INOVANCE

FORWARD, ALWAYS PROGRESSING



MD880 Series AC Drive

Communication Guide

Industrial Automation | Intelligent Elevator | New Energy Vehicle | Industrial Robot | Rail Transit

Data code 19012451A01

Preface

Introduction

The MD880 is developed based on the high-end frequency control technology platform of Inovance. The MD880 products are available in two topological structures, namely, single-drive and multi-drive, and can be air-cooled or water-cooled. The MD880 series is designed for high-end drive applications, featuring high-performance speed and torque control, high reliability, flexible system integration, convenient commissioning and maintenance, and high power density. It is widely applicable in industries such as metallurgy, petroleum, papermaking, cranes, marine, shield tunneling, and test benches.

This guide describes the communication overview, specifications, networking, and communication configuration of the product.

Note

- The drawings in this guide are shown for illustration only and may be different from the product. Actual products may vary.
- The user guide is subject to change due to product upgrade, specification modification as well as efforts to the accuracy and convenience of the guide.

More Data

Name	Data Code	Description
MD880 Series AC Drive Communication Guide (this guide)	19012451	Describes the communication overview, specifications, networking, and communication configuration of the product.
MD880-40 Series Active Rectifier Function Guide	PS00014534	Describes the function applications, fault codes, and parameters of the product.
MD880-40 Series Active Rectifier Commissioning Guide	PS00012614	Describes the commissioning tool, commissioning flow, specific commissioning procedure, and parameters of the product.
MD880LC Series AC Drive (Multi-drive System) Hardware Guide	19012447	Describes the system composition, technical specifications, components, dimensions, options (including installation accessories, cables, and peripheral electrical components), certifications, and standards.

Name	Data Code	Description
MD880LC Series AC Drive (Multi-drive System) Installation and Maintenance Guide	19012448	Describes preparations before installation, unpacking and transportation, mechanical installation, electrical installation, daily maintenance, and component replacement.
MD880-40 Series PCS Function Guide	19012449	Describes the function applications, fault codes, and parameters of the product.
MD880-40 Series PCS Commissioning Guide	19012450	Describes the commissioning tool, commissioning flow, specific commissioning procedure, and parameters of the product.
MD880 Series AC Drive Function Guide (HCU-50S, HCU-51S, HCU-50S-CS, and HCU-51S-CS)	19012581	Describes the function applications and troubleshooting of the product.
MD880 Series AC Drive Commissioning Guide (HCU-50S, HCU-51S, HCU-50S-CS, and HCU-51S-CS)	19012580	Describes the commissioning tool, commissioning flow, specific commissioning procedure, and parameters of the product.
MD880 Series AC Drive Parameter Guide (HCU-50S, HCU-51S, HCU-50S-CS, and HCU-51S-CS)	19012541	Describes the parameters and fault codes of the product.

Revision History

Date	Version	Description
December 2024	A01	**Addition:** • Added the "EtherNet/IP Communication" section (*"11.1 Overview" on page 105*). • Added the "EtherNet/IP Communication Configuration Example" section (*"12.7.1 Example of Using HEIP-10 with AB 1769-L71 as Master" on page 170*). • Added the "EtherNet/IP Communication Troubleshooting" section (*" EtherNet/IP" on page 196*). **Update:** Updated the "Communication Overview" section (*" Communication Overview" on page 17*).
January 2024	A00	First release.

Access to the Guide

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

- Visit *www.inovance.com*, go to Support > Download, search by keyword, and then download the PDF file.
- Scan the QR code on the product with your smart phone.
- Scan the QR code below to install the My Inovance app, and search for the file in the app.

Warranty

Inovance provides warranty service within the warranty period (as specified in your order) for any fault or damage that is not caused by improper operation of the user. A maintenance fee will be charged beyond the warranty period.

Even in the warranty period, a maintenance fee will be charged for repair of the following damage:

- Damage caused by operations not following the instructions in the guide
- Damage caused by fire, flood, or abnormal voltage
- Damage caused by unintended use of the product
- Damage caused by use beyond the specified scope of application of the product
- Damage or secondary damage caused by force majeure (such as natural disaster, earthquake, and lightning strike)

The maintenance fee will be charged according to our latest Price List if not otherwise agreed upon.

For details, see the Product Warranty Card.

Table of Contents

Preface 1
Fundamental Safety Instructions 8
1 Overview of Communication Expansion Types 16
2 Communication Overview 17
2.1 Communication Expansion Topology 17
2.2 Fieldbus Adapters 18
2.3 Communication Base Value 20
2.4 Address 21
2.5 Quick Double-Word Data Configuration 22
2.6 Related Parameters 25
3 Internal Communication 28
3.1 InoLink Communication Network 28
3.2 HOFR Optical Fiber Routing Network 32
4 Communication Expansion Module Installation 35
4.1 Installation Precautions 35
4.2 Installation Tools 35
4.3 Screw and Fastener Tightening Torque 35
4.4 Installation Method 35
4.5 Removal Method 37
5 PROFINET IO Communication 38
5.1 Overview 38
5.2 Protocol Introduction 38
5.3 Hardware Configuration 39
5.3.1 Module Introduction 39
5.3.2 Networking Topology 41
5.4 Communication Parameters 43
5.5 Configuration Procedure 43
6 PROFIBUS DP Communication 44
6.1 Overview 44
6.2 Protocol Introduction 44
6.2.1 Data Transmission Format 44
6.2.2 PKW Data 45
6.3 Hardware Configuration 47
6.3.1 Module Introduction 47

6.3.2 Communication Port 49
6.3.3 Networking Topology 50
6.4 Communication Parameters 57
6.5 Configuration Procedure 59
7 Modbus RTU Communication 61
7.1 Overview 61
7.2 Use of PZD1 61
7.3 Protocol Introduction 62
7.3.1 Modbus RTU Communication Introduction 62
7.3.2 Modbus-Specific Parameters 62
7.3.3 Modbus RTU Communication Examples 66
7.4 Hardware Configuration 67
7.4.1 Module Introduction 67
7.4.2 Communication Port 69
7.4.3 Networking Topology 70
7.5 Communication Parameters 71
7.6 Configuration Procedure 72
8 CANopen Communication 73
8.1 Overview 73
8.2 Protocol Introduction 73
8.2.1 CANopen Protocol Introduction 73
8.2.2 CANopen Predefined Connection Set 73
8.2.3 SDO Data 74
8.3 Hardware Configuration 76
8.3.1 Module Introduction 76
8.3.2 Communication Port 78
8.3.3 Networking Topology 79
8.4 Communication Parameters 80
8.5 Configuration Procedure 81
9 Modbus TCP Communication 82
9.1 Overview 82
9.2 Protocol Introduction 82
9.2.1 Modbus TCP Communication Introduction 82
9.2.2 Modbus-Specific Parameters 83
9.3 Hardware Configuration 87
9.3.1 Module Introduction 87
9.3.2 Networking Topology 89
9.4 Communication Parameters 92

9.5 Configuration Procedure 95
10 EtherCAT Communication 97
10.1 Overview 97
10.2 Protocol Introduction 97
10.2.1 EtherCAT Protocol Introduction 97
10.2.2 PDO Data 97
10.2.3 SDO Data 98
10.3 Hardware Configuration 98
10.3.1 Module Introduction 98
10.3.2 Networking Topology 100
10.4 Communication Parameters 103
10.5 Configuration Procedure 104
11 EtherNet/IP Communication 105
11.1 Overview 105
11.2 Protocol Introduction 105
11.2.1 I/O Messages 105
11.2.2 Data Sent by the Master Station 106
11.2.3 Data Returned by the AC drive 106
11.3 Hardware Configuration 106
11.3.1 Module Introduction 106
11.3.2 Networking Topology 108
11.4 Communication Parameters 110
11.5 Configuration Procedure 114
12 Examples for Communication Settings 115
12.1 PROFIBUS DP 115
12.2 PROFINET IO 124
12.3 Modbus RTU 137
12.4 CANopen 144
12.5 Modbus TCP 152
12.6 EtherCAT 159
12.7 EtherNet/IP 170
12.7.1 Example of Using HEIP-10 with AB 1769-L71 as Master 170
12.7.2 Example of Using HEIP-10 with Inovance H5U as Master 180
12.7.3 Example of Using HEIP-10 with Inovance AC810 as Master 182
13 Communication Troubleshooting 190
13.1 PROFIBUS DP 190
13.2 PROFINET IO 191
13.3 Modbus RTU 191

13.4 CANopen 192
13.5 EtherCAT 194
13.6 EtherNet/IP 196

Fundamental Safety Instructions

Safety Disclaimer

- This chapter provides essential safety instructions for proper use of the product. Before using this product, read through the user guide, especially the safety instructions. Failure to observe the safety instructions may result in equipment damage, personal injuries, or even death.
- "DANGER", "WARNING", and "CAUTION" items in this guide do not indicate all safety precautions that need to be followed; instead, they just supplement the safety precautions.
- Use this product according to the designated environment requirements. Damage caused by improper use is not covered by warranty.
- Inovance shall take no responsibility for any personal injuries or property damage caused by improper usage.

Safety Levels and Definitions



indicates that failure to comply with the notice will result in severe personal injuries or even death.



indicates that failure to comply with the notice may result in severe personal injuries or even death.



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

Safety Precautions

- Some drawings in the guide show the product without covers or protective guards to display more details. Remember to install the covers or protective guards as specified first, and then perform operations in accordance with the instructions.
- Product illustrations in this guide are for reference only. Actual products may vary.
- Operators must take mechanical precautions to protect personal safety and wear protective equipment, such as anti-smashing shoes, safety clothing, safety glasses, protective gloves, and protective sleeves.

Unpacking



- Do not install the equipment if you find damage, rust, or signs of use on the equipment or accessories upon unpacking.
- Do not install the product if you find water seepage or any components being missing or damaged upon unpacking.
- Do not install the equipment if you find the packing list does not conform to the equipment you received.



- Check whether the packing is intact and whether there is damage, water seepage, dampness, and deformation before unpacking.
- Unpack the package in accordance with the package sequence. Do not hit the package with force.
- Check whether there is damage, rust, or signs of bumping on the surface of the equipment and equipment accessories before unpacking.
- Check whether the package contents are consistent with the packing list before unpacking.

Storage and Transportation



- Large-scale or heavy equipment must be transported by qualified professionals using specialized hoisting equipment. Failure to comply may result in personal injuries or equipment damage.
- Before hoisting the equipment, ensure that components, such as the front cover and terminal blocks, are secured firmly with screws. Loosely-connected components may fall off and result in personal injuries or equipment damage.
- Never stand or stay below the equipment when the equipment is being hoisted.
- When hoisting the equipment with a steel rope, ensure the equipment is hoisted at a constant speed without suffering from vibration or shock. Do not turn the equipment over or let the equipment stay hanging in the air. Failure to comply may result in personal injuries or equipment damage.



- Handle the equipment with care during transportation and mind your steps to prevent personal injuries or equipment damage.
- When carrying the equipment with bare hands, hold the equipment casing firmly with care to prevent parts from falling. Failure to comply may result in personal injuries.
- Store and transport the equipment based on the storage and transportation requirements. Failure to comply will result in equipment damage.
- Avoid storing or transporting the equipment in environments with water splash, rain, direct sunlight, strong electric field, strong magnetic field, and strong vibration.
- Avoid storing the product for more than three months. When the product needs to be stored for an extended period, take more strict protection and necessary inspection.
- Pack the equipment strictly before transportation. Use a sealed box for long-distance transportation.
- Never transport the product with devices or materials that may damage or negatively impact the product.

Installation



- The equipment must be operated only by professionals with electrical knowledge. Non-professionals are not allowed to perform operations on the equipment.



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



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

Wiring
 • Equipment installation, wiring, maintenance, inspection, and component replacement must be performed only by professionals. • Before wiring, cut off all the power supplies of the equipment. Wait for at least the time designated on the equipment warning label before further operations because residual voltage still exists after power-off. Measure the DC voltage in the main circuit and ensure the DC voltage is in the safe range. Failure to comply will result in electric shock. • Do not perform wiring, remove the equipment cover, or touch the circuit board while power is on. Failure to comply will result in electric shock. • Check that the equipment is grounded properly. Failure to comply can result in electric shock.
 • Do not connect the input power supply to the output end of the equipment. Failure to comply can result in equipment damage or even fire. • When connecting a drive to a motor, check that the phase sequences of the drive and motor terminals are consistent to prevent reverse motor rotation. • Cables used for wiring must meet cross-sectional area and shielding requirements. The shield of the cable must be reliably grounded at one end. • Fix the terminal screws with the tightening torque specified in the guide. Improper tightening torque may overheat or damage the connecting part, resulting in fire. • After wiring, check that all cables are connected properly and no screws, washers, or exposed cables are left inside the equipment. Failure to comply may result in electric shock or equipment damage.
 • During wiring, follow proper electrostatic discharge (ESD) procedures and wear an anti-static wrist strap. Failure to comply can result in damage to the equipment or internal product circuits. • Use shielded twisted pairs for the control circuit. Connect the shield to the grounding terminal of the equipment for grounding purpose. Failure to comply will result in equipment malfunction.

Power-on



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



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

Operation



- The equipment must be operated only by professionals. Failure to comply can result in personal injuries or even death.
- Do not touch any terminal of the device or remove any part of the device during operation. Failure to comply will result in electric shock.



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

Maintenance



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



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

Repair



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



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

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

Safety Labels

To ensure safe operations, comply with safety signs on the device and do not damage or remove the safety labels. See the following table for descriptions of the safety labels.

Safety Label	Description
	• Read through the safety instructions before operating the equipment. Failure to comply may result in equipment damage, personal injuries, or even death. • Do not touch the terminals or remove the cover while power is on or within 10 minutes after power-off. Failure to comply will result in electric shock.

1 Overview of Communication Expansion Types

Communication Protocol	Expansion Module	Remarks
PROFINET IO	HPFN-10 and HPFN-20	-
PROFIBUS DP	HDP-10	-
Modbus RTU	HMBA-10	-
CANopen	HCAN-10	DS-301
Modbus TCP	HMBT-10	-
EtherCAT	HETC-10	-
EtherNet/IP	HEIP-10	-

2 Communication Overview

This chapter introduces how to connect external controllers or monitoring devices to the MD880 series AC drive through fieldbus adapters and communication expansion modules. It also covers the MD880 communication expansion topology, fieldbus adapters, communication addresses, and communication conversion.

Note

- The communication configuration is generally consistent across the MD880 series, and this guide uses the configuration of the HCU-50S/51S products as an example.
- The parameters described in this guide may vary across different products.

2.1 Communication Expansion Topology

The HCU (MD880 controller) connects to external control systems (fieldbus controllers) through configured communication expansion modules.

- The HCU has three expansion slots (SLOTs). Each SLOT supports up to three channels when an HOFM is installed, but each SLOT supports only one communication expansion channel.
- The HCU supports three fieldbus adapters (see *"2.1 Communication Expansion Topology" on page 17* for details): fieldbus adapter A (FBA-A), fieldbus adapter B (FBA-B), and fieldbus adapter C (FBA-C). The FBA-C is internally bound to Ethernet communication (used for MD880 Ethernet-based commissioning functions). Therefore, the HCU supports up to two external fieldbus channels.

The communication expansion system supports cyclic data and service data:

- Cyclic data is periodic interaction data, also known as process data. It is typically used to transmit control words, references, status words, and actual values.
- Service data is non-periodic interaction data, transmitted only after a corresponding request is executed. It is also known as mailbox data, typically used for reading and writing parameters.

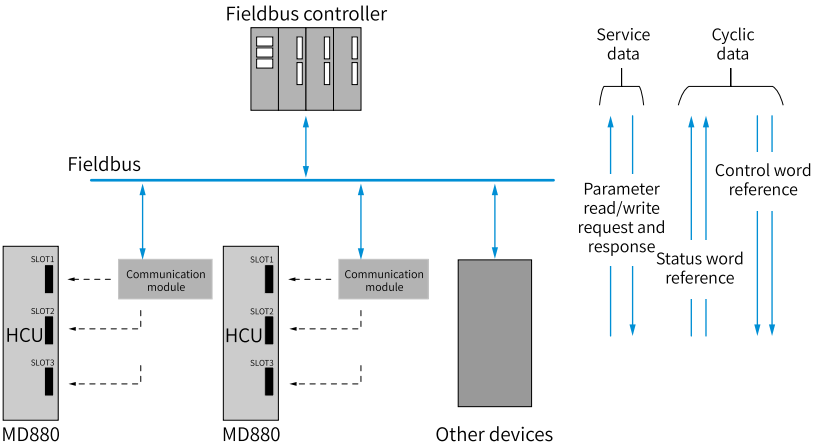


Figure 2-1 Communication expansion topology

2.2 Fieldbus Adapters

Fieldbus adapters are standardized interface modules that enable interaction between external fieldbus data and internal HCU data.

- After installing a communication module in a SLOT, you need to configure the correct SLOT in the communication protocol parameter group. You can view the communication expansion module type and software version for each SLOT in the A2 group. After configuring the SLOT, you need to set the binding relationship between the fieldbus adapter and the external communication protocol (n1 or n2 group).
- With the n1 or n2 group, you can view the input and output process data of the external communication module and configure the data conversion relationship. After conversion, received process data is stored in the connector group and then configured through control channels, reference channels, or other function modules for communication data application. The communication data exchange logic is shown in the following figure.

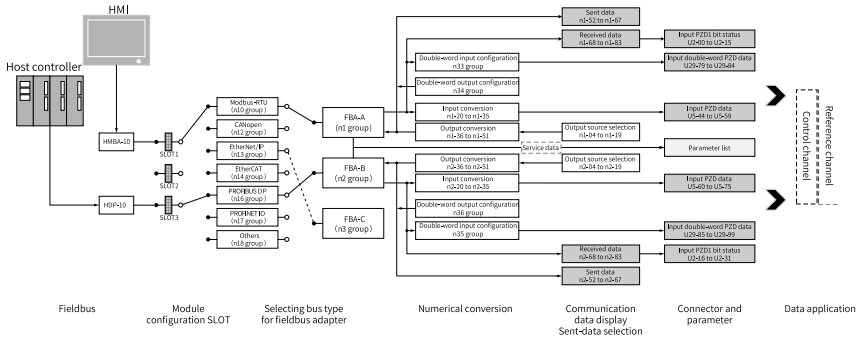


Figure 2-2 Communication data exchange logic

The following uses FBA-A as an example to describe the sending and receiving of process data.

- Receiving process data
  Parameters n1-68 to n1-83 correspond to 16 received raw process data points. For example, if the host controller sends process data PZD1 with a value of 0x0406, this value can be observed in n1-68. After base value conversion, the received data is stored in 16 connectors (U5-44 to U5-59). For base value conversion details, see *"2.3 Communication Base Value" on page 20*.

  Received data is displayed in single-word (16-bit) format. If the host controller needs to send double-word (32-bit) data, two methods are available:

  - Method 1: The received double-word data is split into two single-word data points, which must be combined using the word-to-double-word module in the L3 group before use.
  - Method 2: The received double-word data is split into two single-word data points, which are then combined according to configuration of the n33 and n35 groups. Method 2 enables more convenient and faster double-word configuration than Method 1. For details, see *"2.5 Quick Double-Word Data Configuration" on page 22*.

Note

In PROFIBUS DP communication, the check bit for process data PZD1 must be 1 (the check bit is PZD1.bit10 (default) and can be changed or disabled through n16-10). Otherwise, the HCU considers the received process data invalid, and U5-44 to U5-59 retain the data of the previous cycle, though n1-68 to n1-83 display the actually received raw data.

- Sending process data
  Parameters n1-04 to n1-19 select the process data to be sent to the host controller. When the communication base value is set to 0, no base value conversion is performed.

In this case, the sent data is the value selected by n1-04 to n1-19, and parameters n1-52 to n1-67 display the sent data values that are consistent with the data received by the PLC.

Example:

- When n1-04 is set to 1701 (U7-01 is 100% per-unit value, corresponding to hexadecimal 0x1000), [n1-52] is 0x1000.
- When n1-05 is set to 1702 (U7-02 is 200% per-unit value, corresponding to hexadecimal 0x2000), [n1-53] is 0x2000.

Note

To send double-word (32-bit) data, two methods are available:

- Method 1: Split the double-word data into two single-word (16-bit) data points using the double-word-to-word module, and then configure the two single-word data points to appropriate positions in n1-04 to n1-19.
- Method 2: Configure double-word output using the n34 or n36 group. Method 2 enables more convenient and faster double-word configuration than Method 1. For details, see *"2.5 Quick Double-Word Data Configuration" on page 22*.

2.3 Communication Base Value

Most connector data in the HCU-50S/51S control module is stored as per-unit values, typically with the base values set to the rated parameters of the motor. When the control module interacts with the host controller through a fieldbus adapter, data scaling may be inconsistent. The fieldbus adapter provides a communication base value (the value corresponding to 100%) for each process data channel. Adjusting the communication base value allows scaling of received and sent data.

Base value conversion enables matching between the per-unit values of the AC drive and the per-unit values of the host controller, as well as proportional transformation of data sent or received by the host controller.

The factory default communication base value for the HCU-50S/51S is 0, meaning no numerical conversion is applied, and data sent/received between the PLC and HCU-50S/51S is identical.

For data composed of multiple bits, for example, control words sent by the PLC or status words uploaded by the HCU-50S/51S, base value conversion should not be applied, and the corresponding communication base value parameter should be set to 0.

The formula for calculating the communication base value is as follows:

$$\text{Communication base value} = \frac{\text{Motor per-unit base value}}{\text{PLC per-unit base value}} \times \text{PLC communication base value}$$

In the HCU-50S/51S system, the system per-unit base value is typically the motor per-unit base value, namely, the rated value of the motor, corresponding to parameters d3-00 to d3-05. The host controller communication base value and per-unit base value describe the data representation used by the host controller during communication.

- For example, if 16384 represents a motor current of 50 A, the host controller per-unit base value is 50, and the communication base value is 16384.
- If 500 represents a motor current of 50 A (equivalent to retaining one decimal place), the host controller per-unit base value is 50, and the communication base value is 500.

Communication base value parameters are used to adjust the HCU-50S/51S internal data format to match the communication format of the host controller. To modify the communication base value, you may manually calculate it according to the preceding formula or set it using the "PZD config" page in InoDriveStudio. Navigate to "Inverter" > "Config" > "PZD config" > "Adapter A I/O data config" > "Calculate base value" to access the corresponding window.

For example, in the host controller system, 16384 represents 3000 rpm, whereas in the HCU, d3-02 is 1500 rpm. In the InoDriveStudio communication base value calculator, enter 16384 for the counterpart communication base value (host controller communication base value) and 3000 for the counterpart per-unit value (host controller speed per-unit value). Clicking "Calculate" yields a communication base value of 8192 for the corresponding speed transmission channel. Set the process data communication base value parameter to 8192, which indicates that a host controller communication data value of 8192 corresponds to 1500 rpm (the speed per-unit base value of the HCU).

2.4 Address

Communication addresses, also known as parameter addresses, are used for service data (non-periodic) interactions, commonly for writing or reading parameters. For example, in PROFIBUS DP communication, parameters are read/written through PKW, and in CANopen communication, parameters are read/written by SDO.

For a MD880 series product, communication addresses can be obtained from the corresponding parameter guide or from the address column of the parameter list in InoDriveStudio.

- Parameter guide

b0: Channel 1 Start/Stop Control Word Setting

Param.	Name	Address	Range	Default	Data Type	Unit	Accuracy	User Permission
b0-00	Start/Stop control word source	90	See "Values".	0	UInt16	-	1	Ordinary user
	Values: 0: Terminal start/stop module 1: Fieldbus adapter A 2: Fieldbus adapter B 3: User-defined (b0 group)							
	Parameter description: Sets the start/stop control word source for control channel 1. Value description: 0: Terminal start/stop module. The start/stop command source is the terminal start/stop module configured by the b4 group. 1: Fieldbus adapter A. The start/stop command source is fieldbus adapter A. 2: Fieldbus adapter B. The start/stop command source is fieldbus adapter B. 3: User-defined (b0 group). The start/stop command source is defined by b0-01 to b0-08.							

2.5 Quick Double-Word Data Configuration

Process data transmission uses single-word (16-bit) format. If the host controller needs to send double-word (32-bit) data, you may configure FBA-A using the n33 and n34 groups, and FBA-B using the n34 and n35 groups. This section explains the double-word data configuration for FBA-A. The configuration method for FBA-B is identical.

- Double-word input process data configuration
  For example, the drive processes PZD1+2 double-word input according to the logic shown in *"Figure 2–3 FBA-A double-word input data configuration flowchart" on page 23*. The double-word sent by the host controller is split into two single words, PZD1 and PZD2, which then go through high/low byte conversion and are combined into double-word. The combined double-word goes through high/low word conversion and base value conversion, and is finally stored in the connector.

  Parameter configuration steps:

  1. Set n1-93 to 0 (default) to disable high/low byte conversion for PZD1 and PZD2.
  2. Set n33-00 to 0x0001 to enable PZD1+2 double-word input.
  3. Set n33-01 to 0x0000 (default) to disable high/low word conversion. That is, PZD1 is the low word and PZD2 is the high word.
  4. Set n33-02 to 0 (default) to disable base value conversion.

Note

The high/low byte conversion in Step 1 and high/low word conversion in Step 3 are for aligning with the data format of the host controller and should be set based on actual application requirements.

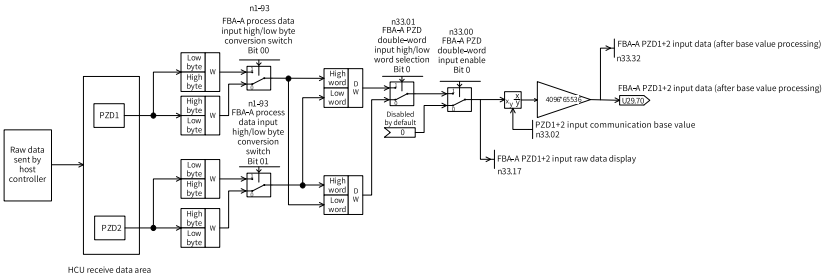


Figure 2-3 FBA-A double-word input data configuration flowchart

Table 2–1 FBA-A process data double-word input parameters

Param.	Definition	Description
n1-93	FBA-A process data input high/ low byte conversion switch	The high/low byte conversion switch for PZD1 to PZD16 process data inputs, which correspond to bit 0 to bit 15
n33-00	FBA-A PZD double-word input enable	The switch to enable PZD1+2 to PZD15+16 double-word inputs, which correspond to bit 0 to bit 14
n33-01	FBA-A PZD double-word input high/low word selection	The high/low word conversion switch for PZD1+2 to PZD15+16 double-word inputs, which correspond to bit 0 to bit 14
n33-02 to n33-16	FBA-A double-word PZD input communication base value	When this parameter is set to 0, base value conversion is disabled.
n33-17 to n33-31	FBA-A double-word input raw data display	Direction: Host controller to drive
n33-32 to n33-46	FBA-A double-word input data display (after base value processing)	Direction: Host controller to drive

- Double-word output process data configuration
  For example, the drive processes PZD1+2 double-word output according to the logic shown in the following figure. The double-word output source goes through base value conversion and then high/low word conversion to be split into two single words. Each of the single words goes through high/low byte conversion and then is stored in the corresponding position of n1-52 to n1-67 for display. The values in n1-52 to n1-67 are the final data sent to the host controller.

  Parameter configuration steps:

  1. Set n34-00 to 0x0001 to enable PZD1+2 double-word output.
  2. Set n34-01 to 0x0000 (default) to disable high/low word conversion. That is, PZD1 is the low word and PZD2 is the high word.

3. Set n1-92 to 0 (default) to disable high/low byte conversion for PZD1 and PZD2.
4. Set n34-17 to 0 (default) to disable base value conversion.
5. Set n34-02 to the data to be sent, bound through the connector.

Note

- The high/low byte conversion in Step 2 and high/low word conversion in Step 3 are for aligning with the data format of the host controller and should be set based on actual application requirements.
- Conflicts may occur when double-word output is enabled. For example, it is not allowed to enable both PZD1+2 and PZD2+3 double-word outputs, because they share PZD2.

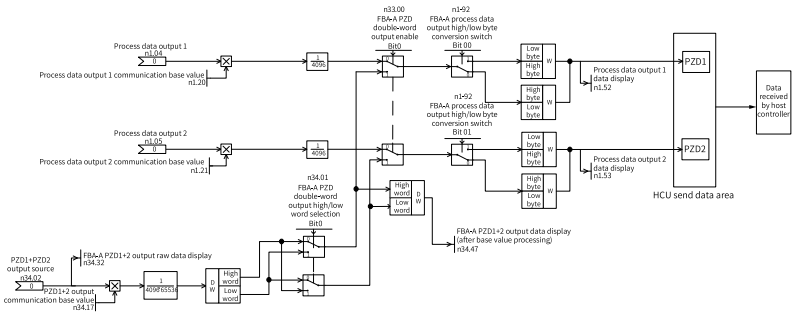


Figure 2-4 FBA-A double-word output data configuration flowchart

Table 2–2 FBA-A process data double-word output parameters

Param.	Definition	Description
n1-92	FBA-A process data output high/ low byte conversion switch	The high/low byte conversion switch for PZD1 to PZD16 process data outputs, which correspond to bit 0 to bit 15
n34-00	FBA-A PZD double-word output enable	The switch to enable PZD1+2 to PZD15+16 double-word outputs, which correspond to bit 0 to bit 14
n34-01	FBA-A PZD double-word output high/low word selection	The high/low word conversion switch for PZD1+2 to PZD15+16 double-word outputs, which correspond to bit 0 to bit 14
n34-02 to n34-16	FBA-A double-word output source	Direction: Drive to host controller
n34-17 to n34-31	FBA-A double-word output communication base value	When this parameter is set to 0, base value conversion is disabled and raw data is output.

Param.	Definition	Description
n34-32 to n34-46	FBA-A double-word output raw data display	Direction: Drive to host controller
n34-47 to n34-61	FBA-A double-word output data display (after base value processing)	Direction: Drive to host controller

2.6 Related Parameters

- FBA-A parameters

Param.	Definition	Description
n1-00	Selects communication type for FBA-A	-
n1-01	Sets disconnection detection time for FBA-A	When this parameter is set to 0, disconnection is reported as warning.
n1-02	0: Standard mode 1: No fault prompt	When this parameter is set to 1 (no fault prompt), all communication-related faults are not reported.
n1-04 to n1-19	Sets data source for FBA-A output process data 1 to 16	Direction: Drive to host controller
n1-20 to n1-35	Sets data conversion base value for FBA-A output process data 1 to 16	When this parameter is set to 0, data conversion is disabled and raw data is output.
n1-36 to n1-51	Sets data conversion base value for FBA-A input process data 1 to 16	When this parameter is set to 0, data conversion is disabled.
n1-52 to n1-67	Displays FBA-A output data values	Direction: Drive to host controller
n1-68 to n1-83	Displays FBA-A input data values	Direction: Host controller to drive
n1-84 and n1-85	Indicates the active source of the current output process data	-
n1-90	Sets whether to save mailbox data write parameter operations to EEPROM	-
n1-91	Sets whether to upload device diagnosis information	-
n1-92	FBA-A process data output high/low byte conversion switch	The high/low byte conversion switch for PZD1 to PZD16 outputs, which correspond to bit 0 to bit 15. This switch is used to enable high/low byte conversion for sent data.

Param.	Definition	Description
n1-93	FBA-A process data input high/ low byte conversion switch	The high/low byte conversion switch for PZD1 to PZD16 inputs, which correspond to bit 0 to bit 15. This switch is used to enable high/low byte conversion for received data.
n33 group	FBA-A input process data double-word configuration	-
n34 group	FBA-A output process data double-word configuration	-
U2-00 to U2-15	FBA-A input PZD1 bit statuses	Connector
U5-44 to U5-59	FBA-A input PZD value	Connector; value after base value conversion
U29-70 to U29-84	FBA-A input PZD double-word data value	Connector; value after base value conversion

- FBA-B parameters

Param.	Definition	Description
n2-00	Selects communication type for FBA-B	-
n2-01	Sets disconnection detection time for FBA-B	When this parameter is set to 0, disconnection is reported as warning.
n2-02	0: Standard mode 1: No fault prompt	When this parameter is set to 1 (no fault prompt), all communication-related faults are not reported.
n2-04 to n2-19	Sets data source for FBA-B output process data 1 to 16	Direction: Drive to host controller
n2-20 to n2-35	Sets data conversion base value for FBA-B output process data 1 to 16	When this parameter is set to 0, data conversion is disabled and raw data is output.
n2-36 to n2-51	Sets data conversion base value for FBA-B input process data 1 to 16	When this parameter is set to 0, data conversion is disabled.
n2-52 to n2-67	Displays FBA-B output data values	Direction: Drive to host controller
n2-68 to n2-83	Displays FBA-B input data values	Direction: Host controller to drive
n2-84 and n2-85	Indicates the active source of the current output process data	-
n2-90	Sets whether to save mailbox data write parameter operations to EEPROM	-

Param.	Definition	Description
n2-91	Sets whether to upload device diagnosis information	-
n2-92	FBA-B process data output high/low byte conversion switch	The high/low byte conversion switch for PZD1 to PZD16 outputs, which correspond to bit 0 to bit 15. This switch is used to enable high/low byte conversion for sent data.
n2-93	FBA-B process data input high/low byte conversion switch	The high/low byte conversion switch for PZD1 to PZD16 inputs, which correspond to bit 0 to bit 15. This switch is used to enable high/low byte conversion for received data.
n35 group	FBA-B input process data double-word configuration	-
n36 group	FBA-B output process data double-word configuration	-
U2-16 to U2-31	FBA-B input PZD1 bit status	Connector
U5-60 to U5-75	FBA-B input PZD value	Connector; value after base value conversion
U29-85 to U29-99	FBA-B input PZD double-word data value	Connector; value after base value conversion

3 Internal Communication

The MD880 series AC drive supports data exchange among multiple HCU controllers through the InoLink network or the HOFR optical fiber routing network. The InoLink network is a built-in communication network of the HCU, requiring only cable cascading between HCUs. The HOFR optical fiber routing network requires an HOFR optical fiber router and HOFM optical fiber expansion modules.

The InoLink network uses an RS485 physical link and is suitable for general master-slave communication scenarios. The HOFR optical fiber routing network uses an optical fiber physical link and is suitable for master-slave communication scenarios requiring higher reliability for communication data.

3.1 InoLink Communication Network

- InoLink communication network
  InoLink is used for master-slave communication among multiple drive units. Master-slave communication refers to a network system where the logical master station sends information (such as control bits and references) to logical slave stations.

  The InoLink link is an RS485 physical link that uses a hand-in-hand cascaded network topology, with termination resistors required at both ends.

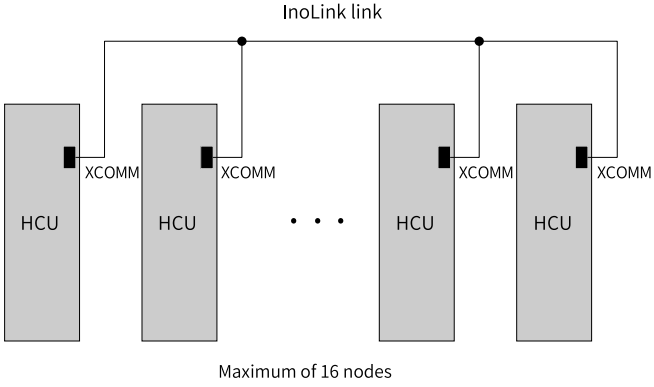


Figure 3-1 InoLink connection diagram

- Communication mechanism
  The existing InoLink communication network supports a maximum of 16 controllers. The controller numbered 1 is the master controller that initiates InoLink communication, and the controllers numbered from 2 to 16 are slave controllers. After the master controller initiates master-slave communication, the slave controllers respond in sequence. Data sent by each controller is received by all other controllers in the network. Therefore, in the InoLink communication network, one node must be

configured as node 1; otherwise, InoLink communication scheduling cannot start properly.

In InoLink communication, the communication master station is not necessarily the logical master station. The logical master station sends information to logical slave stations (other controllers than the master controller), which then use the received information for control purposes. Each logical slave station can set its data reception source, which can be one or more logical master stations.

An InoLink communication network may contain one or more master-slave networks. A master-slave network is a communication network composed of logical master and slave stations.

- Communication specifications
  The maximum communication rate for InoLink is 5 Mbps, and the maximum data exchange cycle is 2 ms. All controllers in the InoLink communication network exchange data once during each exchange cycle.

  The InoLink communication rate, data exchange cycle, total number of nodes, and number of data points sent by nodes can be configured based on actual needs. The InoLink module automatically determines if the data volume exceeds the communication bandwidth. If yes, n0-01 will prompt "excessive data exchange volume." In this case, you need to adjust the communication configuration by increasing the communication rate, reducing communication data, extending the exchange cycle, or reducing the number of nodes to ensure normal communication. See the following table for common communication configurations.

Mode	Rate	Nodes	Cycle	Data Volume
n0-06 = 0 Standard mode	5 Mbps	16	2 ms	8
n0-06 = 1 User-defined mode	5 Mbps	8	1 ms	8
		4	0.5 ms	8
		2	0.2 ms	8
	2 Mbps	8	2 ms	4
		4	1 ms	4
		2	0.5 ms	8
	500 kbps	4	2 ms	8
		2	1 ms	8
		2	2 ms	8

- Fault determination mechanism
  - If a controller receives no data from any controller within 2 seconds, it reports an offline error.
  - If a controller receives no data from the logical master station within 2 seconds, it reports a logical master station offline error.

- If the ID of a new controller is the same as that of an existing controller, the new controller reports an address conflict error.

- Parameter configuration

Param.	Name	Range	Description
n0-00	InoLink communication enable	0: Disable 1: Enable Other: B connector	Enables the InoLink communication function.
n0-01	InoLink status	0: Disable 1: Node address conflict 2: Baud rate mismatch 3: Bus disconnected or bus error 4: Communication normal 5: Listen-only mode 6: Communication unreliable (excessive CRC errors) 7: Excessive data exchange 8: Waiting	Displays current InoLink status.
n0-02	Local node address	0 to 16	0: Local node invalid, no InoLink communication 1: Communication scheduling master station 2 to 16: Other communication nodes
n0-03	InoLink fault mode	0: Standard mode 1: No fault prompt Other: B connector	-
n0-04	InoLink node restart judgment time	0.040s to 2.000s	-
n0-06	InoLink communication mode	0: Standard mode 1: User-defined mode	In standard mode, n0-08 configuration is invalid. In user-defined mode, n0-08 configuration is valid.
n0-07	InoLink actual effective communication mode	0: Standard mode 1: User-defined mode	-

Param.	Name	Range	Description
n0-08	InoLink user-defined communication mode configuration	Ones: InoLink communication rate 0: Follow master 1: 5 Mbps 2: 2 Mbps 3: 1 Mbps 4: 500 kbps Tens: Total number of InoLink nodes 0: 2 1: 4 2: 8 3: 10 4: 12 5: 16 Hundreds: InoLink process data exchange volume 0: 1 1: 2 2: 4 3: 8 Thousands: InoLink data exchange cycle 0: 2 ms 1: 1 ms 2: 0.5 ms 3: 0.2 ms	Configures communication parameters for use-defined mode.
n0-09	InoLink network baud rate	0: To be determined 1: 5 Mbps 2: 2 Mbps 3: 1 Mbps 4: 500 kbps	Displays the current communication rate.
n0-10	InoLink data transmission bit width	0: 8 bits	Displays the parameter.
n0-16	Total number of InoLink nodes	-	Displays the parameter.
n0-17	InoLink process data exchange volume	-	Displays the parameter.
n0-18	InoLink data exchange cycle	0: 2 ms 1: 1 ms 2: 0.5 ms 3: 0.2 ms	-

Param.	Name	Range	Description
n0-19	InoLink auxiliary function setting	Ones: Synchronization time base enable 0: Disable 1: Enable	Supports synchronization of internal scheduling timing for multiple HCUs through InoLink.
n0-20 to n0-27	[Send data 1] Selection to [Send data 8] Selection	0: 0 Other: K connector	Selects the data to be sent by the node.
n0-40 n0-42 n0-44 n0-46 n0-48 n0-50 n0-52 n0-54	[Receive data 1] Source node selection to [Receive data 8] Source node selection	0: Disable 1: Node 1 2: Node 2 16: Node 16	Sets the node address from which data is to be received.
n0-41 n0-43 n0-45 n0-47 n0-49 n0-51 n0-53 n0-55	[Receive data 1] Data selection to [Receive data 8] Data selection	1: Data 1 2: Data 2 3: Data 3 4: Data 4 5: Data 5 6: Data 6 7: Data 7 8: Data 8	Sets the data index of the specified node.
n0-70 to n0-77	[Receive data 1] Factor to [Receive data 8] Factor	–8.000 to +8.000	Sets the scaling factor for received data.
U2-32 to U2-47	InoLink receive data 1 bit status		Connector data. Displays bit statuses of InoLink receive data 1, as values before scaling.
U5-32 to U5-39	InoLink receive data 1 to InoLink receive data 8		Connector data. Displays InoLink receive data 1 to 8, as values after scaling.

3.2 HOFR Optical Fiber Routing Network

The purpose of optical fiber synchronization is similar to the InoLink communication network, both aimed at enabling data exchange among multiple drive devices, differing only in implementation. Optical fiber synchronization requires fiber synchronization expansion modules. Every HCU-50S/51S device connects to the fiber synchronization expansion module through an HOFM optical fiber module. The fiber synchronization expansion module functions like a data switch, enabling data exchange among multiple

drive devices. Unlike InoLink, optical fiber synchronization does not involve any master but operates as a data switch.

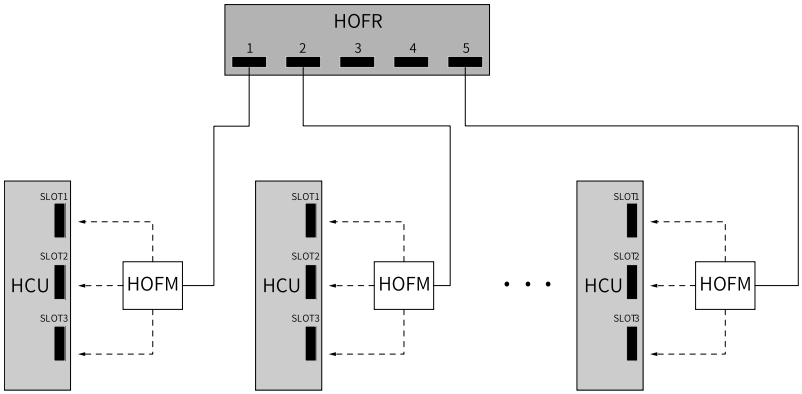


Figure 3-2 HOFR optical fiber routing network diagram

The entire data network supports a maximum of 11 nodes, with each node capable of sending and receiving eight data points. Each node can freely select data 1 to 8 from any three other nodes, combining them to form the node's received data PZD1 to PZD8 for further control and processing. The received data can also have custom scaling factors, with a range of –8.000 to 8.000, allowing flexible adaptation to matching requirements.

Table 3–1 Optical fiber router parameters

Item	Optical Fiber Sync Communication Setting	Description
1	n22-00, n22-01, n22-05	Enables communication node, sets communication interruption judgment time, and configures local node address.
2	n22-02, n22-03	Sets the slot connected to the optical fiber synchronization expansion module and its online status.
3	n22-04	Displays online nodes in the communication network to facilitate configuring the local node. Bits 0 to 10 correspond to node addresses 1 to 11.
4	n22-06 to n22-08	Selects three nodes from which data is to be received.
5	n22-09 to n22-16	Sets which data (data 1 to data 8) from which node (any one of the three selected nodes) to receive.
6	n22-17 to n22-24	Sets the scaling factor for received data. The final data used by the node is the received data multiplied by this factor.
7	n22-25 to n22-32	Selects the data to be sent by the node.

Item	Optical Fiber Sync Communication Setting	Description
8	U0-84 to U0-99	Connector data. Displays bit statuses of receive data 1.
9	U7-55 to U7-62	Connector data. Displays values of receive data 1 to 8, as values after scaling.

Note

The HOFR optical fiber routing network module is only applicable to the HCU-50S and HCU-51S series.

4 Communication Expansion Module Installation

4.1 Installation Precautions



- Ensure that the HCU control module is de-energized during the installation or removal of the communication expansion module to avoid damage to the communication expansion module or HCU control module due to live insertion or removal.
- Do not drop or impact the communication expansion module to avoid damage.
- Do not disassemble the communication expansion module; otherwise, the module may be damaged.
- Do not tighten terminals with excessive torque to avoid damage.

