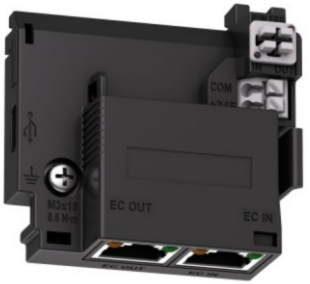
ZSY-TX149

Industrial Ethernet card

User Manual



Preface

Overview

Thank you for choosing z:systems ZSY-TX149 industrial Ethernet communication card.

This manual describes the product features, electrical interface configuration, communication parameter settings, and application examples of communication with the PLC. To ensure that you install and operate the product properly, read this manual and the communication sections in the VFD user manual carefully before you use the product.

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

Precautions

The communication 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 product, and are capable of preventing all kinds of emergencies.

- Before installing, removing, or operating the card, read the safety precautions described in this manual and the variable-frequency drive (VFD) user manual carefully to ensure safe operation.
- We shall not be liable or responsible for any equipment damage or physical injury or death caused due to your or your customers' failure to follow the safety precautions.
- Before opening the VFD housing to install or remove the expansion card, disconnect all power supplies of the VFD and ensure that the voltage inside the VFD is far lower than the human safety voltage. For details, see the description in the VFD user manual. Severe personal injury or even death can result if the instruction is not followed.

- Store the card in a place that is dustproof and damp-proof, free from electric shocks and mechanical pressure.
- The card is electrostatic sensitive. Take measurements to prevent electrostatic discharge when performing related operations.
- When installing the card, tighten the screws to ensure that it is firmly fixed and properly grounded.

Change history

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

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

Terminology and abbreviations

CAN	Controller area network
COB	Communication object, a transmitted unit on a CAN network. Data can be transmitted through the whole network. A COB is part of a CAN message frame.
EDS	Electronic datasheet, an ASCII file for node configuration, required when a CANopen network is configured. An EDS file contains general information about nodes and their dictionary objects (parameters).
NMT	Network management, one of the CAN application-layer service elements in the CAN reference model. It is used for the initialization, configuration, and fault handling of a CAN network.
Object dictionary	Stores information about all COBs identified by a device.
PDO	Process data object, a type of COBs, used to transmit process data, such as control command, set values, state values, and actual values.
PDOn Tx	PDO command sent from the slave to the master; n indicates 1, 2, 3, or 4.
PDOn Rx	PDO command sent from the master to the slave; n indicates 1, 2, 3, or 4.
SDO	Service data object, a type of COB, used to transmit non-time key data, such as parameter values.
RO	Indicates the read-only access.
RW	Indicates the read and write access.
SYNC	Indicates synchronous transmission.
Node-ID	Node ID, that is, address of a communication card.
0x	0x represents a hexadecimal number; for example, 0x10 is equivalent to decimal 16.

Contents

1 Product confirmation 1

1.1 Product features 1

2 PROFINET protocol 6

2.1 Overview 6

2.2 Product features 6

2.2.1 Supported functions 6

2.2.2 Supported communication types 6

2.2.3 Status indicator 6

2.3 Electrical connection 8

2.4 Communication 9

2.4.1 Message format 9

2.4.2 Communication 9

2.5 PLC communication example (S7-1200) 20

2.5.1 Parameter setup 20

2.5.2 Creating a project 24

2.5.3 Adding the GSD file 25

2.5.4 Configuring project basic information 26

2.5.5 Assigning a device name for the IO device (Z:SYSTEMS communication card) 31

2.5.6 Saving, compiling, and downloading 33

2.5.7 Monitoring VFD parameters 35

3 EtherNet IP protocol 37

3.1 Overview **37**

3.2 Product features **37**
3.2.1 Supported functions 37
3.2.2 Supported communication types 37
3.2.3 Status indicator 38

3.3 Electrical connection **39**

3.4 Communication **40**
3.4.1 Communication settings 40
3.4.2 Message format 41
3.4.3 Communication 41

3.5 PLC communication example 1 (1769_L36ERMS) **53**
3.5.1 Creating a project 53
3.5.2 Importing the EDS file 54
3.5.3 Creating a device object 56
3.5.4 Using RSLinx Classic 59
3.5.5 Writing PLC programs 61
3.5.6 Host controller connection and program download 63
3.5.7 Setting the PLC IP address using Studio 5000 V31 65
3.5.8 DLR ring network configuration 65

3.6 PLC communication example 2 (NJ501-1400) **67**
3.6.1 Hardware connection 68
3.6.3 Configuring Sysmac Studio software 73
3.6.5 PLC program downloading and online monitoring 81

4 EtherCAT protocol **90**

4.1 Overview **90**

4.2 Product features **90**
4.2.1 Supported functions 90
4.2.2 Supported services 90
4.2.3 Status indicator 91

4.3 Electrical connection **92**

4.4 Communication **93**
4.4.1 CANopen over EtherCAT reference model 93
4.4.2 EtherCAT slave site information 94
4.4.3 EtherCAT state machine 94
4.4.4 PDO process data mapping 95
4.4.5 Network synchronization based on distributed clocks 97

4.5 CiA402 device profile **98**
4.5.1 CANopen over EtherCAT state machine 98
4.5.2 Device run mode 102

4.6 PLC communication example 1 (TwinCAT2) **105**

4.7 PLC communication example 2 (TM753) **111**

5 Modbus TCP protocol **118**

5.1 Overview **118**

5.2 Product features **118**
5.2.1 Supported functions 118
5.2.2 Supported communication types 118
5.2.3 Status indicator 118

5.3 Electrical connection **120**

5.4 Communication **121**
5.4.1 Communication settings 121
5.4.2 Message format 122
5.4.3 Modbus TCP communication 122
5.4.4 Data address definition 125
5.4.5 Fieldbus scale 131
5.4.6 Error response 132

5.5 PLC communication example 1 (S7-1200) **133**

5.6 PLC communication example 2 (TM753) **142**

Appendix A EtherCAT object dictionary **152**

Appendix B Related function code **164**

1 Product confirmation

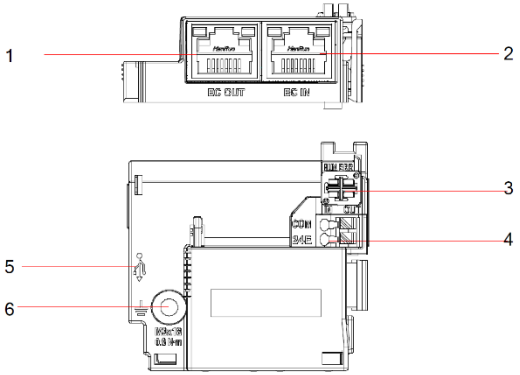
Check the following after receiving the communication card:

- 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, an incorrect model is delivered, or some items are missing, contact the supplier in a timely manner.
- Obtain the ESD file or xml file of the communication card from Z:SYSTEMS.

1.1 Product features

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

Figure 1-1 Product components



No.	Name	Description
1	Communication port (EC OUT)	Supported bus types: PROFINET, EtherCAT, EtherNet IP, and Modbus TCP EtherCAT can be only used in the OUT port, while the other three protocols do not distinguish the direction.
2	Communication port (EC IN)	Supported bus types: PROFINET, EtherCAT, EtherNet IP, and Modbus TCP EtherCAT can be only used in the IN port, while the other three protocols do not distinguish the direction.
3	Indicator	See the indicator description in the corresponding protocol section.
4	+24E	An external 24V connection can be used for communication debugging.
	COM	
5	Type-C	Manufacturer reserved
6	Fixing hole	Used for expansion card and control board installation and fixing.

Table 1-1 Environmental requirements

Item	Specifications
Working temperature	-10–50°C
Storage temperature	-20–60°C
Relative humidity	5%–95% (No condensation)
Other weather conditions	No condensation, ice, rain, snow, or hail; solar radiation < 700 W/m^2
Air pressure	70–106kPa
Vibration and impact	5.8m^2 (0.6g) at the sine vibration of 9Hz to 200Hz

Figure 1-2 RJ45 interface



Table 1-2 RJ45 interface function

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

Table 1-3 Protocol selection (taking ZS28 for example)

Function code	Protocol	Description
	PROFINET	0 (Factory setting)
	EtherCAT	1
	Reserved	2
	EtherNet IP	3

Function code	Protocol	Description
P24.00	Modbus TCP	4
	EtherNet UDP	5
	PROFINET + EtherNet UDP	6
	EtherCAT + EtherNet UDP	7
	Reserved	8–14
	No communication card	15

Table 1-4 Protocol description

Protocol	Description
PROFINET	1. Supports the PROFINET protocol, accommodating PROFINET IO devices, and the medium redundancy protocol (MRP). Configured with the slave GSDML configuration file, it can communicate with Siemens PLC and other master devices. 2. Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. This communication card supports up to 32 IOs. 3. Applicable to linear, star, and ring network topologies.
EtherCAT	1. Supports the CiA301 and CiA402 CoE protocols. Configured with a slave XML configuration file, it can communicate with Beckhoff PLC, Z:SYSTEMS controllers, and other master devices. 2. Supports PDO and SDO services, manufacturer-defined object dictionaries, and SDO reading/writing of VFD function codes, meeting the EtherCAT compliance testing certification requirements within the factory. 3. Applicable to linear, star, and ring network topologies. 4. Equipped with two RJ45 ports, designated for IN and OUT directions.

Protocol	Description
EtherNet IP	1. Supports ODVA standards and DLR ring protocol. When configured with a slave EDS configuration file, it can communicate with Rockwell PLC and other master devices. 2. Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. This communication card supports up to 32 IOs. 3. Applicable to linear, star, and ring network topologies.
Modbus TCP	1. Supports the Modbus TCP protocol. A Modbus TCP slave can communicate with multiple masters simultaneously. It can communicate with Schneider PLC, Z:SYSTEMS controllers, and other master devices. 2. Enables basic operations on VFDs, such as reading and writing process values, reading status values, and reading/writing function codes. 3. Applicable to linear and star network topologies.
EtherNet UDP	1. Supports Z:SYSTEMS Ethernet protocol, connecting to the Z:SYSTEMS Workshop for monitoring and oscilloscope functionalities, allowing for multi-card network monitoring. 2. Applicable to linear and star network topologies.
PROFINET + EtherNet UDP	Supports concurrent PROFINET and EtherNet UDP communications on the same network.
EtherCAT + EtherNet UDP	Supports concurrent EtherCAT and EtherNet UDP communications on the same network, with EtherCAT required to remain online.

2 PROFINET protocol

2.1 Overview

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

2.2 Product features

2.2.1 Supported functions

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

2.2.2 Supported communication types

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

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

It does not support isochronous real-time (IRT) communication channels.

2.2.3 Status indicator

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

Table 2-1 Indicator description

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

Indicator	Color	Definition	Function
ERR	Red	Steady off	No fault
		Blinking (On: 100ms; Off: 100ms)	Communication establishment is abnormal.

2.3 Electrical connection

The PROFINET communication card uses standard RJ45 interfaces, and its electrical connections are shown inFigure 2-1, Figure 2-2, andFigure 2-3.

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

Figure 2-1 Linear network topology electrical connection

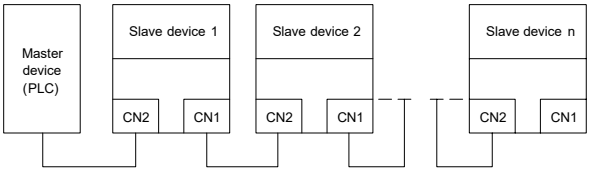
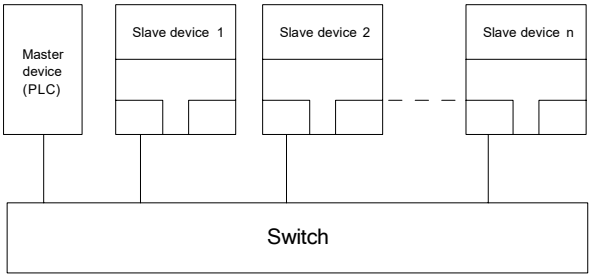
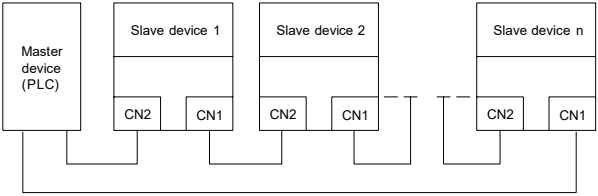


Figure 2-2 Star network topology electrical connection



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

Figure 2-3 Ring network topology electrical connection



2.4 Communication

2.4.1 Message format

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

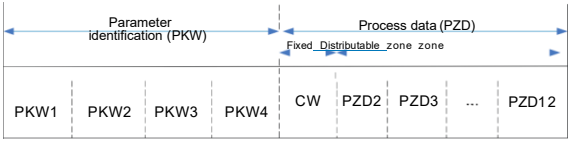
Table 2-2 RT frame structure

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

2.4.2 Communication

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

Figure 2-4 Message structure



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

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—control word (from master to slave; see Table 2-3 and Table 2-4)

SW—status word (from slave to master; see Table 2-6 and Table 2-7)

PZD—Process data (user specified)

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

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

Control word (CW) and status word (SW)

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

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

Actual value: An actual value is a 16-bit word that includes information about VFD device operation. The monitoring function is defined through VFD parameters. The conversion scale of the integer transmitted to the master as the actual value depends on the selected function. For details, see the related VFD user manual.

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

Task message (Master -> VFD)

Control word (CW): The first word in a PZD task message is the control word (CW) of VFD. The representation method can be selected according to function code P14.71. Table 2-3 andTable 2-4 provide descriptions for ZS28 series VFD CWs for reference.

Table 2-3 ZS28 series VFD CWs in decimal

Bit	Name	Value	Description
0–7	Communication-based control command	1	FWD run
		2	REV run
		3	FWD jog
		4	REV jog
		5	Stop
		6	Coast to stop
		7	Fault reset
		8	Jog stop
		9	Stop in emergency manner.

Bit	Name	Value	Description
8	Enable read and write	1	Enable read and write (PKW1–PKW4)
9–10	Motor group setting	0	Select motor 1
		1	Select motor 2
11	Control mode switchover selection	0	No switchover
		1	Enable the switchover between torque control/speed control
12	Clear power consumption quantity	0	Disable the function for resetting power consumption to zero
		1	Enable the function for resetting power consumption to zero
13	Pre-excitation	0	Disable pre-excitation
		1	Enable pre-excitation
14	DC braking	0	Disable DC braking
		1	Enable DC braking
15	Heartbeat reference	0	Disable heartbeat
		1	Enable heartbeat

Table 2-4 ZS28 series VFD CWs in binary

Bit	Name	Description	Priority
0	FWD run	0: Decelerate to stop 1: Run forward	1
1	REV run	0: Decelerate to stop 1: Run reversely	2
2	Fault reset	0: None 1: Fault reset	3
3	Coast to stop	0: None 1: Coast to stop	4
4	Forward jogging	0: None 1: Forward jogging	5
5	Reverse jogging	0: None 1: Reverse jogging	6
6	Jog stop	0: None 1: Stop jogging	7
7	-	Reserved	-
8	Enable read and write (PKW1–4)	0: Disable 1: Enable	-
9	-	Reserved	-
10	Emergency stop	0: None 1: Emergency stop	0: Top priority
11–15	-	Reserved	-

Reference value (REF): The second to twelfth words in a PZD task message are the main settings (REF). The main frequency settings are provided by the main setting signal source.Table 2-5lists the settings of ZS28 series VFD for reference.

Table 2-5 Settings of ZS28 series VFD

Function code	Word	Value range	Default
P23.02	Received PZD2	0–31 0: Invalid 1: Set frequency (0–Fmax, unit: 0.01Hz) 2: PID reference (-1000–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (-1000–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the motor rated current) 5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 8: Upper limit of braking torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 9: Virtual input terminal command (0x000–0x7FF) 10: Virtual output terminal command (0x000–0x01F) 11: Voltage setting special for V/F separation (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage) 12: AO output setting 1 (0–1000, in which 1000 corresponds to 100.0%) 13: AO output setting 2 (-1000–1000, in which 1000 corresponds to 100.0%) 14–18: Reserved 19: Function parameter mapping (PZD2–PZD12 correspond to P14.49–P14.59) 20–31: Reserved	0
P23.03	Received PZD3		0
P23.04	Received PZD4		0
P23.05	Received PZD5		0
P23.06	Received PZD6		0
P23.07	Received PZD7		0
P23.08	Received PZD8		0
P23.09	Received PZD9		0
P23.10	Received PZD10		0
P23.11	Received PZD11		0
P23.12	Received PZD12		0

Response message (VFD -> Master)

Status word (SW): The first word in a PZD response message is the status word (SW) of VFD. The representation method can be selected according to function code P14.71.Table 2-6 andTable 2-7 provide descriptions for ZS28 series VFD SWs for reference.

Table 2-6 ZS28 series VFD SWs in decimal

Bit	Name	Value	Description
0–7	Running status	1	Running forward
		2	Running reversely
		3	VFD stop
		4	VFD in fault
		5	VFD POFF status
8	Bus voltage established	0	Not ready for running
		1	Ready to run
9–10	Motor group feedback	0	Feedback of motor 1
		1	Feedback of motor 2
11	Motor type feedback	0	Asynchronous motor (AM)
		1	Synchronous motor (SM)
12	Overload pre-alarm feedback	0	No pre-alarm upon overload
		1	Pre-alarm upon overload
13–14	Running mode selection	0	Keypad-based control
		1	Terminal-based control
		2	Communication-based control
		3	Reserved
15	Heartbeat feedback	0	No heartbeat feedback
		1	Heartbeat feedback

Table 2-7 ZS28 series VFD SWs in binary

Bit	Name	Description	Priority
0	FWD run	0: None 1: Running forward	1
1	REV run	0: None 1: Running reversely	2
2	Stop	0: None 1: VFD in stopped state	3
3	Fault	0: None 1: VFD in fault	4
4	POFF	0: None 1: VFD in POFF state	5

Bit	Name	Description	Priority
5	Pre-exciting	0: None 1: VFD in pre-exciting state	6
6–15	-	Reserved	-

Actual value (ACT): The second to twelfth words in a PZD task message are the main actual values. The main frequency actual values are provided by the main actual value signal source. Table 2-8 lists the actual status values of ZS28 series VFD for reference.

Table 2-8 Actual status values of ZS28 series VFD

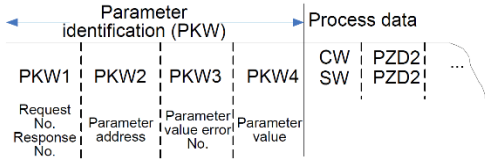
Function code	Word	Value range	Default
P23.13	Sent PZD2	0–32 0: Invalid 1: Running frequency (×100, Hz) 2: Set frequency (×100, Hz) 3: Bus voltage (×10, V) 4: Output voltage (×1, V) 5: Output current (×100, A) 6: Actual output torque (×10, %) 7: Actual output power (×10, %) 8: Rotation speed of running (×1, RPM) 9: Linear speed of running (×1, m/s) 10: Ramp reference frequency (×100, Hz) 11: Fault code 12: AI1 input (×100, V) 13: AI2 input (×100, V) 14: AI3 input (×100, V) 15: Reserved 16: HDI1 frequency value (×100, kHz) 17: Reserved 18: Terminal input state 19: Terminal output status 20: PID reference (×100, %) 21: PID feedback (×100, %) 22–26: Reserved 27: VFD status word 2 28–31: Reserved 32: Function parameter mapping (PZD2–PZD12 correspond to P14.60–P14.70)	0
P23.14	Sent PZD3		0
P23.15	Sent PZD4		0
P23.16	Sent PZD5		0
P23.17	Sent PZD6		0
P23.18	Sent PZD7		0
P23.19	Sent PZD8		0
P23.20	Sent PZD9		0
P23.21	Sent PZD10		0
P23.22	Sent PZD11		0
P23.23	Sent PZD12		0

PKW zone

PKW zone (parameter identification marks PKW1–value zone): PKW zone describes treatment of parameter identification interface, PKW interface is a mechanism

which determine parameters transmission between two communication partners, such as reading and writing parameter values.

Figure 2-5 PKW zone



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

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

Note: If the master requests the value of a parameter, the values in PKW3 and PKW4 of the message that the master 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 2-9 Definitions of the task identification flag PKW1

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

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

Table 2-10 Definitions of the response identification flag PKW1

Response (from slave to master)	
Response code	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 code is returned: 1: Invalid command 2: Invalid data address 3: Invalid data value 4: Operation failure 5: Incorrect password 6: Incorrect data frame 7: Parameter read only 8: Parameter cannot be modified in running

Response (from slave to master)	
Response code	**Function**
	9: Password protection 10: Function code mapping operation failure
4	Reserved

PKW examples:

Example 1: Reading the value of a parameter

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

Request (Master -> VFD)

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



000A: Parameter address

0001: Request for parameter value reading

Response (VFD -> Master)

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



1388: Parameter value in address 10

0001: Response (parameter value updated)

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

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

Request (Master -> VFD)

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

1388: Parameter value in address 10

0004: Modifying a parameter value

Response (VFD -> Master)

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

0001: Response (parameter value updated)

PZD examples: The transmission of the PZD zone is implemented through VFD function code settings. For details about related function codes, see the Z:SYSTEMS user manual.

Example 1: Reading the process data of VFD

In this example, PZD3 is set to "8: Rotating speed during running" through the VFD parameter P23.14. This operation sets the parameter forcibly. The setting remains until the parameter is set to another option.

Response (VFD -> Master)

	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: same as reference" through the VFD parameter P23.03. The parameter specified in each request frame is updated with the information contained in PZD3 until another parameter is specified.

Request (Master -> VFD)

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

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

2.5 PLC communication example (S7-1200)

The following example shows the configuration for using a PROFINET adapter module to communicate with a Siemens S7-1200 series PLC (using TIA Portal V13 as the configuration tool).

2.5.1 Parameter setup

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

Open the TIA PORTAL V13 software.

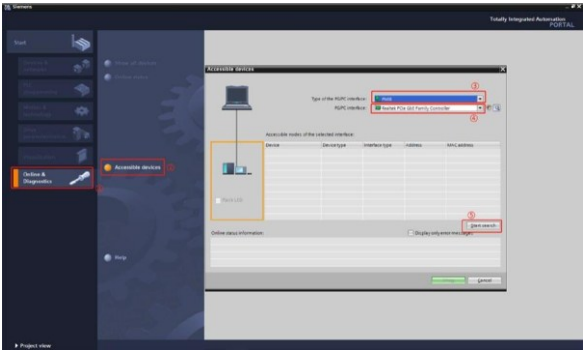
① Click **Online and Diagnostics** on the left.

② Then click **Accessible devices**.

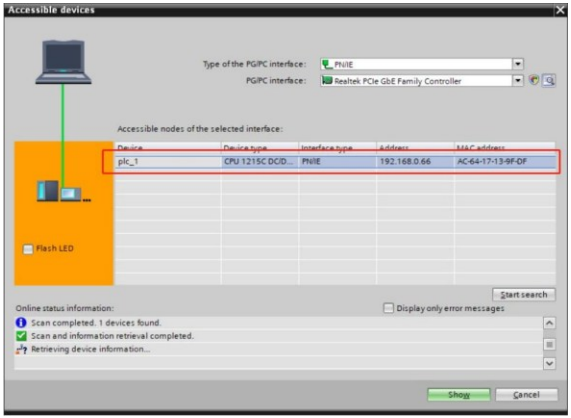
③ In the pop-up **Accessible devices** window, set **Type of PG/PC Interface** to **PN/IE**.

④ Select Ethernet port for **PG/PC Interface**.

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



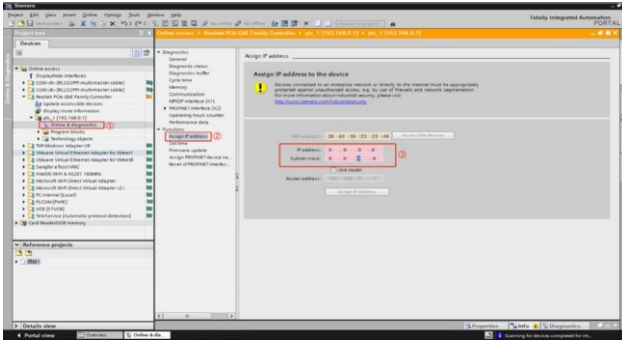
If the connection between the PLC and PC is normal, after the scan is completed, the PLC device will appear in the device column, as shown in the red box in the following figure. The device column will display the device, device type, and device MAC address. Click the **Show** button in the bottom right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



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

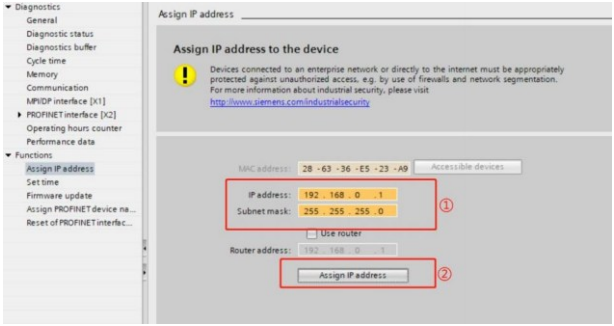
② Choose **Functions** > **Assign IP address** in the menu bar on the right.

