INOVANCE

FORWARD, ALWAYS PROGRESSING

SV680-INT Series Servo Drive

Communication Guide

Industrial Automation　Intelligent Elevator　New Energy Vehicle　Industrial Robot　Rail Transit

Data code PS00015535A01

Preface

Introduction

The SV680-INT series servo drive is a high-end servo drive designed based on global-leading standards and high-end application needs. It is featured with high speed, high precision, high performance, and tuning-free function. Compliant with CE, UL, KC, EAC, UKCA and TUV certification requirements and top international quality standards, it is specially suitable for high-end applications.

Its power ranges from 0.05 kW to 7.5 kW. It supports Modbus, CANopen and EtherCAT communication protocols and carries necessary communication interfaces to work with the host controller for implementing a networked operation of multiple servo drives. The servo drive supports adaptive stiffness level setting, inertia auto-tuning, and vibration suppression for easy use. The drive, together with an MS1 series high-response servo motor (with ultra-low, low or medium inertia) equipped with a 23- or 26-bit single-turn/multi-turn absolute encoder, any third party servo motor, linear motor or DDR motor, serves to deliver a quiet and stable operation and accurate process control through features like fully closed-loop, internal process segment and gantry synchronization.

The drive also comes with features like safe torque off, dynamic braking, and brake output (external relay not needed) as standard and supports extension of seven kinds of functional safety and bus functional safety FSoE (the PINT version further offers 24V backup power) for continuous safe production. The drive aims to achieve quick and accurate position control, speed control, and torque control through high-performance solutions for automation equipment in such industries as electronic manufacturing, lithium batteries, manipulators, packaging, and machine tools.

This manual introduces the communication of the drive, including configuration of Modbus, CANopen, and EtherCAT communication and application cases.

Note

The speed of a servo motor and DDR motor is in RPM and DDL motor is in mm/s. RPM is used throughout the manual. Unless otherwise specified, an RPM value is equivalent to the mm/s one.

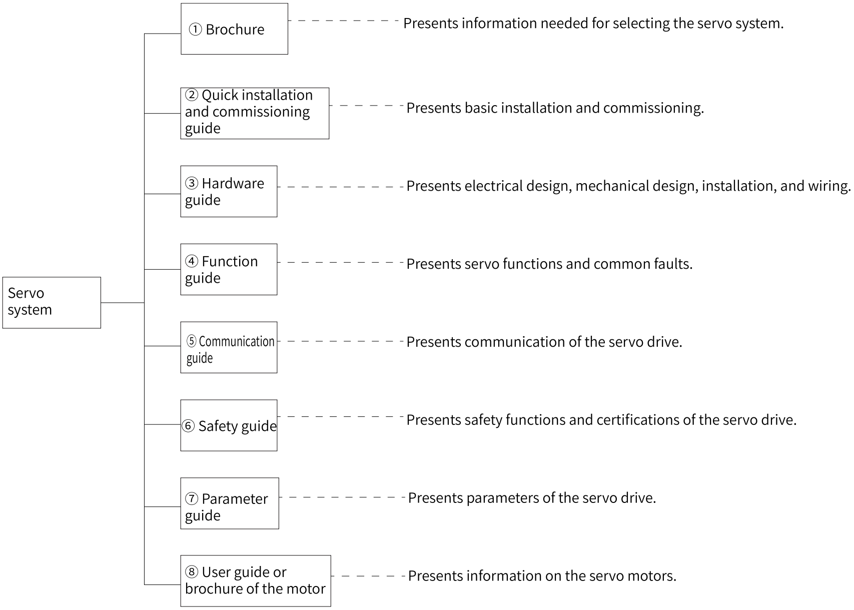
Abbreviation

The following abbreviations will be used herein to refer to the corresponding servo drives.

Abbreviation	Servo drive
[P]	SV680P*****-****
[N]	SV680N*****-****

More documents

The documents related to the drive are shown in the following figure and table.


Servo system
① Brochure
Presents information needed for selecting the servo system.
② Quick installation and commissioning guide
Presents basic installation and commissioning.
③ Hardware guide
Presents electrical design, mechanical design, installation, and wiring.
④ Function guide
Presents servo functions and common faults.
⑤ Communication guide
Presents communication of the servo drive.
⑥ Safety guide
Presents safety functions and certifications of the servo drive.
⑦ Parameter guide
Presents parameters of the servo drive.
⑧ User guide or brochure of the motor
Presents information on the servo motors.

No.	Name	Data Code	Description
①	SV680-INT series flagship servo drive	19120347	Provides instructions on product selection, including the list of supporting components, technical data on the drive, and the selection guide of cables.
②	SV680-INT Series Servo Drive Installation and Commissioning Quick Guide	PS00015536	Describes the model number, installation, terminals and quick commissioning and operation of the drive.
③	SV680-INT Series Servo Drive Hardware Guide	PS00015494	Describes technical data, installation, terminals, required certificates and standards and solutions to common EMC problems of the drive.
④	SV680-INT Series Servo Drive Function Guide	PS00015554	Introduces the functions and faults of the drive, including function overview, adjustment, basic servo functions and fault handling.
⑤	SV680-INT Series Servo Drive Communication Guide	PS00015535	Introduces the communication of the drive, including configuration of Modbus, CANopen, and EtherCAT communication.
⑥	SV680P-INT Series Servo Drive Safety Guide	PS00009740	Describes the safety function and related certifications and standards, wiring, commissioning process, troubleshooting and parameters of the drive.
	SV680N-INT Series Servo Drive Safety Guide	PS00009768	
⑦	SV680-INT Series Servo Drive Parameter Guide	PS00015555	Introduces the parameters of the drive, including a parameter list and description of parameters.

No.	Name	Data Code	Description
⑧	MS1-R Series Servo Motor Selection Guide	PS00004605	Introduces the product information, general specifications, motor selection, cable selection, and required certificates and standards of the servo motor.
	MS1-R Series Servo Motor Installation Guide	PS00005407	Describes installation of the motor, including an installation flowchart, unpacking and transportation, mechanical installation, and electrical installation.
	Direct drive motor module platform and drive	19120011	Introduces the product information, general specifications, motor selection, cable selection, and required standards of the motor.

Revision History

Date	Version	Description
2024-03	A01	Made minor corrections.
2024-02	A00	First release

Access to the Guide

This guide is not delivered with the product. You can obtain the PDF version in the following way:

- Visit *http://www.inovance.com*, go to Support > Download, search by keyword, and then download the PDF file.
- Scan the QR code on the product with your mobile phone.
- Scan the QR code below to install the app, where you can search for and download manuals.

Warranty

Inovance provides warranty service within the warranty period (as specified in your order) for any fault or damage that is not caused by improper operation of the user. You will be charged for any repair work after the warranty period expires.

Within the warranty period,maintenance fee will be charged for the following damage:

- Damage caused by operations not following the instructions in the user guide
- Damage caused by fire, flood, or abnormal voltage
- Damage caused by unintended use of the product

- Damage caused by use beyond the specified scope of application of the product
- Damage or secondary damage caused by force majeure (natural disaster, earthquake, and lightning strike)

The maintenance fee is charged according to the latest Price List of Inovance. If otherwise agreed upon, the terms and conditions in the agreement shall prevail.

For details, see the Product Warranty Card.

Table of Contents

Preface . . . 1
Fundamental Safety Instructions . . . 7
1 Communication Protocols . . . 14
2 Modbus Communication [P] . . . 15
2.1 Communication . . . 15
2.1.1 Communication technical data. . . . 15
2.1.2 Protocols . . . 15
2.2 Hardware Configuration . . . 15
2.2.1 Terminal Layout . . . 15
2.2.2 RS485 Communication Connection Example . . . 16
2.3 Communication Transmission Mode . . . 18
2.4 Data Frame Structure . . . 19
2.5 Communication Parameters . . . 27
3 CANopen Communication [P] . . . 29
3.1 Communication . . . 29
3.1.1 Communication Technical Data . . . 29
3.1.2 Protocols . . . 30
3.2 Hardware Configuration . . . 30
3.2.1 Terminal Layout . . . 30
3.2.2 CAN Communication Connection Example . . . 30
3.3 Communication Transmission Mode . . . 33
3.4 Data Frame Structure . . . 34
3.4.1 Network Management System (NMT) . . . 34
3.4.2 Service data object (SDO) . . . 39
3.4.3 Process Data Object (PDO) . . . 39
3.4.4 Synchronization Object (SYNC) . . . 45
3.4.5 Emergency (EMCY) Object Service . . . 48
3.4.6 SDO Transmission Message . . . 49
3.4.7 SDO transmission framework . . . 51
3.5 Communication Parameters . . . 52
3.6 PN-to-CANopen bridge . . . 53
4 EtherCAT Communication [N] . . . 56
4.1 Communication . . . 56
4.1.1 Communication technical data. . . . 56
4.1.2 Communication Specifications . . . 57
4.1.3 Protocols . . . 57
4.2 Hardware Configuration . . . 59

4.2.1 Terminal Layout . . . 59
4.2.2 EtherCAT Communication Connection Example . . . 60
4.3 Communication Transmission Mode. . . . 62
4.3.1 Structure of EtherCAT Communication . . . 62
4.3.2 Communication State Machine. . . . 63
4.3.3 Distributed clock. . . . 64
4.3.4 Status Indication. . . . 65
4.4 Data Frame Structure . . . 68
4.4.1 Process data . . . 68
4.4.2 Service Data Object (SDO) . . . 73
4.5 Communication Parameters . . . 73
5 Communication Configuration Instance. . . . 75
5.1 Modbus Communication Configuration Case [P] . . . 75
5.1.1 Communication Overview . . . 75
5.1.2 Wiring of Modbus RTU Communication Between SV680P-INT and Third-Party PLCs . 75
5.1.3 Servo Parameter Settings . . . 77
5.1.4 PLC Program Examples . . . 77
5.2 CANopen Communication Configuration Case [P] . . . 78
5.2.1 Connecting SV680P-INT to Schneider 3S Master . . . 78
5.2.2 Connecting SV680P-INT to Beckoff CANopen Master. . . . 97
5.2.3 Connecting SV680P-INT to Inovance H3U CANopen Master . . . 117
5.2.4 Connecting SV680P-INT to Inovance EASY CANopen Master. . . . 128
5.3 EtherCAT Communication Configuration Case [N]. . . . 138
5.3.1 SV680N-INT and AM600 Controller . . . 138
5.3.2 SV680N-INT and Omron Controller. . . . 147
5.3.3 SV680N-INT and Beckhoff Controller . . . 162
5.3.4 SV680N-INT and KEYENCE KV7500 Controller. . . . 179
5.3.4.1 Configuring the Servo Drive. . . .179
5.3.4.2 Configuring KEYENCE KV7500 Software Tool 179
5.3.4.3 Trial Run. . . .194
5.3.5 SV680N-INT and EASY Controller . . . 196

Fundamental Safety Instructions

Safety Precautions

- This chapter presents essential safety instructions for a proper use of the equipment. Before operating the equipment, read through the guide and comprehend all the safety instructions. Failure to comply with the safety precautions may result in death, serious injury, or equipment damage.
- "CAUTION", "WARNING", and "DANGER" items in the guide only indicate some of the precautions that need to be followed; they just supplement the safety precautions.
- Use this equipment according to the designated environment requirements. Damage caused by improper use is not covered by warranty.
- Inovance shall take no responsibility for any personal injuries or property damage caused by improper usage.

Safety Levels and Definitions

⚠ DANGER Indicates that failure to comply with the notice will result in death or severe personal injuries.

⚠ WARNING Indicates that failure to comply with the notice may result in death or severe personal injuries.

⚠ CAUTION Indicates that failure to comply with the notice may result in minor or moderate personal injuries or equipment damage.

Fundamental Safety Instructions

- Drawings in the guide are sometimes shown without covers or protective guards. Remember to install the covers or protective guards as specified first, and then perform operations in accordance with the instructions.
- The drawings in the guide are shown for illustration only and may be different from the product you purchased.
- Users must take mechanical precautions to protect personal safety and wear protective equipment, such as anti-smashing shoes, safety clothing, safety glasses, protective gloves, and protective sleeves.

Unpacking
⚠ WARNING • Do not install the equipment if you find damage, rust, or signs of use on the equipment or accessories upon unpacking. • Do not install the equipment if you find water seepage or missing or damaged components upon unpacking. • Do not install the equipment if you find the packing list does not conform to the equipment you received.
⚠ CAUTION • Check whether the packing is intact and whether there is damage, water seepage, dampness, and deformation before unpacking. • Unpack the package by following the unpacking sequence. Do not strike the package violently. • Check whether there is damage, rust, or injuries on the surface of the equipment and equipment accessories before unpacking. • Check whether the package contents are consistent with the packing list before unpacking.
Storage and Transportation
⚠ WARNING • Large-scale or heavy equipment must be transported by qualified professionals using specialized hoisting equipment. Failure to comply may result in personal injuries or equipment damage. • Before hoisting the equipment, ensure the equipment components such as the front cover and terminal blocks are secured firmly with screws. Loosely-connected components may fall off and result in personal injuries or equipment damage. • Never stand or stay below the equipment when the equipment is being hoisted by the hoisting equipment. • When hoisting the equipment with a steel rope, ensure the equipment is hoisted at a constant speed without suffering from vibration or shock. Do not turn the equipment over or let the equipment stay hanging in the air. Failure to comply may result in personal injuries or equipment damage.

⚠ CAUTION

- Handle the equipment with care during transportation and mind your steps to prevent personal injuries or equipment damage.
- When carrying the equipment with bare hands, hold the equipment casing firmly with care to prevent parts from falling. Failure to comply may result in personal injuries.
- Store and transport the equipment based on the storage and transportation requirements. Failure to comply will result in equipment damage.
- Avoid storing or transporting the equipment in environments with water splash, rain, direct sunlight, strong electric field, strong magnetic field, and strong vibration.
- Avoid storing the equipment for more than three months. Long-term storage requires stricter protection and necessary inspections.
- Pack the equipment strictly before transportation. Use a sealed box for long-distance transportation.
- Never transport the equipment with other equipment or materials that may harm or have negative impacts on this equipment.

Installation

⚠ DANGER

- The equipment must be operated only by professionals with electrical knowledge. Non-professionals are not allowed.

⚠ WARNING

- Read through the guide and safety instructions before installation.
- Do not install this equipment in places with strong electric or magnetic fields.
- Before installation, check that the mechanical strength of the installation site can bear the weight of the equipment. Failure to comply will result in mechanical hazards.
- Do not wear loose clothes or accessories during installation. Failure to comply may result in an electric shock.
- When installing the equipment in a closed environment (such as a cabinet or casing), use a cooling device (such as a fan or air conditioner) to cool the environment down to the required temperature. Failure to comply may result in equipment over-temperature or a fire.
- Do not retrofit the equipment.
- Do not fiddle with the bolts used to fix equipment components or the bolts marked in red.
- When the equipment is installed in a cabinet or final assembly, a fireproof enclosure providing both electrical and mechanical protections must be provided. The IP rating must meet IEC standards and local laws and regulations.
- Before installing equipments with strong electromagnetic interference, such as a transformer, install a shielding equipment for the equipment to prevent malfunction.
- Install the equipment onto an incombustible object such as a metal. Keep the equipment away from combustible objects. Failure to comply will result in a fire.

⚠ CAUTION

- Cover the top of the equipment with a piece of cloth or paper during installation. This is to prevent unwanted objects such as metal chippings, oil, and water from falling into the equipment and causing faults. After installation, remove the cloth or paper on the top of the equipment to prevent over-temperature caused by poor ventilation due to blocked ventilation holes.
- Resonance may occur when the equipment operating at a constant speed executes variable speed operations. In this case, install the vibration-proof rubber under the motor frame or use the vibration suppression function to reduce resonance.

Wiring

⚠ DANGER

- Equipment installation, wiring, maintenance, inspection, or parts replacement must be performed only by professionals.
- Before wiring, cut off all the power supplies of the equipment. and wait for at least the time designated on the equipment warning label before further operations because residual voltage still exists after power-off. After waiting for the designated time, measure the DC voltage in the main circuit to ensure the DC voltage is within the safe voltage range. Failure to comply will result in an electric shock.
- Do not perform wiring, remove the equipment cover, or touch the circuit board with power ON. Failure to comply will result in an electric shock.
- Check that the equipment is grounded properly. Failure to comply can result in electric shock.

⚠ WARNING

- Do not connect the input power supply to the output end of the equipment. Failure to comply can result in equipment damage or even a fire.
- When connecting a drive to the motor, check that the phase sequences of the drive and motor terminals are consistent to prevent reverse motor rotation.
- Cables used for wiring must meet cross sectional area and shielding requirements. The shield of the cable must be reliably grounded at one end.
- Fix the terminal screws with the tightening torque specified in the user guide. Improper tightening torque may overheat or damage the connecting part, resulting in a fire.
- After wiring is done, check that all cables are connected properly and no screws, washers or exposed cables are left inside the equipment. Failure to comply may result in an electric shock or equipment damage.

⚠ CAUTION

- Follow the proper electrostatic discharge (ESD) procedure and wear an anti-static wrist strap to perform wiring. Failure to comply may result in damage to the equipment or to the internal circuit of the product.
- Use shielded twisted pairs for the control circuit. Connect the shield to the grounding terminal of the equipment for grounding purpose. Failure to comply will result in equipment malfunction.

Power-on

⚠ DANGER

- Before power-on, check that the equipment is installed properly with reliable wiring and the motor can be restarted.
- Check that the power supply meets equipment requirements before power-on to prevent equipment damage or a fire.
- After power-on, do not open the cabinet door or protective cover of the equipment, touch any terminal, or disassemble any unit or component of the equipment. Failure to comply will result in an electric shock.

⚠ WARNING

- Perform a trial run after wiring and parameter setting to ensure the equipment operates safely. Failure to comply may result in personal injuries or equipment damage.
- Before power-on, check that the rated voltage of the equipment is consistent with that of the power supply. Failure to comply may result in a fire.
- Before power-on, check that no one is near the equipment, motor, or machine. Failure to comply may result in death or personal injuries.

Operation

⚠ DANGER

- The equipment must be operated only by professionals. Failure to comply will result in death or personal injuries.
- Do not touch any connecting terminals or disassemble any unit or component of the equipment during operation. Failure to comply will result in an electric shock.

⚠ WARNING

- Do not touch the equipment casing, fan, or resistor with bare hands to feel the temperature. Failure to comply may result in personal injuries.
- Prevent metal or other objects from falling into the equipment during operation. Failure to comply may result in a fire or equipment damage.

Maintenance

⚠ DANGER

- Equipment installation, wiring, maintenance, inspection, or parts replacement must be performed only by professionals.
- Do not maintain the equipment with power ON. Failure to comply will result in an electric shock.
- Before maintenance, cut off all the power supplies of the equipment and wait for at least the time designated on the equipment warning label.
- In case of a permanent magnet motor, do not touch the motor terminals immediately after power-off because the motor terminals will generate induced voltage during rotation even after the equipment power supply is off. Failure to comply will result in an electric shock.

⚠ WARNING

- Perform routine and periodic inspection and maintenance on the equipment according to maintenance requirements and keep a maintenance record.

Repair

⚠ DANGER

- Equipment installation, wiring, maintenance, inspection, or parts replacement must be performed only by professionals.
- Do not repair the equipment with power ON. Failure to comply will result in an electric shock.
- Before inspection and repair, cut off all the power supplies of the equipment and wait for at least the time designated on the equipment warning label.

⚠ WARNING

- Submit the repair request according to the warranty agreement.
- When the fuse is blown or the circuit breaker or earth leakage current breaker (ELCB) trips, wait for at least the time designated on the equipment warning label before power-on or further operations. Failure to comply may result in death, personal injuries or equipment damage.
- When the equipment is faulty or damaged, the troubleshooting and repair work must be performed by professionals that follow the repair instructions, with repair records kept properly.
- Replace quick-wear parts of the equipment according to the replacement instructions.
- Do not use damaged equipment. Failure to comply may result in death, personal injuries, or severe equipment damage.
- After the equipment is replaced, check the wiring and set parameters again.

Disposal

⚠ WARNING

- Dispose of retired equipment in accordance with local regulations and standards. Failure to comply may result in property damage, personal injuries, or even death.
- Recycle retired equipment by observing industry waste disposal standards to avoid environmental pollution.

Additional Precautions

Precautions for the dynamic brake

- Dynamic braking can only be used for emergency stop in case of failure and sudden power failure. Do not trigger failure or power failure frequently.
- Ensure that the dynamic braking function has an operation interval of more than 5 minutes at high speed, otherwise the internal dynamic braking circuit may be damaged.

- Dynamic braking is commonly used in rotating mechanical structures. For example, when a motor has stopped running, it keeps rotating due to the inertia of its load. In this case, this motor is in the regenerative state and short-circuit current passes through the dynamic brake. If this situation continues, the drive, and even the motor, may be burned.

Safety label

For safe equipment operation and maintenance, comply with the safety labels on the equipment. Do not damage or remove the safety labels. The following table describes the meaning of the safety labels.

Safety label	Description
危险 DANGER 高压注意 Hazardous Voltage 高温注意 High Temperature	• Never fail to connect the protective earth (PE) terminal. Read through the guide and follow the safety instructions before use. • Do not touch terminals within 15 minutes after disconnecting the power supply to prevent the risk of electric shock. • Do not touch the heatsink with power ON to prevent the risk of burn.

1 Communication Protocols

Supported Protocol	SV680P-INT	SV680N-INT
Modbus	√	×
CANopen	√	×
EtherCAT	×	√

2 Modbus Communication [P]

2.1 Communication

2.1.1 Communication technical data

Item		Specification
Modbus Basic performance of slave	Link layer protocol	RS485
	Application layer protocol	Modbus-RTU, GBT 19582.2-2008, custom command areas
	Baud rate	115200bps
	Duplex mode	Half-duplex
	Data format	8-N-1 ((1) 8-bit data, (2) check, (3) stop bit)

2.1.2 Protocols

The Modbus protocol is a common language applied to electronic controllers. Through this protocol, the controllers can communicate with each other and other devices. It has become a general industry standard. Thanks to this communication protocol, control devices produced by different manufacturers can be connected into an industrial network for centralized monitoring.

2.2 Hardware Configuration

2.2.1 Terminal Layout

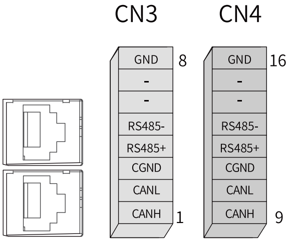


Figure 2-1 Communication Terminal pin layout of the servo drive

Table 2–1 Description of communication terminal pins

Pin No.	Description	Description
1 and 9	CANH	CAN communication port
2 and 10	CANL	
3 and 11	CGND	CAN communication GND
4 and 12	RS485+	RS485 communication port
5 and 13	RS485-	
6 and 14	-	-
7 and 15	-	-
8 and 16	GND	Ground
Enclosure	PE	Shield

2.2.2 RS485 Communication Connection Example

RS485 communication connection with PLC

The following figure shows the cable used for 485 communication between the servo drive and PLC.

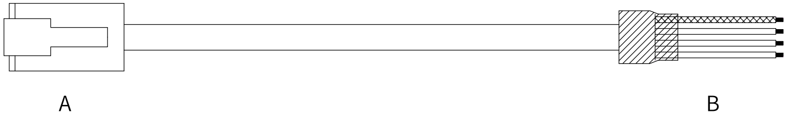


Figure 2-2 Outline drawing of cable used for CAN communication between the servo drive and PLC

Use a three-conductor shielded cable to connect the RS485 bus, with three conductors connected to 485+, 485-, and GND (GND represents non-isolated RS485 circuit) respectively. Connect RS485+ and RS485- with two conductors twisted together and connect the remaining conductor to the RS485 reference ground (GND). Connect the shield to the device ground (PE). Connect a 120Ω termination resistor on each end of the bus to prevent RS485 signal reflection.

Table 2–2 Pin connection relation of the cable used for CAN communication between the servo drive and PLC

RJ45 on the Drive (A)			PLC Side (B)		
Communication Type	Pin No.	Description	Communication Type	Pin No.	Description
RS485	4	485+	RS485	4	485+
	5	485-		5	485-
	8	GND		8	GND
-	Enclosure	PE (shield layer)	-	Enclosure	PE (shield layer)

RS485 communication connection for multi-drive applications

The following figure shows the cable used for parallel connection of multiple servo drives during RS485 communication.

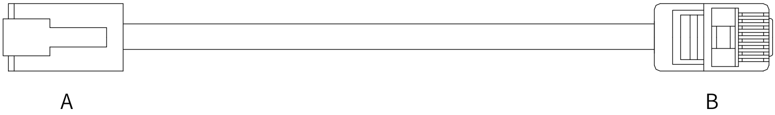


Figure 2-3 Outline drawing of multi-drive communication cable

Table 2–3 Pin connection relation of the cable used for multi-drive RS485 communication (pins in 485 group used only)

RJ45 on the Drive (A)			RJ45 on the Drive Side (B)		
Communication Type	Pin No.	Description	Communication Type	Pin No.	Description
RS485	4	485+	RS485	4	485+
	5	485-		5	485-
	8	GND		8	GND
-	Enclosure	PE (shield layer)	-	Enclosure	PE (shield layer)

In case of a large number of nodes, use the daisy chain mode for RS485 communication. Connect the reference grounds of RS485 signals of all the nodes (up to 128 nodes) together.

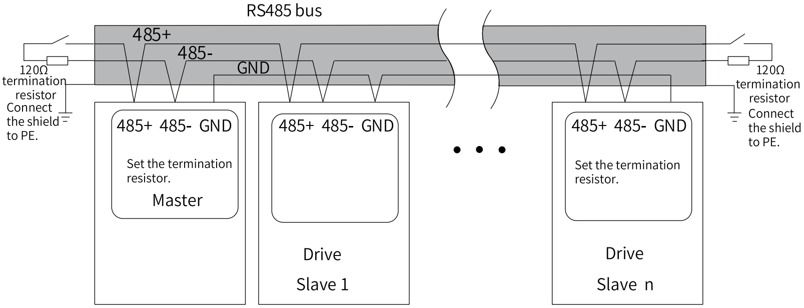


Figure 2-4 RS485 bus topology

Caution

Do not connect ⏚ (GND) terminal to the CGND terminal of the drive. Failure to comply may damage the machine.

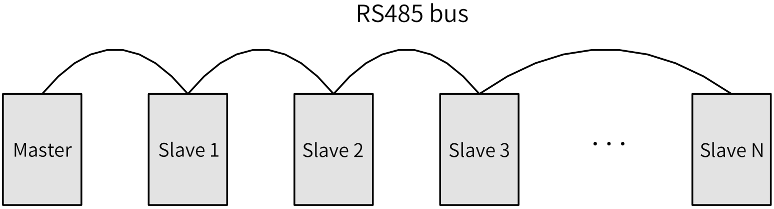


Figure 2-5 Daisy chain mode

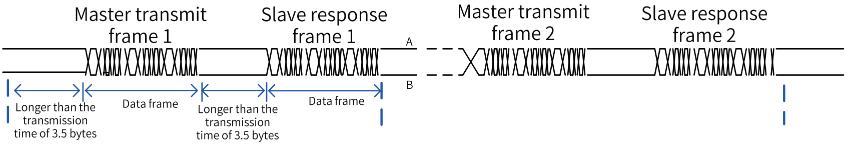
The following table lists the maximum number of nodes and transmission distance supported by the standard RS485 circuit at different transmission rate.

Table 2–4 Transmission distance and number of nodes

No.	Transmission Rate (kbps)	Transmission Distance (m)	Number of Nodes	Cable Size
1	115.2	100	128	AWG26
2	19.2	1000	128	AWG26

2.3 Communication Transmission Mode

In an RS485 communication network, data is transmitted in the asynchronous serial and half-duplex transmission mode. Data is sent frame by frame in the message format specified by the Modbus-RTU protocol. The idle time longer than 3.5-byte transmission time marks the start of a new communication frame.



The built-in communication protocol of the drive is the Modbus-RTU slave communication protocol, which allows the drive to respond to the query command from the master or execute the action according to query command from the master and respond with communication data.

The master can be a PC, an industrial control device, or a PLC, etc. The master can separately communicate with a slave or issue broadcast information to all slaves. When the master sends a query command to a single slave, the slave needs to return a response frame. For a broadcast message sent by the master, the slaves do not need to return a response to the master.

2.4 Data Frame Structure

Parameters of the SV680P-INT servo drive are divided into 16-bit and 32-bit parameters based on the data length. You can read and write parameters through the Modbus RTU protocol.

The command codes for reading/writing parameters vary with the data length.

Operation	Command code
Read 16-bit/32-bit parameters	0x03
Write 16-bit parameters	0x06
Write 32-bit parameters	0x10

Command code for reading parameter: 0x03

In Modbus RTU protocol, command code 0x03 is used to read both 16-bit and 32-bit parameters.

Request frame format:

Value	Description
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address: 1 to 247 **Note: 1 to 247 are decimal values which need to be converted into hexadecimal equivalents.**
CMD	Command code: 0x03

Value	Description
DATA[0]	Register start address (eight high bits): parameter group number of the start register Take H06.11 as an example, "06" is the group number, which means DATA[0] = 0x06. **Note: In this example, "06" is a hexadecimal value that needs no conversion.**
DATA[1]	Register start address (eight low bits): offset within the parameter group of the start register Take H06.11 as an example, "11" is the offset within the parameter group. That is, DATA [1] = 0x0B. **Note: In this example, ''11" is a decimal value that needs to be converted into the hexadecimal equivalent 0x0B.**
DATA[2]	Read the eight high bits N (H) of the number of parameters (hexadecimal)
DATA[3]	Read the eight low bits N (L) of the number of parameters (hexadecimal)
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

Response frame format:

Value	Description
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address, hexadecimal
CMD	Command code: 0x03
DATALENGTH	Number of parameter bytes, equal to reading the number of parameters N x 2
DATA[0]	Parameter data in the first register (eight high bits)
DATA[1]	Parameter data in the first register (eight low bits)
DATA[…]	…
DATA[N*2-2]	Parameter data in the Nth register (eight high bits)
DATA[N*2-1]	Parameter data in the Nth register (eight low bits)
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

In Modbus RTU protocol, command code 0x06 is used to write 16-bit parameters. Command code for writing 32-bit parameters: 0x10

Communication example

- To read data with a length of two words by taking H02.02 as the start register in the drive whose servo axis address is 01:
  Master request frame

01	03	02	02	00	02	CRCL	CRCH

Slave response frame:

01	03	04	00	01	00	00	CRCL	CRCH

The response frame indicates the slave returns data with a length of two words (four bytes), the content of which is 0x0001 and 0x0000.

If the slave response frame is as follows:

01	83	02	CRCL	CRCH

This response frame indicates a communication error occurs and the error code is 0x02. (0x83 indicates an error.)

- To read H05.07 (32-bit) in the drive whose servo axis address is 01:
  Master request frame

01	03	05	07	00	02	CRCL	CRCH

Slave response frame:

01	03	04	00	01	00	00	CRCL	CRCH

The preceding response frame indicates the value of H05.07 is 0x00000001.

Command code for writing 16-bit parameters: 0x06

⚠ Caution

Do not write 32-bit parameters with the command code 0x06. Failure to comply can result in unexpected error.

Request frame format:

Value	Description
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address 1 to 247 **Note: 1 to 247 are decimal values which need to be converted into hexadecimal equivalents.**
CMD	Command code: 0x06

Value	Description
DATA[0]	Register start address (eight high bits): parameter group number of the start register Take H06.11 as an example, "06" is the group number, which means DATA[0] = 0x06. **Note: In this example, "06" is a hexadecimal value that needs no conversion.**
DATA[1]	Register start address (eight low bits): offset within the parameter group of the start register Take H06.11 as an example, "11" is the offset within the parameter group, which means DATA[1] = 0x0B. **Note: In this example, ''11" is a decimal value that needs to be converted into the hexadecimal equivalent 0x0B.**
DATA[2]	Write the 8 high bits of register data (hexadecimal)
DATA[3]	Write the 8 low bits of register data (hexadecimal)
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

Response frame format:

Value	Description
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address, hexadecimal
CMD	Command code: 0x06
DATA[0]	Register start address (eight high bits): parameter group number of the start register Take H06.11 as an example, "06" is the group number, which means DATA[0] = 0x06. **Note: In this example, "06" is a hexadecimal value that needs no conversion.**
DATA[1]	Register start address (eight low bits): offset within the parameter group of the start register Take H06.11 as an example, "11" is the offset within the parameter group, which means DATA[1] = 0x0B. **Note: In this example, ''11" is a decimal value that needs to be converted into the hexadecimal equivalent 0x0B.**
DATA[2]	Write the 8 high bits of register data (hexadecimal)
DATA[3]	Write the 8 low bits of register data (hexadecimal)
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

Communication example

To write data 0x0001 to H02.02 in the drive whose servo axis address is 01:

Master request frame

01	06	02	02	00	01	CRCL	CRCH

Slave response frame:

01	06	02	02	00	01	CRCL	CRCH

This response frame indicates 0x0001 has been written to H02.02 in the drive whose servo axis address is 01.

If the slave response frame is as follows:

01	86	02	CRCL	CRCH

This response frame indicates a communication error occurs and the error code is 0x02. (0x86 indicates an error.)

Command code for writing 32-bit parameters: 0x10

⚠ Caution

Do not write 16-bit parameters with the command code 0x10. Failure to comply can result in unexpected error.

Request frame format:

Value	Description
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address 1 to 247 **Note: 1 to 247 are decimal values which need to be converted into hexadecimal equivalents.**
CMD	Command code: 0x10
DATA[0]	Register start address (eight high bits): parameter group number of the start register Take H11.12 as an example, "11" is the group number, which means DATA[0] = 0x11. **Note: In this example, "11" is a hexadecimal value that needs no conversion.**

Value	Description
DATA[1]	Register start address (eight low bits): offset within the parameter group of the start register Take H11.12 as an example, "12" is the offset within the parameter group, which means DATA[1] = 0x0C. **Note: In this example, ''12" is a decimal value that needs to be converted into the hexadecimal equivalent 0x0C.**
DATA[2]	Write the eight high bits M (H) of the number of parameters (hexadecimal) Take H05.07 as an example, DATA[2] is 00, DATA[3] is 02, and M is H0002. **For 32-bit parameters, each parameter is calculated as two words.**
DATA[3]	Write the eight low bits M (L) of the number of parameters (hexadecimal)
DATA[4]	Write the number of bytes (M x 2) corresponding to the register data Take H05.07 as an example, DATA[4] is H04.
DATA[5]	Write the eight high bits of the start register data (hexadecimal)
DATA[6]	Write the eight low bits of the start register data (hexadecimal)
DATA[7]	Write the eight high bits of the start register address +1 (hexadecimal)
DATA[8]	Write the eight low bits of the start register address +1 (hexadecimal)
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

Response frame format:

Value	Description
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address, hexadecimal
CMD	Command code: 0x10
DATA[0]	Register start address (eight high bits): offset within the parameter group of the start register Take H11.12 as an example, DATA[0] = 0x11.
DATA[1]	Register start address (eight low bits): offset within the parameter group of the start register Take H11.12 as an example, DATA[1] = 0x0C.
DATA[2]	Write the eight high bits M (H) of the number of parameters (hexadecimal)
DATA[3]	Write the eight low bits M (L) of the number of parameters (hexadecimal)

Value	Description
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

Error response frame

Error frame response format:

Value	
START	Equal to or larger than 3.5-character idle time, indicating the start of a frame
ADDR	Servo axis address, hexadecimal
CMD	Command code: 0x80
DATA[0]...[3]	DATA error code.
CRCL	CRC valid byte (low 8 bits).
CRCH	CRC valid byte (high 8 bits).
END	Equal to or larger than 3.5-character idle time, indicating the end of a frame

Error code:

Error code	Description
0x0001	Invalid command code
0x0002	Illegal data address
0x0003	Illegal data
0x0004	Slave device fault

32-bit parameter addressing

When 32-bit parameters are read/written through Modbus commands, the communication address is determined by the address of the parameter with lower offset number. Two offset numbers are operated in one operation.

Note

In the following examples, the servo axis address is 01 by default.

- The Modbus command for reading H11.12 (Displacement 1) is as follows:

01	03	11	0C	00	02	CRCL	CRCH

If the "1st displacement" is 0x40000000 (decimal equivalent: 1073741824), then the following response frames apply:

- When H0E.84 is set to 1 (Low 16 bits before high 16 bits), the response frame is as follows.

01	03	04	00	00	40	00	CRCL	CRCH

- When H0E.84 is set to 0 (High 16 bits before low 16 bits), the response frame is as follows.

01	03	04	40	00	00	00	CRCL	CRCH

- For example, the Modbus command for writing 0x12345678 to H11.12 (Displacement 1) is as follows.
  - If H0E.84 = 1 (Low 16 bits before high 16 bits):

01	10	11	0C	00	02	04	56	78	12	34	CRCL	CRCH

  - If H0E.84 = 0 (High 16 bits before low 16 bits):

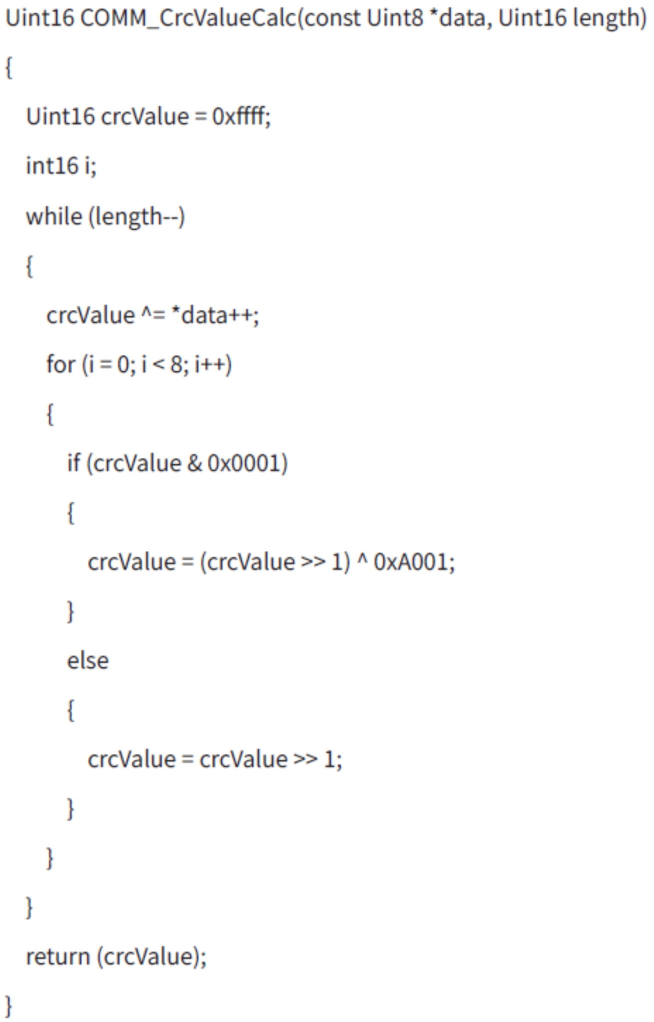
01	10	11	0C	00	02	04	12	34	56	78	CRCL	CRCH

- For example, to write 0x00100000 (decimal: 1048576) to the 32-bit parameter H05-07:
  When H0E.84 is set to 0 (High 16 bits before low 16 bits), the response frame is as follows.

01	10	05	07	00	02	04	00	00	00	10	CRCL	CRCH

CRC check

The host controller and the drive must use the same CRC algorithm during communication. Otherwise, a CRC error can occur. The servo drive uses 16-bit CRC with low byte before high byte. The CRC function is as follows: The polynomial used for CRC is $X^{16} + X^{15} + X^{2} + 1$ (0xA001).



```
Uint16 COMM_CrcValueCalc(const Uint8 *data, Uint16 length)
{
  Uint16 crcValue = 0xffff;
  int16 i;
  while (length--)
  {
    crcValue ^= *data++;
    for (i = 0; i < 8; i++)
    {
      if (crcValue & 0x0001)
      {
        crcValue = (crcValue >> 1) ^ 0xA001;
      }
      else
      {
        crcValue = crcValue >> 1;
      }
    }
  }
  return (crcValue);
}
```

2.5 Communication Parameters

Parameter	Default Value	Description	Remarks
H0E.00	1	Drive axis address	-
H0E.80	9	Baud rate of the serial port	9: 115200 bps

Parameter	Default Value	Description	Remarks
H0E.81	3	Modbus communication data format	3: No parity, 1 stop bit (8-N-1)
H0E.84	1	Modbus communication data sequence	0: High bits before low bits 1: Low bits before high bits

3 CANopen Communication [P]

3.1 Communication

3.1.1 Communication Technical Data

Item	Name	Description
Parameter setting	Node address switching	The node address can only be set manually. The maximum value is 127.
	Baud rate switching	The baud rate can only be set manually.
Description of state machine	State description/display of communication layer	Initializing, Pre-Operational, Operational, Stopped
	Description/display of emergency error codes	Time-Out, State-Switch-Err, PTO-Lend-Err
Error frame recording	Reception error frames can be recorded.	Count of NMT frames with incorrect length
		Count of NMT frames with incorrect command
		Count of heartbeat/node protection frames with incorrect length
Sync deviation detection	Multi-quantile sync deviation detection	1/4-period deviation
		1/2-period deviation
		3/4-period deviation
		1-period deviation
		2-period deviation
Baud rate	20K-1M baud rate	20Kbps, 50Kbps, 100Kbps, 125Kbps, 250Kbps, 500Kbps, 1Mbps
SYNC	SYNC Producer	Synchronous frame production
	SYNC Consumer	Synchronous signal consumption with deviation detection
SDO	Start domain upload/ download	Transmit data ⩽ 4 bytes
	SDO abort error	Report an SDO error code contextually
PDO	Synchronous TPDO	The sync number is 1–240. The default number of TPDOs/RPDOs is 4, which can be configured.
	Asynchronous TPDO	Time-triggered by time. The default number of TPDOs/RPDOs is 4, which can be configured.

Item	Name	Description
EMCY	**Emergency message**	Heartbeat timeout, PDO length error, node state switching error, application layer error
NMT	Bootup Service	Support for node online message transmit
NMTErrCtl	Life Guard	Optional node protection (cannot be used with heartbeat production)
	Heartbeat Consumer	Node heartbeat consumption
	Heartbeat Producer	Node heartbeat production
Expert mode	PDO communication parameters and their mapping are set through parameters.	PDO communication parameters and their mapping are set manually.

3.1.2 Protocols

CANopen is a protocol for the application layer of the network transmission system based on CAN serial bus. It complies with the ISO/ OSI standard model. Different devices in the network exchange data through the object dictionary or objects. The master node obtains or modifies data in the object dictionary of other nodes through PDO or SDO. The CANopen device model is shown in the following figure.

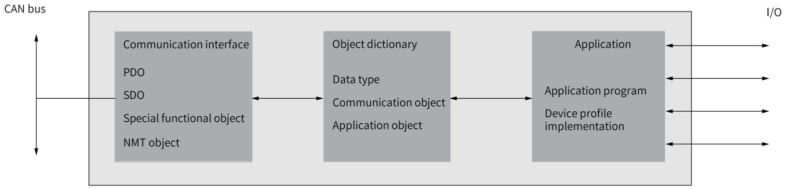


Figure 3-1 CANopen device model

3.2 Hardware Configuration

3.2.1 Terminal Layout

For details, see *"2.2.1 Terminal Layout" on page 15*.

3.2.2 CAN Communication Connection Example

CAN communication with PLC

The following figure shows the cable used for the communication between the servo drive and PLC in CAN communication networking.

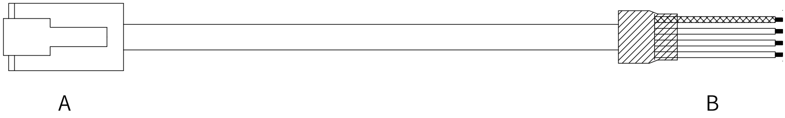


Figure 3-2 Outline drawing of cable used for CAN communication between the servo drive and PLC

Use a three-conductor shielded cable to connect the CAN bus, with the three conductors connected to CANH, CANL, and CGND (CGND represents isolated RS485 circuit) respectively. Connect CANH and CANL with twisted pairs. Connect CGND to the CAN reference ground. Connect the shield to the device ground. Connect a 120Ω termination resistor on each end of the bus to prevent CAN signal reflection.

Table 3–1 Pin connection relation of the cable used for CAN communication between the servo drive and PLC

RJ45 on the Drive Side (A)			PLC Side (B)		
Communication Type	Pin No.	Description	Communication Type	Pin No.	Description
CAN	1	CANH	CAN	1	CANH
	2	CANL		2	CANL
	3	CGND		3	CGND
-	Enclosure	PE (shield layer)	-	Enclosure	PE (shield layer)

CAN communication connection for multi-CAN applications

The following figure shows the cable used for parallel connection of multiple servo drives during CAN communication.

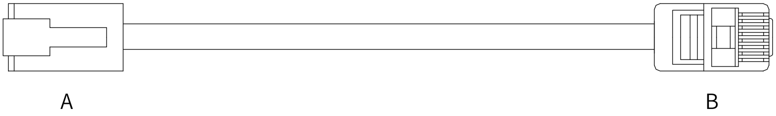


Figure 3-3 Outline drawing of multi-drive communication cable

Table 3–2 Pin connection relation of multi-drive communication cable (pins in CAN group used only)

RJ45 on the Drive (A)			RJ45 on the Drive Side (B)		
Communication Type	Pin No.	Description	Communication Type	Pin No.	Description
CAN	1	CANH	CAN	1	CANH
	2	CANL		2	CANL
	3	CGND		3	CGND
-	Enclosure	PE (shield layer)	-	Enclosure	PE (shield layer)

Use the daisy chain mode for CAN bus, as shown in the following figure.

- Shielded twisted pair cables are recommended for connecting the CAN bus. Twisted pairs are recommended for connecting CANH and CANL.
- Connect a 120Ω termination resistor on each end of the bus to prevent signal reflection.
- Connect the reference grounds of CAN signals of all the nodes together.
- Up to 64 nodes can be connected.

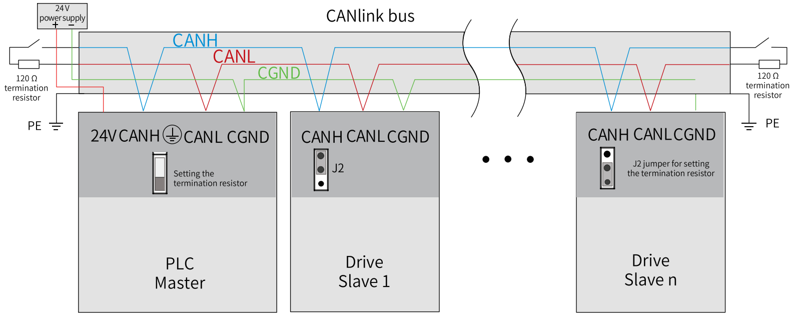


Figure 3-4 CAN bus topology

Caution

Do not connect the CGND terminal of the host controller to the GND terminal of the servo drive. Otherwise, the servo drive may be damaged.

3.3 Communication Transmission Mode

CANopen provides multiple communication objects. Every communication object has different features. You can select a communication object according to different applications. The predefined COB-ID is used. Specific rules are as follows:

- NMT object: 0x000
- SYNC object: 0x080
- SDO object:
  - Transmit SDO — 0x600+Node-Id
  - Receive SDO — 0x580+Node-Id
- PDO object:
  - RPDO1 — 0x200+Node-Id
  - RPDO1 — 0x300+Node-Id
  - RPDO1 — 0x400+Node-Id
  - RPDO1 — 0x500+Node-Id
  - RPDO1 — 0x180+Node-Id
  - RPDO1 — 0x280+Node-Id
  - RPDO1 — 0x380+Node-Id
  - RPDO1 — 0x480+Node-Id
- EMCY object: 0x80+Node-Id

Communication objects are defined as follows:

- NMT
  A network management object (NMT) includes Boot-up messages, Heartbeat protocol, and NMT messages. Based on the master-slave mode, an NMT is used to manage and monitor nodes in the network and implements three functions: node status control, error control, and node activation.

- SDO
  By using indexes and sub-indexes, SDOs enable clients to access entries in the object dictionary of devices. An SDO is achieved through a CMS object of the multi-element domain in CAL and transmitting data in any length is allowed. When the data exceeds four bytes, the data is divided into several packets. The SDO protocol produces a response for every message. SDO request and response packets alway contain eight bytes.

- PDO
  A PDO is used to transmit real-time data from one creator to one or multiple receivers. The length of transmitted data ranges from one to eight bytes. Every CANopen device contains eight default PDO channels, four PDO sending channels and four PDO receiving channels. The PDO supports synchronous and asynchronous transmission modes, which are determined by the communication

parameter corresponding to the PDO. The content of a PDO message is pre-defined and is determined by the mapping parameter corresponding to the PDO.

- SYNC object
  An SYNC object is a packet that is broadcast to the CAN bus periodically by the CANopen master. It is used to achieve basic network clock signals. Every device can determine whether to perform synchronous communication with other network devices using this event according to its own configurations.

3.4 Data Frame Structure

3.4.1 Network Management System (NMT)

The NMT initializes, starts, and stops the network and devices in the network. It belongs to the master-slave system. There is only one NMT master in the CANopen network. A CANopen network that includes the master can be configured.

NMT Service

CANopen works according to the state machine specified by the protocol. Some states are converted automatically and some must be converted through NMT messages transmitted by the NMT master, as shown below.

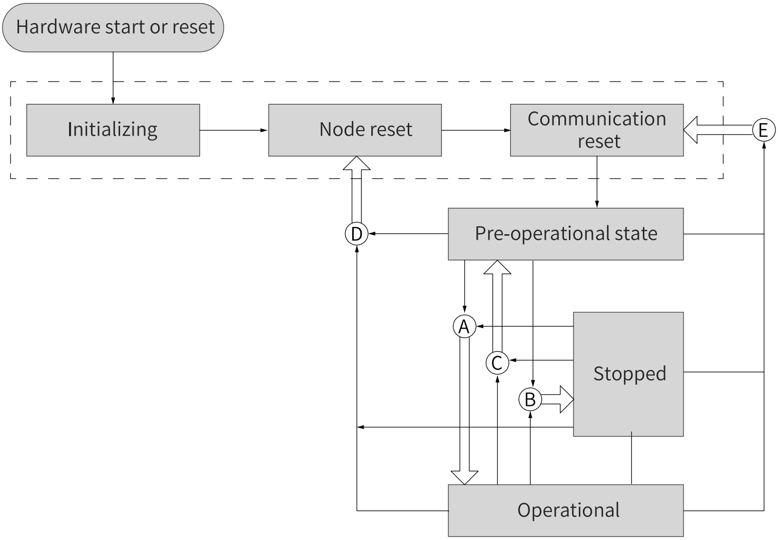


Figure 3-5 Execution process of NMT state machine

In the figure above, conversion marked with a letter is implemented through NMT messages and only the NMT master can send NMT control messages. The message format is shown in *"Table 3–3 " on page 35*.

Table 3–3 NMT message format

COB-ID	RTR	Data/Byte	
		0	1
0x000	0	Command word	Node_ID

The COB-ID of the NMT message is fixed to "0x000".

The data area contains two bytes. The first byte is a command word indicating this frame is for control purpose. See *"Table 3–4 " on page 35* for details.

The second byte (Node_ID) is the CANopen node address. The byte value 0 indicates it is a broadcast message and all slave devices in the network are active.

Table 3–4 NMT message command

Command word	Conversion Code	Description
0x01	A	Instruction for starting a remote node
0x02	B	Instruction for stopping a remote node
0x80	C	Instruction for entering the pre-operational status
0x81	D	Instruction for resetting a node
0x82	E	Instruction for resetting communication

After power-on, the device automatically enters the initialization state, including initializing, node reset, and communication reset. During initializing, parameters of each mode is loaded. During node reset, the manufacturer-defined area and profile area of the object dictionary are restored to values saved last time. During communication reset, communication parameters in the object dictionary are restored to values saved last time.

Next, the device sends Boot-up and enters the pre-operation status, which is the status of the main configuration nodes.

After configuration is done, the node can enter the operational status only after the NMT master sends the NMT message. When CANopen is working properly, it is in the operation status. All modules should work properly.

When the NMT master sends a node stop message, the device enters the stop state and only the NMT module works normally during CANopen communication.

The following table lists CANopen services available in various NMT status.

Table 3–5 Services supported in different NMT states

Service	Pre-operational	Operation	Stop
Process Data Object (PDO)	No	Yes	No
Service data object (SDO)	Yes	Yes	No
Synchronization Object (SYNC)	Yes	Yes	No
Emergency message (EMCY)	Yes	Yes	No
Network Management System (NMT)	Yes	Yes	Yes
Error control	Yes	Yes	Yes

NMT error control

NMT error control is used to detect whether devices in the network are online and the device status, including node guarding, life guarding, and heartbeat.

Note

- Life guarding and heartbeat cannot be used at the same time.
- Set the node guarding, life guarding, and heartbeat time to large values to prevent excessive network load.

- **Node/life guarding**

  In node guarding, the NMT master periodically check the NMT slave state through remote frames. In life guarding, the slave monitors the master state indirectly through the remote frame interval used to monitor the slave. Node guarding complies with the master/slave model. A response must be provided for each remote frame.

  Objects related to node/life guarding include the protection time 100Ch and life factor 100Dh. The value of 100Ch is the remote frame interval (ms) of node guarding under normal conditions. The product of 100Ch multiplied by 100Dh determines the latest time of master query. Node guarding is available normally. When 100Ch and 100Dh of a node are non-zero values and a node guarding request frame is received, life guarding will be activated.

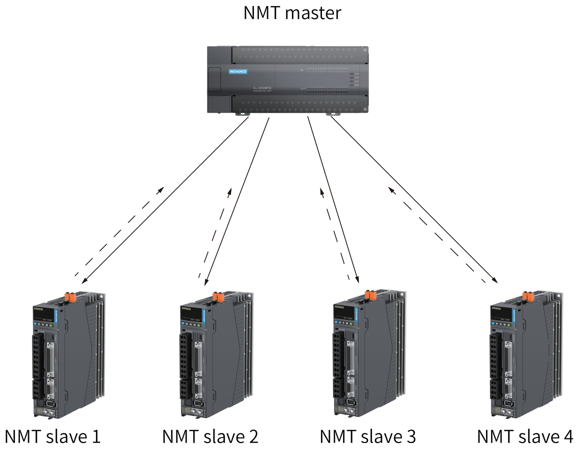


Figure 3-6 Description of node protection

As shown in the figure above, the master sends a node guarding remote frame at the interval defined by 100Ch, and the slave must respond to the remote frame. Otherwise, the slave is considered to be offline.

If the node guarding remote frame is not received by the slave within the time defined by 100Ch × 100Dh, the master is considered to be offline.

The following table describes the remote frame sent by the NMT master node.

Table 3–6 Node guarding remote frame message

COB-ID	RTR
0x700+Node_ID	1

The following table describes the response message returned by NMT from the slave. The data segment is a status word consisting of one byte.

Table 3–7 Response message of node guarding

COB-ID	RTR	Data
0x700 + Node-ID	0	Status word

Table 3–8 Description of response message state

Data bits	Description
bit7	It must be set to 0 or 1 alternatively.
Bit 6 to bit 0	4: Stopped 5: Operation status 127: Pre-operation status

Caution

It is recommended that the guarding time (100Ch) be at least 10 ms. The life factor must be greater than or equal to 2.

- **Heartbeat**

  The heartbeat mode adopts the producer—consumer model. The CANopen device can send heartbeat messages based on the cycle (ms) defined by the producer heartbeat interval object (1017h). In the network, there is always a node configured with the consumer heartbeat function, which monitors the producer based on the consumer time defined by object 1016h. Once the producer heartbeat is not received from the corresponding node within the consumer heartbeat time, a fault occurs on the node.

  After the producer heartbeat interval (1017h) is configured, the node heartbeat function is activated and a heartbeat message starts to be generated. After a valid sub-index is configured for consumer heartbeat (1016h) and a heartbeat frame is received from the corresponding node, monitoring starts.

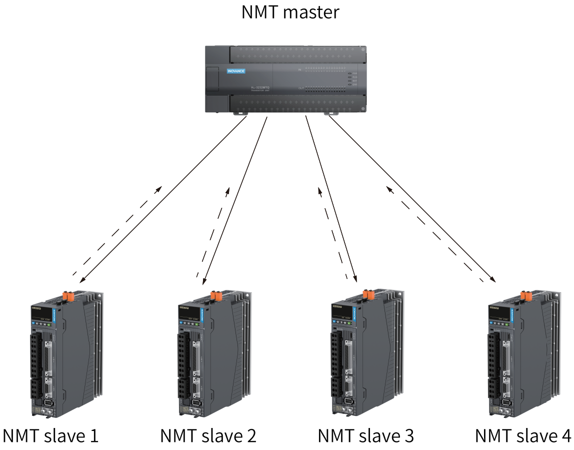


Figure 3-7 Heartbeat diagram

The master sends a heartbeat message based on the producer time. If the slave that monitors the master does not receive the heartbeat message within the time defined by the sub-index of 1016h , the master is considered to be offline. The time of the sub-index of 1016h must be longer than or equal to the master producer time multiplied by 1.8. Otherwise, a false report indicating the master is offline may occur.

The slave sends a heartbeat message at the interval defined by 1017h. If the master (or other slave) that monitors the slave does not receive the heartbeat message within the consumer time, the slave is considered to be offline. If 1017h multiplied by 1.8 is smaller than or equal to the consumer time of the master (or other slaves) that monitors the slave, a false report indicating the slave is disconnected may be reported.

The following table describes the format of a heartbeat message. The data segment contains only one byte. The most significant bit is permanently set to 0 and other bits are consistent with the response message status of node guarding, as shown in the following table.

Table 3–9 Heartbeat message

COB-ID	RTR	Data
0x700 + Node-ID	0	Status word

The SV680P-INT series servo drive is both a heartbeat producer and a heartbeat consumer. It can serve as the heartbeat consumer for five different nodes. It is recommended that the heartbeat producer time be set to a value not lower than 20 ms and the consumer heartbeat time be set to a value not lower than 40 ms (Consumer heartbeat time ＞ 1.8 x Producer heartbeat time).

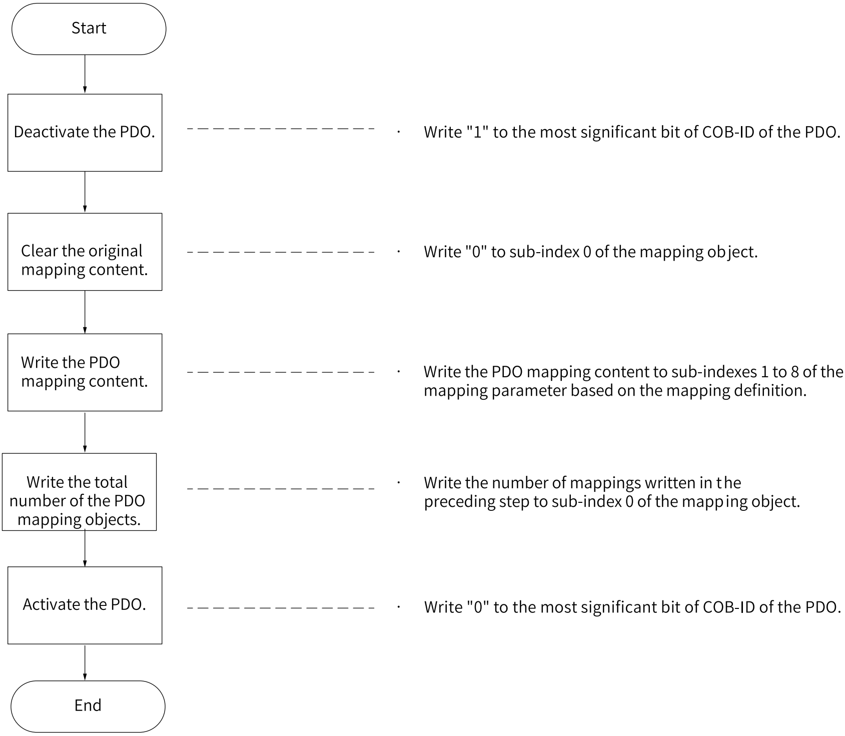
3.4.2 Service data object (SDO)

The SDO is associated with the object dictionary through object index and sub-index. Based on the SDO can read the object content in the object dictionary or modify the object data if allowed.

3.4.3 Process Data Object (PDO)

The PDO is used to transmit real-time data, which is the major data transmission mode in CANopen. PDO transmission features high speed as no response is required and the PDO may consist of less than eight bytes.

The following figure shows the PDO mapping configuration flowchart.


Start
Deactivate the PDO.
Write "1" to the most significant bit of COB-ID of the PDO.
Clear the original mapping content.
Write "0" to sub-index 0 of the mapping object.
Write the PDO mapping content.
Write the PDO mapping content to sub-indexes 1 to 8 of the mapping parameter based on the mapping definition.
Write the total number of the PDO mapping objects.
Write the number of mappings written in the preceding step to sub-index 0 of the mapping object.
Activate the PDO.
Write "0" to the most significant bit of COB-ID of the PDO.
End

PDO transmission framework

PDO transmission complies with the producer- consumer model, that is, in the CAN bus network, the TPDO generated by the producer may be received by one or multiple consumers in the network based on the COB-ID. The transmission model is shown in the following figure.

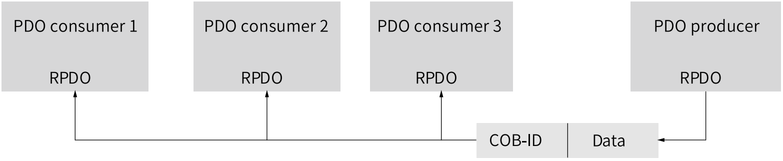


Figure 3-8 PDO transmission model

CANopen communication in SV680P-INT series servo drives only supports point-to-point PDO transmission.

PDO object

PDO can be divided into RPDO (Receive PDO) and TPDO (Transmitted PDO). The final PDO transmission mode and content are determined by communication parameters and mapping parameters. The SV680P-INT series servo drive uses four RPDOs and four TPDOs to transmit the PDO. The following table lists the related objects.

Table 3–10 PDOs of SV680P-INT servo drives

Name		COB-ID	Communication Object	Mapping Object
RPDO	1	200h + Node_ID	1400h	1600h
	2	300h + Node_ID	1401h	1601h
	3	400h + Node_ID	1402h	1602h
	4	500h + Node_ID	1403h	1603h
TPDO	1	180h + Node_ID	1800h	1A00h
	2	280h + Node_ID	1801h	1A01h
	3	380h + Node_ID	1802h	1A02h
	4	480h + Node_ID	1803h	1A03h

PDO Communication Parameters

- **CAN Identifier for PDO**

  The CAN identifier of a PDO, namely COB-ID, includes a control bit and identifier data and determines the bus priority of the PDO.

  The COB-ID is in the sub-index 01 of communication parameters (RPDO: 1400h to 1403h; TPDO: 1800h to 1803h). The most significant bit decides whether the PDO is valid.

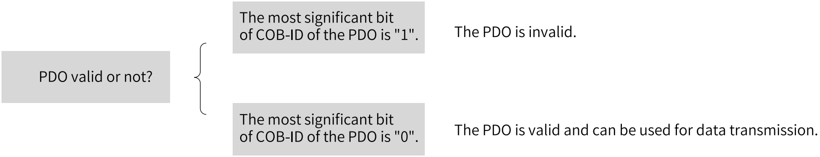

PDO valid or not?
The most significant bit of COB-ID of the PDO is "1".
The PDO is invalid.
The most significant bit of COB-ID of the PDO is "0".
The PDO is valid and can be used for data transmission.

Figure 3-9 Description of PDO validity

  The SV680P-INT servo drive only supports point-to-point PDO transmission. Therefore, the seven least significant bits of the COB-ID must be the station address of the node.

  Example:

  For the node whose station No. is 4, when TPDO3 is invalid, its COB-ID should be 80000384h. When 384h is written for the COB-ID, it indicates the PDO is activated.

- **Transmission type of PDO**

  The PDO transmission type parameter is in the sub-index 02h of communication parameters (RPDO: 1400h–1403h, TPDO: 1800h–1803h). It determines the transmission type of the PDO.

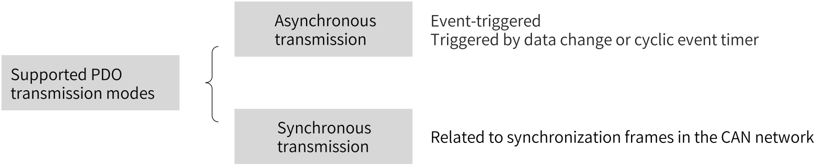

Supported PDO transmission modes
Asynchronous transmission
Event-triggered
Triggered by data change or cyclic event timer
Synchronous transmission
Related to synchronization frames in the CAN network

Figure 3-10 Supported PDO transmission mode

Communication parameters (RPDO: 1400h–1403h, TPDO: 1800H–1803h) Different values of the sub-index 02 stand for different transmission types and define the methods for triggering TPDO transmission or methods for processing received RPDOs. Table 3-26 lists methods for triggering TPDO and RPDO.

Table 3–11 Triggering Methods of TPDO and RPDO

Value of Communication Type	Synchronous		Asynchronous
	Cyclic	Acyclic	
0	-	✓	-
1–240	✓	-	-
241–253	-		
254, 255	-	-	✓

- When the transmission type of a TPDO is 0, if mapping data is changed and a synchronous frame is received, the TPDO is sent.
- When the transmission type of a TPDO is a value in the range 1 to 240 and a corresponding number of synchronous frames are received, the TPDO is sent.
- When the transmission type of a TPDO is 254 or 255, if mapping data is changed or the event timer expires, the TPDO is sent.
- When the transmission type of an RPDO is a value in the range 0 to 240, once a synchronous frame is received, the latest data of the RPDO is updated to the application; when the transmission RPDO of an RPDO is 254 or 255, the received data is directly updated to the application.

- **Disabled time**

  The disabled time is set for TPDOs and is stored on the sub-index 03h of communication parameters (1800h to 1803h) to prevent the CAN from being continuously occupied by PDO with lower priorities. After the parameter (unit: 100 us) is set, the transmission interval of one TPDO must be longer than or equal to the time corresponding to this parameter.

  Example: If the inhibit time of TPDO2 is 300 ms, the transmission interval of TPDOs is no shorter than 30 ms.

- **Event timer**

For TPDO transmitted in asynchronous mode (transmission types 254 or 255), the event timer is defined in sub-index 05 of communication parameters (1800h–1803h). The event timer can be considered as a trigger event. It also triggers corresponding TPDO transmission. If another event, for example, data change, occurs in the operation cycle of the event timer, the TPDO is triggered and the event timer is reset immediately.

PDO mapping parameter

PDO mapping parameters include pointers of process data that corresponds to PDO and that is to be sent or received by PDO, including index, sub-index, and mapping object length. The length of each PDO data can be up to eight bytes and one or multiple objects can be mapped. The sub-index 00 records the number of objects mapped by the PDO and the sub-indexes 01...08 are the mapping content.

The following takes 1600h as an example.

Table 3–12 Description of PDO mapping relation

Index	Sub-index	Description
1600h	00	Number of mapped objects
	01	Content of mapping parameter
	…	
	08	

Table 3–13 Definition of PDO mapping parameters

Places	31	…	16	15	…	8	7	…	0
Description	Index			Sub-index			Object Length		

The index and sub-index together define the position of an object in the object dictionary. The object length indicates the bit length of the object in hexadecimal, as shown below.

Table 3–14 Relation between object length and object bit length

Object Length	Bit Length
08h	8-bit
10h	16-bit
20h	32-bit

For example: the mapping parameter of the 16-bit command word 6040.00h is 60400010h.

The following example describes the PDO mapping relation.

Example:

RPDO1 maps the following three parameters.

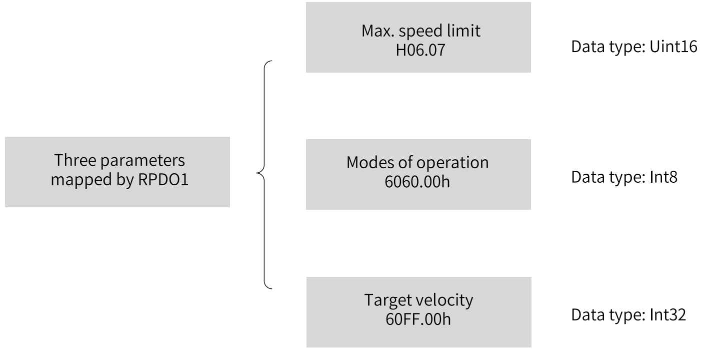
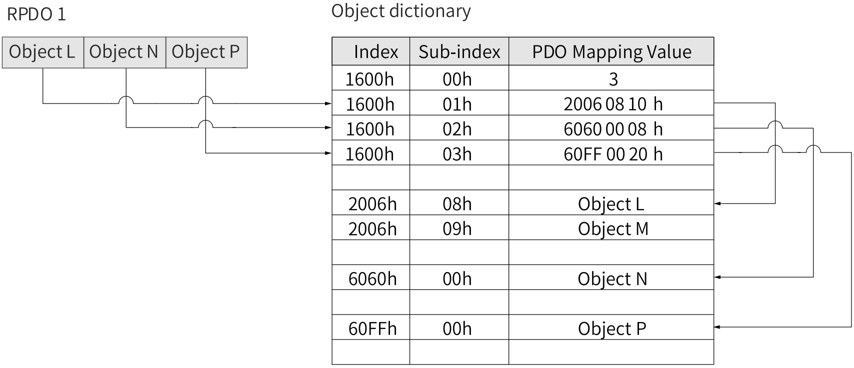


Figure 3-11 Example of PDO1 mapping

Then, the mapping length is seven bytes (2+1+4), namely there are seven bytes in the data segment of RPDO1 during transmission. The mapping relation is shown in the following figure.



