



# CANopen通讯软件使用手册

# 版权申明

---

感谢您购买北京和利时电机技术有限公司的 SE 系列全数字步进驱动器（以下简称驱动器）产品。SE 系列全数字步进驱动器是以美国 TI 公司最新的数字处理芯片 (DSP) 作为核心控制芯片, 采用了先进的全数字式电机控制算法, 完全以软件方式实现了电流环控制, 具备良好的鲁棒性和自适应能力, 可配合多种规格的步进电机, 实现速度、力矩和位置高精度、高响应的控制, 适应于需要快速响应的精密转速控制与定位控制的应用系统, 如: 医疗机械、印刷机械、包装机械、造纸机械、塑料机械、纺织机械、工业机器人、自动化生产线等。

本用户手册是针对 SE 系列全数字步进驱动器的通信手册。在本手册中, 详细地说明了驱动器的串行通信 CANopen 总线通信协议和使用说明, 以此来帮助用户建立上位控制器与驱动器的通信连接。在使用 SE 系列驱动器的通信功能之前, 请仔细阅读本用户手册, 以保证正确使用。

# 目录

---

<b>第一章</b>	<b>通信功能简介 .....</b>	<b>1</b>
<b>第二章</b>	<b>CANopen 协议 .....</b>	<b>2</b>
2.1.	设置 .....	2
2.1.1.	设置站址 .....	2
2.1.2.	设置通信波特率 .....	2
2.2.	CANopen 通信规范 .....	3
2.3.	通信对象标识符地址分配 .....	3
2.4.	通信对象 .....	4
2.4.1.	Network Management Objects (NMT) .....	5
2.4.2.	Synchronization Object (SYNC) .....	5
2.4.3.	Emergency Object (EMCY) .....	5
2.4.4.	Process Data Object (PDO) .....	7
2.4.5.	Service Data Object (SDO) .....	12
2.4.6.	Nodeguard .....	15
2.4.7.	Heartbeat.....	16
2.4.8.	Bootup.....	17
2.5.	网络初始化和系统 Bootup.....	18
2.5.1.	初始化流程 .....	18
2.5.2.	NMT 状态机.....	19
2.5.3.	设备状态和通讯对象的联系 .....	21
2.5.4.	设备状态转换 .....	21
2.5.5.	Bootup.....	22
<b>第三章</b>	<b>CANopen 设备规范 .....</b>	<b>23</b>
3.1	PDO 映射.....	23

# 目录

---

3.1.1	RPDO 映射 .....	23
3.1.2	TPDO 映射 .....	23
3.2	设备控制 .....	24
3.2.1	状态机 .....	24
3.2.1.1	状态转换 .....	26
3.2.2	对象描述 .....	26
3.2.2.1	对象 6040h: 控制字 .....	27
3.2.2.2	对象 6041h: 状态字 .....	28
3.2.2.3	对象 6060h: 操作模式 .....	30
3.2.2.4	对象 6061h: 操作模式显示 .....	30
3.2.3	协议位置模式 (Profile Position Mode) .....	30
3.2.3.1	对象 607Ah: 目标位置 .....	31
3.2.3.2	对象 6081h: 协议速度 .....	31
3.2.3.3	对象 6083h: 协议加速度 .....	31
3.2.3.4	对象 6084h: 协议减速度 .....	32
3.2.3.5	功能描述 .....	32
3.2.4	速度通讯模式 .....	34
3.2.4.1	对象 60FF: 给定速度 .....	34
3.2.4.2	对象 6083h: 协议加速度 .....	34
3.2.4.3	对象 6084h: 协议减速度 .....	35
3.2.5	周期位置模式 .....	35
3.2.5.1	对象 607Ah: 目标位置 .....	35
3.2.5.2	对象 60C2h: 周期描述 .....	36
3.2.6	回原点模式 .....	36
3.3	使用举例 .....	38

# 目录

---

3.3.1	设备控制操作举例 .....	38
3.3.2	PDO 使用举例.....	39
3.3.3	SDO 读写对象词典对象、驱动器内部参数对象及保存、恢复默认参数举例 ..	40
3.3.4	协议位置模式使用举例 .....	41
3.3.5	速度通讯模式使用举例 .....	43
3.3.6	回原点模式使用举例 .....	44
3.4	<b>对象词典描述</b> .....	46
3.4.1	强制性对象 .....	46
3.4.2	任意对象 .....	46
3.4.3	设备协议对象 .....	50
3.4.4	设备商定义对象 .....	51

## 第一章 通信功能简介

森创 SE 系列步进驱动器提供了与上位控制器的标准串行通信 CAN 总线通信硬件接口，可以实现编辑驱动器功能参数、监视运行状态和在线控制电机运转等功能，端口接线方式请参照相应产品说明书中的连线说明章节。

森创 SE 系列步进驱动器的串行通信方式目前支持 ModBus 和 CANopen 总线通信协议两种通信协议，其中 ModBus 协议支持 ModBus-RTU 模式，详见相应的通信手册，下面详细介绍 CANopen 通信协议。

# CANopen 协议

## 第二章 CANopen 协议

通信特征基于:

- (1) CAN 2.0A
- (2) CANopen 技术规范 (DS301 V4.02)  
功能特征符合: 变频器和运动控制设备 (DSP-402 V2.0)

### 2.1. 设置

#### 2.1.1. 设置站址

这里以 SE-20403C 为例, 其他产品略有区别, 原理相同, 具体可详见相应的产品说明书。

有两种方法可设置站址, 通过 SW4 确定方式:

方式一: SW4=OFF, 站址通过 SW1-SW3 来确定站址 (范围 1-8), 通讯方式设置站址无效

站址=SW1+SW2\*2+SW3\*4+1, SWN=ON 该值为 1, SWN=OFF, 该值为 0

举例:

站址	SW1	SW2	SW3
1	OFF	OFF	OFF
8	ON	ON	ON

方式二: SW4=ON, 通讯设置站址, 通过 SDO 设置对象索引为 2000 子索引 3 (范围 1-127) 的数值

\*设置完后需要重新上电才有效

#### 2.1.2. 设置通信波特率

这里以 SE-20403C 为例, 其他产品略有区别, 原理相同, 具体可详见相应的产品说明书。

波特率的设置可以通过 2 种方法, 由 SW4 决定:

方式一: SW4=OFF, 通讯波特率 250K 比特/S

方式二: SW4=ON, 通讯设置波特率, 通过 SDO 设置对象索引为 2000 子索引 4 (范围 20-1000) 的数值, 支持的波特率有: 20K/50K/100K/125K/250K/500K/1000K 比特/S

# CANopen 协议

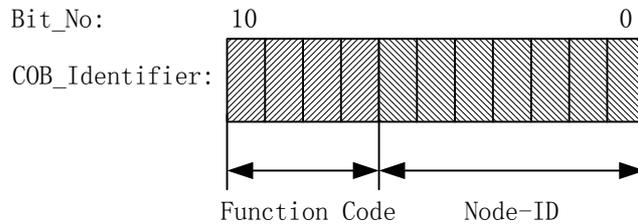
\*设置完后需要重新上电才有效

## 2.2. CANopen 通信规范

该章描述了驱动器依据 CANopen Communication Profile DS301 所提供的服务和通信功能，基本的通信规范操作可参见相关的标准协议文档。

## 2.3. 通信对象标识符地址分配

下图描述了通信对象标识符（COB-Identifier）的结构，其地址分配包含一个功能部分和一个模块 ID 部分，该功能部分决定对象的优先级，而 Node-ID 模块部分（module-ID-part）在相同功能的设备间进行区分。



下表描述了驱动器上电后通信对象标识符的默认值：

用于广播的通信对象（发送给所有的节点）

Object	Function Code	COB-Identifier	Communication Parameters at index
NMT	0000	0h	
SYNC	0001	80h	1005h

用于点对点通信的通信对象（发送给指定节点）

Object	Function Code	COB-Identifier	Communication Parameters at index	Priority
EMERGENCY	0001	81h~FFh		High
TPDO1	0011	181h~1FFh	1800h	↓
RPDO1	0100	201h~27Fh	1400h	↓
TPDO2	0101	281h~2FFh	1801h	↓

# CANopen 协议

RPDO2	0110	301h~37Fh	1401h	↓
TPDO3	0111	381h~3FFh	1802h	↓
RPDO3	1000	401h~47Fh	1402h	↓
TPDO4	1001	481h~4FFh	1803h	↓
RPDO4	1010	501h~57Fh	1403h	↓
SDO(tx)*	1011	581h~5FFh		↓
SDO(rx)*	1100	601h~67Fh		↓
Nodeguard	1110	701h~77Fh	100Eh	low

tx: 发送方向是驱动器发送给主站

rx: 发送方向是主站发送给驱动器

## 2.4. 通信对象

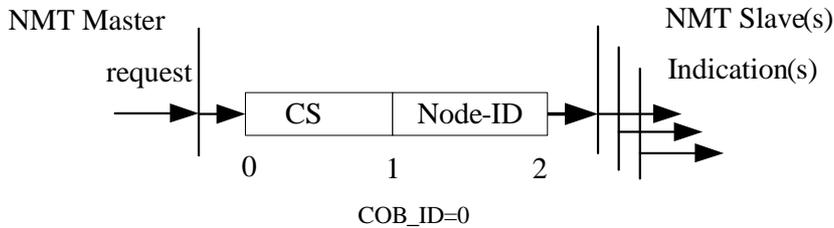
驱动器支持的通信对象如下:

- (1) Network Management Objects (NMT)
- (2) Synchronization Object (SYNC)
- (3) Emergency Object (EMCY)
- (4) Process Data Object (PDO)
- (5) Service Data Object (SDO)
- (6) Nodeguard
- (7) Heartbeat
- (8) Bootup

# CANopen 协议

## 2.4.1. Network Management Objects (NMT)

NMT 通信报文格式如下图所示：



CS: 命令符

驱动器支持下列网络管理功能：

CS=130, 节点通讯复位, 通讯协议的通讯参数载入 E2PROM 的值或默认值

CS=129, 节点复位: 设备协议和制造商指定参数, 载入 E2PROM 的值或者默认值

CS=128, 远程节点预运行: 预运行 CAN 节点

CS=1, 开始远程节点: 开始 CAN 节点

CS=2, 停止远程节点: 停止 CAN 节点

## 2.4.2. Synchronization Object (SYNC)

同步对象 (SYNC) 在网络中提供了基本的网络时钟, 由同步对象生产者周期性广播, 这个周期由标准的参数-通讯循环周期 (对象 1006h 描述) 来指定, 这个周期可以在应用设备预运行状态下, 通过 SDO 访问 1006h 来写入周期。

为了保证同步对象能够及时的使用 CAN 总线, 同步对象标识符具有较高的优先级, 对象 1005h 描述, 缺省值为 80h。同步对象不携带任何数据, 设备接收到同步对象后, 执行同步任务。在后续章节中将详细描述。

## 2.4.3. Emergency Object (EMCY)

应急对象 (EMCY) 由设备内部出现致命错误来触发, 并由该设备发送, 因此应急对象适用于中断类型的报警信号, 每个“错误事件“(error event) 只能发送一次应急对象, 只要在设备上不发生新的错误, 就不能再发送应急对象, 零个或多个应急对象消费者可接受此应急对象, 这些应急消费者的反应是由应用指定, CANopen 定义了应急对象中要传送的若干个应急错误代码, 驱动器支持的错误代码见下表所示, 每个代码代表着不同的错误。

