

# CS 系列伺服驱动器 CANopen使用手册



# 目录

<b>1 · 概述</b>	4
1-1、关于驱动器	5
1-2、关于本手册	5
1-3、参考协议	5
1-4、功能总述	6
<b>2 · 接线与设置</b>	7
2-1、通讯端口信号定义	8
2-2、通讯网络拓扑结构	8
2-3、适配线缆	9
2-4、参数设置	9
<b>3 · 通讯网络配置</b>	10
3-1、设备通讯	11
3-1-1、对象字典	12
3-1-2、通讯对象	12
3-2、网络管理对象 NMT	13
3-2-1、节点控制	- 14 -
3-2-2、错误控制	- 16 -
3-2-3、Boot-up	- 16 -
3-3、服务数据对象 SDO	- 17 -
3-4、过程数据对象 PDO	- 19 -
3-5、同步帧对象 SYNC	- 20 -
3-6、紧急帧对象 EMCY	- 21 -
<b>4 · 设备控制</b>	- 22 -
4-1、设备状态机	- 23 -
4-2、控制字	- 24 -
4-3、状态字	- 26 -
4-4、停机方式	- 27 -
<b>5 · 单位转换</b>	- 28 -
5-1、驱动器内部单位	- 29 -
5-2、用户单位	- 29 -
5-3、单位转换方法	- 30 -
5-4、指令极性	- 31 -
<b>6 · 操作模式</b>	- 32 -
6-1、设置和切换操作模式	- 33 -
6-2、Profile Velocity (PV)模式	- 33 -
6-2-1、操作步骤	- 34 -
6-2-2、可选操作	- 34 -
6-2-3、相关对象	- 34 -
6-3、Profile Position (PP)模式	- 35 -

6-3-1、操作步骤.....	- 35 -
6-3-2、可选操作.....	- 35 -
6-3-3、相关对象.....	- 36 -
6-4、Homing (HM)模式.....	- 37 -
6-4-1、操作步骤.....	- 37 -
6-4-2、相关对象.....	- 38 -
6-5、Interpolated Position (IP) 模式.....	- 38 -
6-5-1、操作步骤.....	- 39 -
6-5-2、相关对象.....	- 39 -
6-6、限位处理流程.....	- 40 -
<b>7 · 字典对象.....</b>	<b>- 41 -</b>
7-1、通信子协议对象.....	- 42 -
7-2、厂家自定义对象.....	- 67 -
7-2、标准设备子协议对象.....	- 71 -
<b>8 · 故障及处理.....</b>	<b>- 95 -</b>
8-1、获取故障信息.....	- 96 -
2、通过状态字.....	- 96 -
8-2、故障处理.....	- 96 -
2、CANopen 相关故障.....	- 97 -
<b>9 · 应用示例—基于 Beckhoff 控制器.....</b>	<b>- 99 -</b>
9-1、基本配置.....	- 100 -
9-2、伺服驱动器设置.....	- 100 -
9-3-1、控制器和 PC 通讯.....	- 100 -
9-3-2、控制器和伺服驱动器通讯.....	- 104 -
9-3-3、SDO 配置.....	- 106 -
9-3-4、变量连接.....	- 108 -

# 1 · 概述

1.1 关于驱动器

1.2 关于本手册

1.3 参考协议

1.4 功能总述

## 1-1、关于驱动器

---

### ◆ 产品确认

---

CS 伺服驱动器标配 CANopen 接口。

关于 HX 伺服驱动器的详细信息和使用方法请参考相关手册。

## 1-2、关于本手册

---

### ◆ 产品确认

---

本手册只描述 CANopen 协议在 HX 伺服驱动器的具体实现，不涉及协议内容的分析和解释。

本手册不可替代 CANopen 协议文档使用。

对 HX 伺服驱动的操作与调试必须由受训的专业人员进行。

CANopen 发布文档为英文，受个人水平、知识结构等等限制，在翻译成中文时难免存在歧义，如有不

明确之处请参照发布文档。

关于 CANopen 的更多信息请参见 CiA 官方网站 ([www.can-cia.org](http://www.can-cia.org))

## 1-3、参考协议

---

### ◆ 产品确认

---

CiA 301: CANopen application layer and communication profile, V4.2.0

CiA 402 Draft Standard Proposal: Device profile for drives and motion control, V3.0.0

## ◆ 产品确认

表 1.1 CS 伺服驱动器 CANopen 功能总述

功能	描述
链路层协议	CAN bus
应用层协议	CANopen
CAN-ID	11bits
波特率	50Kbit/s 100Kbit/s 125Kbit/s 250Kbit/s 500Kbit/s 1000Kbit/s
CAN 帧类型	数据帧

功能	描述
	不支持远程帧
通讯子协议	CiA 301 CiA 402 Draft Standard Proposal
通讯服务对象	NMT SDO PDO SYNC EMCY
SDO 传输方式	加速传输 不支持段传输
PDO 传输方式	时间触发 事件触发 同步触发
PDO 数目	4 路 TPDO 4 路 RPDO
操作模式	Profile Velocity Mode (PV) Profile Position Mode (PP) Homing Mode (HM) Interpolated Position Mode (IP)

# 2 · 接线与设置

2.1 通讯端口信号定义

2.2 通讯网络拓扑结构

2.3 适配线缆

2.4 参数设置

## 2-1、通讯端口信号定义

### ◆ 信号定义

CS 驱动器使用两个 RJ45 端子 (CN6A 和 CN6B) 作为 CAN 通信端口。CN6A 和 CN6B 端子内部直连，方便总线扩展。

表 2.1 总线通讯端子引脚分配

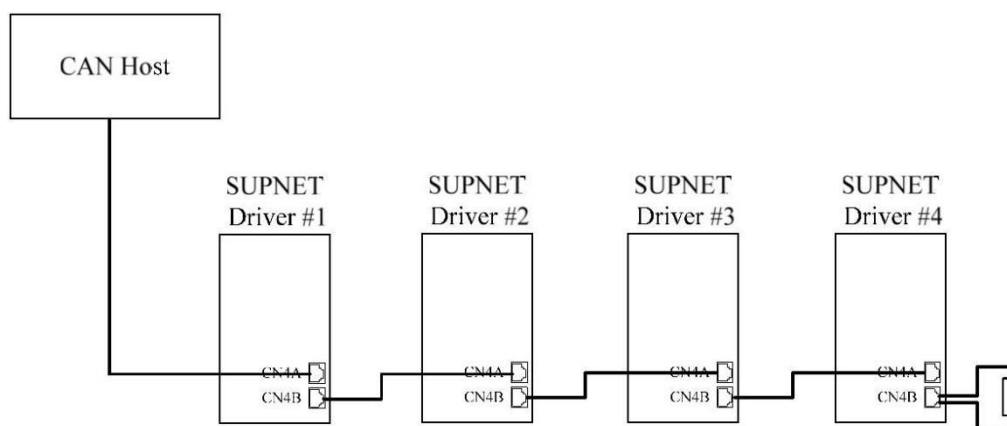
端子记号	名称	功能
4	CANH	CAN 通讯用端子
5	CANL	
8	GND	信号地
6	485+	RS-485 通讯用端子
7	485-	
1	—	保留
2	—	保留
3	—	保留
外壳	FG	机壳地



## 2-2、通讯网络拓扑结构

### ◆ 结构示意

下图为一个 4 轴系统的通讯网络拓扑结构示意图。





## 2-3、适配线缆

### ◆ 注意说明

为保障 CAN 通讯的稳定性和可靠性，通讯线缆的配置请遵循以下原则：

推荐使用双绞双屏蔽线缆，屏蔽层单点接地。

总线两端连接 120 欧姆终端电阻，防止信号反射。

CAN 通讯线缆应尽量远离电机动力线缆，防止受干扰。

有关线缆和通讯稳定性的关系，可参考相关的文献 (Controller Area Network protocol specification, Robert

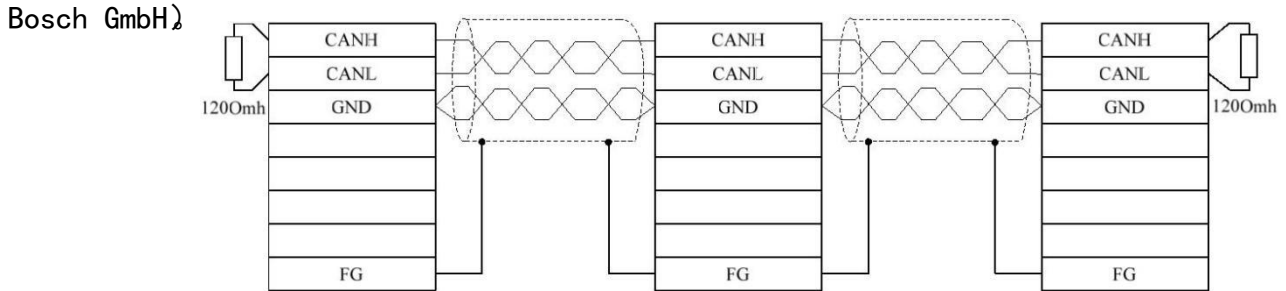


图 2.2 线缆示意图

## 2-4、参数设置

### ◆ 相关参数

通过操作显示面板可设置和 CANopen 通讯相关的 Pn 参数，包括控制方式、节点号和通讯速率。

表 2.2 CANopen 控制相关参数

参数号	参数名称	设定范围	设定单位	出厂设定	生效时间
	控制方式	0x00~0x0F	-	0	再次接通电源后
	0x00: 位置控制(脉冲列指令)				
Pn004.0	0x01: 速度控制(接点指令)				
	0x02: 速度控制(参数指令)				
	0x03~0x0E: 保留				
	0x0F: CANopen 总线控制				
Pn074	CANopen 通讯节点号	0~127	-	1	再次接通电源后
	CANopen 通讯波特率	0~5	-	4	再次接通电源后
	0: 50k				
	1: 100k				
Pn075	2: 125k				
	3: 250k				
	4: 500k				
	5: 1000k				

# 3 · 通讯网络配置

## 3.1 设备通讯

### 3.1.1 对象字典

### 3.1.2 通讯对象

## 3.2 网络管理对象 NMT

### 3.2.1 节点控制

### 3.2.2 错误控制

### 3.2.3 Boot-up

## 3.3 服务数据对象 SDO

## 3.4 过程数据对象 PDO

## 3.5 同步帧对象 SYNC

## 3.6 紧急帧对象 EMCY

### 3-1、设备通讯

#### ◆ 注意说明

CANopen 是 CAN 总线的应用层协议。CANopen 将设备参数组织为一个可访问的对象组（参数组），即对象字典。设备之间使用通讯对象实现数据交互，如 SDO、PDO 等。

CS 伺服驱动器的设备通讯模型如下图所示。

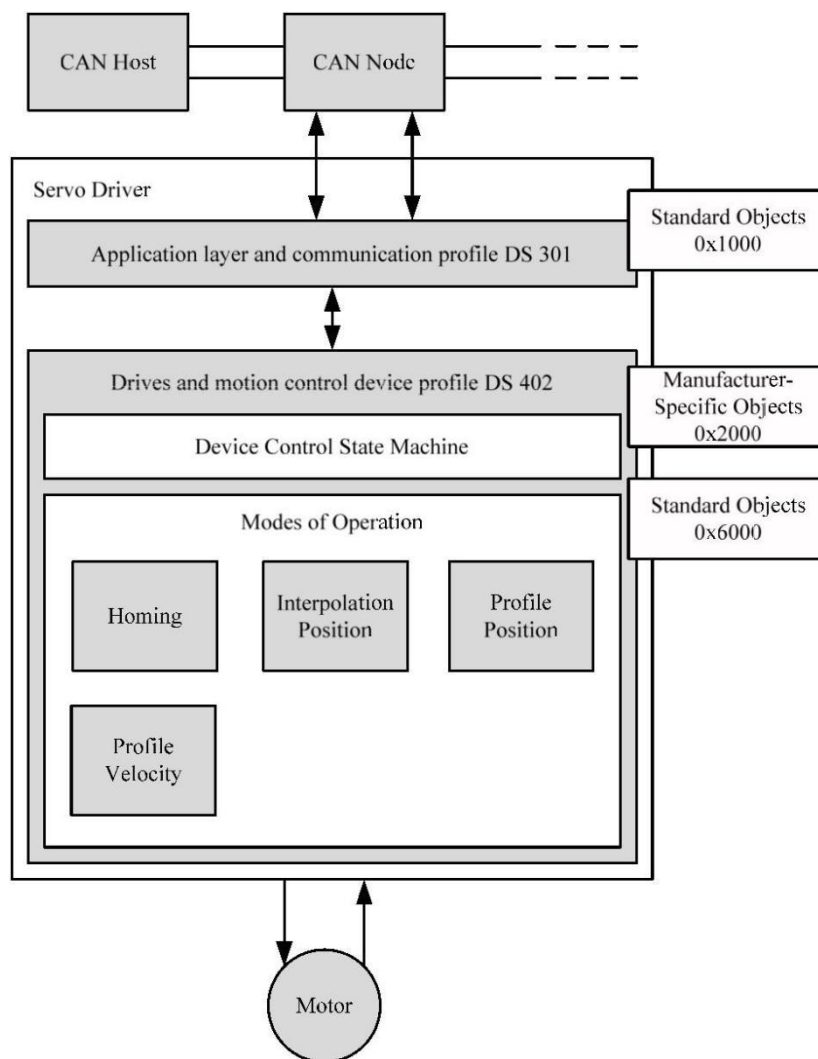


图 3.1 伺服驱动器设备通讯模型

### 3-1-1、对象字典

#### ◆ 注意说明

采用有序的预定义的方式来访问的一组对象。对象字典是一组参数和变量的有序集合，包含了设备描述及设备网络状态的所有参数。通过网络可采用有序的预定义的方式来访问的一组对象。