4.2 Installation Tools

You may need the following tools during installation:

- Phillips screwdriver, size 1#
- Straight screwdriver, size 1#

4.3 Screw and Fastener Tightening Torque

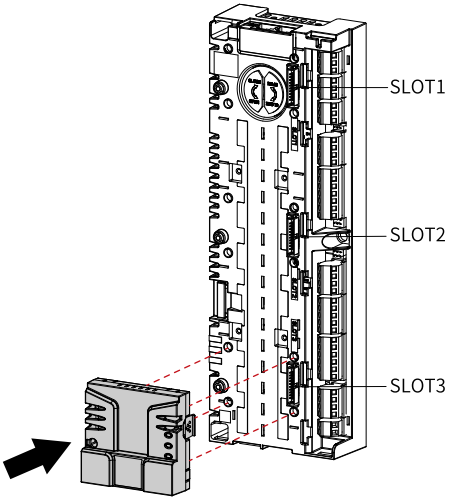
The screws mentioned in this guide must be tightened to the following torques.

Screw	Tightening Torque
M3	0.55 N·m
M4	1.2 N·m

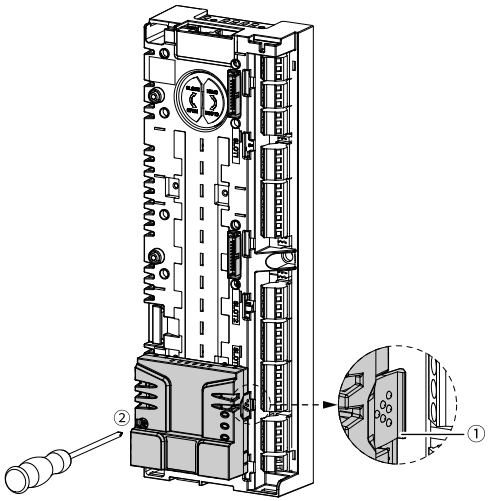
4.4 Installation Method

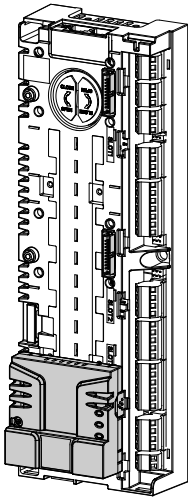
The communication expansion module can be connected to any expansion slots of the HCU control module (installation in SLOT3 is recommended). Select a slot based on the actual on-site conditions. The following example illustrates installation in SLOT3.

1. Place the module onto SLOT3 of the HCU control module, using the four guide posts for alignment and positioning.



2. Press the snap-fit joint (①) of the module inward to insert it into the slot. Tighten the grounding screw (②) of the module with a Phillips screwdriver.





4.5 Removal Method

1. Disconnect the HCU power supply and properly remove all cables connected to the module.
2. Remove the grounding screw of the communication expansion module with a 1# screwdriver.
3. Press the snap-fit joint on the right side of the communication expansion module inward and pull it outward.

5 PROFINET IO Communication

5.1 Overview

The HPFN-10 PROFINET IO industrial Ethernet adapter module complies with the internationally recognized PROFINET IO industrial Ethernet standard. This module can be used on MD880 series AC multidrive to turn the drive into a PROFINET IO slave controlled by the PROFINET IO master. This improves the communication efficiency of the system and enriches the networking functions of the MD880 series drives.

Features:

- Up to 100 Mbps bus communication rate and short communication cycle.
- Supports almost all types of topologies including bus, tree, and star topologies.
- Adopts the distributed clock, with data exchange implemented through a pure hardware mechanism, improving the timing accuracy of data exchange.
- Directly installed in the expansion slot of the HCU control module, requiring no external power supply.

5.2 Protocol Introduction

For the MD880 series, you can select different PZD lengths for data transmission as needed. The supported transmission format types depend on the GSD file, and may vary across different versions of the installed GSD file. You can set the functions corresponding to each PZD during configuration. The following table lists the functions supported by each data format.

Table 5–1 PZD data formats

Data Type	Data Length	Supported Function
Standard telegram 1	PZD-2/2	Cyclic writing of 2 function parameters Cyclic reading of 2 function parameters
Standard telegram 2	PZD-4/4	Cyclic writing of 4 function parameters Cyclic reading of 4 function parameters
Standard telegram 3	PZD-6/6	Cyclic writing of 6 function parameters Cyclic reading of 6 function parameters
Standard telegram 4	PZD-8/8	Cyclic writing of 8 function parameters Cyclic reading of 8 function parameters
Standard telegram 5	PZD-10/10	Cyclic writing of 10 function parameters Cyclic reading of 10 function parameters

Data Type	Data Length	Supported Function
Standard telegram 6	PZD-12/12	Cyclic writing of 12 function parameters Cyclic reading of 12 function parameters
Standard telegram 7	PZD-16/16	Cyclic writing of 16 function parameters Cyclic reading of 16 function parameters
Supplementary telegram	PZD-2/6	Cyclic writing of 2 function parameters Cyclic reading of 6 function parameters

5.3 Hardware Configuration

5.3.1 Module Introduction

Appearance



Figure 5-1 Appearance of the HPFN-10 PROFINET IO industrial Ethernet adapter module

Nameplate and model

HPFN-10
① ②

① **Product code 1** HPFN: Industrial Ethernet adapter module	② **Product code 2** 10: PROFINET IO industrial Ethernet adapter module

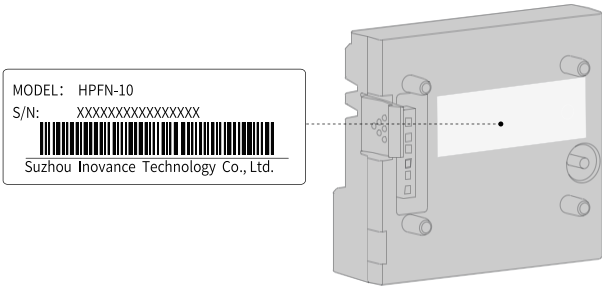


Figure 5-2 Nameplate

Components

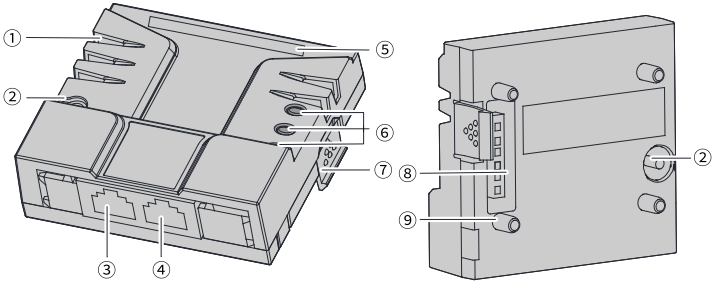


Figure 5-3 Product diagram

Table 5–2 Interfaces

No.	Name	Description
①	Heat dissipation hole	Dissipates heat
②	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
③	PROFINET IO communication terminal X1	Communication terminal 1
④	PROFINET IO communication terminal X2	Communication terminal 2
⑤	MAC address label	Indicates local MAC address
⑥	Status indicator	Indicates module and bus running status
⑦	Snap-fit joint	Fastens the module
⑧	Slot terminal	Implements electrical connection to the HCU control module
⑨	Guide post	Aligns the HPFN-10 module with the HCU control module for easy installation

Outline dimensions

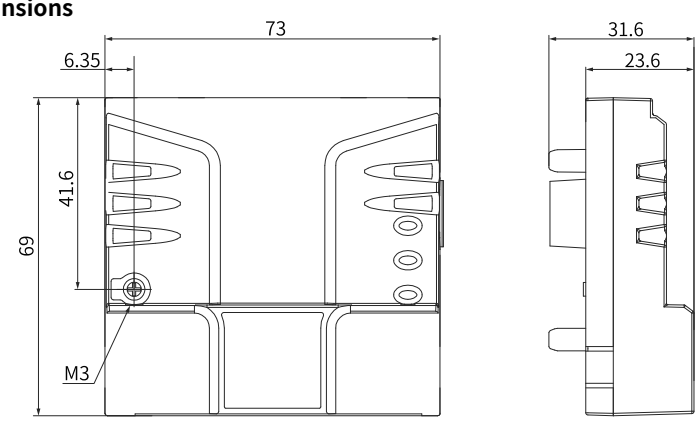


Figure 5-4 Outline dimensions (unit: mm)

5.3.2 Networking Topology

The HPFN-10 module uses a standard Ethernet RJ45 socket to connect to the PROFINET master. Its pin signal assignment is consistent with standard Ethernet pins, and both crossover and straight-through cable routing methods can be used.

Electrical connection for chain networking

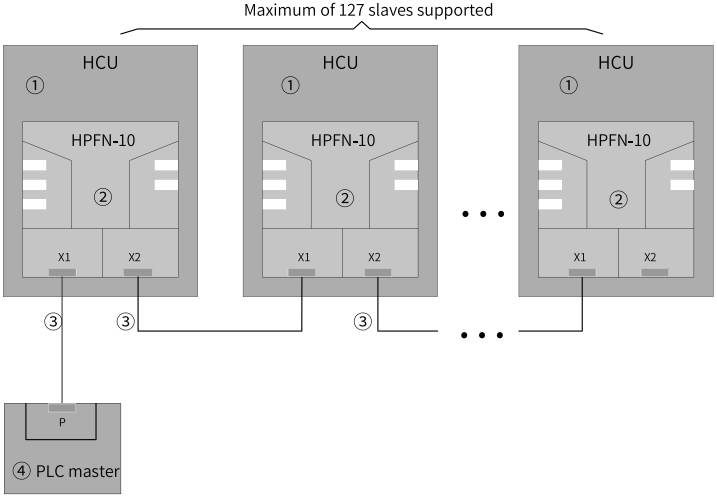


Figure 5-5 Diagram of electrical connection for chain networking

Table 5–3 Electrical connection for chain networking

No.	Name
①	HCU control module
②	HPFN-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	PLC master

Electrical connection for star networking

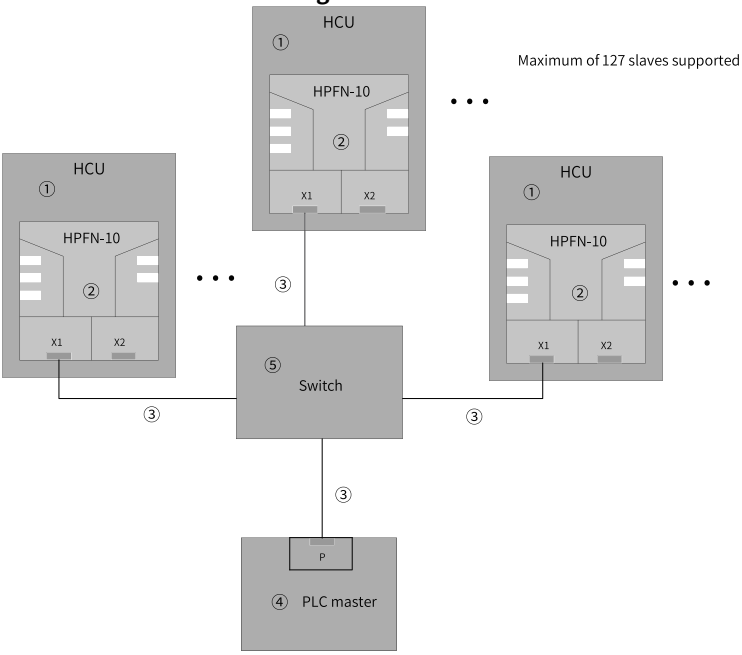


Figure 5-6 Diagram of electrical connection for star networking

Table 5–4 Electrical connection for star networking

No.	Name
①	HCU control module
②	HPFN-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	PLC master
⑤	Switch

5.4 Communication Parameters

Table 5–5 HPFN-10 module parameters

Param.	Definition	Default	Setpoint	Description
n1-00	FBA-A matching bus type	0	8: PROFINET IO module	Use either of these parameter to select the communication module type. Do not set both adapters to the same communication module to avoid communication errors.
n2-00	FBA-B matching bus type	0	8: PROFINET IO module	
n17-00	SLOT selection	0	0: Disable 1: Expansion slot 1_1 2: Expansion slot 1_2 3: Expansion slot 1_3 4: Expansion slot 2_1 5: Expansion slot 2_2 6: Expansion slot 2_3 7: Expansion slot 3_1	Select an expansion slot as needed.
n17-01	Module online/offline state	0	0: Offline 1: Online	-

5.5 Configuration Procedure

Follow these steps to configure the HPFN-10 module:

1. Set the HPFN-10 module parameters.
2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the PROFINET IO module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to PROFINET IO module.
4. Set the network configuration for the master and slave stations.
5. After parameter configuration, power on the HCU again. If all the three status indicators on the HPFN-10 module are solid green, it indicates normal communication between the master and slave stations.

6 PROFIBUS DP Communication

6.1 Overview

The HDP-10 is a PROFIBUS DP fieldbus adapter module that must be used with the HCU control module from Inovance. It is installed and fixed in a slot of the HCU or HESD-10 function expansion module. It communicates with the bus master through the PROFIBUS DP communication protocol.

Features:

- Automatically defects the bus baud rate, supporting a communication rate range of 9.6 kbps to 12 Mbps.
- In a bus topology, supports up to 32 nodes (including the master) without a repeater; with a repeater, supports up to 122 nodes (31 nodes per segment plus 1 repeater).
- Complies with the EMC standard EN 61800-3:2004.
- Supports cyclic and acyclic data exchange methods.
- Powered by the HCU or HESD-10 module, requiring no external power supply.

6.2 Protocol Introduction

6.2.1 Data Transmission Format

In the PROFIBUS DP bus communication protocol, the master station transmits data to the slave station in PPO format. PPO format types include PPO1, PPO2, PPO3, PPO4, PPO5, PPO6, PPO7, and PPO8. The HDP-10 bus communication adapter supports all these data format types.

The following table lists the functions supported by each PPO format type.

Table 6–1 MD880 data transfer format types and functions they support

Data Type	Supported Function
PPO1	Single function parameter operation Cyclic writing of 2 function parameters Cyclic reading of 2 function parameters
PPO2	Single function parameter operation Cyclic writing of 6 function parameters Cyclic reading of 6 function parameters
PPO3	Cyclic writing of 2 function parameters Cyclic reading of 2 function parameters
PPO4	Cyclic writing of 6 function parameters Cyclic reading of 6 function parameters

Data Type	Supported Function
PPO5	Single function parameter operation Cyclic writing of 10 function parameters Cyclic reading of 10 function parameters
PPO6	Cyclic writing of 10 function parameters Cyclic reading of 10 function parameters
PPO7	Single function parameter operation Cyclic writing of 12 function parameters Cyclic reading of 12 function parameters
PPO8	Cyclic writing of 12 function parameters Cyclic reading of 12 function parameters

6.2.2 PKW Data

Data blocks of the PPO data format are divided into two areas, PKW area (parameter area) and PZD area (process data area). The following figure shows the structure of each PPO format type supported by the MD880 AC drive.

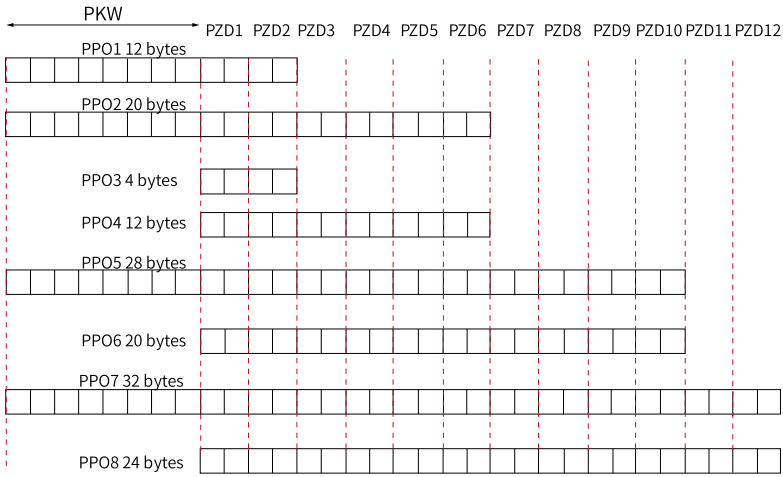


Figure 6-1 Supported PPO format types

The PKW area enables the master to read or write a single AC drive parameter. The communication address of the AC drive parameter is directly specified in the communication data. The PKW area consists of three parts: PKE, IND, and PWE. The PKE, IND, and PWE parts always contain two bytes, two bytes, and four bytes, respectively, as shown in the following table.

Table 6–2 PKW data format

PKW Sent by the Master Station							
Operation command	Address		Reserved			Write operation: Parameter value Read operation: Null	
PKE	PKE	IND	IND	PWE	PWE	PWE	PWE
PKW Returned by the AC Drive							
Operation state	Address		Reserved			Success: Parameter value Failure: Error code	
PKE	PKE	IND	IND	PWE	PWE	PWE	PWE

Table 6–3

PKW Sent by the Master Station		PKW Returned by the AC Drive	
PKE	High 4 bits: Command code 0: No request 1: Read parameter data 2: Modify parameter data (The preceding command code is in decimal format.) Low 4 bits: Reserved Low 8 bits: High bits of parameter address	PKE	High 4 bits: Response code 0: No request 1: Parameter operated correctly 7: Execution failed Low 8 bits: High bits of parameter address
IND	High 8 bits: Low bits of parameter address Low 8 bits: Reserved	IND	High 8 bits: Low bits of parameter address Low 8 bits: Reserved
PWE	High 16 bits: Reserved Low 16 bits: Not used in the read request; indicates the parameter value in the write request	PWE	Request successful: Parameter value Request failed: error code (consistent with standard Modbus) 1: Invalid command 2: Invalid address 3: Invalid data 4: Other errors

6.3 Hardware Configuration

6.3.1 Module Introduction

Appearance

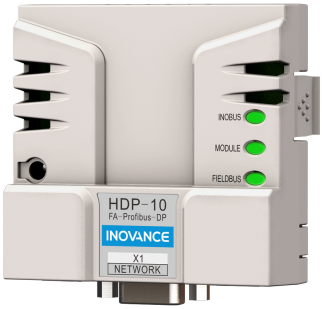


Figure 6-2 Appearance of the HDP-10 PROFIBUS DP fieldbus adapter module

Nameplate and model

HDP-10
① ②

① **Product code 1** HDP: Bus adapter module	② **Product code 2** 10: PROFIBUS DP bus communication module

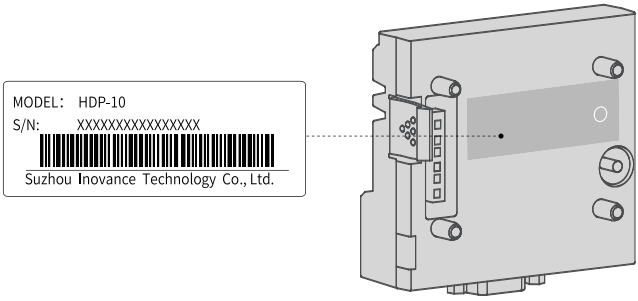


Figure 6-3 Nameplate

Components

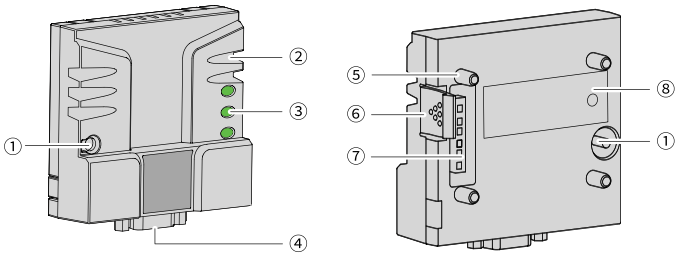


Figure 6-4 Product diagram

Table 6–4 Interfaces

No.	Name	Description
①	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
②	Heat dissipation hole	Dissipates heat
③	Communication status indicator	Indicates module communication status
④	Bus communication terminal	PROFIBUS DP bus communication terminal
⑤	Guide post	Aligns the HDP-10 module with the HCU control module for easy installation
⑥	Snap-fit joint	Fastens the module
⑦	Slot terminal	Implements electrical connection to the HCU
⑧	Nameplate	Includes the module model number and serial number

Outline dimensions

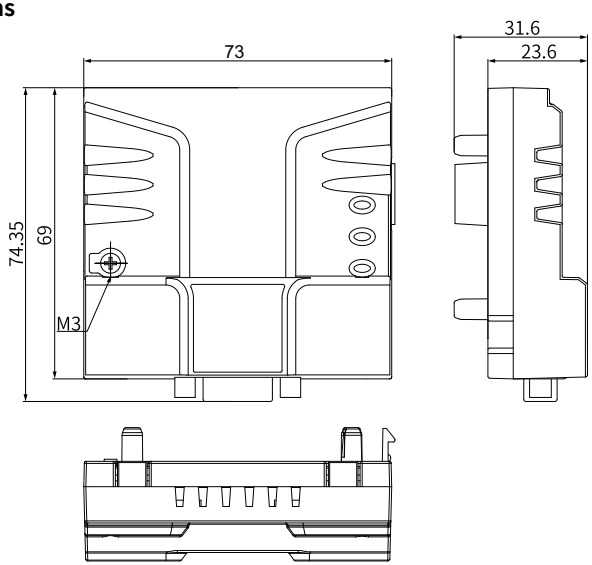


Figure 6-5 Outline dimensions (unit: mm)

6.3.2 Communication Port

The HDP-10 communication module uses a standard DB9 socket to connect to the PROFIBUS master station. Its pin assignment follows Siemens DB9 socket standard, as shown in the following figure.

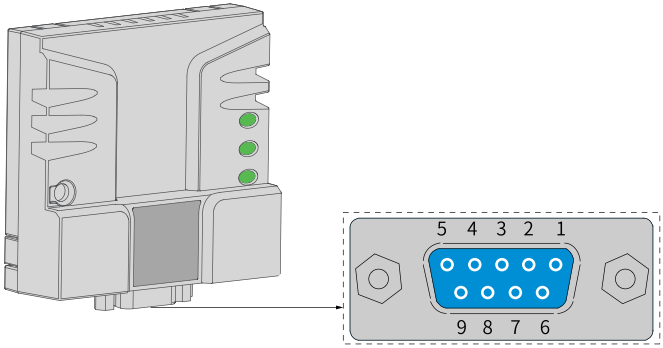


Figure 6-6 HDP-10 module terminal position and pin assignment

Table 6–5 HDP-10 module communication terminal pin assignment

Pin	Name	Function
1	NC	Unconnected internally
2	NC	Unconnected internally
3	Data cable B	Positive terminal of data cable
4	NC	Unconnected internally
5	GND	Ground terminal of isolated 5 V power supply
6	5V	Isolated 5 V power supply
7	NC	Unconnected internally
8	Data cable A	Negative terminal of data cable
9	NC	Unconnected internally

6.3.3 Networking Topology

Bus cable

It is recommended to use a dedicated PROFIBUS DP bus cable with the following specifications.

Table 6–6 Cable specifications

Item	Specification
Conductor	One pair (2 x 22 AWG) of single-strand copper conductors
Insulating sheath color	Green/Red
Shield	Aluminum-plastic tape and tinned copper braid
Sheath material	PVC
Ambient temperature	–30°C to +70°C
Appearance	Purple

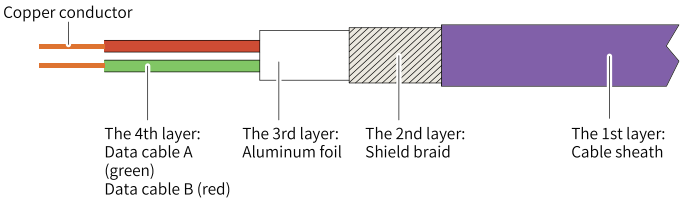


Figure 6-7 Internal layers of the bus cable

Bus terminal

It is recommended to use Siemens dedicated PROFIBUS DP connectors (model: 6ES7 972-0BB12-0XA0).

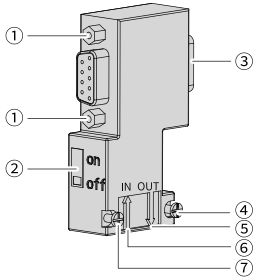


Figure 6-8 Bus terminal structure

No.	Name
①	Terminal fastening screw
②	DIP switch for termination resistor
③	Terminal connecting to HDP-10
④	Connector cover fastening screw
⑤	Bus output jack
⑥	Bus input jack
⑦	Connector cover fastening screw

The internal termination resistor connection schematic for the bus terminal is shown in the following figure.

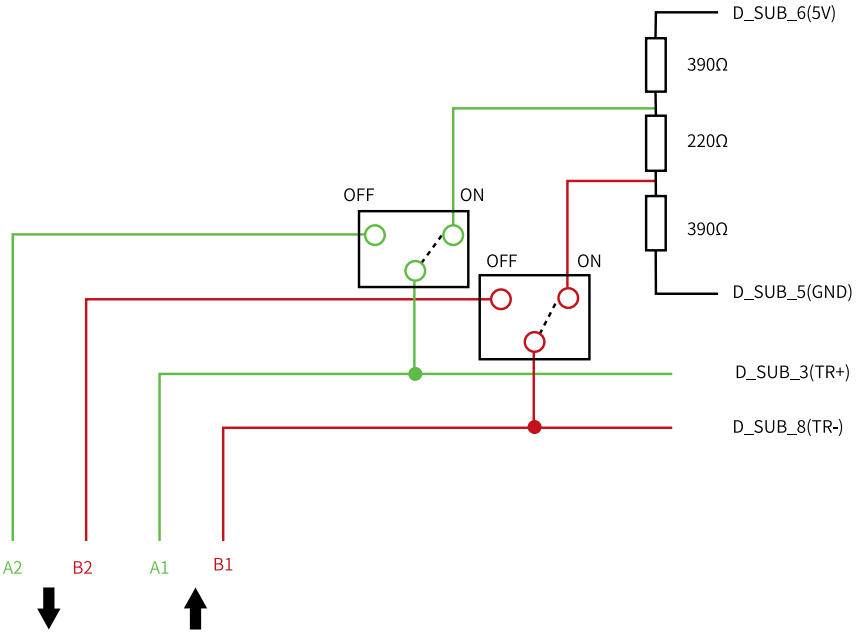


Figure 6-9 Connector termination resistor schematic

Bus cable connection procedure

1. Strip the cable sheath and reverse the length as specified in the following figure (unit: mm).

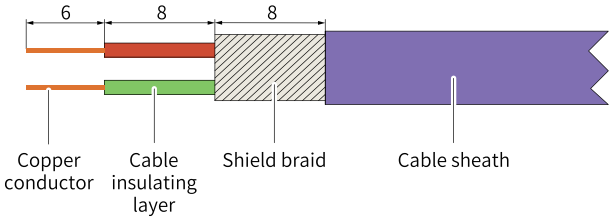


Figure 6-10 Cable stripping dimensions

2. Remove the bus connector terminal cover using a 1# straight screwdriver.

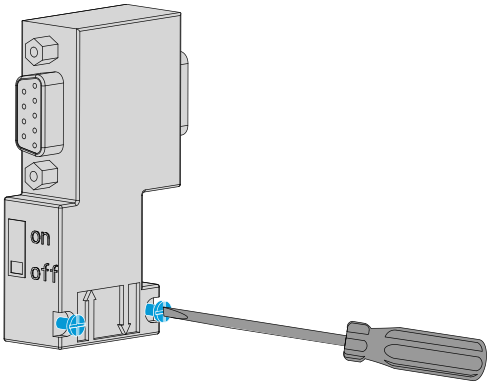


Figure 6-11 Removing the connector cover

3. Secure the cable to the connector installation position using the 1# straight screwdriver. Ensure that the shield is in close contact with the shield metal sheet. Connect green cables to the A terminals and red cables to the B terminals, as shown in the following figure.

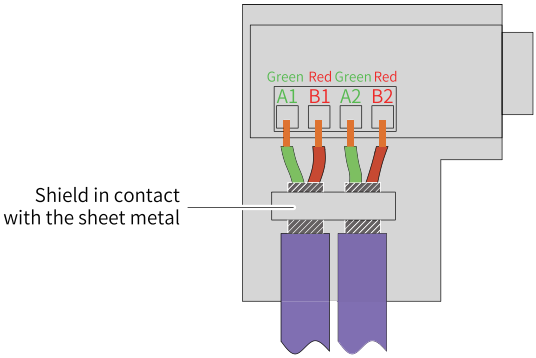


Figure 6-12 Connecting cables to the connector

4. Put back the connector cover and fasten the screws on the cover. Insert the connector in the X1 port of the HDP-10 module and tighten the fixing screws on both sides with a screwdriver.

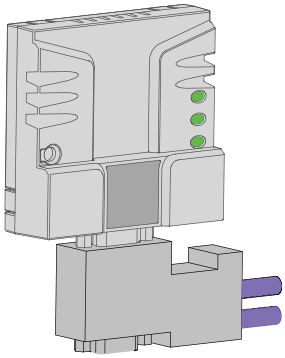


Figure 6-13 Connecting the connector to the HDP-10

Bus topology

PROFIBUS network topology

- PROFIBUS supports a device address range of 0 to 127. Some addresses are reserved for special purposes: address 127 is used for global control or broadcast messages, address 126 is reserved for devices not yet assigned an address, and address 0 is reserved for Class-2 master stations.
- Class-1 master station addresses typically start from address 1. For a single-master system, excluding the three reserved addresses, the maximum number of slave stations in the system is 124 (128 – 3 – 1 = 124).
- A PROFIBUS network can have a maximum of 126 devices (including master and slave stations). Within the same network segment, there can be a maximum of 32 devices, including the master. If the number of devices in a network segment exceeds 32 or is limited by distance or device characteristics, multiple network segments must be created. These network segments can be connected using repeaters or segment couplers. The following figure illustrates the topology of a single-master system.

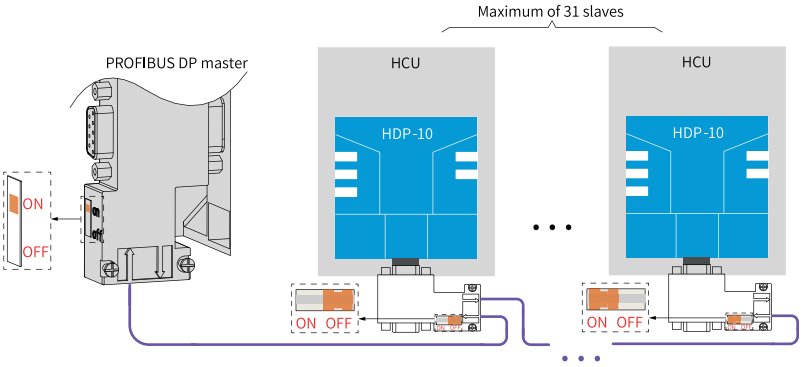


Figure 6-14 Single-master system topology without repeater

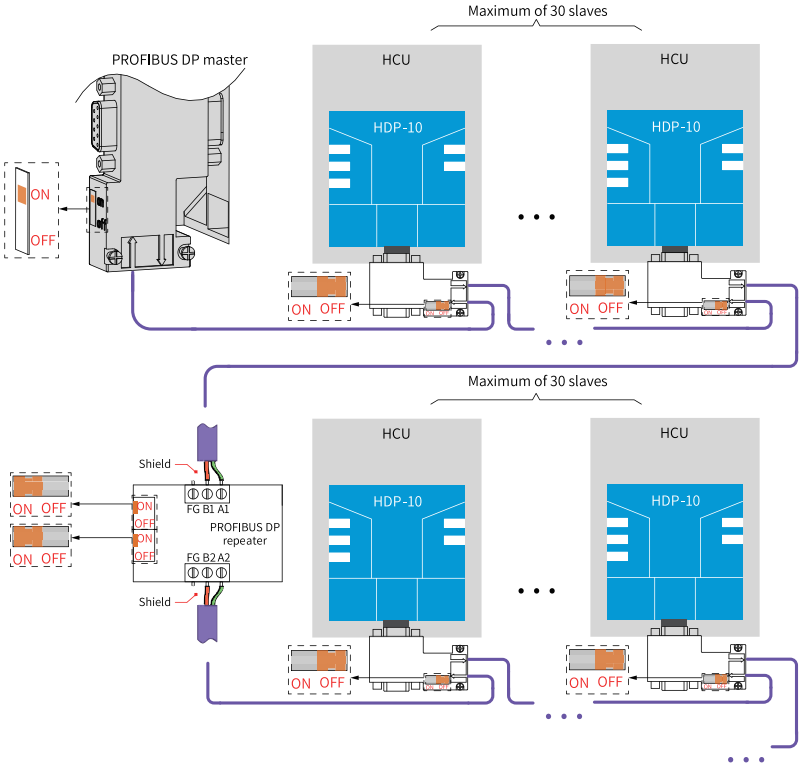


Figure 6-15 Single-master system topology with repeater

Use and description of termination resistors

- When signals are transmitted over a cable, any additional resistance, capacitance, or line termination can cause reflections, similar to echoes. These reflections can cause signal interference by overlapping multiple signals on the line, which is particularly severe during high-speed transmission.
- To minimize reflections, an appropriate termination resistor network should be added at the end of the circuit. This termination resistor network absorbs signal energy, reducing signal reflections (theoretically eliminating them). Both ends of a PROFIBUS DP network require termination resistors, which can be connected as shown in the PROFIBUS cable connector diagram.

Note

Termination resistors should only be connected at the two ends of the network and not at other points, as this will affect network operation. A Siemens connector (model: 6ES7 972-0BB12-0XA0) has a built-in termination resistor. To enable the termination resistor, set the DIP switch to ON.

Transmission distance

Depending on the baud rate set by the master station, the length of the communication cable between the HDP-10 communication module and the PROFIBUS master station must comply with Siemens DB9 wiring standards. The relationship between the baud rate and the cable length is shown in the following table.

Table 6–7 Communication rate and cable length

Transmission Rate (kbps)	Max. Length of Cable Type A (m)	Max. Length of Cable Type B (m)
9.6	1200	1200
19.2	1200	1200
187.5	600	600
500	200	200
1500	100	100
3000	100	Not recommended
6000	100	Not recommended
12000	100	Not recommended

Table 6–8 Technical specifications of cables

Cable Parameter	Type A	Type B
Impedance	135 Ω to 165 Ω (f = 3 MHz to 20 MHz)	100 Ω to 130 Ω (f > 100 kHz)
Capacitance	< 30 pF/m	< 60 pF/m

Cable Parameter	Type A	Type B
Resistance	< 110 Ω/km	Not specified
Cross-sectional area of conductor	⩾ 0.34 mm^2	⩾ 0.22 mm^2

Note

- The data in the table is for reference only and may vary due to cable impedance, capacitance, resistance, and cross-sectional area.
- Cable type A refers to shielded twisted pair cables, whereas cable type B refers to regular twisted pair cables.

6.4 Communication Parameters

Table 6–9 HDP-10 module parameters

Param.	Definition	Default	Setpoint	Description
n1-00	FBA-A matching bus type	0	7: PROFIBUS DP module	Use either of these parameter to select the communication module type. Do not set both adapters to the same communication module to avoid communication errors.
n2-00	FBA-B matching bus type	0	7: PROFIBUS DP module	
n1-91	FBA-A device diagnosis information upload disable	1	0: Enable 1: Disable	This parameter specifies whether to upload fault to the master station when an HCU fault occurs.

Param.	Definition	Default	Setpoint	Description
n16-00	SLOT selection	0	0: Disable 1: Expansion slot 1_1 2: Expansion slot 1_2 3: Expansion slot 1_3 4: Expansion slot 2_1 5: Expansion slot 2_2 6: Expansion slot 2_3 7: Expansion slot 3_1	Select an expansion slot as needed.
n16-01	Module online/offline state	0	0: Offline 1: Online	-
n16-02	Expansion card ID	1	1 to 125	Slave ID
n16-03	Reserved	0	-	-
n16-04	Reserved	0	-	-
n16-05	Reserved	0	-	-

Param.	Definition	Default	Setpoint	Description
n16-09	Signature code	0	0: Default 1: 0806 2: 0956 3: 8045 4: 0812 5: 0815 6: 6012 7: 80b5 8: 4149 9: 0572 10: 4150 11: 0612+JH 12: 0812+SL 13: 0959v2.145 14: 0959v2.161 15: Reserved 16: Reserved	This parameter is used to adapt to different AC drive models. When it is set to 0, the code automatically matches the AC drive model.
n16-10	DP data check bit	0	0: PZD1.bit10 (default) 1: Disable 2: PZD1.bit0 3: PZD1.bit1 4: PZD1.bit2 5: PZD1.bit3 6: PZD1.bit4 7: PZD1.bit5 8: PZD1.bit6 9: PZD1.bit7 10: PZD1.bit8 11: PZD1.bit9 12: PZD1.bit10 13: PZD1.bit11 14: PZD1.bit12 15: PZD1.bit13 16: PZD1.bit14 17: PZD1.bit15	This parameter specifies the data check bit. If data check is enabled (for example, this parameter is set to "0: PZD1.bit10 (default)"), the master station must periodically write 1 and 0 to bit10 of PZD1.

6.5 Configuration Procedure

Follow these steps to configure the HDP-10 module:

1. Set the HDP-10 module parameters.
2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the PROFIBUS DP module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to PROFIBUS DP module.
4. Set the network configuration for the master and slave stations.

5. After parameter configuration, power on the HCU again. If all the three status indicators on the HDP-10 module are solid green, it indicates normal communication between the master and slave stations.

7 Modbus RTU Communication

7.1 Overview

The HMBA-10 is a Modbus RTU fieldbus adapter module that must be used with the HCU control module from Inovance. It is installed and fixed in a slot of the HCU or HESD-10 function expansion module. It enables networking and remote control of multi-drive systems through the Modbus RTU communication network.

7.2 Use of PZD1

When the equipment is started through PZD1 and A9-00 is set to 0, set b0-00 (start/stop control word source of channel 1) to 1 or 2. When A9-00 is set to 1, set b2-00 (start/stop control word source of channel 2) to 1 or 2. In this case, fieldbus process data PZD1 is used as the command source for commands OFF1 and OFF2, as shown in the following table.

Table 7–1 Command source table when starting through PZD1

Command	Command Source
OFF1	PZD1.0
OFF2	PZD1.1
OFF3	PZD1.2
Run enable source	PZD1.3
Fault reset source	PZD1.7
Jog 1 source	PZD1.8
Jog 2 source	PZD1.9
Remote control request must be 1	PZD1.10
Speed inversion source	PZD1.11

When starting the equipment through fieldbus preset value configuration, set the speed reference separately through multi-reference, analog, or communication.

Note

Starting the equipment through PZD1 is only applicable to the HCU-50/51 and HCU-50S/51S series.

7.3 Protocol Introduction

7.3.1 Modbus RTU Communication Introduction

The Modbus serial link protocol is a master-slave protocol where only one master and one or more slaves can be present on the bus at the same time. Modbus communication is always initiated by the master. A slave does not send data unless it receives a request from the master, and slaves cannot communicate with each other. Modbus is compatible with RS232, RS485, and RS422 interfaces, with RS485 being the primary interface used in practical applications. The Modbus protocol includes two modes: RTU and ASCII. The HMBA-10 module only supports RTU.

Slave Address	Param.	Data	CRC
1 byte	1 byte	Maximum 255 bytes	2 bytes

7.3.2 Modbus-Specific Parameters

Control through continuous addresses

The Modbus RTU and Modbus TCP support writing of process data n1-68 to n1-83 (n2-68 to n2-83). After converted by communication base values of n1-36 to n1-51 (n2-36 to n2-51), the process data is updated to the corresponding connector U5-44 to U5-59 (U5-60 to U5-75).

Fieldbus adapter A:

Address	Write Parameter	Name	Description
33807	n1-68	[Process data input 1] Data display	Written through Modbus, converted by the base value of n1-36, and updated to connector U5-44
33808	n1-69	[Process data input 2] Data display	Written through Modbus, converted by the base value of n1-37, and updated to connector U5-45
33809	n1-70	[Process data input 3] Data display	Written through Modbus, converted by the base value of n1-38, and updated to connector U5-46
33810	n1-71	[Process data input 4] Data display	Written through Modbus, converted by the base value of n1-39, and updated to connector U5-47

Address	Write Parameter	Name	Description
33811	n1-72	[Process data input 5] Data display	Written through Modbus, converted by the base value of n1-40, and updated to connector U5-48
33812	n1-73	[Process data input 6] Data display	Written through Modbus, converted by the base value of n1-41, and updated to connector U5-49
33813	n1-74	[Process data input 7] Data display	Written through Modbus, converted by the base value of n1-42, and updated to connector U5-50
33814	n1-75	[Process data input 8] Data display	Written through Modbus, converted by the base value of n1-43, and updated to connector U5-51
33815	n1-76	[Process data input 9] Data display	Written through Modbus, converted by the base value of n1-44, and updated to connector U5-52
33816	n1-77	[Process data input 10] Data display	Written through Modbus, converted by the base value of n1-45, and updated to connector U5-53
33817	n1-78	[Process data input 11] Data display	Written through Modbus, converted by the base value of n1-46, and updated to connector U5-54
33818	n1-79	[Process data input 12] Data display	Written through Modbus, converted by the base value of n1-47, and updated to connector U5-55
33819	n1-80	[Process data input 13] Data display	Written through Modbus, converted by the base value of n1-48, and updated to connector U5-56
33820	n1-81	[Process data input 14] Data display	Written through Modbus, converted by the base value of n1-49, and updated to connector U5-57

Address	Write Parameter	Name	Description
33821	n1-82	[Process data input 15] Data display	Written through Modbus, converted by the base value of n1-50, and updated to connector U5-58
33822	n1-83	[Process data input 16] Data display	Written through Modbus, converted by the base value of n1-51, and updated to connector U5-59

Fieldbus adapter B

Address	Write Parameter	Name	Description
33839	n2-68	[Process data input 1] Data display	Written through Modbus, converted by the base value of n2-36, and updated to connector U5-60
33840	n2-69	[Process data input 2] Data display	Written through Modbus, converted by the base value of n2-37, and updated to connector U5-61
33841	n2-70	[Process data input 3] Data display	Written through Modbus, converted by the base value of n2-38, and updated to connector U5-62
33842	n2-71	[Process data input 4] Data display	Written through Modbus, converted by the base value of n2-39, and updated to connector U5-63
33843	n2-72	[Process data input 5] Data display	Written through Modbus, converted by the base value of n2-40, and updated to connector U5-64
33844	n2-73	[Process data input 6] Data display	Written through Modbus, converted by the base value of n2-41, and updated to connector U5-65
33845	n2-74	[Process data input 7] Data display	Written through Modbus, converted by the base value of n2-42, and updated to connector U5-66
33846	n2-75	[Process data input 8] Data display	Written through Modbus, converted by the base value of n2-43, and updated to connector U5-67

Address	Write Parameter	Name	Description
33847	n2-76	[Process data input 9] Data display	Written through Modbus, converted by the base value of n2-44, and updated to connector U5-68
33848	n2-77	[Process data input 10] Data display	Written through Modbus, converted by the base value of n2-45, and updated to connector U5-69
33849	n2-78	[Process data input 11] Data display	Written through Modbus, converted by the base value of n2-46, and updated to connector U5-70
33850	n2-79	[Process data input 12] Data display	Written through Modbus, converted by the base value of n2-47, and updated to connector U5-71
33851	n2-80	[Process data input 13] Data display	Written through Modbus, converted by the base value of n2-48, and updated to connector U5-72
33852	n2-81	[Process data input 14] Data display	Written through Modbus, converted by the base value of n2-49, and updated to connector U5-73
33853	n2-82	[Process data input 15] Data display	Written through Modbus, converted by the base value of n2-50, and updated to connector U5-74
33854	n2-83	[Process data input 16] Data display	Written through Modbus, converted by the base value of n2-51, and updated to connector U5-75

For more information on using continuous addresses as the start, stop, and reset command sources of the drive through the Modbus, see the "Use of PZD1" section in the *MD880 Series AC Drive Commissioning Guide (HCU-50S, HCU-50S-CS, HCU-51S, and HCU-51S-CS)*.

Control through special addresses

To start/stop the MD880 drive through Modbus communication, the HCU-50S/51S provides several special communication addresses for accessing Modbus, and updates the written data to a specific connector for reading the source parameters. As specified by the Modbus communication protocol, parameters are read and written directly through the communication address and no process data will be produced. However, b0-01/b2-01 of the OFF1 source is not allowed to be set to 1 to start the drive and a connector data must be selected.

