



PD310CAN1

PROMPOWER

User Guide

**PROM
POWER**

Preface

Thank you for purchasing the PD310CAN1 card. This manual describes how to use the product properly to get good benefits. Please be sure to read this manual carefully before using the product (installation, wiring, operation, maintenance, inspection, etc.).

PD310CAN1 card is a CANopen fieldbus adapter card, which conforms to the internationally recognized CANopen protocol standard. The card is installed on PD310 series AC drive to improve the communication efficiency and facilitate the realization of AC drive networking function, so that the AC drive becomes the slave of the fieldbus and accepts the control of the master of the fieldbus.

This manual is based on PD310CANopen_V1.0.eds. EDS and its related documents can be downloaded from the official website. In the course of use, if you have any questions or special requirements, please feel free to contact our office or distributor, or directly with our customer service center, we will be happy to serve you.

Our company is committed to the continuous improvement of our products, therefore, the information related to this series is subject to change without prior notice. We apologize for any inconvenience this may cause.

Catalog

Preface	2
Safety Precautions	4
1 Product Information	5
1.1 CANopen Communication Card Product Feature	5
1.1.1 Supported CANopen services	5
1.1.2 Supported CANopen addresses and baud rates	5
1.1.3 Communication object COB-ID	5
1.1.4 CANopen slave information.....	6
1.1.5 Topology diagram.....	7
1.2 Physical and wiring	8
1.2.1 PD310CAN1 appearance	8
1.2.2 Expansion card interface layout diagram and interface description	8
1.2.3 Terminating Resistors.....	9
1.2.4 Status indicators	9
2 Communication Parameters and Protocols	10
2.1 CANopen communication parameter setting	10
2.2 CANopen protocol information frame data structure.....	12
2.2.1 Service Data Object SDO.....	12
2.2.2 PDO area data.....	14
3 Communication Configuration.....	16
3.1 Communication configuration in CANTest software	16
3.2 Device Configuration in CODESYS.....	23



Safety Precautions

The staff who install and operate this expansion card must have passed the professional electrical training and safety knowledge training and examination, and have been familiar with the procedures and requirements for installation, commissioning, putting into operation and maintenance of this equipment, and can avoid various emergencies.

Before installing, removing and operating the expansion card, please read this manual and the safety precautions section of the AC drive manual carefully to ensure that the operation is done in a safe manner.

We will not be responsible for any injury or equipment damage caused by the user's failure to observe the safety precautions in this manual and the AC drive manual.

- Installation or removal of this expansion card requires disassembly of the AC drive enclosure, so it is important to completely disconnect all power inputs to the AC drive and ensure that the internal voltage of the unit has been secured, as described in the AC drive manual. Failure to comply with this requirement may result in serious personal injury or even death.
- Must be stored in a place that is dustproof, moisture-proof, free from electric shock and free from mechanical stress.
- The expansion card is sensitive to static electricity and must be properly protected against static electricity during relevant operations to avoid damage to components.
- When installing this expansion card, be sure to align the interface to ensure no loosening and proper grounding.

1 Product Information

1.1 CANopen Communication Card Product Feature

1.1.1 Supported CANopen services

- 1) Support for PDO services
- 2) Support for SDO services
- 3) Support for heartbeat messages
- 4) Support for SYNC messages
- 5) Support for NMT (Network Management)
- 6) Support for Emergency Services
- 7) Supports manufacturer-defined object dictionaries for controlling and reading the AC drive's real-time values via SDO

1.1.2 Supported CANopen addresses and baud rates

The PD310CAN1 supports the communication addresses and baud rates shown in the table below.

Table 1-1 PD310CAN1CANopen address and baud rate

Items	Supported specifications
Address	1~127 (decimalism)
Baud rate	125k bps (Bit/sec)
	250k bps (Bit/sec)
	500k bps (Bit/sec)
	800k bps (Bit/sec)
	1000k bps (Bit/sec)

1.1.3 Communication object COB-ID

CANopen provides a variety of communication objects, each of which has different characteristics (specific reference can be made to the CANopen standard protocol), and can be used according to the merits of different applications. The AC drive slave using this communication card adopts the predefined COB-ID, and the specific rules are as follows:

1. NMT object: 0x000
2. SYNC object: 0x080
3. SDO Object:
 - 1) Send SDO——0x600+Node-ID
 - 2) Receive SDO——0x580+Node-ID
4. PDO object:
 - 1) RPDO1——0x200+Node-ID
 - 2) RPDO2——0x300+Node-ID
 - 3) RPDO3——0x400+Node-ID
 - 4) TPDO1——0x180+Node-ID
 - 5) TPDO2——0x280+Node-ID
 - 6) TPDO3——0x380+Node-ID
5. EMCY Objects: 0x80+Node-ID
 - 1) Node-ID: Device ID (station address), set by Fd-02 function code;
 - 2) The communication object COB-ID is in the form of a fixed assignment and cannot be modified. (Refer to CANopen protocol related information)

1.1.4 CANopen slave information

The CANopen slave information file (EDS file) is read by the master and is used to build the master-slave configuration. The "PD310CANopen_V1.0.eds" file contains the information necessary for CANopen communication setup.

1.1.5 Topology diagram

CAN bus connection topology is shown in the figure below, CANH, CANL is recommended to use twisted shielded wire connection. only at both ends of the bus were connected to 120Ω termination resistors to prevent signal reflection. all nodes CAN signals are connected to the reference ground. up to 127 nodes are connected.

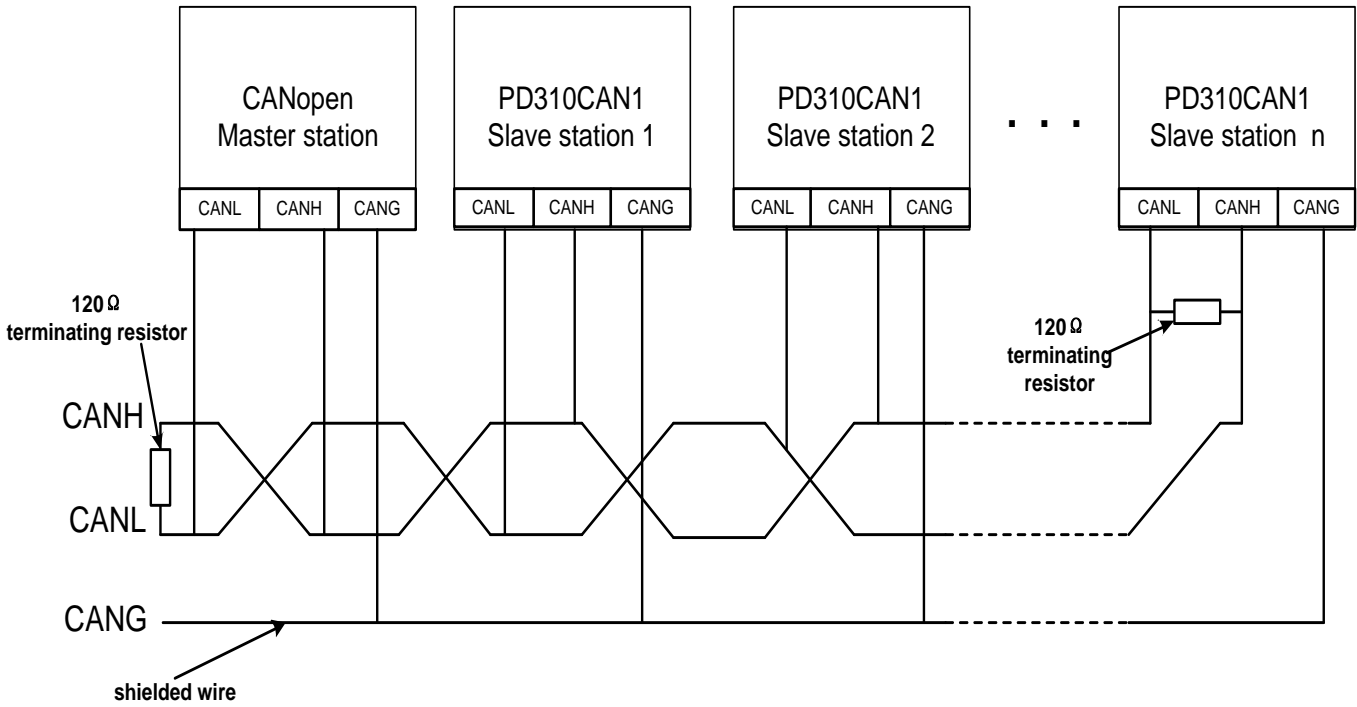


Figure 1-1 CANOpen bus connection topology

1.2 Physical and wiring

1.2.1 PD310CAN1 appearance



Figure 1-2 PD310CAN1 appearance

1.2.2 Expansion card interface layout diagram and interface description

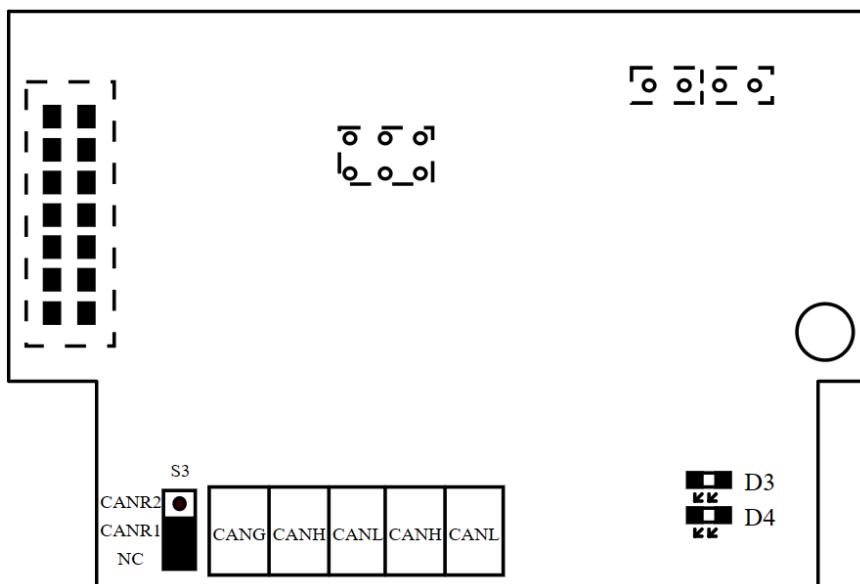


Figure 1-3 Interface layout diagram

1.2.3 Terminating Resistors

When the PD310CAN1 card is used as a CANopen network terminal slave, it is recommended to select the termination resistor to be connected. the jump cap S3 indicates 120 ohm termination resistor access to the network when CANR1 and CANR2 are selected.

1.2.4 Status indicators

The CANopen communication card is set with 2 LEDs to indicate the different states of this communication card (refer to the interface layout diagram of the expansion card), which are defined in the following table.

Table 1-2 LED Status Definitions

LED	Color	Status	Description
D3	Green	ON	PD310CAN1 power on
		OFF	PD310CAN1 is not plugged in
D4	Green	ON	CANopen communication connection failed
		Blinking	CANopen communication connection
		OFF	CANopen communication connection is successful

2 Communication Parameters and Protocols

2.1 CANopen communication parameter setting

Install the PD310CAN1 card correctly on the AC drive and complete the relevant communication parameter configuration before establishing communication with the AC drive.

Note:

Please power on and off again after each change of baud rate or mailing address.

Table 2-1 Communication card configuration parameters

Function Code	Name	Content	Set value	Description
F0-00	Running command source selection	0: Operation keypad control (LED "L/R" off) 1: Terminal control (LED "L/R" on) 2: Communication control (LED "L/R" blinking)	2	Running command source selection
F0-02	Main frequency source X selection	0: UP/DOWN setting (non-recorded after stop) 1: UP/DOWN setting (retentive after stop) 2: AI1 3: AI2 4: Multi-speed 5: Simple PLC 6: PID 7: Communication setting 8: Pulse setting 9: Up/Down Change frequency Shutdown memory, power failure memory not 10: Keyboard potentiometer	7	Primary frequency source select communication
Fd-00	Baud rate	Bit: Modbus baud rate 0: 300bps 1: 600bps 2: 1200bps 3: 2400bps 4: 4800bps 5: 9600bps 6: 19200bps 7: 38400bps Ten bits: CANopen baud rate 0: 125K 1: 250K 2: 500K 3: 800K 4: 1M	25	The CAN baud rate defaults to 500K

Function Code	Name	Content	Set value	Description
Fd-02	Local address	0 to 247 (0 is Modbus Broadcast address. Profibus-DP supports only 1 to 127)	3	The default node address is 3
Fd-06	Communication protocol selection	0: MODbus RTU 1: Profibus-DP 2: CANopen 3: PROFINET 4: MODBUS TCP 5: EtherCAT	2	The communication protocol is CANopen communication
Fd-10~ Fd-19	Receive PZD3~ Receive PZD12	0~65535	0	Process data free allocation area (RPDO), the default value is 0, each function individually set an operation address to receive PZD for CANopen master to write AC drive. For example, if the PLC master needs to write the AC drive function code F0-20, it is necessary to merge and convert F0-20 to hexadecimal, then to decimal, and then input into the function code. Note: F0-20→0xF014→61460
Fd-20~ Fd-29	Send PZD3~ Send PZD12	0~65535	0	Process Data Free Distribution Area (TPDO), the default value is 0. Each function sets an operation address individually and sends PZD for CANopen master to read AC drive. For example, PLC master needs to read AC drive function code F0-07 (preset frequency), it is necessary to merge and convert F0-07 to hexadecimal, then to decimal, and then input into the function code. Note: F0-07→0xF007→61447

2.2 CANopen protocol information frame data structure

The CANopen bus communication method allows fast data exchange between the master (PLC) and the slave (AC drive). The reading and writing of the AC drive data is always carried out in a master-slave manner, the AC drive is fixed as a slave, and each slave itself has a clear address. This message is transmitted using 16 words (16 bits), and the structure is shown in the figure:

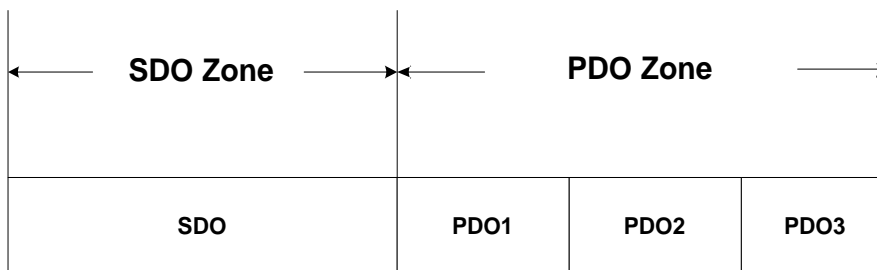


Figure 2-1 CANopen Message Frame Data Structure

2.2.1 Service Data Object SDO

The SDO of CANopen is used to transmit non-time critical data, such as the configuration of communication parameters and the configuration of AC drive operation parameters, which requires searching the object dictionary when reading and writing, and in the process of reading and writing, due to the existence of the concept of sub-index 0, it is necessary to add 1 on top of the sub-index in the actual operation.

