

---

伺服通用系列

CANOPEN 通讯

# 用户手册 V1.2

更新日期:20160913

---

简介.....	4
一、CANOPEN 协议介绍.....	5
1.1、波特率.....	5
1.2、标识符分配设备.....	5
1.3、过程数据对象（PDO）通讯介绍.....	6
1.4、服务数据对象（SDO）通讯协议.....	6
1.5、MNT 节点保护（ NMT Node Guarding）.....	8
1.6、紧急事件（ Emergency）.....	8
二、过程数据对象(PDO)映射.....	9
2.1、驱动器默认参数.....	9
2.2、举例说明 RXPDO 映射.....	10
2.2.1 配置 RXPDO 0.....	10
2.2.2 配置 RXPDO 1.....	12
2.3、举例说明 TXPDO 映射.....	13
2.3.1 配置 TXPDO 0.....	13
2.3.2 配置 TXPDO 1.....	16
三、CANOPEN 伺服驱动器启动流程介绍.....	19
四、CANOPEN 通讯控制伺服驱动器介绍.....	20
4.1、控制状态机介绍.....	20
4.2、ControlWord (控制字)说明.....	22
4.3、StatusWord (状态控制字)说明.....	22
4.4、控制方式介绍.....	23
4.4.1、控制模式相关参数.....	24
4.4.2、OperationMode.....	24
4.4.3、DisplayMode.....	24
五、回零模式（ HOMING MODE）.....	26
5.1、回零模式的控制字.....	26
5.2、回零模式的状态字.....	26
5.3、回零模式相关参数.....	27
六、速度控制模式（ PROFILE VELOCITY MODE）.....	29

---

6.1、速度模式的控制字.....	29
6.2、速度模式的状态字.....	29
6.3、速度控制模式相关参数.....	30
<b>七、电流控制模式 ( PROFILE_TORQUE_MODE )</b> .....	<b>33</b>
7.1、电流模式的状态字.....	33
7.2、电流控制模式相关参数.....	33
<b>八、位置控制模式 ( PROFILE POSITION MODE )</b> .....	<b>37</b>
8.1、位置模式的控制字.....	37
8.2、位置模式的状态字.....	37
8.3、位置控制相关参数.....	37
8.4、位置控制功能.....	42
8.4.1、跟随误差(following error).....	42
8.4.2、位置到达.....	43
8.5、位置控制相关参数.....	44
<b>九、周期同步位置模式 ( CYCLIC SYNC POSITION MODE )</b> .....	<b>47</b>
<b>十、应用举例</b> .....	<b>48</b>
10.1、控制速度环.....	48
10.2、控制位置环.....	50
10.2.1、绝对或相对位置控制模式.....	50
10.2.2、功能描述.....	51
10.3、控制电流环.....	54
10.4、参数保存.....	56
10.5、设置位置齿轮比.....	56
<b>10.6、设置给定速度比</b> .....	<b>56</b>
<b>10.7、设置反馈速度比</b> .....	<b>56</b>
附录.....	57
对象字典--- CiA402 device profile.....	57
对象字典--- CiA301 通讯简介.....	60

---

# 简介

支持标准 CANopen 通讯协议的系列伺服驱动器, 允许通过 CANopen 总线对 Canopen 通讯参数进行设置, 驱动器所有参数可以通过 CANopen 控制, 在 CANopen 控制模式下可以驱动电机的转速, 位置, 力矩, 启停设定。伺服驱动器在 CANopen 总线网络中只能做为从站使用, 所有的参数、参数值和功能都是通过 index 和 sub-index 组成的地址来访问和存取。

CANOPEN 伺服驱动器内部参数已经实现所有参数映射, 并且完全符合 DS301 标准及部分 DS402 协议。并且已经在 Beckhoff-CX9010 主站的 EL6751 CANopen IO 设备成调试并且运行成功。

注意: CANOPEN 通讯技术适用于有相关了解的技术人员开发。

# 一、CANOPEN 协议介绍

## 1.1、波特率

通用伺服驱动器的 CANOPEN 通讯波特率出厂时预设为 500k bit/s。其中波特率的高低与通讯长度和通讯媒介有关，具体看下表。

波特率	最大传输距离	10BE 参数值
1M	40m	0
500k	130m	1
250k	270m	2
125k	530m	3
50k	1300m	4
20k	3300m	5

## 1.2、标识符分配设备

ID 范围为 1 - 127，驱动器出厂的节点 ID 默认值为 0x01；

对象	COD-Ids	通讯参数在 OD 中的索引
NMT	0	-
SYNC	80h	1005h
Emergency	80h+节点地址	1014h
Tx-PDO1	180h+节点地址	1800h
Rx-PDO1	200h+节点地址	1400h
Tx-PDO2	280h+节点地址	1801h
Rx-PDO2	300h+节点地址	1401h
Tx-PDO3	380h+节点地址	1802h
Rx-PDO3	400h+节点地址	1402h
Tx-PDO4	480h+节点地址	1803h
Rx-PDO4	500h+节点地址	1403h
Tx-SDO	580h+节点地址	1200h
Rx-SDO	600h+节点地址	1200h
NMT Node Guarding	0x700 + Node_ID	

## 1.3、过程数据对象（PDO）通讯介绍

目前最大支持 4 个 RPDO 和 TPDO，相关参数符合 DS301 标准。

PDO 的通讯配置，每个 PDO 包含三个通讯参数：ID 号，传输类型和禁止时间。传输类型按 CANOPEN 协议规定有如下多种：

——同步：

**0:** 同步报文，对于 RPDO，接收到 PDO 数据时，数据并不会马上更新，而是要接收到同步的报文数据才会更新。对于 TPDO，在接收到同步报文时才会发送数据。

**1-240:** 同步报文，（周期性）对于 RPDO，在接收到（1~240）同步报文(SYNC)之后，数据才会更新。对于 TPDO，在接收到（1~240）同步报文(SYNC)之后，TPDO 数据才会发送出去。

**252:** 由远程帧来的请求命令同步更新 PDO 并响应请求，（暂时不支持）。

——异步：

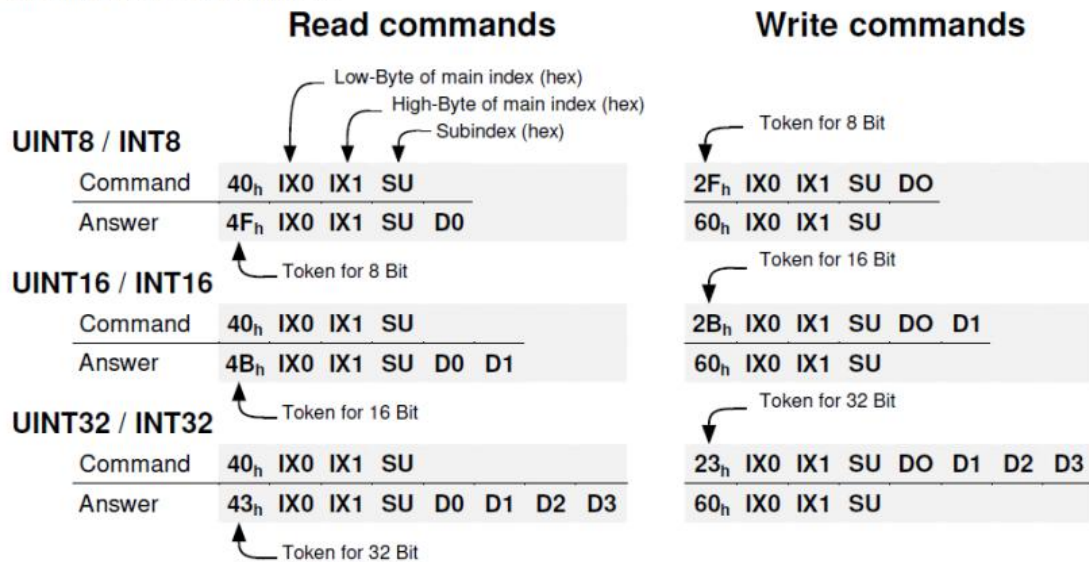
**253:** PDO 连续更新，但只在远程帧请求后触发，（暂时不支持）。

**255:** 非同步（对于 RxPDO，每次接收到数据时立即会更新。对于 TxPDO，在内容发生变化并超过禁止时间内时就发送），禁止时间只适用于 TxPDO，缺省 100 微秒。

CANOPEN 伺服驱动器 采用同步报文方式，并且 RxPDO 和 TxPDO 默认的传输类型均为 255。

## 1.4、服务数据对象（SDO）通讯协议

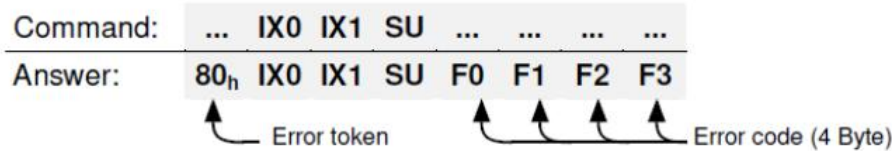
SDO 报文对参数读/写操作格式：



举例:

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

SDO 错误报文格式:



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

## 1.5、MNT 节点保护（ NMT Node Guarding）

CANOPEN 系列伺服驱动器被配置为产生周期性的被称作心跳报文（ Heartbeat）的报文。

Heartbeat Producer → Consumer(s)

COB-ID	Byte 0
0x700 + Node_ID	状态

状态可为下表种的数值：

状态	意义
0	Boot-up
4	Stopped
5	Operational
127(0x7F)	Pre-operational

当一个 Heartbeat 节点启动后它的 Boot-up 报文是其第一个 Heartbeat 报文。Heartbeat 消费者通常是 NMT-Master 节点，它为每个 Heartbeat 节点设定一个超时值，当超时发生时采取相应动作。

## 1.6、紧急事件（ Emergency）

如果伺服驱动器的故障号出现变化，则会主动上报该故障给主机，根据 DS301 协议故障码是 0x1000，上报伺服驱动器故障错误代码 FaultCode。

0-1	2	3-4	5-7
0x1000	0x01	FaultCode	0x00

上报的故障错误代码 FaultCode 是对应驱动器内部参数表地址 0x1005(索引是 0x6105)，具体定义如下：

- Bit0 1: 电源欠压(PwrLow)
- Bit1 1: 位置异常(PosErr)
- Bit2 1: 霍尔错误(反馈异常) (HallErr)
- Bit3 1: 过流(CurOvr)
- Bit4 1: 超载(DrvOvr)
- Bit5 1: EEPROM 故障(EromRrr)
- Bit6 1: IGBT 故障(IGBTErr)
- Bit7 1: 驱动器过热(DrvHot)
- Bit8 1: 电机缺相(MotErr)
- Bit9 1: 电流超差(CurErr)
- Bit10 1: 速度超差(SpdErr)
- Bit11 1: 电机过热(MotHot)
- Bit12 1: 电源过压(PwrHig)



---

## 二、过程数据对象(PDO)映射

### 2.1、驱动器默认参数

驱动器默认参数的 TXPDO 为周期 100ms 发送，RXPDO 为异步接收，其映射如下：

TXPDO 0		
子索引	值(H)	驱动器内部参数名称
0	4	-----
1	60410010	状态字(StatusWord)
2	20000010	驱动器故障信号
3	20010010	驱动器状态信号
4	60610008	显示工作模式 (DisplayMode)

TXPDO 1		
子索引	值(H)	驱动器内部参数名称
0	2	-----
1	60640020	当前位置反馈值
2	60690020	当前速度反馈值

TXPDO 2		
子索引	值(H)	驱动器内部参数名称
0	2	-----
1	60620020	目标位置期望值
2	606B0020	目标速度期望值

RXPDO 0		
子索引	值(H)	驱动器内部参数名称
0	2	-----
1	60400010	控制字(ControlWord)
2	60600008	工作模式 (OperationMode)

RXPDO 1		
子索引	值(H)	驱动器内部参数名称
0	2	-----
1	607A0020	电机位置设定 (TargetPosition)
2	60810020	行规位置模式速度设定 (ProfileVelocity)

RXPDO 2		
子索引	值(H)	驱动器内部参数名称
0	1	-----
1	60FF0020	行规速度环的速度给定

## 2.2、举例说明 RXPDO 映射

如果配置驱动器默认参数的 RXPDO 映射，假设节点 id 为 1。

### 2.2.1 配置 RXPDO 0

(1)、先不使能 RXPDO 0 接收  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	14	01	01	02	00	80

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	14	01	00	00	00	00

(2)、清空 RXPDO 0 映射  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	00	16	00	00	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	16	00	00	00	00	00

(3)、配置 RXPDO 0 的子索引为 01 的映射为 60400010，即控制字。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	16	01	10	00	40	60

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	16	01	00	00	00	00

(4)、配置 RXPDO 0 的子索引为 02 的映射为 60600008，即工作方式。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	16	02	08	00	60	60

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	16	02	00	00	00	00

(5)、配置 RXPDO 0 为异步触发方式。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	00	14	02	FF	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	14	02	00	00	00	00

(6)、设置 RXPDO 0 映射的对象个数为 2

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	00	16	00	02	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	16	00	00	00	00	00

(7)、使能 RXPDO 0 接收

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	14	01	01	02	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	14	01	01	02	00	00

581	60	00	14	01	00	00	00	00
-----	----	----	----	----	----	----	----	----

## 2.2.2 配置 RXPDO 1

(1)、先不使能 RXPDO 1 接收  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	01	14	01	01	03	00	80

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	14	01	00	00	00	00

(2)、清空 RXPDO 1 映射  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	01	16	00	00	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	16	00	00	00	00	00

(3)、配置 RXPDO 1 的子索引为 01 的映射为 607A0020，即给定位置。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	01	16	01	20	00	7A	60

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	16	01	00	00	00	00

(4)、配置 RXPDO 1 的子索引为 02 的映射为 60810020，即工作方式。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	01	16	02	20	00	81	60

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	16	02	00	00	00	00

(5)、配置 RXPDO 1 的为异步触发方式。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	01	14	02	FF	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	14	02	00	00	00	00

(6)、设置 RXPDO 1 映射的对象个数为 2

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	01	16	00	02	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	16	00	00	00	00	00

(7)、使能 RXPDO 1 接收

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	01	14	01	01	03	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	14	01	00	00	00	00

## 2.3、举例说明 TXPDO 映射

如果配置驱动器默认参数的 TXPDO 映射，假设节点 id 为 1.

### 2.3.1 配置 TXPDO 0

(1)、先不使能 TXPDO 0 接收

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	18	01	81	01	00	80

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	18	01	00	00	00	00

(2)、清空 TXPDO 0 映射

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	00	1A	00	00	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	1A	00	00	00	00	00

(3)、配置 TXPDO 0 的子索引为 01 的映射为 60410010，即状态字。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	1A	01	10	00	41	60

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	1A	01	00	00	00	00

(4)、配置 TXPDO 0 的子索引为 02 的映射为 20000010，即驱动器故障信号。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	1A	02	10	00	00	20

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	1A	02	00	00	00	00

(5)、配置 TXPDO 0 的子索引为 03 的映射为 20010010，即驱动器状态信号。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	1A	03	10	00	01	20

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	1A	03	00	00	00	00

(6)、配置 TXPDO 0 的子索引为 04 的映射为 60610008，即显示工作方式。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	1A	04	10	00	01	08

<b>601</b>	22	00	1A	04	08	00	61	60
------------	----	----	----	----	----	----	----	----

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	00	1A	04	00	00	00	00

(7)、设置 TXPDO 0 映射的对象个数为 4  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2F	00	1A	00	03	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	00	1A	00	00	00	00	00

(8)、配置 TXPDO 0 的 为异步触发方式。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2F	00	18	02	FF	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	00	18	02	00	00	00	00

(9)、配置 TXPDO 0 的抑制时间位 10ms。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2B	00	18	03	64	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	00	18	03	00	00	00	00

(10)、配置 TXPDO 0 的触发时间位 10ms。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2B	00	18	05	0A	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	00	18	05	00	00	00	00

(11)、使能 TXPDO 0 接收  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	00	18	01	81	01	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	18	01	00	00	00	00

## 2.3.2 配置 TXPDO 1

(1)、先不使能 TXPDO 1 接收  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	01	18	01	81	02	00	80

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	00	18	01	00	00	00	00

(2)、清空 TXPDO 1 映射  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	2F	01	1A	00	00	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	1A	00	00	00	00	00

(3)、配置 TXPDO 1 的子索引为 01 的映射为 60640020，即当前位置反馈值。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	01	1A	01	20	00	64	60

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
581	60	01	1A	01	00	00	00	00

(4)、配置 TXPDO 1 的子索引为 02 的映射为 60690020，即当前速度反馈值。  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
-------	--------	--------	--------	--------	--------	--------	--------	--------



<b>601</b>	22	01	1A	02	20	00	69	60
------------	----	----	----	----	----	----	----	----

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	01	1A	02	00	00	00	00

(5)、设置 TXPDO 1 映射的对象个数为 2

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2F	01	1A	00	02	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	01	1A	00	00	00	00	00

