MV800 Series Platform Drive

Communication Options User Manual

Document Version: V1.0
Archive Date: 2025/07/14
BOM Code:

Shenzhen Megmeet Electrical Co., Ltd. provides comprehensive technical support for our valued customers. Please contact your nearest Megmeet office or service center, or connect directly with Megmeet headquarters if any assistance is needed.

Address: 5th Floor, Block B, Unisplendour Information Harbor, Langshan Road, Nanshan District, Shenzhen, 518057, China

Website: www.megmeet.com

Tel: +86-755-86600500

Fax: +86-755-86600562

Service email: driveservice@megmeet.com

Foreword

Thank you for choosing Megmeet MV800 Platform Drive Communication Options.

The personnel responsible for installing and operating the options must undergo professional electrical training and safety training, be qualified through examinations, be adequately familiarized with the installation, commissioning, operation, and maintenance procedures and requirements of the product, and be able to prevent various emergency situations.

Before installing, removing, and operating the options, please carefully and thoroughly read the safety precaution sections of this manual and the drive user manual to ensure safe operation.

Any injuries or equipment damage caused by the user's failure to comply with the safety precautions stated in this manual and the drive user manual will not be the responsibility of our company.

Megmeet has conducted strict inspections during the manufacturing and packaging of the product. If any parts are found missing or damaged when unpacking the product, please contact our company or your supplier for further assistance. Due to our commitment to continuous improvement of our products, changes may occur to the provided information without prior notice.

Unpacking inspection

When unboxing the product, please make sure to verify the following:

- Check whether any damage occurred to this communication option
- Check whether the product is the right one via the bar code label on the PCB;
- Check whether the package contents are correct;
- Check whether the communication option and its user manual are included in the package;
- Contact the supplier immediately if any product damage or wrong models are found, or any of the package contents is found missing;
- Please request the option description document from our company;
- Refer to the environmental specifications for usage.

Item	Specifications
Operational temperature	-10 to 50℃
Storage temperature	-20 to 60℃
Humidity	5% to 95%
Other environmental requirements	Solar irradiation below 700 W/m^2; non-condensing, non-freezing, no direct exposure to rain, snow, or hail.
Air pressure	70 to 106 kPa

Item	Specifications
Vibration/Shock	5.9 m/s^2 (0.6g) for sine vibration of 9 to 200 Hz

Safety precautions

WARNING

- It is required to remove the drive enclosure/cover when installing/removing the option. Make sure the drive is completely powered off and the internal voltage is safe before installing/removing the option. Refer to the drive user manual for installing/removing procedures. Failure to comply with this requirement may result in serious injuries or death.
- When storing the option, make sure to safely place it in a location that is dust-proof, moisture-proof, free from electric shock, and without mechanical pressure.
- The expansion option is sensitive to static electricity. Adequate anti-static measures must be taken during operations to prevent damage to the components.

Contents

Chapter 1 EtherCAT Communication Option

1.1 Overview

Thank you for choosing Megmeet MV810-ECAT02 communication option. This manual provides information of the product functions, specifications, installation guidelines, basic operations, and settings, as well as an introduction to the EtherCAT network protocol. To ensure correct installation and operation of this product, please carefully read this manual and the communication protocol section of the drive user manual before using this communication option.

This manual serves as a guide for operating the MV810-ECAT02 communication option, and includes relevant instructions. Detailed information about the EtherCAT protocol is not included herein. If users would like to learn more about the EtherCAT protocol, please refer to the professional articles or reference materials.

This communication option is defined as a EtherCAT slave communication option that can be used with drives supporting EtherCAT communication.

This communication option supports two methods of reading and writing the process variables from the drive: one through PDO, and the other through SDO for reading and writing the object dictionary defined by the manufacturer.

1.2 EtherCAT communication option features

(1) Functions

- EtherCAT COE protocol supported
- Auto-configuration of the network address

(2) Services

- PDO and SDO
- Access to the drive parameters via SDO
- 100 Mbps full duplex
- Speed mode and torque mode
- SM mode, and DC mode with min. 1 ms cycle
- 4 groups of configurable PDO

(3) EtherCAT synchronous cycle

Item	Value
Synchronous cycle	8 ms
	4 ms

Item	Value
	2 ms
	1 ms

(4) Communication interface

EtherCAT adopts the standard RJ45 connector. This option offers two RJ45 interfaces that differ in transmission directions. The interface is shown in Figure 1-1. The IN terminal represents the EtherCAT input interface and the OUT terminal represents the output interface. The pin definitions of the interface are shown in Table 1-1.

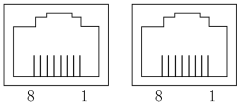


Figure 1-1 RJ45 interface

Table 1-1 RJ45 interface pin definitions

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

(5) Status indicator

This MV810-ECAT02 option has five LED indicators located on the PCBA of the expansion box and in the communication interface. LED indicators on the PCBA of the expansion box indicate the function status and the power supply status. The LED indicator of the communication interface indicates the communication connection status of MV810-ECAT02.

Description of the LED indicators on the PCBA of the expansion box:

LED1 (red) status	Description	Action
Steady on	Normal power supply for ECAT option	No need for actions
Off	No power supply for ECAT	Check whether the ECAT option is properly

LED1 (red) status	Description	Action
	option	connected to the drive

LED2 (green) status	Description	Action
Off	State machine in Init state	Check whether the ECAT option is properly connected to the host controller
Quickly flashing	State machine in Pre-OP state	Check whether the ECAT option is properly connected to the host controller
Slowly flashing	State machine in Safe-OP state	Check whether the ECAT option is properly connected to the host controller
Steady on	State machine in OP state	No need for actions

LED3 (green) status	Description	Action
Steady on	Normal reading/writing of function code by the master	No need for actions
0.5 s flashing	Function code reading/writing error by the master	Identify the cause by referring to the section 1.4.6 *Function code reading/writing fault*

LED4 (red) status	Description	Action
Off	Normal	No need for actions
Steady on	Communication timeout between the master and the option	Check whether the ECAT option is properly connected to the drive
0.5 s flashing	Abnormal ESC operation	Contact the manufacturer

Description of the LED indicator of the communication interface:

LED status	Description	Action
Yellow LED flashing	Normal connection, with data exchange	No need for actions
Green LED steady on	Normal connection	No need for actions
Yellow LED steady on	Normal connection, with no	Check whether the communication between the

LED status	Description	Action
	data exchange	EtherCAT master and slave is normal
Green LED off	Failure in connection	Check whether the network cable is properly connected

1.3 Electrical connection

EtherCAT network generally comprises one master station and multiple slave stations. The network can be structured into a bus type, star type, tree type, or a combination of several types, enabling flexible device connection and wiring. The bus-type network topology is shown in Figure 1-2.

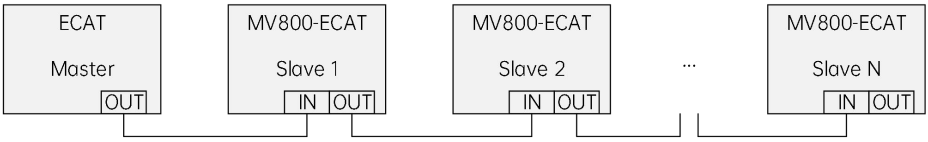


Figure 1-2 Bus-type network topology

1.4 EtherCAT communication

1.4.1 EtherCAT slave device description file

EtherCAT slave device description file (XML file) is used to configure the master and slave through reading by the master. The file contains the information necessary for the EtherCAT communication setting. Megmeet provides an "MV800_ECAT2_CoE_V1.00.xml" file for the EtherCAT communication option.

1.4.2 Function code setting

When using the MV810-ECAT02 to operate the MV800 drive, users need to set the operation command channel and the frequency source of the MV800 drive to the bus communication card, as shown in the following table.

Drive parameter	Value	Function description
P02.02	2	Set the operation command channel to communication control
P02.03	3	Set the communication command channel to EtherCAT

Drive parameter	Value	Function description
P02.05	8	Set the main frequency source to EtherCAT

1.4.3 PDO data mapping

MV810-ECAT02 communication option adopts the following PDO mapping by default (the maximal number of configurable data for 0x1600 and 0x1A00 is 10):

RxPDO (0x1600)	Control word (0x6040)	Target Position (0x210B)	Target Velocity (0x210D)	Target Torque (0x210C)			
TxPDO (0x1A00)	Status word (0x6041)	Error code (0x603F)	Output frequency (0x2202)	Output current (0x2200)	Output torque (0x2203)	Bus voltage (0x2204)	Position actual value (0x220A)

1.4.4 PDO data description

PDO enables real-time reading and modification of the drive data by the master, as well as the cyclic data exchange between the master and the drive. The PDO data description of the MV810-ECAT02 option is shown in the following table. For more detailed information, please refer to the XML file of the device.

RxPDO (sent by the master, and received by the slave):

Index/ Sub-index	Indication	Description	Value range	Access rights
6040h/0	Control word	Bit0: Forward running Bit1: Reverse running Bit2: Forward JOG Bit3: Reverse JOG Bit4: Stop Bit5: Coast to stop Bit6: Fault reset Bit7: Emergency stop	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read/Write
2100h/0	Drive torque upper limit	Unit: 0.1%	0.0 to 300.0% (corresponding to 0 to 3000)	Read/Write
2101h/0	Braking torque	Unit: 0.1%	0.0 to 300.0%	Read/Write

Index/ Sub-index	Indication	Description	Value range	Access rights
	upper limit			
2102h/0	FWD frequency upper limit	Unit: 0.01 Hz	0.00 to 599.00 Hz (corresponding to 0 to 59900)	Read/Write
2103h/0	REV frequency upper limit	Unit: 0.01 Hz	0.00 to 599.00 Hz	Read/Write
2104h/0	Voltage reference (V/F separation)	Unit: 1 V	0 to 1000 V	Read/Write
2105h/0	DO	Bit0: DO1 terminal Bit1: DO2 terminal Bit2: DO3 terminal Bit3: RO terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read/Write
2106h/0	AO1	Unit: 0.01%	0.00 to 100.00% (corresponding to 0 to 10000)	Read/Write
2107h/0	HDO1	Unit: 0.001 kHz	0.000 to 50.000 kHz (corresponding to 0 to 50000)	Read/Write
2108h/0	HDO2	Unit: 0.001 kHz	0.000 to 50.000 kHz	Read/Write
2109h/0	PID reference value	Unit: 0.1%	-100.0 to 100.0% (corresponding to -1000 to 1000)	Read/Write
210Ah/0	PID feedback value	Unit: 0.1%	-100.0 to 100.0%	Read/Write
210Bh/0	Position reference	Null	Null	Read/Write
210Ch/0	Torque reference	Unit: 0.1%	-300.0 to 300.0%	Read/Write
210Dh/0	Frequency reference	Unit: 0.01 Hz	0.00 to 599.00 Hz	Read/Write

TxPDO (sent by the slave, and received by the master):

Index/ Sub-index	Indication	Description	Value range	Access rights
6041h/0	Status word	Bit0: Forward running Bit1: Reverse running Bit2: Stop	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only

Index/ Sub-index	Indication	Description	Value range	Access rights
		Bit3: Fault Bit4: Power failure Bit5: Ready status Bit6: Motor number Bit7: Motor type Bit8: Overload pre-alarm Bit9 to Bit10: Control mode	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Not ready; 1: Ready 0: Motor 1; 1: Motor 2 0: Asynchronous motor; 1: Synchronous motor 0: Disabled; 1: Enabled 0: Keypad; 1: Terminal; 2: Communication	
2200h/0	Output current	Unit: 0.1 A	0.0 to 6553.5 A (corresponding to 0 to 65535)	Read only
2201h/0	Output voltage	Unit: 1 V	0 to 65535 V	Read only
2202h/0	Output frequency	Unit: 0.01 Hz	0.00 to 599.00 Hz (corresponding to 0 to 59900)	Read only
2203h/0	Output torque	Unit: 0.1%	-300.0 to 300.0% (corresponding to -3000 to 3000)	Read only
2204h/0	Bus voltage	Unit: 0.1 V	0.0 to 6553.5 V	Read only
2205h/0	DI state 1	Bit0: DI1 terminal Bit1: DI2 terminal Bit2: DI3 terminal Bit3: DI4 terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only
2206h/0	DI state 2	Bit0: DI5 terminal Bit1: DI6 terminal Bit2: DI7 terminal Bit3: DI8 terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only
2207h/0	DO state	Bit0: DO1 terminal Bit1: DO2 terminal Bit2: DO3 terminal Bit3: RO terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only

Index/ Sub-index	Indication	Description	Value range	Access rights
2208h/0	Motor power	Unit: 0.1%	-300.0 to 300.0% (corresponding to -3000 to 3000)	Read only
2209h/0	Power output	Unit: 0.1 kW	0.0 to 6553.5 kW (corresponding to 0 to 65535)	Read only
220Ah/0	Position actual value	Null	Null	Read only
603Fh/0	Error code	For detailed information, please refer to *MV810 High-Performance Vector Control Drive User Manual*	0: No error generated 1: Overcurrent during acceleration (OC1) 2: Overcurrent during deceleration (OC2) 3: Overcurrent during operation at constant speed (OC3) 4: Overvoltage during acceleration (OV1) 5: Overvoltage during deceleration (OV2) 6: Overvoltage during operation at constant speed (OV3) 7: Undervoltage fault (Uv) 8: Input phase loss (SPI) 9: Output phase loss (SPO) 10: Power module protection (drv) 11: Inverter overheat (OH1) 12: Rectifier bridge overheat (OH2) 13: AC drive overload (OL1) 14: Motor overload (OL2) 15: External fault (EF) 16: EEPROM read/write fault	Read only

Index/ Sub-index	Indication	Description	Value range	Access rights
			(EEP) 17: 485 communication error (CE) 18: EtherCAT communication timeout (E-CAt) 19: Current detection error (ItE) 20: CANopen communication timeout (E-CAN) 21: PID feedback loss (FbL) 22: Reserved 23: Braking resistor overcurrent (brOC) 24: Auto-tuning fault (tUN) 25: Reserved 26: Profinet communication timeout (E-Pn) 27: I/O card communication timeout (E-Io) 28: Modbus TCP communication timeout (E-TCP) 29 to 32: Reserved 33: Short-to-ground fault (GdF) 34: Speed deviation fault (dEv) 35 to 38: Reserved 39: Motor overheat (OH3) 40: Reserved 41: 24 V power supply overload (24OL) 42 to 45: Reserved 46: Board-level communication error (bCE)	

Index/ Sub-index	Indication	Description	Value range	Access rights
			47: Reserved 48: BootLoader failure (bLt) 49: Power board software version mismatching (vEr) 50: Parameter upload and download timeout (UPdnE) 51: AI1 current input overcurrent (AIOC) 52: Reserved 53: Fan locked-rotor (FAn) 54: Pre-overload (POL1) 55: I/O card 24 V overload (IO-OL)	

1.4.5 SDO data description

EtherCAT mailbox data SDO is used to transmit non-cyclic data, including those of communication parameter settings, drive function code parameter settings, etc. This option enables function code reading/writing via SDO communication. The drive function code group (P00 to P98) is mapped to the section (0x2000 to 0x2062) in the EtherCAT object dictionary, with an addition of 1 to the number on the last digit of each function code to form its sub-index in the dictionary. For example:

Drive function code P02.05 is mapped to the main index 0x2002 in the object dictionary, with the sub-index 0x06;

Drive function code P03.07 is mapped to the main index 0x2003 in the object dictionary, with the sub-index 0x08.

1.4.6 Function code reading/writing fault

The object dictionary of the index 0x2064 indicates the fault of drive function code reading/writing by EtherCAT master station: the data corresponding to the sub-index 1 indicate the fault code, with high 8 bits indicating a writing error and low 8 bits indicating a reading error. The data corresponding to the sub-index 2 indicate the index of the function code with reading/writing errors. For example, 0x0200 indicates that there is an error of reading/writing the function code P02.00. Types of fault codes are shown below:

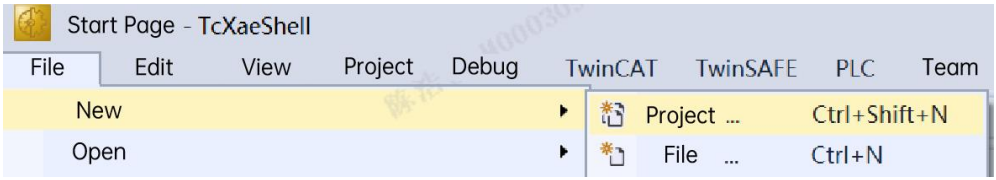
Fault	Fault code
Wrong password	0xF1
Index for operation does not exist	0xF4
Invalid parameter	0xF5
Parameter read only	0xF6
System lock	0xF7
EEPROM performing storage	0xF8

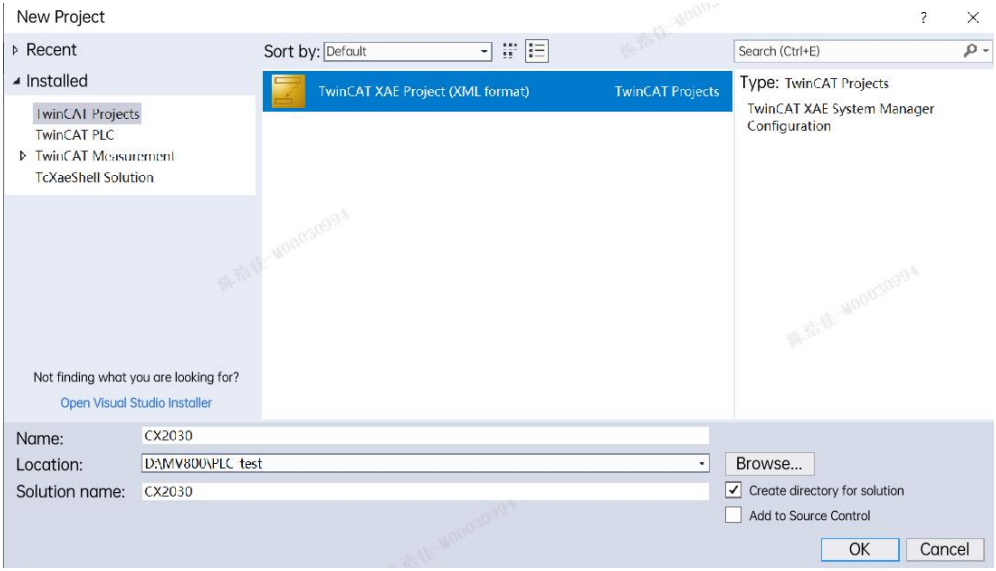
1.5 Communication example of Beckhoff PLC controlling MV800

The following is the demonstration of MV800 configuration process, taking Beckhoff CX2030 master as an example.

1.5.1 Create new project

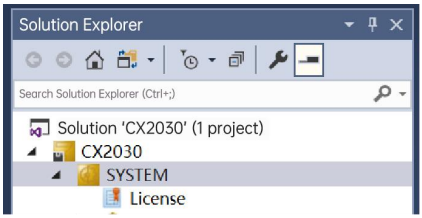
Open TwinCAT software, click "New" and "Project", choose "TwinCAT XAE Project (XML format)", edit "Name" and "Location", and click "OK".



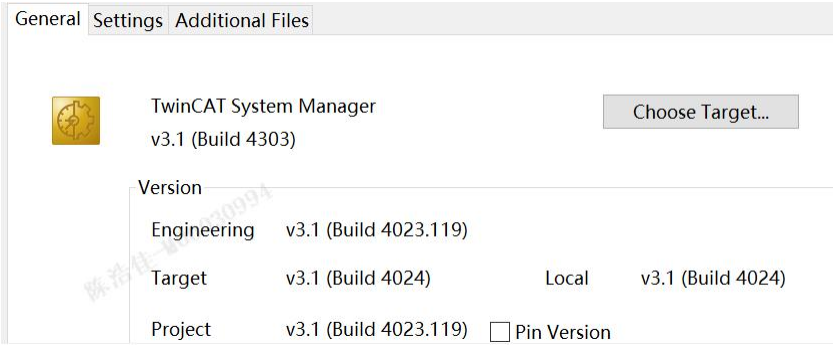


1.5.2 Scan Beckhoff PLC

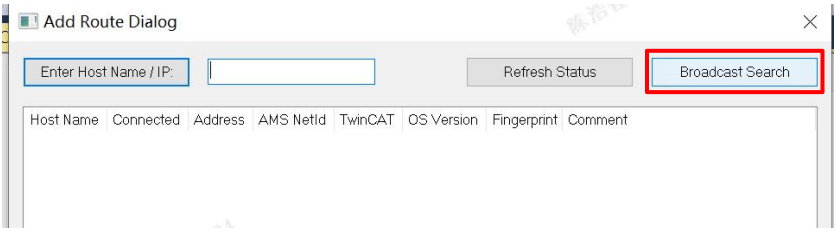
(1) Double click "SYSTEM" on the left project bar.



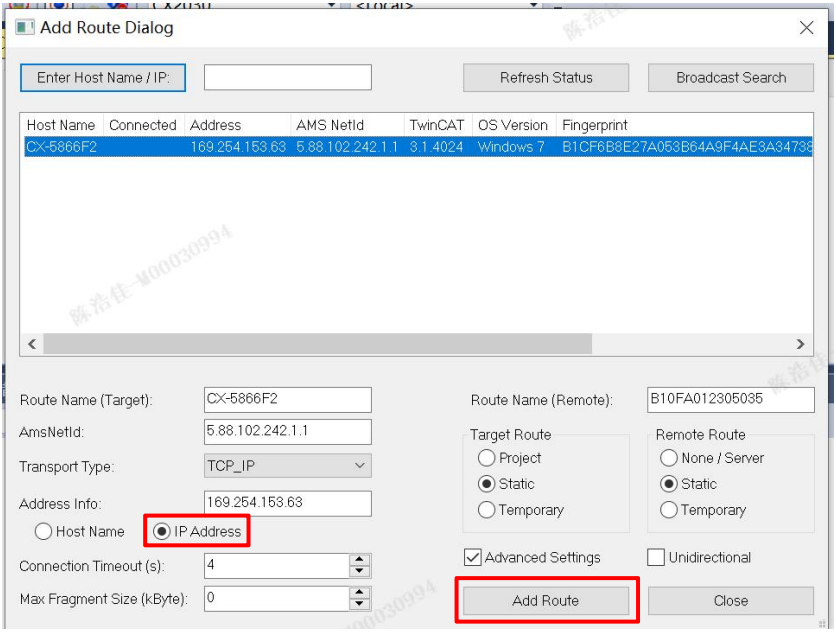
(2) Click "Choose Target..." in the pop-up window.



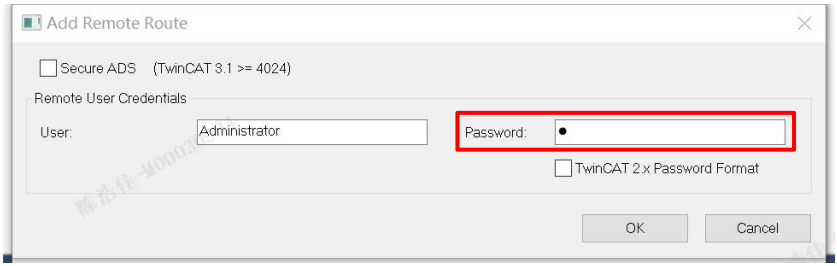
(3) Choose "Search (Ethernet)...", and click "Broadcast Search" to scan the PLC master.



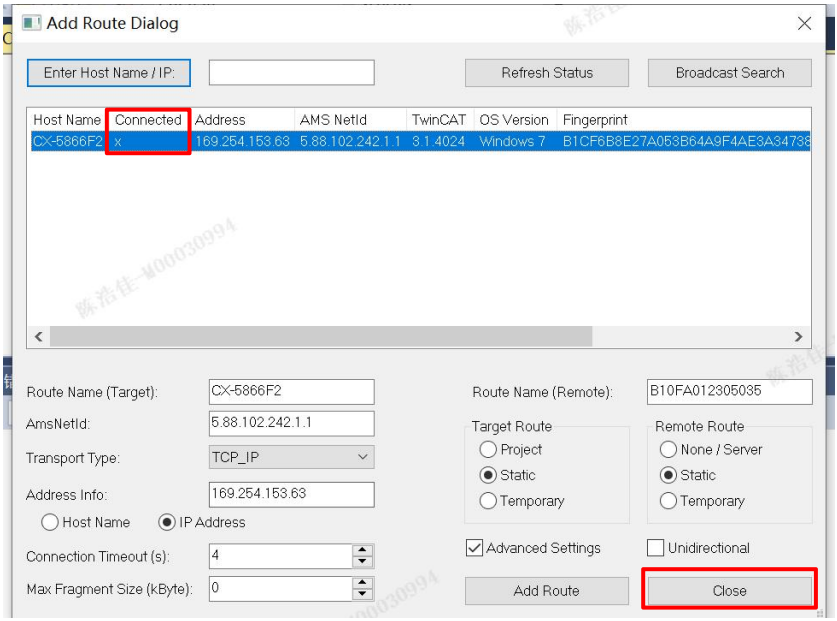
(4) When the scanning finishes (as shown in the following image), choose "IP Address" and click "Add Route".



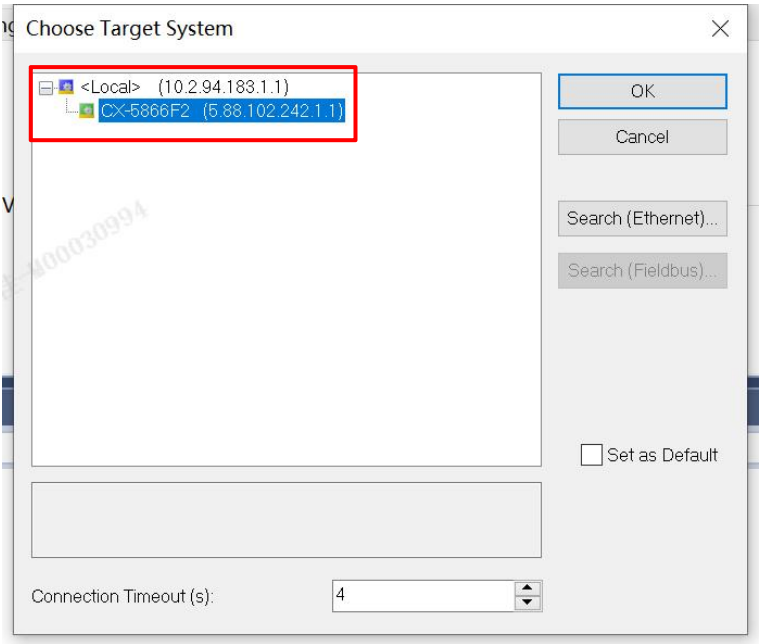
(5) Enter the password (password set as 1 by default for Beckhoff PLC), and click "OK".



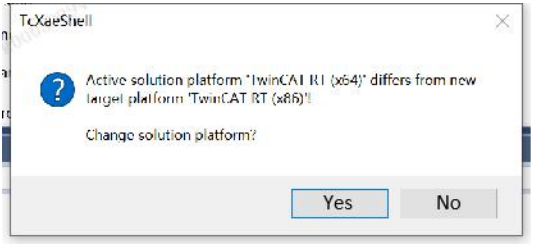
(6) The "x" symbol as shown in the following image indicates a successful connection with the PLC. Then choose "Close".



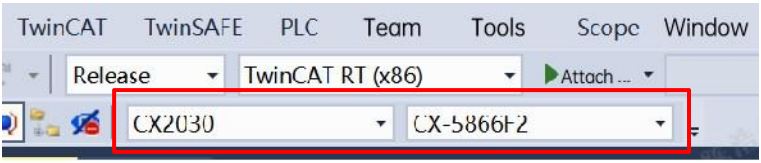
(7) Choose the connected PLC device, and click "OK".



(8) Click "Yes" in the pop-up window.

