

F 系列伺服 CANopen 相关说明

第一章 产品信息

1.1 驱动器型号

FS 32 05 - V T * (***)
1 2 3 4 5 6 7

- 1: 表示控制器类型为: F 系列单轴驱动器;
2: 表示电源电压等级, 32 代表三相或单相交流 220V;
3: 表示此驱动器的额定输出电流等级, 单位为安培 (A);
4: 表示对应电机编码器类型, V: 2500ppr 增量式光电或磁编编码器;
 S: 2500ppr 省线增量式光电编码器或磁编编码器; A: 17 位或 23 位总线式编码器;
5: T 代表标准;
6: 选配的通讯功能, R: 配 RS-485 通讯, N: 配 CANopen 通讯, 缺省表示不带通讯功能;
7: 括号中的信息表示此驱动器软硬件上有特殊之处, 缺省表示无特殊之处。

1.2 性能参数

项目	说明
链路层协议	CAN 总线
应用层协议	CANopen 协议
CAN-ID 类型	标准 ID
波特率	125K、250K 、500K 、750K、1M
支持子协议	DS301、DS402
支持服务	SDO、PDO、NMT、EMCY、HeartBeat
PDO 传输类型	事件触发、同步触发、时间触发
PDO 个数	4 个 TPDO、4 个 RPDO, 使用同步模式最快 1ms
SDO 传输方式	加速传输
支持运行模式	1、 简表位置模式 2、 简表速度模式 3、 简表转矩模式 4、 回零模式 5、 插补模式 (部分支持)

第二章 配线及系统设置

2.1 配线

F 系列驱动器的 CN3 端子为 CANOpen 通信端口端子接口定义如表 2-1 所示。

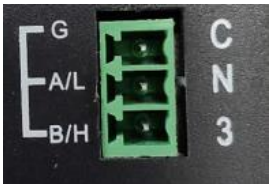


图 2-1 F 系列通讯口图片

针脚号	定义	描述
1	CANH	CAN 通信端口
2	CANL	
3	CGND	CAN 通信地

表 2-1 F 系列 CAN 通信连接端子定义

其他端口对应请参考 F 系列伺服说明书。

CAN 通信网络的连接方式为总线两节方式，CAN 总线上连接多个驱动器示意图如图 2-1 所示。

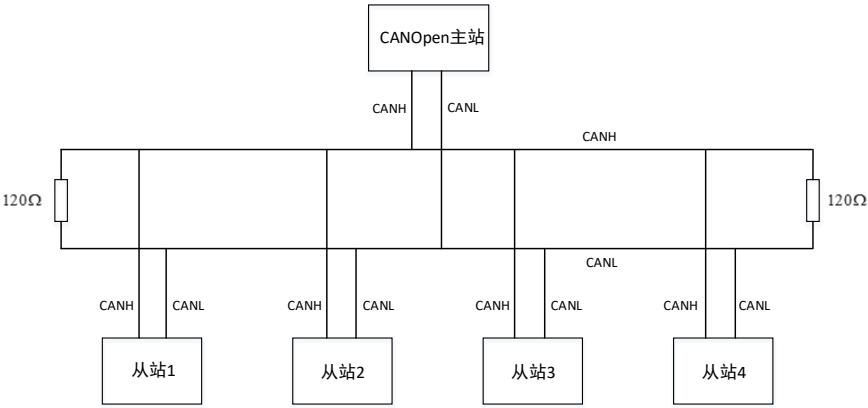


图 2-2 CANOpen 通信网络拓扑结构图

在组建 CAN 网络的时候，只需要把驱动器的 CANH 和 CANL 同总线的 CANH 和 CANL 相连即可。

注意：

- 必须保证总线 CANH 和 CANL 的两端分别接入 120 欧姆电阻。总线接好时可用万用表测试 CANH 和 CANL 之间的电阻，应该大约在 60 欧姆左右。否则会造成通讯失败。
- 总线电缆一般采用带屏蔽层的双绞线，屏蔽层直接接 PE 即可。
- F 系列使用三孔插头，可以通过在总线的 2 段插入色环电阻来方法来接入 120 欧姆。

2.2 系统设置

为了能够使驱动器准确的接入 CANOpen 现场总线网络，驱动器的相关功能码进行设置。设置如以下表所示。

功能码	名称	设定值
P0-09	运行模式	11
P0-82	CANOpen 站号	1-127
P0-84	Can 总线波特率	0: 125K 1: 250K 2: 500K 3: 1.0M
P0-85	Canopen 相关参数	0.按照老版本使用 rpm 和 ms 加速度 1.使用 PUU/S 和 PUU/S ²

表 2-3 F 系列系统设置功能码表

第三章 CANOpen 协议概述

CANopen 是一个基于 CAN 串行总线的网络传输系统的应用层协议，遵循 ISO/OSI 标准模型。CANOpen 的核心是对象字典，网络中不同的设备通过对象字典或者对象来相互交换数据，其中，主节点可以通过过程数据对象（PDO）或者服务数据对象（SDO）来获取或者修改其它节点对象字典列表中的数据。CANOpen 的设备模型如图 3-1 所示：

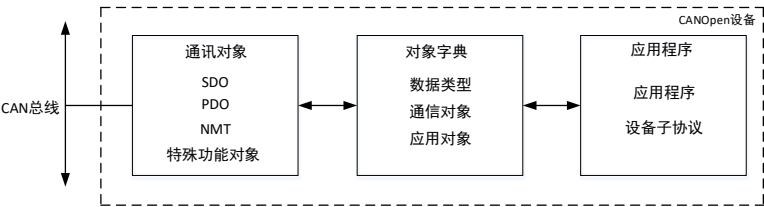


图 3-1 CANOpen 设备模型

3.1 对象字典

CANopen 的核心概念是设备对象字典（OD：Object Dictionary），它是一组参数和变量的有序集合。CANOpen 通讯通过对象字典够访问驱动器的所有参数。对象字典中的每个变量称之为对象，每个对象采用一个 16 位的索引值来寻址，为了允许访问数据结构中的单个元素，同时定义了一个 8 位的子索引，所以 CANOpen 通过 16 位的索引值和 8 位的自索引值来访问对象字典中的各个参数。对象字典的结构如表 3-1 所示：

索引	对象
000h	未使用
0001h—001Fh	静态数据类型(标准数据类型，如 Boolean、Integer16)
0020h—003Fh	复杂数据类型(预定义由简单类型组合成的结构如 PDOCommPar、SDOParmeter)
0040h—005Fh	制造商规定的复杂数据类型
0060h—007Fh	设备子协议规定的静态数据类型
0080h—009Fh	设备子协议规定的复杂数据类型
00A0h—0FFFh	保留
1000h—1FFFh	通信子协议区域（DS301）
2000h—5FFFh	制造商特定子协议区域（如功能码映射）
6000h—9FFFh	标准的设备子协议区域（如 DS402）
A000h—FFFFh	保留

表 3-1 CANOpen 对象字典结构表

如上所示，对象字典 1000h-1FFFh 索引区为通讯子协议区，它存放了常用的通讯参数；对象字典 2000-5FFFh 索引区为制造商定义对象区，它与 F 系列驱动器的功能码相对应；对象字典 6000h-9FFFh 索引区为设备子协议区，设备子协议（例如 DS402）规定

的参数都存放在这个区域。其中，F 系列驱动器参数和对象字典制造商规定子协议区索引（即 2000h-5FFh 区域）的对应关系如

下表 3-2：

对象字典索引	驱动器参数	举例
2000h 参数功能码	P0 参数	对象字典 2000-0Ah 对应 P0-10 参数
2001h 参数功能码	P1 参数	对象字典 2001-00h 对应 P1-00 参数
2002h 参数功能码	P2 参数	对象字典 2002-0Fh 对应 P2-15 参数
2003h 参数功能码	P3 参数	对象字典 2003-39h 对应 P3-57 参数

表 3-2 日鼎 F 系列驱动器参数和对象字典对应关系表

3.2 CANOpen 常用的通信对象

网络管理对象（NMT）

NMT 对象用于管理和监控网络中的各个节点，负责实现网络中各个节点的状态转换和错误控制。它是基于主/从模型的通信对象。

主要包括 Boot-up 报文，Heartbeat 报文以及 NMT 报文。

服务数据对象（SDO）

SDO 对象用来在设备之间传输低优先级数据，典型的是用来配置 CANOpen 网络上的设备。它通过 16 位索引+8 位子索引的方式来访问设备对象字典中的参数。SDO 对象是基于客户端/服务器模型的通信对象，客户端可以向服务器上传或下载对象字典参数，并且每一个 SDO 消息都要有应答。

过程数据对象（PDO）

PDO 对象用来传输一些实时数据，例如伺服运行过程中的实时速度或者实时位置。PDO 消息的内容是预定义的，该预定义的内容由 PDO 的映射参数决定。它是基于生产者/消费者模型的通信对象，因为 PDO 消息不需要应答而且消息的内容是预定义的，不需要通过索引和自索引对对象字典进行查找，所以 PDO 效率比 SDO 的通信效率高，适合用来传输实时数据。

同步对象（SYNC）

同步对象是 CANOpen 主站周期性地广播到总线上的报文，用来实现整个网络的一个时钟基准。从站根据自己的配置，来相应的执行同步事件。同步对象是基于生产者/消费者结构的通信对象，从站收到同步报文以后不需要进行回复。

紧急报文对象（EMCY）

当设备内部出现通信错误或者应用错误时，会向总线发送 EMCY 报文。EMCY 报文是基于生产者/消费者模型的。

3.3 通信对象标识符分配

通信对象标识符（COB-ID）是识别每一个通信对象的唯一符号。在本产品中，采用了 DS301 预定义的标识符的 ID 分配方案，即 COB-ID 和 11 位的 CAN-ID 对应，其中前 4 位表示对象功能代码，后 7 位表示节点 ID 地址，如表 3-3 所示：

bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
功能码				节点 ID						

表 3-3 预定义标识符 ID 定义表

CANopen 预定义连接集定义了 4 个接收 PDO (RPDO)，4 个发送 PDO (TPDO)，1 个服务器 SDO (包含发送和接收)，1 个 EMCY 对象和 1 个 NMT 错误控制对象。同时也支持不需确认的 NMT 对象以及 SYNC 同步对象。如下表所示：

通信对象	功能代码	节点地址	COB-ID	通信对象索引
NMT	0000b	0	0h	-
SYNC	0001b	0	80h	1005h, 1006h
EMCY	0001b	1-127	80h+Node-ID	1014h
TPDO1	0011b	1-127	180h+Node-ID	1800h
RPDO1	0100b	1-127	200h+Node-ID	1400h
TPDO2	0101b	1-127	280h+Node-ID	1801h
RPDO2	0110b	1-127	300h+Node-ID	1401h
TPDO3	0111b	1-127	380h+Node-ID	1802h
RPDO3	1000b	1-127	400h+Node-ID	1402h
TPDO4	1001b	1-127	480h+Node-ID	1803h
RPDO4	1010b	1-127	500h+Node-ID	1403h
SDO (发送)	1011b	1-127	580h+Node-ID	1200h
SDO (接收)	1100b	1-127	600h+Node-ID	1200h
NMT 错误控制	1110b	1-127	700h+Node-ID	1016h, 1017h

表 3-4 CANopen 预定义连接集标识符分配表

3.4 网络管理系统 (NMT)

CANopen 中的网络管理系统负责初始化、启动及停止网络及网络中的设备，属于主 / 从系统。CANopen 网络中有且只有一个 NMT 主机。

3.4.1 NMT 服务

状态机转换

从站的 NMT 状态机由主站通过 NMT 报文来改变也可以通过从站自身的状态改变来触发状态机转换。在 NMT 状态机不同的状态里

边，报文有不同的通讯权限。DS301 规定的状态机如图 3-2 所示：

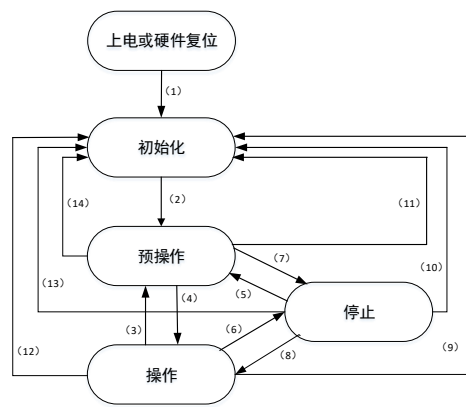


图 3-2 NMT 状态机转移图

图 3-2 中，触发 NMT 状态机转换的条件如表 3-5 所示：

(1)	上电之后自动进入初始化状态
(2)	初始化完成，自动进入预操作状态
(3)，(6)	启动远程节点指令
(4)，(7)	进入预操作状态指令
(5)，(8)	停止远程节点指令
(9)，(10)，(11)	复位节点指令
(12)，(13) (14)	复位通信指令

表 3-5 NMT 状态机转表

以上过程中，除了（1），（2）是自动完成的之外，其他过程都需要 NMT 主机（一般是主站）发送 NMT 报文来实现从站状态机的状态转换。NMT 的报文格式如表 3-6 所示：

COB-ID	Byte0	Byte0
0x000	命令字	节点 ID

表 3-6 NMT 报文格式

NMT 报文的 COB-ID 为 0x000。数据区第一个字节表示命令字，命令字的具体含义如表 3-7 所示。数据区第二个字是从站的节点地址，当该值为 0 时，表示这是一个广播信息，该命令对网络中的所有节点有效。

命令字	含义
0x01	启动远程节点指令

0x02	停止远程节点指令
0x80	进入预操作状态指令
0x81	复位节点指令
0x82	复位通信指令

表 3-7 NMT 报文命令字

状态含义

初始化状态

设备上电之后会自动进入初始化状态，包括正在初始化、 复位节点和复位通信。 正在初始化将各个模块及系统参数初始化，而复位节点将对象字典制造商定义区和标准子协议区的参数恢复到默认值， 复位通信将对象字典中通讯子协议区的参数恢复到默认值。

预操作状态

初始化完成以后，设备向总线发送 Boot-up 报文表示自己上线，然后自动进入了预操作状态。在预操作状态下主要完成配置节点的工作，例如完成 PDO 的配置。在预操作状态下 PDO 报文无效，SDO 报文有效。

操作状态

在收到 NMT 启动节点命令之后，从站进入操作状态。操作状态是 CANOpen 的正常工作状态，各个报文具有效。

停止状态

当收到 NMT 停止节点命令后，设备进入停止状态，在此状态下，SDO 和 PDO 均无效，只有 NMT 报文可以正常工作。

不同状态下支持的 CANOpen 通信对象如表 3-8 所示：

通信对象	预操作状态	操作状态	停止状态
NMT	是	是	是
SDO	是	是	否
PDO	否	是	否
SYNC	是	是	否
EMCY	是	是	否
NMT 错误控制	是	是	是

表 3-8 不同 NMT 状态下支持的 CANOpen 通信对象

3.4.2 NMT 错误控制

NMT 错误控制主要用于检测网络中的设备是否在线和设备所处的状态， 包括节点保护、 和心跳。但是，节点保护和心跳在一

个设备上不能同时存在。目前 F 系列驱动器只支持心跳协议。

心跳报文采用生产者/消费者模型。CANOpen 设备可根据生产者心跳间隔对象 1017h 设置的周期来发送心跳报文，单位为 ms。

消费者根据对象字典中 1016h 设置的消费者时间监视该生产者，一旦在消费者心跳时间范围内未接收到相应节点的生产者心跳，则认为该节点出现故障。配置生产者心跳时间间隔 1017h 后，节点开始产生心跳报文。当一个节点启动后它的 Boot-up 报文是它的第一个心跳报文。心跳报文消费者通常是 NMT 主节点，从机每隔 1017h 时间发送心跳报文，监视从机的主机(或其他从机)，在消费者时间内未接收到心跳报文，则认为该从机掉线。心跳报文的格式如表 3-9 所示：