(6)、配置 TXPDO 1 为异步触发方式。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2F	01	18	02	FF	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	01	18	02	00	00	00	00

(7)、配置 TXPDO 1 的抑制时间位 10ms。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2B	01	18	03	64	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	01	18	03	00	00	00	00

(8)、配置 TXPDO 1 的触发时间位 10ms。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	2B	01	18	05	0A	00	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	01	18	05	00	00	00	00

---

(9)、使能 TXPDO 1 接收  
上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>601</b>	22	01	18	01	81	02	00	00

驱动器响应

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
<b>581</b>	60	01	18	01	00	00	00	00

---

## 三、CANOPEN 伺服驱动器启动流程介绍

如果用户通过 CANOPEN 接口控制本公司的伺服驱动器，则其控制方式遵循上 DS402 协议的控制方式。

首先，CANOPEN 伺服驱动器上电后，会自动进入初始化状态(0x00)，并且会向主站发送一个心跳报文，然后进去预操作状态(0x7F)，等待主站的应答控制驱动器的节点状态。

只有当从站进入操作状态(0x05)，才进入过程数据操作的状态，驱动器会定时上报过程数据，并且主站也可以通过过程数据控制驱动器。

由于控制方式都是标准控制方式，并且已经在倍福主站 CX9010-0001 成功控制，可以为用户提供一份 eds 文件说明，在此不必详细介绍。

即总控制器只需要向 id: 0x00 发送 2 个字节数据: 01 02 (第一个字节表示命令，第二个字节表示节点地址)，即可进入正常操作模式，即可以进行 PDO 通讯。

(总控制器只需要向 id: 0x00 发送 2 个字节数据: 80 02 (第一个字节表示命令，第二个字节表示节点地址)，即可进入预操作操作模式，不可以进行 PDO 通讯)

(总控制器只需要向 id: 0x00 发送 2 个字节数据: 02 02 (第一个字节表示命令，第二个字节表示节点地址)，即可进入停止模式，不可以进行 PDO 和 SDO 通讯)

---

## 四、CANOPEN 通讯控制伺服驱动器介绍

### 4.1、控制状态机介绍

伺服驱动器支持状态机，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：

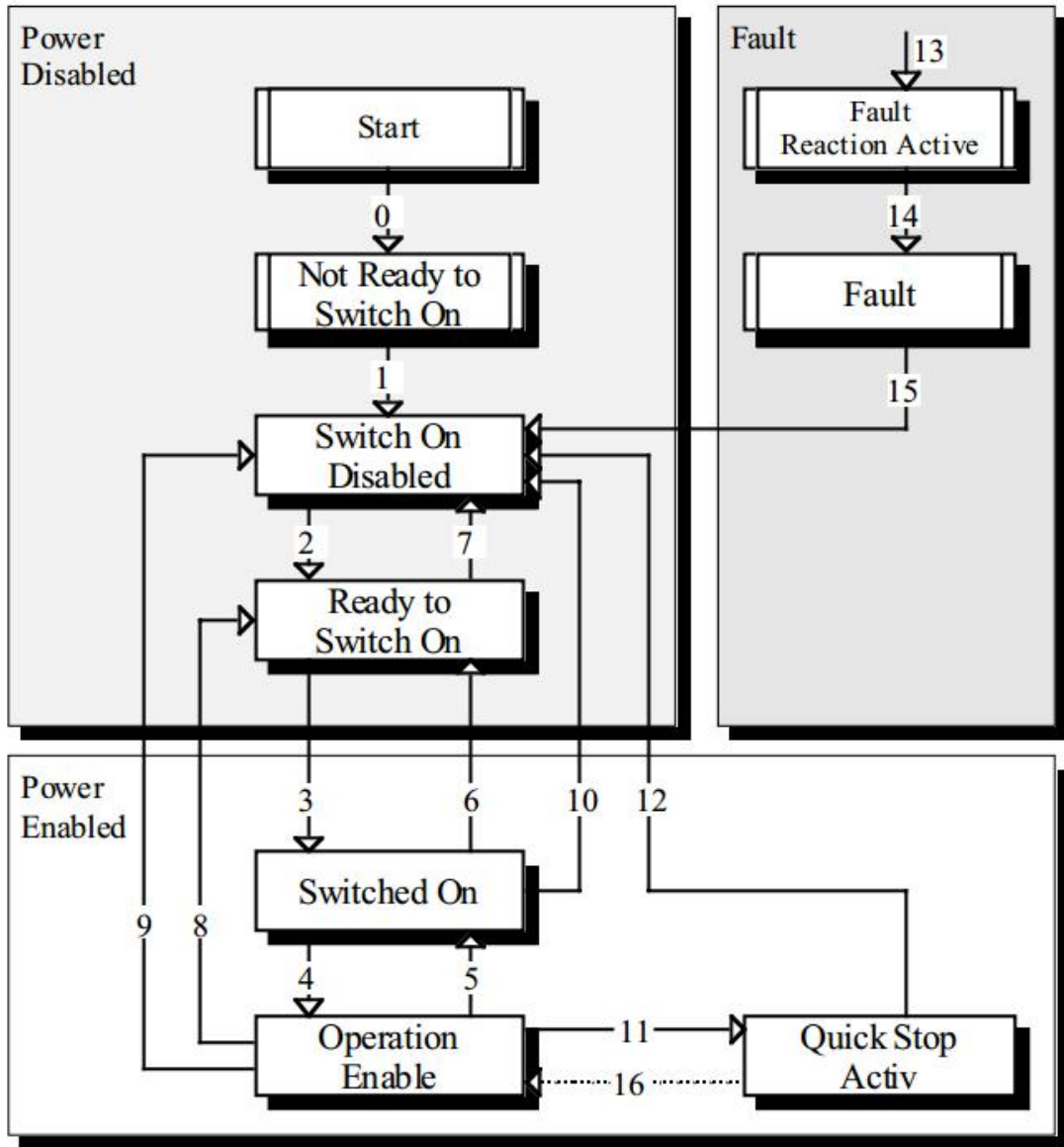
**State(状态):** 如果主电激活或发生报警，伺服驱动都处在不同状态。CANOPEN 总线控制下的状态机将在本节重点讲解。

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)。



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。

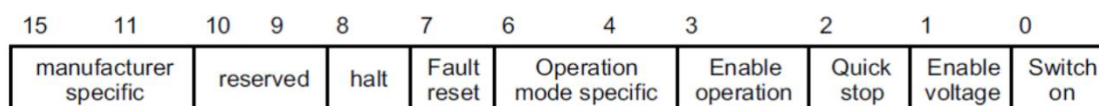
状态名	描述
Not Ready to Switch On	伺服驱动器正在初始化过程中，不能进行 CAN 通讯。
Switch On Disabled	伺服驱动器初始化完成，可以进行 CAN 通讯。

<b>Ready to Switch On</b>	伺服驱动器等待进入 Switch On 状态，电机没有被励磁。
<b>Switched On</b>	伺服驱动器伺服准备好状态，主电已上。
<b>Operation Enable</b>	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。
<b>Quick Stop Active</b>	伺服驱动器将根据设定的方式停机。
<b>Fault Reaction Active</b>	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。
<b>Fault</b>	电机无励磁信号。

## 4.2、ControlWord (控制字)说明

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：



## 4.3、StatusWord (状态控制字)说明

Index	6041h
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

#### 4.4、控制方式介绍

伺服驱动器目前支持 CANOPEN DSP402 中的 4 种控制模式：

回零模式（ HOMING MODE）

速度控制模式（ PROFILE VELOCITY MODE）

电流控制模式（ PROFILE\_TORQUE\_MODE）

位置控制模式（ PROFILE POSITION MODE）

周期同步位置模式（ CYCLIC SYNC POSITION MODE ）

#### 4.4.1、控制模式相关参数

Index	Object	Name	Type	Attr.
6060 h	VAR	OperationMode	INT8	RW
6061 h	VAR	DisplayMode	INT8	RO

#### 4.4.2、OperationMode

伺服驱动器的控制模式由 OperationMode 参数决定。

Index	6060h
Name	OperationMode
Object Code	VAR
Data Type	INT8
Access	RW
PDO Mapping	YES
Units	--
Value Range	1、3、4、6、8
Default Value	--

OperationMode 的值说明：

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）
4	电流控制模式（PROFILE TORQUE MODE）
8	周期同步位置模式（CYCLIC SYNC POSITION MODE）

#### 4.4.3、DisplayMode

伺服驱动器当前的控制模式可以通过读 DisplayMode 参数知道。

Index	6061h
Name	DisplayMode
Object Code	VAR



---

<b>Data Type</b>	INT8
<b>Access</b>	RO
<b>PDO Mapping</b>	YES
<b>Units</b>	--
<b>Value Range</b>	1、3、4、6、8
<b>Default Value</b>	--

