

CAN 总线伺服驱动器应用说明

第一章 配线

1.1 通信接口示意图

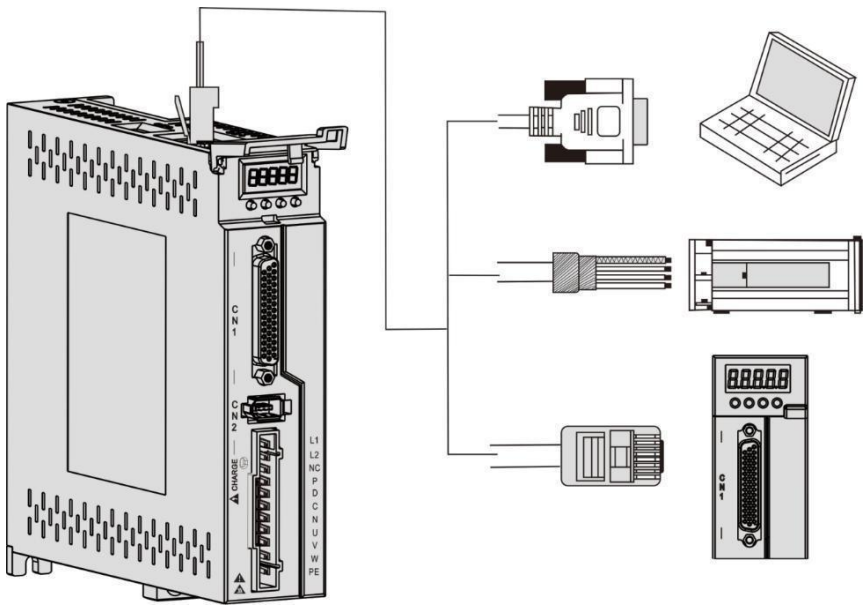


图 1.1 通信口接线示意图

1.2 通讯端口引脚定义说明

通过驱动器上的 CN3 和 CN4 端口，可以实现驱动器与 PC、PLC 及驱动器的通信连接，其中 CN3 和 CN4 的端口引脚定义如下：

针脚号	CN3	名称	CN4	名称	图示
1	CAN H	CAN总线接口	CAN H	CAN总线接口	
2	CAN L	CAN总线接口	CAN L	CAN总线接口	
3	E0V	CAN信号地	E0V	CAN信号地	
4	RS485 -	RS485通讯 接口	RS485 -	RS485通讯 接口	
5	RS485+		RS485+		

6	NC	空端	NC	空端	
7	GND	485信号地	GND	485信号地	
8	NC	空端	NC	空端	

1.3 CAN 通讯组网连接说明

1.与 PLC 的 CAN 通讯连接

采用 CAN 通讯组网时，驱动器与 PLC 的连接线缆如下：

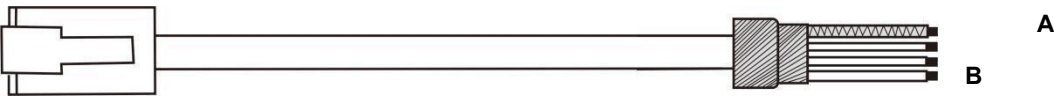


图 1.2 伺服驱动器与 PLC 通讯线缆示意图表

1.3 伺服驱动器与 PLC 通讯线缆引脚连接关系

驱动器侧 RJ45 (A 端)		PLC 侧 (B 端)	
信号名称	针脚号	信号名称	针脚号
CAN_H	1	CAN_H	1
CAN_L	2	CAN_L	2
CGND	3	CGND	3
PE (屏蔽网层)	壳体	PE (屏蔽网层)	壳体

2.多机并联的 CAN 通讯连接

采用 CAN 通讯组网时，伺服驱动器多机并联的连接线缆如下：

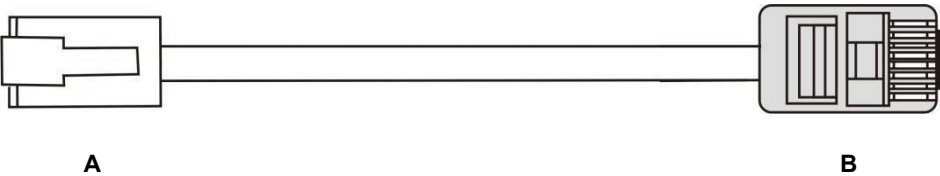


图 1.4 伺服驱动器多机并联线缆示意图

表 1.5 伺服驱动器多机并联通讯线缆引脚连接关系

驱动器侧 RJ45 (A 端)		驱动器侧 RJ45 (B 端)	
信号名称	针脚号	信号名称	针脚号
CAN_H	1	CAN_H	1
CAN_L	2	CAN_L	2
CGND	3	CGND	3
PE (屏蔽网层)	壳体	PE (屏蔽网层)	壳体

3.CAN 通信的总线和多节点的连接方式

CAN 通信网络的连接方式为总线连接方式，如下图所示。各个 CAN 收发设备挂接在总线上，每个分支长度要小于 0.3m。否则会引起反射，造成通信问题。

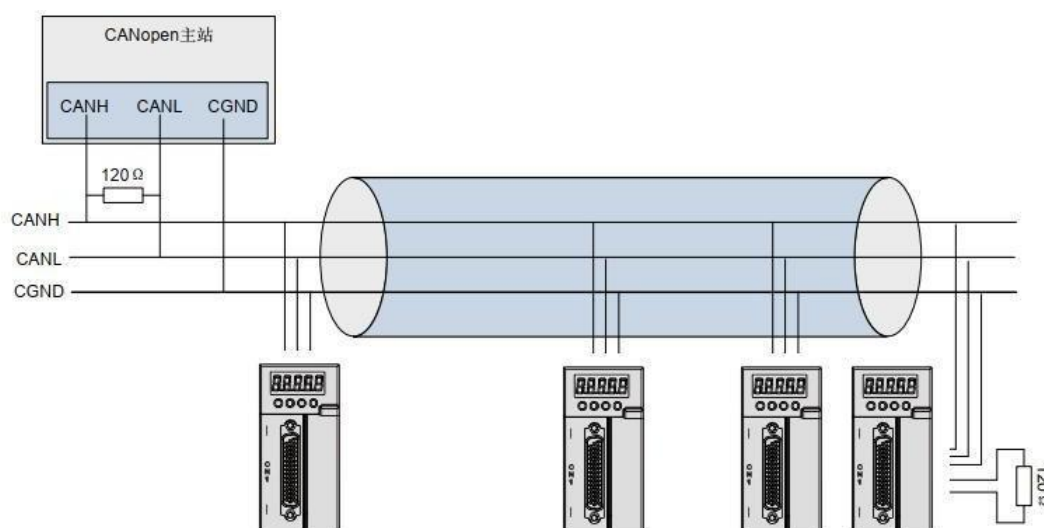


图 1.6 can 总线连接

- 推荐使用带屏蔽双绞线连接，总线两端分别连接两个 120Ω 终端匹配电阻防止信号反射，屏蔽层一般使用单点可靠接地。
- 用万用表测量 CANH 和 CANL 之间的阻值可以确认现场端接电阻是否正确，正常阻值应为 60Ω 左右（两个电阻并联）。
- 挂接设备数量最多为 64 个。
- CAN 设备长距离通信时，须将不同 CAN 电路的公共地 CGND 相互连接，以保证不同通信设备之间参考电位相等。

1.4 推荐线缆及接线方式

- CAN 通信网络推荐使用双绞线缆，双绞线对高频磁场噪声干扰有很好的抵抗能力，也能减小线缆对外的辐射，如图所示。

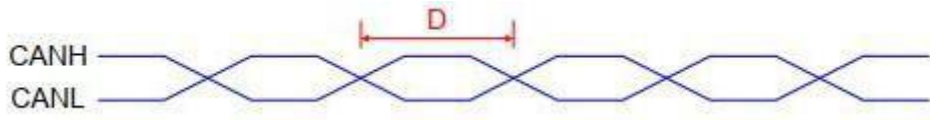


图 1.7 双绞线示意图

- 双绞线的扭距 D 应小于 2cm，扭距越小抗干扰效果越好。
- 短距离低速通信时，为了增加抗干扰能力可以使用双绞屏蔽线，屏蔽层双端接 PE。
- 长距离高速通信时，不建议使用屏蔽线。因为屏蔽层和信号线之间存在的较大分布电容，会导致传输信号延迟。