Note

Addresses of the following connectors are read-only in the software tool. Write operations can only be performed using the addresses specified in the following table.

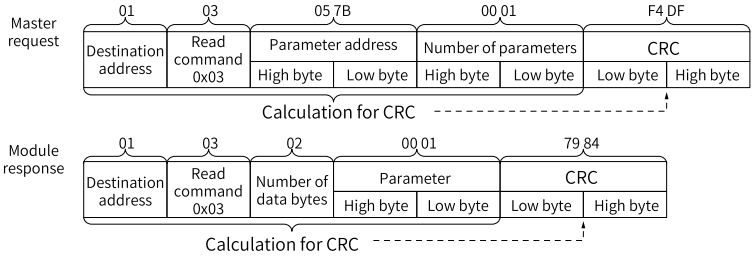
Address	Written to Connector	Name	Description
34196<1>	U1-75	Start/Stop command (Modbus)	Values: 0 and 1. Recommended for user-defined OFF1 command.
34230	U5-40	Modbus communication setting value 1	Value range: –32768 to +32767
34231	U5-41	Modbus communication setting value 2	
34232	U9-50	Modbus communication double-word setpoint	Value range: -2^{31} to $2^{31} - 1$. Suitable for high-precision speed setting. The high 16 bits of the data are written to address 34232, whereas the low 16 bits are written to address 34233.

Note

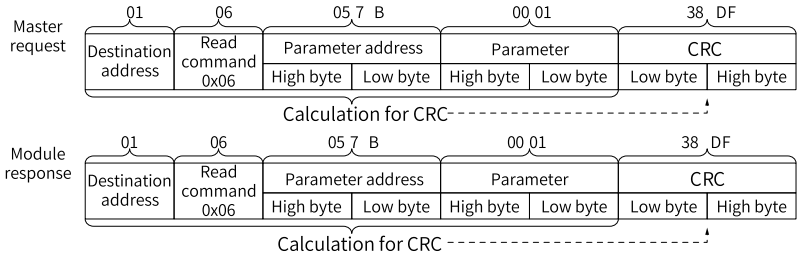
<1>: This address is only applicable to the HCU-50/51 and HCU-50S/51S series.

7.3.3 Modbus RTU Communication Examples

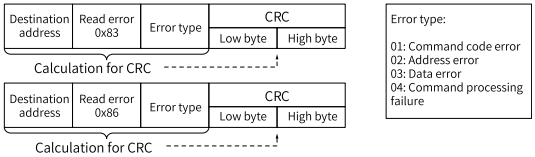
- Frame format for using the 0x03 command to read address 057BH



- Frame format for using the 0x06 command to write 1 to address 057BH



- If the slave detects a communication frame error, or the read or write operation fails due to other reasons, the slave responds with an error frame, as shown below.



7.4 Hardware Configuration

7.4.1 Module Introduction

Appearance

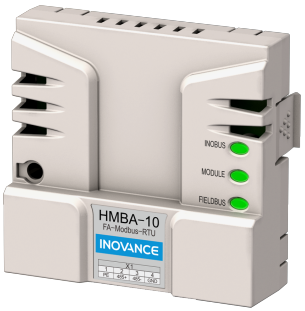


Figure 7-1 Appearance of the HMBA-10 Modbus RTU fieldbus adapter module

Nameplate and model

HMBA-10
① ②

① **Product code 1** HMBA: Bus adapter module	② **Product code 2** 10: Modbus RTU bus adapter module

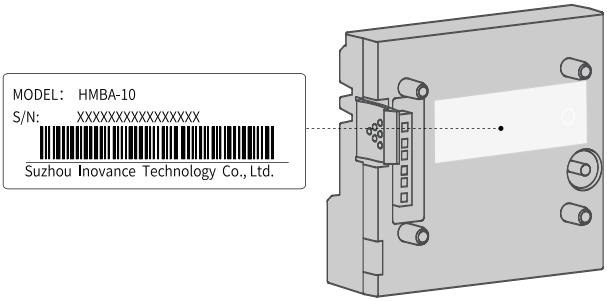


Figure 7-2 Nameplate

Components

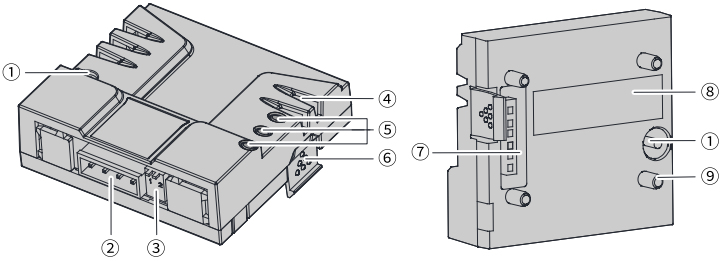


Figure 7-3 Product diagram

Table 7–2 Interfaces

No.	Name	Description
①	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
②	Bus communication terminal	Modbus RTU bus communication terminal
③	Resistor DIP switch	Termination resistor
④	Heat dissipation hole	Dissipates heat
⑤	Status indicator	Indicates module running status
⑥	Snap-fit joint	Fastens the module
⑦	Slot terminal	Implements electrical connection to the HCU control module
⑧	Nameplate	Includes the module model number and serial number
⑨	Guide post	Aligns the HMBA-10 module with the HCU control module for easy installation

Outline dimensions

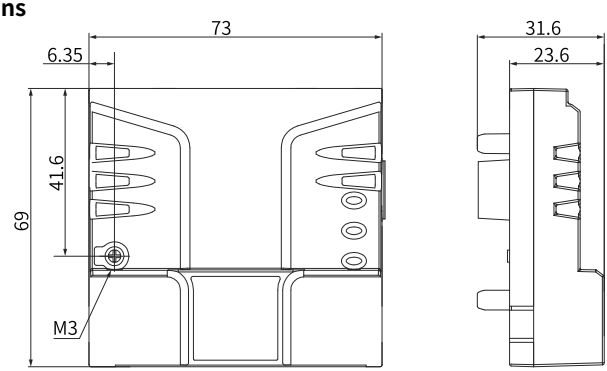


Figure 7-4 Outline dimensions (unit: mm)

7.4.2 Communication Port

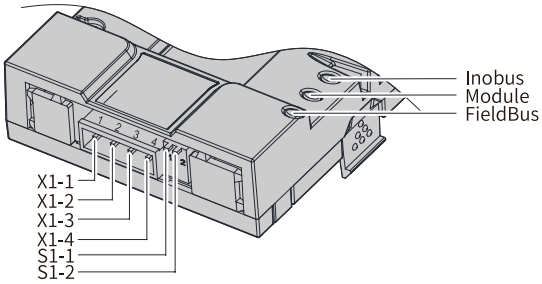


Figure 7-5 Terminal assignment

Table 7–3

Terminal Pin	Name	Wiring	Description
X1-1	SHIELD	Connect this pin to the shield.	Recommended cable: 4-conductor shielded twisted pair cable, with a cross-sectional area of 0.3 mm^2 to 2.0 mm^2
X1-2	485+	Twisted pair	
X1-3	485–		
X1-4	GND	Connect this pin.	
S1-1	DIP switch 1	-	The termination resistor is connected when S1-1 and S1-2 both are set to ON.
S1-2	DIP switch 2	-	

Terminal wiring:

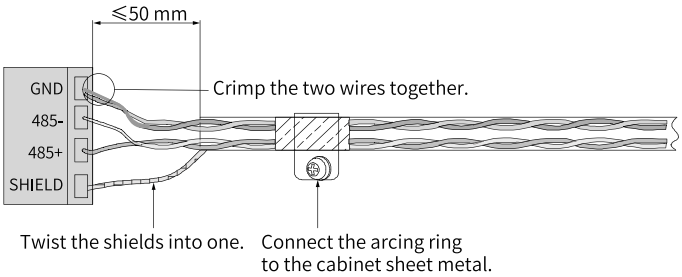


Figure 7-6 Cable

7.4.3 Networking Topology

The Modbus RTU bus supports a maximum of 31 nodes (excluding the master). The termination resistors of the master and the last node must be set to ON. The following figure shows the networking. If the number of nodes for networking exceeds 31, use a repeater.

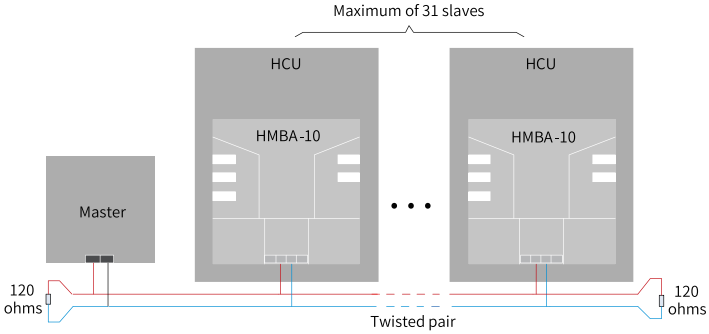


Figure 7-7 Networking diagram

7.5 Communication Parameters

Table 7–4 HDP-10 module parameters

Param.	Definition	Default	Setpoint	Description
n1-00	FBA-A matching bus type	0	1: Modbus RTU module	Use either of these parameter to select the communication module type. Do not set both adapters to the same communication module to avoid communication errors.
n2-00	FBA-B matching bus type	0	1: Modbus RTU module	
n10-00	SLOT selection	0	0: Disable 1: Expansion slot 1_1 2: Expansion slot 1_2 3: Expansion slot 1_3 4: Expansion slot 2_1 5: Expansion slot 2_2 6: Expansion slot 2_3 7: Expansion slot 3_1	Select an expansion slot as needed.
n10-01	Module online/offline state	0	0: Offline 1: Online	-
n10-02	Modbus baud rate	7	0: 1200 bps 1: 2400 bps 2: 4800 bps 3: 9600 bps 4: 19200 bps 5: 38400 bps 6: 57600 bps 7: 115200 bps	Baud rate setting
n10-03	Modbus data format	1	0: Even parity check (8-E-1) 1: Odd parity check (8-O-1) 2: No check (8-N-2) 3: 8-N-1	Data format setting
n10-04	Modbus local address	1	1 to 255	Slave ID

Param.	Definition	Default	Setpoint	Description
n10-05	Modbus response delay	4	0 to 20	This parameter sets the response delay for the HMBA-10 module. When the module receives data, it waits for the delay time before responding.
n10-06	Modbus communication timeout interval	20.0	0 to 60.0	This parameter sets the HMBA-10 module communication time. If the module does not receive any data within the time, a timeout fault will be reported.

7.6 Configuration Procedure

Follow these steps to configure the HMBA-10 module:

1. Set the HMBA-10 module parameters.
2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the Modbus RTU module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to Modbus RTU module.
4. After parameter configuration, power on the HCU again. If all the three status indicators on the HMBA-10 module are solid green, it indicates normal communication between the master and slave stations.

8 CANopen Communication

8.1 Overview

The HCAN-10 is a new-generation fieldbus adapter module from Inovance. It works with the MD880 series HCU control module to enable networking and remote control of multi-drive systems through the CANopen fieldbus communication network.

Features:

- Supports the Node Guard protocol, based on which the master station can read device status.
- Supports the Heartbeat protocol, based on which the slave station periodically reports its status to the master station.
- Supports transmission of PDO data which is 16-bit.
- Supports transmission of SDO data only through the expedited transfer mechanism. Up to four channels of data can be transmitted per frame.
- All command CANIDs follow the default values of the DS301 predefined connection set.
- Directly installed in the expansion slot of the HCU control module, requiring no external power supply.

8.2 Protocol Introduction

8.2.1 CANopen Protocol Introduction

The CANopen protocol was developed from the CAN Application Layer (CAL) in the late 1990s by CAN-in-Automation (CiA), an organization based in Nuremberg, Germany.

After multiple revisions to the CANopen protocol specification, its stability, real-time performance, and interference resistance have been significantly improved. Additionally, CiA has developed industry-specific device profile protocols based on the CANopen foundation protocol (CiA 301), promoting the adoption and development of CANopen across various industries. These device profile protocols redefine the meaning of internal CANopen data or add new control logic tailored to specific industry applications.

8.2.2 CANopen Predefined Connection Set

CANopen defines a mandatory default identifier (CAN-ID) allocation table. The identifiers are available in the pre-operational state and can be modified through dynamic allocation. CANopen devices must provide corresponding identifiers for the communication objects they support. The predefined connection set includes four Receive-PDOs, four Transmit-PDOs, one SDO (occupying two CAN-IDs), one emergency object, and one Node-Error-Control ID. It also supports the NMT-Module-Control service that does not require acknowledgment, as well as broadcast for SYNC and Time Stamp objects.

Table 8–1 CANopen predefined connection set

Broadcast Objects in the CANopen Predefined Master-Slave Connection Set			
Communication Object	Param.	COB-ID	Communication Parameter Index in OD
Network management	0000	000H	-
Synchronization object	0001	080H	1005H, 1006H, and 1007H
Emergency message	0010	100H	1012H and 1013H
Peer Objects in the CANopen Predefined Master-Slave Connection Set			
Communication Object	Param.	COB-ID	Communication Parameter Index in OD
Emergency	0001	081H to 0FFH	1024H and 1015H
Transmit-PDO1	0011	181H to 1FFH	1800H
Receive-RDO1	0100	201H to 27FH	1400H
Transmit-TDO2	0101	281H to 2FFH	1801H
Receive-RDO2	0110	301H to 37FH	1401H
Transmit-TDO3	0111	381H to 3FFH	1802H
Receive-RDO3	1000	401H to 47FH	1402H
Transmit-TDO4	1001	481H to 4FFH	1803H
Receive-RDO4	1010	501H to 57FH	1403H
Transmit-SDO (server)	1011	581H to 5FFH	1200H
Receive-SDO (client)	1100	601H to 67FH	1200H
Network management error control	1110	701H to 77FH	1016H to 1017H

8.2.3 SDO Data

SDO is mainly used by the CANopen master station to read/write parameters of slave nodes. The data format sent by the master station is shown in the following table. For both read and write operations, the parameter address must be incremented by 1. For example, to write to parameter address 194, the actual address to operate on is 194 + 1 = 195 (0x00C3).

Table 8–2 Transmit-SDO for read operation

CAN Frame	CANopen Data	Description
COB-ID	CAN frame header	0x600 + Node-ID
DLC	Length	8

CAN Frame	CANopen Data	Description
DATA0	Read command code	0x40
DATA1	Low-order byte of index	High-order byte of parameter address
DATA2	High-order byte of index	0x20
DATA3	Sub-index	Low-order byte of parameter address
DATA4	Reserved	-
DATA5	Reserved	-
DATA6	Reserved	-
DATA7	Reserved	-

Table 8–3 AC drive response to read operation

CAN Frame	CANopen Data	Description
COB-ID	CAN frame header	0x580 + Node-ID
DLC	Length	8
DATA0	Response command	Success: 0x4B; error: 0x80
DATA1	Low-order byte of index	High-order byte of parameter address
DATA2	High-order byte of index	0x20
DATA3	Sub-index	Low-order byte of parameter address
DATA4	Low-order byte of data	-
DATA5	High-order byte of data	-
DATA6	Error code • 0: Success • Other: Error	-
DATA7		-

Table 8–4 Transmit-SDO for write operation

CAN Frame	CANopen Data	Description
COB-ID	CAN frame header	0x600 + Node-ID
DLC	Length	8
DATA0	Write command code	0x2B
DATA1	Low-order byte of index	High-order byte of parameter address
DATA2	High-order byte of index	0x20

CAN Frame	CANopen Data	Description
DATA3	Sub-index	Low-order byte of parameter address
DATA4	Low-order byte of data	-
DATA5	High-order byte of data	-
DATA6	Reserved	-
DATA7	Reserved	-

Table 8–5 AC drive response to write operation

CAN Frame	CANopen Data	Description
COB-ID	CAN frame header	0x580 + Node-ID
DLC	Length	8
DATA0	Response command	Success: 0x60; error: 0x80
DATA1	Low-order byte of index	High-order byte of parameter address
DATA2	High-order byte of index	0x20
DATA3	Sub-index	Low-order byte of parameter address

8.3 Hardware Configuration

8.3.1 Module Introduction

Appearance

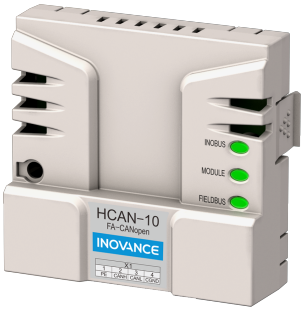


Figure 8-1 Appearance of the HCAN-10

Nameplate and model

HCAN-10
① ②

① **Product code 1** HCAN: CAN bus adapter	② **Product code 2** 10: CANopen bus adapter

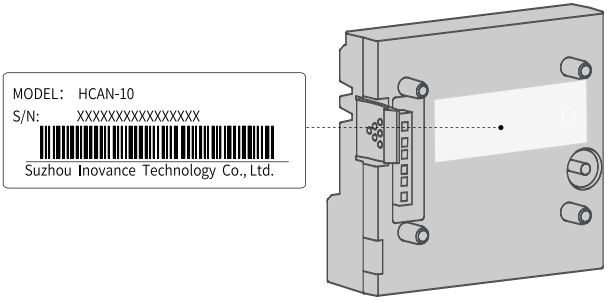


Figure 8-2 Nameplate

Components

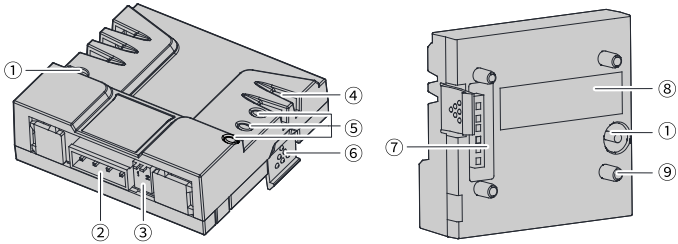


Figure 8-3 HCAN-10 diagram

No.	Name	Description
①	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
②	Bus communication terminal	CANopen bus communication terminal
③	Resistor DIP switch	CAN bus termination resistor
④	Heat dissipation hole	Dissipates heat
⑤	Status indicator	Indicates module running status
⑥	Snap-fit joint	Fastens the module

No.	Name	Description
⑦	Slot terminal	Implements electrical connection to the HCU control module
⑧	Nameplate	Includes the HCAN-10 module model number and serial number
⑨	Guide post	Aligns the HCAN-10 module with the HCU control module for easy installation

Outline dimensions

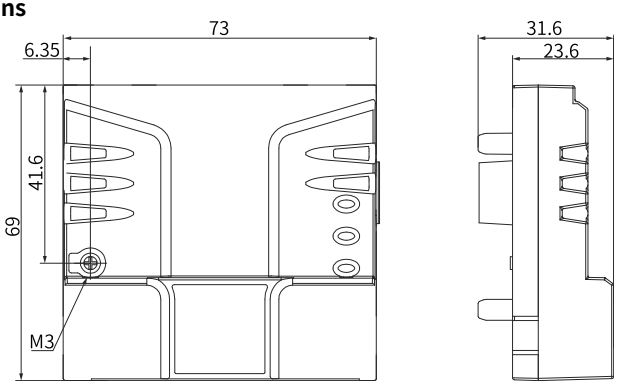


Figure 8-4 Outline dimensions (unit: mm)

8.3.2 Communication Port

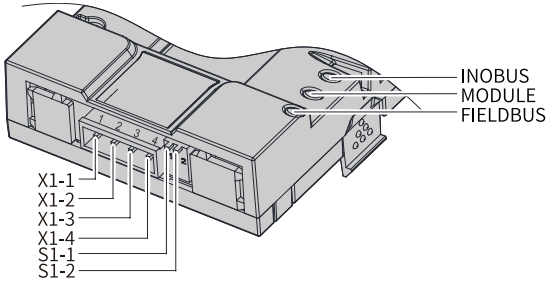


Figure 8-5 Terminal assignment

Terminal Pin	Name	Wiring	Description
X1-1	PE	Connect this pin to the shield.	Recommended cable: 4-conductor shielded twisted pair cable, with a cross-sectional area of 0.3 mm^2 to 2.0 mm^2
X1-2	CANH	Twisted pair	
X1-3	CANL		
X1-4	CGND	Connect this pin.	
S1-1	DIP switch 1	-	Termination resistor selection. The termination resistor is connected when S1-1 and S1-2 both are set to ON.
S1-2	DIP switch 2	-	

Terminal wiring:

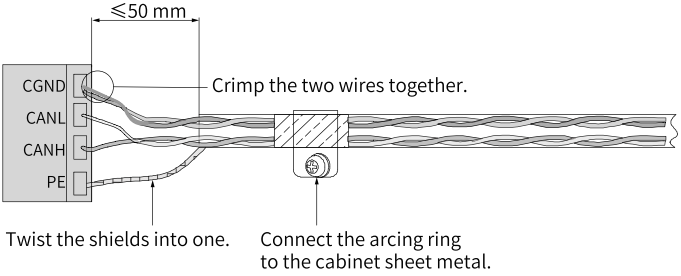


Figure 8-6 Cable

8.3.3 Networking Topology

The CAN bus supports a maximum of 63 nodes (excluding the master). The following figure shows the networking. The termination resistors of the master and the last node must be set to ON.

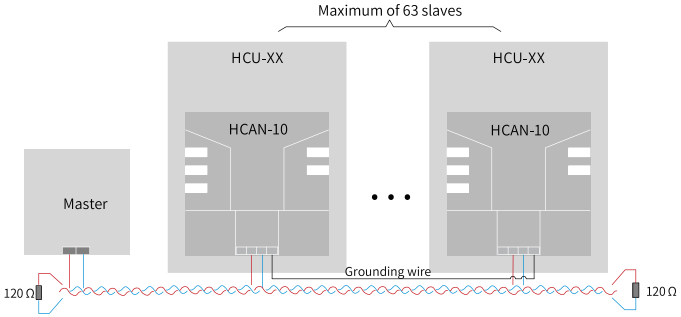


Figure 8-7 Networking diagram

The transmission distance of the CAN bus is directly related to the baud rate and the communication cable. The following table shows the relationship between the maximum bus length and the baud rate. This table is for reference only and the actual data may vary with cable impedance.

Table 8–6 CAN transmission distance

No.	Transmission Distance	Rate	Number of Nodes	Cable Gauge
1	25 m	1 Mbps	64	0.34 mm^2
2	95 m	500 kbps	64	0.34 mm^2
3	560 m	100 kbps	64	0.50 mm^2
4	1100 m	50 kbps	64	0.75 mm^2

8.4 Communication Parameters

Table 8–7 HCAN-10 module parameters

Param.	Definition	Default	Setpoint	Description
n1-00	FBA-A matching bus type	0	3: CANopen module	Use either of these parameter to select the communication module type. Do not set both adapters to the same communication module to avoid communication errors.
n2-00	FBA-B matching bus type	0	3: CANopen module	
n14-00	SLOT selection	0	0: Disable 1: Expansion slot 1_1 2: Expansion slot 1_2 3: Expansion slot 1_3 4: Expansion slot 2_1 5: Expansion slot 2_2 6: Expansion slot 2_3 7: Expansion slot 3_1	Select an expansion slot as needed.
n12-01	Module online/offline state	0	0: Offline 1: Online	-

Param.	Definition	Default	Setpoint	Description
n14-02	CANopen node address	1	0 to 63	This parameter sets the local node address. The value range is 1 to 63, and 0 indicates an invalid address.
n14-03	CANopen baud rate	5	0: 20 kbps 1: 50 kbps 2: 100 kbps 3: 125 kbps 4: 250 kbps 5: 500 kbps 6: 1 Mbps	This parameter sets the CAN network baud rate.
n14-07	Auto entering OP upon power-on	0	0: Disable 1: Enable	When this function is enabled, the drive automatically sends data to the master station upon power-on.
n14-08	Transmitting boot-up upon communication timeout	0	0: Disable 1: Enable	-
n14-09	Communication timeout interval	0.0	0.0 to 60.0	-

8.5 Configuration Procedure

Follow these steps to configure the HCAN-10 module:

1. Set the HCAN-10 module parameters.
2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the CANopen module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to CANopen module.
4. After parameter configuration, power on the HCU again. If all the three status indicators on the HCAN-10 module are solid green, it indicates normal communication between the master and slave stations.

9 Modbus TCP Communication

9.1 Overview

The HMBT-10 Modbus TCP adapter module complies with the internationally recognized Modbus TCP industrial Ethernet standard. The HMBT-10 can be used on MD880 series AC drives to turn the drive into a Modbus TCP slave controlled by the Modbus TCP master. This improves the communication efficiency of the system and enriches the networking functions of the MD880 series drives

Features:

- Supports switch networking instead of router networking.
- Supports up to two Modbus TCP client connections, with a minimum communication cycle of 6 ms, or 12 ms when two clients are connected.
- Supports reading of up to 123 parameters in a single operation.
- Supports networking of up to 253 devices.
- Supports up to 100 m transmission distance between the module and the switch.

9.2 Protocol Introduction

9.2.1 Modbus TCP Communication Introduction

The Modbus TCP protocol adds an MBAP header based on the RTU protocol. As TCP is a reliable-connection-based service, it does not need the CRC check code used in the RTU protocol. Therefore, Modbus TCP does not include a CRC check code.

Comparison of Modbus RTU and Modbus TCP read commands:

Type	MBAP Header	Address Code	Param.	Register Address	Number of Registers	CRC
RTU	None	01	03	01 8E	00 04	25 DE
TCP	00 00 00 00 00 06 00	None	03	01 8E	00 04	None

Command meaning: Read four registers starting from register address 0x18E on the module with address code 01.

Comparison of Modbus RTU and Modbus TCP write commands:

Type	MBAP Header	Address Code	Param.	Register Address	Number of Registers	Data Length	Body	CRC
RTU	None	01	10	01 8E	00 01	02	00 00	A8 7E
TCP	00000000 00 09 00	None	10	01 8E	00 01	02	00 00	None

9.2.2 Modbus-Specific Parameters

Control through continuous addresses

The Modbus RTU and Modbus TCP support writing of process data n1-68 to n1-83 (n2-68 to n2-83). After converted by communication base values of n1-36 to n1-51 (n2-36 to n2-51), the process data is updated to the corresponding connector U5-44 to U5-59 (U5-60 to U5-75).

Fieldbus adapter A:

Address	Write Parameter	Name	Description
33807	n1-68	[Process data input 1] Data display	Written through Modbus, converted by the base value of n1-36, and updated to connector U5-44
33808	n1-69	[Process data input 2] Data display	Written through Modbus, converted by the base value of n1-37, and updated to connector U5-45
33809	n1-70	[Process data input 3] Data display	Written through Modbus, converted by the base value of n1-38, and updated to connector U5-46
33810	n1-71	[Process data input 4] Data display	Written through Modbus, converted by the base value of n1-39, and updated to connector U5-47
33811	n1-72	[Process data input 5] Data display	Written through Modbus, converted by the base value of n1-40, and updated to connector U5-48
33812	n1-73	[Process data input 6] Data display	Written through Modbus, converted by the base value of n1-41, and updated to connector U5-49

Address	Write Parameter	Name	Description
33813	n1-74	[Process data input 7] Data display	Written through Modbus, converted by the base value of n1-42, and updated to connector U5-50
33814	n1-75	[Process data input 8] Data display	Written through Modbus, converted by the base value of n1-43, and updated to connector U5-51
33815	n1-76	[Process data input 9] Data display	Written through Modbus, converted by the base value of n1-44, and updated to connector U5-52
33816	n1-77	[Process data input 10] Data display	Written through Modbus, converted by the base value of n1-45, and updated to connector U5-53
33817	n1-78	[Process data input 11] Data display	Written through Modbus, converted by the base value of n1-46, and updated to connector U5-54
33818	n1-79	[Process data input 12] Data display	Written through Modbus, converted by the base value of n1-47, and updated to connector U5-55
33819	n1-80	[Process data input 13] Data display	Written through Modbus, converted by the base value of n1-48, and updated to connector U5-56
33820	n1-81	[Process data input 14] Data display	Written through Modbus, converted by the base value of n1-49, and updated to connector U5-57
33821	n1-82	[Process data input 15] Data display	Written through Modbus, converted by the base value of n1-50, and updated to connector U5-58
33822	n1-83	[Process data input 16] Data display	Written through Modbus, converted by the base value of n1-51, and updated to connector U5-59

Fieldbus adapter B

Address	Write Parameter	Name	Description
33839	n2-68	[Process data input 1] Data display	Written through Modbus, converted by the base value of n2-36, and updated to connector U5-60
33840	n2-69	[Process data input 2] Data display	Written through Modbus, converted by the base value of n2-37, and updated to connector U5-61
33841	n2-70	[Process data input 3] Data display	Written through Modbus, converted by the base value of n2-38, and updated to connector U5-62
33842	n2-71	[Process data input 4] Data display	Written through Modbus, converted by the base value of n2-39, and updated to connector U5-63
33843	n2-72	[Process data input 5] Data display	Written through Modbus, converted by the base value of n2-40, and updated to connector U5-64
33844	n2-73	[Process data input 6] Data display	Written through Modbus, converted by the base value of n2-41, and updated to connector U5-65
33845	n2-74	[Process data input 7] Data display	Written through Modbus, converted by the base value of n2-42, and updated to connector U5-66
33846	n2-75	[Process data input 8] Data display	Written through Modbus, converted by the base value of n2-43, and updated to connector U5-67
33847	n2-76	[Process data input 9] Data display	Written through Modbus, converted by the base value of n2-44, and updated to connector U5-68
33848	n2-77	[Process data input 10] Data display	Written through Modbus, converted by the base value of n2-45, and updated to connector U5-69
33849	n2-78	[Process data input 11] Data display	Written through Modbus, converted by the base value of n2-46, and updated to connector U5-70

Address	Write Parameter	Name	Description
33850	n2-79	[Process data input 12] Data display	Written through Modbus, converted by the base value of n2-47, and updated to connector U5-71
33851	n2-80	[Process data input 13] Data display	Written through Modbus, converted by the base value of n2-48, and updated to connector U5-72
33852	n2-81	[Process data input 14] Data display	Written through Modbus, converted by the base value of n2-49, and updated to connector U5-73
33853	n2-82	[Process data input 15] Data display	Written through Modbus, converted by the base value of n2-50, and updated to connector U5-74
33854	n2-83	[Process data input 16] Data display	Written through Modbus, converted by the base value of n2-51, and updated to connector U5-75

For more information on using continuous addresses as the start, stop, and reset command sources of the drive through the Modbus, see the "Use of PZD1" section in the *MD880 Series AC Drive Commissioning Guide (HCU-50S, HCU-50S-CS, HCU-51S, and HCU-51S-CS)*.

Control through special addresses

To start/stop the MD880 drive through Modbus communication, the HCU-50S/51S provides several special communication addresses for accessing Modbus, and updates the written data to a specific connector for reading the source parameters. As specified by the Modbus communication protocol, parameters are read and written directly through the communication address and no process data will be produced. However, b0-01/b2-01 of the OFF1 source is not allowed to be set to 1 to start the drive and a connector data must be selected.

Note

Addresses of the following connectors are read-only in the software tool. Write operations can only be performed using the addresses specified in the following table.

Address	Written to Connector	Name	Description
34196<1>	U1-75	Start/Stop command (Modbus)	Values: 0 and 1. Recommended for user-defined OFF1 command.
34230	U5-40	Modbus communication setting value 1	Value range: –32768 to +32767
34231	U5-41	Modbus communication setting value 2	
34232	U9-50	Modbus communication double-word setpoint	Value range: -2^{31} to $2^{31} - 1$. Suitable for high-precision speed setting. The high 16 bits of the data are written to address 34232, whereas the low 16 bits are written to address 34233.

Note

<1>: This address is only applicable to the HCU-50/51 and HCU-50S/51S series.

9.3 Hardware Configuration

9.3.1 Module Introduction

Appearance

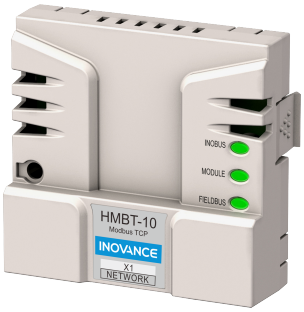


Figure 9-1 Appearance of the HMBT-10

Model and nameplate

HMBT-10
① ②

① **Product code 1** HMBT: Modbus TCP adapter module	② **Product code 2** 10: Version number

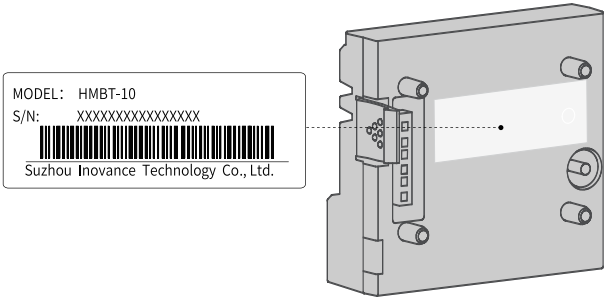


Figure 9-2 Nameplate

Components

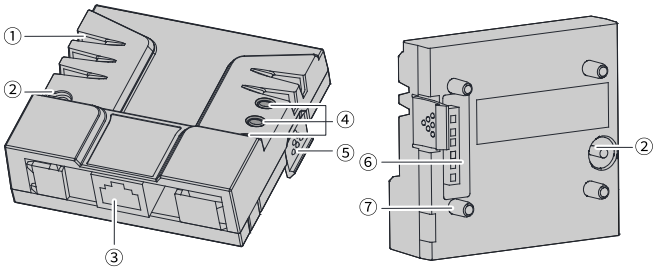


Figure 9-3 Product diagram

Table 9–1 Interfaces

No.	Name	Description
①	Heat dissipation hole	Dissipates heat
②	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
③	Modbus TCP communication terminal X1	Standard Ethernet RJ45 communication terminal
④	Status indicator	Indicates module and bus running status

No.	Name	Description
⑤	Snap-fit joint	Fastens the module
⑥	Slot terminal	Implements electrical connection to the HCU control module
⑦	Guide post	Aligns the HMBT-10 module with the HCU control module for easy installation

Outline dimensions

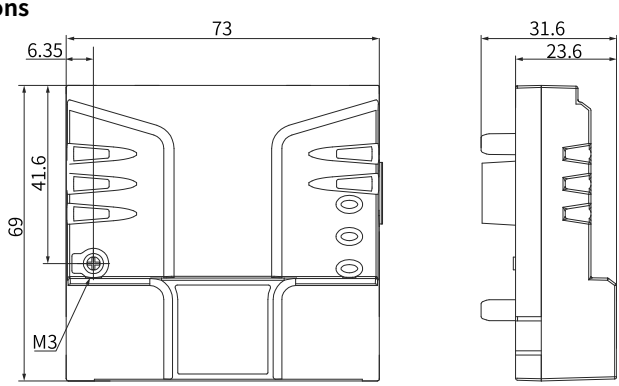


Figure 9-4 Outline dimensions (unit: mm)

9.3.2 Networking Topology

The HMBT-10 module uses a standard Ethernet RJ45 socket to connect to the PLC master. Its pin signal assignment is consistent with standard Ethernet pins, and both crossover and straight-through cable routing methods can be used.

Single-module networking

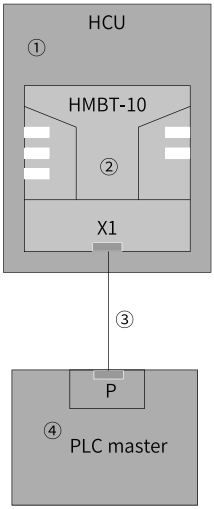


Figure 9-5 Single-module networking topology

Table 9–2 Elements in single-module networking

No.	Name
①	HCU control module
②	HMBT-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	PLC master

Multi-module networking

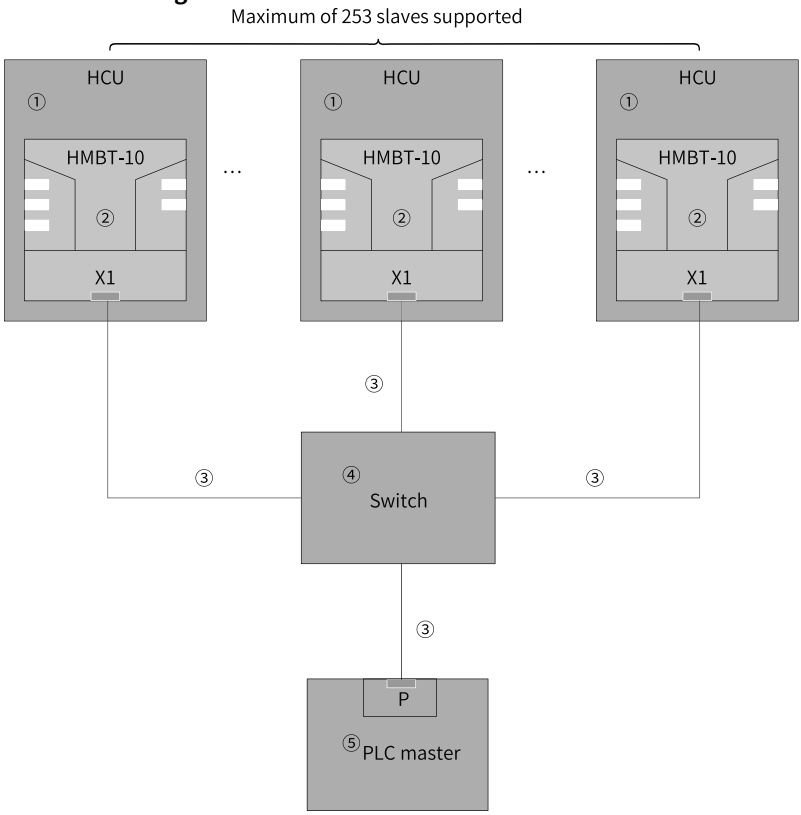


Figure 9-6 Multi-module networking topology

Table 9–3 Elements in multi-module networking

No.	Name
①	HCU control module
②	HMBT-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	Switch
⑤	PLC master

9.4 Communication Parameters

Table 9–4 HMBT-10 module parameters

Param.	Definition	Default	Setpoint	Description
n1-00	FBA-A matching bus type	0	9: Customized communication module	Use either of these parameter to select the communication module type. Do not set both adapters to the same communication module to avoid communication errors.
n2-00	FBA-B matching bus type	0	9: Customized communication module	
n18-00	SLOT selection	0	0: Disable 1: Expansion slot 1_1 2: Expansion slot 1_2 3: Expansion slot 1_3 4: Expansion slot 2_1 5: Expansion slot 2_2 6: Expansion slot 2_3 7: Expansion slot 3_1	Select an expansion slot as needed.
n18-01	Module online/offline state	0	0: Offline 1: Online	-
n18-02	Expansion slot module type	0	0: No expansion card	Upon correct configuration, this parameter shows "116: ModBus TCP Module".
n18-03	Customized communication module ID setting	0	116	If this parameter is not set or set incorrectly, the expansion card status will be offline.
n18-11 to n18-13	Customized communication card parameters 1 to 3	Reserved	-	-

Param.	Definition	Default	Setpoint	Description
n18-14 to n18-17	Customized communication card parameters 4 to 7	Specifies the IP address of the Modbus TCP communication	Value range: 0 to 255	Use and set this parameter as needed.
n18-18 to n18-21	Customized communication card parameters 8 to 11	Specifies the subnet mask of the Modbus TCP communication	Value range: 0 to 255	Use and set this parameter as needed.
n18-22 to n18-25	Customized communication card parameters 12 to 15	Specifies the gateway of the Modbus TCP communication	Value range: 0 to 255	Use and set this parameter as needed.
n18-26	Customized communication card parameter 16	Reserved	-	-

Param.	Definition	Default	Setpoint	Description
n18-27	Customized communication card parameter 17	Specifies the timeout interval for receiving the Modbus TCP data in the unit of 100 ms. When this parameter is set to 0, the timeout detection is disabled.	-	Use and set this parameter as needed.
n18-28	Customized communication card parameter 18	Reserved	-	-

Table 9–5 IP address configuration example

Param.	Definition	Default	Category
n18-14	Customized communication card parameter 4	192	IP address
n18-15	Customized communication card parameter 5	168	
n18-16	Customized communication card parameter 6	0	
n18-17	Customized communication card parameter 7	2	
n18-18	Customized communication card parameter 8	255	Subnet mask
n18-19	Customized communication card parameter 9	255	
n18-20	Customized communication card parameter 10	255	
n18-21	Customized communication card parameter 11	0	
n18-22	Customized communication card parameter 12	192	Gateway
n18-23	Customized communication card parameter 13	168	
n18-24	Customized communication card parameter 14	0	
n18-25	Customized communication card parameter 15	1	

9.5 Configuration Procedure

Follow these steps to configure the HMBT-10 module:

1. Set the HMBT-10 module parameters.

2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the customized communication module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to customized communication module.
4. After parameter configuration, power on the HCU again. If all the three status indicators on the HMBT-10 module are solid green, it indicates normal communication between the master and slave stations. If the FIELDBUS indicator is flashing green, it indicates data exchange between the master and slave stations.

10 EtherCAT Communication

10.1 Overview

The HETC-10 EtherCAT industrial Ethernet bus slave module complies with the internationally recognized EtherCAT industrial Ethernet standard. The HETC-10 can be used on MD880 series AC drives to turn the drive into an EtherCAT slave controlled by the EtherCAT master. This improves the communication efficiency of the system and enriches the networking functions of the MD880 series drives.

Features:

- Passed the Conformance Test Tool (CTT) test.
- Supports up to 16 RXPDOs and 16 TXPDOs, and supports dynamic configuration of PDOs.
- Supports DC synchronous clock mode.
- Supports CoE protocol (PDO+SDO).
- Features a minimum DC cycle of 250 μs (standard version).
- Supports multiple types of topologies including chain, bus, tree, and star topologies.
- Directly installed in the expansion slot of the HCU control module, requiring no external power supply.

10.2 Protocol Introduction

10.2.1 EtherCAT Protocol Introduction

EtherCAT, short for Ethernet Control Automation Technology, is a real-time Ethernet technology developed by the German company Beckhoff. EtherCAT is based on standard Ethernet frame transmission, with a single frame capable of carrying up to 1486 bytes of data. This gives EtherCAT a significant advantage in terms of data transmission capacity.

- EtherCAT supports almost all types of topologies, including star, linear, tree, and daisy-chain topologies, and accommodates various communication media such as electrical cables and optical fiber.
- EtherCAT supports hot-swapping, ensuring flexibility in device connectivity.
- EtherCAT has virtually no device capacity limit, supporting up to 65,535 slave devices, eliminating the need for switches in the network. Data can reach each slave station directly through the topology structure.

10.2.2 PDO Data

The MD880 transmits data according to the selected transmission format of PZD with different length. The types of transmission formats supported depend on the xml file. The MD880 supports a maximum of 16 parameters to be written periodically, and a maximum of

16 parameters to be read periodically. You can add or delete parameters to be written or read in the device configuration as required.

10.2.3SDO Data

EtherCAT service data object (SDO) is used to transfer non-cyclic data, such as communication parameter configurations and drive running parameter configurations.

The CANopen over EtherCAT (CoE) service types include: emergency event message, SDO request, SDO response, TxPDO, RxPDO, remote TxPDO transmit request, remote RxPDO transmit request, and SDO information.

The SDO data can be used to read or write a single parameter of the MD880.

