



MC500 Series PLC

Programming Manual



www.leadshine.com

202402 V2.00

Contents

Chapter 1 Introduction	5
Chapter 2 Quick Start	7
2.1. Programing Environment Launching.....	7
2.1.1. MC500 Series PLC Package Install.....	7
2.1.2. Basic Function Libraries Install	8
2.1.3. Install basic device .XML file.....	8
2.2. Create New Project	10
2.3. Login And Running PLC.....	12
2.4. Device Configuration.....	14
2.4.1. Auto-Scanning Device	14
2.4.2. Off-Line Device Configuration.....	14
2.5. PLC Sample Program Writing	15
2.5.1. Create Main Program.....	15
2.5.2. Variable Definition.....	16
2.5.3. Variable address definition.....	17
2.5.4. Variable type	17
2.5.5. Create Action Program	18
2.5.6. Task Configuration.....	18
2.5.7. Locate Program Compiling Error	18
Chapter 3 Local Module.....	20
3.1. Local Module Configuration	21
3.1.1. Digital module configuration.....	21
3.1.2. Analog module configuration	21
Chapter 4 High Speed Counter	24
4.1. Add high speed device	24
4.2. Configure the high speed counter parameters	24
4.3. Setting input filter parameters	26
Chapter 5 High Speed Outputs	27
5.1. Local High Speed Pulse Axis Configuration	27
5.1.1. Axis Configuration Interface.....	27
5.1.2. High Speed Output Wiring	32
Chapter 6 Serial Port Communication.....	33
6.1. RS485 Communication wiring.....	33
6.2. Modbus-RTU Communication configuration.....	34
6.2.1. Modbus Master Station Configuration	34
6.2.2. Modbus Slave Station Configuration	39
6.2.3. Modbus Variable Address.....	39
Chapter 7 Ethernet Communication	41
7.1. Ethernet/IP Communicating	41
7.1.1. Configuring PLC As Master Station.....	41
Chapter 8 EtherCAT Configuration.....	44
8.1. Overview.....	44
8.2. Add “.Xml” File.....	44
8.3. EtherCAT Master Station Configuration.....	45
8.4. EtherCAT Slave Station Configuration.....	46
8.4.1. General Configuration.....	46
8.5. Process Data Object (PDO)	47
8.5.1. Starting Data Object (SDO)	47
8.5.2. EtherCAT I/O mapping.....	47
8.6. Add 402 Axis To Servo Slave Station	47
Chapter 9 Motion Control.....	49
9.1.1. High Speed Pulse Axis Control Program.....	49
9.2. EtherCAT Axis Configuration	51
9.2.1. General Configuration.....	51
9.2.2. Scaling And Mapping	51

9.2.3. Homing Parameters Configuration	53
9.2.4. EtherCAT Axis Control Program	53
9.3. CANopen axis configuration	55
9.3.1. General configuration	56
9.3.2. CANopen bus manager configuration	57
9.3.3. CANopen slave configuration	57
9.3.4. CANopen axis motion program	58
9.4. E-CAM function	61
9.4.1. Function Block	61
9.4.2. Sample Program	61
9.5. G-code Function	63
9.5.1. G-code in PLC program	63
9.5.2. G-code in processing files	65
Chapter 10 Motion Control Command	66
10.1. Command table	66
10.2. Axis Variables	66
Chapter 11 Libraries Description	68
11.1. Basic Libraries list	68
Q&A	69
Contact us	70

Record of revisions

Manual version	Date	Update contents
V1.10	2023/10/07	Add high speed counter configuration
V1.11	2023/10/12	Modify high speed counter configuration
V1.12	2023/10/18	Add standard wiring diagram
V1.13	2023/10/20	Add project environment configuration
V1.14	2023/10/20	Add software environment configuration, and E-CAM function
V1.15	2023/11/17	Add the G-code function
V2.00	2024/2/04	Adjust some content

Chapter 1 Introduction

MC500 is Leadshine self-developed new generation basic medium-sized PLC product that supports EtherCAT bus control. It can encapsulate and reuse processes through ST.LD, FB/FC functions, and achieve multi-level network communication through RS485, RS232, CAN, Ethernet, and EtherCAT interfaces.

The MC500 series functions:

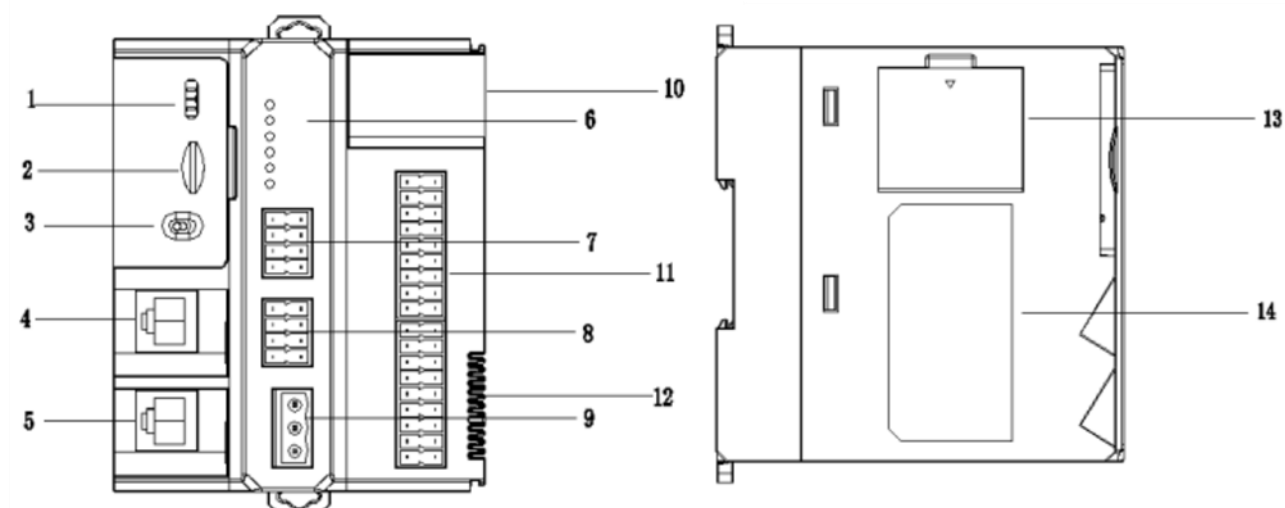
- 1) Interpolation, Multidimensional linear interpolation, circular interpolation, and continuous interpolation
- 2) E-CAM: By digitizing cam movements, the problems of low precision, easy wear and noise in mechanical cams can be solved.
- 3) Flying Shear: By setting motion values, a rotary cutting cam table can be established within the synchronization zone, with the spindle and slave shafts operating at a certain speed ratio.
- 4) Chasing Shear: By setting motion values a chasing cam table can be established, which is suitable for application scenarios such as cutting and filling

MC500 series PLC specification

Model Specifications	MC508CS	MC516CS	MC532CS
	EtherCAT 8 axes + pulse+dir 6 axes	EtherCAT 16 axes + pulse+dir 6 axes	EtherCAT 32 axes + pulse+dir 6 axes
Axes of Pulse +dir	Local 6 axes 200K pulse output		
Extention Capacity	Maximum extend 32 R2 series extension modules		
EtherNET	1* EtherNET port, Modbus,Socket,program upload or download ,debugging		
EtherCAT	EtherCAT master , up to 128 slaves		
Serial port communication	RS232*1,RS485*2,free communication protocol, modbus rtu master and slave		
CAN	Maximum 31 slave		
Capacity of Program file	20 M Byte		
Capacity of data	40 M Byte		
Power-Failure RetentionArea	512K Byte		
USB port	Type-C port, program upload or download,debugging		
SD card slot	User download program, standard micro SD card,FAT32 type, Maximum capacity 32G		
Function	Point to point , E-CAM, Interpolation		
High-speed counter	6 inputs ,200K		

IO Quantity	High-speed input/ normal input: 12 inputs 200K/4 inputs 1K(NPN/PNP) High-speed output/ normal output: 12 outputs 200K/4 outputs 10K(NPN)
RTC clock	RTC
Program software	Leadsys Studio ,CODESYS V3.5(SP15) or higher
Program Language	ST,LD,CFC,SFC FBD,IL
Power input	DC 24V
Power rating	3.6W
Dimension	L 98.50mm*W 81.75mm*H100.00mm

MC500 series port description



1. USB port
2. SD card slot
3. RUN/STOP/RESET Switch
4. Internet port
5. EtherCAT port
6. Running status indicator light
7. RS485 port
8. RS232 port, CAN port
9. Power port
10. I/O status indicator light
11. I/O port
12. Extension module port
13. Battery slot
14. Label

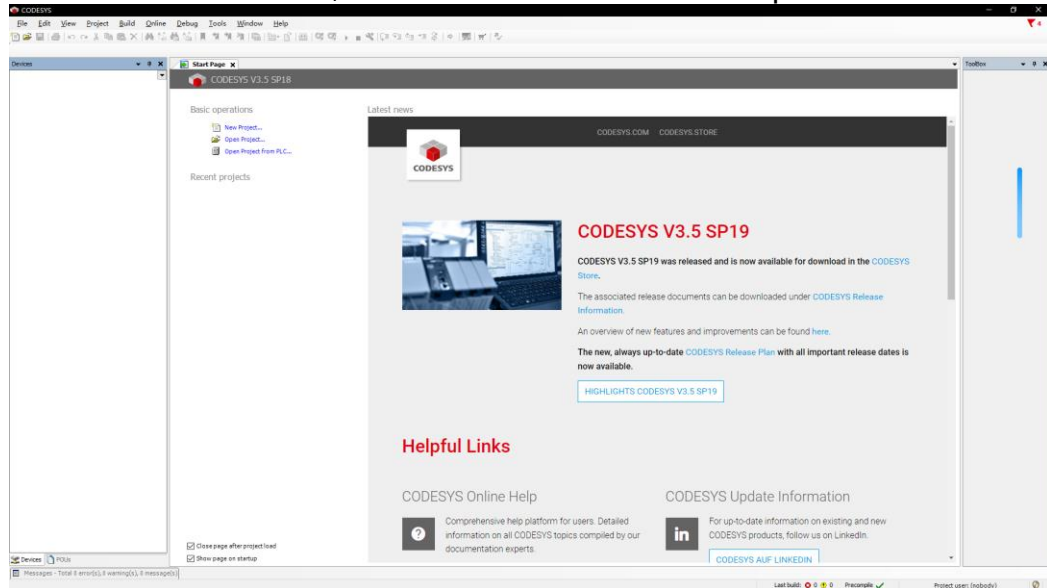
Chapter 2 Quick Start

Below contents introduced the basic steps before using CODSYS software to program. When user need to program in another new upper computer, please follow these contents set configuration.

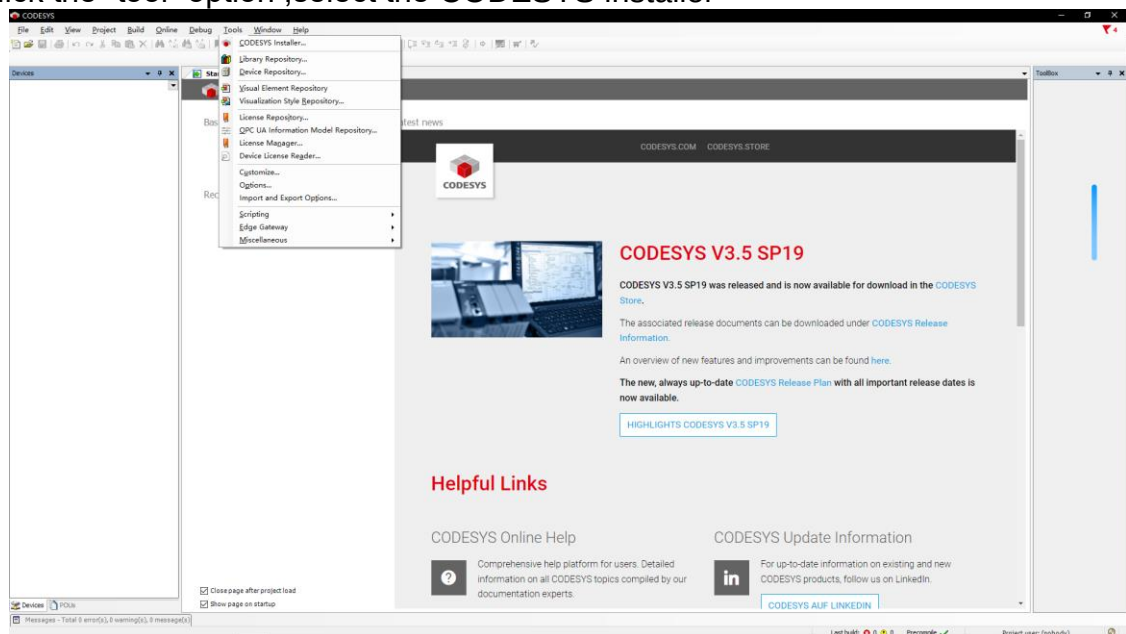
2.1. Programing Environment Launching

2.1.1. MC500 Series PLC Package Install

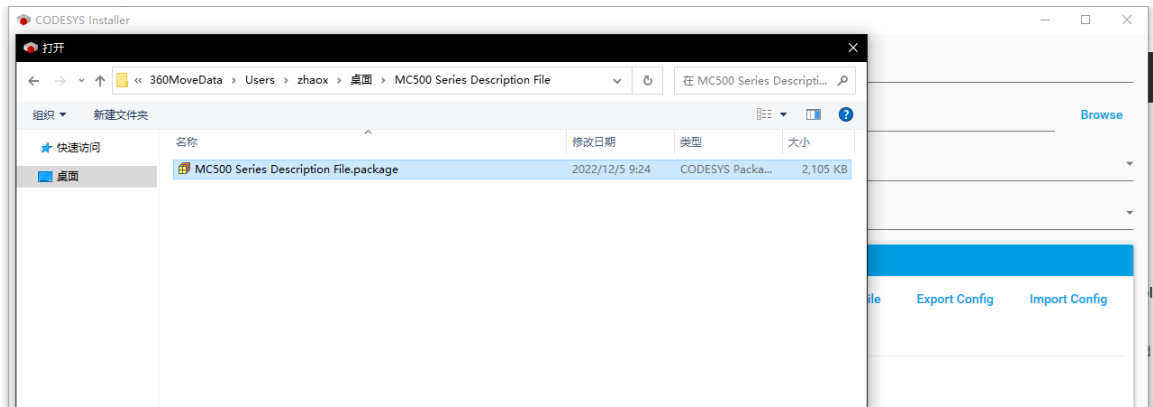
1) Double-click the CODSYS icon , the launch interface as below picture



2) Then click the “tool” option ,select the CODSYS installer

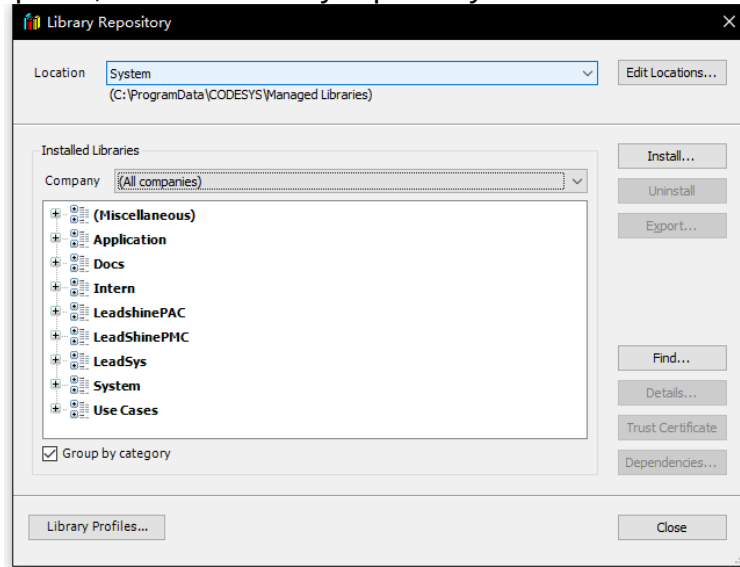


3) Click the “install file”, find the “MC500 series description file .package” to install.

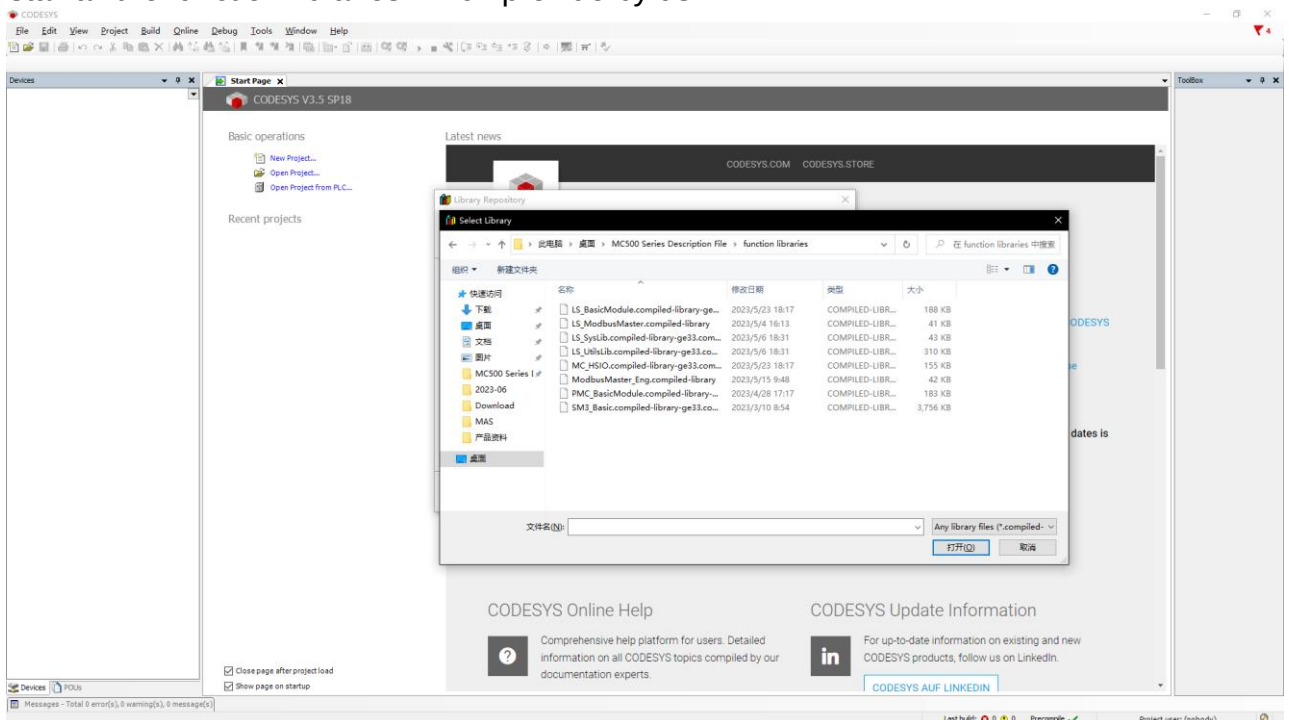


2.1.2. Basic Function Libraries Install

1) Then click the “tool” option ,select the library repository to install the function library



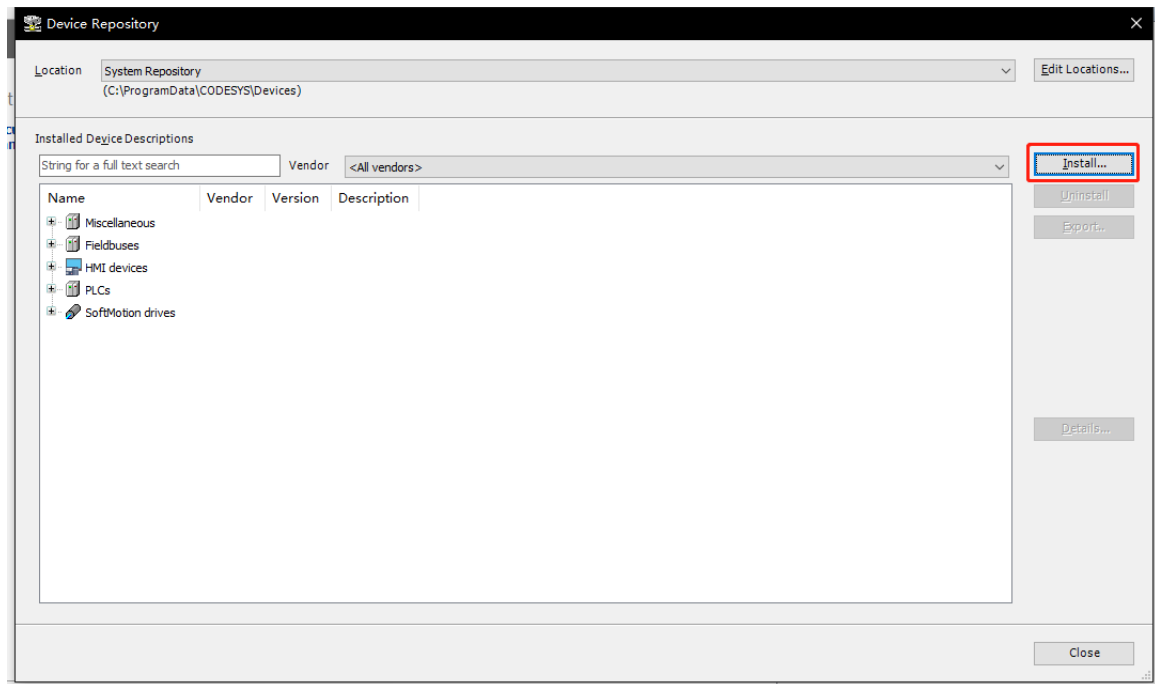
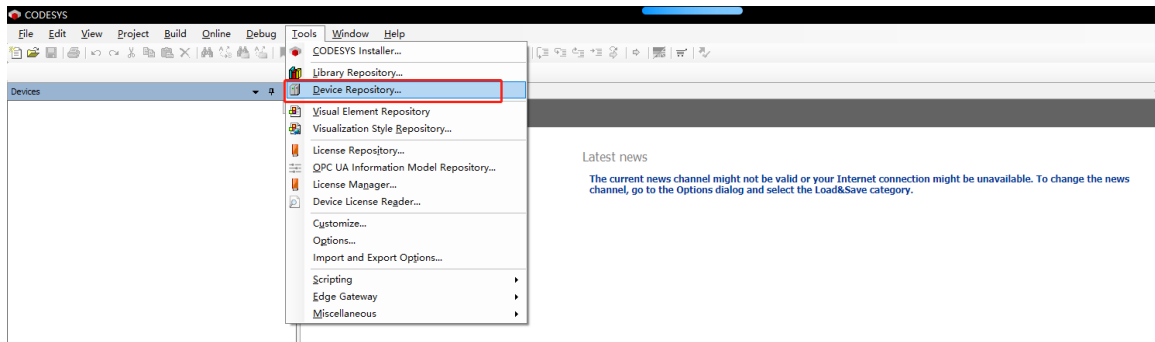
2) Install all the function libraries which provide by us.



2.1.3. Install basic device .XML file

The basic devices include the high speed IO device and local bus device, please refer to follow contents to install the XML files.

1) Click “tool”, select device repository, then click install.



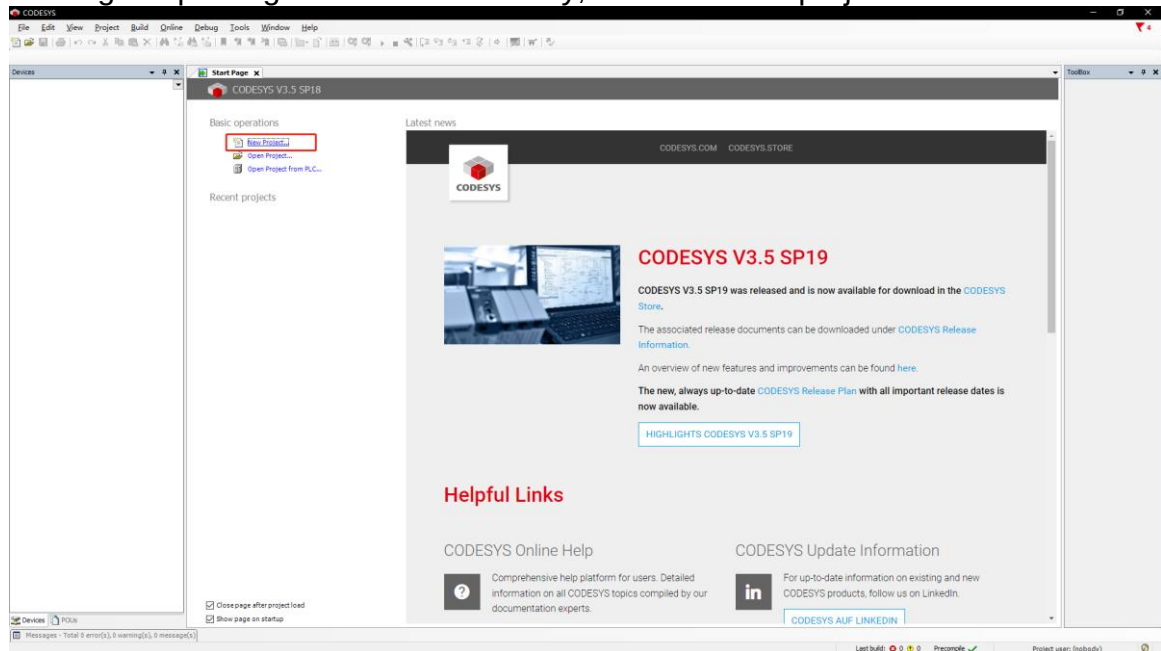
2) Select the devices XML files to install

名称	修改日期	类型	大小
LocalBus_IEC_devices_xml	2022/10/14 13:58	文件夹	
High Speed IO_220728.devdesc.xml	2022/10/14 13:58	XML 文件	62 KB
MC516-IEC-V3.5.15.40 220726.devde...	2022/10/14 13:58	XML 文件	84 KB
MC532-IEC-V3.5.15.40 220726.devde...	2022/10/14 13:58	XML 文件	85 KB

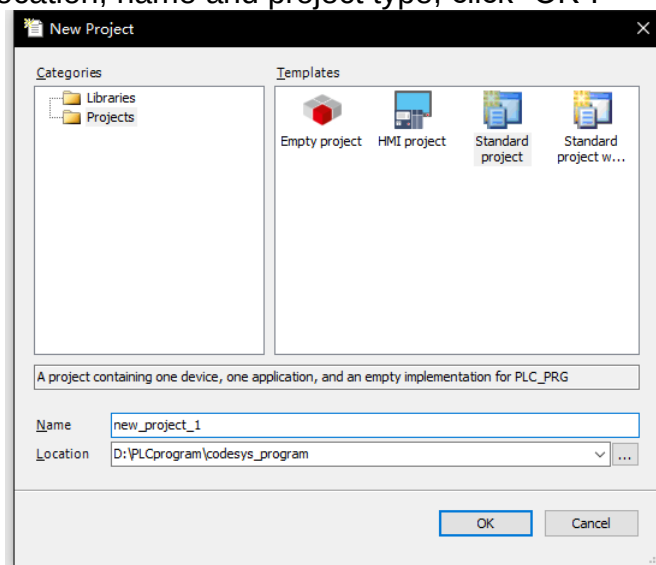
名称	修改日期	类型	大小
LocalBus_Master.devdesc.xml	2022/10/14 13:58	XML 文件	18 KB
LocalBus_PM0016.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM0016P.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM0016R.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM0032.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM0032N1.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM0032N2.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM1600.devdesc.xml	2022/10/14 13:58	XML 文件	15 KB
LocalBus_PM1616.devdesc.xml	2022/10/14 13:58	XML 文件	17 KB
LocalBus_PM3200.devdesc.xml	2022/10/14 13:58	XML 文件	16 KB
LocalBus_PM32001.devdesc.xml	2022/10/14 13:58	XML 文件	16 KB
LocalBus_PM32002.devdesc.xml	2022/10/14 13:58	XML 文件	16 KB
LocalBus_PMA0004IV.devdesc.xml	2022/10/14 13:58	XML 文件	25 KB
LocalBus_PMA0400IV.devdesc.xml	2022/10/14 13:58	XML 文件	21 KB

2.2.Create New Project

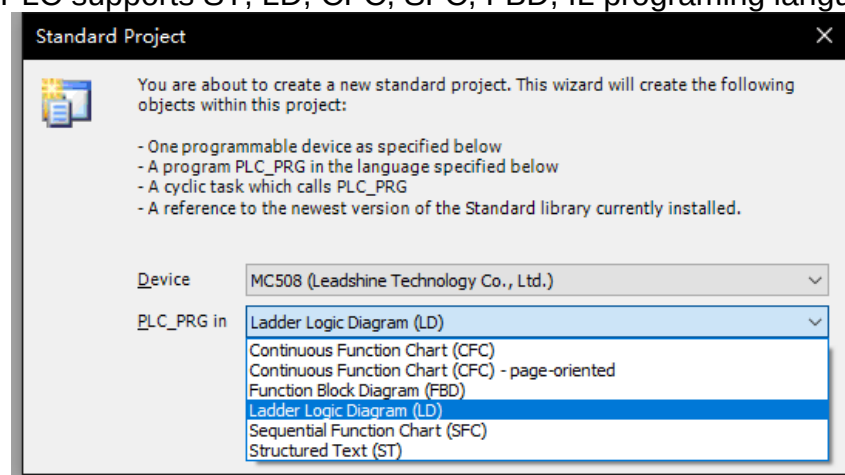
1)After installing the package and function library, click the “new project”



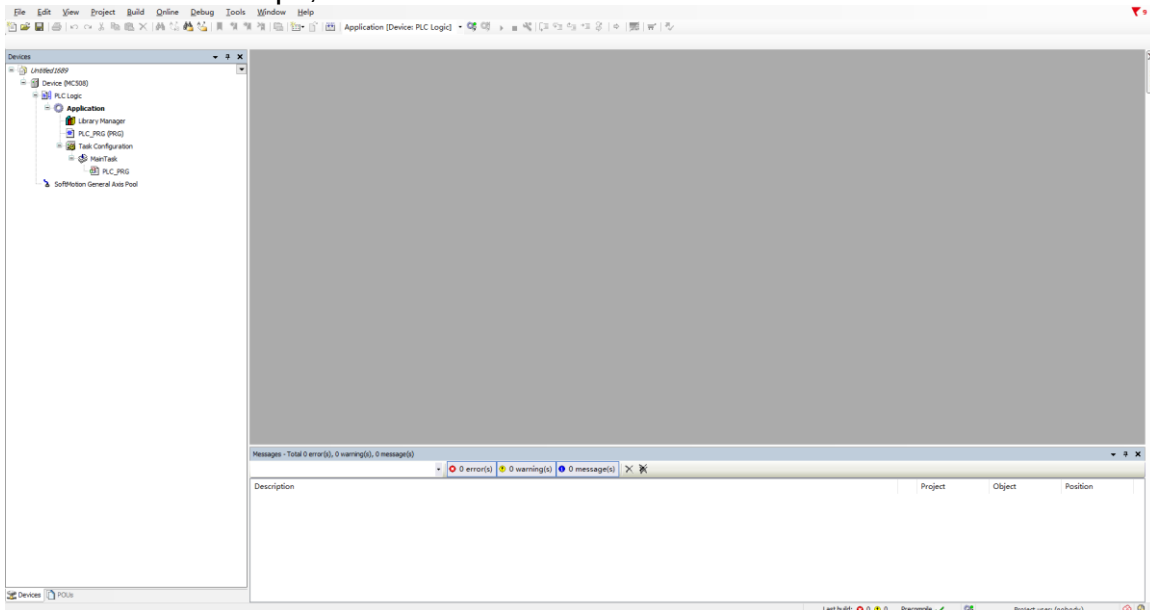
2)Then enter the project location, name and project type, click “OK”.



3)Selecting the device and programming language,
MC500 series PLC supports ST, LD, CFC, SFC, FBD, IL programming language.

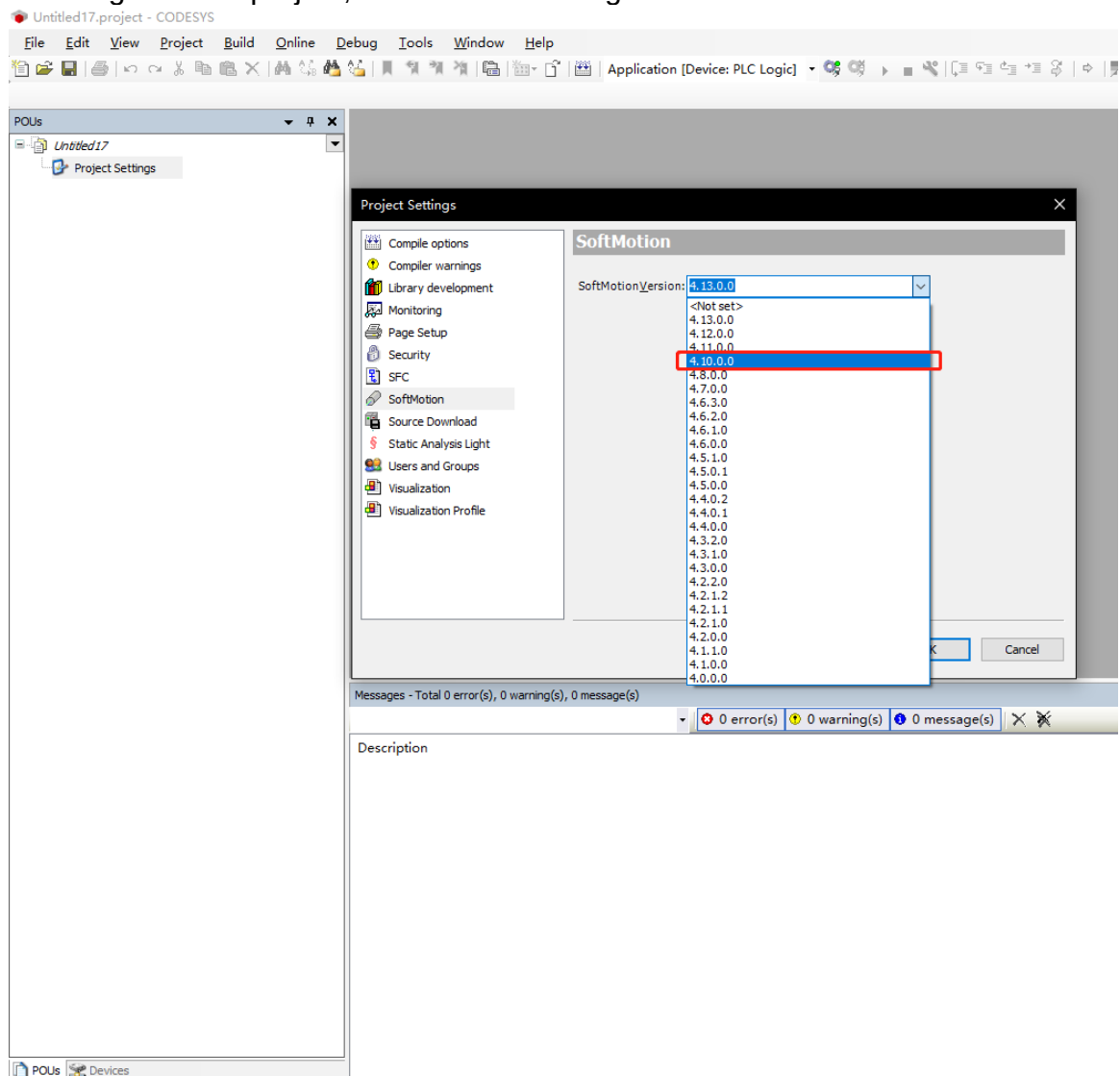


4) After finish above steps, window shows as below



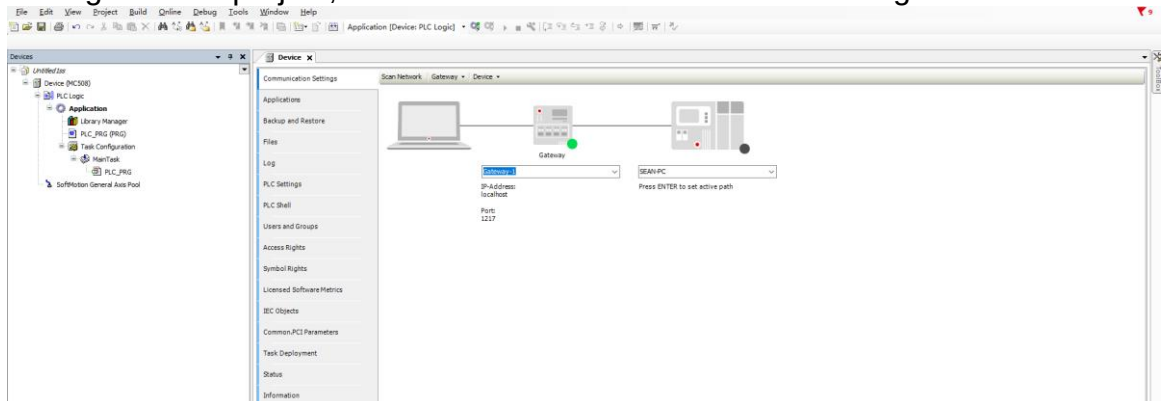
5) Change the SoftMotion version

The function libraries which developed by Leadshine are basic on the version 4.10.0.0, so after creating the new project, user have to change the SoftMotion version.

