# Preface

**Introduction to Materials**

The CM680 is a high-performance general-purpose inverter, featuring a book-type design across the entire series to meet more installation area requirements. The product is equipped with rich hardware configurations and powerful software performance, supporting a variety of communication protocols and efficient control of various types of motors. It offers excellent drive performance and control functions, enabling the application of automated production equipment in industries such as metallurgy, lifting, machine tools, printing, wire drawing, glass, food, fans, and pumps.

This manual introduces the communication expansion card, including its overview, composition, dimensions, installation, electrical connections, and parameter configuration.

**More Information**

| Document Name | Content Summary |
|---|---|
| CM680 Series General Purpose Inverter<br>Quick Installation and Commissioning Manual | Introduces detailed content on product installation, wiring, commissioning, fault handling, function codes, fault codes, etc. |
| CM680 Series General Purpose Inverter<br>Hardware Manual | Introduces the system composition, technical specifications, components, dimensions, optional accessories (installation accessories, cables, peripheral electrical components), expansion cards, and detailed content on daily maintenance and care, compliance with certifications and standards, etc. |
| CM680 Series General Purpose Inverter<br>Installation Manual | Introduces the installation dimensions, space design, detailed installation steps, wiring requirements, cabling requirements, installation requirements for optional accessories, and detailed content on common EMC issue resolution suggestions, etc. |
| CM680 Series General Purpose Inverter<br>Software Manual | Introduces the detailed content of the product's functional applications, communication, fault codes, function codes, etc. |
| CM680 Series General Purpose Inverter<br>Communication Manual | Introduces the detailed content of the product communication expansion card's overview, composition, dimensions, installation, electrical connections, communication parameter configuration, fault codes, etc. |

Version Change Record

| Revision Date | Release Version | Change Content |
|---|---|---|
| 2023–12 | Ver 1.0 | First release of the manual. |
| 2023–04 | Ver 1.1 | Added methods for reading and writing inverter function codes using CANopen, etc. |
| 2024–09 | Ver 1.2 | Modified some parameters and functions. |

**About Manual Acquisition**

This manual is not shipped with the product. If you need to obtain the electronic PDF version, you can do so through the following methods:

Log in to the Changsha Riye Electric official website (www.cssunye.com), go to the “Download Center,” search for the keyword, and download.

**Warranty Statement**

Under normal usage conditions, if the product malfunctions or is damaged, we provide warranty service within the warranty period. After the warranty period, repair fees will be charged.

Within the warranty period, the following situations causing product damage will incur repair fees.

- Damage caused by not operating the product according to the manual.
- Damage caused by fire, water, or abnormal voltage.
- Damage caused by using the product for non-standard functions.
- Damage caused by use beyond the specified product usage range.
- Secondary damage caused by force majeure (natural disasters, earthquakes, lightning strikes).

Service fees are calculated according to the manufacturer's unified standards. In case of a contract, the contract takes precedence. For detailed warranty information, please refer to the 'Product Warranty Card'.

# Table of Contents

# Precautions

**Safety Statement**

1. This chapter explains the safety precautions necessary for the proper use of this product. Before using this product, please read the user manual and correctly understand the information related to safety precautions. Failure to comply with the matters specified in the safety precautions may result in death, serious injury, or equipment damage.

2. The 'Danger', 'Warning', and 'Caution' items in the manual do not represent all safety matters that should be observed, but are provided as a supplement to all safety precautions.

3. This product should be used in an environment that meets the design specifications; otherwise, it may cause malfunctions. Any functional abnormalities or component damage caused by non-compliance with these specifications are not covered by the product warranty.

4. Our company will not assume any legal responsibility for personal injuries or property damage caused by non-compliance with the content of this manual or improper operation of the product.

**Definition of Safety Levels**

In this manual, safety precautions are divided into the following two categories:

 Danger: Situations where failure to follow the required procedures can result in serious injury or even death;

 Caution: Situations where failure to follow the required procedures can result in moderate injury or minor injury, as well as equipment damage;

Please read this chapter carefully when installing, debugging, and maintaining this system, and strictly follow the safety precautions outlined in this chapter. The company will not be liable for any injuries or losses resulting from non-compliance with the operating procedures.

**Safety Precautions**

| Usage Phase | Safety Level | Matters |
|---|---|---|
| Before Installation |  Danger | If you find that the control system is waterlogged, parts are missing, or components are damaged upon unpacking, do not install!<br>If the packing list does not match the actual items, do not install! |
| |  Caution | Handle with care when moving to avoid the risk of damaging the equipment!<br>Do not touch the components of the control system to avoid the risk of static damage! |
| During installation |  Danger | Only trained personnel with knowledge of electrical equipment and electrical expertise should operate. Non-professionals are strictly prohibited from operating! |
| |  Caution | Do not allow wire ends or screws to fall into the drive, as this can cause damage to the drive!<br>Please install the drive in a location with minimal vibration and avoid direct sunlight. |
| When wiring |  Danger | Installation must be performed by a professional electrical engineer; otherwise, unexpected dangers may occur!<br>Before wiring, ensure that the power supply is in a zero-energy state to avoid the risk of electric shock!<br>Please ensure that the inverter is properly and correctly grounded according to standards to avoid the risk of electric shock! |

| Usage Phase | Safety Level | Matters |
|---|---|---|
| | Caution | Please follow the steps specified in the Electrostatic Discharge (ESD) prevention measures and wear an ESD wrist strap when performing wiring and other operations to avoid damaging the equipment or internal circuits of the product.<br>When wiring the control circuit, please use twisted pair shielded wire and connect the shield to the product's grounding terminal for grounding; otherwise, it may cause abnormal operation of the product. |
| Before Powering On | Danger | Before powering on, please ensure that the product is installed properly, the wiring is secure, and the motor device is allowed to restart.<br>Before powering on, please confirm that the power supply meets the product requirements to avoid damaging the product or causing a disaster!<br>Do not open the product cabinet door or protective cover, touch any wiring terminals, or remove any components or parts while the power is on, as there is a risk of electric shock! |
| | Caution | The inverter must be covered with the cover before power can be applied; otherwise, there is a risk of electric shock!<br>Wire according to the circuit connection methods provided in this manual; otherwise, accidents may occur! |
| After power-on | Danger | Do not open the cover after power-on. Otherwise, there is a risk of electric shock!<br>Do not touch any input or output terminals of the inverter. Otherwise, there is a risk of electric shock! |
| | Caution | If parameter identification is required, be aware of the danger of injury from the rotating motor; otherwise, accidents may occur!<br>Do not arbitrarily change the manufacturer's parameters of the inverter; otherwise, it may cause damage to the equipment! |
| During operation | Danger | Non-professional technical personnel should not test signals during operation; otherwise, it may cause personal injury or equipment damage!<br>Do not touch the cooling fan or discharge resistor to test the temperature; otherwise, it may cause burns! |
| | Caution | During inverter operation, avoid dropping objects into the equipment, as this can cause damage to the device!<br>Do not use contactors to control the start and stop of the drive, as this can cause damage to the equipment! |
| During maintenance | Danger | Do not perform repairs or maintenance on the inverter without professional training, as this can result in personal injury or equipment damage!<br>Do not perform repairs or maintenance on the equipment while it is energized, as this poses an electric shock hazard!<br>Ensure that the inverter's input power has been disconnected for at least 10 minutes before performing maintenance or repairs on the drive, as residual charges in the capacitors can cause injury!<br>All pluggable modules must be inserted or removed with the power off!<br>After replacing the communication module, parameter settings and checks must be performed. |

# 1. List of Communication Protocols

| Supported Communication Protocols | Communication Hardware | |
|---|---|---|
| PROFIBUS DP Communication | External Communication Expansion Module | EMH-DP Communication Expansion Card |
| PROFINET Industrial Ethernet Communication | | EMH-PN Communication Expansion Card |
| EtherNet/IP Industrial Ethernet Communication | | EMH-EN Communication Expansion Card |
| EtherCAT Industrial Ethernet Communication | | EMH-EC Communication Expansion Card |
| CANopen Communication | | EMH-OP Communication Expansion Card |

# 2. PROFIBUS DP Communication

## 2.1. Product Information

The EMH-DP Communication Expansion Card is a PROFIBUS fieldbus adapter card that complies with the international standard for PROFIBUS-DP fieldbus. This card is installed on the CM680 Inverter to enable communication with PROFIBUS master station devices, making the inverter a slave station of PROFIBUS DP, accepting control from the master station.

### 2.1.1. Physical Dimensions

Physical dimensions of the PROFIBUS Communication Expansion Card: PCB length 95mm, width 37mm, mounting hole diameter 3.5mm.

**Figure 1 Physical Dimensions of the EMH-DP Communication Expansion Card**

### 2.1.2. Interface Layout

The hardware layout of the EMH-DP Communication Expansion Card is shown in Figure 2. Pin Header X3 is used for connecting to the inverter and is located on the back of the EMH-DP Communication Expansion Card. The EMH-DP Communication Expansion Card provides a 9-pin D-type connector for PROFIBUS communication. Detailed descriptions of each hardware component are provided in Table 1.

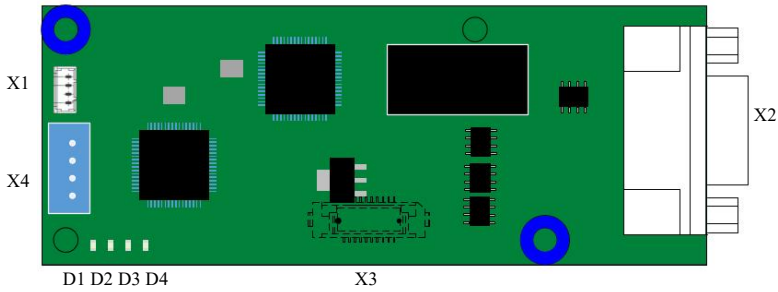



**Figure 2 EMH-DP Communication Expansion Card Interface Layout**

**Table 1 EMH-DP Communication Expansion Card Interface Description**

| Diagram Name | Hardware Name | Function Description |
|---|---|---|
| X1 | Socket | UART Interface: Print Operation Information |
| X2 | 9-pin D-type Connector | For Module Connection to PROFIBUS Network, Refer to Figure 1.3 Description |
| X3 | Board-to-Board Socket | For Connection to Inverter |
| X4 | Socket | For Firmware Download and Debugging |
| D1, D2, D3, D4 | LED Indicator Light | Used to indicate the operating status, refer to Table 1.2 for details |

1. PROFIBUS-DP Communication Interface

The EMH-DP Communication Expansion Card uses a standard DB9 connector to connect with the PROFIBUS master station, and its pin signal definitions follow the SIEMENS DB9 connector standard distribution. As shown in the following figure:

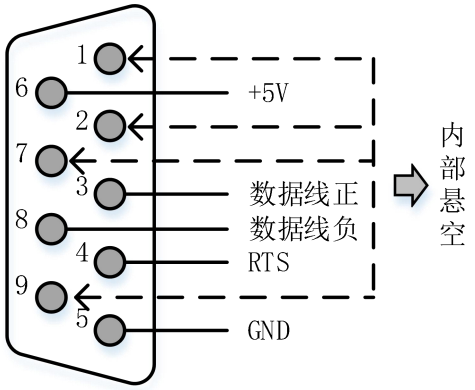



**Figure 3 EMH-DP Communication Expansion Card DP Communication Interface Description**

2. LED Indicator Light

**Table 2 EMH-DP Communication Expansion Card LED Indicator Light Description**

| Diagram Name | Indicator Light | Indicator Light Status | | Function Description |
|---|---|---|---|---|
| D1 | PROFIBUS Communication Status Indicator Light | Red Light | Constantly On | Indicates no communication between the module and the PROFIBUS master station (check PROFIBUS cable connections and station number) |
| | | | Off | Indicates normal communication between the module and the PROFIBUS master station |
| D2 | Inverter communication status indicator light | Red Light | Constantly On | Communication timeout with the inverter |
| | | | Flashing | Abnormal communication with the inverter (interference exists, poor connection, etc.), communication format error (register address error, data error, etc.) |
| | | | Off | Normal communication with the inverter |
| D3 | Program run | Green | Constantly | Main program running normally |

| | | | | |
|---|---|---|---|---|
| | status indicator light | light | On | |
| | | | Off | Main program not started properly |
| D4 | Hardware power indicator light | Green light | Constantly On | Module powered on normally |
| | | | Off | Module not powered on |

## 2.2. Installation and Wiring

### 2.2.1. Installation

The EMH-DP Communication Expansion Card is designed to be embedded in the CM680 Inverter. Before installation, please disconnect the power supply to the inverter, wait for about 10 minutes, and ensure that the inverter's charging indicator light is completely off before proceeding with the installation. After inserting the EMH-DP Communication Expansion Card into the inverter, please secure the corresponding screws to prevent poor contact between the board and the signal socket. The installation diagram is shown in Figure 4.

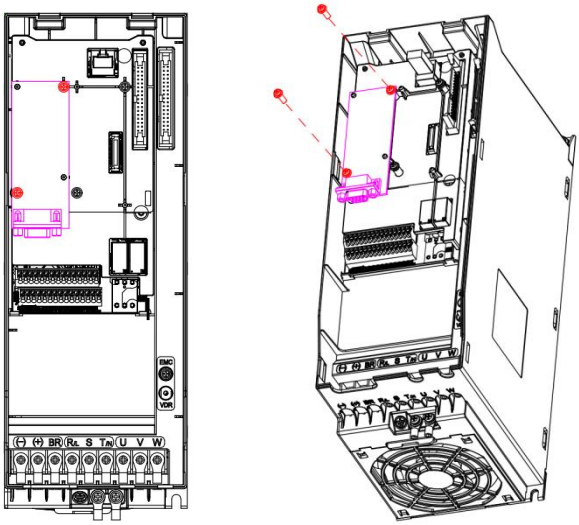


**Figure 4 EMH-DP Communication Expansion Card Installation Diagram**

### 2.2.2. Wiring

The wiring diagram for the PROFIBUS DP bus is shown in Figure 5. Terminal matching resistors must be connected at both ends of the PROFIBUS bus. Adjust the DIP switch according to the markings on the terminal block. Failure to connect or insufficient terminal resistors can affect communication quality, leading to unstable communication.

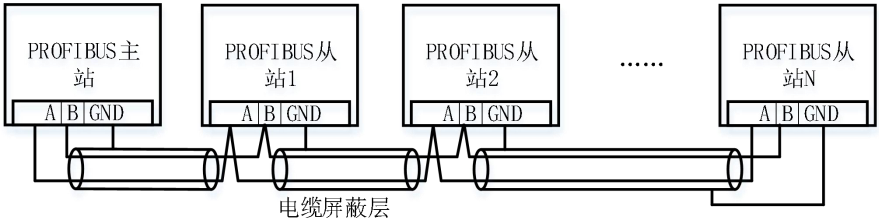



**Figure 5 Schematic Diagram of PROFIBUS DP Bus Wiring**

The PROFIBUS-DP communication rate ranges from 9.6 kbps to 12 Mbps, and the transmission line length must be determined based on the transmission rate, with a transmission distance range of 100m to 1200m. The communication rates supported by the EMH-DP Communication Expansion Card and the communication distances at each rate are shown in Table 3.

**Table 3 Maximum Cable Length for Different PROFIBUS-DP Transmission Rates**

| Communication Rate (bps) | 9.6k | 19.2k | 93.75k | 187.5k | 500k | 1.5M | 3M | 6M | 12M |
|---|---|---|---|---|---|---|---|---|---|
| Length (m) | 1200 | 1200 | 1200 | 1000 | 400 | 200 | 100 | 100 | 100 |

## 2.3. Electrical Connection

Bus Cable Description

It is recommended to use dedicated Profibus DP bus cables. Cable parameters are shown in the table below.

**Table 4 Cable Parameters**

| Parameter | Description |
|---|---|
| Conductor | 1 pair (2×22AWG) solid copper wire |
| Insulation Jacket Color | Green, Red |
| Shielding | Aluminum foil tape + tinned copper wire braid |
| Jacket Material | PVC |
| Appearance | Purple |

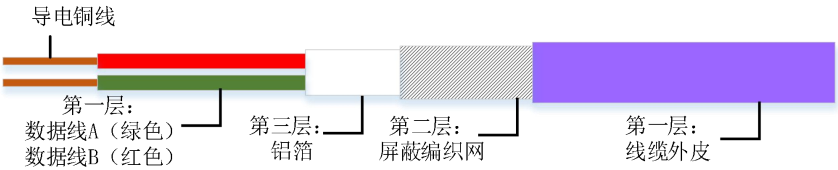



**Fig. 6 Schematic Diagram of Bus Cable Inner Structure**

Bus Terminal Description:

It is recommended to use SIEMENS Profibus DP dedicated connector, model: 6ES7 972-0BB12-0XA

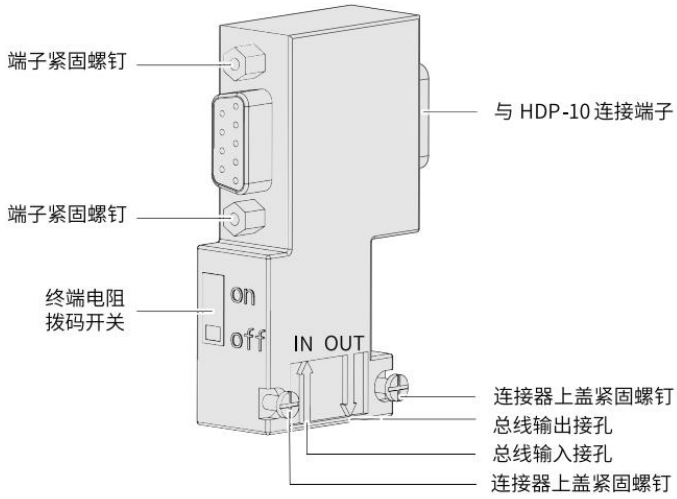



**Fig. 7 Schematic Diagram of Bus Connection Terminal Layout**

Bus Connection Steps

1. Remove the outer sheath of the cable, and leave the length according to the dimensions below.

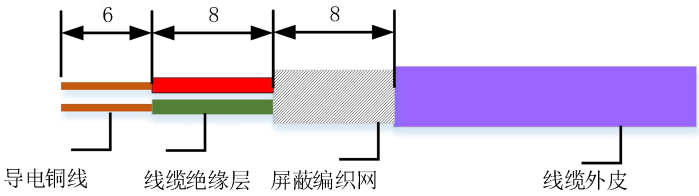



**Fig. 8 Cable Length Reserve Diagram (Unit: mm)**

2. Use a No. 1 flathead screwdriver to open the bus connector terminal cover.

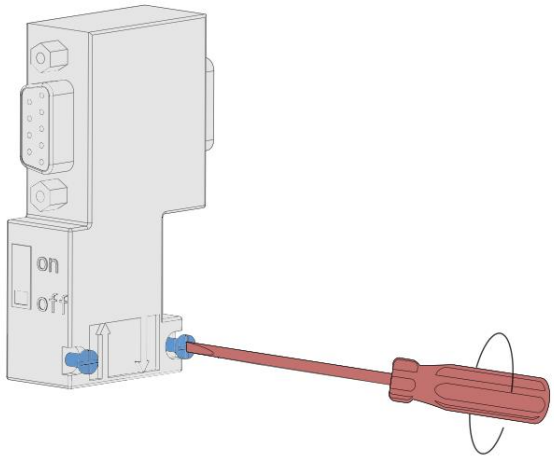



**Fig. 9 Opening the Bus Connector Terminal Cover**

3. Use a No. 1 flathead screwdriver to secure the cable to the connector installation position, ensuring that the shielding layer is in tight contact with the shielding layer metal plate, green cable connects to A, red cable connects to B, as shown in the figure below.

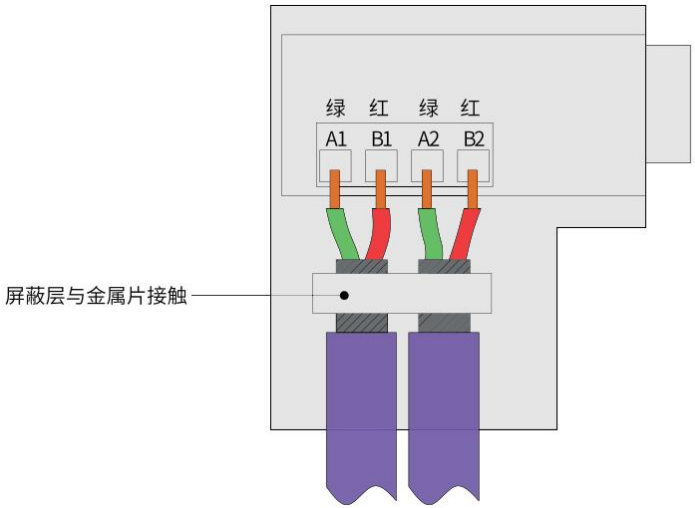



**Figure 10 Shield Layer in Contact with Metal Plate**

4. After covering the connector cover and tightening the screws on the cover, insert the CM680-PROFIBUS corresponding DB9 port, and use a screwdriver to tighten the fixing screws on both sides to prevent loosening.

Note: When installing the Profibus DP bus, ensure that the screws on both sides of the Siemens terminal are securely connected to the EMH-DP module; otherwise, communication may fail or the communication quality may decrease.

## 2.4. Communication Description

### 2.4.1. Key Points of PROFIBUS Communication

1. When using the EMH-DP communication expansion card, it is necessary to set the communication decoding method of the CM680. Set the F8-19 register to 0 via the operation panel, and set the F8-07 register to 0 or 1 according to the actual decoding method used.

2. When setting the inverter command, the F0-05 register of the CM680 must be set to 5.

3. When setting the inverter frequency, the F0-06 register of the CM680 must be set to 8.

4. The device address for PROFIBUS communication on the EMH-DP Communication Expansion Card is set in the F8-11 register. After setting, the module must be restarted for the changes to take effect.

5. During non-periodic communication, a new non-periodic read/write operation must not be initiated until the previous one is completed. The interval between two non-periodic communications should be no less than 100ms to avoid affecting the response of periodic communication.

### 2.4.2. Periodic Data Format

Through PZD area data, the master station can update and read inverter data in real-time and perform periodic data exchange. The communication address is configured by the user during setup.

The EMH-DP Communication Expansion Card provides multiple Modules to transmit PZD data of different lengths. Users select a specific Module for configuration during setup according to their needs. The supported data length for each data format is listed in the table below:

**Table 5 Supported Data Length for Each Data Format**

| Module | Data Length |
|---|---|
| PZD-2/3 | Cyclic Write (master->slave): PZD1~PZD2<br>Cyclic Read (slave->master): PZD1~PZD3 |
| PZD-6/6 | Cyclic Write (master->slave): PZD1~PZD6<br>Cyclic Read (slave->master): PZD1~PZD6 |
| PZD-10/10 | Cyclic Write (master->slave): PZD1~PZD10<br>Cyclic Read (slave->master): PZD1~PZD10 |
| PZD-16/16 | Cyclic Write (master->slave): PZD1~PZD16<br>Cyclic Read (slave->master): PZD1~PZD16 |

When configuring the newly installed communication expansion card, each PZD has a default address, as follows:

- PZD1 (master->slave): 24576 (0x6000), Control Command
- PZD2 (master->slave): 24578 (0x6002), Target Frequency
- PZD1 (slave->master): 8448 (0x2100), Errors and Warnings (Fixed Address)
- PZD2 (slave->master): 24832 (0x6100), Current Status
- PZD3（slave->master）: 24834 (0x6102), actual frequency

● Other addresses: 2096 (0x0830), reserved address, unused PZD should be set to this address to avoid affecting other functions

### 2.4.3. Non-periodic data format

Non-periodic interaction uses slot-index addressing to determine the format of the message data area. To facilitate user operations on consecutive registers of the inverter, the following non-periodic read/write format is defined, with the index set to 100.

1. Non-periodic write

When executing a non-periodic write command, the data area must include the starting address of the register, length, and value, with the function code fixed at 0x10, indicating a write command. The specific format is as follows:

**Table 6 Non-periodic write data format**

| | Function Code | Register Address | Register Length | Data |
|---|---|---|---|---|
| Data Length | 1 Byte | 2 Byte | 1 Byte | (Register Length * 2) Byte |
| Example | 0x10 | 0x1010 | 0x02 | 0x01020304 |

The total length of the data area for non-periodic write values is at least 6 Byte and at most 24 Byte.

2. Non-periodic Read Value

In the non-periodic read value request message, the data area is empty and cannot transmit the register address to be read. A non-periodic write value command must be sent first to cache the register address in the module. The function code for this non-periodic write address command is fixed at 0x03, and the data area length is fixed at 3 Byte. The specific format is as follows:

**Table 7 Non-periodic Read Data Format**

| | Function Code | Register Address |
|---|---|---|
| Data Length | 1 Byte | 2 Byte |
| Example | 0x03 | 0x1010 |

After executing the non-periodic write address command, send the non-periodic read value command. The read value request message includes the register length to be read, and the data area is empty. The data area of the response message is the register value. The format of the data area in the response message is as follows:

**Table 8 Format of the Response Message Data Area**

| | Data |
|---|---|
| Data Length | (Register Length * 2) Byte |
| Example | 0x01020304 |

## 2.5. Communication Configuration

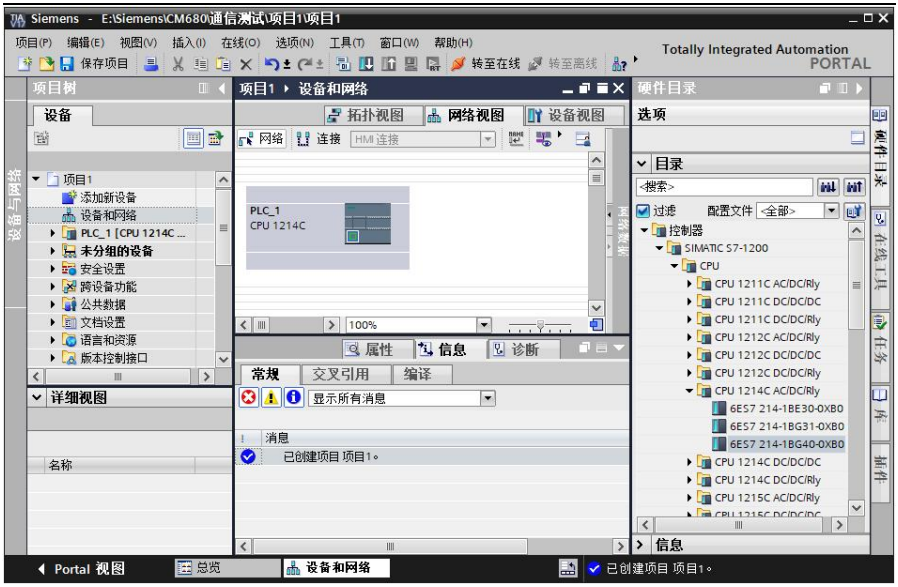
Taking the Siemens S7-1200 series PLC as an example, configure the PLC to establish PROFIBUS communication with the module.

### 2.5.1. Configure Configuration

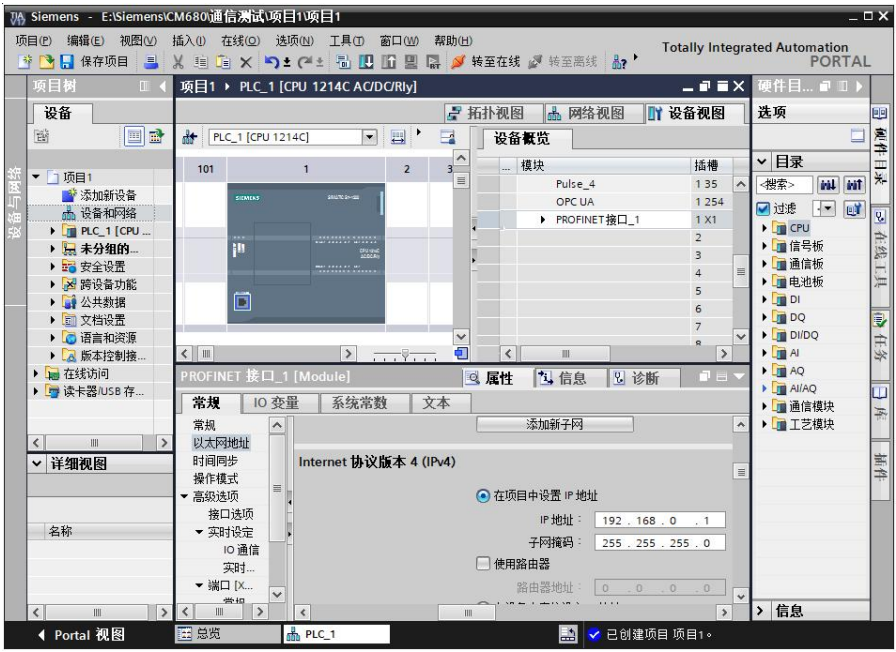
1. Create a project

Double-click to open the TIA Portal software, select 'Create New Project', and the following interface will appear. Choose the save path and project name, then click 'Create'.

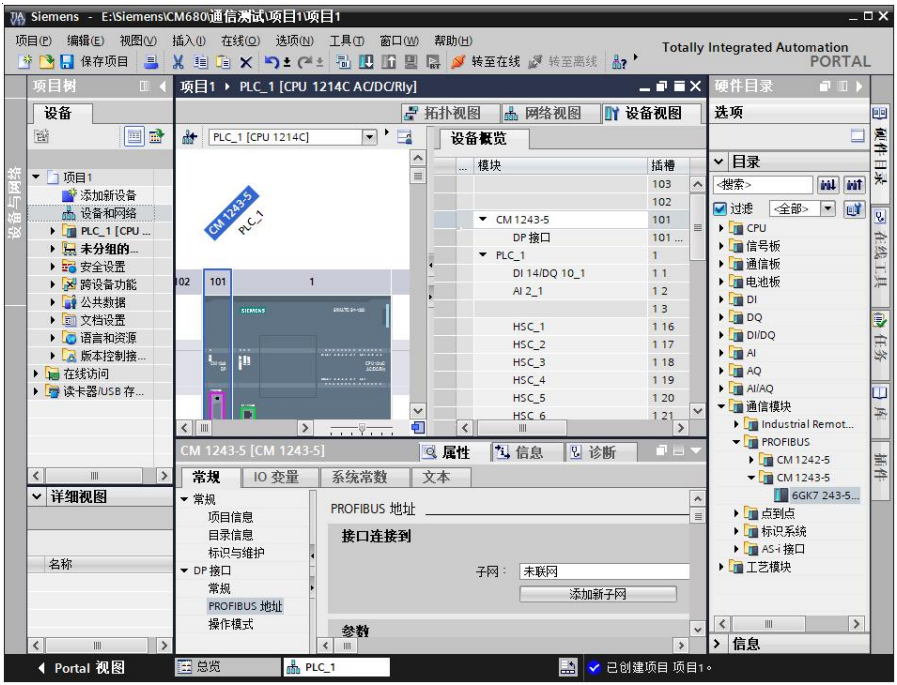

After creation, click 'Project View' to enter the main interface. In the 'Hardware Catalog' on the right, select the actual PLC model being used, for example, S7-1214. Double-click to see the PLC in the configuration page.




Click on the PLC, enter the 'Device View' page, and in the properties bar, you can configure the IP address and other information of the PLC, which can be modified according to the actual situation.



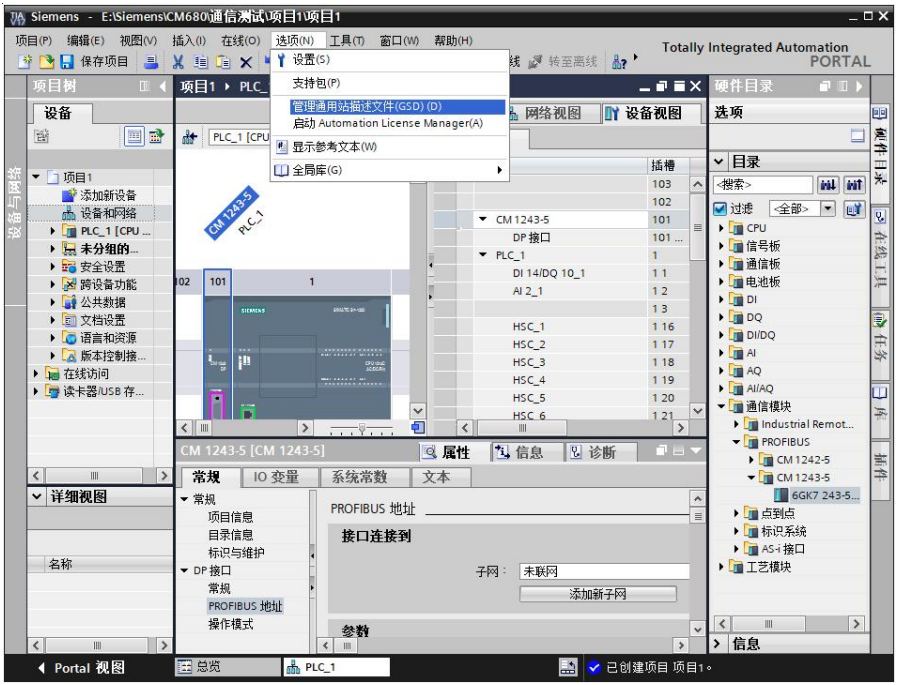
In the hardware catalog, select the CM1243-5 communication module, add it next to the PLC, click on this communication module, and you can configure its PROFIBUS address and other parameters.




This completes the establishment of the master station.

2. Install GSD File

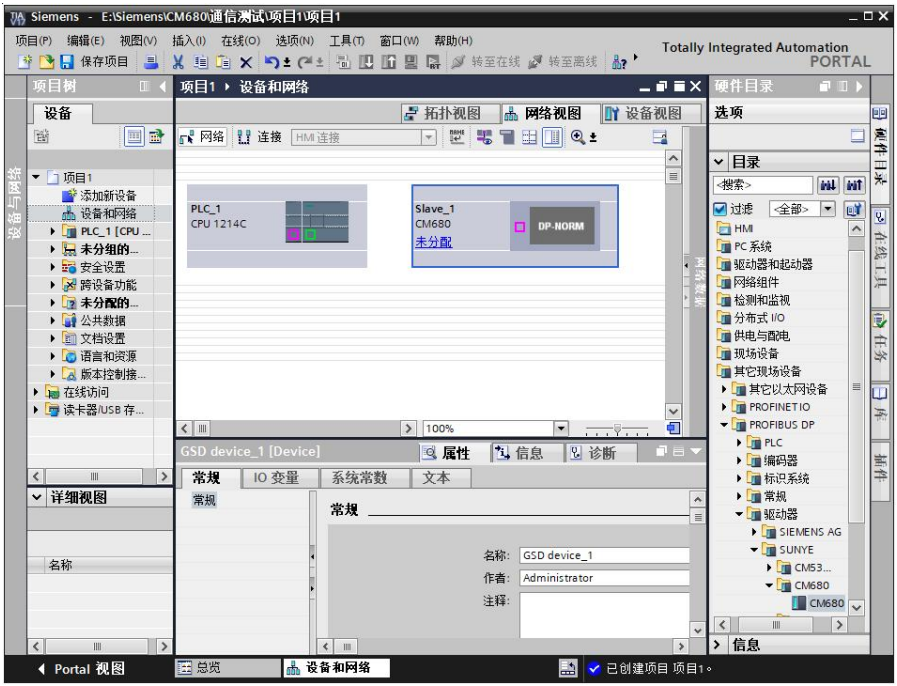
If the GSD file has not been installed before, it needs to be installed. In 'Options', select 'Manage General Station Description (GSD) Files'.



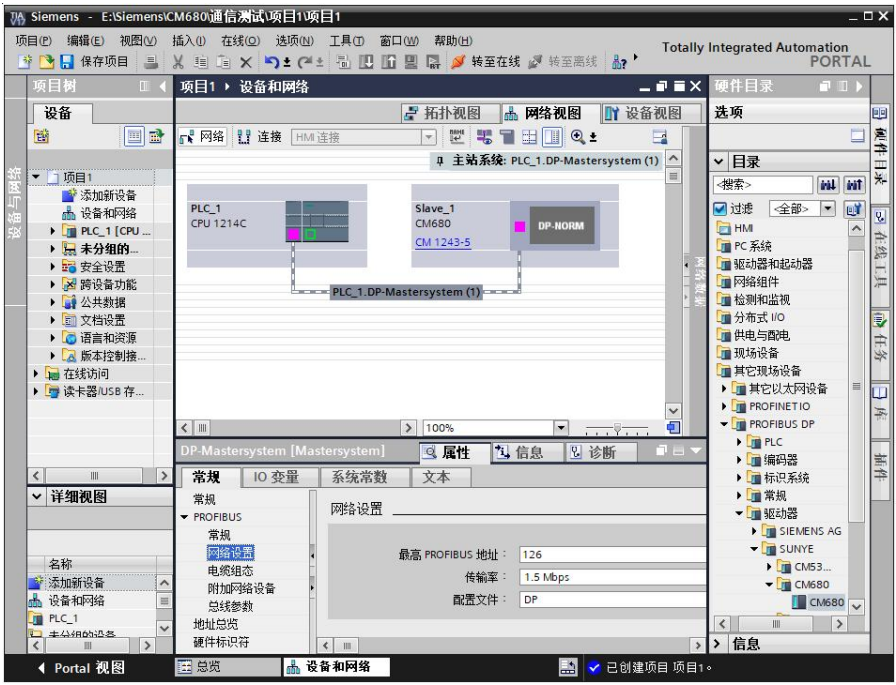
Select the path where the GSD file is stored (Note: Do not store the GSD file in a path with Chinese characters, as this may cause errors), check the GSD file to be installed, click 'Install'. After installation is complete, select 'Close'.

3. Configure Slave Station

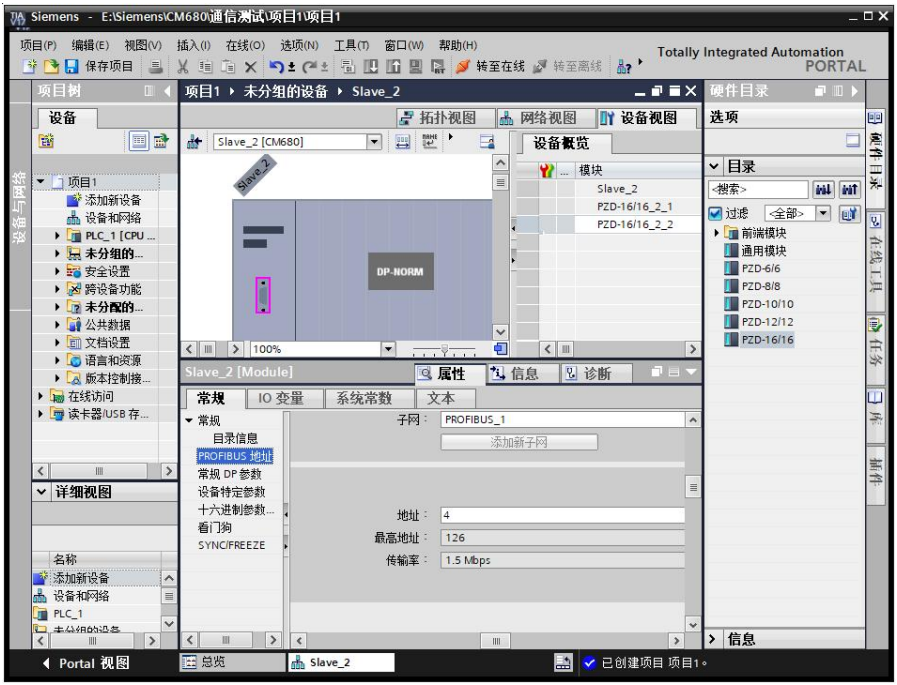
Return to the “Device and Network” — “Network View” interface, in the “Hardware Catalog” on the right, find the CM680 position under Other Field Devices — PROFIBUS DP — Drive — SUNYE, and double-click to add the module.



Click on “Unassigned” on the module in the “Device and Network” interface to assign the master station system to the module. Click on the DP connection line to configure parameters such as the DP transmission baud rate.



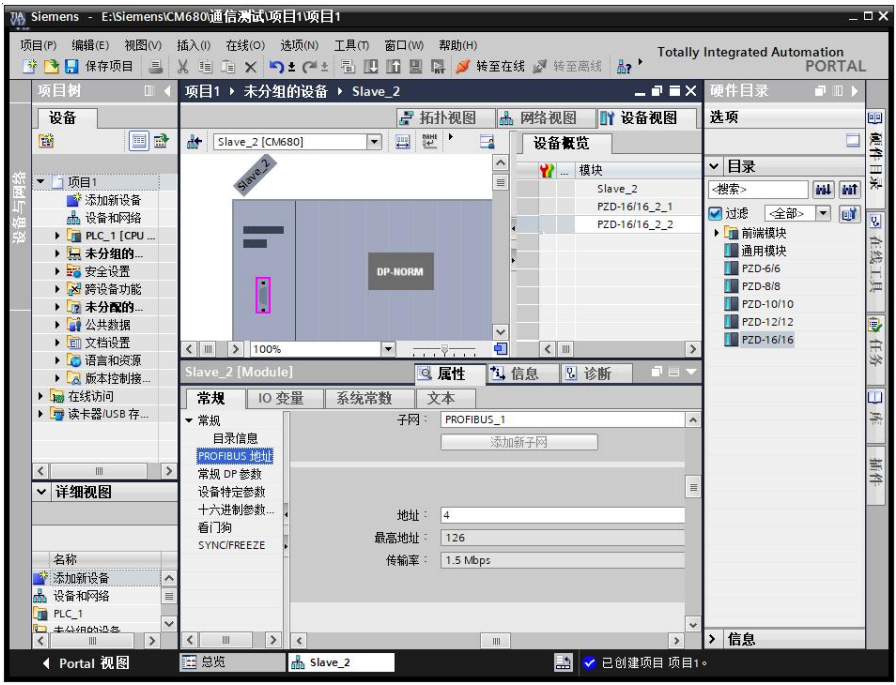
Select the slave station, and in “Properties” -> “General” -> “PROFIBUS Address” -> “Parameters,” set the PROFIBUS device slave address. The slave address must be different from the master address.



### 2.5.2. Configure Periodic Communication

1. Configure Module

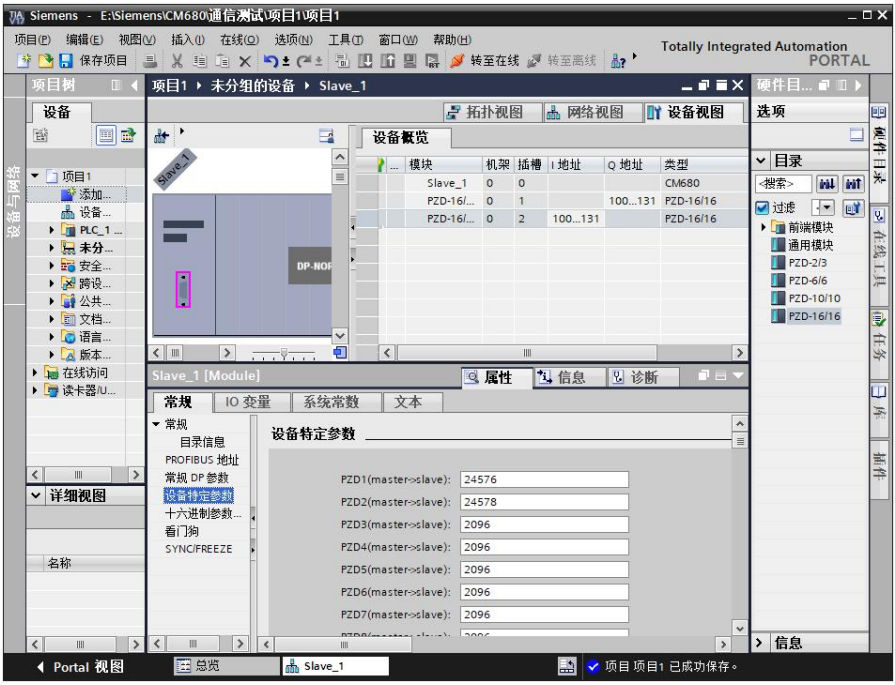
Select the slave station, switch to the “Device View,” and under “Hardware Catalog” -> “Module,” double-click to select the data length to be configured for the slave station. The configured module will be added in the device view. After adding the module, you can modify the input/output mapping in the PLC by editing the “I Address” and “Q Address”.



2. Configure PZD

Through PZD area data, the master station can real-time modify and read inverter data, and perform periodic data exchange.

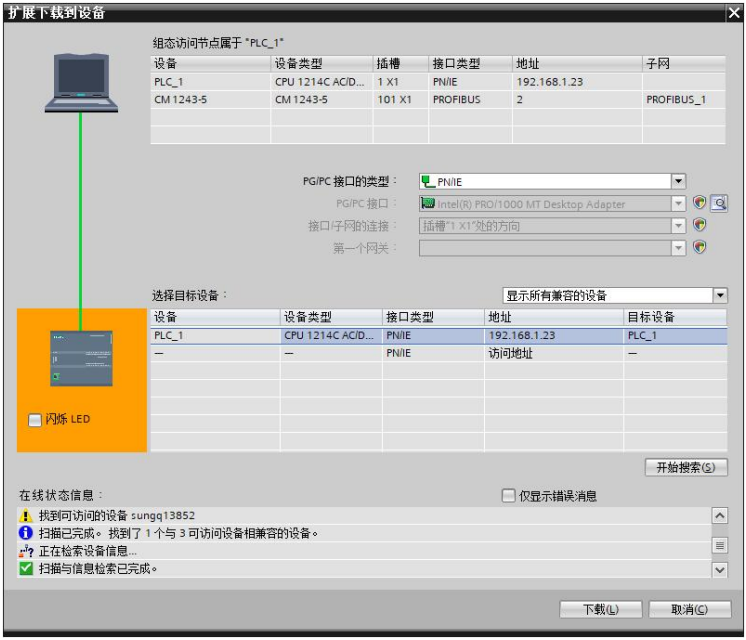
Select the slave station, under “Properties” -> “General” select “Device Specific Parameters”, set the register address for user-defined periodic data. PZDx(Master->Slave) indicates the address where the master station writes to the slave station, and PZDx(Slave->Master) indicates the address where the master station reads from the slave station, in decimal format. Refer to section 3.2 for default addresses.