10.3 Hardware Configuration

10.3.1Module Introduction

Appearance

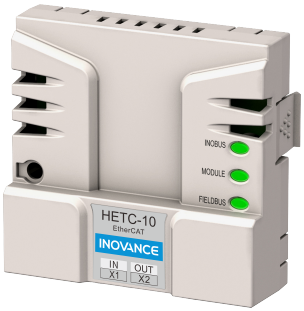


Figure 10-1 Appearance of the HETC-10

Model and nameplate

HETC-10
① ②

① **Product code 1** HETC: EtherCAT industrial bus module	② **Product code 2** 10: Standard version

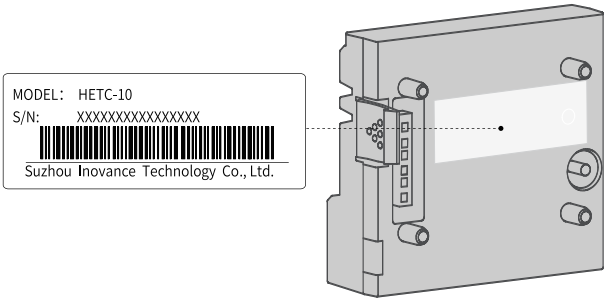


Figure 10-2 Nameplate

Components

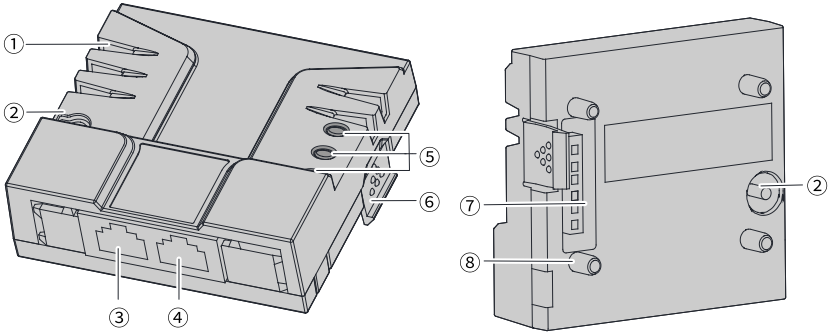


Figure 10-3 Product diagram

Table 10–1 Interfaces

No.	Name	Description
①	Heat dissipation hole	Dissipates heat
②	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
③	EtherCAT communication terminal X1	Communication input terminal
④	EtherCAT communication terminal X2	Communication output terminal
⑤	Status indicator	Indicates module and bus running status
⑥	Snap-fit joint	Fastens the module

No.	Name	Description
⑦	Slot terminal	Implements electrical connection to the HCU control module
⑧	Guide post	Aligns the HETC-10 module with the HCU control module for easy installation

Outline dimensions

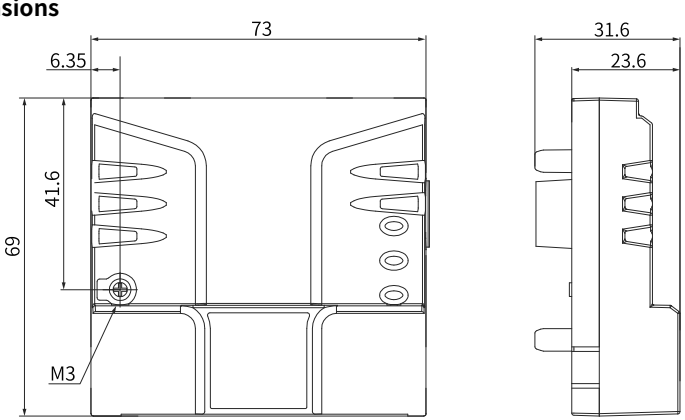


Figure 10-4 Outline dimensions (unit: mm)

10.3.2Networking Topology

EtherCAT supports various topology structures, including chain, bus, tree, and star types. This section describes the chain and star types.

Chain networking

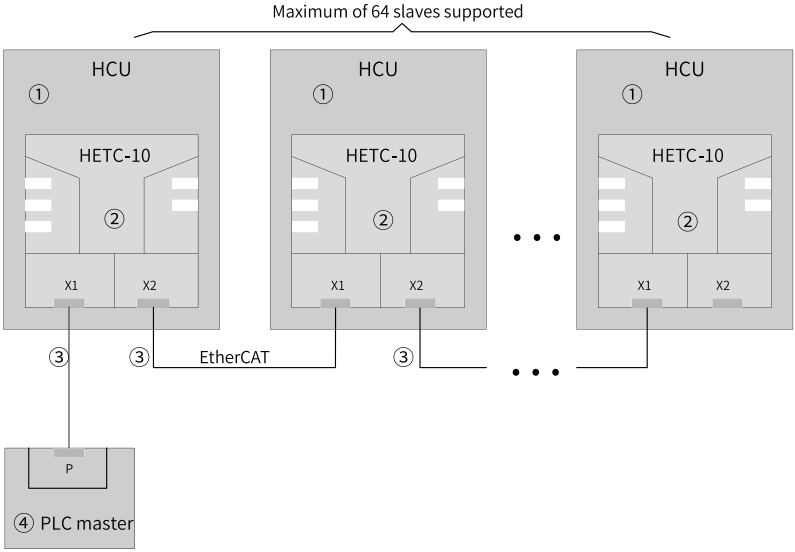


Figure 10-5 EtherCAT chain networking topology

Table 10–2 Elements in chain networking

No.	Name
①	HCU control module
②	HETC-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	PLC master

Star networking

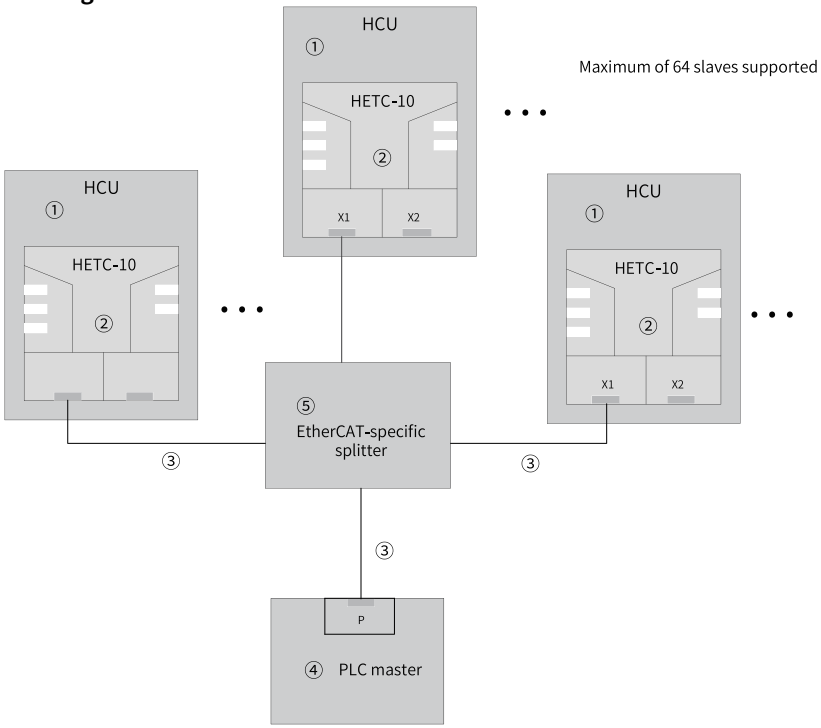


Figure 10-6 Star networking topology

Table 10–3 Elements in star networking

No.	Name
①	HCU control module
②	HETC-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	PLC master
⑤	EtherCAT-specific splitter

10.4 Communication Parameters

Table 10–4 HETC-10 module parameters

Param.	Definition	Default	Setpoint	Description
n1-00	FBA-A matching bus type	0	5: EtherCAT module	Use either of these parameter to select the communication module type. Do not set both adapters to the same communication module to avoid communication errors.
n2-00	FBA-B matching bus type	0	5: EtherCAT module	
n14-00	SLOT selection	0	0: Disable 1: Expansion slot 1_1 2: Expansion slot 1_2 3: Expansion slot 1_3 4: Expansion slot 2_1 5: Expansion slot 2_2 6: Expansion slot 2_3 7: Expansion slot 3_1	Select an expansion slot as needed.
n14-01	Module online/ offline state	0	0: Offline 1: Online	-
n14-02	EtherCAT slave name	1	0 to 65535	-
n14-03	EtherCAT slave alias	1	0 to 65535	-
n14-04	Number of synchronization interrupts allowed by EtherCAT	4	4 to 20	-
n14-06	Alias setting function switch	1	0: Set by n14-03 1: Set by host controller master	-
n14-11	Port 1_0 receive error count	0	0 to 65535	-
n14-12	Port 1_0 invalid frame count	0	0 to 65535	-

Param.	Definition	Default	Setpoint	Description
n14-13	Port 1_0 forward error limit	0	0 to 65535	-
n14-14	PDI and processing unit error count	0	0 to 65535	-
n14-15	Port 1_0 link loss count	0	0 to 65535	-
n14-16	Synchronization signal loss count	0	0 to 65535	-
n14-17	Application layer fault code	0	0 to 65535	-
n14-18	State machine and port link state	0	0 to 65535	-
n14-22	InoBus scheduling period	1000	500 to 1000	-

10.5 Configuration Procedure

Follow these steps to configure the HETC-10 module:

1. Set the HETC-10 module parameters.
2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the EtherCAT module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to EtherCAT module.
4. After parameter configuration, power on the HCU again. If all the three status indicators on the HETC-10 module are solid green, it indicates normal communication between the master and slave stations.

11 EtherNet/IP Communication

11.1 Overview

- The HEIP-10 EtherNet/IP fieldbus adapter is a bus communication module for the MD880 series. It complies with the internationally recognized EtherNet/IP bus standard and features high efficiency, flexible topology, and easy operation.
- When used on the MD880 series drives, the HEIP-10 turns the drive into a fieldbus slave controlled by the fieldbus master. This improves the communication efficiency and implement drive networking functions.
- To use the HEIP-10 with the MD880 series AC drive, you need to configure relevant parameters on the AC drive. (The configuration principles for fieldbus adapters are the same, and this guide illustrates the configuration using fieldbus adapter A as an example.) First, modify the fieldbus adapter type in the AC drive software tool by setting parameter n1-00 to "12: EtherNet/IP card", and enable n41-00 for the corresponding slot. Then, verify the card connection by checking parameters A2-24 to A2-37 based on the slot position. The EDS file for the HEIP-10 module is named "MD880_EIP_V1.02.eds"; ensure the EDS version you use is V1.02 or later.
- This guide is only applicable to the MD880 series AC drive. If you need to use this expansion card with other AC drives, consult with Inovance technical support.

11.2 Protocol Introduction

11.2.1I/O Messages

The HEIP-10 supports 32 I/O messages for data transmission, of which 16 are master-to-slave messages and 16 are slave-to-master messages.

The I/O messages enable the master station to modify and read AC drive data in real time and exchange data with the AC drive periodically. Data communication addresses are directly configured by the AC drive.

Specifically, the I/O messages enable real-time exchange of function parameters and monitoring parameters between the AC drive and the EtherNet/IP master station. Periodic monitoring parameters can be configured through the AC drive. The I/O messages are used for periodic data exchange between the master station and the AC drive, as described in the following table.

Data Flow	Description	Message
I/O messages from the master station (O->T)	Modifies AC drive function parameters in real time	Output I/O messages 0–15
I/O messages from the AC drive (T->O)	Reads AC drive function parameters in real time	Input I/O messages 0–15

11.2.2Data Sent by the Master Station

The following table describes the data sent by the master station.

I/O Messages Sent by the Master Station	
I/O messages 0–15	Real-time modification of parameter values. You can configure the data as needed. For details about the configuration, see the data configuration for I/O messages.

11.2.3Data Returned by the AC drive

The following table describes the data returned by the AC drive.

I/O Messages Returned by the AC Drive	
I/O messages 0–15	Monitoring parameter values (U group). You can configure the U group connectors to correspond to I/O messages 0–15 as needed. For details about the configuration, see the data configuration for I/O messages.

11.3 Hardware Configuration

11.3.1Module Introduction

Appearance



Figure 11-1 Appearance of the HEIP-10

Model and nameplate

HEIP-10
① ②

① **Product code 1** HEIP: EtherNet/IP adapter module	② **Product code 2** 10: Version number

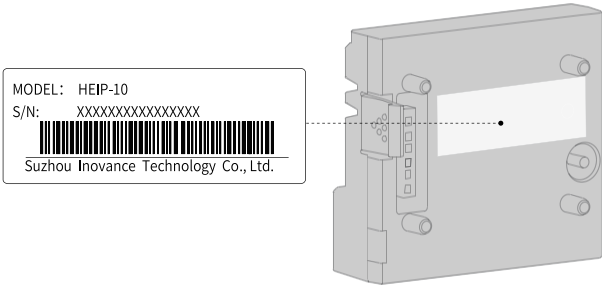


Figure 11-2 Nameplate

Components

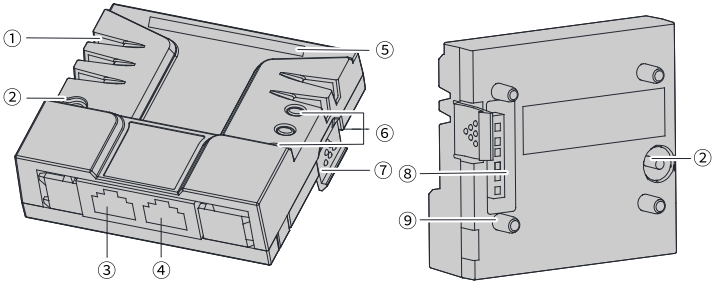


Figure 11-3 Product diagram

Table 11–1 Interfaces

No.	Name	Description
①	Heat dissipation hole	Dissipates heat
②	Fixing screw	Fastens the module to the HCU control module to ensure proper connection of the PE layer
③	Communication terminal X1	Communication terminal 1
④	Communication terminal X2	Communication terminal 2

No.	Name	Description
⑤	MAC address label	Indicates local MAC address
⑥	Status indicator	Indicates module and bus running status
⑦	Snap-fit joint	Fastens the module
⑧	Slot terminal	Implements electrical connection to the HCU control module
⑨	Guide post	Aligns the HEIP-10 module with the HCU control module for easy installation

Outline dimensions

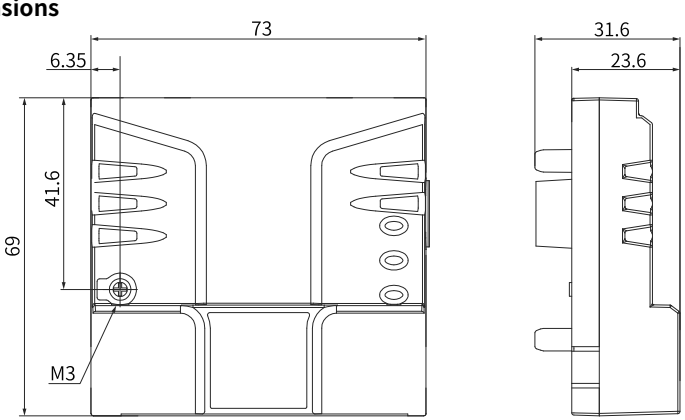


Figure 11-4 Outline dimensions (unit: mm)

11.3.2Networking Topology

EtherNet/IP supports a variety of topologies, including bus, star, and tree topologies. Diversified networking modes can be implemented by using switches.

Electrical connection for chain networking

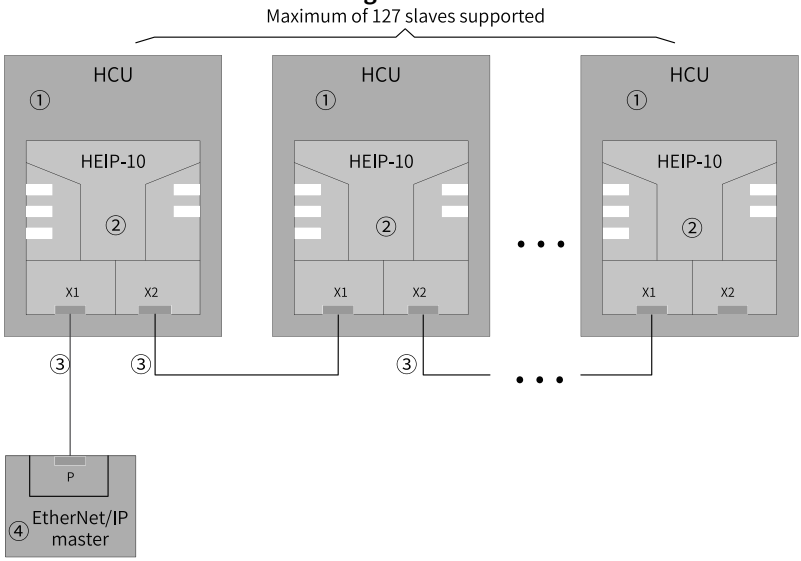


Figure 11-5 Diagram of electrical connection for chain networking

Table 11–2 Electrical connection for chain networking

No.	Name
①	HCU control module
②	HEIP-10 industrial Ethernet module
③	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
④	EtherNet/IP master

Electrical connection for star networking

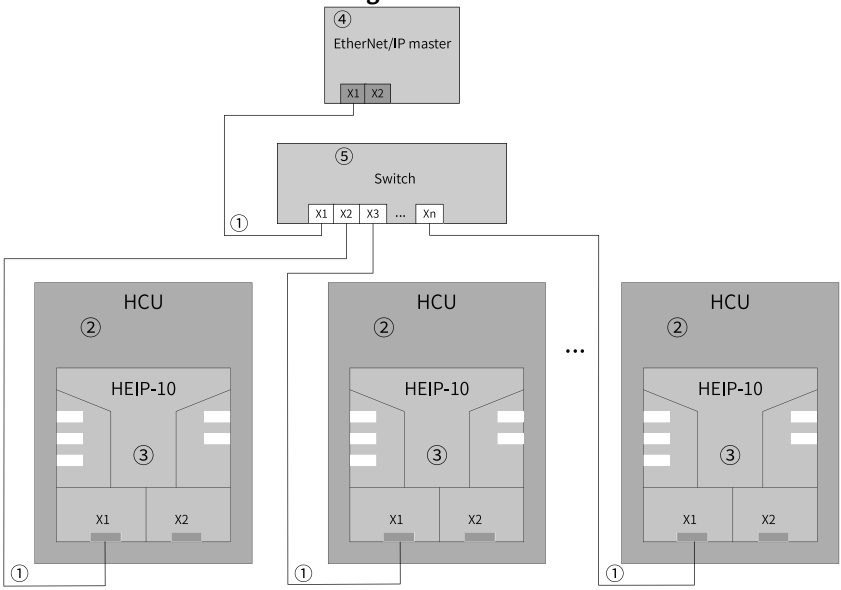


Figure 11-6 Diagram of electrical connection for star networking

Table 11–3 Electrical connection for star networking

No.	Name
①	Network cable (It is recommended to use shielded Cat 5e Ethernet cable, and the registered jack used for crimping must have a metal shell for earth screen protection.)
②	HCU control module
③	HEIP-10 industrial Ethernet module
④	EtherNet/IP master
⑤	Switch

11.4 Communication Parameters

Setting the AC drive communication card type

After powering on the AC drive, you need to set n1-00 (FBA-A matching bus type) or n2-00 (FBA-B matching bus type) and n41-00 (card slot selection) for the HEIP-10 to communicate normally with the AC drive. This section uses fieldbus adapter A as an example.

Param.	Name	Value Range	Setpoint	Meaning
n1-00	FBA-A matching bus type	0 to 12	12: EtherNet/IP	Sets the bus type to EtherNet/IP.
n41-00	Expansion card slot selection	0 to 7	7	Enables the slot (dependent on the slot position; this guide uses SLOT3 as an example).

Setting the HEIP-10 module IP address

Set the following parameters of the expansion card.

Param.	Name	Value Range	Meaning
n41-02	DHCP enable	0: Disable DHCP 1: Enable DHCP	Enables/Disables DHCP for the EtherNet/IP expansion card. When DHCP is enabled, the following IP address parameters are invalid.
n41-03	IP address	1 to 254	Sets the IP address of the EtherNet/IP expansion card.
n41-04 and n41-05		0 to 254	
n41-06		0 to 253	
n41-07 to n41-10	Subnet mask address	0 to 255	Sets the subnet mask of the EtherNet/IP expansion card.
n41-11 to n41-14	Gateway address	0 to 254	Sets the gateway address of the EtherNet/IP expansion card.

The IP address can be a static one or dynamic one assigned by DHCP function.

- The DHCP mode is set through n41-02.
- The static IP address is set through parameters n41-03 to n41-06.

For example, to configure a static IP address as 192.168.0.11, subnet mask as 255.255.255.0, and gateway as 192.168.0.1, set the parameters as follows:

Param.	Function	Setpoint
n41-02	DHCP enable	0
n41-03	IP address 1	192
n41-04	IP address 2	168
n41-05	IP address 3	0
n41-06	IP address 4	11

Param.	Function	Setpoint
n41-07	Subnet mask address 1	255
n41-08	Subnet mask address 2	255
n41-09	Subnet mask address 3	255
n41-10	Subnet mask address 4	0
n41-11	Gateway address 1	192
n41-12	Gateway address 2	168
n41-13	Gateway address 3	0
n41-14	Gateway address 4	1

When using the DHCP BOOTP function, you can check the MAC address by:

- Checking the MAC address label on the expansion card.
- Checking the AC drive parameters n41-15 to n41-20.

Note

By default, after the communication card is powered on, parameters n41-15 to n41-20 display the corresponding MAC address.

The expansion card also supports IP address conflict detection. When the IP address of this card is the same as that of another device in the network, the MODULE indicator becomes red and steady ON, and fault code 67-13 or 68-13 is reported. There are three IP address conflict scenarios, as described in the following table.

No.	Scenario	Symptom	Solution
1	Both devices support IP address conflict detection. The two devices are powered on or connected to the network non-simultaneously.	• If the two devices are adjacent to each other, both devices enter IP address conflict mode. • If not adjacent, the device that is powered on or connected to the network later enters IP address conflict mode.	Check the IP address settings and modify the duplicate IP address.
2	Both devices support IP address conflict detection. The two devices are powered on or connected to the network simultaneously.	Both devices enter IP address conflict mode simultaneously.	
3	One device supports IP conflict detection, whereas the other does not.	Regardless of power-on order, the device that does not support IP address conflict detection will occupy the IP address, whereas the device that supports IP address conflict detection will enter conflict mode.	

Note

- The expansion card actively detects conflicts during power-on and DHCP IP allocation, and then switches to passive detection.
- If an IP address conflict occurs during DHCP allocation, the IP address cannot be successfully assigned.

Viewing data in AC drive communication card parameters

Table 11–4 HEIP-10 module parameters

Param.	Name	Value Range	Description
n41-15	MAC address 1	0 to 255	MAC address of the expansion card
n41-16	MAC address 2	0 to 255	MAC address of the expansion card
n41-17	MAC address 3	0 to 255	MAC address of the expansion card

Param.	Name	Value Range	Description
n41-18	MAC address 4	0 to 255	MAC address of the expansion card
n41-19	MAC address 5	0 to 255	MAC address of the expansion card
n41-20	MAC address 6	0 to 255	MAC address of the expansion card

11.5 Configuration Procedure

Follow these steps to configure the HEIP-10 module:

1. Set the HEIP-10 module parameters.
2. Select either fieldbus adapter A or fieldbus adapter B. Do not set both fieldbus adapters to the EtherNet/IP module type; otherwise, a communication error will occur.
3. Set the bus type of the selected fieldbus adapter to EtherNet/IP module.
4. After parameter configuration, power on the HCU again. If all the three status indicators on the HEIP-10 module are solid green, it indicates normal communication between the master and slave stations.

12 Examples for Communication Settings

12.1 PROFIBUS DP

Networking

In this example, the PLC used is a Siemens S7-300, model 6AG1 315-2EH13-2AB0. The PLC programming software is Siemens TIA Portal V15. The MD880 drive commissioning software is InoDriveStudio, which can be obtained from the Inovance official website at https://www.inovance.com.

Connect devices according to the following figure to complete networking. Set the SOP-20 operating panel to the USB relay mode.

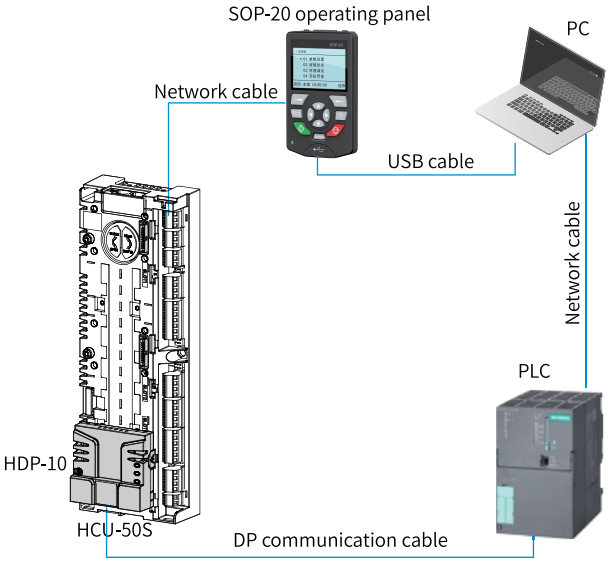
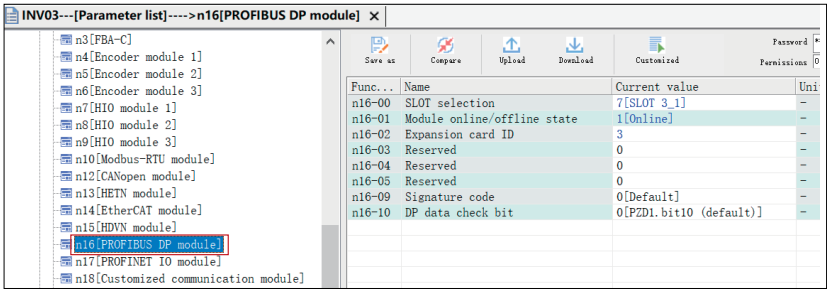


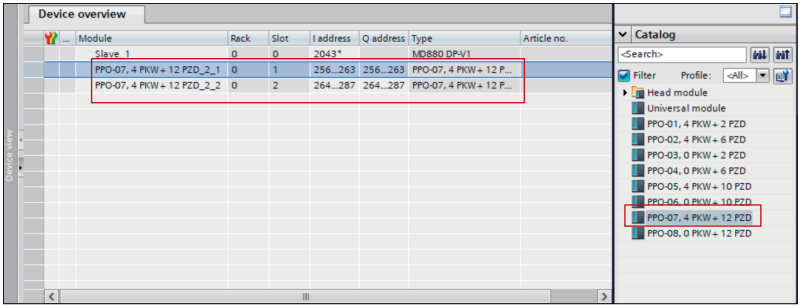
Figure 12-1 PROFIBUS DP networking of MD880 with Siemens PLC

Process data configuration

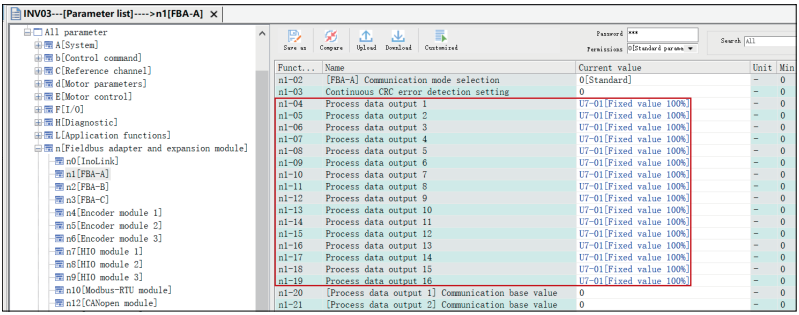
1. After downloading the PLC configuration, use InoDriveStudio to connect to the MD880 drive. See the following figure for settings of parameters in the n16 group used in this example.



2. After loading the PLC configuration to the PLC, verify the transmission of process data, that is, check whether data can be transmitted between the MD880 and the PLC. In TIA Portal V15, the process data of the MD880 has been mapped to addresses in the PLC. The PPO data type is 4PKW+12PZD, as shown in the following figure. PKW1 to PKW4 are mapped to the PLC's process image input/output addresses 256 to 263, occupying 4 words (8 bytes). PZD1 to PZD12 are mapped to the PLC's process image input/output addresses 264 to 287, occupying 12 words (24 bytes).

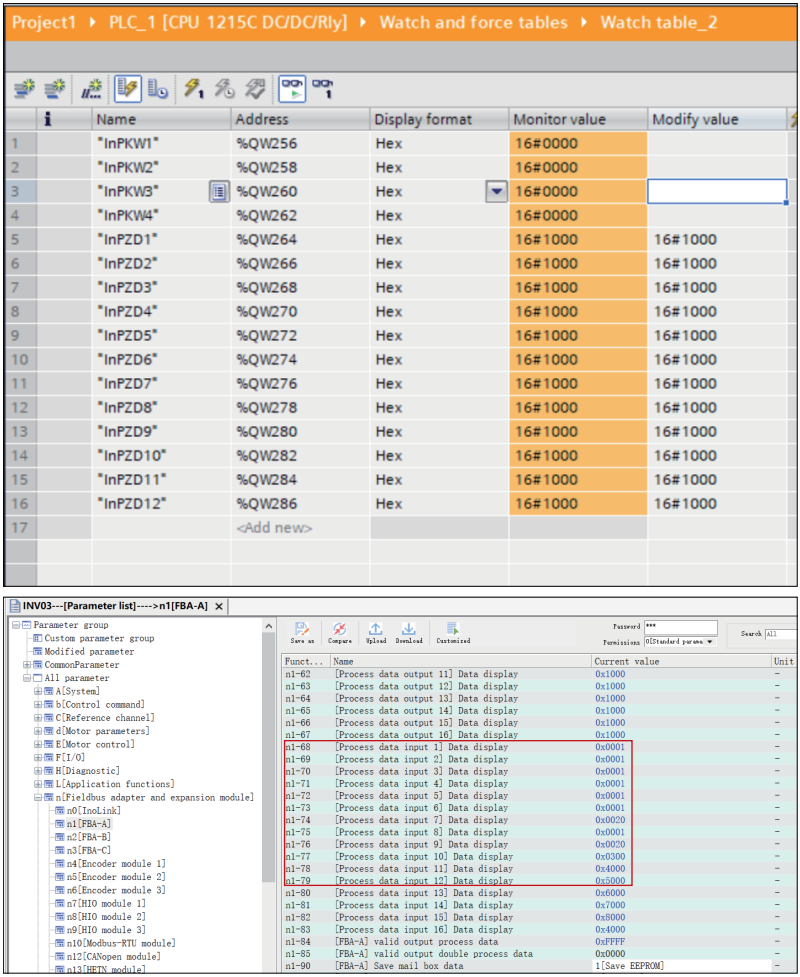


3. Configure the data to be written from the MD880 drive to the PLC using parameters n1-04 to n1-15. A sent fixed value of 100% corresponds to 16#1000, as shown in the following figure.



4. After configuring parameters n1-04 to n1-15, observe the data sent by the MD880 drive in TIA Portal V15. The corresponding addresses are IW264 to IW286. The PLC has received the data sent by the MD880 drive, indicating normal data transmission from the MD880.

5. You can view the raw data written from the PLC to the MD880 drive using parameters n1-68 to n1-79. For input process data that has been converted by the communication base value, you can view them using U5-44 to U5-55. As shown in the following figure, the MD880 drive can successfully receive the data written by the PLC.



Project1 ▸ PLC_1 [CPU 1215C DC/DC/Rly] ▸ Watch and force tables ▸ Watch table_2

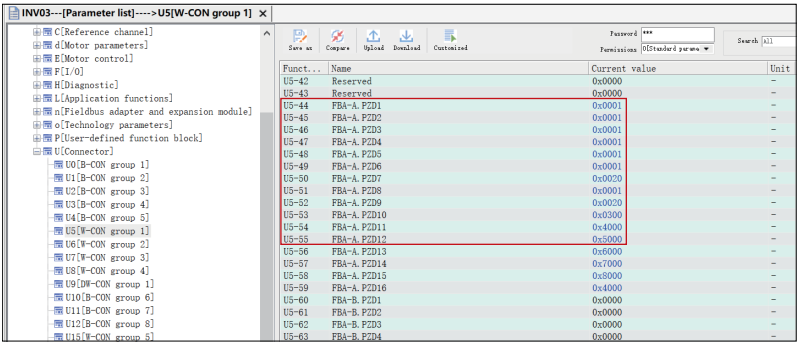
	i	Name	Address	Display format	Monitor value	Modify value
1		"InPKW1"	%QW256	Hex	16#0000	
2		"InPKW2"	%QW258	Hex	16#0000	
3		"InPKW3"	%QW260	Hex	16#0000	
4		"InPKW4"	%QW262	Hex	16#0000	
5		"InPZD1"	%QW264	Hex	16#1000	16#1000
6		"InPZD2"	%QW266	Hex	16#1000	16#1000
7		"InPZD3"	%QW268	Hex	16#1000	16#1000
8		"InPZD4"	%QW270	Hex	16#1000	16#1000
9		"InPZD5"	%QW272	Hex	16#1000	16#1000
10		"InPZD6"	%QW274	Hex	16#1000	16#1000
11		"InPZD7"	%QW276	Hex	16#1000	16#1000
12		"InPZD8"	%QW278	Hex	16#1000	16#1000
13		"InPZD9"	%QW280	Hex	16#1000	16#1000
14		"InPZD10"	%QW282	Hex	16#1000	16#1000
15		"InPZD11"	%QW284	Hex	16#1000	16#1000
16		"InPZD12"	%QW286	Hex	16#1000	16#1000
17			<Add new>			

INV03---[Parameter list]---->n1[FBA-A]

Parameter group
- Custom parameter group
- Modified parameter
- CommonParameter
- All parameter
  - A[System]
  - b[Control command]
  - C[Reference channel]
  - d[Motor parameters]
  - E[Motor control]
  - F[I/O]
  - H[Diagnostic]
  - L[Application functions]
  - n[Fieldbus adapter and expansion module]
    - n0[InoLink]
    - n1[FBA-A]
    - n2[FBA-B]
    - n3[FBA-C]
    - n4[Encoder module 1]
    - n5[Encoder module 2]
    - n6[Encoder module 3]
    - n7[HIO module 1]
    - n8[HIO module 2]
    - n9[HIO module 3]
    - n10[Modbus-RTU module]
    - n12[CANopen module]
    - n13[HETN module]

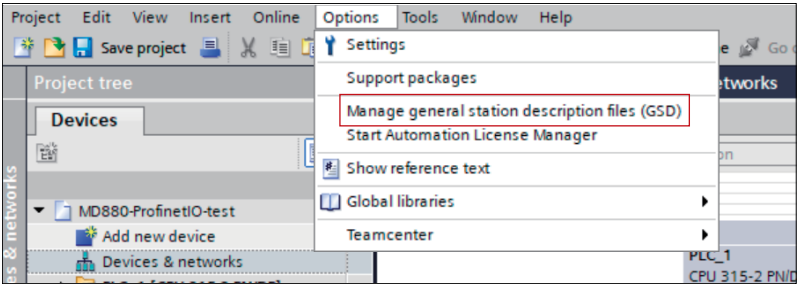
Save as | Compare | Upload | Download | Customized | Password | Permissions | Search

Funct...	Name	Current value	Unit
n1-62	[Process data output 11] Data display	0x1000	-
n1-63	[Process data output 12] Data display	0x1000	-
n1-64	[Process data output 13] Data display	0x1000	-
n1-65	[Process data output 14] Data display	0x1000	-
n1-66	[Process data output 15] Data display	0x1000	-
n1-67	[Process data output 16] Data display	0x1000	-
n1-68	[Process data input 1] Data display	0x0001	-
n1-69	[Process data input 2] Data display	0x0001	-
n1-70	[Process data input 3] Data display	0x0001	-
n1-71	[Process data input 4] Data display	0x0001	-
n1-72	[Process data input 5] Data display	0x0001	-
n1-73	[Process data input 6] Data display	0x0001	-
n1-74	[Process data input 7] Data display	0x0020	-
n1-75	[Process data input 8] Data display	0x0001	-
n1-76	[Process data input 9] Data display	0x0020	-
n1-77	[Process data input 10] Data display	0x0300	-
n1-78	[Process data input 11] Data display	0x4000	-
n1-79	[Process data input 12] Data display	0x5000	-
n1-80	[Process data input 13] Data display	0x6000	-
n1-81	[Process data input 14] Data display	0x7000	-
n1-82	[Process data input 15] Data display	0x8000	-
n1-83	[Process data input 16] Data display	0x4000	-
n1-84	[FBA-A] valid output process data	0xFFFF	-
n1-85	[FBA-A] valid output double process data	0x0000	-
n1-90	[FBA-A] Save mail box data	1[Save EEPROM]	-

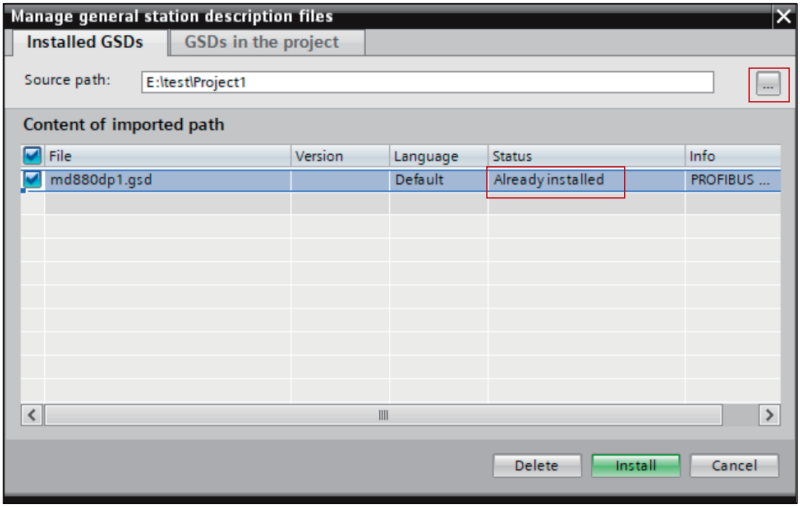


Configuration in TIA Portal V15

1. Install the GSD file for the MD880 drive in TIA Portal V15. You may obtain the GSD file from Inovance or its agent.



a. Input the path where the GSD file is located and click "Install", as shown in the following figure.



b. Once the interface below appears, the installation is completed. Click "Close".

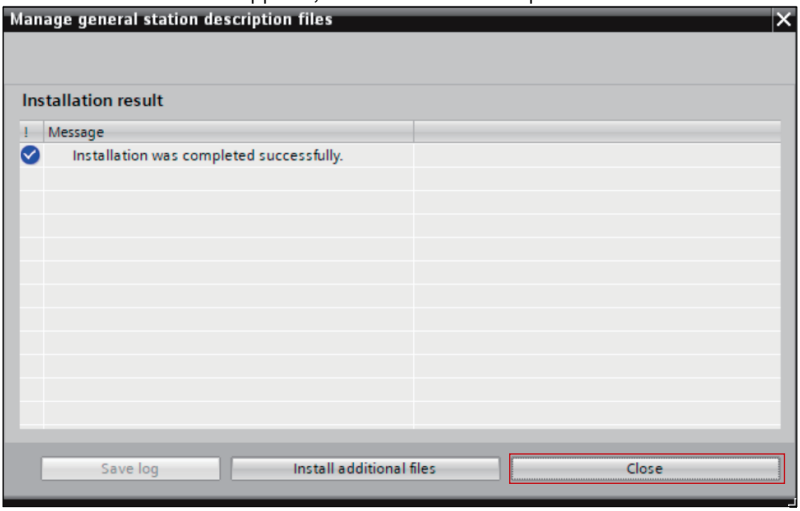
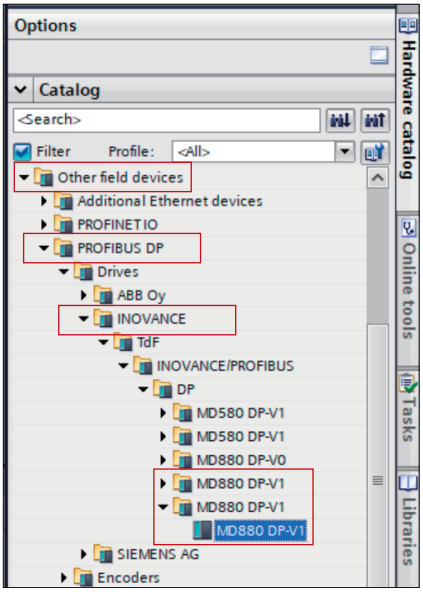


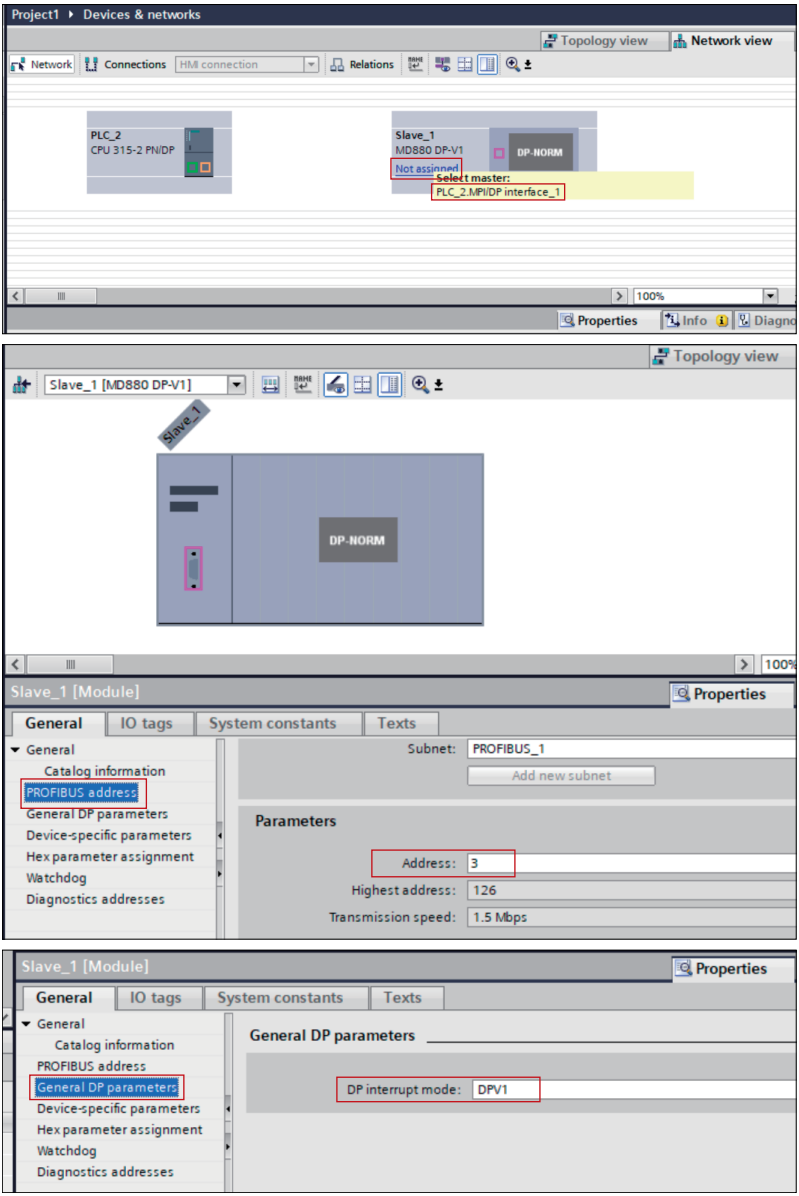
Figure 12-2

c. After the GSD file is installed, the MD880 appears under the "Other field devices" item in the hardware catalog, as shown in the following figure.

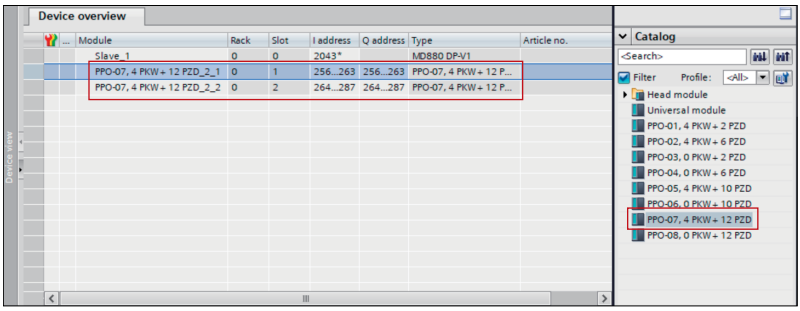


2. After installing the GSD file, drag the MD880-DP-V0 or MD880-DP-V1 device into the "Network view" window. In this example, the MD880-DP-V1 device is selected. On the

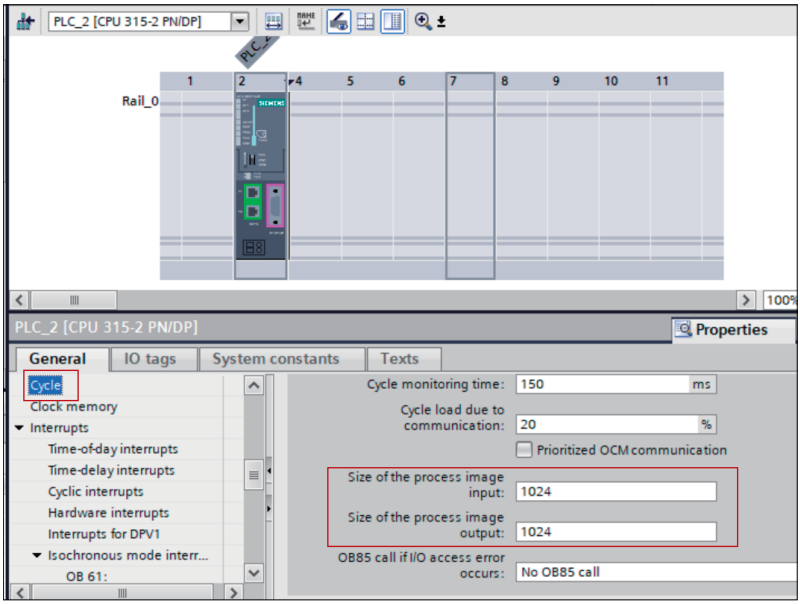
MD880 slave, click "Not assigned" and select the corresponding PROFIBUS network. Select the slave and navigate to "Properties" > "General" to set the slave address. The slave address must match the slave ID set in parameter n16-02.



