CANopen communication protocol

PROMPOWER

User Manual

PROM
POWER

Introduction

General descriptions

Thank you for purchasing the PROMPOWER PMP20 series programmable controllers or CANopen communication modules. This manual mainly introduces the use of the PROMPOWER CAN communication module and the introduction of the CAN communication protocol. Before using the product, please read this manual carefully and make the wiring on the premise of fully understanding the content of the manual. For an introduction to software and programming, please refer to relevant manuals. Please deliver this manual to the end user.

Notices for users

Only operators with a certain level of electrical knowledge can perform wiring and other operations on the product. If there are any areas where the use is unknown, please consult our company's technical personnel. The examples listed in manuals and other technical materials are for user understanding and reference only, and do not guarantee certain actions. When using this product in combination with other products, please confirm whether it meets relevant specifications, principles, etc. When using this product, please confirm whether it meets the requirements and is safe. Please set up backup and safety functions on your own to avoid potential machine malfunctions or losses caused by the malfunction of this product.

Responsibility statement

Although the content in the manual has been carefully checked, errors are inevitable and we cannot guarantee complete consistency. We will regularly review the content in the manual and make corrections in subsequent versions. We welcome your valuable feedback. The content described in the manual is subject to change without prior notice.

Contents

1 CANbus 5
1.1 CANbus overview 5
1.2 Hardware interface 5
1.3 Relationship between CAN communication distance and baud rate 6
1.4 CAN and CANopen 6
2 CANopen technology overview 7
2.1 CANopen overview 7
2.2 CANopen function 8
2.3 CANopen communication parameters 10
2.4 CANopen network structure 10
3 CANopen communication protocol 11
3.1 Object dictionary (OD) 11
3.2 Service data objects (SDO) 13
3.3 Process data object (PDO) 13
3.4 Synchronization object (SYNC) 20
3.5 Network management (NMT) 21
3.6 Heartbeat protection 23
3.7 Node Guarding 24
3.8 Online node station number 25
3.9 Emergency object service (EMCY) 25
4 COBox configuration tool 27
4.1 X-NET Config tool 27
4.2 COBoxConfigForm interface 30

4.3 CANOPEN config tool 35
5 CANopen application example 39
5.1 Two PMP20 series PLCs communicate through PMP-COBOX-ED 39
Appendix 1 CANopen related register 44
Appendix 2 Object Dictionary List 47

1 CANbus

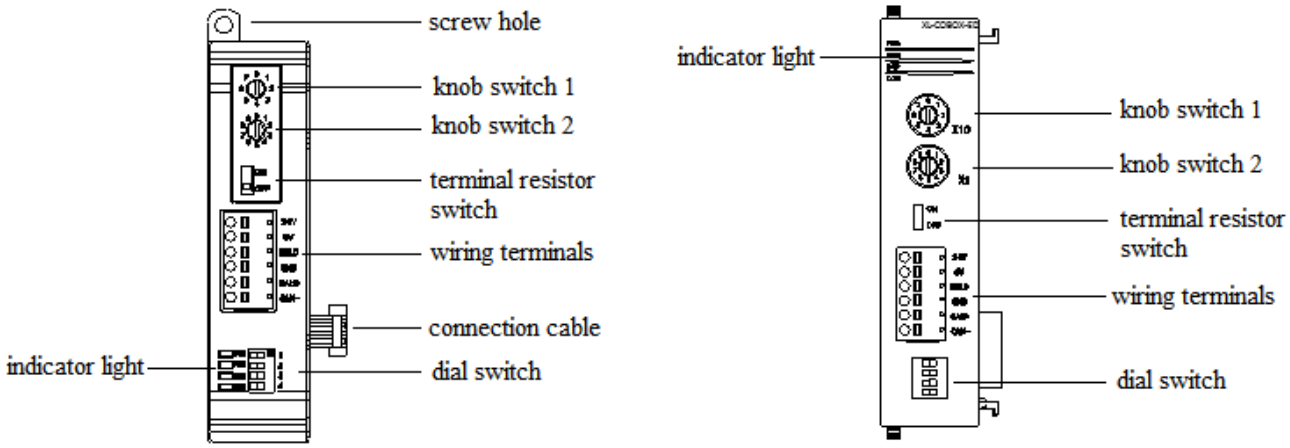
1.1 CANbus overview

CAN (Controller Area Network) bus is a serial data communication protocol developed by German BOSCH Company in 1983 to solve the data exchange between numerous control and test instruments in car. The communication medium can be twisted pair, coaxial cable or optical fiber. The communication rate can reach 1MBPS. One of the biggest characteristics of CAN protocol is that it abolishes the traditional station address geocoding and codes the communication data block instead. This block coding method can also enable different nodes to receive the same data simultaneously, which is very useful in distributed control systems.

The information transmission of CAN is carried out through messages, which have four types of message frames: data frames, remote frames, error frames, and overload frames. The data field of the CAN frame is relatively short, less than or equal to 8 bytes, and the data length is given in the control field. Short frame transmission not only reduces the message error rate, but also helps to reduce the transmission delay time of other sites. The confirmation of frame transmission is jointly completed by the sending station and the receiving station. The ACK field sent by the sending station contains two "idle" bits (recessive bit), and the receiving station immediately sends a "occupied" bit (dominant bit) after receiving the correct CRC field, giving the sending station a confirmation reply.

1.2 Hardware interface

Among the products that PROMPOWER supports CAN communication, the hardware interfaces vary for different products. The PMP-COBOX-ED module can be used as both the master and slave stations of the CANopen network. The PMP-COBOX-ED module supports one channel CANopen communication (for a detailed introduction of PMP-COBOX-ED. The CAN communication port of the PMP-COBOX-ED left extended module is integrated with the module power terminal in a 6PIN port, and the meanings represented by each terminal are shown in the following figure:



1.3 Relationship between CAN communication distance and baud rate

The relationship between CAN communication distance supported by PLC and baud rate is shown in the following table:

Baud rate	Distance	Cable type
1000Kbps	16m	Shielded twisted pair
500Kbps	90m	
250Kbps	180m	
125Kbps	300m	
100Kbps	500m	
50Kbps	800m	
10Kbps	1000m	

1.4 CAN and CANopen

In the OSI model, the direct relationship between the CAN standard and the CANopen protocol is shown in the figure below. The CAN bus only defines the first and second layers, which we call "the bottom layer" temporarily. That is to say, CAN does not specify an application layer, so we will temporarily call it "high-level". CAN applications in almost every industry need a high-level protocol to define the usage of 11/29 bit identifier (frame ID) and 8-byte data (frame data) in CAN messages. CANopen is such a protocol used in motor control, vehicles, rail transit, engineering machinery, medical treatment and other industries.

2 CANopen technology overview

2.1 CANopen overview

CANopen is an upper layer communication protocol built on the CAN bus application layer, including communication sub protocols and device sub protocols. It was jointly developed by manufacturers and users in collaboration with CiA and became the CENELEC EN 50325-4 standard in 2002.

CANopen has established standards in a wide range of industrial communications, such as mechanical engineering, drive systems and components, medical equipment, building automation, transportation, and so on.

The basic communication mechanism is called communication description, and most devices are described in the "device description" protocol. "Device description" defines different types of standard devices and their corresponding functions, and different manufacturers can configure and coordinate the use of a CANopen network through a description file (EDS file).

The length of the CANopen data segment is up to 8 bytes, which can meet the general requirements of control commands, working status, and test data in the industrial field. At the same time, 8 bytes do not take up too much bus time, ensuring real-time communication.

The CAN protocol adopts CRC verification and can provide corresponding error handling functions, ensuring the reliability of data communication. The outstanding characteristics, extremely high reliability, and unique design of CAN are particularly suitable for the interconnection of industrial process monitoring equipment. Therefore, it is increasingly valued by the industry and has been recognized as one of the most promising field buses.

2.2 CANopen function

In the CANopen communication equipment supported by PROMPOWER, it can be used as both the master station and a slave station of the CANopen network.

- **PMP-COBOX-ED as master station**

When used as the main station, it has the following functions:

When PMP-COBOX-ED is used as master station, the functions are as follows:

- Accord to CAN 2.0A standard
- Accord to CANopen standard communication sub-protocol DS301v4.02
- Support NMT Master service
- Error control: support Heartbeat/Node Guarding Protocol
- Support PDO service:
  - RxPDO supports 100, and the data quantity 512 bytes at most
  - TxPDO supports 100, and the data quantity 512 bytes at most
  - PDO transmission type: support event trigger, time trigger, synchronous cycle, synchronous aperiodic
  - PDO mapping: each PDO can map objects with a maximum size of 8 bytes (64 bits)
  - Supported mapping data types:

Storage space	Data type
8bit	INT8U, INT8S
16bit	INT16U, INT16S
32bit	INT32U, INT32S
64bit	INT64U, INT64S

- Support SDO service:
  - Client: 1
  - Supports standard SDO fast transmission mode (expedited SDO)
  - Support SDO service to read and write slave data in PLC ladder diagram
  - Support upper computer to read slave object dictionary with SDO
- Support Emergency Protocol:
  - It can save three latest emergency information for each slave station
  - Emergency information can be read through PLC ladder diagram
- Synchronous information generator (SYNC producer, range 0-65535ms)

- As the interface between PROMPOWER X-NET config software and CANopen network, the configuration software can directly configure CANopen network through PMP-COBOX-ED module.
- It can automatically exchange data with PLC. When programming, users only need to program the D register mapped in PLC, without writing instructions; when connecting with PLC, registers after D6000 will be used temporarily.

- **PMP-COBOX-ED as slave station**

When PMP-COBOX-ED is used as slave station, the functions are as follows:

- Accord to CAN 2.0A standard
- Accord to CANopen standard communication sub-protocol DS301v4.02
- Support NMT Slave service
- Error control: support Heartbeat/Node Guarding Protocol
- Support PDO service:
  - RxPDO supports 4, and the data quantity 32 bytes at most
  - TxPDO supports 4, and the data quantity 32 bytes at most
  - PDO transmission type: support event trigger, time trigger, synchronous cycle, synchronous aperiodic
  - PDO mapping: each PDO can map objects with a maximum size of 8 bytes (64 bits)
  - Supported mapping data types:

Storage space	Data type
8bit	INT8U, INT8S
16bit	INT16U, INT16S
32bit	INT32U, INT32S
64bit	INT64U, INT64S

- Support SDO service:
  - Server side: up to 63
  - Supports standard SDO fast transmission mode (expedited SDO)
- Support Emergency Protocol
- It can automatically exchange data with PLC. When programming, users only need to program the D register mapped in PLC, without writing instructions; when connecting with PLC, registers after D6000 will be used temporarily.

2.3 CANopen communication parameters

Item	Specification
Transmission mode	CAN
Electrical isolation	500 VDC
Transmission cable	Two communication cables, one shielding cable, and one grounding cable
Information type	PDO, SDO, SYNC, Emergency, NMT
Serial transmission speed	Support 10 kbps, 20 kbps, 50 kbps, 100kbps, 125 kbps, 250 kbps, 500 kbps, 1 Mbps

2.4 CANopen network structure

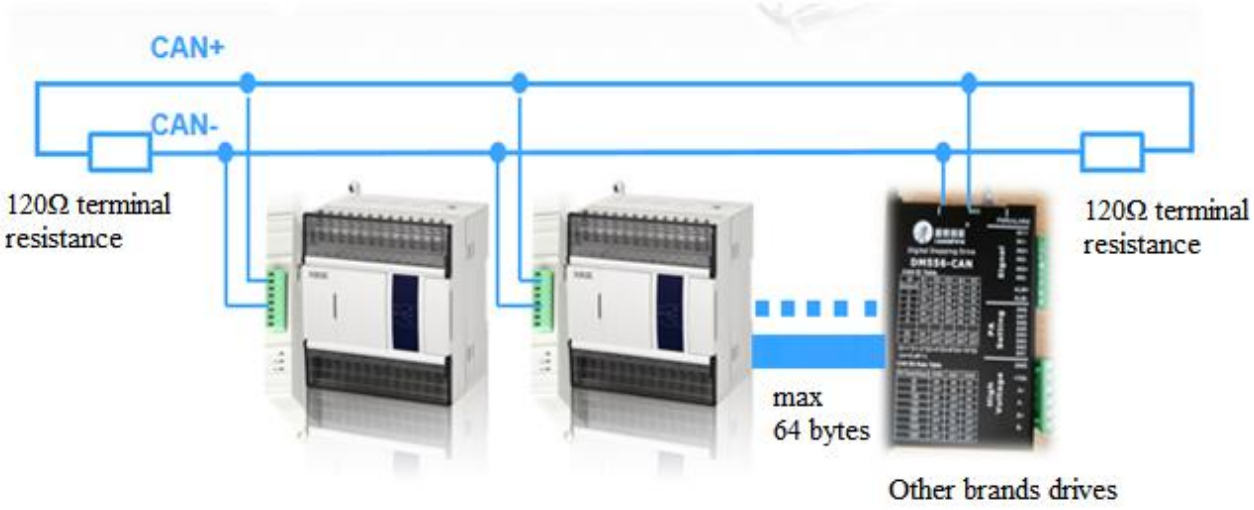
The PROMPOWER PMP-COBOX-ED needs to be used in conjunction with the PMP20 series PLCs when forming a CANopen network. It is connected to the PMP-COBOX-ED through the PLC's built-in left expansion communication port (COM3).

When PMP-COBOX-ED, PMP20 are connected to the CAN bus network, a linear topology structure is used. Only CAN+ (CAN_H) needs to be connected to CAN+ (CAN_H), and CAN- (CAN_L) needs to be connected to CAN- (CAN_L) to establish communication. If the field has high requirements for bus anti-interference ability, GND needs to be connected.

In order to enhance the reliability of CAN communication and eliminate the reflection interference of CAN bus terminal signals, terminal resistors are usually added to the two farthest endpoints of the CAN bus network. The value of terminal resistance is determined by the characteristic impedance of the transmission cable. For example, if the characteristic impedance of a twisted pair is 120Ω, the farthest two terminals on the bus need to install a 120Ω terminal resistance. If the number of nodes is greater than 2, the intermediate nodes do not need to install terminal resistors. PMP-COBOX-ED comes with a 120Ω terminal resistor dial switch (up to ON and down to OFF). If other CANopen devices do not come with terminal resistors, users need to install them by themselves. The CAN bus network supports up to 32 nodes, with a communication speed of up to 1M.

Note: A maximum of 29 nodes are supported at baud rate 1M.

Taking the PMP-COBOX-ED module as an example, the connection of the CANopen bus is shown in the following figure:



3 CANopen communication protocol

The object dictionary in CANopen is the most core concept of the CANopen protocol.

CANopen's data transmission includes two different data transmission mechanisms, using PDO (Process Data Object) to handle the rapid exchange of short process data. Access the entry of object dictionary through SDO (Service Data Objects).