COBID	Byte0
0x700+Node-ID	状态字

表 3-9 心跳报文格式

状态字的内容如表 3-10 所示：

状态字	含义
0	Boot-up
4	Stopped
5	Operational
127	Pre-operational

表 3-10 心跳报文状态字定义表

3.5 服务数据对象（SDO）

服务数据对象用来配置一些非实时的数据，它通过 16 位的索引值和 8 位的子索引值来访问设备的对象字典。SDO 传输方式遵循客户端/服务器模型，SDO 客户端发起报文请求，SDO 服务器作出应答。

SDO 的传输分为不高于 4 个字节和高于 4 个字节的对象数据传输。不高于 4 个字节采用加速 SDO 传输方式，高于 4 个字节采用分段传输或块传输方式。目前，日鼎驱动器只支持加速 SDO 传输方式。SDO 的报文格式如表 3-11 和表 3-12 所示：

COB-ID	Byte0	Byte1	Byte2	Byte3	Byte4-7
0x580+Node-ID	命令字	索引		子索引	数据

表 3-11 发送 SDO 报文格式

COB-ID	Byte0	Byte1	Byte2	Byte3	Byte4-7
0x600+Node-ID	命令字	索引		子索引	数据

表 3-12 接收 SDO 报文格式

所有的 SDO 报文数据段必须是 8 个字节，而且数据段采用小端模式，即低位在前，高位在后。例如，SDO 的 Byte0 和 Byte1 表示索引 1600h 的时候，Byte0 的内容是 00h，Byte1 的内容是 16h。与此类似，后边的 Byte4-7 表示的实际数据也是采用小端模式。Byte0 的命令码决定了该 SDO 的作用，表示该报文是读还是写或者是终止 SDO 传输。

3.5.1 SDO 加速写报文

如果要向设备写一个不大于 4 个字节的数据，则采用加速 SDO 写报文。SDO 加速写报文的格式符合标准 SDO 的格式，其命令字格式如表 3-13 所示：

Bit	7	6		5	4	3	2	1	0
客户端 →	0	0		1	—	n		e	s
← 服务器	0	1		1	—	—	—	—	—

表 3-13 SDO 加速写报文命令字

其中：

n：如果 e=1，且 s=1，则有效，表示数据中无意义数据的字节数（字节 8-n 到字节 7 无意义）。否则，n 为 0。

e：0 表示正常传送，1 表示加速传送。

s：是否指明数据长度，0 表示数据长度未指明，1 表示指明了数据长度。

SDO 客户端加速写报文以及 SDO 服务器的回复报文如表 3-14 所示：

	COB-ID	0	1	2	3	4	5	6	7
客户端 →	600h+Node-ID	23h	索引	子索引	4 个字节有效				
		27h			3 个字节有效				
		2bh			2 个字节有效				
		2fh			1 个字节有效				
← 服务器	580h+Node-ID	60h	索引	子索引	—				

表 3-14 SDO 加速写报文格式

举例：

例如从站的站号为 2，现在要利用 SDO 把简表位置模式下的目标位置（索引值 607Ah，子索引 00h）设置为 10000，即 2710h。

该变量为 INT32 类型的变量，长度为 4 个字节。则主站发送的 SDO 加速写报文如下（表中数据均为 16 进制）。

COB-ID	0	1	2	3	4	5	6	7
602	23	7A	60	00	10	27	00	00

若写入正确，驱动器返回报文格式如下：

COB-ID	0	1	2	3	4	5	6	7
582	60	7A	60	00	00	00	00	00

3.5.2 SDO 加速读报文

如果要从设备读一个不大于 4 个字节的数据，则采用 SDO 加速读报文。SDO 加速读报文的格式符合标准 SDO 的格式，其命令字格式如表 3-15 所示：

Bit	7	6	5	4	3	2	1	0
客户端 →	0	1	0	—	—	—	—	—
← 服务器	0	1	0	—	n		e	s

表 3-15 SDO 加速写报文命令字

其中：n，e，s 的定义和 SDO 加速写报文相同。

SDO 客户端加速读报文以及 SDO 服务器的回复报文如表 3-16 所示：

	COB-ID	0	1	2	3	4	5	6	7
--	--------	---	---	---	---	---	---	---	---

客户端 →	600h+Node-ID	40h	索引	子索引	—	—	—	—
← 服务器	580h+Node-ID	43h	索引	子索引	4 个字节有效			
		47h			3 个字节有效			
		4bh			2 个字节有效			
		4fh			1 个字节有效			

表 3-16 SDO 加速读报文格式

举例：

例如从站的站号为 2，现在要利用 SDO 读取简表位置模式下的当前位置的实际值（索引值为 6064h，子索引值为 00h），则从站发送的 SDO 加速读报文如下（表中数据均为 16 进制）。

COB-ID	0	1	2	3	4	5	6	7
602	40	64	60	00	00	00	00	00

假如此时电机的实际位置为 10000，即 2710h，因为当前位置的实际值为 INT32 类型变量，数据长度为 4，驱动器回复主站的 SDO 报文格式如下：

COB-ID	0	1	2	3	4	5	6	7
582	43	64	60	00	10	27	00	00

3.5.3 SDO 中止报文

在 SDO 报文的读写过程中，一旦发生错误，SDO 客户端或者服务器便通过发送 SDO 中止报文来中止 SDO 传送。SDO 中止报文的命令字格式如表 3-17：

Bit	7	6	5	4	3	2	1	0
客户端/服务器	1	0	0	—	—	—	—	—

表 3-17 SDO 中止报文命令字

SDO 中止报文的格式如表 3-18 所示：

COB-ID	0	1	2	3	4	5	6	7
580h+Node-ID	80	索引		子索引	SDO 中止码			

表 3-18 SDO 中止报文

SDO 中止报文的数据部分传输的是 32 位的 SDO 中止代码，SDO 中止代码表请参考故障处理章节。

举例：

例如从站的站号为 2，现在要利用 SDO 把简表位置模式下的目标位置（索引值 607Ah，子索引 00h）设置为 10000，即 2710h。该变量为 INT32 类型的变量，长度为 4 个字节。若主站发送的 SDO 加速写报文如下（表中数据均为 16 进制）。

COB-ID	0	1	2	3	4	5	6	7
602	2f	7A	60	00	10	27	00	00

从报文可以看出写入的数据长度不匹配，则此时从站人回复以下 SDO 中止报文。

COB-ID	0	1	2	3	4	5	6	7
582	80	7A	60	00	10	00	07	06

3.6 过程数据对象（PDO）

过程数据对象（PDO）用来传输实时数据。由于 PDO 不需要应答，并且 PDO 改变设备的对象字典并不需要像 SDO 那样在对象字典中查找对象，所以 PDO 的传输速度比较快。PDO 的传输遵循生产者/消费者模型，一个生产者产生的 PDO 可以被多个消费者消费。

3.6.1 PDO 对象

PDO 分为 TPDO（发送 PDO）和 RPDO（接收 PDO）。和 PDO 密切相关的是 PDO 的两个参数：通信参数和映射参数，PDO 的通信参数决定了 PDO 的通讯属性，包括 PDO 的 COBID，传输类型，禁止时间，事件时间等。PDO 的映射参数决定了 PDO 的内容，对于 TPDO 来说是首先根据 PDO 映射参数找到该 PDO 映射的对象，然后再将该对象的值读入 PDO 中发到总线上，对于 RPDO 来说是每当接收到一帧 PDO 数据，就根据接收 PDO 的映射参数找到映射的对象，然后再将 PDO 中的数据写入该对象。目前，F 系列驱动器支持 4 个 TPDO 和 4 个 RPDO 如下表 3-19 所示：

名称		COB-ID	通信对象	映射对象
RPDO	1	200h+Node-ID	1400h	1600h
	2	300h+Node-ID	1401h	1601h
	3	400h+Node-ID	1402h	1602h
	4	500h+Node-ID	1403h	1603h
TPDO	1	180h+Node-ID	1800h	1A00h
	2	280h+Node-ID	1801h	1A01h
	3	380h+Node-ID	1802h	1A02h
	4	480h+Node-ID	1803h	1A03h

表 3-19 PDO 对象列表

3.6.2 PDO 通信参数

PDO 的通信参数定义了 PDO 的通信属性，包括 PDO 的 COB-ID，PDO 的传输类型，禁止时间以及事件时间。PDO 通信参数的定义如下。

PDO 通信参数定义如表 3-20 所示：

子索引	含义
0h	支持的子索引个数
1h	COB-ID
2h	传输类型
3h	禁止时间
4h	保留
5h	事件时间

表 3-20 PDO 通信参数定义

PDO 的 COB-ID

PDO 的 COB-ID 位于该 PDO 通信参数的子索引 01h 上，它的结构如表 3-21 所示：

bit	值	含义
31	0	PDO 有效
	1	PDO 无效
30	0	允许使用远程帧
	1	不允许使用远程帧
29	0	表示是 11 位的标准 ID
	1	表示是 29 位的扩展 ID
28-11	0	如果是标准 ID，28-11 位为 0
	x	如果是扩展 ID，表示 29 位 ID 的 28-11 位
10-0	x	表示 COBID 的 10-0 位

表 3-21PDO COB-ID 定义表

PDO 处于无效状态时，即使配置了了 PDO，该 PDO 也不可用。目前，F 系列驱动器只支持标准的 11 位 CAN-ID，不支持扩展的 29 位 COB-ID。

PDO 的传输类型

PDO 的传输类型位于该 PDO 通信参数的子索引 02h 上。它决定了触发 TPDO 传输的方法或者处理 RPDO 的方法。具体对应关系如表 3-22 所示：

通信类型数值	同步		异步
	循环	非循环	
0		是	
1-240	是		
241-253	—		
254、255			是

表 3-22 PDO 传输方式

- 1) 当 TPDO 的传输类型为 0 时，如果映射参数发生改变，并且收到一个同步帧，则发送该 PDO，此模式暂时不支持
- 2) 当 TPDO 的传输类型为 1-240，接收到相应个数的同步帧时，发送该 TPDO。
- 3) 当 TPDO 的传输类型为 254 或 255 时，映射数据发生改变或者事件计时器到达则发送该 TPDO。
- 4) 当 RPDO 的传输类型为 0-240 时，只要接收到一个同步帧，则将该 RPDO 最新的数据更新到应用；当 RPDO 的传输类型为 254 或者 255 时，将接收到的数据直接更新到应用。

禁止时间

PDO 的禁止时间位于通信参数子索引 03h 上，禁止时间相同 COB-ID 的 PDO 在总线上传输的最小间隔时间，它可以防止总线被优先级较高的 PDO 持续占用，造成其他 PDO 或者通信对象无法传输。禁止时间的单位是 100us。禁止时间对 RPDO 来说没有作用。禁止时间一般配合 TPDO 的异步传输类型使用。

事件时间

针对异步传输（传输类型为 254 或 255）的 TPDO，定义事件计时器，它位于通信参数的自索引 05h 上。事件计时器定时事件到也会触发 PDO 的传输。如果在计时器运行周期内出现了数据改变等其它事件，TPDO 也会触发，并且事件计数器会被立即复位。

事件时间对 RPDO 来说没有作用。

3.6.3 PDO 映射参数

PDO 的映射参数表示了 TPDO 将从哪个对象读取参数以及 RPDO 将要数据写入哪个对象中。PDO 映射参数由映射对象的索引，子索引以及映射对象的位长度组成。每个 PDO 数据长度最多可达 8 个字节，可同时映射一个或者多个对象。其中子索引 0h 记录该 PDO 具体映射的对象个数，子索引 1-8 则是映射内容。映射参数内容定义如表 3-23 所示：

字节	0	1	2	3
含义	位长度	子索引	索引	

表 3-23 PDO 映射参数定义

索引和子索引共同决定对象在对象字典中的位置，对象长度指明该对象的具体位长，用十六进制表示。

举例：

例如 RPDO 映射控制字（索引值为 6040h），控制字为 UNSIGNED16 类型变量，位长度为 16，所以位长度字节位 10h，RPDO 映射控制字的映射参数为 60400010h。

CANopen 一般在预操作状态时通过 SDO 来配置 PDO 的通信参数和映射参数，SDO 配置 PDO 的流程如下所示。

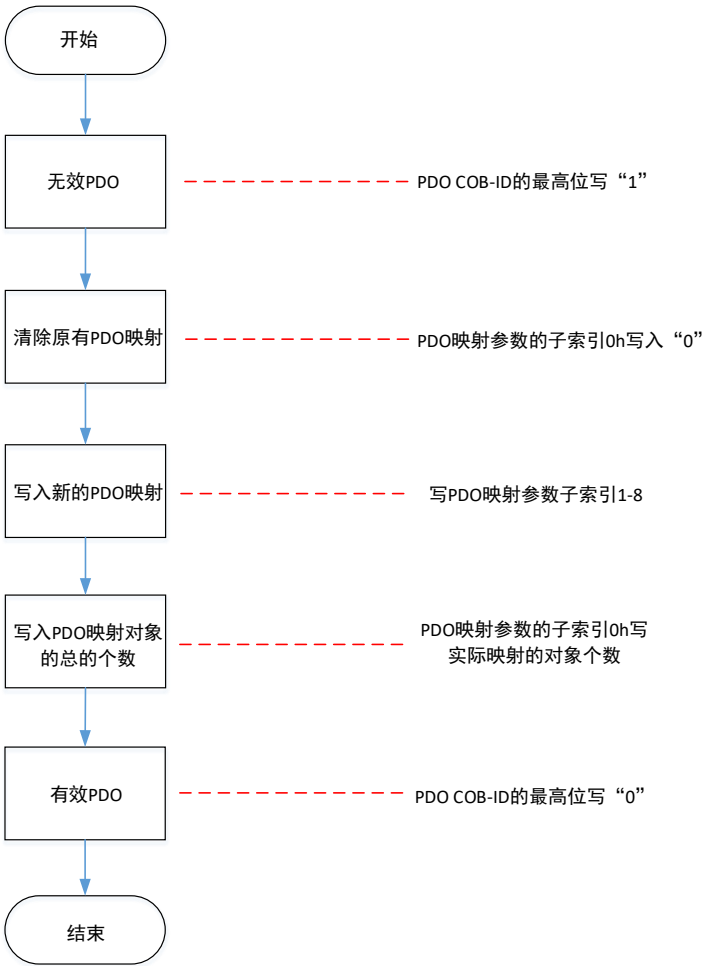


图 3-3 PDO 映射配置流程图

PDO 映射实例：

以 RPD01 为例，假设从站的站号为 2，RPD01 映射了三个对象如下所示：

名称	索引	子索引	长度
控制字	6040h	00h	2
运行模式	6060h	00h	1
目标位置	607A	00h	4

则映射的对象个数为 3，映射对象的数据总长度为 7 个字节，此时经过配置之后 RPD01 映射参数（索引 1600h）如下所示：

索引	子索引	值
1600h	00h	3
	01h	60400010h
	02h	60600008h
	03h	607A0020h

此时，如果设备从总线上收到一帧数据如下所示：

COB-ID	0	1	2	3	4	5	6	7
202	06	00	01	10	27	00	00	00

则映射之后的结果就如下所示：

索引	子索引	值
6040h	00h	6
6060h	00h	1
607Ah	00h	10000

TPDO 映射过程和 RPD0 类似，只不过方向相反。

3.7 同步对象（SYNC）

同步对象（SYNC）是控制多个节点发送与接收之间协调和同步的一种特殊机制，用于 PDO 的同步传输。与 PDO 的传输类似，同步对象的传输遵循的是生产者/消费者模型，由同步生产者发出同步帧，CAN 网络中的其它所有节点都可以作为消费者接收该同步帧，且无需反馈。同一个 CAN 网络中只允许有一个激活的同步发生器。目前，F 系列驱动器只作为同步对象的消费者，不产生同步帧。