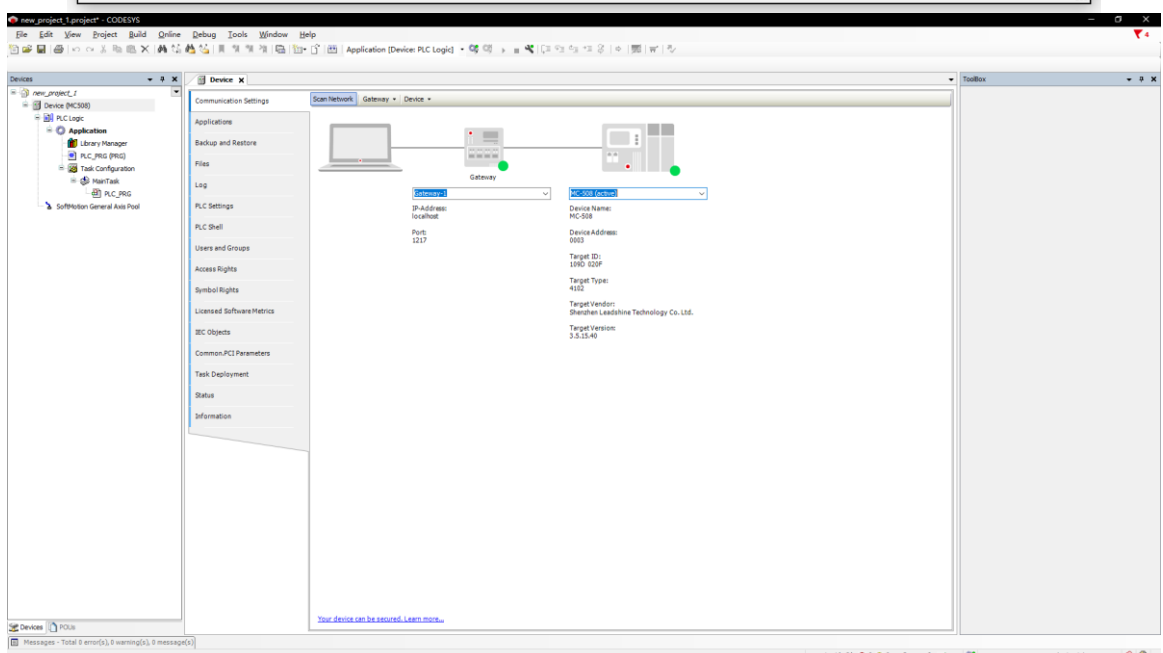
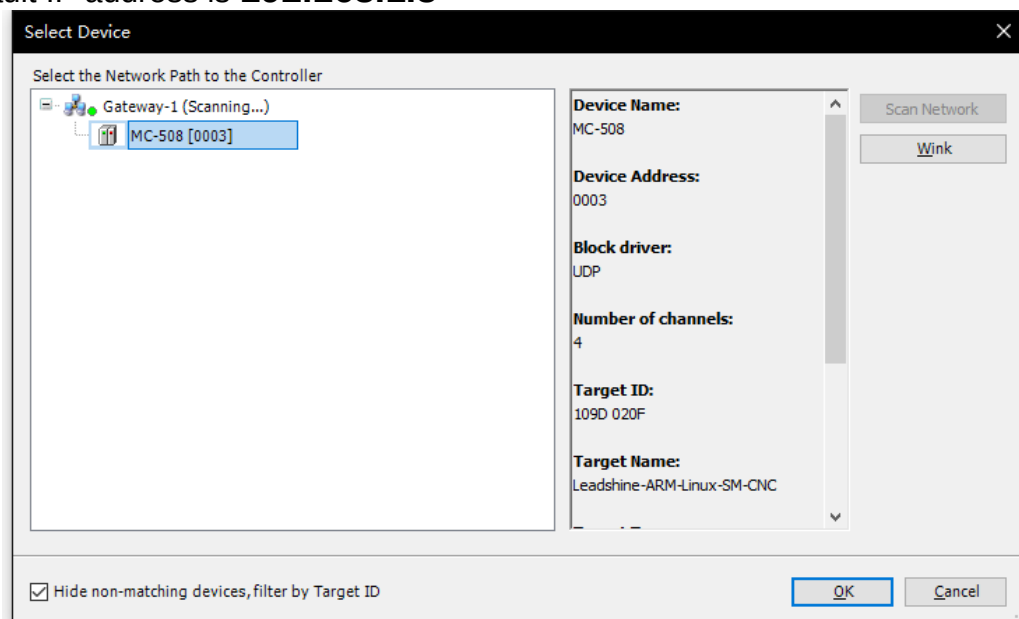


2.3.Login And Running PLC

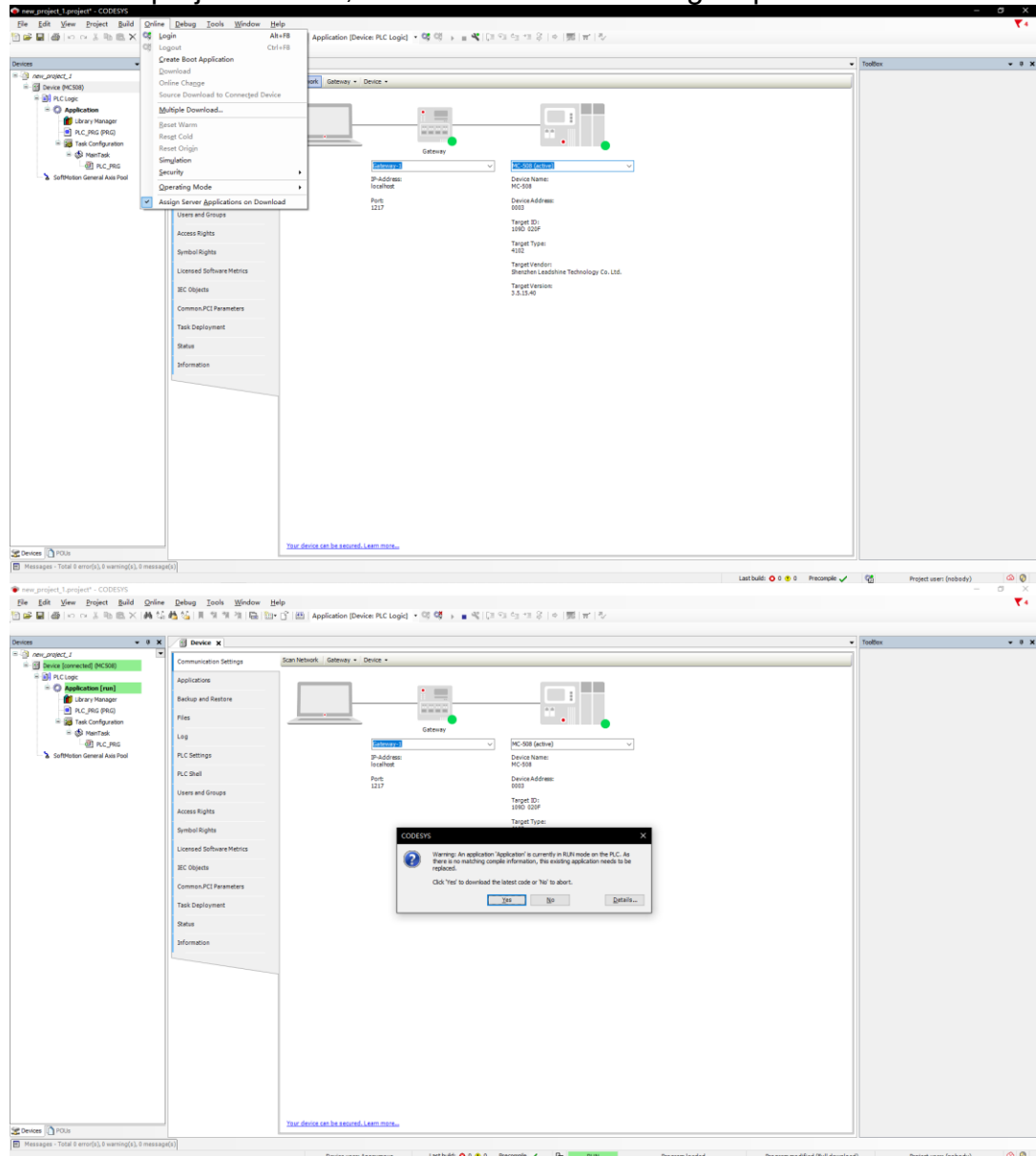
1)After creating the new project, click the device to check the PLC configuration.



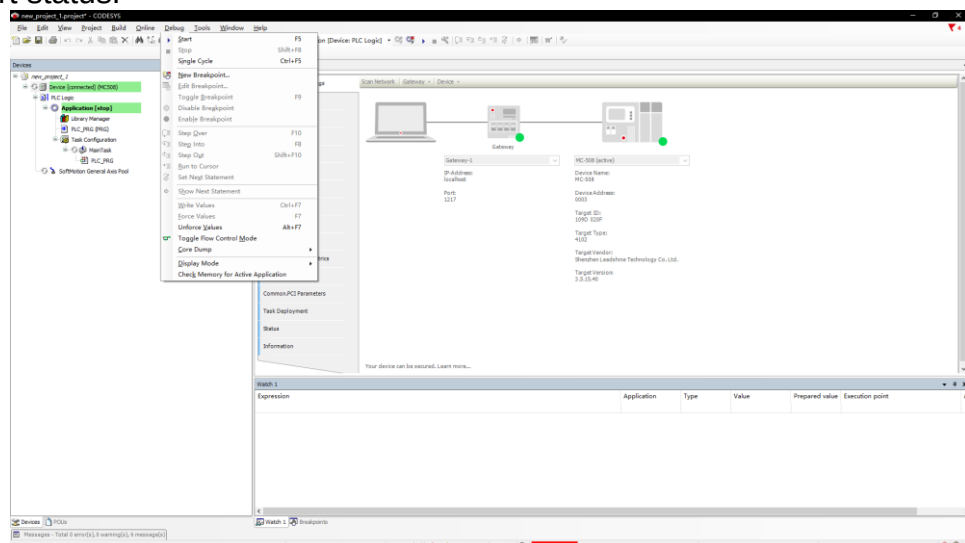
2)Click “Scan Network” to find the PLC, Click “OK” check the information of the device.
The default IP address is **192.168.1.3**



3) Then download the project to PLC, click “online” to select login option



4) After downloading the program, PLC will turn to the stop status, click “Debug” to modify the PLC to start status.

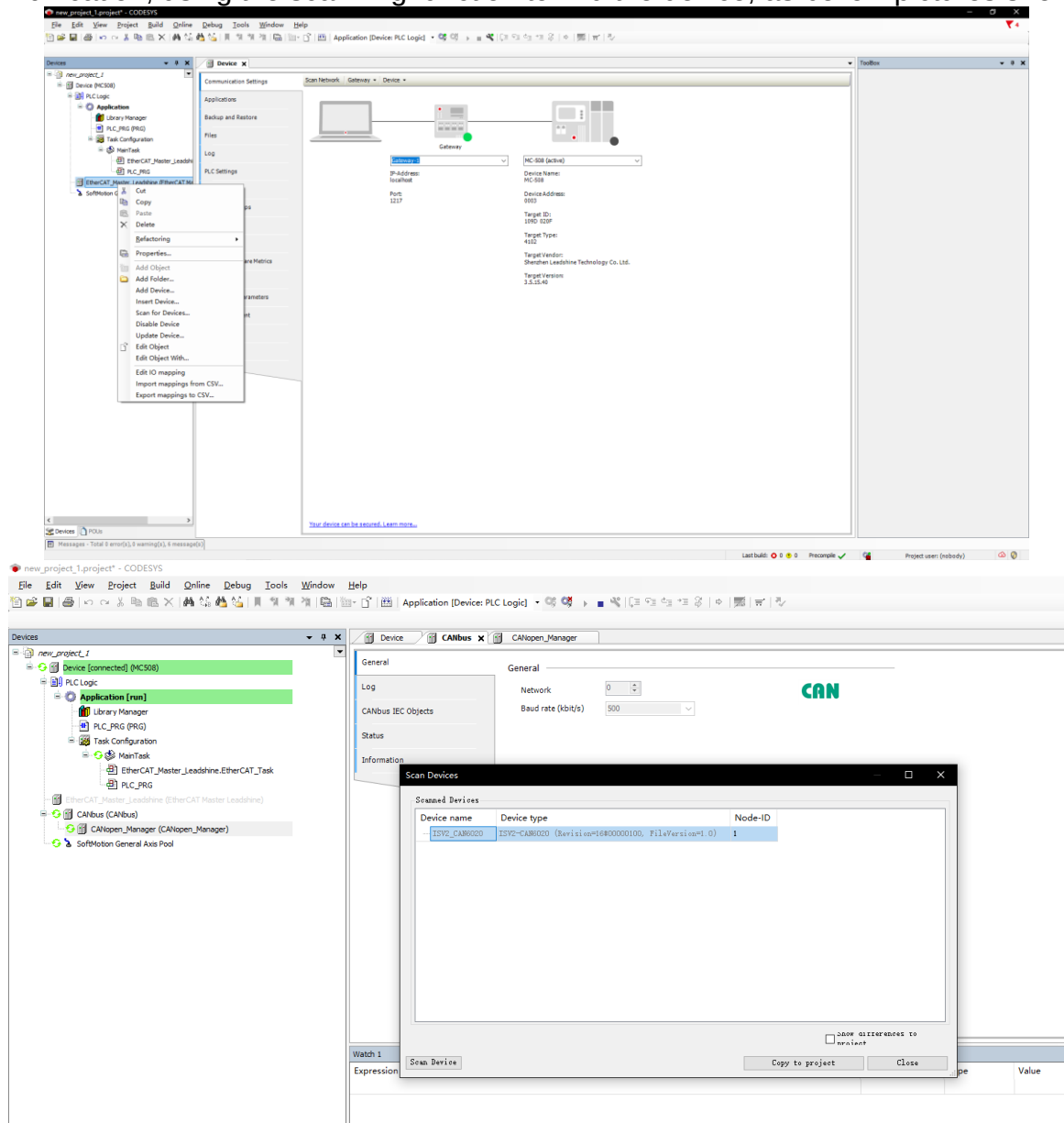


2.4.Device Configuration

Normally, there are two different methods to configure the drive or other device. If drive or device has connected to the PLC, user can use the auto-scanning function to help user to configure

2.4.1.Auto-Scanning Device

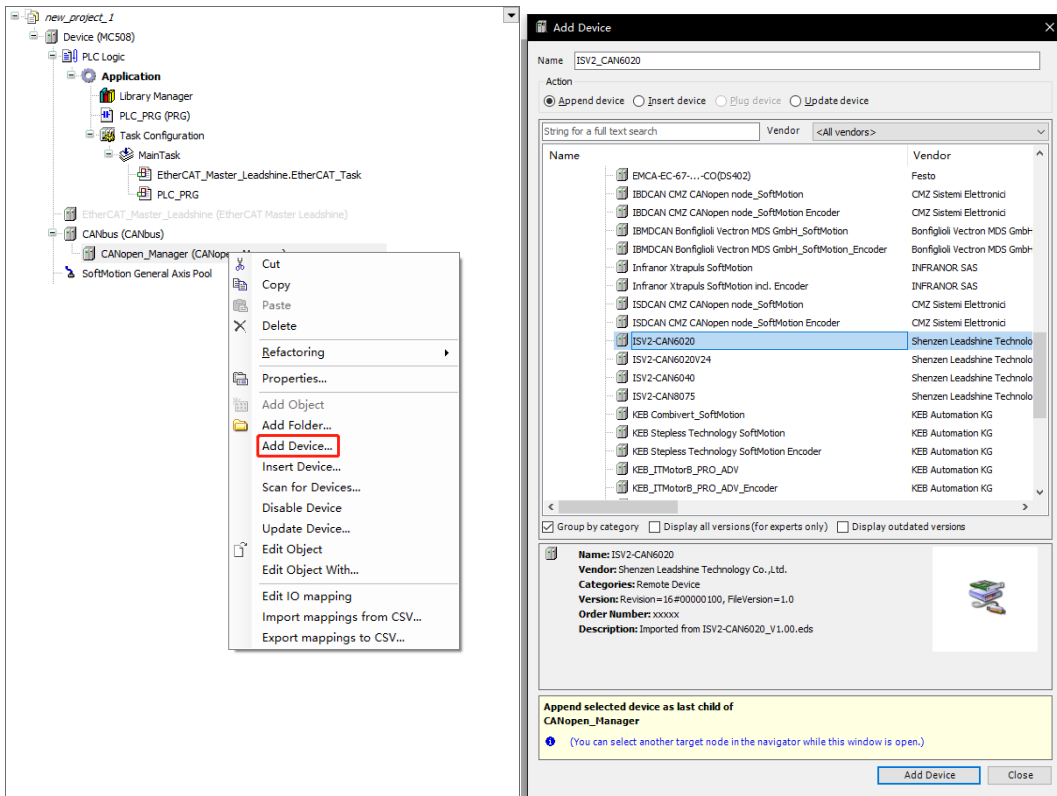
When the device have connected to PLC via EtherCAT, Ethernet/IP, or CANopen communication, using the scanning function to find the device, as bellow pictures show.



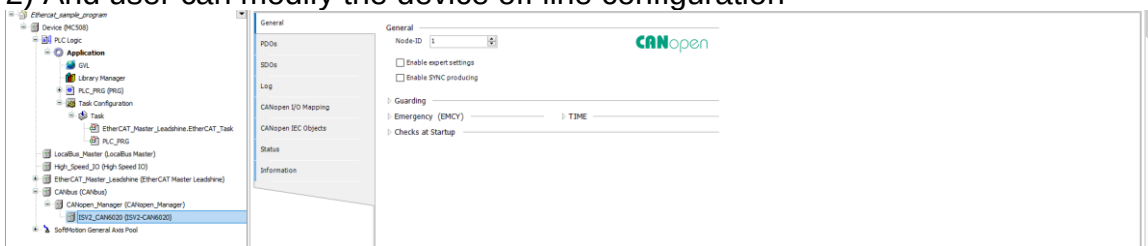
2.4.2.Off-Line Device Configuration

When the device is off line, user can add the device to project manually. (After install the device description file, the device can be found in the list)

- 1) Click “add device”, the device will be added into the project automatically.



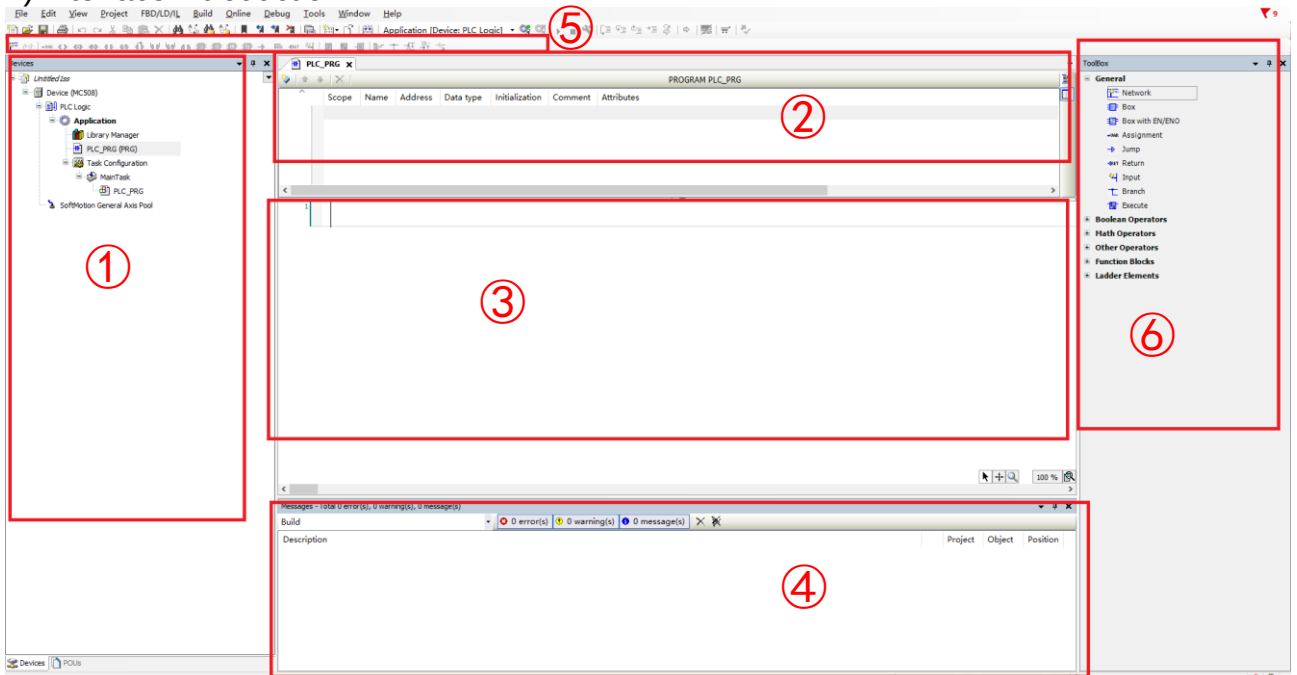
2) And user can modify the device off line configuration



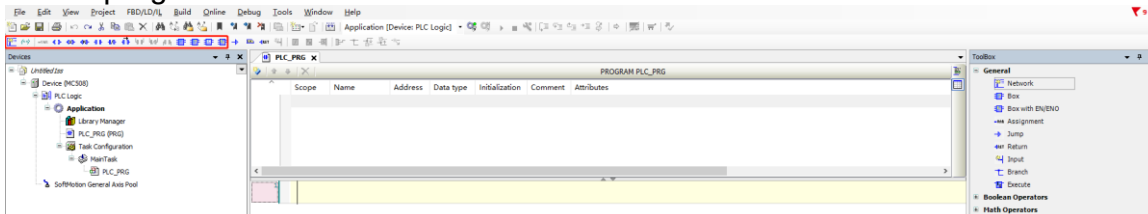
2.5. PLC Sample Program Writing

2.5.1. Create Main Program

1) Interface introduction



- 2) Click the “PLC_PLG”, to open programming interface, click the LD command icon to create program.



- Configure the hardware system based on the hardware connection structure of the medium-sized PLC application system.
- Write the user program according to the control procedure of the application system. During programming, the variables are customized based on the data storage width and use scope, which may be independent of hardware configuration.
- Link the input port variable (I), output status (Q), or value (M) of each hardware port in the system structure with the variables in the user program.
- Configure the synchronization period of network communication (for example, the EtherCAT bus). Configure the execution periods of user program units according to the instantaneity requirements of tasks.
- Login PLC and download the user program, carry out simulated commissioning, and rectify faults until the program runs normally.

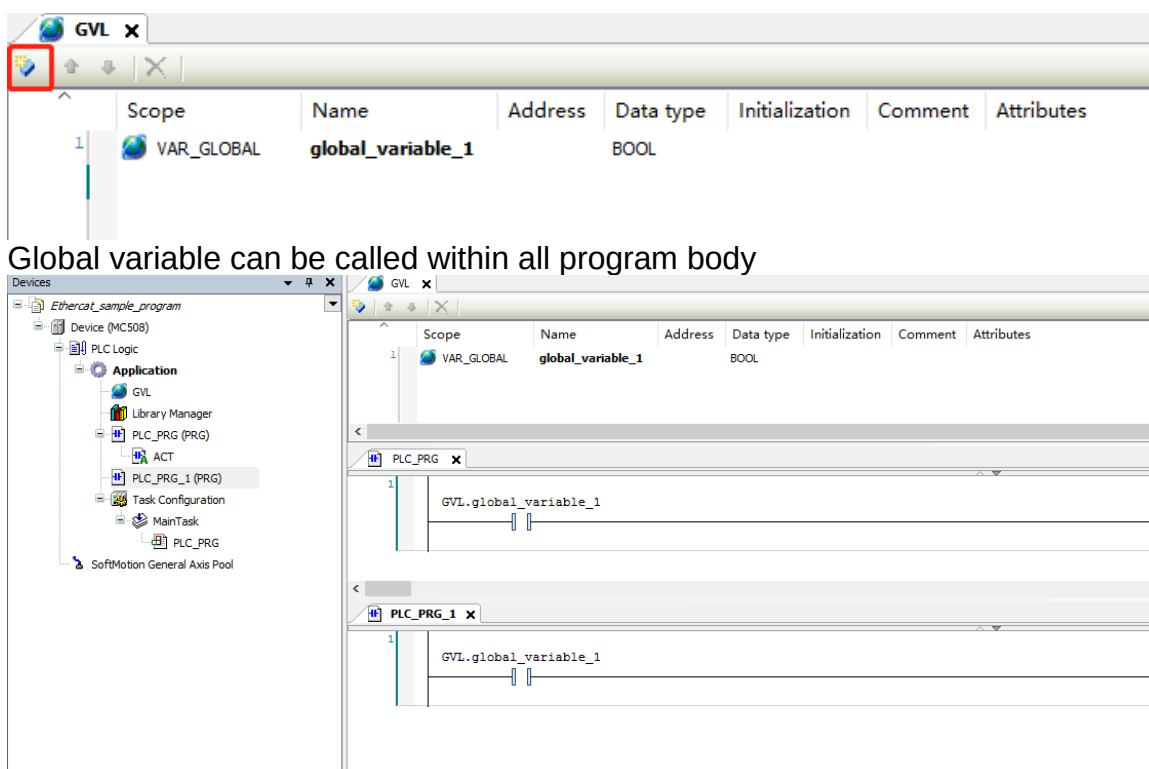
1) Local variable

The screenshot displays the Siemens STEP 7 interface. The top window, titled 'PLC_PRG', shows the 'PROGRAM PLC_PRG' editor. It contains a variable declaration table with the following columns: Scope, Name, Address, Data type, Initialization, Comment, and Attributes. A single entry is shown: 'local_variable_1' of type 'BOOL' in the 'VAR' scope.

Below the declaration table, the ladder logic editor shows a network with a normally open contact labeled 'local_variable_1'. The variable name is highlighted with a dashed border, indicating it is being edited or is the active element.

The bottom window, titled 'PLC_PRG.ACT', shows the active ladder logic program. It displays the same network with the 'local_variable_1' contact, now rendered in a standard ladder logic format with two parallel vertical lines representing the contact.

The variable which be defined in the global variable list is the global variable. right-click “application”, create the “GVL” (global variable list), click “add” icon to create a global variable



2.5.3.Variable address definition

Definition format : <%address area> <prefix symbol> <number>. <number>

For example: %MX0.0, %IX0.0

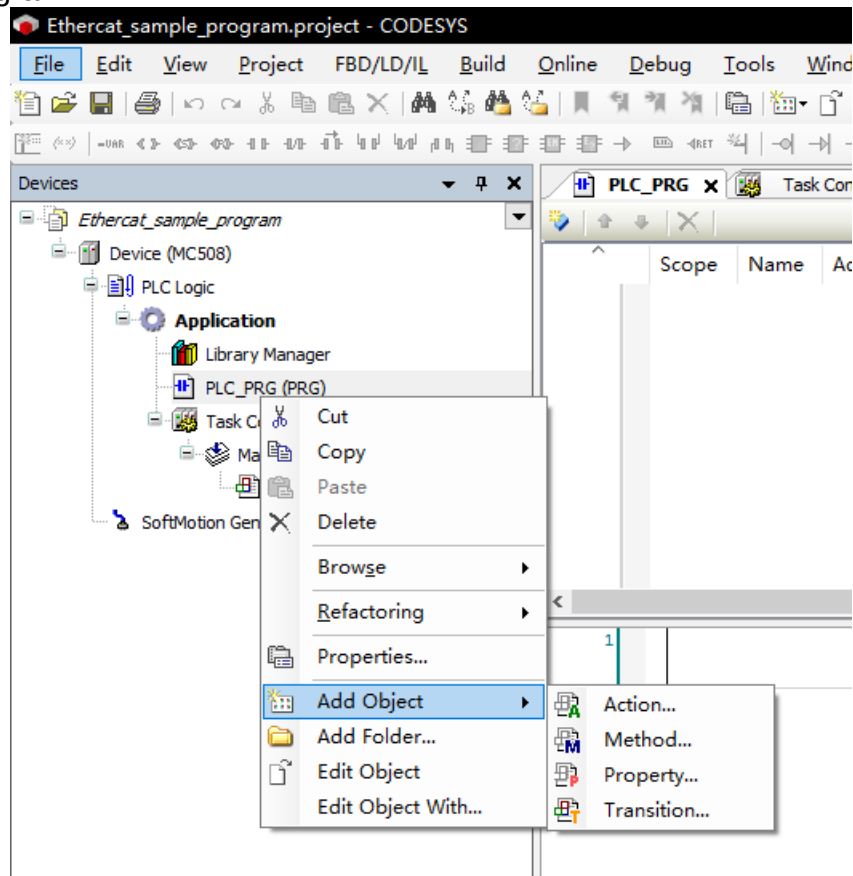
Prefix symbol	Definition	Data type
X	BIT	BOOL
B	BYTE	BYTE
W	WORD	WORD
D	DWORD	DWORD

2.5.4.Variable type

Data type	Data size	Range
BIT	1 bit	0 or 1
BOOL	1Byte	FALSE(0) or TRUE(1)
BYTE	8bit	0~255
WORD	16bit	0~65535
DWORD	32bit	0~4294967295
LWORD	64bit	0~(2 ⁶⁴ -1)
SINT	8bit	-128~127
USINT	8bit	0~255
INT	16bit	-32768~32767
UINT	16bit	0~65535
DINT	32bit	-2147483648~2147483647
UDINT	32bit	0~4294967295
LINT	64bit	-263~(2 ⁶³ -1)
REAL	32bit	1.175494351e ⁻³⁸ ~3.402823466e ⁺³⁸
LREAL	64bit	2.2250738585072014e ⁻³⁰⁸ ~1.7976931348623158e ⁺³⁰⁸
STRING	8×N bit	
WSRING	16×N bit	
TIME	32bit	T#0ms~T#71582m47s295ms
TIME_OF_DAY		TOD#0:0:0~TOD#1193:02:47.295
DATE		D#1970-1-1~D#2106-02-06
DATE_AND_TIME		DT#1970-1-1-0:0:0~DT#2106-02-06-06: 28:15

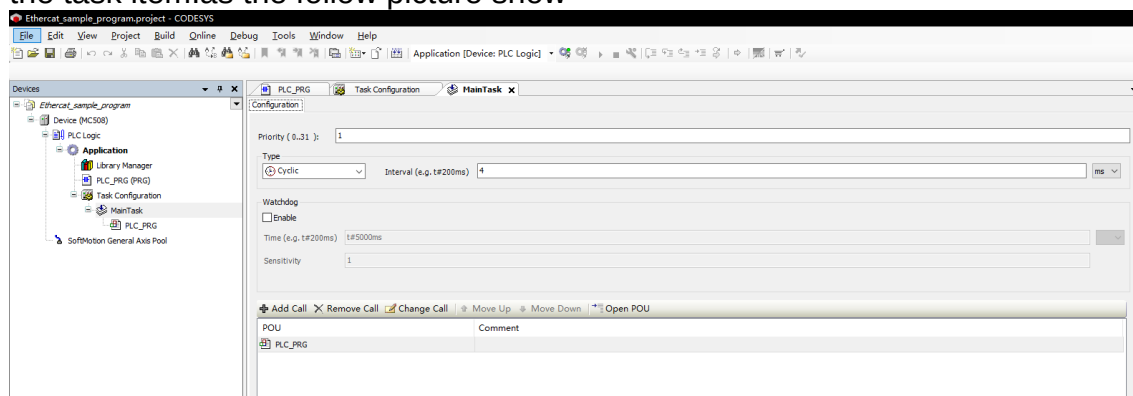
2.5.5.Create Action Program

Right-click the “PLC_PRG” to add the action program, it’s as the subroutine of the PLC_PRG program



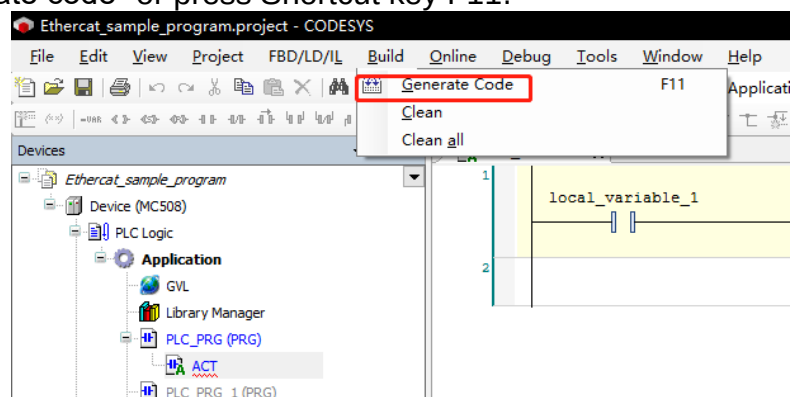
2.5.6.Task Configuration

Double click “MainTask” user can configure the priority, task type, interval, watch dog, and the task item.as the follow picture show

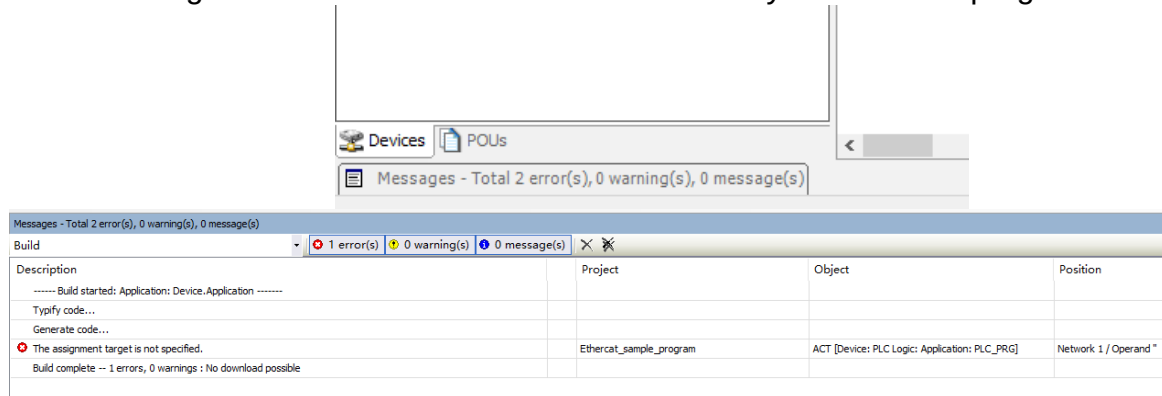


2.5.7.Locate Program Compiling Error

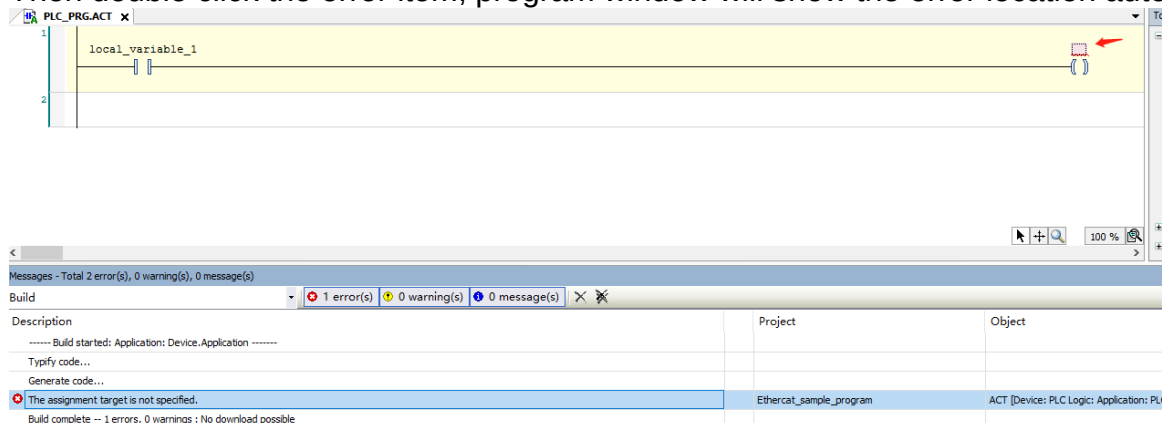
After complete all the program , need to compile the program to check the error,click “build” then click “generate code” or press Shortcut key F11.



