

CANopen

用户手册

MOONS'
moving in better ways

上海安浦鸣志自动化设备有限公司

目录

1 前言	5
1.1 关于手册	5
1.2 参考文档	5
1.3 常用缩写	5
1.4 版本历史	6
2 CANopen协议	7
2.1 CANopen概述	7
2.2 CANopen 网络拓扑结构说明	7
3 驱动器设置	8
3.1 连接电源与电机	8
3.2 连接CANopen通讯接口	8
3.3 CANopen 波特率和地址	9
3.4 驱动器配置	9
4 CANopen通讯	10
4.1 CANopen信息	10
4.2 SDO 消息	11
4.3 PDO消息	12
4.4 网络管理 (NMT)	13
4.5 NMT 状态机	14
5 运动控制	15
5.1 控制电源驱动系统PDS	15
5.1.1 概述	15
5.1.2 有限自动状态机FSA	15
5.2 轮廓位置模式	17
5.2.1 模式说明	17
5.2.2 主要控制寄存器对象	17
5.2.3 功能描述	18
5.3 原点模式	22
5.3.1 模式说明	22
5.3.2 主要控制寄存器对象	22
5.3.3 功能描述	23
5.3.4 方式 1 和 2	23

5.3.5 方式 3 和 4.....	24
5.3.6 方式 5 和 6.....	24
5.3.7 方式 7 和 8.....	25
5.3.8 方式 9 和 10.....	25
5.3.9 方式 11 和 12.....	26
5.3.10 方式 13 和 14.....	26
5.3.11 方法 15 和 16.....	27
5.3.12 方法 17 和 18.....	27
5.3.13 方法 19 和 20.....	27
5.3.14 方法 21 和 22.....	28
5.3.15 方法 23 和 24.....	28
5.3.16 方法 25 和 26.....	29
5.3.17 方法 27 和 28.....	29
5.3.18 方法 29 和 30.....	30
5.3.19 方法 33 和 34.....	30
5.3.20 方法 35.....	30
5.3.21 方法 36.....	30
5.3.22 方法 37.....	30
5.4 轮廓速度模式.....	31
5.4.1 模式说明.....	31
5.4.2 主要控制寄存器对象.....	31
5.4.3 功能描述.....	32
5.5 扭矩模式 (步进伺服支持).....	33
5.5.1 模式说明.....	33
5.5.2 主要控制寄存器对象.....	33
5.5.3 功能描述.....	34
5.6 Q 程序模式.....	35
5.6.1 模式说明.....	35
5.6.2 常规执行Q程序.....	35
5.6.3 同步方式执行Q程序.....	36
5.7 全局控制字与状态字说明.....	37
6 应用功能.....	38
6.1 通讯看门狗.....	38
7 对象字典.....	39

7.1 对象描述39

7.2 电子数据表格40

 7.2.1 通讯配置文件1000h(CiA301).....40

 7.2.2 运动控制配置文件6000h(CiA402).....55

 7.2.3 通用配置文件65

 7.2.4 步进伺服通用配置文件.....72

 7.2.5 步进通用配置文件.....82

8 附录A – SDO消息控制举例.....88

9 联系 MOONS'91

1 前言

1.1 关于手册

本手册描述了鸣志驱动器产品在CANopen通讯当中，执行符合CiA301和CiA402协议标准的使用情况。用户在充分了解CiA通讯协议和本手册内容的基础上，可以开发出基于CANopen通讯的分布式运动控制系统。本手册重心是突出强调描述鸣志驱动器在CANopen通讯上制造商定义部分内容。

1.2 参考文档

MOONS' CANopen Drive Hardware Manual

CiA 301

CiA 303

CiA 402

Bosch CAN Physical Layer Specifications 2.0B

MOONS' Host Command Reference

1.3 常用缩写

CAN	控制局域网
CiA	CAN自动化组织 (标准机构)
COB-ID	通讯对象标识符
EDS	电子数据表
NMT	网络管理
OD	对象字典
PDS	电源驱动系统
PDO	过程数据对象
RPDO	接收过程数据对象
SDO	服务数据对象
TPDO	发送过程数据对象
COS	TPDO检测状态改变
FSA	有限自动状态机

1.4 版本历史

日期	版本	版本变化
9/07/2020	1.0	首版

驱动器型号

本手册主要针对步进伺服与步进驱动器CANopen的通讯说明介绍:

开环步进驱动器:

型号	固件版本
STF03-C STF05-C STF06-C STF10-C	310A及以后
STF03-C-mini	1.02A及以后

步进伺服驱动器:

型号	固件版本
SSDC03-C SSDC05-C SSDC06-C SSDC10-C	310B及以后

2 CANopen协议

2.1 CANopen概述

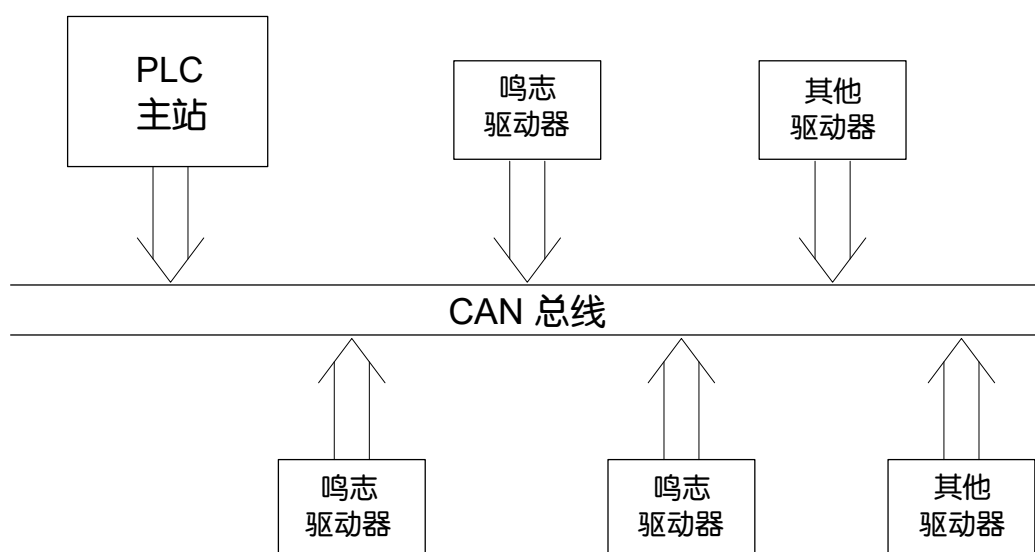
引自CiA组织描述:

CANopen是一种基于CAN总线的通讯系统。它包含了高层协议和配置文件规范。CANopen已开发为具有高度灵活配置功能的标准化嵌入式网络。它最初是为面向运动控制的机器控制系统(例如: 处理器系统)设计的。

今天它已用于各种应用领域, 例如医疗设备, 越野车, 海事电子设备, 铁路应用或楼宇自动化。

2.2 CANopen 网络拓扑结构说明

鸣志CANopen驱动器可以与其他设备集成到同一CAN总线网络系统中, 如下所示。



3 驱动器设置

设置鸣志CANopen步进和步进伺服驱动器有如下4个步骤：

- 连接电源与电机
- 连接CANopen通讯接口
- 设置波特率与节点地址
- 配置驱动器参数

3.1 连接电源与电机

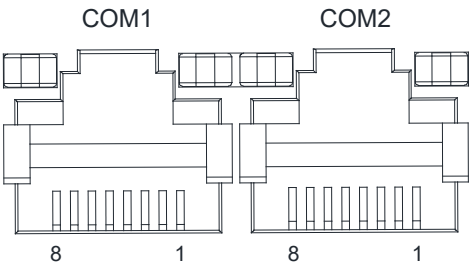
这部分内容请参考鸣志步进或步进伺服硬件手册。相关手册在鸣志官网可以找到。

网址：<https://www.moons.com.cn>

3.2 连接CANopen通讯接口

请参考鸣志CANopen驱动器硬件手册关于通讯接口部分，这里我们以STF驱动器为例做说明。

鸣志STF系列驱动器采用符合CiA303标准的RJ45连接器作为CANopen通讯接口，如下图所示。总线连接采用菊花链方式，在CANopen总线上每个接口末端需再加上120 Ω 电阻。



PIN	COM1	COM2	LINE
1	CAN_H	CAN_H	橙/白
2	CAN_L	CAN_L	橙色
3	GND	GND	绿/白
4	RS-232_TX	NC	蓝
5	RS-232_RX	NC	蓝/白
6	NC	NC	绿
7	GND	GND	棕/白
8	GND	GND	棕

3.3 CANopen 波特率和地址

STF系列驱动器设置波特率和地址共分三部分，波特率一部分与节点ID两部分。波特率是通过一个10位旋钮开关来配置的。而节点地址(Node-ID)的低4位采用一个16位旋钮开关选择，高3位则需要通过Stepper suite软件来配置([点击下载软件](#))。

请参考驱动器硬件手册关于节点地址(Node-ID)旋钮设置。节点地址的有效范围是01h-7Fh。根据CiA 301标准保留节点地址00h。

注意：仅在重新启动电源或发送网络重置命令之后，节点地址和波特率才会生效。在上电的情况下更改旋钮开关位置，电源没有重启或者网络没有重置的情况下，波特率和地址不会立刻生效。

旋钮开关	波特率
0	1000Kbps
1	800Kbps
2	500Kbps
3	250Kbps
4	125Kbps
5	50Kbps
6	20Kbps
7	12.5Kps
8	预留
9	预留

3.4 驱动器配置

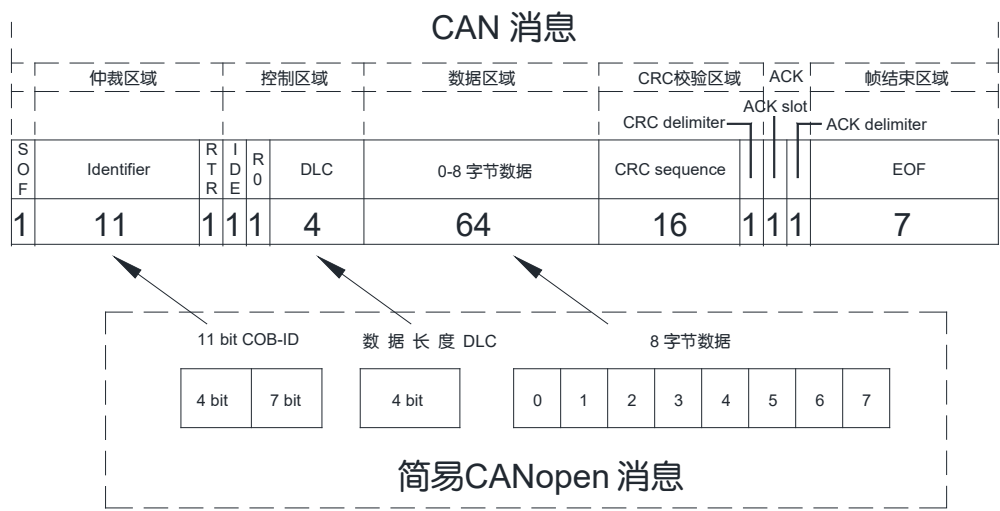
一旦将CAN连接器连接到驱动器，并且设置了节点地址和波特率，接下来就可以配置驱动器参数。可以在鸣志网站上找到Stepper Suite软件进行配置。在任何情况下，都需要使用随附的RS-232串行电缆将驱动器连接到电脑上。有关详细连接信息，请参阅相应的驱动器硬件手册。

注意：首次给驱动器上电时，驱动器将通过RS-232端口自动发送上电数据包。如果存在鸣志应用程序响应，它将通过RS-232将响应发送回驱动器，驱动器将CAN节点保持在初始化状态，直到关闭应用程序为止。如果未检测到响应，则驱动器将继续正常的CANopen启动过程：驱动器上电进入“初始化”状态，发送boot-up启动数据包，进入“Pre-operation”状态，并开始发送带有Pre-operation状态的心跳信号。

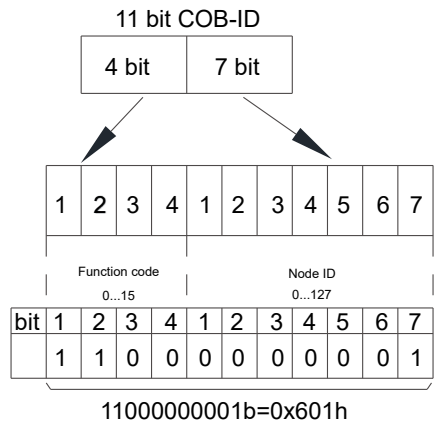
4 CANopen通讯

4.1 CANopen信息

基于CAN消息的简易CANopen通讯信息结构



COB-ID 数据结构解析:



COB	功能码	Resulting COB-ID
NMT	0000b	0 (000h)
SYNC	0001b	128 (080h)
TIME	0010b	256 (100h)
EMCY	0001b	129 (081h) – 255 (0FFh)
PDO1(TX)	0011b	385 (181h) – 511 (1FFh)
PDO1(RX)	0100b	513 (201h) – 639 (27Fh)
PDO2(TX)	0101b	641 (281h) – 767 (2FFh)
PDO2(RX)	0110b	769 (301h) – 895 (37Fh)
PDO3(TX)	0111b	897 (381h) – 1023 (3FFh)
PDO3(RX)	1000b	1025 (401h) – 1151 (47Fh)
PDO4(TX)	1001b	1153 (481h) – 1279 (4FFh)
PDO4(RX)	1010b	1281 (501h) – 1407 (57Fh)
SDO(TX)	1011b	1409 (581h) – 1535 (5FFh)
SDO(RX)	1100b	1537 (601h) – 1663 (67Fh)
NMT error control	1110b	1793 (701h) – 1919 (77Fh)

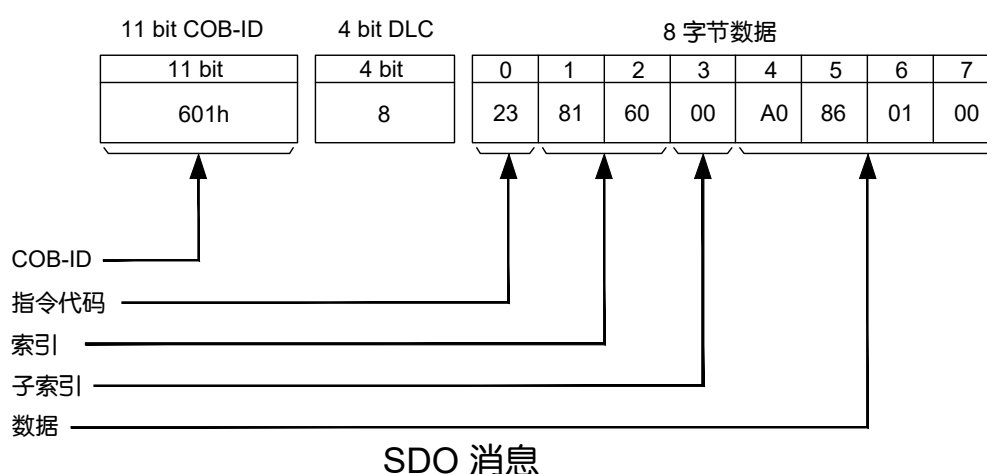
其中NMT，SYNC，TIME 是广播消息，其他是点对点消息。

4.2 SDO 消息

SDO提供对CANopen设备对象字典中对象条目直接访问的方法。由于这些对象条目可能包含任意大小和数据类型的数据。所以SDO可以用于将多个数据集(每个数据集包含一个任意大的数据块)从客户端传输到服务器，反之亦然，客户端应通过多路复用器(对象字典的索引和子索引)控制应传输的数据集。数据集的内容在对象字典中定义。

PDO消息											
	COB-ID		DLC		8 字节数据						
COB	4Bit功能码	7Bit节点地址	4bit	0	1	2	3	4	5	6	7
SDO(TX)	1011b(580h)	0...127(0...7F)	0...8								
SDO(RX)	1100b(600h)	0...127(0...7F)	0...8								

SDO消息解析举例：



指令代码(Command Code)说明:

指令代码	描述	关系
2Fh	写1个字节数据到对象	主站>从站
2Bh	写2个字节数据到对象	主站>从站
27h	写3个字节数据到对象	主站>从站
23h	写4个字节数据到对象	主站>从站
60h	写数据反馈	从站>主站
40h	从对象读取数据	主站>从站
4Fh	读取到1个字节数据	从站>主站
4Bh	读取到2个字节数据	从站>主站
47h	读取到3个字节数据	从站>主站
43h	读取到4个字节数据	从站>主站

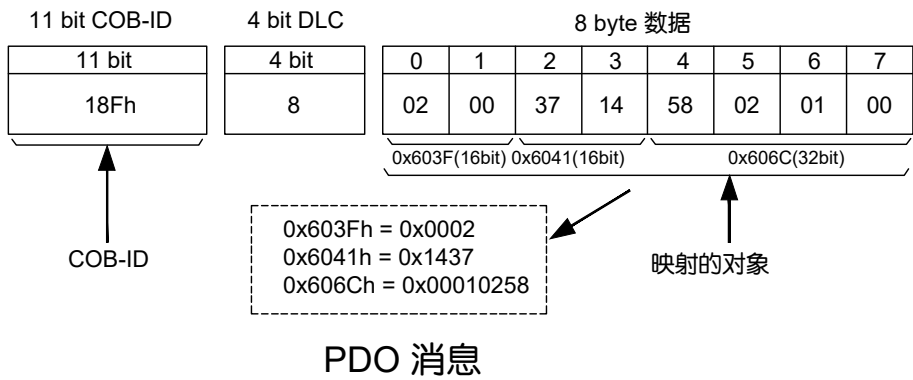
上图举例说明：主站往0x6081h(轮廓速度，没有子索引)对象里面写数据0x000186A0(100000)，数据大小为4个字节。

4.3 PDO消息

实时性数据一般通过“过程数据对象(PDO)”的方式输出。PDO的数据传输执行方式没有数据协议格式。PDO映射位于对象字典中的对象，并提供通讯接口给应用对象。在PDO数据包中，数据格式和映射的应用对象由对象字典中默认的数据结构决定。如果支持可变PDO映射，在通过SDO消息的方式配置对象字典中对象的过程中，将PDO数量和映射的应用对象写入CANopen设备。

PDO消息											
	COB-ID		DLC		8字节数据						
COB	4Bit功能码	7 bit节点地址	4bit	0	1	2	3	4	5	6	7
PDO1(TX)	0011b(180h)	0...127(0...7F)	0...8	映射的对象							
PDO1(RX)	0100b(200h)	0...127(0...7F)	0...8								
PDO2(TX)	0101b(280h)	0...127(0...7F)	0...8								
PDO2(RX)	0110b(300h)	0...127(0...7F)	0...8								
PDO3(TX)	0111b(380h)	0...127(0...7F)	0...8								
PDO3(RX)	1000b(400h)	0...127(0...7F)	0...8								
PDO4(TX)	1001b(480h)	0...127(0...7F)	0...8								
PDO4(RX)	1010b(500h)	0...127(0...7F)	0...8								

PDO消息解析举例：



上图举例说明：CANopen从站地址为15(0x0Fh)的驱动器，通过PDO1(TX)反馈从站的603Fh，6041h，606Ch的数据。

4.4 网络管理 (NMT)

网络管理(NMT)是面向CANopen设备的，并遵循主从结构。NMT对象用于执行NMT服务。通过NMT服务，可以初始化，启动，监视，重置或停止CANopen设备。所有CANopen设备均视为NMT从站。NMT从站通过其节点ID(在[1 ... 127]范围内的值)在网络中作为唯一标识。NMT要求网络中只有一台CANopen设备履行NMT主站的功能。

11 bit COB-ID		4 bit DLC	8 字节数据		
4 bit	7 bit	4 bit	0	1	2.....7
0	0	2	NMT指令	节点地址	0

NMT 消息

NMT 指令解析：

NMT 指令	描述
1(1h)	启动节点
2(2h)	停止节点
128(80h)	进入预运行模式
129(81h)	重启节点
130(82h)	重启通讯

当NMT指令被执行之后，NMT的状态将发生相应变化。状态由心跳报文产生。

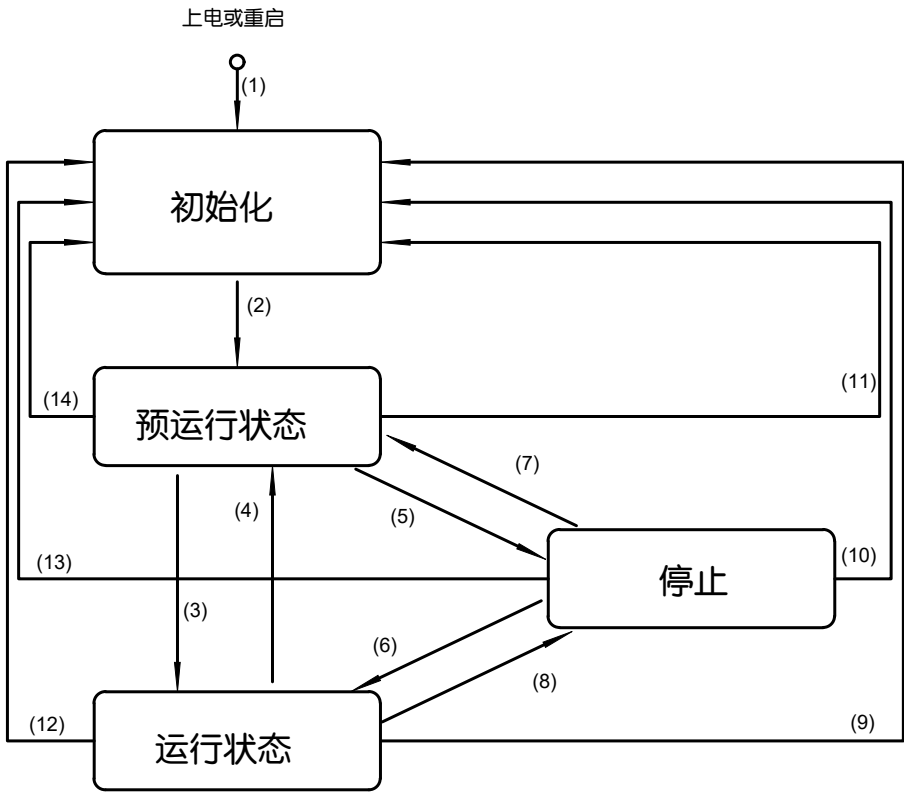
11 bit COB-ID		4 bit DLC	8 字节数据	
4 bit	7 bit	4 bit	0	1.....7
0x700	节点地址	1	NMT状态	0