注： 1、只能通过 DisplayMode 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，DisplayMode 才能跟 OperationMode 一致。

## 五、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。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 HomePosition（607C h）距离。

### 5.1、回零模式的控制字

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

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

### 5.2、回零模式的状态字

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

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

### 5.3、回零模式相关参数

Index	Object	Name	Type	Attr.
607C h	VAR	HomeOffset	INT32	RW
6098 h	VAR	HomingMethod	INT8	RW
6099 h	ARRAY	SHOM_Speed	UINT32	RW

#### 5.3.1 Object 607Ch: Home offset

回零偏移是应用程序与机械零点的的偏差。

<b>Index</b>	<b>607Ch</b>
<b>Name</b>	HomeOffset
<b>Object Code</b>	VAR
<b>Data Type</b>	INT32
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	脉冲个数
<b>Value Range</b>	INT32
<b>Default Value</b>	0

#### 5.3.2 Object 6098h: Homing method

<b>Index</b>	<b>6098h</b>
<b>Name</b>	Homing method

<b>Object Code</b>	VAR
<b>Data Type</b>	INT8
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	----
<b>Value Range</b>	INTEGER8
<b>Default Value</b>	34

### 5.3.3 Object 6099<sub>h</sub>: Homing speeds

<b>Sub-Index</b>	<b>0</b>
<b>Description</b>	number of entries
<b>Entry Category</b>	Mandatory
<b>Access</b>	ro
<b>PDO Mapping</b>	No
<b>Value Range</b>	2
<b>Default Value</b>	2

<b>Sub-Index</b>	<b>1</b>
<b>Description</b>	Speed during search for switch
<b>Entry Category</b>	Mandatory
<b>Access</b>	rw
<b>PDO Mapping</b>	Possible
<b>Value Range</b>	UNSIGNED32
<b>Default Value</b>	100

<b>Sub-Index</b>	<b>2</b>
<b>Description</b>	Speed during search for zero
<b>Entry Category</b>	Mandatory
<b>Access</b>	rw
<b>PDO Mapping</b>	Possible
<b>Value Range</b>	UNSIGNED32
<b>Default Value</b>	100

## 六、速度控制模式 ( PROFILE VELOCITY MODE)

### 6.1、速度模式的控制字

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

Name	Value	Description
Halt	0	Execute the motion
	1	Stop axle

### 6.2、速度模式的状态字

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

Name	Value	Description
Target reached	0	Halt = 0: <i>Target velocity</i> not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: <i>Target velocity</i> 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

### 6.3、速度控制模式相关参数

Index	Name
6069h	VelocitySensorActualValue
606Ah	SensorSelect
606Bh	VelocityDemandValue
606Ch	VelocityActualValue
606D h	Velocity window
606E h	Velocity window time
606F h	Velocity threshold
6070 h	Velocity threshold time
60FF h	TargetVelocity

#### 6.3.1 Object 6069h: Velocity sensor actual value

Index	6069h
Name	Velocity sensor actual value
Object Code	VAR
Data Type	INT32
Access	RO
PDO Mapping	Possible
Units	转/分
Value Range	INT32
Default Value	0

#### 6.3.2 Object 606Ah: Sensor selection code

Index	606Ah
Name	Sensor selection code
Object Code	VAR
Data Type	INT16
Access	RO
PDO Mapping	Possible
Units	---
Value Range	INT16
Default Value	0

#### 6.3.3 Object 606Bh: Velocity demand value

Index	606Bh
Name	Velocity demand value
Object Code	VAR
Data Type	INT32

<b>Access</b>	RO
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	INT32
<b>Default Value</b>	0

#### 6.3.4 Object 606Ch: Velocity actual value

<b>Index</b>	<b>606Ch</b>
<b>Name</b>	Velocity actual value
<b>Object Code</b>	VAR
<b>Data Type</b>	INT32
<b>Access</b>	RO
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	INT32
<b>Default Value</b>	0

#### 6.3.5 Object 606Dh: Velocity window

<b>Index</b>	<b>606Dh</b>
<b>Name</b>	Velocity window
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	UINT16
<b>Default Value</b>	0

#### 6.3.6 Object 606Eh: Velocity window time

<b>Index</b>	<b>606Eh</b>
<b>Name</b>	Velocity window time
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	UINT16
<b>Default Value</b>	0

---

### 6.3.7 Object 606Fh: Velocity threshold

<b>Index</b>	<b>606Fh</b>
<b>Name</b>	Velocity threshold
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	UINT16
<b>Default Value</b>	0

### 6.3.8 Object 6070h: Velocity threshold time

<b>Index</b>	<b>6070h</b>
<b>Name</b>	Velocity threshold time
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	UINT16
<b>Default Value</b>	0

### 6.3.8 Object 60FFh: Target velocity

<b>Index</b>	<b>60FFh</b>
<b>Name</b>	Target velocity
<b>Object Code</b>	VAR
<b>Data Type</b>	INT32
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	转/分
<b>Value Range</b>	INT32
<b>Default Value</b>	0



## 七、电流控制模式 ( PROFILE\_TORQUE\_MODE)

### 7.1、电流模式的状态字

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

Name	Value	Description
Target reached	0	Halt = 0: <i>Target velocity</i> not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: <i>Target velocity</i> reached Halt = 1: Axle has velocity 0

### 7.2、电流控制模式相关参数

Index	Name
6071h	TargetTorque
6072h	Max torque
6073h	Max current
6074h	Torque demand value
6075h	Motor rated current
6076h	Motor rated torque
6077h	Torque actual value
6078h	Current actual value
6087h	Torque slope

#### 7.2.1 Object 6071h: Target torque

<b>Index</b>	<b>6071h</b>
<b>Name</b>	Target torque
<b>Object Code</b>	VAR
<b>Data Type</b>	INT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	INT16
<b>Default Value</b>	0

---

#### 7.2.2 Object 6072<sub>h</sub>: Max torque

<b>Index</b>	<b>6072<sub>h</sub></b>
<b>Name</b>	Max torque
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	UINT16
<b>Default Value</b>	0

#### 7.2.3 Object 6073<sub>h</sub>: Max current

<b>Index</b>	<b>6073<sub>h</sub></b>
<b>Name</b>	Max current
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT16
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	UINT16
<b>Default Value</b>	0

#### 7.2.4 Object 6074<sub>h</sub>: Torque demand value

<b>Index</b>	<b>6074<sub>h</sub></b>
<b>Name</b>	Torque demand value
<b>Object Code</b>	VAR
<b>Data Type</b>	INT16
<b>Access</b>	RO
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	INT16
<b>Default Value</b>	0

#### 7.2.5 Object 6075<sub>h</sub>: Motor rated current

<b>Index</b>	<b>6075<sub>h</sub></b>
<b>Name</b>	Motor rated current
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RO

<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	UINT32
<b>Default Value</b>	0

#### 7.2.6 Object 6076h: Motor rated torque

<b>Index</b>	<b>6076h</b>
<b>Name</b>	Motor rated torque
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	UINT32
<b>Default Value</b>	0

#### 7.2.7 Object 6077h: Torque actual value

<b>Index</b>	<b>6077h</b>
<b>Name</b>	Torque actual value
<b>Object Code</b>	VAR
<b>Data Type</b>	INT16
<b>Access</b>	RO
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	INT16
<b>Default Value</b>	0

#### 7.2.8 Object 6078h: Current actual value

<b>Index</b>	<b>6078h</b>
<b>Name</b>	Current actual value
<b>Object Code</b>	VAR
<b>Data Type</b>	INT16
<b>Access</b>	RO
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	INT16
<b>Default Value</b>	0

#### 7.2.9 Object 6087h: Torque slope

<b>Index</b>	<b>6087h</b>
<b>Name</b>	Torque slope

---

<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	---
<b>Value Range</b>	UINT32
<b>Default Value</b>	0

---

## 八、位置控制模式 ( PROFILE POSITION MODE)

### 8.1、位置模式的控制字

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

Name	Value	Description
New set-point	0	Does not assume <i>target position</i>
	1	Assume <i>target position</i>
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	<i>Target position</i> is an absolute value
	1	<i>Target position</i> is a relative value
Halt	0	Execute positioning
	1	Stop axle with <i>profile deceleration</i> (if not supported with <i>profile acceleration</i> )

### 8.2、位置模式的状态字

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

Name	Value	Description
Target reached	0	Halt = 0: <i>Target position</i> not reached Halt = 1: Axle decelerates
	1	Halt = 0: <i>Target position</i> 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 values
Following error	0	No following error
	1	Following error