# CANopen 协议

错误代码（十六进制）	意义
0000	没有错误
1000	一般错误
2300	过流
3110	主电源过压
3120	主电源欠压
5000	硬件故障
5530	E2 错误
7122	未接电机线

错误代码

应急对象映射到一个单一的具有 8 个数据字节的 CAN 帧，如下图所示：

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code		Error register	Manufacture specific Error Field				

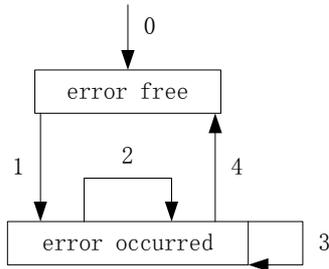
应急对象数据包

第 0-1 的字节包含错误代码上表所示；第 2 个字节是描述错误寄存器的状态，即对象 1001h（Error Register）的数值，不同的数值代表不同的错误状态；第 3-7 的字节用于制造商指定的错误描述。

当接收到应急对象后，就可以分析数据包内容，从而确定错误原因，做出相应的回应。

驱动器可能处于两种错误状态之一见下图所示：

# CANopen 协议



0 设备初始化完毕后，如果没有错误发生，则设备进入 **error free** 状态，没有错误信息发送

1 设备发现一个内部错误，错误信息在报文前三个字节体现（**Error Code** 和 **Error Register**，设备进入 **error** 状态，设备发送该 **EMCY** 报文

2 设备内部有多个错误，当其中一个错误修复后，设备则发生一个错误代码（**Error Code**）为错误复位（即 **0000h**）的 **EMCY**，其中报文 **Error Register** 部分还存有剩余错误的信息

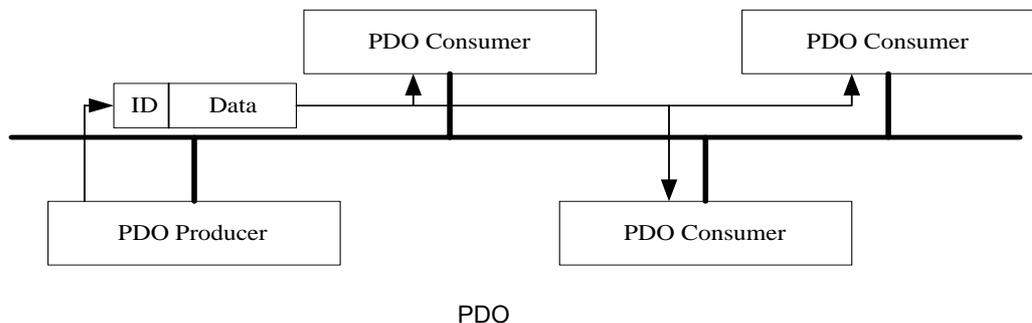
3 设备又发生了一个新的错误，设备仍然处于 **error** 状态，设备发送相关的 **EMCY**

4 当设备内部所有的错误都修复了，则设备进入 **error free** 状态，发送一个错误代码为错误复位或没有错误（即 **0000h**）的 **EMCY**。

## 2.4.4. Process Data Object (PDO)

过程数据对象即 **PDO**（**Process Data Object**）主要用来传输实时数据，例如电机速度、位置、I/O 值等。实时数据以 8 个字节封装成一个 **PDO**，**PDO** 所携带的实时数据类型和内容是由设备对象词典中所规定的该 **PDO** 映射结构决定。设备在预操作状态 (**Pre-Operational State**) 期间支持动态 **PDO** 映射，通过服务数据对象 **SDO** 对 **PDO** 映射进行配置。

每个 **PDO** 有一个唯一的标识符，仅通过一个节点发送但其接收者可不止一个，如下图所示：



PDO

# CANopen 协议

PDO 按照用途分为两种：

- (1) Transmit-PDOs (TPDOs) 用于实时数据的发送
- (2) Receive-PDOs (RPDOs) 用于实时数据的接收

一个 PDO 是由 PDO 通讯参数和 PDO 映射参数来描述, PDO 通讯参数描述了 PDO 的发送方式、触发方式等, PDO 映射参数描述了 PDO 内容方面的信息, 这些参数是由对象词典的对象来描述, 这些对象在对象词典的位置是由下面公式计算:

RPDO 通讯参数的索引 (描述该参数的对象索引) =  $1400h + (\text{RPDO 的序列号} - 1)$

TPDO 通讯参数的索引 (描述该参数的对象索引) =  $1800h + (\text{TPDO 的序列号} - 1)$

RPDO 映射参数的索引 (描述该参数的对象索引) =  $1600h + (\text{RPDO 的序列号} - 1)$

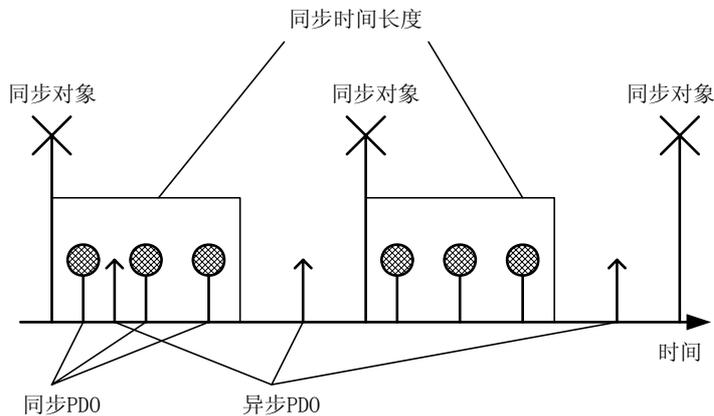
TPDO 映射参数的索引 (描述该参数的对象索引) =  $1A00h + (\text{TPDO 的序列号} - 1)$

例如第一个 TPDO (序列号为 1) 就是由通讯参数  $1800h$  ( $1800h + 1 - 1$ ) 和映射参数  $1A00h$  ( $1A00h + 1 - 1$ ) 来描述。

## 2.4.4.1 PDO 发送类型

PDO 有两种发送方式：同步发送 (Synchronous Transmission) 和异步发送 (Asynchronous Transmission)。

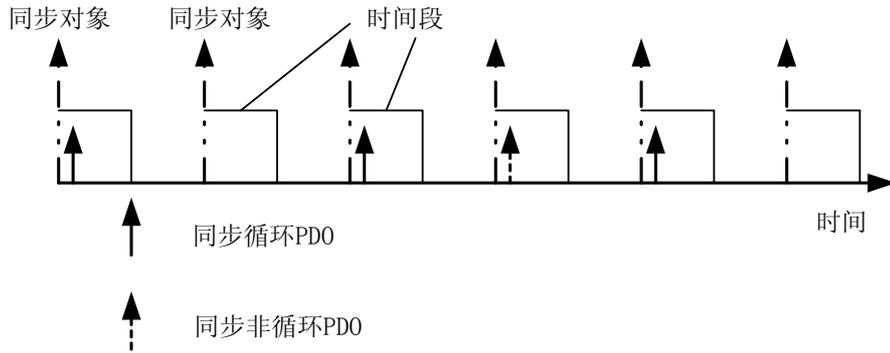
在总线上为了同步网络上的设备, 就会周期性的发送同步对象 (SYNC object)。下图表明了异步 PDO 和同步 PDO 发送原理, 同步 PDO 只有在接收到了同步对象信号后, 在预定的同步时间段内被发送, 而异步 PDO 可以随时发送。



同步和异步发送

同步 PDO 发送又可分成同步循环发送和同步非循环发送, 见下图所示:

# CANopen 协议



同步循环和非循环 PDO

同步循环性的 PDO 在同步时间段内循环被发送。

同步非循环性的 PDO 是由应用所指定的事件触发而被发送，这个信息将在接收到同步对象（SYNC object）后发送，但不是循环发送。

总的来说，一个 PDO 的发送类型如下表所示：

类型编号	循环	非循环	同步	非同步	远程请求
0		√	√		
1-240	√		√		
241-251			保留		
252			√		√
253				√	√
254				√	
255				√	

PDO 发送类型

其中编号是 1-240 的发送类型代表连续发送两个 PDO 之间所接收到的 SYNC\_Object (同步对象) 的数目；编号 252、253 是只有收到远程请求，才发送该 PDO，其中编号 252 当接收到同步对象后就立即更新 PDO 数据但并不发送，编号 253 的代表当接收到远程请求时就立即更新 PDO 数据，这种类型仅适用于 TPDO；编号 254 的发送类型代表触发 PDO 发送的应用事件是由设备商指定的；编号 255 的发送类型代表触发 PDO 发送的应用事件是由设备协议指定。

一个 PDO 的发送类型是在 PDO 通讯参数定义的，其中第一个 RPDO 和 TPDO 分别在对象 1400H 和 1800H 描述。PDO 触发方式

一个 PDO 是由某些信息触发从而被发送，触发方式主要分以下三种：

# CANopen 协议

(1) 事件触发，这是由对象指定事件发生从而导致信息发送；例如对于同步 PDO，当设备接收到同步对象后，同步 PDO 就被发送。

(2) 定时器触发，这是由设备指定的事件发生或者没有发生事件但是到了一定的时间从而触发 PDO 的发送

(3) 远程请求触发，一个异步 PDO 的发送就是由接收到另外一个设备的远程请求后而被发送。

## 2.4.4.2 PDO 通讯参数

PDO 通讯参数描述了 PDO 通讯行为，通讯参数的数据类型是对象词典中 20H 描述，通讯参数结构如下表所示：

Index	Sub-Index	Description	Data Type	.....
20h	0h	number of entries	Unsigned8	.....
	1h	COB-ID	Unsigned32	.....
	2h	transmission type	Unsigned8	.....
	3h	inhibit time	Unsigned16	.....
	4h	reserved	Unsigned8	.....
	5h	event timer	Unsigned16	.....

通讯参数结构

子索引 0 描述了该对象包含的条目（不包括子索引 0）；子索引 1h 描述了该 PDO 的通讯标识符（COB-ID）；子索引 2h 描述了 PDO 的发送类型，子索引 3h 描述了 PDO 的禁止时间（inhibit time）；子索引（不支持），4h 未被使用；子索引 5h 描述了事件定时器（不支持）。PDO 映射参数

PDO 映射参数描述了一个 PDO 中被映射的对象，被映射对象值就是 PDO 所携带的数据。PDO 映射参数的数据类型是对象词典中 21H 描述，映射参数结构如下表所示：

Index	Sub-Index	Description	Data Type	.....
-------	-----------	-------------	-----------	-------

# CANopen 协议

21h	0h	number of mapped	Unsigned8	.....
	1h	1st object	Unsigned32	.....
	2h	2nd object	Unsigned32	.....
	.....	.....	.....	.....
	40h	64th object	Unsigned32	.....

映射参数结构

子索引 0 描述了该 PDO 映射对象的数目（不包括子索引 0）；子索引 1h-40h 描述了被映射的对象，一个 PDO 最多可以映射 64 个对象。

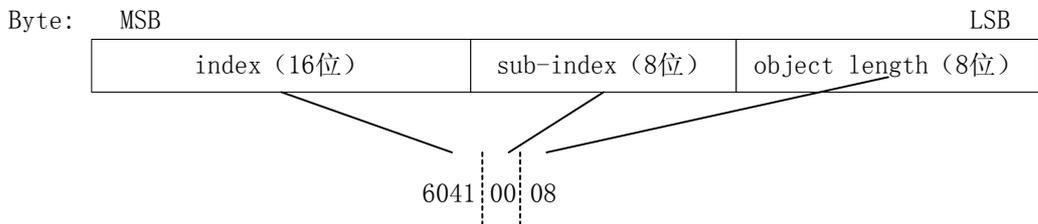
下面以默认的第二个 TPDO 映射参数为例来描述实时数据、PDO、对象词典之间关系。

该 TPDO 映射参数是由对象词典中的对象 1A01h（PDO 映射参数）来描述，该对象的数据类型是 PDO Mapping（PDO 映射即 21h 描述），1A01h 的数据结构下表所示：

Index	Sub-Index	.....	Data Type	Default value	.....
1A01h	0h	.....	Unsigned8	2	.....
	1h	.....	Unsigned32	60410010	.....
	2h	.....	Unsigned32	60610008	.....