同步 PDO 的传输与同步帧密切相关。

- 1) 对于同步 RPD0，只要接收到了该 PDO，在下一个同步对象到来时将接收到的 PDO 数据更新到应用之中。
- 2) 对于同步 TPDO，分为同步循环和同步非循环两种。

对于同步非循环类型（即传输类型为 0）TPDO 映射内容发生改变的时候，在收到下一个 SYNC 对象的时候发送出去。传输类型 0 暂时不支持

对于同步循环类型（即传输类型为 1-240），只要达到传输类型指定的 SYNC 时，不管数据有没有改变，均立刻发送该 PDO
同步对象的 COB-ID 固定为 0x80

举例：

RPD01 的传输类型为 0，RPD02 的传输类型为 10，TPD01 的传输类型为 0，TPD02 的传输类型为 10。则设备只要接收到 RPD01 和 RPD02 的数据， 会在下一个 SYNC 到来时将最新的 PDO 数据更新到相应的应用； 而 TPD01 的映射数据只有发生了改变， 会在下一个 SYNC 时发送 TPD01 ， TPD02 在连续收到 10 个 SYNC 时， 不管数据有无改变， 均会发送 PDO。

3.8 紧急对象（EMCY）

当 CANopen 节点出现错误时，节点会发送一帧紧急报文。紧急报文遵循的是生产者/消费者模型，节点故障发出后， CAN 网络中其它节点可选择处理该故障。伺服驱动器只作为紧急报文生产者， 不处理其它节点紧急报文。紧急报文的格式如表 3-24 所示：

COB-ID	0	1	2	3	4	5	6	7
80h+Node-ID	错误码		错误寄存器	保留	附加信息			

表 3-24 紧急报文格式

- 其中：
- 1) 错误寄存器和对象字典中的错误寄存器（1001h）相对应。
 - 2) 通信出现异常时，错误码和 DS301 所要求保持一致
 - 3) 驱动器出现 DS402 子协议描述错误时，错误码与 DS402 要求保持一致并与对象 603F 对应
 - 4) 驱动器出现用户指定的错误时，错误码与用户自动定义的错误代码相对应。

错误代码异警表请参考故障处理章节。

第四章 运动模式

4.1 设置转换因子

4.1.1 默认单位

驱动器的直接用户为电机，因此默认单位均为电机单位，如下：

- 电机位置单位：pul（脉冲）
- 电机速度单位：0.1 rpm（0.1 转 / 分）
- 电机加速度单位：从 0rpm 加速到额定转速的加速时间 ms

4.1.2 位置因子

位置因子实质意义为：负载位移为1 个用户单位时，对应的电机位移（单位：pul）。

对于 F 系列：

位置因子由分子 6091-1h 和分母 6091-2h 组成，通过位置因子可建立负载位移（用户单位）与电机位移（电机单位）的比例关系：

$$\text{电机位移} = \text{负载位移} \times \text{位置因子}$$

索引	6091 h
名字	位置因子(Gear ratio)
数据结构	ARRAY
元素数	2
数据类型	UINT32

子索引	01h
介绍	分子
可访问性	RW
能否映射	YES
单位	----
数据范围	不可设置成0
默认值	上电时，该值初始化成参数P0-01值

子索引	02h
介绍	分母
可访问性	RW
能否映射	YES
单位	----
数据范围	不可设置成0
默认值	上电时，该值初始化成参数P0-02值

电机与负载间通过减速机及其他机械传动机构连接。因此，位置因子与机械减速比、机械尺寸相关参数、电机分辨率相关。计算方法如下：

举例：

对于滚珠丝杠：

- 负载每次进给量 f_c : 40mm
- 丝杠导程: $p_B=10\text{mm/r}$
- 减速比 n : 1:5
- 日鼎默认 42 号电机装光电编码器时的分辨率 $P = 10000(p/r)$

因此，位置因子计算如下：

1) 负载轴每次进给量：

$$\text{负载轴进给量} = \frac{\text{负载进给量} f_c}{\text{导程} p_B} = \frac{40\text{mm}}{10\text{mm/r}} = 4(r)$$

2) 位置因子：

$$\text{位置因子} = \frac{\text{电机分辨率} P}{\text{负载轴进给量} \times \text{减速比} n} = \frac{10000}{4 \times (1/5)} = 500$$

因此：6091-1h=500，6091-2h=1。其实质意义为：负载位移为 1mm 时，电机位移为 500p。

4.1.3 注意

- 1) 由于默认的 P0-00 和 P0-02 并不会被初始化为 1:1，所以默认的位置因子也不是 1:1. 在使用位置控制的模式的时候要注意，记得使用前使用 SDO 设置，如果是运行中不修改齿轮比的常规用法，可以直接修改 P0-00 和 P0-02 的值，固定后不再通讯修改。F 系列伺服已经可以记录使用 SDO 写入的 6091-1h，6091-2h 的对象字典到 eeprom，修改完后必须重新上电，让伺服按照新的坐标系进行运行。
- 2) 日鼎 CANOpen 暂不支持除了位置因子以外的自定义的转换因子，所以用户暂时无法配置诸如：转速因子，加速度因子的字典项。在使用时请换算到上文提到的标准速度和加速度单位上来比较。

4.2 伺服状态控制

4.2.1 CiA402 伺服状态机

使用 RIDING-CANOpen 驱动器必须按照标准 402 协议规定的流程引导伺服驱动器，伺服驱动器才可运行于指定的状态。

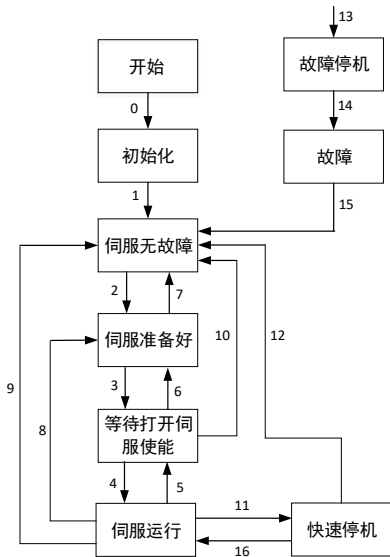


图 4-1 CiA402 状态机切换图

各状态的描述如下表：

状态	描述
初始化	驱动器初始化、内部自检已经完成。 驱动器的参数不能设置，也不能执行驱动功能。
伺服无故障	伺服驱动器无故障或错误已排除。驱动器参数可以设置。
伺服准备好	伺服驱动器已准备好，驱动器参数可以设置。
等待打开伺服使能	伺服驱动器等待打开伺服使能, 驱动器参数可以设置。
伺服运行	驱动器正常运行，已使能某一伺服运行模式，电机已通电，指令不为 0 时，电机旋转。 驱动器参数属性为“运行更改”的可以设置，其他不可。
快速停机	快速停机功能被激活，驱动器正在执行快速停机功能。 驱动器参数属性为“运行更改”的可以设置，其他不可。
故障停机	驱动器发生故障，正在执行故障停机过程中。 驱动器参数属性为“运行更改”的可以设置，其他不可。
故障	故障停机完成，所有驱动功能均被禁止，同时允许更改驱动器参数以便排除故障。 对于可复位故障，参数更改后，可通过控制字 6040h=0x80 使故障复位。

表 4-1 各状态描述

控制命令与状态切换：

CiA402 状态切换	控制字6040h	状态字6041h 的bit0~bit9*1
0 上电→初始化	自然过渡，无需控制指令	0x0000
1 初始化→伺服无故障	自然过渡，无需控制指令，若初始化中发生错误，直接进入 13	0x0250
2 伺服无故障→伺服准备好	0x06	0x0231
3 伺服准备好→等待打开伺服使能	0x07	0x0233
4 等待打开伺服使能→伺服运行	0x0F	0x0237
5 伺服运行→等待打开伺服使能	0x07	0x0233
6 等待打开伺服使能→伺服准备好	0x06	0x0231
7 伺服准备好→伺服无故障	0x00	0x0250
8 伺服运行→伺服准备好	0x06	0x0231
9 伺服运行→伺服无故障	0x00	0x0250
10 等待打开伺服使能→伺服无故障	0x00	0x0250
11 伺服运行→快速停机	0x02	0x0217
12 快速停机→伺服无故障	快速停机方式 605A 选择为 0~3，停机完成后， 自然过渡，无需控制指令	0x0250
13 →故障停机	除“故障”外其他任意状态下，伺服驱动器一旦发生故障，自动切换到故障停机状态，无需控制指令	0x021F
14 故障停机→故障故障	停机完成后，自然过渡，无需控制指令	0x0218
15 故障→伺服无故障	0x80 bit7 上升沿有效； bit7 保持为 1，其他控制指令均无效。	0x0250
16 快速停机→伺服运行	快速停机方式 605A 选择为 5~7，停机	0x0237

		完成后， 发送 0x0F	
--	--	-----------------	--

表 4-2 状态切换与控制命令关系

注意：

*1、因状态字 6041h 的 bit10~bit15(bit14 无意义) 与各伺服模式运行状态有关，在上表中均以“0”表示，具体的各位状态请查看各伺服运行模式。

4.2.2 控制字 6040h

索引	6040 h
名字	控制字(controlword)
数据结构	VAR
数据类型	UINT16
可访问性	RW
能否映射	YES
单位	----
数据范围	----
默认值	0

bit	名称		描述
0	伺服准备好		0- 无效 1- 有效
1	接通主回路电		0- 无效 1- 有效
2	快速停机		0- 无效 1- 有效
3	伺服运行		0- 无效 1- 有效
4~6	-		与各伺服运行模式相关
7	故障复位		对于可复位故障和警告，执行故障复位功能。 ■ bit7 上升沿有效；
8	暂停		暂不支持。
9 ~ 10	NA		预留
11 ~ 15	厂家自定义		预留，未定义

表 4-3 控制字对应位的控制指令：

◆ 注意：

■ 控制字的每一个 bit 位单独赋值无意义，必须与其他位共同构成某一控制指令。

■ bit0~bit3 和 bit7 在各伺服模式下意义相同，必须按顺序发送命令，才可将伺服驱动器按照 CiA402 状态机切换流程导入预计的状态，每一命令对应一确定的状态。

■ bit4~bit6 与各伺服模式相关（请查看不同模式下的控制指令）

4.2.3 状态字 6041h

索引	6041 h
名字	控制字 (Status Word))
数据结构	VAR
数据类型	UINT16
可访问性	RO
能否映射	YES
单位	-----
数据范围	-----
默认值	-----

步骤	操作	说明
0	伺服无故障	—
1	等待打开伺服使能	—
2	伺服运行	—
3	故障	—
4	接通主电回路	—
5	快速停机	—
6	伺服准备好	—
7	警告	—
8	厂家自定义	预留，未定义
9	远程控制	在 PN09 =10（CANOPEN 模式）的情况下直接为 1
10	目标到达	0- 目标位置或速度未到达 1- 目标位置或速度到达
11	软件内部位置超限	0- 位置指令或反馈未达到软件内部位置限制 1- 位置指令或反馈达到软件内部位置限制。
12~13		与各伺服模式相关
14	NA	预留
15	NA	NA

表 4-4 状态字对应位的显示的伺服状态

◆ 注意：

- 状态字的每一个 bit 位单独读取无意义，必须与其他位共同组成，反馈伺服当前状态。
- bit0~bit9 在各伺服模式下意义相同，控制字 6040h 按顺序发送命令后，伺服反馈一确定的状态。
- bit12~bit13 与各伺服模式相关（请查看不同模式下的控制指令）
- bit10、bit11、bit15 在各伺服模式下意义相同，反馈伺服执行某伺服模式后的状态。

4.2.4 停机方式

RIDING-CANOPEN 驱动器在不同一般通过端子等命令停机及故障停机外还增加有依靠通信停机的功能：非故障状态下，控制字 6040h=0x02 时，执行快速停机，停机方式通过对象字典 605A 选择。

605Ah:

索引	605C h
名字	快速停机方式选择(disable_operation_option_code)
数据结构	VAR
数据类型	INT16
可访问性	RW
能否映射	NO
单位	—
数据范围	0, 1
默认值	0

设置快速停机方式：

设定值	停机方式
0	自由停机，保持自由运行状态。
1	以 6085h 设定的减速度斜坡停机，停机完成后保持自由运行状态

4.3 试运行步骤

步骤	操作	说明
1	确认安装	按照附录要求进行安装（试运行前尽量不要把电机安装到机械上），具体可参考《RIDING-F 系列伺服用户手册》。
2	确认接线	按照配线章节将编码器线、电动力线、端子线等连接好，具体可参考《RIDING-F 系列伺服用户手册》。
3	确认电源电压	确保输入电源在驱动器规格要求内。
4	确认通信参数设置	确认系统设置。
5	确认电机型号	确保电机和驱动器型号匹配。
6	上电	确保上电无报警。
7	参数设置	设置相应参数，调节方法与普通 RIDING-F 系列驱动器相同。
8	试运行	进入轮廓速度模式，给定低速指令正反运行无异常，
9	参数调整	调整增益相关参数；观察响应是否满足要求
10	运行	—

4.4 伺服运行模式概述

Riding-CANOpen 支持 4 种伺服模式，对象字典 6502h 用于显示伺服驱动器支持的伺服模式。

支持伺服运行模式 6502h

索引	6502 h
名字	支持伺服运行模式(Supported Drive Modes)
数据结构	VAR
数据类型	UINT32
可访问性	RO
能否映射	Yes
单位	----

数据范围	-----
默认值	2D (6D) h

若 CANopen 设备支持对象字典 6502h，则可通过其了解驱动器支持的伺服模式。

bit	描述	支持与否 0- 不支持 1- 支持
0	pp(轮廓位置模式)	1
1	vl	0
2	pv(轮廓速度模式)	1
3	tq (轮廓转矩模式)	1
4	NA	0
5	hm(原点回零模式)	1
6	ip(插补模式)	0 (1)
7~15	NA	预留
16~31	厂家自定义	预留，未定义

若 CANopen 设备支持对象字典 6502h，则可通过其了解驱动器支持的伺服模式。

部分程序已经支持 ip 模式，使用前记得通过 6502 字典项查询是否支持

模式选择 6060h

索引	6060 h
名字	模式选择 (Modes of Operation)
数据结构	VAR
数据类型	INT8
可访问性	RW
能否映射	Yes
单位	-----
数据范围	0~7
默认值	0

选择伺服运行模式：

设置值	描述
0	NA
1	轮廓位置模式
2	NA
3	轮廓速度模式
4~5	NA
6	回零模式
7	插补模式

- 通过 SDO 选择了不支持的伺服模式，将返回 SDO 错误。
- 通过 PDO 选择了不支持的伺服模式，伺服模式更改无效。

模式显示 6061h:

索引	6061 h
名字	模式显示 (Modes of Operation)
数据结构	VAR
数据类型	INT8
可访问性	RO
能否映射	Yes
单位	----
数据范围	0~7
默认值	0

反映伺服实际运行模式:

显示值	描述
0	NA
1	轮廓位置模式
2	NA
3	轮廓速度模式
4~5	NA
6	回零模式
7	插补模式

模式切换使用注意事项:

- 伺服驱动器处于任何状态下，从轮廓位置模式切入其他模式后，轮廓位置模式未执行的位置指令将被抛弃。
- 伺服处于回零模式，且正在运行时，不可切入其他模式；回零完成或被中断（故障或使能无效）时，可切入其他模式。