心跳消息

NMT状态	描述说明
0	启动状态
4	停止状态
5	运行状态
127(7Fh)	预运行状态

4.5 NMT 状态机

当CANopen设备完成CANopen初始化之后，直接进入预运行状态。在此NMT状态期间，通过SDO进行CANopen设备参数设置和CAN-ID分配(例如，使用配置工具)。然后将CANopen设备直接切换到运行状态。



(1)	上电之后自动进入NMT初始化
(2)	NMT初始化完成，自动进入预运行状态
(3)	表示NMT服务启动远程节点或通过本地控制
(4)(7)	表示NMT服务进入预运行状态
(5)(8)	表示NMT服务停止远程节点
(6)	表示NMT服务启动远程节点
(9)(10)(11)	表示NMT服务重启节点
(12)(13)(14)	表示NMT服务重启通讯

	预运行	运行	停止
PDO		√	
SDO	√	√	
SYNC	√	√	
TIME	√	√	
EMCY	√	√	
节点控制，错误控制	√	√	√

5 运动控制

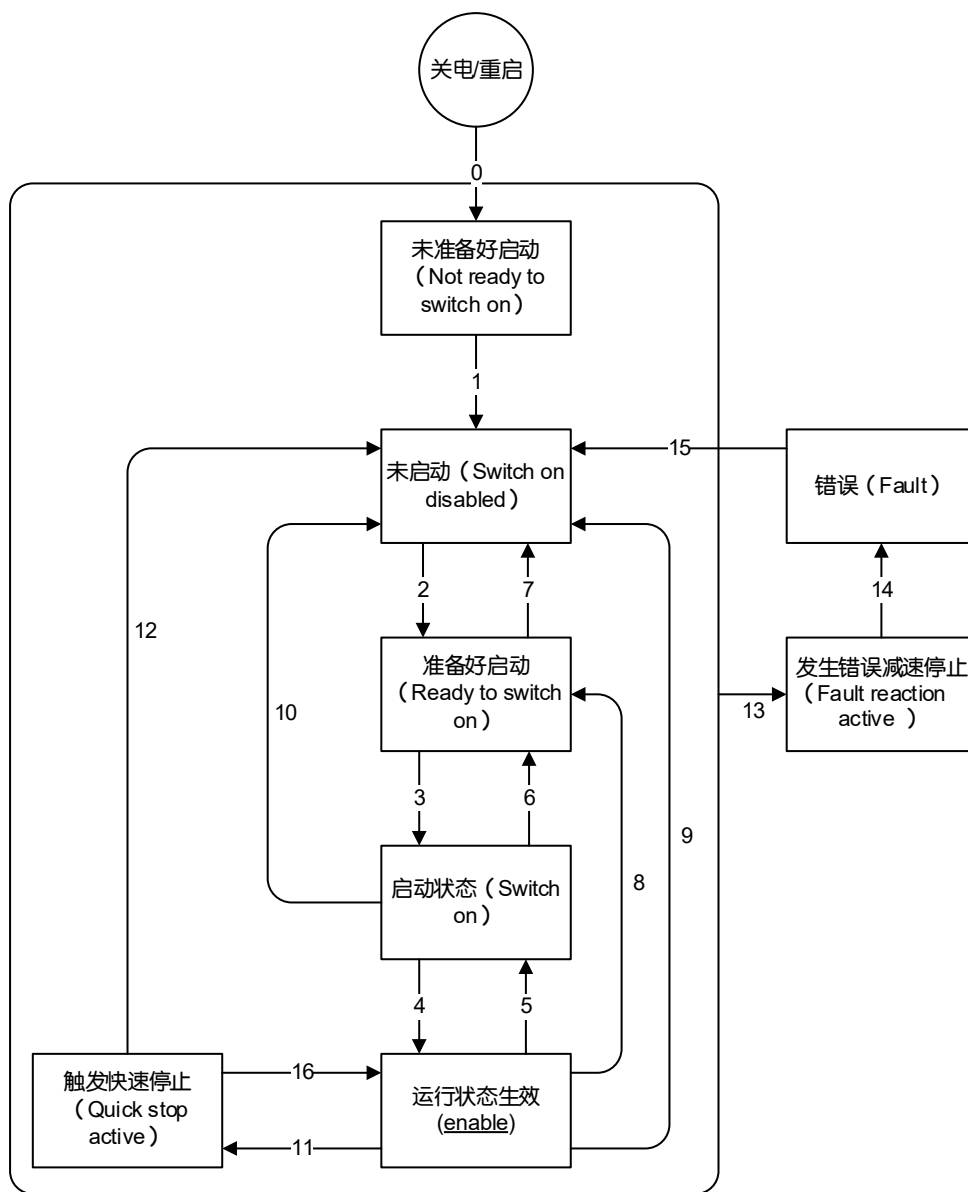
5.1 控制电源驱动系统PDS

5.1.1 概述

PDS, FSA概念是一个抽象的, 类似于控制系统中的黑匣子功能一样来说明PDS系统。他定义了PDS系统的应用程序行为。由于即使在通信网络无法正常工作时也需要PDS系统提供本地控制的要求, 通信系统映射规范中定义了FSA通讯, 且PDS FSA是宽松耦合状态形式。

5.1.2 有限自动状态机FSA

该图解释了在PDS, FSA系统中, 用户发送相关指令和驱动器内部发生错误的情况下遵守的电源控制规范。



电源驱动系统有限自动控制状态机

过渡事件和实际产生的动作:

过程	事件	动作
0	电源启动之后或者重启之后自动过渡	驱动设备自检或者执行自行初始化
1	自动过渡	通讯开始生效激活
2	控制器发送Shut down指令或者本地信号	无
3	从控制器接收到Switch on指令或者本地信号	High-level power 将被打开
4	从控制器接收到激活Operation模式指令或者本地信号	驱动器功能被激活, 所有内部Set points被清除
5	从控制器接收到关闭Operation模式指令/或者本地信号	驱动器功能被关闭
6	从控制器接收到Shut down指令或者本地信号	High-level power被关闭
7	从控制器接收到Quick stop或者Disable voltage指令	无
8	从控制器接收到Shut down指令/或者本地指令	驱动器功能将会被关闭, 并且High-level power也会被关闭
9	从控制器接收到Disable voltage指令或者本地信号	驱动器功能将会被关闭, 并且High-level power也会被关闭
10	从控制器接收到Quick stop或者Disable voltage指令或者本地信号	High-level power被关闭
11	从控制器接受到Quick stop 指令或者本地信号	Quick stop功能被启动
12	当Quick stop指令在急停模式1, 2, 3或者4的情况下完成动作或者从控制器接收到Disable voltage指令将会自动过渡(依据Quick stop option code选择的方式)	驱动器功能和High-level power被关闭
13	Fault信号被触发	错误发送反馈功能机制将被执行
14	自动过渡	驱动器功能将会被关闭, 并且High-level power 也会被关闭
15	从控制器接收到Fault reset指令或者本地信号	如果驱动设备上当前不存在故障, 则将故障状态复位。退出故障状态后, 控制设备应清除控制字中的故障复位bit位。
16	如果Quick stop otion code是5, 6, 7或者8, 控制器可以发送激活Operation指令	驱动器功能将被打开

对象0x6040h的指令代码.

指令	操作字的Bit位(6040h)					过渡
	bit7	bit3	bit2	bit1	bit0	
Shut down	0	x	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Switch on + Enable operation	0	1	1	1	1	3+4
Quick stop	0	x	x	0	x	7, 9, 10, 12
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	↑	x	x	x	x	15

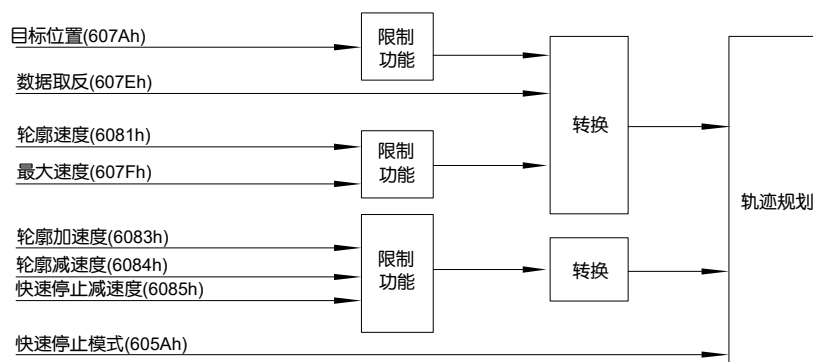
在PDS FSA系统中的状态位。

状态字(6041h)	PDS FSA 状态
xxxx xxxx x0xx 0000	未准备好进入Switch on状态(Not ready to switch on)
xxxx xxxx x1xx 0000	Switch on未激活(Switch on disabled)
xxxx xxxx x01x 0001	已准备好进入Switch on状态(Ready to switch on)
xxxx xxxx x01x 0011	Switch on已激活
xxxx xxxx x01x 0111	Operational模式已激活(Operation enabled)
xxxx xxxx x00x 0111	快速停止被触发(Quick stop active)
xxxx xxxx x0xx 1111	错误被触发(Fault reaction active)
xxxx xxxx x0xx 1000	错误(Fault)

5.2 轮廓位置模式

5.2.1 模式说明

轮廓位置模式是一个点到点的运动模式。由速度，加速度，减速度，目标位置参数组成。当这些参数设置完成之后，驱动器将指令缓存到驱动器并执行动作。当使用单个点到点运动方式且当前正在执行点到点动作时，新的点到点运动动作可以被缓存到驱动器。



轮廓位置模式(PP mode)

5.2.2 主要控制寄存器对象

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6060	Modes of operation	INT8	WO	YES
0x6061	Modes of operation display	INT8	RO	YES
0x607A	Target position	INT32	RW	YES
0x6081	Profile velocity	UINT32	RW	YES
0x6083	Profile acceleration	UINT32	RW	YES
0x6084	Profile deceleration	UINT32	RW	YES
0x6085	Quick stop deceleration	UINT32	RW	YES
0x605A	Quick stop code	INT16	RW	NO

轮廓位置模式下的操作字 (6040h)

15	10	9	8	7	6	5	4	3	0
***	Change on set point	Halt	***	Abs/Rel	Change set immediately	New set point	***		

***: 详情参考对象字描述。

Bit	Name	Value	Description
4	New set point	0	该位由0跳变为1将触发新的位置指令
		1	
5	Change set point immediately	0	在执行新的位置指令之前完成当前位置动作
		1	立刻更新并执行新的位置指令动作
6	Abs/Rel	0	目标位置为绝对位置
		1	目标位置为相对位置
8	Halt	0	位置指令被执行或者继续
		1	停止
9	Change of set point	0	在新的位置动作缓存之前驱动器完成之前的动作，且电机被复位
		1	电机将按照当前设定的速度运动直到到达设定位置之后再直接过渡到新的速度继续下一个位置动作

轮廓位置模式下的状态字 (6041h)。

15	14	13	12	11	10	9	0
****	Following error	Set point acknowledge	****	Target reached	****		

****: 详情参考对象字描述

Bit	Name	Value	Description
10	Target reached	0	Halt (bit 8 in controlword) = 0: 目标位置未到达
			Halt (bit 8 in controlword) = 1: 电机减速
		1	Halt (bit 8 in controlword) = 0: 目标位置到达
			Halt (bit 8 in controlword) = 1: 电机速度为0
12	Set point ACK	0	当前位置已被缓存, 等待新的位置缓存
		1	当前位置正在缓存, 新的位置将被接受
13	Following error	0	没有跟随错误
		1	有跟随错误

5.2.3 功能描述

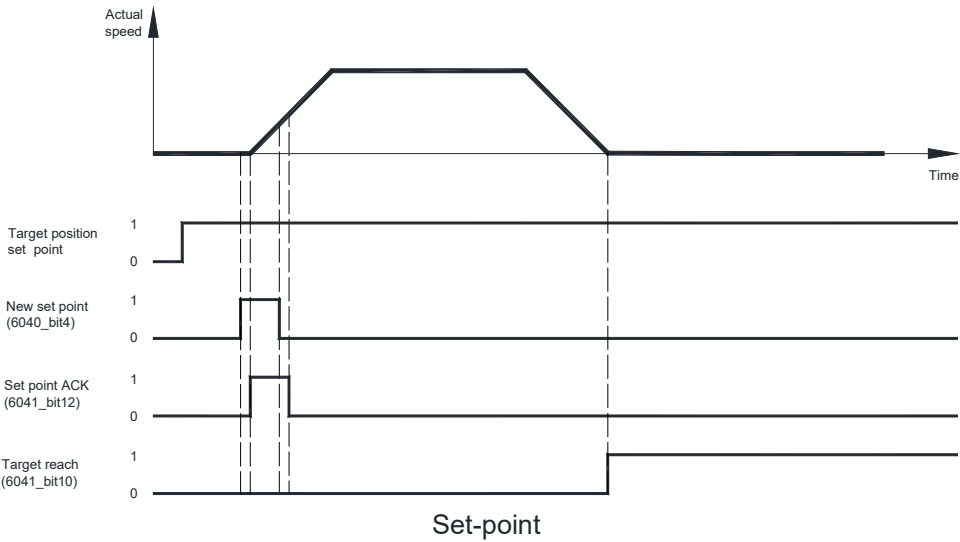
概述

通过控制字(0x6040)中New set point与Change set point immediately位以及状态字中的Set-point ack位, 他们之间的时序控制和状态触发来设置位置模式动作。

如果控制字中的Change set immediately位设置为1, Single set-point模式将会在驱动器上生效, 如果控制字中的Change set immediately位设置为0, Set of set-point 将会在驱动器上生效。

Set point

当Set-point模式被应用到驱动器之后, 控制设备表示通过触发控制字中New set-point位上升沿使Set point模式生效, 状态字中Set point acknowledge位变为1, 然后驱动设备通过使Set-point acknowledge变为0表示确认, 同时接收新的New set point信号。



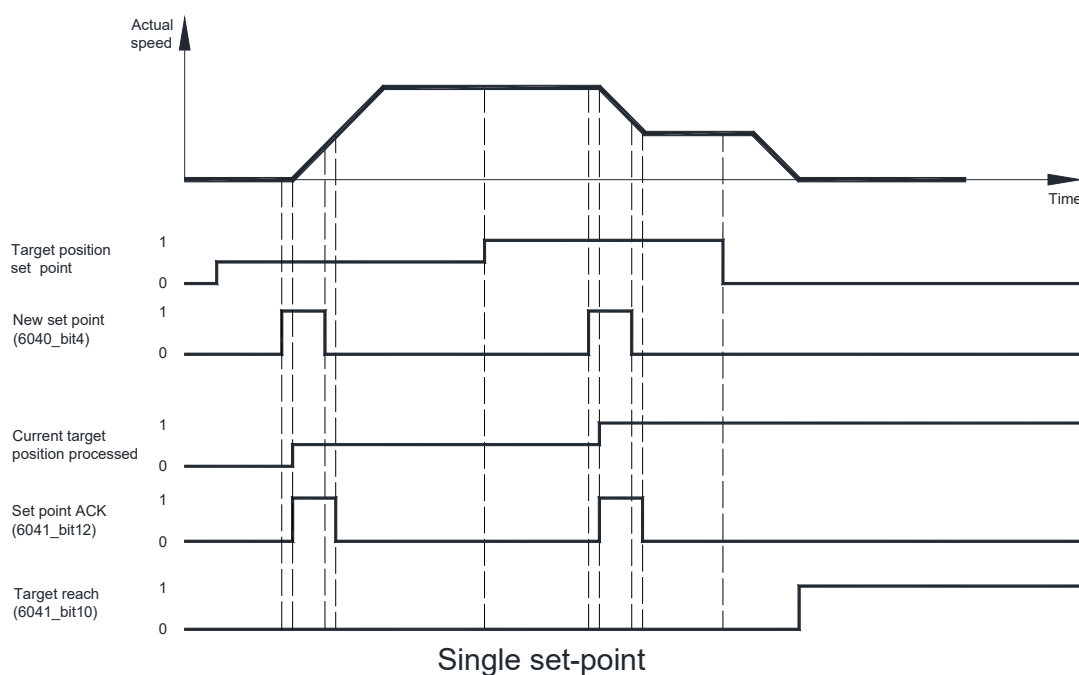
控制对象数据:

Event		Set parameter
使能电机	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + Enable operation	0x6040=15(0Fh)
设置运动模式	Profile position mode	0x6060=1(01h)
设置电机参数	目标位置Distance	0x607A=100000(0186A0h)
	目标速度Velocity	0x6081=20000(4E20h)
	加速度Acceleration	0x6083=50000(C350h)
	减速度Deceleration	0x6084=50000(C350h)
采用绝对运动方式	触发New set point位	0x6040=31(1Fh)
	清除New set point位	0x6040=15(0Fh)
采用相对运动方式	触发New set point位	0x6040=95(5Fh)
	清除New set point位	0x6040=79(4Fh)

如果一个Set-point 动作正在被缓存中执行，且一个新的已被确认，有两种处理方式可以支持：Single set-point(change set point immediately=1)和Set of set point(change set point immediately=0)。

Single set-point

当一个Set-point正在被缓存执行，一个新的Set-point动作通过控制字中的New set-point(bit4)被触发，新的Set-point动作将会立刻被缓存执行。

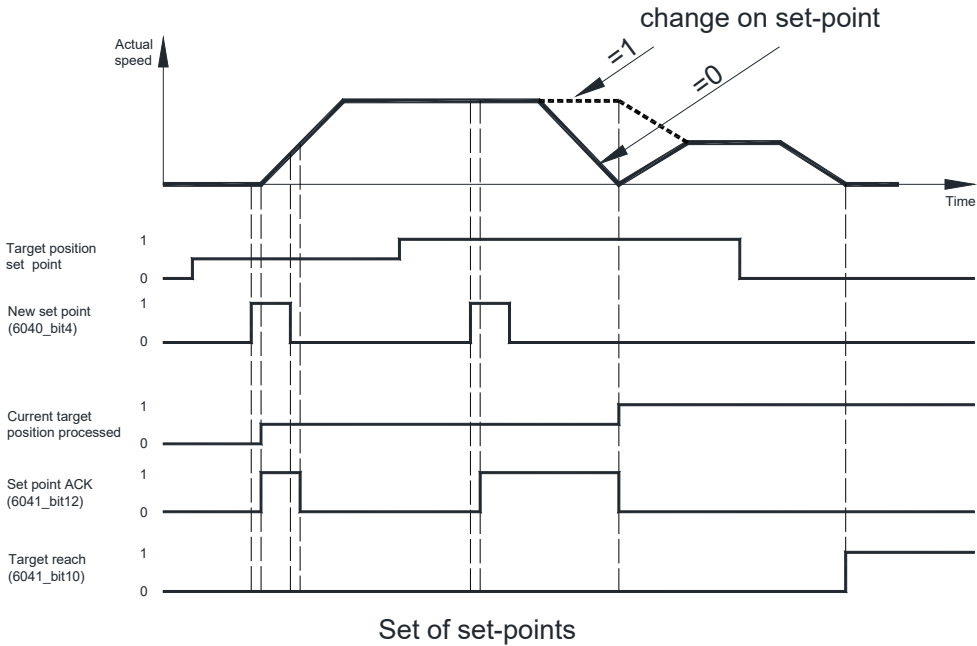


控制对象数据:

Event		Set parameter
使能电机	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + Enable operation	0x6040=15(0Fh)
设置运动模式	Profile position mode	0x6060=1(01h)
设置参数	加速度Acceleration	0x6083=50000(C350h)
	减速度Deceleration	0x6084=50000(C350h)
single set point	第一段速度Velocity	0x6081=30000(7530h)
	第一段目标Distance	0x607A=200000(030D40h)
	触发New set point	0x6040=639(27Fh)
	清除New set point	0x6040=623(26Fh)
	第二段速度Velocity	0x6081=20000(4E20h)
	第二段位置Distance	0x607A=100000(0186A0h)
	触发New set point	0x6040=639(27Fh)
	清除New set point	0x6040=623(26Fh)

Set of set-points

当一个Set-point正在被缓存执行，一个新的set-point动作通过控制字中的new set-point(bit4)被触发，新的Set-point将会在当前位置动作执行到达完成之后再被执行。



控制对象数据:

Event	Set parameter	
使能电机	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + Enable operation	0x6040=15(0Fh)
设置运行模式	Profile position mode	0x6060=1(01h)
设置参数	加速度Acceleration	0x6083=50000(C350h)
	减速度Deceleration	0x6084=50000(C350h))
Set of set-points with change on set-point=0	第一段目标速度Velocity	0x6081=30000(7530h)
	第一段目标位置Distance	0x607A=400000(061A80h)
	触发New set point	0x6040=95(5Fh)
	清除New set point	0x6040=79(4Fh)
	第二段目标速度Velocity	0x6081=20000(4E20h)
	第二段目标位置Distance	0x607A=300000(7530h)
	触发New set point	0x6040=95(5Fh)
	清除New set point	0x6040=79(4Fh)
Set of set-points with change on set-point=1	第一段目标速度Velocity	0x6081=30000(7530h)
	第一段目标位置Distance	0x607A=400000(061A80h)
	触发New set point	0x6040=607(25Fh)
	清除New set point	0x6040=591(24Fh)
	第二段目标速度Velocity	0x6081=20000(4E20h)
	第二段目标位置Distance	0x607A=300000(0493E0h)
	触发New set point	0x6040=607(25Fh)
	清除New set point	0x6040=591(24Fh)

注意：鸣志CANopen驱动器支持设置两段缓存位置动作，当状态字(0x6041)的第12位=1的时候说明缓存已满，新的位置动作将会被忽略。

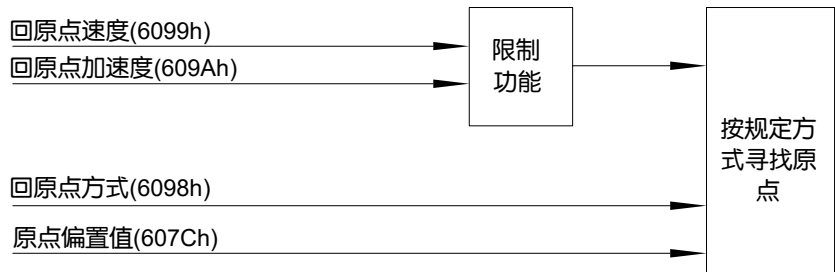
通过Halt指令停止电机(0x6040_bit8)

如果一个驱动器正在缓存执行一个位置动作，Halt指令可以停止电机并且保持在当前位置模式。当复位Halt指令之后可以继续执行当前位置动作。

5.3 原点模式

5.3.1 模式说明

本节描述了驱动器寻找起始位置(也称为基准点, 参考点或零点)的方法。通过限位有多种回原方法可以实现。通过行程末端的行程开关或行程中间的归零开关(零点开关), 以及大多数方法还使用增量编码器的index(Z相信号)脉冲信号来实现功能。



回原点模式(HM mode)

5.3.2 主要控制寄存器对象

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6060	Mode of operation	INT8	WO	YES
0x6098	Home method	INT8	RW	NO
0x6099	Homing speed	ARRAY	-	YES
0x609A	Homing acceleration	INT32	RW	YES
0x2001	Home switch	INT8	RW	YES
0x607C	Homing offset	INT32	RW	YES

原点模式下的控制字:

15	9	8	7	6	5	4	3	0
***		Halt	***	Reserved(0)		Homing operation start		***

***: 详细请参考对象描述

Bit	名称	值	描述
4	Homing operation start	0	不启动回原点动作
		1	启动或者继续回原点动作
8	Halt	0	激活第4位功能
		1	通过Halt option code停止

原点模式下的状态字:

15	14	13	12	11	10	9	0
***		Homing error	Homing attained	***	Target reached		***

***: 详情请参考对象描述

Bit13	Bit12	Bit10	描述
0	0	0	回原点动作正在执行
0	0	1	回原点被打断或者未执行
0	1	0	回原点到达, 但目标位置未到达
0	1	1	回原点动作成功完成
1	0	0	回原点出现错误, 速度不为0
1	0	1	回原点出现错误, 速度为0
1	1	X	预留

5.3.3 功能描述

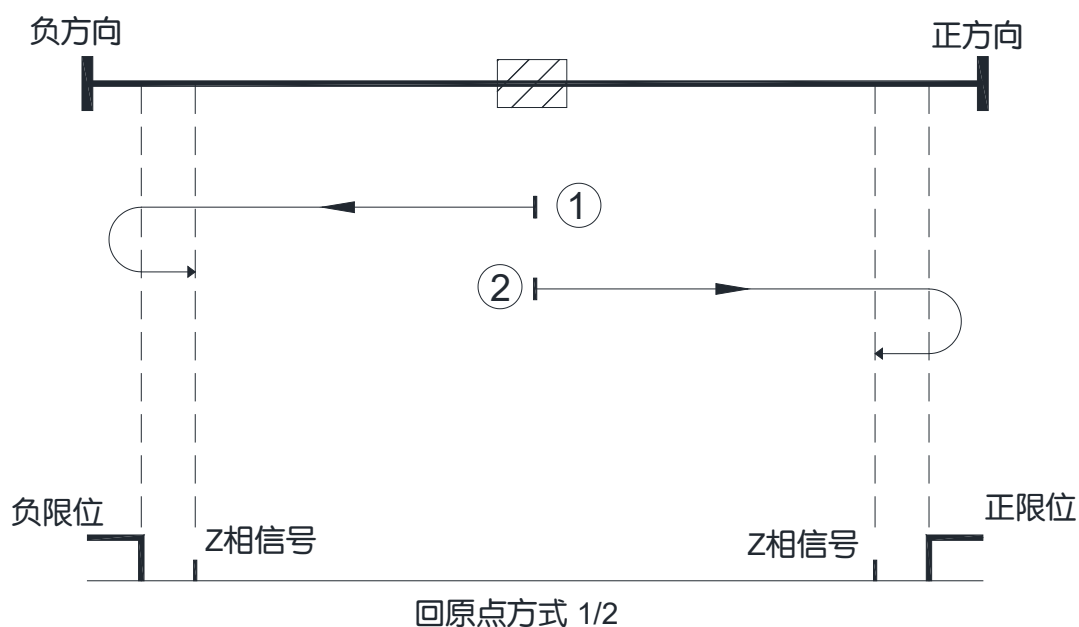
回原模式一般工作在限位开关和原点信号上(0x60FDh)。

控制对象数据:

Event		Set parameter
使能电机	Shut down	0x6040=06(06h)
	Switch on	0x6040=07(07h)
	Switch on+ enable operation	0x6040=15(0Fh)
设置运动模式	原点模式	0x6060=06(06h)
设置回原点方法	回原点方式13	0x6098=13(0Dh)
设置参数	回原点加速度	0x609A=200000(030D40h)
	选择限位开关速度	0x6099_sub1=20000(4E20h)
	选择index信号速度	0x6099_sub2=2000(07D0h)
	回原点offset值	0x607C=100000(0186A0h)
	原点输入信号选择X5	0x2001=5(05h)
Homing mode	启动回原点	0x6040=31(1Fh)
	停止回原点	0x6040=287(011Fh)

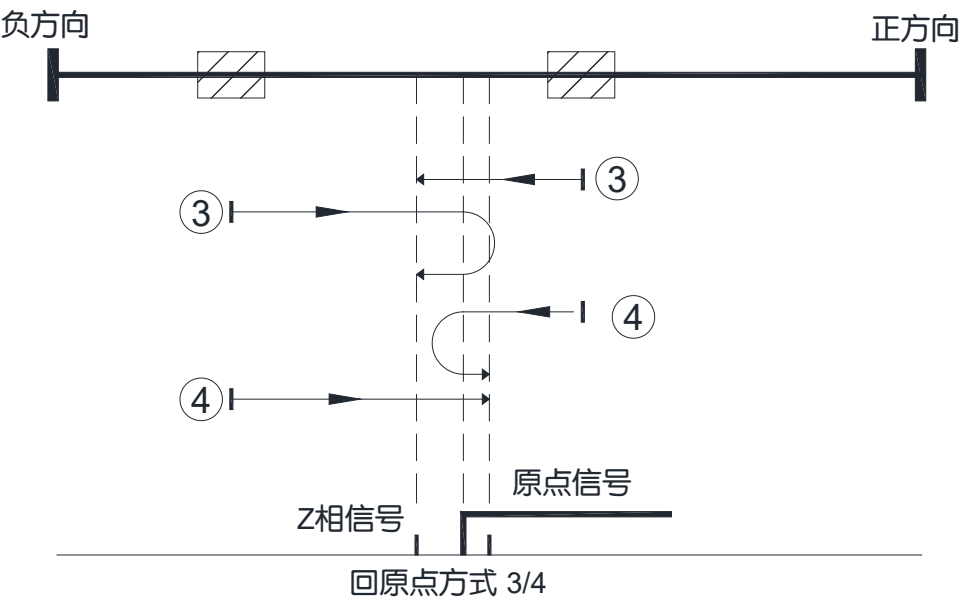
5.3.4 方式 1 和 2

如果当前限位信号未激活，电机运动方向在方式1的时候向左(负限位)或者方式2的时候向右(正限位)。原点的位置在离开限位之后到达第一个Index信号的地方。



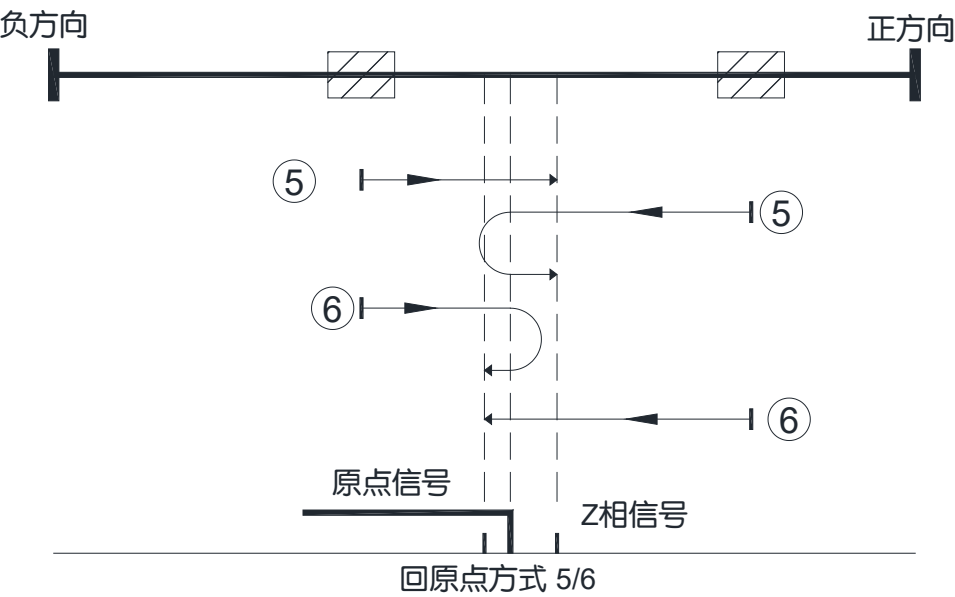
5.3.5 方式 3 和 4

初始运动方向依据当前原点信号状态。原点的位置在原点信号发生变化之后触发第一个Index信号的地方。在执行回原点过程中，如果初始位置在原点位置方向之后，电机运动方向将发生改变，发生改变点将在原点信号状态发生改变之后任意位置。



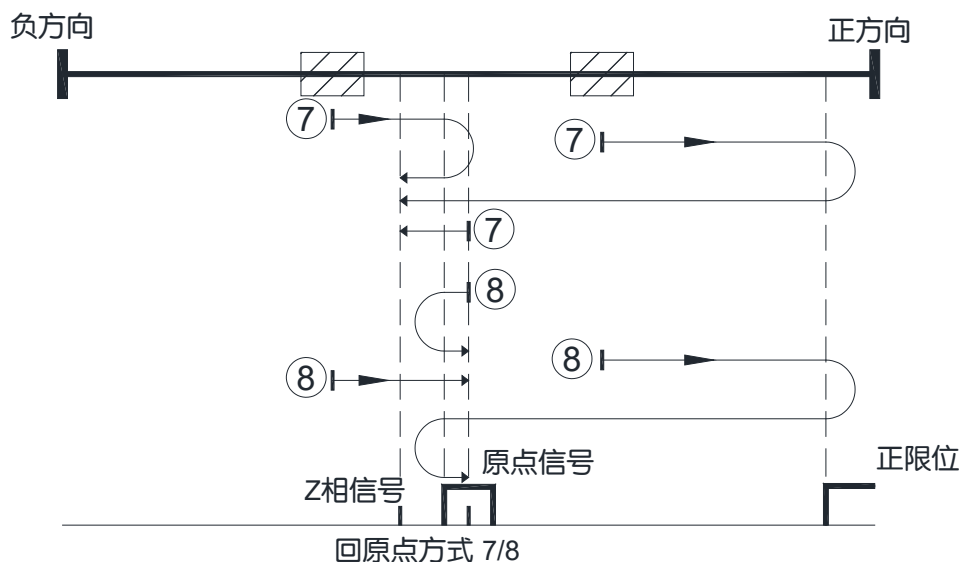
5.3.6 方式 5 和 6

初始运动方向依据当前原点信号状态。原点的位置在原点信号发生变化之后触发第一个Index信号的地方。在执行回原点过程中，如果初始位置在原点位置方向之后，电机运动方向将发生改变，发生改变点将在原点信号状态发生改变之后任意位置。



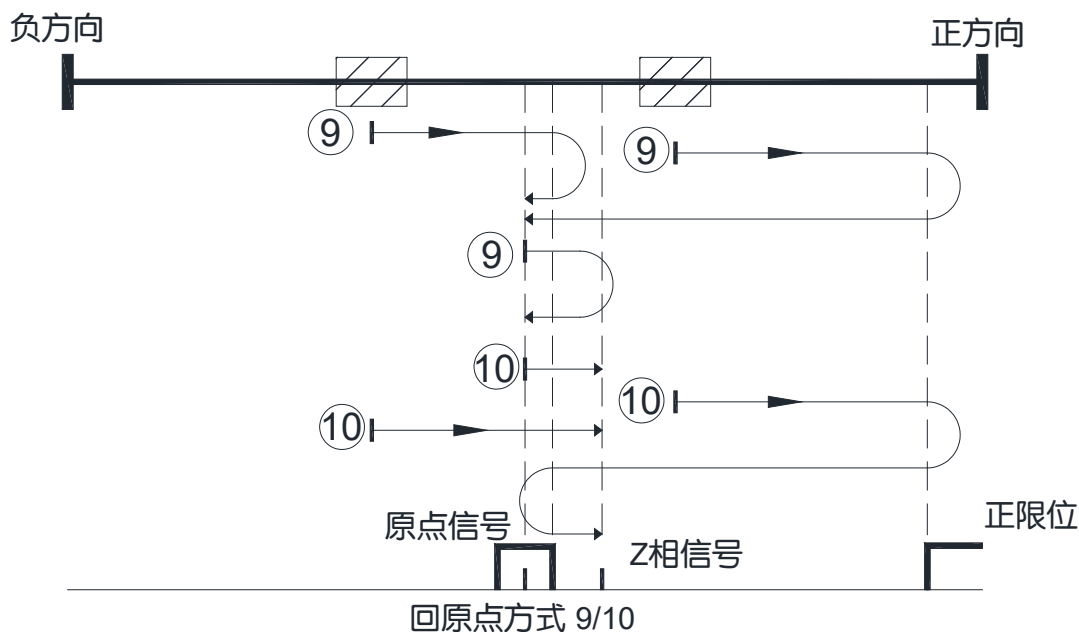
5.3.7 方式 7 和 8

如果正向限位开关未激活，则初始运动方向应向右。使用方法7，原点位置应在原点开关的左侧，原点信号状态下降沿方向第一个Index信号处。方法8是原点位置应在原点开关左侧，原点信号状态上升沿方向第一个Index信号处。



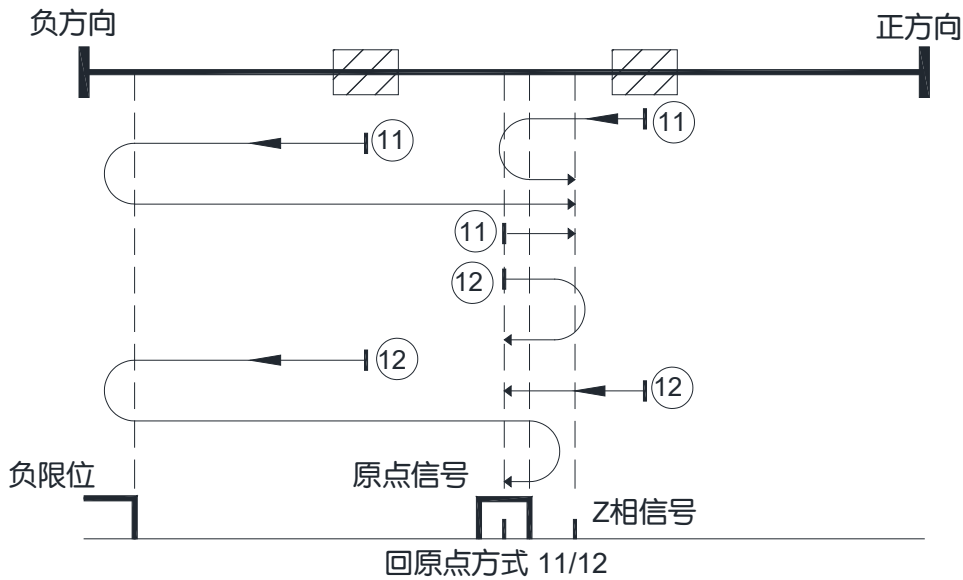
5.3.8 方式 9 和 10

如果正向限位开关未激活，则初始运动方向应向右。使用方法9，原点位置应在原点开关的右侧，原点信号状态上升沿方向第一个Index信号处。方法10是原点位置应在原点开关右侧，原点信号状态下降沿方向第一个Index信号处。



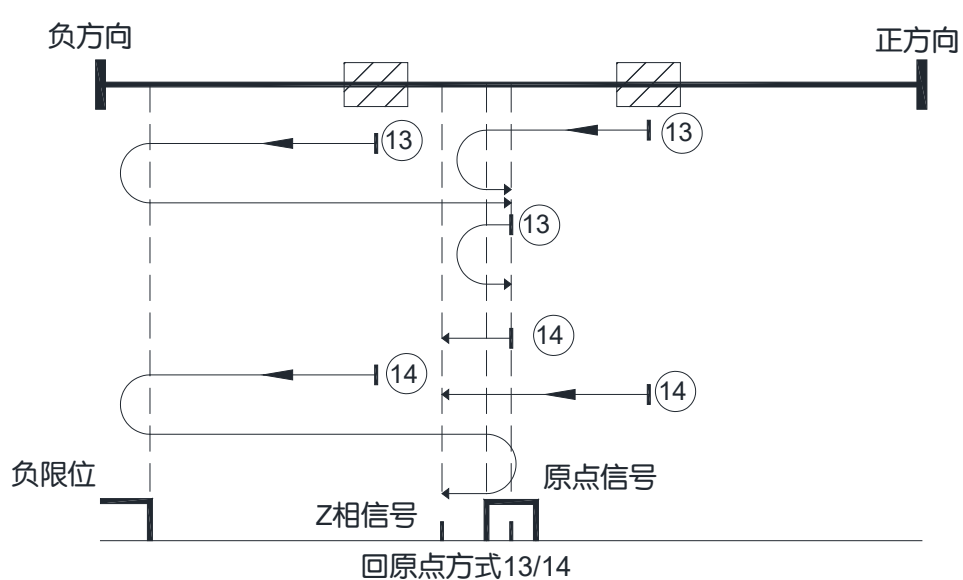
5.3.9 方式 11 和 12

如果负限位开关未激活，则初始运动方向应向左。使用方法11，原点位置应在原点开关的右侧，原点信号状态下降沿方向第一个Index信号处。方法12是原点位置应在原点开关右侧，原点信号状态上升沿方向第一个Index信号处。



5.3.10 方式 13 和 14

如果负向限位开关未激活，则初始运动方向应向左。使用方法13，原点位置应在原点开关的左侧，原点信号状态上升沿方向第一个Index信号处。方法14是原点位置应在原点开关左侧，原点信号状态下降沿方向第一个Index信号处。