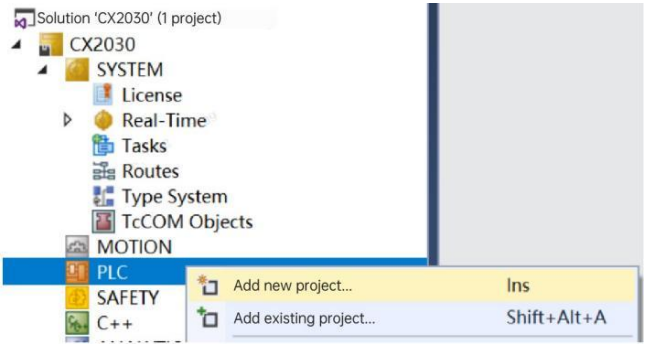


(9) When the PLC model shows on the status bar, it indicates that the communication between the PC and the PLC has been successfully established.

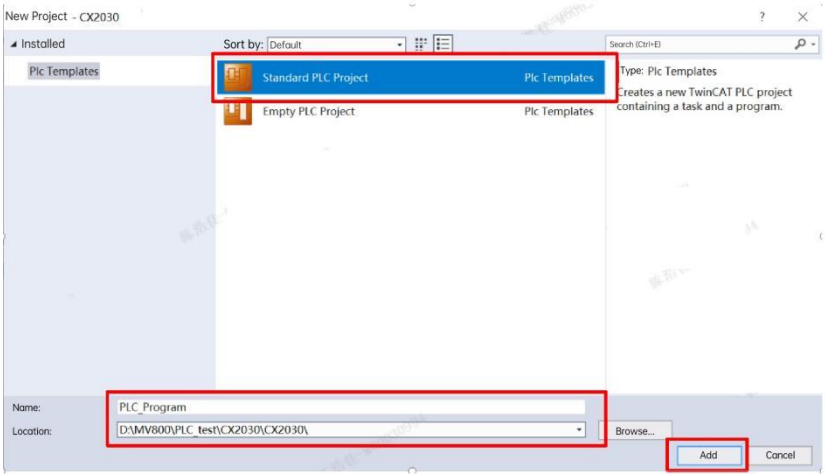


1.5.3 Add a PLC project

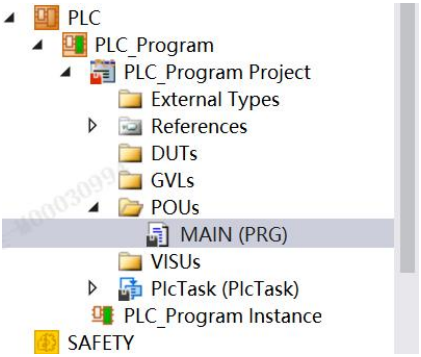
(1) Right-click "PLC" as shown in the following image, and choose "Add new project...".



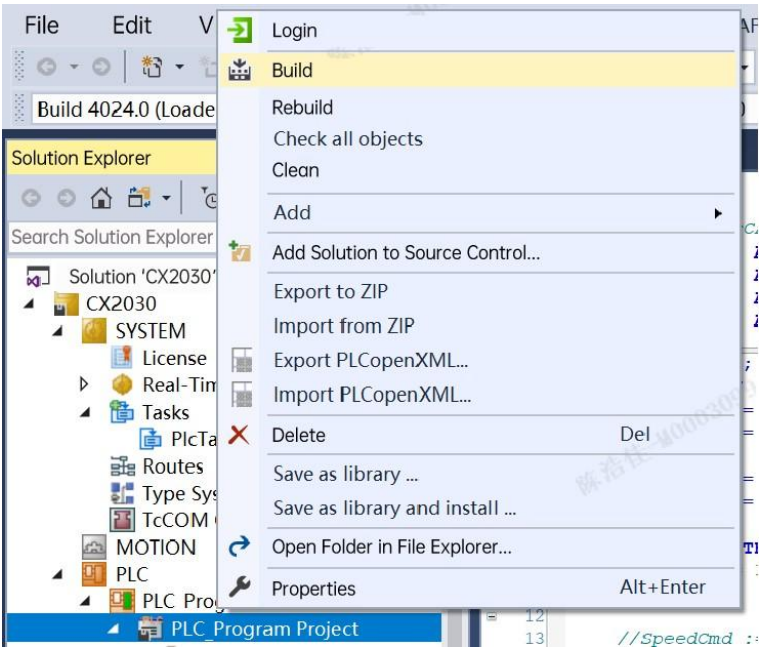
(2) Choose "Standard PLC Project", edit "Name" and "Location", and click "Add".



(3) Compile user's PLC program in "MAIN (PRG)".



(4) Click "Build" as shown in the following image to start PLC programming.

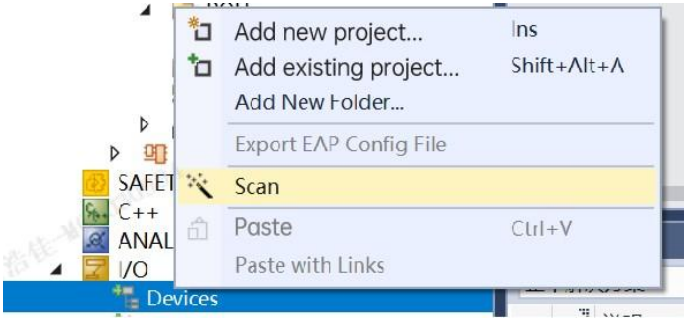


1.5.4 Scan the drive slave and configure PDO parameters

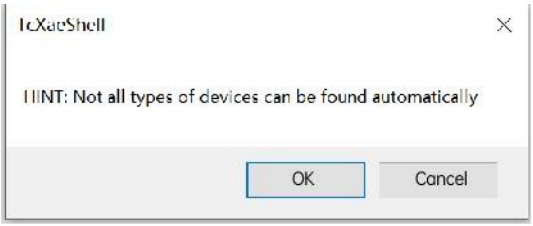
Before scanning the slave, it is required to copy the MV800 EtherCAT XML file into the TwinCAT installation directory.

- TwinCAT2 directory: TwinCAT\IO\EtherCAT
- TwinCAT3 directory: TwinCAT\3.1\config\IO\EtherCAT

Right-click "Devices", and choose "Scan".

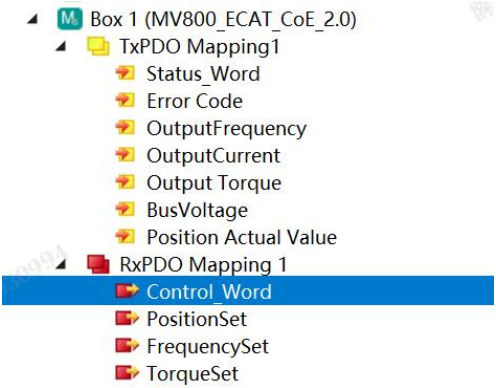


Click "OK".

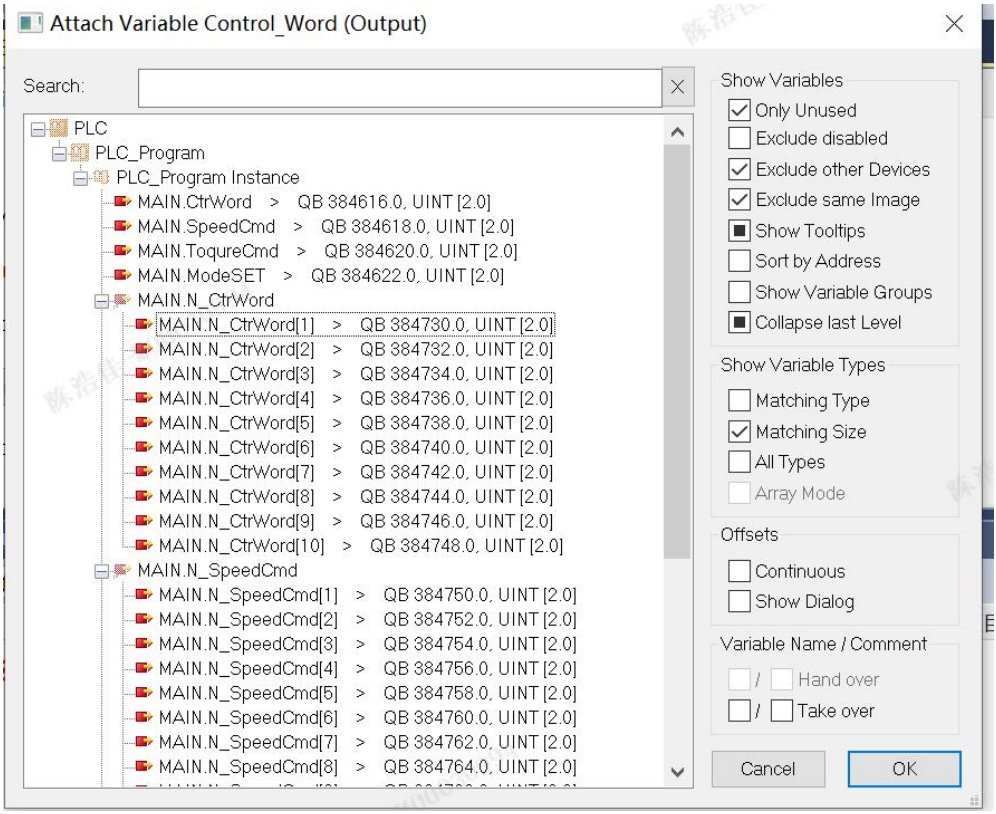


Click "Yes".

When the scanning finishes, double click the parameter in "TxPDO Mapping 1" and "RxPDO Mapping 1", taking "Control_Word" as an example, as shown in the following image.

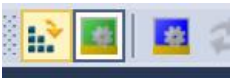


Establish the link between the MV800 drive PDO parameters and the PLC variables in the pop-up window.

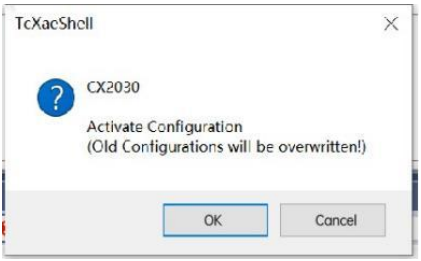


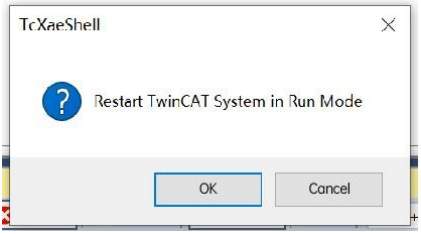
1.5.5 Download PLC program

Click "Activate Configuration".

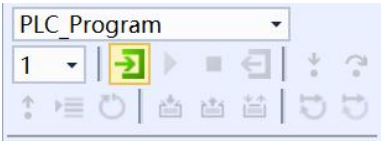


Click "OK".





Click the Login button as shown in the following image to finish the download of the PLC program.



Chapter 2 PROFINET Communication Option

2.1 Overview

Thank you for choosing Megmeet Profinet communication option. This manual provides information of the product functions, specifications, installation guidelines, basic operations, and settings, as well as an introduction to the network protocol. To ensure correct installation and operation of this product, please carefully read this manual and the communication protocol section of the drive user manual before using this communication option.

This manual serves as a guide for operating the PROFINET communication option and includes relevant instructions. Detailed information about the PROFINET protocol is not included herein. If users would like to learn more about the PROFINET protocol, please refer to the professional articles or reference materials.

This communication option is defined as a PROFINET slave communication option that can be used with drives supporting PROFINET communication.

This communication option supports linear bus topology and star-type topology.

This communication option supports 32 I/O, facilitating the operations of reading/writing drive process variables, reading drive state variables, and reading/writing drive function codes.

2.2 PROFINET communication option features

(1) Functions

- Supporting Profinet protocol and Profinet I/O device
- Two PROFINET I/O terminals supporting 100 M full duplex operation
- Linear bus topology and star-type topology

(2) Communication type

- Standard Ethernet channel: this channel is a non-real-time communication channel using TCP/IP protocol, applied mainly for the device parameterization, configuration, and diagnosis data reading.
- Real-time communication channel (RT): RT channel performs real-time communication via optimized communication channels, the priority of which is higher than that of TCP (UDP) / IP. It ensures data transmission between the stations in the same network within a defined time interval, which is strict in time requirement. The bus cycle reaches the millisecond level. This channel is used for transmitting process data, alarm data, etc.

(3) Communication interface

Profinet adopts the standard RJ45 connector. This option offers two RJ45 interfaces that do not differ in transmission directions and can be used interchangeably. The interface is shown in Figure 2-1, and the description is shown in Table 2-1.

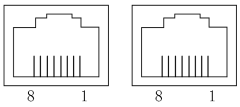


Figure 2-1 Two standard RJ45 interfaces

Table 2-1 RJ45 interface pin definitions

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

(4) Status indicator

This MV810-PNET02 option has LED indicators arranged at 3 locations: LED4 and LED5 on the light guide columns of the expansion box; LED1, LED2, and LED3 on the PCB; communication port LEDs. The LED description is shown below.

Protocol stack LED description:

Protocol stack LED	Color	Status	Function description
LED2	Red	Steady on	PROFINET diagnostic alarm with maintenance required or demanded.
		Off	No PROFINET diagnostic alarm with maintenance required or demanded pending.
LED3	Green	Off	PROFINET chip has not started correctly.
		Flashing	PROFINET chip is waiting for the synchronization of the host CPU (hardware start is complete).
		Steady on	PROFINET chip has started correctly.
LED4	Red	Steady on	PROFINET diagnosis exists.
		Off	No PROFINET diagnosis
LED5	Red	Steady on	No valid connection

Protocol stack LED	Color	Status	Function description
		Flashing	Ready for connection; no communication with the PROFINET I/O controller.
		Off	The PROFINET I/O controller has an active communication link to this PROFINET I/O device.

Communication port LED description:

LED status	Description	Action
Green light on	Normal connection	No need for actions
Green light off	No connection	Check cable connection
Yellow light flashing	Normal data communication	No need for actions
Yellow light steady on or off	No data communication	Check if there is communication between the master and the slave.

This MV810-PN03 option has LED indicators arranged at 5 locations: LED on the expansion box PCBA; communication port LEDs. The LED on the expansion box PCBA is used to display the function status and the power indication. The LED description is shown below.

Expansion box PCBA LED description:

LED4 status (red)	Description	Action
Off	Normal	No need for actions
On	Communication timeout between master and communication card	Check the connection between the PN option and the drive

Communication port LED description:

LED status	Description	Action
Yellow light flashing	Normal connection, with data transmission	No need for actions
Green light on	Normal connection	No need for actions
Yellow light on	Normal connection, without data transmission	Check whether there is communication between master and slave
Green light off	Connection failure	Check the cable connection

2.3 PROFINET network connection

PROFINET adopts the standard RJ45 connector. The network can be structured in linear or star type topology. The electrical wiring is shown in Figure 2-2 and Figure 2-3.

The applicable network cable types for electrical wiring include CAT5, CAT5e, and CAT6. It is required to use high-quality national-standard cables when the communication distance is over 50 m.

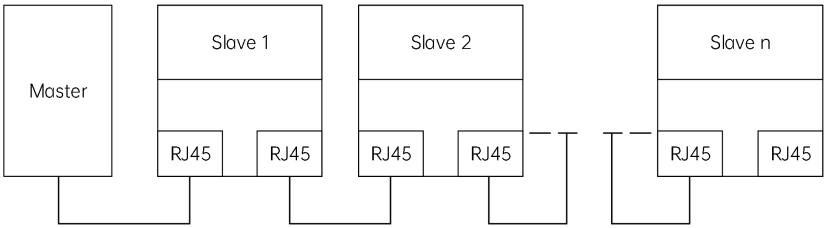


Figure 2-2 Electrical wiring in linear type topology

Note:

A PROFINET exchange should be provided by the user for star type network topology.

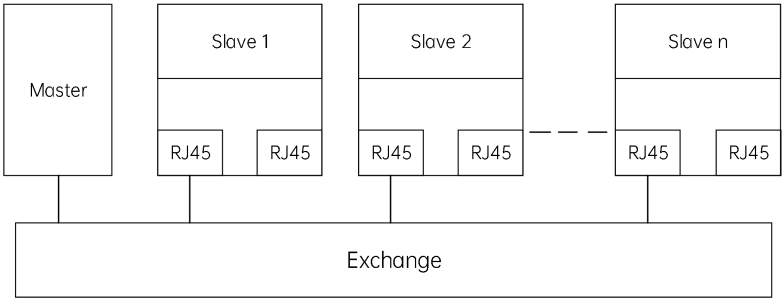


Figure 2-3 Electrical wiring in star type topology

2.4 Communication

2.4.1 Message format

Structure of RT frame (non-synchronous) is shown in Table 2-2.

Table 2-2 RT frame structure

Data head	Ethernet type	VLAN	Ethernet type	Frame identifier	RT user data	Cycle counter	Data status	Transmission status	FCS
	2-byte	2-byte	2-byte	2-byte	36 to 1440	2-byte	1-byte	1-byte	4-byte

					bytes				
	0x8100		0x8892						
	VLAN identifier					APDU status			
Data head									
7-byte preamble			1-byte synchronization		6-byte source MAC address			6-byte destination MAC address	

2.4.2 PROFINET I/O communication

This PROFINET communication option enables 16-word input/output. The message format for data transmission with the drive is shown in Fig. 2-4.

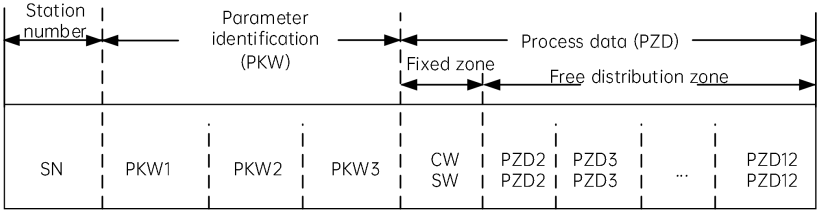


Figure 2-4 Message structure

The 32 I/O explained above facilitate the reference parameter setting and the status value detection of the drive, as well as sending the control command, inspecting the operational status, and reading/writing the function code parameters of the drive. Detailed information of these operations are provided below.

SN: Dedicated to the PN-485 conversion. For details, refer to Section 2.5.

Parameter area:

PKW1: Parameter identification

PKW2: Array index

PKW3: Parameter value

Process data:

CW: Control word (from the master to the slave, refer to Table 2-4)

SW: Status word (from the slave to the master, refer to Table 2-6)

PZD: Process data (designated by the user)

(The reference value is the output from the master to the slave, and the actual value is the input from the slave to the master)

PZD area (process data area): This area is designed to control and inspect the drive. The PZD received from the master and the slave is processed with the highest priority. The priority for processing PZD is higher than that for processing PKW. Only the latest valid data on the interface would be transmitted.

Control word (CW) and status word (SW):

Control word provides the basic method for the fieldbus system to control the drive device, which is sent by the fieldbus master station to the drive device, with the adaptor module functioning as the gateway. The drive device reacts on the basis of the bit code information in the control word, and sends the feedback information of status to the master via the status word.

Reference value: the drive device can receive control information via multiple channels, including analog input, digital input, drive control panel, and communication modules (for example, RS485, CH-PA01 adaptor module, etc.). To control the drive device via PROFINET, it is necessary to set the communication module as the controller of the drive device.

Actual value: actual value is a 16-bit word including the operational information of the drive device. The monitoring function is defined by the drive parameters. The conversion ratio of the integer of the actual value sent to the master differs among the functions. Refer to the drive user manual for more information.

Note:

The drive device always checks the bytes of the control word and the reference value.

SN station number (master to drive)

The SN station number area is defined by one word. The low byte (Byte0): Target station number (the drive station number visited by the PN master). The high byte (Byte1): Source station number (the drive station number with a PN option).

SN station number (drive to master)

The SN station number area is defined by one word. The low byte (Byte0): Target station number (the drive station number with a PN option). The high byte (Byte1): Source station number (the drive station number visited by the PN master).

PKW area

PKW area (parameter identification mark PKW1 - data area): PKW area describes the processing method of the parameter identification interface. PKW interface is defined as a mechanism instead of a physical interface. It determines the parameter transmission method between two communication partners, including parameter reading/writing.

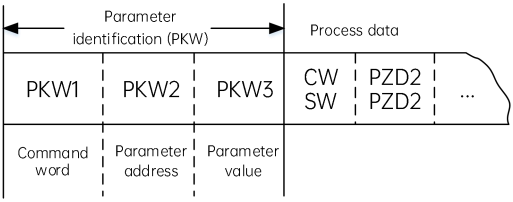


Figure 2-5 Parameter identification area

In cyclic communication, PKW area is composed of four 16-bit words. Definition of each word is shown in the table below:

First word PKW1 (16-bit)		
Bit 15 to 00	Command word or response	0 to 7
Second word PKW2 (16-bit)		
Bit 15 to 00	Basic parameter address	0 to 247
Third word PKW3 (16-bit)		
Bit 15 to 00	Error code of the parameter value or returned value	00

Note:

If the master requests a parameter value, the values in the message PKW3/PKW4 sent by the master to the drive will be invalidated.

Task request and response: when transmitting data to the slave, the master uses the request number, and the slave uses the response number as the positive or negative confirmation.

Table 2-3 Definition of task identification mark PKW1

Request (master to slave)	
Request	Function
0	No task
3	Read a parameter
6	Modify a parameter value (single word) [Modification of both RAM and EEPROM]
7	Modify a parameter value (single word) [Modification of RAM only]

Table 2-4 Definition of response identification mark PKW1

Response number (slave to master)	
Confirmation number	Function
0	No response
3	Read a parameter
6	Modify a parameter value (single word) [Modification of both RAM and EEPROM]
7	Modify a parameter value (single word) [Modification of RAM only]
0x83	Response error
0x86	

Response number (slave to master)	
Confirmation number	Function
0x87	

Table 2-5 Definition of task identification mark PKW3

Request (master to slave)	
Request	Function
Read	Number of parameters to be read (1 by default)
Write	Modify a parameter value

Table 2-6 Definition of response identification mark PKW3

Response number (slave to master)	
Command	Function
Read	Response of a parameter value, or response error
Write	Response of 0, or response error
Description of response error	1: Password error 2: Read/Write command error 3: CRC check error 4: Invalid parameter address 5: Invalid parameter value 6: Read-only parameter 7: System locked 8: Saving parameter

PZD area (master to drive)

Control word (CW): The first word in a PZD task message is a drive CW. Table 2-7 describes the CW of the MV800 series (Byte8: high-byte command word. Byte9: low-byte command word)

Table 2-7 MV800 series CW

Bit	Name	Value	Description
0 to 7	Forward run	0/1	0: Disable; 1: Enable
	Reverse run	0/1	0: Disable; 1: Enable
	Forward jog	0/1	0: Disable. 1: Enable
	Reverse jog	0/1	0: Disable; 1: Enable

Bit	Name	Value	Description
	Decelerate to stop	0/1	0: Disable; 1: Enable
	Coast to stop	0/1	0: Disable; 1: Enable
	Fault reset	0/1	0: Disable; 1: Enable
	Emergency stop	0/1	0: Disable; 1: Enable

Reference value (REF): The second to twelfth words in a PZD task message are the main reference values (REF). These 11 words (PZD2 to PZD12) serve to write the internal parameters of the drive. The parameters can be set via the function codes (P43.02 to P43.12). Table 2-5 provides the available settings of the MV800 series.

Table 2-8 Available settings for the MV800 series

Function code	Word	Range	Default
P43.02	PZD2 received	0: Disable 1: Frequency reference (0.00 to P02.10) 2: Drive torque upper limit reference (0.0 to 300.0% of the motor rated current) 3: Brake torque upper limit reference (0.0 to 300.0% of the motor rated current) 4: Torque reference (-300.0 to 300.0% of the motor rated current) 5: FWD run frequency reference upper limit (0.00 to P02.10) 6: REV run frequency reference upper limit (0.00 to P02.10) 7: Voltage reference (V/F separation) (0 to 1000) 8: Virtual input terminal command (the range of 0 to 0xFF corresponding to the range of DI8 to DI1) 9: Output terminal bus command (output terminal function set to 39; the range of 0 to 0xF corresponding to RO, DO3, DO2, and DO1) 10: AO1 output reference (0 to 100.0%) 11: HDO1 output reference (0 to 100.0%) 12: HDO2 output reference (0 to 100.0%) 13: PID reference (0.0 to 100.0%) 14: PID feedback (0.0 to 100.0%) 15 to 30: Reserved	0
P43.03	PZD3 received		0
P43.04	PZD4 received		0
P43.05	PZD5 received		0
P43.06	PZD6 received		0
P43.07	PZD7 received		0
P43.08	PZD8 received		0
P43.09	PZD9 received		0
P43.10	PZD10 received		0
P43.11	PZD11 received		0
P43.12	PZD12 received		0

PZD area (drive to master)

Status word (SW): The first word in a PZD response message is a drive SW. The definition of the drive SW is explained below: (Byte8: high-byte SW. Byte9: low-byte SW)

Table 2-9 MV800 series SW

Bit	Name	Value	Description
0 to 7	Forward running	0/1	0: Disable. 1: Enable
	Reverse running	0/1	0: Disable. 1: Enable
	Stop	0/1	0: Disable. 1: Enable
	Fault	0/1	0: Disable. 1: Enable
	Power failure	0/1	0: Disable. 1: Enable
	Ready state	0/1	0: Disable. 1: Enable
	Motor number	0/1	0: Motor 1. 1: Motor 2
	Motor type	0/1	0: Asynchronous. 1: Synchronous
8	Overload pre-alarm	1	0: Disable. 1: Enable
9 to 10	Control mode	0	Keypad
		1	Terminal
		2	Communication

Actual value (ACT): The second to twelfth words in a PZD task message are the main actual words ACT. These 11 words (PZD13 to PZD23) serve to read the drive internal parameters. The parameters can be set by the function codes (P43.02 to P43.12).

Table 2-10 MV800 series status value available for reading

Function code	Word	Range	Default
P43.13	PZD2 Transmitted	0: Disable 1: Frequency reference (0.01 Hz) 2: Ramp frequency reference (0.01 Hz) 3: Frequency output (0.01 Hz) 4: Voltage output (1 V) 5: Current output (0.1 A) 6: Bus voltage (0.1 V) 7: Motor power (0.1%)	0
P43.14	PZD3 Transmitted		0
P43.15	PZD4 Transmitted		0
P43.16	PZD5 Transmitted		0
P43.17	PZD6 Transmitted		0
P43.18	PZD7 Transmitted		0
P43.19	PZD8 Transmitted		0

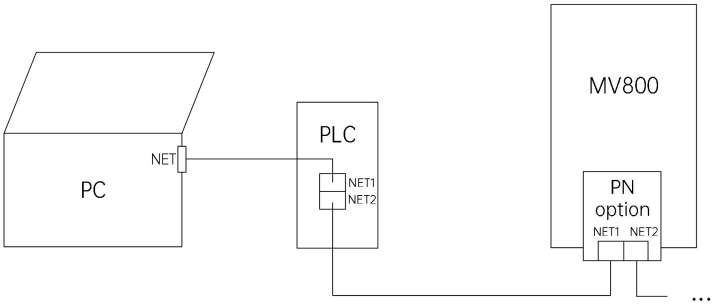
Function code	Word	Range	Default
P43.20	PZD9 Transmitted	8: Reserved 9: Excitation current (0.1 A) 10: Torque current (0.1 A) 11: Status word (0 to 0xFFFF) 12: Fault code (0 to 46) 13: DI1 to DI4 status (0 to 0xFFFF) 14: DI5 to DI8 status 15: DO status (0 to 0xF) 16: AI1 input voltage (0 to 10.00 V) 17: AI2 input voltage (-10.00 V to 10.00 V) 18: HDI input frequency (0 to 50.000 kHz) 19: AO output value (0 to 100.0%) 20: HDO1 output value (0 to 50.000 kHz) 21: HDO2 output value (0 to 50.000 kHz) 22: PID reference (-100.0% to 100.0%) 23: PID feedback (-100.0% to 100.0%) 24: PID deviation (-100.0% to 100.0%) 25: PID output (-100.0% to 100.0%)	0
P43.21	PZD10 Transmitted		0
P43.22	PZD11 Transmitted		0
P43.23	PZD12 Transmitted		0

2.5 Communication example of Siemens PLC controlling MV800

The following example is based upon a Siemens S7-1500 PLC as the master to demonstrate the configuration and usage of the MV800 PN communication.

