

K/iK 系列

交流伺服驱动器

CANopen 使用手册 2016 (V1.9)

目录

1、概述	4
1.1 CAN 主要相关文档	4
1.2 本手册使用的术语和缩语	4
1.3 CANopen 概述	5
2、接线和连接	5
3、CANopen 通讯	6
3.1 CAN 标识符分配表	7
3.2 服务数据对象 SDO	7
3.3 过程数据对象 PDO	9
3.3.1 PDO 参数	11
3.4 SYNC 报文	17
3.5 Emergency 报文	18
3.6 HEARTBEAT 报文	20
3.7 网络管理 (NMT)	21
4、单位换算单元 (Factor Group)	22
4.1 单位换算相关参数	22
4.1.1 position factor	23
4.1.2 velocity factor	24
4.1.3 acceleration factor.....	25
5、位置控制功能	26
5.1 位置控制相关参数	28
6、设备控制	30
6.1 控制状态机	30
6.2 设备控制相关参数	32
6.2.1 controlword.....	32
6.2.2 statusword.....	33
6.2.3 shutdown_option_code	35
6.2.4 disable_operation_option_code	36
6.2.5 quick_stop_option_code	36
6.2.6 halt_option_code.....	37
6.2.7 fault_reaction_option_code	37
7、控制模式	37
7.1 控制模式相关参数	38

7.1.1 modes_of_operation.....	38
7.1.2 modes_of_operation_display	38
7.2 回零模式 (HOMING MODE)	39
7.2.1 回零模式的控制字	39
7.2.2 回零模式的状态字	39
7.2.3 回零模式相关参数	40
7.2.4 回零方法	42
7.3 速度控制模式 (PROFILE VELOCITY MODE)	44
7.3.1 速度模式的控制字	44
7.3.2 速度模式的状态字	44
7.3.3 速度控制模式相关参数	44
7.4 位置控制模式 (PROFILE POSITION MODE)	47
7.4.1 位置模式的控制字	47
7.4.2 位置模式的状态字	48
7.4.3 位置控制相关参数	48
7.4.4 功能描述	51
附录 对象字典表	53

1、概述

1.1 CAN 主要相关文档

Document Name	Source
CiA DS 301 V 4.01: CANopen Communication Profile for Industrial Systems - based on CAL	CiA
CiA DSP 402 V 2.0: CANopen Device Profile	CiA

1.2 本手册使用的术语和缩语

CAN	控制器局域网
CiA	在自动化国际用户和制造商协会中的 CAN。
COB	通讯对象，在 CAN 网络上的一个传输单元。数据在 COB 内部沿着整个网络传输。COB 本身是 CAN 消息帧的一部分。
EDS	电子数据表，在配置 CAN 网络时需要使用的一个节点专用 ASCII- 格式文件。EDS 文件包含关于节点及其字典对象（参数）的常规信息。
LMT	层管理，CAN 给定模型中的 CAN 应用层服务元素之一。它用来配置CAN 给定模型中每层的参数。
NMT	网络管理， CAN 给定模型中的 CAN 应用层服务元素之一。它负责CAN 网络上的初始化、配置和故障处理。
OD	在本地存储某个设备所识别的所有通讯对象（COB）。
参数	参数是驱动器的一个操作指令。可以使用驱动器操作面板或者通过CAN来读取和修改参数。
PDO	进程数据对象，一种COB。用来传输时间关键数据，比如控制命令、给定值和实际值。
RO	表示只读访问。
RW	表示读/ 写访问。
SDO	服务数据对象，一种 COB。用来传输非时间关键数据，比如参数。

1.3 CANopen 概述

CANopen 是一个基于 CAN（控制局域网）串行总线系统和 CAL（CAN 应用层）的高层协议。CANopen 假定相连设备的硬件带有一个符合 ISO 11898 标准的 CAN 收发器和一个 CAN 控制器。

CANopen 通讯协议 CiA DS-301 包括周期和事件驱动型通讯，不仅能够将总线负载减少到最低限度，而且还能确保极短的反应时间。它可以在较低的波特率下实现较高的通讯性能，从而减少了电磁兼容性问题，并降低了电缆成本。

CANopen 设备协议定义了直接访问变频器参数机制以及时间关键进程数据通讯。NCAN-02 满足 CiA（自动化中的 CAN）标准 DSP-402（变频器和运动控制），只支持“制造商专用”操作模式。

CANopen 所用的物理介质是符合 ISO 11898 标准，采用分驱动机制和公共反馈的双线总线。总线的最大长度取决于通讯速度，具体规定如下：

通讯波特率	最大总线长度
1M bit/s	25 m
500k bit/s	100 m
250k bit/s	250 m
125k bit/s	500 m
100k bit/s	600 m
50k bit/s	1000 m

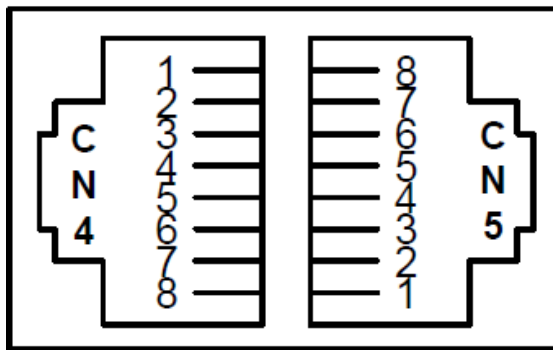
从理论上来说，最多可以有127 个节点。不过，在实际应用中，最大节点数量取决于所用 CAN 收发器的性能。

更多信息可参见自动化国际用户和制造商协会的 CAN 文献（www.can-cia.de）。

2、接线和连接

通讯用连接器（CN4/CN5）的端子信号名称及其功能如下：

端子号	1	2	3	4	5	6	7	8	
名称	CN4	CANH	CANL	GND	GND	RS485+	RS485-	保留	保留
	CN5	CANH	CANL	GND	GND	RS485+	RS485-	内置 120 欧电阻	

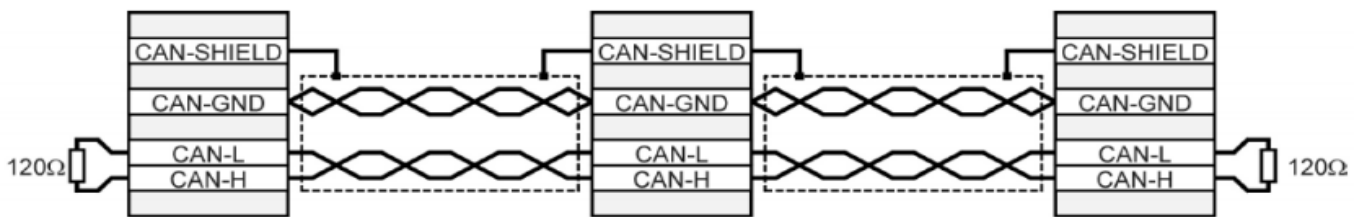


驱动器CN4 总是作为通讯电缆输入端子，CN5 总是作为通讯电缆输出端子（如果还需连接从站，电缆从该端子连接到下一从站设备；如果不需连接其他从站，可以在该端子加终端电阻）。多台驱动器连接时，严禁直连任意 2 台驱动的 CN4。

举例，网络由一 PLC 和 A、B、C 三台驱动器组成，电缆接线如下：

PLC → 驱动器A 的CN4, A 的CN5 → 驱动器B 的CN4, 驱动器B 的CN5 → 驱动器 C 的CN4, 驱动器C 的CN5 → 120 欧终端电阻。

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



总线电缆请选用有两对双绞线的带屏蔽层电缆：一对双绞线分别接 CAN-L 和 CAN-H，另外一对双绞线直接接 DGND。

3、CANopen 通讯

CAL 提供了所有的网络管理服务和报文传送协议，但并没有定义对象的内容或者正在通讯的对象的类型（它只定义了how，没有定义what），而这正是CANopen 切入点。

CANopen 是在CAL 基础上开发的，使用了CAL 通讯和服务协议子集，提供了分布式控制系统的一种实现方案。CANopen 在保证网络节点互用性的同时允许节点的功能随意扩展：或简单或复杂。

CANopen 的核心概念是设备对象字典（OD: Object Dictionary），在其它现场总线（Profibus, Interbus-S）系统中也使用这种设备描述形式。CANopen 通讯通过对象字典（OD）能够访问驱动器的所有参数。注意：对象字典不是CAL 的一部分，而是在CANopen 中实现的。

CANopen 通讯模型定义了如下几种报文（通讯对象）：

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

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

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

本驱动器暂不支持远程帧。其中COB-ID（通讯对象标识符）分配：

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

3.1 CAN 标识符分配表

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

注意：

- 1、 PDO/SDO 的发送/接受是由（slave）CAN 节点方观察的。
- 2、 本驱动器的CANopen 暂支持2 个发送PDO， 2 个接受PDO。

3.2 服务数据对象 SDO

SDO 用来访问一个设备的对象字典。访问者被称作客户（client），对象字典被访问且提供所请求服务的CANopen 设备别称作服务器(server)。客户的CAN 报文和服务器的应答CAN 报文总是包含8 字节数据（尽管不是所有的数据字节都一定有意义）。一个客户的请求一定有来自服务器的应答。

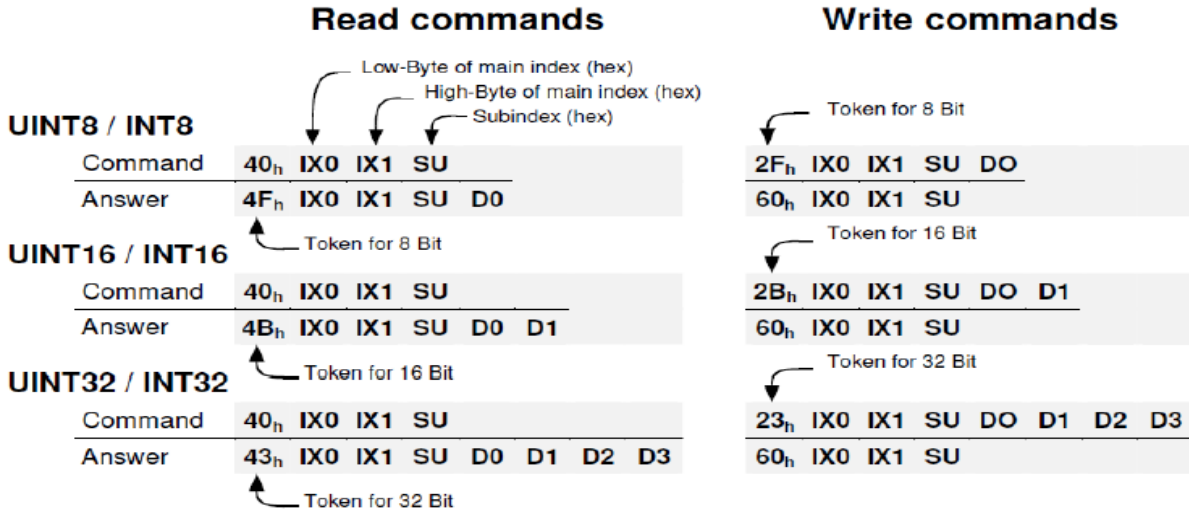
SDO 有2 种传送机制：

- 加速传送（Expedited transfer）： 最多传输4 字节数据
- 分段传送（Segmented transfer）： 传输数据长度大于4 字节

SDO 基本结构如下:

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

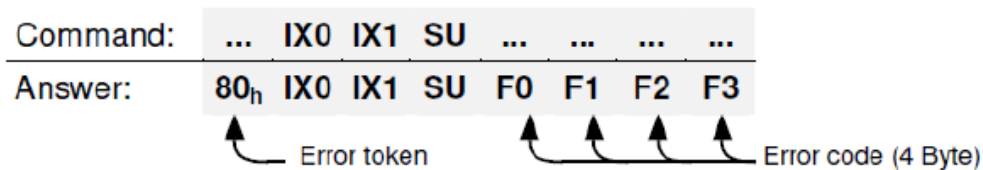
SDO 报文对参数读/写操作格式:



举例:

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

SDO 错误报文格式:



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

3.3 过程数据对象 PDO

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

每个PDO 在对象字典中用2 个对象描述：

- PDO 通讯参数：包含将被PDO 使用的COB-ID，传输类型，禁止时间和定时器周期。
- PDO 映射参数：包含一个对象字典中对象的列表，这些对象映射到PDO 里，包括它们的数据长度（in bits）。生产者和消费者必须知道这个映射，以解释PDO 内容。

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

动器支持动态PDO 映射。PDO 映射的必须遵守以下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 对应映射参数（1600 h 或1601 h 或1A00 h 或1A01 h）子索引0 的内容为0；
- 2、 修改PDO 对应映射参数（1600 h 或1601 h 或1A00 h 或1A01 h）子索引1~4 的内容，映射数据；
- 3、 设置PDO 对应映射参数（1600 h 或1601 h 或1A00 h 或1A01 h）子索引0 的内容为合法的数字（该PDO 映射的对象数）。
- 4、 PDO 映射完成。

PDO 可以有多种传送方式：

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

PDO 传输类型定义表

传输类型值	描述	PDO
0	保留	—
1~240	SYNC 方式： 该值代表两个PDO 之间的SYNC 对象的数目	TPDO/RPDO
240~253	保留	—
254	异步方式： 如果PDO 内容发生变化，触发该PDO 发送	TPDO
255	异步方式： PDO 内容周期性更新和发送	TPDO/RPDO

一个PDO 可以指定一个禁止时间，即定义两个连续PDO 传输的最小间隔时间，避免由于高优先级信息的数据量太大，始终占据总线，而使其它优先级较低的数据无力竞争总线的问题。禁止时间由16 位无符号整数定义，单位100us。

一个PDO 可以指定一个事件定时周期，当超过定时时间后，一个PDO 传输可以被触发（不需要触发位）。事件定时周期由16 位无符号整数定义，单位1ms。

PDO 映射事例：

将下表中3 个对象映射PDO1（发送），PDO1（发送）为异步周期性类型，周期时间10ms，禁止时间2ms。

对象	索引 — 子索引	说明
statusword	6041h – 00 h	状态字
modes_of_operation_display	6061h – 00 h	实际操作模式
Position_Acture_Value	6064h – 00 h	实际位置

1)、清除number_of_mapped_objects

number_of_mapped_objects(1A00 h: 00 h)= 0

2)、设置映射对象参数

Index =6041h Subin. = 01h Length = 10 h ⇒ 1st_mapped_object(1A00 h: 01 h)= 60410010 h

Index =6061h Subin. = 02h Length = 08 h ⇒ 2st_mapped_object(1A00 h: 02 h)= 60610008 h

Index =6064h Subin. = 03h Length = 20 h ⇒ 3st_mapped_object(1A00 h: 03 h)= 60640020 h

3)、设置number_of_mapped_objects

number_of_mapped_objects(1A00 h: 00 h)= 3

4)、设置PDO 通讯参数

PDO1 (发送) 为异步周期性类型 ⇒ transmission_type (1800 h: 02 h)= FF h

禁止时间2ms(20×100us) ⇒ inhibit_time (1800 h: 03 h)= 14 h

周期时间10ms(10×1ms) ⇒ event_time (1800 h: 05 h)= 0A h

5)、PDO 映射完成。

3.3.1 PDO 参数

驱动器包含4 路接受PDO 和4 路发送PDO。以下列出了只列出了第1 路发送/接受PDO 详细的通讯参数和映射参数。另3 路发送/接受PDO 详细的通讯参数和映射参数与此相同。

Index	1800 h
Name	transmit_pdo_parameter_tpdo1
Object Code	RECORD
No. of Elements	4

Sub-Index	01 h
Description	cob_id_used_by_pdo_tpdo1
Data Type	UINT32
Access	RW
PDO Mapping	NO
Units	_____
Value Range	181 h...1FF h, Bit 31 may be set
Default Value	181 h

Sub-Index	02 h
Description	transmission_type_tpdo1
Data Type	UINT8
Access	RW
PDO Mapping	NO
Units	_____
Value Range	1...240,254,255
Default Value	255

Sub-Index	03 h
Description	inhibit_time_tpdo1
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	100µs
Value Range	_____
Default Value	100

Sub-Index	05 h
Description	event_time_tpdo1
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	1ms
Value Range	_____
Default Value	10

Index	1A00 h
Name	transmit_pdo_mapping_tpdo1
Object Code	RECORD
No. of Elements	2
Sub-Index	00 h
Description	number_of_mapped_objects_tpdo1
Data Type	UINT8

Access	RW
PDO Mapping	NO
Units	_____
Value Range	0...4
Default Value	2

Sub-Index	01 h
Description	first_mapped_object_tpdo1
Data Type	UINT32
Access	RW
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	见表

Sub-Index	02 h
Description	second_mapped_object_tpdo1
Data Type	UINT32
Access	RW
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	见表

Sub-Index	03 h
Description	third_mapped_object_tpdo1
Data Type	UINT32
Access	RW
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	见表

Sub-Index	04 h
-----------	------

Description	fourth_mapped_object_tpdo1
Data Type	UINT32
Access	RW
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	见表

1、T-PDO1

Index	Comment	Type	Acc.	Default Value
1800 h_00 h	number of entries	UINT8	RO	04 h
1800 h_01 h	COB-ID used by PDO	UINT32	RW	00000181 h
1800 h_02 h	transmission type	UINT8	RW	FF h
1800 h_03 h	inhibit time (100 μs)	UINT16	RW	64 h
1800 h_05 h	event time (1ms)	UINT16	RW	0A h
1A00 h_00 h	number of mapped objects	UINT8	RW	02 h
1A00 h_01 h	first mapped object	UINT32	RW	60410010 h
1A00 h_02 h	second mapped object	UINT32	RW	60640020 h
1A00 h_03 h	third mapped object	UINT32	RW	00 h
1A00 h_04 h	fourth mapped object	UINT32	RW	00 h

2、T-PDO2

Index	Comment	Type	Acc.	Default Value
1801 h_00 h	number of entries	UINT8	RO	04 h
1801 h_01 h	COB-ID used by PDO	UINT32	RW	00000281 h
1801 h_02 h	transmission type	UINT8	RW	FF h
1801 h_03 h	inhibit time (100 μs)	UINT16	RW	64 h
1801 h_05 h	event time (1ms)	UINT16	RW	0A h
1A01 h_00 h	number of mapped objects	UINT8	RW	02 h
1A01 h_01 h	first mapped object	UINT32	RW	60640020 h
1A01 h_02 h	second mapped object	UINT32	RW	60610010 h
1A01 h_03 h	third mapped object	UINT32	RW	00 h
1A01 h_04 h	fourth mapped object	UINT32	RW	00 h

3、T-PDO3

Index	Comment	Type	Acc.	Default Value
1802 h_00 h	number of entries	UINT8	RO	04 h
1802 h_01 h	COB-ID used by PDO	UINT32	RW	00000281 h
1802 h_02 h	transmission type	UINT8	RW	FF h
1802 h_03 h	inhibit time (100 μ s)	UINT16	RW	64 h
1802 h_05 h	event time (1ms)	UINT16	RW	0A h
1A02 h_00 h	number of mapped objects	UINT8	RW	02 h
1A02 h_01 h	first mapped object	UINT32	RW	60640020 h
1A02 h_02 h	second mapped object	UINT32	RW	60610010 h
1A02 h_03 h	third mapped object	UINT32	RW	00 h
1A02 h_04 h	fourth mapped object	UINT32	RW	00 h

4、T-PDO4

Index	Comment	Type	Acc.	Default Value
1803 h_00 h	number of entries	UINT8	RO	04 h
1803 h_01 h	COB-ID used by PDO	UINT32	RW	00000281 h
1803 h_02 h	transmission type	UINT8	RW	FF h
1803 h_03 h	inhibit time (100 μ s)	UINT16	RW	64 h
1803 h_05 h	event time (1ms)	UINT16	RW	0A h
1A03 h_00 h	number of mapped objects	UINT8	RW	02 h
1A03 h_01 h	first mapped object	UINT32	RW	60640020 h
1A03 h_02 h	second mapped object	UINT32	RW	60610010 h
1A03 h_03 h	third mapped object	UINT32	RW	00 h
1A03 h_04 h	fourth mapped object	UINT32	RW	00 h

tpdo_1_transmit_mask

Index	Comment	Type	Acc.	Default Value
2000 h_00 h	number of entries	UINT8	RO	02 h
2000 h_01 h	tpdo_1_transmit_mask_low	UINT32	RW	FFFFFFFF h
2000 h_02 h	tpdo_1_transmit_mask_high	UINT32	RW	FFFFFFFF h

tpdo_2_transmit_mask

Index	Comment	Type	Acc.	Default Value
2001 h_00 h	number of entries	UINT8	RO	02 h

