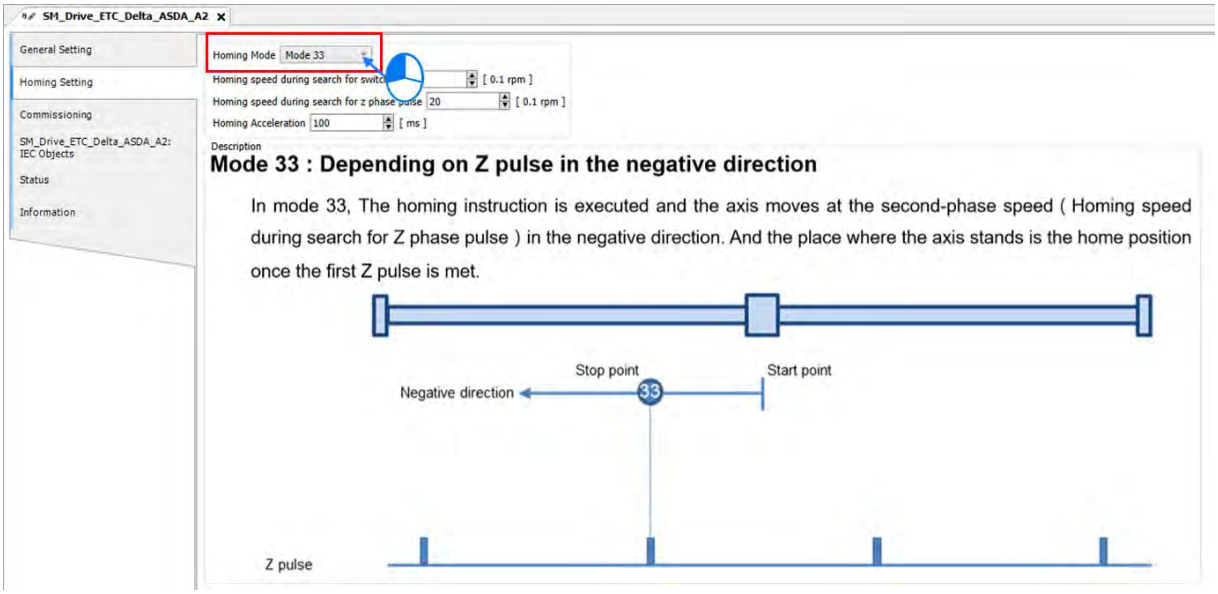
5. Since the mechanism is not equipped with a gearbox, the gear ratio for gearbox would be 1:1.



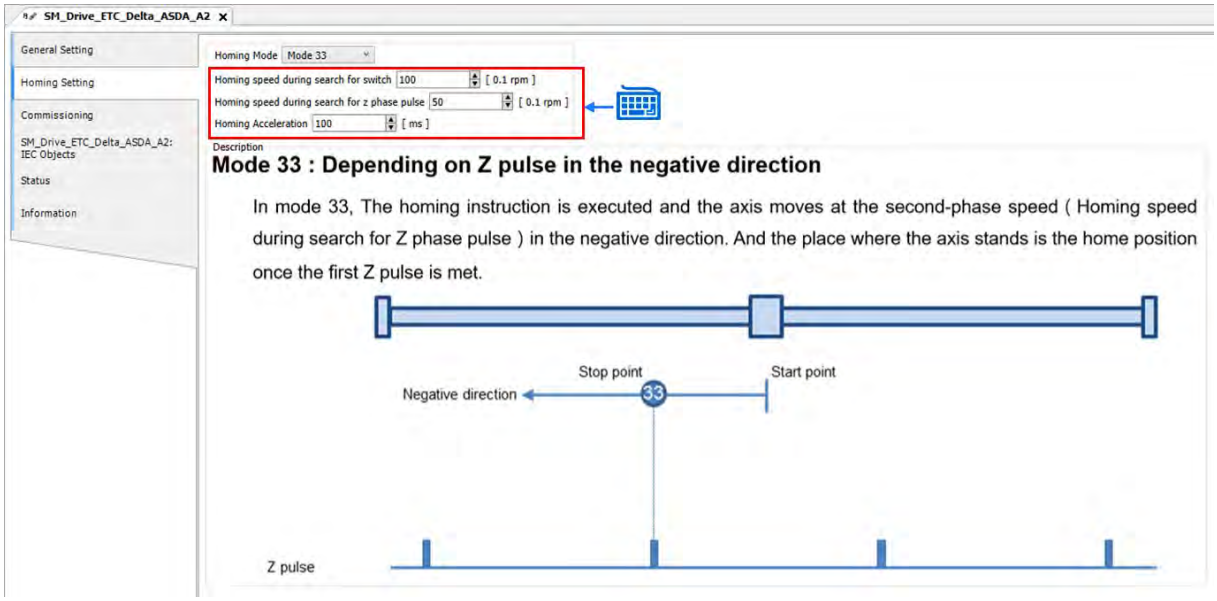
6. After finish configuring gear ratio relationship, click “Homing Setting” to enter the tab page.



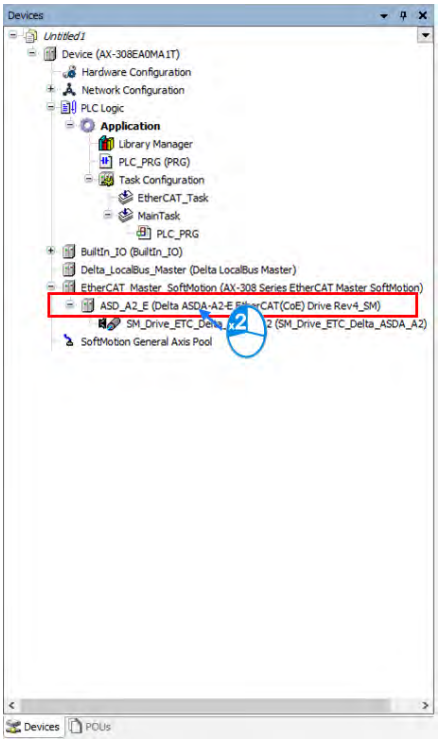
7. Since the axis type is rotary, choose Mode 33 to execute homing instruction depending on Z pulse. The descriptions of each mode in different cases are listed below.



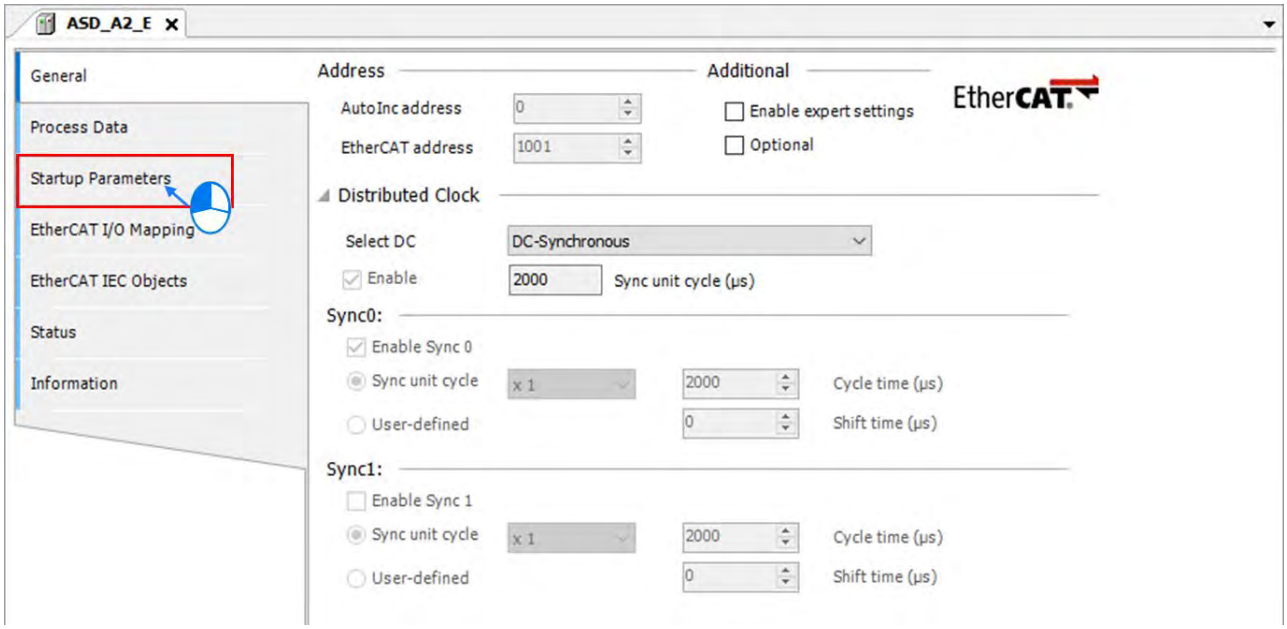
8. Then configure the proper homing speed settings, which should match with the mechanism.



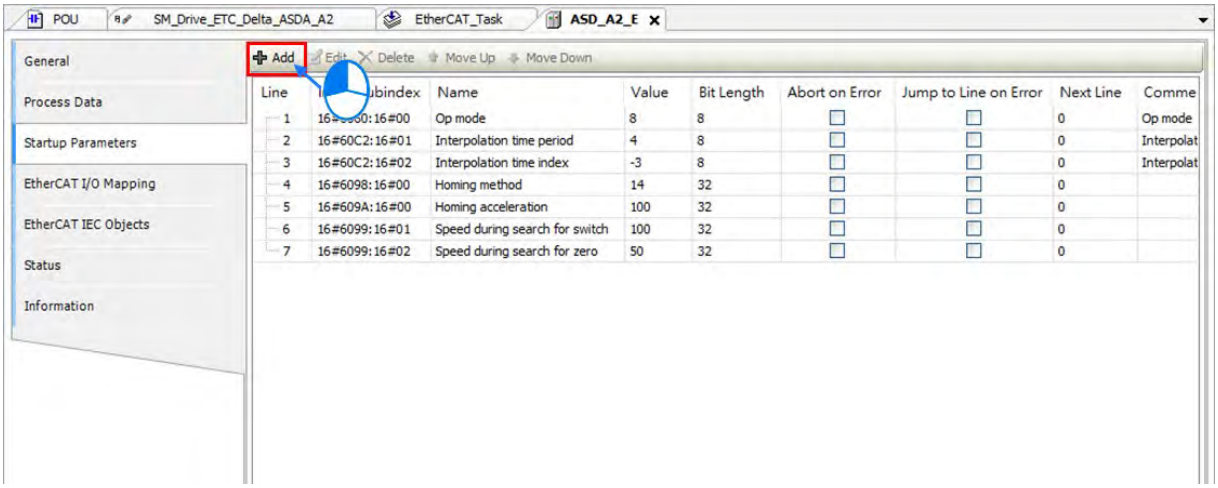
9. After finish the settings of gear ratio and homing mode, the parameters on the servo drive need to be tested for DI9 and DI10 inputs placed on the servo as a result of hardware limit. To configure the pin functions of DI9 and DI10, double click on “ASD_A2_E”.



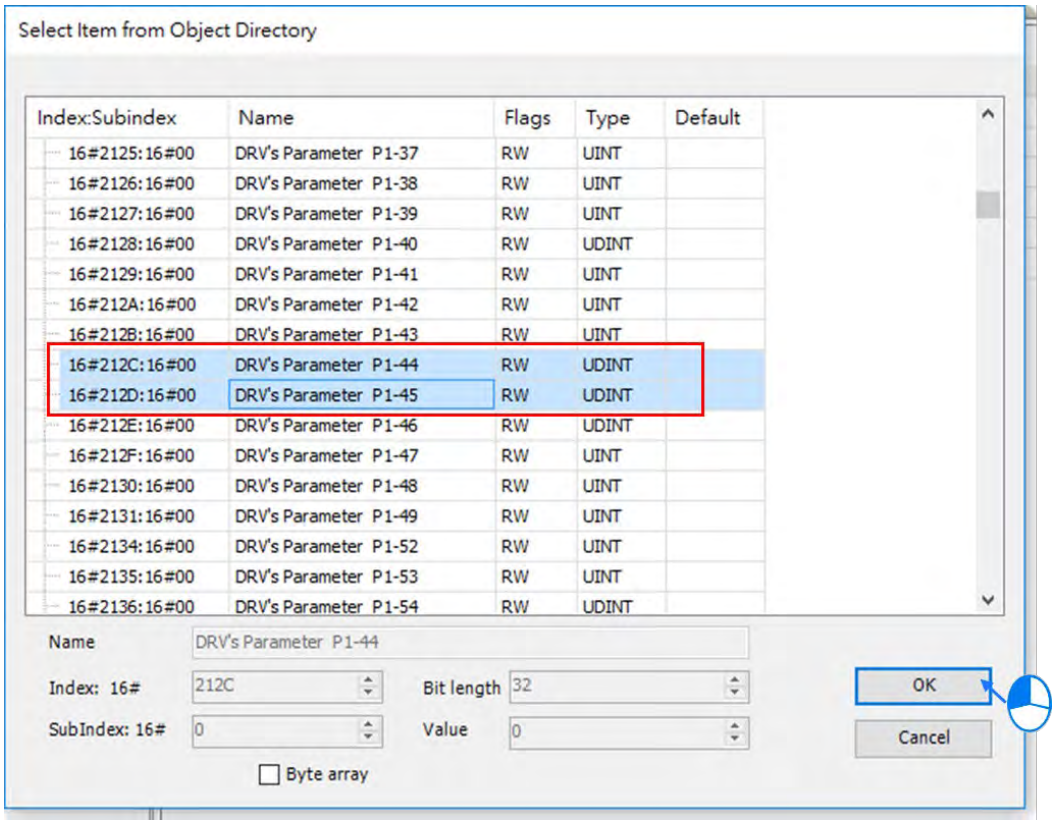
10. Click “Startup Parameters” tab on the ASD_A2_E page.



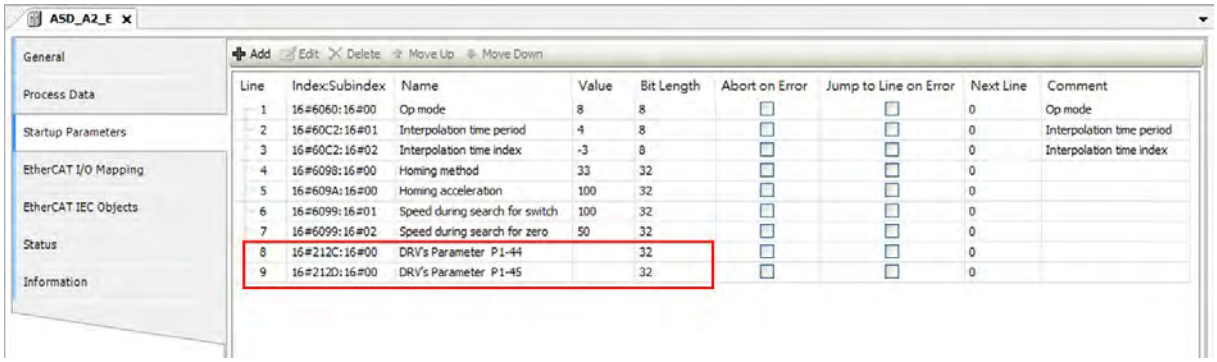
11. Click “Add” to add new parameters on “Startup parameters” page.



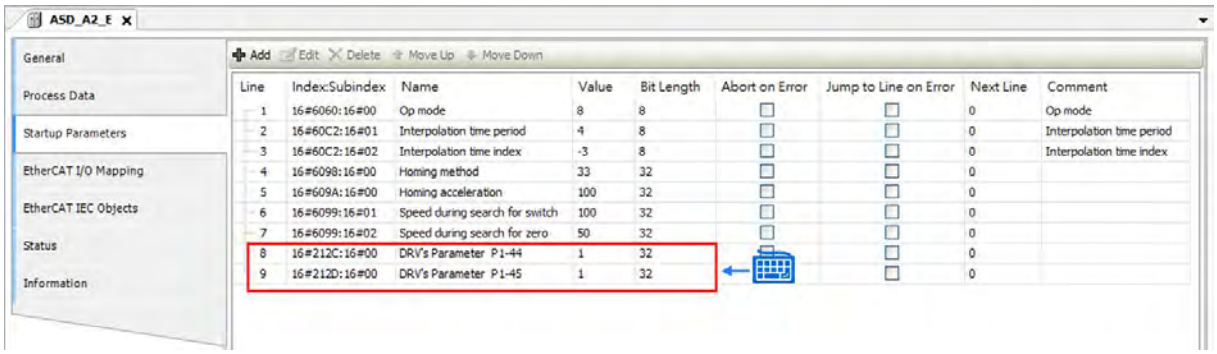
12. Select “P1-44” and hold the Shift button to select multiple parametres. Then click OK to add the selected parameters.



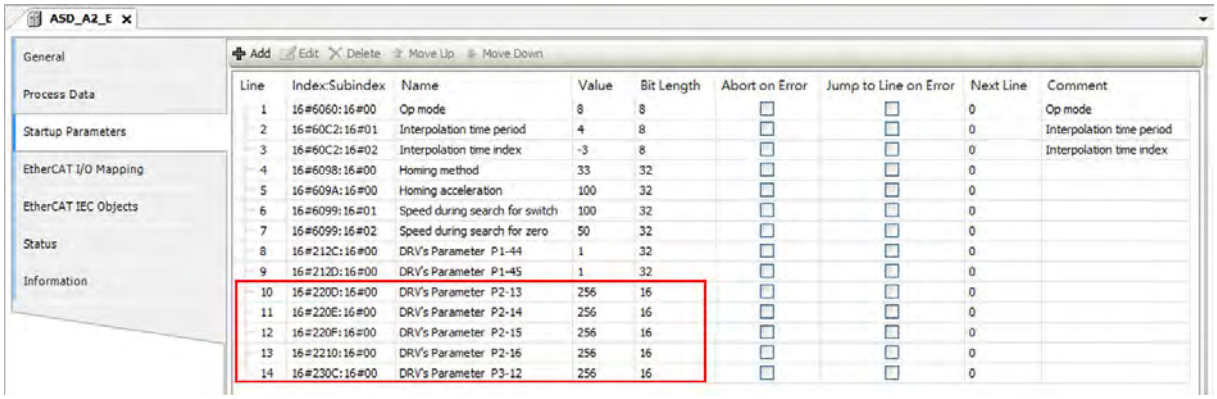
13. The selected parameters you've just added will be displayed on the list.



14. Set both values of P1-44 and P1-45 to 1, which should match the gear ratio in axis parameters. (All the setting values on this parameter list will be downloaded to the servo drive once the EtherCAT communication is established.)



15. Use the same operation to configure the following parameters, which the DI4~DI7 should be turned OFF and the retentive parameters should be configured.