Index	Sub-index	PDO Mapping Value
1600h	00h	3
1600h	01h	2006 08 10 h
1600h	02h	6060 00 08 h
1600h	03h	60FF 00 20 h
2006h	08h	Object L
2006h	09h	Object M
6060h	00h	Object N
60FFh	00h	Object P

Figure 3-12 Example of RPDO mapping relation

The mapping mode of TPDO is the same as that of RPDO, but in the opposite direction. The RPDO decodes the input based on the mapping relation. The TPDO encodes the output based on the mapping relation.

Example:

TPDO2 maps the following two parameters.

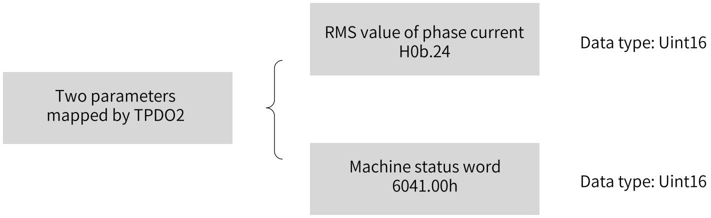
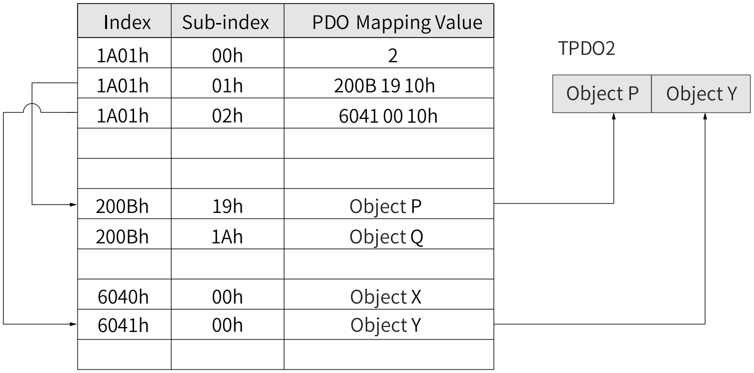

Two parameters mapped by TPDO2
RMS value of phase current H0b.24
Data type: Uint16
Machine status word 6041.00h
Data type: Uint16

Figure 3-13 Example of TPDO2 mapping relation

Then, the mapping length is four bytes (2+2), namely there are four bytes in the data segment of TPDO2 during transmission. The mapping relation is shown in the following figure.



Index	Sub-index	PDO Mapping Value
1A01h	00h	2
1A01h	01h	200B 19 10h
1A01h	02h	6041 00 10h
200Bh	19h	Object P
200Bh	1Ah	Object Q
6040h	00h	Object X
6041h	00h	Object Y

Figure 3-14 Example of TPDO mapping relation

3.4.4 Synchronization Object (SYNC)

The SYNC object is a special mechanism that controls harmony and synchronization between transmission and reception of multiple nodes. It is used for synchronous transmission of the PDO.

The following figure shows the configuration flowchart of the Sync generator.

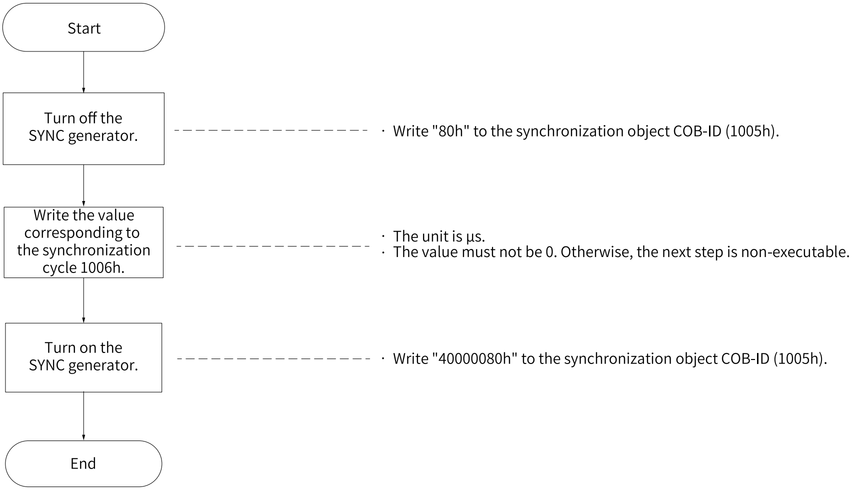

Start
Turn off the SYNC generator.
· Write "80h" to the synchronization object COB-ID (1005h).
Write the value corresponding to the synchronization cycle 1006h.
· The unit is μs.
· The value must not be 0. Otherwise, the next step is non-executable.
Turn on the SYNC generator.
· Write "40000080h" to the synchronization object COB-ID (1005h).
End

Figure 3-15 Synchronization generator configuration flowchart

Note

The SV680P-INT series does not support the Sync generator with cycle lower than 500 us. Synchronization cycles lower than 1 ms are not recommended.

Sync generator

The SV680P-INT servo drive is both a synchronization consumer and a synchronization producer. The objects related to synchronization are synchronization object COB-ID (1005h) and synchronization cycle (1006h).

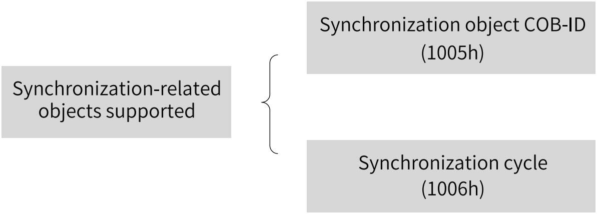

Synchronization-related objects supported
Synchronization object COB-ID (1005h)
Synchronization cycle (1006h)

Figure 3-16 Description of supported objects related to synchronization

The second most significant bit of the synchronization object COB-ID determines whether to activate the Sync generator.

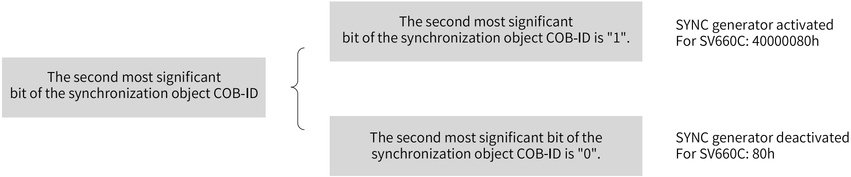

The second most significant bit of the synchronization object COB-ID
The second most significant bit of the synchronization object COB-ID is "1".
SYNC generator activated
For SV660C: 40000080h
The second most significant bit of the synchronization object COB-ID is "0".
SYNC generator deactivated
For SV660C: 80h

Figure 3-17 Activating the synchronization generator

The synchronization cycle (unit: us) is only used for the Sync generator. It indicates the interval of the node in generating the synchronization object.

Synchronization object transmission framework

Synchronization objects are transmitted based on the producer-consumer model, which is similar to PDO transmission. The synchronization producer sends a synchronous frame, and other nodes in the CAN network can receive this frame as consumers, without the need to provide any feedback. Only one Sync generator is allowed to be activated in one CAN network. The following figure shows the transmission framework of synchronization objects.

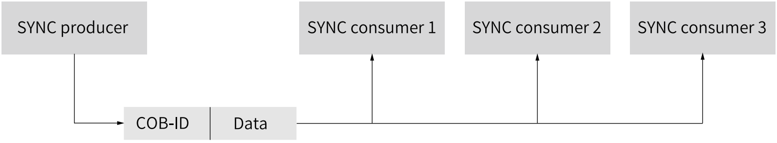


Figure 3-18 Synchronization transmission framework

Transmission of synchronous PDO is closely related to the synchronous frame.

- For an RPDO, so long as the PDO is received, the received PDO is updated to the application in the next synchronization.
- A synchronization TPDO can be transmitted in cyclic synchronization mode or acyclic synchronization mode.

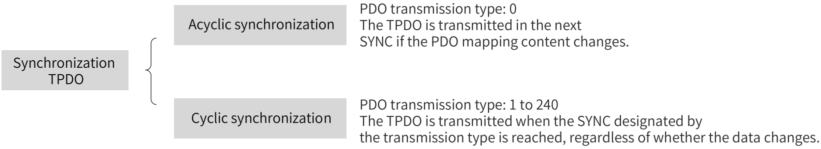

Synchronization TPDO
Acyclic synchronization
PDO transmission type: 0
The TPDO is transmitted in the next SYNC if the PDO mapping content changes.
Cyclic synchronization
PDO transmission type: 1 to 240
The TPDO is transmitted when the SYNC designated by the transmission type is reached, regardless of whether the data changes.

Figure 3-19 Description of synchronization TPDO

The following figure shows the synchronous transmission model.

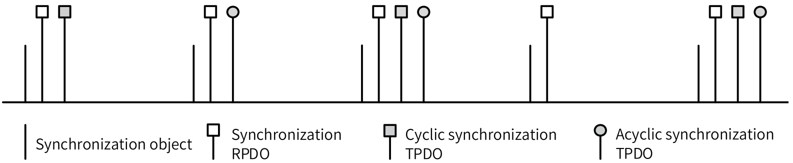


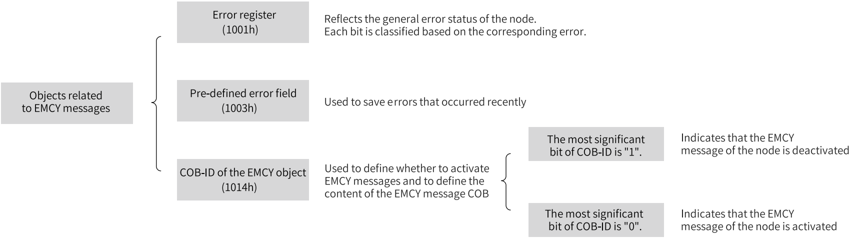
Figure 3-20 Synchronous transmission model

Example:

RPDO1 has a transmission type of 0, RPDO2 has a transmission type of 5, TPDO1 has a transmission type of 0, and TPDO2 has a transmission type of 20. Once RPDO1 and RPDO2 receive the PDO, the latest PDO data will be updated to the corresponding application in the next synchronization. Once the mapping data of TPDO1 changes, TPDO1 will be transmitted in the next synchronization. After TPDO2 experiences 20 SYNC, the PDO will be transmitted no matter whether the data changes.

3.4.5 Emergency (EMCY) Object Service

When an error occurs in a CANopen node, the node sends an emergency message according to the standard mechanism. The emergency message complies with the producer-consumer model. After the node fault is sent, other nodes in the CAN network may handle the fault. The SV680P-INT series servo drive only serves as the emergency message producer, which means it does not process emergency messages of other nodes.



Objects related to EMCY messages

Error register (1001h)

Reflects the general error status of the node.
Each bit is classified based on the corresponding error.

Pre-defined error field (1003h)

Used to save errors that occurred recently

COB-ID of the EMCY object (1014h)

Used to define whether to activate EMCY messages and to define the content of the EMCY message COB

The most significant bit of COB-ID is "1".

Indicates that the EMCY message of the node is deactivated

The most significant bit of COB-ID is "0".

Indicates that the EMCY message of the node is activated

Figure 3-21 Description of objects related to emergency messages

When a fault occurs on the node, the error register and the pre-defined error field must be updated no matter whether the emergency object is activated. The content of the emergency message follows the following specifications.

Table 3–15 Specifications of the content of an emergency message

COB-ID	0	1	2	3	4	5	6	7
80h + Node_ID	Error code		Error register	Re served	Auxiliary byte			

- The error register is always consistent with 1001h.
- When a communication error occurs, the error code is consistent with the one required by DS301 and the auxiliary byte is 0.
- When the error described in the DSP402 sub-protocol occurs on the servo drive, the error code is consistent with DS402 requirements and corresponds to the object 603Fh. The auxiliary byte is extra descriptions.
- When an error specified by the user occurs on the servo drive, the error code is 0xFF00 and the auxiliary byte displays the error code specified by the user.

3.4.6 SDO Transmission Message

SDO transmission include transmission of object data with no more than four bytes and those with more than four bytes. Object data with no more than four bytes are transmitted in the expedited SDO mode. Object data with more than four bytes are transmitted in the segmented SDO mode or block mode.

The SV680P-INT supports expedited SDO transfer and segmented SDO transfer only.

An SDO transmission message consists of a COB-ID and a data segment. As shown in the following table, the COB-ID of T_SDO and R_SDO messages are different.

The data segment adopts the little endian mode, in which least significant bits are arranged in front of most significant bits. The data segment of all SDO messages must consist of eight bytes. The following table describes the format of SDO transmission message.

Table 3–16 Description of SDO transmission message format

COB-ID	Data (data segment)							
580h+Node_ID	0	1	2	3	4	5	6	7
600h+Node_ID	Command code	Index		Sub-index	Data			

The command code specifies the transmission type and transmission data length of the SDO. The index and sub-index indicate the position of the SDO in the list; the data indicates the value of the SDO.

Message written in expedited SDO mode

Expedited SDO transfer is used for reading/writing the object message with no more than four bytes. The transmission message varies the read/write mode and data length. The following table describes the message written in the expedited SDO mode.

Table 3–17 Description

		COB-ID	0	1	2	3	4	5	6	7
Client→		600h+Node_ID	23h	Index		Sub-index	Data			
			27h				Data			-
			2bh				Data		-	-
			2fh				Data	-	-	-
←Server	Normal	580h+Node_ID	60h	Index		Sub-index	-	-	-	-
	Abnormal		80h				Abort Code			

Note

"-" indicates that data exists but is not considered. It is recommended that value 0 be written for the data. The same rule applies to the following descriptions in this section.

Example:

If the slave station No. is 4 and SDO is used to write the speed value (60FF.00h) in the speed mode, write 1000 (namely 0x3E8). The message sent by the master is shown in the following table. (All data are in hexadecimal format.)

Table 3–18 Example of a message sent by the master

COB-ID	0	1	2	3	4	5	6	7
604	23	FF	60	00	E8	03	00	00

If the value is written successfully, the servo drive returns the following message.

Table 3–19 Example of a message returned by the servo drive upon normal write operation

COB-ID	0	1	2	3	4	5	6	7
584	60	FF	60	00	00	00	00	00

If the type of the data written does not match, the fault code 0x06070010 is returned. The message is as follows.

Table 3–20 Example of a message returned upon mismatch of the written data type

COB-ID	0	1	2	3	4	5	6	7
584	80	FF	60	00	10	00	07	06

Message written in expedited SDO mode

Object message with no more than four bytes are read in the expedited SDO mode. The following table describes the message written in the expedited SDO mode.

Table 3–21 Structure of an SDO start packet transmitted

		COB-ID	0	1	2	3	4	5	6	7
Client→		600h+Node_ID	40h	Index		Sub-index	-	-	-	-
←Server	Normal	580h+Node_ID	41h	Index		Sub-index	Data Length			
	Abnormal		80h				Abort Code			

During transmission, the trigger bit (bit6) of the command code sends 0 or 1 alternatively. This rule must be maintained so that the slave can respond to the message. The structure of the process message is shown in the following table.

Table 3–22 Structure of a message during SDO transmission

		COB-ID	0	1	2	3	4	5	6	7
Client→		600h+Node_ID	60h	-	-	-	-	-	-	-
←Server	Normal	580h+Node_ID	00h	Data Length						
	Abnormal		80h	Index		Sub-index	Abort Code			
Client→		600h+Node_ID	70h	-	-	-	-	-	-	-
←Server	Normal	580h+Node_ID	10h	Data Length						
	Abnormal		80h	Index		Sub-index	Abort Code			

The response packet of the end frame transmitted in segmented mode includes the end frame identifier and valid data length of the end frame. The structure of its transmission message is shown in the following table.

Table 3–23 Message structure of the last frame in SDO segmented transmission

		COB-ID	0	1	2	3	4	5	6	7
Client→		600h+Node_ID	60h/70h	Index		Sub-index	-	-	-	-
←Server	Normal	580h+Node_ID	01h/11h	Data						
			03h/13h	Data						-
			05h/15h	Data					-	-
			07h/17h	Data				-	-	-
			09h/19h	Data			-	-	-	-
			0Bh/1Bh	Data		-	-	-	-	-
			0Dh/1Dh	Data	-	-	-	-	-	-
	Abnormal		80h	Index		Sub-index	Abort Code			

3.4.7 SDO transmission framework

SDO transmission complies with the client-server mode, that is, one initiates a request and the other responds to the request. The SDO client in the CAN bus network initiates a request and the SDO server responds to the request. Therefore, data exchange between SDOs requires at least two CAN messages with different CAN identifiers. The SDO transmission model is shown in the following figure.

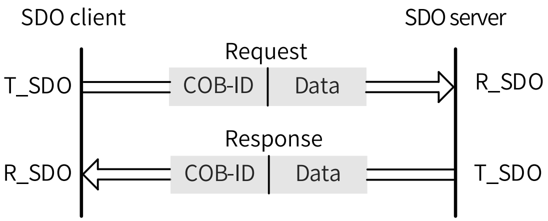


Figure 3-22 Object word in the SDO server read/written by the SDO client

3.5 Communication Parameters

To connect the servo drive to the CANopen fieldbus network, set related parameters of the servo drive properly.

CANopen parameters:

Parameter	Communication Address	Name	Value	Default	Unit	Change Mode
H02.00	2002-01h	Control mode	0: Speed control mode 1: Position control mode 2: Torque control mode 3: Torque/Speed control mode 4: Speed/Position control mode 5: Torque/Position control mode 6: Torque/Speed/Position compound mode 7: Process segment 8: CANopen mode	1	-	At stop
H0E.00	200E-01h	Node address	1 to 127	1	-	At stop
H0E.01	200E-02h	Save objects written through communication to e2prom	0: Not save 1: Save parameters written through communication to e2prom 2: Save object dictionaries written through communication to e2prom 3: Save parameters and object dictionaries written through communication to e2prom 4: Save object dictionaries written before communication (OP) to e2prom	1	-	Real-time

Parameter	Communication Address	Name	Value	Default	Unit	Change Mode
H0E.10	200E-0Bh	CAN selection	0: Pulse/Axis control command 1: Enhanced axis control command 2: CANopen	0	-	At stop
H0E.11	200E-0Ch	CAN baud rate	0: 20kbps 1: 50kbps 2: 100kbps 3: 125kbps 4: 250kbps 5: 500kbps 6: 1Mbps 7: 1Mbps	5	-	At stop

3.6 PN-to-CANopen bridge

H0E.11 sets the baud rate and H0E.10 sets the CAN station number. The SV680P-INT supports 4 RPDOs/TPDOs and 8-bit/16-bit/32-bit data structures. Related parameter:

		2D address				2E address	
OUT	RPDO1	Number of Mapping Objects	2D-20	INPUT	TPDO1	Number of Mapping Objects	2E-14
		Mapped object 1 in RPDO1	2D-21			Mapped object 1 in TPDO1	2E-15
		Mapped object 2 in RPDO1	2D-23			Mapped object 2 in TPDO1	2E-17
		Mapped object 3 in RPDO1	2D-25			Mapped object 3 in TPDO1	2E-19
		Mapped object 4 in RPDO1	2D-27			Mapped object 4 in TPDO1	2E-1B
		Mapped object 5 in RPDO1	2D-29			Mapped object 5 in TPDO1	2E-1D
		Mapped object 6 in RPDO1	2D-2B			Mapped object 6 in TPDO1	2E-1F
		Mapped object 7 in RPDO1	2D-2D			Mapped object 7 in TPDO1	2E-21
		Mapped object 8 in RPDO1	2D-2F			Mapped object 8 in TPDO1	2E-23
	RPDO2	Number of Mapping Objects	2D-31		TPDO2	Number of Mapping Objects	2E-25
		Mapped object 1 in RPDO2	2D-32			Mapped object 1 in TPDO2	2E-26
		Mapped object 2 in RPDO2	2D-34			Mapped object 2 in TPDO2	2E-28
		Mapped object 3 in RPDO2	2D-36			Mapped object 3 in TPDO2	2E-2A
		Mapped object 4 in RPDO2	2D-38			Mapped object 4 in TPDO2	2E-2C
		Mapped object 5 in RPDO2	2D-3A			Mapped object 5 in TPDO2	2E-2E
		Mapped object 6 in RPDO2	2D-3C			Mapped object 6 in TPDO2	2E-30
		Mapped object 7 in RPDO2	2D-3E			Mapped object 7 in TPDO2	2E-32
		Mapped object 8 in RPDO2	2D-40			Mapped object 8 in TPDO2	2E-34

		2D address				2E address	
OUT	RPDO3	Number of Mapping Objects	2D-42	INPUT	TPDO3	Number of Mapping Objects	2E-36
		Mapped object 1 in RPDO3	2D-43			Mapped object 1 in TPDO3	2E-37
		Mapped object 2 in RPDO3	2D-45			Mapped object 2 in TPDO3	2E-39
		Mapped object 3 in RPDO3	2D-47			Mapped object 3 in TPDO3	2E-3B
		Mapped object 4 in RPDO3	2D-49			Mapped object 4 in TPDO3	2E-3D
		Mapped object 5 in RPDO3	2D-4B			Mapped object 5 in TPDO3	2E-3F
		Mapped object 6 in RPDO3	2D-4D			Mapped object 6 in TPDO3	2E-41
		Mapped object 7 in RPDO3	2D-4F			Mapped object 7 in TPDO3	2E-43
		Mapped object 8 in RPDO3	2D-51			Mapped object 8 in TPDO3	2E-45
	RPDO4	Number of Mapping Objects	2D-53		TPDO4	Number of Mapping Objects	2E-47
		Mapped object 1 in RPDO4	2D-54			Mapped object 1 in TPDO4	2E-48
		Mapped object 2 in RPDO4	2D-56			Mapped object 2 in TPDO4	2E-4A
		Mapped object 3 in RPDO4	2D-58			Mapped object 3 in TPDO4	2E-4C
		Mapped object 4 in RPDO4	2D-5A			Mapped object 4 in TPDO4	2E-4E
		Mapped object 5 in RPDO4	2D-5C			Mapped object 5 in TPDO4	2E-50
		Mapped object 6 in RPDO4	2D-5E			Mapped object 6 in TPDO4	2E-52
		Mapped object 7 in RPDO4	2D-60			Mapped object 7 in TPDO4	2E-54
		Mapped object 8 in RPDO4	2D-62			Mapped object 8 in TPDO4	2E-56

Note

- Ensure that the number of bytes in each PDO is no more than 8 bytes.
- For a PDO that does not involve communication, you must clear the value of the parameter so that the device can run normally.
- The number of mappings must match the actual number.

4 EtherCAT Communication [N]

4.1 Communication

4.1.1 Communication technical data

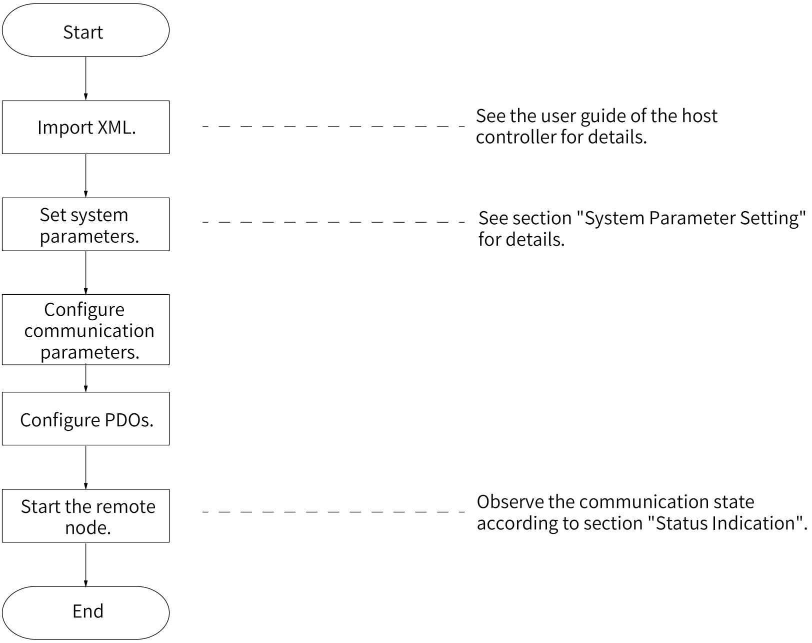
Item		Specification
EtherCAT Basic performance of slave	Communication protocol	EtherCAT protocol
	Service supported	CoE (PDO, SDO)
	Synchronization mode	DC - Distributed clock FreeRun
	Physical layer	100BASE-TX
	Baud rate	100 Mbit/s (100Base-TX)
	Duplex mode	Full duplex
	Topology	Ring and linear
	Transmission medium	Shielded cables of Cat 5e or higher
	Transmission distance	Less than 100 m between two nodes (with proper environment and cables)
	Number of slaves	Up to 65535 by protocol, not exceeding 100 in actual use
	EtherCAT frame length	44 bytes to 1498 bytes
	Process data	Max. 1,486 bytes per Ethernet frame
	Synchronous jitter of two slaves	< 1 us
	Update time	About 30 us for 1000 DI/DOs About 100 µs for 100 servo axes Define different update time for different interfaces.
	Bit error rate	10^{-10} Ethernet standard
EtherCAT Configuration unit	Number of FMMU units	8
	Number of storage synchronization management units	8
	Process data RAM	8 kB
	Distributed Clock	64-bit
	EEPROM capacity	32 kbits

4.1.2 Communication Specifications

Item		Specification
Communication protocol		IEC 61158 Type 12, IEC 61800-7 CiA 402 Drive Profile
Application layer	SDO	SDO request, SDO response
	PDO	Variable PDO mapping
	CiA402	Profile position mode (PP)
		Profile velocity mode (PV)
		Profile torque mode (PT)
		Homing mode (HM)
		Cyclic synchronous position mode (CSP)
		Cyclic synchronous velocity mode (CSV)
		Cyclic synchronous torque mode (CST)
Physical layer	Transmis sion protocol	100BASE-TX (IEEE802.3)
	Maximum distance	100 m
	Interface	RJ45 × 2 (IN, OUT)

4.1.3 Protocols

EtherCAT is an industrial Ethernet-based fieldbus system that features high performance, low cost, easy use and flexible topology. It is applicable to industrial applications requiring ultra-high speed I/O network. EtherCAT adopts standard Ethernet physical layer with twisted pairs or optical fibers (100Base-TX or 100Base-FX) used as the transmission media.


Start
Import XML.
See the user guide of the host controller for details.
Set system parameters.
See section "System Parameter Setting" for details.
Configure communication parameters.
Configure PDOs.
Start the remote node.
Observe the communication state according to section "Status Indication".
End

An EtherCAT system includes the master and the slave. The master requires a common network adapter, and the slave requires a special slave control chip, such as ET1100, ET1200, and FPGA.

EtherCAT can process data at the I/O layer,

- without any sub-bus
- or gateway delay
- One system covers all devices, including input/output devices, sensors, actuators, drives, and displays……
- Transmission rate: 2 × 100 Mbit/s (high-speed Ethernet, full duplex mode).
- Synchronization: synchronization jitter < 1 μs (number of nodes up to 300, cable length within 120 m)

Update time:

256 DI/DOs: 11 μs

1000 DI/DOs distributed in 100 nodes: 30 μs = 0.03 ms

200 AI/AOs (16-bit): 50 μs, sampling rate: 20 kHz

100 servo axes (8 bytes IN + 8 bytes OUT for each): 100 μs = 0.1 ms

12000 digital I/Os: 350 μs

To support more types of devices and applications, EtherCAT establishes the following application protocols:

- CANopen over EtherCAT (CoE)
- Safety over EtherCAT (SoE, compliant with IEC 61800-7-204)
- Ethernet over EtherCAT (EoE)
- File over EtherCAT (FoE)

The slave only needs to support the suitable application protocol.

Note

EtherCAT^{R} is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

4.2 Hardware Configuration

4.2.1 Terminal Layout

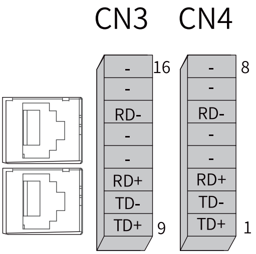


Table 4–1 EtherCAT communication terminal pins

Pin No.	Name	Description
1	TD+	Data transmit positive
2	TD-	Data transmit negative
3	RD+	Data reception+
4 and 5	-	-
6	RD-	Data reception–
7 and 8	-	-
9	TD+	Data transmit positive
10	TD-	Data transmit negative
11	RD+	Data reception+
12 and 13	-	-
14	RD-	Data reception–
15 and 16	-	-

4.2.2 EtherCAT Communication Connection Example

CN3 and CN4 are EtherCAT connectors. Connect CN4 (IN) to the communication port of the master and CN3 (OUT) to the next slave. For assignment of CN3/CN4 terminal pins, see *"Table 4–1 EtherCAT communication terminal pins" on page 59*.

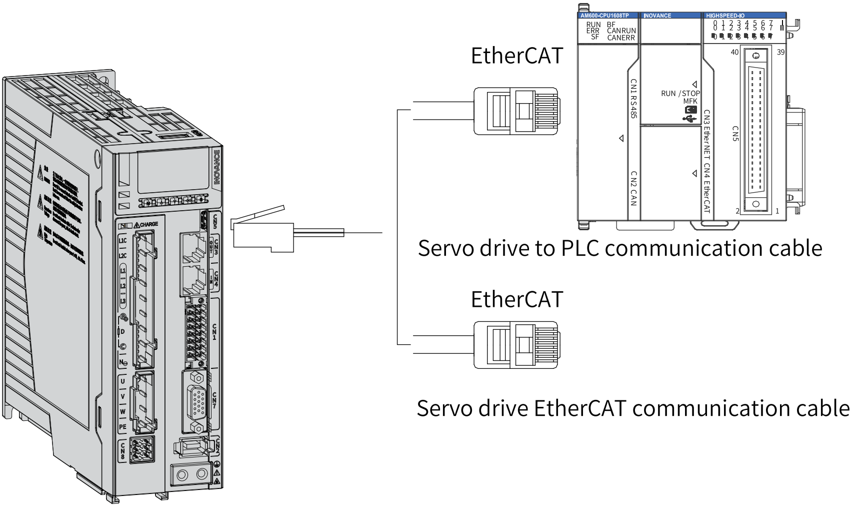


Figure 4-1 Wiring of communication cables

Topology

The communication topology of EtherCAT is flexible without any limit, as shown in *"Figure 4–2 Communication network topology" on page 61*. The drive carries IN and OUT ports.

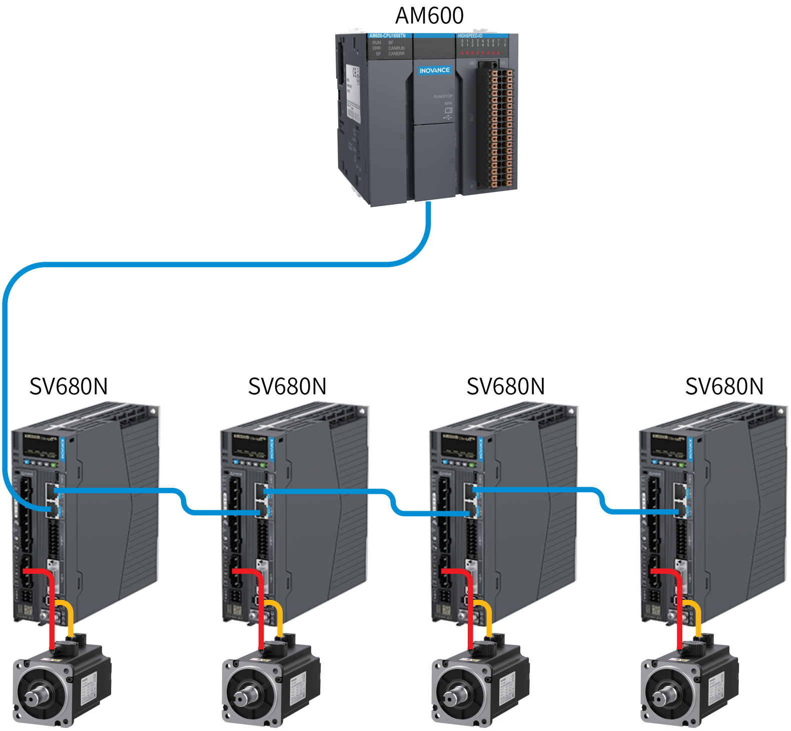
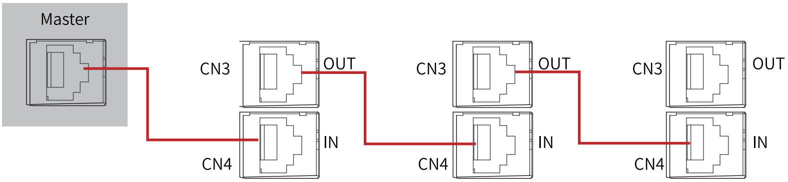
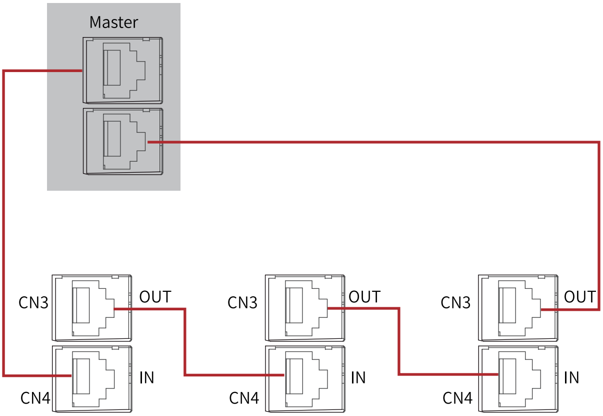


Figure 4-2 Communication network topology

Linear topology



Redundant ring topology



Note

When using the redundant ring, set H0E.36 (EtherCAT AL enhanced link) to 1 (Enable), then power on the drive again.

4.3 Communication Transmission Mode

4.3.1 Structure of EtherCAT Communication

Multiple kinds of application protocols are available for EtherCAT communication. The IEC 61800-7 (CiA 402)-CANopen motion control profile is used for SV680N-INT series servo drives. The following figure shows the EtherCAT communication structure at the CANopen application layer.

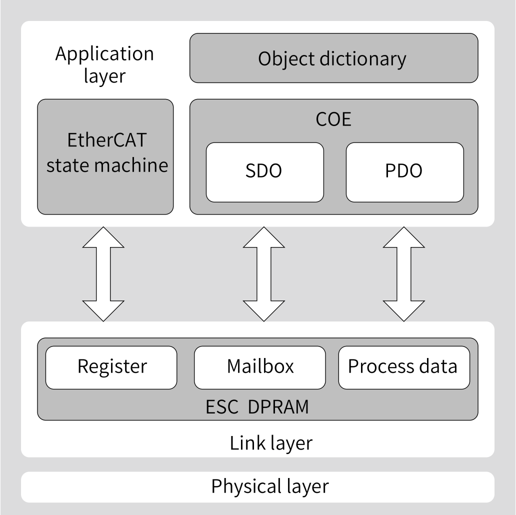


Figure 4-3 EtherCAT communication structure at CANopen application layer

The object dictionary in the application layer includes communication parameters, application process data and PDO mapping data. The process data object (PDO) includes the real-time data generated during operation, which is read and written cyclically. In the SDO mailbox communication, the communication objects and PDO objects are being accessed and modified non-cyclically.

4.3.2 Communication State Machine

The following figure shows the status transition diagram of EtherCAT state machine.

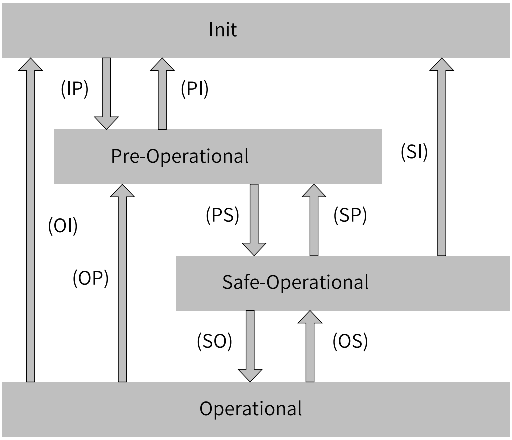


Figure 4-4 EtherCAT state machine

The EtherCAT state machine must support the following four states and coordinate the states between the master and slave application program during initialization and operation.

- Init: initialization, shortened as I
- Pre-Operational: pre-operational, shortened as P
- Safe-Operational: safe-operational, shortened as S
- Operational: operational, shortened as O

Transition from Init state to Operational state must be in the sequence of Init→Pre-Operational→Safe→Operational, and then Operational step by step. In transition from the Operational state to the Init state, certain steps can be skipped. The following table lists the state transition and the initialization process.

Status	SDO	RPDO	TPDO	Description
Init (I)	No	No	No	Communication initialization No communication available in the application layer, EtherCAT slave controller (ESC) register can only be read/written by the master
IP	No	No	No	The master configures the slave addresses, mailboxes, and distributed clocks (DCs). Request the Pre-Operational state.
Pre-Operational (P)	Yes	No	No	Mailbox data communication in the application layer (SDO).
PS	Yes	No	No	The master uses process data mapping of SDO initialization. The master configures the Sync Manager channel used during process data communication. The master configures the FMMU. Request the Safe-Operational state.
Safe-Operational (S)	Yes	No	Yes	SDO, TPDO, and distributed clock mode can be used.
SO	Yes	No	Yes	The master sends valid output data. to request the Operational state.
Operational (O)	Yes	Yes	Yes	Normal operational state Both input and output are valid. Mailbox communication can still be used.

4.3.3 Distributed clock

The distributed clock (DC) enables all EtherCAT devices to use the same system time and allows synchronous execution of slave tasks. A slave produces the synchronization signal according to the synchronized system time. The SV680N-INT drive only supports the DC sync mode. The synchronization period, which is controlled by SYNC0, varies with different motion modes.

Note

- The SYNC signal can be used to synchronize all the salves with an error less than 1 us. The master must synchronize all the slaves to the same clock and continues doing so during operation to prevent clock skew caused by difference in the crystal oscillator. This is usually done by synchronizing the 0x910 register in ESC.
- SYNC starting time = 0x990 register (with ESC) - 0x920
  Note that the DC mode (0x981 = 0x03) can be enabled only before 0x910 reaches the starting time. If the starting time of SYNC is set improperly, the 0x134 status register of ESC will report the error code of 0x2D.

4.3.4 Status Indication

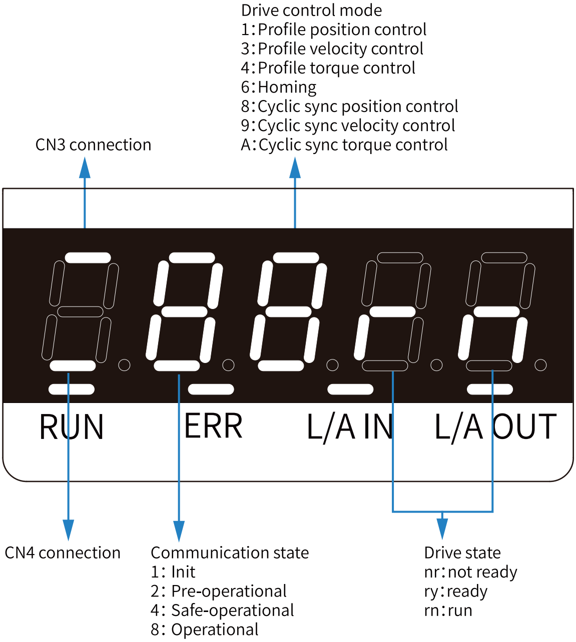


Figure 4-5 Status indication diagram

If the value 0 is displayed, it indicates no value is written or the value 0 is written to 6060h, or H02.00 is set to 0, 1 or 2.

Communication connection status

For the SV680N-INT, the connection status of the two RJ45 ports are indicated by "-" on the upper and lower part of the first LED on the keypad. The upper "-" indicates the status of CN3:PORT1, and the lower "-" indicates the status of CN4:PORT0.

OFF: no communication connection is detected in the physical layer.

ON: communication connection is detected in the physical layer.

Communication status

The 2nd LED indicates the status of the EtherCAT state machine of the slave in the form of characters, as described in the following table.

State of EtherCAT state machine

Status	SDO	RPDO	TPDO	Description	Panel Display
Initialization	No	No	No	Communication initialization	1: Solid ON
Pre-operational	Yes	No	No	Network configuration initialized SDO is available	2: Blinks at an interval of 400 ms
Safe-operational	Yes	No	Yes	SDO, TPDO, and distributed clock mode are available	4: Blinks with a period of 1200 ms, on for 200 ms and off for 1000 ms
Operation	Yes	Yes	Yes	Normal operation state	8: Solid ON

Display of control modes

The 3rd LED indicates the operation mode of the servo drive in the form of hexadecimal without blinking, as described in the following table.

The operation modes include the following:

Modes of operation (6060h)	Panel Display
1: Profile position mode	1
3: Profile velocity mode	3
4: Profile torque mode	4
6: Homing mode	6
8: Cyclic synchronous position mode	8
9: Cyclic synchronous velocity mode	9
10: Cyclic synchronous torque mode	A

Display of servo status

The 4th and 5th LEDs indicate the servo status of the slave.

The statuses include the following:

Status	Description	Panel Display
Reset	Initialization	reset
Not ready	Initialization is done. The control circuit is switched on but the main circuit is not switched on. Not ready	nr
Ready	The main circuit is switched on, but the S-ON signal is inactive. Ready	ry The character "y" blinks when the motor speed is not 0 RPM. When the communication layer is in the pre-operational or safe-operational state, the blinking frequency is the same as that of characters "2" or "4" (see "communication status" in the previous page for details). When the communication layer is in Init or Operational state, the blinking frequency is 2 Hz.
Operation	The S-ON signal is active and the motor is energized. Run	rn The letter "n" blinks when the motor speed is not 0 RPM. When the communication layer is in the pre-operational or safe-operational state, the blinking frequency is the same as that of characters "2" or "4" (see "communication status" in the previous page for details). When the communication layer is in Init or Operational state, the blinking frequency is 2 Hz.

Description of indicators

OFF

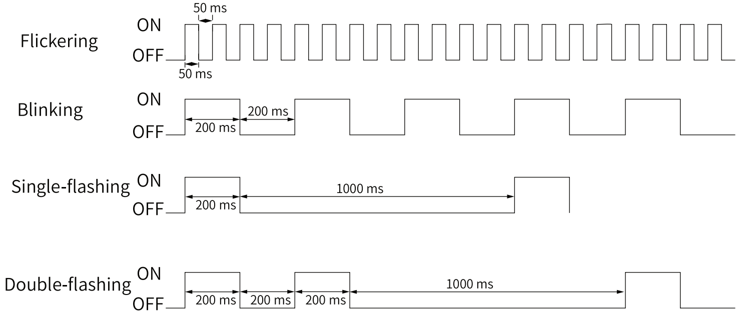


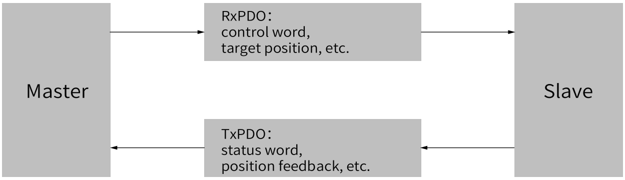
Figure 4-6 Description of indicators

Indicator	Status	Status Indication
RUN	OFF	Initialization.
	Blinking (on for 200 ms and off for another 200 ms)	Pre-Operational.
	Single flash (on for 200 ms and off for 1000 ms)	Safe-Operational.
	ON	Operational.
ERR	OFF	No Network error.
	Blinking (on for 200 ms and off for another 200 ms)	Communication setting error.
	Single flash (on for 200 ms and off for 1000 ms)	Sync event error.
	Double flash (on for 200 ms and off for 200 ms, and then on for 200 ms and off for 1000 ms)	Watchdog timeout.
L/A IN indicator[1] L/A OUT indicator	OFF	Link is not established.
	Flickering (on for 50 ms and off for another 50 ms)	Link is established. A data transceiving signal is present.
	ON	Link is established. No data transceiving signal is present.

4.4 Data Frame Structure

4.4.1 Process data

The real-time data transmission of EtherCAT is achieved through PDO. PDOs can be divided into RPDOs (Receive PDO) and TPDOs (Transmit PDO) based on the data transmission direction. RPDOs transmit the master data to the slave, and TPDOs returns the slave data to the master.



The SV680N-INT series servo drive allows users to assign the PDO list and define the PDO mapping objects.

PDO mapping

PDO mapping is used to establish the mapping relation between the object dictionary and the PDO. 1600h to 17FFh are RPDOs, and 1A00h to 1BFFh are TPDOs. The SV680N-INT provides 7 RPDOs and 6 TPDOs, as listed in the following table.

RPDO (7)	1600h, 1601h	Variable mapping
	1701h to 1705h	Fixed mapping
TPDO (6)	1A00h, 1A01h	Variable mapping
	1B01h to 0x1B04h	Fixed mapping

Fixed PDO mapping

SV680N-INT provides five fixed RPDOs and four fixed TPDOs.

The following table lists the typical instances of RPDOs and TPDOs.

Control Mode	PP/CSP
1701h (Outputs)	Mapping objects (4 mapping objects, 12 bytes)
	6040h (Control word) 607Ah (Target position) 60B8h (Touch probe function) 60FEh sub-index 1 (forced physical outputs)
1B01h (Inputs)	Mapping objects (9 mapping objects, 28 bytes)
	603Fh (error code) 6041h (status word) 6064h (position actual value) 6077h (torque actual value) 60F4h (following error actual value) 60B9h (touch probe status) 60BAh (probe 1 positive edge) 60BCh (probe 2 positive edge) 60FDh (Digital inputs)

Control Mode	PP/PV/PT/CSP/CSV/CST
1702h (Outputs)	Mapping objects (7 mapping objects, 19 bytes)
	6040h (Control word) 607Ah (Target position) 60FFh (Target velocity) 6071h (Target torque) 6060h (Modes of operation) 60B8h (Touch probe function) 607Fh (Max. profile velocity)

1B02h (Inputs)	Mapping objects (9 mapping objects, 25 bytes)
	603Fh (error code) 6041h (status word) 6064h (position actual value) 6077h (torque actual value) 6061h (modes of operation display) 60B9h (touch probe status) 60BAh (probe 1 positive edge) 60BCh (probe 2 positive edge) 60FDh (Digital inputs)

Control Mode	PP/PV/CSP/CSV
1703h (Outputs)	Mapping objects (7 mapping objects, 17 bytes)
	6040h (Control word) 607Ah (Target position) 60FFh (Target velocity) 6060h (Modes of operation) 60B8h (Touch probe function) 60E0h (Positive torque limit value) 60E1h (Negative torque limit value)
1B03h (Inputs)	Mapping objects (10 mapping objects, 29 bytes)
	603Fh (error code) 6041h (status word) 6064h (position actual value) 6077h (torque actual value) 60F4 (following error actual value) 6061h (modes of operation display) 60B9h (touch probe status) 60BAh (probe 1 positive edge) 60BCh (probe 2 positive edge) 60FDh (Digital inputs)

Control Mode	PP/PV/PT/CSP/CSV/CST
1704h (Outputs)	Mapping objects (9 mapping objects, 23 bytes)
	6040h (Control word) 607Ah (Target position) 60FFh (Target velocity) 6071h (Target torque) 6060h (Modes of operation) 60B8h (Touch probe function) 607Fh (Max. profile velocity) 60E0h (Positive torque limit value) 60E1h (Negative torque limit value)

Control Mode	PP/PV/CSP/CSV
1705h (Outputs)	Mapping objects (8 mapping objects, 19 bytes)
	6040h (Control word) 607Ah (Target position) 60FFh (Target velocity) 6060h (Modes of operation) 60B8h (Touch probe function) 60E0h (Positive torque limit value) 60E1h (Negative torque limit value) 60B2h (Torque offset)

Variable PDO mapping

SV680N-INT provides two variable RPDOs and two variable TPDOs.

Variable PDO	Index	Max. Length of the Byte	Default Mapping Object
RPDO1	1600h 1601h	40	6040h (Control word) 607Ah (Target position) 60B8h (Touch probe function)
TPDO1	1A00h 1A01h	40	603Fh (error code) 6041h (status word) 6064h (position actual value) 60BCh (probe 2 positive edge) 60B9h (touch probe status) 60BAh (probe 1 positive edge) 60FDh (Digital inputs)

Sync Manager PDO assignment

The process data can contain multiple PDO mapping data objects during cyclic EtherCAT data communication. The CoE protocol defines the PDO mapping object list of the Sync Manager using data objects 1C10 to 1C2Fh. Multiple PDOs can be mapped to different sub-indexes. The SV680N-INT series servo drive supports assignment of one RPDO and one TPDO, as described in the following table.

Index	Sub-index	Description
1C12h	01h	One of 1600h, 1601h and 1701h to 1705h used as the actual RPDO.
1C13h	01h	One of 1A00h, 1A01h and 1B01h to 1B04h used as the actual TPDO.

PDO configuration

PDO mapping parameters contain indicators of the process data for PDOs, including the index, subindex and mapping object length. The sub-index 0 indicates the number (N) of mapping objects in the PDO, and the maximum length of each PDO is 4 x N bytes. One or multiple objects can be mapped simultaneously. Sub-indexes 1 to N indicate the mapping content. Table 3-27 defines mapping parameters.

Places	31	...	16	15	...	8	7	...	0
Description	Index			Sub-index			Object Length		

The index and sub-index together define the position of an object in the object dictionary. The object length indicates the bit length of the object in hexadecimal, as shown below.

Object Length	Bit Length
08h	8-bit
10h	16-bit
20h	32-bit

For example, the mapping parameter of the 16-bit control word 6040.00h is 60400010h.

- PDO mapping steps:
  Abide by the following procedures for PDO mapping:

1. Configure the mapping group of PDO. Write 0 to sub-index 00h of 1C12h (or 1C13h).

   a. Clear the original mapping group. Write 0 to sub-index 00h of 1C12h (or 1C13h) to clear the original mapping group.
   b. Write the PDO mapping group. Write the mapping group according to application needs. Pre-write values of 1600h/1701h...1705h to 1C12h and values of 1A00h/1B01h...1B04h to 1C13h. Note: Only 1600h and 1A00h, and 1601h and 1A01h are are configurable mapping groups.
   c. Write the total number of this PDO mapping group to sub-index 0 of 1C12h (or 1C13h).

2. Configure the mapping objects of PDO. Write 0 to sub-index 00h of 1600h (or 1A00h) and 1601h(or 1A01h).

   a. Clear the original mapping objects. Write 0 to sub-index 00h of 1600h (or 1A00h), and 1601h (or 1A01h) to clear the original mapping objects.
   b. Write the PDO mapping content. Write the mapping content to sub-index 1...10 of the mapping parameter based on object parameter definitions in XML file. Only mappable objects can be configured as PDO mapping content.
   c. Write the total number of mapping objects. Write the number of mapping objects in step b to sub-index 0.

Note

- Configure the PDO only when the EtherCAT state machine is in pre-operational state ("2" displayed on the keypad). Otherwise, an error will be reported.
- Do not save the PDO configuration parameters to EEPROM. Configure the mapping objects again each time upon power-on. Otherwise, the mapping objects are the default parameters of the servo drive.

An SDO fault code will be returned when the following operations are under execution:

- Modify PDO parameters in status other than pre-operational.
- Write a value outside the range of 1600h/1601h/1701h...1705h to 1C12h. Write a value outside the range of 1A00h/ 1A01h/1B01h...1B04h to 1C13h.

4.4.2 Service Data Object (SDO)

The EtherCAT SDO is used to transfer non-cyclic data, such as communication parameter configuration and servo drive parameter configuration. The CoE service types of EtherCAT include:

- Emergency message
- SDO request:
- SDO response:
- TxPDO
- RxPDO
- Remote TxPDO transmission request
- Remote RxPDO transmission request
- SDO information.

The SV680N-INT series supports SDO request and SDO response.

4.5 Communication Parameters

Parameter address structure

Parameter access address: index+subindex, both of which are in hexadecimal.

CiA402 establishes the following restrictions on the parameter address:

Index (Hex)	Description
0000h to 0FFFh	Data type
1000h to 1FFFh	CoE communication object
2000h to 5FFFh	Manufacturer-specific object
6000h to 9FFFh	Profile object
A000h–FFFFh	Reserved

System parameter setting

Set related parameters to allow the SV680N-INT servo drive to be connected to the EtherCAT fieldbus network.

☆ Related parameters:

Parameter	Communication Address	Name	Value	Default	Unit	Change Mode
H02.00	2002-01h	Control mode	0: Velocity mode 1: Position mode 2: Torque mode 7: Technology segment 9: EtherCAT mode	9	-	At stop
H0E.01	200E-02h	Save objects written through communication to e2prom	0: Not save 1: Save parameters written through communication to e2prom 2: Save object dictionaries written through communication to e2prom 3: Save parameters and object dictionaries written through communication to e2prom 4: Save object dictionaries written before communication (OP) to e2prom	4	-	Real-time
H0E.21	200E-16h	EtherCAT slave alias	0 to 65535	0	-	At stop

Note

Before saving parameters to EEPROM, set H0E.01h to a proper value. Otherwise, parameters will be restored to default values at next power-on. It is recommended to set H0E.01 to 0 after parameters are set properly. This is to prevent damage to the EEPROM device caused by prolonged writing process.

5 Communication Configuration Instance

5.1 Modbus Communication Configuration Case [P]

5.1.1 Communication Overview

The following describes the Modbus RTU communication connection between Inovance H2U and the SV680P-INT series servo drive. It can be achieved by a configuration table or program. In this case, H06.03 (Write speed) and H0b.00 (Read speed) are used for illustration.

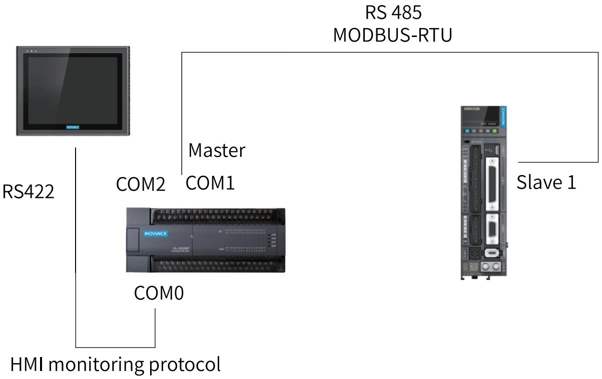


Figure 5-1 Schematic and wiring

5.1.2 Wiring of Modbus RTU Communication Between SV680P-INT and Third-Party PLCs

Inovance H2U and SV680P-INT

Name	Model	Quantity	Remarks
PLC	H2U-1616MT/MR	1 piece	-
Inovance SV680P-INT series servo drive and applicable motor	SV680PT012I-INT MS1H3-*******	1 set	-

COM1 Terminal Layout on PLC Side		CN3/CN4 Terminal Layout on Drive Side	
Signal Name	Pin No.	Signal Name	Pin No.
RS485+	1	RS485+	4
RS485-	2	RS485-	5
-	-	PE (shield layer)	Enclosure

Siemens PLC and SV680P-INT

Siemens S7200 PLC		CN3/CN4 Terminal Layout on Drive Side	
PLC PORT0-RS485	Pin No.	Signal Name	Pin No.
Data+	3	RS485+	4
Data-	8	RS485-	5
PE (shield layer)	Enclosure	PE (shield layer)	Enclosure

Mitsubishi FX3U and SV680P-INT

Mitsubishi FX3U PLC		CN3/CN4 Terminal Layout on Drive Side	
FX3U-485-BD	Pin No.	Signal Name	Pin No.
SDA	Short	RS485+	4
RDA			
SDB	Short	RS485-	5
RDB			
SG	Enclosure	PE (shield layer)	Enclosure

Setting communication parameters through GX PLC software (initialization of communication port 1):

1. Communication port 1 parameter setting (RS485, 19200, 7, N, 1)
2. LD M8002
3. Initial ON
4. MOV H0C91 D8120
5. Communication port 1 setting
6. SET M8161
7. Communication format: 8-bit

Using two major commands (See the user guide for FX3U communication.)

- RS D100 K8 D120 K8
  - D100: station No. being "?"
  - D120: starting address for data receiving (8 bytes)
- CRC D100 D106 K6
  - D100: station No. being "?"
  - D106: CRC checked address

Omron PLC and SV680P-INT

Omron CP1L		CN3/CN4 Terminal Layout on Drive Side	
PLC PORT0-RS485	Pin No.	Signal Name	Pin No.
SDB+	-	RS485+	4
SDA-	-	RS485-	5
PE (shield layer)	Enclosure	PE (shield layer)	Enclosure

Note

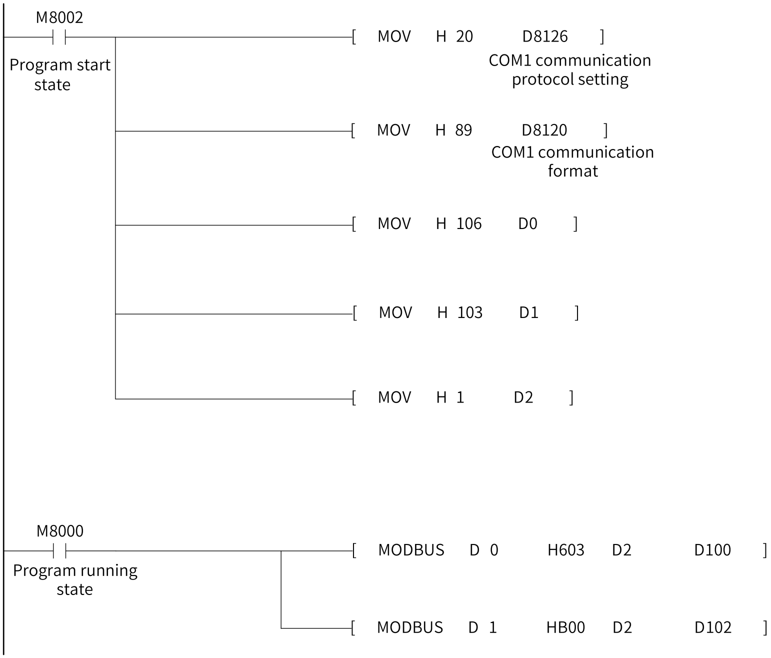
Set 2, 3, 5, and 6 on the DIP switch to ON, and others to OFF. The DIP switch is on the back of PLC communication card.

5.1.3 Servo Parameter Settings

Para.	Setting	Description	Remarks
H0E.00	1	Drive axis address	-
H0E.80	5	Modbus baud rate	5: 9600 bps
H0E.84	1	Modbus communication data sequence	0: High 16 bits before low 16 bits 1: Low 16 bits before high 16 bits

5.1.4 PLC Program Examples

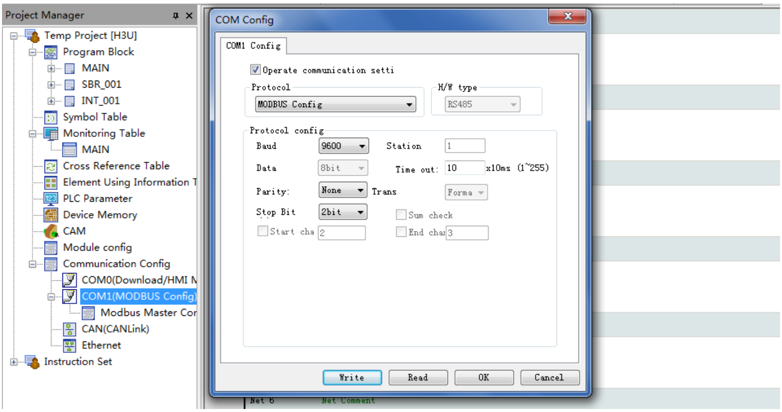
Communication connection implemented through programming



```
M8002
[ MOV   H 20    D8126 ]
[ MOV   H 89    D8120 ]
[ MOV   H 106   D0 ]
[ MOV   H 103   D1 ]
[ MOV   H 1     D2 ]

M8000
[ MODBUS   D 0   H603   D2   D100 ]
[ MODBUS   D 1   HB00   D2   D102 ]
```

Program start state
COM1 communication protocol setting
COM1 communication format
Program running state

Communication connection implemented through configuration table



☑ Operate communication setti

Protocol: MODBUS Config

H/W type: RS485

Protocol config

Baud: 9600

Station: 1

Data: 8bit

Time out: 10 x10ms (1~255)

Parity: None

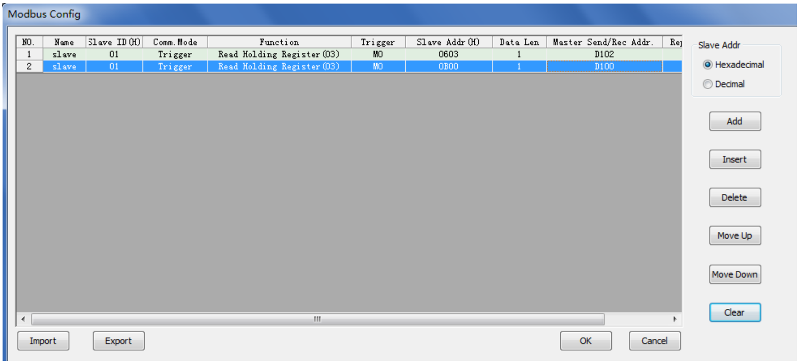
Trans: Forma

Stop Bit: 2bit

□ Sum check

□ Start cha 2

□ End cha 3



NO.	Name	Slave ID (H)	Comm. Mode	Function	Trigger	Slave Addr (H)	Data Len	Master Send/Rec Addr.	Re
1	slave	01	Trigger	Read Holding Register (03)	M0	0603	1	D102	
2	slave	01	Trigger	Read Holding Register (03)	M0	0B00	1	D100	

Slave Addr

● Hexadecimal

○ Decimal

5.2 CANopen Communication Configuration Case [P]

5.2.1 Connecting SV680P-INT to Schneider 3S Master

The following takes the position control mode as example. For details on the position control mode, see section "Position Control Mode" in SV680-INT Series Servo Drive Function Guide.

In the position control mode, assignment of objects used as PDO are listed in the following table.

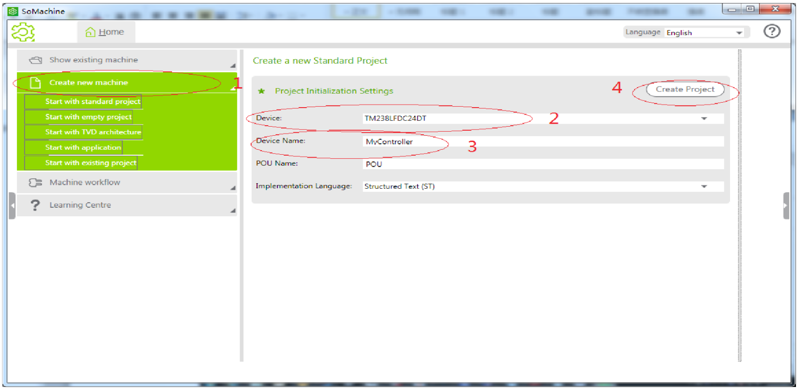
Table 5–1 PDO mapping allocation

PDO	Object	Description	Bit Length
RPDO1	6040.00h	Control word	Uint16
	6060.00h	Mode selection	Int8
RPDO2	6081.00h	Profile velocity	Uint32
	607A.00h	Target position	Int32
TPDO1	6041.00h	Status word	Uint16
	6061.00h	Operation mode display	Int8
TPDO2	606C.00h	Speed feedback	Int32
	6064.00h	Position actual value	Int32
TPDO3	H0b.26	Phase current feedback	Uint16

SDO is used to write acceleration 6083h, deceleration 6084h and emergency stop 605Ah.

SoMachine is the software tool of Schneider 3S series master. This section describes how to connect the SV680P-INT servo drive to Schneider M238.

1. Start SoMachine and click **Create new machine** based on a standard project. Select a master device, for example, TM238LFDC24DT, modify the device name, and click **Create Project**, as shown below.



Create a new Standard Project

Project Initialization Settings

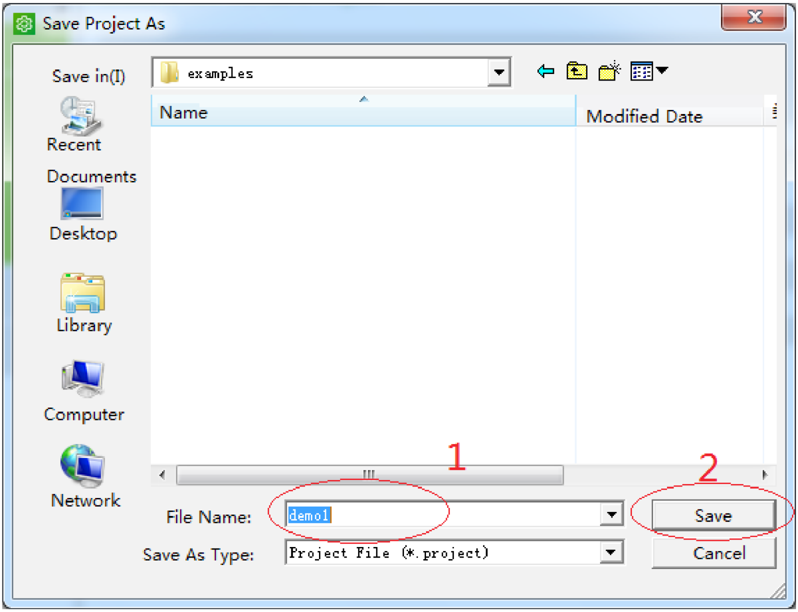
Device: TM238LFDC24DT

Device Name: MyController

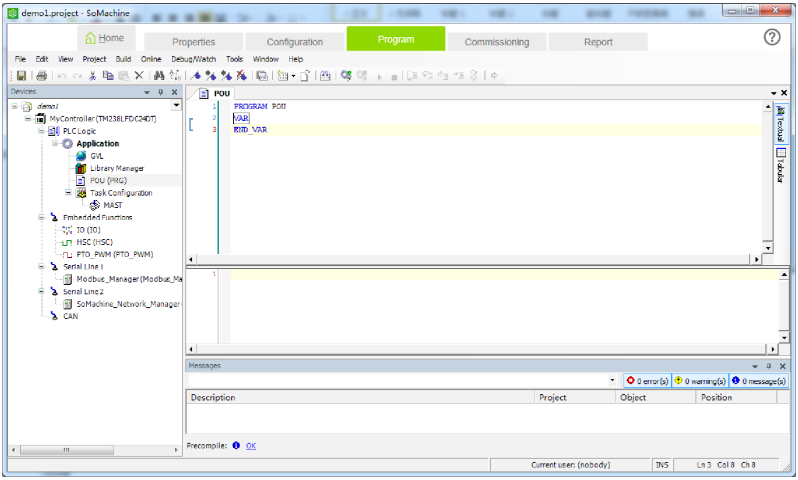
POU Name: POU

Implementation Language: Structured Text (ST)

2. Enter the file name and click **Save** in the dialog box displayed.

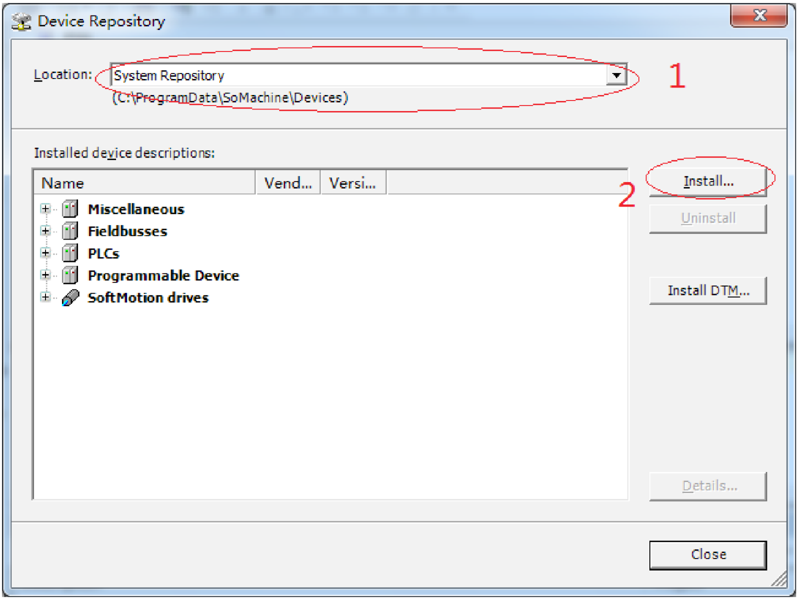


3. The following interface appears.

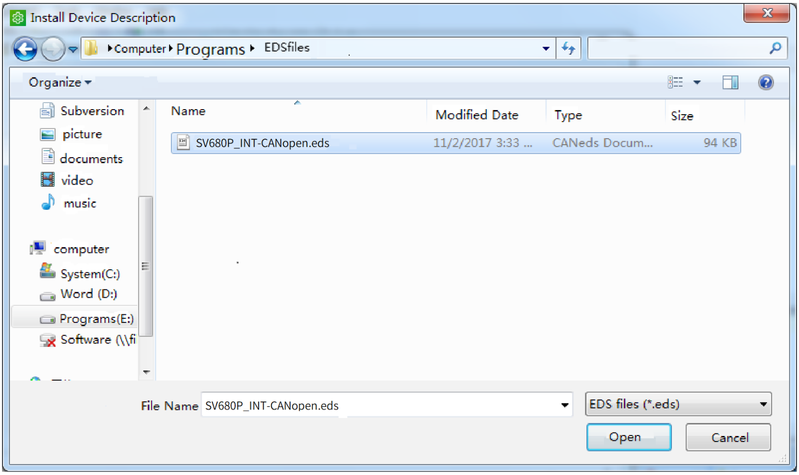


```
PROGRAM POU
VAR
END_VAR
```

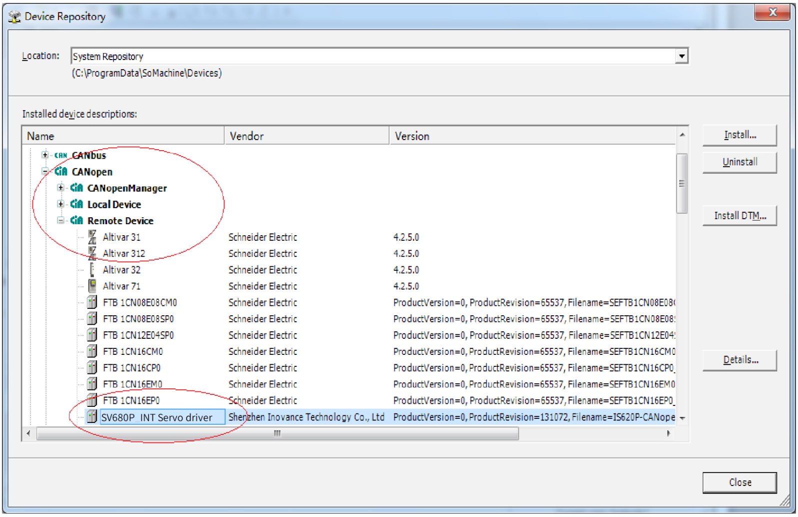
4. Choose **Tools** > **Device Repository** in the toolbar. The **Device Repository** dialog box is displayed. (If the EDS file is imported, steps Step 4 to 6 can be omitted.)