2.5.1 Hardware connection

The Siemens S7-1500 PLC adopts two network interfaces, one of which serves to connect the computer for downloading the TIA PORTAL, with the other one for connecting the PN bus communication card of the drive. After the connection with the network is established, power on the PLC and the drive. The connection is illustrated below:



2.5.2 Establish the connection

For application with a PROFINET master, it is required to configure the GSDML file for the slave first. Add the corresponding slave device into the master system. Consult the agent or the manufacturer for acquisition of the GSDML file.

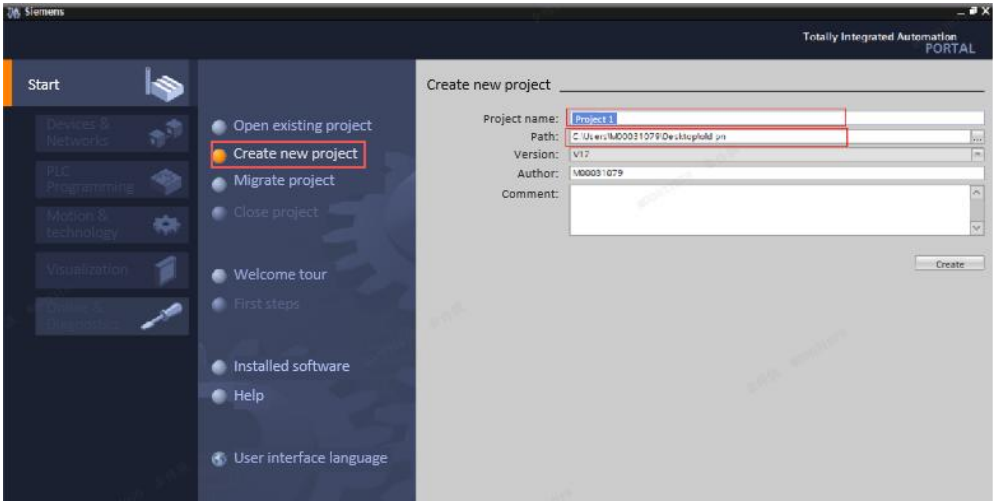
PN02 GSDML file: GSDML-V2.32-MEGMEET-MV800-20230830.xml

PN03 GSDML file: GSDML-V2.42-MEGMEET-MV800-PN03-20250102.xml

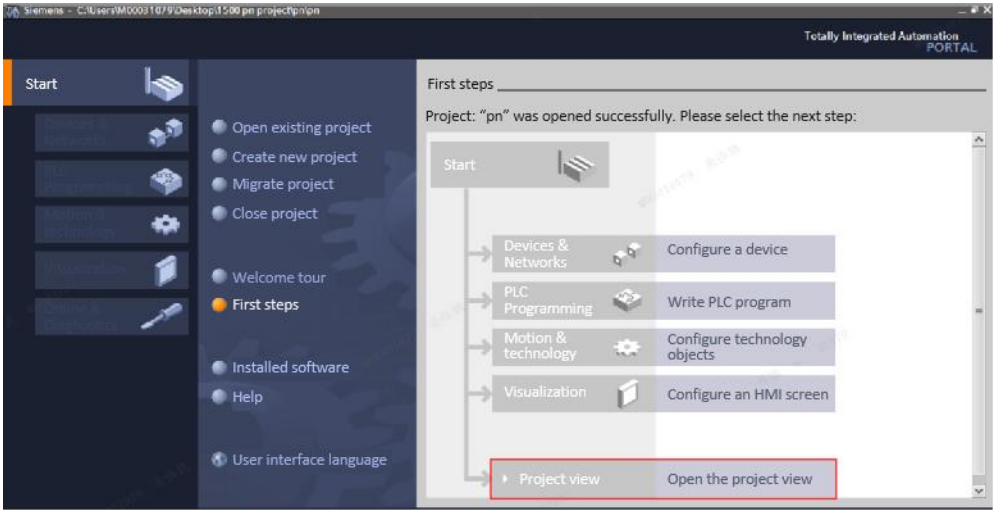
This section illustrates the creation of a communication project. The example is based on PN02, and the same procedure shall apply to PN03.

2.5.2.1 Start a new program

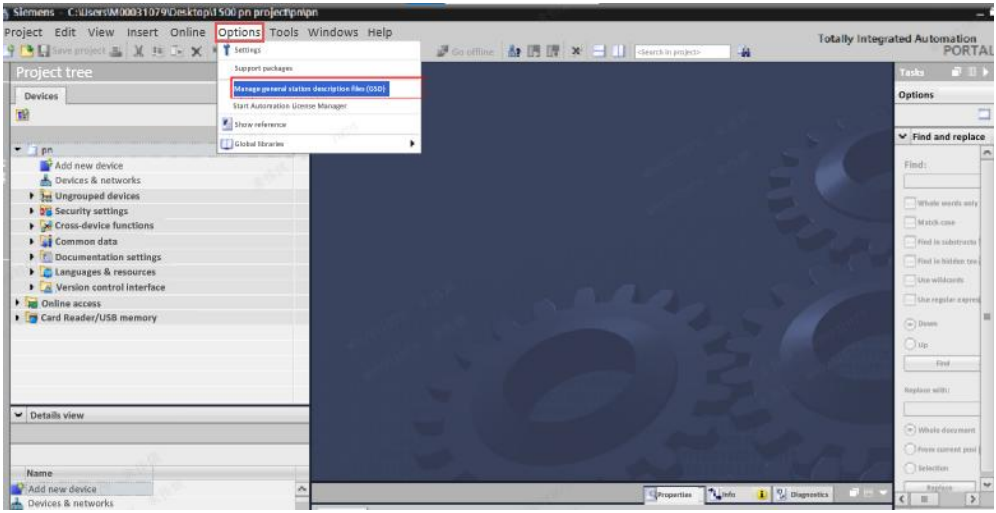
Open the TIA Portal software, start a new program, enter a program name, select a save path, and click the "Create" button.



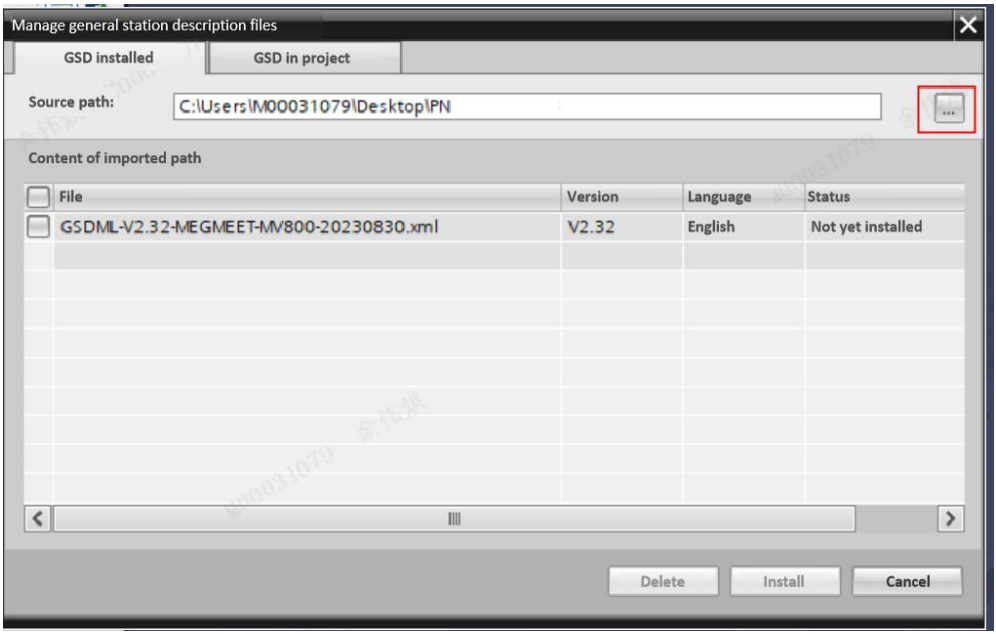
Click “Open the project view.”



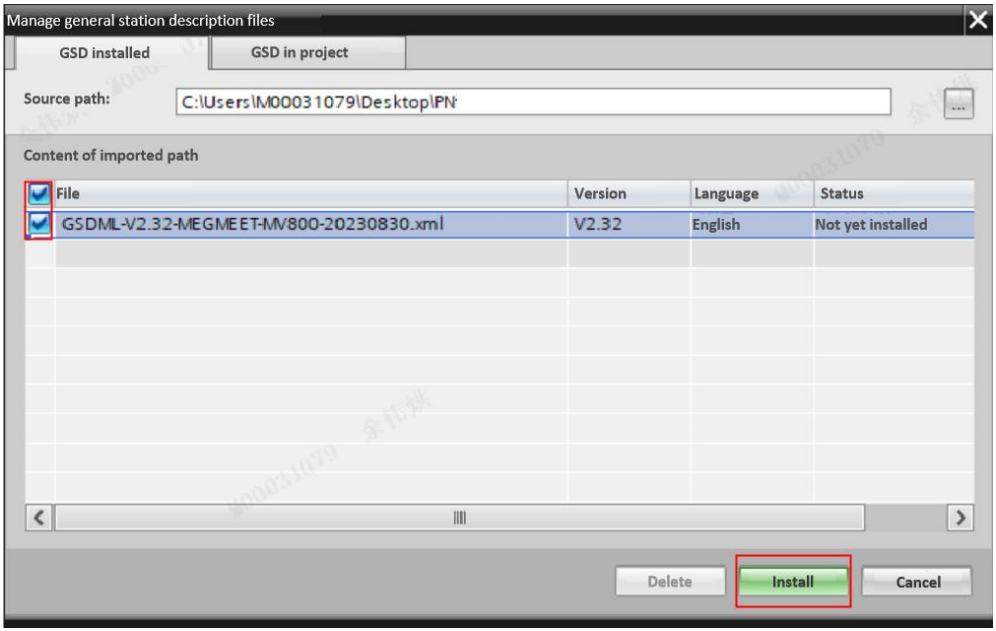
2.5.2.2 Install the GSD file

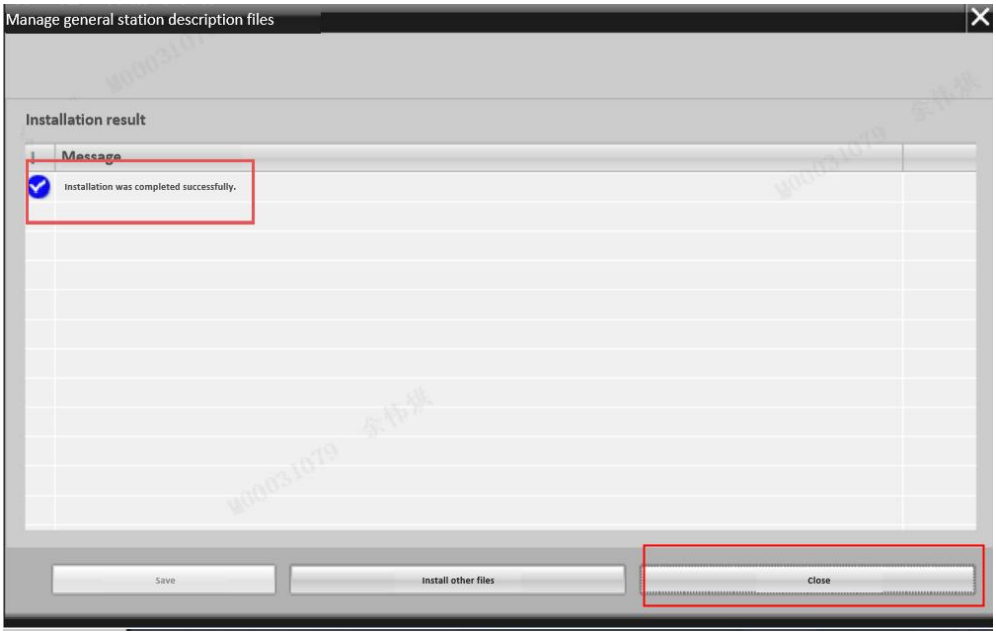


Select the save path for "GSDML-V2.32-MEGMEET-MV800-20230830.xml."



Select the file, and click the "Install" button.



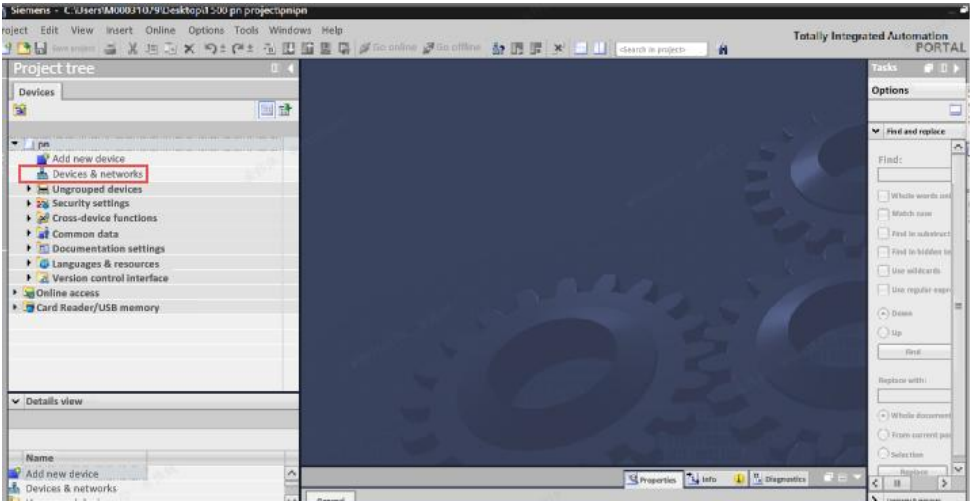


After a few seconds, the system prompts "Installation was completed successfully." Click the "Close"

button.

2.5.2.3 Configuration

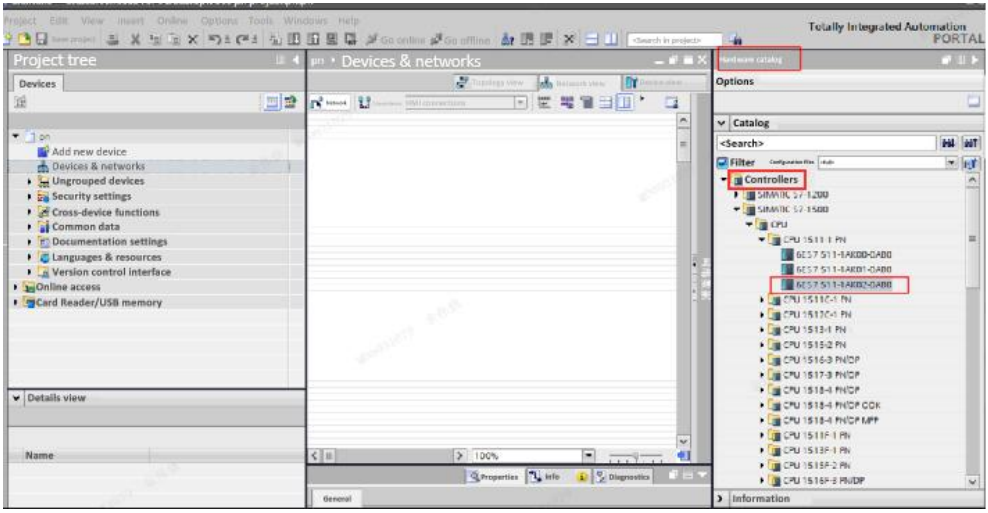
Click the "Devices & networks" button.



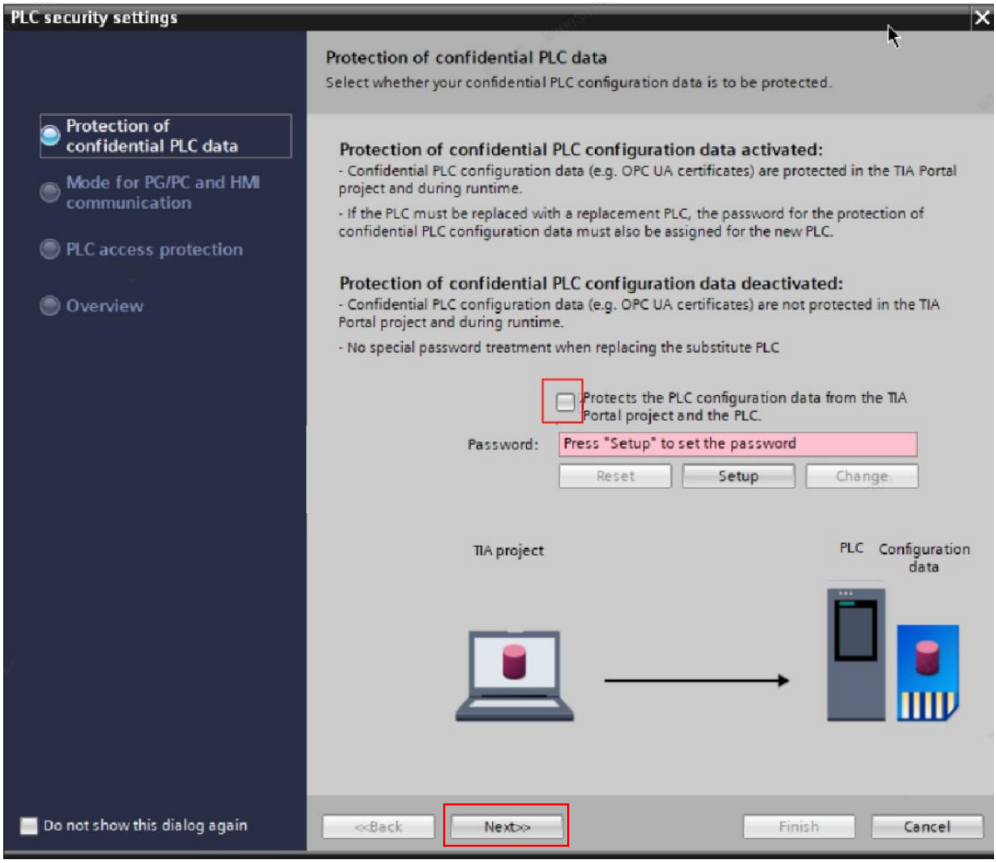
In the "Hardware catalog" on the right sidebar, select the PLC model in actual use. For this example, "1511-1-PN" is used in configuration. Double click the "6ES7 511-1AK02-0AB0" icon.

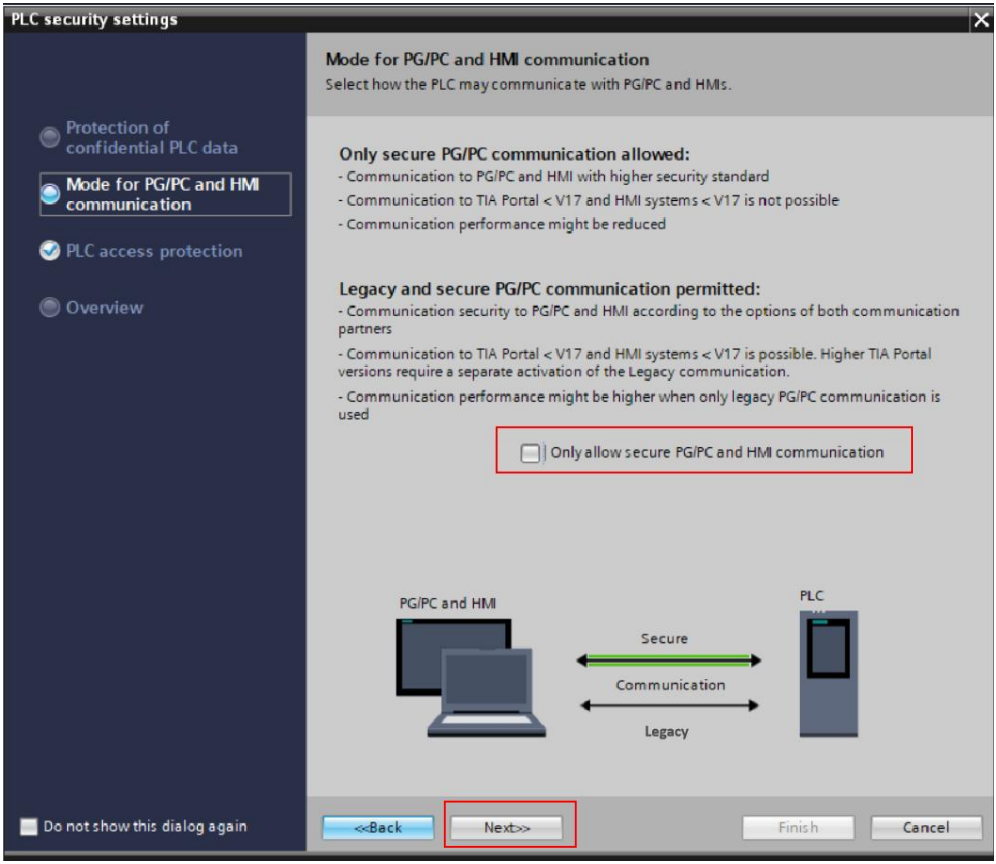
Note:

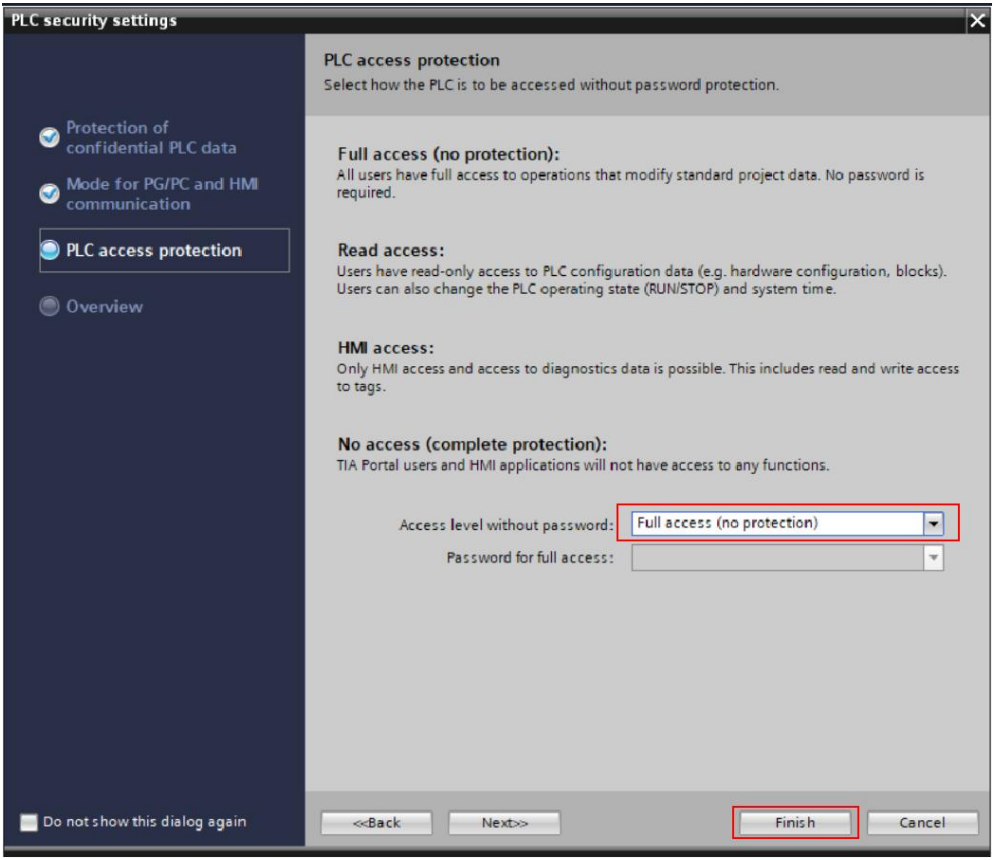
The number combination of the icon shall be identical with that on the back of the PLC unit. Otherwise, it will report an error when downloading the program to the PLC in the following process.



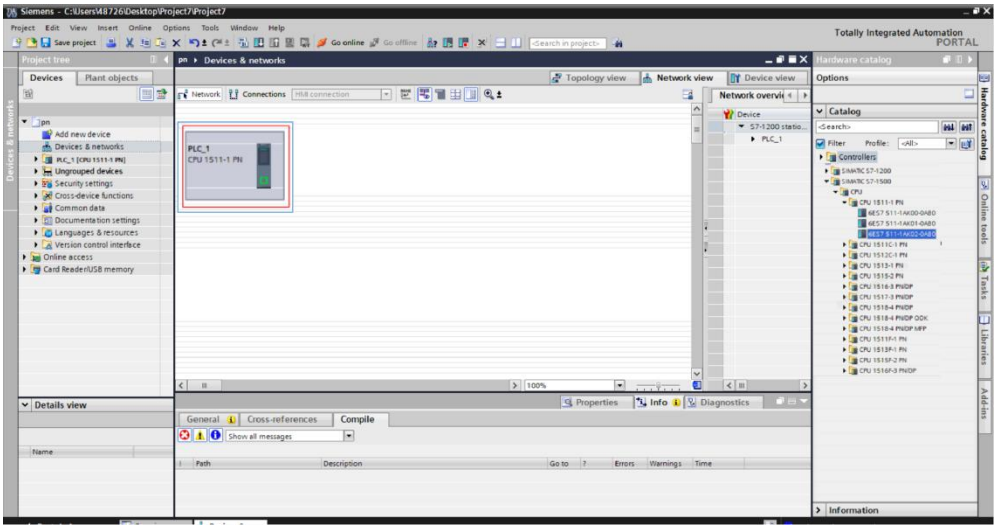
Set the data privacy rights as needed. The following steps of the example are based upon full access open for all users.



