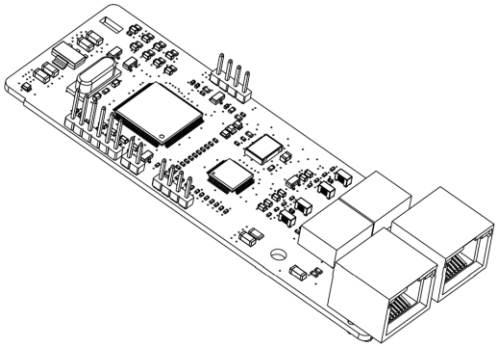
Operation Manual

Ethernet Communication Card



SHENZHEN INVT ELECTRIC CO., LTD.

No.	Change description	Version	Release date
1	First release	V1.0	Octomber 2020
2	…	…	…

Safety precautions

The extension 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.

For any physical injuries or damage to the device caused due to your neglect of the safety precautions described in this manual and the VFD operation manual, our company shall not be held liable.

- You need to open the housing of the VFD when installing or removing the communication card. Therefore, you must disconnect all power supplies of the VFD and ensure that the voltage inside the VFD is safe. For details, see the description in the VFD operation manual. Severe physical injuries or even death may be caused if you do not follow the instructions.
- Store the communication card in a place that is dustproof and dampproof without electric shocks or mechanical pressure.
- The communication card is electrostatic sensitive. Take measurements to prevent electrostatic discharge when performing operations involving it.
- Tighten the screws up when installing the communication card. Ensure that it is firmly fixed and properly grounded.

Terminology, abbreviations, and acronyms

CAN	Controller Area Network
COB	Communication object, a transmitted unit on a CAN network. Communication objects (COBs) carry data and can be transmitted through the whole network. A COB is part of a CAN message frame.
EDS	Electronic data sheet (EDS), 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, status values, and actual values.
PDOn Tx	PDO command transmitted by a slave station to the master station, where n refers to 1, 2, 3, 4.
PDOn Rx	PDO command transmitted by the master station and received by a slave station, where n refers to 1, 2, 3, 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	Indicates that a number with this prefix is a hexadecimal value, for example, 0x10 indicates the decimal value 16.

Contents

1 Product confirmation

Check the following after receiving a communication extension 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 of the communication card from INVT. The file is named communication card model.eds.
- Confirm the environmental requirements for application.

Table 1-1 Environmental requirements

Item	Requirement
Operation temperature	-10–+50°C
Storage temperature	-20–+60°C
Relative humidity	5%–95%
Other weather conditions	No condensation, ice, rain, snow, or hail; solar radiation < 700 W/m^2
Air pressure	70–106 kPa
Vibration and impact	5.9m/s^2 (0.6g) at the sine vibration of 9 Hz to 200 Hz

2 PROFINET communication card

2.1 Overview

1. Thanks for choosing INVT PROFINET communication cards. This manual describes the function specifications, installation, basic operation and settings, and information about the network protocol. To ensure that you install and operate the product properly, read this manual and the communication protocol section in the VFD operation manual carefully before you use the product.
2. This manual only describes how to operate the PROFINET communication card and the related commands but does not provide details about the PROFINET protocol. For more information about the PROFINET protocol, read the related specialized articles or books.
3. This communication card is defined as a PROFINET slave station communication card and is used on a VFD that supports PROFINET communication.
4. The communication card supports the linear network topology and star-shaped network topology.
5. The communication card supports 32 inputs/outputs to read and write process data, read state data, and read and write function parameters of a VFD.

2.2 Features

1. Supported functions

- Supports the PROFINET protocol, and supports PROFINET I/O devices
- Provides two PROFINET I/O ports and supports the 100 M full-duplex operation
- Supports the linear network topology and star-shaped network topology.

2. Supported communication types

- Standard Ethernet channels:

Standard Ethernet channels are non-realtime communication channels that use the TCP/IP protocol, and are mainly used for device parameterization and configuration and to read diagnosis data.

- Real-time (RT) communication channels:

RT channels are optimized channels for real-time communication. They take precedence over TCP (UDP)/IP, which ensures that various stations on a network perform data transmission with high time requirements at a certain interval. The bus period may reach the precision of millisecond. These channels are used to transmit data such as process data and alarm data.

- Isochronous real-time (IRT) communication channels

IRT channels are implemented through the built-in Switch-ASIC IRT chip. IRT communication can further shorten the processing time of the communication stack software, synchronizing data transmission of the program and device. The transmission delay is less than 1 ms, and the jitter is less than 1 μs. The typical application is motion control.

3. Communication ports

Standard RJ45 ports are used in PROFINET communication. The communication card provides two RJ45 ports with no transmission direction defined, and therefore you can insert a cable into the port without regard to its direction. Figure 2-1 shows the ports, and Table 2-1 describes the functions of the ports.

Figure 2-1 Two standard RJ45 ports

Table 2-1 Standard RJ45 port pins

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

4. State indicators

PROFINET communication card provides nine LED indicators to indicate its states. Table 2-2 describes the state indicators.

Table 2-2 State indicators

LED	Color	State	Description
LED1	Green		3.3 V power indicator
LED2 (Bus state indicator)	Red	On	Not connected through a network cable
		Blinking	Connected to the PROFINET controller through a network cable, but no

LED	Color	State	Description
			communication established
		Off	Communication established with the PROFINET controller
LED3 (System fault indicator)	Red	On	PROFINET diagnosis enabled
		Off	PROFINET diagnosis disabled
LED4 (Slave ready indicator)	Green	On	TPS-1 communication stack started
		Blinking	TPS-1 waits for the initialization of MCU
		Off	TPS-1 communication stack not started
LED5 (Maintenance state indicator)	Green		Defined by the manufacturer, depending on the characteristics of the device
LED6/7 (Network port state indicator)	Green	On	PROFINET communication card connected to the PC/PLC through a network cable
		Off	PROFINET communication card not connected to the PC/PLC
LED8/9 (Network port communication indicator)	Green	On	PROFINET communication card communicating with the PC/PLC
		Off	PROFINET communication card not communicating with the PC/PLC

2.3 Electrical wiring

PROFINET communication card provides standard RJ45 ports and supports the linear and star topologies. Figure 2-2 and Figure 2-3 show the electrical wiring diagrams for different topologies.

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

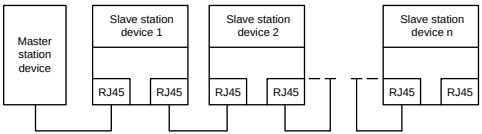


Figure 2-2 Electrical wiring diagram for a linear topology

Note: For the star-shaped network topology, you need to use a PROFINET switch.

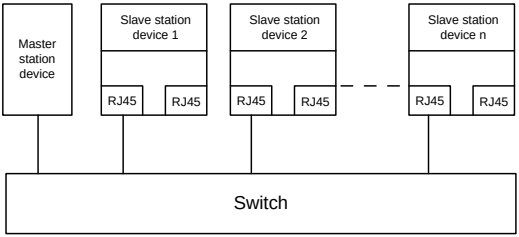


Figure 2-3 Electrical wiring diagram for a star topology

2.4 Communication

2.4.1 Packet format

Table 2-3 describes the structure of an RT frame (non-synchronous).

Table 2-3 Structure of an RT frame

Data header	Ethernet type	VLAN	Ethernet type	Frame identifier	RT user data	Period counter	Data state	Transmission state	FCS
	2 bytes	2 bytes	2 bytes	2 bytes	36–1440 bytes	2 bytes	1 byte	1 byte	4 bytes
	0x8100		0x8892						
	VLAN flag					APDU state			
Data header									
7-byte preamble	1-byte synchronization information	6-byte source MAC address	6-byte destination MAC address						

Table 2-4 describes the structure of the IRT frame (synchronous).

Table 2-4 Structure of an IRT frame

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 PROFINET I/O communication

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

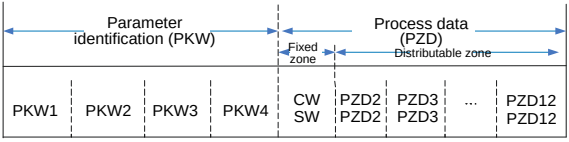


Figure 2-4 Packet structure

By using the 32 inputs/outputs, you can set the reference parameters of the VFD, monitor the status values, transmit control commands, monitor the running state, and read/write the function parameters of the VFD. For specific operations, see the following description.

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—Control word (transmitted from the master to a slave. For description, see

Table 2-5)

SW—Status word (transmitted from a slave to the master. For description, see Table 2-7.)

PZD—Process data (defined by users)

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

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

CWs and SWs

Using CWs is the basic method of the fieldbus system to control VFDs. A CW is transmitted by

the fieldbus master station 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 state information back to the master through an SW.

Reference value: A 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 the communication module as the controller of the VFD device.

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 from the VFD device to the master depends on the set function. For more description, see the related VFD operation manual.

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

Task packet (master station -> VFD)

CW: The first word in a PZD task packet is a VFD CW. You can select the expression method according to P15.43.Table 2.5 and Table 2.6 describe the control words (CWs) of the Goodrive 350 series VFD.Table 2.5 and Table 2.6 describe the control words (CWs) of the Goodrive 350 series VFD.

Table 2-5 Goodrive350 series VFD CWs expressed in decimal format

Bit	Name	Value	Description
0–7	Communication-based control command	1	Forward running
		2	Reverse running
		3	Forward jogging
		4	Reverse jogging
		5	Stop
		6	Coast to stop (emergency stop)
		7	Fault reset
		8	Jogging to stop
		9	Decelerate to stop
8	Enable writing	1	Enable reading and writing (PKW1-PKW4)
9–10	Motor group setting	00	Motor 1
		01	Motor 2
11	Control mode switching	1	Enable torque/speed control switching
		0	Disable switching
12	Reset power consumption to zero	1	Enable
		0	Disable

Bit	Name	Value	Description
13	Pre-excitation	1	Enable
		0	Disable
14	DC braking	1	Enable
		0	Disable
15	Heartbeat reference	1	Enable
		0	Disable

Table 2-6 Goodrive350 series VFD CWs expressed in binary format

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: Disable 1: Enable	3
3	Coast to stop	0: Disable 1: Enable	4
4	Forward jogging	0: Disable 1: Enable	5
5	Reverse jogging	0: Disable 1: Enable	6
6	Jogging to stop	0: Disable 1: Enable	7
7	/	Reserved	
8	Enable reading and writing (PKW1-PKW4)	0: Disable 1: Enable	
9	/	Reserved	
10	Decelerate to stop	0: Disable 1: Enable	0: Top priority
11–15	/	Reserved	

Reference value (REF): The second to twelfth words in a PZD task packet are the main settings. The main frequency settings are provided by the main setting signal source. Table 2-7 describes the settings of Goodrive350 series VFD.

Table 2-7 Settings of Goodrive350 series VFD

Function code	Word	Value range	Default value
P16.32	Received PZD2	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01 Hz)	0
P16.33	Received PZD3	2: PID reference (0–1000, in which 1000 corresponds to 100.0%)	0
P16.34	Received PZD4	3: PID feedback (0–1000, in which 1000 corresponds to 100.0%)	0

Function code	Word	Value range	Default value
P16.35	Received PZD5	4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)	0
P16.36	Received PZD6	5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz)	0
P16.37	Received PZD7	6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz)	0
P16.38	Received PZD8	7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)	0
P16.39	Received PZD9	8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)	0
P16.40	Received PZD10	9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence)	0
P16.41	Received PZD11		0
P16.42	Received PZD12	10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2, RO1, HDO, and Y1 in sequence) 11: Voltage setting (for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor) 12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%) 13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%) 14: MSB of position reference (signed number) 15: LSB of position reference (unsigned number) 16: MSB of position feedback (signed number) 17: LSB of position feedback (unsigned number) 18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0)	0

Response packet (VFD -> master station)

SW: The first word in a PZD response packet is a VFD SW. You can select the expression method according to P15.43.

Table 2.8 and Table 2.9 describe the control words (CWs) of the Goodrive 350 series VFD.

Table 2-8 Goodrive350 series VFD SWs expressed in decimal format

Bit	Name	Value	Description
0–7	Running state	1	Forward running
		2	Reverse running
		3	Stopped
		4	Faulty
		5	POFF
8	Bus voltage established	1	Ready to run
		0	Not ready to run
9–10	Motor group feedback	0	Motor 1
		1	Motor 2
11	Motor type feedback	1	Synchronous motor
		0	Asynchronous motor
12	Overload pre-alarm feedback	1	Overload pre-alarm generated
		0	No overload pre-alarm generated
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

Table 2-9 Goodrive350 series VFD SWs expressed in binary format

Bit	Name	Description	Priority
0	Forward running	0: Disable 1: Enable	1
1	Reverse running	0: Disable 1: Enable	2
2	Stopped	0: Disable 1: Enable	3
3	Fault	0: Disable 1: Enable	4
4	POFF	0: Disable 1: Enable	5
5	Pre-excited	0: Disable 1: Enable	6
6–15	/	Reserved	

Actual value (ACT): The second to twelfth words in a PZD task packet are the main actual values. The main actual frequency values are provided by the main actual value signal source. Table 2.10 lists the actual status values of the Goodrive 350 series VFD.

Table 2-10 Actual status values of Goodrive350 series VFD

Function code	Word	Value range	Default value
P16.43	Transmitted 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: Rotating speed of the running (×1, RPM) 9: Linear speed of the running (×1, m/s) 10: Ramp frequency reference 11: Fault code 12: AI1 value (×100, V) 13: AI2 value (×100, V) 14: AI3 value (×100, V) 15: HDIA frequency (×100, kHz) 16: Terminal input state 17: Terminal output state 18: PID reference (×100, %) 19: PID feedback (×100, %) 20: Rated torque of the motor 21: MSB of position reference (signed number) 22: LSB of position reference (unsigned number) 23: MSB of position feedback (signed number) 24: LSB of position feedback (unsigned number) 25: Status word 26: HDIB frequency value (×100, kHz)	0
P16.44	Transmitted PZD3		0
P16.45	Transmitted PZD4		0
P16.46	Transmitted PZD5		0
P16.47	Transmitted PZD6		0
P16.48	Transmitted PZD7		0
P16.49	Transmitted PZD8		0
P16.50	Transmitted PZD9		0
P16.51	Transmitted PZD10		0
P16.52	Transmitted PZD11		0
P16.53	Transmitted PZD12		0

PKW zone

PKW zone (parameter identification flag PKW1—numerical zone): The PKW zone describes the processing mode of the parameter identification interface. A PKW interface is not a physical interface but a mechanism that defines the transmission mode (such reading and writing a parameter value) of arameter between two communication ends.

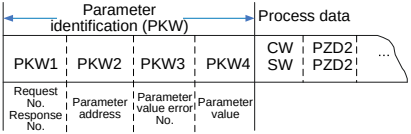


Figure 2-5 Parameter identification zone

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

First word PKW1 (16 bits)		
Bits 15–00	Task or response identification 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 packet 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 2-11 Task identification flag PKW1

Request No. (from the master to a slave)		Response signal	
Request No.	**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 or 4
3	Modifying a parameter value (two words) [modifying the value only on RAM]	2	3 or 4
4	Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM]	1	3 or 4
5	Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM]	2	3 or 4

Note: The requests #2, #3, and #5 are not supported currently.

Table 2-12 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: Password error 6: Data frame error 7: Parameter read only 8: Parameter cannot be modified during VFD running 9: Password protection

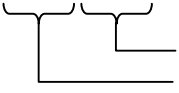
PKW examples

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



0010: Parameter address

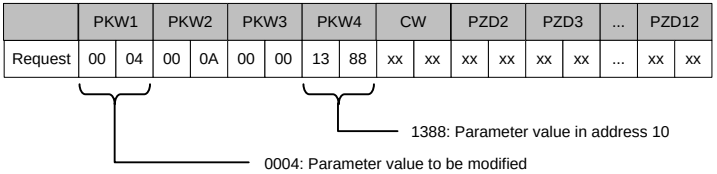
0001: Request for reading parameter values

Response (VFD -> master station)

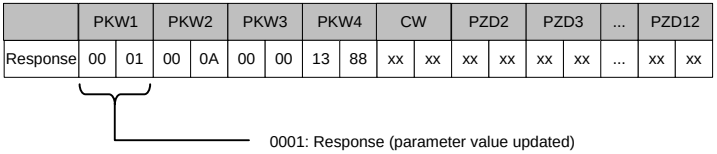
Example 2: Modifying the value of a parameter (on both RAM and EEPROM)

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 (50.00) is in PKW4.

Request (master station -> VFD)



Response (VFD-> master station)



PZD examples: The transmission of the PZD zone is implemented through VFD function code settings. For the function codes, see the related INVT VFD operation manual.

Example 1: Reading the process data of a VFD

In this example, PZD3 is set to "8: Rotating speed of the running" through the VFD parameter P15.14. This operation sets the parameter forcibly. 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, PZD3 is set to "2: PID reference" through the VFD parameter P15.03. 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 tractive force reference in each request frame until another parameter is specified.

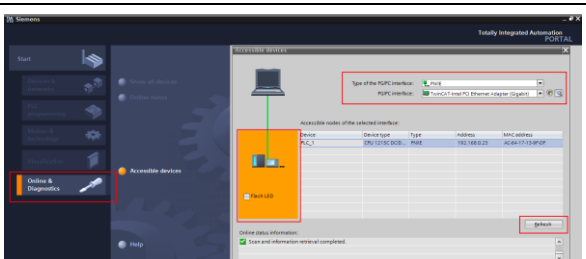
2.5 Example of PLC communication

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

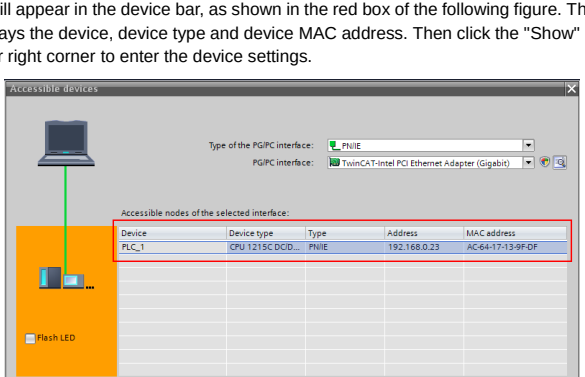
2.5.1 Parameter configuration

Connect the PLC to the PC with a standard network cable, and set the computer IP (e.g. 192.168.0.100) in the PC network settings. Set the IP and name of the PLC.

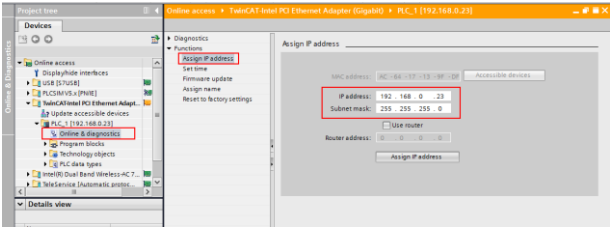
1) Open the "TIA PORTAL V13" software, and click "Online & Diagnostics" --> "Accessible Devices" on the left. Select "PN/IE" in the drop-down list of "Type of the PG/PC interface", select the Ethernet port in the "PG/PC Interface", and finally click "Refresh" to scan the connected PLC devices, as shown in the following figure.



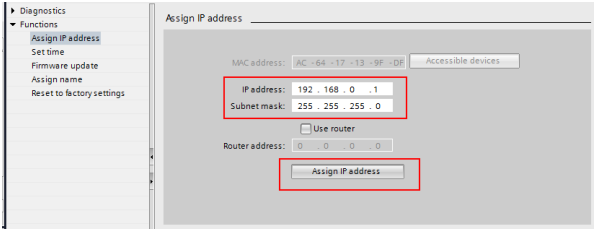
2) If the connection between the PLC and PC is normal, after scanning is completed, the PLC device will appear in the device bar, as shown in the red box of the following figure. The device bar displays the device, device type and device MAC address. Then click the "Show" button in the lower right corner to enter the device settings.



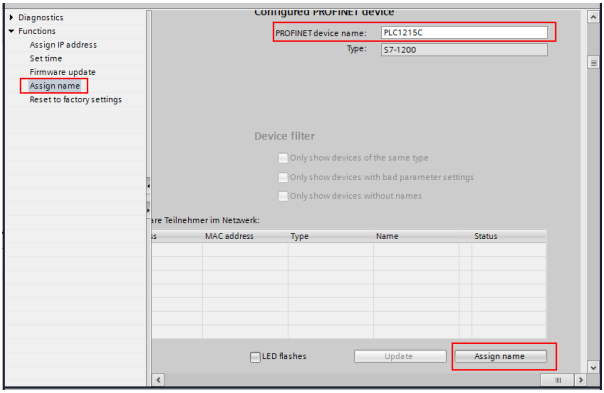
3) Click "Online & Diagnostics" in the device tree, click "Assign IP Address" under the “Functions” on the right of the menu bar, and set the IP address and subnet mask of the PLC shown in the red box marked ③, to ensure that the IP address of the PC and the IP address of the PLC are in the same network segment, as shown in the following figure.



4) Set the IP address of the PLC to "192.168.0.1" and subnet mask to "255.255.255.0" (you can check "Use router", that is, the router assigns IP). Click the "Assign IP address" button after the setting is completed, as shown in the following figure.

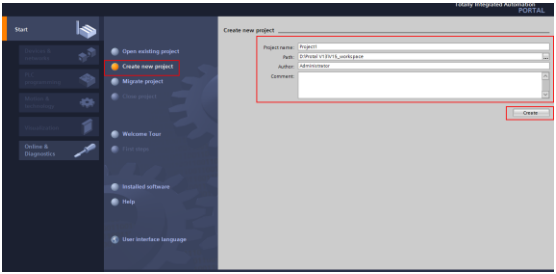


5) Click "Assign Name", and mark the PLC name in the position shown in the red box marked ②, such as "PLC1215C". Click the "Assign Device Name" button, as shown in the following figure.

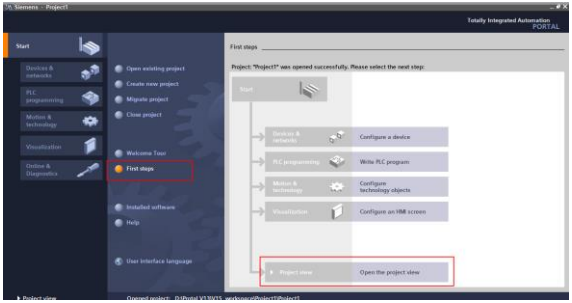


2.5.2 Create a new project

Double click the TIA PORTAL V13 icon to open the TIA PORTAL V13 project tool. Click the “Create new project” button to create a new project, add project name, project storage path, author, comment and other related information, and click the “Create” button to create a new project, as shown in the following figure.

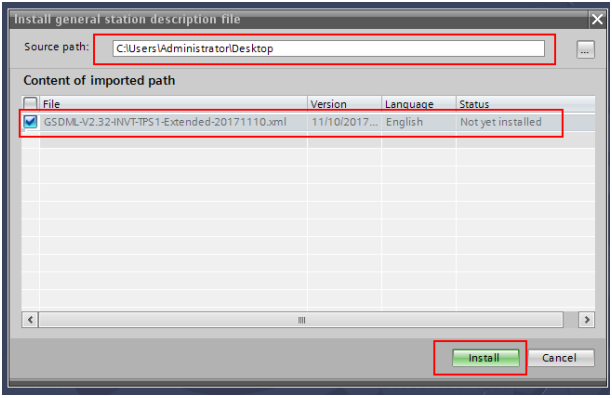


After creating a new project, double click “Open the project view”, as shown in the following figure.

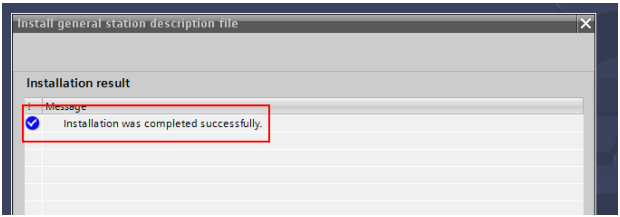


2.5.3 Add GSD files

In the project view, click "Options" on the toolbar, select the "Manage general station description files (GSD)" option from the drop-down list, and a box pops up, as shown in the following figure. Enter the file directory where the INVT GSD file is located in the source path, select the GSD file, and click the "Install" button to start the installation.