对象 1A01h

该 PDO 共映射了 2 个对象（0h 缺省值为 2），每个被映射的对象由 1h-2h 来描述，1h-2h 的数据类型为无符号 32 位（Unsigned32），它的结构如下图所示：



映射对象的描述

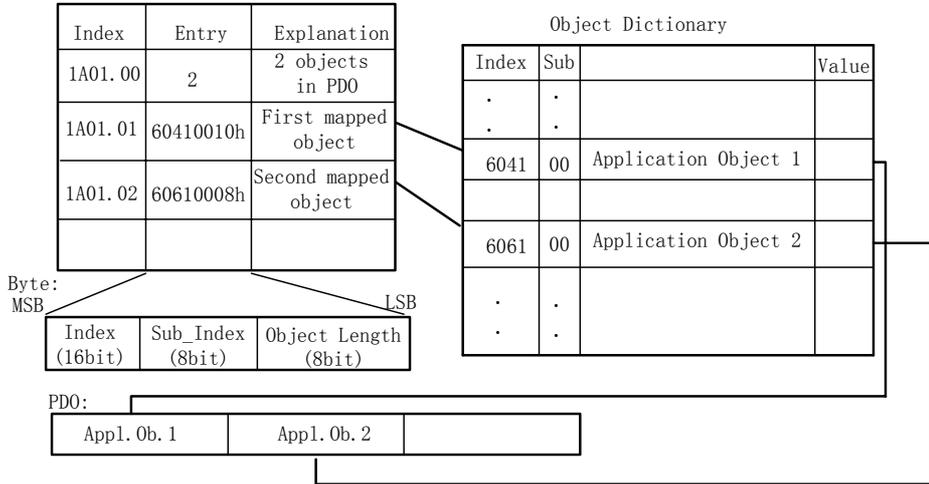
将 32 位分为三部分，分别是被映射对象索引占 16 位，子索引占 8 位，后面 8 位来描述被映射对象的数据长度。

二个被映射对象封装成一个 PDO，对象 6041h 和 6061h 就分别对应 PDO 携带的数

# CANopen 协议

据值。

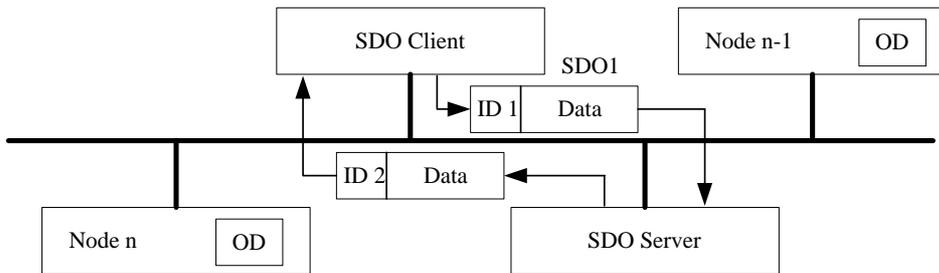
PDO、PDO 映射、对象词典之间总的对应关系如下图所示：



PDO、PDO 映射、对象词典之间关系

## 2.4.5. Service Data Object (SDO)

使用服务数据对象 SDO 可以访问对象词典的对象。一个 SDO 使用两个不同标识符的 CAN 数据帧，因为 SDO 通讯是必须确认的，如下图所示：



SDO(Peer to Peer Communication)

通过 SDO 点对点的通讯，就可以建立起两个设备之间的通讯，被访问对象词典的设备是 SDO 服务端即 SSDO (Server-SDO)，另外一个设备就是 SDO 客户端即 CSDO (Client-SDO)。

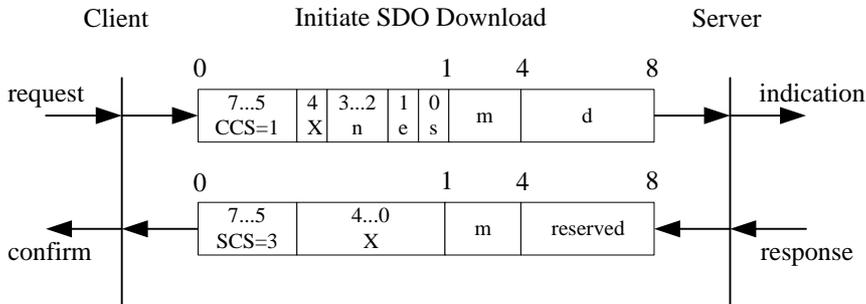
驱动器支持下列 SDO 通信协议：

- (1) 发起 SDO 下载 (Initiate SDO Download)
- (2) 发起 SDO 上传 (Initiate SDO Upload)
- (3) 中断 SDO 传输 (Abort SDO Transfer)

# CANopen 协议

## 2.4.5.1 发起 SDO 下载协议

该协议用于修改对象词典的内容，协议如下图所示：



发起 SDO 下载协议

CCS: 客户端命令符；值为 1 表示发起下载请求

SCS: 服务端命令符；值为 3 表示发起下载反应

n: 只有 e=1 并且 s=1 时，这个值有效；有效时，表明[8-n,7]字节范围内不包含数据

e: 发送类型；值为 0 表示正常传输，值为 1 表示加速传输

s: 数据大小指示；值为 0 表示没有指示数据大小，值为 1 表示指示数据大小

m: 多路器；包含了由 SDO 传输的数据的索引和子索引

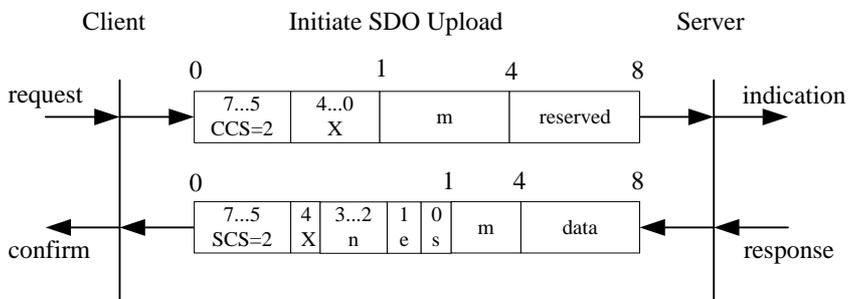
d: 数据值

X: 没有被使用；值总为 0

reserved: 保留，为以后用；值总为 0

## 2.4.5.2 发起 SDO 上传协议

这个协议用于访问对象词典的内容，协议如下图所示：



发起 SDO 上传协议

# CANopen 协议

CCS: 客户端命令符; 值为 2 表示发起上传请求

SCS: 服务端命令符; 值为 2 表示发起上传反应

n: 只有 e=1 并且 s=1 时, 这个值有效; 有效时, 表明[8-n,7]字节范围内不包含数据

e: 发送类型; 值为 0 表示正常传输, 值为 1 表示加速传输

s: 数据大小指示; 值为 0 表示没有指示数据大小, 值为 1 表示指示数据大小

m: 多路器; 包含了由 SDO 传输的数据的索引和子索引

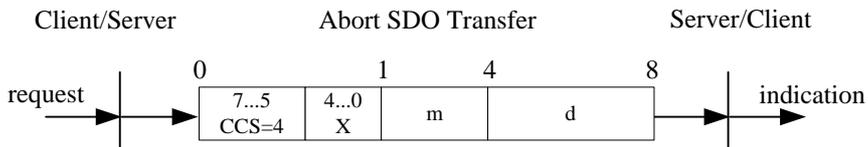
d: 数据值

X: 没有被使用; 值总为 0

reserved: 保留, 为以后用; 值总为 0

## 2.4.5.3 中断 SDO 传输协议

当 SDO 通信出现错误时执行该协议, 协议如下图所示:



中断 SDO 传输协议

CCS: 客户端命令符; 值为 4 表示中断传输请求

X: 没有被使用; 值总为 0

m: 多路器; 包含了由 SDO 传输的数据的索引和子索引

d: 包含了四个字节的关于中断传输原因的中断号

下表描述了支持的中断 SDO 传输的中断号及原因:

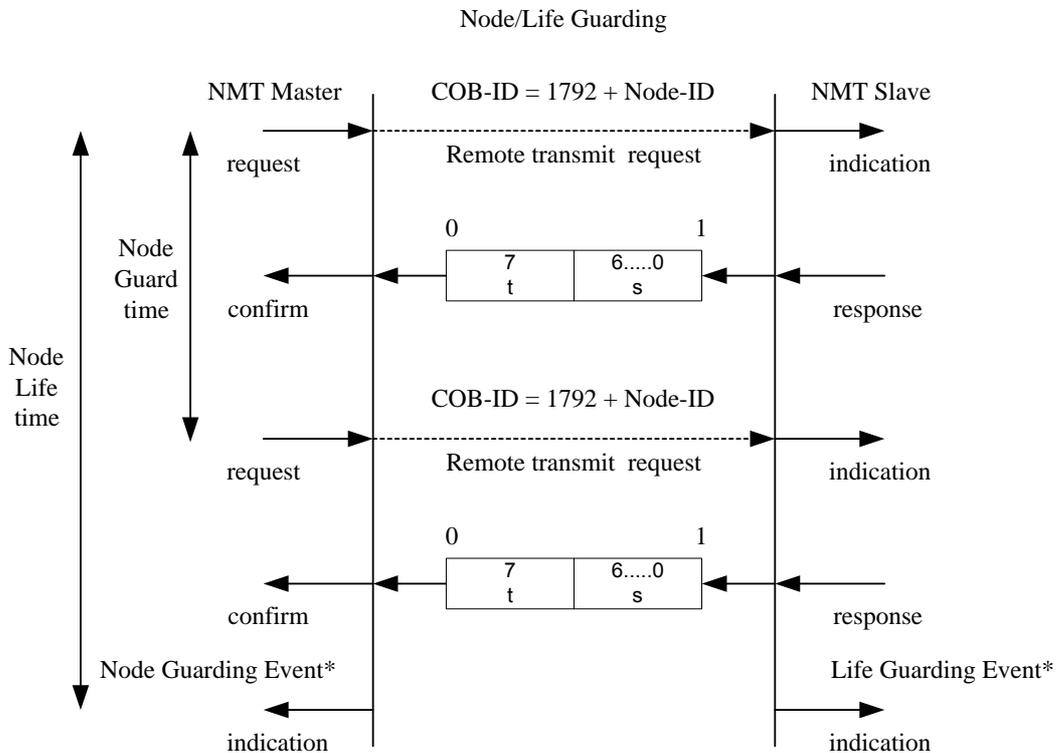
中断号	描述
05030000h	置换位没有更换
05040000h	SDO 协议超时
05040005h	CRC 错误
06010000h	该对象不支持访问
06010001h	该对象不可读

# CANopen 协议

06010002h	该对象不可写
06020000h	该对象不存在
06040041h	该对象不可映射成 PDO
06040042h	该对象的长度超出了 PDO 映射长度
06060000h	由于硬件错误导致访问失败
06090011h	子索引不存在
06090030h	超出参数值的范围
06090031h	大于参数值的最大值
06090032h	小于参数值的最小值