4.5 轮廓位置模式

4.5.1 模式介绍

在此模式下，伺服驱动器收到外部控制器发出的位置指令信号，然后控制并且驱动伺服电机到达目标位置。

用户脉冲单位定义:

$$\text{用户脉冲单位 (PUU)} : \text{PUU/圈} = \text{每圈脉冲数} * \frac{(OD-6091h \text{ Sub2})}{(OD-6091h \text{ Sub1})}$$

$$\text{即: } 1 \text{ PUU} = \frac{(OD-6091h \text{ Sub1})}{(OD-6091h \text{ Sub2})} * 1 \text{ pulse} \quad \text{这里的 PUU 就对应于上面 4.1 节提出的}$$

4.5.2 本模式简要运行步骤

- ① 将字典项【运行模式:6060h】设置为轮廓位置模式（1）。
- ② 将字典项【目标位置:607Ah】设置为目标的位置（单位：PUU）。
- ③ 将字典项【轮廓速度:6081h】设置成运行过程中的目标速度（单位:rpm）。
- ④ 将字典项【轮廓加速度:6083h】设置成速度曲线的上升斜率。（单位：从 0rpm 加速到额定转速的加速时间 ms）。
- ⑤ 将字典项【轮廓减速度:6084h】设置成速度曲线的下降斜率。（单位：从 0rpm 加速到额定转速的加速时间 ms）。

- ⑥ 设置字典项【控制字:6040h】来使能并且让电机运行。
- ⑦ 查询字典项【实际位置值:6064h】来获取电机的反馈位置(单位: PUU) 。
- ⑧ 查询字典项【状态字:6041h】来获取跟踪误差达到, 能否接受新位置指令, 位置到达等信息。

4.5.3 本模式高级用法

1) 主机可以获得关于轮廓位置模式的更多的信息

- 通过查询字典项【位置给定值:6062h】获取内部的位置指令(单位: PUU)
- 通过查询字典项【位置实际值:6063h】来得到位置运行的实际反馈(单位: 脉冲)

2) 跟踪误差

- 设置字典项【用户位置偏差过大阈值:6065h】来设置能接受的最大位置跟踪误差(单位:PUU)
- 通过查询字典项【跟踪误差实际值:60F4h】来得到反馈误差的实际值(单位: PUU)

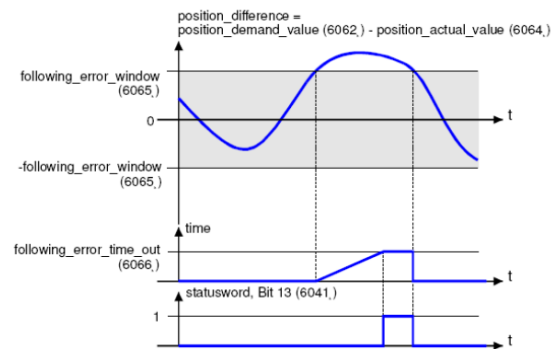


图 4-2 跟随误差功能描述

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。

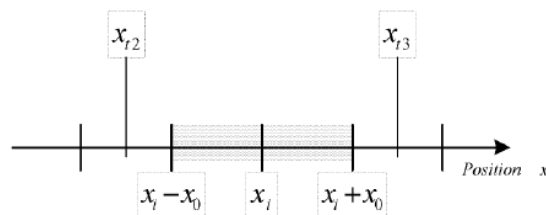


图 4-3 跟随误差举例

上图描述了对“following error”, 窗口功能是如何定义。 $x_i - x_0$ 和 $x_i + x_0$ 范围(位置跟随误差窗口)对称地分布在期望位置(position_demand_value)两边。举例, 位置 x_{t2} 和 x_{t3} 都位于位置跟随误差窗口外。如果驱动器离开窗口, 那么状态字(statusword)的bit13(following_error)将被置1。

3) 位置窗口

- 设置字典项【位置到达阈值 6067h】来定义一个在目标位置附近的串口
(单位 PUU)

- 设置字典项【位置到达时间窗口:6068h】来设定触发位置到达信号的时间(单位:毫秒)

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。

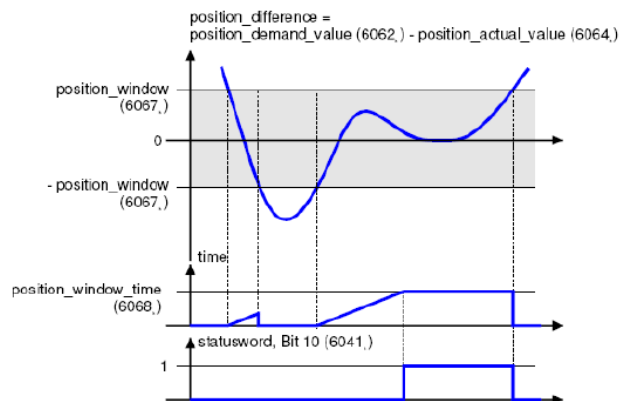


图 4-4 位置到达功能描述

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_window_time)且期间驱动器位置一直位于窗口内, 那么状态字(statusword)的 bit10(target_reached)将被置 1。驱动器位置一离开该窗口, 状态字的 bit10(target_reached)将立即被清零。

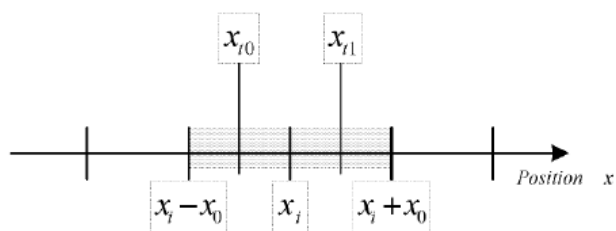


图 4-5 位置到达举例

4. 5. 4 与本模式相关的对象字典项

索引	名称	数据类型	存储属性
6040h	控制字	UINT16	RW
6041h	状态字	UINT16	RO
6060h	模式选择	INT8	RW
6061h	模式显示	INT8	RO
6062h	用户位置指令[PUU]	INT32	RO
6063h	电机位置反馈[PUU]	INT32	RO
6064h	用户位置反馈	INT32	RO
6065h	用户位置偏差过大阈值	UINT32	RW
6067h	位置到达阈值	UINT32	RW
6068h	位置到达时间窗口	UINT16	RW
607Ah	目标位置	INT32	RW
607Dh	软件绝对位置限制	ARRAY	RW
6081h	轮廓速度	UINT32	RW
6083h	轮廓加速度	UINT32	RW
6084h	轮廓减速度	UINT32	RW

6091h	位置因子（齿轮比）	ARRAY	RW
60F4h	用户位置偏差	INT32	RO
60FCh	电机位置指令	INT32	RO

6060, 6062 字典项的解释可见 4.4 节

控制字 6040h 在轮廓位置模式中的说明：

控制字 6040h					
位	bit7~15	bit6	bit5	bit4	bit0~bit3
名称	-	位置指令 类型	位置指令 更新模式	使能新位置指令 (沿变化有效)	-
设定值	请参考表4-2 状态切换 与控制命令关系	-	-	-	请参考表4-2 状态切换 与控制命令关系
描述	-	0- 目标位置607Ah 是绝对位置指令 1- 目标位置 607Ah 是相对位置 指令	0- 非立刻更新 1- 立刻更新	0 → 1 表示预使能一段新的 位移指令，是否使能成功，由 伺服状态决定；1 → 0 表示 预清零控制字 6041h 的 bit12，是否成功清零，由伺 服状态决定。	-

状态字 6041h 在轮廓位置模式中的说明：

状态字 6041h						
位	bit15-bit14	Bit13	Bit12	Bit11	Bit10	bit0~bit9
名称	NA	位置偏差状态	位置指令能够 接收	软件内部设置 超限	目标到达	-
设定值	NA	请参考表4-2 状态切换与控制 命令关系	-	-	-	请参考表4-2 状态切换与控制 命令关系
描述	NA	0- 位置偏差在位置 偏差过大阈值 (6065h) 1- 位置偏差超出 位置偏差过大阈 值 (6065h)	0- 伺服可接收新 的位移指令 1- 伺服不可接收 新的位移指令	0- 位置指令 未达到软件内 部位置 限制 (607Dh) 1- 位置指令 或位置反馈达 到软 件内部位置限 制	0- 目标位 置未到 达 1- 目标位 置到达	-

6062h:

索引	6062h
名字	用户位置指令(Position Demand Value)
数据结构	VAR
数据类型	INT32

可访问性	R O
能否映射	Y e s
单 位	P U U
数 据 范 围	$-2^{31} \sim (2^{31}-1)$
默 认 值	-----
介 绍	反映实时位置指令(用户单位)。

6063h:

索 引	6063h
名 字	电机位置反馈(Position Actual Value)
数 据 结 构	VAR
数 据 类 型	I N T 3 2
可访问性	R O
能否映射	Y e s
单 位	i n c r e m e n t
数 据 范 围	$-2^{31} \sim (2^{31}-1)$
默 认 值	-----
介 绍	反映实时电机绝对位置反馈。

6064h:

索 引	6064h
名 字	用户位置反馈(Position Actual Value)
数 据 结 构	VAR
数 据 类 型	I N T 3 2
可访问性	R O
能否映射	Y e s
单 位	P U U
数 据 范 围	$-2^{31} \sim (2^{31}-1)$
默 认 值	-----
介 绍	反映实时用户绝对位置反馈。 用户位置反馈(6064h) × 位置因子(6091h) = 电机位置反馈(6063h)

6065h:

索 引	6065h
名 字	用户位置偏差过大阈值(Following Error Window)
数 据 结 构	VAR
数 据 类 型	U I N T 3 2
可访问性	R W
能否映射	Y e s
单 位	P U U

数据范围	$0 \sim (2^{32}-1)$
默认值	50000
介绍	设置位置偏差过大阈值(用户单位)。

6067h:

索引	6067h
名字	位置到达阈值(Position Window)
数据结构	VAR
数据类型	UINT32
可访问性	RW
能否映射	Yes
单位	PUU
数据范围	$0 \sim (2^{32}-1)$
默认值	100
介绍	<p>设置位置到达的阈值。</p> <p>用户位置指令6062h 与用户实际位置反馈6064h 的差值在±6067h 以内, 且时间达到6068h 时, 认为位置到达, 轮廓位置模式下,</p> <p>状态字6041h 的bit10=1。</p> <p>轮廓位置模式, 伺服使能有效时, 此标志位有意义; 否则无意义。</p>

6068h:

索引	6068h
名字	位置到达时间窗口(Position Window Time)
数据结构	VAR
数据类型	UINT16
可访问性	RW
能否映射	Yes
单位	ms
数据范围	$0 \sim 65535$
默认值	0
介绍	<p>设置判定位置到达有效的时间窗口。</p> <p>用户位置指令6062h 与用户实际位置反馈6064h 的差值在±6067h 以内, 且时间达到6068h 时, 认为位置到达, 轮廓位置模式下,</p> <p>状态字6041h 的bit10=1。</p> <p>轮廓位置模式, 伺服使能有效时, 此标志位有意义; 否则无意义。</p>

607Ah:

索引	607Ah
----	-------

名 字	目标位置(Target Position)
数 据 结 构	VAR
数 据 类 型	INT32
可 访 问 性	RW
能 否 映 射	Yes
单 位	PUU
数 据 范 围	$-2^{31} \sim (2^{31}-1)$
默 认 值	0
介 绍	----

607Dh:

索 引	607D h
名 字	软件绝对位置限制(Software Position Limit)
数 据 结 构	ARRAY
元 素 数	2
数 据 类 型	INT32

子 索 引	01h
介 绍	最小软件绝对位置限制
可 访 问 性	RW
能 否 映 射	YES
单 位	----
数 据 范 围	----
默 认 值	上电时，该值初始化成参数PR76-PR77组成的左限位的值

子 索 引	02h
介 绍	最大软件绝对位置限制
可 访 问 性	RW
能 否 映 射	YES
单 位	----
数 据 范 围	----
默 认 值	上电时，该值初始化成参数PR78-PR79组成的右限位的值位的值

6081h:

索 引	6081 h
名 字	轮廓速度(Profile Velocity)
数 据 结 构	VAR
数 据 类 型	UINT32
可 访 问 性	RW
能 否 映 射	Yes
单 位	0.1rpm

数据范围	$0 \sim (2^{32}-1)$
默认值	0
介绍	设置轮廓位置模式下该段位移指令的匀速运行速度。 从站接收了该段位移指令后，设定值生效

6083h:

索引	6083h
名字	轮廓加速度(Profile Acceleration)
数据结构	VAR
数据类型	UINT32
可访问性	RW
能否映射	Yes
单位	Ms/0rpm 升到额定速度
数据范围	$0 \sim (2^{32}-1)$
默认值	0
介绍	设置轮廓位置模式下该段位移指令加速段的加速度

6084h:

索引	6084h
名字	轮廓减速度(Profile Deceleration)
数据结构	VAR
数据类型	UINT32
可访问性	RW
能否映射	Yes
单位	Ms/额定速度降到0速
数据范围	$0 \sim (2^{32}-1)$
默认值	0
介绍	设置轮廓位置模式下该段位移指令减速段的加速度。 从站接收了该段位移指令后，设定值生效。

60E0h:

索引	60E0h
名字	正向力矩限制(Positive torque limit)
数据结构	VAR
数据类型	UINT16
可访问性	RW
能否映射	Yes
单位	千分之一额定转矩
数据范围	$0 \sim (2^{16}-1)$
默认值	0
介绍	限制正向旋转最大力矩值

60E1h:

索引	60E1h
名字	反向力矩限制(Negative torque limit)
数据结构	VAR
数据类型	UINT16
可访问性	RW
能否映射	Yes
单位	千分之一额定转矩
数据范围	$0 \sim (2^{16}-1)$
默认值	0
介绍	限制反向旋转最大力矩值

60F4h:

索引	60F4h
名字	用户位置偏差(Following Error Actual Value)
数据结构	VAR
数据类型	INT32
可访问性	RO
能否映射	Yes
单位	PUU
数据范围	$-2^{31} \sim (2^{31}-1)$
默认值	0
介绍	反映实时位置偏差(用户位置单位)。

60FCh:

索引	60FCh
名字	电机位置指令(Position Demand Value)
数据结构	VAR
数据类型	INT32
可访问性	RO
能否映射	Yes
单位	PUU
数据范围	$-2^{31} \sim (2^{31}-1)$
默认值	0
介绍	反映电机实时位置指令 用户位置指令(6062h) × 位置因子(6091h) = 电机位置指令 60FCh

4.6 回零模式

4.6.1 模式介绍

此模式帮助驱动器寻找原点位置,用户可以选择回零方法,回零速度和回零加速度。

4.6.2 本模式简要运行步骤

- ① 将字典项【运行模式:6060h】 设置为回零模式（6）。
- ② 设置【用户原点偏置:607Ch】对象字典（单位：PUU）。
- ③ 将字典项【回原点方法:6098h】设置成希望的回零方法（范围 riding-CANOPEN 支持的有回零方法 3,4,5,6,24,28,34,35）。
- ④ 将字典项【回零速度:6099h Sub-1】设置成搜索减速点信号速度(单位：0.1rpm)。
- ⑤将字典项【回零速度:6099h Sub-2】设置成搜索原点信号速度(单位：0.1rpm)。
- ⑥将字典项【回零加速度:609Ah】设置成搜索原点信号加速度(单位：ms/从 0rpm 到额定转速)。
- ⑦设置字典项【控制字:6040h】来使能并且让电机运行。
- ⑧开始寻找原点开关并回零。
- ⑨查询字典项【状态字:6041h】来获取跟踪伺服运行状态。