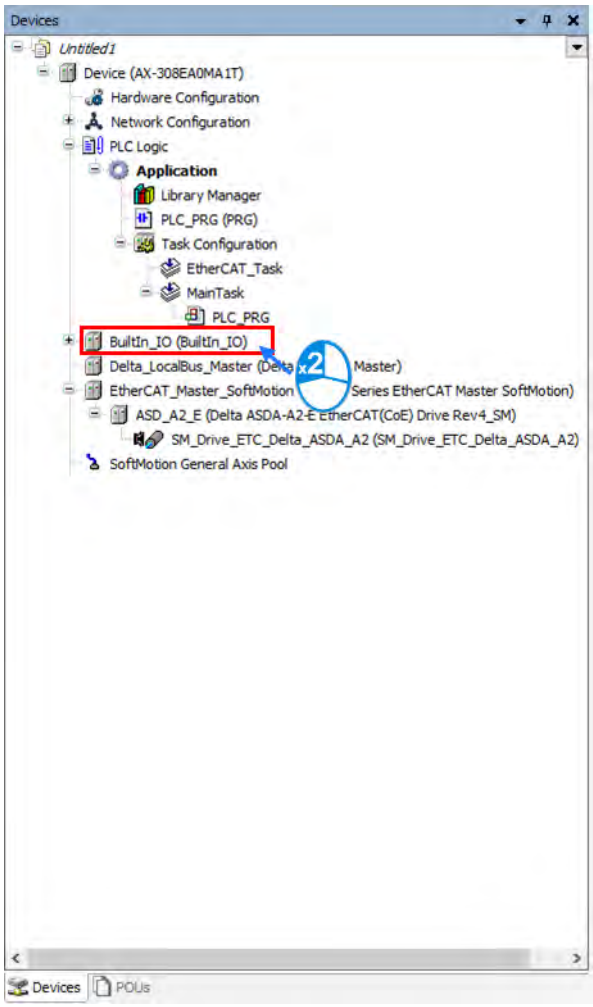
■ Servo parameter

Use the same operation to configure the following parameters, which the DI4~DI7 should be turned OFF and the retentive parameters should be configured.

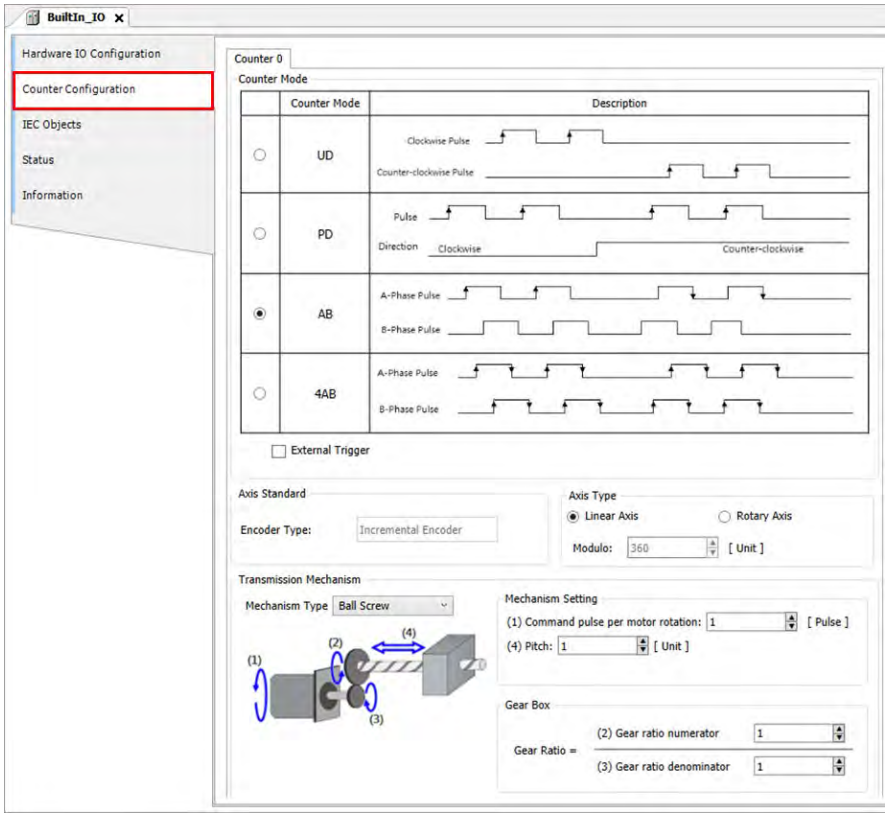
Parameter	Function	Setting Value
P1-44	Gear ratio numerator	1
P1-45	Gear ratio denominator	1
P2-13	DI4 digital input pin function	256
P2-14	DI5 digital input pin function	256
P2-15	DI6 digital input pin function	256
P2-16	DI7 digital input pin function	256
P3-12	Remain unchanged with the parameters before the drive power being cut off.	256

***Note:** Please refer to the user manuals of Delta ASDA-A2 Series for more detailed information of parameters.

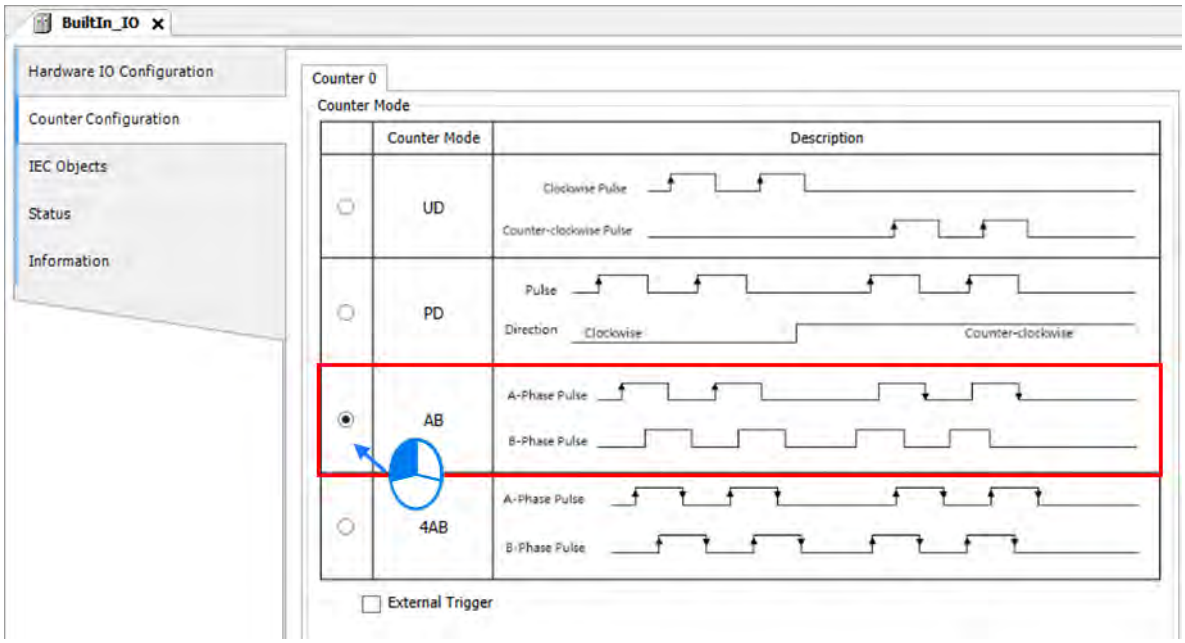
16. After the settings of real axis is completed, double click “BuiltIn_IO” to configure the encoder.



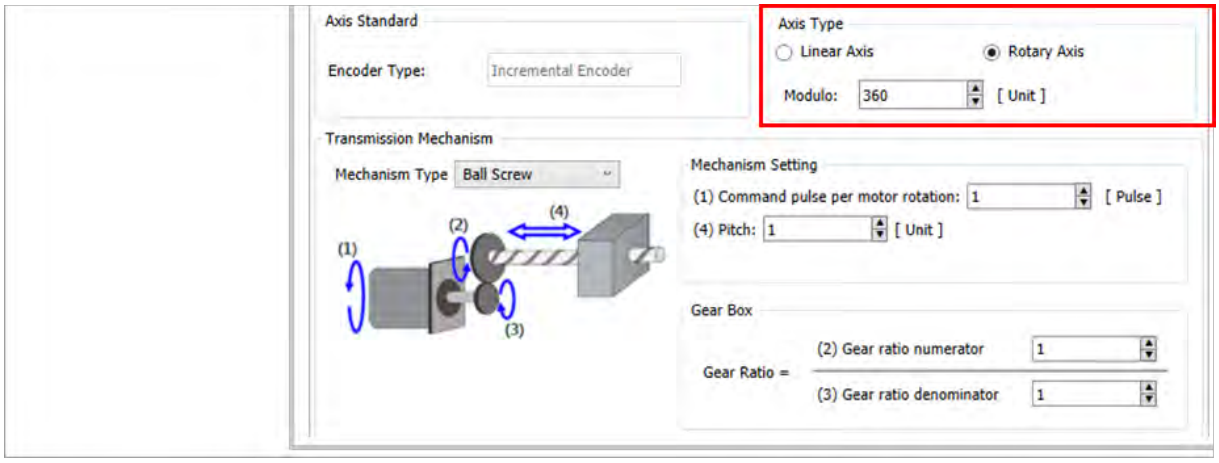
17. Click " Counter Configuration" on the BuiltIn_IO page.



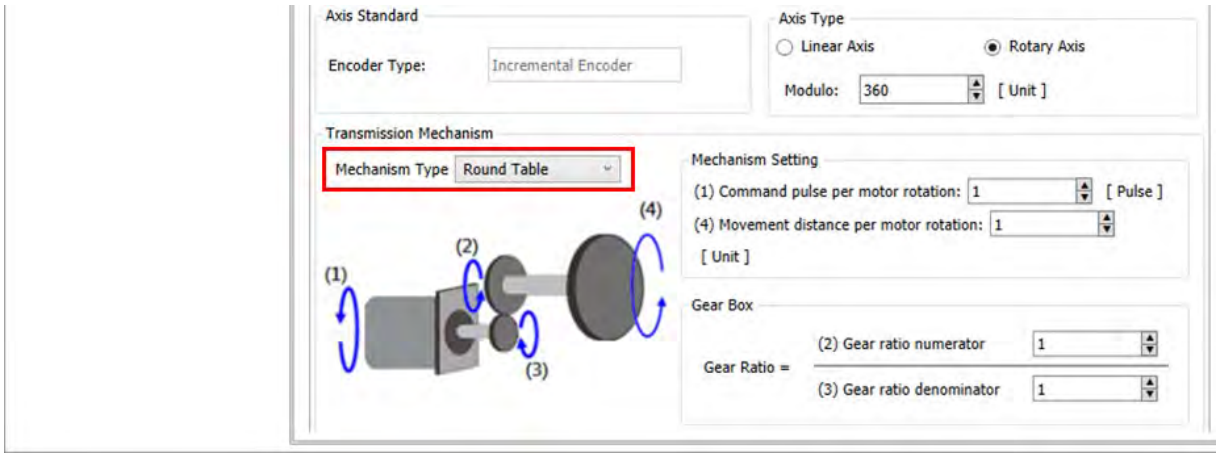
18. Select the counter mode. In this example, an encoder with AB two-phase pulse is used.



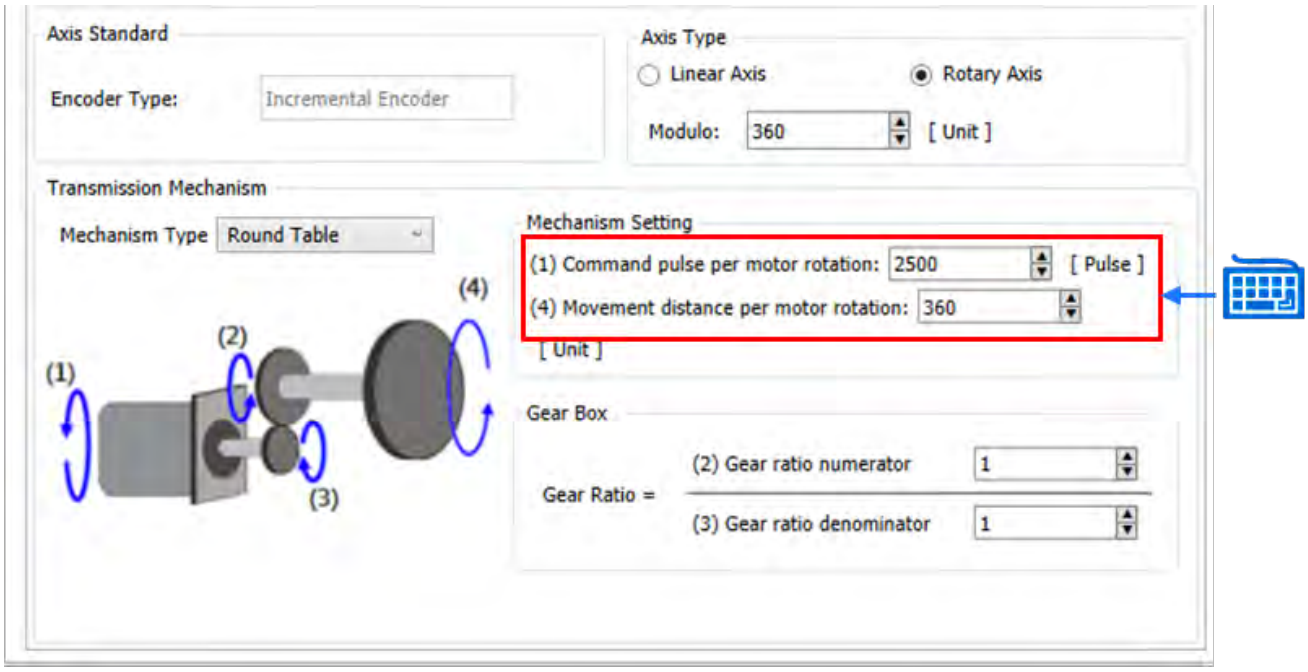
19. Choose “Rotary Axis” as the axis type, which the modulo is set to 360 degrees for a circle.



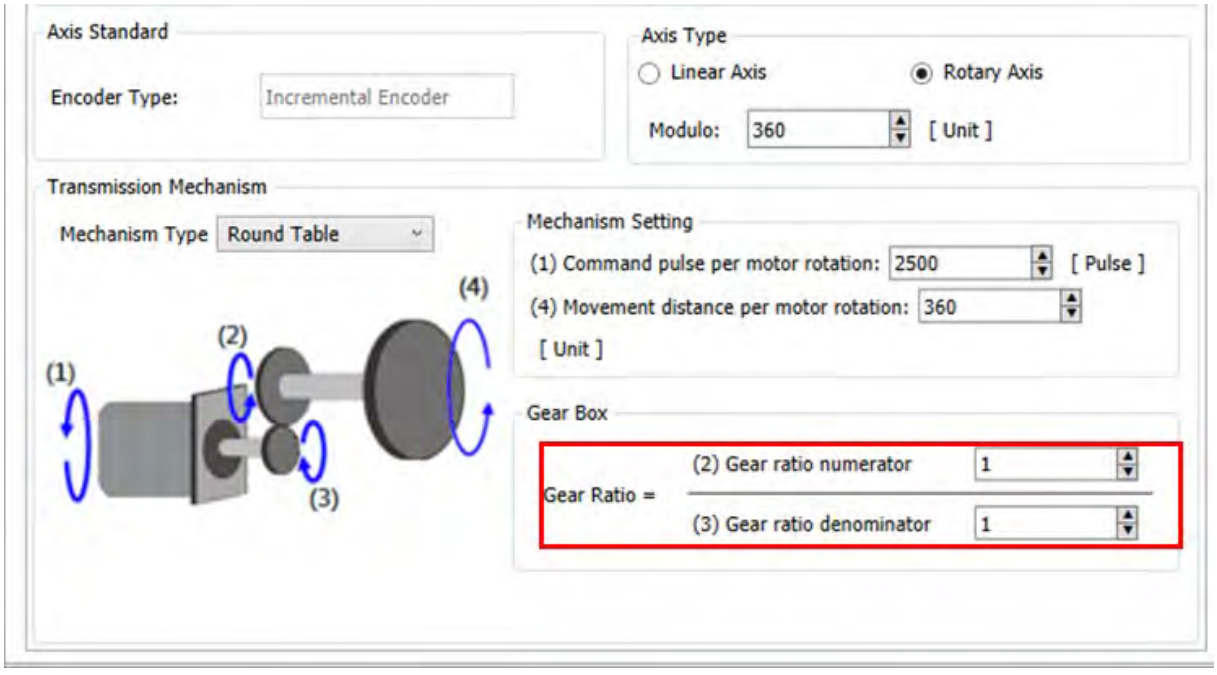
20. “Round Table” is used as the machanism type in this example.



21. Continue to configure gear ratio settings, set “Command pulse per motor rotation” to 2500, which is for per encoder rotation, and the user unit is set to 360 degrees.



22. Since the mechanism is not equipped with a gearbox, the gear ratio for gearbox would be 1:1.

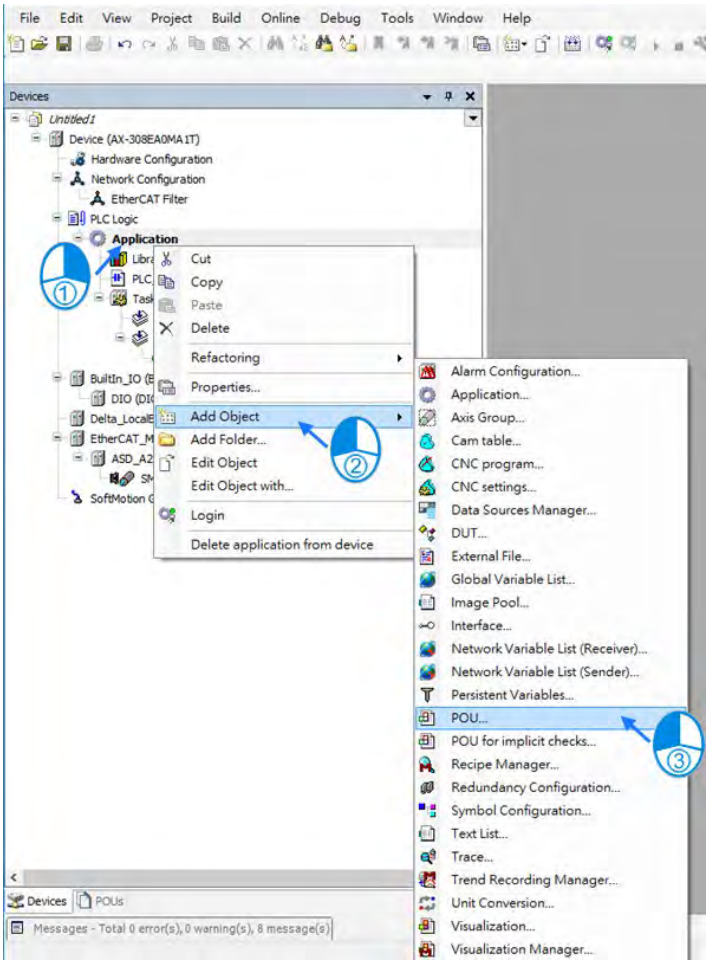


3.5.2.3 Programming for New Projects

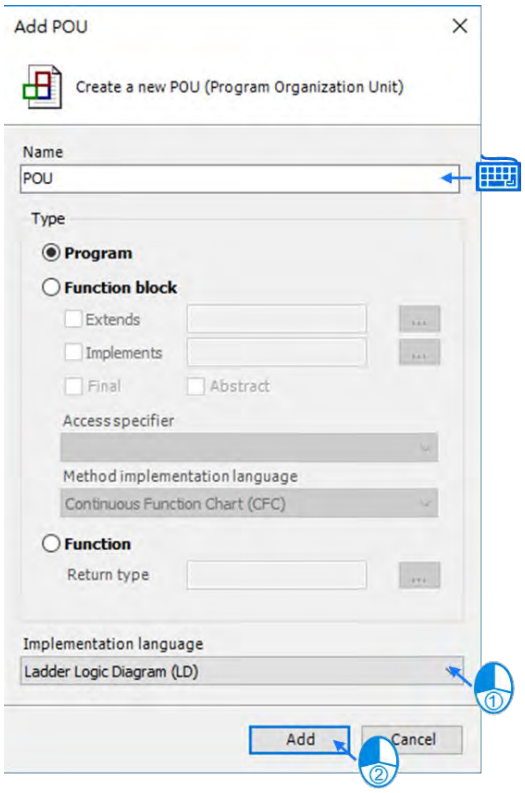
- Operating process in the example
  - "MC_Power" will be triggered and enable two servo axes (Servo ON) after executing "Step 1".
  - "MC_Home" will be triggered after executing "Step 2" to command the real axis back to the home position. While "MC_SetPosition" is required for returning the virtual axis to zero position.
  - Execute "Step 3" to trigger "MC_Jog" and the virtual axis starts to move. At the same time, "MC_GernIn" would be executed and the slave axis starts to follow. Once "Step 3" is aborted, "MC_Jog" will be stopped and "MC_GearOut" will be triggered. After the Done output of MC_GearOut becomes TRUE, "MC_Stop" is triggered to stop the slave axis. The whole process will be performed repeatly till the contact singnal of "Step3" is OFF.
  - The value of Counter_1 will be increased by 1 for each time the operation being executed automatically.