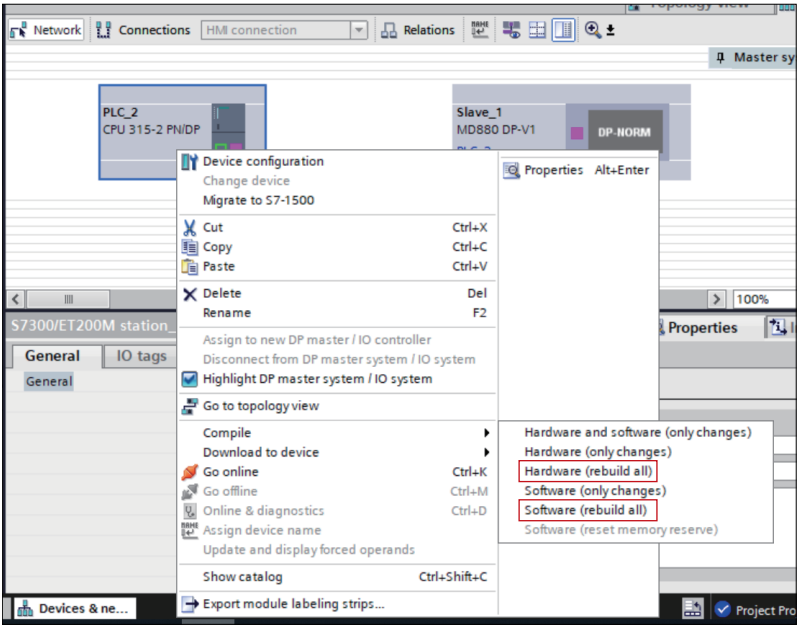
3. Set the PPO data type. In this example, the 4PKW+12PZD type is selected. Drag it into the "Device overview" window and set the I address and Q address.

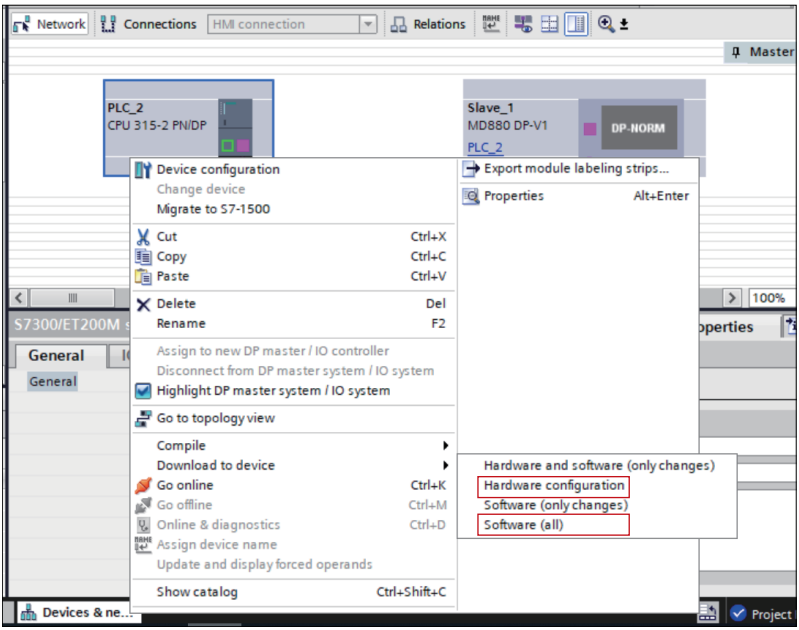


It is critical that the I address and Q address mapping ranges of the MD880 drive slave do not exceed the sizes of process image input and output of the PLC master. Otherwise, the process data of the MD880 drive may fail to be mapped to the configured I and Q addresses.



4. After completing the preceding settings, compile the PLC software and hardware (using the "rebuild all" option), and then download the software and hardware, as shown in the following figure.


Network
Connections
HMI connection
Relations
Master sy
PLC_2
CPU 315-2 PN/DP
Slave_1
MD880 DP-V1
DP-NORM
Device configuration
Change device
Migrate to S7-1500
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename F2
Assign to new DP master / IO controller
Disconnect from DP master system / IO system
Highlight DP master system / IO system
Go to topology view
Compile
Download to device
Go online Ctrl+K
Go offline Ctrl+M
Online & diagnostics Ctrl+D
Assign device name
Update and display forced operands
Show catalog Ctrl+Shift+C
Export module labeling strips...
Properties Alt+Enter
Hardware and software (only changes)
Hardware (only changes)
Hardware (rebuild all)
Software (only changes)
Software (rebuild all)
Software (reset memory reserve)
100%
S7300/ET200M station
Properties
General
IO tags
Devices & ne...
Project Pro


Network
Connections
HMI connection
Relations
Master
PLC_2
CPU 315-2 PN/DP
Slave_1
MD880 DP-V1
PLC_2
DP-NORM
Device configuration
Change device
Migrate to S7-1500
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename F2
Assign to new DP master / IO controller
Disconnect from DP master system / IO system
Highlight DP master system / IO system
Go to topology view
Compile
Download to device
Go online Ctrl+K
Go offline Ctrl+M
Online & diagnostics Ctrl+D
Assign device name
Update and display forced operands
Show catalog Ctrl+Shift+C
Export module labeling strips...
Properties Alt+Enter
Hardware and software (only changes)
Hardware configuration
Software (only changes)
Software (all)
100%
General
Devices & ne...
Project

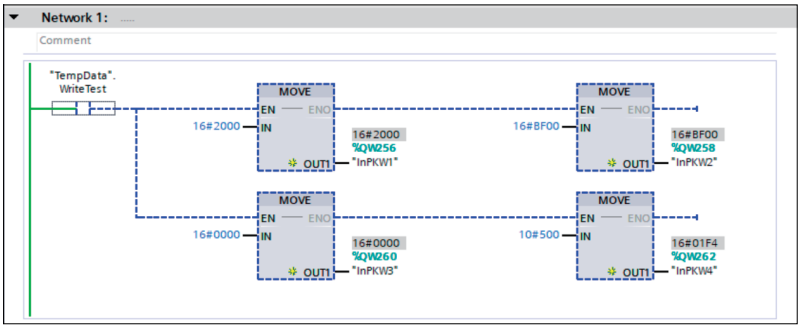
Reading/Writing MD880 drive parameters with PKW

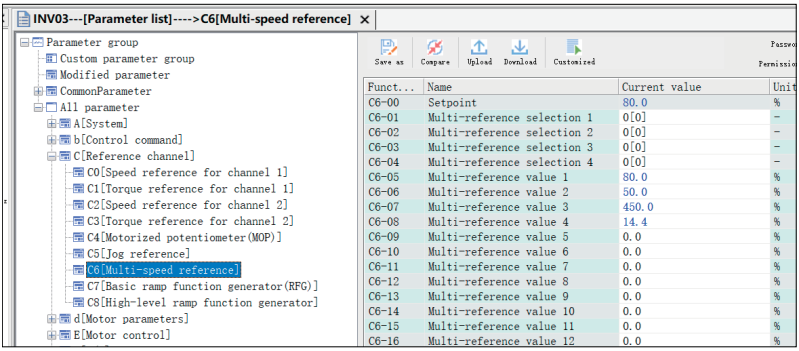
PKW data is mainly used for the master to read/write individual parameters of the drive. This example demonstrates reading and writing parameter C6-05.

1. Use InoDriveStudio to connect to the HCU and view the address and attributes of parameter C6-05. As shown on the page, parameter C6-05 has an address of 191 and the modification mode is "Free modification", indicating it is readable and writable.

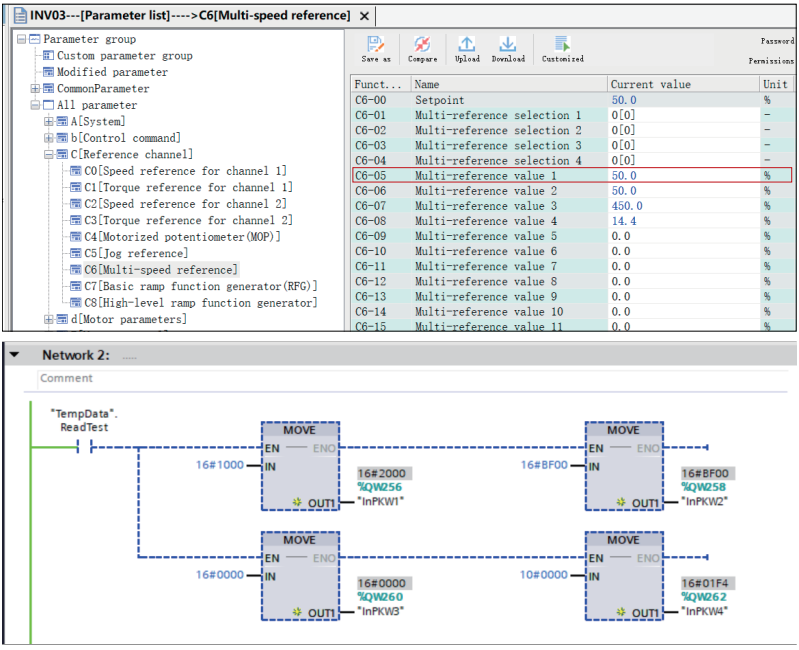
Funct...	Name	Current value	Unit	Min	Max	Default
C6-00	Setpoint	90.0	%	-	-	-
C6-01	Multi-reference selection 1	0[0]	-	0	10	0
C6-02	Multi-reference selection 2	0[0]	-	0	10	0
C6-03	Multi-reference selection 3	0[0]	-	0	10	0
C6-04	Multi-reference selection 4	0[0]	-	0	10	0
C6-05	Multi-reference value 1	90.0	%	-600.0	600.0	10.0
C6-06	Multi-reference value 2	50.0	%	-600.0	600.0	10.0
C6-07	Multi-reference value 3	450.0	%	-600.0	600.0	0.0
C6-08	Multi-reference value 4	14.4	%	-600.0	600.0	0.0
C6-09	Multi-reference value 5	0.0	%	-600.0	600.0	0.0
C6-10	Multi-reference value 6	0.0	%	-600.0	600.0	0.0
C6-11	Multi-reference value 7	0.0	%	-600.0	600.0	0.0
C6-12	Multi-reference value 8	0.0	%	-600.0	600.0	0.0
C6-13	Multi-reference value 9	0.0	%	-600.0	600.0	0.0
C6-14	Multi-reference value 10	0.0	%	-600.0	600.0	0.0
C6-15	Multi-reference value 11	0.0	%	-600.0	600.0	0.0
C6-16	Multi-reference value 12	0.0	%	-600.0	600.0	0.0
C6-17	Multi-reference value 13	0.00	%	-300.00	300.00	0.00
C6-18	Multi-reference value 14	0.00	%	-300.00	300.00	0.00
C6-19	Multi-reference value 15	0.00	%	-300.00	300.00	0.00
C6-20	Multi-reference value 16	0.00	%	-300.00	300.00	0.00

2. The PLC sends a write command to the HCU for parameter C6-05 according to the PKW data format described in *"6.2.2 PKW Data" on page 45*, writing the decimal value 500, which corresponds to 50.0% in InoDriveStudio. The example PLC program is shown in the following figure.





3. Change the C6-05 value to 80%, corresponding to the decimal value 800. The PLC sends a read command to the HCU for parameter C6-05 according to the PKW data format described in *"6.2.2 PKW Data" on page 45*. The example PLC program is shown in the following figure.



12.2 PROFINET IO

Networking

In this example, the PLC used is a Siemens S7-300, model 6AG1 315-2EH13-2AB0. The PLC programming software is Siemens TIA Portal V15. The MD880 drive commissioning software

is InoDriveStudio, which can be obtained from the Inovance official website at https://www.inovance.com.

Connect devices according to the following figure to complete networking. Set the SOP-20 to the USB relay mode.

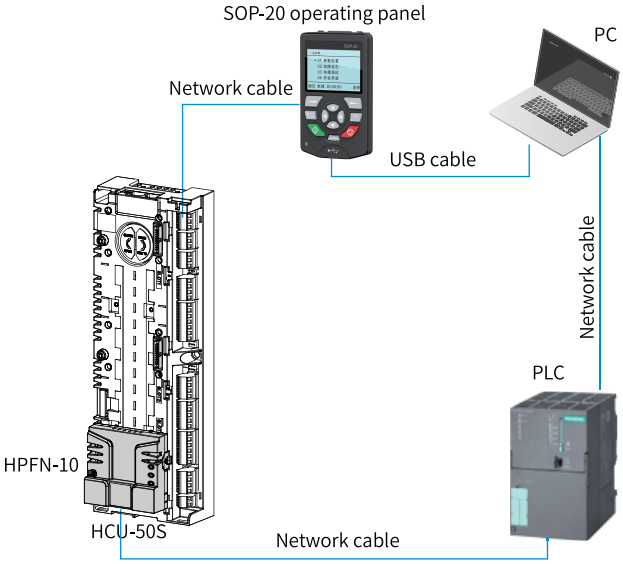
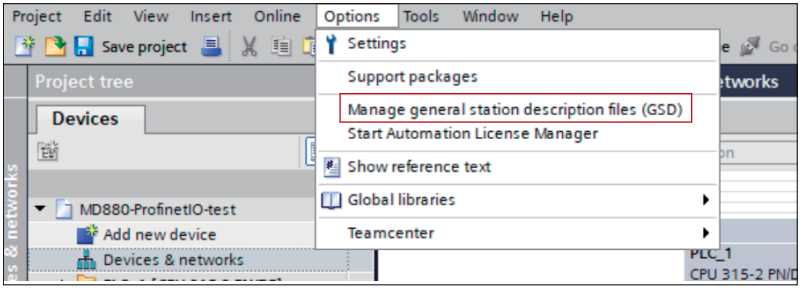


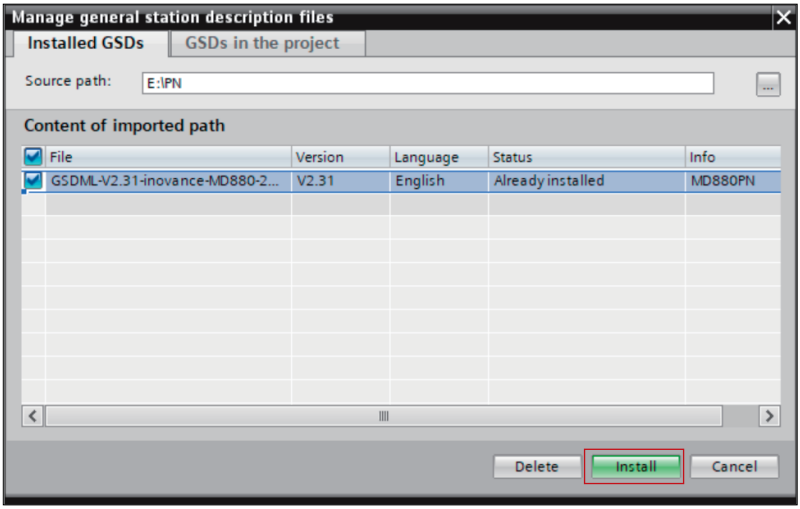
Figure 12-3 PROFINET IO networking of MD880 with Siemens PLC

Configuration in TIA Portal V15

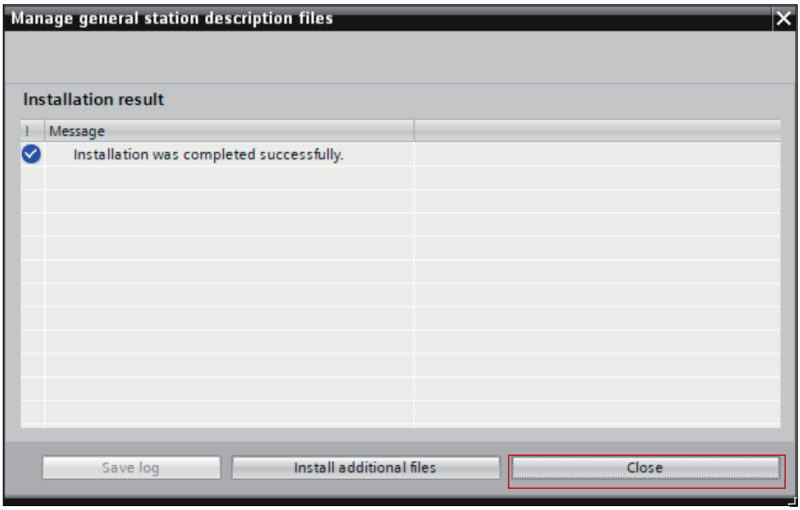
1. Install the GSD file for the MD880 drive in TIA Portal V15. You may obtain the GSD file from Inovance or its agent.



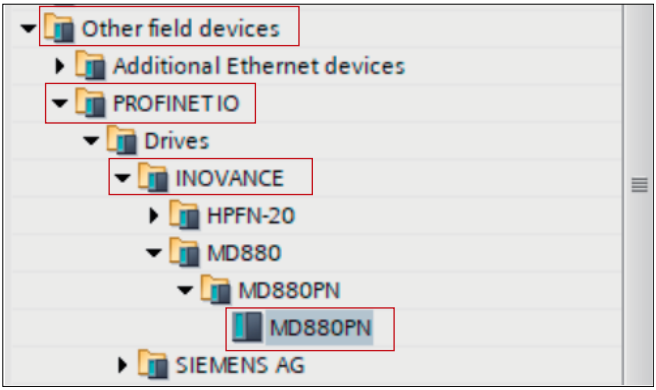
a. Input the path where the GSD file is located and click "Install", as shown in the following figure.



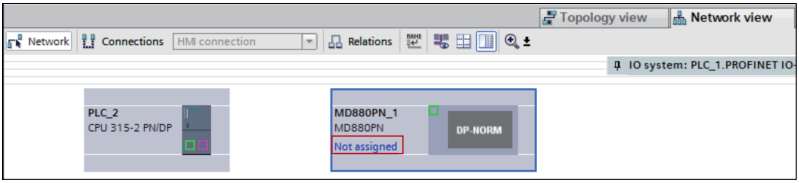
b. Once the interface below appears, the installation is completed. Click "Close".



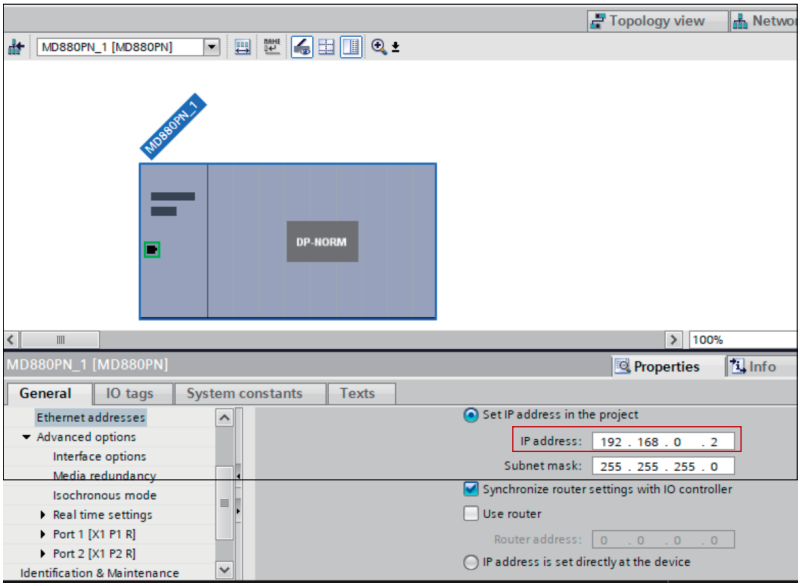
c. After the GSD file is installed, the MD880 appears under the "Other field devices" item in the hardware catalog, as shown in the following figure.



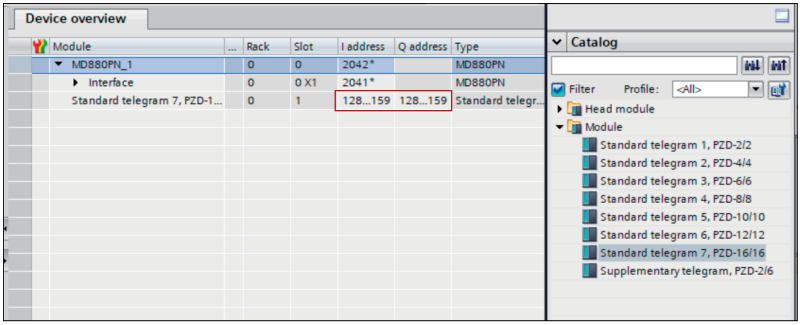
2. After installing the GSD file, drag the MD880PN device into the "Network view" window. On the MD880 slave, click "Not assigned" and select the corresponding PROFINET network, as shown in the following figure.



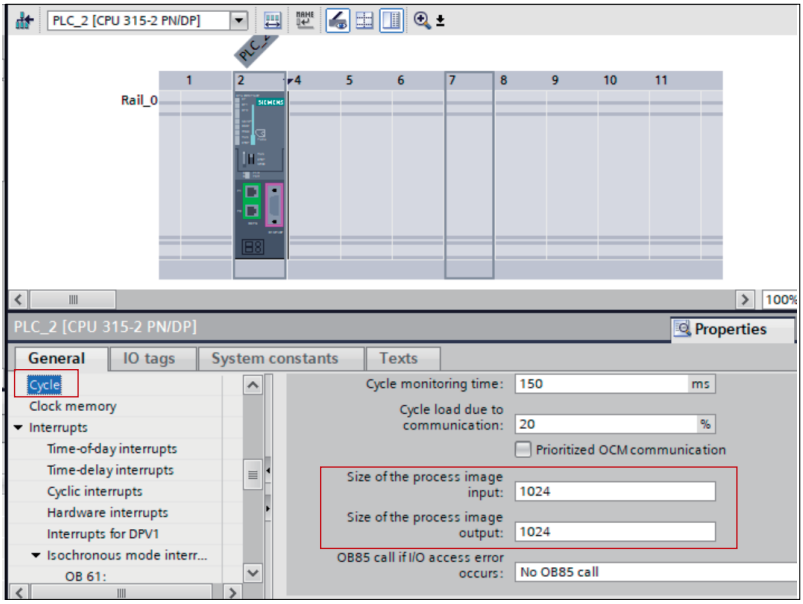
Select the slave and navigate to "Properties" > "General" > "PROFINET interface [X1]" > "Ethernet addresses" to set the IP address. The PLC master IP address and the slave IP address must be on the same network, as shown in the following figure.



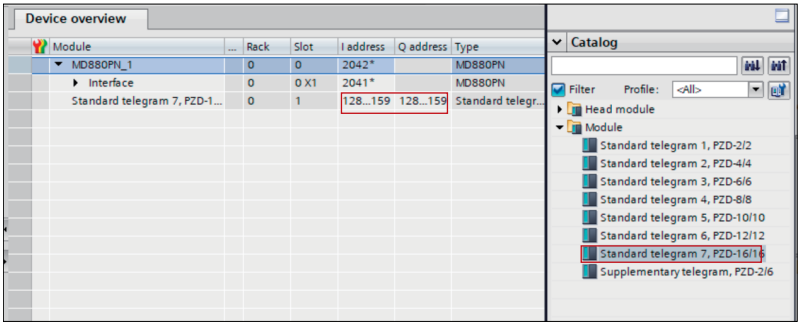
3. Select the slave and click the "Device view" tab. In the "Hardware catalog" pane, click "Module", and double-click the data type to be configured to the slave. In this example, the selected data type is Standard telegram 7.



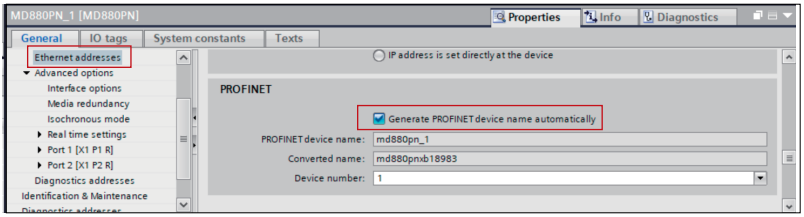
It is critical that the I address and Q address mapping ranges of the MD880 drive slave do not exceed the sizes of process image input and output of the PLC master. Otherwise, the process data of the MD880 drive may fail to be mapped to the configured I and Q addresses.



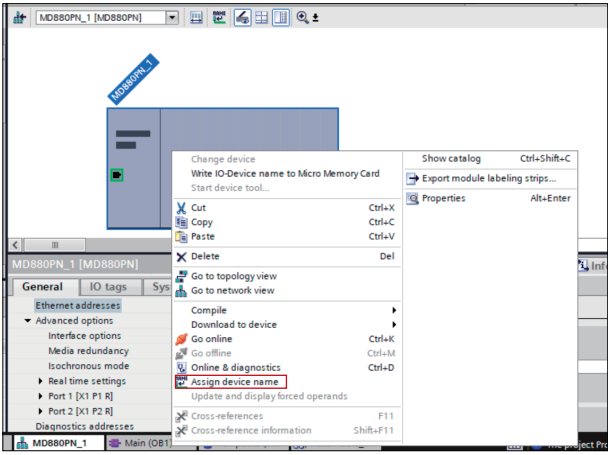
4. After completing the preceding settings, compile the PLC software and hardware (using the "rebuild all" option), and then download the software and hardware, as shown in the following figure.

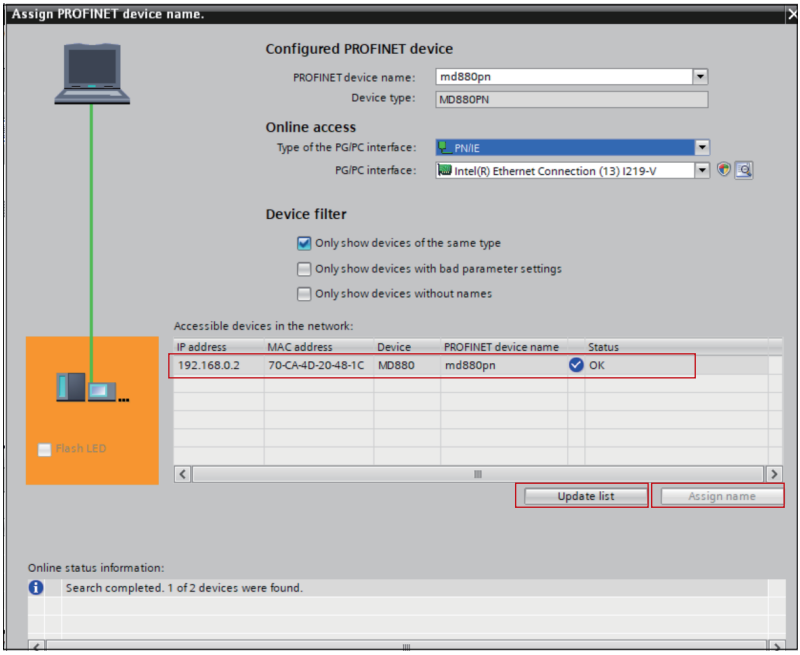


5. In the MD880 device view, right-click the MD880 slave to assign a name. You can use the system-generated name. If you prefer to set a custom name for the MD880 slave, navigate to "PROFINET", deselect "Generate PROFINET device name automatically", and enter a name in the "PROFINET device name" field, as shown in the following figure.

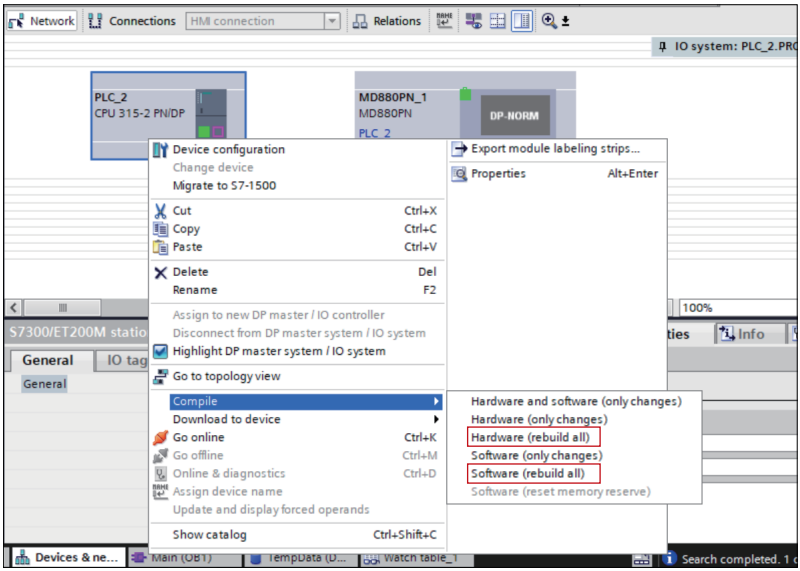


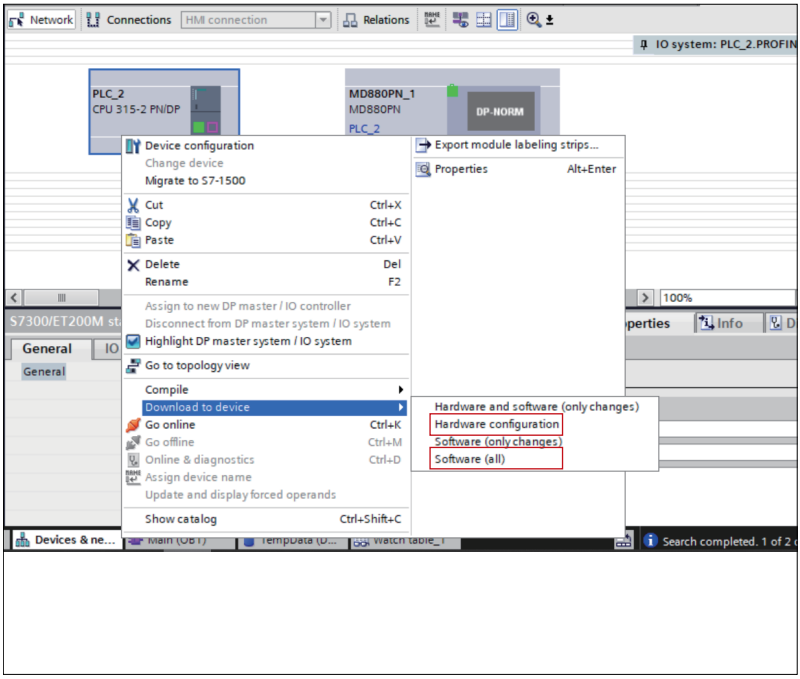
After you click "Assign device name", the following screen appears. Click "Update list". After a short wait, devices on the network are listed. Select the slave to be named. Each slave has its unique MAC address and you can differentiate the slaves based on their MAC addresses. For an HPFN-10 module, you can find the MAC address on its housing or in parameter group n17. Click "Assign device name" and wait for the operation to complete.





6. After completing the preceding settings, compile the PLC software and hardware (using the "rebuild all" option), and then download the software and hardware, as shown in the following figure.

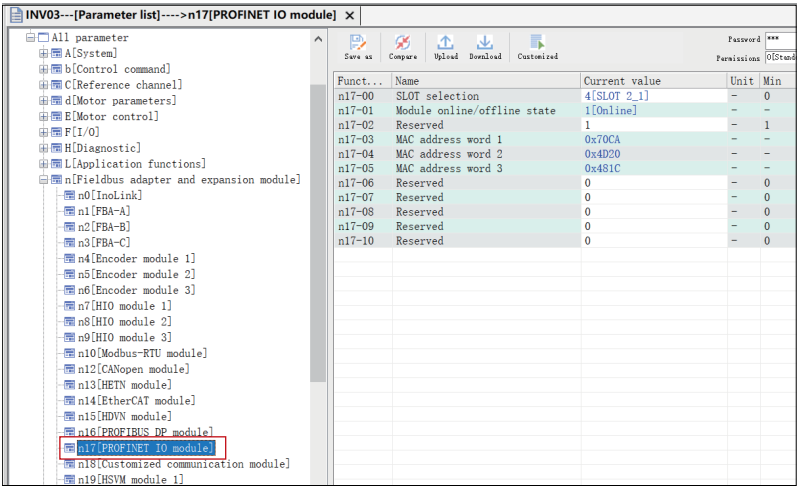




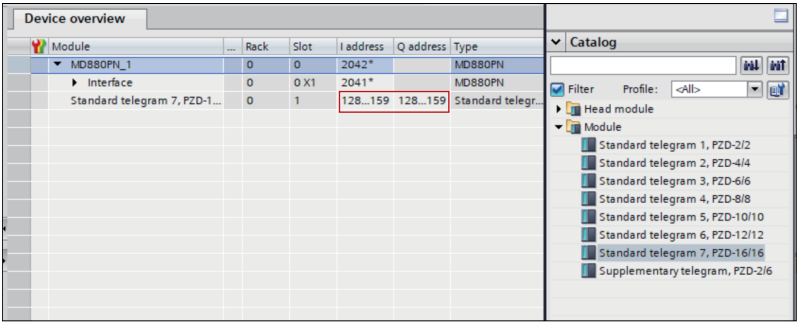
7. Switch back to the "Network view" and add more slaves as needed by repeating the preceding steps. To add a slave with the same configuration as an existing one, select and copy the existing slave and just modify the IP address and device name (note that the device name must not be duplicate). The slave will save the assigned name, and the master identifies each slave based on the device name. (The MAC address is not intuitive to use. The process of assigning a device name is actually binding the device name with the MAC address.)

Process data configuration

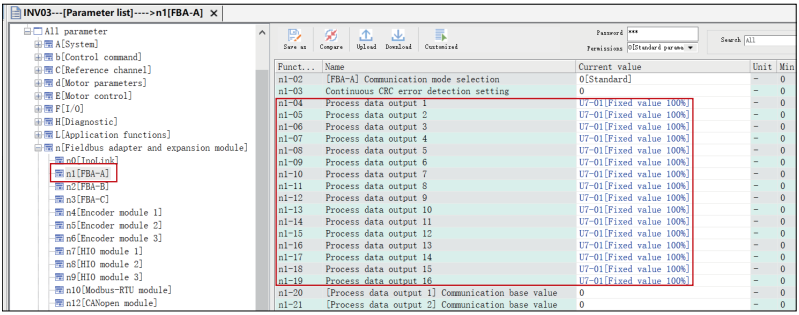
1. After downloading the PLC configuration, use InoDriveStudio to connect to the MD880 drive. See the following figure for settings of parameters in the n17 group used in this example.



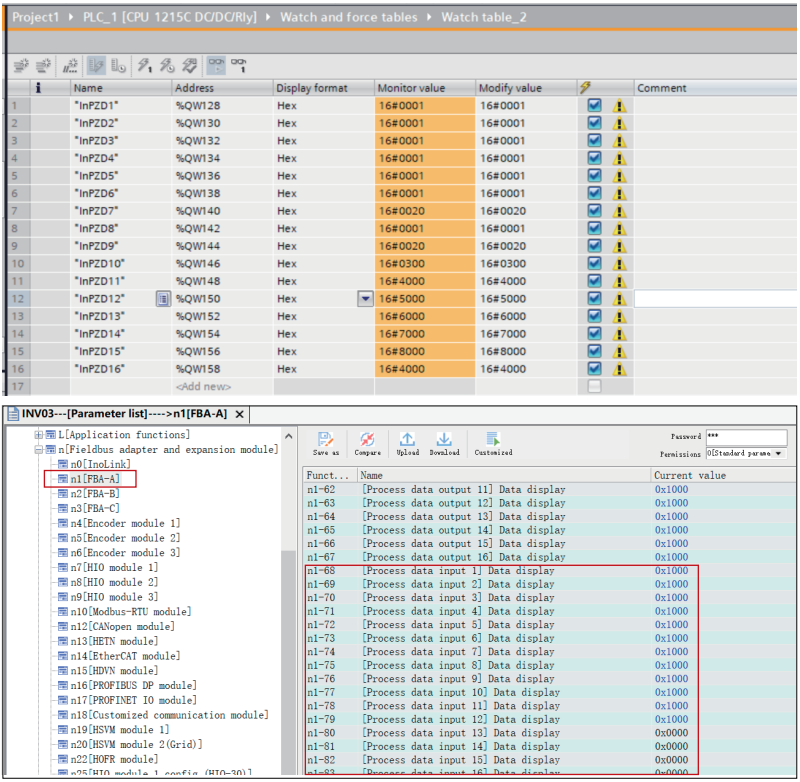
2. After loading the PLC configuration to the PLC, verify the transmission of process data, that is, check whether data can be transmitted between the MD880 and the PLC. In TIA Portal V15, the process data of the MD880 has been mapped to addresses in the PLC. The data type is Standard telegram 7, as shown in the following figure. PZD1 to PZD16 are mapped to the PLC's process image input/output addresses 128 to 159, occupying 16 words (32 bytes).

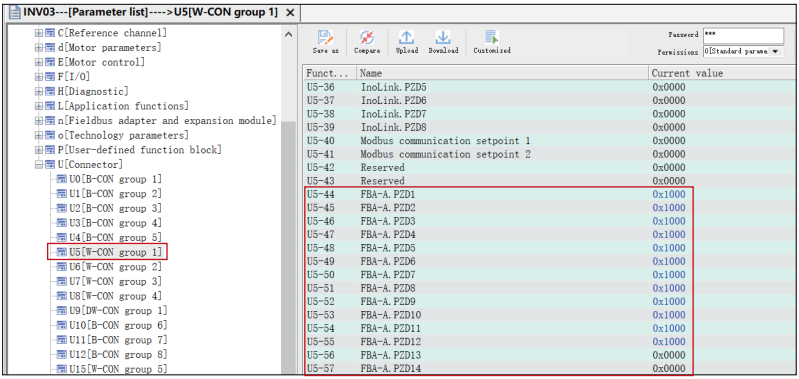


3. Configure the data to be written from the MD880 drive to the PLC using parameters n1-04 to n1-15. A sent fixed value of 100% corresponds to 16#1000, as shown in the following figure.



4. After configuring parameters n1-04 to n1-15, observe the data sent by the MD880 drive in TIA Portal V15. The corresponding addresses are IW128 to IW158. The PLC has received the data sent by the MD880 drive, indicating normal data transmission from the MD880.
5. You can view the raw data written from the PLC to the MD880 drive using parameters n1-68 to n1-83. For input process data that has been converted by the communication base value, you can view them using U5-44 to U5-59. As shown in the following figure, the MD880 drive can successfully receive the data written by the PLC.





Replacing a faulty device

The HPFN-10 card supports direct replacement in case of a slave node failure (only for card failure; if the AC drive fails, replace the AC drive, which is not covered in this guide) without needing to reconfigure the device.

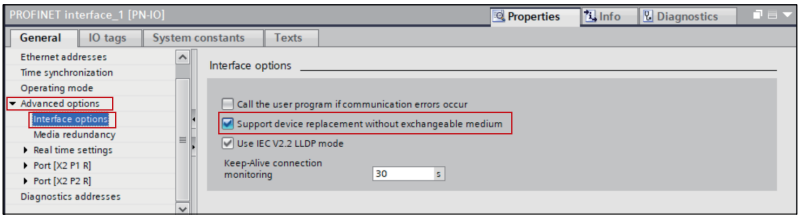
Prerequisites for directly replacing the HPFN-10 card:

- Both the replacement and replaced devices must be HPFN-10.
- The replacement HPFN-10 card must never have been assigned a device name.
- The PLC configuration network must have a defined topology.
- The "Support device replacement without exchangeable medium" option must be enabled in PLC configuration.

To enable direct replacement of the HPFN-10, configure the following settings (settings differ between STEP 7 and TIA Portal).

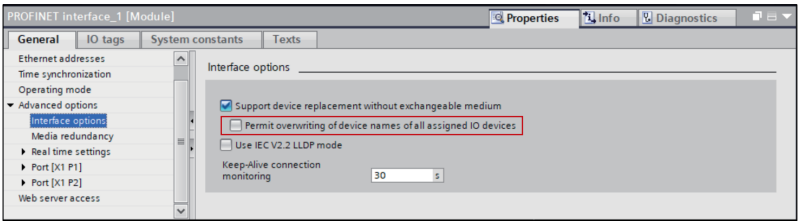
- Select the "Support device replacement without exchangeable medium" option and set the topology in PORTAL.

  1. On the hardware configuration page of PORTAL, select the PROFINET interface of the master. Choose "Properties" > "General" > "Advanced options" > "Interface options", and select "Support device replacement without exchangeable medium", as shown in the following figure.

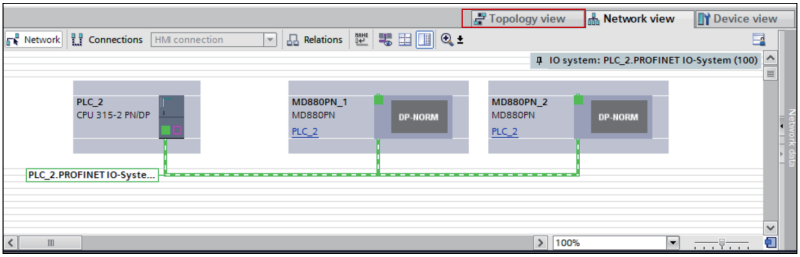


  2. For the S7-1200 or S7-1500 PLC, a sub-option "Permit overwriting of device names of all assigned IO devices" is available. If this sub-option is also selected, the second

prerequisite (the replacement HPFN-10 card must never have been assigned a device name) for direct replacement can be ignored, as shown in the following figure.



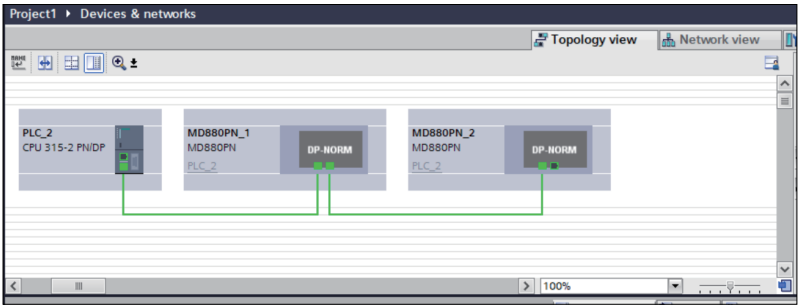
3. Switch to the "Topology view", as shown in the following figure.



4. In the topology view, click and drag an interface to the interface of another device which is directly connected with the first interface, and release the mouse button.

Note

The topology must match the actual network connections of the devices. For example, if the PLC connects from P1 to P2 of Slave 1, then from P1 to the next slave in reality, the topology must reflect this. will cause the replacement function to fail or even result in communication errors.



12.3 Modbus RTU

Networking

In this example, the PLC used is an Inovance AM600-CPU1608TN. The PLC programming software is InoProShop from Inovance. The MD880 drive commissioning software is InoDriveStudio. You can obtain the software from the Inovance official website at https://www.inovance.com.

Connect devices according to the following figure to complete networking.

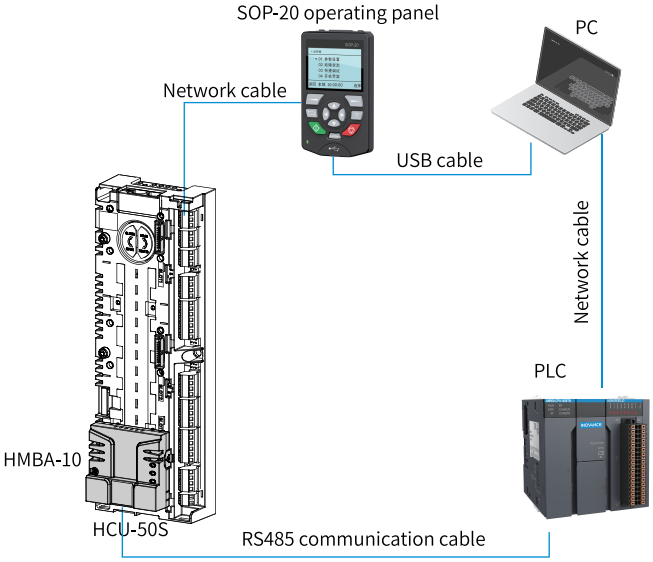


Figure 12-4 Modbus RTU networking of MD880 with AM600

DB9 port of AM600

The RS485 port (CN1) supports two RS485 channels. The two RS485 channels share the same DB9 port, as shown in the following figure.