2001 h_01 h	tpdo_2_transmit_mask_low	UINT32	RW	FFFFFFFF h
2001 h_02 h	tpdo_2_transmit_mask_high	UINT32	RW	FFFFFFFF h

tpdo_3_transmit_mask

Index	Comment	Type	Acc.	Default Value
2002 h_00 h	number of entries	UINT8	RO	02 h
2002 h_01 h	tpdo_1_transmit_mask_low	UINT32	RW	FFFFFFFF h
2002 h_02 h	tpdo_1_transmit_mask_high	UINT32	RW	FFFFFFFF h

tpdo_4_transmit_mask

Index	Comment	Type	Acc.	Default Value
2003 h_00 h	number of entries	UINT8	RO	02 h
2003 h_01 h	tpdo_2_transmit_mask_low	UINT32	RW	FFFFFFFF h
2003 h_02 h	tpdo_2_transmit_mask_high	UINT32	RW	FFFFFFFF h

1、R-PDO1

Index	Comment	Type	Acc.	Default Value
1400 h_00 h	number of entries	UINT8	RO	02 h
1400 h_01 h	COB-ID used by PDO	UINT32	RW	00000201 h
1400 h_02 h	transmission type	UINT8	RW	FF h
1600 h_00 h	number of mapped objects	UINT8	RW	02 h
1600 h_01 h	first mapped object	UINT32	RW	60400010 h
1600 h_02 h	second mapped object	UINT32	RW	60FF0020 h
1600 h_03 h	third mapped object	UINT32	RW	00 h
1600 h_04 h	fourth mapped object	UINT32	RW	00 h

2、R-PDO2

Index	Comment	Type	Acc.	Default Value
1401 h_00 h	number of entries	UINT8	RO	02 h
1401 h_01 h	COB-ID used by PDO	UINT32	RW	00000301 h
1401 h_02 h	transmission type	UINT8	RW	FF h
1601 h_00 h	number of mapped objects	UINT8	RW	02 h
1601 h_01 h	first mapped object	UINT32	RW	60FF0020 h
1601 h_02 h	second mapped object	UINT32	RW	60600010 h

1601 h_03 h	third mapped object	UINT32	RW	00 h
1601 h_04 h	fourth mapped object	UINT32	RW	00 h

3、R-PDO3

Index	Comment	Type	Acc.	Default Value
1402 h_00 h	number of entries	UINT8	RO	02 h
1402 h_01 h	COB-ID used by PDO	UINT32	RW	00000301 h
1402 h_02 h	transmission type	UINT8	RW	FF h
1602 h_00 h	number of mapped objects	UINT8	RW	02 h
1602 h_01 h	first mapped object	UINT32	RW	60FF0020 h
1602 h_02 h	second mapped object	UINT32	RW	60600010 h
1602 h_03 h	third mapped object	UINT32	RW	00 h
1602 h_04 h	fourth mapped object	UINT32	RW	00 h

4、R-PDO4

Index	Comment	Type	Acc.	Default Value
1403 h_00 h	number of entries	UINT8	RO	02 h
1403 h_01 h	COB-ID used by PDO	UINT32	RW	00000301 h
1403 h_02 h	transmission type	UINT8	RW	FF h
1603 h_00 h	number of mapped objects	UINT8	RW	02 h
1603 h_01 h	first mapped object	UINT32	RW	60FF0020 h
1603 h_02 h	second mapped object	UINT32	RW	60600010 h
1603 h_03 h	third mapped object	UINT32	RW	00 h
1603 h_04 h	fourth mapped object	UINT32	RW	00 h

3.4 SYNC 报文

在网络范围内同步：在整个网络范围内当前输入值准同时保存，随后传送（如果需要），根据前一个 SYNC 后接收到的报文更新输出值。

主从模式：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 COB-ID 固定为 080h，COB-ID 可以从对象字典的 1005 h 读出。

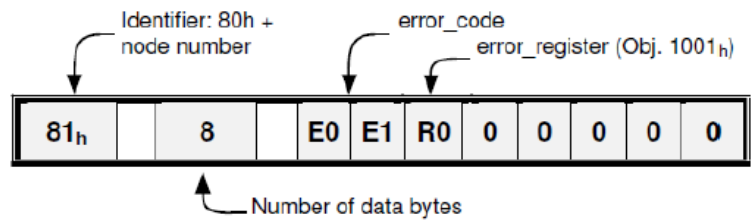
Index	1005 h
Name	cob_id_sync

Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	NO
Units	_____
Value Range	80000080 h, 00000080 h
Default Value	00000080 h

3.5 Emergency 报文

驱动器发生报警时，CANopen 会启动一条Emergency 报文，告知消费者当前驱动器类型及错误代码。

Emergency 报文结构：



报警代码表

error_code (16 进制)	说明
2310	过流报警
3100	瞬间停电报警
3110	过压报警
3120	欠压报警
5080	RAM 芯片异常报警
5210	AD 采样出错报警
5420	泻放电阻损坏报警
5421	再生异常报警
5581	参数和检验异常
5582	电子齿轮错
5583	电机型号参数或驱动器型号参数错误
6100	非法错误代码
6120	PDO 映射出错
6300	CAN 通讯参数出错 (地址或通讯波特率参数错)
7303	串行编码器出错

7305	增量编码器出错
7380	旋转编码器出错
8100	CAN 通讯异常
8110	CAN 总线溢出
8120	CAN 总线进入PASSIVE
8130	心跳出错
8140	CAN 总线BUS OFF
8200	CAN 接受信息长度出错
8210	接受PDO 长度出错
8311	过载报警
8480	超速报警

相关参数

Index	1003 h
Name	pre_defined_error_field
Object Code	ARRAY
No. of Elements	4
Data Type	UINT32

Sub-Index	01 h
Description	standard_error_field_0
Access	RO
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	_____

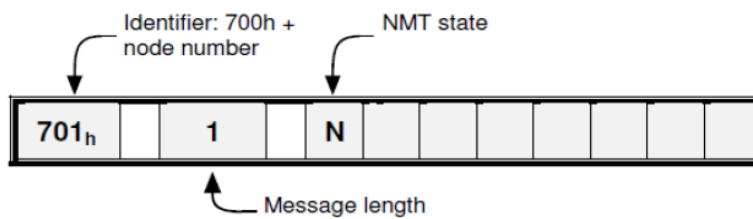
Sub-Index	02 h
Description	standard_error_field_1
Access	RO
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	_____

Sub-Index	03 h
Description	standard_error_field_2
Access	RO
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	_____

Sub-Index	04h
Description	standard_error_field_3
Access	RO
PDO Mapping	NO
Units	_____
Value Range	_____
Default Value	_____

3.6 HEARTBEAT 报文

报文结构:

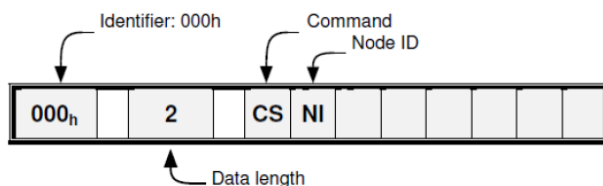


相关参数:

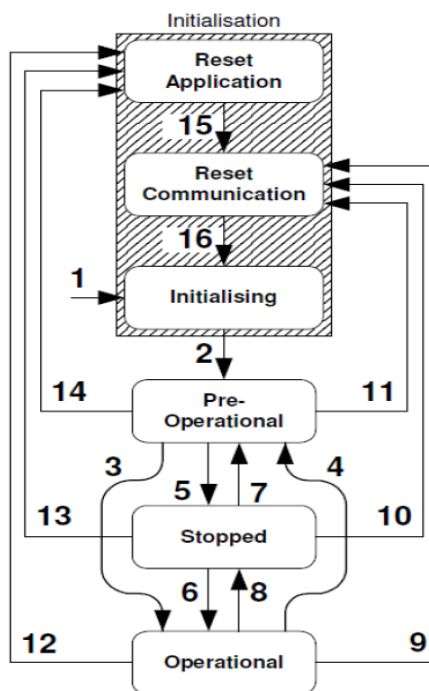
Index	1017 h
Name	producer_heartbeat_time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	ms
Value Range	0 - 65535
Default Value	0

3.7 网络管理 (NMT)

报文结构:



网络管理状态转换图:

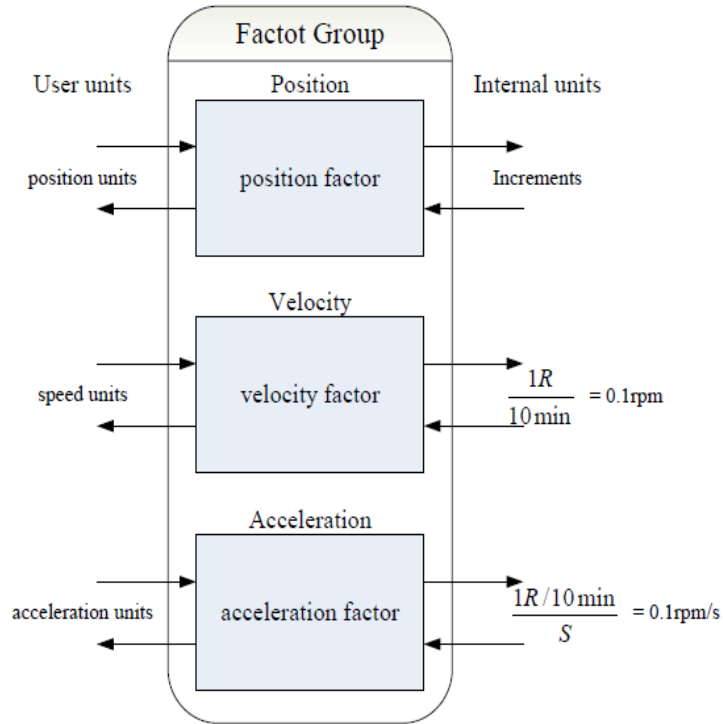


CS	Meaning	Transition	Target state
01 _h	Start Remote Node	3, 6	Operational
02 _h	Stop Remote Node	5, 8	Stopped
80 _h	Enter Pre-Operational	4, 7	Pre-Operational
81 _h	Reset Application	12, 13, 14	Reset Application
82 _h	Reset Communication	9, 10, 11	Reset Communication

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

4、单位换算单元（Factor Group）

伺服驱动器被广泛应用在各种场合。为了在各类应用时都能方便地设置参数，驱动器内部的单位换算单元可以将任意用户单位参数值自动与驱动器内部单位值自动互换。



驱动器默认的用户单位：

对象	名称	单位	说明
长度	position units	Increments	脉冲 *
速度	speed units	1R / 10min	0.1rpm
加速度	Acceleration units	1R/10min/s	0.1rpm/s