CANopen 协议采用了带有 16 位索引和 8 位子索引的对象字典，对象字典的区域分配定义如下表所示。

表 3.1 对象字典区域分配定义表

索引范围	描述
000h	保留
0001h - 025Fh	数据类型
0260h - 0FFFh	保留
1000h - 1FFFh	通讯对象子协议区
2000h - 5FFFh	厂家自定义子协议区
6000h - 9FFFh	标准化设备子协议区
A000h - AFFFh	网络变量（符合 IEC61131-3）
B000h - BFFFh	用于路由网关的系统变量
C000h - FFFFh	保留

### 3-1-2、通讯对象

#### ◆ 注意说明

通讯对象用于提取过程和服务数据、系统时间同步、异常监控、节点控制与节点状态监控。这些对象由其结构、传输类型和标识符（COB-ID）进行定义。

CAN 通过数据帧在主机和总线节点之间传输数据。CAN 数据帧的结构如下图所示。

图 3.2 CAN 数据帧结构

CS 伺服驱动器暂不支持远程帧。

通信对象标识符（COB-ID）指定了在通信过程中对象的优先级以及通信对象的识别。COB-ID 与 CAN

2.0A 的 11 位帧 ID 一一对应，11 位 COB-ID 由两部分组成，分别是 4 位的对象功能码和 7 位的节点 ID。

表 3.2 COB-ID 数据位组成结构

bit 位	0	1	2	3	4	5	6	7	8	9	10
描述	Node-ID										
功能码											

CANopen 通讯对象的默认 COB-ID 分配如下表所示。部分通讯对象的 COB-ID 可更改，参见具体章节信息。

表 3.3 通讯对象 COB-ID 预分配表

通讯对象	COB-ID 功能码	COB-ID	对象索引
NMT	0000b	000h	—
SYNC	0001b	080h	1005h、1006h、1007h
TIME STAMP	0010b	100h	1012h、1013h
EMCY	0001b	081h - 0FFh	1014h、1015h
TPD01	0011b	181h - 1FFh	1800h
RPD01	0100b	201h - 27Fh	1400h
TPD02	0101b	281h - 2FFh	1801h
RPD02	0110b	301h - 37Fh	1401h
TPD03	0111b	381h - 3FFh	1802h
RPD03	1000b	401h - 47Fh	1402h
TPD04	1001b	481h - 4FFh	1803h
RPD04	1010b	501h - 57Fh	1403h
SDO(server-to-client)	1011b	581h - 5FFh	1200h
SDO(client-to-server)	1100b	601h - 67Fh	1200h
HeartBeat	1110b	701h -77Fh	1016h、1017h

目前，CS 伺服驱动器实现的通讯对象包括：NMT、SDO、PDO、SYNC、EMCY 和 HeartBeat。

### 3-2、网络管理对象 NMT

#### ◆ 注意说明

.....  
 NMT 对象使用主-从 (Master-Slave) 模式，用于执行 NMT 服务，包括节点控制服务 (Node control services)、错误控制服务 (Error control services) 和 Boot-up 服务 (Boot-up services)。控制装置通过 NMT

对 CANopen 装置进行初始化、启动、监控、停止和复位等操作。

### 3-2-1、节点控制

#### ◆ 控制说明

NMT 主设备通过节点控制服务对 NMT 从设备的 NMT 状态进行控制，包括启动节点、停止节点、复位节点、复位通讯和进入预操作状态。

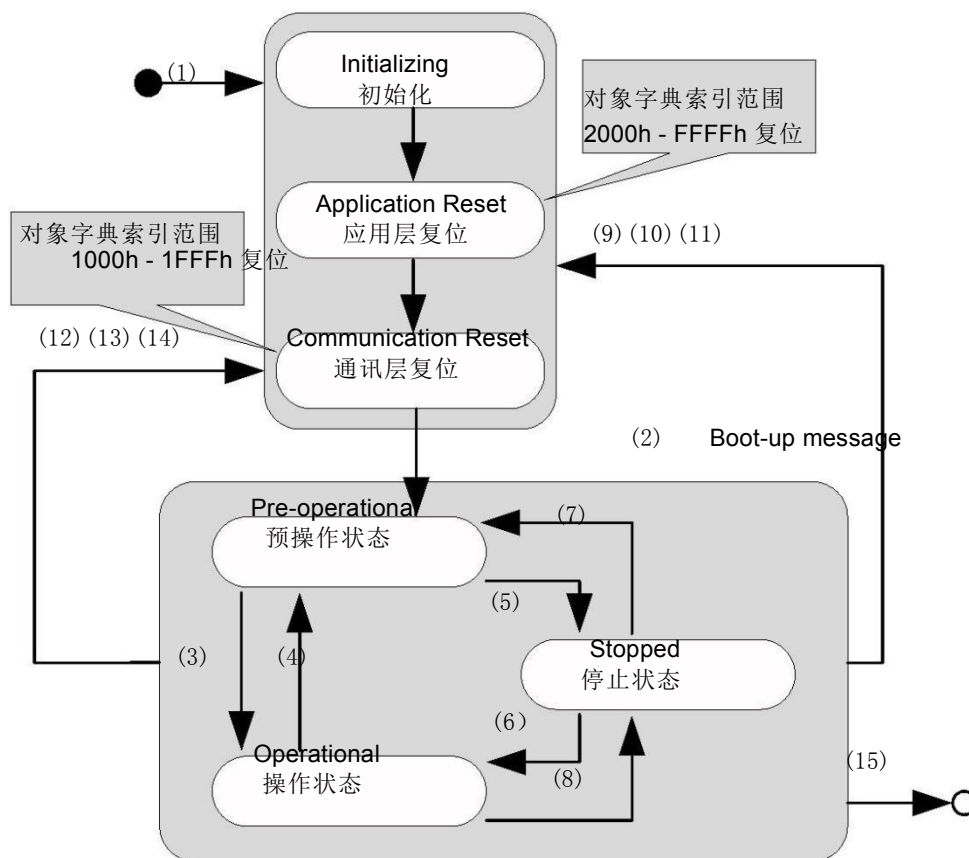
NMT 状态包括 Initialization、Pre-operational、Operational、Stopped 四种。

CS 伺服驱动器上电时处于 Initialization 状态，在完成初始化后自动进入 Pre-operational 状态。随后

伺服驱动器响应 NMT 主设备的节点控制服务，实现各 NMT 状态之间的切换。

在 CS 伺服驱动器中，NMT 状态机和设备控制状态机之间的耦合关系由对象 6007h (Abort connection option code) 建立。

复位通讯只复位 DS301 相关对象，而复位节点将复位所有对象（DS301、DS402、厂家自定义）。



(1) Power On 上电  
 (2) Auto switch to Pre-operational 自动切换到预操作状态  
 (3) and (6) NMT Switch to Operational 网络管理切换到操作状态  
 (4) and (7) NMT Switch to Pre-operational 网络管理切换到预操作状态  
 (5) and (8) NMT Switch to Stopped 网络管理切换到停止状态  
 (9), (10) and (11) NMT Switch to Application Reset 网络管理切换到应用层复位状态  
 (12), (13) and (14) NMT Switch to Communication Reset 网络管理切换到通讯层复位状态  
 (15) Power-off or hardware reset 掉电或硬件复位

图 3.4 NMT 状态机切换图

伺服驱动器在不同的 NMT 状态下，支持不同的通讯对象。

表 3.4 NMT 状态和通讯对象之间的关系

通讯对象	预操作	操作	停止
	Pre-operational	Operational	Stopped
PDO	×	√	×
SDO	√	√	×
SYNC	√	√	×

### 3-2-2、 错误控制

#### ◆ 控制说明

通讯对象	预操作 Pre-operational	操作 Operational	停止 Stopped
TIME	√	√	×
EMCY	√	√	×
Node control and error control	√	√	√

备所处的状态，包括节点保护 (Node guard)、寿命保护 (Life guard) 和心跳 (Heartbeat) CS 伺服驱动器的 CANopen 实现只支持心跳。

CS 伺服驱动器既可以作为心跳生产者 (Heartbeat Producer)，也可以作为心跳消费者 (Heartbeat Consumer)。相关的配置请参见对象 1016h (Heartbeat Consumer Time) 和 1017h (Heartbeat Consumer Time) 的描述。

CS 伺服驱动器作为心跳消费者时，将在收到第一个心跳帧之后开始对被监控节点的状态进行监控。

超时时伺服驱动器将进入报警状态，操作面板显示心跳超时报警。

CS 伺服驱动器作为心跳生产者时，进入 Stopped 状态后，将发送一个标记当前处于 Stopped 状态的心跳数据帧，随后停止发送心跳数据帧。

### 3-2-3、 Boot-up

#### ◆ 控制说明

当 NMT 状态由 Initialization 切换到 Pre-operational 时，HX 伺服驱动器将发出 Boot-up 数据帧。



### 3-3、 服务数据对象 SDO

#### ◆ 对象说明

SDO 提供了对象字典的存取操作接口，用于对象字典的读写操作。SDO 传输使用客户端-服务器

(client-server) 模式，客户端发起请求，服务器进行应答。一个客户端的请求一定有来自服务器的应答。

SDO 有加速传输和分段传输两种传输机制。CS 伺服驱动器目前只支持加速传输方式。

当客户端发出异常请求，服务器将使用 SDO 传输中止码进行应答。

表 3.5 SDO 传输中止码

中止码 Abort code	描述 Description
0503 0000h	Toggle bit not alternated 翻转位未变化
0504 0000h	SDO protocol timed out SDO 协议超时
0504 0001h	Client/server command specifier not valid or unknown 非法或未知的客户端/服务器命令字

中止码 Abort code	描述 Description
0504 0005h	Out of memory 内存溢出
0601 0000h	Unsupported access to an object 对象不支持访问
0601 0001h	Attempt to read a write only object 试图读只写对象
0601 0002h	Attempt to write a read only object 试图写只读对象
0602 0000h	Object does not exist in the object dictionary 对象字典中对象不存在
0604 0041h	Object cannot be mapped to the PDO 对象不能够映射到 PDO
0604 0042h	The number and length of the objects to be mapped would exceed PDO length 映射的对象的数目和长度超出 PDO 长度
0604 0043h	General parameter incompatibility reason 一般性参数不兼容
0604 0047h	General internal incompatibility in the device. 一般性设备内部不兼容
0606 0000h	Access failed due to an hardware error 硬件错误导致对象访问失败
0607 0010h	Data type does not match, length of service parameter does not match 数据类型不匹配, 服务参数长度不匹配
0607 0012h	Data type does not match, length of service parameter too high 数据类型不匹配, 服务参数长度太大
0607 0013h	Data type does not match, length of service parameter too low 数据类型不匹配, 服务参数长度太短
0609 0011h	Sub-index does not exist 子索引不存在
0609 0030h	Invalid value for parameter (download only) 超出参数数值的值范围
0609 0031h	Value of parameter written too high (download only) 写入参数数值太大
0609 0032h	Value of parameter written too low (download only) 写入参数数值太小
0609 0036h	Maximum value is less than minimum value 最大值小于最小值
060A 0023h	Resource not available: SDO connection SDO 连接不可用
0800 0000h	General error 一般性错误
0800 0020h	Data cannot be transferred or stored to the application 数据不能传送或保存到应用

中止码 Abort code	描述 Description
0800 0021h	Data cannot be transferred or stored to the application because of local control 由于本地控制导致数据不能传送或保存到应用
0800 0022h	Data cannot be transferred or stored to the application because of the present device state 由于当前设备状态导致数据不能传送或保存到应用
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error) 对象字典动态产生错误或对象字典不存在
0800 0024h	No data available 数值不存在

### 3-4、过程数据对象 PDO

#### ◆ 对象说明

.....  
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，传输效率高。

CS 伺服驱动器只支持一个生产者到一个消费者的点到点 PDO 传输，包含 4 路 TPDO 和 4 路 RPDO。

PDO 通讯由于没有协议限制，因此在启动 PDO 通讯前需使用 SDO 对传输参数和映射参数进行配置，既动态映射。CS 伺服驱动器支持 PDO 动态映射。

#### PDO 传输方式

CS 伺服驱动器支持协议定义的所有 PDO 传输方式。使用对象 1400h~1403h 设置 RPDO 传输方式。使用对象 1800h~1803h 设置 TPDO 传输方式。

#### 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 4 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 映射参数（如 1600h 或 1A00h）子索引 0 的内容为 0；

- 2、修改 PDO 通讯参数（如 1400h 或 1800h）
- 3、修改 PDO 映射参数（如 1600h 或 1A00h）子索引 1~4 的内容，映射数据；
- 4、设置 PDO 映射参数（如 1600h 或 1A00h）子索引 0 的内容为合法的数字（该 PDO 映射的对象数）；
- 5、PDO 映射完成。

### 3-5、同步帧对象 SYNC

#### ◆ 对象说明

同步帧对象用于控制数据在网络设备间的同步传输。同步帧对象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。

SYNC 报文的默认 COB-ID 为 080h，由对象 1005h(COB-ID SYNC message)定义。CS 伺服驱动器只能接收而不能产生 SYNC 报文。

### 3-6、紧急帧对象 EMCY

#### ◆ 对象说明

紧急帧对象遵循生产者-消费者模式。当设备检测到异常时会使用紧急帧对象传输异常代码，告知消费者当前驱动器错误类型及代码，并更新 1001h (Error register) 和 1003h (Pre-defined error field)。

表 3.6 紧急帧数据位结构

字节	0	1	2	3	4	5	6	7
描述	Error code 错误码		Error register 错误寄存器	Manufacturer-specific 厂家自定义				

在 CS 伺服驱动器的 CANopen 实现中，对厂家自定义区 bit3~bit7 位做了特殊定义。

表 3.7 紧急帧厂家自定义数据位

字节	描述
3	错误区分码： 00h - 无错误 08h~0Fh - 电流类错误 10h~17h - 电压类错误 18h~1Fh - 温度类错误 20h~27h - 通讯类错误 28h~2Fh - 设备相关错误 30h~FFh - 其他类型错误
4	报警号：伺服驱动器报警编号
5	
6	保留（常为 0）
7	保留（常为 0）