### 8.3、位置控制相关参数

Index	Name
6064h	Position actual value

607Ah	TargetPosition
607Bh	Positionrange limit
607Dh	Software position limit
607Fh	Maximun profile_velocity
6081 h	Profile_velocity
6082 h	End_velocity
6083 h	Profile_acceleration
6084 h	Profile_deceleration
6086 h	Motion_profile_type
60C5 h	Maximun acceleration
60C6 h	Maximun deceleration
60F2 h	Positioning option code

### 8.3.1 Object 6064h: Position actual value

<b>Index</b>	<b>6064h</b>
<b>Name</b>	Position actual value
<b>Object Code</b>	VAR
<b>Data Type</b>	INT32
<b>Access</b>	RO
<b>PDO Mapping</b>	YES
<b>Units</b>	position units
<b>Value Range</b>	----
<b>Default Value</b>	----

### 8.3.2 Object 607Ah: Target position

<b>Index</b>	<b>607Ah</b>
<b>Name</b>	Target position
<b>Object Code</b>	VAR
<b>Data Type</b>	INT32
<b>Access</b>	RW
<b>PDO Mapping</b>	YES
<b>Units</b>	position units
<b>Value Range</b>	----
<b>Default Value</b>	0

### 8.3.3 Object 607Bh: Position range limit

<b>Sub-Index</b>	<b>0</b>
<b>Description</b>	number of entries

<b>Entry Category</b>	Mandatory
<b>Access</b>	ro
<b>PDO Mapping</b>	No
<b>Value Range</b>	2
<b>Default Value</b>	2

<b>Sub-Index</b>	<b>1</b>
<b>Description</b>	Min position range limit
<b>Entry Category</b>	Mandatory
<b>Access</b>	rw
<b>PDO Mapping</b>	NO
<b>Value Range</b>	INT32
<b>Default Value</b>	-2 <sup>31</sup>

<b>Sub-Index</b>	<b>2</b>
<b>Description</b>	Max position range limit
<b>Entry Category</b>	Mandatory
<b>Access</b>	rw
<b>PDO Mapping</b>	NO
<b>Value Range</b>	INT32
<b>Default Value</b>	2 <sup>31</sup> -1

#### 8.3.4 Object 607D<sub>h</sub>: Software position limit

<b>Sub-Index</b>	<b>0</b>
<b>Description</b>	number of entries
<b>Entry Category</b>	Mandatory
<b>Access</b>	ro
<b>PDO Mapping</b>	No
<b>Value Range</b>	2
<b>Default Value</b>	-2 <sup>31</sup>

<b>Sub-Index</b>	<b>1</b>
<b>Description</b>	Min position limit
<b>Entry Category</b>	Mandatory
<b>Access</b>	rw
<b>PDO Mapping</b>	NO
<b>Value Range</b>	INT32
<b>Default Value</b>	2 <sup>31</sup> -1

<b>Sub-Index</b>	<b>2</b>
<b>Description</b>	Max position limit
<b>Entry Category</b>	Mandatory
<b>Access</b>	rw
<b>PDO Mapping</b>	NO
<b>Value Range</b>	INT32
<b>Default Value</b>	0

### 8.3.5 Object 6081h: profile\_velocity

profile\_velocity 是指位置启动后，完成加速后最终到达的速度。（单位：转/分钟）

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

### 8.3.6 Object 6082h: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，并且默认与 profile\_velocity 大小相同。（单位：转/分钟）

<b>Index</b>	<b>6082h</b>
<b>Name</b>	end_velocity
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	YES
<b>Units</b>	speed units
<b>Value Range</b>	----
<b>Default Value</b>	0

### 8.3.7 Object 6083h:profile\_acceleration

profile\_acceleration 是到达给定位置期间的加速度。

<b>Index</b>	<b>6083h</b>
<b>Name</b>	profile_acceleration
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32



<b>Access</b>	RW
<b>PDO Mapping</b>	YES
<b>Units</b>	Accelerationunits
<b>Value Range</b>	----
<b>Default Value</b>	2000R/min/s

### 8.3.8 Object 6084h: profile\_deceleration

profile\_deceleration 是到达给定位置期间的减速度。(暂不支持)

<b>Index</b>	<b>6084h</b>
<b>Name</b>	profile_deceleration
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	YES
<b>Units</b>	Accelerationunits
<b>Value Range</b>	----
<b>Default Value</b>	6000R/min/s

### 8.3.9 Object 6086h: 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。写 0 表示选择梯形速度曲线。(目前只支持梯形速度曲线)

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

### 8.3.10 Object 60C5h: Max acceleration

<b>Index</b>	<b>60C5h</b>
<b>Name</b>	Max acceleration
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	----
<b>Value Range</b>	0
<b>Default Value</b>	0

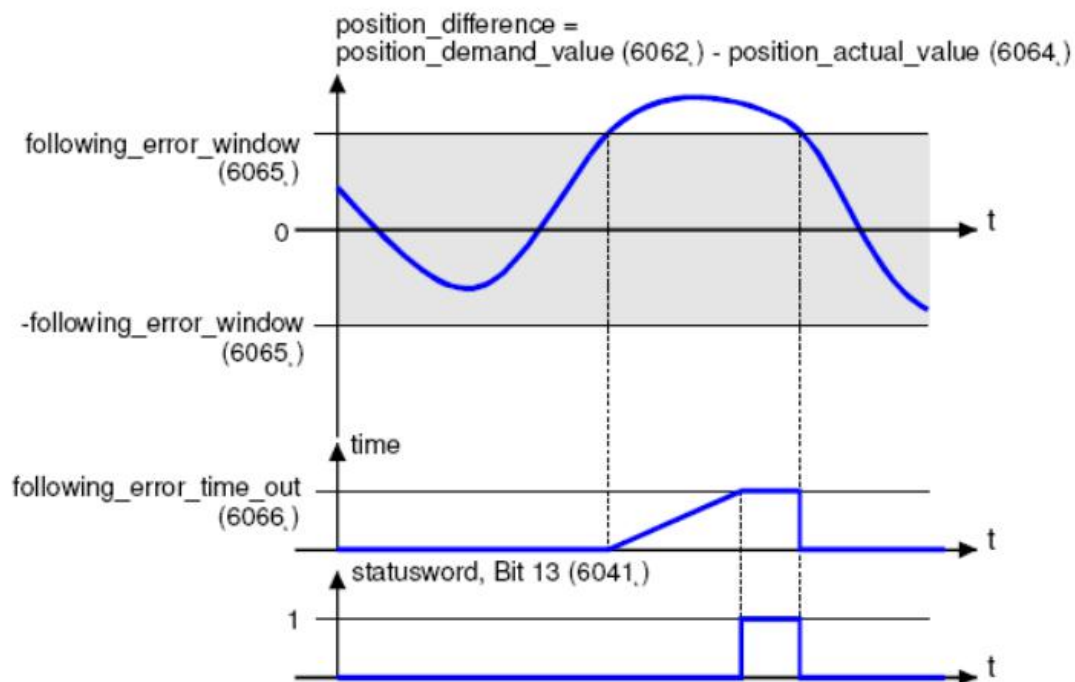
### 8.3.11 Object 60C6h: Max deceleration

<b>Index</b>	<b>60C6h</b>
<b>Name</b>	Max deceleration
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	Possible
<b>Units</b>	----
<b>Value Range</b>	0
<b>Default Value</b>	0

## 8.4、位置控制功能

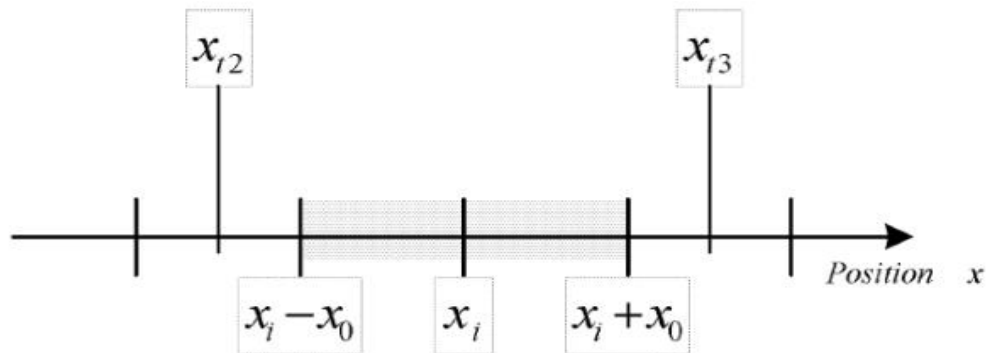
本章节主要描述位置控制模式下所需要的参数。 Trajectory 单元输出的期望位置 (position\_demand\_value) 作为驱动器位置环的输入。此外, 实际位置 (position\_actual\_value) 是通过电机编码器测量出。位置控制器的动作受参数设置影响。为了控制系统的稳定性, 必须对位置环输出值 (control\_effort) 限幅。该输出值作为速度环给定速度。在 Factor group 单元, 所有的输入和输出值都转换成驱动器相应的内部单位。下面介绍位置控制时的子功能:

### 8.4.1、跟随误差(following error)



跟随误差(following error) –功能描述

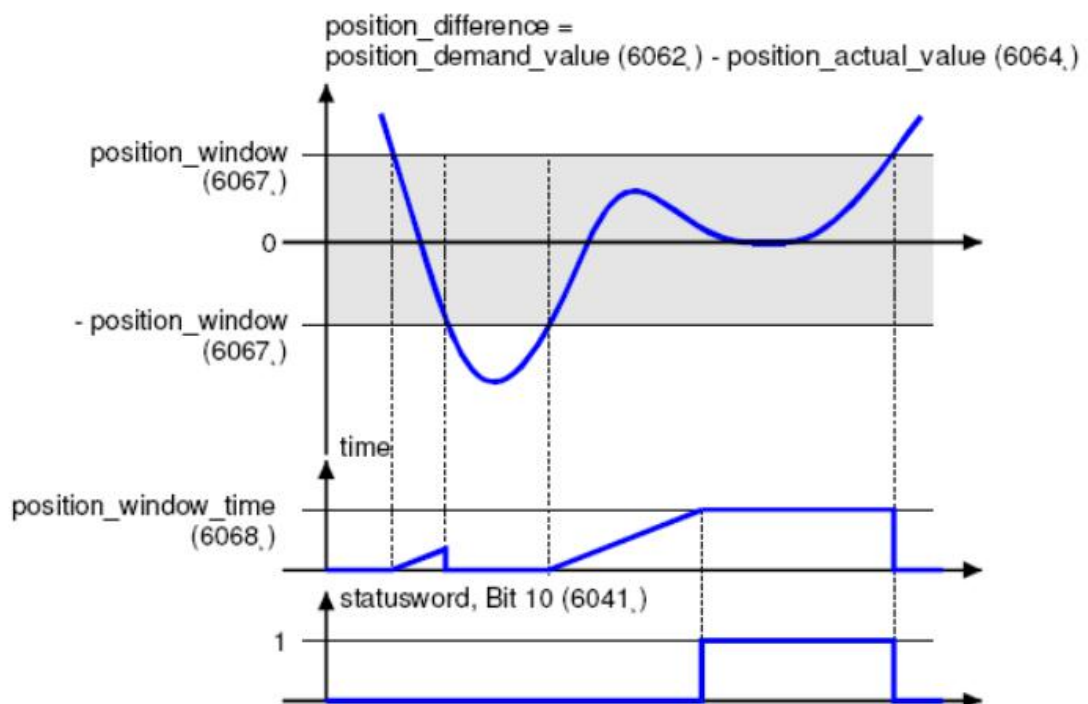
跟随误差指的是实际位置(position\_actual\_value)和期望位置(position\_demand\_value)的偏差。上图所示，如果在设定时间(following\_error\_time\_out)内，跟随误差值大于跟随误差窗口(following\_error\_windows)，那么状态字(statusword)的 bit13(following\_error)将被置 1。



跟随误差(following error) – 举例

### 8.4.2、位置到达

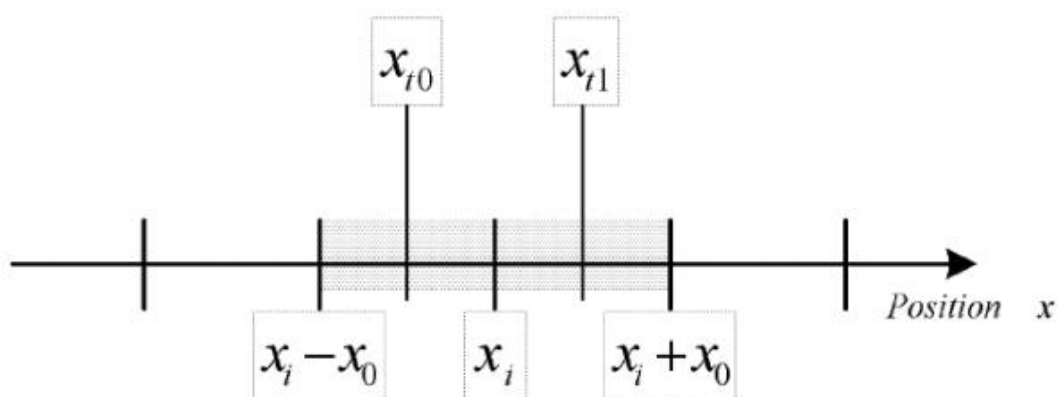
该功能定义了目标位置(target\_position)附近的位置窗口。如果驱动器的实际位置稳定在这个范围(位置窗口)达到设定时间(position\_windows\_time)，那么状态字(statusword)的 bit10(target\_reached)将被置 1。如下图所示。



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  $x_{t0}$  和  $x_{t1}$  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\_windows\_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。



位置到达 - 举例

## 8.5、位置控制相关参数

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

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

<b>Name</b>	position_actual_value
<b>Object Code</b>	VAR

<b>Data Type</b>	INT32
<b>Access</b>	RO
<b>PDO Mapping</b>	YES
<b>Units</b>	position units
<b>Value Range</b>	----
<b>Default Value</b>	----

<b>Index</b>	<b>6065h</b>
<b>Name</b>	following_error_window
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32
<b>Access</b>	RW
<b>PDO Mapping</b>	YES
<b>Units</b>	position units
<b>Value Range</b>	0 – 7FFFFFFF h
<b>Default Value</b>	256

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

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

<b>Index</b>	<b>6067h</b>
<b>Name</b>	position_window
<b>Object Code</b>	VAR
<b>Data Type</b>	UINT32

---

<b>Access</b>	RW
<b>PDO Mapping</b>	YES
<b>Units</b>	position units
<b>Value Range</b>	----
<b>Default Value</b>	400

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

---

## 九、周期同步位置模式（ CYCLIC SYNC POSITION MODE ）

周期同步位置模式本应用在 EtherCAT 通讯上的，即工作模式给定为 9，但在某些时候，为了方便用户使用，省去行规位置模式的应答机制，因此，也支持在 can 总线上应用此模式。

使用周期同步位置模式，上位控制给定多少位置就执行多少位置，无需像 PP 模式的应答机制来位置控制。

在运行过程中，此模式也可以按需要调整运行的电机最大速度（6080\_00，数据类型为 UINT32）。

# 十、应用举例

## 10.1、控制速度环

(1)、对 OperationMode (索引 0x6060) 写 3, 表示选择速度环, 并且可以通过 DisplayMode (索引 0x6061) 观察到当前工作模式;

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
601	2F	60	60	00	03

(2)、假如当前控制字 ControlWord 低 4 位为 0, 则依次对该控制字低 4 位写 6,7,15, 表示开使能。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	06	00

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	07	00

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	0F	00

(3)、对 TargetVelocity (索引 0x60FF) 写需要控制的转速, 单位: 默认 转/分钟。可以通过 VelocityActualValue (索引 0x6069) 观察当前电机的速度反馈值, 可以通过 VelocityDemandValue (索引 0x606B) 观察到给定转速的大小。

上位机设备发送, 往驱动器写 200 转/分的速度

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	23	FF	60	00	C8	00	00	00



## 速度模式

1.运行模式 16#6060

### Value Description

- 128...-2 Reserved
- 1 No mode
- 0 Reserved
- 1 Profile position mode
- 2 Velocity (not supported)
- 3 Profiled velocity mode**
- 4 Torque profiled mode
- 5 Reserved
- 6 Homing mode
- 7 Interpolated position mode
- 8...127 Reserved

2.状态字 16#6041

3.控制字 16#6040

## 流程控制

16 # 0006



16 # 0007



16 # 000F



使能



速度给的 16#60FF

## 10.2、控制位置环

### 10.2.1、绝对或相对位置控制模式

(1)、对工作模式 OperationMode (索引 0x6060) 写 1, 表示选择行规位置环, 并且可以通过 DisplayMode (索引 0x6061) 观察到当前工作模式;

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
601	2F	60	60	00	01

(2)、对该参数 ProfileVelocity (索引 0x6081) 控制位置环速度, 表示位置启动后, 完成加速后最终到达的速度;