- Programming with newly-created POUs

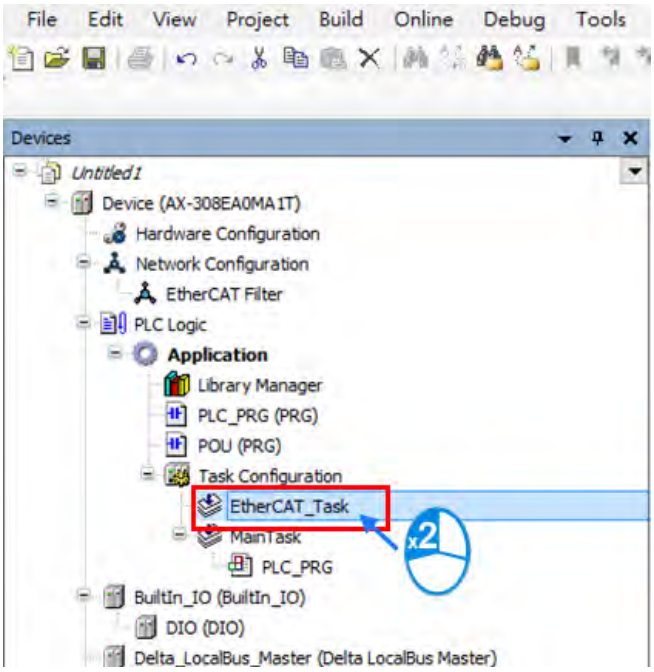
1. First, create a new POU by right-clicking "Application" to choose "Add Object" and select "POU".



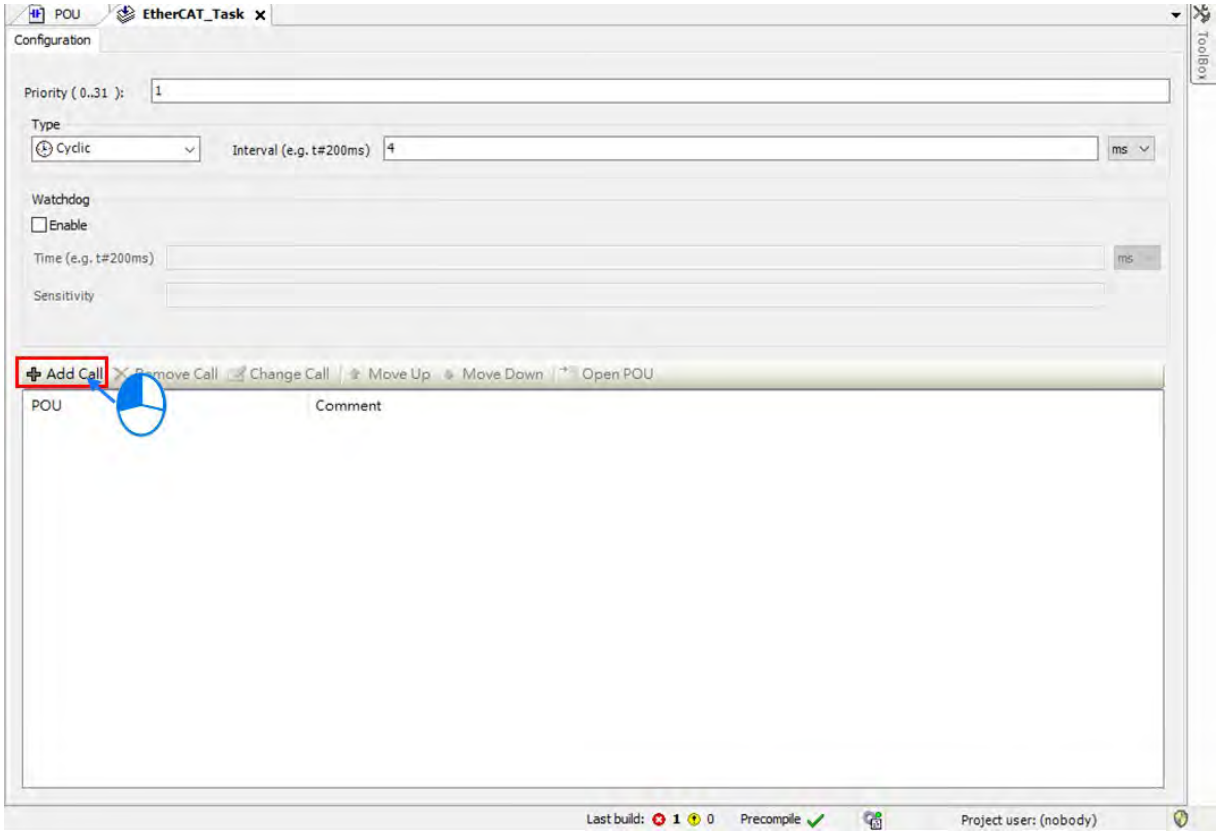
2. Input the name of the new POU in the “Name” field and choose LD language for “Implementation language”. When finished, click “Add” to add the new POU.



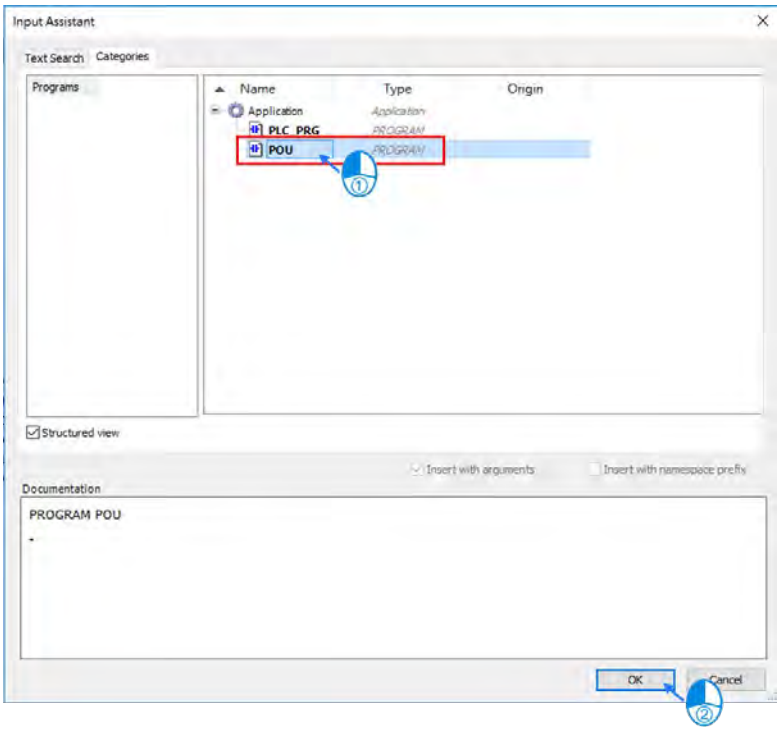
3. The newly added POU would be shown in the project tree, which needs to be added in Task by double-clikcing “EtherCAT_Task”. (Function blocks related to axis motion needs to be established in EtherCAT_Task to ensure normal operation of axes.)



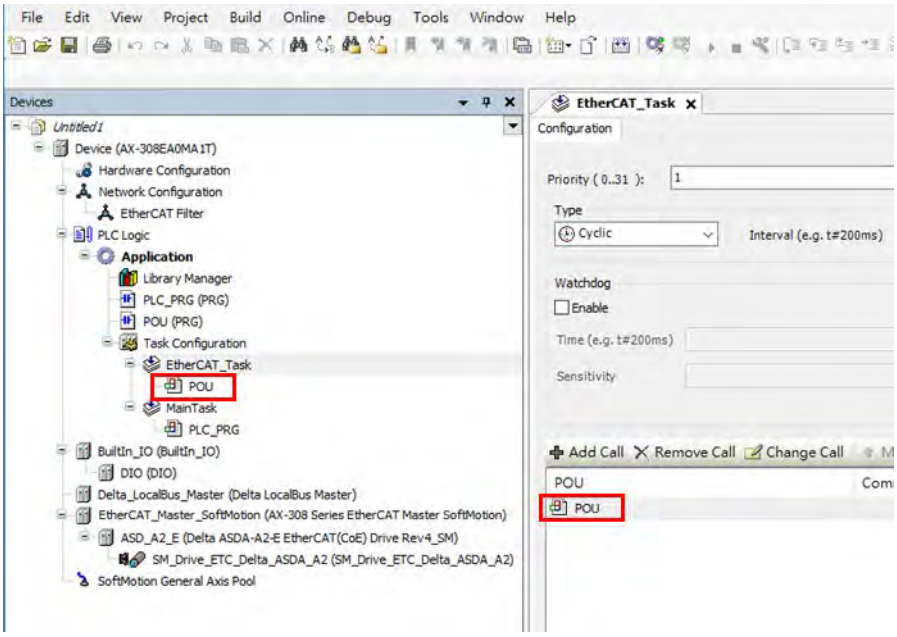
4. Click “Add Call” on the EtherCAT_Task page.



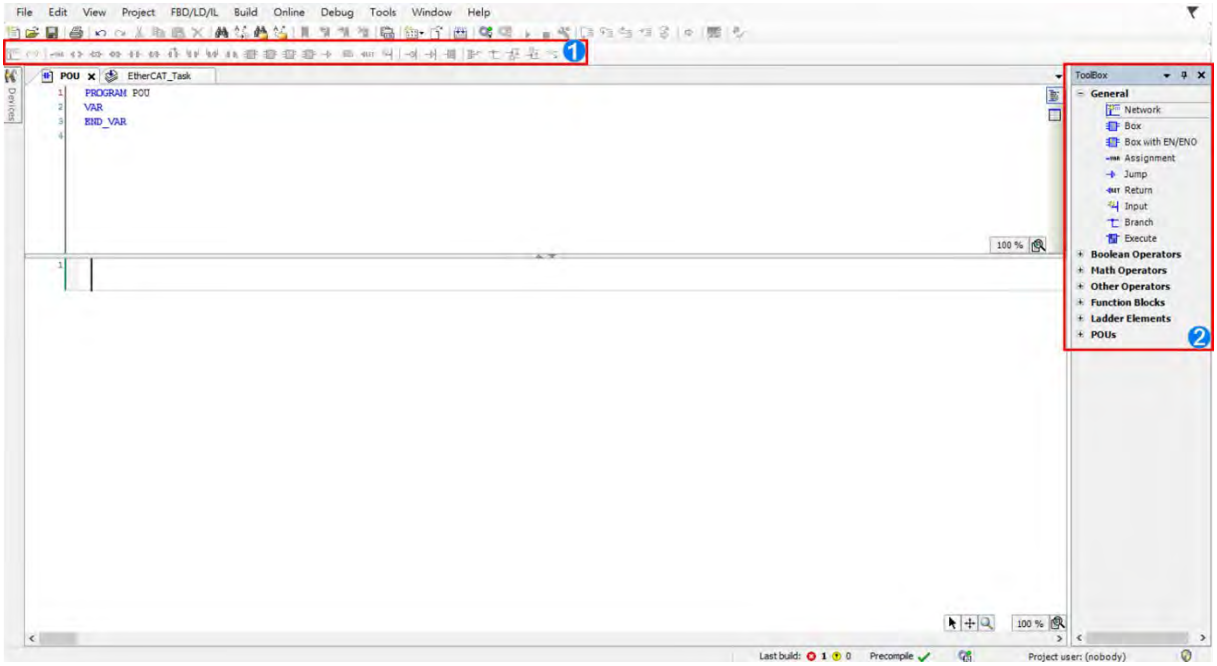
5. Click “OK” after choosing the newly-created POU.



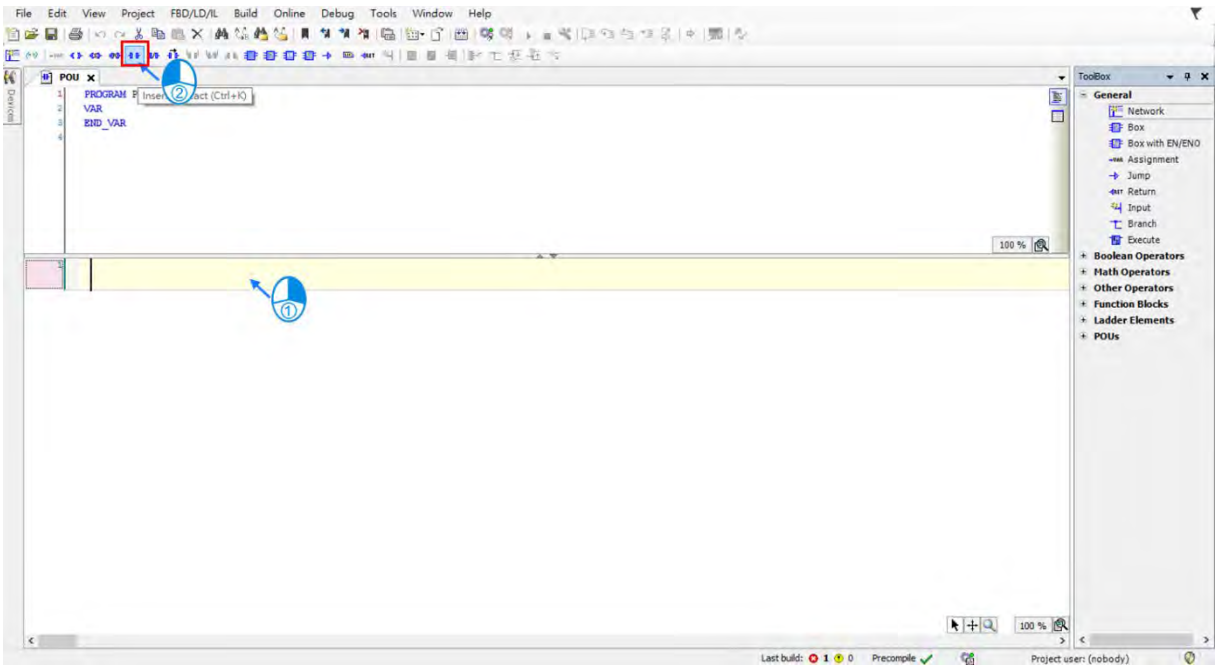
6. After adding the POU in EtherCAT_Task, “POU” will be shown on the lower half of the EtherCAT_Task page as well as in the project tree. Then click the POU from the tree, which is under the category of EtherCAT_Task.



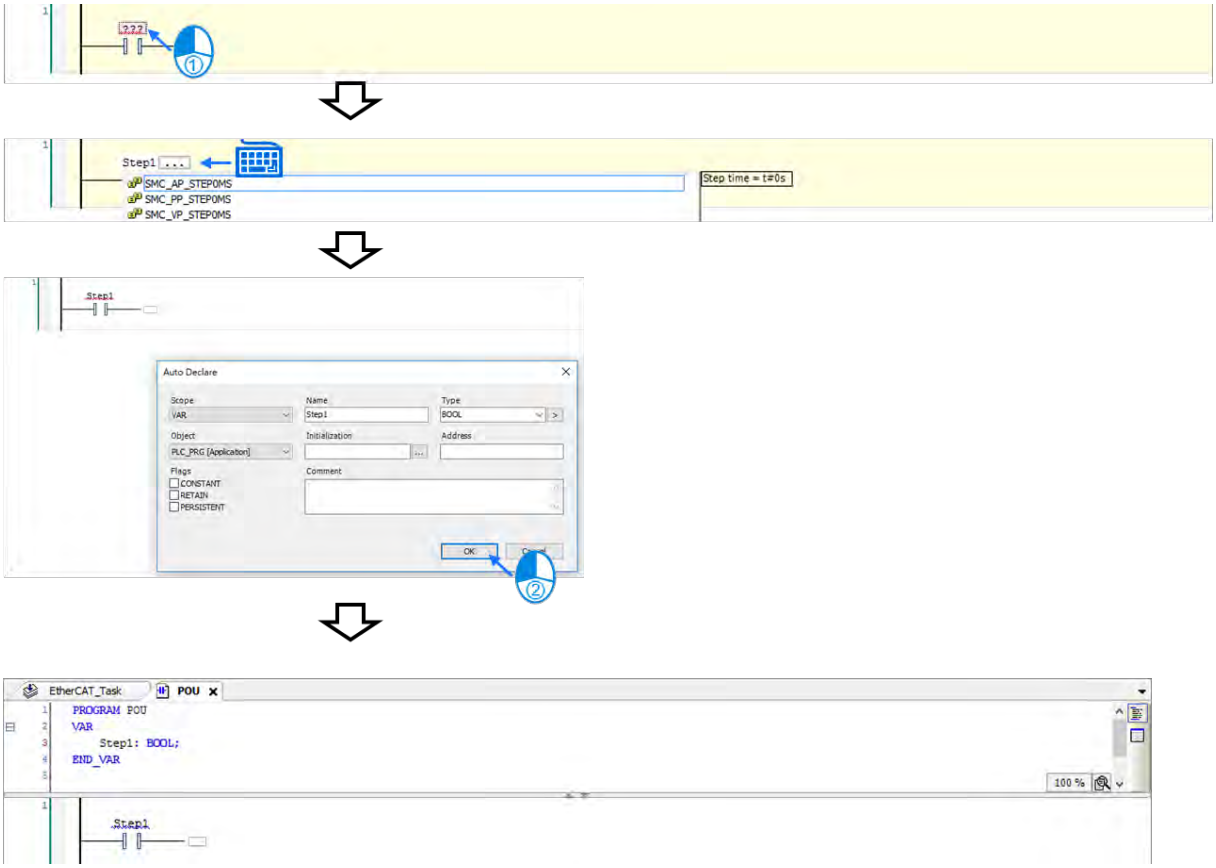
7. With ladder logic programming language, add new commands by using the red-circled field marked with ①, while field ② is for adding required Function or Function Block.