The communication difference between PDO and SDO is that PDO belongs to process data and is used for real-time data transmission, which is the most important data transmission method in CANopen. SDO belongs to service data and is mainly used for parameter configuration of CANopen master and slave nodes.

Using CANopen will occupy some ordinary registers inside the PLC. The default master station will occupy D6000-D6645. PMP-COBOX-ED can right-click on the device list - Properties - Device information interface in the configuration tool to modify the starting address of the occupied registers. The slave station occupies D6000~D6032 and cannot be modified. The main station related register addresses mentioned in this manual are all based on the default D6000 starting address as an example.

3.1 Object dictionary (OD)

The Object Dictionary (OD: Object Dictionary) is the central element of each CANopen device. Each node in the network has an Object Dictionary that describes all parameters of the device and its network behavior, including the storage location of communication data, which is also included in its index. An object dictionary is an ordered group of objects, with each object addressed by a 16 bits index value ranging from 0x0000 to 0xFFFF.

In order to avoid having no index to allocate when there is a large amount of data, an 8-bit index value is also defined under certain indexes. This index value is usually referred to as a sub index, with a range of 0x00 to 0xFF. The specific parameters within each index can be represented by a maximum of 32 bits variables, namely Unsigned32, which is four bytes. This table becomes transitive and is called a device description file or EDS file (Electronic Data Sheet).

Each CANopen device has an object dictionary, and for the master node in the CANopen network, there is no need to access every object dictionary entry of the CANopen slave node.

The items in the CANopen object dictionary are described by a series of sub protocols. The sub protocol describes each object in the object dictionary its function, name, index, sub index, data type, and whether the object is required, read and write properties, etc. This ensures compatibility between devices of the same type from different vendors.

The structure of the object dictionary can be found in the following table:

Index	Object
0000	Reserved
0001~009F	Data type
00A0~0FFF	Reserved
1000~1FFF	Communication sub protocol area
2000~5FFF	Manufacturer's specific protocol area
6000~9FFF	Standard device sub protocol area
A000~ FFFF	Reserved

Users need to pay attention to the communication object sub protocol area and the manufacturer specific sub protocol area with green colour.

(1) The Communication Profile Area defines all object parameters related to communication, such as object dictionaries such as 1000H, 1400H, 1A00H, etc. Its detailed description can be found in Appendix 2.
(2) The Manufacturer specific Profile is usually used to store the application data of the applied sub protocol. The communication object sub protocol area described above is the communication parameter that stores these application data. For details on parameters such as 2000H and 6000H, please refer to Appendix 2.

For special functions not defined in the device sub protocol, manufacturers can also define object dictionary objects in this area according to their needs. Therefore, for different vendors in this area, the definition of the same object dictionary item may not necessarily be the same.

3.2 Service data objects (SDO)

The Service Data Objects is mainly used for the parameter configuration of the CAN-open master station to the slave node. The biggest feature of SDO is that a customer's request must have a response from the server, with the address of the designated receiving node (Node ID), and the designated receiving node needs to respond to the CAN message to confirm that it has been received. If the timeout does not confirm, the sending node will resend the original message.

This communication method belongs to the common "server client" communication model, which is commonly referred to as the polling mode. The visitor is called the client, and the CANopen device that accesses the object dictionary and provides the requested service is called the server. SDO enables clients to access objects in the server object dictionary by using indexes and sub indexes. In a CANopen network, only the NMT host can initiate SDO communication for node parameter configuration or key parameter transfer.

The client can use SDO read and write instructions in the ladder diagram to access the object dictionary of each server. Please refer to chapter 5 SDO read and write instructions.

3.3 Process data object (PDO)

3.3.1 PDO features

Process data object (PDO) is used for real-time data transmission and is the main data transmission method in CANopen. Due to the need to distinguish between the input and output of each CANopen node, PDO is divided into Transmit-PDO (TxPDO) and Receive-PDO (RxPDO). PDO is a one-way transmission that does not require the receiving node to respond to CAN messages for confirmation. In communication terminology, it belongs to the "production consumption" model. The sender of PDO is called the producer, and the receiver is called the consumer. PDO data transmission can be transmitted from one producer to one or more consumers, with a data transmission limit of 1 to 8 bytes.

The attributes of PDO can be configured in the object dictionary, including communication parameters and PDO mapping parameters:

(1) PDO communication parameters: describe the communication function of PDO, define the COB-ID used by the device, transmission type, inhibit time, and event time, etc.

(2) PDO mapping parameters: Contains PDO transmission content information (including index, sub index, and mapping object length). Producers and consumers must be aware of this mapping to interpret PDO content.

PMP-COBOX-ED can use 100 TxPDOs and 100 RxPDOs as the master station, and 4 TxPDOs and 4 RxPDOs as the slave station. Each PDO can map up to 8 bytes.

Master station	Communication parameter index	Mapping parameter index	Mapping object index
TxPDO 0-99	1800-1863	1A00-1A63	6080-60F0
RxPDO 0-99	1400-1463	1600-1663	6000-6070

Slave station	Communication parameter index	Mapping parameter index	Mapping object index
TxPDO 0-3	1800-1803	1A00-1A03	2100
RxPDO 0-3	1400-1403	1600-1603	2000

3.3.2 PDO communication parameters

PDO communication parameters define the COB-ID used by the device, transmission type, inhibit time, and event time. The RxPDO communication parameters are located at 1400h-1463h in the object dictionary index, and the TxPDO communication parameters are located at 1800h-1863h in the object dictionary index.

Each index represents a communication parameter set for a PDO, with sub indexes pointing to specific parameters, as shown in the following table:

Index	Sub index	Description	Type
RxPDO: 1400h -1463h TxPDO: 1800h -1863h	00	Number of parameter entries: how many parameters are there in this index	Unsigned8
	01	COB-ID: The corresponding CAN frame ID sent or received by this PDO	Unsigned32
	02	Sending type: refers to the transmission form that this PDO sends or receives, usually using cyclic synchronization and asynchrony	Unsigned8
		00: Circular asynchronous	
		01-F0: Circular synchronous	
		FF: asynchronous	
	03	Inhibit timer	Unsigned16
	05	Event timer	Unsigned16

Index	Sub index	Description	Type
	06	SYNC start value: The PDO for cyclic synchronous transmission is sent only after receiving several synchronization packets, and this synchronization starting value is the number of synchronization packets. For example, set it to 2, which means sending only after receiving 2 synchronization packets.	Unsigned8

(1) CAN identifier for PDO

The CAN identifier of PDO, also known as the Communication Object Identifier (COB-ID) of PDO, contains control bits and identification data, and is used to specify the corresponding CAN frame ID sent or received by PDO. The COB-ID is located on sub index 01 of the communication parameters (RxPDO: 1400h~1463h, TxPDO: 1800h~1863h).

In PDO predefinition, the PDO number is artificially specified, and each slave station's PDO has a different COB-ID for differentiation. The COB-ID of the master station's PDO is not defined, and is automatically assigned based on the COB-ID of the communication object.

Slave COB-ID naming rule:

RxPDO number	COB-ID (hex)	TxPDO number	COB-ID (hex)
RxPDO 1	200 + slave station number	TxPDO 1	180 + slave station number
RxPDO 2	300 + slave station number	TxPDO 2	280 + slave station number
RxPDO 3	400 + slave station number	TxPDO 3	380 + slave station number
RxPDO 4	500 + slave station number	TxPDO 4	480 + slave station number

(2) PDO transmission type

The transmission type of PDO is located on sub index 02 of the communication parameters (RxPDO: 1400h~1463h, TxPDO: 1800h~1863h).

Asynchronous transmission - triggered by events, including data change triggering and periodic event time triggering.

Synchronous transmission - related to synchronous frames in a network.

The sub index 02 of communication parameters (RxPDO: 1400h~1463h, TxPDO: 1800h~1863h) represents different transmission types and defines the methods for triggering TxPDO transmission or processing received RxPDO. The specific correspondence is shown in the table:

Transport type	Transmission type description		Note
	TxPDO	RxPDO	
AcyclicSyn (0)	When the data of the TxPDO mapping object changes and a synchronization frame is received, the TxPDO data is sent. When there is no change in TxPDO data, TxPDO data is not transmitted.	As long as the PDO is received, the latest RxPDO data received will be updated to the application at the next SYNC.	Synchronous aperiodic
LoopSyn (1-240)	The PDO transmitted synchronously with the SYNC synchronization frame only sends TxPDO data after receiving several synchronization frames. The value of this transmission type is the number of received synchronization frames. For example, if the transmission type is set to 2, it will only be sent after receiving 2 synchronization frames.	As long as the PDO is received, the latest RxPDO data received will be updated to the application at the next SYNC.	Synchronous cycle
241-253	Reserved		
ManufacturerAsyn (254)	TxPDO data is transmitted every other event time or when the data of the mapping object changes, and the event time is immediately reset. After TxPDO data is transmitted once, it is not allowed to transmit the TxPDO data again within the prohibited time period. When the event time is 0 and the data of the TxPDO mapping object changes, the TxPDO will be immediately sent. When there is no change in the TxPDO data, the TxPDO data will not be transmitted.	Immediately update the received data to the application.	Asynchronous
DeviceAsyn (255)	Same to transmission type 254		

Note:

The synchronization and asynchrony here refer to whether the sending of PDO is synchronized with the sending of synchronous frames.

The characteristic of synchronization is that the data update cycle is stable, but it cannot maintain real-time synchronization with data changes. Asynchronous means that once the data changes, the data will be updated immediately. This transmission mode responds quickly, but for frequently changing data, it is easy to generate large data load on the bus. Therefore, a disable time parameter is often configured to reduce the network load.

It is recommended to use synchronous PDO for parameters with low real-time requirements in the network, and asynchronous PDO for parameters with high real-time requirements. However, attention should be paid to configuring the prohibition time to protect the network load from impact.

(3) Inhibit timer

For TxPDO with asynchronous transmission (type 254 or 255), a prohibition time is defined and stored on sub index 03 of communication parameters (1800h~1863h) to constrain the minimum interval for PDO transmission and prevent TxPDO with frequent state changes from continuously occupying the CAN network, resulting in a significant increase in bus load. After setting the value, the transmission interval of the same TxPDO should not be less than the time corresponding to this parameter, which is measured in milliseconds.

Suggestion: When objects with frequent changes (such as feedback position, feedback speed, etc.) are configured to TxPDO and the transmission type of the TxPDO is asynchronous, it is recommended to set a certain prohibition time. Generally, the inhibit time is less than the event time. When the inhibit time is greater than the event time, TxPDO transmission is triggered every other event time.

(4) Event timer

For TxPDO with asynchronous transmission (type 254 or 255), event time is defined and stored on sub index 05 of communication parameters (1800h~1863h), which is the maximum interval for PDO transmission. The event time can also be regarded as a triggering event, and the corresponding TxPDO transmission is triggered every other triggering time. If other events such as data changes occur during the running cycle of the timer, TxPDO will also be triggered, and the event time will be immediately reset. If this time is 0, then TxPDO is an event change sent, and the unit of this parameter is ms.

(5) Setting the synchronization cycle

Synchronization period setting: It is recommended to calculate according to the empirical formula (default baud rate 1M):

Synchronization period (milliseconds) = [Total number of PDOs/9]/(40%) + 2

Assuming a CANopen network has a total of 12 axes, each with a sending and receiving PDO. So the total number of PDOs is 12*2 = 24. In each millisecond, the bus can transmit about 9 PDOs under full load (this data is the number of PDOs at 1M. For other baud rate, proportional operations can be performed with reference to 1M data. For example, the current baud rate is 500K, 500K/1M = 1/2, 9*1/2 = 4.5, so the current number of PDOs is 4.5, and so on for other baud rate). Considering the bus load margin, assume that the bus load is 40% (a relatively reasonable load rate), the required time for 24 PDO transmissions is: 24/9/(40%) =

6.67 (milliseconds). Considering the time cost of SDO, synchronization frames, heartbeat messages, emergency messages, etc. within the network, an additional 2 milliseconds are required. It is recommended to configure a synchronization cycle of 8.67 milliseconds.

The above empirical formula is also applicable to the setting of the inhibit time for asynchronous PDO.

3.3.3 PDO mapping parameters

The PDO mapping parameter contains a list of objects in an object dictionary that are mapped to the corresponding PDO, including indexes, sub indexes, and mapped object lengths. Both producers and consumers must know this mapping parameter in order to correctly interpret PDO content, which involves linking communication parameters, application data, and specific CAN message data. Each PDO data can have a maximum length of 8 bytes and can map one or more objects simultaneously.

The mapping parameter index (RxPDO: 1600h~1663h, TxPDO: 1A00h~1A63h) sub index 0 records the specific number of objects mapped by the PDO, and sub indexes 1-8 represent the content of the mapped objects. The data is stored in the vendor defined area after 2000h and 6000h.