After configuration is complete, download the configuration, and the PLC and module will automatically perform periodic data exchange.

3. Download Configuration and Ladder Diagram

Save the configured network, set the computer's IP address to the same subnet as the PLC (note that it should not conflict with the IP addresses of the slave stations in the configuration, or you can set the PC to automatically obtain an IP address), compile, click download, select the interface, and then click 'Start Search'. Select the detected PLC, and then click download.




### 2.5.3. Configure Non-periodic Communication

1. Example of Non-periodic Write

Add a data block FB_control, and allocate some variables as follows:

WR1_LEN represents the byte length of the send array; in the example below, it is 6 bytes.

The WR1_Record array is filled with the corresponding content according to the non-periodic interaction format. In the example below, the WR1_Record array contains the 6 bytes to be sent, with the following meanings:

WR1_Record[0]: Function code 0x10 indicates write.

WR1_Record[1] and WR1_Record[2]: indicate that the starting address of the registers to be written is 0xF00E.

WR1_Record[3]: indicates that the number of consecutive registers to be written is 1.

WR1_Record[4] and WR1_Record[5]: indicate that the data to be written to the starting register address (0xF00E) is 0x1499.

FB_control

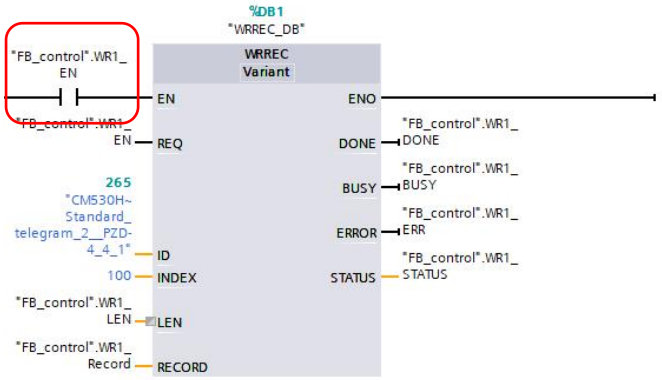
| | 名称 | 数据类型 | 起始值 | 保持 | 从 HMI/OPC... | 从 H... | 在 HMI ... | 设定值 | 监控 | 注释 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Static | | | | | | | | | |
| 2 | WR1_EN | Bool | false | | | | | | | |
| 3 | WR1_DONE | Bool | false | | | | | | | |
| 4 | WR1_BUSY | Bool | false | | | | | | | |
| 5 | WR1_ERR | Bool | false | | | | | | | |
| 6 | WR1_LEN | Byte | 6 | | | | | | | |
| 7 | WR1_STATUS | DWord | 16#0 | | | | | | | |
| 8 | WR1_Record | Array[0..23] of Byte | | | | | | | | |
| 9 | WR1_Record[0] | Byte | 16#10 | | | | | | | |
| 10 | WR1_Record[1] | Byte | 16#F0 | | | | | | | |
| 11 | WR1_Record[2] | Byte | 16#0E | | | | | | | |
| 12 | WR1_Record[3] | Byte | 16#01 | | | | | | | |
| 13 | WR1_Record[4] | Byte | 16#14 | | | | | | | |
| 14 | WR1_Record[5] | Byte | 16#99 | | | | | | | |
| 15 | WR1_Record[6] | Byte | 16#0 | | | | | | | |
| 16 | WR1_Record[7] | Byte | 16#0 | | | | | | | |
| 17 | WR1_Record[8] | Byte | 16#0 | | | | | | | |
| 18 | WR1_Record[9] | Byte | 16#0 | | | | | | | |
| 19 | WR1_Record[10] | Byte | 16#0 | | | | | | | |
| 20 | WR1_Record[11] | Byte | 16#0 | | | | | | | |
| 21 | WR1_Record[12] | Byte | 16#0 | | | | | | | |
| 22 | WR1_Record[13] | Byte | 16#0 | | | | | | | |
| 23 | WR1_Record[14] | Byte | 16#0 | | | | | | | |
| 24 | WR1_Record[15] | Byte | 16#0 | | | | | | | |
| 25 | WR1_Record[16] | Byte | 16#0 | | | | | | | |
| 26 | WR1_Record[17] | Byte | 16#0 | | | | | | | |

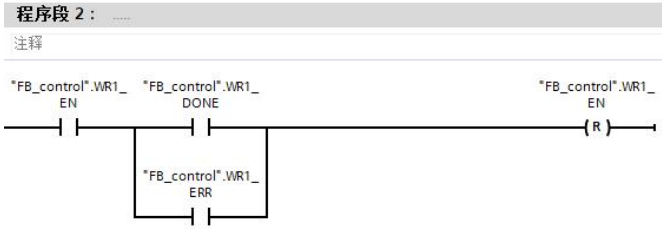
Add the following program in the main program block:

Function block name: WRREC

ID: The ID of the module configured during PROFIBUS module configuration

INDEX: 100



Recompile the PLC program and download it to the PLC. Switch to online mode. Set “FB_control.WR1_EN” to 1, which will trigger a non-periodic write action.

Monitor the FB_control data block online to see the execution results. “FB_control.WR1_Done” being TRUE indicates successful execution, “FB_control.WR1_ERR” being TRUE indicates an execution error, with the error code stored in “FB_control.WR1_STATUS”. Refer to Section 5.2 for specific meanings.

| 名称 | 数据类型 | 起始值 | 监视值 | 保持 |
|---|---|---|---|---|
| Static | | | | |
| WR1_EN | Bool | false | FALSE | |
| WR1_DONE | Bool | false | TRUE | |
| WR1_BUSY | Bool | false | FALSE | |
| WR1_ERR | Bool | false | FALSE | |
| WR1_LEN | Byte | 6 | 16#06 | |
| WR1_STATUS | DWord | 16#0 | 16#0000_0000 | |
| WR1_Record | Array[0..23] of Byte | | | |
| WR1_Record[0] | Byte | 16#10 | 16#10 | |
| WR1_Record[1] | Byte | 16#F0 | 16#F0 | |
| WR1_Record[2] | Byte | 16#0E | 16#0E | |
| WR1_Record[3] | Byte | 16#01 | 16#01 | |
| WR1_Record[4] | Byte | 16#14 | 16#14 | |
| WR1_Record[5] | Byte | 16#99 | 16#99 | |
| WR1_Record[6] | Byte | 16#0 | 16#00 | |
| WR1_Record[7] | Byte | 16#0 | 16#00 | |
| WR1_Record[8] | Byte | 16#0 | 16#00 | |
| WR1_Record[9] | Byte | 16#0 | 16#00 | |
| WR1_Record[10] | Byte | 16#0 | 16#00 | |
| WR1_Record[11] | Byte | 16#0 | 16#00 | |
| WR1_Record[12] | Byte | 16#0 | 16#00 | |
| WR1_Record[13] | Byte | 16#0 | 16#00 | |
| WR1_Record[14] | Byte | 16#0 | 16#00 | |
| WR1_Record[15] | Byte | 16#0 | 16#00 | |
| WR1_Record[16] | Byte | 16#0 | 16#00 | |
| WR1_Record[17] | Byte | 16#0 | 16#00 | |
| WR1_Record[18] | Byte | 16#0 | 16#00 | |

2. Non-periodic Read Value

Add a data block FB_control, and allocate some variables as follows:

WR2_Record array: Enter the function code and address for reading

WR2_Record[0] : Read function code 0x03

WR2_Record[1] and WR2_Record[2]: The register address to be read is 0xF00E

WR2_LEN: The length of the data to be written is 3 bytes, i.e., WR_Record[0]~[2].

RD2_MLEN: The length of the data to be read is 2 bytes, i.e., the data of one register.

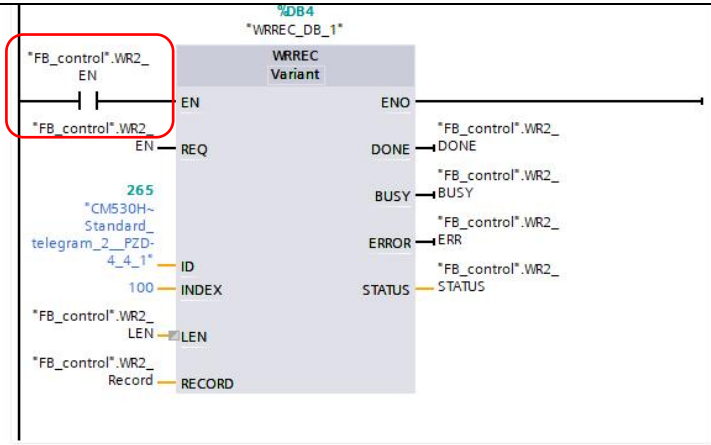
RD2_Record array: The data read is stored in this array.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | | WR2_EN | Bool | false | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 10 | | WR2_DONE | Bool | false | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 11 | | WR2_BUSY | Bool | false | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 12 | | WR2_ERR | Bool | false | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 13 | | WR2_LEN | Byte | 3 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 14 | | WR2_STATUS | DWord | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 15 | | WR2_Record | Array[0...23] of Byte | | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 16 | | WR2_Record[0] | Byte | 16#03 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 17 | | WR2_Record[1] | Byte | 16#F0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 18 | | WR2_Record[2] | Byte | 16#0e | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 19 | | WR2_Record[3] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 20 | | WR2_Record[4] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 21 | | WR2_Record[5] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 22 | | WR2_Record[6] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 23 | | WR2_Record[7] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 24 | | WR2_Record[8] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 25 | | WR2_Record[9] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 26 | | WR2_Record[10] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 27 | | WR2_Record[11] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 28 | | WR2_Record[12] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 29 | | WR2_Record[13] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 30 | | WR2_Record[14] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 31 | | WR2_Record[15] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 32 | | WR2_Record[16] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 33 | | WR2_Record[17] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 34 | | WR2_Record[18] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 35 | | WR2_Record[19] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |
| 36 | | WR2_Record[20] | Byte | 16#0 | ☐ | ☑ | ☑ | ☑ | ☐ | |

Add the following program in the main program block:

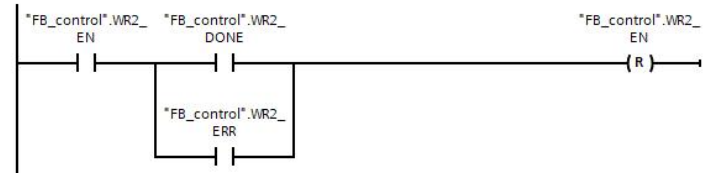
Function block: WRREC is used to write the starting address of the registers to be read

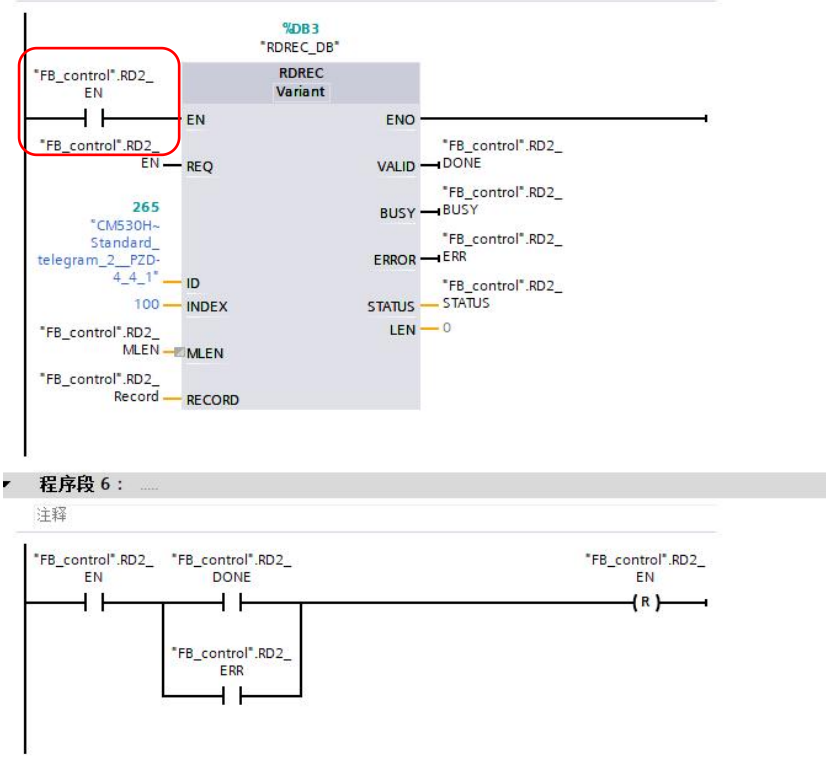
Function Block: RDREC is used to save the data starting from the first address of the read registers.


%DB4
"WRREC_DB_1"
"FB_control".WR2_
EN
WRREC
Variant
EN
ENO
"FB_control".WR2_
EN
REQ
DONE
"FB_control".WR2_
DONE
BUSY
"FB_control".WR2_
BUSY
265
"CM530H~
Standard_
telegram_2__PZD-
4_4_1"
ID
ERROR
"FB_control".WR2_
ERR
100
INDEX
STATUS
"FB_control".WR2_
STATUS
"FB_control".WR2_
LEN
LEN
"FB_control".WR2_
Record
RECORD

**程序段 4：**

注释


"FB_control".WR2_
EN
"FB_control".WR2_
DONE
"FB_control".WR2_
EN
R
"FB_control".WR2_
ERR



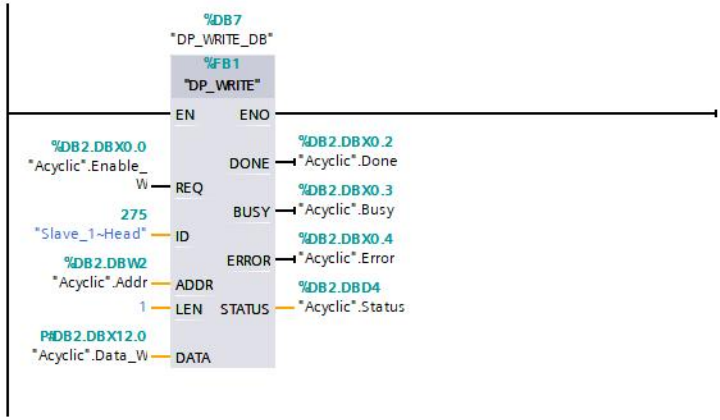

Recompile the PLC program and download it to the PLC. Set 'FB_control.WR2_EN' to 1 to trigger the write read address. When 'FB_control.WR2_Done' is TRUE, it indicates that the write read address was successful. Then set 'FB_control.RD2_EN' to 1, which will trigger the non-periodic read value action. Monitor the 'FB_control' data block online to see the execution results. 'FB_control.RD2_Done' being TRUE indicates successful execution, and 'Error' being TRUE indicates an execution anomaly. The error code is stored in 'FB_control.RD2_STATUS'; refer to Section 5.2 for specific meanings.

| 名称 | 数据类型 | 起始值 | 监视值 | 保持 |
|---|---|---|---|---|
| Static | | | | |
| WR1_EN | Bool | false | FALSE | |
| WR1_DONE | Bool | false | TRUE | |
| WR1_BUSY | Bool | false | FALSE | |
| WR1_ERR | Bool | false | FALSE | |
| WR1_LEN | Byte | 6 | 16#06 | |
| WR1_STATUS | DWord | 16#0 | 16#0000_0000 | |
| WR1_Record | Array[0..23] of Byte | | | |
| WR2_EN | Bool | false | FALSE | |
| WR2_DONE | Bool | false | TRUE | |
| WR2_BUSY | Bool | false | FALSE | |
| WR2_ERR | Bool | false | FALSE | |
| WR2_LEN | Byte | 3 | 16#03 | |
| WR2_STATUS | DWord | 16#0 | 16#0000_0000 | |
| WR2_Record | Array[0..23] of Byte | | | |
| RD2_EN | Bool | false | FALSE | |
| RD2_DONE | Bool | false | TRUE | |
| RD2_BUSY | Bool | false | FALSE | |
| RD2_ERR | Bool | false | FALSE | |
| RD2_STATUS | DWord | 16#0 | 16#0000_0000 | |
| RD2_MLEN | Byte | 2 | 16#02 | |
| RD2_Record | Array[0..23] ... | | | |
| RD2_Record[0] | Byte | 0 | 16#14 | |
| RD2_Record[1] | Byte | 16#0 | 16#99 | |
| RD2_Record[2] | Byte | 16#0 | 16#00 | |
| RD2_Record[3] | Byte | 16#0 | 16#00 | |
| RD2_Record[4] | Byte | 16#0 | 16#00 | |
| RD2_Record[5] | Byte | 16#0 | 16#00 | |
| RD2_Record[6] | Byte | 16#0 | 16#00 | |

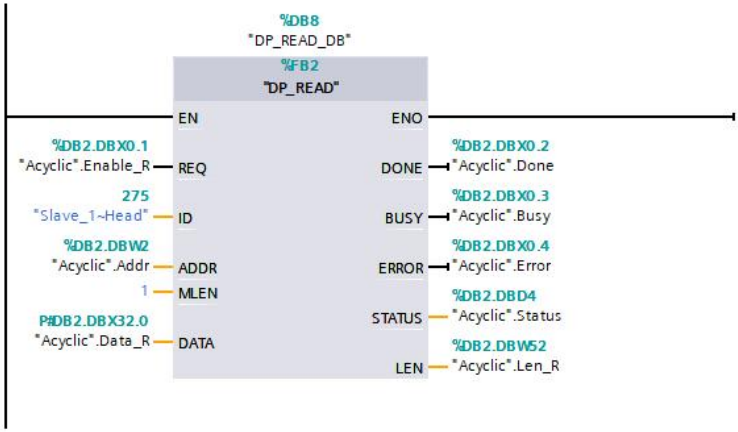
3. Explanation of Encapsulated Non-periodic Read/Write Function Blocks

To allow users to perform non-periodic read/write operations without needing to understand Modbus messages, the manufacturer has encapsulated two non-periodic read/write function blocks for user reference.

DP_WRITE function block: Achieves non-periodic writing, ID: represents the moduleID to be written, ADDR: represents the address of the function code to be written, LEN: represents the number of registers to be written, DATA: array of values to be written to the registers.




DP_READ function block: A function block to achieve non-periodic reading, ID: represents the moduleID to be written, ADDR: represents the address of the function code to be read, MLEN: represents the number of registers to be read, DATA: array to store the values read from the registers.




## 2.6. Fault Information

### 2.6.1. Periodic Fault Code

In the PZD area of periodic communication, the PZD1 sent from the slave station to the master station is fixed to indicate error and alarm information, allowing users to obtain the latest status through periodic communication. In addition to the diagnostic information of the inverter itself, the communication status between the module and the inverter must also be included, allowing users to obtain the cause of errors. Detailed fault status codes are listed in the table below:

**Table 9 PROFIBUS Periodic Fault Status Codes**

| Diagnostic Classification | Status Code | |
|---|---|---|
| Inverter Body Diagnostic Information (Register | 200 | Communication Error (Message Abnormality) |
| | 201 | Communication Error (Timeout Abnormality) |

| | |
|---|---|
| Address: 0x2100) | 202 Address Error<br>203 Parameter Exceeds Threshold |

### 2.6.2. Non-periodic Fault Code

When executing non-periodic communication, the STATUS parameter in the function block is used to return the command execution result, with a data type of DWORD (4 bytes). The data format and meaning are already defined in the TIA software. In addition to the standard definition, the following error codes have the following meanings:

**Table 10 PROFIBUS Non-periodic Fault Status Codes**

| Function _ Num | Error_ Decode | Error_ Code_1 | Error_ Code_2 | Meaning |
|---|---|---|---|---|
| 0xC0 | 0x80 | 0xB8 | 0x01 | Non-periodic Message Function_Num Error (Module only supports 0x5E, 0x5F). |
| 0xC0 | 0x80 | 0xA2 | 0x00 | Non-periodic Data Length Exceeds Limit (0 bytes <= Data Length <= 64 bytes). |
| 0xDE | 0x80 | 0xB0 | 0x00 | Invalid Data Record Number (Index=100, or Index=101). |
| 0xDE | 0x80 | 0xB8 | 0x02 | Non-periodic Read Value Data Length Exceeds Limit (2 bytes <= Data Length <= 24 bytes);<br>Read Fault History Record Length Exceeds Limit (1 byte <= Data Length <= 64 bytes). |
| 0xDE | 0x80 | 0xB8 | 0x03 | Non-periodic Write Address Invalid (WRREC not run in advance to write read address or address error). |
| 0xDE | 0x80 | 0xB8 | 0x05 | Non-periodic Read Value Failed. |
| 0xDF | 0x80 | 0xB0 | 0x00 | Invalid Data Record Number (Index can only be 100 or 101). |
| 0xDF | 0x80 | 0xB8 | 0x01 | Non-periodic Write Value Function Code Error (Only supports 0x03, 0x10 function codes). |
| 0xDF | 0x80 | 0xB8 | 0x02 | Non-periodic write value data length exceeded (3 bytes <= data length <= 24 bytes);<br>Clear fault history record length exceeded (data length = 1 byte). |
| 0xDF | 0x80 | 0xB8 | 0x03 | Non-periodic write address is invalid. |
| 0xDF | 0x80 | 0xB8 | 0x04 | Non-periodic write set value out of range;<br>Clear fault history record set value illegal (set value can only be 0). |

| | | | | |
|---|---|---|---|---|
| 0xDF | 0x80 | 0xB8 | 0x05 | Non-periodic write value failure. |

### 2.6.3. Retrieve Diagnostic Information

When communication between the module and the inverter is abnormal, diagnostic information will be sent to the PLC, which can be obtained through the DPNRM_DG function block from the diagnostic information sent by the slave station.

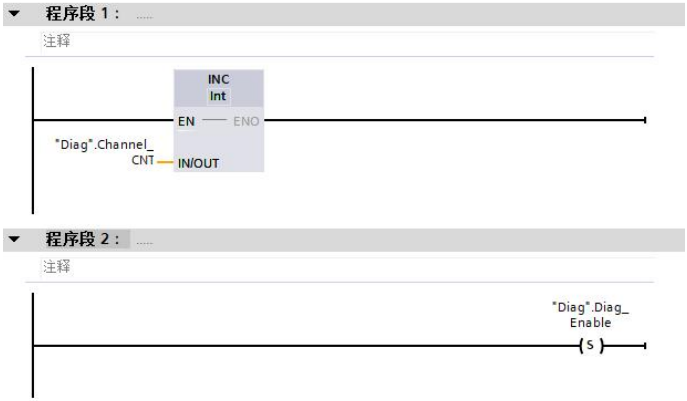
1. Add program block

Add a data block in the program block and include the following variables.

Diag

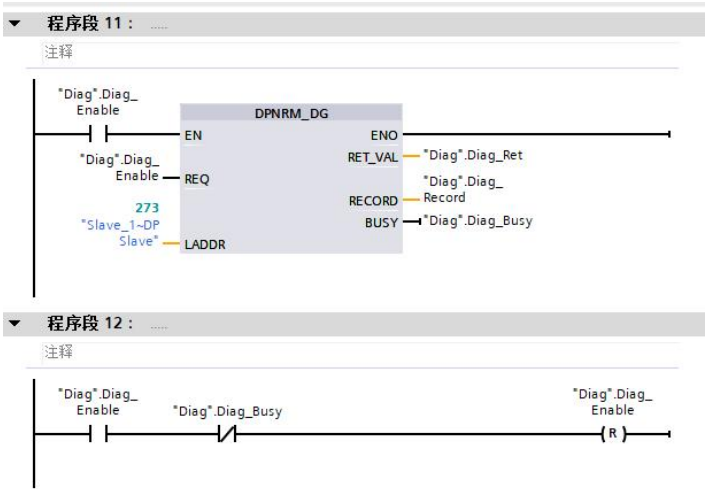
| | 名称 | 数据类型 | 起始值 | 保持 | 从 HMI/OPC... | 从 H... | 在 HMI ... | 设定值 | 注释 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Static | | | | | | | | |
| 2 | Channel_CNT | Int | 0 | | ☑ | ☑ | ☑ | ☑ | |
| 3 | Diag_Ret | Int | 0 | | ☑ | ☑ | ☑ | | |
| 4 | Diag_Record | Array[0..31] of Byte | | | ☑ | ☑ | ☑ | | |
| 5 | Diag_Busy | Bool | false | | ☑ | ☑ | ☑ | | |
| 6 | Diag_Enable | Bool | false | | ☑ | ☑ | ☑ | | |

Add the 'Diagnostic error interrupt' organization block in the program block and include the following program in the organization block. This organization block is triggered to execute once each time the PLC receives a diagnostic message from a slave station.




2. Add function block

Add the following program in the Main program block to read diagnostic information when a diagnostic message is received from a slave station.




Diagnostic information is stored in the RECORD parameter, consisting of 9 bytes, defined as follows.

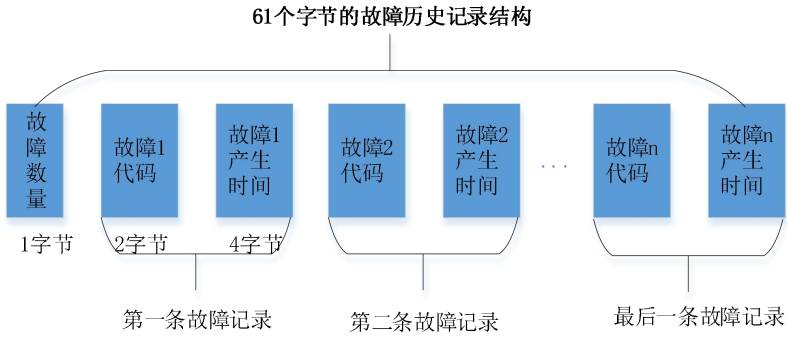
**Table 11 RECORD parameter**

| | Byte 0-Byte 2 | Byte 3 | Byte 4-Byte 5 | Byte 6 | Byte 7-Byte 8 |
|---|---|---|---|---|---|
| Definition | Slave status | Master address | Ident_Number | Device-specific diagnostic length (fixed at 3) | Device-specific diagnostic code, with the same meaning as the periodic fault code (refer to Section 5.1) |

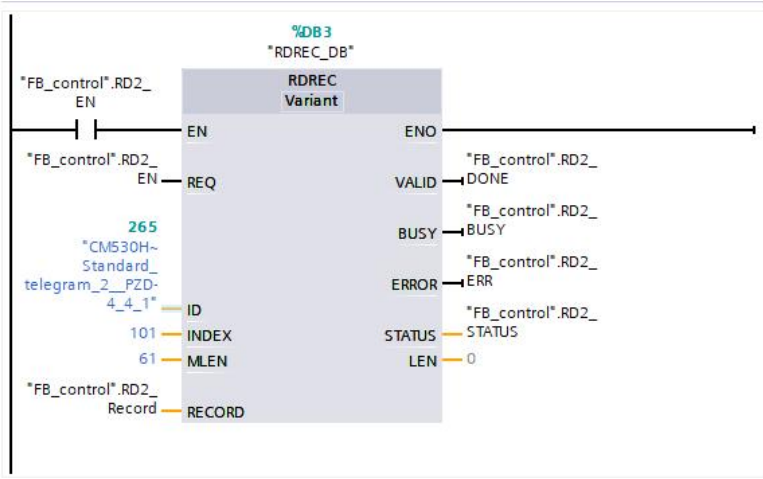
### 2.6.4. Erase fault code

1. Fault code read

Read fault codes using the RDREC function block provided by TIA Portal. The fault codes are the latest 10 fault records from the last power-up to the current read. The record format is as follows:




Example of using the non-periodic read function block: ID: the ID number of the module, INDEX: 101, MLEN: 61, RECORD: the array for storing the read fault records.




Example of parsing the Record array:

Record[0] = 0x02 indicates there are 2 fault records

Record[1] = 0x00 indicates the high byte of the first fault code is 0

Record[2] = 0x33 indicates the first fault code is 0x33, the specific meaning can be found in the table of 5.1 Periodic Fault Codes.

Record[3] = 0x01

Record[4] = 0x02

Record[5] = 0x03

Record[6] = 0x04 Record[3]~ Record[6] indicates the time of the fault occurrence is 0x01020304, which converts to decimal as 16909060, in seconds. Record[1] ~ Record[6] indicates that a fault code 0x33 occurred 16909060 seconds after startup.

Record[7] = 0x00 indicates the high byte of the second fault code is 0.

Record[8] = 0x32 indicates the second fault code is 0x32, the specific meaning can be found in Table 5.1 of Periodic Fault Codes.

Record[9] = 0x00

Record[10] = 0x00

Record[11] = 0x00

Record[12] = 0x05 Record[9]~ Record[12] indicates the time of the fault occurrence is 0x00000005, which converts to decimal as 5, with the unit being seconds

Record[13] = 0x00 Since there are only 2 fault records, all subsequent no-fault data are 0

Record[14] = 0x00

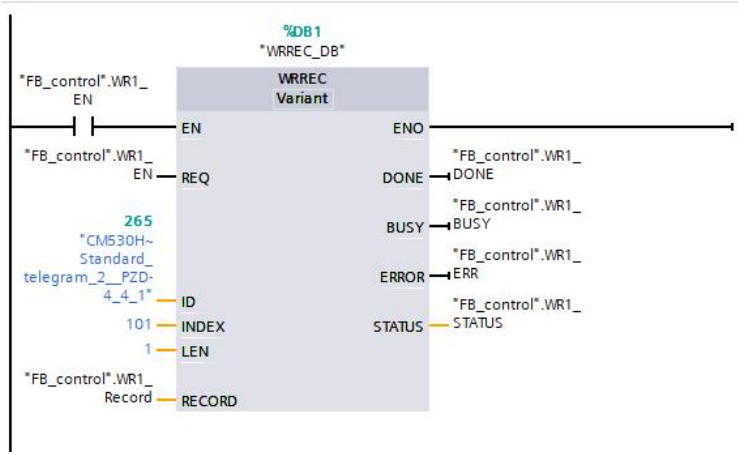
…

Record[60] =0x00

2. Fault Code Erasure

Use the WRREC function block provided by TIA Portal to erase fault codes, clearing all fault code records (TIA Portal fault records will not be erased).

Example of using the non-cyclic write function block: ID: the ID number of the module, INDEX: 101, MLEN: 1, RECORD: the data setting value for writing is 0.



# 3. PROFINET Industrial Ethernet Communication

## 3.1. Product Information

The EMH-PN Communication Expansion Card is a PROFINET fieldbus adapter card that complies with the international standard for PROFINET Ethernet. This card is installed on the CM680 Inverter to enable communication with PROFINET master station devices, making the inverter a PROFINET slave station that accepts control from the master station.

### 3.1.1. Physical Dimensions

Physical dimensions of the PROFINET module: PCB length 95mm, width 37mm, network port height 13mm, installation hole spacing 30mm, hole diameter 3.5mm.

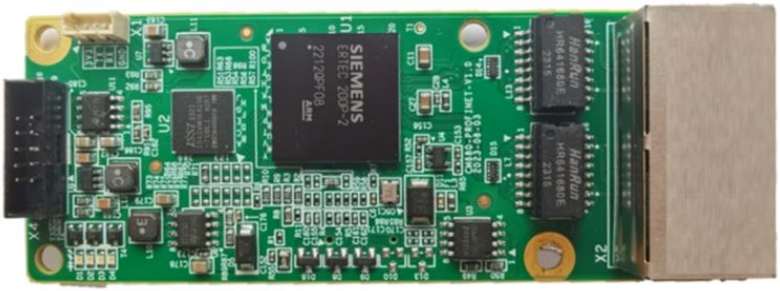



**Figure 11 Physical Dimensions of the EMH-PN Communication Expansion Card**

### 3.1.2. Interface Layout

The hardware layout of the EMH-PN Communication Expansion Card is shown in Figure 12. Pin Header X3 is used for connecting to the inverter and is located on the back of the EMH-PN Communication Expansion Card. The EMH-PN Communication Expansion Card provides two network ports X2 for communication connections. Detailed descriptions of each hardware component are provided in Table 12.

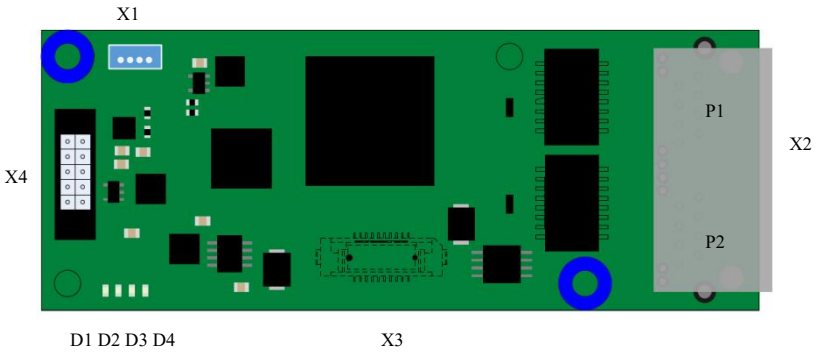



**Figure 12 EMH-PN Communication Expansion Card Interface Layout**

**Table 12 EMH-PN Communication Expansion Card Interface Description**

| Diagram Name | Hardware Name | Function Description |
|---|---|---|
| X1 | Socket | UART Interface: Print Operation Information |
| X2 | Ethernet Port | For connecting the PROFINET module to the Profinet network |
| X3 | Board-to-Board Socket | For Connection to Inverter |
| X4 | Screw Terminal | For Firmware Download and Debugging |
| D1 | PROFINET Communication Status Indicator Light | See Table 6 EMH-PN Communication Expansion Card LED Indicator Light Description |
| D2 | Inverter communication status indicator light | |
| D3 | Program run status indicator light | |
| D4 | Hardware power indicator light | |

**Table 13 EMH-PN Communication Expansion Card LED Indicator Light Description**

| Diagram Name | Indicator Light | Indicator Light Status | | Function Description |
|---|---|---|---|---|
| D1 | PROFINET Communication Status Indicator Light | Red Light | Constantly On | PROFINET Communication Abnormal |
| | | | Off | PROFINET Communication Normal |
| D2 | Inverter communication status indicator light | Red Light | Constantly On | Communication timeout with the inverter |
| | | | Flashing | Inverter Communication Message Abnormal |
| | | | Off | Normal communication with the inverter |
| D3 | Program run status indicator light | Green light | Constantly On | Main program running normally |
| | | | Off | Main Program Operation Abnormal |
| D4 | Hardware power indicator light | Green light | Constantly On | Module powered on normally |
| | | | Off | Module not powered on |

## 3.2. Installation and Wiring

### 3.2.1. Installation

The EMH-PN Communication Expansion Card is designed for internal installation in the CM680 Inverter. Before installation, please disconnect the power supply to the inverter and wait for about 10 minutes until the charging indicator light is completely off before proceeding with the installation. After inserting the EMH-PN Communication Expansion Card into the inverter, please secure the corresponding screws to prevent poor contact of the board-to-board signal sockets. The installation diagram is shown in Figure 11.

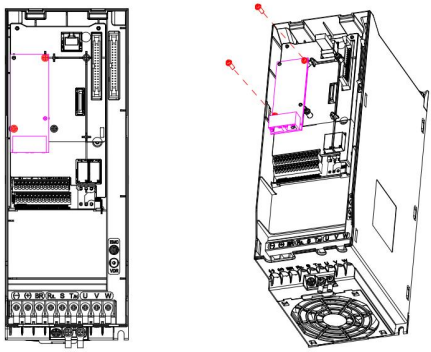


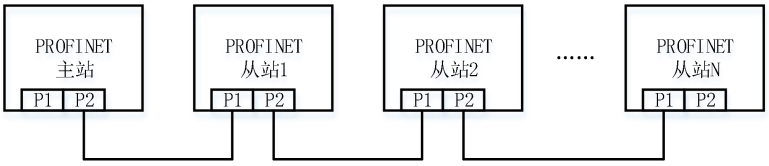
**Figure 13 EMH-PN Communication Expansion Card Installation Diagram**

### 3.2.2. Wiring

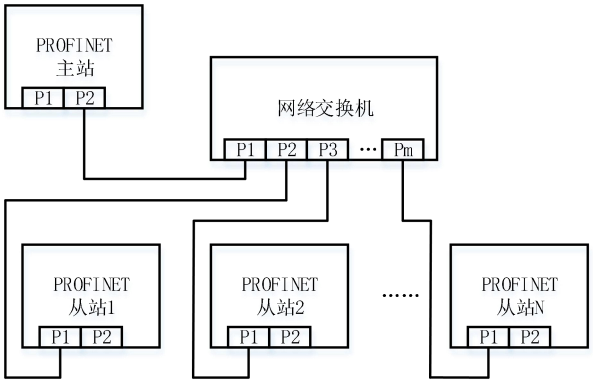
Insert an Ethernet cable into the network port of the EMH-PN Communication Expansion Card, then connect it to the PLC. You can connect to either the P1 port or the P2 port according to the network topology in TIA Portal. If the network topology is not specified, any network port P1 or P2 can be connected.

It is recommended to use a Cat 5e shielded network cable.

Common network topology diagrams are as follows:




Linear Topology




Star Topology

**Figure 14 PROFINET Network Topology**

## 3.3. Communication Description

### 3.3.1. Key Points of PROFINET Communication

- When using the EMH-PN Communication Expansion Card, the communication decoding method of the CM680 needs to be set. Set the F8-19 register to 0 via the operation panel, and set the F8-07 register to 0 or 1 based on the actual decoding method used.
- When setting the inverter command, the F0-05 register of the CM680 needs to be set to 5.
- When setting the inverter frequency, the F0-06 register of the CM680 needs to be set to 8.
- During non-periodic communication, a new non-periodic communication must only be initiated after the completion of the previous non-periodic read/write. The interval between two non-periodic communications should be no less than 100ms to avoid affecting the response to periodic communication.

### 3.3.2. Data Transmission Format

The EMH-PN Communication Expansion Card can select different PZD lengths for transmission based on requirements. Users can configure the functions of each PZD in the configuration. The supported functions for each data format are listed in the table below.

**Table 14 Supported Functions for Each Data Format**

| Data Type | Data Length | Supported Functions |
|---|---|---|
| Standard Telegram 1 | PZD-2/3 | 2 periodic writes, default 0x6000 control command, 0x6002 frequency setting<br>3 periodic reads, |
| Standard Telegram 2 | PZD-16/16 | 16 periodic writes of function parameters<br>Default 0x6000 control command, 0x6002 frequency setting, and other modifiable reserved parameters 0x0830<br>16 functional parameters read cyclically<br>Fixed 0x2100 errors and warnings, default 0x6100 inverter status, 0x6102 operating frequency read, and other modifiable reserved parameters 0x0830 |

### 3.3.3. PZD area data description

Through PZD area data, the master station can modify and read the inverter data in real time, and perform cyclic data exchange. The communication address is configured by the user during configuration. The specific functions are as follows:

- Real-time setting of inverter control command and target frequency
- Real-time reading of inverter current status, operating frequency, and errors and warnings
- Real-time data exchange of functional parameters and monitoring parameters between the inverter and the Profinet master station

PZD process data mainly completes periodic data exchange between the master station and the inverter, see the table below.

**Table 15 Exchange Data**

| | Master station sends data to PZD area | | |
|---|---|---|---|
| Inverter control command | Inverter target frequency | Real-time modification of inverter function parameters | |
| PZD1 | PZD2 | PZD3~PZD16 | |
| | Inverter response data PZD area | | |
| Errors and Warnings | Inverter operating status | Inverter operating frequency | Real-time reading of inverter function parameters |
| PZD1 | PZD2 | PZD3 | PZD4~PZD16 |

### 3.3.4. Description of data sent by the master station

**Table 16 Description of data sent by the master station to PZD**

| Master Station Data PZD Description | |
|---|---|
| PZD1 Control Cmd<br>(Master to Slave) | Inverter Command Word (Command Source Needs to be Set to Communication F0-05 =5) |
| | Bit0: CMD_ACT Bit1: EXT_CMD1 Bit2: EXT_CMD2 Bit3: HALT<br>Bit4: LOCK Bit5: JOG Bit6: QSTOP Bit7: SERVO_ON<br>Bit8~Bit11: GEAR Bit12~Bit13: ACC/DEC Bit14: EN_SW Bit15: RST<br>Modifiable by User. |
| PZD2 Velocity Cmd<br>(Master to Slave) | Inverter Target Frequency Command Word (Main Frequency Source Needs to be Set to Communication F0-06 =8), Modifiable by User. |
| PZD3-PZD16 (Master to Slave) | Default is Address 2096 (Hexadecimal 0x0830), Modifiable by User. Users Can Configure Inverter Function Code Address During Configuration. |

### 3.3.5. Inverter Response Data Description

**Table 17 Inverter Response Data Description**

| Inverter Response Data Description | |
|---|---|
| PZD1 fault code<br>(Slave to Master) | Current error and warning information of the inverter, detailed definitions refer to Section 5.1. |

| | |
|---|---|
| PZD2 Status<br>(Slave to Master) | Bit0: ARRIVE Bit1: DIR Bit2: WARN Bit3: ERROR<br>Bit4: Reserve Bit5: JOG Bit6: QSTOP Bit7: SERVO_ON<br>Bit8: Ready Bit9~Bit15: Reserve<br>Modifiable by User. |
| PZD3<br>Current Velocity<br>(Slave to Master) | Current operating frequency of the inverter (unit: 0.01Hz). User modifiable. |
| PZD4-PZD16<br>(Slave to Master) | Default is Address 2096 (Hexadecimal 0x0830), Modifiable by User. Users Can Configure Inverter Function Code Address During Configuration. |

## 3.4. Communication Configuration

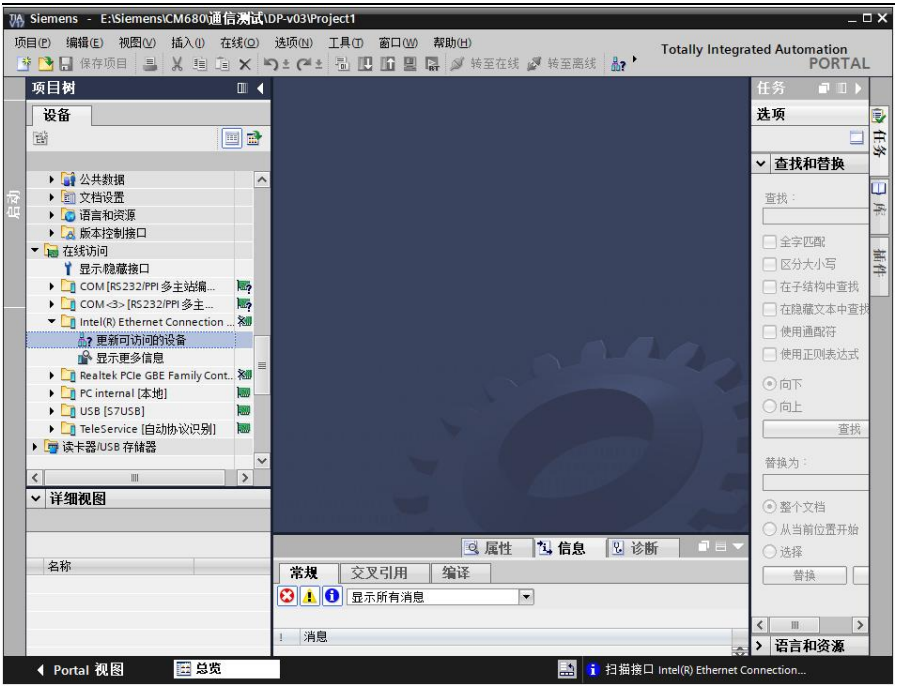
Taking the Siemens S7-1500 series PLC as an example, configure the PLC and module to establish Profinet communication.

### 3.4.1. Configure Configuration

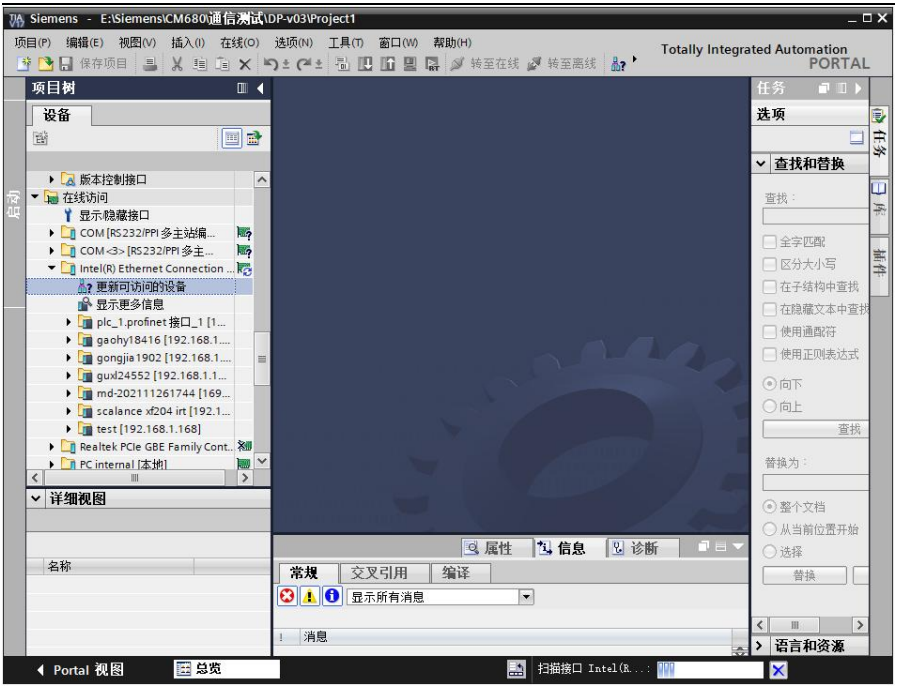
1. Create a project

Double-click to open the TIA Portal software, select 'Create New Project', and the following interface will appear. Choose the save path and project name, then click 'Create'.