Then message window will note whether there are any errors in the program



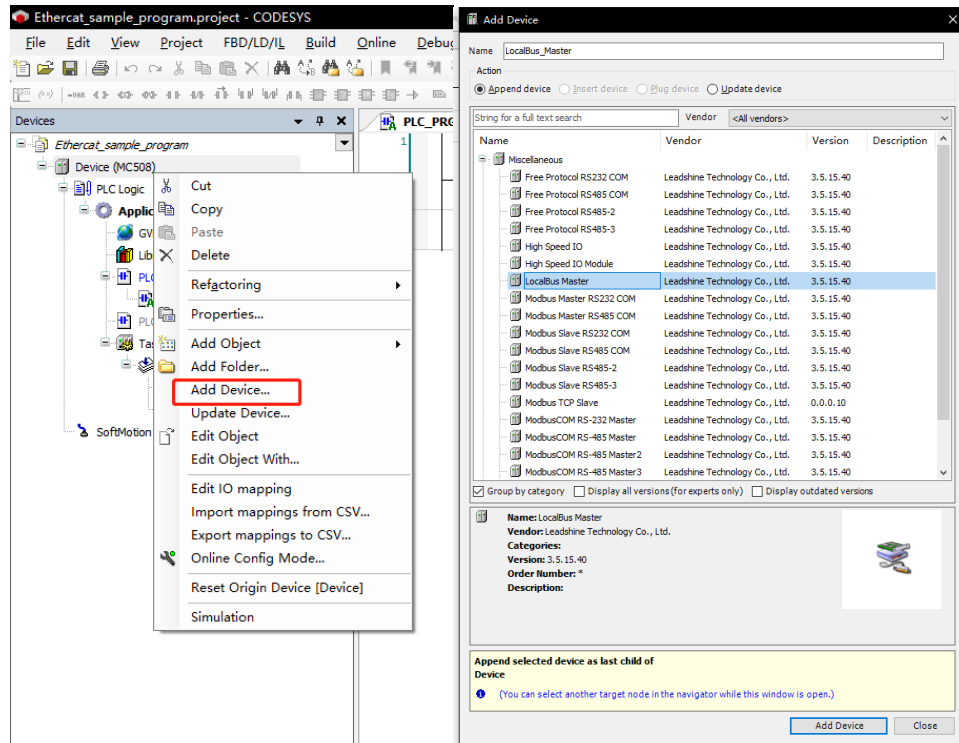
Then double click the error item, program window will show the error location automatically



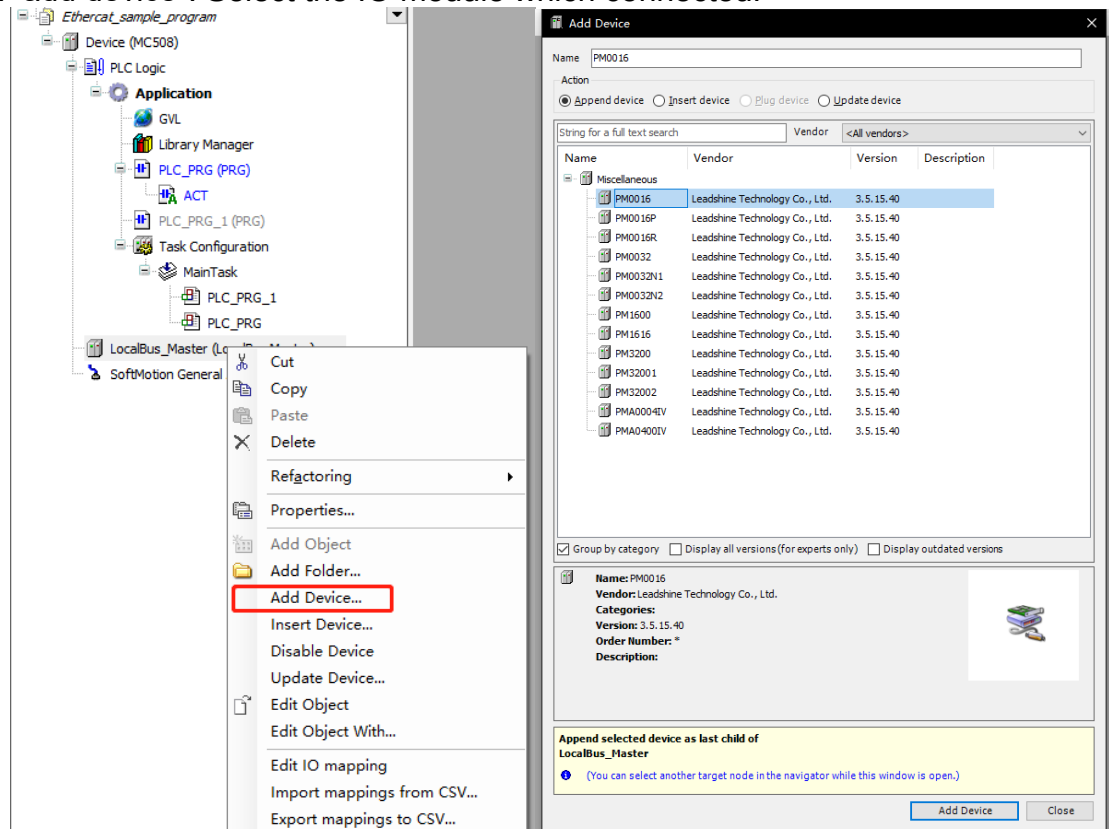
Chapter 3 Local Module

MC500 series PLC right side can extend maximum 32 IO modules (specific model please refer to the IO module catalog), after create the new project, have to add the local bus device before add the module.

1) Right-click the “Device”, click “add device”, then select the “local bus master”, click add device.



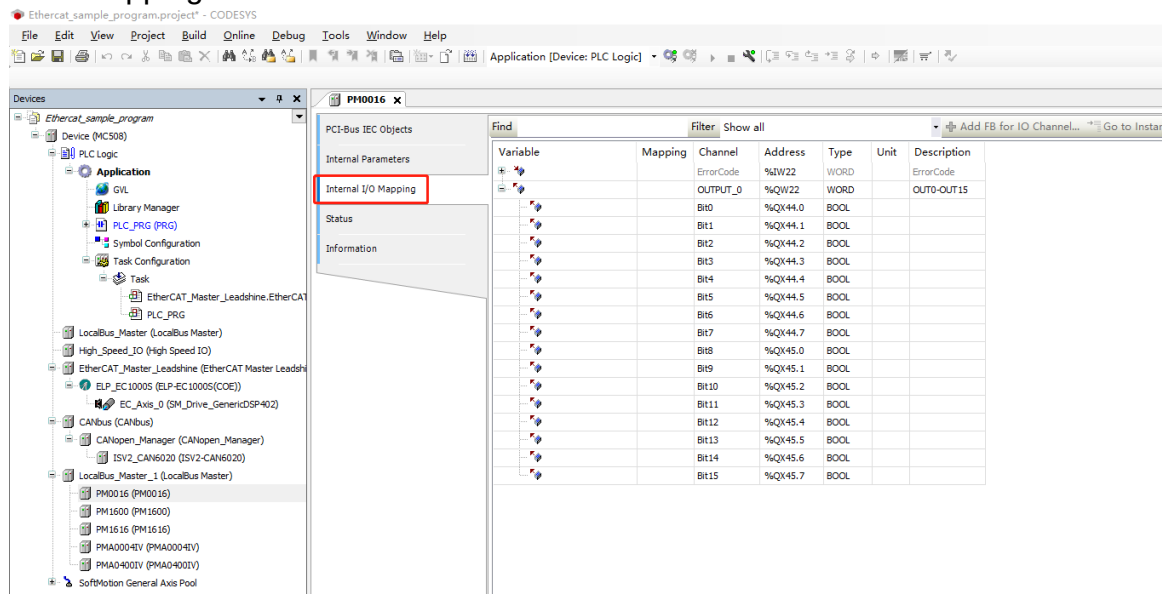
2) Then the device configuration area will show the local bus master, right-click it then select “add device”. Select the IO module which connected.



3.1. Local Module Configuration

3.1.1. Digital module configuration

After add the digital modules, mapping the inputs or outputs. Select “internal I/O mapping”, click mapping icon to select the variable.



3.1.2. Analog module configuration

After add the modules, user need to configure the outputs or inputs type, click the module, select the internal parameters

1) Analog input module

Modify the “AD*config”, “Data_Buffer” to select the different input type, the value details please refer to the below table.

Data_Buffer value	Analog input type	Internal variable range
0	-5~5V,	-32000~32000
1	1~5V,	0~32000
2	-10~+10V	-32000~32000
3	0~10V	0~32000
4	0~20mA	0~32000
5	4~20mA	0~32000
6	0~5V	0~32000
7	-20mA~20mA	-32000~32000

Parameter	Type	Value	Default Value	Unit	Description
LocalBusSlave Info					
SlaveAddr	BYTE	0	0		
SlaveNodeID	dwor	1627389989	1627389989		
AD0 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#01	16#01		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		
AD1 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#02	16#02		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		
AD2 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#03	16#03		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		
AD3 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#04	16#04		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		

Modify the “AD*filter config”, “Data_Buffer” to set the value of the filtering parameter

AD0 filter config				
Index	UINT	16#8001	16#8001	
SubIndex	UINT	16#01	16#01	
Data_Buffer	UINT	4	4	
Data_Size	UINT	1	1	
AD1 filter config				
Index	UINT	16#8001	16#8001	
SubIndex	UINT	16#02	16#02	
Data_Buffer	UINT	4	4	
Data_Size	UINT	1	1	
AD2 filter config				
Index	UINT	16#8001	16#8001	
SubIndex	UINT	16#03	16#03	
Data_Buffer	UINT	4	4	
Data_Size	UINT	1	1	
AD3 filter config				
Index	UINT	16#8001	16#8001	
SubIndex	UINT	16#04	16#04	
Data_Buffer	UINT	4	4	
Data_Size	UINT	1	1	

1) Analog output module

Modify the “DA*config”, “Data_Buffer” to select the different output type, the value details please refer to the below table.

Data_Buffer value	Analog output type	Internal variable range
0	0~5V	0~32000
1	1~5V,	0~32000
2	-5~5V,	-32000~32000
3	0~10V	0~32000
4	-10~+10V	-32000~32000
5	0~20mA	0~32000
6	4~20mA	0~32000

Parameter	Type	Value	Default Value	Unit	Description
LocalBusSlave Info					
SlaveAddr	BYTE	0	0		
SlaveNodeID	dword	1627390469	1627390469		
DA0 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#01	16#01		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		
DA1 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#02	16#02		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		
DA2 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#03	16#03		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		
DA3 config					
Index	UINT	16#8000	16#8000		
SubIndex	UINT	16#04	16#04		
Data_Buffer	UINT	0	0		
Data_Size	UINT	1	1		

And modify the “DA*en”, “Data_Buffer” to select whether enable the output channel. Data_Buffer=1 enable the channel, Data_Buffer=0 disable the channel.

PCI-Bus IEC Objects	Parameter	Type	Value	Default Value	Unit	Description
Internal Parameters	DA0 en					
Internal I/O Mapping	Index	UINT	16#8001	16#8001		
Status	SubIndex	UINT	16#01	16#01		
	Data_Buffer	UINT	1	1		
	Data_Size	UINT	1	1		

And modify the “DA*state when link lost”, “Data_Buffer” to select the output state when the module lose link. The details please refer to the below table.

Data_Buffer value	Channel state
0	Keep previous status
1	Reset status
2	Output the preset value

PCI-Bus IEC Objects	Parameter	Type	Value	Default Value	Unit	Description
Internal Parameters	DA0 state when link lost					
Internal I/O Mapping	Index	UINT	16#8002	16#8002		
Status	SubIndex	UINT	16#01	16#01		
	Data_Buffer	UINT	0	0		
	Data_Size	UINT	1	1		

The “Output the preset value” can be modified on the “DA*value when link lost”, “Data_Buffer”.

For example: output type is “0~10V” the voltage output, set the Data_Buffer value = 27200, when the module lost link, the channel will keep 8.50V voltage output.

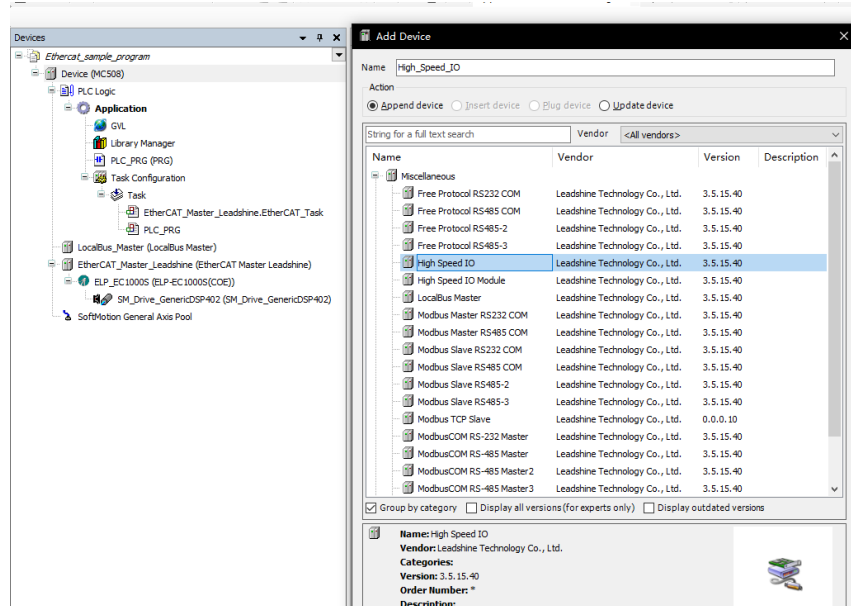
DA0 value when link lost	Index	UINT	16#8003	16#8003		
	SubIndex	UINT	16#01	16#01		
	Data_Buffer	UINT	12000	0		
	Data_Size	UINT	2	2		
DA1 value when link lost	Index	UINT	16#8003	16#8003		
	SubIndex	UINT	16#02	16#02		
	Data_Buffer	UINT	0	0		
	Data_Size	UINT	2	2		
DA2 value when link lost	Index	UINT	16#8003	16#8003		
	SubIndex	UINT	16#03	16#03		
	Data_Buffer	UINT	6400	0		
	Data_Size	UINT	2	2		
DA3 value when link lost	Index	UINT	16#8003	16#8003		
	SubIndex	UINT	16#04	16#04		
	Data_Buffer	UINT	27200	0		
	Data_Size	UINT	2	2		

Chapter 4 High Speed Counter

MC500 series PLC support 6 channels high speed counter inputs, please refer to the follow contents to configure the parameters.

4.1.Add high speed device

Right-click the “Device”, click “add device”, then select the “High Speed IO”, click add device.



4.2.Configure the high speed counter parameters

Double-click “High Speed IO”, into the “internal parameters” interface

1) Enable the high speed counter, modify “High In IO Mode 0”=196611 to enable the counter0, when need user need to enable other counters, please modify the value of “High In IO Mode”.

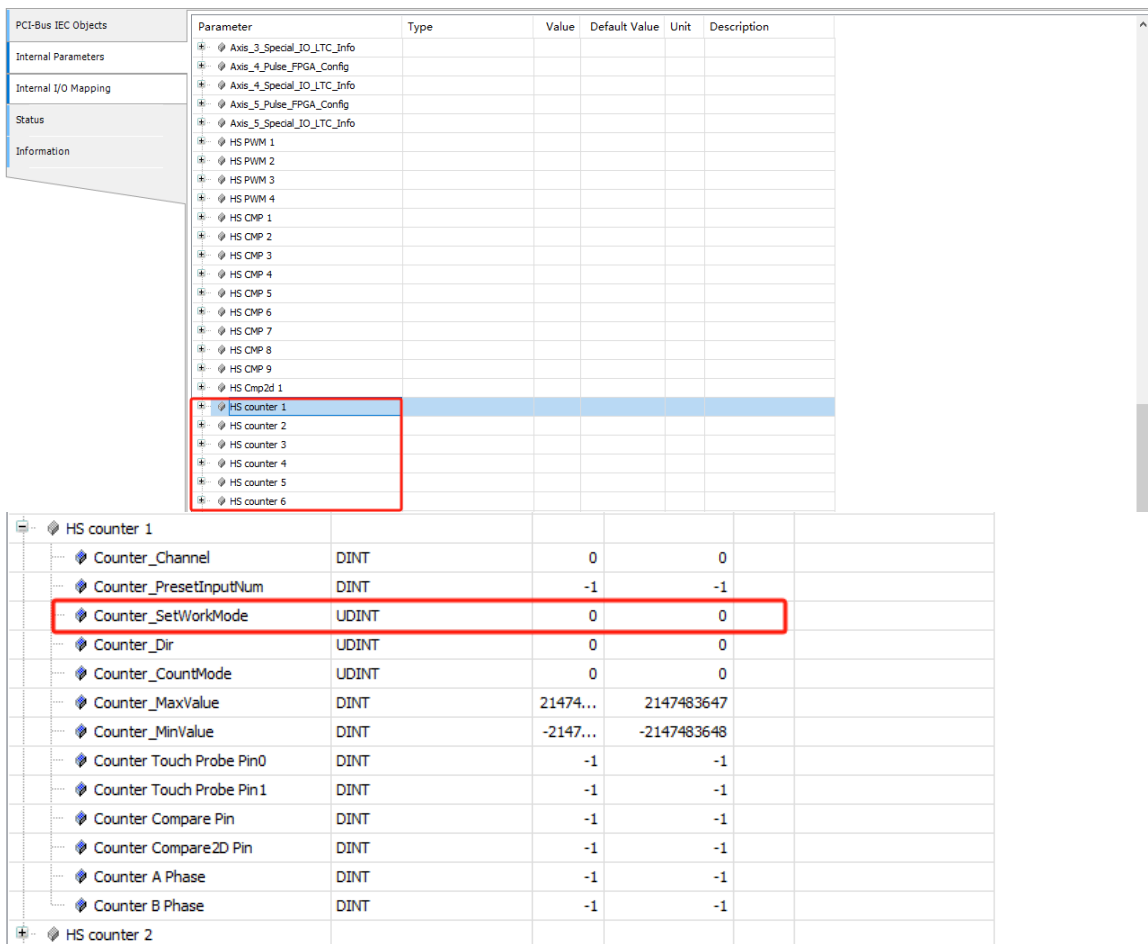
Parameter	Type	Value	Default Value	Unit	Description
Vendor	STRING	'Leadshine Technology Co., Ltd.'	'Leadshine Technology Co., Ltd.'		Vendor of the device
Model Name	STRING	'High Speed IO'	'High Speed IO'		Description of the Device
HS_IO_Mode_0					
High In IO Mode 0	DINT	196611	0		
High Out IO Mode 0	DINT	0	0		
High In IO Mode 1	DINT	0	0		
High Out IO Mode 1	DINT	0	0		
High In IO Mode 2	DINT	0	0		
High Out IO Mode 2	DINT	0	0		
High In IO Mode 3	DINT	0	0		
High Out IO Mode 3	DINT	0	0		
High In IO Mode 4	DINT	0	0		
High Out IO Mode 4	DINT	0	0		
High In IO Mode 5	DINT	0	0		
High Out IO Mode 5	DINT	0	0		
High In IO Mode 6	DINT	0	0		
High Out IO Mode 6	DINT	0	0		
High In IO Mode 7	DINT	0	0		
High Out IO Mode 7	DINT	0	0		
HS_IO_Mode_1					
Axis_Name_0	STRING	'LS_Axis_0'	'LS_Axis_0'		
Axis_Name_1	STRING	'LS_Axis_1'	'LS_Axis_1'		
Axis_Name_2	STRING	'LS_Axis_2'	'LS_Axis_2'		
Axis_Name_3	STRING	'LS_Axis_3'	'LS_Axis_3'		
Axis_Name_4	STRING	'LS_Axis_4'	'LS_Axis_4'		
Axis_Name_5	STRING	'LS_Axis_5'	'LS_Axis_5'		
Axis_Mask	UINT	1	0		
Latch_Mask	UINT	0	0		
Counter_Mask	UINT	1	0		
Cmp_Mask	UDINT	0	0		
Pwm_Mask	UDINT	0	0		
Ioin_Mask	DWORD	3	0		
Ioout_Mask	DWORD	3	0		
Axis_0_Pulse_FPGA_Config					
Axis_0_Special_IO_LTC_Info					
Axis_1_Pulse_FPGA_Config					
Axis_1_Special_IO_LTC_Info					
Axis_2_Pulse_FPGA_Config					
Axis_2_Special_IO_LTC_Info					

Then setting “Counter_Mask”=1 and “Ioin_Mask” =3, (for example: if counter 0 and counter 3 have be enabled, “Counter_Mask”=1+8=9, and “Ioin_Mask”=3+192=195)
Please refer to the mask value table

Axis	“Counter_Mask” value	“Ioin_Mask” value
Counter0	$2^0=1$	$2^0+2^1=3$
Counter 1	$2^1=2$	$2^2+2^3=12$
Counter 2	$2^2=4$	$2^4+2^5=48$
Counter 3	$2^3=8$	$2^6+2^7=192$
Counter 4	$2^4=16$	$2^8+2^9=768$
Counter 5	$2^5=32$	$2^{10}+2^{11}=3072$

2) Setting the counter working mode, please find the “HS counter 1” option in the internal parameters interface, modify the “Counter_SetWorkMode” value to set the count mode, please refer to the follow table.

Count Mode	Value
A/B phase (quadruple)	0
Pulse+direction	1
Single phase	2
CW/CCW	3
A/B phase	4
A/B phase(double)	5



The screenshot shows the 'Internal Parameters' window in the PLC programming software. The left sidebar lists various object categories, and the main area displays a list of parameters for 'HS counter 1'. The 'Counter_SetWorkMode' parameter is highlighted with a red box. Below this, a detailed view of the 'HS counter 1' parameters is shown, with 'Counter_SetWorkMode' also highlighted by a red box.

Parameter	Type	Value	Default Value	Unit	Description
Axis_3_Special_IO_LTC_Info					
Axis_4_Pulse_FPGA_Config					
Axis_4_Special_IO_LTC_Info					
Axis_5_Pulse_FPGA_Config					
Axis_5_Special_IO_LTC_Info					
HS PWM 1					
HS PWM 2					
HS PWM 3					
HS PWM 4					
HS CMP 1					
HS CMP 2					
HS CMP 3					
HS CMP 4					
HS CMP 5					
HS CMP 6					
HS CMP 7					
HS CMP 8					
HS CMP 9					
HS Cmp2d 1					
HS counter 1					
HS counter 2					
HS counter 3					
HS counter 4					
HS counter 5					
HS counter 6					

Parameter	Type	Value	Default Value	Unit	Description
Counter_Channel	DINT	0	0		
Counter_PresetInputNum	DINT	-1	-1		
Counter_SetWorkMode	UDINT	0	0		
Counter_Dir	UDINT	0	0		
Counter_CountMode	UDINT	0	0		
Counter_MaxValue	DINT	21474...	2147483647		
Counter_MinValue	DINT	-2147...	-2147483648		
Counter Touch Probe Pin0	DINT	-1	-1		
Counter Touch Probe Pin1	DINT	-1	-1		
Counter Compare Pin	DINT	-1	-1		
Counter Compare2D Pin	DINT	-1	-1		
Counter A Phase	DINT	-1	-1		
Counter B Phase	DINT	-1	-1		

3) Setting the counting direction

Value =0 is Positive, -1 is negative

Counter_Channel	DINT	0	0
Counter_PresetInputNum	DINT	-1	-1
Counter_SetWorkMode	UDINT	0	0
Counter_Dir	UDINT	0	0
Counter_CountMode	UDINT	0	0
Counter_MaxValue	DINT	21474...	2147483647
Counter_MinValue	DINT	-2147...	-2147483648
Counter Touch Probe Pin0	DINT	-1	-1
Counter Touch Probe Pin1	DINT	-1	-1
Counter Compare Pin	DINT	-1	-1
Counter Compare2D Pin	DINT	-1	-1
Counter A Phase	DINT	-1	-1
Counter B Phase	DINT	-1	-1

4) Setting counting mode

Value=0 is linear counting, -1 is circular counting

Counter_Channel	DINT	0	0
Counter_PresetInputNum	DINT	-1	-1
Counter_SetWorkMode	UDINT	0	0
Counter_Dir	UDINT	0	0
Counter_CountMode	UDINT	0	0
Counter_MaxValue	DINT	21474...	2147483647
Counter_MinValue	DINT	-2147...	-2147483648
Counter Touch Probe Pin0	DINT	-1	-1
Counter Touch Probe Pin1	DINT	-1	-1
Counter Compare Pin	DINT	-1	-1
Counter Compare2D Pin	DINT	-1	-1
Counter A Phase	DINT	-1	-1
Counter B Phase	DINT	-1	-1

4.3.Setting input filter parameters

Please find the “InputFilter Para” option in the internal parameters interface, modify the “time_In” value to set the filter value.

Parameter	Type	Value	Default Value	Unit	Description
HS LTC 5					
HS LTC 6					
HS LTC 7					
HS LTC 8					
HS LTC 9					
HS Encoder 1					
HS Encoder 2					
HS Encoder 3					
InputFilter Para					
Time_In0	UDINT	2	65535		
Time_In1	UDINT	2	65535		
Time_In2	UDINT	2	65535		
Time_In3	UDINT	2	65535		
Time_In4	UDINT	2	65535		
Time_In5	UDINT	65535	65535		
Time_In6	UDINT	2	65535		
Time_In7	UDINT	2	65535		
Time_In8	UDINT	2	65535		
Time_In9	UDINT	2	65535		
Time_In10	UDINT	2	65535		
Time_In11	UDINT	2	65535		
Time_In12	UDINT	65535	65535		
Time_In13	UDINT	65535	65535		
Time_In14	UDINT	65535	65535		
Time_In15	UDINT	65535	65535		
Time_In16	UDINT	65535	65535		
Time_In17	UDINT	65535	65535		
Time_In18	UDINT	65535	65535		
Time_In19	UDINT	65535	65535		
Time_In20	UDINT	65535	65535		
Time_In21	UDINT	65535	65535		
Time_In22	UDINT	65535	65535		
Time_In23	UDINT	65535	65535		
Time_In24	UDINT	65535	65535		
Time_In25	UDINT	65535	65535		
Time_In26	UDINT	65535	65535		
Time_In27	UDINT	65535	65535		
Time_In28	UDINT	65535	65535		
Time_In29	UDINT	65535	65535		

Chapter 5 High Speed Outputs

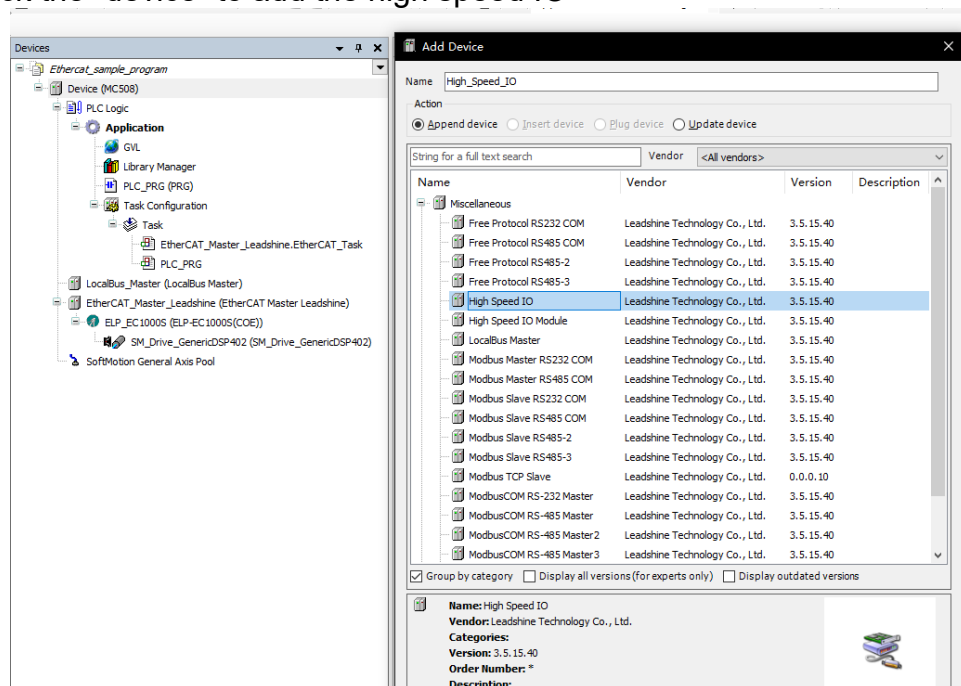
5.1. Local High Speed Pulse Axis Configuration

Please refer to the follow table, 6 high speed pulse axis are configured at the output 0~11, output maximum frequency is 200KHz.