如果所有的异常条件都已消失(复位或异常自恢复),则伺服驱动器会发送一个紧急帧给消费者,其中

错误码 (Error code) 为 0。

错误码 (Error code) 的具体含义请参考第 8 章《故障及处理》。

## 4 · 设备控制

4.1 设备状态机

4.2 控制字

4.3 状态字

4.4 停机方式

## 4-1、设备状态机

### ◆ 注意说明

设备状态机是对设备控制过程和状态的抽象描述。

控制装置通过发送相应的控制字（Controlword, 6040h）控制实现对伺服驱动状态机切换的控制。而状态字（Statusword, 6041h）则反应了伺服驱动器当前的状态信息。同时，NMT 指令、故障信号等也会影响

状态机的跳转。

状态机的描述框图如下所示。

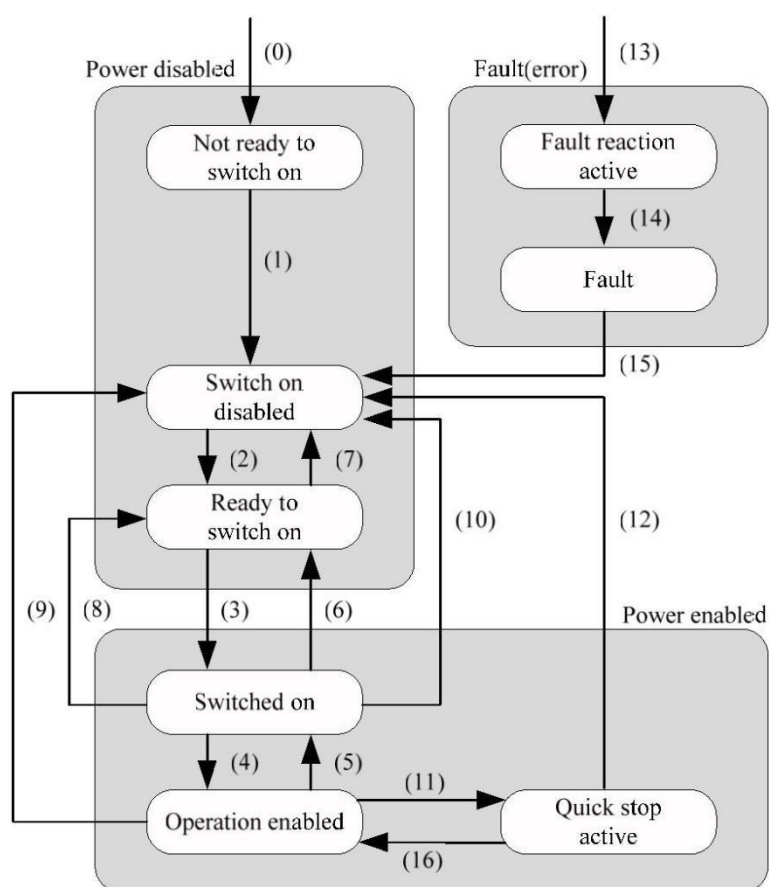


图 4.1 伺服状态切换图

#### Not ready to switch on

伺服驱动器上电初始化，完成后自动跳转至 Switch on disabled，未建立 CAN 通讯连接。

#### Switch on disabled

伺服驱动器上电初始化完成，此时 CAN 通讯已建立，等待上位机控制指令。

**Ready to switch on**

伺服驱动器已准备好给强电驱动部分供电，等待指令进入 Switched on。

**Switched on**

伺服驱动器准备好状态，强电驱动部分已上电，电机无励磁。

**Operation enabled**

伺服驱动器为电机输入励磁信号，按照设定的操作模式控制电机运转。

**Quick stop active**

伺服驱动器将根据设定的方式停机。停机后的状态跳转取决于相关的停机方式。

**Fault reaction active**

伺服驱动器检测到报警发生，按照设定的方式停机。停机后自动跳转至 Fault。

**Fault**

伺服驱动器处于报警状态，电机无励磁。

## 4-2、控制字

### ◆ 注意说明

伺服驱动器接收、解析控制字（Controlword, 6040h）中的指令实现对状态机的控制。控制字的位说明

如下图所示。

表 4.1 控制字数据位描述

bit 位	描述
0	Switch on
1	Enable voltage
2	Quick stop
3	Enable operation
4	Operation mode specific
5	
6	
7	Fault reset
8	Halt
9	Reserved
10	
11	Manufacturer specific
12	
13	
14	
15	



CS 伺服驱动器中对控制字 bit15-bit11 暂未做定义，固定值为 0。

bit8 为 1 时，伺服驱动器挂起（暂停）当前的运行动作，对象 605Dh (Halt option code) 决定如何停机。

bit6-bit4 的实际含义在不同的操作模式下有所不同。具体信息参见操作模式相关章节。

状态机的切换由控制字 bit7、bit3-bit0 组合而成的控制命令触发。

表 4.2 控制字和状态机切换关系

切换序号	切换时机	命令类型	控制字					驱动器响应动作
			bit7	bit3	bit2	bit1	bit0	
(0)	驱动器上电或硬件复位	自动切换	0	×	×	×	×	驱动器初始化
(1)	驱动器初始化完成	自动切换	0	×	×	×	×	驱动器完成初始化，此时 CAN 通讯有效
(2)	收到 Shut down 命令	Shut down	0	×	1	1	0	无动作
(3)	收到 Switch on 命令	Switch on	0	0	1	1	1	驱动器判断是否符合切换条件，是则切换，否则无动作
(4)	收到 Enable operation 命令	Enable operation	0	1	1	1	1	无动作
(5)	收到 Disable operation 命令	Disable operation	0	0	1	1	1	驱动器按照设定的停机方式停机，然后切换状态
(6)	收到 Shut down 命令	Shut down	0	×	1	1	0	无动作
(7)	收到 Quick stop 或 Disable voltage 命令	Quick stop	0	×	0	1	×	无动作
		Disable voltage	0	×	×	0	×	
(8)	收到 Shut down 命令	Shut down	0	×	1	1	0	无动作
(9)	收到 Disable voltage 命令	Disable voltage	0	×	×	0	×	驱动器按照设定的停机方式停机，然后切换状态
(10)	收到 Quick stop 或 Disable voltage 命令	Quick stop	0	×	0	1	×	无动作
		Disable voltage	0	×	×	0	×	
(11)	收到 Quick stop 命令	Quick stop	0	×	0	1	×	驱动器按照设定的停机方式停机，然后切换状态
(12)	伺服停止后自动切换	自动切换	0	×	×	×	×	无动作
(13)	驱动器发生故障	自动切换	0	×	×	×	×	无动作
(14)	伺服停止后自动切换	自动切换	0	×	×	×	×	驱动器按照设定的停机方式停机，然后切换状态
(15)	收到 Fault reset 命令	Fault reset ↑	×	×	×	×	×	驱动器复位已存在的所有故障，然后切换状态
(16)	收到 Enable operation 命令	Enable operation	0	1	1	1	1	驱动器停机动作完成后自动切换

注：×表示可忽略状态。

## 4-3、状态字

### ◆ 注意说明

状态字 (Statusword, 6041h) 实时反应状态机的当前状态。状态字的位说明如下图所示。

表 4.3 状态字数据位描述

bit 位	描述
0	Ready to switch on
1	Switched on
2	Operation enabled
3	Fault
4	Voltage enabled
5	Quick stop
6	Switch on disabled
7	Warning
8	Manufacturer specific
9	Remote
10	Target reached
11	Internal limit active
12	Operation mode specific
13	
14	Manufacturer specific
15	

CS 伺服驱动器中对状态字 bit15-bit14、bit8 暂未做定义，固定值为 0。

bit13-bit12 的实际含义在不同的操作模式下有所不同。具体信息参见操作模式相关章节。

bit11 位 1 时, 表示伺服到达正转或反转极限位置 (POT 或 NOT)。可以通过访问对象 60FDh (digital inputs)

区分。

bit9、bit7 暂未做定义，固定值为 0。

bit6-bit0 反应状态机当前状态，其状态编码方式如下表所示。

表 4.3 状态字和状态机状态之间关系

状态字	状态机状态
xxxx xxxx x0xx 0000b	Not ready to switch on
xxxx xxxx x1xx 0000b	Switch on disabled
xxxx xxxx x01x 0001b	Ready to switch on
xxxx xxxx x01x 0011b	Switched on
xxxx xxxx x01x 0111b	Operation enabled
xxxx xxxx x00x 0111b	Quick stop active
xxxx xxxx x0xx 1111b	Fault reaction active
xxxx xxxx x0xx 1000b	Fault

注：x 表示可忽略状态。

#### 4-4、停机方式

##### ◆ 注意说明

当 CS 伺服驱动器处于 CANopen 控制方式时，Pn 参数中的超程、S-OFF 和报警停机方式无效。

CS 伺服驱动器的 CANopen 实现支持以下几种停机：

**伺服 OFF 停机：**

定义伺服 OFF 时的停机方式，参见对象 605Ch(Disable operation option code)的描述。

**伺服报警停机：**

定义伺服报警时的停机方式，参见对象 605Eh(Fault reaction option code)的描述。

**伺服快速停机 (Quick stop)**

定义伺服处理 Quick stop 指令时的停机方式，参见对象 605Ah(Quick stop option code)的描述。

**伺服超程停机：**

定义伺服到达正转或反转极限位置 (POT 或 NOT) 时的停机方式，参见对象 2003h 的描述。

**伺服挂起停机 (Halt)**

定义伺服处理 Halt 指令时的停机方式，参见对象 605Dh(Halt option code)的描述。

**伺服关闭停机 (Shut down)**

定义伺服 Operation Enabled 状态跳转至 Ready to switch on 状态时的停止方式，参见对象 605Bh(Shutdown

option code)的描述。

## 5 · 单位转换

5.1 驱动器内部单位

5.2 用户单位

5.3 单位转换方法

5.4 指令极性

## 5-1、驱动器内部单位

### ◆ 注意说明

伺服驱动器内部单位直接和电机相关。

#### 位置

电机编码器增量脉冲数，pls。

#### 速度

电机旋转速度，rpm。

#### 加速度

电机旋转加速度，rpm/s。

#### 电流

电机额定电流的千分比，电机额定电流\*1/1000。电机额定电流由对象 6075h (Motor Rated Current) 定义。

## 5-2、用户单位

### ◆ 注意说明

用户单位直接和实际负载的运动相关。

#### 位置

如果负载是直线运动则位置单位可能是 um, mm 等长度单位。对于旋转运动的负载，位置单位可能是度 degree, 弧度 rad 等角度单位。

#### 速度

用户速度单位为位置的单位时间变化率，如 um/s、mm/s、rad/s 等。

#### 加速度

用户加速度单位为速度的单位时间变化率，如 um/s/s、mm/s/s、rad/s/s 等。

#### 电流

用户电流单位和驱动器内部使用单位相同。

### 5-3、单位转换方法

#### ◆ 注意说明

CS 伺服驱动器使用位置传感器（编码器）来测量电机的位置信息。因此，速度和加速度的测量和

表示都是基于位置单位的。例如，用户位置单位为 mm，则速度和加速度的单位自动表示为 mm/s、mm/s/s。

根据协议规定，为实现正确的单位转换需设置以下相关对象（参数）：

**位置编码器分辨率（Position encoder resolution, 608Fh）**

定义电机旋转一圈所对应的电机位置脉冲数，设为 X/Y。

608Fh 在正确选定电机型号参数后由伺服驱动器根据电机型号自动设置。

**齿轮传动比（Gear ratio, 6091h）**

定义电机侧和负载侧的齿轮传动比，设为 M/N。

6091h 为读写对象，由用户根据机械传动参数设置。

**进给常数（Feed constant, 6092h）**

定义负载侧齿轮旋转一圈对应的负载所移动的距离，设为 A/B。

6092h 为读写对象，由用户根据机械传动参数设置。

正确设置上述参数（对象）后，最终得到的单位转换比例因子，即传统意义上的电子齿轮比为：

$$(X/Y) * (M/N) * (A/B)$$

示例

假设伺服驱动器编码器为 2500 线光电编码器，机械齿轮传动比为 2:1，进给量为电机转 1 圈负载移动

10mm，则：

位置编码器分辨率（608Fh）的设置为：

$$608Fh\_01h = 10000$$

$$608Fh\_02h = 1$$

齿轮传动比（6091h）的设置为：

$$60891\_01h = 2$$

$$6091h\_02h = 1$$

进给常数（6092h）的设置为：

$$60892\_01h = 10$$

$$6092h\_02h = 1$$

如果用户给定位置 100mm，则计算得到的驱动器内部位置为：

$$100mm * (1/10) * (2/1) * (10000/1) = 200000 \text{ pls}$$

如果电机当前位置 100000pls, 则反馈给上位装置的用户单位位置为:

$$100000\text{pls} * (10/1) * (1/2) * (1/10000) = 50\text{mm}$$

如果用户给定速度 100mm/s, 则计算得到的驱动器内部速度为:

$$100\text{mm/s} * (1/10) * (2/1) * (10000/1) = 200000\text{pls/s}$$

$$= 200000 * (1/10000) * 60\text{rpm} = 1200\text{rpm}$$

如果用户给定加速度 100mm/s/s, 则计算得到的驱动器内部加速度为:

$$100\text{mm/s/s} * (1/10) * (2/1) * (10000/1) = 200000\text{pls/s/s}$$

$$= 200000 * (1/10000) * 60\text{rpm/s} = 1200\text{rpm/s}$$

#### 5-4、指令极性

##### ◆ 注意说明

对象 607Eh(Polarity)定义了控制指令的极性, 即目标指令乘 1 或-1。

在CS 伺服驱动器中, 607Eh 只影响给定指令, 对实际位置和速度无效。因此, 如果用户设定了反极性, 需特别注意实际位置和速度的符号或方向。607Eh 在以下操作模式有效:

Profile Position(PP)