Index	Sub index	Description	Type
RPDO: 1600h -1663h TPDO: 1A00h -1A63h	00	Record the specific number of objects mapped by this PDO	Unsigned8
	01-08	Record the content of the mapped object	Unsigned32
		The value of 20000108 h is mapped to sub index 01 h of index 2000 h, with an 8-bit object	
		The value 21000208 h is mapped to sub index 02 h of index 2100 h, with an 8-bit object	
		The value 60000316 h is mapped to sub index 03 h of index 6000 h, with an object of 16 bits	

The object indexes that can be mapped by CO-BOX mapping parameters are shown in the table below (where the master station takes the default starting address of D6000 as an example, and the slave station is fixed as D6000-D6031):

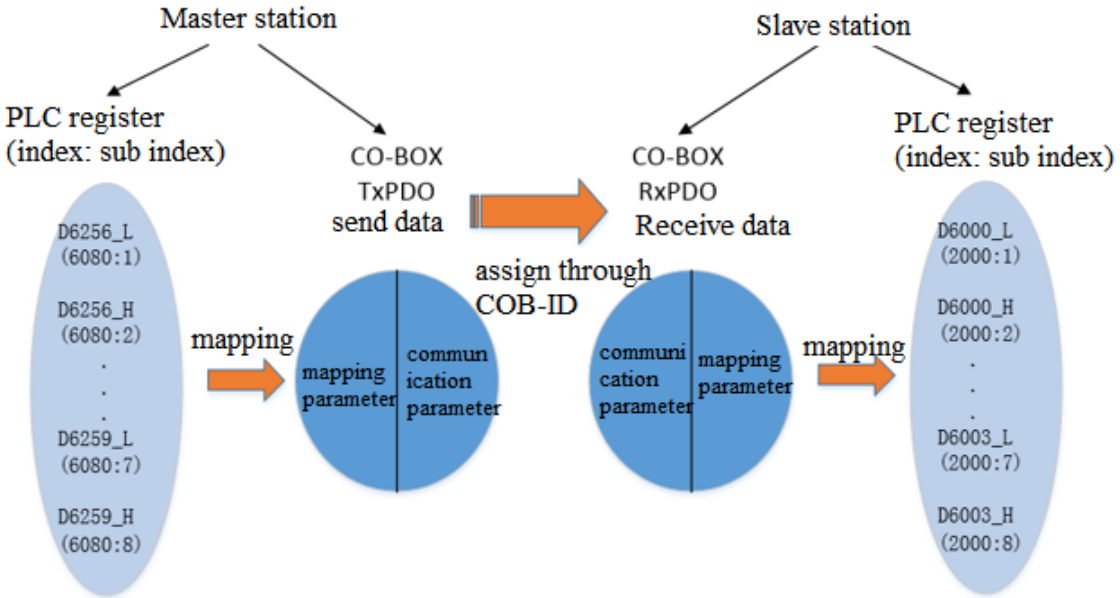
	PDO (mapping index)	Object data index	Sub index	Related register	Note
Master station	RxPDO 0-99 (1600h~1663h)	6000h	0-40h	D6000-D6031	Input register
		6010h	0-40h	D6032-D6063	
		6020h	0-40h	D6064-D6095	

	PDO (mapping index)	Object data index	Sub index	Related register	Note
		6030h	0-40h	D6096-D6127	
		6040h	0-40h	D6128-D6159	
		6050h	0-40h	D6160-D6191	
		6060h	0-40h	D6192-D6223	
		6070h	0-40h	D6224-D6255	
	TxPDO 0-99 (1A00h~1A63h)	6080h	0-40h	D6256-D6287	Output register
		6090h	0-40h	D6288-D6319	
		60A0h	0-40h	D6320-D6351	
		60B0h	0-40h	D6352-D6383	
		60C0h	0-40h	D6353-D6415	
		60D0h	0-40h	D6416-D6447	
		60E0h	0-40h	D6448-D6479	
		60F0h	0-40h	D6480-D6511	
Slave station	RxPDO 0-3 (1600h~1603h)	2000h	0-20h	D6000-D6015	Input register
	TxPDO 0-3 (1A00h~1A03h)	2100h	0-20h	D6016-D6031	Output register

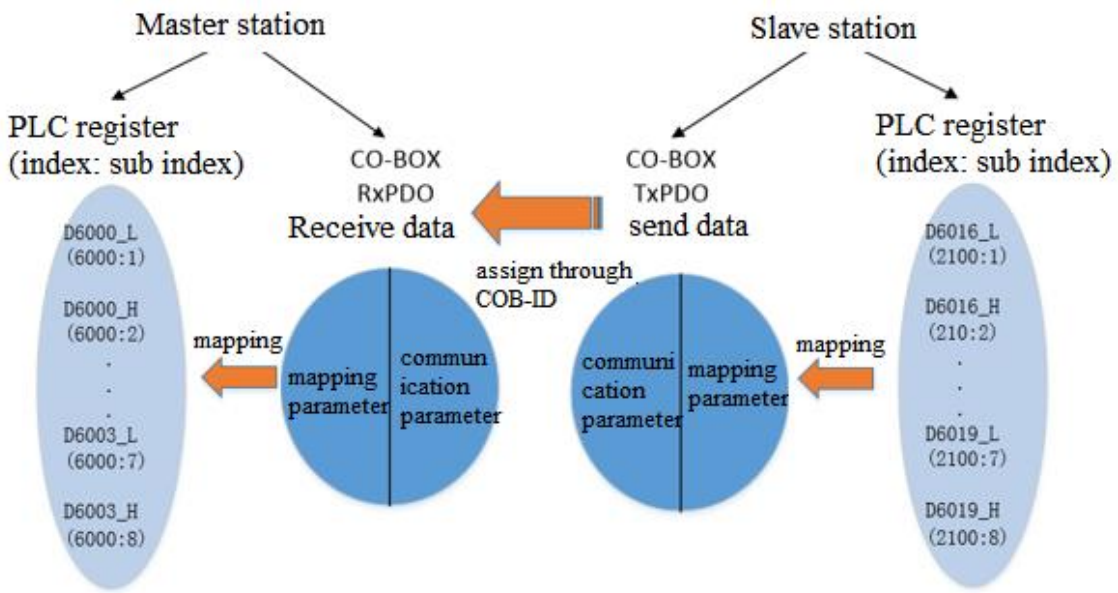
3.3.4 PDO communication diagram

The mapping relationship between the master slave PLC register and PDO, as well as the schematic diagram of the data transmission process, are shown below (where the master slave register address and index number are only examples):

(1) Master station sends data to slave station



(2) Slave station sends data to master station



3.4 Synchronization object (SYNC)

SYNC is a special mechanism that controls the coordination and synchronization between sending and receiving multiple nodes, mainly achieving synchronous transmission of the entire network. The transmission framework of synchronous objects is similar to that of PDO. The transmission of synchronous objects follows the producer consumer model, where the synchronous producer sends out the synchronous frame. All other nodes in the CAN network can receive the synchronous frame as consumers without feedback. Only one active synchronization generator is allowed in the same CAN network.

In the synchronization protocol, the predefined COB-ID of SYNC is 0x80, recorded on index 1005h, and there are two constraint conditions recorded on indexes 1006h and 1007h respectively:

Synchronous cycle period: Index 1006 h specifies the cycle period of synchronous frames.

Synchronization window time: Index 1007 h constrains the timeliness of sending PDO from the node after the synchronization frame is sent, that is, PDO sent within this time is only valid, and PDO exceeding the time will be discarded.

The transmission of synchronous PDO is closely related to synchronous frames. Generally, synchronization messages are sent by CAN network hosts. If there are two synchronization mechanisms within a network, different synchronization beats need to be set. For example, some nodes send PDO once per synchronization frame, while

others only send PDO once after receiving two synchronization frames. Therefore, the synchronization start value in the PDO parameter takes effect.

For synchronous RPDO, once the PDO is received, the received PDO will be updated to the application at the next SYNC.

For synchronous TPDO, it is divided into synchronous loop and synchronous non loop, as shown in the following table:

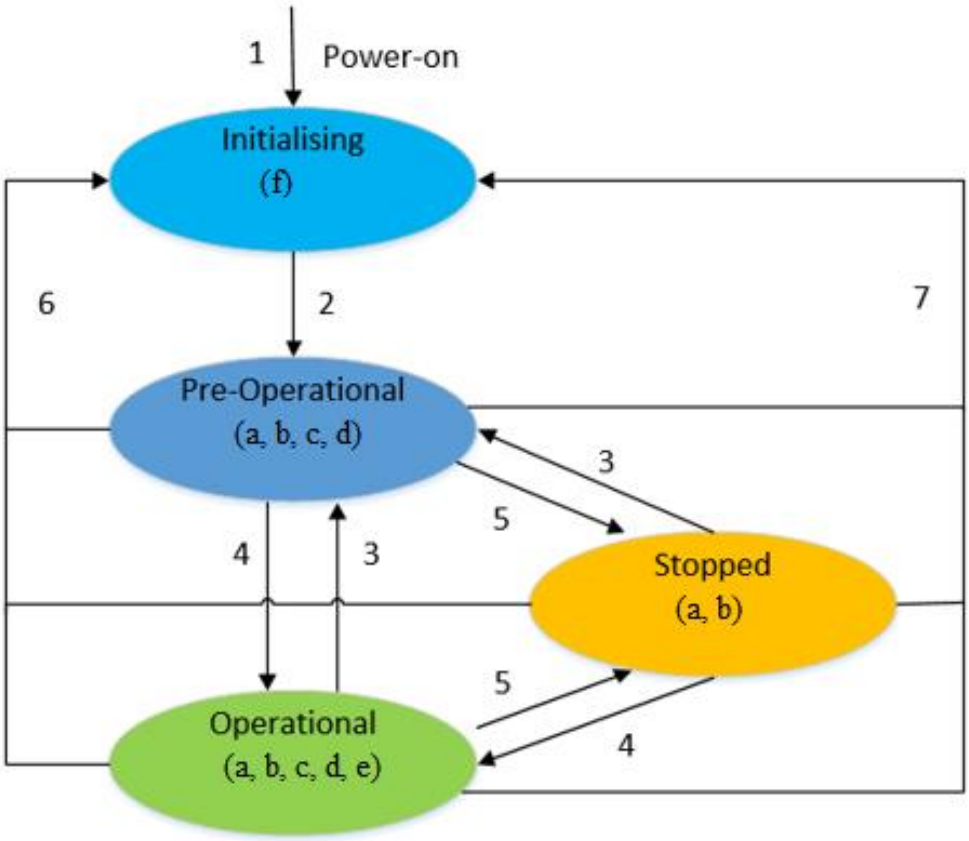
Type		Explanation
Synchronous TPDO	Synchronous non loop	The PDO transmission type is 0, and the content of the PDO mapping object has changed. It will be sent on the next SYNC.
	Synchronous loop	The PDO transmission type is 1-240, and as long as the SYNC specified for the transmission type is reached, regardless of whether the data has changed, the TPDO needs to be sent.

3.5 Network management (NMT)

In order for CANopen networks to be stable, reliable, and controllable, a network management master (NMT Master: Network Management Master) needs to be set up to manage the status of network devices. The NMT master also becomes the CANopen master station, which is a master-slave relationship structure. Therefore, there is only one NMT master node, and the corresponding other CANopen nodes are NMT slaves.

The communication messages between the NMT master and NMT slaves are called NMT network management messages. Only the NMT master node can transmit NMT network management messages, and all slave nodes must support NMT module control services. NMT module control does not require a response.

The NMT command can change the state of a single device or the entire network device at any time, and the state of each device is determined by specific attributes. PDO can only be transmitted in the operational state. The device can be configured in the Pre-Operational state. Only NMT commands can be transmitted in the Stopped state.



Explanation: the letters in parentheses indicate which communication objects can be used in different states.

a: NMT b: Node Guard c: SDO d: Emergency e: PDO f: Boot-up

State transition (3-7 initiated by NMT service), NMT name (in parentheses):

1: Automatically enter the initialization state after the power is turned on.

2: The device initialization ends and automatically enters the pre operation state, sending a Boot up message.

3: **Entering the pre operation state (0x04)**, the CANopen communication of the node is in the operational ready state. At this time, this node cannot perform PDO communication, but can perform SDO parameter configuration and NMT network management operations.

4: **After entering the running state (0x05)**, the node receives a start command from the NMT host, and CANopen communication is activated. After PDO communication is activated, it is transmitted according to the rules specified in the object dictionary. Similarly, SDO can also transmit data and modify parameters to the node.

5: **Entering a stop state (0x06)**, after receiving a stop command from the NMT host, the node's PDO and SDO communication are stopped, but NMT network management can still operate on the node.

6: **Reset node (0x02)**, reset the application program in the node, such as the initial value of switch output and analog output.

7: **Reset communication (0x03)**, the CANopen communication in the node is reset, and from this moment on, this node can start CANopen communication. Resetting communication will restore the communication parameters and mapping parameters of PDO to their initial values.

NMT control word (only available on the main station, default address is D6512):

bit	D6512_H	D6512_L
Value	Network configuration station number 0x1-0x40 (64), valid for a single node 0xFF, all nodes are valid. Except for the above values, all other values are invalid.	2: Reset nodes 3: Reset communication 4: Pre operation 5: Operation 6: Stop

NMT status word (both master and slave stations have, the default address is D6513 for the master station and D6032 for the slave station).

Register	Value	Status
Master station D6513 Slave station D6032	1	Initialising
	2	Application Reset
	3	Communication Reset
	4	Pre-Operatianal
	5	Operatianal
	6	Stopped

3.6 Heartbeat protection

In order to monitor whether the CANopen node is online and the current node status, CANopen applications usually require the online powered slave station to periodically send a message called Heartbeat, in order for the master station to confirm whether the slave station is faulty and disconnected from the network.

The heartbeat Protected mode adopts the producer consumer model. After the CANopen slave station fills in the heartbeat production time (ms) according to 1017h in its object dictionary, the node heartbeat protection function is activated and starts to generate heartbeat messages, while the CAN network master station will check according to the heartbeat consumption (timeout) time filled in 1016h. Once the heartbeat of the corresponding node producer is not received within the consumer heartbeat time range, the node is considered offline or faulty. When configuring heartbeat protection,

it is recommended that the heartbeat consumption (timeout) time 1016h be ≥ the heartbeat production time 1017h×2. Otherwise, it is easy to falsely report that the slave station has dropped.

When the CAN network master station does not receive a heartbeat message from a slave station within 1016h of heartbeat consumption time, it will be considered that the slave station is disconnected and trigger the PMP-COBOX-ED module ERR light to double flash.

3.7 Node Guarding

In CANopen applications, there is also a node guard mode that can monitor the status of the slave station through polling mode, which cannot coexist with the heartbeat message mode. The CAN network master can check the current status of each node through remote frames.

Node protection follows the master-slave model, and each remote frame must receive a response.

The objects related to node guardianship include node guardianship cycle 100Ch and node guardianship factor 100Dh. The value of 100Ch is the remote frame interval for node guardianship under normal circumstances, in milliseconds. The product of 100Ch and 100Dh determines the latest time for host queries. When nodes 100Ch and 100Dh are non zero and receive a request frame from the node guardian, activate the node guardian.

The master station sends node guard remote frames every 100Ch, and the slave must respond, otherwise it is considered that the slave station is disconnected. If the slave station does not receive a node guard remote frame within 100Ch * 100Dh, it is considered that the master station is offline.

Due to the gradual elimination of remote frames in the development of CAN, and the need for more main station overhead and increased network load for node guardianship, the CiA Association no longer recommends using node guardianship, which is more replaced by heartbeat packets.

Note:

1. Node protection and heartbeat protection cannot be used simultaneously.
2. Node protection and heartbeat protection time should not be set too short to avoid increasing network load.

3.8 Online node station number

After any CANopen slave goes online, in order to prompt the master that it has joined the network (for easy hot swapping) or to avoid conflicts with other slave Node-IDs. This slave station must send a node up message (boot up) to facilitate the master station to confirm whether the slave station is online, and the producer is the CANopen slave station. When the node guard and heartbeat protection are activated, the status of the stations in the network can be monitored through the online stations. When the nodes in the CAN network node list are normal, the corresponding bit is ON. When a node in the node list experiences an exception (including initialization failure and other exceptions that cause the slave to drop), the corresponding bit is in the OFF state.

Online node station number status register (only the main station has default addresses of D6514~D6517, which is only valid when node protection and heartbeat protection are configured):

Register	Corresponding online sites in the network							
	bit15	bit14	bit13	...	bit3	bit2	bit1	bit0
D6514	station16	station 15	station 14	...	station 4	station 3	station 2	station 1
D6515	station 32	station 31	station 30	...	station 20	station 19	station 18	station 17
D6516	station 48	station 47	station 46	...	station 36	station 35	station 34	station 33
D6517	station 64	station 63	station 62	...	station 52	station 51	station 50	station 49

3.9 Emergency object service (EMCY)

Emergency object (Emergency) refers to when an internal error occurs in a CANopen node, according to the standardization mechanism, the node will send a frame of emergency message, which records the internal error code of the device. The emergency message follows the producer consumer model, and the main station in the CAN network can receive the fault. Emergency messages belong to diagnostic messages and generally do not affect CANopen communication. In the protocol, the predefined COB-ID of EMCY is 0x80+station number, which is recorded on index 1014h.