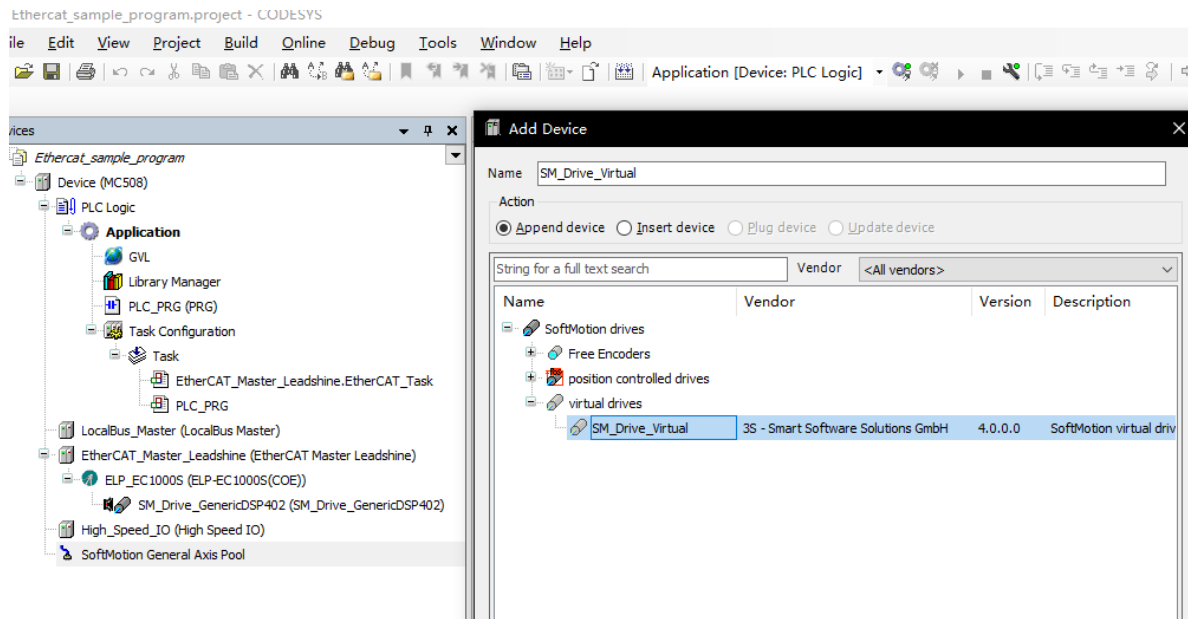
Label	Definition	Pulse No	Label	Definition	Pulse No
SS0	Input common terminal		SS1	Input common terminal	
0	High speed input		8	High speed input	
1	High speed input		9	High speed input	
2	High speed input		10	High speed input	
3	High speed input		11	High speed input	
4	High speed input		12	Normal input	
5	High speed input		13	Normal input	
6	High speed input		14	Normal input	
7	High speed input		15	Normal input	
COM	Output common terminal		COM	Output common terminal	
0	High speed output	AXIS 0 PUL	8	High speed output	AXIS 4 PUL
1	High speed output	AXIS 0 DIR	9	High speed output	AXIS 4 DIR
2	High speed output	AXIS 1 PUL	10	High speed output	AXIS 5 PUL
3	High speed output	AXIS 1 DIR	11	High speed output	AXIS 5 DIR
4	High speed output	AXIS 2 PUL	12	Normal output	
5	High speed output	AXIS 2 DIR	13	Normal output	
6	High speed output	AXIS 3 PUL	14	Normal output	
7	High speed output	AXIS 3 DIR	15	Normal output	

5.1.1. Axis Configuration Interface

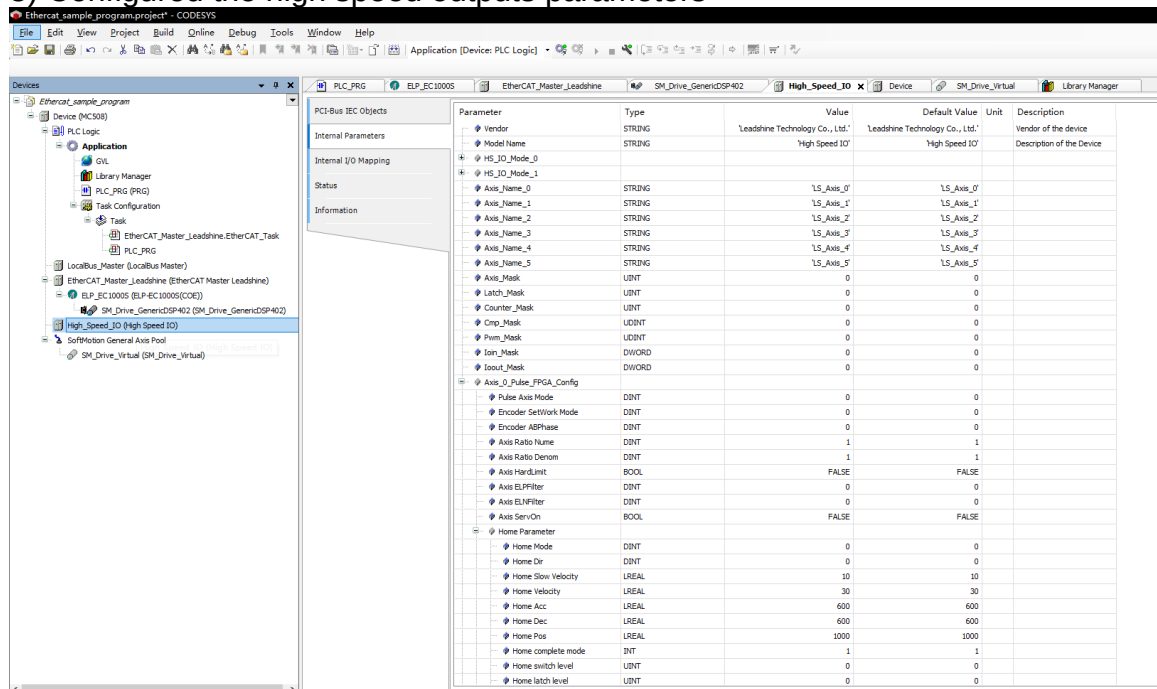
1) Right click the “device” to add the high speed IO



2) Add virtual axis

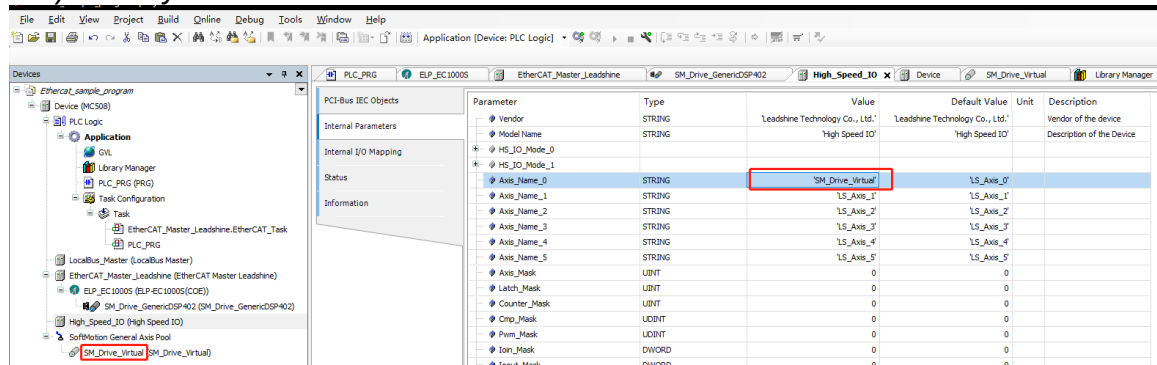


3) Configured the high speed outputs parameters



For example: enable axis 0 to control the stepper motor.

3.1) modify the axis 0 name



3.2) modify "High Out IO Mode 0"=65537, it's means enable axis configuration. Pulse axis0~5 enable switch is "High Out IO Mode 0"~"High Out IO Mode 5"

Parameter	Type	Value	Default Value	Unit	Description
Vendor	STRING	'Leadshine Technology Co., Ltd.'	'Leadshine Technology Co., Ltd.'		Vendor of the device
Model Name	STRING	'High Speed IO'	'High Speed IO'		Description of the Device
HS_IO_Mode_0					
High In IO Mode 0	DINT	0	0		
High Out IO Mode 0	DINT	65537	0		
High In IO Mode 1	DINT	0	0		
High Out IO Mode 1	DINT	0	0		
High In IO Mode 2	DINT	0	0		
High Out IO Mode 2	DINT	0	0		

Then setting “Axis_Mask”=1 and “Ioout_Mask”=3, (for example: if axis 0 and axis 3 have be enabled, “Axis_Mask”=1+8=9, and “Ioout_Mask”=3+192=195)

Please refer to the mask value table

Axis	“Axis_Mask” value	“Ioout_Mask” value
Axis_0	$2^0=1$	$2^0+2^1=3$
Axis_1	$2^1=2$	$2^2+2^3=12$
Axis_2	$2^2=4$	$2^4+2^5=48$
Axis_3	$2^3=8$	$2^6+2^7=192$
Axis_4	$2^4=16$	$2^8+2^9=768$
Axis_5	$2^5=32$	$2^{10}+2^{11}=3072$

Parameter Axis_Mask bit 0~5 match axis 0~5, and Ioout_Mask bit 0~11 match MC500 series PLC output 0~11.

HS_IO_Mode_1				
Axis_Name_0	STRING	'SM_Drive_Virtual'	'LS_Axis_0'	
Axis_Name_1	STRING	'LS_Axis_1'	'LS_Axis_1'	
Axis_Name_2	STRING	'LS_Axis_2'	'LS_Axis_2'	
Axis_Name_3	STRING	'LS_Axis_3'	'LS_Axis_3'	
Axis_Name_4	STRING	'LS_Axis_4'	'LS_Axis_4'	
Axis_Name_5	STRING	'LS_Axis_5'	'LS_Axis_5'	
Axis_Mask	UINT	1	0	
Latch_Mask	UINT	0	0	
Counter_Mask	UINT	0	0	
Cmp_Mask	UDINT	0	0	
Pwm_Mask	UDINT	0	0	
IoIn_Mask	DWORD	0	0	
Ioout_Mask	DWORD	3	0	

3.3) modify axis main parameters, “Axis_0_Pluse_FPGA_Config”~

“Axis_5_Pluse_FPGA_Config” include axis 0~5 main motion parameters.

Axis_0_Pluse_FPGA_Config				
Pulse Axis Mode	DINT	0	0	
Encoder SetWork Mode	DINT	0	0	
Encoder ABPhase	DINT	0	0	
Axis Ratio Num	DINT	1	1	
Axis Ratio Denom	DINT	1	1	
Axis HardLimit	BOOL	FALSE	FALSE	
Axis ELPFilter	DINT	0	0	
Axis ELNFilter	DINT	0	0	
Axis ServOn	BOOL	FALSE	FALSE	
Home Parameter				
Latch Parameter				
EZ Clear				
Axis_0_Special_IO_LTC_Info				
Axis_1_Pluse_FPGA_Config				
Axis_1_Special_IO_LTC_Info				
Axis_2_Pluse_FPGA_Config				
Axis_2_Special_IO_LTC_Info				
Axis_3_Pluse_FPGA_Config				
Axis_3_Special_IO_LTC_Info				
Axis_4_Pluse_FPGA_Config				
Axis_4_Special_IO_LTC_Info				
Axis_5_Pluse_FPGA_Config				
Axis_5_Special_IO_LTC_Info				

3.3.1) “Pulse Axis Mode”, please refer to the follow table to select the mode

Output mode	Value
Pulse high + Direction high	0
Pulse low + Direction high	1
Pulse high + Direction low	2
Pulse low + Direction low	3
Dual pulse high	4
Dual pulse low	5
A B phase	6

Axis_0_Pulse_FPGA_Config					
Pulse Axis Mode	DINT	0	0		
Encoder SetWork Mode	DINT	0	0		
Encoder ABPhase	DINT	0	0		
Axis Ratio Num	DINT	1	1		
Axis Ratio Denom	DINT	1	1		
Axis HardLimit	BOOL	FALSE	FALSE		
Axis ELPFilter	DINT	0	0		
Axis ELNFilter	DINT	0	0		
Axis ServOn	BOOL	FALSE	FALSE		

3.3.2) pulse equivalent, modify parameter Axis Ratio Num and Axis Ratio Denom to change the pulse equivalent.

Axis_0_Pulse_FPGA_Config					
Pulse Axis Mode	DINT	0	0		
Encoder SetWork Mode	DINT	0	0		
Encoder ABPhase	DINT	0	0		
Axis Ratio Num	DINT	1	1		
Axis Ratio Denom	DINT	1	1		
Axis HardLimit	BOOL	FALSE	FALSE		
Axis ELPFilter	DINT	0	0		
Axis ELNFilter	DINT	0	0		
Axis ServOn	BOOL	FALSE	FALSE		

3.3.3) homing parameters

Home Parameter					
Home Mode	DINT	0	0		
Home Dir	DINT	0	0		
Home Slow Velocity	LREAL	10	10		
Home Velocity	LREAL	30	30		
Home Acc	LREAL	600	600		
Home Dec	LREAL	600	600		
Home Pos	LREAL	1000	1000		
Home complete mode	INT	1	1		
Home switch level	UINT	0	0		
Home latch level	UINT	0	0		
EZ latch level	UINT	0	0		
Set latch pos type	UINT	0	0		
HomeElEnable	BOOL	FALSE	FALSE		
HomeElPSwitchLevel	BOOL	TRUE	TRUE		
HomeElNSwitchLevel	BOOL	TRUE	TRUE		
ELStopMode	BOOL	FALSE	FALSE		
DriverCounter	DINT	200	200		
IsRevolveAxis	BOOL	FALSE	FALSE		
RevolveAxisSafeAngel	REAL	90	90		
Home Switch Num	INT	-1	-1		
Limit Switch Num	INT	-1	-1		

① Home mode ,the homing method please refer to the follow table

Home mode	Value
One homing processing	0
One homing processing + reverse find homing signal	1
Two homing processing	2
Mark homing position	3

Home Parameter					
Home Mode	DINT	0	0		
Home Dir	DINT	0	0		
Home Slow Velocity	LREAL	10	10		
Home Velocity	LREAL	30	30		
Home Acc	LREAL	600	600		
Home Dec	LREAL	600	600		
Home Pos	LREAL	1000	1000		
Home complete mode	INT	1	1		
Home switch level	UINT	0	0		
Home latch level	UINT	0	0		
EZ latch level	UINT	0	0		

② Homing signal configuration, MC500 series PLC input 0~5 can be configured as homing detection signal, please refer to the follow table setting the parameter “Home Switch Num” value.

Select input as homing switch	Home Switch Num value
Invalid	-1
IN0	0
IN1	1
IN2	2
IN3	3
IN4	4
IN5	5

EZ latch level	UINT	0	0
Set latch pos type	UINT	0	0
HomeEIEnable	BOOL	FALSE	FALSE
HomeEIPSwitchLevel	BOOL	TRUE	TRUE
HomeEINSwitchlevel	BOOL	TRUE	TRUE
ELStopMode	BOOL	FALSE	FALSE
DriverCounter	DINT	200	200
IsRevolveAxis	BOOL	FALSE	FALSE
RevolveAxisSafeAngel	REAL	90	90
Home Switch Num	INT	0	-1
Limit Switch Num	INT	-1	-1

③ Home switch level, if it's a normally closed signal, set the value to 1 and the normally open signal to 0.

Home Parameter				
Home Mode	DINT	0	0	
Home Dir	DINT	0	0	
Home Slow Velocity	LREAL	10	10	
Home Velocity	LREAL	30	30	
Home Acc	LREAL	600	600	
Home Dec	LREAL	600	600	
Home Pos	LREAL	1000	1000	
Home complete mode	INT	1	1	
Home switch level	UINT	0	0	
Home latch level	UINT	0	0	

④ Limit Switch Num

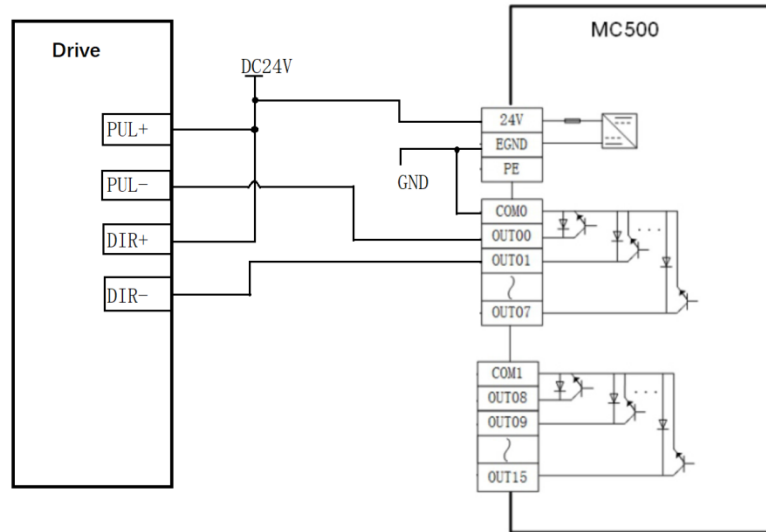
Select input as limit switch	Value
Invalid	-1
IN10	10
IN11	11
IN12	12
IN13	13
IN14	14
IN15	15

⑤ Homing running parameter

Home Parameter				
Home Mode	DINT	0	0	
Home Dir	DINT	0	0	
Home Slow Velocity	LREAL	10	10	
Home Velocity	LREAL	30	30	
Home Acc	LREAL	600	600	
Home Dec	LREAL	600	600	
Home Pos	LREAL	1000	1000	

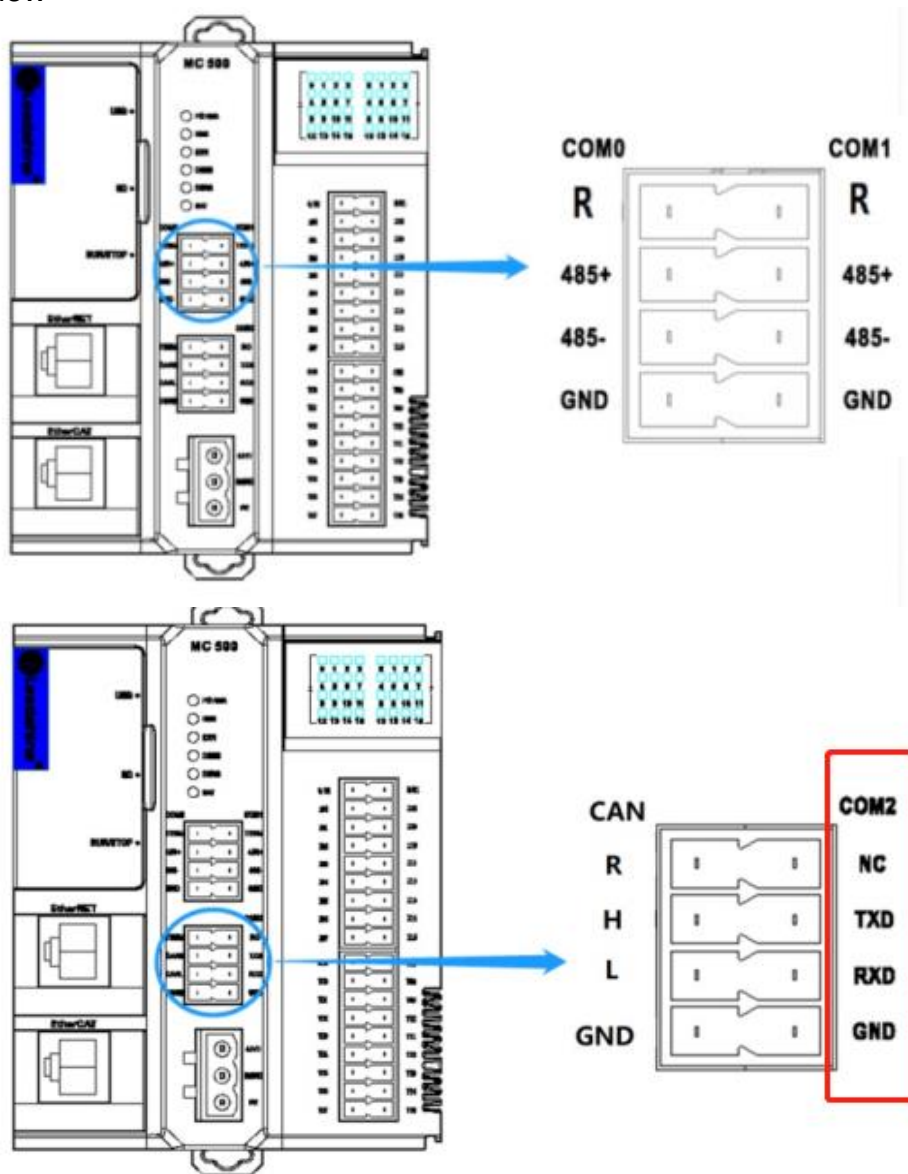
5.1.2.High Speed Output Wiring

MC500 series PLC only support NPN type output, and the maximum output frequency is 200KHz, please refer to the follow picture to connect the drive.



Chapter 6 Serial Port Communication

MC500 series PLC configured 2* RS485 port and 1*RS232 port, as the follow picture show



6.1.RS485 Communication wiring

MC500 series PLC COM0 and COM1 are the RS485 serial port, it can be connected to HMI, frequency converter, or other Modbus-RTU master station device or slave station device.

The communication wiring please refer to the follow picture.



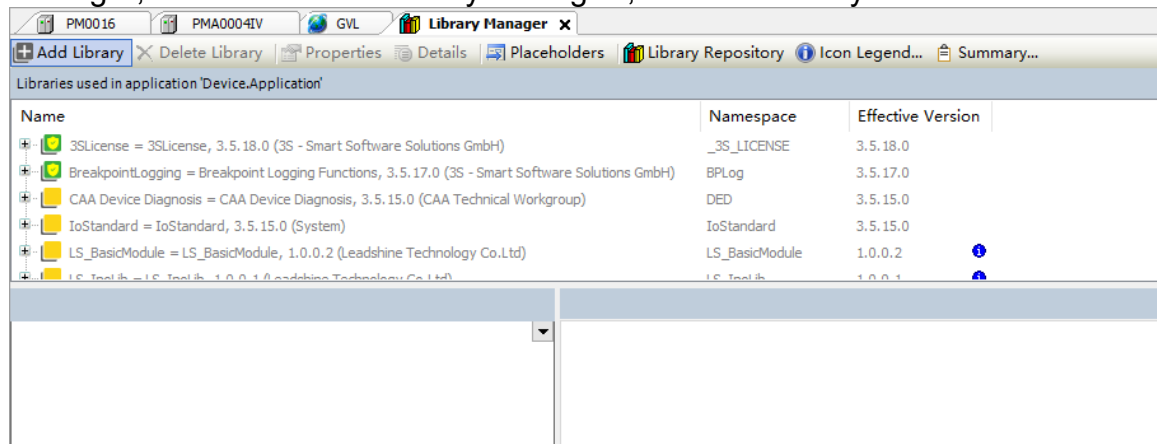
6.2.Modbus-RTU Communication configuration

6.2.1.Modbus Master Station Configuration

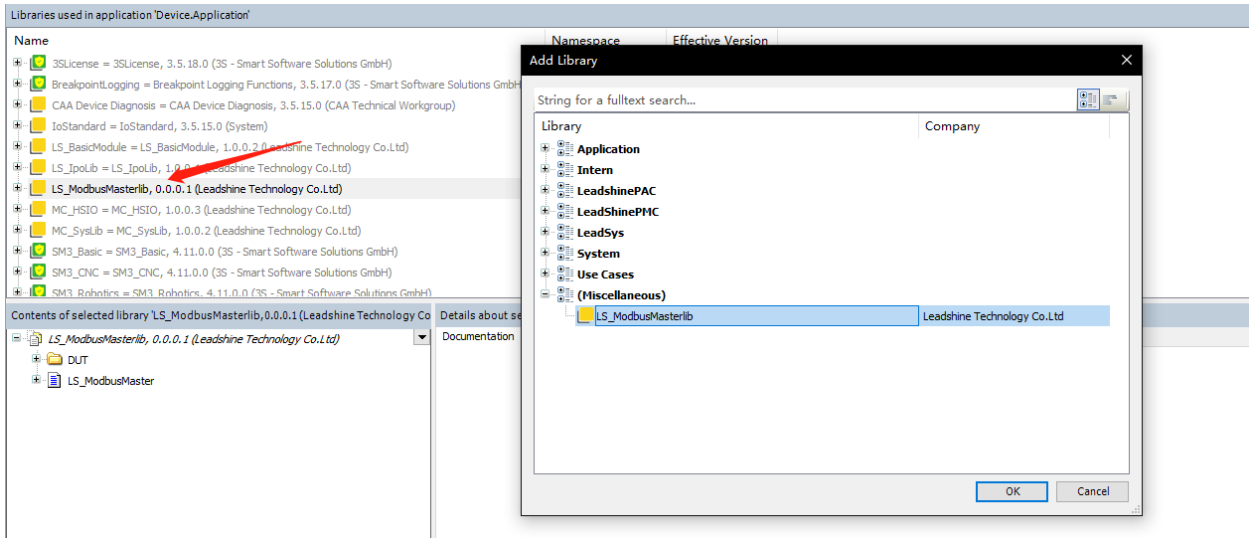
1) Using CODESYS software configure the Modbus-RTU master function will use the special function libraries “ModbusMaster_Eng.compiled-library”

Modbus_master_library	2023/5/15 11:41	文件夹
LS_BasicModule.compiled-library-ge33.compiled-library	2023/4/28 17:17	COMPILED-LIBR...
LS_SysLib.compiled-library-ge33.compiled-library	2023/5/6 18:31	COMPILED-LIBR...
LS_UtilsLib.compiled-library-ge33.compiled-library	2023/5/6 18:31	COMPILED-LIBR...
ModbusMaster_Eng.compiled-library	2023/5/15 9:48	COMPILED-LIBR...
PMC_BasicModule.compiled-library-ge33.compiled-library	2023/4/28 17:17	COMPILED-LIBR...

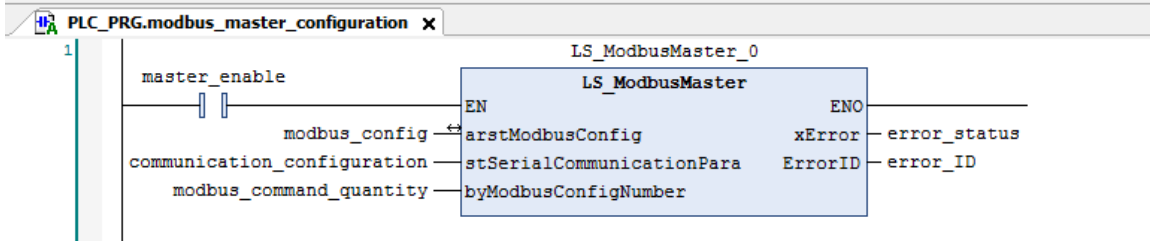
2) After installing the function libraries at the library repository, add it to the library manager, double click the “library manager”, click add library.



3) Find the function library, add to the manager.



4) Create a new subroutine, insert the “LS_ModbusMaster” function box, then define the box input and output variable name.



About the function box description please refer to the follow contents.

Name	Description	Data type	Range	Initialization
arstModbus Config	Modbus communication config	ARRAY[0..127] OF ModbusConfig	-	-
stSerialCommunicationPara	Serial port communication parameters	SerialCommunicationPara	-	-
byModbusConfig Number	Modbus command quantity, maximum 128 configurations	BYTE	0-128	0
xError	Error	BOOL	TRUE/FALSE	FALSE
ErrorID	Error code	ModbusMasterErrorCode	-	0

arstModbus Config (modbus communication configuration structure)

```

usiAddress : USINT :=1 ; //Slave ID
usiFunctionCode : ModbusFunctionCode :=Read_Coils; //Function code
wOffset : WORD :=0; //Address offset
wLength : WORD :=1; //Data length, 01/02/03/04/15/16 function code
xCycle : BOOL :=TRUE; // True:cycle mode, False: trigger mode
uiCycleTime : UINT :=100; //Cycletime, Unit: ms
xTrigger : BOOL ; //Trigger signal, used for trigger mode
sbyRetransmissionNumber : BYTE := 3; //Retransmissions numbers
xError : BOOL := FALSE;
ErrorCode : ModbusMasterErrorCode;
xDone : BOOL := FALSE; //Processing completion signal
arwReadData : ARRAY[0..126] OF WORD; //Read Data Cache, 01/02/03/04 function code
arwWriteData : ARRAY[0..126] OF WORD; //Write Data Cache, 05/06/15/16 function code

```

stSerialCommunicationPara(Serial port communication parameters structure)

```

udiPort : UDINT :=3;// Serial port.3:COM0; 4:COM1
udiBaudrate : UDINT :=115200;//Baud rate
udiPARITY : COM.PARITY :=COM.PARITY.EVEN;//Parity
udiStopBits : COM.STOPBIT :=COM.STOPBIT.ONESTOPBIT;//Stop bit
udiTimeout : UDINT :=1000;//Time out, Unit: ms
udiByteSize : UDINT :=8;//Data bit

```

usiFunctionCode : ModbusFuntionCode :=(modbus function code)

```

Read_Coils := 16#01 , //Read coil
Read_Discrete_Inputs := 16#02 , // Read discrete inputs
Read_Holding_Registers := 16#03 ,//Read holding registers
Read_Input_Registers := 16#04 , // Read input registers
Write_Single_Coil := 16#05 , // Write single coil
Write_Single_Register := 16#06 ,// Write single register
Write_Multiple_Coil := 15 ,//Write multiple coils
Write_Multiple_Register := 16, // Write multiple registers

```

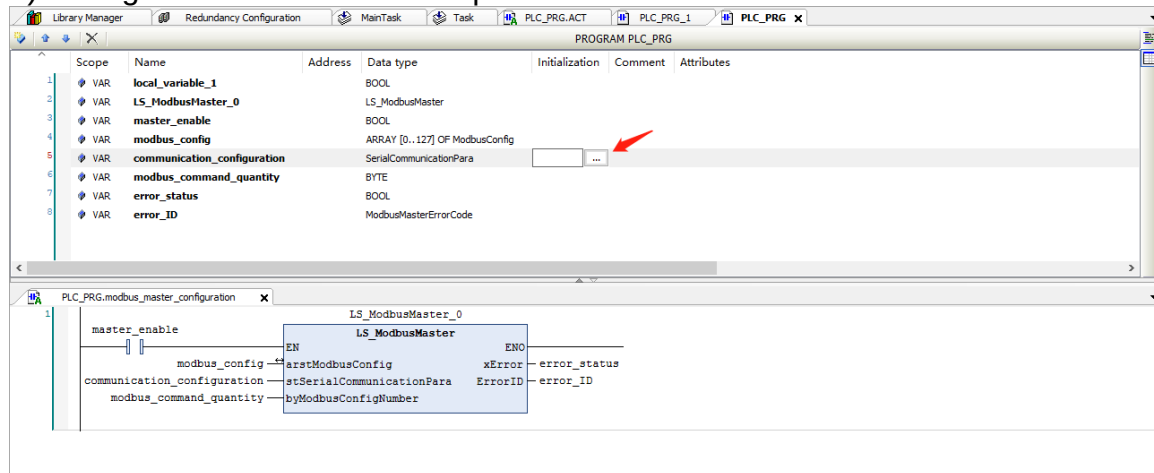
ErrorCode : ModbusMasterErrorCode (modbus master error code)

```

NO_ERROR := 0,
OPEN_SERIAL_ERROR := 100,
NOT_SUPPORT_FUNCTIONCODE := 200,
INVALID_DATA_ADDRESS := 300,
INVALID_DATA_VALUE := 400,
SLAVE_ERROR :=500,
CRC_ERROR :=600,
INVALID_DATA_LENGTH := 700,
TIME_OUT := 800,
INVALID_DEVICE := 16#FFFF

```

5) Configure the communication parameters initialization.

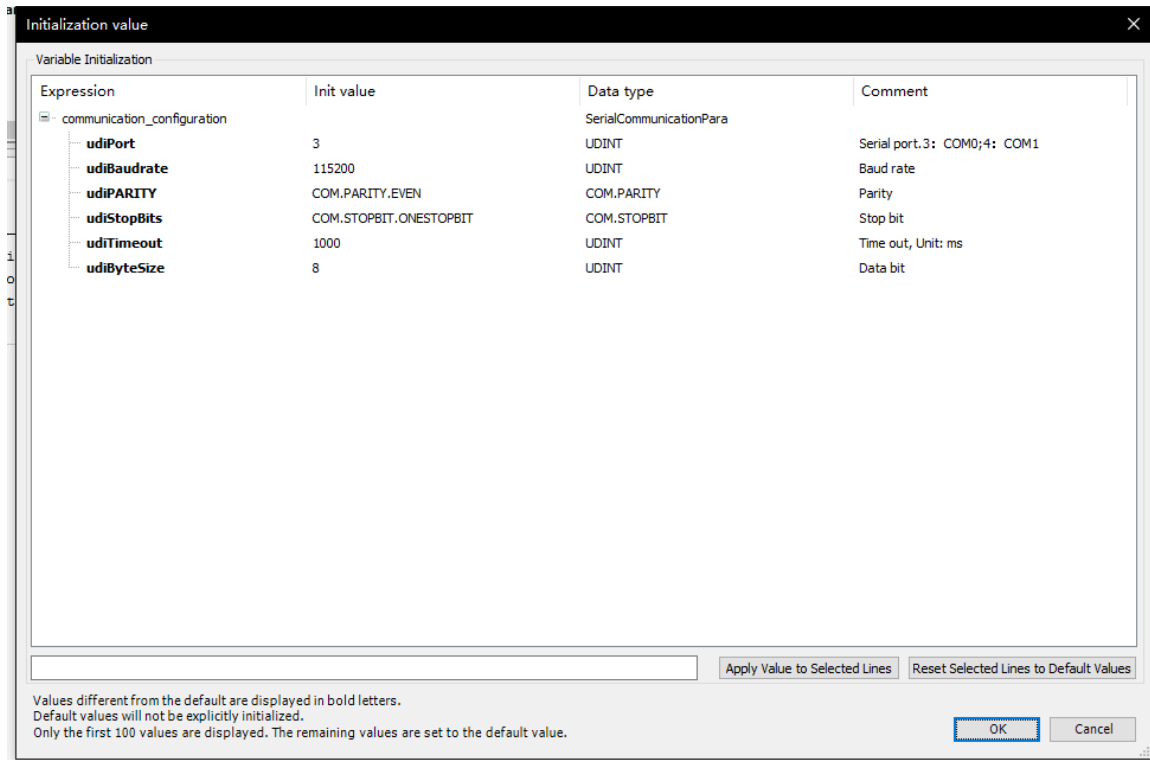


The screenshot displays the PLC programming software interface. The top window shows the 'PROGRAM PLC_PRG' with a variable declaration table. The bottom window shows the 'PLC_PRG.modbus_master_configuration' block diagram.

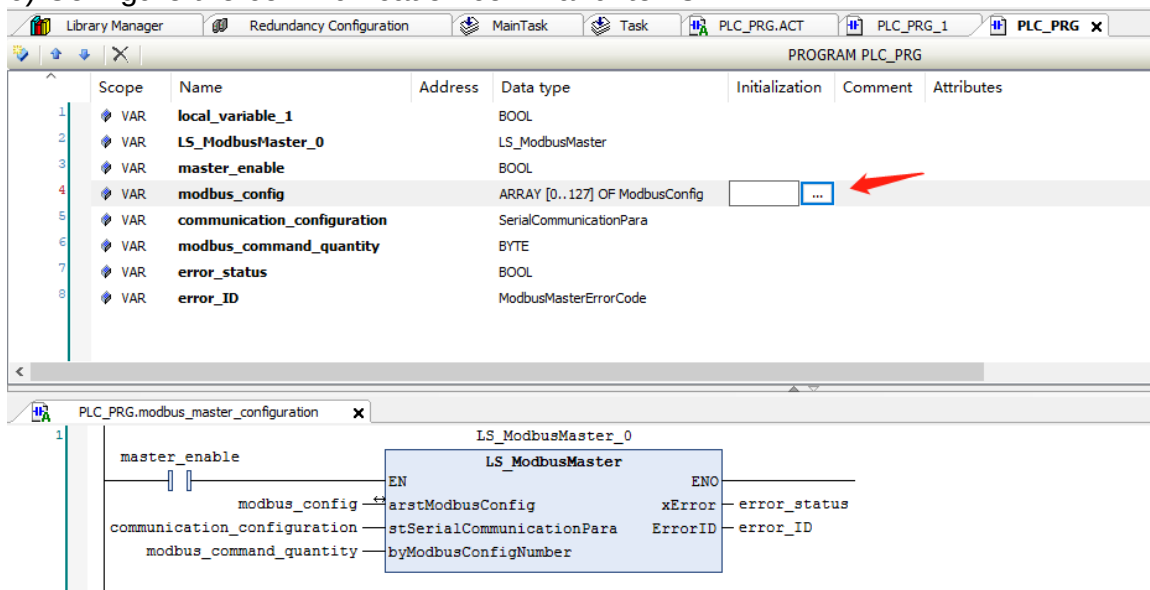
Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR		local_variable_1			
2	VAR		LS_ModbusMaster_0			
3	VAR		master_enable			
4	VAR		modbus_config			
5	VAR		communication_configuration			
6	VAR		modbus_command_quantity			
7	VAR		error_status			
8	VAR		error_ID			

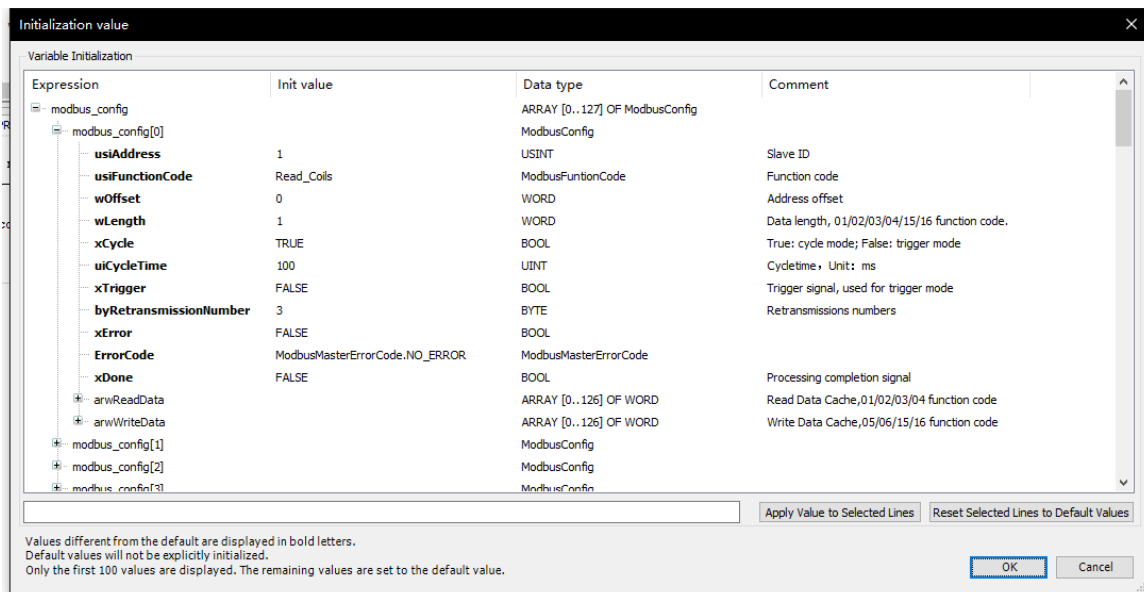
The bottom window shows the 'LS_ModbusMaster_0' block diagram. The block is named 'LS_ModbusMaster' and has the following connections:

- master_enable** (Bool) connected to **EN**.
- modbus_config** (Array) connected to **setModbusConfig**.
- communication_configuration** (SerialCommunicationPara) connected to **stSerialCommunicationPara**.
- modbus_command_quantity** (Byte) connected to **byModbusConfigNumber**.
- error_status** (Bool) connected to **xError**.
- error_ID** (ModbusMasterErrorCode) connected to **ErrorID**.