Profile Velocity(PV)。

# 6 · 操作模式

## 6.1 设置和切换操作模式

## 6.2 Profile Velocity (PV) 模式

### 6.2.1 操作步骤

### 6.2.2 可选操作

### 6.2.3 相关对象

## 6.3 Profile Position (PP) 模式

### 6.3.1 操作步骤

### 6.3.2 可选操作

### 6.3.3 相关对象

## 6.4 Homing (HM) 模式

### 6.4.1 操作步骤

### 6.4.2 相关对象

## 6.5 Interpolated Position (IP) 模式

### 6.5.1 操作步骤

### 6.5.2 相关对象

## 6.6 限位处理流



## 6-1、设置和切换操作模式

### ◆ 模式种类

CS 伺服驱动器支持以下操作模式：

Profile Velocity (PV)；

Profile Position (PP)；

Homing (HM)；

Interpolated Position (IP)。

CS 伺服驱动器支持的操作模式可通过读取对象 6502h (Supported Drive Modes) 进行查看。对象 6061h (Modes of Operation Display) 显示了伺服驱动中当前有效的操作模式。

上电或复位后，HX 伺服驱动器默认的操作模式为 Profile Position (PP)。

通过 SDO 改写操作模式为不支持的操作模式时，驱动器返回相应的中止码。通过 PDO 改写操作模式为不支持的操作模式时，驱动器不做任何响应。

CS 伺服驱动器支持运行过程中更改操作模式，并立即生效。但是如果当前状态不满足切换条件，则驱动器对模式切换操作不做响应。操作模式切换的限制条件为：

状态字 bit10 (Target Reached) 为 1；电机实际速度小于 10rpm。

## 6-2、Profile Velocity (PV) 模式

### ◆ 模式描述

伺服驱动器根据用户给定的目标速度、加速度和减速度，自动生成速度轨迹曲线，并控制电机按照设定轨迹运行。

## 6-2-1、操作步骤

### ◆ 操作说明

- 1、设定操作模式 6060h (Mode of operations) 为 profile velocity mode (H0303)。
- 2、设定控制字 6040h (Controlword) 为 (0X06→0X07→0X0F)，使驱动器上电及电机运作。
- 3、设定规划加速度 6083h (Profile acceleration)。
- 4、设定规划减速度 6084h (Profile deceleration)。
- 5、设定目标速度 60FFh (Target velocity)。
- 6、读状态字 6041h (Statusword) 获取驱动器状态。

## 6-2-2、可选操作

### ◆ 操作说明

- 1、读取 606Bh (Velocity demand value)，获取内部给定速度。
- 2、读取 606Ch (Velocity actual value)，获取实际速度。
- 3、设定 606Dh (Velocity window)，速度到达区间。
- 4、设定 606Eh (Velocity window time)，判定速度到达时间。
- 5、设定 606Fh (Velocity threshold)，分配零速度准位。

## 6-2-3、相关对象

### ◆ 操作说明

索引	名称	数据类型	访问属性
6040h	Controlword	Uint16	RW
6041h	Statusword	Uint16	RO
6060h	Mode of operations	Int8	RW
6061h	Modes of Operation Display	Int8	RO
606Bh	Velocity demand value	Int32	RO
606Ch	Velocity actual value	Int32	RO
606Dh	Velocity window	Uint16	RW
606Eh	Velocity window time	Uint16	RW
606Fh	Velocity threshold	Uint16	RW
60FFh	Target velocity	Int32	RW

## 6-3、Profile Position (PP) 模式

### ◆ 模式描述

Profile Position(PP) 模式是一种点位运行方式。伺服驱动器从外部上位控制器接收位置命令，然后控制伺服电机到达目标位置。

### 6-3-1、操作步骤

#### ◆ 操作说明

- 1、设定操作模式 6060h(Mode of operations)为 profile position mode (H0101)。
- 2、设定目标位置 607Ah(Target position)。
- 3、设定目标速度 6081h(Profile velocity)。
- 4、设定规划 6083h(Profile acceleration)。
- 5、设定规划减速度 6084h(Profile deceleration)。
- 6、设定控制字 6040h(Controlword)为(0X06→0X07→0X0F→0X1F)，使驱动器上电及电机运作。
- 7、读取电机实际位置 6064h(Position actual value)。
- 8、读状态字 6041h(Statusword)获取驱动器状态(following error、set-point acknowledge、target reached)。

### 6-3-2、可选操作

#### ◆ 操作说明

- 1、读取 6062h(Position demand value)，获取给定位置。
- 2、读取 60FCh(Position demand internal value)，获取内部给定位置。
- 3、读取 6063h(Position actual value)，获取电机实际位置。
- 4、设定 6065h(Following error window)，定义实际位置和目标位置的偏差窗口范围。
- 5、读取 60F4h(Following error actual value)，获取实际的跟随误差。
- 6、设定 6067h(Position window)，定义位置窗口误差范围。
- 7、设定 6068h(Position window time)，定义位置窗口时间。

## 6-3-3、相关对象

## ◆ 操作说明

索引	名称	数据类型	访问属性
6040h	Controlword	Uint16	RW
6041h	Statusword	Uint16	RO
6060h	Mode of operations	Int8	RW
6061h	Modes of Operation Display	Int8	RO
6062h	Position demand value	Int32	RO
6063h	Position actual value	Int32	RO
6064h	Position actual value	Int32	RO
6065h	Following error window	Uint32	RW
6067h	Position window	Uint32	RW
6068h	Position window time	Uint16	RW
607Ah	Target position	Int32	RW
6081h	Profile velocity	Uint32	RW
6083h	Profile acceleration	Uint32	RW
6084h	Profile acceleration	Uint32	RW
608Fh	Position encoder resolution	Array/Uint32	RW
6091h	Gear ratio	Array/Uint32	RW
6092h	Feed constant	Array/Uint32	RW
60F4h	Following error actual value	Int32	RO
60FCh	Position demand internal value	Int32	RO

## 6-4、Homing (HM) 模式

### ◆ 模式描述

原点回零模式用于寻找机械原点，并定义零点和机械原点之间的位置关系。机械原点的位置由开关或是 Z 脉冲决定，而零点位置可设定为相对于机械零点的任意位置。

按照 CANopen 协议规定，回零完成后，电机停止位置为机械原点，通过设置 607Ch (Home offset)，可以设定机械原点与零点的关系：

零点 = 机械零点 + 607Ch (Home offset)。

### 6-4-1、操作步骤

#### ◆ 操作说明

- 1、设定操作模式 6060h (Mode of operations) 为 homing mode (H0606)。
- 2、设定零点偏移量 607Ch (Home offset)。
- 3、设定回零方式 6098h (Homing method)。
- 4、设定回零速度 6099h (Homing speeds)。
- 5、设定回零加减速 609Ah (Homing acceleration)。
- 7、设定控制字 6040h (Controlword) 为 (0x06→0x07→ 0x0F)，使驱动器上电。
- 8、设定控制字 6040h (Controlword) 为 (0x0F→0x1F)，执行原点回归动作。
- 9、读取读状态字 6041h (Statusword) 获取驱动器状态 (Target reached、Homing attained、Homing error)。
- 10、回零完成后电机实际位置为回零偏移量 607Ch (Home offset)。
- 11、回零完成后，设定设定操作模式 6060h (Mode of operations) 为需要的模式。

## 6-4-2、相关对象

### ◆ 操作说明

索引	名称	数据类型	访问属性
6040h	Controlword	Uint16	RW
6041h	Statusword	Uint16	RO
6060h	Mode of operations	Int8	RW
6061h	Modes of Operation Display	Int8	RO
607Ch	Home offset	Int32	RW
608Fh	Position encoder resolution	Array/Uint32	RW
6091h	Gear ratio	Array/Uint32	RW
6092h	Feed constant	Array/Uint32	RW
6098h	Homing method	Int8	RW
6099h	Homing speeds	Array/Uint32	RW
609Ah	Homing acceleration	Uint32	RW

## 6-5、Interpolated Position (IP) 模式

### ◆ 模式描述

伺服驱动器根据同步帧 SYNC，同步接收上位机目标位置指令，并对相邻两个周期的目标位置进行插值、细分作为位置环输入。Interpolated Position(IP) 模式主要用于多轴同步控制。

## 6-5-1、操作步骤

### ◆ 操作说明

- 1、设定操作模式 6060h (Mode of operation) 为 interpolation position mode (H0707)。
- 2、设定插值模式 60C0h (interpolation sub mode select)。
- 3、设定插值周期 60C2h (Interpolation time period)。
- 4、设定控制字 6040h (Controlword) 为 (0x06→0x07→ 0x0F)，使驱动器上电。
- 5、设定控制字 6040h (Controlword) 为 (0x0F→ 0x1F)，使能位置插值。
- 6、设定目标位置 60C1h (interpolation data record)，驱动器开始定位运行。
- 7、读取读状态字 6041h (Statusword) 获取驱动器状态。

## 6-5-2、相关对象

### ◆ 操作说明

索引	名称	数据类型	访问属性
6040h	Controlword	Uint16	RW
6041h	Statusword	Uint16	RO
6060h	Mode of operations	Int8	RW
6061h	Modes of Operation Display	Int8	RO
60C0h	interpolation sub mode select	Int16	RW
60C1h	interpolation data record	Record	RW
60C2h	Interpolation time period	Record	RW

## 6-6、限位处理流程

### ◆ 操作说明

本节描述只适用于限位开关接入伺服驱动器的应用场合。

CS 伺服驱动器一旦到达正负极限传感器位置时，将按照伺服超程停机设定的方式立即停机。

驱动器限位后不会产生报警，不再接收限位方向上的目标指令，电机将根据伺服超程停机的设定处于锁死或自由状态。需按照下述操作流程来解除限制：

- 1、读取状态字 6041h(Statusword)，根据 bit11 判断当前是否处于限位状态。
- 2、读取数字输入 60FDh (digital inputs)，区分正负限位。
- 3、将目标指令反向，控制驱动器反向运行，直到状态字 6041h(Statusword)的 bit11 为 0。



## 7 · 字典对象

7.1 通信子协议对象

7.2 厂家自定义对象

## 7-1、通信子协议对象

### ◆ 参数说明

CS 伺服驱动器的 CANopen 实现支持以下通信协议子对象：

- 1000h - 设备类型 (Device Type)
- 1001h - 错误寄存器 (Error Register)
- 1003h - 预定义错误域 (Predefined Error Field)
- 1005h - 同步帧 COB-ID (COB-IDSYNCMesssage)
- 1006h - 通讯循环周期 (Communication Cycle Period)
- 1007h - 同步窗口长度 (Synchronous Window Length)
- 1008h - 制造商设备名称 (Manufacturer Device Name)
- 1014h - 紧急帧 COB-ID (COB-ID EMCY)
- 1016h - 消费者心跳时间 (Consumer Heartbeat Time)
- 1017h - 生产者心跳时间 (Producer Heartbeat Time)
- 1018h - ID 对象 (Identity Object)
- 1200h - SDO 服务器参数 1 (Server SDO Parameter 1)
- 1400h - RPDO 通讯参数 1 (Receive PDO Communication Parameter 1)
- 1401h - RPDO 通讯参数 2 (Receive PDO Communication Parameter 2)
- 1402h - RPDO 通讯参数 3 (Receive PDO Communication Parameter 3)
- 1403h - RPDO 通讯参数 4 (Receive PDO Communication Parameter 4)
- 1600h - RPDO 映射参数 1 (Receive PDO Mapping Parameter 1)
- 1601h - RPDO 映射参数 2 (Receive PDO Mapping Parameter 2)
- 1602h - RPDO 映射参数 3 (Receive PDO Mapping Parameter 3)
- 1603h - RPDO 映射参数 4 (Receive PDO Mapping Parameter 4)
- 1800h - TPDO 通讯参数 1 (Transmit PDO Communication Parameter 1)
- 1801h - TPDO 通讯参数 2 (Transmit PDO Communication Parameter 2)
- 1802h - TPDO 通讯参数 3 (Transmit PDO Communication Parameter 3)
- 1803h - TPDO 通讯参数 4 (Transmit PDO Communication Parameter 4)
- 1A00h - TPDO 映射参数 1 (Transmit PDO Mapping Parameter 1)
- 1A01h - TPDO 映射参数 2 (Transmit PDO Mapping Parameter 2)
- 1A02h - TPDO 映射参数 3 (Transmit PDO Mapping Parameter 3)
- 1A03h - TPDO 映射参数 4 (Transmit PDO Mapping Parameter 4)

## 1000h – 设备类型(Device Type)

<b>Index</b>	1000h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0002 0192h
<b>Description</b>	描述所使用的设备子协议或应用规范

## 1001h – 错误寄存器(Error Register)

<b>Index</b>	1001h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	<p>错误寄存器反应驱动器内部错误信息，是紧急帧的组合部分。</p> <p>每一个 bit 位反应一种错误信息：</p> <ul style="list-style-type: none"> <li>bit0 = 常规错误</li> <li>bit1 = 电流</li> <li>bit2 = 电压</li> <li>bit3 = 温度</li> <li>bit4 = 通讯错误</li> <li>bit5 = 设备相关错误</li> <li>bit6 = 保留</li> <li>bit7 = 厂家自定义</li> </ul> <p>出现错误时，错误相应的位为“1”，且只要有错误，bit0 必须为“1”。</p>

## 1003h – 预定义错误域(Predefined Error Field)

<b>Index</b>	1003h
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	保存已由 EMCY 通知的错误信息，访问该对象可获得伺服驱动器错误的历史记录。

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	已发生的错误数。 写 0 则清除所有错误记录。 非 0 值不能写入。

<b>Sub-Index</b>	01h~08h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	预定义错误域信息。 子索引 0 不为 0 时可访问。 bit32~bit16: 伺服驱动器报警编号。 bit15~bit0: 标准错误码。

## 1005h – 同步帧 COB-ID(COB-ID SYNC Message)

<b>Index</b>	1005h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW

<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0080h
<b>Description</b>	同步帧报文 COB-ID

## 1006h – 通讯循环周期(Communication Cycle Period)

<b>Index</b>	1006h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access RW</b>	
<b>PDO mapping</b>	NO
<b>Units</b>	us
<b>Default value</b>	0000 0000h
<b>Description</b>	<p>设定同步帧报文通讯周期。</p> <p>如果在设定周期的 1.5 倍时间内，伺服驱动器未收到同步帧报文，则判定同步帧报文传输超时。</p> <p>0 值设定无效，不做超时判断。</p>