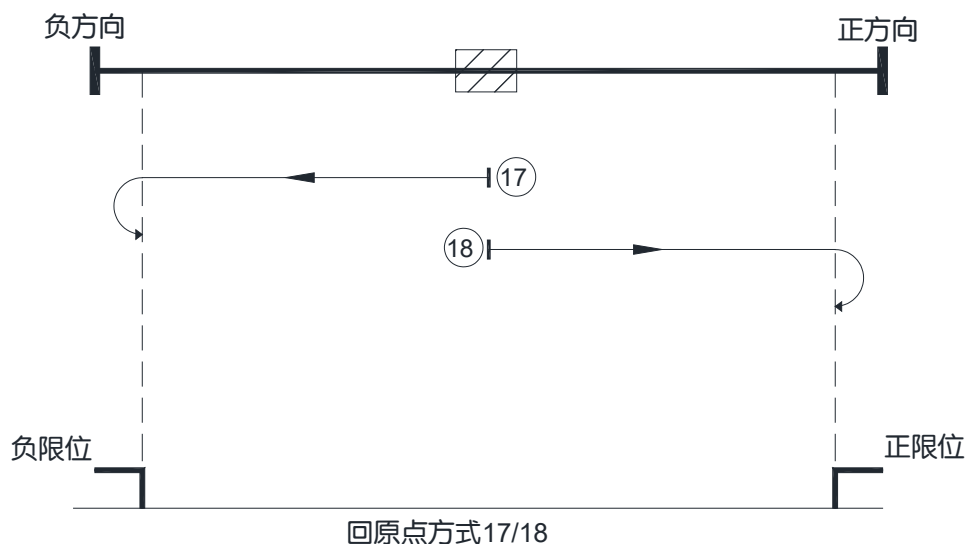


5.3.11 方法 15 和 16

预留

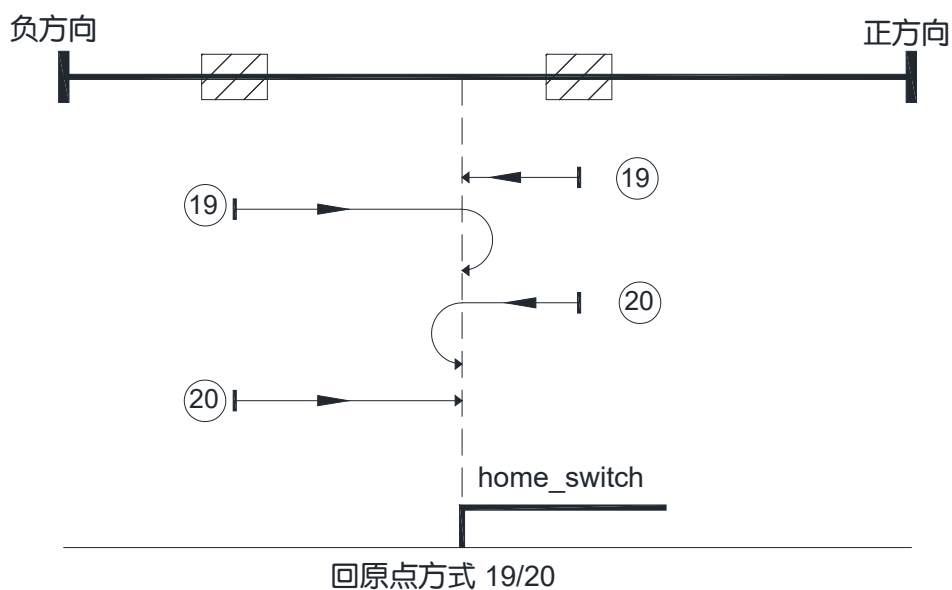
5.3.12 方法 17 和 18

方法17和18与方法1和2相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



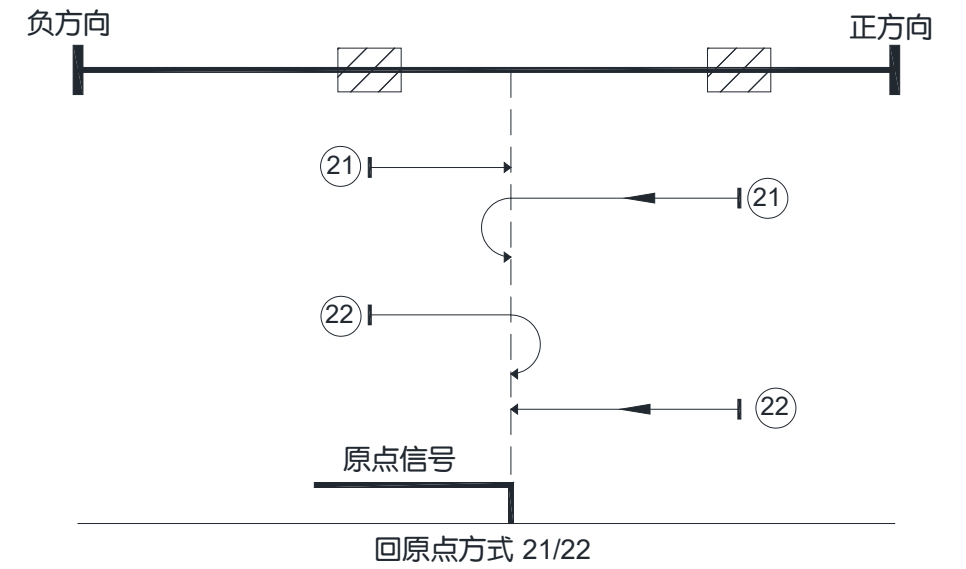
5.3.13 方法 19 和 20

方法19和20与方法3和4相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



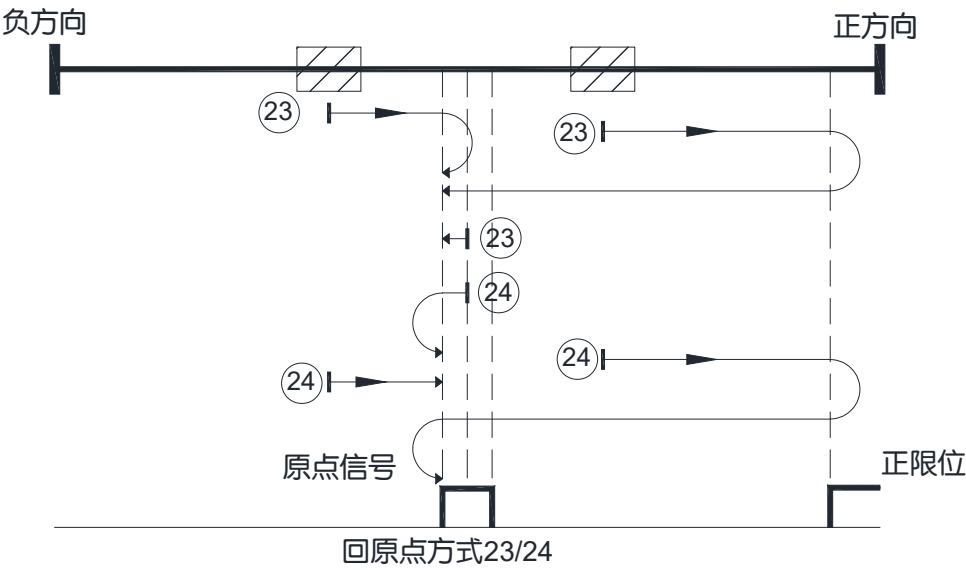
5.3.14 方法 21 和 22

方法21和22与方法5和6相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



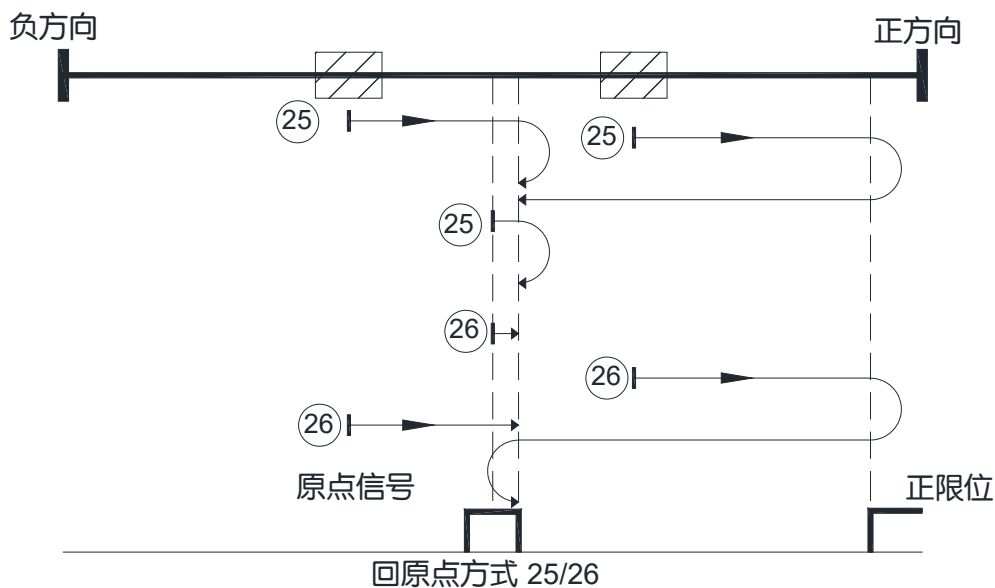
5.3.15 方法 23 和 24

方法23和24与方法7和8相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



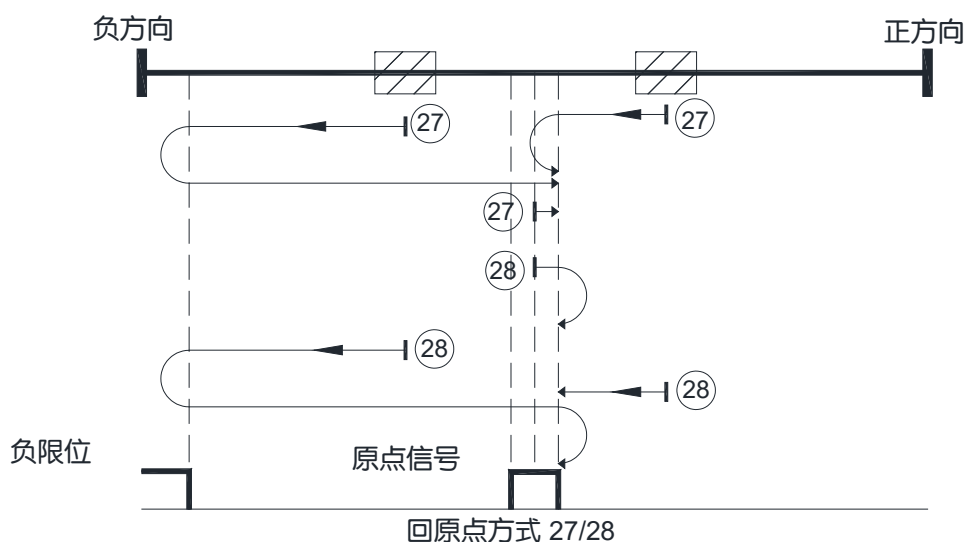
5.3.16 方法 25 和 26

方法25和26与方法9和10相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



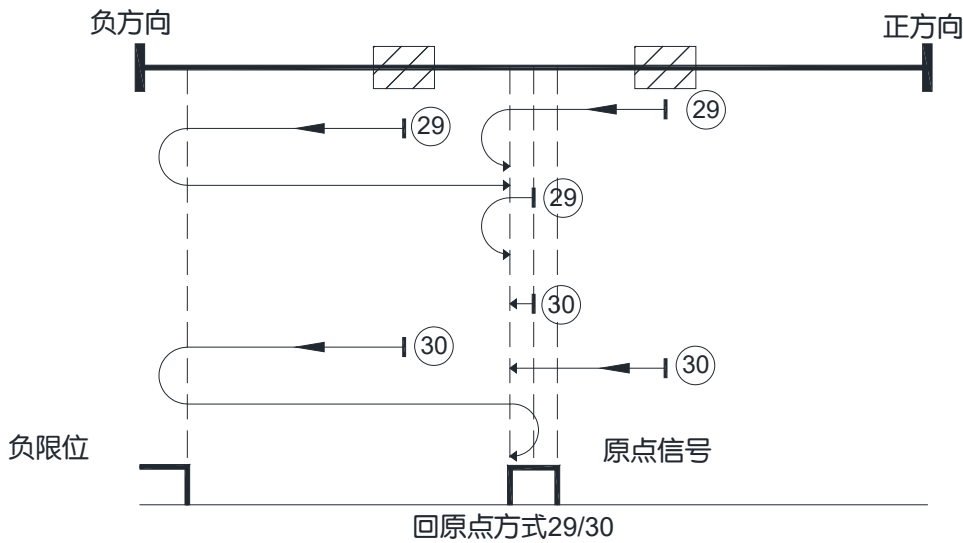
5.3.17 方法 27 和 28

方法27和28与方法11和12相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



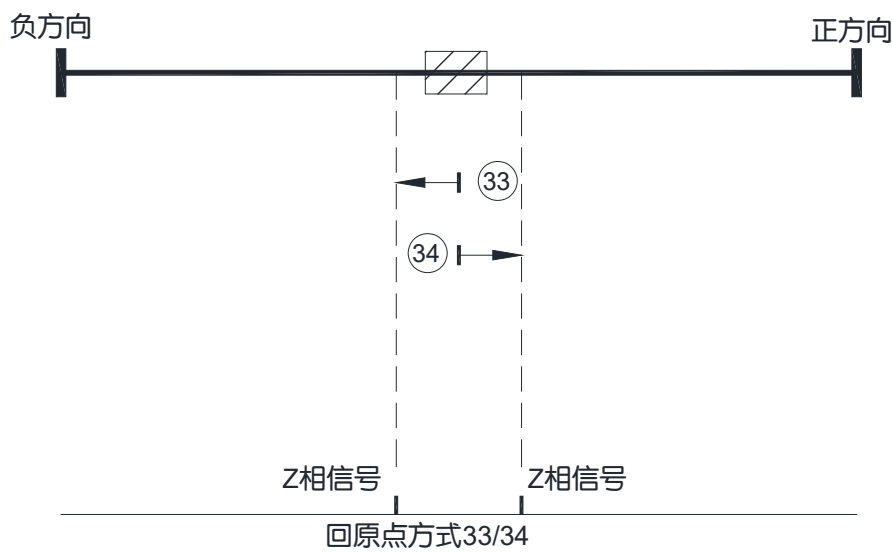
5.3.18 方法 29 和 30

方法29和30与方法13和14相似。不同之处在于，原点位置不取决于Index信号，而仅取决于相关的原点信号或限位开关。



5.3.19 方法 33 和 34

用方法33或34，方法33电机选择负方向移动，方法34选择正方向运动，原点位置即该方向第一个Index信号位置。



5.3.20 方法 35

用这种方法，当前位置即为原点位置。此方法不需要驱动设备处于操作运行状态。

5.3.21 方法 36

预留

5.3.22 方法 37

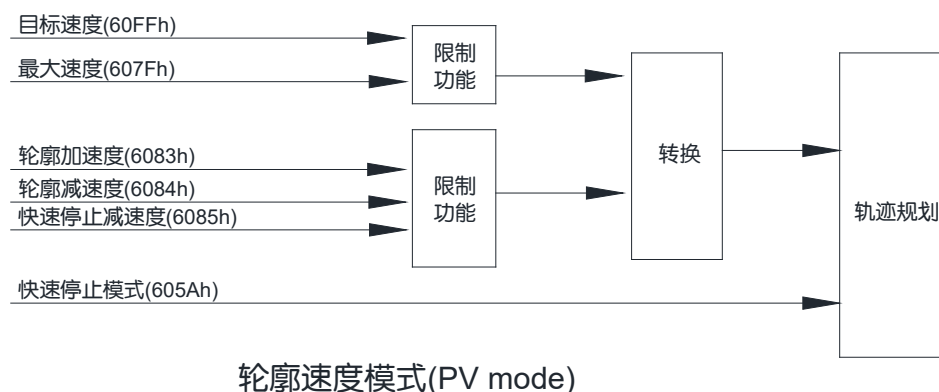
在这种方法中，位置传感器信息(以用户定义的位置单位转换)作为原点位置。此方法不需要驱动设备处于操作运行状态。在 原点位置(即回原点后)，位置实际值(6064h)的计算方式如下：

$$\text{Position actual value (6064h)} = \text{Home offset (607Ch)}$$

5.4 轮廓速度模式

5.4.1 模式说明

轮廓速度模式是一种相对简单的操作模式。一旦设置了速度，加速度和减速度，驱动器将根据加速度值驱动电机加速到运行速度，或者根据减速度值减速停止运动。



5.4.2 主要控制寄存器对象

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6060	Modes of operation	INT8	WO	YES
0x6061	Modes of operation display	INT8	RO	YES
0x60FF	Target velocity	UINT32	RW	YES
0x6083	Profile acceleration	UINT32	RW	YES
0x6084	Profile deceleration	UINT32	RW	YES

轮廓速度模式下的控制字

15	9	8	7	6	5	4	3	0
***		Halt	****	***	***	***	***	

***: 详情参考对象描述

Bit	Name	Value	Description
8	Halt	0	电机被执行或者继续
		1	电机轴根据halt option code停止运动

轮廓速度模式下的状态字

15	14	13	12	11	10	9	0
***	Max slippage error	Speed	***	Target reached	***		

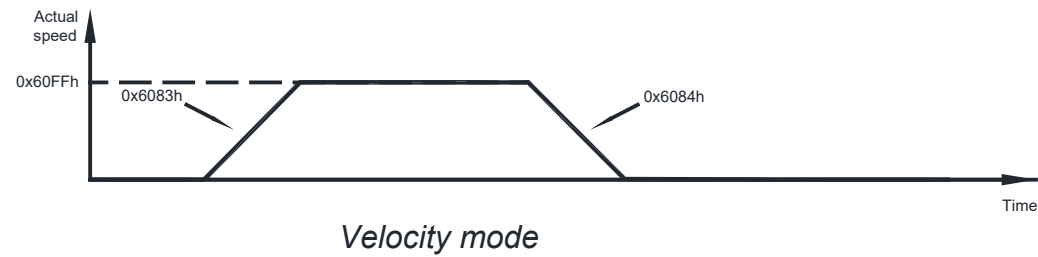
***: 详情参考对象描述

Bit	Name	Value	Description
10	Target reached	0	Halt (bit 8 in controlword) = 0: 目标未到达 Halt (bit 8 in controlword) = 1: 电机减速停止
		1	Halt (bit 8 in controlword) = 0: 目标到达 Halt (bit 8 in controlword) = 1: 电机速度为0
12	Speed	0	速度不为0
		1	速度为0
13	Max slippage error	0	最大偏移未到达
		1	最大偏移到达

5.4.3 功能描述

轮廓速度模式通过设定的速度，加速度，减速度运动，通过Halt(0x6040_bit8)停止运动控制。

- 1. 目标速度(Target velocity 0x60FFh)
- 2. 轮廓加速度(Profile acceleration 0x6083h)
- 3. 轮廓减速度(Profile deceleration 0x6084h)

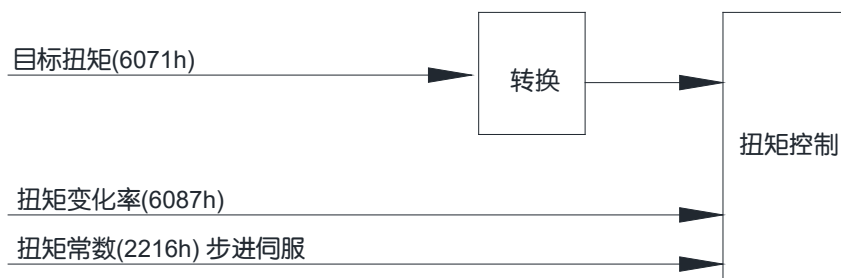


Event		Set parameter
使能电机	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(010Fh)
设置运动模式	轮廓速度模式	0x6060=3(3h)
设置参数	目标速度	0x60FF=20000(4E20h)
	加速度	0x6083=50000(C350h)
	减速度	0x6084=50000(C350h)
速度模式运行	启动	0x6040=F(15h)
	停止	0x6040=271(010Fh)

5.5 扭矩模式 (步进伺服支持)

5.5.1 模式说明

扭矩模式是一个伺服控制操作模式，它需要知道关于电机的扭矩常数，单位为mN/A。在电机规格里面可以找到这方面信息。



扭矩模式(PT mode)

5.5.2 主要控制寄存器对象

Index	Name	Type	Access	Mapping
0x6040	Controlword	UINT16	WO	YES
0x6041	Statusword	UINT16	RO	YES
0x6071	Target torque	INT16	RW	NO
0x6087	Torque slope	INT8	WO	YES
0x2216	Torque constant	INT8	RO	YES

扭矩模式下的操作字

15	9	8	7	6	5	4	3	0
***		Halt	****	***		***		***

***: 详情参考对象描述

Bit	Name	Value	Description
8	Halt	0	运动被执行或者继续
		1	电机通过Halt option code(0x605D)设定方式停止

扭矩模式下的状态字

15	14	13	12	11	10	9	0
***		Reserved		***	Target reached		***

***: 详情参考对象描述

Bit	Name	Value	Description
10	Target reached	0	Halt (bit 8 in controlword) = 0: 目标未到达
			Halt (bit 8 in controlword) = 1: 电机减速度停止
		1	Halt (bit 8 in controlword) = 0: 目标到达
			Halt (bit 8 in controlword) = 1: 电机速度为0

5.5.3 功能描述

激活扭矩模式，需要设置如下参数：

Index	Name	Description
0x2216	扭矩常数	电机参数，在电机规格里面有相关信息
0x6071	目标扭矩	作用到电机上的扭矩
0x6087	扭矩变化率	改变扭矩变化速度

参数计算 – 举例

一个应用要求扭矩为0.353Nm，扭矩变化率为0.177Nm/sec。我们找到扭矩常数为0.07Nm/A。然后我们将这些值写到寄存器对象中。

Index	Value	Units
0x2216	70	m•Nm/A
0x6071	353	m•Nm
0x6087	177	m•Nm/sec

电流验证 – 举例

对于步进伺服来说，检测驱动器所需要的电流是否是在伺服放大器以内是非常重要的。例如，所使用的驱动器的额定连续电流为7安培，峰值电流为14安培，可以连续保持2秒钟。这意味着可以连续保持7安培的电流运行，并且可以在短时间内使用7安培和14安培之间的电流。