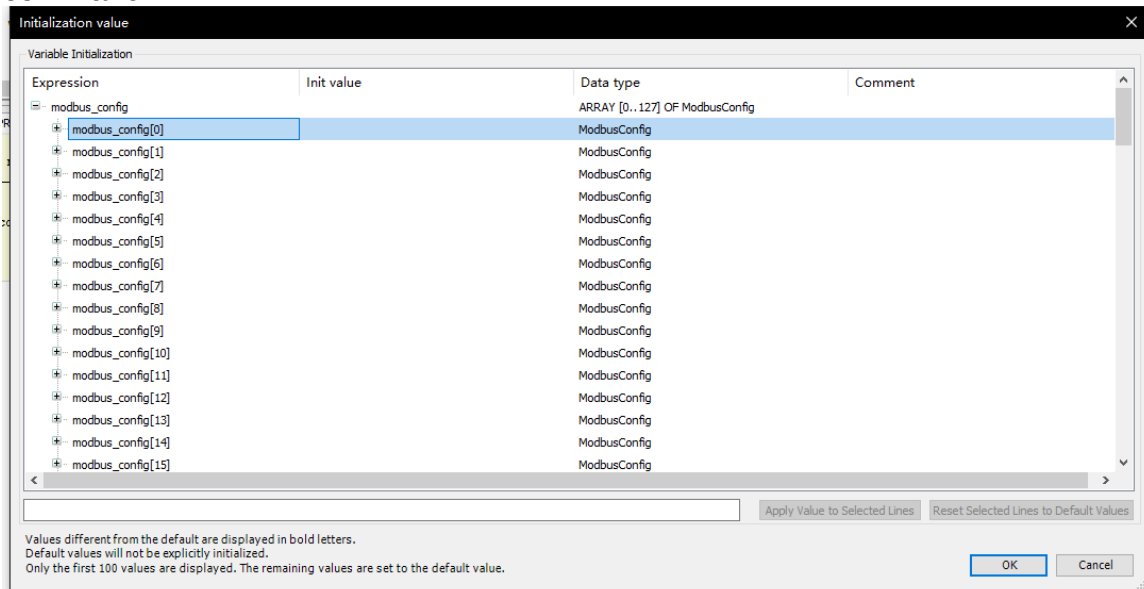
6) Configure the communication command items.



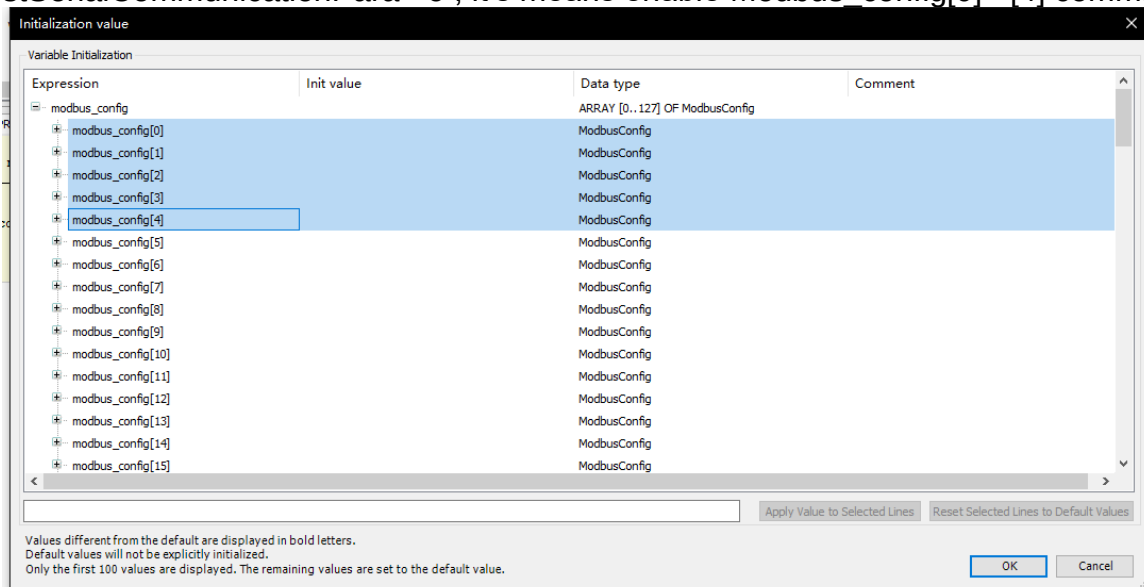


7) Select the command quantity

For example: stSerialCommunicationPara =1, it's means enable Modbus_config[0] command

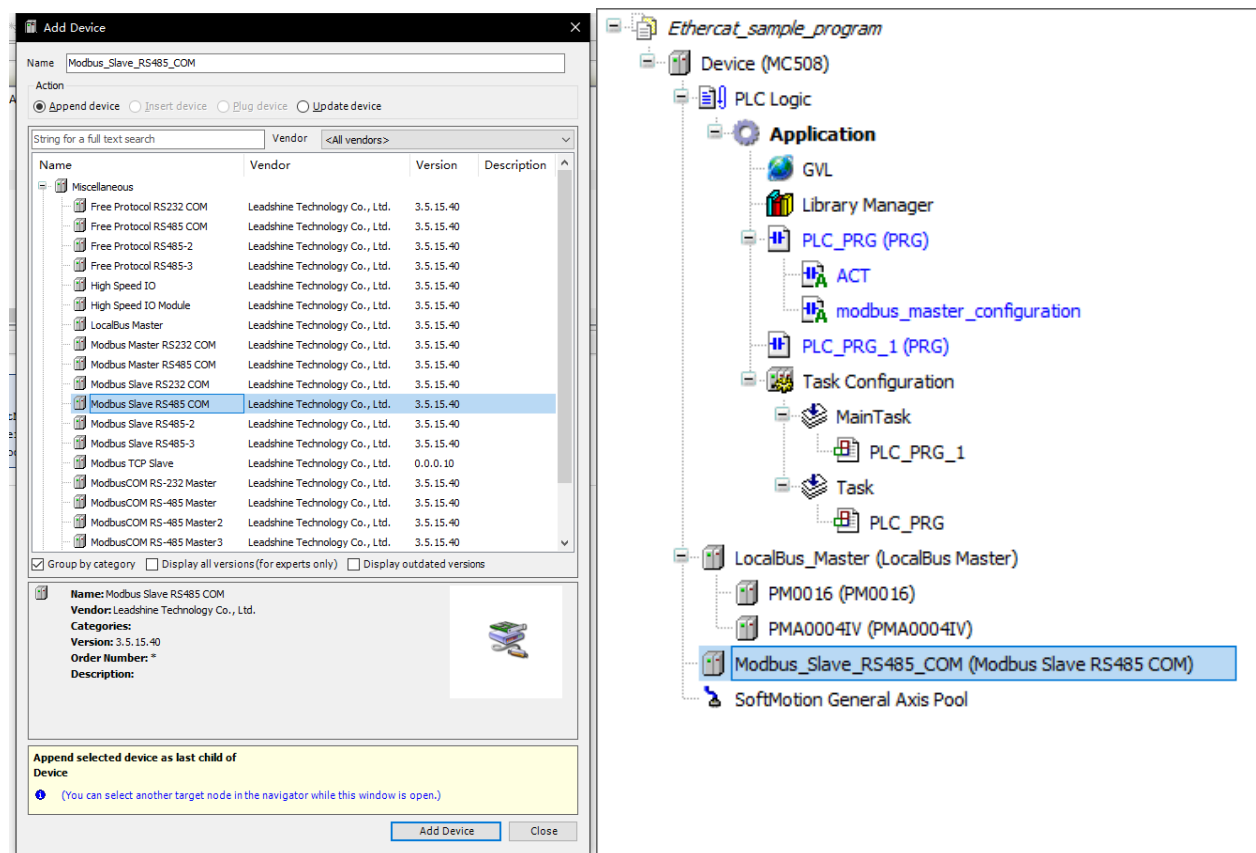


stSerialCommunicationPara =5 , it's means enable Modbus_config[0] ~[4] command



6.2.2.Modbus Slave Station Configuration

1) Add the slave device to configure MC500 series PLC as slave station.



2) Configure the slave station parameters.

Parameter	Type	Value	Default Value	Unit	Description
Vendor	STRING	Leadshine Technology Co., Ltd.	Leadshine Technology Co., Ltd.		Vendor of the device
Mode Name	STRING	'Modbus Com Slave'	'Modbus Com Slave'		Modbus Slave Com Port
Vendor					Vendor of the device
SlaveID	INT	1	1		
FrameIntervalTime	TIME	T#5MS	T#5MS		
Timeout	TIME	T#5S	T#5S		
Baudrate	UDINT	9600	9600		
ByteSize	UDINT	8	8		
StopBits	UDINT	1	1		
Parity	UDINT	0	0		
Data_Mode	WORD	0	0		
modbus slave Device diag					modbus slave Device diag
BaseInfo	BYTE	0	0		
FaulttoSlaveFlag	BYTE	0	0		
ErrorCode	BYTE	0	0		

6.2.3.Modbus Variable Address

MC 500 series PLC include Q type ,I type and M type variable areas which can be accessed by bit, byte word, and dual-word.

For example, %QX, %QB, %QW, and %QD are converted as follows:

QB0 = (QX0.0-QX0.7)

QW0 = (QB0-QB1) = ((QX0.0-QX0.7) + (QX1.0-QX1.7));

QD0 = (QW0-QW1) = (QB0-QB4) = ((QX0.0-QX0.7) + (QX1.0-QX1.7)+(QX2.0-QX2.7)+(QX3.0-QX3.7))\

3) Variables addressing table

Bitwise addressing	Bytewise addressing	Wordwise addressing	DWordwise addressing	Bitwise addressing	Bytewise addressing	Wordwise addressing	DWordwise addressing
QX0.0	QB0	QW0	QD0	MX0.0	MB0	MW0	MD0
QX0.1				MX0.1			
QX0.2				MX0.2			
QX0.3				MX0.3			

QX0.4					MX0.4			
QX0.5					MX0.5			
QX0.6					MX0.6			
QX0.7					MX0.7			
QX1.0	QB1				MX1.0	MB1		
QX1.1					MX1.1			
QX1.2					MX1.2			
QX1.3					MX1.3			
QX1.4					MX1.4			
QX1.5					MX1.5			
QX1.6					MX1.6			
QX1.7					MX1.7			
QX2.0	QB2	QW1			MX2.0	MB2	MW1	
QX2.1					MX2.1			
QX2.2					MX2.2			
QX2.3					MX2.3			
QX2.4					MX2.4			
QX2.5					MX2.5			
QX2.6					MX2.6			
QX2.7					MX2.7			
QX3.0	QB3				MX3.0	MB3		
QX3.1					MX3.1			
QX3.2					MX3.2			
QX3.3					MX3.3			
QX3.4					MX3.4			
QX3.5					MX3.5			
QX3.6					MX3.6			
QX3.7					MX3.7			

4) Variables access range

Areas	Range	Function code	Initial address	Quantity
Q type	%QW0~%QW4095 (QX0.0 ~ QX8191.7)	0x01,0x05,0x0f	0	65536
I type	%IW0~%IW4095 (IX0.0 ~ IX8191.7)	0x02	0	65536
M type	%MW0~%MW65535	0x03,0x06,0x10	0	65536

Chapter 7 Ethernet Communication

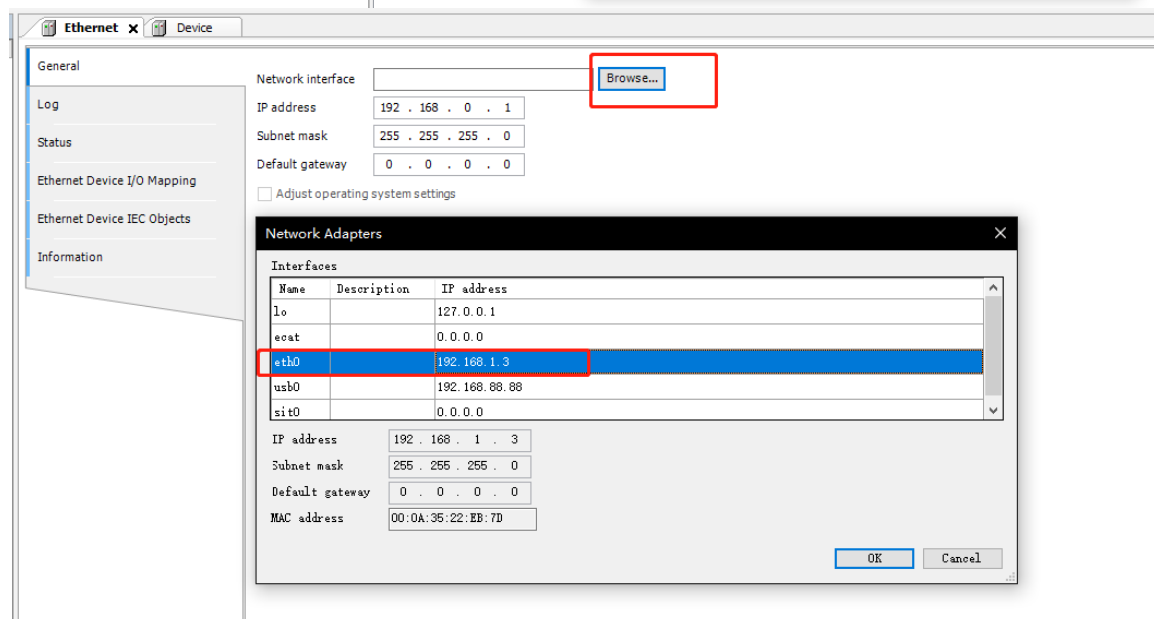
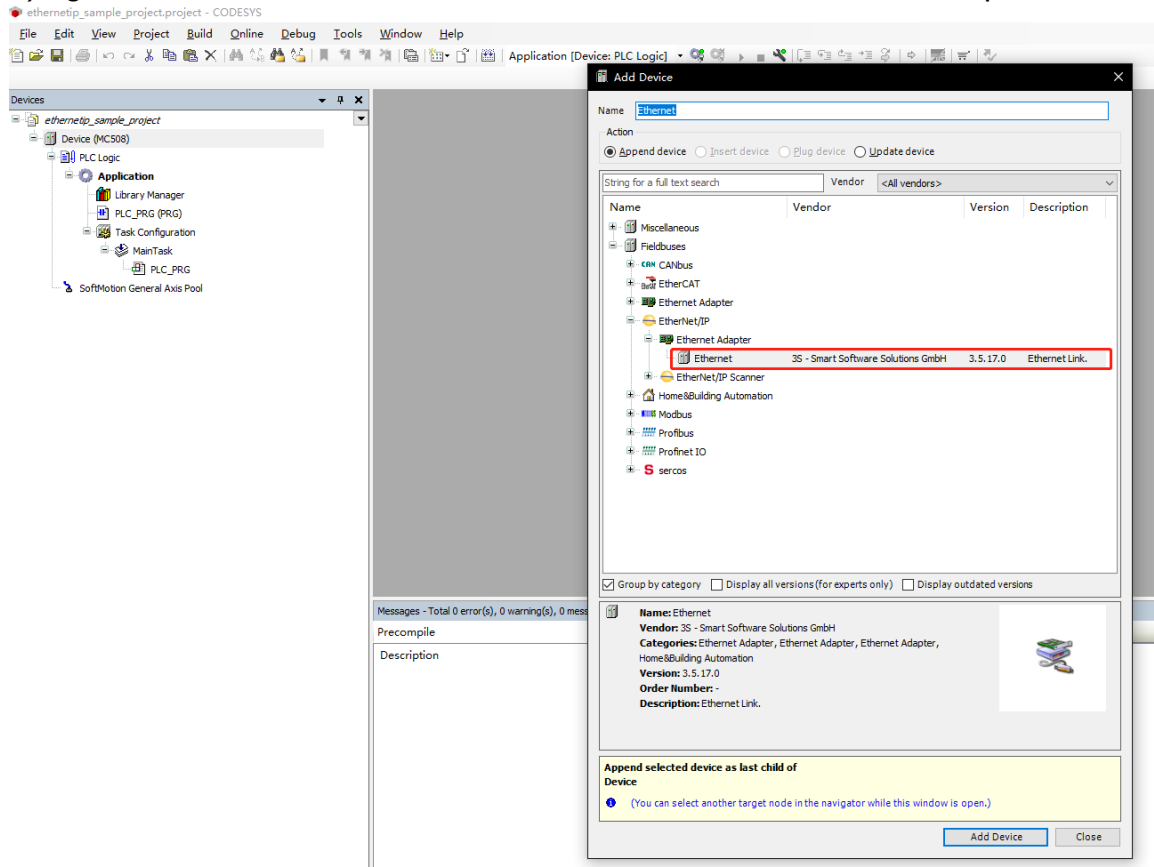
Ethernet/IP is an industrial application layer protocol for industrial automation applications. It is based on the Industrial Ethernet standard, which is introduced by ODVA (OpenDeviceNet Vendors Association) and ControlNet International and is combined with TCP/IP Ethernet.

7.1.Ethernet/IP Communicating

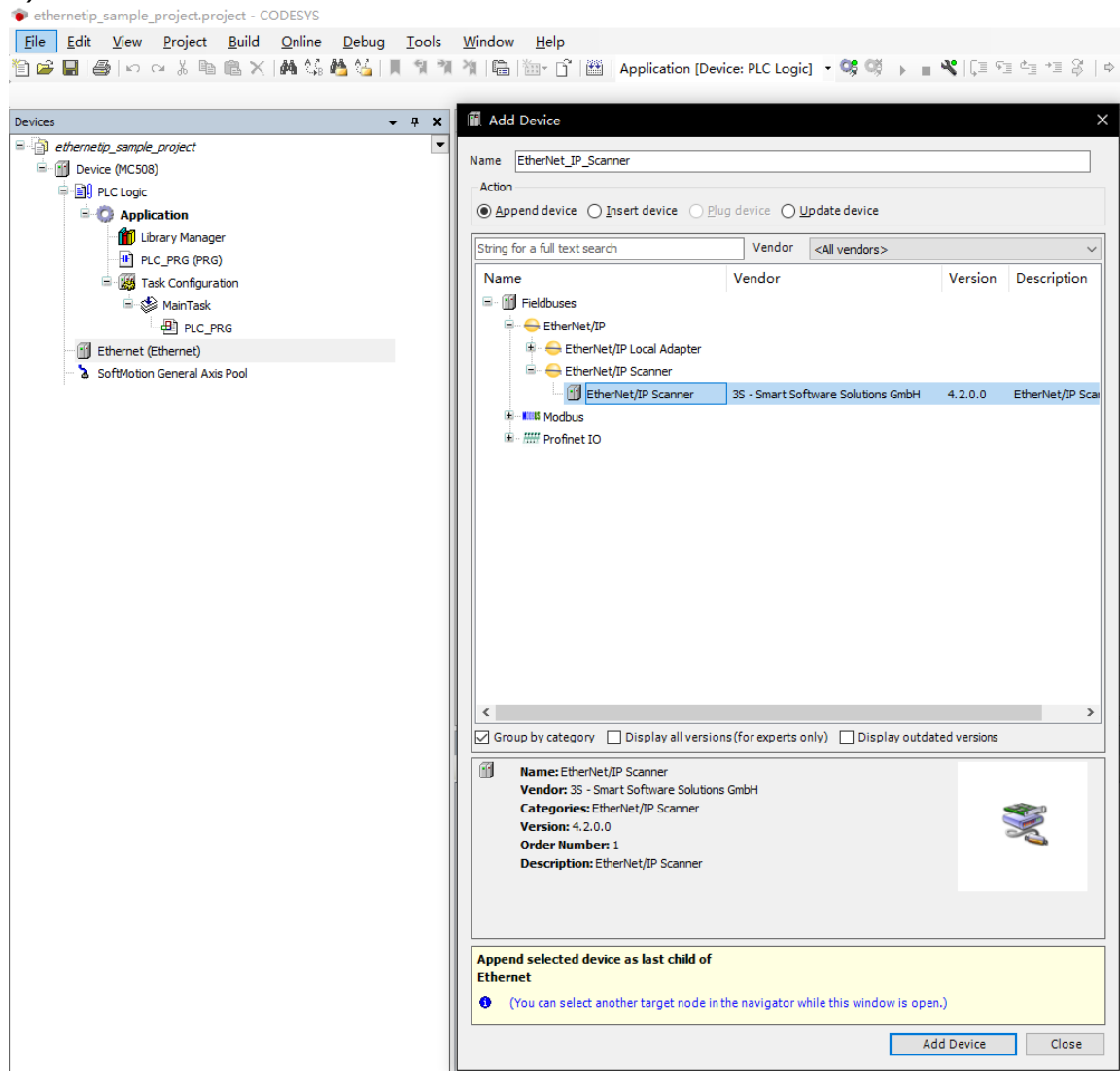
7.1.1.Configuring PLC As Master Station

Add the drive as slave station

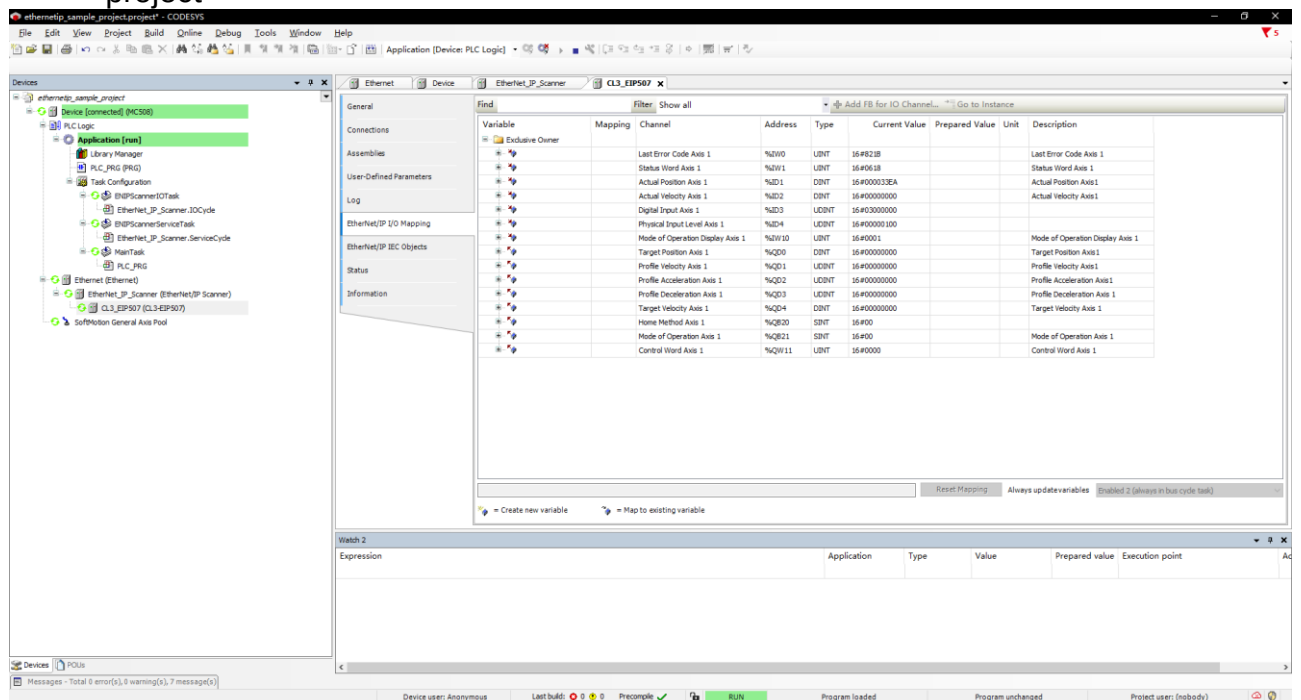
1) right-click the “device” click “Ethernet/IP”, then select the Ethernet port.



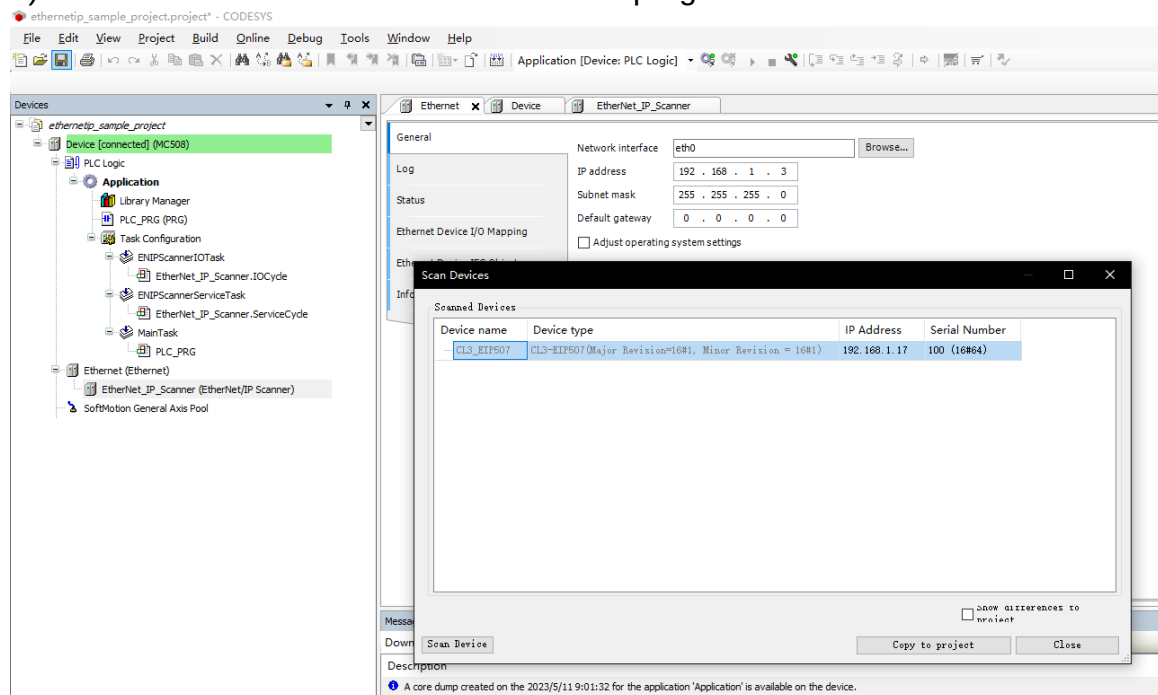
2) Add the Ethernet/IP scanner



3) Using the auto scanning function find the device or add the device manually to the project



4) Download to PLC then create the control program.



Chapter 8 EtherCAT Configuration

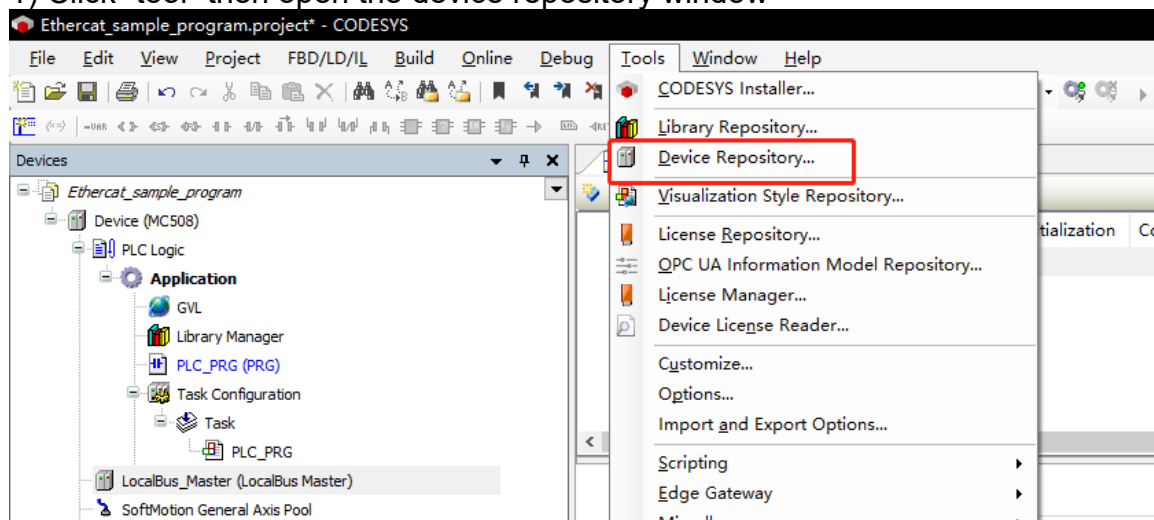
8.1.Overview

EtherCAT is an open industrial field technology over the Ethernet. It features short communication update interval, low synchronization jitter, and low hardware cost. EtherCAT supports the linear, tree, star, and hybrid topologies. EtherCAT slave stations must use dedicated communication chipset ESC, and EtherCAT master stations can use a standard Ethernet controller.

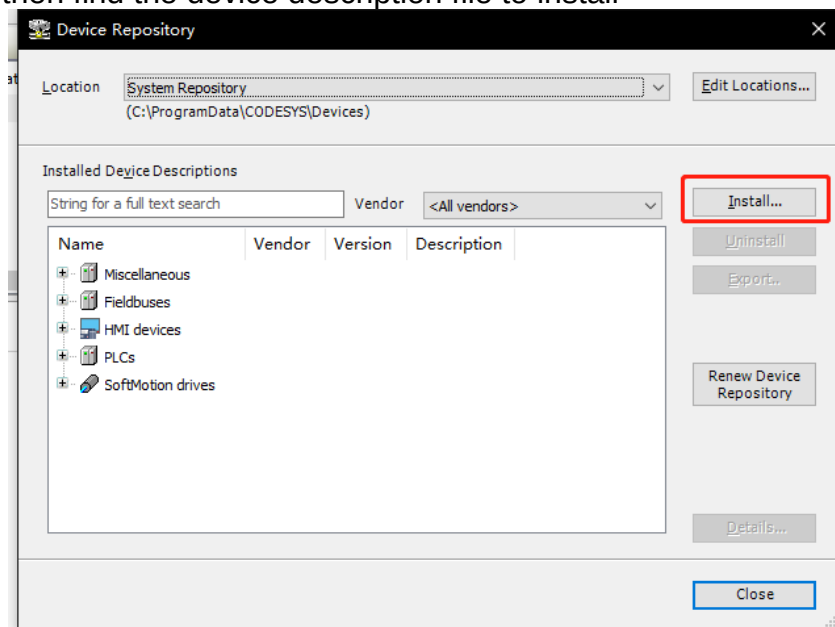
8.2.Add “.Xml” File

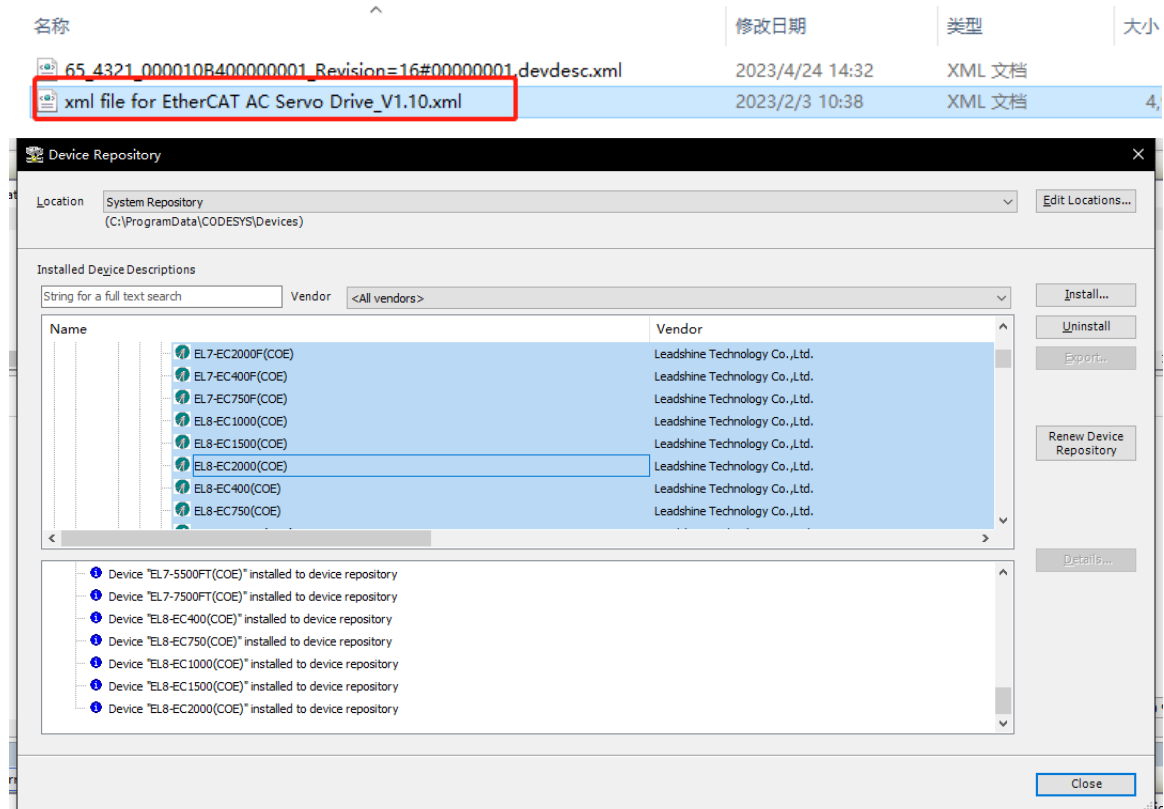
EtherCAT device installation is to import the device description file (with file name extension .XML) in compliance with ETG (EtherCAT Technical Committee) standards into the programming software CODESYS. After the software parses and processes the file, it generates the EtherCAT configure devices that can be added and deleted by users. User need to use third-party the EtherCAT devices, install the device description files provided by the third-party vendors.