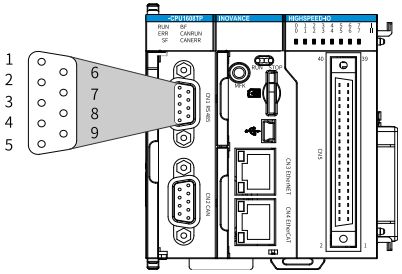


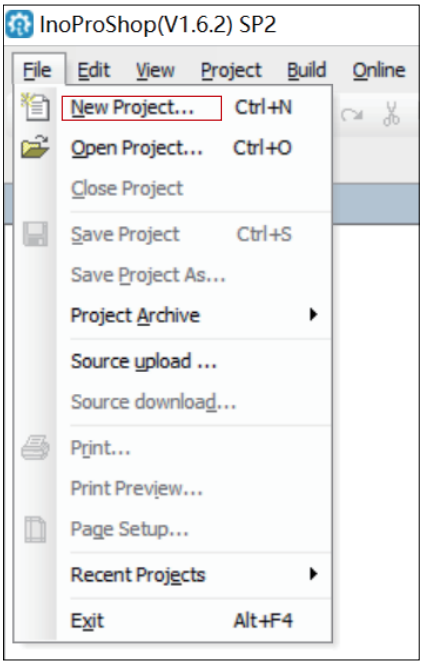
Figure 12-5 RS485 terminal on the CPU module

Table 12–1 DB9 pins

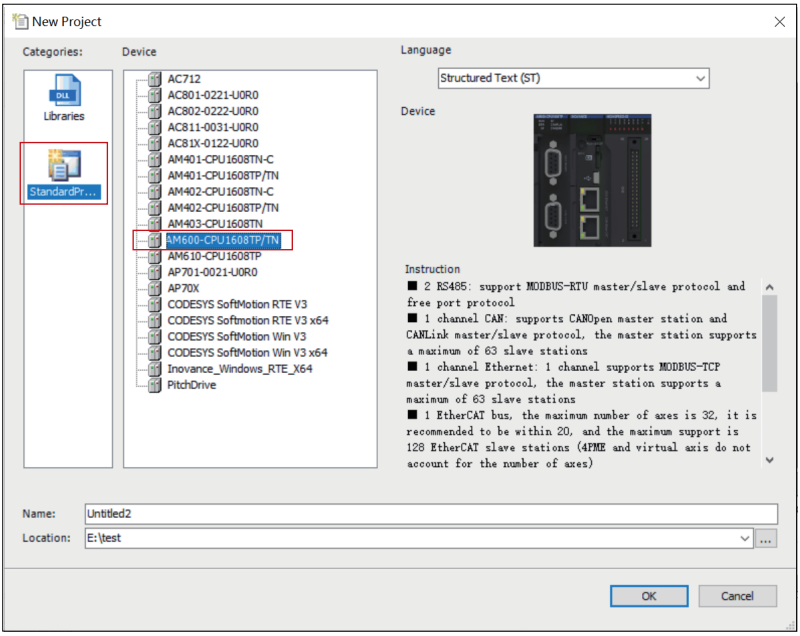
Channel	Pin	Assignment	Function
COM0	1	RS485–	COM0 RS485 negative signal
	2	RS485+	COM0 RS485 positive signal
	5	GND	COM0 power ground
COM1	6	RS485–	COM1 RS485 negative signal
	9	RS485+	COM1 RS485 positive signal
	3	GND	COM1 power ground

PLC configuration

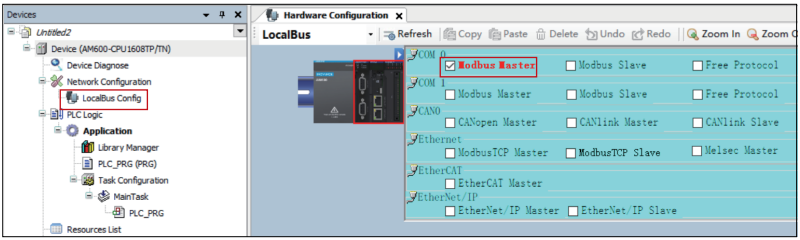
1. Create a project.



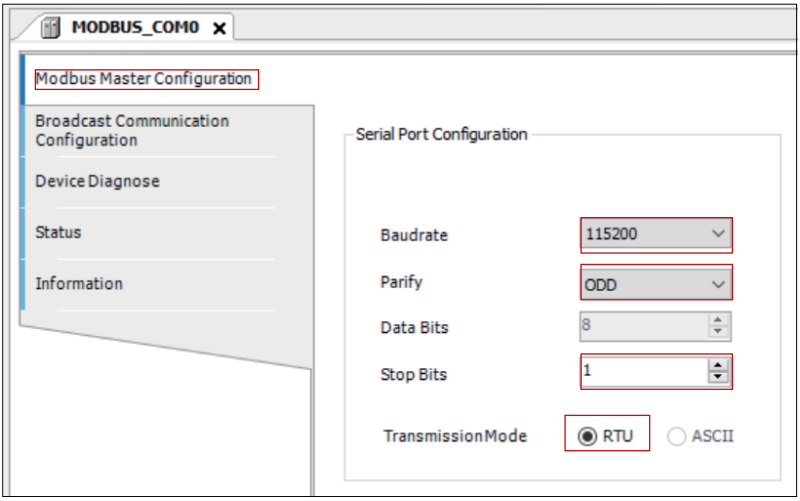
2. Select the PLC type.



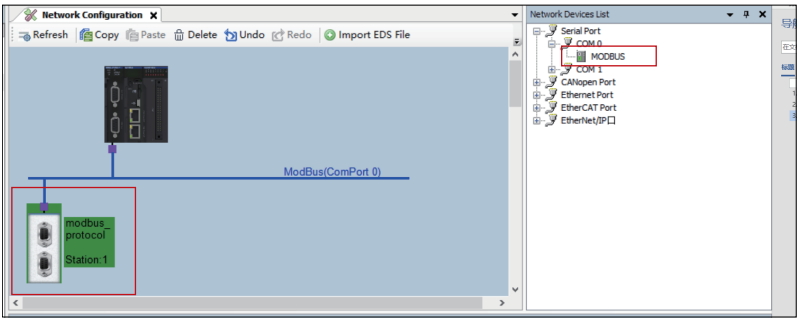
3. Set the AM600 as the Modbus master.



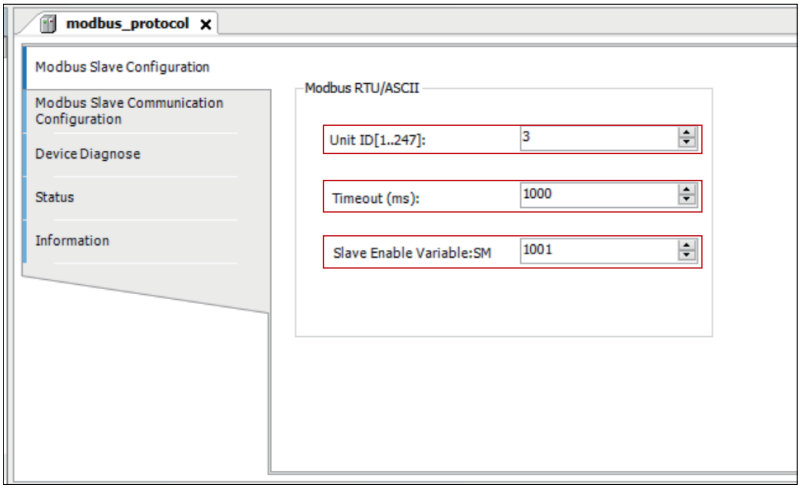
4. Set parameters such as the baud rate.



5. Drag the MODBUS device under the COM 0 port to the "Network Configuration" tab. Choosing between the COM 0 and COM 1 ports depends on the DB9 hardware wiring of the AM600. Selecting the wrong port will cause communication failure.



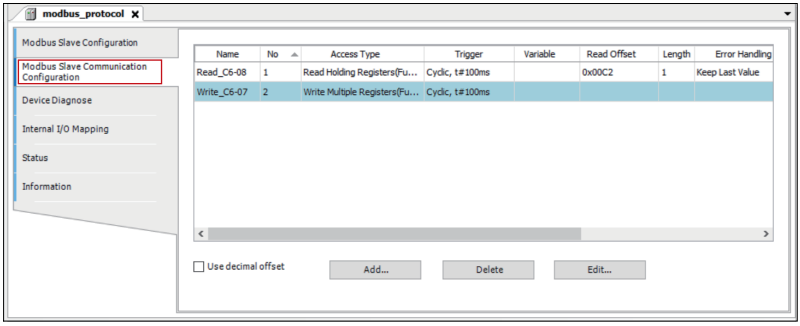
6. Configure the slave unit ID, which must match parameter n10-04 of the MD880 drive.



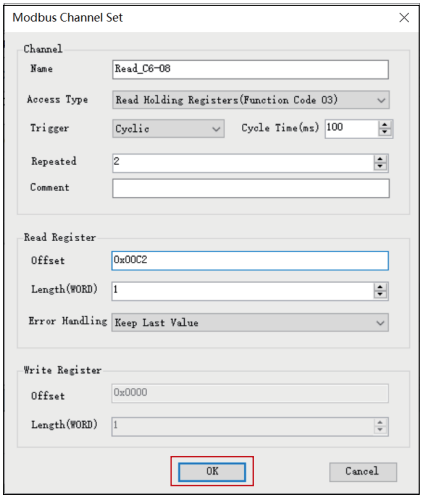
7. Configure the parameters to be read/written. This example explains reading a single register and writing a single register.

Note

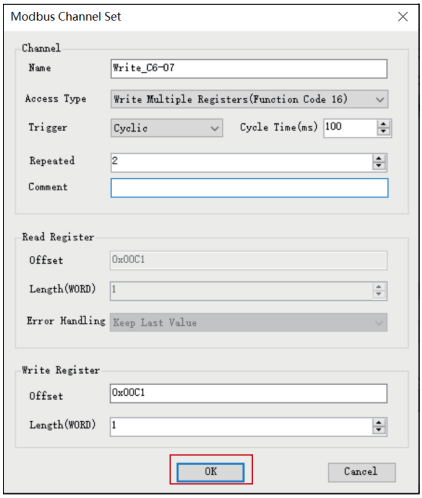
For reading or writing multiple registers, the parameter addresses must be consecutively readable or writable.



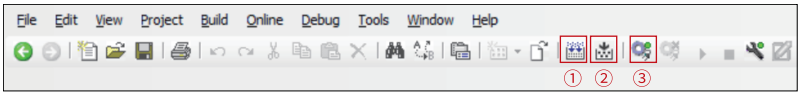
Reading a single register:



Writing a single register:



8. After all settings are complete, download the settings to the PLC.

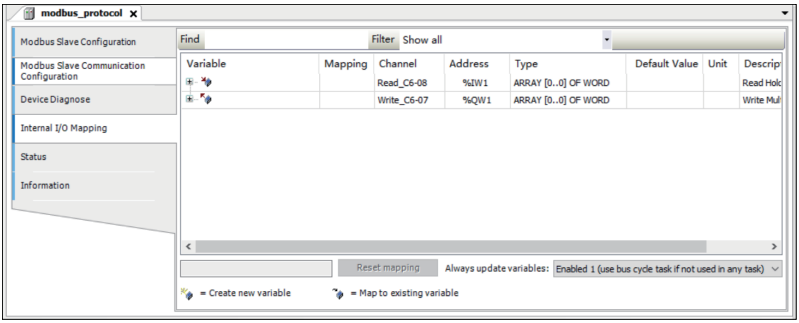


Note

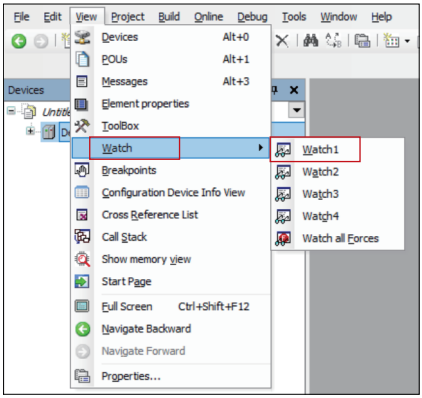
- ①: Check the program.
- ②: Compile.
- ③: Download.

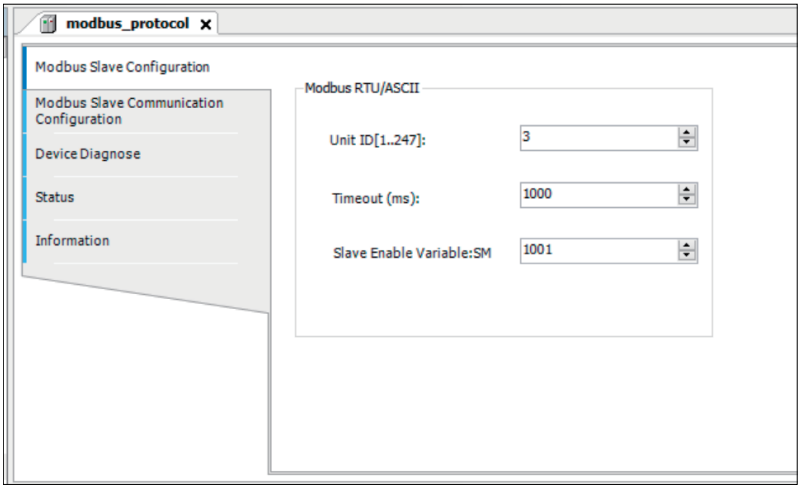
Data read/write check

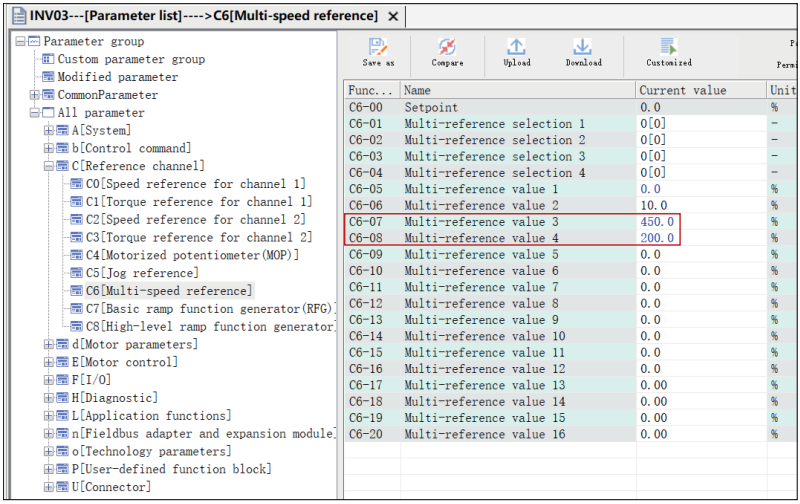
1. IW1 and QW1 are the I/O address mappings in the PLC for the data to be read/written through Modbus RTU.



2. After configuration, check if the parameters can be read/written normally. Monitor addresses IW1 and QW1 in the PLC. Check in the background if the data written by the PLC is correct. The value of the slave enable variable SM1001 must be forced to 1.







12.4 CANopen

Networking

In this example, the PLC used is an Inovance AM600-CPU1608TN. The PLC programming software is InoProShop from Inovance. The MD880 drive commissioning software is InoDriveStudio. You can obtain the software from the Inovance official website at https://www.inovance.com.

Connect devices according to the following figure to complete networking. Set the SOP-20 to the USB relay mode.

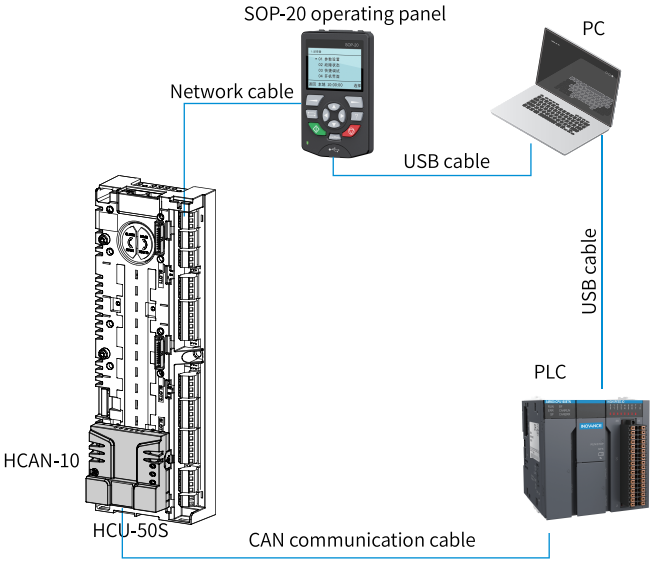


Figure 12-6 CANopen networking of MD880 with AM600

DB9 port of AM600

The CN2 of the AM600-CPU1608TN is the CANopen communication terminal, using a DB9 interface for data transmission, as shown in the following figure.

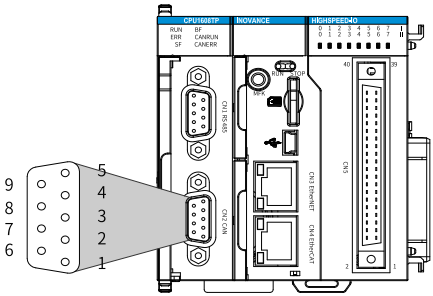


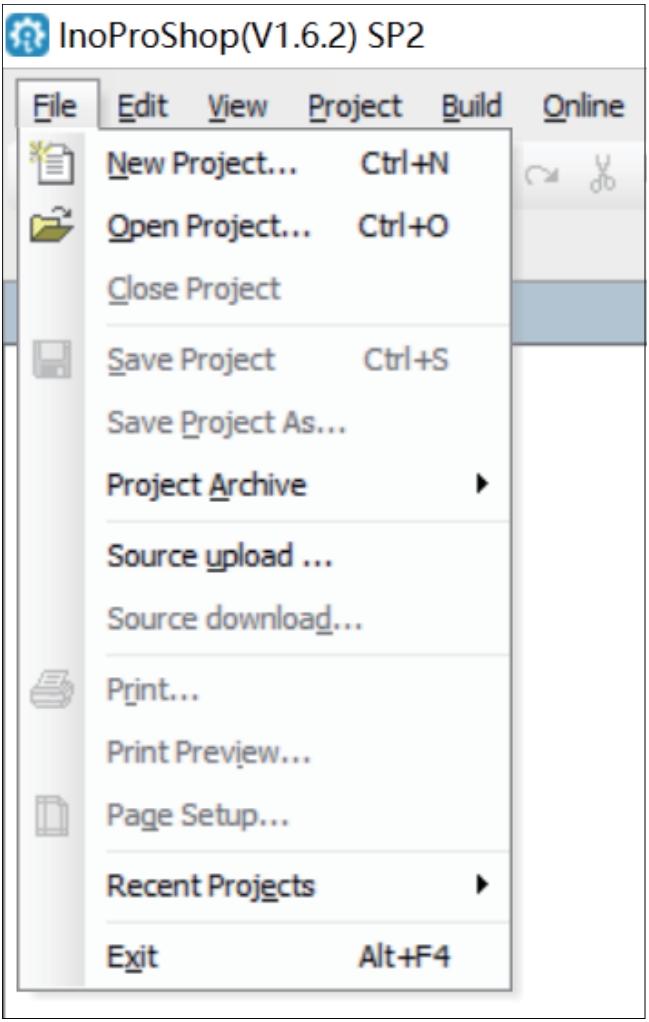
Figure 12-7 CANopen terminal on the CPU module

Table 12–2 CANopen DB9 pins

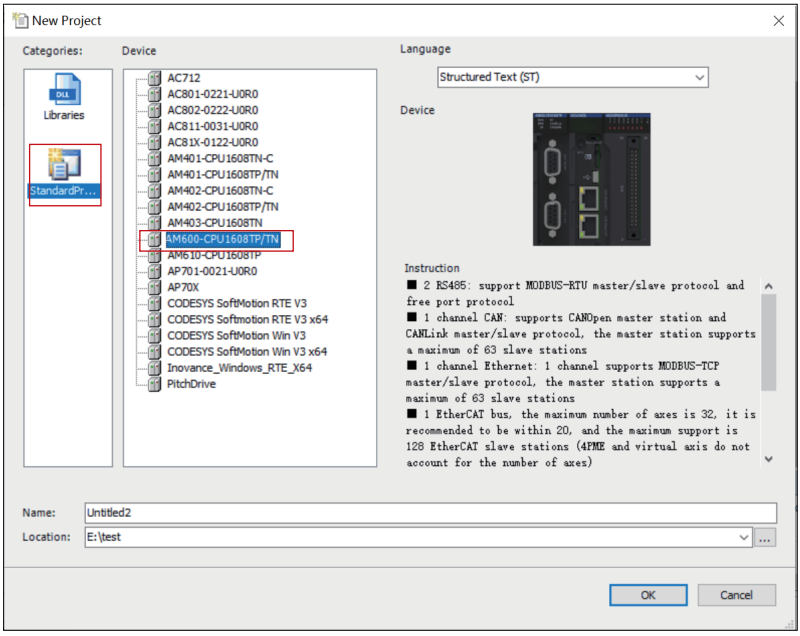
Pin	Signal Assignment
2	CANL
7	CANH
3	GND

PLC configuration

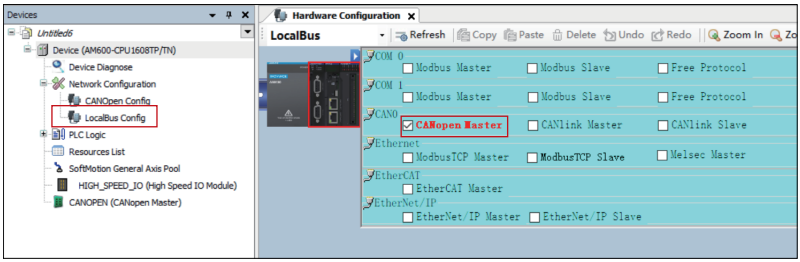
1. Create a project.



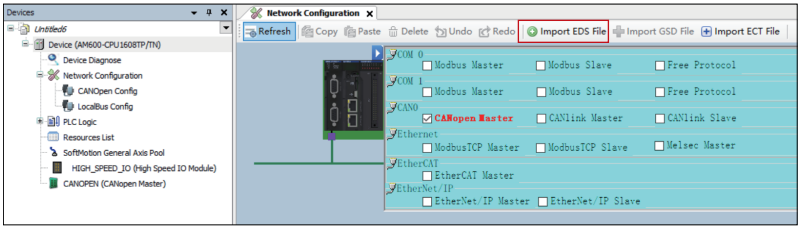
2. Select the PLC type.



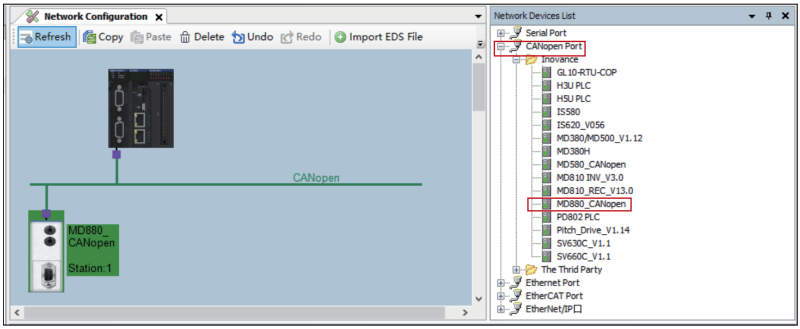
3. Set the AM600 as the CANopen master.



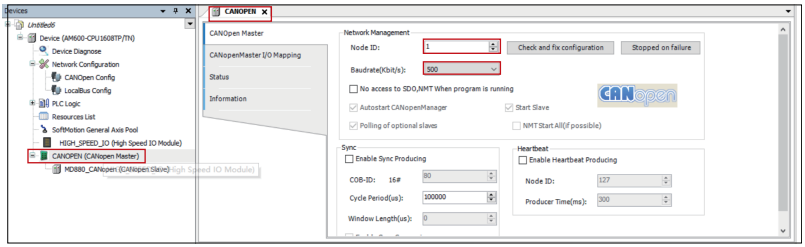
4. CANopen requires installing the EDS file for the MD880 drive. You may obtain the EDS file from Inovance or its agent.



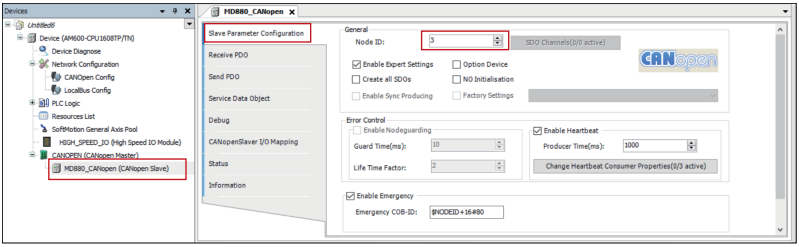
5. After the file is installed, the MD880N_CANopen device appears in the network device list. Drag the device to the "Network Configuration" tab.



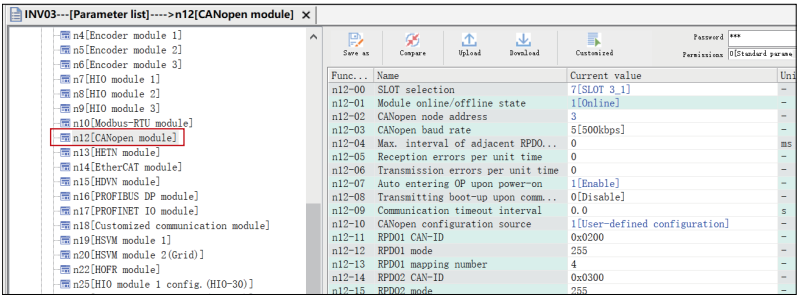
6. Configure the master parameters.

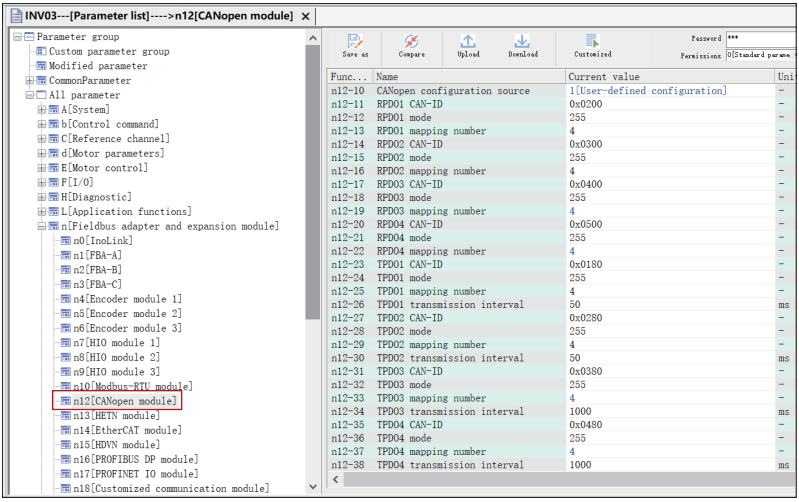


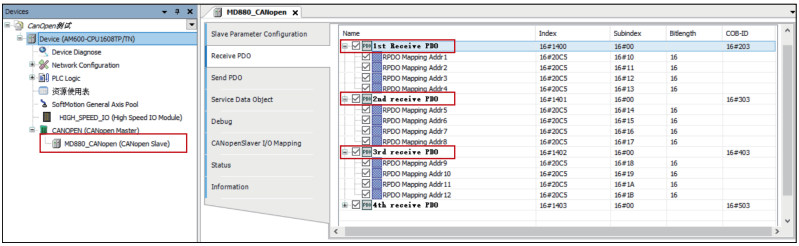
7. Configure the slave parameters. The slave node ID must not conflict with the master node ID and must match the HCAN-10 module parameters.

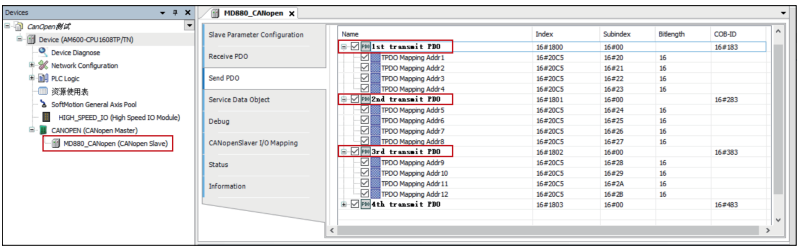


8. Configure the PDO. By default, 8 parameters are periodically written, and 8 parameters are periodically read. You can add or reduce parameters as required. The MD880 drive supports a maximum of 16 parameters for periodic read/write. In this example, 16 parameters are configured for periodic write and 16 parameters are configure for periodic read, as shown in the following figure.

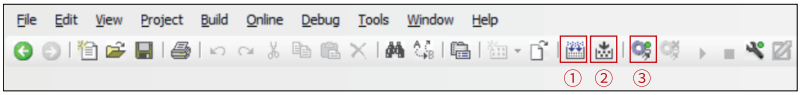








9. After all settings are complete, download the settings to the PLC.

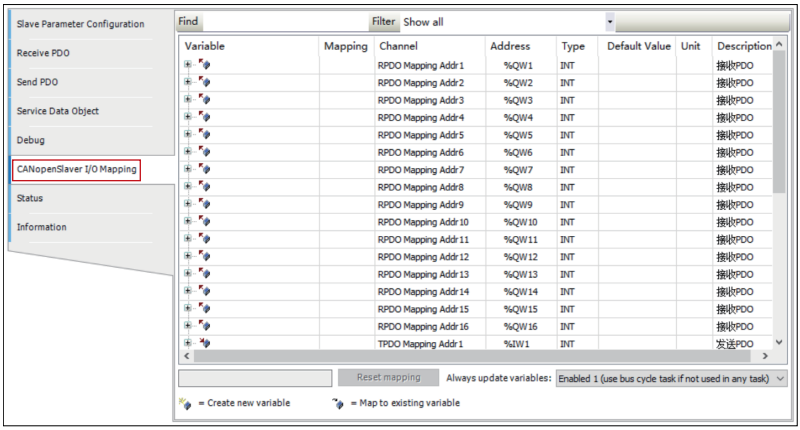


Note

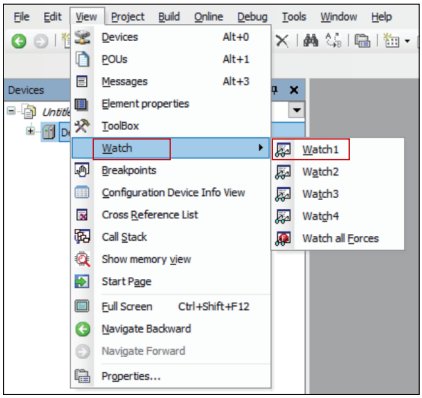
- ①: Check the program.
- ②: Compile.
- ③: Download.

Data read/write check

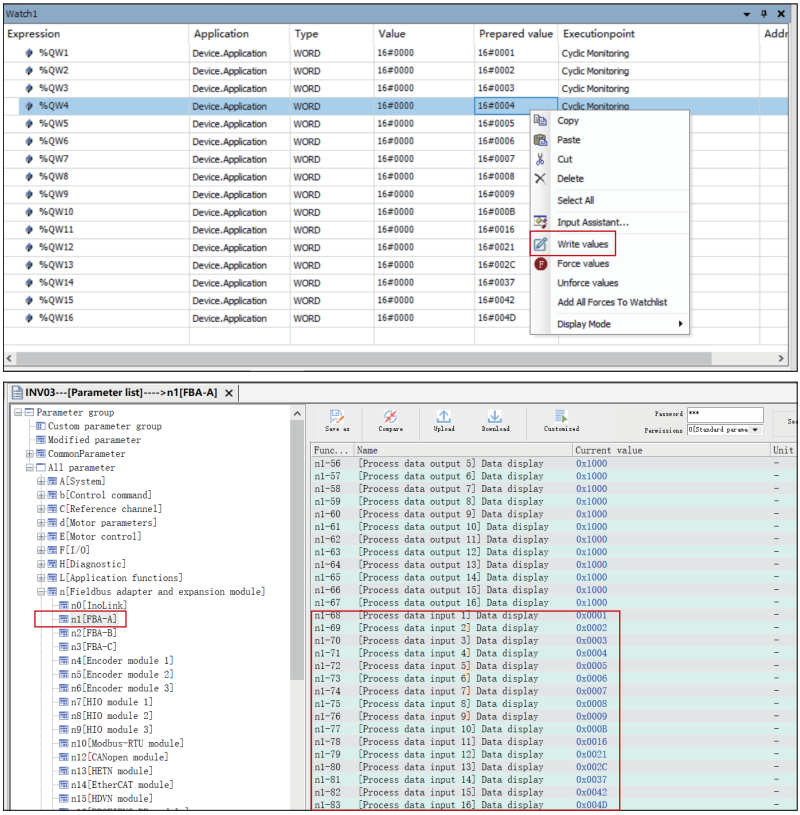
1. After loading the PLC configuration to the PLC, verify the transmission of process data, that is, check whether data can be transmitted between the MD880 and the PLC. Monitor addresses IW1 to IW16 and QW1 to QW16 in the PLC. IW1 to IW16 and QW1 to QW16 are the I/O address mappings of CANopen process data in the PLC.



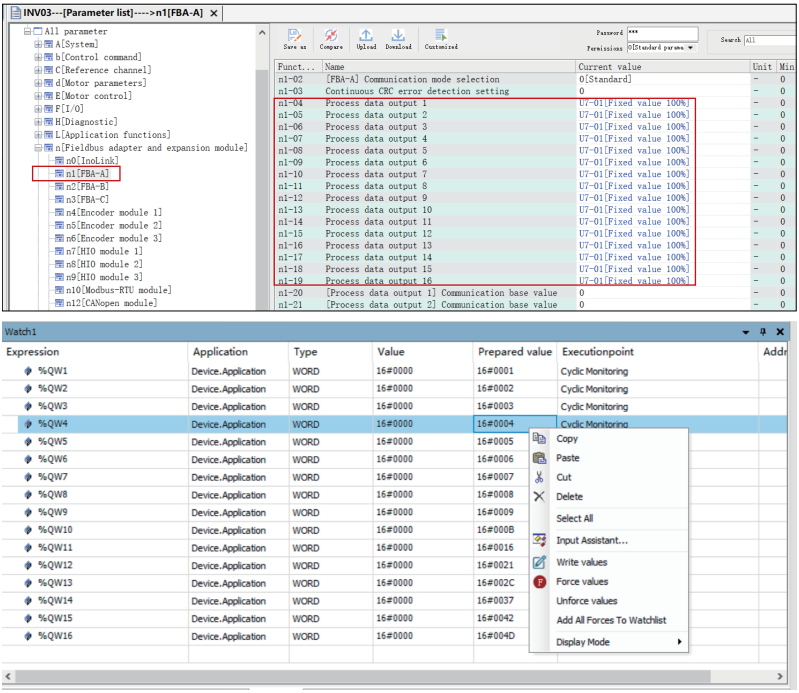
2. Add addresses IW1 to IW16 and QW1 to QW16 to the watch table.



3. Make the PLC write process data to the MD880 drive.



4. Configure the process data to be written from the MD880 to the PLC using parameters n1-04 to n1-19. An output fixed value of 100% corresponds to 16#1000.



Read/Write by SDO

This section takes reading C6-05 and writing C6-06 of the MD880 by SDO as examples.

1. Connect to the MD880 drive using USB-CAN.
2. Read C6-05, whose address is 191. Refer to the SDO data format in *"8.2.3 SDO Data" on page 74* and send the data to the MD880 drive.
3. Write to C6-06, whose address is 192. Refer to *"8.2.3 SDO Data" on page 74* and send the data to the MD880 drive.

12.5 Modbus TCP

Networking

In this example, the PLC used is an Inovance AM600-CPU1608TN. The PLC programming software is InoProShop from Inovance. The MD880 drive commissioning software is InoDriveStudio. You can obtain the software from the Inovance official website at https://www.inovance.com.

Connect devices according to the following figure to complete networking. Set the SOP-20 to the USB relay mode.

Note

The AM600-CPU1608TN has two Ethernet ports. Use the CN3 port for Modbus TCP communication.

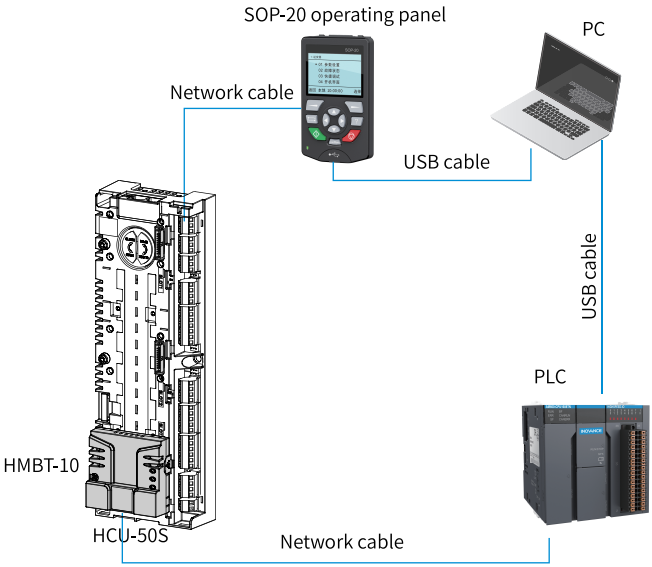
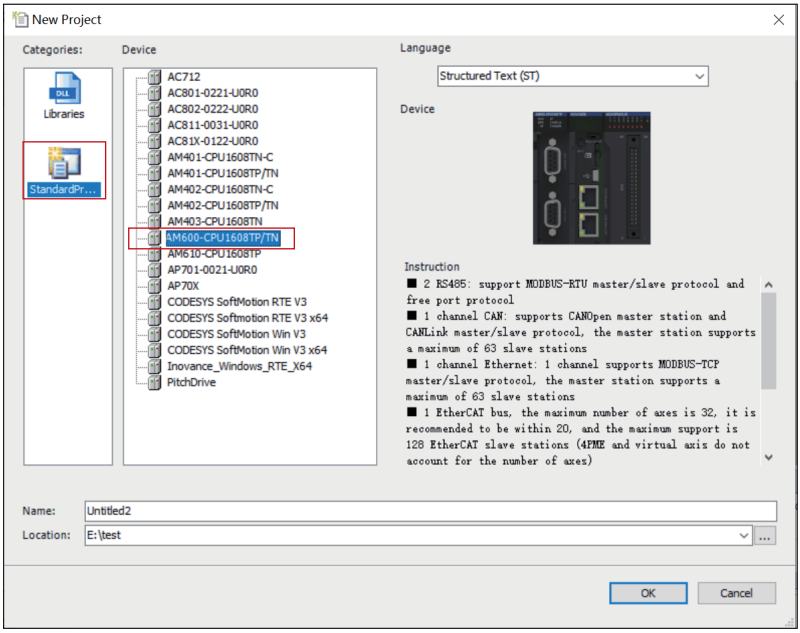


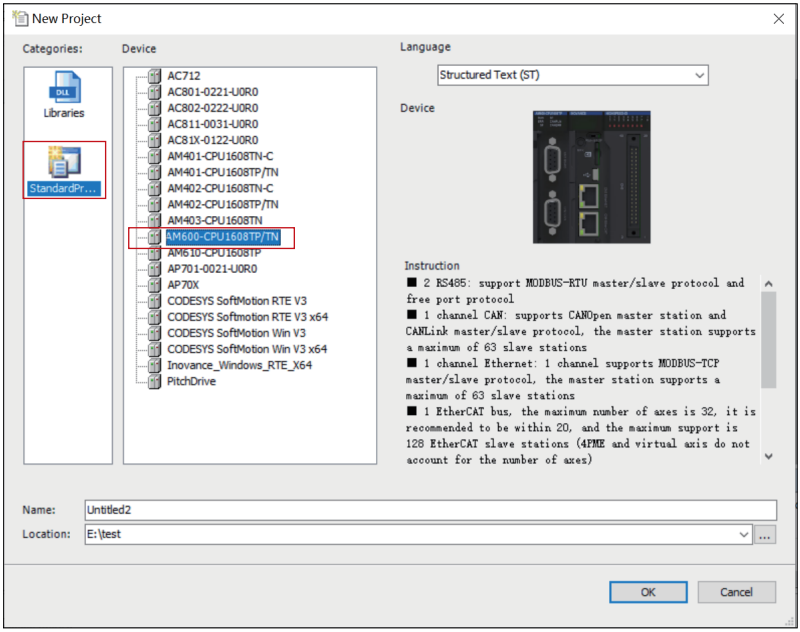
Figure 12-8 Modbus TCP networking of MD880 with AM600

PLC configuration

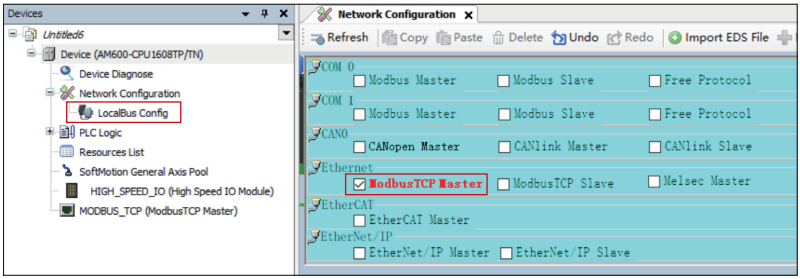
1. Create a project.



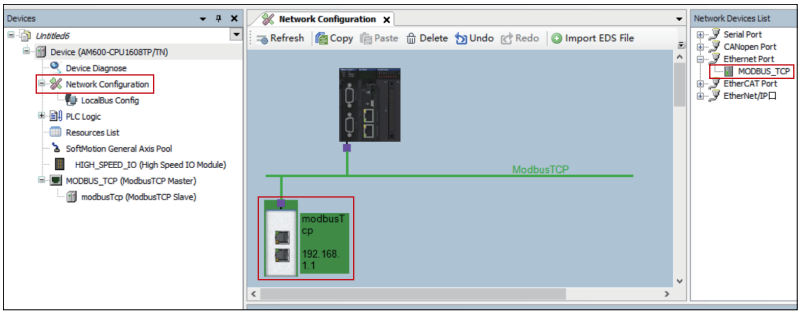
2. Select the PLC type.



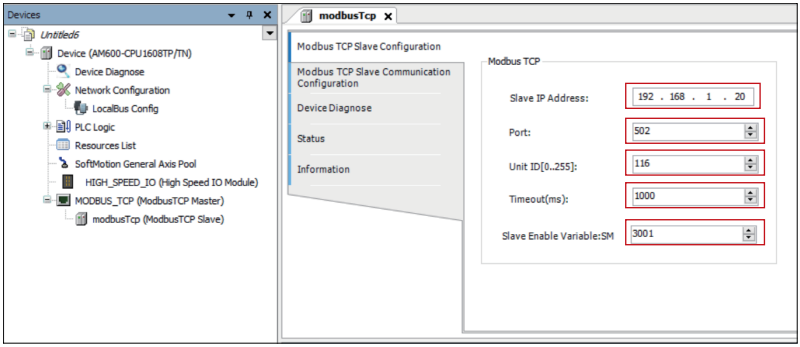
3. Set the AM600 as the Modbus TCP master.



4. Drag the MODBUS TCP device under the Ethernet port to the "Network Configuration" tab.



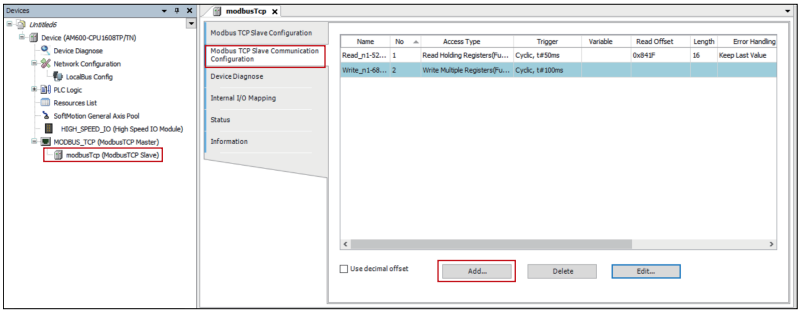
5. Set parameters such as the IP address. The IP address must be the same as that configured for the HMBT-10 module.