对应不同的线缆接线，有如下几种推荐接线方式。

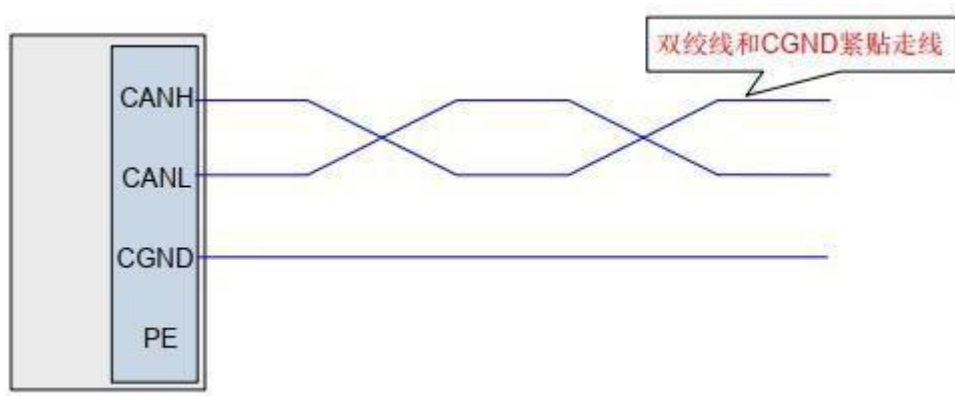


图 1.8 推荐方案一

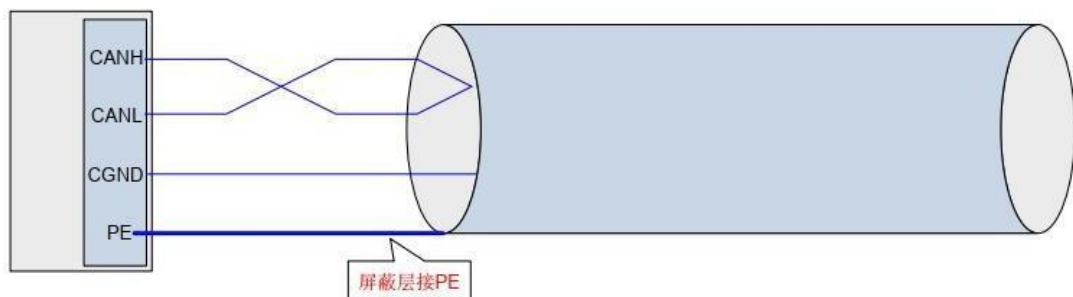


图 1.9 推荐方案二

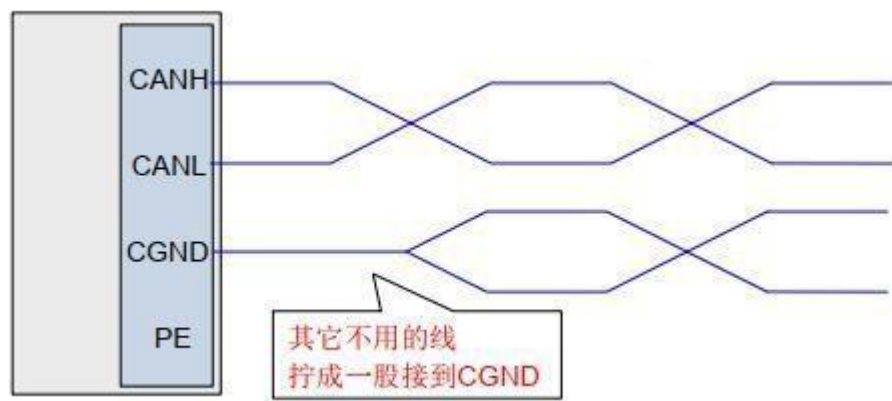


图 1.10 推荐方案三

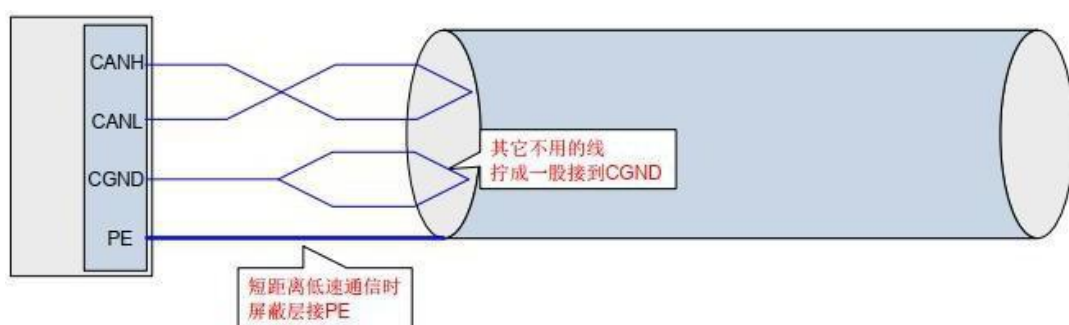


图 1.11 推荐方案四

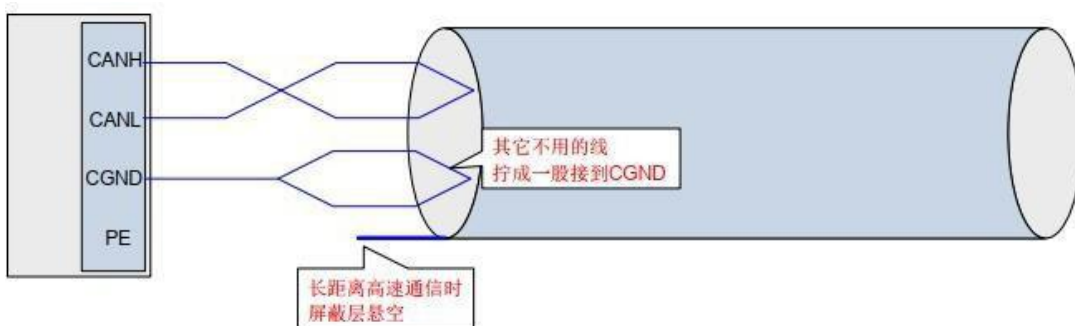


图 1.12 推荐方案五

1.5 CAN 通信接地注意事项

采用 CAN 通信时，注意上位装置的 CGND 端子与伺服驱动器的 CGND 端子相连接，如下图：

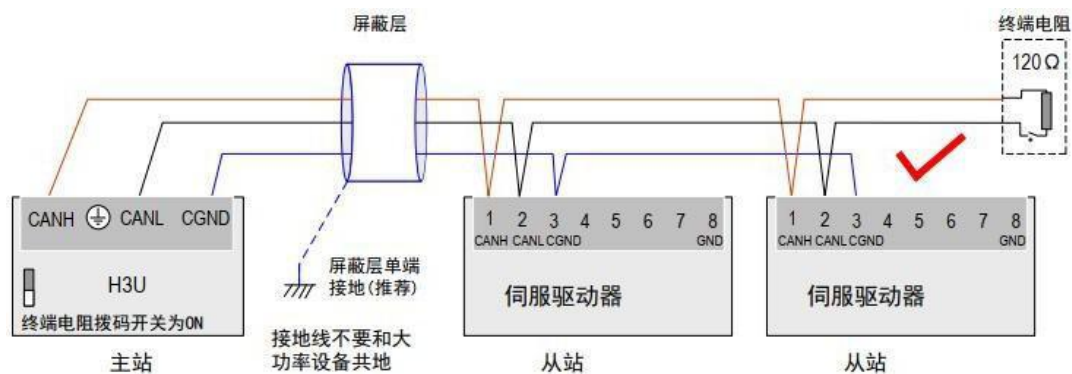


图 1.13 正确的 CAN 连接方法

注意： 1) PLC 内置 CAN 通信终端电阻，相应的拨码开关须置为 ON；

2) 推荐将屏蔽层进行单端接地处理。

切勿将上位装置的 CGND 端子与伺服驱动器的 GND 端子相连接，否则将损坏机器！

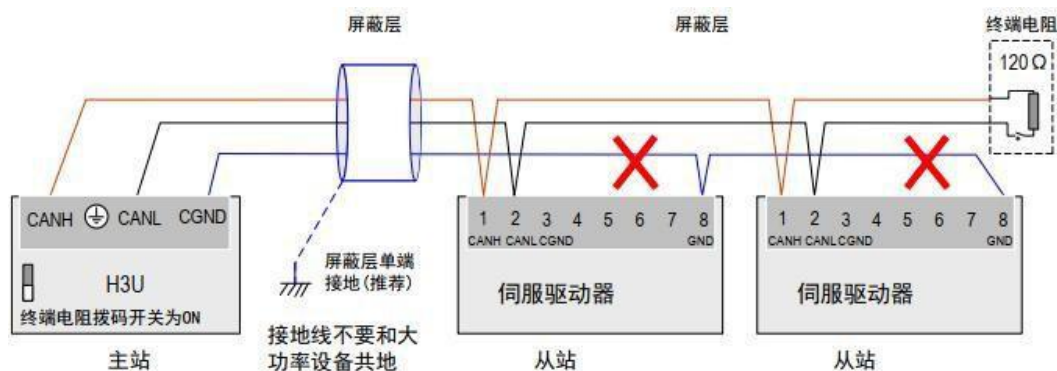


图 1.14 错误的 CAN 连接方法

1.6 没有外接CGND 端口配线说明

1.6.1 设备为非隔离CAN，与其它信号公用GND 或是COM 端口

将该设备 GND 或是 COM 与我司设备的 CGND 连接。

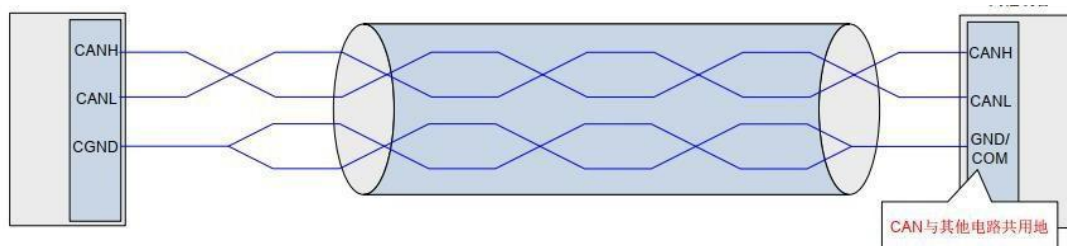


图 1.15 与其它电路共用地的连接方式

1.6.2 设备的 CAN 与其它端口无公共地

CGND 不连接任何线缆，额外用一根不小于 AWG12 的线缆，将各个设备的 PE 相连，该线缆与 CAN 通信线距离大于 5cm，图 1.16 所示。

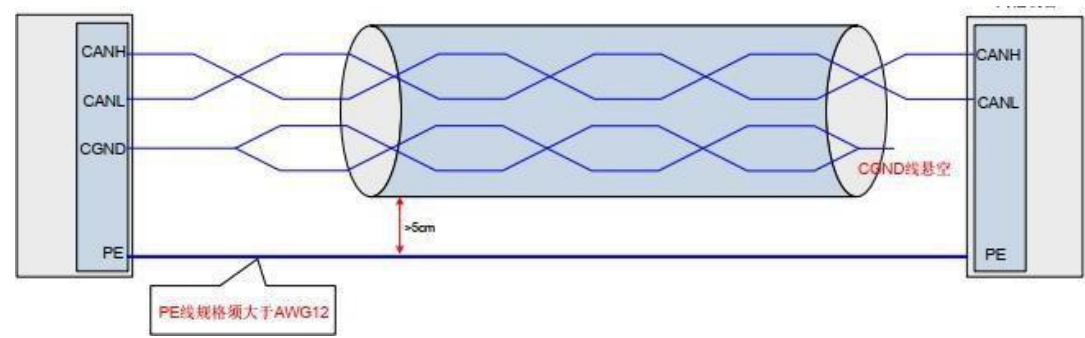


图 1.16 其它设备 CAN 无外接地端口

1.7 CAN 通信线缆布局推荐方式

CAN 通信属于易受干扰设备，如果现场布局时靠近干扰源，很容易出现问题。

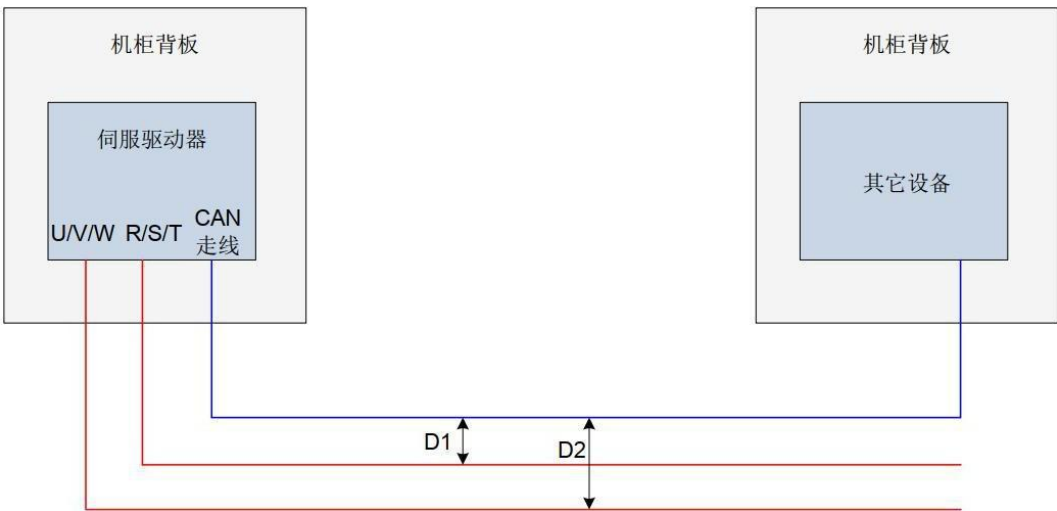


图 1.17 推荐布线方式

- 干扰线与 CAN 线尽量垂直方向走线，平行走线时，R/S/T 与 CAN 信号线距离 $D1 > 20\text{cm}$ ，U/V/W 与 CAN 信号线距离 $D2 > 50\text{cm}$ ；如果机柜背板，干扰线紧贴背板走线，CAN 通信线与机柜背板间隔大于 1cm。
- 出机柜后，R/S/T 电源线，U/V/W 动力线和 CAN 通信线，分别在三个走线槽中走线，走线槽间距 $L3 > 20\text{cm}$ 。如果干扰线和 CAN 通信线在同一线槽中走线，走线间距遵循上述相同原则。

第 2 章 通信网络配置

CANOPEN 使用设置流程如下：

开始 -> 导入 EDS 文件 -> 系统参数设置 -> 通信参数设置 -> 配置 PDO -> 启动远程节点
-> 结束

2.1 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。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 2.1 所示。



图 2.1 canopen 设备模型

2.1.1 对象字典

对象字典是设备规范中最重要的部分。它是一组参数和变量的有序集合，包含了设备描述及 设备网络状态的所有参数。通过网络可以采用有序的预定义的方式来访问的一组对象。CANopen 协议采用了带有 16 位索引和 8 位子索引的对象字典，对象字典的结构如表 2.2 所示。

索引	对象
000	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 (如设备类型, 错误寄存器, 支持的 PDO 数量)