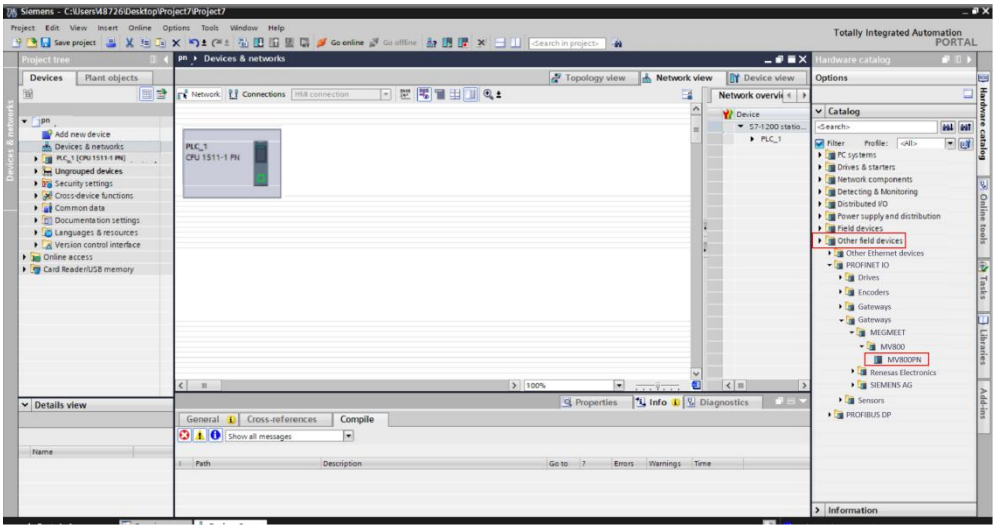

PLC security settings
Protection of confidential PLC data
Mode for PG/PC and HMI communication
PLC access protection
Overview
Mode for PG/PC and HMI communication
Select how the PLC may communicate with PG/PC and HMIs.
Only secure PG/PC communication allowed:
- Communication to PG/PC and HMI with higher security standard
- Communication to TIA Portal < V17 and HMI systems < V17 is not possible
- Communication performance might be reduced
Legacy and secure PG/PC communication permitted:
- Communication security to PG/PC and HMI according to the options of both communication partners
- Communication to TIA Portal < V17 and HMI systems < V17 is possible. Higher TIA Portal versions require a separate activation of the Legacy communication.
- Communication performance might be higher when only legacy PG/PC communication is used
Only allow secure PG/PC and HMI communication
PG/PC and HMI
Secure
Communication
Legacy
PLC
Do not show this dialog again
<<Back
Next>>
Finish
Cancel

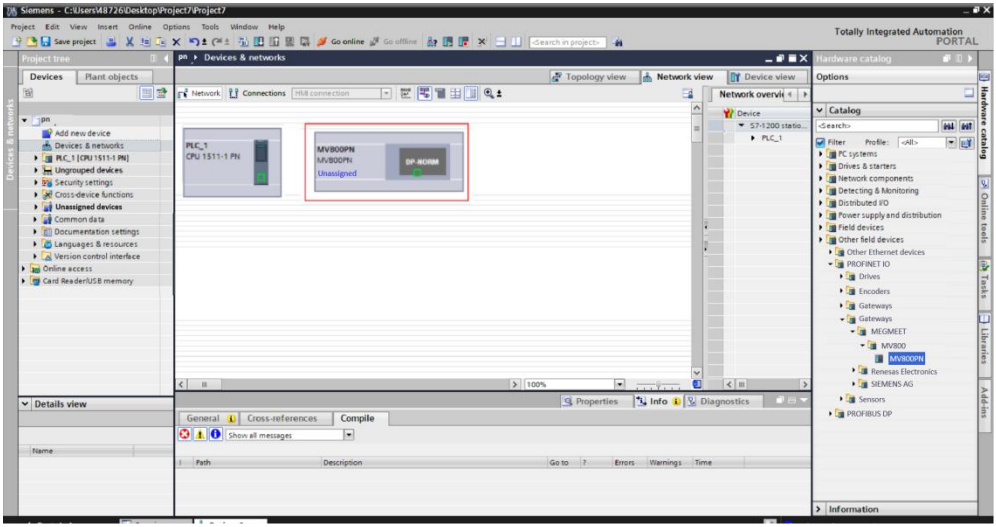

PLC security settings
Protection of confidential PLC data
Mode for PG/PC and HMI communication
PLC access protection
Overview
PLC access protection
Select how the PLC is to be accessed without password protection.
Full access (no protection):
All users have full access to operations that modify standard project data. No password is required.
Read access:
Users have read-only access to PLC configuration data (e.g. hardware configuration, blocks). Users can also change the PLC operating state (RUN/STOP) and system time.
HMI access:
Only HMI access and access to diagnostics data is possible. This includes read and write access to tags.
No access (complete protection):
TIA Portal users and HMI applications will not have access to any functions.
Access level without password:
Full access (no protection)
Password for full access:
Do not show this dialog again
<<Back
Next>>
Finish
Cancel

PLC is now successfully added to the network, as shown below:

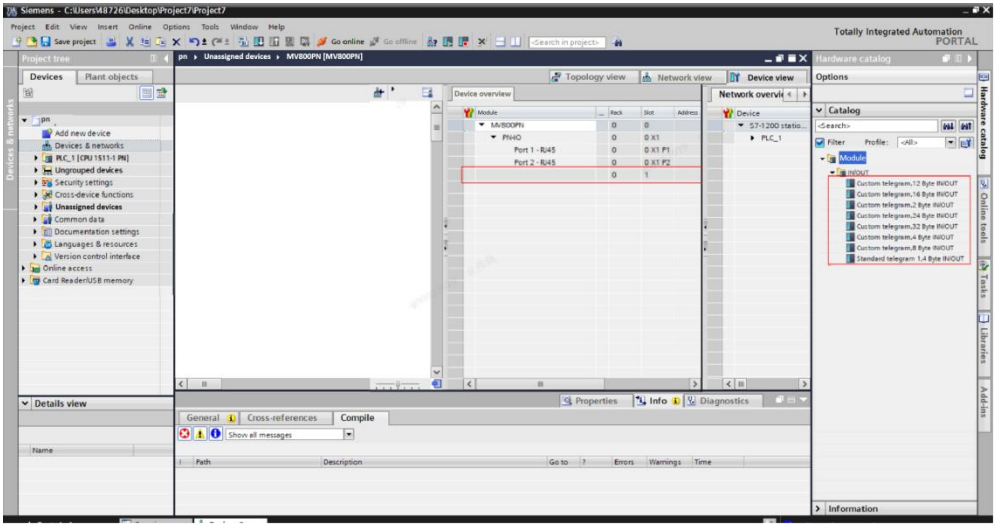


Double click “MV800PN” to add the slave to the network.





Double click the slave icon to enter the interface as shown below. Configure the slave in the interface.

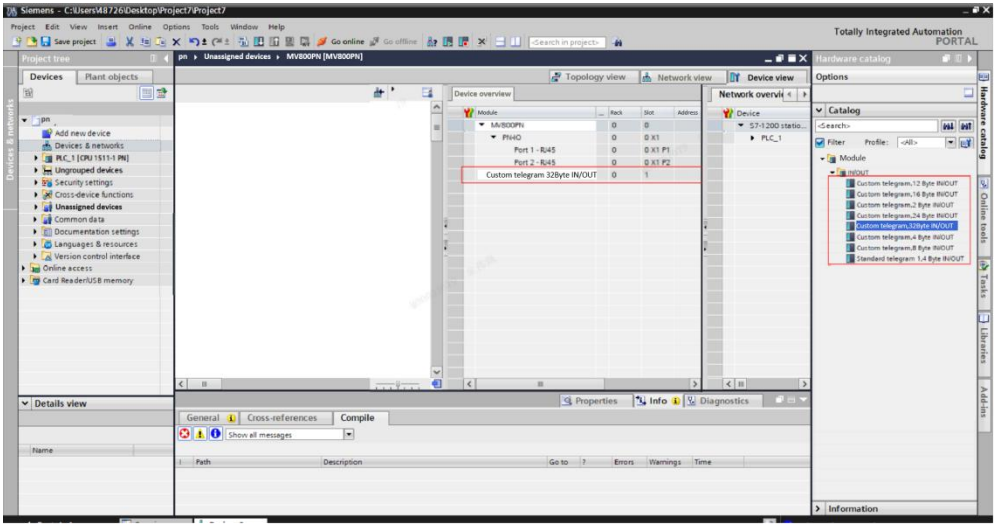


Custom telegram refers to the message customized by the user, and its length is available for configuration. The default selection is "Custom telegram 32Byte IN/OUT."

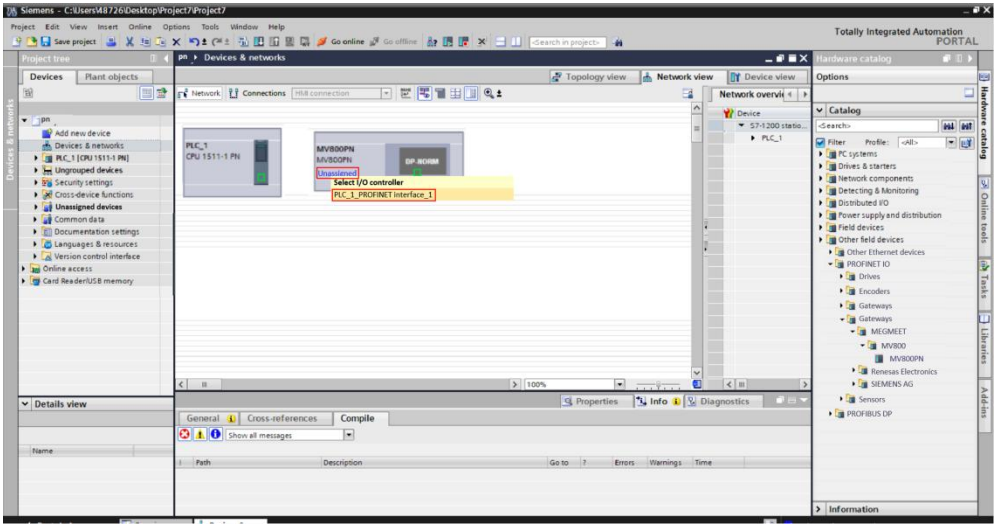
Standard telegram refers to the standard message 1.

This example is based upon the selection of "Custom telegram 32Byte IN/OUT." Double click the icon to add it into the slot.

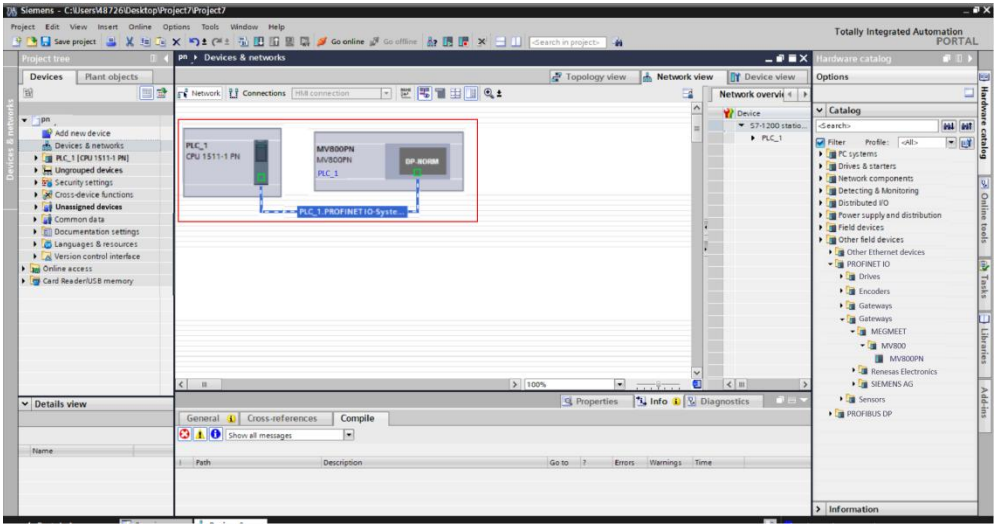
The display of the content in the red frame indicates that the "Custom telegram 32Byte IN/OUT" has been added to the slot.



Click “Devices & networks”, “Unassigned”, and “PLC_1_PROFINET interface_1” in sequence to enter the following interface.



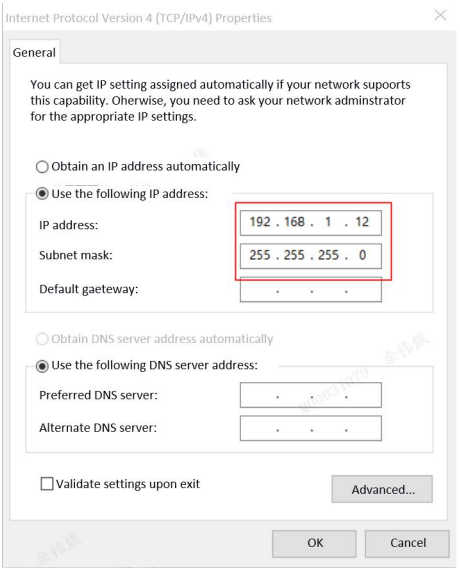
Configuration is completed.



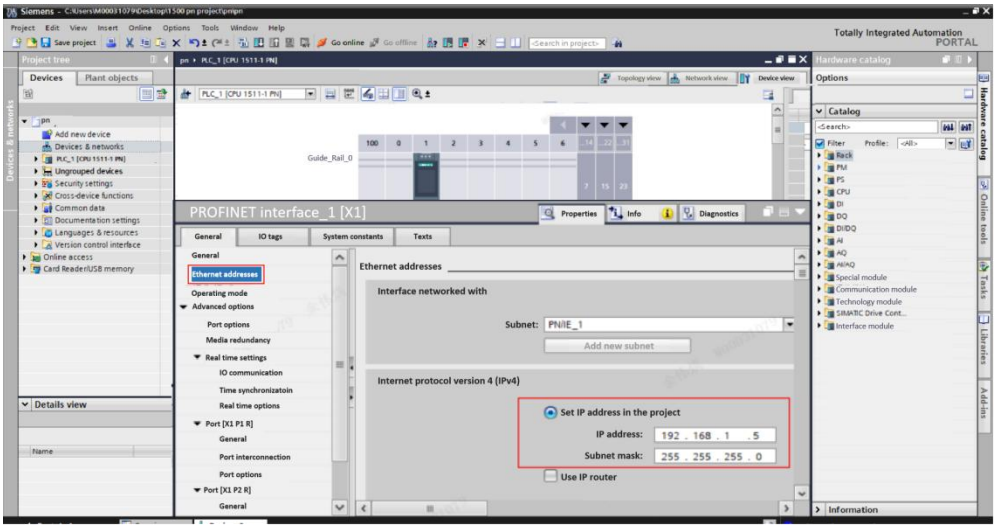
2.5.2.4 Set IP address

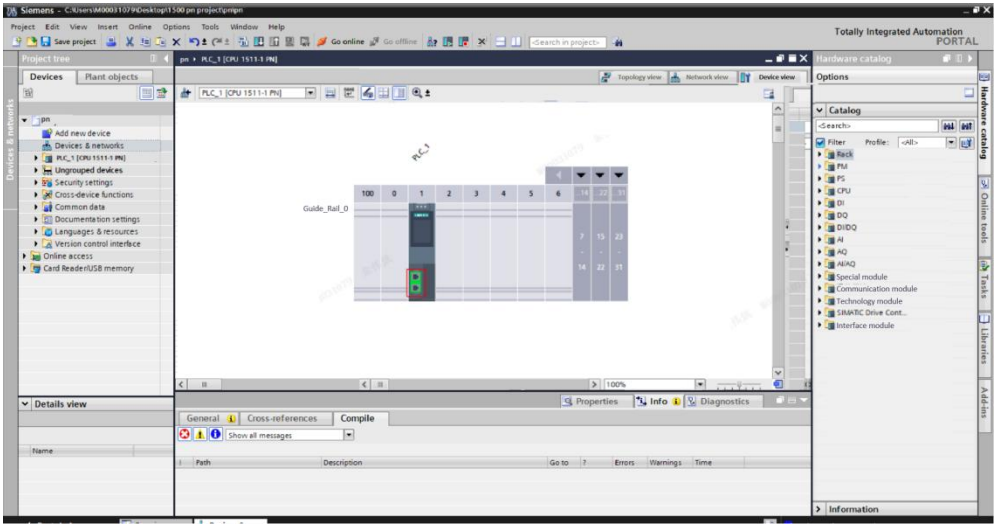
Set the IP address to ensure that the PLC address, slave address, and the PC address are in the same network segment.

Set the IP address of the PC to 192.168.1.12.

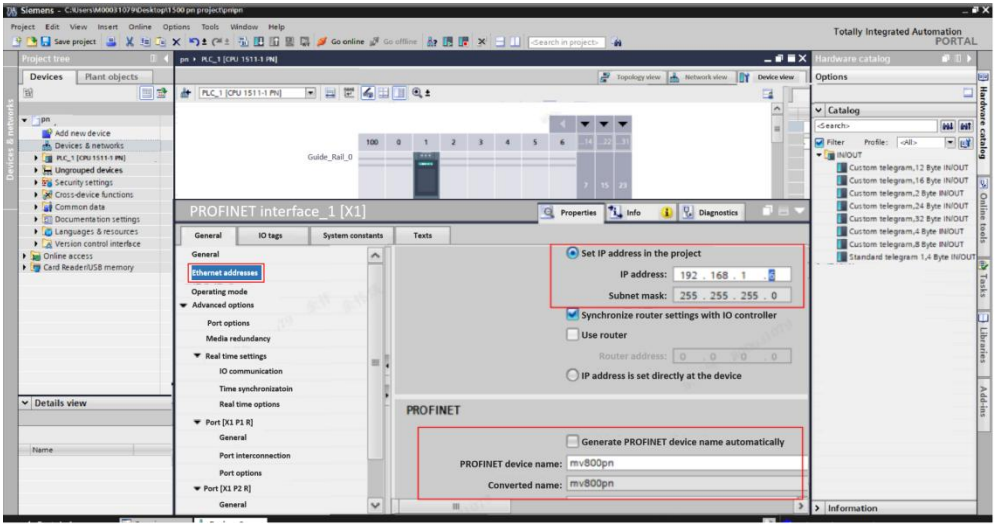


Click “Devices & networks”, and double click the PLC icon and the network interface. Set the IP address of PLC to 192.168.1.5.



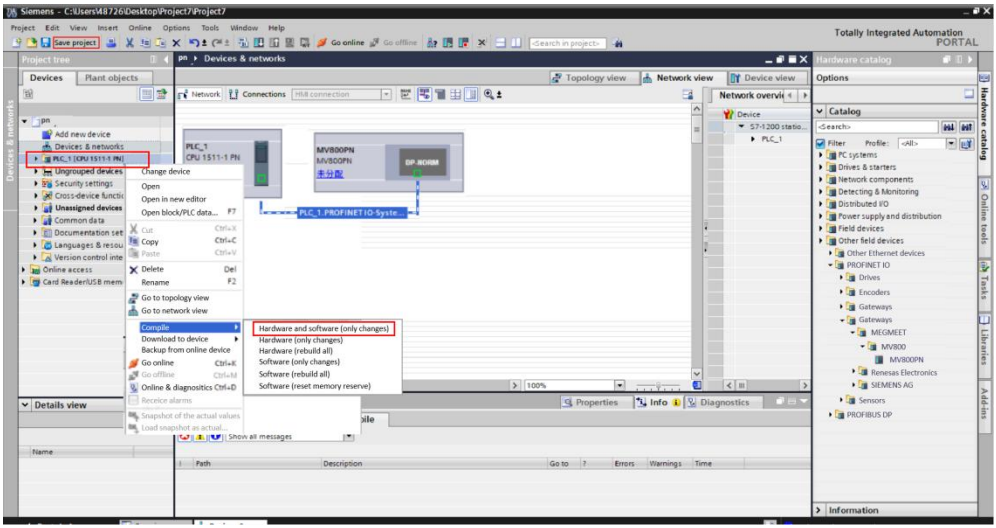


Double click “Devices & networks”, click the slave icon, and double click the network interface. Set the slave address to 192.168.1.6, and uncheck “Generate PROFINET device name automatically.” Enter the name “mv800pn.”

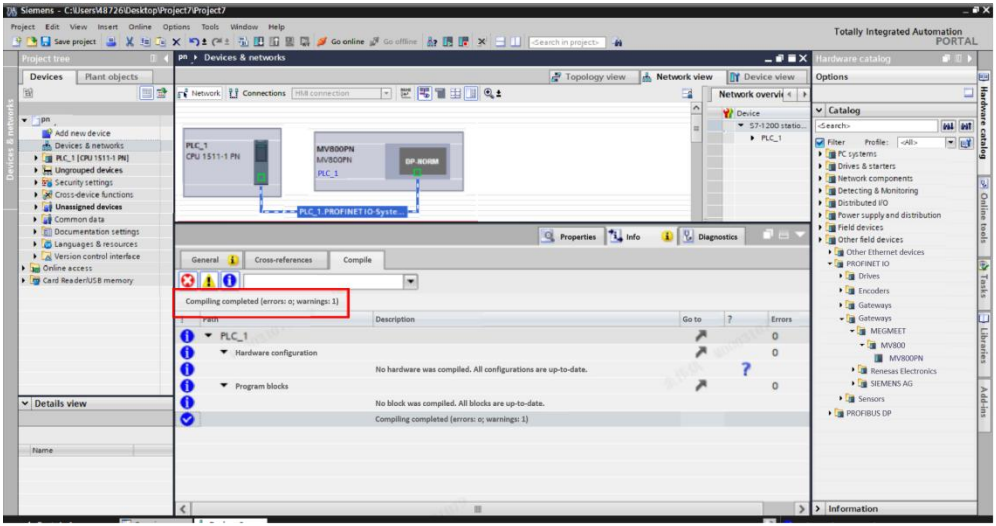


2.5.2.5 Compile and download

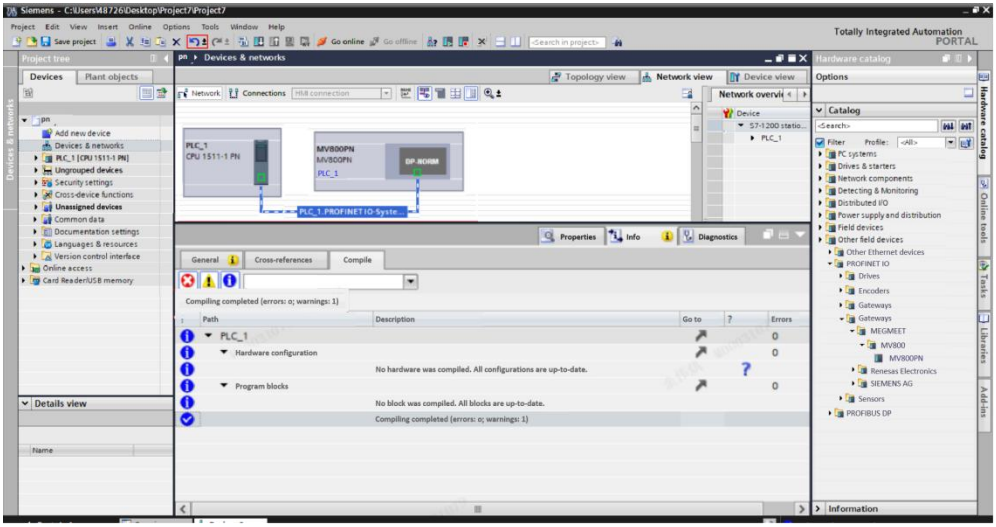
Click "Save project." Right click PLC_1, left click "Compile", "Hardware and software (only changes)" to compile the project.



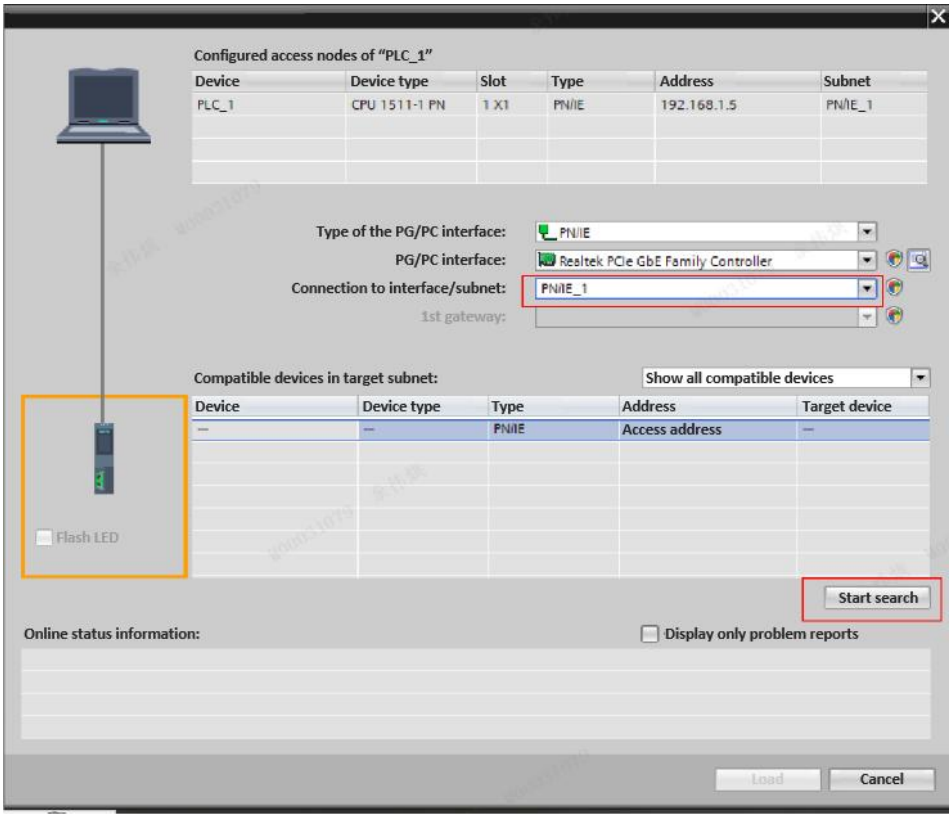
If the compilation error is 0, there is no wrong setting in the previous configuration. Download to compile.



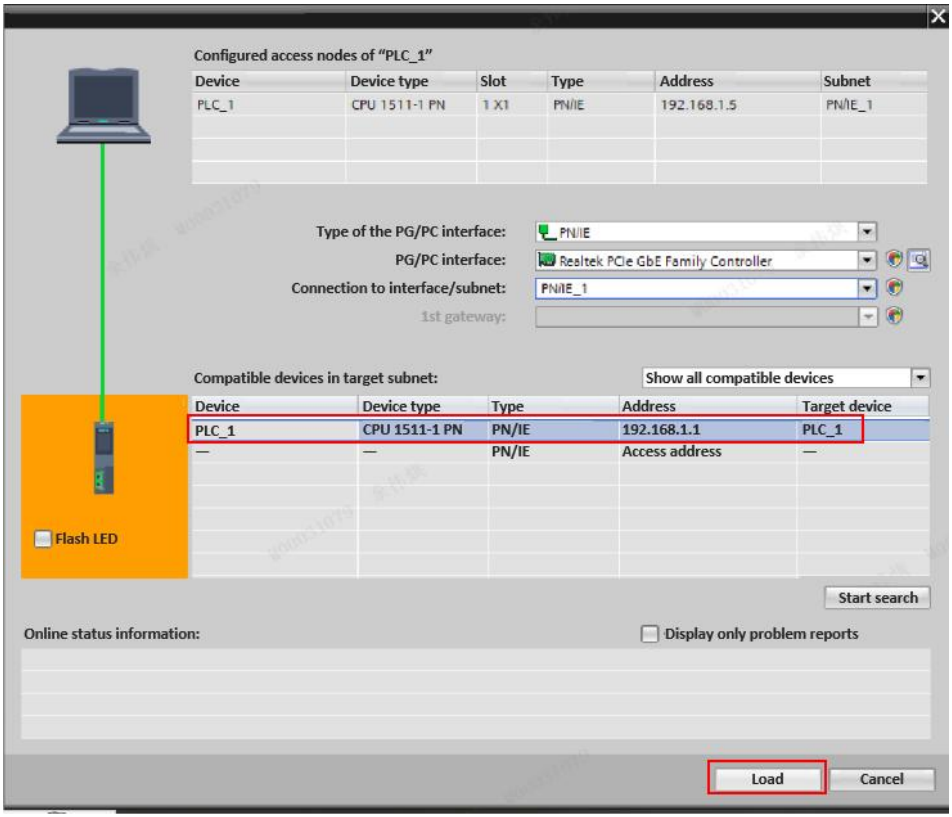
Click “Download.”



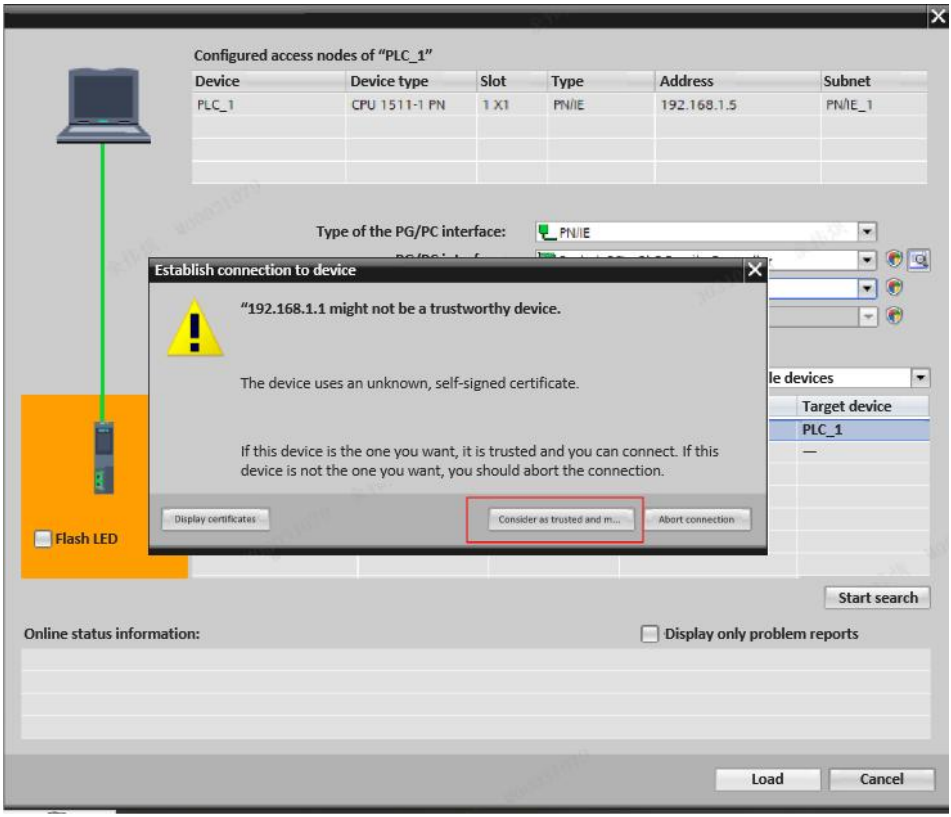
Select the option in the red frame as shown below, and click "Start search."