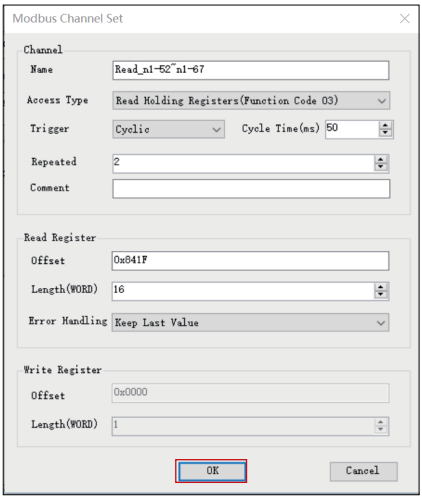
6. Configure the parameters to be read/written. This example explains reading multiple registers and writing multiple registers.

Note

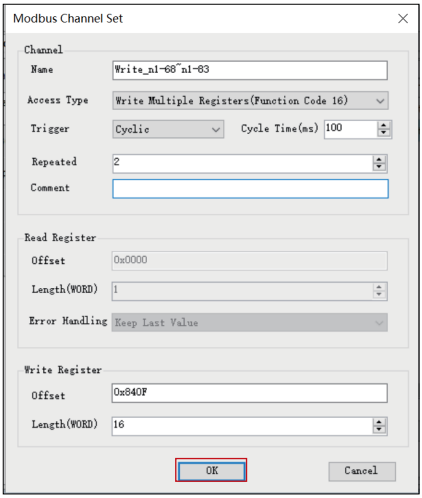
For reading or writing multiple registers, the parameter addresses must be consecutive.



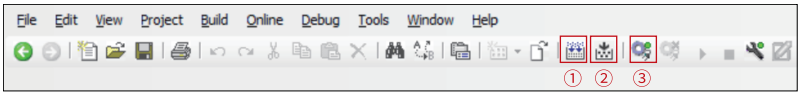
- Reading multiple registers:



- Writing multiple registers:



7. After all settings are complete, download the settings to the PLC.

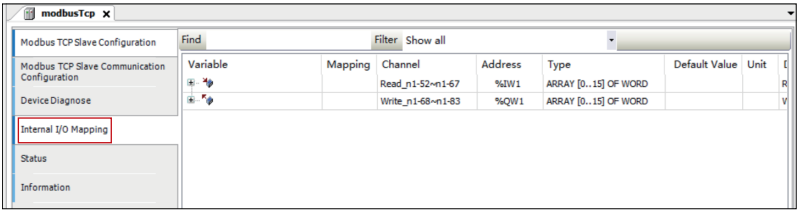


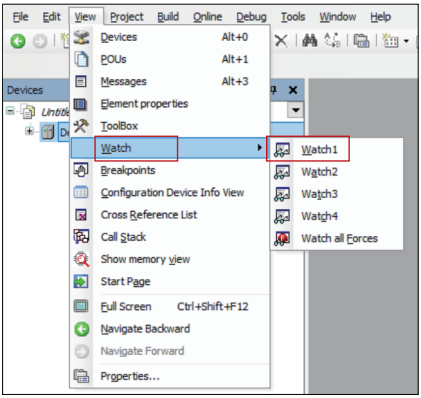
Note

- ①: Check the program.
- ②: Compile.
- ③: Download.

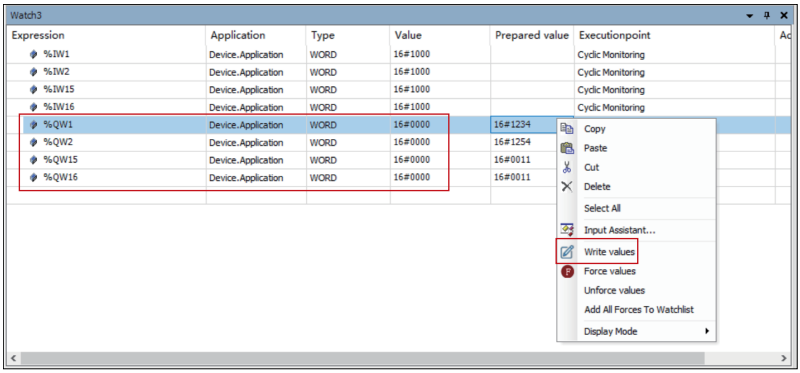
Data read/write check

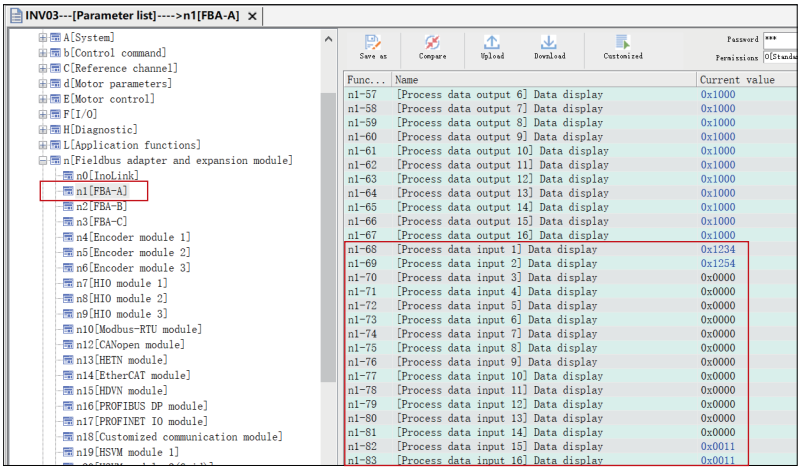
1. After configuration, check if the parameters can be read/written normally. Monitor addresses IW1 to IW16 and QW1 to QW16 in the PLC. IW1 to IW16 and QW1 to QW16 are the I/O address mappings in the PLC for the data to be read/written through Modbus TCP.



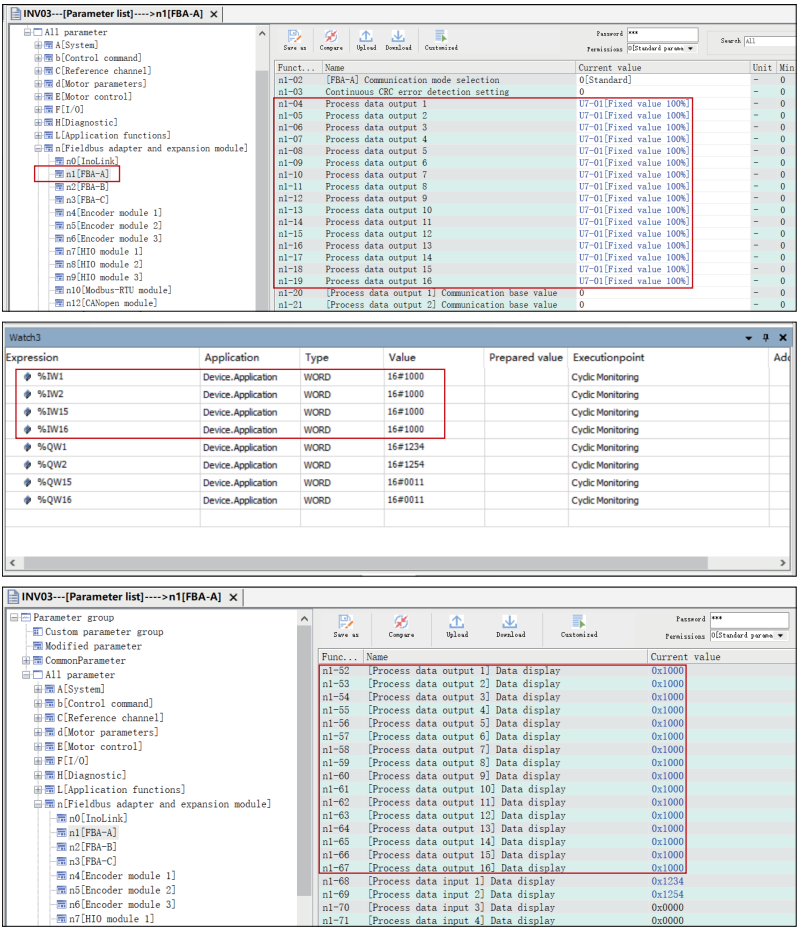


2. Make the PLC write process data to the MD880 drive. Before data writing, set the value of SM3001 to 1.





3. Make the PLC read data from the MD880 drive. The PLC reads data from n1-52 to n1-67. Therefore, parameters n1-04 to n1-19 must be configured. Ensure that an output fixed value of 100% corresponds to 16#1000.



12.6 EtherCAT

Networking

In this example, the PLC used is an Inovance AM600-CPU1608TN. The PLC programming software is InoProShop from Inovance. The MD880 drive commissioning software is InoDriveStudio. You can obtain the software from the Inovance official website at https://www.inovance.com.

Connect devices according to the following figure to complete networking. Set the SOP-20 to the USB relay mode.

Note

The AM600-CPU1608TN has two Ethernet ports. Use the CN4 port for EtherCAT communication.

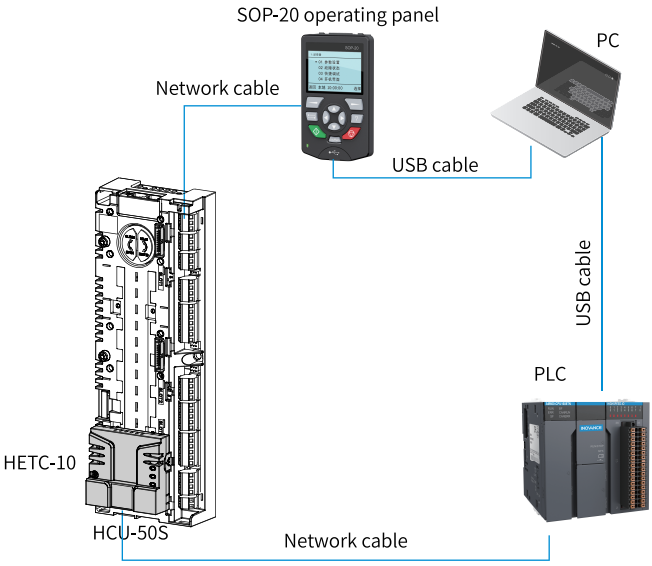
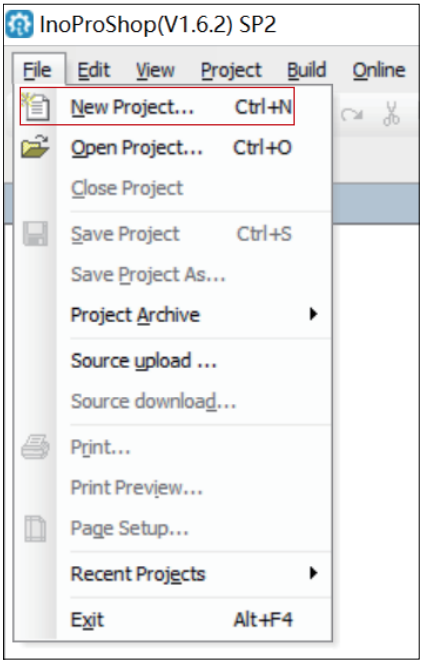


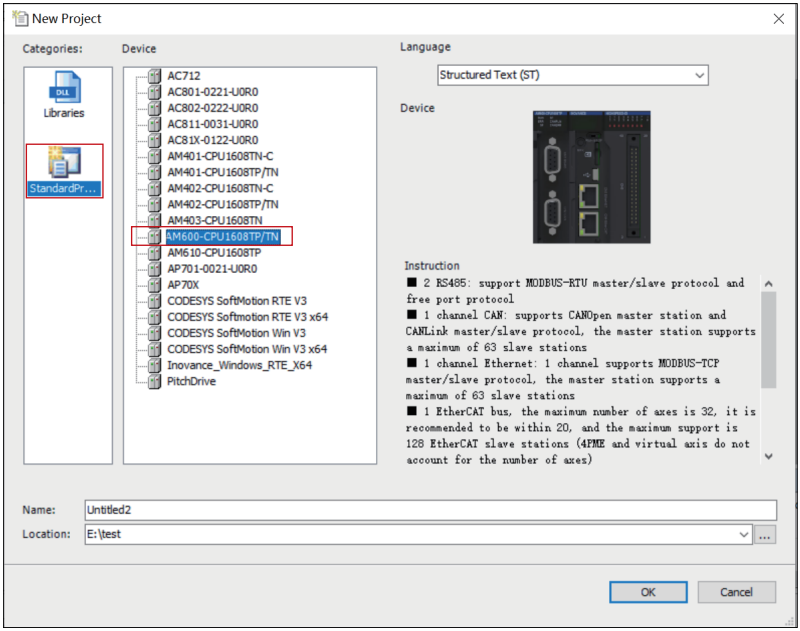
Figure 12-9 EtherCAT networking of MD880 with AM600

PLC configuration

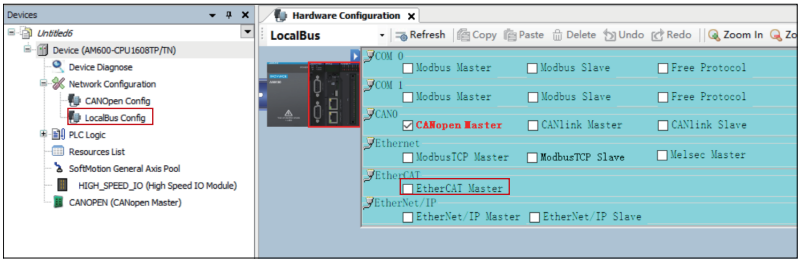
1. Create a project.



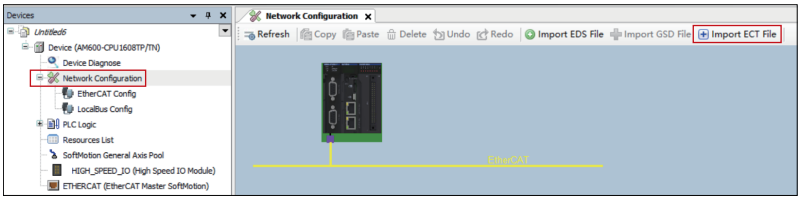
2. Select the PLC type.



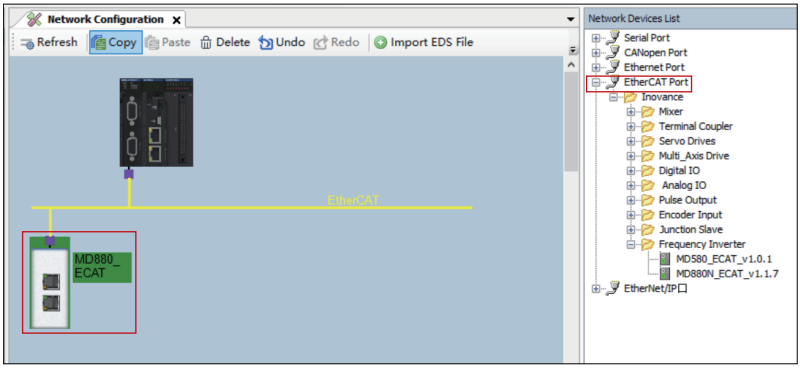
3. Set the AM600 as the EtherCAT master.



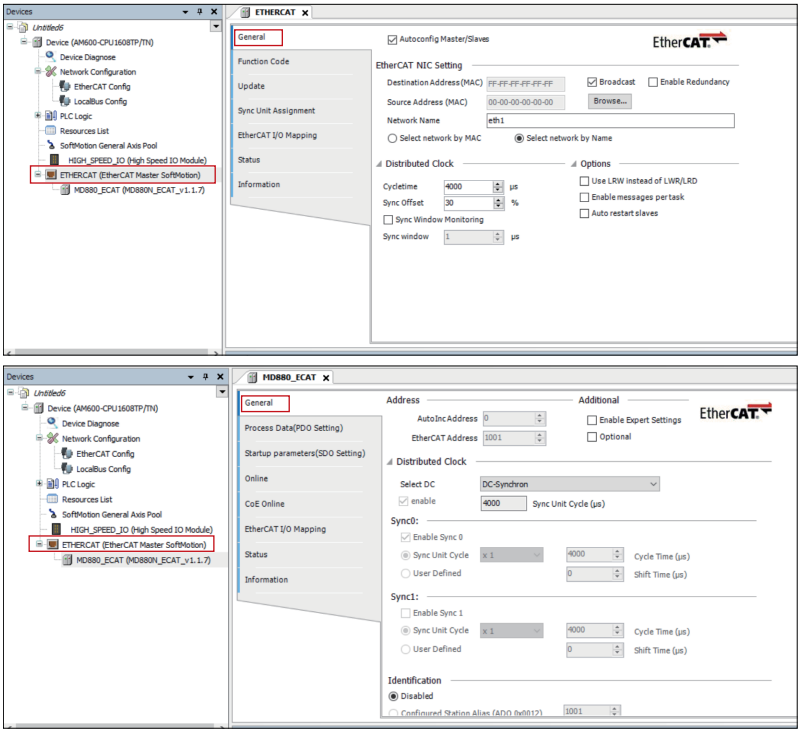
4. EtherCAT requires installing the XML file for the MD880 drive. You may obtain the XML file from Inovance or its agent.



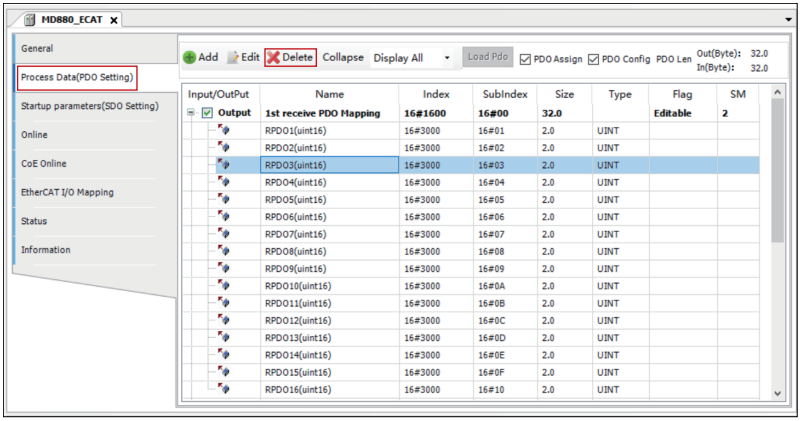
5. After the file is installed, the MD880N_ECAT device appears in the network device list. Drag the device to the "Network Configuration" tab.



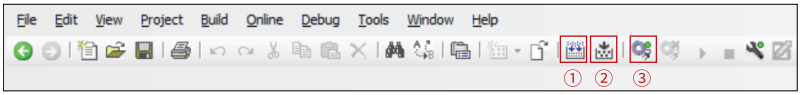
6. On the "General" tab, retain the default parameter settings for the master and slave.



7. Configure the PDO. By default, 16 parameters are periodically written, and 16 parameters are periodically read. You can add or reduce parameters as required. The MD880 drive supports a maximum of 16 parameters for periodic read/write.



8. After all settings are complete, download the settings to the PLC.

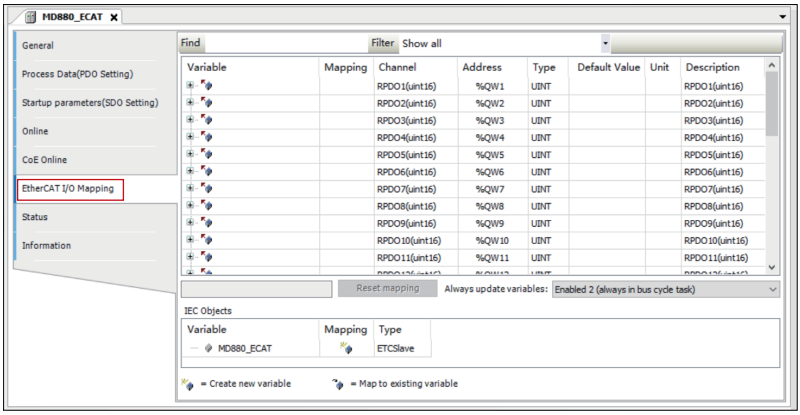


Note

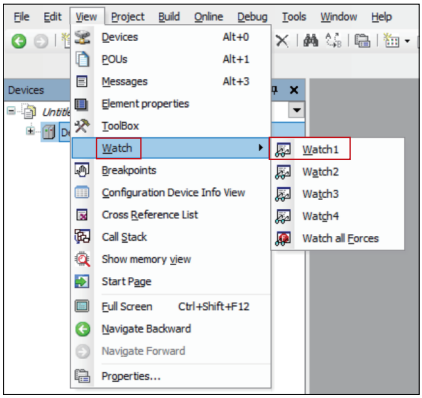
- ①: Check the program.
- ②: Compile.
- ③: Download.

Data read/write check

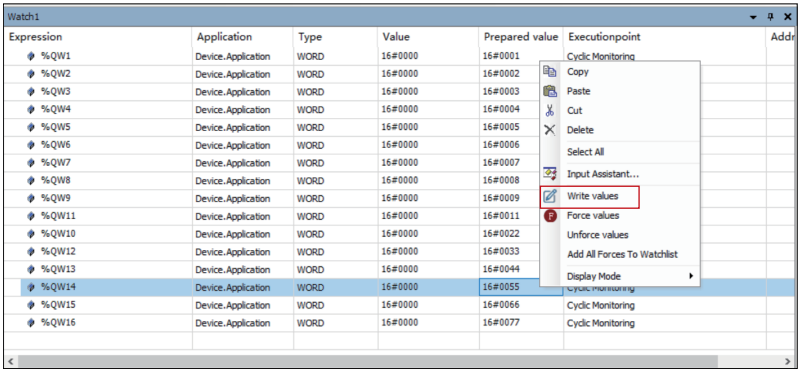
1. After loading the PLC configuration to the PLC, verify the transmission of process data, that is, check whether data can be transmitted between the MD880 and the PLC. Monitor addresses IW1 to IW16 and QW1 to QW16 in the PLC. IW1 to IW16 and QW1 to QW16 are I/O address mappings of EtherCAT process data in the PLC.

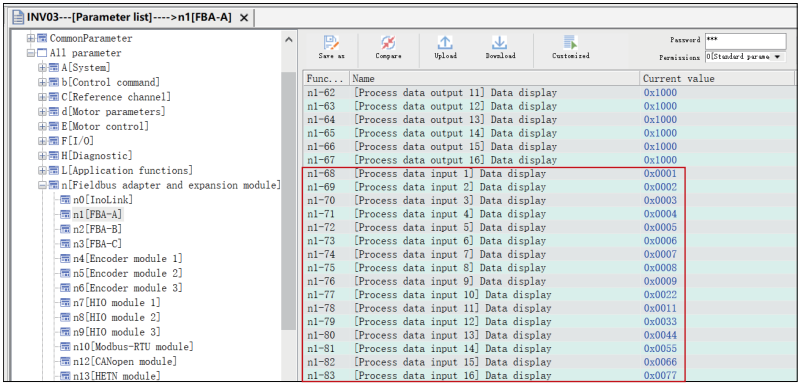


2. Add addresses IW1 to IW16 and QW1 to QW16 to the watch table.

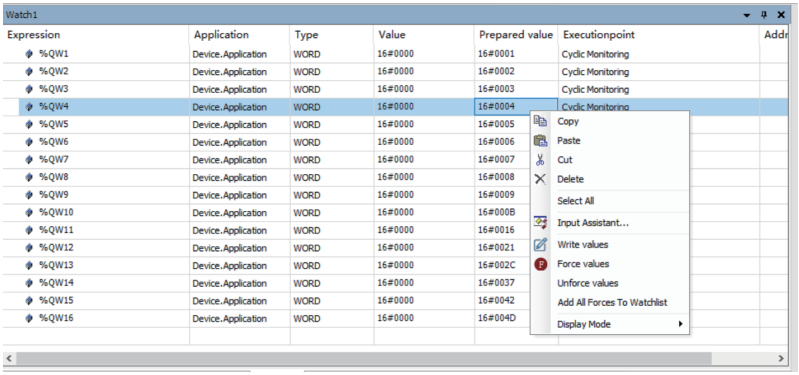


3. Make the PLC write process data to the MD880 drive.





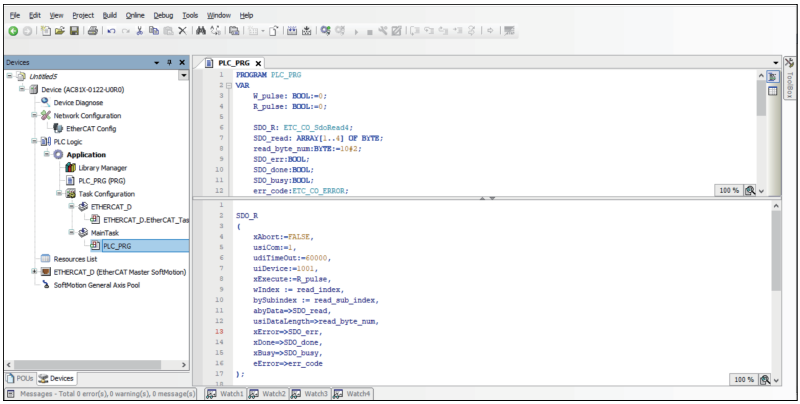
4. Configure the process data to be written from the MD880 to the PLC using parameters n1-04 to n1-19. An output fixed value of 100% corresponds to 16#1000.



Read/Write by SDO

This section takes reading C6-05 and writing C6-06 of the MD880 by SDO as examples. The address of C6-05 is 191, whereas the address of C6-06 is 192.

1. Program in the PLC.



2. Read C6-05 by SDO.

   Variables related to the reading operation are defined as follows:

```
W_pulse: BOOL:=0;
R_pulse: BOOL:=0;

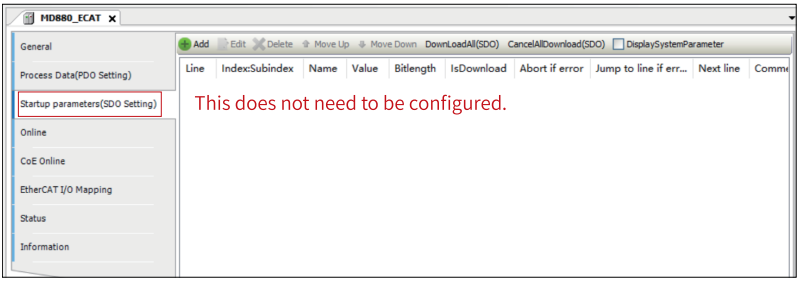
SDO_R: ETC_CO_SdoRead4;
SDO_read: ARRAY[1..4] OF BYTE;
read_byte_num:BYTE:=10#2;
SDO_err:BOOL;
SDO_done:BOOL;
SDO_busy:BOOL;
err_code:ETC_CO_ERROR;
read_index:WORD:=16#2000;
read_sub_index:BYTE:=10#192;
```

The reading method is implemented as follows: Add 1 to the register address, the index is 0x20 plus high bytes, and the sub-index is low bytes. For example, to read the address 191, add 1 to the address, that is 191 + 1 = 192 (0x00C0), so the index is 0x2000, and the sub-index is 0xC0. uiDevice specifies the EtherCAT address (for example, 1001) of the slave configured on the master.

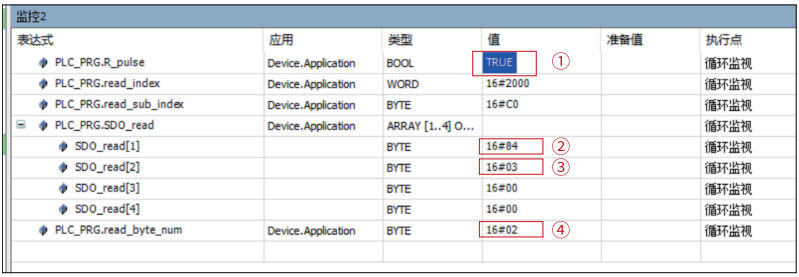
```
SDO_R
(
    xAbort:=FALSE,
    usiCom:=1,
    udiTimeOut:=60000,
    uiDevice:=1001,
    xExecute:=R_pulse,
    wIndex := read_index,
    bySubindex := read_sub_index,
    abyData=>SDO_read,
    usiDataLength=>read_byte_num,
    xError=>SDO_err,
    xDone=>SDO_done,
    xBusy=>SDO_busy,
    eError=>err_code
);
```

Note

When the PLC is used for SDO reading and writing, do not configure read and write addresses in the startup parameters of the slave station (SDO settings). Otherwise, the writer program cannot execute read and write operations.

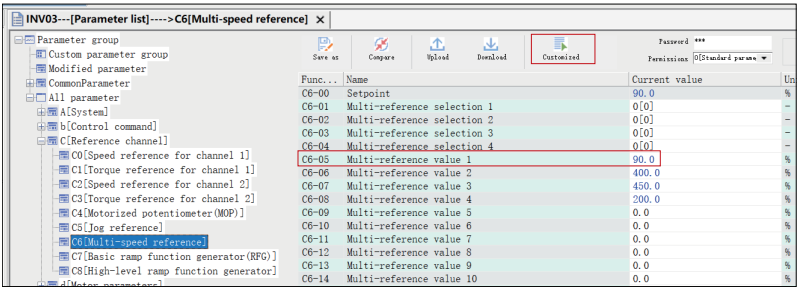


Perform operations in the watch table. When data is ready, a read operation is triggered, as shown in the following figure. The value of C6-05 that is read is 16#0384, corresponding to 10#900. C6-05 is read successfully.



Note

- ①: When the data is ready, TRUE is input.
- ②: Low bytes of read data.
- ③: High bytes of read data.
- ④: Total bytes of read data.



3. Write C6-05 by SDO.

   Variables related to the writing operation are defined as follows:

```
SDO_W: ETC_CO_SdoWrite4;
SDO_write_para: ARRAY[1..4] OF BYTE;
write_byte_num:BYTE:=2;
SDO_write_err:BOOL;
SDO_write_done:BOOL;
SDO_write_busy:BOOL;
SDO_write_err_code:ETC_CO_ERROR;
write_index:WORD:=16#2000;
write_sub_index:BYTE:=193;

wIndex: INT;
bySubindex: BYTE;
abyData: ARRAY [1..4] OF BYTE;
usiDataLength: BYTE;
xError: INT;
xDone: INT;
xBusy: INT;
eError: INT;
xExecute: INT;
```

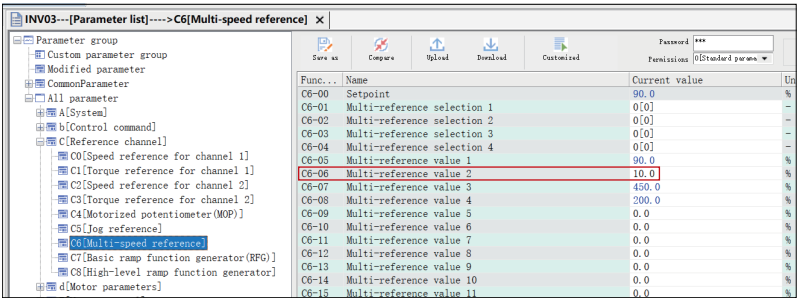
The writing method is implemented as follows: Add 1 to the register address, the index is 0x20 plus high bytes, and the sub-index is low bytes. For example, to write the address 192, add 1 to the address, that is 192 + 1 = 193 (0x00C1), so the index is 0x2000, and the sub-index is 0xC1. uiDevice specifies the EtherCAT address (for example, 1001) of the slave configured on the master.

```
SDO_W
(
    xAbort:=FALSE,
    usiCom:=1,
    udiTimeOut:=60000,
    uiDevice:=1001,
    xExecute:=W_pulse,
    wIndex := write_index,
    bySubindex := write_sub_index,
    abyData:=SDO_write_para,
    usiDataLength:=write_byte_num,
    xError=>SDO_write_err,
    xDone=>SDO_write_done,
    xBusy=>SDO_write_busy,
    eError=>SDO_write_err_code
);
```

Perform operations in the watch table. When data is ready, a write operation is triggered, as shown in the following figure. The value of C6-06 that is written is 16#0FA0, corresponding to 10#4000. C6-06 is written successfully.

Watch4

Expression	Application	Type	Value	Prepared value	Executionpoint
PLC_PRG.R_pulse	Device.Application	BOOL	TRUE	FALSE	Cyclic Monitoring
PLC_PRG.read_index	Device.Application	WORD	16#2000		Cyclic Monitoring
PLC_PRG.read_sub_index	Device.Application	BYTE	16#C0		Cyclic Monitoring
PLC_PRG.SDO_read	Device.Application	ARRAY [1..4] O...			Cyclic Monitoring
SDO_read[1]		BYTE	16#84		Cyclic Monitoring
SDO_read[2]		BYTE	16#03		Cyclic Monitoring
SDO_read[3]		BYTE	16#00		Cyclic Monitoring
SDO_read[4]		BYTE	16#00		Cyclic Monitoring
PLC_PRG.read_byte_num	Device.Application	BYTE	16#02		Cyclic Monitoring



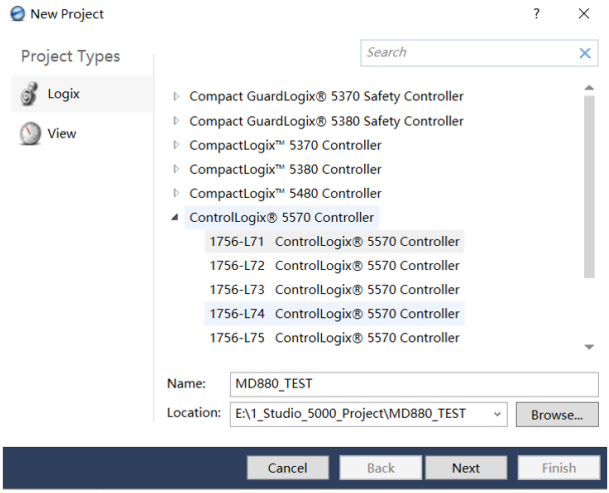
12.7 EtherNet/IP

12.7.1Example of Using HEIP-10 with AB 1769-L71 as Master

In this example, Studio 5000 version 32.00.00 is used, the master is 1769-L71, and the IP address and other settings have been configured in advance according to the user guide. Both network ports of the expansion card can be used.

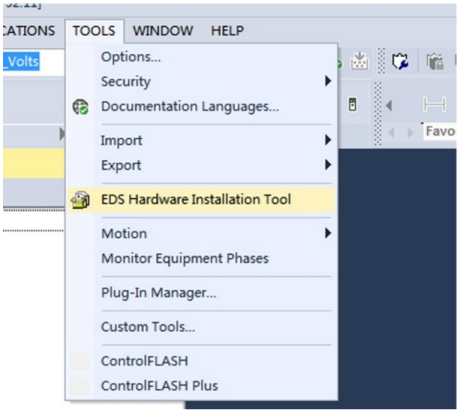


1. Create a project.
   Open Studio 5000 and create a project. Select "1756-L71 ControlLogix® 5570 Controller" under "ControlLogix® 5570 Controller" as the controller model.

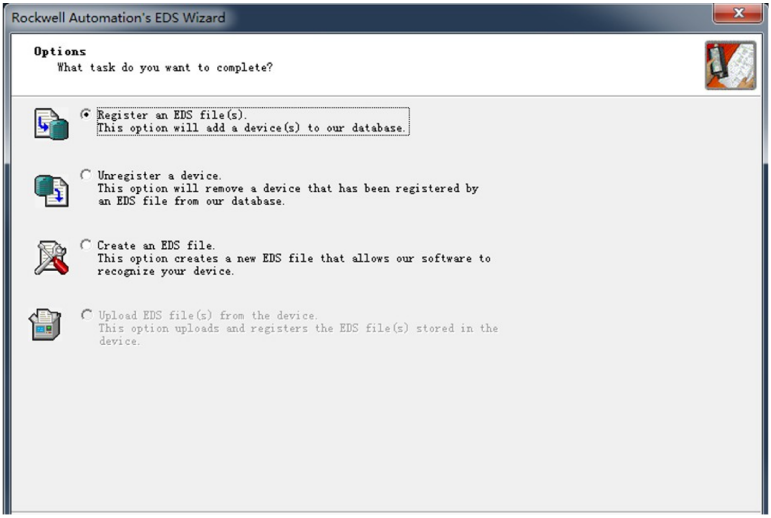


2. Import the EDS file.

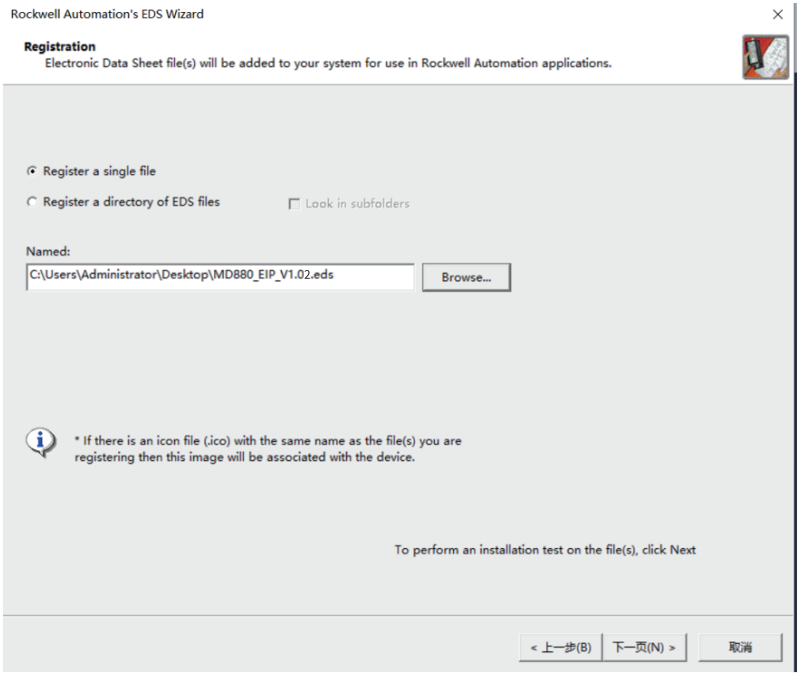
   a. Go to "Tools" > "EDS Hardware Installation Tool".



   b. Select "Register an EDS file(s)", and click "Next".

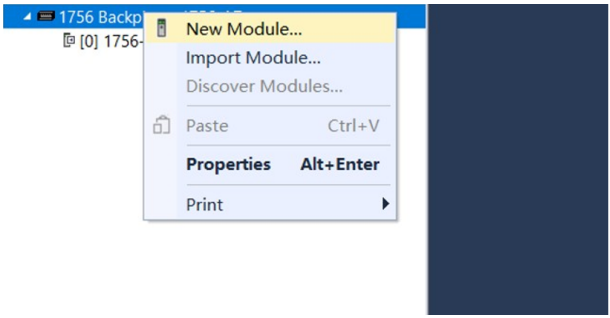


c. Select the EDS file stored in your computer and click "Next".

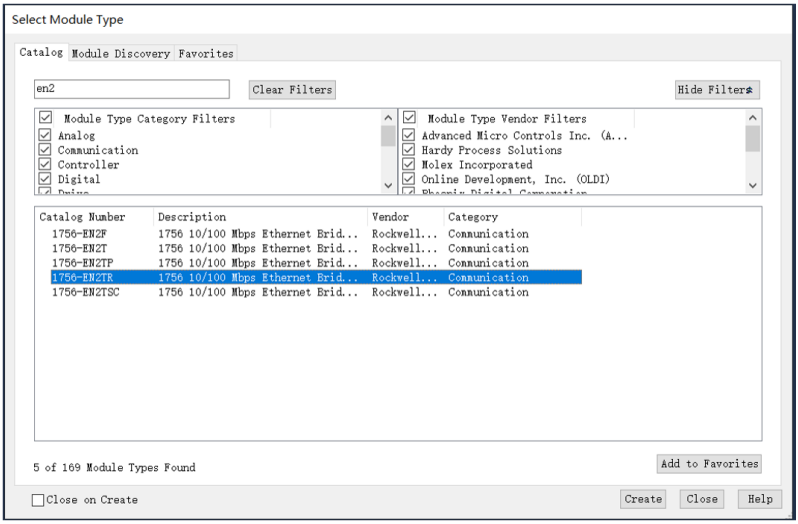


d. Keep clicking "Next" until the "Finish" button appears. Then click "Finish".

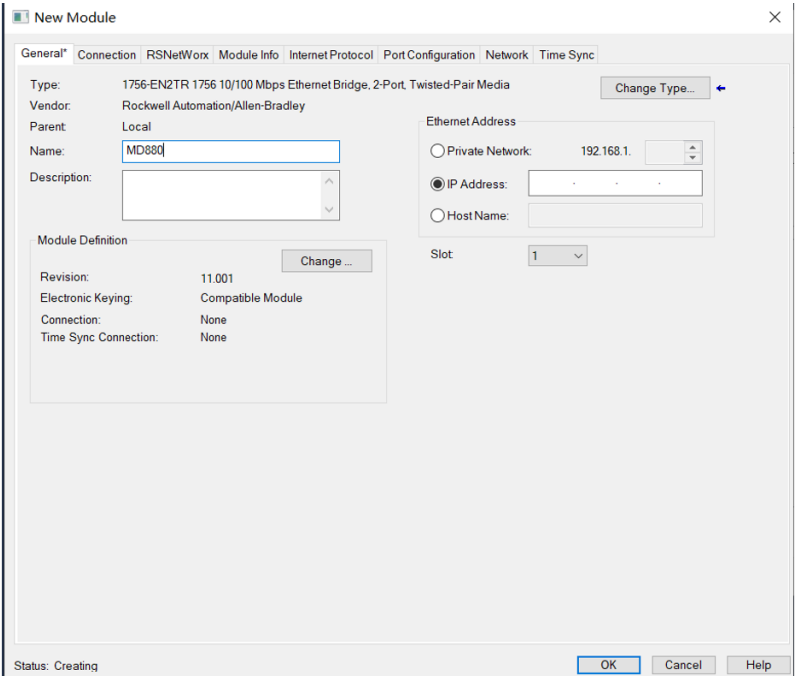
3. Set the IP address for the expansion card. In this example, a static IP is used. Set the drive parameters n41-02 to n41-14 to 0, 192.168.0.11, 255.255.255.0, and 192.168.0.1, respectively.
4. Configure the project on the Studio 5000.
   a. In the left pane, right-click "1756 Backplane" and select "New Module".



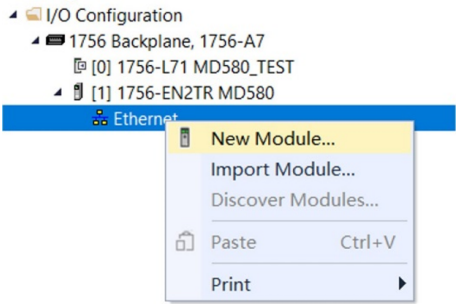
   b. Enter "en2" to perform search, and select "1756-EN2TR" in the result list.



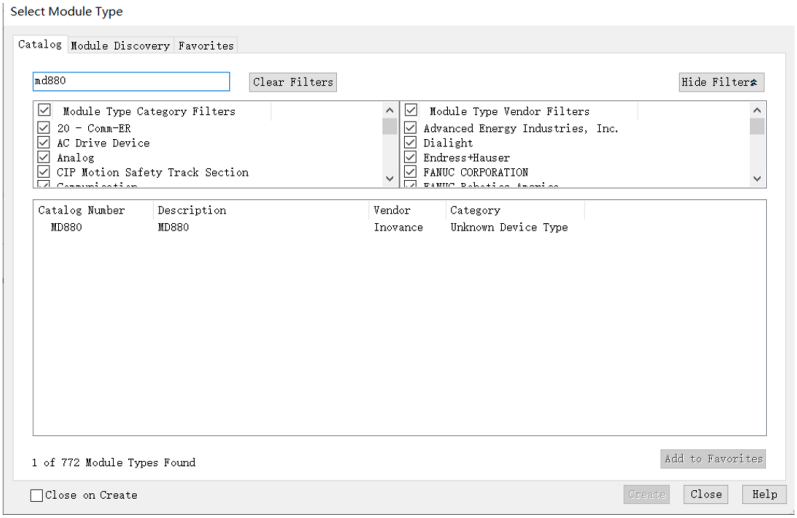
   c. Click "Create", enter a name in "Name", and click "OK".