## 1007h – 同步窗口长度(Synchronous Window Length)

<b>Index</b>	1007h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	us
<b>Default value</b>	0000 0000h
<b>Description</b>	<p>设定同步 PDO 的时间窗口。</p> <p>如果伺服驱动器在窗口时间内未收到 SYNC，则丢弃当前同步周期内的 PDO，并发出 EMCY。</p> <p>0 值设定无效。</p>

## 1008h – 制造商设备名称(Manufacturer Device Name)

<b>Index</b>	1008h
--------------	-------

<b>Object code</b>	VAR
<b>Data type</b>	Visible_String
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	SUPD
<b>Description</b>	厂家定义的设备名称

## 1014h – 紧急帧 COB-ID(COB-ID EMCY)

<b>Index</b>	1014h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0081h
<b>Description</b>	紧急帧报文 COB-ID

## 1016h – 消费者心跳时间(Consumer Heartbeat Time)

<b>Index</b>	1016h
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	设定消费者监控心跳的循环时间

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	1
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
------------------	-----

<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	ms
<b>Default value</b>	0
<b>Description</b>	消费者监控心跳的循环时间

## 1017h – 生产者心跳时间(Producer Heartbeat Time)

<b>Index</b>	1017h
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	设定生产者产生心跳的循环时间

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	1
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	ms
<b>Default value</b>	0
<b>Description</b>	生产者产生心跳的循环时间

## 1018h – ID 对象(Identity Object)

<b>Index</b>	1018h
--------------	-------

<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设备 ID 信息

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	4
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	厂商 ID

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0322h
<b>Description</b>	产品代码

<b>Sub-Index</b>	03h
<b>Object code</b>	VAR
<b>Data type</b>	UInt32



<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	修订版本号

<b>Sub-Index</b>	04h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0100
<b>Description</b>	产品序列号

## 1200h – SDO 服务器参数 1(Server SDO Parameter 1)

<b>Index</b>	1200h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 SDO 服务器参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST

<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0600h + NodeID
<b>Description</b>	客户端到服务器 COB-ID

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0580h + NodeID
<b>Description</b>	客户端到服务器 COB-ID

## 1400h – RPDO 通讯参数 1(Receive PDO Communication Parameter 1)

<b>Index</b>	1400h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO1 通讯参数。

<b>Sub-Index</b>	0h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO

<b>Units</b>	—
<b>Default value</b>	0000 0200h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型。

## 1401h – RPDO 通讯参数 2(Receive PDO Communication Parameter 2)

<b>Index</b>	1401h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO2 通讯参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32

<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0000 0300h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型。

## 1402h – RPDO 通讯参数 3(Receive PDO Communication Parameter 3)

<b>Index</b>	1402h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO3 通讯参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
------------------	-----

<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	8000 0400h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型

## 1403h – RPDO 通讯参数 4(Receive PDO Communication Parameter 4)

<b>Index</b>	1403h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO4 通讯参数。

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	8000 0500h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型

## 1600h – RPDO 映射参数 1(Receive PDO Mapping Parameter 1)

<b>Index</b>	1600h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	<p>设定 RPDO1 映射对象。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST

<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6040 0010h
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6060 0008h
<b>Description</b>	映射对象 2

## 1601h – RPDO 映射参数 2(Receive PDO Mapping Parameter 2)

<b>Index</b>	1601h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO2 映射参数。 在 PDO 无效时更改才能生效。

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO

<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	607A 0020h
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6081 0020h
<b>Description</b>	映射对象 2

## 1602h – RPDO 映射参数 3(Receive PDO Mapping Parameter 3)

<b>Index</b>	1602h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO3 映射参数。 在 PDO 无效时更改才能生效。

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO



<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	607A 0020h
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6081 0020h
<b>Description</b>	映射对象 2

## 1603h – RPDO 映射参数 4(Receive PDO Mapping Parameter 4)

<b>Index</b>	1603h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 RPDO4 映射参数。 在 PDO 无效时更改才能生效。

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO

<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	607A 0020h
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6081 0020h
<b>Description</b>	映射对象 2

## 1800h – TPDO 通讯参数 1(Transmit PDO Communication Parameter 1)

<b>Index</b>	1800h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 TPDO1 通讯参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—

<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	4000 0180h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型

## 1801h – TPDO 通讯参数 2(Transmit PDO Communication Parameter 2)

<b>Index</b>	1801h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 TPDO2 通讯参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST

<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	4000 0280h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型

## 1802h – TPDO 通讯参数 3(Transmit PDO Communication Parameter 3)

<b>Index</b>	1802h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 TPDO3 通讯参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR

<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	C000 0380h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型

## 1803h – TPDO 通讯参数 4(Transmit PDO Communication Parameter 4)

<b>Index</b>	1803h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	设定 TPDO4 通讯参数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	C000 0480h + NodeID
<b>Description</b>	<p>PDO 的 COB-ID。</p> <p>最高为设为 1，则禁能该 PDO 的传输。</p> <p>不支持 CAN 扩展数据帧。</p> <p>在 PDO 无效时更改才能生效。</p>

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	255
<b>Description</b>	PDO 传输类型

## 1A00h – TPDO 映射参数 1(Transmit PDO Mapping Parameter 1)

<b>Index</b>	1A00h
<b>Object code</b>	RECORD
<b>Data type</b>	—

<b>Description</b>	设定 TPDO1 映射对象。 在 PDO 无效时更改才能生效。
--------------------	------------------------------------

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6041 0010h
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6064 0020h
<b>Description</b>	映射对象 2

## 1A01h – TPDO 映射参数 2(Transmit PDO Mapping Parameter 2)

<b>Index</b>	1A01h
<b>Object code</b>	RECORD
<b>Data type</b>	—

<b>Description</b>	设定 TPDO2 映射对象。 在 PDO 无效时更改才能生效。
--------------------	------------------------------------

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	6061 0008h
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	606C 0020h
<b>Description</b>	映射对象 2

### 1A02h – TPDO 映射参数 3(Transmit PDO Mapping Parameter 3)

<b>Index</b>	1A02h
<b>Object code</b>	RECORD
<b>Data type</b>	—



<b>Description</b>	设定 TPDO3 映射对象。 在 PDO 无效时更改才能生效。
--------------------	------------------------------------

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	映射对象 2

## 1A03h – TPDO 映射参数 4(Transmit PDO Mapping Parameter 4)

<b>Index</b>	1A03h
<b>Object code</b>	RECORD
<b>Data type</b>	—

<b>Description</b>	设定 TPDO4 映射对象。 在 PDO 无效时更改才能生效。
--------------------	------------------------------------

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	映射对象 1

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	映射对象 2

## 7-2、厂家自定义对象

### ◆ 操作说明

CS 伺服驱动器的 CANopen 实现支持以下自定义对象：

- 2000h - 当前有效报警编号(Current valid alarm number)
- 2001h - 报警记录(Alarm Record)
- 2002h - 绝对值编码器接口(Absolute Encoder Interface)
- 2003h - 超程停机方式(Overrun Stop Option Code)

### 2000h – 当前有效报警编号(Current Valid Alarm Number)

<b>Index</b>	2000h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	伺服驱动器当前有效报警编号，和显示面板显示的报警号相同。例如，当前显示面板显示报警 A.01，则该对象的数值为 1。

### 2001h – 报警记录(Alarm Record)

<b>Index</b>	2001h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	伺服驱动器报警记录，子索引 1 保存最新的报警信息。

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	报警记录个数。 写 0 则清除所有报警记录。 写非 0 值无效。

<b>Sub-Index</b>	01h-0Ah
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	报警记录 1-10

## 2002h – 绝对值编码器接口(Absolute Encoder Interface)

<b>Index</b>	2002h
<b>Object code</b>	RECORD
<b>Data type</b>	—
<b>Description</b>	绝对值编码器操作接口

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	5
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	清除编码器内部报警，0-1 逻辑跳变有效。

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR

<b>Data type</b>	Uint8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	清除编码器多圈信息，0-1 逻辑跳变有效。

<b>Sub-Index</b>	03h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	清除编码器位置示教，0-1 逻辑跳变有效。

<b>Sub-Index</b>	04h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	Increments
<b>Default value</b>	0
<b>Description</b>	编码器单圈位置信息

<b>Sub-Index</b>	05h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	Increments
<b>Default value</b>	0
<b>Description</b>	编码器多圈位置信息

## 2003h – 超程停机方式(Overrun Stop Option Code)

<b>Index</b>	2003h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	<p>设定伺服驱动器 POT、NOT 输入信号有效时的停机方式。</p> <p>0: 减速停止, 停止后伺服 OFF;</p> <p>1: 反接制动, 停止后伺服 OFF</p> <p>2: 减速停止, 停止后零钳位;</p> <p>3: 反接制动, 停止后零钳位。</p>

## 7-2、标准设备子协议对象

### ◆ 操作说明

CS 伺服驱动器的 CANopen 实现支持以下标准设备子协议对象：

- 603F - 错误码(Error Code)
- 6040 - 控制字(Controlword)
- 6041 - 控制字(Statusword)
- 605A - 快速停止方式(Quick stop Option Code)
- 605B - 关闭停止方式(Shutdown Option Code)
- 605C - 禁能停止方式(Disable Operation Option Code)
- 605D - 中止停止方式(Halt Option Code)
- 605E - 报警停止方式(Fault Reaction Option Code)
- 6060 - 操作模式(Operation Mode)
- 6061 - 当前有效操作模式(Operation Mode Display)
- 6062 - 用户给定位置(Position Demand Value)
- 6063 - 电机实际内部位置(Position actual internal value)
- 6064 - 电机实际位置(Position Actual Value)
- 6065 - 跟随误差窗口(Following Error Window)
- 6066 - 跟随误差超时时间(Following Error Time Out)
- 6067 - 位置窗口(Position Window)
- 6068 - 位置窗口时间(Position Window Time)
- 6069 - 速度传感器实际值(Velocity Sensor Actual Value)
- 606B - 用户给定速度(Velocity Demand Value)
- 606C - 电机实际速度(Velocity Actual Value)
- 606D - 速度窗口(Velocity Window)
- 606E - 速度窗口时间(Velocity Window Time)
- 606F - 速度门限(Velocity Threshold)
- 6070 - 速度门限时间(Velocity Threshold Time)
- 6071 - 目标扭矩(Target Torque)
- 6072 - 最大扭矩(Max Torque)
- 6073 - 最大电流(Max Current)
- 6074 - 给定扭矩(Torque Demand)
- 6075 - 电机额定电流(Motor Rated Current)
- 6076 - 电机额定扭矩(Motor Rated Torque)
- 6077 - 电机实际扭矩(Torque Actual Value)
- 6078 - 电机实际电流(Current Actual Value)
- 607A - 目标位置(Target Position)
- 607B - 输入位置范围限制(Position Range Limit)
- 607C - 零点偏移量(Home Offset)
- 607D - 位置软限位(Software Position Limit)
- 607E - 指令极性(Polarity)
- 607F - 最大规划速度(Max Profile Velocity)
- 6080 - 最大电机速度(Max Motor Speed)

- 6081 - 规划速度(Profile Velocity)
- 6082 - 终点衔接速度(End Velocity)
- 6083 - 规划加速度(Profile Acceleration)
- 6084 - 规划减速度(Profile Deceleration)
- 6085 - 快速停止减速度(Quick Stop Deceleration)
- 6086 - 位置规划曲线类型(Motion Profile Type)
- 6087 - 扭矩变化率(Torque Slope)
- 6088 - 扭矩规划曲线类型(Torque Profile Type)
- 608F - 位置编码器分辨率(Position Encoder Resolution)
- 6091 - 齿轮传动比(Gear Ratio)
- 6092 - 进给常数(Feed Constant)
- 6098 - 回零方式(Homing Method)
- 6099 - 回零速度(Homing Speed)
- 609A - 回零加速度(Homing Acceleration)
- 60C0 - 位置插值模式(Interpolation Sub Mode Select)
- 60C1 - 插值数据记录(Interpolation Data Record)
- 60C2 - 插值周期(Interpolation Time Period)
- 60C5 - 最大加速度(Max Acceleration)
- 60C6 - 最大减速度(Max Deceleration)
- 60F4 - 偏差实际值(Following Error Actual Value)
- 60FA - 位置环输出(Control Effort)
- 60FC - 内部给定位置(Position Demand Internal Value)
- 60FD - 数字输入(Digital Inputs)
- 60FE - 数字输出(Digital Outputs)
- 60FF - 目标速度(Target Velocity)
- 6502 - 可选操作模式(Supported Drive Modes)

## 603F – 错误码(Error Code)

<b>Index</b>	603Fh
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	伺服驱动器最新的错误码，与 1003h-sub01h 的低 16 位相同。



**6040 – 控制字(Controlword)**

<b>Index</b>	6040h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	用户控制指令接口

**6041 – 控制字(Statusword)**

<b>Index</b>	6041h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	驱动器状态指示接口

**605A – 快速停止方式(Quick stop Option Code)**

<b>Index</b>	605Ah
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	设置快速停机方式

**605B – 关闭停止方式(Shutdown Option Code)**

<b>Index</b>	605Bh
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	设置关闭停机方式

**605C – 禁能停止方式(Disable Operation Option Code)**

<b>Index</b>	605Ch
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	设置禁能停机方式

**605D – 中止停止方式(Halt Option Code)**

<b>Index</b>	605Dh
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	1
<b>Description</b>	设置中止/挂起停机方式

**605E – 报警停止方式(Fault Reaction Option Code)**

<b>Index</b>	605Eh
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	设置报警停机方式

**6060 – 操作模式(Operation Mode)**

<b>Index</b>	6060h
<b>Object code</b>	VAR
<b>Data type</b>	Int8
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	1(Profile position)
<b>Description</b>	设置操作模式

**6061 – 当前有效操作模式(Operation Mode Display)**

<b>Index</b>	6061h
<b>Object code</b>	VAR
<b>Data type</b>	Int8
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	1(Profile position mode)
<b>Description</b>	显示当前有效操作模式

**6062 – 用户给定位置(Position Demand Value)**

<b>Index</b>	6062h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	使用用户单位显示位置环给定

**6063 – 电机实际内部位置(Position Actual Internal Value)**

<b>Index</b>	6063h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	用驱动器内部单位显示电机实际位置