5. As shown in the preceding interface, select **System Repository** and click **Install**. Select a directory for saving the EDS file, as shown below.



6. Click **Open**. The EDS file of the SV680P-INT servo drive is imported into SoMachine. In the **Device Repository** dialog box, you can choose **Field Bus** > **CANopen** > **Remote Device** to view devices..



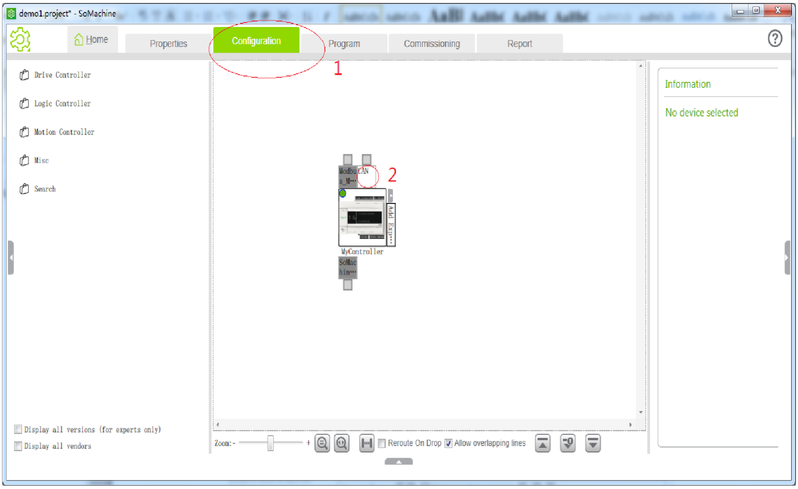
Device Repository

Location: System Repository
(C:\ProgramData\SoMachine\Devices)

Installed device descriptions:

Name	Vendor	Version
CANbus		
CANopen		
CANopenManager		
Local Device		
Remote Device		
Altivar 31	Schneider Electric	4.2.5.0
Altivar 312	Schneider Electric	4.2.5.0
Altivar 32	Schneider Electric	4.2.5.0
Altivar 71	Schneider Electric	4.2.5.0
FTB 1CN08E08CM0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN08E08(
FTB 1CN08E08SP0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN08E08!
FTB 1CN12E04SP0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN12E04!
FTB 1CN16CM0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN16CM0
FTB 1CN16CP0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN16CP0_
FTB 1CN16EM0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN16EM0
FTB 1CN16EP0	Schneider Electric	ProductVersion=0, ProductRevision=65537, Filename=SEFTB1CN16EP0_
SV680P_INT Servo driver	Shenzhen Inovance Technology Co., Ltd	ProductVersion=0, ProductRevision=131072, Filename=IS620P-CANope

7. Close the preceding dialog box and click **Configuration**. In the interface displayed, only M238 master is available. Click **CAN** on the master station.

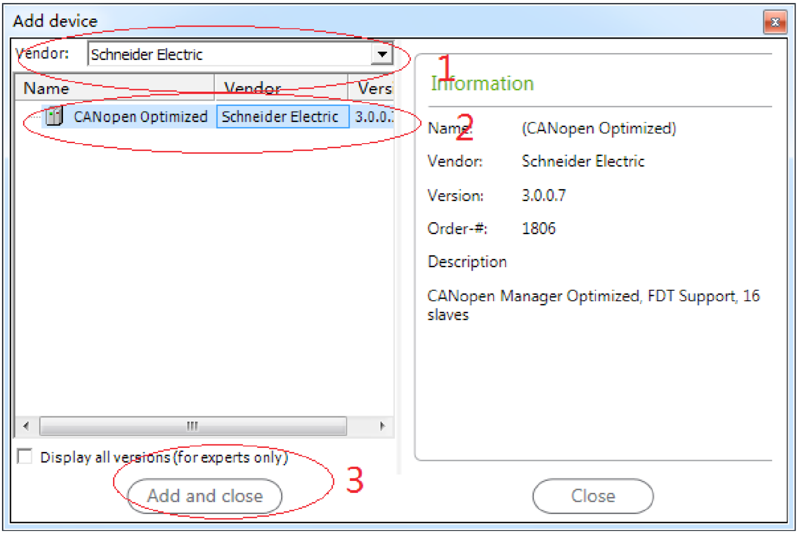


demo1.project* - SoMachine

No device selected

□ Display all versions (for experts only)
□ Display all vendors

8. The **Add device** dialog box is displayed. Add a CANopen gateway, select **Schneider Electric** for **Supplier**, select **CANopen Optimized**, click the **Add and close**.



Add device

Vendor: Schneider Electric

Name	Vendor	Vers
CANopen Optimized	Schneider Electric	3.0.0.

Information

Name: (CANopen Optimized)

Vendor: Schneider Electric

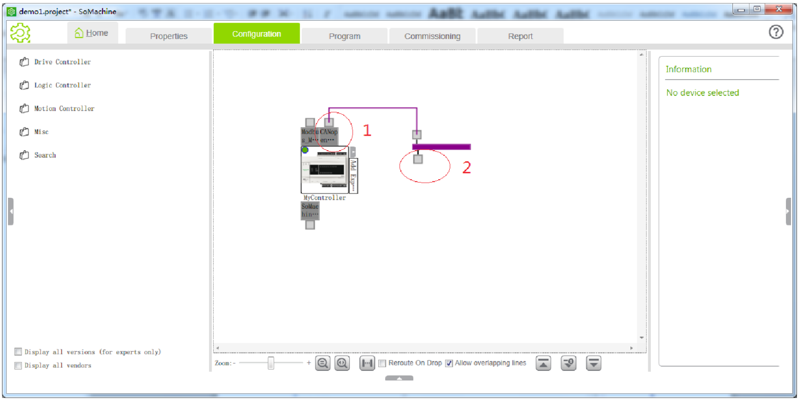
Version: 3.0.0.7

Order-#: 1806

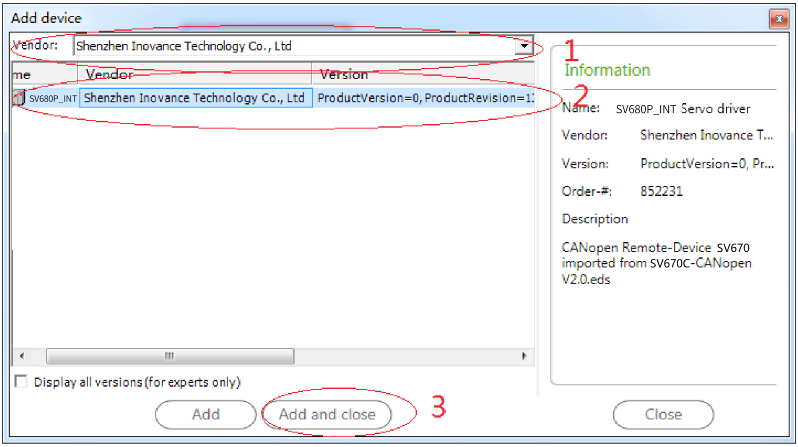
Description

CANopen Manager Optimized, FDT Support, 16 slaves

9. Now, the CANopen gateway appears in the interface. Click the position indicated by **2**.



10. The **Add device** dialog box appears again. Select **Inovance** as the vendor and **SV680** as the device, and then click **Add and close**.



Add device

Vendor: Shenzhen Inovance Technology Co., Ltd

me	Vendor	Version
SV680P_INT	Shenzhen Inovance Technology Co., Ltd	ProductVersion=0, ProductRevision=1:

Information

Name: SV680P_INT Servo driver

Vendor: Shenzhen Inovance T...

Version: ProductVersion=0, Pr...

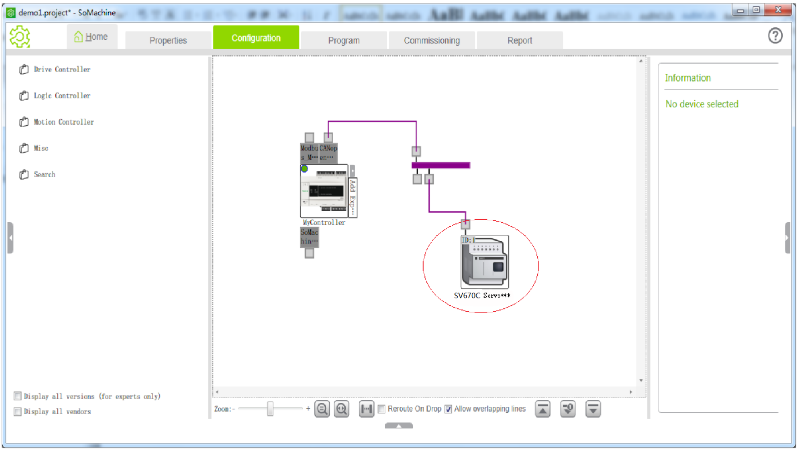
Order-#: 852231

Description

CANopen Remote-Device SV670 imported from SV670C-CANopen V2.0.eds

☐ Display all versions (for experts only)

11. Now, the SV680P-INT servo drive appears in the interface.



demo1.project* - SoMachine

Information

No device selected

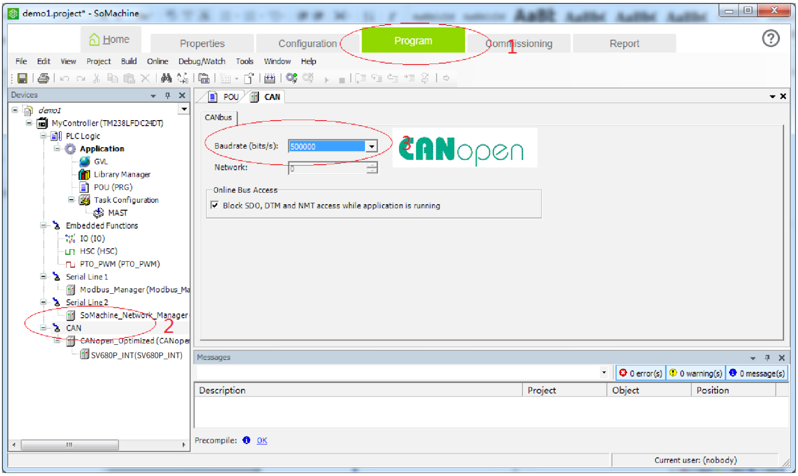
☐ Display all versions (for experts only)

☐ Display all vendors

Zoom: - +

☐ Reroute On Drop ☑ Allow overlapping lines

12. Click **Program** and double-click **CAN** on the left to select a proper baud rate. 500 Kbps is selected here.



Baudrate (bits/s): 500000

Network: 0

Online Bus Access

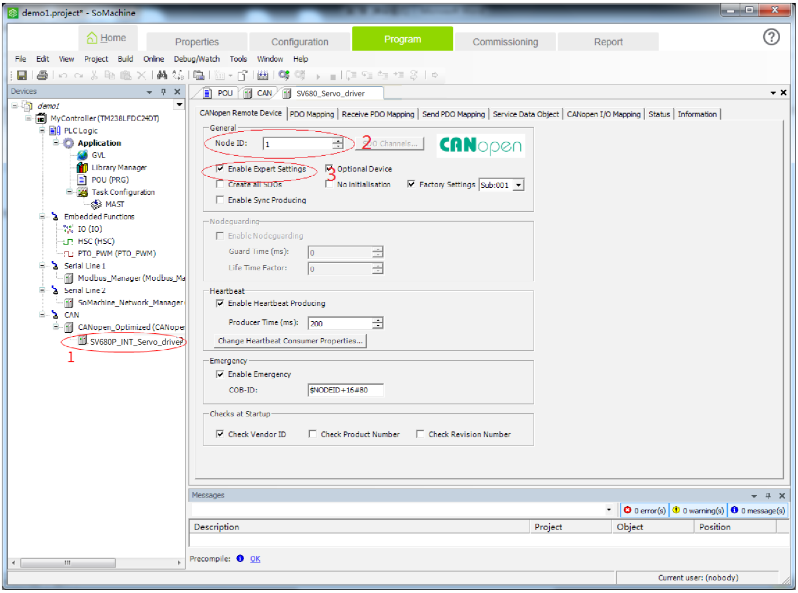
☑ Block SDO, DTM and NMT access while application is running

0 error(s) 0 warning(s) 0 message(s)

Precompile: OK

Current user: (nobody)

13. Double-click **SV680P_INT_Servo_Driver** on the left. The node ID can be modified. Check **Enable Expert Settings**.



General

Node ID: 1

☑ Enable Expert Settings ☑ Optional Device

□ Create all SDOs □ No initialisation ☑ Factory Settings Sub:001

□ Enable Sync Producing

Nodeguarding

□ Enable Nodeguarding

Guard Time (ms): 0

Life Time Factor: 0

Heartbeat

☑ Enable Heartbeat Producing

Producer Time (ms): 200

Change Heartbeat Consumer Properties...

Emergency

☑ Enable Emergency

COB-ID: $NODEID+16#80

Checks at Startup

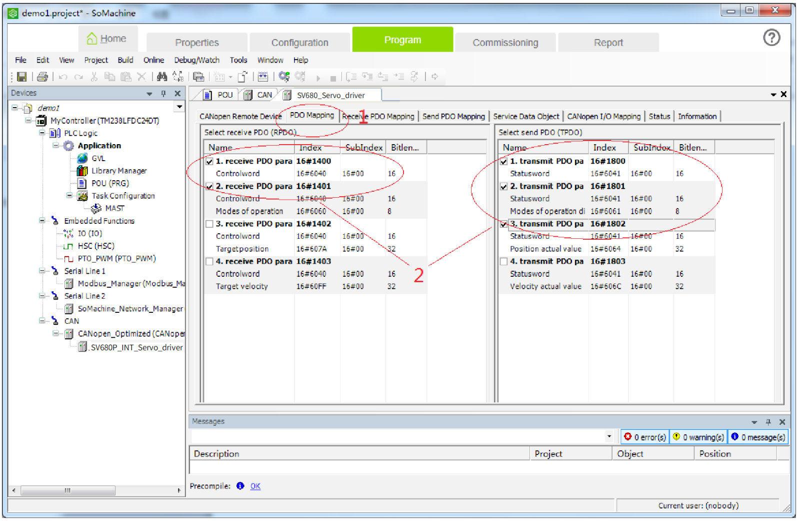
☑ Check Vendor ID □ Check Product Number □ Check Revision Number

0 error(s) 0 warning(s) 0 message(s)

Precompile: OK

Current user: (nobody)

14. Click **PDO Mapping** and check two RPDO and three TPDO.



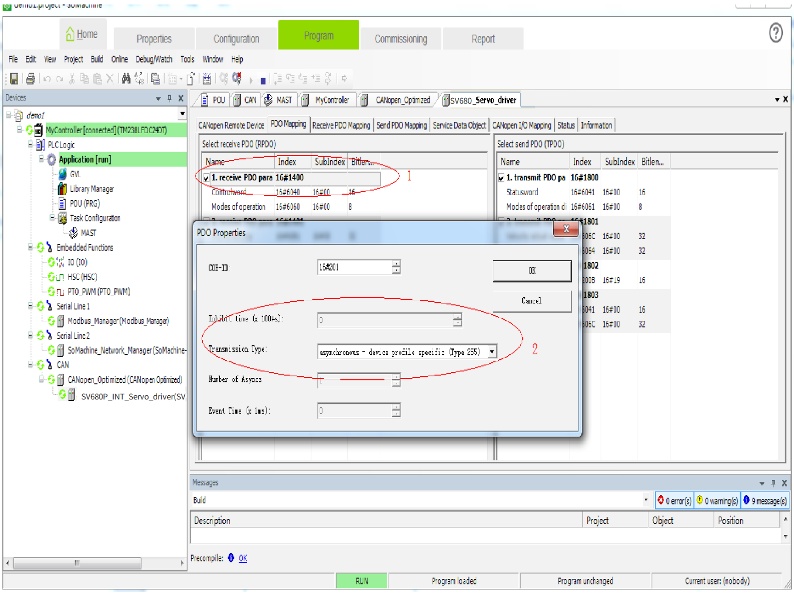
Select receive PDO (RPDO)

Name	Index	SubIndex	Bitlen...
☑ 1. receive PDO para	16#1400		
Controlword	16#6040	16#00	16
☑ 2. receive PDO para	16#1401		
Controlword	16#6040	16#00	16
Modes of operation	16#6060	16#00	8
□ 3. receive PDO para	16#1402		
Controlword	16#6040	16#00	16
Targetposition	16#607A	16#00	32
□ 4. receive PDO para	16#1403		
Controlword	16#6040	16#00	16
Target velocity	16#60FF	16#00	32

Select send PDO (TPDO)

Name	Index	SubIndex	Bitlen...
☑ 1. transmit PDO pa	16#1800		
Statusword	16#6041	16#00	16
☑ 2. transmit PDO pa	16#1801		
Statusword	16#6041	16#00	16
Modes of operation di	16#6061	16#00	8
☑ 3. transmit PDO pa	16#1802		
Statusword	16#6041	16#00	16
Position actual value	16#6064	16#00	32
□ 4. transmit PDO pa	16#1803		
Statusword	16#6041	16#00	16
Velocity actual value	16#606C	16#00	32

15. Double-click **RPDO1**. The **PDO Properties** dialog box is displayed. Modify **Transmission Type** to **Type 255**. Perform the same operation for other PDOs.



PDO Properties

COB-ID: 16#201

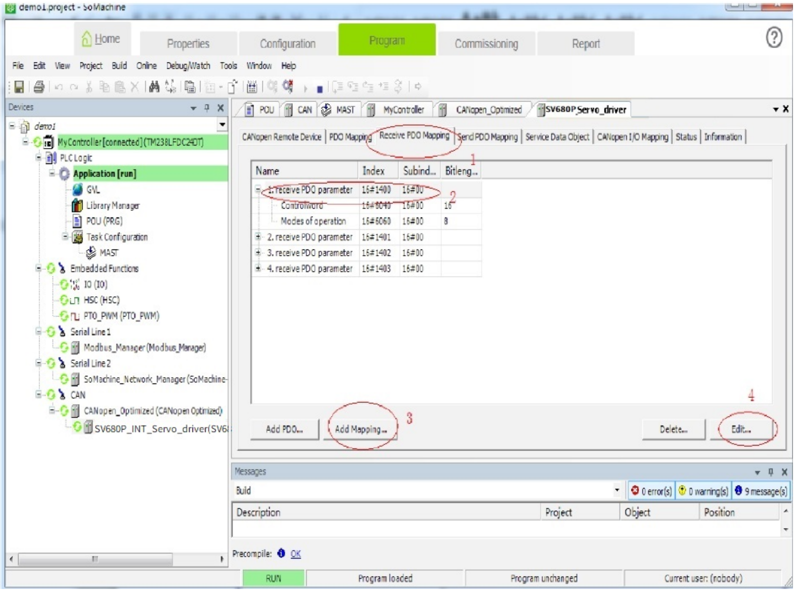
Inhibit time (x 100μs): 0

Transmission Type: asynchronous - device profile specific (Type 255)

Number of Asyncs 1

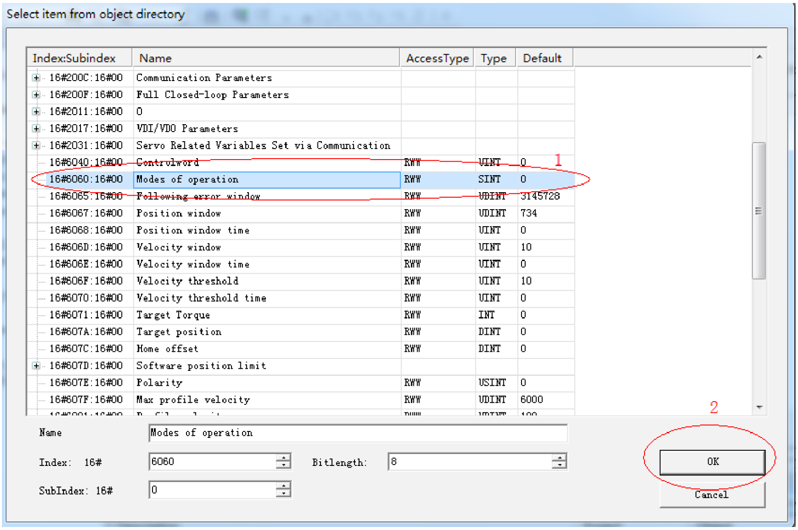
Event Time (x 1ms): 0

16. Select **Receive PDO Mapping** and click **receive PDO parameter**. Click **Add Mapping** or select a mapping and click **Edit**.



Name	Index	Subind...	Bitleng...
1. receive PDO parameter	16#1400	16#00	
Controlword	16#6040	16#00	16
Modes of operation	16#6060	16#00	8
2. receive PDO parameter	16#1401	16#00	
3. receive PDO parameter	16#1402	16#00	
4. receive PDO parameter	16#1403	16#00	

17. Select the proper mapping object in the dialog box displayed according to *"Table 5–1 " on page 79*.


Select item from object directory

Index:Subindex	Name	AccessType	Type	Default
16#200C:16#00	Communication Parameters			
16#200F:16#00	Full Closed-loop Parameters			
16#2011:16#00	0			
16#2017:16#00	VDI/VDO Parameters			
16#2031:16#00	Servo Related Variables Set via Communication			
16#6040:16#00	Controlword	RWW	UINT	0
16#6060:16#00	Modes of operation	RWW	SINT	0
16#6065:16#00	Following error window	RWW	UDINT	3145728
16#6067:16#00	Position window	RWW	UDINT	734
16#6068:16#00	Position window time	RWW	UINT	0
16#606D:16#00	Velocity window	RWW	UINT	10
16#606E:16#00	Velocity window time	RWW	UINT	0
16#606F:16#00	Velocity threshold	RWW	UINT	10
16#6070:16#00	Velocity threshold time	RWW	UINT	0
16#6071:16#00	Target Torque	RWW	INT	0
16#607A:16#00	Target position	RWW	DINT	0
16#607C:16#00	Home offset	RWW	DINT	0
16#607D:16#00	Software position limit			
16#607E:16#00	Polarity	RWW	USINT	0
16#607F:16#00	Max profile velocity	RWW	UDINT	6000

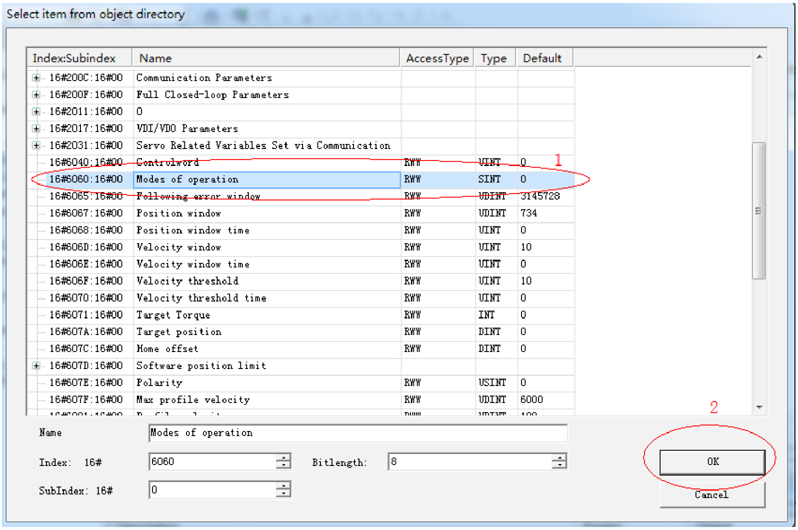
Name: Modes of operation

Index: 16# 6060

Bitlength: 8

SubIndex: 16# 0

18. After the mapping object is added, the RPDO mapping is shown as follows.



Select item from object directory

Index:Subindex	Name	AccessType	Type	Default
16#200C:16#00	Communication Parameters			
16#200F:16#00	Full Closed-loop Parameters			
16#2011:16#00	0			
16#2017:16#00	VDI/VDO Parameters			
16#2031:16#00	Servo Related Variables Set via Communication			
16#6040:16#00	Controlword	RW	UINT	0
16#6060:16#00	Modes of operation	RW	SINT	0
16#6065:16#00	Following error window	RW	UDINT	3145728
16#6067:16#00	Position window	RW	UDINT	734
16#6068:16#00	Position window time	RW	UINT	0
16#606D:16#00	Velocity window	RW	UINT	10
16#606E:16#00	Velocity window time	RW	UINT	0
16#606F:16#00	Velocity threshold	RW	UINT	10
16#6070:16#00	Velocity threshold time	RW	UINT	0
16#6071:16#00	Target Torque	RW	INT	0
16#607A:16#00	Target position	RW	DINT	0
16#607C:16#00	Home offset	RW	DINT	0
16#607D:16#00	Software position limit			
16#607E:16#00	Polarity	RW	USINT	0
16#607F:16#00	Max profile velocity	RW	UDINT	6000

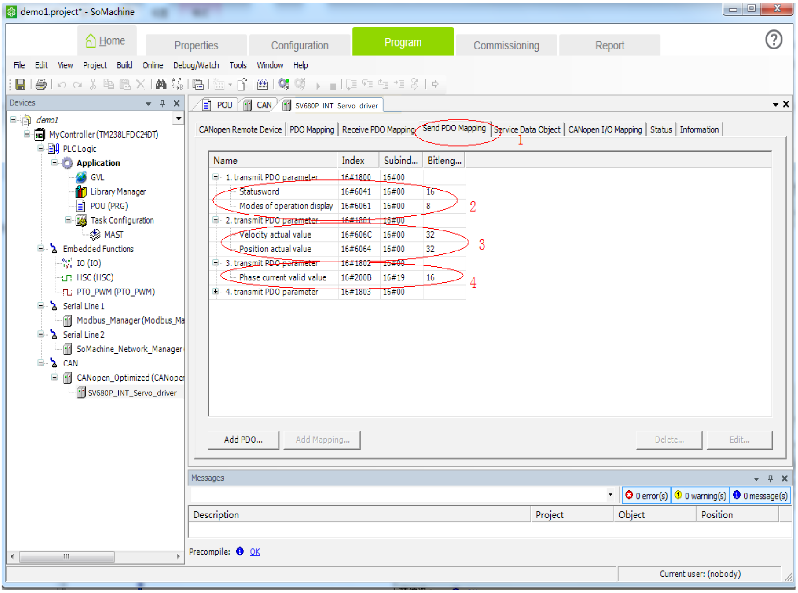
Name: Modes of operation

Index: 16# 6060

Bitlength: 8

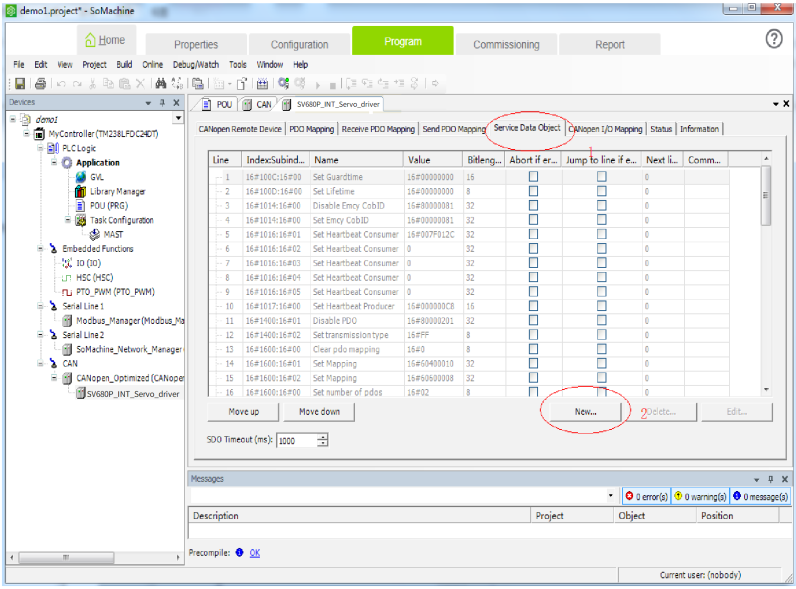
SubIndex: 16# 0

19. Similarly, click **Send PDO Mapping** and perform configuration according to *"Table 5–1 " on page 79*, as shown below.



Name	Index	Subind...	Bitleng...
1. transmit PDO parameter	16#1800	16#00	
Statusword	16#6041	16#00	16
Modes of operation display	16#6061	16#00	8
2. transmit PDO parameter	16#1801	16#00	
Velocity actual value	16#606C	16#00	32
Position actual value	16#6064	16#00	32
3. transmit PDO parameter	16#1802	16#00	
Phase current valid value	16#200B	16#19	16
4. transmit PDO parameter	16#1803	16#00	

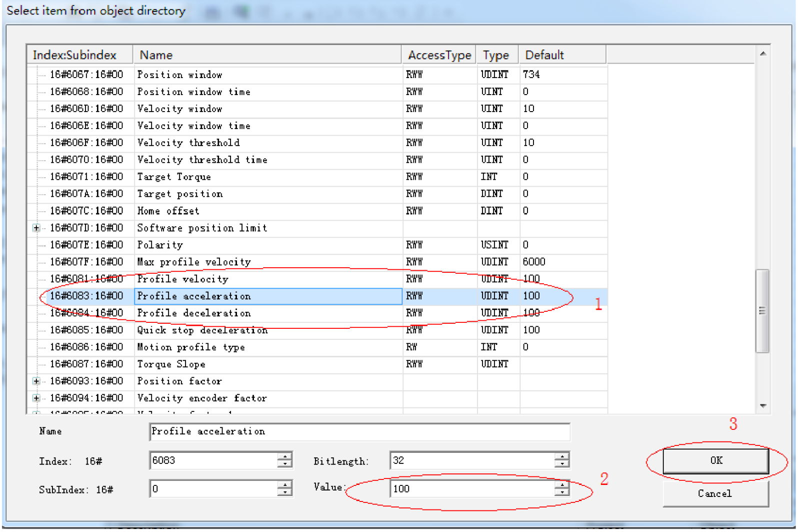
20. Click the **Service Data Object** and click **New** to add a required SDO. (Optional) (If default values are used, steps 20 to 22 can be omitted)



Line	Index:Subind...	Name	Value	Bitleng...	Abort if er...	Jump to line if e...	Next li...	Comm...
1	16#100C:16#00	Set Guardtime	16#00000000	16	□	□	0	
2	16#100D:16#00	Set Lifetime	16#00000000	8	□	□	0	
3	16#1014:16#00	Disable Emcy CobID	16#80000081	32	□	□	0	
4	16#1014:16#00	Set Emcy CobID	16#00000081	32	□	□	0	
5	16#1016:16#01	Set Heartbeat Consumer	16#007F012C	32	□	□	0	
6	16#1016:16#02	Set Heartbeat Consumer	0	32	□	□	0	
7	16#1016:16#03	Set Heartbeat Consumer	0	32	□	□	0	
8	16#1016:16#04	Set Heartbeat Consumer	0	32	□	□	0	
9	16#1016:16#05	Set Heartbeat Consumer	0	32	□	□	0	
10	16#1017:16#00	Set Heartbeat Producer	16#000000C8	16	□	□	0	
11	16#1400:16#01	Disable PDO	16#80000201	32	□	□	0	
12	16#1400:16#02	Set transmission type	16#FF	8	□	□	0	
13	16#1600:16#00	Clear pdo mapping	16#0	8	□	□	0	
14	16#1600:16#01	Set Mapping	16#60400010	32	□	□	0	
15	16#1600:16#02	Set Mapping	16#60600008	32	□	□	0	
16	16#1600:16#00	Set number of pdos	16#02	8	□	□	0	

SDO Timeout (ms): 1000

21. Select the corresponding SDO in the list. You can modify the value and click **OK**. (Optional)



Index:Subindex	Name	AccessType	Type	Default
16#6067:16#00	Position window	RW	UDINT	734
16#6068:16#00	Position window time	RW	UINT	0
16#606D:16#00	Velocity window	RW	UINT	10
16#606E:16#00	Velocity window time	RW	UINT	0
16#606F:16#00	Velocity threshold	RW	UINT	10
16#6070:16#00	Velocity threshold time	RW	UINT	0
16#6071:16#00	Target Torque	RW	INT	0
16#607A:16#00	Target position	RW	DINT	0
16#607C:16#00	Home offset	RW	DINT	0
16#607D:16#00	Software position limit			
16#607E:16#00	Polarity	RW	USINT	0
16#607F:16#00	Max profile velocity	RW	UDINT	6000
16#6081:16#00	Profile velocity	RW	UDINT	100
16#6083:16#00	Profile acceleration	RW	UDINT	100
16#6084:16#00	Profile deceleration	RW	UDINT	100
16#6085:16#00	Quick stop deceleration	RW	UDINT	100
16#6086:16#00	Motion profile type	RW	INT	0
16#6087:16#00	Torque Slope	RW	UDINT	
16#6093:16#00	Position factor			
16#6094:16#00	Velocity encoder factor			

Name: Profile acceleration

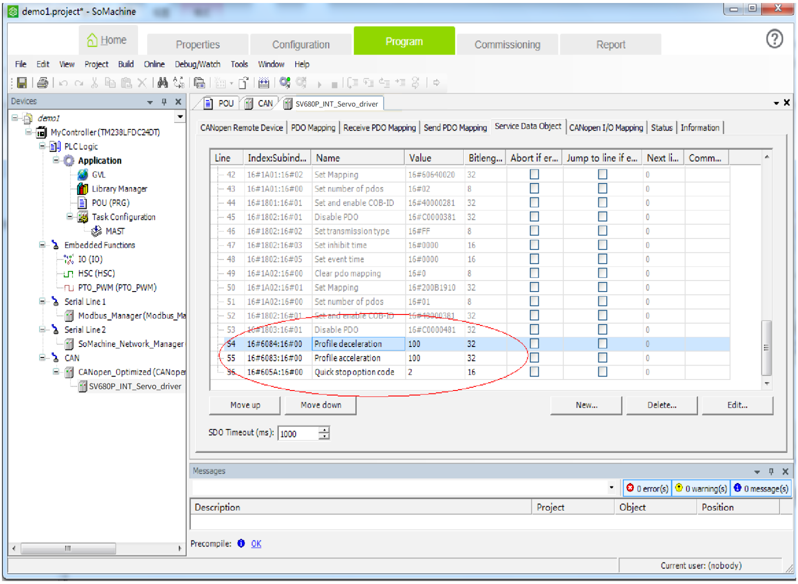
Index: 16# 6083

SubIndex: 16# 0

Bitlength: 32

Value: 100

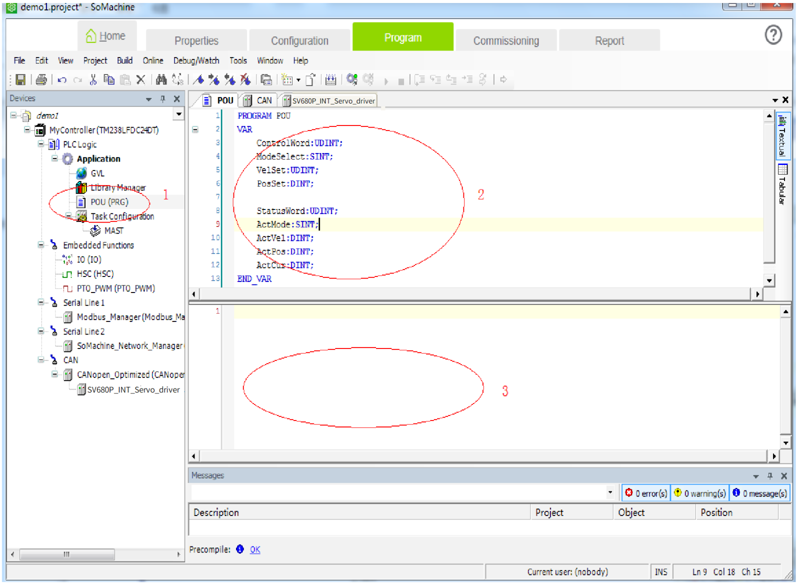
22. The newly added SDO is shown as below. (Optional)



Line	Index:Subind...	Name	Value	Bitleng...	Abort if er...	Jump to line if e...	Next li...	Comm...
42	16#1A01:16#02	Set Mapping	16#60640020	32	□	□	0	
43	16#1A01:16#00	Set number of pdos	16#02	8	□	□	0	
44	16#1801:16#01	Set and enable COB-ID	16#40000281	32	□	□	0	
45	16#1802:16#01	Disable PDO	16#C0000381	32	□	□	0	
46	16#1802:16#02	Set transmission type	16#FF	8	□	□	0	
47	16#1802:16#03	Set inhibit time	16#0000	16	□	□	0	
48	16#1802:16#05	Set event time	16#0000	16	□	□	0	
49	16#1A02:16#00	Clear pdo mapping	16#0	8	□	□	0	
50	16#1A02:16#01	Set Mapping	16#200B1910	32	□	□	0	
51	16#1A02:16#00	Set number of pdos	16#01	8	□	□	0	
52	16#1802:16#01	Set and enable COB-ID	16#40000381	32	□	□	0	
53	16#1803:16#01	Disable PDO	16#C0000481	32	□	□	0	
54	16#6084:16#00	Profile deceleration	100	32	□	□	0	
55	16#6083:16#00	Profile acceleration	100	32	□	□	0	
56	16#605A:16#00	Quick stop option code	2	16	□	□	0	

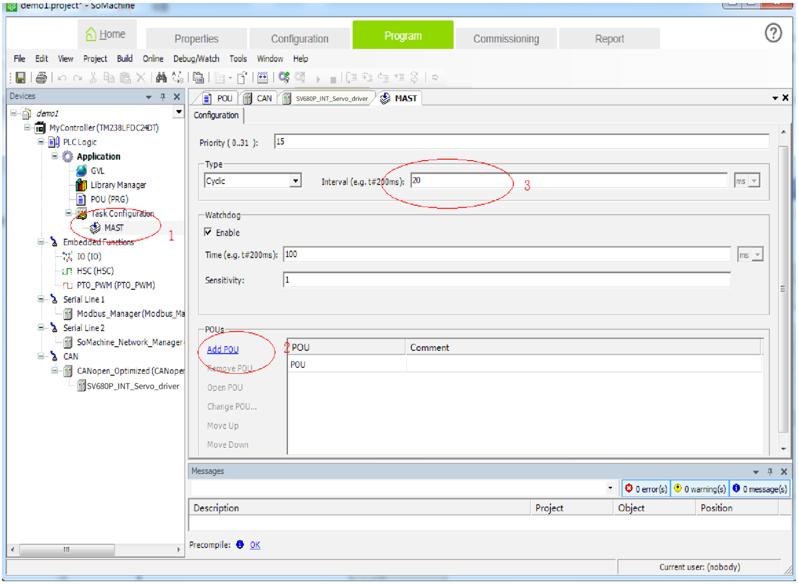
SDO Timeout (ms): 1000

23. Double-click **POU** on the left. Add variable definitions in **2** and add PLC program logic in **3**. Click **Edit** or press “F11”. If no error occurs, go to the next step.



```
PROGRAM POU
VAR
    ControlWord:UDINT;
    ModeSelect:SINT;
    VelSet:UDINT;
    PosSet:DINT;

    StatusWord:UDINT;
    ActMode:SINT;
    ActVel:DINT;
    ActPos:DINT;
    ActCur:DINT;
END_VAR
```

24. Double-click **MAST** to add the PDO, and set the program circulation interval.



Priority (0..31): 15

Type

Cyclic

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

Watchdog

☑ Enable

Time (e.g. t#200ms): 100 ms

Sensitivity: 1

POUs

POU	Comment
POU	

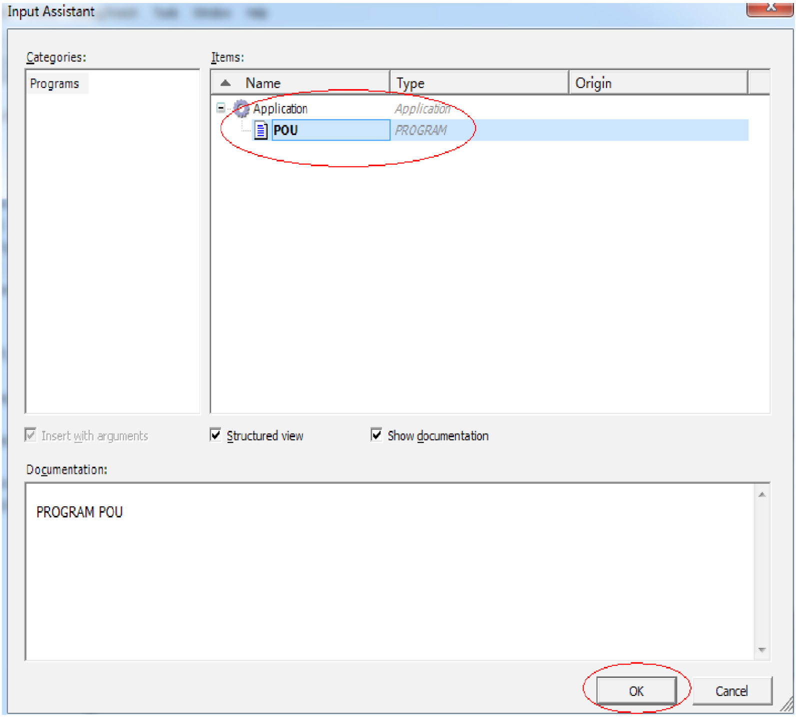
0 error(s) 0 warning(s) 0 message(s)

Description	Project	Object	Position

Precompile: OK

Current user: (nobody)

25. Select the POU added based on the following dialog box and click **OK**.



Categories:

Programs

Items:

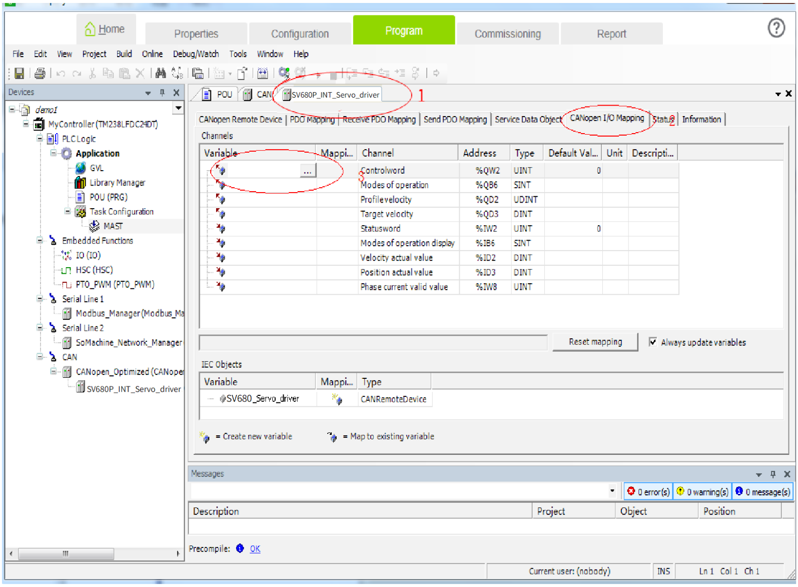
Name	Type	Origin
Application	Application	
POU	PROGRAM	

☑ Insert with arguments ☑ Structured view ☑ Show documentation

Documentation:

PROGRAM POU

26. Select **CANopen I/O Mapping** under **SV680P_INT...** and double-click the variable to display the ... button, and then click the ... button.



Channels

Variable	Mappi...	Channel	Address	Type	Default Val...	Unit	Descripti...
		Controlword	%QW2	UINT	0		
		Modes of operation	%QB6	SINT			
		Profilevelocity	%QD2	UDINT			
		Target velocity	%QD3	DINT			
		Statusword	%IW2	UINT	0		
		Modes of operation display	%IB6	SINT			
		Velocity actual value	%ID2	DINT			
		Position actual value	%ID3	DINT			
		Phase current valid value	%IW8	UINT			

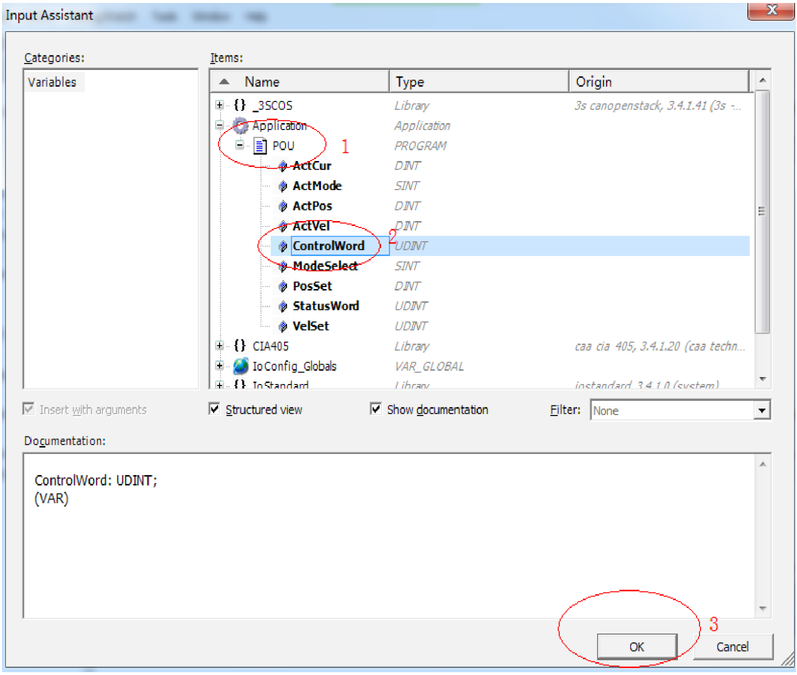
IEC Objects

Variable	Mappi...	Type
SV680_Servo_driver		CANRemoteDevice

Description	Project	Object	Position

Precompile: OK

27. Select the PLC-defined variable based on the following steps.



Variables

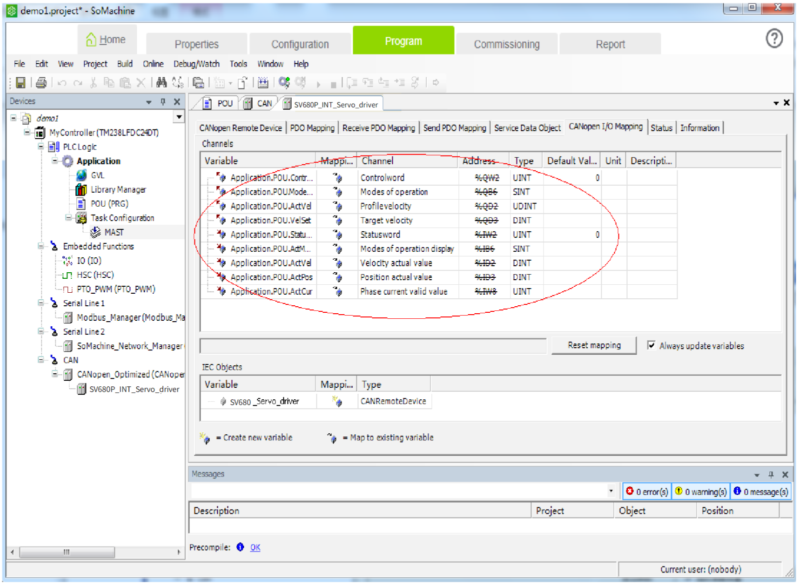
Name	Type	Origin
{} _3SCOS	Library	3s canopenstack, 3.4.1.41 (3s -...
Application	Application	
POU	PROGRAM	
ActCur	DINT	
ActMode	SINT	
ActPos	DINT	
ActVel	DINT	
ControlWord	UDINT	
ModeSelect	SINT	
PosSet	DINT	
StatusWord	UDINT	
VelSet	UDINT	
{} CIA405	Library	caa cia 405, 3.4.1.20 (caa techn...
IoConfig_Globals	VAR_GLOBAL	
{} IoStandard	Library	iostandard, 3.4.1.0 (system)

☑ Insert with arguments ☑ Structured view ☑ Show documentation Filter: None

Documentation:

ControlWord: UDINT;
(VAR)

28. Add other variables in the similar way, and the mapping is shown below.



Channels

Variable	Mappi...	Channel	Address	Type	Default Val...	Unit	Descripti...
Application.POU.Contr...		Controlword	~~%QW2~~	UINT	0		
Application.POU.Mode...		Modes of operation	~~%QB6~~	SINT			
Application.POU.ActVel		Profilevelocity	~~%QD2~~	UDINT			
Application.POU.VelSet		Target velocity	~~%QD3~~	DINT			
Application.POU.Statu...		Statusword	~~%IW2~~	UINT	0		
Application.POU.ActM...		Modes of operation display	~~%IB6~~	SINT			
Application.POU.ActVel		Velocity actual value	~~%ID2~~	DINT			
Application.POU.ActPos		Position actual value	~~%ID3~~	DINT			
Application.POU.ActCur		Phase current valid value	~~%IW8~~	UINT			

☑ Always update variables

IEC Objects

Variable	Mappi...	Type
SV680_Servo_driver		CANRemoteDevice

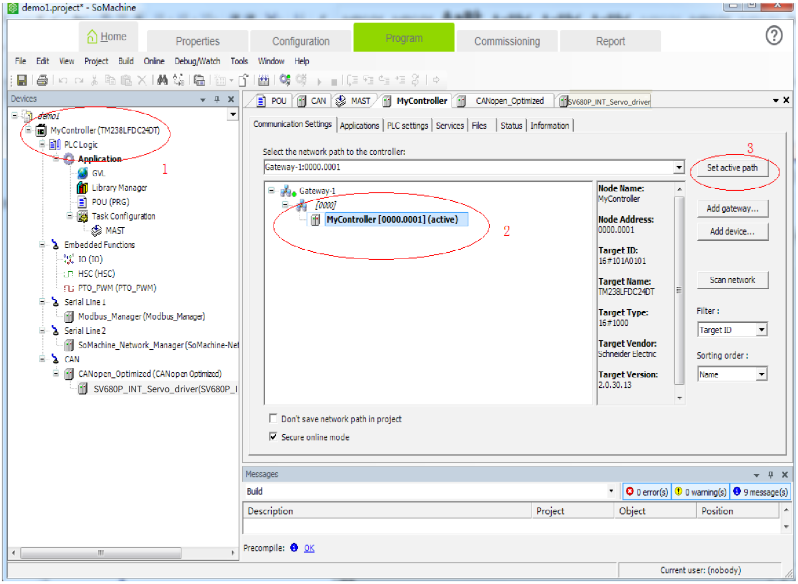
= Create new variable = Map to existing variable

0 error(s) 0 warning(s) 0 message(s)

Description	Project	Object	Position

Precompile: OK

29. Double-click the master name on the left. Select **MyController** and click **Set active path** on the right.



Select the network path to the controller: Gateway-1:0000.0001

Node Name: MyController
Node Address: 0000.0001
Target ID: 16#101A0101
Target Name: TM238LFDC24DT
Target Type: 16#1000
Target Vendor: Schneider Electric
Target Version: 2.0.30.13

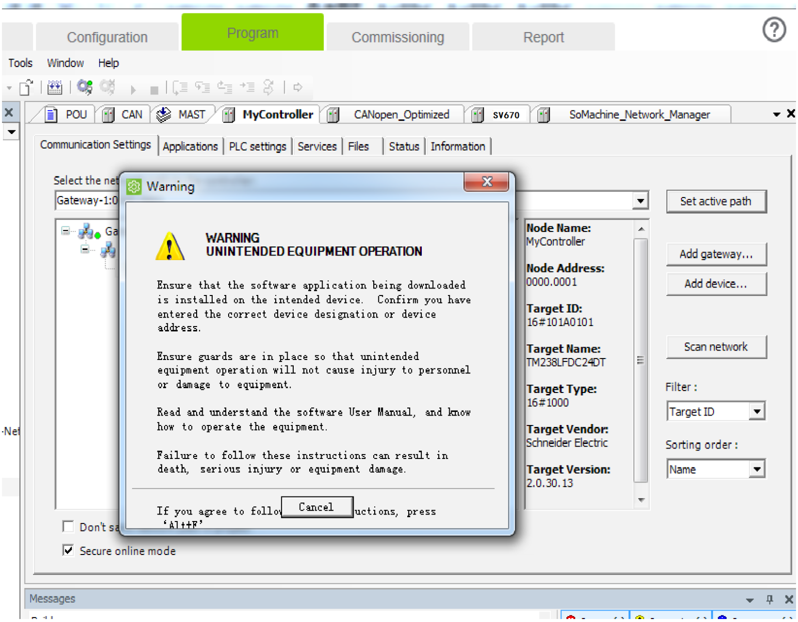
Filter : Target ID
Sorting order : Name

□ Don't save network path in project
☑ Secure online mode

Build
0 error(s) 0 warning(s) 9 message(s)

Precompile: OK

30. The following warning displays. Press Alt+F according to the instructions.



WARNING
UNINTENDED EQUIPMENT OPERATION

Ensure that the software application being downloaded is installed on the intended device. Confirm you have entered the correct device designation or device address.

Ensure guards are in place so that unintended equipment operation will not cause injury to personnel or damage to equipment.

Read and understand the software User Manual, and know how to operate the equipment.

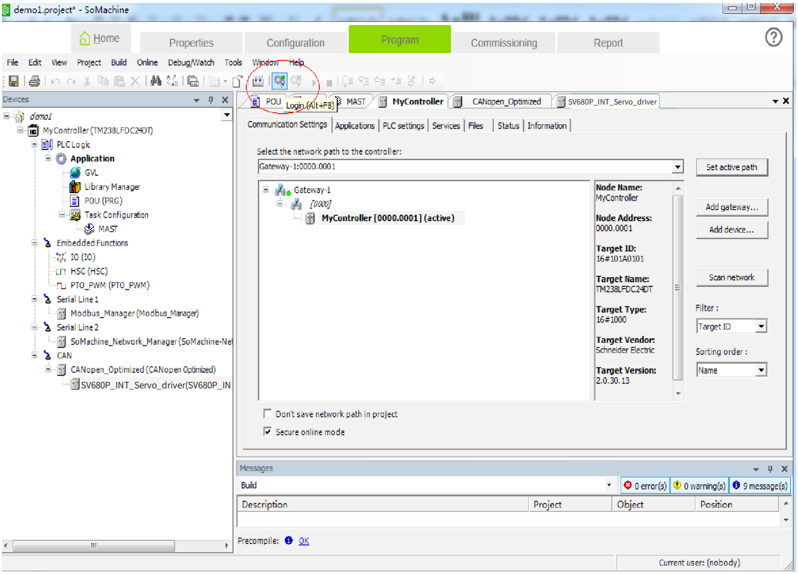
Failure to follow these instructions can result in death, serious injury or equipment damage.

If you agree to follo[Cancel]uctions, press 'Alt+F'

Node Name: MyController
Node Address: 0000.0001
Target ID: 16#101A0101
Target Name: TM238LFDC24DT
Target Type: 16#1000
Target Vendor: Schneider Electric
Target Version: 2.0.30.13

Filter : Target ID
Sorting order : Name

31. Click the icon circled out or select **Online** > **Login** or press Alt+F8.



demo1
MyController (TM238LFDC24DT)
PLC Logic
Application
GVL
Library Manager
POU (PRG)
Task Configuration
MAST
Embedded Functions
IO (IO)
HSC (HSC)
PTO_PWM (PTO_PWM)
Serial Line 1
Modbus_Manager (Modbus_Manager)
Serial Line 2
SoMachine_Network_Manager (SoMachine-Net
CAN
CANopen_Optimized (CANopen Optimized)
SV680P_INT_Servo_driver(SV680P_IN

Select the network path to the controller:
Gateway-1:0000.0001

Gateway-1
[0000]
MyController [0000.0001] (active)

Node Name: MyController
Node Address: 0000.0001
Target ID: 16#101A0101
Target Name: TM238LFDC24DT
Target Type: 16#1000
Target Vendor: Schneider Electric
Target Version: 2.0.30.13

Filter : Target ID
Sorting order : Name

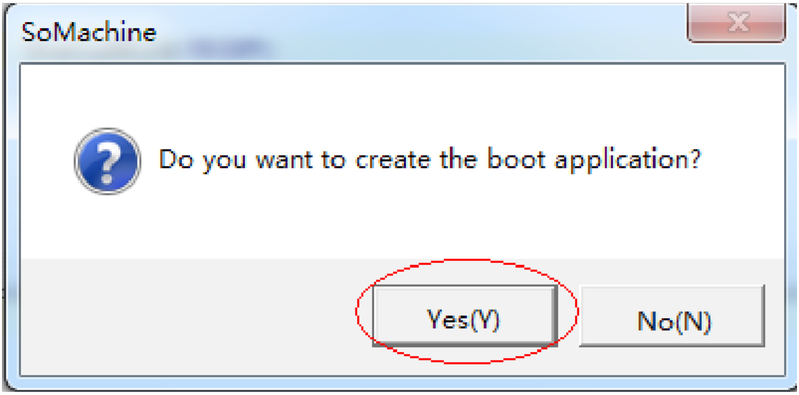
□ Don't save network path in project
☑ Secure online mode

Build
0 error(s) 0 warning(s) 9 message(s)

Precompile: OK

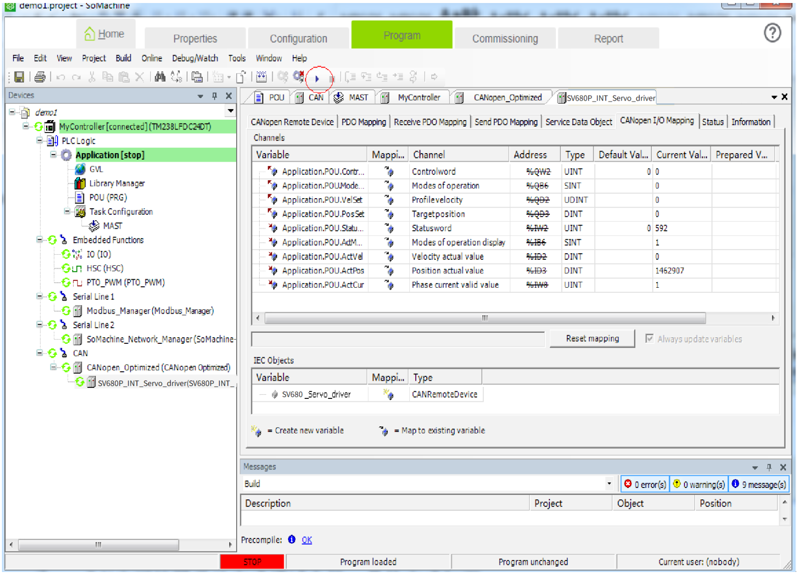
Current user: (nobody)

32. Click **Yes** in the dialog box displayed.



Do you want to create the boot application?

33. After download is done, click the ▶ circled out or click **Online** > **Start** or press F5 to start the PLC program written by the user. The motor operates in the mode defined by the user.



Channels

Variable	Mappi...	Channel	Address	Type	Default Val...	Current Val...	Prepared V...
Application.POU.Contr...		Controlword	~~%QW2~~	UINT	0	0	
Application.POU.Mode...		Modes of operation	~~%QB6~~	SINT		0	
Application.POU.VelSet		Profilevelocity	~~%QD2~~	UDINT		0	
Application.POU.PosSet		Targetposition	~~%QD3~~	DINT		0	
Application.POU.Statu...		Statusword	~~%IW2~~	UINT	0	592	
Application.POU.AdM...		Modes of operation display	~~%IB6~~	SINT		1	
Application.POU.ActVel		Velocity actual value	~~%ID2~~	DINT		0	
Application.POU.ActPos		Position actual value	~~%ID3~~	DINT		1462907	
Application.POU.ActCur		Phase current valid value	~~%IW8~~	UINT		1	

Reset mapping ☑ Always update variables

IEC Objects

Variable	Mappi...	Type
SV680 _Servo_driver		CANRemoteDevice

= Create new variable = Map to existing variable

Messages

Build | 0 error(s) | 0 warning(s) | 9 message(s)

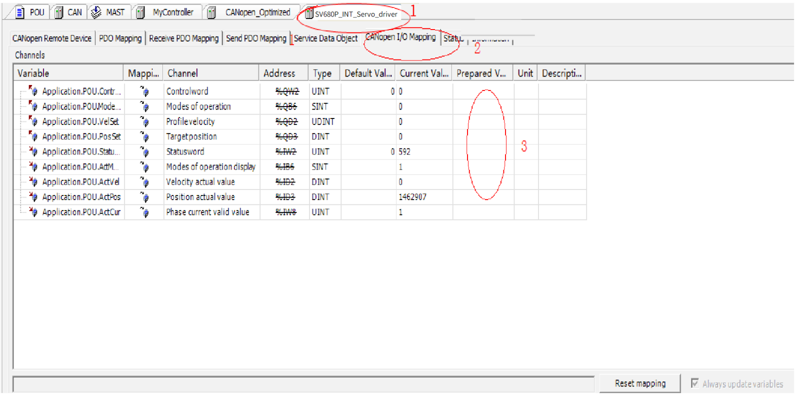
Description	Project	Object	Position

Precompile: OK

STOP | Program loaded | Program unchanged | Current user: (nobody)

34. You can also perform motor commissioning manually according to the following steps.
Select **CANopen I/O Mapping** under **SV680P_INT...** and enter the value needed in the **Prepared V...** column. Next, click **Debug/Watch** > **Forced Value** or press F7 to modify the variable manually.

35. Write 1 to 6060h, 100 to 6081h, and 10485760 (10 revolutions) to 607Ah. Write 6 (0x06), 7 (0x07), 47 (0x2f), and 63 (0x3f) to 6040h in sequence to make the motor run.



Channels

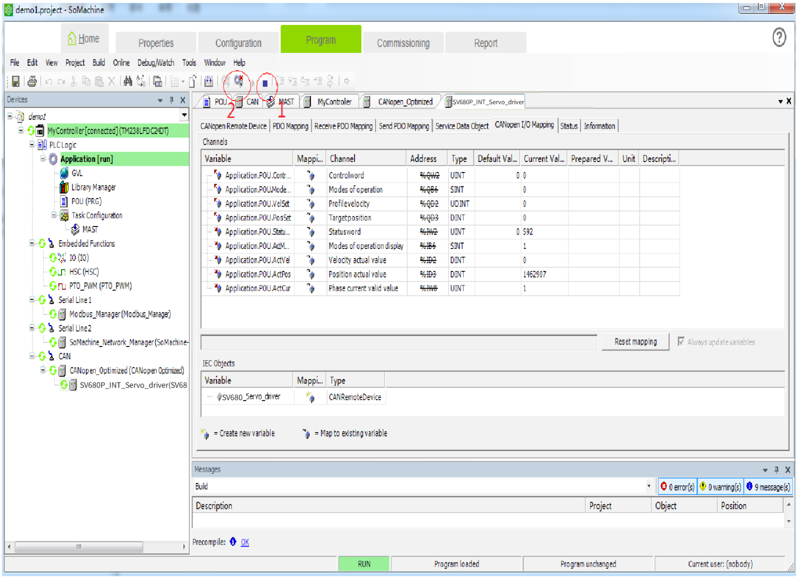
Variable	Mappi...	Channel	Address	Type	Default Val...	Current Val...	Prepared V...	Unit	Descripti...
Application.POU.Contr...		Controlword	~~%QW2~~	UINT	0	0			
Application.POU.Mode...		Modes of operation	~~%QB6~~	SINT		0			
Application.POU.VelSet		Profilevelocity	~~%QD2~~	UDINT		0			
Application.POU.PosSet		Targetposition	~~%QD3~~	DINT		0			
Application.POU.Statu...		Statusword	~~%IW2~~	UINT	0	592			
Application.POU.AdM...		Modes of operation display	~~%IB6~~	SINT		1			
Application.POU.ActVel		Velocity actual value	~~%ID2~~	DINT		0			
Application.POU.ActPos		Position actual value	~~%ID3~~	DINT		1462907			
Application.POU.ActCur		Phase current valid value	~~%IW8~~	UINT		1			

Reset mapping ☑ Always update variables

Note

- When writing multiple values for one variable, execute the "Forced value" command every time a value is written. When writing values for multiple variables, you can execute the "Forced value" command once for all after all the values are written.
- When a new position or speed reference is required, write the new reference and set 6040h to 47(0x2f) and 63(0x3f) in turn. The motor runs to the position according to the new reference regardless of whether the previous reference is executed.
- To stop the motor, set 6040h to 0.
- To terminate manual writing of values, go to the toolbar and choose Debug/Watch > Release Values, or press Alt+F7. Then, variables will be executed according to the PLC program logic instead of manually written values.

36. Execute **1** marked in the following figure, or select **Online** > **Stop** in the toolbar or press Shift + F8 to stop the PLC program. Click **2** marked in the following figure, or select **Online** > **Exit** or press Ctrl + F8 to exit from the online function.



Channels

Variable	Mappi...	Channel	Address	Type	Default Val...	Current Val...	Prepared V...	Unit	Descripti...
Application.POU.Contr...		Controlword	~~%QW2~~	UINT	0	0			
Application.POU.Mode...		Modes of operation	~~%QB6~~	SINT		0			
Application.POU.VelSet		Profilevelocity	~~%QD2~~	UDINT		0			
Application.POU.PosSet		Targetposition	~~%QD3~~	DINT		0			
Application.POU.Statu...		Statusword	~~%IW2~~	UINT	0	592			
Application.POU.ActM...		Modes of operation display	~~%IB6~~	SINT		1			
Application.POU.ActVel		Velocity actual value	~~%ID2~~	DINT		0			
Application.POU.ActPos		Position actual value	~~%ID3~~	DINT		1462907			
Application.POU.ActCur		Phase current valid value	~~%IW8~~	UINT		1			

IEC Objects

Variable	Mappi...	Type
SV680_Servo_driver		CANRemoteDevice

Description	Project	Object	Position

5.2.2 Connecting SV680P-INT to Beckoff CANopen Master

Assign PDO according to *"Table 5–2 " on page 98* in the position control mode.