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



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

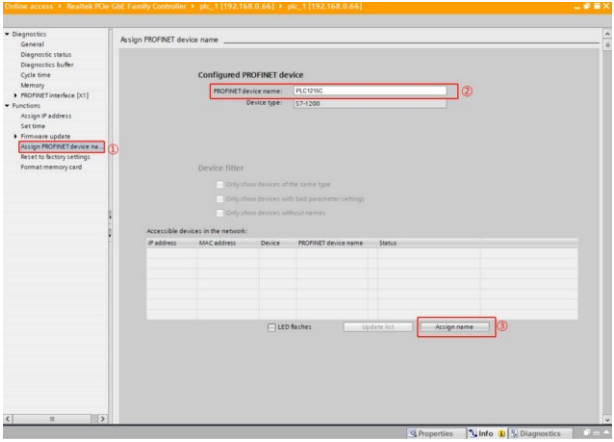
② After setting, click the **Assign IP address**. See the following figure.



① Click **Assign PROFINET device name**.

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

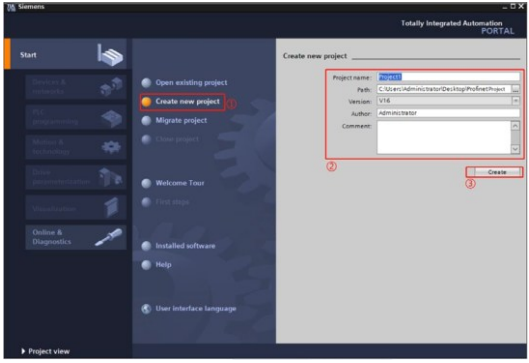
③ Click **Assign name**. See the following figure.



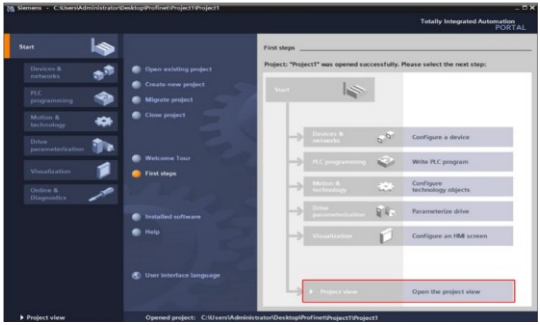
2.5.2 Creating a project

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

① Click **Create new project**.
② Enter project information such as **Project name**, **Path**, **Version**, **Author**, and **Comment**.
③ Click **Create**.



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



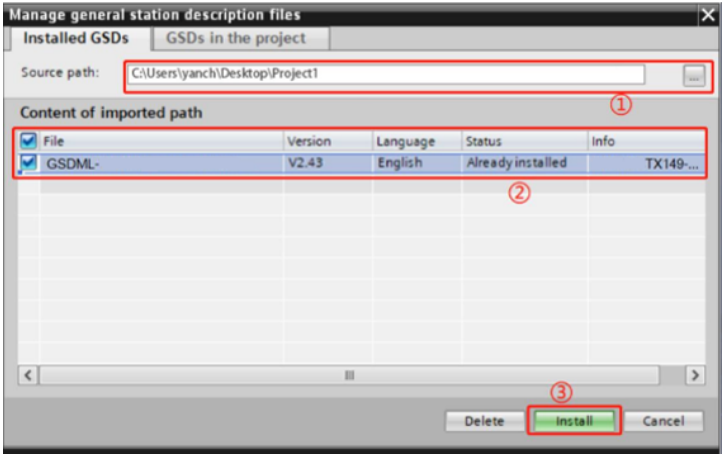
2.5.3 Adding the GSD file

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

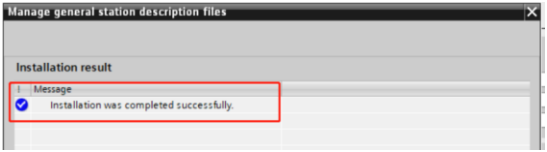
① In the dialog box that appears, enter the source path of the GSD file.

② Select the GSD file.

③ Click **Install**.



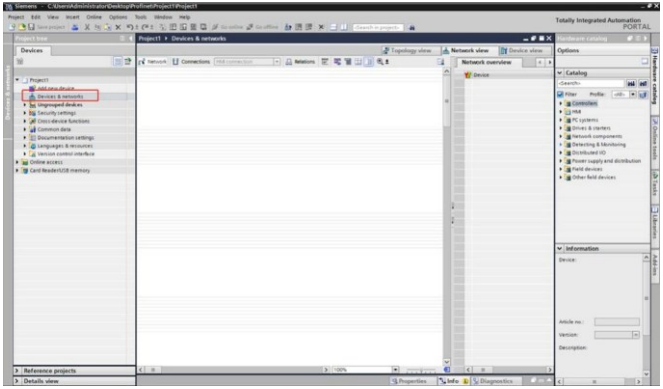
After the installation is successful, a message is displayed, indicating that the GSDML file has been installed successfully.



2.5.4 Configuring project basic information

1. Enter the device and network view interface.

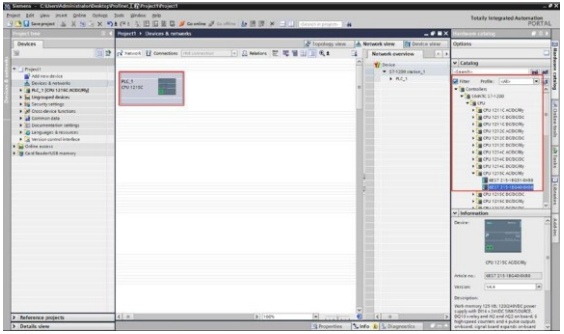
   Choose **Devices and networks** in the project tree on the left, and double-click **Devices and networks** to enter the **Network overview** interface.



2. Add the project device and PROFINET network.

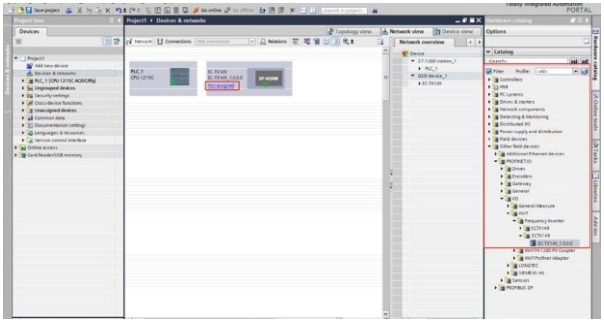
   A. Add **PLC S7-1215C** to the **Devices and networks** view.

   Choose **Controllers** > **SIMATIC S7-1200** > **CPU** > **CPU 1215C AC/DC/RLY** > **6ES7 215-1BG40-0XB0** in the **Hardware catalog** panel on the right, and then double-click or drag the **6ES7 215-1BG40-0XB0** icon to the project.

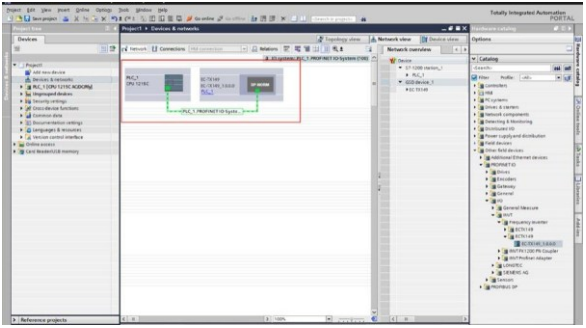


B. Add the Z:SYSTEMS communication card to the **Devices and networks** view.

In the **Hardware catalog** panel on the right, choose **Other field devices** > **PROFINET IO** > **I/O** > **Z:SYSTEMS** > **Frequency Inverter** > **ECTX149**, and double click the **EC-TX109_1.0.0.0** icon or drag it to the view of **Devices & networks**. The communication card is displayed as **Not assigned**.

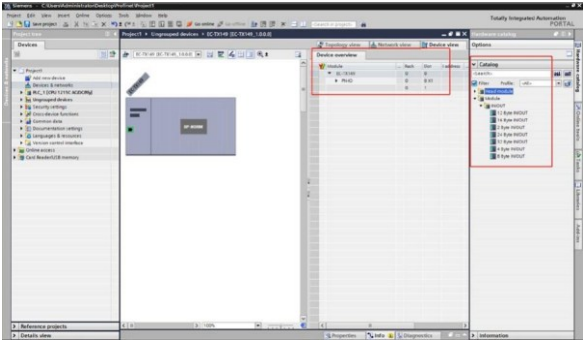


Click the **Not assigned** option of **ZSY-TX149_1.0.0.0**, and select the IO controller **PLC_1.PROFINET interface_1**. In the network view, the CPU and Z:SYSTEMS PROFINET have been connected to the same PROFINET sub network.

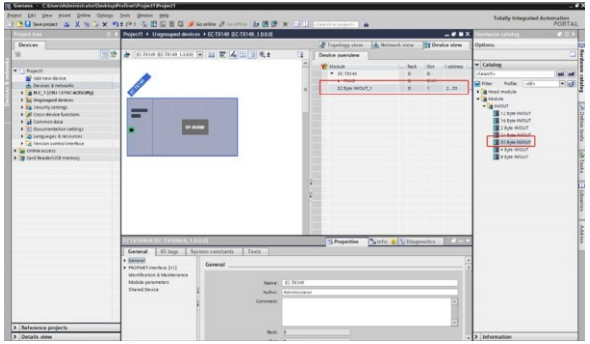


C. Add Z:SYSTEMS I/O sub modules to the project.

Double click the **ZSY-TX149_1.0.0.0** icon in the **Devices & networks** view to enter the Z:SYSTEMS device view interface.



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

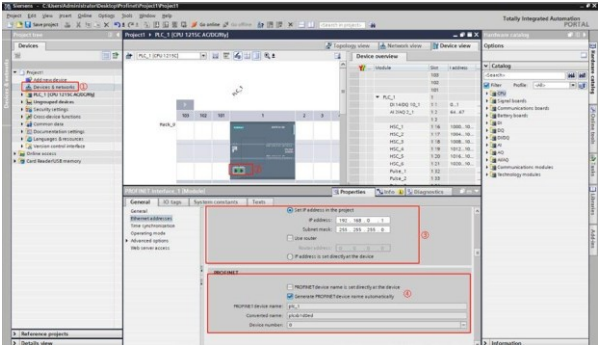


D. Set S7-1215C and Z:SYSTEMS PROFINET basic parameters.

a) Set PLC S7-1215C parameters.

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

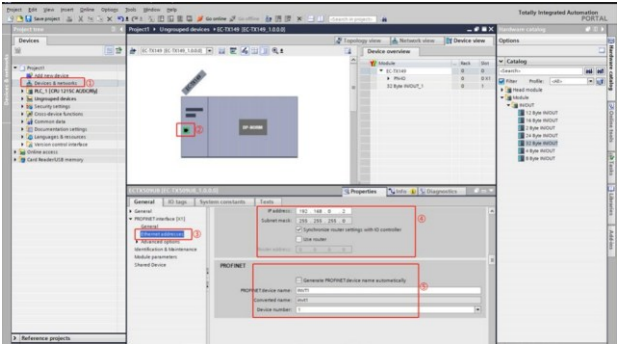
The following figure shows the operation procedure.



b) Set Z:SYSTEMS PROFINET communication card parameters.

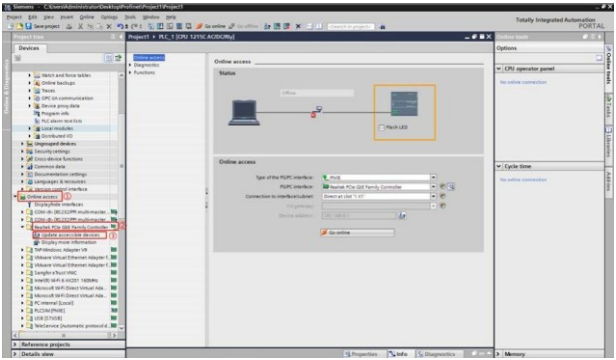
- Double-click **Devices & networks** to enter the **Devices & networks** interface.
- Double click the **ZSY-TX149_1.0.0.0** icon to enter the communication card device view.
- Double click the network interface position of the Z:SYSTEMS PROFINET communication card icon to enter the PROFINET interface editing interface.
- Click the **General** tab, and choose **PROFINET interface [X1]** > **Ethernet addresses**. Set the parameters of the Z:SYSTEMS PROFINET communication card according to the parameters shown in the figure, that is, the IP address and device name of the communication card (using IP 192.168.0.2 and name Z:SYSTEMS1 as an example).

The following figure shows the operation procedure.



2.5.5 Assigning a device name for the IO device (Z:SYSTEMS communication card)

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



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

On all displayed devices, find the Z:SYSTEMS communication card device and click it, for example, the device **ectx149** in the figure.

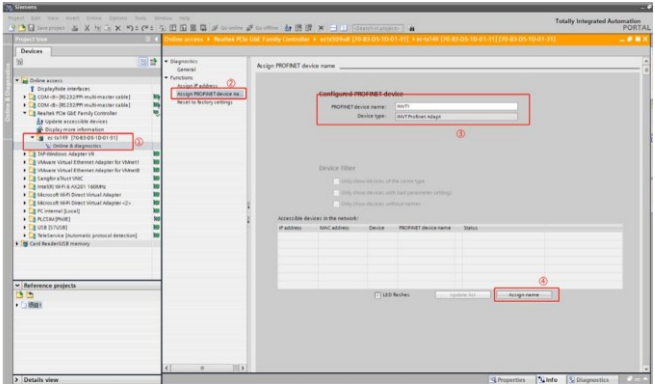
Note: If the communication card is used for the first time, only the default device name can be found.

① Double click **Online and Diagnostics** to enter online diagnostics mode.

② Choose **Functions** > **Assign PROFINET device name**.

③ Access the **Assign PROFINET device name** interface. Set the device name and type, and click **Assign name**.

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



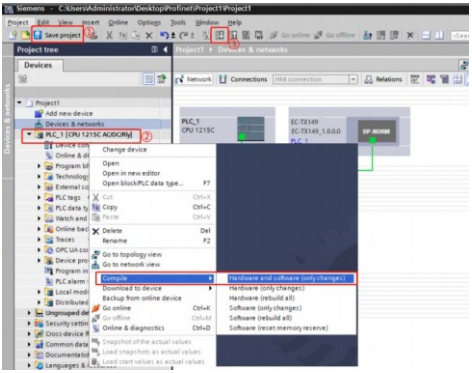
2.5.6 Saving, compiling, and downloading

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

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

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

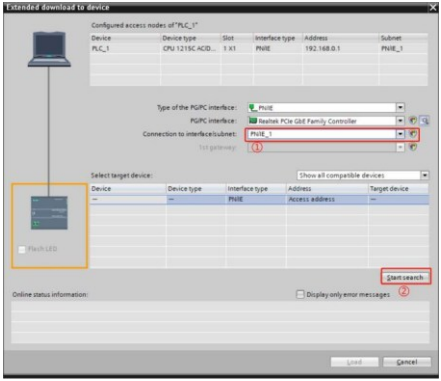
③ Click the icon of download to device to download the project configuration to the PLC.



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

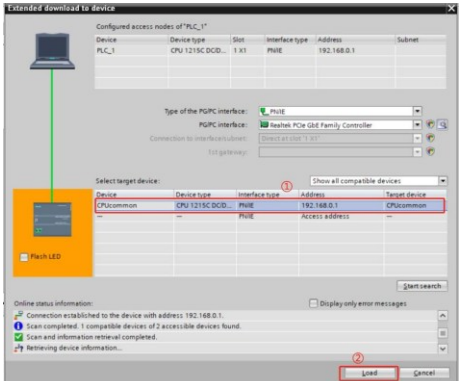
① Select **PN/IE_1** from the **Connection to interface/subnet** drop-down list box.

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



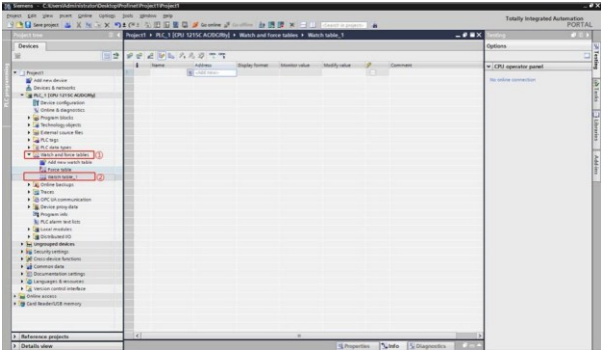
After the search is completed, the PLC S7-1215C connected to the computer will be displayed in the **Show all compatible devices** list, as shown in the following figure.

① Select the target PLC in the following figure.
② Click **Download** to download the configuration information and PLC program to the selected PLC.

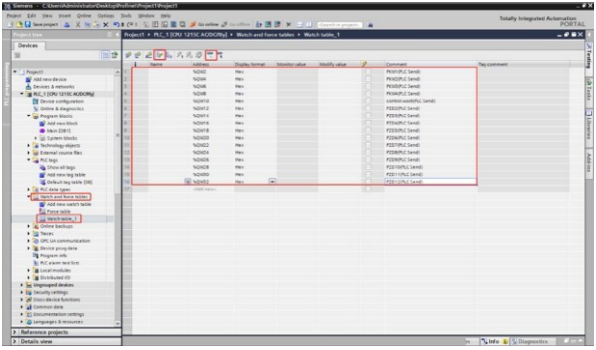


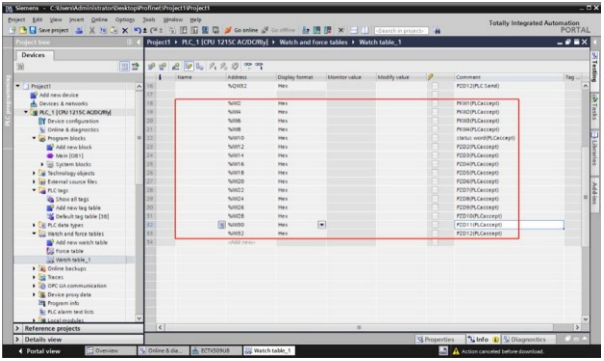
2.5.7 Monitoring VFD parameters

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



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





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

3 EtherNet IP protocol

3.1 Overview

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

3.2 Product features

3.2.1 Supported functions

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

3.2.2 Supported communication types

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

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

used for configuring, downloading, and troubleshooting; implicit messages are used for real-time I/O data transmission.

3.2.3 Status indicator

The EtherNet IP communication card provides four indicators to indicate its states. For details, seeTable 3-1.

Table 3-1 Indicator description

Indicator	Color	Definition	Function
RUN	Green	Steady on	The communication between the communication card and the PLC is online, and data exchange is allowed.
		Blinking (On: 500ms; Off: 500ms)	Abnormal setting of the IP address for either the communication card or the PLC.
		Steady off	The communication between the communication card and PLC is not in Online state.
L/A IN (HOST)	Green	Steady on	The communication card is in the process of handshaking with the VFD.
		Blinking (On: 500ms; Off: 500ms)	The communication card and VFD communicate normally. **Note:** After the handshaking is completed, it should blink regardless of whether there is data transmission between the communication card and the main control board.
		Steady off	The communication card is in the initialization or parameter configuration phase.
L/A OUT (DATA)		Blinking (On: 500ms; Off: 500ms)	The data update between the communication card and main control board is normal.

Indicator	Color	Definition	Function
	Green	Steady off	No data update or abnormal update between the communication card and main control board.
ERR	Red	Steady on	Failed to set up data communication between the communication card and PLC.
		Blinking (On: 500ms; Off: 500ms)	Incorrect PLC configuration.
		Blinking (On: 250ms; Off:250ms)	The communication card failed to send the data to the PLC.
		Blinking (On: 125ms; Off: 125ms)	The connection between the communication card and PLC timed out.
		Steady off	No fault

3.3 Electrical connection

The EtherNet IP communication card adopts standard RJ45 interfaces, which can be used in a linear network topology, star network topology, or ring network topology. The electrical connections are shown inFigure 3-1, Figure 3-2, andFigure 3-3.

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

Figure 3-1 Linear network topology electrical connection

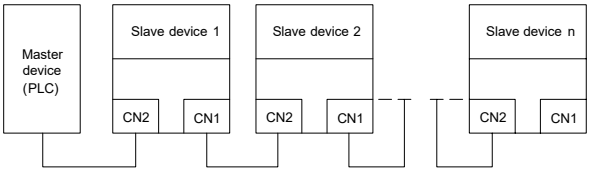
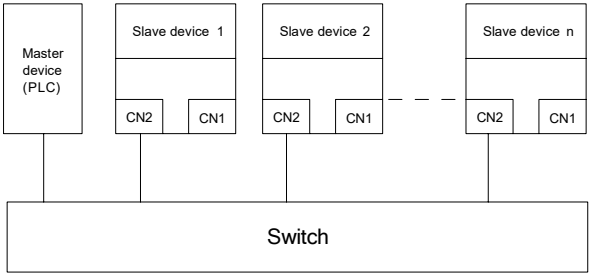
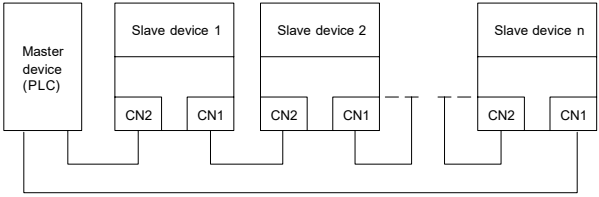


Figure 3-2 Star network topology electrical connection



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

Figure 3-3 Ring network topology electrical connection



3.4 Communication

3.4.1 Communication settings

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

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

The factory IP address and subnet mask of each communication card are

192.168.0.20 and 255.255.255.0 respectively, which can be changed to a network segment address according to the actual requirements.

2. Set the control method.

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

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

3.4.2 Message format

The TCP communication message is shown inTable 3-2.

Table 3-2 TCP communication message

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

The UDP communication message is shown inTable 3-3.

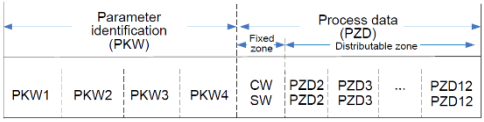
Table 3-3 UDP communication message

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

3.4.3 Communication

The EtherNet IP communication card supports 16-word input/output. Figure3-4 shows the message format for transmitting data with the VFD.

Figure3-4 Message structure



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

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—control word (from master to slave; seeTable 3-4)

SW—status word (from slave to master; seeTable 3-7)

PZD—Process data (user specified)

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

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

Control word (CW) and status word (SW)

Using CWs is the basic method of the fieldbus system to control the VFD. A CW is sent from the fieldbus master to a VFD device. In this case, the adapter module

functions as a gateway. The VFD device responds to the bit code information of the CW and feeds status information back to the master through an SW.

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

Actual value: An actual value is a 16-bit word that includes information about VFD device operation. The monitoring function is defined through VFD parameters. The conversion scale of the integer transmitted to the master as the actual value depends on the selected function. For details, see the related VFD user manual.

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

Task message (Master -> VFD)

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

When P14.71=0 (CW defined in decimal),Table 3-4 provides the descriptions for ZS28 series VFD CWs in decimal.

Table 3-4 ZS28 series VFD CWs in decimal

Bit	Name	Value	Description
0–7	Communication-based control command	1	FWD run
		2	REV run
		3	FWD jog
		4	REV jog
		5	Stop
		6	Coast to stop
		7	Fault reset
		8	Jog stop
		9	Stop in emergency manner.
8	Enable read and write	1	Enable read and write (PKW1–PKW4)
9–10	Motor group setting	0	Select motor 1

Bit	Name	Value	Description
		1	Select motor 2
11	Control mode switchover selection	0	No switchover
		1	Enable the switchover between torque
12	Clear power consumption quantity	0	Disable the function for resetting power consumption to zero
		1	Enable the function for resetting power consumption to zero
13	Pre-excitation	0	Disable pre-excitation
		1	Enable pre-excitation
14	DC braking	0	Disable DC braking
		1	Enable DC braking
15	Heartbeat reference	0	Disable heartbeat
		1	Enable heartbeat

When P14.71=1 (CW defined in binary),Table 3-5provides the descriptions for ZS28 series VFD CWs in binary.