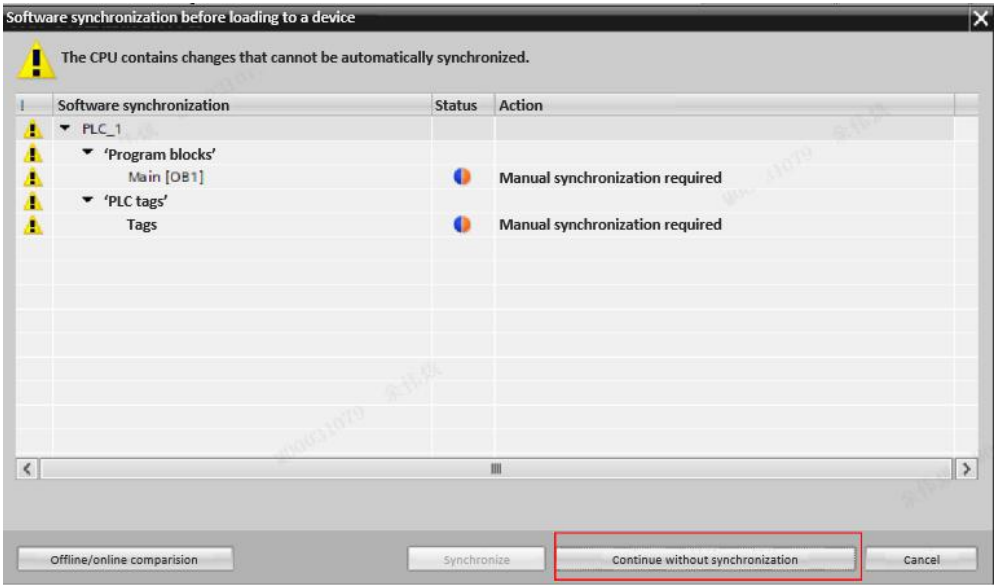


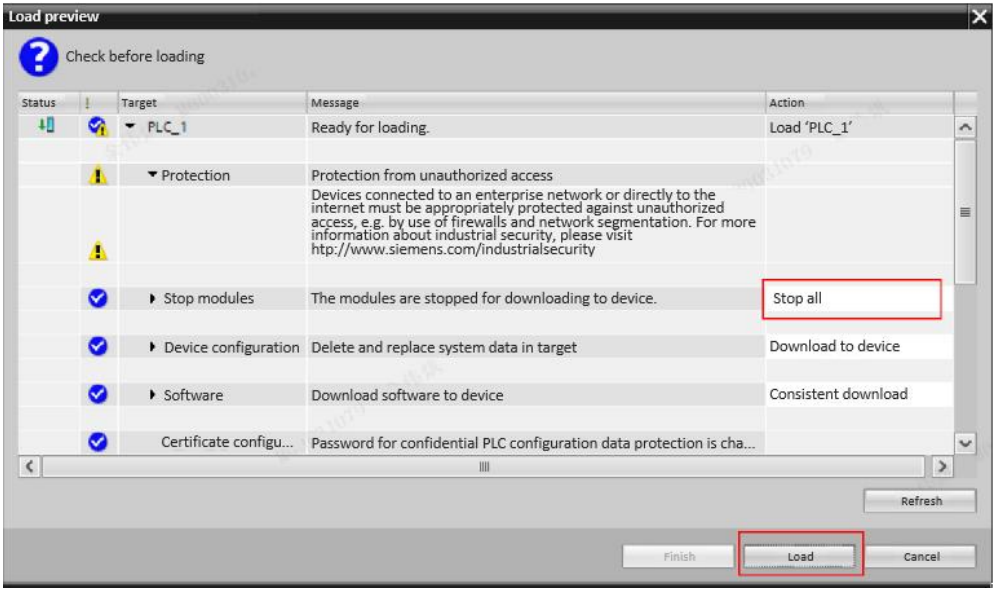
Select PLC in the search result table, and click "Load."



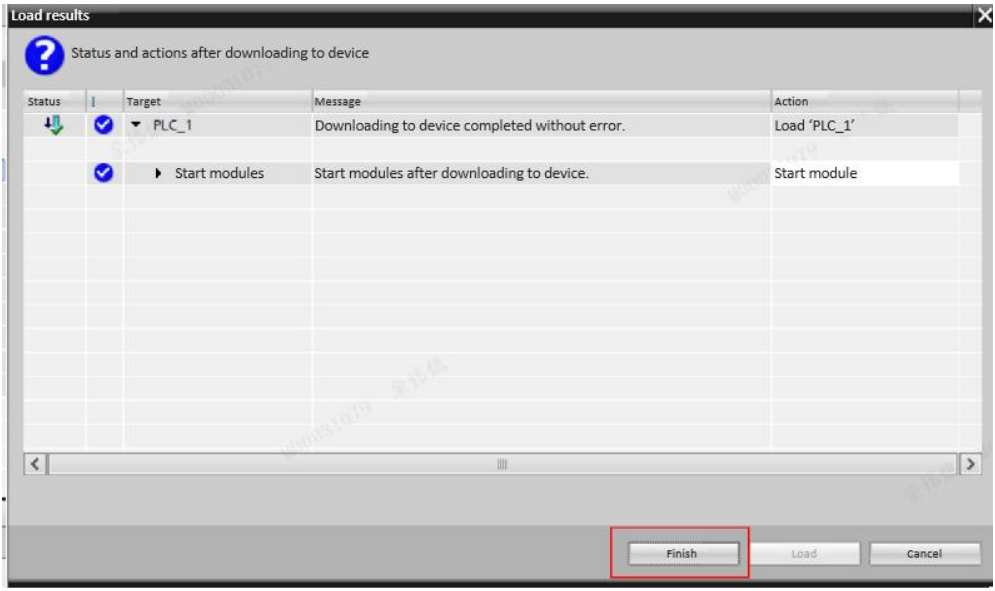
Follow the steps below.




Software synchronization before loading to a device
The CPU contains changes that cannot be automatically synchronized.
Software synchronization
Status
Action
PLC_1
'Program blocks'
Main [OB1]
Manual synchronization required
'PLC tags'
Tags
Manual synchronization required
Offline/online comparision
Synchronize
Continue without synchronization
Cancel

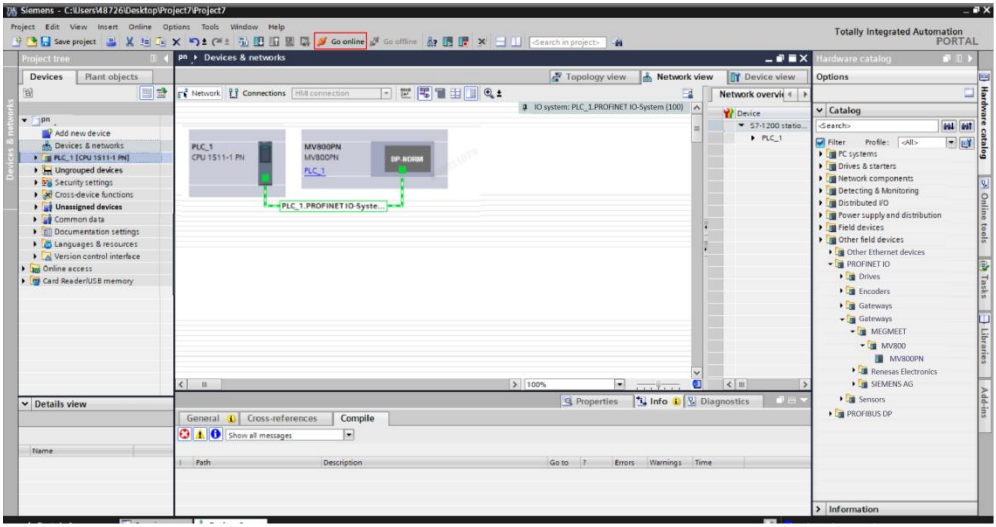

Load preview
Check before loading
Status
Target
Message
Action
PLC_1
Ready for loading.
Load 'PLC_1'
Protection
Protection from unauthorized access
Devices connected to an enterprise network or directly to the internet must be appropriately protected against unauthorized access, e.g. by use of firewalls and network segmentation. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity
Stop modules
The modules are stopped for downloading to device.
Stop all
Device configuration
Delete and replace system data in target
Download to device
Software
Download software to device
Consistent download
Certificate configu...
Password for confidential PLC configuration data protection is cha...
Refresh
Finish
Load
Cancel

Click “Finish” to complete the downloading.

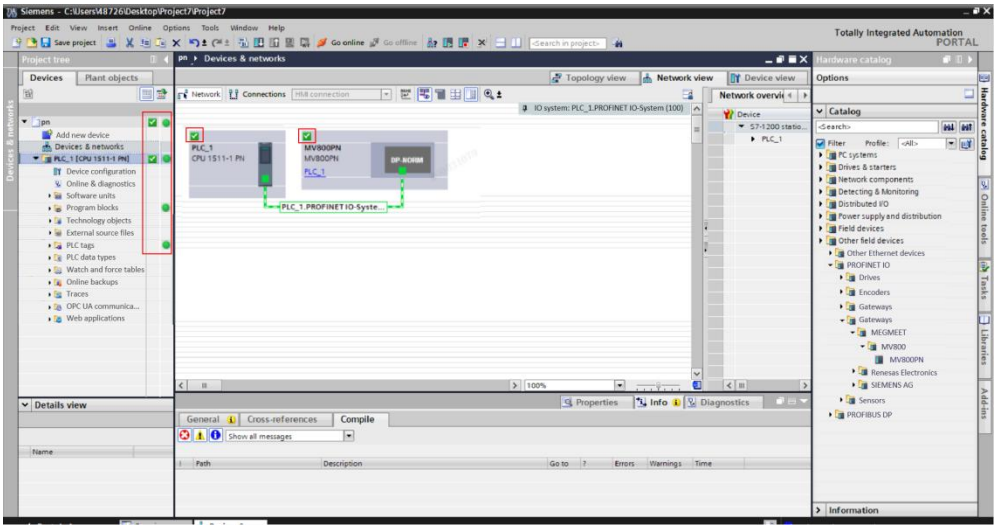


2.5.2.6 Watch

Click “Go online.”



Display of green icons indicates normal PN communication.



2.6 Integrated PN communication application

A single communication option is capable of controlling up to 5 drives, applicable for extruders, printing, and packaging machines, as shown below.

PN communication self-defined message can be divided into two types:

General PN communication

This is the traditional communication mode for controllers and PN devices. Each of the drives shall be equipped with a PN option. The address of the first two bytes in the self-defined message can be empty. The function code can be set as follows:

P02.02 = 2 (communication control)

P02.03 = 3 (PN communication control)

P02.05 = 8 (frequency reference channel set to PN)

P15.00 ones place = 0 (non PN-to-485 function)

P40.01 = 3.0 s (defined as the expansion card identification timeout; modifiable)

P43.01 = 1 (0 is the standard message 1, and 1 is the self-defined message)

Function codes (P43.02 to P43.12) are used to set the parameters modifiable by the controller.

Function codes (P43.13 to P43.23) are used to set the parameters available for reading by the controller.

PN to 485 (one PN option controlling up to 5 drives)

In this mode, only one drive is installed with the PN option, which can transmit the PLC messages to other drives through 485. The frame header and tail will be eliminated during transmission, making the length of the message to 33 bytes. Only self-defined messages are allowed in this mode. PLC uses the first two bytes (485 station number) of the message to visit the corresponding drive. The function code setting can be further divided into two types:

(1) 485 master

P02.02 = 2 (communication control)

P02.03 = 3 (PN communication control)

P02.05 = 8 (frequency reference channel set to PN)

P15.00 ones place = 1 (PN-to-485 function enabled)

P15.02 is used to set the local 485 station number.

P40.01 = 3.0s (defined as the expansion card identification timeout; modifiable)

P43.01 = 1 (use self-defined messages only)

Function codes (P43.02 to P43.12) are used to set the parameters modifiable by the controller.

Function codes (P43.13 to P43.23) are used to set the parameters available for reading by the controller.

(2) 485 slave

P02.02 = 2 (communication control)

P02.03 = 3 (PN communication control)

P02.05 = 8 (frequency reference channel set to PN)

P15.00 个位 = 1 (PN-to-485 function enabled)

P15.02 is used to set the local 485 station number.

P40.01 = 3.0s (defined as the expansion card identification timeout; modifiable)

P43.01 = 1 (use self-defined messages only)

Function codes (P43.02 to P43.12) are used to set the parameters modifiable by the controller.

Function codes (P43.13 to P43.23) are used to set the parameters available for reading by the controller.

Note:

The current baud rate of PN-485 is set to 200k. The time interval between sending the message and receiving the slave response by the master is less than 5 ms. The master transmits a PN message every 50 ms (this cycle time shall be larger than the total time needed for one time of sending and one time of response). Due to high frequency of message sending by the controller, it may take several rounds of

reading and writing before the controller receives the corresponding data and response, which makes the device applicable only for application with low requirement in real-time performance.

Chapter 3 CANopen Communication Option

3.1 Overview

Thank you for choosing Megmeet CANopen communication option. This manual provides information of the product functions, specifications, installation guidelines, basic operations, and settings, as well as an introduction to the network protocol. To ensure correct installation and operation of this product, please carefully read this manual and the communication protocol section of the drive user manual before using this communication option.

This manual serves as a guide for operating the CANopen communication option and includes relevant instructions. Detailed information about the CANopen protocol is not included herein. If users would like to learn more about the CANopen protocol, please refer to the professional articles or reference materials.

The communication card is defined as a CANopen slave communication option that can be used with drives supporting CANopen communication.

This communication option supports two methods of reading and writing the process variables from the drive: one through PDO, and the other through SDO for reading and writing the object dictionary defined by the manufacturer.

3.2 Features

MV800 CANopen provides services including:

(1) NMT

(2) Node guard message

(3) Heartbeat message

(4) 4 TxPDOs, 4 RxPDOs

(5) Quick SDO reading/writing of the drive function codes

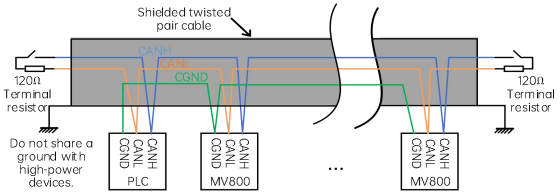
(6) Emergency message

(7) Synchronous mode

3.3 Electrical wiring and transmission distance

The network topology of CAN bus is shown in the figure below. It is recommended to employ the shielded twisted pair cable for CAN bus connection. Each end of the bus shall be connected to a 120 Ohms terminal resistor to prevent signal reflection. As a general rule, the 120 Ohms terminal resistor

would be added to the master station and the last slave station on the sequence. For the MV800-Canopen option, turn the terminal resistor switch 1 and switch 2 to ON position.



The transmission distance of CAN bus is directly correlated with the baud rate and the communication cable properties. The relation between the maximum bus length and the baud rate is shown in the table below:

Baud rate (bps)	Length (m)
1M	25
500k	100
250k	250
125k	500
100k	500
50k	1000
20k	1000

3.4 Parameter settings for CANopen network connection

To operate the MV800 drive using MV810-CANopen, users need to set the operation command channel and the frequency source of the MV800 drive to the bus communication card, as shown in the following table.

Drive parameter	Value	Function description
P02.02	2	Set the operation command channel to communication control
P02.03	3	Set the communication command channel to CANopen
P02.05	8	Set the main frequency source to CANopen

Settings of CANopen node address and baud rate are shown in the table below:

Drive parameter	Value	Function description
P40.01	0 to 10.0	CAN communication disconnection detection time

Drive parameter	Value		Function description
			(measured in second)
P40.20	1 to 127		CANopen station number
P40.21	0: 1 Mbps/s	1: 800 Kbps/s	CAN communication baud rate
	2: 500 Kbps/s	3: 250 Kbps/s	
	4: 125 Kbps/s	5: 100 Kbps/s	
	6: 50 Kbps/s	7: 20 Kbps/s	
	8: 10 Kbps/s		

3.5 Communication

3.5.1 Communication object COB-ID

CANopen provides multiple communication objects, each of which possesses different features (refer to the standard CANopen protocol for detailed information) and suits different applications. This expansion card adopts predefined COB-ID. The rules are shown below:

1) NMT object: 0x000

2) SYNC object: 0x080

3) SDO object

- Send SDO—0x600+Node-Id
- Receive SDO—0x580+Node-Id

4) PDO object:

- RPDO1—0x200+Node-Id
- RPDO2—0x300+Node-Id
- RPDO3—0x400+Node-Id
- RPDO4—0x500+Node-Id
- TPDO1—0x180+Node-Id
- TPDO2—0x280+Node-Id
- TPDO3—0x380+Node-Id
- TPDO4—0x480+Node-Id

5) EMCY object: 0x80+Node-Id

6) Heartbeat/Node guard object: 0x700+Node-Id

- Node-ID: device ID (station address), function code P40.20 setting;
- Communication object COB-ID is distributed in fixed form, and unchangeable.

3.5.2 NMT network management command

NMT message is used by the master to control the slave NMT status, the structure of which is shown in the table below. COB-ID is fixed to 0x00. Data0 refers to the one-byte command word. Data1 serves as the Node-ID of the CANopen slave node, and occupies one byte; when it turns to 0, it serves as a broadcast message, valid for all slave devices in the network.

COB-ID	RTR	Data0	Data1
0x000	0	Command word	Node-ID

Command word types are shown below:

Command word	Description
0x01	Start remote node
0x02	Stop remote node
0x80	Enter pre-operational state
0x81	Reset node
0x82	Reset communication

3.5.3 SYNC message

The SYNC message is sent by the NMT master station with the purpose of facilitating the synchronization of PDO transmission in the entire network when the PDO transmission type is set to SYNC (1 to 240). The structure of the message is shown below:

COB-ID	RTR
0x80	0

3.5.4 Process data object (PDO)

3.5.4.1 PDO trigger mode

The communication parameter of each PDO (the communication parameter for RxPDO sits in the index 1400h to 15FFh in the object dictionary, and the communication parameter for TxPDO sits in the index 1800h to 19FFh) has defined the transmission type, the inhibit time, and the event timer. The transmission type sub-index is 02, the inhibit time sub-index is 0x03, and the event timer sub-index is 0x05. The unit for the inhibit time and the event timer is ms.

Synchronous trigger: PDO transmission is synchronous when the transmission type is set to 1 to 240. For example, when the TxPD01 transmission type is set to n ($1 \leqslant n \leqslant 240$), the slave would send a TxPD01 command once the number of the SYNC messages it receives reaches n, and the same rule applies to other PDO operations.

Asynchronous trigger (254): when the event timer is set to non-0 status, the slave would send TxPDO messages in a periodical fashion. For example, when the TxPD01 event timer is set to 200, the slave

would send a TxPD01 message every 200 ms. When the event timer is set to 0, every time the corresponding TxPDO data changes, the slave would send a TxPDO message; however, the time interval is restricted by the inhibit time, which means the same PDO message would be sent once only in the range of the inhibit time, so the burden of the bus would be minimized.

Asynchronous trigger (255): when the event timer is set to non-0 status, the slave would send TxPDO messages in a periodical fashion. For example, when the TxPD01 event timer is set to 200, the slave would send a TxPD01 message every 200 ms. When the event timer is set to 0, the slave would send a TxPDO message once it receives a RxPDO message. For example, the slave would send a TxPD01 message once it receives a RxPD01 message.

For this option, the PDO transmission type is set to asynchronous trigger (255), the event timer is set to 0, and the inhibit time is set to 0 by default.

3.5.4.2 PDO mapping

This communication card supports 4 TxPDOs and 4 RxPDOs. Each PDO supports mapping of up to four 16-bit data.

Default RxPDO mapping is shown below:

RxPDO	Mapping
RxPDO1	Control word
	Frequency reference
RxPDO2	
RxPDO3	
RxPDO4	

RxPDO	Mapping

Default TxPDO mapping is shown below:

TxPDO	Mapping
TxPDO1	Status word
	Output frequency
TxPDO2	
TxPDO3	
TxPDO4	

Users can configure the PDO mapping to access the following data. For detailed information, refer to the "MV800_Canopen.eds" file.

RxPDO (sent by the master, and received by the slave):

Index/ Sub-index	Indication	Description	Value range	Access rights
6040h/0	Control word	Bit0: Forward running Bit1: Reverse running Bit2: Forward JOG Bit3: Reverse JOG	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read/ Write

Index/ Sub-index	Indication	Description	Value range	Access rights
		Bit4: Stop Bit5: Coast to stop Bit6: Fault reset Bit7: Emergency stop	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	
2100h/0	Drive torque upper limit	Unit: 0.1%	0.0 to 300.0% (corresponding to 0 to 3000)	Read/ Write
2101h/0	Braking torque upper limit	Unit: 0.1%	0.0 to 300.0%	Read/ Write
2102h/0	FWD frequency upper limit	Unit: 0.01 Hz	0.00 to 599.00 Hz (corresponding to 0 to 59900)	Read/ Write
2103h/0	REV frequency upper limit	Unit: 0.01 Hz	0.00 to 599.00 Hz	Read/ Write
2104h/0	Voltage reference (V/F separation)	Unit: 1 V	0 to 1000 V	Read/ Write
2105h/0	DO	Bit0: DO1 terminal Bit1: DO2 terminal Bit2: DO3 terminal Bit3: RO terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read/ Write
2106h/0	AO1	Unit: 0.01%	0.00 to 100.00% (corresponding to 0 to 10000)	Read/ Write
2107h/0	HDO1	Unit: 0.001 kHz	0.000 to 50.000 kHz (corresponding to 0 to 50000)	Read/ Write
2108h/0	HDO2	Unit: 0.001 kHz	0.000 to 50.000 kHz	Read/ Write
2109h/0	PID reference value	Unit: 0.1%	-100.0 to 100.0% (corresponding to -1000 to 1000)	Read/ Write
210Ah/0	PID feedback value	Unit: 0.1%	-100.0 to 100.0%	Read/ Write
210Bh/0	Position reference	Null	Null	Read/ Write

Index/ Sub-index	Indication	Description	Value range	Access rights
210Ch/0	Torque reference	Unit: 0.1%	-300.0 to 300.0%	Read/ Write
210Dh/0	Frequency reference	Unit: 0.01 Hz	0.00 to 599.00 Hz	Read/ Write

TxPDO (sent by the slave, and received by the master):

Index/ Sub-index	Indication	Description	Value range	Access rights
6041h/0	Status word	Bit0: Forward running Bit1: Reverse running Bit2: Stop Bit3: Fault Bit4: Power failure Bit5: Ready status Bit6: Motor number Bit7: Motor type Bit8: Overload pre-alarm Bit9 to Bit10: Control mode	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Not ready; 1: Ready 0: Motor 1; 1: Motor 2 0: Asynchronous motor; 1: Synchronous motor 0: Disabled; 1: Enabled 0: Keypad; 1: Terminal; 2: Communication	Read only
2200h/0	Output current	Unit: 0.1 A	0.0 to 6553.5 A (corresponding to 0 to 65535)	Read only
2201h/0	Output voltage	Unit: 1 V	0 to 65535 V	Read only
2202h/0	Output frequency	Unit: 0.01 Hz	0.00 to 599.00 Hz (corresponding to 0 to 59900)	Read only
2203h/0	Output torque	Unit: 0.1%	-300.0 to 300.0% (corresponding to -3000 to 3000)	Read only
2204h/0	Bus voltage	Unit: 0.1 V	0.0 to 6553.5 V	Read only
2205h/0	DI state 1	Bit0: DI1 terminal Bit1: DI2 terminal Bit2: DI3 terminal Bit3: DI4 terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only
2206h/0	DI state 2	Bit0: DI5 terminal Bit1: DI6 terminal Bit2: DI7 terminal Bit3: DI8 terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only
2207h/0	DO state	Bit0: DO1 terminal Bit1: DO2 terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	Read only

Index/ Sub-index	Indication	Description	Value range	Access rights
		Bit2: DO3 terminal Bit3: RO terminal	0: Disabled; 1: Enabled 0: Disabled; 1: Enabled	
2208h/0	Motor power	Unit: 0.1%	-300.0 to 300.0% (corresponding to -3000 to 3000)	Read only
2209h/0	Power output	Unit: 0.1 kW	0.0 to 6553.5 kW (corresponding to 0 to 65535)	Read only
220Ah/0	Position actual value	Null	Null	Read only
603Fh/0	Error code	Refer to the *Drive fault information* in section 3.5.6 *Emergency message*		Read only

3.5.5 SDO reading/writing operation

3.5.5.1 Drive function code mapping

This CANopen option enables access to the drive function codes via the quick SDO message. The drive function code group (P00 to P98) is mapped to the section (0x2000 to 0x2062) in the CANopen object dictionary, with an addition of 1 to the number on the last digit of each function code to form its sub-index in the dictionary. For example:

Drive function code P02.05 is mapped to the main index 0x2002 in the object dictionary, with the sub-index 0x06;

Drive function code P03.07 is mapped to the main index 0x2003 in the object dictionary, with the sub-index 0x08.

3.5.5.2 SDO reading/writing message

Quick SDO request message is shown below (from the master to the slave):

CAN	Description
11-bit ID	0x600+Node-ID
RTR	0
DATA0	Command specifier (CS)
DATA1	Low-byte index

CAN	Description
DATA2	High-byte index
DATA3	Sub-index
DATA4	Request data bit0 to 7
DATA5	Request data bit8 to 15
DATA6	Request data bit16 to 23
DATA7	Request data bit24 to 31

Quick SDO response message is shown below (from the slave to the master):

CAN	Description
11-bit ID	0x580+Node-ID
RTR	0
DATA0	Command specifier (CS)
DATA1	Low-byte index
DATA2	High-byte index
DATA3	Sub-index
DATA4	Response data bit0 to 7
DATA5	Response data bit8 to 15
DATA6	Response data bit16 to 23
DATA7	Response data bit24 to 31

Types of the command specifier in the request/response message is shown below:

Command specifier	Description	Command specifier	Description
0x2F	Write 1 byte	0x40	Read
0x2B	Write 2 bytes	0x4F	Read 1-byte response
0x27	Write 3 bytes	0x4B	Read 2-byte response
0x23	Write 4 bytes	0x47	Read 3-byte response
0x60	Write success response	0x43	Read 4-byte response
0x80	Abnormal response		

Examples:

When the address of the drive CANopen is set to 0x03, the SDO request for reading the drive function code P02.05 is shown below:

COB-ID	RTR	Data (Hex)
0x603	0	40 02 20 06 00 00 00 00

The value of the drive P02.05 is 8, and the response is shown below:

COB-ID	RTR	Data (Hex)
0x583	0	4B 02 20 06 08 00 00 00

3.5.5.3 SDO exception code

When error occurs during SDO reading/writing, the command specifier in the SDO response message would be 0x80, and the response data would be the exception code shown in the table below:

Exception code	Description	Exception code	Description
0x05040000	SDO visit timeout	0x06070010	Data type mismatching; serve parameter length mismatching
0x06010000	The object does not allow visit.	0x06090011	The sub-index does not exist.
0x06010001	The object for reading is write only.	0x06090030	Reading visit out of parameter value range
0x06010002	The object for writing is read only.	0x08000022	Data can not be transmitted or saved to the application due to the current device status.
0x06020000	The object dictionary does not exist.	0x08000000	General fault

3.5.6 Emergency message

The following message would be sent when internal error occurs inside the communication card or the drive, or when the error is cleared.

COB-ID	RTR	Data0 to 1	Data2	Data3 to 7
0x80+Node-ID	0	Emergency error code	Error register	Error code designated by the manufacturer

※ Emergency error code: refer to the related chapters in the DS301 document; "0x8100" for communication error; "0xFF00" for error designated by the manufacturer.

※ Error register: refer to the 1001H data in the object dictionary in the related DS301 document chapters; bit0 for generated error flag; bit4 for communication error flag; bit7 for error designated by the manufacturer.

※ Error code designated by the manufacturer: refer to the drive fault information in the following table. For detailed information, refer to the *MV810 High-Performance Vector Control Drive User Manual*.