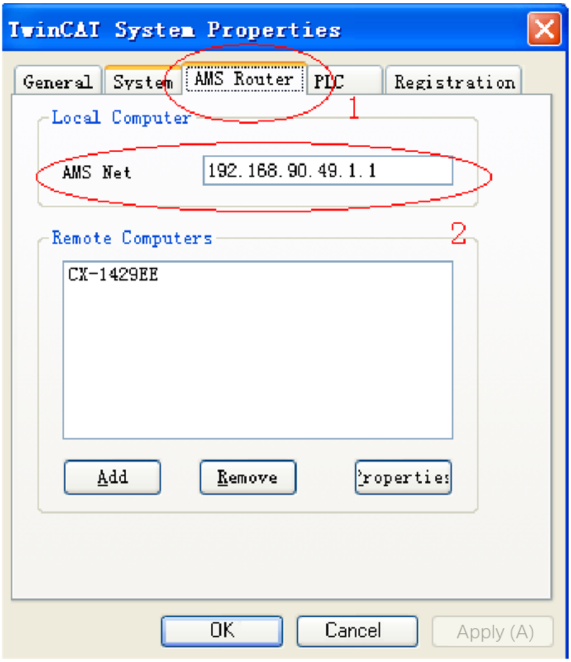
1. Configuring PDO mapping is complex on a Beckoff master node. Therefore, before connecting the network, manually configure the PDO mapping. Based on the

following table and the appendix, change the mapping by modifying parameters. The parameters to be modified are as follows:

Table 5–2 Example of PDO mapping of Beckhoff master

Parameter	Object	Mapping Object	Input
H2d.32	1600.00h	Number of mapped objects in RPDO1	2
H2d.33	1600.01h	6040.00h	60400010h
H2d.35	1600.02h	6060.00h	60600008h
H2d.49	1601.00h	Number of mapped objects in RPDO2	2
H2d.50	1601.01h	6081.00h	60810020h
H2d.52	1601.02h	607A.00h	607A0020h
H2E.20	1A00.00h	Number of mapped objects in TPDO1	2
H2E.21	1A00.01h	6041.00h	60410010h
H2E.23	1A00.02h	6061.00h	60610008h
H2E.37	1A01.00h	Number of mapped objects in TPDO2	2
H2E.38	1A01.01h	606C.00h	606C0020h
H2E.40	1A01.02h	6064.00h	60640020h
H2E.54	1A02.00h	Number of mapped objects in TPDO3	1
H2E.55	1A02.01h	200B.19h	200B1910h
H2E.57	1A02.02h	-	0

2. Connect Beckoff CX9020, as a master node, to the CANopen module of EL6751 and perform the test. Ensure that the IP address of CX9020 is in the same network segment as the IP address of the PC and the first four bytes of AMS Net (**Properties** > **AMS Router** > **AMS Net**) of Beckoff TwinCAT software are the same as the IP address of the PC.



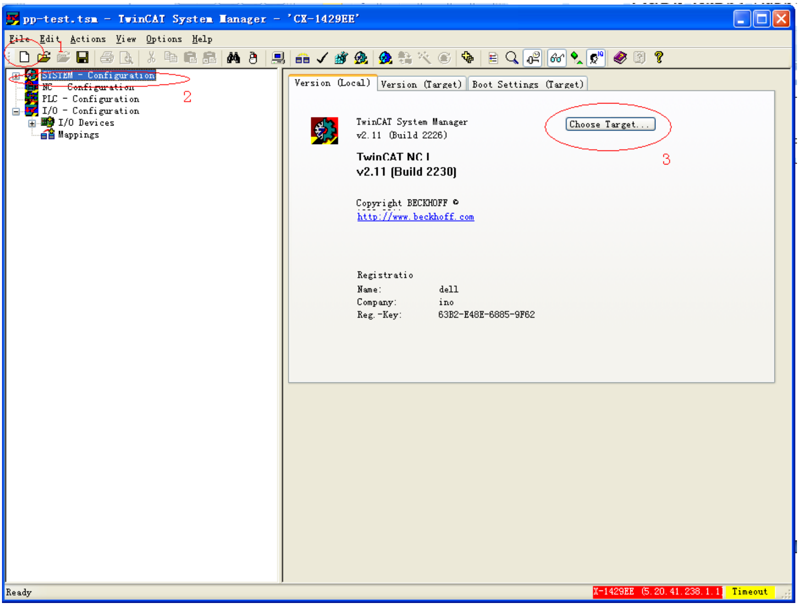
Local Computer

AMS Net 192.168.90.49.1.1

Remote Computers

CX-1429EE

3. Open TwinCAT System Manager and create an empty project. Click **SYSTEM - Configuration** on the left and click **Choose Target...** on the right.



TwinCAT System Manager
v2.11 (Build 2226)

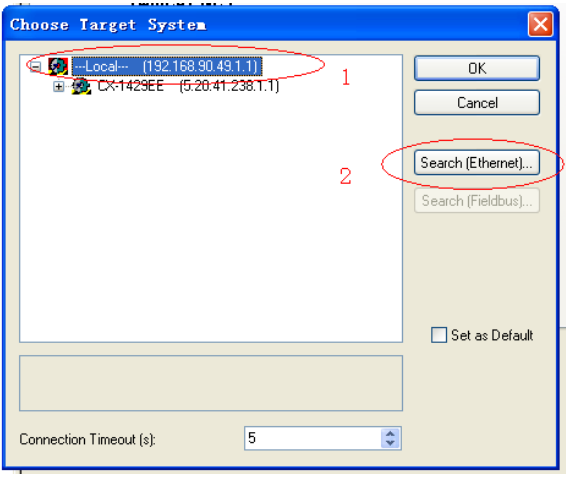
TwinCAT NC I
v2.11 (Build 2230)

Copyright BECKHOFF ©
http://www.beckhoff.com

Registratio
Name: dell
Company: ino
Reg.-Key: 63B2-E48E-6885-9F62

X-1429EE (5.20.41.238.1.1) Timeout

4. In the dialog box that is displayed, select **...local...** and click **Search (Ethernet)**.



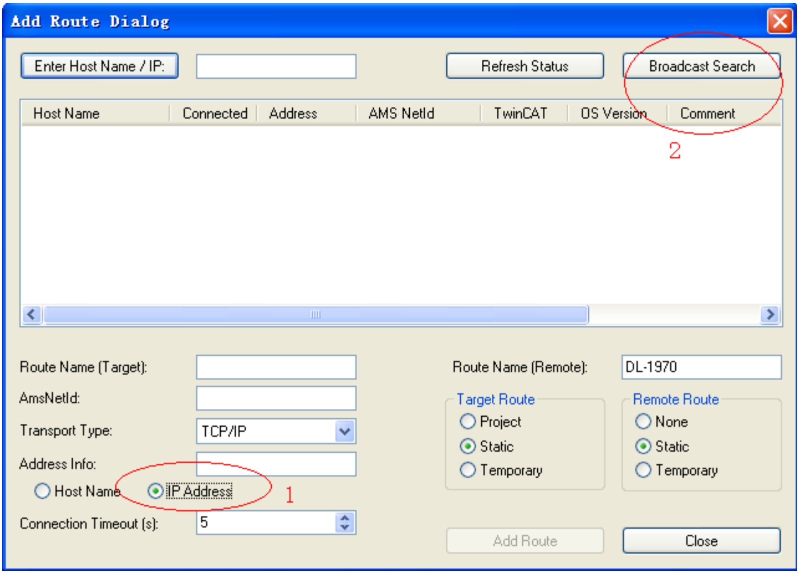
---Local--- (192.168.90.49.1.1)

CX-1429EE (5.20.41.238.1.1)

□ Set as Default

Connection Timeout (s): 5

5. Select the **IP Address** as indicated by **1** and click **Broadcast Search**.



Route Name (Target):

AmsNetId:

Transport Type: TCP/IP

Address Info:

○ Host Name ● IP Address

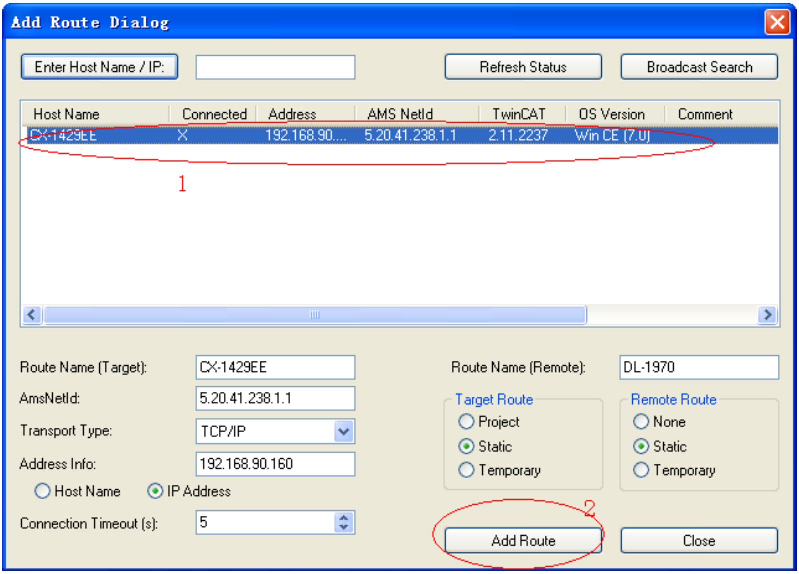
Connection Timeout (s): 5

Route Name (Remote): DL-1970

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

6. The master is displayed. Select the master and click **Add Route**.



Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX-1429EE	X	192.168.90....	5.20.41.238.1.1	2.11.2237	Win CE (7.0)	

Route Name (Target): CX-1429EE

AmsNetId: 5.20.41.238.1.1

Transport Type: TCP/IP

Address Info: 192.168.90.160

○ Host Name ● IP Address

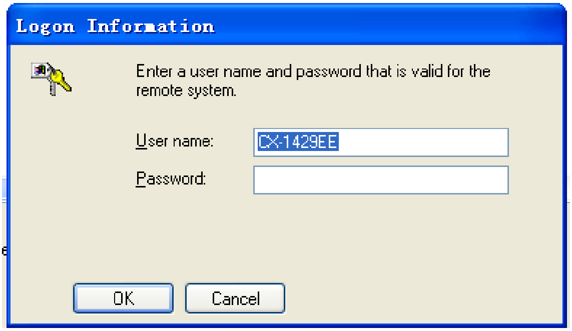
Connection Timeout (s): 5

Route Name (Remote): DL-1970

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

7. In the dialog box displayed, the account is the same with the **Host Name** and the password is empty. Click **OK**.

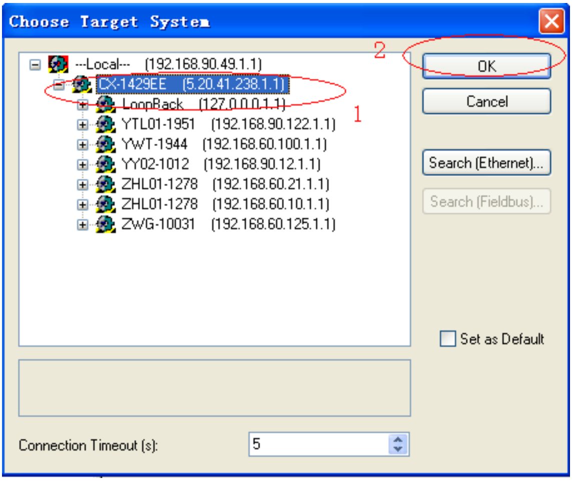


Enter a user name and password that is valid for the remote system.

User name: CX-1429EE

Password:

8. Click **Close** in the interface shown in Step 6, then you can click **+** in the **Choose Target System** dialog box to select the master. Finally, click **OK**.



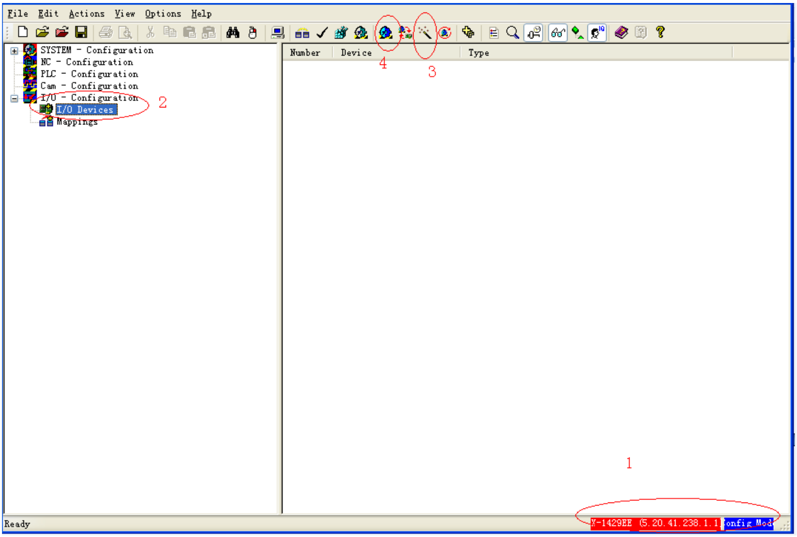
Choose Target System

- ---Local--- (192.168.90.49.1.1)
  - CX-1429EE (5.20.41.238.1.1)
    - LoopBack (127.0.0.0.1.1)
    - YTL01-1951 (192.168.90.122.1.1)
    - YWT-1944 (192.168.60.100.1.1)
    - YY02-1012 (192.168.90.12.1.1)
    - ZHL01-1278 (192.168.60.21.1.1)
    - ZHL01-1278 (192.168.60.10.1.1)
    - ZWG-10031 (192.168.60.125.1.1)

Connection Timeout (s): 5

□ Set as Default

9. The master (in red background) can be seen in the lower right corner of the window, which is in the configuration status (in blue background). If the master is in the operating status (in green background), click the icon indicated by **4** to switch to the configuration status, and then proceed to the next step.
   Select **I/O Devices** on the left and click the icon indicated by **3** or right-click **I/O Devices** and select **Scan Devices**.

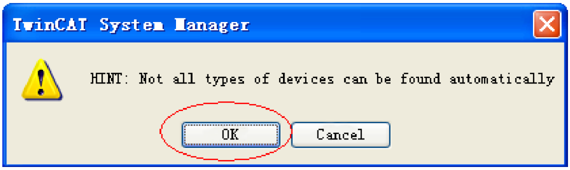


- SYSTEM - Configuration
- NC - Configuration
- PLC - Configuration
- Cam - Configuration
- I/O - Configuration
  - I/O Devices
  - Mappings

Ready

X-1429EE (5. 20. 41. 238. 1. 1) onfig Mod

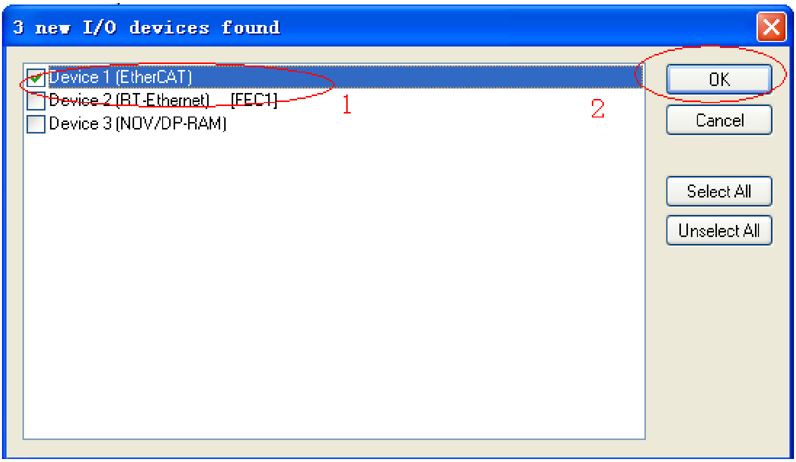
10. Click **OK** in the warning dialog box displayed.



TwinCAT System Manager

HINT: Not all types of devices can be found automatically

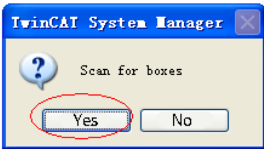
11. Check **Device EtherCAT** and click **OK** in the dialog box displayed.



3 new I/O devices found

☑ Device 1 (EtherCAT)
□ Device 2 (RT-Ethernet) [FEC1]
□ Device 3 (NOV/DP-RAM)

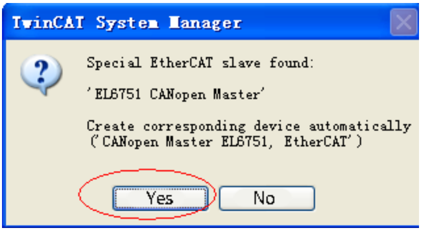
12. Click **Yes** in the dialog box asking whether to scan for boxes.



TwinCAT System Manager

Scan for boxes

13. Click **Yes** in the dialog box asking whether to create 6751 master.



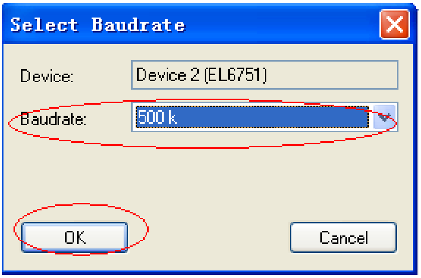
TwinCAT System Manager

Special EtherCAT slave found:

'EL6751 CANopen Master'

Create corresponding device automatically
('CANopen Master EL6751, EtherCAT')

14. Select the baud rate (defaulted to 500 kbps) and click **OK**. The master starts device searching, which may take a while.

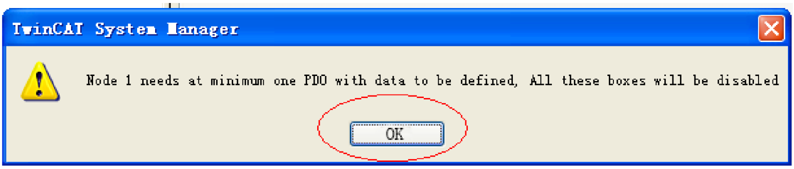


Select Baudrate

Device: Device 2 (EL6751)

Baudrate: 500 k

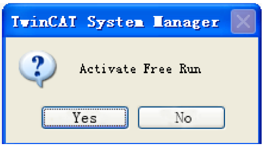
15. After device searching is done, click **OK** in the warning dialog box displayed.



TwinCAT System Manager

Node 1 needs at minimum one PDO with data to be defined, All these boxes will be disabled

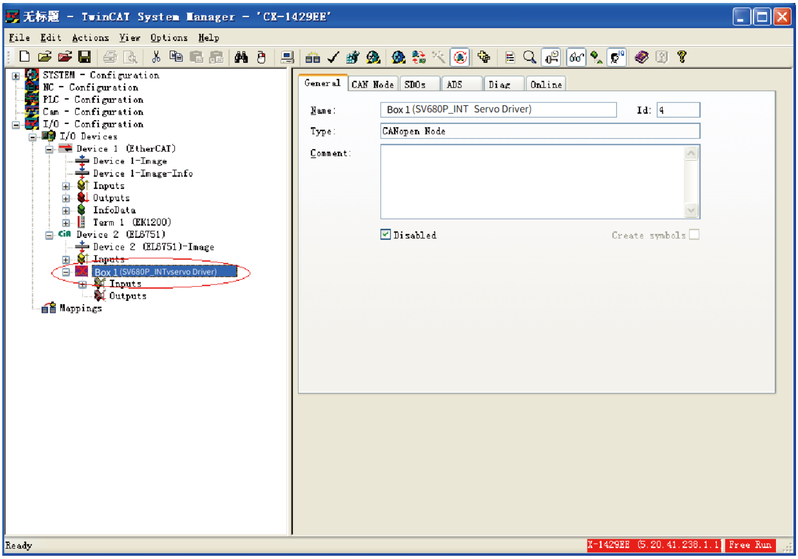
16. Click **Yes** in the dialog box asking whether to activate free run.



TwinCAT System Manager

Activate Free Run

17. The Box of SV680P-INT series servo drive is now displayed on the left. Right-click to insert three TPDOs and 2 RPDOs. Right click **Disabled** to **uncheck it.**



无标题 - TwinCAT System Manager - 'CX-1429EE'

Name: Box 1 (SV680P_INT Servo Driver) Id: 4

Type: CANopen Node

Comment:

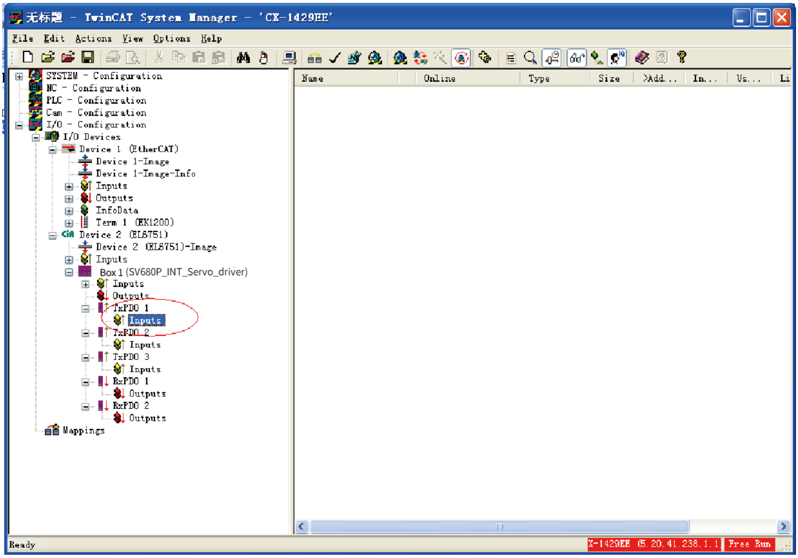
☑ Disabled Create symbols □

Ready X-1429EE (5.20.41.238.1.1) Free Run

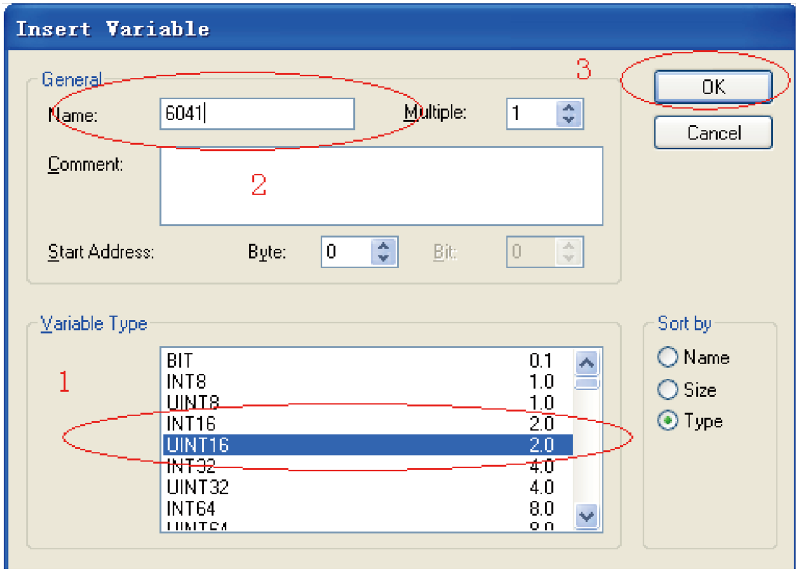
Note

Only servo drives equipped with termination resistors can be scanned by the master.

18. The following figure shows the result after the previous operation is complete. Choose **TPDO1** > **Inputs**, right-click, and choose **Insert Variable**.



19. Map different variables in each PDO according to *"Table 5–2 Example of PDO mapping of Beckhoff master" on page 98*. TPDO1 maps 6041.00h and 6061.00h. To insert the first variable 6041h, select **UINT16** in the **Variable Type** first, and then enter a proper name in the field **Name** and click **OK**.



General

Name: 6041

Multiple: 1

Comment:

Start Address: Byte: 0 Bit: 0

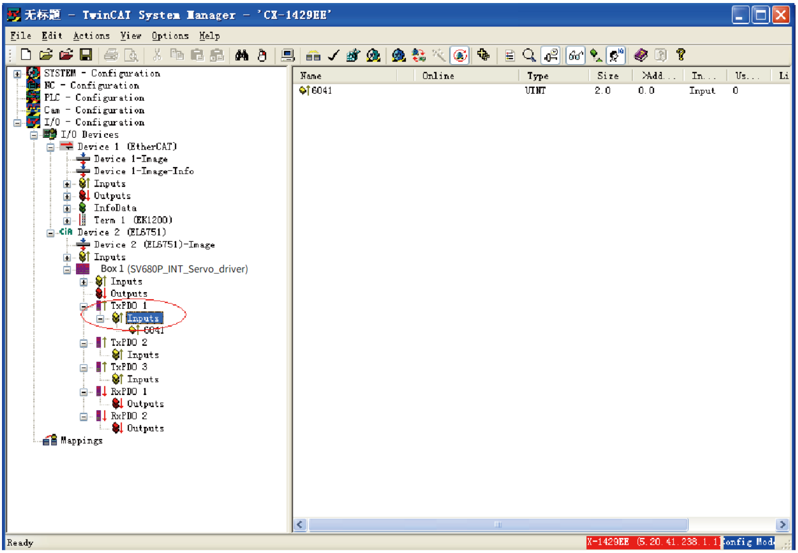
Variable Type

BIT	0.1
INT8	1.0
UINT8	1.0
INT16	2.0
UINT16	2.0
INT32	4.0
UINT32	4.0
INT64	8.0
UINT64	8.0

Sort by

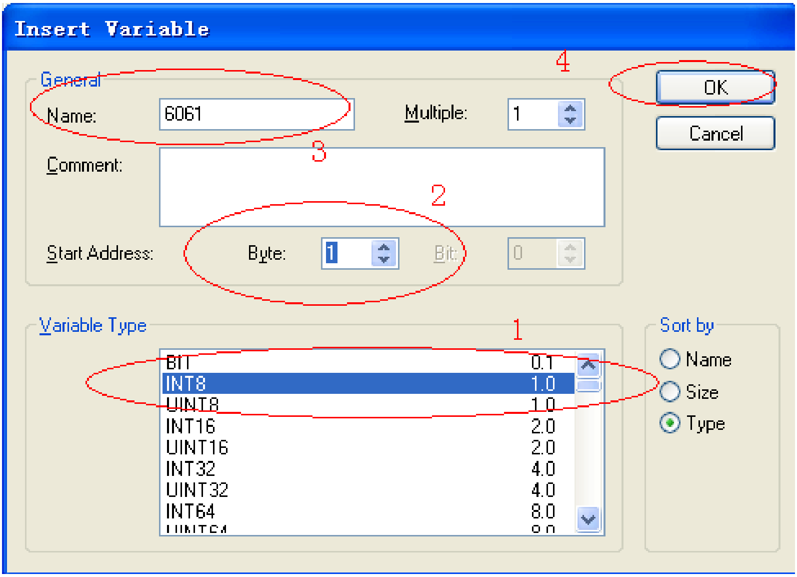
○ Name
○ Size
● Type

20. Now 6041h has been added to TPDO1. Select **Inputs** again, right-click, choose **Insert Variable**, and insert the second variable.



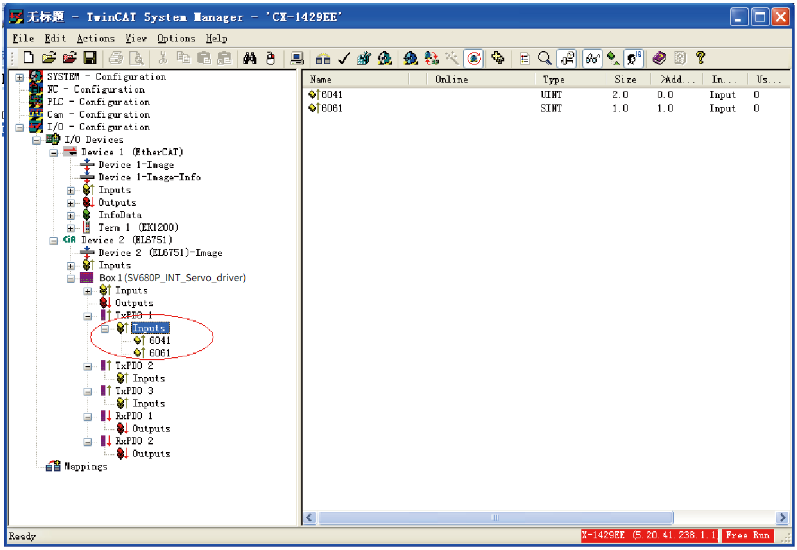
Name	Online	Type	Size	>Add...	In...	Us...	Li
6041		UINT	2.0	0.0	Input	0	

21. For the inserted variable 6061h, select **INT8** (the object dictionary can be queried) for **Variable Type**, enter a large value for **Byte** of **Start Address** to prevent 6061h from being inserted in front of 6041h, enter a proper name, Click **OK**.



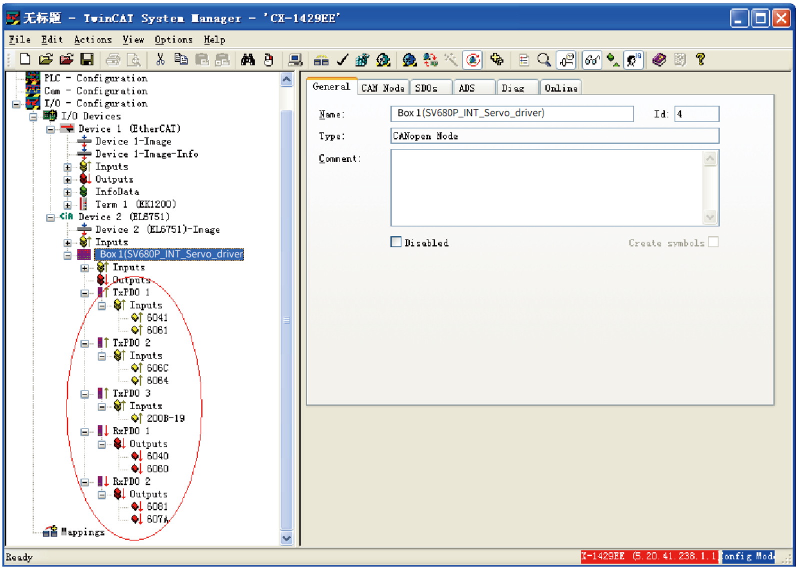
22. You can see that two objects are added to TPDO1. Note that the sequence of the two variables must be the same as that in *"Table 5–2 Example of PDO mapping of Beckhoff master" on page 98*. Otherwise, the second variable must be deleted and inserted again and a large value must be entered in **2** marked in the figure in Step 21.

    After making sure that the variable sequence is correct, choose **TPDO1** > **Inputs**, right-click, and choose **Recalc Address** to allocate addresses. This step must be performed. Otherwise, addresses will be in mess.



Name	Online	Type	Size	>Add...	In...	Us...
6041		UINT	2.0	0.0	Input	0
6061		SINT	1.0	1.0	Input	0

23. Repeat steps 18 to 22 for other PDOs. Add corresponding mapping variables according to *"Table 5–2 Example of PDO mapping of Beckhoff master" on page 98*. The interface after variables are added is shown below.

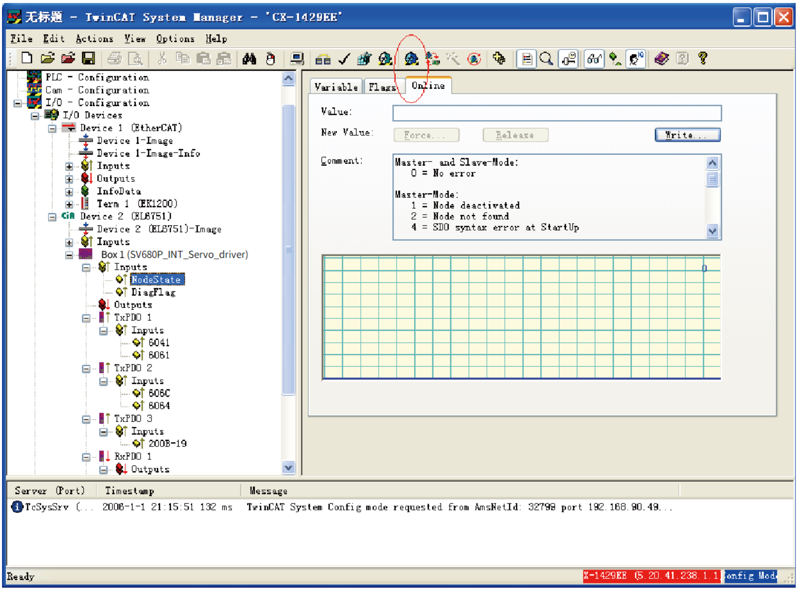


Name: Box 1 (SV680P_INT_Servo_driver)   Id: 4

Type: CANopen Node

□ Disabled

24. Click the icon circled out in the following figure or press Shift + F4.

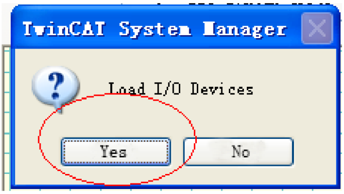


Master- and Slave-Mode:
 0 = No error

Master-Mode:
 1 = Node deactivated
 2 = Node not found
 4 = SDO syntax error at StartUp

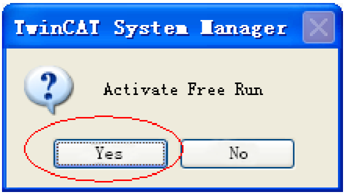
TcSysSrv (... 2006-1-1 21:15:51 132 ms TwinCAT System Config mode requested from AmsNetId: 32798 port 192.168.90.49...

25. Click **Yes** in the following dialog box.



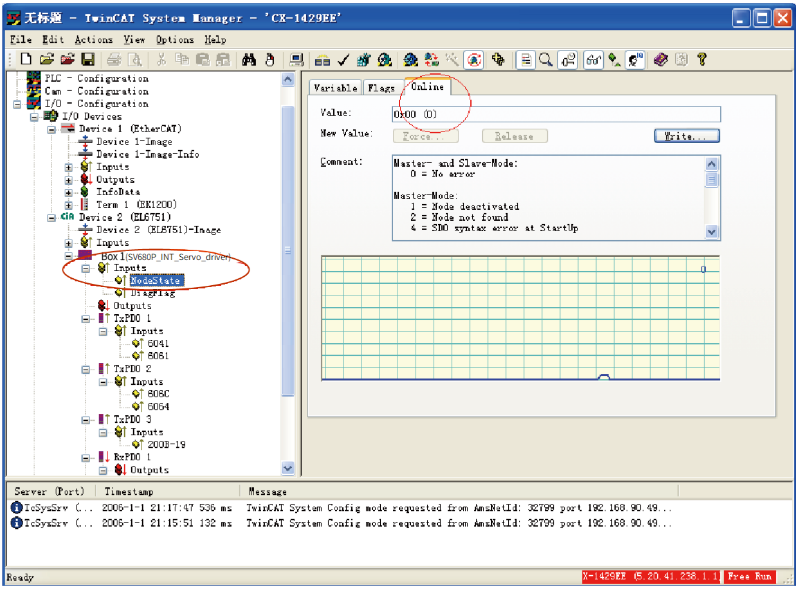
Load I/O Devices

26. Click **Yes** in the dialog box asking whether to activate free run.



Activate Free Run

27. Select the Box of SV680P-INT and select **Inputs** > **NodeState**. The node state in **Online** is 0, indicating the node is in a normal state.



Value: 0x00 (0)

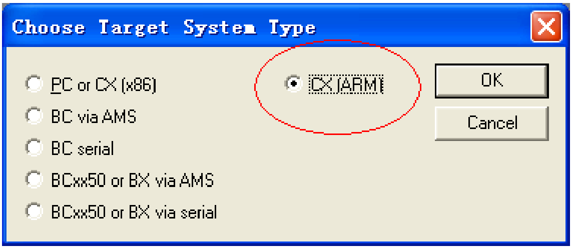
New Value:

Comment:

```
Master- and Slave-Mode:
   0 = No error

Master-Mode:
   1 = Node deactivated
   2 = Node not found
   4 = SDO syntax error at StartUp
```

```
Server (Port)   Timestamp                   Message
TcSysSrv (...   2006-1-1 21:17:47 536 ms    TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
TcSysSrv (...   2006-1-1 21:15:51 132 ms    TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
```

28. Open TwinCAT PLC Control, create a new project and select **CX (ARM)** in the dialog box displayed.



○ PC or CX (x86)

○ BC via AMS

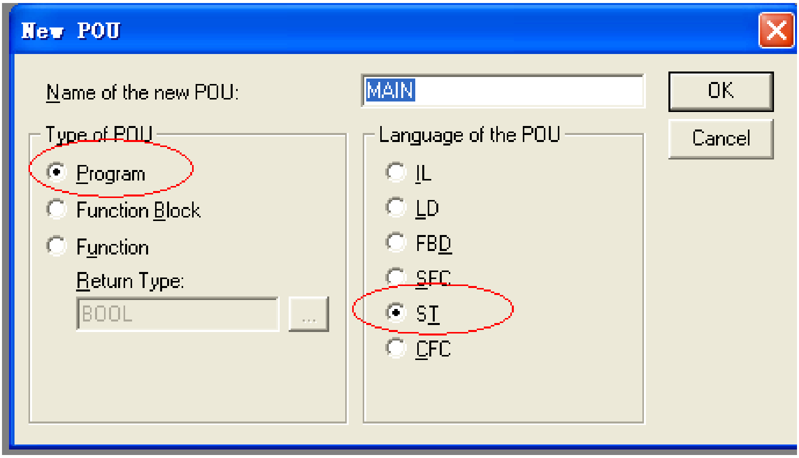
○ BC serial

○ BCxx50 or BX via AMS

○ BCxx50 or BX via serial

● CX (ARM)

29. In the dialog box that is displayed, select the following options:



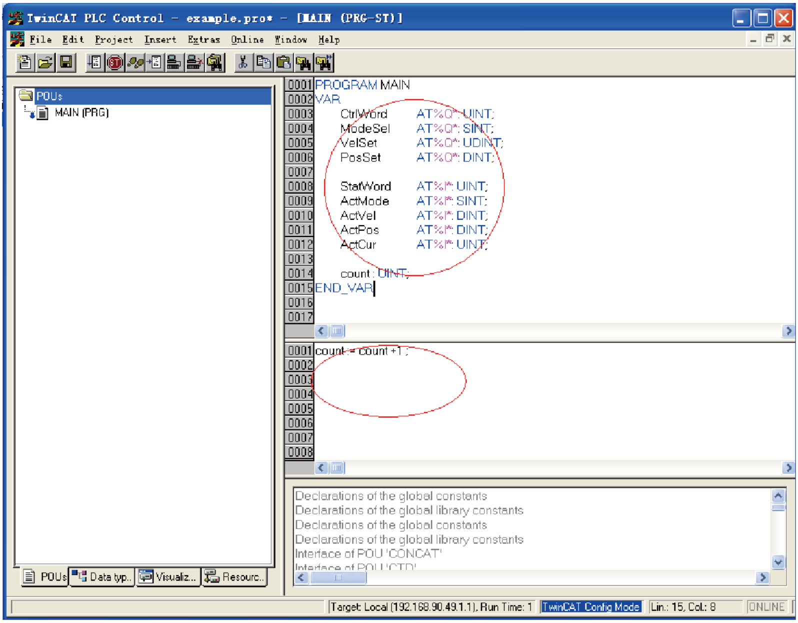
Name of the new POU: MAIN

Type of POU
- ● Program
- ○ Function Block
- ○ Function

Return Type: BOOL

Language of the POU
- ○ IL
- ○ LD
- ○ FBD
- ○ SFC
- ● ST
- ○ CFC

30. Enter corresponding variable definition and the PLC logic.


TwinCAT PLC Control - example.pro* - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data typ...
Visualiz...
Resourc...

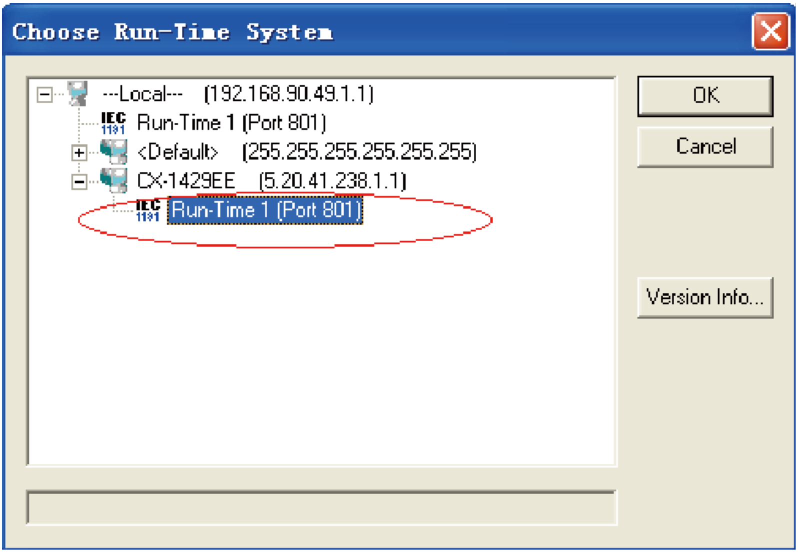
```
PROGRAM MAIN
VAR
	CtrlWord	AT%Q*: UINT;
	ModeSel	AT%Q*: SINT;
	VelSet	AT%Q*: UDINT;
	PosSet	AT%Q*: DINT;

	StatWord	AT%I*: UINT;
	ActMode	AT%I*: SINT;
	ActVel	AT%I*: DINT;
	ActPos	AT%I*: DINT;
	ActCur	AT%I*: UINT;

	count : UINT;
END_VAR
```

```
count := count +1 ;
```

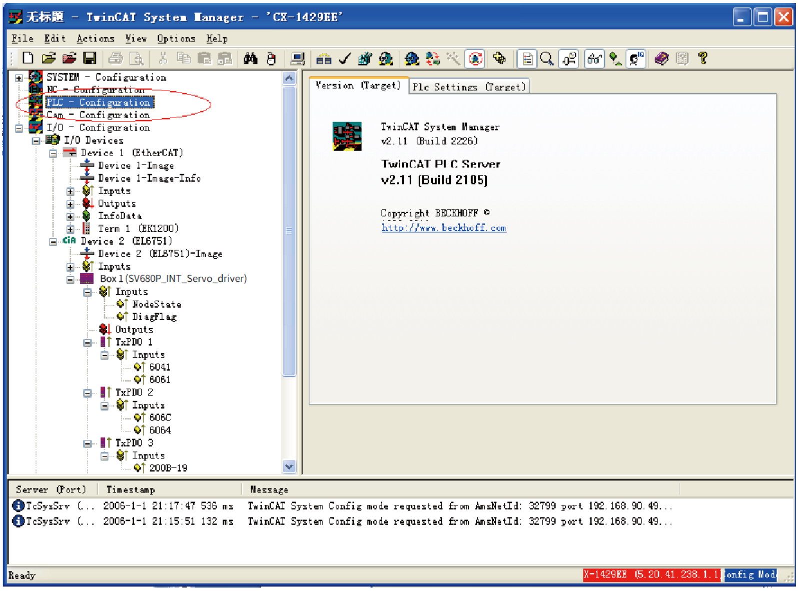
```
Declarations of the global constants
Declarations of the global library constants
Declarations of the global constants
Declarations of the global library constants
Interface of POU 'CONCAT'
Interface of POU 'CTD'
```

Target: Local (192.168.90.49.1.1), Run Time: 1 | TwinCAT Config Mode | Lin.: 15, Col.: 8 | ONLINE

31. In the toolbar, select **Online** > **Choose Run-Time System**. Select the corresponding master port in the dialog box displayed and click **OK**.


Choose Run-Time System
---Local--- (192.168.90.49.1.1)
IEC 1131 Run-Time 1 (Port 801)
<Default> (255.255.255.255.255.255)
CX-1429EE (5.20.41.238.1.1)
IEC 1131 Run-Time 1 (Port 801)
OK
Cancel
Version Info...

32. In TwinCAT System Manager, select **PLC** → **Configuration** on the left, and then right-click to display the short-cut menu. Select **Append PLC Project...** in the short-cut menu to select the PLC program (.tpy). created.



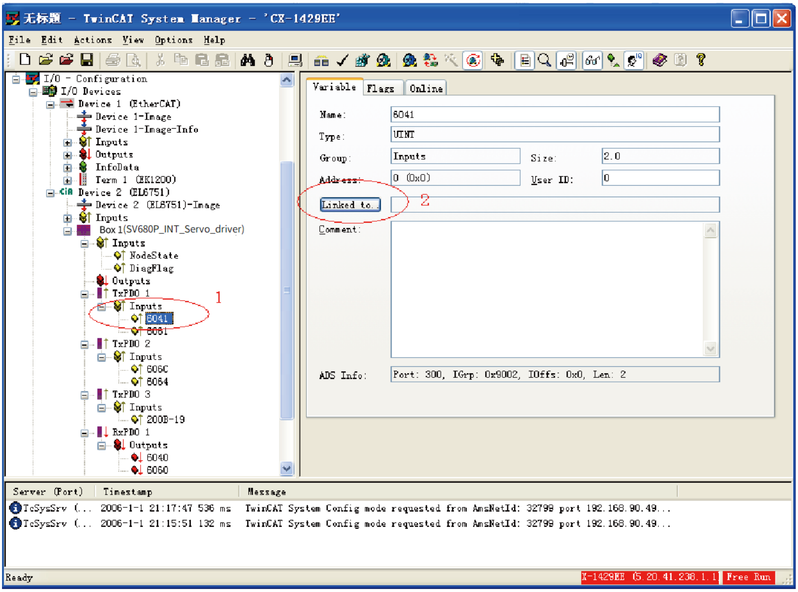
TwinCAT System Manager
v2.11 (Build 2226)

TwinCAT PLC Server
v2.11 (Build 2105)

Copyright BECKHOFF ©
http://www.beckhoff.com

```
Server (Port)   Timestamp                   Message
TcSysSrv (...   2006-1-1 21:17:47 536 ms    TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
TcSysSrv (...   2006-1-1 21:15:51 132 ms    TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
```

33. After the PLC program is added, select the PDO variable and click **Linked to** or double-click the variable to link the variable to the PLC program.



Name: 6041

Type: UINT

Group: Inputs   Size: 2.0

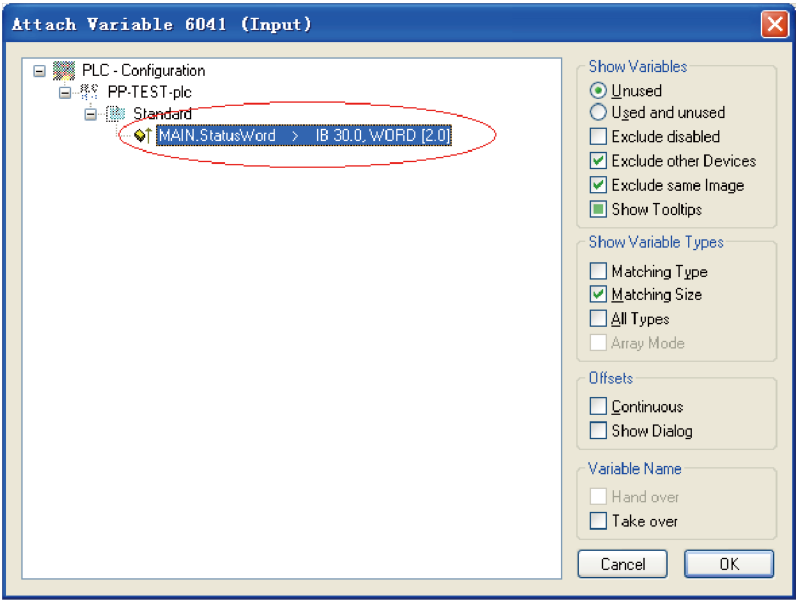
Address: 0 (0x0)   User ID: 0

Comment:

ADS Info: Port: 300, IGrp: 0x9002, IOffs: 0x0, Len: 2

```
Server (Port)   Timestamp                  Message
TcSysSrv (...   2006-1-1 21:17:47 536 ms   TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
TcSysSrv (...   2006-1-1 21:15:51 132 ms   TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
```

34. Select the corresponding PLC variable and click **OK**.



Show Variables
● Unused
○ Used and unused
□ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
▣ Show Tooltips

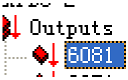
Show Variable Types
□ Matching Type
☑ Matching Size
□ All Types
□ Array Mode

Offsets
□ Continuous
□ Show Dialog

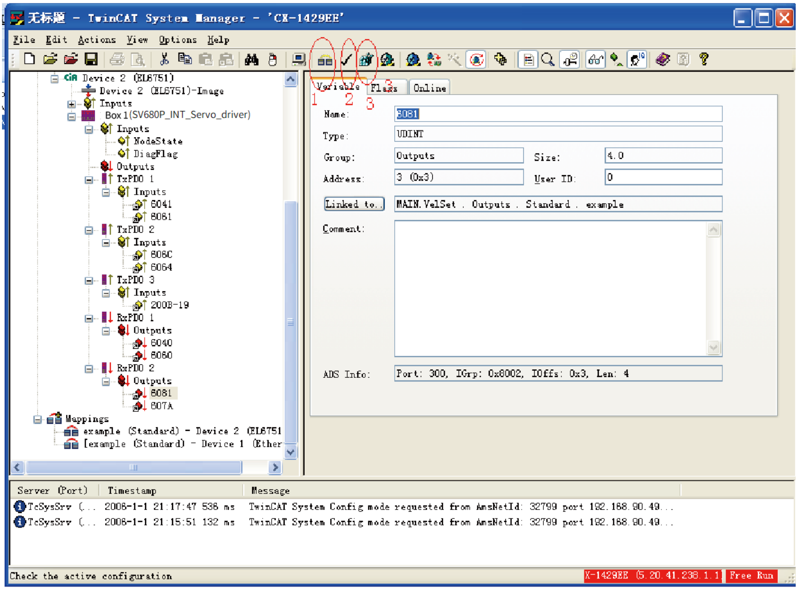
Variable Name
□ Hand over
□ Take over

35. After the variable is linked, a small arrow pointing upper right appears at the bottom left of the variable name icon. As shown in the following figure, the name of

the variable not linked is displayed on the left and the name of the linked variable is displayed on the right.



36. Click **Generate mapping**, **Check Configuration**, and **Activate Configuration** in sequence, as circled out by **1**, **2**, and **3** in the following figure.



Name: 6081

Type: UDINT

Group: Outputs Size: 4.0

Address: 3 (0x3) User ID: 0

Linked to..: MAIN.VelSet . Outputs . Standard . example

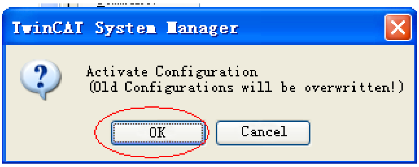
Comment:

ADS Info: Port: 300, IGrp: 0xF002, IOffs: 0x3, Len: 4

```
Server (Port)   Timestamp                  Message
TcSysSrv (...   2006-1-1 21:17:47 536 ms   TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
TcSysSrv (...   2006-1-1 21:15:51 132 ms   TwinCAT System Config mode requested from AmsNetId: 32799 port 192.168.90.49...
```

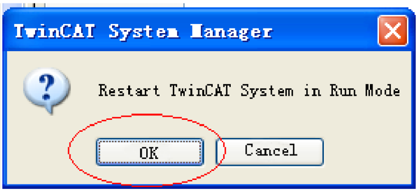
Check the active configuration   X-1429EE (5.20.41.238.1.1) Free Run

37. Click **OK** to activate configuration.



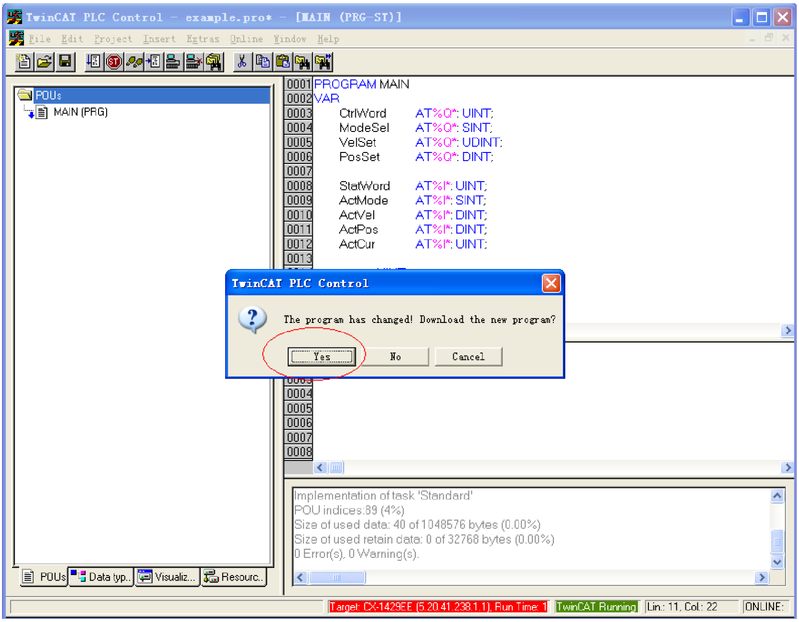
Activate Configuration
(Old Configurations will be overwritten!)

38. Click **OK** to restart TwinCAT system with the run mode.



Restart TwinCAT System in Run Mode

39. Open the project created by TwinCAT PLC Control software before, and click **Online** > **Login** or press F11 to display the dialog box asking whether to download the new program.


TwinCAT PLC Control - example.pro* - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data typ..
Visualiz...
Resourc..
Yes
No
Cancel

```
PROGRAM MAIN
VAR
    CtrlWord     AT%Q*: UINT;
    ModeSel      AT%Q*: SINT;
    VelSet       AT%Q*: UDINT;
    PosSet       AT%Q*: DINT;

    StatWord     AT%I*: UINT;
    ActMode      AT%I*: SINT;
    ActVel       AT%I*: DINT;
    ActPos       AT%I*: DINT;
    ActCur       AT%I*: UINT;
```

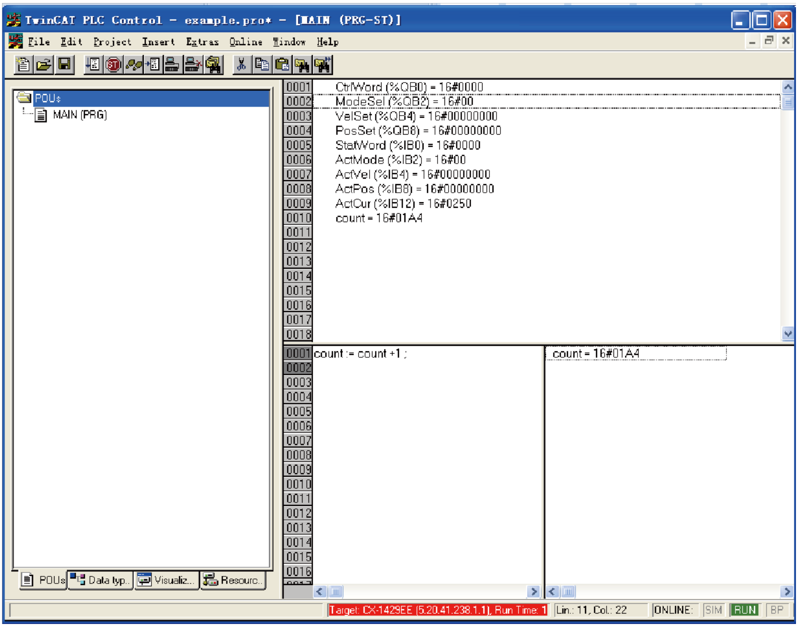
TwinCAT PLC Control

The program has changed! Download the new program?

```
Implementation of task 'Standard'
POU indices:89 (4%)
Size of used data: 40 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 0 Warning(s).
```

Target: CX-1429EE (5.20.41.238.1.1), Run Time: 1 | TwinCAT Running | Lin.: 11, Col.: 22 | ONLINE:

40. Select **Online** > **Run** or press F5 to run the user PLC program.

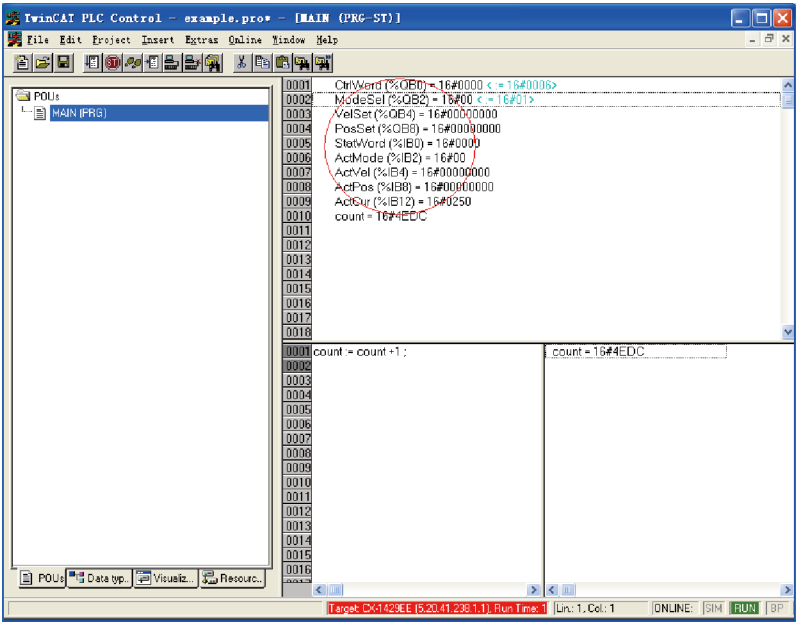

TwinCAT PLC Control - example.pro* - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data typ...
Visualiz...
Resourc...
Target: CX-1429EE (5.20.41.238.1.1), Run Time: 1
Lin.: 11, Col.: 22
ONLINE:
SIM
RUN
BP

```
CtrlWord (%QB0) = 16#0000
ModeSel (%QB2) = 16#00
VelSet (%QB4) = 16#00000000
PosSet (%QB8) = 16#00000000
StatWord (%IB0) = 16#0000
ActMode (%IB2) = 16#00
ActVel (%IB4) = 16#00000000
ActPos (%IB8) = 16#00000000
ActCur (%IB12) = 16#0250
count = 16#01A4
```

```
count := count +1 ;
```

```
count = 16#01A4
```

41. You can perform write commissioning forcibly through the manual mode. The commissioning method is similar to that of the Schneider master.
    Double-click variables circled out in the following figure and enter values.

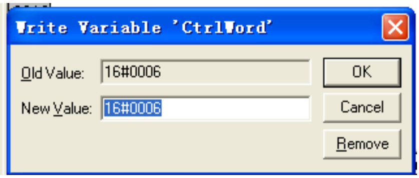

TwinCAT PLC Control - example.pro* - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data typ...
Visualiz...
Resourc...
Target: CX-1429EE (5.20.41.238.1.1), Run Time: 1
Lin.: 1, Col.: 1
ONLINE:
SIM
RUN
BP

```
CtrlWord (%QB0) = 16#0000 <:= 16#0006>
ModeSel (%QB2) = 16#00 <:= 16#01>
VelSet (%QB4) = 16#00000000
PosSet (%QB8) = 16#00000000
StatWord (%IB0) = 16#0000
ActMode (%IB2) = 16#00
ActVel (%IB4) = 16#00000000
ActPos (%IB8) = 16#00000000
ActCur (%IB12) = 16#0250
count = 16#4EDC
```

```
count := count +1 ;
```

```
count = 16#4EDC
```

42. Enter the value and click **OK**.



Write Variable 'CtrlWord'

Old Value: 16#0006

New Value: 16#0006

The value entered is displayed in the square brackets behind the original variable. Click **Online** → **Forced Value** or press F7 to write the value forcibly.

Write 1 to 6060h, 100 to 6081h, and 10485760 (10 revolutions) to 607Ah. Write 6 (0x06), 7 (0x07), 47 (0x2f), and 63 (0x3f) to 6040h in sequence to make the motor run.

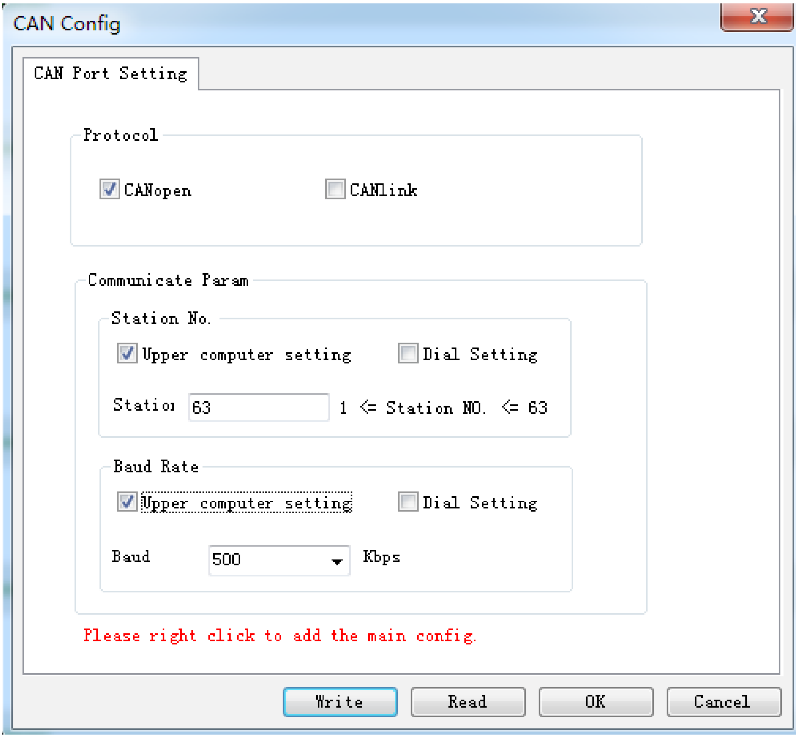
Note

- When writing multiple values for one variable, execute the "Forced value" command every time a value is written. When writing values for multiple variables, you can execute the "Forced value" command once for all after all the values are written.
- When a new position or speed reference is required, write the new reference and set 6040h to 47(0x2f) and 63(0x3f) in turn. The motor runs to the position according to the new reference regardless of whether the previous reference is executed.
- To stop the motor, set 6040h to 0.
- To terminate manual writing of values, go to the toolbar and choose Online > Release Force, or press Shift+F7. Then, variables will be executed according to the PLC program logic instead of manually written values.

43. In the toolbar, choose **Online** > **Stop** to stop executing the PLC program. Choose **Online** > **Logout** to continue editing the PLC program or exit.

5.2.3 Connecting SV680P-INT to Inovance H3U CANopen Master

1. Open AutoShop, double-click "CAN" in Communication Port of the project management interface or right-click "Open" to pop up the "CAN Config" window. Select the CANopen master as the protocol and set **Station No.** and **Baud Rate** of the master.



CAN Config

CAN Port Setting

Protocol

☑ CANopen ☐ CANlink

Communicate Param

Station No.

☑ Upper computer setting ☐ Dial Setting

Statio: 63 1 <= Station NO. <= 63

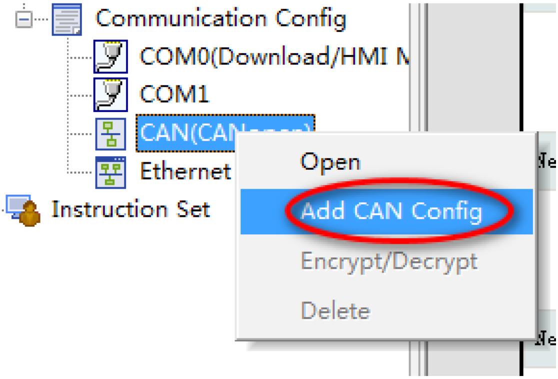
Baud Rate

☑ Upper computer setting ☐ Dial Setting

Baud 500 Kbps

Please right click to add the main config.

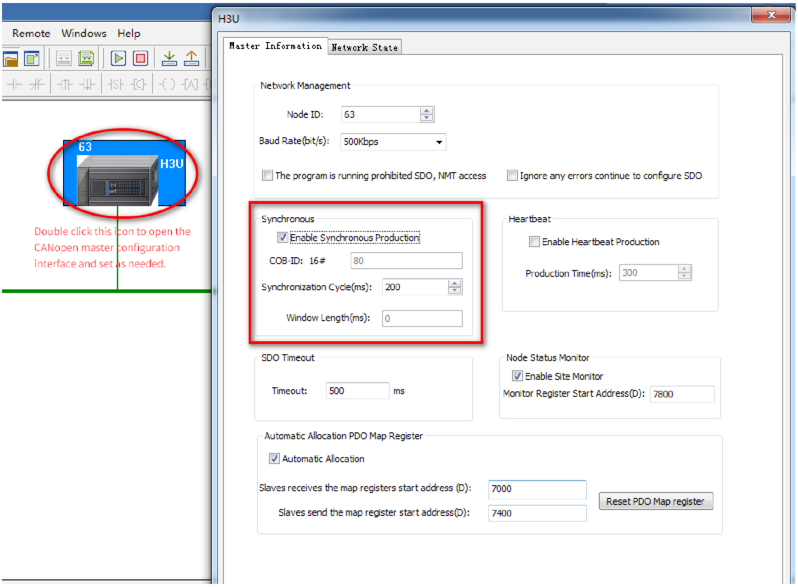
2. Right-click **CAN (CANopen)** and select **Add CAN Config** in the short-cut menu.



3. Double click CANopen Config.

You can see the H3U master icon in the CANopen configuration interface. Double-click this icon to open the master configuration interface, in which you can set parameters such as synchronization and heartbeat.

H3U axis-control commands control the servo drive through PDO communication. The PDO adopts synchronization mode by default when the drive is working with an H3U master. Therefore, you need to check **Enable Synchronous Production** in this interface and set the synchronization period (15ms for 8 axes generally) as needed. For other servo drive models, this option also needs to be checked if the PDO also adopts synchronization mode.



Double click this icon to open the CANopen master configuration interface and set as needed.

Network Management

Node ID: 63

Baud Rate(bit/s): 500Kbps

□ The program is running prohibited SDO, NMT access

□ Ignore any errors continue to configure SDO

Synchronous

☑ Enable Synchronous Production

COB-ID: 16# 80

Synchronization Cycle(ms): 200

Window Length(ms): 0

Heartbeat

□ Enable Heartbeat Production

Production Time(ms): 300

SDO Timeout

Timeout: 500 ms

Node Status Monitor

☑ Enable Site Monitor

Monitor Register Start Address(D): 7800

Automatic Allocation PDO Map Register

☑ Automatic Allocation

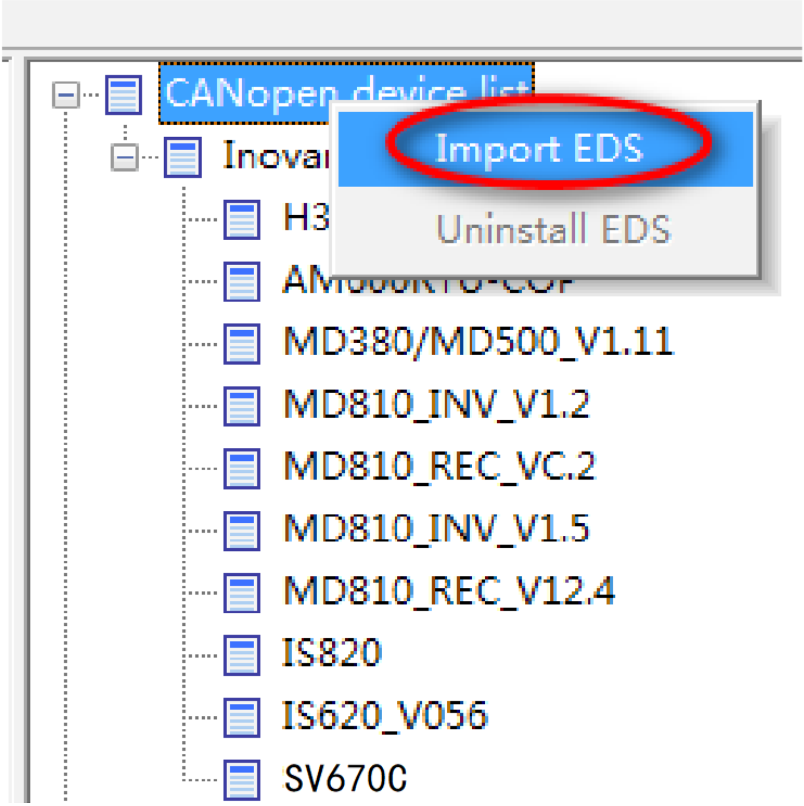
Slaves receives the map registers start address (D): 7000

Slaves send the map register start address(D): 7400

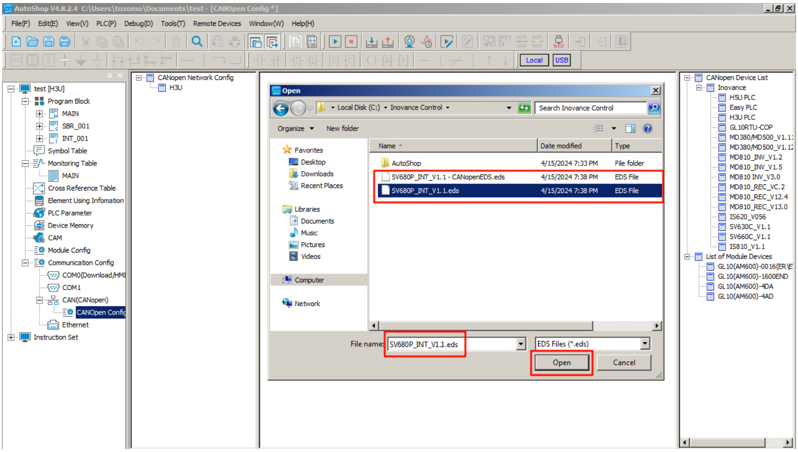
Reset PDO Map register

4. If the EDS files needed is not in the CANopen device list, add the device EDS file.

   a. Click **CANopen device list** and right-click on it to display the short-cut menu. In the short-cut menu, select **Import EDS**.