1) Click “tool” then open the device repository window



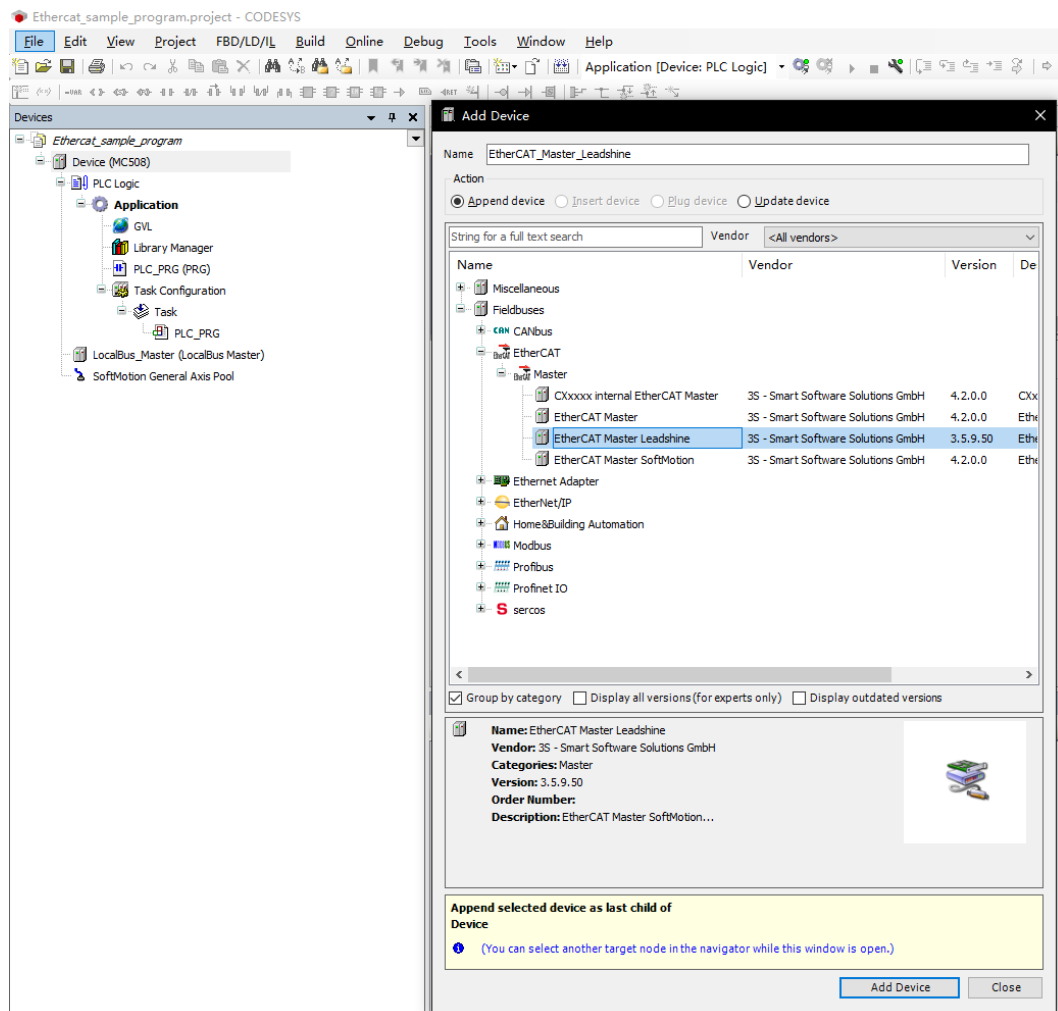
2) Click install ,then find the device description file to install



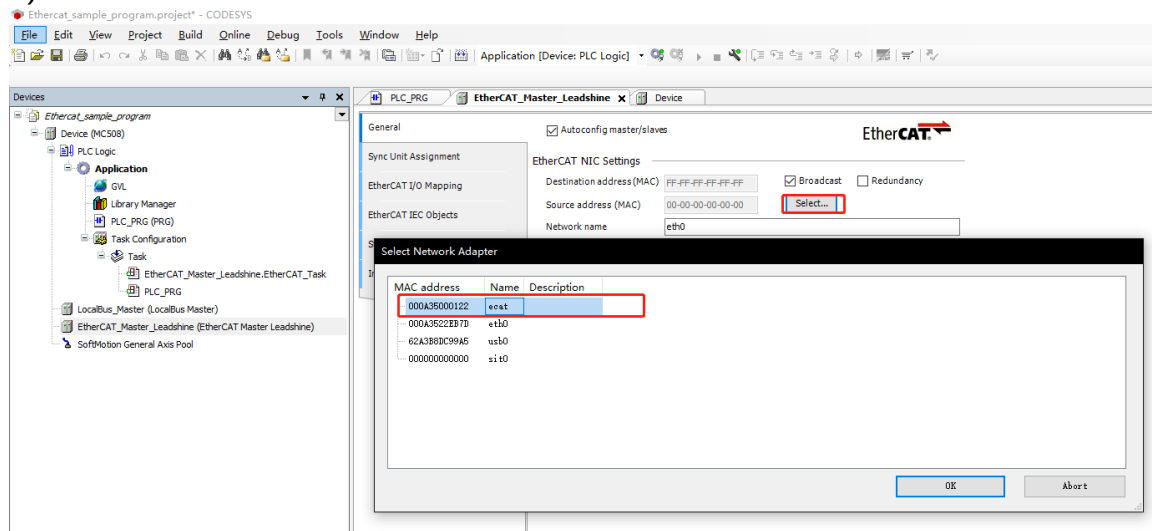


8.3.EtherCAT Master Station Configuration

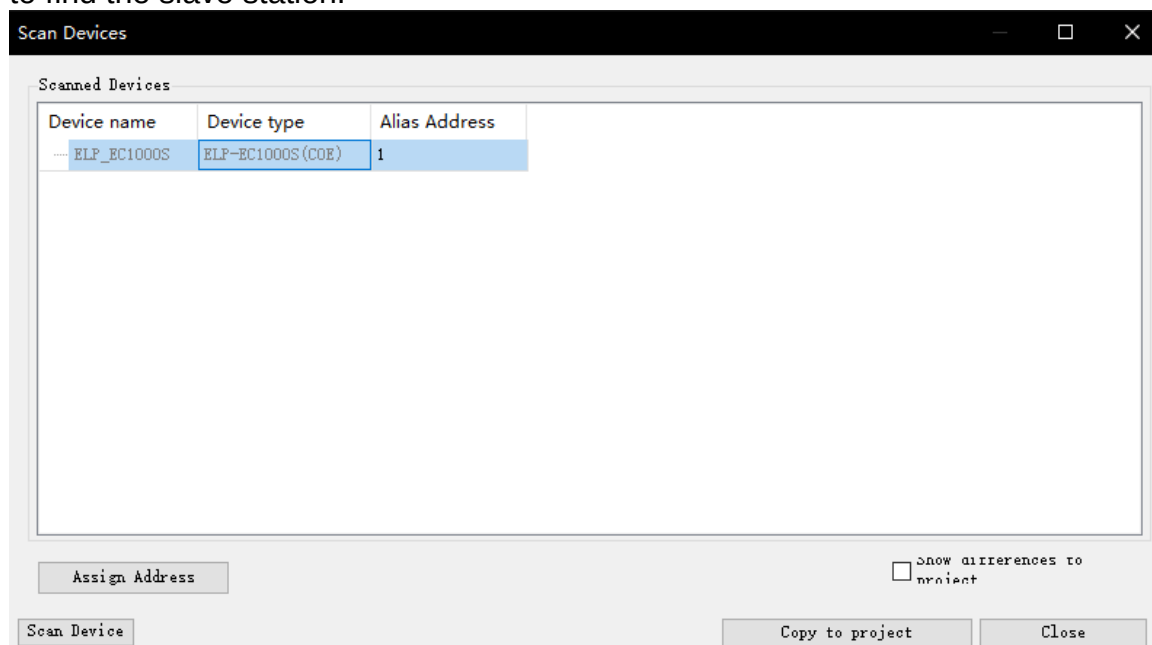
1) After install the .xml file, add the EtherCAT device, right click “Device” to add the EtherCAT master “EtherCAT Master Leadshine”



2) Select the EtherCAT source address



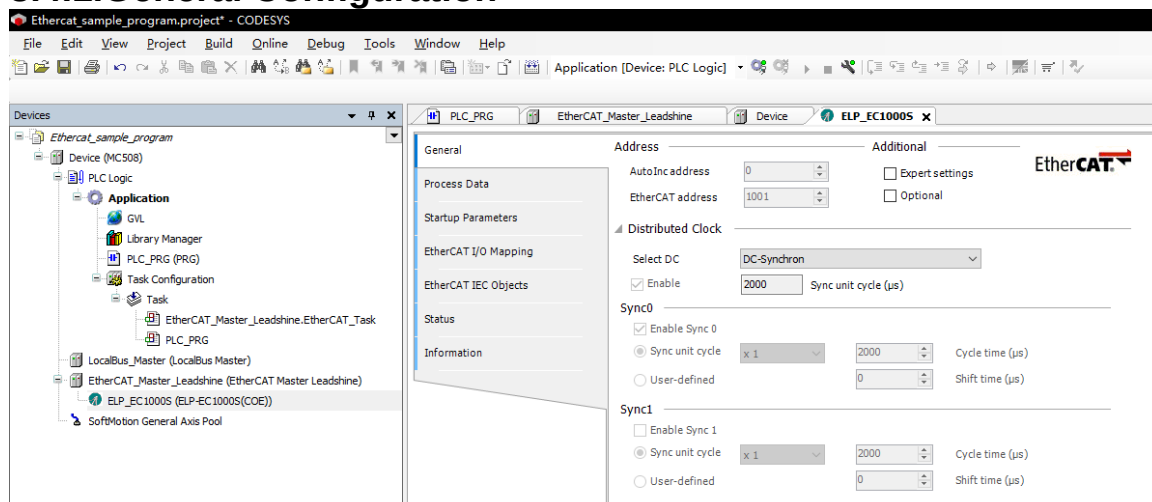
3) When the PLC connected the device via EtherCAT port, use the auto-scanning function to find the slave station.



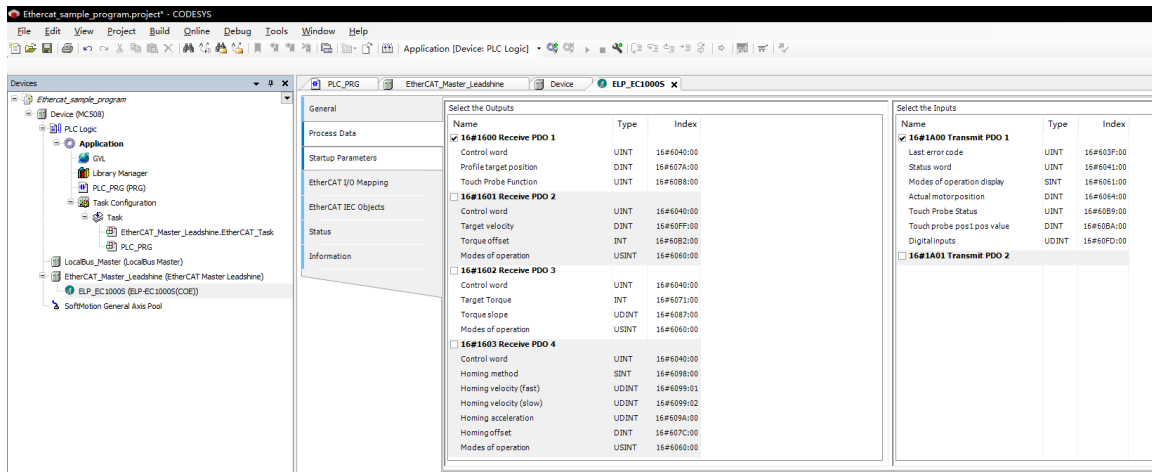
8.4.EtherCAT Slave Station Configuration

After add the device to project, user can modify the general configuration, SDO, PDO, and EtherCAT I/O mapping.

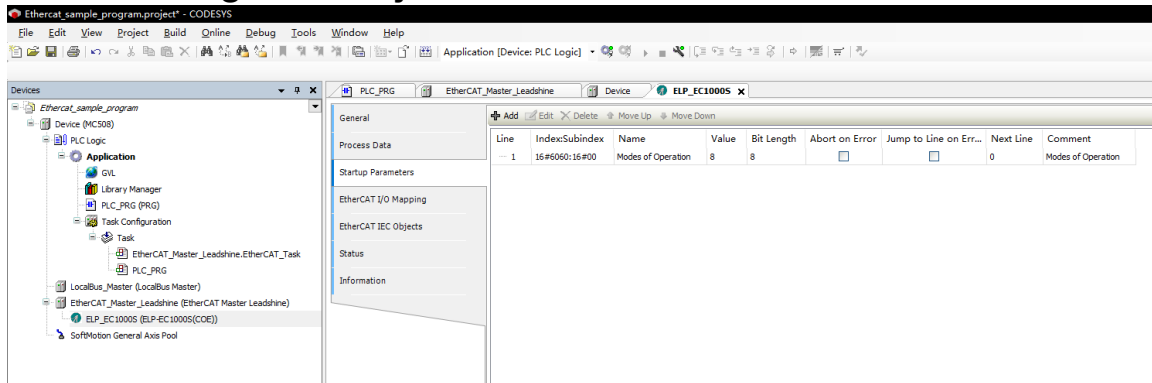
8.4.1.General Configuration



8.5.Process Data Object (PDO)

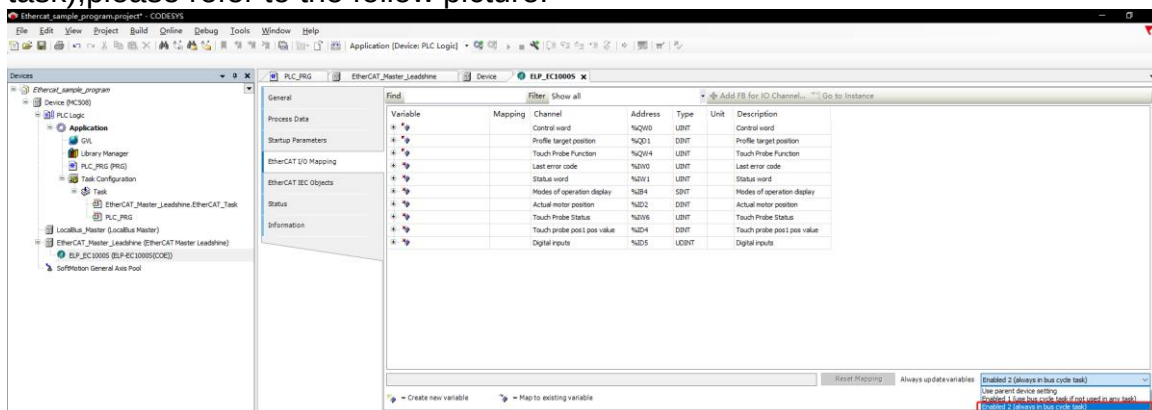


8.5.1.Starting Data Object (SDO)



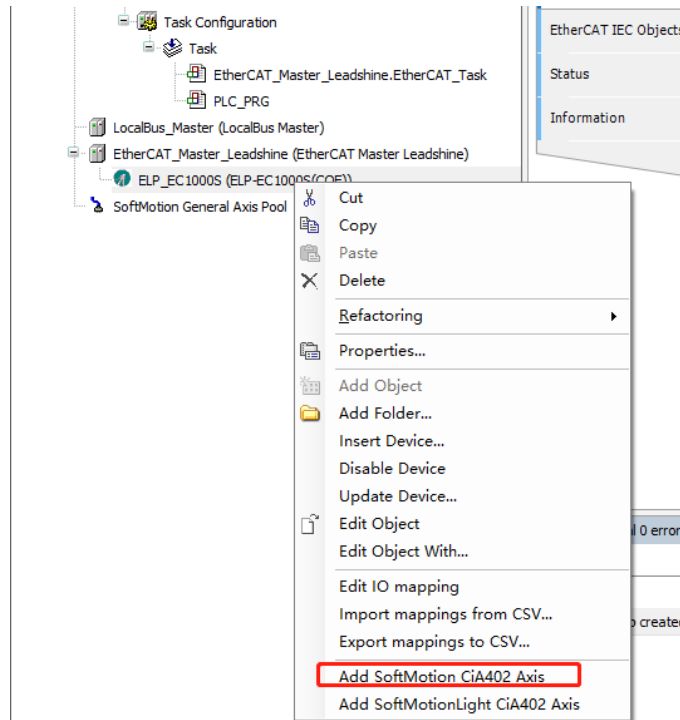
8.5.2.EtherCAT I/O mapping

And the EtherCAT I/O mapping have to select the Enabled 2 (always in bus cycle task),please refer to the follow picture.

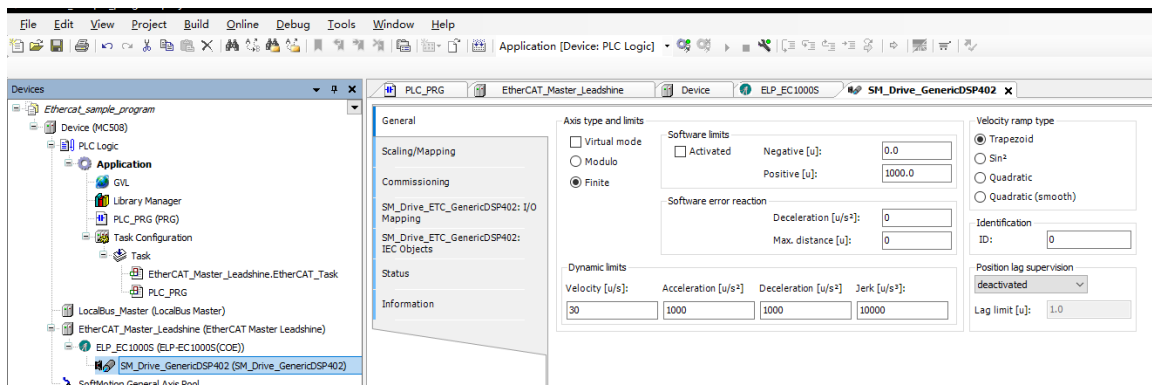


8.6.Add 402 Axis To Servo Slave Station

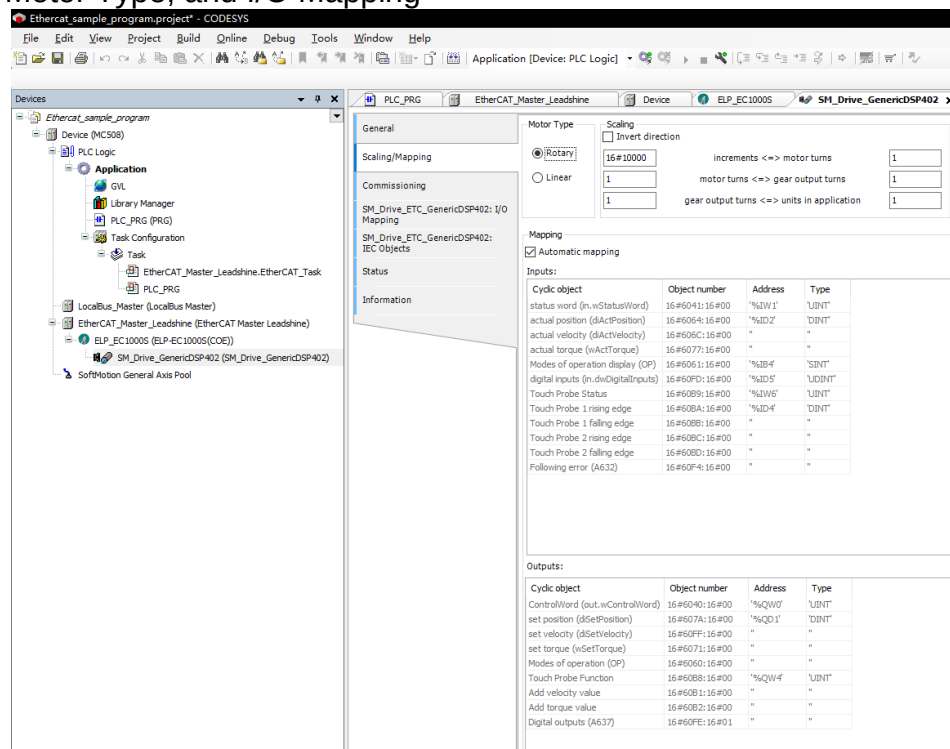
1) Right click the slave drive to Add the SoftMotion CIA402 Axis.



2) Axis general configuration



3) Scaling, Motor Type, and I/O Mapping

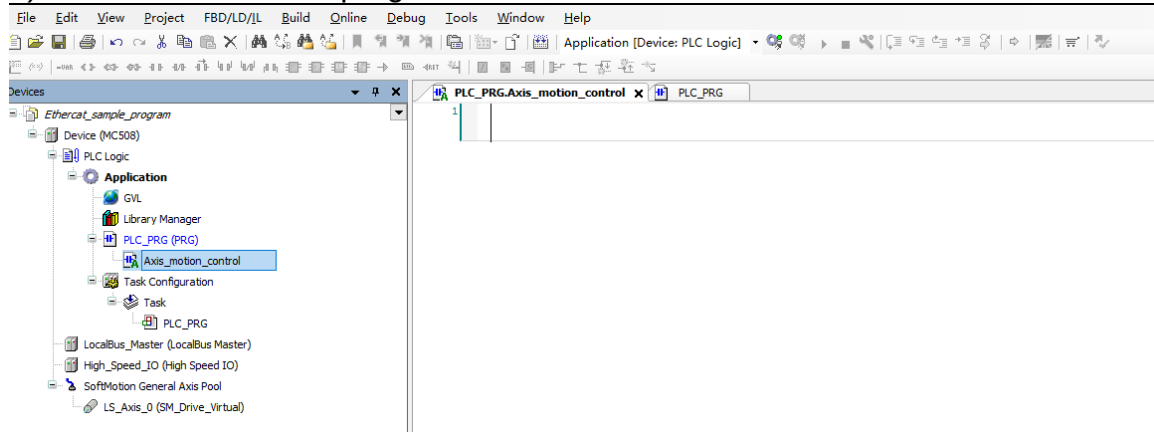


Chapter 9 Motion Control

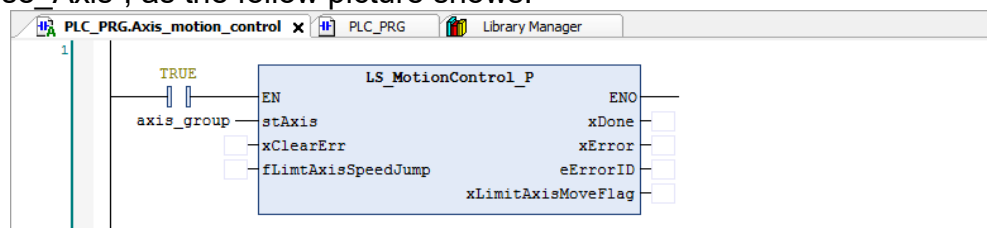
9.1.1.High Speed Pulse Axis Control Program

Finish high speed pulse axis basic configuration, user can add some function blocks to control motor.

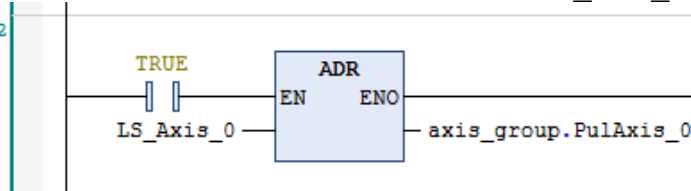
1) Create a new action program.



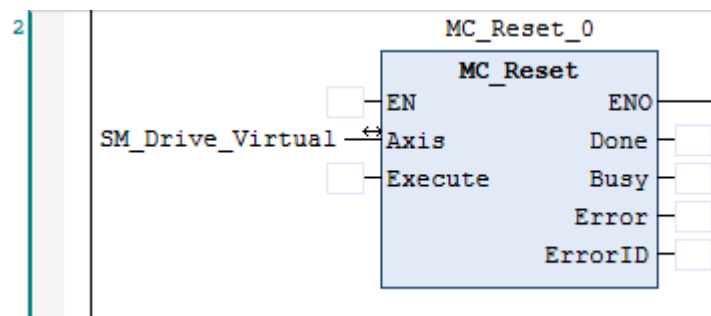
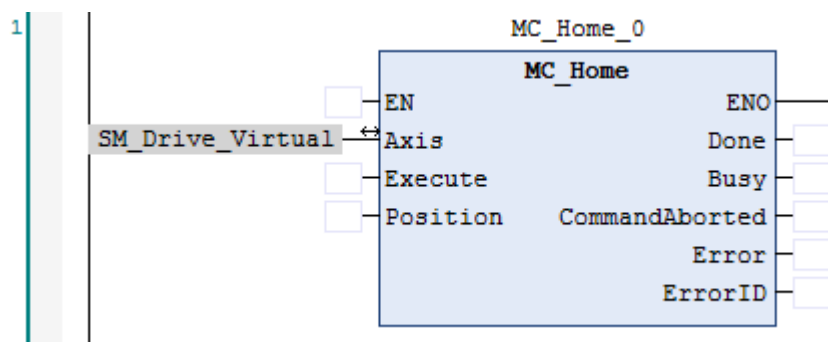
2) Add the “LS_MotionControl_P”, it’s the special function block developed by Leadshine for high speed pulse output axis control, then define pin “stAxis” variable, the data type is “DUT_Pulse_Axis”, as the follow picture shows.

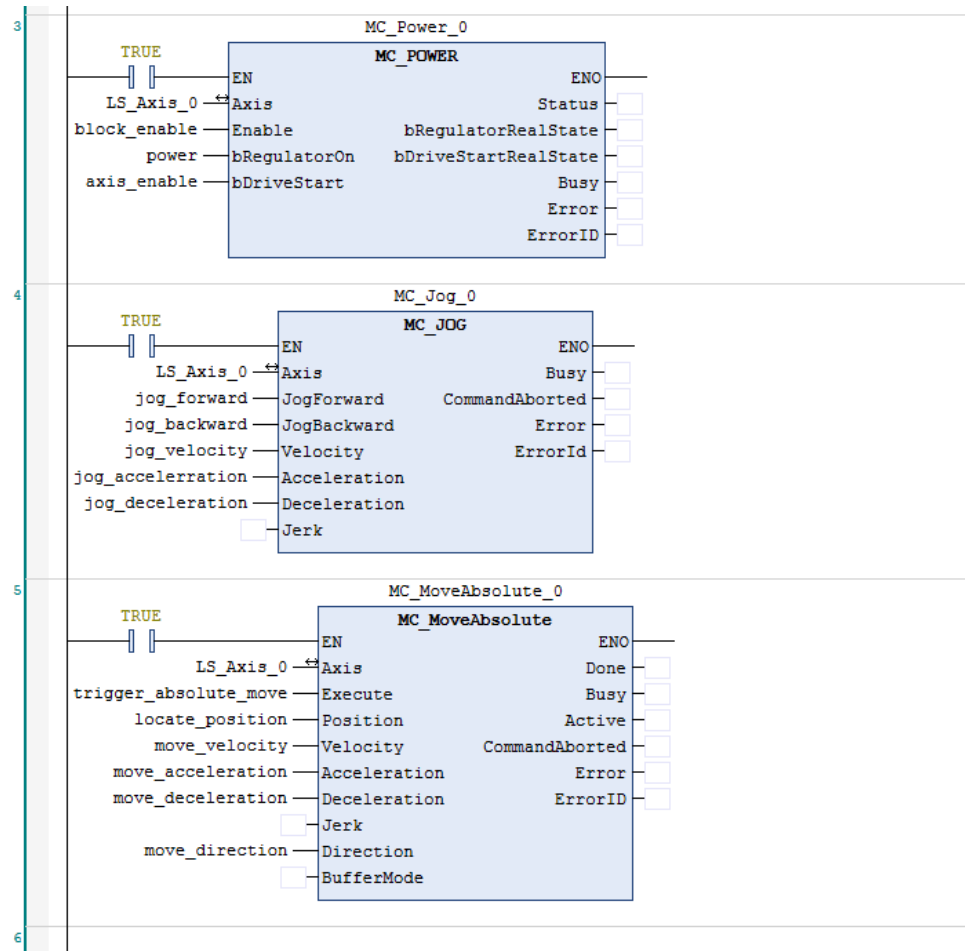


3) Use “ADR” command to define axis 0 name “LS_Axis_0” as the axis group 0 axis



4) Create motion control function blocks



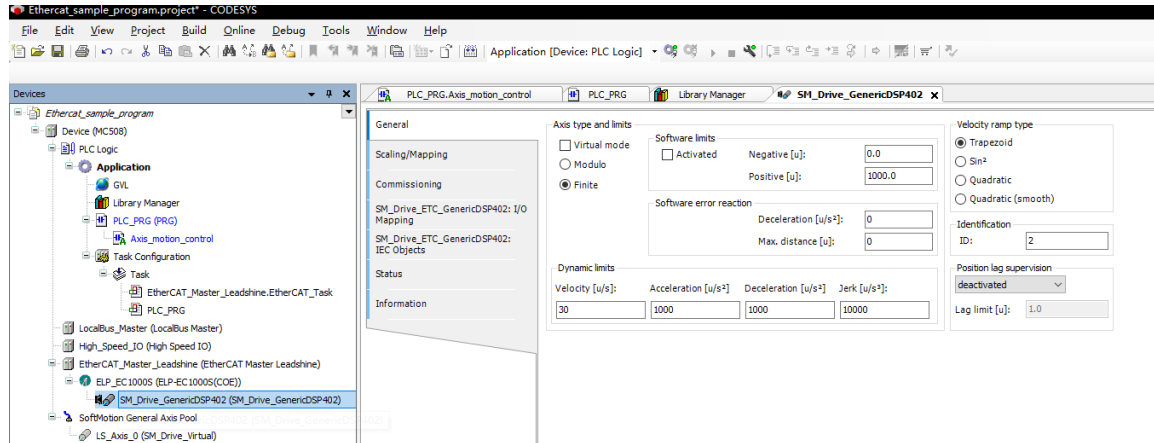


9.2.EtherCAT Axis Configuration

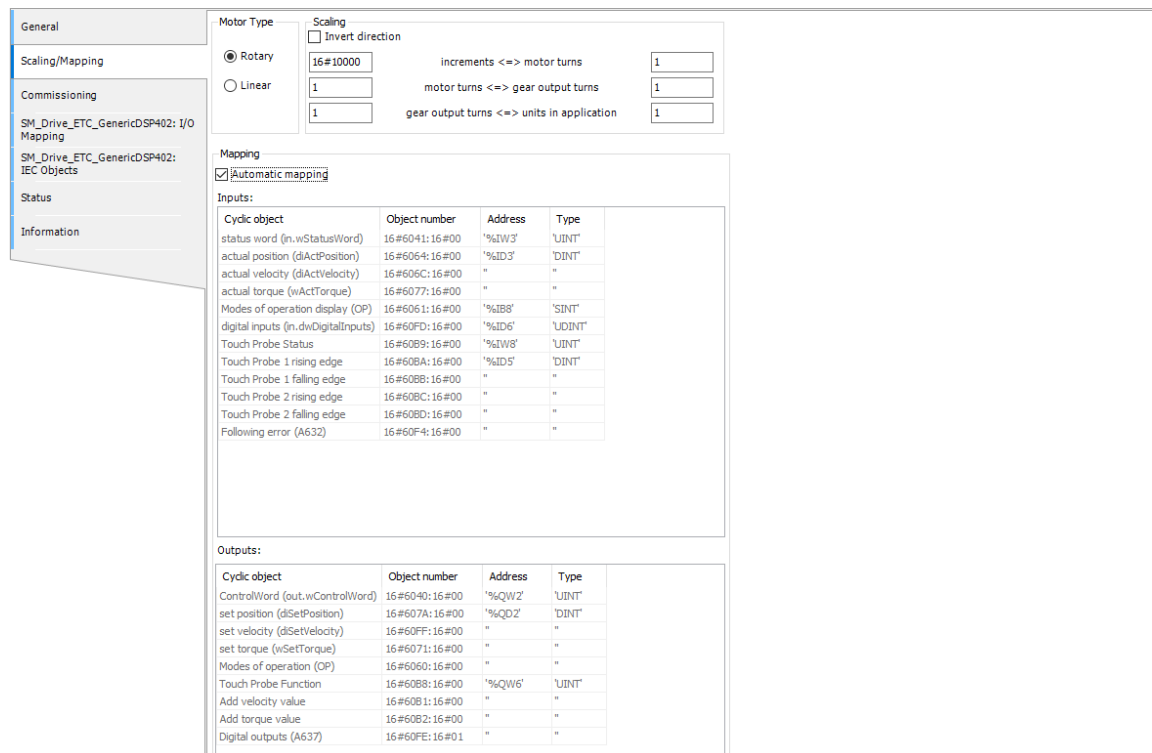
Users have to add the CIA402 axis device to project when using MC500 series PLC control the EtherCAT axis.

9.2.1.General Configuration

In the general interface, user can configure the axis type and software limits, motion maximum etc.



9.2.2.Scaling And Mapping



1) Scaling configuration

User can accord the actual application scenarios to configure “increments” (the motor command pulse counts per revolution), “gear output turns” (gearbox ratio), the unit of actual move distance.

For example: setting the servo motor or stepper motor 10000 pulses per rotation, motor connect with 1:20 gearbox, and gearbox output shaft directly drives the ball screw to move, screw lead is 5 mm.