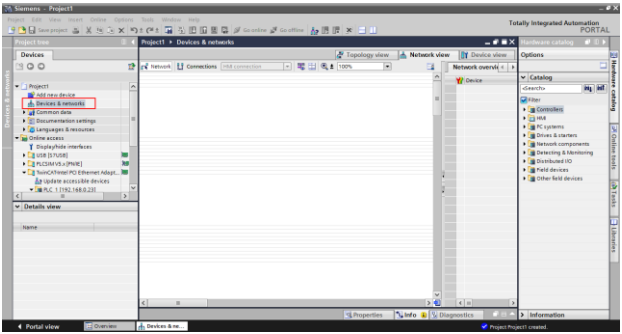
After the installation was completedly successfully, a prompt pops up, indicating that the GSDML file has been installed successfully, as shown in the following figure.



2.5.4 Configure the basic information of the project

1) Enter the “Devices & networks” view interface

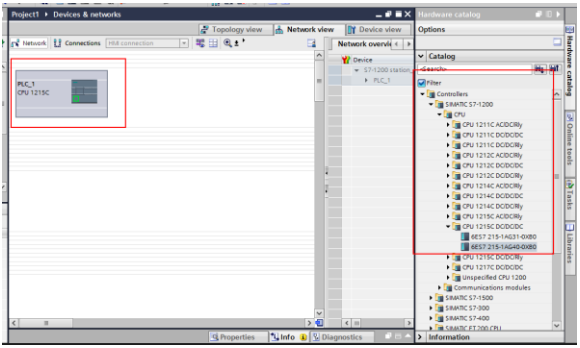
In the project view, select and double click "Devices & networks" in the project tree on the left to enter the “Network overview” view interface, as shown in the following figure.



2) Add Project device and PROFINET network

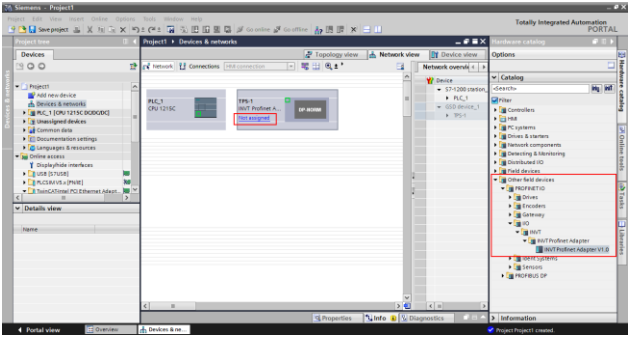
1) Add PLC S7-1215C to the "Devices & networks" view

In the "Hardware catalog" on the right sidebar, select "Controller" → "SIMATIC S7-1200" → "CPU" → "CPU 1215C AC/DC/Rly" → "6ES7 215-1BG40-0XB0", and double click the "6ES7 215-1BG40-0XB0" icon or drag it to the project, as shown in the following figure.

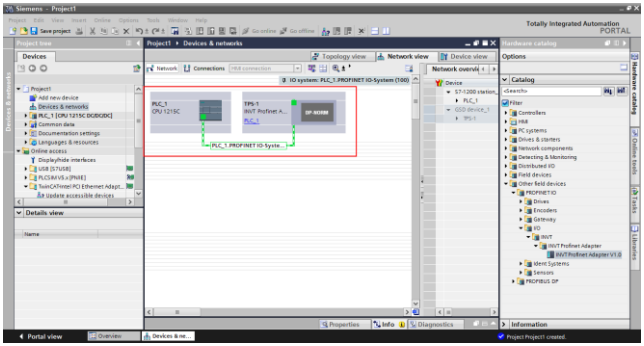


(2) Add the INVT communication card to the "Devices & networks" view

In the "Hardware Catalog", click "Other field devices" → "Profinet IO" → "I/O" → "INVT" → "INVT Profinet Adapter" → "INVT Profinet Adapter V1.0", and double click the "INVT Profinet Adapter V1.0" icon or drag it to the view of "Devices & networks". The communication card is displayed as "Not assigned", as shown in the following figure.

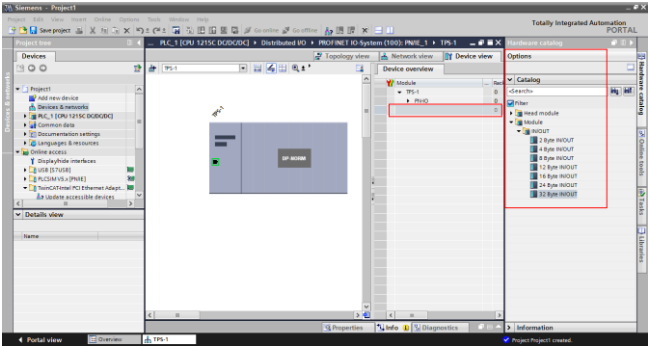


Click the "Not assigned" option of " INVT Profinet Adapter V1.0" and select the IO controller " PLC_1. PROFINET IO-System", then CPU and INVT Profinet in the network view are connected to the same Profinet subnet, as shown in the following figure.

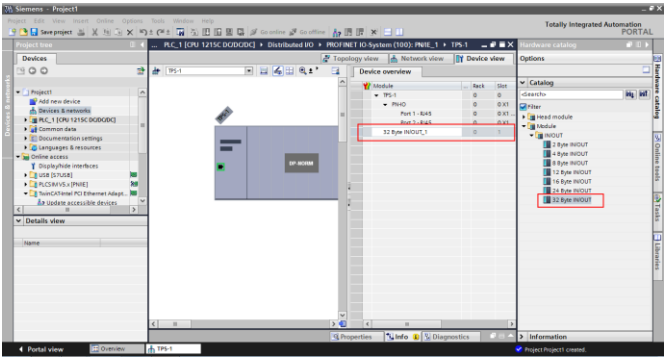


3) Add the INVT I/O sub-module to the project

Double click the "INVT Profinet Adapter V1.0" icon in the "Devices & Networks" view to enter the “Device view” interface, as shown in the following figure.



Click the "Hardware Catalog" on the right → "Module", double click the "32 Byte IN/OUT" module or drag it to the blank space in the "Device view", and the "32 Byte IN/OUT" module is added to the project, as shown in the following figure.



(4) Simple configuration of S7-1215C and INVT Profinet parameters

<1>Configure parameters of PLC S7-1215C

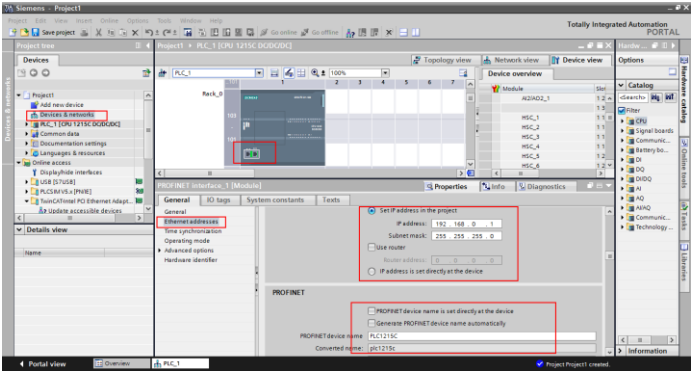
Double click the "Devices & Networks" option to enter the view interface of "Devices & Networks".

Double click the “PLC S7-1215C” icon in the interface to enter the "Device view" interface of the PLC.

Double click the network interface position in the PLC icon to enter the properties editing interface bar of “PROFINET interface_1”.

Click the "Ethernet addresses" option in the "General" list to set the PLC address and name (In this example, IP address of the PLC is 192.168.0.1 and PLC name is PLC1215C).

Operations are shown in the following figure.



<2> Configure parameters of the INVT Profinet communication card

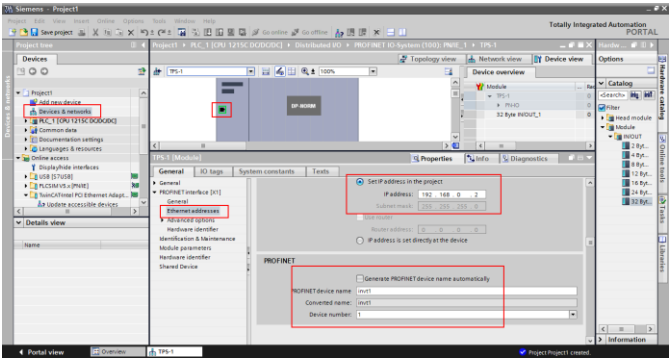
Double click the "Devices & Networks" option to enter the view interface of "Devices & Networks".

Double click the “INVT Profinet Adapter V1.0” icon in the interface to enter the "Device view" interface of the communication card.

Double click the network interface position in the INVT Profinet communication card icon to enter the properties editing interface bar of PROFINET interface.

Click the "PROFINET interface [X1]" option in the "General" list, and click the "Ethernet addresses" option. Configure parameters of the INVT PROFINET communication card according to the parameters shown in the following figure such as IP address and device name of the communication card (in this example, IP address of the communication card is 192.168.0.2 and the name is invt1).

Operations are shown in the following figure.



2.5.5 Assign the device name of the IO device (INVT communication card)

After the CPU and INVT Profinet communication card are successfully connected to the PC through the network cable, click "Online access" on the left to find the network card corresponding to the PC that is connected to the PLC and communication card.

In all displayed devices, find the INVT communication card device and click it, such as emc (192.168.0.2) device, as shown in the following figure (**Note:** When the communication card is used for the first time, there is no device name, and only the default IP can be scanned) .

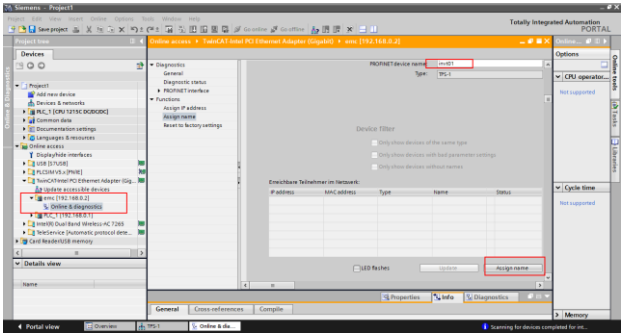
Double click "Online & Diagnostics" to enter the online diagnostics state.

Click "Functions" → "Assign name" to enter the "Assign name" interface.

Enter the communication card name in “PROFINET device name”, and click "Assign Name" in the lower right corner to confirm.

Note: The name of the PROFINET communication card set online must be consistent with that set in the configuration project, otherwise PROFINET communication cannot be carried out between the devices.

The operation steps are shown in the following figure.



2.5.6 Save, compile, and download

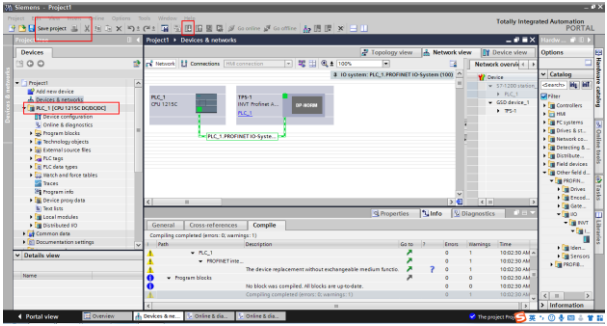
Download the project configuration information to the PLC S7-1215C after the entire project configuration is completed.

Click "Save Project" to save the entire project.

Right click "PLC_1 [CPU 1215C AC/DC/Rly]" → left click "Compile" → "Hardware and software (change only)" to compile the entire project.

Click the "Download to device" icon to download the project configuration to the PLC controller.

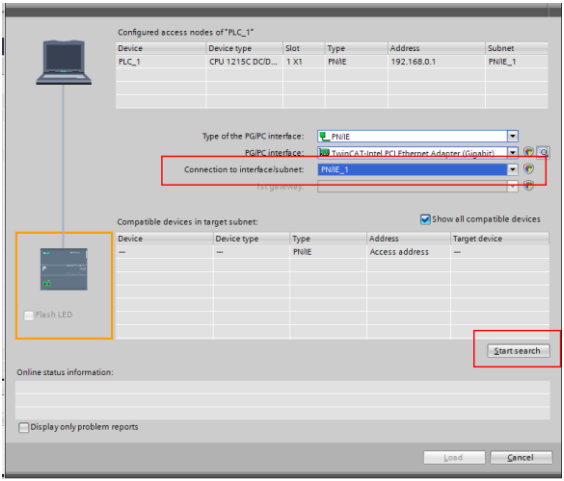
Operations are shown in the following figure.



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

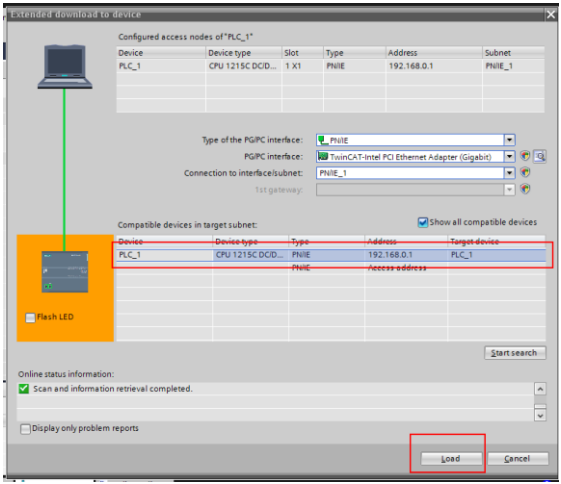
Select the "PN/ IE_1" option in the drop-down list of "Connection to interface/subnet".

Click the "Start search" button in the lower right corner to start scanning and detecting PLC devices in the subnet.



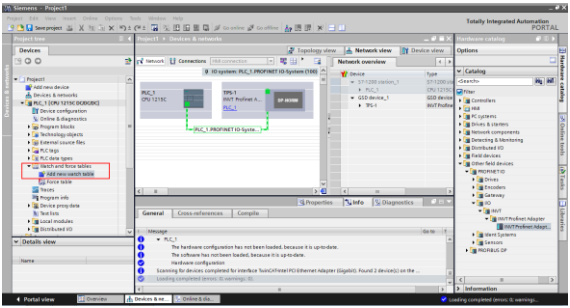
After searching is completed, the PLC S7-1215C that is connected to the PC will be displayed in the list of "Compatible devices in target subnet", as shown in the following figure.

Select the PLC to be downloaded in the following figure, and click the "Download" button to download the configuration information and PLC program to the selected PLC.

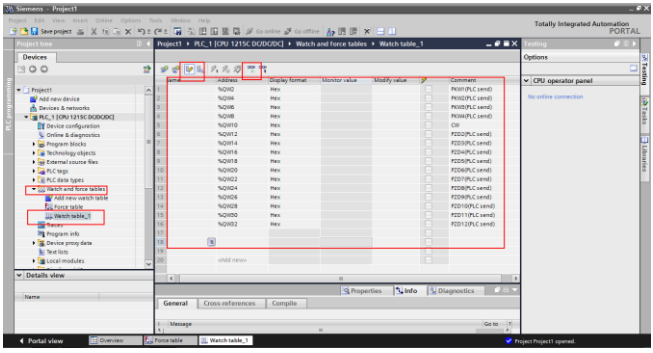


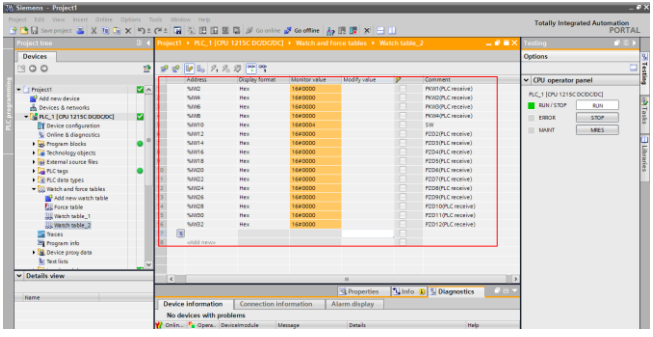
2.5.7 VFD parameter watching

Click "Watch and force tables" in the left menu bar, and double click "Add new watch table" in the drop-down menu, as shown in the following figure.



Create target watch variables—PZD, PKW, control word and status word variables of the VFD in the newly created watch table, as shown in the following figure.





After the watch variables are created, click the "Watch all" button in the watch table to monitor the values of all variables, and click the "Modify parameters" button in the watch table to modify the parameters of the target variable, so as to watch the VFD through the PLC.

3 Ethernet IP communication card

3.1 Overview

1. Thanks for choosing INVT Ethernet IP communication cards. This manual describes the function specifications, installation, basic operation and settings, and information about the network protocol. To ensure that you install and operate the product properly, read this manual and the communication protocol section in the VFD operation manual carefully before you use the product.
2. This manual only describes how to operate the Ethernet IP communication card and the related commands but does not provide details about the EtherNet/IP protocol. For more information about the Ethernet IP protocol, read the related specialized articles or books.
3. This communication card is defined as an Ethernet IP slave station communication card and is used on a VFD that supports EtherNet/IP communication.
4. The communication card supports the star, linear, and ring topologies.
5. The communication card supports 32 inputs/outputs to read and write process data, read state data, and read and write function parameters of a VFD.

3.2 Features

1. Supported functions

- Supports the EtherNet/IP protocol, and supports EtherNet/IP devices.
- Provides two EtherNet/IP ports and supports the 10/100M full-duplex/half-duplex operation.
- Supports the star, linear, and ring topologies (but does not support ring-network monitoring).

2. Supported communication types

EtherNet/IP adopts the application layer protocol CIP, which is also used by DeviceNet and ControlNet. Therefore, they use the same object library and consistent industrial specifications.

CIP uses non-connected UDP/IP and connection-based TCP/IP for information control and transmission over the Ethernet, allowing the sending of explicit and implicit packets. Implicit packets are time-critical control messages and transmitted using UDP/IP. Explicit packets are point-to-point messages that are not time critical and transmitted using TCP/IP. Explicit packets are used for configuration, download, and fault diagnosis, while implicit packets are used for real-time I/O data transmission.

3. Communication ports

Standard RJ45 ports are used in EtherNet/IP communication. The communication card provides two RJ45 ports with no transmission direction defined, and therefore you can insert a cable into the port without regard to its direction. Figure 3-1 shows the ports, and Table 3-1 describes the port pins.

Figure 3-1 Two standard RJ45 ports

Table 3-1 Standard RJ45 port pins

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

4. State indicators

The EtherNet/IP communication card provides four LED indicators and four net port indicators to indicate its states. Table 3-2 describes the state indicators.

Table 3-2 State indicators

LED	Color	State	Description
LED1	Green	On	Indicating that the card and VFD identify each other.
		Blinking (1Hz)	Indicating that the card and VFD communicate normally.
		Off	Indicating that the card and VFD communicate improperly.
LED2	Green	On	Indicating that communication between the card and PLC is online and data interchange is allowed.

LED	Color	State	Description
		Blinking (1Hz)	Indicating IP address conflict between the card and PLC.
		Off	Indicating that communication between the card and PLC is offline.
LED3	Red	On	Failed to set up I/O between the card and PLC.
		Blinking (1Hz)	Incorrect PLC configuration.
		Blinking (2Hz)	The card failed to send data to the PLC.
		Blinking (4Hz)	The connection between the card and PLC timed out.
		Off	No fault
LED4	Red	On	3.3V power indicator
Net port indicator	Yellow	On	Link indicator, indicating successful Ethernet connection.
		Off	Link indicator, indicating that Ethernet connection is not established.
Net port indicator	Green	On	ACK indicator, indicating that data interchange being performed.
		Off	ACK indicator, indicating that data interchange is not be performed.

3.3 Electrical wiring

The Ethernet IP communication card provides standard RJ45 ports and supports the linear, star, and ring topologies. Figure 3-2, Figure 3-3, and Figure 3-4 show the electrical wiring diagrams for different topologies.

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

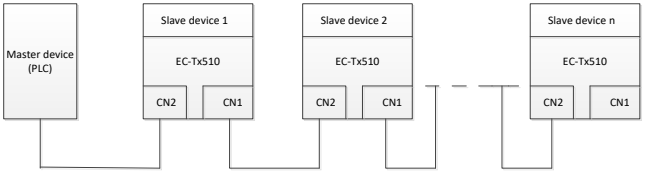


Figure 3-2 Electrical wiring diagram for a linear topology

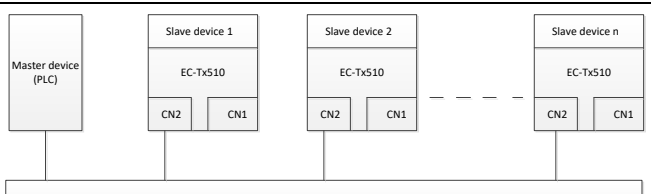


Figure 3-3 Electrical wiring diagram for a star topology

Note: An Ethernet switch must be available when the star topology is used.

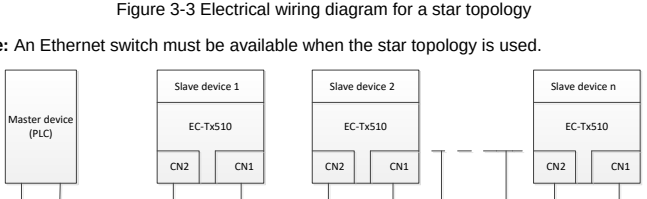


Figure 3-4 Electrical wiring diagram for a ring network

3.4 Communication

3.4.1 Communication settings

The EtherNet/IP communication card can function as only the EtherNet/IP slave station. Before communication, set Goodrive350 function codes, including:

- IP address and subnet mask for the card

The default IP address and subnet mask for each communication card are 192.168.0.1 and 255.255.255.0. You can change them to the address of a network segment.

- Control mode

If you want to control the VFD with the communication card, set the control mode to EtherNet/IP communication control. To be specific, set P00.01=2 (communication as the running command channel) and set P00.02=3 (EtherNet/IP communication channel) to control VFD start and stop. If you want to set a value through EtherNet/IP communication, change the control way of corresponding function codes to EtherNet/IP communication. Appendix B lists related function codes.

Note: After the setting, the card can communicate normally. If you want to control the VFD with the card, set related function codes to enable EtherNet/IP communication control.

3.4.2 Packet format

Table 3-3 describes the structure of a TCP communication packet.

Table 3-3 Structure of a TCP communication packet

MAC-layer packet header	IP-layer packet header	TCP-layer packet header	Valid data	Packet trailer
14 bytes	20 bytes	20 bytes	0–1488 bytes	4 bytes

Table 3-4 describes the structure of a UDP communication packet.

Table 3-4 Structure of a UDP communication packet

MAC-layer packet header	IP-layer packet header	UDP-layer packet header	Valid data	Packet trailer
14 bytes	20 bytes	20 bytes	0-1488 bytes	4 bytes

3.4.3 Ethernet IP communication

The EtherNet/IP communication card supports 16-word input/output. Figure 3-5 shows the packet format for transmitting data with a VFD.

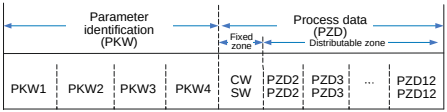


Figure 3-5 Packet structure

By using the 32 inputs/outputs, you can set the reference parameters of the VFD, monitor the status values, transmit control commands, monitor the running state, and read/write the function parameters of the VFD. For specific operations, see the following description.

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—Control word (transmitted from the master to a slave. For description, see Table 3-5.)

SW—Status word (transmitted from a slave to the master. For description, see Table 3-8.)

PZD—Process data (user defined)

(The process data output from the master to a slave is a reference value, and the process data input from a slave to the master is an actual value.)

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

CWs and SWs

Using CWs is the basic method of the fieldbus system to control VFDs. A CW is transmitted by the fieldbus master station 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 state information back to the master through an SW.

Reference value: A 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 EtherNet/IP, you need to set the communication module as the controller of the VFD device.

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 from the VFD device to the master depends on the set function. For more description, see the related VFD operation manual.

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

Task packet (master station -> VFD)

CW: The first word in a PZD task packet is a VFD CW.

When P15.43=0, EtherNet IP control words are defined by byte. Table 3-5 describes Goodrive350 series VFD CWs defined by byte.