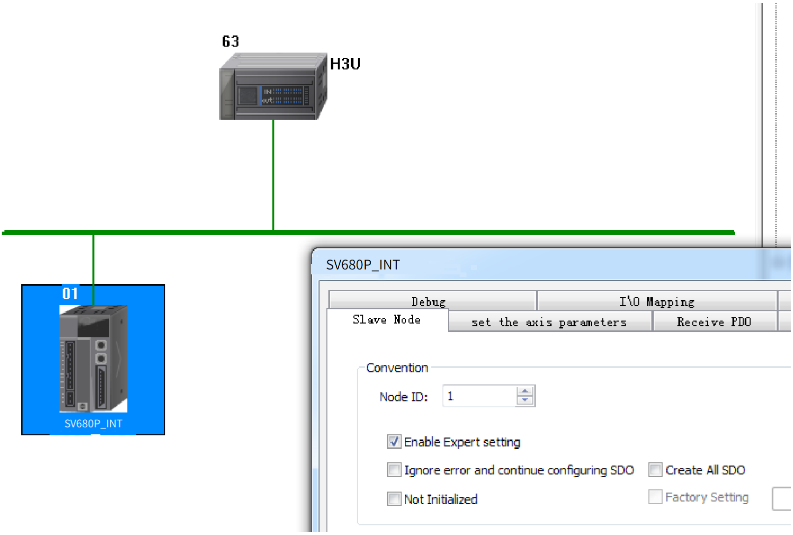


b. In the dialog box displayed, select the EDS file needed and click **Open**.



c. The device added will be displayed in the CANopen device list on the right.

5. Double-click the **SV680P** in the **CANopen device list** to add CANopen slaves. Then, double-click the **SV680P-INT** icon in the configuration to open the slave configuration parameter list.



SV680P_INT

Convention

Node ID: 1

☑ Enable Expert setting

□ Ignore error and continue configuring SDO

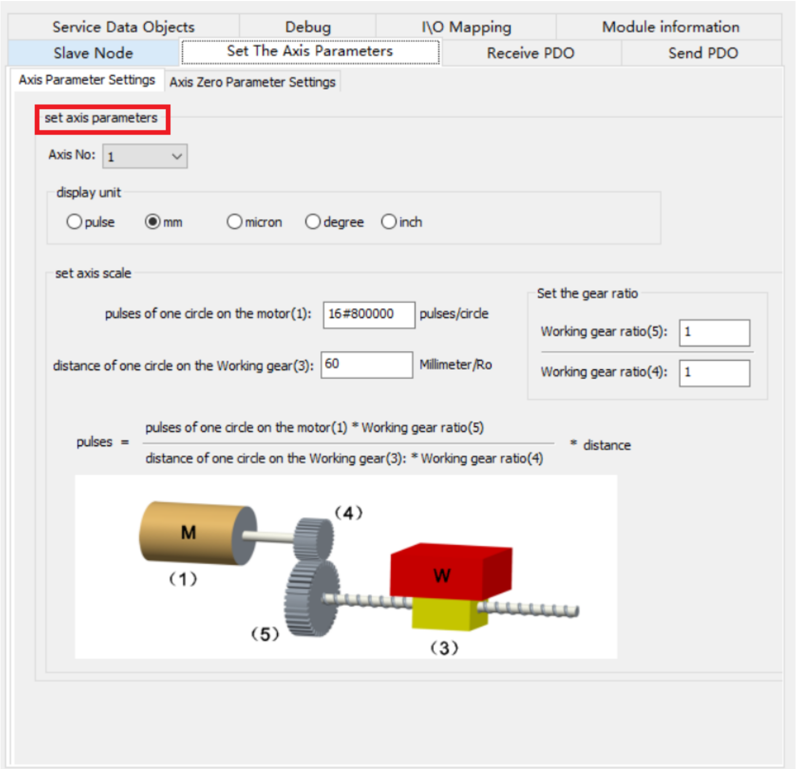
□ Create All SDO

□ Not Initialized

□ Factory Setting

6. The **axis parameters setting** interface is shown as follows, which include **axis parameter setting** and **homing parameter setting**.

   Setting axis parameters



Service Data Objects | Debug | I\O Mapping | Module information

Slave Node | Set The Axis Parameters | Receive PDO | Send PDO

Axis Parameter Settings | Axis Zero Parameter Settings

set axis parameters

Axis No: 1

display unit

○pulse ●mm ○micron ○degree ○inch

set axis scale

pulses of one circle on the motor(1): 16#800000 pulses/circle

distance of one circle on the Working gear(3): 60 Millimeter/Ro

Set the gear ratio

Working gear ratio(5): 1

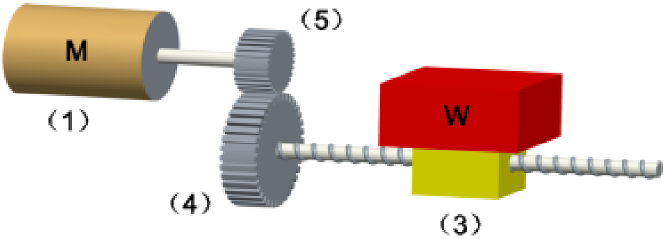
Working gear ratio(4): 1

$$\text{pulses} = \frac{\text{pulses of one circle on the motor(1) * Working gear ratio(5)}}{\text{distance of one circle on the Working gear(3): * Working gear ratio(4)}} \text{ * distance}$$

- For devices without reducers, set the gear ratio to 1:1. Set the pulses per motor revolution and distance per motor revolution correctly. The calculation formula is as follows.

$$\text{Pulses} = \frac{\text{Pulses of one circle on the motor (1)}}{\text{Distance of one circle on the working gear (3)}} \text{ x Distance (in displayed unit)}$$

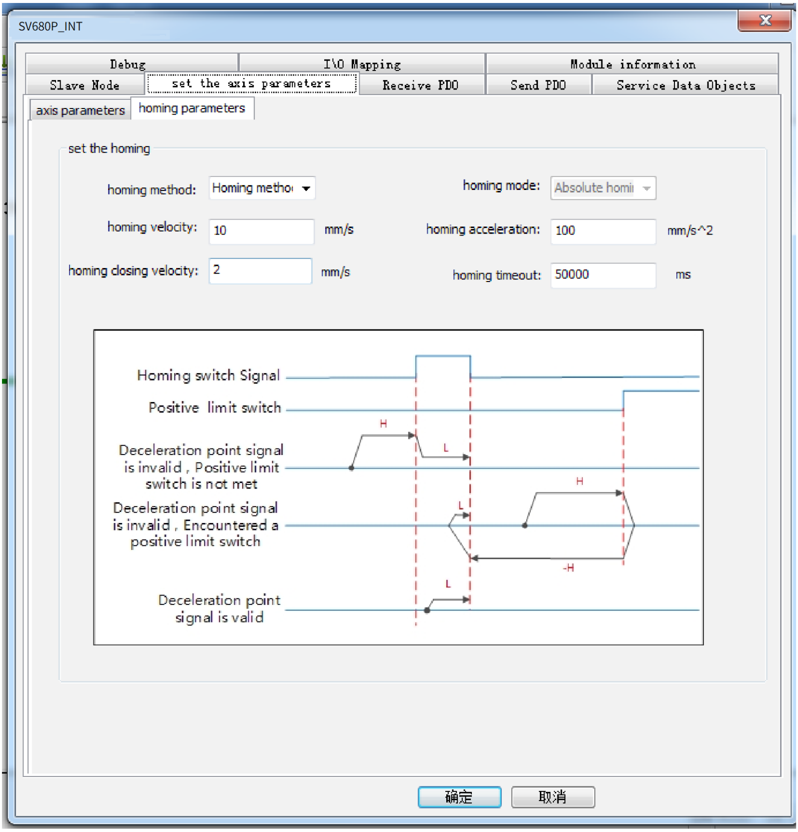
- Applications with reducers are shown as follows.



The calculation formula for devices with reducers is as follows.

$$\text{Pulses} = \frac{\text{Pulses of one circle on the motor (1) x Working gear ratio (5)}}{\text{Distance of one circle on the working gear (3) x Working gear ratio (4)}} \text{ x Distance (in displayed unit)}$$

Setting axis homing parameters



set the homing

homing method: Homing metho

homing mode: Absolute homi

homing velocity: 10 mm/s

homing acceleration: 100 mm/s^2

homing closing velocity: 2 mm/s

homing timeout: 50000 ms

The range of the homing method is 1 to 35. The calculation formula for parameters and object dictionaries of the homing speed, homing acceleration, and homing proximity speed is shown as follows.

$$\text{Object dictionary value} = \frac{\text{Pulses of one circle on the motor (1) x Working gear ratio (5)}}{\text{Distance of one circle on the working gear (3) x Working gear ratio (4)}} \text{ x } \begin{matrix}\text{Setpoint in the software tool}\\ \text{(in displayed unit)}\end{matrix}$$

The relation between preceding parameters and object dictionaries is as follows.

Index	Sub-index	Data type	Description	Unit
6098h	00	SINT	Homing method	-
6099h	01	UDINT	Speed during search for switch	Reference unit/s

Index	Sub-index	Data type	Description	Unit
6099h	02	UDINT	Speed during search for zero	Reference unit/s
609Ah	00	UDINT	Homing acceleration	Reference unit/s^2
60E6h	00	USINT	Homing method	-

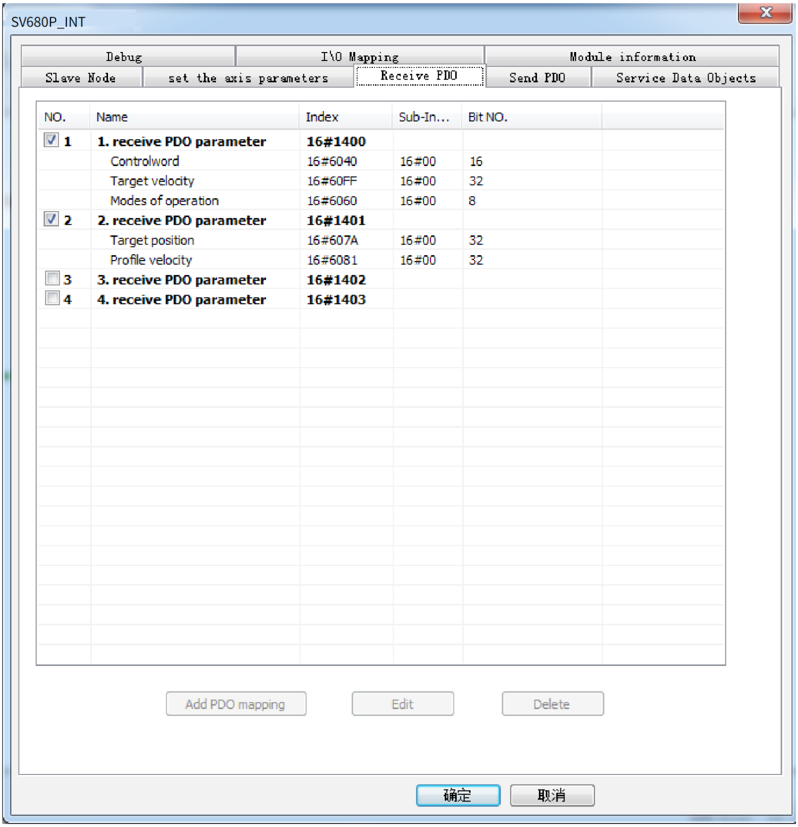
7. The object dictionaries involved in CANopen CiA402 motion control commands interact with the slave in the PDO mode. These object dictionaries, which include 6040h (Control word), 6041h (Status word), 6060h (Modes of operation), 6061h (Modes of operation display), 6081h (Profile velocity), 607Ah (Target position), 60FFh (Target velocity), 6064h (Position actual value), and 606Ch Velocity actual value), must be configured as required below. Otherwise, axis configuration failure may occur during calling axis control commands.

Note

It is recommended to configure the PDO communication to synchronous mode to prevent frame loss caused by interference during communication. The synchronous mode requires synchronous production to be enabled in the master configuration. To ensure communication stability, the network load rate must be lower than 70%.

$$\text{Network load rate} = \frac{328 \text{ x Number of axes} + 79}{\text{Baud rate x SYNC cycle}} \text{ x } 100\%$$

Configuring the RPDOs



NO.	Name	Index	Sub-In...	Bit NO.
☑ 1	**1. receive PDO parameter**	**16#1400**		
	Controlword	16#6040	16#00	16
	Target velocity	16#60FF	16#00	32
	Modes of operation	16#6060	16#00	8
☑ 2	**2. receive PDO parameter**	**16#1401**		
	Target position	16#607A	16#00	32
	Profile velocity	16#6081	16#00	32
□ 3	**3. receive PDO parameter**	**16#1402**		
□ 4	**4. receive PDO parameter**	**16#1403**		

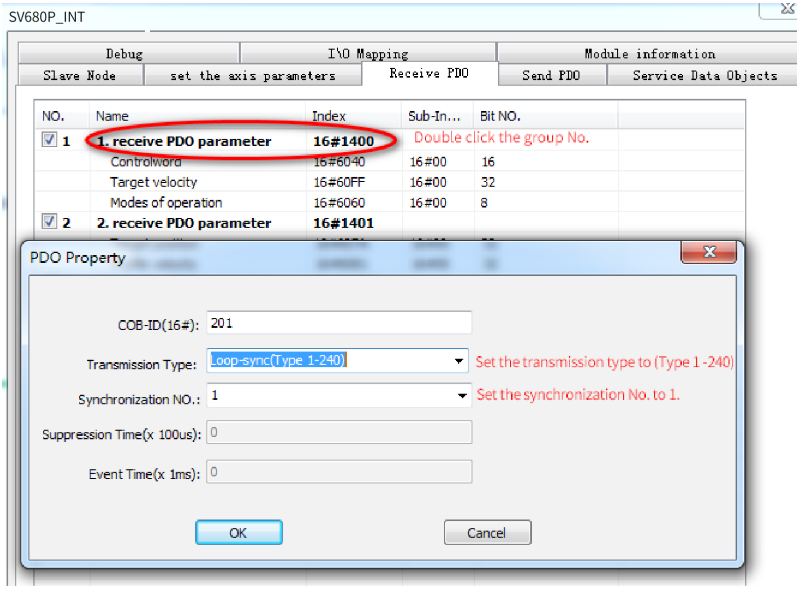
Configure the RPDOs in the following sequence.

Index	Sub-index	Name
6040h	00	Control word
60FFh[1]	00	Target velocity
6060h	00	Modes of operation
607Ah	00	Target position
6081h	00	Profile velocity

Note

[1]: The object dictionary can be replaced by other object dictionaries with a length of 0x20.

It is recommended to use synchronous mode for PDO communication. The method for setting synchronous PDO communication of the slave is as follows.



SV680P_INT

NO.	Name	Index	Sub-In...	Bit NO.
☑ 1	1. receive PDO parameter	16#1400		
	Controlword	16#6040	16#00	16
	Target velocity	16#60FF	16#00	32
	Modes of operation	16#6060	16#00	8
☑ 2	2. receive PDO parameter	16#1401		

Double click the group No.

PDO Property

COB-ID(16#): 201

Transmission Type: Loop-sync(Type 1-240)

Synchronization NO.: 1

Suppression Time(x 100us): 0

Event Time(x 1ms): 0

Set the transmission type to (Type 1 -240)

Set the synchronization No. to 1.

Note

When MCMOVVEL and MCJOG are not in use, this object dictionary can be replaced by other object dictionaries with a length of 0x20.

Steps:

- 1. Double-click the group No. and a dialog box appears.
- 2. Set "Transmission Type" to "Type1-240".
- 3. Set "Synchronization NO." to "1".

Configuring TPDOs:

Configure the TPDOs in the following sequence.

Index	Sub-index	Name
6041h	00	Status word
60FDh[1]	00	Digital inputs
6061h	00	Modes of operation
6064h	00	Position actual value
606Ch	00	Velocity actual value

Note

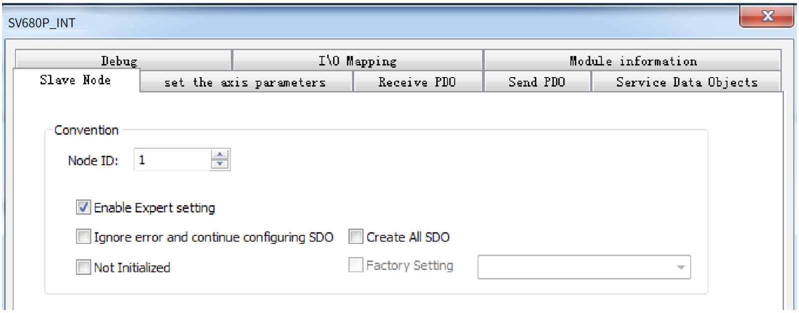
[1]: The object dictionary can be replaced by other object dictionaries with a length of 0x20.

The mode for setting TPDOs is similar to that for RPDOs.

Caution

The EDS must be configured based on the preceding sequence by default. Observe the preceding configuration sequence when adding new objects. A wrong sequence will cause failure of H3U axis control commands. The preceding configuration sequence does not necessarily apply to PLCs from other manufacturers.

8. Download the CANopen configuration to H3U. The H3U starts slave configuration based on the previous configurations. Configuration is performed based on the object dictionaries listed in the **Servo Data Object** interface. To view this list, check **Enable Expert setting** in the **Slave Node** interface first.



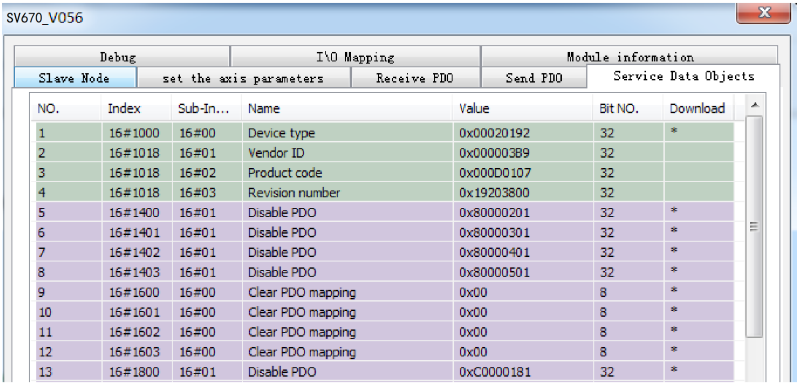
Convention

Node ID: 1

☑ Enable Expert setting

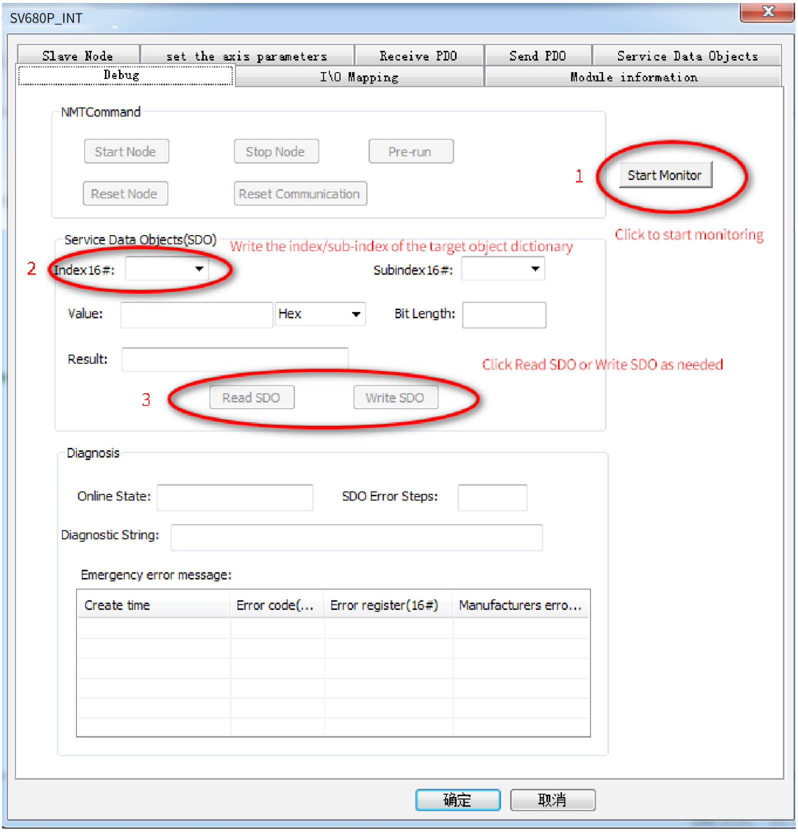
☐ Ignore error and continue configuring SDO ☐ Create All SDO

☐ Not Initialized ☐ Factory Setting



NO.	Index	Sub-In...	Name	Value	Bit NO.	Download
1	16#1000	16#00	Device type	0x00020192	32	*
2	16#1018	16#01	Vendor ID	0x000003B9	32	
3	16#1018	16#02	Product code	0x000D0107	32	
4	16#1018	16#03	Revision number	0x19203800	32	
5	16#1400	16#01	Disable PDO	0x80000201	32	*
6	16#1401	16#01	Disable PDO	0x80000301	32	*
7	16#1402	16#01	Disable PDO	0x80000401	32	*
8	16#1403	16#01	Disable PDO	0x80000501	32	*
9	16#1600	16#00	Clear PDO mapping	0x00	8	*
10	16#1601	16#00	Clear PDO mapping	0x00	8	*
11	16#1602	16#00	Clear PDO mapping	0x00	8	*
12	16#1603	16#00	Clear PDO mapping	0x00	8	*
13	16#1800	16#01	Disable PDO	0xC0000181	32	*

During commissioning, you can monitor the device status online and read/write the object dictionary of the slave through H3U, as shown below.

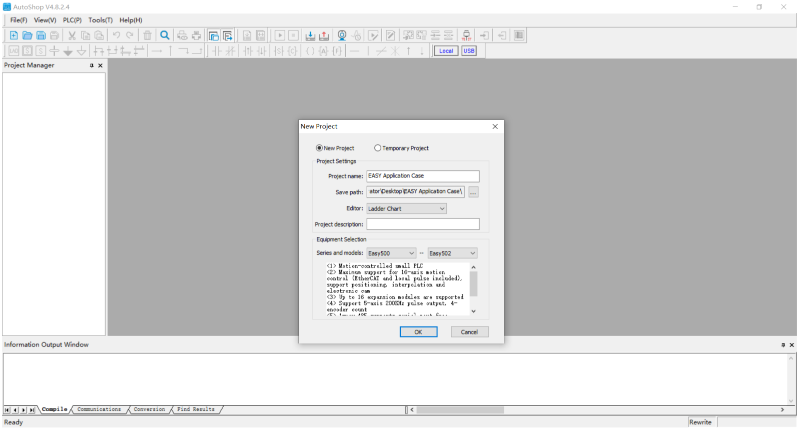


Where:

- 1. Click **Start Monitor**.
- 2. Write the index of the object dictionary to be operated in **Index** and the sub-index in **Subindex**.
- 3. Click **Read SDO** or **Write SDO** as needed.

5.2.4 Connecting SV680P-INT to Inovance EASY CANopen Master

1. Open Autoshop and click **New Project**. In the popup dialog box, first select the editor type, and then select Easy300 as the PLC type. Enter the project name and select the save path, and then click "OK" to create a new project and enter the project main interface.



New Project

● New Project ○ Temporary Project

Project Settings

Project name: EASY Application Case

Save path: ator\Desktop\EASY Application Case\

Editor: Ladder Chart

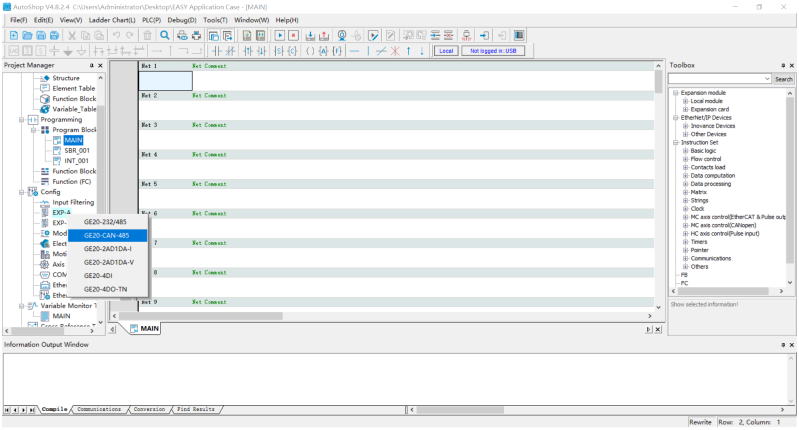
Project description:

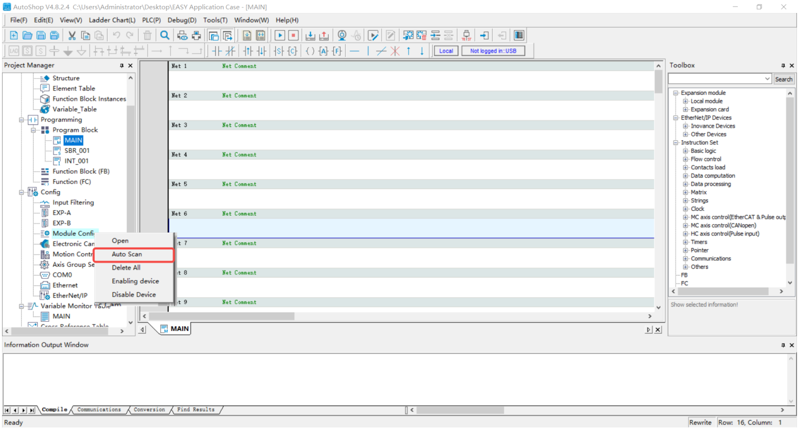
Equipment Selection

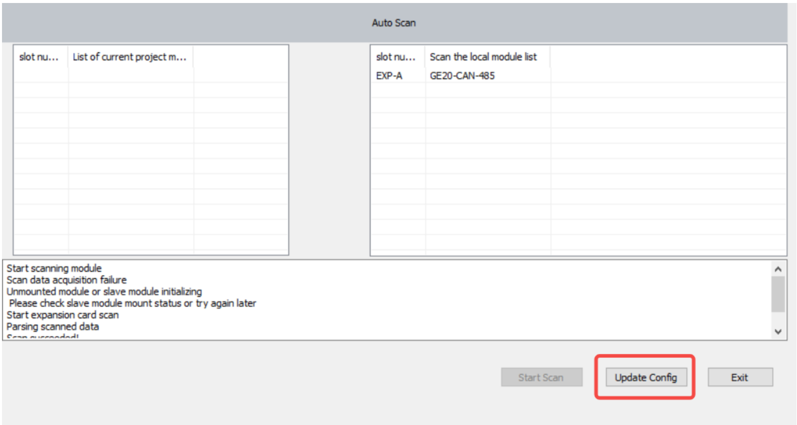
Series and models: Easy500 -- Easy502

```
<1> Motion-controlled small PLC
<2> Maximum support for 16-axis motion
control (EtherCAT and local pulse included),
support positioning, interpolation and
electronic cam
<3> Up to 16 expansion modules are supported
<4> Support 5-axis 200KHz pulse output, 4-
encoder count
```

2. Select **GE20-CAN-485** on the right side of the navigation tree of **Configure EXP-A** in the Project Management window. Or select **Auto Scan** in the navigation tree of **Module Configuration** to add an GE20-CAN-485 expansion card, as shown in the following figure. The GE20-CAN-485 expansion card only supports EXP-A.





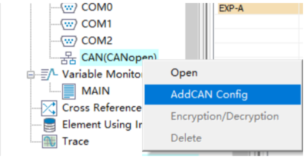

Auto Scan

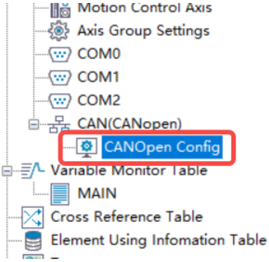
slot nu...	List of current project m...

slot nu...	Scan the local module list
EXP-A	GE20-CAN-485

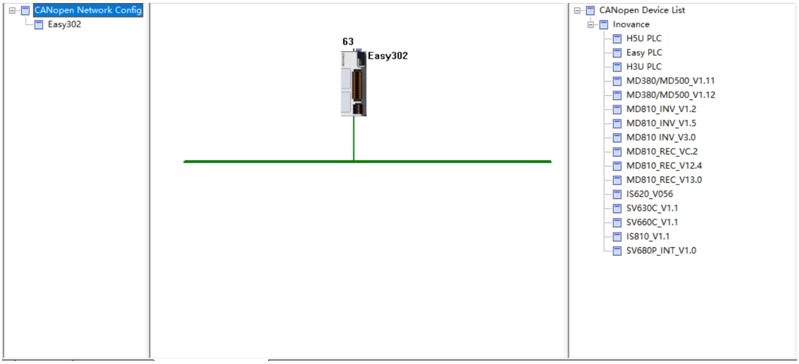
```
Start scanning module
Scan data acquisition failure
Unmounted module or slave module initializing
 Please check slave module mount status or try again later
Start expansion card scan
Parsing scanned data
Scan succeeded!
```

3. Double-click **CAN** in **Configuration** of **Project Management**, select CANopen in the pop-up window, set the station number and baud rate, and click OK. At this time, CAN is configured as a CANopen slave station,. Configure it as a CANopen master station by right-clicking **CAN** in **Configuration** of **Project Management** and selecting **Add CAN Configuration** in the pop-up menu, as shown in the following figure.

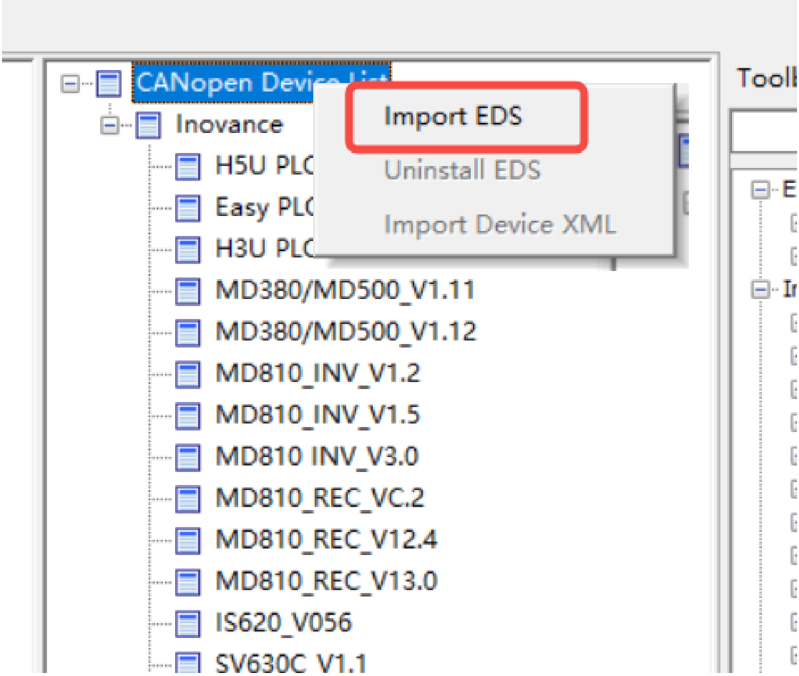




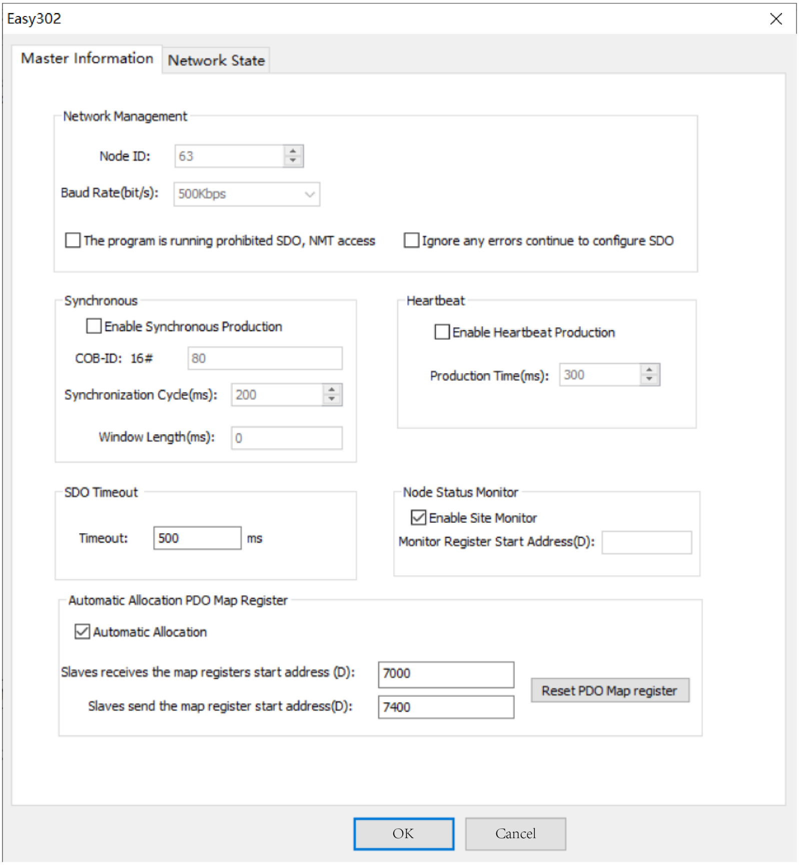
4. Double click CANopen Config to open the CANopen Configuration interface, as shown below:



5. If the EDS files needed is not in the CANopen device list, add the device EDS needed. Click **CANopen device list** and right-click on it to display the short-cut menu. In the short-cut menu, select **Import EDS**. In the dialog box displayed, select the EDS device file needed and click **Open**. The device added will be displayed in the CANopen device list on the right.



6. Double-click the EASY master station to open the master configuration interface, in which you can set parameters such as synchronization and heartbeat.



Easy302

Master Information | Network State

Network Management

Node ID: 63

Baud Rate(bit/s): 500Kbps

□ The program is running prohibited SDO, NMT access

□ Ignore any errors continue to configure SDO

Synchronous

□ Enable Synchronous Production

COB-ID: 16# 80

Synchronization Cycle(ms): 200

Window Length(ms): 0

Heartbeat

□ Enable Heartbeat Production

Production Time(ms): 300

SDO Timeout

Timeout: 500 ms

Node Status Monitor

☑ Enable Site Monitor

Monitor Register Start Address(D):

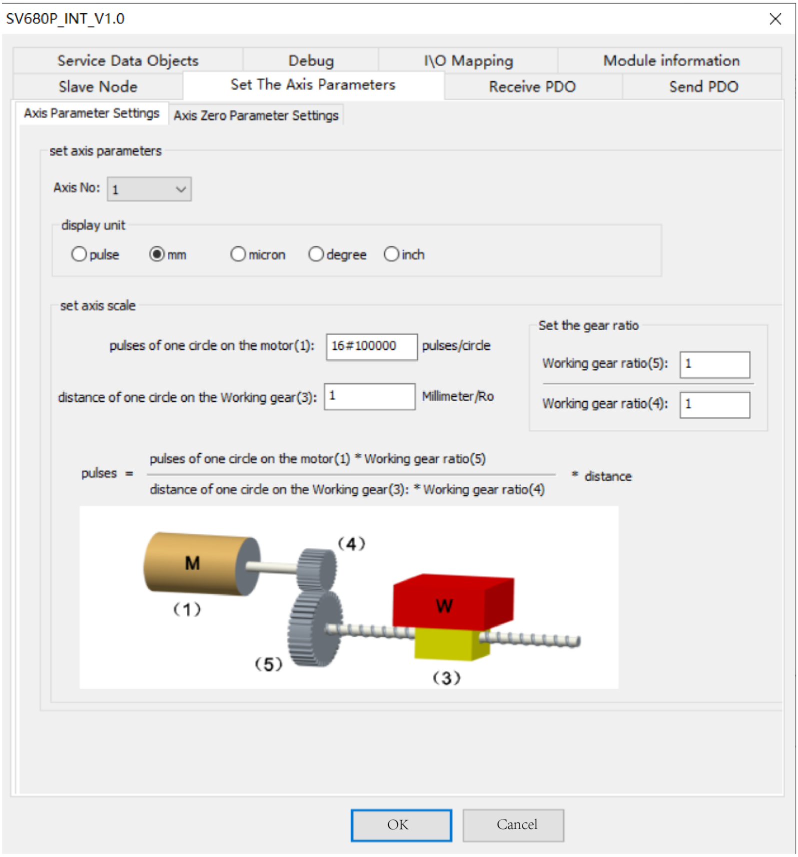
Automatic Allocation PDO Map Register

☑ Automatic Allocation

Slaves receives the map registers start address (D): 7000

Slaves send the map register start address(D): 7400

7. Double-click the **SV680P_INT** in the **CANopen device list** to add CANopen slaves. Then, double-click the **SV680P_INT** icon in the configuration to open the slave configuration parameter list.
8. The **axis parameters setting** interface is shown as follows, which include **axis parameter setting** and **homing parameter setting**.



set axis parameters

Axis No: 1

display unit

○ pulse ● mm ○ micron ○ degree ○ inch

set axis scale

pulses of one circle on the motor(1): 16#100000 pulses/circle

distance of one circle on the Working gear(3): 1 Millimeter/Ro

Set the gear ratio

Working gear ratio(5): 1

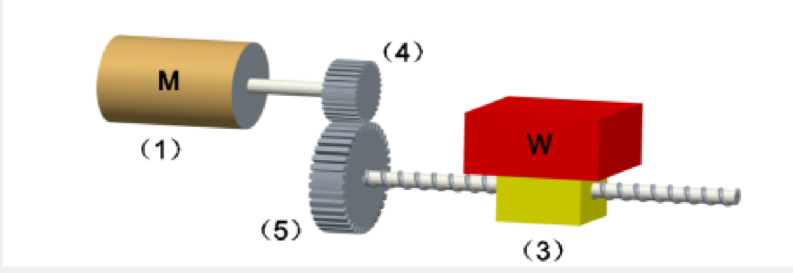
Working gear ratio(4): 1

$$\text{pulses} = \frac{\text{pulses of one circle on the motor(1) * Working gear ratio(5)}}{\text{distance of one circle on the Working gear(3): * Working gear ratio(4)}} \text{ * distance}$$

- **Setting axis parameters**
  For devices without reducers, set the gear ratio to 1:1. Set the pulses per motor revolution and distance per motor revolution correctly. The calculation formula is as follows.

$$\text{Number of pulses} = \frac{\text{Number of pulses per revolution (1)}}{\text{Distance per revolution (3)}} \text{ x Moving distance (displayed unit)}$$

Applications with reducers are shown as follows.

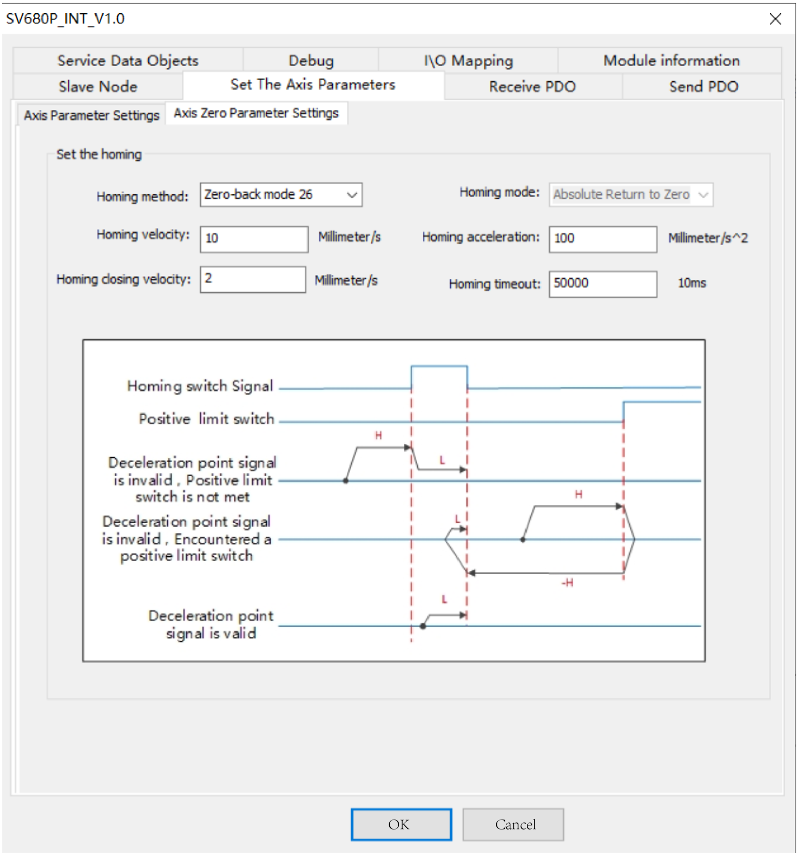


The calculation formula is as follows.

$$\text{Number of pulses} = \frac{\text{Pulses per revolution (1) x Motor gear ratio (5)}}{\text{Distance per revolution (3) x Working gear ratio (4)}} \text{ x Moving distance (displayed unit)}$$

- **Homing**

  The range of homing modes is 1-35. For details of each mode, see *SV680P-INT Series Servo Drive Function Guide.*


SV680P_INT_V1.0
Service Data Objects
Debug
I\O Mapping
Module information
Slave Node
Set The Axis Parameters
Receive PDO
Send PDO
Axis Parameter Settings
Axis Zero Parameter Settings

Set the homing

Homing method: Zero-back mode 26

Homing mode: Absolute Return to Zero

Homing velocity: 10 Millimeter/s

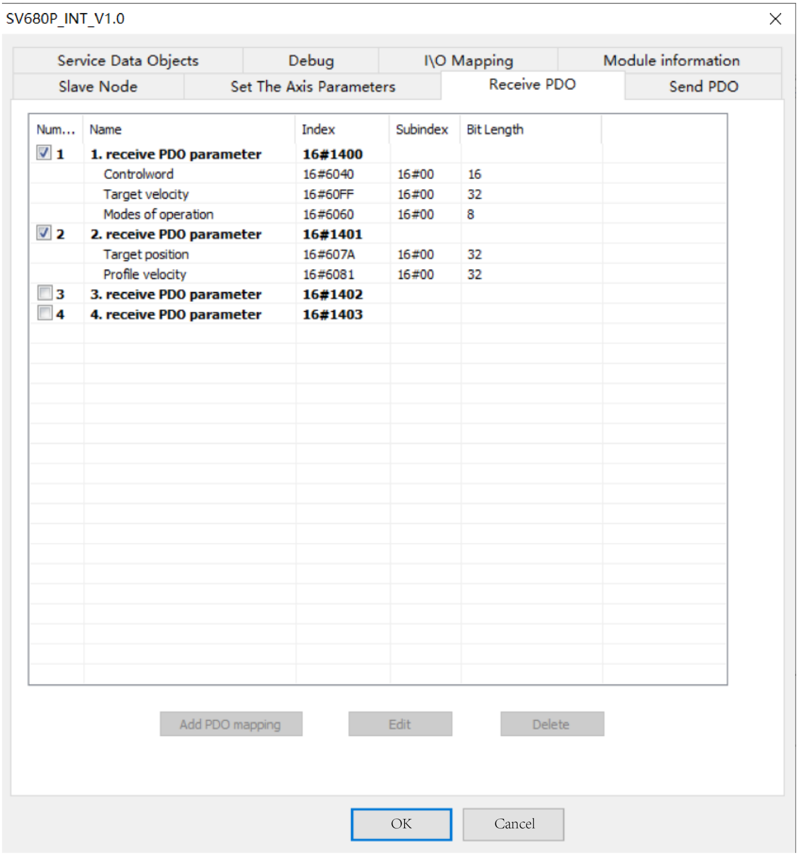
Homing acceleration: 100 Millimeter/s^2

Homing closing velocity: 2 Millimeter/s

Homing timeout: 50000 10ms

Homing switch Signal
Positive limit switch
Deceleration point signal is invalid , Positive limit switch is not met
Deceleration point signal is invalid , Encountered a positive limit switch
Deceleration point signal is valid
H
L
H
L
-H
L
OK
Cancel

9. Click **Receive PDO** or **Transmit PDO**. The following interface is displayed.



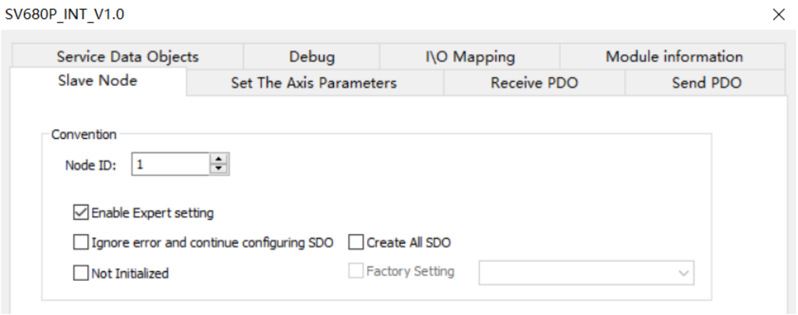
Num...	Name	Index	Subindex	Bit Length
☑ 1	**1. receive PDO parameter**	**16#1400**		
	Controlword	16#6040	16#00	16
	Target velocity	16#60FF	16#00	32
	Modes of operation	16#6060	16#00	8
☑ 2	**2. receive PDO parameter**	**16#1401**		
	Target position	16#607A	16#00	32
	Profile velocity	16#6081	16#00	32
□ 3	**3. receive PDO parameter**	**16#1402**		
□ 4	**4. receive PDO parameter**	**16#1403**		

Receive PDO Parameter: Indicates the data sent by the master station to a slave station.

Send PDO Parameter: Indicates the data sent by a slave station to the master station.

You can check the box in front of the number to enable a PDO. The PDOs in the EDS file that take effect by default are already checked. You can click **Add PDO mapping**, **Edit**, or **Delete** to edit PDO mapping.

10. Download the CANopen configuration to EASY. The EASY starts slave configuration based on the previous configurations. Configuration is performed based on the service object list. To view this list, check **Enable Expert setting** in the **Slave Node** interface first.



Convention

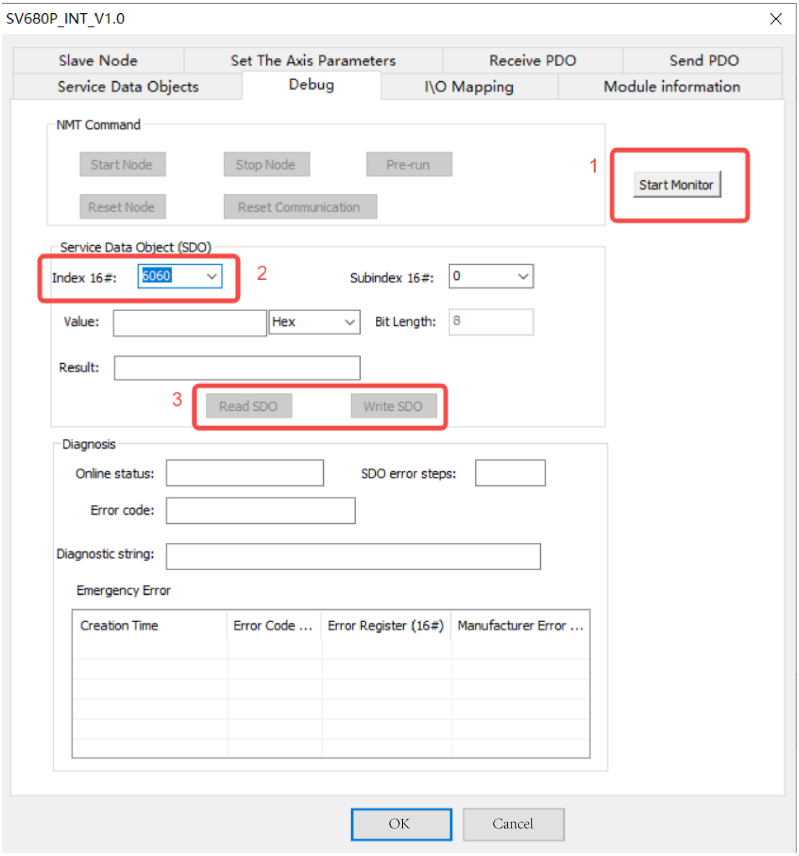
Node ID: 1

☑ Enable Expert setting

□ Ignore error and continue configuring SDO □ Create All SDO

□ Not Initialized □ Factory Setting

During commissioning, you can monitor the device status online and read/write the object dictionary of the slave through EASY, as shown below.



Service Data Object (SDO)

Index 16#: 6060

Subindex 16#: 0

Value: Hex

Bit Length: 8

Result:

Diagnosis

Online status:

SDO error steps:

Error code:

Diagnostic string:

Emergency Error

Creation Time	Error Code ...	Error Register (16#)	Manufacturer Error ...

5.3 EtherCAT Communication Configuration Case [N]

5.3.1 SV680N-INT and AM600 Controller

This section describes how to configure the SV680N-INT series servo drive for cooperation with the AM600 series controller.

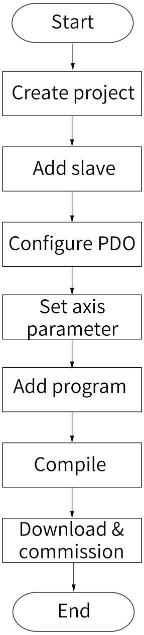
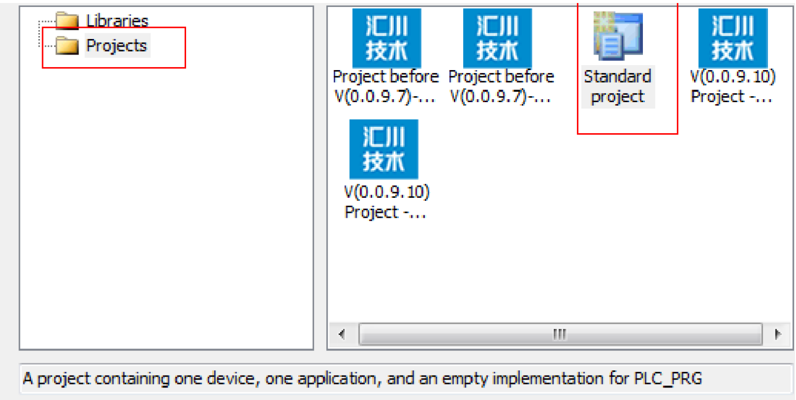


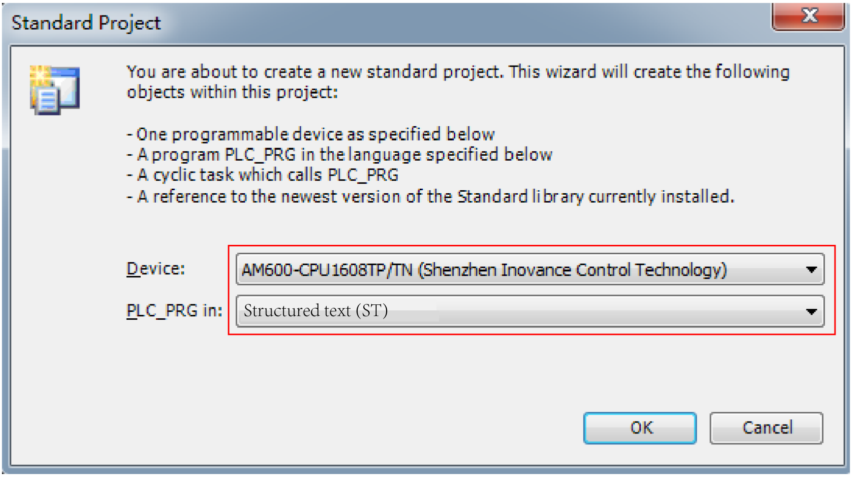
Figure 5-2 Configuration flowchart

Open the software and create an AM600 project.

Select **AM600-CPU1608TP**, as shown in the following interface.



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



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

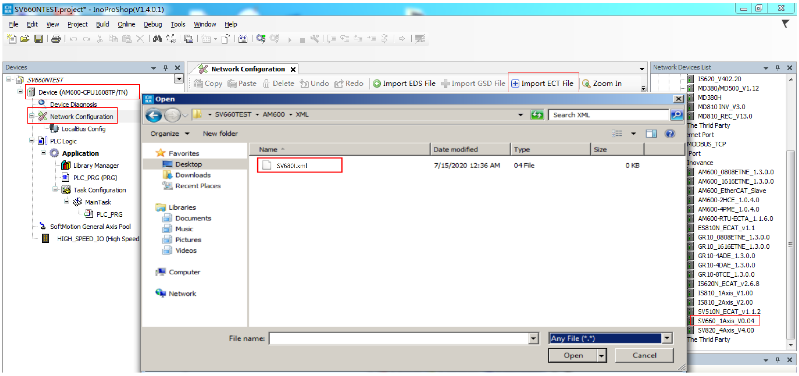
- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

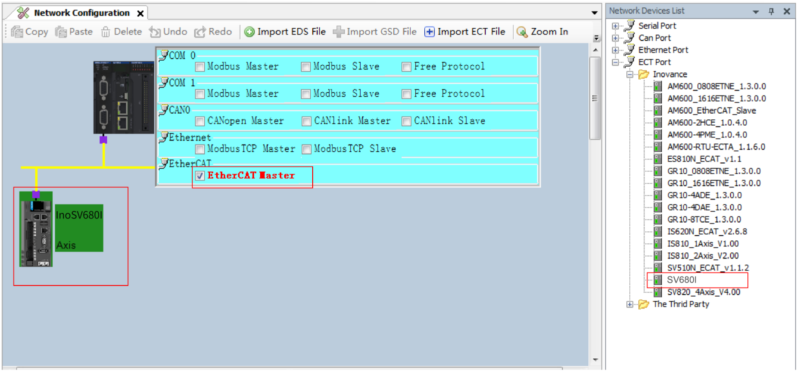
Device: AM600-CPU1608TP/TN (Shenzhen Inovance Control Technology)

PLC_PRG in: Structured text (ST)

Adding the SV680N-INT servo drive as slave

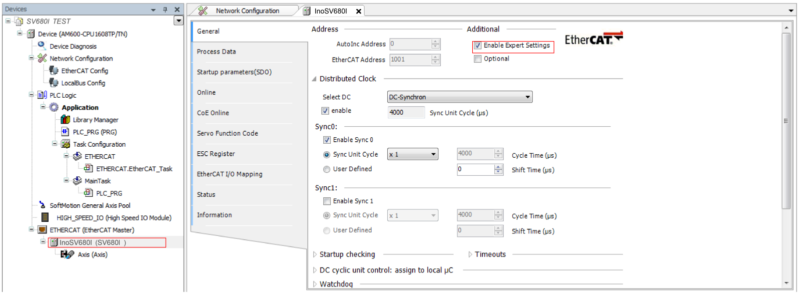
Open the network configuration and import the ECT file of SV680N-INT. Add an SV680N-INT as a slave, as shown in the following interface.





Configuring PDO

Select **Enable Expert Settings** and configure PDOs in the process data as needed. In this case, CSP is used as the operation mode and the default values of 1600 and 1A00 are used for PDO parameters.



Address

AutoInc Address: 0

EtherCAT Address: 1001

Additional

☑ Enable Expert Settings

□ Optional

Distributed Clock

Select DC: DC-Synchron

☑ enable 4000 Sync Unit Cycle (µs)

Sync0:

☑ Enable Sync 0

● Sync Unit Cycle x 1 — 4000 Cycle Time (µs)

○ User Defined — 0 Shift Time (µs)

Sync1:

□ Enable Sync 1

● Sync Unit Cycle x 1 — 4000 Cycle Time (µs)

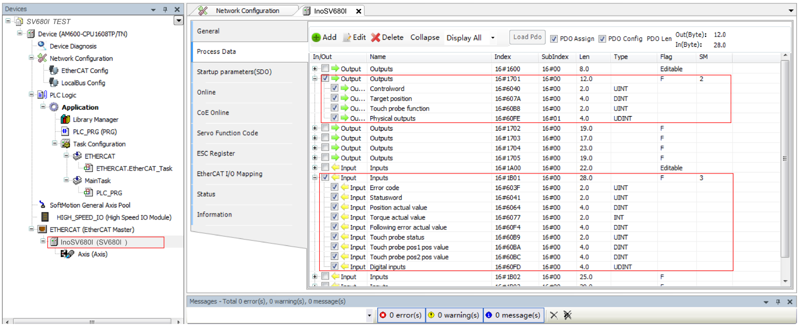
○ User Defined — 0 Shift Time (µs)

Startup checking

Timeouts

DC cyclic unit control: assign to local µC

Watchdog



☑ PDO Assign ☑ PDO Config PDO Len Out(Byte): 12.0 In(Byte): 28.0

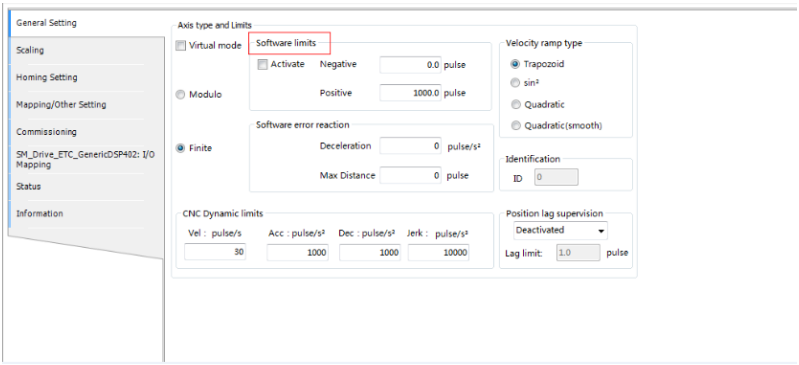
In/Out	Name	Index	SubIndex	Len	Type	Flag	SM
□ Output	Outputs	16#1600	16#00	8.0		Editable	
☑ Output	Outputs	16#1701	16#00	12.0		F	2
☑ Ou…	Controlword	16#6040	16#00	2.0	UINT		
☑ Ou…	Target position	16#607A	16#00	4.0	DINT		
☑ Ou…	Touch probe function	16#60B8	16#00	2.0	UINT		
☑ Ou…	Physical outputs	16#60FE	16#01	4.0	UDINT		
□ Output	Outputs	16#1702	16#00	19.0		F	
□ Output	Outputs	16#1703	16#00	17.0		F	
□ Output	Outputs	16#1704	16#00	23.0		F	
□ Output	Outputs	16#1705	16#00	19.0		F	
□ Input	Inputs	16#1A00	16#00	22.0		Editable	
☑ Input	Inputs	16#1B01	16#00	28.0		F	3
☑ Input	Error code	16#603F	16#00	2.0	UINT		
☑ Input	Statusword	16#6041	16#00	2.0	UINT		
☑ Input	Position actual value	16#6064	16#00	4.0	DINT		
☑ Input	Torque actual value	16#6077	16#00	2.0	INT		
☑ Input	Following error actual value	16#60F4	16#00	4.0	DINT		
☑ Input	Touch probe status	16#60B9	16#00	2.0	UINT		
☑ Input	Touch probe pos1 pos value	16#60BA	16#00	4.0	DINT		
☑ Input	Touch probe pos2 pos value	16#60BC	16#00	4.0	DINT		
☑ Input	Digital inputs	16#60FD	16#00	4.0	UDINT		
□ Input	Inputs	16#1B02	16#00	25.0		F	

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

0 error(s) 0 warning(s) 0 message(s)

Configuring axis parameters

1. Set the software position limit and the operation mode in basic axis settings.



Axis type and Limits

□ Virtual mode

○ Modulo

● Finite

Software limits

□ Activate Negative 0.0 pulse

Positive 1000.0 pulse

Software error reaction

Deceleration 0 pulse/s²

Max Distance 0 pulse

Velocity ramp type

● Trapozoid

○ sin²

○ Quadratic

○ Quadratic(smooth)

Identification

ID 0

CNC Dynamic limits

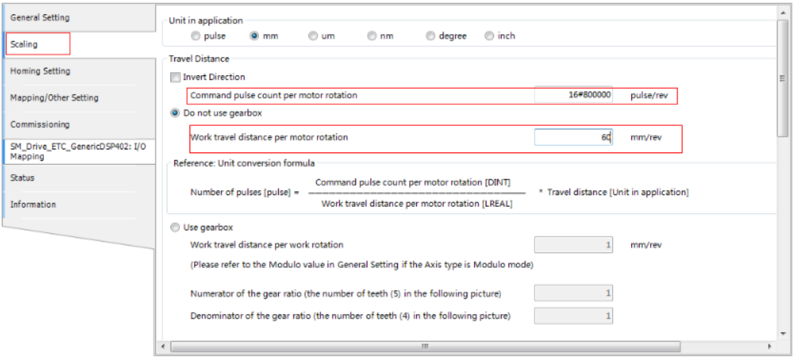
Vel : pulse/s	Acc : pulse/s²	Dec : pulse/s²	Jerk : pulse/s²
30	1000	1000	10000

Position lag supervision

Deactivated

Lag limit: 1.0 pulse

2. Select 16#4000000 for the 26-bit encoder, 16#800000 for the 23-bit encoder and 16#100000 for the 20-bit encoder during unit conversion. In this case, the single-turn travel distance is set to 60 mm and 1 mm/s equals to 1 RPM of the motor.



Unit in application

○ pulse ● mm ○ um ○ nm ○ degree ○ inch

Travel Distance

□ Invert Direction

Command pulse count per motor rotation 16#800000 pulse/rev

● Do not use gearbox

Work travel distance per motor rotation 60 mm/rev

Reference: Unit conversion formula

$$\text{Number of pulses [pulse]} = \frac{\text{Command pulse count per motor rotation [DINT]}}{\text{Work travel distance per motor rotation [LREAL]}} * \text{Travel distance [Unit in application]}$$

○ Use gearbox

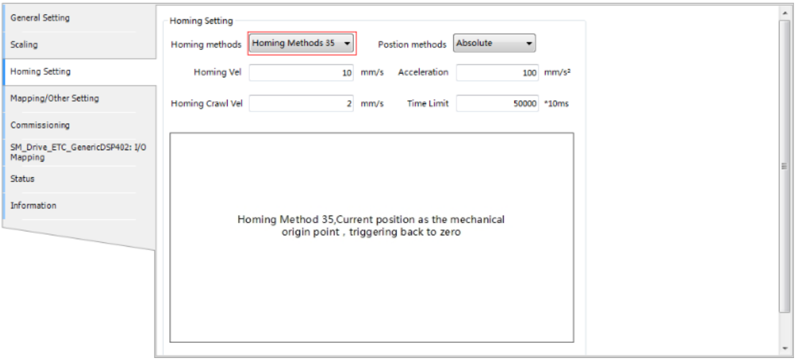
Work travel distance per work rotation 1 mm/rev

(Please refer to the Modulo value in General Setting if the Axis type is Modulo mode)

Numerator of the gear ratio (the number of teeth (5) in the following picture) 1

Denominator of the gear ratio (the number of teeth (4) in the following picture) 1

3. Select the homing mode according to actual needs. For details, see section *"Homing Mode"* in SV680-INT Series Servo Drive Function Guide for details.



Homing Setting

Homing methods: Homing Methods 35

Postion methods: Absolute

Homing Vel: 10 mm/s

Acceleration: 100 mm/s²

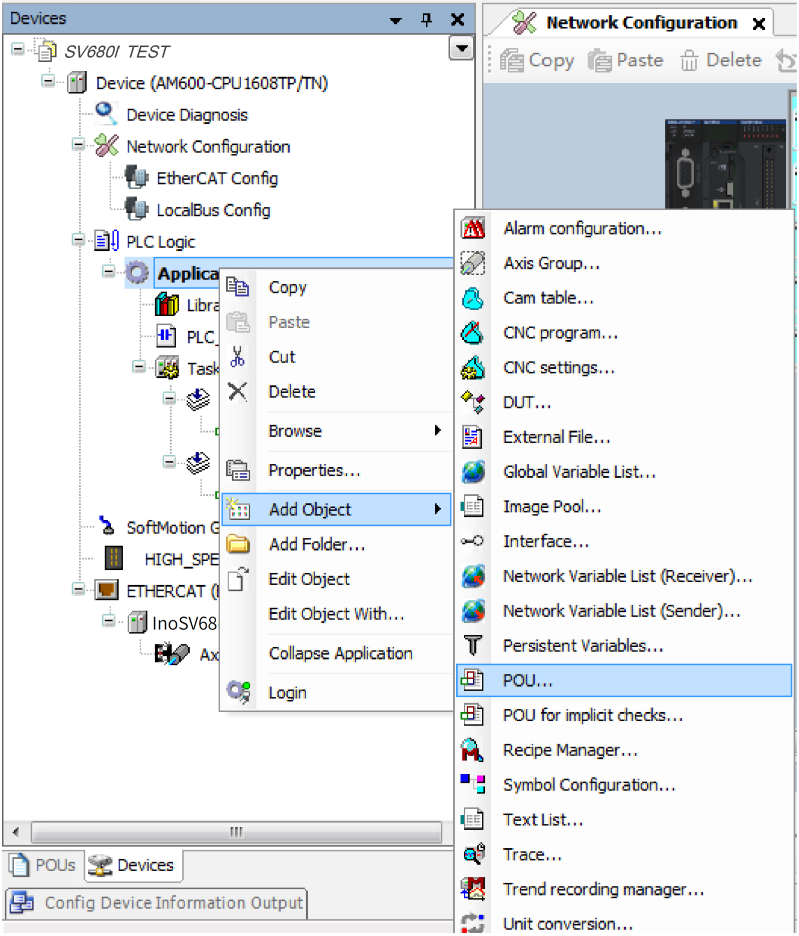
Homing Crawl Vel: 2 mm/s

Time Limit: 50000 *10ms

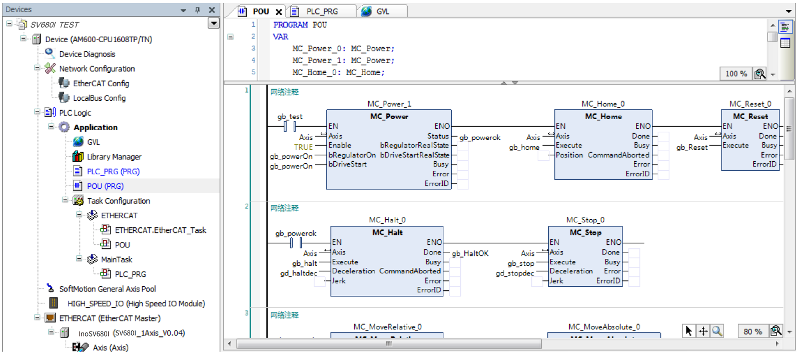
Homing Method 35,Current position as the mechanical origin point , triggering back to zero

Adding a program

Add a program to control the servo axis position, as shown by the following interface. See the following figure.

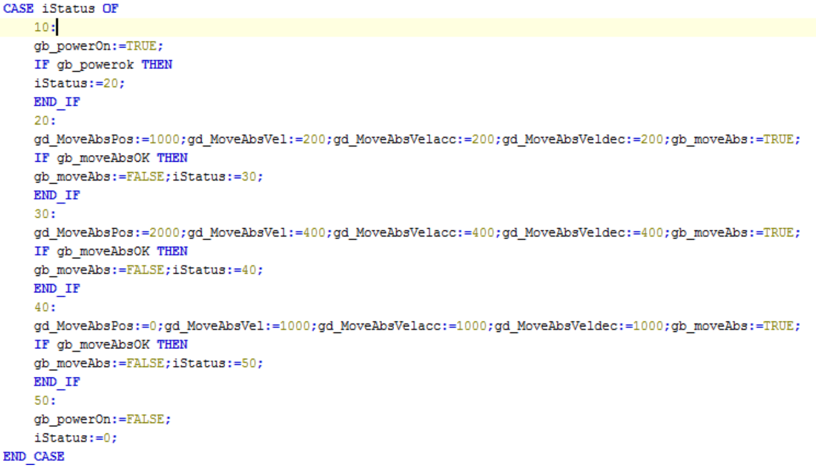


- Implement basic functions such as enabling, homing and positioning through adding function blocks.