8. Click on the desired place to add new commands on POU page, then select  .

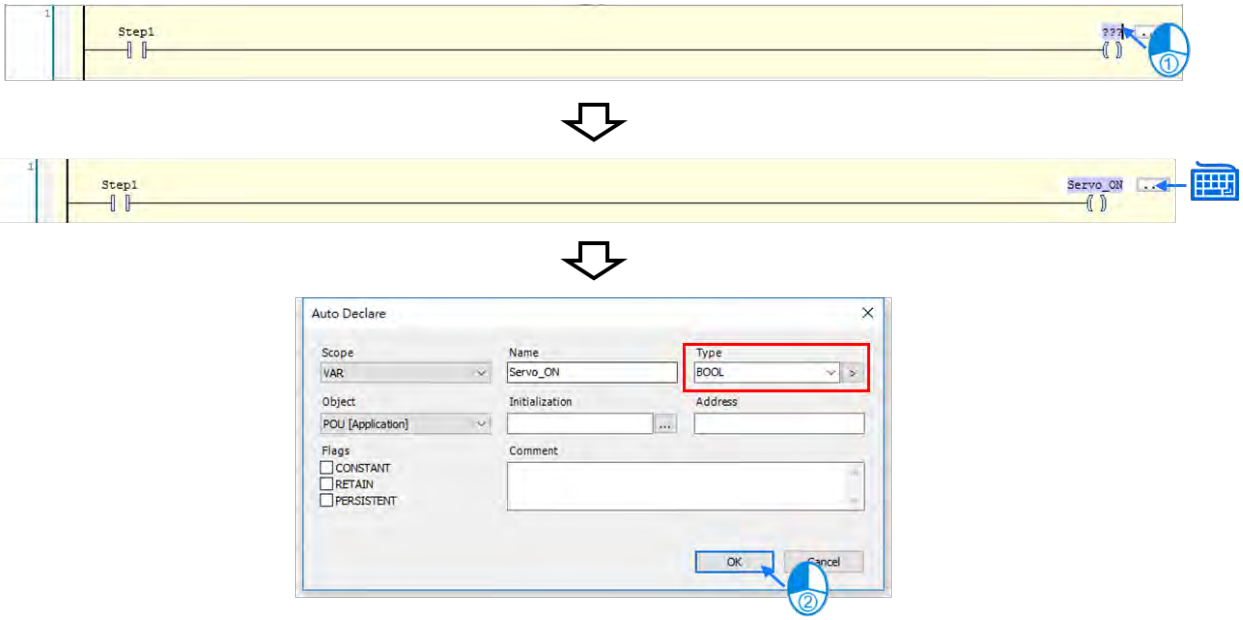


9. After the selected command is shown in the POU, click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.



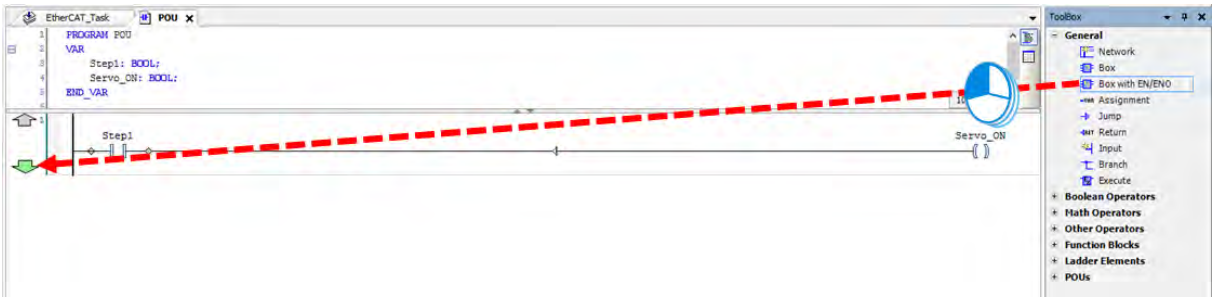
10. Then add the ouput instruction by clicking on the desired place and choosing  .

11. Click “???” to enter the name of instruction variable then press ENTER key to continue with the “Auto Declare” page. Click “OK” after checking whether “Type” is correct.

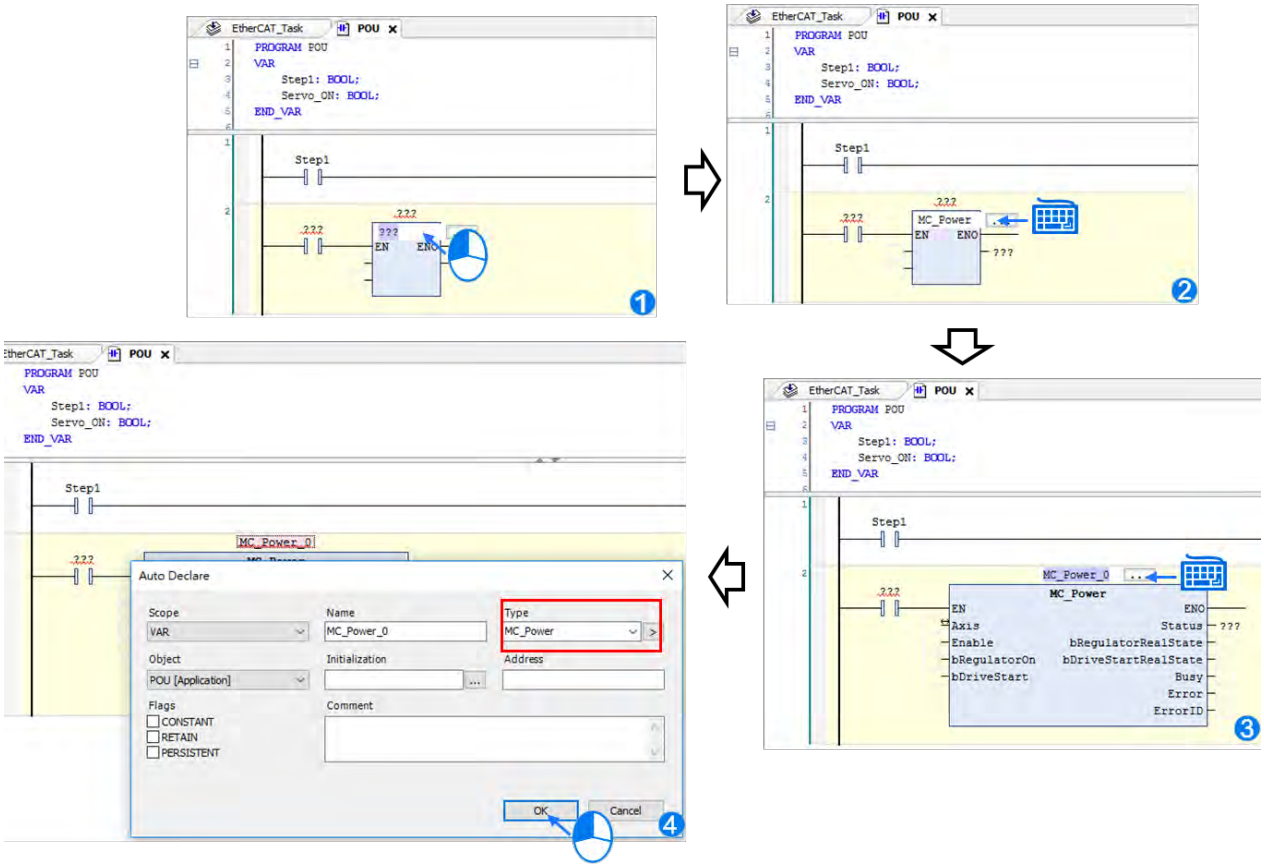


12. The instruction names you have added previously will be displayed as well as the instruction types, where the name and type of the instruction can also be added directly. (Same operation for the examples in later discussion.)

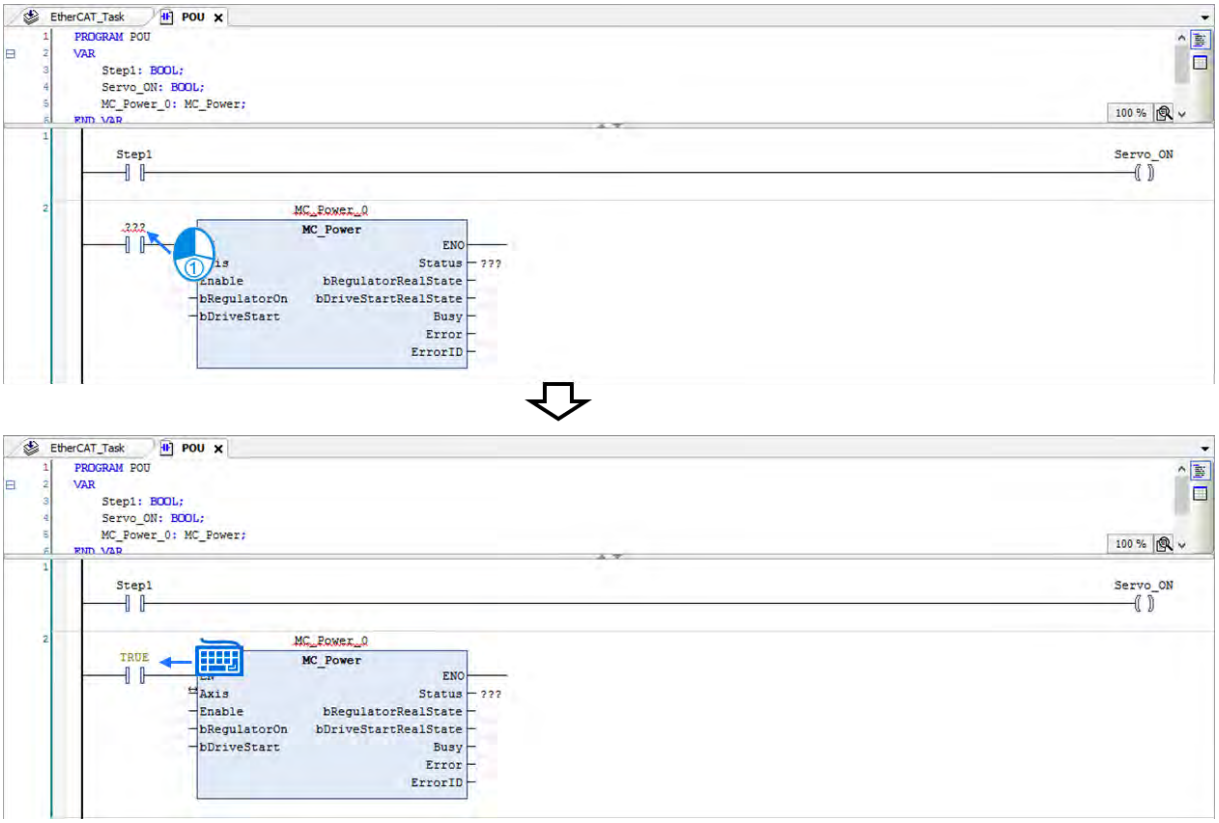
13. To add new function blocks, click and hold “Box With EN / ENO” then drag to  .



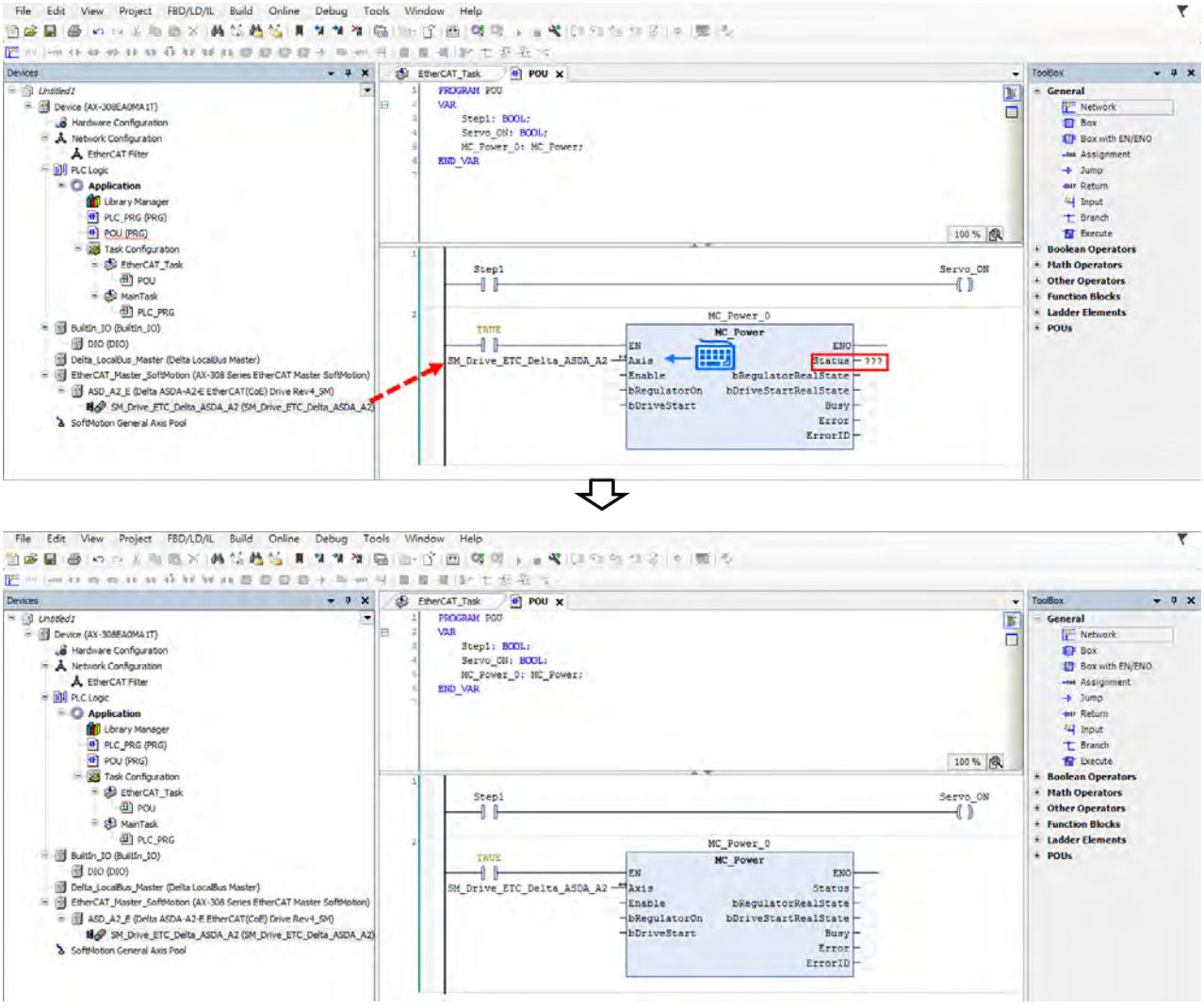
14. Click “???” and insert the name of function block then press ENTER to display the function block. Insert the name and press ENTER key again, then “Auto Declare” window will pop up. Click “OK” after confirming the “Type” of function block is correct.



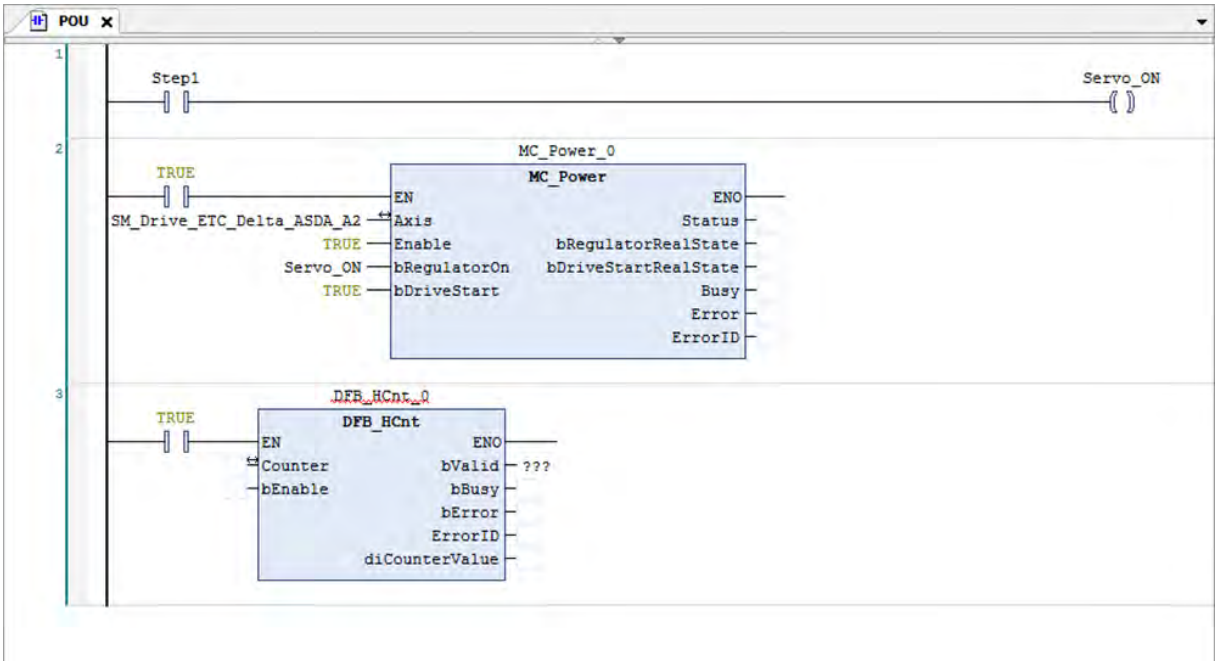
15. Continue to click “???” and insert “TRUE”, then press the ENTER key.



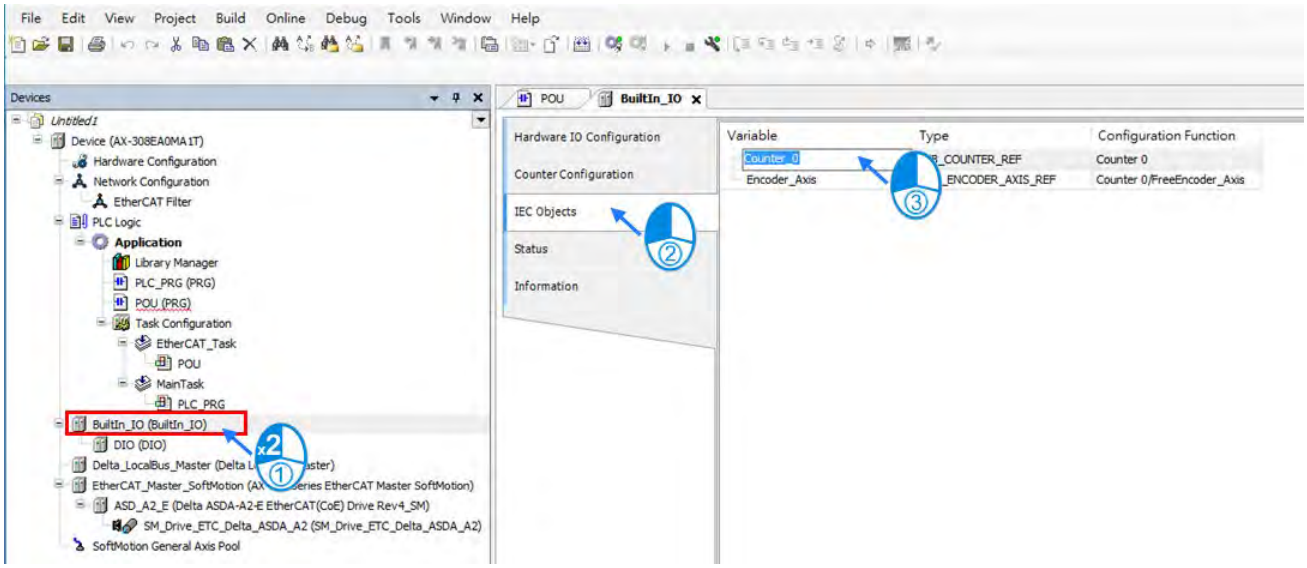
16. Insert the name of servo drive to the Axis input pin of the function block and delete “???” of other pins, since the mark “???” does not represent any variables and errors may occur while parsing.