Table 3-5 ZS28 series VFD CWs in binary

Bit	Name	Description	Priority
0	FWD run	0: Decelerate to stop 1: Run forward	1
1	REV run	0: Decelerate to stop 1: Run reversely	2
2	Fault reset	0: None 1: Fault reset	3
3	Coast to stop	0: None 1: Coast to stop	4
4	Forward jogging	0: None 1: Forward jogging	5
5	Reverse jogging	0: None 1: Reverse jogging	6
6	Jog stop	0: None 1: Stop jogging	7
7	-	Reserved	-
8	Enable read and write (PKW1–4)	0: None 1: Enable read and write	-
9	-	Reserved	-
10	Stop in emergency manner.	0: None 1: Emergency stop	0: Top priority
11–15	-	Reserved	-

Reference value (REF): The second to twelfth words in a PZD task message are the main settings (REF). The main frequency settings are provided by the main setting signal source. Table 3-6 lists the settings of ZS28 series VFD for reference.

Table 3-6 Settings of ZS28 series VFD

Function code	Word	Value range	Default
P23.02	Received PZD2	0–31 0: Invalid 1: Set frequency (0–Fmax, unit: 0.01Hz) 2: PID reference (-1000–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (-1000–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the motor rated current) 5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 8: Upper limit of braking torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 9: Virtual input terminal command (0x000–0x7FF) 10: Virtual output terminal command (0x000–0x01F)	0
P23.03	Received PZD3		0
P23.04	Received PZD4		0
P23.05	Received PZD5		0
P23.06	Received PZD6		0
P23.07	Received PZD7		0
P23.08	Received PZD8		0
P23.09	Received PZD9		0
P23.10	Received PZD10		0
P23.11	Received PZD11		0

Function code	Word	Value range	Default
P23.12	Received PZD12	11: Voltage setting special for V/F separation (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage) 12: AO output setting 1 (0–1000, in which 1000 corresponds to 100.0%) 13: AO output setting 2 (-1000–1000, in which 1000 corresponds to 100.0%) 14–18: Reserved 19: Function parameter mapping (PZD2–PZD12 correspond to P14.49–P14.59) 20–31: Reserved	0

Response message (VFD -> Master)

Status word (SW): The first word in a PZD response message is the status word (SW) of VFD. When P14.71=0(SW defined in decimal), the VFD SW definitions are as follows.

Table 3-7 ZS28 series VFD SWs in decimal

Bit	Name	Value	Description
0–7	Running status	1	Running forward
		2	Running reversely
		3	VFD stop
		4	VFD in fault
		5	VFD POFF status
8	Bus voltage established	0	Not ready for running
		1	Ready to run
9–10	Motor group feedback	0	Feedback of motor 1
		1	Feedback of motor 2
11	Motor type feedback	0	Asynchronous motor (AM)
		1	Synchronous motor (SM)
12	Overload pre-alarm feedback	0	No pre-alarm upon overload
		1	Pre-alarm upon overload
13–14	Running mode selection	0	Keypad-based control
		1	Terminal-based control

Bit	Name	Value	Description
		2	Communication-based control
		3	Reserved
15	Enable	0	No heartbeat feedback
		1	Enable

When P14.71=1(SW defined in binary), the VFD SW definitions are as follows.

Table 3-8 ZS28 series VFD SWs in binary

Bit	Name	Description	Priority
0	FWD run	0: No 1: Running forward	1
1	REV run	0: No 1: Running reversely	2
2	Stop	0: None 1: VFD in stopped state	3
3	Fault	0: None 1: VFD in fault	4
4	POFF	0: None 1: VFD in POFF state	5
5	Pre-exciting	0: None 1: VFD in pre-exciting state	6
6–15	-	Reserved	-

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

Table 3-9 Actual status values of ZS28 series VFD

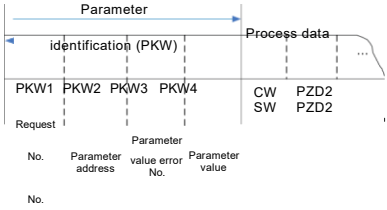
Function code	Word	Value range	Default
P23.13	Sent PZD2	0–32 0: Invalid 1: Running frequency (×100, Hz) 2: Set frequency (×100, Hz) 3: Bus voltage (×10, V) 4: Output voltage (×1, V) 5: Output current (×100, A) 6: Actual output torque (×10, %) 7: Actual output power (×10, %) 8: Rotation speed of running (×1, RPM) 9: Linear speed of running (×1, m/s) 10: Ramp reference frequency (×100, Hz) 11: Fault code	0
P23.14	Sent PZD3		0
P23.15	Sent PZD4		0
P23.16	Sent PZD5		0
P23.17	Sent PZD6		0
P23.18	Sent PZD7		0
P23.19	Sent PZD8		0
P23.20	Sent PZD9		0
P23.21	Sent PZD10		0
P23.22	Sent PZD11		0

Function code	Word	Value range	Default
P23.23	Sent PZD12	12: AI1 input (×100, V) 13: AI2 input (×100, V) 14: AI3 input (×100, V) 15: Reserved 16: HDI1 frequency value (×100, kHz) 17: Reserved 18: Terminal input state 19: Terminal output status 20: PID reference (×100, %) 21: PID feedback (×100, %) 22–26: Reserved 27: VFD status word 2 28–31: Reserved 32: Function parameter mapping (PZD2–PZD12 correspond to P14.60–P14.70)	0

PKW zone

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

Figure 3-5 PKW zone



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

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

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

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

Table 3-10 Definitions of the task identification flag PKW1

Request (from master to slave)		Response signal	
Request code	Function	Positive acknowledgment	Negative acknowledgment
0	No task.	0	-
1	Requesting the value of a parameter	1, 2	3
2	Modifying a parameter value (one word) [modifying the value only on RAM]	1	3, 4
3	Modifying a parameter value (two words) [modifying the value only on RAM]	2	3, 4

Request (from master to slave)		Response signal	
Request code	Function	Positive acknowledgment	Negative acknowledgment
4	Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM]	1	3, 4
5	Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM]	2	3, 4

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

Table 3-11 Definitions of the response identification flag PKW1

Response (from slave to master)	
Response code	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 code is returned: 1: Invalid command 2: Invalid data address 3: Invalid data value 4: Operation failure 5: Incorrect password 6: Incorrect data frame 7: Parameter read only 8: Parameter cannot be modified in running 9: Password protection 10: Function code mapping operation failure
4	Reserved

Mode specified by the standard ODVA protocol

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

Table 3-12 Standard ODVA protocol specified transmission mode

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

SeeTable 3-13, Table 3-14, Table 3-15, andTable 3-16for the definitions of CW1, SW1, CW2, and SW2.

Note: The ODVA protocol messages do not support P14.71=1 (in binary).

Table 3-13 Standard ODVA protocol specified CW1

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

Table 3-14 Standard ODVA protocol specified SW1

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

Table 3-15 Standard ODVA protocol specified CW2

Bit	Name	Value	Description
0	FWD run	0	FWD run disabled
		1	FWD run

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

Table 3-16 Standard ODVA protocol specified extended SW2

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

Z:SYSTEMS extended data modes based on ODVA protocol

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

Table 3-17 Z:SYSTEMS extended data modes based on ODVA protocol

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

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

3.5 PLC communication example 1 (1769_L36ERMS)

The following example shows the configuration for using an EtherNet IP adapter module to communicate with an Allen-Bradley PLC (model: 1769_L36ERMS) (based on Studio 5000 software).

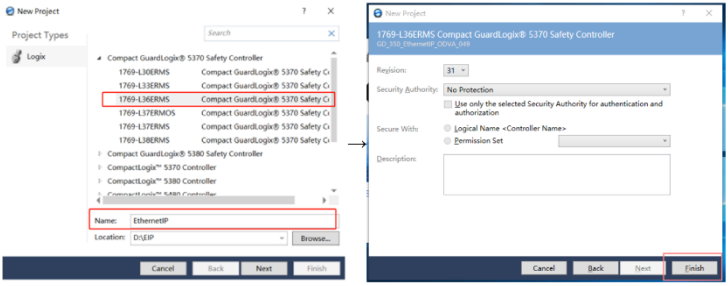
3.5.1 Creating a project

Use a printer cable or Ethernet cable to connect the computer and PLC, open the 

software, and click **New Project**.



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



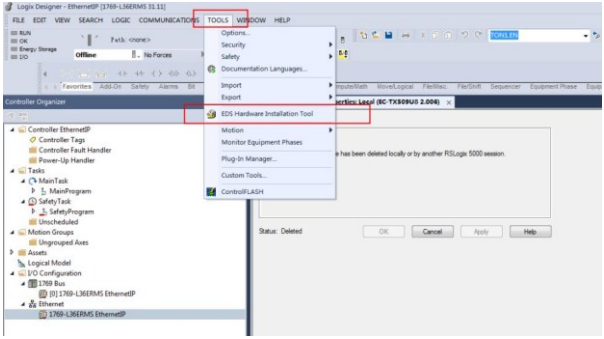
3.5.2 Importing the EDS file

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

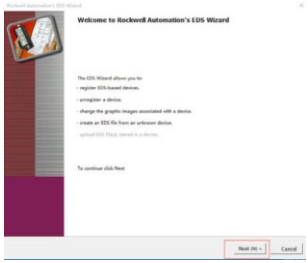
You can request the EDS file of the card from the supplier, or download it from our website.

Website: www.z:systems.com, file name: **ZSY-TX149_1.0.0.0.eds**

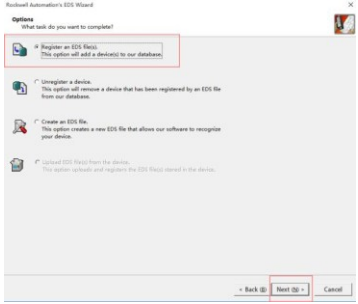
Right click **TOOLS** and choose **EDS Hardware Installation Tool**.



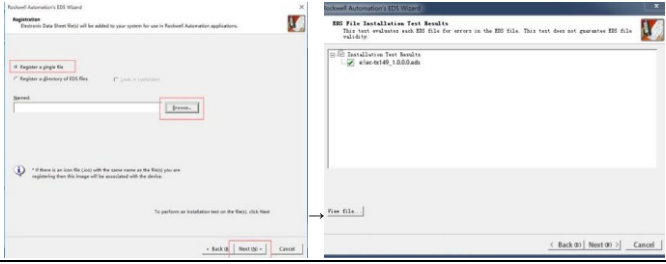
Click **Next**.



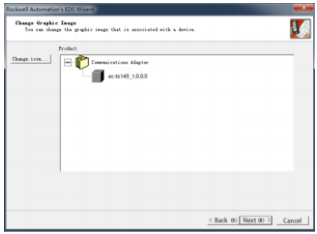
Select the method shown in the following figure and click **Next**.



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

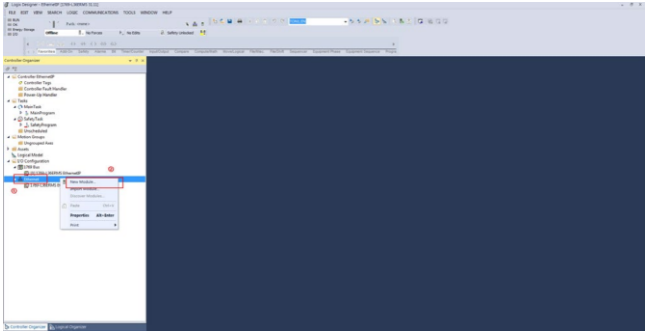


Click **Next** to complete the installation.

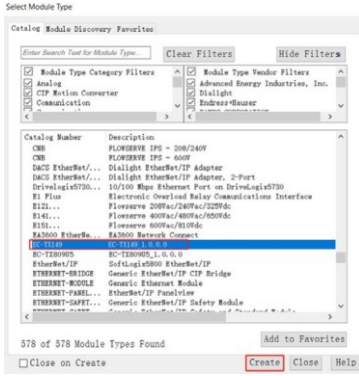


3.5.3 Creating a device object

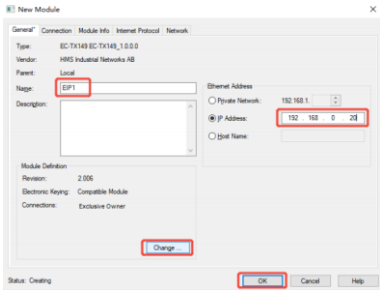
On the left, choose **I/O Configuration**, right click **Ethernet** option, then right click and choose **New Module**.



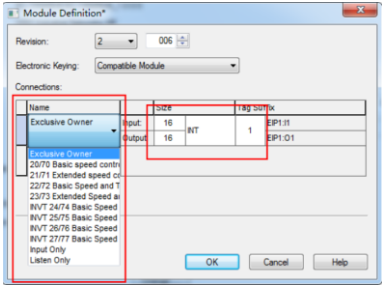
Choose **ZSY-TX149** and click **Create**.



Fill in the module name, and set the module IP address, which needs to be consistent with P24.37–P24.40 on the ZS28 VFD; otherwise communication will fail.

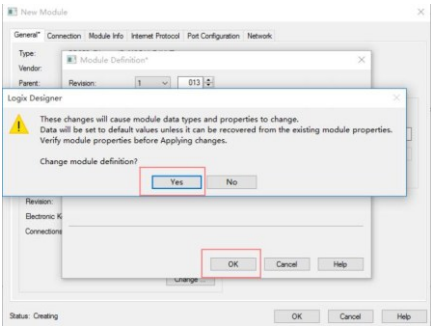


Click **Change** in the preceding figure, and select the protocol type adopted by the module; the IO format of each type is different and needs to be correspondingly selected, as shown in the following table. "Exclusive Owner" is used as an example for explanation.

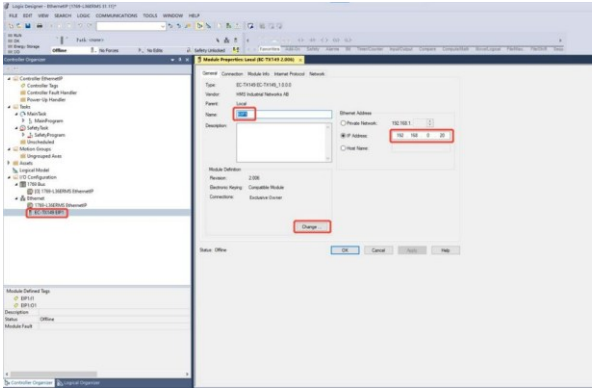


Name	Size	Format
Exclusive Owner	16	INT
20/70 Basic speed control	2	INT
21/71 Extended speed control	2	INT
22/72 Basic Speed and Torque control	3	INT
23/73 Extended Speed and Torque control	3	INT
Z:SYSTEMS 24/74 Basic Speed Control plus Drive Parameters	12	INT
Z:SYSTEMS 25/75 Basic Speed Control plus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

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



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

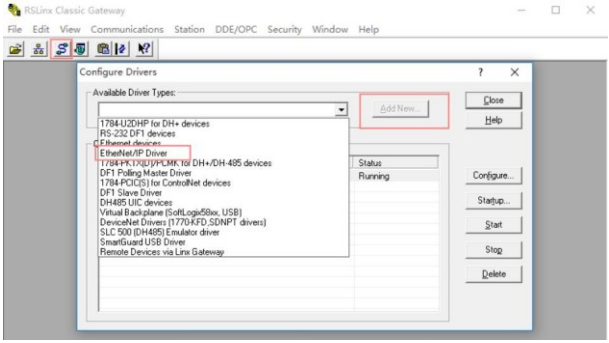


3.5.4 Using RSLinx Classic

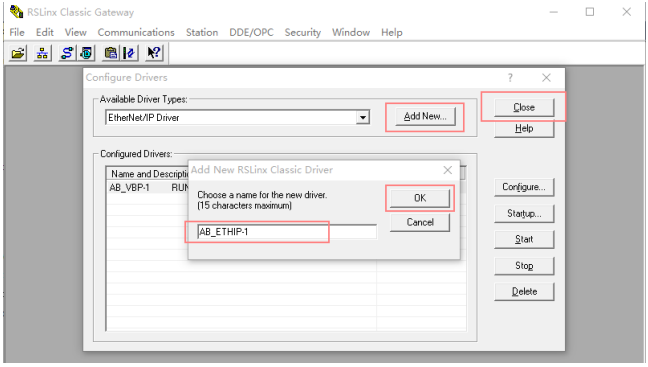
This software is used for the connection between the PC and PLC. Open the Rslinx Classic software.



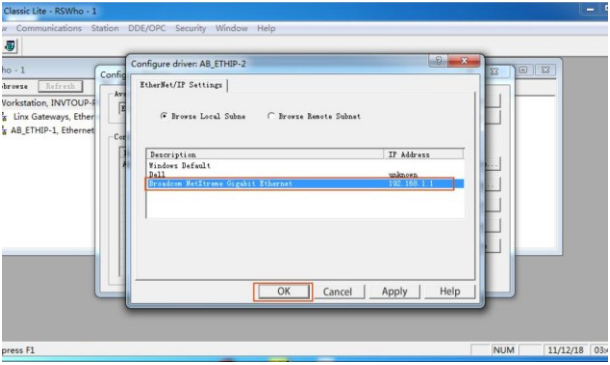
Click the icon. Choose **EtherNet/IP Driver**, and click **Add New ...** In the pop-up **Configure Drivers** window, select **Ethernet/IP Driver** from the **Available Driver Types** dropdown menu.



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

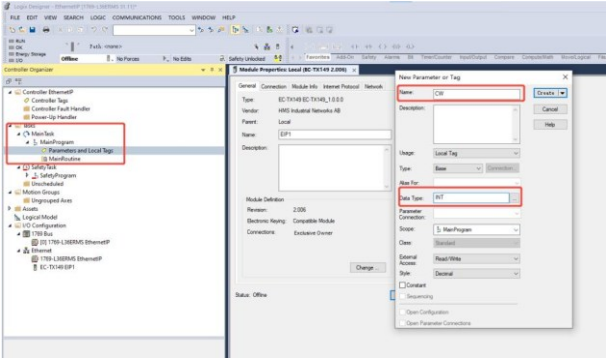


In the pop-up **Configure driver** window, select the computer network card and click **OK**.

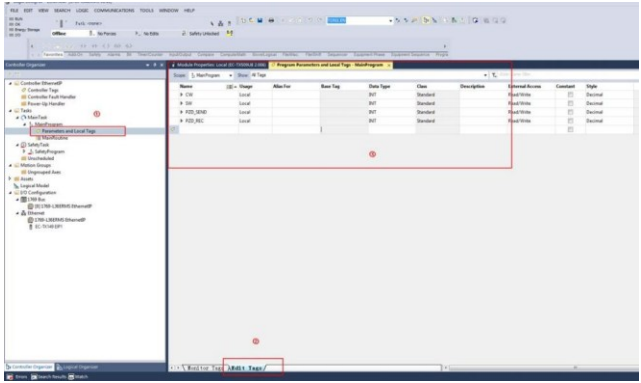


3.5.5 Writing PLC programs

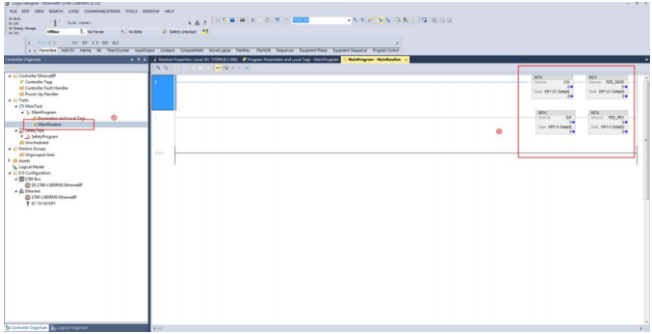
On the left, choose **Tasks** > **MainTask** > **MainProgram** > **MainRoutine** to enter the program editing interface. Right clicking **Parameters and Local Tag** can create global variables.



Example: Create four variables.



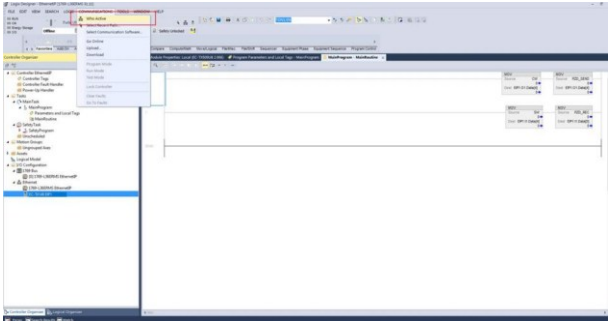
Double click **MainRoutine** and write the following program in the program interface.



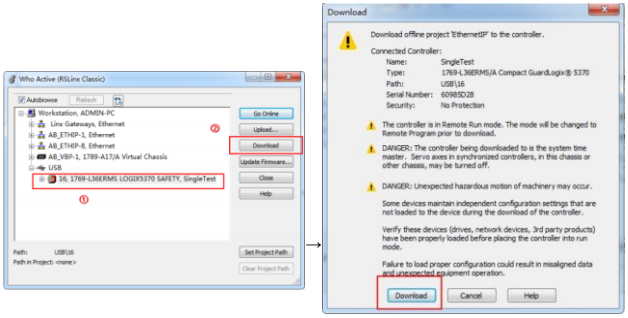
3.5.6 Host controller connection and program download

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

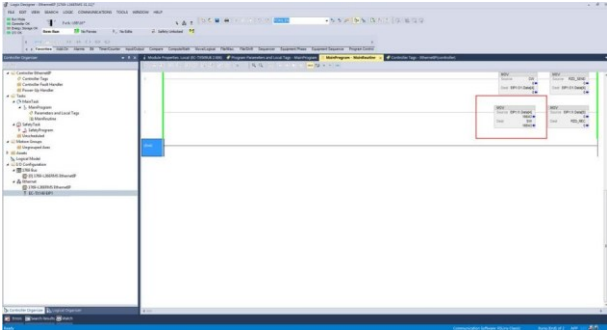
Note: The PLC position cannot be set to "RUN" at this time.



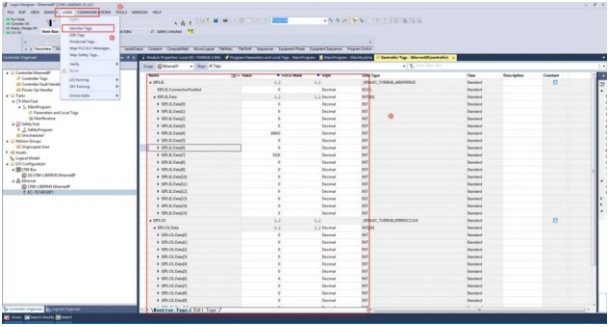
In the pop-up dialog box, select the connected PLC, and click **Download**; wait for the download to complete.



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

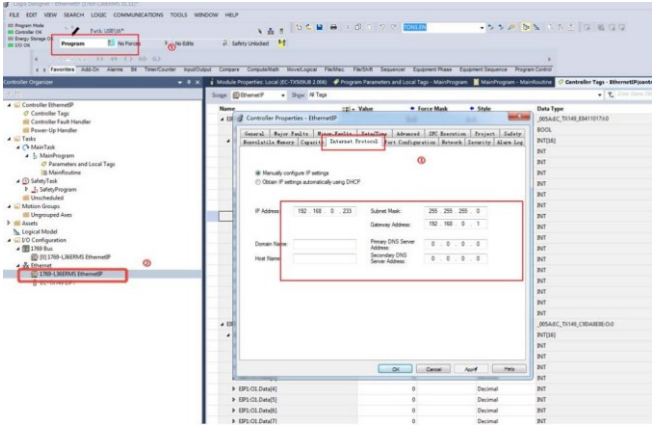


Choose **Logic** > **Monitor Tags** to monitor the real-time data sent by the PLC and uploaded by the VFD.



3.5.7 Setting the PLC IP address using Studio 5000 V31

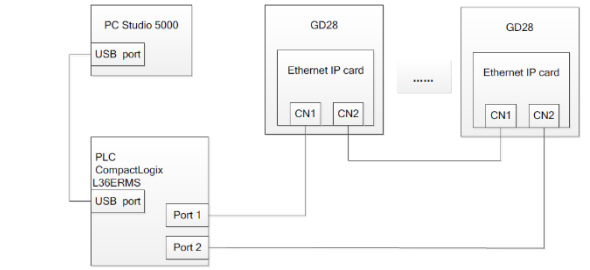
Make sure the PLC is in REM or PROG mode, first click on **1769-L36ERMS** at the bottom left, enter the **Controller Properties** interface, then click **Internet Protocol** to change the PLC IP address.