4.6.3 回零模式

在这里重点讨论下 RIDING-CANOPEN 所支持的回零模式（6098h）的值：

6098h=3

- 原点：Z 信号
- 减速点：原点开关（HW）
- a) 回零启动时减速点信号无效

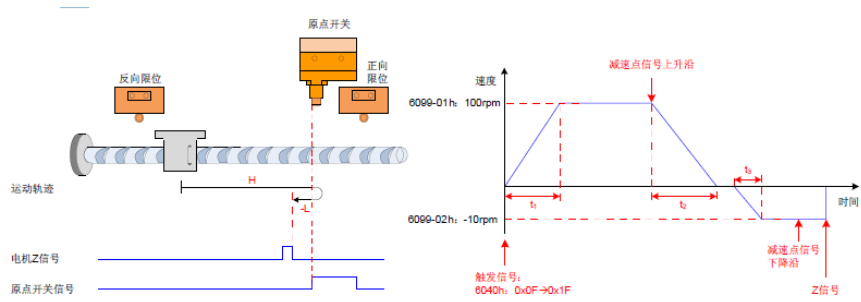


图 4-6 6098h=3 且减速点信号无效模式①

$$t1=\frac{6099-01h}{609Ah} \text{ ms} \quad t2=\frac{6099-01h}{609Ah} \text{ ms} \quad t3=\frac{6099-02h}{609Ah} \text{ ms}$$

开始回零时 HW=0，以正向高速开始回零，遇到 HW 上升沿后，减速→反向→反向低速运行，遇到 HW 下降沿后，继续运行，之后遇到第一个 Z 停机。

- b) 回零启动时减速点信号有效

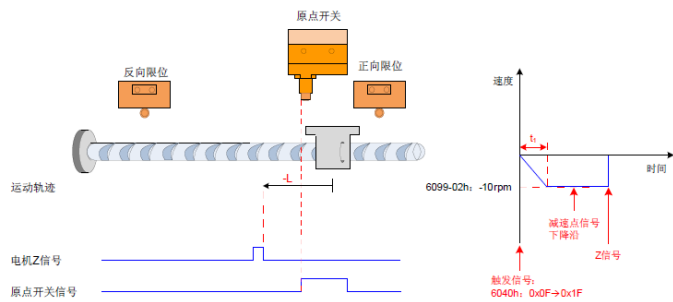


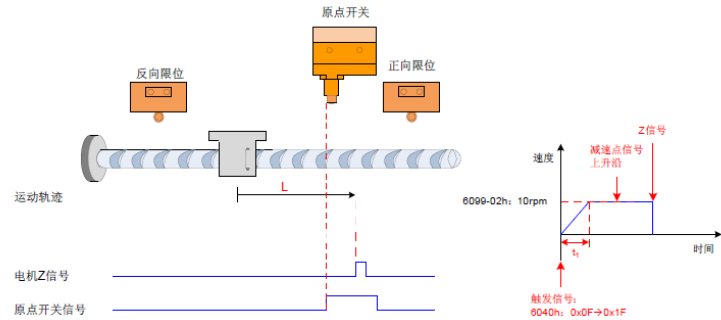
图 4-7 6098h=3 且减速点信号有效模式②

$$t1=\frac{6099-02h}{609Ah} \text{ ms}$$

回零启动时 HW=1，直接反向低速开始回零，遇到 HW 下降沿后的第一个 Z 停机。

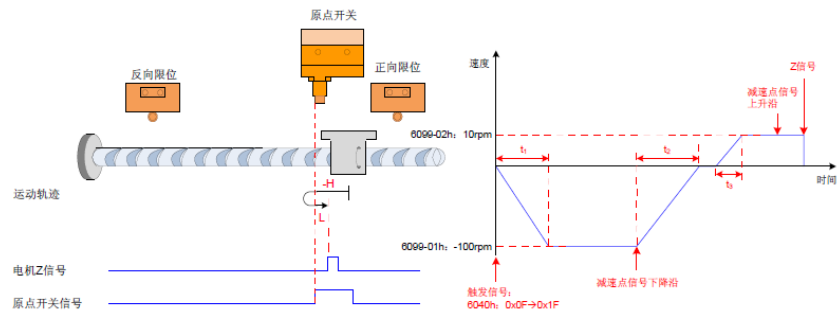
6098h = 4

- 原点：Z 信号
 - 减速点：原点开关 (HW)
- a) 回零启动时减速点信号无效



开始回零时 HW=0，直接正向低速开始回零，遇到 HW 上升沿后第一个 Z 停机。

- b) 回零启动时减速点信号有效



回零启动时 HW=1，以反向高速开始回零，遇到 HW 下降沿后，减速→反向→正向低速运行，遇到 HW 上升沿后的第一个 Z 停机。

6098h=5

- 原点：Z 信号
 - 减速点：原点开关 (HW)
- a) 回零启动时减速点信号无效

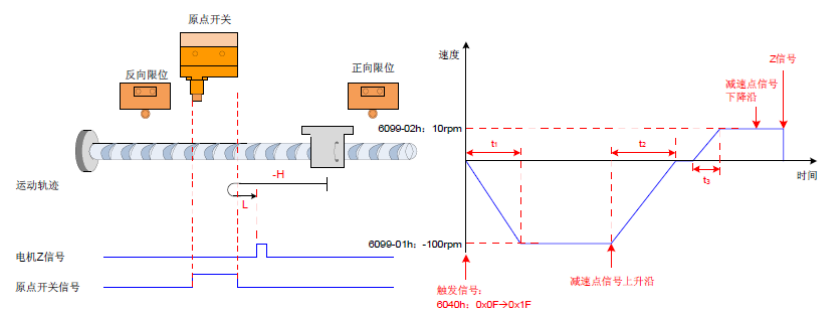


图 4-10 6098h=5 且减速点信号无效模式①

$$t1=\frac{6099-01h}{609Ah} \text{ ms} \quad t2=\frac{6099-01h}{609Ah} \text{ ms} \quad t3=\frac{6099-02h}{609Ah} \text{ ms}$$

开始回零时 HW=0，以反向高速开始回零，遇到 HW 上升沿后，减速→反向→正向低速运行，遇到 HW 下降沿后的第一个 Z 停机。

b) 回零启动时减速点信号有效：

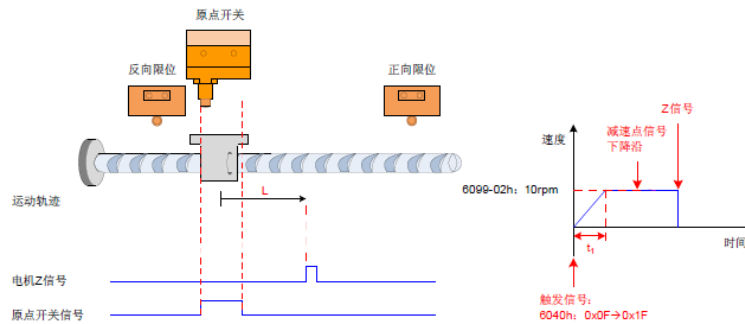


图 4-11 6098h=5 且减速点信号有效模式②

$$t1=\frac{6099-02h}{609Ah} \text{ ms}$$

回零启动时 HW=1，则直接正向低速开始回零，遇到 HW 下降沿后的第一个 Z 停机。

6098h=6

- 原点：Z 信号
- 减速点：原点开关 (HW)

a) 回零启动时减速点信号无效

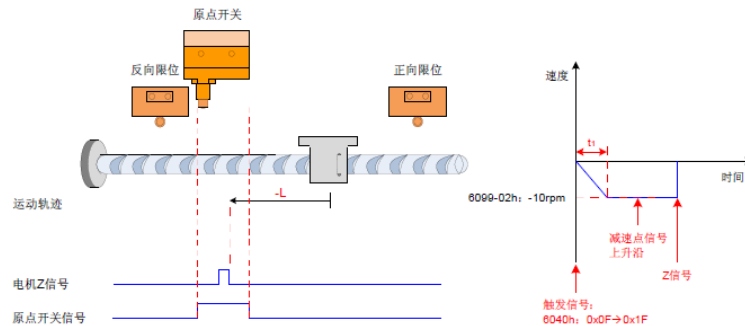


图 4-12 6098h=6 且减速点信号无效模式①

$$t1=\frac{6099-02h}{609Ah} \text{ ms}$$

开始回零时 HW=0，直接反向低速开始回零，遇到 HW 上升沿后第一个 Z 停机。

b) 回零启动时减速点信号有效：

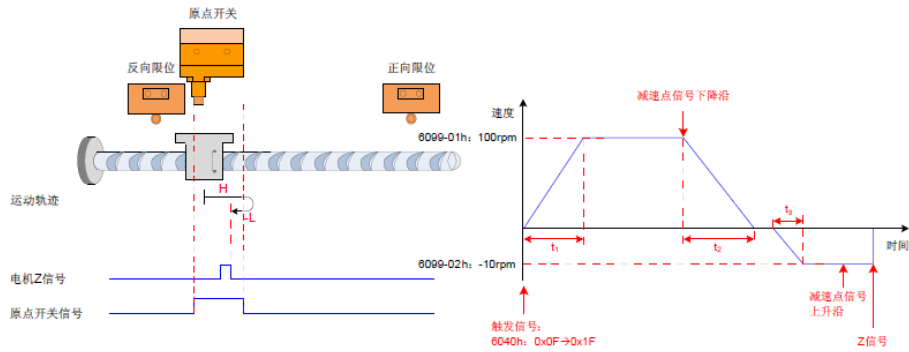


图 4-13 6098h=6 且减速点信号有效模式②

$$t1 = \frac{6099-01h}{609Ah} \text{ ms} \quad t2 = \frac{6099-01h}{609Ah} \text{ ms} \quad t3 = \frac{6099-02h}{609Ah} \text{ ms}$$

回零启动时 HW=1, 以正向高速开始回零, 遇到 HW 下降沿后, 减速→反向→反向低速运行, 遇到 HW 上升沿后的第一个 Z 停机。

6098h=24, 6098h=28 这2种回零方法都是基于回零方法8。

6098h=8

- 原点: Z 信号
- 减速点: 原点开关 (HW)

a) 回零启动时减速点信号无效, 未遇到正向限位开关

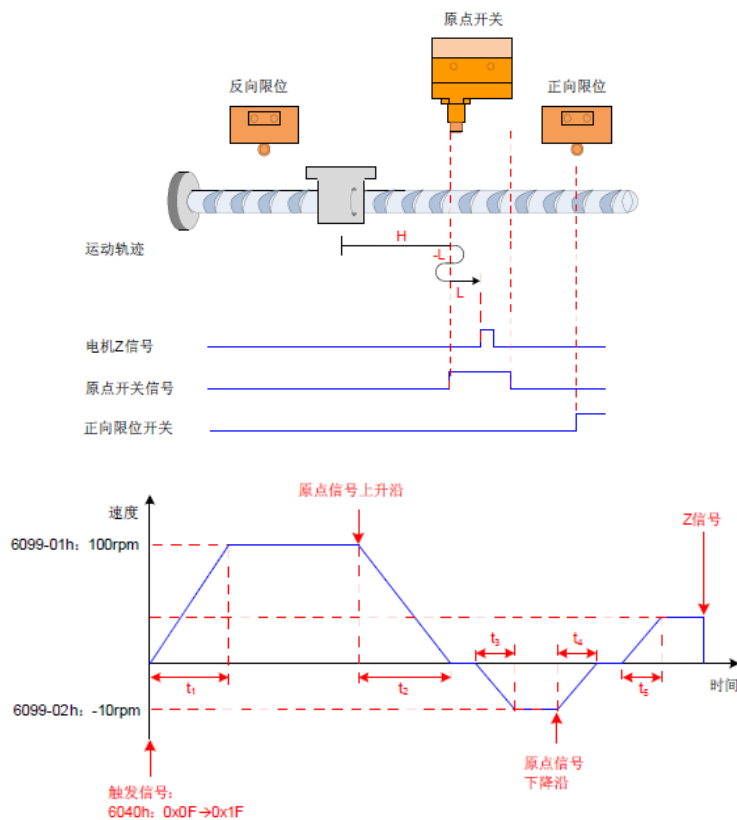


图 4-14 6098h=8 且减速点信号无效, 遇到正限位开关模式②

$$t1 = \frac{6099-01h}{609Ah} \text{ ms} \quad t2 = \frac{6099-01h}{609Ah} \text{ ms} \quad t3 = \frac{6099-02h}{609Ah} \text{ ms} \quad t4 = \frac{6099-02h}{609Ah} \text{ ms} \quad t5 = \frac{6099-02h}{609Ah} \text{ ms}$$

开始回零时 HW=0, 以正向高速开始回零, 若未遇到限位开关, 遇到 HW 上升沿后, 减速→反向→反向低速运行, 遇到 HW 下降沿后, 反向, 正向低速运行, 遇到 HW 上升沿后的第一个 Z 停机。

b) 回零启动时减速点信号无效，遇到正向限位开关

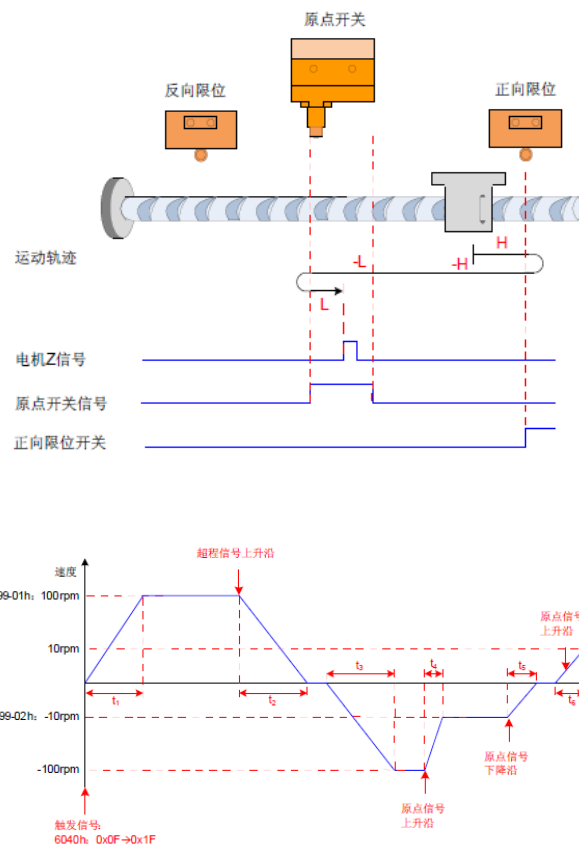


图 4-15 6098h=8 且减速点信号无效，遇到正限位开关模式②

$$t1 = \frac{6099-01h}{6094h} \text{ ms} \quad t2 = \frac{6099-01h}{6094h} \text{ ms} \quad t3 = \frac{[6099-01h] - [6099-02h]}{6094h} \text{ ms} \quad t4 = \frac{6099-02h}{6094h} \text{ ms} \quad t5 = \frac{6099-02h}{6094h} \text{ ms}$$

开始回零时 HW=0，以正向高速开始回零，若遇到限位开关，自动反向，反向高速运行，遇到 HW 上升沿后，减速，反向低速运行，遇到 HW 下降沿后→反向→正向低速，遇到 HW 上升沿后的第一个 Z 停机。

c) 回零启动时减速点信号有效

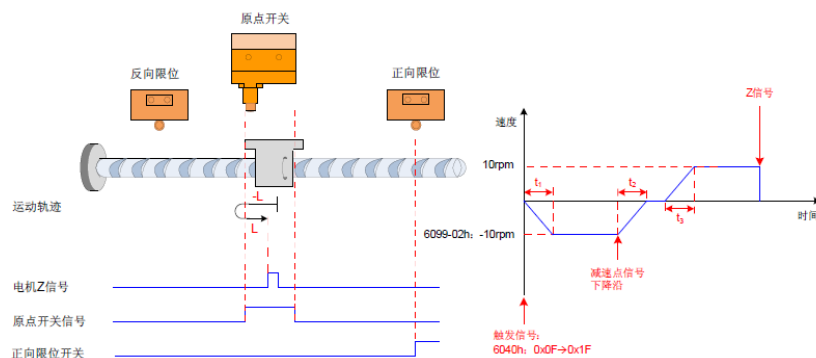


图 4-16 6098h=8 且减速点信号有效模式③

$$t1 = \frac{6099-02h}{6094h} \text{ ms} \quad t2 = \frac{6099-02h}{6094h} \text{ ms} \quad t3 = \frac{6099-02h}{6094h} \text{ ms}$$