17. Use the same operation to create “DFB_HCnt” function block after “MC_Power” has been created.

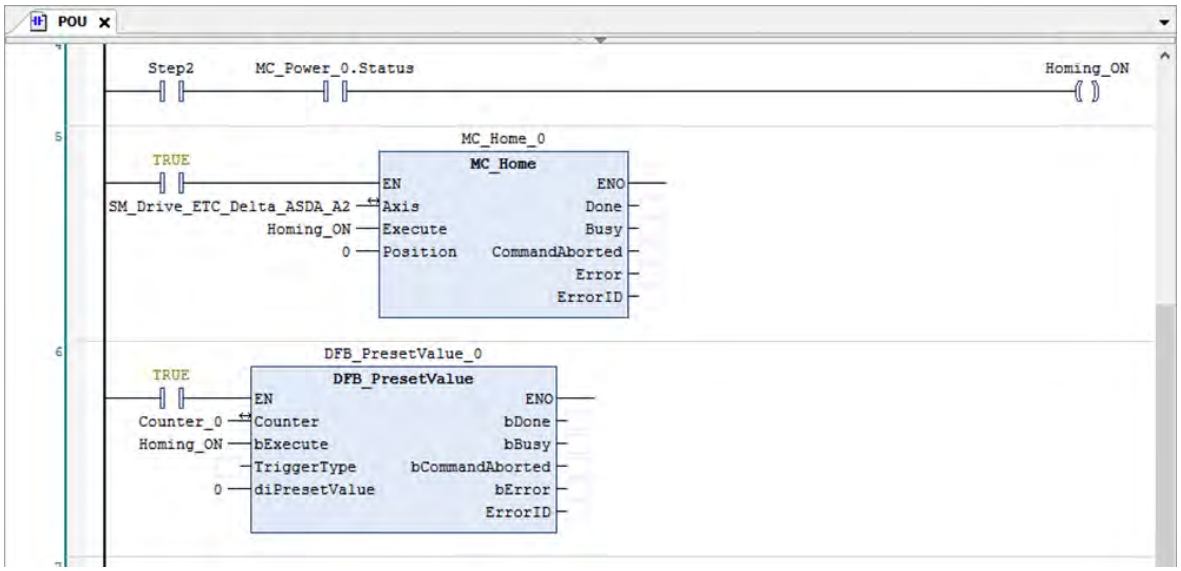


18. Navigate to “BuiltIn_IO” > “IEC Objects” > “Variable” and enter the encoder’s name “Counter_0”. The name of each newly-added encoder axis will be shown here.

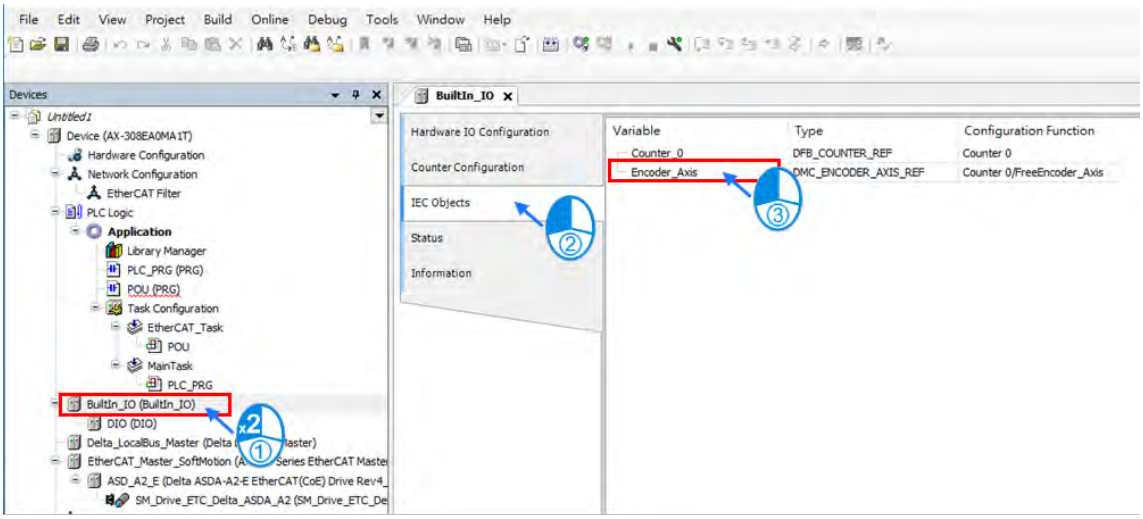


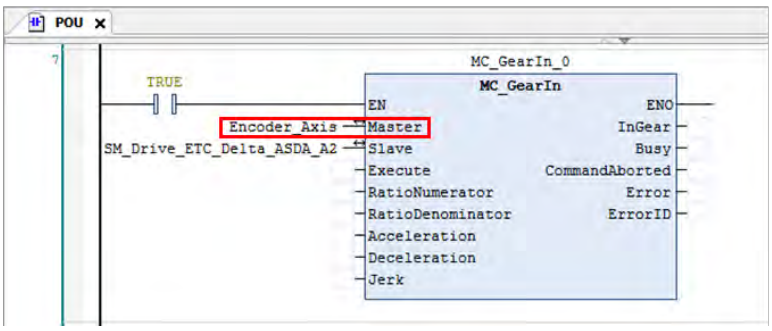
19. Insert the name of encoder to the Counter input pin of the “DFB_HCnt” function block and delete “???” of other pins, since the mark “???” does not represent any variables and errors may occur while parsing. (To enable the counter function, the function block “DFB_HCnt” is required.)

20. Add “MC_Home” function block to make real axis performs homing, while “DFB_PresetValue” is required for encoder to return the counter value to zero.

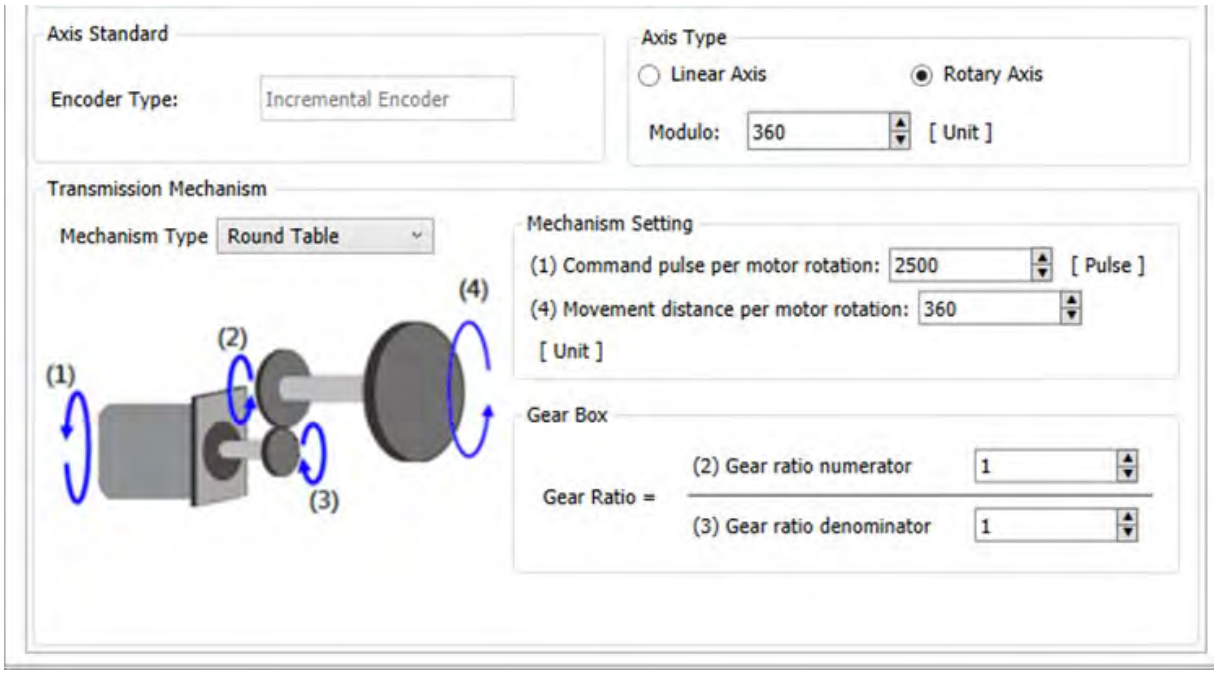


21. When the encoder axis is taken as the master, insert “Encoder_Axis” to the Master input pin of “MC_GearIn”.

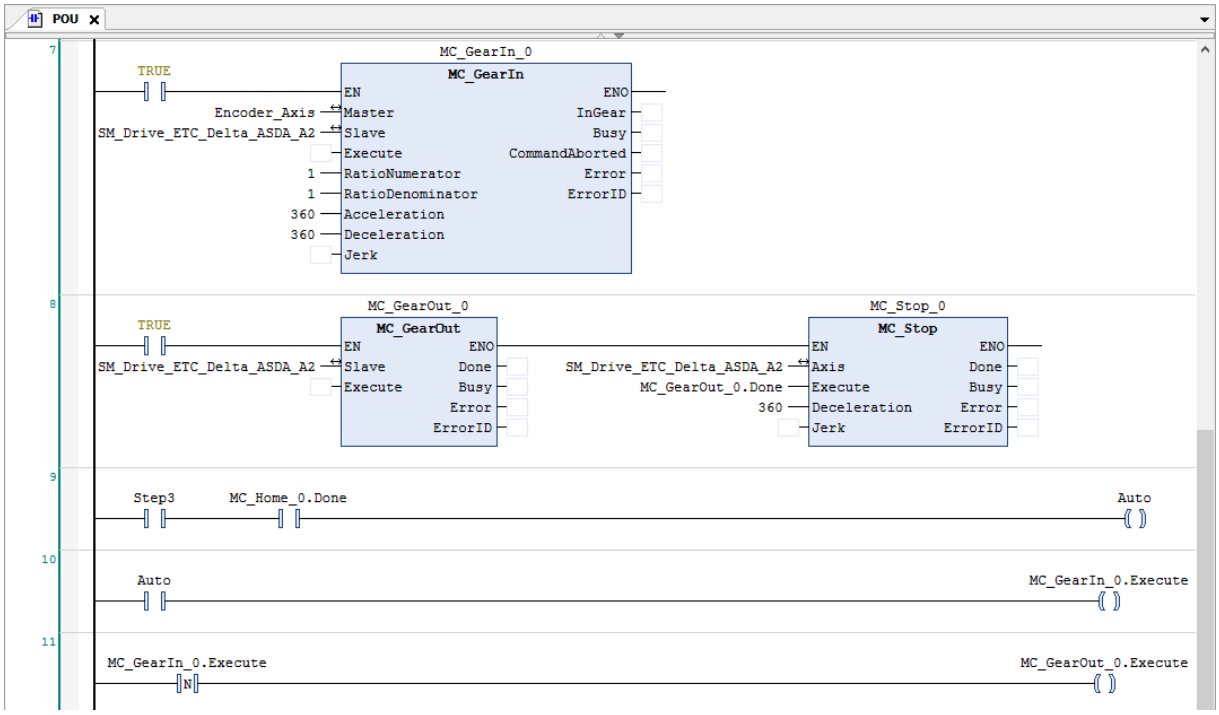




22. The counter value of DFB_HCnt is converted between unit and pulse counts, according to the parameters set for Encoder_Axis as shown below.



23. Use MC_GearIn to make the real axis follow the encoder. MC_GearOut is executed when ‘Step3” being turned off. Then the slave axis moves at a constant speed after being decoupled. Finally, use the function block “MC_Stop” to command a motion stop to the axis.

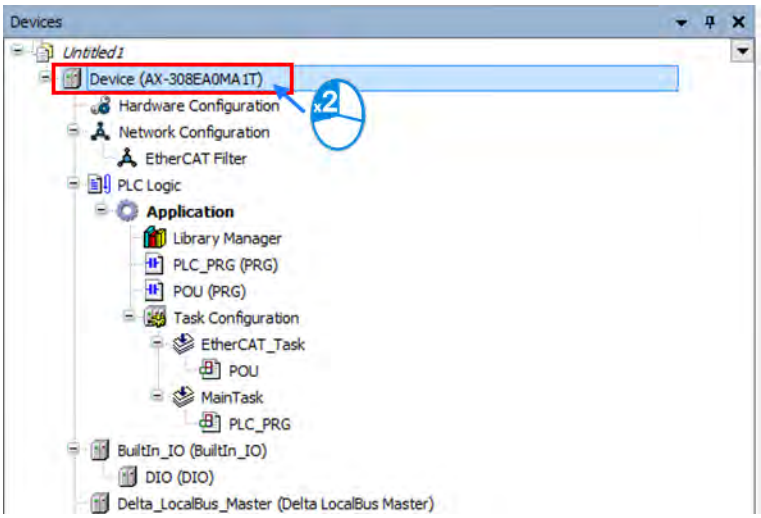


3.5.3 Program Monitoring

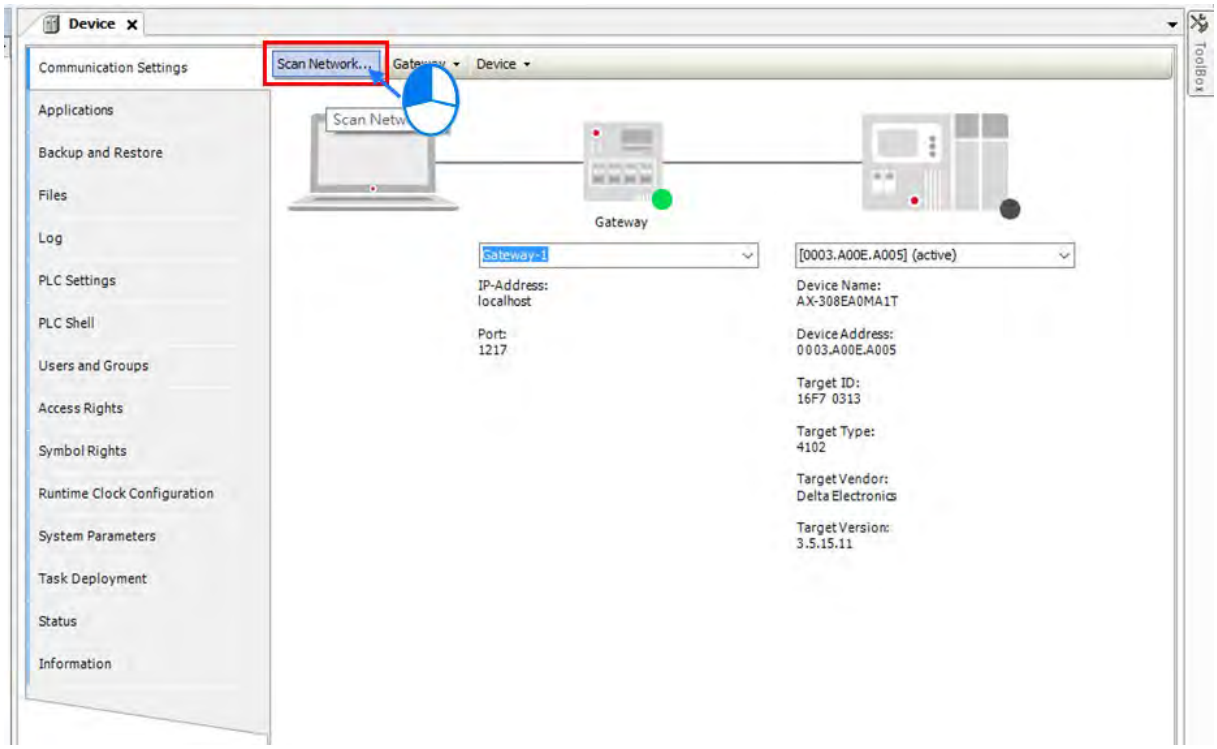
When running a program, the current control status of system can be monitored and part of the device values are allowed to be modified for system testing as well.

3.5.3.1 Setup Connection between Devices

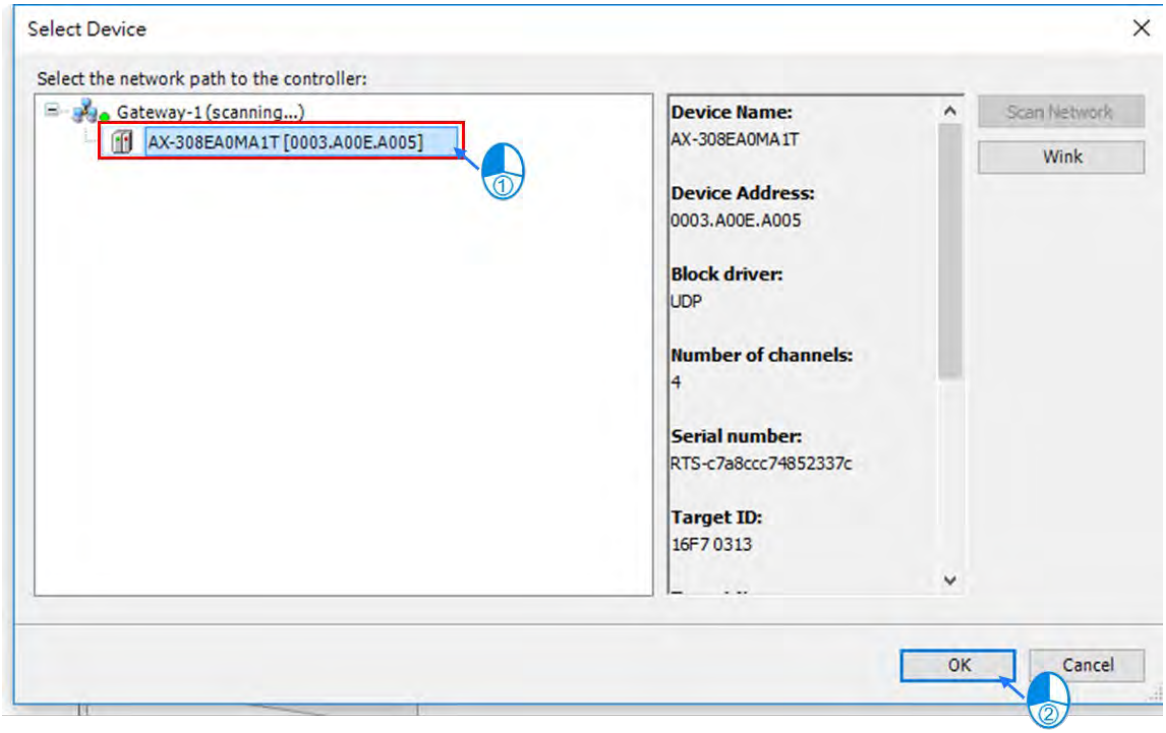
1. Double click on “Device”. (The default IP address for AX-308E is 192.168.1.5)



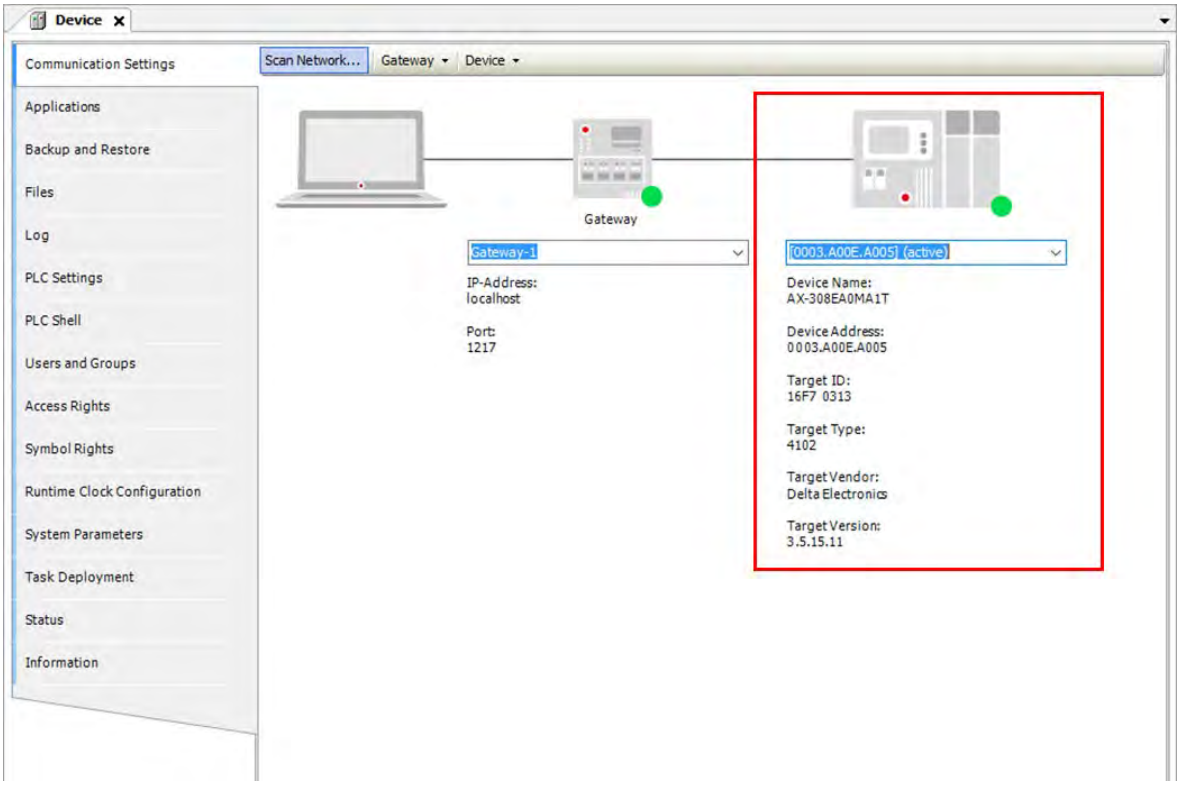
2. Choose “Scan Network” on the Device page.



