EC-TX509-U8
Industrial Ethernet Communication Card
User Manual



SHENZHEN INVT ELECTRIC CO., LTD.

Preface

Overview

Thank you for choosing INVT industrial Ethernet communication card. This manual describes the product features, electrical connections, communication, as well as examples of communication with the PLC. To ensure that you install and operate the product properly, read this manual and the communication sections in the VFD operation manual carefully before you use the product.

This manual only describes how to operate the communication card and the related commands but does not provide details about the PROFINET, EtherCAT, PowerLink, EtherNet IP, Modbus TCP, and EtherNet UDP protocols. For more information about the protocols, read the related specialized articles or documentations.

Precautions

The expansion card can be installed and operated only by people who have taken part in professional training on electrical operation and safety knowledge, obtained the certification, and been familiar with all steps and requirements for installing, performing commissioning on, operating, and maintaining the device, and are capable of preventing all kinds of emergencies.

- Before installing, removing, or operating the communication card, read the safety precautions described in this manual and the variable-frequency drive (VFD) operation manual carefully to ensure safe operation.
- We shall not be liable or responsible for any equipment damage or physical injury or death caused due to your or your customers' failure to follow the safety precautions.
- Before opening the VFD housing to install or remove the expansion card, disconnect all power supplies of the VFD and ensure that the voltage inside the VFD is far lower than the human safety voltage. For details, see the description in the VFD operation manual. Severe personal injury or even death can result if the instruction is not followed.
- Store the communication card in a place that is dustproof and damp-proof, free from electric shocks and mechanical pressure.
- The expansion card is electrostatic sensitive. Take measurements to prevent electrostatic discharge when performing related operations.
- When installing this expansion card, tighten the screws to ensure that it is firmly fixed and properly grounded.

Change history

The manual is subject to change irregularly without prior notice due to product version upgrades or other reasons.

No.	Change description	Version	Release date
1	First release.	V1.0	Jan 2025

Terminology and abbreviations

CAN	Controller area network
COB	Communication object, a transmitted unit on a CAN network. Data can be transmitted through the whole network. A COB is part of a CAN message frame.
EDS	Electronic datasheet, an ASCII file for node configuration, required when a CANopen network is configured. An EDS file contains general information about nodes and their dictionary objects (parameters).
NMT	Network management, one of the CAN application-layer service elements in the CAN reference model. It is used for the initialization, configuration, and fault handling of a CAN network.
Object dictionary	Stores information about all COBs identified by a device.
PDO	Process data object, a type of COBs, used to transmit process data, such as control command, set values, state values, and actual values.
PDOn Tx	PDO command sent from the slave to the master; n indicates 1, 2, 3, or 4.
PDOn Rx	PDO command sent from the master to the slave; n indicates 1, 2, 3, or 4.
SDO	Service data object, a type of COB, used to transmit non-time key data, such as parameter values.
RO	Indicates read-only access.
RW	Indicates the read and write access.
SYNC	Indicates synchronous transmission.
Node-ID	Node ID, that is, address of a communication card.
0x	0x stands for hexadecimal; for example, 0x8F represents hexadecimal 8F.

Contents

1 Product confirmation.....1
1.1 Product features 1
2 PROFINET protocol 6
2.1 Overview 6
2.2 Product features 6
2.2.1 Supported functions.....6
2.2.2 Supported communication types.....6
2.2.3 Status indicator.....7
2.3 Electrical connection 8
2.4 Communication 9
2.4.1 Message format.....9
2.4.2 Communication 10
2.5 PLC communication example (S7-1200).....20
2.5.1 Parameter setup 20
2.5.2 Creating a project 24
2.5.3 Adding the GSD file 25
2.5.4 Configuring project basic information 26
2.5.5 Assigning a device name for the IO device (INVT communication card) 30
2.5.6 Saving, compiling, and downloading 31
2.5.7 Monitoring VFD parameters 33
3 EtherNet IP protocol 36
3.1 Overview 36
3.2 Product features 36
3.2.1 Supported functions.....36
3.2.2 Supported communication types.....36
3.2.3 Status indicator.....36
3.3 Electrical connection 38
3.4 Communication 39
3.4.1 Communication settings 39
3.4.2 Message format.....40
3.4.3 Communication 40
3.5 PLC communication example 1 (1769_L36ERMS).....51
3.5.1 Creating a project 51
3.5.2 Importing the EDS file.....51
3.5.3 Creating a device object 54
3.5.4 Using RSLinx Classic 57

3.5.5 Writing PLC programs....59
3.5.6 Host controller connection and program download....61
3.5.7 Setting the PLC IP address using Studio 5000 V31....63
3.5.8 DLR ring network configuration....63
3.6 PLC communication example 2 (NJ501-1400)....65
3.6.1 Hardware connection....65
3.6.2 Network Configurator software setting....66
3.6.3 Configuring Sysmac Studio software....71
3.6.4 Importing or exporting data tags....77
3.6.5 PLC program downloading and online monitoring....79
4 EtherCAT....87
4.1 Overview....87
4.2 Product features....87
4.2.1 Supported functions....87
4.2.2 Supported services....87
4.2.3 Status indicator....87
4.3 Electrical connection....89
4.4 Communication....90
4.4.1 CANopen over EtherCAT reference model....90
4.4.2 EtherCAT slave node information....91
4.4.3 EtherCAT state machine....91
4.4.4 PDO process data mapping....92
4.4.5 Network synchronization based on distributed clocks....94
4.5 CiA402 device profile....95
4.5.1 CANopen over EtherCAT state machine....95
4.5.2 Device run mode....99
4.6 PLC communication example 1 (TwinCAT2)....103
4.7 PLC communication example 2 (TM753)....110
5 Modbus TCP protocol....116
5.1 Overview....116
5.2 Product features....116
5.2.1 Supported functions....116
5.2.2 Supported communication types....116
5.2.3 Status indicator....116
5.3 Electrical connection....118
5.4 Communication....119
5.4.1 Communication settings....119
5.4.2 Message format....119
5.4.3 Modbus TCP communication....119
5.4.4 Data address definition....122

5.4.5 Fieldbus scale....127
5.4.6 Error message response....128
5.5 PLC communication example 1 (S7-1200)....129
5.6 PLC communication example 2 (TM753)....138
6 PowerLink protocol....147
6.1 Overview....147
6.2 Product features....147
6.2.1 Supported functions....147
6.2.2 Supported services....147
6.2.3 Status indicator....147
6.3 Electrical connection....149
6.4 Communication....150
6.4.1 PowerLink reference model....150
6.4.2 PowerLink slave station information....151
6.4.3 PowerLink state machine....151
6.4.4 PDO process data mapping....153
6.4.5 Clock synchronization....154
6.5 CiA402 device profile....155
6.5.1 CANopen over PowerLink state machine....155
6.5.2 Device run mode....159
6.6 PLC communication example (x20CP3684)....163
Appendix A EtherCAT object dictionary....180
Appendix B PowerLink object dictionary....188
Appendix C Related function codes....195

1 Product confirmation

Check the following after receiving a communication expansion card product:

- Whether the communication card is damaged.
- Whether the received communication card is the one you purchase according to the bar code label on the PCB.
- Whether all the following items are contained in the product package.
- One communication card, one tie wrap, one tie, one M3 screw, and one manual.
- If the communication card is damaged, a wrong model is delivered, or some items are missing, contact the supplier in a timely manner.
- Obtain the ESD file or xml file of the communication card from INVT.

1.1 Product features

- Supports protocol selection through function codes.
- Supports up to eight protocols, including PROFINET, EtherCAT, EtherNet IP, Modbus TCP, EtherNet UDP, and PowerLink, with future support for BACnet/IP and CC-Link IE communication protocols.
- Certain protocols support the simultaneous operation of monitoring functions, thereby fulfilling the on-site oscilloscope monitoring requirements.
- Equipped with two RJ45 ports.
- Reaches the communication rate of up to 100 Mbit/s, with a short communication cycle.
- Supports both linear and star network topologies, with certain protocols also accommodating ring network topology.
- It is recommended to use double-twisted shielded Category 5e Ethernet cables, with crystal heads equipped with iron shells to meet the grounding shield protection.

Figure 1-1 Product components

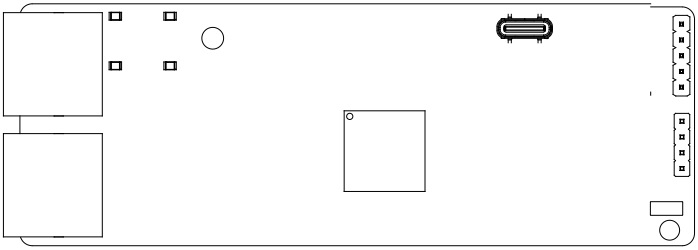


Table 1-1 Environmental requirements

Item	Requirement
Working temperature	-10–50°C
Storage temperature	-20–60.0°C
Relative humidity	5%–95% (No condensation)
Other weather conditions	No condensation, ice, rain, snow, or hail; Solar radiation < 700 W/m²
Air pressure	70–106kPa
Vibration and impact	5.8m/s² (0.6g) at the sine vibration of 9Hz to 200Hz

Figure 1-2 RJ45 interface

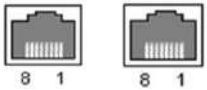


Table 1-2 RJ45 interface function

Pin	Name	Description
1	TX+	Transmit Data+
2	TX-	Transmit Data-
3	RX+	Receive Data+
4	n/c	Not connected
5	n/c	Not connected
6	RX-	Receive Data-
7	n/c	Not connected
8	n/c	Not connected

Figure 1-3 Expansion card configuration flowchart (taking GD350 for example)

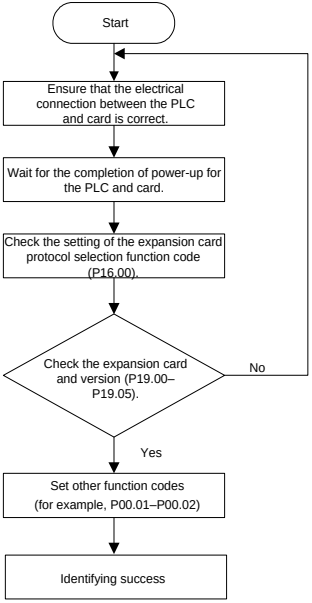


Table 1-3 Protocol selection (taking GD350 for example)

Function code	Protocol	Description
P16.00	PROFINET	0 (Factory setting)
	EtherCAT	1
	PowerLink	2
	EtherNet IP	3
	Modbus TCP	4
	EtherNet UDP	5
	PROFINET+EtherNet UDP	6
	EtherCAT+EtherNet UDP	7
	Reserved	8–15

Table 1-4 Protocol description

Protocol	Description
PROFINET	1. Supports the PROFINET protocol, accommodating PROFINET IO devices, medium redundancy protocol (MRP), and system redundancy protocol (S2). Equipped with the slave station GSDML configuration file, it can communicate with Siemens PLC and other master stations. 2. Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. This communication card supports up to 32 IOs. 3. Applicable to linear, star, and ring network topologies.
EtherCAT	1. Supports the CiA301 and CiA402 CoE protocols. Configured with a slave station XML configuration file, it can communicate with Beckhoff PLC, INVT AX controllers, and other master stations. 2. Supports PDO and SDO services, manufacturer-defined object dictionaries, and SDO reading/writing of VFD function codes, meeting the EtherCAT compliance testing certification requirements within the factory. 3. Applicable to linear, star, and ring network topologies. 4. Equipped with two RJ45 ports, designated for IN and OUT directions.
PowerLink	1. Supports the CiA402 protocol. Configured with a slave station XDD configuration file, it can communicate with B&R PLC and other master stations. 2. Supports PDO and SDO services, manufacturer-defined object dictionaries, and SDO reading/writing of VFD function codes. 3. Applicable to linear and star network topologies.
EtherNet IP	1. Supports ODVA standards and DLR ring protocol. When configured with a slave station EDS configuration file, it can communicate with Rockwell PLC and other master stations. 2. Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. This communication card supports up to 32 IOs. 3. Applicable to linear, star, and ring network topologies.
Modbus TCP	1. Supports the Modbus TCP protocol. A Modbus TCP slave

Protocol	Description
	station can communicate with multiple master stations simultaneously. It can communicate with Schneider PLC, INVT AX controllers, and other master stations. 2. Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. 3. Applicable to linear and star network topologies.
EtherNet UDP	1. Supports INVT Ethernet protocol, connecting to the INVT Workshop for monitoring and oscilloscope functionalities, allowing for multi-card network monitoring. 2. Applicable to linear and star network topologies.
PROFINET+ EtherNet UDP	Supports concurrent PROFINET and EtherNet UDP communications on the same network.
EtherCAT+ EtherNet UDP	Supports concurrent EtherCAT and EtherNet UDP communications on the same network, with EtherCAT required to remain online.

2 PROFINET protocol

2.1 Overview

The communication card using this protocol is defined as a PROFINET slave station, which can be used on VFDs that support PROFINET communication.

2.2 Product features

2.2.1 Supported functions

- Supports the PROFINET protocol and PROFINET IO devices.
- Supports the medium redundancy protocol (MRP) and system redundancy protocol (S2). Equipped with the slave station GSDML configuration file, it can communicate with Siemens PLC and other master stations.
- Equipped with two PROFINET IO ports, supporting 100M half/full duplex operating.
- Applicable to linear, star, and ring network topologies.
- Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. This communication card supports up to 32 IOs.

2.2.2 Supported communication types

Standard Ethernet channel: Standardized channels are non-real-time communication channels using the TCP/IP protocol, mainly used for device parameterization, configuration, and reading diagnostic data.

Real-time communication channel (RT): The RT channel uses optimized communication mechanisms for real-time data transfer, with higher priority than TCP (UDP)/IP protocols, ensuring that different sites in the network can exchange data under strict time requirements and meet millisecond-level bus cycles. The RT channel is typically used to transmit real-time information such as process data, alarm data, and other real-time information.

Real-time communication channel (IRT) (not supported): The IRT channel is implemented based on hardware, using embedded Switch-ASIC synchronous real-time exchange chips for data transmission. IRT communication further improves the efficiency of program, data transmission, and device synchronization by reducing the processing time of the communication stack software, with a transmission delay of less than 1ms and jitter of less than 1μs. Typical applications include high-precision motion control systems.

2.2.3 Status indicator

The PROFINET communication card provides six LED indicators to indicate its states. For details, see Table 2-1.

Table 2-1 Indicator description

Indicator	Color	Definition	Function
LED1 (RUN)	Green	Steady on	Communication established successfully, with normal IO data exchange.
		Blinking (on for 500ms, off for 500ms)	Communication established successfully, but without valid IO data exchange.
		Blinking (on for 100ms, off for 100ms)	In the communication configuration phase. For example, when DCP configuration commands are triggered, it will blink simultaneously with the ERR indicator.
		Steady off	The communication between the communication card and PLC is not in Online state.
LED2 (HOST)	Green	Steady on	The communication card is in the process of handshaking with the VFD.
		Blinking (on for 500ms, off for 500ms)	The communication card and VFD communicate normally. **Note:** After the handshaking is completed, it should blink regardless of whether there is data transmission between the communication card and the main control board.
		Steady off	The communication card is in the initialization or parameter configuration phase.
LED3 (DATA)	Green	Steady off	No data update or abnormal update between the communication card and main control board.
		Blinking (on for 500ms, off for 500ms)	The data update between the communication card and main control board is normal.
LED4	Red	Steady on	3.3V power indicator

Indicator	Color	Definition	Function
(POWER)			
LED5 (ERR)	Red	Steady off	No fault
		Blinking (on for 100ms, off for 100ms)	Communication establishment is abnormal.
LED6 (SYS)	Green	Blinking (on for 500ms, off for 500ms)	Communication card heartbeat indicator (communication card is running normally).

2.3 Electrical connection

The PROFINET communication card uses standard RJ45 interfaces, and its electrical connections are shown in Figure 2-1 and Figure 2-2.

Use CAT5, CAT5e, and CAT6 network cables for electrical wiring. When the communication distance is greater than 50m, use high-quality network cables that meet the high-quality standards.

Figure 2-1 Linear network topology electrical connection

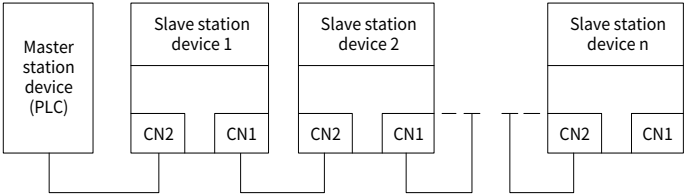
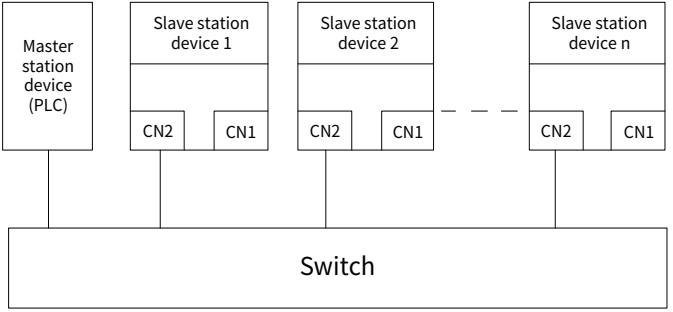
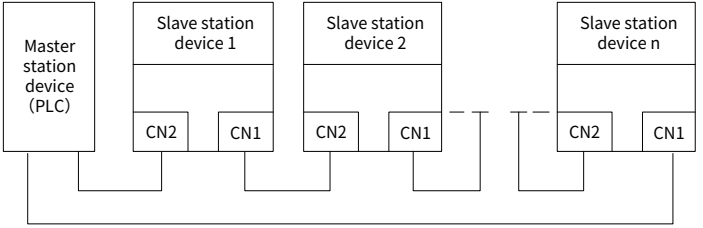


Figure 2-2 Star network topology electrical connection



Note: For the star network topology, you need to prepare switches.

Figure 2-3 Ring network topology electrical connection



2.4 Communication

2.4.1 Message format

Table 2-2 lists the RT frame (non-synchronous) structure.

Table 2-2 RT frame structure

Data header	Ethernet type	VLAN	Ethernet type	Frame identifier	RT user data	Cycle counter	Data status	Transmission status	FCS
-	2 bytes	2 bytes	2 bytes	2 bytes	36–1440 bytes	2 bytes	1 byte	1 byte	4 bytes
	0x8100	-	0x8892	-	-	-	-	-	-

Data header	Ethernet type	VLAN	Ethernet type	Frame identifier	RT user data	Cycle counter	Data status	Transmission status	FCS
	VLAN flag		-	-	-	APDU status			-
Data header									
7-byte preamble	1-byte synchronization information		6-byte source MAC address			6-byte destination MAC address			

Table 2-3 lists the IRT communication protocol and IRT frame (non-synchronous) structure.

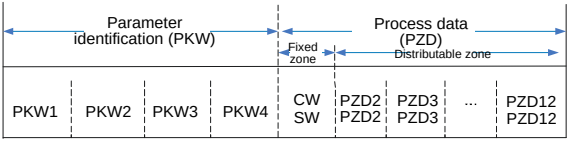
Table 2-3 IRT frame structure

Data header				Ethernet type	VLAN	Ethernet type	Frame identifier	IRT user data	FCS
7-byte preamble	1-byte synchronization	6-byte source MAC address	6-byte destination MAC address	2 bytes	2 bytes	2 bytes	2 bytes	36~1440 bytes	4 bytes

2.4.2 Communication

The PROFINET communication card supports 16-word input/output. Figure 2-4 shows the message format for transmitting data with the VFD.

Figure 2-4 Message structure



Through the preceding 32 inputs/outputs, you can set the reference parameters, monitor status values, send control commands and monitor operation status of the VFD, and read and write VFD function parameters.

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—control word (from the master to the slave; see Table 2-4 and Table 2-5)

SW—status word (from the slave to the master; see Table 2-7 and Table 2-8)

PZD—process data (user specified)

(When the PZD is output from the master to a slave, it is a reference value; and when the PZD is input from a slave to the master, it is an actual value.)

PZD zone: The PZD zone in communication messages is designed for controlling and monitoring VFDs. The master and slave always process the received PZD with the highest priority. The processing of PZD takes priority over that of PKW, and the master and slave always transmit the latest valid data on the interfaces.

CW and SW

Using CWs is the basic method for the fieldbus system to control VFD devices. It is sent from the fieldbus master to a VFD device. In this case, the adapter module functions as a gateway. The VFD device responds to the bit code information of the CW and feeds status information back to the master through an SW.

Reference value: The VFD device may receive control information in multiple channels, including analog and digital input terminals, VFD control panel, and communication modules (such as RS485 and CH-PA01 adapter modules). To enable the control over VFD devices through PROFINET, you need to set communication cards as the controllers of the VFD devices.

Actual value: An actual value is a 16-bit word that includes information about VFD device operation. The monitoring function is defined through VFD parameters. The conversion scale of an integer transmitted as an actual value sent to the master depends on the set function.

Note: A VFD device always checks the bytes of a CW and reference value.

Task message (Master station -> VFD)

Control word (CW): The first word in a PZD task message is the control word (CW) of VFD. The representation method can be selected according to function code P15.43. Table 2-4 and Table 2-5 provide descriptions for Goodrive350 series VFD CWs for example.

Table 2-4 Goodrive350 series VFD CWs in decimal

Bit	Name	Value	Description
0–7	Communication-based control command	1	FWD run
		2	Run reversely

Bit	Name	Value	Description
		3	Jog forward
		4	Jog reversely
		5	Stop
		6	Coast to stop
		7	Fault reset
		8	Stop jogging
		9	Stop in emergency manner
8	WIRTE ENABLE	1	Enable read and write (PKW1–PKW4)
9–10	Motor group setting	00	MOTOR GROUP1 SELECTION
		01	MOTOR GROUP2 SELECTION
11	Control mode switchover selection	1	Enable the switchover between torque control/ speed control
		0	No switchover
12	ELECTRIC CONSUMPTION CLEAR	1	Enabling the function for resetting power consumption to zero
		0	Disabling the function for resetting power consumption to zero
13	PRE-EXCIATION	1	Enable pre-exciting
		0	Disable pre-exciting
14	DC BRAKE	1	Enabling DC braking
		0	Disabling DC braking
15	HEARTBEAT REF (Heartbeat reference)	1	Enable heartbeat
		0	Disable heartbeat

Table 2-5 Goodrive350 series VFD CWs in binary

Bit	Name	Description	Priority
0	Forward running	0: Decelerate to stop 1: Forward running	1
1	Reverse running	0: Decelerate to stop 1: Reverse running	2
2	Fault reset	0: No 1: Fault reset	3
3	Coast to stop	0: No 1: Coast to stop	4
4	Forward jogging	0: No 1: Forward jogging	5
5	Reverse jogging	0: No 1: Reverse jogging	6
6	Stop jogging	0: No 1:Jogging stop	7
7	-	Reserved	-
8	Enable read and write	0: Disable	-

Bit	Name	Description	Priority
	(PKW1–4)	1: Enable	
9	-	Reserved	-
10	Stop in emergency manner	0: No 1: Emergency stop	0: Top priority
11–15	-	Reserved	-

Reference value (REF): The second to twelfth words in a PZD task message are the main settings (REF). The main frequency settings are provided by the main setting signal source. Table 2-6 lists the settings of Goodrive350 series VFD for example.

Table 2-6 Settings of Goodrive350 series VFD

Function code	Word	Value range	Default
P16.32	Received PZD2	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01Hz) 2: PID reference (-1000–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (-1000–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the motor rated current) 5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 8: Upper limit of braking torque (0–3000, in which 1000 corresponds to 100% of the motor rated current) 9: Virtual input terminal command (range: 0x000–0x3FF, with bit 9–bit 10 corresponding to S8/S7/S6/S5/HDIB/HDIA/S4/S3/S2/S1 in sequence) 10: Virtual output terminal command. Range: 0x00–0x0F (with bit3–bit0 corresponding to RO2/RO1/HDO/Y1 in sequence) 11: Voltage setting (special for V/F separation)	0
P16.33	Received PZD3		0
P16.34	Received PZD4		0
P16.35	Received PZD5		0
P16.36	Received PZD6		0
P16.37	Received PZD7		0
P16.38	Received PZD8		0
P16.39	Received PZD9		0
P16.40	Received PZD10		0
P16.41	Received PZD11		0
P16.42	Received PZD12		0

Function code	Word	Value range	Default
		(0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage) 12: AO1 output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%) 13: AO2 output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%) 14: High bit of position reference (signed) 15: Low bit of position reference (unsigned) 16–17: Reserved 18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) 19: Function parameter mapping (PZD2–PZD12 correspond to P14.49–P14.59) 20–31: Reserved	

Response message (VFD -> Master station)

Status word (SW): The first word in a PZD response message is the status word (SW) of VFD. The representation method can be selected according to function code P15.43. Table 2-7 and Table 2-8 provide descriptions for Goodrive350 series VFD SWs for example.

Table 2-7 Goodrive350 series VFD SWs in decimal

Bit	Name	Value	Description
0–7	Running status	1	Running forward
		2	Running reversely
		3	Stopped
		4	In fault
		5	VFD in POFF state
8	Bus voltage established	1	Ready to run
		0	Not ready to run
9–10	Motor group feedback	0	Feedback from motor 1
		1	Feedback from motor 2
11	Motor type feedback	1	Synchronous motor
		0	Asynchronous motor (AM)
12	Overload pre-alarm feedback	1	Overload pre-alarm
		0	No overload pre-alarm

Bit	Name	Value	Description
13–14	RUN/STOP MODE (Running mode selection)	0	Keypad-based control
		1	Terminal-based control
		2	Communication-based control
		3	Reserved
15	HEARTBEAT FEEDBACK (Heartbeat feedback)	1	Heartbeat feedback
		0	No heartbeat feedback

Table 2-8 Goodrive350 series VFD SWs in binary

Bit	Name	Description	Priority
0	Forward running	0: No 1: Running forward	1
1	Reverse running	0: No 1: Running reversely	2
2	Stop	0: No 1: Stopped	3
3	Fault	0: No 1: In fault	4
4	POFF	0: No 1: VFD POFF state	5
5	Pre-exciting	0: No 1: Pre-exciting	6
6–15	-	Reserved	-

Actual value (ACT): The second to twelfth words in a PZD task message are the main actual values. The main frequency actual values are provided by the main actual value signal source. Table 2-9 lists the actual status values of Goodrive350 series VFD for example.

Table 2-9 Actual status values of Goodrive350 series VFD

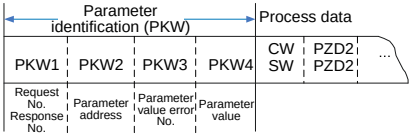
Function code	Word	Value range	Default
P16.43	Sent PZD2	0: Invalid 1: Running frequency (×100, Hz)	0
P16.44	Sent PZD3	2: Set frequency (×100, Hz) 3: Bus voltage (×10, V)	0
P16.45	Sent PZD4	4: Output voltage (×1, V) 5: Output current (×10, A)	0
P16.46	Sent PZD5	6: Actual output torque (×10, %) 7: Actual output power (×10, %)	0
P16.47	Sent PZD6	8: Rotation speed of running (×1, RPM) 9: Linear speed of running (×1, m/s)	0
P16.48	Sent PZD7	10: Ramp reference frequency 11: Fault code	0
P16.49	Sent	12: AI1 input (×100, V)	0

Function code	Word	Value range	Default
	PZD8	13: AI2 input (×100, V)	
P16.50	Sent PZD9	14: AI3 input (× 100, V) 15: HDIA frequency value (×1000, kHz)	0
P16.51	Sent PZD10	16: Terminal input status 17: Terminal output status	0
P16.52	Sent PZD11	18: PID reference (×10, %) 19: PID feedback (×10, %)	0
P16.53	Sent PZD12	20: Invalid 21: High bit of position reference (signed) 22: LSB of position reference (unsigned) 23: MSB of position feedback (signed) 24: LSB of position feedback (unsigned) 25: Status word 26: HDIB frequency value (×1000, kHz) 27: PG card pulse feedback count high bit 28: PG card pulse feedback count low bit 29: PG card pulse reference count MSB 30: PG card pulse reference count LSB 31: Function parameter mapping (PZD2–PZD12 correspond to P14.60–P14.70) 32: Status word 3 33–47: Reserved	0

PKW zone

PKW zone (parameter identification marks PKW1–value zone): PKW zone describes treatment of parameter identification interface, PKW interface is a mechanism which determine parameters transmission between two communication partners, such as reading and writing parameter values.

Figure 2-1 PKW zone



In the periodic communication, the PKW zone consists of four 16-bit words. The

following table lists the definition of each word.

First word PKW1 (16 bits)		
Bits 15–00	Task or response ID flag	0–7
Second word PKW2 (16 bits)		
Bits 15–00	Basic parameter address	0–247
Third word PKW3 (16 bits)		
Bits 15–00	Value (most significant word) of a parameter or error code of the returned value	00
Fourth word PKW4 (16 bits)		
Bits 15–00	Value (least significant word) of a parameter	0–65535

Note: If the master station requests the value of a parameter, the values in PKW3 and PKW4 of the message that the master station transmits to the VFD are no longer valid.

Task request and response: When transmitting data to a slave, the master uses a request number, and the slave uses a response number to accept or reject the request.

Table 2-10 Definitions of the task identification flag PKW1

Request No. (from the master to a slave)		Response signal	
Request	**Function**	**Acceptance**	**Rejection**
0	No task.	0	-
1	Requesting the value of a parameter	1, 2	3
2	Modifying a parameter value (one word) [modifying the value only on RAM]	1	3, 4
3	Modifying a parameter value (two words) [modifying the value only on RAM]	2	3, 4
4	Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM]	1	3, 4
5	Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM]	2	3, 4

Note: Request 3 "Modifying a parameter value (two words) [modifying the value only on RAM]" and request 5 "Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM] are not supported currently.

Table 2-11 Definitions of the response identification flag PKW1

Response No. (from a slave to the master)	
Response No.	**Function**
0	No response
1	Transmitting the value of a parameter (one word)
2	Transmitting the value of a parameter (two words)
3	The task cannot be executed and one of the following error number is returned: 1: Invalid command 2: Invalid data address 3: Invalid data value 4: Operation failure 5: Incorrect password 6: Incorrect data frame 7: Parameter read only 8: Parameter cannot be modified in running 9: Password protection 10: Function code mapping operation failure
4	Reserved

PKW examples:

Example 1: Reading the value of a parameter

You can set PKW1 to 1 and PKW2 to 0A to read a frequency set through keypad (the address of the frequency set through keypad is 10), and the value is returned in PKW4. The following data is in hexadecimal format.

Request (Master station -> VFD)

	PKW1		PKW2		PKW3		PKW4		CW		PZD2		PZD3		...	PZD12	
Request	00	01	00	0A	00	00	00	00	xx	xx	xx	xx	xx	xx	...	xx	xx

000A: Parameter address

0001: Request for parameter value reading

Response (VFD -> Master station)

	PKW1		PKW2		PKW3		PKW4		CW		PZD2		PZD3		...	PZD12	
Response	00	01	00	0A	00	00	13	88	xx	xx	xx	xx	xx	xx	...	xx	xx

1388: Parameter value in address 10

0001: Response (parameter value updated)

Example 2: Modifying the value of a parameter (on both RAM and EEPROM)

You can set PKW1 to 4 and PKW2 to 0A to modify a frequency set through keypad (the address of the frequency set through keypad is 10), and the value to be modified 1388H (50.00) is in PKW4.

Request (Master station -> VFD)

	PKW1		PKW2		PKW3		PKW4		CW		PZD2		PZD3		...	PZD12	
Request	00	04	00	0A	00	00	13	88	xx	xx	xx	xx	xx	xx	...	xx	xx

1388: Parameter value in address 10

0004: Parameter value to be modified

Response (VFD -> Master station)

	PKW1		PKW2		PKW3		PKW4		CW		PZD2		PZD3		...	PZD12	
Response	00	01	00	0A	00	00	13	88	xx	xx	xx	xx	xx	xx	...	xx	xx

0001: Response (parameter value updated)

PZD examples: The transmission of the PZD zone is implemented through VFD

function code settings. For details about related function codes, see the INVT operation manual.

Example 1: Reading the process data of VFD

In the example, PZD3 is set to "8: Running rotation speed" by setting P16.44 to 8. This operation is forcible. The setting remains until the parameter is set to another option.

Response (VFD -> Master station)

	PKW1		PKW2		PKW3		PKW4		CW		PZD2		PZD3		...	PZD12	
Response	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	00	0A	...	xx	xx

Example 2: Writing process data to a VFD device

In this example, "2: PID reference" is taken from PZD3 by setting P16.33 to 2. The parameter specified in each request frame is updated with the information contained in PZD3 until another parameter is specified.

Request (Master station -> VFD)

	PKW1		PKW2		PKW3		PKW4		CW		PZD2		PZD3		...	PZD12	
Response	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	00	00	...	xx	xx

Subsequently, the information contained in PZD3 is used as traction reference in each request frame until another parameter is specified.

2.5 PLC communication example (S7-1200)

This example illustrates how to use a PROFINET adapter module to communicate with a Siemens S7-1200 series PLC (using TIA Portal V13 as the configuration tool).

2.5.1 Parameter setup

Connect the PLC to your PC with a network cable. Set your PC IP address (such as 192.168.0.100) on your PC network. Set the IP address and name of the PLC.

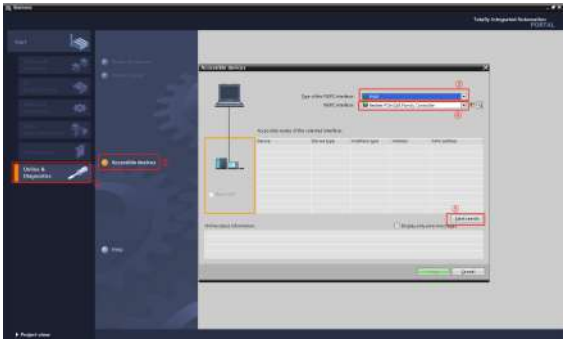
Open the TIA PORTAL V13 software.

① Click **Online and Diagnostics** on the left.

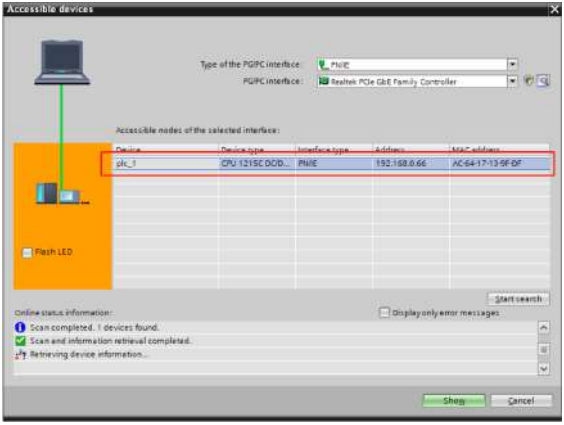
② Then click **Accessible devices**.

③ In the pop-up **Accessible devices** window, set **Type of PG/PC Interface** to **PN/IE**.
④ Select Ethernet port for **PG/PC Interface**.

⑤ Finally click **Start Search** to scan for connected PLC devices. See the following figure.



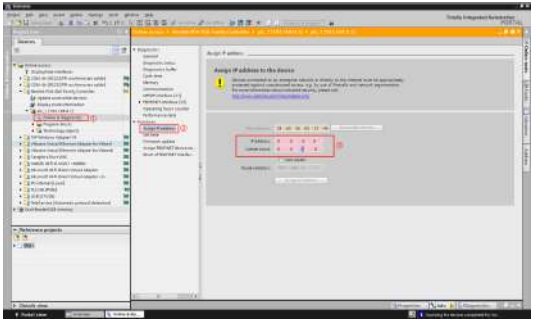
If the connection between the PLC and PC is normal, after the scan is completed, the PLC device will appear in the device column, as shown in the red box in the following figure. The device column will display the device, device type, and device MAC address. Click the **Show** button in the bottom right corner to access device settings.



① Click **Online and Diagnostics** in the device tree on the left.

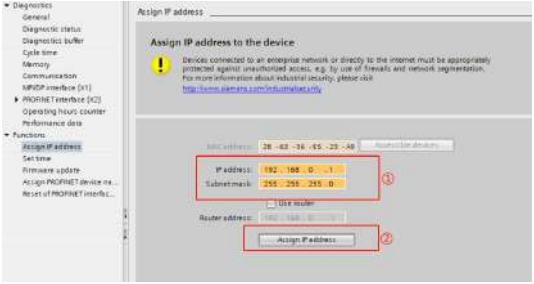
② Choose **Functions** > **Assign IP address** in the menu bar on the right.

③ Set the PLC IP address and subnet mask in the red box to ensure that the PC IP and PLC IP address are in the same network segment, as shown in the following figure.



① Set the PLC IP address to **192.168.0.1**, and subnet mask to **255.255.255.0** (**Use router** can be chosen, which means the router assigns IP).

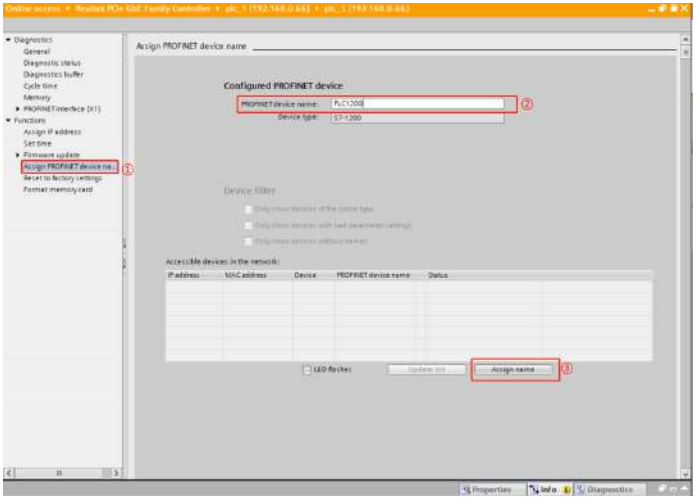
② After setting, click the **Assign IP address**. See the following figure.



① Click **Assign PROFINET device name**.

② On the right, enter the PLC name such as **PLC1215C**.

③ Click **Assign name**. See the following figure.



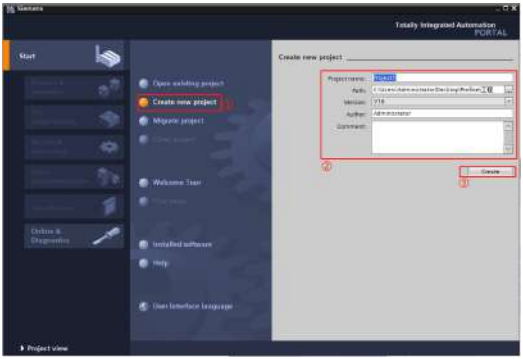
2.5.2 Creating a project

Double click the TIA Portal V13 icon to start the TIA Portal V13 project tool.

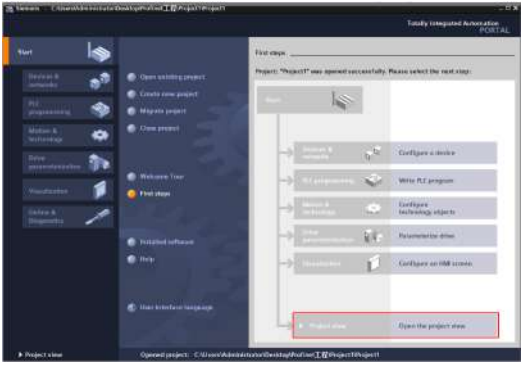
① Click **Create new project**.