Table 3-5 Goodrive350 series VFD CWs expressed in decimal format

Bit	Name	Value	Description
0–7	Communication-based control command	1	Forward running
		2	Reverse running
		3	Forward jogging
		4	Reverse jogging
		5	Stop
		6	Coast to stop (emergency stop)
		7	Fault reset
		8	Jogging to stop
		9	Decelerate to stop
8	Enabling writing	1	Enable writing (mainly through PKW1 to PKW4)
9–10	Motor group setting	00	Motor 1
		01	Motor 2
11	Control mode switching	1	Enable torque/speed control switching
		0	Disable switching
12	Resetting power consumption to zero	1	Enable
		0	Disable
13	Pre-excitation	1	Enable
		0	Disable
14	DC braking	1	Enable
		0	Disable
15	Heartbeat reference	1	Enable
		0	Disable

When P16.56=1, EtherNetIP control words are defined by bit. Table 3-6 describes Goodrive350 series VFD CWs defined by bit.

Table 3-6 Goodrive350 series VFD CWs expressed in binary format

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: Disable 1: Enable	3
3	Coast to stop	0: Disable 1: Enable	4
4	Forward jogging	0: Disable 1: Enable	5
5	Reverse jogging	0: Disable 1: Enable	6
6	Jogging to stop	0: Disable 1: Enable	7

Bit	Name	Description	Priority
7	/	Reserved	
8	Enable reading and writing (PKW1-PKW4)	0: Disable 1: Enable	
9	/	Reserved	
10	Decelerate to stop	0: Disable 1: Enable	0: Top priority
11–15	/	Reserved	

Reference value (REF): The second to twelfth words in a PZD task packet are the main settings. The main frequency settings are provided by the main setting signal source. Table 3-7 describes the settings of Goodrive350 series VFD.

Table 3-7 Settings of Goodrive350 series VFD

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

Function code	Word	Value range	Default value
		12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%) 13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%) 14: MSB of position reference (signed number) 15: LSB of position reference (unsigned number) 16: MSB of position feedback (signed number) 17: LSB of position feedback (unsigned number) 18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0)	

Response packet (VFD -> master station)

Status word (SW): The first word in a PZD response packet is a VFD SW.

P15.43=0 (SWs are defined in decimal format), and the VFD SWs are defined as follows.

Table 3-8 Goodrive350 series VFD SWs expressed in decimal format

Bit	Name	Value	Description
0–7	Running state	1	Forward running
		2	Reverse running
		3	Stopped
		4	Faulty
		5	POFF
8	Bus voltage established	1	Ready to run
		0	Not ready to run
9–10	Motor group feedback	0	Motor 1
		1	Motor 2
11	Motor type feedback	1	Synchronous motor
		0	Asynchronous motor
12	Overload pre-alarm feedback	1	Overload pre-alarm generated
		0	No overload pre-alarm generated
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

P15.43=1 (SWs are defined in binary format), and the VFD SWs are defined as follows.

Table 3-9 Goodrive350 series VFD SWs expressed in binary format

Bit	Name	Description	Priority
0	Forward running	0: Disable 1: Enable	1
1	Reverse running	0: Disable 1: Enable	2
2	Stopped	0: Disable 1: Enable	3
3	Fault	0: Disable 1: Enable	4
4	POFF	0: Disable 1: Enable	5
5	Pre-excited	0: Disable 1: Enable	6
6–15	/	Reserved	

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

Table 3-10 Actual status values of Goodrive350 series VFD

Function code	Word	Value range	Default value
P16.43	Transmitted PZD2	0: Invalid	0
P16.44	Transmitted PZD3	1: Running frequency (×100, Hz)	0
P16.45	Transmitted PZD4	2: Set frequency (×100, Hz)	0
P16.46	Transmitted PZD5	3: Bus voltage (×10, V)	0
P16.47	Transmitted PZD6	4: Output voltage (×1, V)	0
P16.48	Transmitted PZD7	5: Output current (×10, A)	0
P16.49	Transmitted PZD8	6: Actual output torque (×10, %)	0
P16.50	Transmitted PZD9	7: Actual output power (×10, %)	0
P16.51	Transmitted PZD10	8: Rotating speed of the running (×1, RPM) 9: Linear speed of the running (×1, m/s) 10: Ramp frequency reference	0
P16.52	Transmitted PZD11	11: Fault code	0
P16.53	Transmitted PZD12	12: AI1 value (×100, V) 13: AI2 value (×100, V) 14: AI3 value (×100, V) 15: HDIA frequency (×100, kHz) 16: Terminal input state 17: Terminal output state 18: PID reference (×100, %) 19: PID feedback (×100, %) 20: Rated torque of the motor 21: MSB of position reference (signed number)	0

Function code	Word	Value range	Default value
		22: LSB of position reference (unsigned number) 23: MSB of position feedback (signed number) 24: LSB of position feedback (unsigned number) 25: Status word 26: HDIB frequency value (×100, kHz)	

PKW zone

PKW zone (parameter identification flag PKW1—numerical zone): The PKW zone describes the processing mode of the parameter identification interface. A PKW interface is not a physical interface but a mechanism that defines the transmission mode (such reading and writing a parameter value) of arameter between two communication ends.

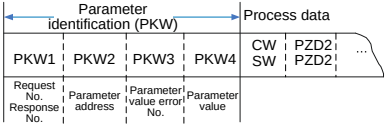


Figure 3-6 Parameter identification zone

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

First word PKW1 (16 bits)		
Bits 15–00	Task or response identification 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 packet 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 3-11 Task identification flag PKW1

Request No. (from the master to a slave)		Response signal	
Request No.	**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 or 4
3	Modifying a parameter value (two words) [modifying the value only on RAM]	2	3 or 4
4	Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM]	1	3 or 4
5	Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM]	2	3 or 4

Note: The requests #2, #3, and #5 are not supported currently.

Table 3-12 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: Password error 6: Data frame error 7: Parameter read only 8: Parameter cannot be modified during VFD running 9: Password protection

➢ Model specified in the standard ODVA agreement

The standard ODVA protocol specifies the data transmission format and CWs/SWs definitions, and the packet format for data transmission with the VFD is shown in Table 3.13.

Table 3.13 Transmission modes specified in standard ODVA protocol

No.	Input/Output	Data length (bytes)	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

In which, CW1/SW1 and CW2/SW2 are defined as shown in Tables 3.14, 3.15, 3.16 and 3.17.

Table 3.14 CW1 specified in standard ODVA protocol

Bit	Name	Value	Description
0	Forward running	0	Disable
		1	Enable
1	Reserved	/	/
2	Fault reset	0	Disable
		1	Enable
3–15	Reserved	/	/

Table 3.15 SW1 specified in standard ODVA protocol

Bit	Name	Value	Description
0	Fault state	0	No fault
		1	Fault
1	Reserved	/	/
2	Running state	0	Not forward running
		1	Forward running
3–15	Reserved	/	/

Table 3.16 CW2 specified in standard ODVA protocol

Bit	Name	Value	Description
0	Forward running	0	Disable
		1	Enable
1	Reverse running	0	Disable
		1	Enable
2	Fault reset	0	Disable
		1	Enable

Bit	Name	Value	Description
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 3.17 SW2 specified in standard ODVA protocol

Bit	Name	Value	Description
0	Fault	0	No fault
		1	Fault
1	Overload prealarm feedback	0	No overload
		1	Overload prealarm
2	Running state 1	0	Stopped
		1	Forward running
3	Running state 2	0	Stopped
		1	Reverse running
4	Bus voltage established	0	Ready to run
		1	Not ready to run
5	Control reference source	0	Local control (keypad)
		1	Remote control (not keypad)
6	Frequency/torque reference source	0	Local control (keypad)
		1	Remote control (not keypad)
7	Reference reached	0	Not reached
		1	Reached
8–15	Reserved	/	/

➢ INVT extended data model based on the ODVA Protocol

Based on the ODVA protocol provisions, these four modes are combined with PZD process data defined by INVT, and the packet format for data transmission with the VFD is shown in Table 3.18.

Table 3.18 INVT extended data model based on the ODVA protocol

No.	Input/Output	Data length (bytes)	Format (word)
6	74/24	24	CW1/SW1 + Speed_ref/act + Null +PZD4–12

No.	Input/Output	Data length (bytes)	Format (word)
7	75/25	24	CW2/SW2 + Speed_ref/act + Null +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, definitions of CWs and SWs are consistent with that of "Model specified in the standard ODVA agreement", and definitions of PZD4–12 are consistent with that of "INVT self-defined mode".

3.5 Example 1 of PLC communication (communicate with Allen-Bradley PLC)

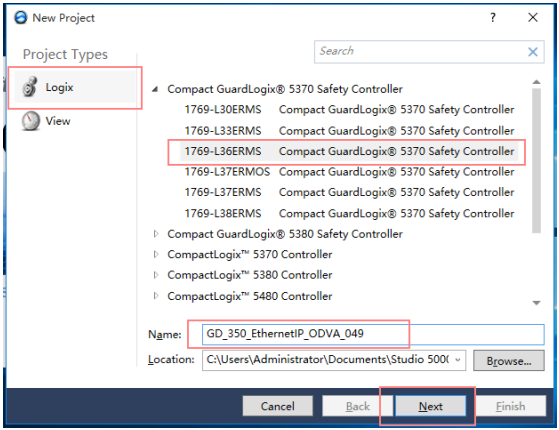
This example shows how to use an Allen-Bradley PLC (model: 1769_L36ERMS) to communicate with an Ethernet IP adapter module (through using the Studio 5000 software as the configuration tool).

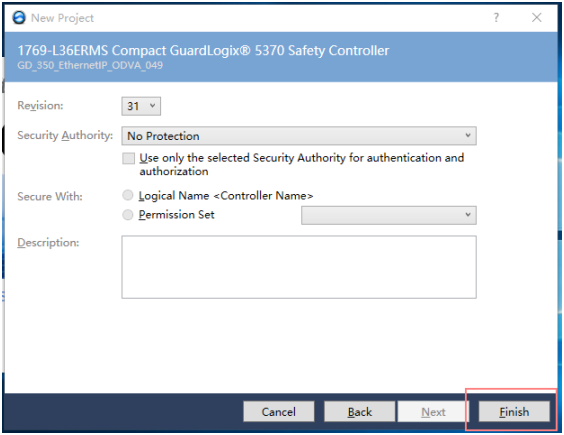
3.5.1 Create a new project

Connect the PC to the PLC with a printer cable or network cable. Open 

software, and click “New Project”.

Select the correct PLC model, fill in the project name, click "Next", and click "Finish".



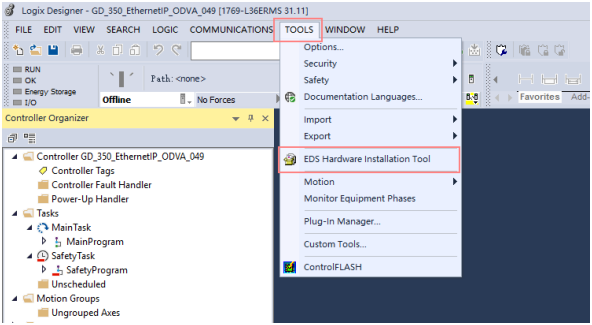


3.5.2 Import an EDS file

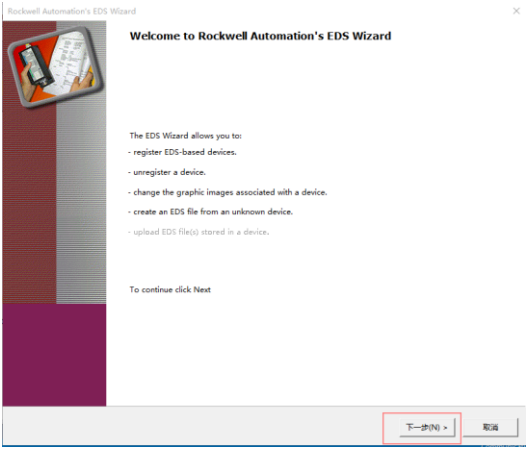
The EDS file is used to specify device attributes for Ethernet IP client. The client identifies the device through product code, device type, and major version attributes.

You can obtain the EDS file of the communication card from the vendor, or downloaded it (file name: GD350_EthernetIP_V1.01.eds) from the INVT website at www.invt.com.cn.

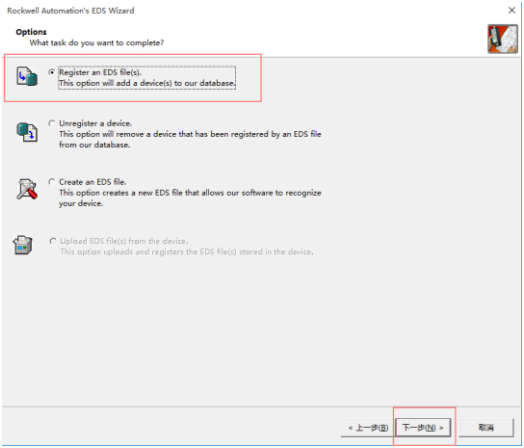
Right click "TOOLS", and select "EDS Hardware Installation Tool".



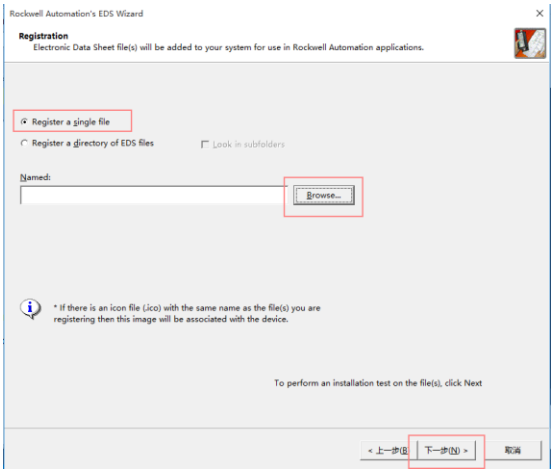
Click "Next".



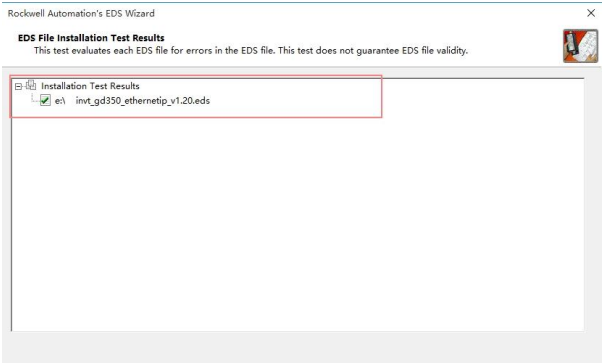
Select the option as shown in the following figure, and click "Next".



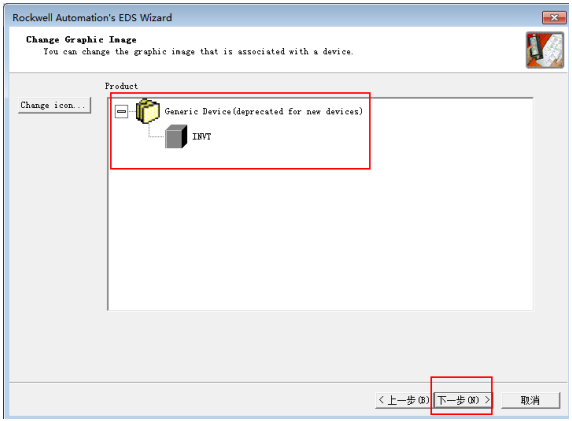
Click "Browse" to select the EDS file that you want to download, and then click "Next".



Continue to click "Next".

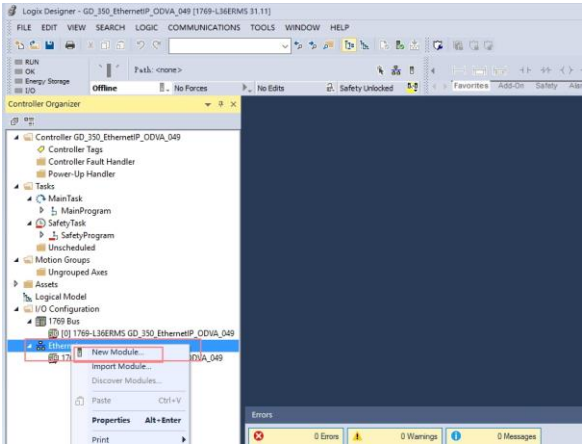


Click "Next" again, and the installation is successful.

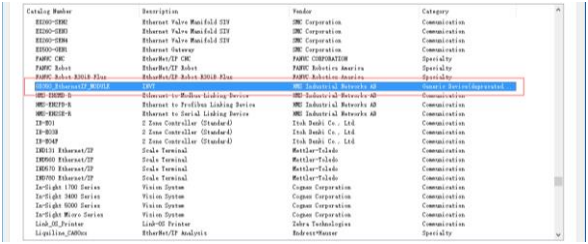


3.5.3 Create a new device object

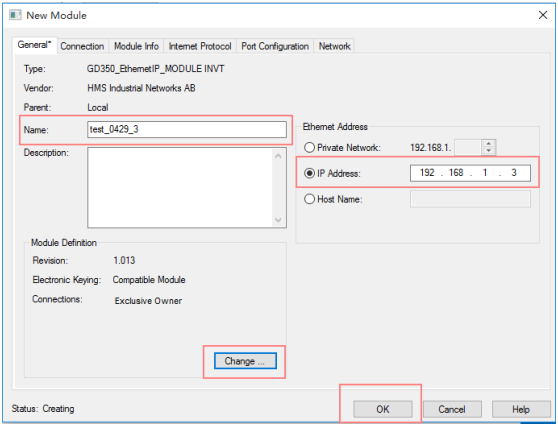
Select "I/O Configuration"—>"Ethernet item" on the left, and right click "New Module".



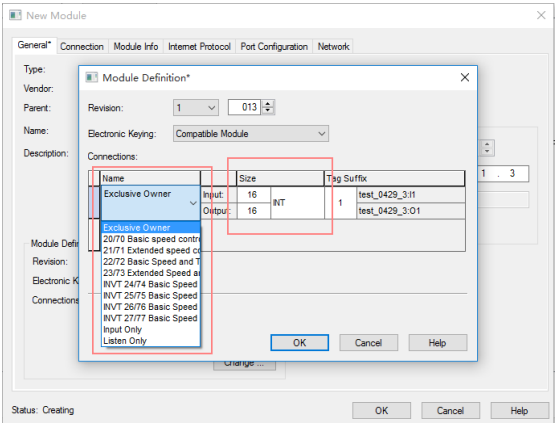
Select "GD350_EthernetIP_MODULE", and click "Create".



Fill in the module name, and set the IP address of the module. The IP address must be consistent with P16.02–P16.05 on the GD350 Ethernet IP communication card, otherwise communication fails.

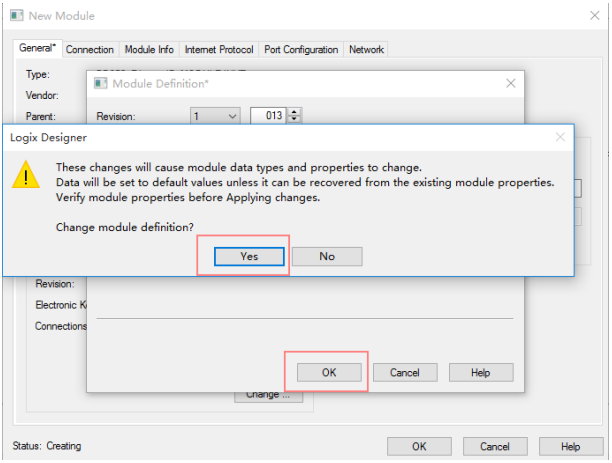


Click the "Change" option to select the protocol type used by the module. Each type differs in IO format, so you need to select the corresponding IO format based on the protocol type, as shown in the following table. Take "Exclusive Owner" as an example.

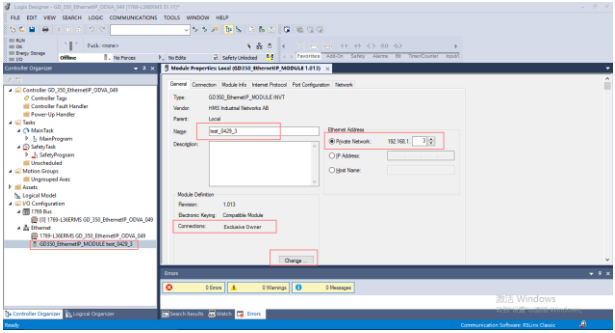


Name	Size	Format
Exclusive Owner	16	INT
20/70 Basic speed control	2	INT
21/71 Extended speed control	2	INT
22/72 Basic Speed and Torque control	3	INT
23/73 Extended Speed and Torque control	3	INT
INVT 24/74 Basic Speed Control plus Drive Parameters	12	INT
INVT 25/75 Enhanced Speed Control plus Drive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

Click "OK", "Yes", "OK", "OK", "OK", and "OK" in turn.



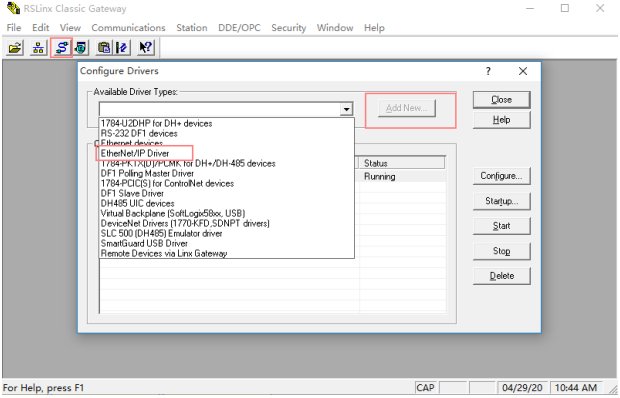
Once the module has been created successfully, you can see it under the "Ethernet" item under "I/O Configuration" on the left, and click it to check the device information.

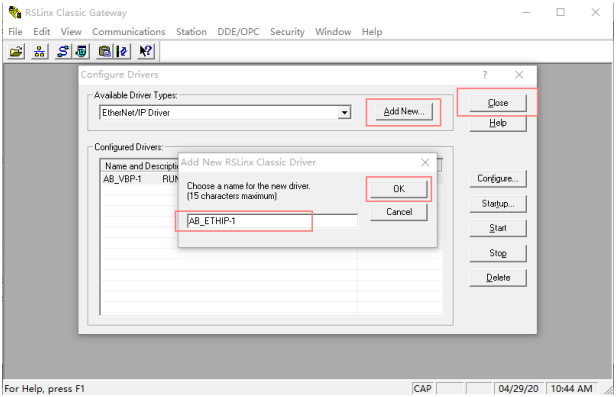


3.5.4 Use of Rslinx Classic

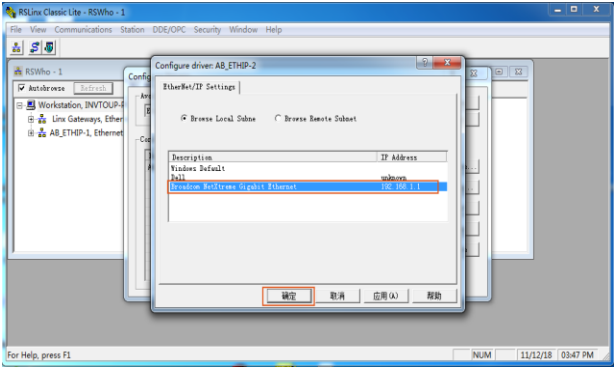
Rslinx Classic is used to connect the PC to the PLC. Open the "Rslinx Classic" software.

Click the "S" icon, and a window of "Configure Drivers" pops up. Select "Ethernet/IP Driver" in the drop-down menu of "Available Driver Types", click "Add New", a window of "Add New RSLinx Classic Driver" pops up, and click "OK".



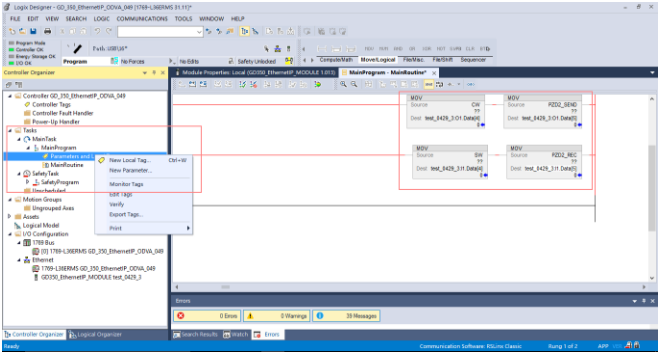


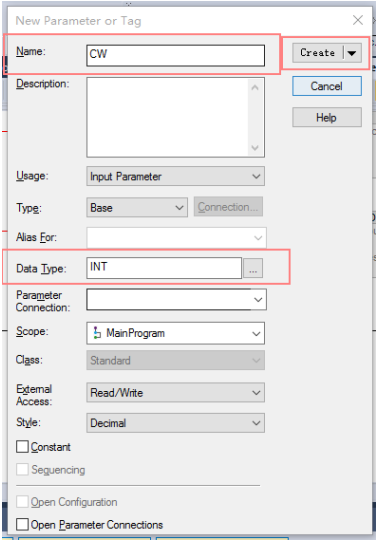
In the "Configure driver" window that pops up, select your computer's network card and click "OK".