Drive fault information	Drive fault information	Drive fault information
0: No error generated 1: Overcurrent during acceleration (OC1) 2: Overcurrent during deceleration (OC2) 3: Overcurrent during operation at constant speed (OC3) 4: Overvoltage during acceleration (OV1) 5: Overvoltage during deceleration (OV2) 6: Overvoltage during operation at constant speed (OV3) 7: Undervoltage fault (Uv) 8: Input phase loss (SPI) 9: Output phase loss (SPO) 10: Power module protection (drv) 11: Inverter overheat (OH1) 12: Rectifier bridge overheat (OH2) 13: AC drive overload (OL1) 14: Motor overload (OL2)	15: External fault (EF) 16: EEPROM read/write fault (EEP) 17: 485 communication error (CE) 18: EtherCAT communication timeout (E-CAt) 19: Current detection error (ItE) 20: CANopen communication timeout (E-CAN) 21: PID feedback loss (FbL) 22: Reserved 23: Braking resistor overcurrent (brOC) 24: Auto-tuning fault (tUN) 25: Reserved 26: Profinet communication timeout (E-Pn) 27: I/O card communication timeout (E-Io) 28: Modbus TCP communication timeout (E-TCP) 29 to 32: Reserved 33: Short-to-ground fault (GdF)	34: Speed deviation fault (dEv) 35 to 38: Reserved 39: Motor overheat (OH3) 40: Reserved 41: 24 V power supply overload (24OL) 42 to 45: Reserved 46: Board-level communication error (bCE) 47: Reserved 48: BootLoader failure (bLt) 49: Power board software version mismatching (vEr) 50: Parameter upload and download timeout (UPdnE) 51: AI1 current input overcurrent (AIOC) 52: Reserved 53: Fan locked-rotor (FAn) 54: Pre-overload (POL1) 55: I/O card 24 V overload (IO-OL)

3.5.7 Node guard message

Node guard service enables the present status inspection of each node by the NMT main node. The NMT main node sends a remote frame to request the slave status using the message below:

COB-ID	RTR
0x700+Node-ID	1

The slave response message is shown below:

COB-ID	RTR	Data0
0x700+Node-ID	0	Status word

The status word of Data0 is described below:

Status word	Description
bit7	It is required to alternate between 0 and 1 in the setting each time.
bit6 to bit0	Status: 0: Initialized 4: Stop 5: Run 127: Pre-run

3.5.8 Heartbeat message

The master station may occasionally require the slave station to initiate a heartbeat message at intervals so it can learn the real-time slave status. The time interval parameter is defined in the object dictionary 0x1017 (16-bit data length, with the unit set to ms). When the time is set to 0, the slave would not send heartbeat messages. The producer heartbeat time of this CANopen communication card is set to 0 by default.

COB-ID	RTR	Data0
0x700+Node-ID	0	Status word

Bit7 of the heartbeat message status word is set to 0 and does not allow changes. The definitions of bit0 to bit6 are the same with the definitions of bit0 to bit6 in the status word of the node guard response frame.

3.6 Fault diagnosis

3.6.1 LED indicator description and fault removal

MV810-CANopen has three LED indicators. Their descriptions are shown below:

LED	Status	Description	Action
LED1 (Red)	Off	No power supply for CANopen	Check whether the CANopen option is properly connected to the drive
	Steady on	Normal power supply for CANopen	No need for actions
LED2 (Green)	Off	State machine in Stopped state	Check whether the CANopen option is properly connected to the host controller
	Flashing	State machine in Pre-OP state	Check whether the CANopen option is properly connected to the host controller
	Steady on	State machine in OP state	No need for actions
LED3 (Red)	Off	Normal	No need for actions
	Flashing	CANopen station number collision	Reset P40.20, power off and restart
	Steady on	CANopen emergency message fault	Solve the problem indicated in the fault information of the emergency message

3.6.2 Function code diagnosis information

Function codes of CANopen commissioning (read only):

Drive parameters	Value	Function
P50.07	0: Boot-up; 4: Stopped; 5: Operational; 127: Pre-operational	CANopen communication status
P50.08	0 to 65535	Accumulated number of CAN sending/receiving errors

Diagnosis:

If the value of P50.08 is greater than 0 and continues to increase, it indicates a case of existing interference with or improper configuration of the network, and that an action is needed for troubleshooting.

Methods for troubleshooting:

Check whether all the nodes have the same baud rate, and whether the addresses have the same setting. Check whether the DIP switch is correctly set in place, and whether the main controller baud rate and address are properly configured.

Check whether the terminal resistors are connected to the two ends of the bus only. Power off the whole unit, and measure the resistance between CANH and CANL in the bus using a multimeter. If the value sits in the range of 50 to 60 Ohms, it indicates a normal state of resistance.

Check whether the node CANH and node CANL are reversely connected, and whether the bus port CGND end is connected (in normal state, it is required to connect the CGND ends of all devices only, and grounding is not required).

3.6.3 Function code reading/writing fault

The object dictionary of the index 0x2064 indicates the drive function code reading/writing fault by the CANopen master station: the data corresponding to the sub-index 1 indicate the fault code, with high 8 bits indicating a writing error and low 8 bits indicating a reading error. The data corresponding to the sub-index 2 indicate the index of the function code with reading/writing errors. For example, 0x0200 indicates that there is an error of reading/writing the function code P02.00. Types of fault codes are shown below:

Fault	Fault code
Wrong password	0xF1
Index for operation does not exist	0xF4
Invalid parameter	0xF5
Parameter read only	0xF6
System lock	0xF7
EEPROM performing storage	0xF8

3.7 Communication example of Inovance H5U PLC controlling MV800

The following example is based upon an Inovance H5U PLC as the master to demonstrate the configuration and usage of the MV800 CANopen communication.

(1) Start AutoShop software, select "New Project", select "H5U Series" in "Series and models", and click the "OK" button to start programming.

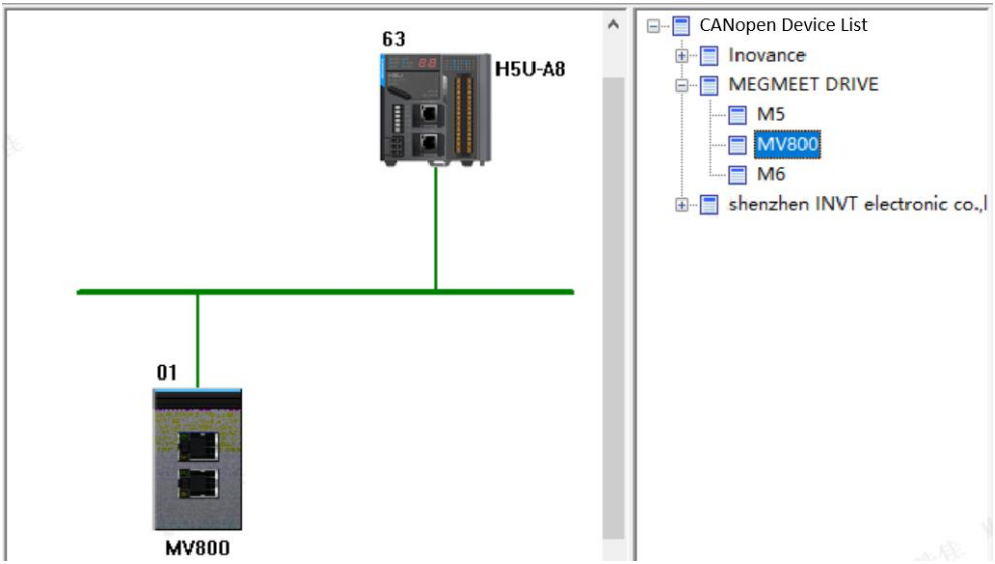
(2) Select and click CAN(CANopen) in the left column to enter "CAN Config", set up the program as the following, and click the "OK" button.

"Protocol": CANopen

"Station No." in "Communicate Param": Upper computer setting (the CANopen station number of PLC shall be different with that of the AC drive)

"Baud Rate" in "Communicate Param": Upper computer setting (the baud rate shall be in consistency with that of the AC drive)

(3) Configure the master and the slave: right click CAN(CANopen) and select "Add CAN Config" in the pop-up window to display CANopen Config ; double click "CANopen Config" to open the configuration view, right click "CANopen device list" in the right column, select "Import EDS" to import the "MV800_Canopen.eds" file; double click the MV800 slave station to add it to the network.



(4) Double click the slave station, and configure the "Receive PDO" and "Send PDO."

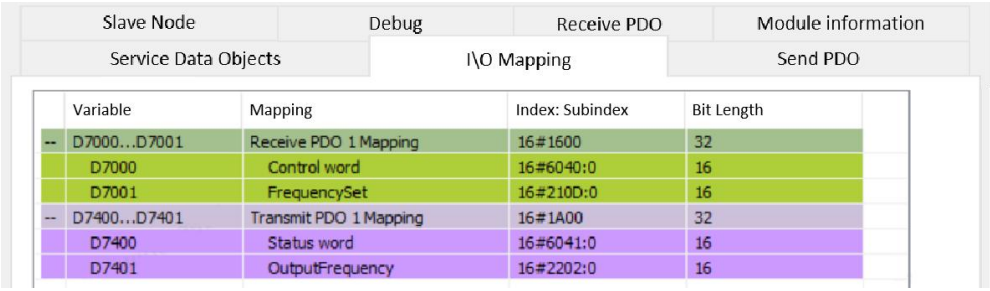
MV800

Slave Node	Debug	I\O Mapping	Module information
Service Data Objects	Receive PDO	Send PDO	

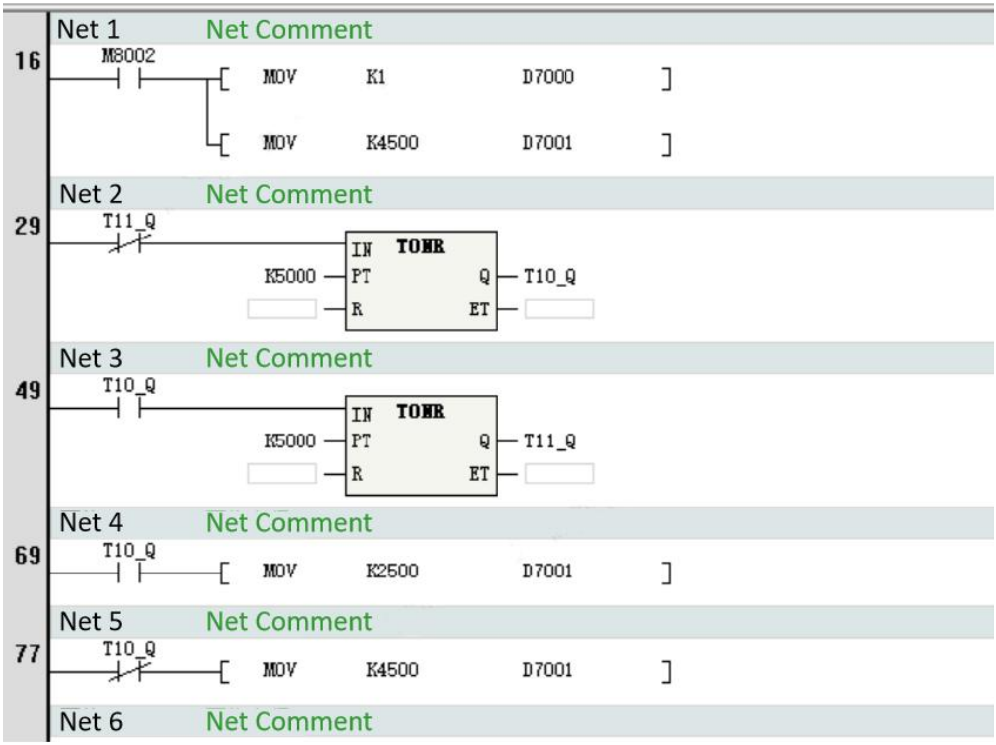
Num...	Name	Index	Subindex	Bit Length
☑ 1	**Receive PDO 1 Parameter**	**16#1400**		
	Control word	16#6040	16#00	16
	FrequencySet	16#210D	16#00	16
☑ 2	**Receive PDO 2 Parameter**	**16#1401**		
☑ 3	**Receive PDO 3 Parameter**	**16#1402**		
☑ 4	**Receive PDO 4 Parameter**	**16#1403**		

(5) Check I\O mapping

When PDO is configured, H5U PLC will automatically map all the PDO data to the D element. During PLC programming, PDO data are read/written via the D element..



(6) Conduct the PLC programming, and download the program to PLC for multi-speed running control of the AC drive, as shown in the following image.



Chapter 4 Modbus TCP Communication Option

4.1 Overview

Thank you for choosing Megmeet Modbus TCP communication option. This manual provides information of the product functions, specifications, installation guidelines, basic operations, and settings, as well as an introduction to the network protocol. To ensure correct installation and operation of this product, please carefully read this manual and the communication protocol section of the drive user manual before using this communication option.

This manual serves as a guide for operating the Modbus TCP communication option, and includes relevant instructions. Detailed information about the Modbus TCP protocol is not included herein. If users would like to learn more about the Modbus TCP protocol, please refer to the professional articles or reference materials.

This communication option is defined as a slave communication option that can be used with drives supporting Modbus TCP communication.

4.2 Features

This MV800 Modbus TCP option offers services including the followings:

(1) Read slave parameters (0x03)

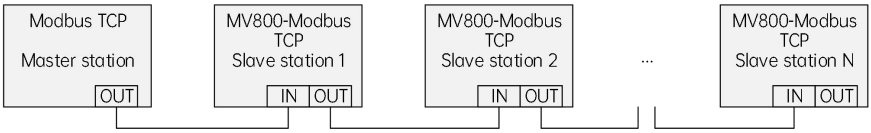
(2) Modify single slave parameter (0x06)

(3) Modify multiple slave parameters (0x10)

(4) Mutable mapping of address (setting via P30 function code group)

4.3 Electrical connection

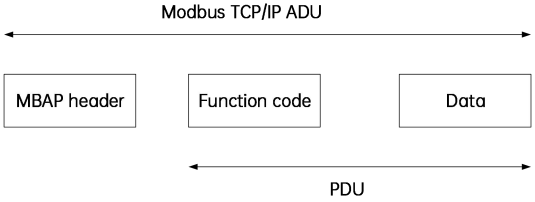
Modbus TCP network is generally composed of a master station and multiple slave stations. The network can be structured into a bus type, star type, tree type, or a combination of several types, enabling flexible device connection and wiring. The bus-type network topology is shown in the figure below.



4.4 Modbus TCP communication

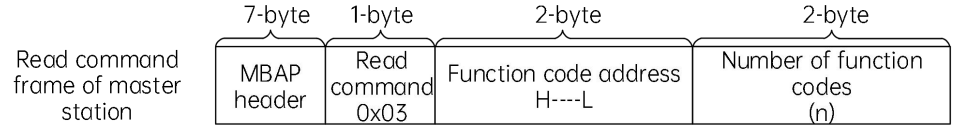
4.4.1 Modbus TCP data frame structure

During Modbus TCP communication, only parameters in a word format can be read/written by the drive. The corresponding read command is 0x03; the write command is 0x06; the multi-data write command is 0x10; the read/write command of parameters in a byte/bit format is not supported. The data format is shown below:



The MBAP header contains the following fields:

Fields	Length	Description	Client	Server
Transaction Identifier	2 Bytes	Identification of a Modbus request/response transaction	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = Modbus protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client (request)	Initialized by the server (response)
Unit Identifier	2 Bytes	Identification of a remote slave connected on a serial line or on other buses	Initialized by the client	Recopied by the server from the received request



Theoretically, the host controller supporting reading of multiple sequential parameters (n up to 12) at a time. However, those sequential parameters shall not pass over the last parameter in the group; otherwise, response error may occur.

Read response frame of master station

7-byte	1-byte	1-byte	(2n)-byte
MBAP header	Read command 0x03	Function code address (2n)	Number of function codes

Write command frame of master station

7-byte	1-byte	2-byte	2-byte
MBAP header	Write command 0x06	Function code address H----L	Number of function codes

Read command frame of slave station

7-byte	1-byte	2-byte	2-byte
MBAP header	Write command 0x06	Function code address H----L	Number of function codes

Multi-data write command frame of master station

7-byte	1-byte	2-byte	2-byte	1-byte	(2n)-byte
MBAP header	Write command 0x10	Function code address H----L	Number of function codes (n)	Number of data bytes (2n)	Function code parameter

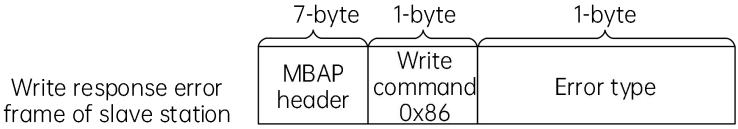
Multi-data write response frame of slave station

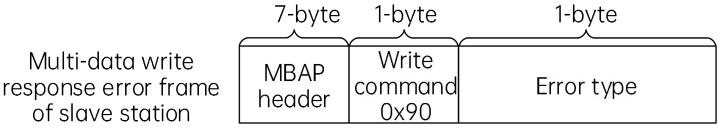
7-byte	1-byte	2-byte	2-byte	1-byte	2-byte
MBAP header	Write command 0x10	Function code address H----L	Number of function codes (n)	Number of data bytes (2n)	Function code address H----L

The read response error code of slave station is 0x83; the write response error code is 0x86; the multi-write response error code is 0x90.

Read response error frame of slave station

7-byte	1-byte	1-byte
MBAP header	Write command 0x83	Error type





Data frame field description:

Command code	0x03 refers to the reading of slave parameters; 0x06 refers to the writing of slave parameters; 0x10 refers to the multi-writing of slave parameters.
Function code address	It refers to the address of the drive internal parameter, indicated in a hexadecimal format. It includes the parameter type and the non-parameter type (e.g. running status parameters, or running commands, etc.). During transmission, the high bytes run ahead and the low bytes follow.
Number of function codes	It refers to the number of parameters read in this frame. When it is set to 1, it indicates that only 1 parameter is read. During transmission, the high bytes run ahead and the low bytes follow. This protocol supports modification of only 1 parameter at a time, and this field is not available.
Number of data bytes	It refers to the data length, which is twice the number of parameters.
Function code parameter	It refers to the response data or data to be written. During transmission, the high bytes run ahead and the low bytes follow.

4.4.2 Parameter setting for Modbus TCP network connection

To operate the MV800 platform AC drive using this MV810-TCP01 option, it is to required to set the command channel and frequency source of the MV800 Platform drive to bus communication card, as shown below.

Parameter	Value	Function description
P02.02	2	It serves to set the operation command channel to communication control.
P02.03	0	It serves to set communication operation channel to Modbus TCP.

Parameter	Value	Function description
P02.05	7	It serves to set the main frequency source to Modbus TCP.

IP address setting (IP, subnet mask, and gateway) is shown below.

Drive parameter	Value	Function description
P40.02	0 to 255	IP address 1
P40.03	0 to 255	IP address 2
P40.04	0 to 255	IP address 3
P40.05	0 to 255	IP address 4
P40.06	0 to 255	Subnet mask 1
P40.07	0 to 255	Subnet mask 2
P40.08	0 to 255	Subnet mask 3
P40.09	0 to 255	Subnet mask 4
P40.10	0 to 255	Gateway 1
P40.11	0 to 255	Gateway 2
P40.12	0 to 255	Gateway 3
P40.13	0 to 255	Gateway 4

4.4.3 Mutable mapping application of address

To use the mutable mapping function, it is required to set the communication action to the tenth place 1 of P15.05 first.

The mutable mapping of the drive parameters is shown in the following table.

Drive parameter	Value	Function name	Description
P30.00	0 to 0xFFFF	Mapped address 1 of the 485 parameter	The mapped address refers to the actual internal address of the parameter; the used address refers
P30.01	0 to 0xFFFF	Used address 1 of the	

Drive parameter	Value	Function name	Description
		485 parameter	to the parameter address used in the message. (For example, the actual operating address of PLC.)
P30.02	0 to 0xFFFF	Mapped address 2 of the 485 parameter	
P30.03	0 to 0xFFFF	Used address 2 of the 485 parameter	
P30.04	0 to 0xFFFF	Mapped address 3 of the 485 parameter	
P30.05	0 to 0xFFFF	Used address 3 of the 485 parameter	
P30.06	0 to 0xFFFF	Mapped address 4 of the 485 parameter	
P30.07	0 to 0xFFFF	Used address 4 of the 485 parameter	
P30.08	0 to 0xFFFF	Mapped address 5 of the 485 parameter	
P30.09	0 to 0xFFFF	Used address 5 of the 485 parameter	

Example:

The address of P02.00 is 0x0200, and the address of P03.00 is 0x0300. Due to the discontinuity of the function code address, it is required to conduct the following address mapping for PLC to operate the above function codes in sequence via the 0x1000 address.

P30.00 = 0x0200, P30.01 = 0x1000

P30.02 = 0x0300, P30.03 = 0x1001

4.5 Fault diagnosis

4.5.1 LED indicator description and fault diagnosis

MV810-TCP01 has LED indicators on five locations, including the LED indicators on the PCBA of the expansion box for indicating the function status and the power status, and the LED indicators of the communication ports for indicating whether the communication connection status of MV810-TCP01 is normal.

Description of LED on the PCBA of the expansion box:

LED4 (Red) status	Description	Action
Off	Normal	No need for actions
Steady on	Communication timeout between the master and the communication card	Check whether the TCP01 option is properly connected to the drive

Description of LED on the communication port:

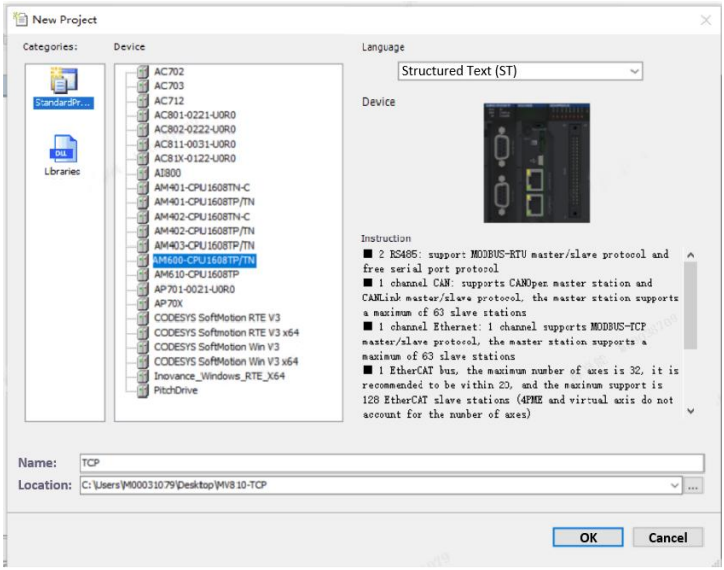
LED status	Description	Action
Yellow light flashing	Normal connection with data transmission available	No need for actions
Green light steady on	Normal connection	No need for actions
Yellow light steady on	Normal connection without data transmission	Check whether there is communication between the master and the slave.
Green light off	Connection failed	Check whether the cable is properly connected.

4.6 Communication example of Inovance AM600 PLC controlling MV800

The following example is based upon an Inovance AM600 PLC as the master to demonstrate the configuration and usage of the MV800 Modbus TCP communication.

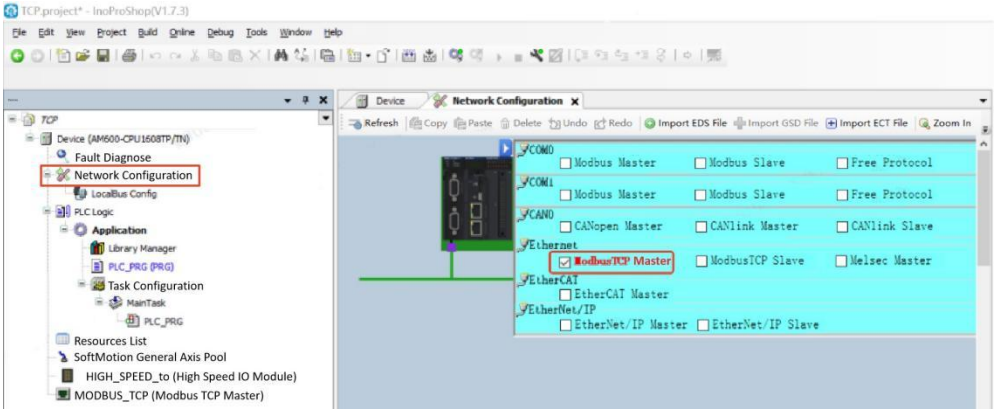
STEP 1: create a new project

Open the "New Project" window in the InoProShop software interface, and select "AM600-CPU1608TP/TN" in the "Device" list, as shown in the following image.



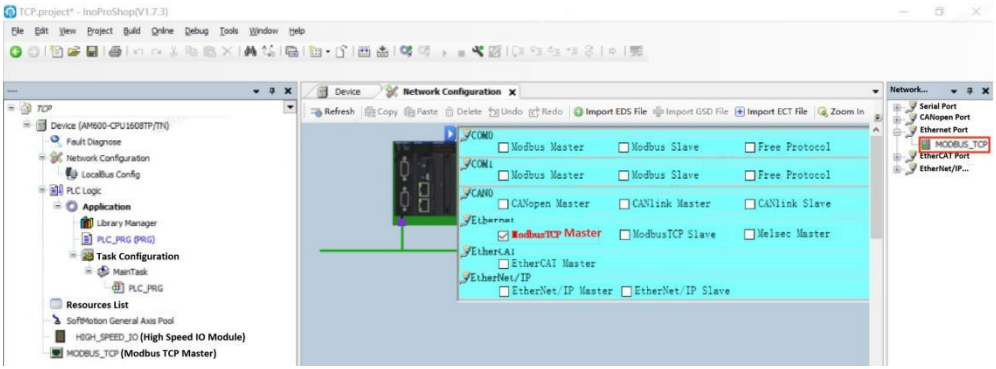
STEP 2: configure the network

Double click "Network Configuration" in the left column, and select "Modbus TCP Master" to add it to the network, as shown in the following image.

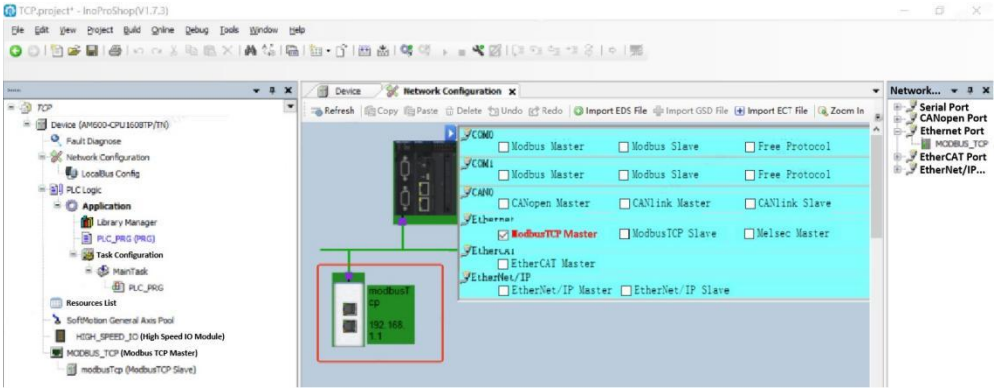


STEP 3: Add a slave

Doube click "MODBUS_TCP" in the right column, as shown in the following image.

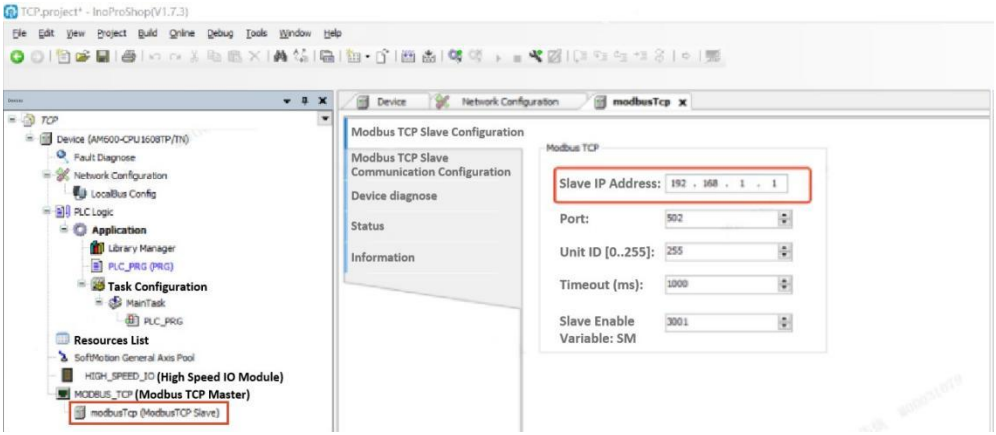


When the slave is successfully added to the network, the interface will display the view as follows.