② Enter project information such as **Project name**, **Path**, **Version**, **Author**, and **Comment**.

③ Click **Create**.



Then double click **Open Project View**. See the following figure.



2.5.3 Adding the GSD file

In the project view, choose **Option (N)** from the toolbar. Then choose **Manage general station description files (GSD)**.

① In the dialog box that appears, enter the source path of the GSD file. ② Select the GSD file. ③ Click **Install**.



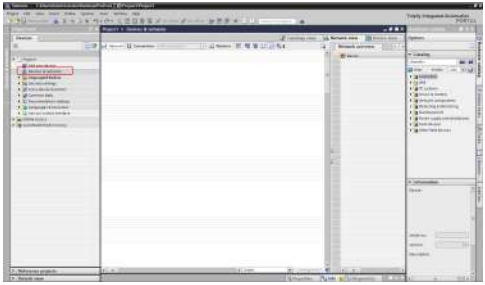
After the installation is successful, a message is displayed, indicating that the GSDML file has been installed successfully.



2.5.4 Configuring project basic information

1. Enter the device and network view interface.

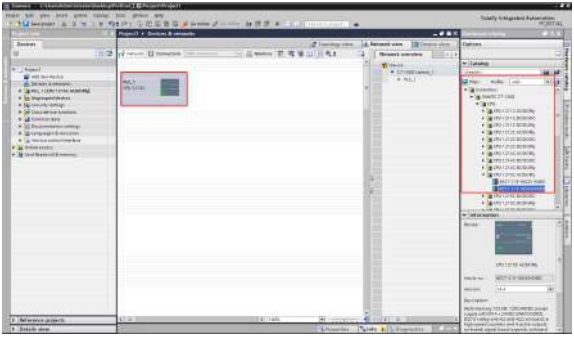
   Choose **Devices and networks** in the project tree on the left, and double click **Devices and networks** to enter the **Network overview** interface.



2. Add the project device and PROFINET network.

   A. Add **PLC S7-1215C** to the **Devices and Networks** view.

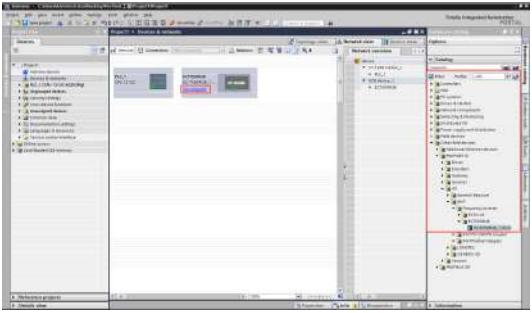
   Choose **Controllers** > **SIMATIC -1200** > **CPU** > **CPU 1215C AC/DC/Rly** > **6ES7 215-1BG40-0XB0** in the **Hardware catalog** panel on the right, and then double click or drag the **6ES7 215-1BG40-0XB0** icon to the project.



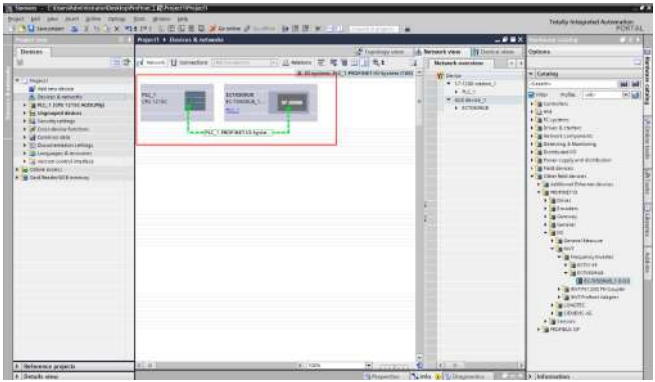
   B. Add the INVT communication card to the **Devices and networks** view.

   In the **Hardware catalog** panel on the right, choose **Other field devices** >

PROFINET IO > **I/O** > **INVT** > **Frequency Inverter** > **ECTX509U8**, and double click the **EC-TX509U8_1.0.0.0** icon or drag it to the view of **Devices & networks**. The communication card is displayed as **Not assigned**.

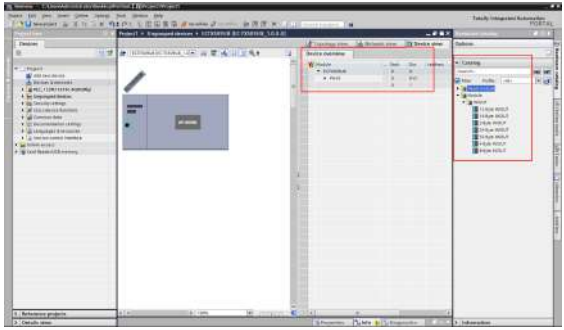


Click the **Not assigned** option of **EC-TX509U8_1.0.0.0**, and select the IO controller **PLC_1.PROFINET interface_1**. In the network view, the CPU and INVT PROFINET have been connected to the same PROFINET sub network.

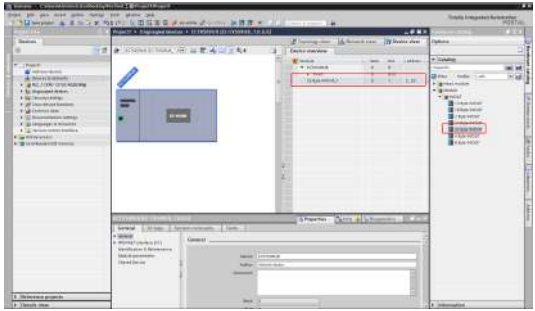


C. Add INVT I/O sub modules to the project.

Double click the **EC-TX509U8_1.0.0.0** icon in the **Devices & networks** view to enter the INVT device view interface.



On the right, choose **Hardware catalog** > **Module**, or double click or drag the **32 Byte IN/OUT** module to the empty area in the **Device view**, as shown in the following figure. Then the **32 Byte IN/OUT** module has been added to the project.

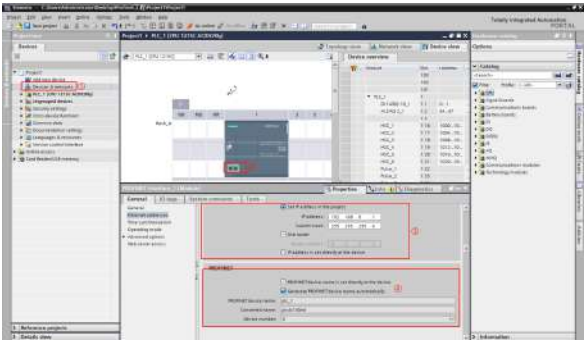


D. Set S7-1215C and INVT PROFINET parameters.

a) Set PLC S7-1215C parameters.

- Double click **Devices & networks** to enter the **Devices & networks** interface.
- Double click the PLC S7-1215C icon in the **Devices & networks** view to enter the PLC device view interface.
- Double click the network interface position in the PLC icon to enter the PLC device PROFINET interface_1 property editing interface. See the following figure.
- Click **General**, choose **Ethernet Address**, and set the PLC address and name (taking **192.168.0.1** and **PLC1215C** for example).

The following figure shows the operation procedure.

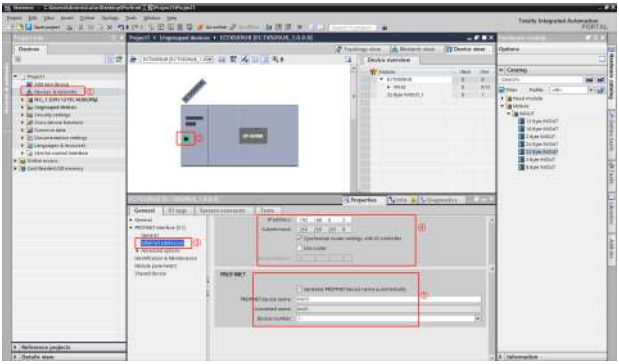


b) Set INVT PROFINET communication card parameters.

- Double click **Devices & networks** to enter the **Devices & networks** interface.
- Double click the **EC-TX509U6_1.0.0.0** icon to enter the communication card device view.
- Double click the network interface position of the INVT PROFINET communication card icon to enter the PROFINET interface editing interface.
- Click the **General** tab, and choose **PROFINET interface [X1]** > **Ethernet addresses**. Set the parameters of the INVT PROFINET communication card according to the parameters shown in the figure, that is, the IP address and device name of the communication card (using IP

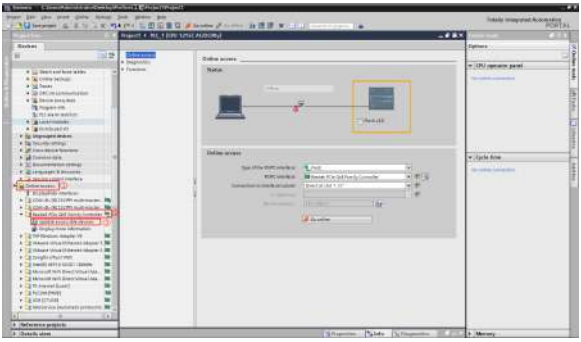
192.168.0.2 and name INVT1 as an example).

The following figure shows the operation procedure.



2.5.5 Assigning a device name for the IO device (INVT communication card)

After the CPU and INVT PROFINET communication card are successfully connected to the computer through a network cable: ① Click **Online access** on the left. ② Find the network card corresponding to the computer connected to the PLC and communication card. ③ Double click **Update accessible devices** and wait for TIA PORTAL to respond.



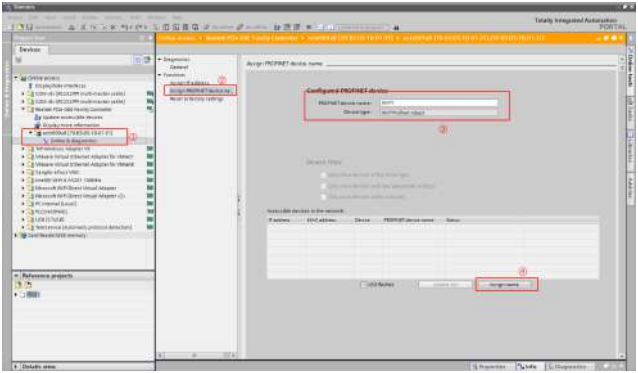
After the TIA PORTAL responds, the PLC and IVNT communication cards will be displayed as the accessible devices. See the following figure.

On all displayed devices, find the INVT communication card device and click it, for example, the device ectx509u8 in the figure.

Note: If the communication card is used for the first time, only the default device name can be found.

① Double click **Online and Diagnostics** to enter online diagnostics mode. ② Choose**Functions** > **Assign PROFINET device name**. ③ Access the **Assign PROFINET device name** interface. Set the device name and type, and click **Assign name**.

Note: The PROFINET communication card name that is set online must be the same as the PROFINET communication card name that is set during project configuration. Otherwise, devices cannot communicate through PROFINET.



2.5.6 Saving, compiling, and downloading

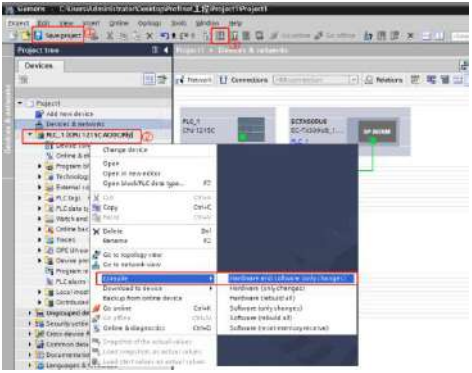
After completing the entire project configuration, download the configuration data to the PLC S7-1215C.

① Click **Save project** to save the entire project.

② Right-click **PLC_1 [CPU 1215C AC/DC/Rly]**, and then choose **Compile** > **Hardware and software (change only)** to compile the project.

③ Click the icon of download to device icon to download the project configuration

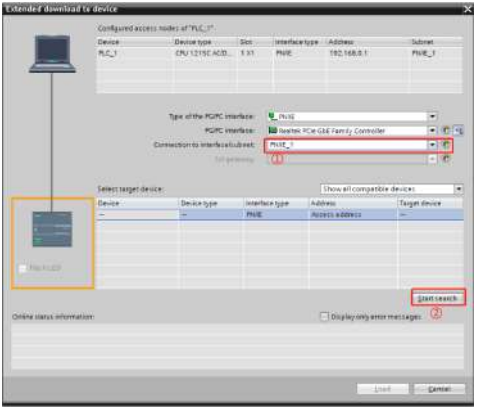
to the PLC.



In the download dialog box, search for the connected PLC device as shown in the following figure.

① Select **PN/IE_1** from the **Connection to interface/subnet** drop-down list box.

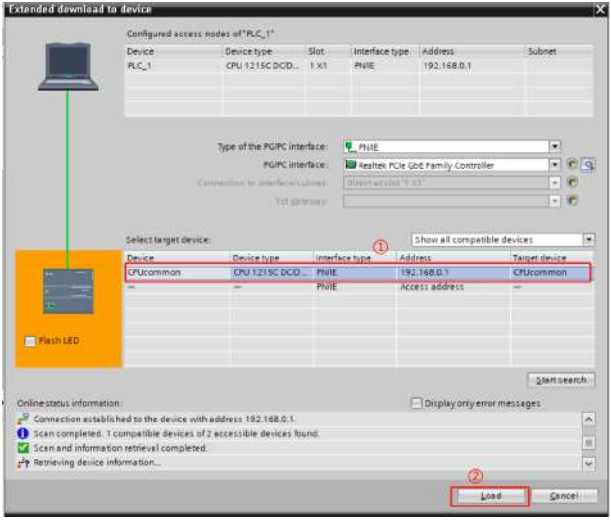
② Click **Start search** at the lower right corner to start scanning for PLC devices in the network.



After the search is completed, the PLC S7-1215C connected to the computer will be displayed in the **Show all compatible devices** list, as shown in the following figure.

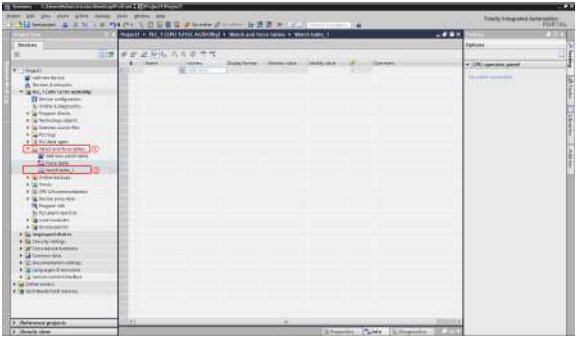
① Select the target PLC in the following figure.

② Click **Load** to download the configuration information and PLC program to the selected PLC.

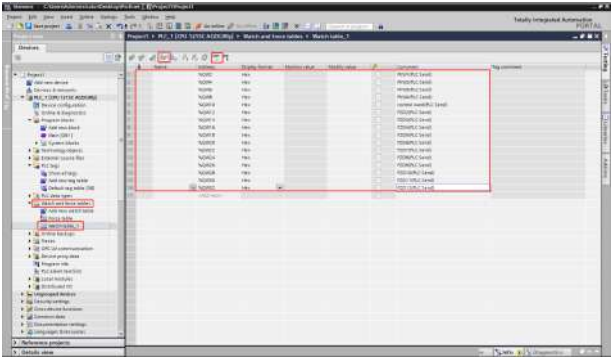


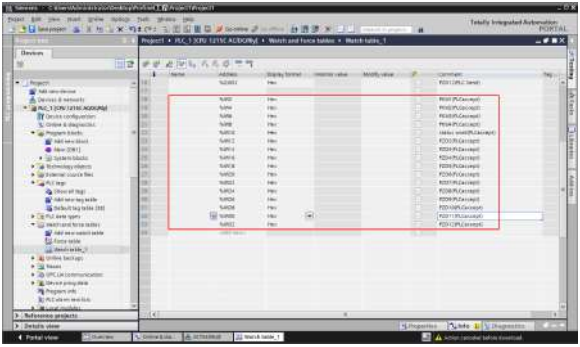
2.5.7 Monitoring VFD parameters

Choose **Watch and force tables** > **Add new watch table** in the project tree on the left.



Create target watch variables—PZD, PKW, CW and SW variables of the VFD in the newly created **Watch table_1** table.





After creating the monitoring variables, click **Watch all** or **Modify value** in the monitoring table to monitor values or modify values, thereby achieving the goal of monitoring the VFD parameters through the PLC.

3 EtherNet IP protocol

3.1 Overview

The communication card using this protocol is defined as an EtherNet IP slave station, which can be used on VFDs that support EtherNet IP communication.

3.2 Product features

3.2.1 Supported functions

- Supports the EtherNet IP protocol and EtherNet IP slave nodes.
- Supports ODVA standards and DLR ring protocol. When configured with a slave station EDS configuration file, it can communicate with Rockwell PLC and other master stations.
- Equipped with two EtherNet IP ports, supporting 10/100M half/full duplex operating.
- Applicable to linear, star, and ring network topologies.
- Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. This communication card supports up to 32 IOs.

3.2.2 Supported communication types

EtherNet IP uses the same application layer protocol CIP as DeviceNet and ControlNet,Therefore, they share the same object library and consistent industry standards, ensuring good compatibility.

CIP uses the User Datagram Protocol/Internet Protocol (UDP/IP) for connectionless control and information transmission, and the Transmission Control Protocol/Internet Protocol (TCP/IP) for connection-based transmission over Ethernet. It allows the transmission of both explicit and implicit messages. Implicit messages, which involve time-critical control information, are transmitted using UDP/IP. Explicit messages, which do not have strict time requirements and involve point-to-point information, are transmitted using TCP/IP. Explicit messages are used for configuring, downloading, and troubleshooting; implicit messages are used for real-time I/O data transmission.

3.2.3 Status indicator

The EtherNet IP communication card provides six indicators to indicate its states. For details, see Table 3-1.

Table 3-1 Indicator description

Indicator	Color	Definition	Function
LED1 (RUN)	Green	Steady on	The communication between the communication card and the PLC is online, and data exchange is allowed.
		Blinking (on for 500ms, off for 500ms)	Abnormal setting of the IP address for either the communication card or the PLC.
		Steady off	The communication between the communication card and PLC is not in Online state.
LED2 (HOST)	Green	Steady on	The communication card is in the process of handshaking with the VFD.
		Blinking (on for 500ms, off for 500ms)	The communication card and VFD communicate normally. **Note:** After the handshaking is completed, it should blink regardless of whether there is data transmission between the communication card and the main control board.
		Steady off	The communication card is in the initialization or parameter configuration phase.
LED3 (DATA)	Green	Blinking (on for 500ms, off for 500ms)	The data update between the communication card and main control board is normal.
		Steady off	No data update or abnormal update between the communication card and main control board.
LED4 (POWER)	Red	Steady on	3.3V power indicator
LED5 (ERR)	Red	Steady on	Failed to set up data communication between the communication card and PLC.

Indicator	Color	Definition	Function
		Blinking (on for 500ms, off for 500ms)	Incorrect PLC configuration.
		Blinking (on for 250ms, off for 250ms)	The communication card failed to send data to the PLC.
		Blinking (on for 125ms, off for 125ms)	The connection between the communication card and PLC timed out.
		Steady off	No fault
LED6 (SYS)	Green	Blinking (on for 500ms, off for 500ms)	Communication card heartbeat indicator (communication card is running normally).

3.3 Electrical connection

The EtherNet IP communication card adopts standard RJ45 interfaces, which can be used in a linear network topology and a star network topology, as shown in Figure 3-1 and Figure 3-2.

Use CAT5, CAT5e, and CAT6 network cables for electrical wiring. When the communication distance is greater than 50m, use high-quality network cables that meet the high-quality standards.

Figure 3-1 Linear network topology electrical connection

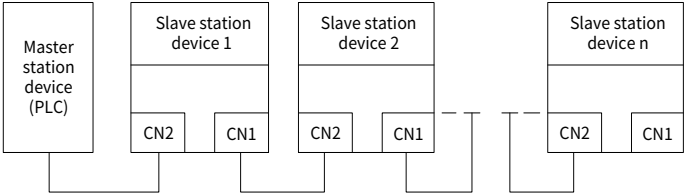
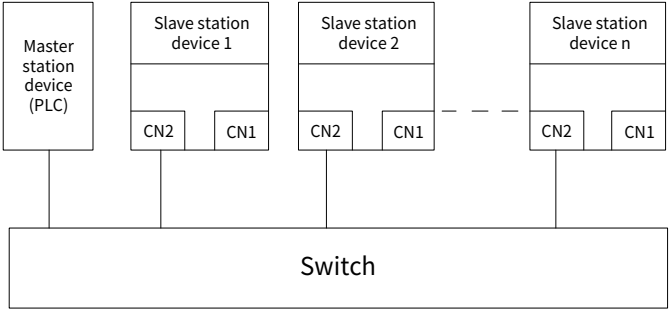
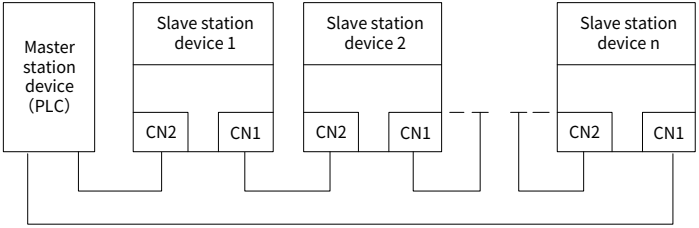


Figure 3-2 Star network topology electrical connection



Note: For the star network topology, you need to prepare EtherCAT switches.

Figure 3-3 Ring network topology electrical connection



3.4 Communication

3.4.1 Communication settings

The communication card can only be used as an EtherNet IP slave, and function codes should to be set on the VFD before communication. The procedure is as follows:

Step 1 Set the communication card IP address and subnet mask.

The factory IP address and subnet mask of each communication card are 192.168.0.20 and 255.255.255.0 respectively, which can be changed to a network segment address according to the actual requirements.

Step 2 Set the control method.

To control the VFD through EtherNet IP communication, set the control mode to EtherNet IP communication control. To be specific, set P00.01=2 and P00.02=3, which will implement the control of VFD start and stop. In short, if a value needs to be set through EtherNet IP communication, the corresponding function code should be modified to EtherNet IP communication control. For related function codes, see Appendix 2.

Note: After steps 1 and 2 are implemented properly, the communication card can communication properly. If a VFD needs to be controlled, related function nodes must be set and the control mode is EtherNet IP communication.

3.4.2 Message format

The TCP communication message is shown in Table 3-2.

Table 3-2 TCP communication message

Header of MAC layer	Header of IP layer	Header of TCP layer	Valid data	Trailer
14 bytes	20 bytes	20 bytes	0–1488 bytes	4 bytes

The UDP communication message is shown in Table 3-3.

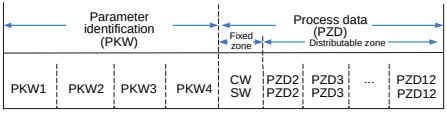
Table 3-3 UDP communication message

Header of MAC layer	Header of IP layer	Header of UDP layer	Valid data	Trailer
14 bytes	20 bytes	20 bytes	0–1488 bytes	4 bytes

3.4.3 Communication

The EtherNet IP communication card supports 16-word input/output. Figure 3-4 shows the message format for transmitting data with the VFD.

Figure 3-4 Message structure



Through the preceding 32 IOs, you can set the reference parameters, monitor status values, send control commands and monitor operation status of the VFD, and read and write VFD function parameters.

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—control word (from the master station to the slave station; see Table 3-4)

SW—status word (from the slave station to the master station; see Table 3-7)

PZD—process data (user specified)

(When the PZD is output from the master to a slave, it is a reference value; and when the PZD is input from a slave to the master, it is an actual value.)

PZD zone: The PZD zone in communication messages is designed for controlling and monitoring VFDs. The master and slave always process the received PZD with the highest priority. The processing of PZD takes priority over that of PKW, and the master and slave always transmit the latest valid data on the interfaces.

CW and SW

Using CWs is the basic method for the fieldbus system to control VFD devices. It is sent from the fieldbus master to a VFD device. In this case, the adapter module functions as a gateway. The VFD device responds to the bit code information of the CW and feeds status information back to the master through an SW.

Reference value: The VFD device may receive control information in multiple channels, including analog and digital input terminals, VFD control panel, and communication modules (such as RS485 and CH-PA01 adapter modules). To enable the control on VFD devices through EtherNet IP, you need to set communication cards as the controllers of the VFD devices.

Actual value: An actual value is a 16-bit word that includes information about VFD device operation. The monitoring function is defined through VFD parameters. The conversion scale of an integer transmitted as an actual value sent to the master depends on the set function.

Note: A VFD device always checks the bytes of a CW and reference value.

Task message (Master station -> VFD)

Control word (CW): The first word in a PZD task message is the control word (CW) of VFD.

When P15.43=0 (CW defined in decimal), Table 3-4 provides the definitions of GD350 series VFD CWs in decimal.

Table 3-4 Goodrive350 series VFD CWs in decimal

Bit	Name	Value	Description
0–7	Communication-based control command	1	FWD run
		2	Run reversely
		3	Jog forward
		4	Jog reversely
		5	Stop
		6	Coast to stop
		7	Fault reset
		8	Stop jogging
		9	Stop in emergency manner
8	WIRTE ENABLE	1	Enable read and write (PKW1–PKW4)
9–10	Motor group setting	00	MOTOR GROUP1 SELECTION
		01	MOTOR GROUP2 SELECTION
11	Control mode switchover selection	1	Enable the switchover between torque control/ speed control
		0	No switchover
12	ELECTRIC CONSUMPTION CLEAR	1	Enabling the function for resetting power consumption to zero
		0	Disabling the function for resetting power consumption to zero
13	PRE-EXCIATION	1	Enable pre-exciting
		0	Disable pre-exciting
14	DC BRAKE	1	Enabling DC braking
		0	Disabling DC braking
15	HEARTBEAT REF (Heartbeat reference)	1	Enable heartbeat
		0	Disable heartbeat

When P15.43=1 (CW defined in binary), Table 3-5 provides the definitions of GD350 series VFD CWs in binary.

Table 3-5 Goodrive350 series VFD CWs in binary

Bit	Name	Description	Priority
0	Forward running	0: Decelerate to stop 1: Forward running	1
1	Reverse running	0: Decelerate to stop 1: Reverse running	2
2	Fault reset	0: No 1: Fault reset	3

Bit	Name	Description	Priority
3	Coast to stop	0: No 1: Coast to stop	4
4	Forward jogging	0: No 1: Forward jogging	5
5	Reverse jogging	0: No 1: Reverse jogging	6
6	Stop jogging	0: No 1:Jogging stop	7
7	-	Reserved	-
8	Enable read and write (PKW1–4)	0: Disable 1: Enable	-
9	-	Reserved	-
10	Stop in emergency manner	0: No 1: Emergency stop	0: Top priority
11–15	-	Reserved	-

Reference value (REF): The second to twelfth words in a PZD task message are the main settings (REF). The main frequency settings are provided by the main setting signal source. Table 3-6 lists the main settings of Goodrive350 series VFD for example.

Table 3-6 Settings of Goodrive350 series VFD

Function code	Word	Value range	Default
P16.32	Received PZD2	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01Hz) 2: PID reference (-1000–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (-1000–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the motor rated current) 5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 8: Upper limit of braking torque (0–3000, in which 1000 corresponds to 100% of the motor rated current) 9: Virtual input terminal command (range: 0x000–0x3FF,	0
P16.33	Received PZD3		0
P16.34	Received PZD4		0
P16.35	Received PZD5		0
P16.36	Received PZD6		0
P16.37	Received PZD7		0
P16.38	Received PZD8		0
P16.39	Received PZD9		0
P16.40	Received PZD10		0

Function code	Word	Value range	Default
P16.41	Received PZD11	with bit 9–bit 10 corresponding to S8/S7/S6/S5/HDIB/HDIA/S4/S3/S2/S1 in sequence)	0
P16.42	Received PZD12	10: Virtual output terminal command. Range: 0x00–0x0F (with bit3–bit0 corresponding to RO2/RO1/HDO/Y1 in sequence) 11: Voltage setting (special for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage) 12: AO1 output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%) 13: AO2 output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%) 14: High bit of position reference (signed) 15: Low bit of position reference (unsigned) 16–17: Reserved 18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) 19: Function parameter mapping (PZD2–PZD12 correspond to P14.49–P14.59) 20–31: Reserved	0

Response message (VFD -> Master station)

Status word (SW): The first word in a PZD response message is the status word (SW) of VFD. When P15.43=0(SW defined in decimal), the VFD SW definitions are as follows.

Table 3-7 Goodrive350 series VFD SWs in decimal

Bit	Name	Value	Description
0–7	Running status	1	Running forward
		2	Running reversely
		3	Stopped
		4	In fault
		5	VFD in POFF state
8	Bus voltage established	1	Ready to run
		0	Not ready to run
9–10	Motor group feedback	0	Feedback from motor 1
		1	Feedback from motor 2

Bit	Name	Value	Description
11	Motor type feedback	1	Synchronous motor
		0	Asynchronous motor (AM)
12	Overload pre-alarm feedback	1	Overload pre-alarm
		0	No overload pre-alarm
13–14	RUN/STOP MODE	0	Keypad-based control
		1	Terminal-based control
		2	Communication-based control
		3	Reserved
15	HEARTBEAT FEEDBACK	1	Heartbeat feedback
		0	No heartbeat feedback

When P15.43=1(SW defined in binary), the VFD SW definitions are as follows.

Table 3-8 Goodrive350 series VFD SWs in binary

Bit	Name	Description	Priority
0	Forward running	0: No 1: Running forward	1
1	Reverse running	0: No 1: Running reversely	2
2	Stop	0: No 1: Stopped	3
3	Fault	0: No 1: VFD in fault	4
4	POFF	0: No 1: VFD in POFF state	5
5	Pre-exciting	0: No 1: VFD in pre-exciting state	6
6–15	-	Reserved	-

Actual value (ACT): The second to twelfth words in a PZD task message are the main actual values. The main frequency actual values are provided by the main actual value signal source.

Table 3-9 Actual status values of Goodrive350 series VFD

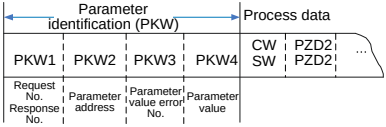
Function code	Word	Value range	Default
P16.43	Sent PZD2	0: Invalid 1: Running frequency (×100, Hz)	0
P16.44	Sent PZD3	2: Set frequency (×100, Hz) 3: Bus voltage (×10, V)	0
P16.45	Sent PZD4	4: Output voltage (×1, V) 5: Output current (×10, A)	0

Function code	Word	Value range	Default
P16.46	Sent PZD5	6: Actual output torque (×10, %) 7: Actual output power (×10, %)	0
P16.47	Sent PZD6	8: Rotation speed of running (×1, RPM) 9: Linear speed of running (×1, m/s)	0
P16.48	Sent PZD7	10: Ramp reference frequency 11: Fault code	0
P16.49	Sent PZD8	12: AI1 input (×100, V) 13: AI2 input (×100, V)	0
P16.50	Sent PZD9	14: AI3 input (× 100, V) 15: HDIA frequency value (×1000, kHz)	0
P16.51	Sent PZD10	16: Terminal input status 17: Terminal output status	0
P16.52	Sent PZD11	18: PID reference (×10, %) 19: PID feedback (×10, %)	0
P16.53	Sent PZD12	20: Invalid 21: High bit of position reference (signed) 22: Low bit of position reference (unsigned) 23: MSB of position feedback (signed) 24: LSB of position feedback (unsigned) 25: Status word 26: HDIB frequency value (×1000, kHz) 27: PG card pulse feedback count high bit 28: PG card pulse feedback count low bit 29: PG card pulse reference count MSB 30: PG card pulse reference count LSB 31: Function parameter mapping (PZD2–PZD12 correspond to P14.60–P14.70) 32: Status word 3 33–47: Reserved	0

PKW zone

PKW zone (parameter identification marks PKW1–value zone): PKW zone describes treatment of parameter identification interface, PKW interface is a mechanism which determine parameters transmission between two communication partners, such as reading and writing parameter values.

Figure 3-5 PKW zone



In the periodic communication, the PKW zone consists of four 16-bit words. The following table lists the definition of each word.

First word PKW1 (16 bits)		
Bits 15–00	Task or response ID flag	0–7
Second word PKW2 (16 bits)		
Bits 15–00	Basic parameter address	0–247
Third word PKW3 (16 bits)		
Bits 15–00	Value (most significant word) of a parameter or error code of the returned value	00
Fourth word PKW4 (16 bits)		
Bits 15–00	Value (least significant word) of a parameter	0–65535

Note: If the master station requests the value of a parameter, the values in PKW3 and PKW4 of the message that the master station transmits to the VFD are no longer valid.

Task request and response: When transmitting data to a slave, the master uses a request number, and the slave uses a response number to accept or reject the request.

Table 3-10 Definitions of the task identification flag PKW1

Request No. (from the master to a slave)		Response signal	
Request	**Function**	**Acceptance**	**Rejection**
0	No task.	0	-
1	Requesting the value of a parameter	1, 2	3
2	Modifying a parameter value (one word) [modifying the value only on RAM]	1	3, 4
3	Modifying a parameter value (two words) [modifying the value only on RAM]	2	3, 4
4	Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM]	1	3, 4
5	Modifying a parameter value (two words)	2	3, 4

Request No. (from the master to a slave)		Response signal	
Request	Function	Acceptance	Rejection
	[modifying the value on both RAM and EEPROM]		

Note: Request 3 "Modifying a parameter value (two words) [modifying the value only on RAM]" and request 5 "Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM] are not supported currently.

Table 3-11 Definitions of the response identification flag PKW1

Response No. (from a slave to the master)		
Response No.	Function	
0	No response	
1	Transmitting the value of a parameter (one word)	
2	Transmitting the value of a parameter (two words)	
3	The task cannot be executed and one of the following error number is returned:	
	1: Invalid command 2: Invalid data address 3: Invalid data value 4: Operation failure 5: Incorrect password	6: Incorrect data frame 7: Parameter read only 8: Parameter cannot be modified in running 9: Password protection 10: Function code mapping operation failure
4	Reserved	

Mode specified by the standard ODVA protocol

The standard ODVA protocol specifies the data transmission format and defines the control word/status word. Table 3-12 provides the message format for data transmission with the VFD.

Table 3-12 Standard ODVA protocol specified transmission mode

No.	Input/Output	Data length (byte)	Format (word)
2	70/20	4	CW1/SW1+Speed_ref/act
3	71/21	4	CW2/SW2+Speed_ref/act
4	72/22	6	CW1/SW1+Speed_ref/act+Torque_ref/act
5	73/23	6	CW2/SW2+Speed_ref/act+Torque_ref/act

See Table 3-13, Table 3-14, Table 3-15, and Table 3-16 for the definitions of CW1,

SW1, CW2, and SW2.

Note: The ODVA protocol messages do not support P15.43=1 (binary definition).

Table 3-13 Standard ODVA protocol specified CW1

Bit	Name	Value	Description
0	FWD run	0	FWD run disabled
		1	FWD run
1	Reserved	-	-
2	Fault reset	0	No function
		1	Fault reset
3–15	Reserved	-	-

Table 3-14 Standard ODVA protocol specified SW1

Bit	Name	Value	Description
0	Fault state	0	No fault
		1	Fault occurred
1	Reserved	-	-
2	Run status	0	Not in FWD run
		1	FWD run
3–15	Reserved	-	-

Table 3-15 Standard ODVA protocol specified CW2

Bit	Name	Value	Description
0	FWD run	0	FWD run disabled
		1	FWD run
1	Run reversely	0	REV run disabled
		1	Run reversely
2	Fault reset	0	No function
		1	Fault reset
3–4	Reserved	-	-
5	Control reference source	0	Local control (keypad)
		1	Remote control (EtherNet IP communication)
6	Frequency reference source	0	Local reference (keypad)
		1	Remote reference (EtherNet IP communication)
7–15	Reserved	-	-

Table 3-16 Standard ODVA protocol specified extended SW2

Bit	Name	Value	Description
0	Fault	0	No fault
		1	Fault occurred
1	Overload pre-alarm feedback	0	No overload pre-alarm
		1	Overload pre-alarm
2	Run status word 1	0	Stop
		1	FWD run
3	Run status word 2	0	Stop
		1	Run reversely
4	Bus voltage established	0	Not ready to run
		1	Ready to run
5	Control reference source	0	Local control (keypad)
		1	Remote control (non keypad)
6	Frequency/torque reference source	0	Local control (keypad)
		1	Remote control (non keypad)
7	Reaching reference	0	Not reached
		1	Reached
8–15	Reserved	-	-

INVT extended data modes based on ODVA protocol

Table 3-17 lists the transmission message format for communication with VFDs in the four modes, based on the ODVA protocol and combined with the PZD process data defined by INVT.

Table 3-17 INVT extended data modes based on ODVA protocol

No.	Input/Output	Data length (byte)	Format (word)
6	74/24	24	CW1/SW1+Speed_ref/act+Empty word+PZD4–12
7	75/25	24	CW2/SW2+Speed_ref/act+Empty word+PZD4–12
8	76/26	24	CW1/SW1+Speed_ref/act+Torque_ref/act+PZD4–12
9	77/27	24	CW2/SW2+Speed_ref/act+Torque_ref/act+PZD4–12

In these four modes, the definitions of control word and status word are consistent with the "mode defined by the standard ODVA protocol", and PZD4–12 is consistent with the "custom mode defined by INVT", which will not be described further here.

3.5 PLC communication example 1 (1769_L36ERMS)

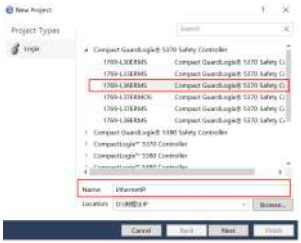
This example illustrates how to use an EtherNet IP adapter module to communicate with an Allen-Bradley PLC (model: 1769_L36ERMS) (by using Studio 5000 software as the configuration tool).

3.5.1 Creating a project

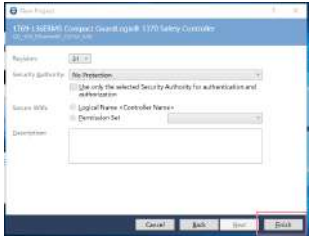
Use a printer cable or Ethernet cable to connect the computer and PLC, open the  software, and click **New Project**.



Choose the correct PLC model, fill in the project name, click **Next**, and then click **Finish**.



→



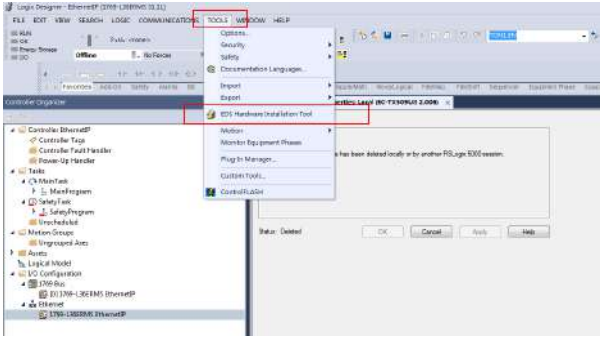
3.5.2 Importing the EDS file

The Electronic Data Sheet (EDS) file is used to specify the device properties of an EtherNet IP client. The client identifies a device by product code, device type, and main version.