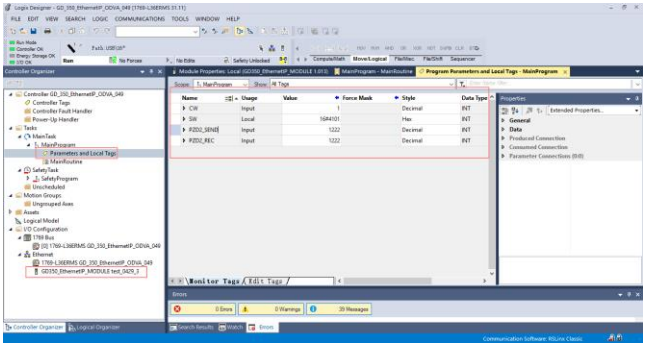


3.5.5 Writing PLC programs

Click on "Tasks"-->"MainTask"-->"MainProgram"--> on the left. Right click on "MainProgram" and "Parameters and Local Tag" above "MainRoutine" to create global variables. Right click "Parameters and Local Tag" above "MainProgram" to create global variables.

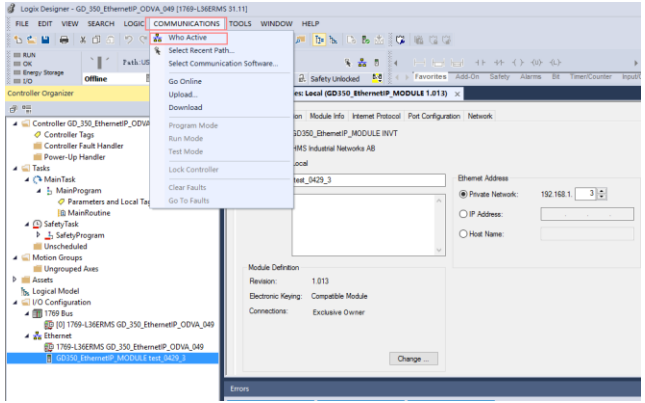

Logix Designer - GD_350_EthernetIP_ODVA_049 [1769-L36ERMS 31.11]*
FILE EDIT VIEW SEARCH LOGIC COMMUNICATIONS TOOLS WINDOW HELP
Controller Organizer
Controller GD_350_EthernetIP_ODVA_049
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Parameters and Local Tags
MainRoutine
Safety Task
SafetyProgram
Unscheduled
Motion Groups
Ungrouped Axes
Assets
Logical Model
I/O Configuration
1769 Bus
[0] 1769-L36ERMS GD_350_EthernetIP_ODVA_049
Ethernet
1769-L36ERMS GD_350_EthernetIP_ODVA_049
GD350_EthernetIP_MODULE test_0429_3
Module Properties: Local (GD350_EthernetIP_MODULE 1.013)
MainProgram - MainRoutine*
New Local Tag... Ctrl+W
New Parameter...
Monitor Tags
Edit Tags
Verify
Export Tags...
Print
MOV
Source CW
Dest test_0429_3:O1.Data[0]
MOV
Source PZD2_SEND
Dest test_0429_3:O1.Data[1]
MOV
Source SW
Dest test_0429_3:I1.Data[0]
MOV
Source PZD2_REC
Dest test_0429_3:I1.Data[1]
Errors
0 Errors
0 Warnings
38 Messages
Controller Organizer
Logical Organizer
Search Results
Watch
Errors
Ready
Communication Software: RSLinx Classic
Rung 1 of 2

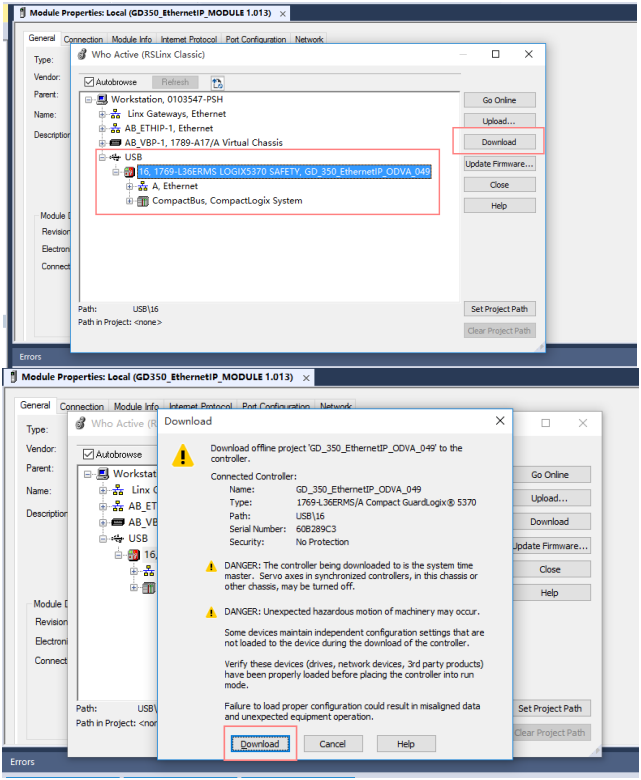

New Parameter or Tag
Name: CW
Create
Description:
Cancel
Help
Usage: Input Parameter
Type: Base
Connection...
Alias For:
Data Type: INT
Parameter Connection:
Scope: MainProgram
Class: Standard
External Access: Read/Write
Style: Decimal
Constant
Sequencing
Open Configuration
Open Parameter Connections



3.5.6 PC connection and program download

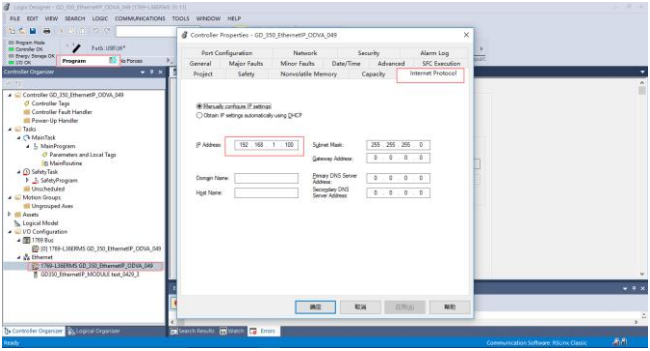
Click on "COMMUNICATIONS" under "Who Active", and in the pop-up screen, click on PLC Project under the "USB" option. "Dowmload". **Note:** The PLC dial code cannot be "RUN" at this time.





3.5.7 Configuring PLC IP Addresses through the studio5000 V31 software

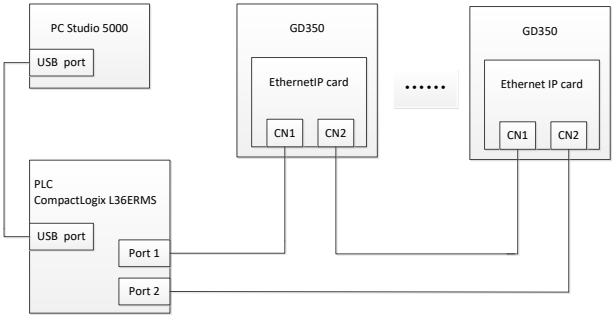
Make sure that the PLC is in REM or PROG mode, click "1769-L36ERMS" at the bottom left to enter the "Controller Properties" interface, and then click "Internet Protocol" to change the IP address of the PLC.



3.5.8 DLR Ring Network Configuration

(1) Using Logix Designer for setup

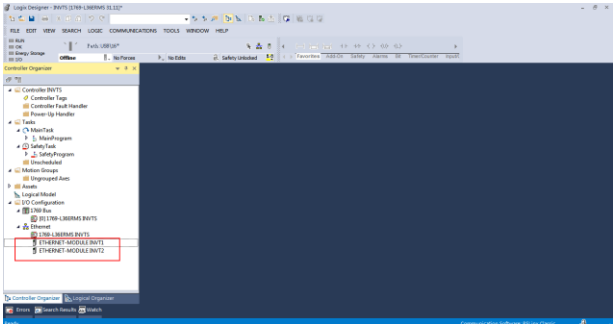
Open the Studio 5000 software and use an Allen-Bradley CompactLogix PLC with ring networking capability, which requires at least two GD350 Ethernet IP communication cards. More GD350 Ethernet IP communication cards can be added, but it is recommended that the maximum number of nodes used on the DLR ring network shall not exceed 32. The connection method is shown in the following figure.



Note: An EDS file must be added.

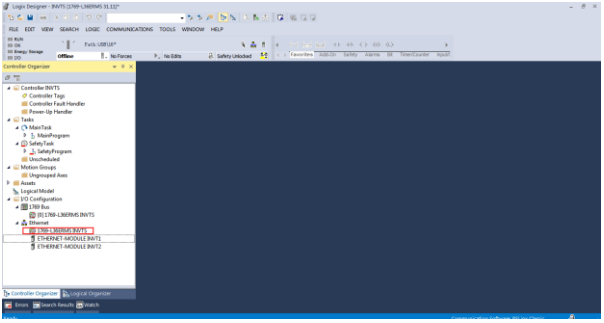
(2) Add an Ethernet IP communication card to the Studio 5000 software

The method of addition is the same as that of the linear star connection.

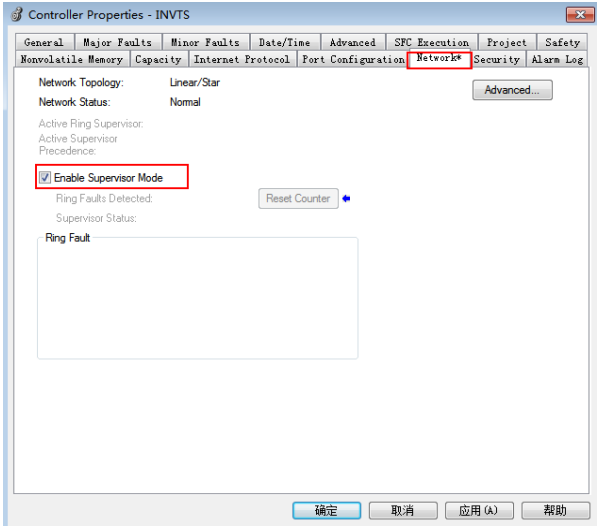


(3) Enabling PLC ring network monitor function

Double click "1769-L36ERMS INVTS" under the "I/O Configuration" folder, as shown in the following figure.

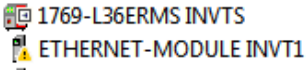


Enter "Network" under the "Controller Properties" option and select "Enable Supervisor Mode".



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

(4) Return to Logix Designer and make sure that none of the communication cards has encountered the following fault.



(5) Download the project to the PLC, bring the PLC online, and put it in programming mode.

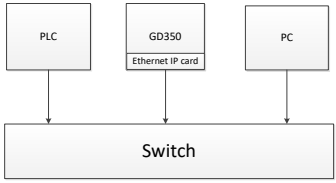
3.6 Example 2 of PLC communication (communicate with ORMON PLC)

This example shows how to use an ORMON PLC (model: NX1P2-9024DT) to communicate with an Ethernet IP adapter module (through using the Sysmac Studio software as the configuration tool).

3.6.1 Hardware connections

The NX1P2-9024DT is not configured with a USB download port, and communication and

download between the PC and PLC is conducted through the built-in Ethernet IP port. In this case, a switch is needed in the experiment, and the connection method is as follows.



3.6.2 Network Configurator software setting

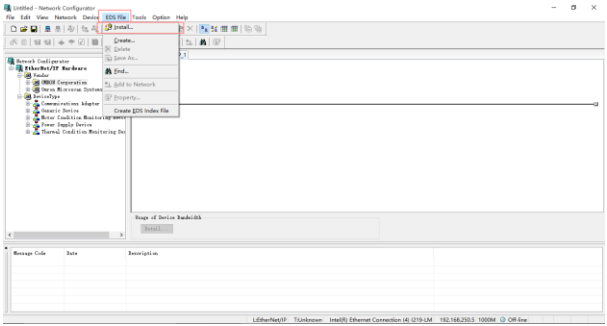
3.6.2.1 Launch Network Configurator software

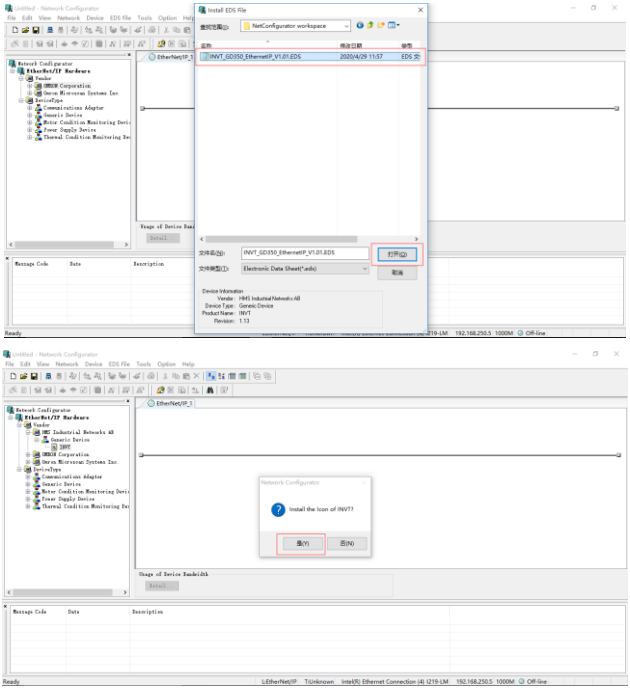
Start the Network Configurator software 

as an administrator in the following directory: "C:\Program Files (x86)\OMRON\CX-One\NetworkConfigurator\Program\NetConfigurator.exe".

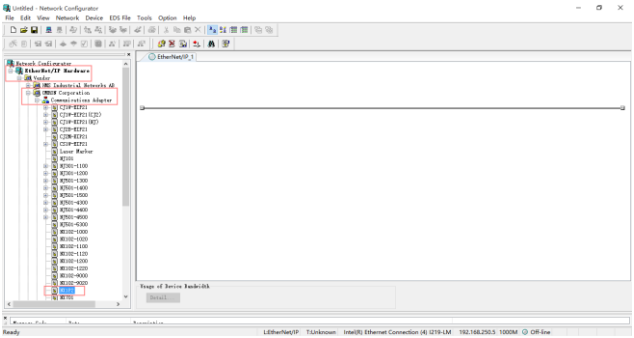
3.6.2.2 Load the EDS file

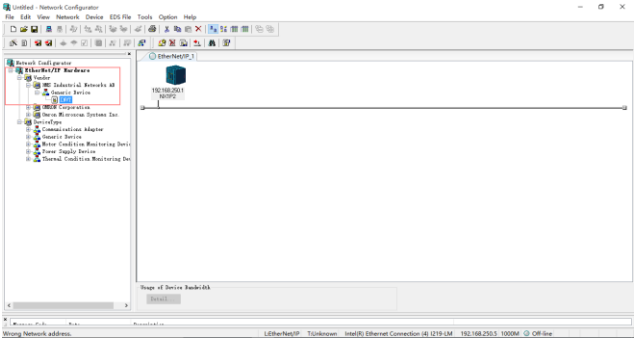
Select "EDS File"->"Install", and add EDS file: INVT_GD350_EthernetIP_V1.01. Click "Open", "Yes", and then click "Cancel".





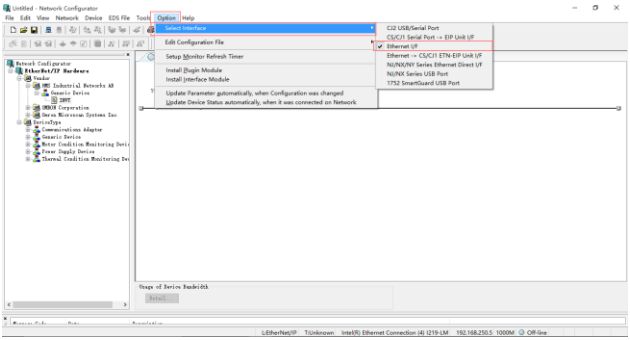
Add "NX1P2" and "INVT" in the following location to the Ethernet IP bus. After these two device are added successfully, the bus shows two devices. The default IP addresses are "192.168. 250.1" and "192.168.250.2", and GD350 function codes P16.02–P16.05 are changed into 192, 168, 250 and 2 respectively.



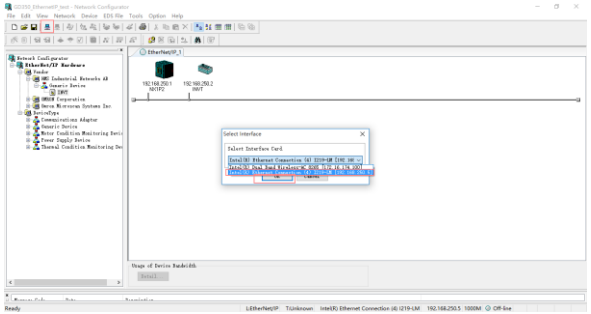


3.6.2.3 Connection setting

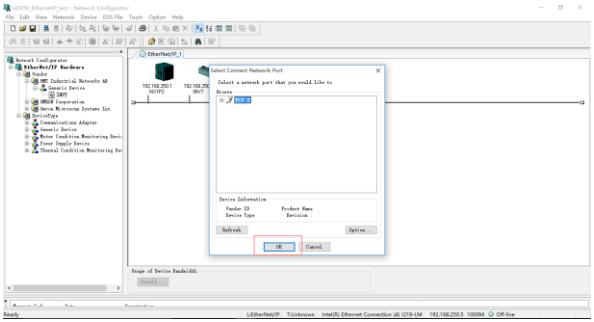
Click "Option" → "Select Interface", and select "Ethernet I/F".



Click the "Connect" icon to select the corresponding network port, and click "OK".

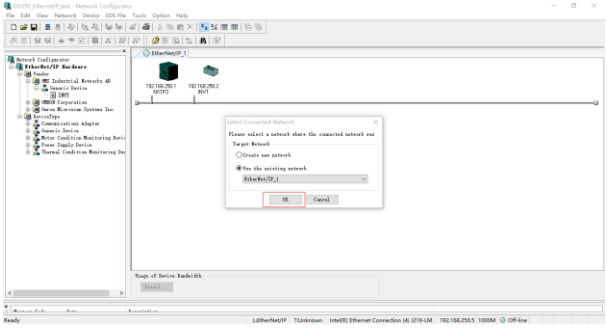


Select "TCP:2", and click "OK".

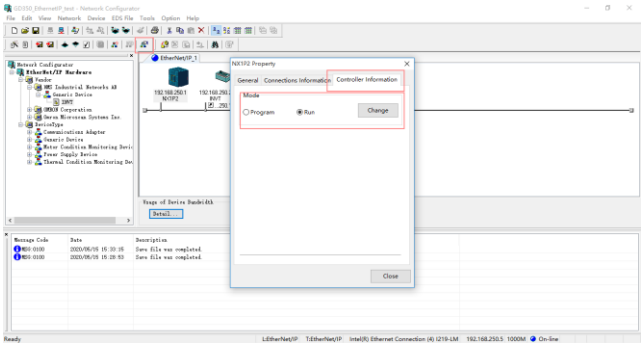


Select "Use the existing network"→"EtherNet/IP_1", click "OK", and the PLC is connected successfully.

After the PLC is connected successfully, the blue indicator above the PLC device icon is on.

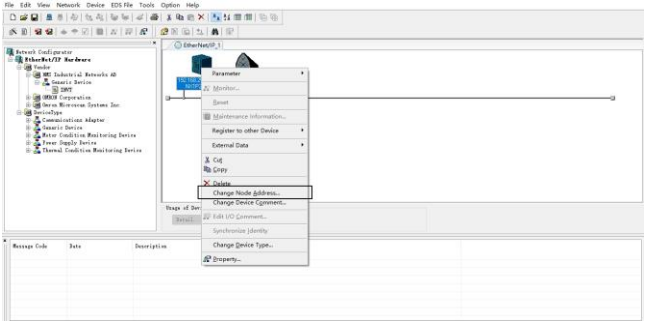


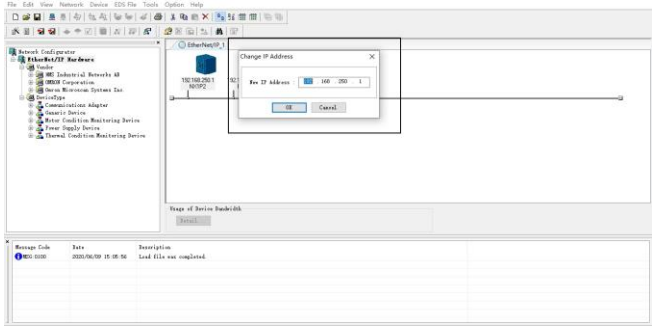
Click the "Device Property" icon, and the "Controller Information" tab pops up. You can switch the PLC status between "Program" and "Run" in the tab.



3.6.2.4 Modify IP address

Right click the device icon and select "Change Node Address" to change the PLC IP address.



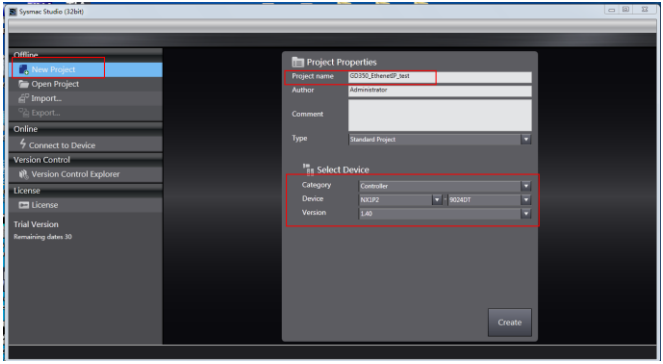


3.6.3 Sysmac Studio software settings

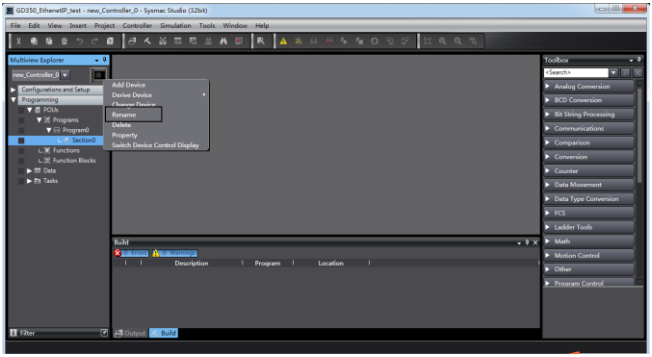
3.6.3.1 Create a new project

Double click the 

icon to open the software, select "New Project", enter "Project name", select the device type, and click "Create".

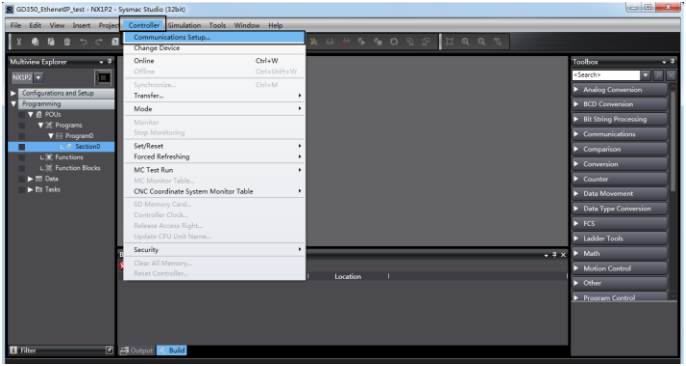


After a new project is created completely, you can enter the following interface. Right click the device icon and select "Rename" to change the device name (you can choose not to change it).



3.6.3.2 Connection setting

Click "Controller" in the menu bar, and select "Communications setup".