**6064 – 电机实际位置(Position Actual Value)**

<b>Index</b>	6064h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	—
<b>Description</b>	位置传感器测得的电机实际位置

**6065 – 跟随误差窗口(Following Error Window)**

<b>Index</b>	6065h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined position units
<b>Default value</b>	30000
<b>Description</b>	设置允许的最大位置跟随误差

**6066 – 跟随误差超时时间(Following Error Time Out)**

<b>Index</b>	6066h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	ms
<b>Default value</b>	200
<b>Description</b>	设置跟随误差超时时间。当跟实际跟随误差超过 6065h 且持续时间达到 6066h，则状态字 bit13 置 1。

**6067 – 位置窗口(Position Window)**

<b>Index</b>	6067h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined position units
<b>Default value</b>	50
<b>Description</b>	设置定位完成可接受的位置偏差范围

**6068 – 位置窗口时间(Position Window Time)**

<b>Index</b>	6066h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	ms
<b>Default value</b>	20
<b>Description</b>	设置定位完成时实际位置和目标位置在位置窗口范围内保持的最小时间

**6069 – 速度传感器实际值(Velocity Sensor Actual Value)**

<b>Index</b>	6066h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	rpm
<b>Default value</b>	—
<b>Description</b>	速度传感器测得的电机实际速度

**606B – 用户给定速度(Velocity Demand Value)**

<b>Index</b>	6066h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	—
<b>Description</b>	速度规划器产生的速度环给定

**606C – 电机实际速度(Velocity Actual Value)**

<b>Index</b>	606Ch
--------------	-------

<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	—
<b>Description</b>	电机实际速度

## 606D – 速度窗口(Velocity Window)

<b>Index</b>	606Dh
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	2000
<b>Description</b>	设定速度到达窗口范围

## 606E – 速度窗口时间(Velocity Window Time)

<b>Index</b>	606Eh
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	ms
<b>Default value</b>	10
<b>Description</b>	设定速度到达窗口有效持续时间

## 606F – 零速门限(Velocity Threshold)

<b>Index</b>	606Fh
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW

<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	0
<b>Description</b>	设置用于判断用户速度是否为 0 的阈值

## 6070 – 零速时间窗口(Velocity Threshold Time)

<b>Index</b>	6070h
<b>Object code</b>	VAR
<b>Data type</b>	Uint16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	ms
<b>Default value</b>	0
<b>Description</b>	设置用于判断用户速度是否为 0 的时间窗口

## 607A – 目标位置(Target Position)

<b>Index</b>	607Ah
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined position units
<b>Default value</b>	—
<b>Description</b>	设置 PP 模式下伺服驱动器的目标位置

## 607B – 输入位置范围限制(Position Range Limit)

<b>Index</b>	607Bh
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined position units
<b>Default value</b>	—



<b>Description</b>	设置位置输入的数值范围，超范围则输入数值翻转
--------------------	------------------------

## 607C – 零点偏移量(Home Offset)

<b>Index</b>	607Ch
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined position units
<b>Default value</b>	—
<b>Description</b>	设置机械零点偏离电机原点的物理位置。

## 607D – 位置软限位(Software Position Limit)

<b>Index</b>	607Dh
<b>Object code</b>	ARRAY
<b>Data type</b>	Int32
<b>Description</b>	设置软件绝对位置限制的最小值与最大值

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined position units
<b>Default value</b>	-7FFF FFFFh

<b>Description</b>	设置最小软件绝对位置限制
<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined position units
<b>Default value</b>	7FFF FFFFh
<b>Description</b>	设置最大软件绝对位置限制

### 607E – 指令极性(Polarity)

<b>Index</b>	607Eh
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	设置位置指令或者速度指令的极性

### 607F – 最大规划速度(Max Profile Velocity)

<b>Index</b>	607Fh
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	0
<b>Description</b>	设置用户最大运行速度

### 6080 – 最大电机速度(Max Motor Speed)

<b>Index</b>	6080h
--------------	-------

<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	rpm
<b>Default value</b>	依据电机型号自动设定
<b>Description</b>	电机最大运行速度

## 6081 – 规划速度(Profile Velocity)

<b>Index</b>	6081h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	0
<b>Description</b>	设置 PP 模式下该段位移指令的匀速运行速度

## 6082 – 终点衔接速度(End Velocity)

<b>Index</b>	6082h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined velocity units
<b>Default value</b>	0
<b>Description</b>	设置 PP 模式下定位完成时的运行速度

## 6083 – 规划加速度(Profile Acceleration)

<b>Index</b>	6083h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW

<b>PDO mapping</b>	YES
<b>Units</b>	user-defined acceleration units
<b>Default value</b>	1000000
<b>Description</b>	设置 PP 模式下该段位移指令加速段的加速度

## 6084 – 规划减速度(Profile Deceleration)

<b>Index</b>	6084h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined acceleration units
<b>Default value</b>	1000000
<b>Description</b>	设置 PP 模式下该段位移指令减速段的加速度

## 6085 – 快速停止减速度(Quick Stop Deceleration)

<b>Index</b>	6085h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined acceleration units
<b>Default value</b>	1000000
<b>Description</b>	设置快速停机时的减速度

## 6086 – 位置规划曲线类型(Motion Profile Type)

<b>Index</b>	6086h
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	0

<b>Description</b>	设置电机位置指令或速度指令的曲线类型
--------------------	--------------------

## 608F – 位置编码器分辨率(Position Encoder Resolution)

<b>Index</b>	608Fh
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	编码器分辨率

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	pls
<b>Default value</b>	根据电机型号自动配置
<b>Description</b>	编码器增量计数

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RO
<b>PDO mapping</b>	NO
<b>Units</b>	round
<b>Default value</b>	根据电机型号自动配置
<b>Description</b>	电机旋转圈数

## 6091 – 齿轮传动比(Gear Ratio)

<b>Index</b>	6091h
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	设置齿轮传动比

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	round
<b>Default value</b>	1
<b>Description</b>	电机旋转圈数

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	round
<b>Default value</b>	1
<b>Description</b>	负载旋转圈数

## 6092 – 进给常数(Feed Constant)

<b>Index</b>	6092h
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	设置进给常数

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined position units
<b>Default value</b>	1
<b>Description</b>	进给量

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	round
<b>Default value</b>	1
<b>Description</b>	负载旋转圈数

**6098 – 回零方式(Homing Method)**

<b>Index</b>	6098h
<b>Object code</b>	VAR
<b>Data type</b>	Int8
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	1
<b>Description</b>	设置原点回零方式

**6099 – 回零速度(Homing Speed)**

<b>Index</b>	6099h
<b>Object code</b>	ARRAY
<b>Data type</b>	UInt32
<b>Description</b>	设置回零速度

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined velocity units
<b>Default value</b>	50000
<b>Description</b>	搜索原点开关速度



<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined velocity units
<b>Default value</b>	1000
<b>Description</b>	搜索零点速度

## 609A – 回零加速度(Homing Acceleration)

<b>Index</b>	609Ah
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined acceleration units
<b>Default value</b>	2000000
<b>Description</b>	设置回零动作加速度和减速度

## 60C0 – 位置插值模式(Interpolation Sub Mode Select)

<b>Index</b>	60C0h
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	2000000
<b>Description</b>	设置 IP 模式下位置插值方式

## 60C1 – 插值数据记录(Interpolation Data Record)

<b>Index</b>	60C1h
<b>Object code</b>	ARRAY

<b>Data type</b>	Int32
<b>Description</b>	设置 IP 模式下位置指令

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	1
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	user-defined velocity units
<b>Default value</b>	50000
<b>Description</b>	位置指令

## 60C2 – 插值周期(Interpolation Time Period)

<b>Index</b>	60C1h
<b>Object code</b>	Record
<b>Data type</b>	Interpolation time period record(0080h)
<b>Description</b>	设置 IP 模式下位置指令插值周期

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	
<b>Default value</b>	2

<b>Description</b>	可访问接口数
--------------------	--------

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	UInt8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	1
<b>Description</b>	插值周期时间值

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Int8
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	-3
<b>Description</b>	插值周期时间单位

## 60C5 – 最大加速度(Max Acceleration)

<b>Index</b>	60C5h
<b>Object code</b>	VAR
<b>Data type</b>	Int16
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	设置最大运行加速度

## 60C6 – 最大减速度(Max Deceleration)

<b>Index</b>	60C6h
<b>Object code</b>	VAR
<b>Data type</b>	Int16

<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	设置最大运行减速度

## 60F4 – 偏差实际值(Following Error Actual Value)

<b>Index</b>	60F4h
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	user-defined position units
<b>Default value</b>	—
<b>Description</b>	实时反映位置偏差

## 60FC – 内部给定位置(Position Demand Internal Value)

<b>Index</b>	60FCh
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	pls
<b>Default value</b>	—
<b>Description</b>	实时反映电机位置指令

## 60FD – 数字输入(Digital Inputs)

<b>Index</b>	60FDh
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	—

<b>Default value</b>	—
<b>Description</b>	反映驱动器当前输入端子逻辑

## 60FE – 数字输出(Digital Outputs)

<b>Index</b>	60FEh
<b>Object code</b>	ARRAY
<b>Data type</b>	Uint32
<b>Description</b>	设置或读取数字输入口的逻辑状态

<b>Sub-Index</b>	00h
<b>Object code</b>	VAR
<b>Data type</b>	Uint8
<b>Access</b>	RO/CONST
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	2
<b>Description</b>	可访问接口数

<b>Sub-Index</b>	01h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0
<b>Description</b>	物理输出逻辑状态

<b>Sub-Index</b>	02h
<b>Object code</b>	VAR
<b>Data type</b>	Uint32
<b>Access</b>	RW
<b>PDO mapping</b>	NO
<b>Units</b>	—
<b>Default value</b>	0

<b>Description</b>	位屏蔽码
--------------------	------

## 60FF – 目标速度(Target Velocity)

<b>Index</b>	60FFh
<b>Object code</b>	VAR
<b>Data type</b>	Int32
<b>Access</b>	RW
<b>PDO mapping</b>	YES
<b>Units</b>	User-defined velocity units
<b>Default value</b>	0
<b>Description</b>	设置 PV 模式下用户速度指令

## 6502 – 可选操作模式(Supported Drive Modes)

<b>Index</b>	6502h
<b>Object code</b>	VAR
<b>Data type</b>	UInt32
<b>Access</b>	RO
<b>PDO mapping</b>	YES
<b>Units</b>	—
<b>Default value</b>	0000 0065h
<b>Description</b>	伺服驱动器支持的操作模式

## 8 · 故障及处理

8.1 获取故障信息

8.2 故障处理

## 8-1、获取故障信息

### ◆ 操作说明

#### 1、通过紧急帧

当内部出现异常时，伺服驱动器以生产者的形式向网络发送紧急帧。网络上的其他设备通过接收紧急

帧来获取伺服驱动器异常错误的详细信息。驱动器异常和紧急帧数据之间的关系参见 3.6 节描述。

读取对象 603Fh(Error Code) 亦可获得驱动器最新的异常信息。

#### 2、通过状态字

状态字 6041h(Statusword) 的 bit3 同样可反映伺服驱动器当前是否处于异常状态。具体的报警编号可通过

读取对象 2000h(Current Valid Alarm Number) 获得。

## 8-2、故障处理

### ◆ 操作说明

#### 1、一般性故障

伺服驱动器一般性故障是指和 CANopen 通讯无关的故障类别。对于可复位报警，允许通过操作面板或是控制字对其进行复位。

显示	报警名称	错误码		
		EMCY	603Fh	2001h
A. 01	参数破坏	6311h	6311h	01h
A. 03	超速	8010h	8010h	03h
A. 04	过载	3210h	3210h	04h
A. 05	位置偏差计数器溢出	8200h	8200h	05h
A. 06	位置偏差脉冲溢出	8201h	8201h	06h
A. 07	电子齿轮设置和给定脉冲频率配置不合理	6312h	6312h	07h
A. 08	电流检测第一通道有问题	5010h	5010h	08h
A. 09	电流检测第二通道有问题	5011h	5011h	09h
A. 10	增量编码器 ABC 断线	5012h	5012h	0Ah
A. 11	增量编码器 UVW 断线	5013h	5013h	0Bh



显示	报警名称	错误码		
		EMCY	603Fh	2001h
A. 12	过流	2310h	2310h	0Ch
A. 13	过压	3211h	3211h	0Dh
A. 14	欠压	3212h	3212h	0Eh
A. 15	泄放电阻损坏	5014h	5014h	0Fh
A. 16	再生异常	5015h	5015h	10h
A. 18	IGBT 过热报警	4210h	4210h	12h
A. 20	电源线缺相	3213h	3213h	14h
A. 21	瞬间停电报警	3214h	3214h	15h
A. 23	制动过流报警	2311h	2311h	17h
A. 42	电机型号错	7101h	7101h	2Ah
A. 43	伺服驱动器/编码器型号错	7102h	7102h	2Bh
A. 50	串行编码器通讯超时	5014h	5014h	32h
A. 51	串行编码器通讯校验和错误	5015h	5015h	33h
A. 52	串行编码器通讯控制域奇偶位、截止位错误	5016h	5016h	34h
A. 53	绝对值编码器检测到超速报警	5017h	5017h	35h
A. 54	串行编码器绝对状态出错	5018h	5018h	36h
A. 55	绝对值编码器单圈信息错误	5019h	5019h	37h
A. 56	绝对值编码器多圈信息溢出	501Ah	501Ah	38h
A. 57	绝对值编码器多圈信息错误	501Bh	501Bh	39h
A. 58	电池电压低于 3.1v	501Ch	501Ch	3Ah
A. 59	电池电压低于 2.5v	501Dh	501Dh	3Bh

## 2、CANopen 相关故障

CANopen 相关故障的描述如下表所示。对于可复位报警，允许通过操作面板或是控制字对其进行复位。

表 8.2 CANopen 相关故障描述

显示	报警名称	错误码		
		EMCY	603Fh	2001h
A. 66	CAN 通讯异常	8020h	8020h	38h
A. 67	同步帧 SYNC 异常	8030h	8030h	39h
A. 68	单位转换因子设置错误	6312h	6312h	40h
A. 69	心跳超时	8130h	8130h	41h