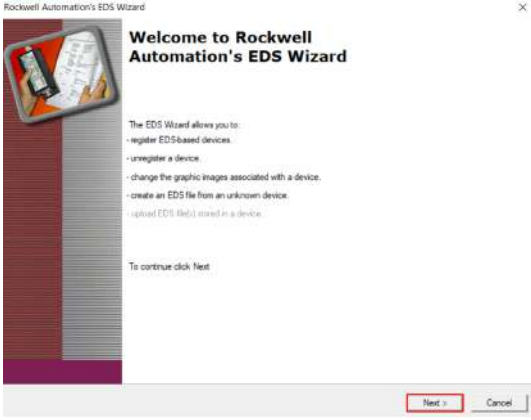
You can request the EDS file of the card from the supplier, or download it from our website.

Website: www.invt.com.cn, file name: EC-TX509U8_1.0.0.0.eds

Right click **TOOLS** and choose **EDS Hardware Installation Tool**.



Click **Next**.



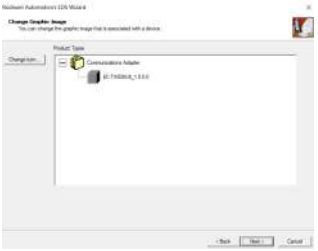
Select the method shown in the following figure and click **Next**.



Click **Browse** to select the EDS file you want to download, and then click **Next**.

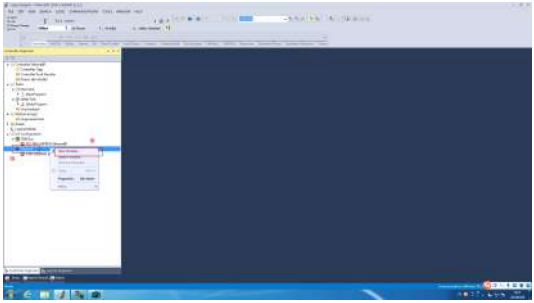


Click **Next** to complete the installation.



3.5.3 Creating a device object

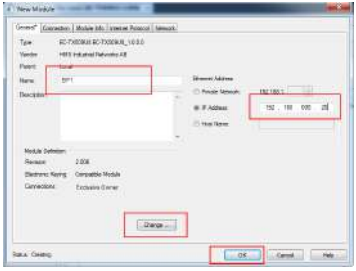
On the left, choose **I/O Configuration**, right click **Ethernet** option, then right click and choose **New Module**.



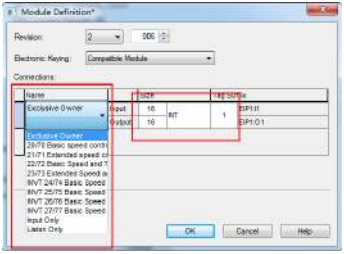
Choose **EC-TX509U8** and click **Create**.



Fill in the module name, and set the module IP address, which needs to be consistent with P16.58–P16.61 on the GD350 EtherNet IP card, otherwise communication will fail.



Click **Change** in the preceding figure, and select the protocol type adopted by the module; the IO format of each type is different and needs to be correspondingly selected, as shown in the following table. "Exclusive Owner" is used as an example for explanation.

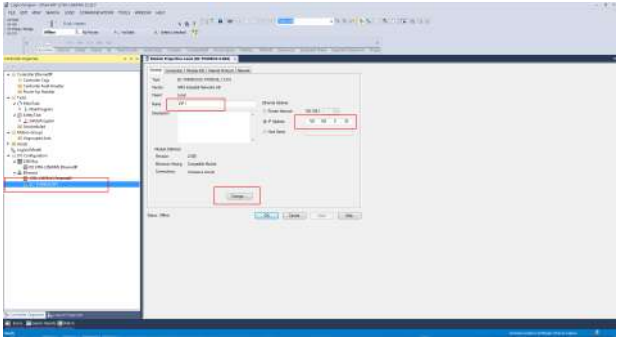


Name	Size	Format
Exclusive Owner	16	INT
20/70 Basic speed control	2	INT
21/71 Extended speed control	2	INT
22/72 Basic Speed and Torque control	3	INT
23/73 Extended Speed and Torque control	3	INT
INVT 24/74 Basic Speed Control plus Drive Parameters	12	INT
INVT 25/75 Basic Speed Control plus Drive Parameters	12	INT
INVT 26/76 Basic Speed Control plus Drive Parameters	12	INT
INVT 27/77 Basic Speed Control plus Drive Parameters	12	INT

Click **OK**, **Yes**, **OK**, **OK**, and **OK** in sequence.



After the module is successfully created, you can see and verify the device information under **Ethernet** under **I/O Configuration** on the left.

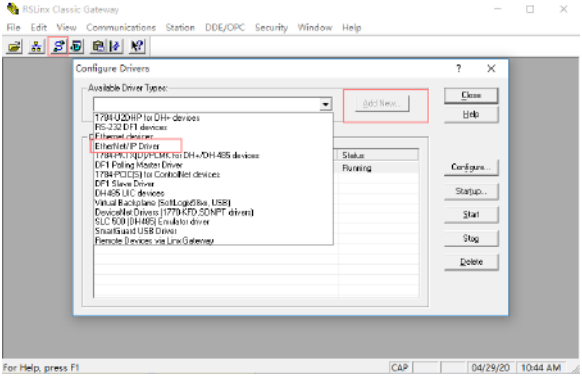


3.5.4 Using RSLinx Classic

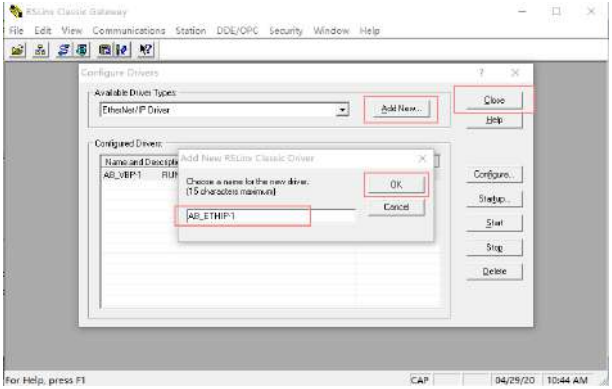
This software is used for the connection between the PC and PLC. Open the Rslinx Classic software.



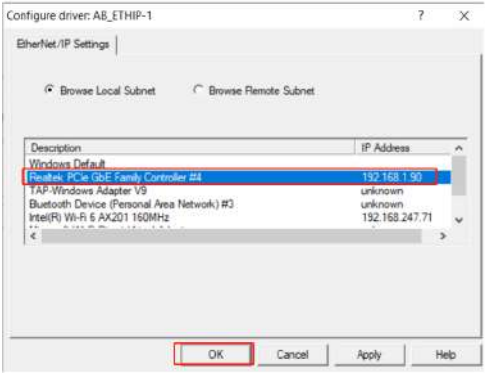
Click the S-shaped icon. Choose **EtherNet/IP Driver**, and click **Add New…**. In the pop-up **Configure Drivers** window, select **Ethernet/IP Driver** from the **Available Driver Types** dropdown menu.



Click **Add New**. In the pop-up **Add New RSLinx Classic Driver** window, click **OK**.

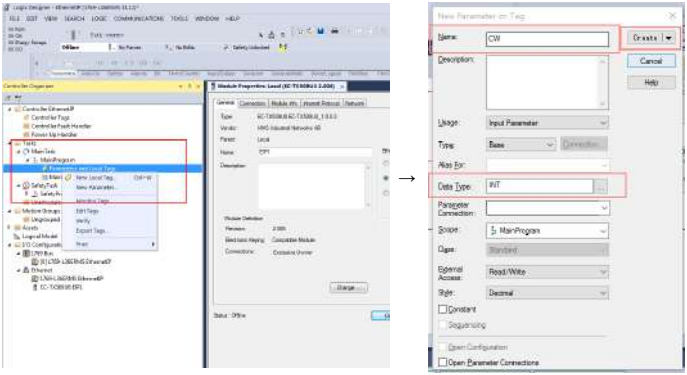


In the pop-up **Configure driver** window, select the computer network card and click **OK**.

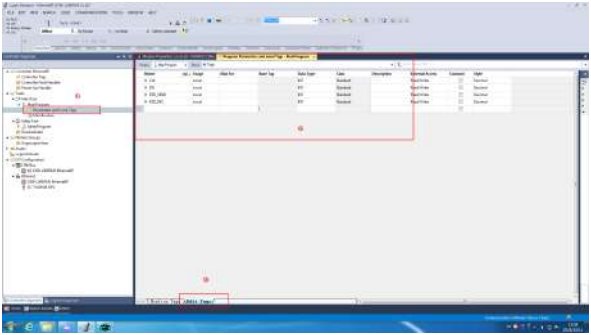


3.5.5 Writing PLC programs

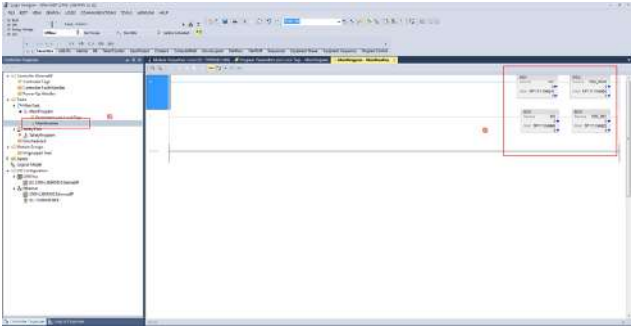
On the left, choose **Tasks** > **MainTask** > **MainProgram** > **MainRoutine** to enter the program editing interface. Right clicking **Parameters and Local Tag** can create global variables.



Example: Create four variables.



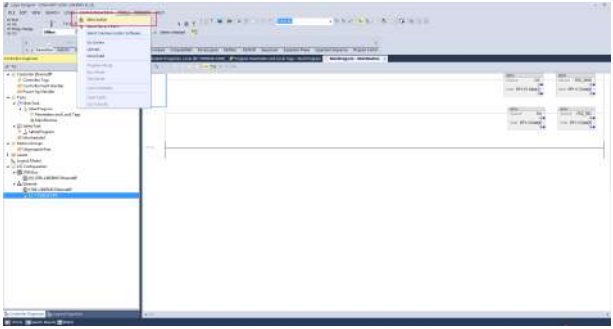
Double click **MainRoutine** and write the following program in the program interface.



3.5.6 Host controller connection and program download

Choose **OMMUNICATIONS** > **Who Active**. In the pop-up interface, click the PLC project under **USB**, and click **Download**.

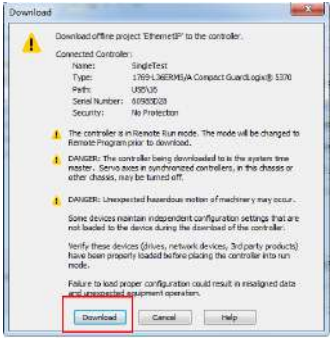
Note: The PLC position cannot be set to "RUN" at this time.



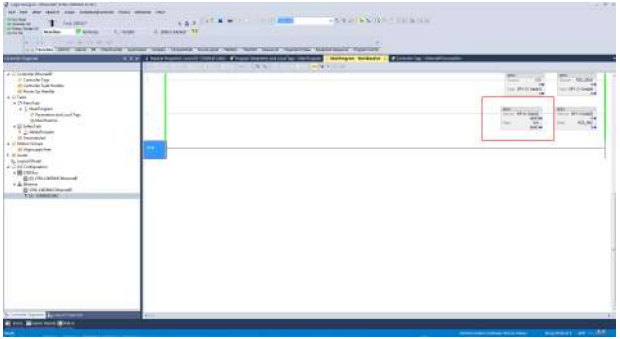
In the pop-up dialog box, select the connected PLC, and click **Download**; wait for the download to complete.



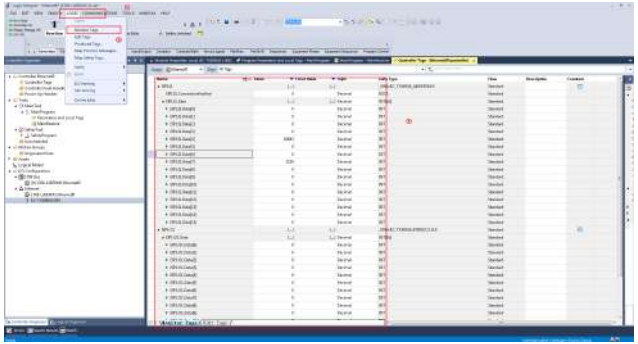
→



After the download is complete, the host controller enters the online status, and you can see the VFD feedback parameters on the program interface.

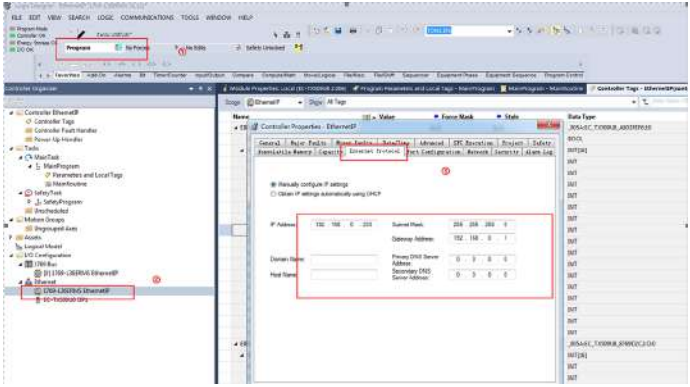


Choose **Logic** > **Monitor Tags** to monitor the real-time data sent by the PLC and uploaded by the VFD.



3.5.7 Setting the PLC IP address using Studio 5000 V31

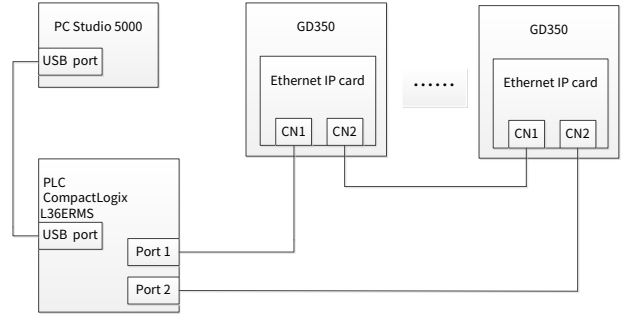
Make sure the PLC is in REM or PROG mode, first click on **1769-L36ERMS** at the bottom left, enter the **Controller Properties** interface, then click **Internet Protocol** to change the PLC IP address.



3.5.8 DLR ring network configuration

1. Set up using Logix Designer.

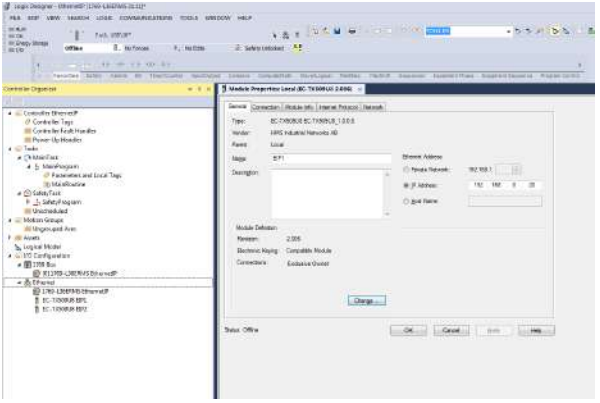
   Open Studio 5000 software, and use Allen-Bradley CompactLogix PLC equipped with ring networking capability. At least two GD350 EtherNet IP communication cards are required. More GD350 EtherNet IP communication cards can be added, but it is recommended to add 32 nodes on the DLR ring network at most, as shown in the following diagram for the connection method.



Note: The EDS file must be added.

2. Add the EtherNet IP communication cards in the Studio 5000.

   The adding method is the same as the line or star connection method.

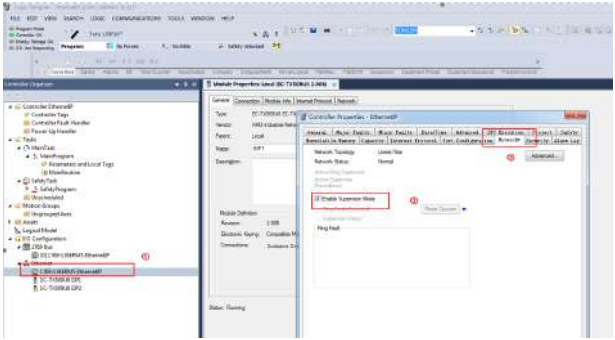


3. Enable the PLC ring network monitor function.

   Double click **1769-L36ERMS EtherNet IP** under the **I/O Configuration** folder, as shown in the following figure.

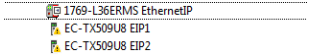
   Go to **Controller Properties** and select **Network***, choose **Enable Supervisor**

Mode, then click **OK**.



Note: The ring network monitoring function be enabled only when the PLC is in programming mode.

4. Return to Logix Designer and ensure that none of the communication cards are experiencing the following faults.



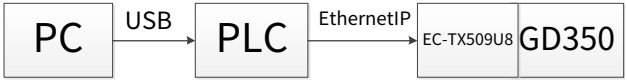
5. Download the project to the PLC, make the PLC online and in the programming mode.

3.6 PLC communication example 2 (NJ501-1400)

This example illustrates how to use an EtherNet IP adapter module to communicate with an ORMON PLC (model: NJ501-1400) (by using Studio 5000 software as the configuration tool).

3.6.1 Hardware connection

NJ501-1400 is equipped with a USB interface and an EthernetIP interface; the connection between the host controller and PLC is done through USB, with the EthernetIP port used as a communication connection method.



3.6.2 Network Configurator software setting

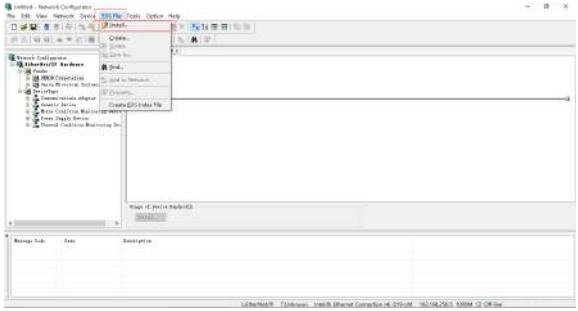
3.6.2.1 Starting the Network Configurator software

Start the Network Configurator software in the following directory as an administrator: 

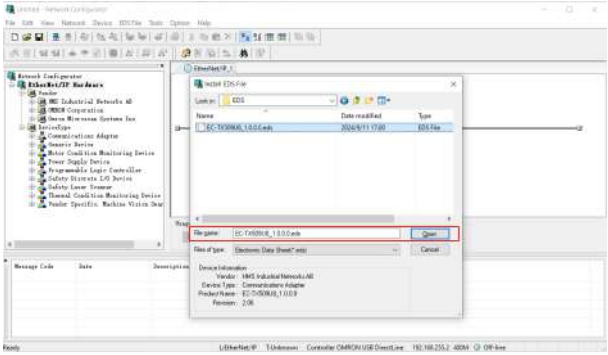
C:\Program Files (x86)\OMRON\CX-One\NetworkConfigurator\Program\NetConfigurator.exe

3.6.2.2 Uploading the EDS file

Choose **EDS File > Install**.



Add the EDS file **EC-TX509U8_1.0.0.0.eds**, click **Open**.



Click **Yes**.

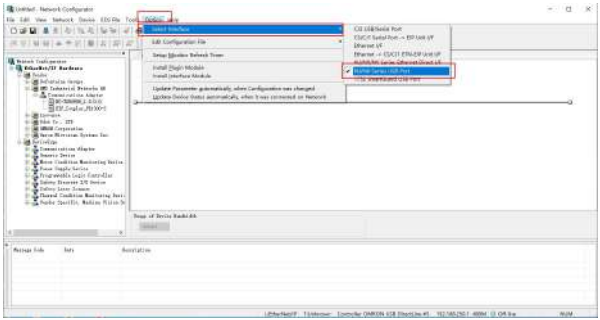


Add devices **NJ501-1400** and **EC-TX509U8_1.0.0.0** to the EtherNet IP bus at the following location. After completion, two devices will be displayed on the bus. The default IP addresses are **192.168.250.1** and **192.168.250.2**. Modify the GD350 function codes P16.58–P16.61 to 192.168.250.2.

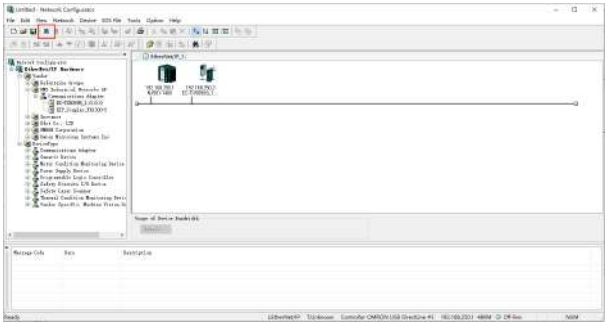


3.6.2.3 Connection setting

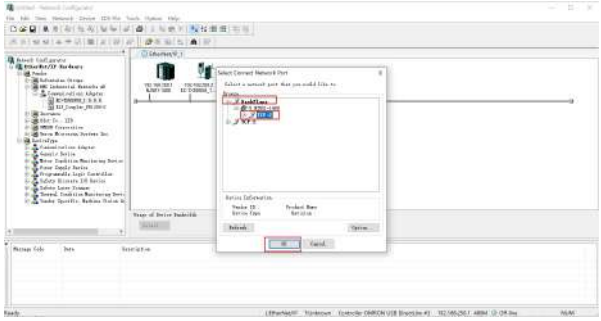
Choose **Option** > **Select Interface** > **NJ/NX Series USB Port**.



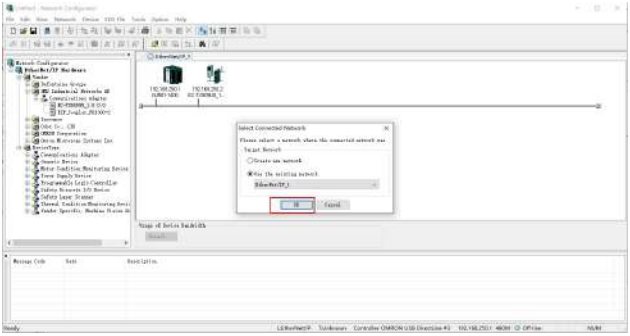
Click the icon for connection.



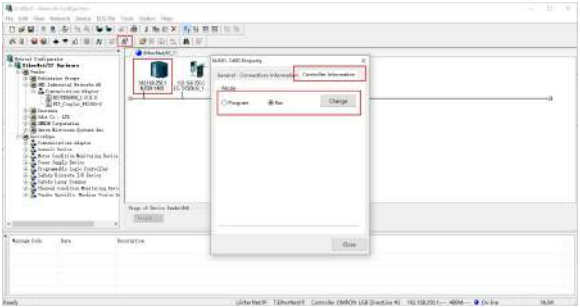
Choose **Backplane** > **0 NJ501-1400** > **TCP:2**, and then click **OK**.



Choose **Use the existing network**, select **EtherNet/IP_1**, and click **OK**. PLC connection is successful.

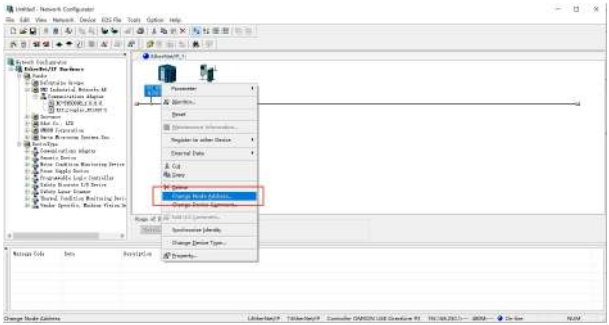


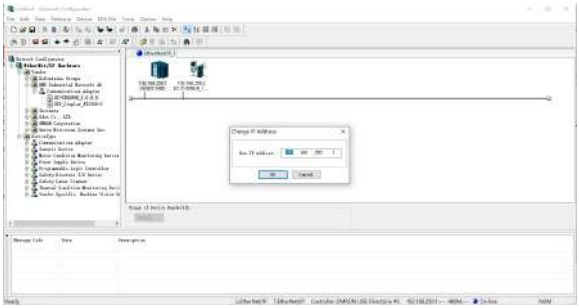
After the connection is successful, the blue indicator above the PLC device icon will turn on. Choose the PLC, click the icon for **Device Property**. On the **Controller Information** tab in the pop-up interface, you can switch between the **Program** and **Run** states for the PLC.



3.6.2.4 Changing the IP address

Right click the device icon, and choose **Change Node Address** to change the PLC IP address.

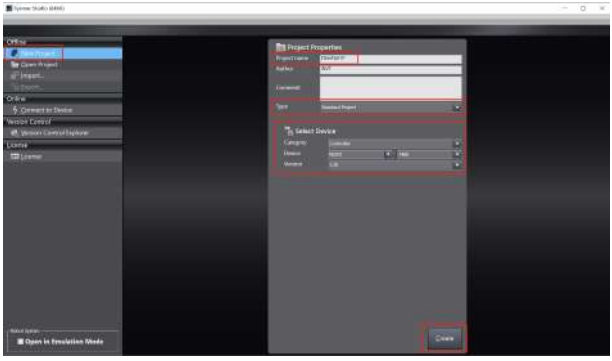




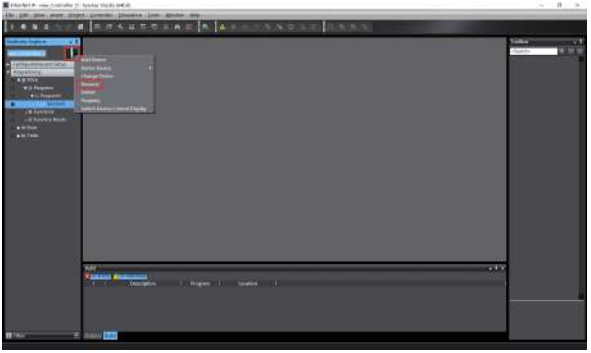
3.6.3 Configuring Sysmac Studio software

3.6.3.1 Creating a project

Double click the  icon to open the software. Choose **New Project**, enter the project name, select the device type, and click **Create**.

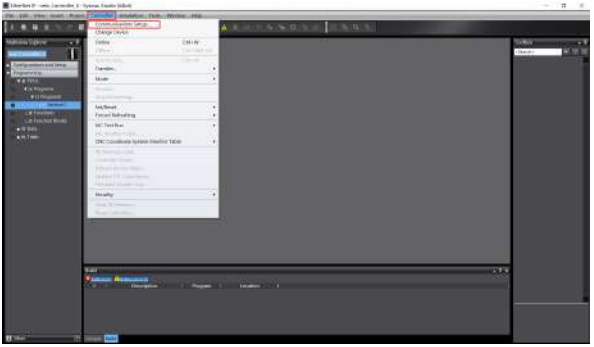


After the creating, enter the following interface, right-click the device icon, and choose **Rename** to change the device name (it can also remain unchanged).

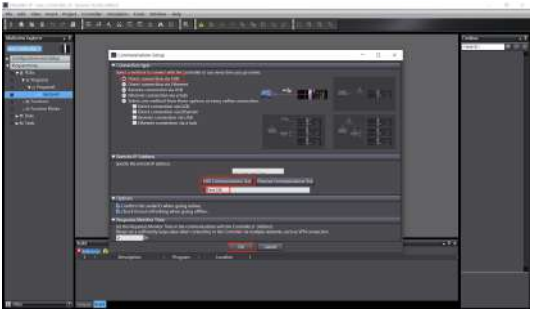


3.6.3.2 Connection setting

Choose **Controller > Communications Setup**.



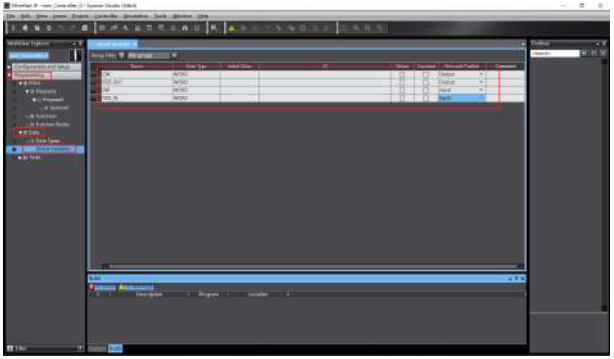
Choose **USB Direct connection via USB**, and click **USB Communications Test**. The status bar displays **Test OK**. Click **OK**.



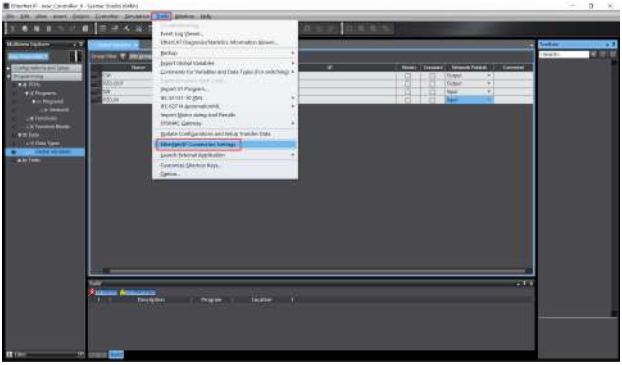
3.6.3.3 Setting data tags

On the left, choose **Programming** > **Data** > **Global Variables**, and add global variables according to actual needs. Note that **Data Type** is **WORD** and **Network Publish** is **Input** or **Output**.

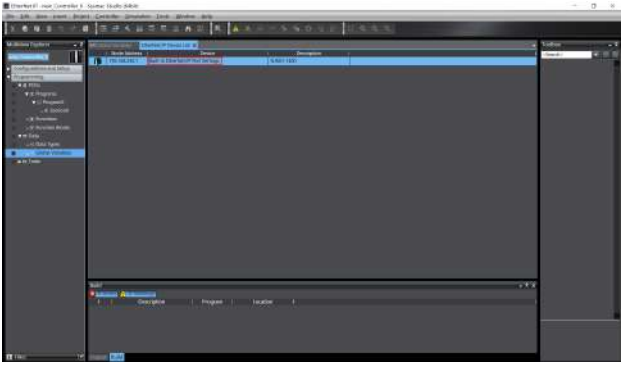
In this example, the "ODVA Basic speed control assembly" transmission mode is used to create four global variables.



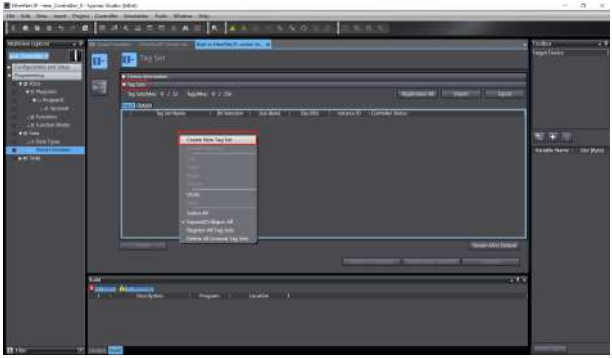
In the top menu bar, choose **Tools** > **EtherNet/IP Connection Settings**.



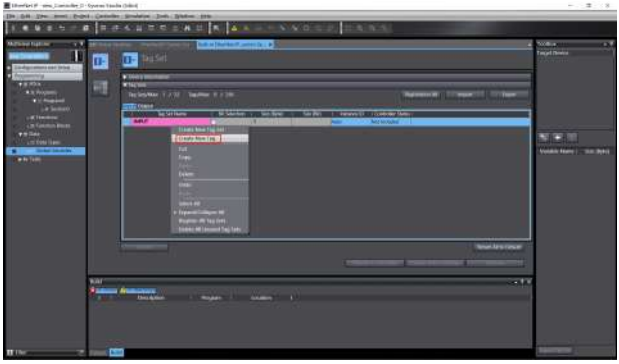
Double click **Built-in EtherNet/IP Port Settings**.

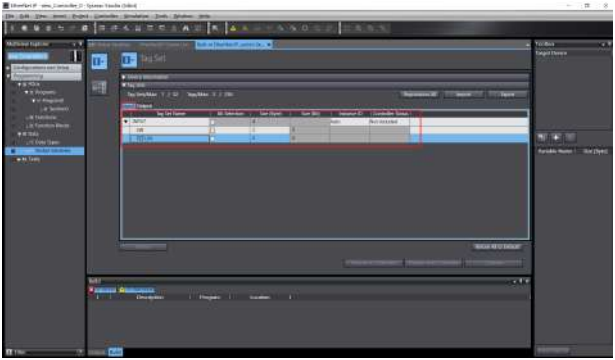


Right click on the blank area under **Tag Set**, and choose **Create New Tag Set**.

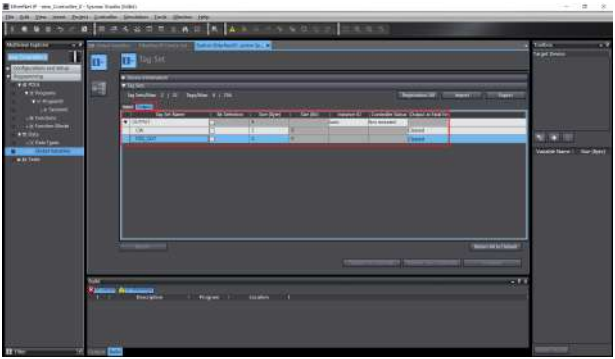


Enter the tag set name as **INPUT**, right click the tag set and choose **Create New Tag**, and add input global variables to the tag set **INPUT**. Pay attention to the order of data arrangement.





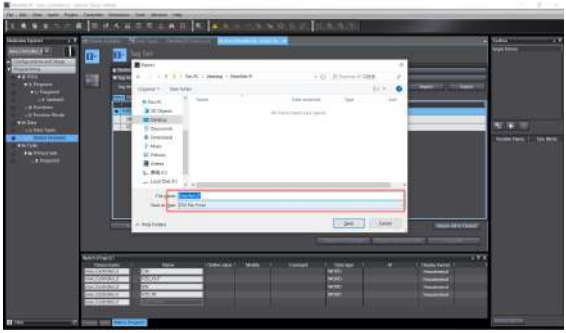
Similarly, set the tag set **OUTPUT** and output tags.



3.6.4 Importing or exporting data tags

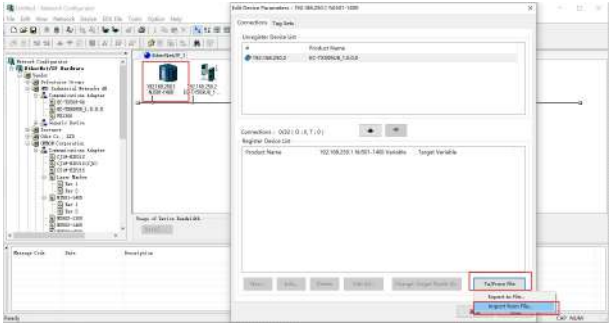
3.6.4.1 Exporting data tags from Sysmac Studio

After data tags are set, click **Export** to export the data tag to the local storage and save it as **EthernetIP.csv**.

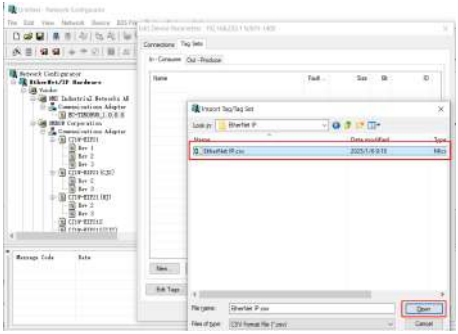


3.6.4.2 Importing data tags to the Network Configurator

In the Network Configurator software, double click the PLC device icon, click **To/From File** in the lower right corner, and choose **Import from File...**.

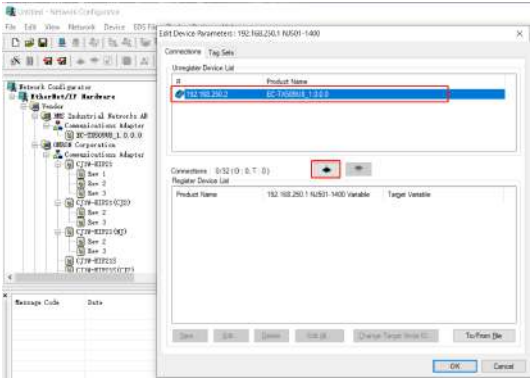


Select the **EthernetIP.csv** file exported from Sysmac Studio, and click **Open**.

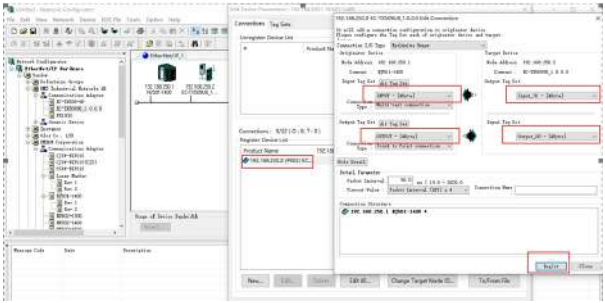


3.6.4.3 Connection corresponding data tags

Choose device **192.168.250.2** under the **Connections** tab and click the down button.



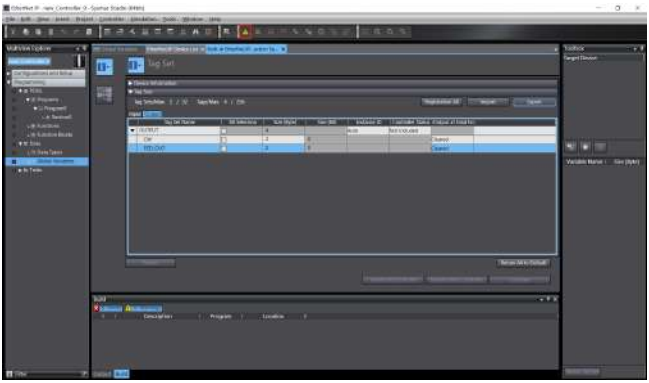
Double click the device **192.168.250.2**, set the data input and output tags, and click **Regist**.



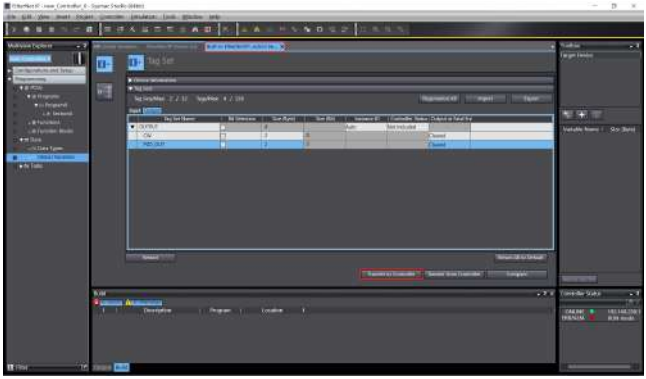
3.6.5 PLC program downloading and online monitoring

3.6.5.1 Downloading from Sysamc Studio

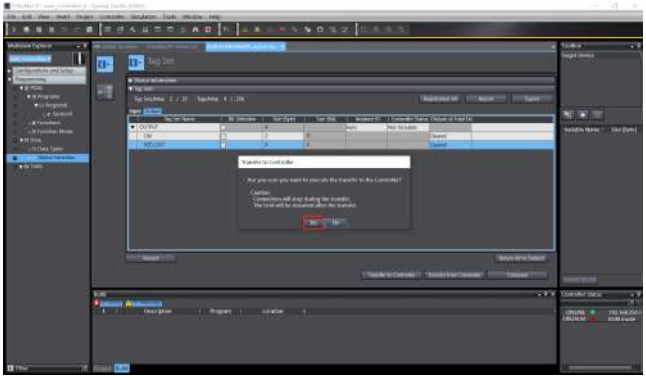
Click the online button.