Example of SDO operation

Note:

AC drive CANopen address is "0x03" for example.

1. Read AC drive command source channel

Read the AC drive function code "F0-00", the master station sends CANopen message as shown in the table.

Read the F0-00 master transmit message.

Message identification (Hex)	RTR	Data (Hex)
0x603	0	40F0200100000000

The AC drive responds to the CANopen message as shown in the table. The value of function code F0-00 is "0x0002" and the current command source of the AC drive is the communication channel.

Read F0-00 AC drive Response Message

Message identification (Hex)	RTR	Data (Hex)
0x583	0	4BF0200102000000

2. Setting the Command Source Channel to the Operator Panel

The command channel is set to the operation panel, the write frequency conversion function code F0-00 is "0", and the master station sends CANopen messages as shown in the table.

Write F0-00 Master Send Message

Message identification (Hex)	RTR	Data (Hex)
0x603	0	2BF0200100000000

The AC drive responds to the CANopen message as shown in the table, the function code F0-00 rewrites the value "0", and the current command source is set to the panel channel.

Write F0-00 AC drive response message

Message identification (Hex)	RTR	Data (Hex)
0x583	0	60F0200100000000

2.2.2 PDO area data

PDO area data realizes real-time master station to change and read AC drive data and periodic data interaction, default transmission type: 0xFE: asynchronous, manufacturer-specific event, event timer period is 500ms. the communication address of PDO data is configured by the AC drive directly, which is divided into 3 PDOs, and there are 4 parameters for each PDO, among which the first two parameters of PDO1 have been fixed, and the parameters of the last two PDOs and PDO2 and PDO3 can be set by the user through the Fd group parameter. The last two parameters of PDO1 and the parameters of PDO2 and PDO3 can be set by the user through the Fd group parameters, and the user does not need to change the mapping corresponding to the PDOs in the EDS file. It mainly contains the following contents:

- a. AC drive control commands, target frequency real-time giving
- b. Real-time reading of the current state and operating frequency of the AC drive
- c. Real-time interaction of function parameters and monitoring parameter data between the AC drive and the CANopen master station.

The PDO process data mainly accomplishes the periodic data interaction between the master station and the AC drive, and the interaction data are listed in the following table:

Table 2-2 PDO Zone Interaction Data

RPDO (The master sends data to the slave)				
RPDO group	Parameter 1	Parameter 2	Parameter 3	Parameter 4
RPDO1	AC drive command	AC drive target frequency	Fd-10 setting	Fd-11 setting
RPDO2	Fd-12 setting	Fd-13 setting	Fd-14 setting	Fd-15 setting
RPDO3	Fd-16 setting	Fd-17 setting	Fd-18 setting	Fd-19 setting
TPDO (The slave sends data to the master)				
TPDO group	Parameter 1	Parameter 2	Parameter 3	Parameter 4
TPDO1	AC drive status	AC drive operation frequency	Fd-20 setting	Fd-21 setting
TPDO2	Fd-22 setting	Fd-23 setting	Fd-24 setting	Fd-25 setting
TPDO3	Fd-26 setting	Fd-27 setting	Fd-28 setting	Fd-29 setting

Examples of PDO operations

Note:

The CANopen address of the AC drive is "0x03" for example.

1. Modifying AC drive parameters via RPDO1

Start the AC drive and modify the operating frequency to 50.00 Hz at the same time.

Message identification (Hex)	RTR	Data (Hex)
0x203	0	0100 8813 0000 0000

2. To modify the AC drive parameter F0-07 through RPDO2, you need to set the first parameter of RPDO2 to F0-07 and the RAM address to 0x0007, that is, set Fd-12 = 0007.

Message identification (Hex)	RTR	Data (Hex)
0x303	0	8813 0000 0000 0000

3. Reading AC drive parameters via TPDO1

The first parameter read is 0x0003: indicating that the current state of the AC drive is positive operation.

The second parameter is read as 0x1388: indicating that the AC drive runs at 50.00Hz (two decimal places).

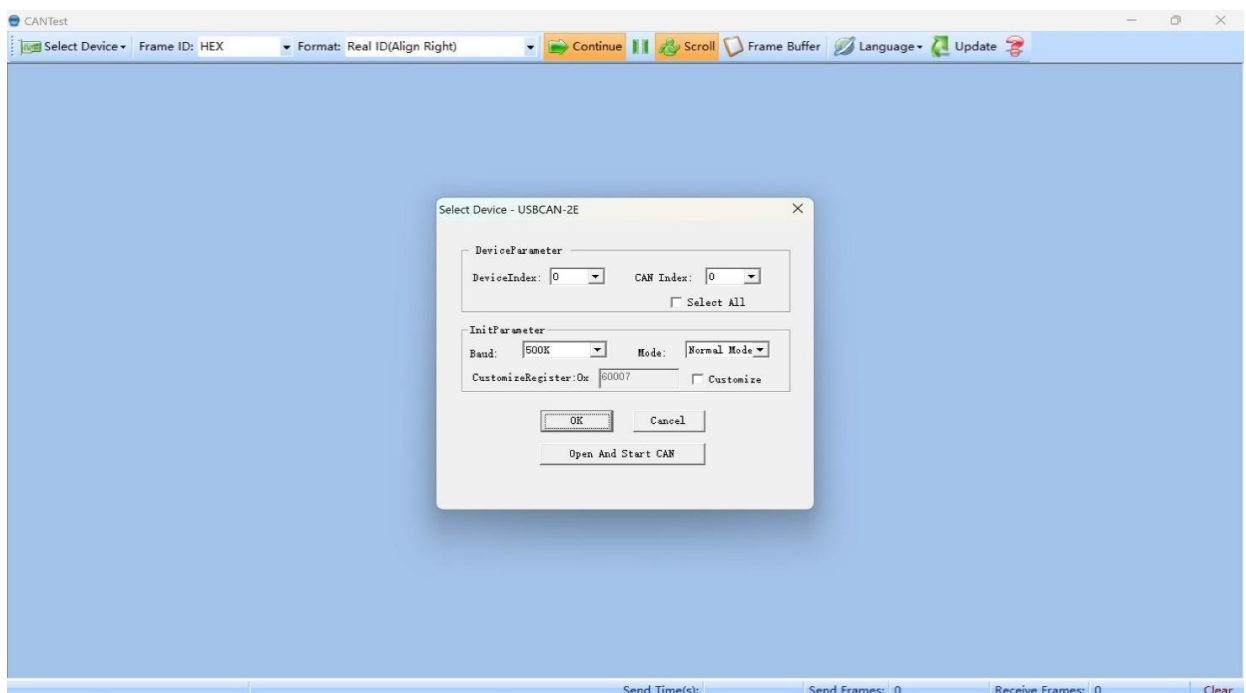
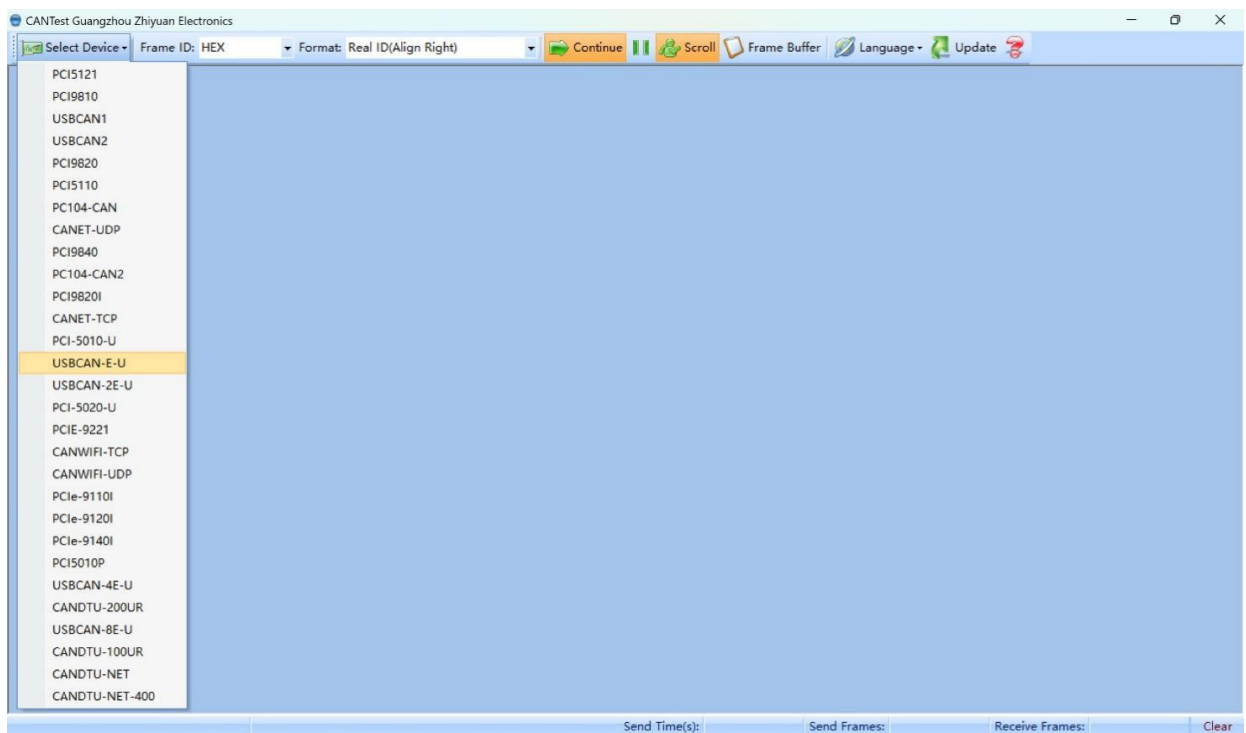
Reading to the last two parameters requires the user to set the corresponding addresses via Fd-20 and Fd-21.

Message identification (Hex)	RTR	Data (Hex)
0x183	0	0100 8813 0000 0000

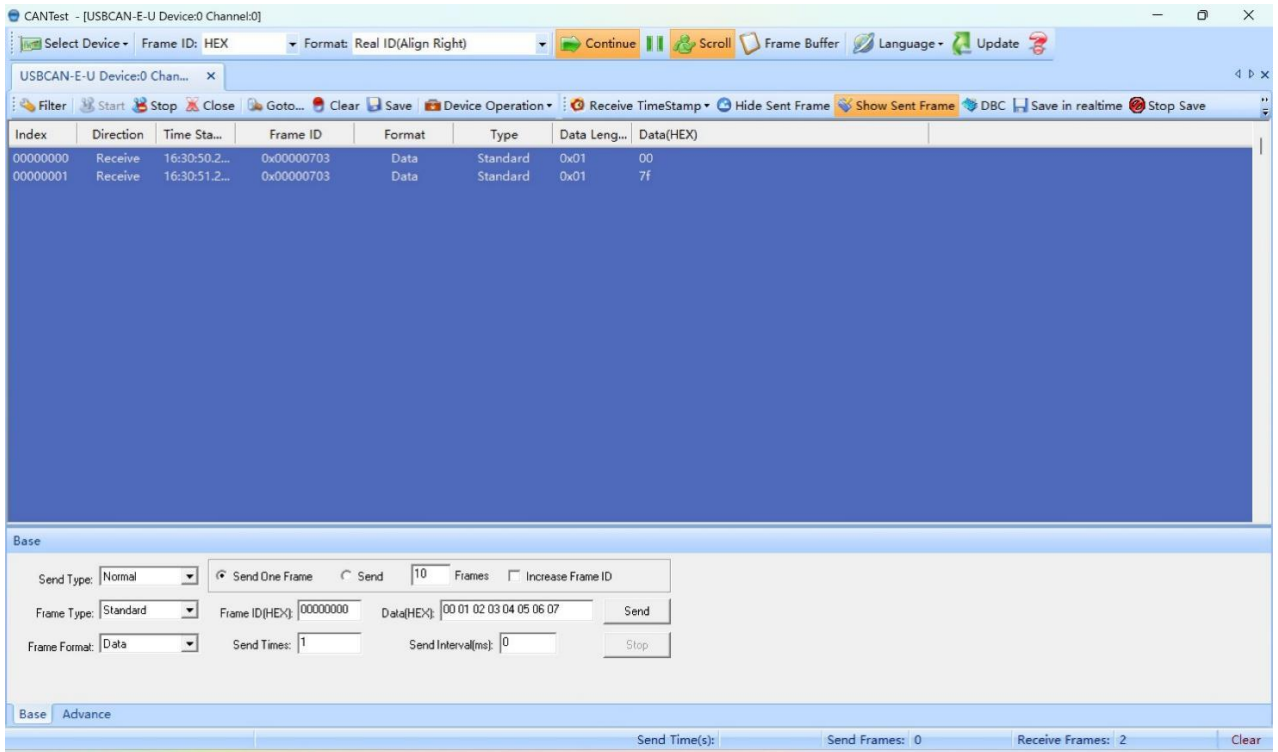
3 Communication Configuration

3.1 Communication configuration in CANTest software

- 1) Install the CANTest software.
- 2) Set the AC drive parameters Fd-00 = 25 (baud rate 500K), Fd-02 = 3 (node address), Fd-06 = 2 (CANopen protocol).
- 3) Open CANTest, click "Select Device", select "USBCAN-E-U", set the baud rate to 500K, and then click "Open and Start CAN".



- 4) When the AC drive is powered up, you can see that PD310CAN1 sends a Boot-UP message and then starts sending heartbeat messages.