Error information register (only available on the main station, occupied two words for each node's error information, default addresses of D6518~D6645).

Station 1 error message			Station 2 error message		
D6518_H	D6518_L	D6519	D6520_H	D6520_L	D6521
Error slave station number	Error type	Error code	Error slave station number	Error type	Error code

Station 3 error message			Station 4 error message		
D6522_H	D6522_L	D6523	D6524_H	D6524_L	D6525
Error slave station number	Error type	Error code	Error slave station number	Error type	Error code
……			……		
Station n error message			**Station 64 error message**		
(D6516+n*2)_H	(D6516+n*2)_L	D6517+n*2	D6644_H	D6644_L	D6645
Error slave station number	Error type	Error code	Error slave station number	Error type	Error code

Error type

Bit	Meaning
0	Common error
1	Current
2	Voltage
3	Temperature
4	Communication error (overflow, error status)
5	Device protocol specification
6	Reserved
7	Manufacturer specified

Error code

Error code (hex)	Meaning
0000	Error reset or no error
8100	Communication - General
8110	CAN overflow (object lost)
8120	CAN passive mode error
8130	Node protection or heartbeat error
8140	Restore from bus shutdown
8150	CAN-ID conflict
8210	PDO length error cannot be processed
8220	PDO over the length
8230	DAM Mpdo unable to process target object unavailable
8240	Unexpected SYNC data length

4 COBox configuration tool

4.1 X-NET Config tool

Before using the CANopen device PMP-COBOX-ED, it is necessary to configure the PLC serial port parameters. Here, we will introduce the use of the X-NET configuration tool to configure the PLC serial port parameters.

4.1.1 X-NET Config installation

1. The Config software must use version V2.2 or higher, which is generally built-in to the PROMPOWER PLC Studio software compression package.
2. Double click on the X-NET config tool and follow the installation wizard to install it.

4.1.2 X-NET Config tool using

When using software to configure the PLC, it is necessary to first connect the PLC to the computer using a USB download cable.

The USB download cable requires the installation of a driver to be used, and the driver usually completes the installation automatically when installing the X-NET Config tool.

Taking the PMP-COBOX-ED main station module as an example to illustrate the configuration process.

1. Configure the COM3 serial port

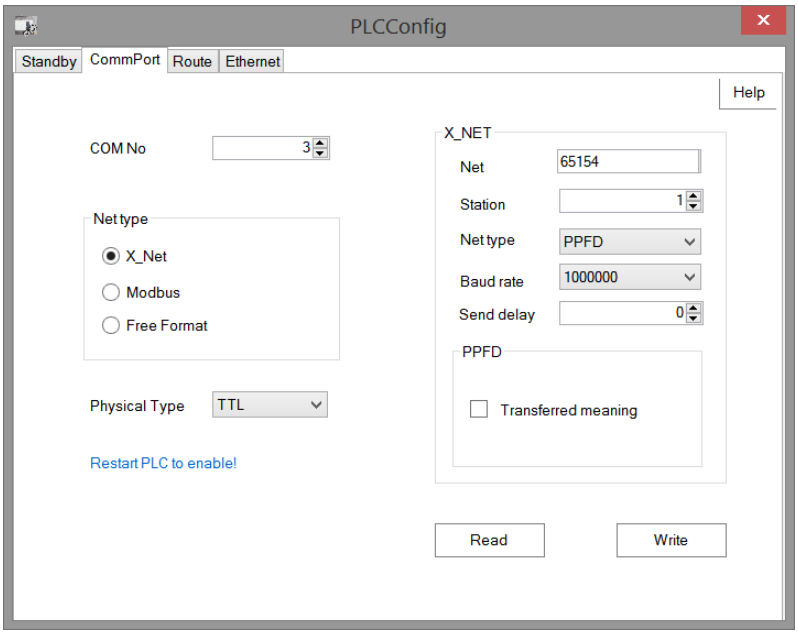
Open X-Net Config tool, click 

the interface for finding devices pops up. After finding the PLC, set the parameters of COM3 (Left Extended ED).

The serial port parameters of COM3 are set here so that PMP-COBOX-ED can communicate with the PLC through the left extended port.

So select "X-NET" in the "Net Type", "TTL" in the physical layer, "PPFD" in the network type, "1M" in the baud rate, "65154" in the network number, and the station number can be arbitrarily specified.

The parameters are shown in the following figure:

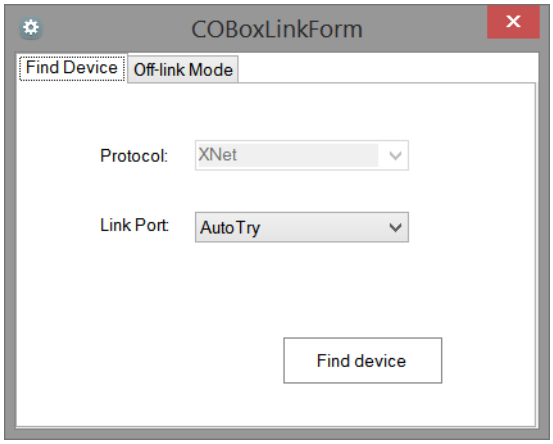


After the configuration is completed, click on 'Write' and a prompt 'Write Successfully' will pop up, indicating that the configuration is completed. It will take effect after being powered on again. After re powering on, the COM light of PMP-COBOX-ED flashes, indicating that the communication between PMP-COBOX-ED and PLC is normal.

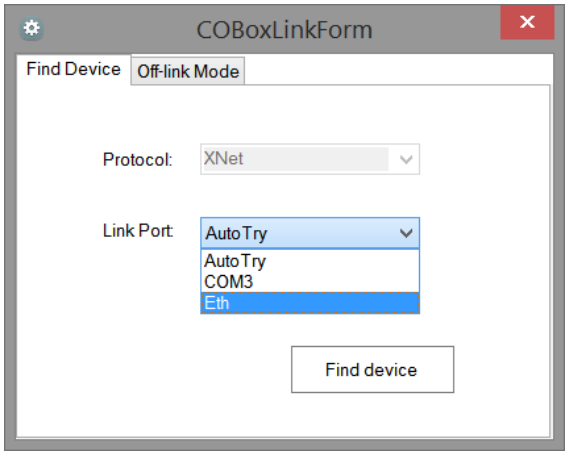
Note: the baud rate and station number here are not the baud rate and station number of PMP-COBOX-ED in CANopen network, but the parameters of PLC serial port. All PLCs using PMP-COBOX-ED need to set COM3 parameters, which will take effect after being powered on again.

2. Search CANopen device

Click COBox, the following screen will also pop up to find the device. Only X-NET can be used to find the device, that is, only USB port or Ethernet port can be used to find it. After correctly finding PMP-COBOX-ED, you can enter the home screen of CANopen configuration. Alternatively, offline mode can also be used to enter the configuration interface, but scanning, uploading, and downloading operations cannot be performed. After configuration is completed, pay attention to saving the configuration.



If you are searching for a device through an Ethernet port, you can select "Eth" for the "Link port" option:



[Find device]: The PLC of the CANopen master station must be connected, and the PLC serial port 3 parameter configuration is correct and the communication with PMP-COBOX-ED is normal, then PMP-COBOX-ED can be found and the configuration home screen can be entered for CANopen configuration. The search method can only use X-NET, and the search port can select automatic search, specified port and Ethernet search.

If the prompt "Find timeout" indicates abnormal communication between PLC and PMP-COBOX-ED after clicking on the [find device] button, please check:

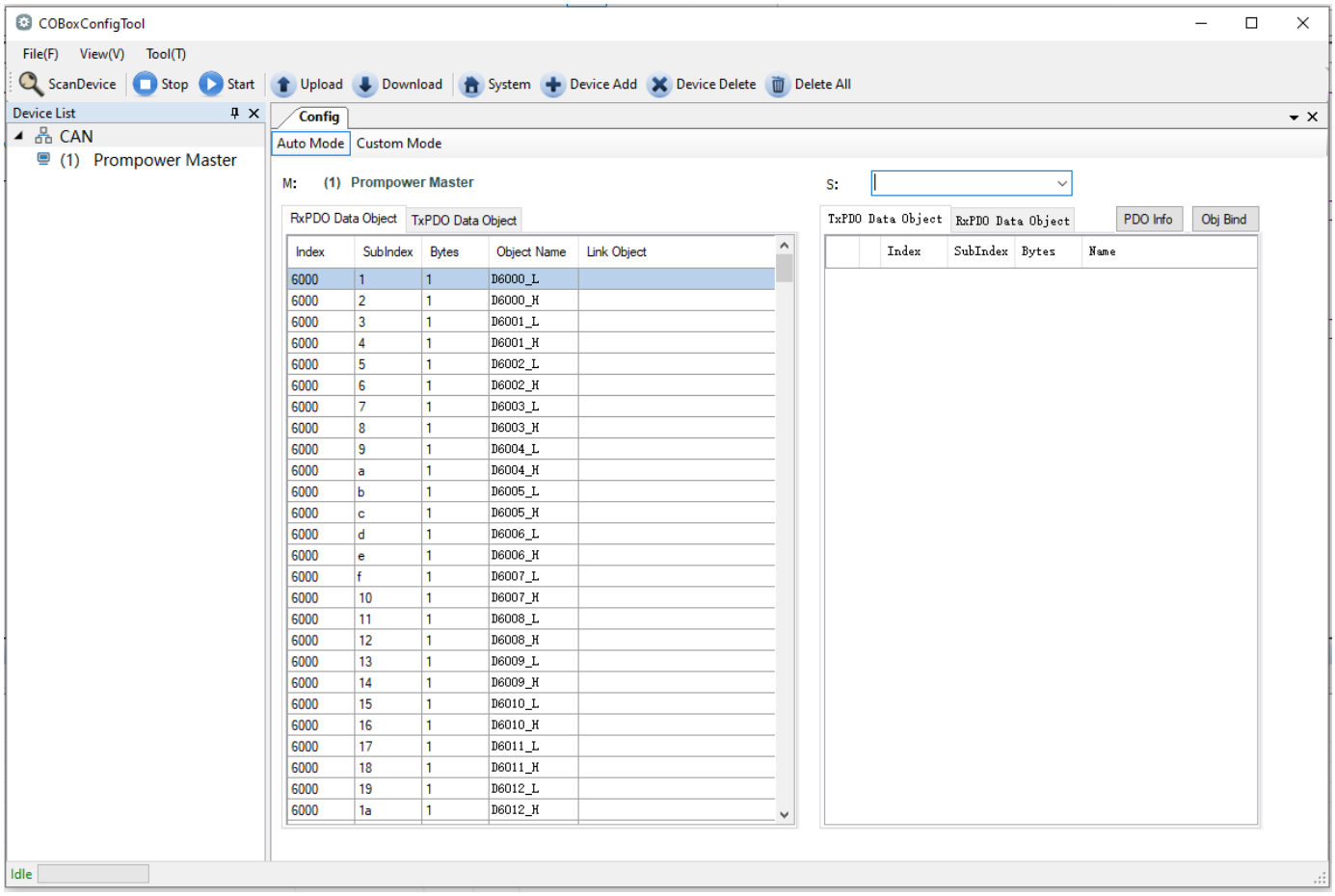
① Whether the COM3 parameter of PLC is correct;
② Communication connection between PLC and PMP-COBOX-ED;
③ Is the search for a CANopen master station.

[Off link mode]: When the device is not connected, you can also enter the configuration Home screen to configure CANopen, but you can only configure PDO and cannot perform scanning, uploading, downloading and other operations.

Note:

Only the PLC connected to the CANopen master station can find PMP-COBOX-ED and enter the CANopen configuration interface. The configuration of the slave station is also configured in this interface and does not need to be configured separately.

4.2 COBoxConfigForm interface



After entering the CANopen configuration interface, as shown in the figure, it is mainly divided into the following areas:

① Menu toolbar
② Device List Bar
③ Main Station Configuration Area
④ Slave Station Configuration Area

4.2.1 Menu toolbar

[Scan Slave Station]: Scan all stations in the network, and all stations scanned will be displayed in the device list;

[Stop]: Switch all devices in the network to a stopped state;

[Start]: Switch all devices in the network to the running state;

[Upload]: Upload the configuration of the CANopen device to ConfigTool;

[Download]: Download the configuration of ConfigTool to the CANopen device;

[System]: The synchronization cycle, heartbeat time, node protection and other parameters of the network can be set;

[Device Add]: Manually add a station;

[Device Delete]: Delete a single station;

[Delete All]: Delete all stations;

[File] → **[Save]**: Save the current configuration in the form of .cocfg file to the computer;

[File] → **[Open]**: Open the .cocfg file from the computer to the ConfigTool configuration tool;

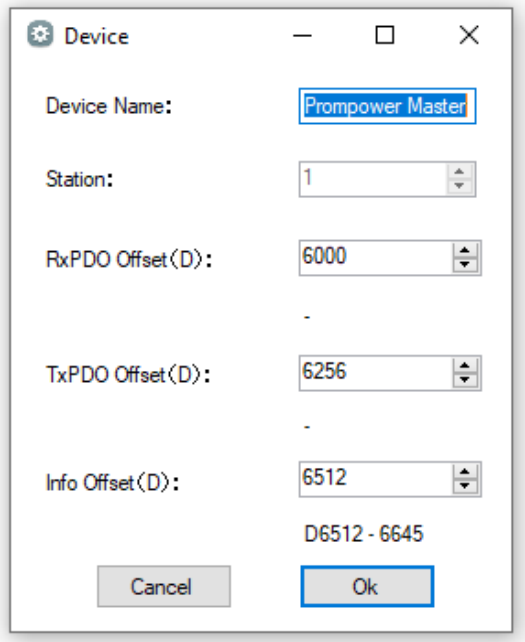
[File] → **[Add EDS]**: To add a new EDS file, it is generally necessary to add the device's EDS file before connecting to a new CANopen device. This file is provided by the device manufacturer;

[Tool] → **[Sdo]**: Perform SDO read and write operations, and read and write the values of the object dictionary in the upper computer. If you need to use the read and write SDO function in the ladder diagram, please use the SDO read and write instructions, as detailed in Chapter 4.