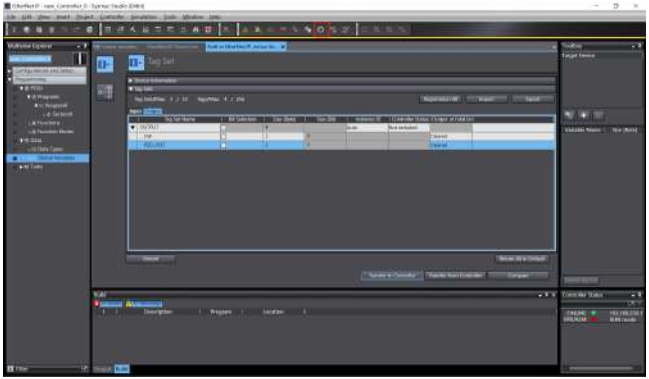
On the **Built-in EtherNet/IP Port Settings** tab, click **Transfer to Controller**.



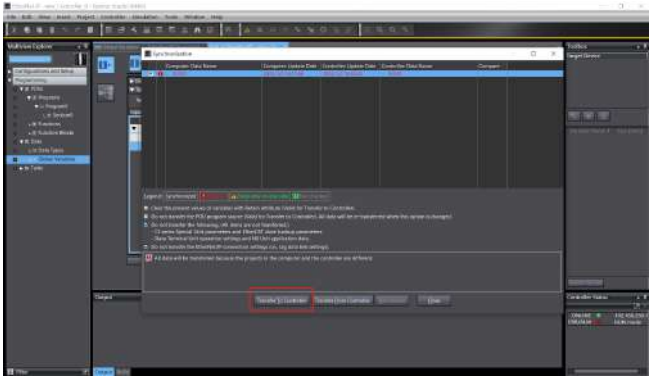
Click **Yes**.



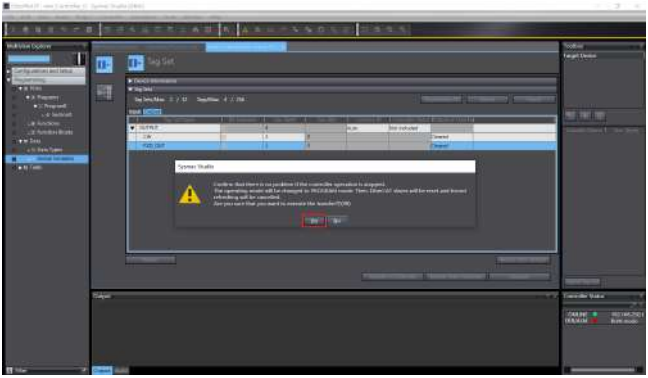
Click the synchronization button.



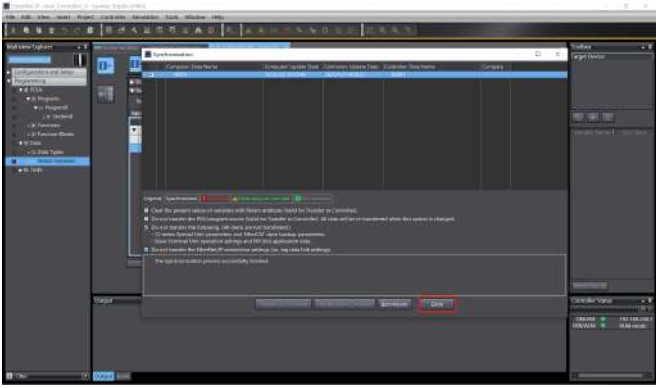
Select device **NJ501-1400**, and click **Transfer to Controller**.



Click **Yes**.

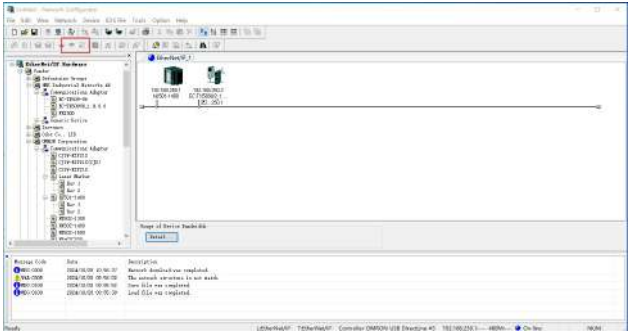


At this time, there are two green lights under **Controller Status** in the bottom right corner. Click **Close**.

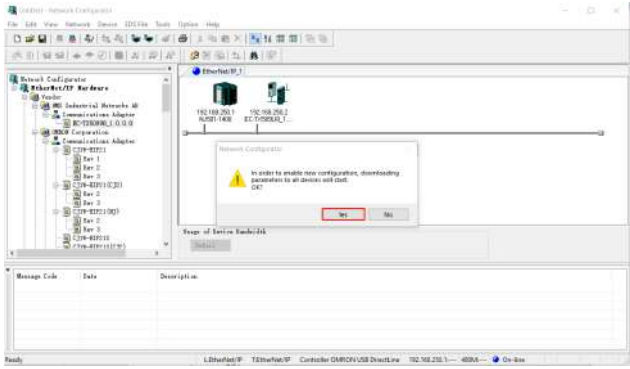


3.6.5.2 Downloading from Network Configurator

Click on the **Download to Device** icon.

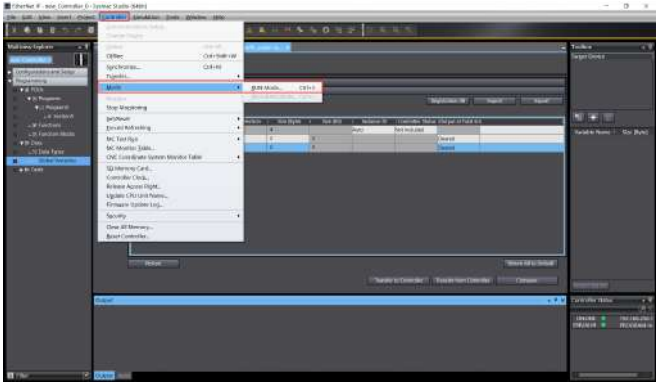


Click the **Download to Network** icon. Click **Yes**.

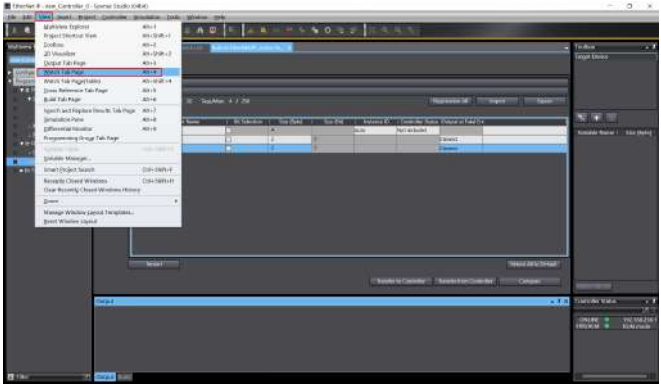


3.6.5.3 Sysamc Studio online monitoring

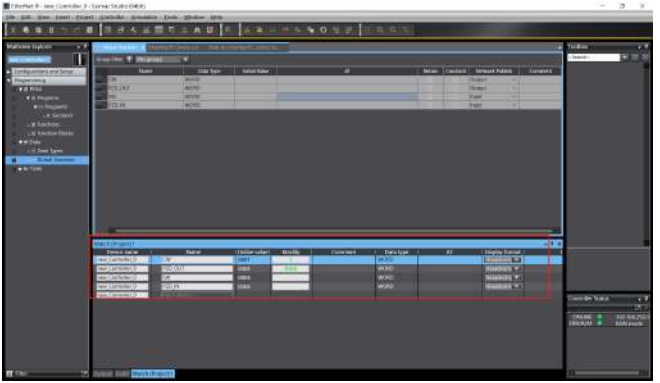
Choose **Controller** > **Mode** > **RUN mode** (or click the Run button) to switch the PLC to the Run mode, and then click **Yes**.



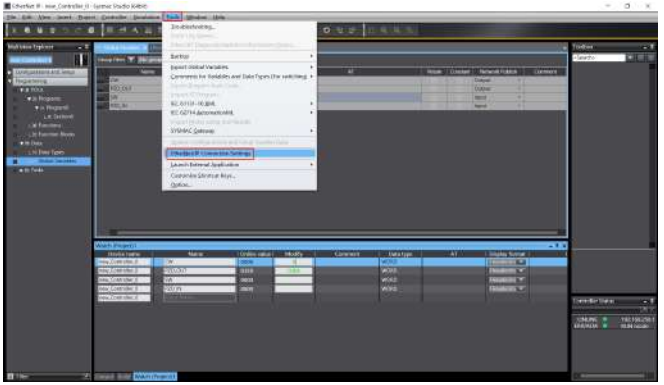
Choose **View** > **Watch Tab Page**.



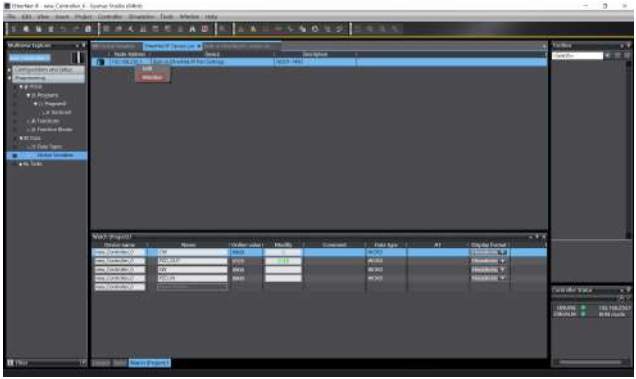
Enter the variable name in the **Watch** window to monitor the variable value; you can change the variable value in real time in the **Modify** column.



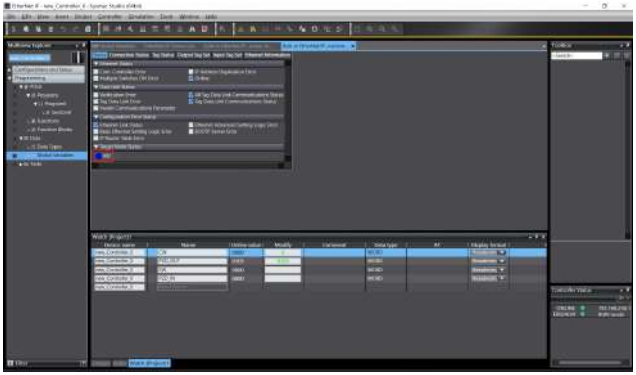
In the top menu bar, choose **Tools** > **Ethernet/IP Connection Settings**.



Right click the PLC and choose **Monitor**.



View **Target Node Status** under **Status**. The status is displayed in blue if communication is successful, and in red if communication fails.



4 EtherCAT

4.1 Overview

The communication card using this protocol is defined as an EtherCAT slave station, which can be used on VFDs that support EtherCAT communication.

4.2 Product features

4.2.1 Supported functions

- Supports the EtherCAT CiA 402 protocol. Configured with a slave station XML configuration file, it can communicate with Beckhoff PLC, INVT AX controllers, and other master stations.
- Automatic network address configuration
- Equipped with two RJ45 ports, supporting 10/100M half/duplex operating, designated for IN and OUT directions.
- Applicable to linear, star, and ring network topologies.

4.2.2 Supported services

- PDO service
- SDO service
- Use of SDO to read and write VFD function codes
- Manufacturer-defined object dictionary
- Two types of objects for VFD reading and writing, that is, PDOs and SDOs (requiring the object dictionary defined by the manufacturer)

Supported EtherCAT synchronization cycles

Table 4-1 Supported EtherCAT synchronization cycles

Item	Supported specification
Synchronization cycle	1ms
	2ms

4.2.3 Status indicator

The EtherCAT communication card provides six indicators to indicate its states. Figure 4-1 shows the positions of the four indicators. Table 4-2 provides the indicator definitions.

Figure 4-1Status indicator positions

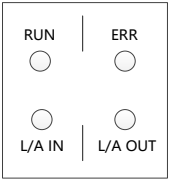


Table 4-2 Status indicator definitions

Indicator	Color	Definition	Function
LED1 (RUN)	Green	Steady on	In OP state.
		Blinking (on for 200ms, off for 200ms)	In PreOP state.
		Single flash (on for 200ms, off for 1s)	In SafeOP state.
		Steady off	In Init state.
LED2 (L/A IN)	Green	Steady on	IN Link established, without data transmission.
		Blinking (on for 50ms, off for 50ms)	IN Link established, with data transmission.
		Steady off	IN LINK not established.
LED3 (L/A OUT)	Green	Steady on	OUT Link established, without data transmission.
		Blinking (on for 50ms, off for 50ms)	OUT Link established, with data transmission.
		Steady off	OUT LINK not established.
LED4 (POWER)	Red	Steady on	3.3V power indicator
LED5 (ERR)	Red	Steady on	The OP fault occurred.
		Blinking (on for 200ms, off for 200ms)	The Init/Preop fault occurred.
		Single flash (on for 200ms, off for 1s)	The Safeop fault occurred.
		Steady off	No fault
LED6 (SYS)	Green	Blinking (on for 500ms, off for 500ms)	Communication card heartbeat indicator (communication card is running normally).

4.3 Electrical connection

An EtherCAT network often consists of a master station (such as PLC) and multiple slave stations (such as drives or bus expansion terminals). Each EtherCAT slave station has two standard Ethernet interfaces. Figure 4-2 and Figure 4-3 show the electrical connections.

Figure 4-2 Linear network topology electrical connection

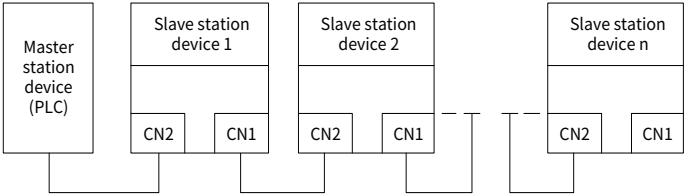
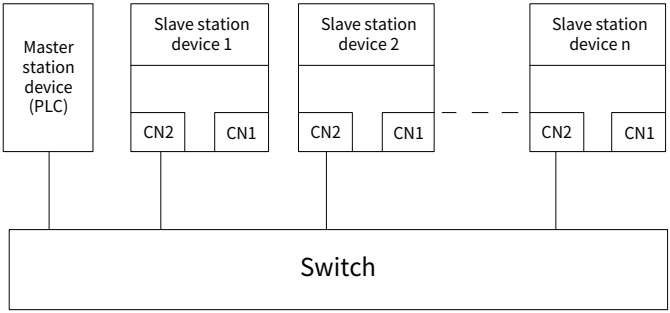
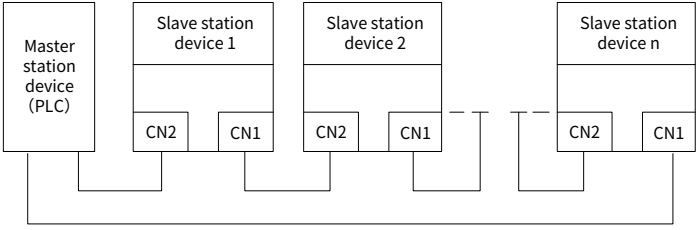


Figure 4-3 Star network topology electrical connection



Note: For the star network topology, you need to prepare EtherCAT switches.

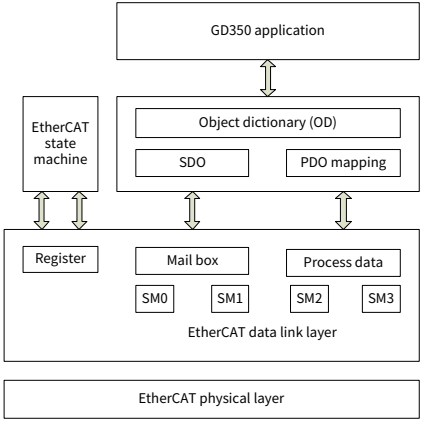
Figure 4-4 Ring network topology electrical connection



4.4 Communication

4.4.1 CANopen over EtherCAT reference model

Figure 4-5 CoE reference model



The EtherCAT (CoE) network reference model consists of two parts: the data link layer and the application layer. The data link layer is mainly responsible for the EtherCAT communication protocol, and the application layer embeds the CANopen Drive Profile (DS402) communication protocol. The object dictionary in CoE includes parameters, application data, and PDO mapping configuration information.

Process data objects (PDO) are composed of mappable objects from the object

dictionary, and the content of PDO data is defined by PDO mappings. The reading and writing of PDO data is periodic and does not require searching for the object dictionary, while email communication (SDO) is non-periodic communication and requires searching for the object dictionary when reading and writing.

Note: In order to correctly parse SDO and PDO data on the EtherCAT data link layer, it is necessary to configure the FMMU and Sync Manager. See the following table.

Table 4-3 EtherCAT Sync Manager configuration

Sync Manager	Configuration	Size	Start address
Sync Manager 0	Assigned to received SDOs.	512 bytes	0x1000
Sync Manager 1	Assigned to send SDOs.	512 bytes	0x1400
Sync Manager 2	Assigned to received PDOs.	128 bytes	0x1800
Sync Manager 3	Assigned to sent PDOs.	128 bytes	0x1C00

4.4.2 EtherCAT slave node information

The EtherCAT slave station information file (XML file) is used for the master station to read, in order to construct the configuration between the master and slave stations. The XML file contains the information necessary for EtherCAT communication settings.

4.4.3 EtherCAT state machine

The EtherCAT state machine is used to describe the states of the slave application and state transitions. State change requests are typically initiated by the master station, with responses from the slave station. The specific state transition method is as shown in the following figure.

Figure 4-6 EtherCAT state machine flowchart

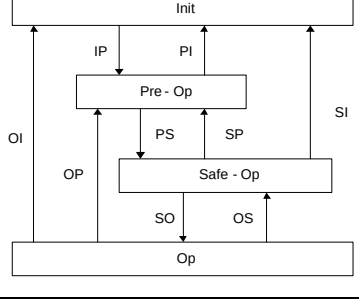


Table 4-4 EtherCAT state machine description

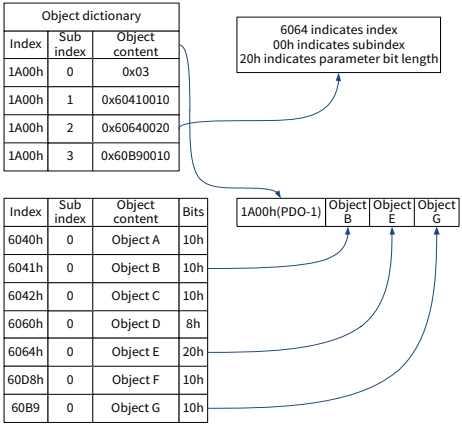
Status	Description
Init	SDO communication is unavailable; PDO communication is unavailable.
Init to Pre-Op	The master station configures the data link layer address and synchronization management (SM) channel for SDO communication. The master station initializes the distributed clock (DC) synchronization information. The master station requests to transition to the Pre-Op state. The master station configures the application layer control register. The slave station checks whether the mailbox has been properly initialized.
Pre-Op	SDO communication is available; PDO communication is unavailable.
Pre-Op to Safe-Op	The master station configures the synchronization management (SM) channel and FMMU channel for PDO communication. The master station configures PDO mappings through SDO communication. The master station requests to transition to the Safe-Op state. The slave station checks whether the PDOs and distributed clocks (DCs) are correctly configured.
Safe-Op	SDO communication is available; PDO receiving communication is available; PDO sending communication is unavailable, in Safe state.
Safe-Op to Op	The master station requests to switch to the Op state.
Op	SDO communication is available; PDO communication is available.

4.4.4 PDO process data mapping

The process data of an EtherCAT slave station is composed of synchronization manager channel objects. Each synchronization manager channel object describes the consistency area of the EtherCAT process data and contains multiple process data objects. EtherCAT slave stations with application control functions should support PDO mapping and the reading of SM PDOs Assign objects.

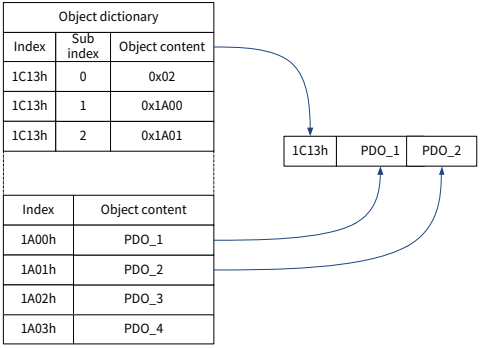
The master station can select the required objects for PDO mapping from the object dictionary. PDO mapping configuration is located in the Object Dictionary from the 1600h to 1603h area (RxPDOs: Receive PDOs) and from 1A00h to 1A03h area (TxPDOs: Transmit PDOs). Figure 4-7 shows the PDO mapping method.

Figure 4-7 PDO mapping method



In addition to mapping the PDO objects, the exchange of EtherCAT process data also requires assigning the PDOs to the Sync Manager channels. The relationship between PDOs and Sync Managers is established through synchronous management of PDO allocation objects (1C12h and 1C13h). Figure 4-8 shows the method for mapping between PDOs and Sync Manager.

Figure 4-8 Mapping between PDOs and Sync Managers



Default PDO mapping (Position, Velocity, Torque, Torque limit, Touch probe):

RxPDO (0x1600)	Control word (0x6040)	Target Position (0x607A)	Target Velocity (0x60FF)	Target Torque (0x6071)	Max. Torque (0x6072)	Mode of Operation (0x6060)	Profile velocity (0x6081)	Touch Probe Function (0x60B8)
TxPDO (0x1A00)	Statusword (0x6041)	Position Actual Value (0x6064)	Speed Actual Value (0x606C)	Torque Actual Value (0x6077)	Following Error Actual Value (0x60F4)	Mode of Operation Display (0x6061)	Error Code (0x603F)	Touch Probe Value (0x60BA)

4.4.5 Network synchronization based on distributed clocks

The Distributed Clock (DC) function ensures that all EtherCAT devices share a unified system time, thus enabling synchronized execution of device tasks. In the EtherCAT network, the first slave station with DC function, has its clock selected as the reference clock of the network, and the rest of the slave and master stations synchronize with this reference clock.

Free-Run mode: The servo drive's run cycle is independent of the master station's communication cycle, allowing for independent run.

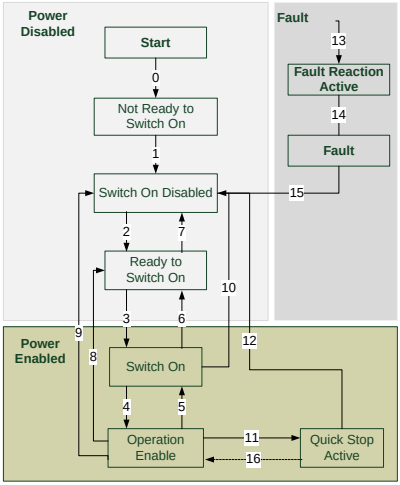
DC mode: The servo drive achieves synchronization operations through the Sync0 events sent by the master station.

4.5 CiA402 device profile

The main station controls the drive through control word (0x6040), and reads status word (0x6041) to obtain the actual status of the drive. The servo drive achieves motor control internally based on the control instructions from the master station.

4.5.1 CANopen over EtherCAT state machine

Figure 4-9 CANopen over EtherCAT state machine



State	Description
Not Ready to Switch On	The drive is in the initialization process.
Switch On Disabled	Drive initialization is completed.
Ready to Switch On	The drive is waiting to enter the Switch On state, and the motor is not excited.
Switched On	The drive is ready, and the main circuit power is normal.
Operation Enable	The drive is enabled and it controls the motor in accordance with the control mode.
Quick Stop Active	The drive stops according to the set mode.
Fault Reaction Active	The drive detects that an alarm has occurred, and stops according to the set mode, while the motor still has

State	Description
	excitation signal.
Fault	The drive is in a fault state and the motor has no excitation signal.

The control word 6040h includes the following content:

- Bits for state control
- Bits related to control mode
- Bits defined by manufacturer

The detailed description of each bit of 6040h is as follows.

15 11	10 9	8	7	6	4 3	2	1	0
Factory defined	Reserved	Reserved	Fault reset	Operation mode	Servo running	Quick stop	Switch on main circuit	Servo being ready
O	O	O	M	O	M	M	M	M
MSB	LSB							

Bits 0–3 and bit 7 (bits for state control):

Command	Bit of the control word					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	0	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	0–1	X	X	X	X	15

Bits 4, 5, 6, and 8 (bits related to control mode):

Bit	Operation mode		
	Profile position mode	Profile velocity mode	Homing mode
4	New set-point	Reserved	Homing operation start

Bit	Operation mode		
	Profile position mode	Profile velocity mode	Homing mode
5	Change set immediately	Reserved	Reserved
6	Rel	Reserved	Reserved
8	Halt	Halt	Halt

Note: In the Profile position mode, when bit4=New set-point, a new position can be triggered.

When the control word is set to 0x0F, the drive is enabled; otherwise, the driver stops. If a fault occurs, a reset command is issued when bit 7 in the control word is set to 1.

6041h status word includes the following content:

- Drive's present status bit
- Status bits related to control mode
- Status bits defined by manufacturer

The detailed description of each bit of 6041h is as follows.

Bit	Description	M/O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	fault	M
4	Voltage enable	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacture specific	O
9	Remote	M
10	Target reached	M
11	Internal limit active	M
12–13	-	-
14–15	Manufacturer specific	O

Bits0–3, 5, and 6:

Value(binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on

Value(binary)	State
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Bit4: Voltage enable, when this bit is set to 1, it indicates that the main circuit power supply is normal.

Bit9: Remote, when this bit is 1, it indicates that the slave station is in OP state, and the master station can control the drive through PDO.

Bit 10: Target reached.

Mode 1 and mode 8: After the present positioning is completed, set it to 1 for keeping. If the positioning is restarted, clear it to 0.

Mode 6: Set it to 1 after homing is completed; set it to 0 if homing is not completed.

Mode 2: Set it to 1 when the ramp reference frequency reaches the set frequency and the control word's bit 4, bit 5, and bit 6 are all set to 1; otherwise, set it to 0.

Other modes: 0

Bit 11: External limitations

Mode 1 and mode 8: Set it to 1 when FWD/REV limit is reached; otherwise set it to 0.

Mode 4 and mode 10: In electric mode: If the electric torque reaches the upper limit (upper limit is not 0), set it to 1; otherwise, set it to 0. In braking mode, if the braking torque reaches the upper limit (upper limit is not 0), set it to 1; otherwise set it to 0.

Mode 2, mode 3, and mode 9: Set it to 1 when the output frequency reaches P00.03; otherwise, set it to 0.

Other modes: 0

Bit 12: Manufacturer defined 1

Mode 1: Set to 1 when bit 4 of the control word is 1; otherwise, set to 0.

Mode 8, mode 9, and mode 10: Set it to 1 when the VFD is in running state; otherwise, set it to 0.

Other modes: 0

Bit 14: When this bit is 1, it indicates the motor is in zero-speed state.

Bit7–bit8, bit15: reserved.

4.5.2 Device run mode

Set the VFD parameters P00.01=2 (Running command channel), P00.02=3 (EtherCAT communication channel), and P16.75 (Communication timeout time). The position mode and homing mode require the drive to operate in closed-loop mode.

4.5.2.1 Profile Position Mode

1. Set 【6060h: Mode of operations】 to 1 (Profile Position Mode).
2. Set the appropriate number of encoder pulses P20.01 according to the encoder model, set the position command source P21.00 tens place to 1; for digital position, set P21.16 digital positioning mode (16#0200, bit 9=1 Position source EtherCAT given).
3. Set 【6081h: Profile velocity】, with the frequency-to-velocity relationship V=60f/p, where V is the velocity in RPM (revolutions per minute), f is the frequency, and p is the number of motor pole pairs.
4. Set【6083h: Profile acceleration】and【6084h: Profile deceleration】and write the corresponding values to P21.20 and P21.21.

   Note: The units of 6083h and 6084h are ms.
5. Set 【6040h: Control word】 to enable the drive (set to 0x4F to enable, bit 6 set to 1: only relative position supported).
6. Set【607Ah: Target position】to the target position (unit: user unit).
7. Query 【6064h: Position actual value】to obtain the motor actual position feedback.
8. Query 【6041h: Status word】 to obtain the drive status feedback (following error, target reached, and internal limit active).
9. For function details, see function parameters in group P21 in function code list chapter and commissioning chapter in the GD350 user manual.
10. During the positioning process, if you need to change the Profile Velocity (6081h), please make the change when the system is stopped.

4.5.2.2 VFD Mode

1. Set 【6060h: Mode of operations】 to 2 (VFD mode).
2. Set 【6046h: vl velocity min max amount】 to set the max. and min. rotation speeds. If you do not set it, the default values on the drive are used.

3. Set the object dictionary 【6048h: vl velocity acceleration】, 【6049: vl velocity deceleration】, and 【604Ah:02 QuickStop Speed】. Do not set them to the default values of the drive.
   - The acceleration time is 60*【6048h:02 Acceleration Delta Time】*P00.04/(【6048h:01 Acceleration Delta Speed】* Number of motor pole pairs) * 0.1, with second as the unit, corresponding to P00.11. The value will remain the same as the previous one when it exceeds 3600.0 seconds.
   - The deceleration time is 60*【6049h:02 Deceleration Delta Time】*P00.04/(【6049h:01 Deceleration Delta Speed】* Number of motor pole pairs)*0.1, with second as the unit, corresponding to P00.12. The value will remain the same as the previous one when it exceeds 3600.0 seconds.
   - The emergency stop time is 60 * 【604Ah:02 QuickStop Delta Speed】 * P00.04/(【604Ah:01 QuickStop Delta Time】 * Number of motor pole pairs)*0.1, with second as the unit, corresponding to P00.12. The value will remain the same as the previous one when it exceeds 60.0 seconds.
4. Set【604Ch: vl dimension factor】to adjust the electronic gear ratio, which is 1:1 by default.
5. Set【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
6. Set 【6042h: vl target velocity】to set the target speed.
7. Set 【6040h: Control word】to run the drive (when it is set to 0x7F, the drive runs).
8. Query 【6044h: vl velocity actual value】to obtain the motor actual position feedback.

4.5.2.3 Profile Velocity Mode

1. Set 【6060h: Mode of operations】 to 3 (Profile Velocity Mode).
2. Set【6083h: Profile acceleration】and【6084h: Profile deceleration】and write the corresponding values to P00.11 and P00.12.

   Note: The units of 6083h and 6084h are ms.
3. Set P00.01=2, P00.02=3, and P00.06=13.
4. Set【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
5. Set 【60FFh: Target velocity】 to set the target rotational speed (unit: rpm).
6. Query 【6041h: Status word】 to obtain the drive status feedback (Speed zero,

Max slippage error, Target reached, Internal limit active).

4.5.2.4 Profile Torque Mode

1. Set 【6060h: Mode of operations】 to 4 (Profile Torque Mode).
2. Set 【6087h】 to set the ramp torque.
3. Set【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
4. Set P03.11=11 and P03.32=1 (torque control enabled).
5. Set 【6071h: Target torque】 to set the target torque.
6. Query 【6041h: Status word】 to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

4.5.2.5 Homing Mode

1. Set 【6060h: Mode of operations】 to 6 (homing mode).
2. Set P22.00.Bit0=1 to enable spindle positioning function, and set P22.03–P22.06 zero positions.
3. Set the 【6040h: Control word】 to enable the drive (set to 0x0F to enable). When bit 4 changes from 0 to 1, the homing operation starts, and when bit4 changes from 1 to 0, the homing operation is interrupted.
4. The motor queries for the limit switch and home switch to complete the homing action.
5. Query 【6041h: Status word】to obtain the drive status feedback (Homing error, Homing attained, Target reached).
6. For function details, see function parameters in group P22 in function code list chapter and commissioning chapter in the GD350 user manual.

4.5.2.6 Touch Probe Function

Record the present position according to the Z phase rising edge or falling edge signal, and set the control word to record one or multiple times.

1. When the control word (60B8h: Touch Probe Control) is 23 (2#01 0111), the function indicates that the Z phase rising edge triggers recording the present position continuously. The recorded value is Turns of motor rotation x 10000. Every time the motor rotates a turn, a value is updated. When the status word (60B9h: Touch Probe Statu) is 67/195, in continuous triggering mode, the probe status word bit 6 is 1, and each triggering will cause bit 7 to toggle.

2. When the control word (60B8h: Touch Probe Control) is 21 (2#01 0101), the function indicates that the Z phase rising edge triggers recording the present position. The recorded value is Turns of motor rotation x 10000. The status word is 3 (60B9h: Touch Probe Statu).
3. When the control word (60B8h: Touch Probe Control) is 39 (2#11 0111), the function indicates that the Z phase falling edge triggers recording the present position continuously. The recorded value is Turns of motor rotation x 10000. Every time the motor rotates a turn, a value is updated. When the status word is 5 (60B9h: Touch Probe Statu), in continuous triggering mode, the probe status word bit 6 is 1, and each triggering will cause bit 7 to toggle.
4. When the control word (60B8h: Touch Probe Control) is 37 (2#01 0101), the function indicates that the Z phase rising edge triggers recording the present position. The recorded value is Turns of motor rotation x 10000, which is recorded only once. When th status word (60B9h: Touch Probe Statu) is 5, the falling edge triggering control word is 37.

Note: The difference from CiA 402 is that this mode only supports one probe, and only one locked value can be recorded at the same time.

4.5.2.7 Cyclic Synchronous Position Mode

1. Set 【6060h: Mode of operations】 to 8 (Cyclic synchronous position mode).
2. Set P00.00=3, P00.01=2, P00.02=3, and P00.06=13.
3. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable).
4. Set【607Ah: Target position】to the target position (unit: user unit).
5. Query 【6064h: Position actual value】to obtain the motor actual position feedback.
6. Query 【6041h: Status word】 to obtain the drive status feedback (following error, target reached, and internal limit active).
7. For function details, see function parameters in group P21 in function code list chapter and commissioning chapter in the GD350 user manual.

4.5.2.8 Cyclic Synchronous Velocity Mode

1. Set 【6060h: Mode of operations】 to 9 (Cyclic synchronous velocity mode).
2. Set 【6083h: Profile acceleration】 and 【6084h: Profile deceleration】.
3. Set P00.01=2, P00.02=3, and P00.06=13.
4. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start

the motor operation.

5. Set 【60FFh: Target velocity】 to set the target rotational speed (unit: rpm).
6. Query 【6041h: Status word】 to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

4.5.2.9 Cyclic Synchronous Torque Mode

1. Set 【6060h: Mode of operations】 to 10 (Cyclic Synchronous torque Mode).
2. Set P03.11=11 (Communication) and P03.32=1 (Torque control enabled).
3. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
4. Set 【6072h: Max torque】 to the maximum torque and 【6071h: Target torque】 to the target torque.
5. Query 【6041h: Status word】 to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

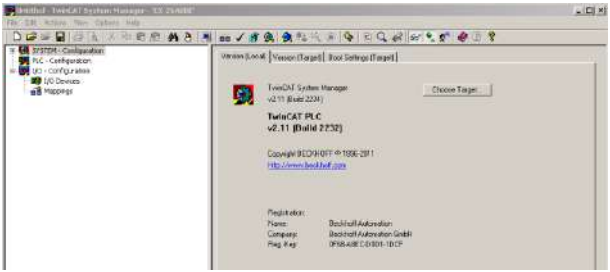
4.6 PLC communication example 1 (TwinCAT2)

This example illustrates how to use the VFD EtherCAT module to communicate with Beckhoff TwinCAT2 that serves as the master station.

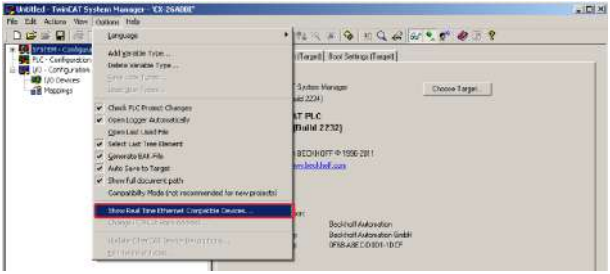
Step 1 Install the TwinCAT2 software.

Step 2 Copy GD350 EtherCAT configuration file (**EC-TX509U8_1.0.0.0.xml**) to the TwinCAT2 installation directory **C:\TwinCAT\Io\EtherCAT**.

Step 3 Open TwinCAT2.

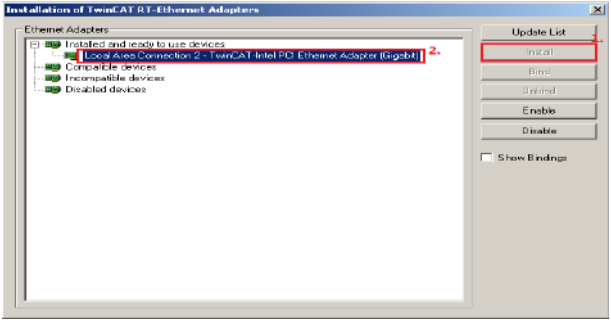


Step 4 Install the network card driver.

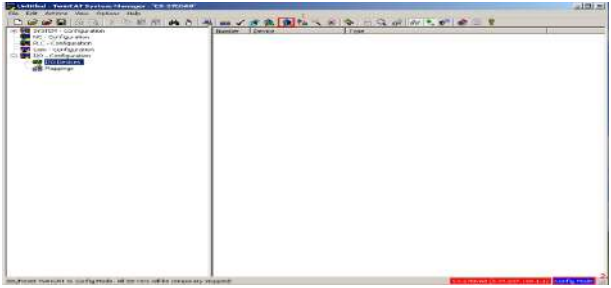


In the top menu bar, choose **Options** > **Show Realtime Ethernet Compatible Devices...**. In the pop-up dialog box, choose the local network card, click **Install**. Once the network card is installed, it is displayed under **Installed and ready to use devices**.

Note: Please use a network card with an Intel chipset.

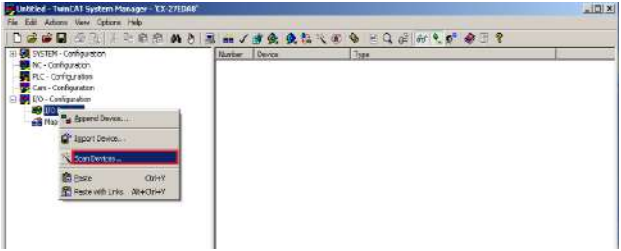


Step 5 Set TwinCAT2 to configuration mode.