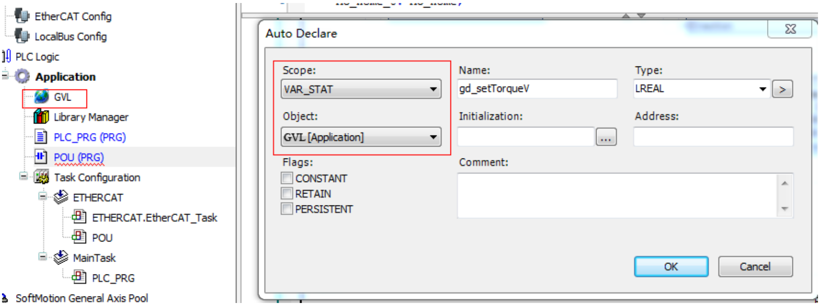


```
PROGRAM POU
VAR
    MC_Power_0: MC_Power;
    MC_Power_1: MC_Power;
    MC_Home_0: MC_Home;
```

- To implement directed motion through the logic program, some variables may need to be called to different POUs. Therefore, set the variables as global variables.



```
CASE iStatus OF
    10:
    gb_powerOn:=TRUE;
    IF gb_powerok THEN
    iStatus:=20;
    END_IF
    20:
    gd_MoveAbsPos:=1000;gd_MoveAbsVel:=200;gd_MoveAbsVelacc:=200;gd_MoveAbsVeldec:=200;gb_moveAbs:=TRUE;
    IF gb_moveAbsOK THEN
    gb_moveAbs:=FALSE;iStatus:=30;
    END_IF
    30:
    gd_MoveAbsPos:=2000;gd_MoveAbsVel:=400;gd_MoveAbsVelacc:=400;gd_MoveAbsVeldec:=400;gb_moveAbs:=TRUE;
    IF gb_moveAbsOK THEN
    gb_moveAbs:=FALSE;iStatus:=40;
    END_IF
    40:
    gd_MoveAbsPos:=0;gd_MoveAbsVel:=1000;gd_MoveAbsVelacc:=1000;gd_MoveAbsVeldec:=1000;gb_moveAbs:=TRUE;
    IF gb_moveAbsOK THEN
    gb_moveAbs:=FALSE;iStatus:=50;
    END_IF
    50:
    gb_powerOn:=FALSE;
    iStatus:=0;
END_CASE
```



Auto Declare

Scope: VAR_STAT

Name: gd_setTorqueV

Type: LREAL

Object: GVL [Application]

Initialization:

Address:

Flags:

□ CONSTANT

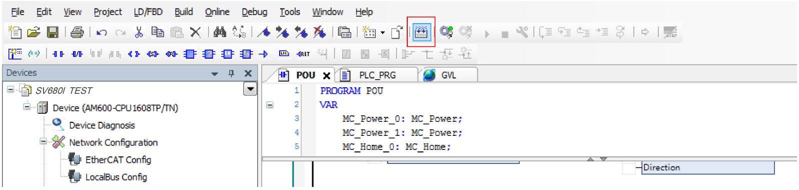
□ RETAIN

□ PERSISTENT

Comment:

Compiling

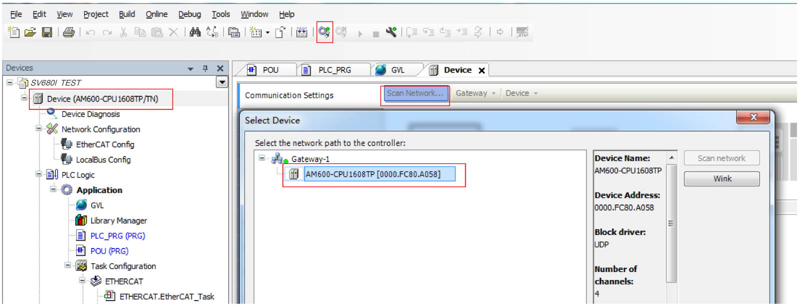
After compiling the program, click the icon indicated by the red square box to check whether the program is correct.



```
PROGRAM POU
VAR
    MC_Power_0: MC_Power;
    MC_Power_1: MC_Power;
    MC_Home_0: MC_Home;
```

Downloading and commissioning

1. After checking that the program is correct, download the program to PLC. The program can be activated after running. Before downloading, scan the PLCs first to select the PLC to be downloaded, and then click the download icon, as shown in the following interface.



Select the network path to the controller:

Gateway-1
AM600-CPU1608TP [0000.FC80.A058]

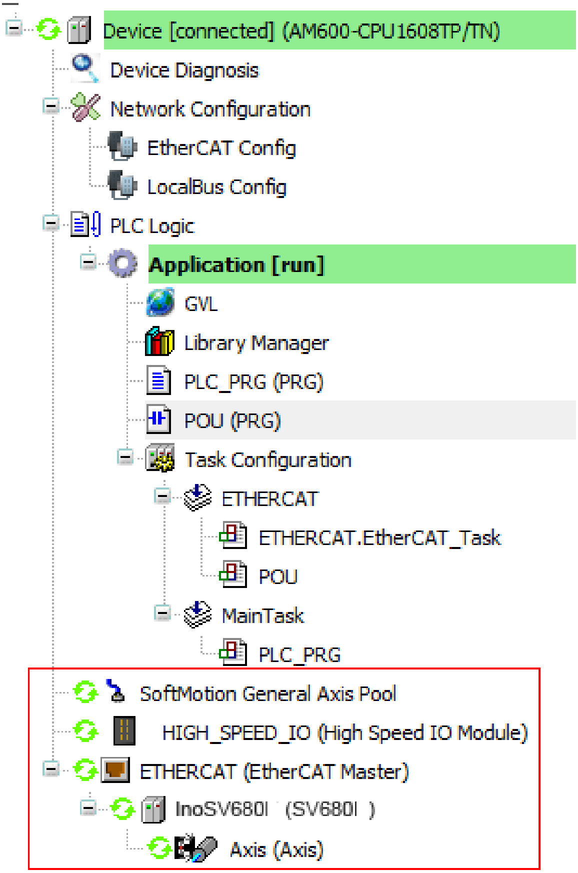
Device Name:
AM600-CPU1608TP

Device Address:
0000.FC80.A058

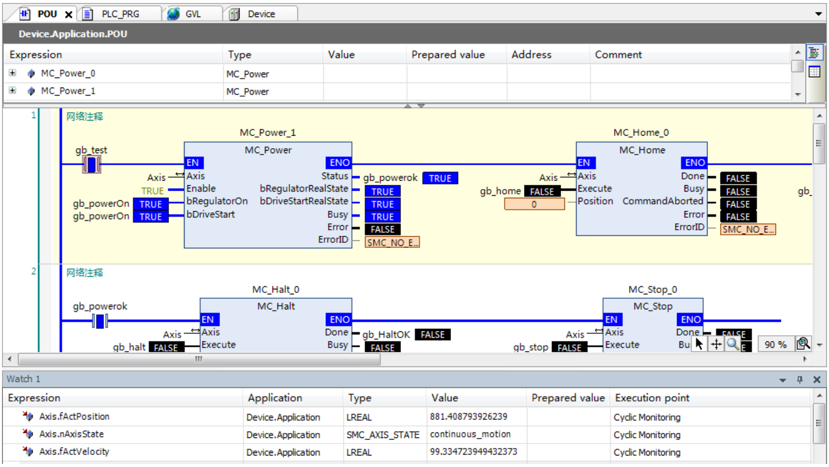
Block driver:
UDP

Number of channels:
4

2. After log-in, ensure the servo drive and the axis are in normal state.



3. Monitor critical parameters through the monitoring function. Start the testing program to perform basic tests such as enabling, homing and positioning.

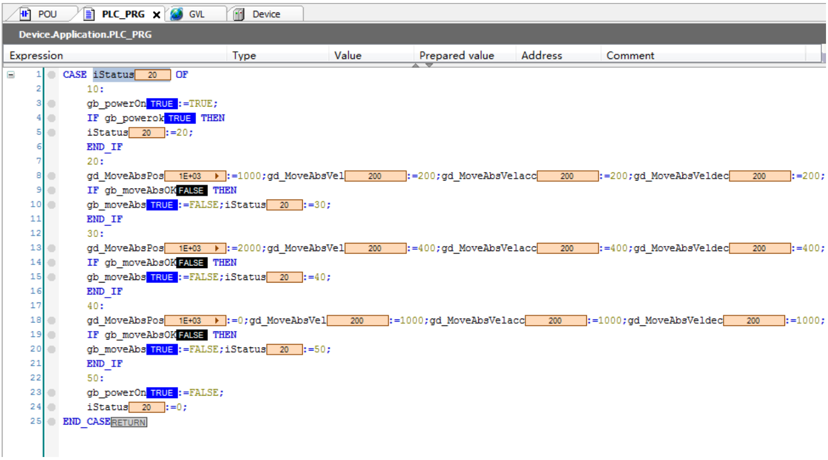


Device.Application.POU

Expression	Type	Value	Prepared value	Address	Comment
MC_Power_0	MC_Power				
MC_Power_1	MC_Power				

Expression	Application	Type	Value	Prepared value	Execution point
Axis.fActPosition	Device.Application	LREAL	881.408793926239		Cyclic Monitoring
Axis.nAxisState	Device.Application	SMC_AXIS_STATE	continuous_motion		Cyclic Monitoring
Axis.fActVelocity	Device.Application	LREAL	99.334723949432373		Cyclic Monitoring

4. After the testing is done, perform directed running program.



Device.Application.PLC_PRG

Expression	Type	Value	Prepared value	Address	Comment

```
CASE iStatus 20 OF
    10:
    gb_powerOn TRUE :=TRUE;
    IF gb_powerok TRUE THEN
    iStatus 20 :=20;
    END_IF
    20:
    gd_MoveAbsPos 1E+03 :=1000;gd_MoveAbsVel 200 :=200;gd_MoveAbsVelacc 200 :=200;gd_MoveAbsVeldec 200 :=200;
    IF gb_moveAbsOK FALSE THEN
    gb_moveAbs TRUE :=FALSE;iStatus 20 :=30;
    END_IF
    30:
    gd_MoveAbsPos 1E+03 :=2000;gd_MoveAbsVel 200 :=400;gd_MoveAbsVelacc 200 :=400;gd_MoveAbsVeldec 200 :=400;
    IF gb_moveAbsOK FALSE THEN
    gb_moveAbs TRUE :=FALSE;iStatus 20 :=40;
    END_IF
    40:
    gd_MoveAbsPos 1E+03 :=0;gd_MoveAbsVel 200 :=1000;gd_MoveAbsVelacc 200 :=1000;gd_MoveAbsVeldec 200 :=1000;
    IF gb_moveAbsOK FALSE THEN
    gb_moveAbs TRUE :=FALSE;iStatus 20 :=50;
    END_IF
    50:
    gb_powerOn TRUE :=FALSE;
    iStatus 20 :=0;
END_CASE RETURN
```

5.3.2 SV680N-INT and Omron Controller

This section describes how to configure the SV680N-INT series servo drive for working with an Omron NX701 controller.

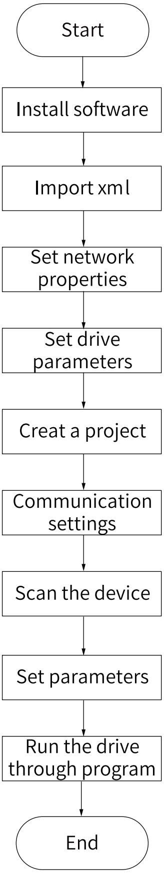


Figure 5-3 Configuration flowchart

Note

When more than 25 drives are networked with Omron NX701, you need to modify the cable length defined in the Omron master station. The cable length is calculated based on the fact that one drive needs a length of 36 m.

Installing the Sysmac Studio software

Install the Sysmac Studio software.

It is recommended to install V1.10 or above.

Importing the xml device description file

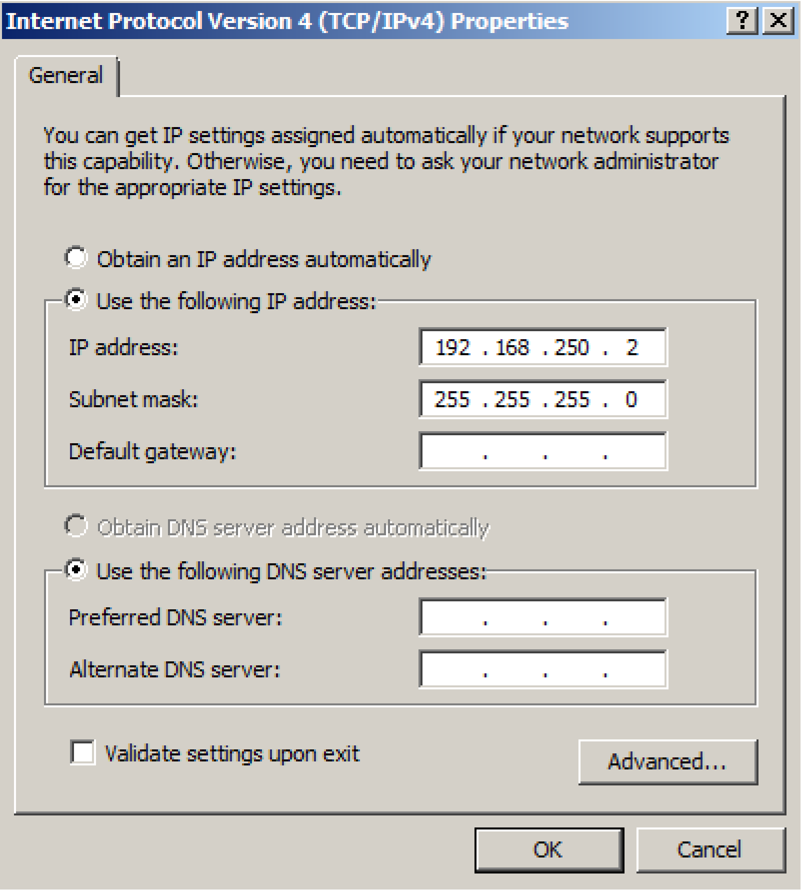
Importing the device description file (V2.5 or later recommended).

It is recommended to import the device description file of "SV680_INT_EOE_1Axis_02002_240110.xml" or later version. The file path is as follows: OMRON\Sysmac Studio \IODeviceProfiles\EsiFiles\UserEsiFiles.

If the xml file is saved under this path for the first time, the Sysmac Studio software must be restarted.

Setting the network connection attribute

- If the PC is connected to the controller through an USB, skip this step.
- If the PC is connected to the controller through Ethernet, set the TCP/IP attribute of the PC, as shown below.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 250 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

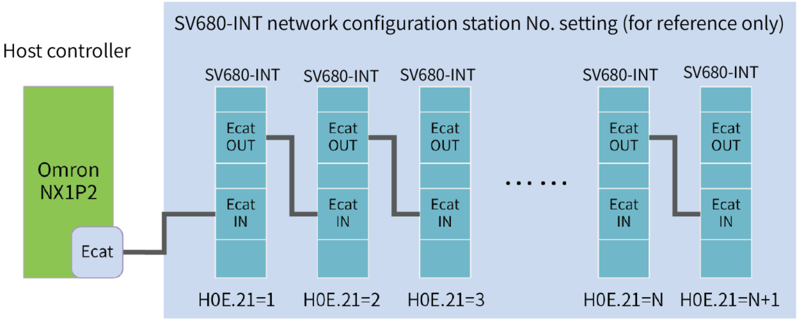
Configuring the servo drive

Recommended version:

Use MCU software version (H01.00) of 0900.1 or later for SV680N-INT series servo drives.

Use FPGA software version (H01.01) of 0902.1 or later for SV680N-INT series servo drives.

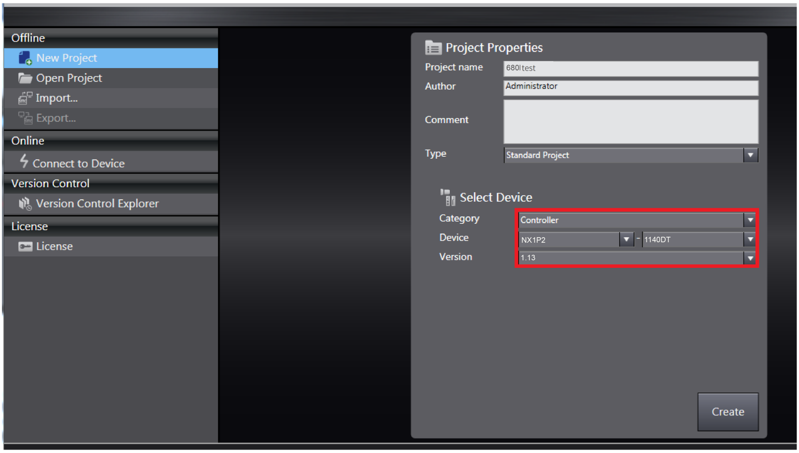
Pay attention to the setting of H0E.21.

<table>
<tr><th>Parameter</th><th>Name</th><th>Value range</th><th>Unit</th><th>Initial Value</th><th>Mode</th><th>Setting Condition</th><th>Effective Time</th><th>Value</th></tr>
<tr><td>H0E.21</td><td>EtherCAT slave alias</td><td>0-65535</td><td>-</td><td>0</td><td>-</td><td>Stop setting</td><td>At once</td><td>Non-zero</td></tr>
<tr><td colspan="9">When an Omron controller is used, set the EtherCAT communication station number in H0E.21. It is recommended to set the station number according to the actual connection sequence for the convenience of configuration management.

 </td></tr>
</table>

Create a project.

Device: Set a device according to the actual controller model.

Version: Use V1.09 or later versions. For NX1P2-1140DT, only V1.13 is supported.



Project Properties

Project name: 680Itest

Author: Administrator

Comment:

Type: Standard Project

Select Device

Category: Controller

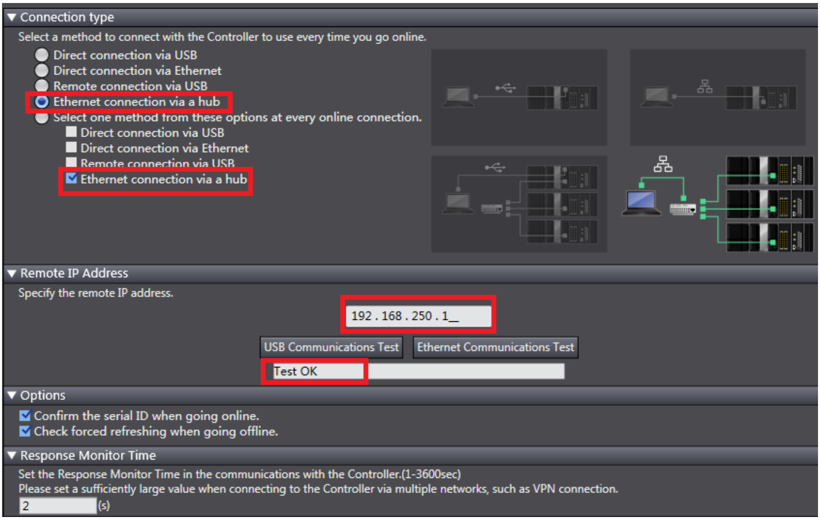
Device: NX1P2 - 1140DT

Version: 1.13

Communication setting

After entering the main interface, set the connection mode between the PC and the controller in Controller → Connection type.

- Select Remote connection via USB to perform USB Communication Test directly. If the test is succeeded, proceed to the next step.
- Select Ethernet connection via a hub, in this case, set the IP address to 192.168.250.1 (controlled by NX), and then perform Ethernet Communication Test. If the test is succeeded, proceed to the next step.



Connection type

Select a method to connect with the Controller to use every time you go online.

○ Direct connection via USB
○ Direct connection via Ethernet
○ Remote connection via USB
● Ethernet connection via a hub
○ Select one method from these options at every online connection.
☐ Direct connection via USB
☐ Direct connection via Ethernet
☐ Remote connection via USB
☑ Ethernet connection via a hub

Remote IP Address

Specify the remote IP address.

192 . 168 . 250 . 1_

Test OK

Options

☑ Confirm the serial ID when going online.
☑ Check forced refreshing when going offline.

Response Monitor Time

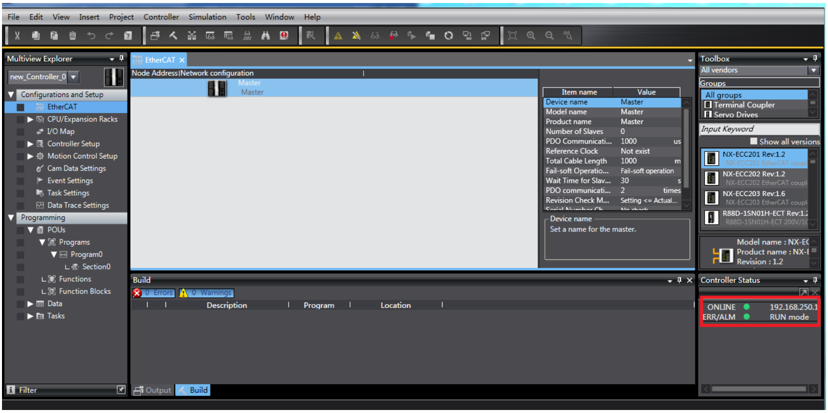
Set the Response Monitor Time in the communications with the Controller.(1-3600sec)
Please set a sufficiently large value when connecting to the Controller via multiple networks, such as VPN connection.

2 (s)

Scanning the device

Switch the controller to the online and running mode.

1. Check that the controller status in the lower right corner is online and running.



Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	0
PDO Communicati...	1000 us
Reference Clock	Not exist
Total Cable Length	1000 m
Fail-soft Operatio...	Fail-soft operation
Wait Time for Slav...	30 s
PDO communicati...	2 times
Revision Check M...	Setting <= Actual...

Device name
Set a name for the master.

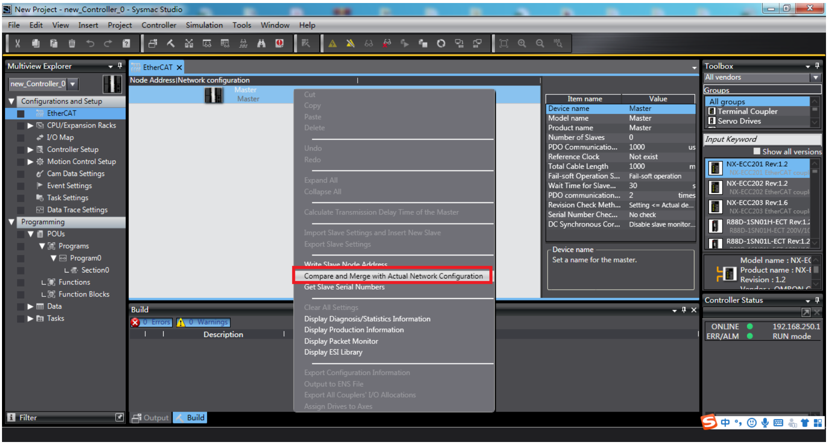
Model name : NX-EC
Product name : NX-E
Revision : 1.2

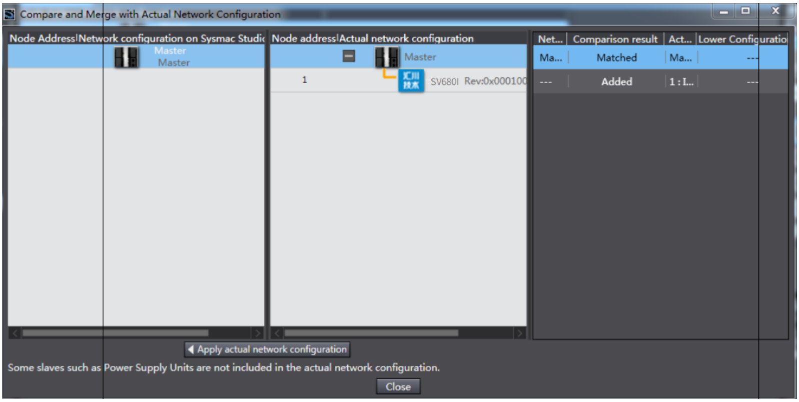
ONLINE ● 192.168.250.1
ERR/ALM ● RUN mode

2. A prompt window appears if it is a new controller.

3. Click Yes in the window. The name shown in the window is the project name. Scan the device and add the slave station.

Right click Configurations and Setup→EtherCAT→Master, and select Compare and Merge with Actual Network Configuration. The controller scans all the slaves in the network (an error will be reported if the station number is 0). After scanning, click Apply actual network configuration in the pop-up window to add the slave. You can view the added slave station in the main page.

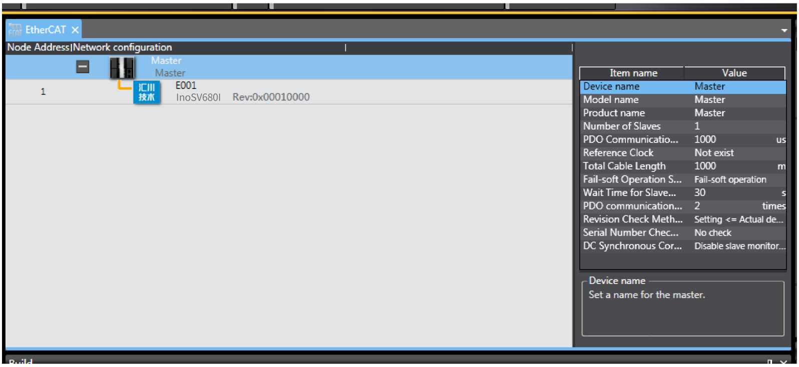

New Project - new_Controller_0 - Sysmac Studio
File Edit View Insert Project Controller Simulation Tools Window Help
Multiview Explorer
EtherCAT
Node Address|Network configuration
Cut
Copy
Paste
Delete
Undo
Redo
Expand All
Collapse All
Calculate Transmission Delay Time of the Master
Import Slave Settings and Insert New Slave
Export Slave Settings
Write Slave Node Address
Compare and Merge with Actual Network Configuration
Get Slave Serial Numbers
Clear All Settings
Display Diagnosis/Statistics Information
Display Production Information
Display Packet Monitor
Display ESI Library
Export Configuration Information
Output to ENS File
Export All Couplers' I/O Allocations
Assign Drives to Axes
Toolbox
Controller Status
ONLINE 192.168.250.1
ERR/ALM RUN mode


Compare and Merge with Actual Network Configuration

Net...	Comparison result	Act...	Lower Configuratio
Ma...	Matched	Ma...	---
---	Added	1 : I...	---

Node Address|Network configuration on Sysmac Studio
Master
Node address|Actual network configuration
Master
1
SV680I Rev:0x000100
Apply actual network configuration
Close

Some slaves such as Power Supply Units are not included in the actual network configuration.



Item name	Value
Device name	Master
Model name	Master
Product name	Master
Number of Slaves	1
PDO Communicatio...	1000 us
Reference Clock	Not exist
Total Cable Length	1000 m
Fail-soft Operation S...	Fail-soft operation
Wait Time for Slave...	30 s
PDO communication...	2 times
Revision Check Meth...	Setting <= Actual de...
Serial Number Chec...	No check
DC Synchronous Cor...	Disable slave monitor...

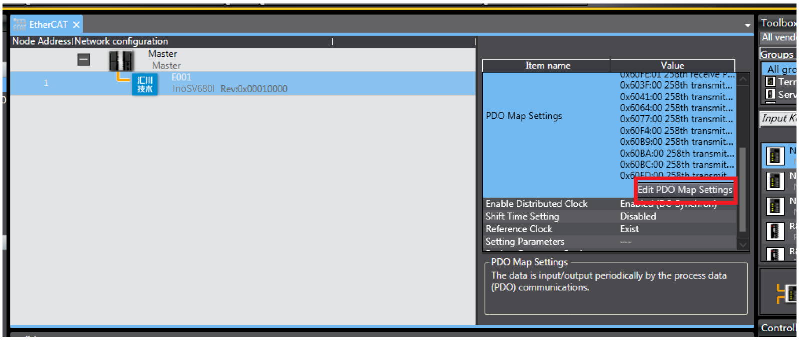
Device name
Set a name for the master.

Parameter settings

Switch the controller to the offline mode and set PDO mapping, axis parameters, and distributed clock.

Setting PDO mapping

1. Setting the PDO mapping.

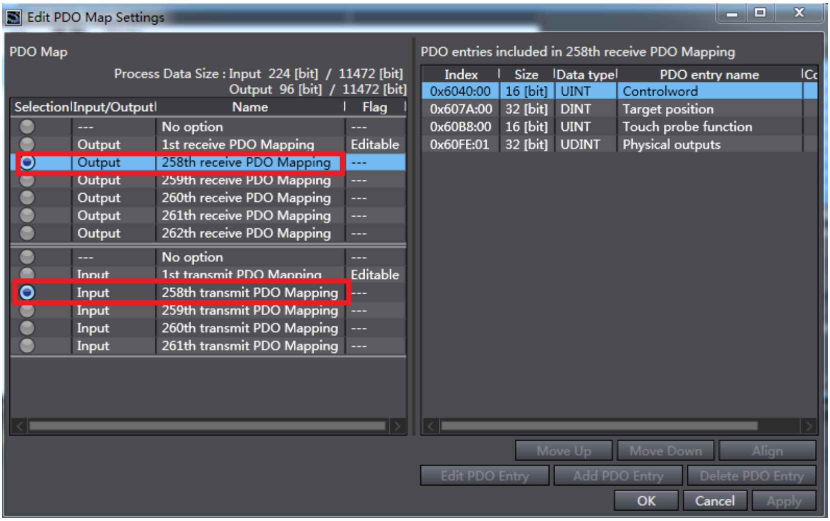


Item name	Value
PDO Map Settings	0x60FE:01 258th receive P... 0x603F:00 258th transmit... 0x6041:00 258th transmit... 0x6064:00 258th transmit... 0x6077:00 258th transmit... 0x60F4:00 258th transmit... 0x60B9:00 258th transmit... 0x60BA:00 258th transmit... 0x60BC:00 258th transmit... 0x60FD:00 258th transmit... Edit PDO Map Settings
Enable Distributed Clock	Enabled (DC Synchron)
Shift Time Setting	Disabled
Reference Clock	Exist
Setting Parameters	---

PDO Map Settings

The data is input/output periodically by the process data (PDO) communications.

2. Select the editable RPDO and TPDO provided by the drive for configuration.



Edit PDO Map Settings

PDO Map

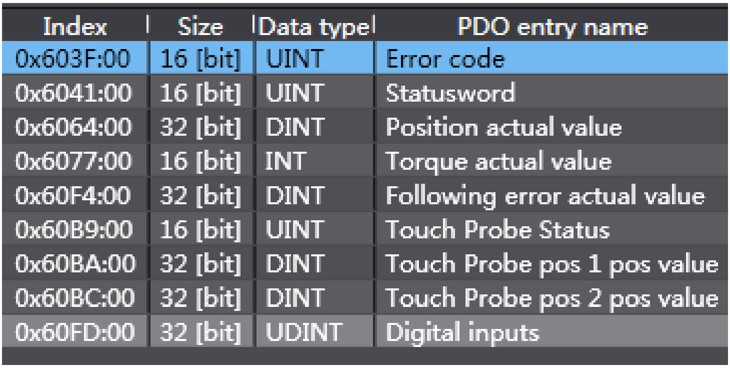
Process Data Size : Input 224 [bit] / 11472 [bit]
Output 96 [bit] / 11472 [bit]

Selection	Input/Output	Name	Flag
○	---	No option	---
○	Output	1st receive PDO Mapping	Editable
●	Output	258th receive PDO Mapping	---
○	Output	259th receive PDO Mapping	---
○	Output	260th receive PDO Mapping	---
○	Output	261th receive PDO Mapping	---
○	Output	262th receive PDO Mapping	---
○	---	No option	---
○	Input	1st transmit PDO Mapping	Editable
●	Input	258th transmit PDO Mapping	---
○	Input	259th transmit PDO Mapping	---
○	Input	260th transmit PDO Mapping	---
○	Input	261th transmit PDO Mapping	---

PDO entries included in 258th receive PDO Mapping

Index	Size	Data type	PDO entry name	Cc
0x6040:00	16 [bit]	UINT	Controlword	
0x607A:00	32 [bit]	DINT	Target position	
0x60B8:00	16 [bit]	UINT	Touch probe function	
0x60FE:01	32 [bit]	UDINT	Physical outputs	

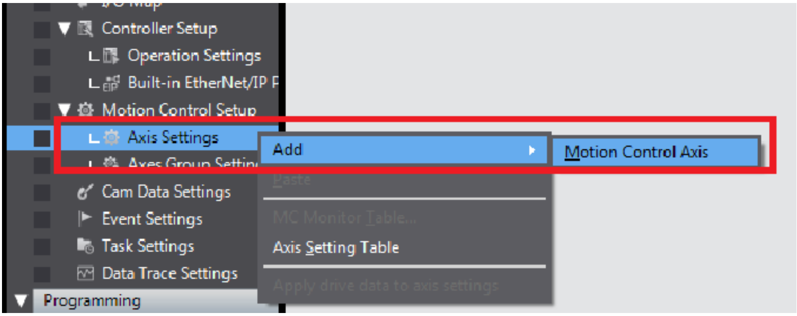
3. Modify the PDO mapping object through Add PDO Entry and Delete PDO Entry. The frequently used mapping parameters are shown in the following interface.



Index	Size	Data type	PDO entry name
0x603F:00	16 [bit]	UINT	Error code
0x6041:00	16 [bit]	UINT	Statusword
0x6064:00	32 [bit]	DINT	Position actual value
0x6077:00	16 [bit]	INT	Torque actual value
0x60F4:00	32 [bit]	DINT	Following error actual value
0x60B9:00	16 [bit]	UINT	Touch Probe Status
0x60BA:00	32 [bit]	DINT	Touch Probe pos 1 pos value
0x60BC:00	32 [bit]	DINT	Touch Probe pos 2 pos value
0x60FD:00	32 [bit]	UDINT	Digital inputs

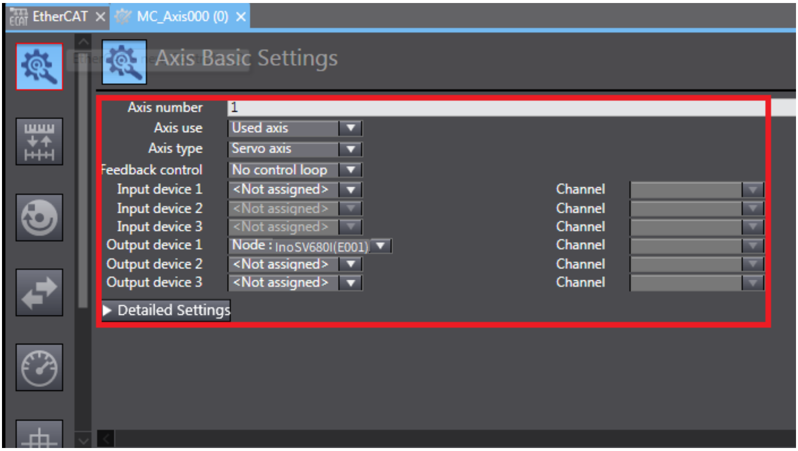
Setting axis parameters

1. Right click Motion Control Setup→Axis settings →Add→Motion Control Axis, as shown in the following interface.



2. MC_Axis000 can be renamed through a simple click. For example, if it is named as "Rewind axis", the axis variable "Rewind axis" used in the NX program represents control on this SV680N-INT servo axis.
3. Double-click **MC_Axis000** and configure an SV680N-INT device at the corresponding station in a corresponding **Axis Basic Settings** interface.

   a. Axis assignment



Axis Basic Settings

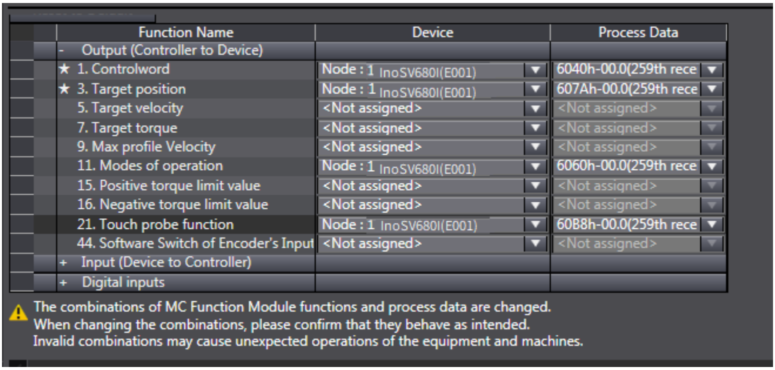
Axis number	1		
Axis use	Used axis		
Axis type	Servo axis		
Feedback control	No control loop		
Input device 1	<Not assigned>	Channel	
Input device 2	<Not assigned>	Channel	
Input device 3	<Not assigned>	Channel	
Output device 1	Node : InoSV680I(E001)	Channel	
Output device 2	<Not assigned>	Channel	
Output device 3	<Not assigned>	Channel	

▶ Detailed Settings

- Axis number: Represents the Ethernet communication station No. of the servo drive, which is also the value of H0E.21.
- Axis use: Represents the axis in use.
- Axis type: Represents the servo axis.
- Output device 1: Select the servo drive.

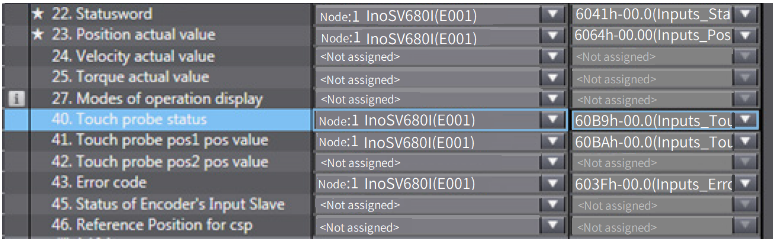
b. Detailed settings

- Select the PDO mapping objects according to step 8, which is to assign the output parameters (controller to device) and input parameters (device to controller). Note that the object name, node number, and index number must be set correctly. Each mapping object selected in step 8 must be assigned correctly. Otherwise, an error will be reported.



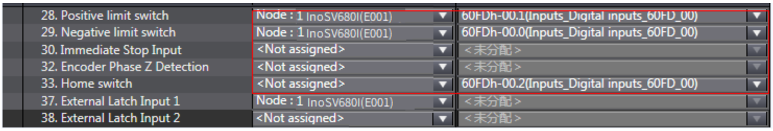
Function Name	Device	Process Data
- Output (Controller to Device)		
★ 1. Controlword	Node : 1 InoSV680I(E001)	6040h-00.0(259th rece
★ 3. Target position	Node : 1 InoSV680I(E001)	607Ah-00.0(259th rece
5. Target velocity	<Not assigned>	<Not assigned>
7. Target torque	<Not assigned>	<Not assigned>
9. Max profile Velocity	<Not assigned>	<Not assigned>
11. Modes of operation	Node : 1 InoSV680I(E001)	6060h-00.0(259th rece
15. Positive torque limit value	<Not assigned>	<Not assigned>
16. Negative torque limit value	<Not assigned>	<Not assigned>
21. Touch probe function	Node : 1 InoSV680I(E001)	60B8h-00.0(259th rece
44. Software Switch of Encoder's Input	<Not assigned>	<Not assigned>
+ Input (Device to Controller)		
+ Digital inputs		

⚠ The combinations of MC Function Module functions and process data are changed.
When changing the combinations, please confirm that they behave as intended.
Invalid combinations may cause unexpected operations of the equipment and machines.



★ 22. Statusword	Node:1 InoSV680I(E001)	6041h-00.0(Inputs_Sta
★ 23. Position actual value	Node:1 InoSV680I(E001)	6064h-00.00(Inputs_Pos
24. Velocity actual value	<Not assigned>	<Not assigned>
25. Torque actual value	<Not assigned>	<Not assigned>
27. Modes of operation display	<Not assigned>	<Not assigned>
40. Touch probe status	Node:1 InoSV680I(E001)	60B9h-00.0(Inputs_Tou
41. Touch probe pos1 pos value	Node:1 InoSV680I(E001)	60BAh-00.0(Inputs_Tou
42. Touch probe pos2 pos value	<Not assigned>	<Not assigned>
43. Error code	Node:1 InoSV680I(E001)	603Fh-00.0(Inputs_Erro
45. Status of Encoder's Input Slave	<Not assigned>	<Not assigned>
46. Reference Position for csp	<Not assigned>	<Not assigned>

- 60FDh must be mapped to the same as that in the Omron controller, as shown in the following interface. bit0...bit2 of SV680-INT indicate the negative position limit, positive position limit, and the home respectively. bit16...bit20 indicate the status of DI1...DI5.



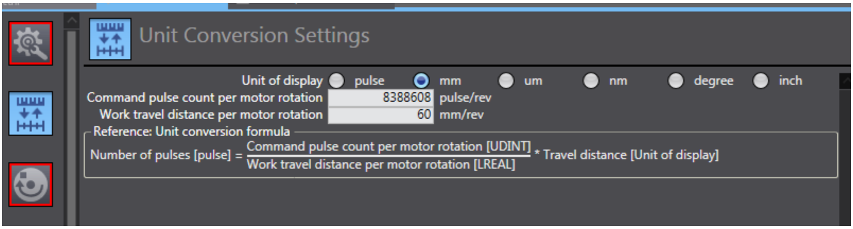
28. Positive limit switch	Node : 1 InoSV680I(E001)	60FDh-00.1(Inputs_Digital inputs_60FD_00)
29. Negative limit switch	Node : 1 InoSV680I(E001)	60FDh-00.0(Inputs_Digital inputs_60FD_00)
30. Immediate Stop Input	<Not assigned>	<未分配>
32. Encoder Phase Z Detection	<Not assigned>	<未分配>
33. Home switch	<Not assigned>	60FDh-00.2(Inputs_Digital inputs_60FD_00)
37. External Latch Input 1	Node : 1 InoSV680I(E001)	<未分配>
38. External Latch Input 2	<Not assigned>	<未分配>

Note

The Omron software tool only allows you to configure axes for the SV680N-INT series manually.

Setting unit conversion

Set Command pulse count per motor rotation based on the resolution of the motor encoder (example: 67108864 PPR for motor equipped with 26-bit encoder). For the convenience of commissioning, set the **Work travel distance per motor rotation** to 60 mm/rev, indicating 1 mm/s equals to 1 RPM of the motor.



Unit Conversion Settings

Unit of display ○ pulse ● mm ○ um ○ nm ○ degree ○ inch

Command pulse count per motor rotation 8388608 pulse/rev

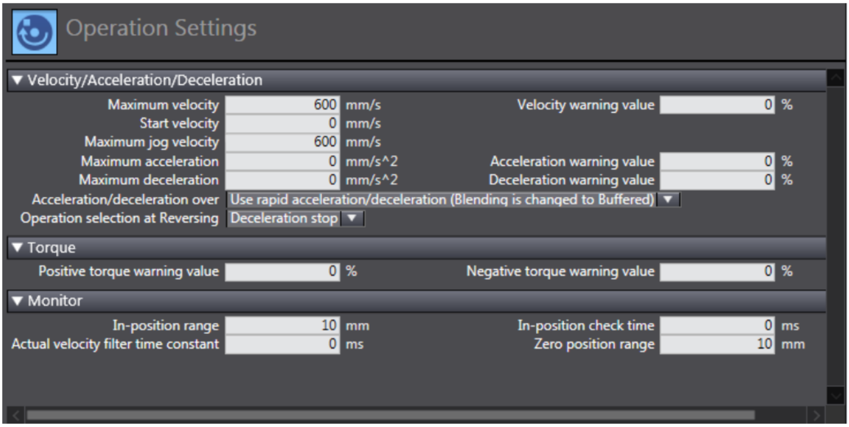
Work travel distance per motor rotation 60 mm/rev

Reference: Unit conversion formula

$$\text{Number of pulses [pulse]} = \frac{\text{Command pulse count per motor rotation [UDINT]}}{\text{Work travel distance per motor rotation [LREAL]}} * \text{Travel distance [Unit of display]}$$

Select the Display Unit based on the actual running unit when setting the gear ratio. All the position-type parameters in the host controller will be displayed in this unit.

Operation settings



Operation Settings

▼ Velocity/Acceleration/Deceleration

Maximum velocity 600 mm/s

Start velocity 0 mm/s

Maximum jog velocity 600 mm/s

Maximum acceleration 0 mm/s^2

Maximum deceleration 0 mm/s^2

Acceleration/deceleration over: Use rapid acceleration/deceleration (Blending is changed to Buffered)

Operation selection at Reversing: Deceleration stop

Velocity warning value 0 %

Acceleration warning value 0 %

Deceleration warning value 0 %

▼ Torque

Positive torque warning value 0 %

Negative torque warning value 0 %

▼ Monitor

In-position range 10 mm

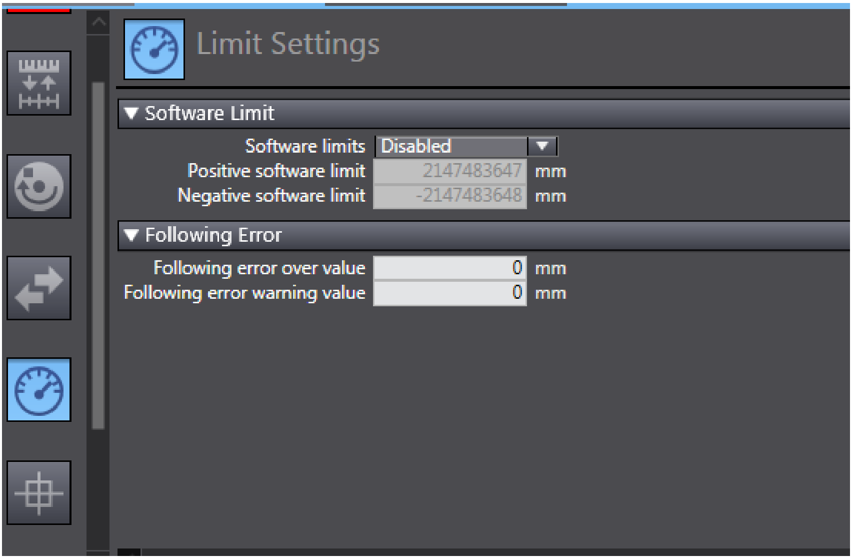
Actual velocity filter time constant 0 ms

In-position check time 0 ms

Zero position range 10 mm

- Velocity/Acceleration/Deceleration: Set the maximum speed of the load (if the motor speed converted exceeds 1900 RPM, a parameter setting error) which is marked by a red box, will be reported by the host controller software) according to actual conditions. If the acceleration/deceleration rate is 0, the motion profile will be generated based on the maximum acceleration/deceleration rate (there is no need to set the acceleration/deceleration rate in general cases).
- Torque: If the warning value is 0, no warning will be reported. There is no need to set the warning value in general cases.
- Monitor: Set Positioning Range and Zero Position Range based on actual motor and mechanical conditions. If the set value is too small, positioning or homing may not be completed.

Position limit



Limit Settings

Software Limit

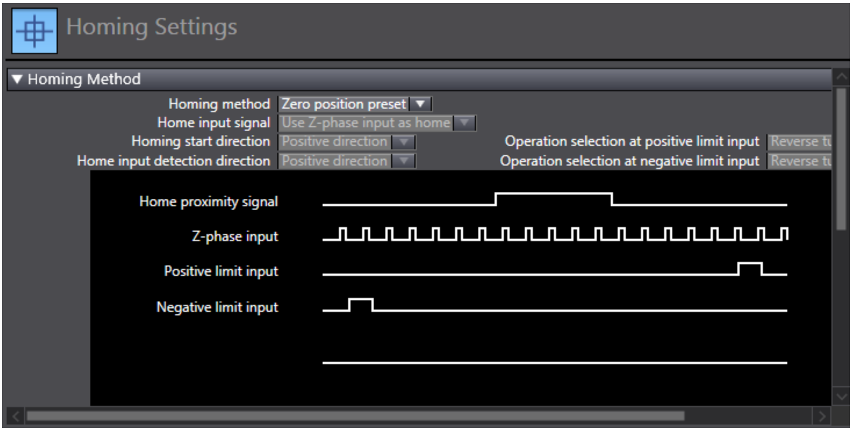
Software limits: Disabled
Positive software limit: 2147483647 mm
Negative software limit: -2147483648 mm

Following Error

Following error over value: 0 mm
Following error warning value: 0 mm

You can use the function of software position limit. The software position limit will be activated after homing.

Homing



Homing Settings

Homing Method

Homing method: Zero position preset
Home input signal: Use Z-phase input as home
Homing start direction: Positive direction
Home input detection direction: Positive direction
Operation selection at positive limit input: Reverse tu
Operation selection at negative limit input: Reverse tu

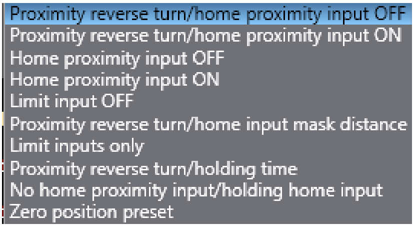
The homing mode involves cooperation between the servo drive and host controller. Set the homing mode according to the following table.

Description of NX Software	Servo Drive Function	Terminal Configuration
Home proximity signal	Home switch (FunIN.31)	-
Positive limit input	P-OT (FunIN.14)	DI1
Negative limit input	N-OT (FunIN.15)	DI2

Select the homing mode of the host controller and set the homing speed, acceleration, and home offset based on actual mechanical conditions.

- Introduction to homing
  Function block: MC_Home and MC_HomeWithParameter

1. Set MC_Home in the preceding figure and MC_HomeWithParameter in the function block.
2. The two function blocks both include 10 types of homing modes.

MC_Home	MC_HomeWithParameter
 Proximity reverse turn/home proximity input OFF Proximity reverse turn/home proximity input ON Home proximity input OFF Home proximity input ON Limit input OFF Proximity reverse turn/home input mask distance Limit inputs only Proximity reverse turn/holding time No home proximity input/holding home input Zero position preset	Designate the homing action to be modified. 0: Proximity reverse turn/home proximity input OFF 1: Proximity reverse turn/home proximity input ON 4: Home proximity input OFF 5: Home proximity input ON 8: Limit input OFF 9: Proximity reverse turn/home input mask distance 11: Limit inputs only 12: Proximity reverse turn/holding time 13: No home proximity input/holding home input 14: Zero position preset

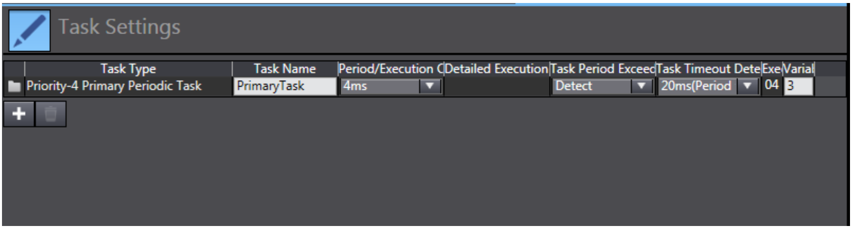
- Home proximity input OFF: The search for the home signal starts after the falling edge of the home proximity switch is reached.
- Home proximity input ON: The search for the home signal starts after the rising edge of the home proximity switch is reached.
- Proximity reverse turn: The home proximity signal is ON when homing starts, and reverse running applies after the falling edge of the home proximity signal is reached.
- Home input mask distance: The home signal is masked by the host controller within the set distance after receiving the homing signal (for example, edge change of home proximity signal), and the home signal is received only after the set distance passes.
- Holding time: The home signal is masked by the host controller within the set period of time after receiving the homing signal (for example, edge change of home proximity signal), and home signal is received only after the set period of time elapses.
- Zero position preset: The home offset is being written to the position reference/ position feedback in the host controller with current position as the home and motor at a standstill.

Note

In all the homing modes, the home signal is searched at low speed. In case of operations at high speed, the home signal is hidden during decelerating from high speed to low speed.

Distributed clock

The default clock is 1 ms. The synchronization clock (cycle of primary fixed-cycle tasks) named "PDO communication cycle" can be modified in Task Settings. The modification will be activated after switching to the online status at next power-on.

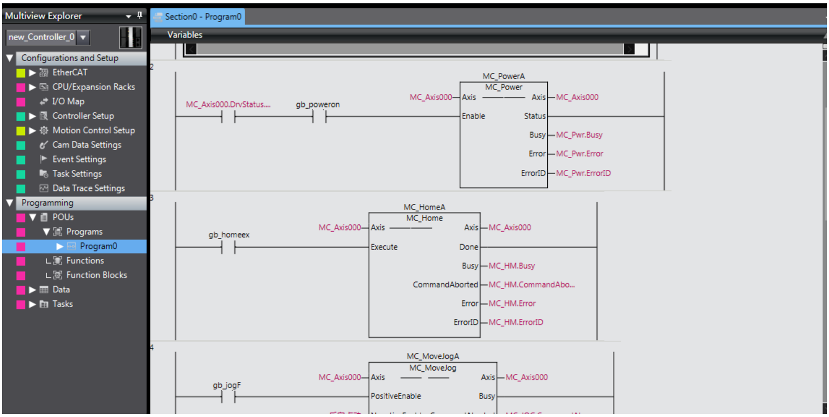


Task Settings

Task Type	Task Name	Period/Execution C	Detailed Execution	Task Period Exceed	Task Timeout Dete	Exe	Varial
Priority-4 Primary Periodic Task	PrimaryTask	4ms		Detect	20ms(Period	04	3

Program-controlled servo operations

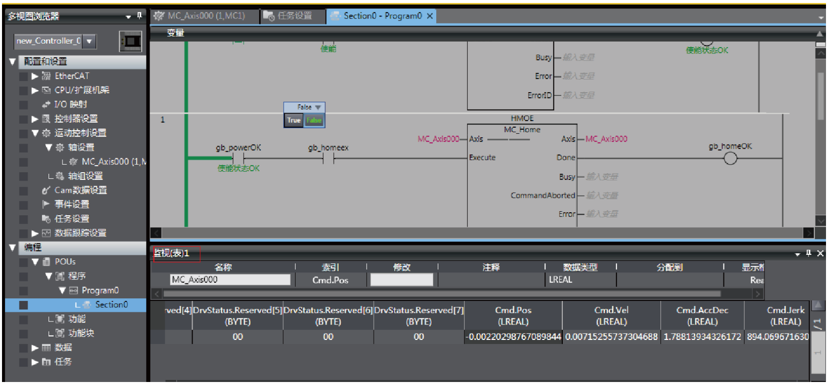
1. After configurations are done, you can control the servo operations through the PLC program.
   If the MC_POWER module is used, it is recommended to add the servo status bit MC_Axis000.DrvStatus. Ready (MC_Axis000 is the axis name). Where MC_Axis000 is the axis name. This is to prevent the situation where the PLC program is running but the communication configuration is not done.



2. After all the settings and programming are done, switch to the online state, and click to download the program to the controller.

Click to use the synchronization function. This function serves to compare the difference between the current program and the program in the controller, allowing users to determine whether to download the program to the controller, upload it from the controller "" or leave it unchanged based on the differences.

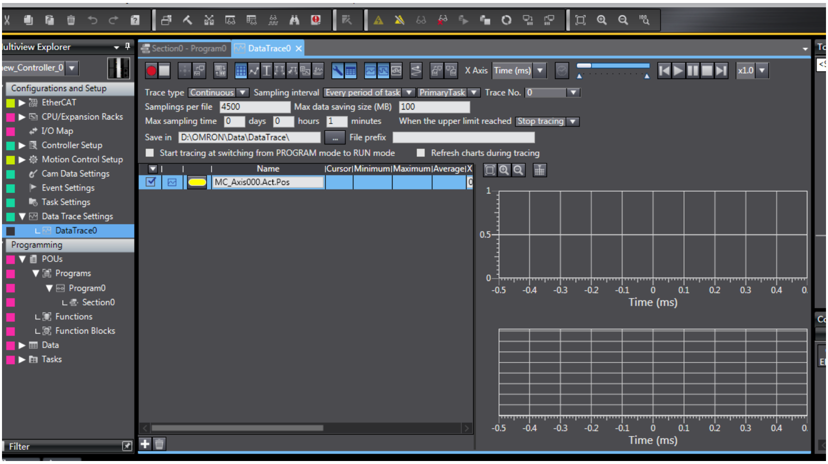
You can monitor the data through the monitoring list or collect the data waveform by using the data tracking function during operation.



监视(表)1

名称	索引	修改	注释	数据类型	分配到	显示格
MC_Axis000	Cmd.Pos			LREAL		Rea

ved[4]	DrvStatus.Reserved[5] (BYTE)	DrvStatus.Reserved[6] (BYTE)	DrvStatus.Reserved[7] (BYTE)	Cmd.Pos (LREAL)	Cmd.Vel (LREAL)	Cmd.AccDec (LREAL)	Cmd.Jerk (LREAL)
	00	00	00	-0.00220298767089844	0.00715255737304688	1.78813934326172	894.069671630



Trace type: Continuous; Sampling interval: Every period of task, PrimaryTask; Trace No.: 0

Samplings per file: 4500; Max data saving size (MB): 100

Max sampling time: 0 days 0 hours 1 minutes; When the upper limit reached: Stop tracing

Save in: D:\OMRON\Data\DataTrace\ ...; File prefix:

□ Start tracing at switching from PROGRAM mode to RUN mode □ Refresh charts during tracing

			Name	Cursor	Minimum	Maximum	Average	X
☑			MC_Axis000.Act.Pos					0

5.3.3 SV680N-INT and Beckhoff Controller

This section describes how to configure the SV680N-INT servo drive for working with Beckhoff TwinCAT3.

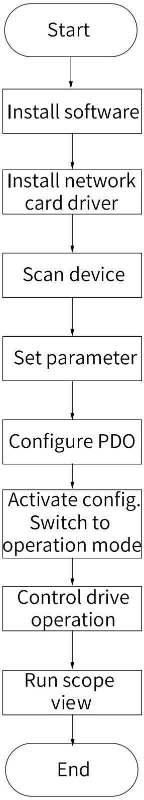


Figure 5-4 Configuration flowchart

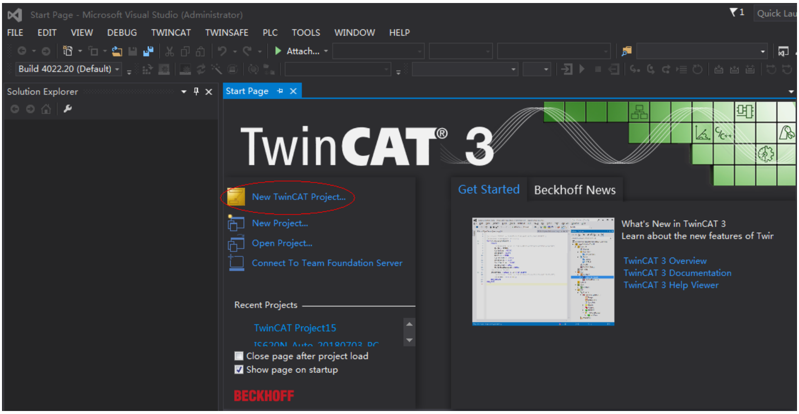
Installing the TwinCAT software

The TwinCAT3 software, which supports Windows7 32-bit or 64-bit systems, can be downloaded from the official website of Beckhoff.

Note

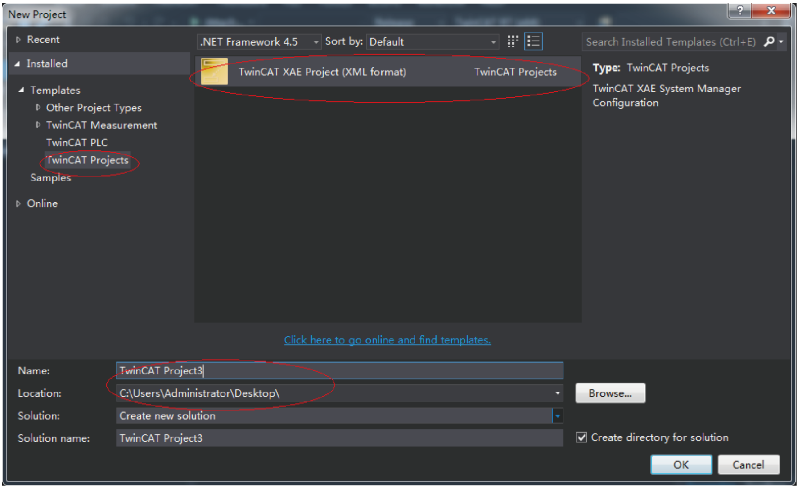
The Ethernet card must be 100 M Ethernet card equipped with Intel chip. If other brands are used, the EtherCAT operation may fail.

1. Copy the SV680N-INT EtherCAT configuration file (SV680_1Axis_V0.04-0506) to the TwinCAT installation directory: TwinCAT\3.1\Config\Io\EtherCAT.
2. Open TwinCAT3 and create a New Twincat3 Project.



□ Close page after project load

☑ Show page on startup



Type: TwinCAT Projects

TwinCAT XAE System Manager Configuration

Name: TwinCAT Project3

Location: C:\Users\Administrator\Desktop\

Solution: Create new solution

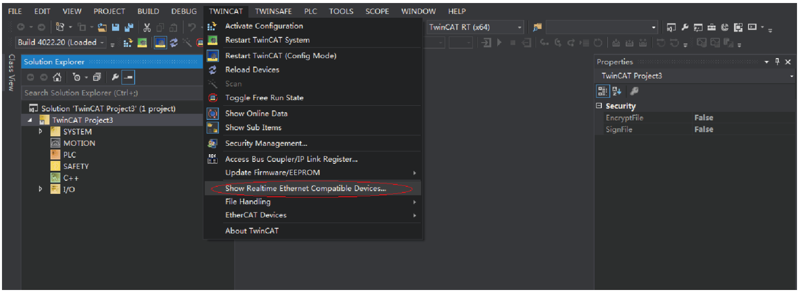
Solution name: TwinCAT Project3

☑ Create directory for solution

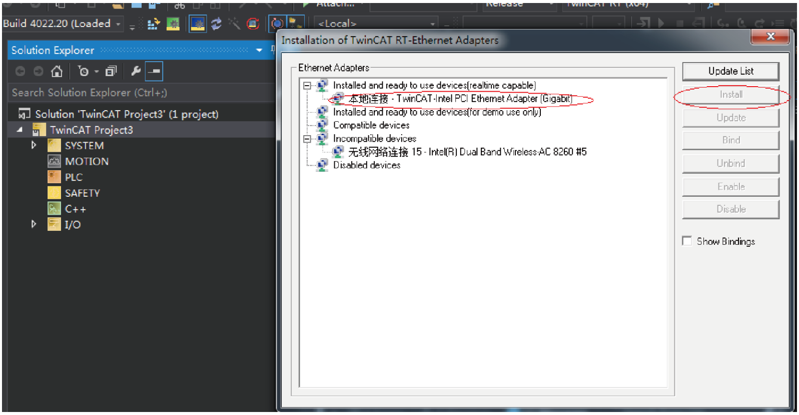
Installing the network adapter driver

Install the TwinCAT network adapter driver.

1. Open Show Real Time Ethernet Compatible Devices… in the menu shown in the preceding figure to display the following dialog box. Select local connection under Incompatible devices, and click Install.



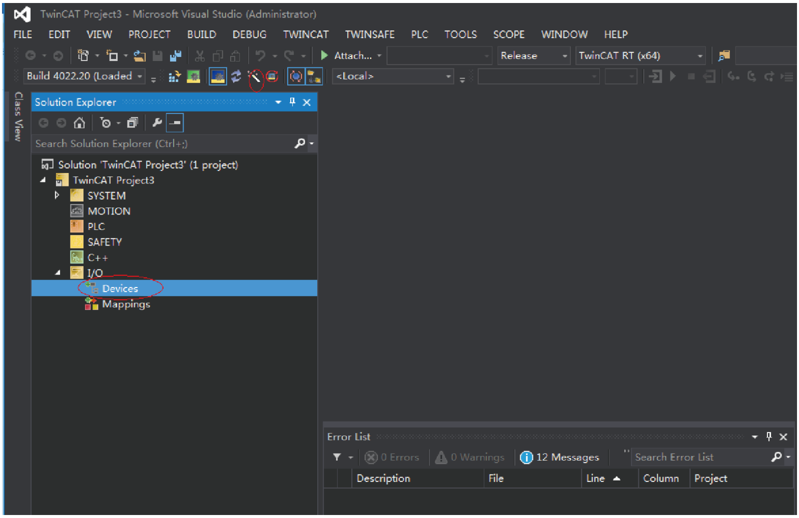
2. After installation is done, the network adapter installed will be displayed under Installed and ready to use devices(realtime capable).


Build 4022.20 (Loaded
<Local>
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'TwinCAT Project3' (1 project)
TwinCAT Project3
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Installation of TwinCAT RT-Ethernet Adapters
Ethernet Adapters
Installed and ready to use devices(realtime capable)
本地连接 - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
Installed and ready to use devices(for demo use only)
Compatible devices
Incompatible devices
无线网络连接 15 - Intel(R) Dual Band Wireless-AC 8260 #5
Disabled devices
Update List
Install
Update
Bind
Unbind
Enable
Disable
□ Show Bindings

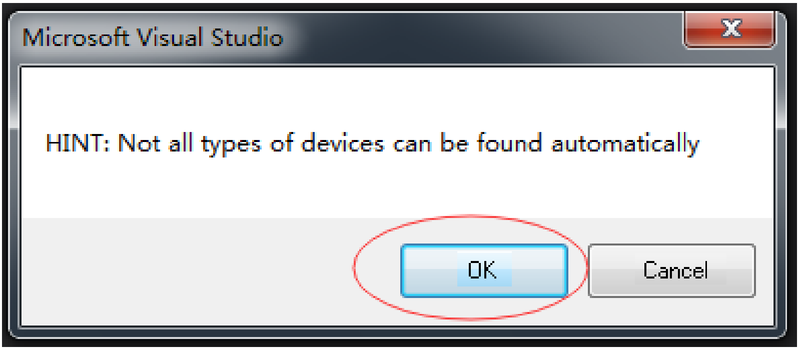
Search for devices.

1. Create a project and start searching for devices. Select Devices, and click as shown below.

See the following figure:

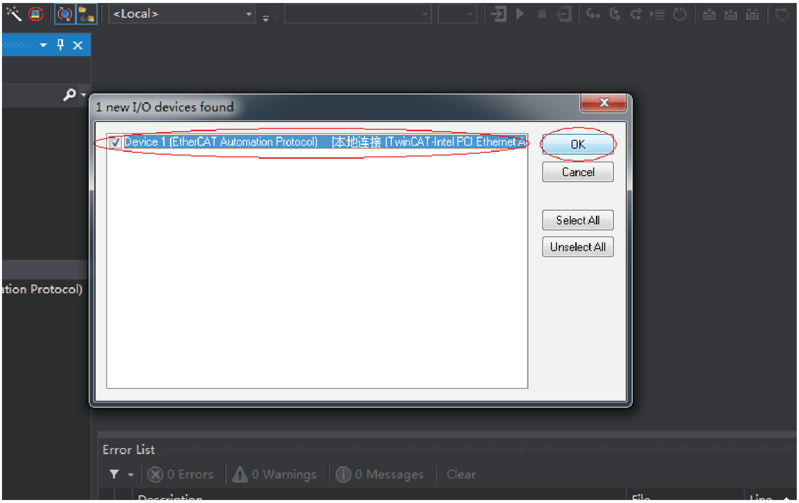


2. Click **OK**.


Microsoft Visual Studio

HINT: Not all types of devices can be found automatically

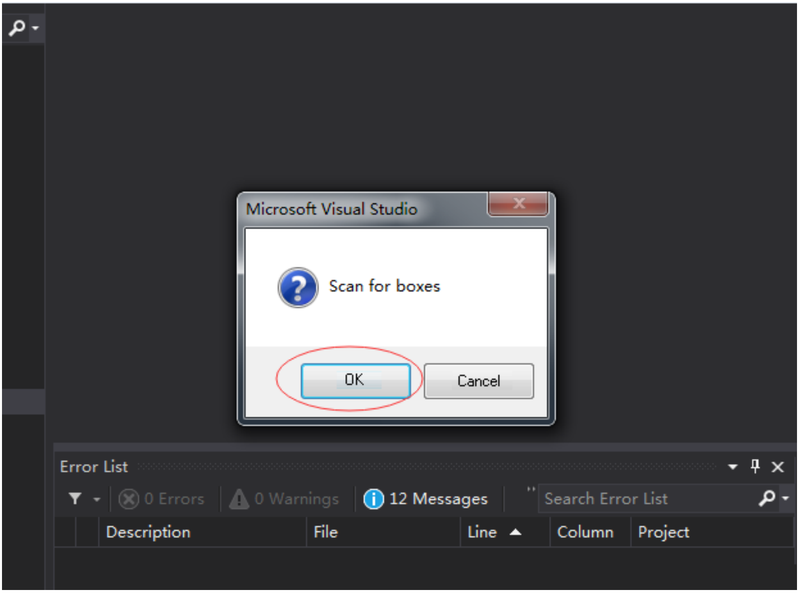
3. Click **OK**.