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 DSP-402 协议)
A000h—FFFFh	保留

表 2.2 对象字典结构图

2.1.2 常用的通信对象

1) 网络管理对象 (NMT)

网络管理对象包括 Boot-up 消息, Heartbeat 协议及 NMT 消息, 基于主从通信模式, NMT 用于管理和监控网络中的各个节点, 主要实现三种功能: 节点状态控制、错误控制和节点启动。

2) 服务数据对象 (SDO)

- 1、包括接收 SDO(R-SDO) 和发送 SDO(T-SDO)。
- 2、通过使用索引和子索引, SDO 使客户机能够访问设备对象字典中的项。
- 3、SDO 通过 CAL 中多元域的 CMS 对象来实现, 允许传送任何长度的数据, 当数据超过 4 个字节时分拆成几个报文。
- 4、协议是确认服务类型, 为每个消息生成一个应答。SDO 请求和应答报文总是包含 8 个字节。

3) 过程数据对象 (PDO)

- 1、包括接收 PDO(RPDO) 和发送 PDO(TPDO)。
- 2、用来传输实时数据, 数据从一个创建者传到一个或多个接收者。数据传送限制在 1 到 8 个字节。

- 3、每个 CANopen 设备包含 8 个缺省的 PDO 通道， 4 个发送 PDO 通道和 4 个接收 PDO 通道。
- 4、PDO 包含同步和异步两种传输方式，由该 PDO 对应的通信参数决定。
- 5、PDO 消息的内容是预定义的，由该 PDO 对应的映射参数决定。

4) 同步对象 (SYNC)

同步对象是由 CANopen 主站周期性地广播到 CAN 总线的报文，用来实现基本的网络时钟信号，每个设备可以根据自己的配置，决定是否使用该事件来跟其它网络设备进行同步通信。

5) 紧急报文 (EMCY)

设备内部通信故障或者应用故障错误时发送的报文。

2.1.3 通信对象识别码

通信对象标识符 (COB-ID) 指定了在通信过程中对象的优先级以及通信对象的识别。COB-ID 与 CAN 2.0A 的 11 位帧 ID 一一对应， 11 位 COB-ID 由两部分组成，分别是 4 位的对象功能代码和 7 位的节点地址，如表 2.3 所示：

10	9	8	7	6	5	4	3	2	1	0
功能代码				节点 ID						

表 2.3 COB-ID 组成说明

CANopen 的各个通信对象都有默认的 COB-ID，可以通过 SDO 进行读取，部分可以通过 SDO 进行修改。对象列表如表 2.4 所示。

通信对象	功能代码	节点地址	COB-ID	相应对象索引
网络管理	0000b	0	0h	-
同步对象	0001b	0	80h	1005h, 1006h
紧急报文对象	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
T_SDO	1011b	1~127	580h + Node ID	1200h
R_SDO	1100b	1~127	600h + Node ID	1200h
网络管理错误控制	1110b	1~127	700h + Node ID	1016h, 1017h

表 2.4 对象 COB-ID 列表

举例：4 号从站 TPDO2 的 COB-ID 为 $280h + 4 = 284h$ 。

2.2 系统设置

为了能够使 DB100-CAN 系列伺服驱动器准确的接入 CANopen 现场总线网络，需要伺服驱动器的相关参数进行设置。

参数	名称	设定范围	默认设置值
PA-14	位置信息输入方式	0-4 4: 表示接收 CAN 总线输入信息	4
PA-105	CANOPEN 通信地址	1-127	1
PA-106	CAN 通信波特率设置	0-6 0: 20K; 1: 50K; 2: 100K; 3: 125K; 4: 250K; 5: 500K; 6: 1M。	5

2.3 网络管理系统（NMT）

网络管理系统 (NMT) 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，可配置包括本身在内的 CANopen 网络。