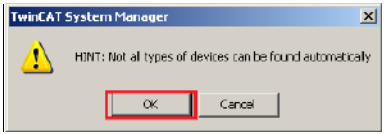


Step 6 Scan for devices.

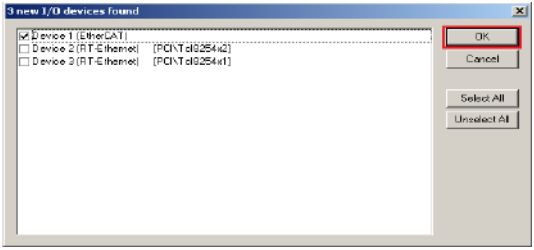
A. Right click **I/O Devices** and choose **Scan Devices...** to scan for devices.



B. Click **OK** in the pop-up dialog box.



C. Click **OK** in the pop-up dialog box.



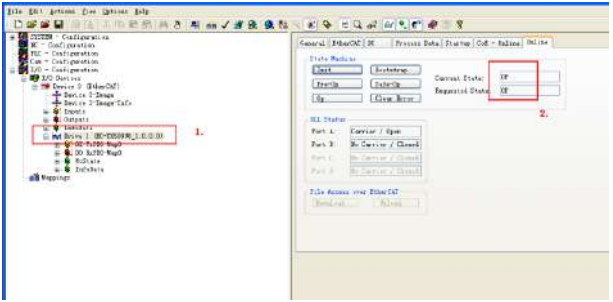
D. Click **Yes** in the pop-up dialog box.



E. Click **Yes** in the pop-up dialog box. The device will enter free run mode.

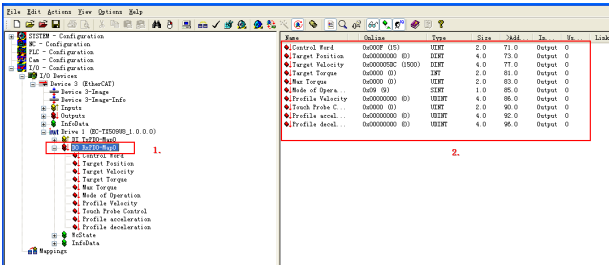


F. As shown in the following figure, **Drive 1 (EC-TX509U8_1.0.0.0)** is the scanned slave station device. Check whether the device enters the OP state.

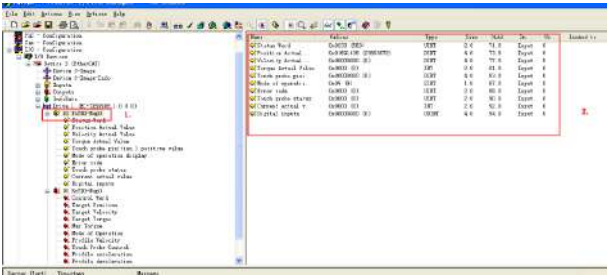


Step 7 Perform process data input and output.

A. Choose **DO RxPDO-Map0**. The data is sent from the master station to the VFD. Command giving and speed giving, and other operations can be performed.

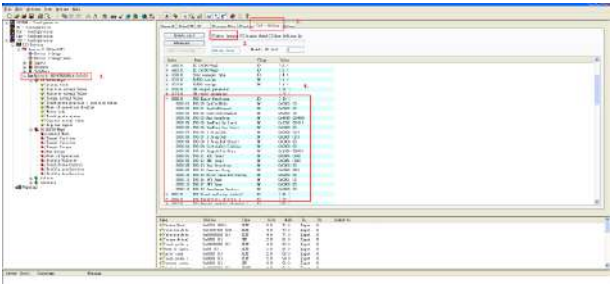


B. Choose **DI TxPDO-Map0**. The data is sent from the VFD to the master station, including the VFD status and speed.

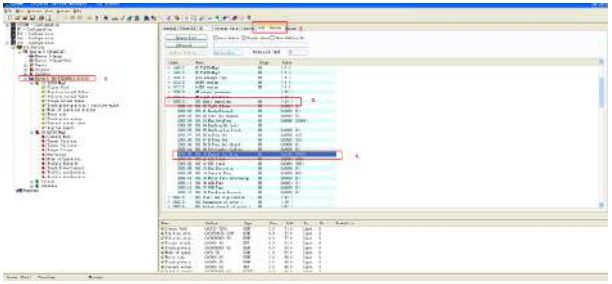


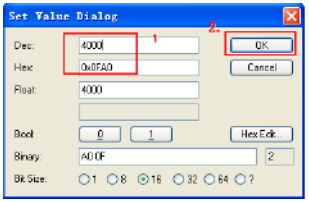
Step 8 Perform SDO data read and write operations.

A. Click **CoE-Online**. Select 0x2000–0x2063, and select **Auto Update**. Then you can view the parameters of the corresponding function codes.



B. Similarly, you can write function parameters through 0x2000–0x2063. Select **0x2000**, double click **2000:0B**, a read dialog box will pop up, write the parameters, click **OK**, check the keypad, and you can see the parameter has been successfully written.

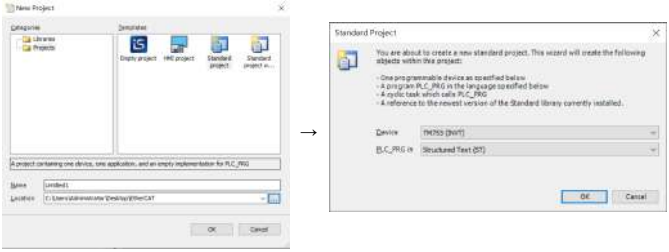




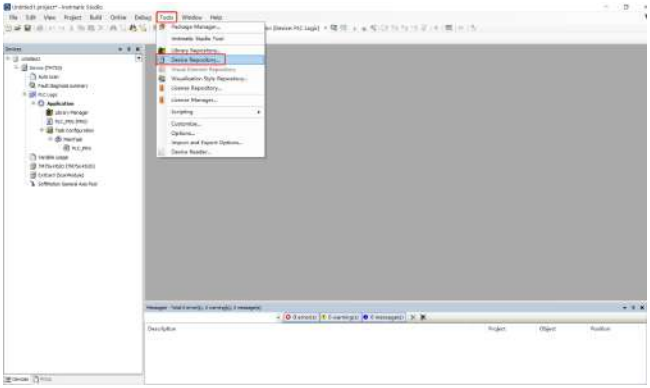
4.7 PLC communication example 2 (TM753)

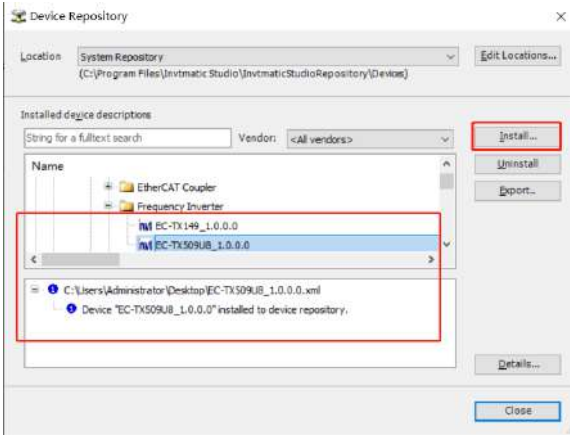
This example illustrates how to use the VFD EtherCAT module to communicate with TM753 that serves as the master station.

Step 1 Open the Invtmatic Studio software and create a new project TM753. Select device **TM753**. The interface is shown in the following figure.

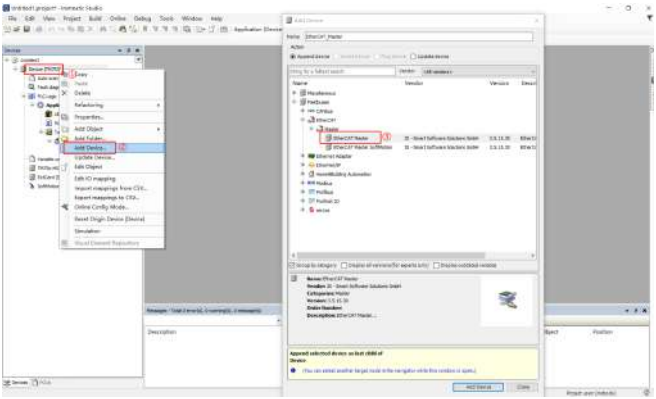


Step 2 Add an EtherCAT device. In the top menu bar, choose **Tools** > **Device Repository**, and import the EtherCAT configuration file of EC-TX509U8.



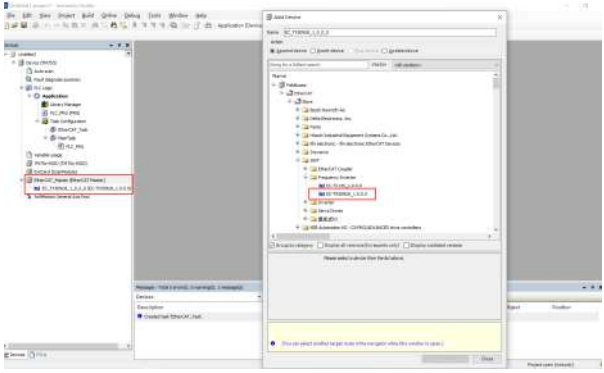


Step 3 Right click **Device**, and choose **Add Device**. In the pop-up window, choose **Fieldbuses** > **EtherCAT** > **Master**, and double click **EtherCAT Master**.

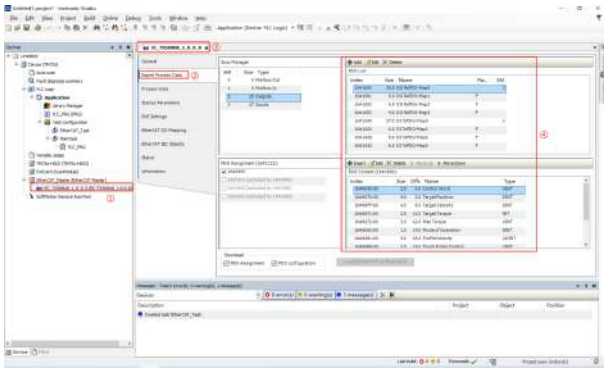


Step 4 Right click **EtherCAT Master**, and choose **Add Device**. In the pop-up

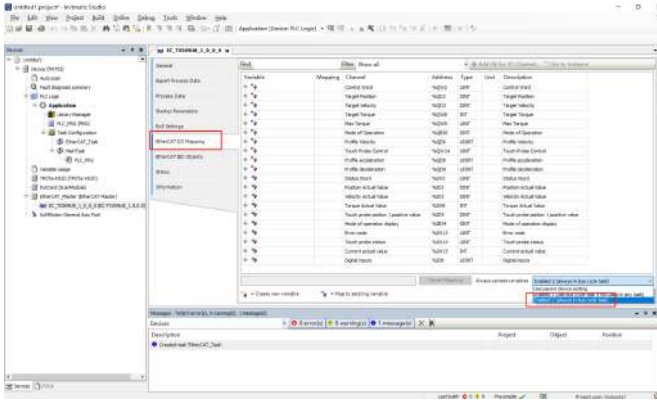
window, choose **Fieldbuses** > **EtherCAT** > **Slave** > **INVT** > **Frequency Inventory** > **EC-TX509U8_1.0.0.0**.



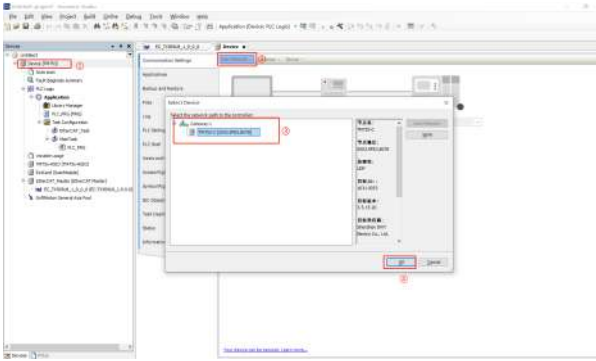
Step 5 Perform PDO configuration. Double click **EC-TX509U8_1.0.0.0** that was added to enter the parameter configuration interface. Enter the **Expert Process Data** interface to configure the PDO parameters. PDO parameter configuration can be done according to requirements. This example uses the default PDO configuration.



Step 6 Choose **EtherCAT I/O mapping**, and change **Always update variables** to **Enabled 2 (always in bus cycle task)**.

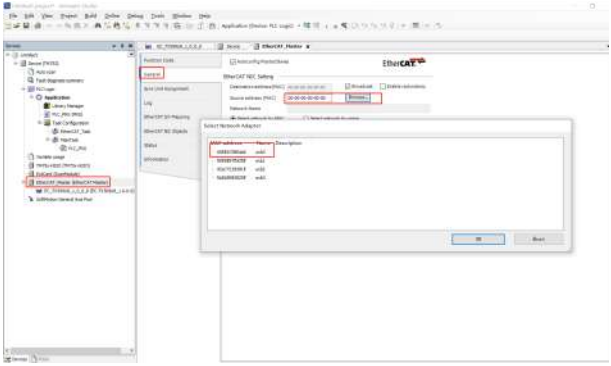


Step 7 Scan for the PLC. Double click **Device**, click **Scan Network...**, select the scanned PLC, and click **OK**.

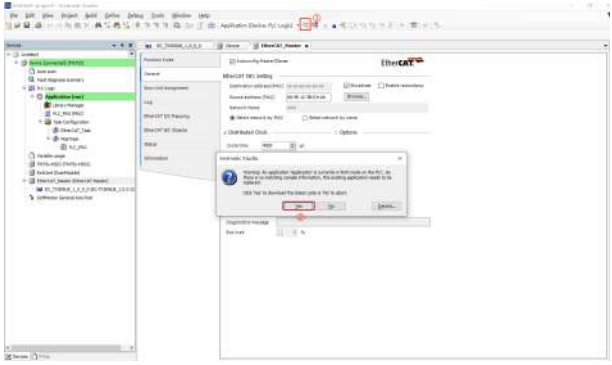


Step 8 Perform EtherCAT master station configuration. Double click **EtherCAT_Master**, choose **General**, click **Browse**, and double click to

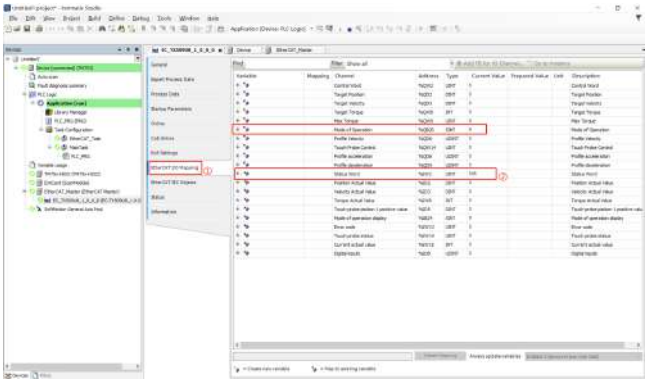
select **eth0**.



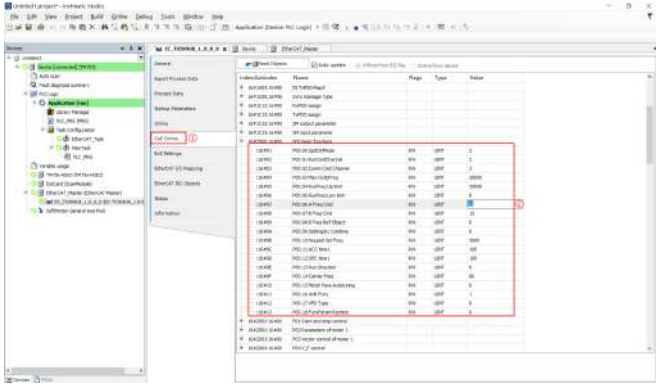
Step 9 Download the program.



Step 10 Perform parameter monitoring. TPDO data can be monitored in real time and RPDO data can be written by choosing **EtherCAT I/O mapping**.



Step 11 In the **COE online** interface, you can monitor the values of the function codes, and you can also directly modify the values of the function codes (Note: P00.00 modification is invalid).



5 Modbus TCP protocol

5.1 Overview

The communication card using this protocol is defined as a Modbus TCP slave station, which can be used on VFDs that support Modbus TCP communication.

5.2 Product features

5.2.1 Supported functions

- Supports the Modbus TCP protocol and Modbus TCP slave stations
- Supports the concurrent communication with multiple master stations. It can communicate with Schneider PLCs, INVT AX controllers, and other master stations.
- Equipped with two RJ45 ports, supporting 10/100M half/full duplex operating.
- Supports basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes.
- Applicable to linear and star network topologies.

5.2.2 Supported communication types

Modbus TCP application layer uses the same Modbus protocol as Modbus RTU, based on the Transmission Control Protocol/Internet Protocol (TCP/IP) as the transmission protocol for Ethernet online control and information. It allows for the transmission of explicit messages between points without strict timing requirements.

Similar to Modbus RTU, Modbus TCP requires the PLC/host controller to send read or write commands, with the communication card forwarding the data and returning the operation result in order to complete a data transmission.

5.2.3 Status indicator

Modbus TCP communication card provides six indicators to indicate its states. For details, see Table 5-1.

Table 5-1 Indicator description

Indicator	Color	Definition	Function
LED1 (RUN)	Green	Steady on	The communication between the communication card and the PLC is online, and data exchange is

Indicator	Color	Definition	Function
			allowed.
		Blinking (on for 500ms, off for 500ms)	Abnormal setting of the IP address for either the communication card or the PLC.
		Steady off	The communication between the communication card and PLC is not in Online state.
LED2 (HOST)	Green	Steady on	The communication card is in the process of handshaking with the VFD.
		Blinking (on for 500ms, off for 500ms)	The communication card and VFD communicate normally. **Note:** After the handshaking is completed, it should blink regardless of whether there is data transmission between the communication card and the main control board.
		Steady off	The communication card is in the initialization or parameter configuration phase.
LED3 (DATA)	Green	Blinking (on for 500ms, off for 500ms)	The data update between the communication card and main control board is normal.
		Steady off	No data update or abnormal update between the communication card and main control board.
LED4 (POWER)	Red	Steady on	3.3V power indicator
LED5 (ERR)	Red	Steady on	The communication between the communication card and PLC is offline.
		Blinking (on for 500ms, off for 500ms)	An attempt to operate an unsupported CMD control word instruction or PR function code value.
		Blinking (on for 62.5ms, off	An attempt to operate on a

Indicator	Color	Definition	Function
		for 62.5ms)	non-existent node address.
		Steady off	The communication between the communication card and PLC is normal.
LED6 (SYS)	Green	Blinking (on for 500ms, off for 500ms)	Communication card heartbeat indicator (communication card is running normally).

5.3 Electrical connection

The Modbus TCP communication card adopts standard RJ45 interfaces, which can be used in a linear network topology and a star network topology, as shown in Figure 5-1 and Figure 5-2.

Use CAT5, CAT5e, and CAT6 network cables for electrical wiring. When the communication distance is greater than 50m, use high-quality network cables that meet the high-quality standards.

Figure 5-1 Linear network topology electrical connection

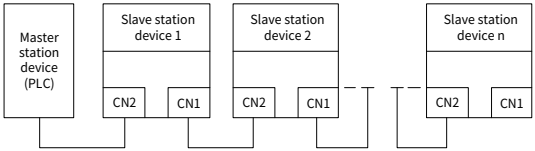
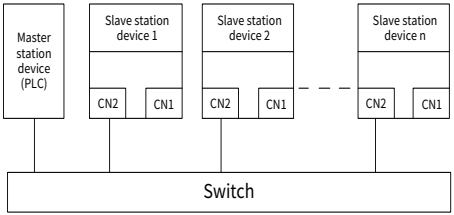


Figure 5-2 Star network topology electrical connection



Note: For the star network topology, you need to prepare switches.

5.4 Communication

5.4.1 Communication settings

The communication card can only be used as a Modbus TCP slave, and function codes should to be set on the VFD before communication. The procedure is as follows:

1. Set the communication card communication station address, IP address and subnet mask.

The factory station address, IP address, and subnet mask of each communication card are 1, 192.168.0.20, and 255.255.255.0 respectively, which can be changed to a network segment address according to the actual requirements.

2. Set the control method.

To enable the VFD control through Modbus TCP communication, set the control mode to Modbus TCP communication control. To be specific, set P00.01=2 and P00.02=0 to control VFD start and stop. In short, if a value needs to be set through Modbus TCP communication, the corresponding function code should be modified to Modbus TCP communication control. For related function codes, see Appendix C Related function codes.

Note: After steps 1 and 2 are implemented properly, the communication card can communication properly. If a VFD needs to be controlled, related function nodes must be set and the control mode is Modbus TCP communication.

5.4.2 Message format

The TCP communication message is shown in Table 5-2.

Table 5-2 TCP communication message

Header of MAC layer	Header of IP layer	Header of TCP layer	Valid data	Trailer
14 bytes	20 bytes	20 bytes	0–1488 bytes	4 bytes

5.4.3 Modbus TCP communication

The Modbus TCP communication card supports the Modbus protocol at the application layer. Modbus TCP protocol messages are located in the TCP message valid data zone and divided into two parts: the first part is the MBAP (message header, 7 bytes), and the second part is the PDU (protocol data unit, with variable length). See Table 5-3.

Table 5-3 Modbus TCP protocol message

MBAP				PDU	
Transaction processing identifier	Protocol identifier	Length	Unit identifier	Function code	Data
2 bytes	2 bytes	2 bytes	1 byte	1 byte	n bytes
The message sequence number increments by 1 after each communication to differentiate between different messages.	0000=Modbus-TCP protocol	Subsequent data length	Device address (Station No.)	Modbus function code	Includes VFD function codes and data, with variable length.

Through the preceding messages, you can set the reference parameters, monitor status values, send control commands and monitor operation status of the VFD, and read and write VFD function parameters.

Parameter description:

- Unit identifier: Station address (1–247)
- Function code: Modbus function code

Table 5-4 Modbus function code

Function code	Description
0x01	Read coil (not supported)
0x05	Write a single coil (not supported)
0x0F	Write multiple coils (not supported)
0x02	Read discrete input (not supported)
0x04	Read input register (not supported)
0x03	Read holding register
0x06	Write a single holding register
0x10	Write multiple holding registers

Data: The data of the first word is the VFD function code address. For example, P00.00 corresponds to the address 0000h; the subsequent data is the value of read and write.

Message example:

1. Command code 03H, reading N words (continuously up to 16 words)

The command code 03H is used by the master to read data from the VFD. The count of data to be read depends on the "data count" in the command. A maximum of 16 pieces of data can be read. The addresses of the read parameters must be contiguous. Each piece of data occupies 2 bytes, that is, one word. The command format is presented using the hexadecimal system (a number followed by "H" indicates a hexadecimal value). One hexadecimal value occupies one byte.

The 03H command is used to read information including the parameters and running status of the VFD.

For example, if the master reads two contiguous pieces of data (that is, to read content from the data addresses 0004 H and 0005 H) from the VFD whose address is 01H, the command frame structure is described in the following.

Example	Request	0001	0000	0006	01	03	0004	0004
	Meaning	MBAP				Function code	Read address	Data byte
	Response	0001	0000	0007	01	03	04	13880000
	Meaning	MBAP				Function code	Data byte	Data

According to the data, the value at 0004H is 1388H (50.00Hz), and the value at 0005H is 0000H (00.00Hz).

2. Function code 06H: writing a word

This command is used by the master to write data to the VFD. One command can be used to write only one piece of data. It is used to modify the parameters and running mode of the VFD.

For example, if the master writes 5000 (1388H) to 0004H of the VFD whose address is 02H, the frame structure is as follows.

Example	Request	0001	0000	0006	02	06	0004	1388
	Meaning	MBAP				Function code	Write address	Data
	Response	0001	0000	0006	02	06	0004	1388
	Meaning	MBAP				Function code	Write address	Data

3. Function code 10H: continuous writing

The command code 10H is used by the master to write data to the VFD. The quantity of data to be written is determined by data count, and a maximum of 16 pieces of

data can be written.

For example, if the master writes 5000 (1388H) and 50 (0032H) to 0004H and 0005H of the VFD whose address is 02H, the frame structure is as follows.

Example	Request	0001	0000	000B	02	10	0004	0002	04	13880032
	Meaning	MBAP				Function code	Write address	Number of registers	Data byte	Data
	Response	0001	0000	0006	02	10	0004	0002		
	Meaning	MBAP				Function code	Write address	Number of registers		

5.4.4 Data address definition

This section describes the address definition of communication data. The addresses are used for controlling the running, obtaining the state information, and setting related function parameters of the VFD.

The address of a function code consists of two bytes, with the MSB on the left and LSB on the right. The MSB ranges from 00 to ffH, and the LSB also ranges from 00 to ffH. The high-order byte is the hexadecimal form of the group number before the dot mark, and low-order byte is that of the number behind the dot mark. Take P14.00 as an example. The group number is 14, that is, the MSB of the parameter address is the hexadecimal form of 0E; and the number behind the dot mark is 00, that is, the LSB is the hexadecimal form of 00. Therefore, the function code address is 0E00H in the hexadecimal form. For example, the parameter address for function code P14.03 is 0E03H.

Function code	Name	Parameter description	Setting range	Default
P14.00	Local communication address	1–247	1–247	1
P14.03	Communication response delay	0–200ms	0–200	5ms

The parameters in the P99 group are set by the manufacturer and cannot be read or modified. Some parameters cannot be modified when the VFD is running; some cannot be modified regardless of the VFD status. Pay attention to the setting range, unit, and description of a parameter when modifying it.

The service life of the Electrically Erasable Programmable Read-Only Memory (EEPROM) may be reduced if it is frequently used for storage. Some function codes do not need to be stored during communication. The application requirements can be met by modifying the value of the on-chip RAM, that is, modifying the MSB of the corresponding function code address from 0 to 1. For example, if P00.07 is not to be stored in the EEPROM, you need only to modify the value of the RAM, that is, set the address to 8007H. The address can be used only for writing data to the on-chip RAM, and it is invalid when used for reading data.

Addresses of other Modbus functions

In addition to modifying the parameters of the VFD, the master can also control the VFD, such as starting and stopping it, and monitoring the operation status of the VFD. The following table lists other function parameters.

Function description	Address definition	Data description	R/W
Communication-based control command	2000H	0001H: Forward running	R/W
		0002H: Reverse running	
		0003H: Forward jogging	
		0004: Reverse jogging	
		0005H: Stop	
		0006H: Coast to stop	
		0007H: Fault reset	
		0008H: Jogging to stop	
		0009H: Emergency stop	
Communication-based setting address	2001H	Communication-based frequency setting (0–Fmax, unit: 0.01Hz)	R/W
	2002H	PID reference (0–1000, in which 1000 corresponds to 100.0%)	R/W
	2003H	PID feedback (0–1000, in which 1000 corresponds to 100.0%)	R/W
	2004H	Torque setting (-3000–3000, in which 1000 corresponds to 100.0% of the motor rated current)	R/W
	2005H	Upper limit setting of forward running frequency (0–Fmax; unit: 0.01Hz)	R/W
	2006H	Upper limit setting of reverse running frequency (0–Fmax; unit: 0.01Hz)	R/W
	2007H	Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to	R/W

Function description	Address definition	Data description	R/W
		100.0% of the motor rated current)	
	2008H	Braking torque upper limit. (0–3000, in which 1000 corresponds to 100.0% of the motor rated current)	R/W
	2009H	Special control command word Bit1–bit0=00: Motor 1 =01: Motor 2 Bit2: =1: Enable speed/torque control switchover =0: Disable Bit3: =1 Clear electricity consumption data =0: Keep electricity consumption data Bit4: =1 Enable pre-excitation =0: Disable pre-excitation Bit5: =1: Enable DC braking =0: Disable DC braking	R/W
	200AH	Virtual input terminal command. Range: 0x000–0x3FF (corresponding to S8/S7/S6/S5/HDIB/HDIA/S4/S3/S2/S1 in sequence)	R/W
	200BH	Virtual output terminal command (0x00–0x0F) Corresponding to local RO2/RO1/HDO/Y1	R/W
	200CH	Voltage setting (used for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage)	R/W
	200DH	AO setting 1 (-1000–+1000, in which 1000 corresponding to 100.0%)	R/W
	200EH	AO setting 2 (-1000–+1000, in which 1000 corresponding to 100.0%)	R/W
VFD status word 1	2100H	0001H: Forward running	R
		0002H: Running reversely	
		0003H: Stopped	
		0004H: Faulty	
		0005H: In POFF state	
		0006H: In pre-exciting state	

Function description	Address definition	Data description		R/W
VFD status word 2	2101H	Bit0: =0: Not ready to run =1: Ready to run Bit2–bit1=00: Motor 1 =01: Motor 2 Bit3: =0: AM =1: SM Bit4: =0: Disable overload alarm =1: Enable overload alarm Bit6–Bit5: =00: Keypad-based control =01: Terminal-based control =10: Communication-based control Bit 7: Reserved Bit8: =0: Speed control =1: Torque control Bit9: =0: Non position control =1: Position control Bit11–Bit10: =00: Vector 0 =01: Vector 1 =10: Closed-loop vector =11: Space voltage vector		R
VFD fault code	2102H	See the description of fault types.		R
VFD identification code	2103H	GD350-----0x01A0		R
Running frequency	3000H	0–Fmax (Unit: 0.01Hz)	Compatible with CHF100A and CHV100 communication addresses	R
Set frequency	3001H	0–Fmax (Unit: 0.01Hz)		R
Bus voltage	3002H	0.0–2000.0V (Unit: 0.1V)		R
Output voltage	3003H	0–1200V (Unit: 1V)		R
Output current	3004H	0.0–3000.0A (Unit: 0.1A)		R
Rotation speed of running	3005H	0–65535 (Unit: 1 RPM)		R
Output power	3006H	-300.0–300.0% (Unit: 0.1%)		R
Output torque	3007H	-250.0–250.0% (Unit: 0.1%)		R
Closed-loop setting	3008H	-100.0–100.0% (Unit: 0.1%)		R
Closed-loop feedback	3009H	-100.0–100.0% (Unit: 0.1%)		R

Function description	Address definition	Data description	R/W
Input IO state	300AH	0x00–0x3F Corresponding to the local HDIB/ HDIA/S4/S3/S2/S1	R
Output IO state	300BH	0x0–0xF Corresponding to local RO2/RO1/HDO/Y1	R
Analog input 1	300CH	0.00–10.00V (Unit: 0.01V)	R
Analog input 2	300DH	0.00–10.00V (Unit: 0.01V)	R
Analog input 3	300EH	-10.00–10.00V (Unit: 0.01V)	R
Analog input 4	300FH		R
Read input of HDIA high-speed pulse	3010H	0.00–50.00kHz (Unit: 0.01Hz)	R
Read input of HDIB high-speed pulse	3011H		R
Read the actual step of multi-step speed	3012H	0–15	R
External length value	3013H	0–65535	R
External counting value	3014H	0–65535	R
Torque setting	3015H	-300.0–300.0% (Unit: 0.1%)	R
VFD identification code	3016H		R
Fault code	5000H		R

The Read/Write (R/W) characteristics indicate whether a function parameter can be read and written. For example, "Communication-based control command" can be written, and therefore the command code 06H is used to control the VFD. The R characteristic indicates that a function parameter can only be read, and W indicates that a function parameter can only be written.

Note: Some parameters in the preceding table are valid only after they are enabled. Take the running and stop operations as examples, you need to set "Running command channel" (P00.01) to "Communication", and set "Communication mode of running commands" (P00.02) to Modbus. For another

example, when modifying "PID reference", you need to set "PID reference source" (P09.00) to Modbus communication.

The following table describes the encoding rules of device codes (corresponding to the VFD identification code 2103H).

8 MSBs	Meaning	8 LSBs	Meaning
0x01	GD	0x08	GD35 series VFD
		0x09	GD35-H1 series VFD
		0x0a	GD300 series VFD
		0xa0	GD350 series VFD
		0xa1	GD350-UL series VFD
		0xa2	GD350A series VFD

5.4.5 Fieldbus scale

In practical applications, communication data is represented in the hexadecimal form, but hexadecimal values cannot represent decimals. For example, 50.12 Hz cannot be represented in the hexadecimal form. In such cases, multiply 50.12 by 100 to obtain an integer 5012, and then 50.12 can be represented as 1394H in the hexadecimal form (5012 in the decimal form).

In the process of multiplying a non-integer by a multiple to obtain an integer, the multiple is referred to as a fieldbus scale.

The fieldbus scale depends on the number of decimal places in the value specified in "Setting range" or "Default". If there are n (for example, 1) decimal places in the value, the fieldbus scale m (then m=10) is the result of 10 to the power of n.

Function code	Name	Parameter description	Setting range	Default
P01.20	Wake-up-from-sleep delay	0.0–3600.0s (valid when P01.15 is 2)	0.00–3600.0	0.0s
P01.21	Power-off restart selection	0: Disable restart 1: Enable restart	0–1	0

The value specified in "Setting range" or "Default" contains one decimal place, and therefore the fieldbus scale is 10. If the value received by the upper computer is 50, "Delay of auto fault reset" of the rectifier is 5.0 (5.0=50/10).

To set "Wake-up-from-sleep delay" to 5.0s through Modbus communication, you need first to multiply 5.0 by 10 according to the scale to obtain an integer 50, that is, 32H in the hexadecimal form,

After receiving the command, the VFD converts 50 into 5.0 based on the fieldbus

scale, and then sets "Wake-up-from-sleep delay" to 5.0s.

5.4.6 Error message response

Operation errors may occur in communication-based control. For example, some parameters can only be read, but a write command is sent. In this case, the VFD returns an error message response.

Error message responses are sent from the VFD to the master. The following table lists the codes and definitions of the error message responses.

Code	Name	Meaning
01H	Invalid command	The command code received by the host controller is not allowed to be executed. The possible causes are as follows: The function code is applicable only on new devices and is not implemented on this device. The slave is in faulty state when processing this request.
02H	Invalid data address	For the VFD, the data address in the request of the upper computer is not allowed. In particular, the combination of the register address and the number of the to-be-sent bytes is invalid.
03H	Invalid data value	The received data domain contains a value that is not allowed. The value indicates the error of the remaining structure in the combined request. **Note:** It does not mean that the data item submitted for storage in the register includes a value unexpected by the program.
04H	Operation failure	The parameter setting is invalid in the write operation. For example, a function input terminal cannot be set repeatedly.
05H	Incorrect password	The password entered in the password verification address is different from that is specified by P07.00.
06H	Incorrect data frame	The data frame sent from the upper computer is incorrect in the length, or in the RTU format, the value of the CRC check bit is inconsistent with the CRC value calculated by the lower computer.
07H	Parameter read-only	The parameter to be modified in the write operation of the upper computer is a read-only parameter.
08H	Parameter cannot be	The parameter to be modified in the write operation of the upper computer cannot be modified during the

Code	Name	Meaning
	modified in running	running of the VFD.
09H	Password protection	If the upper computer does not provide the correct password to unlock the system to perform a read or write operation, the error of “system being locked” is reported.

When returning a response, the slave uses a function code domain and fault address to indicate whether it is a normal response (no error) or exception response (an error occurs). In a normal response, the slave returns the corresponding function code and data address or sub-function code. In an exception response, the slave returns a code that is equal to a normal code, but the first bit is logic 1.

For example, if the master sends a request message to a slave for reading a group of function code address data, the following code is generated:

0 0 0 0 0 0 1 1 (03H in the hexadecimal form)

In a normal response, the slave returns the same function code. In an exception response, the slave returns the following code:

1 0 0 0 0 0 1 1 (83H in the hexadecimal form)

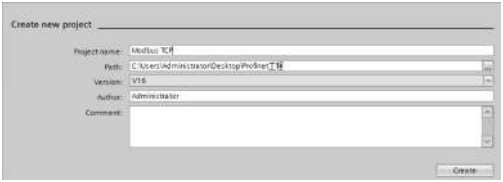
In addition to the modification of the code, the slave returns a byte of exception code that describes the cause of the exception. After receiving the exception response, the typical processing of the master is to send the request message again or modify the command based on the fault information.

5.5 PLC communication example 1 (S7-1200)

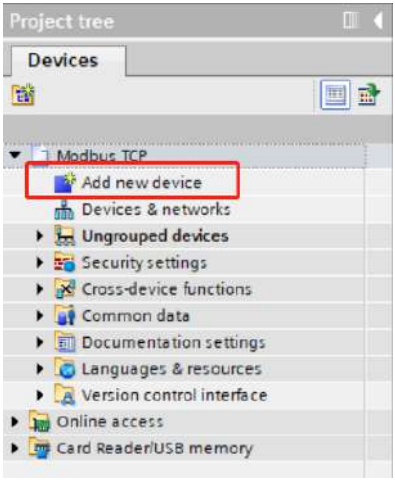
This example illustrates how to use the Modbus TCP communication expansion card to communicate with a SIEMENS PLC (using TIA Portal V15 software as a configuration tool), where Modbus TCP does not have a device description file.

Step 1 Use the TIA Portal V15 software to add a Modbus TCP program block.

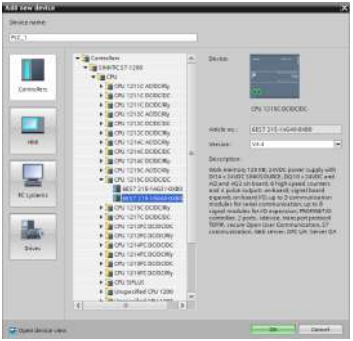
Step 2 Open TIA Portal V15, create a new project as shown in the following figure.



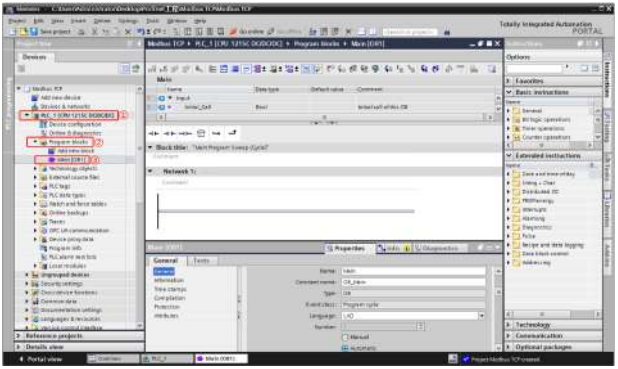
Step 3 After creating, click the **Project View** in the bottom left corner. Double click **Add new device** in the opened interface. See the following figure.



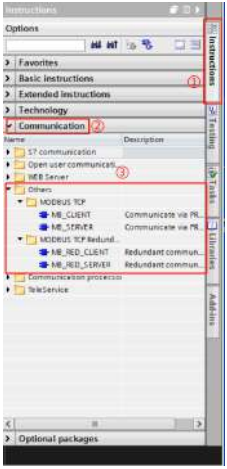
Step 4 Select the corresponding PLC model. The PLC model used in this example is shown in the following figure. Click **OK**.

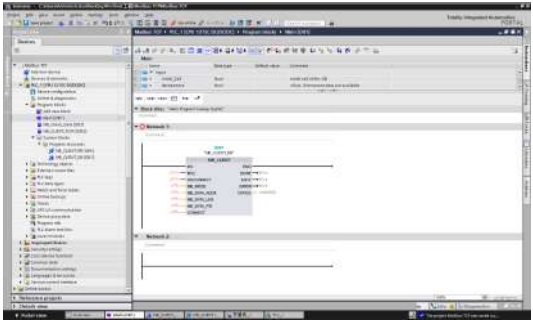


Step 5 On the left, choose **Program Block**, and double click **Main[OB1]** to open the programming interface, as shown in the following figure.