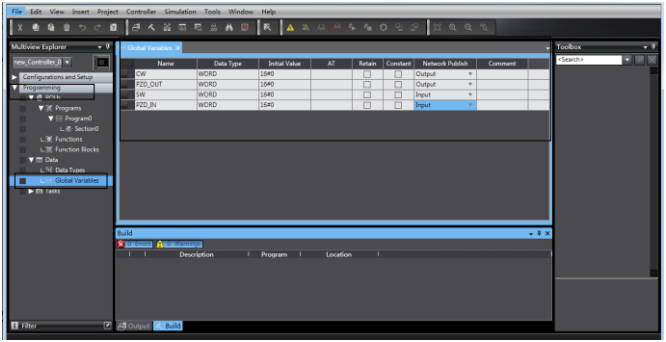


Select "Ethernet-Hub Connection" as the connection method, enter the remote IP address "192.168.250.1", and click "Ethernet communication test". Click "OK" when the status bar shows "Test succeeded".

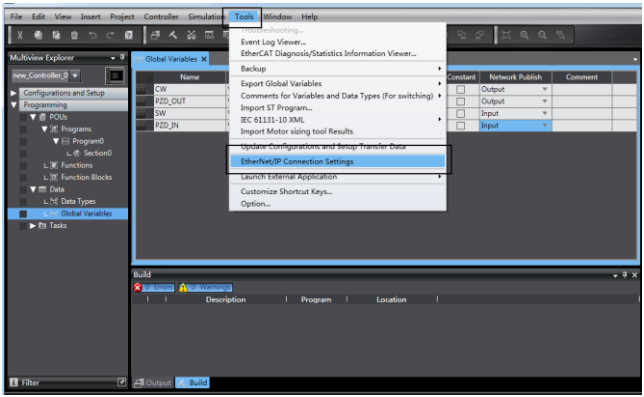
3.6.3.3 Set data labels

Select "Programming" → "Data" → "Global Variables" in the left menu bar, and add global

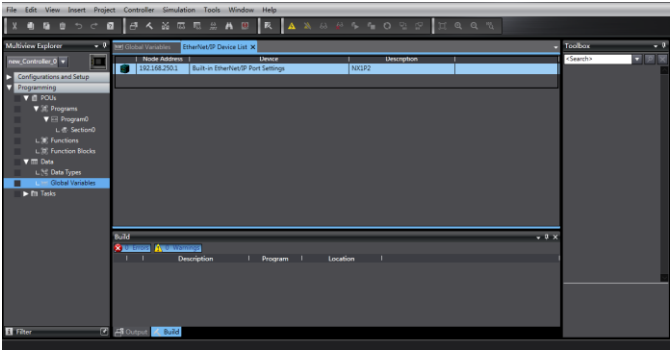
variables as needed. Note that you shall select "WORD" in the "Data Type" column and select "Input/Output" in the "Network Publish" column. Take "ODVA Basic speed control assembly" as an example, and create four global variables.



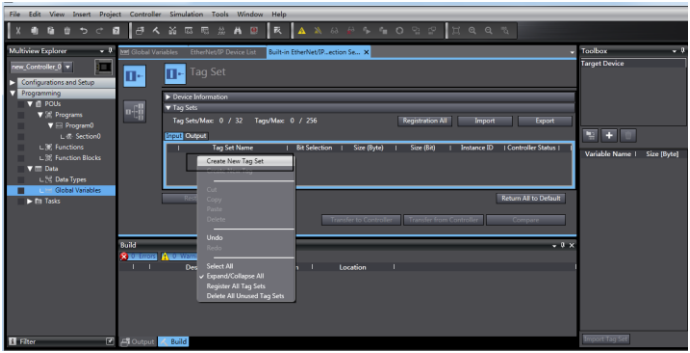
Click "Tools" in the top menu bar, and select "EtherNet/IP Connection Settings".



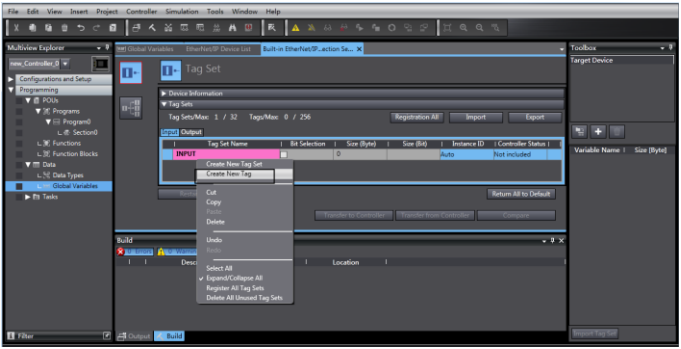
Double click "Built-in EtherNet/IP Port Settings".

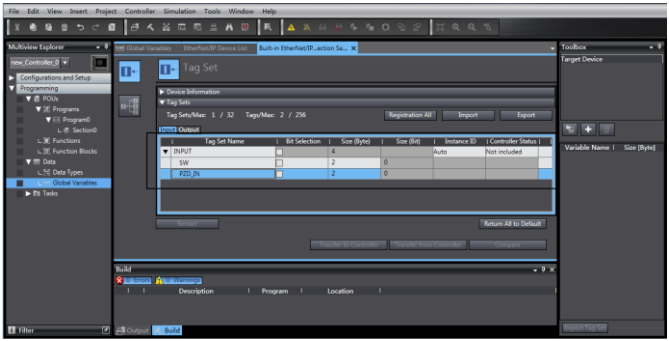


Right click the blank area under "Tag Set", and select "Create New Tag Set".

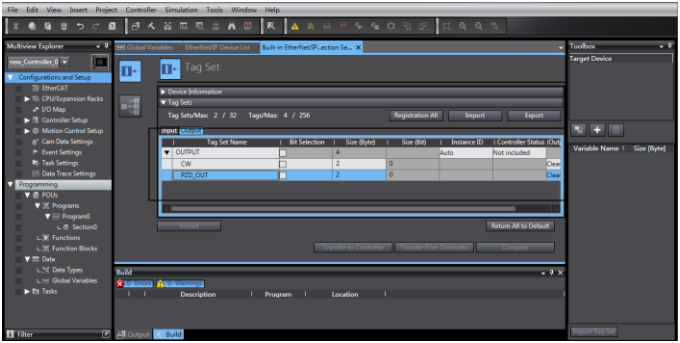


The input tag set is named "INPUT", right click "INPUT" to select "Create New Tag", and add the input global variables to the "INPUT" tag set. Pay attention to the order of the data sequence.





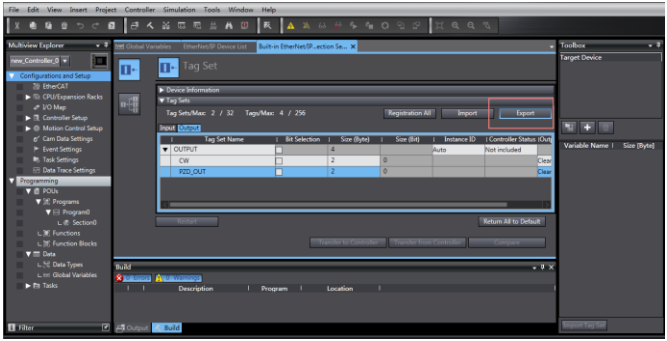
Repeat above steps for "OUTPUT" tag set and "OUTPUT" tag.



3.6.4 Import and export data tags

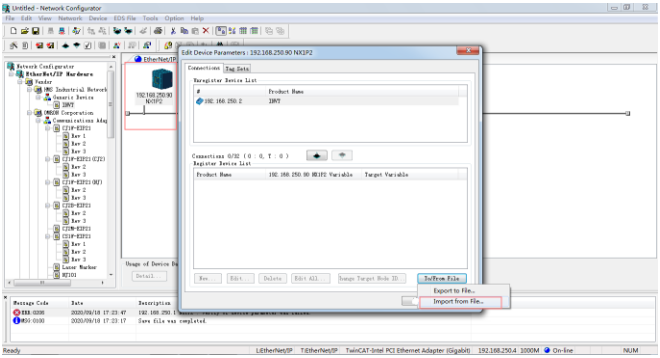
3.6.4.1 Export data tags from Sysmac Studio

After data tags are set completely, click "Export" to export the data tag to a local folder, and save it as "GD350_test.csv" format.

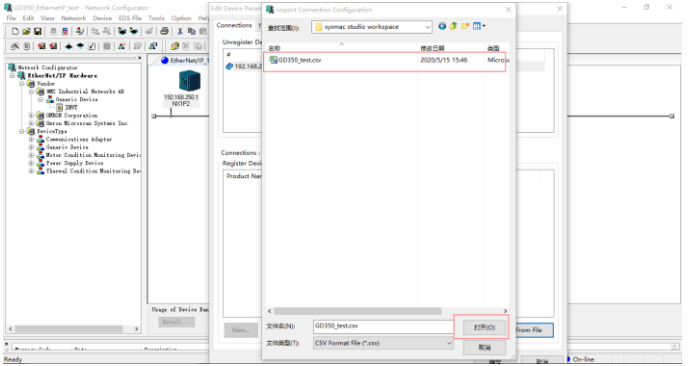


3.6.4.2 Import data tags into Network Configurator

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

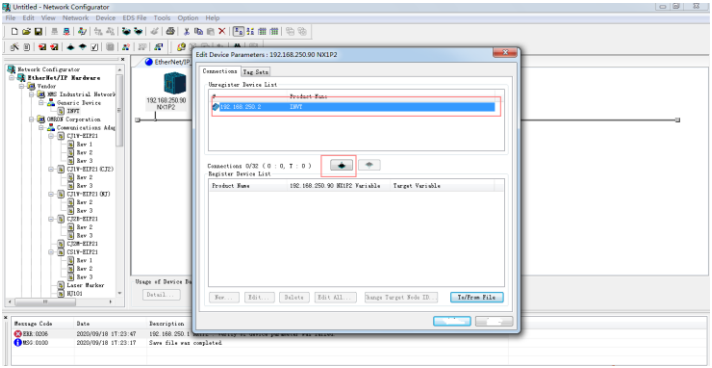


Select the file "GD350_test.csv" exported from Sysmac Studio, and click "Open".

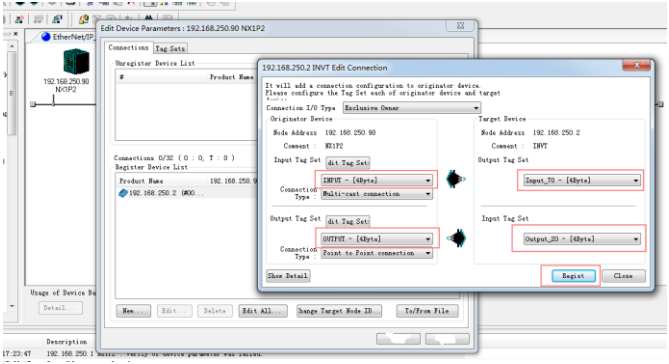


3.6.4.3 Data tag corresponding connection

Select the device "192.168.250.2" under the "Connections" tab, and click the Move Down button.



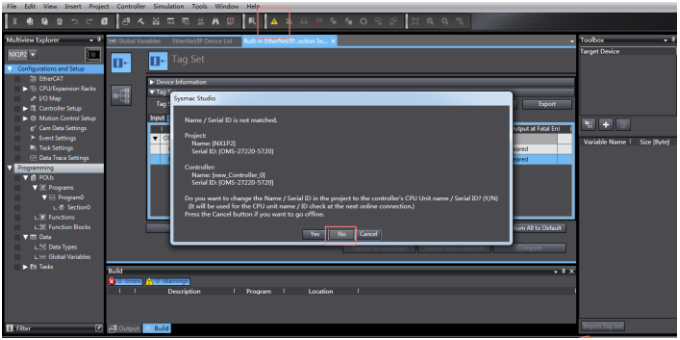
Double click the device "192.168.250.2", set the data input/output tags, and click "Regist".



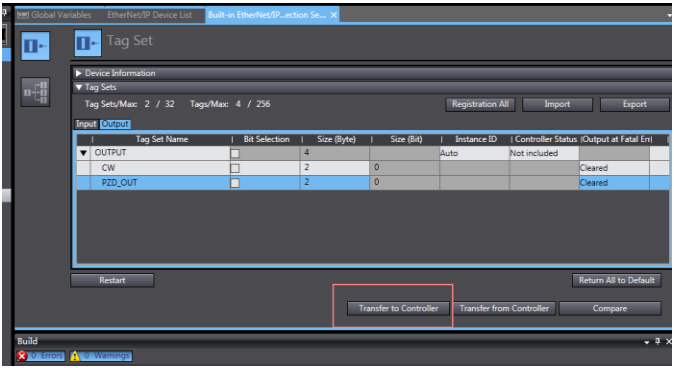
3.6.5 PLC program downloading and online monitoring

3.6.5.1 Sysamc Studio downloading

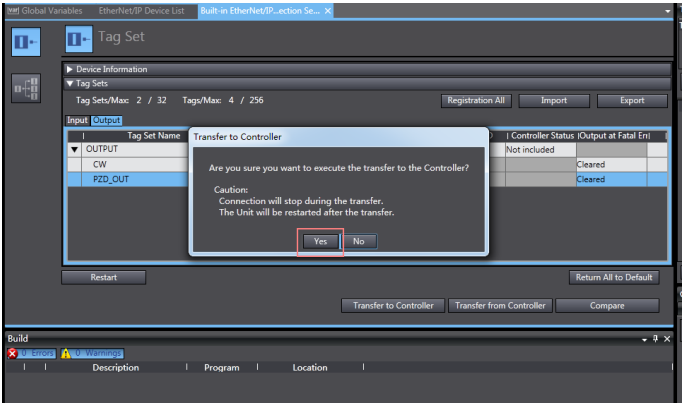
Click the Online button (If the device name has been changed, the following interface will pop up, and you can click "No").



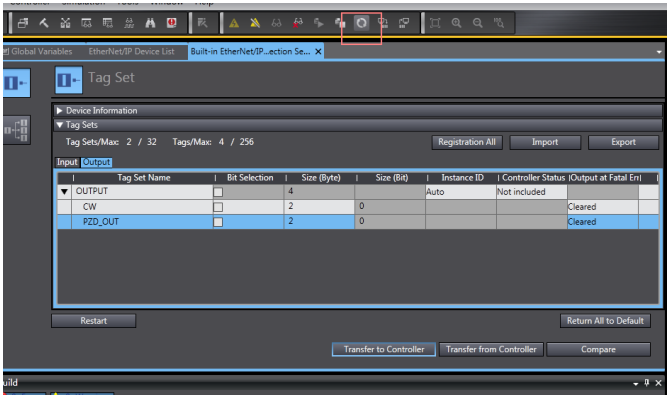
Click "Transfer to Controller" under the " Built-in EtherNet/IP Port Settings" tab.



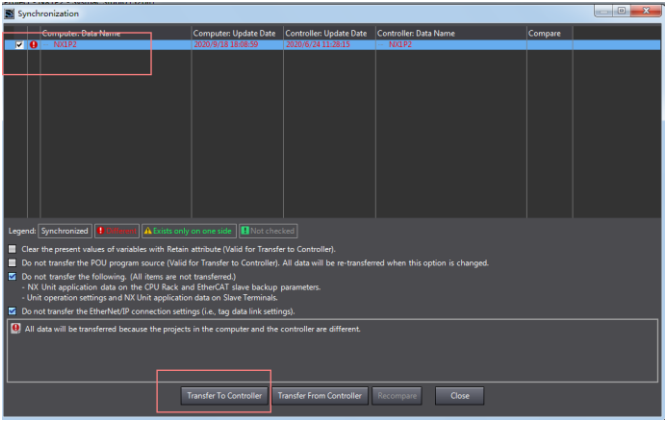
Click "Yes".



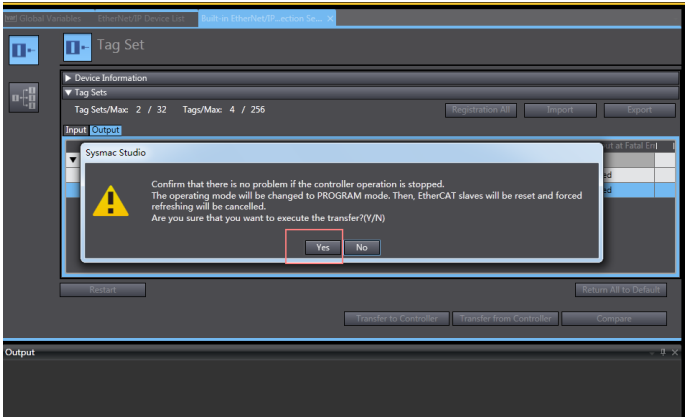
Click the "Sync" function button.



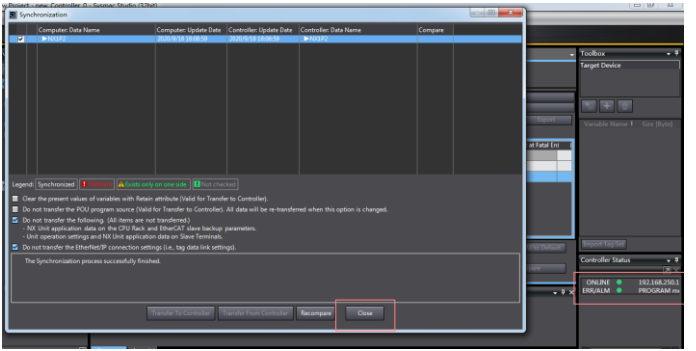
Select the device "NX1P2", and click "Transfer To Controller".



Click "Yes".

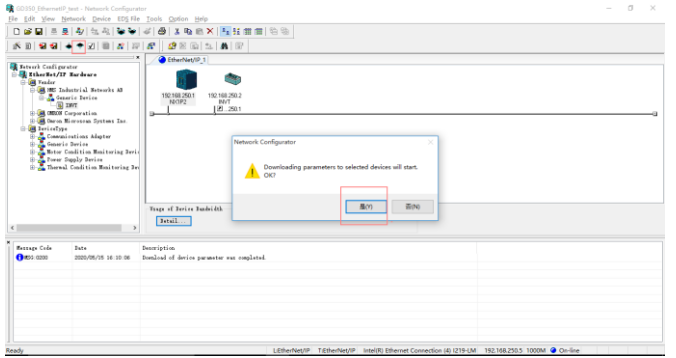


Click "Close" when the "Controller" status in the lower right corner is two green lights.

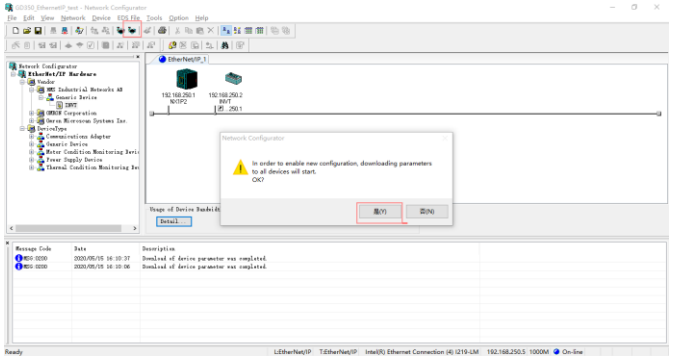


3.6.5.2 Network Configurator downloading

Click the icon of "Download to Device", and click "Yes".

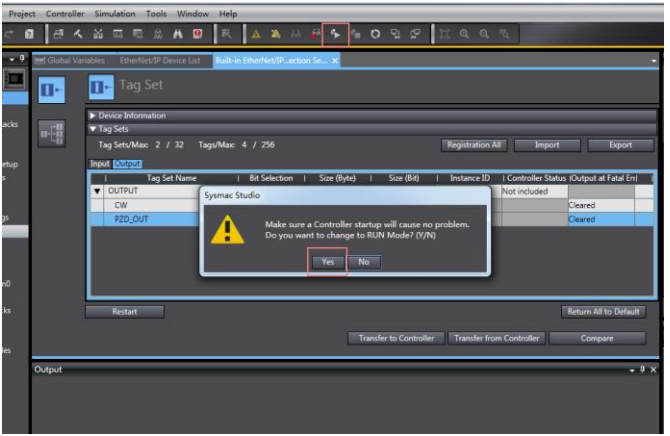


Click the icon of "Download to Network", and click "Yes".

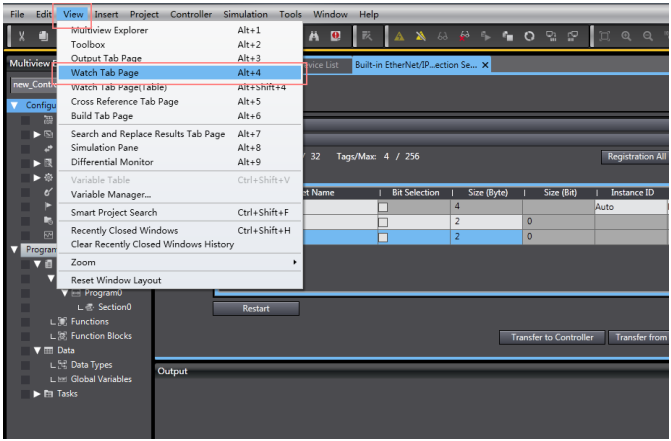


3.6.5.3 Sysamc Studio online monitoring

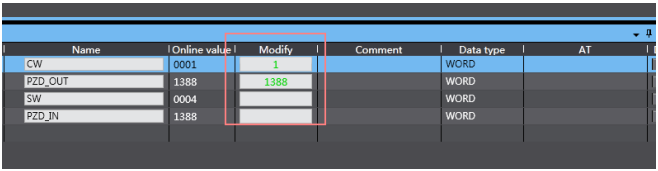
Click the "Run" icon, turn the PLC to "Run Mode", and click "Yes".



Click "View" on the top menu bar, and select "Watch Tab Page".



Enter the variable name in the "Watch Tab Page" to monitor the value of the variable, and change the value in real time in the "Modify" box.



4 EtherCAT communication card

4.1 Overview

1. Thanks for choosing INVT EC-TX508 communication cards. This manual describes the function specifications, installation, basic operation and settings, and information about the EtherCAT protocol. To ensure that you install and operate the product properly, read this manual and the communication protocol section in the VFD operation manual carefully before you use the product.
2. This manual only describes how to operate the EC-TX508 communication card and the related commands but does not provide details about the EtherCAT protocol. For more information about the EtherCAT protocol, read the related specialized articles or books.
3. EC-TX508 communication card is defined as an EtherCAT slave station communication card and is used on a VFD that supports EtherCAT communication.
4. The EtherCAT communication of this communication card supports two types of process data for reading data from and writing data to VFDs. They are PDOs (process data objects) and SDOs (service data objects) for reading data from and writing data to the object dictionary defined by the manufacturer.

4.2 Features

1. Supported functions

- Supports the EtherCAT COE 402 protocol.
- Supports automatic network address setting

2. Supported services

- Supports the PDO service
- Supports the SDO service
- Supports the object dictionary defined by the manufacturer
- Allowing SDOs to read data from and write data to VFD function codes

3. Supported EtherCAT synchronization cycle

Table 4-1 Supported synchronization cycle

Item	Supported specification
Synchronization cycle	250us
	500us

Item	Supported specification
	1ms
	2ms

4. Communication ports

Standard RJ45 ports are used in EtherCAT communication. The communication card provides two RJ45 ports with transmission direction defined. Figure 4-1 shows the ports. IN (indicating input) and OUT (indicating ouput) are EtherCAT wiring network ports. Table 4-2 describes the port pins.

Figure 4-1 RJ45 ports

Table 4-2 RJ45 port pins

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

5. State indicators

The EtherCAT communication card provides four LED indicators and four net port indicators to indicate its states. Table 4-3 describes the state indicators.

Table 4-3 State indicators

Item	Color	Function description
RUN	Green	The green indicator indicates EtherCAT running state. Init state: It remains off. Pre-OP state: It blinks off 0.2s and on 0.2s. Safe-OP state: It blinks off 1s and on 0.2s. OP state: It remains on.
ALM	Red	The red indicator indicates EtherCAT fault state. No fault: It remains off.