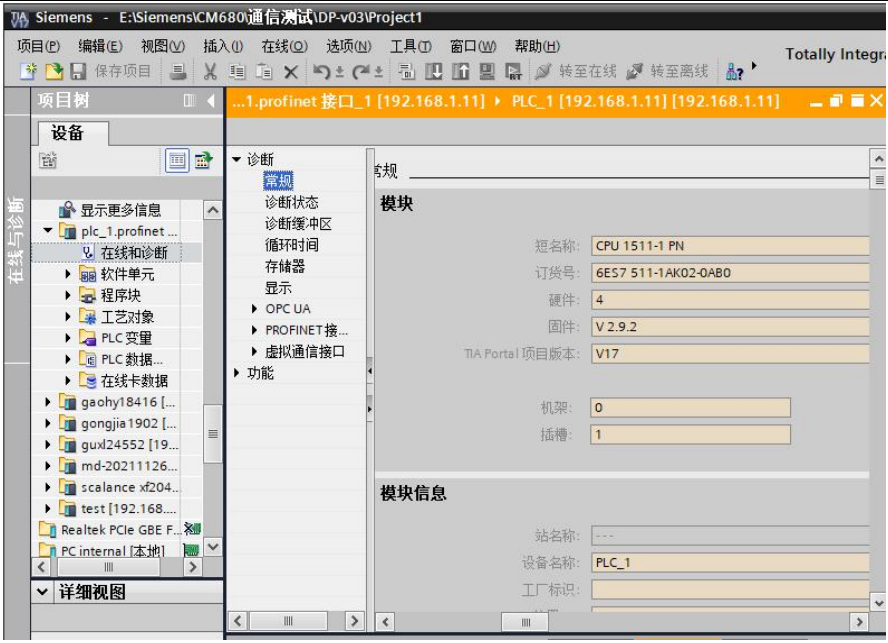

After creation, click 'Project View' to enter the main interface. Connect the PLC and PROFINET module to your computer with an Ethernet cable and power them on. In the TIA Portal project tree, under the Online Access menu, click Update Accessible Devices.



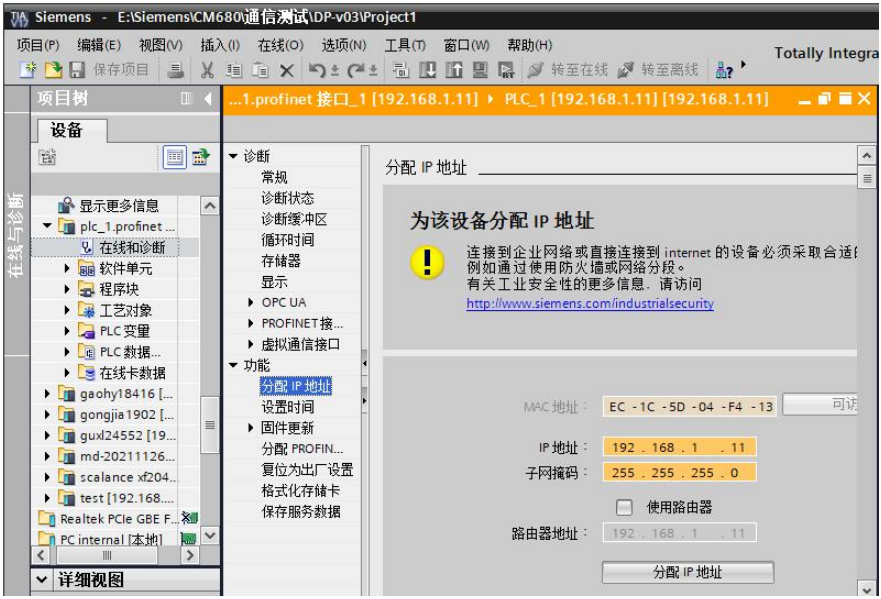

List the devices connected to the computer network.



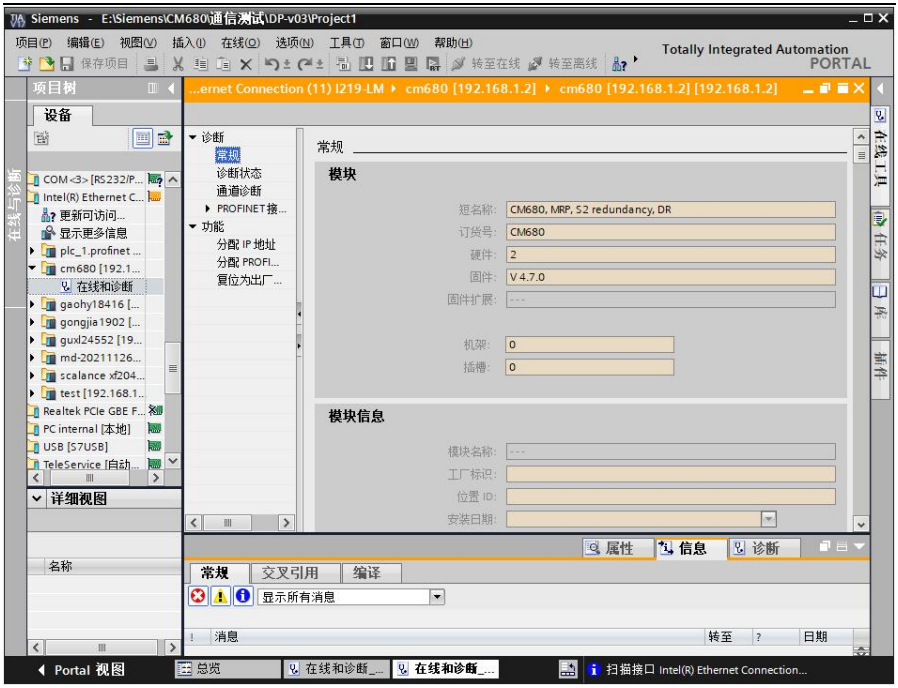

Click on the online and diagnostics of the PLC device to view its model and other information.



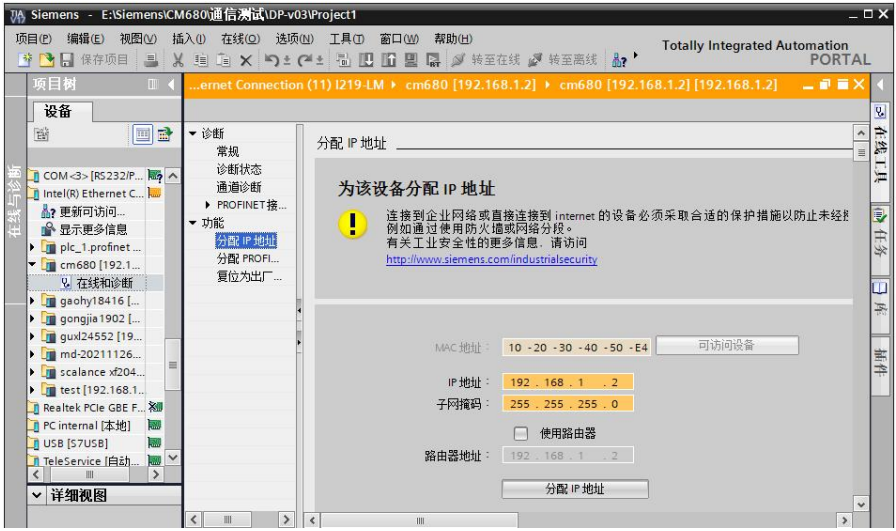

Open the 'Function' tree to modify the PLC's IP address, device name, and other configurations.



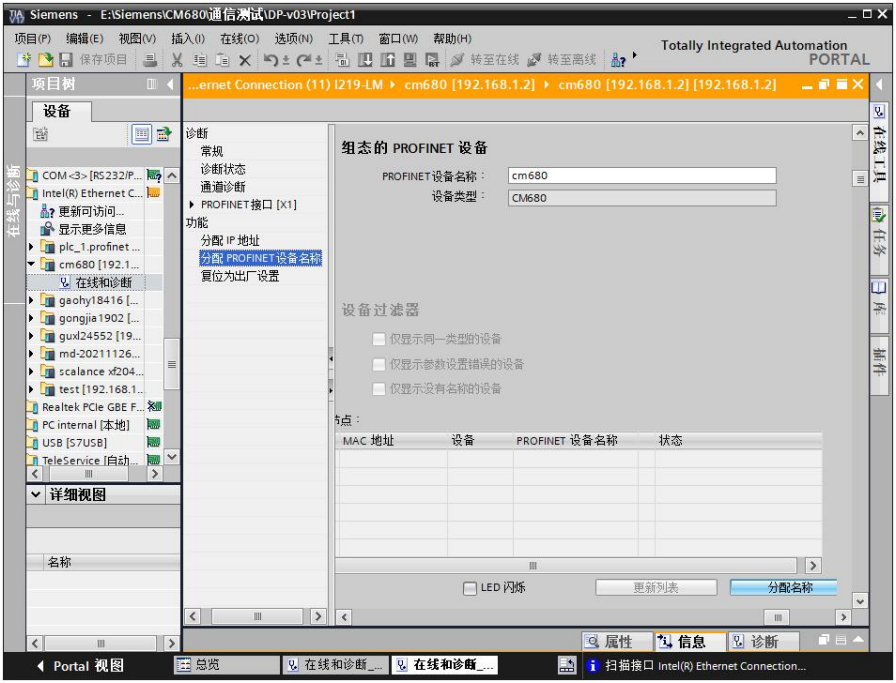

Click on the online and diagnostics of the module to view module information.



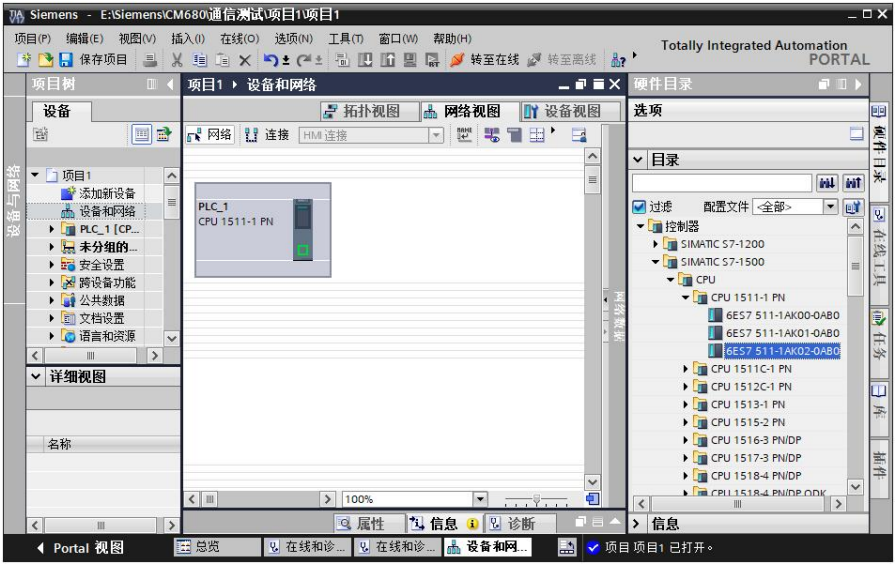

In the function menu, you can modify the device name and IP address. The IP address should be in the same subnet as the PLC's IP address. Click the [Assign IP Address] button to save the modified settings.



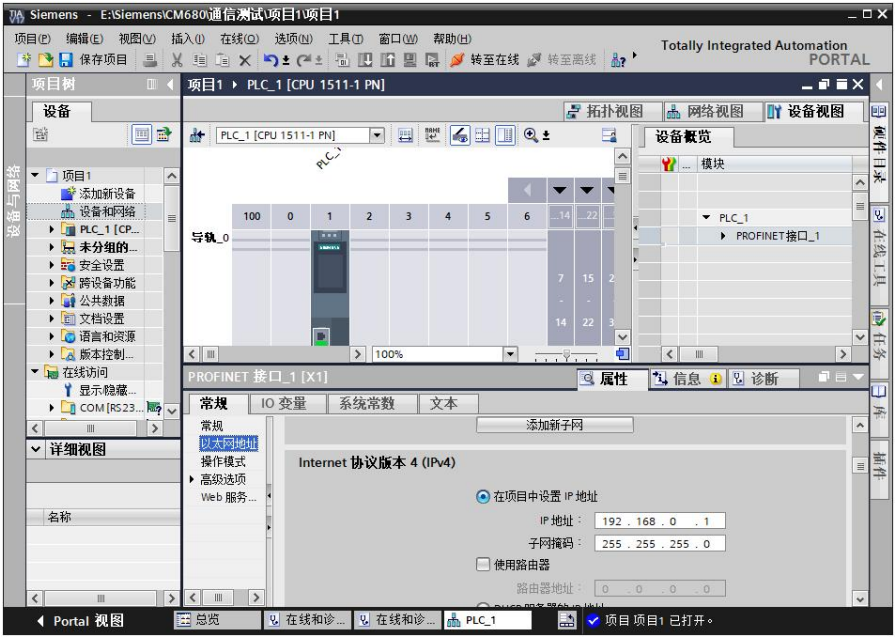
On the 'Assign PROFINET Device Name' page, modify the PROFINET device name. After making changes, click the [Assign Name] button to save the modified settings.



In the 'Hardware Catalog' on the right, select the actual PLC model being used; for example, S7-1511. Double-clicking will display the PLC on the configuration page.



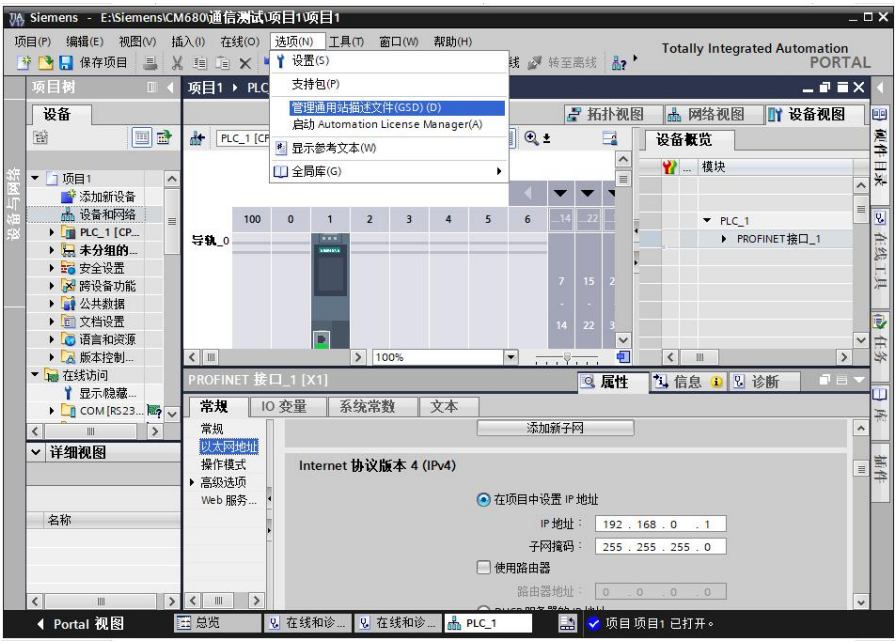

Click on the PLC, in the property bar, you can configure information such as the PLC's IP address, which can be modified according to the actual situation.



This completes the establishment of the master station.

2. Install GSDML file

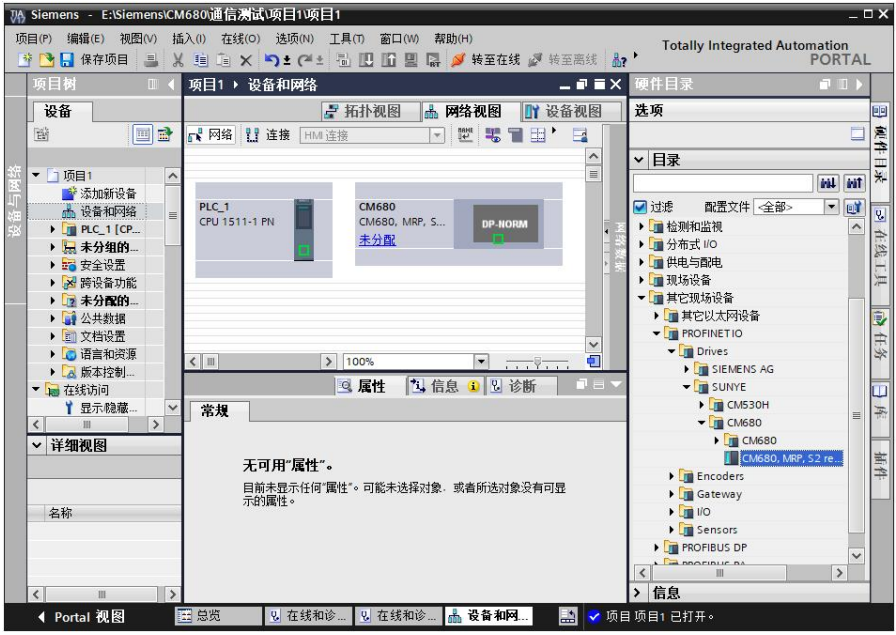
If the GSDML has not been installed, it needs to be installed. In 'Options', select 'Manage General Station Description (GSD) Files'.



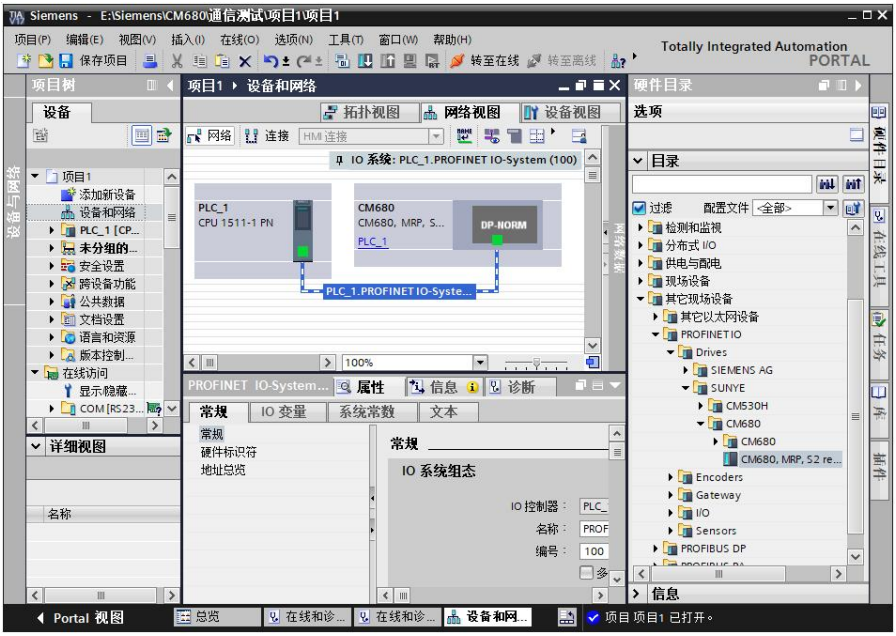

Select the path where the GSDML is stored (Note: Do not store the GSDML file in a path with Chinese characters, as this may cause errors), check the GSDML to be installed, and click 'Install'. After installation, select 'Close'.


3. Configure Slave Station

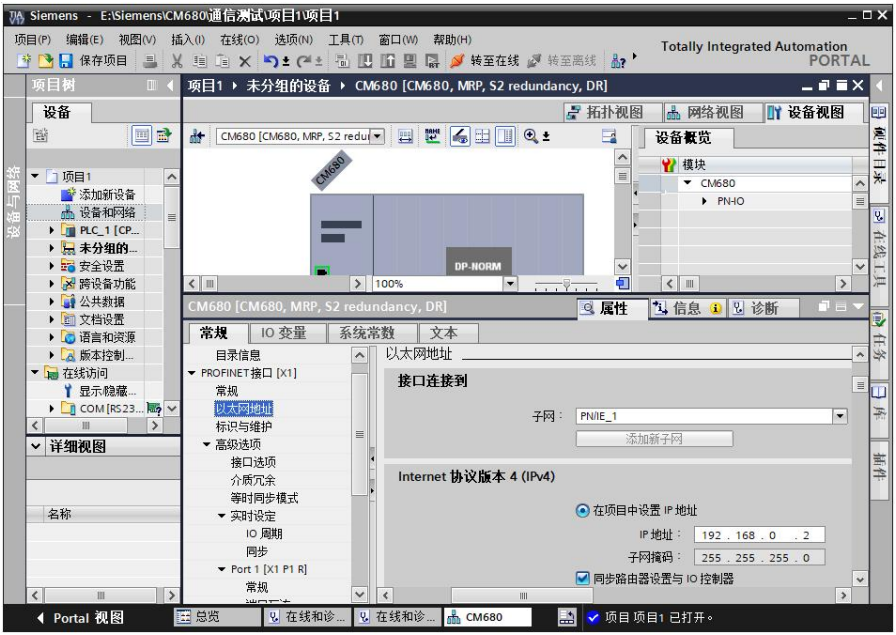
Return to the 'Device and Network' — 'Network View' interface, in the 'Hardware Catalog' on the right, find the CM680 position under Other Field Devices — PROFINET IO — Drives — SUNYE, and double-click to add the module.



Click on 'Unassigned' on the module in the 'Device and Network' interface to assign the master station system to the module.



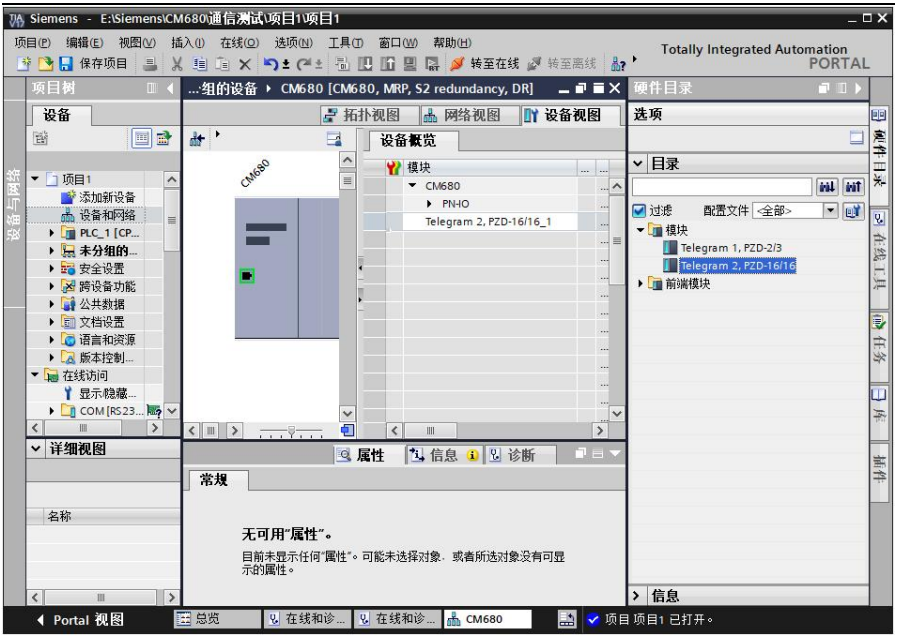
Select the slave station, and set the IP address and device name in 'Properties' -> 'General' -> 'PROFINET Interface [X1]' -> 'Ethernet Address'. The device name must match the device name set in the online access.



### 3.4.2. Configure Periodic Communication

1. Configure Module

Select the slave station, switch to the “Device View,” and under “Hardware Catalog” -> “Module,” double-click to select the data length to be configured for the slave station. The configured module will be added in the device view. After adding the module, you can modify the input/output mapping in the PLC by editing the “I Address” and “Q Address”.




2. Configure PZD

Through PZD area data, the master station can real-time modify and read inverter data, and perform periodic data exchange.

PZD process data mainly completes periodic data exchange between the master station and the inverter, the exchanged data is as follows:

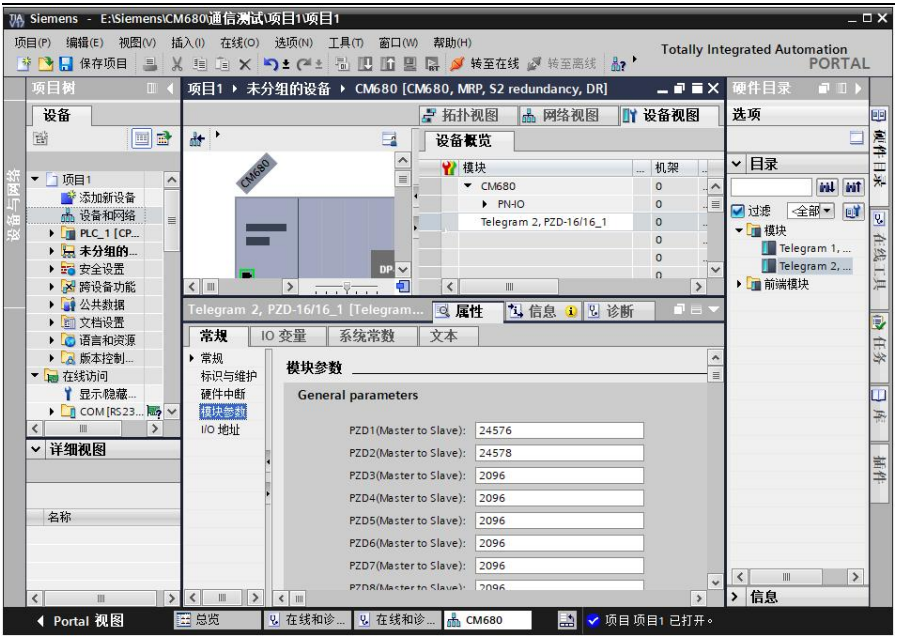
**Table 18 PZD Process Data**

| | | | |
|---|---|---|---|
| | Master station sends data to PZD area | | |
| Inverter Command | Inverter target frequency | Real-time modification of inverter function parameters | |
| PZD1 | PZD2 | PZD3~PZD16 | |
| | Inverter response data PZD area | | |
| Errors and Warnings | Inverter Status | Inverter operating frequency | Real-time reading of inverter function parameters |
| PZD1 | PZD2 | PZD3 | PZD4~PZD16 |

For specific bit definitions, refer to sections 3.4 and 3.5.

PZD3 to PZD12 are user-defined periodic data exchanges, which are set in the hardware configuration.

Select the module added in the previous step, and under 'Properties' -> 'General', choose 'Module Parameters' to set the register address for user-defined periodic data. PZDx(Master to Slave) indicates the address written by the master station to the slave station, and PZDx(Slave to Master) indicates the address read by the master station from the slave station, in decimal format.




After configuration is complete, download the configuration, and the PLC and module will automatically perform periodic data exchange.

### 3.4.3. Configure Non-periodic Communication

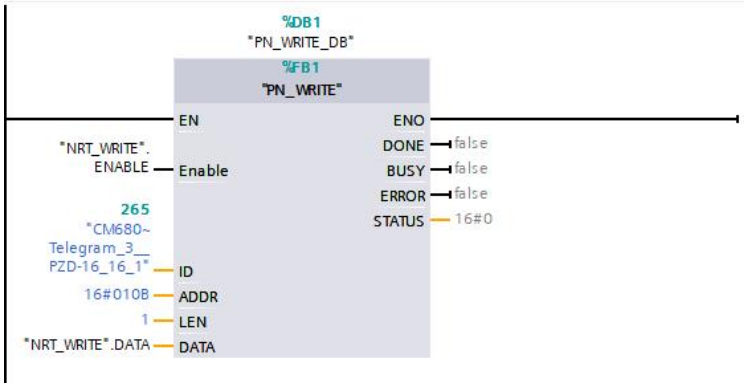
1. Non-periodic Interaction Data Format

Non-periodic interaction uses the slot-subslot-index method for addressing, thereby determining the format of the message data area. To facilitate user operations on consecutive registers of the inverter, the following non-periodic read/write format is defined, with the index set to 100.

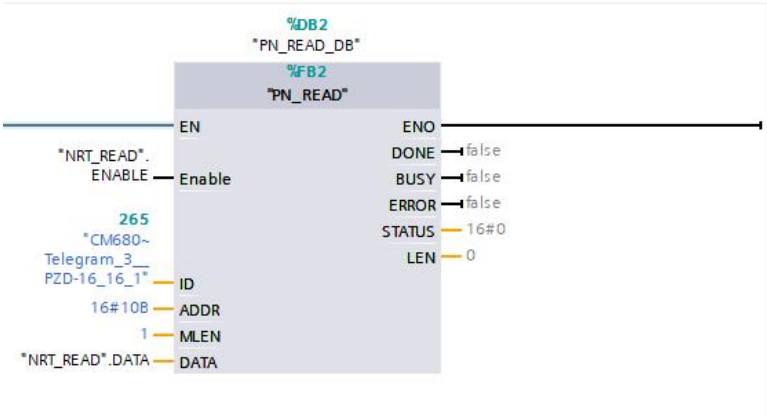
The manufacturer has encapsulated two non-periodic read/write function blocks for user reference.

PN_WRITE function block: implements non-periodic writing, ID: represents the moduleID to be written, ADDR: represents the address of the function code to be written,

LEN: represents the number of registers to be written, DATA: array of values to be written into the registers.

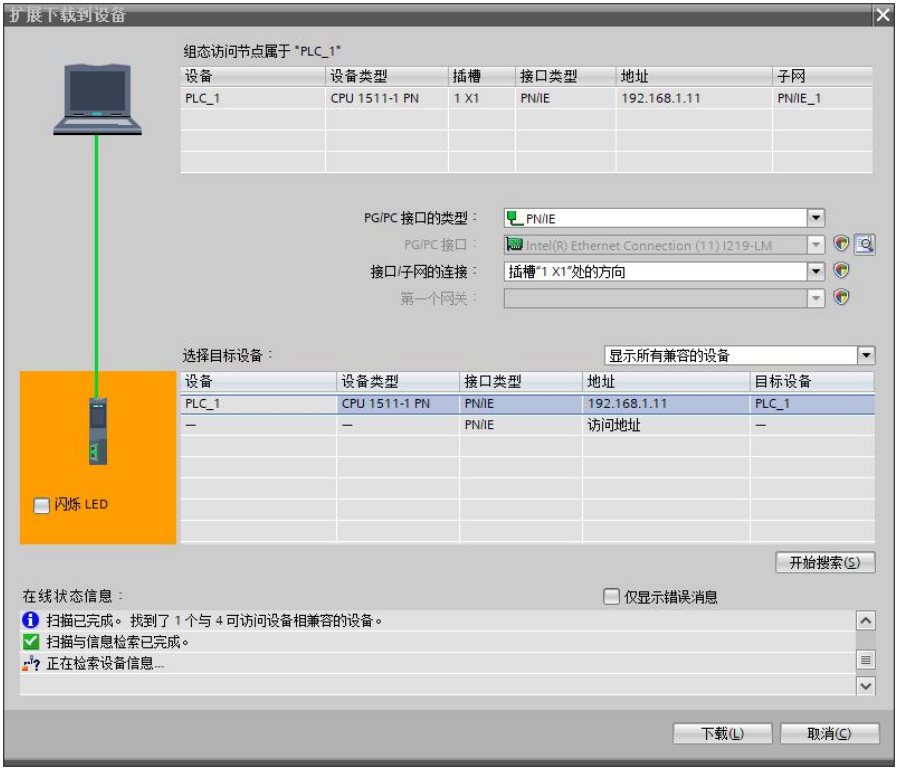



PN_READ function block: a function block for non-periodic reading, ID: represents the moduleID to be written, ADDR: represents the address of the function code to be read, MLEN: represents the number of registers to be read, DATA: array to store the values read from the registers.



2. Download Configuration and Ladder Diagram

Save the configured network, set the computer's IP address to the same subnet as the PLC (note that it should not conflict with the IP addresses of the slave stations in the configuration, or you can set the PC to automatically obtain an IP address), compile, click download, select the interface, and then click 'Start Search'. Select the detected PLC, and then click download.




## 3.5. Fault Information

### 3.5.1. Periodic Fault Code

In the PZD Area for periodic communication, the high byte of the status word indicates alarm information, allowing users to obtain the latest status through periodic communication. In addition to the diagnostic information of the inverter itself, the communication status between the module and the inverter must also be included to help users identify the cause of errors.

**Table 19 Periodic Fault Code**

| Diagnostic Classification | Status Code |
|---|---|
| Inverter Body Diagnostic | 200 Communication Error (Message Abnormality)<br>201 Communication Error (Timeout Abnormality) |

| | |
|---|---|
| Information (Register Address: 0x2100) | 202 Address Error<br>203 Parameter Exceeds Threshold |

### 3.5.2. Non-periodic Fault Code

When executing non-periodic communication, the STATUS parameter in the function block is used to return the command execution result, with a data type of DWORD (4 bytes). The data format and meaning are already defined in the TIA software. In addition to the standard definition, the following error codes have the following meanings:

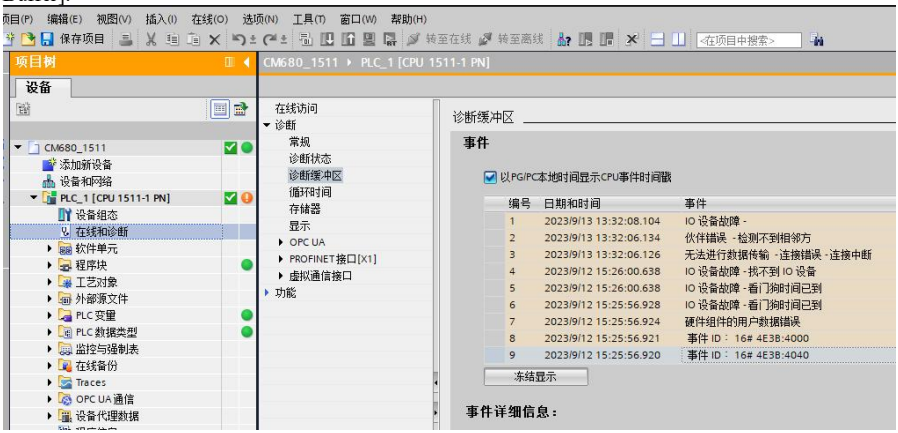
**Table 20 Non-periodic Fault Code**

| Function_ Num | Error_ Decode | Error_ Code_1 | Error_ Code_2 | Meaning |
|---|---|---|---|---|
| 0xDE | 0x80 | 0xB8 | 0x00 | Invalid data record number (index can only be 100 or 101) |
| 0xDE | 0x80 | 0xB8 | 0x02 | Non-periodic read value data length exceeds limit (2<=data length<=24), unit: bytes |
| 0xDE | 0x80 | 0xB8 | 0x03 | Non-periodic write read address is invalid (WRREC has not been run in advance to write the read address or the address is incorrect) |
| 0xDE | 0x80 | 0xB8 | 0x05 | Non-periodic read value failed (communication error between module and inverter) |
| 0xDF | 0x80 | 0xB8 | 0x00 | Invalid data record number (index can only be 100 or 101) |
| 0xDF | 0x80 | 0xB8 | 0x01 | Non-periodic write value function code error (only supports 0x03, 0x10 function codes) |
| 0xDF | 0x80 | 0xB8 | 0x02 | Non-periodic Write Value Data Length Exceeded (3<=Data Length<=24), Unit: Bytes |
| 0xDF | 0x80 | 0xB8 | 0x03 | Non-periodic Write Address Invalid |
| 0xDF | 0x80 | 0xB8 | 0x04 | Non-periodic Write Set Value Out of Range |
| 0xDF | 0x80 | 0xB8 | 0x05 | Non-periodic Write Value Failure (Module and Inverter Communication Error) |

### 3.5.3. Module Information

When communication between the module and the inverter is abnormal, diagnostic information will be sent to the PLC. This can be viewed in the TIA Portal software under PLC's [Online & Diagnostics] — [Diagnostics] — [Diagnostic

Buffer].




### 3.5.4. Diagnostic Buffer Fault Meaning and Handling

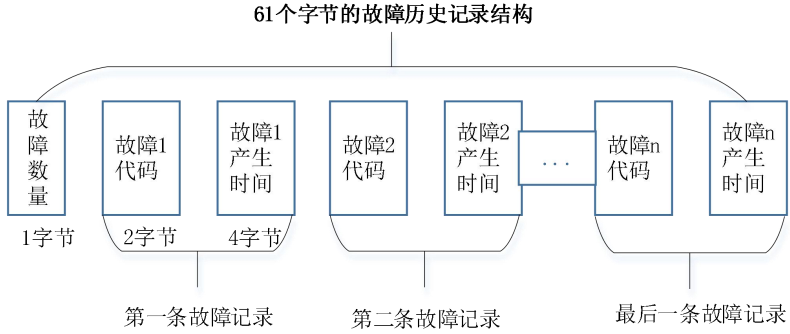
**Table 21 Fault Meaning and Handling**

| Diagnostic Buffer Events | Error Meaning | Correction Handling |
|---|---|---|
| Address Error | Periodic Address or Non-periodic Address Error | Periodic Communication PZD Address Contains Invalid Value, Refer to Function Code Table, Correct the Error Address. |
| Slave status error | Inverter status error | Check the high byte error code definition of PZD1 and take corresponding actions |
| Communication error | Communication timeout with the inverter | If the error cannot be eliminated, restart the inverter |
| Parameters error | Periodic parameters error | Modify the periodic parameters to make them comply with the set range |

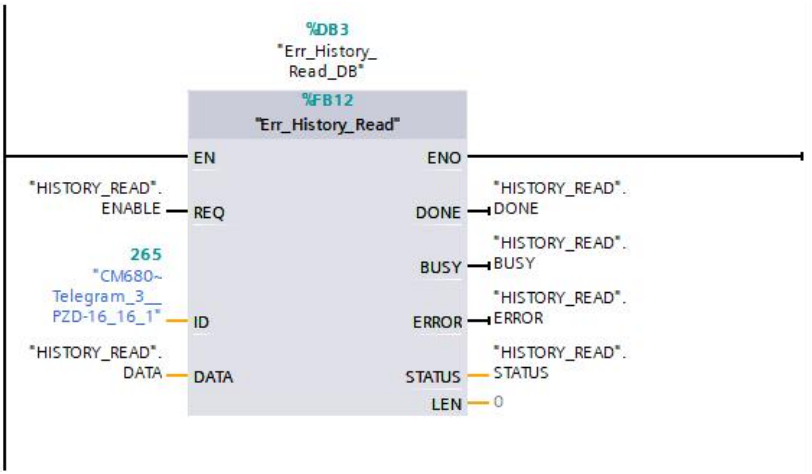
### 3.5.5. Non-periodic Read or Erase Fault Code

Fault code reading

Read fault codes using the function block Err_History_Read provided by the manufacturer. The fault codes are the latest 10 fault records from the module's power-on to the current read. The record format is as follows:




Example of using the non-periodic read function block: ID: the ID number of the module, DATA: the array for storing the read fault records.




Example of parsing the DATA array:

DATA[0] = 0x02 indicates that there are 2 fault records

DATA[1] = 0x00 indicates the high byte of the first fault code is 0

DATA[2] = 0x33 indicates the first fault code is 0x33, the specific meaning can be found in Table 5.1 Periodic Fault Code.

DATA[3] = 0x01

DATA[4] = 0x02

DATA[5] = 0x03

DATA[6] = 0x04 DATA[3]~ DATA[6] indicate the time of the fault occurrence is 0x01020304, which converts to decimal as 16909060, in seconds. DATA[1] ~ DATA[6] indicate that the fault code 0x33 occurred 16909060 seconds after startup.

DATA[7] = 0x00 indicates the high byte of the second fault code is 0

DATA[8] = 0x32 indicates the second fault code is 0x32, the specific meaning can be found in the table of periodic fault codes in section 5.1.

DATA[9] = 0x00

DATA[10] = 0x00

DATA[11] = 0x00

DATA[12] = 0x05 DATA[9]~ DATA[12] indicate the time of the fault occurrence is 0x00000005, which converts to decimal as 5, with the unit being seconds

DATA[13] = 0x00 Since there are only 2 fault codes, the subsequent no-fault data are all 0

DATA[14] = 0x00

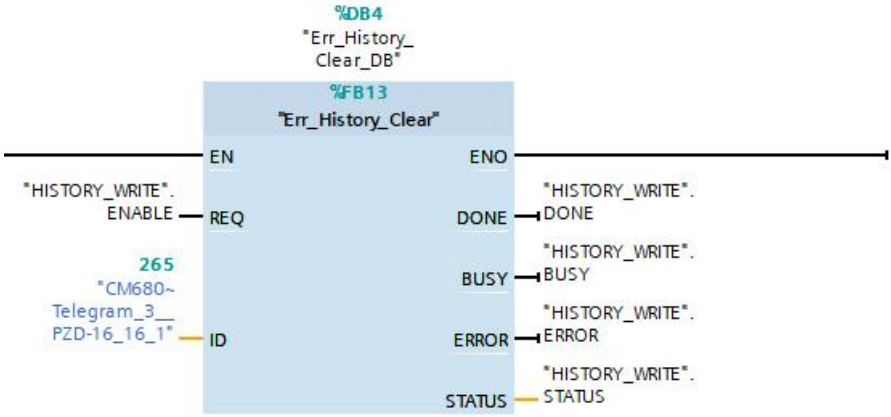
…

DATA[60] =0x00

Fault Code Erase

Use the function block Err_History_Clear provided by the manufacturer to erase fault codes, which will clear all fault code records (will not erase the fault records in TIA Portal).

Example of using the non-cyclic write function block: ID: the ID number of the module.



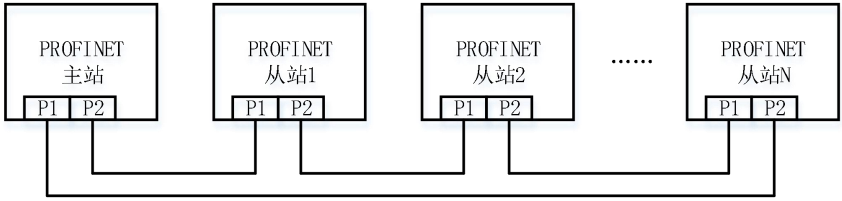

### 3.6. MRP function description

MRP refers to media redundancy (Media Redundancy Protocol). In PROFINET, this function is implemented through an MRP ring network, with only one MRP ring network allowed in a PROFINET network. Configuration is required during setup when using MRP.

#### 3.6.1. MRP ring network

First, an MRP ring network requires an MRP manager, typically a PLC, while PROFINET modules can only act as MRP clients.

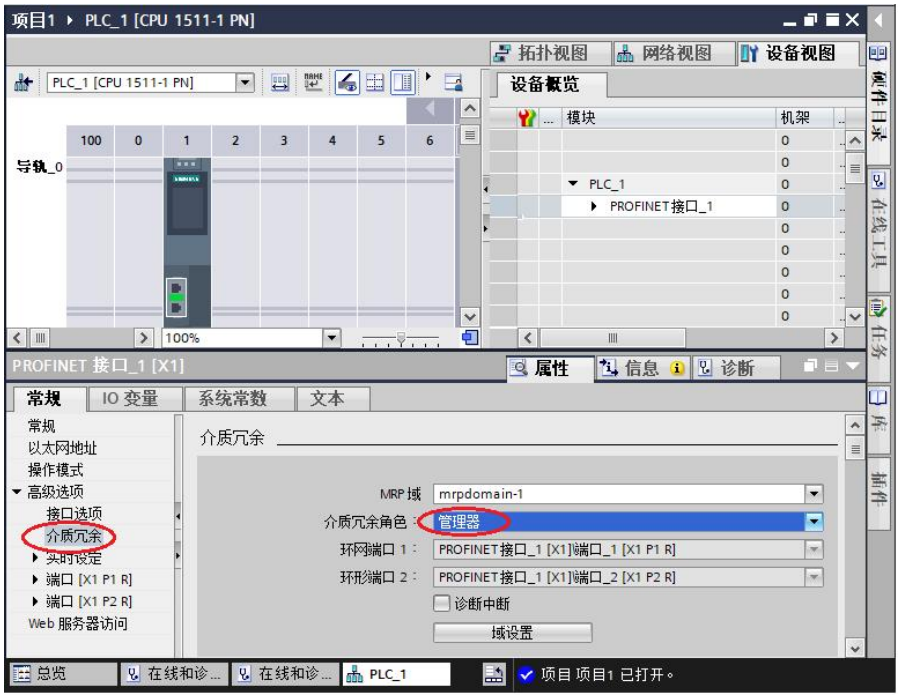
Example of ring topology:




### 3.6.2. MRP configuration

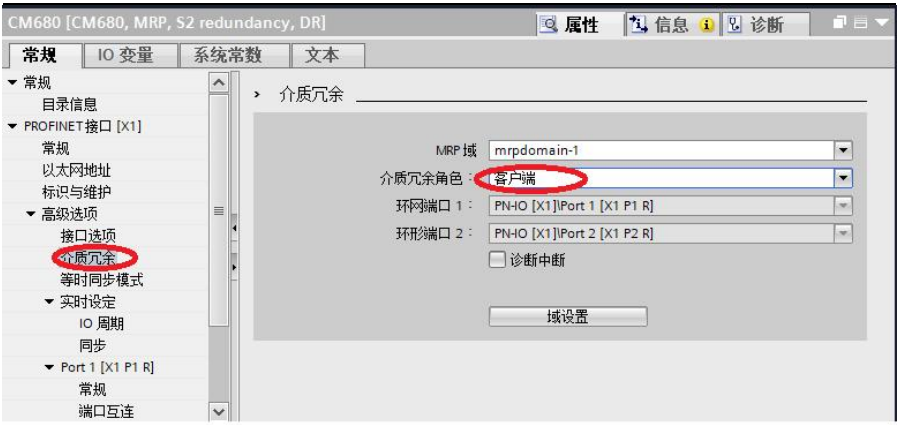
Step one: configure the MRP manager

An MRP manager must exist in the MRP ring network. The EMH-PN card cannot serve as a manager; typically, a PLC is used as the manager. Select the site to be configured as the manager, and in the interface shown below, select the manager from the 'Media Redundancy Function' drop-down menu.