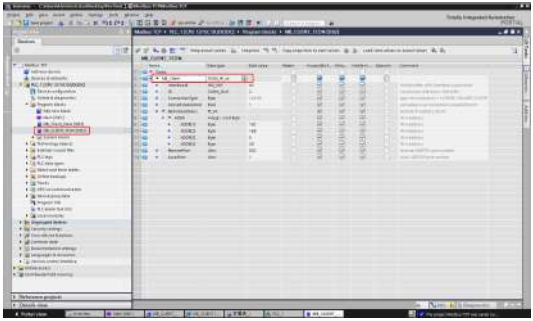
Step 6 On the right, choose **Instructions** > **Communication** > **Others** > **MODBUS TCP**, double click or drag **MB_CLIENT** to add it to **Main[OB1]**, as shown in the following figure.



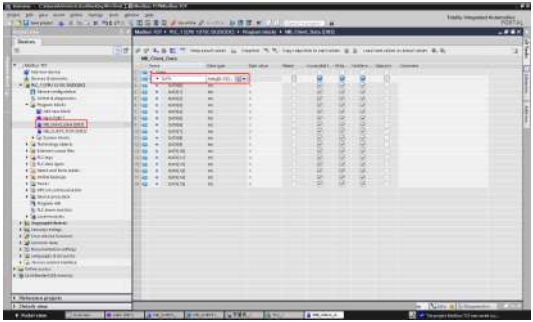


Step 7 On the left, choose **Program blocks** > **Add new block**, a global data block named **MB_CLIENT_TCON** for mapping the pin CONNECT of the function block MB_CLIENT. Open the data block, create a variable **MB_Client**,

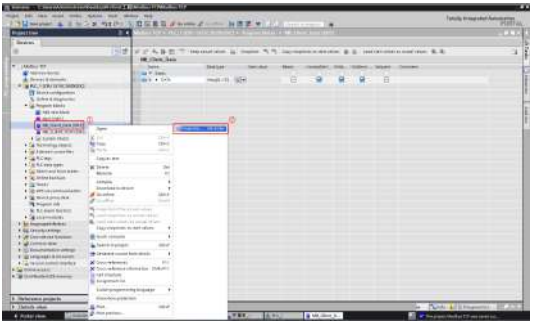
manually enter **TCON_IP_v4** in the data type box, and configure the parameters of the data block.

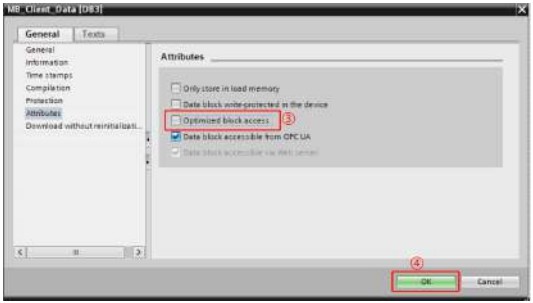


Step 8 Add another global data block for data storage, named **MB_Client_Data**. Open this data block and create variable **DATA**.



Step 9 Change the attributes of data blocks **MB_CLIENT_TCON** and **MB_Client_Data**, and deselect **Optimized block access**.

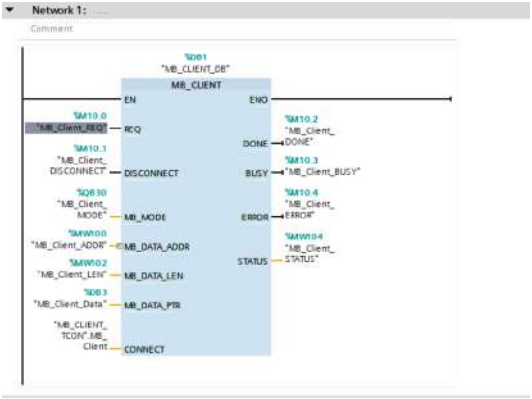




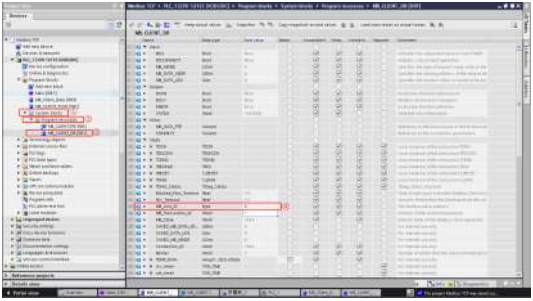
Step 10 In PLC variables, create a new variable table and define the following variables.



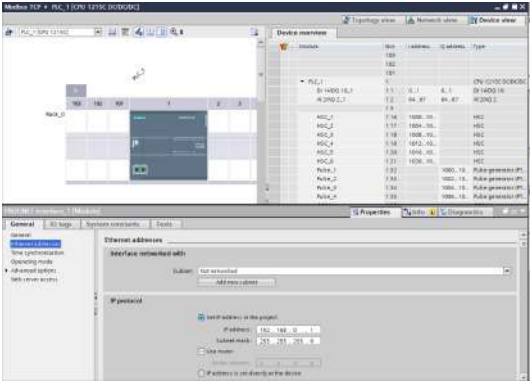
Step 11 Connect the variables to the corresponding pins of the **MB_CLIENT** block.



Step 12 Change the ID in the MB_CLIENT background data block to match the connection ID. This can be modified by choosing **Program blocks** > **System blocks** > **Program resources** > **MB_Client_DB** on the left. Modify the ID to be consistent with the ID in **MB_CLIENT_TCON**.

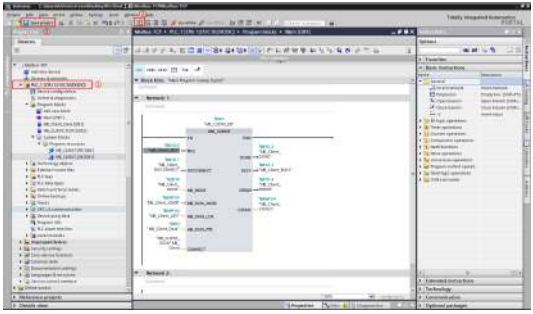


Step 13 Change the PLC IP address to match the Modbus TCP slave station. Double click **Device view**, right-click the network interface position, choose **Properties**, and set the parameters in the pop-up interface.

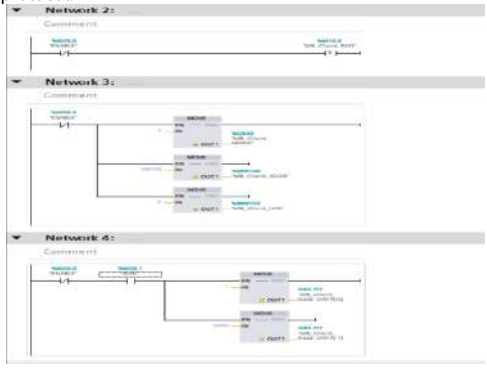


Step 14 At this point, a Modbus TCP connection has been established. Right click **PLC_1 [CPU 1215C DC/DC/DC]**, and choose **Compile** > **Hardware and software (change only)** to compile the entire project. Click **Save Project**

to save the entire project, then click the **Download to Device** icon to download the project configuration to the PLC. See the following figure.



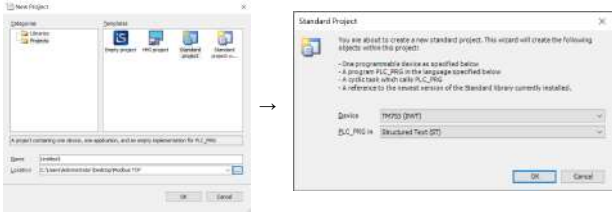
Step 15 You can now write and download the following PLC program. Additionally, set P00.01=2, P00.02=0, P00.06=8, P14.00=2 (consistent with the ID in **MB_CLIENT_TCON**), P16.58–P16.61=192.168.0.20 (consistent with the IP in **MB_CLIENT_TCON**), and keep P16.62–P16.69 at their default values. This will allow control of the VFD to run at 50.00Hz using the Modbus TCP protocol.



5.6 PLC communication example 2 (TM753)

This example illustrates how to use the VFD Modbus TCP module to communicate with TM753 that serves as the master station.

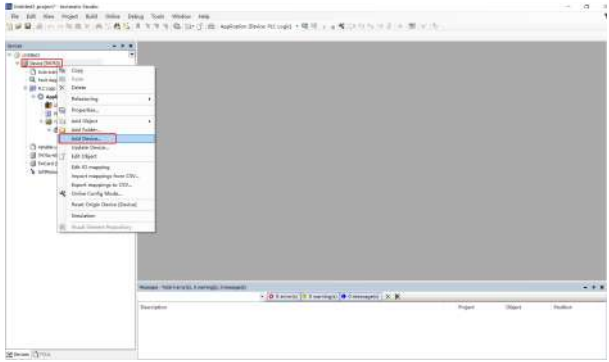
Step 1 Open the Invtmatic Studio software and create a new project TM753. Select device **TM753**. The interface is shown in the following figure.



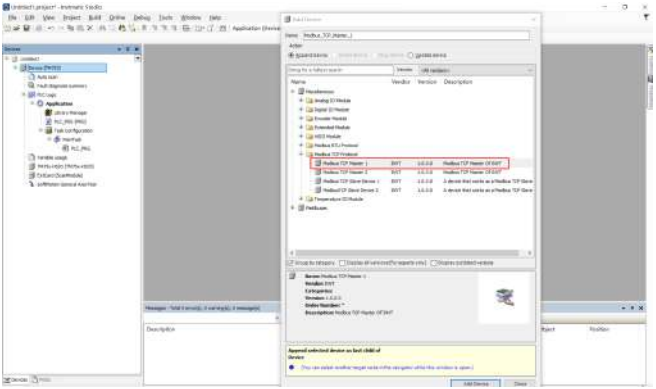
Step 2 Add Modbus TCP related configuration.

According to the actual cable connection to the Ethernet port of TM753, select **Modbus TCP Master1** or **Modbus TCP Master2**; in this example, the Ethernet 1 port is actually connected, so select **Modbus TCP Master1**. After adding the master station, select the corresponding **Modbus TCP Slave1**.

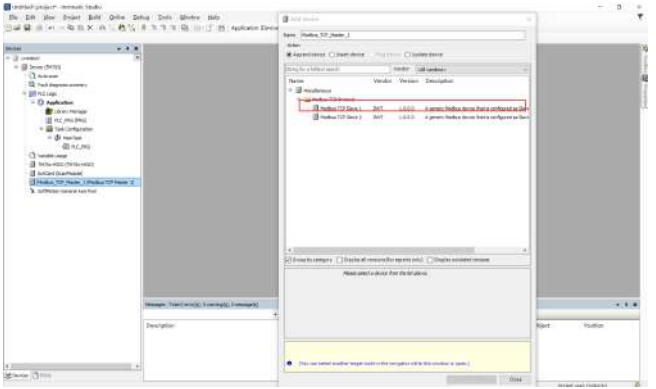
- Add devices.



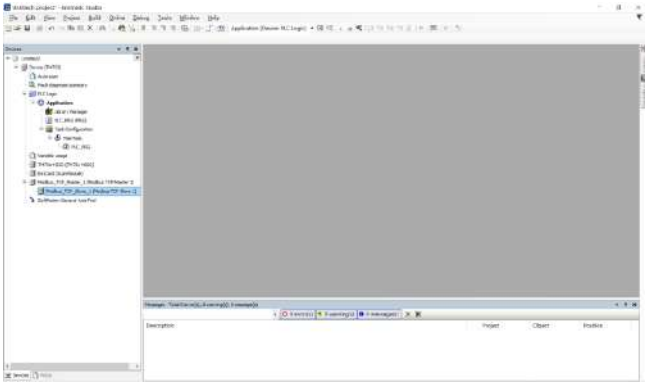
- Add the Modbus TCP master station.



- Add the Modbus TCP slave station.



- The adding is completed.

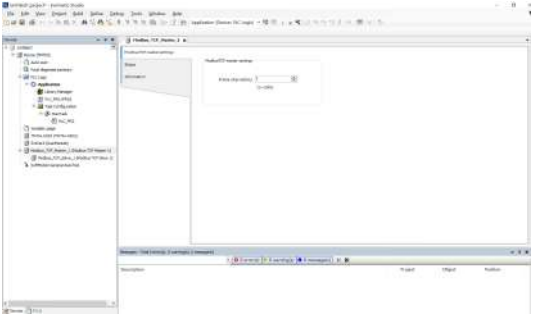


The Modbus TCP configuration is completed. If there are multiple Modbus TCP slave stations, add multiple **Modbus TCP Slave**.

Step 3 Perform Modbus TCP master station parameter setting.

Double click the master station device in the device tree to open the Modbus master station configuration window. Double click **Modbus_TCP_Master1** as shown in the following figure for Modbus TCP master station parameter configuration.

- Perform Modbus TCP master station configuration.



The frame interval refers to the time interval between the main station receiving the last response data frame and waiting for the next request data frame. This parameter can be used to adjust the data exchange rate.

Step 4 Perform Modbus TCP slave station configuration.

Double click the slave device in the device tree to open the Modbus TCP slave station configuration window, and double click **Modbus_TCP_Slave1** to set the Modbus TCP slave parameters.

- Modbus TCP slave station configuration

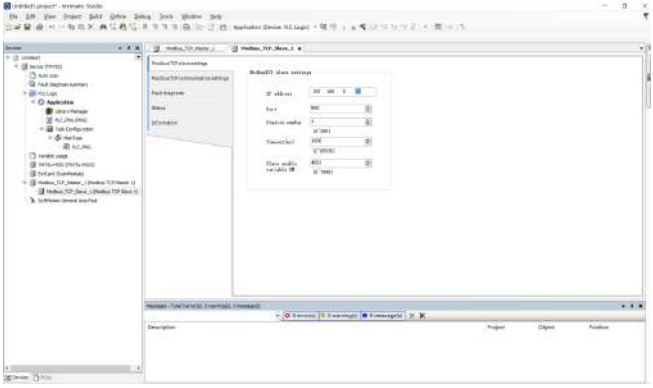


Table 5-5 Configuration parameters

Configuration item	Function
IP address	IP address of the Modbus TCP slave station to which the master station connects.
Port	TCP port number of the Modbus TCP slave station to which the master station connects.
Station number	Protocol station address of the Modbus TCP slave station to which the master station connects.
Timeout time (ms)	If the slave station does not respond within this duration after the master station sends a frame, the master station will report a receiving timeout.

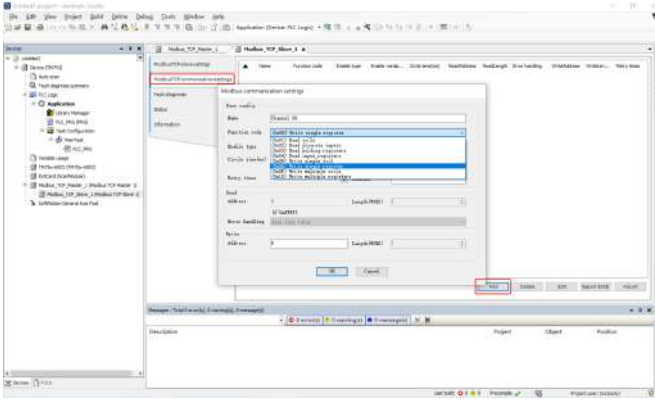
Configuration item	Function
Slave enable variable	After enabling the variable in the program, the master station will start sending communication frames to the slave station.

- Example

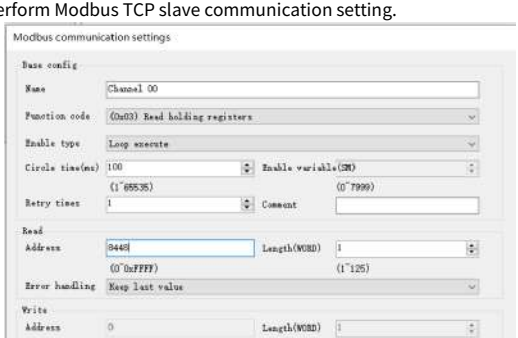
Configuration item	Setting
IP address	192.168.0.20
Port	502
Station number	1
Timeout time (ms)	1000
Slave enable variable	4001

Click **ModbusTCP communication settings** to configure Modbus TCP commands. After clicking **Add**, a dialog box will appear for adding a new channel for the Modbus TCP slave station. Select the function code, set the operation address and length, and click **OK** button to create a new channel. Each channel represents an independent Modbus TCP request.

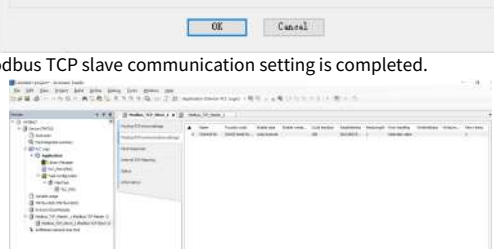
- Configure Modbus TCP commands.



- Perform Modbus TCP slave communication setting.



- Modbus TCP slave communication setting is completed.



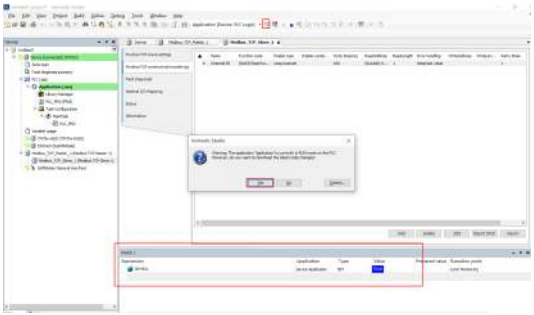
As shown in the preceding figure, a Modbus TCP request is defined to cyclically read the holding register (function code 03) with a 100ms period. The read address is 0x2100 (VFD status word).

Step 5 Compile and download.

So far, the communication configuration is completed, and compile and download to the PLC. Click **Log in** to download, and after the download is complete, run the PLC. Set the VFD IP address, P16.58–P16.61 to **192.168.0.20**, set the station address P14.00 to **1**, keep default settings for P16.62–P16.69. Keep consistent with Modbus TCP slave settings, and set SM4001 enabling to

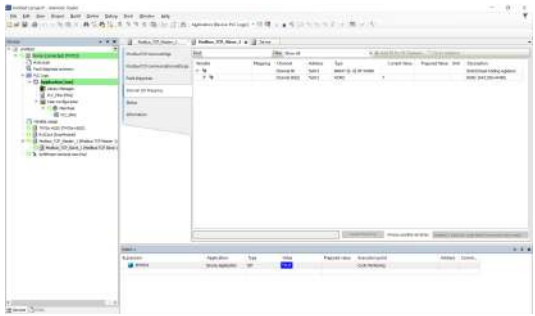
True to establish successful Modbus TCP communication.

- Compile and download.



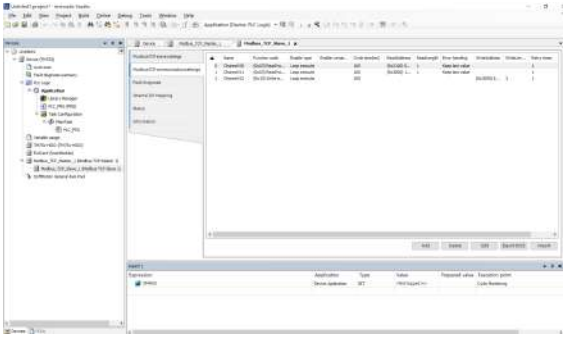
After adding the master/slave communication configuration in the Modbus TCP slave communication settings, the values that are read can be viewed in the Internal I/O mapping interface. The internal I/O mapping will automatically allocate a mapping address for each configured mapping, such as %IW1 in the first row of the following figure, which represents mapping a value read from a register to address %IW1.

- View internal I/O mapping parameters.



Similarly, multiple Modbus TCP commands can be added to control the VFD, for example, adding control word (address 0x2000), setting the set frequency (address 0x2001), checking status word (address 0x2100) and running frequency (address 0x3000) settings. Adding commands are shown in the following figure. The control word and set frequency addresses are contiguous, directly adding the function of writing multiple registers, with a length set to 2.

- Multiple Modbus TCP command settings

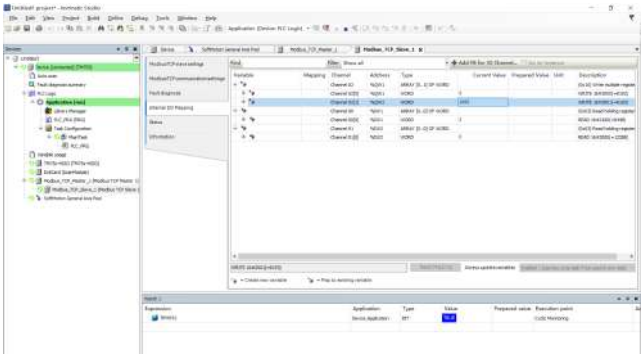


After adding, download to PLC and run it. Enable SM4001, then check the Internal I/O mapping interface.

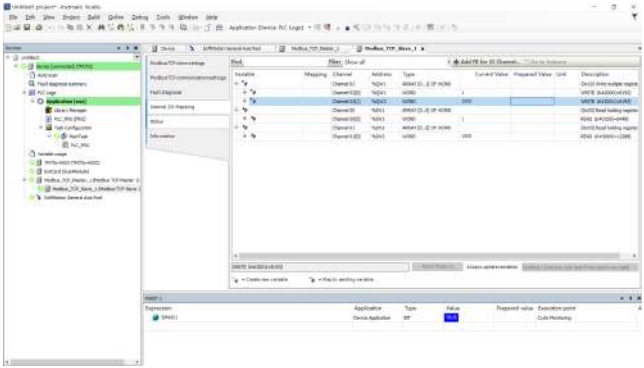
Set VFD function codes, P00.01=2, P00.02=0, and P00.06=8.

As shown in the following figure, give the running command (1: forward run) and set the frequency (1000: 10Hz) on the internal I/O mapping interface to control the VFD to run at 10Hz frequency using Modbus TCP.

- Parameter reference



- Parameter view



6 PowerLink protocol

6.1 Overview

The communication card using this protocol is defined as a PowerLink slave station, which can be used on VFDs that support PowerLink communication.

6.2 Product features

6.2.1 Supported functions

- Supports the PowerLink CiA 402 protocol.
- Configured with a slave station XDD configuration file, it can communicate with B&R PLC and other master stations.
- Automatic network address configuration
- Equipped with two RJ45 ports, supporting 10/100M half duplex operating.
- Applicable to linear and star network topologies.

6.2.2 Supported services

- PDO service
- SDO service
- Use of SDO to read and write VFD function codes
- Manufacturer-defined object dictionary
- Two types of objects for VFD reading and writing, that is, PDOs and SDOs (requiring the object dictionary defined by the manufacturer)

6.2.3 Status indicator

The EtherCAT communication card provides six indicators to indicate its states. Figure 6-1 shows the positions of the four indicators. Table 6-1 provides the indicator definitions.

Figure 6-1Status indicator positions

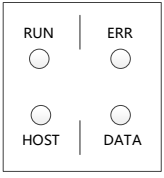


Table 6-1 Status indicator definitions

Indicator	Color	Definition	Function
LED1 (RUN)	Green	Steady on	In NMT_CS_OPERATIONAL state
		Blinking (on for 50ms, off for 50ms)	In NMT_CS_BASIC_ETHERNET state.
		Blinking (on for 200ms, off for 200ms)	In NMT_CS_STOPPED state
		Single flash (on for 500ms, off for 500ms)	In NMT_CS_PRE_OPERATIONAL_1 state.
		Double flashes (on for 250ms, off for 250ms)	In NMT_CS_PRE_OPERATIONAL_2 state.
		Triple flashes (on for 125ms, off for 125ms)	In NMT_CS_READY_TO_OPERATE state
		Steady off	In Init state.
LED2 (HOST)	Green	Steady on	The communication card is in the process of handshaking with the VFD.
		Blinking (on for 500ms, off for 500ms)	The communication card and VFD communicate normally. **Note:** After the handshaking is completed, it should blink regardless of whether there is data transmission between the communication card and the main control board.
		Steady off	The communication card is in the initialization or parameter configuration phase.
LED3 (DATA)	Green	Blinking (on for 500ms, off for 500ms)	The data update between the communication card and main control board is normal.
		Steady off	No data update or abnormal update

Indicator	Color	Definition	Function
			between the communication card and main control board.
LED4 (POWER)	Red	Steady on	3.3V power indicator
LED5 (ERR)	Red	Steady on	Data frame lost, communication disconnected, or NMT internal critical error (such as illegal interruption and memory overrun).
		Steady off	No fault
LED6 (SYS)	Green	Blinking (on for 500ms, off for 500ms)	Communication card heartbeat indicator (communication card is running normally).

6.3 Electrical connection

A PowerLink network often consists of a master station (such as PLC) and multiple slave stations (such as drives or bus expansion terminals). Each PowerLink slave station has two standard Ethernet interfaces. Figure 6-2 and Figure 6-3 show the electrical connections.

Figure 6-2 Linear network topology electrical connection

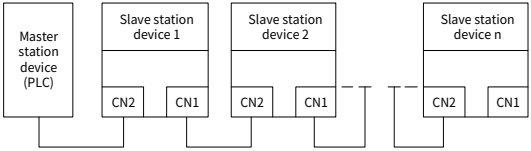
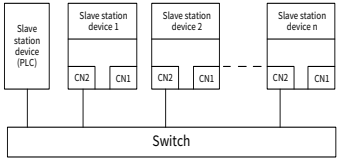


Figure 6-3 Star network topology electrical connection

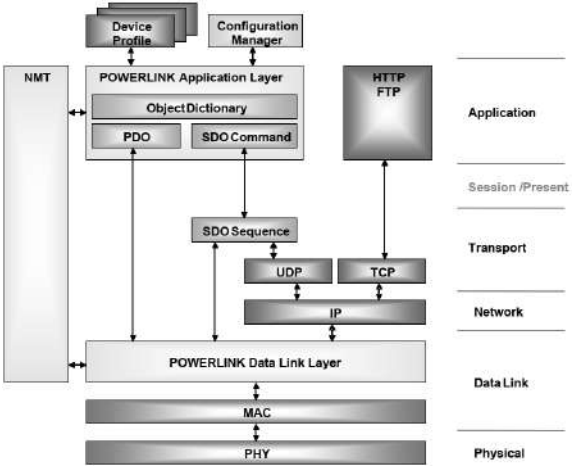


Note: For the star network topology, you need to prepare switches and hubs.

6.4 Communication

6.4.1 PowerLink reference model

Figure 6-4 PowerLink reference model



The PowerLink network reference model consists of two parts: the data link layer and the application layer. The data link layer is primarily responsible for the PowerLink communication protocol, which utilizes an additional bus scheduling mechanism to extend the standard Ethernet data link layer, ensuring that only one node can access the network at the same time. The scheduling mechanism is divided into isochronous phase and asynchronous phase. The application layer embeds the CANopen drive Profile (DS402) communication protocol. The object dictionary in CoE includes parameters, application data, and PDO mapping configuration information.

Process data objects (PDO) are composed of mappable objects from the object dictionary, and the content of PDO data is defined by PDO mappings. The reading and writing of PDO data is periodic and does not require searching for the object dictionary, while email communication (SDO) is non-periodic communication and requires searching for the object dictionary when reading and writing.

6.4.2 PowerLink slave station information

The PowerLink slave station information file (XML file) is used for the master station to read, in order to construct the configuration between the master and slave stations. The XML file contains the information necessary for PowerLink communication settings.

6.4.3 PowerLink state machine

The PowerLink state machine is used to describe the states of the slave application and state transitions. State change requests are typically initiated by the master station, with responses from the slave station. The specific state transition method is as shown in the following figure.

Figure 6-5 PowerLink state machine flowchart

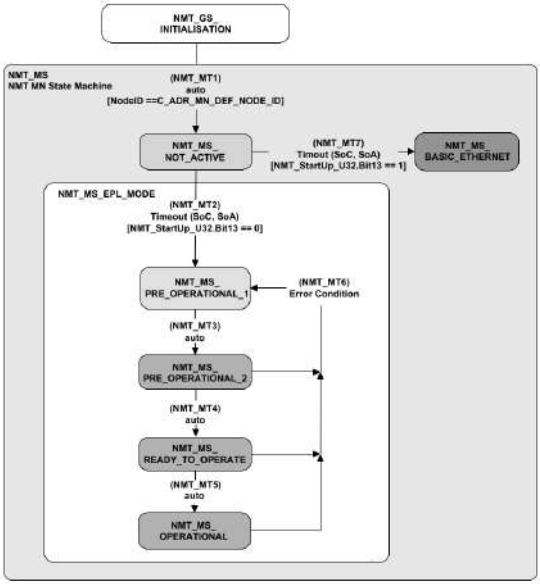


Table 6-2 PowerLink state machine description

Status	Description
NMT_GS_INITIALISATION	In the power-up phase, the communication attributes are initialized, and the node ID configuration phase is established.
NMT_MS_NOT_ACTIVE	During the node management phase, the Master Node (MN) acquires the operational status of the slave nodes through the Service Object Address (SOA) and Service Object Command (SOC) frames. SDO data is unavailable, and PDO data is also unavailable.
NMT_MS_PRE_OPERATIONAL_1	The MN executes non-real-time cyclical communication, completing the transmission of communication attribute configuration data while verifying IDs and downloading the configuration PDO mapping. SDO data is available, whereas PDO data remains unavailable.
NMT_MS_PRE_OPERATIONAL_2	The MN performs a synchronous cycling process, initiating the polling of identified Communication Nodes (CN), as well as those not flagged as AsyncOnly by the Preq frames. Simultaneously, it verifies the heartbeat of the nodes, searching for configured but unrecognized CNs through the SoA IdentRequest frame. SDO data is accessible, while PDO data remains unavailable.
NMT_MS_READY_TO_OPERATE	The MN continues with the synchronous cycling process and is expected to begin identifying time-critical CNs according to PDO mapping requirements. The MN cyclically accesses recognized asynchronous CNs through the SoA StatusRequest frames. SDO data is available, whereas PDO data remains unavailable.
NMT_MS_OPERATIONAL	The MN engages in a synchronous cycling process, sending the NMTStartNode command to force the CN from NMT_CS_READY_TO_OPERATE

Status	Description
	to NMT_CS_OPERATION. Cyclic access to recognized asynchronous CNs is performed through the SoA status request frames. The search for configured but unrecognized CNs is performed via the SoA IdentRequest frames. SDO data is available; PDO data is available.

6.4.4 PDO process data mapping

The process data of the PowerLink slave station is always executed in a synchronous manner through PReq and/or PRes frames. Among them, PRes frames are sent in broadcast mode using a producer/consumer pattern, while PReq frames with unicast addresses follow a master/slave relationship. The transfer type of PDO is continuous transfer, and does not support "event-driven" or "change-driven" transfer types.

From the perspective of devices, the use of PDOs can be divided into two types: data transmission and data reception. The distinction between transmitting PDOs (TPDOs) and receiving PDOs (RPDOs) needs to be made. Devices that support TPDO are called PDO producers or PDO master devices, while devices that can receive PDOs are called PDO consumers or PDO slave devices. The PowerLink slave station with application control function should support the reading of PDO mapping objects, with a maximum data size of 10 for each single PDO access.

The master station can select the required objects for PDO mapping from the object dictionary. PDO mapping configuration is located in the Object Dictionary from the 1600h to 1603h area (RxPDOs: Receive PDOs) and from 1A00h to 1A03h area (TxPDOs: Transmit PDOs).

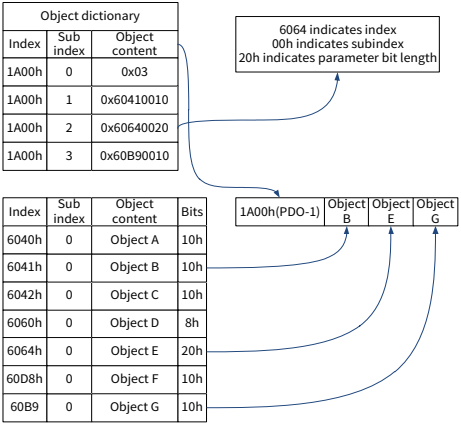
The master station can select the required objects for PDO mapping from the object dictionary. The PDO mapping configuration is located in the following area of the object dictionary:

- 1600h to 1603h: Receive PDOs (RxPDOs)
- 1A00h to 1A03h: Transmit PDOs (TxPDOs)

Note: Mapping is only supported for 0x6000 series objects, and not supported for 0x4000 series manufacturer-specific parameter mapping. Even if added to the PDO mapping, they will still be transmitted through the SDO channel. In the EPL, the maximum number of data accessed in a single SDO is 10. When the number exceeds 10, only the last 10 data can be successfully written.

Figure 6-6 shows the PDO mapping method.

Figure 6-6 PDO mapping method



6.4.5 Clock synchronization

PowerLink complies with the IEEE1588 distributed clock system standard to achieve high-precision clock synchronization. Each device comes with a clock to ensure precise synchronization during data exchange. The PowerLink cycle consists of two main roles: master node (MN, management node) and child node (CN, controlled node, also known as slave).

During the startup phase of the PowerLink cycle, the master node will send configuration information to each slave node and multicast synchronization start frames (SoC) to all nodes through Ethernet. After receiving the SoC, a slave node enters a waiting state for data communication.

After the SoC, the master node (MN) starts communicating with each slave node in turn:

1. The MN first sends PReq1 to the first node, which will broadcast PRs1 to the network after receiving it.
2. Subsequently, the MN sends PReq2 to the second node, which will broadcast PRs2 to the network after receiving it.
3. This process continues in a cyclic manner until the final slave node broadcasts

PRsN.

4. Finally, the MN sends the Start of Asynchronous frame (SoA), marking the beginning of the asynchronous communication phase.

After SoA, the network entered the asynchronous data transmission phase. The entire process forms a complete PowerLink loop frame, consisting of the following two stages:

Isochronous synchronization phase: This is accomplished through the SoC and PReq/PRs.

Asynchronous data transmission phase: This phase begins with the SoA.

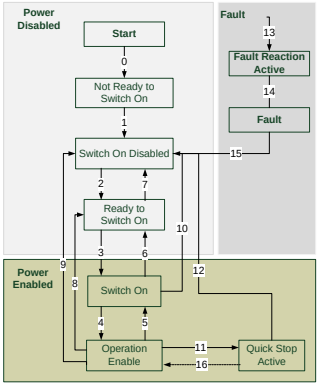
The timing for all phases is configurable to accommodate various application requirements.

6.5 CiA402 device profile

The main station controls the drive through control word (0x6040), and reads status word (0x6041) to obtain the actual status of the drive. The servo drive achieves motor control internally based on the control instructions from the master station.

6.5.1 CANopen over PowerLink state machine

Figure 6-7 CANopen over PowerLink state machine



State	Description
Not Ready to Switch On	The drive is in the initialization process.

Switch On Disabled	Drive initialization is completed.
Ready to Switch On	The drive is waiting to enter the Switch On state, and the motor is not excited.
Switched On	The drive is ready, and the main circuit power is normal.
Operation Enable	The drive is enabled and it controls the motor in accordance with the control mode.
Quick Stop Active	The drive stops according to the set mode.
Fault Reaction Active	The drive detects that an alarm has occurred, and stops according to the set mode, while the motor still has excitation signal.
Fault	The drive is in a fault state and the motor has no excitation signal.

The control word 6040h includes the following content:

- Bits for state control
- Bits related to control mode
- Bits defined by manufacturer

The detailed description of each bit of 6040h is as follows.

15 11	10 9	8	7	6	4 3	2	1	0
Factory defined	Reserved	Reserved	Fault reset	Operation mode	Servo running	Quick stop	Switch on main circuit	Servo being ready
O	O	O	M	O	M	M	M	M
MSB	LSB							

Bits 0–3 and bit 7 (bits for state control):

Command	Bit of the control word					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	X	X	0	X	7,9,10,12
Quick stop	0	X	0	1	X	7,10,11

Command	Bit of the control word					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset	0–1	X	X	X	X	15

Bits 4, 5, 6, and 8 (bits related to control mode):

Bit	Operation mode		
	Profile position mode	Profile velocity mode	Homing mode
4	New set-point	Reserved	Homing operation start
5	Change set immediately	Reserved	Reserved
6	Rel	Reserved	Reserved
8	Halt not supported	Halt not supported	Halt not supported

Note: In the Profile position mode, when bit4=New set-point, a new position can be triggered.

When the control word is set to 0x0F, the drive is enabled; otherwise, the drive stops. If a fault occurs, a reset command is issued when bit 7 in the control word is set to 1.

6041h status word includes the following content:

- Drive's present status bit
- Status bits related to control mode
- Status bits defined by manufacturer

The detailed description of each bit of 6041h is as follows.

Bit	Description	M/O
0	Ready to switch on	M
1	Switched on	M
2	Operation enabled	M
3	fault	M
4	Voltage enable	M
5	Quick stop	M
6	Switch on disabled	M
7	Warning	O
8	Manufacture specific	O
9	Remote	M

Bit	Description	M/O
10	Target reached	M
11	Internal limit active	M
12–13	-	-
14–15	Manufacturer specific	O

Bits0–3, 5, and 6:

Value(binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Bit 4: Voltage enable, when this bit is set to 1, it indicates that the main circuit power supply is normal.

Bit 9: Remote, when this bit is 1, it indicates that the slave station is in OP state, and the master station can control the drive through PDOs.

Bit 10: Target reached.

Mode 1 and mode 8: After the present positioning is completed, set it to 1 for keeping. If the positioning is restarted, clear it to 0.

Mode 6: Set it to 1 after homing is completed; set it to 0 if homing is not completed.

Mode 2: Set it to 1 when the ramp reference frequency reaches the set frequency and the control word's bit 4, bit 5, and bit 6 are all set to 1; otherwise, set it to 0.

Other modes: 0

Bit 11: External limitations

Mode 1 and mode 8: Set it to 1 when FWD/REV limit is reached; otherwise set it to 0.

Mode 4 and mode 10: In electric mode: If the electric torque reaches the upper limit (upper limit is not 0), set it to 1; otherwise, set it to 0. In braking mode, if the braking torque reaches the upper limit (upper limit is not 0), set it to 1; otherwise set it to 0.

Mode 2, mode 3, and mode 9: Set it to 1 when the output frequency reaches P00.03; otherwise, set it to 0.

Other modes: 0

Bit 12: Manufacturer defined 1

Mode 1: Set to 1 when bit 4 of the control word is 1; otherwise, set to 0.

Mode 8, mode 9, and mode 10: Set it to 1 when the VFD is in running state; otherwise, set it to 0.

Other modes: 0

Bit 14: When this bit is 1, it indicates the motor is in zero-speed state.

Bit 7–Bit8, Bit 15: Reserved

6.5.2 Device run mode

Set the VFD parameters P00.01=2 (Running command channel), P00.02=3 (PowerLink communication channel), and P16.75 (Communication timeout time). The position mode and homing mode require the drive to operate in closed-loop mode.