2.3.1 NMT 服务

CANopen 按照协议规定的状态机执行相应工作。其中，部分为内部自动实现转换，部分必须由 NMT 主机发送 NMT 报文实现转换，具体如下图。

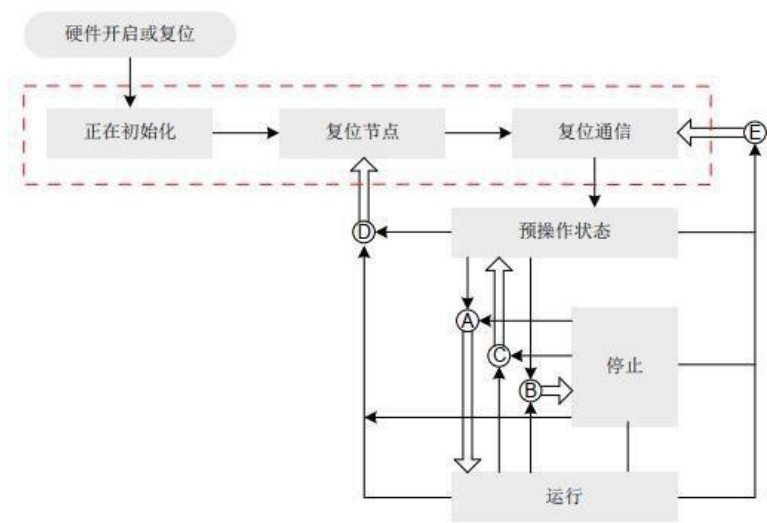


图 2.5 NMT 状态执行图

上图中带字母的转换由 NMT 报文实现，且只有 NMT 主机能够发送 NMT 控制报文，报文格式如表 2.6 所示。

COB-ID	RTR	Data/ 字节	
		0	1
0x000	0	命令字	Node_ID

表 2.6 NMT 报文格式

NMT 报文的 COB-ID 固定是“0x000”。

数据区由两个字节组成：第一个字节是命令字，表明该帧的控制作用，具体如表 2.7 说明；第二个字节是 CANopen 节点地址，当其为“0”时为广播消息，网络中的所有从设备均有效。

命令字	转向代号	说明
0x01	A	启动远程节点指令
0x02	B	停止远程节点指令
0x80	C	进入预操作状态指令
0x81	D	复位节点指令
0x82	E	复位通信指令

表 2.7 NMT 报文命令

设备上电后会进入初始化状态，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的参数加载，而复位节点将对象字典制造商定义区和子协议区恢复到上次保存值，复位通信将对象字典中通信参数恢复到上次保存值。

而后设备发送 **Boot-up**，自动进入预操作状态，此状态为主要的配置节点状态。

完成配置后，节点需要 NMT 主机发送 NMT 报文进入操作状态。操作状态是 CANopen 正常工作时的状态，各个模块都应正常工作。

当 NMT 主机发送停止节点报文时，设备进入停止状态，CANopen 通信只有 NMT 模块正常工作。

各种 NMT 状态下支持的 CANopen 服务如表 2.8 所示。

服务	预操作	操作	停止
过程数据对象 (PDO)	否	是	否
服务数据对象 (SDO)	是	是	否
同步对象 (SYNC)	是	是	否
紧急报文 (EMCY)	是	是	否
网络管理系统 (NMT)	是	是	是
错误控制	是	是	是

表 2.8 各种 NMT 状态下支持的服务

2.3.2 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、寿命保护和心跳。

注：1、不允许同时使用寿命保护和心跳。

2、节点保护，寿命保护和心跳保护的时间不宜设置过短，以免增大网络负载。

1) 节点寿命保护

节点保护是 NMT 主机通过远程帧，周期地查询 NMT 从机的状态；寿命保护则是从站通过

收到的用于监视从站的远程帧间隔来间接监视主站的状态。节点保护遵循的是主从模型，每个远程帧都必须得到应答。

与节点 / 寿命保护相关的对象包括保护时间 100Ch 和寿命因子 100Dh。100Ch 的值是正常情况下节点保护远程帧间隔，单位是 ms，100Ch 和 100Dh 的乘积决定了主机查询的最迟时间。正常情况下，节点保护都是可以实现的。当节点 100Ch 和 100D 都为非零，且接收到一帧节点保护请求帧时，激活寿命保护。

主站每隔 100Ch 时间发送节点保护远程帧从机必须做出应答，否则认为从站掉站；

从站 100Ch×100Dh 时间内未接收到节点保护远程帧，则认为主站掉站。

NMT 主节点发送远程帧如下表所示。

COB-ID	RTR
0x700+Node_ID	1

表 2.9 节点保护远程报文

NMT 从节点返回的应答报文如下表所示，数据段为一个字节的状态字，内容如表 2.11 所示。

COB-ID	RTR	Data
0x700+Node-ID	0	状态字

表 2.10 节点保护应答报文

数据位	说明
bit7	必须在每次中交替置“0”或者“1”
bit6~bit0	4- 停止状态 5- 操作状态 127- 预操作状态

表 2.11 应答报文状态说明

注：保护时间 100C 不建议低于 10ms，寿命因子必须不小于 2。

2) 心跳

心跳模式采用的是生产者——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。网络总具有消费者心跳功能的节点，根据对象 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，节点心跳功能激活，开始产生心跳报文。配置消费者心跳 1016h 的有效子索引后，接收到相应节点发出的一帧心跳即开始监视。

主机按其生产者时间发送心跳报文，监视主机的从机在 1016h 子索引时间内，未接收到心

跳报文，则认为主机掉站。 1016h 某子索引时间 $\geq$ 主机生产者时间 $\times 1.8$ ，否则易误报从机认为主机掉站。

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，则认为该从机掉站。 1017h $\times 1.8 \leq$ 监控该从机的主机（或其他从机）的消费者时间，否则易误报从机掉站。

心跳报文格式如表所示，数据段只含有一个字节，最高位固定为“0”，其它为与表 2.12 节点保护应答报文状态一致。

COB-ID	RTR	Data
0x700+Node-ID	0	状态字

表 2.12 心跳报文

伺服驱动器既是心跳生产者，也是心跳消费者，最多可以同时作为 5 个不同节点的心跳消费者。建议心跳生产者的时间不要低于 20ms，而消费者心跳时间不要低于 40ms，且为相应生产者心跳时间的 1.8 倍以上。

2.4 服务数据对象（SDO）

服务数据对象 (SDO) 通过对象索引和子索引与对象字典建立联系，通过 SDO 可以读取对象字典中的对象内容，或者在允许的情况下修改对象数据。

2.4.1 SDO 传输框架

SDO 传输方式遵循客户端——服务器模式，即一问一答方式。由 CAN 总线网络中的 SDO 客户端发起，SDO 服务器作出应答。因此，SDO 之间的数据交换至少需要两个 CAN 报文才能实现，而且两个 CAN 报文的 CAN 标识符不一样。SDO 的传输模型如图 2.13 所示。

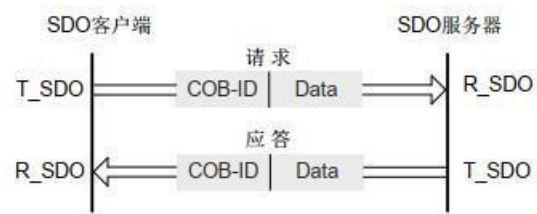


图 2.13 SDO 客户端读写 SDO 服务器上的对象字

2.4.2 SDO 传输报文

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。伺服驱动器只支持加速 SDO 传输和分段传输。

SDO 传输报文由 COB-ID 和数据段组成。由表 2.14 可以看出，T_SDO 和 R_SDO 报文的 COB-ID 不一致。数据段采用小端模式，即低位在前，高位在后排列。所有的 SDO 报文数据段都必须是 8 个字节。SDO 传输报文格式如下表：

COB-ID	Data							
580h+Node_ID/ 600h+Node_ID	0	1	2	3	4	5	6	7
	命令代码	索引		子索引	数据			

表 2.14 SDO 传输报文格式说明

其中，命令代码指明了该段 SDO 的传输类型和传输数据长度，索引和子索引是对象在列表的位置，数据是该对象的数值。

1) SDO 加速写传输报文

对于不高于 4 个字节的读写，采用加速 SDO 传输。按照读写方式及内容数据长度的一致，传输报文各不相同。加速 SDO 写报文如下表：

		COB-ID	0	1	2	3	4	5	6	7
客户端→		600h+Node_ID	23h	索引		子索引	数据			
			27h				数据			-
			2bh				数据		-	-
			2fh				数据	-	-	-
←服务器	正常	580h+Node_ID	60h	索引		子索引	-	-	-	-
	异常		80h				中止代码			

表 2.15 加速 SDO 写报文说明

注：“-”表示有数据但不予考虑，写数据时建议写 0，下同。

举例：从站站号为 4，用 SDO 写速度模式下运行速度值 60FFh-00，写入数值为 1000，即 0x3E8，主站发送报文如下。（所有均为 16 进制）

COB-ID	0	1	2	3	4	5	6	7
604	23	FF	60	00	E8	03	00	00

表 2.16 举例主站发送报文

若写入正常，则伺服驱动器返回如下报文。

COB-ID	0	1	2	3	4	5	6	7
584	60	FF	60	00	00	00	00	00

表 2.17 举例写入正常驱动器返回报文若

写入数据类型不匹配，则返回故障代码 0x06070010，报文如下。

COB-ID	0	1	2	3	4	5	6	7
584	80	FF	60	00	10	00	07	06

表 2.18 举例写入数据类型不匹配返回报文

2) SDO 加速读返回报文

SDO 读操作不高于 4 个字节的对象报文时，采用加速方式。加速 SDO 读报文如下表。

		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	正常	580h+Node_ID	43h	索引	子索引	数据				
			47h			数据			-	
			4bh			数据		-	-	-
			4fh			数据	-	-	-	-
	异常		80h			中止代码				