3. After the “Select Device” window pops up, select the “AX-308E” device and click on “OK”.

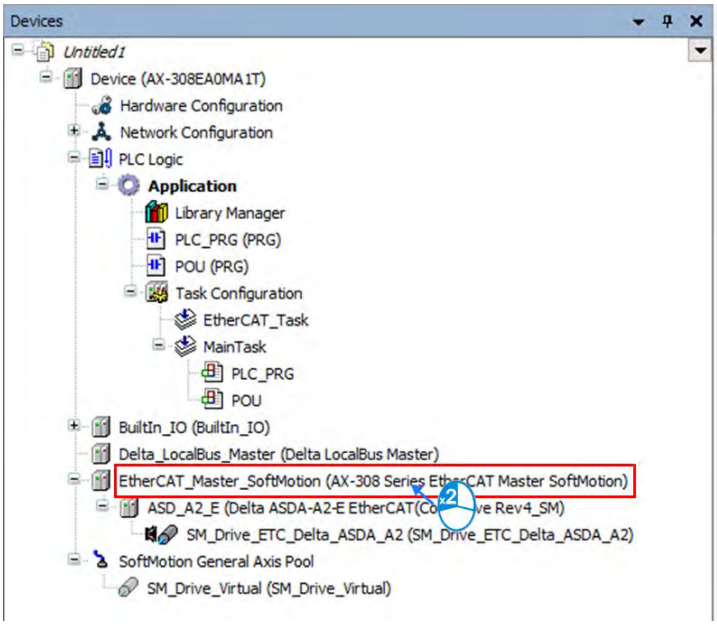


4. When the device is connected with your PC, information of the device will be displayed as shown in the following figure marked by the red box.

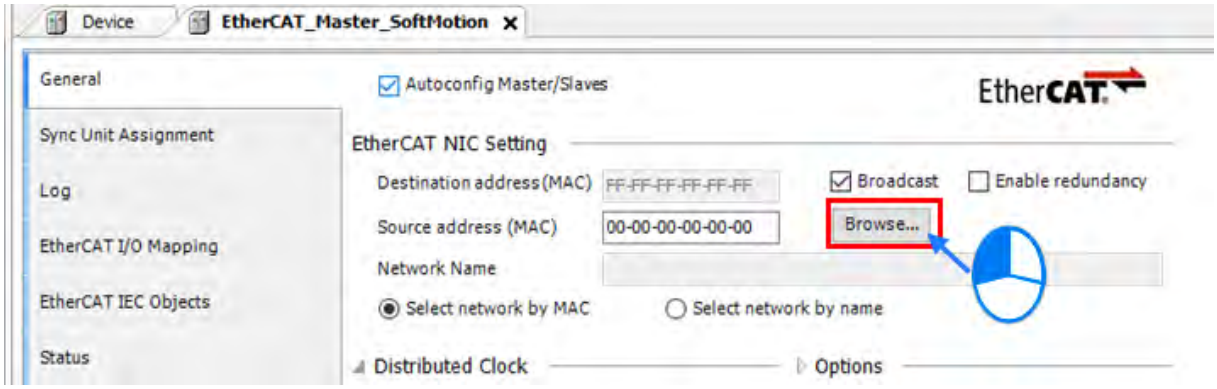


3.5.3.2 Process Monitoring and Control

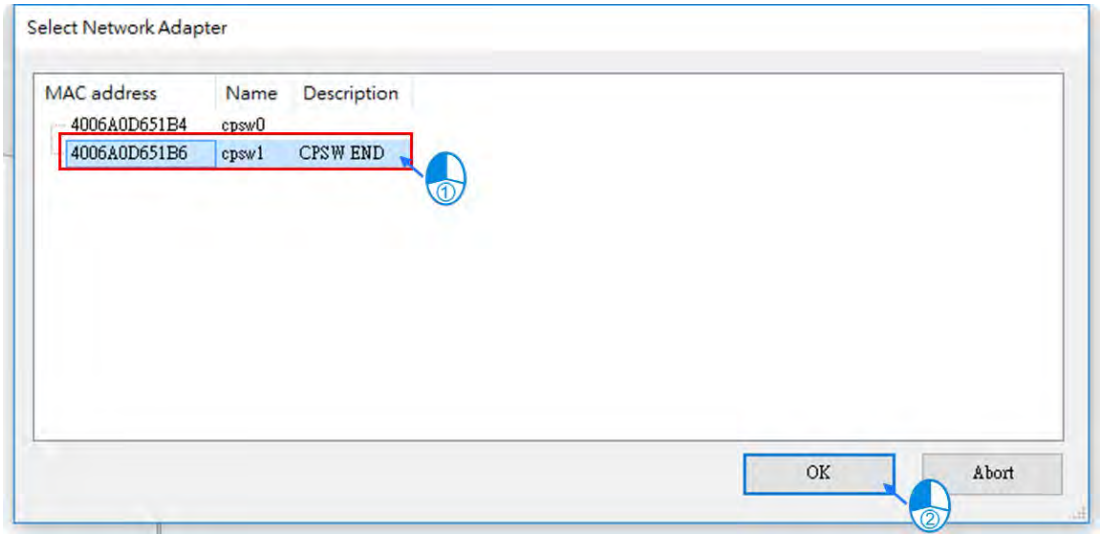
1. The correctness of EtherCAT communication should be checked before program monitoring starts. After confirmation, double click on “EtherCAT_Master_SoftMotion”.



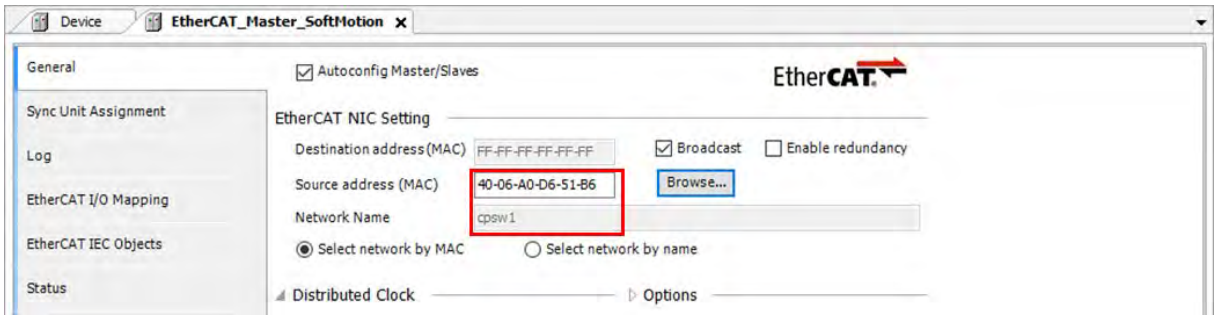
2. Choose “Browse” after entering the EtherCAT_Master_SoftMotion page.



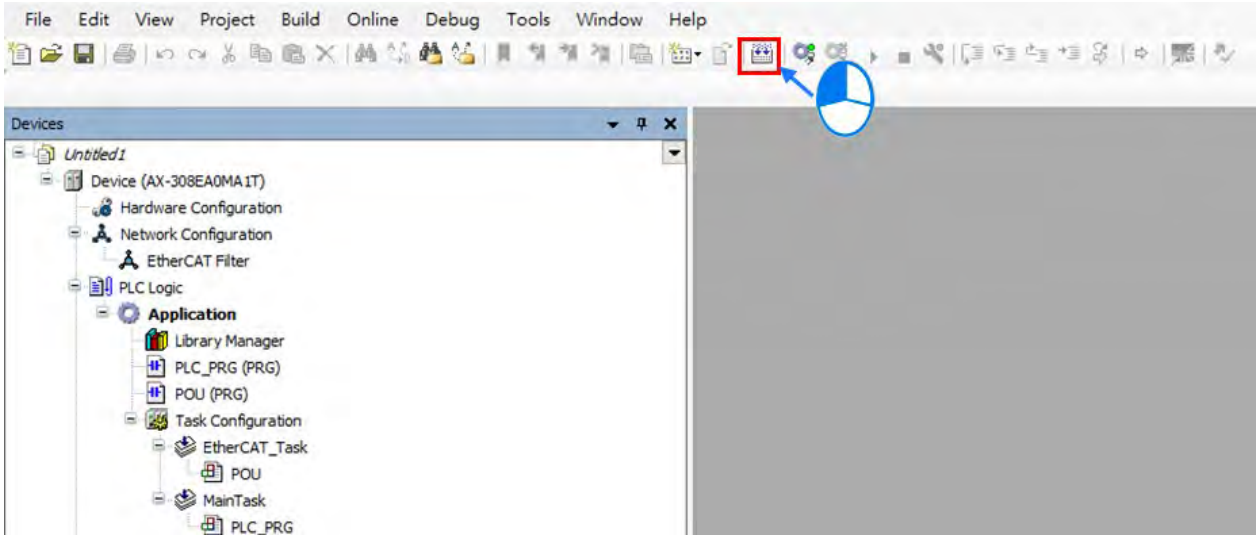
3. Select EtherCAT port “cpsw1” and click “OK”.



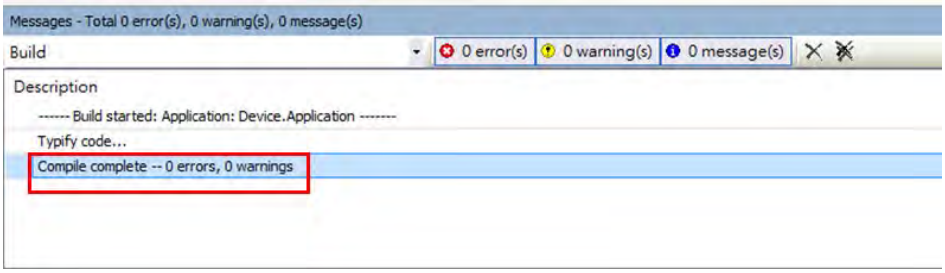
4. After finish choosing network adaptor, the setting value of Source address (MAC) is displayed.



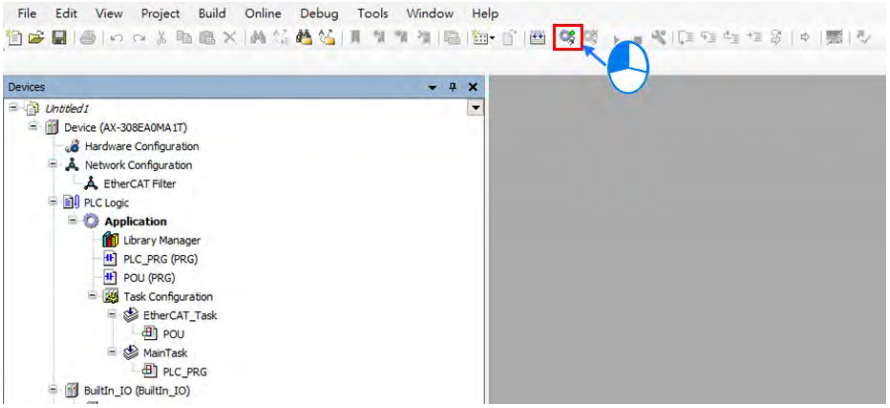
5. Click on the Compile button  to verify the correctness of program.



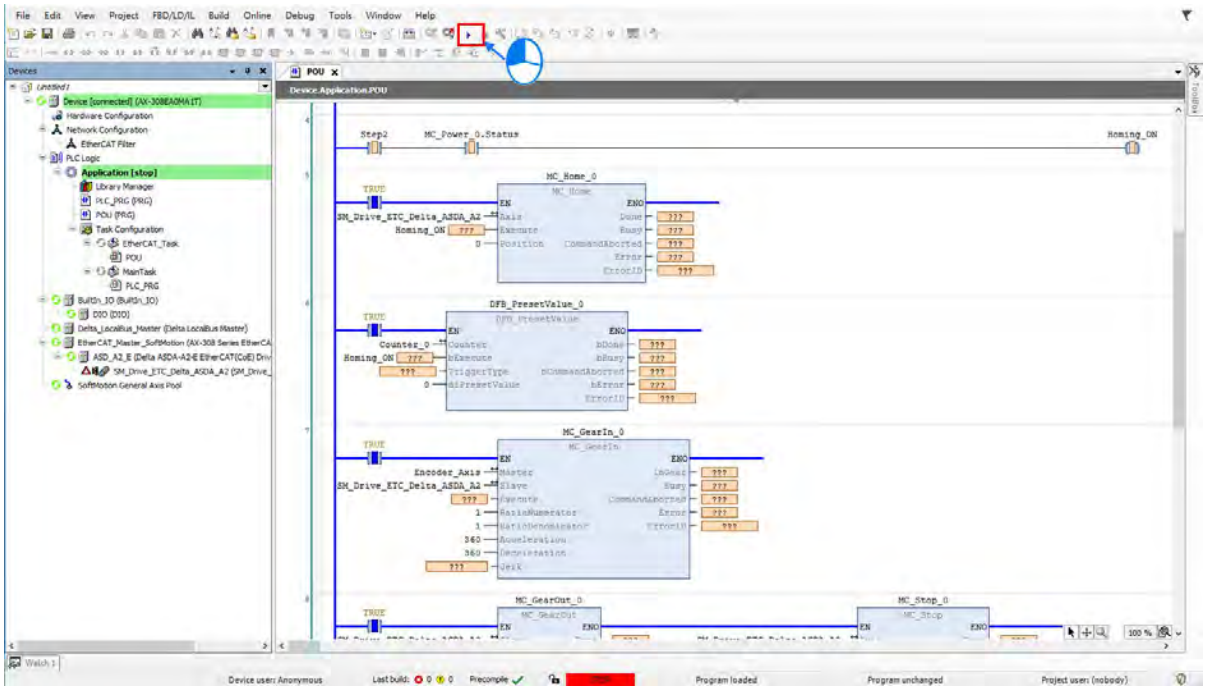
6. Upon completion of compiling, a report message of programming errors and warnings is displayed.



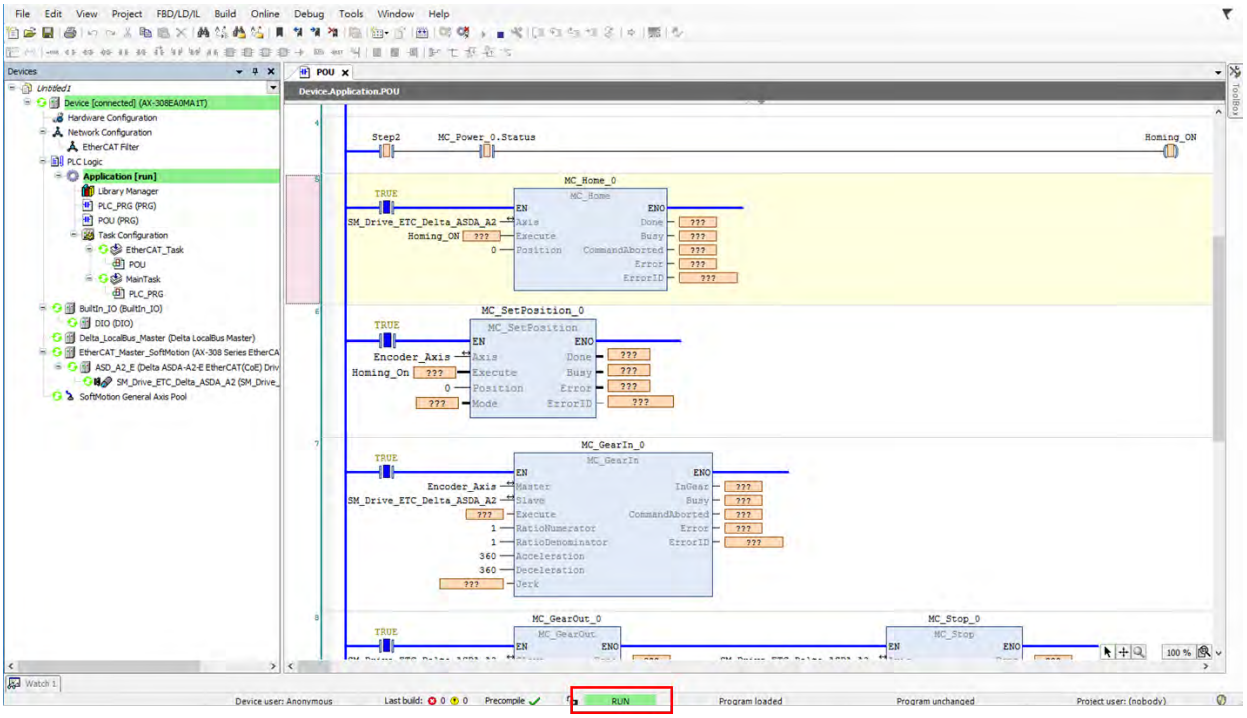
7. And then click on  to perform online monitoring.



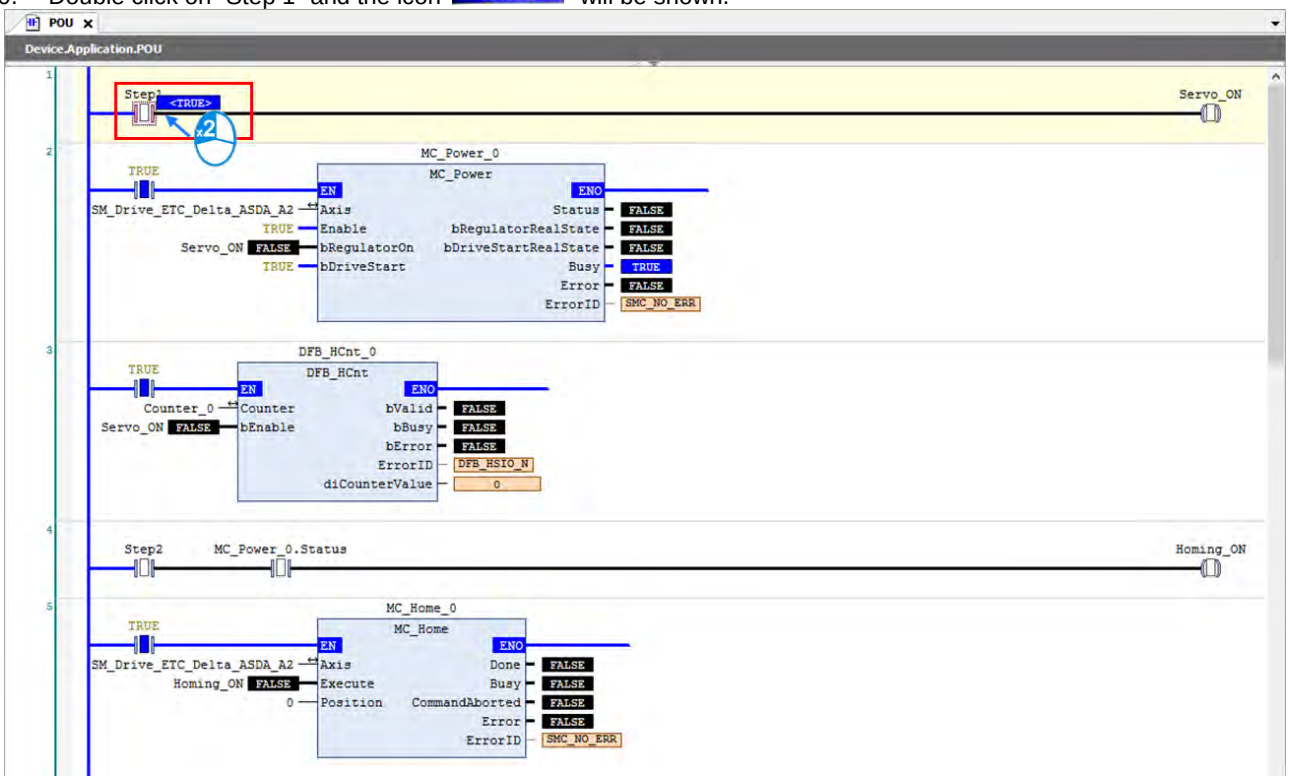
8. The program needs to be downloaded after monitoring action is performed. When the download is completed, the status of PLC shows STOP until you click on  and the status would shift to RUN.