3.5.8 DLR ring network configuration

1. Set up using Logix Designer.

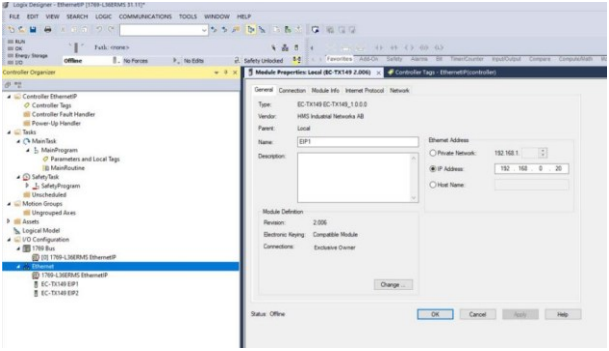
   Open the Studio 5000 software, and use Allen-Bradley CompactLogix PLC equipped with ring networking capability. At least two ZS28 EtherNet IP communication cards are required. More ZS28 EtherNet IP communication cards can be added, but it is recommended to use up to 32 nodes on the DLR ring network. The connection method is shown in the following figure.



Note: The EDS file must be added.

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

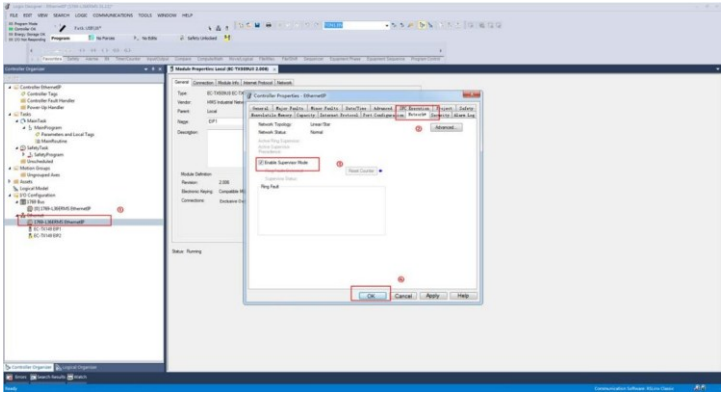
   The adding method is the same as the line or star connection method.



3. Enable the PLC ring network monitor function.

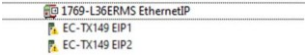
   Double click **1769-L36ERMS EtherNet IP** under the **I/O Configuration** folder, as shown in the following figure.

   Go to **Controller Properties** and select **Network***, choose **Enable Supervisor Mode**, and then click **OK**.



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

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



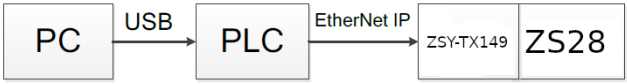
5. Download the project to the PLC; make the PLC online and in the programming mode.

3.6 PLC communication example 2 (NJ501-1400)

The following example shows the configuration for using an EtherNet IP adapter module to communicate with an ORMON PLC (model: NJ501-1400) (based on the Sysmac Studio software).

3.6.1 Hardware connection

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



3.6.2 Network Configurator software setting

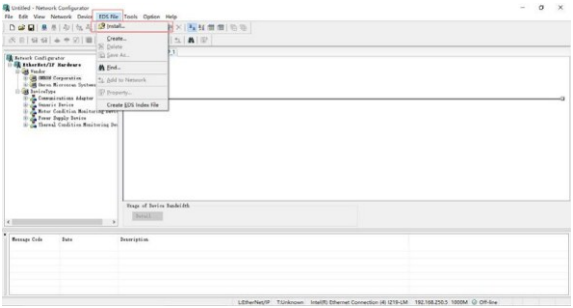
3.6.2.1 Starting the Network Configurator software

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

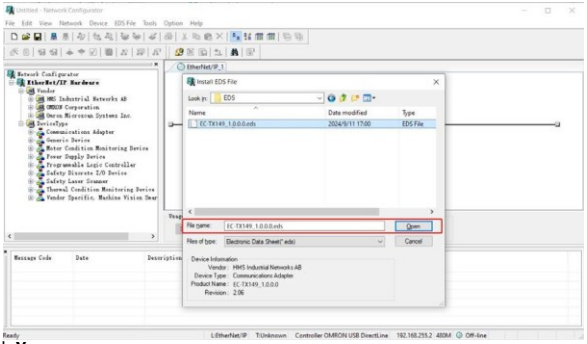
C:\Program Files (x86)\OMRON\CX-One\Network Configurator\Program\NetConfigurator.exe

3.6.2.2 Uploading the EDS file

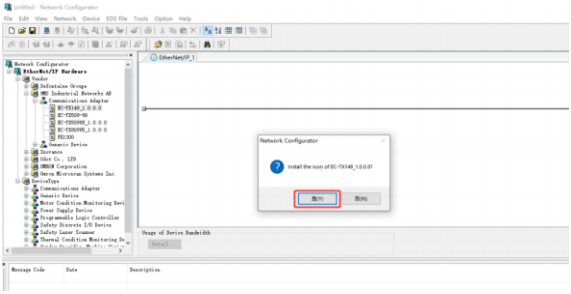
Choose **EDS File > Install**.



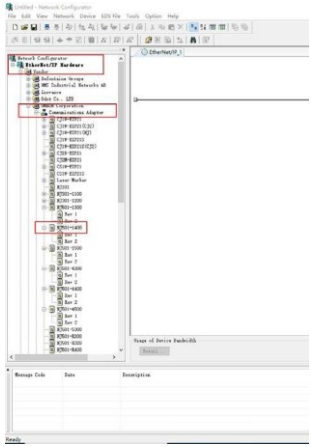
Add the EDS file **ZSY-TX149_1.0.0.0.eds**, and click **Open**.



Click **Yes**.

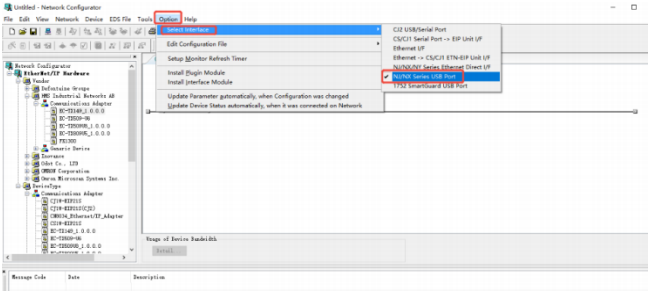


Add devices **NJ501-1400** and **ZSY-TX149_1.0.0.0** to the EtherNet IP bus at the following location. After completion, two devices will be displayed on the bus. The default IP addresses are **192.168.250.1** and **192.168.250.2**. Modify the ZS28 VFD function codes P24.37–P24.40 to 192.168.250.2.

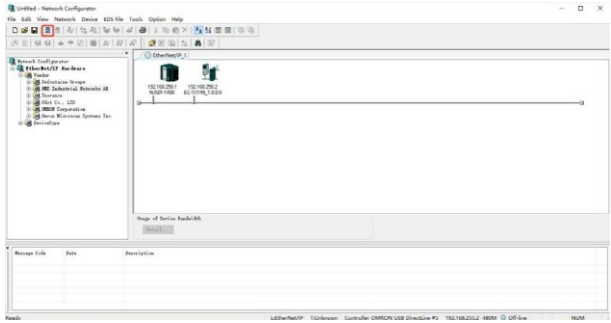


3.6.2.3 Connection setting

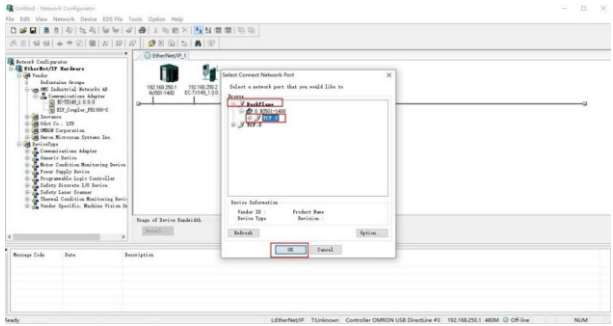
Choose **Option** > **Select Interface** > **NJ/NX Series USB Port**.



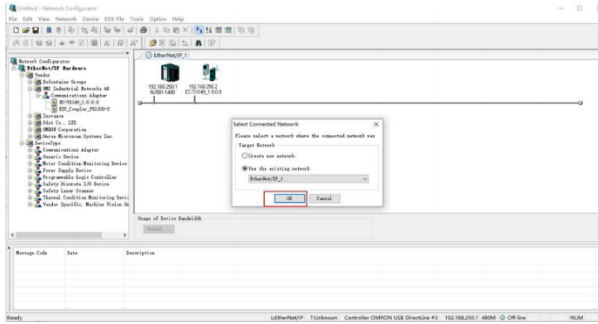
Click the icon for connection.



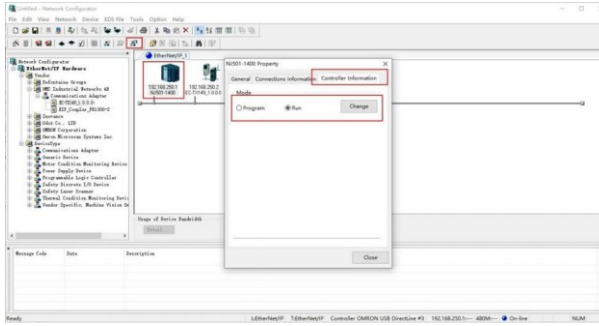
Choose **Backplane** > **0 NJ501-1400** > **TCP:2**, and then click **OK**.



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

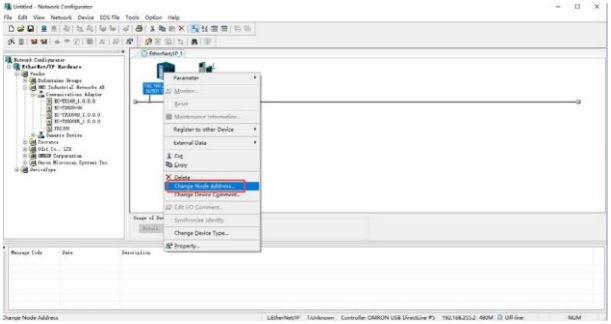


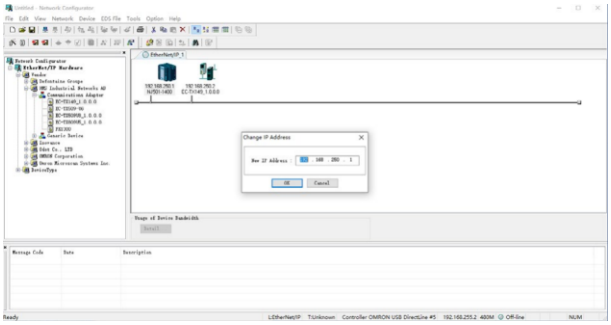
After the connection is successful, the blue indicator above the PLC device icon will turn on. Choose the PLC, click the icon for **Device Property**. On the **Controller Information** tab in the pop-up interface, you can switch between the **Program** and **Run** states for the PLC.



3.6.2.4 Changing the IP address

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

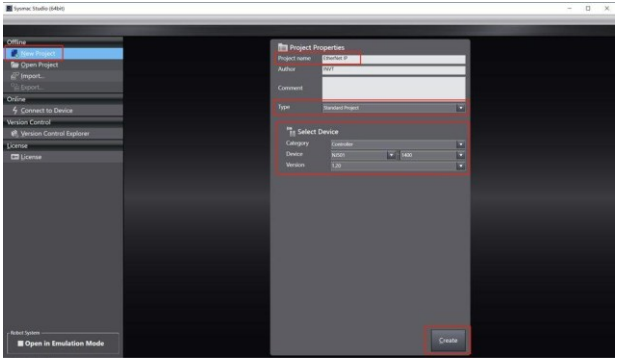




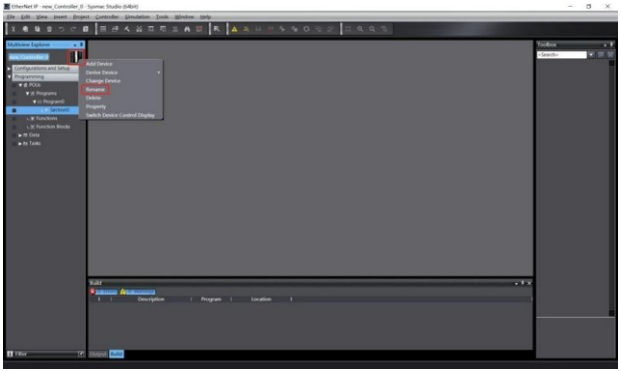
3.6.3 Configuring Sysmac Studio software

3.6.3.1 Creating a project

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

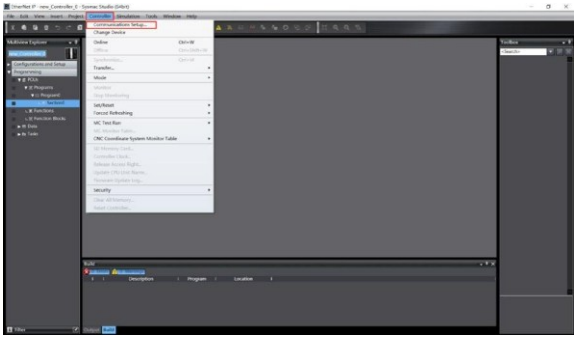


After the creating, enter the following interface, right-click the device icon, and choose **Rename** to change the device name (it can also remain unchanged).

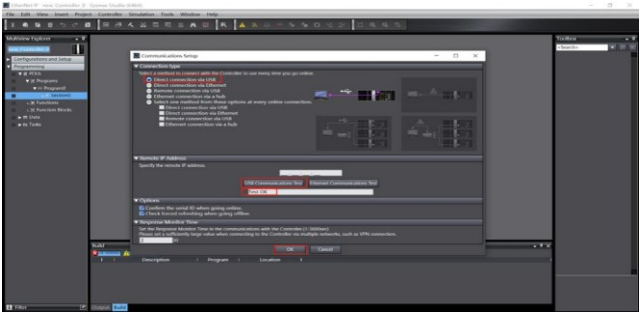


3.6.3.2 Connection setting

Choose **Controller > Communications Setup**.



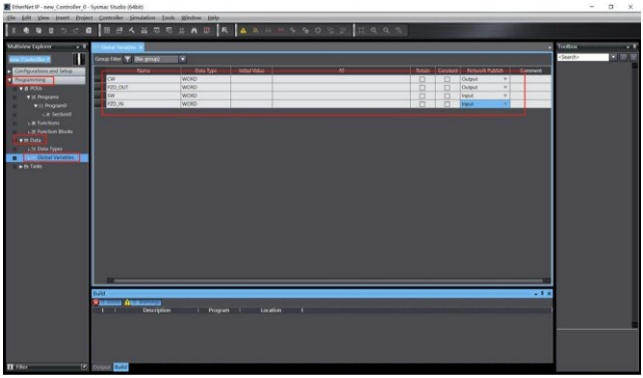
Choose **USB Direct connection via USB**, and click **USB Communications Test**. The status bar displays **Test OK**. Click **OK**.



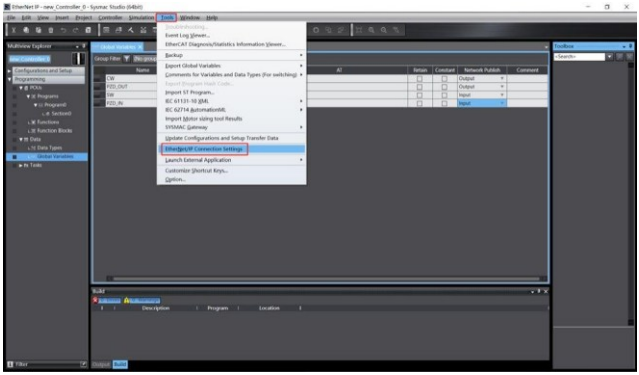
3.6.3.3 Setting data tags

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

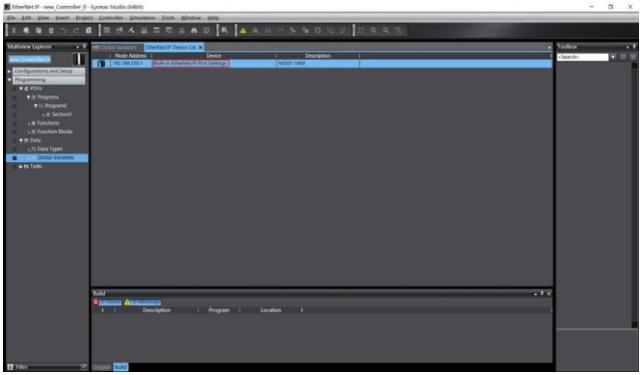
In this example, the "ODVA Basic speed control assembly" transmission mode is used to create four global variables.



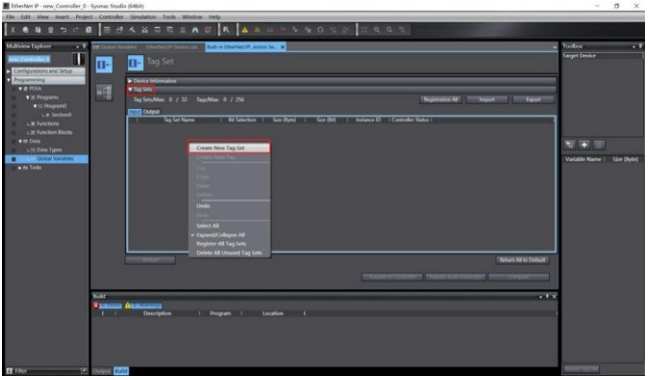
In the top menu bar, choose **Tools** > **EtherNet/IP Connection Settings**.



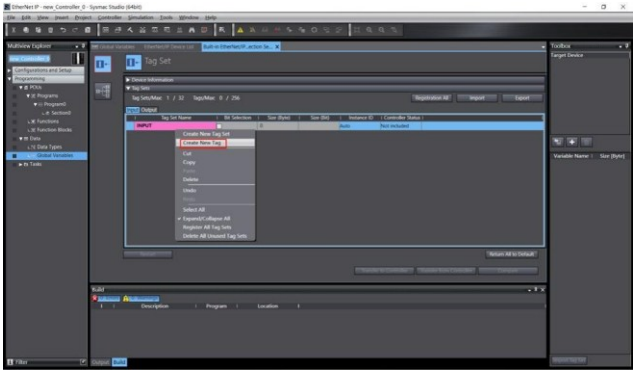
Double click **Built-in EtherNet/IP Port Settings**.

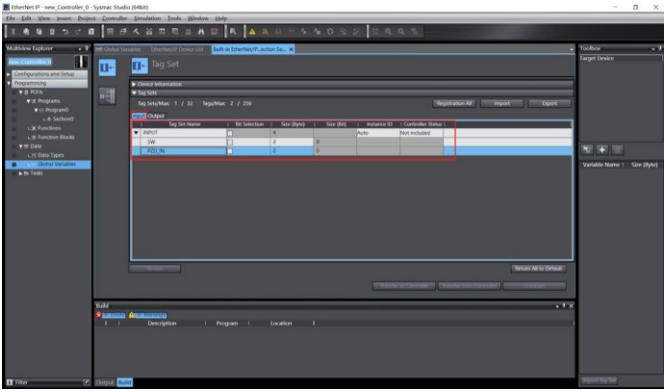


Right click on the blank area under **Tag Set**, and choose **Create New Tag Set**.

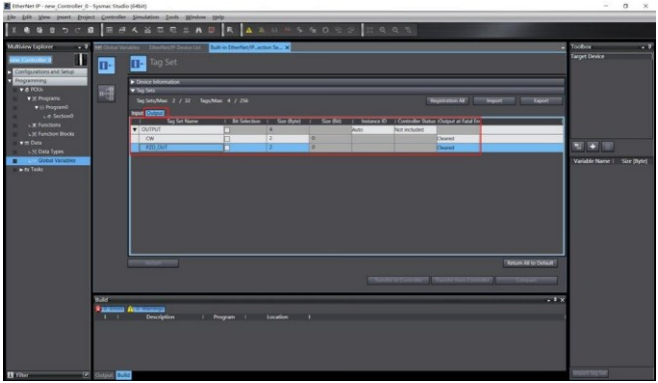


Enter the tag set name as **INPUT**, right click the tag set and choose **Create New Tag**, and add input global variables to the tag set **INPUT**. Pay attention to the order of data arrangement.





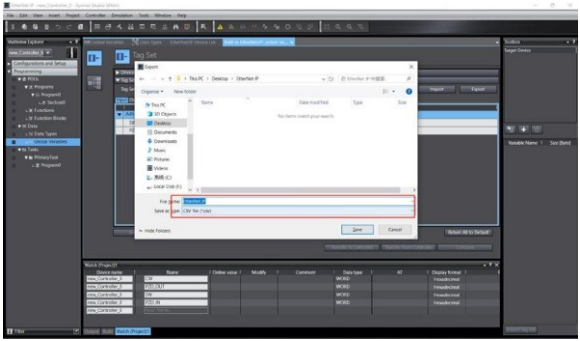
Similarly, set the tag set **OUTPUT** and output tags.



3.6.4 Importing or exporting data tags

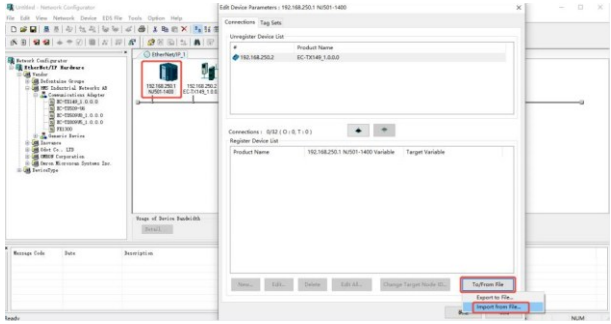
3.6.4.1 Exporting data tags from Sysmac Studio

After data tags are set, click **Export** to export the data tags to the local and save as **EtherNet IP.csv**.

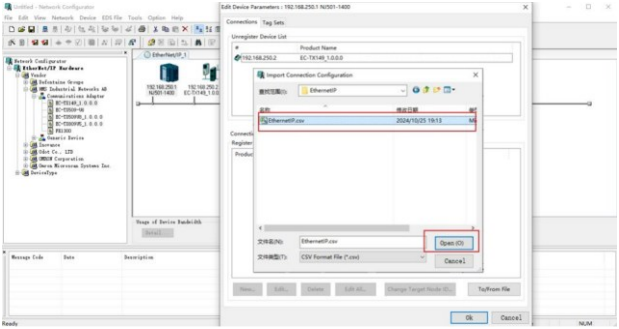


3.6.4.2 Importing data tags to the Network Configurator

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

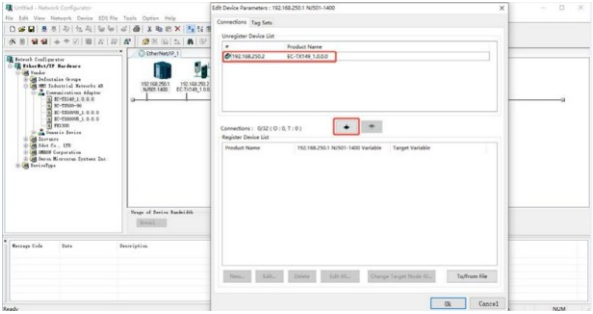


Select the **EtherNet IP.csv** file exported from Sysmac Studio, and click **Open**.