In this case, scaling configuration refer to follow picture

Scaling

☐ Invert direction

10000

increments <=> motor turns

1

20

motor turns <=> gear output turns

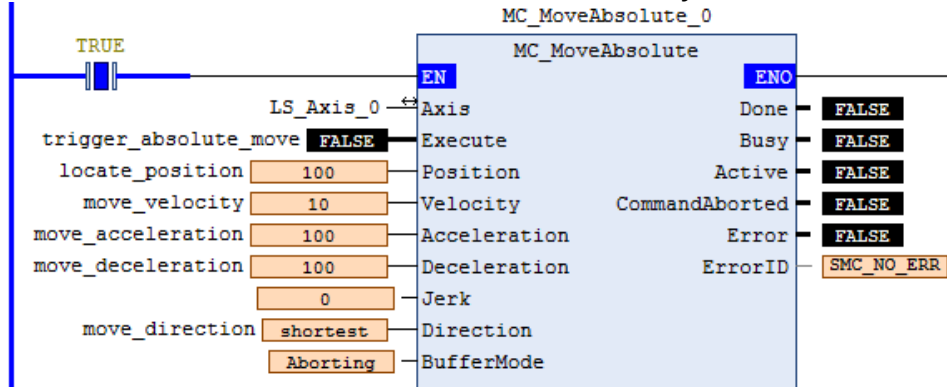
1

1

gear output turns <=> units in application

5

Using the motion command “MC_MoveAbsolute”, setting the position=100,velocity=10.It’s means that move actual load 100mm, the velocity is 10 mm/s



2) Mapping configuration

When automatic mapping is selected, the slave station is associated with axis. The slave station data is mapped to the axis directly, otherwise, user can manually modify the address in axis mapping, in which:

Input format is %I+ Type letters + Arabic numbers

Mapping

☐ Automatic mapping

Inputs:

Cyclic object	Object number	Address	Type
status word (in.wStatusWord)	16#6041:16#00	'%IW3'	'UINT'
Modes of operation display (OP)	16#6061:16#00	'%IB8'	'SINT'
actual position (diActPosition)	16#6064:16#00	'%ID3'	'DINT'
actual velocity (diActVelocity)	16#606C:16#00	"	"
actual torque (wActTorque)	16#6077:16#00	"	"
Touch Probe Status	16#60B9:16#00	'%IW8'	'UINT'
Touch Probe 1 rising edge	16#60BA:16#00	'%ID5'	'DINT'
Touch Probe 1 falling edge	16#60BB:16#00	"	"
Touch Probe 2 rising edge	16#60BC:16#00	"	"
Touch Probe 2 falling edge	16#60BD:16#00	"	"
Following error (A632)	16#60F4:16#00	"	"
digital inputs (in.dwDigitalInputs)	16#60FD:16#00	'%ID6'	'UDINT'

Output format is %Q+ Type letters + Arabic numbers

Outputs:

Cyclic object	Object number	Address	Type
ControlWord (out.wControlWord)	16#6040:16#00	'%QW2'	'UINT'
set position (diSetPosition)	16#607A:16#00	'%QD2'	'DINT'
set velocity (diSetVelocity)	16#60FF:16#00	"	"
set torque (wSetTorque)	16#6071:16#00	"	"
Modes of operation (OP)	16#6060:16#00	"	"
Touch Probe Function	16#60B8:16#00	'%QW6'	'UINT'
Add velocity value	16#60B1:16#00	"	"
Add torque value	16#60B2:16#00	"	"
Digital outputs (A637)	16#60FE:16#01	"	"

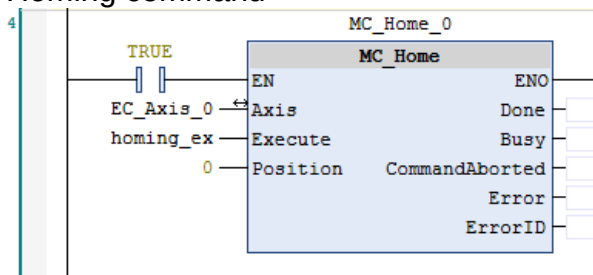
9.2.3.Homing Parameters Configuration

User need to use the SDO to configure the homing parameters, then via motion command “MC_Home” to trigger homing.

1) SDO configuration

Line	Index/Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Err...	Next Line	Comment
1	16#6060:16#00	Modes of Operation	8	8	☐	☐	0	Modes of Operation
2	16#6098:16#00	Homing method	19	8	☐	☐	0	
3	16#6099:16#01	Homing velocity (fast)	10000	32	☐	☐	0	
4	16#6099:16#02	Homing velocity (slow)	1000	32	☐	☐	0	
5	16#609A:16#00	Homing acceleration	100000	32	☐	☐	0	

2) Homing command



9.2.4.EtherCAT Axis Control Program

The EtherCAT axis motion command is same with the high speed motion command except the homing command.

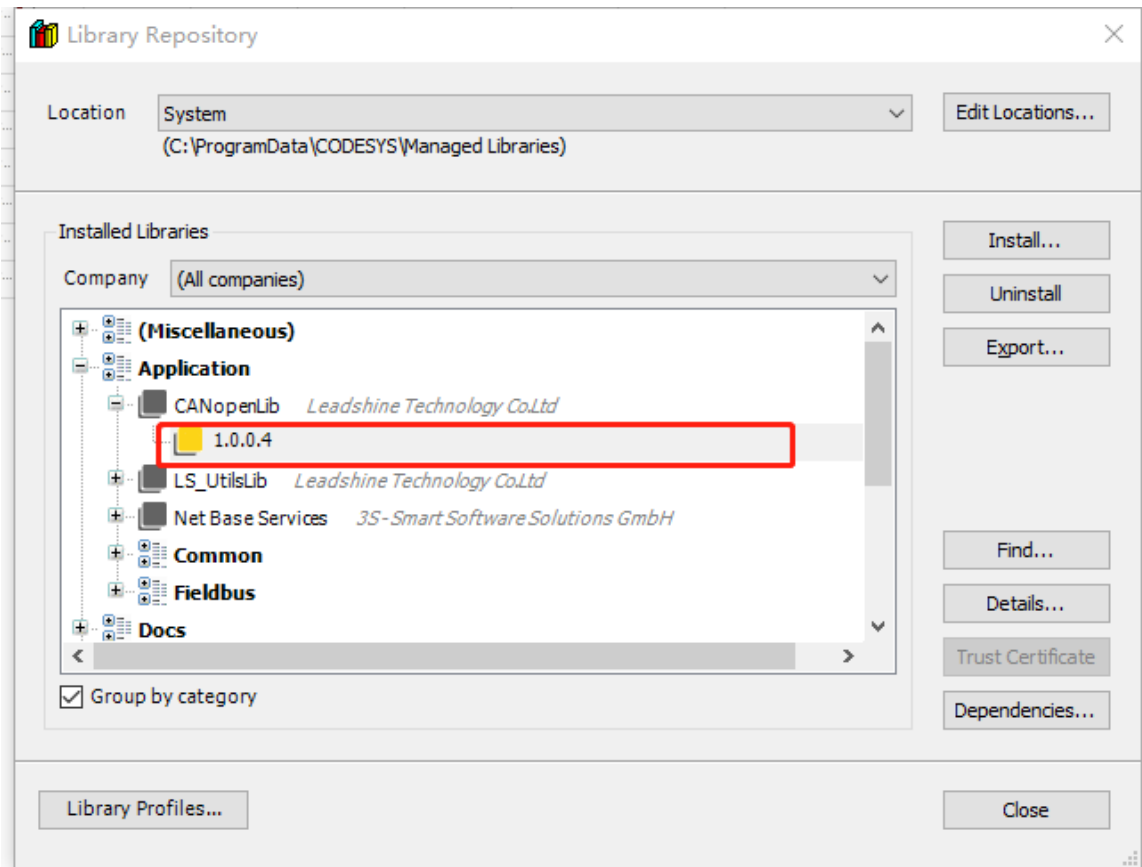


9.3.CANopen axis configuration

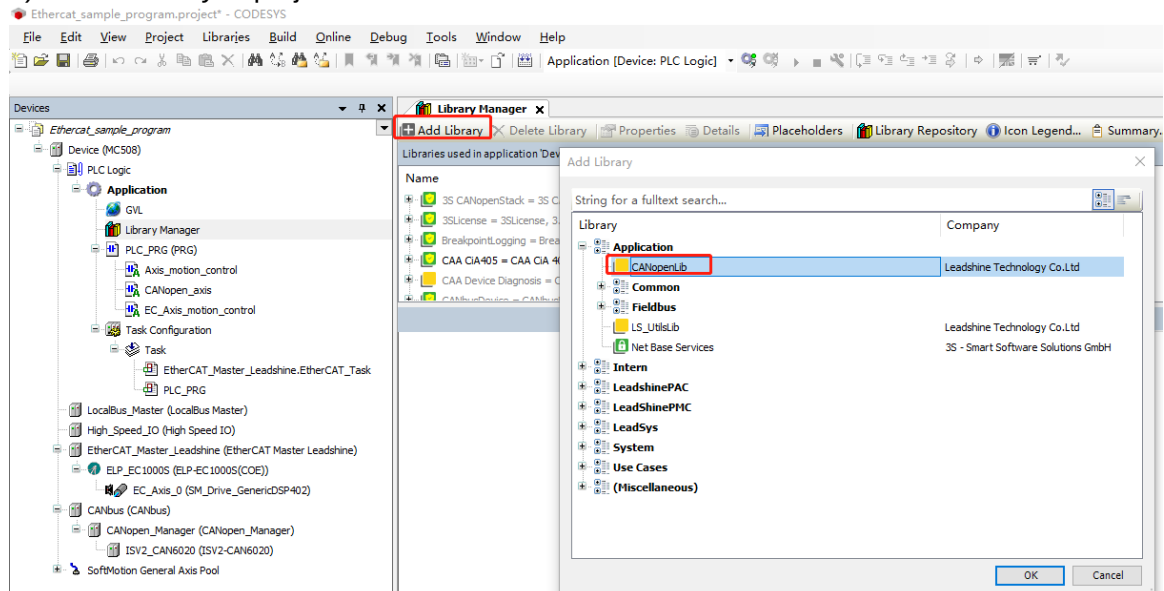
MC500 series PLC support standard CANopen protocol, user can via canopen function libraries to control the CANopen slave axis.

1) Install the CANopen library

名称	修改日期	类型	大小
CANopenLib.compiled-library	2022/11/21 9:57	COMPILED-LIBR...	103 KB

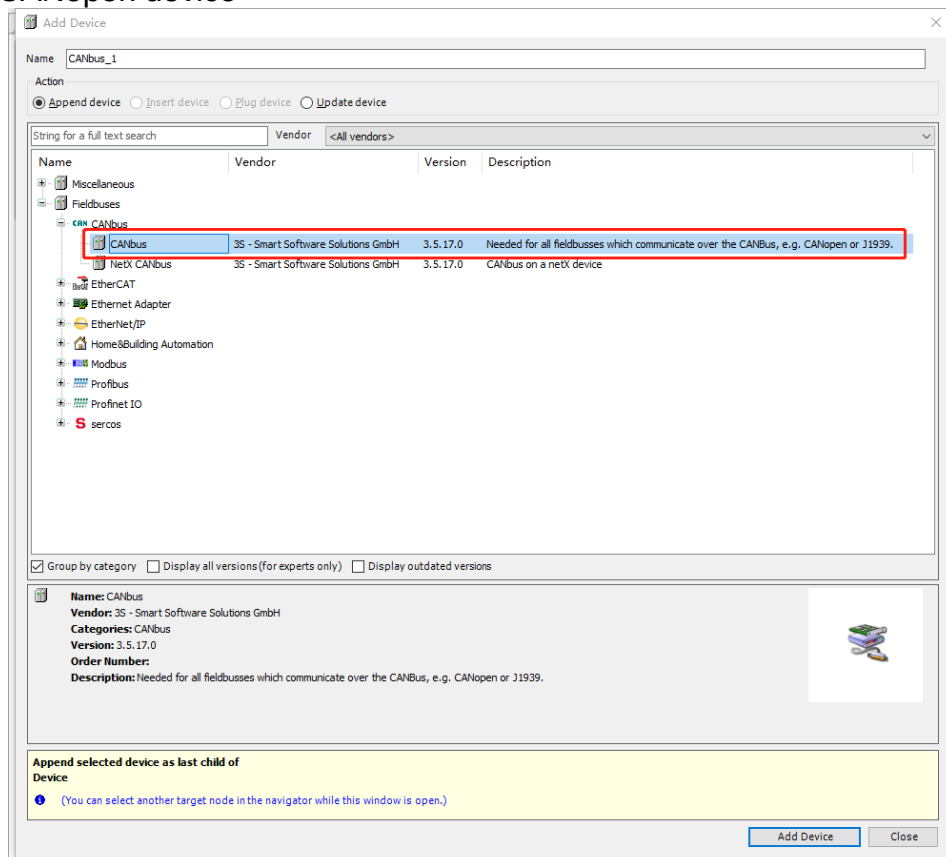


2) Add the library to project

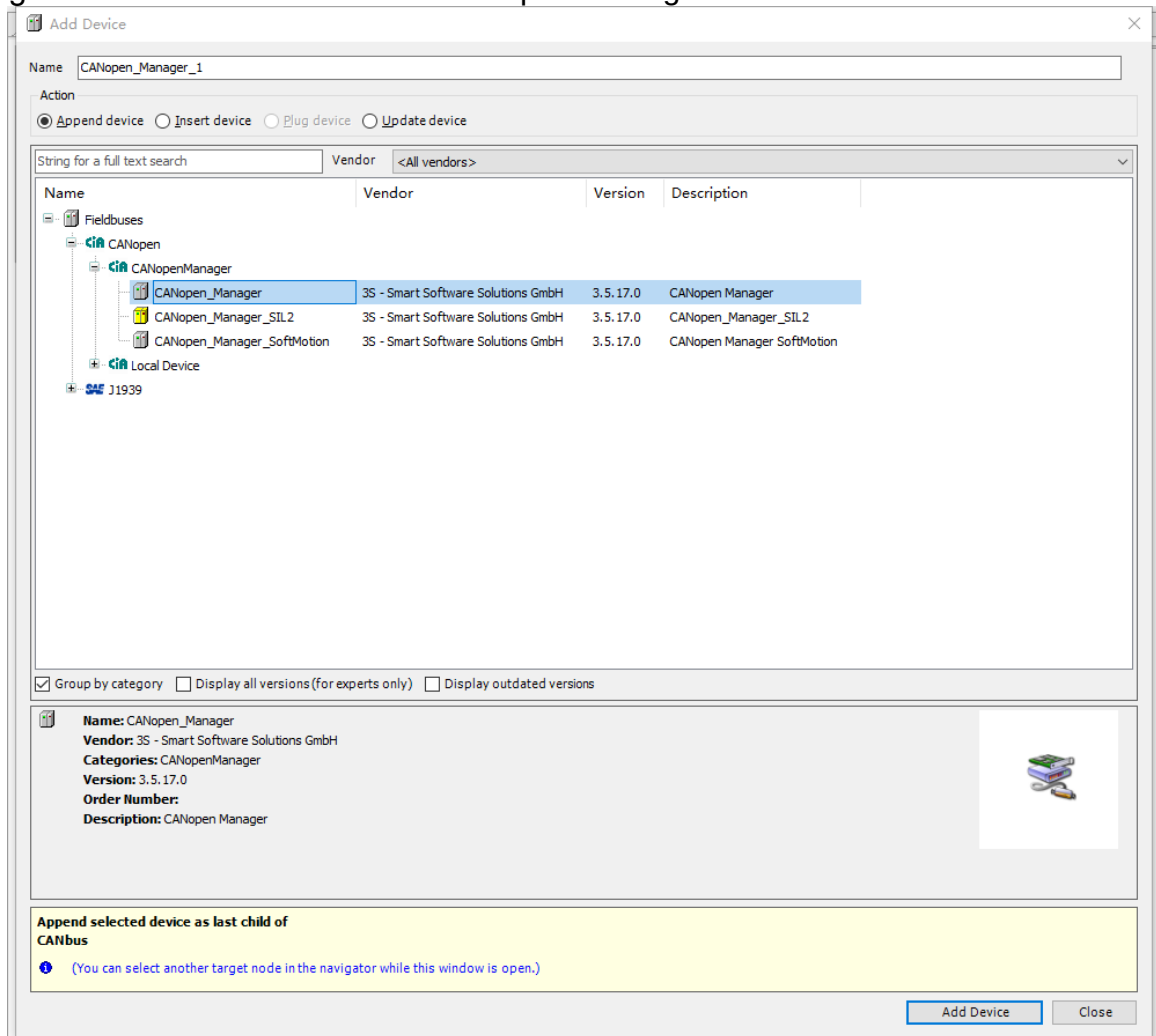


9.3.1.General configuration

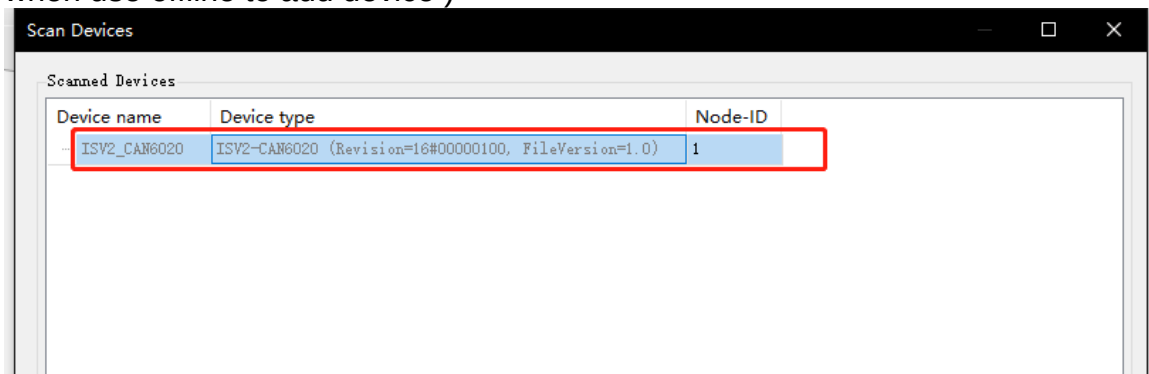
1) Add the CANopen device



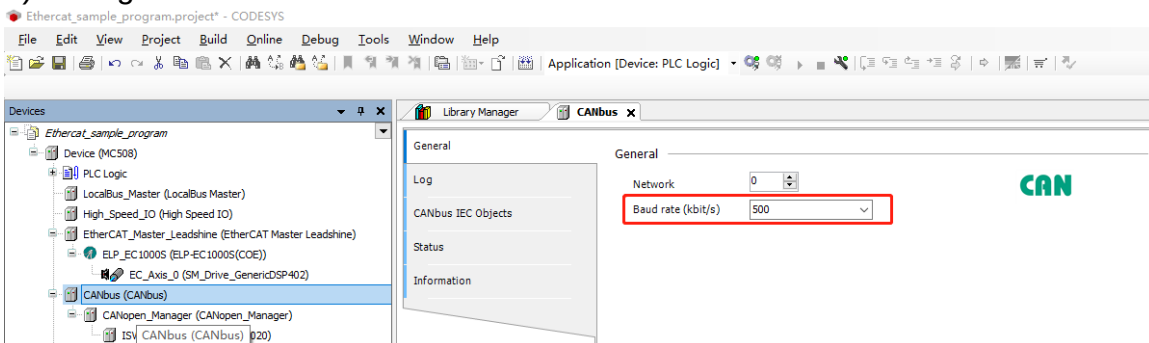
2) Right click "CANbus" to add the CANopen Manager



3) Auto-Scanning or offline to add the device(have to add the .eds device description file when use offline to add device)

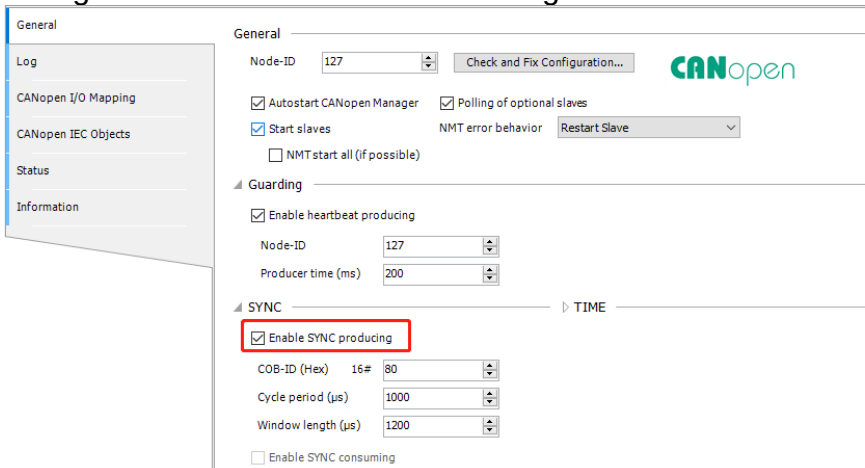


4) Configure the baud rate



9.3.2.CANopen bus manager configuration

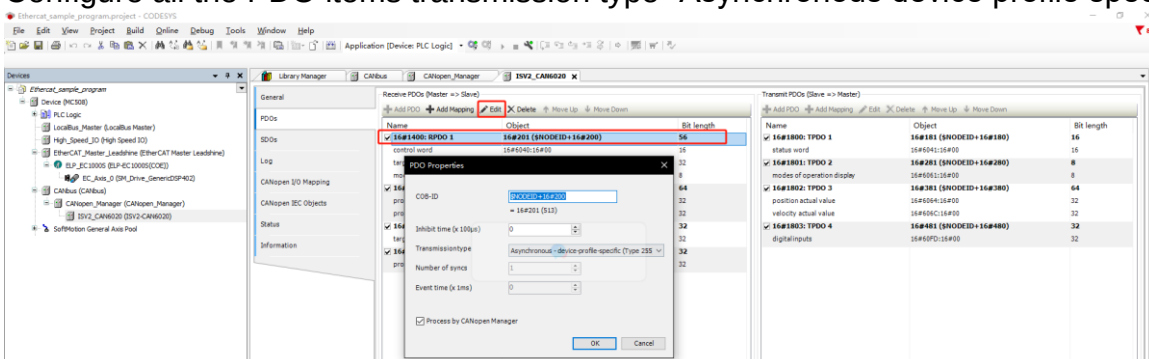
1) Communication configuration, select the “Enable SYNC producing”, the other configuration refer to the default setting.



9.3.3.CANopen slave configuration

1) PDO configuration

Configure all the PDO items transmission type “Asynchronous device profile specific”

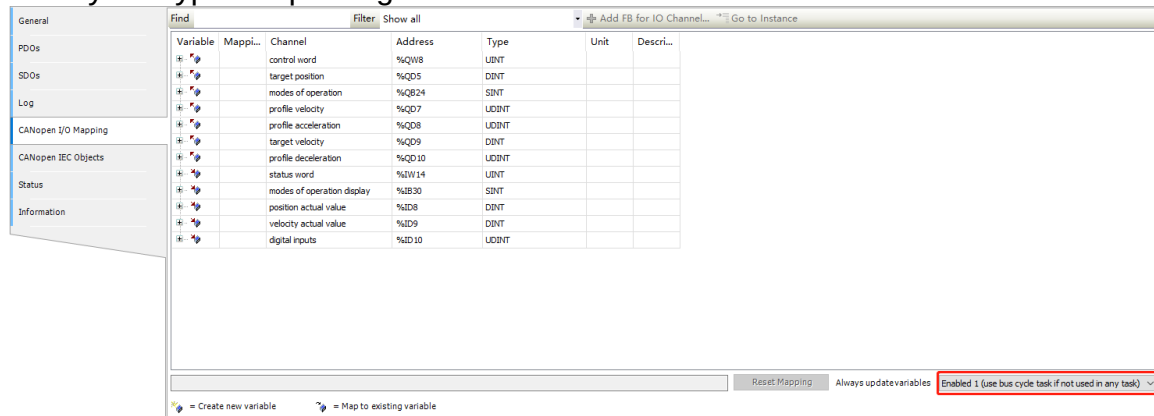


Please refer to the follow table to configure the PDO items order, and the extend PDO items only can be added at the table end.

1		Control Word 16#6040	TPDO1	Status Word 16#6041
2	RPDO1	Target Position 16#607A	TPDO2	Modes Of Operation Display 16#6061
3		Modes Of Operation 16#6060		
4	RPDO2	Profile Velocity 16#6081	TPDO3	Position Actual Value 16#6064
5		Profile Acceleration 16#6083	TPDO4	Velocity Actual Value 16#606C
6	RPDO3	Target Velocity 16#60FF		
7	RPDO4	Profile Deceleration 16#6084		

2) I/O mapping configuration

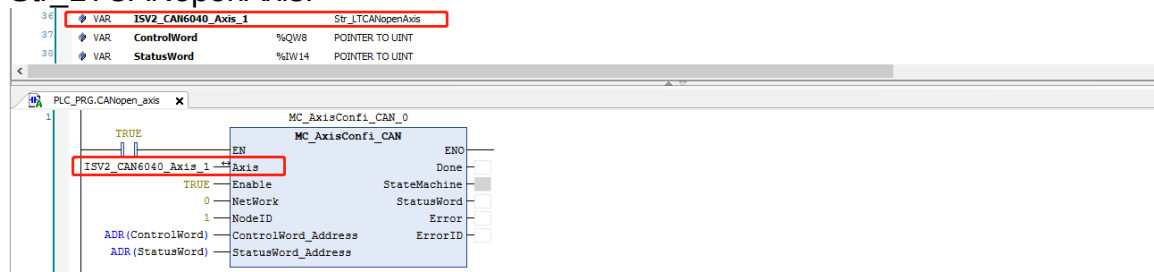
Modify the type of updating variables



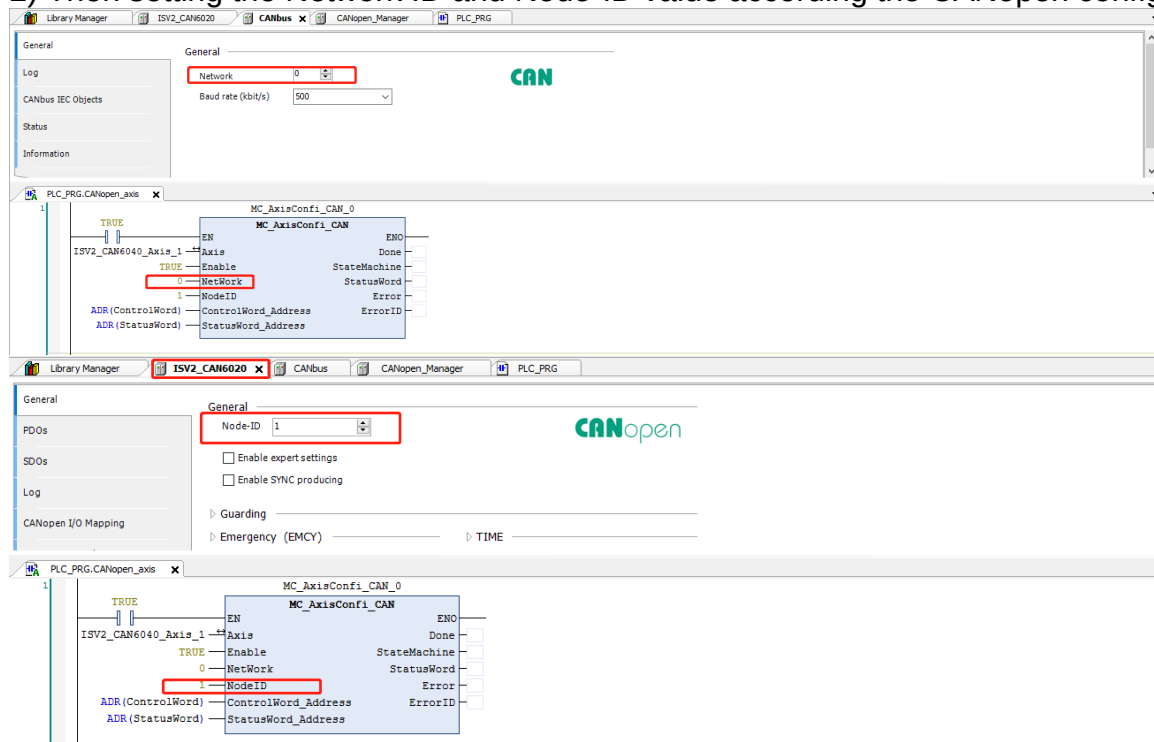
9.3.4.CANopen axis motion program

1) Create axis variable to bind the CANopen slave

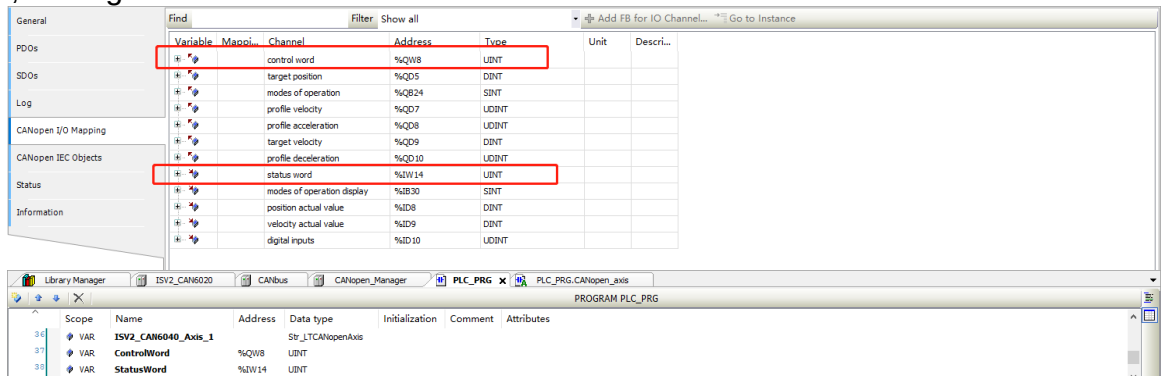
Add "MC_AxisConfI_CAN" function block, define axis variable which data type is Str_LTCANopenAxis.



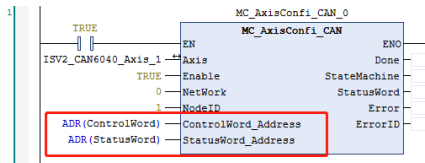
2) Then setting the Network ID and Node-ID value according the CANopen configuration



3) Refer the CANopen I/O mapping table, define the address of control word and status word, setting the control word and status word via ADR command

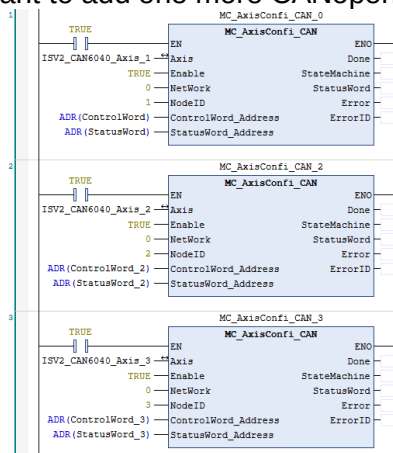


Variable	Mapping	Channel	Address	Type	Unit	Descri...
control word	%QW8		%QW8	UDINT		
target position	%QD5		%QD5	DINT		
modes of operation	%QB24		%QB24	SINT		
profile velocity	%QD7		%QD7	UDINT		
profile acceleration	%QD8		%QD8	UDINT		
target velocity	%QD9		%QD9	DINT		
profile deceleration	%QD10		%QD10	UDINT		
status word	%IW14		%IW14	UDINT		
modes of operation display	%B30		%B30	SINT		
position actual value	%ID8		%ID8	DINT		
velocity actual value	%ID9		%ID9	DINT		
digital inputs	%ID10		%ID10	UDINT		

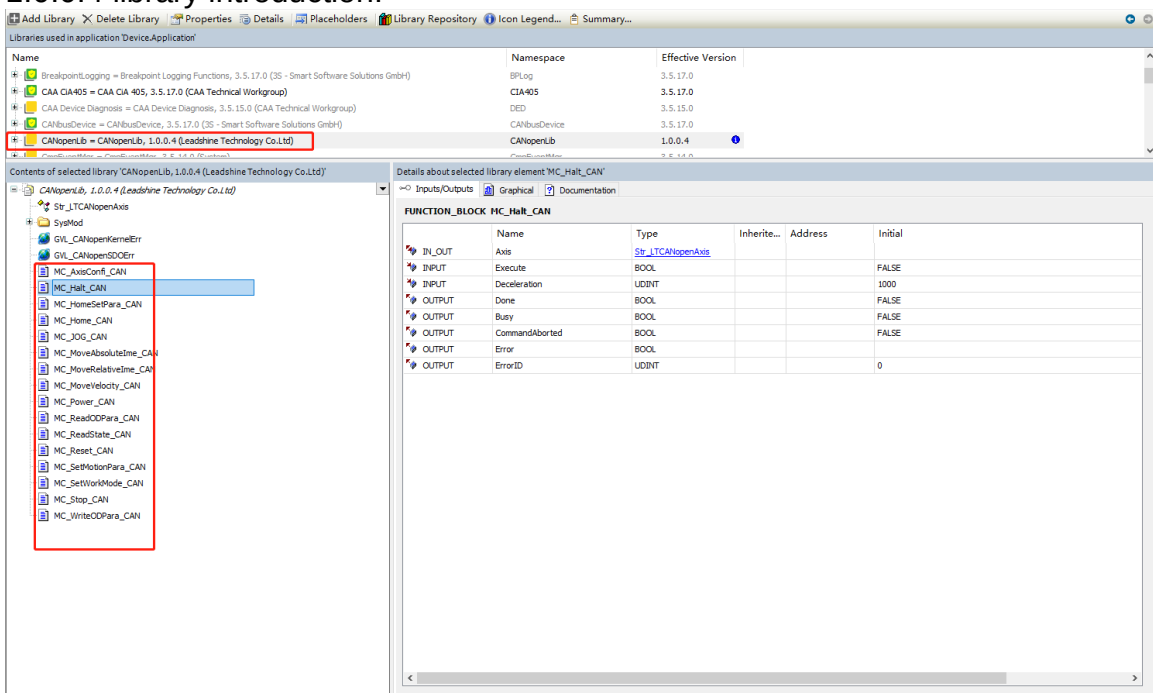


Note:

1. The information of "Str_LTCANopenAxis" please refer to the library "CANopenLib 1.0.0.0".
2. User have to bind the Network-ID and Node-ID to the "MC_AxisConfi_CAN" function block when want to add one more CANopen axis, as the follow picture shows.



4) Add CANopen axis motion command, about the details please refer to the CANopenLib 1.0.0.4 library introduction.



Libraries used in application Device.Application

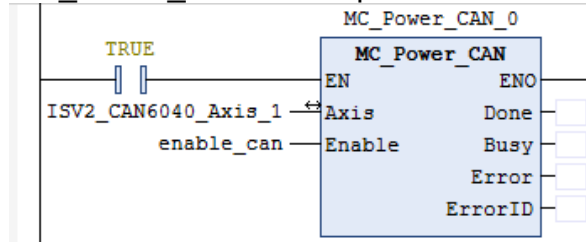
Name	Namespace	Effective Version
BreakpointLogging = BreakpointLogging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.17.0
CAA CIA405 = CAA CIA 405, 3.5.17.0 (CAA Technical Workgroup)	CIA405	3.5.17.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.15.0 (CAA Technical Workgroup)	DED	3.5.15.0
CANbusDevice = CANbusDevice, 3.5.17.0 (3S - Smart Software Solutions GmbH)	CANbusDevice	3.5.17.0
CANopenLib = CANopenLib, 1.0.0.4 (Leadshine Technology Co.Ltd)	CANopenLib	1.0.0.4

Contents of selected library "CANopenLib, 1.0.0.4 (Leadshine Technology Co.Ltd)"

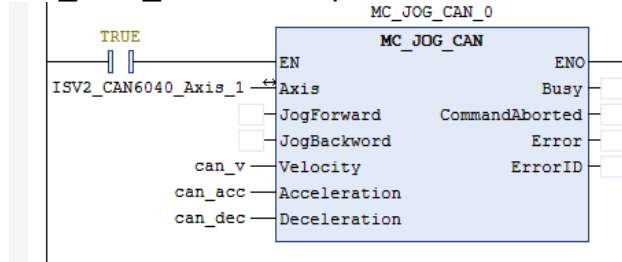
Details about selected library element "MC_Halt_CAN"

FUNCTION_BLOCK MC_Halt_CAN	Name	Type	Inherit...	Address	Initial
IN_OUT	Axis	Str_LTCANopenAxis			
INPUT	Execute	BOOL			FALSE
INPUT	Deceleration	UDINT			1000
OUTPUT	Done	BOOL			FALSE
OUTPUT	Busy	BOOL			FALSE
OUTPUT	CommandAborted	BOOL			FALSE
OUTPUT	Error	BOOL			
OUTPUT	ErrorID	UDINT			0

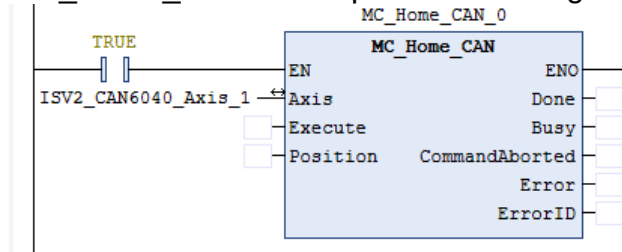
MC_Power_CAN: CANopen axis enable function block.