4.2.2 Device list bar

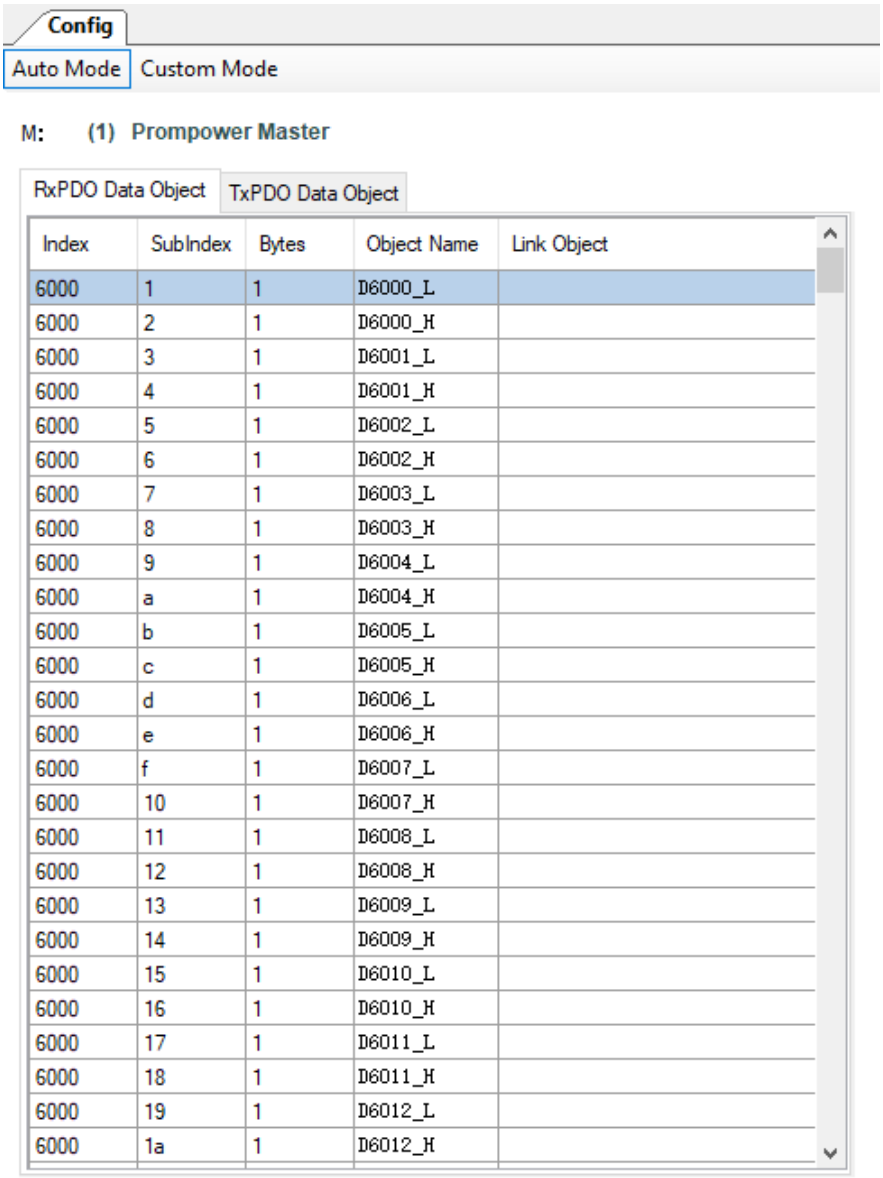
Display the station numbers and names of all stations in the network.

[Right click on the main station] → **[Property]**: You can modify the name of the main station and the starting address of the input register, output register, and other information occupation area of the main station, as shown in the following figure:



[Right click on the slave station] → **[Clone]**: Clone a station with the same PDO mapping object as the cloned slave station. Only the PDO configured mapping object will be cloned, and the connection will not be cloned.

4.2.3 Master station configuration area



Index	SubIndex	Bytes	Object Name	Link Object
6000	1	1	D6000_L	
6000	2	1	D6000_H	
6000	3	1	D6001_L	
6000	4	1	D6001_H	
6000	5	1	D6002_L	
6000	6	1	D6002_H	
6000	7	1	D6003_L	
6000	8	1	D6003_H	
6000	9	1	D6004_L	
6000	a	1	D6004_H	
6000	b	1	D6005_L	
6000	c	1	D6005_H	
6000	d	1	D6006_L	
6000	e	1	D6006_H	
6000	f	1	D6007_L	
6000	10	1	D6007_H	
6000	11	1	D6008_L	
6000	12	1	D6008_H	
6000	13	1	D6009_L	
6000	14	1	D6009_H	
6000	15	1	D6010_L	
6000	16	1	D6010_H	
6000	17	1	D6011_L	
6000	18	1	D6011_H	
6000	19	1	D6012_L	
6000	1a	1	D6012_H	

[Auto Mode]: When PDO is enabled in the slave station, the PDO objects of the slave station are automatically corresponding to the registers of the master station in a top-down order;

[Custom Mode]: When the slave station enables PDO, the first address of the master station register can be manually selected to correspond one by one with the PDO objects of the slave station in order;

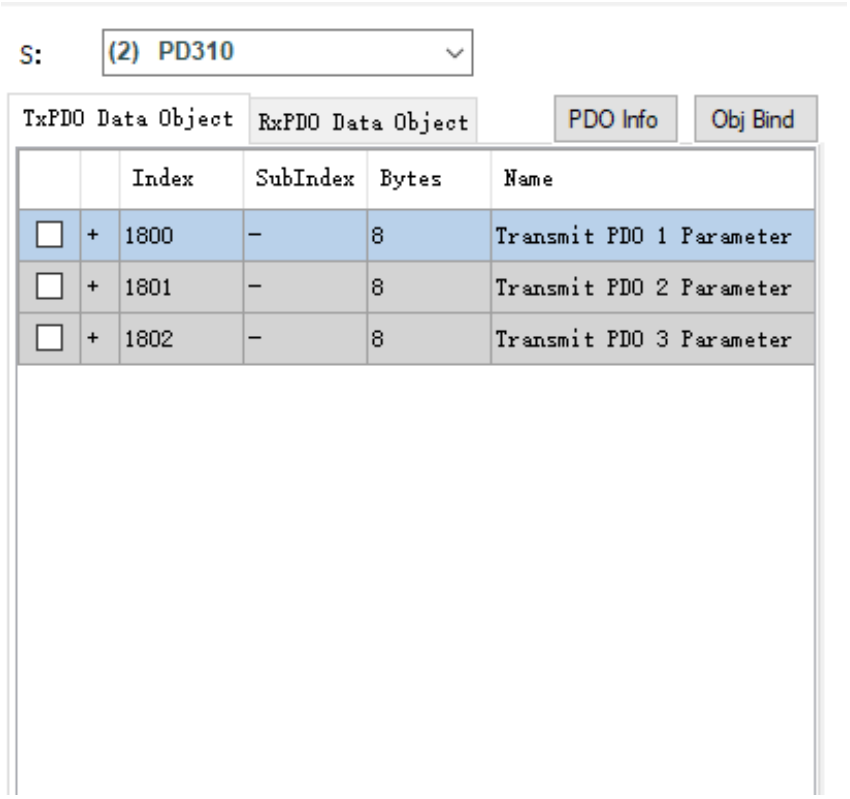
[RxPDO Data Object]: The master station switches to the input register corresponding to RxPDO, and the slave station will also switch to TxPDO accordingly;

[TxPDO Data Object]: The master station switches to the output register corresponding to TxPDO, and the slave station will also switch to RxPDO accordingly;

[Master Station Registers]: All input or output registers of the master station. When the slave station enables PDO, the mapping objects of the slave station will automatically correspond to the master station's registers in order. You can also customize the start register;

[Connection object]: After the slave station enables PDO, the station number, PDO index, and object name of the connected slave station are displayed.

4.2.4 Slave station configuration area



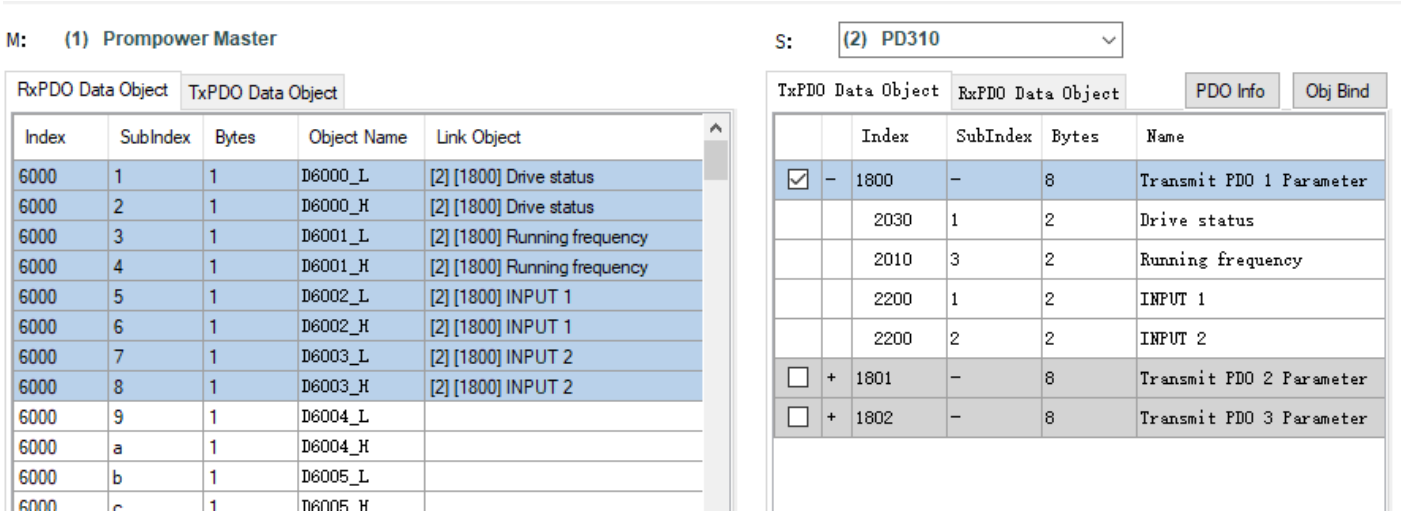
[Slave Station Equipment]: The currently configured slave station can be switched by clicking on the left device list bar or selected by clicking on the drop-down menu;

[RxPDO Data Object]: The slave station switches to the input register corresponding to RxPDO, and the master station also switches to TxPDO accordingly;

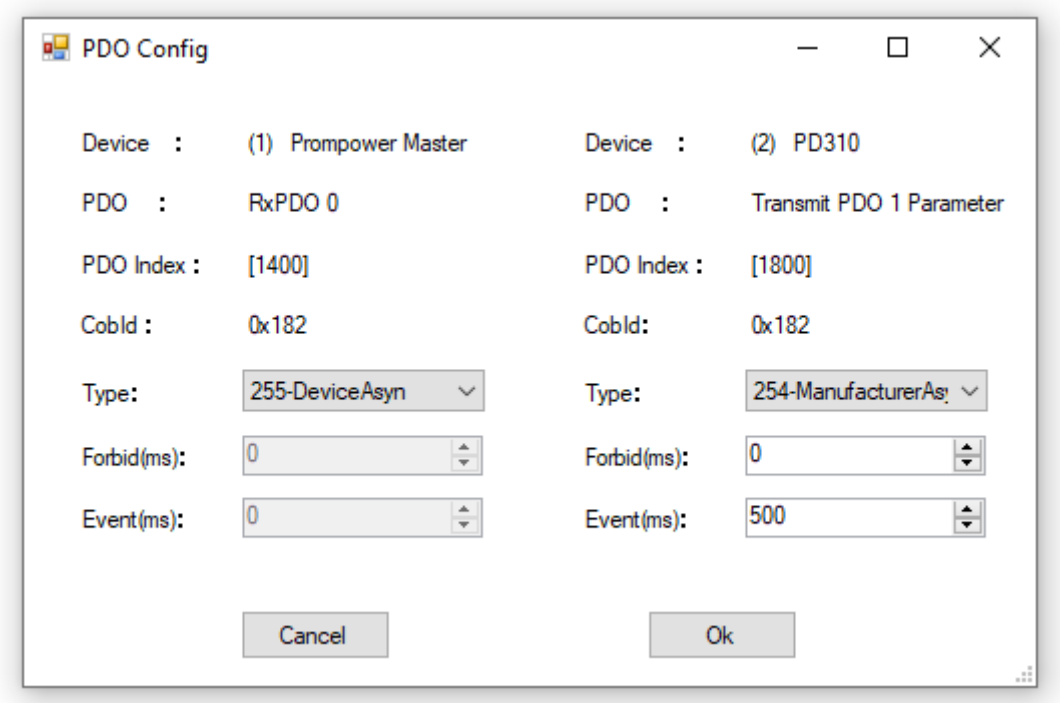
[TxPDO data object]: The slave station switches to the output register corresponding to TxPDO, and the master station also switches to RxPDO accordingly;

[PDO and objects]: The PDO of the slave station is configured with 8 mapping objects by default. Click "+" to view the currently configured mapping objects, click ☑ to enable this PDO, after enabling, it can be viewed in the configuration area of the main station.

[Object Binding]: Configure the mapping object for this PDO, with a maximum of 8 bytes bound to each PDO, as shown in the following figure:



[PDO Properties]: Set the communication parameters of the PDO. TxPDO can set the communication type, inhibit time, trigger time, and RxPDO can set the communication type, as shown in the following figure:

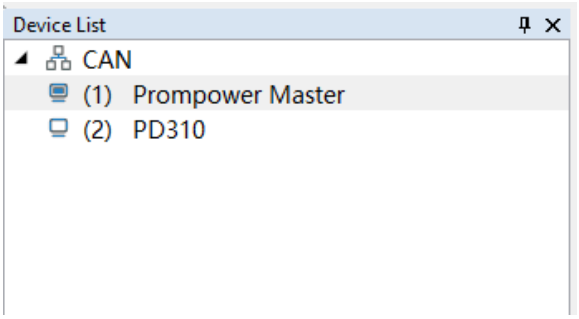


Note:

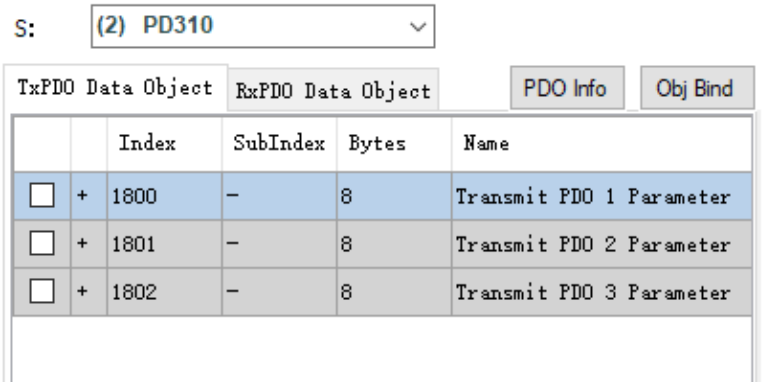
Non enabled PDO cannot set PDO attributes, and enabled PDO cannot perform object binding. Therefore, you must first configure object binding for PDO before setting PDO attributes. If you need to modify the mapping content after enabling PDO, please cancel the activation before modifying PDO object binding.