The screenshot shows the CANTest software interface. The main window displays a table of received frames:

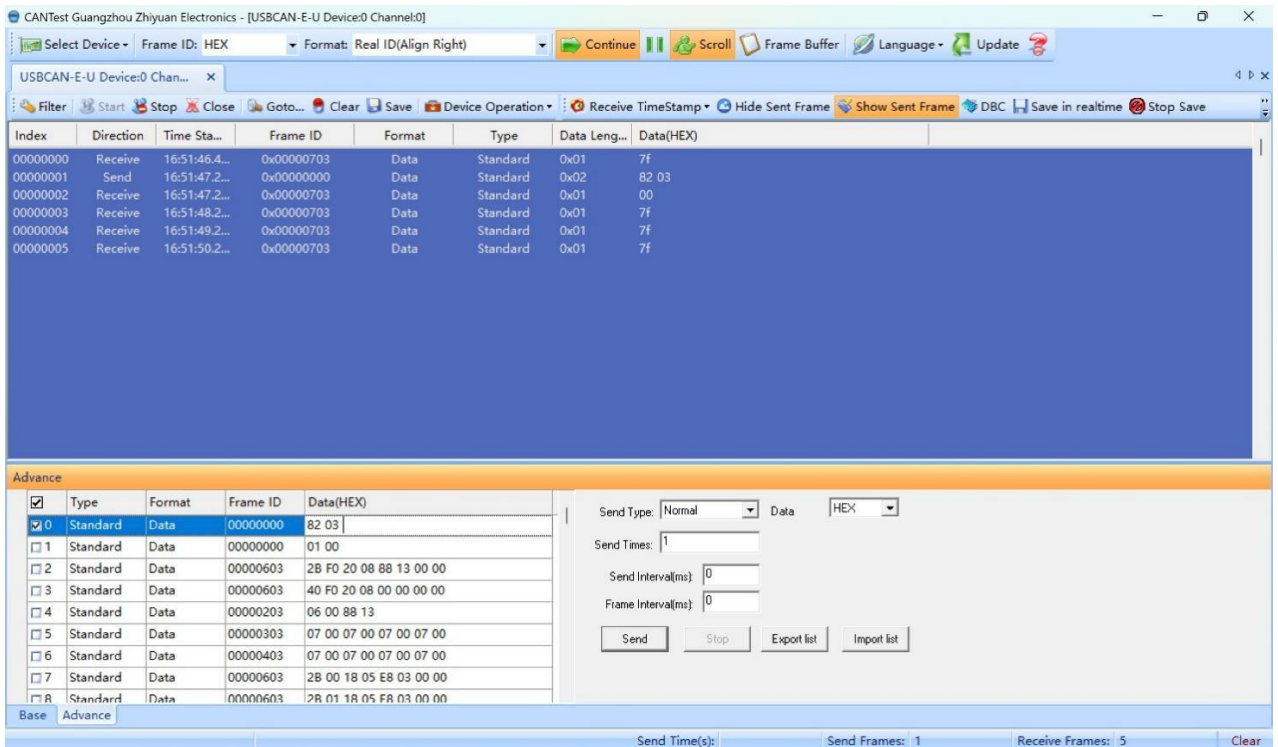
Index	Direction	Time Sta...	Frame ID	Format	Type	Data Leng...	Data(HEX)
00000000	Receive	16:30:50.2...	0x00000703	Data	Standard	0x01	00
00000001	Receive	16:30:51.2...	0x00000703	Data	Standard	0x01	7f

Below the table, the 'Base' configuration panel is visible, showing settings for sending frames:

- Send Type: Normal
- Send One Frame: Send 10 Frames Increase Frame ID
- Frame Type: Standard
- Frame ID(HEX): 00000000
- Data(HEX): 00 01 02 03 04 05 06 07
- Frame Format: Data
- Send Times: 1
- Send Interval(ms): 0

At the bottom, the status bar shows: Send Time(s): Send Frames: 0 Receive Frames: 2 Clear

- 5) Send NMT command through CANTest software with frame ID 0x00, first reset the node application layer, select standard frame for data format (all following frame types are standard frames), and then the data content is 0x8203 (82h denotes reset the node application layer command, 03h denotes the slave ID).



The screenshot shows the CANTest software interface with the 'Advance' configuration panel open. The main window displays a table of frames:

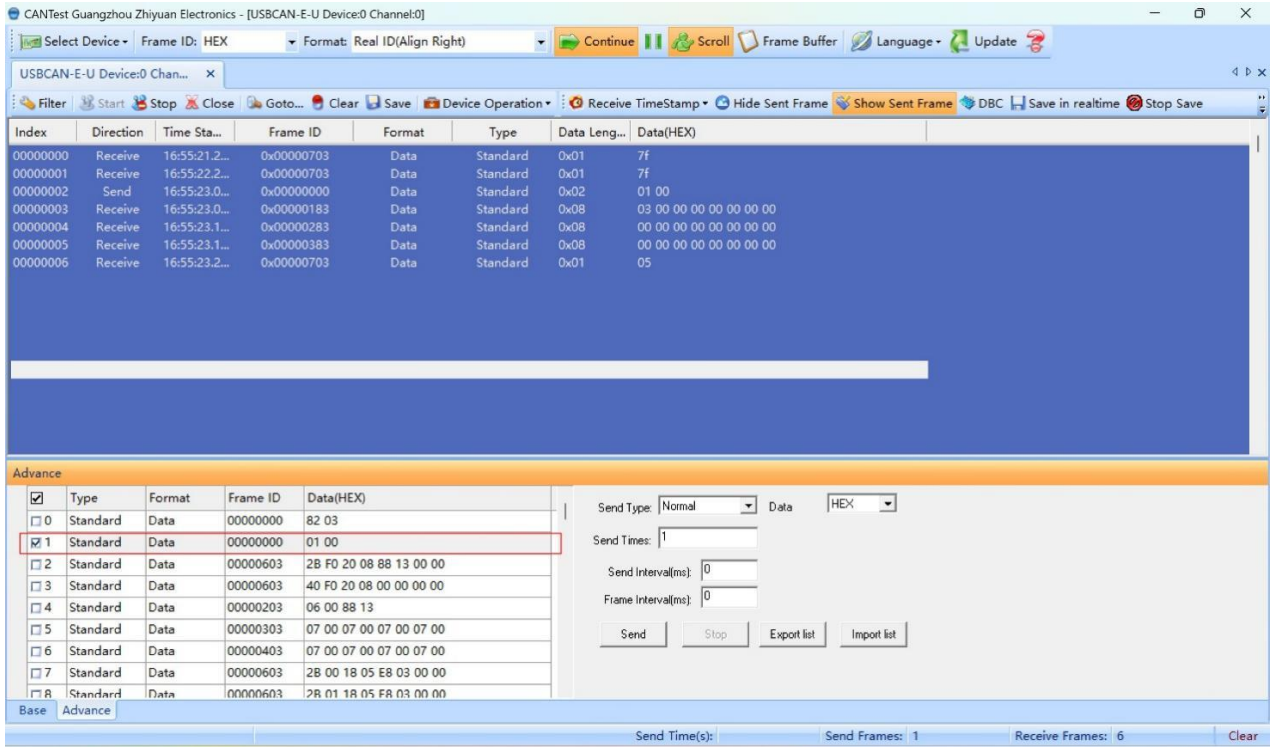
Index	Direction	Time Sta...	Frame ID	Format	Type	Data Leng...	Data(HEX)
00000000	Receive	16:51:46.4...	0x00000703	Data	Standard	0x01	7f
00000001	Send	16:51:47.2...	0x00000000	Data	Standard	0x02	82 03
00000002	Receive	16:51:47.2...	0x00000703	Data	Standard	0x01	00
00000003	Receive	16:51:48.2...	0x00000703	Data	Standard	0x01	7f
00000004	Receive	16:51:49.2...	0x00000703	Data	Standard	0x01	7f
00000005	Receive	16:51:50.2...	0x00000703	Data	Standard	0x01	7f

The 'Advance' configuration panel shows the following settings:

- Type: Standard
- Format: Data
- Frame ID: 00000000
- Data(HEX): 82 03
- Send Type: Normal
- Data: HEX
- Send Times: 1
- Send Interval(ms): 0
- Frame Interval(ms): 0

At the bottom, the status bar shows: Send Time(s): Send Frames: 1 Receive Frames: 5 Clear

- 6) Send NMT command through CANTest software to start the slave station (data content is 0x0103), you can see that the slave station starts to send TPDO message periodically.



The screenshot shows the CANTest software interface for a USBCAN-E-U Device. The main window displays a list of CAN bus frames with the following data:

Index	Direction	Time Sta...	Frame ID	Format	Type	Data Leng...	Data(HEX)
00000000	Receive	16:55:21.2...	0x00000703	Data	Standard	0x01	7f
00000001	Receive	16:55:22.2...	0x00000703	Data	Standard	0x01	7f
00000002	Send	16:55:23.0...	0x00000000	Data	Standard	0x02	01 00
00000003	Receive	16:55:23.0...	0x00000183	Data	Standard	0x08	03 00 00 00 00 00 00 00
00000004	Receive	16:55:23.1...	0x00000283	Data	Standard	0x08	00 00 00 00 00 00 00 00
00000005	Receive	16:55:23.1...	0x00000383	Data	Standard	0x08	00 00 00 00 00 00 00 00
00000006	Receive	16:55:23.2...	0x00000703	Data	Standard	0x01	05

The 'Advance' section shows a configuration table for sending frames:

Type	Format	Frame ID	Data(HEX)
<input type="checkbox"/> 0	Standard	Data	00000000 82 03
<input checked="" type="checkbox"/> 1	Standard	Data	00000000 01 00
<input type="checkbox"/> 2	Standard	Data	00000603 2B F0 20 08 88 13 00 00
<input type="checkbox"/> 3	Standard	Data	00000603 40 F0 20 08 00 00 00 00
<input type="checkbox"/> 4	Standard	Data	00000203 06 00 88 13
<input type="checkbox"/> 5	Standard	Data	00000303 07 00 07 00 07 00 07 00
<input type="checkbox"/> 6	Standard	Data	00000403 07 00 07 00 07 00 07 00
<input type="checkbox"/> 7	Standard	Data	00000603 2B 00 18 05 E8 03 00 00
<input type="checkbox"/> 8	Standard	Data	00000603 2B 01 18 05 E8 03 00 00

Configuration parameters for the selected frame (index 1):

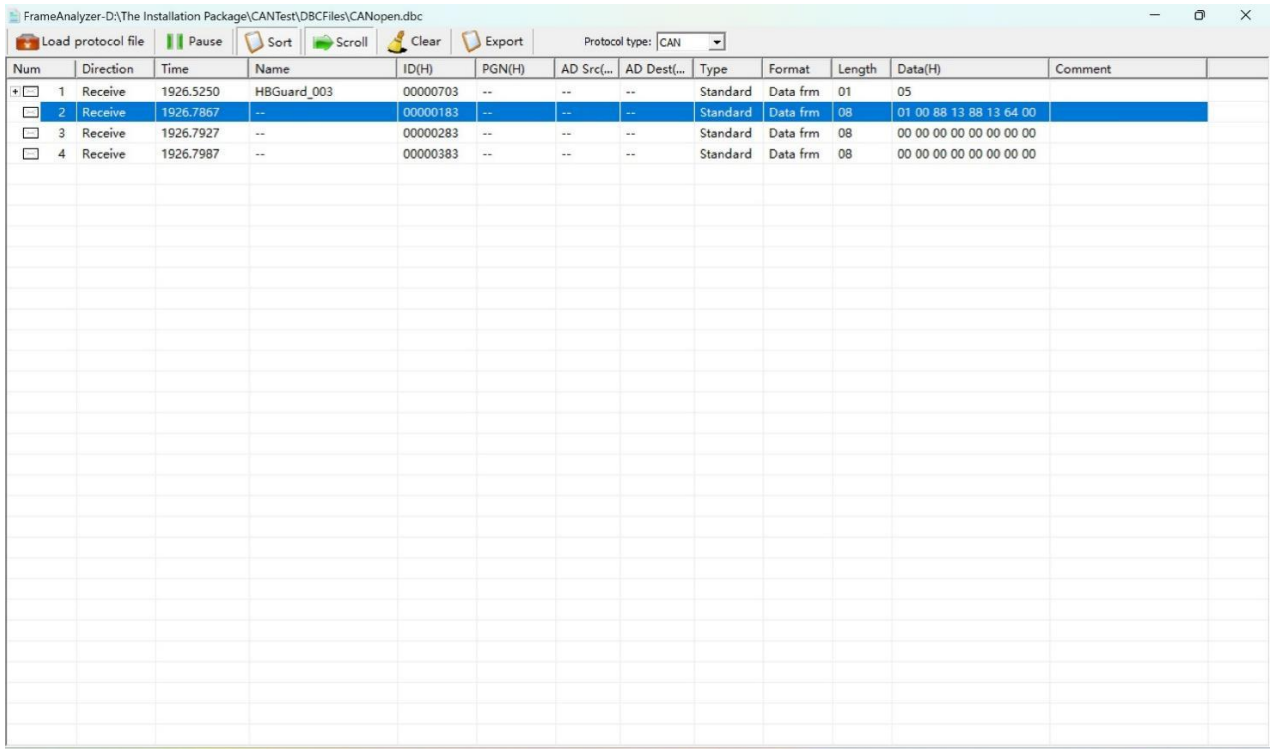
- Send Type: Normal
- Data: HEX
- Send Times: 1
- Send Interval(ms): 0
- Frame Interval(ms): 0

Buttons: Send, Stop, Export list, Import list

Status bar: Send Time(s): Send Frames: 1 Receive Frames: 6 Clear

7) Read the AC drive parameters through TPDO, from Chapter 2, we can see that the first two parameters of TPDO1 have been fixed as the current state of the AC drive and the frequency of AC drive operation, and the last two parameters need to be set by the user through Fd-20 and Fd-21 respectively.