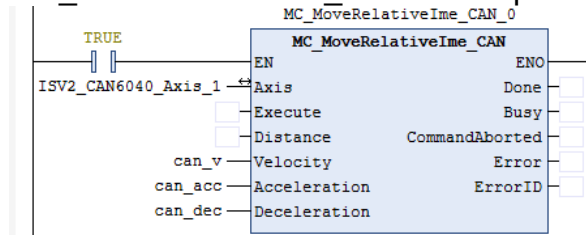
MC_JOG_CAN: CANopen axis JOG function block (support online change speed)



MC_Home_CAN: CANopen axis homing function block



MC_MoveRelativeIme_CAN: CANopen axis relative move function block.



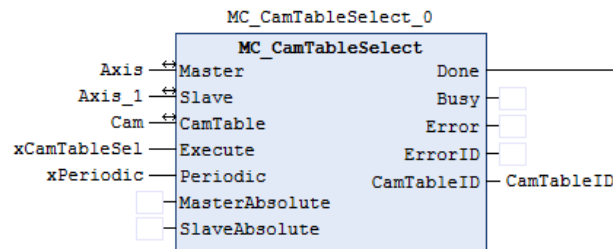
About other motion command, please refer to the function library introduction

9.4.E-CAM function

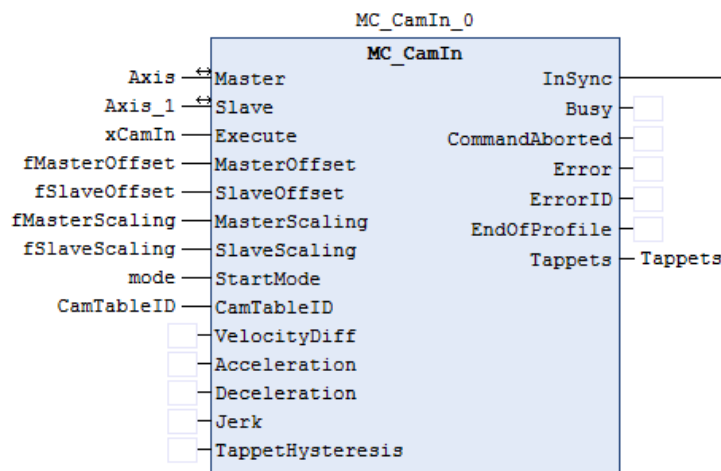
Generally, by digitizing cam movements, the problems of low precision, easy wear and noise in mechanical cams can be solved. MC500 series PLC provide the CAM graph and CAM function block to achieve the E-CAM function.

9.4.1.Function Block

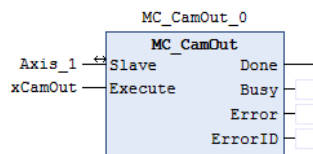
1) MC_CamTableSelect



2) MC_CamIn



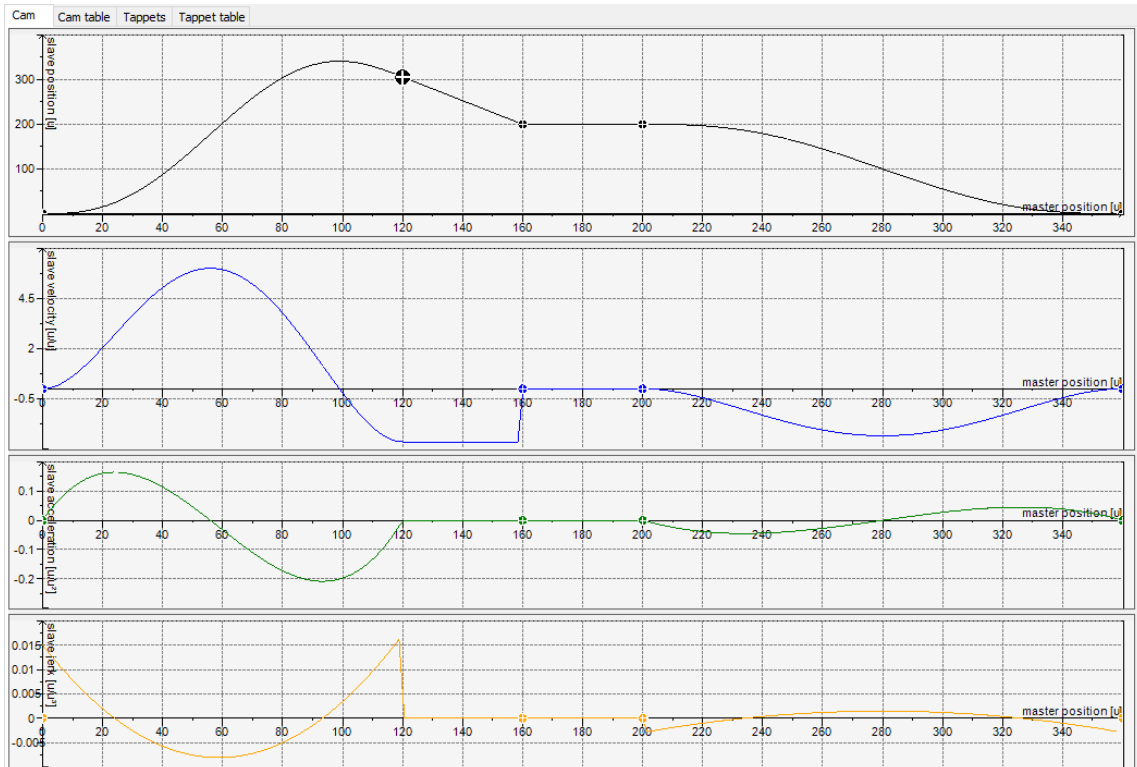
3) MC_CamOut



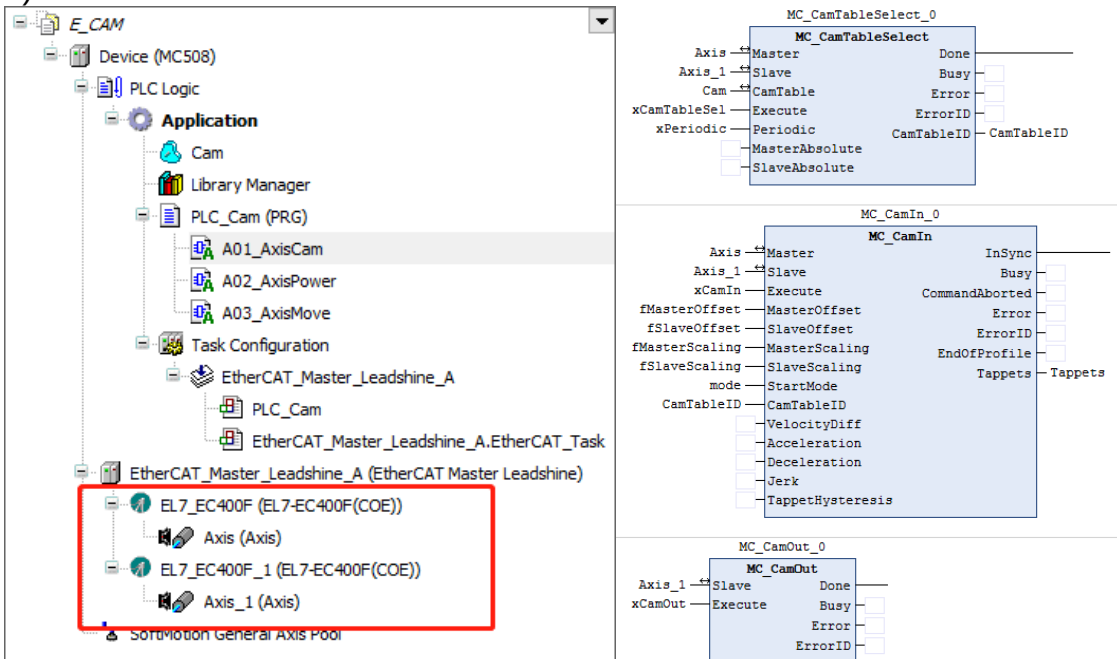
9.4.2.Sample Program

1) Add the CAM to application then change the CAM table, or change the key points of the graph manually

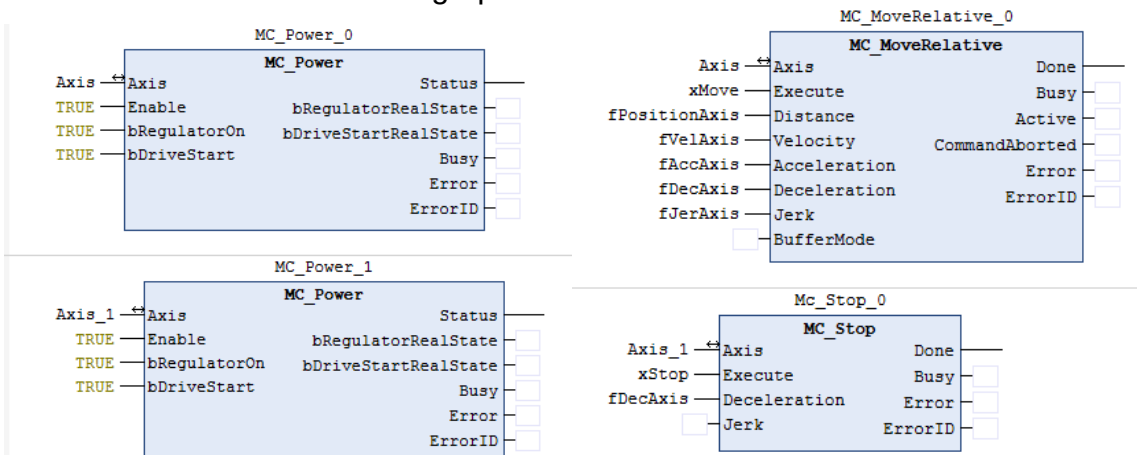
Cam	Cam table	Tappets	Tappet table								
	X		Y	V	A	J	Segm...	min(P...	max(P...	max(Velocity)	max(Acceleration)
	0		0	0	0	0					
+							Poly5	0	341.29...	6.0091941755686324	0.2089733388172457
+	120	306.02410888671875	-2.6506027221679687		0	0					
+	160		200	0	0	0	Line	200	306.02...	2.6506027221679687	0
+							Line	200	200	0	0
+	200		200	0	0	0					
+							Poly5	0	200	2.34375	0.04510548978043951
	360		0	0	0	0					



2) Add master and slave axis and E-CAM function blocks.

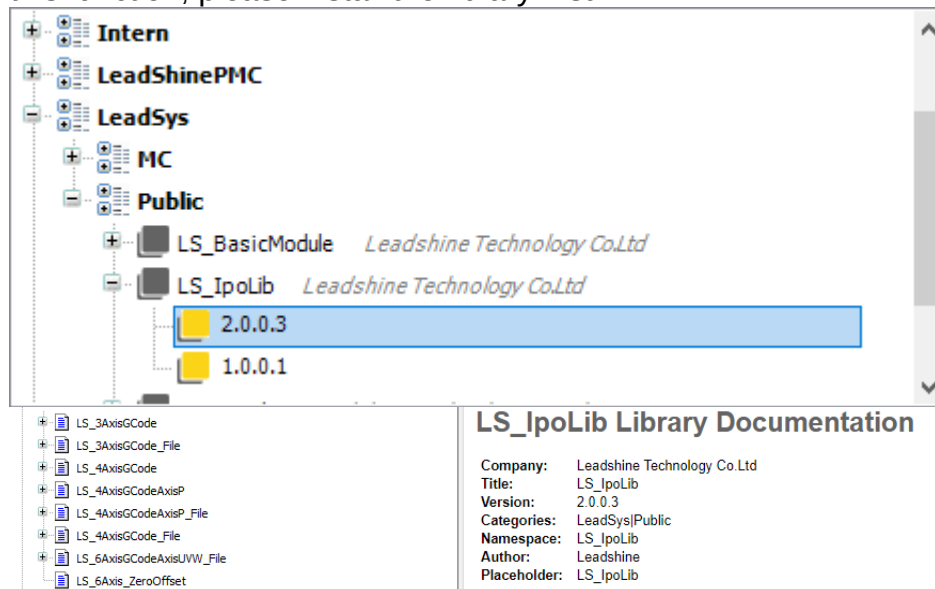


3) Add axis control blocks, execute the “MC_MoveRelative” block then the master axis and slave axis will follow the CAM graph rotate.



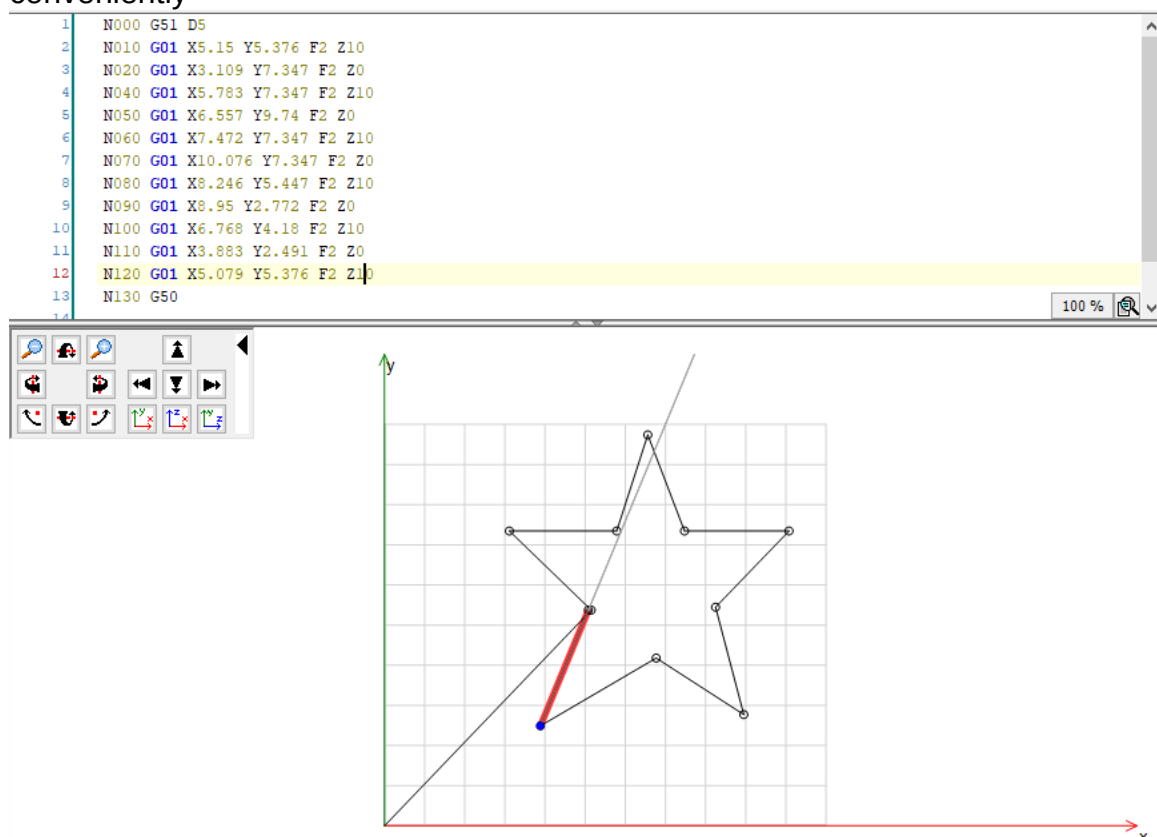
9.5.G-code Function

MC500 series PLC support standard G-code file, user can define the G-code in the program or import the processing file into the PLC (MC500 only support .cnc type G-code file yet.). Leadshine have developed the motion control libraries “LS_lplib”, before using this function, please install the library first.

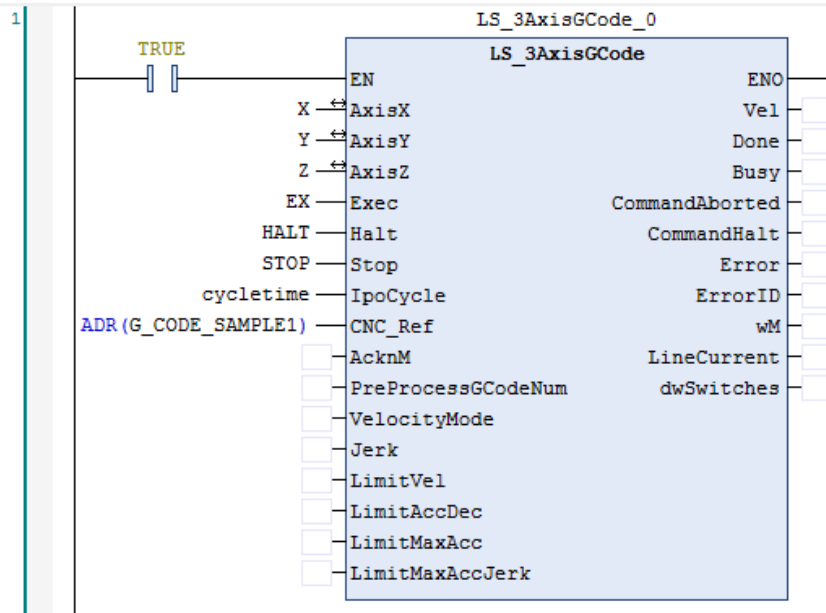


9.5.1.G-code in PLC program

User can add the CNC program into the application bar, and change the processing path conveniently

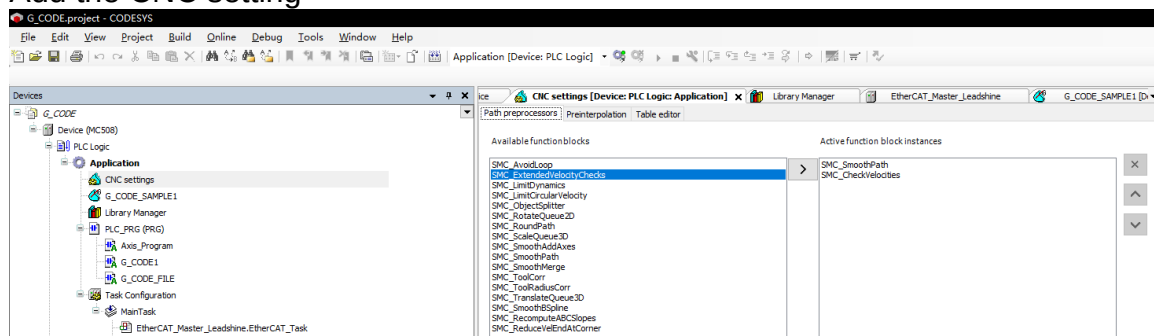


2) Function block

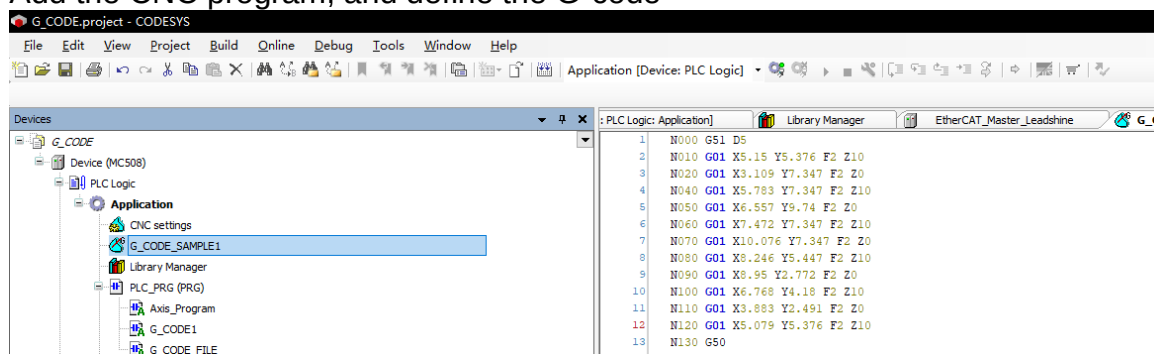


3) Sample program

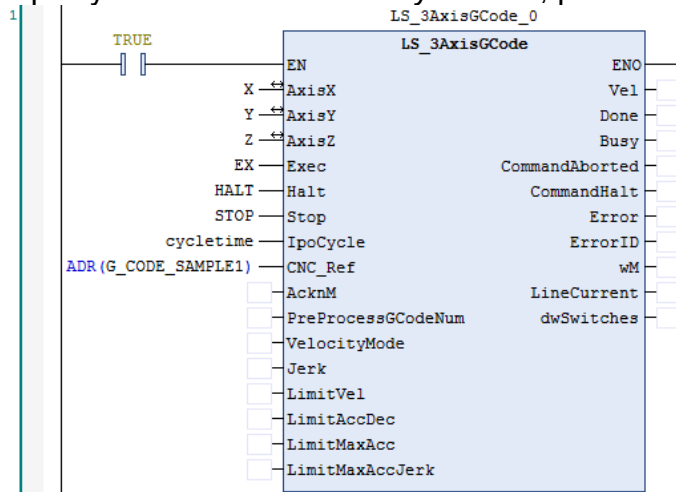
Add the CNC setting



Add the CNC program, and define the G-code



Add the G-code function block. Define the axes and the name of the CNC program, and the "IpoCycle" is the POU task cycle time, please keep the same value.

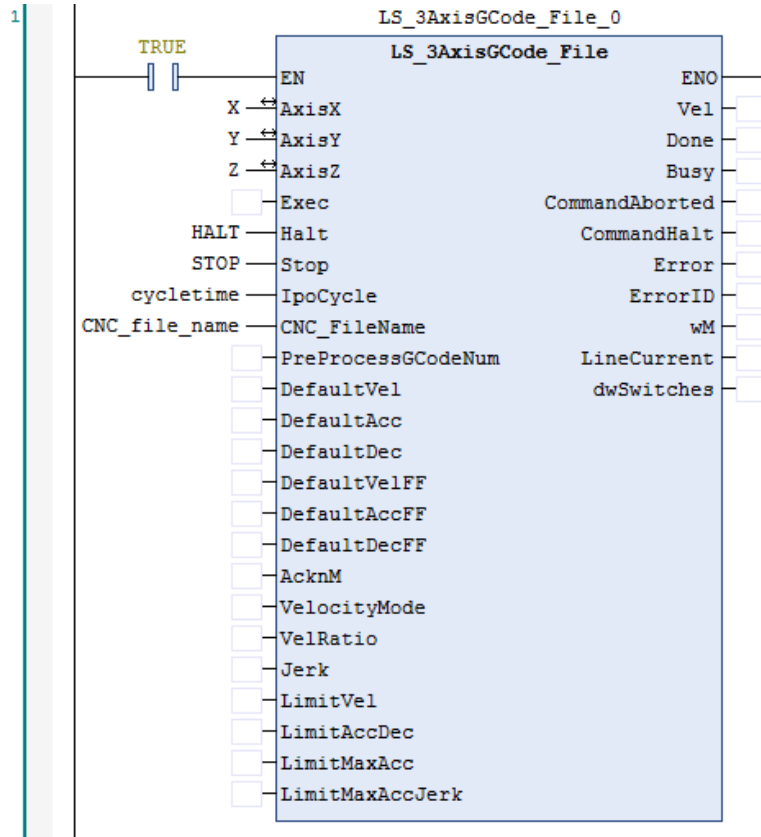


9.5.2.G-code in processing files

Open the .cnc file in the computer, then download it into PLC, and configure the function block to execute the G-code command.

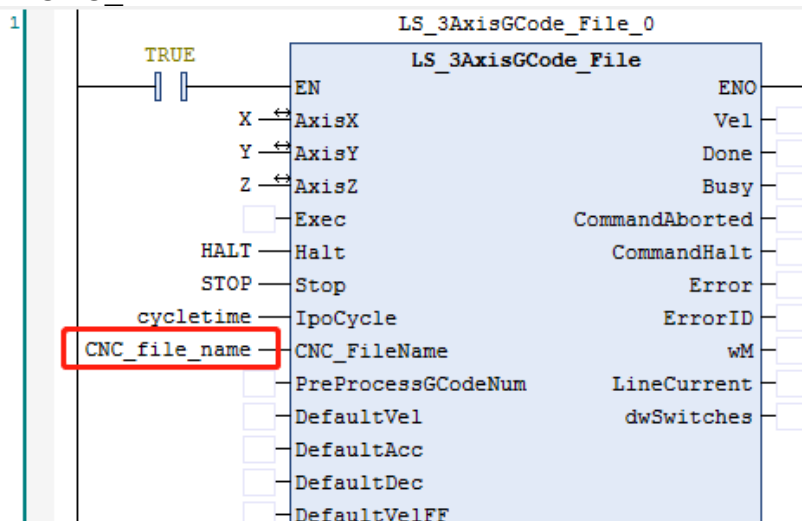
Path variable of the "LS_3AxisGCode_File" is /usr/src/CODESYSControl/UsrData/, so before execute the command, user need use file operation tool to copy the file to default path.

1) Function Block



2) Sample Program

Prepare the .cnc file, download it into the PLC, use file operation tool to copy the file to the path "/usr/src/CODESYSControl/UsrData/".add the function block "LS_3AxisGCode_File". the "CNC_FileName" have to be defined same with the file name.



Chapter 10 Motion Control Command

10.1.Command table

MC command	Description
MC_Power	Enable the axis
MC_SetPosition	Sets the axis position.
MC_ReadStatus	Reads the current axis status.
MC_Jog	Indicates jog control.
SMC_Inch	Controls motion in single step mode (the motion distance can be controlled).
MC_MoveRelative	Moves the axis in relative position mode
MC_MoveAdditive	Adds a specified movement distance from the previous axis position.
MC_MoveAbsolute	Moves the axis to a specified absolute position.
MC_MoveVelocity	Moves the axis at a constant speed.
MC_Halt	Stops axis motion (which can be interrupted).
MC_Stop	Stops axis motion (which cannot be interrupted).
MC_Reset	Resets the axis.
SMC3_ReinitDrive	Initialize Drive
SMC_ClearFBError	Deletes previous function block errors.
SMC_SetControllerMode	Sets the control mode.
SMC_SetTorque	Sets the torque.
MC_Home	Moves the axis to home.

10.2.Axis Variables

Variable name	Data type	Description
nAxisState	SMC_AXIS_STATE	Status of PLCopen axis state machine 0:Power_off: axis disable 1:ErrorStop: axis error stop 2:Stopping: stop 3:StandStill: enable 4:Discrete_Motion: in positioning motion 5:Continuous_Motion:in speed motion 6: Synchronized_Motion:in synchronized motion 7:Homing:in homing process
bRegulatorOn	BOOL	Enable axis (MC_Power. bRegulatorOn)
bDriveStart	BOOL	Disable the quick stop mechanism (MC_Power. bDriveStart)
bCommunication	BOOL	Symbol of axis communication normal
wCommunicationState	WORD	Status of axis communication 0~9: Initialize detection and associate the slave station with the synchronization axis 10~18:Initialize communication 19:Detect communication initialize completed 20~28:initialize 402 status machine and SDO 80~89:Waiting for master station synchronize all slaves 90:initialize completed 100: Axis reaches operable state 200: Received axis reinitialization command 201~209: Preparing to reinitialize the axis 210: The axis data has been reinitialized and the communication status has been switched to the state 10(Run communication initialization) >=1000: Abnormal communication status
sTaskCycle	LREAL	Task Cycle Time
dwRatioTechUnitsNum	DWORD	Axis Scaling Numerator
iRatioTechUnitsNum	DINT	Axis Scaling Denominator
nDirection	MC_DIRECTION	Axis Direction (MC_MoveVelocity.Direction)

fFactorACC	LREAL	Acceleration Scaling Ratio
fFactorTor	LREAL	Torque Scaling Ratio
fFactorJerk	LREAL	Jerk Scaling Ratio
iMovementType	INT	Axis Type,0:Modulo ;1:Linear
fPositionPeriod	LREAL	Modulus Value
eRampType	SMC_RAMPTYPE	Velocity Ramp Type
byControllerMode	BYTE	Axis Control Mode (SMC_SetControllerMode)
byRealControllerMode	BYTE	Feedback Of Axis Control Mode
fSetPosition	LREAL	Set Position
fActPosition	LREAL	Actual Position
fAimPosition	LREAL	Target Position
fSetVelocity	LREAL	Set Velocity
fActVelocity	LREAL	Actual Velocity
fMaxVelocity	LREAL	Maximum Velocity
fSWMaxVelocity	LREAL	Limit Velocity
bConstantVelocity	BOOL	Reach Target Velocity
fSetAcceleration	LREAL	Set Acceleration
fActAcceleration	LREAL	Actual Acceleration
fMaxAcceleration	LREAL	Maximum Acceleration
diSetPosition	DINT	Set Position 607A
diActPosition	DINT	Actual Position 6063
diSetVelocity	DINT	Set Position 60FF
diActVelocity	DINT	Actual Velocity 606C
fSetJerk	LREAL	Set Jerk
fActJerk	LREAL	Actual Jerk
fSetTorque	LREAL	Set Torque%
fActTorque	LREAL	Actual Torque%
fSWLimitPositive	LREAL	Positive Limit Location
fSWLimitNegative	LREAL	Negative Limit Location
usiSWEndSwitchState	USINT	Status Of Axis Limits
bSWLimitEnable	BOOL	Enable Limit Switch
fOffsetPosition	LREAL	Position Offset
dwPosOffsetForResiduals	DWORD	Residual Position Offset Error
dwOneTurn	DWORD	Modulus Pulse
dwLastPosition	DWORD	Pulse Count
iTurn	INT	Turns
dwActPosition	DWORD	Actual Position
bVirtual	BOOL	Virtual Axis Switch

Chapter 11 Libraries Description

11.1. Basic Libraries list

Name	Description	Note
LS_BasicModule	Leadshine basic module library	
MC_HSIO	Leadshine IEC controller high speed IO library	
LS_IpoLib	Leadshine interpolation motion library	
CANopenLib	Leadshine CANopen servo and stepper library	
LS_ModbusMasterlib	Leadshine MC500 series PLC Modbus RTU master station function block	
LS_SysLib	Leadshine system general function library	
LS_UtilsLib	Leadshine data process, filter	
MC_SysLib	Leadshine MC series controller system function library	

Q&A

1. After compiling the program, the message interface occur some error of the function library “MC_HSI0”.

Description	Project	Object
Generate code...		
C0231: Expression of type 'BOOL' expected in this place	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_Init_Para_ID [LS_MotionControl_P]
C0077: Unknown type: 'null'	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_Init_Para_ID [LS_MotionControl_P]
C0046: Identifier 'null' not defined	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_Init_Para_ID [LS_MotionControl_P]
C0231: Expression of type 'BOOL' expected in this place	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_Process_Axis [LS_MotionControl_P]
C0077: Unknown type: 'null'	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_Process_Axis [LS_MotionControl_P]
C0046: Identifier 'null' not defined	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_Process_Axis [LS_MotionControl_P]
C0231: Expression of type 'BOOL' expected in this place	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_SyncPos [LS_MotionControl_P]
C0077: Unknown type: 'null'	mc_hsi0, 1.0.2.3 (Leadshine technology co.ltd)	ACT_SyncPos [LS_MotionControl_P]

: Please check the project soft motion version, change it to 4.10.0.0 version

2. EtherCAT master can't find the slave drive

: Please check the EtherCAT source (MAC) address setting, refer to the chapter 6.3 to set the master station

General
Sync Unit Assignment
EtherCAT I/O Mapping
EtherCAT IEC Objects
Status

☒ Autoconfig master/slaves

Destination address (MAC)

FF-FF-FF-FF-FF-FF

☒ Broadcast
☐ Redundancy

Source address (MAC)

00-00-00-00-00-00

Select...

Network name

eth0

☐ Select network by MAC
☒ Select network by name

3. After compiling the program, message interface occur “Could not open library, reason the library has not been installed to the system”

Messages - Total 16 error(s), 0 warning(s), 0 message(s)		
Description	Project	Object
Could not open library 'MC_HSI0'. (Reason: The library 'MC_HSI0, 1.0.2.3 (Leadshine Technology Co.Ltd)' has not been installed to the system.)	MC500_Series_PLC_Sample_1	Library Manager [Device: PLC Logic: Application]
Could not open library 'MC_SysLib'. (Reason: The library 'MC_SysLib, 1.0.0.6 (Leadshine Technology Co.Ltd)' has not been installed to the system.)	MC500_Series_PLC_Sample_1	Library Manager [Device: PLC Logic: Application]
Could not open library 'LS_IpoLib'. (Reason: The library 'LS_IpoLib, 2.0.0.3 (Leadshine Technology Co.Ltd)' has not been installed to the system.)	MC500_Series_PLC_Sample_1	Library Manager [Device: PLC Logic: Application]
Could not open library 'LS_BasicModule'. (Reason: The library 'LS_BasicModule, 1.0.0.3 (Leadshine Technology Co.Ltd)' has not been installed to the system.)	MC500_Series_PLC_Sample_1	Library Manager [Device: PLC Logic: Application]

: Check the function libraries, installing all of the library files which provide by us.

4. After adding “LS_ModbusMaster”, occur error notice.

Messages - Total 2 error(s), 0 warning(s), 0 message(s)			
Build			
Description	Project	Object	Position
Build started: Application: Device: Application -----			
Typify code...			
Generate code...			
C0039: VAR_IN_OUT 'arstModbusConfig' must be assigned in call of 'LS_ModbusMaster'	MC500_Series_PLC_Sample_1	Modbus_Master [Device: PLC Logic: Application: PLC_PRG]	Network 2 / Operand 'LS_ModbusMaster_1'
Build complete -- 1 errors, 0 warnings : No download possible			

: Please define the function block input and outputs variables completely

Contact us

Leadshine Technology Co., Ltd.

Headquarters Address:

15-20/F, Block B, Nanshan i-Valley, No.3157, Nanshan District,
Shenzhen City, Guangdong Province, China

Tel: +86 755 26411692

Fax: +86 755 26402718

Website: www.leadshine.com

Emails: sales@leadshine.com

Leadshine Global Retailers Network



Get in touch with us or any of your local Leadshine certified retailers by visiting our global website.

Technical Support

Tel: 86-755-2641-8447

86-755-2641-8774 (Asia, Australia, Africa)

86-755-2665-5136 (North and South America)

86-755-8654-2465 (Europe)

Fax: 86-755-2640-2718

Email: tech@leadshine.com

Author Email: Sean@leadshine.com

Sales Hot Line

Tel: 86-755-2641-7674 (Asia, Australia, Africa)

86-755-2641-7617 (North and South America)

86-755-2640-9254 (Europe)

Email: sales@leadshine.com

Leadshine
overseas



Leadshine
America, Inc.

North America Office

Address: 26050 Towne
Centre Dr. Foothill
Ranch California
United States

Tel: 1-949-608-7270

Fax: 1-949-638-7298

Website:

www.leadshineusa.com

Emails:

sales@leadshineusa.com