6.5.2.1 Profile Position Mode

1. Set 【6060h: Mode of operations】 to 1 (Profile Position Mode).
2. Set the appropriate number of encoder pulses P20.01 according to the encoder model, set the position command source P21.00 tens place to 1; for digital position, set P21.16 digital positioning mode (16#0200, bit 9=1 Position source PowerLink given).
3. Set 【6081h: Profile velocity】, with the frequency-to-velocity relationship V=60f/p, where V is the velocity in RPM (revolutions per minute), f is the frequency, and p is the number of motor pole pairs.
4. Set 【6083h: Profile acceleration】 and 【6084h: Profile deceleration】 and write the corresponding values to P21.20 and P21.21.

   Note: The units of 6083h and 6084h are ms.
5. Set 【6040h: Control word】 to enable the drive (set to 0x4F to enable, bit 6 set to 1, only relative position supported).
6. Set【607Ah: Target position】to the target position (unit: user unit).
7. Query 【6064h: Position actual value】to obtain the motor actual position feedback.
8. Query 【6041h: Status word】 to obtain the drive status feedback (following error, target reached, and internal limit active).

9. For function details, see function parameters in group P21 in function code list chapter and commissioning chapter in the GD350 user manual.

6.5.2.2 VFD Mode

1. Set 【6060h: Mode of operations】 to 2 (VFD mode).
2. Set 【6046h: vl velocity min max amount】 to set the max. and min. rotation speeds. If you do not set it, the default values on the drive are used.
3. Set the object dictionary 【6048h: vl velocity acceleration】, 【6049: vl velocity deceleration】, and 【604Ah: QuickStop Speed】. Do not set them to the default values of the drive.
   - The acceleration time is 60* 【6048h:02 Acceleration Delta Time】 *P00.04/(【6048h:01 Acceleration Delta Speed】* Number of motor pole pairs) * 0.1, with second as the unit, corresponding to P00.11. The value will remain the same as the previous one when it exceeds 3600.0 seconds.
   - The deceleration time is 60* 【6049h:02 Deceleration Delta Time】 *P00.04/(【6049h:01 Deceleration Delta Speed】* Number of motor pole pairs)*0.1, with second as the unit, corresponding to P00.12. The value will remain the same as the previous one when it exceeds 3600.0 seconds.
   - The emergency stop time is 60 * 【604Ah:02 QuickStop Delta Speed】 * P00.04/(【604Ah:01 QuickStop Delta Time】 * Number of motor pole pairs)*0.1, with second as the unit, corresponding to P00.12. The value will remain the same as the previous one when it exceeds 60.0 seconds.
4. Set 【604Ch: vl dimension factor】to adjust the electronic gear ratio, which is 1:1 by default.
5. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
6. Set 【6042h: vl target velocity】to set the target speed.
7. Set 【6040h: Control word】to run the drive (when it is set to 0x7F, the drive runs).
8. Query 【6044h: vl velocity actual value】to obtain the motor actual position feedback.

6.5.2.3 Profile Velocity Mode

1. Set 【6060h: Mode of operations】 to 3 (Profile Velocity Mode).
2. Set 【6083h: Profile acceleration】 and 【6084h: Profile deceleration】 and write the corresponding values to P00.11 and P00.12.

Note: The units of 6083h and 6084h are ms.

3. Set P00.01=2, P00.02=3, and P00.06=13.
4. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
5. Set 【60FFh: Target velocity】 to set the target rotational speed (unit: rpm).
6. Query【6041h: Status word】to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

6.5.2.4 Profile Torque Mode

1. Set 【6060h: Mode of operations】 to 4 (Profile Torque Mode).
2. Set 【6087h】 to set the ramp torque.
3. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
4. Set P03.11=11 and P03.32=1 (torque control enabled).
5. Set 【6071h: Target torque】 to set the target torque.
6. Query【6041h: Status word】to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

6.5.2.5 Homing Mode

1. Set 【6060h: Mode of operations】 to 6 (homing mode).
2. Set P22.00.Bit0=1 to enable spindle positioning function, and set P22.03–P22.06 zero positions.
3. Set the 【6040h: Control word】 to enable the drive (set to 0x0F to enable). When bit 4 changes from 0 to 1, the homing operation starts, and when bit4 changes from 1 to 0, the homing operation is interrupted.
4. The motor queries for the limit switch and home switch to complete the homing action.
5. Query 【6041h: Status word】to obtain the drive status feedback (Homing error, Homing attained, Target reached).
6. For function details, see function parameters in group P22 in function code list chapter and commissioning chapter in the GD350 user manual.

6.5.2.6 Touch Probe Function

Record the present position according to the Z phase rising edge or falling edge signal, and set the control word to record one or multiple times.

1. When the control word (60B8h: Touch Probe Control) is 23 (2#01 0111), the function indicates that the Z phase rising edge triggers recording the present position continuously. The recorded value is Turns of motor rotation x 10000. Every time the motor rotates a turn, a value is updated. When the status word (60B9h: Touch Probe Statu) is 3, in continuous triggering mode, the probe status word bit 6 is 1, and each triggering will cause bit 7 to toggle.
2. When the control word (60B8h: Touch Probe Control) is 21 (2#01 0101), the function indicates that the Z phase rising edge triggers recording the present position continuously. The recorded value is Turns of motor rotation x 10000. The status word is 3 (60B9h: Touch Probe Statu).
3. When the control word (60B8h: Touch Probe Control) is 39 (2#11 0111), the function indicates that the Z phase falling edge triggers recording the present position continuously. The recorded value is Turns of motor rotation x 10000. Every time the motor rotates a turn, a value is updated. When the status word is 5 (60B9h: Touch Probe Statu), in continuous triggering mode, the probe status word bit 6 is 1, and each triggering will cause bit 7 to toggle.
4. When the control word (60B8h: Touch Probe Control) is 21 (2#01 0101), the function indicates that the Z phase rising edge triggers recording the present position continuously. The recorded value is Turns of motor rotation x 10000, which is recorded only once. When th status word (60B9h: Touch Probe Statu) is 5, the falling edge triggering control word is 37.

Note: The difference from CiA 402 is that only one type of probe is supported and only one locking value can be recorded at a time.

6.5.2.7 Cyclic Synchronous Position Mode (Not supported)

1. Set 【6060h: Mode of operations】 to 8 (Cyclic synchronous position mode).
2. Set P00.00=3, P00.01=2, P00.02=3, and P00.06=13.
3. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable).
4. Set【607Ah: Target position】to the target position (unit: user unit).
5. Query 【6064h: Position actual value】to obtain the motor actual position feedback.
6. Query 【6041h: Status word】 to obtain the drive status feedback (following error, target reached, and internal limit active).
7. For function details, see function parameters in group P21 in function code list chapter and commissioning chapter in the GD350 user manual.

6.5.2.8 Cyclic Synchronous Velocity Mode (Not supported)

1. Set 【6060h: Mode of operations】 to 9 (Cyclic synchronous velocity mode).
2. Set 【6083h: Profile acceleration】 and 【6084h: Profile deceleration】.
3. Set P00.01=2, P00.02=3, and P00.06=13.
4. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
5. Set 【60FFh: Target velocity】 to set the target rotational speed (unit: rpm).
6. Query【6041h: Status word】to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

6.5.2.9 Cyclic Synchronous Torque Mode (Not supported)

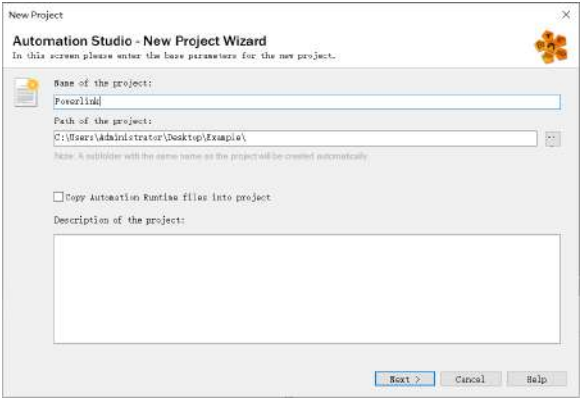
1. Set 【6060h: Mode of operations】 to 10 (Cyclic Synchronous torque Mode).
2. Set P03.11=11 (Communication) and P03.32=1 (Torque control enabled).
3. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.
4. Set 【6072h: Max torque】 to the maximum torque and 【6071h: Target torque】 to the target torque.
5. Query【6041h: Status word】to obtain the drive status feedback (Speed zero, Max slippage error, Target reached, Internal limit active).

6.6 PLC communication example (x20CP3684)

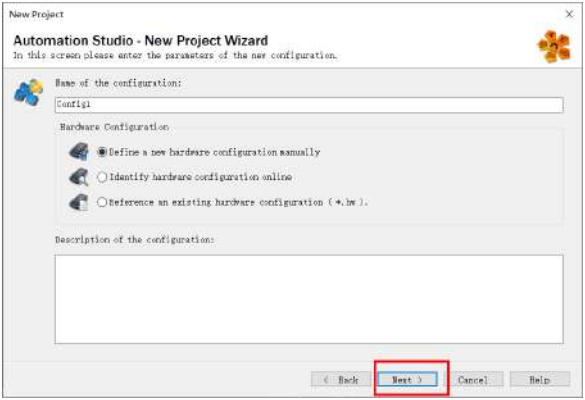
This example illustrates how to use the communication card to establish simple PowerLink communication between the Beckhoff x20CP3684 as the master station and a VFD.

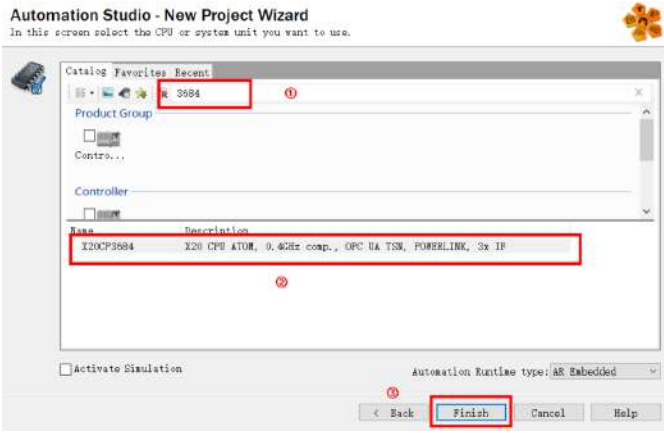
Install Beckhoff Automation Studio 4.12 English.

Open Automation Studio software, and create a new project named X20CP3684. Select device **X20CP3684**. See the following figure.

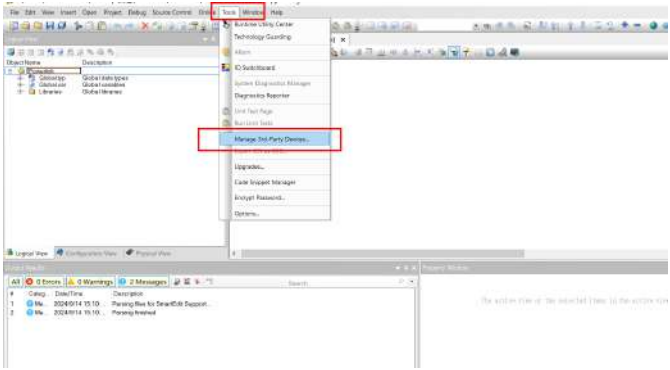


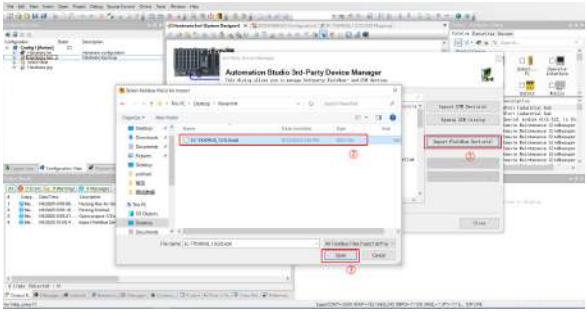
Note: Please ensure that the path created here is entirely in English.





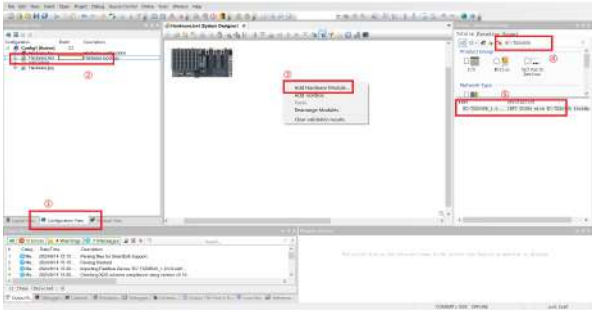
Import the PowerLink configuration file (**EC-TX509U8_1.0.0.0.XDD**). In the menu bar, choose **Tool** > **Manage 3rd-Party Devices...**. In the dialog that appears, select **Import Fieldbus Device(s)**, specify the location of the configuration file (**EC-TX509U8_1.0.0.0.XDD**), select the imported XDD file, and click **Open**; wait for the import to be successful.

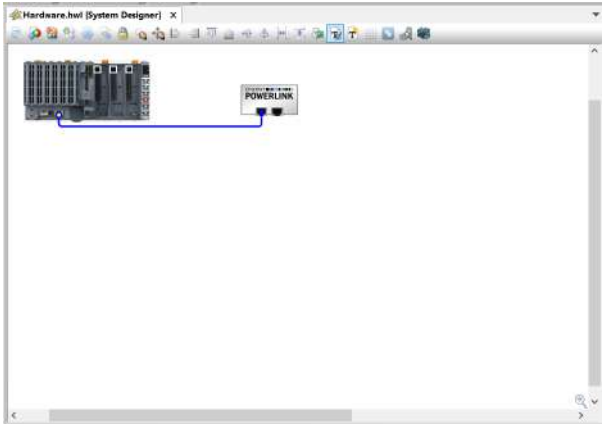




Add devices.

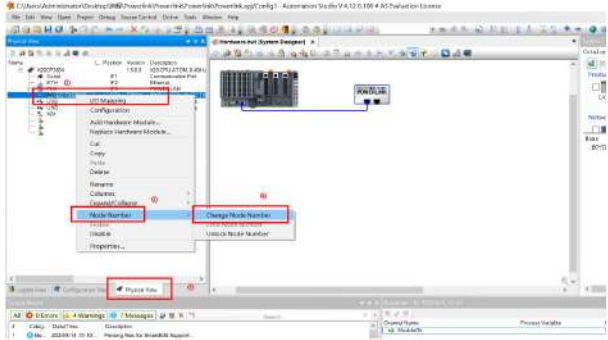
Click **Configuration View**, double click **Hardware** to enter the configuration interface, right click and choose **Add Hardware Model...**, then search for EC-TX509U8 on the right side to quickly locate the device; double click or drag it to a blank area, connecting the device port to the PLC's PowerLink port, thus completing the configuration.





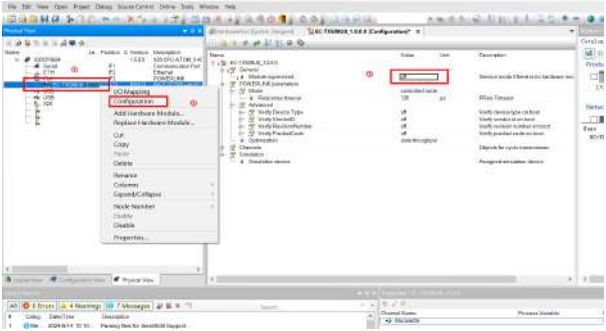
Set slave station parameters.

Click **Physical View** to enter the **Physical View** interface, right click the added PowerLink slave station **EC-TX509U8_1.0.0.0**, and modify the slave site (**1** for example).

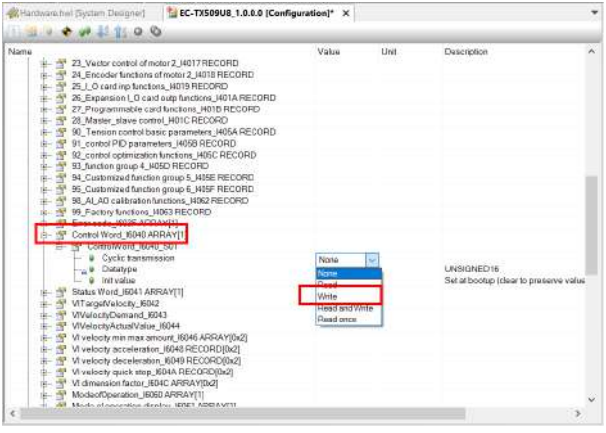


Note: The slave station site number setting need to be consistent with the function code P16.78.

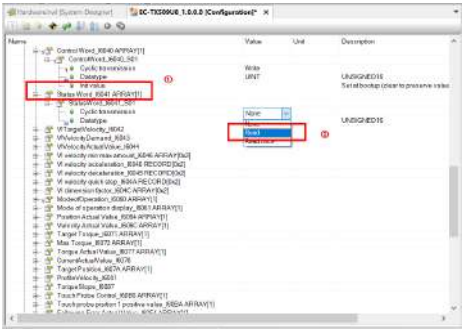
After modifying the address, right click the added PowerLink slave **EC-TX509U8_1.0.0.0** again and choose **Configuration** to enter the configuration interface; when setting up the PowerLink network, disable the PLC's service mode: set **Module Supervised** to **OFF**.



Select **Channels** to configure the PDO parameters. Choose **Control Write**, and set **Cyclic Transmission** to **Write**.



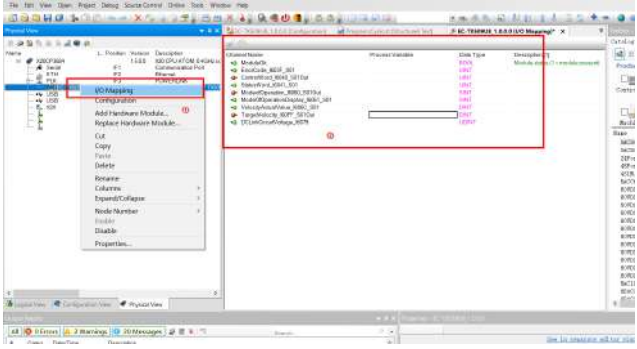
Similarly, set the following parameters to **Write**: **Mode of Operation**; **Target Velocity**. Choose **Status Word**, and set **Cyclic Transmission** to **Read**.



Similarly, set the following parameters to **Read**: Error Code; Mode of Operation Display; VelocityActualValue; DCLinkCircuitVoltage. Once configuration is complete, click **Save**.

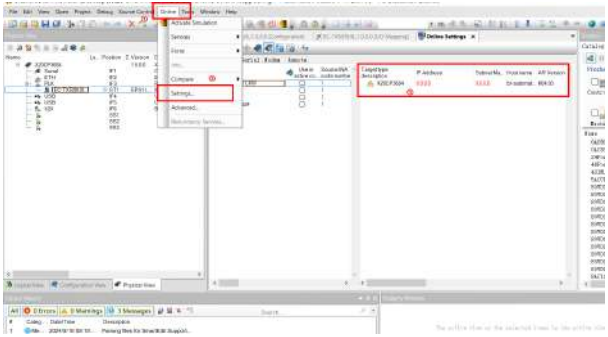


Right click the added **EC-TX509U8_1.0.0.0** and choose **I/O Mapping** to enter the PDO mapping interface, where you can view the completed PDO parameters.

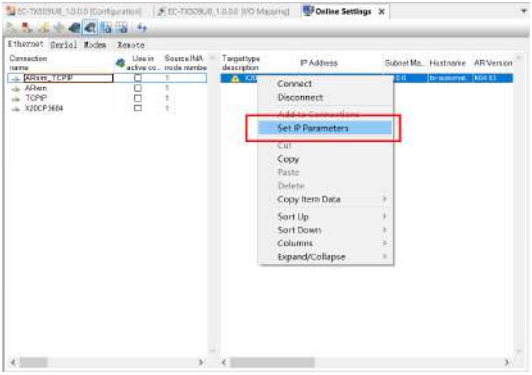


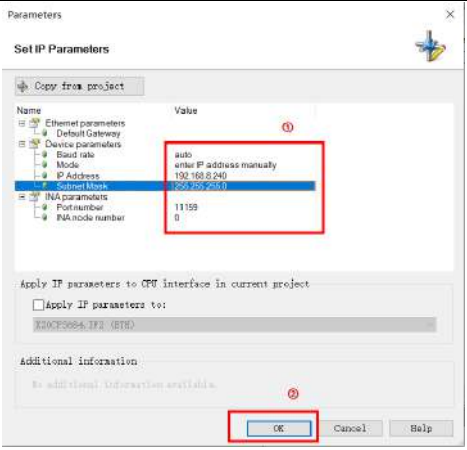
Perform PLC connection and download.

Connect the ETH port of the PLC to the computer's network port. In the Automation Studio menu bar, choose **Online** > **Setting...** to enter the scanning interface, where the detected PLC will be visible.

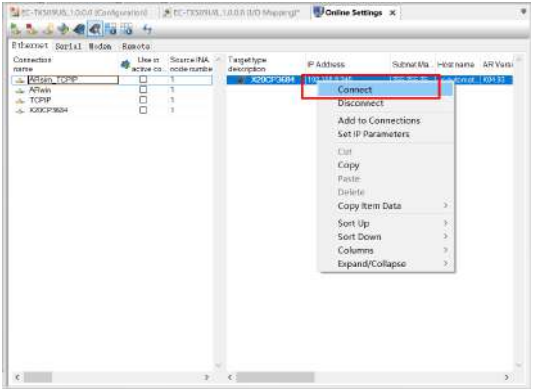


When the scanned PLC IP address is 0, right click the scanned PLC, and choose **Set IP Parameters** to set the IP address. After the IP setting is completed, right click **Connect**. See the following figure.

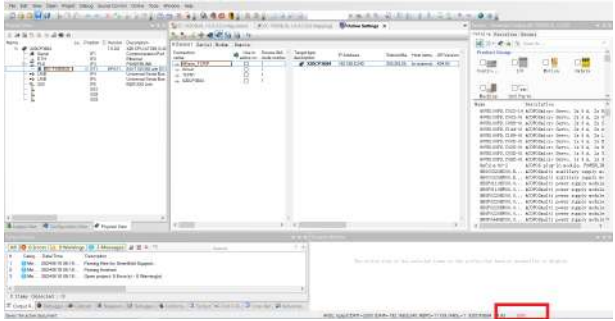




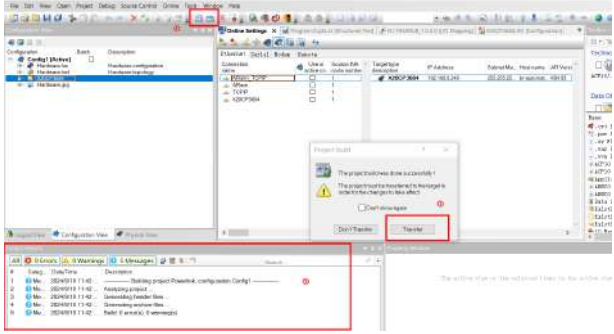
Note: The set IP address should be in the same network segment as the computer.

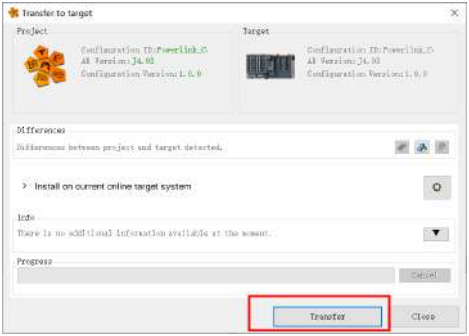


After successful connection, you can see the status of the PLC below.

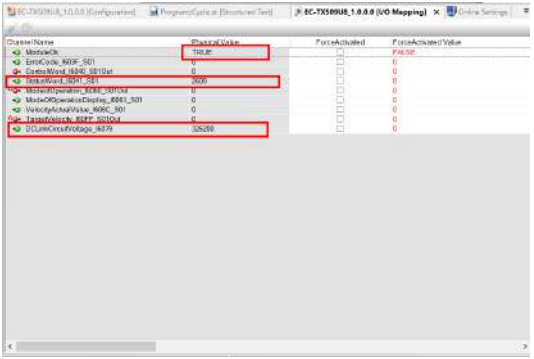


Click **Build** to initiate the compilation. Once the compilation is complete without errors, click **Transfer** on the pop-up window, and then click **Transfer** in the pop-up **Transfer to target** window. Wait for the download to complete.



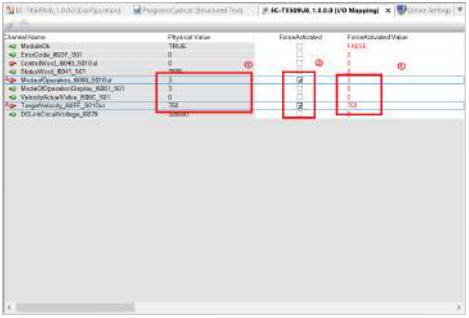


Configure the VFD: set P16.78=1 (station address); P00.01=2; P00.02=3; P00.06=13. Connect the PLC's PLK port to the EC-TX509U8 port. Click **Monitor** to monitor PDO parameters in real time. If **ModuleOk** displays as **True**, it indicates that communication has been successfully established.



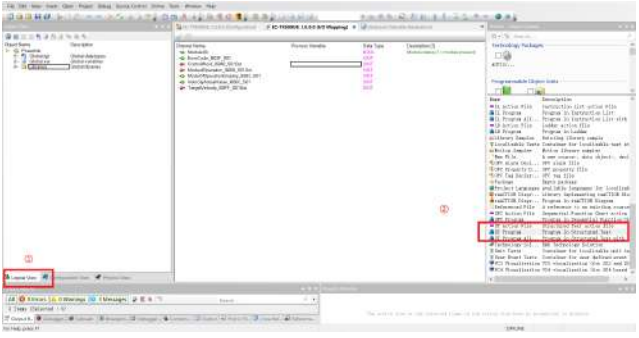
Parameter reference

The PDO parameters can be directly assigned in the **I/O Mapping** interface. Enter the parameters under **Force Activated Value** and select **Force Activated** to successfully write the parameters.

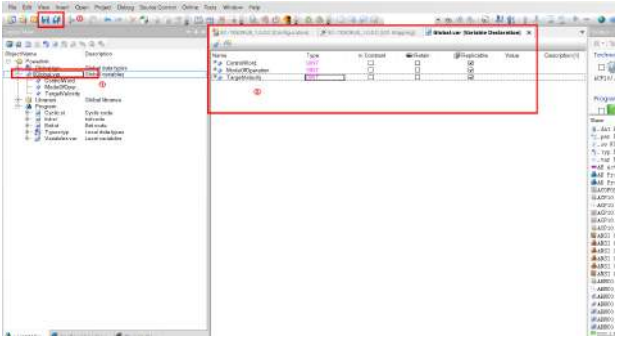


Parameters can also be set through PLC programming.

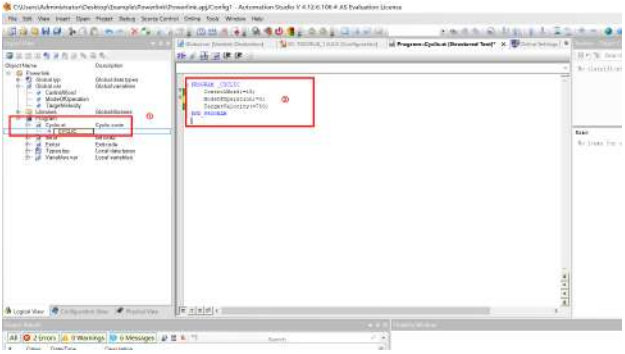
Click **Logical View**, where various programming languages can be selected on the right side to create a program (this example will use ST Program). Simply double click **ST Program** to complete the creation of an ST language program.



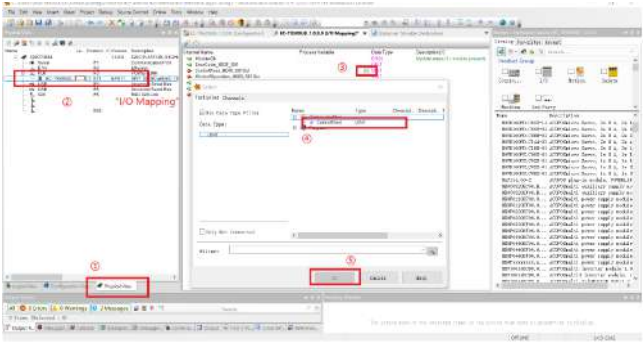
Double click **Global.Var** to define global variables, such as **Control Word**. After completion, click **Save**.



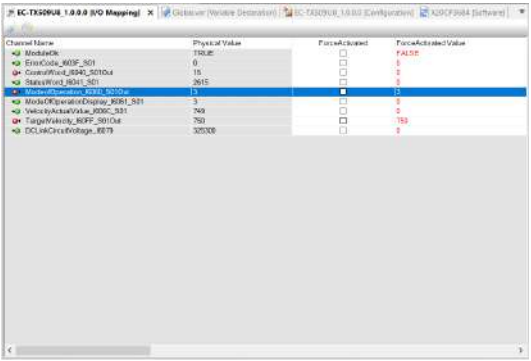
Assign values to the defined variables. Choose **Program** > **Cyclic.st**, double click **Cyclic**, and enter the assignment statements, as shown in the following figure.



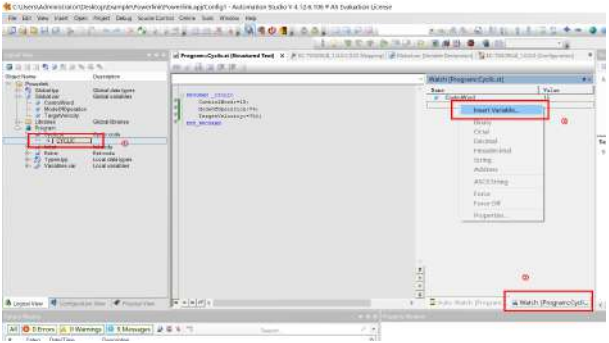
Once the program is written, return to the **I/O Mapping** interface to map the created variables to the relevant parameters.



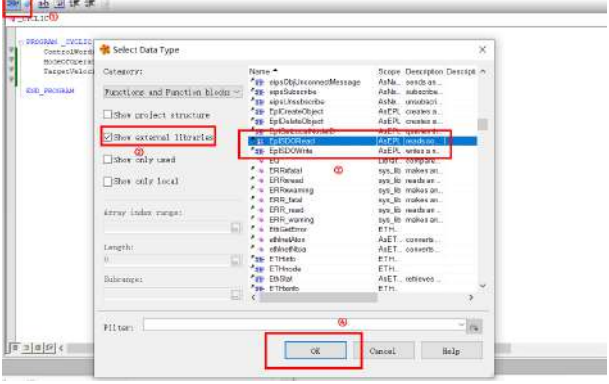
After mapping, if compilation is successful, proceed with the download; then click **Monitor** to confirm the successful setting of parameters.

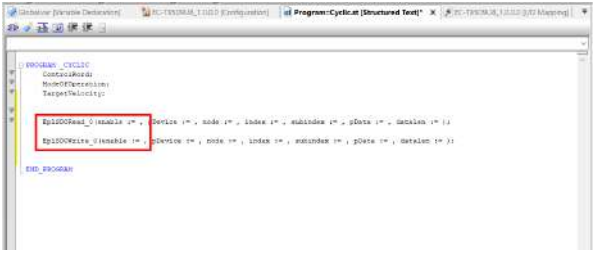


While monitoring, access the programming interface. In the **Watch** window, right click and choose **Insert Variable...** to add the corresponding variables for observation and modification.

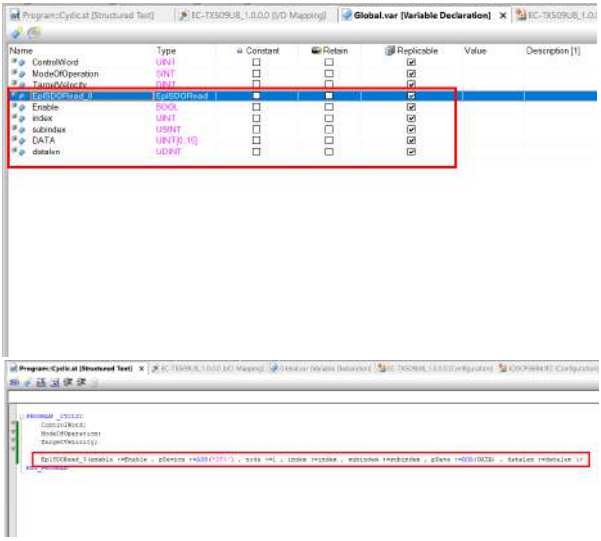


For SDO read/write operations, choose **Program** > **Cyclic.st**, double click **Cyclic** to enter the programming interface. Click the top left corner **Insert Function/Function Block...** to add the **EplSDORead** function block (or **EplSDOWrite**).



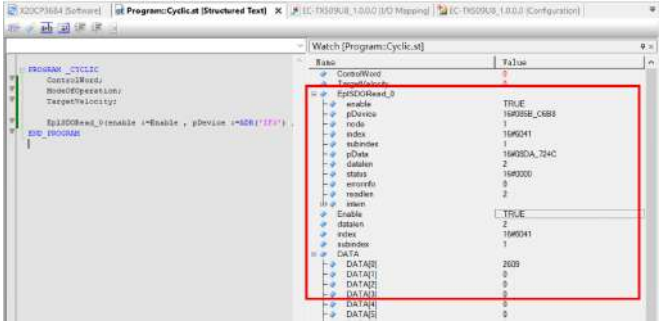


Define the following variables in **Global.Var** (for example, using **EplSDORead**) and configure them within the program.



Upon completing the configuration, if the compilation is error-free, proceed with the download; assign values to parameters. For example, to read the status word (0x6041), set Index=16#6041 and subIndex=1; set datalen=2; Enable=True; the read

value can be observed in Data[0].