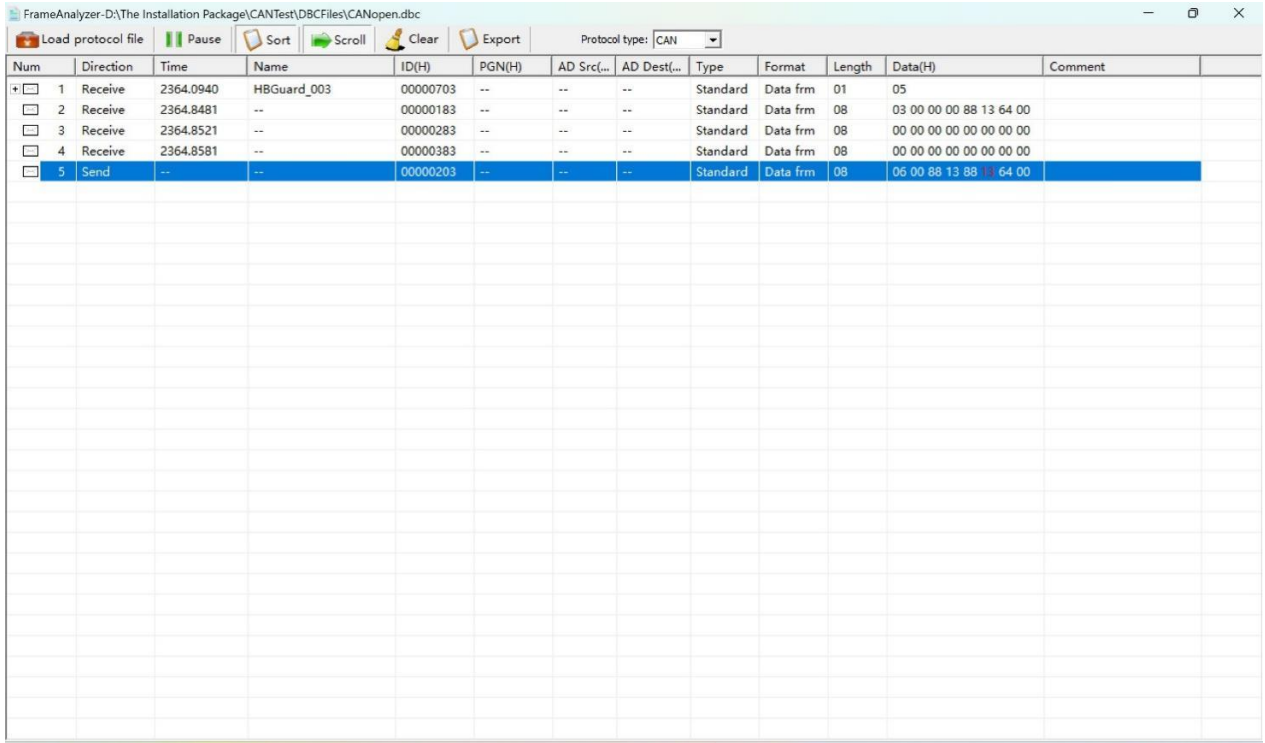
Example: Set Fd-20 = 0x0007 (preset frequency of AC drive). Fd-21 = 0x0010 (acceleration time of AC drive1), then observe the content returned by TPDO1. The first word 0x0001 indicates that the AC drive is in positive running, the second word 0x1388 indicates that the running frequency is 50.00Hz, the third word 0x1388 indicates that the AC drive parameter F0-07 is read as 50.00Hz, the fourth word 0x0064 indicates that the AC drive parameter F0-16 is 10.0s.



Num	Direction	Time	Name	ID(H)	PGN(H)	AD Src(...)	AD Dest(...)	Type	Format	Length	Data(H)	Comment
1	Receive	1926.5250	HGuard_003	00000703	--	--	--	Standard	Data frm	01	05	
2	Receive	1926.7867	--	00000183	--	--	--	Standard	Data frm	08	01 00 88 13 88 13 64 00	
3	Receive	1926.7927	--	00000283	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
4	Receive	1926.7987	--	00000383	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	

- 8) Setting AC drive parameters through RPDO, from Chapter 2, it is known that the first two parameters of RPDO1 have been fixed as AC drive command and AC drive target frequency, and the last two parameters need to be set by the user through Fd-10 and Fd-11 respectively.

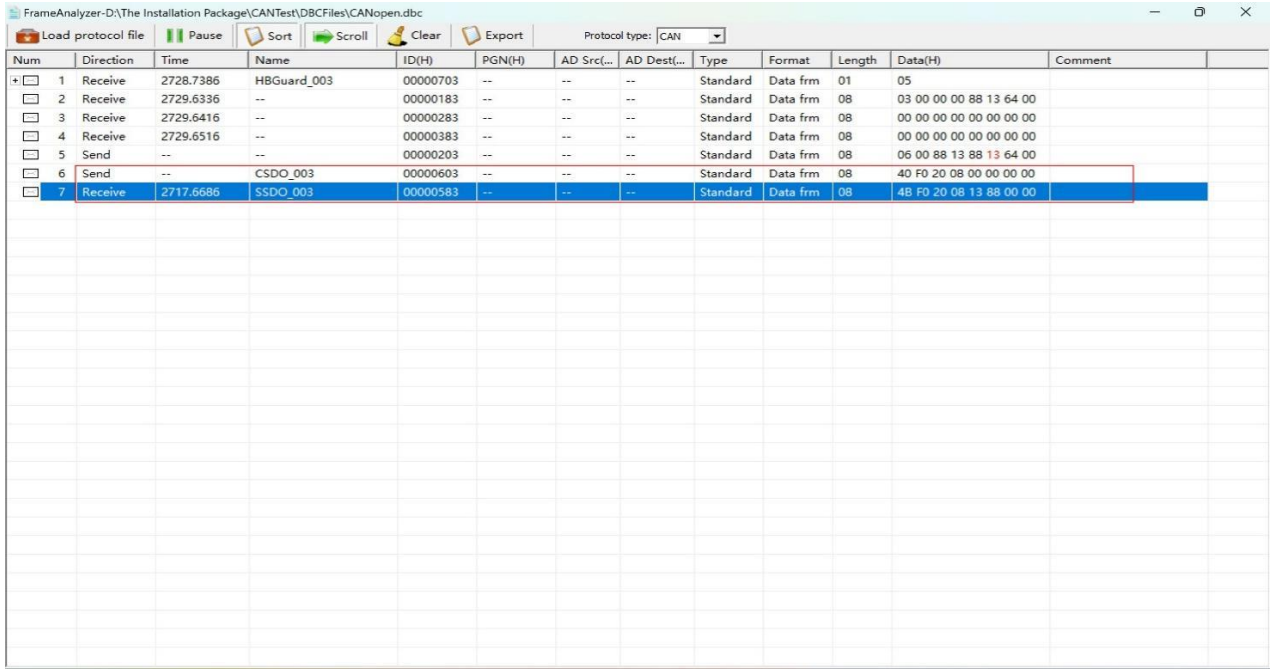
Example: Set Fd-10 = 0x0007 (preset frequency of AC drive). Fd-11 = 0x0010 (acceleration time of AC drive 1), you can observe whether the parameters are modified successfully or not through the previous step TPDO.



Num	Direction	Time	Name	ID(H)	PGN(H)	AD Src(...)	AD Dest(...)	Type	Format	Length	Data(H)	Comment
1	Receive	2364.0940	HBGuard_003	00000703	--	--	--	Standard	Data frm	01	05	
2	Receive	2364.8481	--	00000183	--	--	--	Standard	Data frm	08	03 00 00 00 88 13 64 00	
3	Receive	2364.8521	--	00000283	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
4	Receive	2364.8581	--	00000383	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
5	Send	--	--	00000203	--	--	--	Standard	Data frm	08	06 00 88 13 88 64 00	

9) Read AC drive parameter F0-07 (preset frequency) via SDO

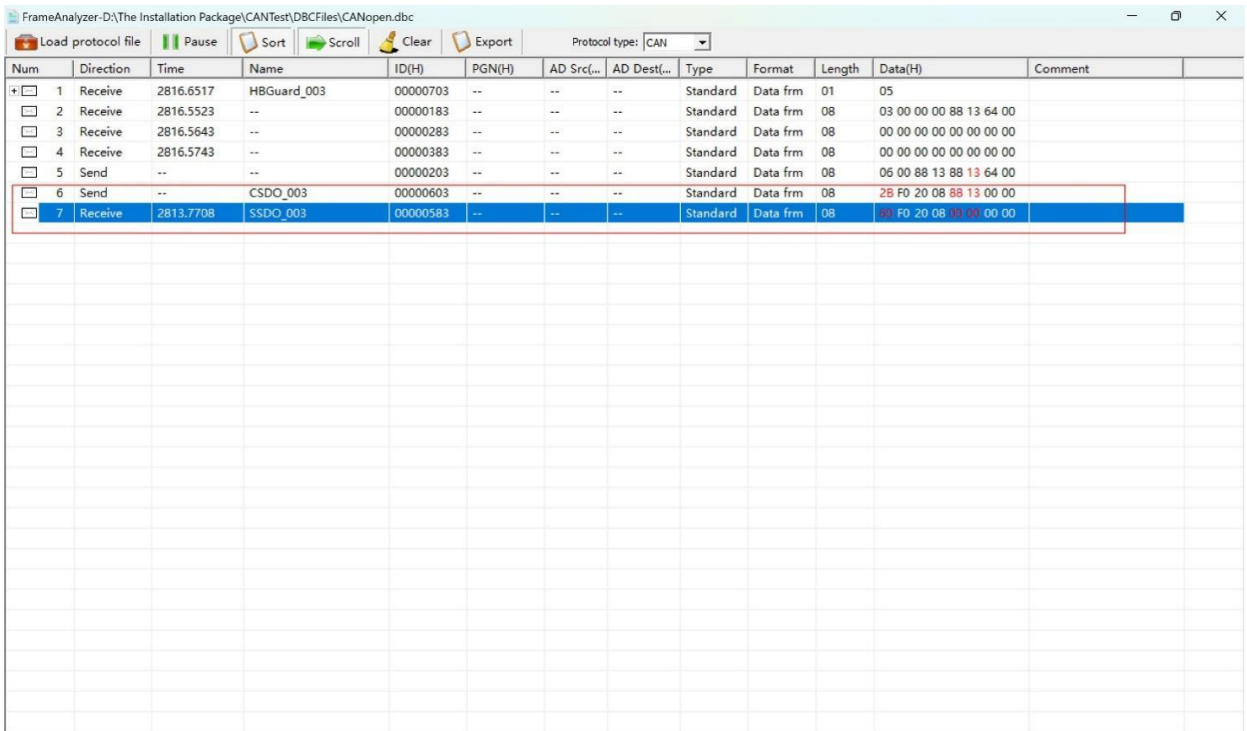
Frame ID is 0x600+0x03 (Fd-02 set address), data content is 40 F0 20 08 00 00 00 00 00, note that due to the concept of index 0 in the CANopen object dictionary, reading the AC drive through SDO needs to add 1 more to the current address. in this example, F0-07, the actual address that needs to be written is F0 20 08. Reading the return frame (ID0x500+0x03) through SDO, you can get the value of AC drive F0-07.



Num	Direction	Time	Name	ID(H)	PGN(H)	AD Src(...)	AD Dest(...)	Type	Format	Length	Data(H)	Comment
1	Receive	2728.7386	HGuard_003	00000703	--	--	--	Standard	Data frm	01	05	
2	Receive	2729.6336	--	00000183	--	--	--	Standard	Data frm	08	03 00 00 00 88 13 64 00	
3	Receive	2729.6416	--	00000283	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
4	Receive	2729.6516	--	00000383	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
5	Send	--	--	00000203	--	--	--	Standard	Data frm	08	06 00 88 13 88 13 64 00	
6	Send	--	CSDO_003	00000603	--	--	--	Standard	Data frm	08	40 F0 20 08 00 00 00 00	
7	Receive	2717.6686	SSDO_003	00000583	--	--	--	Standard	Data frm	08	4B F0 20 08 13 88 00 00	

10) Modification of AC drive parameters F0-07 (preset frequency) via SDO

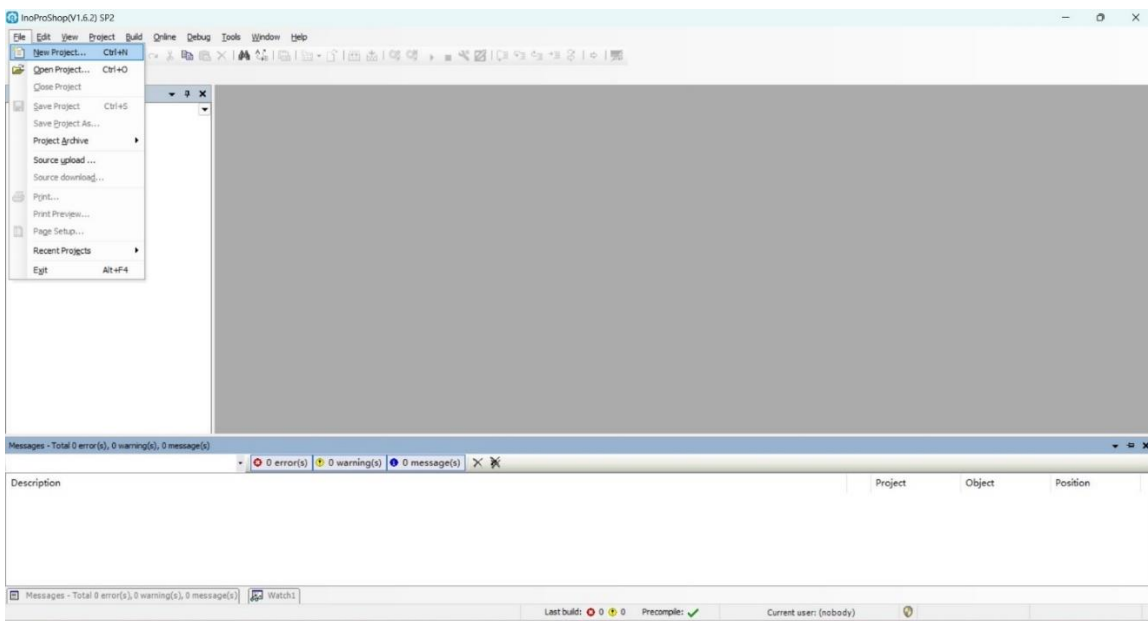
Frame ID is 0x600+0x03 (Fd-02 set address), data content is 2B F0 20 08 88 13 00 00, note that due to the existence of the concept of index 0 in the CANopen object dictionary, so read the AC drive through the SDO need to be in the current address and then added to the 1. In this example, the F0-07, the actual need to write the address of the F0 20 08. Read the return through the SDO frame (ID0x500+0x03), you can see whether the modification is successful.