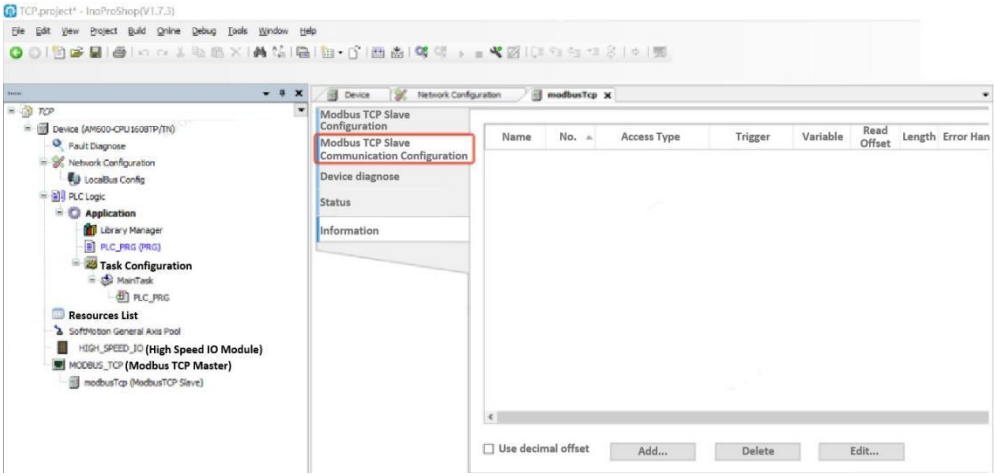
STEP 4: configure slave information

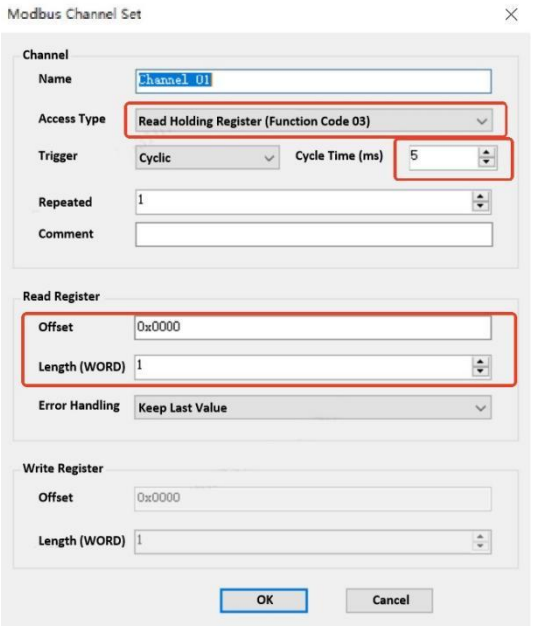
Double click "modbusTCP (ModbusTCP Slave)" in the left column to configure the slave IP address in the displayed interface.



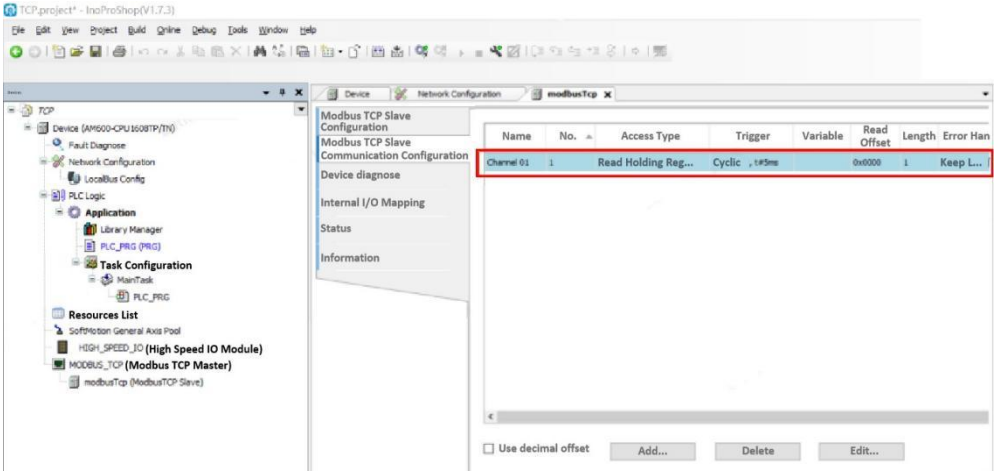
STEP 5: configure read/write command

Click "Modbus TCP Slave Communication Configuration", and click the "Add" button to display the "Modbus Channel Set" dialog box. Then, configure "Function Code", "Cycle Time", "Offset", "Length", etc. as shown in the following images.



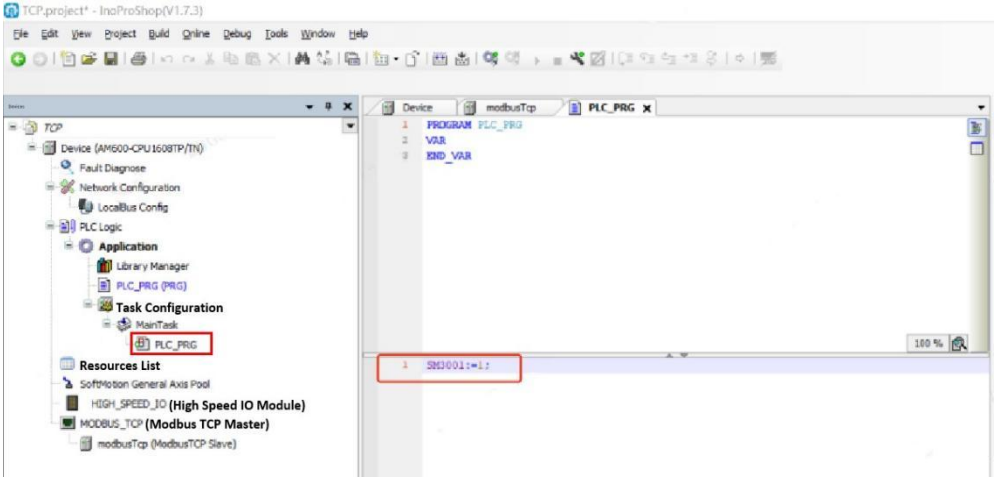


When the setting is successfully completed, the following view will be displayed.

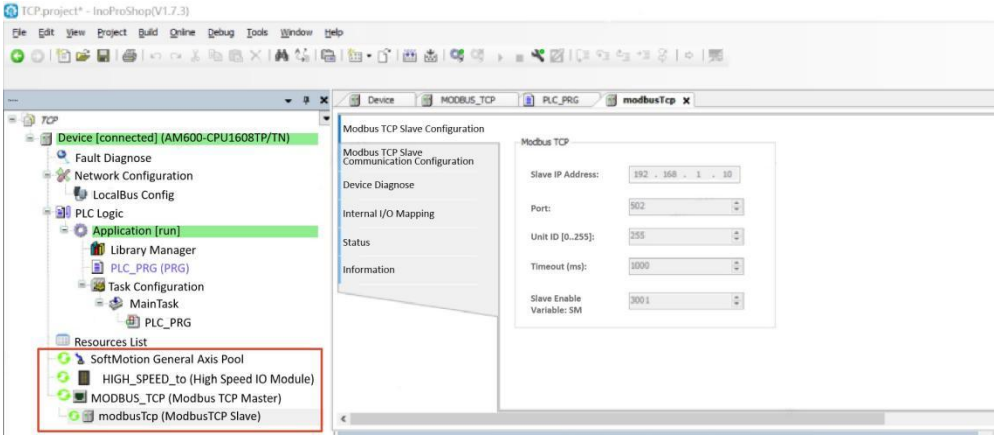


STEP 6: enable Modbus TCP communication

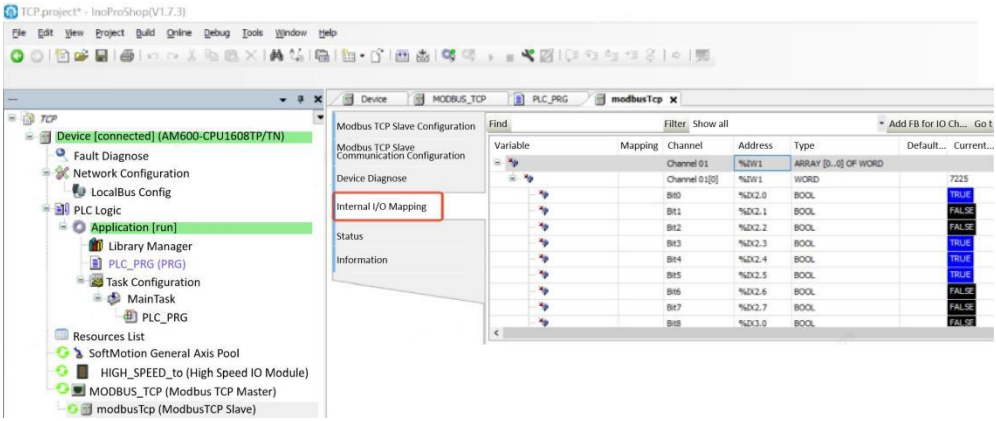
Enable "Slave Enable Variable" in "PLC_PRG" as shown in the image below.



Save the project, build and download it to PLC, and confirm the success of Modbus TCP communication connection between PLC and MV800 drive by checking whether there are green marks before the master and slave items in the left project tree, as shown in the image below.



Check the read/write variables in the "Internal I/O Mapping" interface, as shown in the image below.


TCP.project* - InoProShop(V1.7.3)
File Edit View Project Build Online Debug Tools Window Help
TCP
Device [connected] (AM600-CPU1608TP/TN)
Fault Diagnose
Network Configuration
LocalBus Config
PLC Logic
Application [run]
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask
PLC_PRG
Resources List
SoftMotion General Axis Pool
HIGH_SPEED_to (High Speed IO Module)
MODBUS_TCP (Modbus TCP Master)
modbusTcp (ModbusTCP Slave)
Device
MODBUS_TCP
PLC_PRG
modbusTcp
Modbus TCP Slave Configuration
Modbus TCP Slave Communication Configuration
Device Diagnose
Internal I/O Mapping
Status
Information
Find
Filter Show all
Add FB for IO Ch...
Variable
Mapping
Channel
Address
Type
Default...
Current...
Channel 01
%IW1
ARRAY [0..0] OF WORD
Channel 01[0]
%IW1
WORD
7225
Bit0
%IX2.0
BOOL
TRUE
Bit1
%IX2.1
BOOL
FALSE
Bit2
%IX2.2
BOOL
FALSE
Bit3
%IX2.3
BOOL
TRUE
Bit4
%IX2.4
BOOL
TRUE
Bit5
%IX2.5
BOOL
TRUE
Bit6
%IX2.6
BOOL
FALSE
Bit7
%IX2.7
BOOL
FALSE
Bit8
%IX3.0
BOOL
FALSE

Chapter 5 Ethernet/IP Communication Option

5.1 Overview

Thank you for choosing Megmeet Ethernet/IP communication option. This manual provides information of the product functions, specifications, installation guidelines, basic operations, and settings, as well as an introduction to the application of this option. To ensure correct installation and operation of this product, please carefully read this manual and the communication protocol section of the drive user manual before using this communication option.

This manual serves as a guide for operating the Ethernet/IP communication option and includes relevant instructions. Detailed information about the Ethernet/IP protocol is not included herein. If users would like to learn more about the Ethernet/IP protocol, please refer to the professional articles or reference materials.

This communication card is defined as a slave communication option that can be used with drives supporting Ethernet/IP communication.

This communication option provides the "MEGMEET_MV800_EthernetIP_V1.01.EDS" file, which can be obtained from the manufacturer or by downloading from our official website.

5.2 Features

MV800 Ethernet/IP option provides services including:

(1) PZD control of data exchange

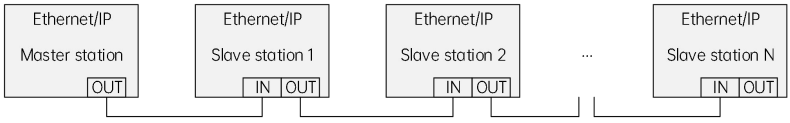
(2) PKW access to drive parameters

(3) 100 Mbps full duplex

(4) Linear bus topology and star-type topology

5.3 Electrical connection

Ethernet/IP network is generally composed of a master station and multiple slave stations. The network can be structured into a bus type, star type, tree type, or a combination of several types, enabling flexible device connection and wiring. The bus-type network topology is shown in the figure below.



5.4 Ethernet/IP communication application

The PLC can read/write the drive PKW/PZD via the EIP protocol.

The data format is explained as the table below.

Parameter	Byte	Description
PKW0	Byte0	This is the EIP slave station (where the EIP communication option is installed). Byte0: Target station number Byte1: Source station number (local station number) Slave response: Byte0: Target station number Byte1: Source station number (local station number)
	Byte1	
PKW1	Byte2	Function code read/write command (read/write one function code at a time) 0x03: Read one function code 0x06: Write one function code, and save to EEPROM 0x07: Write one function code; not save to EEPROM Byte2: Command word high byte; Byte3: Command word low byte Slave response: Byte2: 0 Byte3: 0x03, response to read operation 0x06 + 0x07, response to write operation 0x80 + command code, response error
	Byte3	
PKW2	Byte4	Address of the function code to be read/written Byte4: Address high byte; Byte5: Address low byte Slave response: Byte4: Address high byte; Byte5: Address low byte
	Byte5	
PKW3	Byte6	For write operation, PKW3 refers to the written value; for read operation, PKW3 refers to the number of function codes to be read (fixed value: 1) Byte6: Parameter high byte; Byte7: Parameter low byte Slave response: Byte6: Function code value high byte (response to read operation), 0 (response to write operation), error code high byte (response error) Byte7: Function code value low byte (response to read operation), 0 (response to write operation), error code low byte (response error)
	Byte7	

<table>
<tr><th>Parameter</th><th>Byte</th><th>Description</th></tr>
<tr><td rowspan="2">PZD1</td><td>Byte8</td><td>The master sends control command word:
Bit0: FWD run; 0: Disable; 1: Enable
Bit1: REV run; 0: Disable; 1: Enable
Bit2: FWD jog; 0: Disable; 1: Enable
Bit3: REV jog; 0: Disable; 1: Enable
Bit4: Decelerate to stop; 0: Disable; 1: Enable
Bit5: Coast to stop; 0: Disable; 1: Enable
Bit6: Fault reset; 0: Disable; 1: Enable</td></tr>
<tr><td>Byte9</td><td>Bit7: Emergency stop; 0: Disable; 1: Enable
Byte8: Command word high byte
Byte9: Command word low byte
Status word of slave response:
Bit0: FWD running 0: Disable; 1: Enable
Bit1: REV running 0: Disable; 1: Enable
Bit2: Stop 0: Disable; 1: Enable
Bit3: Fault 0: Disable; 1: Enable
Bit4: Power failure 0: Disable; 1: Enable
Bit5: Ready state 0: Disable; 1: Enable
Bit6: Motor number 0: Motor 1; 1: Motor 2
Bit7: Motor type 0: Asynchronous; 1: Synchronous
Bit8: Overload pre-alarm 0: Disable; 1: Enable
Bit9 to Bit10: Control mode 0: Keypad; 1: Terminal; 2: Communication
Byte8: Status word high byte
Byte9: Status word low byte</td></tr>
<tr><td rowspan="2">PZD2</td><td>Byte10</td><td rowspan="5">The eleven words (PZD2 to PZD12) are used to read/write the drive internal parameters which include P43.02 to P43.23 (parameters P43.02 to P43.12 are used for write operation; P43.13 to P43.23 are used for read operation).
<table><tr><td>P43.02</td><td>PZD2 received</td><td rowspan="4">0: Disable
1: Frequency reference (0.00 to P02.10)
2: Drive torque upper limit reference (0.0 to 300.0% motor rated current)
3: Brake torque upper limit reference (0.0 to</td></tr><tr><td>P43.03</td><td>PZD3 received</td></tr><tr><td>P43.04</td><td>PZD4 received</td></tr><tr><td>P43.05</td><td>PZD5 received</td></tr></table></td></tr>
<tr><td>Byte11</td></tr>
<tr><td rowspan="2">PZD3</td><td>Byte12</td></tr>
<tr><td>Byte13</td></tr>
<tr><td rowspan="2">PZD4</td><td>Byte14</td></tr>
<tr><td>Byte15</td></tr>
<tr><td>PZD5</td><td>Byte16</td></tr>
</table>

Parameter	Byte	Description		
	Byte17	P43.06	PZD6 received	300.0% motor rated current) 4: Torque reference (-300.0 to 300.0% motor rated current) 5: FWD run frequency reference upper limit (0.00 toP02.10) 6: REV run frequency reference upper limit (0.00 to P02.10) 7: Voltage reference (V/F separation) (0 to 1000) 8: Virtual input terminal command (0 to 0xFF are corresponding to DI8 to DI1) 9: Output terminal bus command (the output terminal function is set to 39; 0 to 0xF are corresponding to RO, DO3, DO2, and DO1) 10: AO1 output reference (0 to 100.0%) 11: HDO1 output reference (0 to 100.0%) 12: HDO2 output reference (0 to 100.0%) 13: PID reference (0.0 to 100.0%) 14: PID feedback (0.0 to 100.0%) 15 to 30: Reserved
PZD6	Byte18	P43.07	PZD7 received	
	Byte19	P43.08	PZD8 received	
PZD7	Byte20	P43.09	PZD9 received	
	Byte21	P43.10	PZD10 received	
PZD8	Byte22	P43.11	PZD11 received	
	Byte23	P43.12	PZD12 received	
PZD9	Byte24			
	Byte25			
PZD10	Byte26			
	Byte27			
PZD11	Byte28			
	Byte29			
PZD12	Byte30			
	Byte31	P43.13	PZD2 feedback	0: Disable 1: Frequency reference (0.01 Hz) 2: Ramp frequency reference (0.01 Hz) 3: Frequency output (0.01 Hz) 4: Voltage output (1 V) 5: Current output (0.1 A) 6: Bus voltage (0.1 V) 7: Motor power (0.1%) 8: Torque output (0.1%) 9: Excitation current (0.1 A) 10: Torque current (0.1 A) 11: Status word (0 to 0xFFFF) 12: Fault code (0 to 46) 13: DI1 to DI4 status (0 to 0xFFFF)
		P43.14	PZD3 feedback	
		P43.15	PZD4 feedback	
		P43.16	PZD5 feedback	
		P43.17	PZD6 feedback	
		P43.18	PZD7 feedback	
		P43.19	PZD8 feedback	
		P43.20	PZD9 feedback	
		P43.21	PZD10 feedback	

<table>
<tr><th>Parameter</th><th>Byte</th><th colspan="3">Description</th></tr>
<tr><td></td><td></td><td colspan="3">
<table>
<tr><td>P43.22</td><td>PZD11 feedback</td><td>14: DI5 to DI8 status
15: DO status (0 to 0xF)</td></tr>
<tr><td>P43.23</td><td>PZD12 feedback</td><td>16: AI1 input voltage (0 to 10.00 V)
17: AI2 input voltage (-10.00V to 10.00 V)
18: HDI input frequency (0 to 50.000 kHz)
19: AO output value (0 to 100.0%)
20: HDO1 output value (0 to 50.000 kHz)
21: HDO2 output value (0 to 50.000 kHz)
22: PID reference (-100.0% to 100.0%)
23: PID feedback (-100.0% to 100.0%)
24: PID deviation (-100.0% to 100.0%)
25: PID output (-100.0% to 100.0%)
26 to 30: Reserved</td></tr>
</table>
Byte10: Parameter high byte; Byte11: Parameter low byte
(Similar with other bytes)
</td></tr>
</table>

5.4.1 Parameter setting for Ethernet/IP connection

To operate the MV800 drive using MV810-EIP, users need to set the operation command channel and the frequency source of the MV800 drive to the bus communication card, as shown in the following table.

Drive parameter	Value	Function description
P02.02	2	Set the operation command channel to communication control
P02.03	3	Set the communication command channel to EtherNet/IP
P02.05	8	Set the main frequency source to EtherCAT, Profinet, CANopen, and EtherNet/IP

Settings of IP address (IP, subnet mask, and gateway) are shown in the table below.

Drive parameter	Value	Function
P40.02	0 to 255	IP address 1
P40.03	0 to 255	IP address 2

Drive parameter	Value	Function
P40.04	0 to 255	IP address 3
P40.05	0 to 255	IP address 4
P40.06	0 to 255	Subnet mask 1
P40.07	0 to 255	Subnet mask 2
P40.08	0 to 255	Subnet mask 3
P40.09	0 to 255	Subnet mask 4
P40.10	0 to 255	Gateway 1
P40.11	0 to 255	Gateway 2
P40.12	0 to 255	Gateway 3
P40.13	0 to 255	Gateway 4

5.5 Fault diagnosis

5.5.1 LED indicator description and fault removal

MV810-EIP has LED indicators on five locations, including the LED indicators on the PCBA of the expansion box for indicating the function status and the power status, and the LED indicators of the communication ports for indicating whether the communication connection status of MV810-EIP is normal.

Description of LED on the PCBA of the expansion box:

LED4 (Red) status	Description	Action
Off	Normal	No need for actions
Steady on	Communication timeout between the master and the communication card	Check whether the EIP option is properly connected to the drive.

Description of LED on the communication port:

LED status	Description	Action
Yellow light flashing	Normal connection with data transmission available	No need for actions

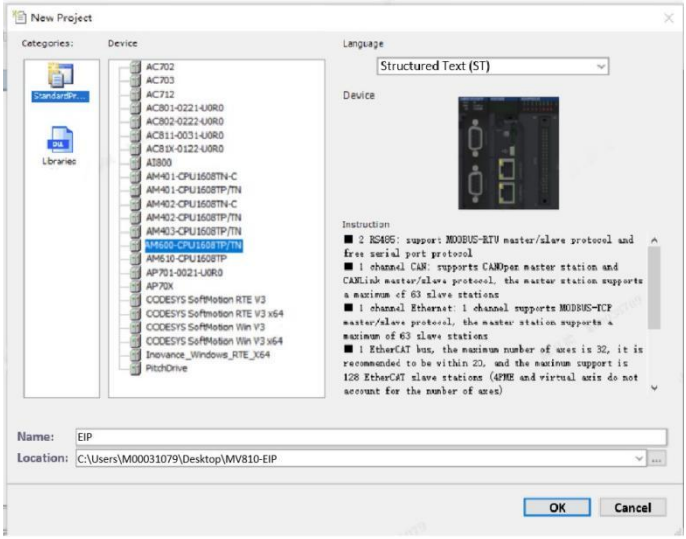
LED status	Description	Action
Green light steady on	Normal connection	No need for actions
Yellow light steady on	Normal connection without data transmission	Check whether there is communication between the master and the slave.
Green light off	Connection failed	Check whether the cable is properly connected.

5.6 Communication example of Inovance AM600 PLC controlling MV800

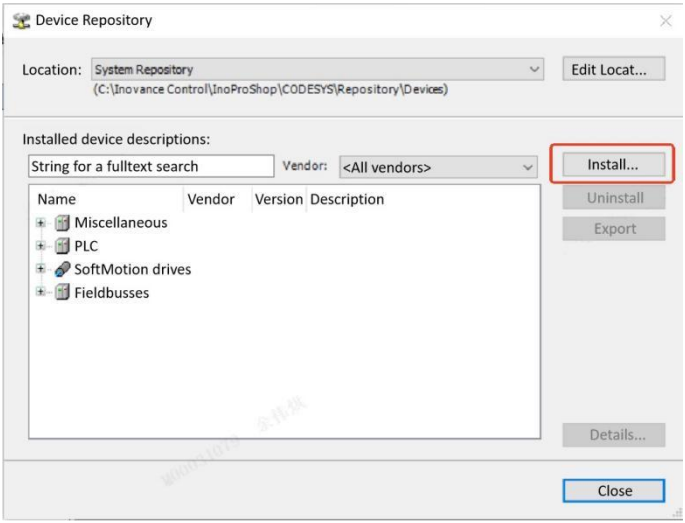
The following example is based upon an Inovance AM600 PLC as the master to demonstrate the configuration and usage of the MV800 Ethernet/IP communication.

STEP 1: create a new project

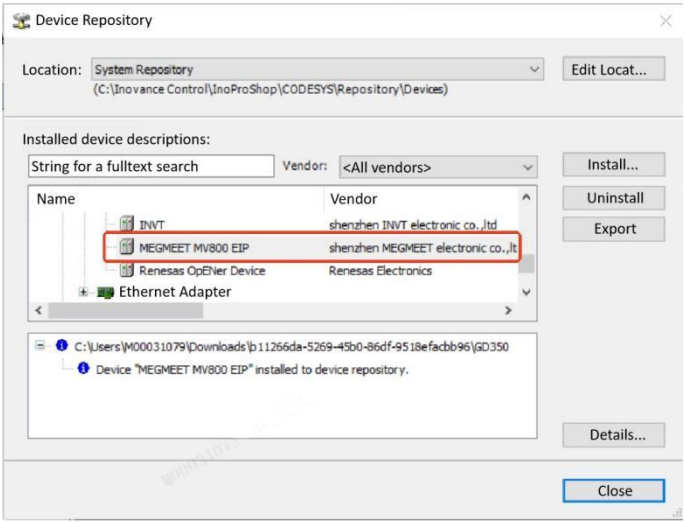
Open the "New Project" window in the InoProShop software interface, and select "AM600-CPU1608TP/TN" in the "Device" list, as shown in the following image.



STEP 2: import the "MEGMEET_MV800_EthernetIP_V1.01.EDS" file as shown in the following image.

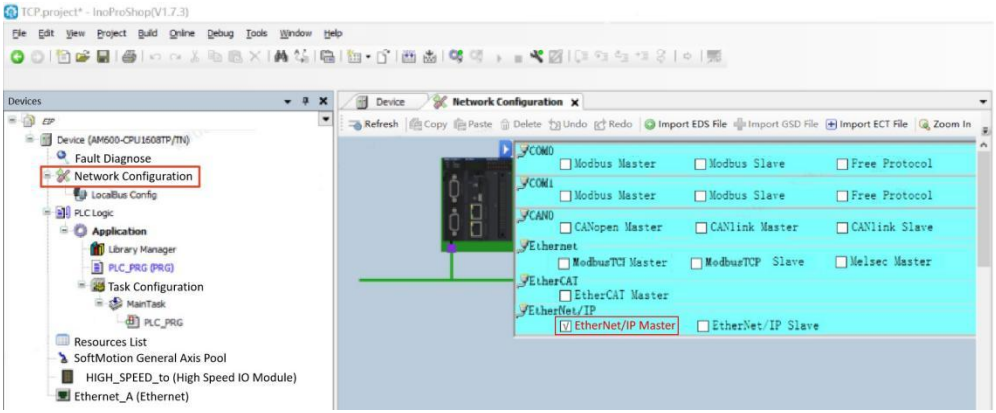


When the file is successfully imported, the following view will be displayed.