Appendix A EtherCAT object dictionary

Index	Subindex	Description	Access permission	Data type	Default
1000h	0	Device type	RO	UINT32	0x00000192
1001h	0	Error register	RO	UINT8	0
1008h	0	Manufacturer device name	RO	String	EC-TX509U8
1009h	0	Manufacturer hardware version	RO	String	Hardware version depended
100Ah	0	Manufacturer software version	RO	String	Software version depended
1018h	**ID object**				
	0	Included max. subindexes	RO	UINT8	4
	1	Supplier ID	RO	UINT32	0x0000072C
	2	Product code	RO	UINT32	0x00009000
	3	Revision No.	RO	UINT32	0x00000001
	4	SN.	RO	UINT32	0x00000001
1600h	**RX PDO1 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x60400010
	2	Second mapping object	RW	UINT32	0x607A0020
	3	Third mapping object	RW	UINT32	0x60FF0020
	4	Fourth mapping object	RW	UINT32	0x60710010
	5	Fifth mapping object	RW	UINT32	0x60720010
	6	Sixth mapping object	RW	UINT32	0x60600008
	7	Seventh mapping object	RW	UINT32	0x60810020
	8	Eighth mapping object	RW	UINT32	0x60B80010

Index	Subindex	Description	Access permission	Data type	Default
1601h	RX PDO2 mapping parameters				
	0	Number of supported mapping objects	RW	UINT8	2
	1	First mapping object	RW	UINT32	0x60400010
	2	Second mapping object	RW	UINT32	0x607A0020
1602h	RX PDO3 mapping parameters				
	0	Number of supported mapping objects	RW	UINT8	2
	1	First mapping object	RW	UINT32	0x60400010
	2	Second mapping object	RW	UINT32	0x607A0020
1603h	RX PDO4 mapping parameters				
	0	Number of supported mapping objects	RW	UINT8	2
	1	First mapping object	RW	UINT32	0x60400010
	2	Second mapping object	RW	UINT32	0x607A0020
1A00h	TX PDO1 mapping parameters				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x60410010
	2	Second mapping object	RW	UINT32	0x60640020
	3	Third mapping object	RW	UINT32	0x606C0020
	4	Fourth mapping object	RW	UINT32	0x60770010
	5	Fifth mapping object	RW	UINT32	0x60F40020
	6	Sixth mapping object	RW	UINT32	0x60610008
	7	Seventh mapping	RW	UINT32	0x60B90010

Index	Subindex	Description	Access permission	Data type	Default
		object			
	8	Eighth mapping object	RW	UINT32	0x60BA0020
1A01h	**TX PDO2 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x60410010
	2	Second mapping object	RW	UINT32	0x60640020
1A02h	**TX PDO3 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x60410010
	2	Second mapping object	RW	UINT32	0x60640020
1A03h	**TX PDO4 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x60410010
	2	Second mapping object	RW	UINT32	0x60640020
1C00h	**Sync Manager communication type**				
	0	Max. subindexes	RO	UINT8	4
	1	SM0 communication type	RO	UINT8	0x01
	2	SM1 communication type	RO	UINT8	0x02
	3	SM2 communication type	RO	UINT8	0x03
	4	SM3 communication type	RO	UINT8	0x04
1C12h	**RxPDO allocating**				
	0	Max. subindexes	RW	UINT8	1

Index	Subindex	Description	Access permission	Data type	Default
	1	RxPDO allocated object index	RW	UINT16	0x1600
1C13h	**TxPDO allocating**				
	0	Max. subindexes	RW	UINT8	1
	1	RxPDO allocated object index	RW	UINT16	0x1A00
1C32h	**Sync Manager synchronization output parameters**				
	0x00	Max. subindexes	RO	UINT8	0x20
	0x01	Synchronization mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switchover time	RO	UINT32	0
	0x04	Supported synchronization type	RO	UINT16	0x4006
	0x05	Min. cycle time	RO	UINT32	0x0003D090
	0x06	Calculation and replication time	RO	UINT32	0
	0x07	Reserved	RW	UINT32	0
	0x08	Obtain cycle time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0
	0x0A	Sync0 time	RW	UINT32	-
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Cyclic timeout counter	RO	UINT32	0
	0x0D	Counter of too short switching	RO	UINT32	0
	0x20	Synchronization error	RO	UINT8	0
1C33h	**Sync Manager synchronization input parameters**				
	0x00	Max. subindexes	RO	UINT8	0x20
	0x01	Synchronization mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switchover time	RO	UINT32	0
	0x04	Supported	RO	UINT16	0x4006

Index	Subindex	Description	Access permission	Data type	Default
		synchronization type			
	0x05	Min. cycle time	RO	UINT32	0x0003D090
	0x06	Calculation and replication time	RO	UINT32	0
	0x07	Reserved	RW	UINT32	0
	0x08	Obtain cycle time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0
	0x0A	Sync0 time	RW	UINT32	-
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Cyclic timeout counter	RO	UINT32	0
	0x0D	Counter of too short switching	RO	UINT32	0
	0x20	Synchronization error	RO	UINT8	0
2000h	0x00–0x13	Function code	RW	UINT16	-
2001h	0x00–0x23	Function code	RW	UINT16	-
2002h	0x00–0x21	Function code	RW	UINT16	-
2003h	0x00–0x42	Function code	RW	UINT16	-
2004h	0x00–0x3C	Function code	RW	UINT16	-
2005h	0x00–0x35	Function code	RW	UINT16	-
2006h	0x00–0x23	Function code	RW	UINT16	-
2007h	0x00–0x56	Function code	RW	UINT16	-
2008h	0x00–0x84	Function code	RW	UINT16	-
2009h	0x00–0x1D	Function code	RW	UINT16	-
200Ah	0x00–0x20	Function code	RW	UINT16	-
200Bh	0x00–0x40	Function code	RW	UINT16	-
200Ch	0x00–0x21	Function code	RW	UINT16	-
200Dh	0x00–0x14	Function code	RW	UINT16	-
200Eh	0x00–0x47	Function code	RW	UINT16	-
200Fh	0x00–0x46	Function code	RW	UINT16	-
2010h	0x00–0x55	Function code	RW	UINT16	-
2011h	0x00–0x40	Function code	RW	UINT16	-
2012h	0x00–0x2D	Function code	RW	UINT16	-
2013h	0x00–0x28	Function code	RW	UINT16	-

Index	Subindex	Description	Access permission	Data type	Default
2014h	0x00–0x28	Function code	RW	UINT16	-
2015h	0x00–0x22	Function code	RW	UINT16	-
2016h	0x00–0x19	Function code	RW	UINT16	-
2017h	0x00–0x14	Function code	RW	UINT16	-
2018h	0x00–0x28	Function code	RW	UINT16	-
2019h	0x00–0x21	Function code	RW	UINT16	-
201Ah	0x00–0x35	Function code	RW	UINT16	-
201Bh	0x00–0x1E	Function code	RW	UINT16	-
201Ch	0x00–0x1E	Function code	RW	UINT16	-
603Fh	0	Fault code	RO	UINT16	0
6040h	0	Control word	RW	UINT16	0
6041h	0	Status word	RO	UINT16	0
6043h	0	Output speed	RO	INT16	0
6044h	0	Feedback speed	RO	INT16	0
6046h	**Speed range**				
	1	Min. value	RO	UINT32	0
	2	Max. value	RO	UINT32	0
6048h	**Acceleration**				
	1	Acceleration increment	RO	UINT32	0
	2	Acceleration time increment	RO	UINT16	0
6049h	**Deceleration**				
	1	Deceleration increment	RO	UINT32	0
	2	Deceleration time increment	RO	UINT16	0
604Ah	**Quick stop**				
	1	Fast stop speed increment	RW	UINT32	0
	2	Fast stop time increment	RW	UINT16	0
604Ch	**Speed gear ratio**				
	1	Numerator of speed gear ratio	RW	INT32	1
	2	Denominator of	RW	INT32	1

Index	Subindex	Description	Access permission	Data type	Default
		speed gear ratio			
6060h	0	Operation mode selection	RW	UINT16	0
6061h	0	Operation mode display	RO	UINT16	0
6062h	0	Position command	RO	DINT32	0
6063h	0	Position feedback	RO	DINT32	0
6064h	0	Position feedback	RO	DINT32	0
6065h	0	Position deviation range	RW	UDINT32	0
6066h	0	Too-large position deviation timeout	RW	UINT16	0
6067h	0	Position pulse range	RW	UDINT32	0
606Ch	0	Actual speed	RW	DINT32	0
6071h	0	Target torque	RW	INT16	0
6072h	0	Max. torque	RW	UINT16	0
6077h	0	Actual torque value	RO	INT16	0
6078h	0	Actual current value	RO	INT16	0
6079h	0	Bus voltage	RO	UDINT32	0
607Ah	0	Target position	RW	INT16	0
6081h	0	Speed in industrial regulations	RW	UDINT32	0
6083h	0	ACC in industrial regulations	RW	UDINT32	0
6084h	0	DEC in industrial regulations	RW	UDINT32	0
6087h	0	Torque ramp	RW	UDINT32	0
6091h	**Gear ratio**				
	0	Number of subindexes	RW	UINT8	2
	1	Motor resolution	RW	UINT32	0x00000001
	2	Bearing axle resolution	RW	UINT32	0x00000001
6098h	0	Homing method	RW	INT16	0
6099h	**Homing speed**				
	0	Reserved	RW	UINT32	0

Index	Subindex	Description	Access permission	Data type	Default
60B0h	0	Position offset	RW	INT32	0
60B1h	0	Speed offset	RW	INT32	0
60B2h	0	Torque offset	RW	INT16	0
60B8h	0	Probe control	RW	UINT16	0
60B9h	0	Probe status	RO	UINT16	0
60BAh	0	Probe position rising edge	RO	INT32	0
60BBh	0	Probe position falling edge	RO	INT32	0
60E0h	0	Forward torque limit	RW	UINT16	0
60E1h	0	Reverse torque limit	RW	UINT16	0
60F4h	0	Position deviation	RO	INT32	0
60FDh	0	Digital input	RO	UINT32	0
60FEh	0	Digital output	RO	INT32	0
60FFh	0	Target speed	RW	INT32	0
6502h	0	Drive mode	RO	UINT32	0x000003A5

Appendix B PowerLink object dictionary

Index	Subindex	Description	Access permission	Data type	Default
1000h	0	Device type	RO	UINT32	0x00010192
1001h	0	Error register	RO	UINT8	0
1006h	0	NMT cycle length	RW	UINT32	3000
1008h	0	Manufacturer device name	RO	String	EC-TX509U8
1009h	0	Manufacturer hardware version	RO	String	Hardware version depended
100Ah	0	Manufacturer software version	RO	String	Software version depended
1018h	**ID object**				
	0	Included max. subindexes	RO	UINT8	4
	1	Supplier ID	RO	UINT32	0x000004D8
	2	Product code	RO	UINT32	0x00009252
	3	Revision No.	RO	UINT32	0x00000001
	4	SN.	RO	UINT32	0x00000001
1020h	**CFM check configuration object**				
	0	Included max. subindexes	RO	UINT8	2
	1	Configuration data	RW	UINT32	0x00000000
	2	Configuration time	RW	UINT32	0x00000000
1030h	**NMT interface record object**				
	0	Number of supported mapping objects	RO	UINT8	9
	1	Interface index	RW	UINT16	1
	2	Interface description	RW	String	Interface 1
	3	Interface type	RW	UINT8	0x06
	4	Max. transmission bytes of interface	RW	UINT16	1518
	5	PHY physical address	RW	String	-
	6	Interface name	RW	String	Interface 1
	7	Open state	RW	UINT8	1
	8	Management state	RW	UINT8	1
	9	Valid	RW	Boolean	true

Index	Subindex	Description	Access permission	Data type	Default
1300h	0	SDO timeout time	RW	UINT32	150000
1400h	**PDO RX communication configuration object**				
	0	Included max. subindexes	RO	UINT8	2
	1	Node ID	RW	UINT8	0
	2	Mapped version	RW	UINT8	0
1600h	**RX PDO1 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x00000000
	2	Second mapping object	RW	UINT32	0x00000000
	3	Third mapping object	RW	UINT32	0x00000000
	4	Fourth mapping object	RW	UINT32	0x00000000
	5	Fifth mapping object	RW	UINT32	0x00000000
	6	Sixth mapping object	RW	UINT32	0x00000000
	7	Seventh mapping object	RW	UINT32	0x00000000
	8	Eighth mapping object	RW	UINT32	0x00000000
1800h	**PDO TX communication configuration object**				
	0	Number of supported mapping objects	RW	UINT8	2
	1	Node ID	RW	UINT8	254
	2	Mapped version	RW	UINT8	0
1A00h	**TX PDO1 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapping object	RW	UINT32	0x00000000
	2	Second mapping object	RW	UINT32	0x00000000
	3	Third mapping object	RW	UINT32	0x00000000
	4	Fourth mapping object	RW	UINT32	0x00000000
	5	Fifth mapping object	RW	UINT32	0x00000000
	6	Sixth mapping object	RW	UINT32	0x00000000
	7	Seventh mapping	RW	UINT32	0x00000000

Index	Subindex	Description	Access permission	Data type	Default
		object			
	8	Eighth mapping object	RW	UINT32	0x00000000
1C0Bh	**DLL SoC node missing object**				
	0	Number of supported mapping objects	RW	UINT8	3
	1	Number of communication activations	RW	UINT32	0
	2	Number of threshold exceeding times	RW	UINT32	0
	3	Threshold size	RW	UINT32	15
1C0Dh	**DLL PReq node missing object**				
	0	Number of supported mapping objects	RW	UINT8	3
	1	Number of communication activations	RW	UINT32	0
	2	Number of threshold exceeding times	RW	UINT32	0
	3	Threshold size	RW	UINT32	15
1C0Fh	**DLL CRC error object**				
	0	Number of supported mapping objects	RW	UINT8	3
	1	Number of communication activations	RW	UINT32	0
	2	Number of threshold exceeding times	RW	UINT32	0
	3	Threshold size	RW	UINT32	15
1C14h	0	DLL SoC node missing coefficient	RW	UINT32	100000
1F82h	0	NMT feature value	RO	UINT32	0x45
1F83h	0	NMT EPL version	RO	UINT8	0x20
1F8Ch	0	Present NMT state	RO	UINT8	-
1F93h	**EPL node ID object**				
	0	Number of supported mapping objects	RW	UINT8	2

Index	Subindex	Description	Access permission	Data type	Default
	1	Node ID	RO	UINT8	-
	2	Node hardware ID	RO	Boolean	-
1F98h	**NMT cycle time recording object**				
	0	Number of supported mapping objects	RO	UINT8	9
	1	Max. payload of Tx	RO	UINT16	100
	2	Max. payload of Rx	RO	UINT16	100
	3	PRes max. delay	RO	UINT32	100000
	4	PRes valid payload upper limit	RW	UINT16	36
	5	PRes valid payload lower limit	RW	UINT16	36
	6	ASnd max. delay	RO	UINT32	150000
	7	Multi-cycle counting	RW	UINT8	0
	8	ASnd max. bytes	RW	UINT16	1518
	9	Polarity	RW	UINT16	2
1F99h	0	NMT basic communication timeout time	RW	UINT8	5000000
1F9Eh	0	NMT reset command	RW	UINT32	0xFF
4000h	0x00–0x13	Function code	RW	UINT16	-
4001h	0x00–0x23	Function code	RW	UINT16	-
4002h	0x00–0x21	Function code	RW	UINT16	-
4003h	0x00–0x42	Function code	RW	UINT16	-
4004h	0x00–0x3C	Function code	RW	UINT16	-
4005h	0x00–0x35	Function code	RW	UINT16	-
4006h	0x00–0x23	Function code	RW	UINT16	-
4007h	0x00–0x56	Function code	RW	UINT16	-
4008h	0x00–0x84	Function code	RW	UINT16	-
4009h	0x00–0x1D	Function code	RW	UINT16	-
400Ah	0x00–0x20	Function code	RW	UINT16	-
400Bh	0x00–0x40	Function code	RW	UINT16	-
400Ch	0x00–0x21	Function code	RW	UINT16	-
400Dh	0x00–0x14	Function code	RW	UINT16	-
400Eh	0x00–0x47	Function code	RW	UINT16	-
400Fh	0x00–0x46	Function code	RW	UINT16	-
4010h	0x00–0x55	Function code	RW	UINT16	-
4011h	0x00–0x40	Function code	RW	UINT16	-

Index	Subindex	Description	Access permission	Data type	Default
4012h	0x00–0x2D	Function code	RW	UINT16	-
4013h	0x00–0x28	Function code	RW	UINT16	-
4014h	0x00–0x28	Function code	RW	UINT16	-
4015h	0x00–0x22	Function code	RW	UINT16	-
4016h	0x00–0x19	Function code	RW	UINT16	-
4017h	0x00–0x14	Function code	RW	UINT16	-
4018h	0x00–0x28	Function code	RW	UINT16	-
4019h	0x00–0x21	Function code	RW	UINT16	-
401Ah	0x00–0x35	Function code	RW	UINT16	-
401Bh	0x00–0x1E	Function code	RW	UINT16	-
401Ch	0x00–0x1E	Function code	RW	UINT16	-
603Fh	0	Fault code	RO	UINT16	0
6040h	0	Control word	RW	UINT16	0
6041h	0	Status word	RO	UINT16	0
6043h	0	Output speed	RO	INT16	0
6044h	0	Feedback speed	RO	INT16	0
6046h	**Speed range**				
	1	Min. value	RO	UINT32	0
	2	Max. value	RO	UINT32	0
6048h	**Acceleration**				
	1	Acceleration increment	RO	UINT32	0
	2	Acceleration time increment	RO	UINT16	0
6049h	**Deceleration**				
	1	Deceleration increment	RO	UINT32	0
	2	Deceleration time increment	RO	UINT16	0
604Ah	**Quick stop**				
	1	Fast stop speed increment	RW	UINT32	0
	2	Fast stop time increment	RW	UINT16	0
604Ch	**Speed gear ratio**				
	1	Numerator of speed gear ratio	RW	INT32	1
	2	Denominator of	RW	INT32	1

Index	Subindex	Description	Access permission	Data type	Default
		speed gear ratio			
6060h	0	Operation mode selection	RW	UINT16	0
6061h	0	Operation mode display	RO	UINT16	0
6062h	0	Position command	RO	DINT32	0
6063h	0	Position feedback	RO	DINT32	0
6064h	0	Position feedback	RO	DINT32	0
6065h	0	Position deviation range	RW	UDINT32	0
6066h	0	Too-large position deviation timeout	RW	UINT16	0
6067h	0	Position pulse range	RW	UDINT32	0
606Ch	0	Actual speed	RW	DINT32	0
6071h	0	Target torque	RW	INT16	0
6072h	0	Max. torque	RW	UINT16	0
6077h	0	Actual torque value	RO	INT16	0
6078h	0	Actual current value	RO	INT16	0
6079h	0	Bus voltage	RO	UDINT32	0
607Ah	0	Target position	RW	INT16	0
6081h	0	Speed in industrial regulations	RW	UDINT32	0
6083h	0	ACC in industrial regulations	RW	UDINT32	0
6084h	0	DEC in industrial regulations	RW	UDINT32	0
6087h	0	Torque ramp	RW	UDINT32	0
6091h	**Gear ratio**				
	0	Number of subindexes	RW	UINT8	2
	1	Motor resolution	RW	UINT32	0x00000001
	2	Bearing axle resolution	RW	UINT32	0x00000001
6098h	0	Homing method	RW	INT16	0
6099h	**Homing speed**				
	0	Reserved	RW	UINT32	0
60B0h	0	Reserved	RW	INT32	0
60B1h	0	Reserved	RW	INT32	0

Index	Subindex	Description	Access permission	Data type	Default
60B2h	0	Reserved	RW	INT16	0
60B8h	0	Probe control	RW	UINT16	0
60B9h	0	Probe status	RO	UINT16	0
60BAh	0	Probe position rising edge	RO	INT32	0
60BBh	0	Probe position falling edge	RO	INT32	0
60E0h	0	Forward torque limit	RW	UINT16	0
60E1h	0	Reverse torque limit	RW	UINT16	0
60F4h	0	Position deviation	RO	INT32	0
60FDh	0	Digital input	RO	UINT32	0
60FEh	0	Digital output	RO	INT32	0
60FFh	0	Target speed	RW	INT32	0
6502h	0	Drive mode	RO	UINT32	0x000003A5

Appendix C Related function codes

Function code	Name	Parameter description	Setting range	Default
P00.01	Channel of running commands	0: Keypad 1: Terminal 2: Communication	0–2	0
P00.02	Communication mode of running commands	0: Modbus/Modbus TCP communication 1: PROFIBUS/CANopen/DeviceNet communication 2: Ethernet communication 3: EtherCAT/PROFINET/EtherNet IP/PowerLink communication 4: Programmable card 5: Wireless communication card 6: USB (reserved) **Note:** The options 1, 2, 3, 4, and 5 are add-on functions and are available only when corresponding expansion cards are configured.	0–6	0
P00.06	Setting channel of A frequency command	0: Keypad digital 1: AI1 2: AI2 3: AI3 4: High-speed pulse HDIA 5: Simple PLC program 6: Multi-step speed running 7: PID control 8: Modbus/Modbus TCP communication 9: PROFIBUS/CANopen/DeviceNet communication 10: Ethernet communication 11: High-speed pulse HDIB 12: Pulse train AB 13: EtherCAT/PROFINET/EtherNet IP/PowerLink communication	0–15	0
P00.07	Setting channel of B frequency command		0–15	15

Function code	Name	Parameter description	Setting range	Default
		14: Programmable card 15: Reserved		
P00.11	ACC time 1	0.0–3600.0s	0.0–3600.0	Model depended
P00.12	DEC time 1	0.0–3600.0s	0.0–3600.0	Model depended
P03.11	Torque setting method selection	0: Keypad (P03.12) 1: Keypad (P03.12) 2: AI1 3: AI2 4: AI3 5: Pulse frequency HDIA 6: Multi-step torque 7: Modbus/Modbus TCP communication 8: PROFIBUS/CANopen/DeviceNet communication 9: Ethernet communication 10: Pulse frequency HDIB 11: EtherCAT/PROFINET/EtherNet IP/PowerLink communication 12: Programmable card	0–12	0
P03.14	Setting source of forward rotation frequency upper limit in torque control	0: Keypad (P03.16) 1: AI1 (100% corresponding to the max. frequency) 2: AI2 (same as the above) 3: AI3 (same as the above) 4: Pulse frequency HDIA (same as the above) 5: Multi-step setting (same as the above) 6: Modbus/Modbus TCP communication (same as the above) 7: PROFIBUS/CANopen/DeviceNet communication (same as the above)	0–12	0

Function code	Name	Parameter description	Setting range	Default
		8: Ethernet communication (same as the above) 9: Pulse frequency HDIB 10: EtherCAT/PROFINET/EtherNet IP/PowerLink communication (same as the above) 11: Programmable card (Same as the above) 12: Reserved		
P03.15	Setting source of reverse rotation upper-limit frequency in torque control	0: Keypad (P03.17) 1: AI1 (100% corresponding to the max. frequency) 2: AI2 (same as the above) 3: AI3 (same as the above) 4: Pulse frequency HDIA (same as the above) 5: Multi-step setting (same as the above) 6: Modbus/Modbus TCP communication (same as the above) 7: PROFIBUS/CANopen/DeviceNet communication (same as the above) 8: Ethernet communication (same as the above) 9: Pulse frequency HDIB (same as the above) 10: EtherCAT/PROFINET/EtherNet IP/PowerLink communication (same as the above) 11: Programmable card (Same as the above) 12: Reserved	0–12	0
P03.18	Setting source of electromotive torque upper limit	0: Keypad (P03.20) 1: AI1 2: AI2	0–11	0

Function code	Name	Parameter description	Setting range	Default
		3: AI3 4: Pulse frequency HDIA 5: Modbus/Modbus TCP communication 6: PROFIBUS/CANopen/DeviceNet communication 7: Ethernet communication 8: Pulse frequency HDIB 9: EtherCAT/PROFINET/EtherNet IP/PowerLink communication 10: Programmable card 11: Reserved		
P03.19	Setting source of braking torque upper limit	0: Keypad (P03.21) 1: AI1 2: AI2 3: AI3 4: Pulse frequency HDIA 5: Modbus/Modbus TCP communication 6: PROFIBUS/CANopen/DeviceNet communication 7: Ethernet communication 8: Pulse frequency HDIB 9: EtherCAT/PROFINET/EtherNet IP/PowerLink communication 10: Programmable card 11: Reserved	0–11	0
P04.27	Voltage setting channel selection	0: Keypad (The output voltage is determined by P04.28.) 1: AI1 2: AI2 3: AI3 4: HDIA 5: Multi-step speed running (The setting is determined by related parameters in group P10.) 6: PID	0–13	0

Function code	Name	Parameter description	Setting range	Default
		7: Modbus/Modbus TCP communication 8: PROFIBUS/CANopen/DeviceNet communication 9: Ethernet communication 10: HDIB 11: PROFINET/EtherNet IP communication 12: Programmable card 13: Reserved		
P06.01	Y1 output	0: Invalid 1: Running 2: Running forward 3: Running reversely 4: Jogging 5: VFD in fault 6: Frequency level detection FDT1 7: Frequency level detection FDT2 8: Frequency reached 9: Running in zero speed 10: Frequency upper limit reached 11: Frequency lower limit reached 12: Ready for running 13: Pre-exciting 14: Overload pre-alarm 15: Underload pre-alarm 16: Simple PLC stage completed 17: Simple PLC cycle completed 18: Set counting value reached 19: Designated counting value reached 20: External fault is valid 21: Reserved 22: Running time reached 23: Modbus/ Modbus TCP communication virtual terminal	0–63	0
P06.02	HDO output		0–63	0
P06.03	RO1 output		0–63	1
P06.04	RO2 output		0–63	5

Function code	Name	Parameter description	Setting range	Default
		output 24: PROFIBUS/CANopen/DeviceNet communication virtual terminal output 25: Ethernet communication virtual terminal output 26: DC bus voltage established 27: Z pulse output 28: Superposing pulses 29: STO action 30: Positioning completed 31: Spindle zeroing completed 32: Spindle indexing completed 33: Reserved 34: EtherCAT/PROFINET/EtherNet IP/PowerLink communication virtual terminal output 35: Reserved 36: Speed/position control switchover completed 37: Any frequency reached 38–40: Reserved 41: Y1 from the programmable card 42: Y2 from the programmable card 43: HDO from the programmable card 44: RO1 from the programmable card 45: RO2 from the programmable card 46: RO3 from the programmable card 47: RO4 from the programmable card		

Function code	Name	Parameter description	Setting range	Default
		48: IO card detected PT100 OH pre-alarm 49: IO card detected PT1000 OH pre-alarm 50: AIAO detected OT pre-alarm 51: Stopped or running in zero speed 52: Tension control disconnection 53: Specified roll diameter reached 54: Roll diameter of stop reached 55: Length reached 56: Fire mode enabled 57–63: Reserved		
P06.14	AO1 output	0: Running frequency	0–63	0
P06.16	HDO high-speed pulse output	1: Set frequency 2: Ramp reference frequency 3: Rotational speed (100% corresponds to the speed corresponding to the max. output frequency) 4: Output current (100% corresponds to twice the VFD rated current) 5: Output current (100% corresponds to twice the motor rated current) 6: Output voltage (100% corresponds to 1.5 times the VFD rated voltage) 7: Output power (100% corresponds to twice the motor rated power) 8: Set torque (100% corresponds to twice the motor rated torque) 9: Output torque (Absolute value, 100% corresponds to twice the	0–63	0

Function code	Name	Parameter description	Setting range	Default
		motor rated torque) 10: AI1 input 11: AI2 input 12: AI3 input 13: HDIA input value 14: Value 1 set through Modbus/Modbus TCP communication 15: Value 2 set through Modbus/Modbus TCP communication 16: Value 1 set through PROFIBUS/CANopen/DeviceNet communication 17: Value 2 set through PROFIBUS/CANopen/DeviceNet communication 18: Value 1 set through Ethernet communication 19: Value 2 set through Ethernet communication 20: HDIB input value 21: Value 1 set through EtherCAT/PROFINET/EtherNet IP communication 22: Torque current (100% corresponds to triple the motor rated current) 23: Exciting current (100% corresponds to triple the motor rated current) 24: Set frequency (bipolar) 25: Ramp reference frequency (bipolar) 26: Rotational speed of running (bipolar) 27: Value set through		

Function code	Name	Parameter description	Setting range	Default
		EtherCAT/PROFINET/EtherNet IP communication 28: AO1 from the programmable card 29: AO2 from the programmable card 30: Rotational speed of running (100% corresponds to the speed at twice the motor rated frequency) 31: Output torque (Actual value, 100% corresponds to twice the motor rated torque) 32: AIAO detected temperature output 33: Set tension output 34–63: Reserved		
P07.27	Present fault type	0: No fault 1: Inverter unit U-phase protection (OUt1) 2: Inverter unit V-phase protection (OUt2) 3: Inverter unit W-phase protection (OUt3) 4: Overcurrent during ACC (OC1) 5: Overcurrent during DEC (OC2) 6: Overcurrent during constant speed running (OC3) 7: Overvoltage during ACC (OV1) 8: Overvoltage during DEC (OV2) 9: Overvoltage during constant speed running (OV3) 10: Bus undervoltage fault (UV) 11: Motor overload (OL1) 12: VFD overload (OL2) 13: Phase loss on input side (SPI)	/	/
P07.28	Last fault type	0: No fault		
P07.29	2nd-last fault type			
P07.30	3rd-last fault type			
P07.31	4th-last fault type			
P07.32	5th-last fault type			

Function code	Name	Parameter description	Setting range	Default
		14: Phase loss on output side (SPO) 15: Rectifier module overheating (OH1) 16: Inverter module overheating (OH2) 17: External fault (EF) 18: Modbus/Modbus TCP communication fault (CE) 19: Current detection fault (ItE) 20: Motor autotuning fault (tE) 21: EEPROM operation error (EEP) 22: PID feedback offline fault (PIDE) 23: Braking unit fault (bCE) 24: Running time reached (END) 25: Electronic overload (OL3) 26: Keypad communication error (PCE) 27: Parameter upload error (UPE) 28: Parameter download error (DNE) 29: PROFIBUS communication fault (E_dP) 30: Ethernet communication fault (E-NET) 31: CANopen communication fault (E-CAN) 32: To-ground short-circuit fault 1 (ETH1) 33: To-ground short-circuit fault 2 (ETH2) 34: Speed deviation fault (dEu) 35: Mal-adjustment fault (STo) 36: Underload fault (LL) 37: Encoder disconnection fault (ENC1O)		

Function code	Name	Parameter description	Setting range	Default
		38: Encoder direction reversal fault (ENC1D) 39: Encoder Z-pulse disconnection fault (ENC1Z) 40: Safe torque off (STO) 41: Exception to safety circuit of channel 1 (STL1) 42: Exception to safety circuit of channel 2 (STL2) 43: Exception in both channels 1 and 2 (STL3) 44: Safety code FLASH CRC fault (CrCE) 45: Programmable card customized fault 1 (P-E1) 46: Programmable card customized fault 2 (P-E2) 47: Programmable card customized fault 3 (P-E3) 48: Programmable card customized fault 4 (P-E4) 49: Programmable card customized fault 5 (P-E5) 50: Programmable card customized fault 6 (P-E6) 51: Programmable card customized fault 7 (P-E7) 52: Programmable card customized fault 8 (P-E8) 53: Programmable card customized fault 9 (P-E9) 54: Programmable card customized fault 10 (P-E10) 55: Duplicate expansion card type (E-Err) 56: Encoder UVW lost (ENCUV) 57: PROFINET communication		

Function code	Name	Parameter description	Setting range	Default
		timeout fault (E-PN) 58: CAN communication timeout fault (SECAN) 59: Motor overtemperature fault (OT) 60: Failure to identify the card in slot 1 (F1-Er) 61: Failure to identify the card in slot 2 (F2-Er) 62: Failure to identify the card in slot 3 (F3-Er) 63: Communication timeout of the card in slot 1 (C1-Er) 64: Communication timeout of the card in slot 2 (C2-Er) 65: Communication timeout of the card in slot 3 (C3-Er) 66: EtherCAT communication timeout fault (E-CAT) 67: BACnet communication timeout fault (E-BAC) 68: DeviceNet communication timeout fault (E-DEV) 69: CAN slave fault in master/slave synchronization (S-Err) 70: EC PT100 detected OH (OtE1) 71: EC PT1000 detected OH (OtE2) 72: EtherNet IP communication timeout fault (E-EIP) 73: No upgrade bootload (E-PAO) 74: AI1 disconnection (E-AI1) 75: AI2 disconnection (E-AI2) 76: AI3 disconnection (E-AI3) 77–80: Reserved 81: Tension communication timeout fault (E-tC)		

Function code	Name	Parameter description	Setting range	Default
		82: PowerLink communication timeout fault (E-EPL)		
P08.31	Channel for switching between motor 1 and motor 2	Ones place: Switchover channel 0: Switch over through terminals 1: Modbus/Modbus TCP communication 2: PROFIBUS/CANopen/DeviceNet communication 3: Ethernet communication 4: EtherCAT/PROFINET/EtherNet IP communication Tens place: indicates whether to enable switchover during running 0: Disable 1: Enable	0x00–0x14	0x00
P09.00	PID reference source selection	0: Setting through P09.01 1: AI1 2: AI2 3: AI3 4: High-speed pulse HDIA 5: Multi-step running 6: Modbus/Modbus TCP communication 7: PROFIBUS/CANopen/DeviceNet communication 8: Ethernet communication 9: High-speed pulse HDIB 10: PROFINET/EtherNet IP communication 11: Programmable card 12: Reserved	0–12	0
P09.02	PID feedback source selection	0: AI1 1: AI2 2: AI3 3: High-speed pulse HDIA 4: Modbus/Modbus TCP communication	0–10	0

Function code	Name	Parameter description	Setting range	Default
		5: PROFIBUS/CANopen/DeviceNet communication 6: Ethernet communication 7: High-speed pulse HDIB 8: PROFINET/EtherNet IP communication 9: Programmable card 10: Reserved		
P14.00	Local communication address	1–247	1–247	1
P14.03	Communication response delay	1–200ms	1–200	5ms
P14.05	Transmission fault processing	0: Report an alarm and coast to stop 1: Keep running without reporting an alarm 2: Stop in enabled stop mode without reporting an alarm (applicable only to communication mode) 3: Stop in enabled stop mode without reporting an alarm (applicable to any mode)	1–3	0
P14.06	Communication processing action	Ones place: 0: Respond to write operations 1: Not respond to write operations Tens place: 0: Communication user password protection is invalid. 1: Communication user password protection is valid. Hundreds place: 0: User-defined addresses specified by P14.07 and P14.08 are invalid.	0x000–0x111	0x000

Function code	Name	Parameter description	Setting range	Default
		1: User-defined addresses specified by P14.07 and P14.08 are valid.		
P14.09	Modbus TCP communication timeout time	0.0 (invalid)–60.0s	0.0–60.0	0.0s
P15.43	Communication control word expression format	0: Decimal format 1: Binary format	0–1	0
P16.00	Expansion card protocol selection	0: PROFINET 1: EtherCAT 2: PowerLink 3: EtherNet IP 4: Modbus TCP 5: EtherNet UDP 6: PROFINET+EtherNet UDP 7: EtherCAT+EtherNet UDP 8–15: Reserved	0–15	0
P16.14	Ethernet card monitoring variable address 1	0x0000–0xFFFF	0x0000–0xFFFF	0x0000
P16.15	Ethernet card monitoring variable address 2	0x0000–0xFFFF	0x0000–0xFFFF	0x0000
P16.16	Ethernet card monitoring variable address 3	0x0000–0xFFFF	0x0000–0xFFFF	0x0000
P16.17	Ethernet card monitoring variable address 4	0x0000–0xFFFF	0x0000–0xFFFF	0x0000
P16.24	Time to identify expansion card in card slot 1	0.0–600.0s The value 0.0 indicates that identification fault will not be detected.	0.0–600.00	0.0s
P16.25	Time to identify expansion card in card slot 2	0.0–600.0s The value 0.0 indicates offline fault will not be detected.	0.0–600.00	0.0s

Function code	Name	Parameter description	Setting range	Default
P16.26	Time to identify expansion card in card slot 3	0.0–600.0s The value 0.0 indicates offline fault will not be detected.	0.0–600.0	0.0s
P16.27	Communication timeout time of expansion card in card slot 1	0.0–600.0s The value 0.0 indicates offline fault will not be detected.	0.0–600.0	0.0s
P16.28	Communication timeout time of expansion card in card slot 2	0.0–600.0s The value 0.0 indicates offline fault will not be detected.	0.0–600.0	0.0s
P16.29	Communication timeout time of expansion card in card slot 3	0.0–600.0s The value 0.0 indicates offline fault will not be detected.	0.0–600.0	0.0s
P16.31	PROFINET communication timeout time	0.0 (invalid)–60.0s	0.0–60.0	5.0s
P16.32	Received PZD2	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01Hz) 2: PID reference (-1000–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (-1000–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the motor rated current) 5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01Hz)	0–31	0
P16.33	Received PZD3		0–31	0
P16.34	Received PZD4		0–31	0
P16.35	Received PZD5		0–31	0
P16.36	Received PZD6		0–31	0
P16.37	Received PZD7		0–31	0
P16.38	Received PZD8		0–31	0
P16.39	Received PZD9		0–31	0
P16.40	Received PZD10		0–31	0
P16.41	Received PZD11		0–31	0
P16.42	Received PZD12		0–31	0

Function code	Name	Parameter description	Setting range	Default
		7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 8: Upper limit of braking torque (0–3000, in which 1000 corresponds to 100% of the motor rated current) 9: Virtual input terminal command (range: 0x000–0x3FF, with bit 9–bit 10 corresponding to S8/S7/S6/S5/HDIB/HDIA/S4/S3/S2/S1 in sequence) 10: Virtual output terminal command (range: 0x00–0x0F, bit3–bit0 corresponding to RO2/RO1/HDO/Y1 in sequence) 11: Voltage setting (special for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage) 12: AO1 output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%) 13: AO2 output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%) 14: High bit of position reference (signed) 15: Low bit of position reference (unsigned) 16: High bit of position feedback (signed) 17: Low bit of position feedback (unsigned)		

Function code	Name	Parameter description	Setting range	Default
		18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) 19: Function parameter mapping (PZD2–PZD12 correspond to P14.49–P14.59) 20–31: Reserved		
P16.43	Sent PZD2	0: Invalid 1: Running frequency (×100, Hz) 2: Set frequency (×100, Hz) 3: Bus voltage (×10, V) 4: Output voltage (×1, V) 5: Output current (×10, A) 6: Actual output torque (×10, %) 7: Actual output power (×10, %) 8: Rotation speed of running (×1, RPM) 9: Linear speed of running (×1, m/s) 10: Ramp reference frequency 11: Fault code 12: AI1 input (×100, V) 13: AI2 input (×100, V) 14: AI3 input (× 100, V) 15: HDIA frequency value (×1000, kHz) 16: Terminal input status 17: Terminal output status 18: PID reference (×10, %) 19: PID feedback (×10, %) 20: Invalid 21: High bit of position reference (signed) 22: Low bit of position reference (unsigned) 23: MSB of position feedback	0–47	0
P16.44	Sent PZD3		0–47	0
P16.45	Sent PZD4		0–47	0
P16.46	Sent PZD5		0–47	0
P16.47	Sent PZD6		0–47	0
P16.48	Sent PZD7		0–47	0
P16.49	Sent PZD8		0–47	0
P16.50	Sent PZD9		0–47	0
P16.51	Sent PZD10		0–47	0
P16.52	Sent PZD11		0–47	0
P16.53	Sent PZD12		0–47	0

Function code	Name	Parameter description	Setting range	Default
		(signed) 24: LSB of position feedback (unsigned) 25: Status word 26: HDIB frequency value (×1000, kHz) 27: PG card pulse feedback count high bit 28: PG card pulse feedback count low bit 29: PG card pulse reference count MSB 30: PG card pulse reference count LSB 31: Function parameter mapping (PZD2–PZD12 correspond to P14.60–P14.70) 32: Status word 3 33–47: Reserved		
P16.54	EtherNet IP communication timeout time	0.0–60.0s	0.0–60.0	5.0s
P16.58	Industrial Ethernet communication card IP address 1	0–255	0–255	192
P16.59	Industrial Ethernet communication card IP address 2	0–255	0–255	168
P16.60	Industrial Ethernet communication card IP address 3	0–255	0–255	0
P16.61	Industrial Ethernet communication card IP address 4	0–255	0–255	20
P16.62	Industrial Ethernet communication card subnet mask 1	0–255	0–255	255

Function code	Name	Parameter description	Setting range	Default
P16.63	Industrial Ethernet communication card subnet mask 2	0–255	0–255	255
P16.64	Industrial Ethernet communication card subnet mask 3	0–255	0–255	255
P16.65	Industrial Ethernet communication card subnet mask 4	0–255	0–255	0
P16.66	Industrial Ethernet communication card gateway 1	0–255	0–255	192
P16.67	Industrial Ethernet communication card gateway 2	0–255	0–255	168
P16.68	Industrial Ethernet communication card gateway 3	0–255	0–255	0
P16.69	Industrial Ethernet communication card gateway 4	0–255	0–255	1
P16.70	Saving EtherCAT written function codes	0: No 1: Yes	0–1	0
P16.72	EtherCAT input unit selection	0: Input rotation speed unit is PRM 1: Input rotation speed unit is plus/s	0–1	0
P16.73	EtherCAT slave node address	0x0000–0xFFFF	0x0000–0XFFFF	0xFFFF
P16.74	EtherCAT-DC synchronization cycle (reserved)	0–1: Reserved 2: 1ms 3: 2ms 4–5: Reserved	0–5	0
P16.75	EtherCAT communication timeout time	0.0–60.0s	0.0–60.0	5.0s

Function code	Name	Parameter description	Setting range	Default
P16.76	EtherCAT supported PLC type	0: Beckhoff 1: AX70 2: OMRON 3: Trio 4: LNC 5–8: Reserved	0–8	0
P16.77	EtherCAT run mode	0: Free-run mode 1: SM mode (synchronized in data input and output) 2: DC mode (synchronized in distributed clocks)	0–2	0
P16.78	PowerLink slave station address	0x00–0xFF	0x00–0xFF	0x01
P16.79	PowerLink communication timeout time	0.0–60.0s	0.0–60.0	5.0s
P19.00	Type of expansion card in slot 1	0: No card 1: PLC card 2: I/O card 3: Incremental PG card	0–50	0
P19.01	Type of expansion card in slot 2		0–50	0
P19.02	Type of expansion card in slot 3	4: Incremental PG card with UVW 5: Ethernet communication card 6: DP communication card 7: Bluetooth card 8: Resolver PG card 9: CANopen communication card 10: WIFI card 11: PROFINET communication card 12: Sine-cosine PG card without CD signals 13: Sine-cosine PG card with CD signals 14: Absolute encoder PG card (reserved) 15: CAN master/slave communication card	0–50	0

Function code	Name	Parameter description	Setting range	Default
		16: Modbus TCP communication card 17: EtherCAT communication card 18: BACnet communication card (reserved) 19: DeviceNet communication card (reserved) 20: PT100/PT1000 temperature detection card 21: EtherNet IP card 22: MECHATROLINK card (reserved) 23–25: Reserved 26: PROFINET+Ethernet communication card 27–35: Reserved 36: All-in-one expansion card—PROFINET communication card 37: Reserved 38: PowerLink communication card 39–40: Reserved 41: All-in-one expansion card—EtherCAT communication card 42: All-in-one expansion card—PowerLink communication card 43: All-in-one expansion card—EtherNet IP communication card 44: All-in-one expansion card—Modbus TCP communication card 45: All-in-one expansion		

Function code	Name	Parameter description	Setting range	Default
		card—Ethernet communication card 46: All-in-one expansion card—PROFINET + EtherNet UDP communication card 47: All-in-one expansion card—EtherCAT + EtherNet UDP communication card 48–50: Reserved		
P21.20	Positioning ACC time	0.00–300.00s	0.00–300.00	3.00s
P21.21	Positioning DEC time	0.00–300.00s	0.00–300.00	3.00s

Your Trusted Industry Automation Solution Provider

Shenzhen INVT Electric Co., Ltd.
Address: INVT Guangming Technology Building, Songbai Road, Matian, Guangming District, Shenzhen, China
INVT Power Electronics (Suzhou) Co., Ltd.
Address: No. 1 Kunlun Mountain Road, Science & Technology Town, Gaoxin District, Suzhou, Jiangsu, China
Website: www.invt.com

INVT mobile website

INVT e-manual

66001-01442

Copyright© INVT.
Manual information may be subject to change without prior notice.

202501 (V1.0)