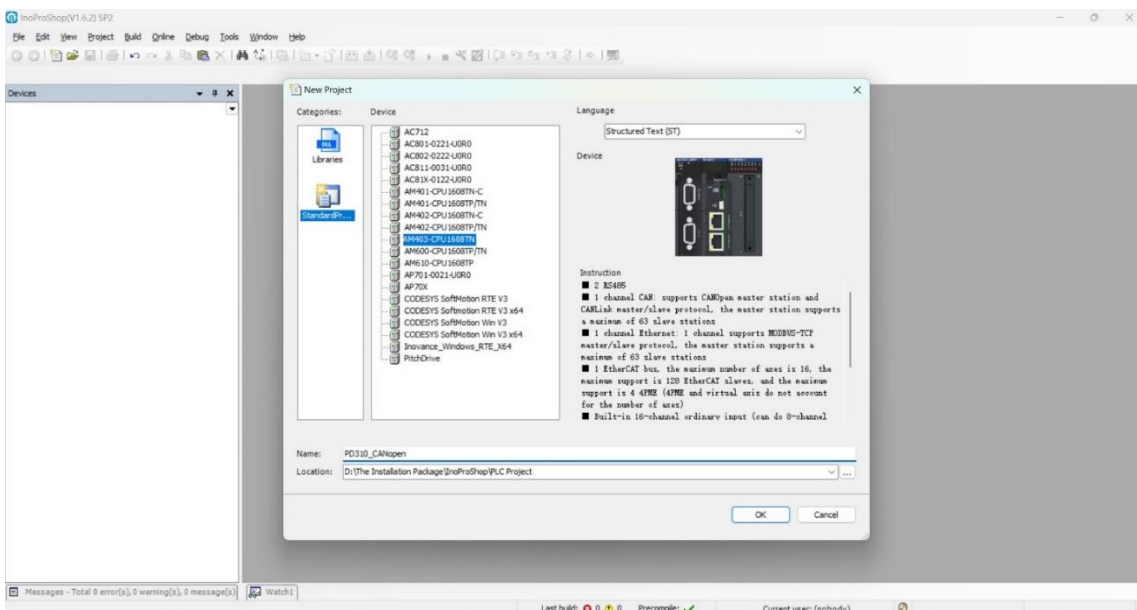
Num	Direction	Time	Name	ID(H)	PGN(H)	AD Src(...)	AD Dest(...)	Type	Format	Length	Data(H)	Comment
1	Receive	2816.6517	HBGuard_003	00000703	--	--	--	Standard	Data frm	01	05	
2	Receive	2816.5523	--	00000183	--	--	--	Standard	Data frm	08	03 00 00 00 00 88 13 64 00	
3	Receive	2816.5643	--	00000283	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
4	Receive	2816.5743	--	00000383	--	--	--	Standard	Data frm	08	00 00 00 00 00 00 00 00	
5	Send	--	--	00000203	--	--	--	Standard	Data frm	08	05 00 88 13 88 13 64 00	
6	Send	--	CSDO_003	00000603	--	--	--	Standard	Data frm	08	2B F0 20 08 88 13 00 00	
7	Receive	2813.7708	SDDO_003	00000583	--	--	--	Standard	Data frm	08	2B F0 20 08 88 13 00 00	

3.2 Device Configuration in CODESYS

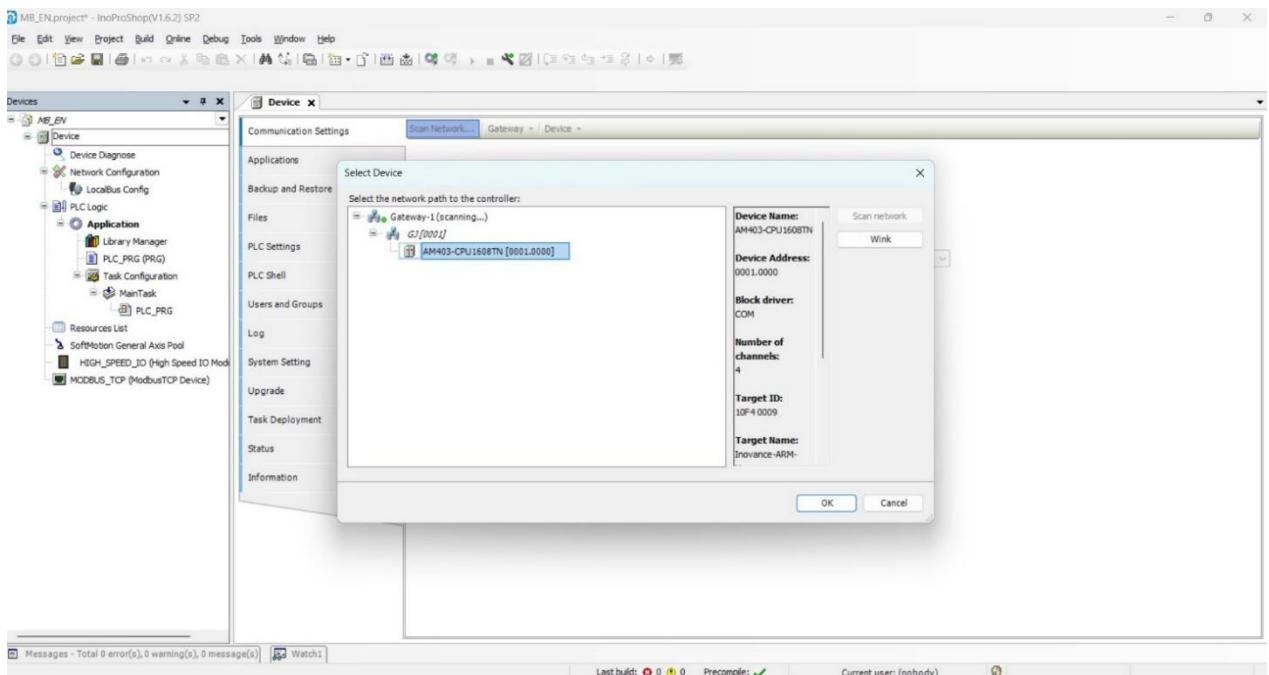
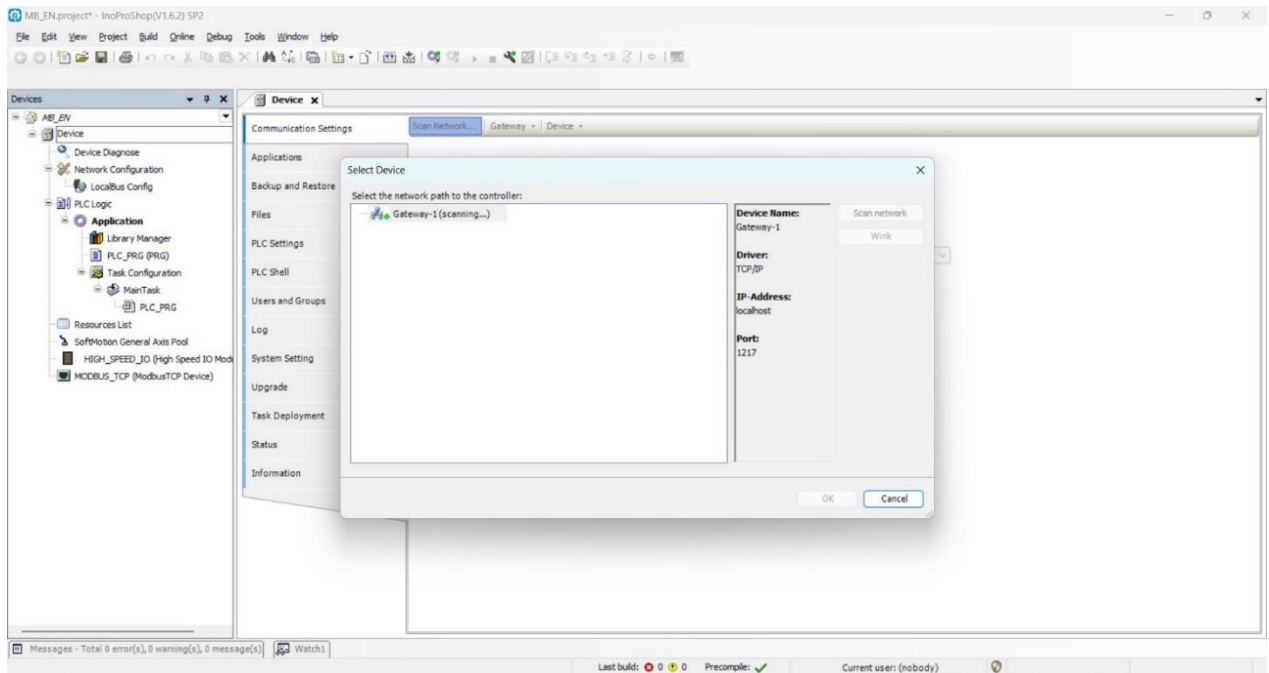
- 1) Install the appropriate CODESYS software, since the PLC used in this example is the AM403, the corresponding software is installed.
- 2) Connect the PLC and the PD310CAN1 card. Use twisted shielded cable to connect the CANopen master and the PD310CAN1 card.
- 3) Set the corresponding AC drive and expansion card parameters after power-on, mainly communication protocol Fd-06 = 02, etc. For detailed description, refer to Table 2-1 Communication Card Configuration Parameters.
- 4) Open CODESYS software, choose to create a new project.



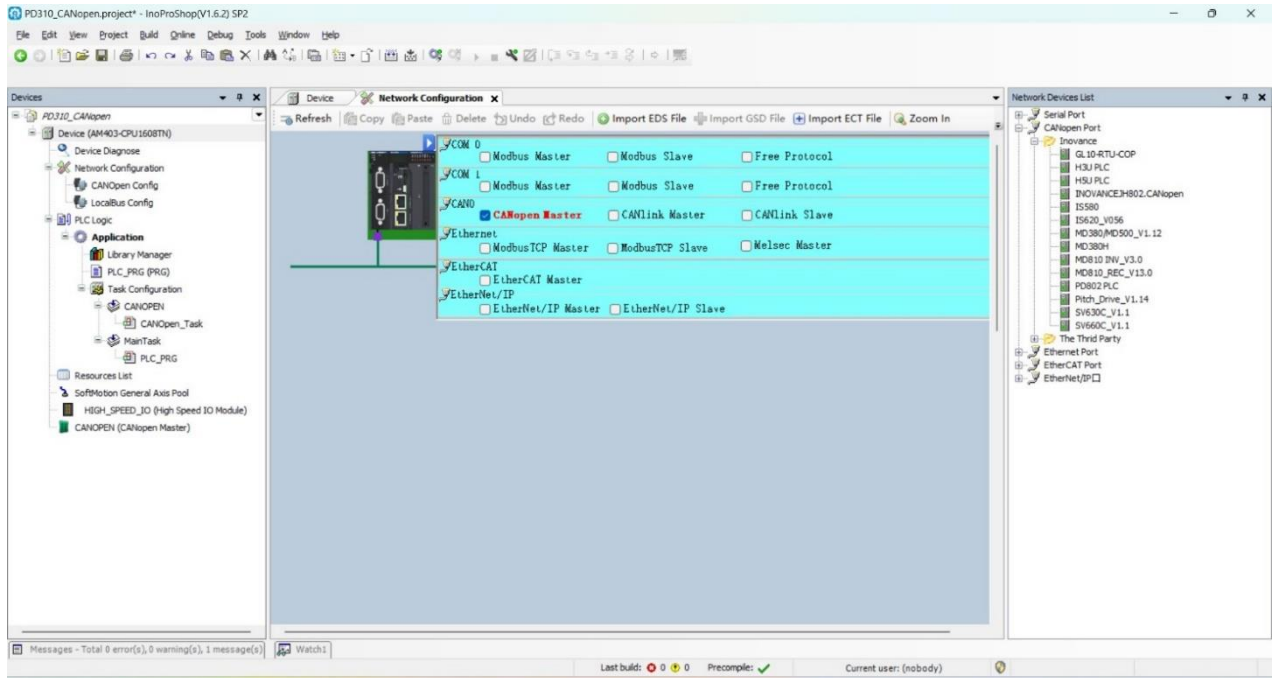
- 5) Select the corresponding PLC model, enter the project name and save path (no Chinese path), and click Create.



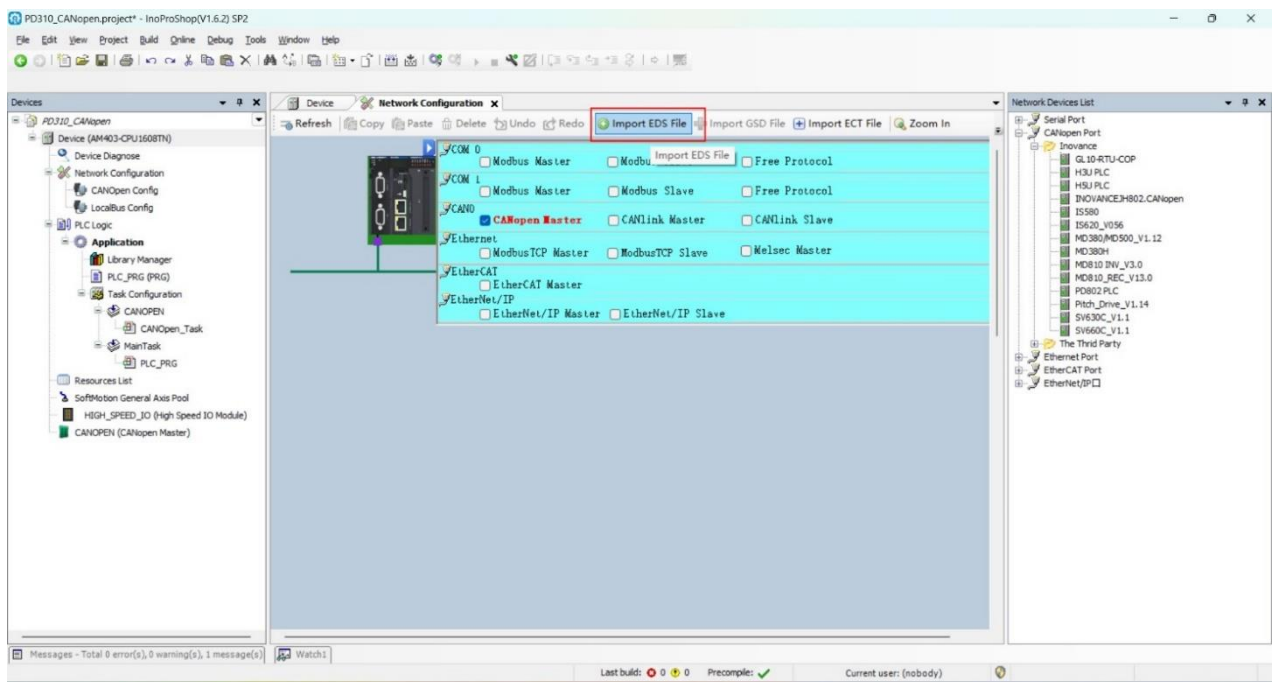
6) After creating the project, click Device to scan for PLC.