4.3 CANOPEN config tool

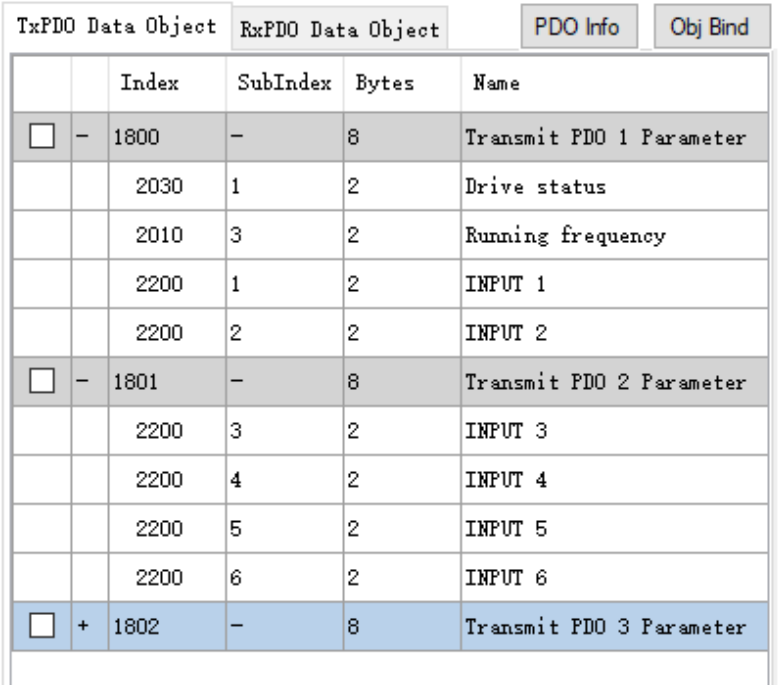
Click ScanDevice on the CANopen configuration interface, waiting for the scan to end. The list of devices on the left will display all stations in the network.



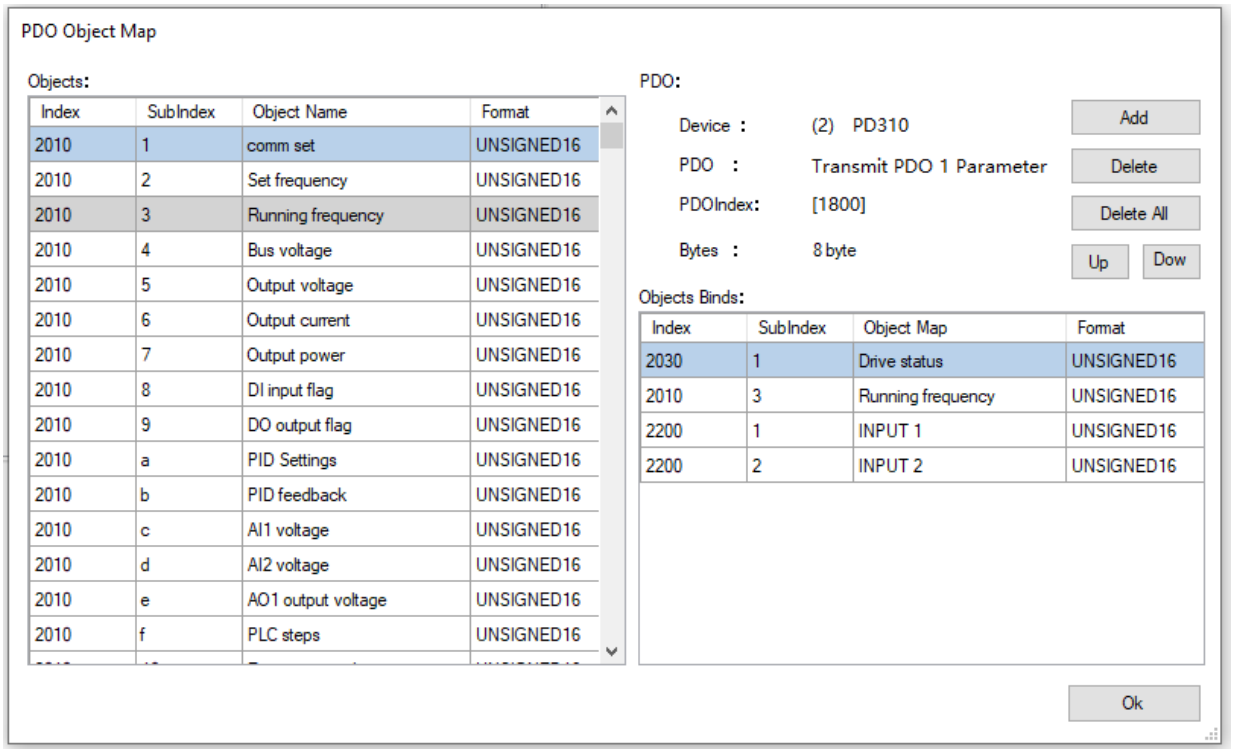
Click on the second slave station to configure the TxPDO mapping object for the slave station. On the right side of the slave station configuration area, you can see four PDOs, as shown in the following figure:



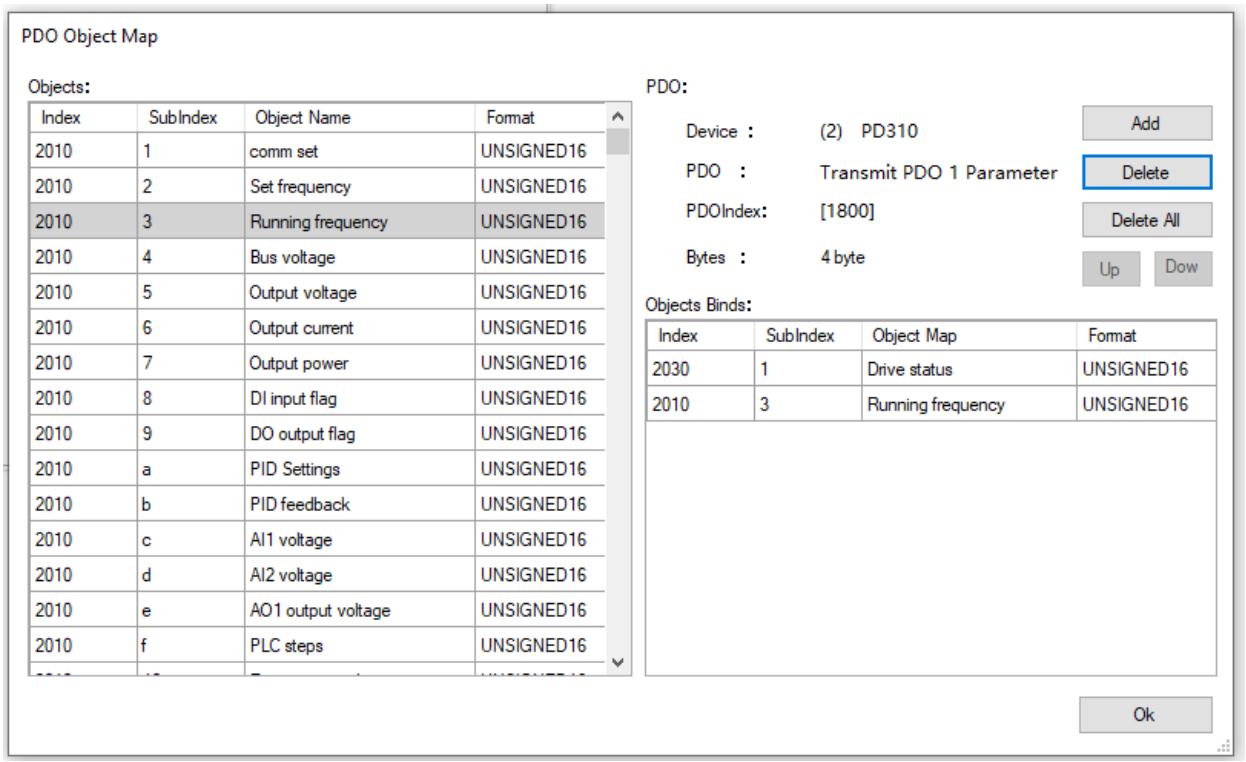
Click "+" to view the mapping objects of each PDO, as shown in the figure. The PMP-COBOX-ED slave station defaults to mapping 8 objects for each PDO.



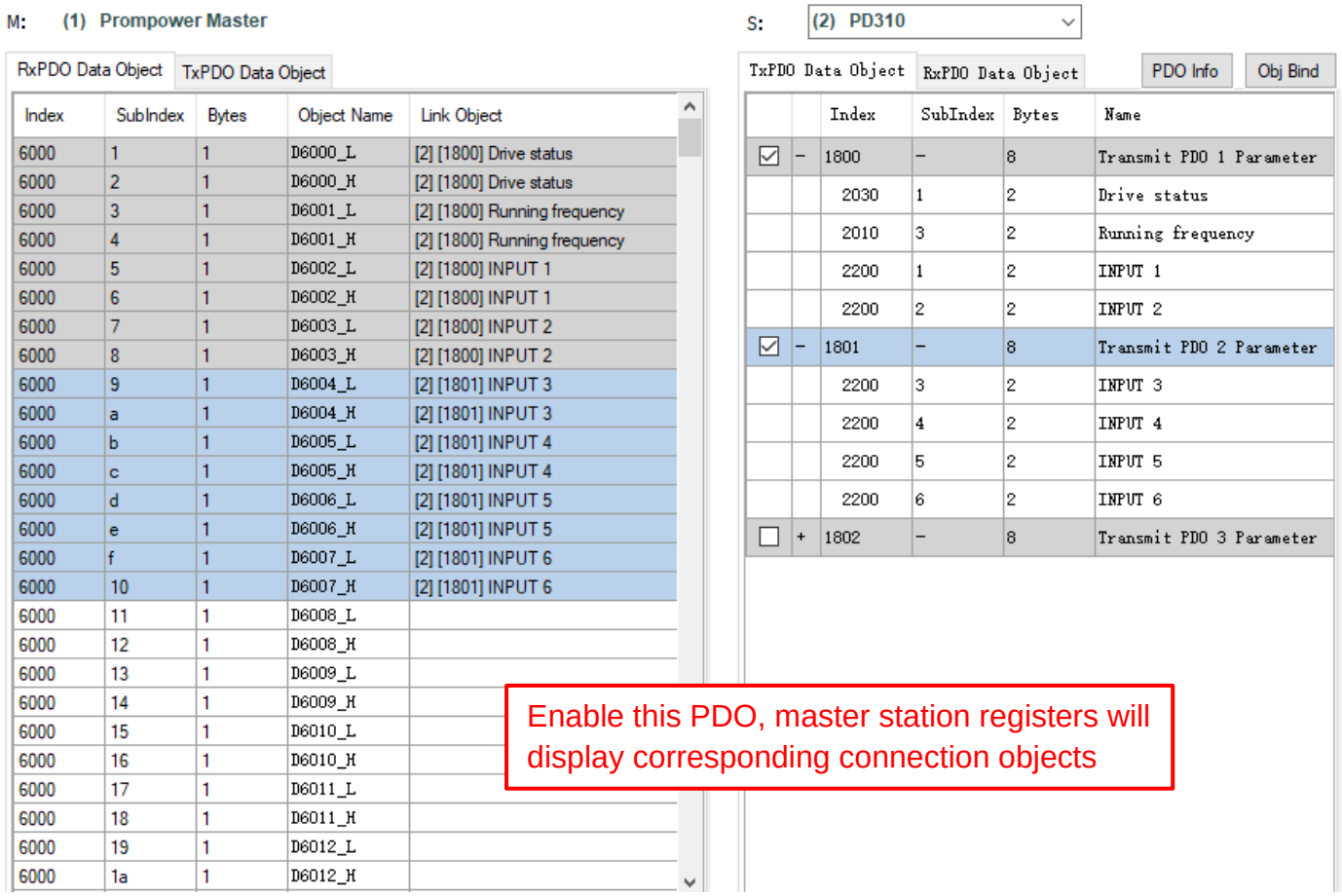
Select TxPDO0, click Obj Bind, you can modify the mapping objects of PDO according to your needs. In the "Objects" list on the left, gray indicates configured objects and white indicates unconfigured objects. Here, we delete some registers, as shown in the following figure:



After clicking **[Delete]**, the binding objects of TxPDO0 have been reduced from 4 to 2. In the "Objects" list on the left, you can see that the 2 objects have changed from gray to white.



After config, click to enable PDO. Here, we use auto mode to enable the first two PDOs. In the master station configuration area, the slave station objects will automatically correspond to the master station registers, as shown in the figure. You can also choose custom mode to select the first address of the corresponding register.

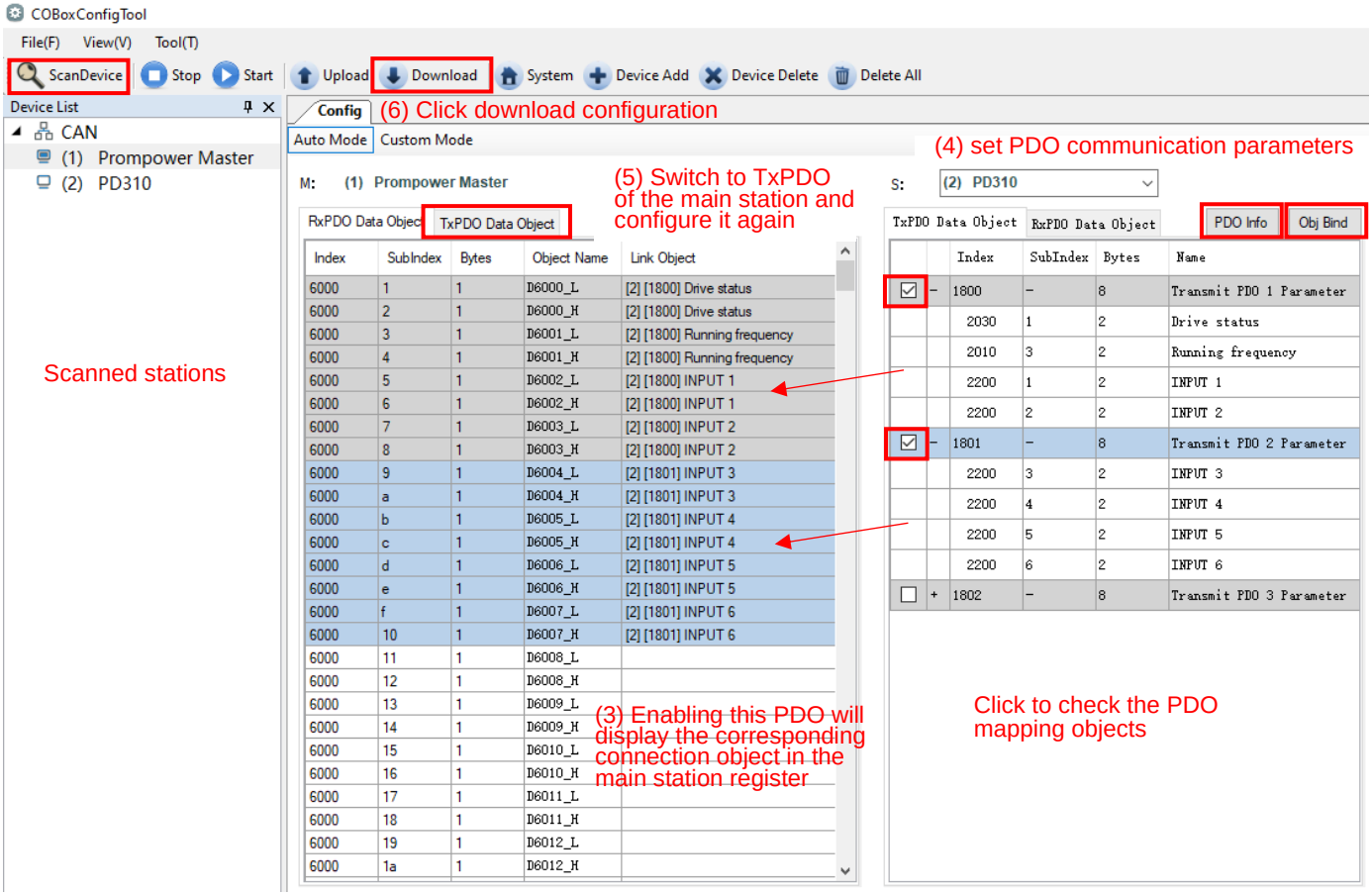


For enabled PDOs, you can click on **[PDO Info]** to set communication parameters for the selected PDO according to your own needs.

The RxPDO configuration method for the slave station is the same as TxPDO. Click on the slave station RxPDO data object, and the master station will also switch to the TxPDO data object accordingly.

After completing the configuration, click 'Download'. After the download is completed, it will automatically run and the configuration will take effect immediately.

Please refer to the following figure for the complete configuration method.


COBoxConfigTool
File(F) View(V) Tool(T)
ScanDevice Stop Start Upload Download System Device Add Device Delete Delete All
Device List
CAN
(1) Prompower Master
(2) PD310
Scanned stations
Config
(6) Click download configuration
Auto Mode Custom Mode
M: (1) Prompower Master
(5) Switch to TxPDO of the main station and configure it again
RxPDO Data Objec TxPDO Data Object
Index SubIndex Bytes Object Name Link Object
6000 1 1 D6000_L [2] [1800] Drive status
6000 2 1 D6000_H [2] [1800] Drive status
6000 3 1 D6001_L [2] [1800] Running frequency
6000 4 1 D6001_H [2] [1800] Running frequency
6000 5 1 D6002_L [2] [1800] INPUT 1
6000 6 1 D6002_H [2] [1800] INPUT 1
6000 7 1 D6003_L [2] [1800] INPUT 2
6000 8 1 D6003_H [2] [1800] INPUT 2
6000 9 1 D6004_L [2] [1801] INPUT 3
6000 a 1 D6004_H [2] [1801] INPUT 3
6000 b 1 D6005_L [2] [1801] INPUT 4
6000 c 1 D6005_H [2] [1801] INPUT 4
6000 d 1 D6006_L [2] [1801] INPUT 5
6000 e 1 D6006_H [2] [1801] INPUT 5
6000 f 1 D6007_L [2] [1801] INPUT 6
6000 10 1 D6007_H [2] [1801] INPUT 6
6000 11 1 D6008_L
6000 12 1 D6008_H
6000 13 1 D6009_L
6000 14 1 D6009_H
6000 15 1 D6010_L
6000 16 1 D6010_H
6000 17 1 D6011_L
6000 18 1 D6011_H
6000 19 1 D6012_L
6000 1a 1 D6012_H
(3) Enabling this PDO will display the corresponding connection object in the main station register
(4) set PDO communication parameters
S: (2) PD310
TxPDO Data Object RxPDO Data Object PDO Info Obj Bind
Index SubIndex Bytes Name
1800 – 8 Transmit PDO 1 Parameter
2030 1 2 Drive status
2010 3 2 Running frequency
2200 1 2 INPUT 1
2200 2 2 INPUT 2
1801 – 8 Transmit PDO 2 Parameter
2200 3 2 INPUT 3
2200 4 2 INPUT 4
2200 5 2 INPUT 5
2200 6 2 INPUT 6
1802 – 8 Transmit PDO 3 Parameter
Click to check the PDO mapping objects

5 CANopen application example

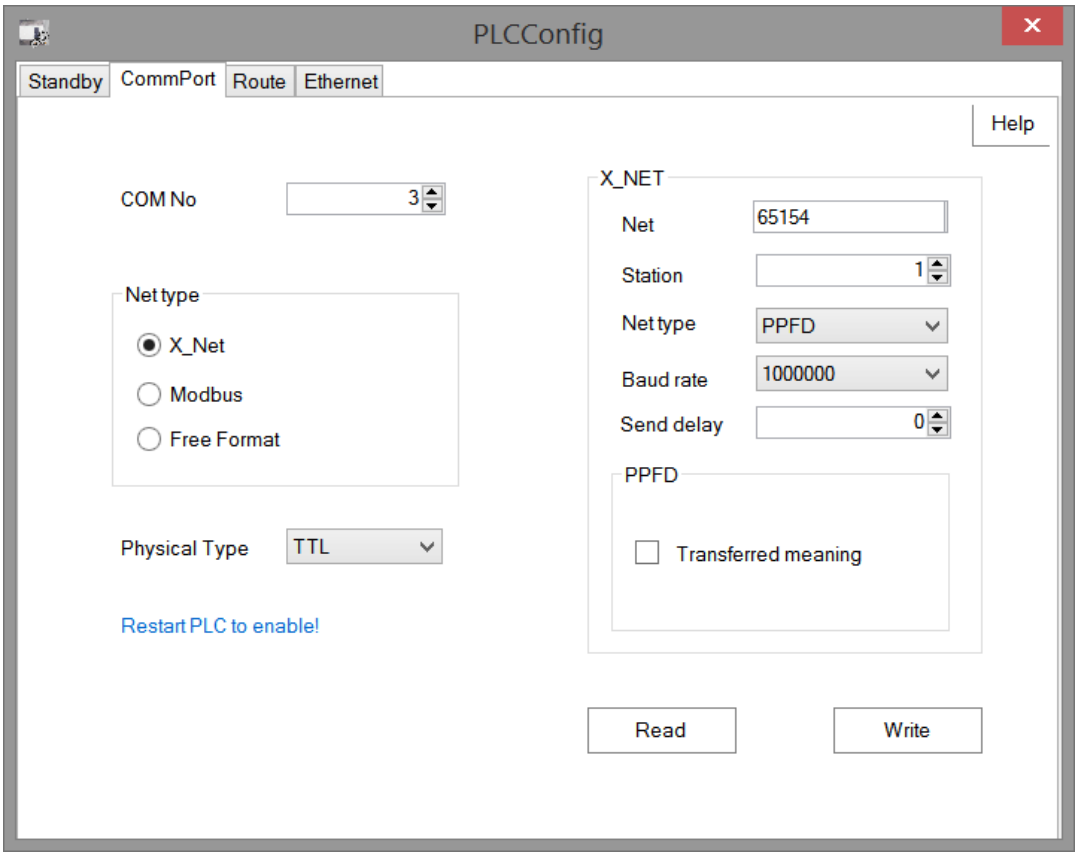
5.1 Two PMP20 series PLCs communicate through PMP-COBOX-ED

Case Description: We take two PMP20 PLCs for CANopen communication through the extended PMP-COBOX-ED as an example.

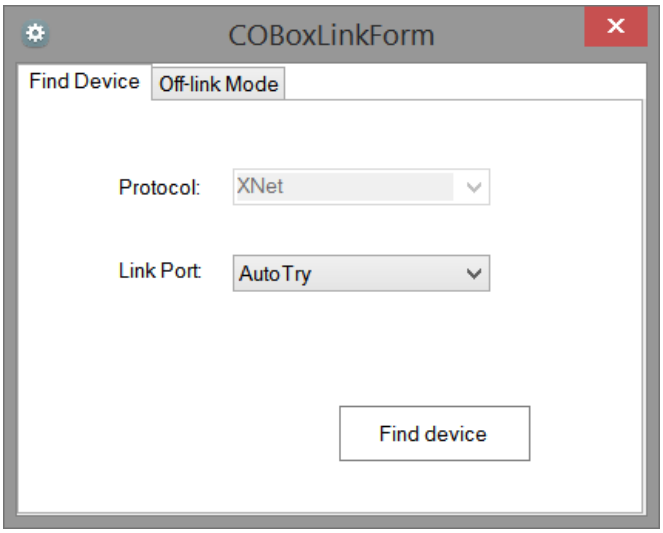
1. Write the values of the 10 registers D0~D9 of PLC A into D0~D9 of PLC B;
2. Read the values of the 10 registers D10~D19 of PLC B into D10-D19 of PLC A.

Step 1: Short circuit the CAN+ and CAN- of the two PMP-COBOX-EDs, turn the terminal resistance of the two PMP-COBOX-EDs to ON, and set the station number and baud rate of the master and slave stations through the dial code and knob on the PMP-COBOX-ED. Here we set PMP-COBOX-ED master station of No. A PLC, station No. 1, PMP-COBOX-ED slave station of No. B PLC, station No. 2, and baud rate of 1M.

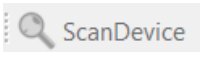
Step 2: Modify the COM3 parameters of two PMP5-32T-Es through the XNetConfig tool and write them in. After successful writing, power on again to take effect. The specific parameters are shown in the following figure:



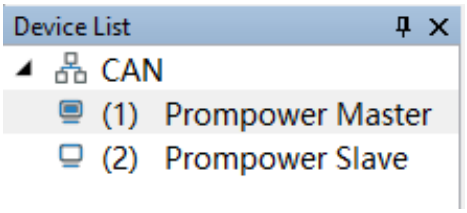
Step3: Use USB to connect master station PLC, click COBox, it will show below window:



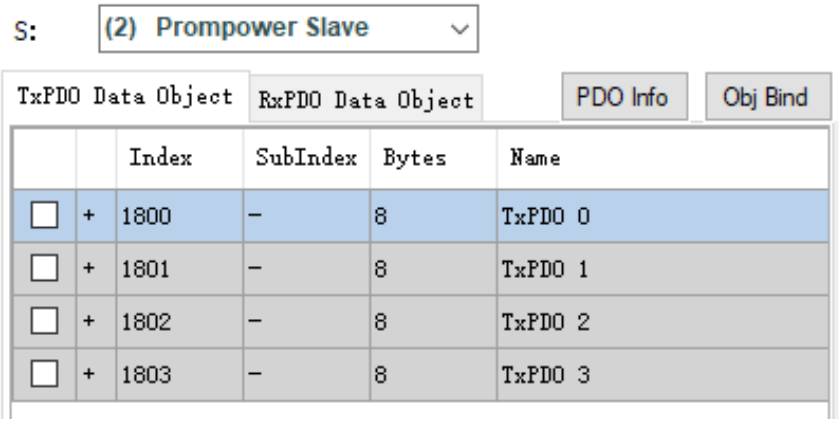
Click **[Find device]**, if the communication between PLC and PMP-COBOX-ED is normal, you can find PMP-COBOX-ED and enter the configuration Home screen for CANopen configuration. If the prompt "Find timeout" indicates abnormal communication between PLC and PMP-COBOX-ED, please check: ① whether the COM3 parameter of PLC is correct; ② Communication connection between PLC and PMP-COBOX-ED; ③ Is it search for a CANopen master station.

Enter CANopen configuration interface, click CANopen 

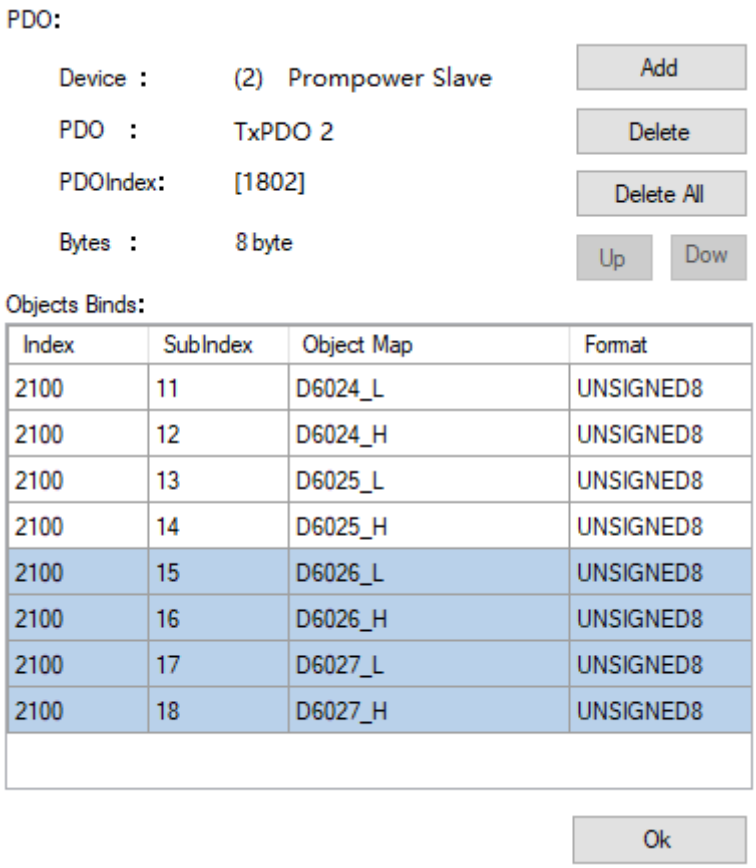
, waiting for the scan to end, the list of devices on the left will display all stations in the network. In this case, there are only two devices, so only the 1st master station and the 2nd slave station will be displayed, as shown in the following figure:



Step 4: Click on the slave station 2 to configure the TxPDO mapping object for the slave station. On the right side of the slave station configuration area, you can see four PDOs, as shown in the figure.