## 2.4.6. Nodeguard

该 Nodeguard 协议用来发现网络上的远程错误，整个协议见下图所示：



# CANopen 协议

---

## 节点监视协议

**s:** 由 7 位 (0-6) 数据组成从站当前所处状态, 其中值为 4 表示从站处于停止状态, 值为 5 从站处于运行状态, 值为 127 从站处于预运行状态。

**t:** 由 1 位 (7) 数据构成的触发位, 即在从设备连续发送响应 (**response**) 时这个位必须更换, 即这次响应的 **t** 为 1 下次就为 0 进行交替更换。当节点监视协议开始执行后, 从设备发出的第一个响应中的这个触发位值为 0。只有从设备进入复位通讯后, 这个触发位被重新复位为 0。

**NMT** 主站周期地向从站发送远程请求, 这个周期就是节点监视时间 (**Node Guard time**), 每个从站的节点监视时间可能不同, 然后从设备做出响应, 发送一个内容为从站状态 (**s**) 和触发位 (**t**) 的数据包给主站。节点生命时间 (**Node Life time**) 等于节点监视时间与生命时间因子 (**Life time factor**) 的乘积, 对于每个从站节点生命时间可能不同。

假如在节点生命时间里, 主站没有向从站发送远程请求, 则通过生命监视事件 (**Life Guarding Event**) 的发生预示远程节点错误发生, 即主站出现问题。

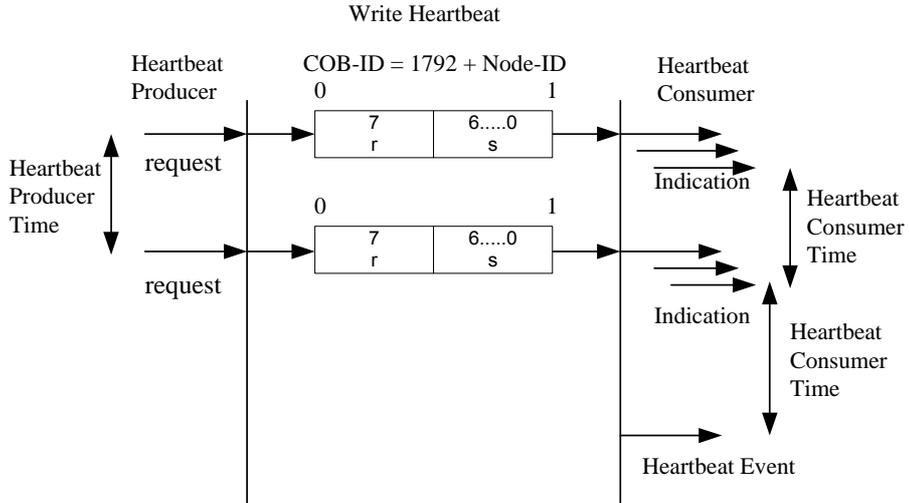
假如在节点生命时间里:

- (1) 假如从站没有对主站发送的远程请求做出响应, 则通过节点监视事件 (**Node Guarding Event**) 的发生预示远程节点发生错误, 即从站出现问题。
- (2) 假如从站对主站发送的远程请求所作的响应中, 状态值与所期望的状态 (从站没有进入主站所控制的状态) 不符, 则通过节点监视事件 (**Node Guarding Event**) 的发生预示远程节点发生错误, 即从站出现问题。

### 2.4.7. Heartbeat

该 **Heartbeat** 协议可以替代 **Nodeguard** 协议来监视从站。整个协议见下图所示。心跳生产者周期的发送心跳信息, 内容包括所处的状态和保留位, 一个或者更多的消费者接到此信息。心跳消费者在消费者心跳时间 (**Heartbeat Consumer Time**) 内监视接到的生产者发送的心跳信息, 假如在消费者心跳时间内没有收到心跳信息, 则发送心跳事件 (**Heartbeat Event**), 预示心跳生产者错误发生。

# CANopen 协议



心跳协议

r: 占 1 位 (7), 保留使用, 值为 0。

s: 占 7 位 (0-6), 表示心跳生产者所处状态, 值为 0 表示所处 BOOTUP 状态, 值为 4 表示所处停止状态, 值为 5 表示所处运行状态, 值为 127 表示所处预运行状态。

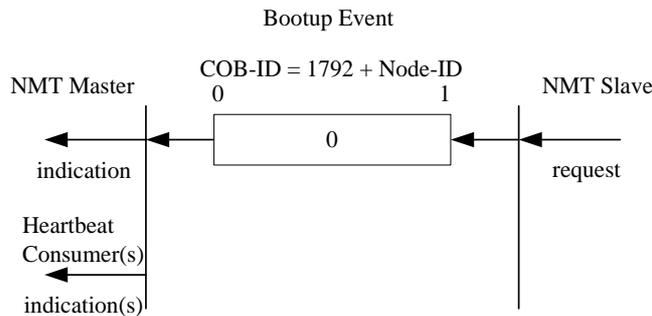
当心跳生产者的心跳时间值不是 0, 则心跳协议就立即执行。

如果一个设备的心跳时间值不为 0, 当它启动时, 状态从“初始化”进入“预运行”状态时心跳协议就执行了, 这种情况下 BootUp 信息就是第一个心跳信息。

对于一个设备不允许心跳协议和监视协议同时执行。

## 2.4.8. Bootup

这个 Bootup 协议是被 NMT 从站用来通知 NMT 主站: NMT 从站初始化结束已经进入预运行状态。整个协议下图所示, 通讯对象标识符是 1792+Node-ID, 数据帧只包含了一个字节, 值且为 0。



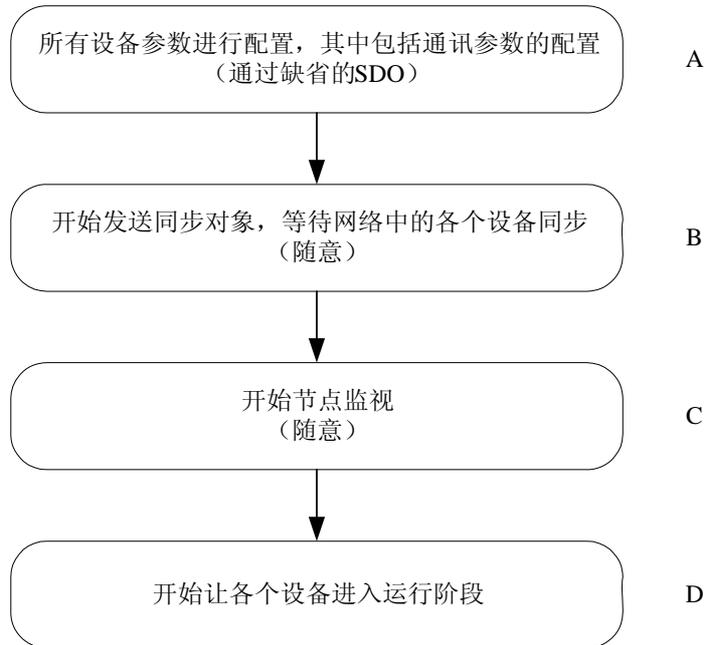
# CANopen 协议

## BootUp 协议

### 2.5. 网络初始化和系统 Bootup

#### 2.5.1. 初始化流程

在 CANopen 现场总线里，整个网络的初始过程的流程见下图所示，由 NMT 主站进行控制从而进行实际应用或者配置。



网络初始化流程图

在步骤 A 中，当设备上电后自动进入预运行状态。在这个状态中，可以通过缺省的 SDO 访问该设备，网络上所有的支持参数配置的节点都可以对它进行设备参数配置。例如通过 SDO 对 PDO 的标识符或者映射参数等进行配置。

假如在应用中，需要同步网络中某些或者所有节点，则在步骤中 B 中需要建立相应的机构来完成。它需要在节点进入运行状态即步骤 D 之前，确保通过发送同步对象同步所有节点。这第一个同步对象是在设备进入预运行状态之后的一个同步周期内发送的。

在步骤 C 中，如果设备支持节点监视，则可以通过在步骤 A 中进行监视参数配置，然后在该步骤激活节点监视。

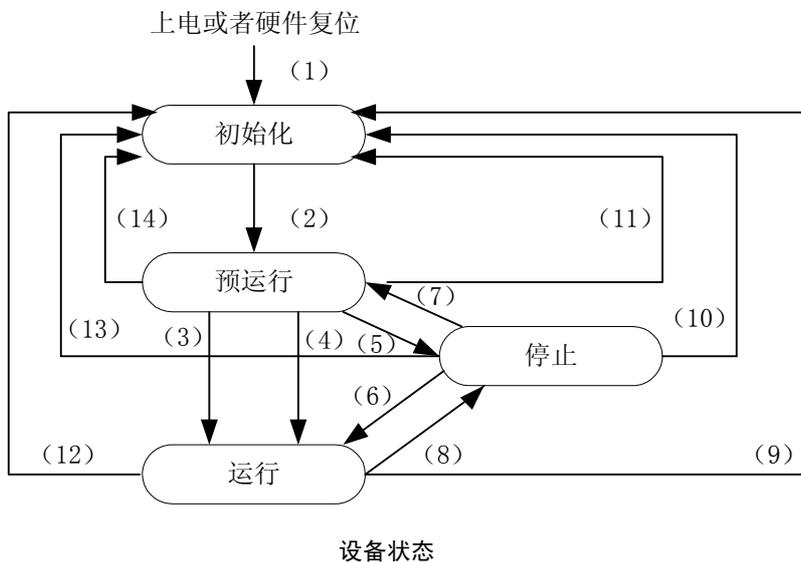
在步骤 D 中，所有的节点都可以通过 PDO 对象进行通讯。

# CANopen 协议

## 2.5.2. NMT 状态机

设备的状态流程图如下图所示，当设备完成初始化状态后直接进入预运行状态，在此状态期间可以通过 SDO 进行对设备参数进行配置，然后节点就可以进入运行状态了。

(1)	上电后自动进入初始化状态
(2)	初始化完成后自动进入预运行状态
(3), (6)	开始远程节点指示
(4), (7)	进入预运行状态指示
(5), (8)	停止远程节点指示
(9), (10), (11)	复位节点指示
(12), (13), (14)	复位通讯指示



### 2.5.2.1 初始化状态 (Initialisation)

初始化状态被分成三个子状态，这样就可以使节点的全部参数或者部分参数复位，见图 6.3 所示。

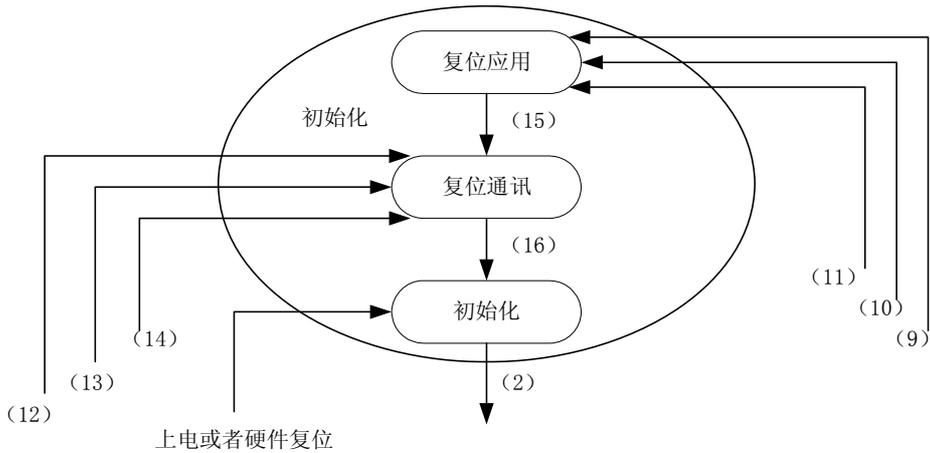
# CANopen 协议

复位应用 (**Reset\_Application**): 在这个状态里, 制造商指定协议和标准设备协议中所指定的参数值被复位恢复成上电值。上电值设置完后自动进入复位通讯状态。