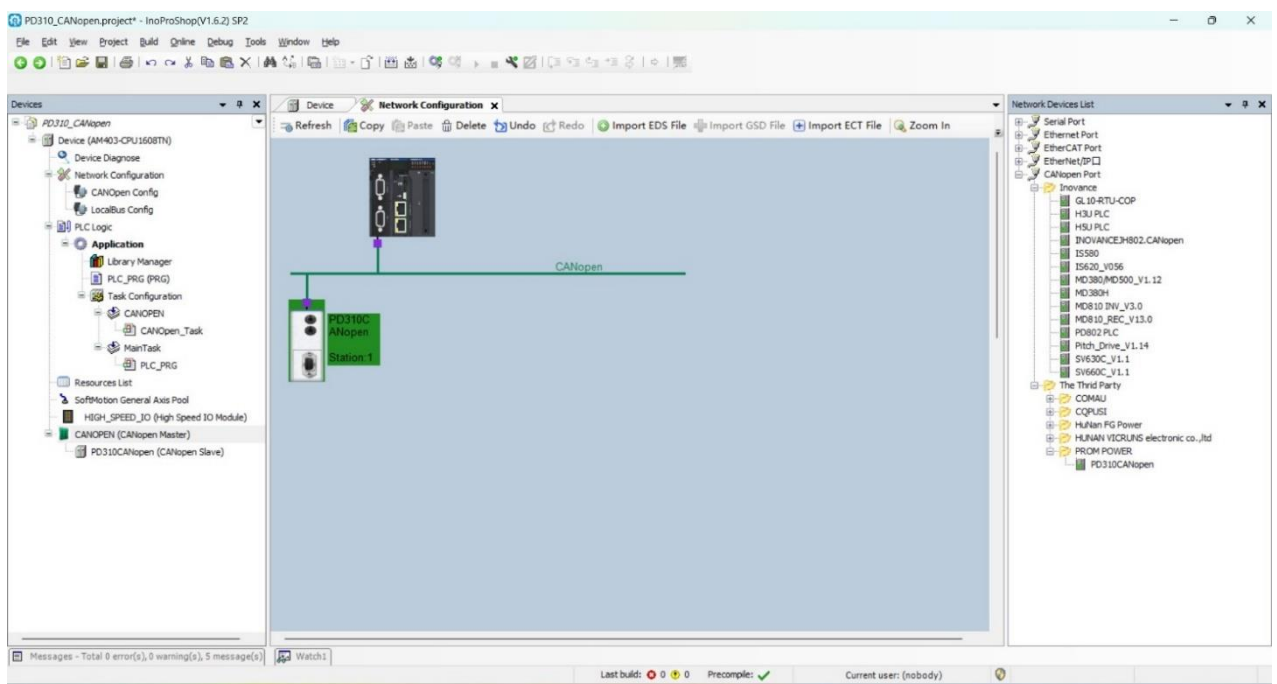
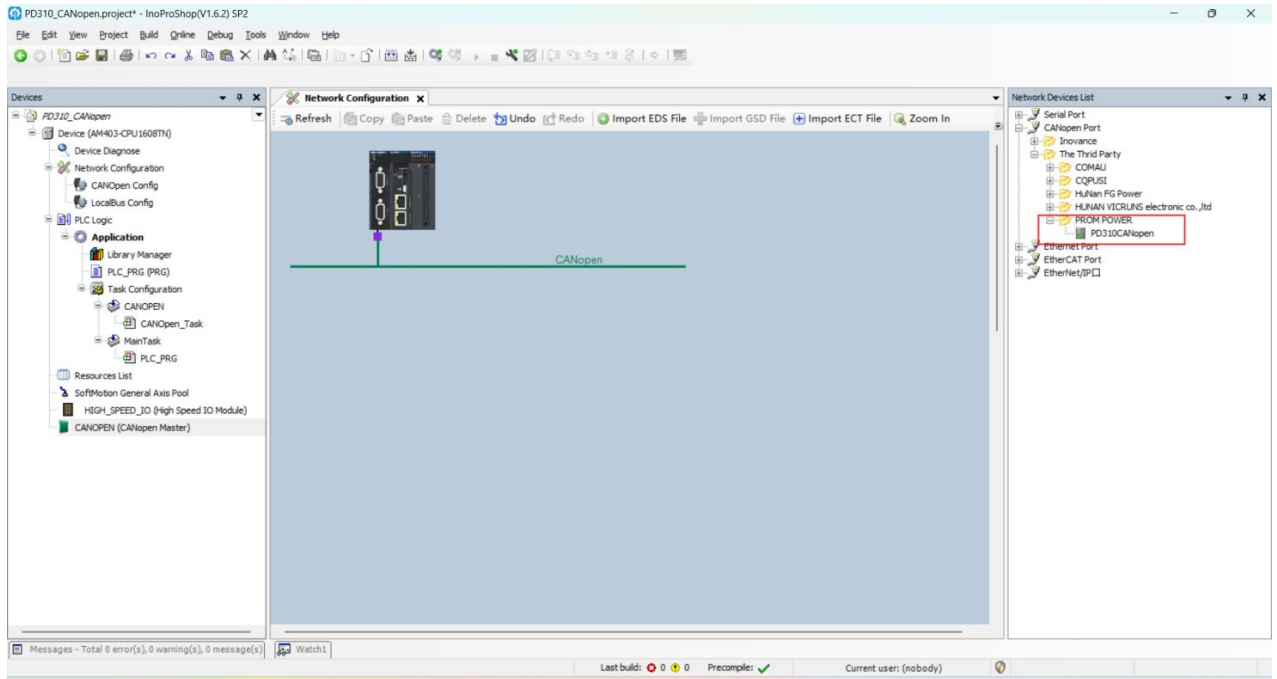
7) After connecting the PLC, click Network Configuration and select CANopen Master.



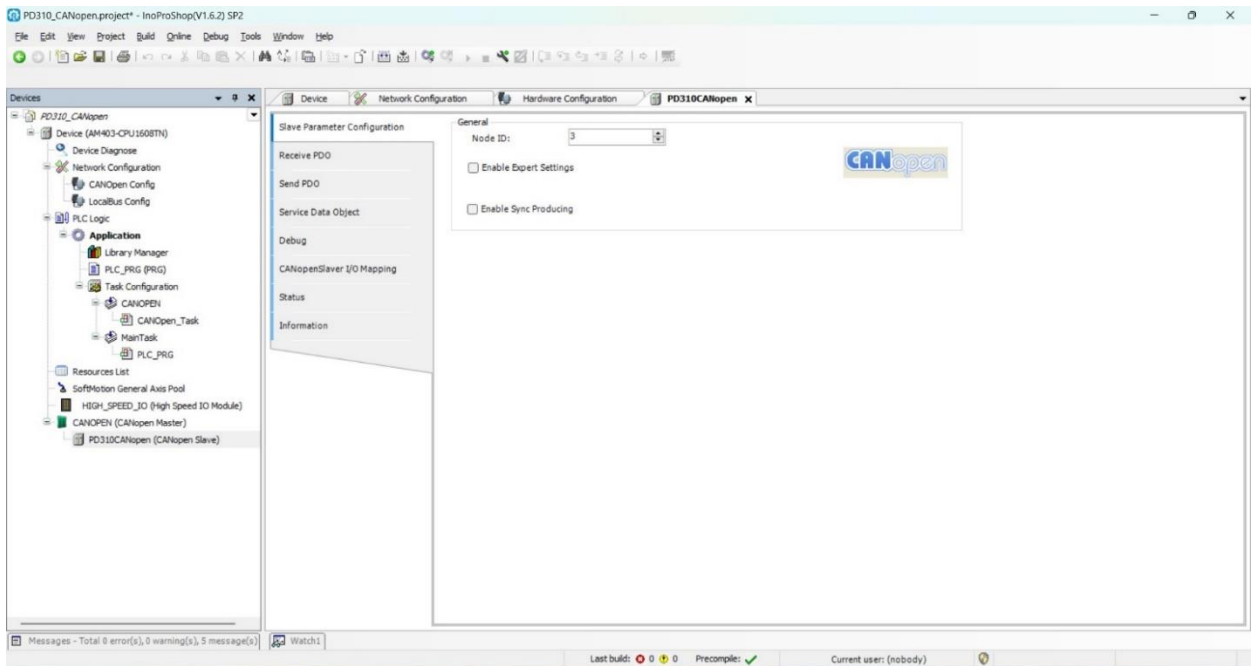
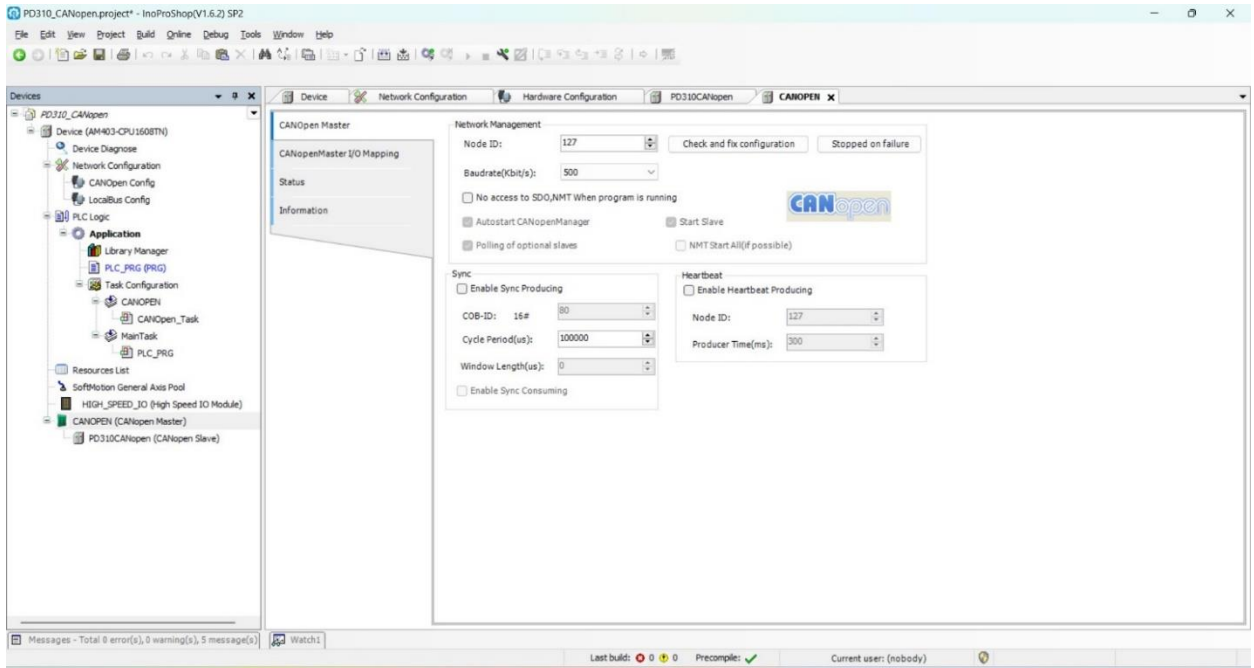
8) Install the PD310CANopen_V1.0.eds file.



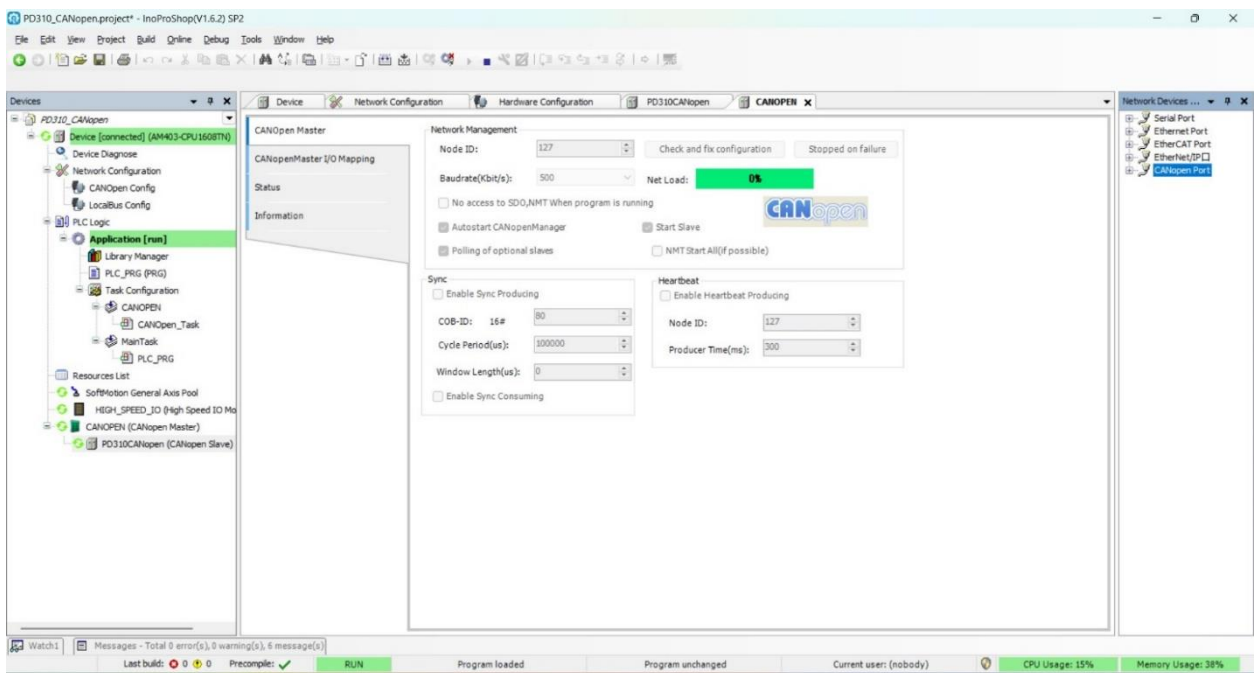
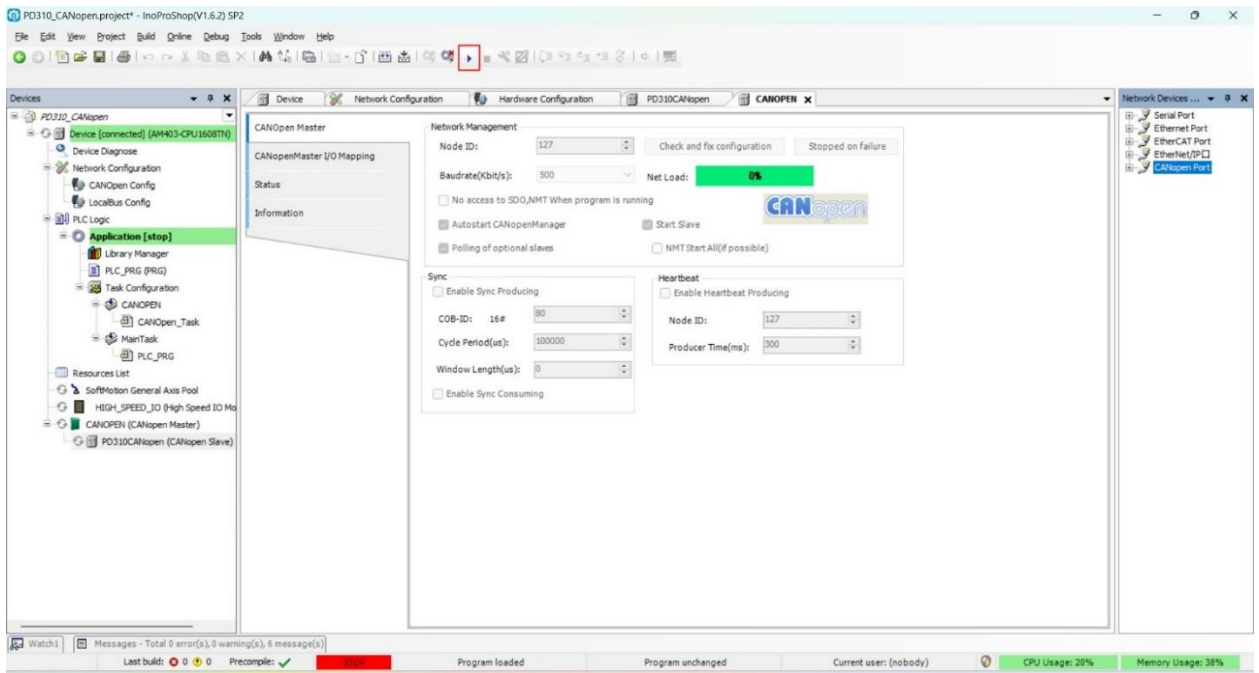
- After installing the corresponding device description file, find the PD310CANopen device in the right device column and drag it in to make a network connection.