*：普通增量编码器每转1 圈输出10000 脉冲；

旋转编码器每转1 圈输出65536 脉冲；

17 位增量编码器每转1 圈输出131072 脉冲；

17 位绝对值编码器每转1 圈输出131072 脉冲

4.1 单位换算相关参数

Index	Object	Name	Type	Attr.
6093 h	ARRAY	position factor	UINT32	RW
6094 h	ARRAY	velocity factor	UINT32	RW
6097 h	ARRAY	acceleration factor	UINT32	RW

注:

编码器类型	encoder_resolution(Unit: Inc)
普通增量编码器	10000
旋转变压器编码器	65535
17 位增量型编码器	131072
17 位绝对值编码器	131072

4.1.2 velocity factor

velocity factor 模块将用户端的所有速度单位转换成驱动器内部单位 (0.1rpm)，同时也将驱动器输出的速度单位 (0.1rpm) 转换成用户端的速度单位 (velocity units)。velocity factor 参数组包含: numerator (分子) 和 division (分母)。

Index	6094 h
Name	velocity factor
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01 h
Description	numerator
Access	RW
PDO Mapping	YES
Units	_____
Value Range	_____
Default Value	1

Sub-Index	02 h
Description	division
Access	RW
PDO Mapping	YES
Units	_____
Value Range	_____
Default Value	1

为了方便的计算出 velocity factor,定义下面 3 个变量:

time_factor_v: 驱动器内时间单位和用户时间单位比（例： 1min = 1/10 10min）

gear_ratio: 负载轴和电机轴的减速比（电机转n 圈，负载轴转m 圈， **gear_ratio** = m/n）

feed_constant: 负载轴每旋转1 圈带动负载运动的用户单位（position units）距离
position factor 的计算公式：

$$\text{velocity factor} = \frac{\text{numerator}}{\text{division}} = \frac{\text{gear_ratio} * \text{time_factor_v}}{\text{feed_constant}}$$

4.1.3 acceleration factor

acceleration factor 模块将用户端的所有加速度单位转换成驱动器内部单位（0.1rpm），同时也将驱动器输出量的加速度单位（0.1rpm/s）转换成用户端的加速度单位（acceleration units）。acceleration factor。参数组包含： numerator（分子） 和 division（分母）。

Index	6094 h
Name	acceleration factor
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01 h
Description	numerator
Access	RW
PDO Mapping	YES
Units	_____
Value Range	_____
Default Value	1

Sub-Index	02 h
Description	division
Access	RW
PDO Mapping	YES
Units	_____
Value Range	_____
Default Value	1

为了方便的计算出velocity factor,定义下面3 个变量：

time_factor_a: 驱动器内时间单位平方和用户时间平方单位比

(例: $1\text{min}^2 = 1\text{min} * \text{min} = 60\text{s} * 1\text{min} = 60/10 \mathbf{10\text{min/s}}$)

gear_ratio: 负载轴和电机轴的减速比, (电机转n 圈, 负载轴转m 圈, $\text{gear_ratio} = m/n$)

feed_constant: 负载轴每旋转1圈带动负载运动的用户单位 (position units) 距离
position factor 的计算公式:

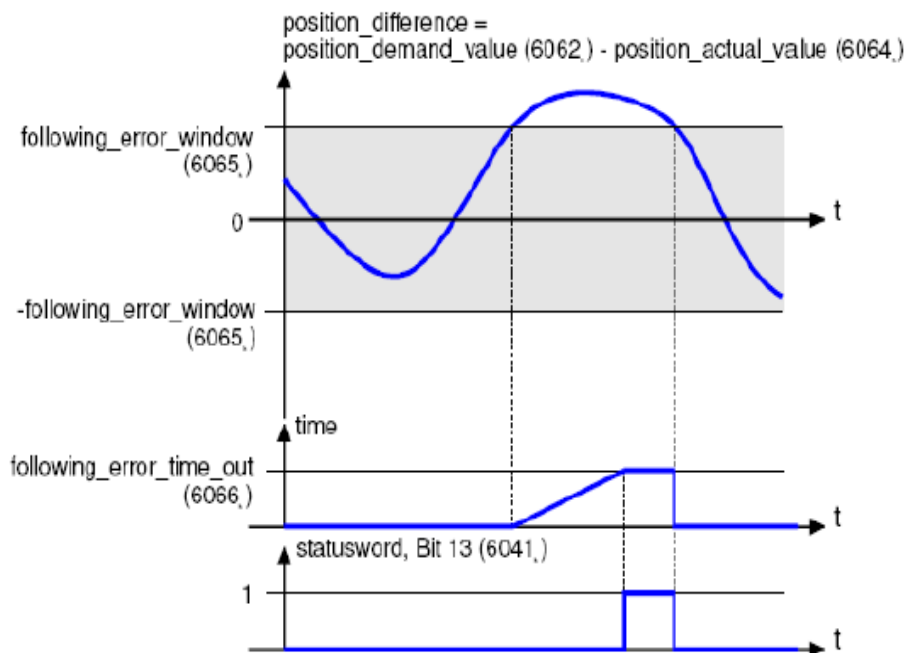
$$\text{acceleration factor} = \frac{\text{numerator}}{\text{division}} = \frac{\text{gear_ratio} * \text{time_factor_a}}{\text{feed_constant}}$$

5、位置控制功能

本章节主要描述位置控制模式下所需要的参数。Trajectory 单元输出的期望位置(position_demand_value)作为驱动器位置环的输入。此外, 实际位置(position_actual_value)是通过电机编码器测量出。位置控制器的动作受参数设置影响。为了控制系统的稳定性, 必须对位置环输出值(control_effort)限幅。该输出值作为速度环给定速度。在Factor group 单元, 所有的输入和输出值都转换成驱动器相应的内部单位。

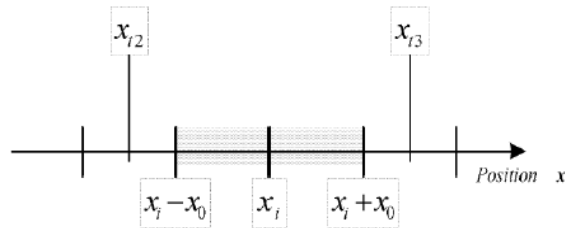
下面介绍位置控制时的子功能:

1、跟随误差(following error)



跟随误差(following error) -功能描述

跟随误差指的是实际位置 (position_actual_value) 和期望位置 (position_demand_value) 的偏差。上图所示, 如果在设定时间 (following_error_time_out) 内, 跟随误差值大于跟随误差窗口 (following_error_windows), 那么状态字 (statusword) 的 bit13 (following_error) 将被置1。

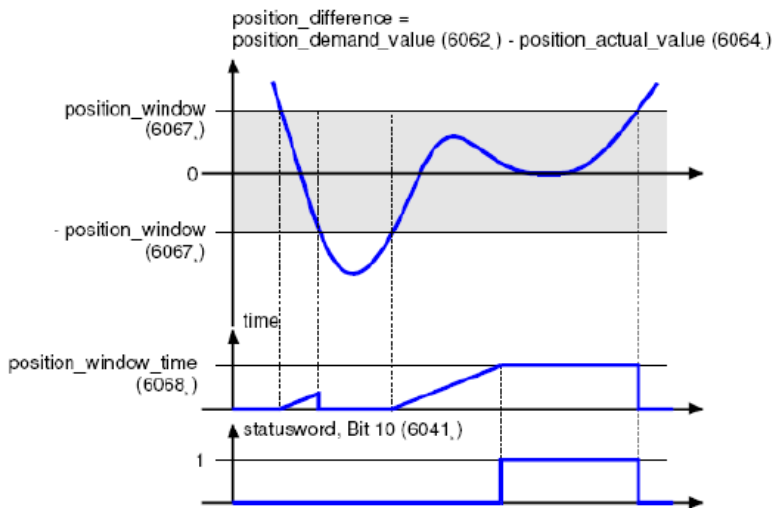


跟随误差(following error) – 举例

上图描述了对“following error”，窗口功能是如何定义。 $x_i - x_0$ 和 $x_i + x_0$ 范围(位置跟随误差窗口)对称地分布在期望位置 (position_demand_value) 两边。举例，位置 x_{i2} 和 x_{i3} 都位于位置跟随误差窗口外。如果驱动器离开窗口后，在设定时间 (following_error_time_out) 内没有返回窗口内，那么状态字 (statusword) 的bit13 (following_error) 将被置1。

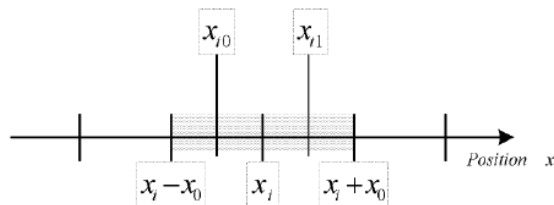
2、位置到达

该功能定义了目标位置(target_position)附近的位置窗口。如果驱动器的实际位置稳定在这个范围(位置窗口)达到设定时间(position_windows_time)，那么状态字(statusword)的bit10(target_reached)将被置1。如下图所示。



位置到达 – 功能描述

下图显示 位置窗口 (position_windows) 对称分布在目标位置 (target_position) 附近，即 $x_i - x_0$ 到 $x_i + x_0$ 区间范围。举例，位置 x_{i0} 和 x_{i1} 在位置窗口 (position_windows) 内。如果驱动器在窗口内，一个定时期开始计时。如果定时期达到设定值 (position_window_time) 且期间驱动器位置一直位于窗口内，那么状态字 (statusword) 的bit10(target_reached)将被置1。驱动器位置一离开该窗口，状态字的bit10(target_reached)将立即被清零。



5.1 位置控制相关参数

Index	Object	Name	Type	Attr.
6062 h	VAR	position_demand_value	INT32	RO
6063 h	VAR	position_actual_value*	INT32	RO
6064 h	VAR	position_actual_value	INT32	RO
6065 h	VAR	following_error_window	UINT32	RW
6066 h	VAR	following_error_time_out	UINT16	RW
6067 h	VAR	position_window	UINT32	RW
6068 h	VAR	position_time	UINT16	RW
60FA h	VAR	control_effort	INT32	RO

Index	6062 h
Name	position_demand_value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES
Units	position units
Value Range	--
Default Value	--

Index	6064 h
Name	position_ actual _value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES
Units	position units
Value Range	--
Default Value	--

Index	6065 h
Name	following_error_window

Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	position units
Value Range	0 – 7FFFFFFF h
Default Value	256

Index	6066 h
Name	following_error_time_out
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	ms
Value Range	0 – 65535
Default Value	0

Index	60FA h
Name	control_effort
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	--

Index	6067 h
Name	position_window
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES

Units	position units
Value Range	--
Default Value	400

Index	6068 h
Name	position_time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	ms
Value Range	0 – 65535
Default Value	0

6、设备控制

本章节主要描述主机如何通过CANopen 总线控制驱动器，比如控制伺服ON 或清除报警等。

6.1 控制状态机

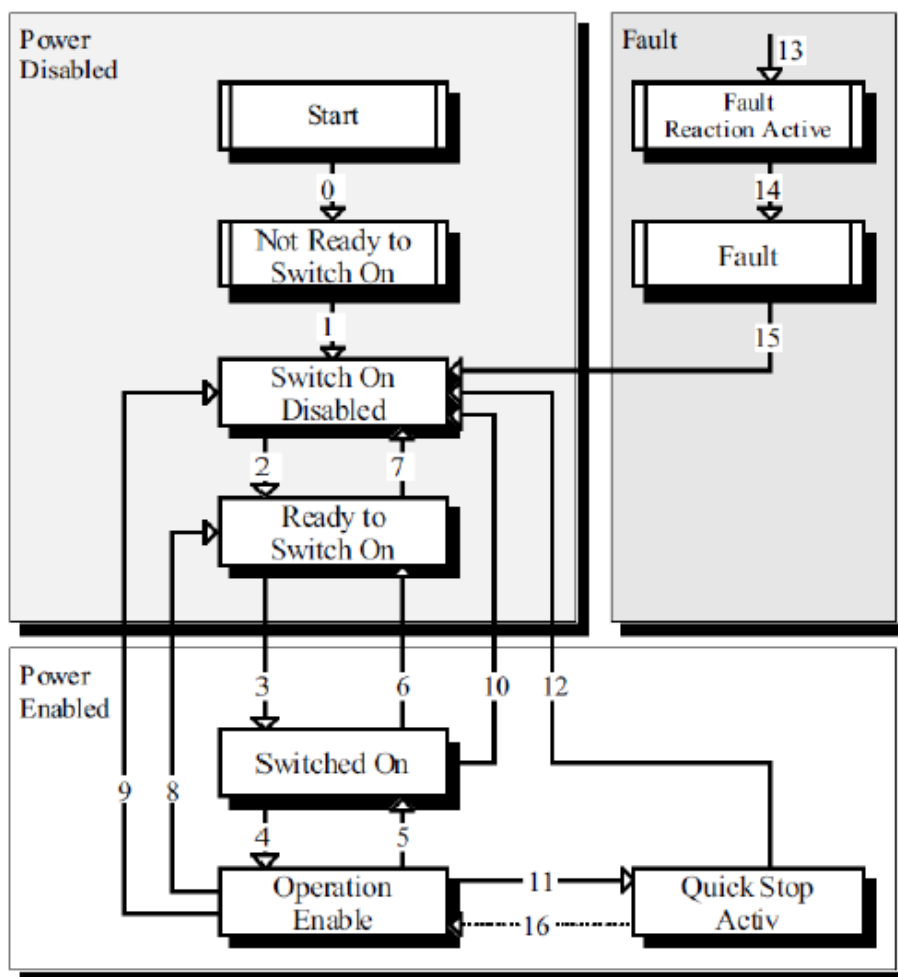
主站通过controlword（控制字）对驱动器的控制，通过读驱动器的statusword（状态字）能知道驱动器当前状态。本章节将使用到下面的一些术语：

State(状态): 如果主电激活或发生报警，伺服驱动都处在不同状态。CANopen 总线控制下的状态机将在本节重点讲解。例如：**SWITCH_ON_DISABLED**

State Transition(状态传输): 状态机也定义了如何从一个状态转移到另一个状态。状态转移依靠主站控制的controlword 或驱动器自身，例如驱动器发生报警。

Command(命令): 为了启动State Transition(状态传输)，定义了controlword 的位组合，这些位组合被称作Command(命令)。

State diagram(状态图): 所有的 State (状态)和 State Transition (状态传输)就组成了 State diagram(状态图)。



状态机图

如上图所示，状态机可以分成三部分：“Power Disabled”（主电关闭）、“Power Enabled”（主电打开）和“Fault”。所有状态在发生报警后均进入“Fault”。在上电后，驱动器完成初始化，然后进入 SWITCH_ON_DISABLED 状态。在该状态，可以可以进行CAN 通讯，可以对驱动器进行配置（例如，将驱动器的工作模式设置成“PP”模式）。此时，主电仍然关闭，电机没有被励磁。经过State Transition(状态传输)2、3、4 后，进入OPERATION ENABLE。此时，主电已开启，驱动器根据配置的工作模式控制电机。因此，在该状态之前必须先确认已经正确配置了驱动器的参数和相应的输入值为零。State Transition(状态传输)9 完成关闭电路主电。一旦驱动器发生报警，驱动器的状态都进入FAULT。

状态名	说明
Not Ready to Switch On	伺服驱动器正在初始化过程中，不能进行 CAN 通讯。
Switch On Disabled	伺服驱动器初始化完成，可以进行 CAN 通讯。
Ready to Switch On	伺服驱动器等待进入 Switch On 状态，电机没有被励磁。
Switched On	伺服驱动器伺服准备好状态，主电已上。
Operation Enable	伺服驱动器伺服给电机输入励磁信号，按照控制模式控制电机。
Quick Stop Active	伺服驱动器将根据设定的方式停机。

Fault Reaction Active	伺服驱动器检测到报警发生，按照设定的方式停机，电机仍然有励磁信号。
Fault	电机无励磁信号。

6.2 设备控制相关参数

Index	Object	Name	Type	Attr.
6040 h	VAR	controlword	UINT16	RW
6041 h	VAR	statusword	UINT16	RO
605A h	VAR	quick_stop_option_code	INT16	RW
605B h	VAR	shutdown_option_code	INT16	RW
605C h	VAR	disabled_operation_option_code	INT16	RW
605D h	VAR	halt_option_code	INT16	RW
605E h	VAR	fault_reaction_option_code	INT16	RW

6.2.1 controlword


Index	6040 h
Name	controlword
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	--
Default Value	0

controlword 位说明如下图所示：

15	11	10	9	8	7	6	4	3	2	1	0
manufacturer specific	reserved	halt	Fault reset	Operation mode specific	Enable operation	Quick stop	Enable voltage	Switch on			

Bit0 ~ 3 和 Bit7:

状态机的传输由这 5 位组成的相应控制命令触发。

Command	Bit of the <i>controlword</i>					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3*
Switch on	0	1	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		X	X	X	X	15

设备控制命令表

注：表中 X 表示该位可以被忽略；

Bit4、5、6、8：

这 4 位在不同控制模式下，定义不同。

Bit	控制模式		
	profile position mode	profile velocity mode	homing mode
4	new_set_point	保留	start_homeing_operation
5	change_set_immediatly	保留	保留
6	abs/rel	保留	保留
8	Halt	Halt	Halt

其他位： 均是保留位。

6.2.2 statusword

Index	6041 h
Name	statusword
Object Code	VAR
Data Type	UINT16
Access	RO
PDO Mapping	YES
Units	--
Value Range	--
Default Value	--

statusword 位说明如下图所示

位	说明
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
9~8	保留
10	Target reached
11	Internal limit active
13~12	Operation mode specific
15~14	保留

Bit0 ~ 3 、Bit5 和 Bit6:

这几位的组合表示驱动器所处的状态。

Value (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Bit4: Voltage enabled

当该位为1 时，表示主电已上。

Bit5: Quick stop

当该位为0 时，表示驱动器将按照设置（605A h: quick_stop_option_code）停机。

Bit7: Warning

当该位为1 时，表示驱动器检测到报警。

Bit10: Target reached

该位在不同控制模式下，含义不同。

Profile Position Mode 时，当设定位置到达后，该位将被置位；当Halt 启动，速度减速到零后，该位将被置位；当新的位置设定后，该位将被清除。

Profile Velocity Mode 时，当速度到达给定速度后，该位将被置位；当Halt 启动，速度减速到零后，该位将被置位。

Bit11: Internal limit active

当该位为1 时，表示内部转矩超过设定值。

Bit12、13:

该 2 位在不同控制模式下，含义不同。

Bit	控制模式		
	profile position mode	profile velocity mode	homing mode
12	Set-point acknowledge	Speed	Homing attained
13	Following error	Max slippage error	Homing error

其他位：均是保留位。

6.2.3 shutdown_option_code

当发生 Operation Enable → Ready to Switch On 状态转移时，shutdown_option_code 决定如何停机。

Index	605B h
Name	shutdown_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0,1
Default Value	0

值	说明
0	关闭电机励磁信号，电机自由停止。
1	电机减速停止后，关闭电机励磁信号。

6.2.4 disable_operation_option_code

当发生 Operation Enable → Switched On 状态转移时, disable_operation_option_code 决定如何停机。

Index	605C h
Name	disable_operation_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0,1
Default Value	0

值	说明
0	关闭电机励磁信号, 电机自由停止。
1	电机减速停止后, 关闭电机励磁信号。

6.2.5 quick_stop_option_code

当发生 Operation Enable → Quick Reaction Active 状态转移时, quick_stop_option_code 决定如何停机。

Index	605A h
Name	quick_stop_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0,1,2,5,6
Default Value	0

值	说明
0	关闭电机励磁信号, 电机自由停止。
1	电机减速停止后, 关闭电机励磁信号。
2	电机紧急减速停止后, 关闭电机励磁信号
5	电机减速停止后, 仍然停留在 QuickStop 状态。
6	电机紧急减速停止后, 仍然停留在 QuickStop 状态

6.2.6 halt_option_code

当 controlword 的 bit8(halt)为 1 时, halt_option_code 决定如何停机。.