Item	Color	Function description
		Init or Pre-OP state: It blinks off 0.2s and on 0.2s. Safe-OP fault state: It blinks off 1s and on 0.2s. OP fault state: It remains on.
PWR	Red	3.3V power indicator
Net port indicator (IN)	Yellow	Off: Indicates that Ethernet connection is not established. On: Indicates that Ethernet connection is established successfully.
	Green	Off: Without connection On: Wilth connection but inactive Blinks: With connection and active
Net port indicator (OUT)	Yellow	Off: Indicates that Ethernet connection is not established. On: Indicates that Ethernet connection is established successfully.
	Green	Off: Without connection On: Wilth connection but inactive Blinks: With connection and active

4.3 Electrical wiring

The EtherCAT network usually consists of a master station (PLC) and several slave stations (drives or bus extension terminals). Each EtherCAT slave station are configured with two standard Ethernet interfaces, and the electrical wiring diagram is shown in Figure 4-2.

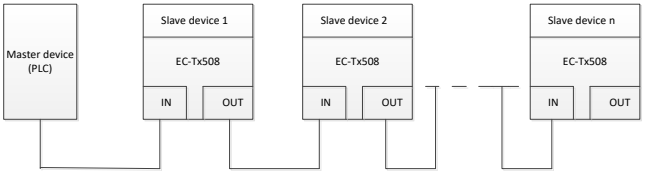


Figure 4-2 Electrical wiring diagram for a linear topology

4.4 Communication

4.4.1 CoE reference model

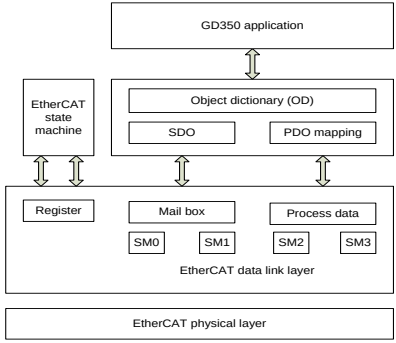


Figure 4-3 CoE reference model

CoE network reference model consists of the data link layer and application layer. The data link layer is responsible for EtherCAT communication protocol. CANopen drive Profile (DS402) communication rules are embedded in the application layer. The object dictionary in CoE includes the parameters, application data, and PDO mapping configuration information.

PDOs are composed of the objects (in the object dictionary) that can perform PDO mapping. The content in PDO data is defined by PDO mapping. PDO data is periodically read and written, which does not require searching the object dictionary. Mail box communication (SDO) is not periodic, which requires searching the object dictionary.

Note: To parse SDO and PDO data correctly on the EtherCAT data link layer, it is necessary to configure FMMU and Sync Manager (SM).

Table 4-4 EtherCAT Sync Manager configuration

Synchronization management	Configuration	Size	Start address
Sync Manager 0	Assigned to receive SDO	512byte	0x1000
Sync Manager 1	Assigned to send SDO	512byte	0x1400
Sync Manager 2	Assigned to receive PDO	128byte	0x1800

Synchronization management	Configuration	Size	Start address
Sync Manager 3	Assigned to send PDO	128byte	0x1C00

4.4.2 EtherCAT slave station information

EtherCAT slave station information file (.xml) is read by the master station to construct the master and slave station configuration. This file contains mandatory information about EtherCAT communication settings. INVT provides this file EC-TX508_100.xml.

4.4.3 EtherCAT state machine

EtherCAT state machine is used to describe the states and state change of slave station applications. Generally, the master station sends a state change request, while the slave station responds. The state change flow is shown in the following figure.

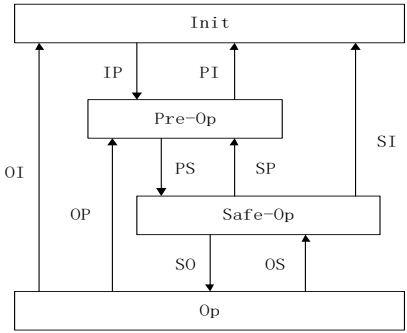


Figure 4-4 EtherCAT state machine flowchart

Table 4-5 EtherCAT state machine description

State	Description
Init	Both SDO and PDO communication are unavailable.
Init to Pre-Op	The master station configures the data link layer address and SM channel for SDO communication. The master station initializes DC synchronization information. The master station requests the jump to the Pre-Op state. The master station configures the application layer control register. The slave station checks whether the mailbox is initialized properly.

State	Description
Pre-Op	SDO communication is available but PDO is unavailable.
Pre-Op to Safe-Op	The master station configures the SM and FMMU channels for PDO communication. The main station configures PDO mapping through SDO communication. The master station requests the jump to the Safe-Op state. The slave station checks whether the PDO and DC are configured correctly.
Safe-Op	SDO communication is available. Communication of receiving PDOs is available, but that of sending PDOs is unavailable, in the Safe state.
Safe-Op to Op	The master station requests the jump to the Op state.
Op	Both SDO and PDO communication are available.

4.4.4 PDO mapping

The process data of an EtherCAT slave station is composed of SM channel objects. Each SM channel object describes the consistent area of the EtherCAT process data and includes multiple PDOs. An EtherCAT slave station with the application control function shall support PDO mapping and reading of SM PDO assigned objects.

The master station can select objects from the object dictionary to perform PDO mapping. PDO mapping configuration is located in the range of 1600h–1603h (RxPDOs: receiving PDOs) and range of 1A00h–1A03h (TxPDOs: sending PDOs) in the object dictionary. The PDO mapping method is shown in the following figure.

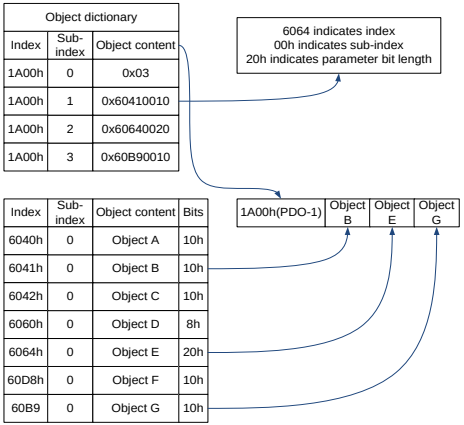


Figure 4-5 PDO mapping method

In addition to PDO mapping, EtherCAT process data switching needs to assign PDOs to SM channels. The relationship between PDOs and SM channels is established through SM PDO assigned objects (1C12h and 1C13h). The mapping between SM channels and PDOs is shown in the following figure.

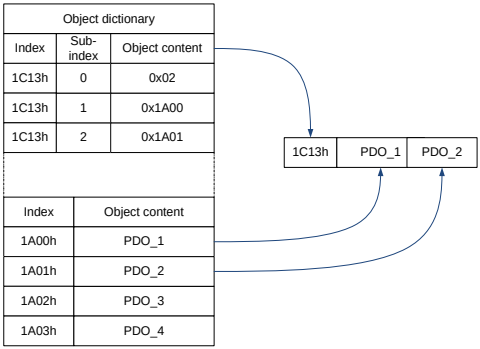


Figure 4-6 SM PDO assignment

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 DC-based network synchronization

The DC (distributed clock) can enable all EtherCAT devices to use the same system time so as to control the synchronous execution of all device tasks. In the EtherCAT network, the clock with the DC function of the first slave station connected to the master station is used as the reference clock across the network. The other slave stations and master station use this reference clock for synchronization.

Free-Run: The running cycle and communication cycle of each servo drive are not related to the communication cycle of the master station.

DC Mode: The servo drive performs synchronization through Sync0 of the master station.

4.5 CiA402 device protocol

The master station controls the drive through the control word (0x6040) and obtains the current state of the drive by reading the status word (0x6041). The servo drive implements motor control based on master station control commands.

4.5.1 CoE state machine

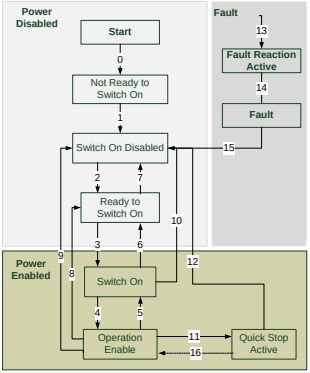


Figure 4-7 CANopen over EtherCAT state machine

Status	Description
Not Ready to Switch On	The drive is in the initialization process.
Switch On Disabled	Drive initialization completes.
Ready to Switch On	The drive is preparing to enter the Switch On state, but the motor is not excited.
Switched On	The drive is in the ready state, and the main circuit power supply is normal.
Operation Enable	The drive is enabled and controls the motor based on the control mode.
Quick Stop Active	The drive stops in the set manner.
Fault Reaction Active	When detecting an alarm, the drive stops in the set manner, but the motor still has the exciting signal.
Fault	The drive is in the faulty state, and the motor has no exciting signal.

6040h control word includes:

1. Bit for status control;

2. Bit related to control mode;

3. Factory-defined control bit.

The bits of 6040h are described as follows.

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

BITS 0–3 AND 7 (used for status 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 (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	Abs/rel	Reserved	Reserved
8	Halt	Halt	Halt

Control word is set to 0x0F for enabling the drive. Otherwise, the drive will stop. When a fault occurs, if bit 7 of the control word is set to 1, the reset command is enabled.

6041h status word includes:

1. Current status bit of drive;

2. Status bit related to control mode;

3. Factory-defined status bit.

The bits of 6041h are described 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	Operation mode specific	O
14-15	Manufacturer specific	O

BIT0–3, 5, AND6:

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

BIT4: Voltage enable, when this bit is 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 the OP state, and the master station can control the drive through PDO.

BIT10: Target reached, this bit differs in meaning under different control modes. When this bit is 1, in cyclic synchronous position mode, it indicates that target position is reached, while in cyclic synchronous velocity mode, it indicates that reference speed is reached; in homing mode, it indicates that homing is completed.

BIT14: When this bit is 1, it indicates motor zero-speed state.

BIT7–8, BIT11–13, and BIT15: Reserved.

4.5.2 Device running mode

Set P00.01=2 (command running channel), P00.02=3 (EtherCAT communication channel), and P16.18 Communication timeout. Position mode and homing mode are achieved only when the drive is in the closed-loop mode.

6.5.2.1 Cyclic Synchronous Position Mode

1. Set 【6060h: Mode of operations】 to 8 (Cyclic synchronous position mode).

2. Set 【6040h: Control word】 to enable the drive (set it to 0x0F for enabling).

3. Set 【607Ah: Target position】 to the target position (unit: user unit).

4. Query for 【6064h: Position actual value】 to obtain actual position feedback of the motor.

5. Query for 【6041h: Status word】 to obtain the status feedback of the drive (following error, target reached and internal limit active).

6. For function details, see function parameters in group P21 in GD350 and the specific function commissioning instructions in the basic operations in the manual.

6.5.2.2 Profile Position Mode

1. Set 【6060h: Mode of operations】 to 1 (Profile Position Mode).

2. Set tens of P21.00 (Position command source) to 1 (Digital position), and set P21.16 (Digital positioning mode).

3. Set 【6040h: Control word】 to enable the drive (set it to 0x0F for enabling).

4. Set 【607Ah: Target position】 to the target position (unit: user unit).

5. Query for 【6064h: Position actual value】 to obtain actual position feedback of the motor.

6. Query for 【6041h: Status word】 to obtain the status feedback of the drive (following error, target reached and internal limit active).

7. For function details, see function parameters in group P21 in GD350 and the specific function commissioning instructions in the basic operations in the manual.

6.5.2.3 Homing Mode

1. Set 【6060h: Mode of operations】 to 6 (homing mode).

2. Set P22.00.Bit0=1 to enable the spindle positioning, and set P22.03–P22.06.

3. Set 【6040h: Control word】 to enable the drive (set it to 0x0F for enabling). Homing operation start (Bit4) changes from 0 to 1 (Control word Bit4 is set to 1). However, the change from 1 to 0 will terminate Homing.

4. The motor queries the limit switch and Home switch to complete Homing.

5. Query 【6041h: Status word】 to obtain the status feedback of the drive (Homing error, Homing attained, and Target reached).

6. For function details, see function parameters in group P22 in GD350 and the specific function commissioning instructions in the basic operations in the manual.

6.5.2.4 Cyclic Synchronous Velocity Mode

1. Set 【6060h: Mode of operations】 to 9 (Cyclic synchronous velocity mode).

2. Set 【6040h: Control word】 to enable the drive (set it to 0x0F for enabling) and start the motor for running.

3. Set【60FFh: Target velocity】to set the target rotation speed (unit: rpm), which corresponds to P00.10 (a positive value indicates forward rotation and a negative value indicates reverse rotation).

4. Query【6041h: Status word】 to obtain the status feedback of the drive (Speed zero, Max slippage error, Target reached, and Internal limit active).

6.5.2.5 Cyclic Synchronous torque Mode

1. Set 【6060h: Mode of operations】 to 10 (Cyclic synchronous torque mode).

2. Set the VFD to torque control (P03.32=1).

3. Set 【6040h: Control word】 to enable the drive (set it to 0x0F for enabling) and start the motor for running.

4. Set 【6071h: Target torque】 to set the target torque.

5. Query【6041h: Status word】 to obtain the status feedback of the drive (Speed zero, Max slippage error, Target reached, and Internal limit active).

4.6 Function code modification

Index	Sub-index	Description	Permission	Data type	Default
2000h	0	Read function codes	RW	UINT32	0
Bits 16–31: Read function code addresses Bits 00–15: No function Example of read operation: Read the keypad-set frequency (value of P00.10) SDO operates 2000h to write 0x00A00000. View the response value of 2001h read operation.					
2001h	0	Read response	RO	UINT32	0
Bits 16–31: 0x0001 read success Bits 00–15: Parameter value read by 2000h Bits 16–31: 0x0003 read error Bits 00–15: Error codes 0x0002 illegal data address 0x0009 password protection					
2002h	0	Write function codes	RW	UINT32	0
Bits 16–31: Write function codes Bits 00–15: Written data Write operation example: Modify the value of P00.10 to 50.00. SDO operates 2002h to write 0x000A1388. View the response value of 2003h write operation.					
2003h	0	Write response	RO	UINT32	0
Bits 16–31: 0x0001 write success Bits 00–15: Parameter value written by 2002h Bits 16–31: 0x0003 write error Bits 00–15: Error codes 0x0002 illegal data address 0x0003 illegal data value 0x0007 read-only parameter 0x0008 the parameter is unchangeable during running					

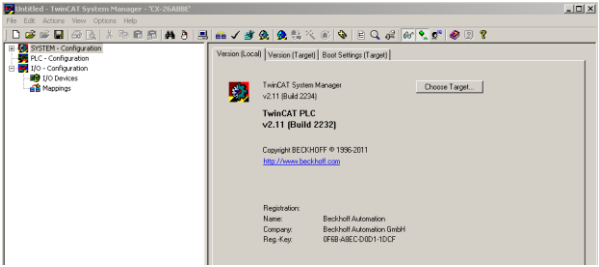
4.7 Example of TwinCAT2 application

This example shows how to use TwinCAT2 as the main station to communicate with the EtherCAT module of the VFD.

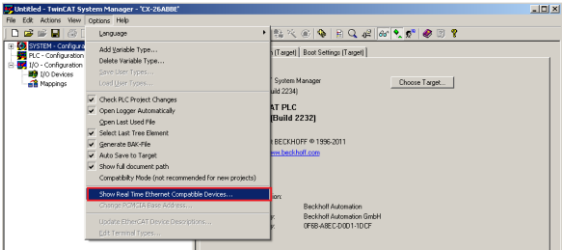
1. Install TwinCAT2 software

2. Copy the EtherCAT configuration file (EC-TX508_100.xml) of GD350 to the installation directory of TwinCAT2 ("C:\TwinCAT\Io\EtherCAT").

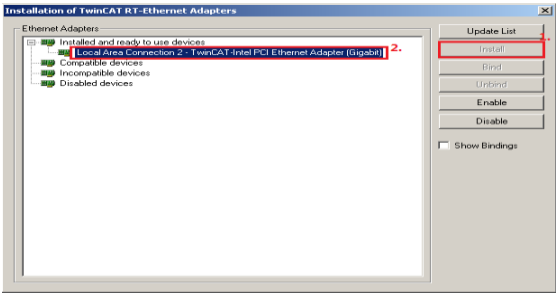
3. Open TwinCAT2



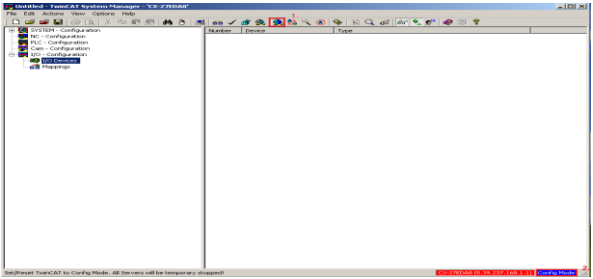
4. Install the network card drive



Open the menu as shown in the above figure, select "Show Realtime Ethernet Compatible Devices…", pop up a dialog box as shown in the following figure, select the local area network card, and click "Install". After the network card is installed successfully, it will shown under the menu "Installed and ready to use devices". (Note: Please choose the network cards configured with Intel chips)

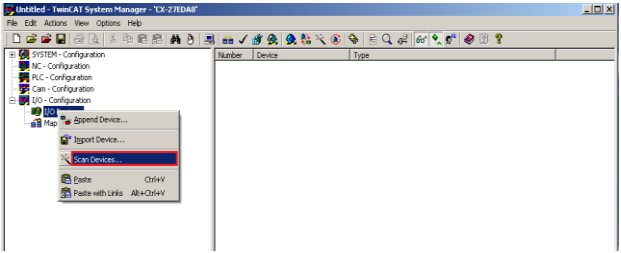


5. Set TwinCAT2 to be in the configuration mode

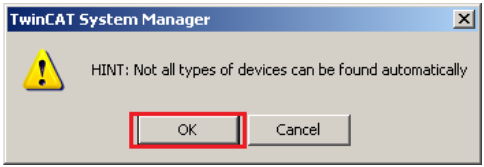


6. Scan device

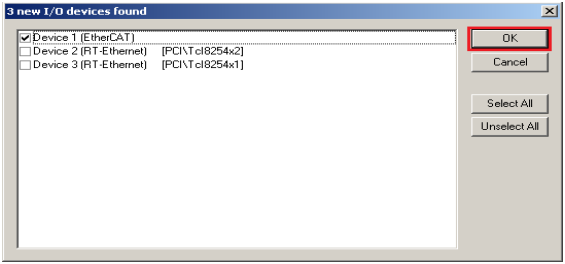
Select "I/O Devices" menu, and right-click to select "Scan Devices…" to scan the device.



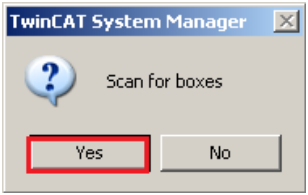
Pop up the following dialog box, and select "OK".



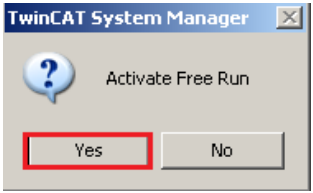
Pop up the following dialog box, and select "OK".



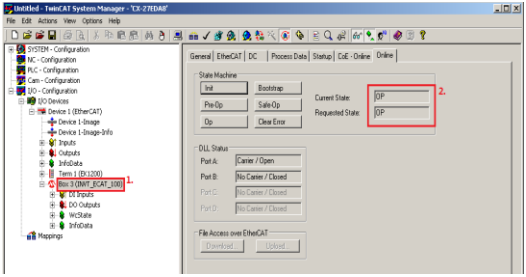
Pop up the following dialog box, and select "Yes".



Pop up the following dialog box, and select "Yes". Then the device enters the free running mode.

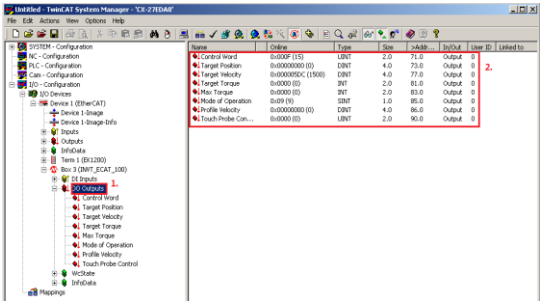


The following figure shows "Box3" which is the slave device scanned, and view that the device enters the "OP" state.

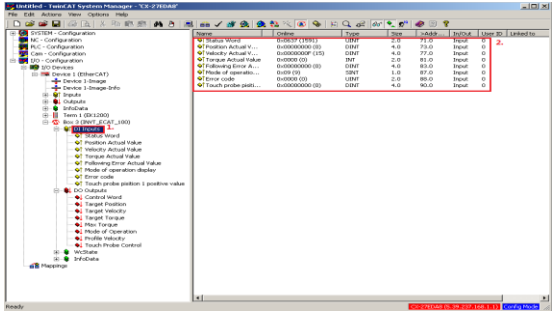


7. Process data input and output

Select "DO Outputs" menu, and there are data sent from the master station to the VFD, which can be used to set commands and rotation speed.

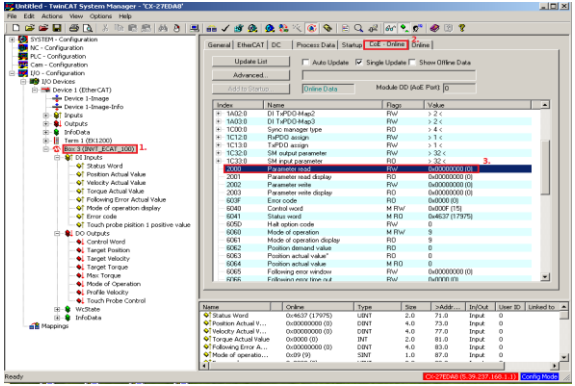


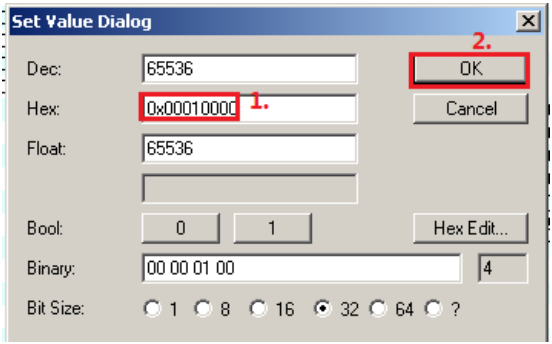
Select "DI Intputs" menu, and there are data sent from the VFD to the master station, which can be used to return the statuses and and rotation speed.



8. SDO data operation

Select "CoE–Online" menu, as shown in the figure below. Read the VFD function code parameters through index 0x2000, and double click 0x2000 to pop up a dialog box. Write the parameter address in the dialog box, and click "OK". The returned results are stored in index 0x2001. Similarly, Write the VFD function code parameters through index 0x2002, and the written results are stored in index 0x2003.




Set Value Dialog
Dec:
65536
Hex:
0x00010000
1.
Float:
65536
Bool:
0
1
Binary:
00 00 01 00
4
Bit Size:
1 8 16 32 64 ?
2.
OK
Cancel
Hex Edit...

5 Modbus TCP communication card

5.1 Overview

1. Thanks for choosing INVT Modbus TCP communication cards. This manual describes the function specifications, installation, basic operation and settings, and information about the network protocol. To ensure that you install and operate the product properly, read this manual and the communication protocol section in the VFD operation manual carefully before you use the product.
2. This manual only describes how to operate the Modbus TCP communication card and the related commands but does not provide details about the Modbus TCP protocol. For more information about the Modbus TCP protocol, read the related specialized articles or books.
3. This communication card is defined as a Modbus TCP slave station communication card and is used on a VFD that supports Modbus TCP communication.
4. The communication card supports the star-shaped network topology and linear network topology.
5. The communication card supports 32 inputs/outputs to read and write process data, read state data, and read and write function parameters of a VFD.

5.2 Features

1. Supported functions

- Supports the Modbus TCP protocol and Modbus TCP slave stations.
- Provides two Modbus TCP ports and supports the 10/100M full/half-duplex operation
- Supports the star-shaped network topology and linear network topology.

2. Supported communication types

Modbus TCP uses TCP/IP for information control and transmission over the Ethernet, allowing the sending of explicit packets, namely, point-to-point messages that are not time critical. The Modbus TCP application layer adopts the Modbus protocol, which is also used by Modbus RTU.

Same as Modbus RTU, Modbus TCP requires the PLC/PC to send the read or write commands, and the communication card returns the operation result after data forwarding to complete the data transmission.