- 10) Set the device information of master and slave respectively mainly baud rate as well as node address (slave node address needs to match the address set by Fd-02).



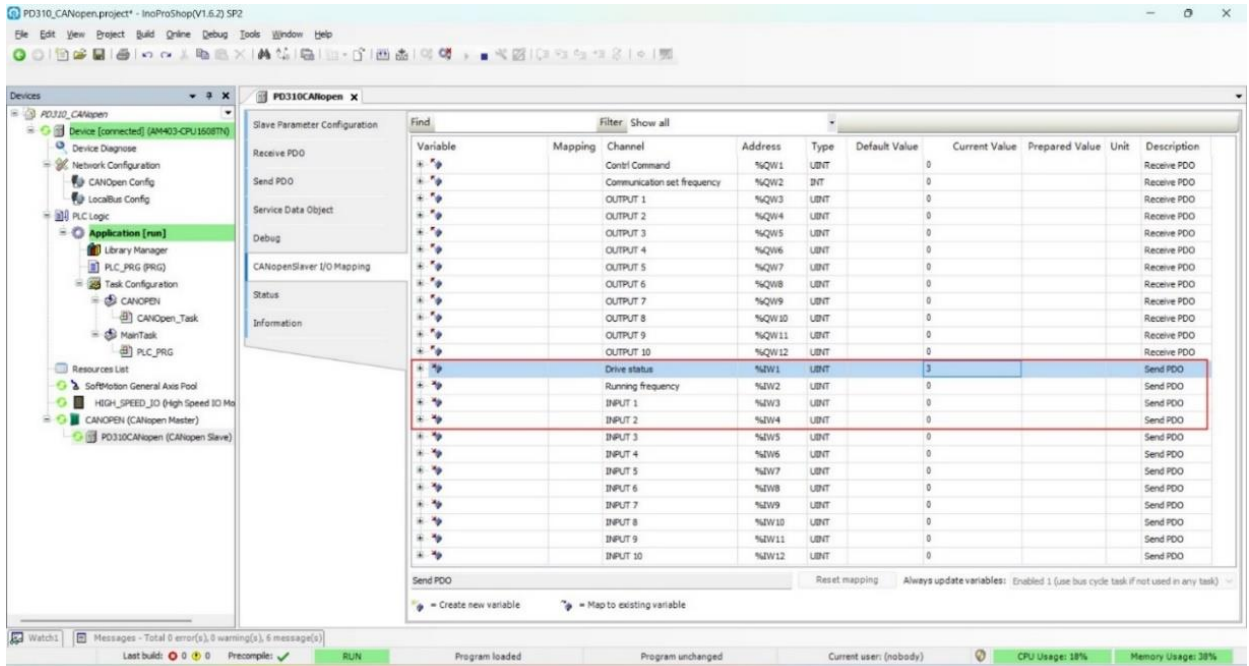
11) Then click on Compile and then download, then click Run and the device is connected successfully as shown in the picture below:



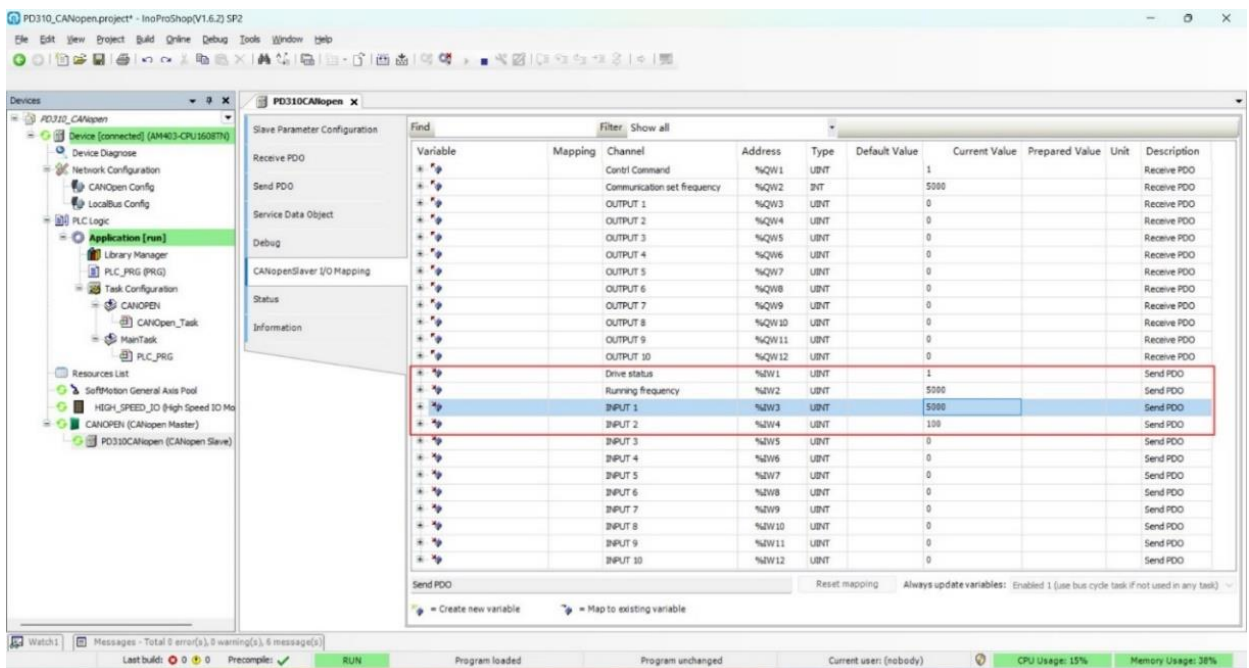
12) When the device is successfully connected, the CANopen master will automatically reset the slave application layer and start the slave at the same time, and the slave will send TPDO periodically, as shown in the figure below, the place marked in red is TPDO1, and it can be seen in the second chapter that the first two parameters of TPDO1 have already been fixed as the current state of the AC drive and the frequency of the AC drive operation, and the last two parameters need to be set by the user through Fd-20 and Fd-21 respectively.

Example:

Set Fd-20 = 0x0007 (preset frequency of AC drive). Fd-21 = 0x0010 (acceleration time of AC drive 1), then observe the content returned by TPDO1. The first parameter indicates that the AC drive is in positive running, the second parameter indicates that the running frequency is 50.00H, the third parameter indicates that the AC drive parameter F0-07 is read as 50.00Hz, and the fourth parameter indicates that the AC drive parameter F0-16 is 10.0s.



Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared Value	Unit	Description
+		Control Command	%QW1	UINT	0				Receive PDO
+		Communication set frequency	%QW2	INT	0				Receive PDO
+		OUTPUT 1	%QW3	UINT	0				Receive PDO
+		OUTPUT 2	%QW4	UINT	0				Receive PDO
+		OUTPUT 3	%QW5	UINT	0				Receive PDO
+		OUTPUT 4	%QW6	UINT	0				Receive PDO
+		OUTPUT 5	%QW7	UINT	0				Receive PDO
+		OUTPUT 6	%QW8	UINT	0				Receive PDO
+		OUTPUT 7	%QW9	UINT	0				Receive PDO
+		OUTPUT 8	%QW10	UINT	0				Receive PDO
+		OUTPUT 9	%QW11	UINT	0				Receive PDO
+		OUTPUT 10	%QW12	UINT	0				Receive PDO
+		Drive status	%IW1	UINT	0	3			Send PDO
+		Running frequency	%IW2	UINT	0				Send PDO
+		INPUT 1	%IW3	UINT	0				Send PDO
+		INPUT 2	%IW4	UINT	0				Send PDO
+		INPUT 3	%IW5	UINT	0				Send PDO
+		INPUT 4	%IW6	UINT	0				Send PDO
+		INPUT 5	%IW7	UINT	0				Send PDO
+		INPUT 6	%IW8	UINT	0				Send PDO
+		INPUT 7	%IW9	UINT	0				Send PDO
+		INPUT 8	%IW10	UINT	0				Send PDO
+		INPUT 9	%IW11	UINT	0				Send PDO
+		INPUT 10	%IW12	UINT	0				Send PDO

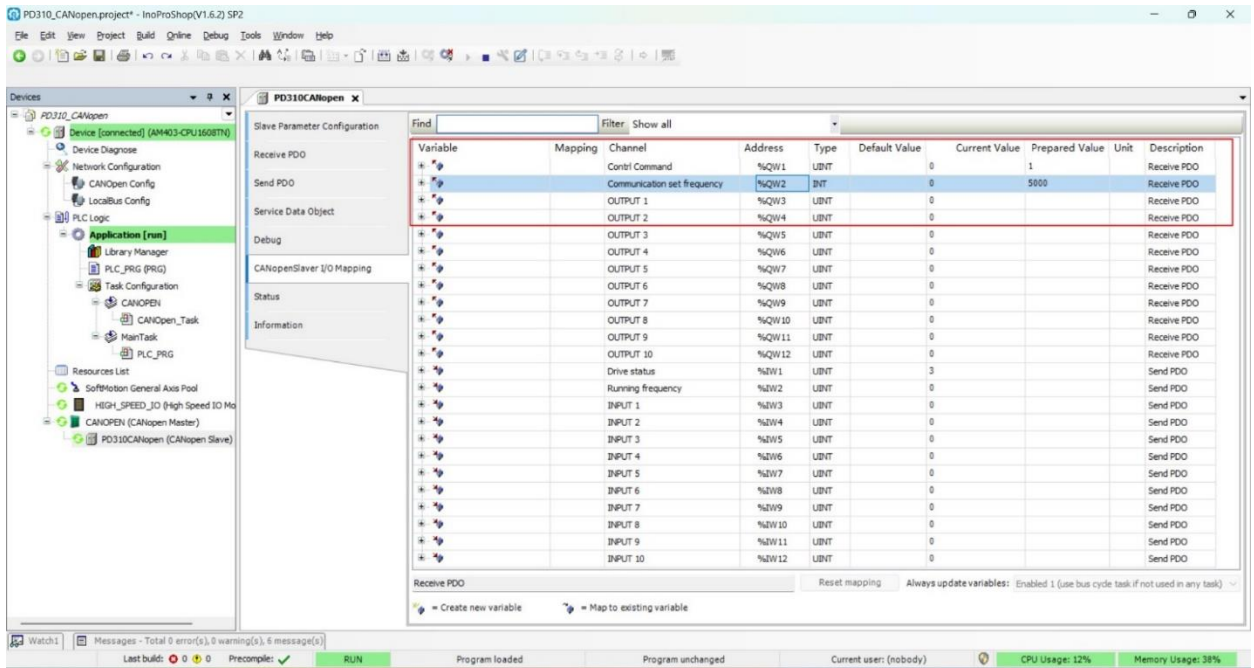


Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared Value	Unit	Description
+		Control Command	%QW1	UINT	0				Receive PDO
+		Communication set frequency	%QW2	INT	5000				Receive PDO
+		OUTPUT 1	%QW3	UINT	0				Receive PDO
+		OUTPUT 2	%QW4	UINT	0				Receive PDO
+		OUTPUT 3	%QW5	UINT	0				Receive PDO
+		OUTPUT 4	%QW6	UINT	0				Receive PDO
+		OUTPUT 5	%QW7	UINT	0				Receive PDO
+		OUTPUT 6	%QW8	UINT	0				Receive PDO
+		OUTPUT 7	%QW9	UINT	0				Receive PDO
+		OUTPUT 8	%QW10	UINT	0				Receive PDO
+		OUTPUT 9	%QW11	UINT	0				Receive PDO
+		OUTPUT 10	%QW12	UINT	0				Receive PDO
+		Drive status	%IW1	UINT	0	1			Send PDO
+		Running frequency	%IW2	UINT	0	5000			Send PDO
+		INPUT 1	%IW3	UINT	0	5000			Send PDO
+		INPUT 2	%IW4	UINT	0	100			Send PDO
+		INPUT 3	%IW5	UINT	0				Send PDO
+		INPUT 4	%IW6	UINT	0				Send PDO
+		INPUT 5	%IW7	UINT	0				Send PDO
+		INPUT 6	%IW8	UINT	0				Send PDO
+		INPUT 7	%IW9	UINT	0				Send PDO
+		INPUT 8	%IW10	UINT	0				Send PDO
+		INPUT 9	%IW11	UINT	0				Send PDO
+		INPUT 10	%IW12	UINT	0				Send PDO