上位机设备发送, 往驱动器写 800 转/分的速度

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	23	81	60	00	20	03	00	00

(3)、假如当前控制字 ControlWord 低 4 位为 0, 则依次对该控制字低 4 位写 6,7,15, 表示开使能。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	06	00

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	07	00

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	0F	00

(4)、对 TargetPosition (索引 0x607A) 写需要转到的目标位置, 单位: 默认 脉冲个数, 可以通过 PositionFdb (索引 0x6064) 观察当前电机的位置反馈值,

上位机设备发送, 假设给定绝对位置 10000 个脉冲

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	23	7A	60	00	10	27	00	00

(5)、把控制字 ControlWordbit4 置 1 表示设置一个新的位置, 上位机读取状态字 bit12 为 1 表示应答成功后, 控制字 bit4 再清零 0, 然后驱动器就开始执行规划。

上位机设备发送, 假设用绝对位置方式控制, 并且立即执行,

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	1F	00

## 10.2.2、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。

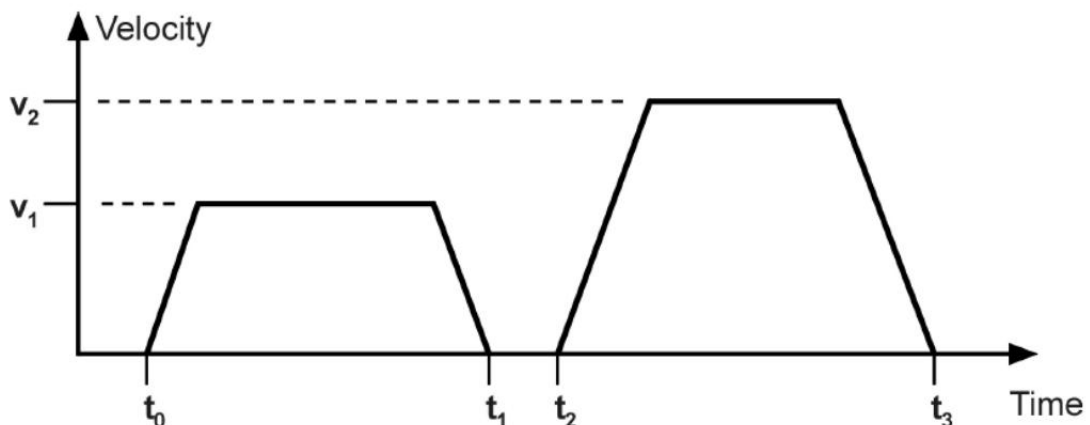
以上两种方法都可被控制字（controlword）的 bit4、bit5 和状态字 statusword 的 bit12（set\_point\_acknowledge）实时改变。

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。

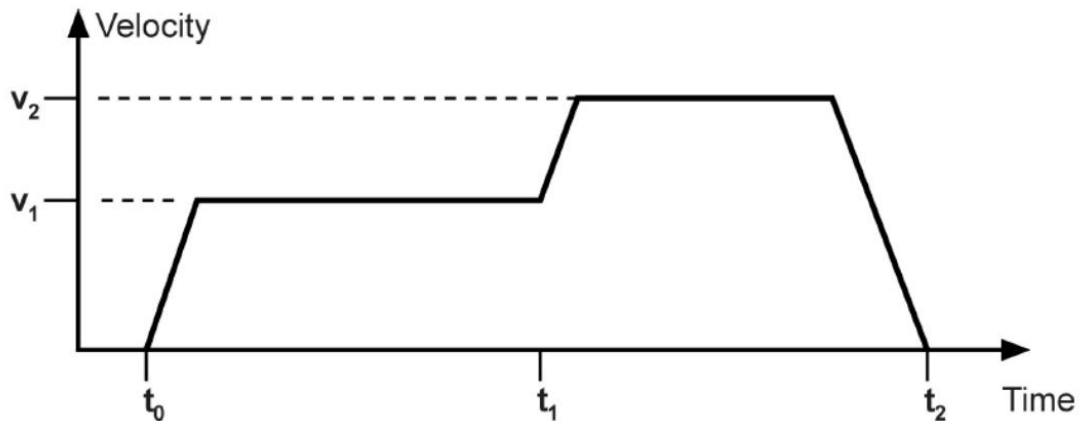


### 连续设定方法的步骤：

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。



注意：

- 1、 驱动器上电默认加速度等参数，如果用户没有设置加速度，则以默认参数控制，如果需要设置加速度，则必须在当前位置完成后，再设置，运行后设置无效。如果走连续位置，只设置了一个速度，那么后面的位置将以第一个速度为准。

## 位置模式

1.运行模式 16#6060

### Value Description

-128...-2 Reserved

-1 No mode

0 Reserved

**1 Profile position mode**

2 Velocity (not supported)

3 Profiled velocity mode

4 Torque profiled mode

5 Reserved

6 Homing mode

7 Interpolated position mode

8...127 Reserved

2.状态字 16#6041

3.控制字 16#6040

## 流程控制

16 # 0006



16 # 0007



16 # 000F



使能



写入目标位置 16 # 607A (如 100000)



16 # 001F

## 10.3、控制电流环

(1)、对 OperationMode (索引 0x6060) 写 4, 表示选择电流环, 并且可以通过 DisplayMode (索引 0x6061) 观察到当前工作模式;

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
601	2F	60	60	00	04

(2)、假如当前控制字 ControlWord 低 4 位为 0, 则依次对该控制字低 4 位写 6,7,15, 表示开使能。

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	06	00

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	07	00

上位机设备发送

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	40	60	00	0F	00

(3)、对 TargetTorque (索引 0x6071) 写需要控制的力矩值, 单位: 默认 0.1 安培。可以通过 TorqueActualValue (索引 0x6077) 观察当前电机的力矩反馈值, 可以通过 CurrentFdb (索引 0x6078) 观察到当前电机的电流反馈值。

注意: 电流环模式是以全速运行模式运行的。

上位机设备发送, 给定 2 安培的电流

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
601	2B	71	60	00	14	00

---

1.运行模式 16#6060

**Value Description**

- 128...-2 Reserved
- 1 No mode
- 0 Reserved
- 1 Profile position mode
- 2 Velocity (not supported)
- 3 Profiled velocity mode
- 4 Torque profiled mode**
- 5 Reserved
- 6 Homing mode
- 7 Interpolated position mode
- 8...127 Reserved

2.状态字 16#6041

3.控制字 16#6040

**流程控制**

16 # 0006



16 # 0007



16 # 000F



使能



写入目标扭矩 16 # 6071 (如 50 就是 5% )



运行

## 10.4、参数保存

用户在修改完参数后，如果想让该参数再重上电后仍然有效，则可以向 SavePara（索引 0x1010，子索引是 01）写 0x65766173，驱动器收到指令后，会把驱动器参数保存到外部存储器中，保存时间需要约 2-3 秒。

名称	索引 (H)	子索引	属性	数据类型
保存参数	1010	01	RW	Uint32

例如通过 SDO 方式发送，假设节点 id 是 0x01

COBID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
601	22	10	10	01	73	61	76	65

## 10.5、设置位置齿轮比

默认电子齿轮比为 1，如果需要改动，可以通过 can 来配置或者通过驱动器自带调试软件改动，如果当前电子齿轮比为 2，位置给定（607A\_00）给定 1000 个脉冲，在运行后，电机实际会转到 2000 个脉冲位置。

$\text{electronic gear ratio} = (6093\_01) / (6093\_02);$

$\text{position\_internal\_TargetPosition\_value} = \text{TargetPosition\_value} * \text{electronic\_gear\_ratio};$

$\text{position\_actual\_internal\_value} = \text{position\_actual\_value} * \text{electronic\_gear\_ratio};$

## 10.6、设置给定速度比

默认给定速度比为 1，如果电机配置减速机，假设为 7:1，在速度环情况下，如果想通过 60FF\_00 写 200r/m，的速度，电机经过减速机后也转 200r/m 的速度。

$\text{SpeedRef\_Gain} = (6094\_01) / (6094\_02) = 7/1 = 7;$

## 10.7、设置反馈速度比

默认反馈速度比为 1，如果电机配置减速机，假设为 7:1，在速度环情况下，如果想通过 60FF\_00 写 200r/m，的速度，要使反馈的速度也为 200r/m，则需要设置反馈速度比。