1 new I/O devices found

☑ Device 1 (EtherCAT Automation Protocol) [本地连接 (TwinCAT-Intel PCI Ethernet A

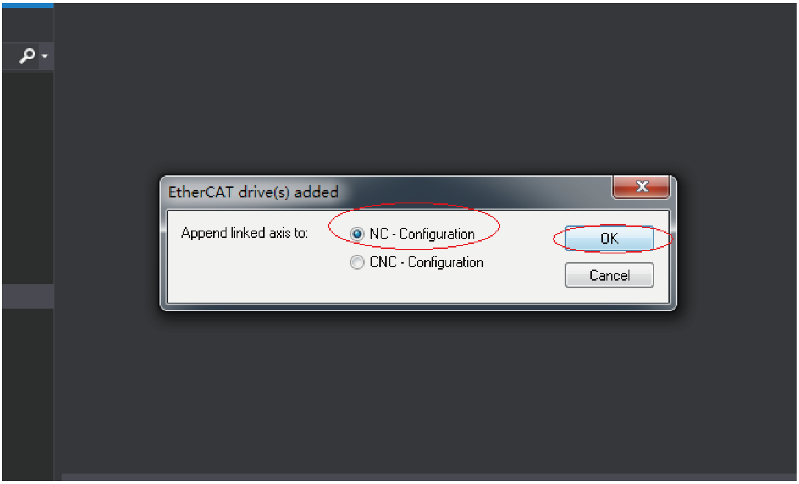
4. Click **OK**.



Microsoft Visual Studio

Scan for boxes

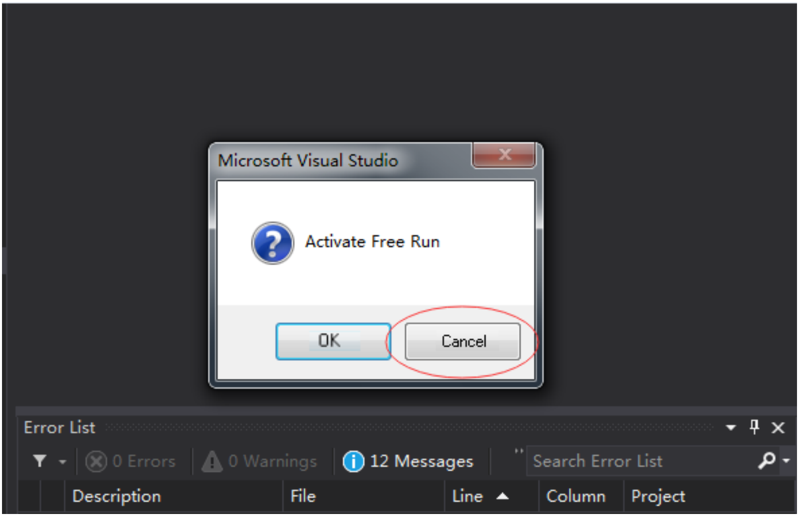
5. Click **OK**.



EtherCAT drive(s) added

Append linked axis to: ● NC - Configuration ○ CNC - Configuration

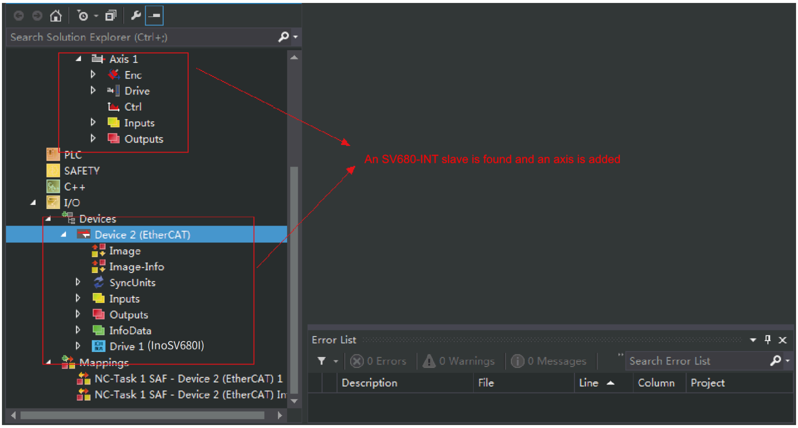
6. Click **Cancel**.



Microsoft Visual Studio

Activate Free Run

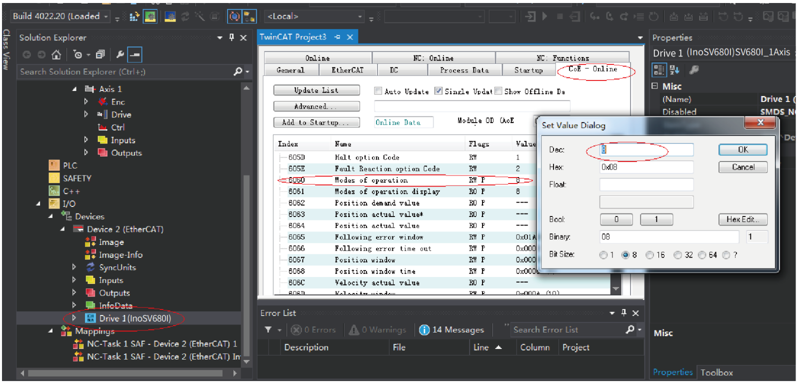
7. The search for the device is done, as shown below.



An SV680-INT slave is found and an axis is added

Configuring servo drive parameters

Configure parameters through SDO communication in CoE - Online interface. When H0E.01(200E-02h) is set to 3, the parameter values modified through SDO communication will be saved upon power failure. To modify 6060h to the CSP mode (8), follow the procedure shown in the following figure.



Index	Name	Flags	Value
605D	Halt option Code	RW	1
605E	Fault Reaction option Code	RW	2
6060	Modes of operation	RW P	8
6061	Modes of operation display	RO P	8
6062	Position demand value	RO P	---
6063	Position actual value*	RO P	---
6064	Position actual value	RO P	---
6065	Following error window	RW P	0x01A
6066	Following error time out	RW P	0x000
6067	Position window	RW P	0x000
6068	Position window time	RW P	0x000
606C	Velocity actual value	RO P	---
606D	Velocity window	RW P	0x000A (10)

Dec: 8
Hex: 0x08
Float:
Bool: 0 1
Binary: 08 1
Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

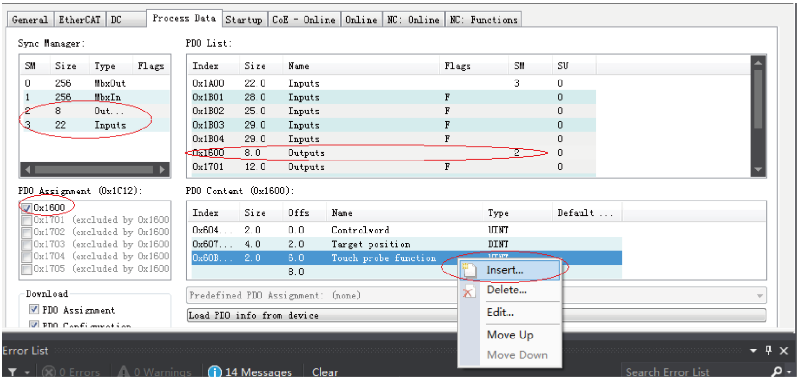
Note

This operation is available only when H02.00 (Control mode) is set to 9 (EtherCAT mode).

Configuring PDO

Select 0x1600 and 0x1A00 as shown in the following figure. Change the current PDO only if it does not fulfill your needs. To modify the PDO, right-click on the PDO

Content window, click Delete to delete the redundant PDO or click Insert to add the PDO needed.



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	8	Out...	
3	22	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	22.0	Inputs		3	0
0x1B01	28.0	Inputs	F		0
0x1B02	25.0	Inputs	F		0
0x1B03	29.0	Inputs	F		0
0x1B04	29.0	Inputs	F		0
0x1600	8.0	Outputs		2	0
0x1701	12.0	Outputs	F		0

PDO Assignment (0x1C12):

☑ 0x1600
□ 0x1701 (excluded by 0x1600
□ 0x1702 (excluded by 0x1600
□ 0x1703 (excluded by 0x1600
□ 0x1704 (excluded by 0x1600
□ 0x1705 (excluded by 0x1600

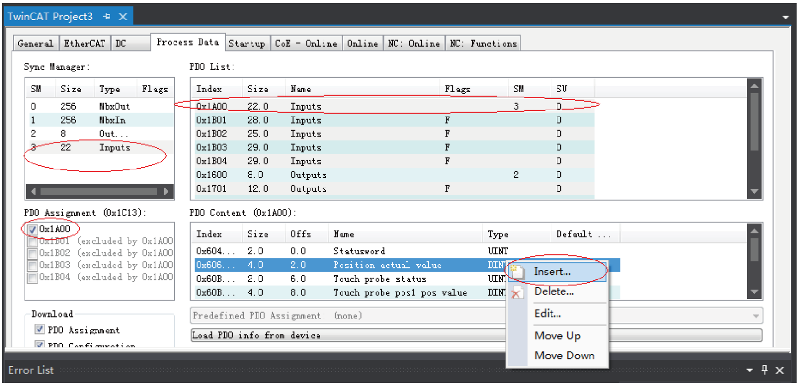
PDO Content (0x1600):

Index	Size	Offs	Name	Type	Default ...
0x604...	2.0	0.0	Controlword	UINT	
0x607...	4.0	2.0	Target position	DINT	
0x60B...	2.0	6.0	Touch probe function	UINT	
		8.0			

Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignment: (none)

Load PDO info from device



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	8	Out...	
3	22	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	22.0	Inputs		3	0
0x1B01	28.0	Inputs	F		0
0x1B02	25.0	Inputs	F		0
0x1B03	29.0	Inputs	F		0
0x1B04	29.0	Inputs	F		0
0x1600	8.0	Outputs		2	0
0x1701	12.0	Outputs	F		0

PDO Assignment (0x1C13):

☑ 0x1A00
□ 0x1B01 (excluded by 0x1A00
□ 0x1B02 (excluded by 0x1A00
□ 0x1B03 (excluded by 0x1A00
□ 0x1B04 (excluded by 0x1A00

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default ...
0x604...	2.0	0.0	Statusword	UINT	
0x606...	4.0	2.0	Position actual value	DINT	
0x60B...	2.0	6.0	Touch probe status	UINT	
0x60B...	4.0	8.0	Touch probe pos1 pos value	DINT	

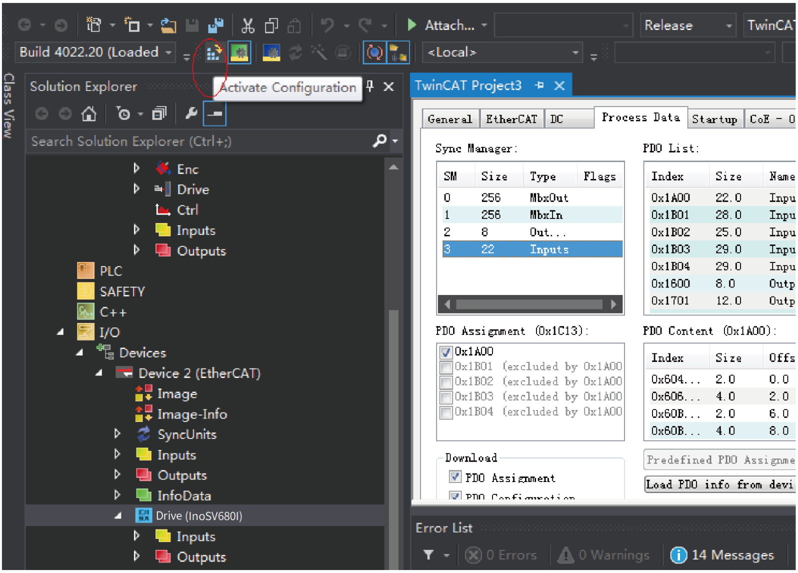
Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignment: (none)

Load PDO info from device

Activate the configuration and switch to the operation mode.

1. Click .



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	8	Out...	
3	22	Inputs	

PDO List:

Index	Size	Name
0x1A00	22.0	Inpu
0x1B01	28.0	Inpu
0x1B02	25.0	Inpu
0x1B03	29.0	Inpu
0x1B04	29.0	Inpu
0x1600	8.0	Outp
0x1701	12.0	Outp

PDO Assignment (0x1C13):

☑ 0x1A00
□ 0x1B01 (excluded by 0x1A00
□ 0x1B02 (excluded by 0x1A00
□ 0x1B03 (excluded by 0x1A00
□ 0x1B04 (excluded by 0x1A00

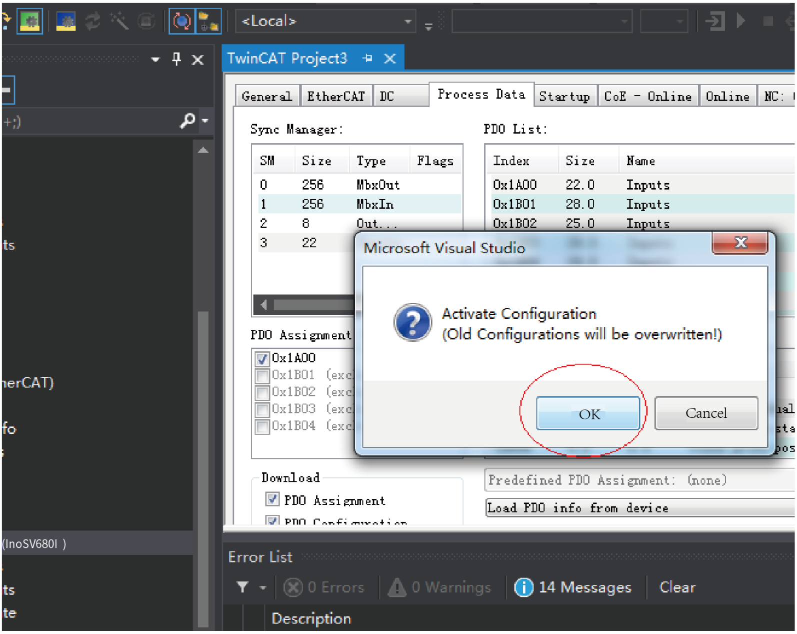
PDO Content (0x1A00):

Index	Size	Offs
0x604...	2.0	0.0
0x606...	4.0	2.0
0x60B...	2.0	6.0
0x60B...	4.0	8.0

Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignme
Load PDO info from devi

2. Click **OK**.



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	8	Out...	
3	22		

PDO List:

Index	Size	Name
0x1A00	22.0	Inputs
0x1B01	28.0	Inputs
0x1B02	25.0	Inputs

PDO Assignment

☑ 0x1A00
□ 0x1B01 (exc
□ 0x1B02 (exc
□ 0x1B03 (exc
□ 0x1B04 (exc

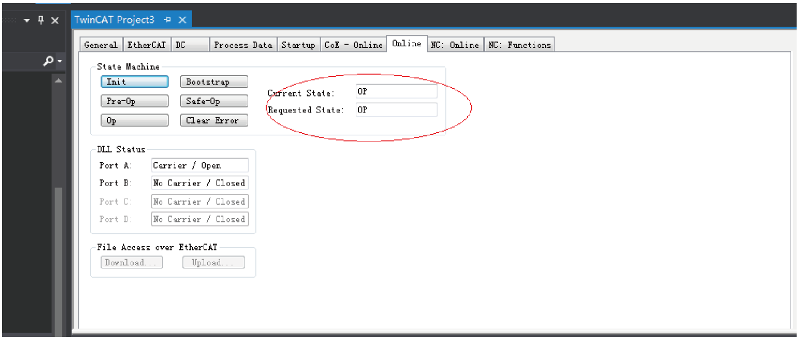
Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignment: (none)
Load PDO info from device

Microsoft Visual Studio

Activate Configuration
(Old Configurations will be overwritten!)

3. After you click OK, the device enters OP status as shown in the Online interface. Meanwhile, the 3rd LED on the keypad displays "8", and the keypad displays _88RY.



Current State: OP

Requested State: OP

DLL Status

Port A:	Carrier / Open
Port B:	No Carrier / Closed
Port C:	No Carrier / Closed
Port D:	No Carrier / Closed

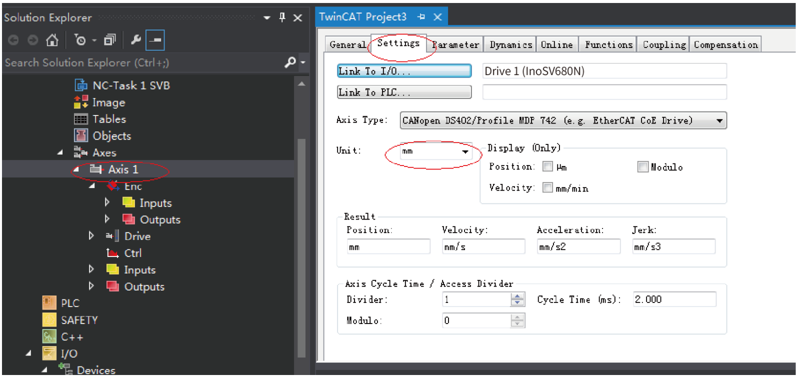
Controlling servo drive operation

Control the servo drive through NC or PLC programs.

1. For operating in CSP mode

a. Set the unit.

Set the unit to "mm" during the test.



Link To I/O...: Drive 1 (InoSV680N)

Axis Type: CANopen DS402/Profile MDP 742 (e.g. EtherCAT CoE Drive)

Unit: mm

Display (Only)

Position: □ µm □ Modulo

Velocity: □ mm/min

Result

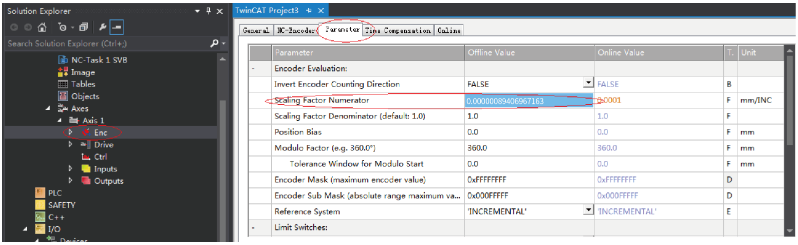
Position:	Velocity:	Acceleration:	Jerk:
mm	mm/s	mm/s2	mm/s3

Axis Cycle Time / Access Divider

Divider: 1 Cycle Time (ms): 2.000

Modulo: 0

b. Set the scaling factor.

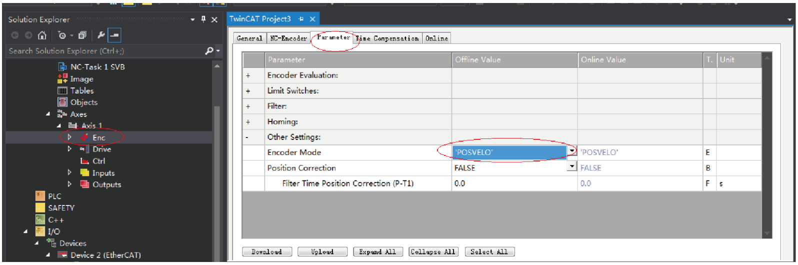


	Parameter	Offline Value	Online Value	T.	Unit
-	Encoder Evaluation:				
	Invert Encoder Counting Direction	FALSE	FALSE	B	
	Scaling Factor Numerator	0.000000089406967163	0.0001	F	mm/INC
	Scaling Factor Denominator (default: 1.0)	1.0	1.0	F	
	Position Bias	0.0	0.0	F	mm
	Modulo Factor (e.g. 360.0°)	360.0	360.0	F	mm
	Tolerance Window for Modulo Start	0.0	0.0	F	mm
	Encoder Mask (maximum encoder value)	0xFFFFFFFF	0xFFFFFFFF	D	
	Encoder Sub Mask (absolute range maximum va...	0x000FFFFF	0x000FFFFF	D	
	Reference System	'INCREMENTAL'	'INCREMENTAL'	E	
-	Limit Switches:				

Scaling factor: Indicates the distance corresponding to the encoder pulses per position feedback.

For example, 67108864 PPR corresponds to a distance of 60 mm, and the scaling factor is: 60/67108864 = 0.00000089406967163 mm/Inc.

c. Set the encoder feedback mode to PosVelo.



	Parameter	Offline Value	Online Value	T.	Unit
+	Encoder Evaluation:				
+	Limit Switches:				
+	Filter:				
+	Homing:				
-	Other Settings:				
	Encoder Mode	'POSVELO'	'POSVELO'	E	
	Position Correction	FALSE	FALSE	B	
	Filter Time Position Correction (P-T1)	0.0	0.0	F	s

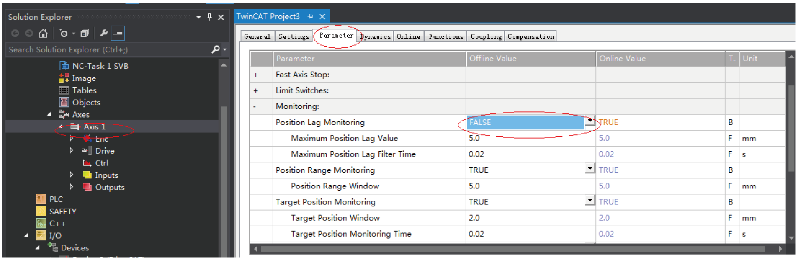
Descriptions for Other Settings:

Encoder mode: There are three encoder modes: POS, POSVELO, and POSVELOACC.

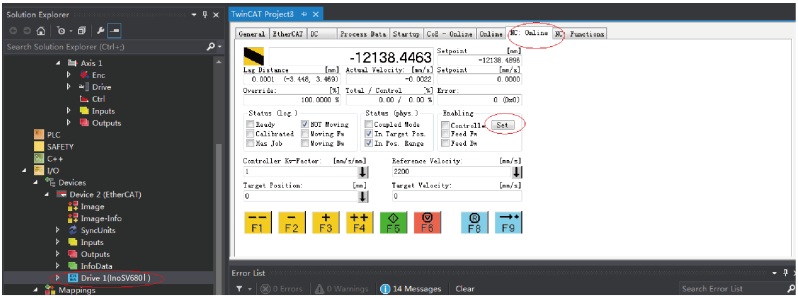
- POS: The encoder only calculates the position, which is used when the position loop is in the servo drive.
- POSVELO: The encoder only calculates the position, which is used when the position loop is in TWinCAT NC.
- POSVELOACC: The TWinCAT NC uses the encoder to determine the position, speed, and acceleration.

d. Jogging test

Hide the system deviation temporarily.



	Parameter	Offline Value	Online Value	T.	Unit
+	Fast Axis Stop:				
+	Limit Switches:				
-	Monitoring:				
	Position Lag Monitoring	FALSE	TRUE	B	
	Maximum Position Lag Value	5.0	5.0	F	mm
	Maximum Position Lag Filter Time	0.02	0.02	F	s
	Position Range Monitoring	TRUE	TRUE	B	
	Position Range Window	5.0	5.0	F	mm
	Target Position Monitoring	TRUE	TRUE	B	
	Target Position Window	2.0	2.0	F	mm
	Target Position Monitoring Time	0.02	0.02	F	s



-12138.4463

Setpoint [mm]: -12138.4896

Lag Distance [mm]: 0.0001 (-3.448, 3.469)

Actual Velocity: [mm/s]: -0.0022

Setpoint [mm/s]: 0.0000

Override: [%]: 100.0000 %

Total / Control [%]: 0.00 / 0.00 %

Error: 0 (0x0)

Status (log.)
☐ Ready
☐ Calibrated
☐ Has Job
☑ NOT Moving
☐ Moving Fw
☐ Moving Bw

Status (phys.)
☐ Coupled Mode
☑ In Target Pos.
☑ In Pos. Range

Enabling
☐ Controller
☐ Feed Fw
☐ Feed Bw

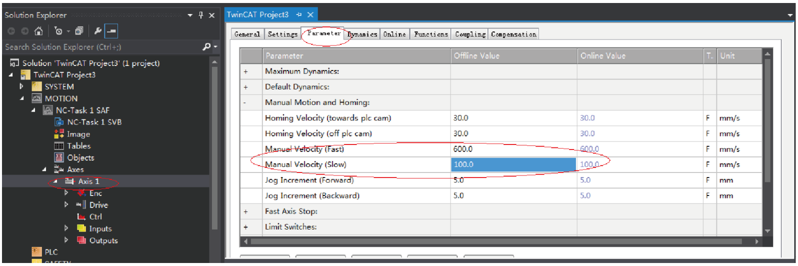
Controller Kv-Factor: [mm/s/mm]: 1

Reference Velocity: [mm/s]: 2200

Target Position: [mm]: 0

Target Velocity: [mm/s]: 0

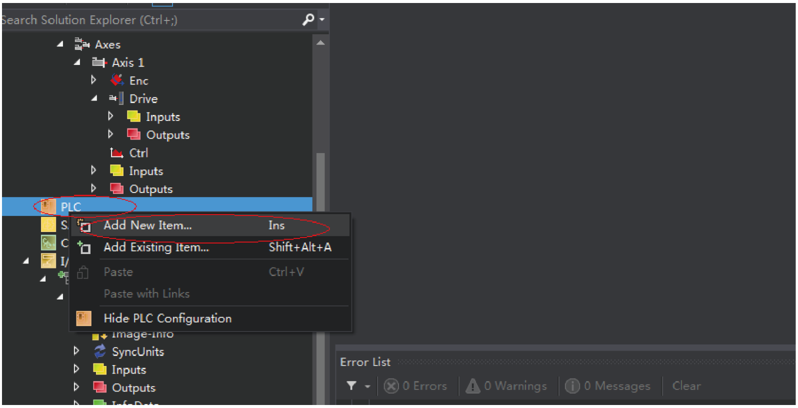
Click Set to display a dialog box and then click All to enable the servo drive. Perform jogging through F1 to F4. The jog speed is set as follows.

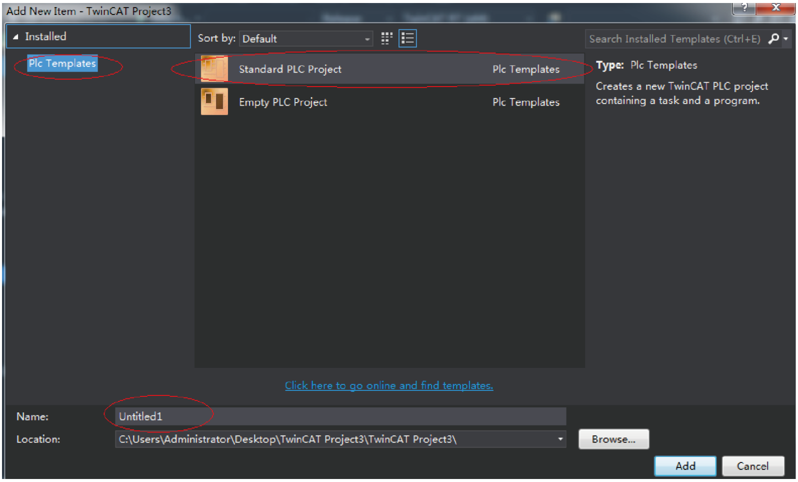


	Parameter	Offline Value	Online Value	T.	Unit
+	Maximum Dynamics:				
+	Default Dynamics:				
-	Manual Motion and Homing:				
	Homing Velocity (towards plc cam)	30.0	30.0	F	mm/s
	Homing Velocity (off plc cam)	30.0	30.0	F	mm/s
	Manual Velocity (Fast)	600.0	600.0	F	mm/s
	Manual Velocity (Slow)	100.0	100.0	F	mm/s
	Jog Increment (Forward)	5.0	5.0	F	mm
	Jog Increment (Backward)	5.0	5.0	F	mm
+	Fast Axis Stop:				
+	Limit Switches:				

2. Controlling the servo drive operations through the PLC

a. Create a PLC program.





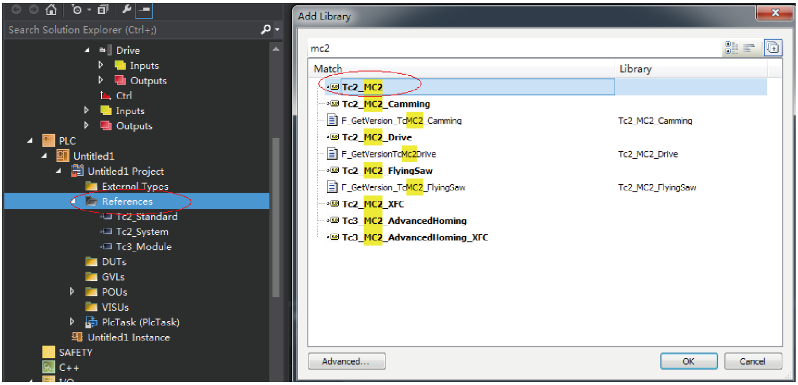
Type: Plc Templates

Creates a new TwinCAT PLC project containing a task and a program.

Name: Untitled1

Location: C:\Users\Administrator\Desktop\TwinCAT Project3\TwinCAT Project3\

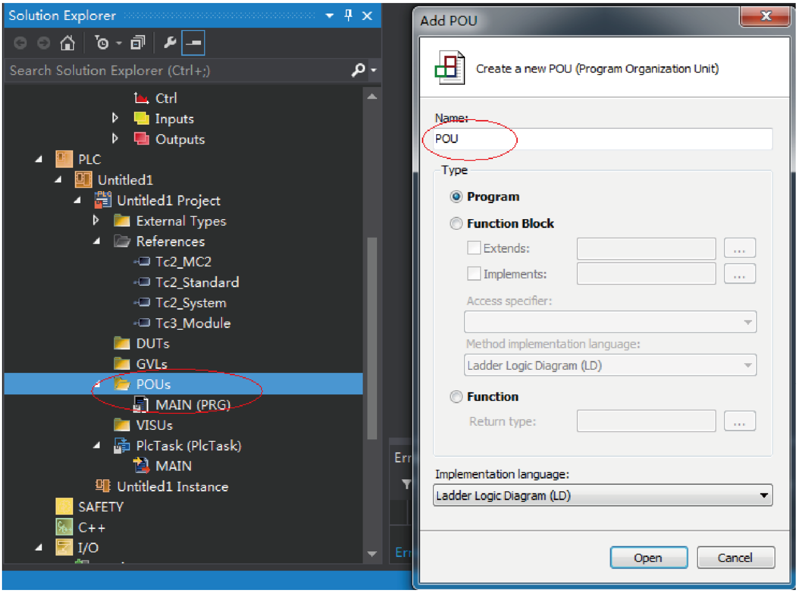
b. Add a motion control library for calling the motion control function blocks.



mc2

Match	Library
Tc2_MC2	
Tc2_MC2_Camming	
F_GetVersion_TcMC2_Camming	Tc2_MC2_Camming
Tc2_MC2_Drive	
F_GetVersionTcMc2Drive	Tc2_MC2_Drive
Tc2_MC2_FlyingSaw	
F_GetVersion_TcMC2_FlyingSaw	Tc2_MC2_FlyingSaw
Tc2_MC2_XFC	
Tc3_MC2_AdvancedHoming	
Tc3_MC2_AdvancedHoming_XFC	

c. Create a POU program.



Create a new POU (Program Organization Unit)

Name: POU

Type

● Program

○ Function Block

□ Extends:

□ Implements:

Access specifier:

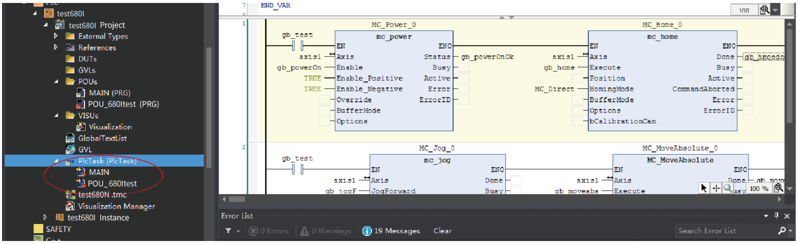
Method implementation language: Ladder Logic Diagram (LD)

○ Function

Return type:

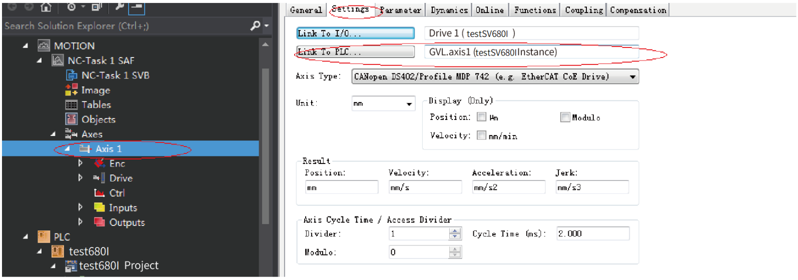
Implementation language: Ladder Logic Diagram (LD)

d. Call the motion module to implement some simple actions and input the final program to PLCtask.



```
END_VAR
```

e. Link the axis to the variable defined in the PLC.



Link To I/O... Drive 1 (testSV680I)

Link To PLC... GVL.axis1 (testSV680IInstance)

Axis Type: CANopen DS402/Profile MDP 742 (e.g. EtherCAT CoE Drive)

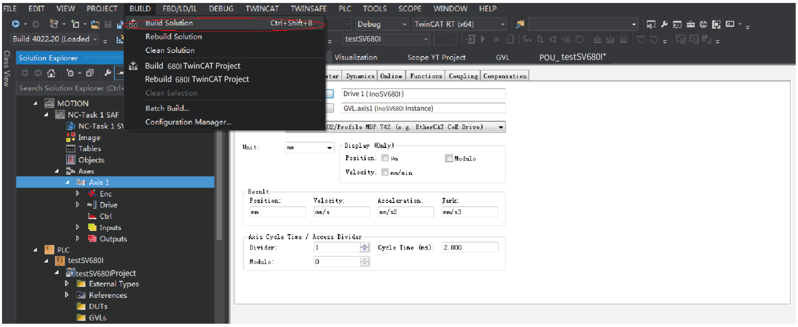
Unit: mm

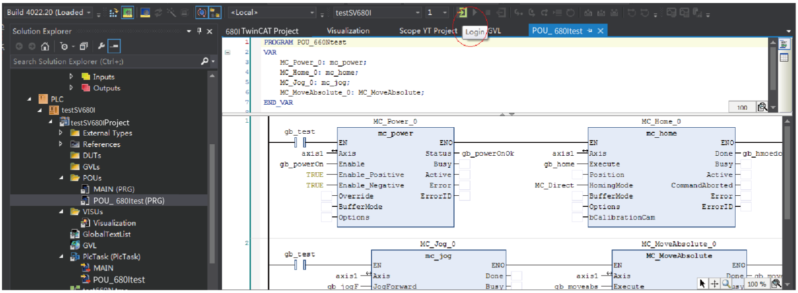
Display (Only)
Position: □ µm □ Modulo
Velocity: □ mm/min

Result
Position: mm Velocity: mm/s Acceleration: mm/s2 Jerk: mm/s3

Axis Cycle Time / Access Divider
Divider: 1 Cycle Time (ms): 2.000
Modulo: 0

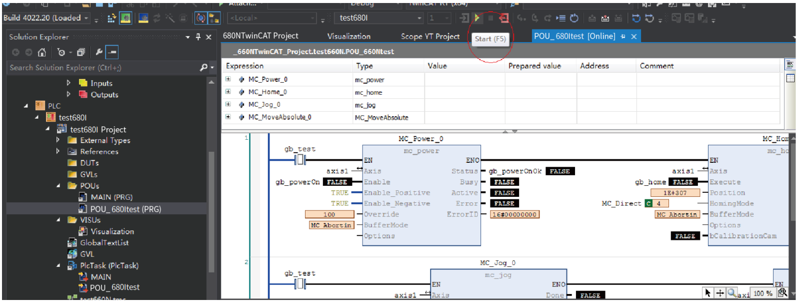
f. Compile the program. If there is not fault, activate the configuration and log onto the PLC.





```
PROGRAM POU_680Ntest
VAR
    MC_Power_0: mc_power;
    MC_Home_0: mc_home;
    MC_Jog_0: mc_jog;
    MC_MoveAbsolute_0: MC_MoveAbsolute;
END_VAR
```

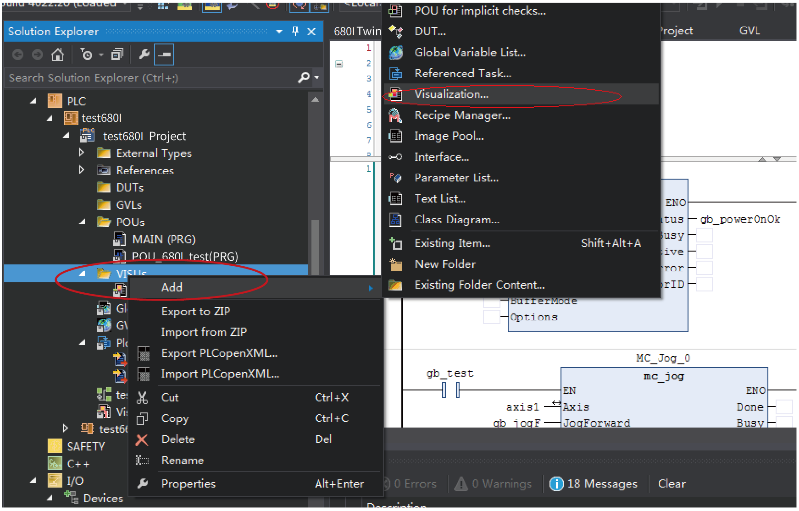
g. Click Start to make the servo drive run.

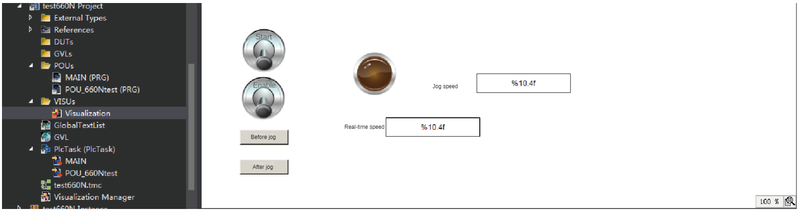


Expression	Type	Value	Prepared value	Address	Comment
MC_Power_0	mc_power				
MC_Home_0	mc_home				
MC_Jog_0	mc_jog				
MC_MoveAbsolute_0	MC_MoveAbsolute				

3. Controlling the servo drive operations through the HMI

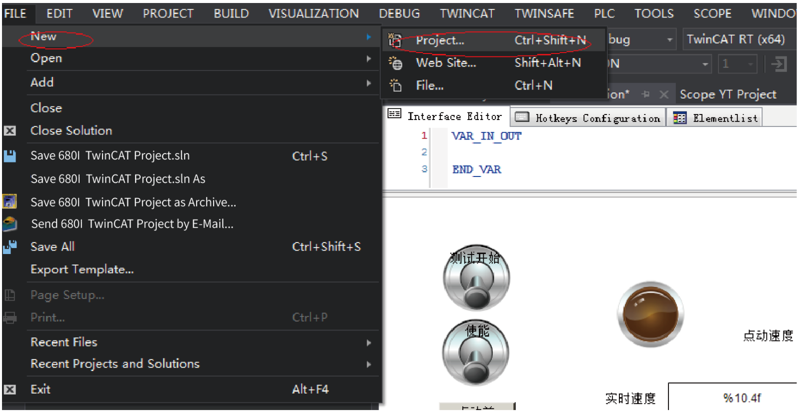
Add the HMI interface to control the servo drive through the HMI interface.



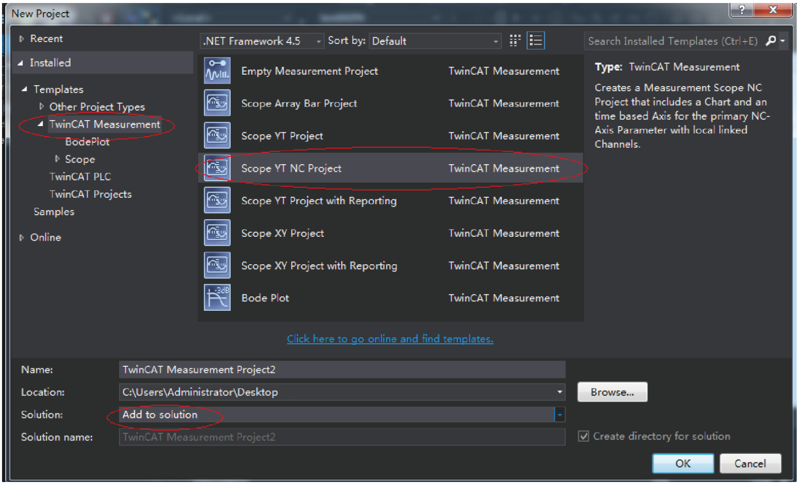


Use the scope view function.

1. Add a scope view project as shown in the following figure.



```
VAR_IN_OUT

END_VAR
```


New Project
Recent
Installed
Templates
Other Project Types
TwinCAT Measurement
BodePlot
Scope
TwinCAT PLC
TwinCAT Projects
Samples
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates (Ctrl+E)
Empty Measurement Project TwinCAT Measurement
Scope Array Bar Project TwinCAT Measurement
Scope YT Project TwinCAT Measurement
Scope YT NC Project TwinCAT Measurement
Scope YT Project with Reporting TwinCAT Measurement
Scope XY Project TwinCAT Measurement
Scope XY Project with Reporting TwinCAT Measurement
Bode Plot TwinCAT Measurement
Click here to go online and find templates.
Browse...
OK
Cancel

Type: TwinCAT Measurement

Creates a Measurement Scope NC Project that includes a Chart and an time based Axis for the primary NC-Axis Parameter with local linked Channels.

Name: TwinCAT Measurement Project2

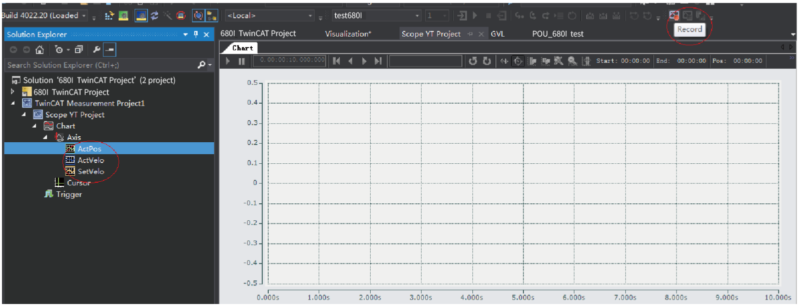
Location: C:\Users\Administrator\Desktop

Solution: Add to solution

Solution name: TwinCAT Measurement Project2

☑ Create directory for solution

2. Add parameters to be monitored and monitor these parameters during operation of the PLC.



5.3.4 SV680N-INT and KEYENCE KV7500 Controller

5.3.4.1 Configuring the Servo Drive

- Servo drive version
  It is recommended to use the device description file "SV680N-INT-Ecat_v0.09.xml" or above for trial run of SV680N-INT series servo drives. It is recommended to use the MCU software of version 901.4 (H01.00 = 901.4) or later for the drive.

- Related Parameters
  The definition of 60FDh of the SV680N-INT series differs from that of IS620N: bit0: negative limit; bit1: positive limit; bit2: home switch; bit16...bit20 correspond to DI1...DI5 respectively.

5.3.4.2 Configuring KEYENCE KV7500 Software Tool

As software tool versions earlier than KV STUDIO 9.45 do not support extension of KEYENCE EtherCAT module "KV-XH16EC", the version of the KEYENCE software tool used must be KV STUDIO 9.45 or later.

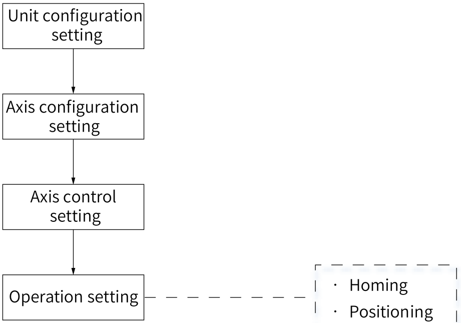
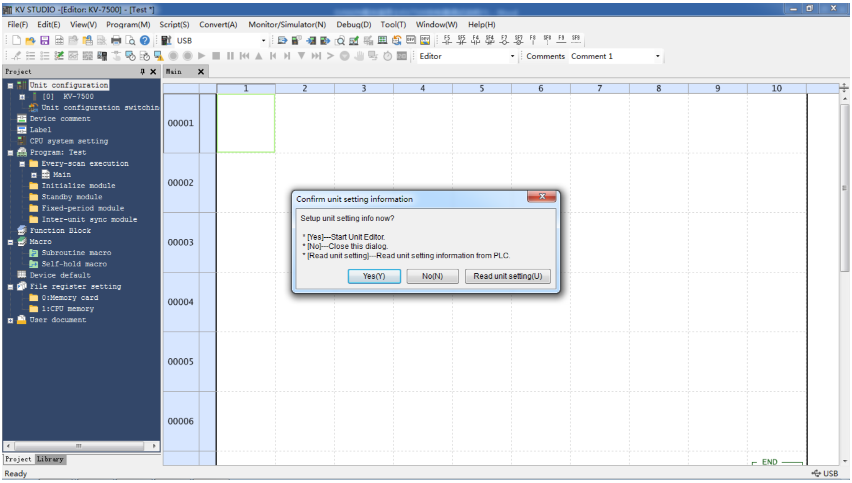


Figure 5-5 Configuration flowchart

Unit configuration setting

Create a project and click OK to display the following window.



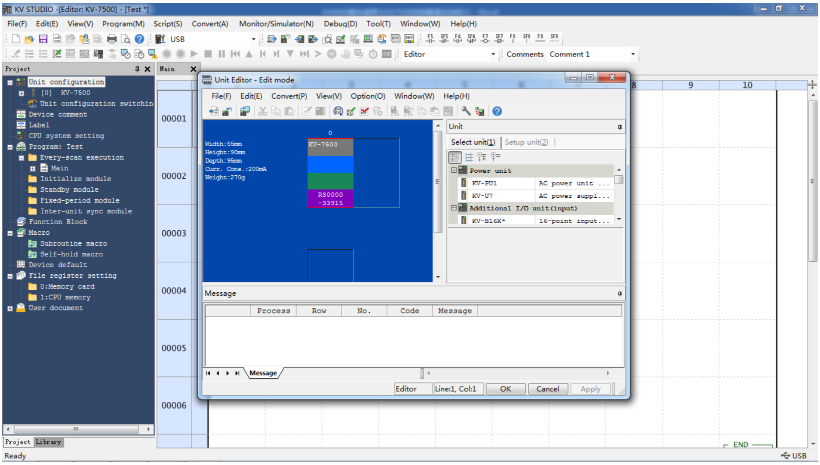
Confirm unit setting information

Setup unit setting info now?

* [Yes]---Start Unit Editor.
* [No]---Close this dialog.
* [Read unit setting]---Read unit setting information from PLC.

Click Yes, No, or Read unit setting as needed.

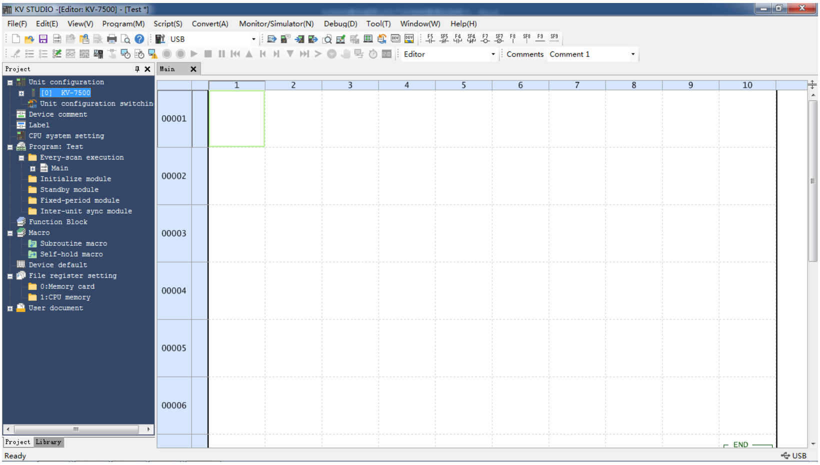
- Click Read unit setting when the physical PLC unit is connected properly and able to communicate with the software tool. The software tool obtains unit configurations automatically according to the physical connection.
- If you click Yes, the Unit editor window opens, allowing you to select units for configuration through dragging or double-clicking.



Width:55mm
Height:90mm
Depth:95mm
Curr. Cons.:200mA
Weight:270g

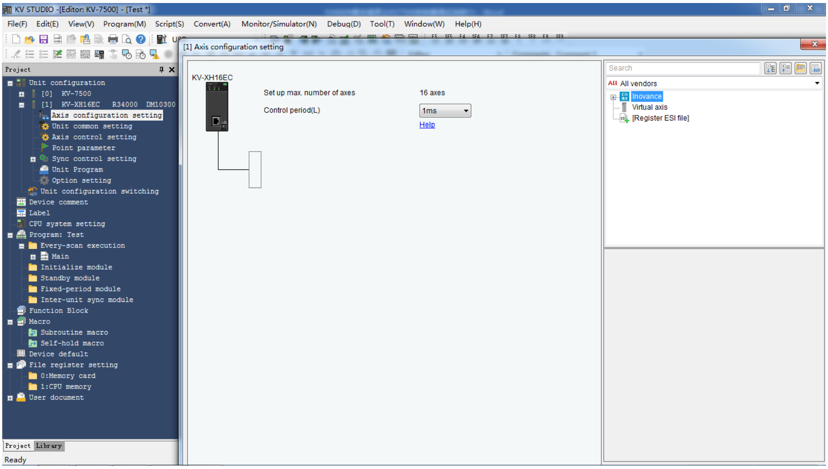
Power unit	
KV-PU1	AC power unit ...
KV-U7	AC power suppl...
Additional I/O unit(input)	
KV-B16X*	16-point input...

- If you click No, you can click Tool > Unit editor or double-click [0] KV7500 under Unit configuration.



Axis configuration setting

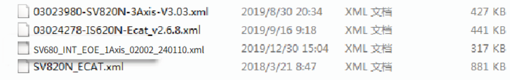
1. Enter “Axis configuration setting”.
2. Double-click “Register ESI file”.



Set up max. number of axes 16 axes

Control period(L) 1ms

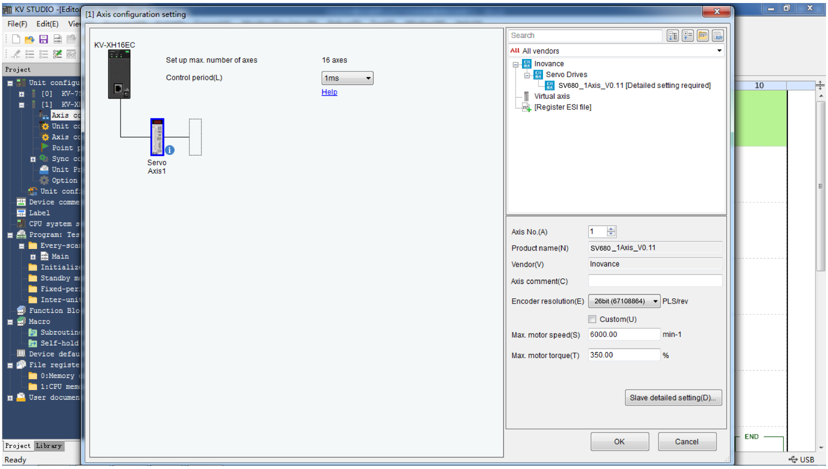
3. Find the storage directory of the device description file ".xml" and open it.
4. Importing the “. XML” file.



03023980-SV820N-3Axis-V3.03.xml	2019/8/30 20:34	XML 文档	427 KB
03024278-IS620N-Ecat_v2.6.8.xml	2019/9/16 9:18	XML 文档	441 KB
SV680_INT_EOE_1Axis_02002_240110.xml	2019/12/30 15:04	XML 文档	317 KB
SV820N_ECAT.xml	2018/3/21 8:47	XML 文档	881 KB

5. After the device description file is imported, you can start to add axes. You can also set the control period in “Axis configuration setting”. The default control cycle is 1 ms and the minimum control cycle is 250 us.

6. You can add the axes needed through dragging or double-clicking. Select the corresponding axis and set critical information such the Encoder resolution, Max. motor speed, and Max. motor torque for this axis.

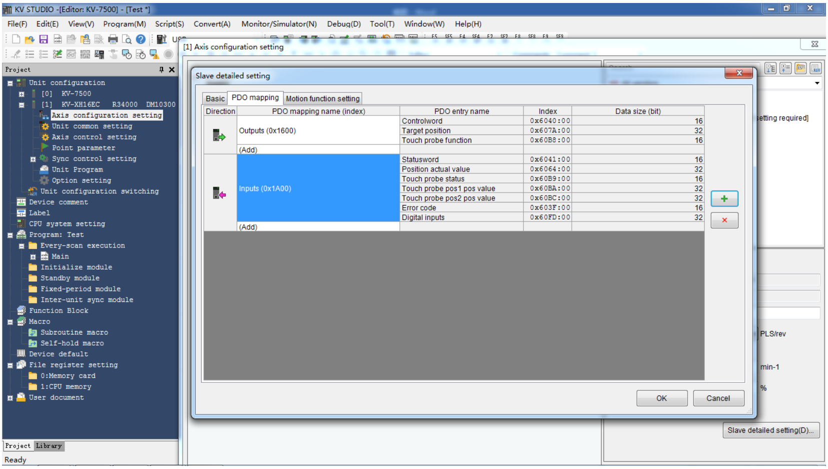


[1] Axis configuration setting

Set up max. number of axes	16 axes
Control period(L)	1ms

Axis No.(A)	1	
Product name(N)	SV680_1Axis_V0.11	
Vendor(V)	Inovance	
Axis comment(C)		
Encoder resolution(E)	26bit (67108864)	PLS/rev
	□ Custom(U)	
Max. motor speed(S)	6000.00	min-1
Max. motor torque(T)	350.00	%

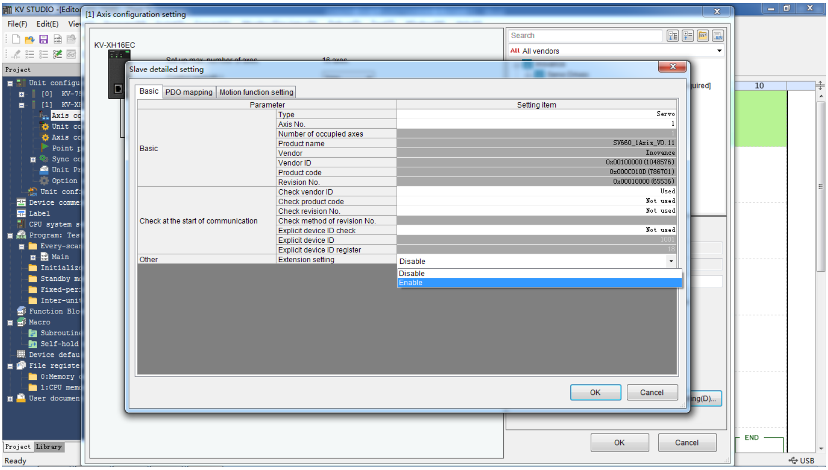
7. You can add PDO setting in detailed setting of the slave.



Slave detailed setting

Direction	PDO mapping name (index)	PDO entry name	Index	Data size (bit)
	Outputs (0x1600)	Controlword	0x6040:00	16
		Target position	0x607A:00	32
		Touch probe function	0x60B8:00	16
	(Add)			
	Inputs (0x1A00)	Statusword	0x6041:00	16
		Position actual value	0x6064:00	32
		Touch probe status	0x60B9:00	16
		Touch probe pos1 pos value	0x60BA:00	32
		Touch probe pos2 pos value	0x60BC:00	32
		Error code	0x603F:00	16
		Digital inputs	0x60FD:00	32
	(Add)			

8. If extension setting is needed, set Extension setting to Enable.

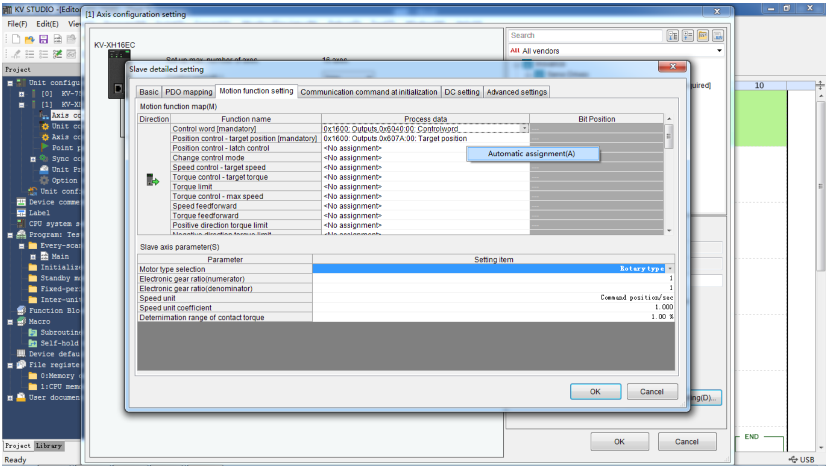


Parameter		Setting item
Basic	Type	Servo
	Axis No.	1
	Number of occupied axes	1
	Product name	SV660_1Axis_V0. 11
	Vendor	Inovance
	Vendor ID	0x00100000 (1048576)
	Product code	0x000C010D (786701)
	Revision No.	0x00010000 (65536)
Check at the start of communication	Check vendor ID	Used
	Check product code	Not used
	Check revision No.	Not used
	Check method of revision No.	
	Explicit device ID check	Not used
	Explicit device ID	1001
	Explicit device ID register	18
Other	Extension setting	Disable

Disable
Enable

9. For motion function settings, you can double-click or click on the combo box (small triangle icon) to select the PDO configuration needed from the dropdown list. You can also right-click > Automatic assignment > Yes, in this way the assigned contents will correspond to preceding PDO contents automatically.

   During manual assignment, do not neglect any contents in the PDO mapping. Otherwise, a pop-up window will be displayed to remind you of the missing contents when you click OK. For Communication command at initialization, DC setting, and Advanced settings, use the default values. After settings are done, click OK.



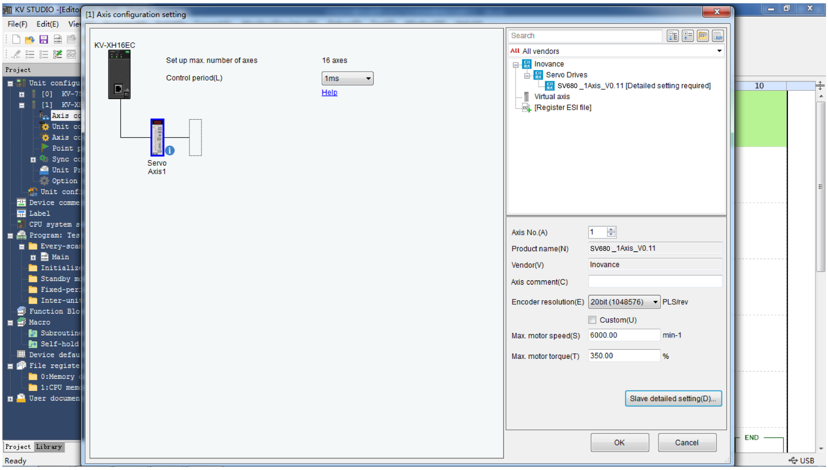
Motion function map(M)

Direction	Function name	Process data	Bit Position
	Control word [mandatory]	0x1600: Outputs.0x6040:00: Controlword	---
	Position control - target position [mandatory]	0x1600: Outputs.0x607A:00: Target position	---
	Position control - latch control	<No assignment>	---
	Change control mode	<No assignment>	---
	Speed control - target speed	<No assignment>	---
	Torque control - target torque	<No assignment>	---
	Torque limit	<No assignment>	---
	Torque control - max speed	<No assignment>	---
	Speed feedforward	<No assignment>	---
	Torque feedforward	<No assignment>	---
	Positive direction torque limit	<No assignment>	---
	Negative direction torque limit	<No assignment>	

Slave axis parameter(S)

Parameter	Setting item
Motor type selection	Rotary type
Electronic gear ratio(numerator)	1
Electronic gear ratio(denominator)	1
Speed unit	Command position/sec
Speed unit coefficient	1. 000
Determination range of contact torque	1. 00 %

10. After Slave detailed setting is done, the exclamation symbol disappears.



Set up max. number of axes: 16 axes

Control period(L): 1ms

Axis No.(A): 1

Product name(N): SV680 _1Axis_V0.11

Vendor(V): Inovance

Axis comment(C):

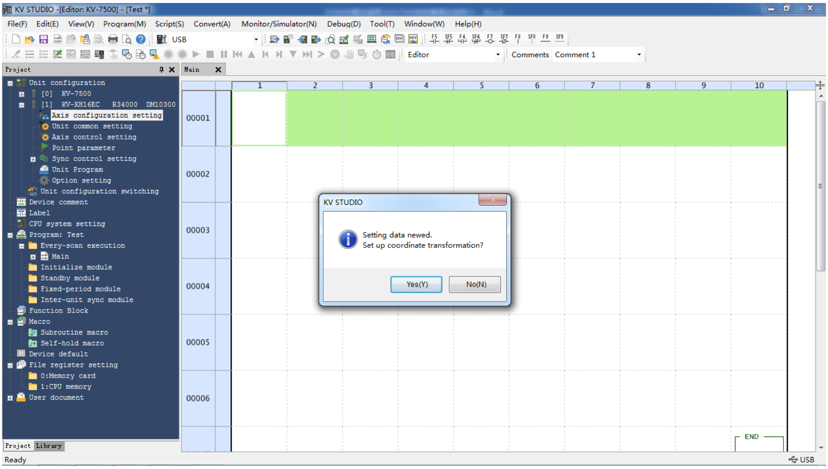
Encoder resolution(E): 20bit (1048576) PLS/rev

□ Custom(U)

Max. motor speed(S): 6000.00 min-1

Max. motor torque(T): 350.00 %

11. After adding the axes, click OK, and the following dialog box opens, asking you whether to set up coordinate (namely electronic gear ratio) transformation.

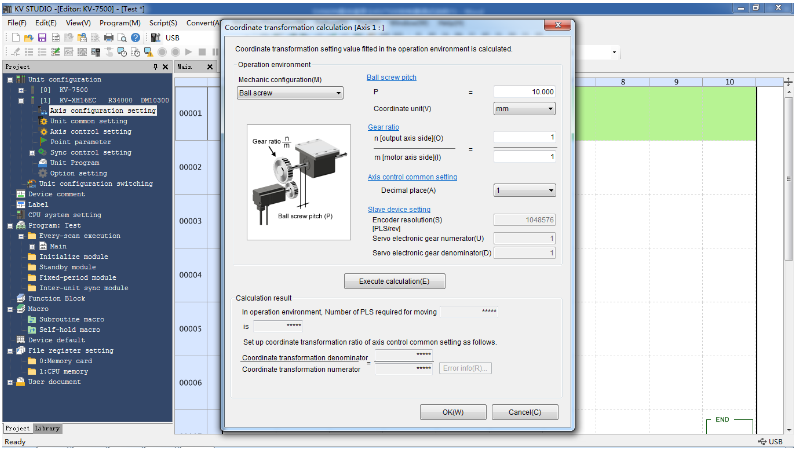


KV STUDIO

Setting data newed.
Set up coordinate transformation?

Yes(Y) No(N)

- Click Yes and the coordinate transformation dialog box opens. Set mechanical parameters and the coordinate unit based on actual conditions and click Execute calculation. The software calculates the denominator and numerator for coordinate transformation automatically and writes parameters to Axis control setting automatically.


Coordinate transformation calculation [Axis 1 :]

Coordinate transformation setting value fitted in the operation environment is calculated.

Operation environment

Mechanic configuration(M): Ball screw

Ball screw pitch
P = 10.000
Coordinate unit(V): mm

Gear ratio
n [output axis side](O): 1
m [motor axis side](I): 1

Axis control common setting
Decimal place(A): 1

Slave device setting
Encoder resolution(S) [PLS/rev]: 1048576
Servo electronic gear numerator(U): 1
Servo electronic gear denominator(D): 1

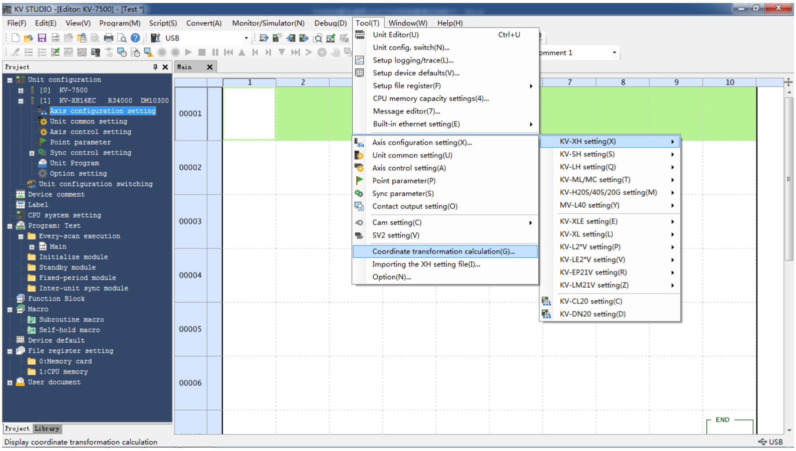
Calculation result

In operation environment, Number of PLS required for moving ***** is *****

Set up coordinate transformation ratio of axis control common setting as follows.

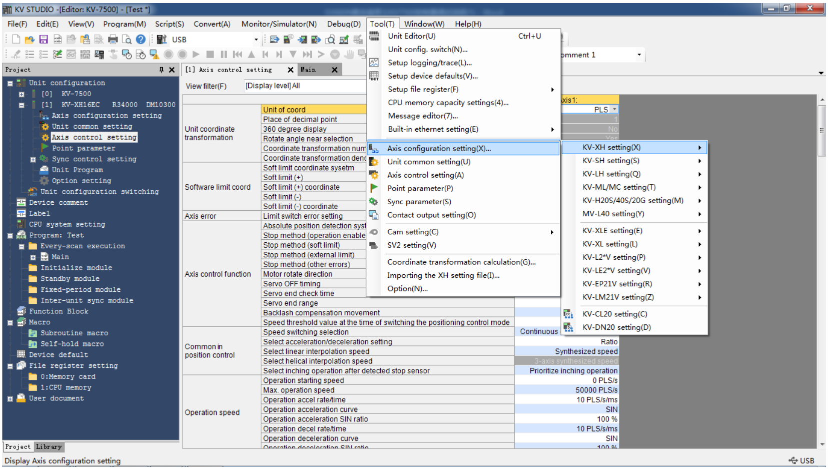
Coordinate transformation denominator: *****
Coordinate transformation numerator: *****

- If you click No, you can click Tool > Coordinate transformation calculation > KV-XH setting > Coordinate transformation calculation.


KV STUDIO -[Editor: KV-7500] - [Test *]
File(F) Edit(E) View(V) Program(M) Script(S) Convert(A) Monitor/Simulator(N) Debug(D) Tool(T) Window(W) Help(H)
Unit Editor(U) Ctrl+U
Unit config. switch(N)...
Setup logging/trace(L)...
Setup device defaults(V)...
Setup file register(F)
CPU memory capacity settings(4)...
Message editor(7)...
Built-in ethernet setting(E)
Axis configuration setting(X)...
Unit common setting(U)
Axis control setting(A)
Point parameter(P)
Sync parameter(S)
Contact output setting(O)
Cam setting(C)
SV2 setting(V)
Coordinate transformation calculation(G)...
Importing the XH setting file(I)...
Option(N)...
KV-XH setting(X)
KV-SH setting(S)
KV-LH setting(Q)
KV-ML/MC setting(T)
KV-H20S/40S/20G setting(M)
MV-L40 setting(Y)
KV-XLE setting(E)
KV-XL setting(L)
KV-L2*V setting(P)
KV-LE2*V setting(V)
KV-EP21V setting(R)
KV-LM21V setting(Z)
KV-CL20 setting(C)
KV-DN20 setting(D)
Display coordinate transformation calculation

Axis control setting

1. To open axis control setting, click Tool > Axis configuration setting > KV-XH setting > Axis control setting, or click Axis control setting under Project.
2. In axis control setting, you can set items including Unit coordinate transformation, Software limit coord, Axis error, Axis control function, Common in position control, Operation speed, and JOG.