Step Two: Configure MRP client

Select the slave station, in the interface shown below, in the 'Media Redundancy Function' drop-down select 'Client'. Before configuring the client, the manager must be configured first, otherwise, an error will occur.




Step Three: Download Configuration

After configuring all devices in the MRP ring network, compile and download to the PLC.

Note:

◆ All devices in the ring network must be configured as MRP manager or MRP client;

◆ MRP configuration does not require setting up a topology diagram; if a topology diagram is needed, it should be set up after completing the MRP configuration;

◆ Devices not configured for MRP should not use the ring network, as this can lead to connection failures or repeated station drops.

# 4. EtherNet/IP Industrial Ethernet Communication

## 4.1. Product Information

The EMH-EN Communication Expansion Card is an EtherNet/IP fieldbus adapter card that complies with the general EtherNet/IP fieldbus standard. This card is installed on the CM680 Inverter to enable communication with EtherNet/IP master station devices, making the inverter a slave station of EtherNet/IP, accepting control from the master station.

### 4.1.1. Physical Dimensions

EtherNet/IP Module Physical Dimensions: PCB length 95mm, width 37mm, mounting hole diameter 3.5mm.

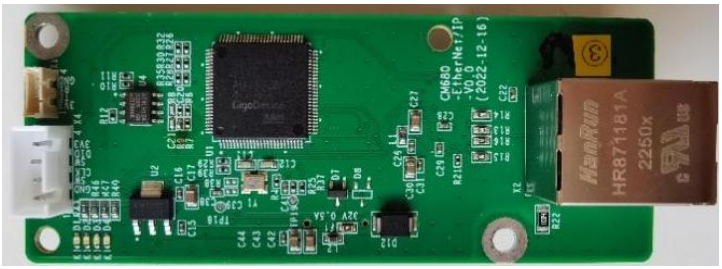


**Figure 15 Physical Dimensions of the EMH-EN Communication Expansion Card**

### 4.1.2. Interface Layout

The hardware layout of the EMH-EN Communication Expansion Card is shown in Figure 16. Pin Header X3 is used for connecting to the inverter and is located on the back of the EMH-EN Communication Expansion Card. The EMH-EN Communication Expansion Card provides an Ethernet connector for connecting to EtherNet/IP communication. For detailed descriptions of the hardware, refer to Table 22.

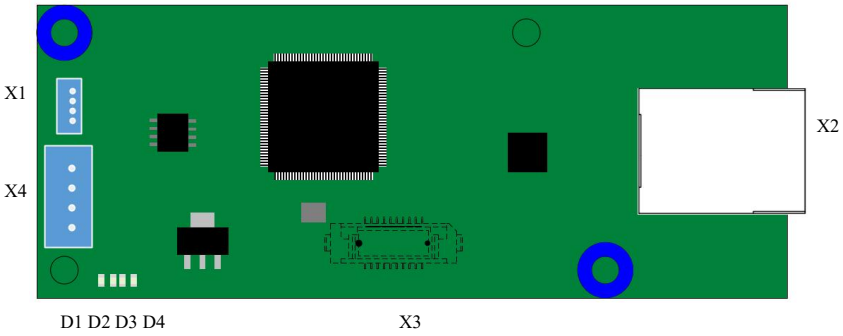



**Figure 16 EMH-EN Communication Expansion Card Interface Layout**

**Table 22 EMH-EN Communication Expansion Card Interface Description**

| Diagram Name | Hardware Name | Function Description |
|---|---|---|
| X1 | Socket | UART Interface: Print Operation Information |
| X2 | Ethernet Socket | For Module Connection to EtherNet/IP Network |
| X3 | Board-to-Board Socket | For Connection to Inverter |
| X4 | Socket | For Firmware Download and Debugging |
| D1, D2, D3, D4 | LED Indicator Light | Used to indicate the operating status, refer to Table 1.2 for details |

LED Indicator Light

**Table 23 EMH-EN Communication Expansion Card LED Indicator Light Description**

| Diagram Name | Indicator Light | Indicator Light Status | | Function Description |
|---|---|---|---|---|
| D1 | EtherNet/IP Communication Status Indicator Light | Red Light | Constantly On | Indicates No Communication Between the Module and EtherNet/IP Master Station (Check EtherNet/IP Cable Connection and Station Number) |
| | | | Off | Indicates Normal Communication Between the Module and EtherNet/IP Master Station |
| D2 | Inverter communication status indicator light | Red Light | Constantly On | Communication timeout with the inverter |
| | | | Flashing | Inverter Communication Message Abnormal |
| | | | Off | Normal communication with the inverter |
| D3 | Program run status indicator light | Green light | Constantly On | Main program running normally |
| | | | Off | Main program not started properly |
| D4 | Hardware power indicator light | Green light | Constantly On | Module powered on normally |
| | | | Off | Module not powered on |

## 4.2. Installation and Wiring

### 4.2.1. Installation

The EMH-EN Communication Expansion Card is designed for internal installation in the CM680 Inverter. Before installation, please disconnect the power supply to the inverter and wait approximately 10 minutes until the inverter charging indicator light is completely off before proceeding with the installation. After inserting the EMH-EN Communication Expansion Card into the inverter, please secure the corresponding screws to prevent poor contact between the board signal sockets. The installation diagram is shown in Figure 14.

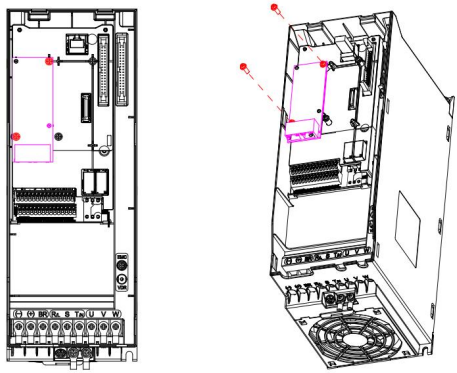


**Figure 17 EMH-EN Communication Expansion Card Installation Diagram**

#### 4.2.2. Wiring

The wiring diagram for the EtherNet/IP bus is shown in Figure 15. Insert a network cable into the network port of the EMH-EN Communication Expansion Card, then connect it directly to the PLC, or indirectly through a switch. It is recommended to use a Cat 5e shielded network cable.

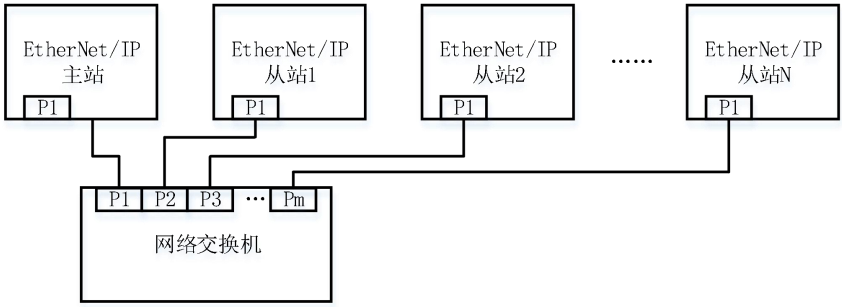



**Figure 18 EMH-EN Bus Wiring Diagram**

## 4.3. Communication Description

#### 4.3.1. Key Points for EtherNet/IP Communication

- When using the EMH-EN Communication Expansion Card, you need to set the communication decoding method for the CM680. Set the F8-19 register to 0 via the operation panel, and set the F8-07 register to 0 or 1 based on the actual decoding method used.
- When setting the inverter command, the F0-05 register of the CM680 needs to be set to 5.
- When setting the inverter frequency, the F0-06 register of the CM680 needs to be set to 8.
- The IP address for EtherNet/IP communication of the EMH-EN Communication Expansion Card is set in registers F8-27 to F8-38. The module must be restarted for the settings to take effect.
- During non-periodic communication, a new non-periodic communication must only be initiated after the completion of the previous non-periodic read/write. The interval between two non-periodic communications should be no less than 100ms to avoid affecting the response to periodic communication.

### 4.3.2. Data Transmission Format

The EMH-EN Communication Expansion Card supports a maximum of 16 register addresses for input (slave to master) and 16 register addresses for output (master to slave) per cycle.

The first input register addresses are fixed as 0x2100 (Errors and Warnings), the second default is 0x6100 (Current Status), and the third default is 0x6102 (Actual Frequency). The remaining 13 registers are 0x0830 (Reserved Address).

The first two output register addresses are default 0x6000 (Control Word) and 0x6002 (Target Frequency). The remaining 14 registers default to 0x0830 (Reserved Address). For detailed information, see the table below.

**Table 24 Cycle Data Table**

| Master Station Sends Data | | | |
|---|---|---|---|
| OUT ADDR[1] | OUT ADDR[2] | OUT ADDR[3-16] | |
| 0x6000<br>(Control cmd) | 0x6002<br>（Velocity Cmd） | 0x0830 (reserved) | |
| Slave Station Responds Data | | | |
| IN ADDR[1] | IN ADDR[2] | IN ADDR[3] | IN ADDR [4-16] |
| 0x2100<br>(fault warning code) | 0x6100<br>(Status) | 0x6102<br>(Current Velocity) | 0x0830 (reserved) |

### 4.3.3. Description of data sent by the master station

**Table 25 Master Station Sent Data PZD Description**

| Master Station Data PZD Description | |
|---|---|
| OUT ADDR[1]<br>0x6000<br>(Master to Slave) | Inverter Command Word (Command Source Needs to be Set to Communication F0-05 =5) |
| | Bit0: CMD_ACT Bit1: EXT_CMD1 Bit2: EXT_CMD2 Bit3: HALT<br>Bit4: LOCK Bit5: JOG Bit6: QSTOP Bit7: SERVO_ON<br>Bit8~Bit11: GEAR Bit12~Bit13: ACC/DEC Bit14: EN_SW Bit15: RST |
| OUT ADDR[2]<br>0x6002<br>(Master to Slave) | Inverter Target Frequency Command Word (Main Frequency Source must be set to Communication F0-06 = 8) |
| OUT ADDR[3-16]<br>(Master to Slave) | Default is address 2096 (Hexadecimal 0x0830), users can configure the inverter function code address during configuration. |

### 4.3.4. Inverter Response Data Description

**Table 26 Inverter Response Data Description**

| Inverter Response Data Description | |
|---|---|
| IN ADDR[1]<br>0x2100<br>(Slave to Master) | Current error and warning information of the inverter, detailed definitions refer to Section 4.5. |
| IN ADDR[2]<br>0x6100<br>(Slave to Master) | Bit0: ARRIVE Bit1: DIR Bit2: WARN Bit3: ERROR<br>Bit4: Reserve Bit5: JOG Bit6: QSTOP Bit7: SERVO_ON<br>Bit8: Ready Bit9~Bit15: Reserve |
| IN ADDR[3]<br>0x6102<br>(Slave to Master) | Current operating frequency of the inverter (unit: 0.01Hz). |
| IN ADDR[4-16]<br>(Slave to Master) | Default is address 2096 (Hexadecimal 0x0830), users can configure the inverter function code address during configuration. |

## 4.4. Communication Configuration

Configure the PLC and module to establish EtherNet/IP communication. The configuration process varies for different PLC models; please consult technical support for details.

### 4.4.1. Configure Configuration

1. Create a project

Open the upper-level computer software, create a new project, and select the device model as AM402-CPU1608TP/TN.

2. Import the EDS file and add the slave station.

Open the network configuration interface, click on the PLC to select the current communication protocol as EtherNet/IP master station, import the EDS file, and import the EIP expansion card's EDS file. Import the device in the network device list.

3、 Configure the slave station parameters.

Set the slave station IP address (the slave station IP address can be read and modified via the inverter function code).

Connect the PLC, configure the data, and configure the cyclic input address IN ADDR[1-16] and cyclic output address OUT ADDR[1-16]. The addresses are in decimal, and you need to convert the hexadecimal function codes to decimal.

4. Configure the Master Station

Scan the network to select the master station to be configured.

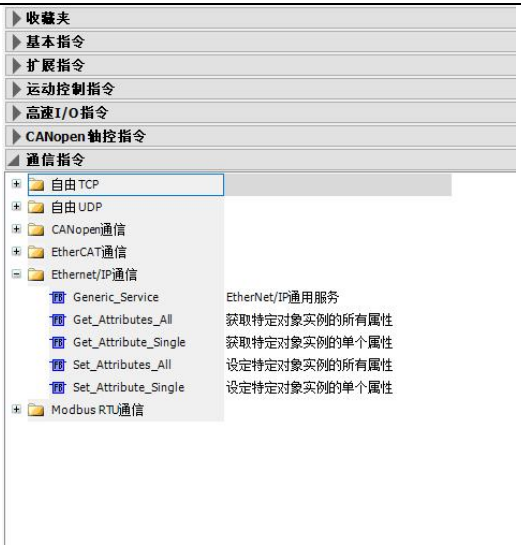
Assign an IP address to the master station's network port.

Download the project to the PLC.

The upper-level computer can be used to view periodic input and output data.

### 4.4.2. Configure Non-periodic Communication

Use the Get_Attribute_Single and Set_Attribute_Single function blocks for non-periodic read and write operations.



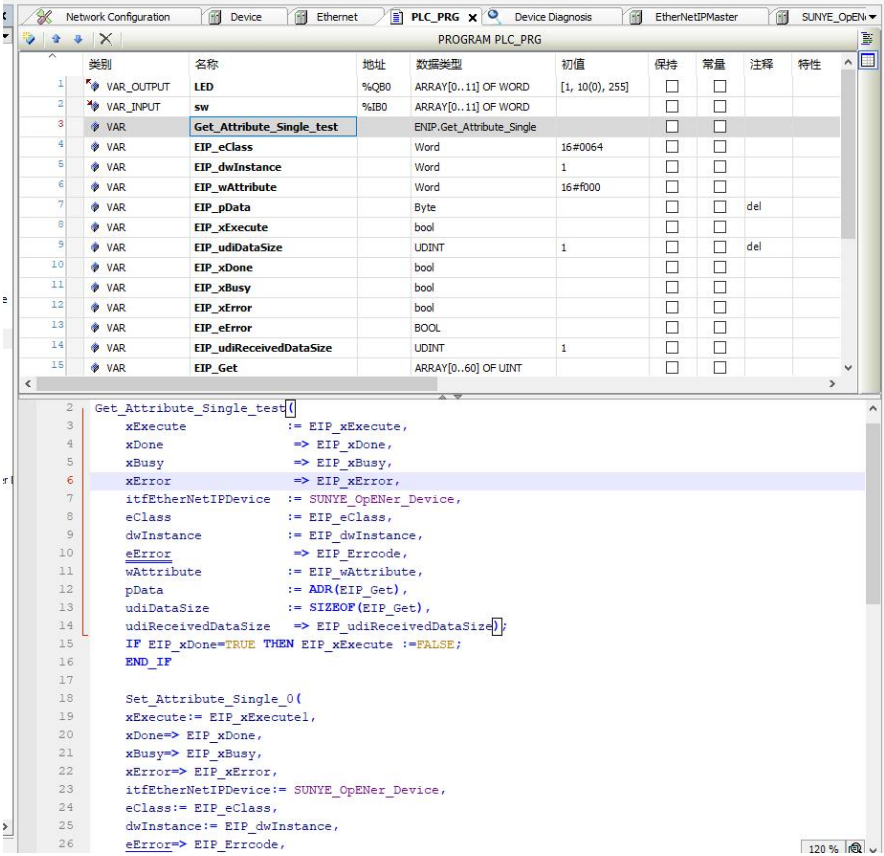

Non-periodic read and write function code parameter configuration

eClass configuration 100

dwInstance configuration 1

wAttribute configuration the address of the function code to be accessed

pData configuration the address for writing parameters or reading parameters




## 4.5. Fault Information

### 4.5.1. Periodic Fault Code

In the periodic communication PZD area, PZD1 sent from the slave station to the master station indicates error and alarm information, allowing users to obtain the latest status through periodic communication. In addition to the diagnostic information of the inverter itself, the communication status between the module and the inverter must also be included, allowing users to obtain the cause of errors. Detailed fault status codes are listed in the table below:

**Table 27 Periodic Fault Codes**

| Diagnostic Classification | Status Code |
|---|---|
| Inverter Body Diagnostic Information (Register Address: 0x2100) | 200 Communication Error (Message Abnormality)<br>201 Communication Error (Timeout Abnormality)<br>202 Address Error<br>203 Parameter Exceeds Threshold |

### 4.5.2. Read and clear fault history

1. Use the Get_Attribute_ALL function block for non-periodic reading of historical fault records, and use the Set_Attribute_ALL function block for non-periodic erasing of historical fault records.

eClass Configuration 101

dwInstance configuration 1

pData configuration the address for writing parameters or reading parameters

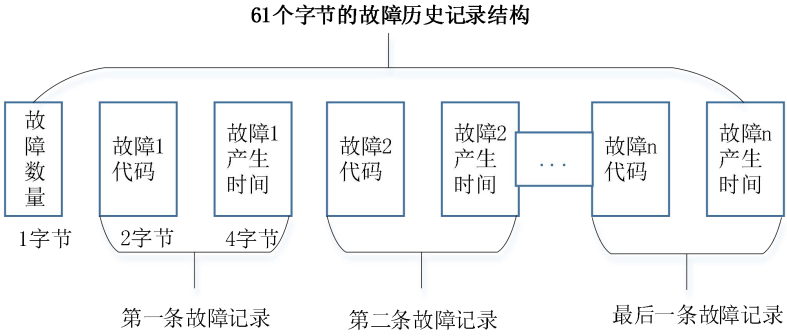
PROGRAM PLC_PRG

| | 类别 | 名称 | 地址 | 数据类型 | 初值 | 保持 | 常量 | 注释 | 特性 |
|---|---|---|---|---|---|---|---|---|---|
| 8 | VAR | EIP_udiDataSize | | UDINT | 1 | ☐ | ☐ | del | |
| 9 | VAR | EIP_xDone | | bool | | ☐ | ☐ | | |
| 10 | VAR | EIP_xBusy | | bool | | ☐ | ☐ | | |
| 11 | VAR | EIP_xError | | bool | | ☐ | ☐ | | |
| 12 | VAR | EIP_eError | | BOOL | | ☐ | ☐ | | |
| 13 | VAR | EIP_udiReceivedDataSize | | UDINT | 1 | ☐ | ☐ | | |
| 14 | VAR | EIP_Get | | ARRAY[0..60] OF UINT | | ☐ | ☐ | | |
| 15 | VAR | Set_Attribute_Single_test | | ENIP.Set_Attribute_Single | | ☐ | ☐ | | |
| 16 | VAR | EIP_xExecute1 | | BOOL | | ☐ | ☐ | | |
| 17 | VAR | Set_Attribute_Single_0 | | ENIP.Set_Attribute_Single | | ☐ | ☐ | | |
| 18 | VAR | EIP_Set | | ARRAY[0..9] OF UINT | | ☐ | ☐ | | |
| 19 | VAR | Get_Attributes_All_0 | | ENIP.Get_Attributes_All | | ☐ | ☐ | | |
| 20 | VAR | EIP_udiSendDataSize | | UDINT | | ☐ | ☐ | | |
| 21 | VAR | Set_Attributes_All_0 | | ENIP.Set_Attributes_All | | ☐ | ☐ | | |
| 22 | VAR | EIP_Errcode | | word | | ☐ | ☐ | | |

```
33      Get_Attributes_All_0(
34          xExecute:= EIP_xExecute,
35          xDone=> EIP_xDone,
36          xBusy=> EIP_xBusy,
37          xError=> EIP_xError,
38          itfEtherNetIPDevice:= SUNYE_OpENer_Device,
39          eClass:= EIP_eClass,
40          dwInstance:= EIP_dwInstance,
41          eError=> EIP_Errcode,
42          pData:= ADR(EIP_Get),
43          udiDataSize:= SIZEOF(EIP_Get),
44          udiReceivedDataSize=> EIP_udiReceivedDataSize);
45  IF EIP_xDone=TRUE THEN EIP_xExecute :=FALSE;
46  END_IF
47  Set_Attributes_All_0(
48              xExecute:= EIP_xExecute1,
49              xDone=> EIP_xDone,
50              xBusy=> EIP_xBusy,
51              xError=>EIP_xError,
52              itfEtherNetIPDevice:= SUNYE_OpENer_Device,
53              eClass:= EIP_eClass,
54              dwInstance:= EIP_dwInstance,
55              eError=> EIP_Errcode,
56              pData:= ADR(EIP_Set),
57              udiDataSize:=  SIZEOF(EIP_Set));
```

120 %

2. The structure of the fault history is: 2 bytes for the number of faults, 2 bytes for the fault code, and 4 bytes for the fault occurrence time, with the unit being seconds.




3. Error Code

**Table 28 Error Code Meaning**

| Error Code | Meaning |
|---|---|
| 201 | Send Error |
| 202 | Communication Timeout |
| 203 | Format Error |
| 204 | Address Error |
| 205 | Data Error |
| 206 | Inverter Error |
| 207 | Message Error |

# 5. EtherCAT Industrial Ethernet Communication

## 5.1. Product Information

The EMH-EC Communication Expansion Card is an EtherCat fieldbus adapter card that complies with the international standard for EtherCat fieldbus. This card is installed on the CM680 Inverter to enable communication with EtherCat master station devices, making the inverter a slave station of EtherCat, accepting control from the master station.

### 5.1.1. Physical Dimensions

Physical dimensions of the EtherCat module: PCB length 95mm, width 37mm, mounting hole diameter 3.5mm.

**Figure 19 Physical Dimensions of the EMH-EC Communication Expansion Card**

### 5.1.2. Interface Layout

The hardware layout of the EMH-EC Communication Expansion Card is shown in Figure 20. Pin Header X3 is used for connecting to the inverter and is located on the back of the EMH-EC Communication Expansion Card. The EMH-EC Communication Expansion Card provides 2 network ports, used for connecting the CM680-ECAT card to the master station (or the previous slave station) and the next slave station (if any) for communication.

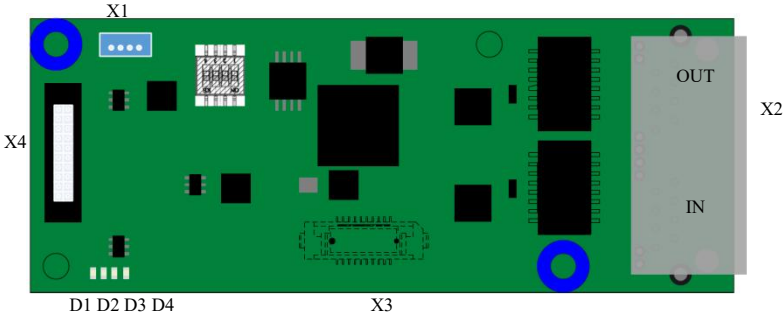



**Figure 20 EMH-EC Communication Expansion Card Interface Layout**

**Table 29 EMH-EC Communication Expansion Card Interface Description**

| Diagram Name | Hardware Name | Function Description |
|---|---|---|
| X1 | Socket | UART Interface: Print operation information, upgrade program |
| X2 | IN/OUT Network Port | Connect to the master station (or the previous slave station) / connect to the next slave station (if any) |
| X3 | Board-to-Board Socket | For Connection to Inverter |
| X4 | Socket | For Firmware Download and Debugging |
| D1, D2, D3, D4 | LED Indicator Light | Used to indicate the operating status, see Table 30 for details |

LED Indicator Light

**Table 30 EMH-EC Communication Expansion Card LED Indicator Light Description**

| Diagram Name | Indicator Light | Indicator Light Status | | Function Description |
|---|---|---|---|---|
| D4 | EtherCat Communication Status Indicator Light | Red Light | Constantly On | Indicates that the EtherCat module and EtherCat master station are in init |
| | | | Blinking (200ms on-off interval) | indicates that the module is communicating normally with the EtherCat master station |
| | | | single flash (200ms on, 1000ms off) | indicates that the EtherCat module and the master station are in PreOp |
| | | | flickering (1200ms on-off interval) | indicates that the EtherCat module and the master station are in SafeOp |
| | | | double flash (100ms off 100ms on 100ms off 100ms on 800ms off) | indicates that there is a communication error between the EtherCat module and the master station |
| | | | Off | indicates that the EtherCat module is communicating normally with the EtherCat master station, in OP |
| D3 | Inverter communication status indicator light | Red Light | Constantly On | the EtherCat module is just powered on or has timed out communication with the inverter |
| | | | Off | the EtherCat module and the inverter are communicating normally |
| | | | Blinking (200ms on-off interval) | indicates that the EtherCat module and the inverter have received non-periodic error codes |
| | | | double flash (100ms off 100ms on 100ms off 100ms on 800ms off) | indicates that the EtherCat module and the inverter have periodic communication errors |
| D2 | Program run status indicator light | Green light | Constantly On | Main program running normally |
| | | | Off | Main program not started properly |
| D1 | Hardware power indicator light | Green light | Constantly On | Module powered on normally |
| | | | Off | Module not powered on |

## 5.2. Installation and Wiring

### 5.2.1. Installation

The EMH-EC Communication Expansion Card is designed to be embedded in the CM680 Inverter. Before installation, please disconnect the power supply to the inverter and wait approximately 10 minutes until the inverter's charging indicator light is completely off before proceeding with the installation. After inserting the EMH-EC Communication Expansion Card into the inverter, please secure the corresponding screws to avoid poor contact of the board-to-board signal sockets. The installation diagram is shown in Figure 18.

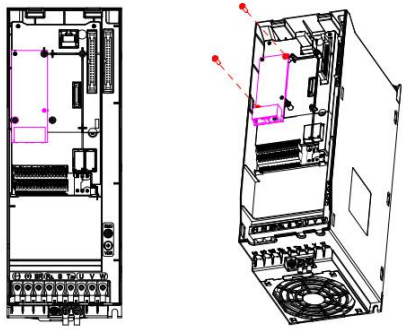


**Figure 21 EMH-EC Communication Expansion Card Installation Diagram**

### 5.2.2. Wiring

The topologies supported by EtherCat include bus, star, tree, and combinations of these topologies, making device connection and wiring very flexible and convenient. The bus connection topology diagram is shown in Figure 19.

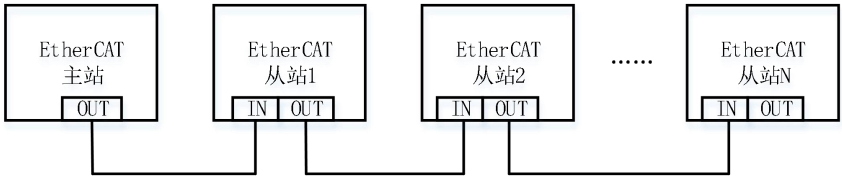



**Figure 22 EtherCat Bus Wiring Diagram**

## 5.3. Communication Description

### 5.3.1. Key Points of EtherCat Communication

- When using the EMH-EC Communication Expansion Card, the communication decoding method of the CM680 needs to be set. When using the CANopen standard DS402 specification, set the F8-19 register to 1 via the operation panel. When using manufacturer-defined command codes, set the F8-19 register to 0 via the operation panel. Additionally, check whether the manufacturer-defined command codes are for the 0x6000 or 0x2000 section. When using the 0x6000 section, set the F8-07

register to 1 via the operation panel. When using the 0x2000 section, set the F8-07 register to 0 via the operation panel.

- When setting the inverter command, the F0-05 register of the CM680 needs to be set to 5.
- When setting the inverter frequency, the F0-06 register of the CM680 needs to be set to 8.
- The alias address for EMH-EC Communication Expansion Card EtherCat communication is set in the F8-11 register, and the module must be restarted for the setting to take effect.
- During non-periodic communication, a new non-periodic communication must only be initiated after the completion of the previous non-periodic read/write. The interval between two non-periodic communications should be no less than 100ms to avoid affecting the response to periodic communication.

### 5.3.2. Periodic Data Read/Write Description

Periodic data exchange is the basic data interaction method of EtherCat. After the master and slave stations establish a normal connection, they will continuously send periodic interaction messages.

Periodic Interaction Data Format:

Periodic interaction sets up 8 PDO areas, 4 of which are RxPDOs, and the other 4 are TxPDOs. There are some default settings for these 8 PDO areas, as shown in the table below:

**Table 31 Periodic Interaction Data**

| PDO | Explanation | Number of PDOs |
|---|---|---|
| RxPDO1 | Default transmission PDO, used for object dictionary 402, such as 0x6040 | Min(16 objects, 32 bytes of data) |
| RxPDO2 | Default transmitted PDO, used for the object dictionary of standard Modbus addresses, such as 0x3060 | Min(16 objects, 32 bytes of data) |
| RxPDO3 | Default transmitted PDO, used for the object dictionary of standard Modbus addresses, such as 0x3020 | Min(16 objects, 32 bytes of data) |
| RxPDO4 | Non-default freely configurable transmitted PDO | Min(16 objects, 32 bytes of data) |
| TxPDO1 | Default received PDO, used for the object dictionary of 402, such as 0x6041 | Min(16 objects, 32 bytes of data) |
| TxPDO2 | Default received PDO, used for the object dictionary of standard Modbus addresses, such as 0x3061 | Min(16 objects, 32 bytes of data) |
| TxPDO3 | Default received PDO, used for the object dictionary of standard Modbus addresses, such as 0x3021 | Min(16 objects, 32 bytes of data) |
| TxPDO4 | Non-default freely configurable received PDO | Min(16 objects, 32 bytes of data) |

The default configured RPDO is 0x1601, and the default configured TPDO is 0x1A01. The first three items of 0x1601 are default mappings, mapping to objects of 402, which are 0x6040, 0x6042, 0x6060, representing control word, set target speed, set mode; the remaining mappings in 0x1601 can be freely configured; The first four items of 0x1A01 are default mappings, with the object at address 402 being 0x603F, 0x6041, 0x6043, 0x6061, representing warning/error, status word, actual operating speed, actual operating mode, respectively. The remaining mappings in 0x1A01 can be freely configured as shown in the table below:

**Table 32 Periodic Interaction Data Format 1**

| Master Station sends RPDO area (0x1601) | | | | |
|---|---|---|---|---|
| Default RPDO | | | Variable RPDO | |
| Control Word （0x6040） | Set Target Speed （0x6042） | Set Mode （0x6060） | Real-time modification of inverter function parameters | |
| RPDO1 | RPDO2 | RPDO3 | RPDO4~RPDO16 | |
| Inverter Slave Station replies with TPDO area (0x1A01) | | | | |
| Default TPDO | | | | Variable TPDO |
| Warning/Error （0x603F） | Status Word （0x6041） | Actual Running Speed （0x6043） | Actual Running Mode （0x6061） | Real-time reading of inverter function parameters |
| TPDO1 | TPDO2 | TPDO3 | TPDO4 | TPDO5~TPDO16 |

Another default configuration is 0x1602 and 0x1A02, with a maximum mapping of 16 register addresses. The first two items of 0x1602 are the default RPDO mappings, which are 0x6000, 0x6002, representing control word and set target frequency. The remaining mappings in 0x1602 can be freely configured; The first three items of 0x1A02 are the default mappings, with addresses 0x2100, 0x6100, 0x6102, representing warning/error, status word, and actual frequency. The remaining mappings in 0x1A02 can be freely configured, as shown in the table below:

**Table 33 Periodic Interaction Data Format 2**

| Master Station Sends RPDO Area (0x1602) | | | |
|---|---|---|---|
| Fixed RPDO | | | Variable RPDO |
| Control Word （0x6000） | Set Target Frequency （0x6002） | | Real-time modification of inverter function parameters |
| RPDO1 | RPDO2 | | RPDO3~RPDO16 |
| Inverter Slave Station Responds with TPDO Area (0x1A02) | | | |
| Fixed TPDO | | | Variable TPDO |
| Warning/Error （0x2100） | Status Word （0x6100） | Actual Operating Frequency （0x6102） | Real-time reading of inverter function parameters |
| TPDO1 | TPDO2 | TPDO3 | TPDO4~TPDO16 |

Another default configuration is 0x1603 and 0x1A03, with a maximum mapping of 16 register addresses. The first two items of 0x1603 are the default RPDO mappings, 0x2000, 0x2001, representing control words and set target frequency. The remaining mappings in 0x1603 can be freely configured; The first three items of 0x1A03 are default mappings, with mapping addresses 0x2100, 0x2101, 0x2103, representing warning/error, status word, actual frequency. The remaining mappings in 0x1A03 can be freely configured, as shown in the table below:

**Table 34 Periodic Data Exchange Format 3**

| Master Station sends RPDO area (0x1603) | | |
|---|---|---|
| Fixed RPDO | | Variable RPDO |
| Control Word （0x2000） | Set Target Frequency （0x2001） | Real-time modification of inverter function parameters |
| RPDO1 | RPDO2 | RPDO3~RPDO16 |
| Inverter Slave Station replies with TPDO area (0x1A03) | | |

| Fixed TPDO | | | Variable TPDO |
|---|---|---|---|
| Warning/Error （0x2100） | Status Word （0x2101） | Actual Operating Frequency （0x2103） | Real-time reading of inverter function parameters |
| TPDO1 | TPDO2 | TPDO3 | TPDO4~TPDO16 |

Additionally, 0x1604 and 0x1A04 have no default mapping configurations and are available for user configuration.

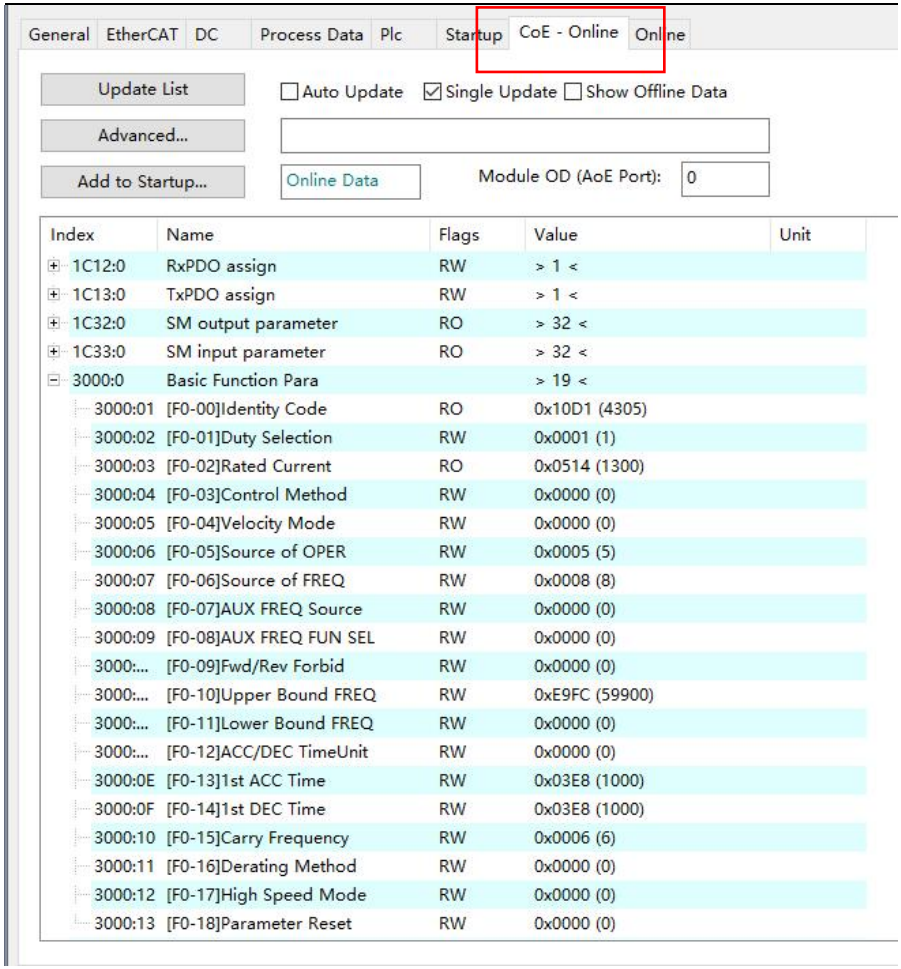
### 5.3.3. Non-periodic Data Read/Write Instructions

Non-periodic data read/write is used for obtaining or writing non-real-time data, interacting only when necessary.

There are two methods of non-periodic interaction, one is to access non-periodic data through the COE-Online method, and the other is to complete non-periodic interaction through custom object dictionary indexes.

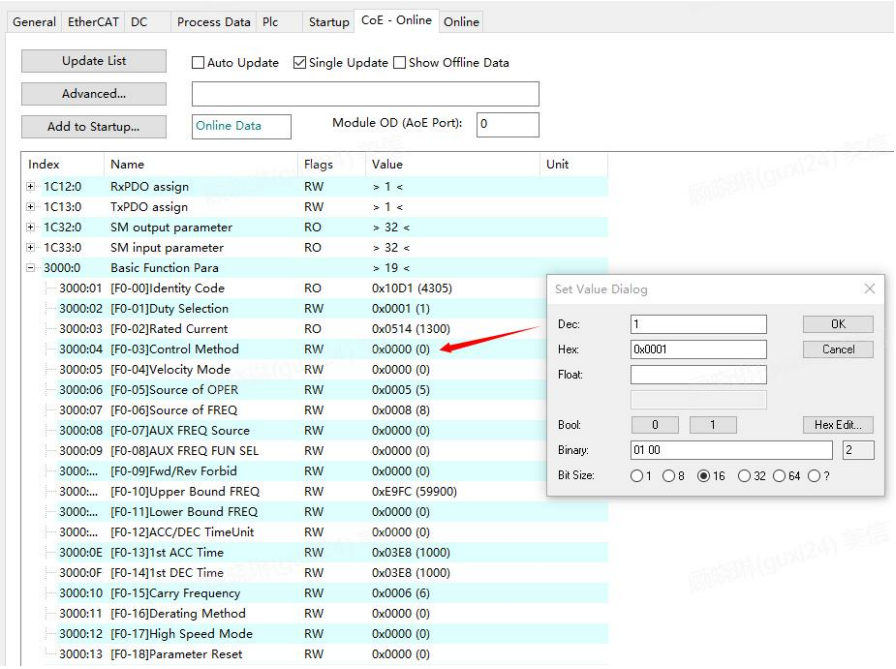
#### 5.3.3.1. Non-periodic COE-Online Acquisition

COE-Online accesses data from all object dictionaries listed in the ESI file in sequence, including data reading and writing. Since the content of the object dictionary is written based on the function codes of the inverter, the sub-indexes under an index are not necessarily continuous. Therefore, using the Complete access method to access data can cause data misalignment, so COE-Online only supports non-Complete access methods to access sub-index objects one by one. As shown in the figure below, the parameters of the inverter are read non-periodically in sequence through the COE-Online method.




| Index | Name | Flags | Value | Unit |
|---|---|---|---|---|
| 1C12:0 | RxPDO assign | RW | > 1 < | |
| 1C13:0 | TxPDO assign | RW | > 1 < | |
| 1C32:0 | SM output parameter | RO | > 32 < | |
| 1C33:0 | SM input parameter | RO | > 32 < | |
| 3000:0 | Basic Function Para | | > 19 < | |
| 3000:01 | [F0-00]Identity Code | RO | 0x10D1 (4305) | |
| 3000:02 | [F0-01]Duty Selection | RW | 0x0001 (1) | |
| 3000:03 | [F0-02]Rated Current | RO | 0x0514 (1300) | |
| 3000:04 | [F0-03]Control Method | RW | 0x0000 (0) | |
| 3000:05 | [F0-04]Velocity Mode | RW | 0x0000 (0) | |
| 3000:06 | [F0-05]Source of OPER | RW | 0x0005 (5) | |
| 3000:07 | [F0-06]Source of FREQ | RW | 0x0008 (8) | |
| 3000:08 | [F0-07]AUX FREQ Source | RW | 0x0000 (0) | |
| 3000:09 | [F0-08]AUX FREQ FUN SEL | RW | 0x0000 (0) | |
| 3000:... | [F0-09]Fwd/Rev Forbid | RW | 0x0000 (0) | |
| 3000:... | [F0-10]Upper Bound FREQ | RW | 0xE9FC (59900) | |
| 3000:... | [F0-11]Lower Bound FREQ | RW | 0x0000 (0) | |
| 3000:... | [F0-12]ACC/DEC TimeUnit | RW | 0x0000 (0) | |
| 3000:0E | [F0-13]1st ACC Time | RW | 0x03E8 (1000) | |
| 3000:0F | [F0-14]1st DEC Time | RW | 0x03E8 (1000) | |
| 3000:10 | [F0-15]Carry Frequency | RW | 0x0006 (6) | |
| 3000:11 | [F0-16]Derating Method | RW | 0x0000 (0) | |
| 3000:12 | [F0-17]High Speed Mode | RW | 0x0000 (0) | |
| 3000:13 | [F0-18]Parameter Reset | RW | 0x0000 (0) | |

The following figure shows the non-periodic modification of a sub-index value through the COE-Online method to change the inverter parameter value.




### 5.3.3.2. Non-periodic acquisition of custom object dictionary

1. Non-periodic interaction data format

Non-periodic interaction is accomplished using specific object dictionary indexes to determine the format of the message data area. To facilitate user operations on continuous registers of the inverter, the following non-periodic read/write formats are defined. Non-periodic reads use a combination of object dictionary indexes 0x5300 and 0x5301, while non-periodic writes use the object dictionary index 0x5600.

1) Non-periodic write value (0x5600)

When executing a non-periodic write value command, the data area must include the starting address of the register, length, and value. The data area has a fixed length of 16, and unused data areas are automatically filled with 0. The specific format is as follows:

**Table 35 Non-periodic Write Value Format**

| | Register Address | Register Length | Data |
|---|---|---|---|
| Data Length | 2 Byte | 2 Byte | Valid Data Area (Register Length * 2) Byte + Invalid Data Area |
| Example | 0x0400 | 0x10 | 0x0001 0002 0003 0004 0005… |

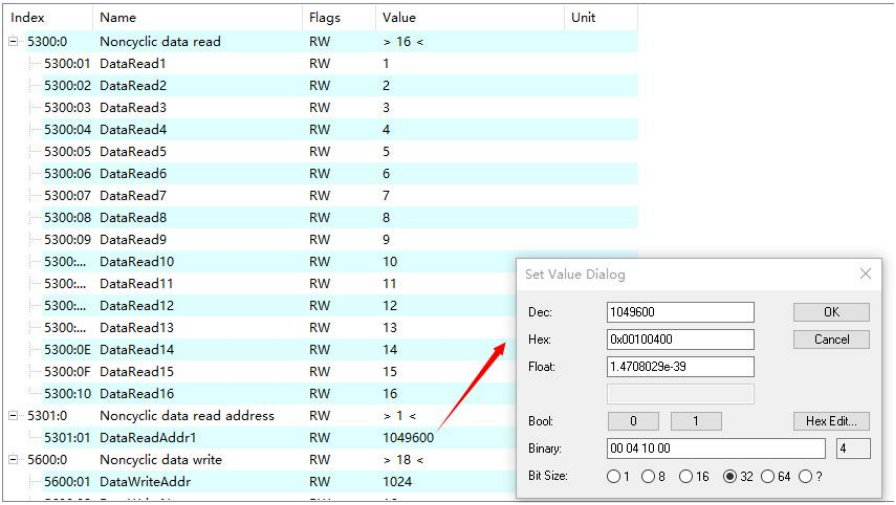
The figure below is an example of an actual object dictionary:

| Index | Name | Flags | Value |
|---|---|---|---|
| 5600:0 | Noncyclic data write | RW | > 18 < |
| 5600:01 | DataWriteAddr | RW | 1024 |
| 5600:02 | DataWriteNum | RW | 16 |
| 5600:03 | DataWrite1 | RW | 1 |
| 5600:04 | DataWrite2 | RW | 2 |
| 5600:05 | DataWrite3 | RW | 3 |
| 5600:06 | DataWrite4 | RW | 4 |
| 5600:07 | DataWrite5 | RW | 5 |
| 5600:08 | DataWrite6 | RW | 6 |
| 5600:09 | DataWrite7 | RW | 7 |
| 5600:... | DataWrite8 | RW | 8 |
| 5600:... | DataWrite9 | RW | 9 |
| 5600:... | DataWrite10 | RW | 10 |
| 5600:... | DataWrite11 | RW | 11 |
| 5600:0E | DataWrite12 | RW | 12 |
| 5600:0F | DataWrite13 | RW | 13 |
| 5600:10 | DataWrite14 | RW | 14 |
| 5600:11 | DataWrite15 | RW | 15 |
| 5600:12 | DataWrite16 | RW | 16 |

The total length of the non-periodic write value data area is at least 2 Byte and a maximum of 32 Byte.

2) Non-periodic Read Value (0x5300 + 0x5301)

In the non-periodic read value request message, the register address and number to be read must first be written to the index 0x5301, as shown in the figure, the address written is 0x0400, reading 16 registers, so it is 0x00 10 04 00, the data read is stored in the index 0x5300, as shown in the figure, the fixed maximum number of non-periodic reads is 16 registers, the values read are sequentially filled into the sub-indexes of 0x5300, for example, the 16 register values read this time are 1~16, if there are unused data areas, they are automatically filled with 0.