复位通讯 (**Reset\_Communication**): 在这个状态里, 通讯协议中的参数被复位成上电值。该状态完成之后自动进入初始化状态。

初始化 (**Initialising**): 该状态是设备上电或者硬件复位后进入的第一个子状态。当完成基本的节点初始化后, 设备执行写 **BootUp** 对象服务, 从而自动进入预运行状态。

上电值是最后储存的参数。



(2)	初始化完成后自动进入预运行状态
(12), (13), (14)	复位通讯指示
(9), (10), (11)	复位节点指示
(15)	执行完应用复位
(16)	执行完通讯复位

初始化状态结构

## 2.5.2.2 预运行状态 (Pre-Operational)

在该状态, 可以通过 **SDO** 进行通讯, 由于 **PDO** 不存在, 所有不允许 **PDO** 通讯。**PDO** 的配置、设备参数以及应用对象的分配都可以通过配置应用来进行。

通过发送开始远程节点请求, 节点可以进入运行状态。

# CANopen 协议

## 2.5.2.3 运行状态 (Operational)

在该状态，所有的通讯对象都可以执行。

## 2.5.2.4 停止状态 (Stopped)

设备被转换进入停止状态，除了已经激活存在的节点监视和心跳监视之外，通讯迫使被停止。

## 2.5.3. 设备状态和通讯对象的联系

下表显示了设备通讯状态和通讯对象之间的联系，设备只有处于某个通讯状态下，才能执行某些通讯对象。

通讯对象	初始化状态	预运行状态	运行状态	停止状态
PDO			√	
SDO		√	√	
同步对象		√	√	
时间帖对象		√	√	
应急对象		√	√	
Boot-Up 对象	√			
NMT 对象		√	√	√

状态和通讯对象

## 2.5.4. 设备状态转换

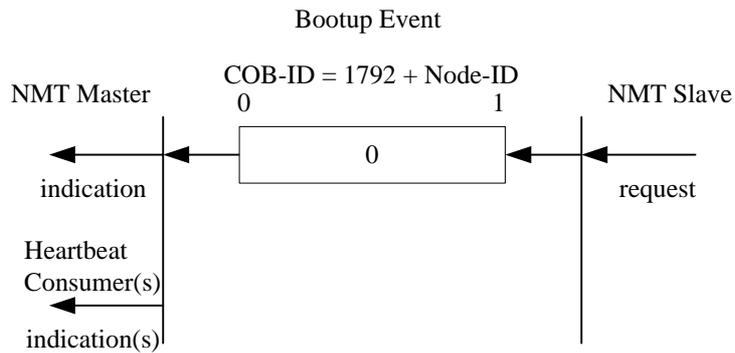
设备状态的改变可以由下面几方面所引起：

- (1) 接收到用于网络管理的 NMT 对象
- (2) 硬件复位
- (3) 由设备协议定义的应用事件发起的模块控制服务

# CANopen 协议

## 2.5.5. Bootup

这个协议是被 NMT 从站用来通知 NMT 主站：NMT 从站初始化结束后已经进入预运行状态。整个协议见下图所示，数据帧只包含了一个字节，值且为 0。



BootUp 协议

# CANopen 协议

## 第三章 CANopen 设备规范

Emergency Message 和一般对象的描述参见设备协议 DSP402 和通讯协议 DS301。

### 3.1 PDO 映射

驱动器内容预先定义了 PDO 映射结构，支持动态修改映射结构，用户可以根据需要更改映射结构。

#### 3.1.1 RPDO 映射

预定义的 RPDO 映射如下表所示：

PDO 序号	映射对象索引	映射对象名称	描述
1	6040h	控制字	控制设备状态
2	607Ah 6081h	目标位置 目标最高速度	控制设备目标位置和目标速度（协议位置模式）
3	60FFh	目标速度	控制设备目标速度（速度通讯模式）
4	607Ah	目标位置	控制设备目标位置（周期位置模式）

#### 3.1.2 TPDO 映射

预定义的 TPDO 映射如下表所示：

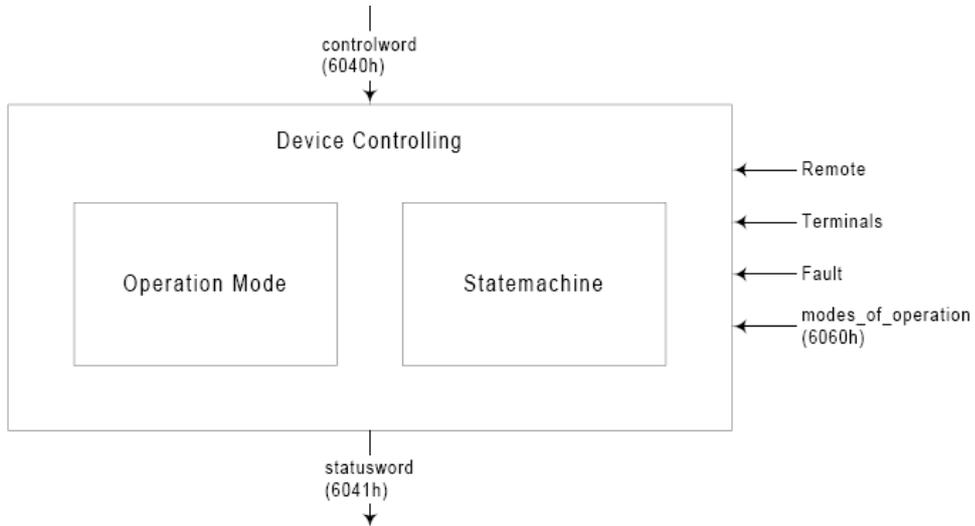
PDO 序号	映射对象索引	映射对象名称	描述
1	6041h	状态字	显示状态
2	6064h 606Ch	实际位置 实际速度	显示实际位置和实际速度
3	无		
4	6061h	实际模式	显示当前模式

# CANopen 协议

## 3.2 设备控制

设备控制功能模块控制电机的各种功能，可以分成两部分：

- (1) 状态机的设备控制
- (2) 操作模式功能

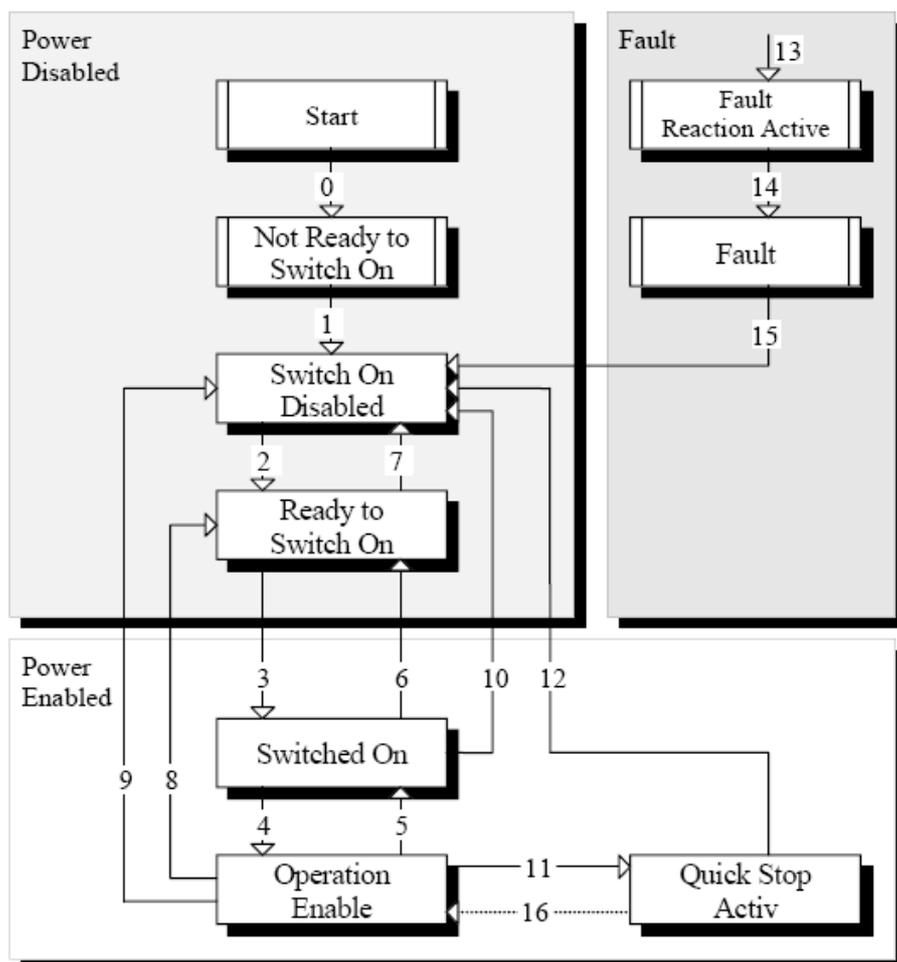


设备的状态由控制字（6040h）来控制  
设备的状态由状态字（6041h）来显示

### 3.2.1 状态机

状态机描述了设备的各种状态和驱动器的控制顺序。

# CANopen 协议



状态	描述
Not Ready to Switch On	驱动器未准备好接通，正在进行初始化
Switch On Disabled	禁止接通，驱动器初始化完毕，配置和调整参数可以修改，驱动器锁定
Ready to Switch On	准备接通，配置和调整参数可以修改，运动功能不能使用，驱动器锁定
Switched On	已经接通，功率级已经准备好运行，驱动器锁定
Operation Enable	运行激活，没有故障，运动功能激活，可以调整参数
Quick Stop Active	快速停车激活，执行紧急停车，可以调整参数和执行运动功能
Fault Reaction Active	故障反应激活，出现非致命故障
Fault	故障，驱动器锁定，参数可以调整

# CANopen 协议

## 3.2.1.1 状态转换

状态转换是由驱动器内部事件或者控制字（6040h）命令所引起。

转换	事件	行为
0	复位	驱动器初始化
1	驱动器初始化完毕	无
2	来自主机控制字命令“Shut Down”	无
3	来自主机控制字命令“Switch On”	功率级接通，电机带有力矩
4	来自主机控制字命令“Enable Operation”	运动功能可以执行
5	来自主机控制字命令“Disable Operation”	运动功能禁止
6	来自主机控制字命令“Shut Down”	功率级关闭，电机无力矩
7	来自主机控制字命令“Quick Stop”	无
8	来自主机控制字命令“Shut Down”	功率级关闭，电机无力矩
9	来自主机控制字命令“Disable Voltage”	功率级关闭，电机无力矩
10	来自主机控制字命令“Disable Voltage” / “Shut Down”	功率级关闭，电机无力矩
11	来自主机控制字命令“Quick Stop”	紧急停车被执行
12	来自主机控制字命令“Disable Voltage”	功率级关闭，电机无力矩
13	驱动器出现致命故障	执行相应的处理
14	故障处理结束	功率级关闭，电机无力矩
15	来自主机控制字命令“故障复位”	故障位清除
16	来自主机控制字命令“Enable Operation”	运动功能可以执行