3. Communication ports

Standard RJ45 ports are used in Modbus TCP communication. The communication card provides two RJ45 ports with no transmission direction defined, and therefore you can insert a cable into the port without regard to its direction. Figure 5-1 shows the ports, and Table 5-1 describes the functions of the ports.

Figure 5-1 Two standard RJ45 ports

Table 5-1 Standard RJ45 port pins

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

4. State indicators

The Modbus TCP communication card provides4 LED indicators and 4 network port indicators to indicate its states. Table 5-2 describes the state indicators.

Table 5-2 State indicators

LED	Color	State	Description
LED1	Green	On	Indicating that the card and VFD identify each other.
		Blinking (1Hz)	Indicating that the card and VFD communicate normally.
		Off	Indicating that the card and VFD communicate improperly.
LED2	Green	On	The communication between the card and PLC is online and data interchange is allowed.

LED	Color	State	Description
		Blinking (1Hz)	Indicating IP address conflict between the card and PLC.
		Off	Indicating that communication between the card and PLC is offline.
LED3	Red	On	Modbus TCP has not received valid data.
		Blinking (1Hz)	Indicating that the packet address is unused or undefined.
		Blinking (8Hz)	Indicating incorrect packet address.
		Off	No fault
LED4	Red	On	3.3V power indicator
Network port indicator	Yellow	On	Link indicator, indicating successful Ethernet connection.
		Off	Link indicator, indicating that Ethernet connection is not established.
Network port indicator	Green	On	ACK indicator, indicating that data interchange being performed.
		Off	ACK indicator, indicating that data interchange is not be performed.

5.3 Electrical wiring

The Modbus TCP communication card provides standard RJ45 ports and supports the linear and star topologies. Figure 5-2 and Figure 5-3 show the electrical wiring diagrams for different topologies.

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

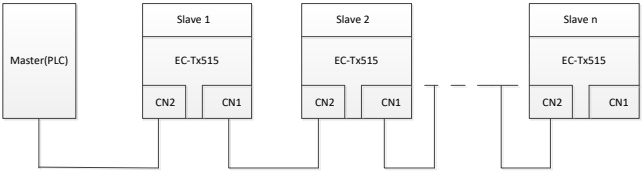


Figure 5-2 Electrical wiring diagram for a linear topology

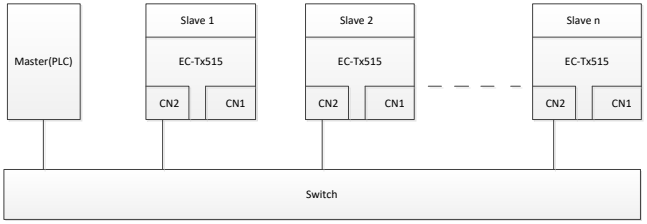


Figure 5-3 Electrical wiring diagram for a star topology

Note: An Ethernet switch must be available when the star topology is used.

5.4 Communication

5.4.1 Communication settings

The Modbus TCP communication card can function as only the Modbus TCP slave station. Before communication, set Goodrive350 function codes, including:

1. Communication station address, IP address and subnet mask for the card

The default station address, IP address, and subnet mask for each communication card are 1, 192.168.0.1, and 255.255.255.0 respectively. You can change them to the address of a network segment.

2. Control mode

If you want to control the VFD with the communication card, set the control mode to Modbus TCP communication control. To be specific, set P00.01=2 (communication as the running command channel) and set P00.02=0 (Modbus TCP communication channel) to control VFD start and stop. If you want to set a value through Modbus TCP communication, change the control way of corresponding function codes to Modbus TCP communication. Appendix B lists related function codes.

Note: After the setting, the card can communicate normally. If you want to control the VFD with the card, set related function codes to enable Modbus TCP communication control.

5.4.2 Packet format

Table 5-3 describes the structure of a TCP communication packet.

Table 5-3 Structure of a TCP communication packet

MAC-layer packet header	IP-layer packet header	TCP-layer packet header	Valid data	Packet trailer
14 bytes	20 bytes	20 bytes	0–1488 bytes	4 bytes

5.4.3 Modbus TCP communication

The application layer of the Modbus TCP communication card supports the Modbus protocol. The Modbus TCP protocol packet is located in the valid data area of the TCP communication packet. It consists of two parts. The first part is MBAP (packet header, occupying 7 bytes), and the second part is PDU (protocol data unit whose length is variable), as shown in Table 5.4.

Table 5-4 Modbus TCP protocol packet

MBAP				PDU	
Transaction identifier	Protocol identifier	Length field	Unit identifier	Function Code	Data
2 bytes	2 bytes	2 bytes	1 byte	1 byte	n bytes
Sequence number of packets, incremented by 1 after each communication for distinguishing between different packets	0000=Modbus-TCP protocol	Data length	Device address (station number)	Modbus function code	Includes VFD function codes and data, and the length is variable.

Through the above packets, you can set the VFD reference parameters, monitor the status value, send control commands, monitor the running status of the VFD, and read and write the VFD function codes. For specific operations, see the follow-up.

Parameter description:

Unit identifier: Slave station number (1–247).

Function code: Modbus function codes, as shown in Table 5.5.

Table 5-5 Modbus function codes

Function code	Description
0x01	Read coils
0x05	Write single coil
0x0F	Write multiple coils
0x02	Read discrete quantity
0x04	Read input register

Function code	Description
0x03	Read holding register
0x06	Write single holding register
0x10	Write multiple holding registers

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

Packet examples:

(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 command is used to read parameters and operation status of the VFD.

For example, starting from the data address of 0004H, to read two contiguous pieces of data (that is, to read content from the data addresses 0004H and 0005H) of the VFD whose address is 01H, the frame structures are described in the following.

Example	Request	0001	0000	0006	01	03	0004	0004
	Meaning	MBAP				Function code	Write address	Number of bytes
	Response	0001	0000	0007	01	03	04	1388 0000
	Meaning	MBAP				Function code	Number of bytes	Data

From the response, the data in 0004H is 1388H (50.00Hz), and that in 0005H is 0000H (00.00Hz).

(2) Command code 06H, writing one 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, to write 5000 (1388H) to 0004H of the VFD whose address is 02H, the frame structures are described in the following.

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) Command 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, to write 5000 (1388H) and 50 (0032H) respectively to 0004H and 0005H of the VFD whose slave address is 02H, the frame structures are described in the following.

Example	Request	0001	0000	000B	02	10	0004	0002	04	1388 0032
	Meaning	MBAP				Function code	Write address	Number of registers	Number of bytes	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 status information, and setting function parameters of the VFD.

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

Function code	Name	Parameter description	Setting range	Default value
P14.00	Local communication address	1–247	1–247	1

Function code	Name	Parameter description	Setting range	Default value
P14.03	Communication response delay	0–200ms	0–200	5

Note:

- 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 in 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.

Description of other function addresses

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 describes other function parameters.

Function	Address	Data description	R/W
Communication-based control command	2000H	0001H: Forward running	R/W
		0002H: Reverse running	
		0003H: Forward jogging	
		0004H: Reverse jogging	
		0005H: Stop	
		0006H: Coast to stop (emergency stop)	
		0007H: Fault reset	
		0008H: Jogging to stop	
Communication-based value setting	2001H	Communication-based frequency setting (0–Fmax, unit: 0.01 Hz)	R/W
	2002H	PID setting, range (0–1000, 1000 corresponding to 100.0%)	R/W
	2003H	PID feedback, range (0–1000, 1000 corresponding to 100.0%)	R/W

Function	Address	Data description	R/W
	2004H	Torque setting (-3000–+3000, 1000 corresponding to 100.0% of the motor rated current)	R/W
	2005H	Setting of the upper limit of the forward running frequency (0–Fmax, unit: 0.01 Hz)	R/W
	2006H	Setting of the upper limit of the reverse running frequency (0–Fmax, unit: 0.01 Hz)	R/W
	2007H	Upper limit of the electromotion torque (0–3000, 1000 corresponding to 100.0% of the motor rated current)	R/W
	2008H	Upper limit of the brake torque (0–3000, 1000 corresponding to 100.0% of the motor rated current)	R/W
	2009H	Special control command word: Bit0–1: =00: Motor 1 =01: Motor 2 Bit2: =1 Enable speed/torque control switchover =0: Disable speed/torque control switchover Bit3: =1 Clear electricity consumption =0: Not clear electricity consumption Bit4: =1 Pre-excitation; =0: Disable pre-excitation Bit5: =1 DC brake =0: Disable DC brake	R/W
	200AH	Virtual input terminal command, range: 0x000–0x3FF Corresponding to S8/S7/S6/S5/HDIB/HDIA/S4/ S3/ S2/S1	R/W
	200BH	Virtual output terminal command, range: 0x00–0x0F Corresponding to local RO2/RO1/HDO/Y1	R/W
	200CH	Voltage setting (used for V/F separation) (0–1000, 1000 corresponding to 100.0% of the motor rated voltage)	R/W
	200DH	AO output setting 1 (-1000–+1000, 1000 corresponding to 100.0%)	R/W
	200EH	AO output setting 2 (-1000–+1000, 1000 corresponding to 100.0%)	R/W

Function	Address	Data description		R/W
VFD status word 1	2100H	0001H: Forward running		R
		0002H: Reverse running		
		0003H: Stopped		
		0004H: Faulty		
		0005H: POFF		
		0006H: Pre-excited		
VFD status word 2	2101H	Bit0: =0: Not ready to run =1: Ready to run Bi1–2: =00: Motor 1 =01: Motor 2 Bit3: =0: Asynchronous motor =1: Synchronous motor Bit4: =0: No overload alarm =1: Overload alarm Bit5–Bit6: =00: Keypad-based control =01: Terminal-based control =10: Communication-based control Bit7: Reserved Bit8: =0: Speed control =1: Torque control Bit9: =0: Non position control =1: Position control Bit11–Bit10: =0: Vector 0 =1: Vector 1 =2: Closed-loop vector =3: 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
Rotating speed	3005H	0–65535 (Unit: 1RPM)		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

Function	Address	Data description	R/W
Closed-loop feedback	3009H	-100.0–100.0% (Unit: 0.1%)	R
Input state	300AH	000–3F Corresponding to the local HDIB/ HDIA/S4/S3/S2/S1	R
Output state	300BH	000–0F Corresponding to the 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 current step of multi-step speed	3012H	0–15	R
External length	3013H	0–65535	R
External count value	3014H	0–65535	R
Torque setting	3015H	-300.0–300.0% (Unit: 0.1%)	R
Identification code	3016H		R
Fault code	5000H		R

The Read/Write (R/W) characteristics indicate whether a function can be read and written. For example, "Communication-based control command" can be written, and therefore the command code 6H is used to control the VFD. The R characteristic indicates that a function can only be read, and W indicates that a function 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 running command channel" (P00.02) to the Modbus communication channel. For another example, when modifying "PID setting",

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 identification code 2103H of the VFD).

Eight high-order bits of code	Meaning	Eight low-order bits of code	Meaning
01	GD	0x08	GD35 vector VFD
		0x09	GD35-H1 vector VFD
		0x0a	GD300 vector VFD
		0xa0	GD350 vector VFD
		0xa1	GD350-UL vector VFD
		0xa2	GD350A vector 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, we can multiply 50.12 by 100 to obtain an integer 5012, and then 50.12 can be represented as 1394H (5012 in the decimal form) in the hexadecimal 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 decimals in the value specified in "Detailed parameter description" or "Default value". If there are *n* decimals in the value, the fieldbus scale m is the n^{th}-power of 10. Take the following table as an example, m is 10.

Function code	Name	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	Restart after power failure	0: Disable 1: Enable	0–1	0

The value specified in "Setting range" or "Default" contains one decimal, so the fieldbus scale is 10. If the value received by the upper computer is 50, the value of "Wake-up-from-sleep delay" of the VFD is 5.0 (5.0=50/10).

To set the "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 transmitted. In this case, the VFD returns an error message response.

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

Code	Name	Description
01H	Invalid command	The command code received by the upper computer 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 the 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-transmitted 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 is set to an invalid value in the write operation. For example, a function input terminal cannot be set repeatedly.
05H	Password error	The password entered in the password verification address is different from that set in P07.00.
06H	Data frame error	The length of the data frame transmitted by the upper computer is incorrect, 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.

Code	Name	Description
08H	Parameter cannot be modified in running	The parameter to be modified in the write operation of the upper computer cannot be modified during the running of the VFD.
09H	Password protection	A user password is set, and the upper computer does not provide the password to unlock the system when performing a read or write operation. The error of "system locked" is reported.

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

For example, if the master device transmits a request message to a slave device for reading a group of function code address data, the code is generated as follows:

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

For an exception response, the following code is returned:

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

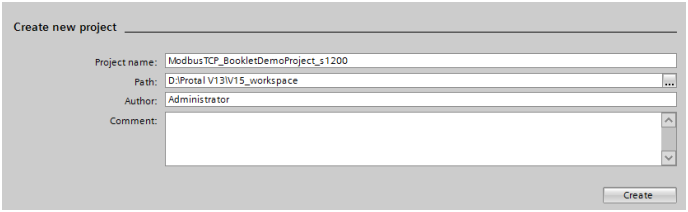
In addition to the modification of the code, the slave device returns a byte of exception code that describes the cause of the exception. After receiving the exception response, the typical processing of the master device is to transmit the request message again or modify the command based on the fault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

5.5 Example of PLC communication

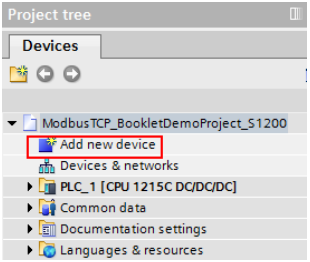
This example shows how to use SIEMENS PLC (S7-1200) to communicate with Modbus TCP communication extension card (through the TIA Portal V13 software), and Modbus TCP is not configured with device description file.

Use TIA Portal V13 software to add a Modbus TCP block.

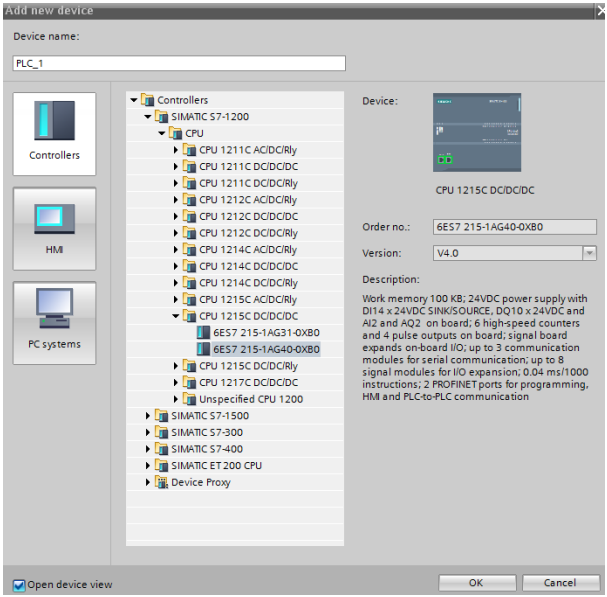
Open TIA Portal V13, and create a new project as shown in the following figure.



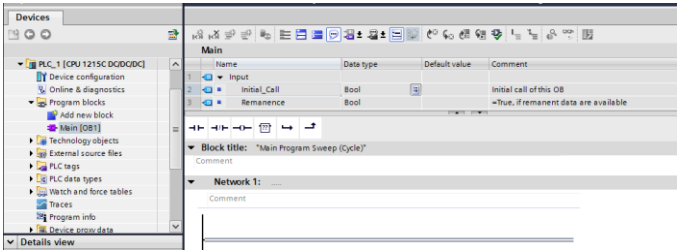
After a new project is created, click "Project view" in the lower left corner, and double click "Add new device" in the interface, as shown in the following figure.



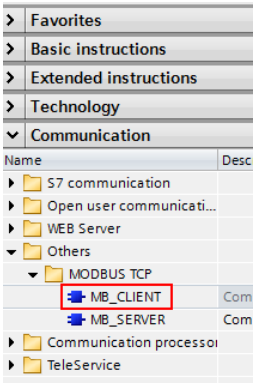
Select the correct PLC model, and click "OK" (PLC models used by our company are shown in the following figure).



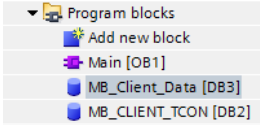
Click "Program blocks", and double click "Main[OB1]" to open the programming interface, as shown in the following figure.



Select "Others" under the "Communication" bar on the right, then select "MODBUS TCP" → "MB_CLIENT", as shown in the following figure.



Add 2 data blocks in "Add new block" under "Program blocks", namely "MB_CLIENT_TCON" and "MB_Client_Data", as shown in the following figure.



Set the variables of these two data blocks respectively, as shown in the following figure.

(1) MB_CLIENT_TCON data block

MB_CLIENT_TCON

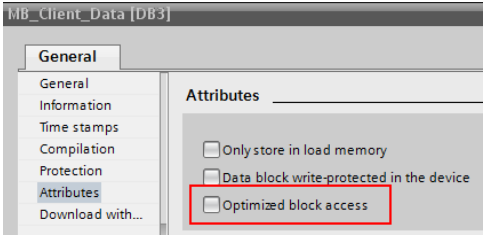
	Name	Data type	Start value
1	Static		
2	TCON	TCON_IP_v4	
3	InterfaceId	HW_ANY	64
4	ID	CONN_OUC	2
5	ConnectionType	Byte	16#0B
6	ActiveEstablished	Bool	1
7	RemoteAddress	IP_V4	
8	ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	192
10	ADDR[2]	Byte	168
11	ADDR[3]	Byte	0
12	ADDR[4]	Byte	2
13	RemotePort	UInt	502
14	LocalPort	UInt	0

(2) MB_Client_Data data block

MB_Client_Data

	Name	Data type	Start value
1	Static		
2	data	Array[0..9] of Int	
3	data[0]	Int	0
4	data[1]	Int	0
5	data[2]	Int	0
6	data[3]	Int	0
7	data[4]	Int	0
8	data[5]	Int	0
9	data[6]	Int	0
10	data[7]	Int	0
11	data[8]	Int	0
12	data[9]	Int	0

Uncheck the "Optimized block access" of the block, as shown in the following figure.

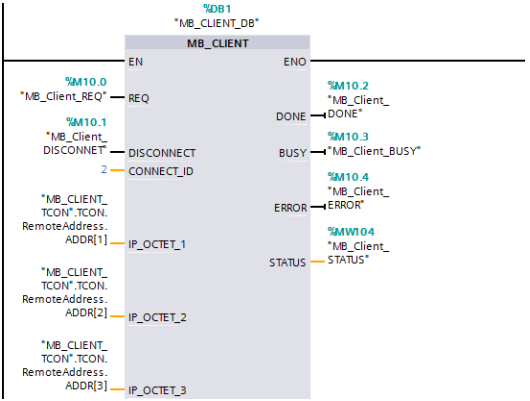


Double click "Show all" under "PLC tags", and create variables, as shown in the following figure.

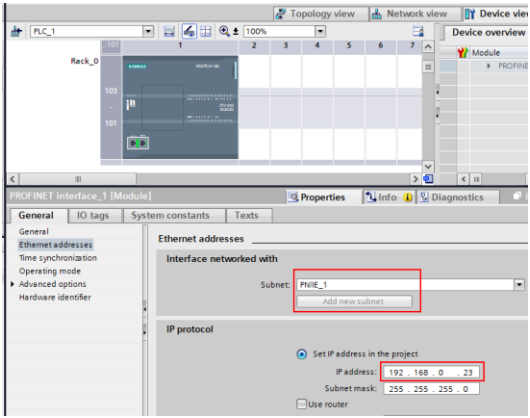
PLC tags

	Name	Tag table	Data type	Address
1	MB_Client_REQ	Default tag table	Bool	%M10.0
2	MB_Client_DISCONNET	Default tag table	Bool	%M10.1
3	MB_Client_MODE	Default tag table	USInt	%MB20
4	MB_Client_ADDR	Default tag table	Word	%MW100
5	MB_Client_LEN	Default tag table	UInt	%MW102
6	MB_Client_DONE	Default tag table	Bool	%M10.2
7	MB_Client_BUSY	Default tag table	Bool	%M10.3
8	MB_Client_ERROR	Default tag table	Bool	%M10.4
9	MB_Client_STATUS	Default tag table	Word	%MW104
10	AUTO_RUN	Default tag table	Bool	%M0.0
11	RUN_TERM	Default tag table	Bool	%I0.0

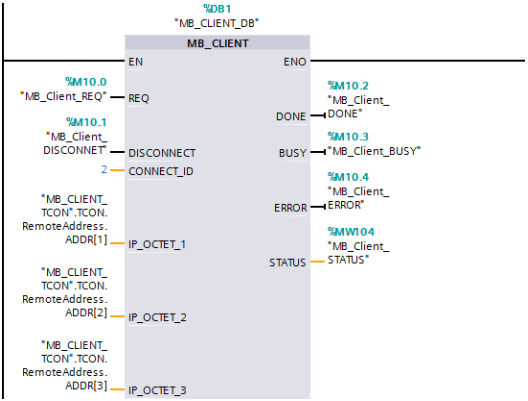
Configure the Modbus TCP block as shown in the following figure.

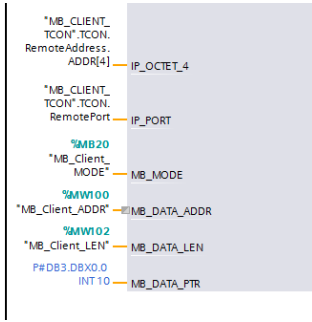


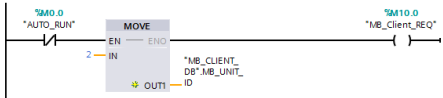
Double click "Device configuration", right click the network port to select "Properties", set the following parameters in the pop-up interface, and modify the local Ethernet network segment to be the same as the following network segment.

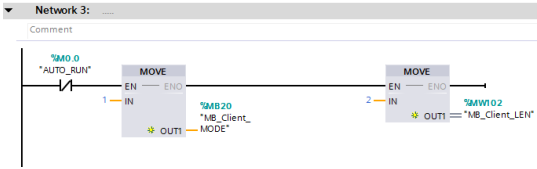


Write the PLC program as follows.




"MB_CLIENT_
TCON".TCON.
RemoteAddress.
ADDR[4]
IP_OCTET_4
"MB_CLIENT_
TCON".TCON.
RemotePort
IP_PORT
%MB20
"MB_Client_
MODE"
MB_MODE
%MW100
"MB_Client_ADDR"
MB_DATA_ADDR
%MW102
"MB_Client_LEN"
MB_DATA_LEN
P#DB3.DBX0.0
INT 10
MB_DATA_PTR


%M0.0
"AUTO_RUN"
MOVE
EN
ENO
2
IN
OUT1
"MB_CLIENT_
DB".MB_UNIT_
ID
%M10.0
"MB_Client_REQ"


Network 3:
Comment
%M0.0
"AUTO_RUN"
MOVE
EN
ENO
1
IN
OUT1
%MB20
"MB_Client_
MODE"
MOVE
EN
ENO
2
IN
OUT1
%MW102
"MB_Client_LEN"


%M0.0
"AUTO_RUN"
MOVE
EN
ENO
48193
IN
OUT1
%MW100
"MB_Client_ADDR"


%I0.0
"RUN_TERM"
MOVE
EN
ENO
1
IN
OUT1
%DB3.DBW0
"MB_Client_
Data".data[0]
MOVE
EN
ENO
5000
IN
OUT1
%DB3.DBW2
"MB_Client_
Data".data[1]

Download the PLC program to the PLC after the program has been written completely. Set VFD function codes such as P00.01=2, P00.02=0, P00.06=8, P14.00=2, P16.02–P16.05= 192.168.0.2, and keep default values of P16.06–P16.13. Then, you can use the I0.0 input terminal to control the VFD to start and stop at 50.00Hz through ModbusTCP protocol.