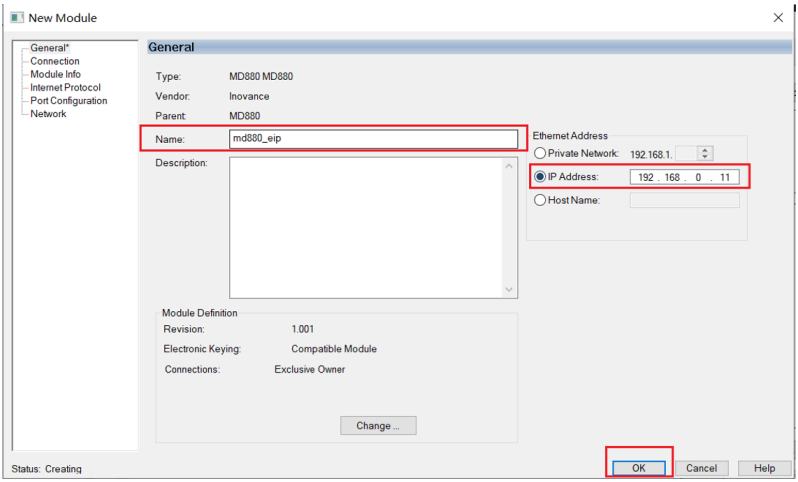
5. On the added device list, right-click "Ethernet" and select "New Module".



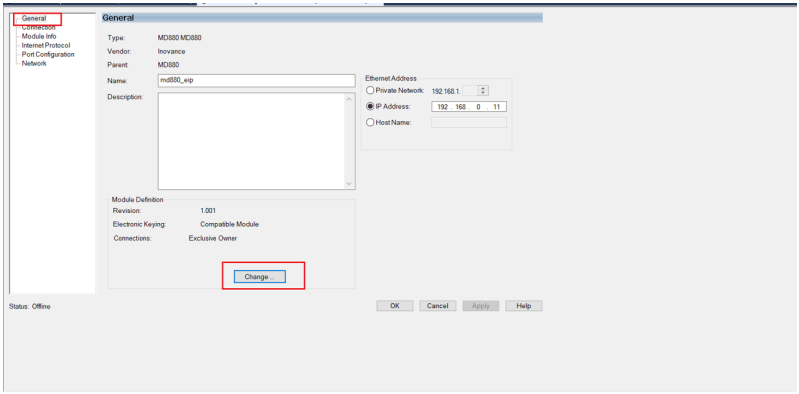
a. Locate "MD880"and click "Create".



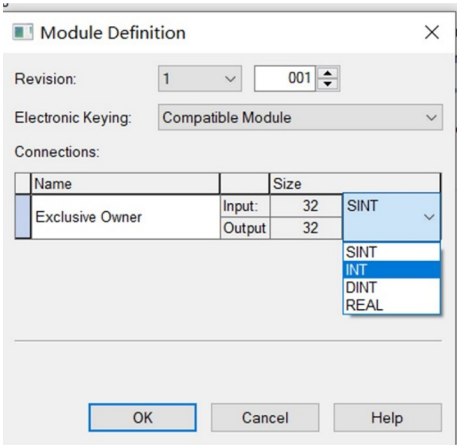
b. On the displayed configuration interface, enter the configured IP address and specify a name.



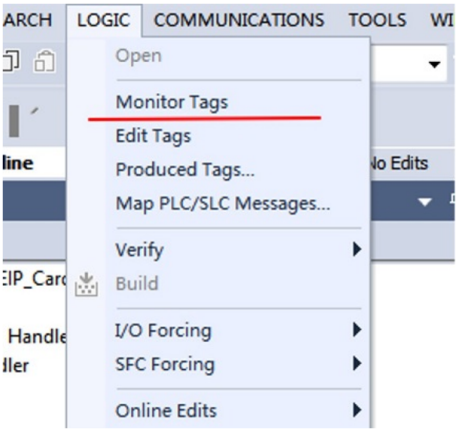
c. Double-click the added md880_eip. In the displayed interface, click "General", and find "Change".



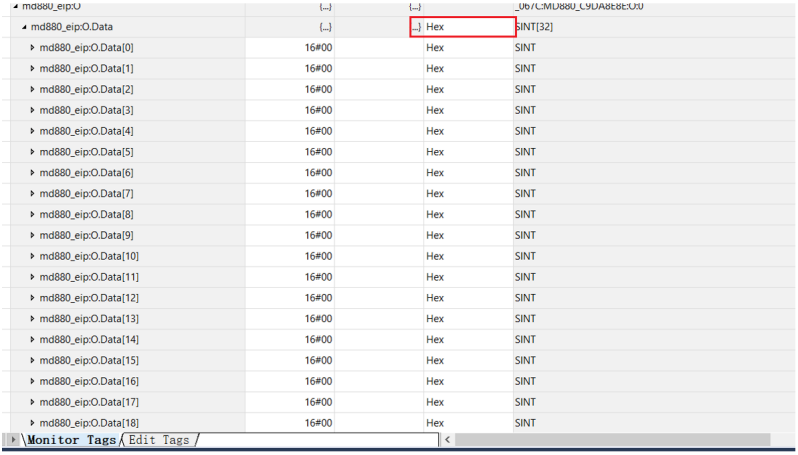
d. Click "Change", select INT from the SINT drop-down list on the right, click "OK", ignore the warning, and click "Yes".



e. Choose "LOGIC" > "Monitor Tags".



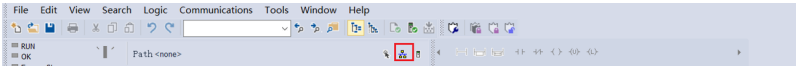
f. Unfold "md880_eip:O.Data", and select "Hex" in the "Style" column.



The parameters under "md880_eip:O.Data" are related to PDO mappings. Every two parameters form a group. Data 0 to data 32 are I/O message mappings (T->O). You can configure other fields as required.

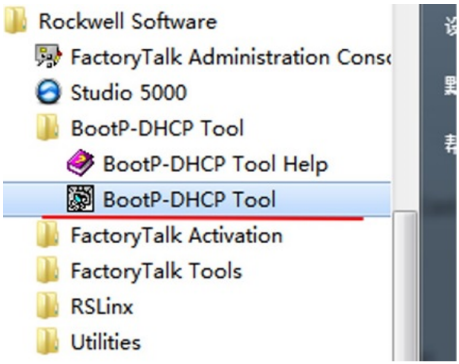
The md880_eip:I.Data and md880_eip:O.Data are I/O data during transmission. Values written to O.Data are actually written into the parameters corresponding to the configured mappings. The parameter configured by I/O message mapping (T->O) [0] is uploaded to I.Data regularly.

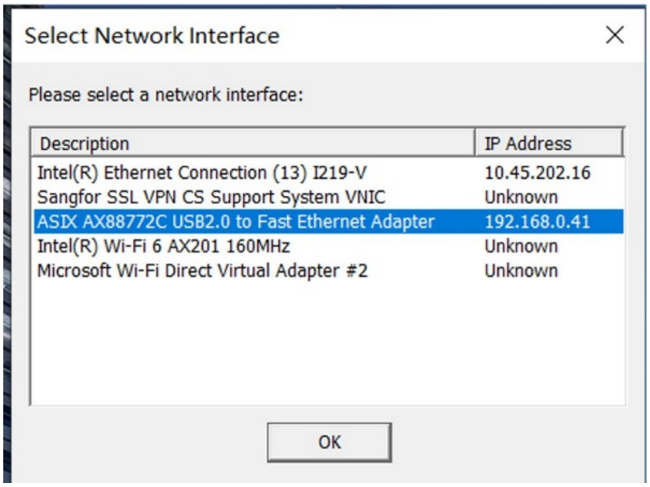
g. After the configuration is complete, click the search button to search for devices.



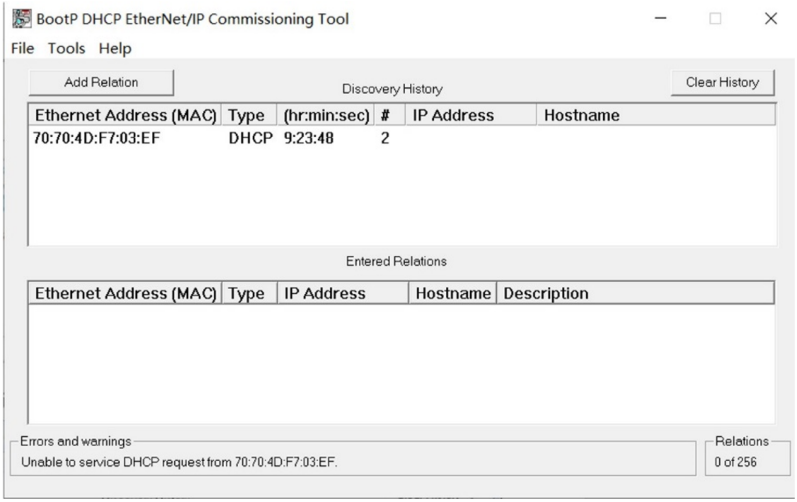
Select the device (connected through USB or Ethernet), click "Download", and download the code to the PLC.

6. Use the DHCP function.
   a. Set n41-02 to 1 to enable the DHCP function, restart the AC drive, and connect the PC and AC drive to the same network. On the start menu, select "BootP-DHCP Tool", and then select a network interface.

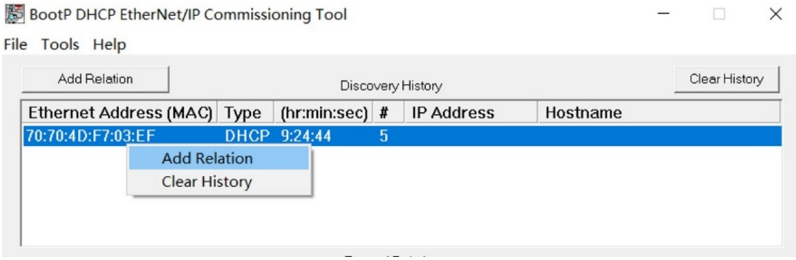




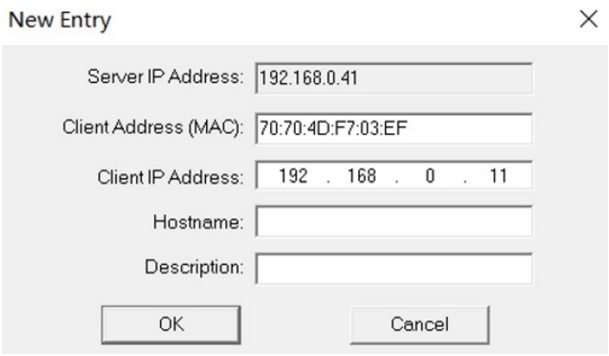
   b. After power-on, find the device request in the BootP DHCP EtherNet/IP Commissioning Tool.



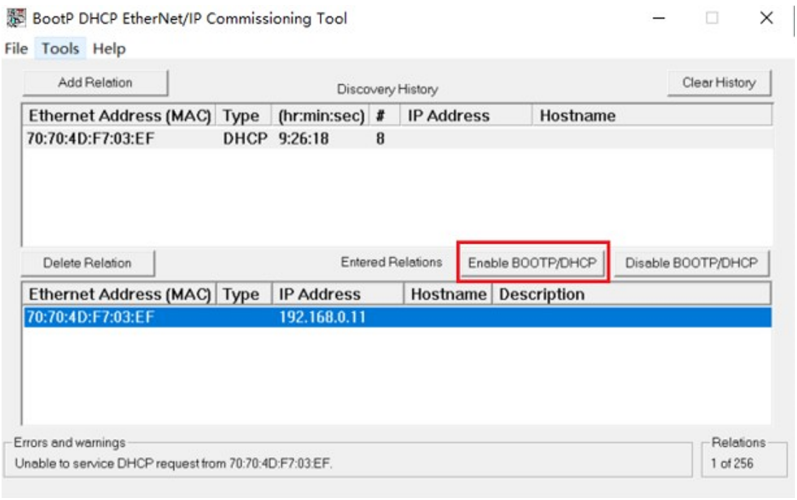
c. Right-click the request and choose "Add Relation".



d. Set the IP address and click "OK".



e. Select the device and click "Enable BOOTP/DHCP".



f. Wait for completion.

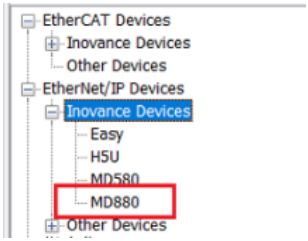
Note

After DHCP is configured, the IP address obtained through DHCP is written to n41-03 to n41-14. If DHCP is no longer needed, you need to set parameter n41-02 to 0 to disable DHCP. Otherwise, DHCP will be used to configure IP addresses each time you power on the device.

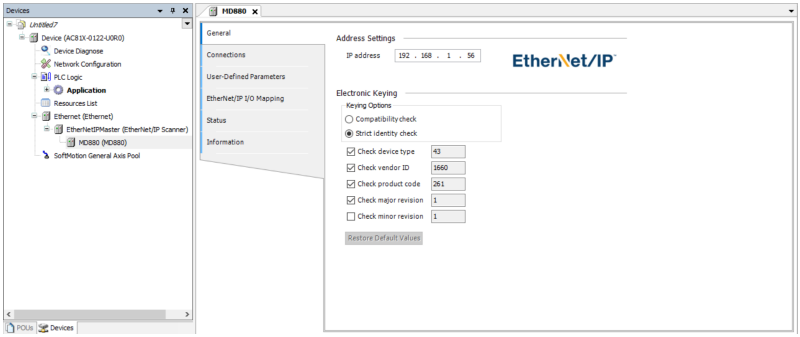
12.7.2Example of Using HEIP-10 with Inovance H5U as Master

In this example, AutoShop V4.7.9.0 is used and the master is H5U-1614MTD. IP addresses and other parameters have been configured in advance according to the user guide. Both sides of the EIP card can be used.

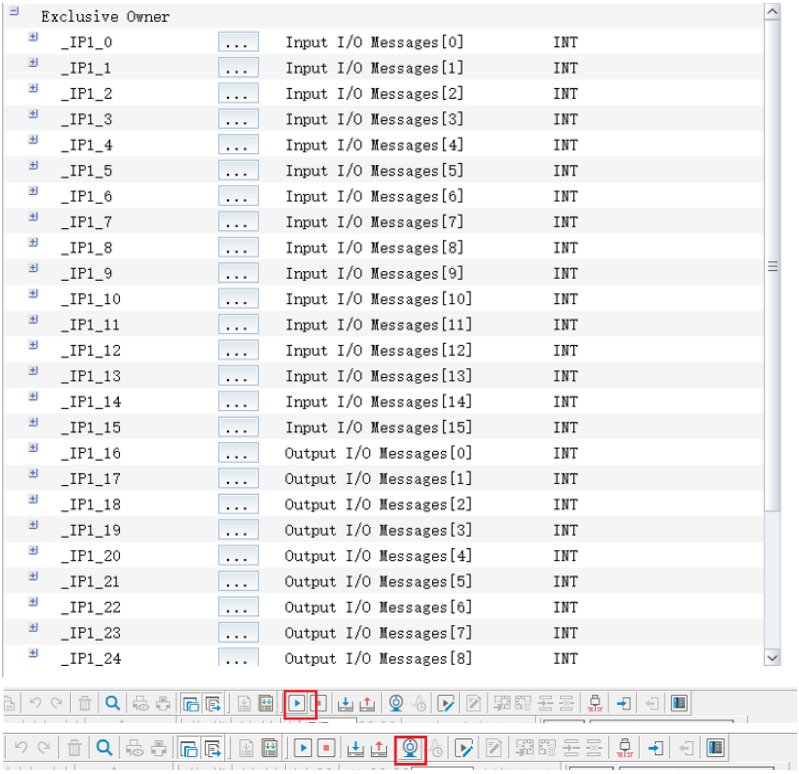
1. Create a project.
   Open the AutoShop, create and name a project, and select the H5U series as the device type.
2. Set the IP address for the MD880_EIP card. In this example, a static IP is used. Set the drive parameters n41-02 to n41-14 to 0, 192.168.0.11, 255.255.255.0, and 192.168.0.1, respectively. On the AutoShop, click the tool menu, and select "Communication Settings (C)". Take "USB" as an example of the communication type. Click "Test" to determine whether the PLC can be connected.
3. In the right tool box, right-click "EtherNet/IP Device" and select "Import EDS". Double-click the displayed "MD880".



4. Configure the slave parameters. Enter the configured static IP address in the "IP address" text box.



5. Configure the EtherNet/IP I/O mappings according to actual conditions, and compile and download to the PLC.



Exclusive Owner			
_IP1_0	...	Input I/O Messages[0]	INT
_IP1_1	...	Input I/O Messages[1]	INT
_IP1_2	...	Input I/O Messages[2]	INT
_IP1_3	...	Input I/O Messages[3]	INT
_IP1_4	...	Input I/O Messages[4]	INT
_IP1_5	...	Input I/O Messages[5]	INT
_IP1_6	...	Input I/O Messages[6]	INT
_IP1_7	...	Input I/O Messages[7]	INT
_IP1_8	...	Input I/O Messages[8]	INT
_IP1_9	...	Input I/O Messages[9]	INT
_IP1_10	...	Input I/O Messages[10]	INT
_IP1_11	...	Input I/O Messages[11]	INT
_IP1_12	...	Input I/O Messages[12]	INT
_IP1_13	...	Input I/O Messages[13]	INT
_IP1_14	...	Input I/O Messages[14]	INT
_IP1_15	...	Input I/O Messages[15]	INT
_IP1_16	...	Output I/O Messages[0]	INT
_IP1_17	...	Output I/O Messages[1]	INT
_IP1_18	...	Output I/O Messages[2]	INT
_IP1_19	...	Output I/O Messages[3]	INT
_IP1_20	...	Output I/O Messages[4]	INT
_IP1_21	...	Output I/O Messages[5]	INT
_IP1_22	...	Output I/O Messages[6]	INT
_IP1_23	...	Output I/O Messages[7]	INT
_IP1_24	...	Output I/O Messages[8]	INT

12.7.3Example of Using HEIP-10 with Inovance AC810 as Master

In this example, the InoProShop (V1.7.3) and AC810 are used.

1. Configure parameters of the HEIP-10 module, such as the IP address, gateway, and subnet mask.
   Before connecting to the AC810 master station, set parameters such as the IP address, gateway, and mask on the InoDriveStudio. If DHCP is used, use the BOOTP to assign IP addresses.

   In this example, a static IP is used:

   - For example, to set the static IP address to 192.168.0.11, the subnet mask to 255.255.255.0, and the gateway to 192.168.0.1, set n41-02 to n41-14 of the AC drive on the background software according to the following table.

Param.	Function	Setpoint
n41-02	DHCP enable	0
n41-03	IP address 1	192
n41-04	IP address 2	168

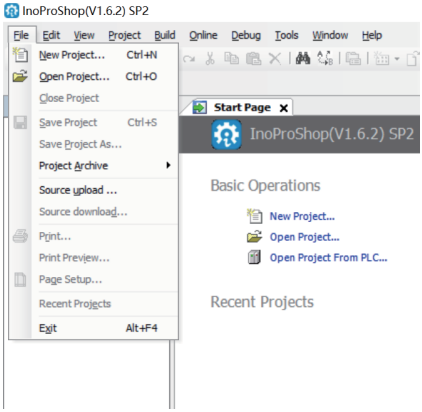
Param.	Function	Setpoint
n41-05	IP address 3	0
n41-06	IP address 4	11
n41-07	Subnet mask address 1	255
n41-08	Subnet mask address 2	255
n41-09	Subnet mask address 3	255
n41-10	Subnet mask address 4	0
n41-11	Gateway address 1	192
n41-12	Gateway address 2	168
n41-13	Gateway address 3	0
n41-14	Gateway address 4	1

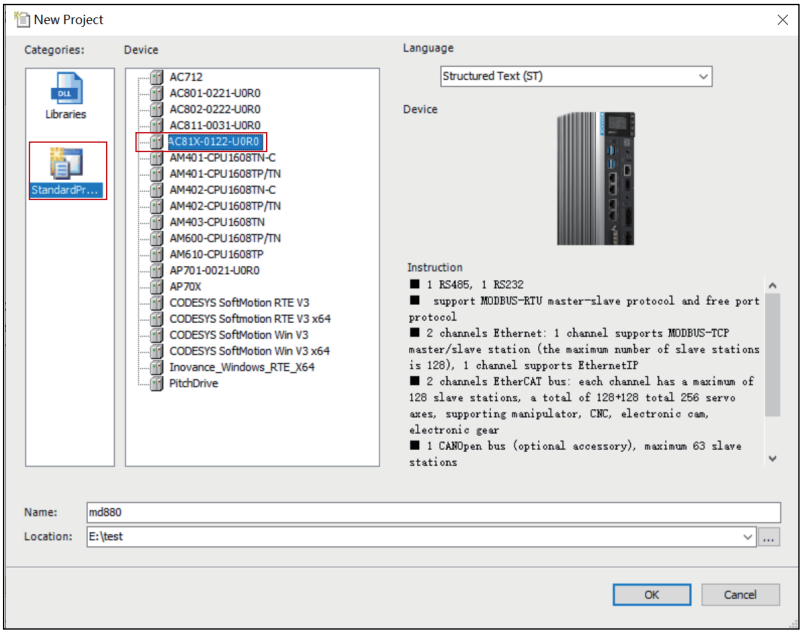
- Connect the HEIP-10 module to the LAN port A of the AC810 master station and connect the AC810 to the PC.

Note

The AC810 provides only LAN ports A and B and supports the EtherNet/IP protocol.

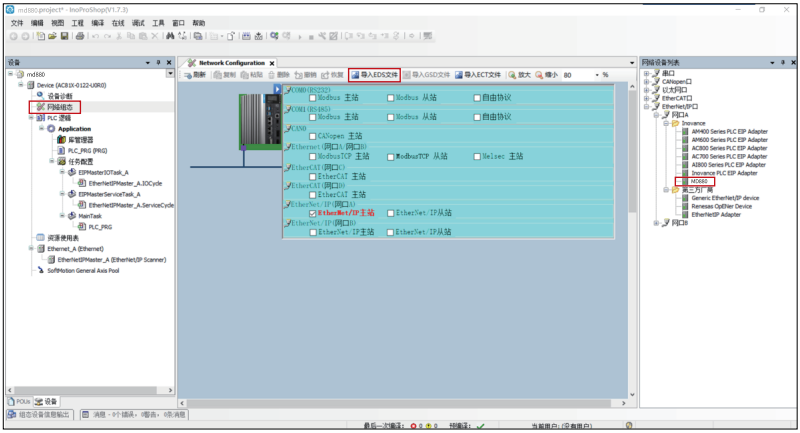
2. Create a project for the AC810.
   Open the InoProShop and create a project. Select "AC81X-0122-U0R0" in the "Device" pane.





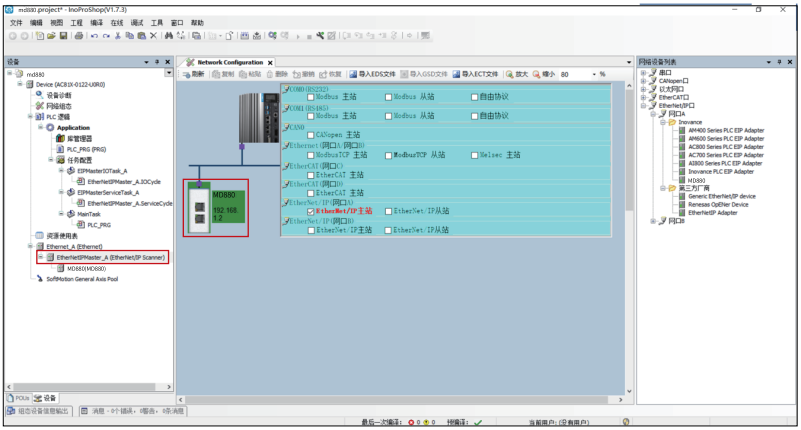
3. Import the EDS file and add a slave.

   a. Click "Network Configuration" in the left "Devices" pane. On the "Network Configuration" tab, click "Import EDS File", and then import the EDS file of the HEIP-10 module. Click the PLC, select "EtherNet/IP Master" for the network port of the connected device. Double-click the MD880 on the right device list to import the device. As the HEIP-10 module is connected to the LAN port A, configure LAN port A as the EtherNet/IP master.

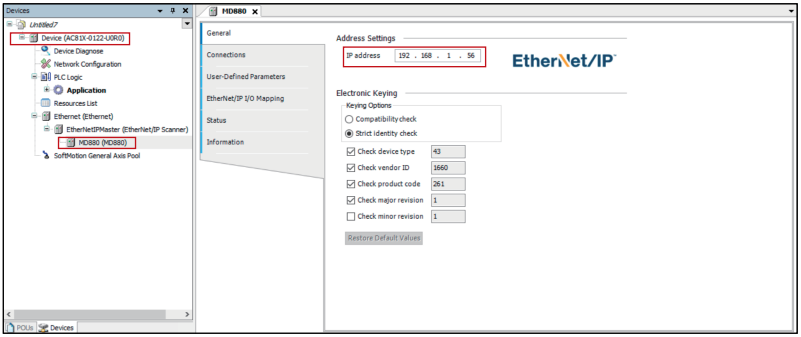


   b. Add the slave and configure the slave parameters.

In the "Devices" pane, right-click "EtherNetIPMaster_A(EtherNet/IP Scanner)" and select the device adding option, or in the "Network Device List" pane, double-click "MD880". The HEIP-10 module is added.

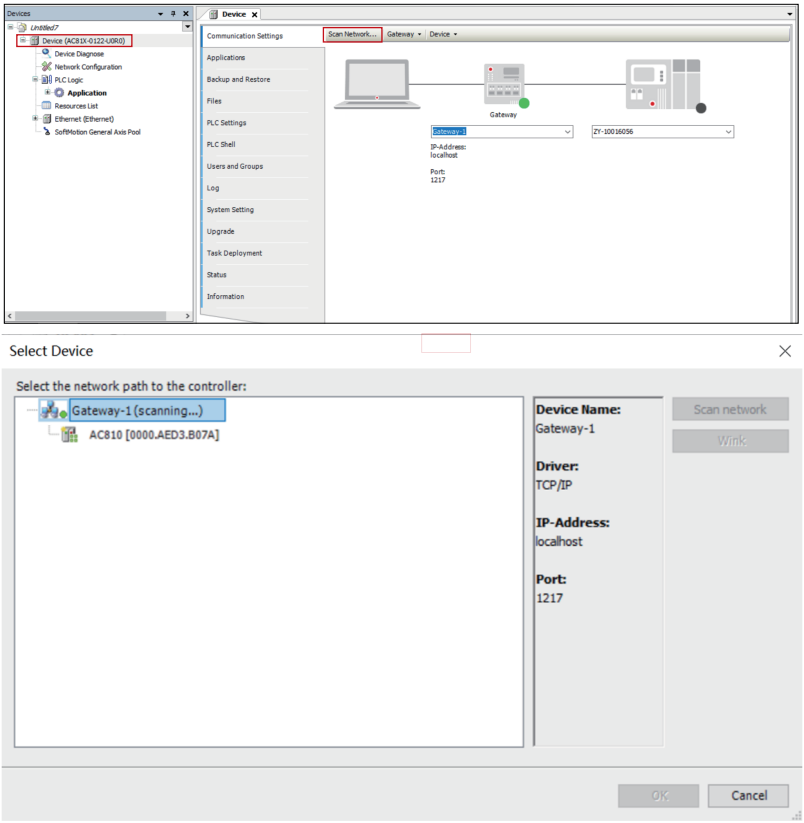


Double-click the added MD880 device to configure the IP address (for example, 192.168.0.11). The IP address must be the same as that configured on the InoDriveStudio.

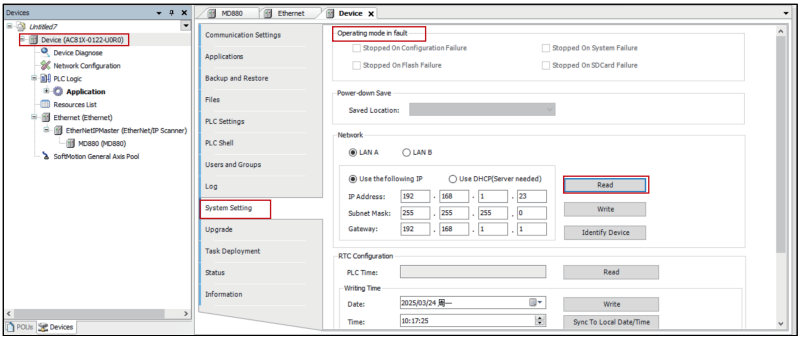


4. Configure the IP address for the master.

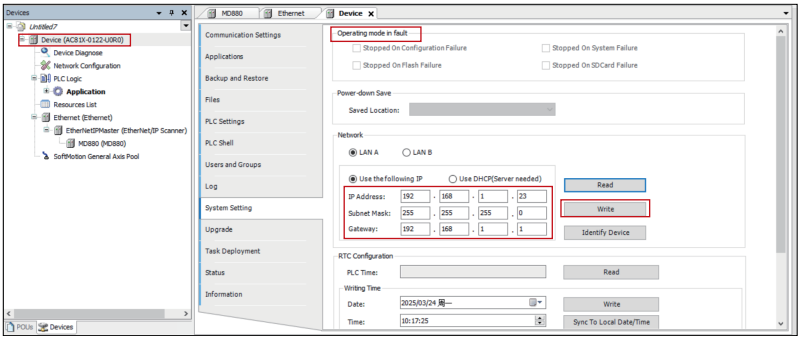
   a. Double-click "Device" in the left pane. On the "Communication Settings" tab, click "Scan Network", and then select AC810 in the scanned device list.



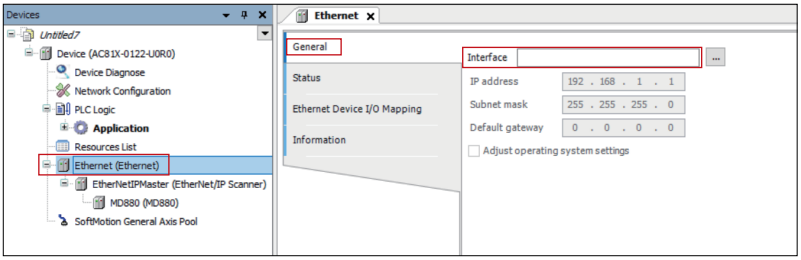
b. To configure the IP address (LAN port A as an example) of the master, double-click "Device" in the left pane. On the "Device" tab, click "System Setting", and then click "Read" to obtain the master IP address.

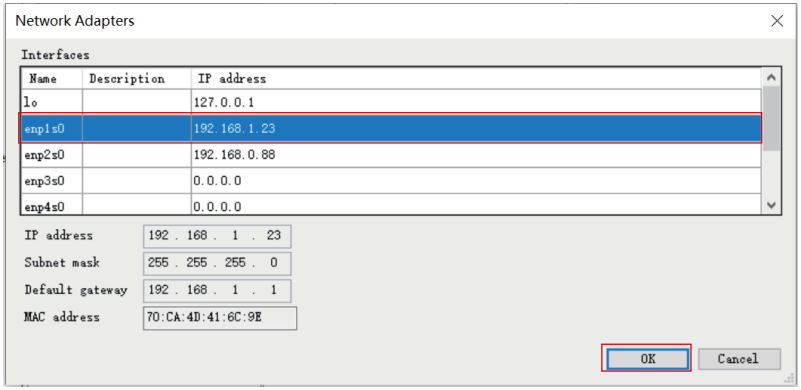


Change the IP address of the master to the same network segment as the configured IP address of the slave. As the network segment is 192.168.0.X, change the IP address of the master to 192.168.0.122, and then click "Write".



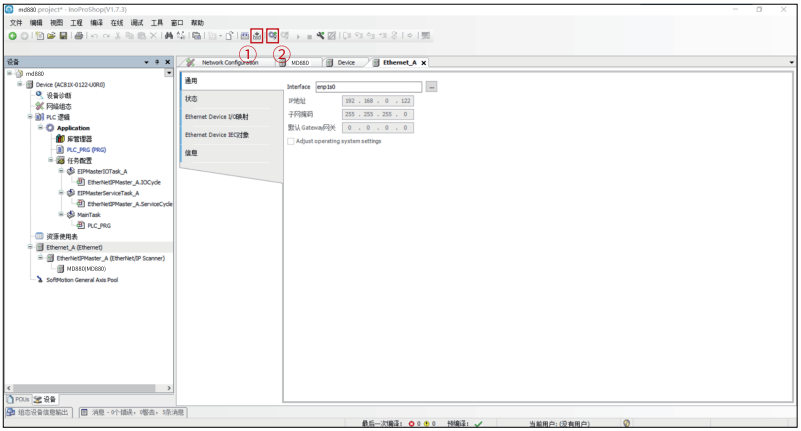
c. Select the IP address of the master. Double-click "Ethernet_A(Ethernet)" in the left pane. On the "Ethernet" tab, click "General". Click "..." after "Interface", select the configured IP address of the LAN port A.





5. Compile and run.

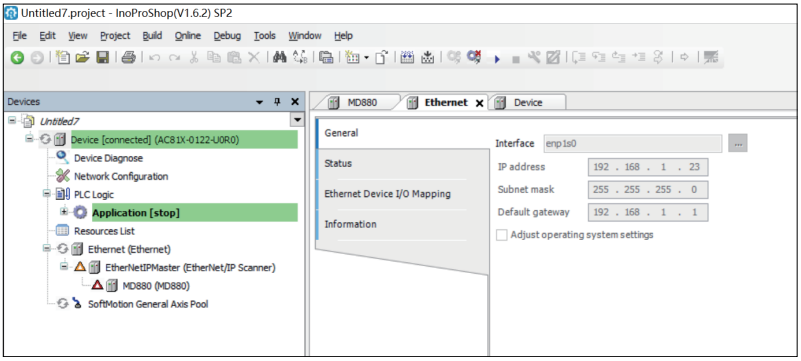
a. Click the compiling key and then click the login key.



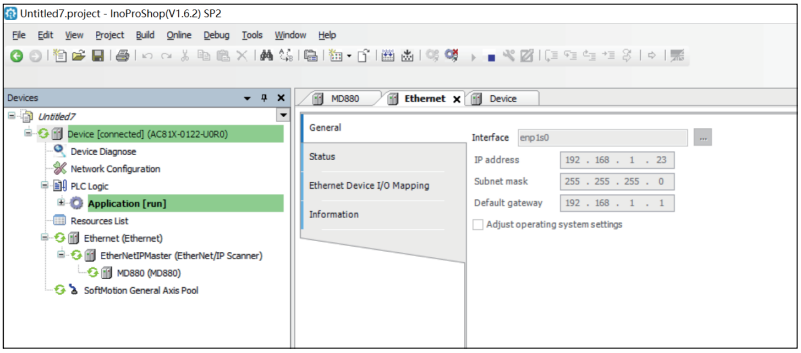
Note

- ①: Compiling
- ②: Login

b. Click the start key to start the master.

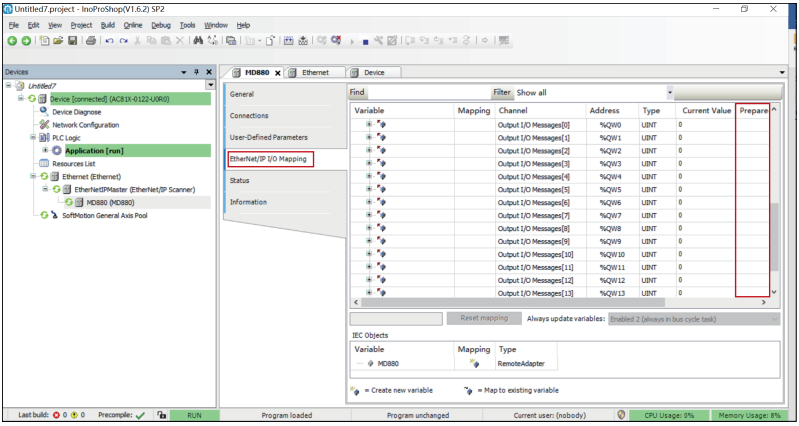


On the slave side, the original red triangle icon changes to the green cycle icon, indicating that communication is successfully.



6. View and configure the process data.

   In the left "Devices" pane, double-click "MD880". Click "EtherNet/IP I/O mapping" to view the process data. You can configure the process data as required.



13 Communication Troubleshooting

13.1 PROFIBUS DP

The module provides three status indicators to indicate bus communication faults, as described below.

Indicator	Color	Status	Solution
INOBUS	Green light steady ON	Communication with the HCU control module is normal.	-
	Red light flashing	Establishing communication with the HCU control module or communication failed	• Check if the HDP-10 on the HCU is loose. • Check for physical connection issues between the HCU and HDP-10 slot.
MODULE	Green light steady ON	Module status normal	-
	Red light steady ON	Device ID invalid	Check if the expansion card ID setting (n16-02) is correct.
	Red light flashing	Establishing communication with the HCU control module	• Check if the HDP-10 on the HCU is loose. • Check for physical connection issues between the HCU and HDP-10 slot.
FIELDBUS	Green light steady ON	The bus network connection is normal.	-
	Green light flashing	Establishing bus network communication	-
	Red light steady ON	Bus network communication failed	• Check if the PROFIBUS DP communication link is normal. • For long communication distances, check if the termination resistors at the first and last devices on the PROFIBUS DP link are set to "ON".
	Red light flashing	Master lost	Check if the master station is operating normally.
	Orange light flashing	Slave configuration error	Check the n16 group communication parameter settings.

13.2 PROFINET IO

The module provides three status indicators to indicate bus communication faults, as described below.

Indicator	Color	Status	Solution
INOBUS	Green light flashing	Handshaking between the HPFN-10 module and the drive failed	• Check that only n1-00 or n2-00 is set to "PROFINET IO", not both. • Check that n17-00 is set properly. • Check that the HPFN-10 socket is intact.
MODULE	Green light flashing	Protocol stack starting	Just wait.
	Red light flashing	MAC address error	The HPFN-10 module does not have the MAC address. Replace it.
	Red light steady ON	Protocol stack start timeout	Power off and on again or replace the module.
	Orange light steady ON	Incorrect HCU communication parameter setting	Set the correct slot in parameter n17-00 to activate the PROFINET card
FIELDBUS	Red light flashing	Master not found	• Check that the cable is correctly connected. • Check that the slave is configured. • Check that a device name is assigned to the slave.
	Red light steady ON	Communication with the master station PLC lost	Check that the cable is correctly connected.
	Orange light flashing	Configuration error	Check that the GSDML configuration is correct.

13.3 Modbus RTU

The module provides three status indicators to indicate bus communication faults, as described below.

Indicator	Color	Status	Solution
INOBUS	Green light flashing	Establishing communication with the HCU control module	-
	Green light steady ON	Communication with the HCU control module is normal.	-
	Red light flashing	Communication with the HCU control module has failed.	Check if the HMBA-10 module is damaged.
MODULE	Green light flashing	Module initialization	-
	Green light steady ON	Module initialization completed or working normally	-
	Red light flashing	Module fault	Check if the HMBA-10 module is damaged.
FIELDBUS	Green light flashing	Module initialization	-
	Green light steady ON	Bus communication normal	-
	Red light flashing	Bus communication fault	1. Check if the n10-04 address setting is correct. 2. Check the RS485 communication link. 3. Check if the termination resistor DIP switch of the HMBA-10 module is set correctly. For point-to-point connection, both the master and slave stations need to have the termination resistors connected.

13.4 CANopen

The module provides three status indicators to indicate bus communication faults, as described below.

Indicator	Color	Status	Solution
INOBUS	Green light flashing	Establishing communication with the HCU control module	-
	Green light steady ON	Communication with the HCU control module is normal.	-
MODULE	Green light steady ON	Module operation normal	-
	Red light flashing once and off for 1 second	CAN bus passive error alarm	• Check if parameter n12-00 is set correctly. • Check if the bus adapter type setting (n1-00) is correct. • Check the CANopen communication link.
	Red light flashing twice and off for 1 second	CANopen heartbeat timeout	Check if the master is sending heartbeat messages normally.
	Red light flashing three times and off for 1 second	PDO receive length inconsistent with the configuration	Check if the RPDO and TPDO length settings in the HCAN-10 module parameters are correct.
	Red light flashing	Reading communications parameters	Check if the communication baud rate and slave address settings of the HCAN-10 module are correct.
	Red light steady ON	CAN bus disconnected	• Check the CANopen communication link. • Check if the termination resistor DIP switch of the HCAN-10 module is set correctly. For point-to-point connection, both the master and slave stations need to have the termination resistors connected.

Indicator	Color	Status	Solution
FIELDBUS	Green light steady ON	Bus in operation state	-
	Green light flashing once and off for 1 second	Bus stopped	• Check the CANopen communication link. • The master is not running. Check the master configuration.
	Green light flashing	Bus in pre-operational state	• Check the CANopen communication link. • The master is not running. Check the master configuration.

13.5 EtherCAT

The module provides three status indicators to indicate bus communication faults, as described below.

Indicator	Color	Status	Solution
INOBUS	Green light flashing	Establishing communication with HCU or internal communication timeout	Check if the HCU is powered on and check the connection between the expansion card and the HCU.
MODULE	Green light steady ON	Module running normally	-
	Orange light steady ON	Validating and updating EEPROM XML information	EEPROM information is initialized on first power-on.
	Red light flashing once per 200 ms	Parameter read error	-
	Red light flashing once per 1000 ms	EEPROM loading error	Power off and then power on the device again. If the problem persists, contact Inovance technical support.
	Red light flashing twice per 2000 ms	Hardware initialization error	Check if the ESI file was downloaded correctly or contact Inovance technical support.
	Red light flashing three times per 2000 ms	Slave state transition error	Invalid state transition; check communication configuration parameters.
	Red light steady ON	Bus disconnected	Check the communication cable; the device has exited OP state.
	OFF	Expansion card not activated	• Check if communication between the expansion module and HCU is normal. • Check if n14-00 is set correctly. • Check if the bus adapter type setting (n1-00) is correct.
FIELDBUS	Green light steady ON	OP state	-
	Green light ON 20 ms and OFF 1000 ms	Safe-OP state	-
	Green light ON 200 ms and OFF 200 ms	Pre-OP state	-
	OFF	Init state	-

13.6 EtherNet/IP

Possible faults during the usage of the HEIP-10 with the AC drive are listed in the table below.

Table 13–1 Fault causes and solutions

Fault Symptom	Cause	Solution
HEIP-10 cannot communicate with the AC drive.	• The AC drive does not support EtherNet/IP communication. • HEIP-10 communication configuration error • HEIP-10 hardware fault	• Check if the AC drive supports EtherNet/IP. • Correctly configure HEIP-10 communication parameters. • Replace the HEIP-10.
The AC drive reports a communication error during system operation.	• Abnormal communication data • Damaged or abnormal network cable connection • External interference	• Check if the EtherNet/IP master station program is normal. • Check if the network cable connection is normal; replace the cable. • Use shielded Cat5e twisted pair cables as required, ensure proper grounding of the HEIP-10, troubleshoot interference causes, and seek technical support.

When a communication card fault occurs, the specific cause can be identified through the AC drive. Related fault information is described in the following tables.

Table 13–2 Fieldbus adapter A fault subcodes and solutions

Fault Code	Cause	Solution
67-01	Connection timeout	Check if the network cable is properly connected; check if the master station is running.
67-05	Ethernet hardware error	Contact Inovance for technical support.
67-12	MAC not programmed	Contact Inovance for technical support.
67-13	IP address conflict	Check if another device has the same IP address as this device.
67-14	Link lost	Check if the network cable is properly connected.

Table 13–3 Fieldbus adapter B fault subcodes and solutions

Fault Code	Cause	Solution
68-01	Connection timeout	Check if the network cable is properly connected; check if the master station is running.
68-05	Ethernet hardware error	Contact Inovance for technical support.
68-12	MAC not programmed	Contact Inovance for technical support.
68-13	IP address conflict	Check if another device has the same IP address as this device.
68-14	Link lost	Check if the network cable is properly connected.

Copyright © Shenzhen Inovance Technology Co., Ltd.

19012451A01

Shenzhen Inovance Technology Co., Ltd.

www.inovance.com

Add.: Inovance Headquarters Tower, High-tech Industrial Park,
Guanlan Street, Longhua New District,
Shenzhen 518000, P.R. China
Tel: (0755) 2979 9595 Fax: (0755) 2961 9897

Suzhou Inovance Technology Co., Ltd.

www.inovance.com

Add.: No. 52, Tian E Dang Road, Wuzhong District, 215104,
Suzhou City, Jiangsu Province, P.R. China
Tel: (0512) 6637 6666 Fax: (0512) 6285 6720