使用上面示例中的目标扭矩和扭矩常数，可以检查电流消耗，如下所示：

$$0.353\text{Nm}/(0.07\text{ Nm/A}) = 5.044\text{A}$$

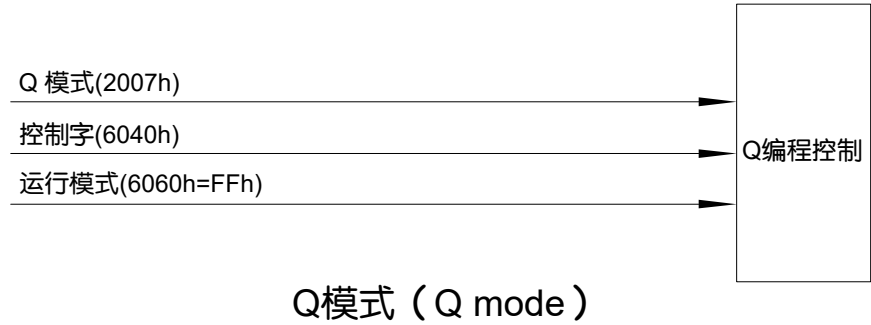
产生的电流5.044A低于驱动器的7A连续额定电流，并且远低于14A的峰值额定电流。驱动器可能会持续保持7A的电流，并在持续两秒钟内达到14A的峰值。7A和14A之间的值可以按比例保持较长时间。

	Event	Set parameter
使能电机	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(010Fh)
设置运行模式	Profile velocity mode	0x6060=4(04h)
设置参数	目标扭矩	0x6071=20(14h)
	扭矩变化率	0x6087=500(01F4h)
	扭矩常数	0x2216=100(64h)
扭矩模式运行	启动	0x6040=15(0Fh)
	停止	0x6040=271(10Fh)

5.6 Q 程序模式

5.6.1 模式说明

为了扩展鸣志的CANopen驱动器的功能，可以使用Q编程语言来执行复杂的运动曲线，这在CiA402协议的范围内是不可能的。必须使用软件编写Q程序并将其预先加载到CANopen驱动器中。Q程序还可以访问和操作存储的CANopen通用寄存器。



5.6.2 常规执行Q程序

通过常规方式执行储存在驱动器里面的Q程序，需要在Mode of operation操作模式 (0x6060h)对象里面写-1(FFh)值。当前运行模式可以通过Mode of operation display(0x6061h)来获取。0x6061h显示当前执行的操作模式。同时需要将要执行的Q程序通过软件写入对应的程序段，保存到驱动器。

控制对象寄存器-举例：

	Event	Set parameter
使能驱动器	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(10Fh)
设置运行模式	Normal Q mode	0x6060= -1(FFh)
设置需要运行的Q程序段	Q procedure segment	0x2007=1(01h)
执行Q运行	启动	0x6040=31(1Fh)
	停止	0x6040=287(011Fh)

5.6.3 同步方式执行Q程序

通过同步方式执行储存在驱动器里面的Q程序，需要在Mode of operation操作模式(6060h)对象里面写-2(FFh)值。当前运行模式可以通过Mode of operation display(0x6061h)来获取。0x6061h显示当前执行的操作模式。同时需要将要执行的Q程序通过软件写入对应的程序段，保存到驱动器。

控制对象数据-举例：

Event	Set parameter	
使能电机	Shut down	0x6040=6(06h)
	Switch on	0x6040=7(07h)
	Switch on + enable operation +halt	0x6040=271(10Fh)
设置运行模式	同步方式执行Q程序	0x6060=-2(FFh)
设置参数	同步信号	0x1005=128(80h)
	Q程序段	0x2007=1(01h)
同步执行Q程序	启动	COB-ID sync message"80"
	停止	0x6040=271(10Fh)

5.7 全局控制字与状态字说明

BIT	控制字(0x6040h)					状态字(0x6041h)				
	原点	位置	速度	扭矩	Q	原点	位置	速度	扭矩	Q
15	预留					预留				
14						预留				
13						Homing error	Following error	MAX slippage error	预留	预留
12						Homing attained	Set point acknowledge	Speed	预留	预留
11						Internal limit active				
10						Target reached				
9	预留	Change on set point	预留	预留	预留	Remote				
8	Halt					预留				
7	Fault reset					Warning				
6	预留	Abs/Rel	预留	预留	预留	Switched on disabled				
5	预留	Change set immediately	预留	预留	预留	Quick stop				
4	Homing operation start	New set point	预留	预留	Q program start	Voltage enabled				
3	Enable operation					Fault				
2	Quick stop					Operation enabled				
1	Enable voltage					Switched on				
0	Switch on					Ready to switch on				

6 应用功能

6.1 通讯看门狗

通讯看门狗监控CANopen的通讯状态。如果通讯连接中断被发现，并且延续一段时间(通过0x2060-03h定义)，相应动作(0x2060_05h)将会被触发。

举例说明如何使用通讯看门狗做安全控制。

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	CONST	NO	NO	4
1	Watchdog enable	UINT8	RW	NO	NO	0
2	Watchdog status	UINT8	RW	NO	NO	0
3	Watchdog timeout	UINT16	RW	NO	NO	0x01F4
4	Watchdog trigger event	UINT8	RW	NO	NO	0x0F
5	Timeout option code	UINT16	RW	NO	NO	0

首先需要设置通讯看门狗相关参数：

Object	Name	Description
0x2060_03h	Watchdog timeout	该对象设置通讯看门狗循环监控时间，单位1ms
0x2060_04h	Watchdog trigger event	该对象设置循环触发条件
		0x01：RPDO1作为循环喂狗条件
		0x02：RPDO2 作为循环喂狗条件
		0x04：RPDO3 作为循环喂狗条件
		0x08：RPDO4 作为循环喂狗条件
0x2060_05h	Watchdog timeout option code	0x10：SYNC信息作为循环喂狗条件
		0x0：电机停止运动并去使能
		0x1：电机停止运动并保持使能
		0x2-0x14：驱动器执行相对应的Q程序1-12段

然后0x2060_01h写1使能watchdog功能，并且0x2060h_02h状态也变为01。

注意：0x2060h必须在operation运行模式下才可以被设置。

当CANopen通讯被中断之后，0x2060_01h将会被设置为0，0x2060_02h设置为1。

往0x2060h_02h里面写1可以重启通讯看门狗功能，再往0x2060_01h可以再次激活看门狗。

7 对象字典

对象字典(OD)是任何CANopen节点的核心。它提供了到节点的所有通信和运行参数的链接。在电子数据表(EDS)中定义了对象字典，该字典列出了所有受支持的对象以及所有子对象。

可以使用标准服务数据对象(SDO)协议访问任何OD条目，某些OD条目可以使用低负载过程数据对象(PDO)协议访问。

7.1 对象描述

设备配置文件中最重要的是对象字典描述。对象字典本质上是一组通过有序预定义的闪存空间，可通过网络访问的对象。字典中的每个对象都使用16位索引进行寻址。标准对象字典的整体布局如下所示。此布局非常符合其他工业串行总线系统概念：

索引(hex)	对象(Object)
0	不用(Not used)
0001-001F	静态数据类(Static data types)
0020-003F	补充数据类(Complex data Types)
0040-005F	厂商自定义补充数据类(Manufacturer Specific Complex Data Types)
0060-007F	设备配置文件规定的静态数据类(Device Profile Specific Static Data Types)
0080-009F	设备配置文件规定的补充数据类(Device Profile Specific Complex Data Types)
00A0-0FFF	预留后续用(Reserved for further use)
1000-1FFF	通讯配置区域(Communication Profile Area)
2000-5FFF	厂商规定配置区域(Manufacturer Specific Profile Area)
6000-9FFF	标准设备配置区域(Standardized Device Profile Area)
A000-FFFF	预留后续用(Reserved for further use)

7.2 电子数据表格

鸣志网站上提供的EDS文件列出了OD中每个受支持对象的所有属性，在此之后，列出了步进和步进伺服驱动器的“对象字典”描述。

7.2.1 通讯配置文件1000h(CiA301)

Index	Sub	Name	Type	Access	Mapping
0x1000	-	Device type	UINT32	RO	NO
0x1001	-	Error register	UINT8	RO	YES
0x1002	-	Manufacture status register	UINT32	RO	YES
0x1003	8	Pre-defined error field	ARRAY	-	-
0x1005	-	COB-ID SYNC message	UINT32	RW	NO
0x1006	-	Communication cycle period	UINT32	RW	NO
0x1007	-	Synchronous window length	UINT32	RW	NO
0x1008	-	Manufacturer device name	STRING	RO	NO
0x1009	-	Manufacturer hardware version	STRING	RO	NO
0x100A	-	Manufacturer software version	STRING	RO	NO
0x1010	2	Store parameters	ARRAY	RO	NO
0x1011	2	Restore default parameters	ARRAY	RO	NO
0x1014	-	COB-ID EMCY	UINT32	RO	NO
0x1017	-	Producer heartbeat time	UINT16	RW	NO
0x1018	4	Identity object	ARRAY	-	-
0x1019	-	Synchronous counter overflow value	UINT8	RW	NO
0x1029	6	Error behavior object	ARRAY	-	-
0x1200	2	SDO server parameter	ARRAY	-	-
0x1400	2	RPDO communication parameter1	ARRAY	-	-
0x1401	2	RPDO communication parameter2	ARRAY	-	-
0x1402	2	RPDO communication parameter3	ARRAY	-	-
0x1403	2	RPDO communication parameter4	ARRAY	-	-
0x1600	8	RPDO mapping parameter1	ARRAY	-	-
0x1601	8	RPDO mapping parameter2	ARRAY	-	-
0x1602	8	RPDO mapping parameter3	ARRAY	-	-
0x1603	8	RPDO mapping parameter4	ARRAY	-	-
0x1800	6	TPDO communication parameter1	ARRAY	-	-
0x1801	6	TPDO communication parameter2	ARRAY	-	-
0x1802	6	TPDO communication parameter3	ARRAY	-	-
0x1803	6	TPDO communication parameter4	ARRAY	-	-
0x1A00	8	TPDO mapping parameter1	ARRAY	-	-
0x1A01	8	TPDO mapping parameter2	ARRAY	-	-
0x1A02	8	TPDO mapping parameter3	ARRAY	-	-
0x1A03	8	TPDO mapping parameter4	ARRAY	-	-

0x1000 Device type

包含有关设备类型的信息。该对象描述了设备的类型及其功能。它包含一个16位的描述设备文件或者应用文件以及另外一个16位的附加信息组成。附加的信息参数是设备特定的配置文件。它的规范不属于标准的范围，它是在相应的设备配置文件中定义的。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	-

Bit 0-15: 设备配置文件编号

Bit 16-31: 附加信息

注意:

COS: 可以通过TPDO获取变化信息

0x1001 Error register

该对象描述设备的错误寄存器。设备在该寄存器记录内部错误，所有设备必须记录。它是紧急对象的一部分。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RO	YES	NO	-

Bit 0: 普通错误

Bit 1: 电流

Bit 2: 电压

Bit 3: 温度

Bit 4: 通讯错误 (过载, 状态错误)

Bit 5: 预留 (总是 0)

Bit 6: 预留 (总是 0)

Bit 7: 预留 (总是 0)

0x1002 Manufacturer status register

该对象是用于制造商特定目的的通用状态寄存器。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	YES	NO	0

该对象总是 0。

0x1003 Pre-defined error field

该对象保留了设备上发生的错误，并已通过“紧急对象”发出了信号。这样做可以提供错误历史记录。

1. 子索引sub-index 0包含从子索引1开始的数组中记录的实际错误数。
2. 每一个新的错误都会储存在sub-index 1里面，旧的会按照列表顺序移动到下一个。
3. 往sub-index 0 里面写0可以清除历史错误 (清空数组)。大于0的数值是不被允许写入的。这必然导致中止消息(错误代码: 06090030h)。
4. 这32位错误数字包含一个16位的错误代码和一个16位关于厂商自定义的其他错误信息。错误代码包含在低位2字节，厂商自定义的其他错误信息在高位2字节数据里面。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	8

Sub index	Name	Data Type	Access Type	Default Value	PDO Mapping
0	Number of errors	UNSIGNED8	RW	0	NO
1	Standard error field	UNSIGNED32	RO	0	NO
2	Standard error field	UNSIGNED32	RO	0	NO
3	Standard error field	UNSIGNED32	RO	0	NO
4	Standard error field	UNSIGNED32	RO	0	NO
5	Standard error field	UNSIGNED32	RO	0	NO
6	Standard error field	UNSIGNED32	RO	0	NO
7	Standard error field	UNSIGNED32	RO	0	NO
8	Standard error field	UNSIGNED32	RO	0	NO

错误数目 (sub index = 0):

Bit 0-7: 写0可以清除历史错误

标准错误区域 (sub index = 1~ 8):

Bit 0-15: 错误代码通过紧急对象触发

Bit 16-31: 厂商规定增加的信息

0x1005 COB-ID SYNC message

该对象定义了同步发生器对象的COB-ID，此外，他定义了设备是否产生SYNC信号。

x	gen.	frame	0 0000h	11-bit CAN-ID
			29-bit CAN-ID	

Bit	Value	Description
x	x	Do not care
gen.	0	CANopen device does not generate SYNC message
	1	CANopen device generates SYNC message
frame	0	11-bit ID valid (CAN base frame)
	1	29-bit ID valid (CAN extended frame)
29-bit CAN-ID	X	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	X	11-bit CAN-ID of the CAN base frame

位29、30属于静态的(不可更改)。如果一个CANopen设备无法生成SYNC消息，则试图设置bit30为1的时候，驱动器通过中止消息(中止代码：06090030h)来响应。当CANopen设备只支持标准CAN帧类型，则试图更改位29的时候，驱动器通过中止消息来响应(中止代码：0609 0030h)。将位30设置为1后，SYNC对象的第一次传输将在1个同步周期内开始。当0x1019(synchronous counter overflow value)的值大于1，通过设置bit30为1，则第一个SYNC消息开启的同时0x1019重新设置为1。当该对象存在(bit30=1)，则不允许更改位0-29。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	0x00000080

Bit 0-10: COB-ID for SYNC object

Bit 11-29: set to 0

Bit 30: 1(0) - node generates (does NOT generate) SYNC object

Bit 31: set to 0

0x1006 Communication cycle period

对象定义通讯周期，以毫秒为单位。该周期定义了SYNC间隔。如果不使用，则为0。如果同步生成器上的通信周期更改为不等于0的新值，则同步对象的传输将在新值的1个同步周期内恢复。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	0

位 0-31: SYNC循环通讯周期单位为ms(0表示不使用，不检测)

0x1007 Synchronous window length

该对象包含同步PDO的时间窗口长度，单位为ms，如果是0表示不使用

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	0

0x1008 Manufacturer device name

该对象包含自定义设备名称

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	Visible string	CONST	NO	NO	AMA CANopen Motor Driver

名称单位为string。

0x1009 Manufacturer hardware version

该对象包含产商硬件版本号描述。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	Visible string	CONST	NO	NO	-

名称单位为string。

0x100A Manufacturer software version

该对象包含厂商软件版本号描述

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	Visible string	CONST	NO	NO	-

名称单位为string。

0x1010 Store parameters

该对象支持将参数保存在非易失性存储器中。通过读取访问设备 提供有关其保存功能的信息。区分了几个参数组：

为了避免错误地存储参数，仅在特定参数被写入时才执行存储。参数为“save(保存)”。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	2

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	RO	2	NO
1	Save all parameter	UNSIGNED32	RW	0	NO
2	Save communication parameters	UNSIGNED32	RW	0	NO

Signature ISO 8859("ASCII")hex	MSB		LSB	
	e	v	a	s
	65h	76h	61h	73h

Storage write access signature

1 Save config parameters

步进驱动器

Index	Name	Note
0x2601	Idle current	
0x2603	Acceleration current	
0x2604	Steps per revolution	
0x2607	Brake configuration	
0x2609	Alarm reset input	
0x260A	Jerk delay time	
0x260B	Define limits	
0x260C	Input filter	sub-index 3...8
0x260D	Motor selection	
0x260E	Open winding detect velocity limit	
0x260F	Drive configuration	
0x2610	Alarm mask	
0x2611	Load ratio	
0x2612	Custom motor	sub-index1...15
0x2613	Pulse noise filter	
0x2614	Smooth filter	
0x2615	Move output	
0x2616	Alarm output	
0x2617	Servo enable	

步进伺服驱动器

Index	Name	Note
0x2201	Peak current	
0x2202	Hard stop current	
0x2203	Idle current	
0x2204	In position error range	
0x2205	In position time	
0x2208	Rotation reverse	
0x2209	Velocity max	
0x220A	Acceleration max	
0x220B	Smooth filter	
0x220C	Brake configuration	
0x220D	Servo enable	
0x220E	Alarm reset input	
0x220F	Define limits	
0x2210	Input filter	sub-index 3...8
0x2211	Notch filter	
0x2212	Analog configuration	
0x2213	Alarm output	
0x2214	Move output	
0x2215	Jog mode	
0x221A	Alarm mask	
0x2220	Position gain	
0x2221	Position derigain	
0x2222	Position derifilter	
0x2224	Velocity gain	
0x2225	Velocity integergain	
0x2226	Acceleration feedforward	
0x2227	PID filter	
0x2252	In position counts	

2 Save communication parameters

Index	Name	Note
0x1005	COB-ID SYNC message	
0x1006	Communication cycle period	
0x1007	Syncchronous window length	
0x1014	COB-ID EMCY	
0x1017	Producer default parameters	
0x1019	Synchronous counter overflow value	
0x1029	Error behavior	
0x1200	SDO server parameter	
0x1400-0x1403	RPDO communication parameter	
0x1600-0x1603	RPDO mapping parameter	
0x1800-0x1803	TPDO communication parameter	
0x1a00-0x1a03	TPDO mapping parameter	

0x1011 Restore default parameters

通过该对象可以将关于通讯和设备文件的默认参数初始化。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	2

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	RO	2	NO
1	Restore all parameter	UNSIGNED32	RO	0	NO
2	Restore communication default parameters	UNSIGNED32	RO	0	NO

	MSB		LSB	
Signature ISO 8859("ASCII")hex	d	a	o	l
	64h	61h	6Fh	6Ch

Storage write access signature

0x1014 COB-ID emergency

该对象定义了紧急消息的COB-ID。

Object Type	Data Type	Access Type	PDO mapping	COS	Default value
ARRAY	UINT32	RO	NO	NO	0x80+\$NODEID

Valid	0b	Frame	0 0000h	11-bit CAN-ID
			29-bit CAN-ID	

Structure of the EMCY Identifier entry

Bit	Value	Description
Valid	0	EMCY exists / is valid
	1	EMCY does not exist / is not valid
30	0	Reserved(always 0)
Frame	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
29-bit CAN-ID	X	If bit 29=0 If bit 29=1: bits 28-11 of 29-bit-COB-ID
11-bit CAN-ID	X	Bits 10-0 of COB-ID

Description of EMCY COB-ID entry

当CANopen设备只支持CAN 标准数据类型的时候，试图设置bit29为1的，则驱动器会通过SDO发生中止消息来响应。当该对象位31=0 的时候，位0到29不能发生改变。

0x1017 Producer heartbeat time

该对象定义心跳报文循环时间，如果该对象为0表示不用，单位为ms。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	1000

0x1018 Identity object

该对象包含设备的一般信息。

1. Vendor ID包含一个对每个厂家独一无二的代码。
2. 该对象包含了厂商代码信息
3. 制造商特定的修订号(子索引3h)由一个主要修订号和一个次要修订号组成。主版本号标识特定的CANopen行为。如果扩展了CANopen功能，则主要版本必须增加。次要修订号标识具有相同CANopen行为的不同版本。
4. 制造商特定的序列号(子索引4h)标识特定的设备。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT32	RW	NO	NO	4

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Max sub-index	UNSIGNED8	RO	4	NO
1	Vendor-ID	UNSIGNED32	RO	-	NO
2	Product code	UNSIGNED32	RO	-	NO
3	Revision number	UNSIGNED32	RO	-	NO
4	Serial number	UNSIGNED32	RO	-	NO

Max sub-index

Vendor-ID

Bit 0-31: CiA分配 (上海安浦鸣志自动化设备有限公司标识为0x000002D9)

Product code

Bit 0-31: 厂商定义

Revision number

Bit 0-15: 主修订号(CANopen控制行为不变)

Bit 16-31: 次修订号 (CANopen控制行为变化)

Serial number

Bit 0-31: 厂商定义

0x1019 Synchronous counter overflow value

该对象包含SYNC消息的最大计数器。如果为0，则生产方将发送SYNC消息，不包含任何数据，否则发送数据长度为1。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

如果值为0，则以0的数据长度发送SYNC消息。如果值为2至240，则SYNC消息具有一个数据字节，其中包含计数器。其他值保留。

0x1029 Error behavior

该对象包含CANopen驱动器的错误信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RW	NO	NO	6

Sub-index	Name	Data Type	Access Type	Default value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	RO	6	NO
1	Communication	UNSIGNED8	RO	1	NO
2	Communication other	UNSIGNED8	RO	1	NO
3	Communication passive	UNSIGNED8	RO	1	NO
4	Generic	UNSIGNED8	RO	1	NO
5	Device profile	UNSIGNED8	RO	1	NO
6	Manufacturer specific	UNSIGNED8	RO	1	NO

所有子索引的值定义为:

0x00 - 如果在运行状态operation, 切换到NMT预运行状态Pre-operation

0x01 - 无动作

0x02 - 切换到NMT停止状态

子索引:

01 - 通讯错误 - 总线关闭或者心跳错误

02 - 通信其他错误(严重错误-参见“错误状态位”), 除了CAN总线被动, 包括总线关闭或心跳错误。

03 - 被动通信-任何通信错误, 包括CAN总线被动错误。

04 - 一般错误 (严重错误-参见“错误状态位”)。

05 - 设备文件错误 - 错误寄存器中的第7位被设置。

06 - 制造商特定的错误-错误寄存器中的第7位被设置。

0x1200 SDO server parameter

该对象包含用于访问驱动器SDO的COB-ID(通信对象ID, 也称为CAN消息ID)值。

1. Sub-index 0记录该对象子索引数量

2. Sub-index 1用来记录驱动器RPDO数据COB-ID, 值为0x600+Node ID。

3. Sub-index 2 用来记录驱动器TPDO数据COB-ID, 值为0x580+Node ID。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
RECORD	UINT32	RW	NO	NO	2

sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	NO	NO	2
1	COB-ID client to server	UNSIGNED32	RO	NO	NO	0x600+\$NODEID
2	COB-ID server to client	UNSIGNED32	RO	NO	NO	0x580+\$NODEID