回零启动时 HW=1，则直接反向低速开始回零，遇到 HW 下降沿后→反向→正向低速，遇到 HW 上升沿后的第一个 Z 停机。

6098h =24 的情况与方法 8 几乎一样就是最后一步会寻找 HW 上升沿来代替 Z 脉冲。

6098h =28 的情况与方法 8 几乎一样就是，运行方向和方法 8 正好完全相反，并且与方法 24 一样，在最后一步会寻找 HW 上升沿来代替 Z 脉冲。

6098h=33 和34

- 原点：Z 信号
 - 减速点：无
- a) 回零方式33：反向低速运行，遇到的第一个Z 信号停机
- b) 回零方式34：正向低速运行，遇到的第一个Z 信号停机

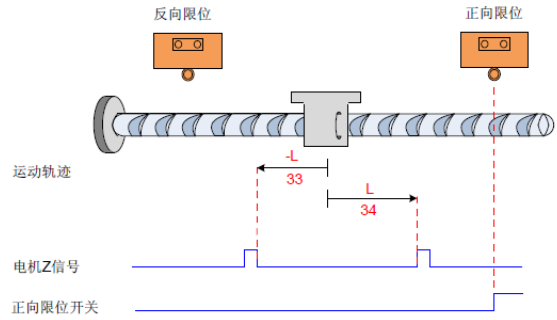


图 4-16 6098h=33 34

6098h=35

- 原点：当前点
- 减速点：无

35 模式是比较特殊的一种回零方式，此方式不需要任何外部信号。以此方式回零时，直接以当前点作为零点，重新建立坐标系。

4. 6. 4 与本模式相关的对象字典项

索引	名称	数据类型	存储属性
6040h	控制字	UINT16	RW
6041h	状态字	UINT16	RO
6060h	模式选择	INT8	RW
6061h	模式显示	INT8	RO
607Ch	用户原点偏置	INT32	RW
6091h	位置因子（齿轮比）	UINT32	RW
6098h	回零方法	INT8	RW
6099h	回零速度	ARRAY	RW
609Ah	回零加速度	UINT32	RW

控制字 6040h 在回零模式中的说明：

控制字 6040h					
位	bit7~15	bit6	bit5	bit4	bit0~bit3
名称	-	N/A	N/A	使能原点回零：	-
设定值	请参考表4-2 状态切换 与控制命令关系	-	-	-	请参考表4-2 状态切换 与控制命令关系
描述	-	-	-	0：未激活原点回零 0 → 1：使能原点回零； 1：原点回零进行中；	-

				1 → 0: 中断原点回零。 原点回零过程中, bit4 必须保持为 1。	
--	--	--	--	--	--

状态字 6041h 在回零模式中的说明:

状态字 6041h						
位	bit15-bit14	Bit13	Bit12	Bit11	Bit10	bit0~bit9
名称	NA	原点回零错误	原点回零完成	N/A	目标到达	-
设 定 值	NA	请参考表4-2 状态切换与控制 命令关系	-	-	-	请参考表4-2 状态切换与控 制命令关系
描述	NA	0- 无错误 1- 发生原点回零 错	0- 原点回零未完成 1- 原点回零完成	-	0-未找到零位 1- 找到零位	-

607Ch:

索引	607Ch
名字	原点偏置(Home Offset)
数据结构	VAR
数据类型	INT32
可访问性	RW
能否映射	Yes
单位	PUU
数据范围	-231~(231-1)
默认值	0
介绍	<p>设置位置类控制模式(轮廓位置模式、插补模式、原点回零)下机械零点偏离电机原点的物理位置。</p> <p>■ 原点偏置生效条件: 本次上电运行, 已完成原点回零操作, 状态字 6041h 的 bit15=1。</p> <p>■ 原点偏置的作用点:</p> <p>原点回零后: 用户当前位置6064h = 607Ch</p>

6098h:

索引	6098h
名字	回零方式(Homing Method)
数据结构	VAR
数据类型	INT8
可访问性	RW
能否映射	Yes
单位	-----
数据范围	0~35
默认值	0

介绍	具体设置值含义请参照上节
----	--------------

6099h:

索引	6099 h
名字	回零速度(Homing Speeds)
数据结构	ARRAY
元素数	2
数据类型	UINT32

子索引	01h
介绍	搜索减速点信号速度
可访问性	RW
能否映射	YES
单位	0.1rpm
数据范围	-----
默认值	将被初始化成PR22表示的速度

子索引	02h
介绍	搜索原点信号速度
可访问性	RW
能否映射	YES
单位	0.1rpm
数据范围	-----
默认值	将被初始化成PR23表示的速度

609Ah:

索引	609Ah
名字	回零加速度(Homing Acceleration)
数据结构	VAR
数据类型	UINT32
可访问性	RW
能否映射	Yes
单位	Ms/0rpm 升到额定速度
数据范围	$0 \sim (2^{32}-1)$
默认值	初始化成 PR25 参数
介绍	设置原点回零模式下的加速度。 原点回零启动后，设定值生效

4.6.5 注意

原点回归超程保护：原点回归启动后，当电机以第一速度寻找参考点，当移动部分超出设计的安全移动范围，限位开关开始动作，电机立刻反向并以第一速度继续寻找参考点，直到碰到限位开关的过程中都没有收到参考点信号，说明连在 REF 端口上的

接近开关出了故障，要发出报警信号（GOL）并停机，此时 6041h 里的原点回归失败位也被置“1”。

4.7 轮廓速度模式

4.7.1 模式简介

在此模式下伺服接收速度命令并且规划加减速过程。

4.7.2 本模式简要运行步骤

- ① 将字典项【运行模式:6060h】 设置为轮廓速度模式（3）。
- ② 设置字典项【控制字:6040h】来使能并且让电机运行（在伺服切换成使能状态的情况下，内部速度命令会重置并且 60ff 会被清 0）。
- ③ 将字典项【轮廓加速度:6083h】设置成速度曲线的上升斜率(单位：从 0rpm 加速到额定转速的加速时间 ms)。
- ④ 将字典项【轮廓减速度:6084h】设置成速度曲线的下降斜率(单位：从 0rpm 加速到额定转速的加速时间 ms)。
- ⑤ 设置字典项【目标速度:60FFh】单位 0.1rpm （当伺服使能后接收到了目标速度指令，驱动器会立即工作,当工作模式 6060 变化，或者转化到快速停车或者断掉使能状态下之后，系统都会清除 60ffh 的值）。
- ⑥ 查询字典项【状态字:6041h】来获取跟踪误差达到,能否接受新位置指令，位置到达等信息。

4.7.3 本模式高级用法

- 1) 主机可以获得关于轮廓速度模式的更多的信息
 - 通过查询字典项【位置给定值:606Bh】获取内部的速度指令(单位：0.1rpm)。
 - 通过查询字典项【位置实际值:606Ch】来得到速度运行的实际反馈(单位：0.1rpm)。
- 2) 设置速度到达阈值
 - 设置字典项【速度到达阈值:606Dh】 来设置速度到达区(单位:0.1rpm)。
 - 设置字典项【速度到达阈值时间:606Eh】 来设置触发速度到达信号输出的时间(单位:ms)。
 - 设置字典项【0 速度到达阈值:606Fh】 来设置 0 速度到达区(单位:0.1rpm)。

4.7.4 与本模式相关的对象字典项

索引	名称	数据类型	存储属性
6040h	控制字	UINT16	RW
6041h	状态字	UINT16	RO
6060h	模式选择	INT8	RW
6061h	模式显示	INT8	RO
606Bh	用户实际速度指令	INT32	RO
606Ch	用户实际速度反馈	INT32	RO
606Dh	速度到达阈值	Uint16	RW
606Eh	速度到达时间窗口	Uint16	RW
606Fh	零速阈值	Uint16	RW
60FFh	目标速度	INT32	RW

设置轮廓速度模式下的控制指令(6041h)与状态机中完全一致,可以完全参照 4.2 节。

注意：实际电机启动其实是依赖于 60FF 字典项的目标位置的输入。

状态字 6041h 在轮廓速度模式中的说明：

状态字 6041h						
位	bit15-bit14	Bit13	Bit12	Bit11	Bit10	bit0~bit9
名称	N/A	N/A	零速信号	N/A	目标到达	-
设定值	N/A	N/A	-	-	-	请参考表4-2 状态切换与控制 命令关系
描述	N/A	N/A	0- 用户速度不为 零 1- 用户速度为零	N/A	0- 目标速度未到 达 1- 目标速度到达	-

606Bh:

索引	606Bh
名字	用户实际速度指令 (Velocity Demand Value)
数据结构	VAR
数据类型	INT32
可访问性	R0
能否映射	Yes
单位	0.1rpm
数据范围	-231~(231-1)
默认值	0
介绍	反映用户实际速度指令

606Ch:

索引	606Ch
名字	用户实际速度反馈 (Velocity Actual Value)
数据结构	VAR
数据类型	INT32
可访问性	R0
能否映射	Yes
单位	0.1rpm
数据范围	-231~(231-1)
默认值	0
介绍	反映用户实际速度反馈值。

606Dh:

索引	606Dh
名字	速度到达阈值 (Velocity Window)
数据结构	VAR
数据类型	UINT16
可访问性	RW

能否映射	Yes
单位	0.1rpm
数据范围	0-65535
默认值	500
介 绍	<p>设置速度到达的阈值。</p> <p>目标速度60FFh 与用户实际速度606Ch 的差值在$\pm 606Dh$ 以内，且时间达到606Eh 时，认为速度到达，轮廓速度模式下，状态字6041h 的bit10=1。</p> <p>轮廓速度模式，伺服使能有效时，此标志位有意义；否则无意义。</p>

606Eh:

索引	606Eh
名字	速度到达时间窗口(Velocity Window Time)
数据结构	VAR
数据类型	UINT16
可访问性	RW
能否映射	Yes
单位	ms
数据范围	0-65535
默认值	0
介 绍	<p>设置判定速度到达有效的时间窗口。</p> <p>目标速度60FFh 与用户实际速度反馈606Ch 的差值在$\pm 606Dh$ 以内，且时间达到606Eh 时，认为速度到达，轮廓速度模式下，状态字6041h 的bit10=1。</p> <p>轮廓速度模式，伺服使能有效时，此标志位有意义；否则无意义。</p>

606Fh:

索引	606Fh
名字	零速阈值(Velocity Threshold)
数据结构	VAR
数据类型	UINT16
可访问性	RW
能否映射	Yes
单位	0.1rpm
数据范围	0-65535
默认值	500
介 绍	<p>设置用于判断用户速度是否为0 的阈值。</p> <p>用户速度反馈606Ch 在$\pm 606Fh$ 内，且时间达到6070h 设定值表示用户速度为0，不满足两者之中任一条件，认为用户速度不为0。</p> <p>轮廓速度模式，此标志位有意义；否则无意义。</p> <p>此标志位与伺服使能与否无关。</p>

60FFh:

索引	60FFh
名字	目标速度(Target Velocity)
数据结构	VAR
数据类型	INT32
可访问性	RW
能否映射	Yes
单位	0.1rpm
数据范围	-231~(231-1)
默认值	0
介绍	设置轮廓速度模式下，用户速度指令。

4.8 轮廓转矩模式

4.8.1 模式简介

在此模式下伺服接收转矩命令并且控制转矩。

4.8.2 本模式简要运行步骤

- ① 将字典项【运行模式:6060h】设置为轮廓转矩模式（4）。
- ② 设置字典项【控制字:6040h】来使能并且让电机运行（在伺服切换成使能状态的情况下，内部转矩命令会重置，这表明只要点击使能了，就在接收转矩信息）。
- ③ 将字典项【力矩模式下限制速度:205Ah】设置此模式下的速度限制值(单位：0.1rpm)。
- ④ 设置字典项【目标转矩:6071h】单位:千分之一额定转矩（当伺服使能后接收到了目标转矩指令，驱动器会立即工作,当工作模式 6060 变化，或者转化到快速停车或者断掉使能状态下之后，系统都会清除 6071h 的值）。

4.8.3 本模式高级用法

主机可以获得关于轮廓转矩模式的更多的信息

- 通过查询字典项【位置给定值:6074h】获取经过力矩限幅的给定值(单位：千分之一额定转矩)。
- 通过查询字典项【额定电流:6075h】来得到额定电流(单位:mA)。
- 通过查询字典项【实际力矩反馈:6077h】获取经过瞬时力矩输出值(单位：千分之一额定转矩)。
- 通过查询字典项【实际电流反馈:6078h】获取经过瞬时力矩输出值(单位：千分之一额定电流)。

4.8.4 与本模式相关的对象字典项

索引	名称	数据类型	存储属性
6040h	控制字	UINT16	RW
6041h	状态字	UINT16	RO
6060h	模式选择	INT8	RW
6061h	模式显示	INT8	RO
6071h	目标转矩	INT16	RW
6074h	实际力矩给定	INT16	RO
6075h	电机额定电流	UINT32	RO
6077h	力矩实际值	INT16	RO
6078h	实际输出力矩	INT16	RO

设置轮廓转矩模式下的控制指令(6041h)与状态机中完全一致,可以完全参照 4.2 节

状态字 6041h 在轮廓速度模式中的说明:

状态字 6041h						
位	bit15-bit14	Bit13	Bit12	Bit11	Bit10	bit0~bit9
名称	N/A	N/A	N/A	N/A	目标到达	-
设定值	N/A	N/A	-	-	-	请参考表4-2 状态切换与控制 命令关系
描述	N/A	N/A	N/A	N/A	0- 目标转矩未到 达 1- 目标转矩到达	-

6071h:

索引	6071h
名字	目标转矩(Target torque)
数据结构	VAR
数据类型	INT16
可访问性	RW
能否映射	Yes
单位	千分之一额定转矩
数据范围	-3000-3000
默认值	0
介绍	设定轮廓力矩模式下, 力矩的最大值

6074h:

索引	6074h
名字	实际力矩给定(Torque demand value)
数据结构	VAR
数据类型	INT16
可访问性	R0
能否映射	Yes
单位	千分之一额定转矩
数据范围	-32768-32767
默认值	0
介绍	显示轮廓力矩模式下, 输入驱动器力矩环的给定值。

6075h:

索引	6075h
名字	电机额定电流(Motor rated current)
数据结构	VAR
数据类型	UINT32

可访问性	R0
能否映射	Yes
单位	毫安
数据范围	0-(232-1)
默认值	0
介绍	一般从电机铭牌可知

6077h:

索引	6077h
名字	实际输出力矩(Torque actual value)
数据结构	VAR
数据类型	INT16
可访问性	R0
能否映射	Yes
单位	千分之一额定转矩
数据范围	-32768-32767
默认值	0
介绍	显示轮廓力矩模式下，实际的力矩输出值。

4.8.5 注意

请记得适当的设置字典项【力矩模式下限制速度:205Ah】，此值默认是 0，所以忘记设置后会造成输入额定转矩后电机不转的现象，设置其为非 0 数后，电机才能运行。

4.8 插补模式

4.8.1 模式简介

插补模式可实现多轴或单轴伺服驱动器的同步动作。上位机在伺服非使能状态下设置插补周期后， 根据实际需要，预先规划好位移曲线， 然后在伺服运行状态下， 将位移曲线上不同的绝对位置点， 周期性的发送至从机， 从机同步接收该位移指令， 并将位移指令增量按位置环控制周期细分， 均匀发送。 伺服驱动器内部完成位置、 速度与转矩控制。