Index	605D h
Name	halt_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	1,2
Default Value	0

值	说明
1	电机减速停止
2	电机紧急减速停止

6.2.7 fault_reaction_option_code

当检测到报警时, fault_reaction_option_code 决定如何停机。

Index	605D h
Name	fault_reaction_option_code
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	NO
Units	--
Value Range	0
Default Value	0

值	说明
0	关闭电机励磁信号, 电机自由停止。

7、控制模式

伺服驱动器目前支持CANopen DSP402 中的3 种控制模式:

回零模式（HOMING MODE）

速度控制模式（PROFILE VELOCITY MODE）

位置控制模式（PROFILE POSITION MODE）

本章节主要描述上述的三种控制方式。

7.1 控制模式相关参数

Index	Object	Name	Type	Attr.
6060 h	VAR	modes_of_operation	INT8	RW
6061 h	VAR	modes_of_operation_display	INT8	RO

7.1.1 modes_of_operation

伺服驱动器的控制模式由 modes_of_operation 参数决定。

Index	6060 h
Name	modes_of_operation
Object Code	VAR
Data Type	INT8
Access	RW
PDO Mapping	YES
Units	--
Value Range	1,3,6
Default Value	0

值	说明
0	未置控制模式（NOP MODE）
1	位置控制模式（PROFILE POSITION MODE）
3	速度控制模式（PROFILE VELOCITY MODE）
6	回零模式（HOMING MODE）

7.1.2 modes_of_operation_display

伺服驱动器当前的控制模式可以通过读 modes_of_operation_display 参数知道。

Index	6061 h
Name	modes_of_operation_display
Object Code	VAR

Data Type	INT8
Access	RO
PDO Mapping	YES
Units	--
Value Range	1,3,6
Default Value	0

- 注：1、只能通过**modes_of_operation_display** 参数才能知道驱动器当前所处的控制模式；
 2、只有当**TargetReached** 状态时，驱动器的控制模式才会从当前模式切换到设置的控制模式，
modes_of_operation_display 才能跟 **modes_of_operation** 一致。

7.2 回零模式（HOMING MODE）

伺服驱动器目前支持多种回零方式，用户应该合理选择相应的回零方式。例如：如果电机使用的是增量编码器，那么可以选择通过C 脉冲的回零方式；如果电机使用的是串行编码器或旋转变压器，那么就不能选择通过C 脉冲的回零方式。

用户可以设置回零方式、回零的速度和加速度。驱动器在找到参考点位置后，可以设置零位位置向前偏移该参考点home_offset（607C h）距离。

7.2.1 回零模式的控制字

15 ~ 9	8	7 ~ 5	4	3 ~ 0
*	Halt	*	home_start_operation	*

*: 参考前面章节

Name	Value	Description
Homing operation start	0	Homing mode inactive
	0→1	Start homing mode
	1	Homing mode active
	1→0	Interrupt homing mode
Halt	0	Execute the instruction of bit 4
	1	Stop axle with homing acceleration

7.2.2 回零模式的状态字

15 ~ 14	13	12	11	10	9 ~ 0
*	homing_error	homing_attained	*	target_reached	*

*: 参考前面章节

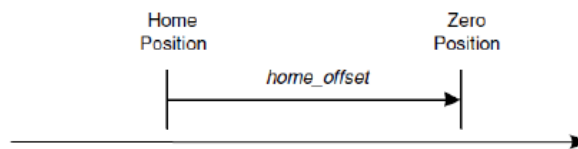
Name	Value	Description
Target reached	0	Halt=0:Home position not reached Halt=1:Axle decelerates
	1	Halt=0:Home position reached Halt=1:Axle has velocity 0
Homing attained	0	Homing mode not yet completed
	1	Homing mode carried out successfully
Homing error	0	No homing error
	1	Homing error occurred; Homing mode carried out not successfully; The error cause is found by reading the error code

7.2.3 回零模式相关参数

Index	Object	Name	Type	Attr.
607C h	VAR	home_offset	INT32	RW
6098 h	VAR	homing_method	INT8	RW
6099 h	ARRAY	homing_speeds	UINT32	RW
609A h	VAR	homing_acceleration	INT32	RW

home_offset

home_offset 参数确定了参考点位置与零位之间的距离。



Index	607C h
Name	home_offset
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	position units
Value Range	--
Default Value	0

homing_method

有 4 种信号可作为回零信号：正限位开关、负限位开关、参考点开关和 C 脉冲。

Index	6098 h
Name	homing_method
Object Code	VAR
Data Type	INT8
Access	RW
PDO Mapping	YES
Units	--
Value Range	1,2,3,4,17,18,19,20
Default Value	1

回零方式表格

方式	方向	目标位置	参考点位置	DS402
1	负	NOT	C 脉冲	1
2	正	POT	C 脉冲	2
3	负	参考点开关	C 脉冲	3
4	正	参考点开关	C 脉冲	4
17	负	NOT	NOT	17
18	正	POT	POT	18
19	负	参考点开关	参考点开关	19
20	正	参考点开关	参考点开关	20

homing_speeds

找参考点需要 2 种速度：找参考点速度和找零位速度。

Index	6099 h
Name	homing_speeds
Object Code	ARRAY
No. of Elements	2
Data Type	INT32

Sub-Index	01 h
Name	speed_during_search_for_switch
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES

Units	speed units
Value Range	--
Default Value	0

Sub-Index	02 h
Name	speed_during_search_for_zero
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

homing_acceleration

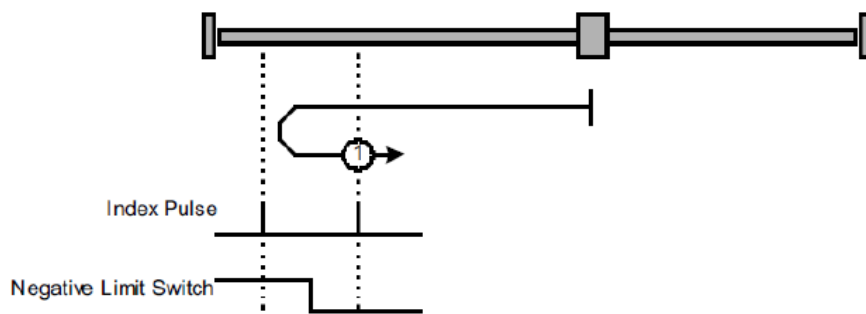
回零期间的加速度和减速度，均是 homing_acceleration。

Index	609A h
Name	homing_acceleration
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	0

7.2.4 回零方法

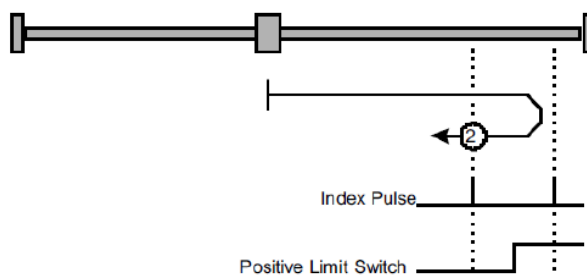
方法1 使用C 脉冲和负限位开关

驱动器首先较快的向负方向移动，直到撞到负限位开关（N-OT）才减速停止；然后驱动器慢速返回，寻找目标零位位置。本回零方法的目标零位位置是离开限位开关后编码器的第一个C 脉冲位置。



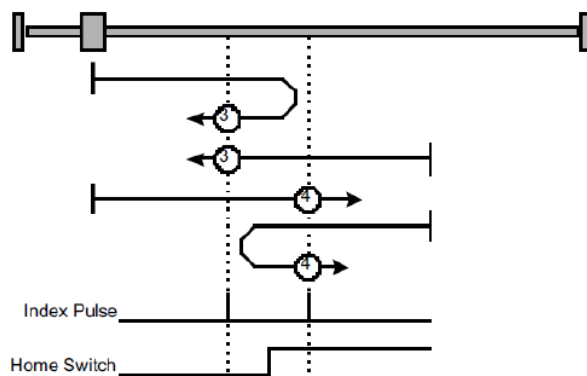
方法2 使用C 脉冲和正限位开关

驱动器首先较快的向正方向移动，直到撞到正限位开关（P-OT）才减速停止；然后驱动器慢速返回，寻找目标零位位置。本回零方法的目标零位位置是离开限位开关后编码器的第一个C 脉冲位置。



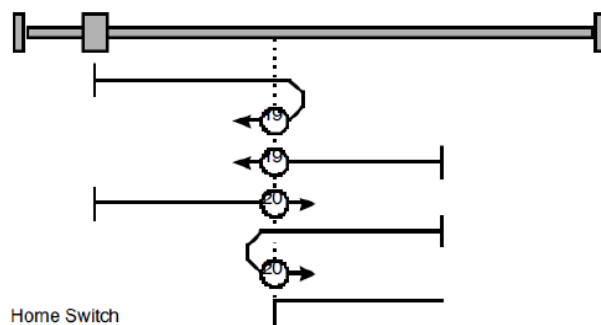
方法3 和4 使用C 脉冲和参考点开关

驱动器初始方向移动依赖于参考点开关状态。目标零位位置是参考点开关左边或右边的第一个C 脉冲位置。



方法17 ~ 20 不使用C 脉冲

这4 种回零方法分别类似方法1~4，只是目标零位位置不再使用C 脉冲，而依赖限位开关或参考点开关的变化。例，如下图所示方法19 和20 回零方法分别类似方法3~4。



7.3 速度控制模式（PROFILE VELOCITY MODE）

7.3.1 速度模式的控制字

15 ~ 9	8	7 ~ 4	3 ~ 0
*	Halt	*	*

*: 参考前面章节

Name	Value	Description
Halt	0	Execute the motion
	1	Stop axle

7.3.2 速度模式的状态字

15 ~ 14	13	12	11	10	9 ~ 0
*	MaxSlippageError	Speed	*	Target_reached	*

*: 参考前面章节

Name	Value	Description
Target reached	0	Halt=0:Target velocity not (yet) reached Halt=1:Axle decelerates
	1	Halt=0:Target velocity reached Halt=1:Axle has velocity 0
Speed	0	Speed is not equal 0
	1	Speed is equal 0
Max slippage error	0	Maximum slippage not reached
	1	Maximum slippage reached

7.3.3 速度控制模式相关参数

Index	Object	Name	Type	Attr.
6069 h	VAR	velocity_sensor_actual_value	INT32	RO
606B h	VAR	velocity_demand_value	INT32	RO
606C h	VAR	velocity_actual_value	INT32	RO
609D h	VAR	velocity_window	UINT16	RW

606E h	VAR	velocity_window_time	UINT16	RW
606F h	VAR	velocity_threshold	UINT16	RW
6070 h	VAR	velocity_threshold_time	UINT16	RW
60FF h	VAR	target_velocity	INT32	RW

velocity_sensor_actual_value

主站可以通过读 velocity_sensor_actual_value 参数确定电机当前转速，该参数单位是内部速度单位。

Index	6069 h
Name	velocity_sensor_actual_value
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	0.1rmps (1R/10min)
Value Range	--
Default Value	--

velocity_demand_value

主站可以通过读 velocity_demand_value 参数确定驱动器当前给定速度值，该参数单位是用户速度单位。

Index	606B h
Name	velocity_demand_value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	--

velocity_actual_value

主站可以通过读 velocity_demand_value 参数确定电机当前转速，该参数单位是用户速度单位。

Index	606C h
Name	velocity_actual_value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	YES