## 3.2.2 对象描述

在这一节将描述控制字对象 6040h、状态字对象 6041h、操作模式对象 6060h 和操作模式显示对象 6061h。

# CANopen 协议

## 3.2.2.1 对象 6040h: 控制字

索引	6040h
名称	控制字
数据类型	Unsigned16
属性	Rw
PDO 映射	Yes
值范围	0~65535

位的分配:

位序号	名称	位序号	名称
0	Switch On	8	Halt
1	Disable Voltage	9	保留
2	Quick Stop	10	保留
3	Enable Operation	11	制造商指定
4	操作模式指定	12	制造商指定
5	操作模式指定	13	制造商指定
6	操作模式指定	14	制造商指定
7	Fault Reset	15	制造商指定

控制字中的命令:

命令	位 7 Fault Reset	位 3 Enable Operation	位 2 Quick Stop	位 1 Disable Voltage	位 0 Switch On	状态转换
Shut Down	0	X	1	1	0	2, 6, 8
Switch On	0	X	1	1	1	3
Disable Voltage	0	X	X	0	X	7, 9, 10, 12
Quick Stop	0	X	0	1	X	7, 10, 11
Disable Operation	0	0	1	1	1	5
Enable Operation	0	1	1	1	1	4, 16
Fault Reset	1	X	X	X	X	15

位值为“X”表示不相关

# CANopen 协议

控制字中与操作模式相关的位：

操作模式	位 4	位 5	位 6	位 8
周期位置模式	保留	保留	保留	电机运动停止
速度通讯模式	保留	保留	保留	电机运动停止
协议位置模式	新设置的点	立即改变设置	绝对：0 相对：1	电机运动停止

## 3.2.2.2 对象 6041h：状态字

索引	6040h
名称	控制字
数据类型	Unsigned16
属性	Ro
PDO 映射	Yes
值范围	0~65535

位的分配：

位序号	名称	位序号	名称
0	Ready to Switch On	8	Manufacture Specific
1	Switched On	9	Remote
2	Operation Enabled	10	Target Reached
3	Fault	11	Internal Limit Active
4	Voltage Disabled	12	Operation Mode Specific
5	Quick Stop	13	Operation Mode Specific
6	Switch On Disabled	14	Manufacture Specific
7	Warning	15	Manufacture Specific

# CANopen 协议

状态:

状态	位 6 Switch On Disable	位 5 Quick Stop	位 3 Fault	位 2 Operation Enable	位 1 Switched On	位 0 Ready to Switch On
Not Ready Switch On	0	X	0	0	0	0
Switch On Disabled	1	X	0	0	0	0
Ready to Switch On	0	1	0	0	0	1
Switched On	0	1	0	0	1	1
Operation Enabled	0	1	0	1	1	1
Fault	0	X	1	1	1	1
Fault Reaction Active	0	X	1	1	1	1
Quick Stop Active	0	0	0	1	1	1

位值为“X”表示不相关

状态字中与操作模式相关的位:

操作模式	位 12	位 13
周期位置模式	保留	保留
速度通讯模式	保留	保留
协议位置模式	设置点确认	

状态其他位的描述:

位 4: **voltage disabled** 位, 当这个位为 0 时, **Disable voltage** 请求有效

位 7: 置位表明驱动器内部存在警告, 虽然没有达到故障, 但应该引起重视

位 10: 置位表明驱动器到达目标位置

位 9、11: 目前不支持

# CANopen 协议

## 3.2.2.3 对象 6060h: 操作模式

该对象用于设置驱动器的模式，然后由对象 6061h 来显示。

索引	6060h
名称	操作模式
数据类型	Integer8
属性	wo
PDO 映射	Yes
值范围	-128~127

值的描述:

值	模式
FAh	烤机模式
FDh	速度通讯模式
1	协议位置模式
6	回原点模式
8	周期位置模式

## 3.2.2.4 对象 6061h: 操作模式显示

该对象用于显示当前驱动器的模式。

索引	6061h
名称	操作模式显示
数据类型	Integer8
属性	Ro
PDO 映射	Yes
值范围	-128~127

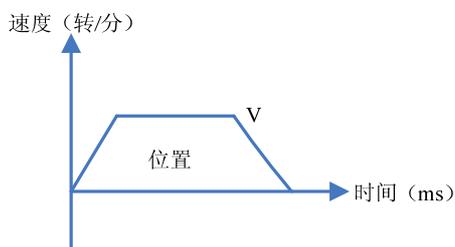
值的描述与对象 6060h 一致。

## 3.2.3 协议位置模式 (Profile Position Mode)

该模式下，上位机通过发送位置指令（相对或者绝对），驱动器根据参数设置，自动按照给定的速度、加减速时间走到指定的位置。

# CANopen 协议

步进电机速度时间曲线见下图：



下面描述与该模式有关的对象。

### 3.2.3.1 对象 607Ah：目标位置

目标位置是驱动器在协议位置模式下将要到达的位置。通过控制字（6040h）中的位 6 来区分绝对和相对运动。

索引	607Ah
名称	目标位置
数据类型	Integer32
属性	Rw
PDO 映射	Yes
值范围	$-2^{31} \sim 2^{31}-1$

### 3.2.3.2 对象 6081h：协议速度

协议速度是电机加速后最后达到的速度。

索引	6081h
名称	协议速度
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	$0 \sim 2^{31}-1$

### 3.2.3.3 对象 6083h：协议加速度

加速度为直线加速。

# CANopen 协议

索引	6083h
名称	协议加速度（单位 MS）
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	0~2 <sup>32</sup> -1

## 3.2.3.4 对象 6084h：协议减速度

索引	6084h
名称	协议减速度（单位 MS）
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	0~2 <sup>32</sup> -1
2~32767	不支持

## 3.2.3.5 功能描述

可以通过两种方式到达目标位置：

- (1) 一系列设置点：当驱动器到达目标位置后驱动器处理下一个目标位置点，因此驱动器到达一个目标位置点后电机速度并不为 0（目前驱动器不支持该方式）
- (2) 单个设置点：当驱动器到达目标位置后，向上位机发送一个到达位置的信号，然后接收到一个新的目标位置。在开始下一个目标位置运动之前，到达目标位置时电机速度降为 0

这两种模式是通过控制字中的“立即改变设置”“新设置点”和状态字中的“设置点确认”位定时来控制。当驱动器单元正在处理一个设置点，为了准备一系列设置点，这些位都是通过“请求—响应”的机制来完成的，这样缩短了主机程序的反应时间。

# CANopen 协议

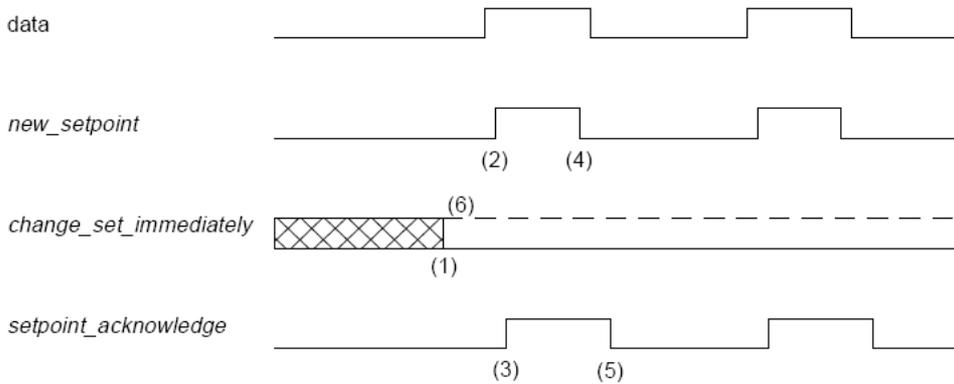


图 1 主机中设置点传输

图 1、图 2、图 3 描述了这两种方式的区别。在控制字中的立即改变设置（`change_set_immediately`）位决定是哪种方式。图 1 中“`change_set_immediately`”位一直是“0”（实线）则运动方式是“单个设置点”（1）。主机通过控制字中的“`new_setpoint`”值为 1 的信号，来表明新的数据有效（2）。驱动器准备好新的数据缓存后，驱动器响应该信号，在状态字中的“`setpoint_acknowledge`”值为 1，表明收到主机信号（3），主机释放“`new_setpoint`”位（4），之后驱动器置“`setpoint_acknowledge`”位为 0，表明已经接受了新的数据（5）。这种机制导致了在图 2 驱动器在  $t_1$  时刻到达目标位置  $X_1$  后，电机速度为 0，发送给主机信号后，设置点如上面所描述，下一个目标位置  $X_2$  在  $t_2$  时刻开始  $t_3$  时刻到达。

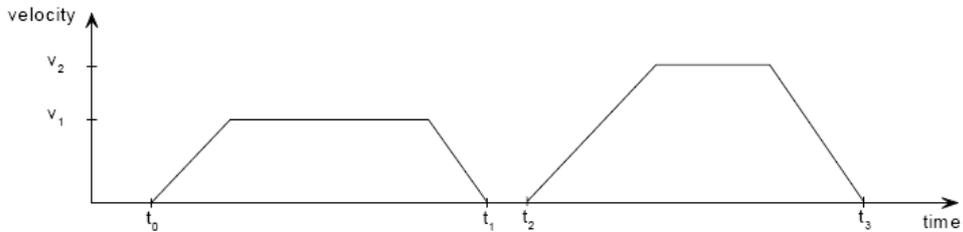


图 2 单个设置点

当图 1 中“`change_set_immediately`”位设置为 1（虚线）时，主机通知驱动器采取第二种方式（一系列点设置），在到达前一个位置时要处理新的位置点，其他位的信号定时没有改变。这种机制使得驱动器预先处理下一个位置点  $X_2$ ，在  $t_1$  时刻到达位置  $X_1$  时保持速度，然后驱动器立即开始移动到下一个已经处理好的位置点  $X_2$ 、见图 3 所示。

# CANopen 协议

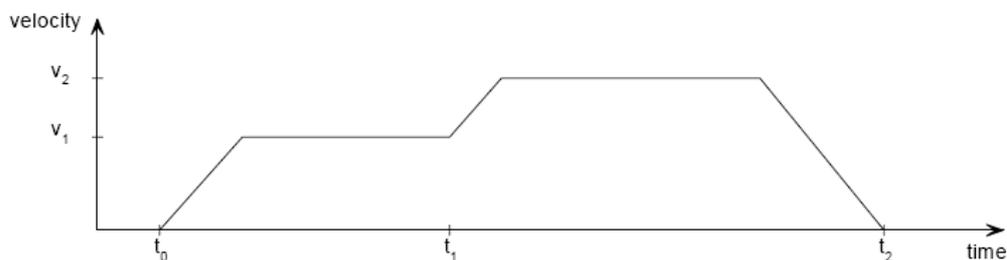


图 3 一系列设置点

所涉及的控制字中的位:

位 4: `new_setpoint` (正值有效) (新设置点)

位 5: `change_set_immediately` (立即改变设置)

位 6: `absolute/relative` (绝对/相对)

所涉及的状态字中的位:

位 12: `setpoint_acknowledge` (设置点确认)

位 13: `following` (跟踪误差)

## 3.2.4 速度通讯模式

该模式下，可以控制步进电机的速度、加减速时间，下面描述与该模式有关的对象。

### 3.2.4.1 对象 60FF: 给定速度

给定速度是电机加速后最后达到的速度。

索引	60FFh
名称	给定速度
数据类型	Integer32
属性	Rw
PDO 映射	Yes
值范围	$-2^{31} \sim 2^{31}-1$

### 3.2.4.2 对象 6083h: 协议加速度

加速度为直线加速。

索引	6083h
名称	协议加速度 (单位 MS)