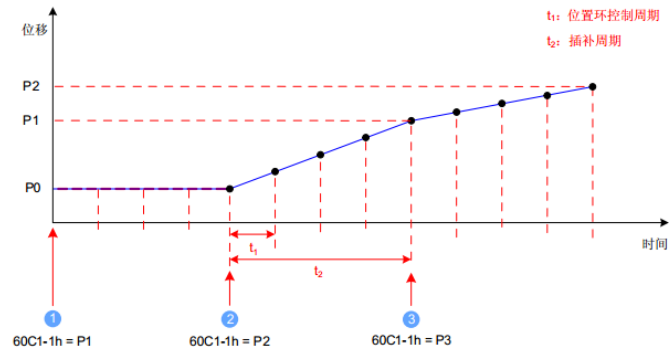


图 4-17 插补模式下的基本时序

4.8.2 本模式简要运行步骤

①读取字典项【反馈位置:6064h】读取位置后，上位机进行运行轨迹的规划。

- ②将字典项【插补时间间隔:60C2h】设置成所需要的同步周期,暂时只支持 1-20ms 的插补周期,设置更高的值只会限制在此范围中。同时配置 PDO 参数,把想周期读取和发送的参数设置成对应的 TPD0 和 RPD0。
- ③将字典项【运行模式:6060h】设置为插补模式(6)。
- ④根据配置的同步周期,先发送映射 60C1h 的 PDO,之后再发送 SYNC 帧,伺服收到 SYNC 帧后,会同步的更新当前目标位置。伺服接收的位置命令是绝对位置指令。
- ⑤设置字典项【控制字:6040h】来使能并且让电机运行。
- ⑥查询字典项【状态字:6041h】来获取跟踪误差达到,能否接受新位置指令,位置到达等信息。

4.8.3 本模式高级用法

1) 主机可以获得关于轮廓位置模式的更多的信息

- 通过查询字典项【位置给定值:6062h】获取内部的位置指令(单位:PUU)。
- 通过查询字典项【位置实际值:6063h】来得到位置运行的实际反馈(单位:脉冲)。

2) 跟踪误差

- 设置字典项【用户位置偏差过大阈值:6065h】来设置能接受的最大位置跟踪误差(单位:PUU)。
- 通过查询字典项【跟踪误差实际值:60F4h】来得到反馈误差的实际值(单位:PUU)。

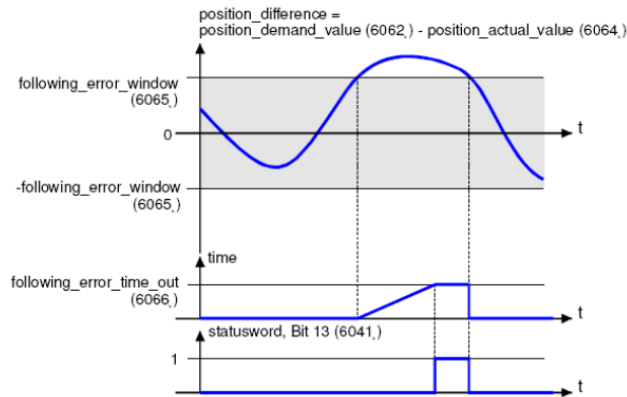


图 4-18 跟随误差功能描述

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。

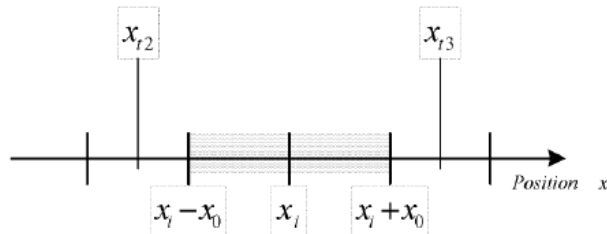


图 4-19 跟随误差举例

上图描述了对“following error”,窗口功能是如何定义。 $x_i - x_0$ 和 $x_i + x_0$ 范围(位置跟随误差窗口)对称地分布在期望位置(position_demand_value)两边。举例,位置 x_{t2} 和 x_{t3} 都位于位置跟随误差窗口外。如果驱动器离开窗口,那么状态字(statusword)的bit13(following_error)将被置1。

3) 位置窗口

- 设置字典项【位置到达阈值 6067h】来定义一个在目标位置附近的串口

(单位 PUU)

●设置字典项【位置到达时间窗口:6068h】来设定触发位置到达信号的时间(单位:毫秒)

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。

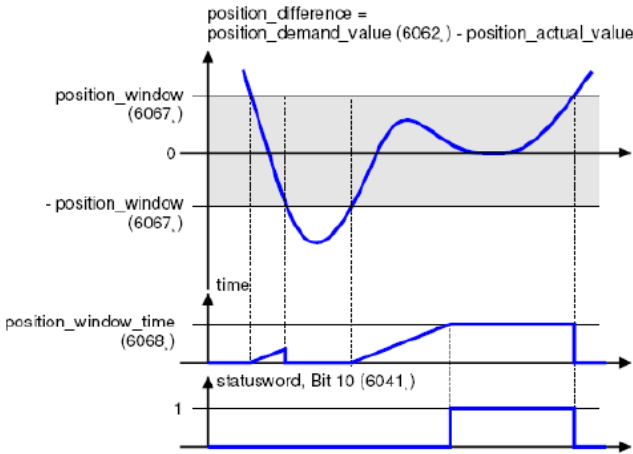


图 4-20 位置到达功能描述

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_window_time)且期间驱动器位置一直位于窗口内，那么状态字(statusword)的 bit10(target_reached)将被置 1。驱动器位置一离开该窗口，状态字的 bit10(target_reached)将立即被清零。

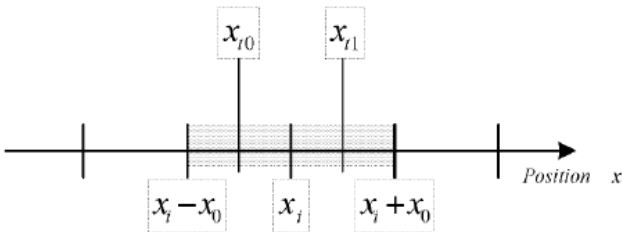


图 4-21 位置到达举例

4. 8. 4 与本模式相关的对象字典项

索引	名称	数据类型	存储属性
6040h	控制字	UINT16	RW
6041h	状态字	UINT16	RO
6060h	模式选择	INT8	RW
6061h	模式显示	INT8	RO
6063h	电机位置反馈[PUU]	INT32	RO
6064h	用户位置反馈	INT32	RO
6067h	位置到达阈值	UINT32	RW
6068h	位置到达时间窗口	UINT16	RW
6091h	位置因子（齿轮比）	ARRAY	RW
60C1h	插值位置	ARRAY	RW

60C2h	插值时间间隔	ARRAY	RW
-------	--------	-------	----

IP 模式当开始使能后会持续的检测 SYNC 帧的到来时间，请务必在 IP 模式使能前就把 PDO 帧和 SYNC 帧配置完整并正常的进行发送。如果 SYNC 发送太快，驱动器会报 CO_04 报警，如果发送过慢，驱动器会报 CO_05 报警。

控制字 6041h 在轮廓位置模式中的说明：

控制字 6040h					
位	bit7~15	bit6	bit5	bit4	bit0~bit3
名称	-	.		使能 ip 模式	-
设定值	请参考表4-2 状态切换 与控制命令关系	-	-	-	请参考表4-2 状态切换 与控制命令关系
描述	-	.		0- 不使能 ip 模式 1- 使能 ip 模式	-

状态字 6041h 在插补模式中的说明：

状态字 6041h						
位	bit15-bit14	Bit13	Bit12	Bit11	Bit10	bit0~bit9
名称	N/A	N/A	Ip 模式使能	N/A	目标到达	-
设定值	N/A	N/A	-	-	-	请参考表4-2 状态切换与控制 命令关系
描述	N/A	N/A	0- Ip模式没使能 1- Ip 模式已经使能	N/A	0- 目标位置未到达 1- 目标位置到达	-

60C1h:

索引	60C1 h
名字	插补数据记录 (Interpolation data record)
数据结构	ARRAY
元素数	1
数据类型	INT32

子索引	01h
介绍	插补位置
可访问性	RW
能否映射	YES
单位	PUU
数据范围	$-2^{31} \sim (2^{31}-1)$
默认值	0

60C2h:

索引	60C2h
名字	插补时间间隔 (Interpolation time period)

数 据 结 构	ARRAY
元 素 数	2
数 据 类 型	

子 索 引	01h
介 绍	插 补 间 隔 时 间
可 访 问 性	RW
能 否 映 射	YES
单 位	
数 据 范 围	
默 认 值	1

子 索 引	02h
介 绍	插 补 时 间
可 访 问 性	RW
能 否 映 射	YES
单 位	
数 据 范 围	-2, -3
默 认 值	-3

4.9 数字输入和数字输出

CiA402 协议除了规定运动控制相关对象字典外，还提供了直接读取伺服输入或控制伺服输出端口的能力，具体情况。

1) 数字输入（60FDh）

索引	子索引	名称	范围	数据类型	访问属性	PDO映射	单位	默认值
60FDh	00h	DI 数字输入	-	UINT32	RO	TPDO	-	0

60FDh 对象按位映射，其各位含义如下表：

Bit	31	30	29	28	27	26	25	24
功能	保留							DI9
Bit	23	22	21	20	19	18	17	16
功能	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1
Bit	15	14	13	12	11	10	9	8
功能	保留							
Bit	7	6	5	4	3	2	1	0
功能	保留			不支持		原点信号 (HOME)	正向限位 (POT)	反向限位 (NOT)

当各位值为“0”时，代表该输入无效；值为“1”时，代表该输入有效。

注：Bit0~Bit2 使用时，必须正确配置 IO 功能分配参数 P0~11~P0~15

2) 数字输出 (60FEh):

索引	子索引	名称	范围	数据类型	访问属性	PDO 映射	单位	默认值
60FEh	D0 数字输出			UINT32	RW	ALL		
	00h	子索引个数	—	UINT8	RO	NO	—	2
	01h	物理输出	0h-FFFFFFFFh	UINT32	RW	ALL	—	0
	02h	屏蔽位	0h-FFFFFFFFh	UINT32	RW	ALL	—	0

01 子索引值按位映射到物理输出上:

Bit	31	30	29	28	27	26	25	24
功能	保留							
Bit	23	22	21	20	19	18	17	16
功能	保留		D06	D05	D04	D03	D02	D01
Bit	15	14	13	12	11	10	9	8
功能	保留							
Bit	7	6	5	4	3	2	1	0
功能	保留							不支持

Bit16-Bit21 对应着物理输出 D01-D06，当 01 子索引某一位值为 1 时，且 02 子索引该位也为 1 时，使能该 D0 口输出，否则不输出有效电平。逻辑关系如下表所示：

01 子索引 Bitx 值	02 子索引 Bitx 值	输出状态
0	0	输出无效 (switch off)
1	0	输出无效 (switch off)
0	1	输出无效 (switch off)
1	1	输出有效 (switch on)

第五章 对象字典

5.1 对象分类说明

5.1.1 读写类型说明

读写类型	说明
RW	可读写
WO	只写
RO	只读
CONST	常量 只读

5.1.2 对象分类说明

类别	含义	DS301 值
VAR	单一简单数值，包含数据类型 Int8、Uint16、String 等	7
ARR	具有相同类型的数据块	8
REC	具有不同类型的数据块	9

5.1.3 数据类型说明

参照标准 DS301 标准。

5.2 对象组 1000h 分配一览

索引	名称	读写属性	对象类型	数据类型
1000h	device type	RO	VAR	UINT32
1001h	error register	RO	VAR	UINT8
1003h	pre-defined error field	RW	ARRAY	UINT32
1005h	COB-ID SYNC	RW	VAR	UINT32
1006h	communication cycle period	RW	VAR	UINT32
1014h	COB-ID EMCY	RO	VAR	UINT32
1016h	Consumer heartbeat time	RW	ARRAY	UINT32
1017h	Producer heartbeat time	RW	VAR	UINT16
1018h	Identity Object	RO	RECORD	UINT32
1200h	1st Server SDO parameter	RO	RECORD	UINT32/SDO parameter
1400h~03h	Receive PDO parameter	RW	RECORD	UINT16/32
1600h~03h	Receive PDO mapping	RW	RECORD	UINT32
1800h~03h	Transmit PDO parameter	RW	RECORD	UINT16/32
1A00h~03h	Transmit PDO mapping	RW	RECORD	UINT32

表 5.1 1000h 对象组

5.3 对象组 2000h 说明

F 系列伺服：

200Xh 对象组为日鼎技术公司定义的对象表，与相应机器的参数对应。

P0 组参数对应于对象字典的 2000h，组内的标号对应于其子索引+1。比如 P0-XX 参数对应对象字典 2000h-(XX+1)h，P1-XX 参数则对应于对象字典的 2001h-(XX+1)h，P0-07 对应于 2000-08h，P1-99 对应于 2001h-64h。

注：该区域请使用 SDO 进行访问。

5.4 对象组 6000h 分配一览

索引	名称	读写属性	对象类型	数据类型
603Fh	Error Code	VAR	RO	UINT16
6040h	Controlword	VAR	RW	UINT16
6041h	Statusword	VAR	RO	UINT16
605Bh	Shutdown option code	VAR	RW	INT16
6060h	Modes of operation	VAR	RW	INT8
6061h	Modes of operation display	VAR	RO	INT8
6062h	Position demand value [PUU]	VAR	RO	INT32
6063h	Position actual value [inc]	VAR	RO	INT32
6064h	Position actual value	VAR	RO	INT32
6065h	Following error window	VAR	RW	UINT32
6067h	Position windows	VAR	RW	UINT32
6068h	Position window time	VAR	RW	UINT16
606Bh	Velocity demand value	VAR	RO	INT32
606Ch	Velocity actual value	VAR	RO	INT32
606Dh	Velocity window	VAR	RW	UINT16
606Eh	Velocity window time	VAR	RW	UINT16
606Fh	Velocity threshold	VAR	RW	UINT16
6071h	Target torque	VAR	RW	INT16
6074h	Torque demand value	VAR	RO	INT16
6075h	Motor rated current	VAR	RO	UINT32
6076h	Motor rated torque	VAR	RO	UINT32
6077h	Torque actual value	VAR	RO	UINT16
6078h	Current actual value	VAR	RO	INT16
607Ah	Target position	VAR	RW	INT32
607Ch	Home Offset	VAR	RW	INT32
607Dh	Software position limit	ARRAY	RW	INT32
607Eh	Polarity	VAR	RW	UINT8
607Fh	Max profile velocity	VAR	RW	UINT32
6081h	Profile velocity	VAR	RW	UINT32
6083h	Profile acceleration	VAR	RW	UINT32
6084h	Profile deceleration	VAR	RW	UINT32
6085h	Quick stop deceleration	VAR	RW	UINT32
6091h	Gear ratio	ARRAY	RW	UINT32

6098h	Homing method	VAR	RW	INT8
6099h	Homing speeds	ARRAY	RW	UINT32
609Ah	Homing acceleration	VAR	RW	UINT32
60C1h	Interpolation data record	ARRAY	RW	INT32
60C2h	Interpolation time period	ARRAY	RW	INT8
60C5h	Max acceleration	VAR	RW	UINT32
60C6h	Max deceleration	VAR	RW	UINT32
60E0	Positive torque limit	VAR	RW	UINT16
60E1	Negative torque limit	VAR	RW	UINT16
60F4h	Following error actual value	VAR	RO	INT32
60FCh	Position demand value	VAR	RO	INT32
60FDh	Digital inputs	VAR	RO	UINT32
60FEh	Digital outputs	ARRAY	RO	INT32
60FFh	Target velocity	VAR	RW	INT32
6502h	Supported drive modes	VAR	RO	UINT32