表 2.19 SDO 加速读报文格式说明

举例：从站站号为 4，用 SDO 读功能码最大转速限制，即对象 0x2006-08，主站发送报文如下。（所有均为 16 进制）

COB-ID	0	1	2	3	4	5	6	7
604	40	06	20	08	00	00	00	00

表 2.30 举例主站发送报文

最大转速默认值为 6000rpm，即 0x1770，正常情况时返回报文如下。

COB-ID	0	1	2	3	4	5	6	7
584	4b	06	20	08	70	17	00	00

表 2.31 举例正常时最大转速返回报文

若写入命令字不匹配，返回无效命令字错误，故障代码 0x05040001，报文如下。

COB-ID	0	1	2	3	4	5	6	7
584	80	06	20	08	01	00	04	05

表 2.32 举例写入命令不匹配返回报文

3) SDO 分段读传输报文

对于大于 4 个字节的对象，需要采用分段读操作来执行。分段传输报文结构与加速传输报

文类似，起始发送帧与加速传输保持一致。起始传输报文结构如下：

		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	正常	580h+Node_ID	41h	索引		子索引	数据长度			
	异常		80h				中止代码			

表 2.33 SDO 起始传输报文结构

传输过程由命令代码的触发位 (bit 6) 交互发送 0 和 1，必须保持此规律从站才给予响应。过程报文结构如下。

		COB-ID	0	1	2	3	4	5	6	7
客户端→		600h+Node_ID	60h	-	-	-	-	-	-	-
←服务器	正常	580h+Node_ID	00h	数据长度						
	异常		80h	索引		子索引	中止代码			
客户端→		600h+Node_ID	70h	-	-	-	-	-	-	-
←服务器	正常	580h+Node_ID	10h	数据长度						
	异常		80h	索引		子索引	中止代码			

表 2.34 SDO 传输过程报文结构

分段传输的末尾帧应答包含有末尾帧标志及末尾帧的有效数据长度，其传输报文结构如下。

		COB-ID	0	1	2	3	4	5	6	7
客户端→		600h+Node_ID	60h/0x70h	索引		子索引	-	-	-	-
←服务器	正常	580h+Node_ID	01h/11h	数据						
			03h/13h	数据						-
			05h/15h	数据					-	-
			07h/17h	数据				-	-	-
			09h/19h	数据			-	-	-	-
			0Bh/1Bh	数据		-	-	-	-	-
			0Dh/1Dh	数据	-	-	-	-	-	-
	异常		80h	索引		子索引	中止代码			

表 2.35 SDO 分段传输末尾帧报文结构

2.5 过程数据对象（PDO）

过程数据对象 (PDO) 用来传输实时的数据，是 CANopen 中最主要的数据传输方式。由于 PDO 的传输不需要应答，且 PDO 的长度可以小于 8 个字节，因此传输速度快。

PDO 的映射配置遵循流程如下：

- 1、开始；2、无效 PDO，即 PDO 的 COB-ID 最高位写 1；3、清除原有的映射内容，即对映射

对象的子索引 0 写入 “0”；4、写 PDO 映射内容，即按照映射定义分别写入映射参数子索引 1~8；5、写入该 PDO 映射对象总个数，即将上一步骤中写入的映射个数写到映射对象子索引 0；6、有效该 PDO，即 PDO 的 COB-ID 最高位写为 “0”；7、结束。

2.5.1 PDO 传输框架

PDO 的传输遵循的是生产者消费者模型，即 CAN 总线网络中生产者产生的 TPDO 可根据 COB-ID 由网络上一个或者多个消费者 RPDO 接收，传输模型如下图所示。

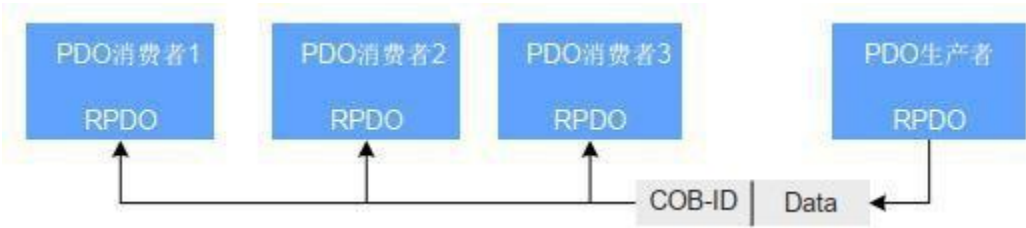


图 2.36 PDO 传输模型

目前，在伺服驱动器中，CANopen 通信只支持点对点的 PDO 传输方式。

2.5.2 PDO 传输对象

按照接收与发送的不同，PDO 可分为 RPDO 和 TPDO。PDO 由通信参数和映射参数共同决定最终传输的方式及内容。伺服驱动器使用了 4 个 RPDO 和 4 个 TPDO 来实现 PDO 的传输，相关对象列表如下。

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 2.37 驱动器 PDO 对象列表

2.5.3 PDO 通信参数

1) PDO 的 CAN 标识符

PDO 的 CAN 标识符即 PDO 的 COB-ID，包含控制位和标识数据，确定该 PDO 的总线优先级。COB-ID 位于通信参数 (RPDO: 1400h~1403h, TPDO: 1800h~1803h) 的子索引 01 上，最高位决定该 PDO 是否有效。

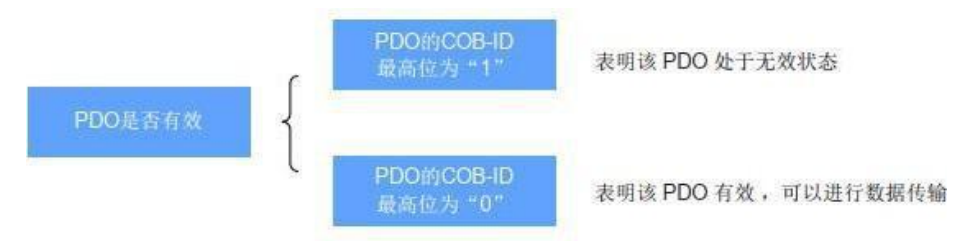


图 2.38 PDO 是否有效说明

伺服驱动器只支持点对点的 PDO 传输，因此 COB-ID 低 4 位必须为该节点的站号地址。

举例：对于站号为 4 的节点，TPDO3 在无效状态下其 COB-ID 应该为“80000384h”，而对该 COB-ID 写入“384h”时，表明激活该 PDO。

2) PDO 的传输类型

PDO 的传输类型位于通信参数 (RPDO: 1400h~1403h, TPDO: 1800h~1803h) 的子索引 02 上，决定该 PDO 遵循何种传输方式。

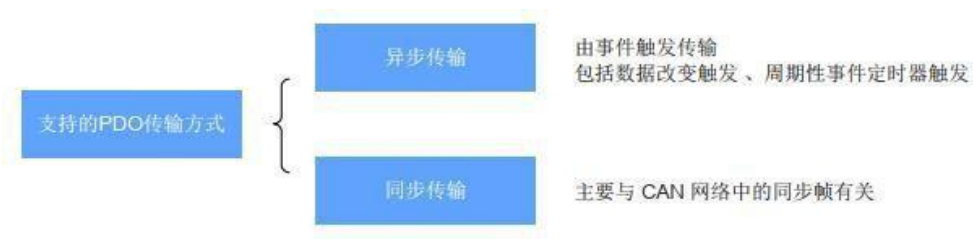


图 2.39 支持的 PDO 传输方式

通信参数 (RPDO: 1400h~1403h, TPDO: 1800h~1803h) 子索引 02 不同的数值代表不同的传输类型，定义了触发 TPDO 传输或处理收到的 RPDO 的方法，具体对应关系如表所示。

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

表 2.40 TPDO 和 RPDO 触发方法

- 当 TPDO 的传输类型为 0 时，如果映射数据发生改变，且接收到一个同步帧，则发送该 TPDO；
- 当 TPDO 的传输类型为 1~240 时，接收到相应个数的同步帧时，发送该 TPDO。
- 当 TPDO 的传输类型是 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 。
- 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。

3) 禁止时间

针对 TPDO 设置了禁止时间，存放在通信参数 (1800h~1803h) 的子索引 03 上，防止 CAN 网络被优先级较低的 PDO 持续占有。该参数的单位是 100us，设置数值后，同一个 TPDO 传输间隔减不得小于该参数对应的时间。

举例：TPDO2 的禁止时间为 300，则 TPDO 的传输间隔不会小于 30ms。

4) 事件计时器

针对异步传输 (传输类型为 254 或 255) 的 TPDO， 定义事件计时器， 位于通信参数 (1800h~1803h) 的子索引 05 上。事件计时器也可以看做是一种触发事件，它也会触发相应的 TPDO 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，且事件计数器会被立即复位。

2.5.4 PDO 映射参数

PDO 映射参数包含指向 PDO 需要发送或者接收到的 PDO 对应的过程数据的指针，包括索引、子索引及映射对象长度。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 0 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下。

位数	31		16	15		8	7		0
含义	索引			子索引			对象长度		

表 2.41 PDO 映射参数内容定义位数

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，即：

对象长度	位长
08h	8 位
10h	16 位
20h	32 位

表 2.41 对象长度与对象位长关系表

2.6 同步对象 (SYNC)

同步对象 (SYNC) 是控制多个节点发送与接收之间谐调和同步的一种特殊机制，用于 PDO 的同步传输。

同步发生器的配置流程如下：

1、开始；2、关闭同步发生器，即同步对象 COB-ID(1005h)写入“80h”；3、写入同步循环周期 1006h 对应数值，单位为 us，不可为 0，否则下一步无法执行；4、打开同步发生器，即同步对象 COB-ID(1005h)写入 40000080h；5、结束。