# 9 · 应用示例—基于 Beckhoff 控制器

## 9.1 基本配置

## 9.2 伺服驱动器设置

### 9.3.1 控制器和 PC 通讯

### 9.3.2 控制器和伺服驱动器通讯

### 9.3.3 SDO 配置

### 9.3.4 变量连接

## 9-1、基本配置

### ◆ 操作说明

- 1、Beckhoff ([www.beckhoff.com](http://www.beckhoff.com)) 控制器及配套软件  
CX1020-0123  
EL6751  
TwinCAT (V2. 11. 2234)
- 2、CS 伺服驱动器
- 3、配套线缆及终端电阻

## 9-2、伺服驱动器设置

### ◆ 操作说明

- 1、设置控制方式 Pn004.0 为 CANopen 总线控制 (0x0F)。
- 2、设置节点号 Pn074 为 1。
- 3、设置通讯波特率 Pn075 为 5, 1Mbit/s。
- 4、伺服驱动器重新上电。

## 9-3-1、控制器和 PC 通讯

### ◆ 操作说明

- 1、在 TwinCAT 软件左侧导航栏中选中 **SYSTEM - Configuration**。然后在右侧窗口中选中标签 **Version (Local)**，单击 **Choose Target...**

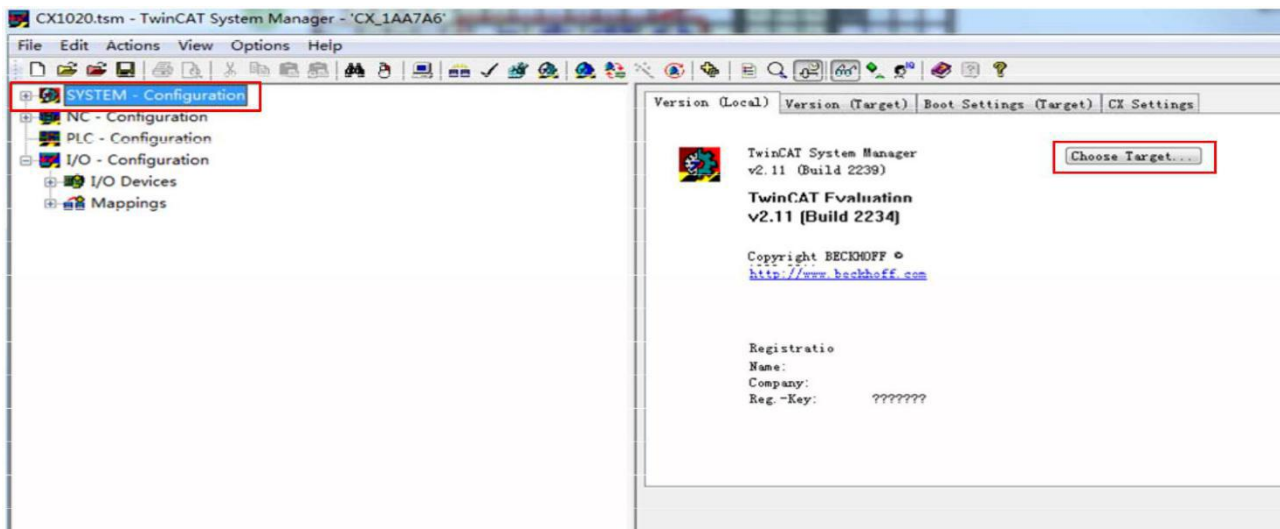


图 9.1

2、点击 **Search(Ethernet)**，搜索网络中的控制器。

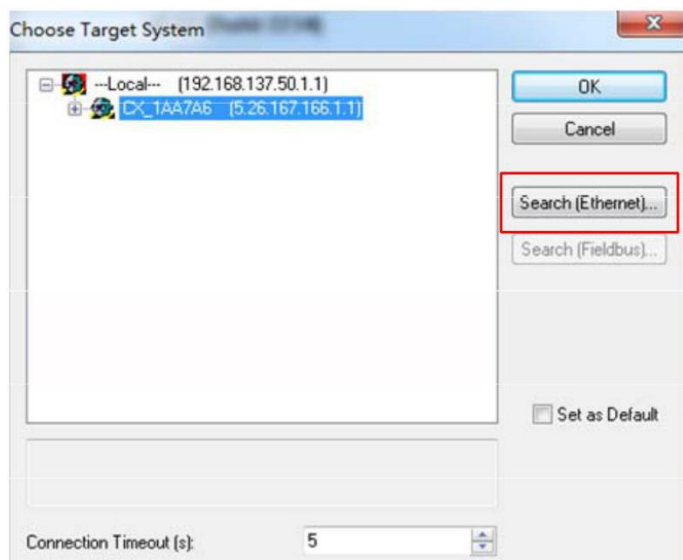
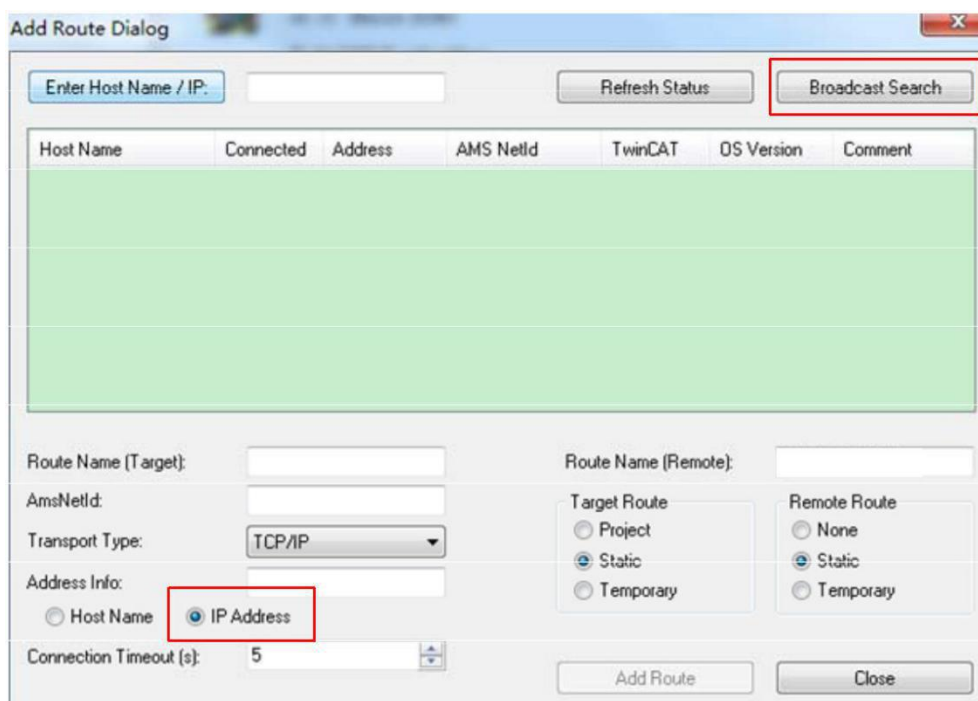
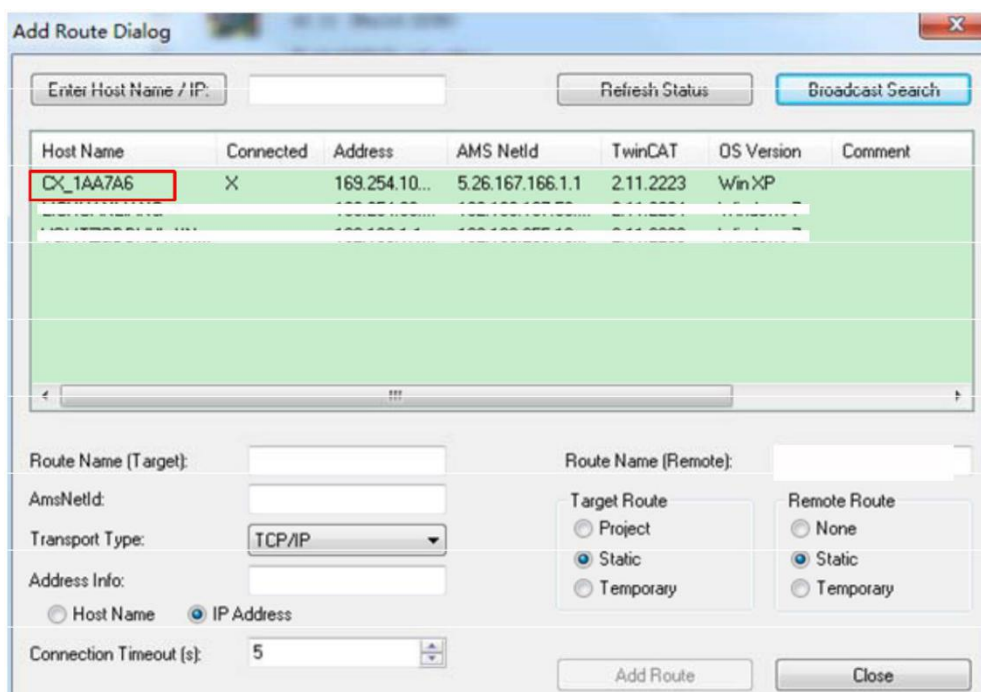


图 9.2

3、选中 **IPAddress**，然后单击 **Broadcast Search**。等待控制器名称出现在列表中，表示搜索完成。





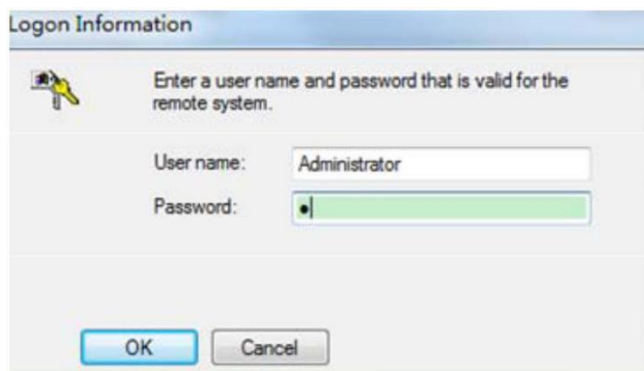
4、选中列表中的控制器名称，单击 **Add Route**。

5、在登录对话框中输入：

User Name: Administrator

Password: 1

单击 **OK**。



7、在 Add Route 对话框中确保标识符“X”出现在控制器名称的右侧，表示 PC 和控制之间的通讯已经建立。

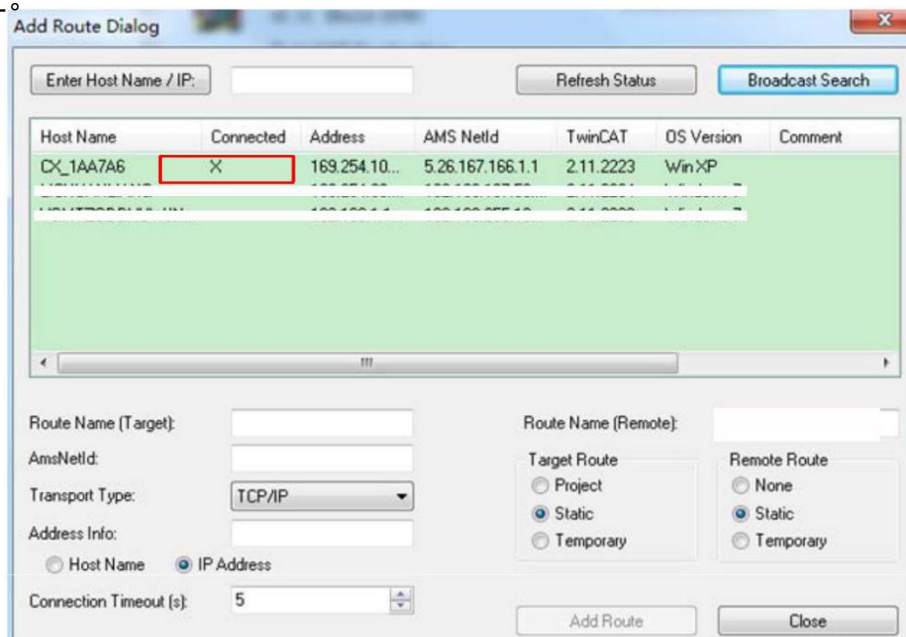
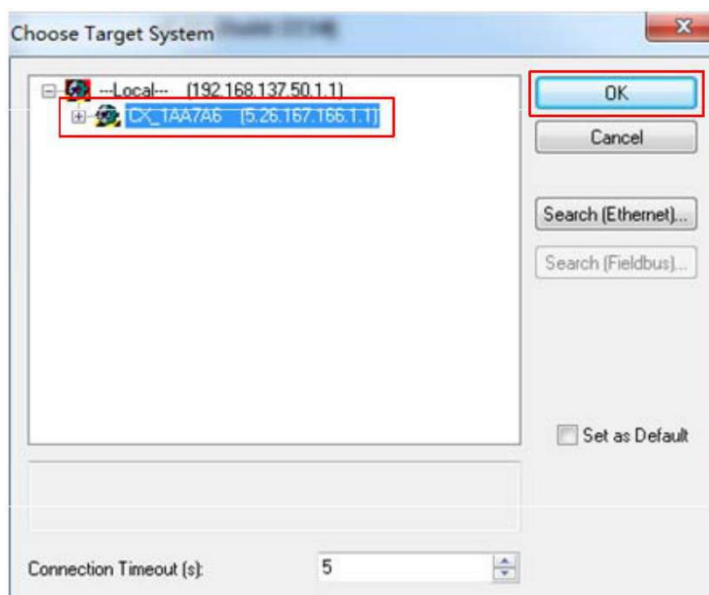