$\text{SpeedFdbGain} = (6095\_01) / (6095\_02) = 1/7;$



## 附录

### 对象字典--- CiA402 device profile

Index/SubIndex	Name	Data Type	Attribute	Mappable?	Comment
1000/00	Device Type	UINT32	RO	NO	
1001/00	Error Register	UINT8	RO	NO	
1008/00	DeviceName	UINT32	RO	NO	
1009/00	HardwareVersion	UINT32	RO	NO	
100A/00	SoftwareVersion	UINT32	RO	NO	
1010/01	SavePara	UINT32	Sub0:RO Sub1:RW	NO	
1015/00	EmcyInhibitTime	UINT16	RW	NO	
1016/01	ConsumerHeartTime	UINT32	Sub0:RO Sub1:RW	NO	
1017/00	ProductHeartTime	UINT16	RW	NO	
1018/04	Identity Object	UINT32	RW	NO	
1400/01-1403/01	RPDO_COB_ID	UINT32	RW	NO	
1400/02-1403/02	RPDO_TransType	UINT8	RW	NO	
1600 1601 1602 1603	RPDO mapping parameters	UINT32	RW	NO	PDO mapping parameters Sub1-Sub4
1800/01-1803/01	TPDO_COB_ID	UINT32	RW	NO	
1A00 1A01 1A02 1A03	TPDO mapping parameters	UINT32	RW	NO	PDO mapping parameters Sub1-Sub4
22A2/00	Drive Temperature° C	INT16	RO	TxMap	
2000/00	FaultCode	UINT16	RO	TxMap	
2001/00	WorkState	UINT16	RO	TxMap	
2002/00	Motor Temperature° C	INT16	RO	TxMap	
2003/00	Drive Volt	INT16	RO	TxMap	V
2004/00	nWARN	UINT16	RW	NO	
2005/00	CurrentBase	INT16	RW	NO	
2006/00	LineEncoder	INT32	RW	NO	
2007/00	MotorType	UINT16	RW	NO	
2008/00	PolePairs	UINT16	RW	NO	
2009/00	CANOPEN_NodeID	UINT16	RW	NO	
200A/00	CANOPEN_Baudrate	UINT16	RW	NO	
200B/00	CurrentOverGate	INT16	RW	NO	

200C/00	SpeedBase	INT16	RW	NO	
---------	-----------	-------	----	----	--

Index/SubIndex	Name	DataType	Attribute	Mappable?	Comment
603F/00	ErrorCode	UINT8	RO	TxMap	
6040/00	ControlWord	UINT16	RW	RxMap	
6041/00	StatusWord	UINT16	RO	TxMap	
6502/00	supported_drive_mode	UINT32	RO	NO	
605A/00	Quick_stop_option_code	INT16	RW	NO	
605B/00	Shutdown_option_code	INT16	RW	NO	
605C/00	Disable_operation_option_code	INT16	RW	NO	
605D/00	Halt_option_code	INT16	RW	NO	
605E/00	Fault_reaction_option_code	INT16	RW	NO	
6060/00	OperationMode	INT8	RW	RxMap	
6061/00	DisplayMode	INT8	RO	TxMap	
6062/00	Position demand value	INT32	RO	TxMap	
6063/00	position_actual_internal_value	INT32	RO	TxMap	
6064/00	Position actual value	INT32	RO	TxMap	
6067/00	ReachPosition	UINT32	RW	NO	
6069/00	Velocity Sensor ActualValue	INT32	RO	TxMap	
606A/00	SensorSelect	INT16	RO	NO	
606C/00	VelocityActualValue	INT32	RO	TxMap	
606D/00	Velocity window	UINT16	RW	NO	
606E/00	Velocity window time	UINT16	RW	NO	
606F/00	Velocity threshold	UINT16	RW	NO	
6070/00	Velocity threshold time	UINT16	RW	NO	
6071/00	TargetTorque	INT16	RW	RxMap	
6072/00	Max_torque	INT16	RW	NO	
6073/00	Max Currentt	INT16	RW	NO	
6074/00	torque_demand	INT16	RO	TxMap	
6075/00	Motor_rated_current	UINT32	RW	NO	
6076/00	Motor_rated_torque	UINT32	RW	NO	
6077/00	TorqueActualValue	INT16	RO	TxMap	
6078/00	Currentt ActualValue	INT16	RO	TxMap	
607A/00	Position Target value	INT32	RW	RxMap	
607B/00	position_range_limit	INT32	RW	RxMap	ARRAY
607C/00	Home Offset	INT32	RW	NO	
607D/00	Software_position_limit	INT32	RW	RxMap	ARRAY
607E/00	Polarity	UINT8	RW	FALSE	
607F/00	max_profile_velocity	UINT32	RW	NO	
6080/00	max_motor_speed	UINT32	RW	NO	
6081/00	ProfileVelocity	UINT32	RW	RxMap	

6082/00	EndVelocity	UINT32	RW	RxMap	
6083/00	Profileacceleration	UINT32	RW	RxMap	
6084/00	profile_deceleration	UINT32	RW	RxMap	
6085/00	quick_stop_deceleration	UINT32	RW	RxMap	
6086/00	motion_profile_type	IINT16	RW	NO	
6087/00	torque_slope	UINT32	RW	NO	
608F/02	Position encoder resolution	UINT32	Sub 0:RO Sub 1, 2:RW	NO	
6093/02	Position factor	UINT32	Sub 0:RO Sub 1, 2:RW	NO	
6094/02	Velocity encoder factor	UINT32	Sub 0:RO Sub 1, 2:RW	NO	
6095/02	Velocity factor 1	UINT32	Sub 0:RO Sub 1, 2:RW	NO	
6097/02	Acceleration factor	UINT32	Sub 0:RO Sub 1, 2:RW	NO	
6098/00	Homing Method	INT8	RW	NO	
6099/01	High Homing Speed	INT32	RW	NO	
6099/02	Low Homing Speed	INT32	RW	NO	
60B1/00	velocity_offset	INT32	RW	NO	
60B2/00	torque_offset	INT16	RW	NO	
60F2/00	Positioning_option_code	UINT16	RW	NO	
60FD/00	Digital_Inputs	UINT32	RO	NO	
60FE/00	Digital_Outputs	UINT32	RO	NO	
60FF/00	TargetVelocity	INT32	RW	RxMap	

**Objects 0x1400 - 0x1403: Receive PDO communication parameter**

- Object description:

<b>Index</b>	<b>1400h - 1403h</b>
Name	Receive PDO Parameter
Object code	RECORD
Data type	PDO CommPar (object 0x20)
Category	Conditional: mandatory for each supported PDO

- Entry description:

<b>Sub-index</b>	<b>0</b>
Description	Number of entries
Entry category	Optional
Access	Read only
PDO mapping	No
Value range	UNSIGNED8
Default value	2

<b>Sub-index</b>	<b>1</b>
Description	COB-ID used by PDO
Entry category	Optional
Access	Read only
PDO mapping	No
Value range	UNSIGNED32
Default value	Index 1400h: 0x27F Index 1401h: 0x37F Index 1402h: 0x47F Index 1403h: 0x57F

---

<b>Sub-index</b>	<b>2</b>
Description	Transmission type
Entry category	Optional
Access	Read only
PDO mapping	No
Value range	UNSIGNED8
Default value	255

**Objects 0x1800 - 0x1803: *Transmit PDO communication parameter***

- Object description:

<b>Index</b>	<b>1800h - 1803h</b>
Name	Transmit PDO parameter
Object code	RECORD
Data type	PDO CommPar (object 0x20)
Category	Conditional: mandatory for each supported PDO

- Entry description:

<b>Sub-index</b>	<b>0</b>
Description	Number of entries
Entry category	Optional
Access	Read only
PDO mapping	No
Value range	UNSIGNED8
Default value	No

<b>Sub-index</b>	<b>1</b>
Description	COB-ID used by PDO
Entry category	Optional
Access	Object 1800h: Read/Write Object 1801h: Read/Write Object 1802h: Read/Write Object 1803h: Read/Write
PDO mapping	No
Value range	UNSIGNED32
Default value	Index 1800h: 0x400001FF Index 1801h: 0x400002FF Index 1802h: 0x400003FF Index 1803h: 0x400004FF

<b>Sub-index</b>	<b>2</b>
Description	Transmission type
Entry category	Optional
Access	Object 1800h: Read/Write Object 1801h: Read/Write Object 1802h: Read/Write Object 1803h: Read/Write
PDO mapping	No
Value range	0...240
Default value	0

<b>Sub-index</b>	<b>3</b>
Description	Inhibit time
Entry category	Optional
Access	Read/Write
PDO mapping	No
Value range	UNSIGNED16
Default value	0 (no inhibit time between messages)

<b>Sub-index</b>	<b>4</b>
Description	Reserved
Entry category	
Access	
PDO mapping	
Value range	
Default value	

<b>Sub-index</b>	<b>5</b>
Description	Event timer
Entry category	Optional
Access	Read/Write
PDO mapping	No
Value range	0 - not used UNSIGNED16
Default value	No