0x1400~0x1403 Receive PDO communication parameter

该对象包含设备能够接收到RPDO的通信参数。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	2

sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	NO	NO	2
1	COB-ID used by RPDO	UNSIGNED32	RO	NO	NO	0x1400h: 0x200+\$NODEID 0x1401h: 0x300+\$NODEID 0x1402h: 0x400+\$NODEID 0x1403h: 0x500+\$NODEID
2	Transmission type	UNSIGNED32	RO	NO	NO	0xFF

Sub-index 0包含通讯记录中有效条目的数量，其值至少为2。

Sub-index 1里面包含RPDO的COB-ID。

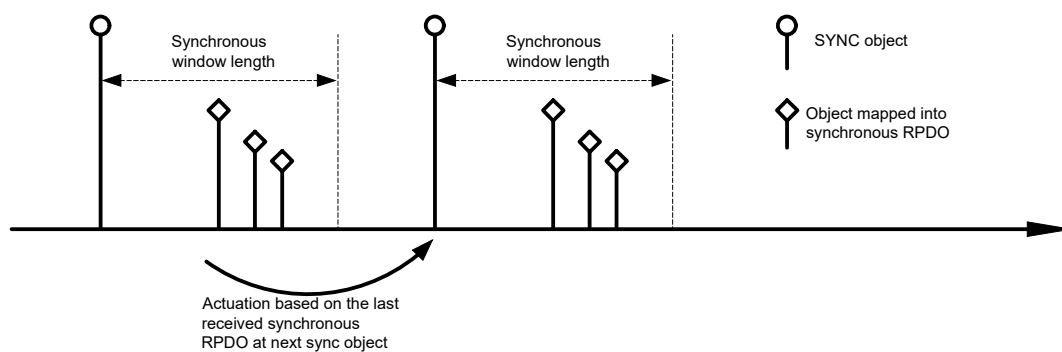
31	30	29	28	11	10	0
Valid	Reserved	Frame	00000h		11-bit CAN-ID	
			29-bit CAN-ID			

Bit(s)	Value	Description
Valid	0	PDO exists/ is valid
	1	PDO does not exists/ is not valid
Reserved	X	Do not care
Frame	0	11bit CAN-ID valid(CAN base frame)
	1	29bit CAN-ID valid(CAN extended frame)
29-bit CAN-ID	X	29-bit CAN-ID of the CAN extended frame
11-bit CAN-ID	X	11-bit CAN-ID of the CAN base frame

Sub-index 2 里面定义RPDO数据接收的方式，如果尝试将一些不支持的值写入进去改变传输类型方式，则应通过SDO进行响应中止传输服务(中止代码：0609 0030h)。

Value	Description
00	Synchronous
.....
F0	Synchronous
F1	Reserved
.....
FD	Reserved
FE	Event-driven(manufacturer-specific)
FF	Event-driven(device profile and application specific)

- Synchronous表示在下一个SYNC信号发生之后CANopen驱动器响应接收到的数据，如下图
- Event-driven意思为PDO数据可以在任何时间接收，CANopen驱动器将立刻响应该接收的数据。



0x1600~0x1603 Receive PDO mapping parameter

该对象包含CANopen设备可以接收的PDO映射参数。

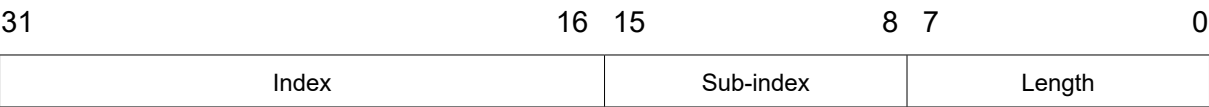
Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	8

sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Number of mapped application objects in RPDO	UNSIGNED8	RW	NO	NO	1
1	1st application object	UNSIGNED32	RW	NO	NO	0x060400010
2	2st application object	UNSIGNED32	RW	NO	NO	0x00000000
3	3st application object	UNSIGNED32	RW	NO	NO	0x00000000
4	4th application object	UNSIGNED32	RW	NO	NO	0x00000000
5	5th application object	UNSIGNED32	RW	NO	NO	0x00000000
6	6th application object	UNSIGNED32	RW	NO	NO	0x00000000
7	7th application object	UNSIGNED32	RW	NO	NO	0x00000000
8	8th application object	UNSIGNED32	RW	NO	NO	0x00000000

Sub-index 0 Number of mapped application objects in RPDO:
包含映射记录或特定记录中有效对象条目的数量。

Value	Description
00	Mapping disabled
01	Sub-index 01h valid
02	Sub-index 01h and 02h valid
03	Sub-index 01h to 03h valid
04	Sub-index 01h to 04h valid
05	Sub-index 01h to 05h valid
06	Sub-index 01h to 06h valid
07	Sub-index 01h to 07h valid
08	Sub-index 01h to 08h valid

Sub-index 01h~08h application object:
包含映射的应用对象信息。该对象的索引，子索引和数据长度通过该PDO来描述(如下图)。数据长度包含应用程序对象的长度(以bit为单位)，这可以用于验证映射。



Structure of PDO Mapping Entry

如果支持，应使用以下过程进行新的映射，该过程需要在“预运行(Pre-operational)”期间和“运行(Operational)”期间进行：

1. 通过0x1400~1403(RPDO communication parameter)的Sub-index 01的有效(valid)控制位关闭RPDO功能
2. 通过设置Sub-index 0h为0来关闭映射
3. 通过更改相应子索引的值来修改映射
4. 通过将子索引00h设置为已映射对象的数量来启用映射
5. 通过0x1400~1403(RPDO communication parameter)的Sub-index 01的有效(valid)控制位开启RPDO功能

0x1800~0x1803 Transmit PDO communication parameter

该对象包含CANopen设备有关TPDO通讯参数。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	2

sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Max sub-index	UNSIGNED8	RO	NO	NO	6
1	COB-ID used by TPDO	UNSIGNED32	RW	NO	NO	Index 1800h 0x180+\$NODE-ID Index 1801h 0x280+\$NODE-ID Index 1802h 0x380+\$NODE-ID Index 1803h 0x480+\$NODE-ID
2	Transmission type	UNSIGNED8	RW	NO	NO	255
3	Inhibit time	UNSIGNED16	RW	NO	NO	100
4	Compatibility entry	UNSIGNED8	RW	NO	NO	0
5	Event timer	UNSIGNED16	RW	NO	NO	0
6	SYNC start value	UNSIGNED8	RW	NO	NO	0

Max sub-index:

包含有效的子索引数量。

COB-ID:

Bit0-10: PDO的COB-ID, 如果发生变化需要将bit31位设置位0

Bit11-29: 采用11位COB-ID需要设置为0

Bit30: 0(1)PDO的RTR被允许(不被允许)

Bit31: 0(1)使用PDO通讯(不使用)

Transmission type:

Value(hex)	Description
00	Synchronous(acyclic)
01	Synchronous(cyclic every SYNC)
02	Synchronous(cyclic every 2 nd SYNC)
03	Synchronous(cyclic every 3 rd SYNC)
04	Synchronous(cyclic every 4 nd SYNC)
.....
F0	Synchronous(cyclic every 240 nd SYNC)
F1	Reserved
.....
FB	Reserved
FC	RTR-only(synchronous)
FD	RTR-only(even-driven)
FE	Event-driven(manufacturer-specific)
FF	Event-driven(device profile and application specfic)

Inhibit time:

Bit 0-15: TPDO最小间隔禁止时间, 单位ms, 0表示不用

Compatibility entry:

Bit 0-7: 不用。

Event timer:

Bit 0-15: TPDO传输事件时间, 单位ms。0表示不用

SYNC start value:

Value = 0: 同步信号计数器不做处理

Value = 1-240: SYNC信息计数器等于该值将会作为第一个接收到的SYNC消息处理。

0x1A00~0x1A03 Transmit PDO mapping parameter

包含设备可以传输的TPDO映射对象。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
RECORD	-	-	NO	NO	8

sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Number of mapped application objects in TPDO	UNSIGNED8	RW	NO	NO	1
1	1st application object	UNSIGNED32	RW	NO	NO	0x060400010
2	2st application object	UNSIGNED32	RW	NO	NO	0x00000000
3	3st application object	UNSIGNED32	RW	NO	NO	0x00000000
4	4th application object	UNSIGNED32	RW	NO	NO	0x00000000
5	5th application object	UNSIGNED32	RW	NO	NO	0x00000000
6	6th application object	UNSIGNED32	RW	NO	NO	0x00000000
7	7th application object	UNSIGNED32	RW	NO	NO	0x00000000
8	8th application object	UNSIGNED32	RW	NO	NO	0x00000000

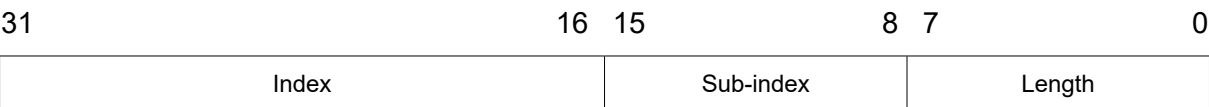
Number of mapped application objects in TPDO:

包含映射记录或特定记录中有效对象条目的数量。

Value	Description
00	Mapping disabled
01	Sub-index 01h valid
02	Sub-index 01h and 02h valid
03	Sub-index 01h to 03h valid
04	Sub-index 01h to 04h valid
05	Sub-index 01h to 05h valid
06	Sub-index 01h to 06h valid
07	Sub-index 01h to 07h valid
08	Sub-index 01h to 08h valid

Sub-index 01h~08h application object:

包含映射的应用对象信息。该对象的索引，子索引和数据长度通过该PDO来描述(如下图)。数据长度包含应用程序对象的长度(以bit为单位)，这可以用于验证映射。



Structure of PDO Mapping Entry

如果支持，应使用以下过程进行新的映射，该过程需要在“预运行(Pre-operational)”期间和“运行(Operational)”期间进行：

1.

通过0x1800~1803(TPDO communication parameter)的Sub-index 01的有效(valid)控制位关闭TPDO功能
2.

通过设置Sub-index 0h为0来关闭映射
3.

通过更改相应子索引的值来修改映射
4.

通过将子索引00h设置为已映射对象的数量来启用映射
5.

通过0x1800~1803(TPDO communication parameter)的Sub-index 01的有效(valid)控制位开启TPDO功能

7.2.2 运动控制配置文件6000h(CiA402)

Index	Sub	Name	Type	Access	Mapping
0x603F	-	Error code	UINT16	RO	YES
0x6040	-	Controlword	UINT16	WO	YES
0x6041	-	Statusword	UINT16	RO	YES
0x605A	-	Quick stop option code	INT16	RW	NO
0x6060	-	Modes of operation	INT8	WO	YES
0x6061	-	Modes of operation display	INT8	RO	YES
0x6064	-	Position actual value	INT32	RO	YES
0x6065	-	Following error window	UINT32	RW	NO
0x606C	-	Velocity actual value	INT32	RO	YES
0x6071	-	Target torque	INT16	RW	YES
0x6073	-	Max current	UINT16	RW	YES
0x6074	-	Torque demand	INT16	RO	YES
0x6078	-	Current actual value	INT16	RO	YES
0x607A	-	Target position	INT32	RW	YES
0x607C	-	Home offset	INT32	RW	YES
0x607E	-	Polarity	UINT8	RW	YES
0x607F	-	Max profile velocity	UINT32	RW	YES
0x6081	-	Profile velocity	UINT32	RW	YES
0x6083	-	Profile acceleration	UINT32	RW	YES
0x6084	-	Profile deceleration	UINT32	RW	YES
0x6085	-	Quick stop deceleration	UINT32	RW	YES
0x6087	-	Torque slope	UINT32	RW	YES
0x6098	-	Homing method	INT8	RW	YES
0x6099	2	Homing speed	ARRAY	-	NO
0x609A	-	Homing acceleration	UINT32	RW	YES
0x60F4	-	Following error actual value	INT32	RO	YES
0x60FD	-	Digital inputs	UINT32	RW	YES
0x60FE	2	Digital outputs	ARRAY	-	NO
0x60FF	-	Target velocity	INT32	RW	YES
0x6502	-	Supported drive modes	UINT32	RO	NO

0x603F Error code

该对象捕获保存驱动器最后发生的错误报警信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	YES	YES	0

在错误代码中，每一bit位包含一类报警或者错误状态信息。

BIT	StepSERVO Drive	Stepper Drive
0	位置超限 Position limit	
1	负方向超限 CCW limit	
2	正方向超限 CW limit	
3	过温 Over temp	
4	内部电压错误 Internal Voltage	
5	过压 Over Voltage	
6	欠压 Under Voltage	
7	过流 Over Current	
8	电机绕组开路 Open Motor Winding	
9	编码器错误 Bad Encoder	未使用 Not Used
10	通讯错误 Comm Error	
11	内存错误 Bad Flash	
12	没有运动 No Move	
13	重载 Current Foldback	未使用 Not Used
14	Q程序为空 Blank Q Segment	
15	未使用 Not Used	

0x6040 Controlword

该对象指明了控制PDS，FSA系统所需要接收到的指令。如下图所示，其中bit7、3、2、1和0必须支持。同时根据操作模式的不同，应支持bit0到9。如果相关功能不可实现，则应生成适当的紧急消息。可以支持制造商特定的位。所有控制字的bit功能实现都是有效独立于PDS FSA状态。任何运动的开始都是在指定的操作模式下进行，并在相关文档中进行了描述。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	WO	YES	NO	0

The bits of the controlword are defined as follows:

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																								
O		O		O	M	O		M	M	M	M																								
MSB												LSB																							
0						-						Optional						M						-						Mandatory					

0x6041 Statusword

该对象提供PDS FSA的状态信息，如下描述，其中bit10，9和6到0必须支持，oms位依据不同的操作模式生效。如果在oms位上相关功能不被允许，则对应的bit位响应为0。可以支持制造商特定的位显示。所有状态字的bit位实现状态显示都是有效独立于PDS FSA状态。

- The current state of the drive
- The operating state of the mode
- Manufacturer specific options

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	YES	YES	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms		oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso

Key:

ms	Manufacturer-specific
oms	Operation mode specific
ila	Internal limit active
tr	Target reached
rm	Remote
w	Warning
sod	Switch on disabled
qs	Quick stop
ve	Voltage enabled
f	Fault
oe	Operation enabled
so	Switched on
rtso	Ready to switch on

Statusword(6041h)	PDS FSA 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	Switch 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

0x605A Quick stop option code

该对象定义了当执行快速停止功能之后该执行什么样的动作。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RW	NO	NO	0

Quick stop option code	Action
-32768...-1	Manufacturer Specific
0	Disable drive function
1	Slow down on slow down ramp and transit into switch on disabled
2	Slow down on quick stop ramp and transit into switch on disabled
3	Slow down on the current limit and transit into switch on disabled
4	Slow down on the voltage limit and transit into switch on disabled
5	Slow down on slow down ramp and stay in quick stop active
6	Slow down on quick stop ramp and stay in quick stop active
7	Slow down on slow current limit and stay in quick stop active
8	Slow down on voltage limit and stay in quick stop active
9...32767	Reserved

这里只支持1和2的动作方式

0x6060 Mode of operation

该对象用来选择需要的操作模式。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT8	WO	YES	NO	0

Value	Action
-2	SYNC Q Mode(manufacturer specific)
-1	Normal Q Mode(manufacturer specific)
1	Profile Position Mode
3	Profile Velocity Mode
4	Torque Profile Mode(step servo only)
6	Homing Mode

-1: 常规Q程序模式 (manufacturer specific mode)

-2: 同步Q程序模式 (manufacturer specific mode)

鸣志CANopen驱动器目前不支持速度模式和内部位置模式，同时扭矩模式只在步进伺服上面支持。

0x6061 Mode of operation display

该对象显示当前的操作模式，反馈的值与对象0x6060h设置的一致

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT8	RO	YES	YES	0

0x6064 Position actual value

该对象反馈当前实际位置。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RO	YES	YES	0

0x6065 Following error window

该对象定义了位置误差允许的区间范围，如果当前位置超出了该对象定义的范围，将会产生跟随误差错误。单位为counts。

发生跟随误差错误的情况：

- 驱动器堵转
- 指定速度未达到
- 闭合系数错误

如果该对象的值设置为0或者0xFFFF FFFFh，表示关闭该功能。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

0x606C Velocity actual value

该对象提供电机当前实际的速度值。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RO	YES	YES	0

该对象单位为counts/s。

0x6071 Target torque

该对象参数用来输入在扭矩模式下的目标扭矩。写入的值以额度力矩的百分比处理。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RW	YES	NO	0

该对象只能允许在步进伺服下，并且该对象参数和其他扭矩值，如0x6078h和0x2216都有关联。

0x6073 Max current

该值表示电机中允许的最大电流。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	YES	NO	0

该对象单位为0.01Amps。

0x6074 Torque demand value

该对象是输出扭矩限制值

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RO	YES	NO	0

该对象只能用在步进伺服驱动器，单位为mNm。

0x6078 Current actual value

该对象当前实际值是指驱动电机中的瞬时电流。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT16	RO	YES	NO	0

该对象只能用在步进伺服驱动器，单位为0.01Amps。

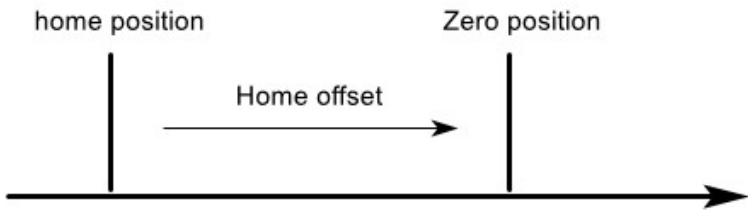
0x607A Target position

该对象是驱动器在当前运动控制参数设置生效，位置模式下将要移动的距离。就像速度，加速度，减速度一样。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RW	YES	NO	0

0x607C Home offset

该原点偏置对象是应用在零位置与机器原点位置(归位期间找到的位置)之间的差，以位置单位进行测量。在回原点期间找到机器的原点位置，一旦完成原点复位，则通过将原点偏移量添加到原点位置来将零位置从原点位置偏移。随后所有绝对移动均应相对于该新的零位置进行。下图对此进行了说明。



Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RW	YES	NO	0

0x607E Polarity

根据极性标志的值，位置给定值和位置实际值乘以1或-1。在PP PV模式下生效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RO	YES	NO	0

7	6	5	4	3	2	1	0
position polarity	velocity polarity	reserved					

MSB

LSB

Value	Description
0	multiply by 1
1	multiply by -1

0x607F Max profile speed

最大轮廓速度是轮廓运动期间任一方向上的最大允许速度。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

单位为counts/s。

0x6081 Profile velocity

轮廓速度是轮廓运动期间通常在加速完成结束时达到的速度，正反运动方向均有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

单位为counts/s。

0x6083 Profile acceleration

轮廓加速度单位为counts/s²。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

0x6084 Profile deceleration

轮廓减速度单位为counts/s²。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

0x6085 Quick stop deceleration

该对象是当驱动器执行快速停止指令之后电机停止用到的减速度，需要0x605A设置为2，单位为counts/s²

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

0x6087 Torque slope

该对象描述扭矩变化率。以0.1%额定转矩为单位变化

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

0x6098 Home method

该对象定义了回原点期间采用的回原点方式。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT8	RW	YES	NO	0

Value	Description
128	Manufacturer specific
0	No homing operation required
1...37	Methods 1 to 35
38-127	Reserved

0x6099 Homing speed

该对象定义了回原点期间的回原点速度。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	2

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Number of mapped entries	UNSIGNED8	RO	NO	NO	2
1	Speed during search for switch	UNSIGNED32	RW	YES	NO	0
2	Speed during search for zero	UNSIGNED32	RW	YES	NO	0

单位为counts/s²。

0x609A Homing acceleration

该对象定义了回原点加速度，在标准回原点模式下，该对象同时用在加速度和减速度下。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	YES	NO	0

单位为counts/s²。

0x60F4 Following error actual value

该对象提供跟随误差实际误差值，单位为counts。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RO	YES	NO	0

0x60FD Digital inputs

该对象提供数字输入状态，反馈逻辑输入的电平状态。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	YES	NO	0

31	24	23	22	21	20	19	18	17	16	15	4	3	2	1	0
Reserved	X8	X7	X6	X5	X4	X3	X2	X1	Reserved	Interlock	Home switch	Positive limit switch	Negative limit switch		
MSB											LSB				

Bit3(Interlock)提供互锁输入的状态。如果逻辑输入信号变为未激活，则驱动器应进入禁用或故障反应激活状态。这意味着驱动器的电源被禁用并被锁定以防打开。

0x60FE Digital outputs

该对象控制数字输出，该对象反馈逻辑输出电平状态。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	2

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	1...2	RO	NO	NO	2
1	Physical outputs	UNSIGNED32	RW	YES	NO	0
2	Bit mask	UNSIGNED32	RW	NO	NO	0