图 9.6

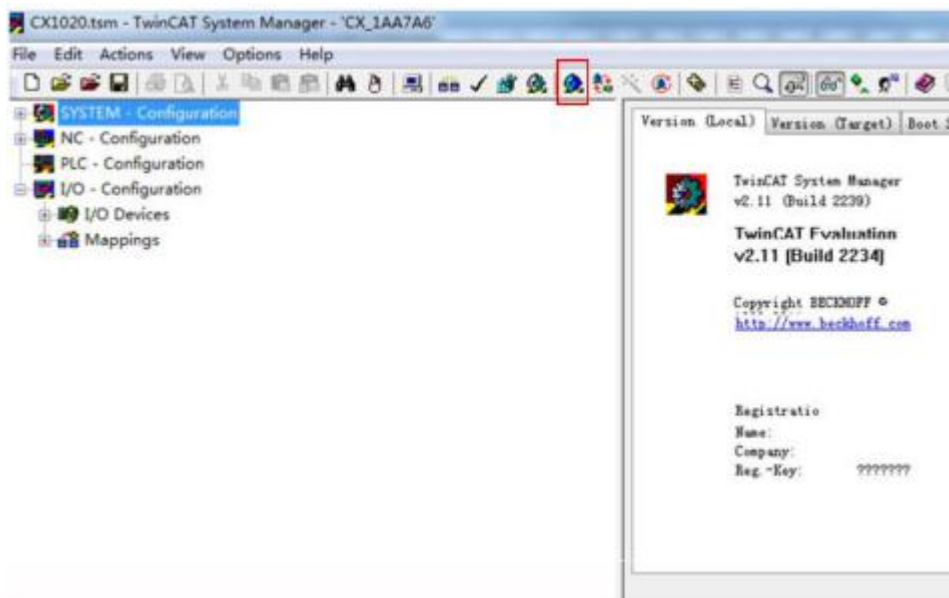
6、在 Choose Target System 对话框中，选中控制器名称，单击 OK。



## 9-3-2、控制器和伺服驱动器通讯

### ◆ 操作说明

1、单击 **Set/Reset TwinCAT Config Mode**, 确保 TwinCAT 处于 **Config Mode**。

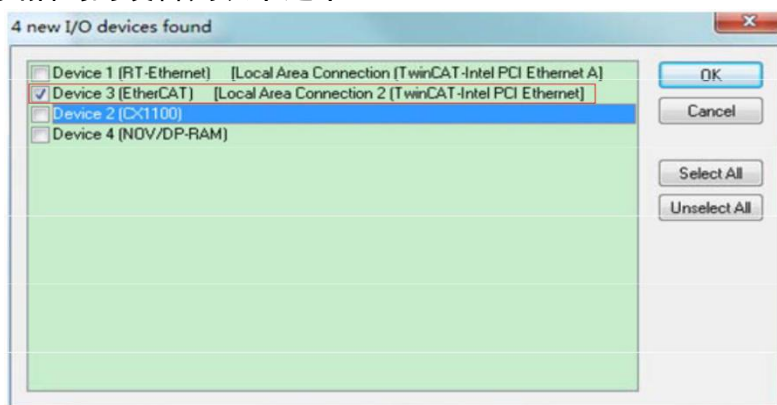


8、在左侧导航栏中, 展开 **I/O - Configuration**, 选中 **I/O Devices** 然后点击右键, 选择 **Scan devices**。

9、警告窗口, 单击确定。



10、在扫描到的设备列表中选中 “EtherCAT”





单击 **OK**。

11、搜索 BOX，单击是。



图 9.10

12、建立 CANopen 主站，单击是。



图 9.11

13、设置通讯波特率。单击 **OK** 开始搜索 BOX。

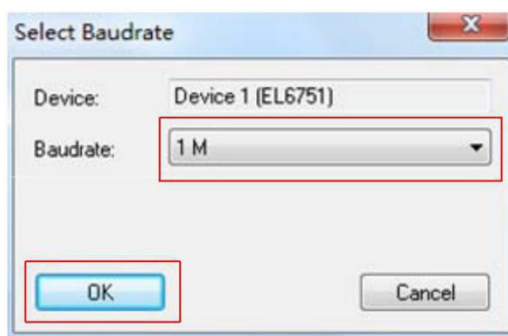


图 9.12

14、单击是，使能自由运行。



图 9.13

15、BOX 搜索完毕，在导航栏显示搜索结果，搜索到的驱动器名称为“CS。

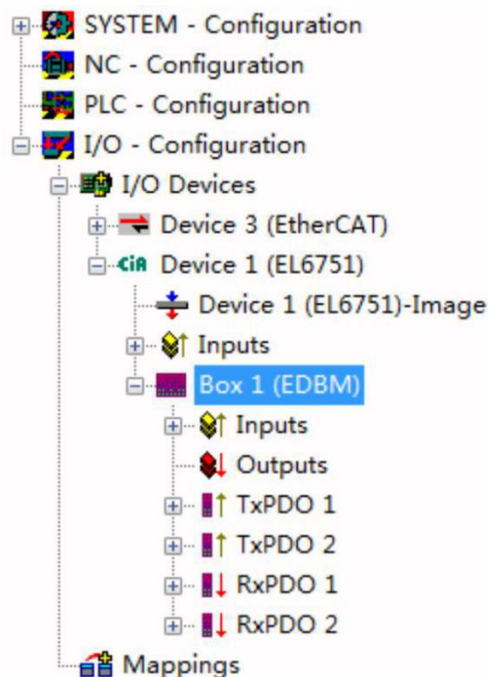


图 9.14

### 9-3-3、SDO 配置

#### ◆ 操作说明

SDO 配置用于完成控制所需对象的初始化，包括 PDO 参数、加减速度等。

#### 1、固定配置

在导航栏选中 BOX，在右侧窗口的 SDO 标签中，单击 **Append**，输入需要配置的 SDO。使用这种方式配置的 SDO 在控制器进入 Run Mode 后自动写入。

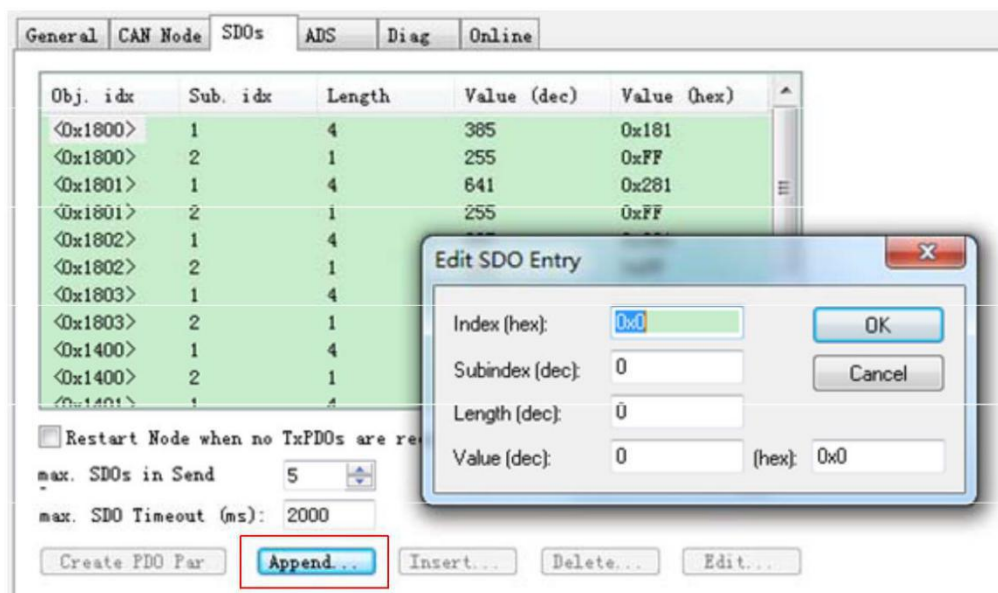


图 9.15

## 2、实时配置

在导航栏选中 BOX，然后在右侧窗口中，选择 Online 标签，点击 **Advanced**。

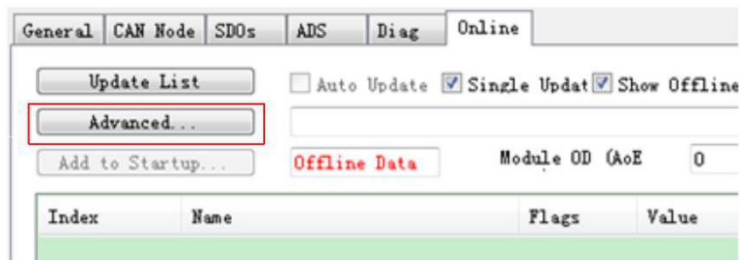


图 9.16

选择 EDS 文件所在目录并导入。

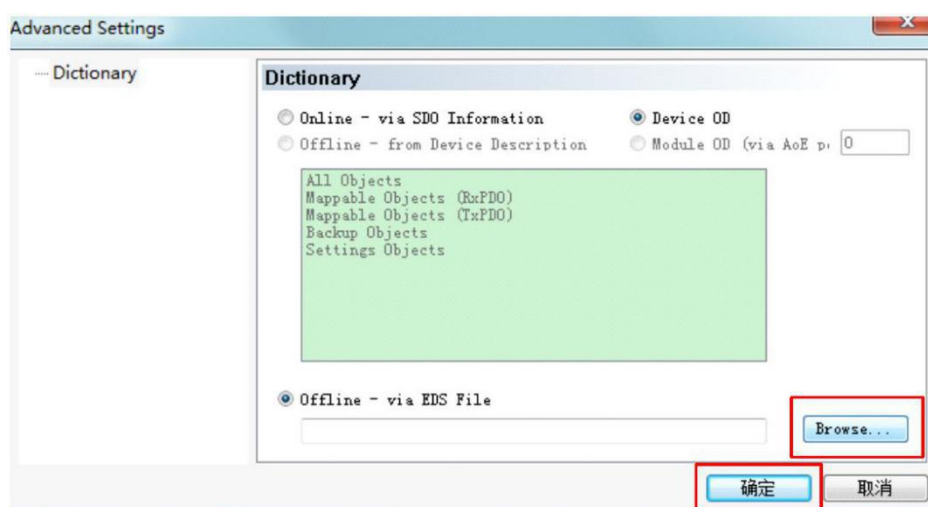


图 9.17

Value 列反应对象的实时数据值。双击需要更改的对象，输入配置信息，即可完成 SDO 写入。

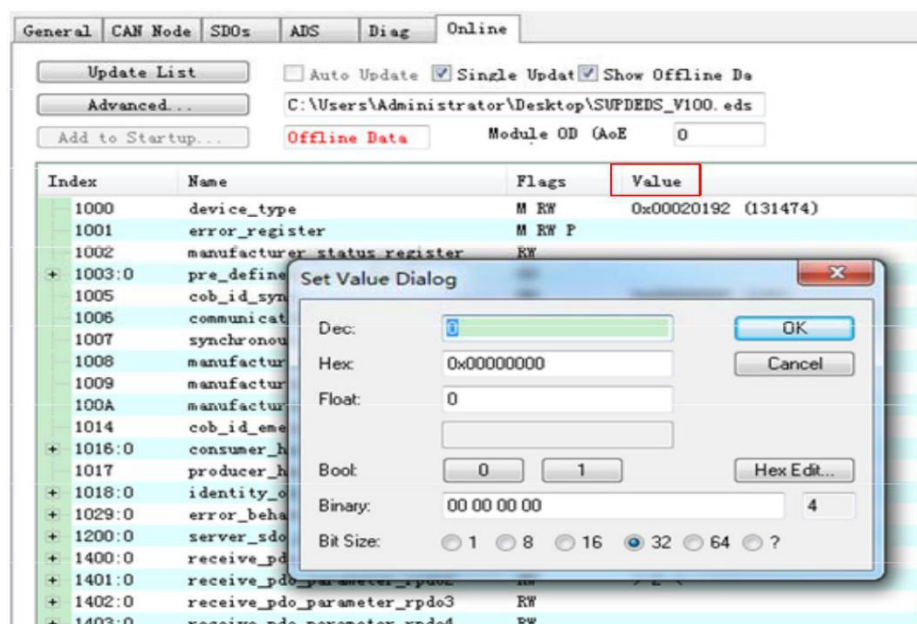


图 9.18

### 9-3-4、变量连接

#### ◆ 操作说明

选中需要连接变量的 PDO。

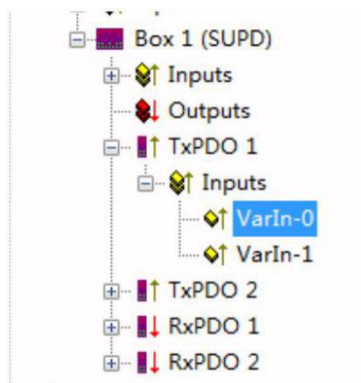
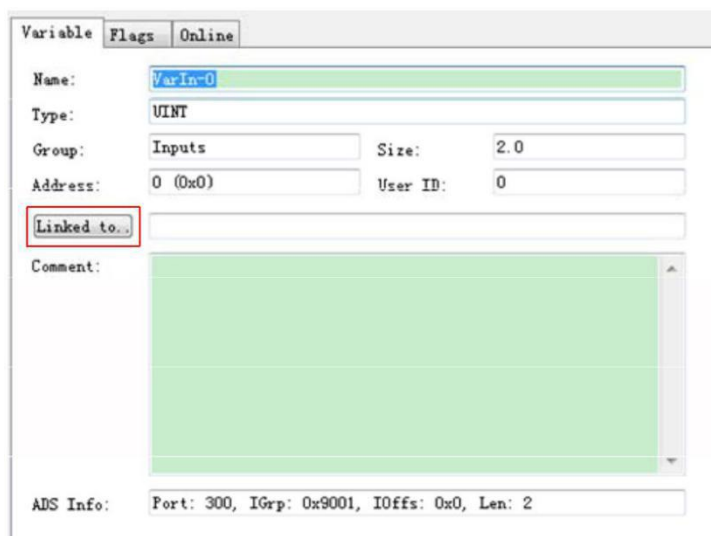


图 9.19

在右侧窗口中，将 PDO 连接至指定的控制变量。



PDO 和控制变量连接完成后，即可使用相关变量在 NC Task 或 PLC Task 中完成对伺服驱动器的控制。

NC 和 PLC 的使用方法请参考 Beckhoff 相关手册。