注：伺服驱动器不支持使用循环周期低于 500us 的同步发生器，不建议使用低于 1ms 的同步循环周期。

2.6.1 同步发生器

伺服驱动器不仅是同步消费者，也可以是同步生产者。支持与同步相关的对象分别是同步对象 COBID(1005h) 和同步循环周期 (1006h)。

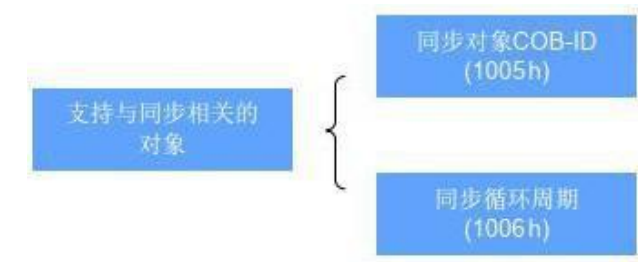


图 2.42 支持与同步相关的对象说明

同步对象 COB-ID 的次高位决定是否激活同步发生器。为 1 时激活，0 关闭。

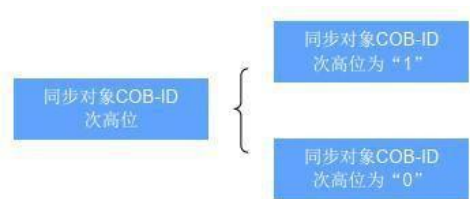


图 2.43 激活同步发生器说明

同步循环周期只针对于同步发生器，单位为 us，表明节点产生同步对象时的间隔。

2.6.2 同步对象传输框架

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。同步对象的传输框架如图 2.44 所示。



图 2.44 同步对象传输框架

同步 PDO 的传输与同步帧紧密联系。

- 对于同步 RPDO，只要接收到了该 PDO，在下一个 SYNC 时将接收到的 PDO 更新到应用。
- 对于同步 TPDO，分为同步循环和同步非循环。



图 2.45 同步 TPDO 说明图

同步传输模型示意图如图 2.46 所示。

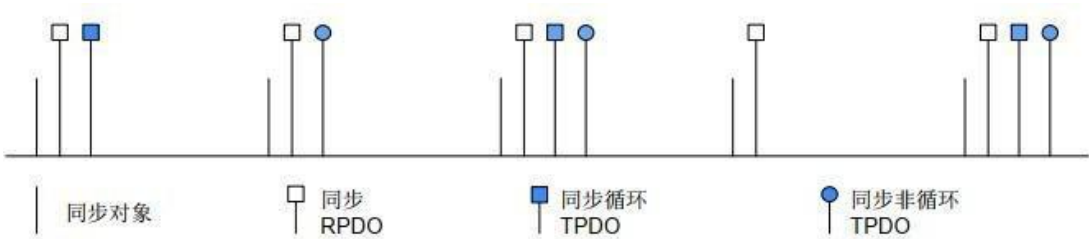


图 2.46 同步传输模型

举例：

RPDO1 的传输类型为 0，RPDO2 的传输类型为 5，TPDO1 的传输类型为 0，TPDO2 的

传输类型为 20。则 RPDO1 和 RPDO2 只要接收到 PDO，会在下一个 SYNC 时将最新的PDO 数据更新到相应的应用；而 TPDO1 的映射数据只有发生了改变，会在下一个 SYNC 时发送 TPDO1， TPDO2 累计经历 20 个 SYNC 时，不管数据有无改变，均会发送 PDO。

2.7 紧急对象服务（EMCY）

当 CANopen 节点出现错误时，按照标准化机制，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。

紧急对象的 COB-ID(1014h)，最高位为 1 表明关闭节点紧急报文，为 0 表明打开紧急报文。

COB-ID	0	1
80h + Node_ID	错误码	

图 2.47 紧急报文格式

驱动器出现异常时，如果打开紧急报文，则会发送一帧紧急报文数据，数据长度为 2，低位在前高位在后，组合成的 16 位数即为驱动器的报警代码。

第 3 章 运动模式

3.1 设置运行比例

未上使能时，设置齿轮比 6091，设置齿轮比分子与电机编码器精度一致，默认 17 位编码器，设置值为 131072。设置齿轮比分母与用户单位一致，默认 100000，即发送 100000 数据增量，电机旋转一圈。

3.2 伺服状态控制

3.2.1 CIA402 状态机

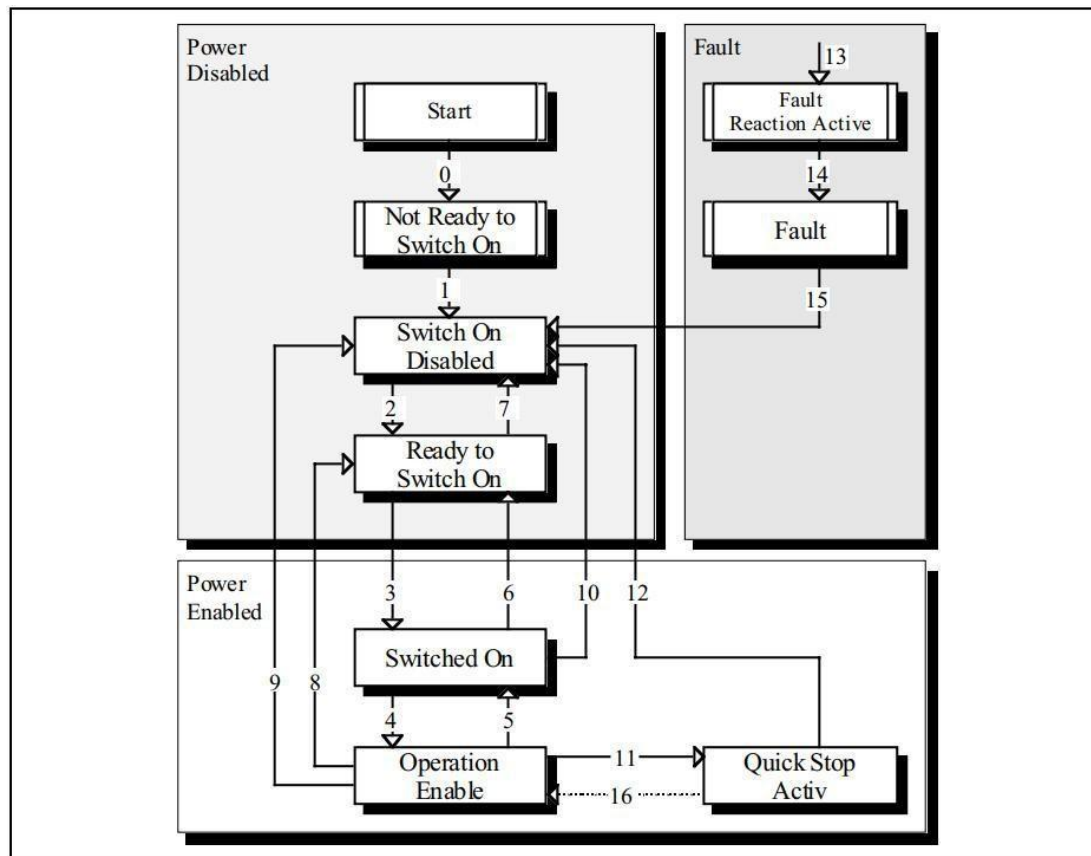


图 3.1 CIA402 状态机

3.2.2 控制字 6040h

0	1	2	3	4,5,6	7	8	9,10	11-15
Switch on	Enable voltage	Quick stop	Enable operation	Operation mode specific	Fault reset	Halt	Reserved	Manufacturer specific
M	M	M	M		M			

BIT0-BIT3 和 BIT7，相关操作及状态转换如下所示：

Command	Bit of the <i>controlword</i>					Transitions
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on	
Shutdown	0	X	1	1	0	2,6,8
Switch on	0	0	1	1	1	3*
Switch on	0	1	1	1	1	3**
Disable voltage	0	X	X	0	X	7,9,10,12
Quick stop	0	X	0	1	X	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset		X	X	X	X	15

图 3.2 控制字及状态转换示意图

BIT456 和 BIT8，

位	运行模式				
	轮廓位置模式	轮廓速度模式	轮廓转矩模式	回零模式	插补模式
4	新位置使能	保留	保留	启动回零	使能插补模式
5	非立即执行	保留	保留	保留	保留
6	相对/绝对	保留	保留	保留	保留
8	暂停	暂停	暂停	暂停	暂停

3.2.3 状态字 6041h

Bit	名称	描述
0	准备打开	
1	等待打开使能	
2	运行使能	

3	错误	
4	电源使能	
5	快速停止	
6	伺服准备好	
7	警告	
8	厂家自定义	
9	远程定义	1 canopen 远程控制模式
10	目标到达	0: 位置或速度未到达 1: 位置或速度到达
11	软限位	0: 软限位未超限 1: 软限位超限
12-13	模式相关	
14-15	预留	

Bit12 和 bit13

Bit	运行模式				
	轮廓位置模式	轮廓速度模式	轮廓转矩模式	回零模式	插补模式
12	位置运行中	保留	保留	回零完成	插补有效
13	跟随错误	速度超限	保留	回零错误	保留

3.3 伺服运行模式概述

伺服预运行模式可通过对象字典 6060h 进行设置。伺服当前运行模式可通过对象字典 6061h 进行查看。

伺服支持运行模式如下：

- 1：轮廓速度模式；
- 3：轮廓速度模式；
- 4：轮廓转矩模式；
- 6：回零模式；

7：插补模式；

设置了不支持的伺服运行模式，驱动器模式显示 6061h 将显示为 0。

3.4 轮廓位置模式

3.4.1 相关对象

此模式主要用于点对点定位应用。此模式下，上位机给目标位置（绝对或者相对）、位置曲线的速度、加减速及减速度，伺服内部的轨迹发生器将根据设置生成目标位置曲线指令，驱动器内部完成位置控制，速度控制，转矩控制。