# CANopen 协议

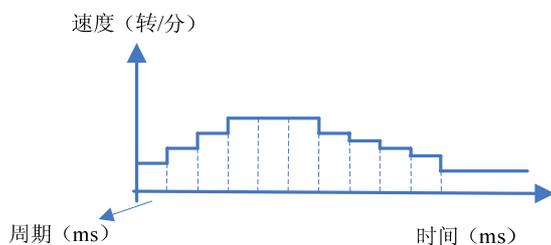
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	$0 \sim 2^{32}-1$

### 3.2.4.3 对象 6084h: 协议减速度

索引	6084h
名称	协议减速度 (单位 MS)
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	$0 \sim 2^{32}-1$
2~32767	不支持

### 3.2.5 周期位置模式

该模式下，上位机通过周期性发送位置指令，从而控制步进电机的位置。步进电机速度时间曲线见下图：



下面描述与该模式有关的对象。

#### 3.2.5.1 对象 607Ah: 目标位置

目标位置是驱动器在周期位置模式下将要到达的绝对位置。

# CANopen 协议

索引	607Ah
名称	目标位置
数据类型	Integer32
属性	Rw
PDO 映射	Yes
值范围	$-2^{31} \sim 2^{31}-1$

## 3.2.5.2 对象 60C2h: 周期描述

索引	60C2h 子索引 1
名称	周期 (单位 100us)
数据类型	Unsigned32
属性	Rw
PDO 映射	No
值范围	$1 \sim 2^{31}$

## 3.2.6 回原点模式

目前支持两种回原点模式，在回原点的过程中需要使用原点信号，下面描述与该模式有关的对象。

对象 6098h: 回原点方式

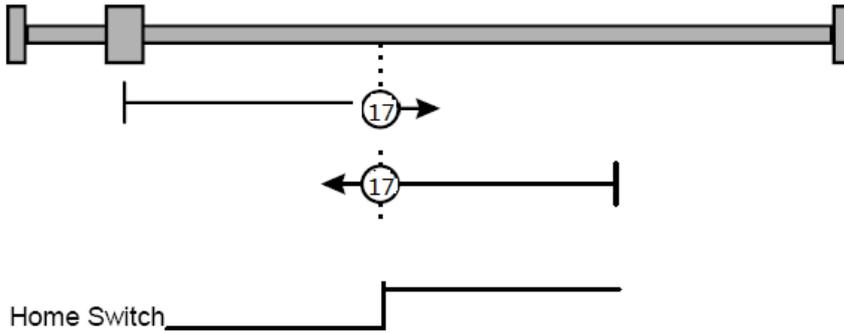
索引	6098h
名称	回原点方式
数据类型	Integer8
属性	wo
PDO 映射	Yes
值范围	-128~127

本驱动器支持两种回原点方式：

第一种：回原点限位开关输入（6098h=17）

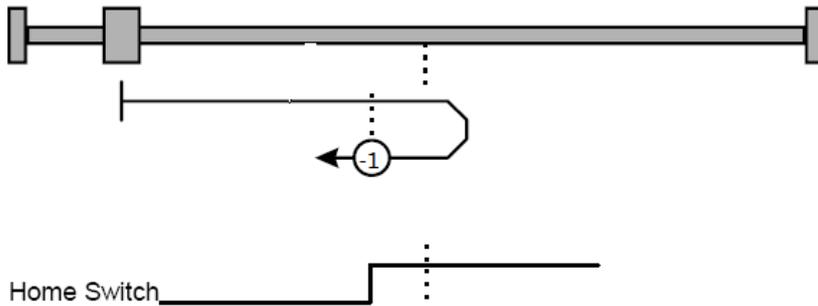
该方式下，驱动器带动电机以对象 6099sub1 的速度寻找原点，当遇到原点开关上升沿或者下降沿，则驱动器带动电机停止，寻找零点结束。

# CANopen 协议



第二种：回原点限位开关输入+高低档速度寻找（6098h=-1）

该方式下，驱动器带动电机以对象 6099sub1 的速度寻找原点，当遇到原点开关上升沿或者下降沿，则驱动器带动电机减速停止，然后以对象 6099sub2 的速度反方向低速寻找原点，当遇到原点开关上升沿或者下降沿，则驱动器带动电机停止，寻找零点结束。



对象 6099h：回原点速度  
加速度为直线加速。

索引	6099h
名称	回原点速度
数据类型	Unsigned32

子索引	01h
名称	回原点寻找限位开关速度
数据类型	int32
属性	Rw
PDO 映射	Yes
值范围	-231~231-1

# CANopen 协议

子索引	02h
名称	低速寻找限位开关速度
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	0~232-1

对象 609Ah: 回原点加速度  
加速度为直线加速。

索引	609Ah
名称	回原点加速度（单位 MS）
数据类型	Unsigned32
属性	Rw
PDO 映射	Yes
值范围	0~2 <sup>32</sup> -1

## 3.3 使用举例

### 3.3.1 设备控制操作举例

当驱动器上电后并且发送了 **Bootup** 信息后，表明驱动器初始化完毕，现在可以进行 SDO 通讯，此时可以通过 SDO 访问对象词典或者控制驱动器的状态机。

访问状态机的状态可以通过读 SDO 6041h 来实现。例如驱动器上电后初始化完毕，此时状态机的状态为“Switch On Disabled”，读对象 6041h 可以知道当前驱动器状态：

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	40	41	60	00	00 00 00 00
581h	40	41	60	00	40 00 00 00

第一个报文为上位机查询驱动器状态的报文

第二个报文为驱动器回答上位机的状态询问

控制状态机的状态可以通过 SDO 写对象 6040h 来实现。例如上位机预控制驱动器状态为“Switched On”即设置 6040h 值为 7，如果驱动器接收成功，则返回一个确认信息。

# CANopen 协议

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	23	40	60	00	07 00 00 00
581h	60	40	60	00	00 00 00 00

第一个报文为上位机控制驱动器状态的报文 控制字：0x0007

第二个报文为驱动器应答上位机的状态控制

### 3.3.2 PDO 使用举例

实时数据的传输一般利用 PDO 通讯，这样可以减少报文开销，提高传输速度。下面举例说明 PDO 通讯的使用：

例如：

驱动器网络状态	工作状态	工作模式	RPDO 通讯目的	TPDO 通讯目的
预运行	Operation Enabled	速度通讯模式	实时控制电机的速度	实时显示电机状态和速度以及位置

采用第三个 RPDO 来控制电机和速度，第二个 TPDO 来实时显示电机状态包括位置、速度，PDO 映射操作如下：

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据	注释
		低字节	高字节			
RPDO3 映射请求：						
601h	23	02	16	00	00 00 00 00	主站→设备
581h	60	02	16	00	00 00 00 00	主站←设备
映射第一个参数（控制字 6040h）：						
601h	23	02	16	01	10 00 40 60	主站→设备
581h	60	02	16	01	00 00 00 00	主站←设备
映射第二个参数（功能参数对象 60FFh 内部速度）：						
601h	23	02	16	02	20 00 ff 60	主站→设备
581h	60	02	16	02	00 00 00 00	主站←设备
结束映射						
601h	23	02	16	00	02 00 00 00	主站→设备

# CANopen 协议

581h	60	02	16	00	00 00 00 00	主站←设备
TPDO2 映射请求:						
601h	23	01	1A	00	00 00 00 00	主站→设备
581h	60	01	1A	00	00 00 00 00	主站←设备
映射第一个参数 (当前位置 6064h):						
601h	23	01	1A	01	20 00 64 60	主站→设备
581h	60	01	1A	01	00 00 00 00	主站←设备
映射第三个参数 (当前速度 606Ch):						
601h	23	01	1A	02	20 00 6C 60	主站→设备
581h	60	01	1A	02	00 00 00 00	主站←设备
结束映射						
601h	23	01	1A	00	02 00 00 00	主站→设备
581h	60	01	1A	00	00 00 00 00	主站←设备

## (1) 主站控制从站进入运行状态

通讯对象标识符 (COB-ID)	命令符	数据
0h	01	01 00 00 00 00 00 00

## (2) 主站控制电机运行速度的 RPDO (速度为 100)

通讯对象标识符 (COB-ID)	数据
401h	0F 00 64 00 00 00 00 00

## (3) 设备发送的 TPDO (状态)

通讯对象标识符 (COB-ID)	数据
281h	64 10 01 10 64 00 00 00
....	.....
281h	23 12 13 f0 64 00 00 00
....	.....

### 3.3.3 SDO 读写对象词典对象、驱动器内部参数对象及保存、恢复默认参数举例

例如读对象 2000h 索引为 1 (即电机电流, 值是 10):

# CANopen 协议

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	40	00	20	01	00 00 00 00
581h	40	00	20	01	0a 00 00 00

例如修改对象 2000h 索引为 1 (即电机电流): 值为 20:

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	23	00	20	01	14 00 00 00
581h	60	00	20	01	00 00 00 00

例如保存通讯参数

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	23	10	10	02	73 61 76 65
581h	60	10	10	02	00 00 00 00

例如保存所有参数

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	23	10	10	01	73 61 76 65
581h	60	10	10	01	00 00 00 00

例如恢复默认所有参数

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据
		低字节	高字节		
601h	23	11	10	01	6C 6F 61 64
581h	60	11	10	01	00 00 00 00

### 3.3.4 协议位置模式使用举例

这一节描述协议位置模式的使用, PDO 的设置如下:

RPDO1: 默认映射 (控制字 6040h)

RPDO2: 默认映射 (目标位置 607Ah、协议速度 6081h)

TPDO1: 默认映射 (状态字 6041h)

TPDO2: 默认映射 (当前位置 6064h、当前速度 606Ch)

# CANopen 协议

下面将设备状态切换到协议位置模式并设置相应的参数：

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据	注释
		低字节	高字节			
601h	23	83	60	00	32 00 00 00	加速度：50
581h	60	83	60	00	00 00 00 00	
601h	23	84	60	00	32 00 00 00	减速度：50
581h	60	84	60	00	00 00 00 00	
601h	2F	60	60	00	01 00 00 00	协议位置模式
581h	60	60	60	00	00 00 00 00	

该位置运动是由 RPDO1 启动、RPDO2 来通知具体位置等参数，使用新设置点（6040h 控制字）和设置点确认（6041h 状态字）握手。

## (1) 主站控制从站进入运行状态

通讯对象标识符 (COB-ID)	命令符	数据
0h	01	01 00 00 00 00 00 00

## (2) 设置点

通讯对象标识符 (COB-ID)	数据	注释
301h	F4 01 00 00 E8 03 00 00	位置：500；速度：1000

## (3) 新设置点的控制字

通讯对象标识符 (COB-ID)	数据	注释
201h	1F 00 00 00 00 00 00 00	新设置点

## (4) 等待设备设置点确认的状态字

通讯对象标识符 (COB-ID)	数据	注释
181h	37 12 00 0000 00 00 00	设置点确认

# CANopen 协议

(5) 控制字中新设置点的位复位:

通讯对象标识符 (COB-ID)	数据	注释
201h	0F 00 00 00 00 00 00 00	新设置点位复位

(6) 设备清除设置点确认位

通讯对象标识符 (COB-ID)	数据	注释
181h	37 02 00 00 00 00 00 00	清除确认位

等待着运动完成:

.....