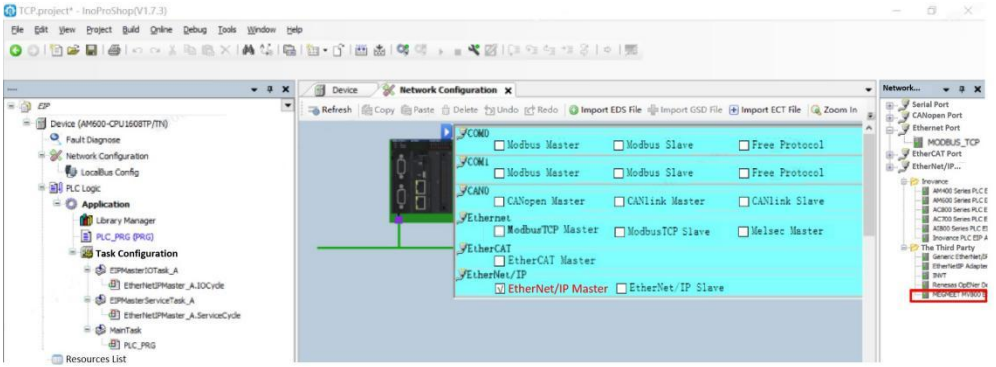


STEP 3: configure the network

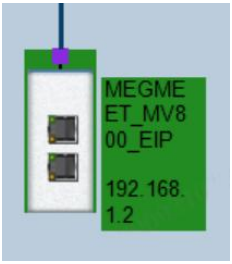
Open the "Network Configuration" interface, and select "EtherNet/IP Master" as the current PLC communication protocol, as shown in the following image.



Double click "MEGMEET MV800 EIP" in the right column to add the slave, as shown in the following image.

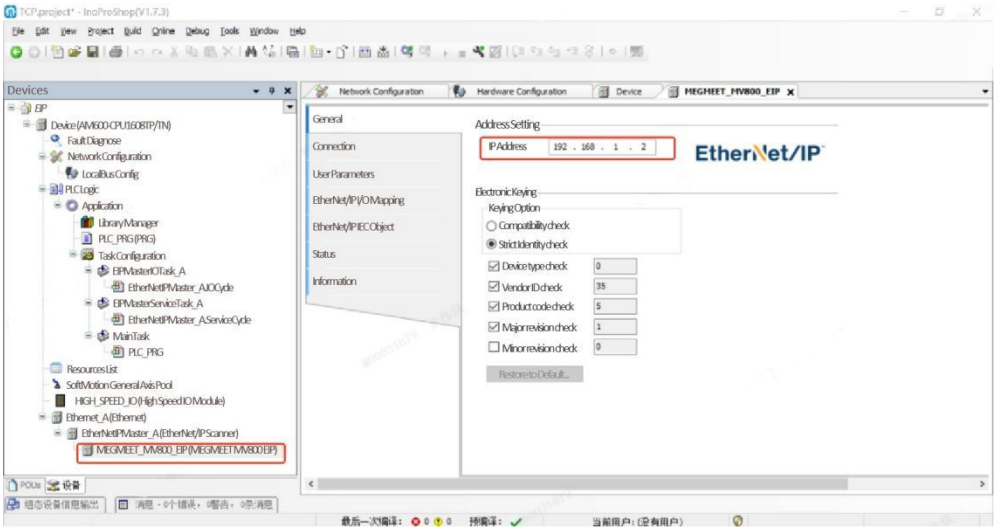


When the slave is added successfully, the following view will be displayed.



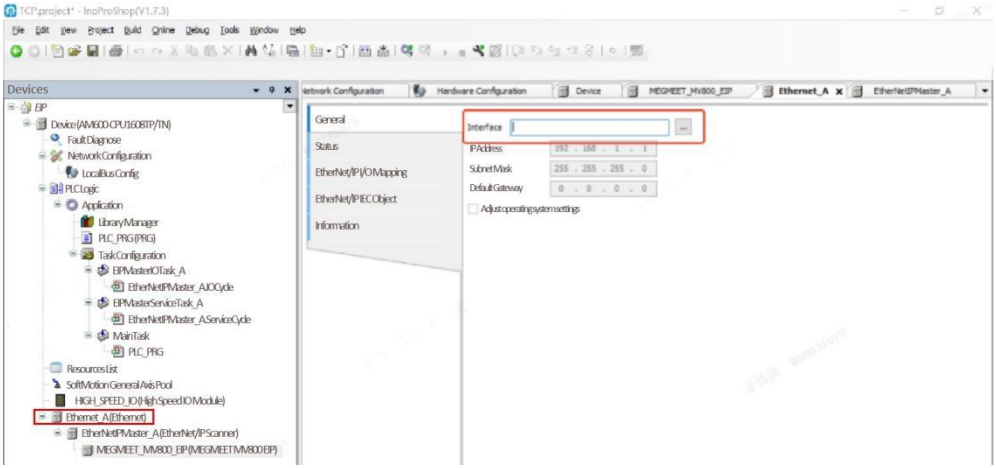
STEP 4: configure the slave parameters

Set the slave IP address, as shown in the following image.

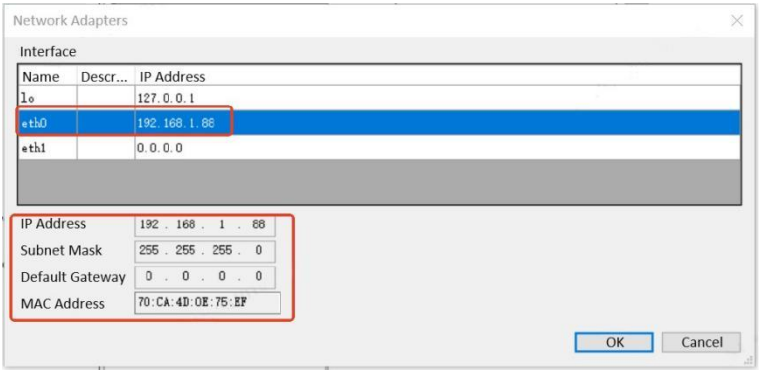


STEP 5: configure master IP

Select the Internet interface for the master, as shown in the following image.

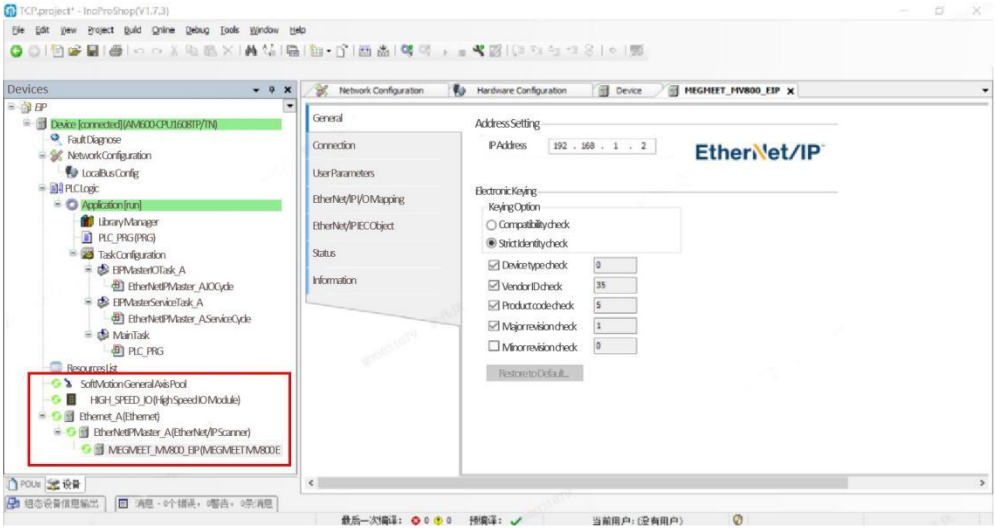


Double click "eth0", and the information, including the IP address, subnet mask, etc., will be automatically filled. The value of the IP address will be in consistency with that of "eth0", as shown in the following image.

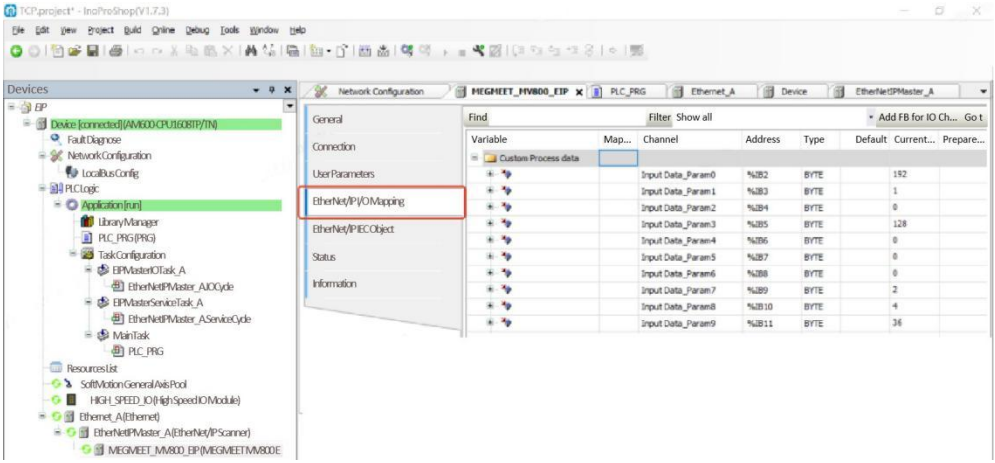


STEP 6: when the project is saved, build and download it to PLC.

Save the project, build and download it to PLC, and confirm the success of EIP communication connection between PLC and MV800 drive by checking whether there are green marks before the master and slave items in the left project tree, as shown in the image below.



Check the read/write variables in the "EtherNet/IP I/O Mapping" interface, as shown in the image below.


TCP.project* - InoProShop(V1.7.3)
File Edit View Project Build Online Debug Tools Window Help
Devices
EIP
Device [connected] (AM600-CPU1608TP/TN)
Fault Diagnose
Network Configuration
LocalBus Config
PLC Logic
Application [run]
Library Manager
PLC_PRG (PRG)
Task Configuration
EIPMasterIOTask_A
EtherNetIPMaster_AIOCycle
EIPMasterServiceTask_A
EtherNetIPMaster_AServiceCycle
IMainTask
PLC_PRG
Resources List
SoftMotion General Axis Pool
HIGH_SPEED_IO (High Speed IO Module)
Ethernet_A (Ethernet)
EtherNetIPMaster_A (EtherNet/IP Scanner)
MEGMEET_MV800_EIP (MEGMEET MV800 E
Network Configuration
MEGMEET_MV800_EIP
PLC_PRG
Ethernet_A
Device
EtherNetIPMaster_A
General
Connection
User Parameters
EtherNet/IP I/O Mapping
EtherNet/IP IEC Object
Status
Information
Find
Filter Show all
Add FB for IO Ch...
Go t
Variable
Map...
Channel
Address
Type
Default
Current...
Prepare...
Custom Process data
Input Data_Param0 %IB2 BYTE 192
Input Data_Param1 %IB3 BYTE 1
Input Data_Param2 %IB4 BYTE 0
Input Data_Param3 %IB5 BYTE 128
Input Data_Param4 %IB6 BYTE 0
Input Data_Param5 %IB7 BYTE 0
Input Data_Param6 %IB8 BYTE 0
Input Data_Param7 %IB9 BYTE 2
Input Data_Param8 %IB10 BYTE 4
Input Data_Param9 %IB11 BYTE 36

Appendix I EtherCAT Object Dictionary

Index	Sub-index	Description	Access rights	Data type	Default value
1000h	0	Device type	RO	UINT32	0x00000402
1001h	0	Error register	RO	UINT8	0
1008h	0	Manufacturer device name	RO	String	MV800_ECAT_CoE
1009h	0	Manufacturer hardware version	RO	String	Determined by the hardware version
100Ah	0	Manufacturer software version	RO	String	Determined by the software version
1018h	ID object				
	0	The biggest sub-index included	RO	UINT8	4
	1	Supplier ID	RO	UINT32	0x000006AE
	2	Product code	RO	UINT32	0x00000902
	3	Revision number	RO	UINT32	0x00000200
	4	Serial number	RO	UINT32	0x00000000
1600h	RX PDO1 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0x60400010
	2	The second mapping object	RW	UINT32	0x210B0010
	3	The third mapping object	RW	UINT32	0x210D0010
	4	The fourth mapping object	RW	UINT32	0x210C0010
	5	The fifth mapping object	RW	UINT32	0x00000000
	6	The sixth mapping object	RW	UINT32	0x00000000
	7	The seventh mapping object	RW	UINT32	0x00000000
	8	The eighth mapping object	RW	UINT32	0x00000000
	9	The ninth mapping object	RW	UINT32	0x00000000

	10	The tenth mapping object	RW	UINT32	0x00000000
1601h	RX PDO2 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	2
	1	The first mapping object	RW	UINT32	0x60400010
	2	The second mapping object	RW	UINT32	0x210B0010
1602h	RX PDO3 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	2
	1	The first mapping object	RW	UINT32	0x60400010
	2	The second mapping object	RW	UINT32	0x210D0010
1603h	RX PDO4 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	2
	1	The first mapping object	RW	UINT32	0x60400010
	2	The second mapping object	RW	UINT32	0x210C0010
1A00h	TX PDO1 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	7
	1	The first mapping object	RW	UINT32	0x60410010
	2	The second mapping object	RW	UINT32	0x603F0010
	3	The third mapping object	RW	UINT32	0x22020010
	4	The fourth mapping object	RW	UINT32	0x22000010
	5	The fifth mapping object	RW	UINT32	0x22030010
	6	The sixth mapping object	RW	UINT32	0x22040010
	7	The seventh mapping object	RW	UINT32	0x220A0010
	8	The eighth mapping object	RW	UINT32	0x00000000
	9	The ninth mapping object	RW	UINT32	0x00000000
	10	The tenth mapping object	RW	UINT32	0x00000000

1A01h	TX PDO2 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	2
	1	The first mapping object	RW	UINT32	0x60410010
	2	The second mapping object	RW	UINT32	0x220A0010
1A02h	TX PDO3 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	2
	1	The first mapping object	RW	UINT32	0x60410010
	2	The second mapping object	RW	UINT32	0x22020010
1A03h	TX PDO4 mapping parameter				
	0	Number of supported mapping objects	RW	UINT8	2
	1	The first mapping object	RW	UINT32	0x60410010
	2	The second mapping object	RW	UINT32	0x22030010
1C00h	Synchronized management of communication types				
	0	The biggest sub-index	RO	UINT8	4
	1	SM0 communication type	RO	UINT8	0x01
	2	SM1 communication type	RO	UINT8	0x02
	3	SM2 communication type	RO	UINT8	0x03
	4	SM3 communication type	RO	UINT8	0x04
1C12h	RxPDO distribution				
	0	The biggest sub-index	RW	UINT8	1
	1	Object index distributed by RxPDO	RW	UINT16	0x1600
1C13h	TxPDO distribution				
	0	The biggest sub-index	RW	UINT8	1
	1	TxPDO distribution	RW	UINT16	0x1A00
1C32h	Synchronized management of synchronously output parameters				

	0x00	The biggest sub-index	RO	UINT8	0x20
	0x01	Sync mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switching time	RO	UINT32	0
	0x04	Synchronization type supported	RO	UINT16	0x4006
	0x05	The shortest cycle time	RO	UINT32	0x0003D090
	0x06	Calculate/Copy time	RO	UINT32	0
	0x07	Reserve	RW	UINT32	0
	0x08	Learn cycle time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0
	0x0A	Sync0 time	RW	UINT32	-
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Loop timeout counter	RO	UINT32	0
	0x0D	Switching time too short counter	RO	UINT32	0
	0x20	Sync error	RO	UINT8	0
1C33h	Synchronized management of synchronously input parameters				
	0x00	The biggest sub-index	RO	UINT8	0x20
	0x01	Sync mode	RW	UINT16	0x02
	0x02	Cycle time	RO	UINT32	0
	0x03	Switching time	RO	UINT32	0
	0x04	Synchronization type supported	RO	UINT16	0x4006
	0x05	The shortest cycle time	RO	UINT32	0x0003D090
	0x06	Calculate/Copy time	RO	UINT32	0
	0x07	Reserve	RW	UINT32	0
	0x08	Learn cycle time	RW	UINT16	0
	0x09	Delay time	RO	UINT32	0

	0x0A	Sync0 time	RW	UINT32	-
	0x0B	SM event loss counter	RO	UINT32	0
	0x0C	Loop timeout counter	RO	UINT32	0
	0x0D	Switching time too short counter	RO	UINT32	0
	0x20	Sync error	RO	UINT8	0
2000h		P00 function code group			
2001h		P01 function code group			
2002h		P02 function code group			
2003h		P03 function code group			
2004h		P04 function code group			
2005h		P05 function code group			
2006h		P06 function code group			
2007h		P07 function code group			
2008h		P08 function code group			
2009h		P09 function code group			
200Ah		P10 function code group			
200Bh		P11 function code group			
200Ch		P12 function code group			
200Dh		P13 function code group			
200Eh		P14 function code group			
200Fh		P15 function code group			
2010h		P16 function code group			
2012h		P18 function code group			
2014h		P20 function code group			
2015h		P21 function code group			
2016h		P22 function code group			
2017h		P23 function code group			

2018h		P24 function code group			
201Ah		P26 function code group			
2028h		P40 function code group			
2029h		P41 function code group			
202Bh		P43 function code group			
2032h		P50 function code group			
2061h		P97 function code group			
2062h		P98 function code group			
2064h	Function code reading/writing error indication				
	0	Number of sub-indexes	RO	UINT8	2
	1	Error code	RO	UINT32	0
	2	Index of the function code with error	RO	UINT32	0
2100h	0	Drive torque upper limit	RW	UINT16	0
2101h	0	Braking torque upper limit	RW	UINT16	0
2102h	0	FWD frequency upper limit	RW	UINT16	0
2103h	0	REV frequency upper limit	RW	UINT16	0
2104h	0	Voltage reference (V/F separation)	RW	UINT16	0
2105h	0	DO	RW	UINT16	0
2106h	0	AO1	RW	UINT16	0
2107h	0	HDO1	RW	UINT16	0
2108h	0	HDO2	RW	UINT16	0
2109h	0	PID reference value	RW	INT16	0
210Ah	0	PID feedback value	RW	INT16	0
210Bh	0	Position reference	RW	UINT16	0
210Ch	0	Torque reference	RW	INT16	0
210Dh	0	Frequency reference	RW	UINT16	0

2200h	0	Output current	RO	UINT16	0
2201h	0	Output voltage	RO	UINT16	0
2202h	0	Output frequency	RO	UINT16	0
2203h	0	Output torque	RO	INT16	0
2204h	0	Bus voltage	RO	UINT16	0
2205h	0	DI state 1	RO	UINT16	0
2206h	0	DI state 2	RO	UINT16	0
2207h	0	DO state	RO	UINT16	0
2208h	0	Motor power	RO	INT16	0
2209h	0	Power output	RO	UINT16	0
220Ah	0	Position actual value	RO	UINT16	0
603Fh	0	Error code	RO	UINT16	0
6040h	0	Control word	RW	UINT16	0
6041h	0	Status word	RO	UINT16	0

Appendix II CANopen Object Dictionary

Index	Sub-index	Description	Access rights	Data type	Default value
1000h	0	Device type	RO	UINT32	0x00000320
1001h	0	Error register	RO	UINT8	0
1003h	Error code register				
	0	Number of errors	RW	UINT8	0
	1	Error code	RO	UINT32	0
	2	Error code	RO	UINT32	0
	3	Error code	RO	UINT32	0
	4	Error code	RO	UINT32	0
	5	Error code	RO	UINT32	0
	6	Error code	RO	UINT32	0
	7	Error code	RO	UINT32	0
	8	Error code	RO	UINT32	0
1005h	0	SYNC COB ID	RW	UINT32	0x80
100Ch	0	Protection time	RW	UINT16	0
100Dh	0	Life cycle factor	RW	UINT8	0
1014h	0	Emergency COB ID	RW	UINT32	0x80+Node-ID
1017h	0	Producer heartbeat time	RW	UINT16	0
1018h	0	Number of sub-indexes	RO	UINT8	4
	1	Manufacturer ID	RO	UINT32	0x264
	2	Product code	RO	UINT32	0x320
	3	Revision number	RO	UINT32	0x01
	4	Serial number	RO	UINT32	0
1018h	ID object				
	0	Number of sub-indexes	RO	UINT8	4
	1	Supplier ID	RO	UINT32	0x000006AE

	2	Product code	RO	UINT32	0x00000902
	3	Revision number	RO	UINT32	0x00000200
	4	Serial number	RO	UINT32	0x00000000
1200h	Server SDO				
	0	Number of sub-indexes	RO	UINT8	2
	1	COB ID client to server	RO	UINT32	0x600+Node-ID
	2	COB ID server to client	RO	UINT32	0x580+Node-ID
1400h	RxPDO1 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x200+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0
	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1401h	RxPDO2 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x80000300+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0
	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1402h	RxPDO3 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x80000400+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0

	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1403h	RxPDO4 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x80000500+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0
	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1400h	RxPDO1 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0x60400010
	2	The second mapping object	RW	UINT32	0x210D0010
	3	The third mapping object	RW	UINT32	0
	4	The fourth mapping object	RW	UINT32	0
1401h	RxPDO2 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0
	2	The second mapping object	RW	UINT32	0
	3	The third mapping object	RW	UINT32	0
	4	The fourth mapping object	RW	UINT32	0
1402h	RxPDO3 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0
	2	The second mapping object	RW	UINT32	0
	3	The third mapping object	RW	UINT32	0

	4	The fourth mapping object	RW	UINT32	0
1403h	RxPDO4 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0
	2	The second mapping object	RW	UINT32	0
	3	The third mapping object	RW	UINT32	0
	4	The fourth mapping object	RW	UINT32	0
1800h	TxPDO1 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x180+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0
	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1801h	TxPDO2 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x80000280+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0
	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1802h	TxPDO3 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x80000380+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0

	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1803h	TxPDO4 communication parameter				
	0	Number of sub-indexes	RO	UINT8	6
	1	COB-ID used by PDO	RW	UINT32	0x80000480+Node-ID
	2	Transmission type	RW	UINT8	0xFF
	3	Inhibit time	RW	UINT16	0
	4	/	/	UINT8	/
	5	Event timer	RW	UINT16	0
	6	/	/	UINT8	/
1A00h	TxPDO1 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0x60410010
	2	The second mapping object	RW	UINT32	0x22020010
	3	The third mapping object	RW	UINT32	0
	4	The fourth mapping object	RW	UINT32	0
1A01h	TxPDO2 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0
	2	The second mapping object	RW	UINT32	0
	3	The third mapping object	RW	UINT32	0
	4	The fourth mapping object	RW	UINT32	0
1A02h	TxPDO3 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0
	2	The second mapping object	RW	UINT32	0
	3	The third mapping object	RW	UINT32	0

	4	The fourth mapping object	RW	UINT32	0
1A03h	TxPDO4 mapping parameter				
	0	Number of sub-indexes	RW	UINT8	4
	1	The first mapping object	RW	UINT32	0
	2	The second mapping object	RW	UINT32	0
	3	The third mapping object	RW	UINT32	0
	4	The fourth mapping object	RW	UINT32	0
2000h		P00 function code group			
2001h		P01 function code group			
2002h		P02 function code group			
2003h		P03 function code group			
2004h		P04 function code group			
2005h		P05 function code group			
2006h		P06 function code group			
2007h		P07 function code group			
2008h		P08 function code group			
2009h		P09 function code group			
200Ah		P10 function code group			
200Bh		P11 function code group			
200Ch		P12 function code group			
200Dh		P13 function code group			
200Eh		P14 function code group			
200Fh		P15 function code group			
2010h		P16 function code group			
2012h		P18 function code group			
2014h		P20 function code group			
2015h		P21 function code group			
2016h		P22 function code group			

2017h		P23 function code group			
2018h		P24 function code group			
201Ah		P26 function code group			
2028h		P40 function code group			
2029h		P41 function code group			
202Bh		P43 function code group			
2032h		P50 function code group			
2061h		P97 function code group			
2062h		P98 function code group			
2064h	Function code reading/writing error indication				
	0	Number of sub-indexes	RO	UINT8	2
	1	Error code	RO	UINT32	0
	2	Index of the function code with error	RO	UINT32	0
2100h	0	Drive torque upper limit	RW	UINT16	0
2101h	0	Braking torque upper limit	RW	UINT16	0
2102h	0	FWD frequency upper limit	RW	UINT16	0
2103h	0	REV frequency upper limit	RW	UINT16	0
2104h	0	Voltage reference (V/F separation)	RW	UINT16	0
2105h	0	DO	RW	UINT16	0
2106h	0	AO1	RW	UINT16	0
2107h	0	HDO1	RW	UINT16	0
2108h	0	HDO2	RW	UINT16	0
2109h	0	PID reference value	RW	INT16	0
210Ah	0	PID feedback value	RW	INT16	0
210Bh	0	Position reference	RW	UINT16	0
210Ch	0	Torque reference	RW	INT16	0

210Dh	0	Frequency reference	RW	UINT16	0
2200h	0	Output current	RO	UINT16	0
2201h	0	Output voltage	RO	UINT16	0
2202h	0	Output frequency	RO	UINT16	0
2203h	0	Output torque	RO	INT16	0
2204h	0	Bus voltage	RO	UINT16	0
2205h	0	DI state 1	RO	UINT16	0
2206h	0	DI state 2	RO	UINT16	0
2207h	0	DO state	RO	UINT16	0
2208h	0	Motor power	RO	INT16	0
2209h	0	Power output	RO	UINT16	0
220Ah	0	Position actual value	RO	UINT16	0
603Fh	0	Error code	RO	UINT16	0
6040h	0	Control word	RW	UINT16	0
6041h	0	Status word	RO	UINT16	0

Appendix III Warranty and Service

Megmeet rigorously adheres to the ISO 9001:2008 standard in manufacturing motor drive products. If any irregularities occur with our products, please contact the product supplier or the headquarters directly. Megmeet is committed to delivering comprehensive technical support services to all our clients.

1. Warranty period

The warranty period for the product is 18 months from the date of purchase, but not exceeding 24 months after the manufacturing date recorded on the nameplate.

2. Warranty scope

During the warranty period, any abnormalities arising from the responsibility of our company can be repaired or replaced free of charge by our company. However, a certain amount of repair charges may apply even within the warranty period under the following circumstances:

(1) Damage caused by fire, flood, severe lightning strikes, or similar reasons;

(2) Man-made damage caused by users' unauthorized modifications;

(3) Damage due to dropping or transportation after purchase;

(4) Damage caused by usage beyond the standard specifications or requirements;

(5) Damage resulting from operation/use not in accordance with the user manual.

3. After-sales service

(1) If there are special requirements for the installation and commissioning of the drive product, or if the product's performance or functionality is not satisfactory, please contact the product distributor or Megmeet.

(2) In case of any abnormalities, please seek assistance by contacting the product supplier or Megmeet.

(3) During the warranty period, any abnormalities caused by manufacturing and design defects will be repaired free of charge by our company.

(4) Beyond the warranty period, repairs will be conducted at the customer's request and charged by our company.

(5) Service fees are calculated based on actual costs. Any agreements in place will take precedence.

Shenzhen Megmeet Electrical Co., Ltd.

Add: 5th Floor, Block B, Unisplendour Information Harbor, Langshan Road, Shenzhen, 518057, China

Tel: +86-755-86600500

Fax: +86-755-86600562

Website: www.megmeet.com

Shenzhen Megmeet Electrical Co., Ltd.

Drive Warranty Bill

Customer company:	
Detailed address:	
Zip code:	Contact:
Tel:	Fax:
Machine model:	
Power:	Machine No.:
Contract No.:	Purchase date:
Service unit:	
Contact:	Tel:
Maintenance person:	Tel:
Maintenance date:	
Comment on service: ☐ **Excellent** ☐ **Good** ☐ **Fair** ☐ **Unsatisfactory** Other comment: User's signature: Date:	
Customer Service Center follow-up record: ☐ **Follow-up phone call** ☐ **Follow-up letter** Other: Signature of the technical support engineer: Date:	

Note: This bill becomes invalid if the user can not be visited.

Shenzhen Megmeet Electrical Co., Ltd.

Drive Warranty Bill

Customer company:	
Detailed address:	
Zip code:	Contact:
Tel:	Fax:
Machine model:	
Power:	Machine No.:
Contract No.:	Purchase date:
Service unit:	
Contact:	Tel:
Maintenance person:	Tel:
Maintenance date:	
Comment on service: ☐ **Excellent** ☐ **Good** ☐ **Fair** ☐ **Unsatisfactory** Other comment: User's signature: Date:	
Customer Service Center follow-up record: ☐ **Follow-up phone call** ☐ **Follow-up letter** Other: Signature of the technical support engineer: Date:	

Note: This bill becomes invalid if the user can not be visited.