表 5.2 6000h 对象组

第六章 故障处理

当通讯或者驱动器出现异常时，伺服驱动器以生产者的形式向网络发送紧急报文，或者 SDO 传输异常时发送中止应答。节点与紧急报文相关的有节点错误码及辅助信息。

6.1 CANOpen 通讯故障码

F 系列的通讯故障码：

CANOpen 错误码	说明	附加错误信息
2311	过电流 1 (OC1)	1
2312	过电流 2 (OC2)	2
8400	过速 (OS)	3
3110	母线过压 (HU)	4
5420	泄放回路故障 (EP)	5
5530	存储器故障 (dE)	6
7305	编码器故障 (EC)	7
5210	电流反馈回路故障 (EH)	8
3230	过载 (OL)	9
3120	直流母线欠压 (LU)	10
8611	超程 (OF)	11
4210	驱动器过热 (AH1)	12
7500	CPLD 通信错误 (PLD)	13
7112	制动电阻过热 (rH1, BROH)	14
3121	使用掉电报警 (POL)	16
7122	电机驱动器不匹配 (CE)	17
FF01	电机未选择 (ND)	18
8613	回原点失败 (GOH)	22
5442	左限位报警 (PPOT)	23
5443	右限位报警 (PNOT)	24
FF03	绝对值编码器 BAT1	19
FF04	绝对值编码器 BAT2	20
7320	绝对值编码器多圈报警	21
8160	Canopen 运行模式选择错误 (CO_01)	27
8163	Canopen 状态机切换错误 (CO_02)	28
8161	Ip 模式同步过快 (CO_04)	38
8162	Ip 模式同步过快 (CO_05)	39

以上和报警中和 canopen 通讯直接相关的是最后 4 项。

- CO_01 报警表示还没有设置 canopen 的运行模式的状态下就使能伺服。解决方法可以检查上位机相关的程序，保证每次在设置了 6060 对象字典后，再进行伺服的使能操作。
- CO_02 报警表示 canopen 的 301 状态机切换错误，比较典型的引发原因有：1. 在正常使能后，仍有 NMT 帧进行 301 状态机的复位，停止节点等操作。2. 上位机断电后，伺服驱动器没有断电，上位机再次上电后的初始化操作进行了 301 状态机的操作。

对应解决的方法就是 1. 检查上位机代码，注意伺服使能后不要进行 NMT 操作。2. 上位机初始化时序进行优化，可以通过 SDO 的检查状态字 6041，判断伺服是否仍然处于使能状态，之后进行断使能操作，最后进行 301 状态机的初始化等。

c). CO_04 报警表示运行在 canopen 的插补模式(ip)下,设置在同步模式的 RPDO 没能按照设置的同步周期按时接收到，伺服在连续几个周期内收到多个 SYNC 同步帧。

解决此报警一般需要检查对应的上位机的 PDO 设置与实际发送的 SYNC 帧同步周期是否对应。

d). CO_05 报警表示运行在 canopen 的插补模式(ip)下,设置在同步模式的 RPDO 没有按照设置的同步周期按时接收到，接收到的 SYNC 同步帧过少。

解决此报警的思路有：1. 检查连接状态，按要求接入终端电阻，尽量使用带屏蔽层的双绞线进行总线连接，检查接地，保证信号质量。2. 检查上位机的 PDO 设置与实际发送的 SYNC 帧同步周期是否对应。3. 谨慎提高 P0-85 参数的 bit4-7，加大此参数能降低 CO_05 报警的灵敏度，但是也可能引发伺服的动作失调，只建议在恒速旋转等简单应用场合下使用，且最大不建议超过 5。

6.2 SDO 中止代码

中止代码	代码功能描述
05030000h	触发位没有交替改变
05030005h	内存溢出
05040001h	非法或未知的 SDO 命令字
05040000h	SDO 协议超时
06010000h	对象不支持访问
06010001h	试图读只写对象
06010002h	试图写只读对象
06020000h	对象字典中对象不存在
06040041h	对象不能映射到 PDO
06040042h	映射的对象的数目和长度超出 PDO 长度
06070010h	对象长度不匹配
06090011h	子索引不存在
06090030h	超出参数的值范围
08000000h	一般性错误
080000a2h	EEPROM 写错误

表 6-2 SDO 中止代码表

第七章 通讯例程

7.1 简表位置模式

假设从站站号为 2，简表位置模式下的通讯例程如下所示。

配置 RPD01

```
602 2F 00 16 00 00 00 00 00//清除 RPD01 映射
602 23 00 16 01 10 00 40 60//将控制字 6040h 映射到 RPD01 映射参数的第一个子索引
602 2F 00 16 00 01 00 00 00// 设置 RPD01 有效的映射变量个数
```

配置 RPD02

```
602 2F 01 16 00 00 00 00 00//清除 RPD01 映射
602 23 01 16 01 20 00 7A 60//将目标位置 607Ah 映射到 RPD02 映射参数的第一个子索引
602 23 01 16 02 20 00 81 60//将目标速度 6081h 映射到 RPD02 映射参数的第二个子索引
602 2F 01 16 00 02 00 00 00//设置 RPD02 有效的映射变量个数
```

配置 TPD01

```
602 2F 00 1A 00 00 00 00 00//清除 TPD01 映射
602 23 00 1A 01 20 00 64 60//将实际位置值 6064h 映射到 TPD01 映射参数的第一个子索引
602 2F 00 1A 00 01 00 00 00// 设置 TPD01 有效的映射变量个数
```

发送 NMT 指令进入操作状态

```
000 01 02
```

设置运行模式

```
602 2F 60 60 00 01 00 00 00//设置控制模式为 1（位置模式）
```

设置目标位置和目标速度

```
302 50 C3 00 00 E8 03 00 00//由以上 PDO 的映射关系可知 C350 为目标位置，50000 个脉冲；03E8 为目标速度，100r/min（速度单位为 0.1r/min）
```

设置 DS402 状态机

```
202 06 00//进入伺服准备好状态
202 07 00//进入等待打开伺服使能状态
202 0F 00//进入伺服使能状态
202 1F 00//6040 的 bit6 设置为 0，设置电机运行于绝对模式下
202 0F 00//6040 的 bit4 清零
202 5F 00//6040 的 bit6 设置为 1，电机运行于相对位置模式下
202 0F 00//6040 的 bit4 清零，为下次启动做准备
```

如果还需要 PDO 映射其他变量，可按照上面类似的方法来映射其他变量。只要不超过一个 PDO 可以映射的最大变量个数（8 个字节），一个 PDO 可以映射多个变量。TPDO 默认为异步模式，定时时间为 0，即只在映射变量发生变化的时候发送，如果需要修改为定时模式，可以配置 TPDO 通信参数（1800h-1803h）的子索引 05h，如果需要修改 TPDO 的禁止时间，可以修改 TPDO 通信参数（1800h-1803h）的子索引 03h。

驱动器是以 6040 的 bit4 的 ↑（上升沿）接收新的位置命令，所以每次执行完一次运行后需要把此位清零。上位机应根据驱动器的状态字 6041 的 bit12 来判断是否要给伺服新数据。当驱动器的状态字 6041 的 bit12 为 0 时，表示驱动器可以接收新的位置数据和命令了，如果为 1，即使驱动器有接收数据，但是命令是不被采纳的。

7.2 简表速度模式

配置 RPD01

```
602 2F 00 16 00 00 00 00 00//清除 RPD01 映射
```

602 23 00 16 01 10 00 40 60//将控制字 6040h 映射到 RPD01 映射参数的第一个子索引

602 2F 00 16 00 01 00 00 00// 设置 RPD01 有效的映射变量个数

配置 RPD02

602 2F 01 16 00 00 00 00 00//清除 RPD01 映射

602 23 01 16 01 20 00 FF 60//将目标速度 60FFh 映射到 RPD02 映射参数的第一个子索引

602 2F 01 16 00 01 00 00 00//设置 RPD02 有效的映射变量个数

配置 TPD01

602 2F 00 1A 00 00 00 00 00//清除 TPD01 映射

602 23 00 1A 01 20 00 6C 60//将速度实际值 6064h 映射到 TPD01 映射参数的第一个子索引

602 2F 00 1A 00 01 00 00 00// 设置 TPD01 有效的映射变量个数

发送 NMT 指令进入操作状态

000 01 02

设置运行模式

602 2F 60 60 00 03 00 00 00//设置控制模式为 3（速度模式）

设置 DS402 状态机

202 06 00//进入伺服准备好状态；

202 07 00//进入等待打开伺服使能状态

202 0F 00//进入伺服使能状态

设置给定目标速度

302 E8 03 00 00//给定目标速度 100r/min

如果还需要 PDO 映射其他变量，方法和位置模式下类似。在速度模式下，必须首先设定伺服进入使能状态后，给定目标速度，电机才能运行起来，否则无效。例如，首先发送 302 E8 03 00 00，然后改变 DS402 状态机使其进入伺服使能状态，伺服不会启动。

7.3 简表转矩模式

配置 RPD01

602 2F 00 16 00 00 00 00 00 //清除 RPD01 映射

602 23 00 16 01 10 00 40 60 //将控制字 6040h 映射到 RPD01 映射参数的第一个子索引

602 2F 00 16 00 01 00 00 00 // 设置 RPD01 有效的映射变量个数

配置 RPD02

602 2F 01 16 00 00 00 00 00//清除 RPD01 映射

602 23 01 16 01 10 00 71 60//将目标转矩 6071h 映射到 RPD02 映射参数的第一个子索引

602 2F 01 16 00 01 00 00 00//设置 RPD02 有效的映射变量个数

配置 TPD01

602 2F 00 1A 00 00 00 00 00 //清除 TPD01 映射

602 23 00 1A 01 20 00 77 60 //将实际转矩值 6077h 映射到 TPD01 映射参数的第一个子索引

602 2F 00 1A 00 01 00 00 00 // 设置 TPD01 有效的映射变量个数

发送 NMT 指令进入操作状态

000 01 02

设置运行模式

602 2F 60 60 00 04 00 00 00//设置控制模式为 4（转矩）

设置 DS402 状态机

202 06 00 //进入伺服准备好状态；

```
202 07 00 //进入等待打开伺服使能状态
202 0F 00 //进入伺服使能状态
设置给定目标转矩
302 64 00 //给定目标转矩（单位为千分之额定转矩）
```

如果还需要 PDO 映射其他变量，方法和位置模式下类似。在转矩模式下，必须首先设定伺服进入使能状态后，给定目标转矩，电机才能运行起来，否则无效。例如，首先发送 302 64 00，然后改变 DS402 状态机使其进入伺服使能状态，伺服不会启动。

7.4 回零模式

配置 RPD01

```
602 2F 00 16 00 00 00 00 //清除 RPD01 映射
602 23 00 16 01 10 00 40 60 //将控制字 6040h 映射到 RPD01 映射参数的第一个子索引
602 2F 00 16 00 01 00 00 00 // 设置 RPD01 有效的映射变量个数
```

配置 RPD02

```
602 2F 01 16 00 00 00 00 //清除 RPD01 映射
602 23 01 16 01 20 01 99 60 //将寻找原点开关速度 6099h 的子索引 01h 映射到 RPD02 映射参数的第一个子索引
602 23 02 16 01 20 02 99 60 //将寻找零点速度 6099h 的子索引 02h 映射到 RPD02 映射参数的第二个子索引
602 2F 01 16 00 02 00 00 00 //设置 RPD02 有效的映射变量个数
```

发送 NMT 指令进入操作状态

```
000 01 02
```

设置运行模式

```
602 2F 60 60 00 06 00 00 00 //设置为回零模式
```

设置回零方式

```
602 2F 98 60 00 03 00 00 00 //回零方式设置为 3，具体请参考回零方式选择
```

设置 DS402 状态机

```
202 06 00 //进入伺服准备好状态；
202 07 00 //进入等待打开伺服使能状态
202 0F 00 //进入伺服使能状态
202 1F 00 //启动回零
```

如果还需要 PDO 映射其他变量，方法和位置模式下类似。驱动器是以 6040 的 bit4 的 ↑（上升沿）作为回零的启动信号。回零模式的方法有很多种，具体可以参考本说明书对回零模式的描述。

7.5 插补模式

首先，映射和配置 PDO

```
//默认接收 1 个 PDO： RPD01： 60C1h—01h
```

```
//默认发送 2 个 PDO： TPD01： 6041h TPD02： 6064h/606Ch
```

```
//脉冲 pulse，转速 0.1rpm
```

例子：配置 1 个 RPDO

RPDO MAPPING

```
报文： 601(ID) 2F 00 16 00 00 00 00 //RPD01 stop
```

```
报文： 601(ID) 23 00 16 01 20 01 C1 60 //60C1h, sub01
```

```
报文： 601(ID) 2F 00 16 00 01 00 00 00 // RPD01 enable
```

配置 2 个 TPD0, TPD01: 6041h TPD02: 6064h/606Ch

TPDO MAPPING

报文: 601(ID) 2F 00 1A 00 00 00 00 00//TPD01 stop

报文: 601(ID) 23 00 1A 01 10 00 41 60//6041h

报文: 601(ID) 2F 00 1A 00 01 00 00 00 00// TPD01 enable

报文: 601(ID) 2F 01 1A 00 00 00 00 00 00//RPD02 stop

报文: 601(ID) 23 01 1A 01 20 00 64 60//6064h and 606Ch

报文: 601(ID) 23 01 1A 02 20 00 6C 60//

报文: 601(ID) 2F 01 1A 00 02 00 00 00 00// TPD02 enable

然后设置 SYNC 周期

报文: 601(ID) 2F C2 60 01 01 00 00 00 00//60C2h-01---->1ms, 根据实际需要设置

报文: 601(ID) 2F C2 60 02 FD 00 00 00 00//

配置接收发送 PDO 都为接收到一个同步帧即有效方式

然后设置 1400h。

报文: 601(ID) 2F 00 14 02 01 00 00 00 00//1400-02---->SYNC

然后设置 1800h。

报文: 601(ID) 2F 00 18 02 01 00 00 00 00 00//1800-02---->SYNC

然后设置 1801h。

报文: 601(ID) 2F 01 18 02 01 00 00 00 00 00//1801-02---->SYNC

设置控制模式

报文: 601(ID) 2F 60 60 00 07 00 00 00 00//设置 6060h 为 7 (IP 位置控制)

然后, 设置状态机

报文: 601(ID) 2B 40 60 00 06 00 00 00 00//设置 6040h 为 6

报文: 601(ID) 2B 40 60 00 07 00 00 00 00//设置 6040h 为 7

报文: 601(ID) 2B 40 60 00 0F 00 00 00 00//设置 6040h 为 F, 使电机使能

报文: 601(ID) 2B 40 60 00 1F 00 00 00 00//设置 6040h 为 1F, 使驱动进入 IP_ACTIVE 状态

启动通讯

报文: 00 01 01

第二轴的设置与之相同

上位机则按照设置测 SYNC 周期(1ms)发送:

报文: 201(ID) 10 00 00 00 00//16 个脉冲, 第一轴插补位置数据

报文: 202(ID) 20 00 00 00 00//32 个脉冲, 第一轴插补位置数据

报文: 80(ID)//根据设置的插补周期循环发送 SYNC

则驱动返回

报文: 181(ID) xx xx 00 00//第一轴状态字

报文: 182(ID) xx xx xx xx//第一轴的位置和速度

报文: 281(ID) xx xx 00 00//第二轴状态字

报文: 282(ID) xx xx xx xx//第二轴的位置和速度

时序图为:

