

## **TH270-HC-玻璃机械 CANOPEN 使用简易说明 1.1**

## 1.简介

## 2.节点和波特率设置

## 3.接线定义

## 4.底层数据结构

## 5.CANopen 通讯协议

### 5.1、CAL 协议

### 5.2、CAN 标识符分配表

### 5.3、服务数据对象SDO说明

### 5.4、过程数据对象PDO说明

### 5.5、SYNC 报文

### 5.6、Emergency 报文

### 5.7、HEARTBEAT 报文

### 5.8、网络管理（NMT）

## 6.对象字典

### 6.1、主要对象字典索引表

### 6.2 驱动器默认 PDO 配置表

## 7.应用案例

### 7.1、与永宏 PLC 连接

#### 7.1.1 硬件构架说明

#### 7.1.2 驱动器参数准备

#### 7.1.3 设计方案准备

#### 7.1.4 CBCAN master configurator 配置

#### 7.1.5 PLC 编程

\*分料

\*电子凸轮

### 7.2、与台达 PLC 连接

#### 7.2.1 硬件构架说明

#### 7.2.2 驱动器参数准备

#### 7.2.3 设计方案准备

#### 7.2.4 Canopen builder 配置

#### 7.2.5 PLC 编程

## 8.附件

### 8.1 永宏案例文件夹

### 8.2 台达案例文件夹

## 1、简介：

从OSI网络模型的角度来看，现场总线网络一般只实现了第1层（物理层）、第2层（数据链路层）、第7层（应用层）。因为现场总线通常只包括一个网段，因此不需要第3层（传输层）和第4层（网络层），也不需要第5层（会话层）第6层（描述层）的作用。

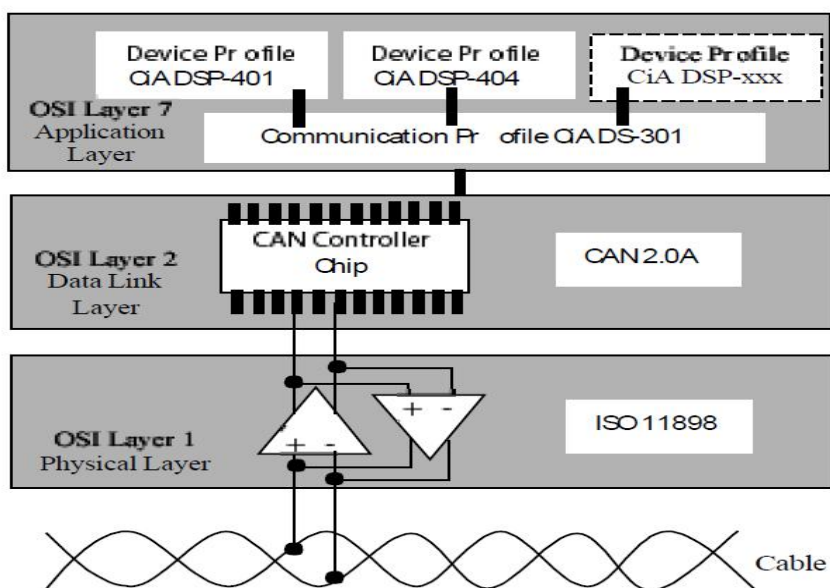
CAN（Controller Area Network）现场总线仅仅定义了第1层、第2层（见ISO11898标准）；实际设计中，这两层完全由硬件实现，设计人员无需再为此开发相关软件（Software）或固件（Firmware）。

同时，CAN只定义物理层和数据链路层，没有规定应用层，本身并不完整，需要一个高层协议来定义CAN报文中的11/29位标识符、8字节数据的使用。而且，基于CAN总线的工业自动化应用中，越来越需要一个开放的、标准化的高层协议：这个协议支持各种CAN厂商设备的互用性、互换性，能够实现在CAN网络中提供标准的、统一的系统通讯模式，提供设备功能描述方式，执行网络管理功能。

- 应用层（Application layer）：为网络中每一个有效设备都能够提供一组有用的服务与协议。
- 通讯描述（Communication profile）：提供配置设备、通讯数据的含义，定义数据通讯方式。
- 设备描述（Device profile）：为设备（类）增加符合规范的行为。

基于CAN的高层协议：CAL协议和基于CAL协议扩展的CANopen协议。CANopen协议是CAN-in-Automation(CiA)定义的标准之一，并且在发布后不久就获得了广泛的承认。尤其是在欧洲，CANopen协议被认为是在基于CAN的工业系统中占领导地位的标准。大多数重要的设备类型，例如数字和模拟的输入输出模块、驱动设备、操作设备、控制器、可编程控制器或编码器，都在称为“设备描述”的协议中进行描述；“设备描述”定义了不同类型的标准设备及其相应的功能。依靠CANopen协议的支持，可以对不同厂商的设备通过总线进行配置。

在OSI模型中，CAN标准、CANopen协议之间的关系如下图所示：



## 2、节点和波特率设置

节点和波特率设置参数表:

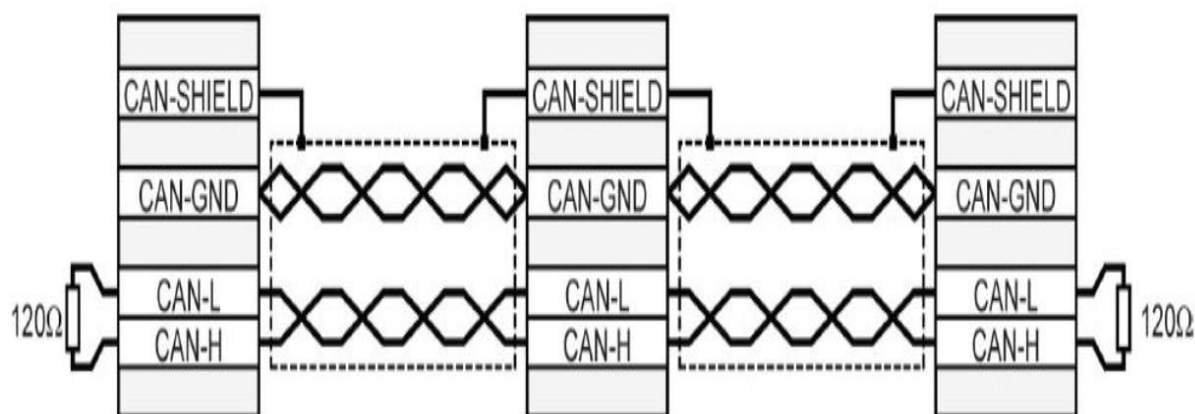
参数号	参数名	范围	说明	
PD31	CAN 通讯波特率选择	0~6	意义	最大总线长度 (米)
			0:20KBPS;	5000
			1:50KBPS;	1000
			2:100KBPS;	600
			3:125KBPS;	500
			4:250KBPS;	250
			5:500KBPS;	100
			6:1MBPS	25
PD32	CAN 通讯节点号选择	1~127		
PD33	CAN 通讯协议功能选择	0~1	0: 由上位机完成初始化 1: 驱动器自己完成初始化	

从理论上来说,最多可以有127个节点。不过,在实际应用中,最大节点数量取决于所用CAN收发器的性能。每个节点都是一个独立的CANopen设备,而每个CANopen设备又可包含最多8个逻辑设备(logical device),对于伺服驱动器来说,每个逻辑设备对应一个其所控制的轴。

### 3、驱动器CN3接线定义:

端子号	名称	功能
1	CAN_H	CAN通讯高
2	CAN_L	CAN通讯低
3	GND	信号地
4	GND	信号地
5		
6		
7	DATA+	485数据+
8	DATA-	485数据-

CAN总线线路必须用在CAN-L和CAN-H线之间每端连接的120欧姆(1%, 1/4W)电阻来终结,如下所示。



总线电缆请选用有两对双绞线的带屏蔽层电缆：一对双绞线分别接CAN-L 和CAN-H。

#### 4、CAN 底层通讯数据结构：

CAN 通过数据帧在主机（控制器）和总线节点之间传输数据。下图说明了数据帧的结构。

帧头	仲裁域		控制域	数据域	校验域	应答域	帧尾
	COB-ID (通讯对象标识符)	RTR (远程请求)					
1 位	11 或 29 位	1 位	6 位	0~8 字节	16 位	2 位	7 位

11 位（CAN2.0A）COBID 说明：

功能码				NODE ID（节点地址）						
10	9	8	7	6	5	4	3	2	1	0

#### 5、CANOPEN 通讯协议：

##### 5.1、CAL 协议

CAL（CAN Application Layer）协议是目前基于 CAN 的高层通讯协议中的一种，最早由 Philips 医疗设备部门制定。现在 CAL 由独立的 CAN 用户和制造商集团 CiA（CAN in Automation）协会负责管理、发展和推广。

5.2、CANOPEN 是在 CAL 基础上开发的，使用了 CAL 通讯和服务协议子集，提供了分布式控制系统的一种实现方案。CANopen 在保证网络节点互用性的同时允许节点功能随意扩展。CAN 标识符分配表：

通讯对象	功能码 COB-ID bit10~7 (2 进制)	COB-ID (16 进制)	相应通讯参数 在 OD 中的索引
NMT	0000	000h	—
SYNC	0001	080h	1005h、1006h、1007h
TIME STAMP	0010	100h	1012h、1013h
EMCY	0001	081h~0FFh	1024h、1015h
PDO1 (发送)	0011	181h~1FFh	1800h
PDO1 (接受)	0100	201h~27Fh	1400h
PDO2 (发送)	0101	281h~2FFh	1801h
PDO2 (接受)	0110	301h~37Fh	1401h
PDO3 (发送)	0111	381h~3FFh	1802h
PDO3 (接受)	1000	401h~47Fh	1402h
PDO4 (发送)	1001	481h~4FFh	1803h
PDO4 (接受)	1010	501h~57Fh	1403h
SDO (发送)	1011	581h~5FFh	1200h
SDO (接受)	1100	601h~67Fh	1200h
Heartbeat	1110	701h~77Fh	1016h、1017h

注意：

- 1、PDO/SDO 的发送/接受是由 (slave) CAN 节点观察的。
- 2、本驱动器的最大节点数为 127 时 CANopen 支持 4 个发送 PDO，4 个接收 PDO，当最大节点数为 63 时支持 6 个发送 PDO (TPDO5: 0x4C0+NODEID, TPDO5: 0x3C1+NODEID)，6 个接收 PDO (RPDO5: 0x440+NODEID, TPDO5: 0x340+NODEID)

CANopen 通讯模型定义了如下几种报文 (通讯对象)：

缩写	详称	说明
SDO	Service Data Object	用于非时间关键数据，比如参数。
PDO	Process Data Object	用于传输时间关键进程数据 (给定值、控制字、状态信息等)。
SYNC	Synchronization Message	用于同步 CAN 节点。
EMCY	Emergency Message	用于传输驱动器的报警事件。
NMT	Network Management	用于 CANopen 网络管理。
Heartbeat	Error Control Protocol	用于监测所有节点的生命状态。

### 5.3、服务数据对象SDO

SDO 用来访问一个设备的对象字典。访问者被称作客户(client)，对象字典被访问且提供所请求服务的CANopen 设

备被称作服务器(server)。客户的CAN 报文和服务器的应答CAN 报文总是包含8 字节数据 (尽管不是所有的数据字节都一

定有意义)。一个客户的请求一定有来自服务器的应答。

SDO 有2 种传送机制：

加速传送 (Expedited transfer)：最多传输4 字节数据

分段传送 (Segmented transfer) : 传输数据长度大于 4 字节

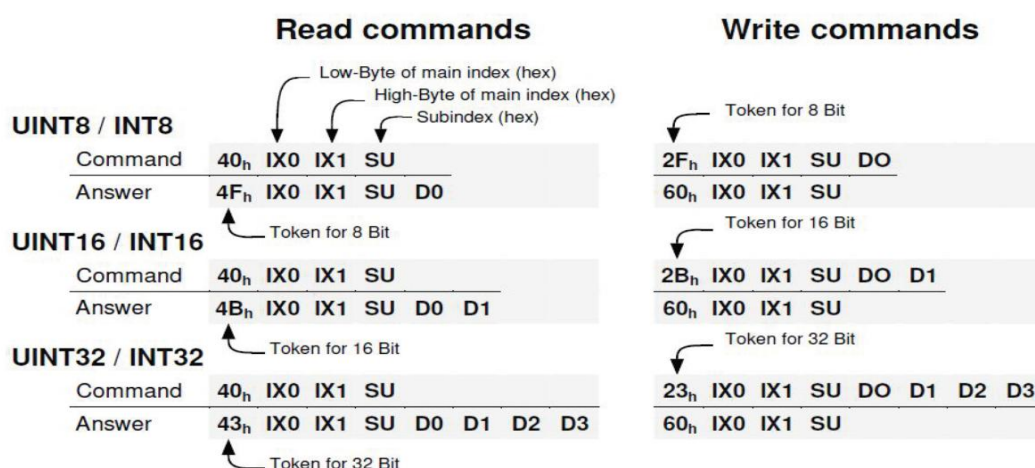
SDO 基本结构如下:

Byte0	Byte1~2	Byte3	Byte4~7
SDO 命令	对象索引	对象子索引	数据

SDO 命令说明:

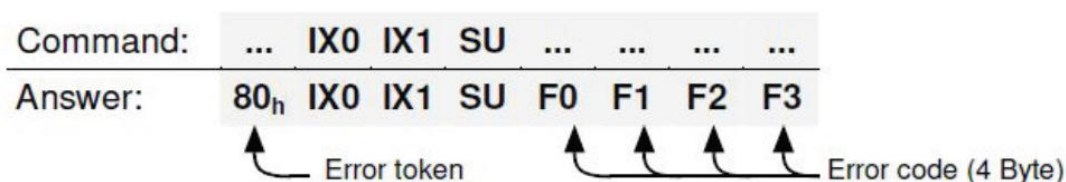
请求	请求命令说明	应答命令	应答说明
0x23	写 32 位索引	0x60(0x80)	正确 (错误)
0x2B	写 16 位索引	0x60(0x80)	正确 (错误)
0x2F	写 8 位索引	0x60(0x80)	正确 (错误)
0x40	读 32 位索引	0x43(0x80)	正确 (错误)
0x40	读 16 位索引	0x4B(0x80)	正确 (错误)
0x40	读 8 位索引	0x4F(0x80)	正确 (错误)

SDO 报文对参数读/写操作格式:



<b>UINT8 / INT8</b>	Reading of Obj. 6061_00 <sub>h</sub> Returning data: 01 <sub>h</sub>	Writing of Obj. 1401_02 <sub>h</sub> Data: EF <sub>h</sub>
Command:	40 <sub>h</sub> 61 <sub>h</sub> 60 <sub>h</sub> 00 <sub>h</sub>	2F <sub>h</sub> 01 <sub>h</sub> 14 <sub>h</sub> 02 <sub>h</sub> EF <sub>h</sub>
Answer:	4F <sub>h</sub> 61 <sub>h</sub> 60 <sub>h</sub> 00 <sub>h</sub> 01 <sub>h</sub>	60 <sub>h</sub> 01 <sub>h</sub> 14 <sub>h</sub> 02 <sub>h</sub>
<b>UINT16 / INT16</b>	Reading of Obj. 6041_00 <sub>h</sub> Returning data: 1234 <sub>h</sub>	Writing of Obj. 6040_00 <sub>h</sub> Data: 03E8 <sub>h</sub>
Command:	40 <sub>h</sub> 41 <sub>h</sub> 60 <sub>h</sub> 00 <sub>h</sub>	2B <sub>h</sub> 40 <sub>h</sub> 60 <sub>h</sub> 00 <sub>h</sub> E8 <sub>h</sub> 03 <sub>h</sub>
Answer:	4B <sub>h</sub> 41 <sub>h</sub> 60 <sub>h</sub> 00 <sub>h</sub> 34 <sub>h</sub> 12 <sub>h</sub>	60 <sub>h</sub> 40 <sub>h</sub> 60 <sub>h</sub> 00 <sub>h</sub>
<b>UINT32 / INT32</b>	Reading of Obj. 6093_01 <sub>h</sub> Returning data: 12345678 <sub>h</sub>	Writing of Obj. 6093_01 <sub>h</sub> Data: 12345678 <sub>h</sub>
Command:	40 <sub>h</sub> 93 <sub>h</sub> 60 <sub>h</sub> 01 <sub>h</sub>	23 <sub>h</sub> 93 <sub>h</sub> 60 <sub>h</sub> 01 <sub>h</sub> 78 <sub>h</sub> 56 <sub>h</sub> 34 <sub>h</sub> 12 <sub>h</sub>
Answer:	43 <sub>h</sub> 93 <sub>h</sub> 60 <sub>h</sub> 01 <sub>h</sub> 78 <sub>h</sub> 56 <sub>h</sub> 34 <sub>h</sub> 12 <sub>h</sub>	60 <sub>h</sub> 93 <sub>h</sub> 60 <sub>h</sub> 01 <sub>h</sub>

SDO 错误报文格式:





Error code F3 F2 F1 F0	Description
05 03 00 00 <sub>h</sub>	Toggle bit not alternated
05 04 00 01 <sub>h</sub>	Client / server command specifier not valid or unknown
06 01 00 00 <sub>h</sub>	Unsupported access to an object
06 01 00 01 <sub>h</sub>	Attempt to read a write only object
06 01 00 02 <sub>h</sub>	Attempt to write a read only object
06 02 00 00 <sub>h</sub>	Object does not exist in the object dictionary
06 04 00 41 <sub>h</sub>	Object cannot be mapped to the PDO
06 04 00 42 <sub>h</sub>	The number and length of the objects to be mapped would exceed PDO length
06 04 00 47 <sub>h</sub>	General internal incompatibility in the device
06 07 00 10 <sub>h</sub>	Data type does not match, length of service parameter does not match
06 07 00 12 <sub>h</sub>	Data type does not match, length of service parameter too high
06 07 00 13 <sub>h</sub>	Data type does not match, length of service parameter too low
06 09 00 11 <sub>h</sub>	Sub-index does not exist
06 04 00 43 <sub>h</sub>	General parameter incompatibility
06 06 00 00 <sub>h</sub>	Access failed due to an hardware error * <sup>1)</sup>
06 09 00 30 <sub>h</sub>	Value range of parameter exceeded
06 09 00 31 <sub>h</sub>	Value of parameter written too high
06 09 00 32 <sub>h</sub>	Value of parameter written too low
06 09 00 36 <sub>h</sub>	Maximum value is less than minimum value
08 00 00 20 <sub>h</sub>	Data cannot be transferred or stored to the application * <sup>1)</sup>
08 00 00 21 <sub>h</sub>	Data cannot be transferred or stored to the application because of local control
08 00 00 22 <sub>h</sub>	Data cannot be transferred or stored to the application because of the present device state * <sup>3)</sup>
08 00 00 23 <sub>h</sub>	No Object Dictionary is present * <sup>2)</sup>

#### 5.4、过程数据对象PDO

PDO 用来传输实时数据，数据从一个生产者传到一个或多个消费者。数据传送限制在1 到 8 个字节。PDO 通讯没有协议约束（意味着数据内容已预先定义），因此消费者可以在很短的时间内处理完接受到的数据。PDO 数据内容只由它的CAN ID 定义，假定生产者和消费者知道这个PDO 的数据内容。

每个PDO 在对象字典中用2 个对象描述：

**PDO 通讯参数：**包含将被PDO 使用的COB-ID，传输类型，禁止时间和定时器周期。

**PDO 映射参数：**包含一个对象字典中对象的列表，这些对象映射到PDO 里，包括它们的数据长度（in bits）。生产者和消费者必须知道这个映射，以解释PDO 内容。

PDO 消息的内容是预定义的（或者在网络启动时配置的），映射应用对象到PDO 中是在设备对象字典中描述的。如果设备（生产者和消费者）支持动态映射，那么使用SDO 报文可以配置PDO 映射参数，本驱动器支持动态PDO 映射。PDO

映射必须遵守以下2 个规则：

- 1)、每个PDO 最多可映射8 个对象；
- 2)、每个 PDO 的长度必须不超过 64 位。

PDO 映射流程：

- 1)、设置PDO 对应映射参数（如1600 h 或1A00 h）子索引0 的内容为0；
- 2)、修改PDO 对应映射参数（如1600 h 或1A00 h）子索引1~8 的内容，映射数据；
- 3)、设置PDO 对应映射参数（如1600 h 或1A00 h）子索引0 的内容为合法的数字（该PDO



映射的对象数)。

4)、PDO 映射完成。

PDO 可以有多种传送方式:

同步(通过接收SYNC 对象实现同步)周期: 传送在每1 到240 个SYNC 消息后触发异步由设备子协议中规定的对象特定事件触发传送。

### 5.5、SYNC 报文

同步对象用于控制数据在网络设备间的同步传输, 例如同步启动多个轴。同步报文的传输是基于生产者-消费者模型的, 所有支持同步PDO 的节点都可以作为消费者(同时)接收到此报文, 并使用该对象与其他节点进行同步。

一般应用方式为SYNC 主节点定时发送SYNC 对象, SYNC 从节点收到后同步执行任务。CANopen 建议用一个最高优先级的COB-ID 以保证同步信号正常传送。SYNC 报文可以不传送数据以使报文尽可能短。

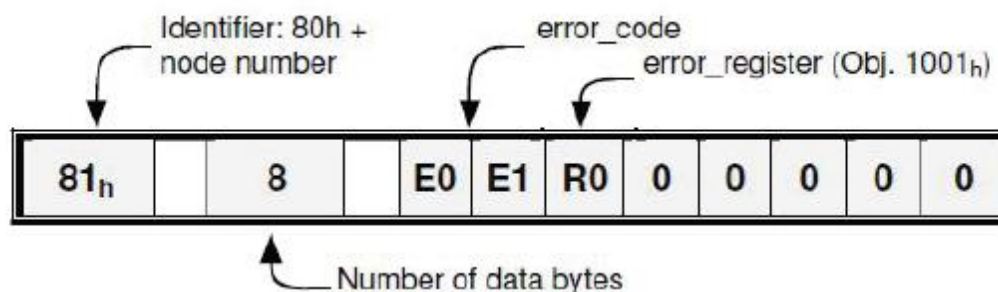
<b>Index</b>	1005 h
<b>Name</b>	cob_id_sync
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	NO
<b>Units</b>	—
<b>Value Range</b>	80000080 h, 00000080 h
<b>Default Value</b>	00000080 h

### 5.6、Emergency 报文

驱动器发生报警时, CANopen 会启动一条Emergency 报文, 告知消费者当前驱动器类型以及错误代码。在603Fh 对象中的低字节可以读到与面板显示一致的报警代码。

SYNC 报文的COB-ID 固定为080h, COB-ID 可以从对象字典的1005 h 读出。

Emergency 报文结构:



相关索引:

<b>Index</b>	1003 <sub>h</sub>
<b>Name</b>	pre_defined_error_field
<b>Object Code</b>	ARRAY
<b>No. of Elements</b>	4
<b>Data Type</b>	UINT32

<b>Sub-Index</b>	01 <sub>h</sub>
<b>Description</b>	standard_error_field_0
<b>Access</b>	RO
<b>PDO Mapping</b>	NO
<b>Units</b>	—
<b>Value Range</b>	—
<b>Default Value</b>	—

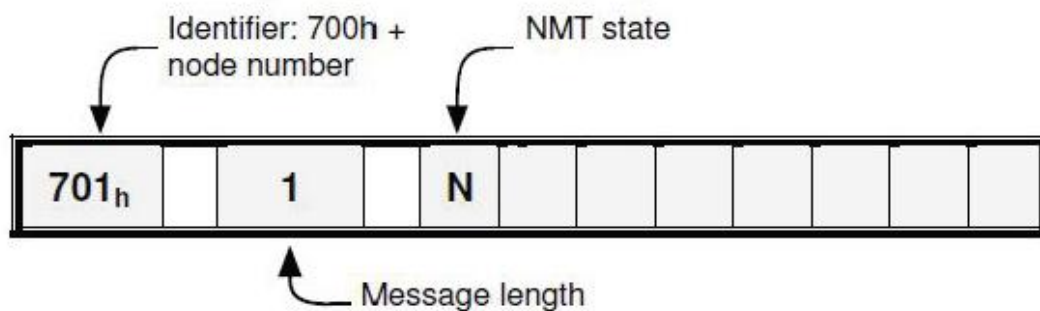
<b>Sub-Index</b>	02 <sub>h</sub>
<b>Description</b>	standard_error_field_1
<b>Access</b>	RO
<b>PDO Mapping</b>	NO
<b>Units</b>	—
<b>Value Range</b>	—
<b>Default Value</b>	—

<b>Sub-Index</b>	03 <sub>h</sub>
<b>Description</b>	standard_error_field_2
<b>Access</b>	RO
<b>PDO Mapping</b>	NO
<b>Units</b>	—
<b>Value Range</b>	—
<b>Default Value</b>	—

<b>Sub-Index</b>	04 <sub>h</sub>
<b>Description</b>	standard_error_field_3
<b>Access</b>	RO
<b>PDO Mapping</b>	NO
<b>Units</b>	—
<b>Value Range</b>	—
<b>Default Value</b>	—

## 5.7、HEARTBEAT 报文

报文结构:

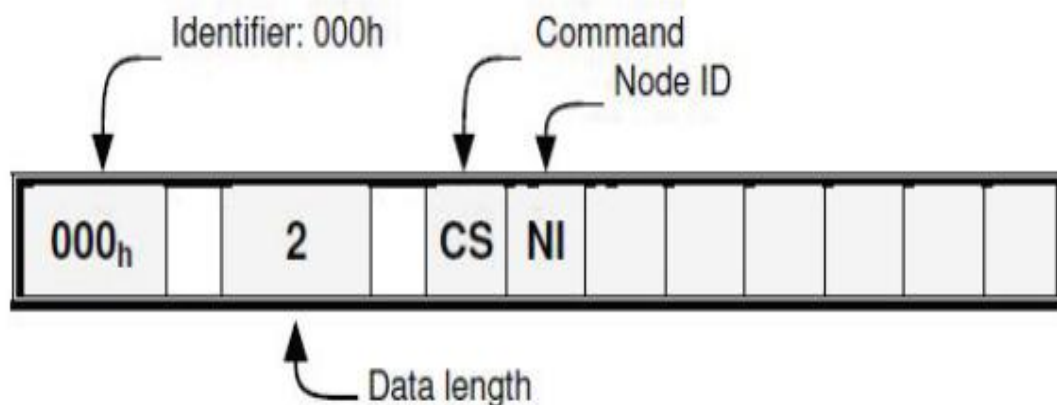


相关参数:

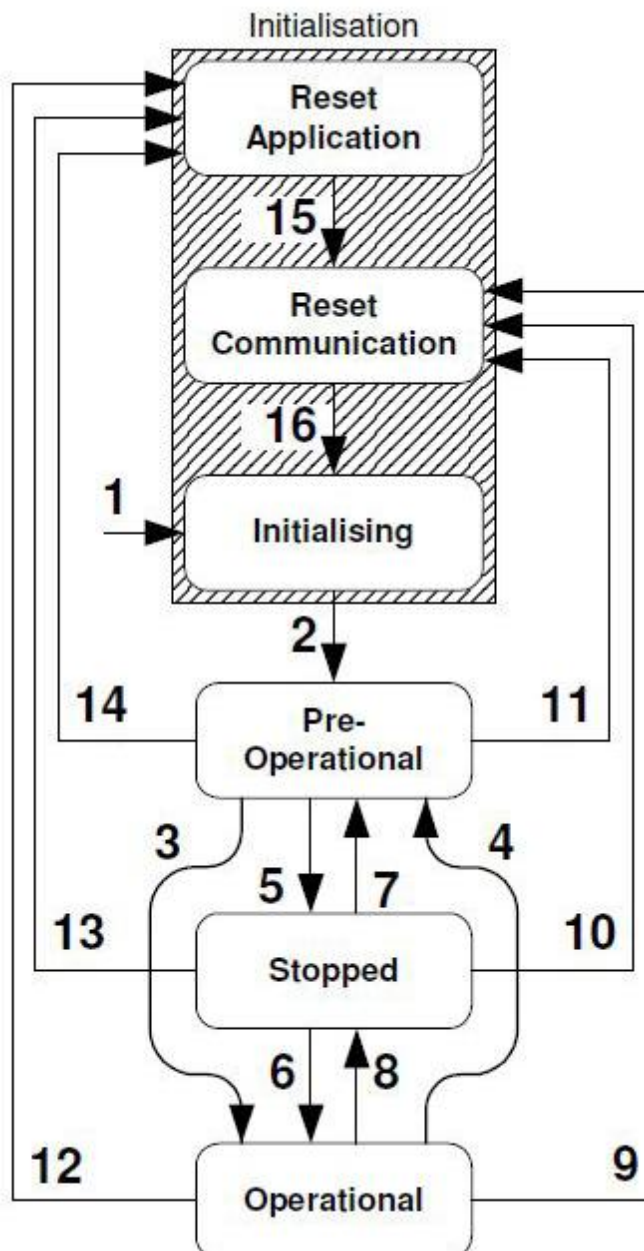
<b>Index</b>	1017 <sub>h</sub>
<b>Name</b>	producer_heartbeat_time
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	NO
<b>Units</b>	ms
<b>Value Range</b>	0 - 65535
<b>Default Value</b>	0

### 5.8、网络管理（NMT）：

报文结构:



网络管理状态转换图:



Name	Meaning	SDO	PDO	NMT
<b>Reset Application</b>	No communication. All CAN objects are set to their reset values (application parameter set).	-	-	-
<b>Reset Communication</b>	No communication. The CAN controller will be re-initialised.	-	-	-
<b>Initialising</b>	State after Hardware Reset. Reset of the CAN node, sending of the Bootup message	-	-	-
<b>Pre-Operational</b>	Communication via SDOs possible. PDOs inactive (No sending / receiving)	X	-	X
<b>Operational</b>	Communication via SDOs possible. PDOs active (sending / receiving)	X	X	X
<b>Stopped</b>	No communication except heartbeat + NMT	-	-	X

CS	Meaning	Transition	Target state
01 <sub>h</sub>	Start Remote Node	3, 6	Operational
02 <sub>h</sub>	Stop Remote Node	5, 8	Stopped
80 <sub>h</sub>	Enter Pre-Operational	4, 7	Pre-Operational
81 <sub>h</sub>	Reset Application	12, 13, 14	Reset Application
82 <sub>h</sub>	Reset Communication	9, 10, 11	Reset Communication

## 6、对象字典 OD

对象字典（OD: Object Dictionary）是一个有序的对象组；每个对象采用一个 16 位的索引值来寻址，为了允许访问数据结构中的单个元素，同时定义了一个 8 位的子索引，对象字典的结构参照表 3-1。不要被对象字典中索引值低于 0x0FFF 的‘data types’项所迷惑，它们仅仅是一些数据类型定义。一个节点的对象字典的有关范围在 0x1000 到 0x9FFF 之间。

CANopen 对象字典通用结构：

索引	对象
0000	Not used
0001 - 001F	静态数据类型（标准数据类型，如 Boolean, Integer 16）
0020 - 003F	复杂数据类型 (预定义由简单类型组合成的结构如 PDOCommPar, SDOPParameter)
0040 - 005F	制造商规定的复杂数据类型
0060 - 007F	设备子协议规定的静态数据类型
0080 - 009F	设备子协议规定的复杂数据类型
00A0 - 0FFF	Reserved
1000 - 1FFF	通讯子协议区域 (如设备类型, 错误寄存器, 支持的 PDO 数量)
2000 - 5FFF	制造商特定子协议区域
6000 - 9FFF	标准的设备子协议区域 (例如“DSP-401 I/O 模块设备子协议”: Read State 8 Input Lines 等)
A000 - FFFF	Reserved

CANopen 网络中每个节点都有一个对象字典。对象字典包含了描述这个设备和它的网络行为的所有参数。

一个节点的对象字典是在电子数据文档（EDS: Electronic Data Sheet）中描述或者记录在纸上。不必要也不需要通过 CAN-bus“审问”一个节点的对象字典中的所有参数。如果一个节点严格按照在纸上的对象字典进行描述其行为，也是可以的。节点本身只需要能够提供对象字典中必需的对象（而在 CANopen 规定中必需的项实际上是很少的），以及其它可选择的、构成节点部分可配置功能的对象。

### 6.1、主要对象字典索引表：

索引	子索引	属性	功能	索引	子索引	属性	功能
3000	0	RW	PA0	3080	0	RW	PB0
3001	0	RW	PA1	3081	0	RW	PB1
3002	0	RO	PA2	3082	0	RO	PB2
3003	0	RW	PA3	3083	0	RW	PB3
3004	0	RW	PA4	3084	0	RW	PB4

3005	0	RW	PA5	3085	0	RW	PB5
3006	0	RW	PA6	3086	0	RW	PB6
3007	0	RW	PA7	3087	0	RW	PB7
3008	0	RW	PA8	3088	0	RW	PB8
3009	0	RW	PA9	3089	0	RW	PB9
300A	0	RW	PA10	308A	0	RW	PB10
300B	0	RW	PA11	308B	0	RW	PB11
300C	0	RW	PA12	308C	0	RW	PB12
300D	0	RW	PA13	308D	0	RW	PB13
300E	0	RW	PA14	308E	0	RW	PB14
300F	0	RW	PA15	308F	0	RW	PB15
3010	0	RW	PA16	3090	0	RW	PB16
3011	0	RW	PA17	3091	0	RW	PB17
3012	0	RW	PA18	3092	0	RW	PB18
3013	0	RW	PA19	3093	0	RW	PB19
3014	0	RW	PA20	3094	0	RW	PB20
3015	0	RW	PA21	3095	0	RW	PB21
3016	0	RW	PA22	3096	0	RW	PB22
3017	0	RW	PA23	3097	0	RW	PB23
3018	0	RW	PA24	3098	0	RW	PB24
3019	0	RW	PA25	3099	0	RW	PB25
301A	0	RW	PA26	309A	0	RW	PB26
301B	0	RW	PA27	309B	0	RW	PB27
301C	0	RW	PA28	309C	0	RW	PB28
301D	0	RW	PA29	309D	0	RW	PB29
301E	0	RW	PA30	309E	0	RW	PB30
301F	0	RW	PA31	309F	0	RW	PB31
3020	0	RW	PA32	30A0	0	RW	PB32
3021	0	RW	PA33	30A1	0	RW	PB33
3022	0	RW	PA34	30A2	0	RW	PB34
3023	0	RW	PA35	30A3	0	RW	PB35
3024	0	RW	PA36	30A4	0	RW	PB36
3025	0	RW	PA37	30A5	0	RW	PB37
3026	0	RW	PA38	30A6	0	RW	PB38
3027	0	RW	PA39	30A7	0	RW	PB39
3028	0	RW	PA40	30A8	0	RW	PB40
3029	0	RW	PA41	30A9	0	RW	PB41
302A	0	RW	PA42	30AA	0	RW	PB42
302B	0	RW	PA43	30AB	0	RW	PB43
302C	0	RW	PA44	30AC	0	RW	PB44
302D	0	RW	PA45	30AD	0	RW	PB45
302E	0	RW	PA46	30AE	0	RW	PB46
302F	0	RW	PA47	30AF	0	RW	PB47



3030	0	RW	PA48	30B0	0	RW	PB48
3031	0	RW	PA49	30B1	0	RW	PB49
3032	0	RW	PA50	30B2	0	RW	PB50
3033	0	RW	PA51	30B3	0	RW	PB51
3034	0	RW	PA52	30B4	0	RW	PB52
3035	0	RW	PA53	30B5	0	RW	PB53
3036	0	RW	PA54	30B6	0	RW	PB54
3037	0	RW	PA55	30B7	0	RW	PB55
3038	0	RW	PA56	30B8	0	RW	PB56
3039	0	RW	PA57	30B9	0	RW	PB57
303A	0	RW	PA58	30BA	0	RW	PB58
303B	0	RW	PA59	30BB	0	RW	PB59
303C	0	RW	PA60	30BC	0	RW	PB60
303D	0	RW	PA61	30BD	0	RW	PB61
303E	0	RW	PA62	30BE	0	RW	PB62
303F	0	RW	PA63	30BF	0	RW	PB63
3040	0	RW	PA64	30C0	0	RW	PB64
3041	0	RW	PA65	30C1	0	RW	PB65
3042	0	RW	PA66	30C2	0	RW	PB66
3043	0	RW	PA67	30C3	0	RW	PB67
3044	0	RW	PA68	30C4	0	RW	PB68
3045	0	RW	PA69	30C5	0	RW	PB69
3046	0	RW	PA70	30C6	0	RW	PB70
3047	0	RW	PA71	30C7	0	RW	PB71
3048	0	RW	PA72	30C8	0	RW	PB72
3049	0	RW	PA73	30C9	0	RW	PB73
304A	0	RW	PA74	30CA	0	RW	PB74
304B	0	RW	PA75	30CB	0	RW	PB75
304C	0	RW	PA76	30CC	0	RW	PB76
304D	0	RW	PA77	30CD	0	RW	PB77
304E	0	RW	PA78	30CE	0	RW	PB78
304F	0	RW	PA79	30CF	0	RW	PB79
3050	0	RW	PA80	30D0	0	RW	PB80
3051	0	RW	PA81	30D1	0	RW	PB81
3052	0	RW	PA82	30D2	0	RW	PB82
3053	0	RW	PA83	30D3	0	RW	PB83
3054	0	RW	PA84	30D4	0	RW	PB84
3055	0	RW	PA85	30D5	0	RW	PB85
3056	0	RW	PA86	30D6	0	RW	PB86
3057	0	RW	PA87	30D7	0	RW	PB87
3058	0	RW	PA88	30D8	0	RW	PB88
3059	0	RW	PA89	30D9	0	RW	PB89
305A	0	RW	PA90	30DA	0	RW	PB90

305B	0	RW	PA91	30DB	0	RW	PB91
305C	0	RW	PA92	30DC	0	RW	PB92
305D	0	RW	PA93	30DD	0	RW	PB93
305E	0	RW	PA94	30DE	0	RW	PB94
305F	0	RW	PA95	30DF	0	RW	PB95
3060	0	RW	PA96	30E0	0	RW	PB96
3061	0	RW	PA97	30E1	0	RW	PB97
3062	0	RW	PA98	30E2	0	RW	PB98
3063	0	RW	PA99	30E3	0	RW	PB99
3064	0	RW	PA100	30E4	0	RW	PB100
3065	0	RW	PA101	30E5	0	RW	PB101
3066	0	RW	PA102	30E6	0	RW	PB102
3067	0	RW	PA103	30E7	0	RW	PB103
3068	0	RW	PA104	30E8	0	RW	PB104
3069	0	RW	PA105	30E9	0	RW	PB105
306A	0	RW	PA106	30EA	0	RW	PB106
306B	0	RW	PA107	30EB	0	RW	PB107
306C	0	RW	PA108	30EC	0	RW	PB108
306D	0	RW	PA109	30ED	0	RW	PB109
306E	0	RW	PA110	30EE	0	RW	PB110
306F	0	RW	PA111	30EF	0	RW	PB111
3070	0	RW	PA112	30F0	0	RW	PB112
3071	0	RW	PA113	30F1	0	RW	PB113
3072	0	RW	PA114	30F2	0	RW	PB114
3073	0	RW	PA115	30F3	0	RW	PB115
3074	0	RW	PA116	30F4	0	RW	PB116
3075	0	RW	PA117	30F5	0	RW	PB117
3076	0	RW	PA118	30F6	0	RW	PB118
3077	0	RW	PA119	30F7	0	RW	PB119
3078	0	RW	PA120	30F8	0	RW	PB120
3079	0	RW	PA121	30F9	0	RW	PB121
307A	0	RW	PA122	30FA	0	RW	PB122
307B	0	RW	PA123	30FB	0	RW	PB123
307C	0	RW	PA124	30FC	0	RW	PB124
307D	0	RW	PA125	30FD	0	RW	PB125
307E	0	RW	PA126	30FE	0	RW	PB126
307F	0	RW	PA127	30FF	0	RW	PB127
603f	0	RO	Error_code	6071	0	RW	Tq_target_torque
6040	0	RW	controlword	6077	0	RW	Tg_torque_actual_value
6041	0	RO	Stateword	607a	0	RW	Target_positio

							n
6042	0	RO	Vl_Target_velocity	60fd	0	RW	IO_input_state
6044	0	RO	VL_torgue_effort	60fe	1	RO	IO_output_set
6064	0	RO	Pc_position_actual_value	60ff	0	RW	Target_velocity
606C	0	RO	Pv_velocity_actual_value				

索引	子索引	属性	功能	索引	子索引	属性	功能
31b2	0	RW	PJ00	31C2	0	RW	PJ16
31B3	0	RW	PJ01	31C3	0	RW	PJ17
31B4	0	RW	PJ02	31C4	0	RW	PJ18
31B5	0	RW	PJ03	31C5	0	RW	PJ19
31B6	0	RW	PJ04	31C6	0	RW	PJ20
31B7	0	RW	PJ05	31C7	0	RW	PJ21
31B8	0	RW	PJ06	31C8	0	RW	PJ22
31B9	0	RW	PJ07	31C9	0	RW	PJ23
31BA	0	RW	PJ08	31CA	0	RW	PJ24
31BB	0	RW	PJ09	31CB	0	RW	PJ25
31BC	0	RW	PJ10	31CC	0	RW	PJ26
31BD	0	RW	PJ11	31CD	0	RW	PJ27
31BE	0	RW	PJ12	31CE	0	RW	PJ28
31BF	0	RW	PJ13	31CF	0	RW	PJ29
31C0	0	RW	PJ14	31D0	0	RW	PJ30
31C1	0	RW	PJ15				

## 6.2 驱动器默认 PDO 配置表:

注 1: 默认配置根据需求也能进行修改, 或者通过变更 EDS 文件的方法也能修改默认配置。

注 2: 下表只是表示出厂缺省配置的情况。

index	COB-ID	R/T	BYTE SIZE	TYPE	description	INDEX	SUB-INDEX	description
1400	200+NODE_ID	RX	2	1	RXPDO1	6040	0	controlword
1401	300+NODE_ID	RX	6	255	RXPDO2	6040	0	controlword
						60FD	0	IO_input_state
1402	400+NODE_ID	RX	6	255	RXPDO3	6040	0	controlword
						607A	0	Target_position
1403	500+NODE_ID	RX	6	255	RXPDO4	6040	0	controlword

						60FF	0	Pv_Target_velocity
1404	540+NODE_ID	RX	6	255	RXPDO5	6040	0	controlword
						6071	0	Target_torque
1405	440+NODE_ID	RX	6	255	RXPDO6	6040	0	controlword
						6042	0	Vl_Target_velocity
1800	180+NODE_ID	TX	6	255	TXPDO1	6041	0	Stateword
						603F	0	Error_code
1801	280+NODE_ID	TX	6	255	TXPDO2	6041	0	Stateword
						60FE	1	IO_output_set
1803	380+NODE_ID	TX	6	255	TXPDO3	6041	0	Stateword
						6064	1	Pc_position_actual_value
1804	480+NODE_ID	TX	6	2	TXPDO4	6041	0	Stateword
						606C	0	Pv_velocity_actual_value
1805	4C0+NODE_ID	TX	4	2	TXPDO5	6041	0	Stateword
						6077	0	Tg_torque_actual_value
1806	3C0+NODE_ID	TX	4	2	TXPDO6	6041	0	Stateword
						6044	0	VL_torgue_effort

## 7.应用案例

### 7.1、与永宏 PLC 连接

#### 7.1.1 硬件构架说明

略

#### 7.1.2 驱动器参数准备

PD31=6 波特率=1MHZ

PD32=2 驱动器节点 NODE-ID=2

PD33=0 驱动器上位机完成初始化

#### 7.1.3 设计方案准备

例如：采用 CANopen 对 IN 端子的信号传送，实现内定位和电子凸轮的控制。

规划：PLC 的 CAN 模块作为主站，节点=1

驱动器作为从站，节点=2

驱动器的使能 SON、故障 RES 采用 PDO 进行通讯

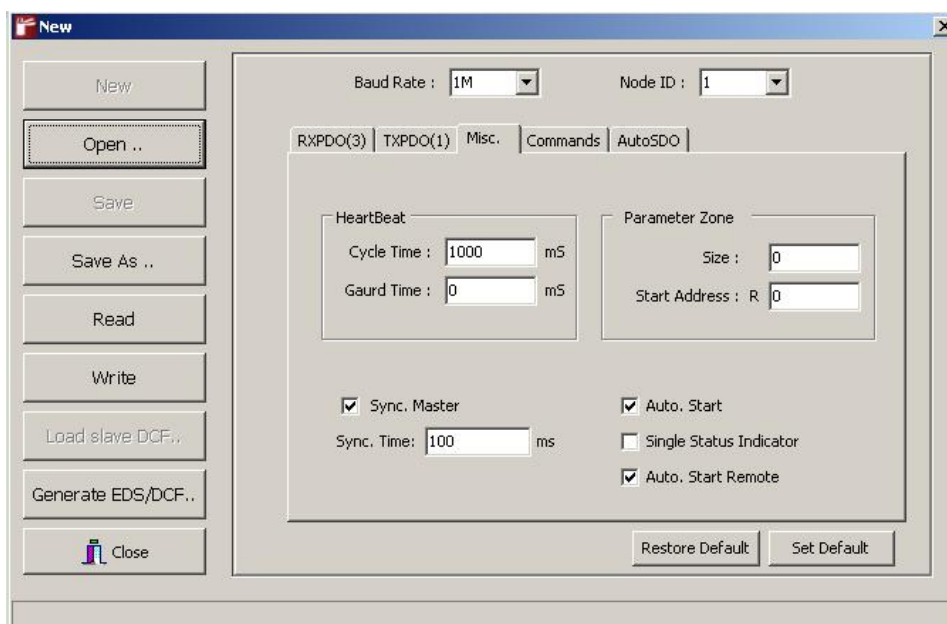
驱动器的故障代码 ERR\_CODE、输出端子 IO\_OUTPUT\_SET 通过 PDO 进

行通讯

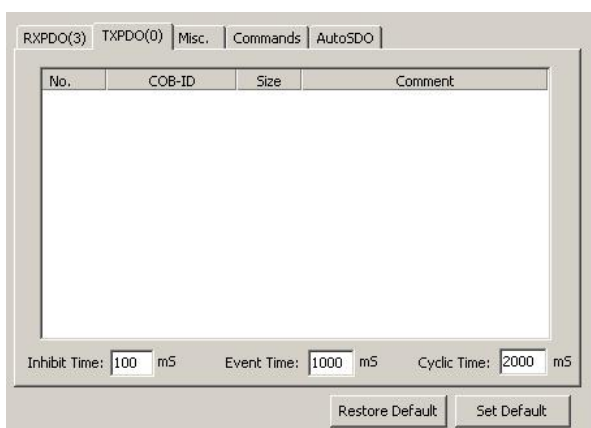
驱动器的输入端子 IO\_INPUT\_STATE 通过 SDO 进行发送。

#### 7.1.4 CBCAN master configurator 配置

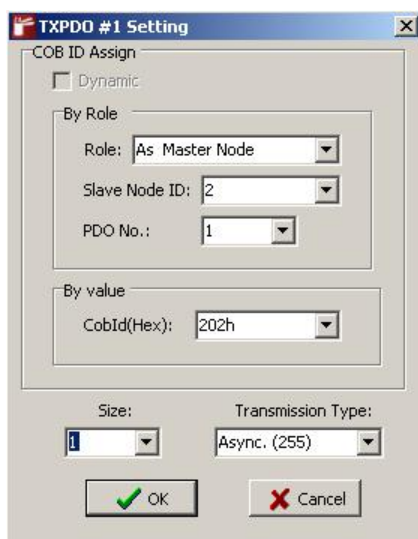
##### (1) MISC 设定



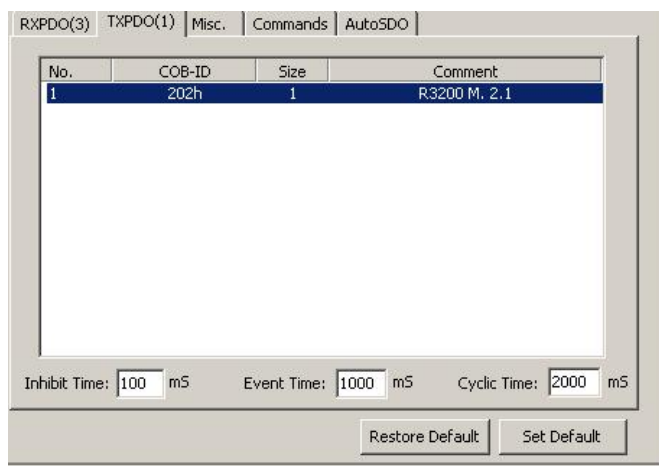
波特率=1M ,主站 NODE-ID=1 心跳 HEARTBEAT=1S ,同步信号=100MS  
 (2) TXPDO 配置 (注: 这个 TXPDO 表示为 PLC 的 CAN 接口模块的 TXPDO)



请添加 ADD:



主站, 从节点=2, PDO1, COB-ID=202H, 长度=1WORD, 类型=255



这样作为主站 PLC\_CAN 模块的 TXPDO1 与作为从站 KT270 驱动器的 RXPDO1 就做好了连接关系。实际上驱动器的 CONTROLWORD 与永宏 PLC 的 R3200 就做了映射关系。

index	COB-ID	R/T	BYTE SIZE	TYPE	description	INDEX	SUB-INDEX	description
1400	200+NODE_ID	RX	2	1	RXPDO1	6040	0	controlword

驱动器中 CONTROLWORD 的控制字中

Bit0=son

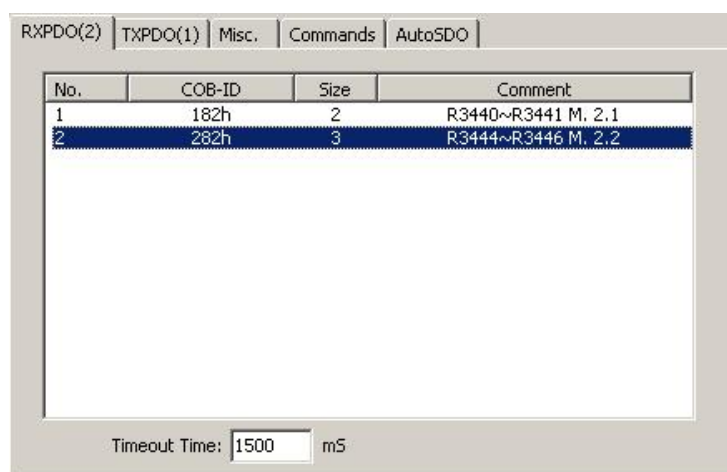
Bit7=res

这样，PLC 程序中只要对 R3200 送 0001H 就可以自动点亮 SON,让伺服使能。

对 R3200 送 0000H 就可以自动关闭 SON,让伺服使能关。

这样，PLC 程序中只要对 R3200 送 0080H 就可以自动点亮 RES,让伺服复位。

(3) RXPDO 的配置 (注: 这个 RXPDO 表示为 PLC 的 CAN 接口模块的 RXPDO) 类似 TXPDO 一样添加 COB-ID 的设置,使其达到下图样子。



1800	180+NODE_ID	TX	6	255	TXPDO1	6041	0	Stateword
						603F	0	Error_code
1801	280+NODE_ID	TX	6	255	TXPDO2	6041	0	Stateword
						60FE	1	IO_output_set



Stateword:

Error\_code: FFXH

高位 FFH 固定不变  
低位 XXH 报警代码

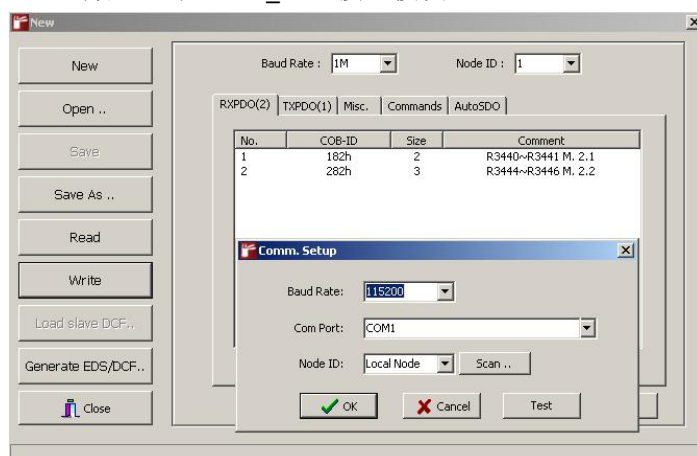
IO\_output\_set:

Bit7~Bit0=OU7~OUT0

Bit31~Bit8 系统保留用

这样：PLC 的故障代码 Error\_code 在 R3441 中，将该值-0FF00H=故障代码值  
PLC 的输出端口 IO\_output\_set 在 R3445 中。

#### (4) 将配置写入 PLC\_CAN 接口模块



按 OK 即可完成配置。

#### (5) SDO 报文构成

实现对驱动器端口 60FD 的控制。首先要将 PB117=255 这样，控制端口就与外部物理端口分开了，改为收通讯控制。

60FD: IO\_INPUT-state

Bit15~Bit0=IN15~IN0

Bit31~Bit16 系统保留

所以如果要发 IN0=1 发送 23 FD 60 00 01 00 00 00 的报文即可。

(注：数据是低字节在前，高字节在后)

#### 7.1.5 PLC 编程

\*分料

见附件

\*电子凸轮

见附件

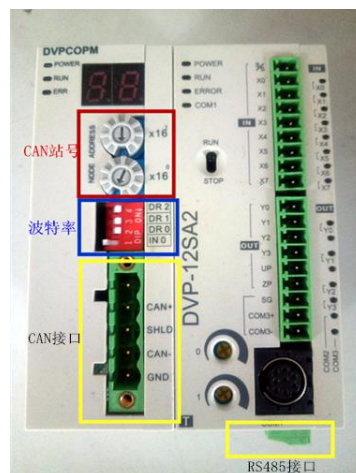
## 7.2 与台达 PLC 连接

以台达 SA2 系列 PLC 为例。

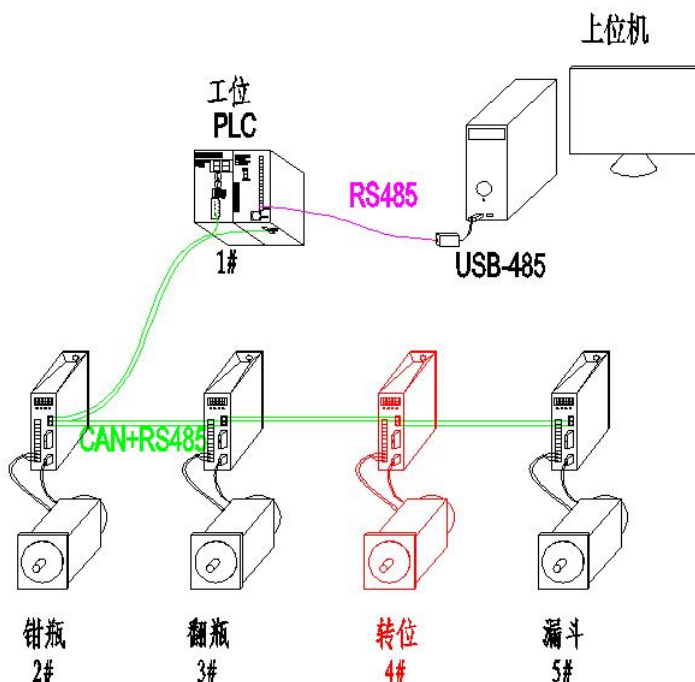
\*主机采用 DVP-12SA2

\*CAN 接口模块采用 DVP-COPM-SL

\*本案例按照转位机进行描述。



### 7.2.1 硬件构架说明



\*上位机将用户参数通过 RS485 与 SA2 的 COM3 通讯。

\*驱动器内定位参数通过 RS485 与 SA2 的 COM2 通讯（通讯类型选用 PLC-LINK 方式）。

PLC 做主机 站号 1

驱动器作为从机 站号 4

波特率设定为 115200 8 位数据位 1 停止位 无校验位

\*驱动器内定位命令通过 CANOPEN 与 SA2 的 CAN 接口模块通讯。

PLC 做主机，站号 1

驱动器做从机 站号 4

波特率设定为 1M

### 7.2.2 驱动器通讯参数准备

采用 CANOPEN 与 485 双通讯的相关参数见下表：

RS485 MODBUS RTU 参数						
序号	参数名	功能	设定范围	出厂设定值	应用设定值	本案应用需要
1	PA51	驱动器通讯	0-31	1	根据机位而定	4

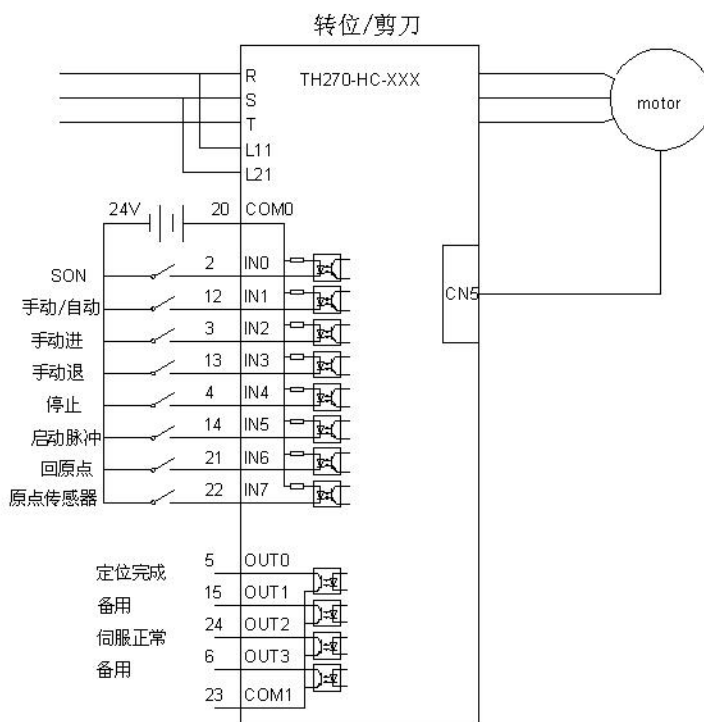
序号	参数号	参数名	范围	说明	本案应用需要	
2	PA52	通讯速率	0-5	1	0:无通讯 1: 9600 2: 19200 3: 38400 4: 57600 5: 115200	5
3	PC3	通讯格式	0-15	0	bit0 检验位 0:无 1:有 bit1 奇偶 0:ODD 1:EVEN bit2 停止位 0:1 位 1:2 位 bit3 屏蔽通讯报警 0 允许报警 1 屏蔽报警	0
CAN 通讯参数						
1	PD31	CAN 通讯波特率选择	0~6	意义	最大总线长度 (米)	6
				0:20KBPS;	5000	
				1:50KBPS;	1000	
				2:100KBPS;	600	
				3:125KBPS;	500	
				4:250KBPS;	250	
				5:500KBPS;	100	
				6:1MBPS	25	
2	PD32	CAN 通讯节点号选择	1~127			4
3	PD33	CAN 通讯协议功能选择	0~1	0: 由上位机完成初始化 1: 驱动器自己完成初始化		1

### 7.2.3 设计方案准备

转位控制采用驱动器内定位控制运动，CANopen 负责命令发出，采用 PDO 收发方式，485 负责参数传送。原点信号采用外传感器信号。

CAN 通讯对驱动器输入端口进行软控制。

内定位端口硬件规划：



## 内定位相关参数设定

PB 参数	功能	出厂值	设定值	说明
PB1	信号来源	0	2	0: P 标准脉冲控制方式 1: PR 寄存器信号 2: PP 程序 (自编程) 信号
PB2	坐标系	0	0	0: 绝对坐标系 1: 相对坐标系
PB3	内定位电子齿轮比分子	1	1	
PB4	内定位电子齿轮比分母	1	1	
PB35	原点产生方式	1	16	BIT0: 0:找 Z 脉冲 1: 不找 Z 脉冲 BIT2 BIT1: 00:接近开关 01: 正极限 10: 负极限 11: 使能位置 Bit4: 0:标准 1: 连续带线性升降速
PB36	回原点速度 (快速)	500	1000	
PB37	回原点速度 (慢速)	100	400	找 Z 脉冲时速度
PB38	原点后自动偏移高位	0	10	PB35 bit4=1 时, 回原点产生的自动偏移
PB39	原点后自动偏移低位	0	0	
PB45	输入和输出自定义功能	0	7	0: 标准 7: I/O 功能自定义 其余值意义详见说明书
PB46	IN0 信号定义	1	1	0: 标准 IN 1: SON

PB47	IN1 信号定义	2	39	0: 标准 IN 2: RES 39: 手自动 无效时: 采用 PB1 指定模式 有效时: 切换至 PD30 指定模式
PB48	IN2 信号定义	3	37	0: 标准 IN 3: LSP 37: 手动进
PB49	IN3 信号定义	4	38	0: 标准 IN 4: LSN 38: 手动退
PB50	IN4 信号定义	5	0	0: 标准 IN 5: CLE
PB51	IN5 信号定义	6	0	0: 标准 IN 6: INH
PB52	IN6 信号定义	7	0	0: 标准 IN 7: TL+
PB53	IN7 信号定义	8	27	0: 标准 IN 8: TL- 27: 原点信号器
PB56	OUT0 信号定义	2	0	0: 标准 OUT 2: RD
PB57	OUT1 信号定义	1	0	0: 标准 OUT 1: ALM
PB58	OUT2 信号定义	3	1	1: ALM 3: INP
PB59	OUT3 信号定义	5	0	0: 标准 OUT 5: CDO
PB60	IN0 有效模式	1	1	0: 断开有效 1. 接通有效 2. 接通瞬间有效 3. 断开瞬间有效
PB61	IN1 有效模式	1	1	
PB62	IN2 有效模式	0	1	
PB63	IN3 有效模式	0	1	
PB64	IN4 有效模式	1	2	
PB65	IN5 有效模式	1	2	
PB66	IN6 有效模式	1	2	
PB67	IN7 有效模式	1	1	
PB70	模数位置最大值 (高位)	9999	0	增量定位时, 选择 0-4095 之间。 绝对定位时, 选择 99999999~ -99999999
PB71	模数位置最大值 (低位)	9999	4095	
PB72	模数位置最小值 (高位)	-9999	0	
PB73	模数位置最小值 (低位)	-9999	0	
PB101	应用场合	0	4	0: 标准应用 4: 玻璃机械专用

PB117	IN 端口控制来源	0	127	IN0~IN7 分别有 BIT0~BIT7 来控制 0: 物理端口 1: 网络端口 485 与 CAN 都网络口
PA123	DP-RES	0	7	7-PP 版本号 9-PP 指令指针
PD20	加速度类型	0	0	0: 线性加减速 1: S 型加减速
PD30	IN 切换工作模式	0	3	IN 端口指定=39, 并电平有效时, 触发模式切换。 0: P 脉冲方式 1: PR 寄存器定位方式 2: PP 程序定位方式 3: 手动方式
PA15	电机方向	0		

配合 PP 程序所用寄存器 (所需根据 PP 程序而定)

PB 参数	功能	出厂值	地址	说明
PJ4	增量速度	0	0x7D4	
PJ5	加速时间	0	0x7D5	
PJ6	减速时间	0	0x7D6	
PJ7	增量距离 (圈内数)	0	0x7D7	增量运动设置<=4096
PJ8	增量距离 (圈数)	0	0x7D8	

#### 7.2.4 Canopen builder 配置

\*台达 CAN 接口 DVPCOPM-SL 模块在首次做总线连接时需要进行项目下装。


当 CAN 的通讯线 CAN+, CAN-, CAN\_GND 连接可靠后, 就可以用 Canopen builder 软件进行配置设定 (见右图)。



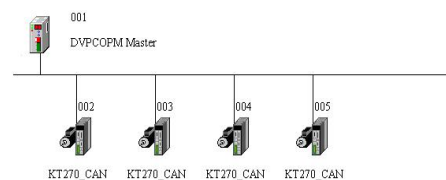
**注意:** 1.在使用配置软件时, 台达 PLC 的编辑软件不能处于监视模式。

2.最后一台驱动器必须要加装 120 欧终端电阻 (见右图)。否则模块会产生 F4 报警。



\*打开 Canopen builder 后, 点击  在线按钮。

系统会读取在线的所有 CAN 装置, 并将阵列显示在总线上。(如右图)




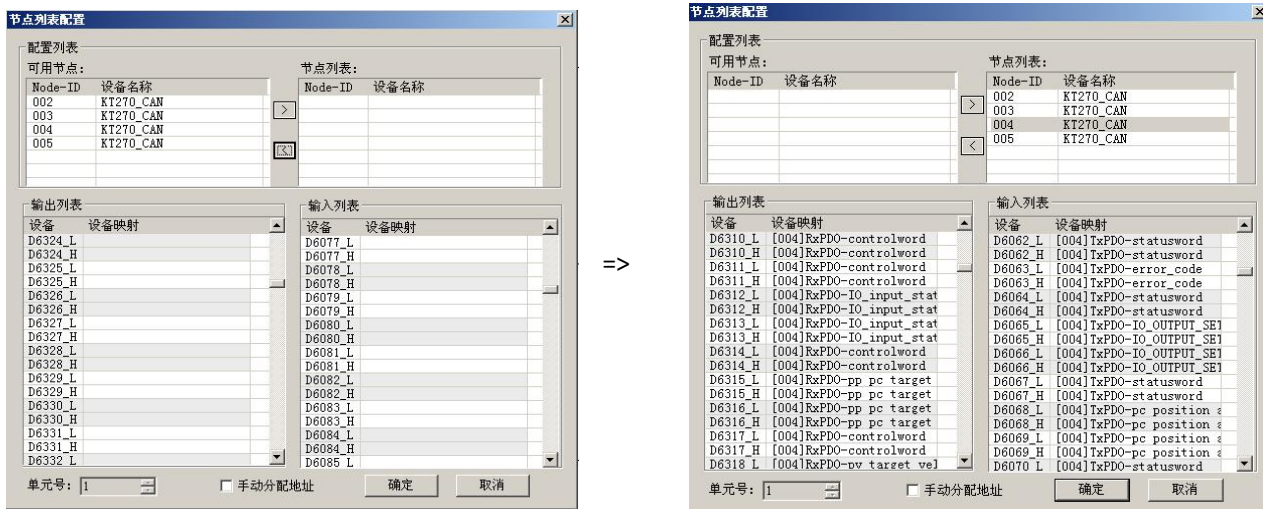
**注意:** 如果系统读取时显示为无法识别的设备时, 说明 Canopen builder 软件内缺少 TH270 驱动器的识别文件。TH270 驱动器的识别文件为 TH270.EDS。本文件可向惠斯通科技有限公司免费申请。



\*双击 | 主站图标, 将可用节点添加至节点列表。



将左侧可用节点中的全部添加  到节点列表中。



当添加完成后，点击 NODE-ID=4 的节点后，在下方会显示站号 4 的 PDO 映射表。我们将要用到 IO\_INPUT\_SET 和 IO\_OUTPUT\_SET。

D6312_L	[004]RxPDO-IO_input_stat	D6064_L	[004]TxPDO-statusword
D6312_H	[004]RxPDO-IO_input_stat	D6064_H	[004]TxPDO-statusword
D6313_L	[004]RxPDO-IO_input_stat	D6065_L	[004]TxPDO-IO_OUTPUT_SE1

驱动器 接收 PDO	功能	PLC 内映射地址	读/写
IO_INPUT_STATE	驱动器 IN 端口	D6312	只写

驱动器 发送 PDO	功能	PLC 内映射地址	读/写
IO_ONPUT_SET	驱动器 OUT 端口	D6065	只读

\*点击  下载当前配置，此时 DVPCOPM-SL 模块即完成了当前 CAN 节点及映射配置。

### 7.2.5 PP 程序编制

PP 程序是驱动器内部做的预规划，根据 IN 端口信号做出的定位动作数据块。可以减少 PLC 程序处理的篇幅。

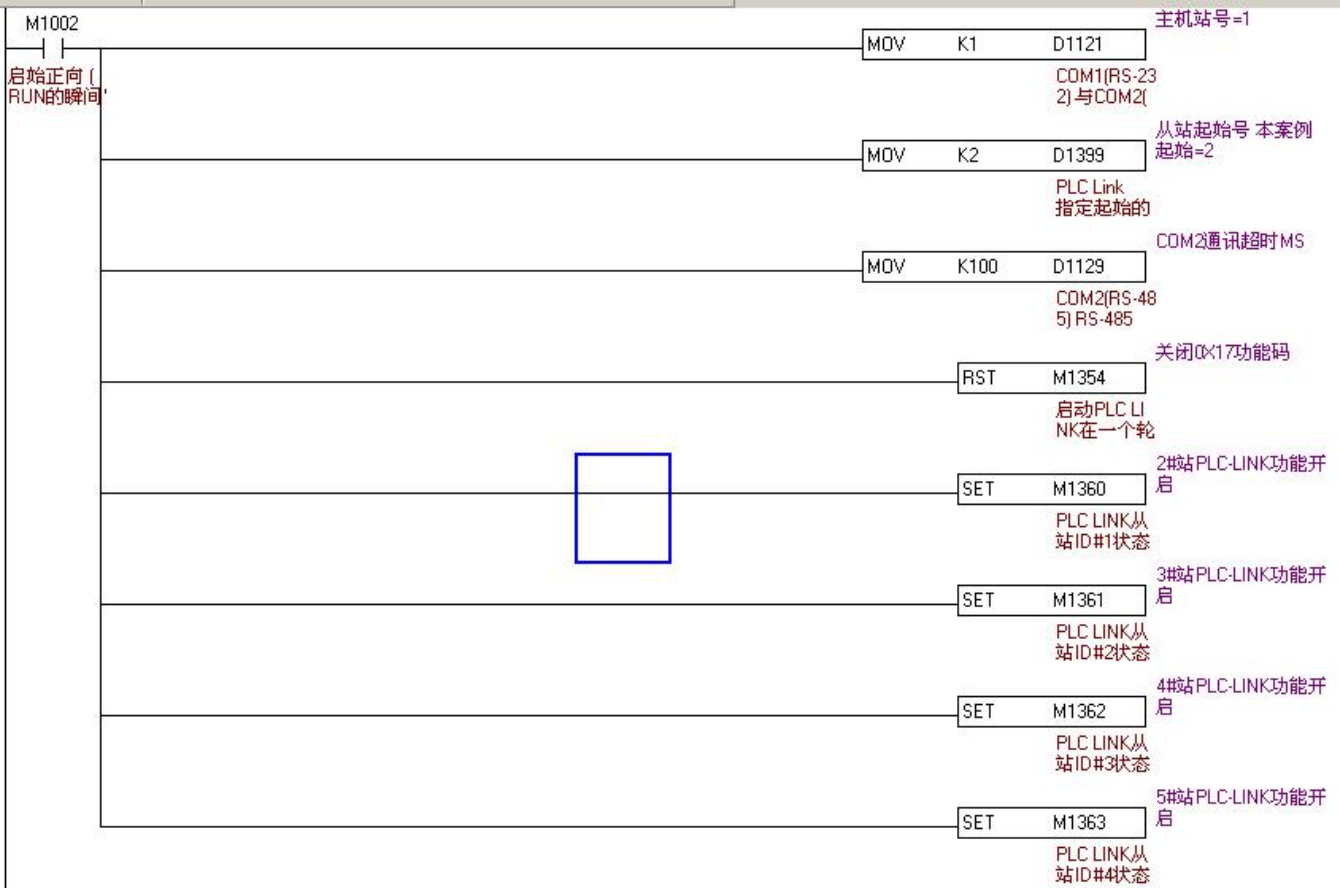
- 本案例的大概思路：设置 IN4 是停止命令  
 IN5 是启动脉冲  
 IN6 回原点命令  
 IN7 原点传感器输入

标号	指令	操作数	注释
	EI	0	开中断
	CODI	0	相对坐标系统
	INT6	7	IN6 回原点中断
	INT4	5	IN4 停止中断
KEY	LDB_I	6	

	JNT	KEY10	IF IN6=0 THEN KEY10
	RET_CO	0	回原点
	SETB_M	0	
KEY1	LDB_M	0	
	JNT	KEY	
	LDB_I	6	
	JT	KEY1	等待命令释放
	SETB_0	0	完成信号=1
	WAT_T	2	完成信号持续 20MS
	RSTB_0	0	
KEY10	LDB_I	5	
	JNT	KEY	IF IN5=0 THEN KEY
	SPD	PJ4	增量偏移速度
	ACC_TIME	PJ5	加速时间
	DEC_TIME	PJ6	减速时间
	SETB_M	0	
	POS_H	PJ8	偏移距离
	POS_L	PJ7	偏移距离
KEY11	LDB_M	0	
	JNT	KEY	
	LDB_I	5	
	JT	KEY11	
	SETB_0	0	
	WAT_T	2	
	RSTB_0	0	
	JUMP	KEY	
INT6	CANCEL	0	IN6 回原点信号输入后中断原运行命令
	RSTB_M	0	
	RETI	0	
INT4	CANCEL	0	IN4 停止信号输入后中断原运行命令
	RSTB_M	0	
	RETI	0	
	END	0	

### 7.2.6 PLC 编程

#### 1. COM2: RS485 RTU 通讯格式设置

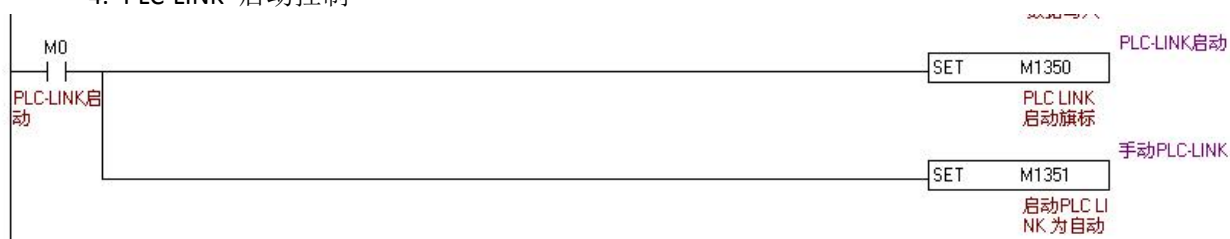


## 2. PLC-LINK 配置

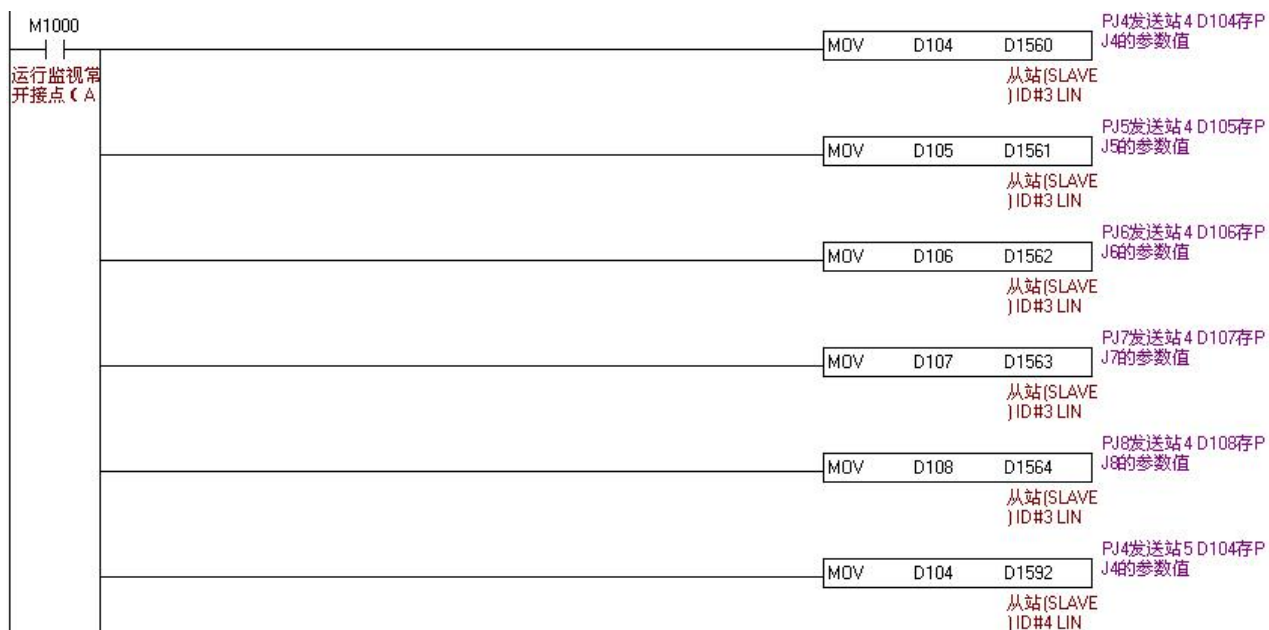
## 3. PJ4~PJ8 参数地址与 PLC-LINK 地址关系配置



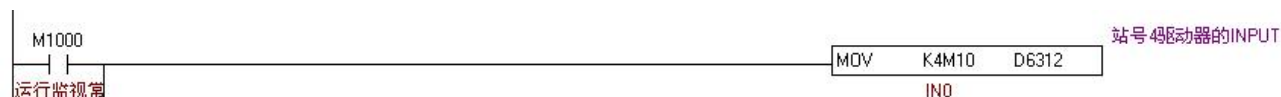
#### 4. PLC-LINK 启动控制



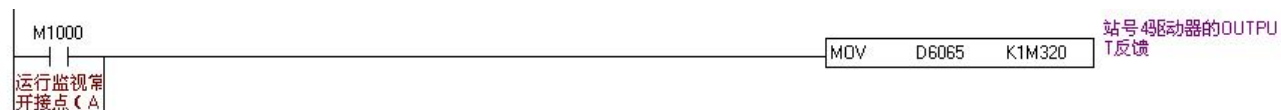
5. 用户设置 PJ4~PJ8 存放于 D104~D108，系统将 D104~D108 的数据通过 PLC-LINK 发送寄存器传送至驱动器。



6. 设置 M10~M17 的 8 位内部继电器作为驱动器 INPUT 端口的控制映射。然后通过 CAN 总线发出。



7. 设置 M320~M323 的 4 位内部继电器作为驱动器 OUTPUT 端口控制映射。然后通过 CAN 总线读取。



8. 增量控制工序 (略)