View filter(F) [Display level] All

		xis1:
Unit coordinate transformation	Unit of coord	PLS
	Place of decimal point	1
	360 degree display	No
	Rotate angle near selection	Yes
	Coordinate transformation num	
	Coordinate transformation denc	
Software limit coord	Soft limit coordinate sysetm	
	Soft limit (+)	
	Soft limit (+) coordinate	
	Soft limit (-)	
	Soft limit (-) coordinate	
Axis error	Limit switch error setting	
Axis control function	Absolute position detection syst	
	Stop method (operation enable	
	Stop method (soft limit)	
	Stop method (external limit)	
	Stop method (other errors)	
	Motor rotate direction	
	Servo OFF timing	
	Servo end check time	
	Servo end range	
	Backlash compensation movement	
	Speed threshold value at the time of switching the positioning control mode	
Common in position control	Speed switching selection	Continuous
	Select acceleration/deceleration setting	Ratio
	Select linear interpolation speed	Synthesized speed
	Select helical interpolation speed	3-axis synthesized speed
	Select inching operation after detected stop sensor	Prioritize inching operation
Operation speed	Operation starting speed	0 PLS/s
	Max. operation speed	50000 PLS/s
	Operation accel rate/time	10 PLS/s/ms
	Operation acceleration curve	SIN
	Operation acceleration SIN ratio	100 %
	Operation decel rate/time	10 PLS/s/ms
	Operation deceleration curve	SIN
	Operation deceleration SIN ratio	100 %

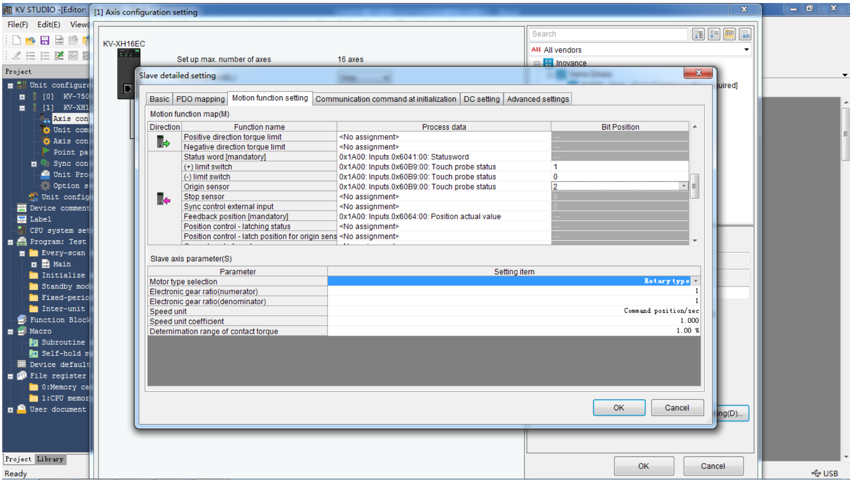
Running setting

Homing

Before homing, assign (+) limit switch, (-) limit switch, and Origin sensor in Motion function setting under Axis configuration setting to each bit of 60FDh. 60FDh is defined as follows:

bit0: negative limit; bit1: positive limit; bit2: home switch; bit16...bit20 correspond to DI1...DI5 respectively.

In automatic assignment, you need to assign (+) limit switch, (-) limit switch, and origin sensor manually, you can assign them to corresponding bits of 60FDh based on the relation shown in the following figure or to bit16...bit20, in this case, you also need to assign them to corresponding DIs of the servo drive.



Set up max. number of axes 16 axes

Slave detailed setting

Basic | PDO mapping | Motion function setting | Communication command at initialization | DC setting | Advanced settings

Motion function map(M)

Direction	Function name	Process data	Bit Position
	Positive direction torque limit	<No assignment>	---
	Negative direction torque limit	<No assignment>	---
	Status word [mandatory]	0x1A00: Inputs.0x6041:00: Statusword	---
	(+) limit switch	0x1A00: Inputs.0x60B9:00: Touch probe status	1
	(-) limit switch	0x1A00: Inputs.0x60B9:00: Touch probe status	0
	Origin sensor	0x1A00: Inputs.0x60B9:00: Touch probe status	2
	Stop sensor	<No assignment>	0
	Sync control external input	<No assignment>	0
	Feedback position [mandatory]	0x1A00: Inputs.0x6064:00: Position actual value	---
	Position control - latching status	<No assignment>	---
	Position control - latch position for origin sens	<No assignment>	---

Slave axis parameter(S)

Parameter	Setting item
Motor type selection	Rotary type
Electronic gear ratio(numerator)	1
Electronic gear ratio(denominator)	1
Speed unit	Command position/sec
Speed unit coefficient	1.000
Determination range of contact torque	1.00 %

Set the restriction parameters for homing in Axis control setting > Origin return. The following homing methods are available. For detailed trajectories, see KEYENCE instruction manual for positioning/motion control unit KV-XH16EC.

Default	Value range	Description
DOG type (with phase Z)	DOG type (with phase Z)	Decelerating upon DOG signal input and homing through phase Z signal
	DOG type (without phase Z)	Decelerating upon DOG signal input and homing through falling edge of DOG signal
	DOG-type jogging (with phase Z)	Pausing after moving based on Dog ON upon DOG signal input. Then moving to the homing direction through position-type speed control and homing with phase Z signal.
	DOG-type jogging (without phase Z)	Moving based on Dog ON upon DOG signal input before homing
	DOG type (contact)	Homing executed when the ON duration of the torque limit signal keeps longer than the compression torque time upon DOG signal input
	Origin sensor and phase Z	Homing executed in the initial phase Z position after the origin sensor is ON
	Rising edge of origin sensor	Homing executed through the rising edge of the origin sensor
	Middle point of origin sensor (without phase Z)	Taking the middle point of the ON range of origin sensor as the origin and comparing it with that in "Rising edge of origin sensor". Even if the light-receptive performance of the origin sensor degrades, the homing position can hardly change with the time.
	Rising edge of limit switch	Homing executed with the limit switch in the negative direction (direction where the current coordinate decreases) acting as the origin sensor
	Immediate homing of phase Z	Homing executed with phase Z signal
	Data setting type	Taking current coordinate as the origin coordinate

The following homing methods are available in IS620N and SV680N-INT series servo drives.

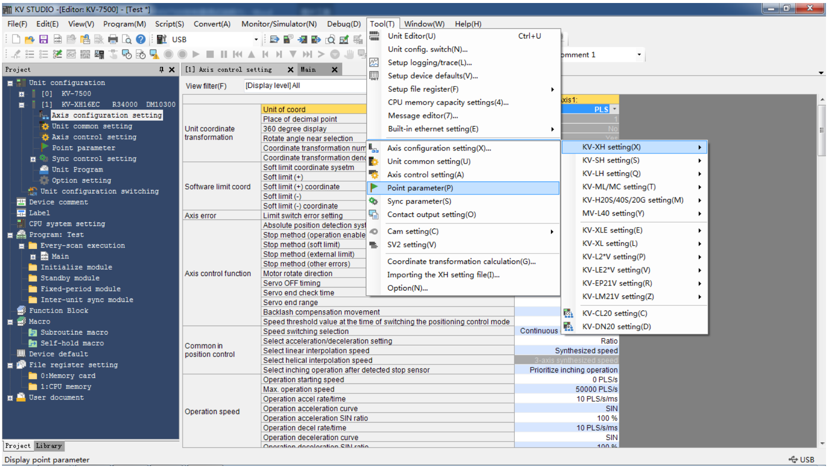
No.	Homing mode	IS620N	SV680N-INT
1	DOG-type (with phase Z)	OK	OK
2	DOG-type (without phase Z)	OK	OK
3	DOG-type jogging (with phase Z)	No	No
4	DOG-type jogging (without phase Z)	No	No

5	DOG-type (contact)	OK	Homing is available, but the reference coordinate after homing is not 0. Updating to the xml coordinate of IS620N zeros out the reference coordinate.
6	Origin sensor and phase Z	OK	OK
7	Rising edge of origin sensor	OK	OK
8	Middle point of origin sensor	No	No
9	Rising edge of limit switch	Homing is available, but the reference coordinate after homing is not 0.	Homing is available, but the reference coordinate after homing is not 0.
10	Immediate homing of phase Z	OK	OK

Positioning

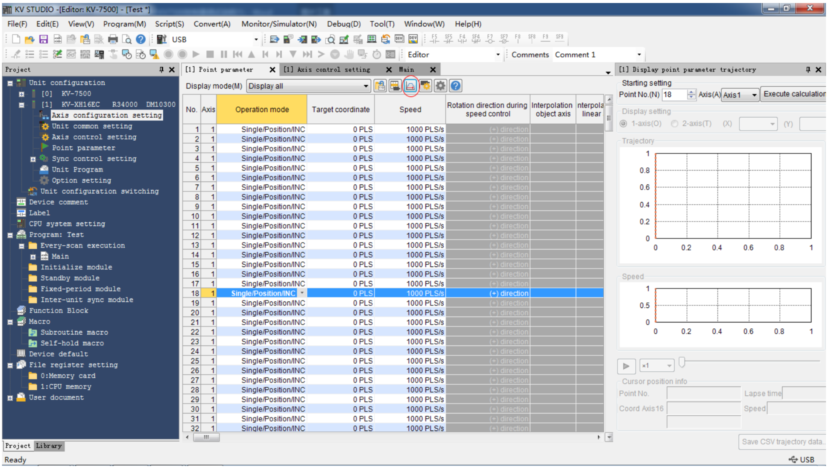
Set the unit coordinate transformation properly before positioning. The unit coordinate transformation is "PLS" by default, which allows no modification on the numerator or denominator. Assume N revolutions are required by the servo drive, in this case, the number of commands that need to be sent by the host controller is N x Pulses per revolution. If coordinate transformation calculation has been executed, the unit coordinate transformation parameters will correspond to the unit transformation results automatically.

1. To set the motion profile of the servo drive, click Tool > Point parameter.


KV STUDIO -[Editor: KV-7500] - [Test *]
File(F) Edit(E) View(V) Program(M) Script(S) Convert(A) Monitor/Simulator(N) Debug(D) Tool(T) Window(W) Help(H)
USB
Project
Unit configuration
[0] KV-7500
[1] KV-XH16EC R34000 DM10300
Axis configuration setting
Unit common setting
Axis control setting
Point parameter
Sync control setting
Unit Program
Option setting
Unit configuration switching
Device comment
Label
CPU system setting
Program: Test
Every-scan execution
Main
Initialize module
Standby module
Fixed-period module
Inter-unit sync module
Function Block
Macro
Subroutine macro
Self-hold macro
Device default
File register setting
0:Memory card
1:CPU memory
User document
Project Library
[1] Axis control setting
Main
View filter(F) [Display level] All
Unit coordinate transformation
Unit of coord
Place of decimal point
360 degree display
Rotate angle near selection
Coordinate transformation num
Coordinate transformation deno
Software limit coord
Soft limit coordinate sysetm
Soft limit (+)
Soft limit (+) coordinate
Soft limit (-)
Soft limit (-) coordinate
Axis error
Limit switch error setting
Axis control function
Absolute position detection syst
Stop method (operation enable
Stop method (soft limit)
Stop method (external limit)
Stop method (other errors)
Motor rotate direction
Servo OFF timing
Servo end check time
Servo end range
Backlash compensation movement
Speed threshold value at the time of switching the positioning control mode
Common in position control
Speed switching selection Continuous
Select acceleration/deceleration setting Ratio
Select linear interpolation speed Synthesized speed
Select helical interpolation speed 3-axis synthesized speed
Select inching operation after detected stop sensor Prioritize inching operation
Operation speed
Operation starting speed 0 PLS/s
Max. operation speed 50000 PLS/s
Operation accel rate/time 10 PLS/s/ms
Operation acceleration curve SIN
Operation acceleration SIN ratio 100 %
Operation decel rate/time 10 PLS/s/ms
Operation deceleration curve SIN
Axis1:
PLS
Unit Editor(U) Ctrl+U
Unit config. switch(N)...
Setup logging/trace(L)...
Setup device defaults(V)...
Setup file register(F)
CPU memory capacity settings(4)...
Message editor(7)...
Built-in ethernet setting(E)
Axis configuration setting(X)...
Unit common setting(U)
Axis control setting(A)
Point parameter(P)
Sync parameter(S)
Contact output setting(O)
Cam setting(C)
SV2 setting(V)
Coordinate transformation calculation(G)...
Importing the XH setting file(I)...
Option(N)...
KV-XH setting(X)
KV-SH setting(S)
KV-LH setting(Q)
KV-ML/MC setting(T)
KV-H20S/40S/20G setting(M)
MV-L40 setting(Y)
KV-XLE setting(E)
KV-XL setting(L)
KV-L2*V setting(P)
KV-LE2*V setting(V)
KV-EP21V setting(R)
KV-LM21V setting(Z)
KV-CL20 setting(C)
KV-DN20 setting(D)
Display point parameter
USB

Set the target coordinate and speed per positioning segment as needed. After settings are done, you can call the corresponding point number through the program to start operation.

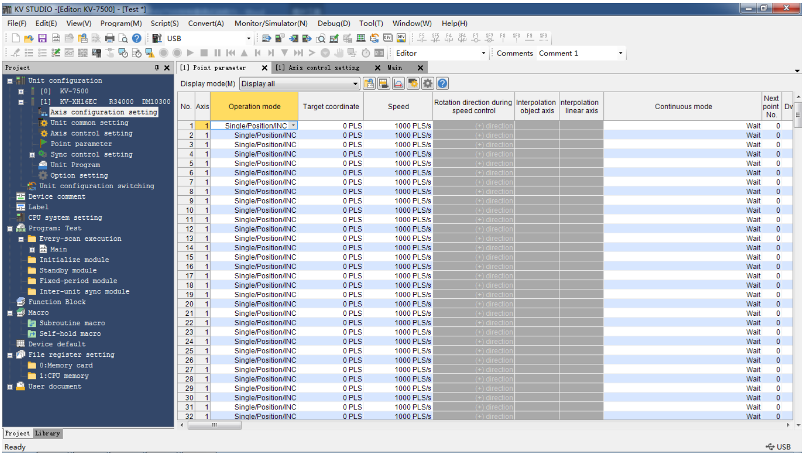
2. You can preview the parameter trajectory through the following short-cut.



No.	Axis	Operation mode	Target coordinate	Speed	Rotation direction during speed control	Interpolation object axis	nterpola linear
1	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
2	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
3	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
4	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
5	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
6	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
7	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
8	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
9	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
10	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
11	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
12	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
13	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
14	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
15	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
16	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
17	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
18	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
19	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
20	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
21	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
22	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
23	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
24	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
25	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
26	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
27	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
28	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
29	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
30	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
31	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		
32	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction		

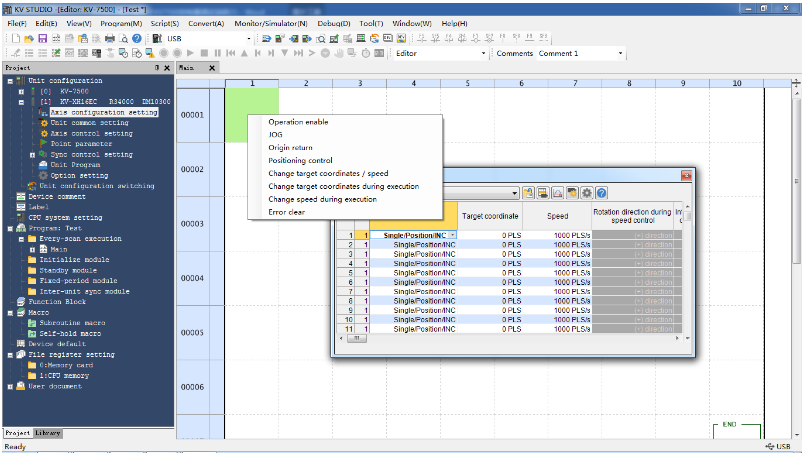
3. You can write ladder diagrams through regular methods. You can also use the following short-cut method provided by KEYENCE.

a. Drag down the Point parameter window with the left mouse button, and zoom out the window to put it in a proper place.



No.	Axis	Operation mode	Target coordinate	Speed	Rotation direction during speed control	Interpolation object axis	nterpolation linear axis	Continuous mode	Next point No.	Dv
1	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
2	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
3	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
4	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
5	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
6	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
7	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
8	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
9	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
10	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
11	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
12	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
13	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
14	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
15	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
16	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
17	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
18	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
19	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
20	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
21	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
22	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
23	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
24	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
25	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
26	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
27	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
28	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
29	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
30	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
31	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	
32	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction			Wait	0	

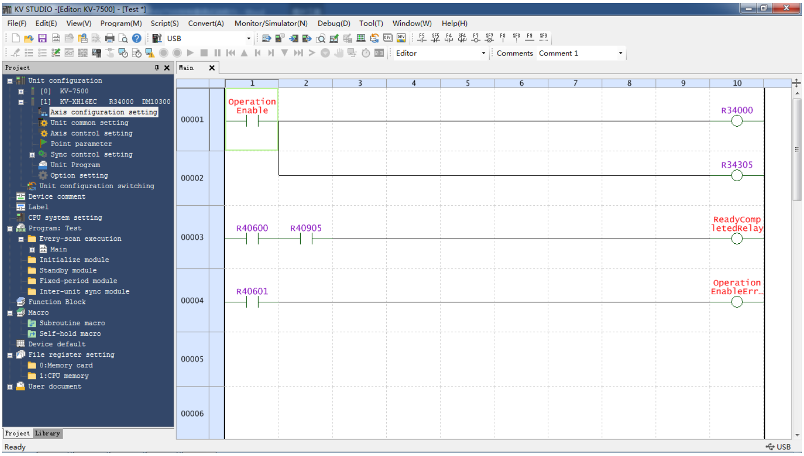
b. Move the mouse to the point parameter, such as "No.1-Axis1", and wait until the mouse icon to change from an arrow to a small hand. Then drag towards the program edit interface with the right mouse button, and the following short-cut pops out.


KV STUDIO -[Editor: KV-7500] - [Test *]
File(F) Edit(E) View(V) Program(M) Script(S) Convert(A) Monitor/Simulator(N) Debug(D) Tool(T) Window(W) Help(H)
Operation enable
JOG
Origin return
Positioning control
Change target coordinates / speed
Change target coordinates during execution
Change speed during execution
Error clear
END
Ready

			Target coordinate	Speed	Rotation direction during speed control
1	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
2	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
3	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
4	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
5	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
6	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
7	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
8	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
9	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
10	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction
11	1	Single/Position/INC	0 PLS	1000 PLS/s	(+) direction

c. Select the desired function.

If the operation is enabled, click it to automatically generate a DEMO program. Then designate the part in red as the relay needed. After these actions are done, this function is done compiling.

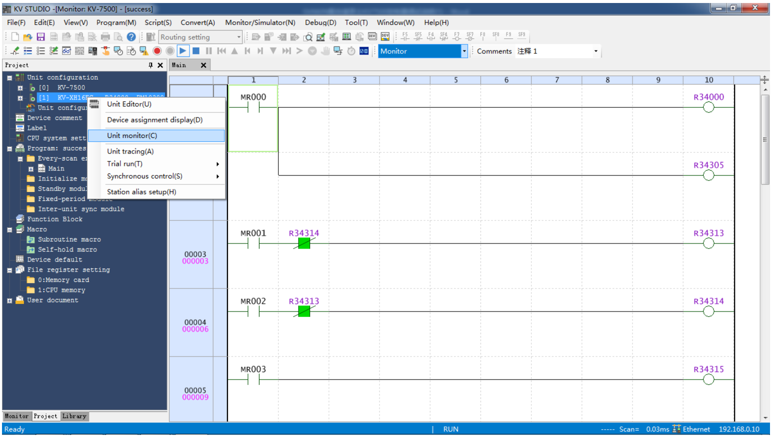


4. Unit monitor

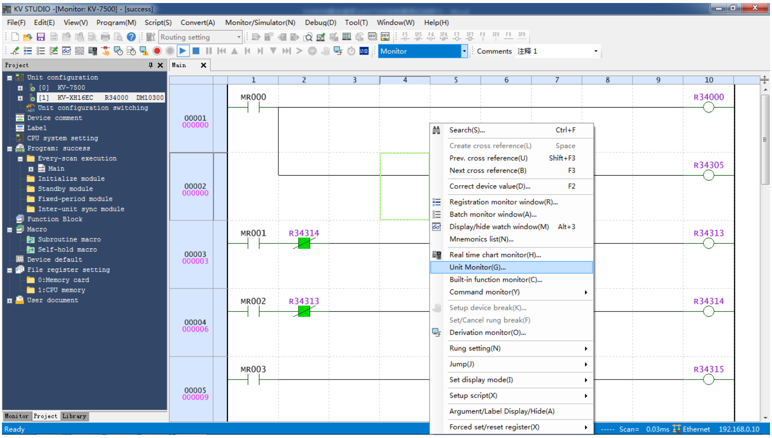
The unit monitor supports monitoring on the operating state of KV-XH16EC or the internal data.

a. Open “Unit monitor”. There are three ways:

- Select the unit to be monitored and right-click to select Unit monitor in the short-cut menu.

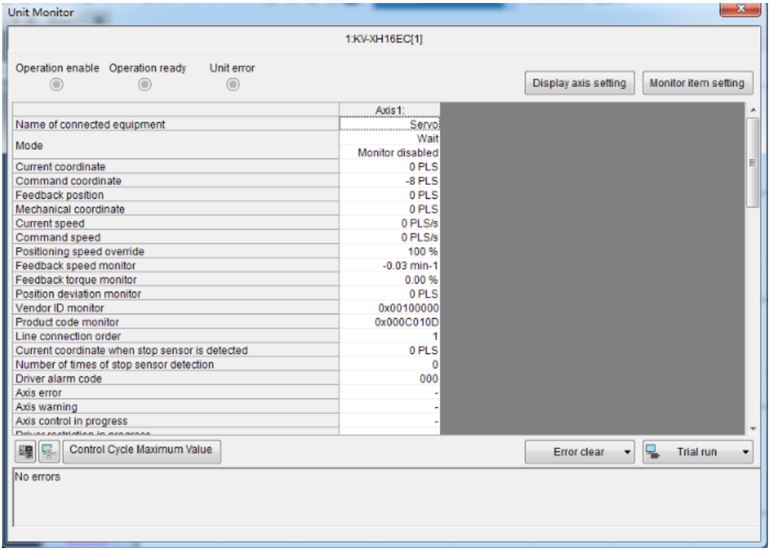


- Double-click with left mouse button to open the Unit monitor.
- Right-click the blank section in the main program to select Unit monitor in the pop-up menu.


KV STUDIO -[Monitor: KV-7500] - [success]
Search(S)... Ctrl+F
Create cross reference(L) Space
Prev. cross reference(U) Shift+F3
Next cross reference(B) F3
Correct device value(D)... F2
Registration monitor window(R)...
Batch monitor window(A)...
Display/hide watch window(M) Alt+3
Mnemonics list(N)...
Real time chart monitor(H)...
Unit Monitor(G)...
Built-in function monitor(C)...
Command monitor(Y)
Setup device break(K)...
Set/Cancel rung break(F)
Derivation monitor(O)...
Rung setting(N)
Jump(J)
Set display mode(I)
Setup script(X)
Argument/Label Display/Hide(A)
Forced set/reset register(X)
Ready
Scan= 0.03ms Ethernet 192.168.0.10

b. The unit monitor displays the operating state of each axis.

1). To change the operating state of the monitor item, click Monitor item setting on the top right corner.



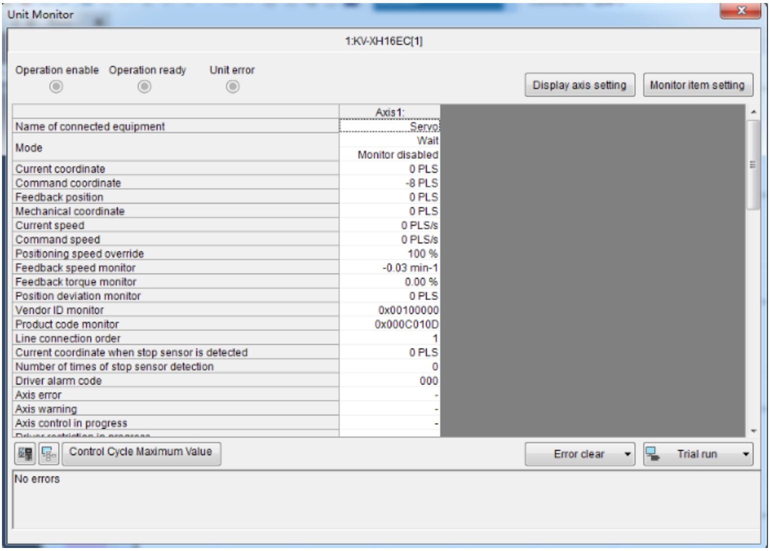
1:KV-XH16EC[1]

	Axis1:
Name of connected equipment	Servo
Mode	Wait Monitor disabled
Current coordinate	0 PLS
Command coordinate	-8 PLS
Feedback position	0 PLS
Mechanical coordinate	0 PLS
Current speed	0 PLS/s
Command speed	0 PLS/s
Positioning speed override	100 %
Feedback speed monitor	-0.03 min-1
Feedback torque monitor	0.00 %
Position deviation monitor	0 PLS
Vendor ID monitor	0x00100000
Product code monitor	0x000C010D
Line connection order	1
Current coordinate when stop sensor is detected	0 PLS
Number of times of stop sensor detection	0
Driver alarm code	000
Axis error	-
Axis warning	-
Axis control in progress	-

No errors

2). To check whether I/O signals such as limit switch signals and origin sensor signals are normal, open Unit monitor and find the corresponding monitoring position.

If corresponding message is received, a small black circle will be displayed.



1:KV-XH16EC[1]

	Axis1:
Name of connected equipment	Servo
Mode	Wait Monitor disabled
Current coordinate	0 PLS
Command coordinate	-8 PLS
Feedback position	0 PLS
Mechanical coordinate	0 PLS
Current speed	0 PLS/s
Command speed	0 PLS/s
Positioning speed override	100 %
Feedback speed monitor	-0.03 min-1
Feedback torque monitor	0.00 %
Position deviation monitor	0 PLS
Vendor ID monitor	0x00100000
Product code monitor	0x000C010D
Line connection order	1
Current coordinate when stop sensor is detected	0 PLS
Number of times of stop sensor detection	0
Driver alarm code	000
Axis error	-
Axis warning	-
Axis control in progress	-

No errors

The error state of the unit can also be displayed in the Unit monitor. The axis error can be cleared using the Error clear button in the bottom right.

5.3.4.3 Trial Run

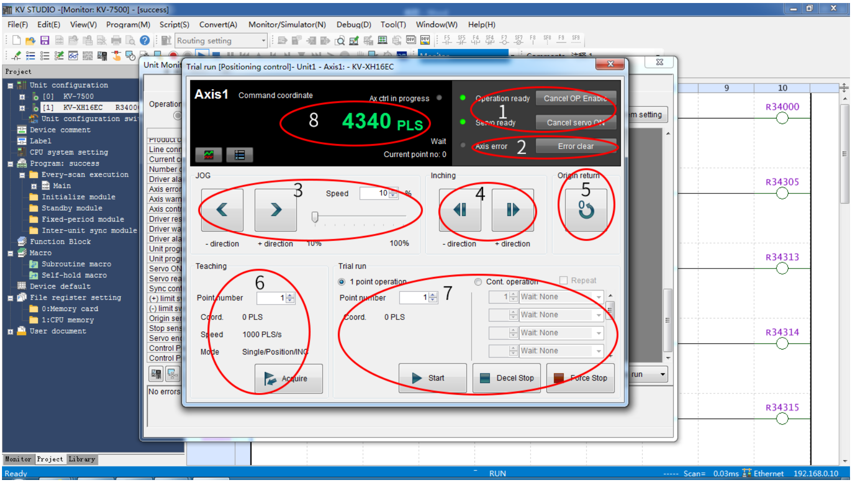
In trial run, actions can be acknowledged directly, without the need for programming ladder diagrams.

1. You can find the Trial run button at the bottom right of the unit monitor interface.
2. Select the control mode from positioning control, speed control, and torque control.
3. Then, select the object axis for trial run.

Note

If trial run is executed in the speed control mode or torque control mode, a warning will be reported. To execute trial run, set the control mode to position control.

The following introduces trial run → positioning control.



1. OP enable/Servo ON.

   Unrelated to the status of the ladder diagram program. OP enable and Servo ON can be executed through Commissioning. After operations are done, the Operation ready and Servo ready indicators turn green. To ensure safety, set the CPU unit to PROG mode and execute operations again after stopping ladder diagram program.

   Confirm the following items when the Servo ready indicator is not in green.

   - No error occurs on the axis.
   - No warning occurs on the servo drive.
   - The main circuit power supply of the servo drive is switched on.
   - The Ethernet cable is connected.

2. Axis error/Error clear

   Check the error details and clear the error. After rectifying the error cause, click the Clear button to clear the error.

3. JOG.

   Click the "FWD" and "REV" buttons to perform JOG operation in forward/reverse directions respectively. The jogging speed is the value in General Axis Control Settings→JOG High Speed multiplied by a ratio. You can set the ratio at a 1% increment between 10% and 100%.

4. Inching.

   Click the "FWD" and "REV" buttons to perform inching in forward/reverse directions respectively. The inching runs at the speed specified in General Axis Control Settings→JOG Start Speed. The inching runs with the movement specified in General Axis Control Settings→JOG Inch Movement.

5. Origin return

   Click the Origin return button to execute homing.

6. Teaching

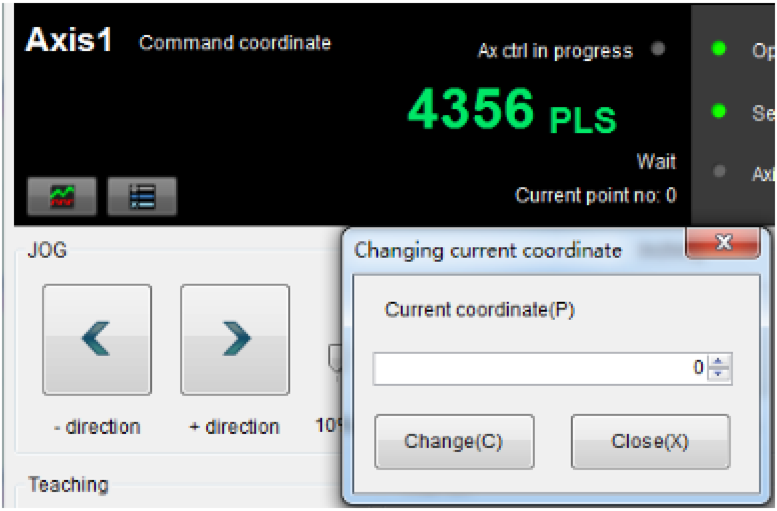
   Click the Acquire button to save current command coordinate value to the buffer memory of the target coordinate of the designated point number. The teaching function is available only in the online edit mode. The teaching value will also be reflected to the buffer memory and the point parameter.

7. Trial run

   Designate a point number and click the Start button to execute point positioning. To stop operation, click the **Stop** button. Clicking the 1 point operation button makes the servo drive execute positioning of one point. Clicking the Cont. operation button makes the servo drive execute positioning of ten points at most. Clicking the Repeat button makes the servo drive return to the point in the first row and execute positioning repeatedly after positioning of the point in the last row is done. The time interval between points can be set to a value within 0.1s to 20.0s.

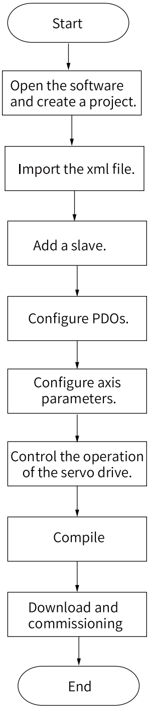
8. Changing current coordinate

   Click Command coordinate and the Changing current coordinate dialog box opens. Enter the coordinate needing to be changed and click the Change button to change the current coordinate of the axis in trial run, and then close the Changing current coordinate dialog box. If you click the Close button after changing current coordinate, the Changing current coordinate dialog box will be closed with current coordinate unchanged.



5.3.5 SV680N-INT and EASY Controller

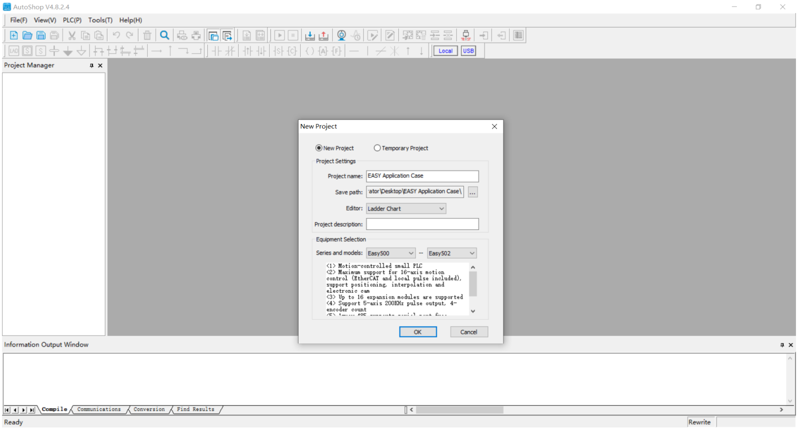
This section describes how to configure the SV680N-INT series servo drive for cooperation with the EASY series controller.


Start
Open the software and create a project.
Import the xml file.
Add a slave.
Configure PDOs.
Configure axis parameters.
Control the operation of the servo drive.
Compile
Download and commissioning
End

1. **Open the software, and create an EASY project.**

   a. Open Autoshop and click "New Project". In the popup dialog box, first select the editor type, and then select Easy500 as the PLC type.

   b. Enter the project name and select the save path, and then click "OK" to create a new project and enter the project main interface.



New Project

● New Project ○ Temporary Project

Project Settings

Project name: EASY Application Case

Save path: ator\Desktop\EASY Application Case\

Editor: Ladder Chart

Project description:

Equipment Selection

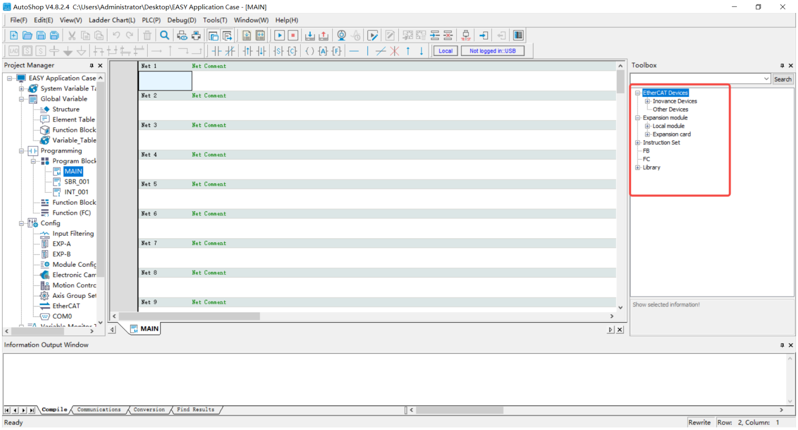
Series and models: Easy500 -- Easy502

```
<1> Motion-controlled small PLC
<2> Maximum support for 16-axis motion
control (EtherCAT and local pulse included),
support positioning, interpolation and
electronic cam
<3> Up to 16 expansion modules are supported
<4> Support 5-axis 200KHz pulse output, 4-
encoder count
```

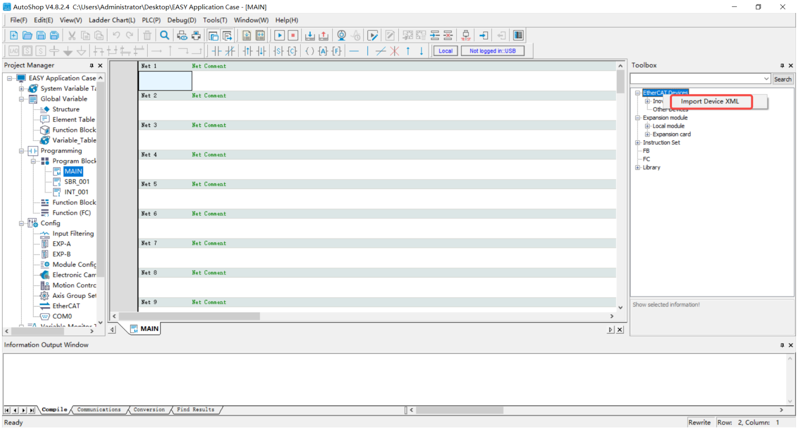
OK Cancel

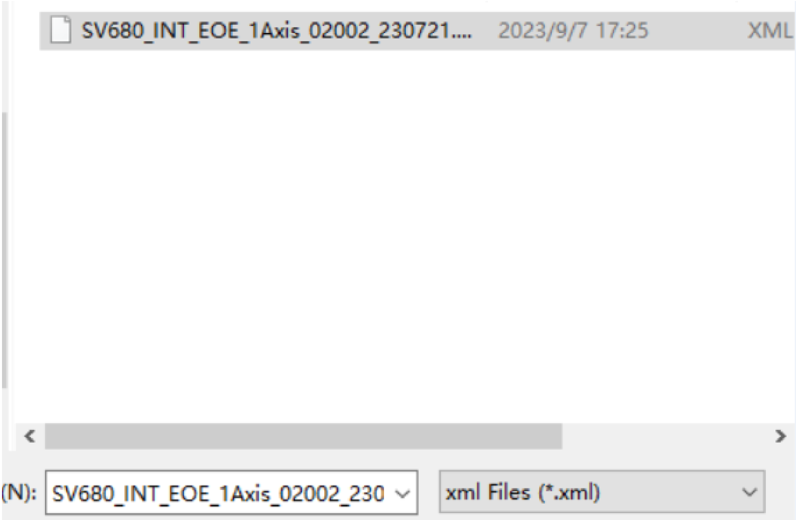
2. **Importing device XML**

   a. Open the toolbox and find the EtherCAT Devices list.



   b. Right-click on EtherCAT Devices, and in the pop-up dialog box, select the desired XML file and import it.



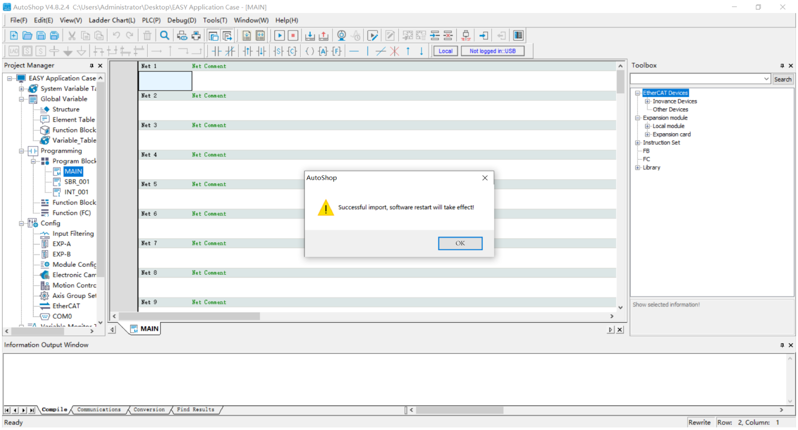


SV680_INT_EOE_1Axis_02002_230721.... 2023/9/7 17:25 XML

(N): SV680_INT_EOE_1Axis_02002_230

xml Files (*.xml)

c. You need to restart the application to let the imported xml take effect.



AutoShop V4.8.2.4 C:\Users\Administrator\Desktop\EASY Application Case - [MAIN]

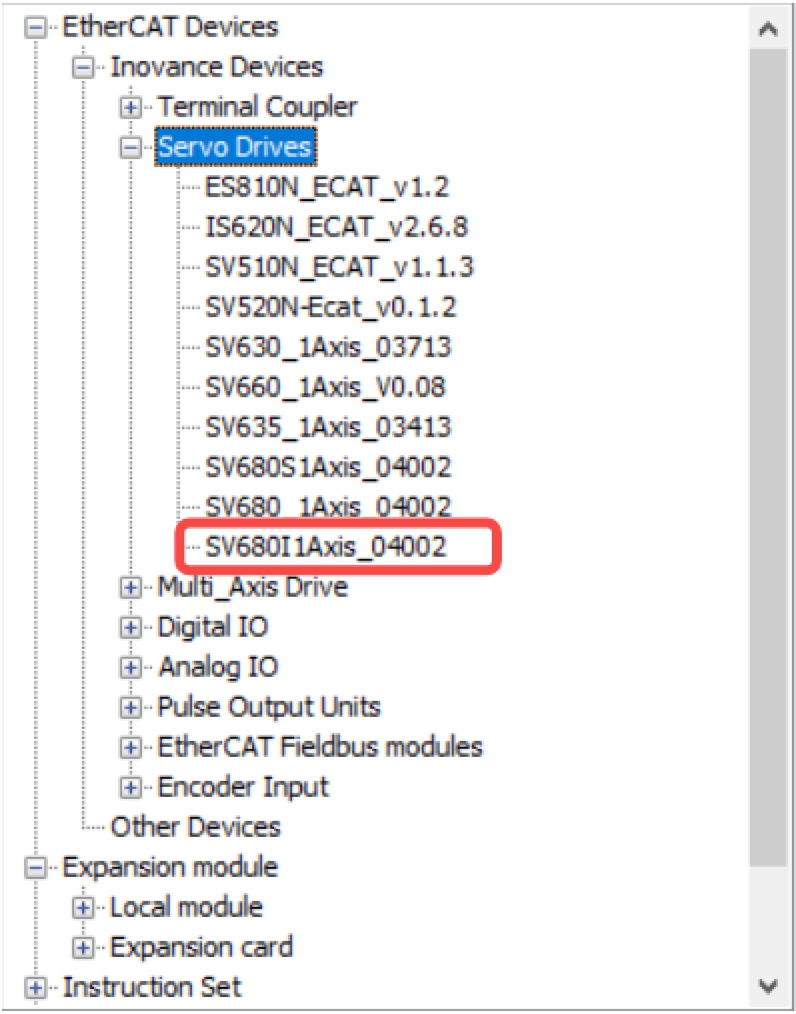
AutoShop

Successful import, software restart will take effect!

OK

After clicking "OK", you need to restart the application manually to let the newly added device take effect.

d. After reopening the application, you can see the newly added device.

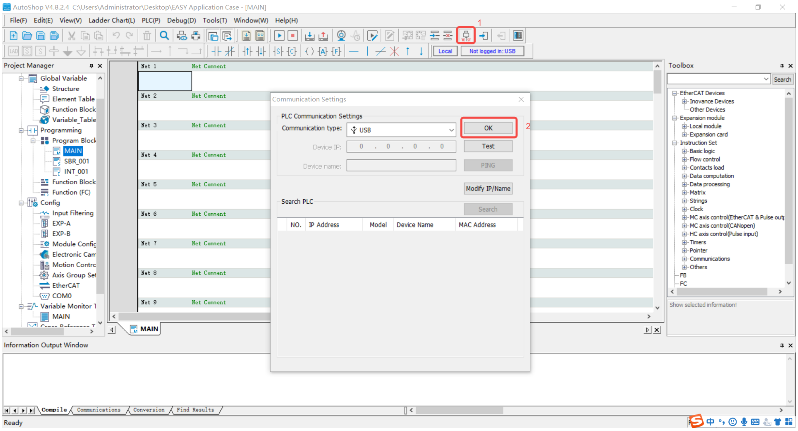


- EtherCAT Devices
  - Inovance Devices
    - Terminal Coupler
    - Servo Drives
      - ES810N_ECAT_v1.2
      - IS620N_ECAT_v2.6.8
      - SV510N_ECAT_v1.1.3
      - SV520N-Ecat_v0.1.2
      - SV630_1Axis_03713
      - SV660_1Axis_V0.08
      - SV635_1Axis_03413
      - SV680S1Axis_04002
      - SV680_1Axis_04002
      - SV680I1Axis_04002
    - Multi_Axis Drive
    - Digital IO
    - Analog IO
    - Pulse Output Units
    - EtherCAT Fieldbus modules
    - Encoder Input
  - Other Devices
- Expansion module
  - Local module
  - Expansion card
- Instruction Set

3. **Adding a slave station**

First, connect the PLC through Ethernet.

a. Select the target host.



Communication Settings

PLC Communication Settings

Communication type: USB

Device IP: 0 . 0 . 0 . 0

Device name:

OK

Test

PING

Modify IP/Name

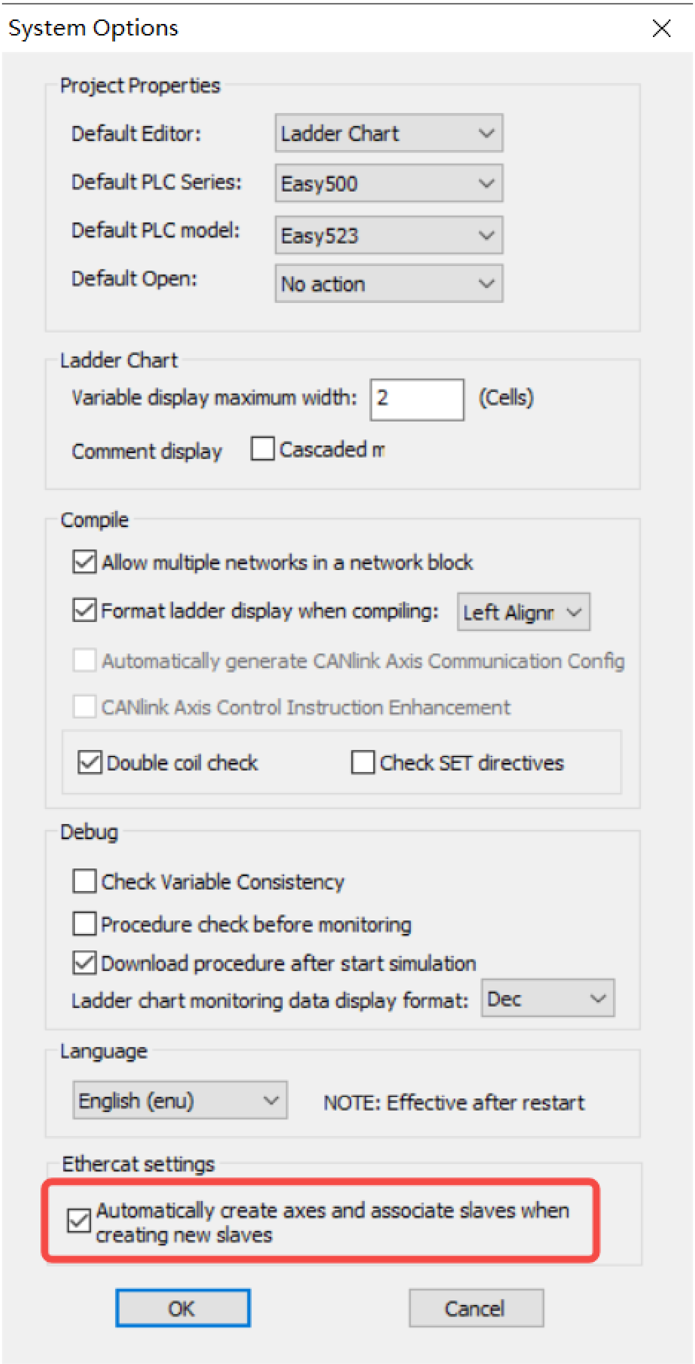
Search PLC

Search

NO.	IP Address	Model	Device Name	MAC Address

b. Set whether to automatically associate motion control axes as needed.

If you select "Auto create axis and associate slave station when creating new slave station" in EtherCAT Settings, a motion control axis will be automatically added for each drive-type EtherCAT slave station.



System Options

Project Properties

Default Editor: Ladder Chart

Default PLC Series: Easy500

Default PLC model: Easy523

Default Open: No action

Ladder Chart

Variable display maximum width: 2 (Cells)

Comment display ☐ Cascaded m

Compile

☑ Allow multiple networks in a network block

☑ Format ladder display when compiling: Left Alignr

☐ Automatically generate CANlink Axis Communication Config

☐ CANlink Axis Control Instruction Enhancement

☑ Double coil check ☐ Check SET directives

Debug

☐ Check Variable Consistency

☐ Procedure check before monitoring

☑ Download procedure after start simulation

Ladder chart monitoring data display format: Dec

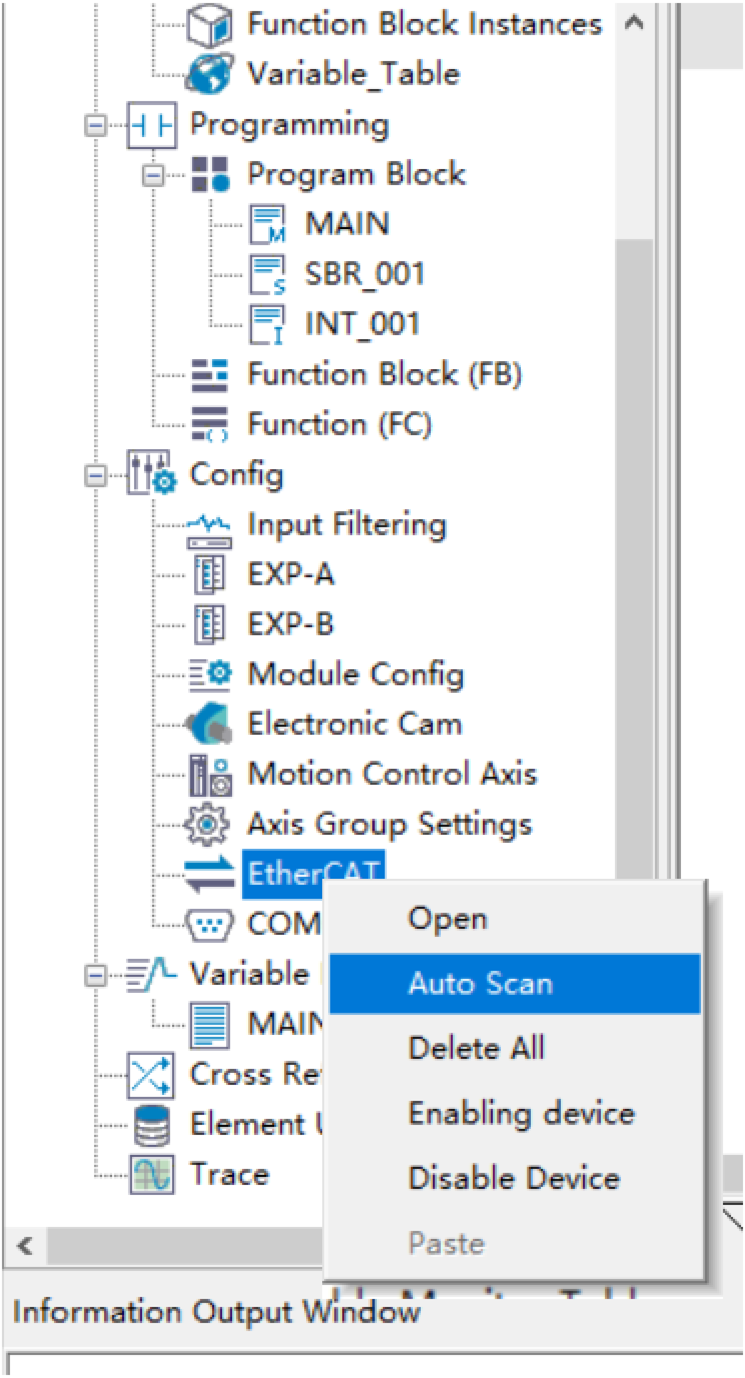
Language

English (enu) NOTE: Effective after restart

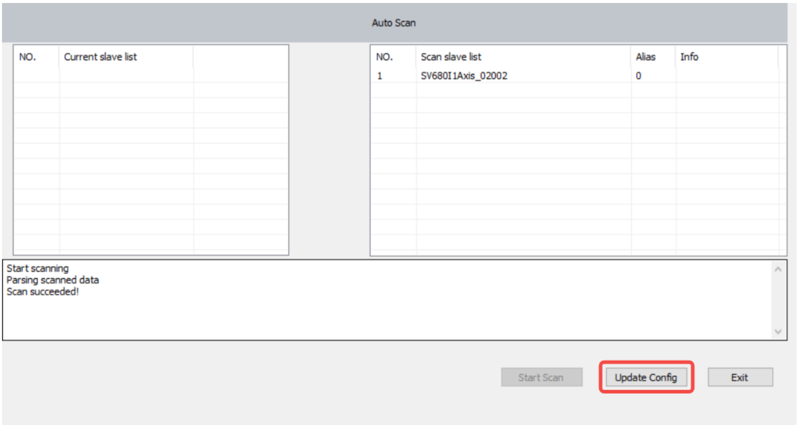
Ethercat settings

☑ Automatically create axes and associate slaves when creating new slaves

c. Right click the EtherCAT tab and select Auto Scan.


Function Block Instances
Variable_Table
Programming
Program Block
MAIN
SBR_001
INT_001
Function Block (FB)
Function (FC)
Config
Input Filtering
EXP-A
EXP-B
Module Config
Electronic Cam
Motion Control Axis
Axis Group Settings
EtherCAT
COM
Variable
MAIN
Cross Re
Element
Trace
Open
Auto Scan
Delete All
Enabling device
Disable Device
Paste
Information Output Window

d. Select Start Scan in the pop-up dialog box. After the scan completes, you can see all scanned slave devices. Click Update Configuration to update the scanned devices to the configuration list.

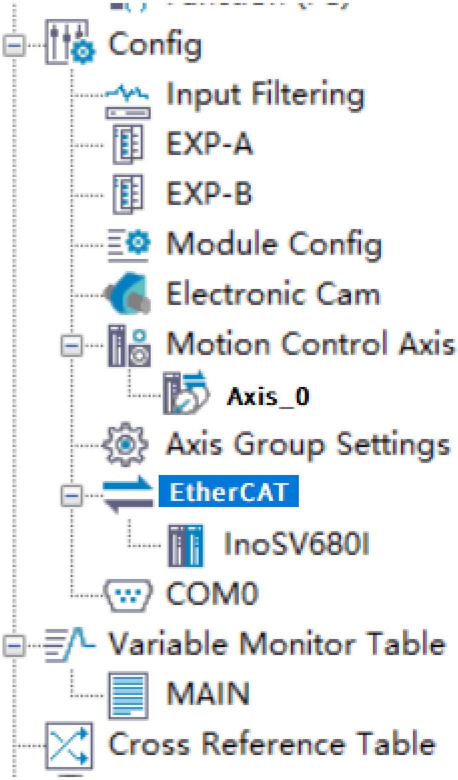


NO.	Current slave list

NO.	Scan slave list	Alias	Info
1	SV680I1Axis_02002	0	

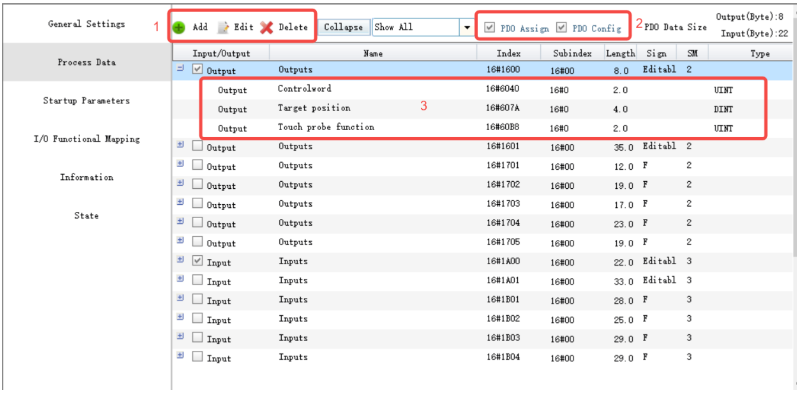
```
Start scanning
Parsing scanned data
Scan succeeded!
```

e. The configuration list is as follows. The SV680I in the configuration will be automatically associated with the motion control axis.



4. **Configuring PDO**

The Process Data interface is used to edit PDO. The interface is as follows:



PDO Data Size Output(Byte):8 Input(Byte):22

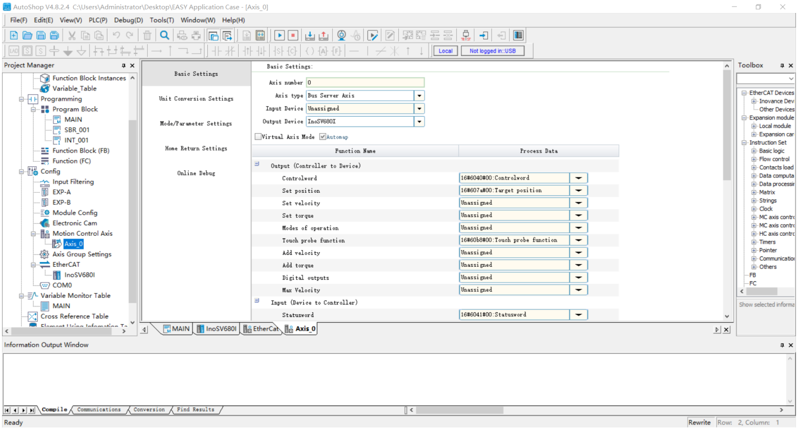
Input/Output	Name	Index	Subindex	Length	Sign	SM	Type
☑ Output	Outputs	16#1600	16#00	8.0	Editabl	2	
Output	Controlword	16#6040	16#0	2.0			UINT
Output	Target position	16#607A	16#0	4.0			DINT
Output	Touch probe function	16#60B8	16#0	2.0			UINT
□ Output	Outputs	16#1601	16#00	35.0	Editabl	2	
□ Output	Outputs	16#1701	16#00	12.0	F	2	
□ Output	Outputs	16#1702	16#00	19.0	F	2	
□ Output	Outputs	16#1703	16#00	17.0	F	2	
□ Output	Outputs	16#1704	16#00	23.0	F	2	
□ Output	Outputs	16#1705	16#00	19.0	F	2	
☑ Input	Inputs	16#1A00	16#00	22.0	Editabl	3	
□ Input	Inputs	16#1A01	16#00	33.0	Editabl	3	
□ Input	Inputs	16#1B01	16#00	28.0	F	3	
□ Input	Inputs	16#1B02	16#00	25.0	F	3	
□ Input	Inputs	16#1B03	16#00	29.0	F	3	
□ Input	Inputs	16#1B04	16#00	29.0	F	3	

PDOs include output PDOs and input PDOs in terms of data flow direction. The output PDO represents the process data sent by the EtherCAT master station to the EtherCAT slave station, such as the control word 0x6040. The input PDO represents the process data sent by the EtherCAT slave station to the master station. Each slave station may have multiple groups of PDOs or a single group of PDOs, as shown in the above figure. Some PDOs can be added and deleted.

PDO control according to control requirements in process data.

5. **Configuring axis parameters**

a. In Genera Setting, you can set the axis type and select the physical driving device.. The interface is as shown below.



Basic Settings:

Axis number: 0

Axis type: Bus Server Axis

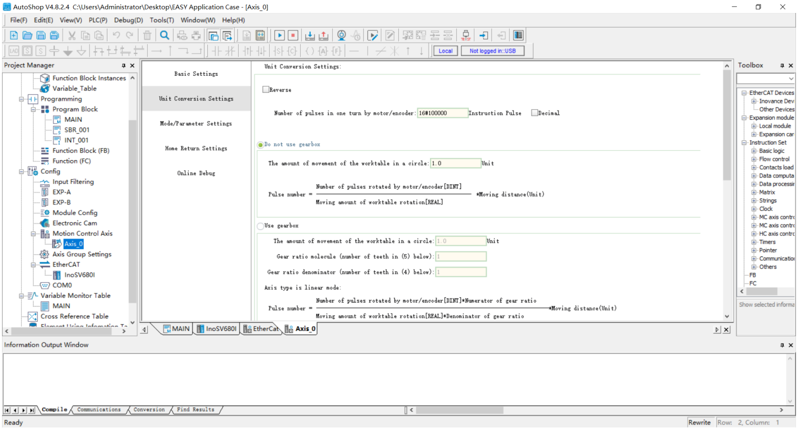
Input Device: Unassigned

Output Device: InoSV680I

□ Virtual Axis Mode ☑ Automap

Function Name	Process Data
Output (Controller to Device)	
Controlword	16#6040#00:Controlword
Set position	16#607a#00:Target position
Set velocity	Unassigned
Set torque	Unassigned
Modes of operation	Unassigned
Touch probe function	16#60b8#00:Touch probe function
Add velocity	Unassigned
Add torque	Unassigned
Digital outputs	Unassigned
Max Velocity	Unassigned
Input (Device to Controller)	
Statusword	16#6041#00:Statusword

b. In Scaling, select 16#4000000 for the 26-bit encoder and 16#800000 for the 23-bit encoder.



Unit Conversion Settings:

□ Reverse

Number of pulses in one turn by motor/encoder: 16#100000 Instruction Pulse □ Decimal

● Do not use gearbox

The amount of movement of the worktable in a circle: 1.0 Unit

Pulse number = Number of pulses rotated by motor/encoder[DINT] / Moving amount of worktable rotation[REAL] *Moving distance(Unit)

○ Use gearbox

The amount of movement of the worktable in a circle: 1.0 Unit

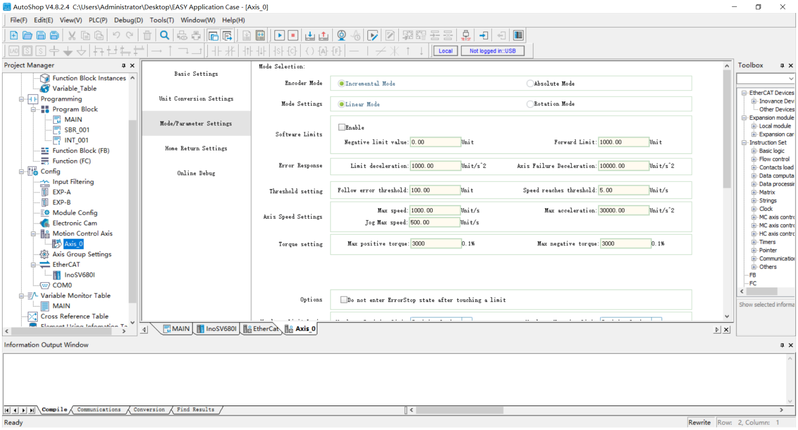
Gear ratio molecule (number of teeth in (5) below): 1

Gear ratio denominator (number of teeth in (4) below): 1

Axis type is linear mode:

Pulse number = Number of pulses rotated by motor/encoder[DINT]*Numerator of gear ratio / Moving amount of worktable rotation[REAL]*Denominator of gear ratio *Moving distance(Unit)

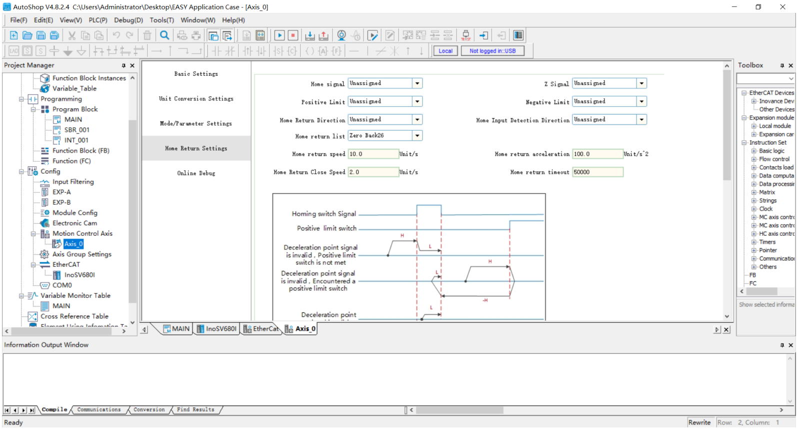
c. In General Setting, you can set the software position limit and the operation mode. The interface is as follows. Note that the parameter lists displayed vary with different axis types you select.



Mode Selection:

Encoder Mode	● Incremental Mode	○ Absolute Mode
Mode Settings	● Linear Mode	○ Rotation Mode
Software Limits	□ Enable Negative limit value: 0.00 Unit	Forward Limit: 1000.00 Unit
Error Response	Limit deceleration: 1000.00 Unit/s^2	Axis Failure Deceleration: 10000.00 Unit/s^2
Threshold setting	Follow error threshold: 100.00 Unit	Speed reaches threshold: 5.00 Unit/s
Axis Speed Settings	Max speed: 1000.00 Unit/s Jog Max speed: 500.00 Unit/s	Max acceleration: 30000.00 Unit/s^2
Torque setting	Max positive torque: 3000 0.1%	Max negative torque: 3000 0.1%
Options	□ Do not enter ErrorStop state after touching a limit	

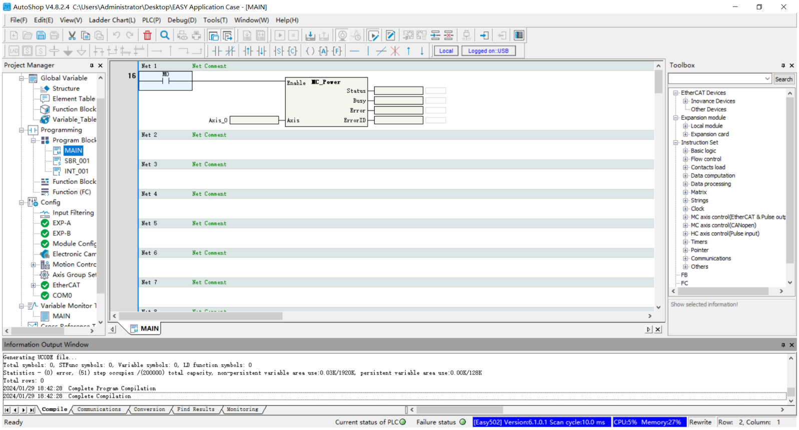
d. Select the homing mode according to actual needs. For details, see section *"Homing Mode"* in SV680-INT Series Servo Drive Function Guide for details.



Home signal	Unassigned	Z Signal	Unassigned
Positive Limit	Unassigned	Negative Limit	Unassigned
Home Return Direction	Unassigned	Home Input Detection Direction	Unassigned
Home return list	Zero Back26		
Home return speed	10.0 Unit/s	Home return acceleration	100.0 Unit/s^2
Home Return Close Speed	2.0 Unit/s	Home return timeout	50000

6. **Controlling servo drive operation**

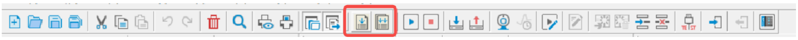
After configurations are done, you can control the servo drive operations through the PLC program.



```
Generating UCODE file...
Total symbols: 0, STFunc symbols: 0, Variable symbols: 0, LD function symbols: 0
Statistics - (0) error, (51) step occupies /(200000) total capacity, non-persistent variable area use:0.03K/1920K, persistent variable area use:0.00K/128K
Total rows: 0
2024/01/29 18:42:28  Complete Program Compilation
2024/01/29 18:42:28  Complete Compilation
```

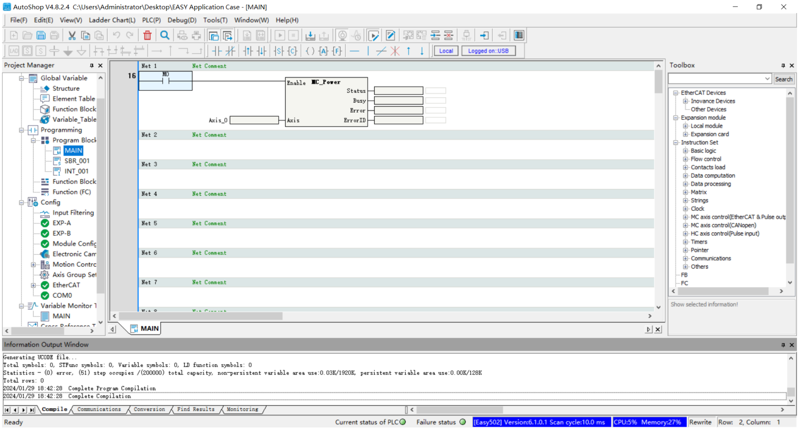
7. **Compiling**

   After compiling the program, click the icon indicated by the red square box to check whether the program is correct.



8. **Downloading and commissioning**

   After checking that the program is correct, download the program to PLC. The program can be activated after running. Click ▶ to switch the PLC to operation state.



```
Generating UCODE file...
Total symbols: 0, STFunc symbols: 0, Variable symbols: 0, LD function symbols: 0
Statistics - (0) error, (51) step occupies /(200000) total capacity, non-persistent variable area use:0.03K/1920K, persistent variable area use:0.00K/128K
Total rows: 0
2024/01/29 18:42:28  Complete Program Compilation
2024/01/29 18:42:28  Complete Compilation
```

PS00015535A01

Shenzhen Inovance Technology Co., Ltd.

www.inovance.com

Add.: Inovance Headquarters Tower, High-tech Industrial Park,
Guanlan Street, Longhua New District,
Shenzhen 518000, P.R. China
Tel: (0755) 2979 9595 Fax: (0755) 2961 9897

Suzhou Inovance Technology Co., Ltd.

www.inovance.com

Add.: No.52, Tian'e Dang Road, Wuzhong District,
Suzhou 215104, P.R. China
Tel: (0512) 6637 6666 Fax: (0512) 6285 6720