Units	speed units
Value Range	--
Default Value	--

velocity_window

velocity_actual_value (606C h) 和 target_velocity (60FF h) 的差值被定义为实际速度误差窗口。如果实际速度误差窗口在 velocity_window_time (606E h) 参数设置的时间内一直比 velocity_window (606D h) 小, 那么 statusword 的 bit10 (target_reached) 将被置位, 表示设定速度到达。

Index	606D h
Name	velocity_window
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	20 R/10min

velocity_window_time

velocity_window_time 参数和 velocity_window 参数一起组成速度窗口比较器。

Index	606E h
Name	velocity_window_time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	ms
Value Range	--
Default Value	0

velocity_threshold

velocity_threshold (阈值速度) 指接近零速度的一个范围, 用以判断电机是否停止。。

Index	606F h
Name	velocity_threshold
Object Code	VAR
Data Type	UINT16
Access	RW

PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	10 R/10min

velocity_threshold_time

velocity_threshold_time（阈值速度时间）设置电机在低于阈值速度时的最短时间，单位：**ms**。当电机速度低于阈值速度的时间超过阈值速度时间时，状态字bit12（速度为零）将被置1。

Index	6070 h
Name	velocity_threshold_time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	ms
Value Range	--
Default Value	0

target_velocity

target_velocity 是给定目标速度。

Index	60FF h
Name	target_velocity
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

7.4 位置控制模式（PROFILE POSITION MODE）

7.4.1 位置模式的控制字

15 ~ 9	8	7	6	5	7 ~ 4	3 ~ 0
*	Halt	*	abs / rel	change set immediately	*	*

*: 参考前面章节

Name	Value	Description
New set-point	0	Does not assume target position
	1	Assume target position
Change set immediately	0	Finish the actual positioning and then start the next positioning
	1	Interrupt the actual positioning and start the next positioning
abs/rel	0	Target position is an absolute value
	1	Target position is a relative value
Halt	0	Execute positioning
	1	Stop axle with profile deceleration(if not supported with profile acceleration)

7.4.2 位置模式的状态字

15 ~ 14	13	12	11	10	9 ~ 0
*	Following error	Set_point acknowledge	*	Target reached	*

*: 参考前面章节

Name	Value	Description
Target reached	0	Halt=0:Target positon not reached Halt=1:Axle decelerates
	1	Halt=0:Target position reached Halt=1:Velocity of axle is 0
Set-point acknowledge	0	Trajectory generator has not assumed the positioning values(yet)
	1	Trajectory generator has assumed the positioning vaules
Following error	0	No following error
	1	Following error

7.4.3 位置控制相关参数

Index	Object	Name	Type	Attr.
607A h	VAR	target_position	INT32	RW
6081 h	VAR	profile_velocity	UINT32	RW
6082 h	VAR	end_velocity	UINT32	RW
6083 h	VAR	profile_acceleration	UINT32	RW
6084 h	VAR	profile_deceleration	UINT32	RW
6085 h	VAR	quick_stop_deceleration	UINT32	RW

6086 h	VAR	motion_profile_type	INT16	RW
--------	-----	---------------------	-------	----

target_position

target_position 是给定目标位置，该位置可以是相对值，也可以是绝对值，取决于 controlword 的 bit6。

Index	607A h
Name	target_position
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	position units
Value Range	--
Default Value	0

profile_velocity

profile_velocity 是指位置启动后，完成加速后最终到达的速度。

Index	6081 h
Name	profile_velocity
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--
Default Value	0

end_velocity

end_velocity 是到达给定位置（target_position）时的速度。通常为了在到达给定位置时停止驱动器，该参数设置为0；但在连续多点位置时，该值可以设置成非零值。

Index	6082 h
Name	end_velocity
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	--

Default Value	0
---------------	---

profile_acceleration

profile_acceleration 是到达给定位置期间的加速度。

Index	6083 h
Name	profile_acceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	100000 R/10min/s

profile_deceleration

profile_deceleration 是到达给定位置期间的加速度。

Index	6084 h
Name	profile_deceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	100000 R/10min/s

quick_stop_deceleration

quick_stop_deceleration 是急停（Quick Stop）的减速度。

Index	6085 h
Name	quick_stop_deceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	acceleration units
Value Range	--
Default Value	200000 R/10min/s

motion_profile_type

motion_profile_type 被用来选择何种速度曲线。目前位置下只支持梯形速度曲线。

Index	6086 h
Name	motion_profile_type
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0
Default Value	0

7.4.4 功能描述

给定目标位置有两种方法：

单步设定：

电机到达目标位置后，驱动器通知主机“目标位置到达”，然后获取新的目标位置并开始运动。在获取新的目标位置前，电机速度通常为零。

连续设定：

电机到达目标位置后，立即继续朝下一个事先设置好的目标位置运动。这样可达到无停顿的连续运动效果，在两个目标位置之间，电机可以无需减速到零。

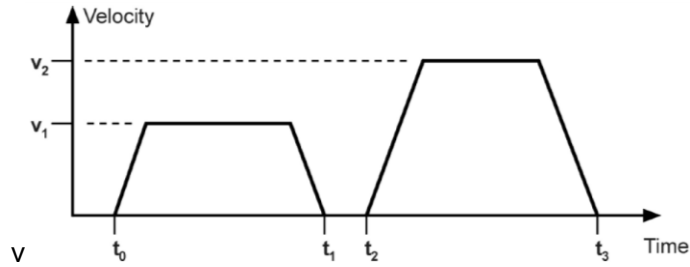
以上两种方法都可被控制字（controlword）的bit4、bit5 和状态字statusword 的bit12

（set_point_acknowledge）实时改变。通过握手机制，可中断正在执行中的位置控制，利用这几个字位重设目标位置并启动执行。

单步设定方法的步骤：

首先设置网络管理（NMT）状态为“操作（Operational）”，并且设置控制模式参数（6060 h）为1。

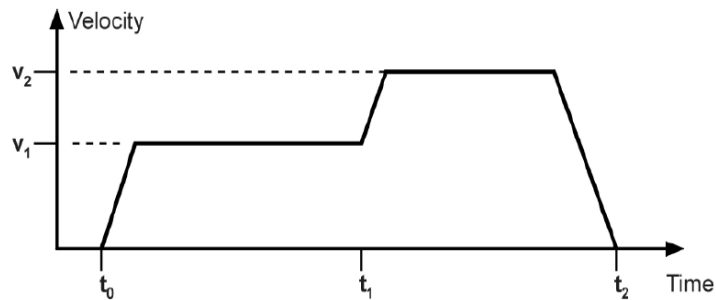
- 1、 根据实际需要设置给定目标位置（target_position：607A h）等参数；
- 2、 将控制字controlword 的bit4（new_set_point）设为“1”；bit5（change_set_immediately）设为“0”；bit6（绝对/相对）则由目标位置类型（绝对或相对）而定；
- 3、 在状态字 statusword 的 bit12（set_point_acknowledge）设置好驱动器应答，然后开始执行位置控制；
- 4、 到达目标位置后，驱动器通过状态字statusword 的bit10（target_reached）应答。然后依照程序继续运动或接受新的目标位置。



连续设定方法的步骤:

首先设置网络管理 (NMT) 状态为“操作 (Operational)”，并且设置控制模式参数 (6060 h) 为1。

- 1、 根据实际需要设置第1 个给定目标位置 (target_position : 607A h)、目标速度、加/减速等相关参数;
- 2、 将控制字controlword 的bit4 (new_set_point) 设为“1”; bit5 (change_set_immediately) 设为“0”; bit6 (绝对/相对) 则由目标位置类型 (绝对或相对) 而定;
- 3、 在状态字statusword 的bit12 (set_point_acknowledge) 设置好驱动器应答, 然后开始执行位置控制;
- 4、 设置第2 个给定目标位置 (target_position : 607A h)、目标速度、加/减速等相关参数;
- 5、 将控制字controlword 的bit4 (new_set_point) 设为“1”; bit5 (change_set_immediately) 设为“0”; bit6 (绝对/相对) 则由目标位置类型 (绝对或相对) 而定;
- 6、 到达第1 个目标位置后, 驱动器不停机继续走第2 个目标位置控制; 当到达第2 个目标位置后, 驱动器通过状态字statusword 的bit10 (target_reached) 应答。然后依照程序继续运动或接受新的目标位置。



附录 对象字典表

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单 位
							All	PP	PV	HM	
1000	--	VAR	device_type	UINT32	RO	NO	●				
1001	--	VAR	error_register	UINT8	RO	NO	●				
1003	--	VAR	pre_defined_error_field	UINT8	RW	NO	●				
1005	--	VAR	cob_id_sync	UINT32	RW	NO	●				
1006	--	VAR	communication_cycle_period	UINT32	RW	NO	●				
1007	--	VAR	synchronous_window_length	UINT32	RW	NO	●				
1008	--	VAR	manufacturer_device_name	STR	RO	NO	●				
1009	--	VAR	manufacturer_hardware_version	STR	RO	NO	●				
100A	--	VAR	manufacturer_software_version	STR	RO	NO	●				
1014	--	VAR	cob_id_emergency_message	UINT32	RW	NO	●				
1016	--	ARRAY	consumer_heartbeat_time	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		consumer_heartbeat_time1	UINT32	RW	NO	●				
1017		VAR	producer_heartbeat_time	UINT16	RW	NO	●				
1029	--	ARRAY	error_behaviour	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		communication_error	UINT8	RW	NO	●				

1200	--	RECORD	server_sdo_parameter	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		cob_id_client_server	UINT32	RO	NO	●				
	2		cob_id_server_client	UINT32	RO	NO	●				
1400	--	RECORD	receive_pdo_parameter_rpdo1	--	--	--	●				
	0		number_of_entries_rpdo1	UINT8	RO	NO	●				
	1		cob_id_used_by_pdo_rpdo1	UINT32	RO	NO	●				
	2		transmission_type_rpdo1	UINT8	RW	NO	●				
1401	--	RECORD	receive_pdo_parameter_rpdo2	--	--	--	●				
	0		number_of_entries_rpdo2	UINT8	RO	NO	●				
	1		cob_id_used_by_pdo_rpdo2	UINT32	RO	NO	●				
	2		transmission_type_rpdo2	UINT8	RW	NO	●				
1402	--	RECORD	receive_pdo_parameter_rpdo3	--	--	--	●				
	0		number_of_entries_rpdo3	UINT8	RO	NO	●				
	1		cob_id_used_by_pdo_rpdo3	UINT32	RO	NO	●				
	2		transmission_type_rpdo3	UINT8	RW	NO	●				
1403	--	RECORD	receive_pdo_parameter_rpdo4	--	--	--	●				
	0		number_of_entries_rpdo4	UINT8	RO	NO	●				
	1		cob_id_used_by_pdo_rpdo4	UINT32	RO	NO	●				
	2		transmission_type_rpdo4	UINT8	RW	NO	●				
1600	--	RECORD	receive_pdo_mapping_rpdo1	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				