控制字 6040		
位	名称	描述
0	伺服准备好	4 位全为 1，表示伺服当前主电路充电，处于使能状态
1	接通主电路	
2	快速停机	
3	伺服运行	
4	新目标位置	从 0-->1 表示有新的位置
5	立即更新	0：非立即 1：立即 (驱动器只支持非立即更新功能)
6	绝对位置/相对位置	0：目标位置为绝对位置 1：目标位置为相对位置
状态字 6041		
位	名称	描述
10	目标到达	0：目标位置未到达 1：目标位置已到达
12	目标位置更新	0：可更新目标位置 1：不可更新目标位置
13	跟随误差	0：没有位置偏差过大故障 1：发生位置偏差过大故障

3.4.2 位置曲线发生器

控制指令时序---非立刻更新型

- a) 上位机首先更新根据需要修改位移指令的其他属性 (加速时间 6083, 减速时间 6084, 最大运行速度 6081, 目标位移 607A)。
- b) 上位机将 6040 的 bit4 由 0 置 1, 提示从站有新的位移指令需要使能。
- c) 从站在接收到 6040 的 bit4 的上升沿后, 对是否可接收该新的位移指令做出判断:
若 6040 的 bit5 的初始状态为 0, 且此时 6041 的 bit12 为 0, 表明从站可接收新的位移指令①; 从站接收新的位移指令后, 将 6041 的 bit12 由 0 置 1, 表明新的位移指令①已接收, 且当前从站处于不能继续接收新位移指令状态。
- d) 上位机接收到状态字 6041 的 bit12 变为 1 后, 可以释放位移指令数据, 并将控制字 6040 的 bit4 由 1 置 0, 表明当前无新的位置指令。由于 6040 的 bit4 为沿变化有效, 因此, 此操作不会中断正在执行的位移指令。

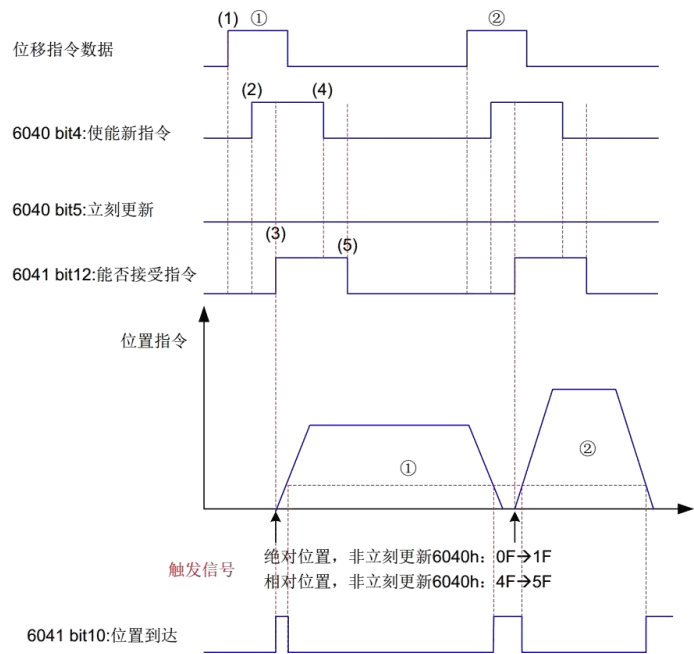


图 3.3 非立即更新型时序图和电机运行曲线

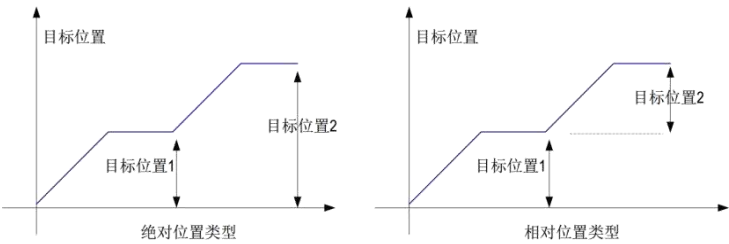
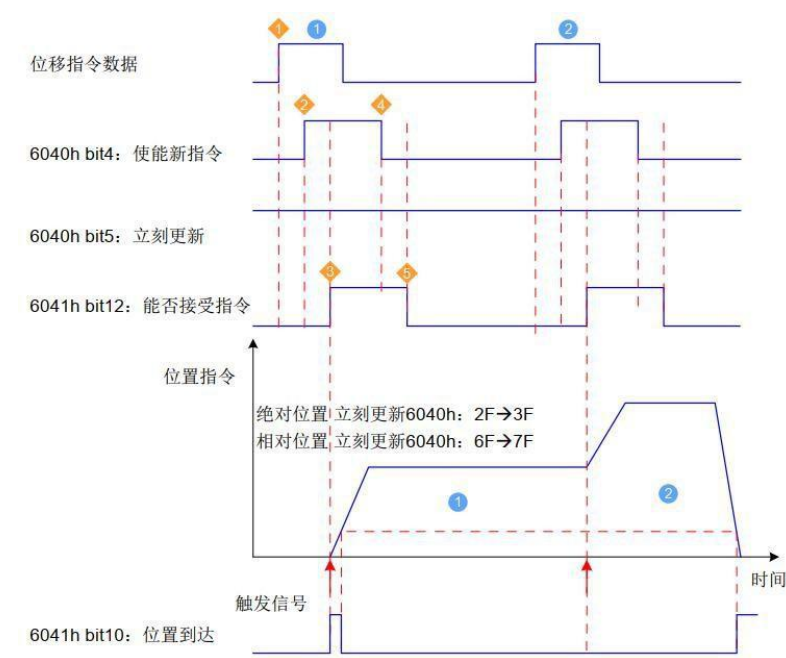


图 3.4 绝对位置指令和相对位置指令的区别

控制指令时序---立刻更新型



a) 上位机首先更新根据需要修改位移指令的其他属性 (加速时间 6083, 减速时间 6084, 最大运行速度 6081, 目标位移 607A)。

b) 上位机将 6040 的 bit4 由 0 置 1, 提示从站有新的位移指令需要使能。

c) 从站在接收到 6040 的 bit4 的上升沿后, 对是否可接收该新的位移指令做出判断:

若 6040 的 bit5 的初始状态为 0, 且此时 6041 的 bit12 为 0, 表明从站可接收新的位移指令①; 从站接收新的位移指令后, 将 6041 的 bit12 由 0 置 1, 当 6040 的 bit4 置 0 时, 6041bit12 置 0, 此时电机处于运转时依然可以响应控制指令。

3.4.3 建议配置:

RPDO	TPDO	备注
6040: 控制字	6041: 状态字	必选
607A: 目标位置	6064: 位置反馈	必选
6081: 轮廓速度		必选
6083: 轮廓加速度		必选
6060: 模式选择	6061: 运行模式显示	必选

3.5 轮廓速度模式

3.5.1 相关对象

此模式下，上位机将目标速度、加速度发送给伺服驱动器，速度、转矩调节由伺服内部执行。

控制字 6040		
位	名称	描述
0	伺服准备好	4 位全为 1，表示伺服当前主电路充电，处于使能状态
1	接通主电路	
2	快速停机	
3	伺服运行	

状态字 6041		
位	名称	描述
10	目标到达	0: 目标速度未到达 1: 目标速度已到达
11	软件内部限位	0: 未到达软限位 1: 到达软限位

注：轮廓加速度应用于速度模式的加减速，加速度和减速度统一由轮廓加速度设置。

3.5.2 建议配置

轮廓速度模式，基本配置如下：

RPDO	TPDO	备注
6040：控制字	6041：状态字	必选
607A：目标位置		可选
	6064：位置反馈 606C：速度反馈	可选
60FF：目标速度		必选

6083: 轮廓加速度		可选
6060: 模式选择	6061: 运行模式显示	可选

3.6 轮廓转矩模式

此模式下，上位控制器将目标转矩 6071h、转矩斜坡常数 6087h 发送给伺服驱动器，转矩调节由伺服内部执行。当速度达到限幅值将进入调速阶段。

3.6.1 相关对象

控制字 6040		
位	名称	描述
0	伺服准备好	4 位全为 1，表示伺服当前主电路充电，处于使能状态
1	接通主电路	
2	快速停机	
3	伺服运行	

状态字 6041		
位	名称	描述
10	目标到达	0: 目标转矩未到达
		1: 目标转矩已到达
11	软件内部限位	0: 未到达软限位
		1: 到达软限位