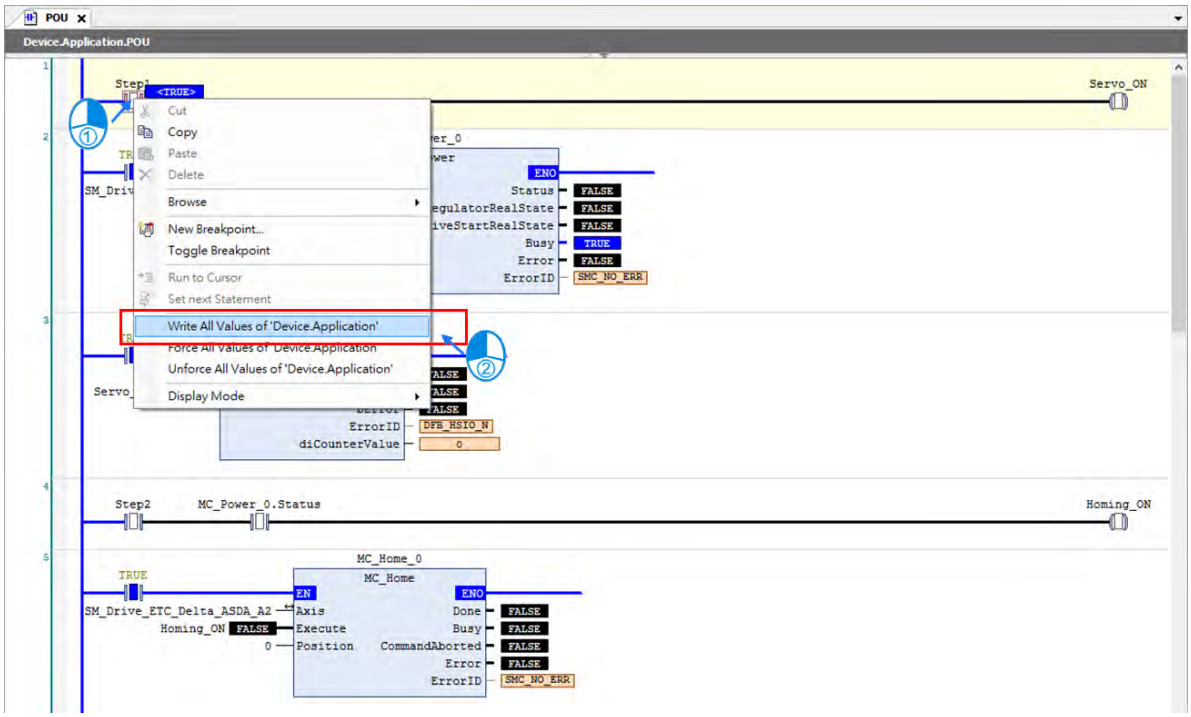
9. When the PLC is in the RUN mode, the status would be displayed as RUN at the bottom of the page and the program is ready to be executed.



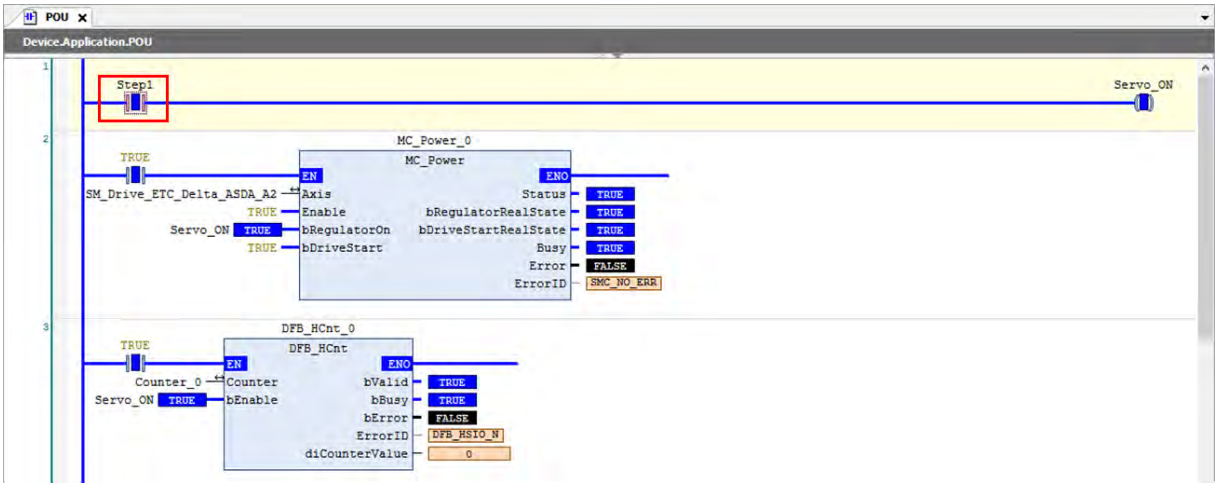
10. Double click on “Step 1” and the icon <TRUE> will be shown.



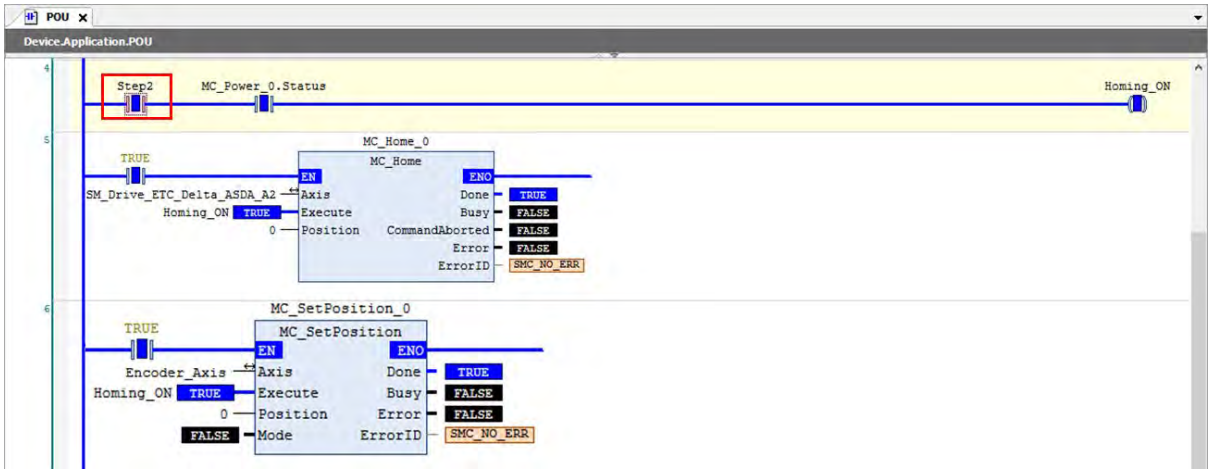
11. Right click on “Step 1” to select “Write All Values of ‘Device.Application” and all the corresponding status will be filled in, which can also be performed by using the shortcut key **Ctrl + F7**.



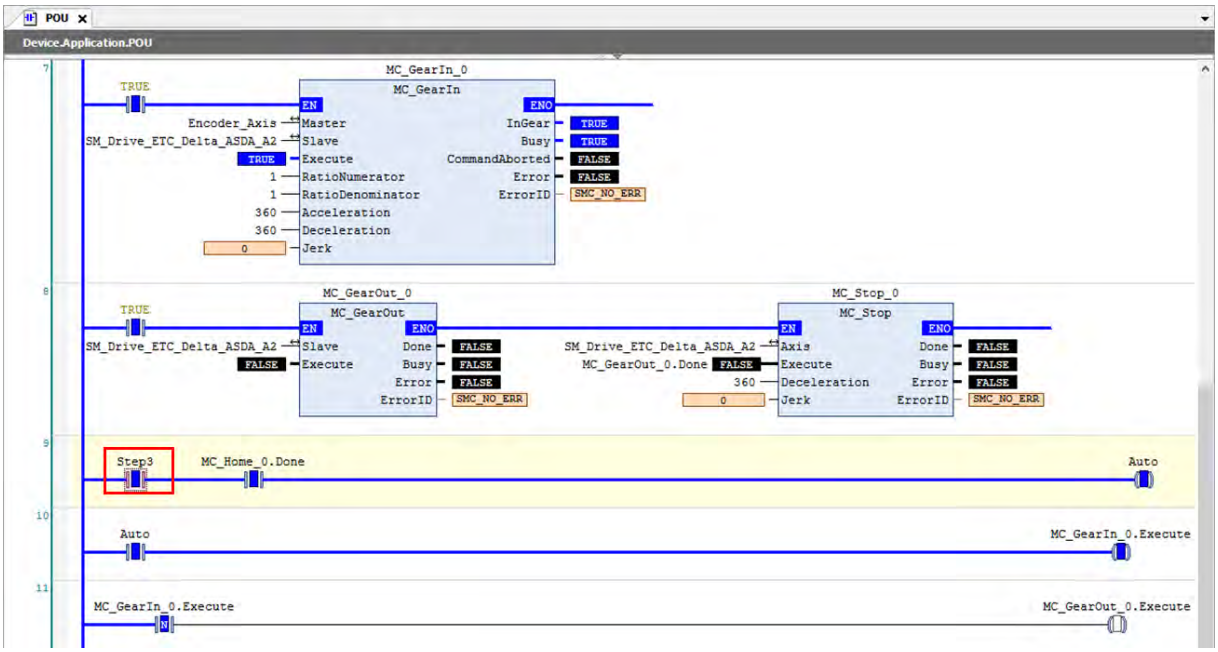
12. After activating “Step 1”, Servo ON changes to TRUE as well as activating the encoder. (Same operation to execute the following programs.)



13. Make the axis start homing and change the encoder value to zero by activating “Step 2”.

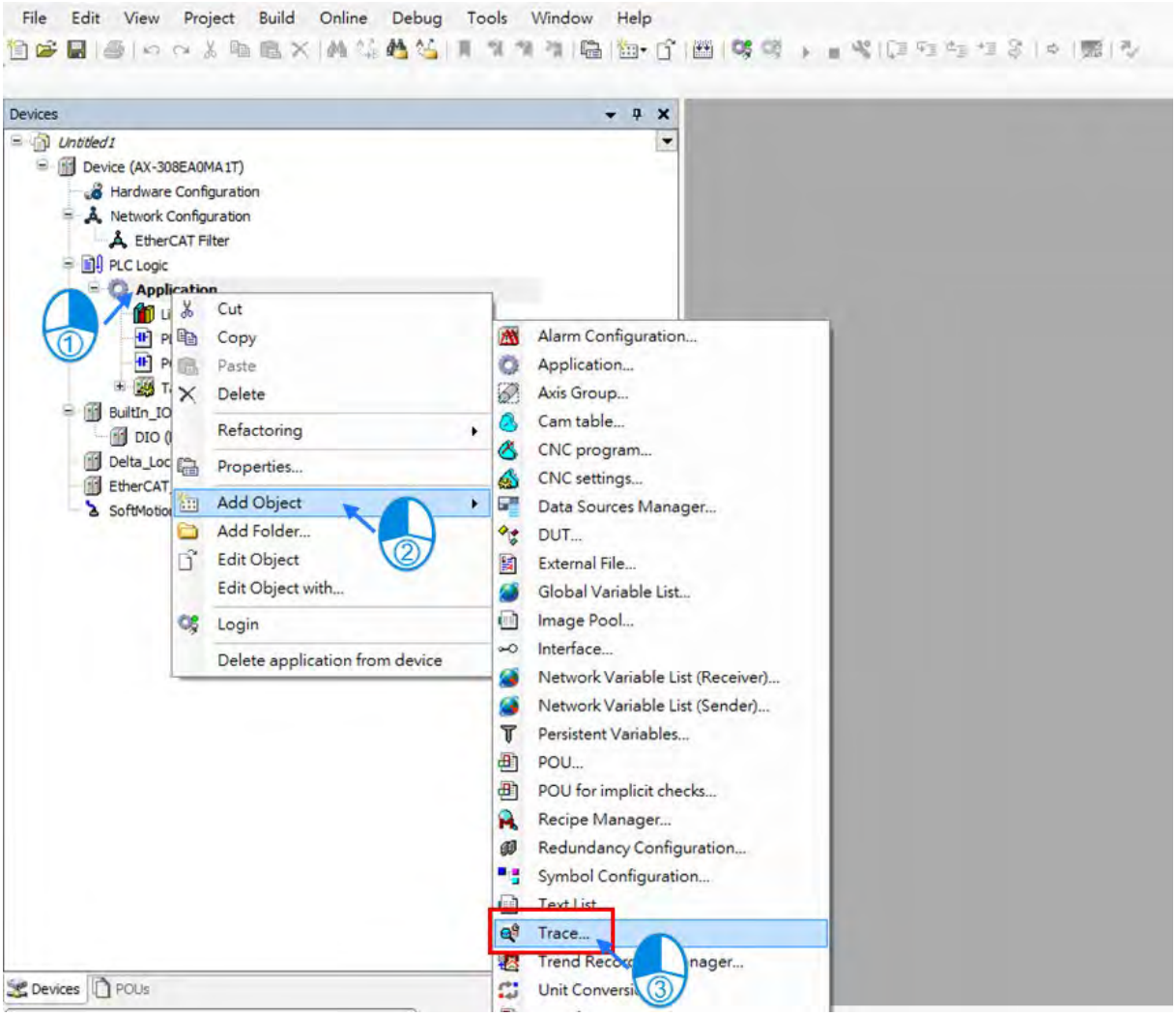


14. After activating “Step 3” and just to rotate the encoder, the slave axis will start to follow. Once “Step 3” is aborted, “MC_GearOut” will be triggered. When the Done output of MC_GearOut changes to TRUE, “MC_Stop” is triggered again. (After being decoupled, the slave axis will move at the current velocity, which needs “MC_Stop” to stop its movement.)

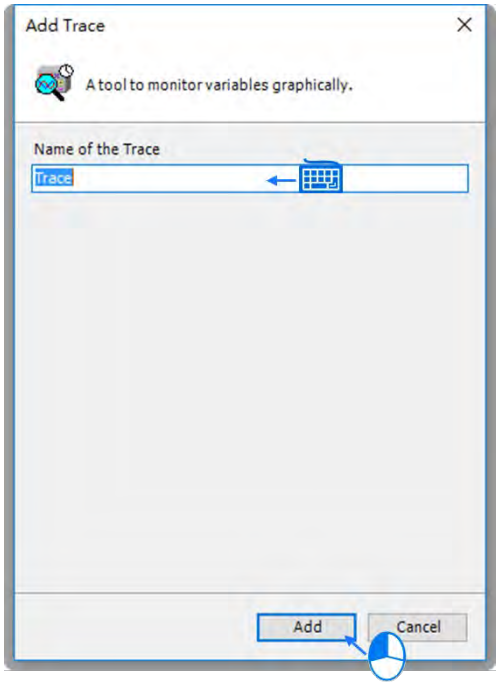


3.5.3.3 Monitor with Trace Tool

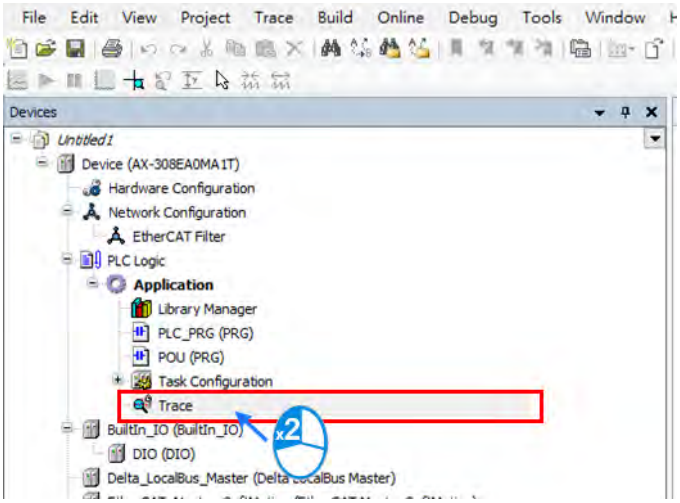
To monitor the status of input/ output pins and axes with Trace function while executing the program, navigate to “Application” > “Add Object” > “Trace”. Then you’ll open the “Add Trace” window.



1. Insert the name of Trace then click “Add” on “Add Trace” window.

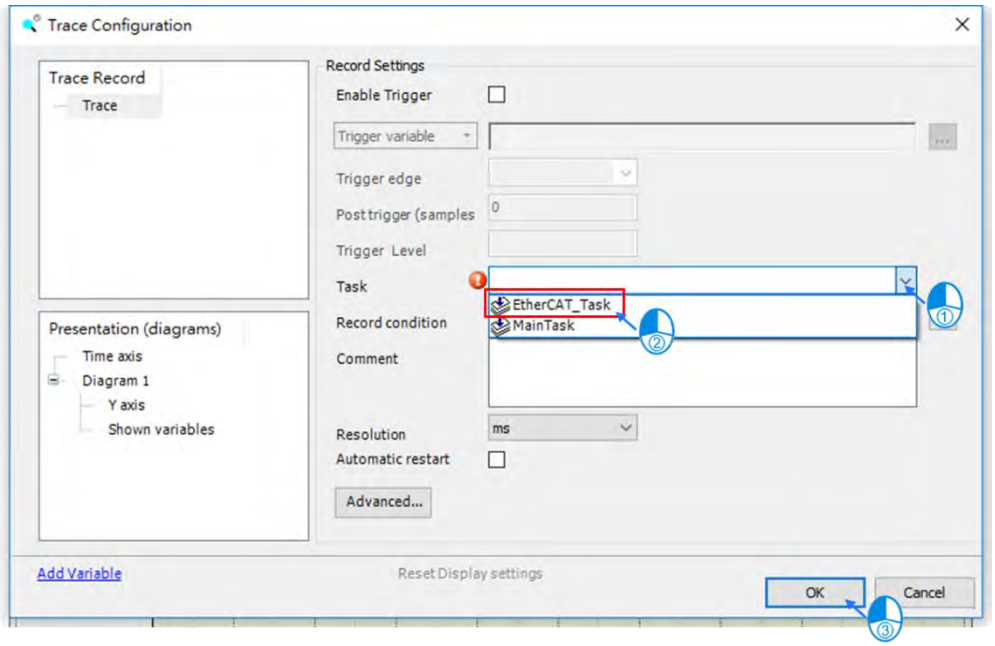


2. After the Trace is successfully added, it will be shown on the project tree on the left side of the screen. Then double click on “Trace” to open the Trace page.