The specific format of the 0x5301 index is as follows

**Table 36 0x5301 Index Format**

| | Number of Registers | Register Address |
|---|---|---|
| Data Length | 2 Byte | 2 Byte |
| Example | 0x0400 | 0x0010 |

After sending a non-periodic read value command, the read value is filled into the 0x5300 index, and unused index values are automatically filled with 0. The data area format is as follows:

**Table 37 Data Area Format**

| | Data |
|---|---|
| Data Length | (Register Length * 2) Byte |
| Example | 0x0001 0002 0003 0004 0005… |

3) Execution Result

If an exception occurs during non-periodic interaction, an error code will be returned. Refer to Table 12 for specific error code definitions.

**Table 38 EMH-EC Communication Expansion Card SDO Transmission Error Codes**

| Error Code | Meaning |
|---|---|
| 0x05030000 | No change in the toggle bit during segmented transfer |
| 0x05040000 | SDO transfer timeout |
| 0x05040001 | Invalid or unknown command code |
| 0x05040005 | Memory overflow |
| 0x06010000 | Operation on a certain object is not supported |
| 0x06010001 | Read a write-only data object |
| 0x06010002 | Write a read-only data object |
| 0x06010003 | The entry cannot be written because sub-index 0 is not 0 |
| 0x06010004 | The object cannot be accessed via complete access |
| 0x06020000 | The data object does not exist in the data dictionary |
| 0x06040041 | The data object cannot be mapped to PDO |
| 0x06040042 | The number and length of data objects to be mapped exceed the PDO data length |
| 0x06040043 | Parameter Incompatible |
| 0x06040047 | Device Internal Incompatibility |
| 0x06060000 | Hardware Error |
| 0x06070010 | Parameter Length Mismatch |
| 0x06070012 | Parameter Length Too Long |
| 0x06070013 | Parameter Length Too Short |
| 0x06090011 | Subindex Does Not Exist |
| 0x06090030 | Write Operation: Data Value Out of Range |
| 0x06090031 | Data Value Too Large |
| 0x06090032 | The data value to be written is too small |
| 0x06090033 | 0xF030 and 0xF050 do not match |
| 0x06090036 | The maximum value is less than the minimum value |
| 0x08000000 | General Error |
| 0x08000020 | Data cannot be read or written |
| 0x08000021 | Due to local control reasons, the data cannot be transmitted or saved to the application |
| 0x08000022 | Data cannot be read or written in the current state |
| 0x08000023 | The object cannot be found in the current object dictionary |

### 5.4. Communication Configuration

The following describes the simple configuration and usage process of the CM680-ECAT card using Beckhoff's TwinCAT3 master station as an example.

Note: For the network card, an Intel chip-based 100 Mbps Ethernet card must be selected. Network cards from other brands may pose a risk of not supporting EtherCAT operation.

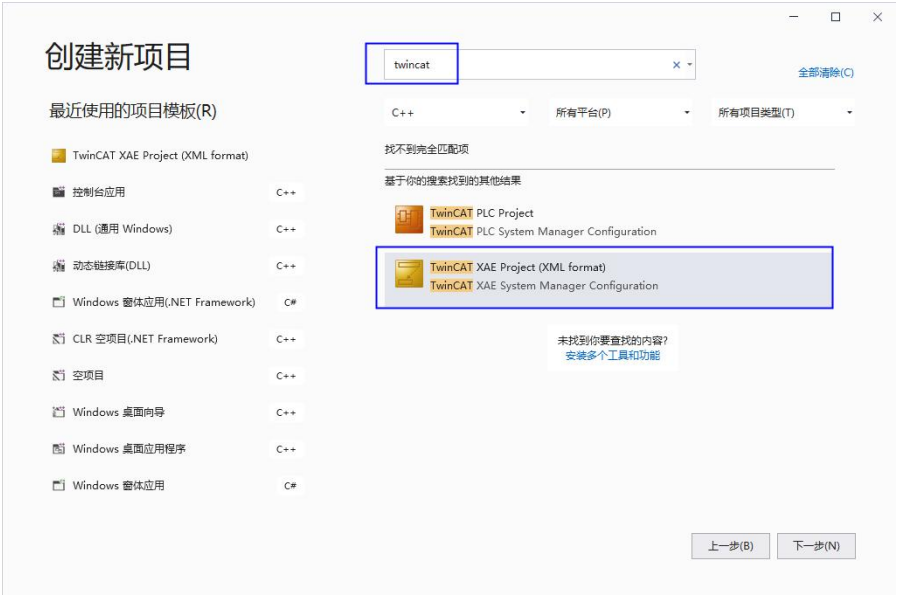
### 5.4.1. Configure Configuration

1. Install TwinCAT3

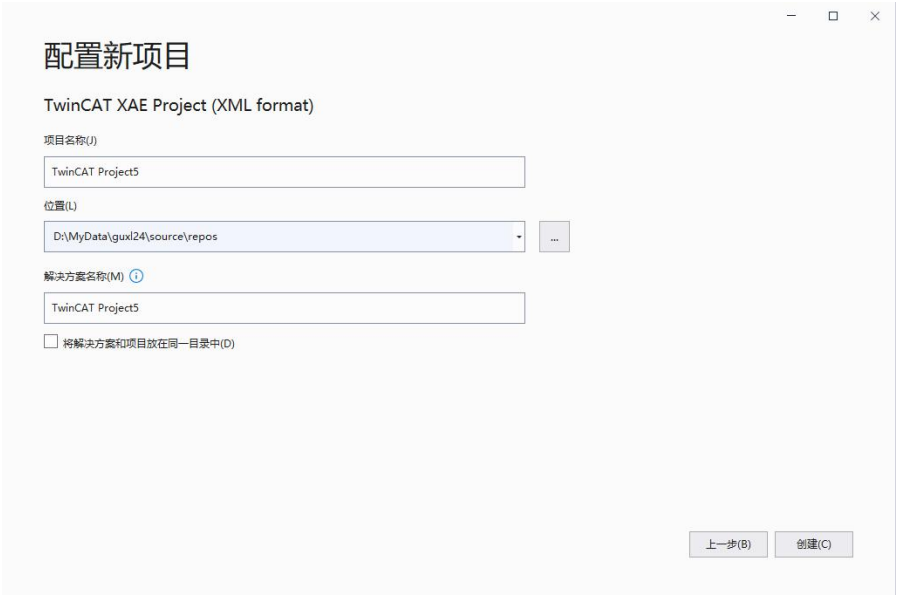
2. Copy the CM680 EtherCat configuration file (CM680H_PLUS_IO_V1.5.xml) to the TwinCAT installation directory, TwinCAT3 directory: TwinCAT\3.1\config\IO\EtherCAT.

3. Open TwinCAT

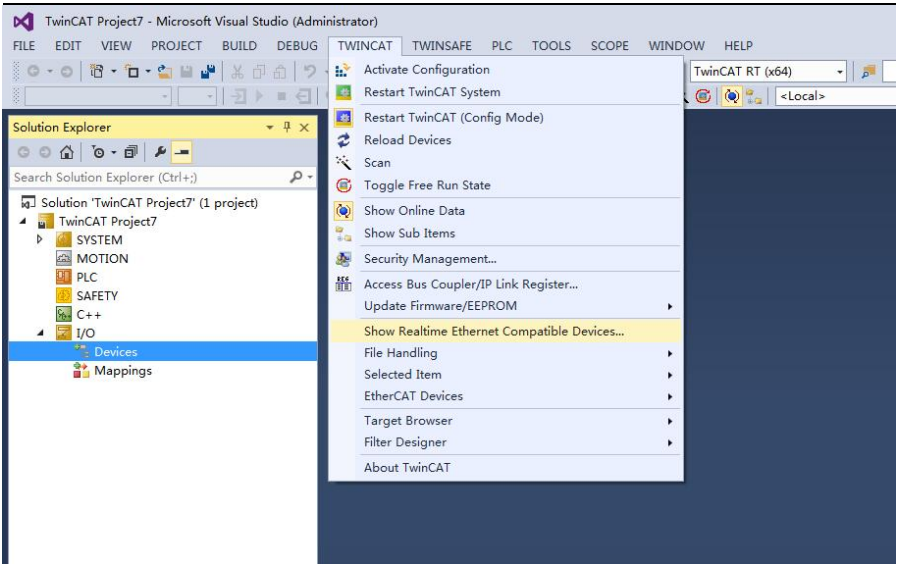
Create a new project, click on 'Create New Project', and the following interface will appear. You can search for TwinCAT templates, select the 'TwinCAT XAE Project (XML format)' template, and click Next.



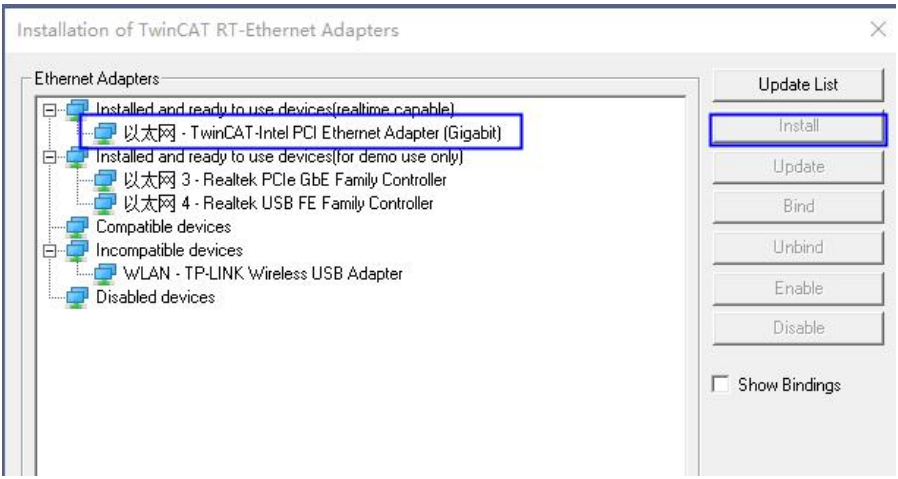
The following interface will appear. After filling in the project name and save location, click Create.



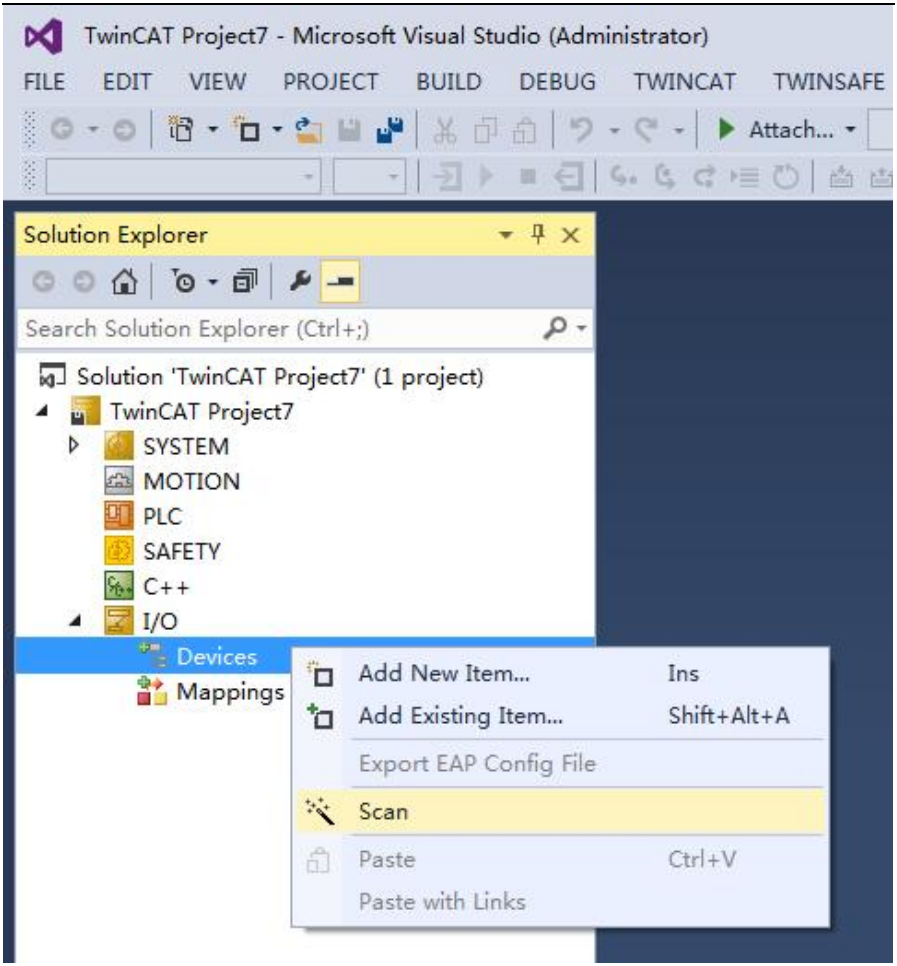

After creation, you need to install the TwinCAT network card driver.




Open the menu 'Show Real Time Ethernet Compatible Devices...', and the following dialog box will appear. In the 'Incompatible devices' section, select the local site and click 'Install'. After installation, the installed network card will appear in the 'Installed and ready to use devices' section as shown in the figure below.



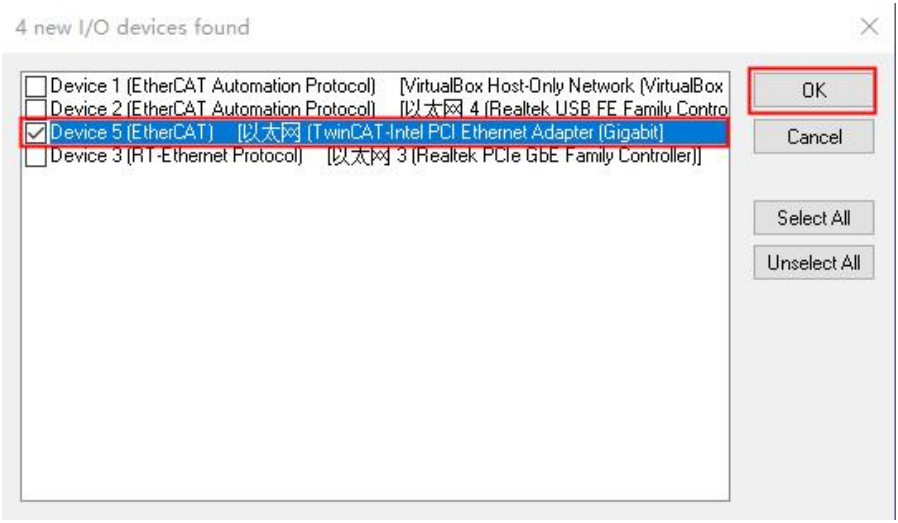

After the driver installation is complete, connect the device and search for it by right-clicking on Device and selecting Scan, as shown in the figure below.


TwinCAT Project7 - Microsoft Visual Studio (Administrator)
FILE EDIT VIEW PROJECT BUILD DEBUG TWINCAT TWINSAFE
Attach...
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'TwinCAT Project7' (1 project)
TwinCAT Project7
SYSTEM
MOTION
PLC
SAFETY
C++
I/O
Devices
Mappings
Add New Item... Ins
Add Existing Item... Shift+Alt+A
Export EAP Config File
Scan
Paste Ctrl+V
Paste with Links

Click OK.

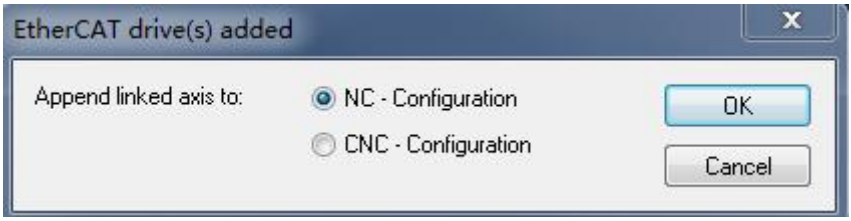

Microsoft Visual Studio
HINT: Not all types of devices can be found automatically
确定
取消

Select the Ethernet that has just installed the Ethercat driver and click OK.

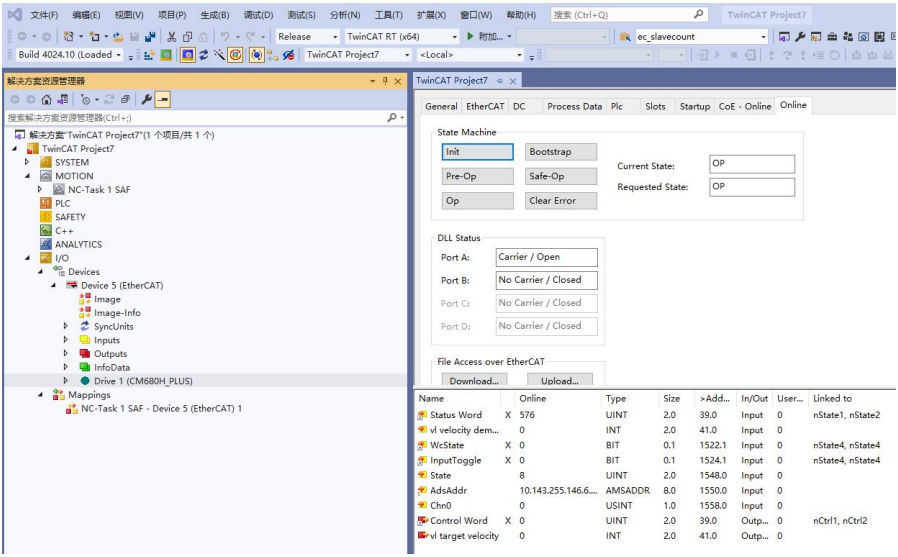



Click Yes.


Click OK.



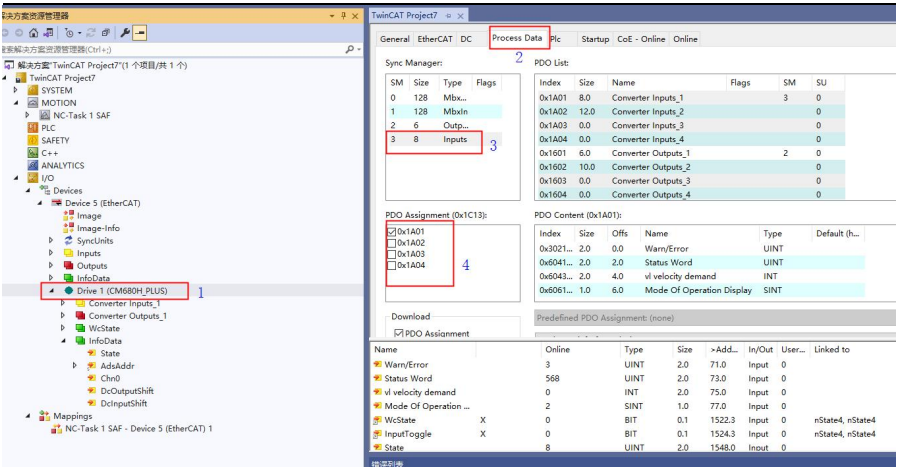
Click Yes. At this point, the device search is complete, as shown in the figure below.

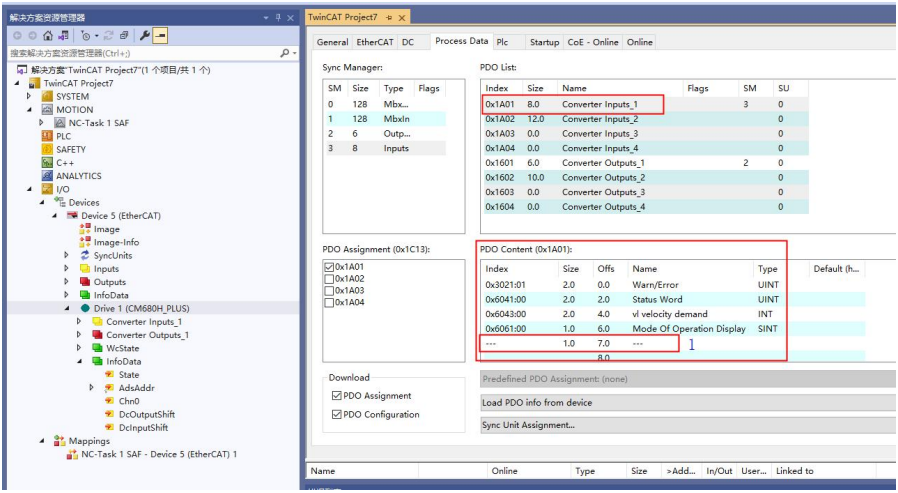

### 5.4.2. Configure Periodic Communication

1. Configure TPDO

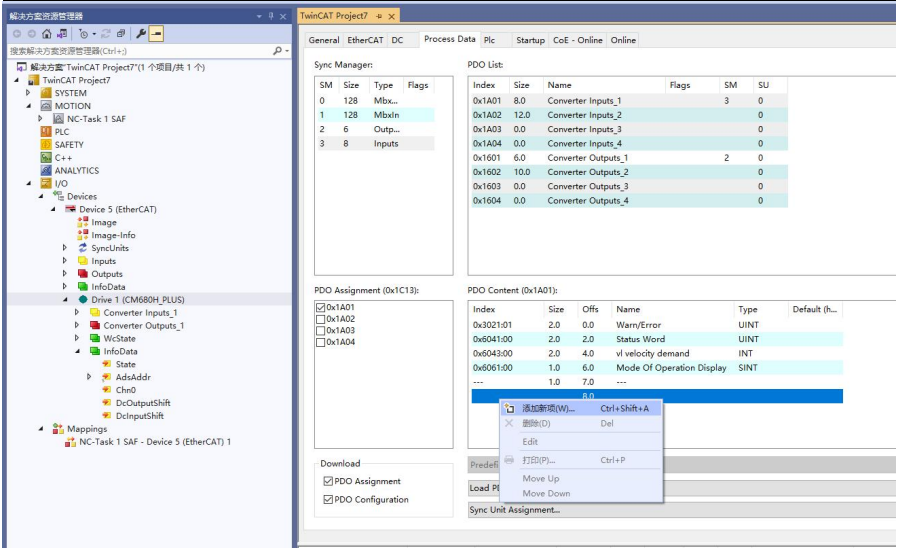
Select the slave station, then select the 'Process Data' item on the right, and choose SM3, which is Inputs. You will see four TPDO options: 0x1A01, 0x1A02, 0x1A03, and 0x1A04. Among these, 0x1A01 is the default configuration for the standard DS402 protocol object dictionary, 0x1A02 and 0x1A03 are default configurations for custom function codes, and 0x1A04 has no default mapping object configuration and is available for user configuration. Users can select one of the four TPDOs according to their needs, and check the box before the selected TPDO. The currently selected TPDO is 0x1A01, as shown in the following steps.



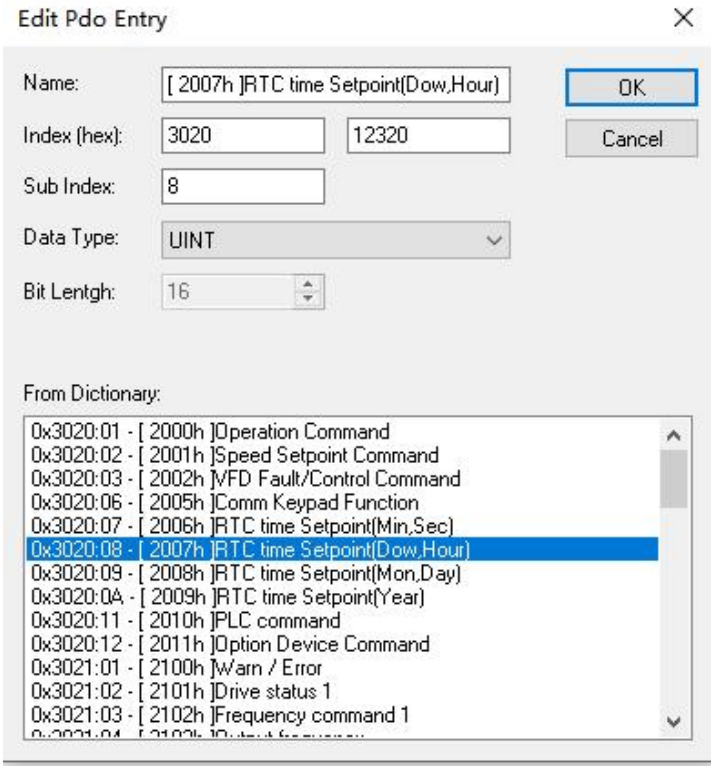

For the default TPDO 0x1A01, the default mapping objects configured are 4, as shown in the following figure: 0x3021:01 Warn/Error, 0x6041:00 Status Word, 0x6043:00 Velocity Demand, 0x6061:00 Mode of Operation Display. Since 0x6061:00 Mode of Operation Display is 1 byte, to align the bytes, an unused empty object needs to be added, as shown in Figure 1.




For the default configuration of 0x1A01, it is not recommended that users modify the first (4+1) configured objects. However, users can add new objects after these, as shown in the following figure, by right-clicking and adding a new item after the last object.




Then, a list of all object dictionaries will pop up, as shown in the figure below, from which you can select the object dictionary you want to add.

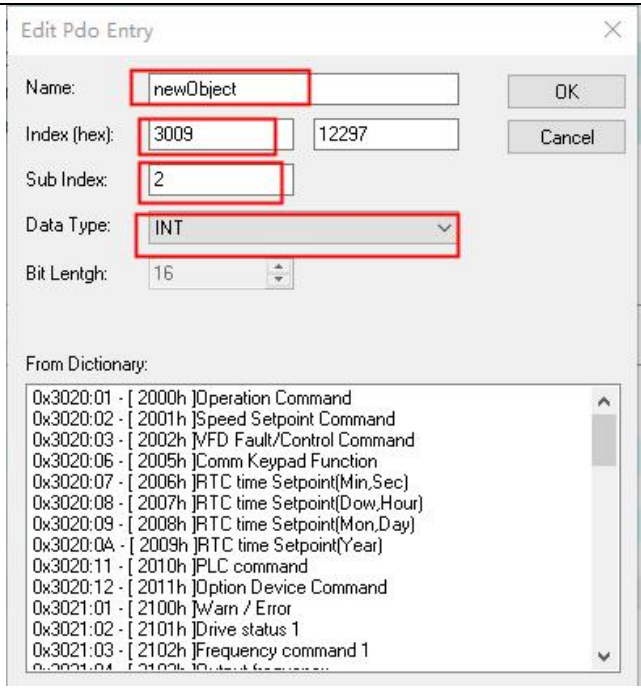



For the object dictionary you want to add that does not appear in the popped-up object dictionary list, you can manually input it, as shown in the figure below, in the red box marked, input the object dictionary you want to add, and also pay attention to the mapping relationship between the object dictionary and the inverter function code as follows:

Object Dictionary Index = 0x3000 + Function Code Group Number;

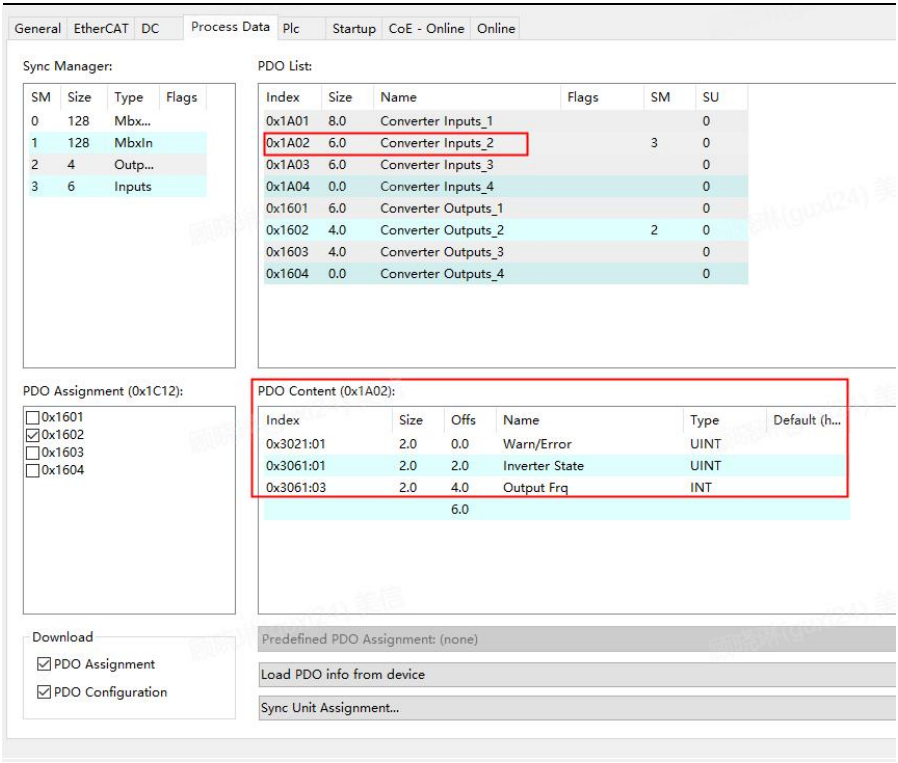
Object Dictionary Sub-index = Function Code Group Internal Offset in Hexadecimal + 1;

Therefore, as shown in the figure below, the object dictionary 0x3009+2 manually added represents the inverter function code 0x0901.



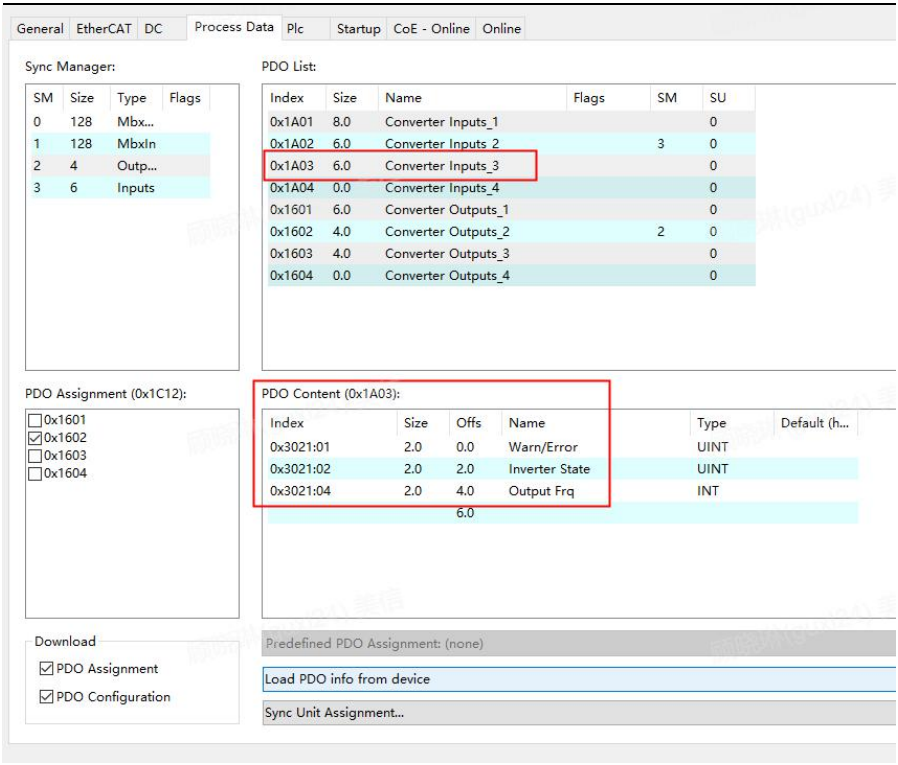

In addition, note that 1) if the length of the object dictionary being added is less than 2 bytes, it must be padded with dummy object dictionaries to achieve 2-byte alignment; 2) the maximum number of object dictionaries that can be added under a single PDO is min(16, the number of objects corresponding to 32 bytes of object dictionary length).

For the default TPDO 0x1A02, the default mapping objects configured are 3, as shown in the figure below, which are 0x3021:01 Warn/Error, 0x3061:01 Inverter State, 0x3061:03 Output Frq.




For the default configuration of 0x1A02, it is not recommended that users modify the first three configured objects, but users can add desired objects after these. The method of adding objects is the same as the configuration for 0x1A01. Additionally, if customers want to use 0x1A04 without default configuration, the method of adding objects is also the same.

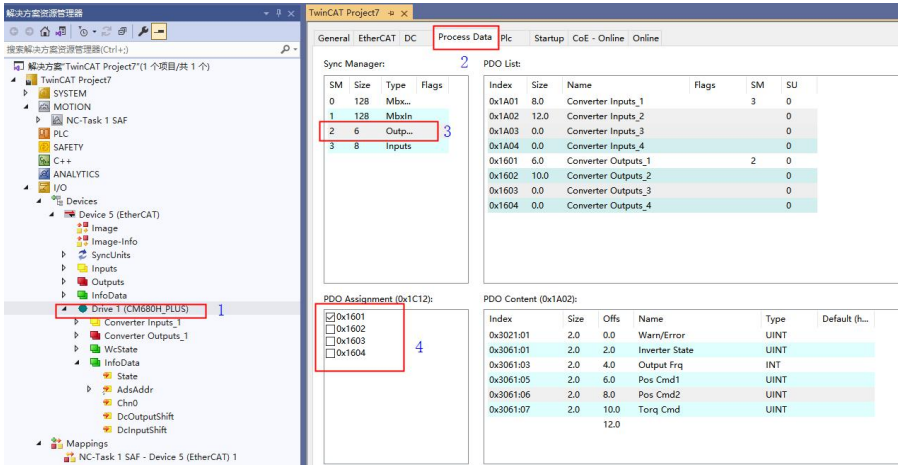
For the default TPDO 0x1A03, there are 3 default mapped objects, as shown in the figure below, which are 0x3021:01 Warn/Error, 0x3021:02 Inverter State, 0x3021:04 Output Frq.



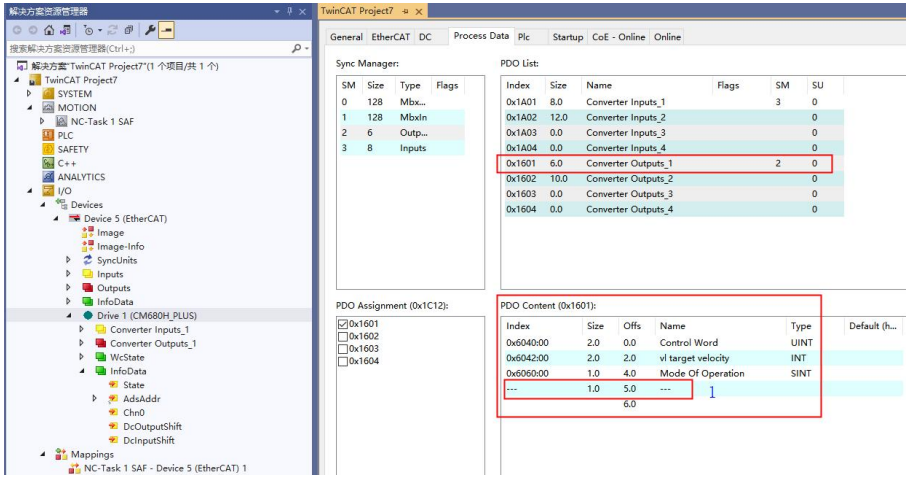

For the default configuration of 0x1A03, it is not recommended that users modify the first 3 configured objects, but users can add objects they wish to include after these, using the same method as configuring 0x1A01.

2. Configure RPDO

Select the slave station, then select the 'Process Data' item on the right, and select SM2, i.e., Outputs. You can see four TPDO options: 0x1601, 0x1602, 0x1603, and 0x1604. Among them, 0x1601 is the standard DS402 protocol object dictionary configured by the manufacturer by default, 0x1602 is the custom function code configured by the manufacturer by default, and 0x1603 and 0x1604 do not have default mapped objects and are available for user configuration. Users can choose one of the four RPDOs according to their needs, and check the box before the selected RPDO. The default selected is 0x1601, as shown in the following steps.

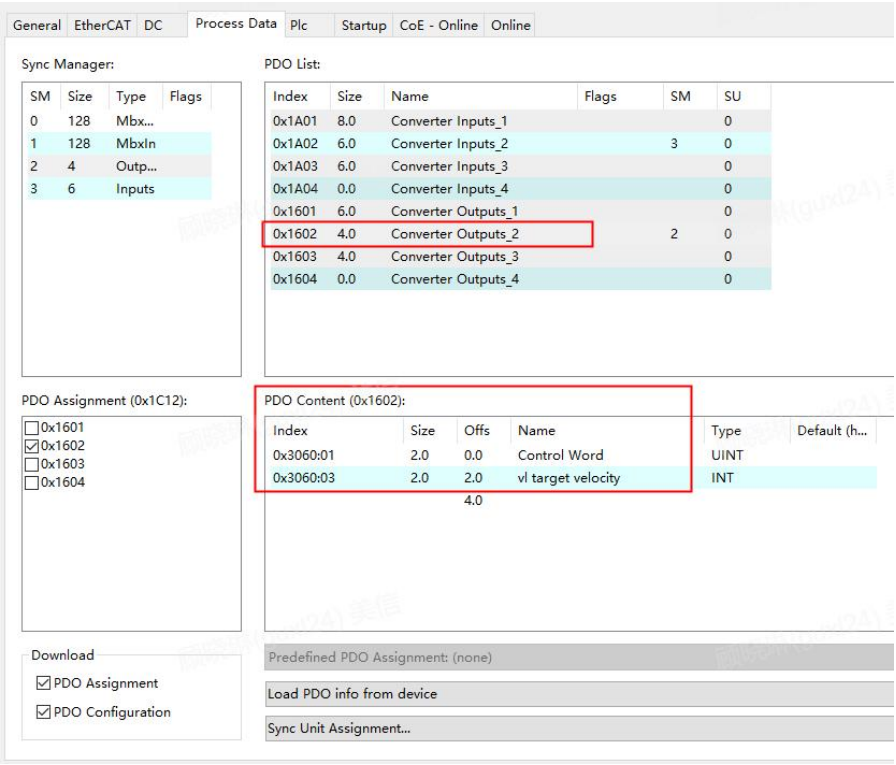



For the default RPDO 0x1601, there are 3 default mapped objects configured, as shown in the following figure: 0x6040:00 Control Word, 0x6042:00 vl target velocity, 0x6060:00 Mode of Operation. Since 0x6060:00 Mode of Operation is 1 byte, to align the bytes, an unused empty object needs to be added, as shown in 1 in the following figure.



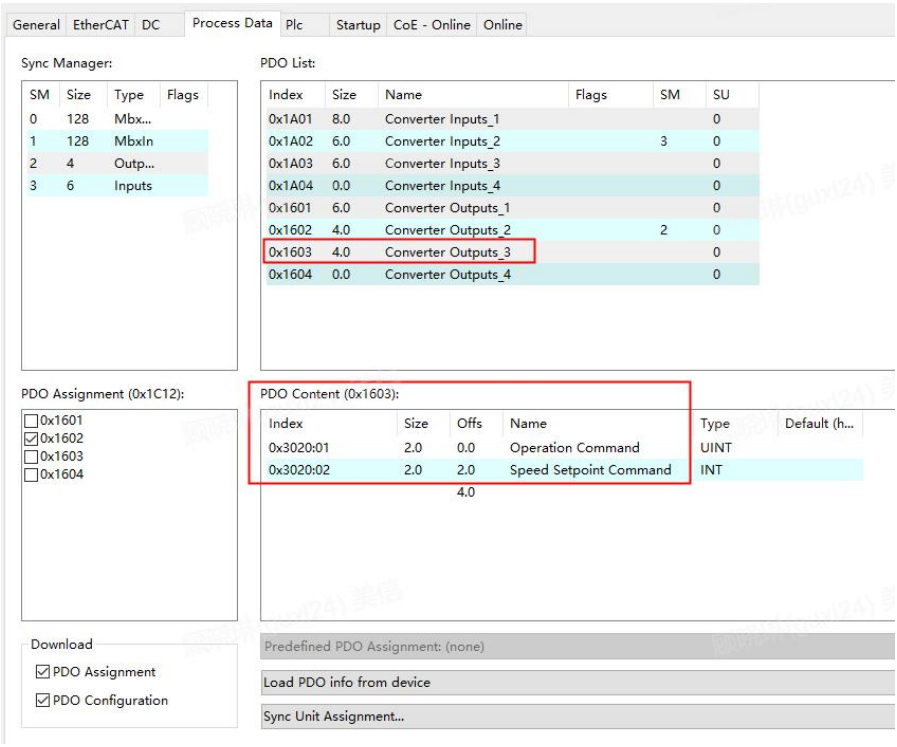
For the default configuration 0x1601, it is not recommended that users modify the first (3+1) configured objects, but users can add objects they wish to include after these, using the same method as configuring 0x1A01.

For the default RPDO 0x1602 configuration, there are 2 default mapped objects, as shown in the figure below, which are 0x3060:01 Control Word, 0x3060:03 vl target velocity.




For the 0x1602 with default configuration, it is not recommended that users modify the first 2 configured objects, but users can add objects they wish to increase after these, using the same method as configuring 0x1A01. Additionally, if customers want to use 0x1604 without default configuration, the method of adding objects is the same.

For the default RPDO 0x1603 configuration, there are 2 default mapped objects, as shown in the figure below, which are 0x3020:01 Operation Command, 0x3020:02 Speed Setpoint Command.

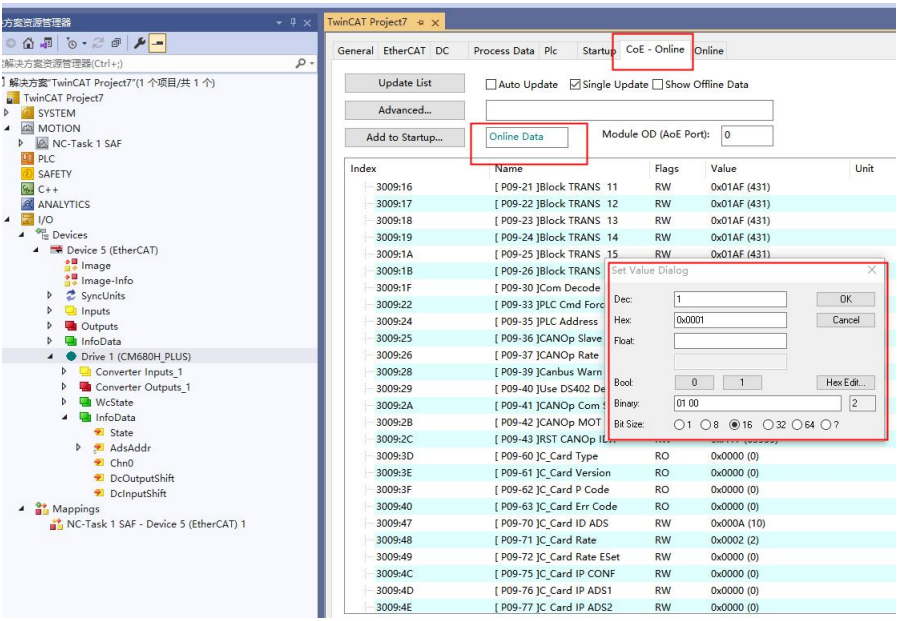



For the 0x1603 with default configuration, it is not recommended that users modify the first 2 configured objects, but users can add objects they wish to increase after these, using the same method as configuring 0x1A01.

### 5.4.3. Configure Non-periodic Communication

#### 5.4.3.1. Non-periodic COE-Online Acquisition

After the user activates to PreOP and higher states, they can monitor data in real-time through the COE-Online SDO data list, and can also modify SDO data by double-clicking the object dictionary.

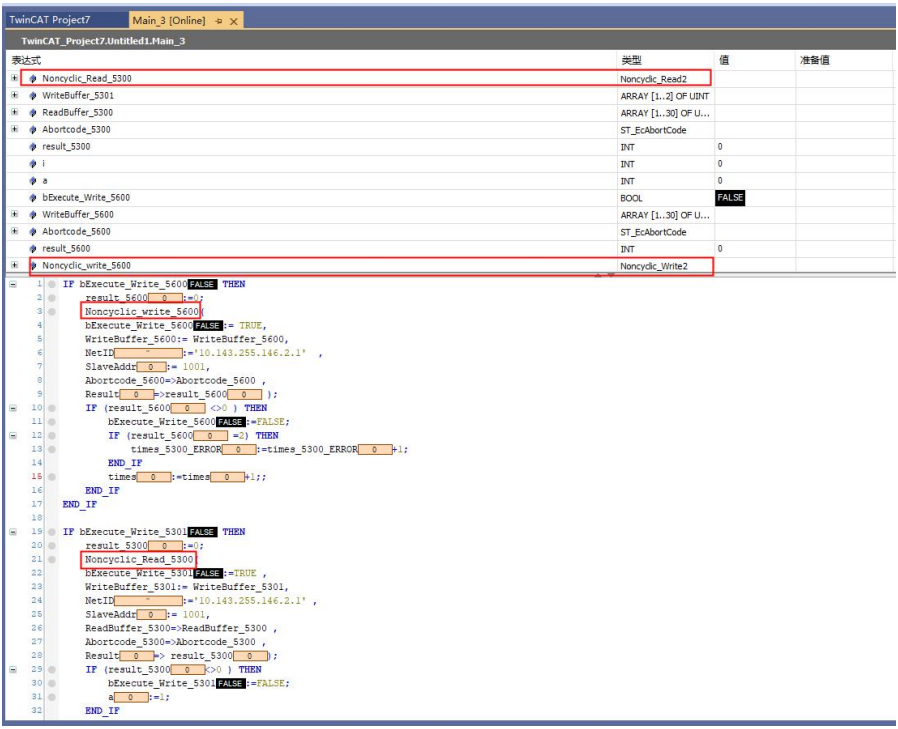


### 5.4.3.2. Custom object dictionary for non-cyclic read/write

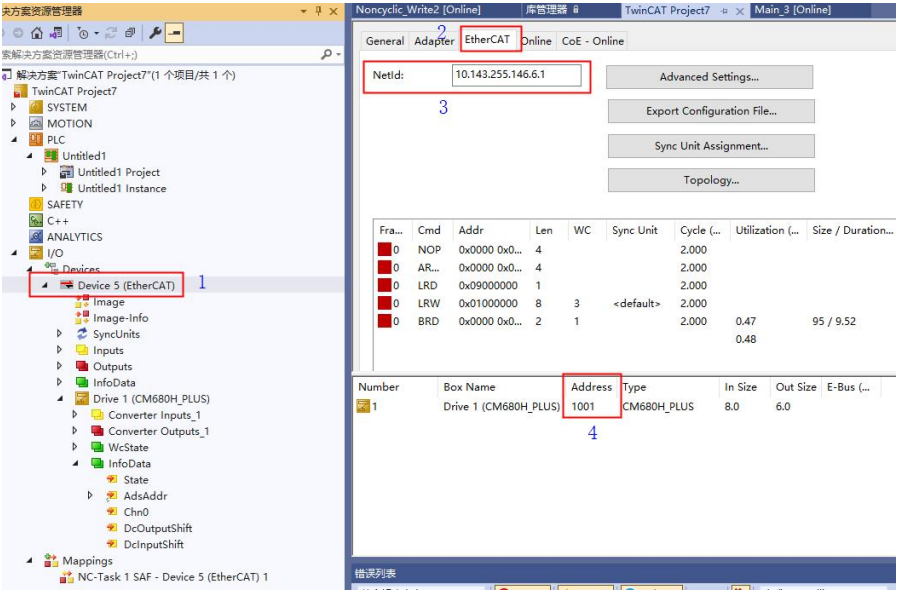
1. Non-cyclic write

Section 5.3.3.2 has already introduced using a custom object dictionary to operate multiple consecutive registers of the inverter, and now the specific configuration steps are demonstrated.