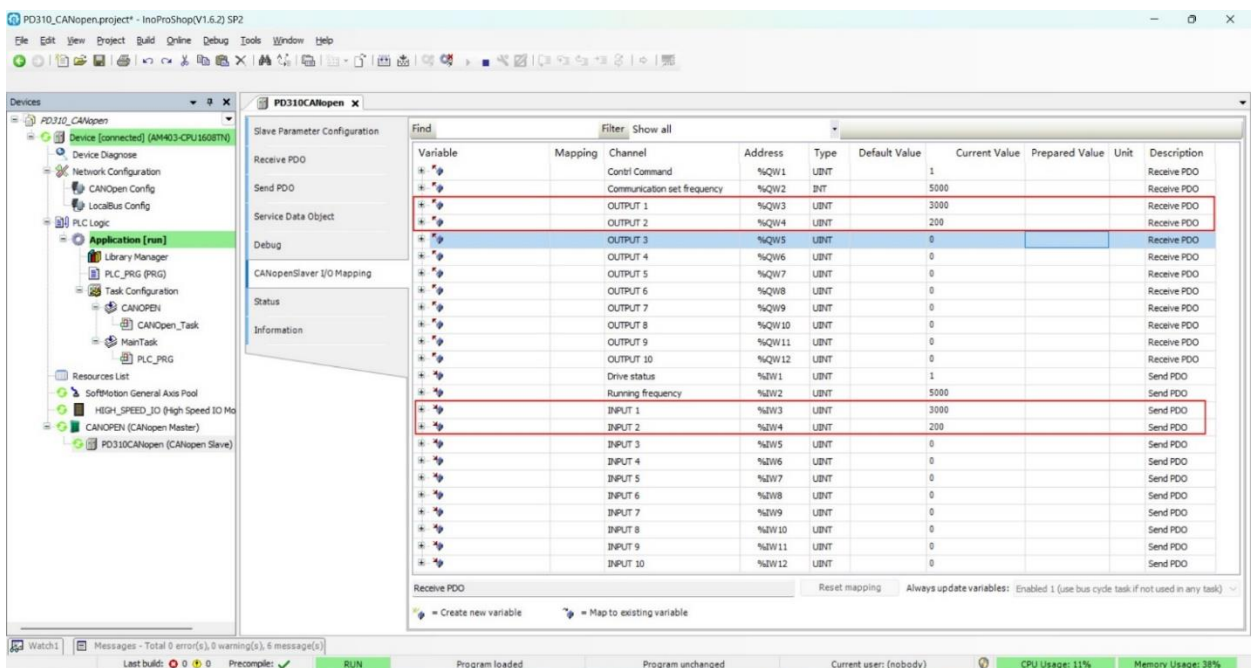
13) Modify the AC drive parameters through RPDO, it is known from Chapter 2 that the first two parameters of RPDO1 have been fixed as AC drive command and AC drive target frequency, and the last two parameters need to be set by the user through Fd-10 and Fd-11 respectively.

Example:

Set Fd-10 = 0x0007 (preset frequency of AC drive). Fd-11 = 0x0010 (acceleration time of AC drive 1), you can observe whether the parameters are modified successfully or not through the previous step TPDO.



Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared Value	Unit	Description
Control Command			%QW1	UINT	0	1			Receive PDO
Communication set frequency			%QW2	INT	0	5000			Receive PDO
OUTPUT 1			%QW3	UINT	0				Receive PDO
OUTPUT 2			%QW4	UINT	0				Receive PDO
OUTPUT 3			%QW5	UINT	0				Receive PDO
OUTPUT 4			%QW6	UINT	0				Receive PDO
OUTPUT 5			%QW7	UINT	0				Receive PDO
OUTPUT 6			%QW8	UINT	0				Receive PDO
OUTPUT 7			%QW9	UINT	0				Receive PDO
OUTPUT 8			%QW10	UINT	0				Receive PDO
OUTPUT 9			%QW11	UINT	0				Receive PDO
OUTPUT 10			%QW12	UINT	0				Receive PDO
Drive status			%W1	UINT	3				Send PDO
Running frequency			%W2	UINT	0				Send PDO
INPUT 1			%W3	UINT	0				Send PDO
INPUT 2			%W4	UINT	0				Send PDO
INPUT 3			%W5	UINT	0				Send PDO
INPUT 4			%W6	UINT	0				Send PDO
INPUT 5			%W7	UINT	0				Send PDO
INPUT 6			%W8	UINT	0				Send PDO
INPUT 7			%W9	UINT	0				Send PDO
INPUT 8			%W10	UINT	0				Send PDO
INPUT 9			%W11	UINT	0				Send PDO
INPUT 10			%W12	UINT	0				Send PDO

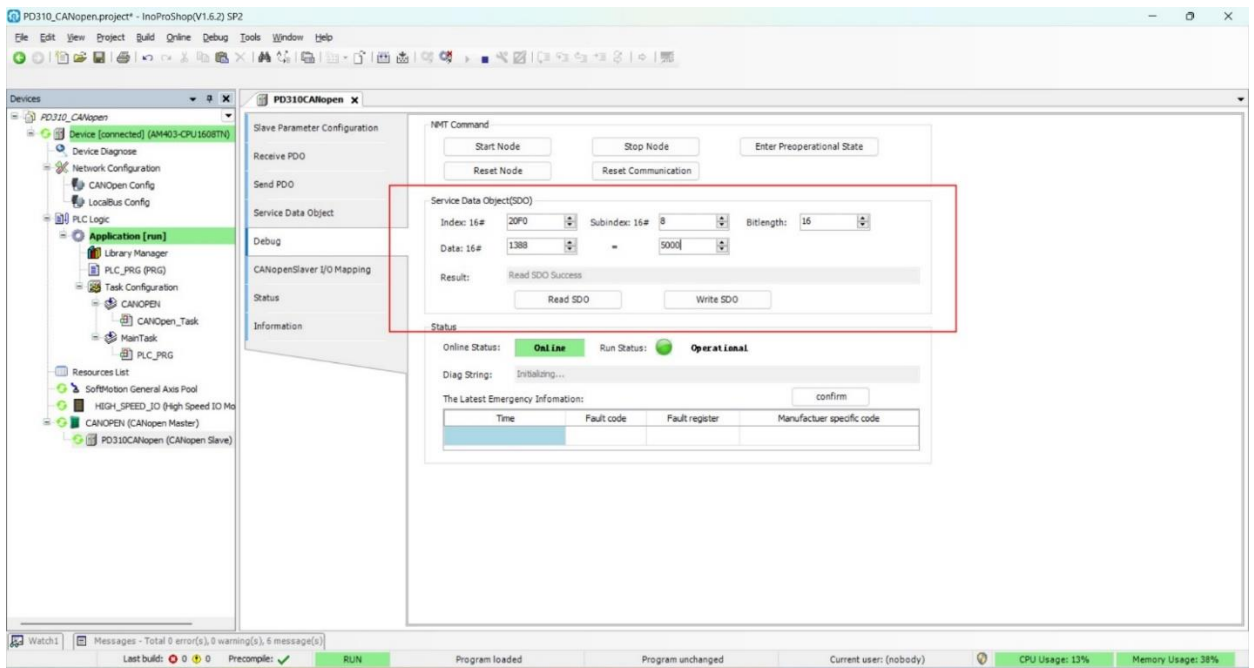


Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared Value	Unit	Description
Control Command			%QW1	UINT	1				Receive PDO
Communication set frequency			%QW2	INT	5000	3000			Receive PDO
OUTPUT 1			%QW3	UINT	3000				Receive PDO
OUTPUT 2			%QW4	UINT	200				Receive PDO
OUTPUT 3			%QW5	UINT	0				Receive PDO
OUTPUT 4			%QW6	UINT	0				Receive PDO
OUTPUT 5			%QW7	UINT	0				Receive PDO
OUTPUT 6			%QW8	UINT	0				Receive PDO
OUTPUT 7			%QW9	UINT	0				Receive PDO
OUTPUT 8			%QW10	UINT	0				Receive PDO
OUTPUT 9			%QW11	UINT	0				Receive PDO
OUTPUT 10			%QW12	UINT	0				Receive PDO
Drive status			%W1	UINT	1				Send PDO
Running frequency			%W2	UINT	5000				Send PDO
INPUT 1			%W3	UINT	3000				Send PDO
INPUT 2			%W4	UINT	200				Send PDO
INPUT 3			%W5	UINT	0				Send PDO
INPUT 4			%W6	UINT	0				Send PDO
INPUT 5			%W7	UINT	0				Send PDO
INPUT 6			%W8	UINT	0				Send PDO
INPUT 7			%W9	UINT	0				Send PDO
INPUT 8			%W10	UINT	0				Send PDO
INPUT 9			%W11	UINT	0				Send PDO
INPUT 10			%W12	UINT	0				Send PDO

- 14) The AC drive parameters are read via SDO, which needs to be selected via the object dictionary.

Example:

Read the AC drive parameter F0-07 through SDO, because the concept of index 0 exists in the object dictionary, the index that needs to be input is 0x20F0 (20h is the index high byte, which indicates the customized area of the equipment vendor, and F0h is the index low byte, which indicates the Function Code F0 group), and the sub-index is $0x07+1 = 0x08$. The final index that needs to be input is 0x20F008, and the value of the parameter F0-07 is read. The value of parameter F0-07 is shown below.



The screenshot displays the InoProShop software interface for configuring a CANopen slave. The 'Service Data Object (SDO)' configuration window is open, showing the following details:

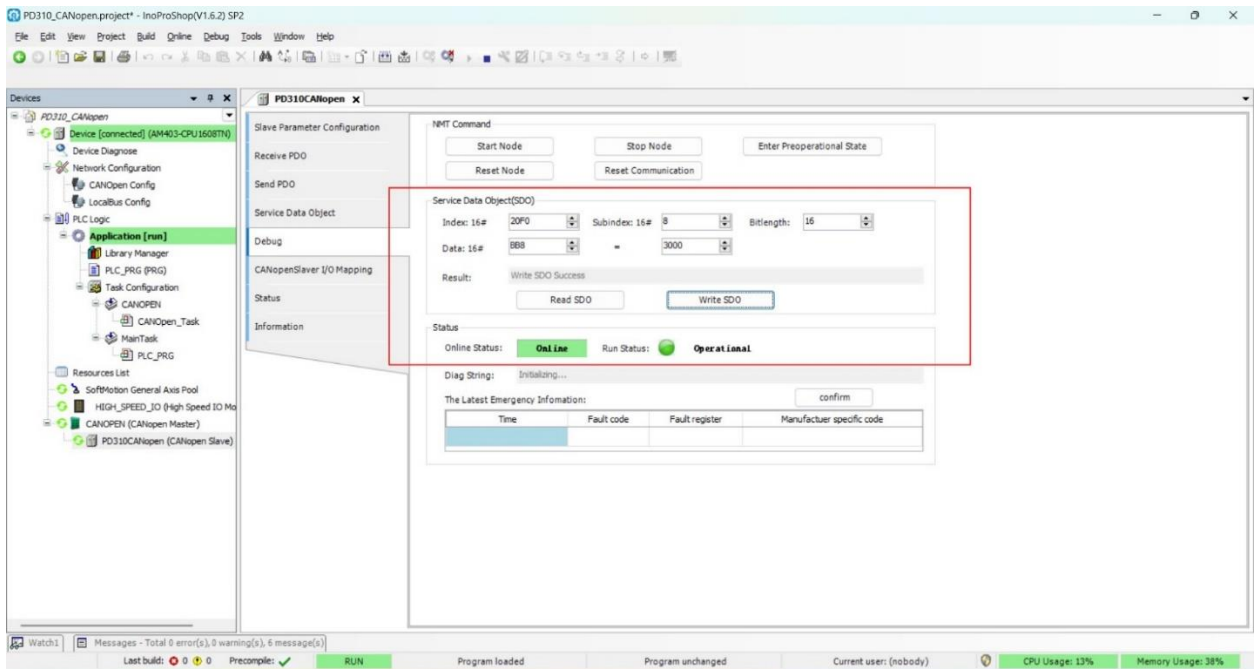
- Index: 16#**: 20F0
- Subindex: 16#**: 8
- Bitlength**: 16
- Data: 16#**: 1388 (with a range of 5000)
- Result**: Read SDO Success
- Status**: Online Status: **Online**, Run Status: **Operational**

The interface also shows a 'Messages' window at the bottom with the text: 'Messages - Total 0 error(s), 0 warning(s), 0 message(s)'. The status bar at the bottom indicates 'Program loaded', 'Program unchanged', 'Current user: (nobody)', 'CPU Usage: 13%', and 'Memory Usage: 38%'.

- 15) The AC drive parameters are modified via SDO, which requires a selection via the object dictionary.

Example:

Write the AC drive parameter F0-07 to 3000 (30.00Hz) through SDO, since the concept of index 0 exists in the object dictionary, the index that needs to be input is 0x20F0 (20h is the index high byte, which indicates the customized area of the equipment vendor, and F0h is the index low byte, which indicates the F0 group of the function code), and the sub-index is 0x07 + 1 = 0x08. The index that finally needs to be input is 0x20F008, and the result is shown below. The final index to be entered is 0x20F008, enter the modified value 3000, and the result is shown below.



The screenshot shows the InoProShop software interface for configuring a device. The 'Service Data Object (SDO)' section is highlighted with a red box. The configuration is as follows:

Index: 16#	Subindex: 16#	Bitlength:
20F0	8	16

Data: 16# BBB = 3000

Result: Write SDO Success

Buttons: Read SDO, Write SDO

Status: Online (green), Run Status: Operational (green)

Diag String: Initializing...

The Latest Emergency Information:

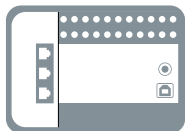
Time	Fault code	Fault register	Manufacturer specific code

At the bottom of the interface, the status bar shows: Last build: 0 0 0, Precompile: ✓, RUN, Program loaded, Program unchanged, Current user: (nobody), CPU Usage: 13%, Memory Usage: 38%.

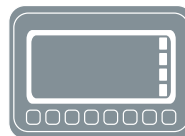
ВСЕ ДЛЯ АВТОМАТИЗАЦИИ:



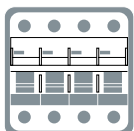
Реле



ПЛК



Панели оператора



НКА



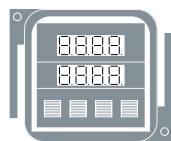
Электропривод



Датчики



Блоки питания



Управление

Официальный дистрибьютор:



**PROM
POWER**

www.prompower.ru