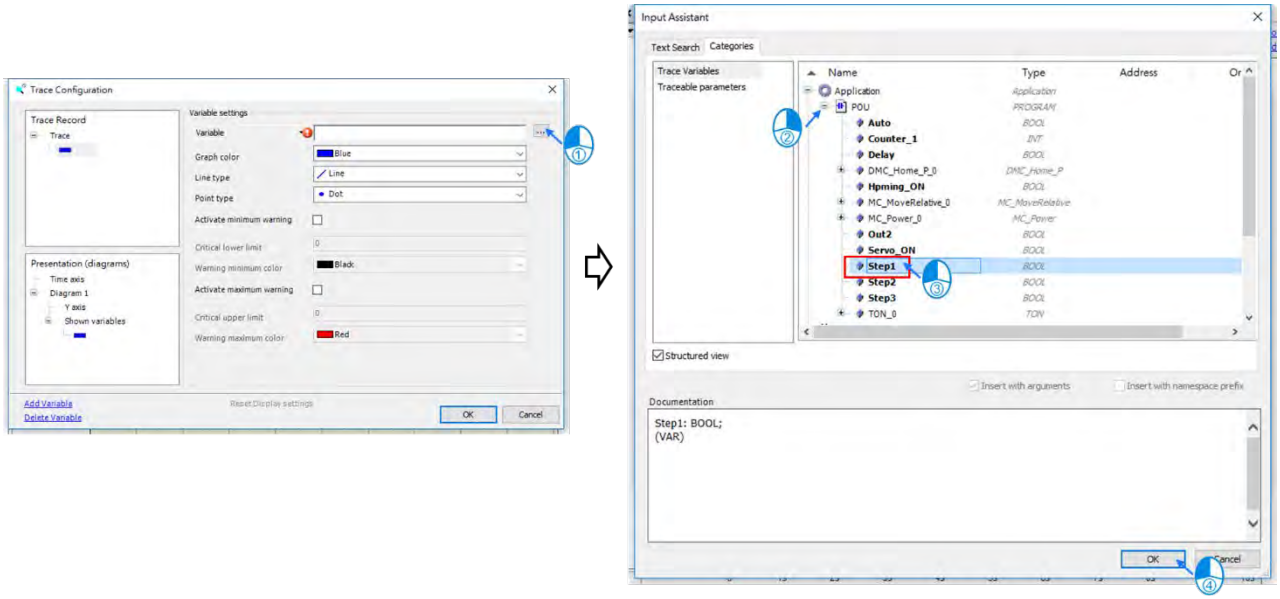
3. Click “Configuration” on the Trace page to open the configuration window.

4. Select “EtherCAT_Task” from the drop-down list of Task on the Trace Configuration window.

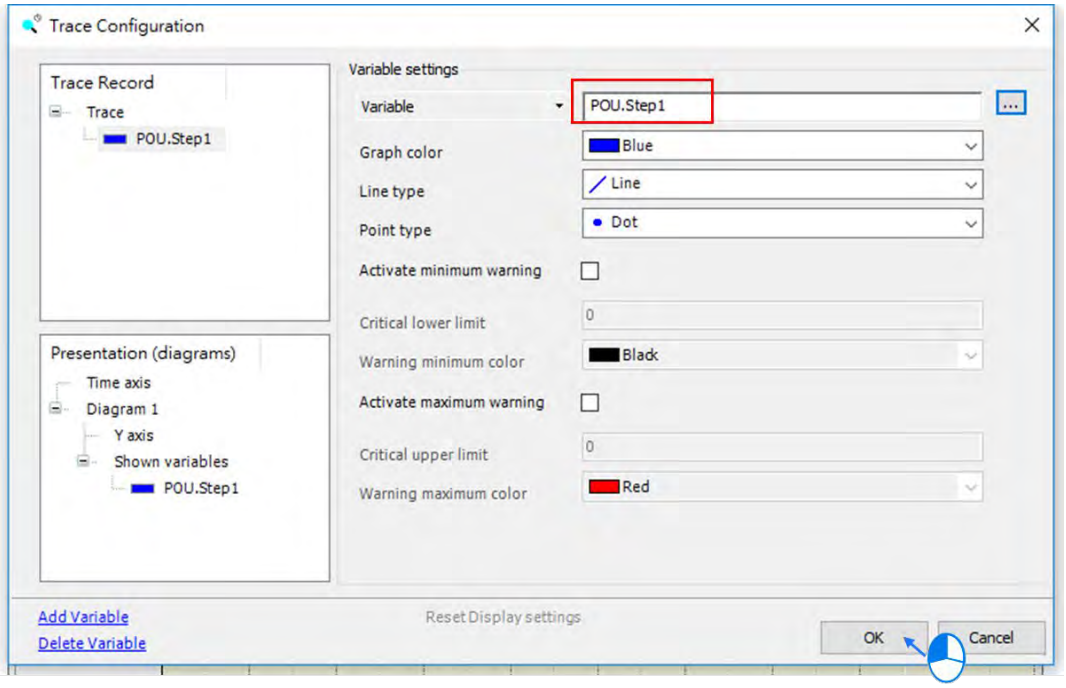


5. After finishing configuration, click “Add Variable” on the Trace page to open the Trace Configuration window.

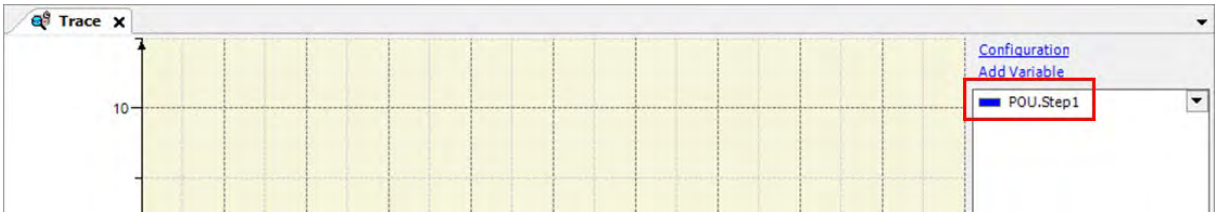
6. Click the button ... on the Trace Configuration page to add the required trace variables or traceable parameters. After the Input Assistant window is opened, expand POU on the tree and select “Step 1” then click on “OK”.



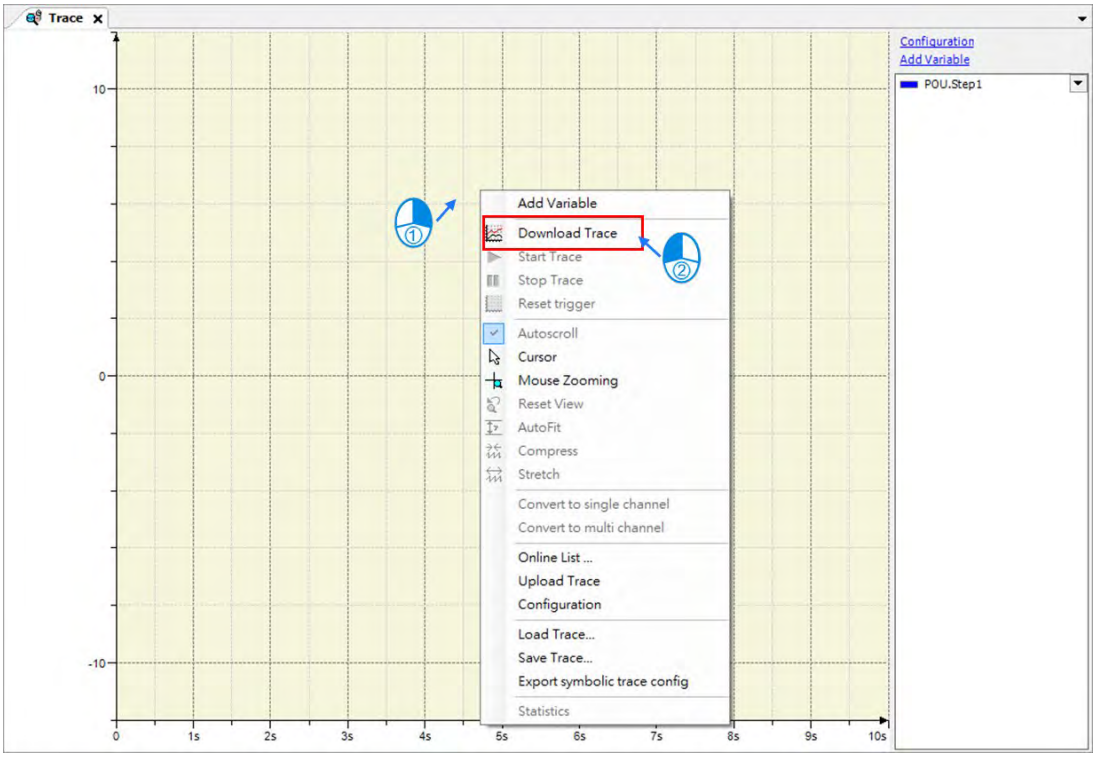
7. Make sure the chosen variable is shown in the variable field of variable settings and then click “OK”.



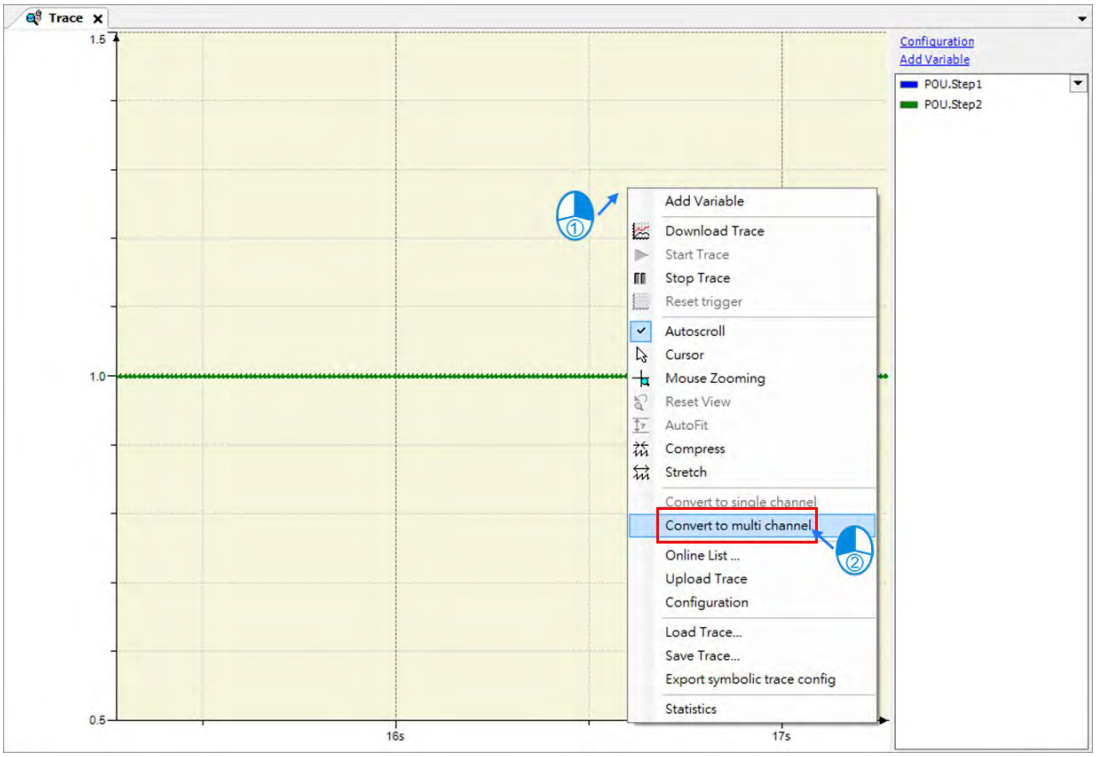
8. Afterwards, the added variable will be shown on the right side of the Trace page.



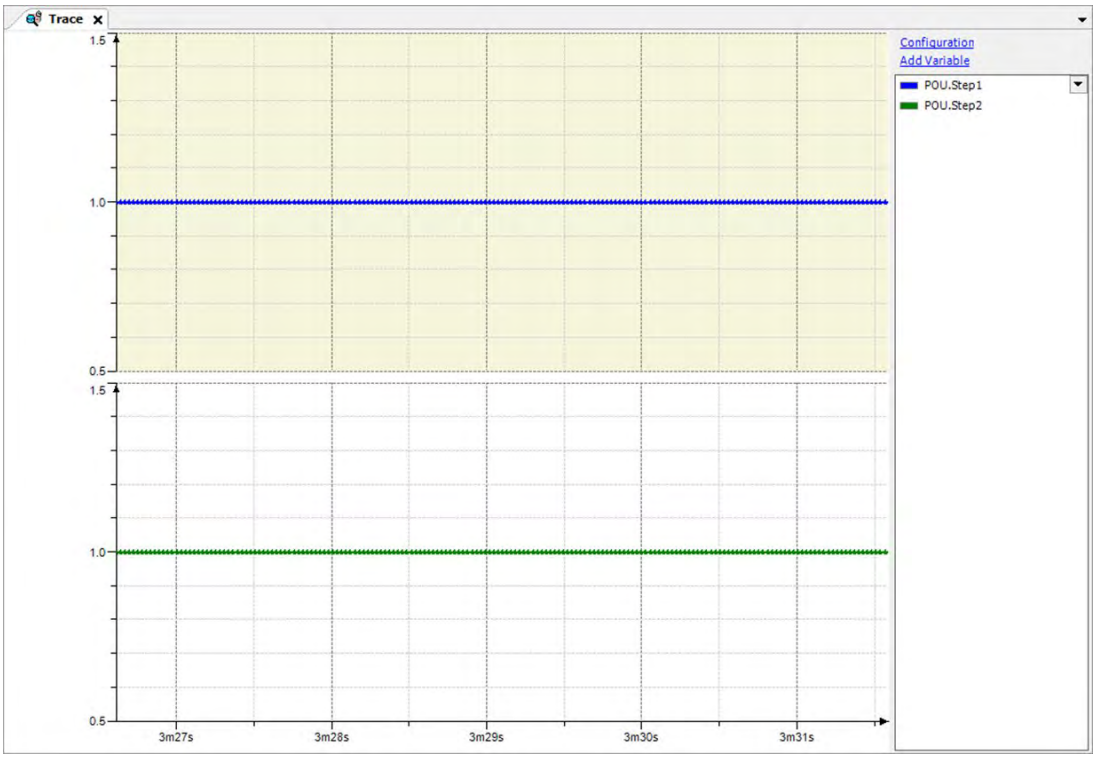
9. Right click on the Trace and select “Download Trace” to start monitoring.



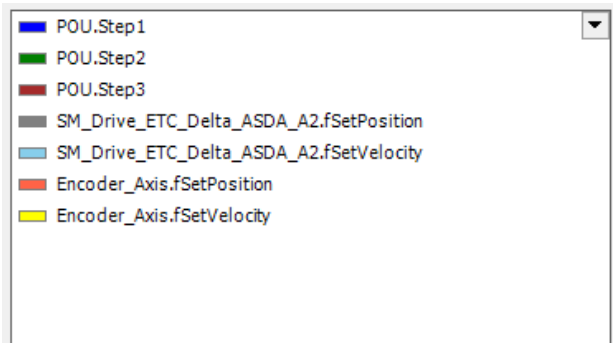
10. In case that there’re more than one variable or parameter need to be monitored, you can just right click on the scope and select “Convert to multi channel”.



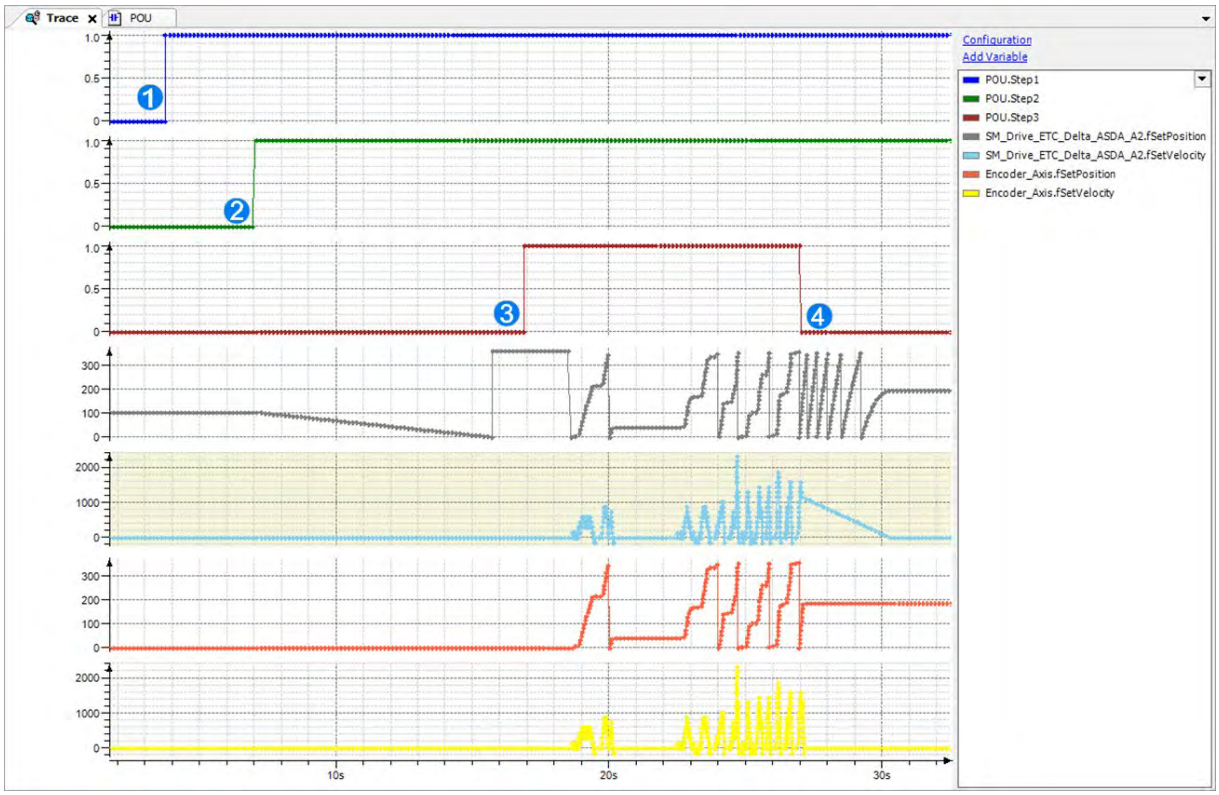
11. After choosing “Convert to multi channel”, two data-recording oscilloscope of the two chosen variables added on the right will be displayed on the page.



12. Repeat the above steps to add more required trace variables and traceable parameters.



13. While a program is running, you can observe trigger events and the current position of axis as well as the velocity via the data-recording oscilloscope.



Introduction of the process shown on the scope:

① Activate “Step 1” to enable the axis and the encoder.

② Activate “Step 2” to make the real axis perform homing and return the counting value of encoder to zero.

③ MC_GearIn will be triggered after activating Step3 and command the axis controlled via EtherCAT to follow the encoder axis. Once the master axis rotates, the slave axis will starts to follow the master axis.

④ MC_GearOut is triggered after “Step 3” is deactivated. When the Done output of MC_GearOut shifts to TRUE, MC_Stop is triggered to command a deceleration stop to the slave axis.

MEMO