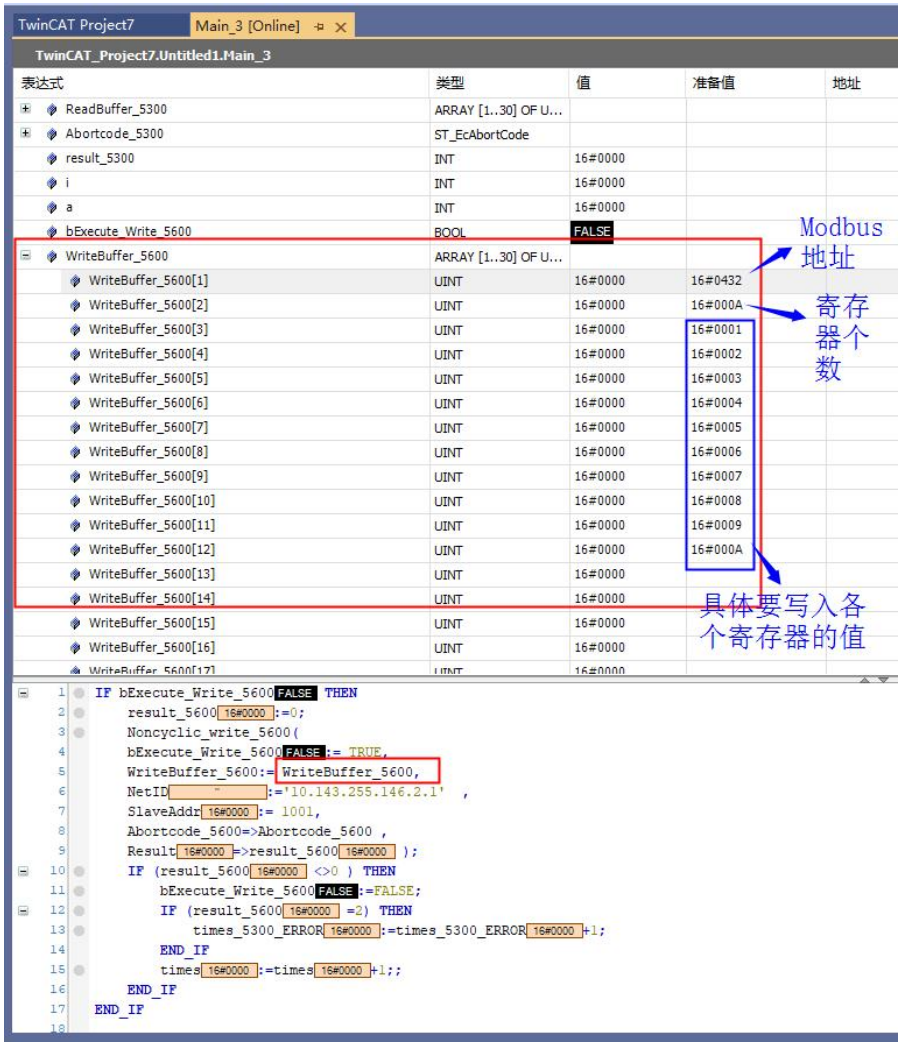
For the convenience of users, two non-cyclic read/write function blocks that can be used directly are encapsulated: the read function block is Noncyclic_Read2, and the write function block is Noncyclic_Write2. Both function blocks include the capability to obtain SDO transmission error codes. Each execution of a non-cyclic read or write will retrieve an SDO transmission error code, and if there is no error, the error code will return 0. The example code after calling these two function blocks is shown in the figure below, where the red box indicates the instantiation and invocation of the function blocks.



The variables bExecute_Write_5600 and bExecute_Write_5301 indicate whether to perform a non-periodic read. The values of sNetID and nSlaveAddr in the function block need to be entered according to the actual operation of EtherCAT, as shown in the figure below:

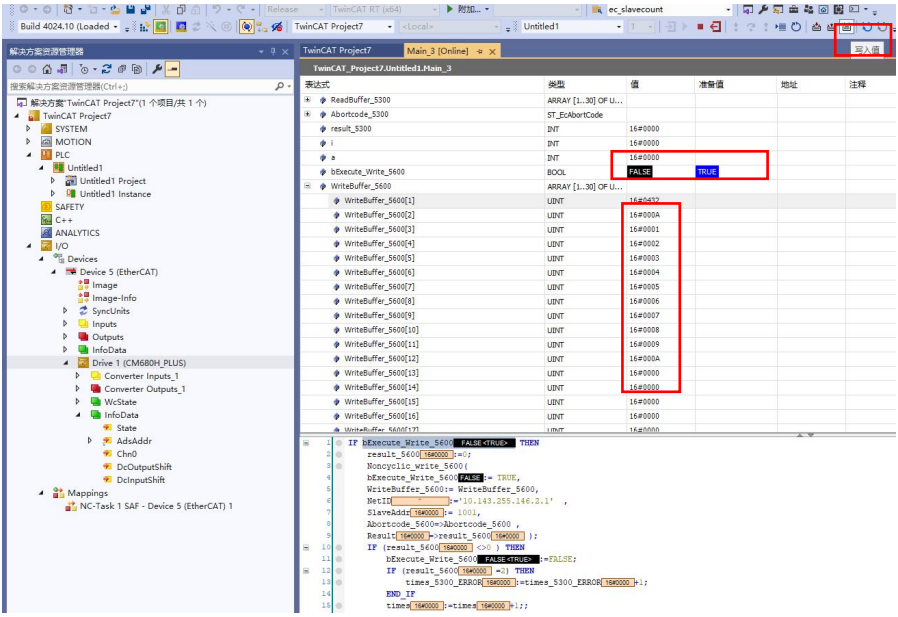


The array WriteBuffer_5600 is used to write the starting address of the Modbus to be written, the number of registers to be written, and the values of each register to be written, as shown in the figure below.

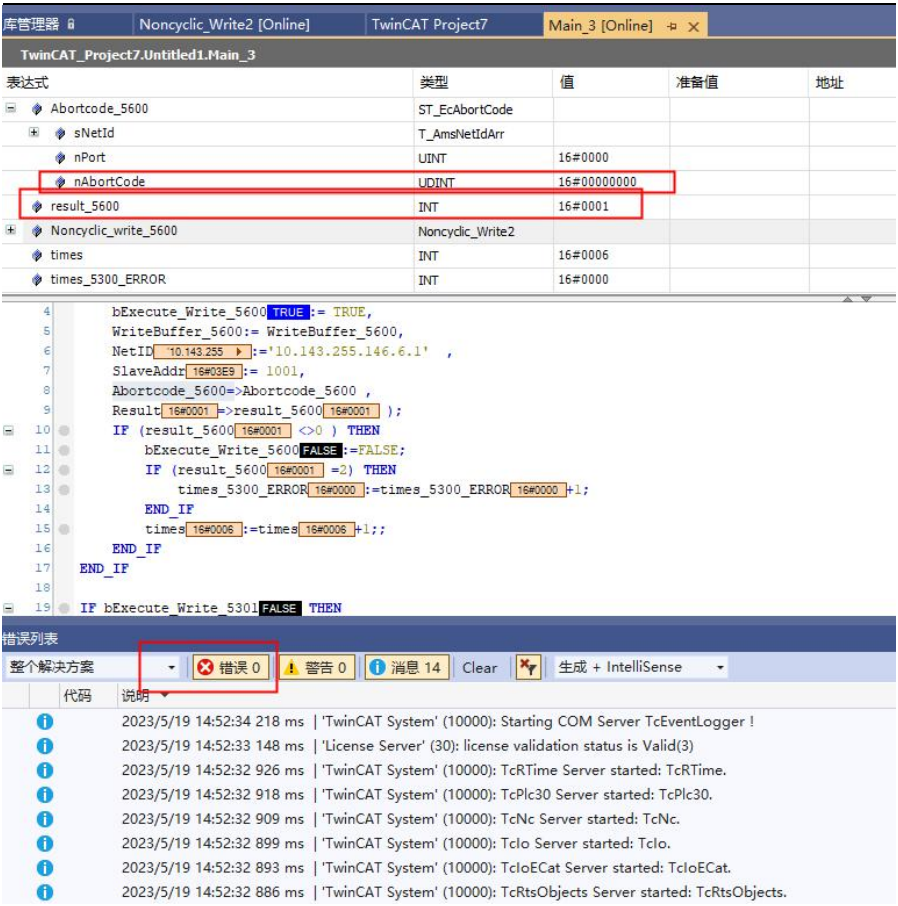



When the values are ready, they are written, and bExecute_Write_5600 is set to True, which will execute a non-periodic write operation. You can check the values returned by the function block's

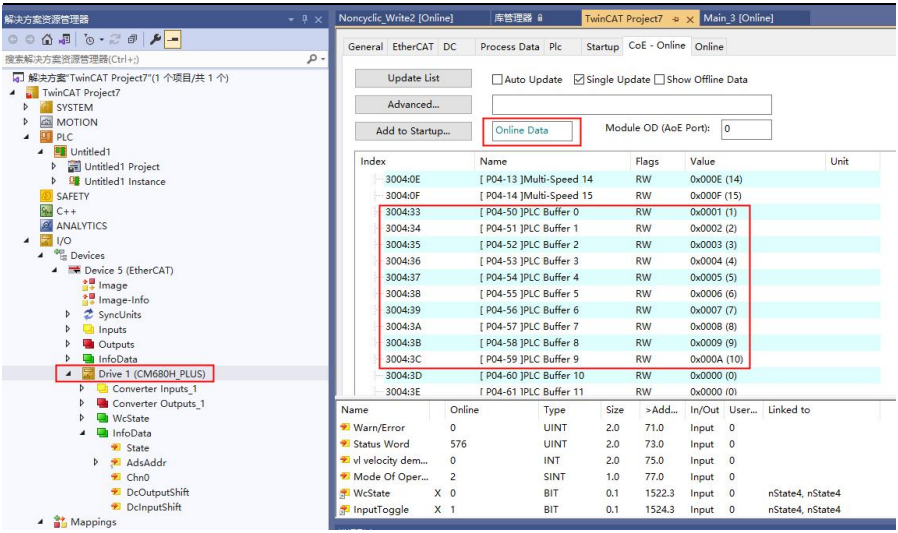
Result and Abortcode_5600, as well as the error list below, to determine if the aperiodic write was successful. If Result equals 1 and nAbortCode under Abortcode_5600 equals 0, then the aperiodic write was successful. If Result equals 2, it indicates that an SDO transfer error code has been obtained. The specific error can be seen in the nAbortCode value under Abortcode_5600, which is explained in Table 3.1. If Result equals 3, it indicates an ADS error. For detailed information on ADS errors, refer to the TwinCAT 3 help documentation.



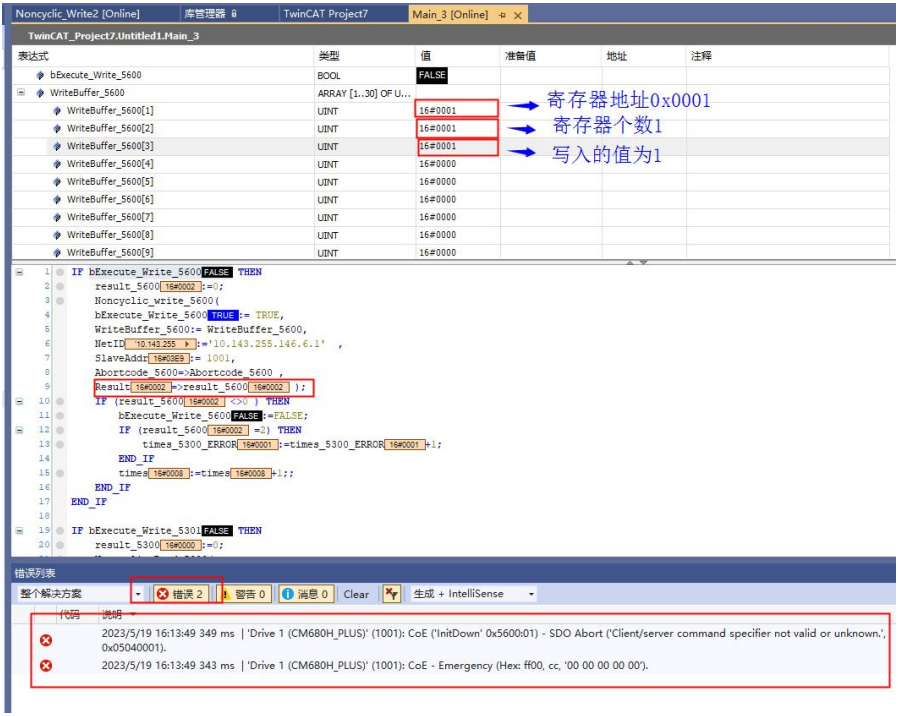

In the example, result_5600 equals 1, nAbortCode equals 0, and there are no errors listed below, so the aperiodic write was successful.

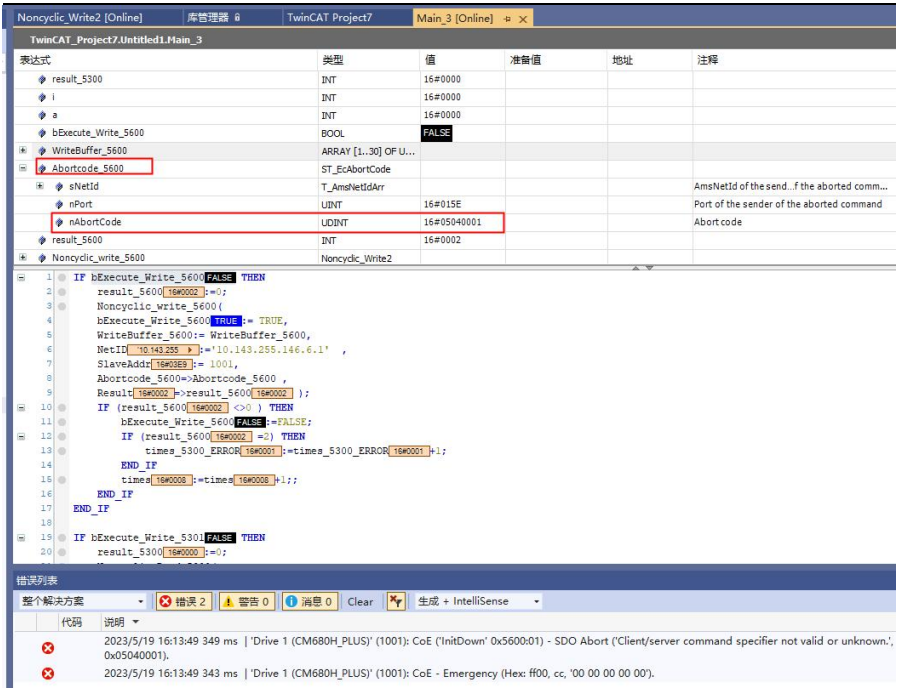



We can also switch to COE-Online to confirm that the values starting from register 0x0432 have indeed been changed to the values 1~10 that we previously wrote.



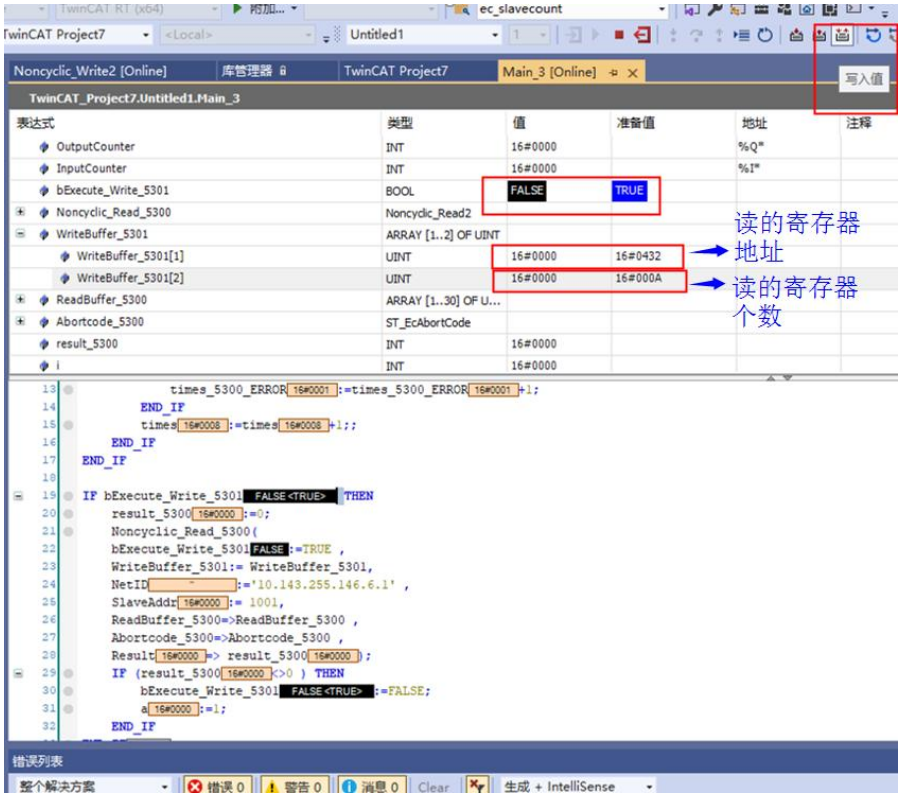

If a non-cyclical write operation is performed on a read-only register address 0x0001, it will return an SDO transfer error code and an Emergency error. As shown in the following two figures.



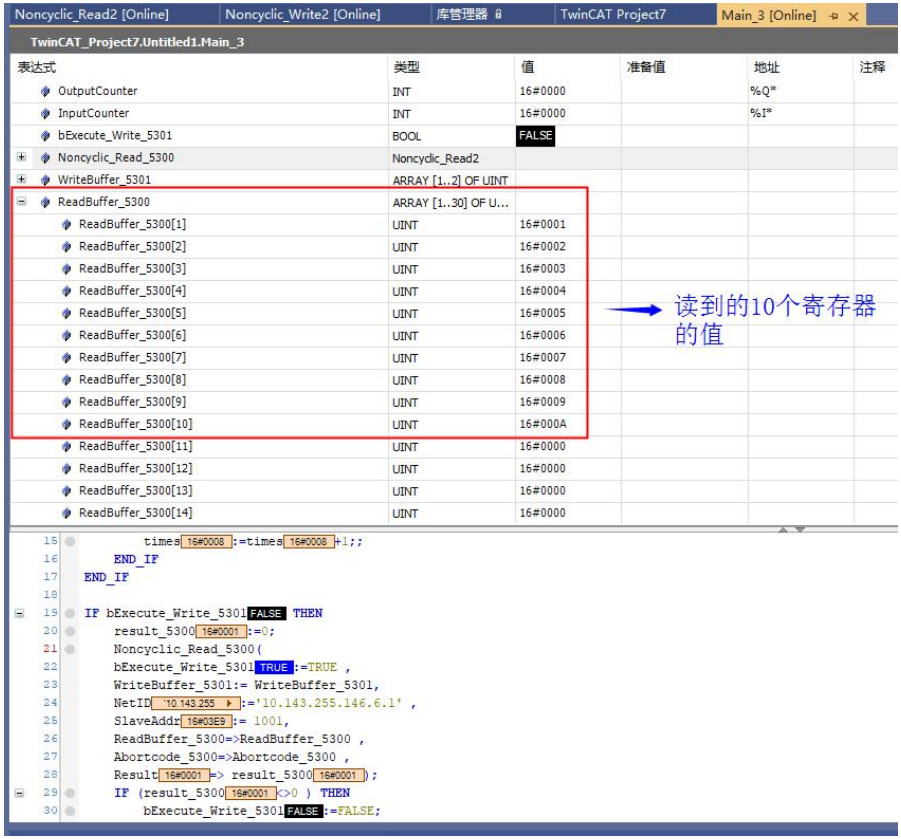

Noncyclic_Write2 [Online]
库管理器
TwinCAT Project7
Main_3 [Online]
TwinCAT_Project7.Untitled1.Main_3
表达式
类型
值
准备值
地址
注释
result_5300
INT
16#0000
i
INT
16#0000
a
INT
16#0000
bExecute_Write_5600
BOOL
FALSE
WriteBuffer_5600
ARRAY [1..30] OF U...
Abortcode_5600
ST_EcAbortCode
sNetId
T_AmsNetIdArr
AmsNetId of the send...f the aborted comm...
nPort
UINT
16#015E
Port of the sender of the aborted command
nAbortCode
UDINT
16#05040001
Abort code
result_5600
INT
16#0002
Noncyclic_write_5600
Noncyclic_Write2
IF bExecute_Write_5600 FALSE THEN
result_5600 16#0002 :=0;
Noncyclic_write_5600(
bExecute_Write_5600 TRUE := TRUE,
WriteBuffer_5600:= WriteBuffer_5600,
NetID '10.143.255 ':='10.143.255.146.6.1' ,
SlaveAddr 16#03E9 := 1001,
Abortcode_5600=>Abortcode_5600 ,
Result 16#0002 =>result_5600 16#0002 );
IF (result_5600 16#0002 <>0 ) THEN
bExecute_Write_5600 FALSE :=FALSE;
IF (result_5600 16#0002 =2) THEN
times_5300_ERROR 16#0001 :=times_5300_ERROR 16#0001 +1;
END_IF
times 16#0008 :=times 16#0008 +1;;
END_IF
END_IF
IF bExecute_Write_5301 FALSE THEN
result_5300 16#0000 :=0;
错误列表
整个解决方案
错误 2
警告 0
消息 0
Clear
生成 + IntelliSense
代码
说明
2023/5/19 16:13:49 349 ms | 'Drive 1 (CM680H_PLUS)' (1001): CoE ('InitDown' 0x5600:01) - SDO Abort ('Client/server command specifier not valid or unknown.', 0x05040001).
2023/5/19 16:13:49 343 ms | 'Drive 1 (CM680H_PLUS)' (1001): CoE - Emergency (Hex: ff00, cc, '00 00 00 00 00').

2. Non-cyclical Read

Write the parameters for non-cyclical read into the WriteBuffer_5301 array, as shown in the figure below. The register address to be read is the previously non-cyclical written address 0x0432, and the number of registers to be read is 10. Then set bExecute_Write_5301 to TRUE, write the value, and a non-cyclical read will be executed once.



If the non-cyclical read is executed successfully, the values of the 10 read register addresses will be saved in the ReadBuffer_5300 array, as shown in the figure below.




The method to determine whether the non-cyclical read is executed successfully is the same as the method for non-cyclical write, by checking the values of the Result and Abortcode_5300 returned by the function block, as well as the error list below. If there is an SDO transfer error code for non-periodic reads, the method to check it is the same as for non-periodic writes.

3. Explanation of Encapsulated Non-periodic Read/Write Function Blocks

To allow users to perform non-periodic read/write operations without needing to understand Modbus messages, the manufacturer has encapsulated two non-periodic read/write function blocks for user reference.

Noncyclic_Write2 function block: implements non-periodic writing, where bExecute_Write_5600: can always be set to TRUE, WriteBuffer_5600: configures the register, including the register address, number of registers, and the value to be written to the register, NetID: the AMS route of the EtherCAT master station, SlaveAddr: the station address of the slave station, Abortcode_5600: the SDO transfer error code returned, Result: the execution result (1 indicates successful execution, 2 indicates an SDO transfer error code was obtained, 3 indicates an ADS error occurred while obtaining the SDO transfer code).

```
IF bExecute_Write_5600 THEN
    result_5600:=0;
    Noncyclic_write_5600(
    bExecute_Write_5600:= TRUE,
    WriteBuffer_5600:= WriteBuffer_5600,
    NetID:='10.143.255.146.6.1'  ,
    SlaveAddr:= 1001,
    Abortcode_5600=>Abortcode_5600 ,
    Result=>result_5600 );
    IF (result_5600 <>0 ) THEN
        bExecute_Write_5600:=FALSE;
        IF (result_5600 =2) THEN
            times_5300_ERROR:=times_5300_ERROR+1;
        END_IF
        times:=times+1;;
    END_IF
END_IF
```

Noncyclic_Read2 function block: a function block that implements non-periodic reading, where bExecute_Write_5301: can always be set to TRUE, WriteBuffer_5301: configures the register, including the register address and number of registers, NetID: the AMS route of the EtherCAT master station, SlaveAddr: the station address of the slave station, ReadBuffer_5300: the value read from the register, Abortcode_5300: the SDO transfer error code returned, Result: the execution result (1 indicates successful execution, 2 indicates an SDO transfer error code was obtained, 3 indicates an ADS error occurred while obtaining the SDO transfer code).

```
IF bExecute_Write_5301 THEN
    result_5300:=0;
    Noncyclic_Read_5300(
    bExecute_Write_5301:=TRUE ,
    WriteBuffer_5301:= WriteBuffer_5301,
    NetID:='10.143.255.146.6.1' ,
    SlaveAddr:= 1001,
    ReadBuffer_5300=>ReadBuffer_5300 ,
    Abortcode_5300=>Abortcode_5300 ,
    Result=> result_5300);
    IF (result_5300<>0 ) THEN
        bExecute_Write_5301:=FALSE;
        a:=1;
    END_IF
END_IF
```

## 5.5. Fault Information

### 5.5.1. Periodic Fault Code

In the periodic communication PDO area, PDO1 sent from the slave station to the master station indicates error and alarm information. Users can obtain the latest status through periodic communication. In addition to the diagnostic information of the inverter itself, the communication status between the module and the inverter must also be included, allowing users to obtain the cause of errors. Detailed fault status codes are listed in the table below:

**Table 39 Periodic Fault Codes**

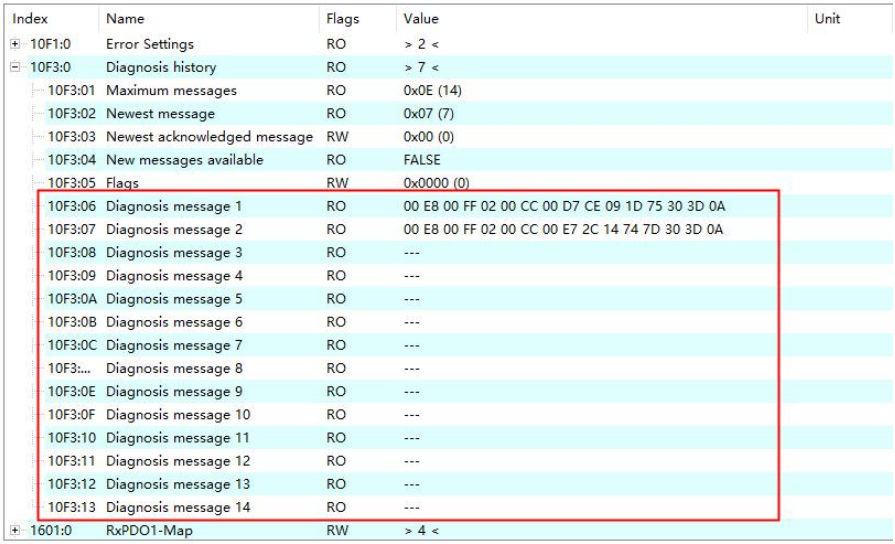
| Diagnostic Classification | Status Code |
|---|---|
| Inverter Body Diagnostic Information (Register Address: 0x2100) | 200 Communication Error (Message Abnormality)<br>201 Communication Error (Timeout Abnormality)<br>202 Address Error<br>203 Parameter Exceeds Threshold<br>204 Incorrect Slave Station Number<br>205 Incorrect Periodic Data Value<br>206 SPI Transmission Without Memory<br>207 SPI Transmission Error<br>208 Periodic Communication Error |

### 5.5.2. Non-periodic Fault Code

When executing non-periodic communication, the error code of SDO transmission will be returned. For details on error codes, see Table 38.

### 5.5.3. Retrieve Diagnostic Information

When communication between the module and the inverter is abnormal, diagnostic information is sent to the PLC. Diagnostic information can be obtained through object dictionary 0x10F3. As shown in the figure below, up to the latest 14 diagnostic messages can be displayed.



| Index | Name | Flags | Value | Unit |
|---|---|---|---|---|
| 10F1:0 | Error Settings | RO | > 2 < | |
| 10F3:0 | Diagnosis history | RO | > 7 < | |
| 10F3:01 | Maximum messages | RO | 0x0E (14) | |
| 10F3:02 | Newest message | RO | 0x07 (7) | |
| 10F3:03 | Newest acknowledged message | RW | 0x00 (0) | |
| 10F3:04 | New messages available | RO | FALSE | |
| 10F3:05 | Flags | RW | 0x0000 (0) | |
| 10F3:06 | Diagnosis message 1 | RO | 00 E8 00 FF 02 00 CC 00 D7 CE 09 1D 75 30 3D 0A | |
| 10F3:07 | Diagnosis message 2 | RO | 00 E8 00 FF 02 00 CC 00 E7 2C 14 74 7D 30 3D 0A | |
| 10F3:08 | Diagnosis message 3 | RO | --- | |
| 10F3:09 | Diagnosis message 4 | RO | --- | |
| 10F3:0A | Diagnosis message 5 | RO | --- | |
| 10F3:0B | Diagnosis message 6 | RO | --- | |
| 10F3:0C | Diagnosis message 7 | RO | --- | |
| 10F3:... | Diagnosis message 8 | RO | --- | |
| 10F3:0E | Diagnosis message 9 | RO | --- | |
| 10F3:0F | Diagnosis message 10 | RO | --- | |
| 10F3:10 | Diagnosis message 11 | RO | --- | |
| 10F3:11 | Diagnosis message 12 | RO | --- | |
| 10F3:12 | Diagnosis message 13 | RO | --- | |
| 10F3:13 | Diagnosis message 14 | RO | --- | |
| 1601:0 | RxPDO1-Map | RW | > 4 < | |

The format of the diagnostic information is as follows:

**Table 40 Diagnostic Information Format**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Example | 00 | E8 | 10 | FF | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Purpose | (L) (H) (Fixed Value) | | (L) (H) Error code | | (L) (H) (Fixed Value) | | (L) (H) Text ID | | (L) (H) (Fixed Value) | | | | | | | |
| | Diag code | | | | Flags | | Text ID | | Time stamp | | | | | | | |

Diag code … diagnostic code for identifying information

Error code diagnostic code as defined.

Flags … value is fixed at 0002h.

Text ID … Returns the Text ID defined for each error code.

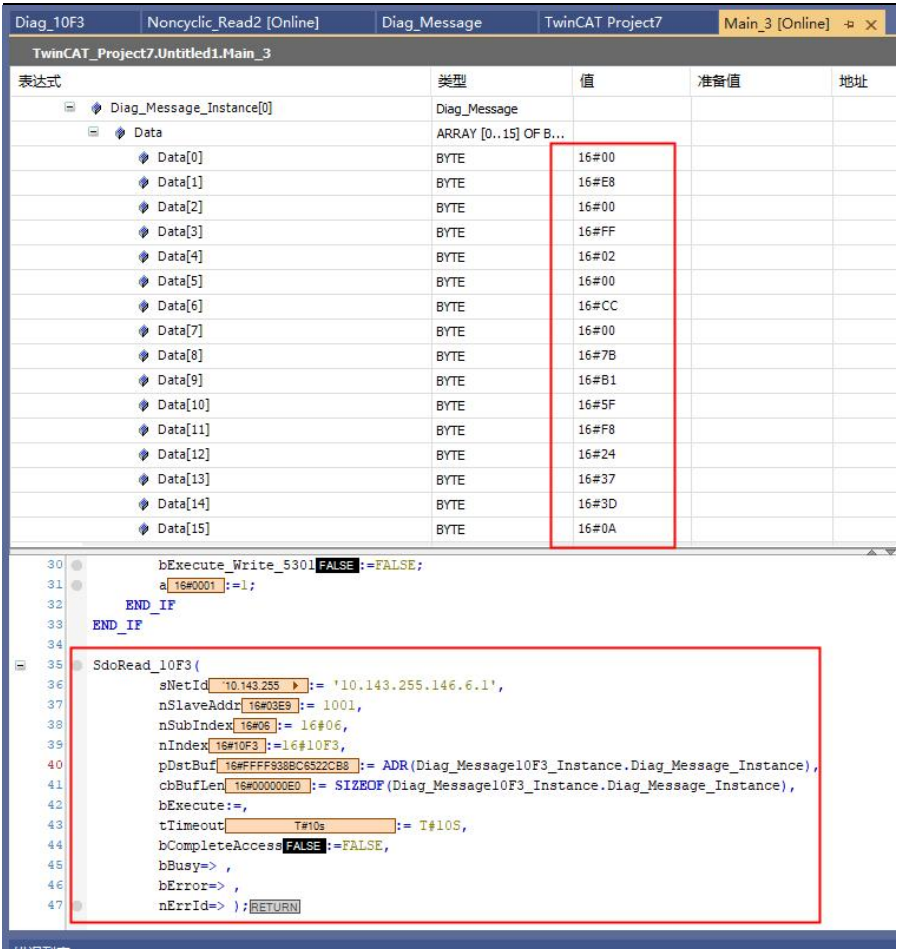
Set the upper 8-bit as the main alarm number and the lower 8-bit as the auxiliary alarm number.

Time stamp … The time when the error occurred.

### 5.5.4. Non-periodic Read or Erase Fault Code

1. Fault code read

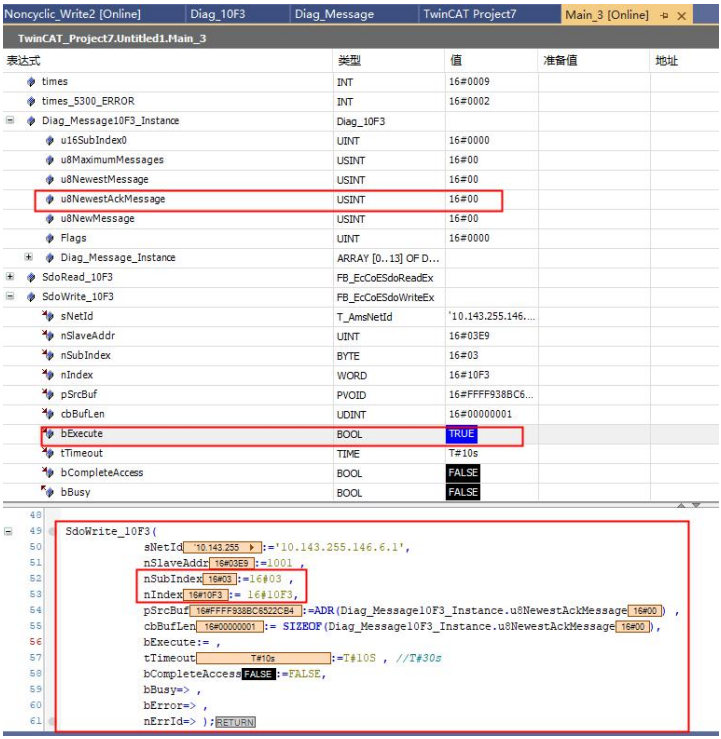
Read the 0x10F3 object using the FB_EcCoESdoReadEx function block provided by Twincat3 or obtain it directly via COE-Online. The following figure shows the consistent results obtained through FB_EcCoESdoReadEx and COE-Online. Note that complete access is not supported for operations on 0x10F3.


Diag_10F3
Noncyclic_Read2 [Online]
Diag_Message
TwinCAT Project7
Main_3 [Online]
TwinCAT_Project7.Untitled1.Main_3
表达式
类型
值
准备值
地址
Diag_Message_Instance[0]
Diag_Message
Data
ARRAY [0..15] OF B...
Data[0] BYTE 16#00
Data[1] BYTE 16#E8
Data[2] BYTE 16#00
Data[3] BYTE 16#FF
Data[4] BYTE 16#02
Data[5] BYTE 16#00
Data[6] BYTE 16#CC
Data[7] BYTE 16#00
Data[8] BYTE 16#7B
Data[9] BYTE 16#B1
Data[10] BYTE 16#5F
Data[11] BYTE 16#F8
Data[12] BYTE 16#24
Data[13] BYTE 16#37
Data[14] BYTE 16#3D
Data[15] BYTE 16#0A
bExecute_Write_5301 FALSE :=FALSE;
a 16#0001 :=1;
END_IF
END_IF
SdoRead_10F3(
sNetId '10.143.255 :='10.143.255.146.6.1',
nSlaveAddr 16#03E9 := 1001,
nSubIndex 16#06 := 16#06,
nIndex 16#10F3 :=16#10F3,
pDstBuf 16#FFFF938BC6522CB8 := ADR(Diag_Message10F3_Instance.Diag_Message_Instance),
cbBufLen 16#000000E0 := SIZEOF(Diag_Message10F3_Instance.Diag_Message_Instance),
bExecute:=,
tTimeout T#10s := T#10S,
bCompleteAccess FALSE :=FALSE,
bBusy=> ,
bError=> ,
nErrId=> );RETURN

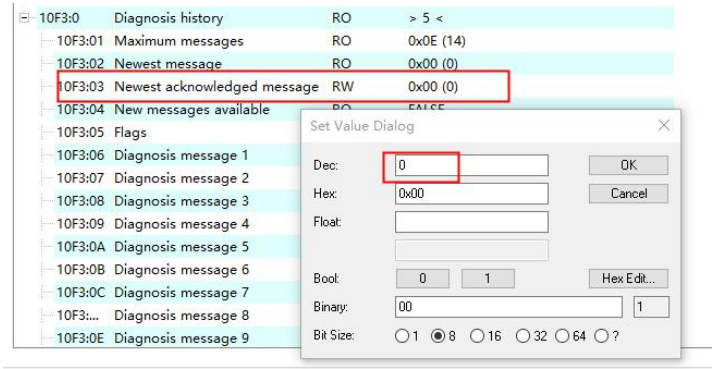
| | | | |
|---|---|---|---|
| 10F3:0 | Diagnosis history | RO | > 6 < |
| 10F3:01 | Maximum messages | RO | 0x0E (14) |
| 10F3:02 | Newest message | RO | 0x06 (6) |
| 10F3:03 | Newest acknowledged message | RW | 0x00 (0) |
| 10F3:04 | New messages available | RO | FALSE |
| 10F3:05 | Flags | RW | 0x0000 (0) |
| 10F3:06 | Diagnosis message 1 | RO | 00 E8 00 FF 02 00 CC 00 7B B1 5F F8 24 37 3D 0A |
| 10F3:07 | Diagnosis message 2 | RO | --- |
| 10F3:08 | Diagnosis message 3 | RO | --- |
| 10F3:09 | Diagnosis message 4 | RO | --- |
| 10F3:0A | Diagnosis message 5 | RO | --- |
| 10F3:0B | Diagnosis message 6 | RO | --- |
| 10F3:0C | Diagnosis message 7 | RO | --- |
| 10F3:... | Diagnosis message 8 | RO | --- |

2. Fault Code Erasure

Write 0 to sub-index 3 of 0x10F3 using the FB_EcCoESdoWriteEx function block provided by Twincat3 to clear all diagnostic information, as shown in the following figure:



Or write 0 directly to sub-index 3 of 0x10F3 via COE-Online, as shown in the following figure:



# 6. CANopen Communication

## 6.1. Product Information

The EMH-OP Communication Expansion Card is a CANopen fieldbus adapter card that complies with the CiA standard. This card is installed on the CM680 Inverter to enable communication with CANopen master station devices, making the inverter a slave node of the CANopen network, accepting control from the master station.

### 6.1.1. Physical Dimensions

Physical dimensions of the CANopen module: PCB length 95mm, width 37mm, mounting hole diameter 3.5mm.

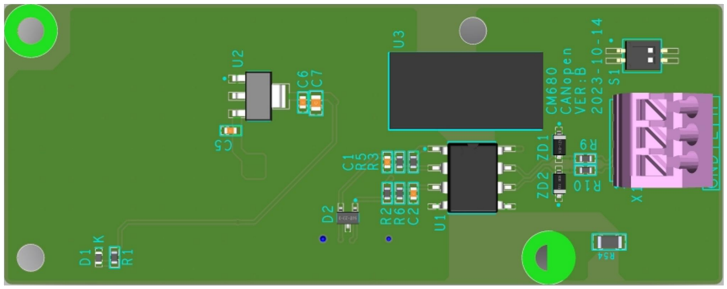



**Figure 23 Physical Dimensions of the EMH-OP Communication Expansion Card**

### 6.1.2. Interface Layout

The hardware layout of the EMH-OP Communication Expansion Card is shown in Figure 24. Connector X2 is used for connecting to the inverter and is located on the back of the EMH-OP Communication Expansion Card. The EMH-OP Communication Expansion Card provides spring-type terminal blocks for communication connections. Detailed descriptions of each hardware component are provided in Table 41.

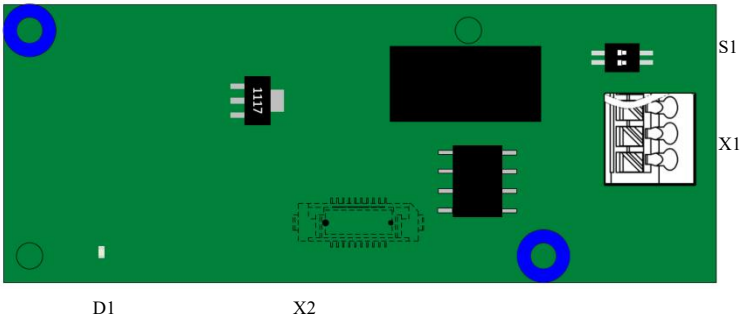



**Figure 24 EMH-OP Communication Expansion Card Interface Layout**

**Table 41 EMH-OP Communication Expansion Card Interface Description**

| Diagram Name | Hardware Name | Function Description |
|---|---|---|
| X1 | CAN Socket | For module connection to the CANopen network, refer to Figure 1.3 for details |
| X2 | Board-to-Board Socket | For connection to the inverter (located on the back of the board) |
| D1 | LED Indicator Light | For indicating power status, refer to Table 1.2 for details |
| S1 | Termination Resistor Setting | For setting the CAN bus termination resistor, refer to Table 1.6 for details |

1. CANopen communication interface

The EMH-OP communication expansion card uses a spring-type terminal block X1 to connect to the CAN bus, and the shield is twisted together and connected to the PE (J703-1) of the main control board. If conditions permit, the shield of the communication line can also be connected to the cabinet metalwork. The pin signal definitions are shown in the figure below:

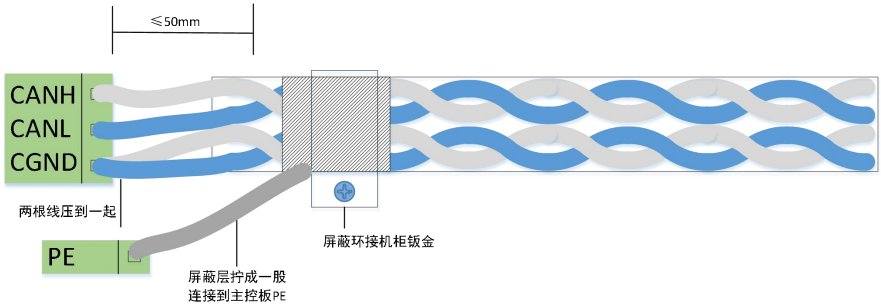



**Figure 25 EMH-OP Communication Expansion Card CAN Communication Interface Description**

2. LED Indicator Light

**Table 42 EMH-OP Communication Expansion Card LED Indicator Light Description**

| Diagram Name | Indicator Light | Indicator Light Status | | Function Description |
|---|---|---|---|---|
| D1 | Hardware power indicator light | Green light | Constantly On | Module powered on normally |
| | | | Off | Module not powered on |

3. Terminal Impedance

The DIP switch S1 on the EMH-OP Communication Expansion Card is a 2-position DIP switch used to configure the CAN bus terminal resistor. It is recommended to set the terminal resistor at both ends of the network topology. Switching to “ON” represents “1”, and switching down represents “0”. For detailed information, see Table 15.