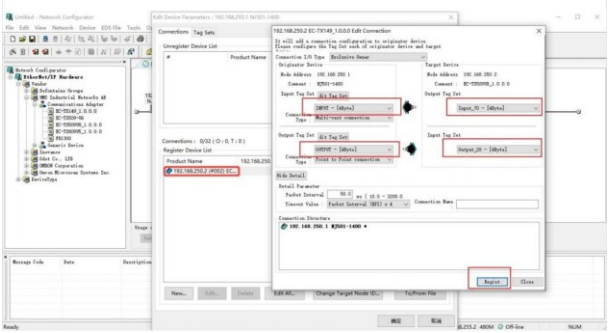


3.6.4.3 Connection corresponding data tags

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



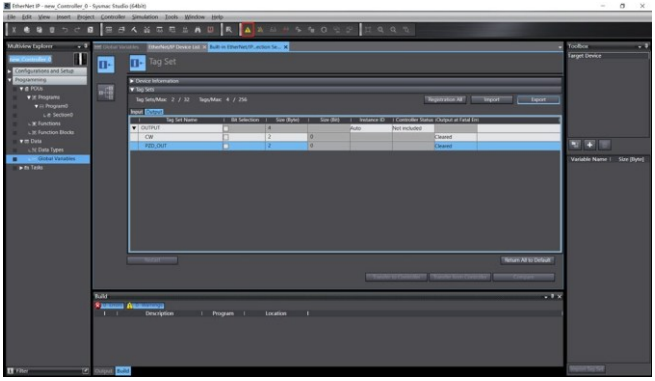
Double click the device **192.168.250.2**, set the data input and output tags,and click **Regist.**



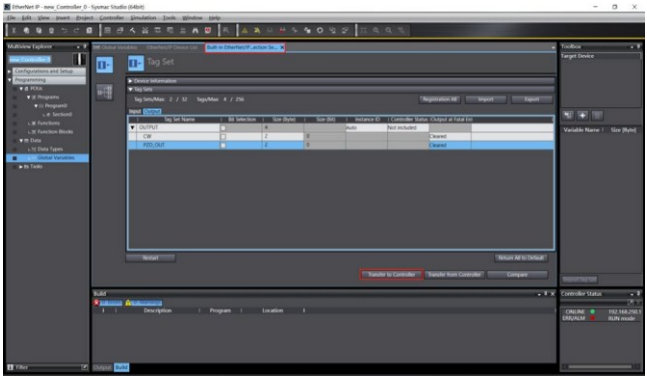
3.6.5 PLC program downloading and online monitoring

3.6.5.1 Downloading from Sysamc Studio

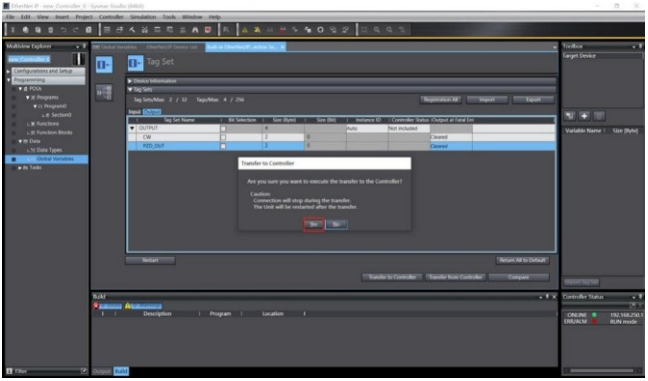
Click the online button.



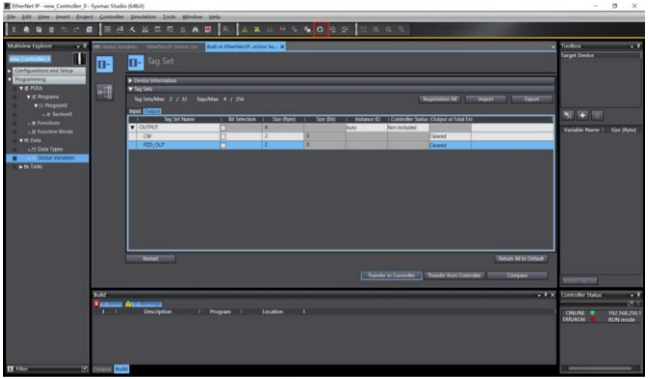
On the **Built-in EtherNet/IP Port Settings** tab, click **Transfer to Controller**.



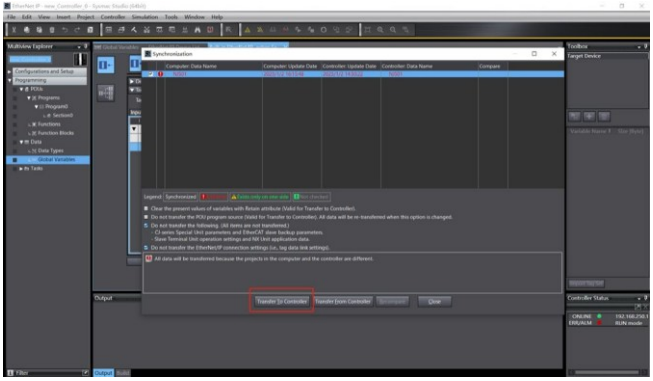
Click **Yes**.



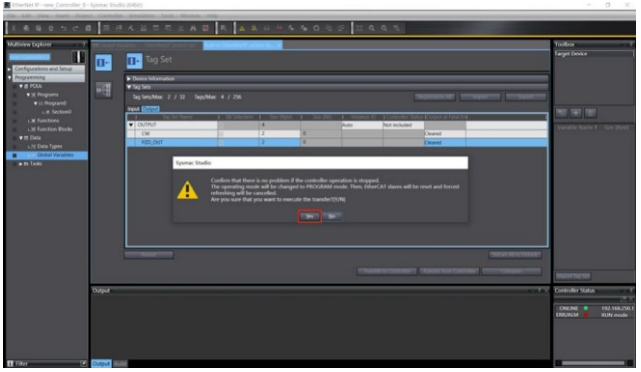
Click the synchronization button.



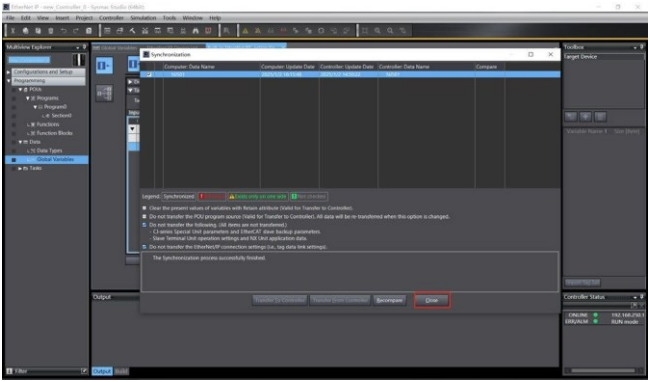
Select device **NJ501-1400**, and click **Transfer to Controller**.



Click **Yes**.

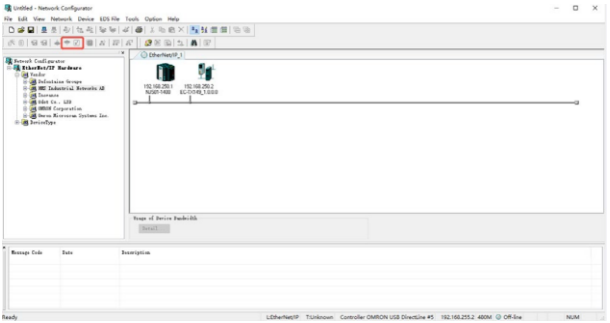


At this time, there are two green lights under **Controller Status** in the bottom right corner. Click **Close**.

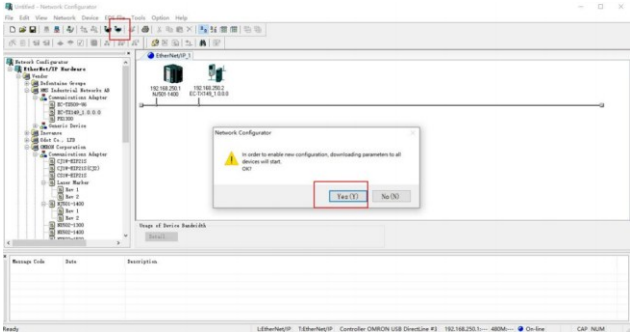


3.6.5.2 Downloading from Network Configurator

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

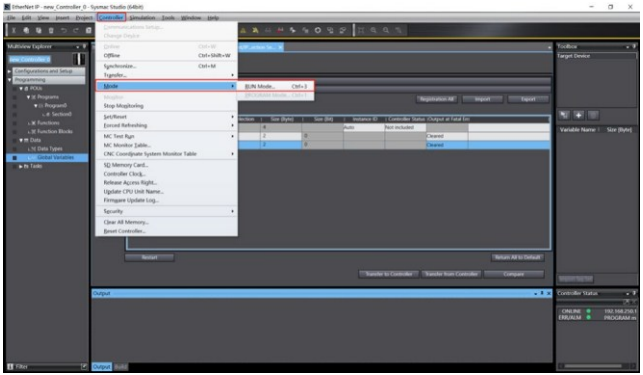


Click the **Download to Network** icon. Click **Yes**.

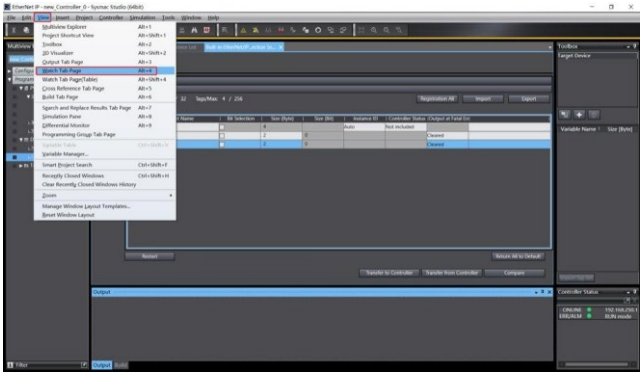


3.6.5.3 Online monitoring on Sysamc Studio

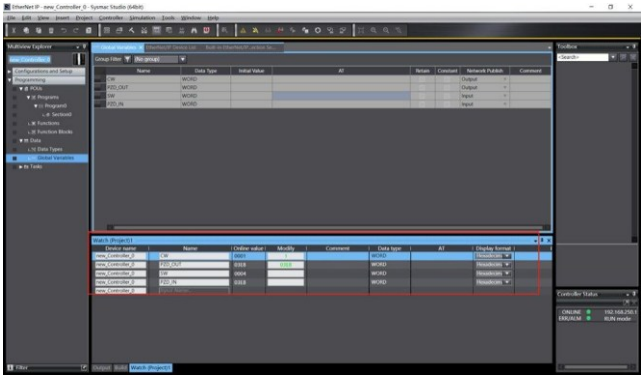
Choose **Controller** > **Mode (or Run)** to switch the PLC to the Run mode, then click **Yes**.



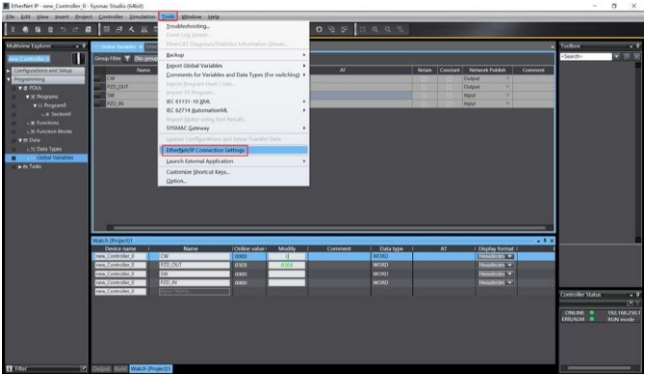
Choose **View** > **Watch Tab Page**.



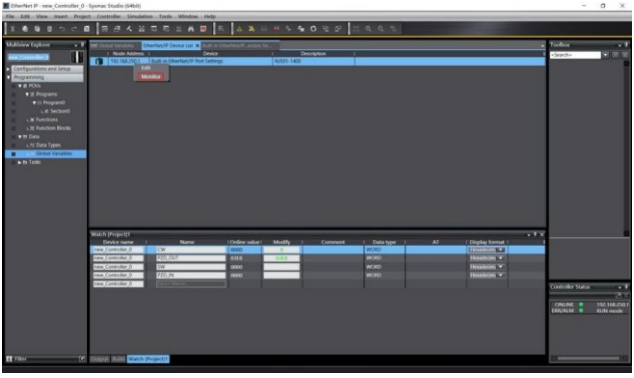
Enter the variable name in the **Watch** window to monitor the variable value; you can change the variable value in real time in the **Modify** column.



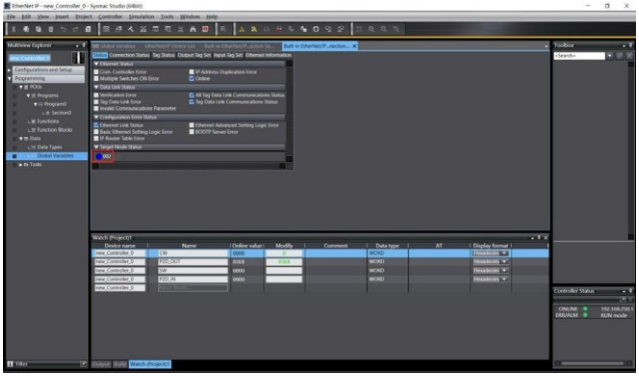
In the top menu bar, choose **Tools** > **Ethernet/IP Connection Settings**.



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



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



4 EtherCAT protocol

4.1 Overview

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

4.2 Product features

4.2.1 Supported functions

- Supports the CiA301 and CiA402 CoE protocols. Configured with a slave XML configuration file, it can communicate with Beckhoff PLC, Z:SYSTEMS controllers, and other master devices.
- Automatic network address configuration
- Equipped with two RJ45 ports, designated for IN and OUT directions.
- Applicable to linear, star, and ring network topologies.

4.2.2 Supported services

- PDO service
- SDO service
- Use of SDO to read and write VFD function codes
- Manufacturer-defined object dictionary
- Two types of process objects for VFD reading and writing:
  - PDO: A real-time, cyclic data transmission mechanism used for fast exchange of process variables such as speed, torque, and current. Data is transmitted through predefined mapping relationships without requiring additional communication command configuration.
  - SDO: Used for non-cyclic, point-to-point parameter read/write. By accessing specific indexes and subindexes in the object dictionary (OD), it allows reading and writing of manufacturer-defined objects, such as fault codes and advanced parameters.

Supported EtherCAT synchronization cycles

Table 4-1 Supported EtherCAT synchronization cycles

Item	Supported specification
Synchronization cycle	1ms
	2ms

4.2.3 Status indicator

The EtherCAT communication card provides four indicators to indicate its states, as shown inFigure 4-1. Table 4-2provides the indicator definitions.

Figure 4-1Status indicator positions

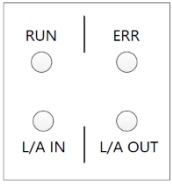


Table 4-2 Status indicator definitions

Indicator	Color	Definition	Function
RUN	Green	Steady on	In OP state.
		Blinking (On: 200ms; Off: 200ms)	In PreOP state.
		Single flash (on for 200ms, off for 1s)	In SafeOP state.
		Steady off	In Init state.
L/A IN	Green	Steady on	IN Link established, without data transmission.
		Blinking (On: 50ms; Off: 50ms)	IN Link established, with data transmission.
		Steady off	IN LINK not established.
L/A OUT	Green	Steady on	OUT Link established, without data transmission.
		Blinking (On: 50ms; Off: 50ms)	OUT Link established, with data transmission.
		Steady off	OUT LINK not established.
ERR	Red	Steady on	The OP fault occurred.
		Blinking (On: 200ms; Off: 200ms)	The Init/Preop fault occurred.
		Single flash (on for 200ms, off for 1s)	The Safeop fault occurred.
		Steady off	No fault

4.3 Electrical connection

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

Figure 4-2 Linear network topology electrical connection

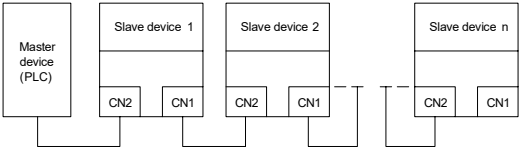
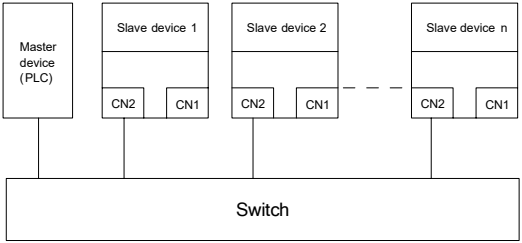
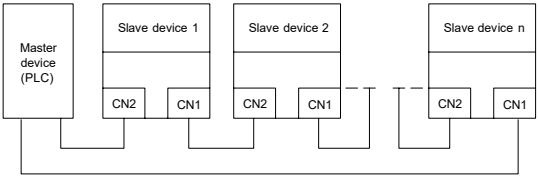


Figure 4-3 Star network topology electrical connection



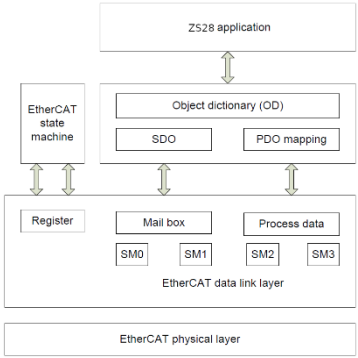
Note: For the star network topology, you need to prepare EtherCAT switches. Figure 4-4 Ring network topology electrical connection



4.4 Communication

4.4.1 CANopen over EtherCAT reference model

Figure 4-5 CoE reference model



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

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

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

Table 4-3 EtherCAT Sync Manager configuration

Sync Manager	Configuration	Size	Start address
Sync Manager 0	Assigned to receive SDO	512 bytes	0x1000
Sync Manager 1	Assigned to send SDO	512 bytes	0x1400
Sync Manager 2	Assigned to receive PDO	128 bytes	0x1800
Sync Manager 3	Assigned to send PDO	128 bytes	0x1C00

4.4.2 EtherCAT slave site information

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

4.4.3 EtherCAT state machine

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

Figure 4-6 EtherCAT state machine flowchart

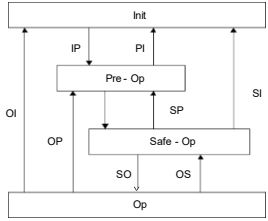


Table 4-4 EtherCAT state machine description

State	Description
Init	SDO communication is unavailable; PDO communication is unavailable.

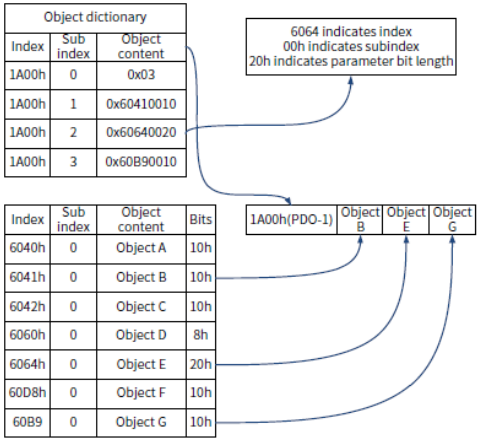
State	Description
Init to Pre-Op	The master configures the data link layer address and synchronization management (SM) channel for SDO communication. The master initializes the distributed clock (DC) synchronization information. The master requests a transition to the Pre-Op state. The master configures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

4.4.4 PDO process data mapping

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

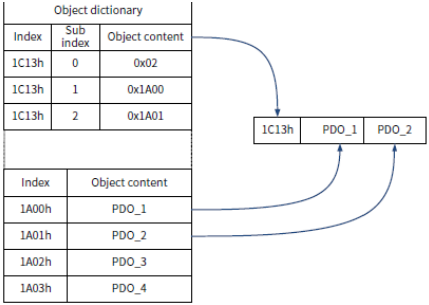
The master can select the required objects for PDO mapping from the object dictionary. PDO mapping configuration is located in the object dictionary within the 1600h to 1603h range for RxPDOs (Receive PDOs) and the 1A00h to 1A03h range for TxPDOs (Transmit PDOs).Figure 4-7shows the PDO mapping method.

Figure 4-7 PDO mapping method



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

Figure 4-8 Mapping between PDOs and Sync Managers



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

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

4.4.5 Network synchronization based on distributed clocks

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

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

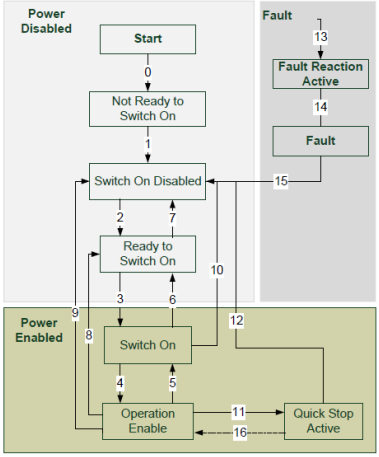
DC mode: The servo drive achieves synchronization operations through the Sync0 events sent by the master.

4.5 CiA402 device profile

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

4.5.1 CANopen over EtherCAT state machine

Figure 4-9 CANopen over EtherCAT state machine



State	Description
Not Ready to Switch On	The drive is in the initialization process.
Switch On Disabled	Drive initialization is completed.
Ready to Switch On	The drive is waiting to enter the Switch On state, and the motor is not excited.
Switched On	The drive is ready, and the main circuit power is normal.

State	Description
Operation Enable	The drive is enabled and it controls the motor in accordance with the control mode.
Quick Stop Active	The drive stops according to the set mode.
Fault Reaction Active	The drive detects that an alarm has occurred, and stops according to the set mode, while the motor still has excitation signal.
Fault	The drive is in a fault state and the motor has no excitation signal.

The control word 6040h includes the following content:

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

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

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

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

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

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

Bit	Operation mode		
	Profile position mode	Profile velocity mode	Homing mode
4	New set-point	Reserved	Homing operation start
5	Change set immediately	Reserved	Reserved
6	Rel	Reserved	Reserved
8	Halt	Halt	Halt

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

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

The status word 6041h includes the following content:

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

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

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

Bits 0–3, 5, and 6:

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

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

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

Bit 10: Target reached.

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

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

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

Other modes: 0

Bit 11: External limitations

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

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

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

Other modes: 0

Bit 12: Manufacturer defined 1

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

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

Other modes: 0

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

Bit 7–bit 8, and bit 15: Reserved.

4.5.2 Device run mode

Set the VFD parameters P00.01=2 (Running command channel), P00.02=3 (EtherCAT communication channel), and P24.27 (Communication timeout time).

4.5.2.1 VFD Mode

1. Set 【6060h: Mode of operations】 to 2 (VFD mode).

2. Set 【6046h: vl velocity min max amount】 to set the max. and min. rotation speeds. If you do not set it, the default values on the drive are used.

3. Set Object Dictionary 【6048h: vl velocity acceleration】, 【6049: vl velocity deceleration】, and【604Ah:02 Quick Stop Speed】. If you do not set them, the default values on the drive are used.

   - The acceleration time is 60 * 【6048h:02 Acceleration Delta Time】* P00.04/(【6048h:01 Acceleration Delta Speed】* Number of motor pole pairs)

     * 0.1, with second as the unit, corresponding to P00.11. The value will remain the same as the previous one when it exceeds 3600.0 seconds.

   - The deceleration time is 60* 【6049h:02 Deceleration Delta Time】

P00.04/(【6049h:01 Deceleration Delta Speed】 Number of motor pole pairs)*0.1, with second as the unit, corresponding to P00.12. The value will remain the same as the previous one when it exceeds 3600.0 seconds.

- The emergency stop time is 60*【604Ah:02 Quick Stop Delta Speed】

  *P00.04/(【604Ah:01 Quick Stop Delta Time】 * Number of motor pole pairs)*0.1, with second as the unit, corresponding to P01.26. The value will remain the same as the previous one when it exceeds 60.0 seconds.

4. Set 【604Ch: vl dimension factor】 to adjust the electronic gear ratio, which is 1:1 by default.

5. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.

6. Set 【6042h: vl target velocity】 to set the target speed.

7. Set 【6040h: Control word】 to run the drive (when it is set to 0x7F, the drive runs).

8. Query 【6044h: vl velocity actual value】to obtain the motor actual speed feedback.

4.5.2.2 Profile Velocity Mode

1. Set 【6060h: Mode of operations】 to 3 (Profile Velocity Mode).

2. Set【6083h: Profile acceleration】 and 【6084h: Profile deceleration】 and write the corresponding values to P00.11 and P00.12.

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

3. Set P00.01=2, P00.02=3, and P00.06=14.

4. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.

5. Set 【60FFh: Target velocity】 to set the target rotational speed (unit: rpm).

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

4.5.2.3 Profile Torque Mode

1. Set 【6060h: Mode of operations】 to 4 (Profile Torque Mode).

2. Set 【6087h】 to set the ramp torque.

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

4. Set P03.11=14 and P03.32=1 (torque control enabled).

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

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

4.5.2.4 Cyclic Synchronous Velocity Mode

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

2. Set 【6083h: Profile acceleration】 and 【6084h: Profile deceleration】.

3. Set P00.01=2, P00.02=3, and P00.06=14.

4. Set 【6040h: Control word】 to enable the drive (set to 0x0F to enable) and start the motor operation.

5. Set 【60FFh: Target velocity】 to set the target rotational speed (unit: rpm).

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

Note: Cyclic Synchronous Velocity Mode of ZS28 does not have synchronization properties.