Appendix A EtherCAT object dictionary

Index	Subindex	Description	Access permission	Data type	Default value
1000h	0	Device type	RO	UINT32	0x00000192
1001h	0	Error register	RO	UINT8	0
1008h	0	Factory device name	RO	String	INVT-EtherCAT
1009h	0	Factory hardware version	RO	String	Hardware version depended
100Ah	0	Factory software version	RO	String	Software version depended
1018h	**ID object**				
	0	Included max. sub-index	RO	UINT8	4
	1	Supplier ID	RO	UINT32	0x000004D8
	2	Product code	RO	UINT32	0x00009252
	3	Revision number	RO	UINT32	0x00000001
	4	Serial number	RO	UINT32	0x00000001
1600h	**RX PDO1 mapping parameter**				
	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	RW	UINT32	0x60600008

Index	Subindex	Description	Access permission	Data type	Default value
		object			
	7	Seventh mapping	RW	UINT32	0x60810020
	8	Eighth mapping object	RW	UINT32	0x60B80010
1601h	**RX PDO2 mapping parameter**				
	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 parameter**				
	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 parameter**				
	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 parameter**				
	0	Number of supported mapping objects	RW	UINT8	8

Index	Subindex	Description	Access permission	Data type	Default value
	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 object	RW	UINT32	0x60B90010
	8	Eighth mapping object	RW	UINT32	0x60BA0020
1A01h	**TX PDO2 mapping parameter**				
	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 parameter**				
	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 parameter**				
	0	Number of supported	RW	UINT8	8

Index	Subindex	Description	Access permission	Data type	Default value
		mapping objects			
	1	First mapping object	RW	UINT32	0x60410010
	2	Second mapping object	RW	UINT32	0x60640020
1C00h	**SM communication type**				
	0	Max. sub-index	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 assignment**				
	0	Max. sub-index	RW	UINT8	1
	1	RxPDO assigned object index	RW	UINT16	0x1600
1C13h	**TxPDO assignment**				
	0	Max. sub-index	RW	UINT8	1
	1	TxPDO assigned object index	RW	UINT16	0x1A00
1C32h	**SM synchronization output parameter**				
	0x00	Max. sub-index	RO	UINT8	0x20
	0x01	Synchronization mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switching time	RO	UINT32	0
	0x04	Supported	RO	UINT16	0x4006

Index	Subindex	Description	Access permission	Data type	Default value
		synchronization type			
	0x05	Min. periodic time	RO	UINT32	0x0003D090
	0x06	Calculation and replication time	RO	UINT32	0
	0x07	Reserved	RW	UINT32	0
	0x08	Obtained periodic time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0
	0x0A	Sync0 time	RW	UINT32	-
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Circulation timeout counter	RO	UINT32	0
	0x0D	Counter of too short switching	RO	UINT32	0
	0x20	Synchronization error	RO	UINT8	0
1C33h	**SM synchronization input parameter**				
	0x00	Max. sub-index	RO	UINT8	0x20
	0x01	Synchronization mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switching time	RO	UINT32	0
	0x04	Supported synchronization type	RO	UINT16	0x4006
	0x05	Min. periodic time	RO	UINT32	0x0003D090
	0x06	Calculation and replication time	RO	UINT32	0
	0x07	Reserved	RW	UINT32	0
	0x08	Obtained periodic time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0
	0x0A	Sync0 time	RW	UINT32	-

Index	Subindex	Description	Access permission	Data type	Default value
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Circulation timeout counter	RO	UINT32	0
	0x0D	Counter of too short switching	RO	UINT32	0
	0x20	Synchronization error	RO	UINT8	0
2000h	0	Read function codes	RW	UINT32	0
2001h	0	Read response	RO	UINT32	0
2002h	0	Write function codes	RW	UINT32	0
2003h	0	Write response	RO	UINT32	0
603Fh	0	Error code	RO	UINT16	0
6040h	0	Control word	RW	UINT16	0
6041h	0	Status word	RO	UINT16	0
605Dh	0	Suspension mode	RW	INT16	0
6060h	0	Operation mode	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

Index	Subindex	Description	Access permission	Data type	Default value
606Ch	0	Actual speed	RW	DINT32	0
6071h	0	Target torque	RW	INT16	0
6072h	0	Max. torque	RW	UINT16	0
6073h	0	Max. current	RO	UINT16	0
6075h	0	Motor rated current	RO	UDINT32	0
6076h	0	Motor rated torque	RO	UDINT32	0
6077h	0	Actual torque	RO	INT16	0
6078h	0	Actual current	RO	INT16	0
6079h	0	Bus voltage	RO	UDINT32	0
607Ah	0	Target position	RW	INT16	0
607Bh	**Position range limit**				
	0	Number of sub-indexes	RW	UINT8	2
	1	Min. position range limit	RW	INT32	0
	2	Max. position range limit	RW	INT32	0
607Ch	0	Coordinate deviation	RW	DINT32	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
6091h	**Gear ratio**				
	0	Number of sub-indexes	RW	UINT8	2
	1	Motor resolution	RW	UINT32	0x00000001
	2	Bearing axle resolution	RW	UINT32	0x00000001

Index	Subindex	Description	Access permission	Data type	Default value
6093h	**Position factor**				
	0	Number of sub-indexes	RW	UINT8	2
	1	Molecule	RW	UINT32	0x00000001
	2	Set constant	RW	UINT32	0x00000001
6098h	0	Zeroing mode	RW	INT16	0
6099h	**Zeroing speed**				
	0	Number of sub-indexes	RW	UINT8	2
	1	Search limit switch speed	RW	UINT32	0
	2	Search zero-phase speed	RW	UINT32	0
60B8h	0	Probe control	RW	UINT16	0
60B9h	0	Probe status	RO	UINT16	0
60BAh	0	Probe position rising edge	RO	INT32	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 Related function codes

Function code	Name	Parameter description	Setting range	Default value
P00.01	Channel of running commands	0: Keypad 1: Terminal 2: Communication	0–2	0
P00.02	Communication channel of running commands	0: Modbus RTU/Modbus TCP communication 1: PROFIBUS/CANopen/DeviceNet communication 2: Ethernet communication 3: EtherCAT/PROFINET/EtherNet IP communication 4: PLC programmable extension card 5: Wireless communication card **Note:** Channels 1, 2, 3, 4, and 5 are extension functions that require corresponding extension cards.	0–5	0
P00.06	Frequency A command setting mode	0: Keypad 1–7: Reserved 8: Modbus RTU/Modbus TCP communication 9: PROFIBUS/CANopen/DeviceNet communication 10: Ethernet communication 11–12: Reserved 13: EtherCAT/PROFINET/EtherNet IP communication 14–15: Reserved	0–15	0
P00.07	Frequency B command setting mode		0–15	2
P03.11	Torque setting mode	0–1: Keypad 2–6: Reserved 7: Modbus RTU/Modbus TCP communication 8: PROFIBUS/CANopen/DeviceNet communication 9: Ethernet communication 10: Reserved 11: EtherCAT/PROFINET/EtherNet IP communication	0–12	0

Function code	Name	Parameter description	Setting range	Default value
		12: Reserved		
P03.14	Setting mode of upper frequency limit of forward running in torque control	0: Keypad (P03.16) 1–5: Reserved 6: Modbus RTU/Modbus TCP communication 7: PROFIBUS/CANopen/DeviceNet communication 8: Ethernet communication 9: Reserved 10: EtherCAT/PROFINET/EtherNet IP communication 11–12: Reserved	0–12	0
P03.15	Setting mode of upper frequency limit of reverse running in torque control	0: Keypad (P03.17) 1–5: Reserved 6: Modbus RTU/Modbus TCP communication 7: PROFIBUS/CANopen/DeviceNet communication 8: Ethernet communication 9: Reserved 10: EtherCAT/PROFINET/EtherNet IP communication 11–12: Reserved	0–12	0
P03.18	Setting mode of upper limit of electromotive torque	0: Keypad (P03.20) 1–4: Reserved 5: Modbus RTU/Modbus TCP communication 6: PROFIBUS/CANopen/DeviceNet communication 7: Ethernet communication 8: Reserved 9: EtherCAT/PROFINET/EtherNet IP communication 10–11: Reserved	0–11	0
P03.19	Setting mode of upper limit of brake torque	0: Keypad (P03.21) 1–4: Reserved 5: Modbus RTU/Modbus TCP communication	0–11	0

Function code	Name	Parameter description	Setting range	Default value
		6: PROFIBUS/CANopen/DeviceNet communication 7: Ethernet communication 8: Reserved 9: EtherCAT/PROFINET/EtherNet IP communication 10–11: Reserved		
P04.27	Voltage setting channel	0: Keypad (P04.28) 1–6: Reserved 7: Modbus RTU/Modbus TCP communication 8: PROFIBUS/CANopen/DeviceNet communication 9: Ethernet communication 10: Reserved 11: EtherCAT/PROFINET/EtherNet IP communication 12–13: Reserved	0–13	0
P06.01	Y1 output	0: Invalid 1–22: Reserved 23: Modbus RTU/Modbus TCP communication virtual terminal output 24: PROFIBUS/CANopen/DeviceNet communication virtual terminal output 25: Ethernet communication virtual terminal output 26–33: Reserved 34: EtherCAT/PROFINET/EtherNet IP communication virtual terminal output 35–63: Reserved	0–63	0
P06.02	HDO output		0–63	0
P06.03	Relay output RO1		0–63	1
P06.04	Relay output RO2		0–63	5
P06.14	Analog output AO1	0: Running frequency 1–13: Reserved 14: Modbus RTU/Modbus TCP communication setting 1 15: Modbus RTU/Modbus TCP communication setting 2 16: PROFIBUS/CANopen/DeviceNet communication setting 1	0–47	0
P06.16	HDO high-speed pulse output		0–47	0

Function code	Name	Parameter description	Setting range	Default value
		17: PROFIBUS/CANopen/DeviceNet communication setting 2 18: Ethernet communication setting 1 19: Ethernet communication setting 2 20: Reserved 21: EtherCAT/PROFINET/EtherNet IP communication setting 1 22–26: Reserved 27: EtherCAT/PROFINET/EtherNet IP communication setting 2 28–47: Reserved		
P07.27	Type of current fault	0: No fault 18: 485/Modbus TCP communication fault (CE) 29: PROFIBUS communication fault (E-DP) 30: Ethernet communication fault (E-NET) 31: CANopen communication fault (E-CAN) 57: PROFINET communication timeout fault (E-PN) 58: CAN communication timeout fault (ESCAN) 60: Card identification failure in slot 1 (F1-Er) 61: Card identification failure in slot 2 (F2-Er) 62: Card identification failure in slot 3 (F3-Er) 63: Card communication failure in slot 1 (C1-Er) 64: Card communication failure in slot 2 (C2-Er) 65: Card communication failure in slot 3 (C3-Er) 66: EtherCAT communication fault (E-CAT)	/	/

Function code	Name	Parameter description	Setting range	Default value
		67: BACnet communication fault (E-BAC) 68: DeviceNet communication fault (E-DEV) 69: CAN slave fault in master/slave synchronous communication (S-Err) 70: EtherNet IP communication timeout (E-EIP)		
P07.28	Type of last fault	/	/	/
P07.29	Type of 2nd-last fault	/	/	/
P07.30	Type of 3rd-last fault	/	/	/
P07.31	Type of 4th-last fault	/	/	/
P07.32	Type of 5th-last fault	/	/	/
P08.31	Motor 1 and motor 2 switching channel	0x00–0x14 LED ones place: Switching channel 0: Terminal 1: Modbus RTU/Modbus TCP communication 2: PROFIBUS/CANopen/DeviceNet communication 3: Ethernet communication 4: EtherCAT/PROFINET/EtherNet IP communication LED tens place: Switching in running 0: Disabled 1: Enabled	00–14	0x00
P09.00	PID reference source	0: Keypad (P09.01) 1–5: Reserved 6: Modbus RTU/Modbus TCP communication 7: PROFIBUS/CANopen/DeviceNet communication 8: Ethernet communication 9: Reserved	0–12	0

Function code	Name	Parameter description	Setting range	Default value
		10: EtherCAT/PROFINET/EtherNet IP communication 11–12: Reserved		
P09.02	PID feedback source	0: AI1 1–3: Reserved 4: Modbus RTU/Modbus TCP communication 5: PROFIBUS/CANopen/DeviceNet communication 6: Ethernet communication 7: Reserved 8: EtherCAT/PROFINET/EtherNet IP communication 9–10: Reserved	0–10	0
P14.00	Local communication address	1–247	1–247	1
P14.03	Communication response delay	1–200ms	1–200	5
P14.05	Transmission error 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	0x00–0x11 Ones place: 0: Respond to write operations 1: Not respond to write operations Tens place: Communication encrypting 0: Disable 1: Enable Hundreds place: Self-define the communication command address 0: Disable	0x000–0x111	0x000

Function code	Name	Parameter description	Setting range	Default value
		1: Enable		
P14.09	Modbus-TCP communication timeout time	0.0 (invalid)–60.0s	0.0–60.0	0.0s
P15.01	Module address	0–127	0–127	2
P15.02	Received PZD2	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01 Hz) 2: PID reference (0–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (0–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)	0–31	0
P15.03	Received PZD3		0–31	0
P15.04	Received PZD4		0–31	0
P15.05	Received PZD5		0–31	0
P15.06	Received PZD6		0–31	0
P15.07	Received PZD7		0–31	0
P15.08	Received PZD8		0–31	0
P15.09	Received PZD9		0–31	0
P15.10	Received PZD10		0–31	0
P15.11	Received PZD11		0–31	0
P15.12	Received PZD12	5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor) 8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor) 9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence) 10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2, RO1, HDO, and Y1 in sequence)	0–31	0

Function code	Name	Parameter description	Setting range	Default value
		11: Voltage setting (for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor) 12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%) 13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%) 14: MSB of position reference (signed number) 15: LSB of position reference (unsigned number) 16: MSB of position feedback (signed number) 17: LSB of position feedback (unsigned number) 18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) 19–31: Reserved		
P15.13	Transmitted 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: Rotating speed of the running (×1, RPM) 9: Linear speed of the running (×1, m/s) 10: Ramp frequency reference 11: Fault code 12: AI1 value (×100, V) 13: AI2 value (×100, V) 14: AI3 value (×100, V) 15: HDIA frequency (×100, kHz) 16: Terminal input state	0–31	0
P15.14	Transmitted PZD3		0–31	0
P15.15	Transmitted PZD4		0–31	0
P15.16	Transmitted PZD5		0–31	0
P15.17	Transmitted PZD6		0–31	0
P15.18	Transmitted PZD7		0–31	0
P15.19	Transmitted PZD8		0–31	0
P15.20	Transmitted PZD9		0–31	0
P15.21	Transmitted PZD10		0–31	0

Function code	Name	Parameter description	Setting range	Default value
P15.22	Transmitted PZD11	17: Terminal output state 18: PID reference (×100, %) 19: PID feedback (×100, %) 20: Rated torque of the motor 21: MSB of position reference (signed number) 22: LSB of position reference (unsigned number) 23: MSB of position feedback (signed number) 24: LSB of position feedback (unsigned number) 25: Status word 26: HDIB frequency value (×100, kHz) 27–31: Reserved	0–31	0
P15.23	Transmitted PZD12		0–31	0
P15.24	Temporary variable 1 used for transmitted PZD	0–65535	0–65535	0
P15.25	DP communication timeout time	0.0 (invalid)–300.0s	0.0–300.0	0.0s
P15.26	CANopen communication timeout time	0.0 (invalid)–300.0s	0.0–300.0	0.0s
P15.27	CANopen communication baud rate	0: 1000 kbps 1: 800 kbps 2: 500 kbps 3: 250 kbps 4: 125 kbps 5: 100 kbps 6: 50 kbps 7: 20 kbps	0–7	0
P15.28	CAN communication address	0–127	0–127	1
P15.29	CAN baud rate setting	0: 50Kbps 1: 125Kbps	0–4	1

Function code	Name	Parameter description	Setting range	Default value
		2: 250Kbps 3: 500Kbps 4: 1M bps		
P15.30	CAN communication timeout time	0.0 (invalid)–300.0s	0.0–300.0	0.0s
P15.31	DeviceNet communication timeout time	0.0 (invalid)–300.0s	0.0–300.0	0.0s
P15.32	Displayed node baud rate	0	0	0
P15.33	Enable polling	0–1	0–1	1
P15.34	Output instance in polling	19: INVT self-defined output 20: ODVA basic speed control output 21: ODVA extended speed control output 22: ODVA speed and torque control output 23: ODVA extended speed and torque control output 24: INVT basic speed control output 25: INVT extended speed control output 26: INVT speed and torque control output 27: INVT extended speed and torque control output	19–27	19
P15.35	Input instance in polling	69: INVT self-defined input 70: ODVA basic speed control input 71: ODVA extended speed control input 72: ODVA speed and torque control input 73: ODVA extended speed and torque control input 74: INVT basic speed control input 75: INVT extended speed control input 76: INVT speed and torque control input 77: INVT extended speed and torque control input	69–77	69

Function code	Name	Parameter description	Setting range	Default value
P15.36	Enable state change/period	0–1	0–1	0
P15.37	Output instance in state change/period	19: INVT self-defined output 20: ODVA basic speed control output 21: ODVA extended speed control output 22: ODVA speed and torque control output 23: ODVA extended speed and torque control output 24: INVT basic speed control output 25: INVT extended speed control output 26: INVT speed and torque control output 27: INVT extended speed and torque control output	19–27	19
P15.38	Input instance in state change/period	69: INVT self-defined input 70: ODVA basic speed control input 71: ODVA extended speed control input 72: ODVA speed and torque control input 73: ODVA extended speed and torque control input 74: INVT basic speed control input 75: INVT extended speed control input 76: INVT speed and torque control input 77: INVT extended speed and torque control input	69–77	69
P15.39	Output length of component 19	8–32	8–32	32
P15.40	Input length of component 69	8–32	8–32	32
P15.41	BACnet communication mode setting	0: Enable P16.22 (I-Am service) 1: Enable P15.42 (Baud rate of BACnet_MSTP)	0–1	0
P15.42	Baud rate of BACnet_MSTP	0–5	0–5	0

Function code	Name	Parameter description	Setting range	Default value
P15.43	Communication control word/status word expression method	0: In decimal format 1: In binary format	0–1	0
P15.44–P15.69	Reserved			
P16.01	Ethernet communication rate setting	0: Self-adaption 1: 100M full duplex 2: 100M half duplex 3: 10M full duplex 4: 10M half duplex	0–4	0
P16.02	IP address 1	0–255	0–255	192
P16.03	IP address 2	0–255	0–255	168
P16.04	IP address 3	0–255	0–255	0
P16.05	IP address 4	0–255	0–255	1
P16.06	Subnet mask 1	0–255	0–255	255
P16.07	Subnet mask 2	0–255	0–255	255
P16.08	Subnet mask 3	0–255	0–255	255
P16.09	Subnet mask 4	0–255	0–255	0
P16.10	Gateway 1	0–255	0–255	192
P16.11	Gateway 2	0–255	0–255	168
P16.12	Gateway 3	0–255	0–255	1
P16.13	Gateway 4	0–255	0–255	1
P16.14	Ethernet monitoring variable address 1	0x0000–0xFFFF	0000–FFFF	0x0000
P16.15	Ethernet monitoring variable address 2	0x0000–0xFFFF	0000–FFFF	0x0000
P16.16	Ethernet monitoring variable address 3	0x0000–0xFFFF	0000–FFFF	0x0000

Function code	Name	Parameter description	Setting range	Default value
P16.17	Ethernet monitoring variable address 4	0x0000–0xFFFF	0000–FFFF	0x0000
P16.18	EtherCAT communication timeout time	0.0s (invalid)–300.0s	0.0–300.0	0.5s
P16.19	Reserved			
P16.20	MSD of BACnet device number	Independent code of BACnet device (0–4194303)	0–4194	0
P16.21	LSD of BACnet device number		0–999	1
P16.22	BACnet "I-Am" service setting	0: Transmission at power-on 1: Continuous transmission	0–1	0
P16.23	BACnet communication timeout time	0.0 (invalid)–300.0s	0.0–300.0	0.0s
P16.24	Extension card identification time of slot 1	0.0–600.0s When this parameter is set to 0.0, disconnection fault detection is not performed.	0.0–600.00	0.0
P16.25	Extension card identification time of slot 2	0.0–600.0s When this parameter is set to 0.0, disconnection fault detection is not performed.	0.0–600.00	0.0
P16.26	Extension card identification time of slot 3	0.0–600.0s When this parameter is set to 0.0, disconnection fault detection is not performed.	0.0–600.00	0.0
P16.27	Extension card communication timeout time of slot 1	0.0–600.0s When this parameter is set to 0.0, disconnection fault detection is not performed.	0.0–600.00	0.0
P16.28	Extension card communication timeout time of slot 2	0.0–600.0s When this parameter is set to 0.0, disconnection fault detection is not performed.	0.0–600.00	0.0

Function code	Name	Parameter description	Setting range	Default value
P16.29	Extension card communication timeout time of slot 3	0.0–600.0s When this parameter is set to 0.0, disconnection fault detection is not performed.	0.0–600.00	0.0
P16.30	Reserved			
P16.31	PROFINET communication timeout time	0.0 (invalid)–300.0s	0.0–300.0	0.0s
P16.32	Received PZD2	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01 Hz) 2: PID reference (0–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (0–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor) 5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor) 8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor) 9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence) 10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2,	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 value
		RO1, HDO, and Y1 in sequence) 11: Voltage setting (for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor) 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: MSB of position reference (signed number) 15: LSB of position reference (unsigned number) 16: MSB of position feedback (signed number) 17: LSB of position feedback (unsigned number) 18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) 19–31: Reserved		
P16.43	Transmitted 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: Rotating speed of the running (×1, RPM) 9: Linear speed of the running (×1, m/s) 10: Ramp frequency reference 11: Fault code 12: AI1 value (×100, V) 13: AI2 value (×100, V) 14: AI3 value (×100, V)	0–31	0
P16.44	Transmitted PZD3		0–31	0
P16.45	Transmitted PZD4		0–31	0
P16.46	Transmitted PZD5		0–31	0
P16.47	Transmitted PZD6		0–31	0
P16.48	Transmitted PZD7		0–31	0
P16.49	Transmitted PZD8		0–31	0
P16.50	Transmitted PZD9		0–31	0

Function code	Name	Parameter description	Setting range	Default value
P16.51	Transmitted PZD10	15: HDIA frequency (×100, kHz) 16: Terminal input state 17: Terminal output state 18: PID reference (×100, %) 19: PID feedback (×100, %) 20: Rated torque of the motor 21: MSB of position reference (signed number) 22: LSB of position reference (unsigned number) 23: MSB of position feedback (signed number) 24: LSB of position feedback (unsigned number) 25: Status word 26: HDIB frequency value (×100, kHz) 27–31: Reserved	0–31	0
P16.52	Transmitted PZD11		0–31	0
P16.53	Transmitted PZD12		0–31	0
P16.54	Ethernet IP communication timeout time	0.5–60.0s	0.5–60.0 s	0.5s
P16.55	Ethernet IP communication rate setting	0: Self-adaption 1: 100M full duplex 2: 100M half duplex 3: 10M full duplex 4: 10M half duplex	0–4	0
P19.00	State of card slot 1	0: No card 1: PLC programmable card 2: I/O card 3: Incremental PG card 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	0–65535	0
P19.01	State of card slot 2		0–65535	0
P19.02	State of card slot 3		0–65535	0

Function code	Name	Parameter description	Setting range	Default value
		signals 13: Sine-cosine PG card with CD signals 14: Absolute encoder PG card 15: CAN master/slave communication card 16: Modbus TCP communication card 17: EtherCAT communication card 18: BACnet communication card 19: DeviceNet communication card 20: Reserved 21: Ethernet IP communication card		

Service line: 86-755-23535967 E-mail: overseas@invt.com.cn Website: www.invt.com

The products are owned by **Shenzhen INVT Electric Co.,Ltd.**

Two companies are commissioned to manufacture: (For product code, refer to the 2nd/3rd place of S/N on the name plate.)

Shenzhen INVT Electric Co.,Ltd. (origin code: 01)
Address: INVT Guangming Technology Building, Songbai Road, Matian, Guangming District, Shenzhen, China

INVT Power Electronics (Suzhou) Co.,Ltd. (origin code: 06)
Address: No. 1 Kunlun Mountain Road, Science & Technology Town, Gaoxin District, Suzhou, Jiangsu, China

Industrial Automation: HMI, PLC, VFD, Servo System, Elevator Intelligent Control System, Rail Transit Traction System

Energy & Power: UPS, DCIM, Solar Inverter, SVG, New Energy Vehicle Powerstain System, New Energy Vehicle Charging System, New Energy Vehicle Motor

66001- 00751

202010 (V1.0)