	1		first_mapped_object_rpdo1	UINT32	RW	NO	●				
	2		second_mapped_object_rpdo1	UINT32	RW	NO	●				
	3		third_mapped_object_rpdo1	UINT32	RW	NO	●				
	4		fourth_mapped_object_rpdo1	UINT32	RW	NO	●				

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单 位
							All	PP	PV	HM	
1601	--	RECORD	receive_pdo_mapping_rpdo2	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		first_mapped_object_rpdo2	UINT32	RW	NO	●				
	2		second_mapped_object_rpdo2	UINT32	RW	NO	●				
	3		third_mapped_object_rpdo2	UINT32	RW	NO	●				
	4		fourth_mapped_object_rpdo2	UINT32	RW	NO	●				
1602	--	RECORD	receive_pdo_mapping_rpdo3	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		first_mapped_object_rpdo3	UINT32	RW	NO	●				
	2		second_mapped_object_rpdo3	UINT32	RW	NO	●				
	3		third_mapped_object_rpdo3	UINT32	RW	NO	●				
	4		fourth_mapped_object_rpdo3	UINT32	RW	NO	●				
1603	--	RECORD	receive_pdo_mapping_rpdo4	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				

	1		first_mapped_object_rpdo4	UINT32	RW	NO	●				
	2		second_mapped_object_rpdo4	UINT32	RW	NO	●				
	3		third_mapped_object_rpdo4	UINT32	RW	NO	●				
	4		fourth_mapped_object_rpdo4	UINT32	RW	NO	●				
1800	--	RECORD	transmit_pdo_parameter_tpdo1	--	--	--	●				
	0		number_of_entries_tpdo1	UINT32	RO	NO	●				
	1		cob_id_used_by_pdo_tpdo1	UINT32	RO	NO	●				
	2		transmission_type_tpdo1	UINT8	RW	NO	●				
	3		inhibit_time_tpdo1	UINT16	RW	NO	●				
	5		event_timer_tpdo1	UINT16	RW	NO	●				

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单位
							All	PP	PV	HM	
1801	--	RECORD	transmit_pdo_parameter_tpdo2	--	--	--	●				
	0		number_of_entries_tpdo2	UINT32	RO	NO	●				
	1		cob_id_used_by_pdo_tpdo2	UINT32	RO	NO	●				
	2		transmission_type_tpdo2	UINT8	RW	NO	●				
	3		inhibit_time_tpdo2	UINT16	RW	NO	●				
	5		event_timer_tpdo2	UINT16	RW	NO	●				
1802	--	RECORD	transmit_pdo_parameter_tpdo3	--	--	--	●				
	0		number_of_entries_tpdo3	UINT32	RO	NO	●				

	1		cob_id_used_by_pdo_tpdo3	UINT32	RO	NO	●				
	2		transmission_type_tpdo3	UINT8	RW	NO	●				
	3		inhibit_time_tpdo3	UINT16	RW	NO	●				
	5		event_timer_tpdo3	UINT16	RW	NO	●				
1803	--	RECORD	transmit_pdo_parameter_tpdo4	--	--	--	●				
	0		number_of_entries_tpdo4	UINT32	RO	NO	●				
	1		cob_id_used_by_pdo_tpdo4	UINT32	RO	NO	●				
	2		transmission_type_tpdo4	UINT8	RW	NO	●				
	3		inhibit_time_tpdo4	UINT16	RW	NO	●				
	5		event_timer_tpdo4	UINT16	RW	NO	●				
1A00	--	RECORD	transmit_pdo_mapping_tpdo1	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		first_mapped_object_tpdo1	UINT32	RW	NO	●				
	2		second_mapped_object_tpdo1	UINT32	RW	NO	●				
	3		third_mapped_object_tpdo1	UINT32	RW	NO	●				
	4		fourth_mapped_object_tpdo1	UINT32	RW	NO	●				

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单位
							All	PP	PV	HM	
1A01	--	RECORD	transmit_pdo_mapping_tpdo2	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				

	1		first_mapped_object_tpdo2	UINT32	RW	NO	●				
	2		second_mapped_object_tpdo2	UINT32	RW	NO	●				
	3		third_mapped_object_tpdo2	UINT32	RW	NO	●				
	4		fourth_mapped_object_tpdo2	UINT32	RW	NO	●				
1A02	--	RECORD	transmit_pdo_mapping_tpdo3	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		first_mapped_object_tpdo3	UINT32	RW	NO	●				
	2		second_mapped_object_tpdo3	UINT32	RW	NO	●				
	3		third_mapped_object_tpdo3	UINT32	RW	NO	●				
	4		fourth_mapped_object_tpdo3	UINT32	RW	NO	●				
1A03	--	RECORD	transmit_pdo_mapping_tpdo4	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		first_mapped_object_tpdo4	UINT32	RW	NO	●				
	2		second_mapped_object_tpdo4	UINT32	RW	NO	●				
	3		third_mapped_object_tpdo4	UINT32	RW	NO	●				
	4		fourth_mapped_object_tpdo4	UINT32	RW	NO	●				

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映 射	支持				单位
							All	PP	PV	HM	
2000	--	RECORD	mask_tpdo1	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				

	1		mask1_tpdo1	UINT32	RW	NO	●				
	2		mask2_tpdo1	UINT32	RW	NO	●				
2001	--	RECORD	mask_tpdo2	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		mask1_tpdo2	UINT32	RW	NO	●				
	2		mask2_tpdo2	UINT32	RW	NO	●				
2002	--	RECORD	mask_tpdo3	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		mask1_tpdo3	UINT32	RW	NO	●				
	2		mask2_tpdo3	UINT32	RW	NO	●				
2003	--	RECORD	mask_tpdo4	--	--	--	●				
	0		number_of_entries	UINT8	RO	NO	●				
	1		mask1_tpdo4	UINT32	RW	NO	●				
	2		mask2_tpdo4	UINT32	RW	NO	●				

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单位
							All	PP	PV	HM	
603F	--	VAR	error_code	UINT16	RW	YES	●				
6040	--	VAR	controlword	UINT16	RW	YES	●				

6041	--	VAR	statusword	UINT16	RO	YES	●				
605A	--	VAR	quick_stop_option_code	INT16	RW	NO	●				
605B	--	VAR	shutdown_option_code	INT16	RW	NO	●				
605C	--	VAR	disable_operation_option_code	INT16	RW	NO	●				
605D	--	VAR	stop_option_code	INT16	RW	NO	●				
605E	--	VAR	fault_reaction_option_code	UINT16	RW	NO	●				
6060	--	VAR	modes_of_operation	INT8	RW	YES	●				
6061	--	VAR	modes_of_operation_display	INT8	RO	YES	●				
6062	--	VAR	position_demand_value	INT32	RO	YES		●			position units
6063	--	VAR	position_actual_value*	INT32	RO	YES		●			inc
6064	--	VAR	position_actual_value	INT32	RO	YES		●			position units
6065	--	VAR	following_error_window	UINT32	RW	YES		●			position units
6066	--	VAR	following_error_time_out	UINT16	RW	YES		●			ms
6067	--	VAR	position_window	UINT32	RW	YES		●			position units
6068	--	VAR	position_window_time	UINT16	RW	YES		●			ms
6069	--	VAR	velocity_sensor_actual_value	UINT16	RW	YES			●		speed units
606B	--	VAR	velocity_demand_value	INT32	RO	YES			●		speed units
606C	--	VAR	velocity_actual_value	INT32	RO	YES			●		speed units
606D	--	VAR	velocity_window	UINT16	RW	YES			●		speed units
606E	--	VAR	velocity_window_time	UINT16	RW	YES			●		ms
606F	--	VAR	velocity_threshold	UINT16	RW	YES			●		speed units
6070	--	VAR	velocity_threshold_time	UINT16	RW	YES			●		ms

607A	--	VAR	target_position	INT32	RW	YES		●			position units
------	----	-----	-----------------	-------	----	-----	--	---	--	--	----------------

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单位
							All	PP	PV	HM	
607B	--	ARRAY	position_range_limit	--	--	--		●			
	0		number_of_entries	UINT8	RW	NO		●			
	1		min_position_range_limit	INT32	RW	NO		●			position units
	2		max_position_range_limit	INT32	RW	NO		●			position units
607C	--	VAR	home_offset	INT32	RW	YES		●		●	position units
6081	--	VAR	profile_velocity	UINT32	RW	YES		●			speed units
6082	--	VAR	end_velocity	UINT32	RW	YES		●			speed units
6083	--	VAR	profile_acceleration	UINT32	RW	YES		●	●		acceleration units
6084	--	VAR	profile_deceleration	UINT32	RW	YES		●	●		acceleration units
6085	--	VAR	quick_stop_deceleration	UINT32	RW	YES		●	●		acceleration units
6086	--	VAR	motion_profile_type	INT16	RO	YES		●	●		
6093	--	ARRAY	position_factor	--	--	--		●		●	
	0		number_of_entries	UINT32	RW	NO		●		●	
	1		numerator	UINT32	RW	NO		●		●	
	2		divisor	UINT32	RW	NO		●		●	
6094	--	ARRAY	velocity_encoder_factor	--	--	--	●				
	0		number_of_entries	UINT32	RW	NO	●				

	1		numerator	UINT32	RW	NO	●				
	2		divisor	UINT32	RW	NO	●				
6097	--	ARRAY	acceleration_factor	--	--	--	●				
	0		number_of_entries	UINT32	RW	NO	●				
	1		acceleration_factor	UINT32	RW	NO	●				
	2		divisor	UINT32	RW	NO	●				
6098	--	VAR	homing_method	INT8	RW	YES				●	

Index 索引	Subindex 子索引	Object 对象类型	Name 名称	Type 数据类型	Attr. 属性	PDO 映射	支持				单位
							All	PP	PV	HM	
6099	--	ARRAY	homing_speeds	--	--	--				●	speed units
	0		number_of_entries	UINT8	RW	YES				●	
	1		speed_during_search_for_switch	UINT32	RW	YES				●	speed units
	2		speed_during_search_for_zero	UINT32	RW	YES				●	speed units
609A	--	VAR	homing_acceleration	UINT32	RW	YES				●	acceleration units
60FA	--	VAR	control_effort	INT32	RO	YES		●			
60FC	--	VAR	position_demand_value*	INT32	RO	YES		●			inc
60FF	--	VAR	target_velocity	UINT32	RW	YES			●		speed units