通讯对象标识符 (COB-ID)	数据	注释
181h	37 06 00 00 00 00 00 00	位置完成

### 3.3.5 速度通讯模式使用举例

这一节描述速度通讯模式的使用, PDO 的设置如下:

RPDO1: 默认映射 (控制字 6040h)

RPDO3: 默认映射 (目标速度 60FFh)

TPDO1: 默认映射 (状态字 6041h)

TPDO2: 默认映射 (当前位置 6064h、当前速度 606Ch)

下面将设备状态切换到速度通讯模式并设置相应的参数:

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据	注释
		低字节	高字节			
601h	2F	60	60	00	FD 00 00 00	设置模式
581h	60	60	60	00	00 00 00 00	

该位置运动是由 RPDO1 启动、RPDO3 来通知具体速度等参数。

(1) 主站控制从站进入运行状态

# CANopen 协议

通讯对象标识符 (COB-ID)	命令符	数据
0h	01	01 00 00 00 00 00 00

(2) 设置速度为 100:

通讯对象标识符 (COB-ID)	数据	注释
401h	64 00 00 00 00 00 00 00	速度: 100

(3) 启动:

通讯对象标识符 (COB-ID)	数据	注释
201h	0F 00 00 00 00 00 00 00	启动

(4) 设备反馈的状态字:

通讯对象标识符 (COB-ID)	数据	注释
181h	37 02 00 0000 00 00 00	确认

运动执行:

.....

通讯对象标识符 (COB-ID)	数据	注释
281h	27 04 00 00 64 00 00 00	
.....	.....	
281h	27 05 00 00 64 00 00 00	
.....	.....	

### 3.3.6 回原点模式使用举例

这一节描述回原点模式的使用，PDO 的设置如下：

RPDO1: 默认映射（控制字 6040h）

# CANopen 协议

TPDO1: 默认映射 (状态字 6041h)

其他参数按默认值

对象 6098=17

对象 6099 子索引 01=100

下面将设备状态切换到回原点模式并设置相应的参数:

通讯对象标识符 (COB-ID)	命令符	索引		子索引	数据	注释
		低字节	高字节			
601h	2F	60	60	00	06 00 00 00	设置模式
581h	60	60	60	00	00 00 00 00	

该位置运动是由 RPDO1 启动回原点。

(1) 主站控制从站进入运行状态

通讯对象标识符 (COB-ID)	命令符	数据
0h	01	01 00 00 00 00 00 00

(2) 启动回原点:

通讯对象标识符 (COB-ID)	数据	注释
201h	1F 00 00 00 00 00 00 00	启动

(3) 设备反馈的状态字:

通讯对象标识符 (COB-ID)	数据	注释
181h	37 02 00 0000 00 00 00	确认

运动执行:

.....

通讯对象标识符 (COB-ID)	数据	注释
281h	27 04 00 00 64 00 00 00	
.....	.....	
281h	27 05 00 00 64 00 00 00	
.....	.....	

(4) 设备回原点成功执行反馈的状态字:

# CANopen 协议

通讯对象标识符 (COB-ID)	数据	注释
181h	37 12 00 0000 00 00 00	回原点成功执行

(5) 停止回原点:

通讯对象标识符 (COB-ID)	数据	注释
201h	0F 00 00 00 00 00 00	停止回原点

## 3.4 对象词典描述

### 3.4.1 强制性对象

索引	子索引	名称	数据类型	访问类型	PDO映射	最大值	最小值	缺省值
1000		DeviceType	Unsigned 32	ro	no			40192 h
1001		ErrorRegister	Unsigned 8	ro	no			0
1018		Identity Object						
	0	Largest SubIndex supported	Unsigned 8	ro	no			4
	1	Vendor ID	Unsigned 16	ro	no			0x00484c53
	2	Product Code	Unsigned 16	ro	no			0x53465f30
	3	Revision Nr	Unsigned 16	ro	no			0x56312E30
	4	Serial Nr	Unsigned 16	ro	no			0

### 3.4.2 任意对象

索引	子索引	名称	数据类型	访问类型	PDO映射	最小值	最大值	缺省值
1005		COB-ID SYNC	Unsigned 16	ro	no			80h
1010		Store parameters						
	0	Largest SubIndex supported	Unsigned 16	ro	no			2
	1	Save all parameters	Unsigned32	rw	no			

# CANopen 协议

	2	Save communicaton parameters	Unsigned32	rw	no			
1011		restore parameters						
	0	Largest SubIndex supported	Unsigned 16	ro	no			2
	1	restore all parameters	Unsigned32	rw	no			
	2	restore communicaton parameters	Unsigned32	rw	no			
1014		COB-ID EMCY	Unsigned 16	ro	no			80h+node
1017		Producer Heartbeat Time	Unsigned 16	rw	no	0x7fff		0
1400		RPDO1 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			200h+node
	2	TransmissionType	Unsigned 8	rw	no			255
1401		RPDO2 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			300h+node
	2	TransmissionType	Unsigned 8	rw	no			255
1402		RPDO3 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			400h+node
	2	TransmissionType	Unsigned 8	rw	no			255
1403		RPDO4 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			500h+node
	2	TransmissionType	Unsigned 8	rw	no			255
1600		RPDO1 Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	1
	1	RPDO1 Mapping_1	Unsigned 32	rw	no			60400010h
1601		RPDO2 Mapping						

# CANopen 协议

	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	4
	1	RPDO2 Mapping_1	Unsigned 32	rw	no			607A0020h
	2	RPDO2 Mapping_2	Unsigned 32	rw	no			60810020h
	3	RPDO2 Mapping_3	Unsigned 32	rw	no			0
	4	RPDO2 Mapping_4	Unsigned 32	rw	no			0
1602		RPDO3 Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	4
	1	RPDO3 Mapping_1	Unsigned 32	rw	no			60FF0020h
	2	RPDO3 Mapping_2	Unsigned 32	rw	no			0
	3	RPDO3 Mapping_3	Unsigned 32	rw	no			0
	4	RPDO3 Mapping_4	Unsigned 32	rw	no			0
1603		RPDO4 Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	4
	1	RPDO4 Mapping_1	Unsigned 32	rw	no			607A0020h
	2	RPDO4 Mapping_2	Unsigned 32	rw	no			0
	3	RPDO4 Mapping_3	Unsigned 32	rw	no			0
	4	RPDO4 Mapping_4	Unsigned 32	rw	no			0
1800		TPDO1 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			180h+node
	2	TransmissionType	Unsigned 8	rw	no			255
	3	Inhit time	Unsigned 16	rw	no			1
1801		TPDO2 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			280h+node
	2	TransmissionType	Unsigned 8	rw	no			255
	3	Inhit time	Unsigned 16	rw	no			200
1802		TPDO3 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			380h+node

# CANopen 协议

	2	TransmissionType	Unsigned 8	rw	no			255
	3	Inhit time	Unsigned 16	rw	no			200
1803		TPDO4 CommPar						
	0	Largest SubIndex supported	Unsigned 8	ro	no	2	5	2
	1	COB-ID	Unsigned 16	ro	no			480h+node
	2	TransmissionType	Unsigned 8	rw	no			255
	3	Inhit time	Unsigned 16	rw	no			200
1a00		TPDO1 Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	1	1
	1	TPDO1 Mapping_1	Unsigned 32	rw	no			60410010h
1a01		TPDO2 Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	4
	1	TPDO2Mapping_1	Unsigned 32	rw	no			60640020h
	2	TPDO2Mapping_2	Unsigned 32	rw	no			606C0020h
	3	TPDO2Mapping_3	Unsigned 32	rw	no			0
	4	TPDO2Mapping_4	Unsigned 32	rw	no			0
1a02		TPDO3Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	4
	1	TPDO3Mapping_1	Unsigned 32	rw	no			0
	2	TPDO3Mapping_2	Unsigned 32	rw	no			0
	3	TPDO3Mapping_3	Unsigned 32	rw	no			0
	4	TPDO3Mapping_4	Unsigned 32	rw	no			0
1a03		TPDO4Mapping						
	0	Largest SubIndex supported	Unsigned 8	rw	no	1	4	4
	1	TPDO4Mapping_1	Unsigned 32	rw	no			60610010h
	2	TPDO4Mapping_2	Unsigned 32	rw	no			0
	3	TPDO4Mapping_3	Unsigned 32	rw	no			0
	4	TPDO4Mapping_4	Unsigned 32	rw	no			0

# CANopen 协议

## 3.4.3 设备协议对象

索引	子索引	名称	数据类型	访问类型	PDO映射	最大值	最小值	缺省值	模式
6040		controlword	unsigned16	wo	yes				All
6041		statusword	unsigned 16	ro	yes				All
6060		Mode_operation	unsigned 8	wo	yes				all
6061		mode_of_operation_display	unsigned 8	ro	yes				all
6064		Positon_actual_value	Integer32	ro	yes				All
606C		Speed_actual_value	Integer32	ro	yes				All
607A		Target_positon	Integer32	rw	yes				Pp,Cp
6081		Profile_velocity	Unsigned 16	rw	yes				Pp
6083		Profile_acceleration	Unsigned 16	rw	No				Pp、Sp
6084		Profile_deceleration	Unsigned 16	rw	No				Pp、Sp
6098		home_mode	unsigned 8	Rw	No			17	Hp
6099		homing_speeds							
	0	nrOfEntries	unsigned 8	Ro	No			2	Hp
	1	speed_during_search_for_switch	Integer32		Yes			100	
	2	speed_during_search_for_zero	Unsigned 32		Yes			10	Hp
609A		home_acceleration	Unsigned 32	Rw	No			100	Hp
60C2	1	Ip_time_uint	Unsigned 32	Rw	No		1	84	Cp
60ff		Targent_speed	Integer32	Rw	Yes				Sp
Cp: Cycle position mode                      Sp:Speed mode                      pp: profile positon mode									

# CANopen 协议

## 3.4.4 设备商定义对象

驱动器功能参数在对象词典中的定义如下，具体参数定义见产品说明书。

索引	子索引	名称	数据类型	访问类型	PDO映射	最大值	最小值	缺省值	备注
2000h	0	Largest SubIndex supported	Unsigned 8	Ro	No			5	
2000h	1	给定电流 (x0.1A)	Unsigned 16	rw	no	30	5	10	驱动器不同该值范围有修改
2000h	2	给定细分数	Unsigned 16	Rw	No	25600	200	25600	仅支持 200/400/800/1600/3200/6400/12800/25600
2000h	3	驱动器地址	Unsigned 16	Rw	No	127	1	1	仅在 SW4=On 时该参数有效
2000h	4	传输波特率	Unsigned 16	Rw	No	1000	20	250	仅在 SW4=On 时该参数有效 支持 1000/500/250/125/100/50/20

# 北京和利时电机技术有限公司

## BEIJING HOLLYSYS ELECTRIC TECHNOLOGY CO.,LTD.

---

制 造 商：北京和利时电机技术有限公司（原四通电机）  
地 址：北京市海淀区学清路9号汇智大厦A座10层  
通讯地址：北京2877信箱 邮编：100085  
电话总机：(010)62932100  
销售热线：(010)62927938  
传 真：(010)62927946  
网 址：www.syn-tron.com

南京办事处：

地 址：南京市黄浦路2号黄埔科技大厦B座1807室  
邮 编：210016  
电 话：(025)84293632/37/52/53  
传 真：(025)84514509

深圳分公司：

地 址：深圳市南山区艺园路115号 田厦IC产业园2-004A室  
邮 编：518052  
电 话：(0755)26581960/61/62  
传 真：(0755)26581969

Ver 09/2016

内容如有更改，恕不另告