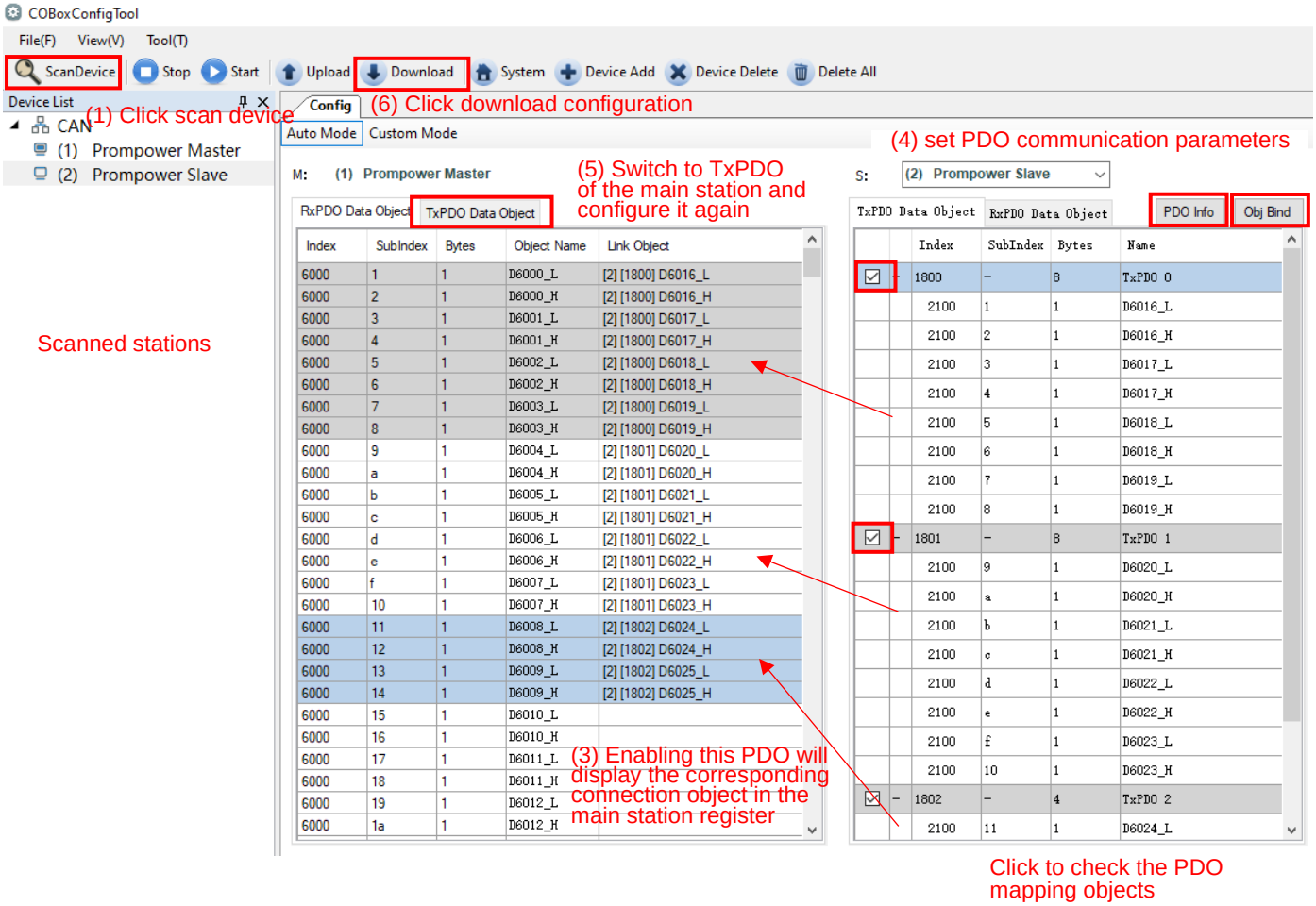


Click "+" to view the mapping objects for each PDO. The PMP-COBOX-ED slave station defaults to mapping 8 objects for each PDO. However, in this case, only 10 registers (20 objects) are needed. We can delete some mapping objects that are not needed. Select TxPDO2 and click Obj Bind to modify the mapping objects for PDO according to the requirements. Here, we delete the last 2 registers, as shown in the figure.

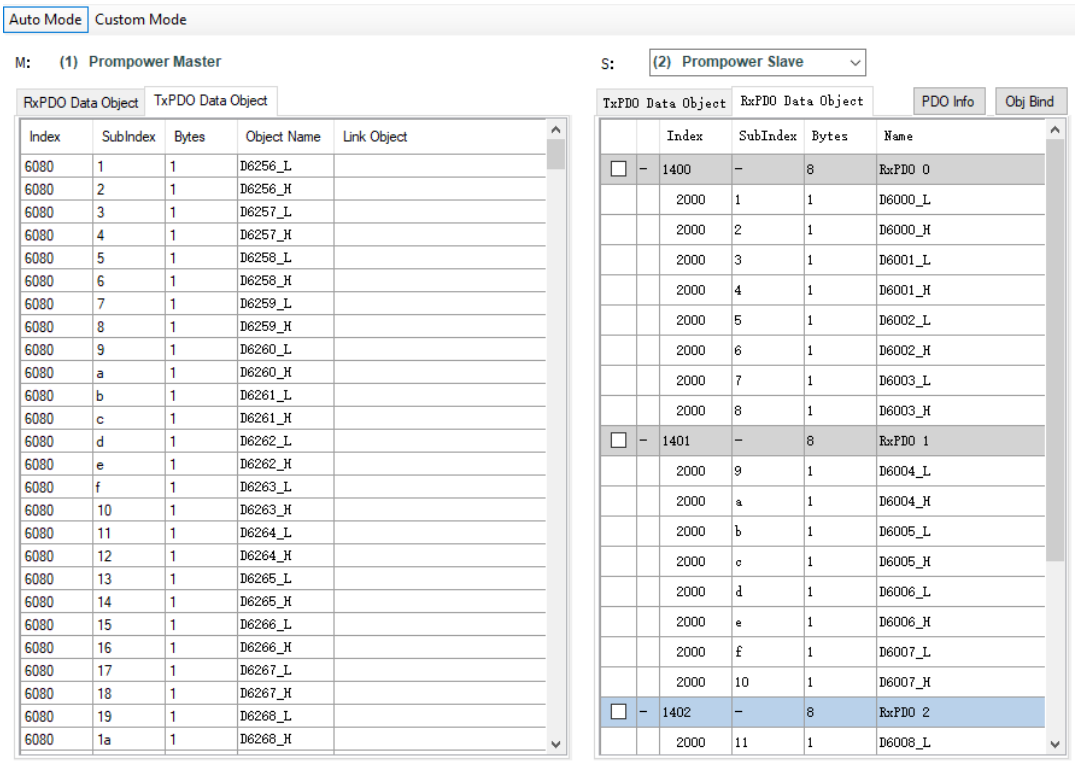


After configuring, click  to enable the PDO.

Here, we only need to enable the first three PDOs. In the master station configuration area, the objects of the slave station will automatically correspond to the master station register. You can also choose custom mode to select the first address of the corresponding register. Click on **[PDO Info]** to set communication parameters for the selected PDO according to your own needs. The specific configuration method is as follows:



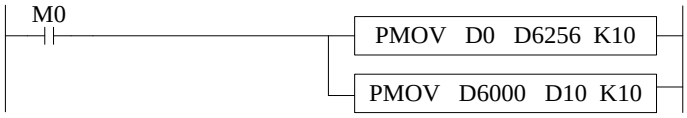
Click on the TxPDO data object of the master station, and the slave station will also switch to the RxPDO data object accordingly. The RxPDO configuration method of the slave station is the same as TxPDO, and the specific configuration is shown in the following figure:



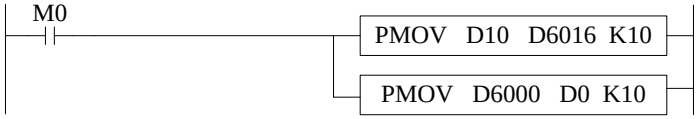
After completing the configuration, click 'Download'. After the download is completed, it will automatically run and the configuration will take effect immediately. At this point, the D6256-D6265 of the master station will be written in real-time to the D6000-D6009 of the slave station, and the D6016-D6025 of the slave station will be written in real-time to the D6000~D6009 of the master station.

Step 5: Write a program. In this case, the communication objects are D0-D9 and D10-D19, while the CANopen slave can only use D6000-D6031. Therefore, it is necessary to assign the values of D0 and D10 to D6000. The program is as follows:

Master station program:



Slave station program:



Appendix 1 CANopen related register

Master station occupation register (default D6000 as starting address, modifiable)

Input mapping area	Output mapping area	NMT control word	NMT status word	Online station no.	Node error info
D6000~D6255	D6256~D6511	D6512	D6513	D6514~D6517	D6518~D6645

NMT control word

Bit	D6512_H	D6512_L
Value	Network configuration station number 0x1-0x40 (64), valid for a single node 0xFF, all nodes are valid. Except for the above values, all other values are invalid.	2: Reset nodes 3: Reset communication 4: Pre operation 5: Operation 6: Stop

NMT status word

Register	Value	Status
Master station D6513	1	Initialising
	2	Application Reset
	3	Communication Reset
	4	Pre-Operatianal
	5	Operatianal
	6	Stopped

Online Station Number Register

Register	Corresponding online stations in the network							
	bit15	bit14	bit13	...	bit3	bit2	bit1	bit0
D6514	station16	station15	station14	...	station4	station3	station2	station1
D6515	station32	station31	station30	...	station20	station19	station18	station17
D6516	station48	station47	station46	...	station36	station35	station34	station33
D6517	station64	station63	station62	...	station52	station51	station50	station49

Node Error Information Register

Station 1 error message			Station 2 error message		
D6518_H	D6518_L	D6519	D6520_H	D6520_L	D6521
Error slave station	Error type	Error code	Error slave station	Error type	Error code
Station 3 error message			**Station 4 error message**		
D6522_H	D6522_L	D6523	D6524_H	D6524_L	D6525
Error slave station	Error type	Error code	Error slave station	Error type	Error code
……			……		
Station n error message			**Station 64 error message**		
(D6516+n*2)_H	(D6516+n*2)_L	D6517+n*2	D6644_H	D6644_L	D6645
Error slave station	Error type	Error code	Error slave station	Error type	Error code

Error type

Bit	Meaning
0	Common error
1	Current
2	Voltage
3	Temperature
4	Communication error (overflow, error status)
5	Device protocol specification
6	Reserved
7	Manufacturer specified

Error code

Error code (hex)	Meaning
0000	Error reset or no error
8100	Communication - General
8110	CAN overflow (object lost)
8120	CAN passive mode error
8130	Node protection or heartbeat error
8140	Restore from bus shutdown
8150	CAN-ID conflict
8210	PDO length error cannot be processed
8220	PDO over the length
8230	DAM Mpdo unable to process target object unavailable
8240	Unexpected SYNC data length

Slave occupation register (fixed D6000-D6032, cannot be modified)

Input mapping area	Output mapping area	NMT status word
D6000~D6015	D6016~D6031	D6032

NMT status word

Register	Value	Status
Slave station D6032	1	Initialising
	2	Application Reset
	3	Communication Reset
	4	Pre-Operatianal
	5	Operatianal
	6	Stopped

Appendix 2 Object Dictionary List

Object dictionary index	Sub index	Parameter name	R/W	Default value	Can be configured as PDO	Data type	○ master station/ △ slave station
1000	00	Device type	RO	F0191	NO	Unsigned32	○/△
1001	00	Error register	RO	0	NO	Unsigned8	○/△
1003	00	Sub index number	RO	0	NO	Unsigned32	○/△
	01~03	Predefined error fields			NO		
1005	00	Synchronous message COB-ID	R/W	0x80	NO	Unsigned32	○/△
1006	00	Synchronous cyclic cycle	R/W	0	NO	Unsigned32	○
1007	00	Synchronization window time	R/W	0	NO	Unsigned32	○/△
1008	00	Device name	R0		NO		○/△
1009	00	Hardware version	R0		NO		○/△
100A	00	Software version	R0		NO		○/△
100C	00	Node guard cycle	R/W	0	NO	Unsigned16	○/△
100D	00	Node guard factor	R/W	0	NO	Unsigned8	○/△
1014	00	Emergency COB-ID	R/W	0x80+ station no.	NO	Unsigned32	○/△
1016	00	Sub index numbers	R/W	64	NO	Unsigned8	○
	01~40	Heartbeat consumption time	R/W	0	NO	Unsigned32	○
1017	00	Heartbeat production time	R/W	0	NO	Unsigned32	△
1018	00	Sub index numbers	R0	4	NO	Unsigned8	○/△
	01	Manufacturer ID	R0	85134136	NO	Unsigned32	○/△
	02	Product code	R0	140401	NO	Unsigned32	○/△
	03	Modify code	R0	0	NO	Unsigned32	○/△
	04	Serial no.	R0	0	NO	Unsigned32	○/△
1400~ 1463	00	Sub index numbers	RO	5	NO	Unsigned8	○/△

Object diction-ary index	Sub index	Parameter name	R/W	Default value	Can be configured as PDO	Data type	○ master station/ △ slave station
	01	RxPDO's COB-ID	R/W	80000000+ station no.	NO	Unsigned32	○/△
	02	Sending type	R/W	255	NO	Unsigned8	○/△
1600~ 1663	00	Sub index numbers	R/W	0	NO	Unsigned8	○/△
	01~08	RxPDO's mapping object	R/W	00000000	NO	Unsigned32	○/△
1800~ 1863	00	Sub index numbers	RO	6	NO	Unsigned8	○/△
	01	TxPDO's COB-ID	R/W	80000000+ station no.	NO	Unsigned32	○/△
	02	Sending type	R/W	255	NO	Unsigned8	○/△
	03	Prohibited time	R/W	500	NO	Unsigned16	○/△
	05	Event time	R/W	0	NO	Unsigned16	○/△
	06	Sync Start Value	R/W	0	NO	Unsigned8	○/△
1A00~ 1A63	00	Sub index numbers	R/W	0	NO	Unsigned8	○/△
	01~08	TXPDO's mapping object	R/W	00000000	NO	Unsigned32	○/△
2000	00~20	Slave station input register	RO	0	RxPDO	Unsigned8	△
2100	00~20	Slave station output register	WO	0	TxPDO	Unsigned8	△
6000	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6010	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6020	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6030	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6040	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6050	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6060	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○

Object dictionary index	Sub index	Parameter name	R/W	Default value	Can be configured as PDO	Data type	○ master station/ △ slave station
6070	00~40	Master station input register	RO	0	RxPDO	Unsigned8	○
6080	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
6090	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
60A0	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
60B0	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
60C0	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
60D0	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
60E0	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
60F0	00~40	Master station output register	WO	0	TxPDO	Unsigned8	○
7010	00	RxPDO object starting address	R/W	6000	NO	Unsigned32	○
	01	RxPDO occupied min address	R/W	FFFFFFFF	NO	Unsigned32	○
	02	RxPDO occupied max address	R/W	FFFFFFFF	NO	Unsigned32	○
7011	00	TxPDO object starting address	R/W	6256	NO	Unsigned32	○
	01	TxPDO occupied min address	R/W	FFFFFFFF	NO	Unsigned32	○
	02	TxPDO occupied max address	R/W	FFFFFFFF	NO	Unsigned32	○
7012	00	Status information start address	R/W	6512	NO	Unsigned32	○
	01	Minimum address occupied by status information	R/W	6512	NO	Unsigned32	○
	02	Status information occupies the maximum address	R/W	6645	NO	Unsigned32	○

ВСЕ ДЛЯ АВТОМАТИЗАЦИИ:

Официальный дистрибьютор:

PROM POWER

www.prompower.ru

v.1