4.5.2.5 Cyclic Synchronous Torque Mode

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

2. Set P03.11=14 (Communication) and P03.32=1 (Torque control enabled).

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

4. Set 【6072h: Max torque】 to the maximum torque and 【6071h: Target torque 】 to the target torque.

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

Note: Cyclic Synchronous Torque Mode of ZS28 does not have synchronization properties.

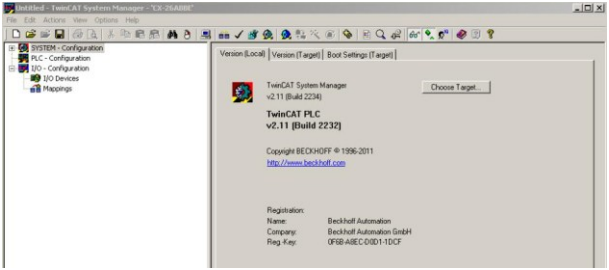
4.6 PLC communication example 1 (TwinCAT2)

The following example shows the configuration for using an EtherCAT adapter module to communicate with Beckhoff TwinCAT2 that serves as the master.

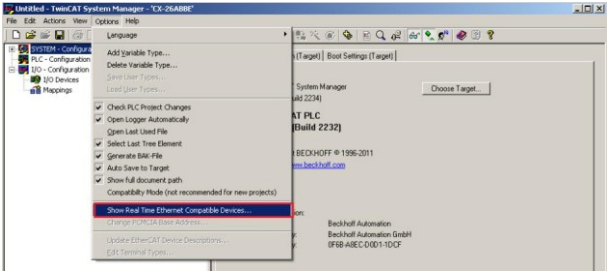
Step 1 Install the TwinCAT2 software.

Step 2 Copy ZS28 configuration file (**ZSY-TX149_1.0.0.0.xml**) to the TwinCAT2 installation directory **C:\TwinCAT\Io\EtherCAT**.

Step 3 Open TwinCAT2.



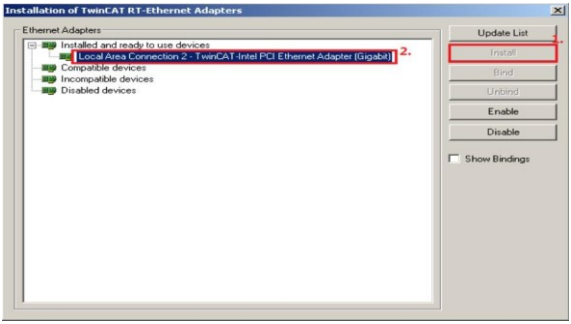
Step 4 Install the network card driver.



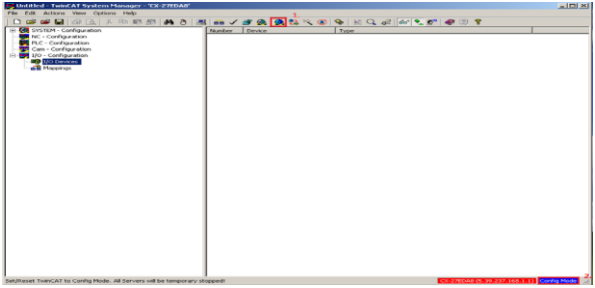
In the top menu bar, choose **Options** > **Show Realtime Ethernet Compatible Devices...** In the pop-up dialog box, choose the local network card, click **Install**.

Once the network card is installed, it is displayed under **Installed and ready to use devices**.

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

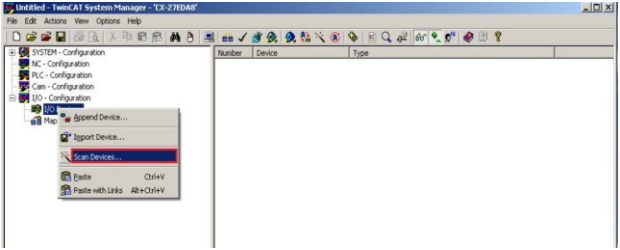


Step 5 Set TwinCAT2 to configuration mode.

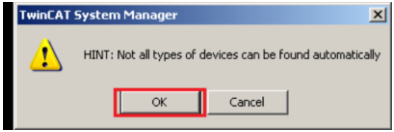


Step 6 Scan for devices.

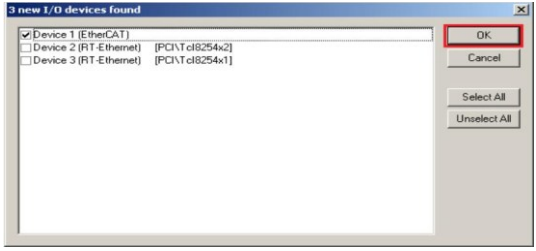
A. Right click **I/O Devices** and choose **Scan Devices...** to scan for devices.



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



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



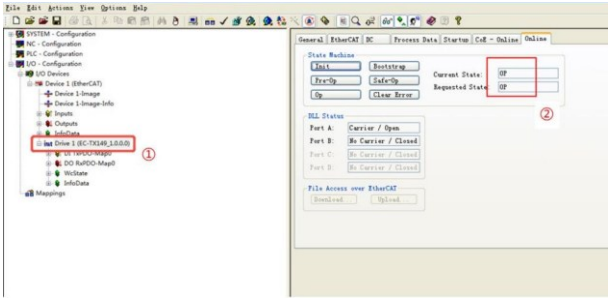
D. Click **Yes** in the pop-up dialog box.



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

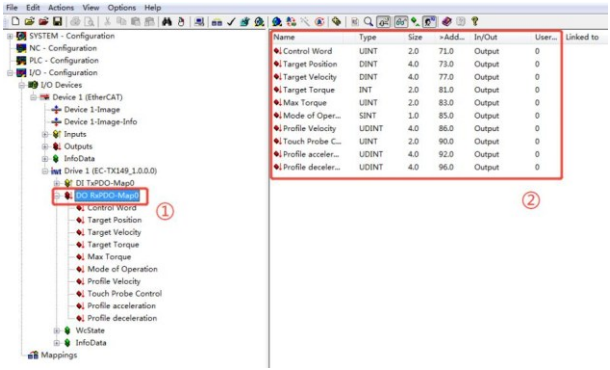


F. As shown in the following figure, **Drive 1 (ZSY-TX149_1.0.0.0)** is the scanned slave device. Check whether the device enters the OP state.

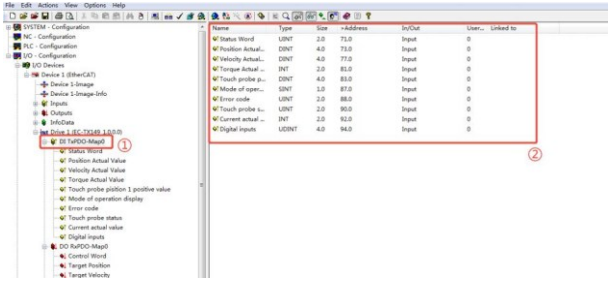


Step 7 Perform process data input and output.

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

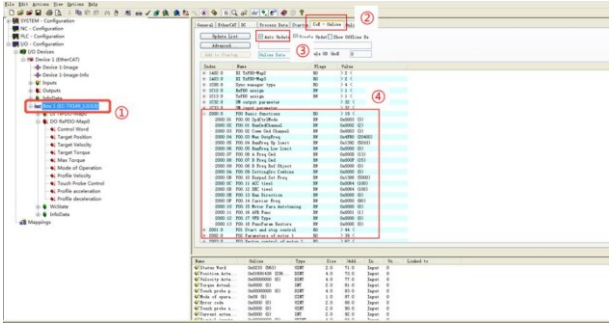


B. Choose **DI TxPDO-Map0**. The data is sent from the VFD to the master, including the VFD status and speed.



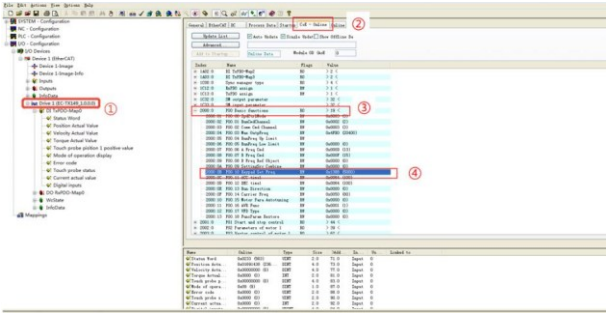
Step 8 Perform SDO data read and write operations.

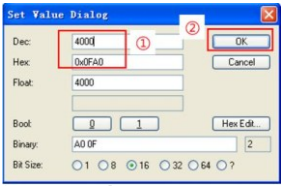
A. Click **CoE-Online**. Select 0x2000–0x2063, and select **Auto Update**. Then you can view the parameters of the corresponding function codes.



B. Similarly, you can write function parameters through 0x2000–0x2063. Select

0x2000, and double-click **2000:0B**. A dialog box will pop up. Write the parameters, click **OK**, and check the keypad. Then you can find that parameter writing is success.



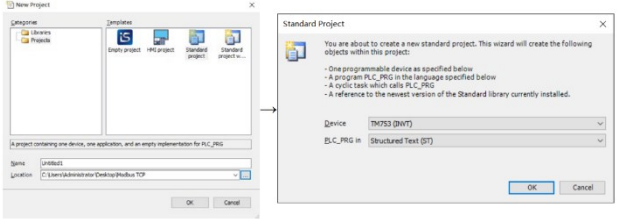


4.7 PLC communication example 2 (TM753)

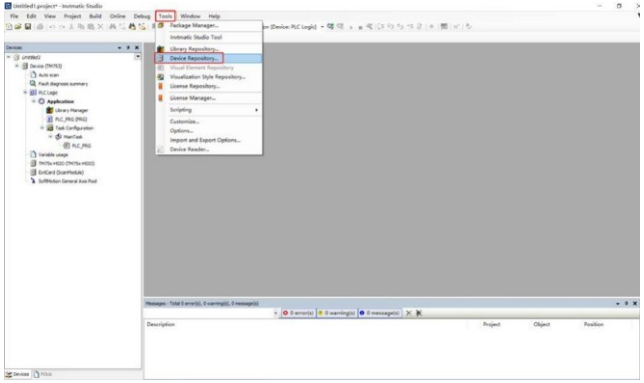
The following example shows the configuration for using an EtherCAT adapter module to communicate with TM753 that serves as the master.

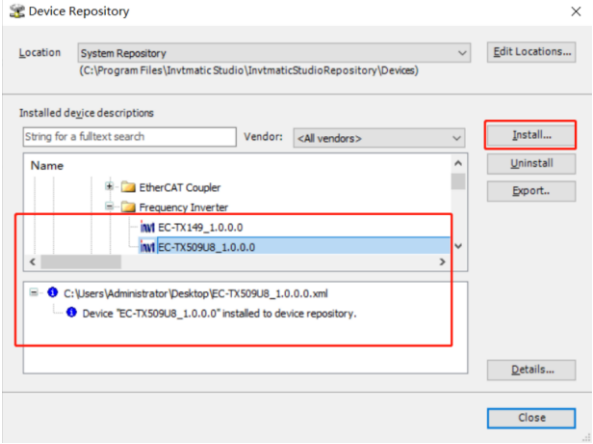
Step 1 Open the Z:Systemsmatic Studio software and create a new project TM753.

Select device **TM753**. The interface is shown in the following figure.

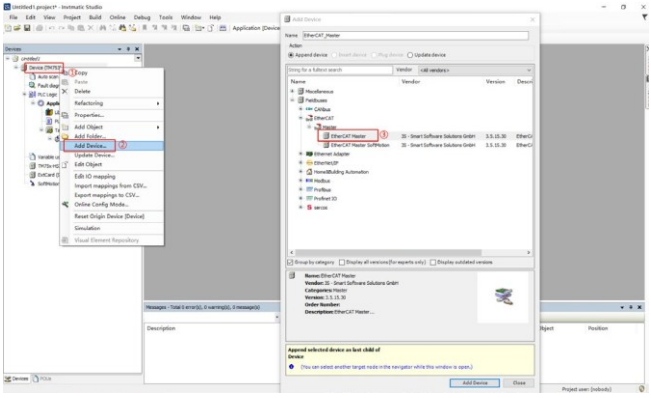


Step 2 Add an EtherCAT device. In the top menu bar, choose **Tools** > **Device Repository**, and import the EtherCAT configuration file of ZSY-TX149.

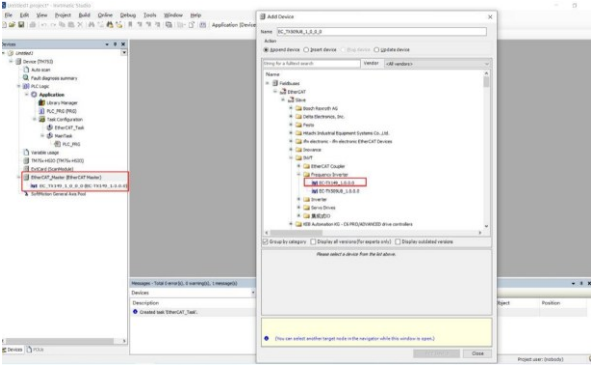




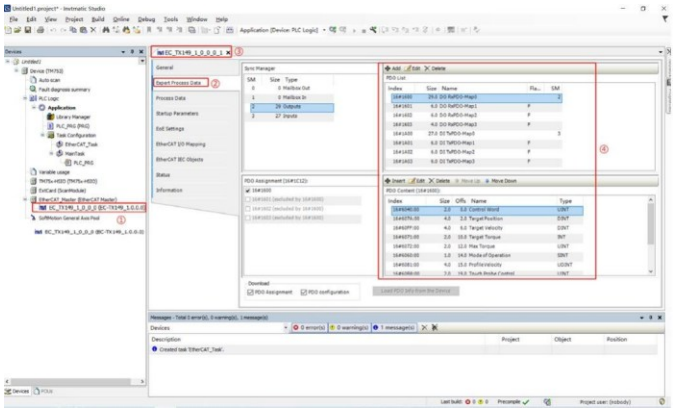
Step 3 Right click **Device**, and choose **Add Device**. In the pop-up window, choose **Fieldbuses** > **EtherCAT** > **Master**, and double click **EtherCAT Master**.



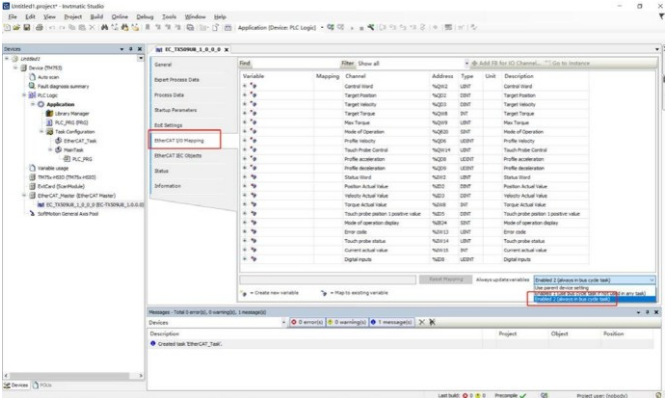
Step 4 Right click **EtherCAT Master**, and choose **Add Device**. In the pop-up window, choose **Fieldbuses** > **EtherCAT** > **Slave** > **Z:SYSTEMS** > **Frequency Inventory** > **ZSY-TX149_1.0.0.0**.



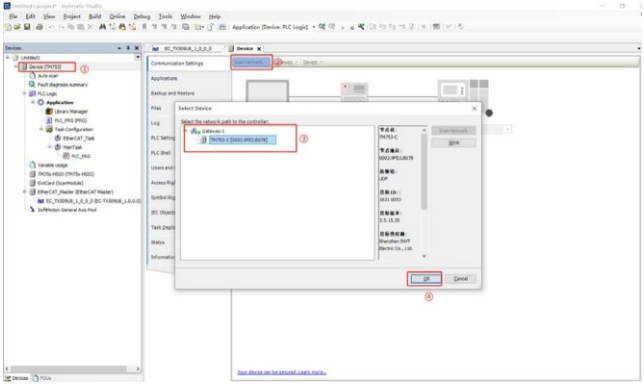
Step 5 Perform PDO configuration. Double-click **ZSY-TX149_1.0.0.0** that was added to enter the parameter configuration interface. Enter the **Expert Process Data** interface to configure the PDO parameters. PDO parameter configuration can be done according to requirements. This example uses the default PDO configuration.



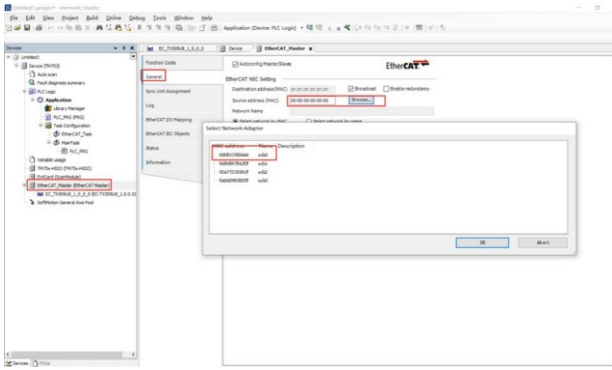
Step 6 Choose **EtherCAT I/O mapping**, and change **Always update variables** to **Enabled 2 (always in bus cycle task)**.



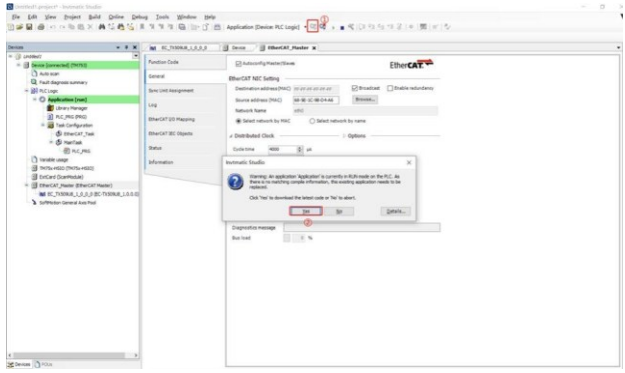
Step 7 Scan for the PLC. Double-click **Device**, click **Scan Network...**, select the scanned PLC, and click **OK**.



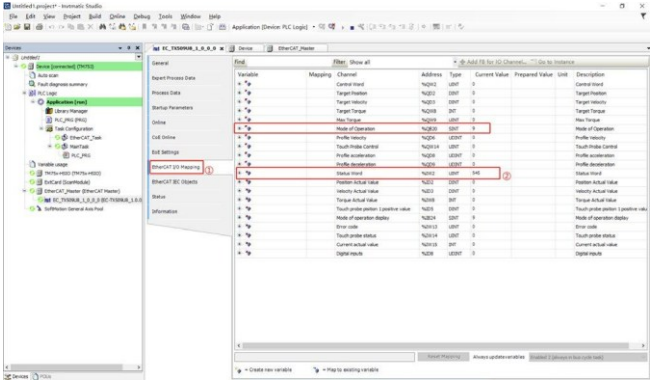
Step 8 Perform EtherCAT master configuration. Double-click **EtherCAT_Master**, choose **General**, click **Browse**, and double-click to select **eth0**.



Step 9 Download the program.

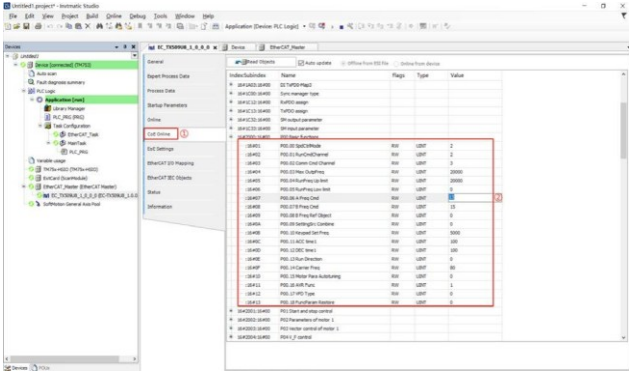


Step 10 Perform parameter monitoring. TPDO data can be monitored in real time and RPDO data can be written by choosing **EtherCAT I/O mapping**.



Step 11 In the **COE online** interface, you can monitor the values of the function codes, and you can also directly modify the values of the function codes.

Note: P00.00 modification is invalid.



5 Modbus TCP protocol

5.1 Overview

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

5.2 Product features

5.2.1 Supported functions

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

5.2.2 Supported communication types

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

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

5.2.3 Status indicator

The Modbus TCP communication card provides four indicators to indicate its states. For details, seeTable 5-1.

Table 5-1 Indicator description

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

Indicator	Color	Definition	Function
LED4 (POWER)	Red	Steady on	3.3V power indicator
LED5(ERR)	Red	Steady on	The communication between the communication card and PLC is offline.
		Blinking (On: 500ms; Off: 500ms)	An attempt to operate an unsupported CMD control word instruction or PR function code value.
		Blinking (On: 62.5ms; Off: 62.5ms)	An attempt to operate on a non-existent node address.
		Steady off	The communication between the communication card and PLC is normal.
LED6 (SYS)	Green	Blinking (On: 500ms; Off: 500ms)	Communication card heartbeat indicator (communication card is running normally).

5.3 Electrical connection

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

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

Figure 5-1 Linear network topology electrical connection

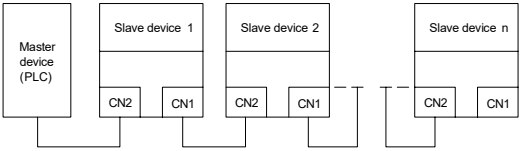
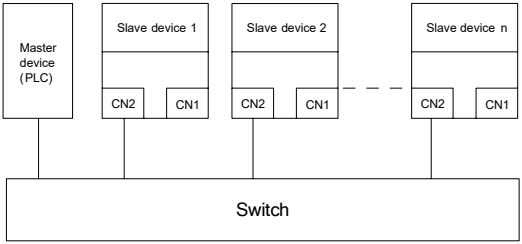


Figure 5-2 Star network topology electrical connection



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

5.4 Communication

5.4.1 Communication settings

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

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

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

Step 2 Set the control method.

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

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

5.4.2 Message format

The TCP communication message is shown inTable 5-2.

Table 5-2 TCP communication message

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

5.4.3 Modbus TCP communication

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

Table 5-3 Modbus TCP protocol message

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

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

Parameter description:

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

Table 5-4 Modbus function code

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

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

Message example:

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

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

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

For example, if the master reads two contiguous pieces of data (that is, to read content from the data addresses 0004H and 0005H) from 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	Read address	Data byte
	Response	0001	0000	0007	01	03	04	1388 0000
	Meaning	MBAP				Function code	Data byte	Data

The content shows that the data in 0004H is 1388H (50.00Hz), and that in 0005H is 0000H (00.00Hz).

2. Command code 06H: writing a word

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

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

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

3. 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 structure is as follows.

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

The address of a function code consists of two bytes, with the MSB on the left and LSB on the right. The MSB ranges from 00 to ffH, and the LSB also ranges from 00 to ffH. The MSB is the hexadecimal form of the group number before the dot mark, and LSB is that of the number behind the dot mark. Take P14.00 as an example.

The group number is 14, that is, the MSB of the parameter address is 0E in the hexadecimal form; and the number behind the dot mark is 00, that is, the LSB is 00 in the hexadecimal form. Therefore, the function code address is 0E00H in the hexadecimal form. For P14.03, the parameter address is 0E03H.

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

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

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

Address description of other Modbus functions

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

Function description	Address	Data description	R/W
Communication-based control command	2000H	0001H: Run forward	R/W
		0002H: Run reversely	
		0003H: Jog forward	
		0004H: Jog reversely	
		0005H: Stop	
		0006H: Coast to stop	
		0007H: Fault reset	
		0008H: Stop jogging	
		0009H: Emergency stop	
Communication-based setting address	2001H	Communication-based frequency setting (0–Fmax, unit: 0.01Hz)	R/W
	2002H	PID reference (0–1000, in which 1000 corresponds to 100.0%)	R/W
	2003H	PID feedback (0–1000, in which 1000 corresponds to 100.0%)	R/W
	2004H	Torque setting (-3000–3000, in which 1000 corresponds to 100.0% of the motor rated current)	R/W
	2005H	Upper limit setting of forward running frequency (0–Fmax; unit: 0.01Hz)	R/W
	2006H	Upper limit setting of reverse running frequency (0–Fmax; unit: 0.01Hz)	R/W
	2007H	Electromotive torque upper limit (0–3000, in which 1000 corresponds to 100.0% of the motor rated current)	R/W
	2008H	Upper limit of the braking torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current)	R/W
	2009H	Special CW Bit0 = 0: Motor 1 = 1: Motor 2 Bit1: Reserved Bit2: = 1: Enable the switchover between torque control/ speed control = 0: No switchover Bit3: = 1: Clear electricity consumption data = 0: Keep electricity consumption	R/W

Function description	Address	Data description	R/W
		Bit4: = 1: Enable pre-excitation = 0: Disable pre-excitation Bit5: = 1: Enable DC braking = 0: Disable DC braking	
	200AH	Virtual input terminal command (0x000–0x7FF)	R/W
	200BH	Virtual output terminal command (0x000–0x01F)	R/W
	200CH	Voltage setting (used when V/F separation is implemented) (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage)	R/W
	200DH	AO setting 1 (-1000–+1000, in which 1000 corresponding to 100.0%)	R/W
	200EH	AO setting 2 (-1000–+1000, in which 1000 corresponding to 100.0%)	R/W
VFD status word 1	2100H	0001H: Forward running	R
		0002H: Reverse running	
		0003H: Stopped	
		0004H: VFD in fault	
		0005H: VFD in POFF state	
		0006H: Pre-exciting	
VFD status word 2	2101H	Bit0: = 0: Not ready to run = 1: Ready to run Bit2–Bit1: =00: Motor 1 =01: Motor 2 = 10–11: Reserved Bit3: = 0: AM = 1: SM Bit4: = 0: No pre-alarm upon overload =1: Overload pre-alarm Bit6–Bit5: = 00: Keypad-based control = 01: Terminal-based control = 10: Communication-based control Bit7: Reserved Bit8: =0: Speed control =1: Torque control Bit 9: Reserved Bit11–bit10: = 00: Vector 0 = 01: Vector 1 = 10: Closed-loop vector =11: Reserved	R

Function description	Address	Data description		R/W
VFD fault code	2102H	See the description of fault types.		R
VFD identification code	2103H	ZS28-----0x1202		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	0.0–300.0A (Unit: 0.1A)		R
Rotational 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
PID setting	3008H	-100.0–100.0% (Unit: 0.1%)		R
PID feedback	3009H	-100.0–100.0% (Unit: 0.1%)		R
Input IO state	300AH	0x000–0x7FF Corresponding to the local terminals: HDI1/Reserved/Reserved/DI8/DI7/DI6/DI5/DI4/DI3/DI2/DI1		R
Output IO state	300BH	0x00–0x1F Corresponding to the local terminals RO1/HDO1/Reserved/Reserved/Reserved		R
Analog input 1	300CH	0.00–10.00V (Unit: 0.01V)		R

Function description	Address	Data description	R/W
Analog input 2	300DH	0.00–10.00V (Unit: 0.01V)	R
Analog input 3	300EH	-10.00–10.00V (Unit: 0.01V)	R
Reserved	300FH	Reserved	R
HDI1 high-speed pulse input	3010H	0.00–50.00kHz (Unit: 0.01Hz)	R
Reserved	3011H	Reserved	R
Present step of simple PLC	3012H	0–15	R
External length value	3013H	0–65535	R
External counting value	3014H	0–65535	R
Torque setting	3015H	-300.0–300.0% (Unit: 0.1%)	R
VFD identification code	3016H		R
Fault code	5000H		R

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

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

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

5.4.5 Fieldbus scale

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

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

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

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

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

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

After receiving the command, the VFD converts 50 into 5.0 based on the fieldbus scale, and then sets "Wake-up-from-sleep delay" to 5.0s.

5.4.6 Error response

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

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

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

Code	Name	Meaning
08H	Parameter cannot be modified in running	The parameter to be modified in the write operation of the host controller cannot be modified during the running of the VFD.
09H	Password protection	If the host controller does not provide the correct password to unlock the system to perform a read or write operation, the error of "system being locked" is reported.

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

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

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

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

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

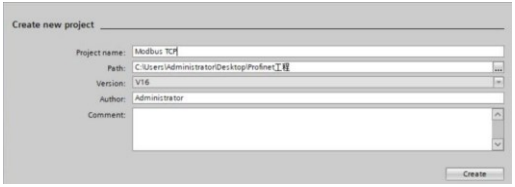
In addition to the modification of the code, the slave returns a byte of exception code that describes the cause of the exception. After receiving the exception response, the typical processing of the master is to send the request message againor modify the command based on the fault information.

5.5 PLC communication example 1 (S7-1200)

The following example shows the configuration for using a Modbus TCP adapter module to communicate with a Siemens PLC (using TIA Portal V15 as the configuration tool) without a Modbus TCP device description file.

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

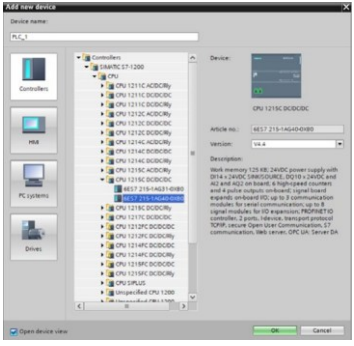
Step 2 Open TIA Portal V15, and create a new project as shown in the following figure.



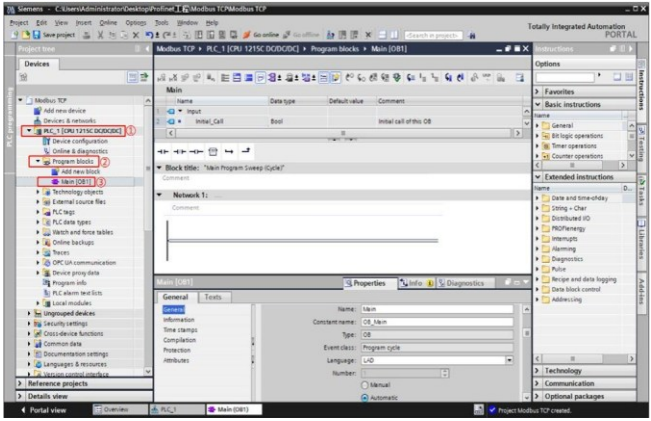
Step 3 After creating, click the **Project View** in the bottom left corner. Double-click **Add new device** in the opened interface. See the following figure.



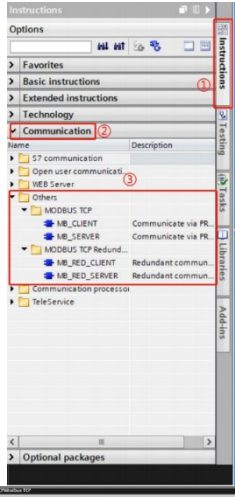
Step 4 Select the corresponding PLC model. The PLC model used in this example is shown in the following figure. Click **OK**.

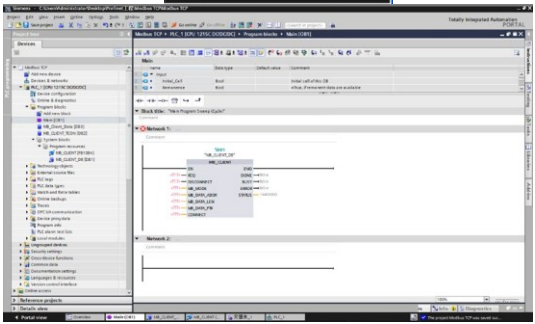


Step 5 On the left, choose **Program Block**, and double-click **Main[OB1]** to open the programming interface, as shown in the following figure.

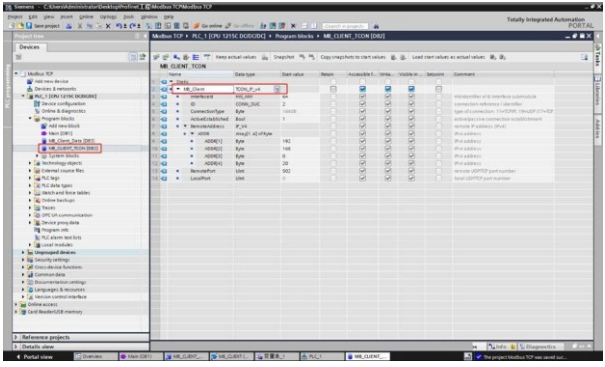


Step 6 On the right, choose **Instructions** > **Communication** > **Others** > **MODBUS TCP**, double click or drag **MB_CLIENT** to add it to **Main[OB1]**, as shown in the following figure.

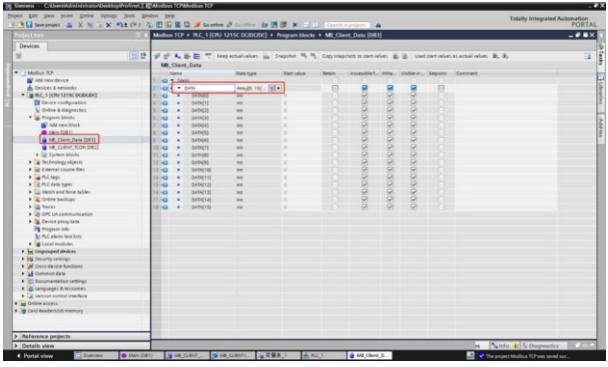




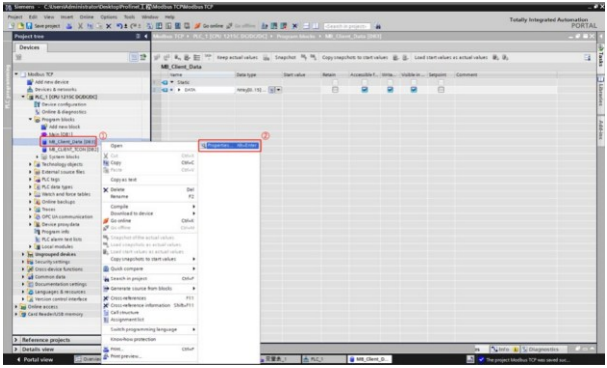
Step 7 On the left, choose **Program blocks** > **Add new block**, a global data block named **MB_CLIENT_TCON** for mapping the pin CONNECT of the function block MB_CLIENT. Open the data block, create a variable **MB_Client**, manually enter **TCON_IP_v4** in the data type box, and configure the parameters of the data block.

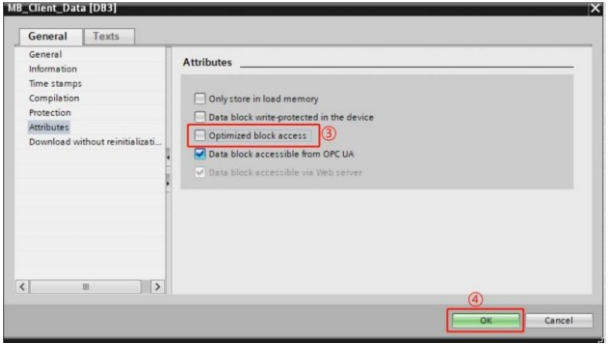


Step 8 Add another global data block for data storage, named **MB_Client_Data**. Open this data block and create variable **DATA**.

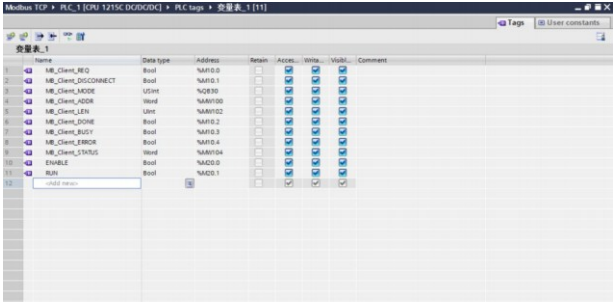


Step 9 Change the attributes of data blocks **MB_CLIENT_TCON** and **MB_Client_Data**, and deselect **Optimized block access**.

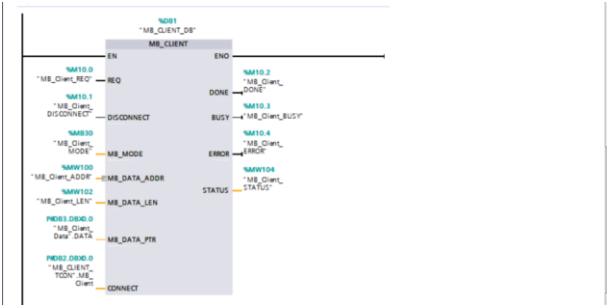




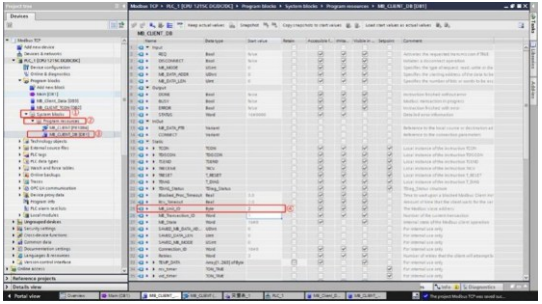
Step 10 In PLC variables, create a new variable table and define the following variables.



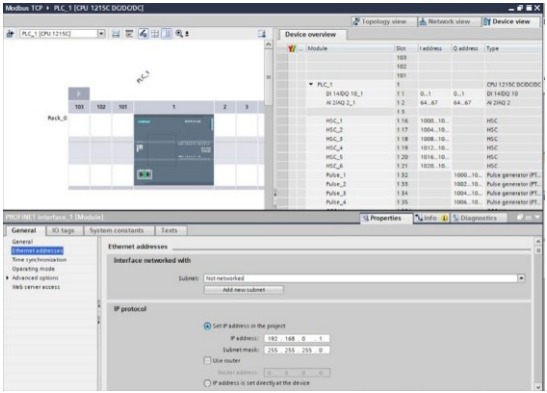
Step 11 Connect the variables to the corresponding pins of the **MB_CLIENT** block.



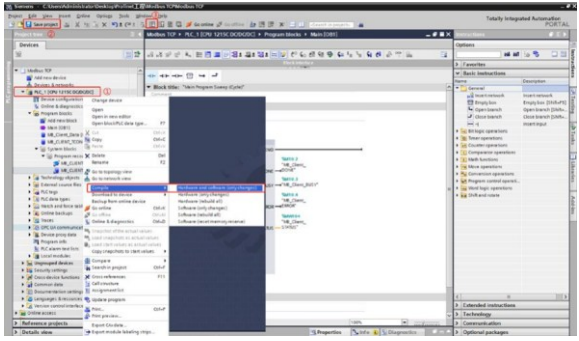
Step 12 Change the ID in the MB_CLIENT background data block to match the connection ID. This can be modified by choosing **Program blocks** > **System blocks** > **Program resources** > **MB_Client_DB** on the left. Modify the ID to be consistent with the ID in **MB_CLIENT_TCON**.



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

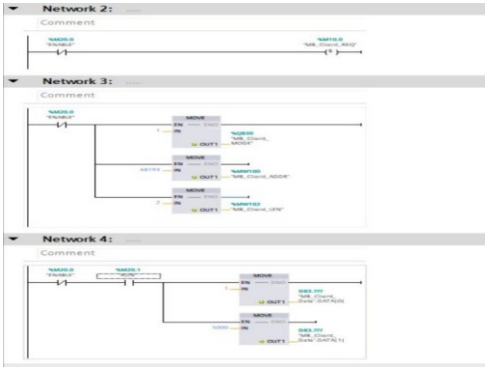


Step 14 At this point, a Modbus TCP connection has been established. Right click **PLC_1 [CPU 1215C DC/DC/DC]**, and choose **Compile** > **Hardware and software (change only)** to compile the entire project. Click **Save Project** to save the entire project, and then click the **Download to Device** icon to download the project configuration to the PLC. See the following figure.



Step 15 You can now write and download the following PLC program. Additionally, set P00.01=2, P00.02=0, P00.06=10, P14.00=2 (consistent with the ID in **MB_CLIENT_TCON**), P24.37–P24.40=192.168.0.20 (consistent with the IP in **MB_CLIENT_TCON**), and keep P24.41–P24.44 at their default values. This will allow control of the VFD to run at 50.00Hz

using the Modbus TCP protocol.

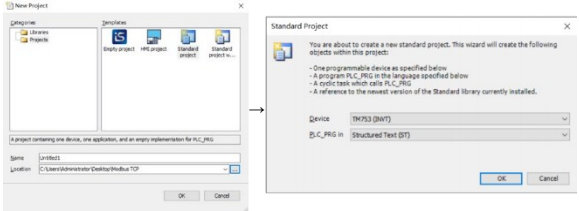


5.6 PLC communication example 2 (TM753)

The following example shows the configuration for using a Modbus TCP adapter module to communicate with TM753 that serves as the master.

Step 1 Open the Z:Systemsmatic Studio software and create a new project TM753.

Select device **TM753**. The interface is shown in the following figure.

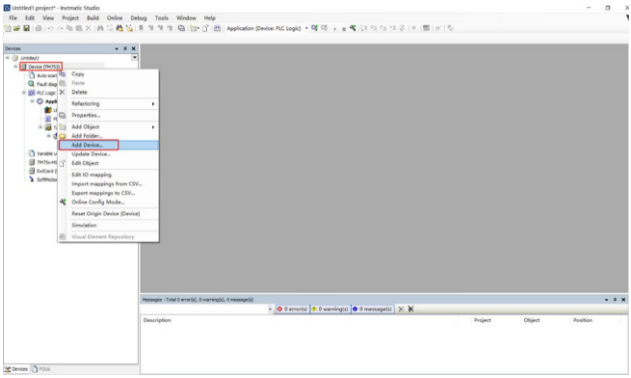


Step 2 Add Modbus TCP related configuration.

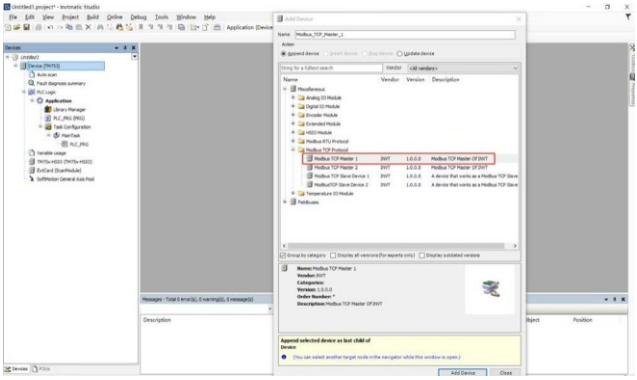
According to the actual cable connection to the Ethernet port of TM753, select **Modbus TCP Master1** or **Modbus TCP Master2**; select **Modbus**

TCP Master1 since the Ethernet 1 port is actually connected in this example. After adding the master, select the corresponding **Modbus TCP Slave1**.

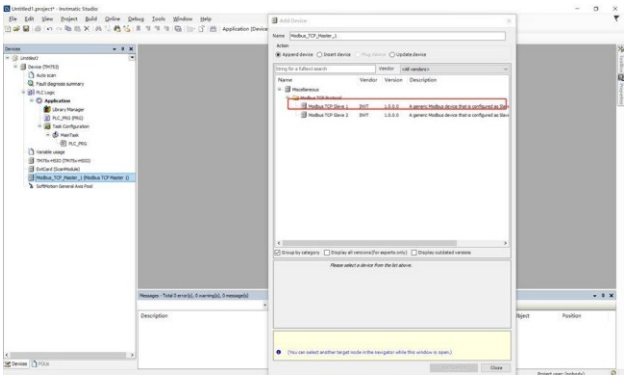
Add devices.



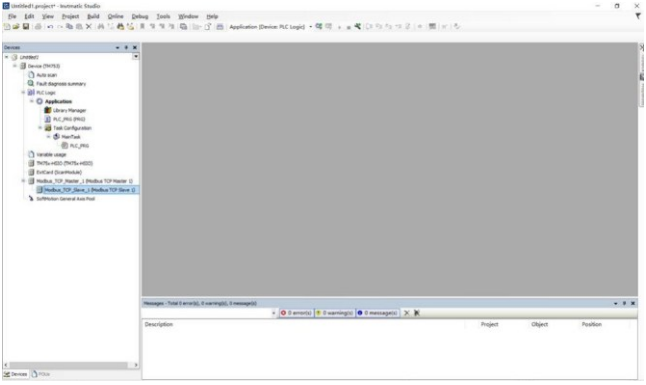
(1) Add the Modbus TCP master.



(2) Add the Modbus TCP slave.



The adding is completed.

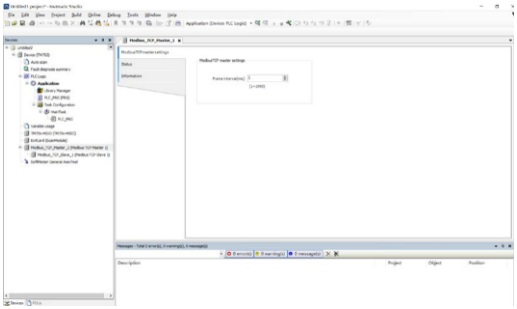


The Modbus TCP configuration is completed. If there are multiple Modbus TCP slaves, add multiple **Modbus TCP Slave**.

Step 3 Perform Modbus TCP master station parameter setting.

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

Perform Modbus TCP master station configuration.



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

Step 4 Perform Modbus TCP slave parameter setting.

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

(1) Perform Modbus TCP slave station configuration.

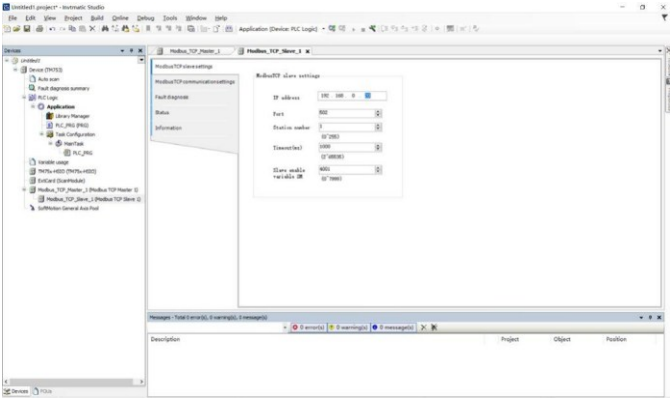


Table 5-5 Configuration parameters

Configuration item	Setting
IP address	IP address of the Modbus TCP slave station to which the master station connects.
Port	TCP port number of the Modbus TCP slave station to which the master station connects.
Station number	Protocol station address of the Modbus TCP slave station to which the master station connects.
Timeout time (ms)	If the slave station does not respond within this duration
Slave station enabling variable	IP address of the Modbus TCP slave station to which the master station connects.

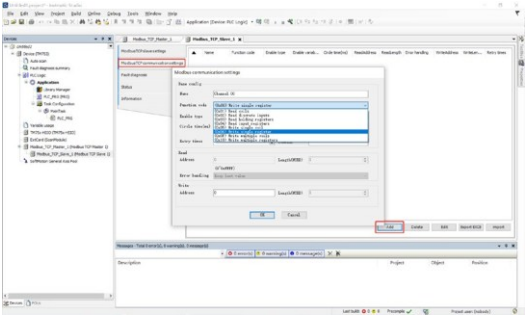
Example

Configuration item	Setting
IP address	192.168.0.20
Port	502
Station number	1

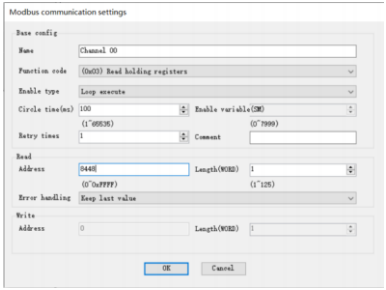
Timeout time (ms)	1000
Slave station enabling variable	4001

(2) Configure Modbus TCP commands.

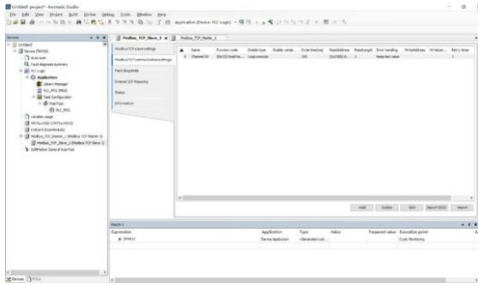
Click **Modbus TCP communication settings** to configure Modbus TCP commands. After clicking **Add**, a dialog box will appear for adding a new channel for the Modbus TCP slave station. Select the function code, set the operation address and length, and click **OK** button to create a new channel. Each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



(3) Perform Modbus TCP slave communication setting.



Modbus TCP slave communication setting is completed.

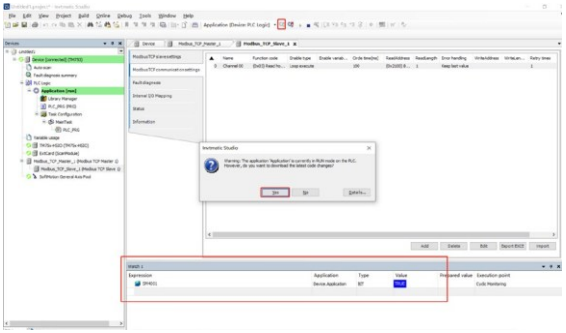


As shown in the preceding figure, a Modbus TCP request is defined to cyclically read the holding register (function code 03) with a 100ms period. The read address is 0x2100 (VFD status word).

Step 5 Compile and download.

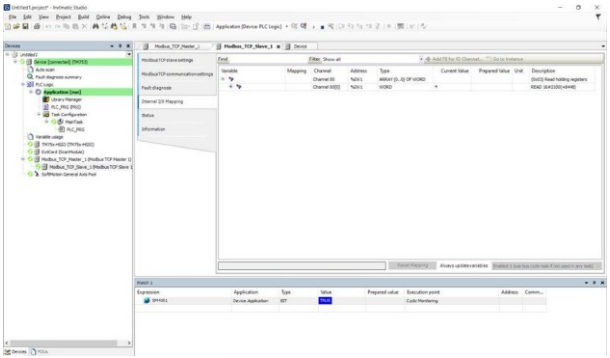
So far, the communication configuration is completed. Then compile and download it to the PLC. Click **Log in** to download, and after the download is complete, run the PLC. Set the VFD IP address, P24.37–P24.40 to **192.168.0.20**, set the station address P14.00 to **1**, and keep default settings for P24.41–P24.44. Keep consistent with Modbus TCP slave settings, and set SM4001 enabling to **True** to establish successful Modbus TCP communication.

(1) Compilation download.



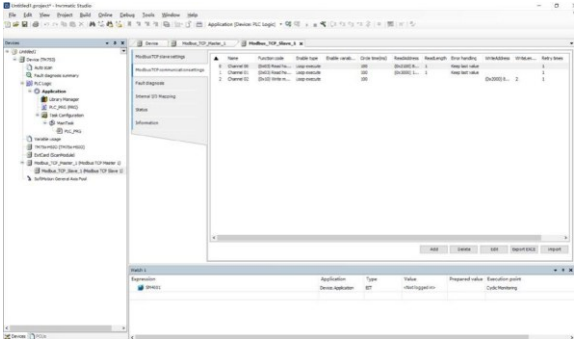
After adding the master/slave communication configuration in the Modbus TCP slave communication settings, the values that are read can be viewed in the Internal I/O mapping interface. The internal I/O mapping will automatically allocate a mapping address for each configured mapping, such as %IW1 in the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

(2) Internal I/O mapping parameter view.



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

(3) Configuration of multiple Modbus TCP commands.

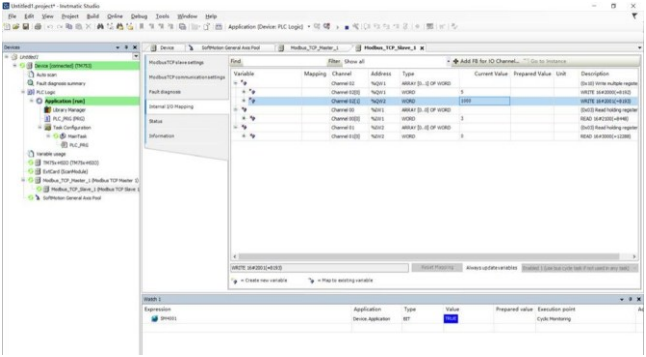


After adding, download to PLC and run it. Enable SM4001, then check the Internal I/O mapping interface.

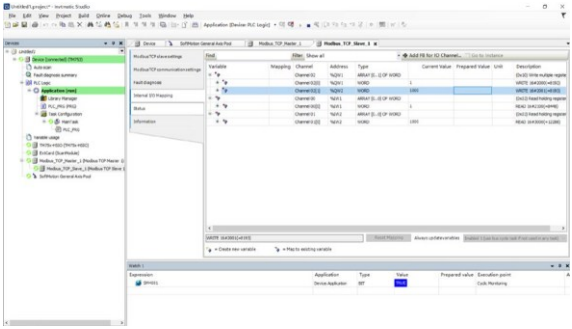
Set VFD function codes, P00.01=2, P00.02=0, and P00.06=10.

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

(4) Parameter reference.



(5) Parameter view.



Appendix A EtherCAT object dictionary

Index	Subindex	Description	Access permission	Data type	Default
1000h	0	Device type	RO	UINT32	0x00000192
1001h	0	Error register	RO	UINT8	0
1008h	0	Manufacturer device name	RO	String	ZSY-TX149
1009h	0	Manufacturer hardware version	RO	String	Hardware version depended
100Ah	0	Manufacturer software version	RO	String	Software version depended
1018h	**ID object**				
	0	Included max. subindexes	RO	UINT8	4
	1	Supplier ID	RO	UINT32	0x0000072C
	2	Product code	RO	UINT32	0x00009000
	3	Revision No.	RO	UINT32	0x00000001
	4	Sequence No.	RO	UINT32	0x00000001
1600h	**RX PDO1 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapped object	RW	UINT32	0x60400010
	2	Second mapped object	RW	UINT32	0x607A0020
	3	Third mapped object	RW	UINT32	0x60FF0020
	4	Fourth mapped object	RW	UINT32	0x60710010
	5	Fifth mapping object	RW	UINT32	0x60720010
	6	Sixth mapping object	RW	UINT32	0x60600008
	7	Seventh mapping object	RW	UINT32	0x60810020
	8	Eighth mapping object	RW	UINT32	0x60B80010
1601h	**RX PDO2 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	2

Index	Subindex	Description	Access permission	Data type	Default
	1	First mapped object	RW	UINT32	0x60400010
	2	Second mapped object	RW	UINT32	0x607A0020
1602h	**RX PDO3 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	2
	1	First mapped object	RW	UINT32	0x60400010
	2	Second mapped object	RW	UINT32	0x607A0020
1603h	**RX PDO4 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	2
	1	First mapped object	RW	UINT32	0x60400010
	2	Second mapped object	RW	UINT32	0x607A0020
1A00h	**TX PDO1 mapping parameters**				
	0	Number of supported mapping objects	RW	UINT8	8
	1	First mapped object	RW	UINT32	0x60410010
	2	Second mapped object	RW	UINT32	0x60640020
	3	Third mapped object	RW	UINT32	0x606C0020
	4	Fourth mapped 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 ojebct	RW	UINT32	0x60BA0020

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

Index	Subindex	Description	Access permission	Data type	Default
	4	SM3 communication type	RO	UINT8	0x04
1C12h	**RxPDO allocating**				
	0	Max. subindexes	RW	UINT8	1
	1	RxPDO allocated object index	RW	UINT16	0x1600
1C13h	**TxPDO allocating**				
	0	Max. subindexes	RW	UINT8	1
	1	RxPDO allocated object index	RW	UINT16	0x1A00
1C32h	**Sync Manager synchronization output parameters**				
	0x00	Max. subindexes	RO	UINT8	0x20
	0x01	Synchronization mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switchover time	RO	UINT32	0
	0x04	Supported synchronization type	RO	UINT16	0x4006
	0x05	Min. cycle time	RO	UINT32	0x0003D090
	0x06	Calculation and replication time	RO	UINT32	0
	0x07	Reserved	RW	UINT32	0
	0x08	Obtain cycle time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0

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

Index	Subindex	Description	Access permission	Data type	Default
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Cyclic timeout counter	RO	UINT32	0
	0x0D	Counter of too short switching	RO	UINT32	0
	0x20	Synchronization error	RO	UINT8	0
2000h	0x00–0x13	Function code	RW	UINT16	-
2001h	0x00–0x23	Function code	RW	UINT16	-
2002h	0x00–0x21	Function code	RW	UINT16	-
2003h	0x00–0x42	Function code	RW	UINT16	-
2004h	0x00–0x3C	Function code	RW	UINT16	-
2005h	0x00–0x35	Function code	RW	UINT16	-
2006h	0x00–0x23	Function code	RW	UINT16	-
2007h	0x00–0x56	Function code	RW	UINT16	-
2008h	0x00–0x84	Function code	RW	UINT16	-
2009h	0x00–0x1D	Function code	RW	UINT16	-
200Ah	0x00–0x20	Function code	RW	UINT16	-
200Bh	0x00–0x40	Function code	RW	UINT16	-
200Ch	0x00–0x21	Function code	RW	UINT16	-
200Dh	0x00–0x14	Function code	RW	UINT16	-

Index	Subindex	Description	Access permission	Data type	Default
200Eh	0x00–0x47	Function code	RW	UINT16	-
200Fh	0x00–0x46	Function code	RW	UINT16	-
2010h	0x00–0x55	Function code	RW	UINT16	-
2011h	0x00–0x40	Function code	RW	UINT16	-
2012h	0x00–0x2D	Function code	RW	UINT16	-
2013h	0x00–0x28	Function code	RW	UINT16	-
2014h	0x00–0x28	Function code	RW	UINT16	-
2015h	0x00–0x22	Function code	RW	UINT16	-
2016h	0x00–0x19	Function code	RW	UINT16	-
2017h	0x00–0x14	Function code	RW	UINT16	-
2018h	0x00–0x28	Function code	RW	UINT16	-
2008h	0x00–0x84	Function code	RW	UINT16	-
2009h	0x00–0x1D	Function code	RW	UINT16	-
200Ah	0x00–0x20	Function code	RW	UINT16	-
200Bh	0x00–0x40	Function code	RW	UINT16	-
200Ch	0x00–0x21	Function code	RW	UINT16	-
200Dh	0x00–0x14	Function code	RW	UINT16	-
200Eh	0x00–0x47	Function code	RW	UINT16	-
200Fh	0x00–0x46	Function code	RW	UINT16	-

Index	Subindex	Description	Access permission	Data type	Default
2010h	0x00–0x55	Function code	RW	UINT16	-
2011h	0x00–0x40	Function code	RW	UINT16	-
2012h	0x00–0x2D	Function code	RW	UINT16	-
2013h	0x00–0x28	Function code	RW	UINT16	-
2014h	0x00–0x28	Function code	RW	UINT16	-
2015h	0x00–0x22	Function code	RW	UINT16	-
2016h	0x00–0x19	Function code	RW	UINT16	-
2017h	0x00–0x14	Function code	RW	UINT16	-
2018h	0x00–0x28	Function code	RW	UINT16	-
2019h	0x00–0x21	Function code	RW	UINT16	-
201Ah	0x00–0x35	Function code	RW	UINT16	-
201Bh	0x00–0x1E	Function code	RW	UINT16	-
201Ch	0x00–0x1E	Function code	RW	UINT16	-
603Fh	0	Error code	RO	UINT16	0
6040h	0	Control word	RW	UINT16	0
6041h	0	Status word	RO	UINT16	0
6043h	0	Output speed	RO	INT16	0
6044h	0	Feedback speed	RO	INT16	0
	Speed range				

Index	Subindex	Description	Access permission	Data type	Default
6046h	1	Min. value	RO	UINT32	0
	2	Max. value	RO	UINT32	0
6048h	**Acceleration**				
	1	Acceleration increment	RO	UINT32	0
	2	Acceleration time increment	RO	UINT16	0
6049h	**Deceleration**				
	1	Deceleration increment	RO	UINT32	0
	2	Deceleration time increment	RO	UINT16	0
604Ah	**Quick stop**				
	1	Fast stop speed increment	RW	UINT32	0
	2	Fast stop time increment	RW	UINT16	0
604Ch	**Speed gear ratio**				
	1	Numerator of speed gear ratio	RW	INT32	1
	2	Denominator of speed gear ratio	RW	INT32	1
6060h	0	Operation mode selection	RW	UINT16	0
6061h	0	Operation mode display	RO	UINT16	0
6062h	0	Position command	RO	DINT32	0

Index	Subindex	Description	Access permission	Data type	Default
6063h	0	Position feedback	RO	DINT32	0
6064h	0	Position feedback	RO	DINT32	0
6065h	0	Position deviation range	RW	UDINT32	0
6066h	0	Too-large position deviation timeout	RW	UINT16	0
6067h	0	Position pulse range	RW	UDINT32	0
606Ch	0	Actual speed	RW	DINT32	0
6071h	0	Target torque	RW	INT16	0
6072h	0	Max. torque	RW	UINT16	0
6077h	0	Actual torque value	RO	INT16	0
6078h	0	Actual current value	RO	INT16	0
6079h	0	Bus voltage	RO	UDINT32	0
607Ah	0	Target position	RW	INT16	0
6081h	0	Speed in industrial regulations	RW	UDINT32	0
6083h	0	ACC in industrial regulations	RW	UDINT32	0
6084h	0	DEC in industrial regulations	RW	UDINT32	0
6087h	0	Torque ramp	RW	UDINT32	0
	Gear ratio				

Index	Subindex	Description	Access permission	Data type	Default
6091h	0	Number of subindexes	RW	UINT8	2
	1	Motor resolution	RW	UINT32	0x00000001
	2	Bearing axle resolution	RW	UINT32	0x00000001
6098h	0	Homing method	RW	INT16	0
6099h	**Homing speed**				
	0	Reserved	RW	UINT32	0
60B0h	0	Position offset	RW	INT32	0
60B1h	0	Speed offset	RW	INT32	0
60B2h	0	Torque offset	RW	INT16	0
60B8h	0	Probe control	RW	UINT16	0
60B9h	0	Probe status	RO	UINT16	0
60BAh	0	Probe position rising edge	RO	INT32	0
60BBh	0	Probe position falling edge	RO	INT32	0
60E0h	0	Forward torque limit	RW	UINT16	0
60E1h	0	Reverse torque limit	RW	UINT16	0
60F4h	0	Position deviation	RO	INT32	0
60FDh	0	Digital input	RO	UINT32	0
60FEh	0	Digital output	RO	INT32	0

Index	Subindex	Description	Access permission	Data type	Default
60FFh	0	Target speed	RW	INT32	0
6502h	0	Drive mode	RO	UINT32	0x000003A5

Appendix B Related function code

Function code	Name	Parameter description	Setting range	Default
P00.01	Channel of running commands	0: Keypad 1: Terminal 2: Communication	0–2	0
P00.02	Communication mode of running commands	0: Modbus/Modbus TCP 1: Reserved 2: Ethernet 3: EtherCAT/PROFINET/ EtherNet IP 4–6: Reserved	0–6	0
P00.06	Setting channel of A frequency command	1–9: See the *ZS28 series VFD User Manual*. 10: Modbus/Modbus TCP communication 11: See the *ZS28 series VFD User Manual*. 12: Ethernet communication 13: See the *ZS28 series VFD User Manual*. 14: EtherCAT/PROFINET/EtherNet IP communication 15: See the *ZS28 series VFD User Manual*.	0–15	0
P00.07	Setting channel of B frequency command			1
P03.11	Torque setting method selection of motor 1			0
P03.14	Forward rotation upper-limit frequency source in torque control for motor 1			0
P03.15	Reverse rotation upper-limit frequency source in torque control for motor 1			0
P03.18	Setting source of electromotive torque upper limit for motor 1			0
P03.19	Setting source of braking torque upper limit for motor 1			0

Function code	Name	Parameter description	Setting range	Default
P04.13	Voltage setting channel selection for motor 1			0
P09.00	PID reference source			0
P09.02	PID feedback source			0
P35.11	Torque setting method selection of motor 2			0
P35.14	Forward rotation upper-limit frequency source in torque control for motor 2			0
P35.15	Reverse rotation upper-limit frequency source in torque control for motor 2			0
P35.18	Setting source of electromotive torque upper limit for motor 2			0
P35.19	Setting source of braking torque upper limit for motor 2			0
P36.13	Voltage setting channel selection for motor 2			0
P06.04	HDO1 output selection	0–22: See the *ZS28 series VFD* User Manual.	0-63	0

Function code	Name	Parameter description	Setting range	Default
P06.05	RO1 output selection	23: Modbus/ Modbus TCP communication virtual terminal output 24: See the ZS28 series *VFD User* Manual. 25: Ethernet communication virtual terminal output 26–33: See the *ZS28 series VFD User Manual.* 34: EtherCAT/PROFINET/EtherNet IP communication virtual terminal output 35–63: *See the ZS28 series VFD User Manual.*		1
P06.26	AO1 output selection	0–15: See the *ZS28 series VFD User Manual.* 16: Value 1 set through Modbus/Modbus TCP communication 17: Value 2 set through Modbus/Modbus TCP communication 18–19: See the *ZS28 series VFD User Manual.* 20: Value 1 set through Ethernet communication 21: Value 2 set through Ethernet communication 22: Value 1 set through EtherCAT/PROFINET/EtherNet IP communication 23: Value 2 set through EtherCAT/PROFINET/EtherNet IP communication	0–63	0
P06.28	HDO1 high-speed pulse output			0

Function code	Name	Parameter description	Setting range	Default
P07.27	Present fault type	0–17: See the *ZS28 series VFD User Manual*. 18: Modbus/Modbus TCP communication fault (E18) 19–29: See the *ZS28 series VFD User Manual*. 30: Ethernet communication fault (E30) 31–56: See the *ZS28 series VFD User Manual*. 57: PROFINET communication timeout fault (E57) 58–59: See the *ZS28 series VFD User Manual*. 60: Communication card identifying failure (E60) 61–62: See the *ZS28 series VFD User Manual*. 63: Communication card communication timeout fault (E63) 64–65: See the *ZS28 series VFD User Manual*. 66: EtherCAT communication timeout fault (E66) 67–94: See the *ZS28 series VFD User Manual*. 95: EtherNet IP communication timeout (E95)	0–588	0
P07.28	Last fault type			0
P07.29	2nd-last fault type			0
P07.30	3rd-last fault type			0
P07.31	4th-last fault type			0
P07.32	5th-last fault type			0
P08.31	Motor switchover selection	Ones place: Switching channel selection 0: Terminal 1: Modbus/Modbus TCP communication 2: Reserved 3: Ethernet	0x00–0x14	0x00

Function code	Name	Parameter description	Setting range	Default
		4: EtherCAT/PROFINET/ EtherNetIP communication Tens place: indicates whether to enable switchover during running 0: Disable 1: Enable		
P14.03	Communication response delay	-	0–200ms	5
P14.05	Transmission fault processing	Ones place: 0: Respond to write operations 1: Not respond to write operations Tens place: 0: Communication password protection is invalid. 1: Communication password protection is valid. Hundreds place: 0: User-defined addresses specified in group P16 are invalid. 1: User-defined addresses specified in group P16 are valid. Thousands place: 0: CRC failure, with response of error type 0x06 1: CRC checksum failure, without response	0x0000–0x1111	0x0000
P14.48	Channel selection for mapping between PZDs and function codes	Ones place: Channel for mapping function codes to PZDs 0: Reserved 1: Reserved	0x00–0x12	0x12

Function code	Name	Parameter description	Setting range	Default
		2: Group P23 Tens place: Save function at power off 0: Disable 1: Enable		
P14.49–P14.59	PZD2–PZD12 receive mapping function codes	-	0x0000-0xFFFF	0x0000
P14.60–P14.70	PZD2–PZD2 send mapping function codes	-	0x0000–0xFFFF	0x0000
P14.71	PZD communication control word expression format	0: In decimal format 1: Binary format	0–1	0
P16.00–P16.31	User-defined read addresses 1–16, corresponding to local addresses	-	0x0000–0xFFFF	0xFFFF
P16.32–P16.63	User-defined write addresses 1–16, corresponding to local addresses	-	0x0000–0xFFFF	0xFFFF
P23.02–P23.12	PZD2–PZD12 receive	0: Invalid 1: Set frequency (0–Fmax, unit: 0.01Hz) 2: PID reference (-1000–1000, in which 1000 corresponds to 100.0%) 3: PID feedback (-1000–1000, in which 1000 corresponds to 100.0%) 4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the motor rated current)	0–31	0

Function code	Name	Parameter description	Setting range	Default
		5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01Hz) 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01Hz) 7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 8: Upper limit of braking torque (0–3000, in which 1000 corresponds to 100.0% of the motor rated current) 9: Virtual input terminal command (0x000–0x7FF) 10: Virtual output terminal command (0x000–0x01F) 11: Voltage setting special for V/F separation (0–1000, in which 1000 corresponds to 100.0% of the motor rated voltage) 12: AO output setting 1 (0–1000, in which 1000 corresponds to 100.0%) 13: AO output setting 2 (-1000–1000, in which 1000 corresponds to 100.0%) 14–18: Reserved 19: Function parameter mapping (PZD2–PZD12 correspond to P14.49–P14.59) 20–31: Reserved		

Function code	Name	Parameter description	Setting range	Default
P23.13–P23.23	PZD2– PZD12 send	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 (×100, A) 6: Actual output torque (×10, %) 7: Actual output power (×10,%) 8: Rotation speed of running (×1, RPM) 9: Linear speed of running (×1, m/s) 10: Ramp reference frequency (×100, Hz) 11: Fault code 12: AI1 input (×100, V) 13: AI2 input (×100, V) 14: AI3 input (×100, V) 15: Reserved 16: HDI1 frequency value (×100, kHz) 17: Reserved 18: Terminal input state 19: Terminal output status 20: PID reference (×100, %) 21: PID feedback (×100, %) 22–26: Reserved 27: VFD status word 2 28–31: Reserved 32: Function parameter mapping (PZD2–PZD12 correspond to P14.60–P14.70)	0–32	0

Function code	Name	Parameter description	Setting range	Default
P24.00	Expansion card protocol selection	0: PROFINET 1: EtherCAT 2: Reserved 3: EtherNet IP 4: Modbus TCP 5: EtherNet UDP 6: PROFINET + EtherNet UDP 7: EtherCAT + EtherNet UDP 8–14: Reserved 15: No communication expansion card	0–15	0
P24.02	Ethernet monitoring card IP address 1	-	0–255	192
P24.03	Ethernet monitoring card IP address 2	-	0–255	168
P24.04	Ethernet monitoring card IP address 3	-	0–255	0
P24.05	Ethernet monitoring card IP address 4	-	0–255	1
P24.06	Ethernet monitoring card subnet mask 1	-	0–255	255
P24.07	Ethernet monitoring card subnet mask 2	-	0–255	255
P24.08	Ethernet monitoring card subnet mask 3	-	0–255	255

Function code	Name	Parameter description	Setting range	Default
P24.09	Ethernet monitoring card subnet mask 4	-	0–255	0
P24.24	Time to identify expansion card	-	0.0–600.0	0.0s
P24.27	Expansion card communication timeout time	-	0.0–600.0	0.0s
P24.30	EtherCAT communication timeout time	-	0.0–60.0	5.0s
P24.31	PROFINET communication timeout time	-	0.0–60.0	5.0s
P24.32	EtherNet IP communication timeout time	-	0.0–60.0	5.0s
P24.34	Modbus TCP communication timeout time	-	0.0–60.0	5.0s
P24.37	Industrial Ethernet communication card IP address 1	-	0–255	192
P24.38	Industrial Ethernet communicationcard IP address 2	-	0–255	168
P24.39	Industrial Ethernet communicationcard IP address 3	-	0–255	0
P24.40	Industrial Ethernet communication card IP address 4	-	0–255	20

Function code	Name	Parameter description	Setting range	Default
P24.41	Industrial Ethernet communication card subnet mask 1	-	0–255	255
P24.42	Industrial Ethernet communication card subnet mask 2	-	0–255	255
P24.43	Industrial Ethernet communication card subnet mask 3	-	0–255	255
P24.44	Industrial Ethernet communication card subnet mask 4	-	0–255	0
P24.49	Saving EtherCAT written function codes	0: Do not save 1: Save	0–1	0
P24.50	EtherCAT DC synchronization cycle	0–1: Reserved 2: 1ms 3: 2ms 4: 4ms 5: 8ms	0–5	0
P24.51	EtherCAT slave station address	-	0x0000– 0xFFFF	0xFFFF
P29.00	Expansion card type	0: No card 1–35: Reserved 36: All-in-one expansion card—PROFINET communication card 37–40: Reserved 41: All-in-one expansion	0–63	0

Function code	Name	Parameter description	Setting range	Default
		card—EtherCAT communication card 42: Reserved 43: All-in-one expansion card—EtherNet IP communication card 44: All-in-one expansion card—Modbus TCP communication card 45: All-in-one expansion card—Ethernet communication card 46: All-in-one expansion card—PROFINET + Ethernet communication card 47: All-in-one expansion card—EtherCAT + Ethernet communication card 48–63: Reserved		
P29.03	Expansion card software version	-	0.00–655.35	0.00
P29.32	EtherCAT control word	-	0x0000–0xFFFF	0x0000
P29.33	EtherCAT status word	-	0x0000–0xFFFF	0x0000

z : s y s t e m s

z:systems GmbH

Messestraße 11

6850 Dornbirn

Austria

Tel: +43/5572/202601-0

Fax: +43/5572/202601-20

E-mail: info@zsystems.at