**Table 43 EMH-OP Communication Expansion Card Terminal Resistor Settings**

| DIP Switch Number | | Terminal Impedance Setting |
|---|---|---|
| 1 | 2 | |
| 0 | 0 | Do Not Use Terminal Resistor |
| 1 | 1 | Use Terminal Resistor |

## 6.2. Installation and Wiring

### 6.2.1. Installation

The EMH-OP Communication Expansion Card is designed to be embedded in the CM680 Inverter. Before installation, please disconnect the power supply to the inverter, wait for about 10 minutes until the charging indicator light is completely off before proceeding with the installation. After inserting the EMH-OP Communication Expansion Card into the inverter, please secure the corresponding screws to prevent poor contact of the board-to-board signal sockets. The installation diagram is shown in Figure 2.1.

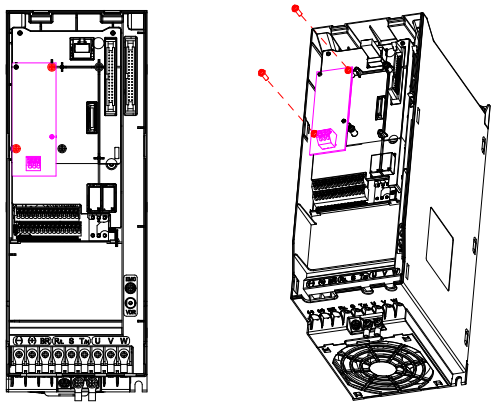


**Figure 26 Schematic Diagram of EMH-OP Communication Expansion Card Installation**

### 6.2.2. Wiring

The wiring diagram for the CAN bus is shown in Figure 2.2. The CAN bus is recommended to use shielded twisted pair cables. Terminal matching resistors should be connected at both ends of the CAN bus. Adjust the DIP switches according to the indications on the terminal blocks. Failure to connect or insufficiently connect the terminal resistors can affect communication quality and cause unstable communication.

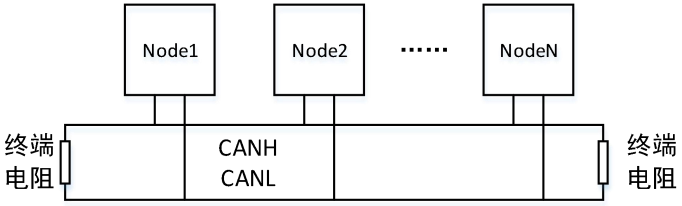



**Figure 27 CAN Bus Wiring Diagram**

## 6.3. Communication Description

### 6.3.1. Key Points of CANopen Communication

The CANopen communication sub-protocol CiA 301 (DS 301) includes periodic communication and event-triggered communication.

The CANopen device sub-protocol defines direct access to inverter parameters and key process data.

The inverter's CANopen interface supports the CiA 402 (DS 402, inverter and motion control sub-protocol) specification and the inverter manufacturer's custom specifications.

The maximum transmission distance of CANopen is limited by the transmission speed; the higher the speed, the shorter the communication distance.

Theoretically, up to 127 communication nodes can be connected in a CAN network; in practice, the number of nodes is limited by the transmission capability of the CAN transceiver.

Unless otherwise specified, the inverter is defined as a slave node (Server) in the CAN network.

**Table 44 CANopen Protocol**

| | | |
|---|---|---|
| Application layer | CANopen Application layer | CiA 402<br>CiA 301 |
| Presentation layer | | |
| Session layer | | |
| Transport layer | | |
| Network layer | | |
| Data link layer | | CAN2.0A/B |
| Physical layer | | ISO 11898 |

**Table 45 CANopen Parameter Settings**

| Name | Description | Default Value |
|---|---|---|
| F8-07 Communication Decoding Method | Select manufacturer-defined protocol, 0 indicates custom protocol one, 1 indicates custom protocol two | 1 |
| F8-14 CANopen Node Address | Set node address, each device in the same network must have a unique node address. 0: Disable CANopen Slave Station, 1~127: Slave Node Address Number, a single CAN network supports up to 127 slave nodes. | 0 |
| F8-15 CAN Bus Communication Speed | User selects CANopen communication speed, all devices in the same network must use the same communication speed. | 0 |
| F8-18 CANopen Warning Record | Displays inverter CANopen warning information | Read-Only |
| F8-19 CiA402 Protocol Selection | Used to select CiA 402 protocol and manufacturer-defined protocol for CANopen index parsing. 0: Inverter manufacturer-defined protocol, 1: CiA 402 protocol | 1 |

| Name | Description | Default Value |
|---|---|---|
| F8-07 Communication Decoding Method | Select manufacturer-defined protocol, 0 indicates custom protocol one, 1 indicates custom protocol two | 1 |
| F8-20 CANopen communication status | Display inverter CANopen node status | Read-Only |
| F8-21 CiA402 operation status | Display inverter motor operation status | Read-Only |
| F8-22 CANopen index reset | CANopen index data reset command | 0xFFFF |

### 6.3.2. CANopen communication protocol

The CANopen communication protocol includes NMT (Network Management Object), SYNC (Synchronous Object), SDO (Service Data Object), PDO (Process Data Object, including transmitting TPDO, Tx PDO and receiving RPDO, Rx PDO, transmission and reception are referenced to the CANopen node itself), and EMCY (Emergency Object).

**Table 46 Communication Protocol**

| Communication Object | Function Code (binary) | COB-ID(hex) |
|---|---|---|
| NMT | 0000 | 00 |
| SYNC | 0001 | 80 |
| EMCY | 0001 | 81 ... FF |
| TPDO1 | 0011 | 181 ... 1FF |
| RPDO1 | 0100 | 201 ... 27F |
| TPDO2 | 0101 | 281 ... 2FF |
| RPDO2 | 0110 | 301 ... 37F |
| TPDO3 | 0111 | 381 ... 3FF |
| RPDO3 | 1000 | 401 ... 47F |

| Communication Object | Function Code (binary) | COB-ID(hex) |
|---|---|---|
| TPDO4 | 1001 | 481 ... 4FF |
| RPDO4 | 1010 | 501 ... 57F |
| TSDO | 1011 | 581 ... 5FF |
| RSDO | 1100 | 601 ... 67F |

COB-ID encoding format is as follows:

**Table 47 COB-ID (CAN Object ID, 11 or 29 bits)**

| Function Code | | | | Node ID | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

#### 6.3.2.1. NMT

NMT follows the Master/Slave protocol model, where the NMT host can unconditionally control the state transitions of the slave. NMT management involves six states from the start of power-up of a CANopen node, including Initialization, Application Layer Reset, Communication Reset, Pre-Operational Mode, Operational Mode, and Stop Mode. The data field of a CAN communication frame corresponding to an NMT message is 2 bytes long, and the ID field is 0.

**Table 48 NMT Message**

| ID field | Data field (Byte 0) | Data field (Byte 1) |
|---|---|---|
| 0 | NMT CMD | Node ID |

If the Node ID is 0, it indicates that all NMT broadcast frames will be responded to by all slave stations. The NMT CMD list is as follows:

**Table 49 NMT CMD**

| NMT CMD | Meaning |
|---|---|
| 01h | Start Remote Node |
| 02h | Stop Remote Node |
| 80h | Enter Pre-Operation Mode |
| 81h | Node Reset |
| 82h | Communication Reset |

The list of node statuses is as follows:

**Table 50 Node Status**

| Node Status Word | Meaning |
|---|---|
| 00h | Boot-up |
| 04h | Stop State |
| 05h | Operation Mode |
| FFh | Pre-Operation Mode |

The CANopen state machine is as follows:

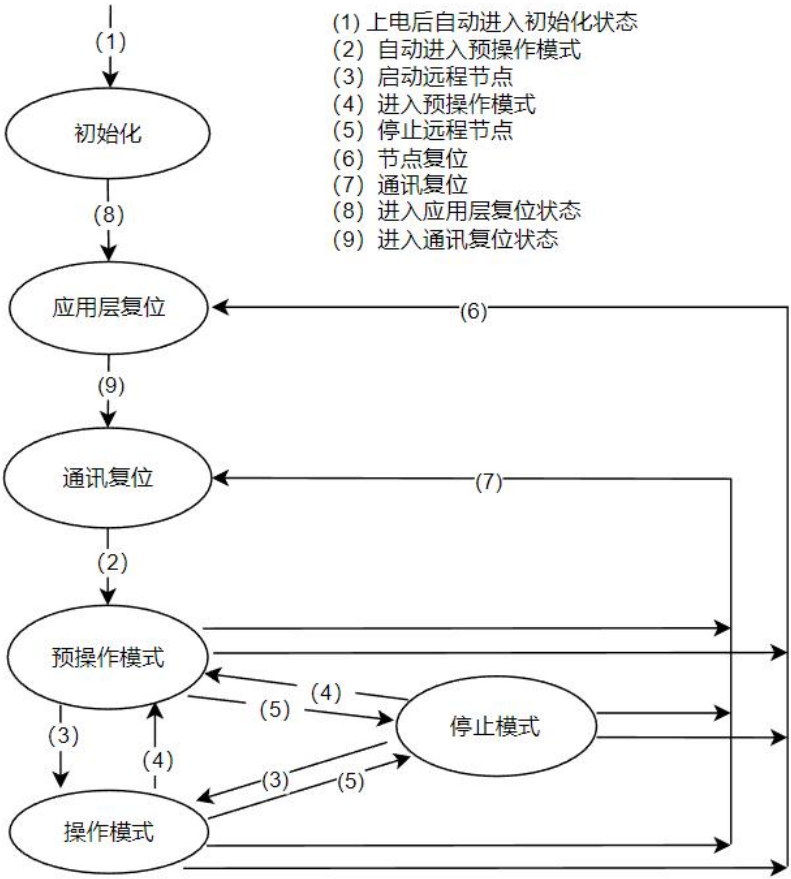



**Figure 28 CANopen State Machine**

The effective working states of each communication protocol object are shown in the following table:

**Table 51 Effective Working States**

| | Initialization | Pre-Operation Mode | Operation Mode | Stop State |
|---|---|---|---|---|
| PDO | \ | \ | Active | \ |
| SDO | \ | Active | Active | \ |
| SYNC | \ | Active | Active | \ |
| EMCY | \ | Active | Active | \ |
| NMT | \ | Active | Active | Active |
| Boot-up | Active | \ | \ | \ |

### 6.3.2.2. SDO

SDO follows the Client/Server protocol model, used for transmitting data with less stringent timing requirements (parameter setting), data length 0~4 bytes. Common SDO protocol is Fast SDO (SDO command byte bit1 set to 1 indicates Fast SDO), meaning all data is transmitted during initialization (initial SDO upload/download). If the transmission length exceeds 4 bytes, Fast SDO cannot be used, and standard SDO must be used for segmented transmission. The object dictionary is a group object of the CANopen node, each node has its own object dictionary, which contains multiple parameters, describing the supported parameter attributes and values. SDO accesses the device object dictionary through index and sub-index, each object has a single index value, and a single index can have multiple sub-index values. The SDO data frame format is as follows:

**Table 52 SDO Data Frame Format**

SDO Data Frame Analysis

| data byte num | Comment | |
|---|---|---|
| byte0 | SDO CMD | |
| byte1 | SDO Index Low Byte | |
| byte2 | SDO Index High Byte | |
| byte3 | SDO sub-index | |
| byte4 | SDO data low word low byte | These 4 bytes are invalid when the SDO client queries the server status |
| byte5 | SDO data low word high byte | |
| byte6 | SDO data high word low byte | |
| byte7 | SDO data high word high byte | |

SDO CMD Analysis

| bit num | Comment |
|---|---|
| bit0 | size |
| bit1 | expedited |
| bit2 | Number of unused bytes in SDO |
| bit3 | |
| bit4 | \ |
| bit5 | cmd<br>1: init download (send parameters)<br>2: init upload (status query)<br>3: stop SDO transfer |
| bit6 | |
| bit7 | |

#### 6.3.2.3. PDO

PDO follows the Producer/Consumer protocol model, where each network node can receive information from the transmitting node and can also determine whether to process the received information, used for transmitting time-critical process data (reference values, control commands, status information, etc.), which can be transmitted one-to-one or one-to-many. PDO transmission can be initiated by an internal timer (asynchronous communication), upon receiving a synchronization frame (synchronous communication), and through remote requests. The CAN interface supports up to 8 PDOs (including 4 TPDOs and 4 RPDOs), with only TPDO1 and RPDO1 enabled by default. PDO mapping can be used to transmit multiple application objects in a single PDO frame. PDO object mapping can only be modified when the node is in pre-operational mode.

- PDO mapping format CANopen object mapping value is in hexadecimal encoding, PDO mapping data format is as follows:

**Table 53 PDO Mapping Data Format**

| Type | MSB | | LSB |
|---|---|---|---|
| UINT32 | 31 16 | 15 8 | 7 0 |
| Description | Index (e.g., 6040h) | Sub-Index (e.g., 02h) | Object bit length (e.g., 10h corresponds to 16 bits, 08h corresponds to 8 bits) |

Note:

1. PDO mapping sub-index starts from 1.

2. If the PDO mapping input value is 0, the mapping of the current sub-index and all subsequent sub-indices will be ignored.

PDO transmission type settings and transmission methods correspond as follows:

**Table 54 PDO Transmission Type**

| Transmission type setting value | Cyclic | Non-cyclic | Synchronous | Asynchronous | Remote Request |
|---|---|---|---|---|---|
| 0 | \ | Active | Active | \ | \ |
| 1~240 | Active | \ | Active | \ | \ |
| 252 | \ | \ | Active | \ | Active |
| 253 | \ | \ | \ | Active | Active |
| 255 | \ | \ | \ | Active | \ |

A transmission type value of 0 indicates that two PDOs transmit synchronous non-cyclic information (currently unsupported). A transmission type value of 1~240 indicates the number of SYNCs between two PDO transmissions. A transmission type value of 252 indicates that data is updated immediately after receiving SYNC (currently unsupported). A transmission type value of 253

indicates that data is updated immediately after receiving RTR (currently unsupported). A transmission type value of 255 indicates asynchronous non-cyclic transmission.

- PDO Mapping Configuration via SDO This section provides an example of configuring a slave station's PDO mapping via the CANopen master station's SDO. Other PDO configurations should refer to this example. The slave station's PDO configuration is as follows:

1. Node ID = 1;
2. RPDO1: 6040h Control Word, 6042h Target Speed;
3. TPDO1 : 6041h Status Word, 6043h Actual Target Speed;
4. TPDO1 is sent to the master station with a 100ms cycle.

In the pre-operational mode of the CANopen slave node, the object configuration is as shown in the following table: RPDO1 mapping configuration steps:

**Table 55 RPDO1 Mapping Configuration Steps**

| Step | Index (hex) | sub-Index (hex) | Name | Write Value （hex） | Description |
|---|---|---|---|---|---|
| 1 | 1400 | 01 | RPDO1 Parameter: COB-ID | 80000201 | Disable RPDO1 |
| 2 | 1400 | 02 | RPDO1 Parameter: Transmission Type | FF | Transmission Type 255 |
| 3 | 1600 | 00 | RPDO1 Mapping: Number of Entries | 0 | Write 0 before Mapping Configuration |
| 4 | 1600 | 01 | RPDO1 Mapping: Mapped Object 1 | 60400010 | 6040h Control Word, Subindex 00, Data Length 16bits |
| 5 | 1600 | 02 | RPDO1 Mapping: Mapped Object 2 | 60420010 | 6042h Target Speed, Subindex 00, Data Length 16bits |
| 6 | 1600 | 00 | RPDO1 Mapping: Number of Entries | 2 | Map 2 Objects |
| 7 | 1400 | 01 | RPDO1 Parameter: COB-ID | 201 | Enable RPDO1 |

TPDO1 Mapping Configuration Steps:

**Table 56 TPDO1 Mapping Configuration Steps**

| Step | Index (hex) | sub-Index (hex) | Name | Write Value （hex） | Description |
|---|---|---|---|---|---|

| Step | Index (hex) | sub-Index (hex) | Name | Write Value （hex） | Description |
|---|---|---|---|---|---|
| 1 | 1800 | 01 | TPDO1 Parameter: COB-ID | 80000183 | Disable TPDO1 |
| 2 | 1800 | 02 | TPDO1 Parameter: Transmission Type | FF | Transmission Type 255 |
| 3 | 1800 | 05 | TPDO1 Parameter: Event Timer | 64 | 100ms |
| 4 | 1A00 | 00 | TPDO1 Mapping: Number of Entries | 0 | Write 0 before Mapping Configuration |
| 5 | 1A00 | 01 | TPDO1 Mapping: Mapped Object 1 | 60410010 | 6041h Status Word, Subindex 00, Data Length 16bits |
| 6 | 1A00 | 02 | TPDO1 Mapping: Mapped Object 2 | 60430010 | 6043h Actual Target Speed, Subindex 00, Data Length 16bits |
| 7 | 1A00 | 00 | TPDO1 Mapping: Number of Entries | 2 | Map 2 Objects |
| 8 | 1800 | 01 | TPDO1 Parameter: COB-ID | 183 | Enable TPDO1 |

#### 6.3.2.4. EMCY

The emergency object is triggered when an internal fault occurs in the CANopen device, informing the CANopen master station of the fault code via an emergency frame. The emergency frame is a diagnostic message and does not affect CANopen communication. Refer to 6. CANopen Fault Codes for CANopen fault codes.

### 6.3.3. Communication Sub-Protocol

The communication sub-protocol specifies the meaning of parameters, control words, status words, and variable values (reference values and actual values) transmitted and received via the CAN bus. The inverter supports two sub-protocols:

- CiA 402. The inverter internally converts the CiA 402 protocol. The inverter defaults to using the CiA 402 protocol.
- Inverter manufacturer protocol. There are two protocols: Protocol One (F8-07=0) can only operate the inverter in speed mode, while Protocol Two (F8-07=1) can operate the inverter in both speed and torque modes. The selection of the sub-protocol standard can be achieved through parameter F8-19.

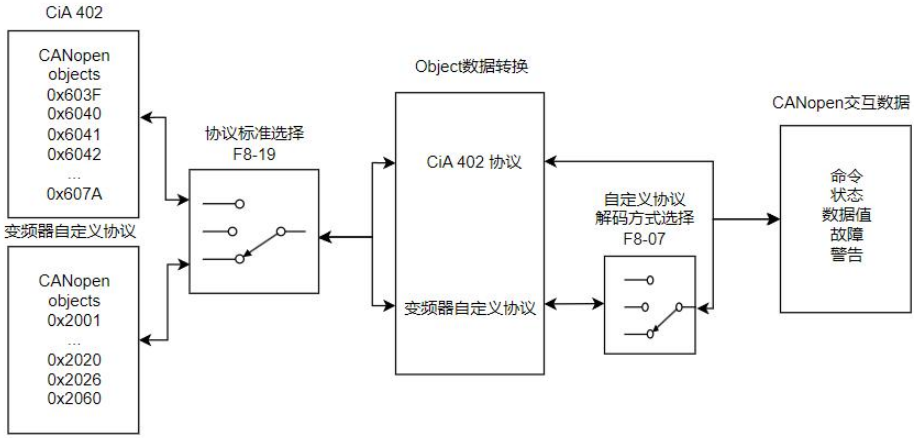



**Figure 29 Communication Sub-Protocol Selection Diagram**

The CANopen control index definitions supported by the inverter are as follows:

**Table 57 CANopen Control Index**

| CANopen Protocol Standard Selection | Speed Mode | | Torque Mode | | Operation Control | |
|---|---|---|---|---|---|---|
| | index-subindex | Meaning | index-subindex | Meaning | index-subindex | Meaning |
| CiA402 | 0x6042-0x00 | Target Speed (rpm) | 0x6071-0x00 | Target Torque (%) | 0x6040-0x00 | Operation Command |
| | \ | \ | 0x6072-0x00 | Maximum Torque Limit (%) | 0x605A-0x00 | Fast Shutdown Handling Method |
| | \ | \ | \ | \ | 0x605C-0x00 | Prohibited Operation Handling Method |
| Midea Protocol One | 0x2020-0x02 | Target Frequency (Hz) | \ | \ | 0x2020-0x01 | Operation Command |
| Midea Protocol Two | 0x2060-0x03 | Target Frequency (Hz) | 0x2060-0x07 | Target Torque (%) | 0x2060-0x01 | Operation Command |
| | 0x2060-0x04 | Torque Limit (%) | 0x2060-0x08 | Frequency Limit (%) | \ | \ |

General Indexes Do Not Require Sub-Protocol Selection, All Can Be Used, Relevant Indexes Are as Follows:

- Read-Only (RO) Attribute Indexes.
- Parameter Group Read-Write Indexes (0x2010-0x01 ~ 0x201F-0x64, 0x2070-0x01 ~ 0x20B0-0x64)
- Acceleration/Deceleration Time Setting Indexes (0x604F-0x00, 0x6050-0x00)

#### 6.3.3.1. Sub-protocol Introduction

CiA 402 protocol is a CANopen application layer protocol specifically designed for inverters and motion control scenarios, built on top of the CiA 301 protocol. Divides the inverter into 9 operating states, including Start, Initialization (Not ready to switch on), Standby (Switch on disable), Ready to Operate (Ready to switch on), Operation (Switch on), Operation Enabled (Operate enable), Quick Stop (Quick stop active), Fault Reaction (Fault reaction active), and Fault. After the inverter is powered on and completes initialization, it will remain in the Ready to Operate state. The inverter can only be controlled to operate when it is in the normal operating state.

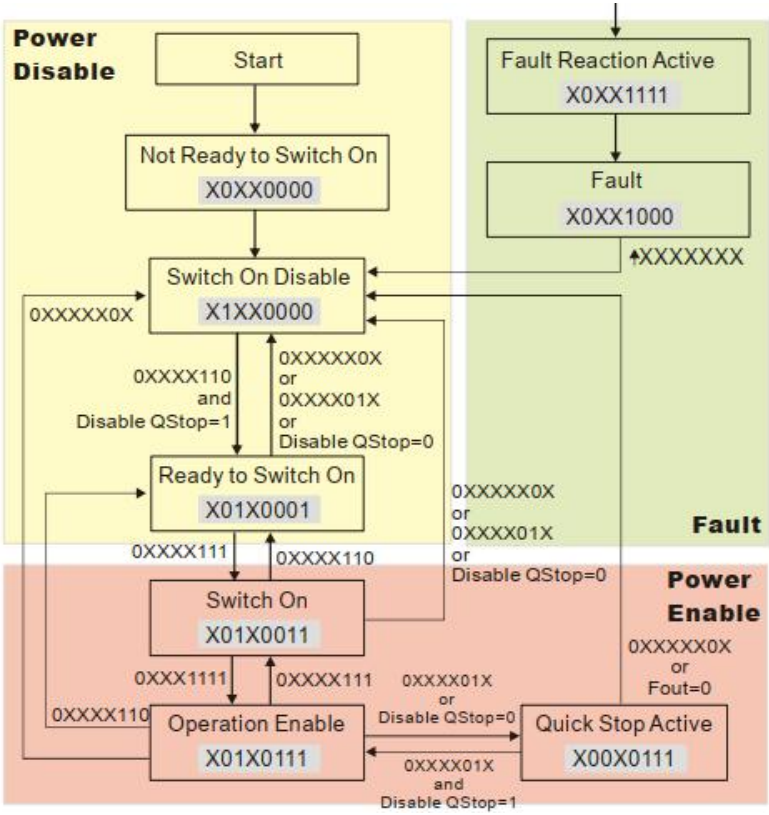



**Figure 30 Inverter State Machine**

Note: When the setting value of Index 605A is 5~7, the inverter can be controlled to return directly from fast shutdown to operational enable by disabling fast shutdown and control command. Otherwise, the inverter cannot return directly from fast shutdown to operational enable.

**Table 58Control Word (Index 6040h)**

| bits | 15~9 | 8 | 7 | 6~4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

| bits | 15~9 | 8 | 7 | 6~4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Meaning | Reserved | Halt | Fault Reset | Normal Operation | Operational Enable | Fast Shutdown | Enable Voltage | Running |

Note: To start the motor, first send 0xxxx110 (06h) to enter the Ready to Switch ON state under no error conditions, then send 0xxxx111 (07h) to enter the Switch ON state, and the motor will start rotating.

**Table 59Status Word (Index 6041h)**

| bits | 15~14 | 13~12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Meaning | Reserved | Normal Operation | Amplitude Limiting Action | Target Value Reached | Remote Control | Reserved | Warning | Disable Operation | Fast Shutdown | Voltage Enable | Fault | Operation Enable | Running | Operation Preparation Complete |

CANopen Object Dictionary

**Table 60Standard Object Dictionary (Object Dictionary)**

| Index | Object |
|---|---|
| 0000h | Reserved |
| 0001h ~ 025Fh | Data Types (Data types) |
| 0260h ~ 0FFFh | Reserved |
| 1000h ~ 1FFFh | 通讯对象子协议区(Communication profile area ) |
| 2000h ~ 5FFFh | 制造商特定子协议区(Manufacturer-specific profile area) |
| 6000h ~ 9FFFh | 标准化设备子协议区(Standardized profile area) |
| A000h ~ AFFFh | Standardized Network variable area, compliant with IEC61131-3 |
| B000h ~ BFFFh | Standardized System variable area for routing gateways |

| Index | Object |
|---|---|
| C000h ~ FFFFh | Reserved |

**Table 61Communication profile area**

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| 1000 | 0 | Device type | unsigned 32 | RO | Device Type Description |
| 1001 | 0 | Fault Register | unsigned 8 | RO | Describe fault type, refer to the fault register description table below for details |
| 1005 | 0 | Synchronous Message COB-ID | unsigned 32 | RW | Synchronous message COB-ID for synchronous transmission |
| 1006 | 0 | Synchronous Communication Cycle Period | unsigned 32 | RW | Set synchronous communication cycle period (us) |
| 100C | 0 | Guard Time | unsigned 16 | RW | Guard time (ms) * Lifetime factor = Node guard protocol lifetime |
| 100D | 0 | Lifetime Factor | unsigned 8 | RW | Guard time (ms) * Lifetime factor = Node guard protocol lifetime |
| 1010 | 1 | Save Parameters | unsigned 32 | RW | Write SDO value 0x65766173 (“evas”) to save all parameters |
| | 2 | | | | Write SDO value 0x65766173 (“evas”) to save communication parameters (index 0x1000~0x1FFF) |
| 1011 | 1 | Restore default parameters | unsigned 32 | RW | Write SDO value 0x64616F6C (“daol”) to restore all parameters to default values |
| | 2 | | | | Write SDO value 0x64616F6C (“daol”) |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | | | | | to restore communication parameters (index 0x1000~0x1FFF) to default values |
| 1014 | 0 | Emergency message COB-ID | unsigned 32 | RW | Define emergency object (EMCY) COB-ID, default value: 0x80 + Node-ID |
| 1016 | 1 | Consumer heartbeat time | unsigned 32 | RW | Value definition: bits 31 ~ 24 reserved, bits 23 ~ 16 Node-ID, bits 15 ~ 0 heartbeat time (ms) |
| 1017 | 0 | Production End Heartbeat Time | unsigned 32 | RW | Define Heartbeat Cycle Time (ms), Write 0 to Disable Heartbeat Function |
| 1018 | 1 | Supplier ID | unsigned 32 | RO | Supplier ID, Inverter Manufacturer Value: 0x476 (Temporary Use) |
| | 2 | Product Code | | | Product Code |
| | 3 | Version Number | | | Version Number |
| 1400 | 0 | RPDO1 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default Value: 0x200 + Node-ID |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| 1401 | 0 | RPDO2 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default Value: 0x80000300 + Node-ID (Disable |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | | | | | RPDO2) |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| 1402 | 0 | RPDO3 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default value:0x80000400 + Node-ID (Disable RPDO3) |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| 1403 | 0 | RPDO4 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default value:0x80000500 + Node-ID (Disable RPDO4) |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| 1600 | 0 | RPDO1 Mapping Parameters | unsigned 8 | RO | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x60400010, 0x6040-0x00 indicates the 16-bit control word in CiA 402 protocol |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x60420010, 0x6042-0x00 indicates the 16-bit status word in CiA 402 protocol |
| | 3 | Mapped Object 3 | unsigned 32 | RW | |
| | 4 | Mapped Object | unsigned | RW | |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | | 4 | 32 | | |
| 1601 | 0 | RPDO2 Mapping Parameters | unsigned 8 | RO | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x20264110, 0x2026-0x41 indicates the 16-bit DO status setting value in Custom Protocol One |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x2026A110, 0x2026-0xA1 indicates the 16-bit AO1 ratio setting value in Custom Protocol One |
| | 3 | Mapped Object 3 | unsigned 32 | RW | Default value:0x2026A210, 0x2026-0xA2 indicates the 16-bit AO2 ratio setting value in Custom Protocol One |
| | 4 | Mapped Object 4 | unsigned 32 | RW | |
| 1602 | 0 | RPDO3 Mapping Parameters | unsigned 8 | RO | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x60400010, 0x6040 indicates the 16-bit control word in CiA 402 Protocol |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x607A0020, 0x6042 indicates the 32-bit target position in CiA 402 Protocol |
| | 3 | Mapped Object 3 | unsigned 32 | RW | Default value:0x60600008, 0x6042 indicates the 8-bit operation mode selection in CiA 402 |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | | | | | Protocol |
| | 4 | Mapped Object 4 | unsigned 32 | RW | |
| 1603 | 0 | RPDO4 Mapping Parameters | unsigned 8 | RO | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x60400010, 0x6040 indicates the 16-bit control word in CiA 402 Protocol |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x60710010, 0x6042 indicates the 16-bit target torque in CiA 402 Protocol |
| | 3 | Mapped Object 3 | unsigned 32 | RW | Default value:0x60600008, 0x6042 indicates the 8-bit operation mode selection in CiA 402 Protocol |
| | 4 | Mapped Object 4 | unsigned 32 | RW | |
| 1800 | 0 | TPDO1 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default value:0x180 + Node-ID |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| | 3 | Inhibition Time | unsigned 8 | RW | PDO Transmission Minimum Interval, default value:0 (not used) |
| | 5 | Event Timer | unsigned 8 | RW | Default value:0 (not used) |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| 1801 | 0 | TPDO2 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default value:0x80000280 + Node-ID (TPDO2 disabled) |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| | 3 | Inhibition Time | unsigned 8 | RW | PDO Transmission Minimum Interval, default value:0 (not used) |
| | 5 | Event Timer | unsigned 8 | RW | Default value:0 (not used) |
| 1802 | 0 | TPDO3 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default value:0x80000380 + Node-ID (TPDO3 disabled) |
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| | 3 | Inhibition Time | unsigned 8 | RW | PDO Transmission Minimum Interval, default value:0 (not used) |
| | 5 | Event Timer | unsigned 8 | RW | Default value:0 (not used) |
| 1803 | 0 | TPDO4 Communication Parameters | unsigned 8 | RO | Maximum Supported Sub-index Count |
| | 1 | COB-ID | unsigned 32 | RW | Default value:0x80000480 + Node-ID (TPDO4 disabled) |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | 2 | Transmission Type | unsigned 8 | RW | Default Value: 5 |
| | 3 | Inhibition Time | unsigned 8 | RW | PDO Transmission Minimum Interval, default value:0 (not used) |
| | 5 | Event Timer | unsigned 8 | RW | Default value:0 (not used) |
| 1A00 | 0 | TPDO1 Mapping Parameters | unsigned 8 | RW | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x60410010, 0x6041-0x00 indicates the 16-bit status word in the CiA 402 protocol |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x60430010, 0x6043-0x00 indicates the 16-bit real-time target speed (rpm) in the CiA 402 protocol |
| | 3 | Mapped Object 3 | unsigned 32 | RW | |
| | 4 | Mapped Object 4 | unsigned 32 | RW | |
| 1A01 | 0 | TPDO2 Mapping Parameters | unsigned 8 | RW | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x20260110, 0x2026-0x01 indicates the status of 16-bit DI terminals in Custom Protocol One |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x20266110, 0x2026-0x61 indicates the 16-bit AI1 ratio value in Custom |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | | | | | Protocol One |
| | 3 | Mapped Object 3 | unsigned 32 | RW | Default value:0x20266210, 0x2026-0x62 indicates the 16-bit AI2 ratio value in Custom Protocol One |
| | 4 | Mapped Object 4 | unsigned 32 | RW | Default value:0x20266310, 0x2026-0x63 indicates the 16-bit AI3 ratio value in Custom Protocol One |
| 1A02 | 0 | TPDO3 Mapping Parameters | unsigned 8 | RW | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x60410010, 0x6041-0x00 indicates the 16-bit status word in the CiA 402 protocol |
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x60640020, 0x6064-0x00 indicates the 32-bit actual position in CiA 402 Protocol |
| | 3 | Mapped Object 3 | unsigned 32 | RW | Default value:0x60610008, 0x6061-0x00 indicates 8-bit operating mode in CiA 402 protocol |
| | 4 | Mapped Object 4 | unsigned 32 | RW | |
| 1A03 | 0 | TPDO4 mapping parameters | unsigned 8 | RW | Number of Mapped Objects (0~4) |
| | 1 | Mapped Object 1 | unsigned 32 | RW | Default value:0x60410010, 0x6041-0x00 indicates the 16-bit status word in the CiA 402 protocol |

| Index (hex) | sub-Index (hex) | Name | Data format | Read/Write | Meaning |
|---|---|---|---|---|---|
| | 2 | Mapped Object 2 | unsigned 32 | RW | Default value:0x60770010, 0x6077-0x00 indicates 32-bit actual torque value in CiA 402 protocol |
| | 3 | Mapped Object 3 | unsigned 32 | RW | Default value:0x60610008, 0x6061-0x00 indicates 8-bit operating mode in CiA 402 protocol |
| | 4 | Mapped Object 4 | unsigned 32 | RW | |

**Table 62Fault Register Description**

| Bit | Description |
|---|---|
| 0 | General Fault |
| 1 | Current |
| 2 | Voltage |
| 3 | Temperature |
| 4 | Communication Fault |
| 5 | Device Protocol Specified |
| 6 | Reserved |
| 7 | Vendor Specified |

### 6.3.4. Custom Protocol One

**Table 63 Custom Protocol One**

| Index | Sub-Index | R/W | Data format | Meaning |
|---|---|---|---|---|
| 0x2010+PrGroupNum<br>0x2070+PrGroupNum-0x<br>E | PrNum+1 | R/W | unsigned 16 | Inverter Parameter Setting and Reading※ |
| | 0x01 | R/W | unsigned 16 | Control Command Word |
| | 0x02 | R/W | unsigned 16 | Frequency Command (0.01Hz) |
| 0x2020 | 0x03 | R/W | unsigned 16 | Trigger Command |
| | 0x01 | | | Hbyte:warn code<br>Lbyte:Error code |
| | 0x02 | | | Inverter operating status |
| | 0x03 | | | Frequency Command (0.01Hz) |
| | 0x04 | | | Output Frequency (0.01Hz) |
| | 0x05 | | | Output Current (0.1A) |
| | 0x06 | | | DC Bus Voltage (0.1V) |
| | 0x07 | | | Output Voltage (0.1V) |
| | 0x08 | | | Current multi-speed command segment |
| | 0x09 | | | Reserved |
| 0x2021 | 0x0A | R | unsigned 16 | Display count value |
| | 0x0B | | | Output power factor angle (0.1 degrees) |
| | 0x0C | | | Output torque (0.1%) |
| | 0x0D | | | Actual motor speed (rpm) |
| | 0x0E | | | PG feedback pulse count (0-65535) |
| | 0x0F | | | PG2 pulse command count (0-65535) |
| | 0x10 | | | Output power (0.001kwh) |
| | 0x11 | | | Parameter group number |
| | 0x13 | | | Firmware Version (highword) |
| | 0x14 | | | Hbyte: Startup Screen Selection F7-20<br>Lbyte: Multi-function Display Selection F7-21 |
| | 0x01 | | | Output Current (0.01A), when current exceeds 655.36A, decimal place automatically switches to 1 |
| | 0x02 | | | Display count value |
| 0x2022 | 0x03 | R | unsigned 16 | Output Frequency (0.01Hz) |
| | 0x04 | | | DC Bus Voltage (0.1V) |
| | 0x05 | | | Output Voltage (0.1V) |
| | 0x06 | | | Output power factor angle (0.1 degrees) |

| Index | Sub-Index | R/W | Data format | Meaning |
|---|---|---|---|---|
| | 0x07 | | | Three-phase Output Power (kW) |
| | 0x08 | | | Actual motor speed (rpm) |
| | 0x09 | | | Output torque (0.1%) |
| | 0x0A | | | PG feedback pulse count (0-65535) |
| | 0x0B | | | After PID Function Activation, Display PID Feedback Value (0.01%) |
| | 0x0C | | | AI1 Terminal Input Value (0-100%) |
| | 0x0D | | | AI2 Terminal Input Value (0-100%) |
| | 0x0E | | | AI3 Terminal Input Value (0-100%) |
| | 0x0F | | | Power Module IGBT Temperature (0.1℃) |
| | 0x11 | | | Digital Terminal Input Mode Selection (F5—15) |
| | 0x12 | | | Digital Terminal Output Mode Direction (F6—04) |
| | 0x13 | | | Current multi-speed command segment |
| | 0x14 | | | Digital Terminal Input Status (1: on) |
| | 0x15 | | | Digital Terminal Output Status (1: on) |
| | 0x16 | | | Motor Rotation Count (0—65535), Count Reset on Direction Change or Stop |
| | 0x17 | | | Pulse Input Frequency (PG2.0.01Hz) |
| | 0x18 | | | Pulse Input Position (PG2, 0—65535) |
| | 0x1A | | | Overload Count (0.00-100.00%) |
| | 0x1B | | | GFF Percentage Value (0.01%) |
| | 0x1C | | | DC Bus Voltage (0.1V) |
| | 0x1E | | | Synchronous Motor Pole Segment |
| | 0x1F | | | User Physical Quantity Output |
| | 0x20 | | | Actual Output Frequency Gain Coefficient (0.01) |
| | 0x21 | | | Motor Rotation Count (Stop Hold, Reset Before Operation) |
| | 0x22 | | | Motor Running Position (Stop Hold, Reset Before Operation) |
| | 0x25 | | | Inverter Carrier Frequency (kHz) |
| | 0x27 | | | Inverter Status |
| | 0x28 | | | Output torque (0.1%) |
| | 0x29 | | | Torque Command (0.1%) |
| | 0x2A | | | Energy Display (0.1kWh) |
| | 0x2B | | | PG2 Pulse Input (highword) |
| | 0x2C | | | PG2 Pulse Input (lowword) |

| Index | Sub-Index | R/W | Data format | Meaning |
|---|---|---|---|---|
| | 0x2D | | | Actual Motor Position (highword) |
| | 0x2E | | | Actual Motor Position (lowword) |
| | 0x2F | | | PID Reference Target (0.01%) |
| | 0x30 | | | PID Offset (0.01%) |
| | 0x31 | | | PID Output Frequency (0.01Hz) |
| 0x2026 | 0x01 | R | unsigned 16 | DI Terminal Input Status |
| | 0x02 | R | unsigned 16 | Extended DI Terminal Input Status |
| | 0x03-0x40 | R | \ | Reserved |
| | 0x41 | R/W | unsigned 16 | DO Terminal Output Status |
| | 0x42-0x60 | R | | Reserved |
| | 0x61 | R | | AI1 Proportional Value |
| | 0x62 | R | | AI2 Ratio Value |
| | 0x63 | R | | AI3 Ratio Value |
| | 0x64-0x6A | R | | Reserved |
| | 0x6B | R | | Expansion Card AI10 Input Ratio Value |
| | 0x6C | R | | Expansion Card AI11 Input Ratio Value |
| | 0xA1 | R/W | | AO1 Output Ratio Value |
| | 0xA2 | R/W | | AO2 Output Ratio Value |
| | 0xA3-0xAA | R/W | | Reserved |
| | 0xAB | R/W | | Expansion Card AO10 Output Ratio Value |
| | 0xAC | R/W | | Expansion Card AO11 Output Ratio Value |

※Note the mapping relationship between object dictionary and inverter function codes is as follows:

Object Dictionary Index = 0x2010 + Function Code Group Number (F0-FF Function Group);

or 0x2070 + Function Code Group Number (U0-L8 Function Group) - 0xE

Object Dictionary Sub-index = Function Code Group Internal Offset in Hexadecimal + 1;

**Table 64Control Command Word Bit Definition:**

| Control Command Word | |
|---|---|
| bit 1-0 | 0: No Function |
| | 1: Stop |
| | 2: Start |
| | 3: JOG Start |
| bit 3-2 | Reserved |

| Control Command Word | |
|---|---|
| bit 5-4 | 0: No Function |
| | 1: Forward Command |
| | 2: Reverse Command |
| | 3: Change Direction |
| bit 7-6 | 0: First Ramp |
| | 1: Second Ramp |
| | 2: Third Ramp |
| | 3: Fourth Ramp |
| bit 11-8 | 0: Main Speed |
| | 1-15: First to Fifteenth Segment Speed Selection |
| bit 12 | 1: Enable bit6-bit11 function |
| bit 15-13 | Reserved |

**Table 65Trigger Command Word Bit Definition**

| Trigger Command | |
|---|---|
| bit 0 | 1: External Fault (EF) |
| bit 1 | 1: Reset Command |
| bit 2 | 1: External Interrupt (BB) |
| bit 15-3 | Reserved |

**Table 66Inverter Operation Status Word Bit Definition**

| Inverter operating status | |
|---|---|
| bit 1-0 | 0: Stop |
| | 1: Accelerating |
| | 2: Standby |
| bit 2 | JOG operation |
| bit 4-3 | 0: Forward |
| | 1: Reverse to forward |
| | 2: Forward to reverse |
| | 3: Reverse |
| bit 7-5 | Reserved |
| bit 8 | 1: Main frequency source is communication interface |
| bit 9 | 1: Main frequency source is analog input |
| bit 10 | 1: Main command source is communication interface |

| | |
|---|---|
| bit 11 | 1: Parameter lock |
| bit 12 | 1: Enable keypad parameter copy function |
| bit 15-13 | Reserved |

**Table 67Inverter status word bit definition**

| Inverter Status | |
|---|---|
| bit 1-0 | 0: No direction |
| | 1: Forward |
| | 2: Reverse |
| bit 3-2 | 0: Inverter Ready |
| | 1: Inverter Fault |
| bit 4 | 1: Inverter Output 0: No Output |
| bit 5 | 1: Warning 0: No Warning |

For the mapping relationship between terminal inputs/outputs and CANopen index, refer to 5.3 CANopen Analog AIO and DIO Description.

### 6.3.5 Custom Protocol Two

**Table 68 Custom Protocol Two**

| Index | Sub-Index | R/W | Data format | Bit Definition | | Speed Mode | Torque Mode |
|---|---|---|---|---|---|---|---|
| | | | | bit | Meaning | | |
| 0x2060 | 0x01 | R/W | unsigned 16 | 0 | Ack | 0: fcmd = 0<br>1: fcmd = fset | \ |
| | | | | 1 | Dir | 0: Forward Command<br>1: Reverse Command | \ |
| | | | | 2 | Move | \ | \ |
| | | | | 3 | Halt | 0: Continue Running to Target Speed<br>1: Temporarily Stop According to Deceleration Settings | \ |
| | | | | 4 | Hold | 0: Continue Running to Target Speed<br>1: Maintain Current Frequency | \ |
| | | | | 5 | JOG | 0: OFF 1: RUN | \ |
| | | | | 6 | Qstop | Fast Shutdown | |
| | | | | 7 | RUN | 0: Shutdown 1: Startup | |
| | | | | 8 | Multi-Speed Selection Bit | Multi-Speed Selection Bit 0 | \ |
| | | | | 9 | | Multi-Speed Selection Bit 1 | \ |
| | | | | 10 | | Multi-Speed Selection Bit 2 | \ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | 11 | | Multi-Speed Selection Bit 3 | \ |
| | | | | 12 | Acceleration and Deceleration Time Selection Bit | Acceleration and Deceleration Time Selection Bit 0 | \ |
| | | | | 13 | | Acceleration and Deceleration Time Selection Bit 1 | \ |
| | | | | 14 | Software Terminal Function Enable | Software Multi-Speed and Acceleration/Deceleration Time Switching Enable | \ |
| | | | | 15 | Reset | Clear Fault Code | |
| | 0x02 | | | 0-15 | Run Mode Command | 0: Speed Mode | 2: Torque Mode |
| | 0x03 | | | 0-15 | | Frequency Command (0.01Hz) | \ |
| | 0x04 | | | 0-15 | | Torque Limit Value (0.01pu) | |
| | 0x05 | | signed 32 | 0-31 | Reserved | Reserved | |
| | 0x06 | | unsigned 16 | 0-15 | Reserved | Reserved | |
| | 0x07 | | signed 16 | 0-15 | | \ | Torque Command |
| | 0x08 | | unsigned 16 | 0-15 | | \ | Speed Limit |
| 0x2061 | 0x01 | R | unsigned 16 | 0 | Arrive | Frequency Command Arrival | Torque Command Arrival |
| | | | | 1 | Dir | 0: Forward 1: Reverse | |
| | | | | 2 | Warn | Warning Occurred | |
| | | | | 3 | Error | Fault Occurred | |
| | | | | 4 | Reserved | Reserved | |
| | | | | 5 | JOG | JOG operation | |
| | | | | 6 | Qstop | Fast Shutdown | |
| | | | | 7 | PowerOn | Excitation | |
| | | | | 15-8 | Reserved | Reserved | |
| | 0x02 | | | 0-15 | Reserved | Reserved | |
| | 0x03 | | | 0-15 | Frequency | Actual Output Frequency | |
| | 0x04 | | | 0-15 | Reserved | Reserved | |
| | 0x05 | | signed 32 | 0-31 | Reserved | Reserved | |
| | 0x06 | | unsigned 16 | 0-15 | Reserved | Reserved | |
| | 0x07 | | signed 16 | 0-15 | Torque | Actual Torque | |

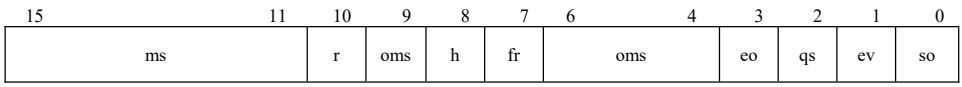
### 6.3.6 CiA 402

**Table 69 CiA 402 Supported Index**

| Index | Sub-Index | Definition | Default Value | R/W | Data format | Unit | PDO Mapping |
|---|---|---|---|---|---|---|---|
| 0x6007 | 00h | Communication Exception Action Selection | 2 | R/W | signed 16 | \ | Yes |
| 0x603F | 00h | CANopen Fault Code | 0 | RO | unsigned 16 | \ | Yes |
| 0x6040 | 00h | Control Word | 0 | R/W | unsigned 16 | \ | Yes |
| 0x6041 | 00h | Status Word | 0 | RO | unsigned 16 | \ | Yes |
| 0x6042 | 00h | Target Speed (vl) | 0 | R/W | signed 16 | rpm | Yes |
| 0x6043 | 00h | Actual Target Speed (vl) | 0 | RO | signed 16 | rpm | Yes |
| 0x6044 | 00h | Actual Speed (vl) | 0 | RO | signed 16 | rpm | Yes |
| 0x604F | 00h | First Acceleration Time (vl) | 10000 | R/W | unsigned 32 | ms | Yes |
| 0x6050 | 00h | First Deceleration Time (vl) | 10000 | R/W | unsigned 32 | ms | Yes |
| 0x6051 | 00h | Rapid Stop Time (vl) | 0 | R/W | unsigned 32 | ms | Yes |
| 0x605A | 00h | Rapid Stop Option | 2 | R/W | signed 16 | \ | No |
| 0x605C | 00h | Stop Option | 1 | R/W | signed 16 | \ | No |
| 0x6060 | 00h | Operation Mode Selection | 2 | R/W | signed 8 | \ | Yes |
| 0x6061 | 00h | Operation Mode Display | 2 | RO | signed 8 | \ | Yes |
| 0x6064 | 00h | Actual Position Value (pp) | 0 | RO | signed 32 | \ | Yes |
| 0x 6071 | 00h | Target Torque (tq) | 0 | R/W | signed 16 | 0.001 | Yes |
| 0x 6072 | 00h | Maximum Torque (tq) | 150 | R/W | unsigned 16 | 0.001 | No |
| 0x 6075 | 00h | Motor Rated Current (tq) | 0 | RO | unsigned 32 | mA | No |
| 0x 6077 | 00h | Actual Torque Value (tq) | 0 | RO | signed 16 | 0.1%* | Yes |
| 0x 6078 | 00h | Actual Current Value (tq) | 0 | RO | signed 16 | 0.1%* | No |
| 0x 6079 | 00h | DC bus voltage (tq) | 0 | RO | unsigned 32 | mV | Yes |
| 0x 607A | 00h | Target position (PP) | 0 | R/W | signed 32 | \ | Yes |

*: Motor rated values are 100%

Control word (6040h-00h) bit definitions are as follows:



Note: ms = manufacture-specific; r = reserved; oms = operation mode specific; h = halt;

fr = fault reset; eo = enable operation; qs = quick stop; ev = enable voltage; so = switch on

Status word (6041h-00h) bit definitions are as follows:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ms | | oms | | ila | tr | rm | ms | w | sod | qs | ve | f | oe | so | rtso |
| MSB | | | | | | | | | | | | | | | LSB |

Note:ms = manufacture-specific; oms = operation mode specific; lia = internal limit active; tr = target reached; rm = remote; w = warning; sod = switch on disable; qs = quick stop;

ve = voltage enable; f = fault; oe = operation enable; so = switch on; rtso = ready to switch on

Status word encoding mapping table is as follows:

**Table 70 Status Word Encoding Table**

| statusword | PDS FSA state |
|---|---|
| xxxx xxxx x0xx 0000b | Not ready to switch on |
| xxxx xxxx x1xx 0000b | Switch not disabled |
| xxxx xxxx x01x 0001b | Ready to switch on |
| xxxx xxxx x01x 0011b | Switched on |
| xxxx xxxx x01x 0111b | Operation enabled |
| xxxx xxxx x00x 0111b | Quick stop active |
| xxxx xxxx x0xx 1111b | Fault reaction active |
| xxxx xxxx x0xx 1000b | Fault |

## 6.4. CANopen Control Inverter Function Debugging

Preparations for Controlling the Inverter via CANopen are as follows:

1. Wiring, refer to 2.2 CAN Wiring Method

2. Set the frequency command source, parameter F0-06 = 6, select the frequency command source as CANopen setting.

3. Set the operation command source, parameter F0-05 = 3, select the operation command source as CANopen setting.

4. Set the CANopen slave station node address, parameter F8-14 sets the slave station address (default value is 0, indicating the slave station function is disabled, 1~127 are valid slave station node addresses). After setting the slave station node address, if a station number error or CANopen memory error is reported, the CANopen can be reset by setting parameter F0-18 = 7.

5. Set the CAN communication speed, parameter F8-15 sets the CANopen communication speed, parameter value setting range 0 (1Mbps), 1 (500kbps), 2 (250kbps), 3 (125kbps), 4 (100kbps), 5 (50kbps).

6. The master station uses NMT to control the slave station to enter the operational mode.

### 6.4.1. CANopen selects the CiA 402 protocol.

Under the CiA 402 protocol, speed mode and torque mode are supported. Setting parameter F8-19 = 1 enables the CiA 402 protocol. To enable the fast shutdown function with external terminals, set one of the parameters F5-00~F5-07 or LC-00~LC-02 to 53.

#### 6.4.1.1. Speed Mode

The debugging steps are as follows:

1. Set the operation mode at index 6060h-00h to 2, configuring the inverter to speed mode.

2. The control command word controls the inverter to enter the operational enable state; send the control command word 0x0E at index 6040h-00h, then send 0x0F to enable the inverter to enter the operational enable state. Note the control of the bits 6~4 of the control command word, defined as follows:

**Table 71 Control Command Word Bits 6~4**

| Bit 6 | bit 5 | bit 4 | Function |
|---|---|---|---|
| 1 | 0 | 1 | Run at the current frequency |
| 1 | 1 | 1 | Run to target |
| x | x | x | Decelerate to 0Hz |

4. Set the target speed, set the target speed at index 6042h-00h, unit is rpm, set a positive value for forward rotation, a negative value for reverse rotation. Note the relationship between motor speed and frequency (information on the number of motor pole pairs).

5. Set acceleration and deceleration time, set the inverter acceleration and deceleration time at indices 604Fh-00h and 6050h-00h, unit is ms.

6. Get the current operating speed, read index 6043h-00h to obtain the current operating speed, unit is rpm.

#### 6.4.1.2. Torque Mode

The debugging steps are as follows:

1. Set index 6060h-00h to 4, setting the inverter to torque mode, at this time 6042h-00h is used to set the speed limit.

2. The control command word controls the inverter to enter the operational enable state; send the control command word 0x0E at index 6040h-00h, then send 0x0F to enable the inverter to enter the operational enable state.

3. Set the target torque, index 6071h-00h sets the target torque, index 6072h-00h sets the maximum torque.

4. Get the current torque, read index 6077h-00h to obtain the current torque, unit 0.1%.

### 6.4.2. CANopen selects the inverter manufacturer's custom protocol

Parameter F8-19 = 0 enables the inverter manufacturer's custom protocol.

#### 6.4.2.1. Custom Protocol One

The custom protocol only supports speed mode. Debugging steps are as follows:

1. Parameter F8-07 = 0 enables custom protocol one.

2. Set target frequency, index 2020h-02 sets the target frequency, unit is 0.01Hz, for example, 5000 represents 50Hz.

3. Operation control, index 2020h-01h set to 2 for operation, index 2020h-01h set to 1 for shutdown.

#### 6.4.2.2. Custom Protocol Two

Custom Protocol Two supports speed mode and torque mode.

- Speed Mode Debugging Steps are as follows:

1. Parameter F8-07 = 1 enables Custom Protocol Two.

2. Set target frequency, index 2060h-03 sets the target frequency, unit is 0.01Hz, for example, 5000 represents 50Hz.

3. Operation control, index 2060h-01h set to 80h for excitation, index 2060h-01h set to 81h for operation.

- Torque Mode Debugging Steps:

1. Parameter F8-07 = 1 enables Custom Protocol Two.

2. Set the target torque, index 2060h-07 sets the target torque, unit is 0.1%, for example, 1000 represents 100%.

3. Operation Control, set index 2060h-01h to 80h for excitation, after the inverter is excited, it automatically runs to the target torque.

4. Get Current Torque, read index 2061h-07h to obtain the current torque, unit is 0.1%.

Note: In Torque Mode, if the speed reaches the limit value, the inverter may reduce the output torque to ensure the speed remains within the limit.

### 6.4.3. Analog AIO and DIO

The inverter can simulate the status of IO ports through CANopen data, achieving software simulation of AO (Analog Output) and DO (Digital Output) functions, and can obtain the status of AI (Analog Input) and DI (Digital Input) and send it to the CAN bus. CANopen index and DIO (Digital input and output) and AIO (Analog input and output) mapping table as follows: DI terminal (physical terminal name MI, Multi-function input) and CANopen index (2026h-01h) data bit mapping table:

**Table 72 DI terminal and CANopen index data bit mapping table**

| Terminal Name | Attribute | bit |
|---|---|---|
| FWD | RO | bit 0 |
| REV | RO | bit 1 |
| MI 1 | RO | bit 2 |
| MI 2 | RO | bit 3 |

| Terminal Name | Attribute | bit |
|---|---|---|
| MI 3 | RO | bit 4 |
| MI 4 | RO | bit 5 |
| MI 5 | RO | Bit 6 |
| MI 10 | RO | bit 10 |
| MI 11 | RO | bit 11 |
| MI 12 | RO | bit 12 |

DO terminal (physical terminal name MO, Multi-function output) and CANopen index (2026h-41h) data bit mapping table:

**Table 73 DO terminal and CANopen index data bit mapping table**

| Terminal Name | Parameter Setting | Attribute | bit |
|---|---|---|---|
| RY 1 | 02-13 = 50 | RW | bit 0 |
| RY 2 | 02-14 = 50 | RW | bit 1 |
| MO 1 | 02-16 = 50 | RW | bit 3 |
| MO 2 | 02-17 = 50 | RW | bit 4 |
| MO 10 | 02-36 = 50 | RW | bit 5 |
| MO 11 | 02-37 = 50 | RW | Bit 6 |
| MO 12 | 02-38 = 50 | RW | bit 7 |

AI Terminal (Physical Terminal Names AI1, AI2, AI3, Analog Input) and CANopen Index (2026h) Data Bit Mapping Table:

**Table 74 AI Terminal and CANopen Index Data Bit Mapping Table**

| Terminal Name | Attribute | sub-index |
|---|---|---|
| AI1 | RO | 61h |
| AI2 | RO | 62h |
| AI3 | RO | 63h |
| AI 10 | RO | 64h |
| AI 11 | RO | 65h |

AO Terminal and CANopen Index (2026h) Data Bit Mapping Table:

**Table 75 AO Terminal and CANopen Index Data Bit Mapping Table**

| Terminal Name | Parameter Setting | Attribute | sub-index |
|---|---|---|---|
| AO 1 | 03-20 = 20 | RW | A1h |
| AO 2 | 03-23 = 20 | RW | A2h |
| AO 10 | 14-12 = 20 | RW | A3h |
| AO 11 | 14-13 = 20 | RW | A4h |

Taking CANopen control of AO and DO as an example, the parameter configuration steps are as follows:

- Set parameter F6-14 = 20, AO1 is defined as controlled by CANopen index 2026h-A1h, controlling the data of index 2026h-A1h can control the output voltage value of AO1. If you want to control AO1 to output 6V voltage (duty = 60%), you need to set index 2026h-A1h to 6000.

- Set parameter F6-00 = 50, RY1 is defined as controlled by bit 0 of CANopen index 2026h-41h, controlling the bit 0 of index 2026h-41h data can control the ON/OFF state of RY1. If you want to control RY1 to output ON, you need to set bit 0 of index 2026h-41h to 1.

## 6.5. CANopen Fault Code

Fault codes can be read through object 0x603F, and fault types can be read through object 0x1001. Fault codes xx80h~xxFFh and FF00h ~ FFFFh are manufacturer-defined.

**Table 76 CANopen Fault Codes**

| Fault Code | Meaning |
|---|---|
| 2213h | Overcurrent during acceleration and deceleration |

| Fault Code | Meaning |
|---|---|
| 2240h | Ground fault |
| 2250h | Short circuit fault |
| 2310h | Continuous overcurrent |
| 2314h | Overcurrent at constant speed |
| 2331h | U phase output missing |
| 2332h | V phase output missing |
| 2333h | W Phase Output Phase Loss |
| 3130h | Input Phase Loss |
| 3210h | DC Bus Overvoltage |
| 3220h | DC Bus Undervoltage |
| 3330h | Motor Connection Switching Error |
| 4310h | Inverter Overtemperature |
| 5530h | EEPROM Abnormal |
| 7301h | PG Feedback Abnormal |
| 7320h | Position Out of Limit |
| 7500h | Modbus Communication Fault |
| 8100h | CANopen Software Fault |
| 8130h | CANopen Communication Timeout |
| 8140h | CANopen Disconnection |
| 8311h | Over Torque Fault |
| 8321h | Current Underflow |
| 8A00h | Low Frequency Overload Protection |

| Fault Code | Meaning |
|---|---|
| 9000h | External Trigger Fault |
| FF00h | IGBT Temperature Detection Abnormal (th1o) |
| FF01h | Capacitor temperature detection abnormal (th2o) |
| FF04h | U phase current detection error (cd1) |
| FF05h | V phase current detection error (cd2) |
| FF06h | W phase current detection error (cd3) |
| FF07h | CBC hardware circuit abnormal (hd0) |
| FF08h | OC hardware circuit abnormal (hd1) |
| FF20h | Motor overheating (oh3) |
| FF21h | Motor auto-identification error (AUE) |
| FF22h | PID feedback disconnection |
| FF25h | AI Disconnection (ACE) |
| FF26h | Password Input Incorrect Three Times (Pcod) |
| FF27h | Deceleration Energy Regeneration Action (dEb) |
| FF28h | Over Slip (oSL) |
| FF29h | PG Card Hardware Error (PGF5) |
| FF2Bh | U Phase Short Circuit Detected Before Operation (Aoc) |
| FF2Ch | V Phase Short Circuit Detected Before Operation (boc) |
| FF2Dh | W Phase Short Circuit Detected Before Operation (coc) |
| FF30h | S1 Internal Loop Diagnosed with Abnormality (STL1) |
| FF31h | STO (STO) |
| FF32h | S2 Internal Loop Diagnostics Detected an Abnormality |

| Fault Code | Meaning |
|---|---|
| | (STL2) |
| FF33h | Internal Loop Diagnostics Detected an Abnormality (STL3) |
| FF3Dh | Motor Auto-Identification Error (DC Test Phase) (AuE1) |
| FF3Eh | Motor Auto-Identification Error (High-Frequency Stall Phase) (AuE2) |
| FF3Fh | Motor Auto-Identification Error (Rotation Test Phase) (AuE3) |

Fault Type Data Bit Definitions Are as Follows:

| bit | Meaning |
|---|---|
| 0 | General Fault |
| 1 | Current |
| 2 | Voltage |
| 3 | Temperature |
| 4 | Communication Fault |
| 5 | Device Sub-Protocol Specified |
| 6 | Reserved |
| 7 | Vendor Specified |

## 6.6. Communication Configuration

For Example, to Configure the Module to Establish CANopen Communication with the PLC. The configuration process varies for different PLC models; please consult technical support for details.

### 6.6.1. Configure Configuration

1. Create a project

Double-Click to Open the Host Computer Programming Software, Click on 'New Project'. Select the PLC Device Type and Programming Language, Change the Save Path and Project Name, Then Click 'OK'.

Enter the 'Network Configuration' interface, click the PLC icon, and check 'CANopen Master Station'.

2. Install EDS File

If the EDS file has not been installed before, it needs to be installed first. Enter the 'Network Configuration' interface, click 'Import EDS File', find the path where the EDS file is stored, and click 'Open'.

After the EDS file is successfully installed, you can find the CANopen device in the network device list — CANopen port — third-party manufacturers — SUYNE.

3. Configure Slave Station

In the network device list, find the CANopen module, double-click to add the module.

In the 'Device' list, under 'CANopen', find the module device, double-click to enter the configuration interface. In the 'Slave Parameter Configuration' page, check 'Enable Expert Mode' to open more configuration options.

In the 'Slave Station Parameter Configuration' page, the node ID can be configured, and the value must be consistent with the inverter F8-14 setting.

In the 'CANopen Master Station' page, the baud rate can be configured, and the value must be consistent with the inverter F8-15 setting.

### 6.6.2. Configure Periodic Communication

1. Configure RPDO

The EMH-OP Communication Expansion Card supports 4 RPDO transmissions, with a maximum of 16 register data transmissions.

2. Configure TPDO

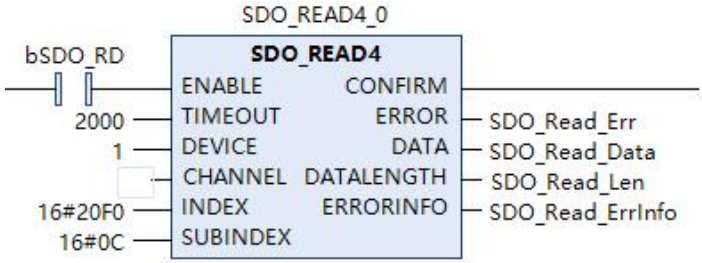
The EMH-OP Communication Expansion Card supports 4 TPDO transmissions, with a maximum of 16 register data transmissions.

After configuration is complete, download the configuration, and the PLC and module will automatically perform periodic data exchange.

### 6.6.3. Configure Non-periodic Communication

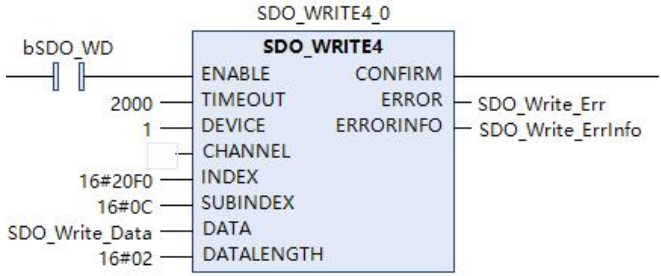
1. SDO Read Operation

The module supports the fast SDO transfer mechanism, with each data transmission length being 0~4 bytes. The SDO_READ4 function block of the PLC can achieve SDO read operations.




2. SDO Write Operation

The SDO_WRITE4 function block of the PLC can achieve SDO write operations.


SDO_WRITE4_0
bSDO_WD
SDO_WRITE4
ENABLE
CONFIRM
2000
TIMEOUT
ERROR
SDO_Write_Err
1
DEVICE
ERRORINFO
SDO_Write_ErrInfo
CHANNEL
16#20F0
INDEX
16#0C
SUBINDEX
SDO_Write_Data
DATA
16#02
DATALENGTH

# 7. Appendix Periodic General Fault Codes

**Table 77 Periodic General Fault Codes**

| Diagnostic Classification | Status Code |
|---|---|
| Inverter Body Diagnostic Information (Register Address: 0x2100) | Low Byte (Fault Error Code):<br>0 No Abnormal Record<br>1 Overcurrent During Acceleration<br>2 Overcurrent During Deceleration<br>3 Overcurrent During Constant Speed<br>4 Ground Short Circuit<br>5 IGBT Upper and Lower Bridge Short Circuit<br>6 Overcurrent During Stop<br>7 Overvoltage During Acceleration<br>8 Overvoltage During Deceleration<br>9 Overvoltage During Constant Speed<br>10 Overvoltage During Stop<br>11 Undervoltage During Acceleration<br>12 Undervoltage During Deceleration<br>13 Undervoltage During Constant Speed<br>14 Undervoltage during Stop<br>15 Input Phase Loss Protection<br>16 IGBT Overtemperature<br>17 Ambient Overtemperature<br>21 Inverter Overload<br>22 Motor 1 Overload Protection<br>23 Motor 2 Overload Protection<br>24 Motor Overheat<br>26 Over Torque 1<br>27 Over Torque 2<br>28 Low Current<br>29 Limit Reached<br>31 Memory Read Error<br>33 U Phase Current Detection Error<br>34 V Phase Current Detection Error<br>35 W Phase Current Detection Error<br>36 cc Hardware Circuit Abnormality<br>37 oc Hardware Circuit Abnormality<br>38 ov Hardware Circuit Abnormality<br>40 Motor Parameter Identification<br>55 Illegal Communication Address<br>56 Communication Data Error<br>57 Communication Write to Read-Only Address<br>58 Modbus Transmission Timeout<br>62 Deceleration Regenerative Braking Action<br>63 Excessive Slip<br>64 Please Reset Machine Code<br>65 PG Card Hardware Error<br>69 Feedback Speed Divergence<br>70 Feedback Speed Deviation Excessive<br>72 STO1 Fault<br>76 STO<br>77 STO2 Fault<br>79 U Phase Overcurrent<br>80 V Phase Overcurrent<br>81 W Phase Overcurrent<br>82 U Phase Output Phase Loss<br>83 V Phase Output Phase Loss<br>84 W Phase Output Phase Loss<br>87 Low Frequency Overload Protection<br>101 CANopen Disconnection 1<br>102 CANopen Disconnection 2<br>104 CANopen Hardware Disconnection<br>105 CANopen Index Error<br>106 CANopen Station Number Error<br>107 CANopen Memory Error<br>120 EMS Overcurrent<br>128 Over Torque 3<br>129 Over Torque 4<br>134 Motor 3 Overload Protection<br>135 Motor 4 Overload Protection<br>141 Ground Fault Before Operation<br>142 Parameter Identification Error 1 |

| | |
|---|---|
| | Error<br>41 PID Disconnection<br>42 PG Feedback Setting Error<br>43 PG Feedback Disconnection<br>44 PG Feedback Stall<br>45 PG Slip Abnormality<br>48 AI Current Signal Disconnection<br>49 External Fault<br>50 External Terminal Emergency Stop<br>51 External Interruption<br>52 Password Input Incorrect Three Times (Pcod)<br>54 Illegal Communication Command<br>143 Parameter Identification Error 2<br>144 Tension Band Break<br>147 Tension PID Deviation Too Large<br>High Byte (Warn Warning Code):<br>0 No Abnormal Record<br>1 Communication Command Error<br>2 Communication Address Error<br>3 Communication Data Error<br>4 Inverter Unable to Process<br>5 Communication Transmission Timeout<br>7 Parameter Copy Error<br>8 Parameter Copy Error<br>9 IGBT Overheat Warning<br>11 PID Feedback Signal Warning<br>12 AI Current Signal Disconnection<br>15 PG Feedback Error<br>17 Overspeed Warning<br>18 Speed Deviation Too Large<br>19 Input Phase Loss<br>20 Over Torque 1<br>21 Over Torque 2<br>22 Motor Overheat<br>24 Over Slip<br>25 Parameter Identification In Progress<br>28 Output Phase Loss Warning<br>30 Different Model Parameter Copy Error<br>31 Over Torque 3<br>32 Over Torque 4<br>36 CANopen Software Disconnection 1<br>37 CANopen Software Disconnection 2<br>39 CANopen Hardware Disconnection<br>40 CANopen Index Error |

| | | |
|---|---|---|
| | 41 | CANopen Station Number Error |
| | 42 | CANopen Memory Error |

# 8. Appendix Modbus Communication

## 8.1. Communication Introduction

The CM680 series inverter uses an RS485 communication interface to connect with a PC/PLC for communication, forming a single-master multiple-slave PC/PLC communication network. Each communication slave must have a unique slave address, with the setting range for the slave address being 1 to 254, and 0 being the broadcast communication address. Users can control slaves (inverters) centrally through the master (usually a PC or PLC), achieving functions such as setting inverter operation commands, modifying and reading parameters, and reading inverter operating status and fault information through the Modbus communication protocol.

The CM680 series inverter supports Modbus-ASCII and Modbus-RTU slave station communication protocols. This communication protocol is a serial communication protocol that defines the content and format of information transmitted in serial communication. If the slave station encounters an error during information reception or cannot execute the command issued by the master station, the slave station will send a fault information back to the master station.

## 8.2. RS485 Connection Topology

The RS485 bus typically uses a daisy chain topology, where the 485+ interfaces of the master station and multiple slave stations are connected in sequence, and the 485- interfaces are connected similarly. The daisy chain topology of the RS485 bus is shown in Figure 1. The daisy chain topology has advantages such as low signal reflection, high communication success rate, and no need for additional equipment. To reduce the impact of interference on the output signal, it is recommended to use shielded twisted pair cables for RS485 bus transmission.

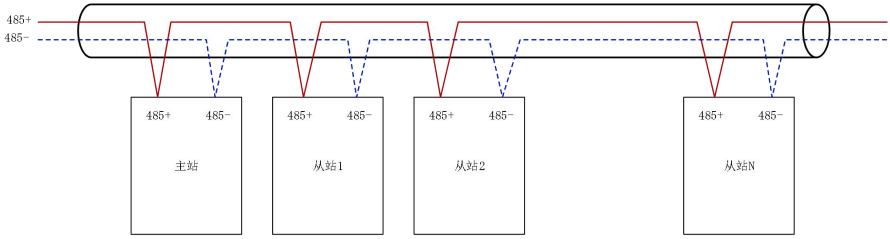



**Figure 1 Daisy Chain RS485 Bus Topology**

## 8.3. Communication Transmission Method

The RS485 communication protocol supports both half-duplex and full-duplex modes. In most cases, we use the half-duplex mode, where at any given time, only one device (master or slave) can transmit data while the other can only receive data. The RS485 communication transmission method is shown in Figure 3. Data is transmitted in the form of messages as specified in the Modbus-RTU protocol. One frame of data is sent at a time, and a new communication frame begins when the idle time on the communication line exceeds the transmission time of 3.5 bytes.

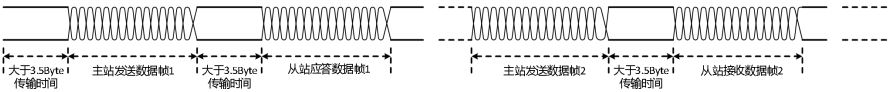



**Figure 2 RS485 Communication Transmission Method**

The CM680 Inverter is equipped with Modbus-RTU and Modbus-ASCII communication protocols, allowing users to choose which protocol to use. In the network, only one device (the master) can establish the protocol, that is, issue a “query/command.”Other devices (slaves) can only respond to the

master's "query/command" and provide data, or perform the corresponding operations as directed by the master. The master can be a PC, PLC, etc., and the slave refers to the inverter. The master can communicate independently with a specific slave or broadcast information to all slaves. When the master sends a "query/command," the slave must return a response, but if the master sends a broadcast message, the slave does not need to respond.

### 8.4. Communication Data Frame Structure

The Modbus-RTU protocol communication data format is shown in Figure 4, the inverter only supports read or write operations for Word-type parameters, the corresponding communication read command is 03; the write command is 06, the multiple write command is 10, and byte or bit read/write operations are not supported.

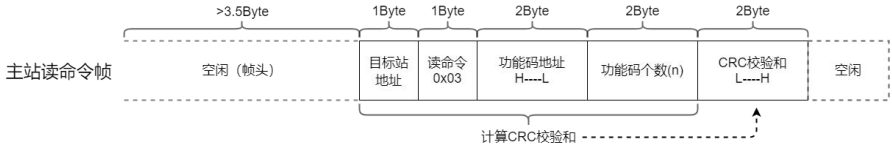



**Figure 3(a) Master Station Read Command Frame**

Note: Theoretically, the host can read multiple consecutive parameters at once (i.e., n can be up to 12), but it should be noted that it cannot exceed the last parameter of the current parameter group, otherwise, an error response will be returned.

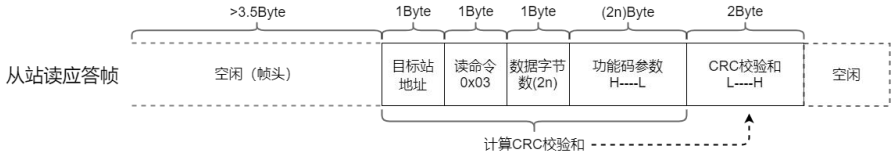



**Figure 3(b) Slave Station Read Response Frame**

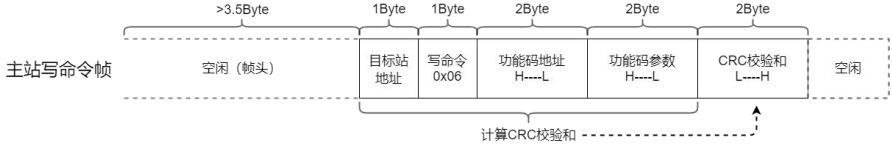



**Figure 3(c) Master Station Write Command Frame**

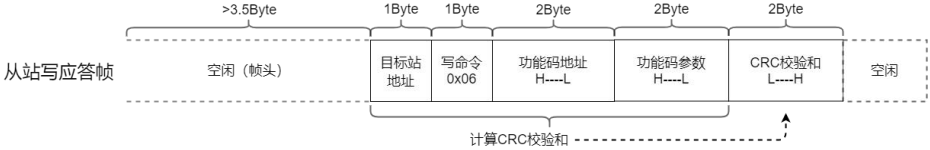



**Figure 3(d) Slave Station Write Response Frame**

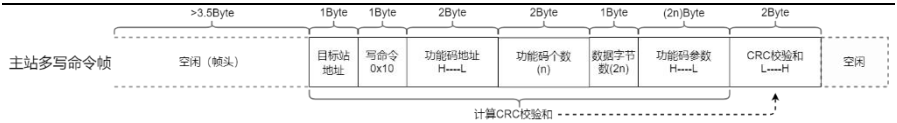



**Figure 3(e) Master Station Multiple Write Command Frame**

Note: Multiple writes and multiple reads are the same, with a maximum of 12 parameters that can be operated.

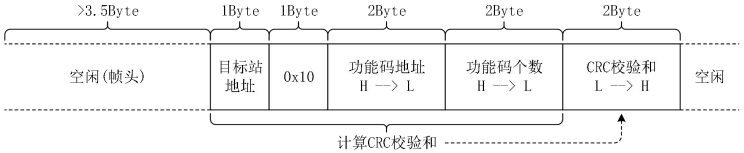



**Figure 3(f) Slave Station Multiple Write Response Frame**

The read response error command for the slave station is 0x83, the write response error command is 0x86, the multiple write response error command is 0x90, and CRC check errors do not respond. In addition, in the error types, 0x01 indicates command code error, 0x02 indicates address error, 0x03 indicates data error, and 0x04 indicates the command cannot be processed.

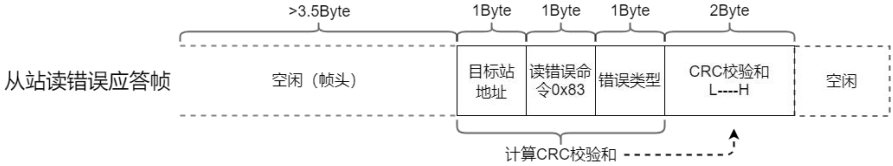



**Figure 3(g) Slave Station Read Error Response Frame**

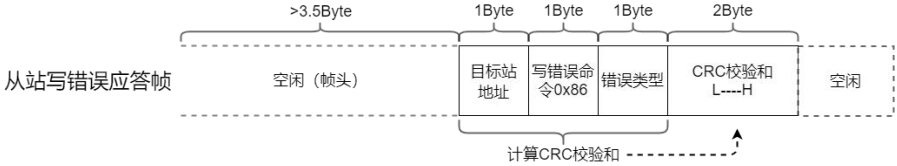



**Figure 3(h) Slave Station Write Error Response Frame**

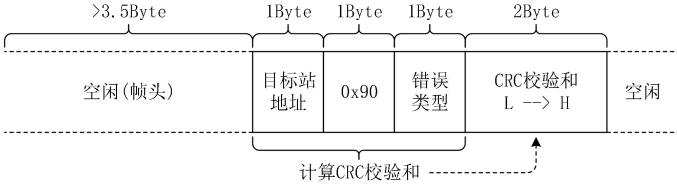



**Figure 3(i) Slave Station Multiple Write Error Response Frame**

### 8.5. Related Parameters

| Function Code | Name | Content | Default Value | Parameter Description |
|---|---|---|---|---|
| F8-00 | Baud Rate Setting | 4.8～115.2 kb/s | 115.2 | Used to set the |

| Function Code | Name | Content | Default Value | Parameter Description |
|---|---|---|---|---|
| | | | | data transfer rate between the host computer and the inverter. The higher the baud rate, the faster the communication. Note: The baud rate set on the host computer and the inverter must be the same, otherwise communication cannot be performed. |
| F8-01 | Communication Data Format | 0: 7,N,1 for ASCII<br>1: 7,N,2 for ASCII<br>2: 7,E,1 for ASCII<br>3: 7,O,1 for ASCII<br>4: 7,E,2 for ASCII<br>5: 7,O,2 for ASCII<br>6: 8,N,1 for ASCII<br>7: 8,N,2 for ASCII<br>8: 8,E,1 for ASCII<br>9: 8,O,1 for ASCII<br>10: 8,E,2 for ASCII<br>11: 8,O,2 for ASCII<br>12: 8,N,1 for RTU<br>13:8,N,2 for RTU<br>14:8,E,1 for RTU<br>15:8,O,1 for RTU<br>16:8,E,2 for RTU<br>17:8,O,2 for RTU | 12 | The data format set by the master station and the inverter must be consistent; otherwise, communication cannot proceed. |
| F8-01 | Communication Data Format | | 12 | The data format set by the master station and the inverter must be consistent; otherwise, communication cannot proceed. |
| F8-02 | Communication Address | 1～254 | 1 | Used to set the communication address of the inverter. |
| F8-03 | Response Delay | 0.0～200.0ms | 2.0 | The interval time between the end of data reception by the inverter and the sending of data to the master station.<br>If the response delay is less |

| Function Code | Name | Content | Default Value | Parameter Description |
|---|---|---|---|---|
| | | | | than the system processing time, the response delay will be based on the system processing time;<br>If the response delay is greater than the system processing time, the system will delay sending data to the master station until the response delay time is reached. |
| F8-04 | Communication Timeout Time | 0.0～100.0s | 0.0 | When this parameter is set to a non-zero value, if the interval between one communication and the next exceeds the communication timeout time, the system will report a Modbus transmission timeout (E058) fault. In most cases, it is set to zero, which means it is invalid. |

### 8.6. Communication Parameter Address

This section contains the communication content, used for controlling the inverter's operation, inverter status, and related parameter settings. R indicates read-only, and RW indicates read-write.

The parameter address is indicated by the function code group number and label:

High Byte: F0～FF groups are 0x00～0x0F, U0～U1 are 0x10～0x11, H1～H3 are 0x12～0x15, L0～L8 are 0x16～0x1E.

Low Byte: 0x00～0xFF

For example, address F0-11 is represented as 0x000B

Note:

FF Group: Neither readable nor writable parameters; U Group: Read-only, not writable parameters.

Some parameters cannot be modified when the inverter is in operation; some parameters cannot be modified regardless of the inverter's state. When changing function code parameters, pay attention to the parameter range, unit, and related notes.

| Function Code Group Number | Communication Access Address |
|---|---|
| F0～FE Groups | 0x0000～0x0FFF |
| U0～U1 Groups | 0x1000～0x11FF |
| H1～H3 Groups | 0x1200～0x15FF |
| L0～L8 Groups | 0x1600～0x1EFF |

20xx Definition:

| Address | Parameter Description | Address | Parameter Description |
|---|---|---|---|
| 2102H | Given Frequency (Unit: 0.01Hz) | 2210H | DI Switch Status, Read-Only |
| 2103H | Output Frequency (Unit: 0.01Hz), Read-Only | 2211H | DO Switch Status, Read-Only |
| 2200H | Output Current (Unit: A), Read-Only | 2212H | Multi-Speed, Read-Only |
| 2201H | Count Value, Read-Only | 2213H | DI CPU Pin Status, Read-Only |
| 2202H | Actual motor speed (unit: rpm), read-only | 2214H | DO CPU pin status, read-only |
| 2203H | Bus voltage (unit: V), read-only | 2215H | PG1 encoder pulse count, read-only |
| 2204H | Output voltage (unit: V), read-only | 2216H | Pulse input frequency (unit: Hz), read-only |
| 2205H | Power factor angle (unit: deg), read-only | 2217H | Pulse input position, read-only |
| 2206H | Output power (unit: kW), read-only | 221AH | GFF Detection Percentage (Unit: %) |
| 2207H | Actual Feedback Speed (Unit: rpm), Read-Only | 221EH | User-Defined Display |
| 2208H | Output Torque (Unit: Nm), Read-Only | 2223H | Speed Mode, Read-Only |
| 2209H | PG Position, Read-Only | 2224H | Carrier Frequency (Unit: Hz), Read-Only |
| 220AH | PID Feedback Value, Read-Only | 222CH | PG1 Pulse Low Byte (Unit: 1Hz), Read-Only |
| 220BH | AI1, read-only | 222DH | PG1 pulse high word (unit: 0.01Hz), read-only |
| 220CH | AI2, read-only | 222EH | PID setpoint, read-only |
| 220DH | AI3, read-only | 222FH | PID compensation, read-only |
| 220EH | IGBT temperature (unit: °C), read-only | 2230H | PID output frequency (unit: Hz), read-only |
| 220FH | Bus capacitor temperature (unit: °C), read-only | | |

Stop/Run Parameters Section:

| Type | Command Address | Command Content |
|---|---|---|
| Control Command Input (Write Only) | 2000H | xx12: Forward Operation xx22: Reverse Operation xx13: Forward Jog<br>xx23: Reverse Jog xxx1: Stop Operation |
| Set Frequency (Read/Write) | 2001H | Set Frequency (Unit: 0.01Hz) |

| Type | Command Address | Command Content |
|---|---|---|
| External Signal (Read/Write) | 2002H | 0001: External Fault Signal 0002: Fault Reset Signal 0004: Base Fault Signal |
| Inverter Fault Address (Read Only) | 2100H | xx01: Overcurrent During Acceleration xx33: External Interrupt<br>xx02: Overcurrent During Deceleration xx34: Password Entered Incorrectly Three Times<br>xx03: Overcurrent During Constant Speed xx36: Invalid Communication Command<br>xx04: Ground Protection xx37: Invalid Communication Address<br><br>xx05: IGBT Upper and Lower Bridge Short Circuit xx38: Communication Data Error<br>xx06: Overcurrent During Stop xx39: Communication Write to Read-Only Address<br>xx07: Overvoltage During Acceleration xx3A: Modbus Transmission Timeout<br>xx08: Overvoltage During Deceleration xx3E: Regenerative Braking Action During Deceleration<br>xx09: Overvoltage During Constant Speed xx3F: Excessive Slip<br><br>xx0A: Overvoltage During Stop xx40: Please reset the machine type code<br>xx0B: Undervoltage during acceleration xx41: PG card hardware error<br>xx0C: Undervoltage during deceleration xx45: Feedback speed divergence<br>xx0D: Undervoltage during constant speed xx46: Feedback speed deviation too large<br>xx0E: Undervoltage during stop xx48: STO1 fault<br><br>xx0F: Input phase loss protection xx4C:STO<br>xx10: IGBT temperature too high xx4D: STO2 fault<br>xx11: Ambient temperature too high xx4F: U phase overcurrent<br><br>xx15: Inverter overload xx50: V phase overcurrent<br><br>xx16: Motor 1 overload protection xx51:W Phase Overcurrent<br><br>xx17:Motor 2 Overload Protection xx52:U Phase Output Under<br><br>xx18:Motor Overheat PTC xx53:V Phase Output Phase Loss<br><br>xx1A:Over Torque 1 xx54:W Phase Output Phase Loss<br><br>xx1B:Over Torque 2 xx57:Low Frequency Overload Protection<br>xx1C:Low Current xx65:CANopen Disconnection<br><br>xx1D:Limit Reached xx66:CANopen Disconnection<br><br>xx1F:Memory Read Error xx68:CANopen Hardware Disconnection<br>xx21:U Phase Current Detection Error xx69:CANopen Index Error<br><br>xx22:V Phase Current Detection Error xx6A:CANopen Station Number Error<br>xx23:W Phase Current Detection Error xx6B:CANopen Memory Error<br>xx24:cc Hardware Circuit Abnormality xx78:EMS Overcurrent |
| Inverter Fault Address (Read Only) | 2100H | |

| Type | Command Address | Command Content |
|---|---|---|
| | | xx25:oc Hardware Circuit Abnormality xx80:Over Torque 3<br><br>xx26:ov Hardware Circuit Abnormality xx81:Over Torque 4<br><br>xx28:Motor Parameter Identification Error xx86:Motor 3 Overload Protection<br>xx29:PID Disconnection AI2 xx87:Motor 4 Overload Protection<br>xx2A:PG Feedback Setting Error xx8D:Pre-Operation Ground Fault<br>xx2B:PG Feedback Disconnection xx8E:Parameter Identification Error 1<br>xx2C:PG Feedback Stall xx8F: Parameter Identification Error 2<br>xx2D: PG Slip Abnormality xx92: Broken Belt Detection<br><br>xx31: External Terminal Abnormality xx93: Tension PID Feedback Error Detection<br>xx32: External Terminal Emergency Stop |
| | | 01xx: Communication Command Error 16xx: Motor Overheating<br><br>02xx: Communication Address Error 18xx: Over Slip<br><br>03xx: Communication Data Error 19xx: Parameter Identification In Progress<br>04xx: Inverter Unable to Process 1Cxx: Output Phase Loss Warning<br><br>05xx: Communication Transmission Timeout 1Exx: Different Model Copy Error<br>07xx: Parameter Copy Error 1Fxx: Over Torque 3<br>08xx: Parameter Copy Error 20xx: Over Torque 4<br>09xx:IGBT Overheat Warning 24xx:CANopen Software Disconnection<br><br>0Axx:Environmental Overheat Warning 25xx:CANopen Software Disconnection<br>0Bxx:PID Feedback Signal Warning 27xx:CANopen Hardware Disconnection<br>0Dxx:Low Current Warning 28xx:CANopen Index Error<br><br>0Fxx:PG Feedback Error 29xx:CANopen Node Address Error<br><br>11xx:Overspeed Warning 2Axx:CANopen Memory Error<br><br>12xx:Excessive Speed Deviation 2Bxx:CANopen SDO Transmission Timeout<br>13xx:Input Phase Loss 2Cxx:CANopen SDO Reception Overflow<br><br>14xx:Over Torque 12Dxx:CANopen Initialization Error<br><br>15xx:Over Torque 22Exx:CANopen Format Error |

60xx definition

| command address | bit | bit name | value | Parameter Description |
|---|---|---|---|---|

| command address | bit | bit name | value | Parameter Description |
|---|---|---|---|---|
| 6000h （RW） | 0 | CMD_ACT | 0 | fcmd =0 |
| | | | 1 | fcmd = Fset(Fpid) |
| | 1 | EXT_CMD1 | 0 | forward direction command |
| | | | 1 | reverse direction command |
| | 2 | - | - | - |
| | 3 | HALT | 0 | continue to target speed |
| | | | 1 | temporarily stop according to deceleration settings |
| | 4 | LOCK | 0 | continue to target speed |
| | | | 1 | Maintain Current Frequency |
| | 5 | JOG | 0 | JOG OFF |
| | | | Pulse 1 | JOG RUN |
| | 6 | QSTOP | 0 | None |
| | | | 1 | Quick Stop |
| | 7 | SERVO_ON | 0 | Servo OFF |
| | | | 1 | Servo ON |
| | 11~8 | GEAR | 0000 | Main Speed |
| | | | 0001~ 1111 | 1~15 Segment Speed Frequency Switching |
| | 13~12 | ACC/DEC | 00 | First Segment Acceleration/Deceleration Time |
| | | | 01 | Second Segment Acceleration/Deceleration Time |
| | | | 10 | Third Segment Acceleration/Deceleration Time |
| | | | 11 | Fourth Segment Acceleration/Deceleration Time |
| | 14 | EN_SW | 0 | Do Not Allow Multi-Segment Command and Acceleration/Deceleration Time Switching |
| | | | 1 | Allow Multi-Segment Command and Acceleration/Deceleration Time Switching |
| | 15 | RST | Pulse 1 | Clear Error Code |
| 6002h （RW） | 15-0 | Velocity Cmd | | Frequency Command (0.01Hz) |
| 6100h (R) | 0 | ARRIVE | 0 | Frequency Command Not Reached |
| | | | 1 | Frequency Command Arrival |
| | 1 | DIR | 0 | Forward |
| | | | 1 | Reverse |
| | 2 | WARN | 0 | No Warning |
| | | | 1 | Warning Occurred |
| | 3 | ERROR | 0 | No Error |
| | | | 1 | Error Occurred |
| | 5 | JOG | 0 | None |
| | | | 1 | On JOG |
| | 6 | QSTOP | 0 | None |
| | | | 1 | On Quick Stop |
| | 7 | SERVO_ON | 0 | PWM OFF |
| | | | 1 | PWM ON |
| | 8 | Ready | 0 | Ready OFF |
| | | | 1 | Ready ON |
| | 15-9 | - | - | - |
| 6102h （R） | 15-0 | Velocity cmd | | Actual Output Frequency |