第一个子索引定义分配的输出。第二个子索引设定屏蔽，用于指定应使用哪个输出。

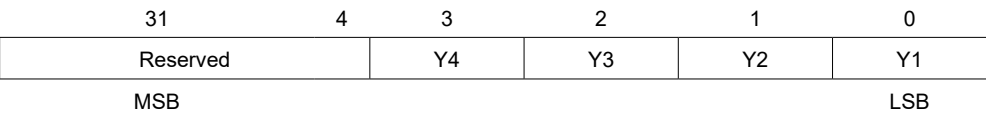
注意：第二个子索引是边沿触发的，必须先设置第二个子索引，然后再设置第一个子索引的位，才能更改输出状态

Physical outputs:

31	4	3	2	1	0
Reserved	Y4	Y3	Y2	Y1	
MSB			LSB		

Field	Value	Definition
Each bit	0	switch off
	1	switch on

Bit mask:



Field	Value	Definition
Each bit	0	Disable output
	1	Enable output

0x60FF Target velocity

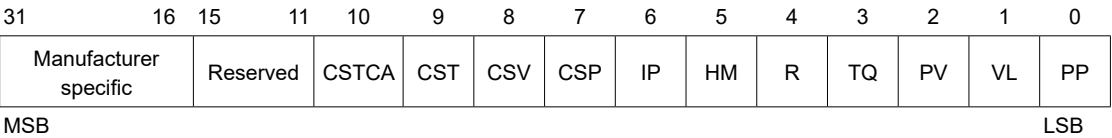
该对象用来配置目标速度，单位为counts/s。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	INT32	RW	YES	NO	0

0x6502 Supported drive modes

该对象提供驱动器支持的控制模式信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	0



鸣志CANopen驱动器支持的模式：

- Bit0: 轮廓位置模式
- Bit2: 轮廓速度模式
- Bit3: 扭矩模式 (StepSERVO only)
- Bit5: 原点模式

7.2.3 通用配置文件

Index	Sub	Name	Type	Access	Mapping
0x2001	-	Home switch	UINT8	RW	YES
0x2006	-	DSP clear alarm	UINT8	WO	NO
0x2007	-	Q segment NO.	UINT8	RW	YES
0x2009	-	Homing offset mode	UINT16	RW	NO
0x200B	-	DSP status code	UINT32	RO	YES
0x200C	-	Zero position	UINT8	WO	NO
0x200F	-	DSP alarm code	UINT32	RO	YES
0x2019	4	Device temperature	ARRAY	-	NO
0x2020	-	Node ID	UINT16	RW	NO
0x2021	-	Bit rate	UINT16	RW	NO
0x2030	-	DC bus voltage	UINT16	RO	NO
0x2031	-	DSP version	STRING(4)	RO	NO
0x2040	2	PDO1 transmit mask	ARRAY	-	NO
0x2041	2	PDO2 transmit mask	ARRAY	-	NO
0x2042	2	PDO3 transmit mask	ARRAY	-	NO
0x2043	2	PDO4 transmit mask	ARRAY	-	NO
0x2050	-	Product series	STRING(4)	RO	NO
0x2051	-	Customer name	STRING(4)	RO	NO
0x2060	5	Comm. watchdog	ARRAY	-	NO
0x2070	-	Switch value	UINT32	RO	NO
0x2100	23	User registers	ARRAY	-	NO

本章描述鸣志CANopen驱动器在自定义对象这块的说明。

0x2001 Home switch

该对象用来选择回原点的时候原点开关。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	YES	NO	3

Value	Home sensor
1	X1
2	X2
3	X3
4	X4
5	X5
6	X6
7	X7
8	X8

0x2006 DSP clear alarm

该对象提供一个清除驱动器报警的功能，设置该值为01h可以清除603f和200f的错误代码。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	WO	NO	COS	0

0x2007 Q segment NO.

该对象用来配置在Q模式下将要执行的Q程序段。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	YES	NO	0

0x2009 Homing offset mode

该对象用来设置原点偏置模式。

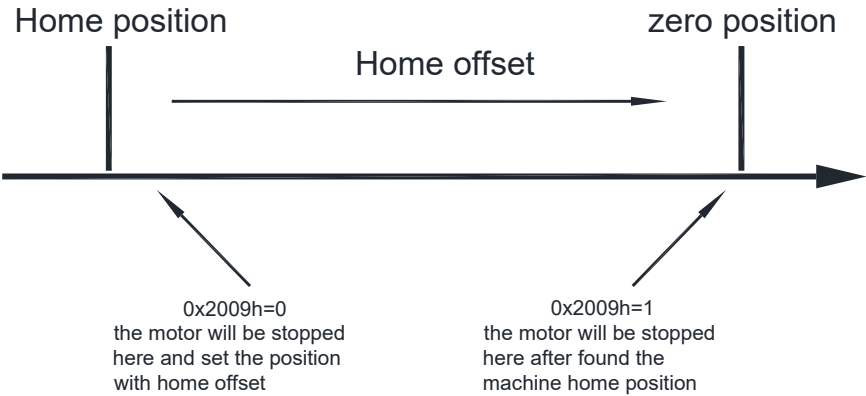
Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	0

设置为0:

电机会停止在机械原点位置，并且当前位置值由原点偏置(home offset)设定。

设置为1:

电机在找到机械原点位置之后移动一个由原点偏置(home offset)提供的距离，并将该位置设置为零点位置。



0x200B DSP status code

该对象反馈当前驱动器DSP状态信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	YES	YES	0

BIT	Status Code Bit Definition
0	驱动器使能, 如果未使能该位为0
1	Sampling - for Quick Tuner
2	驱动器错误-检查报警代码
3	在当前位置
4	移动中, 电机正在移动
5	Jog运行, 当前真正jog模式下
6	停止, 执行停止指令之后
7	等待, 执行一个输入等待指令之后
8	保存, 参数已被保存之后
9	警告, 检查报警代码
10	原点, 执行一个SH指令之后
11	等待时间, 执行一个WT指令之后
12	Wizard running - timing wizard is running
13	检测编码器
14	Q程序正在运行
15	初始化

0x200C Zero position

该对象提供设置所有位置参数为0的功能, 例如当前位置信息(0x6064).

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	WO	NO	NO	0

设置该对象值为01h可以清除所有位置信息。

0x200F DSP alarm code

该对象指明位于0x603F里面报警代码的高16位报警信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	WO	NO	NO	0

0x2019 Device temperature

该对象包含驱动器温度信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	NO	NO	4
1	Drive temperature	UNSIGNED16	RO	NO	NO	0
2	DSP temperature	UNSIGNED16	RO	NO	NO	0
3	Reserved 1	UNSIGNED16	RO	NO	NO	0
4	Reserved 2	UNSIGNED16	RO	NO	NO	0

该对象单位为0.1摄氏度。

0x2020 Node ID

该对象允许用户设置CAN通讯地址高4位数据，下次上电之后该变化生效，并且储存在对象0x1010h里面。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

Value	ID range
0	0x00-0x0F
1	0x10-0x1F
2	0x20-0x2F
3	0x30-0x3F
4	0x40-0x4F
5	0x50-0x5F
6	0x60-0x6F
7	0x70-0x7F

0x2021 Bit rate

该对象允许用户设置CAN通讯的波特率，下次上电之后该变化生效，并且储存在对象0x1010h里面。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

0x2030 DC bus voltage

该对象提供当前驱动器的母线电压值。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	0

最小单位为0.1v。

0x2031 DSP version

该对象包含驱动器的DSP版本号。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	STRING4	RO	NO	NO	0

0x2040...2043 PDOx transmit mask

这些符合COS(表示“更改数据”)传输事件的TPDO可以被该对象屏蔽。默认情况下, 检查到变化时会判断PDO中的所有数据。

每个索引都有2个子索引, 子索引1用于屏蔽(或显示)较低的32位(4字节)数据中的位。子索引2对于高32位是相同的。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	2

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	2
1	PDOx mask lowerBytes	UING32	RW	NO	NO	0xFFFFFFFF
2	PDOx mask upperBytes	UINT32	RW	NO	NO	0xFFFFFFFF

PDOx mask LowerBytes

Bit0-31:

当数据符合COS时, 该对象可以屏蔽TPDO的低四字节数据

PDOx mask upperBytes

Bit0-31: 当数据符合COS时, 该对象可以屏蔽TPDO的高四字节数据

0x2050 Product series

该对象包含产品系列名称。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	STRING4	RO	NO	NO	--

0x2051 Customer name

该对象包含定制驱动器名称。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	STRING4	RO	NO	NO	--

0x2060 Comm. watchdog

该对象可以在轮廓速度模式下执行一个安全功能，看门狗功能只能在特定的RPDO信息或同步信息 (SYNC为0x80)下被复位，如果在看门狗循环触发时间内没有收到特定的信息，例如线路中断的情况下，看门狗将被触发，电机在0x6085设置的减速度下减速到0停止。

看门狗通过以下方式配置：

- 看门狗配置为使能或者非使能
- 看门狗循环时间
- 特定的RPDO或SYNC消息配置。RPDO1、2、3、4和SYNC可以选择监视信息来触发看门狗。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	CONST	NO	NO	4
1	Watchdog enable	UINT8	RW	NO	NO	0
2	Watchdog status	UINT8	RW	NO	NO	0
3	Watchdog timeout	UINT16	RW	NO	NO	0x01F4
4	Watchdog trigger event	UINT8	RW	NO	NO	0x0F
5	Timeout option code	UINT16	RW	NO	NO	0

Watchdog enable:

- 值 = 0 关闭看门狗功能
 = 1 激活看门狗功能

该对象只能在操作模式下(operation mode)设置

Watchdog status 2:

- 值 = 0 正常状态
 = 1 看门狗被触发
 = 2 看门狗已被激活，摒弃监视CAN-bus总线消息，但是没有被触发

Watchdog timeout:

设置看门狗循环时间，该对象单位为ms。

Watchdog trigger event:

该对象用来选择循环触发检测对象

- 值 = 0x01 RPDO1
 = 0x02 RPDO2
 = 0x04 RPDO3
 = 0x08 RPDO4
 = 0x10 SYNC message

Timeout option code:

- 值 = 0x0 停止运动并去使能
 = 0x01 停止运动并使能
 = 0x2-0x0Eh 执行相应的Q程序

下文有关于通讯看门狗的应用方法。

0x2070 Switch value

该对象包含旋钮开关的信息，例如地址和波特率开关设置信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RO	NO	NO	3

数据低16位是保存的设置信息。

0x2100 User registers

该对象提供了23个普通用户寄存器，由于属于非保持性，所以里面的信息在重新上电之后不会保存。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	23

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UNSIGNED8	RO	YES	NO	23
1	User register1	UNSIGNED32	RW	YES	NO	0
2	User register2	UNSIGNED32	RW	YES	NO	0
3	User register3	UNSIGNED32	RW	YES	NO	0
4	User register4	UNSIGNED32	RW	YES	NO	0
5	User register5	UNSIGNED32	RW	YES	NO	0
6	User register6	UNSIGNED32	RW	YES	NO	0
7	User register7	UNSIGNED32	RW	YES	NO	0
8	User register8	UNSIGNED32	RW	YES	NO	0
9	User register9	UNSIGNED32	RW	YES	NO	0
10	User register10	UNSIGNED32	RW	YES	NO	0
11	User register11	UNSIGNED32	RW	YES	NO	0
12	User register12	UNSIGNED32	RW	YES	NO	0
13	User register13	UNSIGNED32	RW	YES	NO	0
14	User register14	UNSIGNED32	RW	YES	NO	0
15	User register15	UNSIGNED32	RW	YES	NO	0
16	User register16	UNSIGNED32	RW	YES	NO	0
17	User register17	UNSIGNED32	RW	YES	NO	0
18	User register18	UNSIGNED32	RW	NO	NO	0
19	User register19	UNSIGNED32	RW	NO	NO	0
20	User register20	UNSIGNED32	RW	NO	NO	0
21	User register21	UNSIGNED32	RW	NO	NO	0
22	User register22	UNSIGNED32	RW	NO	NO	0
23	User register23	UNSIGNED32	RW	NO	NO	0

7.2.4 步进伺服通用配置文件

Index	Sub	Name	Type	Access	Mapping
0x2201	-	Peak current	UINT16	RW	YES
0x2202	-	Torque limit of hard stop homing	UINT16	RW	NO
0x2203	-	Idle current	UINT16	RW	NO
0x2204	-	In position error range	UINT16	RW	NO
0x2205	-	In position time	UINT16	RW	NO
0x2206	-	Acceleration current	UINT16	RW	NO
0x2207	-	Steps per revolution	UINT16	RO	NO
0x2208	-	Rotation reverse	UINT8	RW	NO
0x2209	-	Velocity max	UINT32	RW	NO
0x220A	-	Acceleration max	UINT32	RW	NO
0x220B	-	Smooth filter	UINT16	RW	NO
0x220C	3	Brake configuration	UINT8	RO	NO
0x220D	-	Servo enable	UINT16	RW	NO
0x220E	-	Alarm reset input	UINT16	RW	NO
0x220F	-	Define limits	UINT16	RW	NO
0x2210	8	Digital inputs filter	ARRAY	-	NO
0x2211	8	Notch filter	ARRAY	-	NO
0x2212	4	Analog configuration	ARRAY	-	NO
0x2213	-	Alarm output	UINT16	RW	NO
0x2214	-	Move output	UINT16	RW	NO
0x2215	-	Jog mode	UINT16	RW	NO
0x2216	-	Torque constant	UINT16	RO	NO
0x2217	-	Misc flag	UINT16	RW	NO
0x2218	-	Encoder resolution	UINT16	RO	NO
0x221A	-	Alarm mask	UINT16	RW	NO
0x221C	4	Analog input	ARRAY	-	NO
0x2220	-	Position gain	UINT16	RW	NO
0x2221	-	Position derigain	UINT16	RW	NO
0x2222	-	Positioin derifilter	UINT16	RW	NO
0x2224	-	Velocity gain	UINT16	RW	NO
0x2225	-	Velocity inteergain	UINT16	RW	NO
0x2226	-	Acceleration feedforward	UINT16	RW	NO
0x2227	-	PID filter	UINT16	RW	NO
0x2252	-	In position counts	UINT16	RW	NO

0x2201 Peak current

该对象是步进伺服的峰值(RMS)电流。峰值电流设置了给定电机应使用的最大电流。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	YES	NO	-

该对象单位为mA。

0x2202 Torque limit of hard stop homing

该对象用来设置硬限位回原点模式下，当电机撞击硬限位位置的时候限制的电流。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2203 Idle current

该对象配置电机在空闲状态下的空闲电流，单位为0.01Amps。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2204 In position error range

该对象用来设置静态位置误差区间，当实际位置与目标位置在误差区间内持续一段时间，时间由0x2205定义，即认定电机动作完成或者已在当前位置。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2205 In position time

该对象用来设置在设定范围内的检测时间，当0x2204对象定义之后，该参数将配置在当前位置状态检测的时间依据。时间是循环处理判断的，一个循环为200us。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

如果读取的值使100表示20ms。

0x2206 Acceleration current

该对象用来设置当驱动器在步进模式下运行的加速度电流

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

如果读取的值为100表示电流为1Amps。

0x2207 Steps per revolution

该对象用来获取电机每转的步数。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	-

该对象单位为steps/rev

0x2208 Rotation reverse

该对象可以用来改变电机默认方向。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT8	RW	NO	NO	0

该值可以被设置为0和1。

值 = 0 默认旋转方向

 = 1 使电机反向

0x2209 Velocity max

该对象用来设置驱动器的最大速度，同样在模拟量速度模式下，该值可以限制驱动器最大速度

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

该对象单位为counts/s.

0x220A Acceleration max

该对象用来设置最大加减速

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

该对象单位为counts/s²。

0x220B Smooth filter

该对象提供一个滤波去改变电机在位置模式下运动的平滑状态。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	2500

请注意一个较低的滤波参数可能会使运行更平滑，但系统响应会变得更延迟

0x220C Brake configuration

该对象用来设置刹车输出功能相关参数，该对象有3个子索引，第一个用作配置刹车输出，第二个配置刹车释放之后移动等待时间，最后一个配置刹车抱闸之后去使能等待时间。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	3

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	3
1	刹车输出	UINT16	RW	NO	NO	0
2	刹车释放之后移动等待时间	UINT16	RW	NO	NO	0
3	刹车抱闸之后去使能等待时间	UINT16	RW	NO	NO	0

刹车输出:

- 值 = 1 驱动器使能，输出闭合。驱动器去使能，输出打开
 = 2 驱动器使能，输出打开。驱动器去使能，输出闭合
 = 3 不作为刹车输出功能，可以用作普通输出口

刹车释放/刹车抱闸动作等待时间:

单位为ms，如过写100，意思为延迟0.1s。

0x220D Servo enable

该对象用来设置使能输入口的功能，驱动器将X5默认为使能输入功能。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

- 值 = 1 当X5打开的时候驱动器使能
 = 2 当X5闭合的时候驱动器使能
 = 3 将X5配置为普通输入口

0x220E Alarm reset input

该对象用来设置报警复位输入口的功能，驱动器X6默认定义为报警复位功能

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

- 值 = 1 当X6打开的时候报警复位
 = 2 当X6闭合的时候报警复位
 = 3 将X6配置为普通输入口

0x220F Define limits

该对象用来设置限位功能定义。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

Value	Definition
1	输入口闭合，限位功能生效
2	输入口打开，限位功能生效
3	不做限位功能，用作普通输入口
4...20	详情参考"host command reference"手册

0x2210 Input filter

该对象在IO输入口设置一个数字滤波。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	3

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Filter input 1	UINT16	RW	NO	NO	0
2	Filter input 2	UINT16	RW	NO	NO	0
3	Filter input 3	UINT16	RW	NO	NO	0
4	Filter input 4	UINT16	RW	NO	NO	0
5	Filter input 5	UINT16	RW	NO	NO	0
6	Filter input 6	UINT16	RW	NO	NO	0
7	Filter input 7	UINT16	RW	NO	NO	0
8	Filter input 8	UINT16	RW	NO	NO	0

该参数单位为200us。如果该对象值为100表示20ms滤波延迟

0x2211 Notch filter

这8个对象用来配置陷波滤波器，该对象只在步进伺服驱动器有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	8

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Notch filter A tuning	UINT16	RW	NO	NO	-
2	Notch filter B tuning	UINT16	RW	NO	NO	-
3	Notch filter C tuning	UINT16	RW	NO	NO	-
4	Notch filter D tuning	UINT16	RW	NO	NO	-
5	Notch filter E tuning	UINT16	RW	NO	NO	-
6	Notch filter F tuning	UINT16	RW	NO	NO	-
7	Notch filter G tuning	UINT16	RW	NO	NO	-
8	Notch filter H tuning	UINT16	RW	NO	NO	-

0x2212 Analog configuration

该对象用来配置关于模拟量的运行模式，当运行模拟量速度/位置/扭矩的时候，我们需要设置相关参数到该对象里面。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	4
1	模拟量类型	UINT16	RW	NO	NO	0
2	模拟量死区	UINT16	RW	NO	NO	0
3	模拟量偏置	UINT16	RW	NO	NO	0
4	模拟量滤波	UINT16	RW	NO	NO	0

模拟量类型:

- 值 = 0 单端信号 +/- 10 volts
- = 1 单端信号 0 - 10 volts
- = 2 单端信号 +/- 5 volts
- = 3 单端信号 0 - 5 volts
- = 4 差分信号 +/- 10 volts
- = 5 差分信号 0 - 10 volts
- = 6 差分信号 +/- 5 volts
- = 7 差分信号 0 - 5 volts

模拟量死区

单位为0.001V

模拟量偏置

单位为0.001V

模拟量滤波

滤波值 = $72090 / [(1400 / x) + 2.2]$

x 表示 模拟量滤波期望的值，单位为 Hz

0x2213 Alarm output

该对象用来设置当驱动器发生错误的时候报警输出口功能，驱动器Y1端口默认为报警输出口功能

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

- 值 = 1 当驱动器发生错误的时候打开
- = 2 当驱动器发生错误的时候闭合
- = 3 将Y1设置为普通输出口

0x2214 Motion output

该对象用来定义电机运动输出口的功能。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	3

SSDC系列有4个输出端口，他们可以被设置为各种功能。

Output	Y4				Y3				Y2				Y1			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1

举例：

Y4的值为0011B表示Y4被配置为普通输出口功能。

- 值 = 1 电机在移动的时候断开
 = 2 电机在移动的时候闭合
 = 3 普通输出口 (错误报警输出/刹车输出)
 = 4 Tach output 功能, 100 pulses/rev
 = 5 Tach output 功能, 200 pulses/rev
 = 6 Tach output 功能, 400 pulses/rev
 = 7 Tach output 功能, 800 pulses/rev
 = 8 Tach output 功能, 1600 pulses/rev
 = 9 预留
 = 10 预留
 = 11 Timing out (50 pulses/rev)

0x2215 Jog mode

该对象用来设置速度模式，鸣志驱动器有两种速度模式。

- 值 = 1 位置模式下的速度控制
 = 2 纯速度模式下的速度控制

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

0x2216 Torque constant

该对象配置电机的扭矩常数，单位为Nm/Amps。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	0

该对象只支持步进伺服驱动器。

0x2217 Misc flag

该对象用来提供驱动器其他状态和错误信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT	RW	NO	NO	0

该对象的值是一个包含各种驱动器状态的16进制值，写0可以清除当前值。

描述	Hex Value	Decimal Value
位置限位标志	0x0001	1
传感器发现标志位	0x0002	2
低端过流	0x0004	4
高端过流	0x0008	8
读数错误	0x0010	16
A相电流偏置错误	0x0020	32
B相电流偏置错误	0x0040	64
A相开环	0x0080	128
B相开环	0x0100	256
逻辑电路供电错误	0x0200	512
门限电压供电错误	0x0400	1024
内存擦除错误	0x4000	16384
内存保存错误	0x8000	32768

0x2218 Encoder resolution

该对象用来提供电机编码器的配置信息，它包含电机每转反馈的脉冲数

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RO	NO	NO	-

0x221A Alarm mask

该对象用来屏蔽LED灯报警信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

详细说明:

bit	15	14	13	12	11	10		6	5		2	
		Q程序 为空	重载	未使能 先动	保存失败	通讯错 误		欠压	负限位 报警		正限位 报警	

0x221C Analog input

该对象提供模拟量输入的值。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	4

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	4
1	Analog input 1	UINT16	RO	YES	YES	-
2	Analog input 2	UINT16	RO	YES	YES	-
3	Analog input 3	UINT16	RO	YES	YES	-
4	Analog input 4	UINT16	RO	YES	YES	-

0x2220 Position gain

该对象用来配置步进伺服在位置环下的比例增益，该对象只在步进伺服下有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2221 Position derigain

该对象用来配置步进伺服在位置环控制下的微分增益，该对象只在步进伺服下有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2222 Position derifilter

该对象提供一个非常简单的单极点低通滤波器用来限制高频噪音，以及使系统变得更安静和稳定。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2224 Velocity gain

该对象用来配置步进伺服在速度环控制中的比例增益，该对象只在步进伺服下有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2225 Velocity integergain

该对象用来配置步进伺服，在速度环下的积分增益，该对象只在步进伺服下有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2226 Acceleration feed forward

该对象用来配置一个加速度/减速度前馈增益，用来加快系统在扭矩指令下的响应速度，该对象只在步进伺服下有效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2227 PID filter

此对象在速度环的末尾提供一个扭矩指令过滤器。该滤波器是一个非常简单的单极点低通滤波器，用于限制速度的高频响应，从而限制位置控制环路。该对象仅在步进伺服驱动器上生效。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

0x2252 In position counts (position limit)

该对象用来设置静态位置误差值，用来作为步进伺服运动状态的判断依据。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

7.2.5 步进通用配置文件

Index	Sub	Name	Type	Access	mapping
0x2601		Idle current	UINT8	RW	NO
0x2603		Acceleration current	UINT16	RW	NO
0x2604		Steps per revolution	UINT16	RW	NO
0x2607		Brake configuration	ARRAY	-	NO
0x2609		Alarm reset input	UINT16	RW	NO
0x260A		Jerk delay time	UINT16	RW	NO
0x260B		Define limits	UINT16	RW	NO
0x260C		Input filter	ARRAY	-	NO
0x260D		Motor selection	UINT16	RW	NO
0x260E		Open winding detect velocity limit	UINT32	RW	NO
0x260F		Drive configuration	ARRAY	-	NO
0x2610		Alarm mask	UINT16	RW	NO
0x2611		Load ratio	UINT16	RW	NO
0x2612		Custom motor	ARRAY	-	NO
0x2614		Smooth filter	UINT16	RW	NO
0x2615		Motion output	UINT16	RW	NO
0x2616		Alarm output	UINT16	RW	NO
0x2617		Servo enable	UINT16	RW	NO

0x2601 Idle current

该对象配置/监控电机在空闲状态下的电流大小。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

单位为0.01Amps。

0x2603 Acceleration current

该对象配置驱动器的加速电流。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

它的单位与最大电流和空闲电流设置一样。

0x2604 Steps per revolution

该对象指示电机每转的步数，它允许你改变当前驱动器每转一圈反馈的步数。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

该值单位为steps/rev。

0x2607 Brake configuration

该对象用来设置刹车输出参数配置，参考0x220C。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	3

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	3
1	刹车输出	UINT16	RW	NO	NO	0
2	刹车释放之后移动等待时间	UINT16	RW	NO	NO	0
3	刹车抱闸之后去使能等待时间	UINT16	RW	NO	NO	0

刹车输出:

- 值 = 1 驱动器使能，输出闭合。驱动器去使能，输出打开
 = 2 驱动器使能，输出打开。驱动器去使能，输出闭合
 = 3 不作为刹车输出功能，可以用作普通输出口

刹车释放/抱闸动作等待时间:

单位为ms，如写入100，意思为延迟0.1s。

0x2609 Alarm reset input

该对象用来设置报警复位输入功能，参考对象0x220E设置。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

- 值 = 1 当X6打开的时候清除报警
 = 2 当X6闭合的时候清除报警
 = 3 将X6设置为普通输入口

0x260A Jerk delay time

该对象用来提供设置加加速度延迟时间。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	3

该值单位为ms。

0x260B Define limits

该对象用来定义限位功能。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	-

Value	Definition
1	输入口闭合, 限位功能生效
2	输入口打开, 限位功能生效
3	不做限位功能, 作为普通输入口
4...20	详情参考“host command reference”手册

0x260C Input filter

该对象用来设置输入口数字滤波。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RO	NO	NO	8

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	Filter input 1	UINT16	RW	NO	NO	60
2	Filter input 2	UINT16	RW	NO	NO	0
3	Filter input 3	UINT16	RW	NO	NO	0
4	Filter input 4	UINT16	RW	NO	NO	0
5	Filter input 5	UINT16	RW	NO	NO	0
6	Filter input 6	UINT16	RW	NO	NO	0
7	Filter input 7	UINT16	RW	NO	NO	0
8	Filter input 8	UINT16	RW	NO	NO	0

该参数单位为200us。如果该参数值为100, 意思为20ms延迟。

0x260D Motor selection

该对象用来设置驱动器匹配的电机。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

Details:

Value	Selection	Value	Selection
0	AM8HY2050	14	AM23HS2450
1	AM8HY4043	15	AM23HS3454
2	AM11HS008	16	AM23HS3455
3	AM11HS3007	17	AM23HS5412
4	AM11HS5008	18	AM23HS84A0
5	AM14HYB401	19	AM23HS84B0
6	AM17HD2438	20	AM23HSA4A0
7	AM17HD4452	21	AM23HSA4B0
8	AM17HDB410	22	AM24HS2402
9	AM23HS0420	23	AM24HS5401
10	AM23HS0421	24	AM34HD0404
11	AM23HS04A0	25	AM34HD1404
12	AM23HS04B0	26	AM34HD2403
13	AM23HS2449	27	AM34HD3402

0x260E Open winding detect velocity limit

该对象用来设置当电机在运动的时候，电机断线检测功能生效的最低速度限值。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT32	RW	NO	NO	-

该对象单位为counts/s。

0x260F Drive configuration

该对象用来设置驱动器相关配置参数。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UINT8	RW	NO	NO	8

Sub-index	Name	Data Type	Access Type	PDO Mapping	COS	Default value
0	Highest sub-index supported	UINT8	RO	NO	NO	8
1	电机反向	UINT16	RW	NO	NO	0
2	平滑滤波	UINT16	RW	NO	NO	0
3	电子阻尼	UINT16	RW	NO	NO	0
4	断线检测	UINT16	RW	NO	NO	0
5	预留	UINT16	RW	NO	NO	0
6	预留	UINT16	RW	NO	NO	0
7	预留	UINT16	RW	NO	NO	0
8	预留	UINT16	RW	NO	NO	0

Description of sub-index:

Sub-index	Name	Description
1	电机反向	0: 保持当前运动方向 1: 改变当前电机运动方向
2	平滑滤波	0: 开启平滑滤波 1: 关闭平滑滤波
3	电子阻尼	0: 开启电子阻尼抗震功能 1: 关闭电子阻尼抗震功能
4	断线检测	0: 关闭电机断线检测功能 1: 开启电机断线检测功能

0x2610 Alarm mask

该对象用来设置屏蔽LED灯报警的信息，参照对象0x221A设置。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	-

详细说明:

bit	15	14	13	12	11	10		6	5		2	
		Q程序 为空	重载	未使能 先动	保存失败	通讯错 误		欠压	负限位 报警		正限位 报警	

0x2611 Load ratio

该对象设置负载惯量比。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UINT16	RW	NO	NO	1

Value = 1 1倍转子惯量
 = 2 1-3倍转子惯量
 = 3 3-5倍转子惯量
 = 4 5-7倍转子惯量
 = 5 7-10倍转子惯量

0x2612 Custom motor

该对象包含电机的定制信息。

Object Type	Data Type	Access Type	PDO mapping	COS	Sub Number
ARRAY	UNSIGNED32	RW	NO	NO	22

Sub index	Name	Data Type	Access Type	Default Value	PDO Mapping
0	Highest sub-index supported	UNSIGNED8	R0	0	NO
1	Motor name 1	UNSIGNED32	RW	0	NO
2	Motor name 2	UNSIGNED32	RW	0	NO
3	Motor name 3	UNSIGNED32	RW	0	NO
4	Motor name 4	UNSIGNED32	RW	0	NO
5	Motor parameter 1	UNSIGNED16	RW	0	NO
6	Motor parameter 2	UNSIGNED16	RW	0	NO
7	Motor parameter 3	UNSIGNED16	RW	0	NO
8	Motor parameter 4	UNSIGNED16	RW	0	NO
9	Motor parameter 5	UNSIGNED16	RW	0	NO
A	Motor parameter 6	UNSIGNED16	RW	0	NO
B	Motor parameter 7	UNSIGNED16	RW	0	NO
C	Motor parameter 8	UNSIGNED16	RW	0	NO
D	Motor parameter 9	UNSIGNED16	RW	0	NO
E	Motor parameter 10	UNSIGNED16	RW	0	NO
F	Motor parameter 11	UNSIGNED16	RW	0	NO
10	Motor parameter 12	UNSIGNED16	RW	0	NO
11	Reserved 1	UNSIGNED16	RW	0	NO
12	Reserved 2	UNSIGNED16	RW	0	NO
13	Reserved 3	UNSIGNED16	RW	0	NO
14	Reserved 4	UNSIGNED16	RW	0	NO
15	Reserved 5	UNSIGNED32	RW	0	NO

0x2614 Smooth filter

该对象提供一个滤波参数去改变在位置指令下的平滑效果，参照对象0x220B设置方法。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	2500

0x2615 Motion output

用法同对象0x2214。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	0

STF系列有4个数字输出口，他们可以设置为各种功能。

Output	Y4				Y3				Y2				Y1			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

举例：

当Y4的值为0011B，意思为Y4输出口定义为普通输出口。

- 值 = 1 电机在移动的时候断开
- = 2 电机在移动的时候闭合
- = 3 普通输出口 (错误报警输出/刹车输出)
- = 4 Tach output 功能, 100 pulses/rev
- = 5 Tach output 功能, 200 pulses/rev
- = 6 Tach output 功能, 400 pulses/rev
- = 7 Tach output 功能, 800 pulses/rev
- = 8 Tach output 功能, 1600 pulses/rev
- = 9 预留
- = 10 预留
- = 11 Timing out (50 pulses/rev)

0x2616 Alarm output

该对象用来设置当驱动器发生错误的时候报警输出端口的功能。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	3

- 值 = 1 当驱动器出现错误的时候Y1输出断开
- = 2 当驱动器出现错误的时候Y1输出闭合
- = 3 Y1设置为普通输入口

0x2617 Servo enable

该对象用来设置使能输入口的使用方法。用法同对象0x220D。

Object Type	Data Type	Access Type	PDO mapping	COS	Default Value
VAR	UNSIGNED16	RW	NO	NO	3

详细说明：

- 值 = 1 X5低电平的时候驱动器使能
- = 2 X5高电平的时候驱动器使能
- = 3 X5作为普通输入口

8 附录A – SDO消息控制举例

轮廓位置模式

使能电机

ID(Hex)	DLC	Data(Hex)	Description
603	8	2B 40 60 00 06 00 00 00	‘准备好启动
603	8	2B 40 60 00 07 00 00 00	‘启动驱动
603	8	2B 40 60 00 0F 00 00 00	‘激活操作模式

设置运动模式

603	8	2F 60 60 00 01 00 00 00	‘设置为轮廓位置模式
-----	---	-------------------------	------------

设置运动轨迹参数

603	8	23 81 60 00 20 4E 00 00	‘设置速度为20000 step/s
603	8	23 83 60 00 40 9C 00 00	‘设置加速度为40000 step/s ²
603	8	23 84 60 00 40 9C 00 00	‘设置减速度为40000 step/s ²

绝对运动

603	8	23 7A 60 00 40 0D 03 00	‘设置目标位置为200000 steps
603	8	2B 40 60 00 1F 00 00 00	‘设置New Set Point位为1(电机开始绝对运动)
603	8	2B 40 60 00 0F 00 00 00	‘清除New Set Point

相对运动

603	8	23 7A 60 00 40 0D 03 00	‘设置目标位置为200000 steps
603	8	2B 40 60 00 5F 00 00 00	‘设置New Set Point位为1(电机开始相对运动)
603	8	2B 40 60 00 4F 00 00 00	‘清除New Set Point

多段移动，完成当前轨迹运动，并停止之后，执行下一段运动

603	8	23 81 60 00 40 9C 00 00	‘设置速度为40000 step/s
603	8	23 7A 60 00 40 0D 03 00	‘设置目标位置为200000 steps
603	8	2B 40 60 00 5F 00 00 00	‘设置New Set Point 位为1
603	8	2B 40 60 00 4F 00 00 00	‘清除New Set Point
603	8	23 81 60 00 80 38 01 00	‘设置速度为80000 step/s
603	8	23 7A 60 00 C0 27 09 00	‘设置目标位置为600000 steps
603	8	2B 40 60 00 5F 00 00 00	‘设置New Set Point 位为1
603	8	2B 40 60 00 4F 00 00 00	‘清除New Set Point

多段移动，完成当前轨迹运动之后，立刻按照新的参数运动

603	8	23 81 60 00 40 9C 00 00	‘设置速度为40000 step/s
603	8	23 7A 60 00 40 0D 03 00	‘设置目标位置为200000 steps
603	8	2B 40 60 00 5F 02 00 00	‘设置New Set Point为1
603	8	2B 40 60 00 4F 02 00 00	‘清除New Set Point
603	8	23 81 60 00 80 38 01 00	‘设置速度为80000 step/s
603	8	23 7A 60 00 C0 27 09 00	‘设置目标位置为600000 steps
603	8	2B 40 60 00 5F 02 00 00	‘设置New Set Point为1
603	8	2B 40 60 00 4F 02 00 00	‘清除New Set Point

多段移动，按照新的轨迹参数立刻改变运动

603	8	23 81 60 00 40 9C 00 00	‘设置速度为40000 step/s
603	8	23 7A 60 00 40 0D 03 00	‘设置目标位置为200000 steps
603	8	2B 40 60 00 7F 02 00 00	‘设置New Set Point为1
603	8	2B 40 60 00 6F 02 00 00	‘清除New Set Point
603	8	23 81 60 00 80 38 01 00	‘设置速度为80000 step/s
603	8	23 7A 60 00 C0 27 09 00	‘设置目标位置为600000 steps
603	8	2B 40 60 00 7F 02 00 00	‘设置New Set Point为1
603	8	2B 40 60 00 6F 02 00 00	‘清除New Set Point

轮廓速度模式

使能电机

ID(Hex)	DLC	数据(Hex)	描述
603	8	2B 40 60 00 06 00 00 00	‘准备好启动
603	8	2B 40 60 00 07 00 00 00	‘启动驱动器
603	8	2B 40 60 00 0F 01 00 00	‘激活操作模式，并开启halt

设置运动模式

603	8	2F 60 60 00 03 00 00 00	‘设置为轮廓速度模式
-----	---	-------------------------	------------

设置运动参数

603	8	23 FF 60 00 20 4E 00 00	‘设置速度为20000 step/s
603	8	23 83 60 00 40 9C 00 00	‘设置加速度为40000 step/s ²
603	8	23 84 60 00 40 9C 00 00	‘设置减速度为40000 step/s ²

启动/停止运动

603	8	2B 40 60 00 0F 00 00 00	‘电机启动
603	8	23 FF 60 00 80 38 01 00	‘改变目标速度为80000 step/s
603	8	2B 40 60 00 0F 01 00 00	‘电机停止

回原点模式

使能电机

ID(Hex)	DLC	数据(Hex)	描述
603	8	2B 40 60 00 06 00 00 00	‘准备好启动
603	8	2B 40 60 00 07 00 00 00	‘启动驱动器
603	8	2B 40 60 00 0F 00 00 00	‘激活操作模式

设置运动模式

603	8	2F 60 60 00 06 00 00 00	‘设置为回原点模式
603	8	2F 98 60 00 13 00 00 00	‘设置回原点为19号方式

设置运动参数

603	8	23 9A 60 00 50 C3 00 00	‘设置回原点加速度为50000 step/s ²
603	8	23 99 60 01 10 27 00 00	‘设置回原点速度 (寻找限位开关信号) 为10000 step/s
603	8	23 99 60 02 88 13 00 00	‘设置找Z相信号速度为5000 step/s
603	8	23 7C 60 00 40 9C 00 00	‘设置原点offset为40000 Steps
603	8	2F 01 20 00 03 00 00 00	‘设置原点信号为x3

启动/停止回原点运动

603	8	2B 40 60 00 1F 00 00 00	‘开始启动回原点运动
603	8	2B 40 60 00 1F 01 00 00	‘停止回原点运动

常规Q程序模式

使能电机

ID(Hex)	DLC	数据(Hex)	描述
603	8	2B 40 60 00 06 00 00 00	‘准备好启动
603	8	2B 40 60 00 07 00 00 00	‘启动驱动器
603	8	2B 40 60 00 0F 00 00 00	‘激活操作模式

设置运动模式

603	8	2F 60 60 00 FF 00 00 00	‘设置常规Q程序模式
603	8	2F 07 20 00 01 00 00 00	‘设置启动Q程序段

启动/停止Q程序模式

603	8	2B 40 60 00 1F 00 00 00	‘Q程序启动
603	8	2B 40 60 00 1F 01 00 00	‘Q程序停止

同步Q模式

使能电机

ID(Hex)	DLC	数据(Hex)	描述
603	8	2B 40 60 00 06 00 00 00	‘准备好启动
603	8	2B 40 60 00 07 00 00 00	‘启动驱动器
603	8	2B 40 60 00 0F 00 00 00	‘激活操作模式

设置同步Q程序模式

603	8	2F 60 60 00 FE 00 00 00	‘设置同步Q程序模式
603	8	2F 07 20 00 01 00 00 00	‘设置Q程序段
603	8	23 05 10 00 80 00 00 00	‘设置同步触发信号

启动/停止Q程序模式

80	0		Q程序启动
603	8	2B 40 60 00 0F 01 00 00	‘Q程序停止

PDO映射

映射到TPDO2

ID(Hex)	DLC	数据(Hex)	描述
000	2	80 03	返回Pre-optation模式
603	8	23 01 18 01 80 02 00 80	‘关闭TPDO2
603	8	2F 01 1A 00 00 00 00 00	‘设置映射对象的数量为0
603	8	23 01 1A 01 10 00 41 61	‘映射对象0x6041到TPDO2的子索引1里面.
603	8	23 01 1A 02 20 00 0A 70	‘映射对象0x700A到TPDO2的子索引2里面.
603	8	2F 01 1A 00 02 00 00 00	‘设置映射对象总数量为2
603	8	23 01 18 01 80 02 00 00	‘开启TPDO2

9 联系 MOONS'

客户咨询中心



400-820-9661

■ 鸣志总部

上海市闵行区闵北工业区鸣嘉路168号
邮编: 201107

■ 国内办事处

深圳

深圳市南山区桃源街道信宜五路13号塘朗工业B区集悦城
众创产业园53栋401
邮编: 518000

北京

北京市朝阳区东三环中路16号京粮大厦1206室
邮编: 100022

南京

南京市江宁区天元中路126号新城发展中心2号楼11楼
1101/1102室
邮编: 211106

青岛

青岛市市北区凤城路16号卓越大厦1710室
邮编: 266000

武汉

武汉市江汉区解放大道686号世贸大厦3001室
邮编: 430022

成都

成都市锦江区东御街19号茂业天地3907室
邮编: 610066

西安

西安市唐延路1号旺座国际城D座1006室
邮编: 710065

宁波

浙江省宁波市江东区惊驾路565号泰富广场B座309室
邮编: 315040

广州

广州市天河区林和西路9号耀中广场B座40层06室
邮编: 510610

重庆

重庆市江北区福泉路18号源著南区20栋2108室
邮编: 400000

■ 北美公司

MOONS' INDUSTRIES (AMERICA), INC. (Chicago)
1113 North Prospect Avenue, Itasca, IL 60143 USA

MOONS' INDUSTRIES (AMERICA), INC. (Boston)
36 Cordage Park Circle, Suite 310 Plymouth, MA 02360 USA

APPLIED MOTION PRODUCTS, INC.
404 Westridge Dr. Watsonville, CA 95076, USA

LIN ENGINEERING, INC.
16245 Vineyard Blvd., Morgan Hill, CA 95037

■ 欧洲公司

MOONS' INDUSTRIES (EUROPE) S.R.L.
Via Torri Bianche n.1 20871 Vimercate(MB) Italy

AMP & MOONS' AUTOMATION(GERMANY)GMBH
Börsenstr. 14
60313 Frankfurt am Main Germany

■ 东南亚公司

MOONS' INDUSTRIES (SOUTH-EAST ASIA) PTE. LTD.
33 Ubi Avenue 3 #08-23 Vertex Singapore 408868

■ 日本公司

MOONS' INDUSTRIES JAPAN CO., LTD.
Room 601, 6F, Shin Yokohama Koushin Building
2-12-1, Shin-Yokohama, Kohoku-ku, Yokohama
Kanagawa, 222-0033, Japan