3.6.2 建议配置

轮廓转矩模式（PT），基本配置如下：

RPDO	TPDO	备注
6040: 控制字	6041: 状态字	必选
6071: 目标转矩		必选

6087：转矩斜坡		
	6064：位置反馈 606C：速度反馈 6077：转矩反馈	可选
6060：模式选择	6061：运行模式显示	必选

3.7 回零模式

原点回零模式用于寻找机械原点，并定位机械原点与机械零点的位置关系。

机械原点：机械上某一固定的位置，可对应某一确定的原点开关，可对应电机 Z 信号。

机械零点：机械上绝对 0 位置。

原点回零成后，电机停止位置为机械原点，此时 607Ch 自动设定为机械原点与机械零点关系：

机械原点 = 机械零点 + 607Ch(原点偏置)，当 607Ch=0 时，机械原点与机械零点重合。

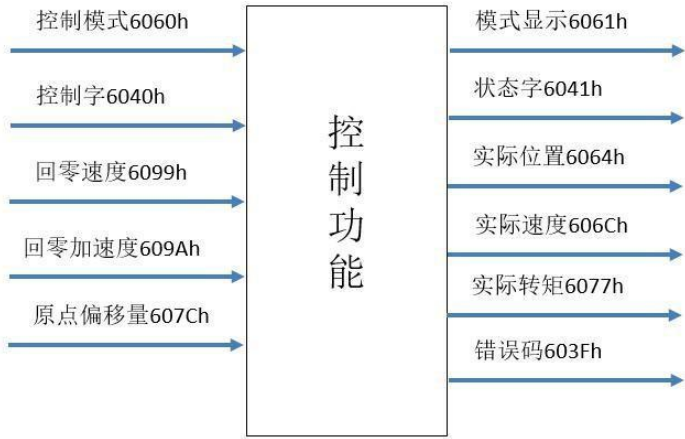


图 3.5 原点回归模式

3.7.1 相关对象

控制字 6040		
位	名称	描述
0	伺服准备好	4 位全为 1，表示伺服当前主电路充电，处于使能状态

1	接通主电路	
2	快速停机	
3	伺服运行	
4	启动回零	0-->1: 启动回零 1-->0: 驱动接收到回零信号

状态字 6041		
位	名称	描述
10	目标到达	0: 目标位置未到达 1: 目标位置已到达
12	回零	0: 回零已完成, 可接收回零信号 1: 回零进行中, 不可接收回零信号
13	回零错误	0: 回零没发生错误 1: 回零发生超时或偏差过大错误

3.7.2 建议配置

RPDO	TPDO	备注
6040: 控制字	6041: 状态字	必选
6098: 回零方式		可选
6099-01: 搜索限位开关信号速度		可选
6099-02: 搜索零点信号速度		可选
609A: 回零加速度		可选
	6064: 位置反馈 606C: 速度反馈 6077: 转矩反馈	可选
6060: 模式选择	6061: 运行模式显示	必选

3.7.3 回零方式说明

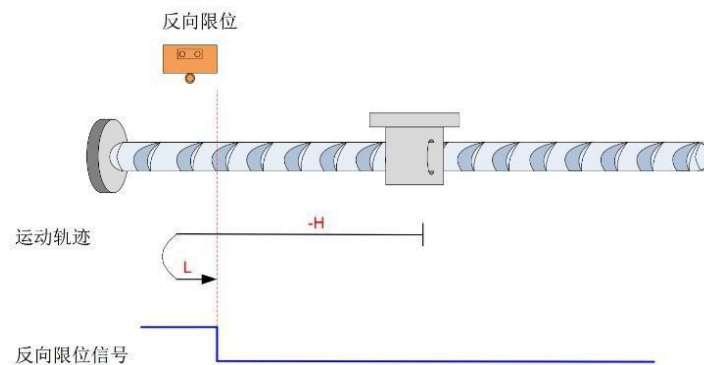
伺服目前支持的回零模式只有 5 种，17,18,23,27,35。

1) 6098h=17

机械原点：反向超程开关

减速点：反向超程开关

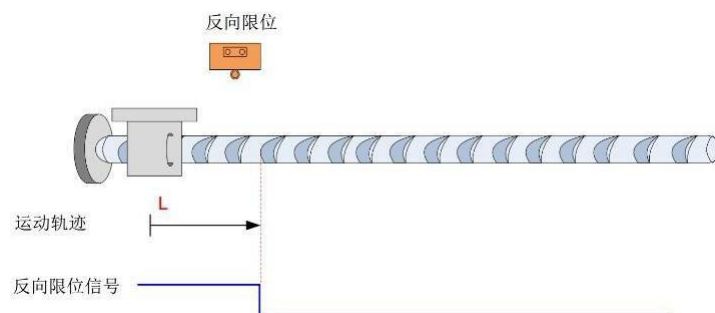
回零启动时减速点信号无效。



注：图中“H”代表高速 6099-1h，“L”代表低速 6099-2h。

开始回零时 $N-OT=0$ ，以反向高速开始回零，遇到 $N-OT$ 上升沿后，减速，反向，正向低速运行，遇到 $N-OT$ 下降沿后停机。

回零启动时减速点信号有效。



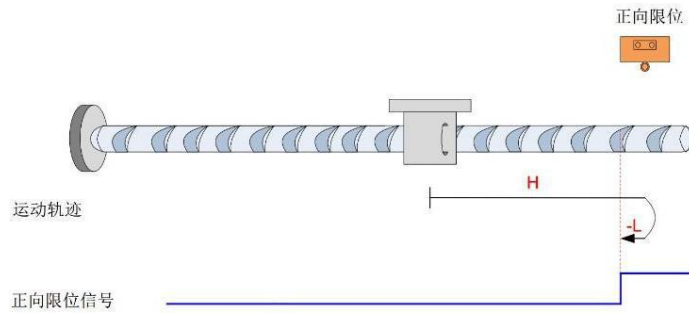
回零启动时 $N-OT=1$ ，直接正向低速开始回零，遇到 $N-OT$ 下降沿后停机。

2) 6098h=18

原点：正向超程开关

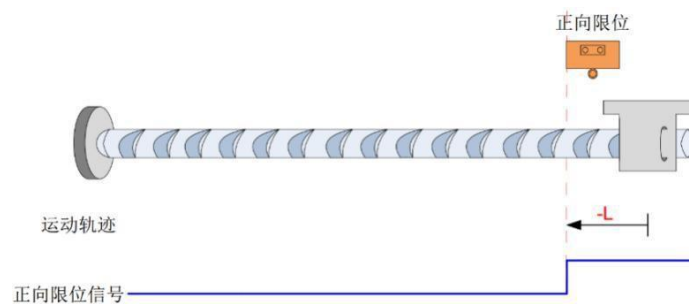
减速点：正向超程开关

回零启动时减速点信号无效。



开始回零时 $P-OT=0$ ，以正向高速开始回零，遇到 $P-OT$ 上升沿后，减速，反向，反向低速运行，遇到 $P-OT$ 下降沿后停机。

回零启动时减速点信号有效。



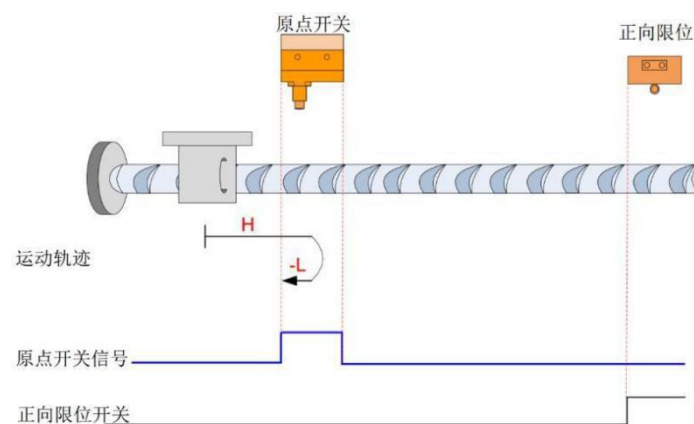
回零启动时 $P-OT=1$ ，直接反向低速开始回零，遇到 $P-OT$ 下降沿停机。

3) 6098h=23

原点：原点开关

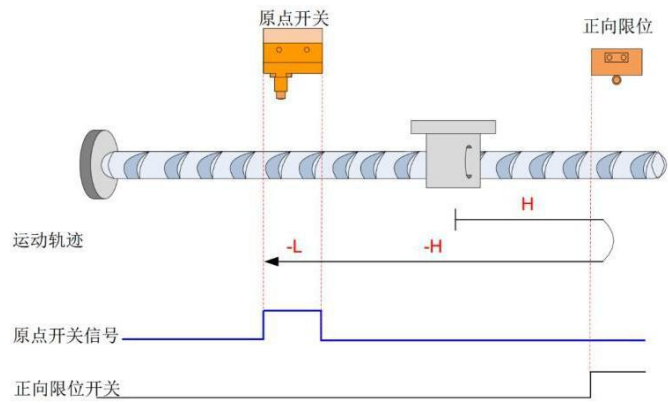
减速点：原点开关

回零启动时减速点信号无效，未遇到正向限位开关。



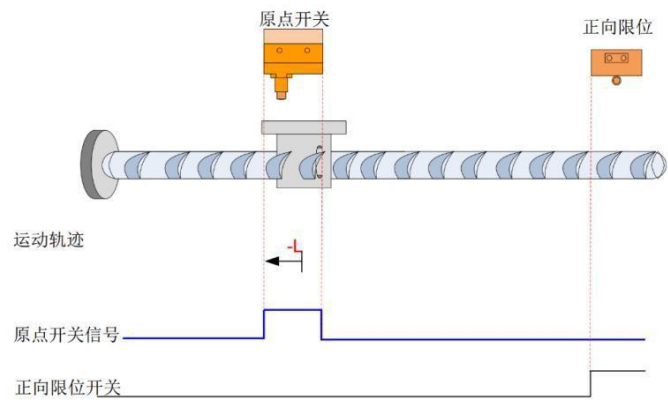
开始回零时 $HW=0$ ，以正向高速开始回零，未遇到限位开关，遇到 HW 上升沿后，减速，反向，反向低速运行，遇到 HW 下降沿停机。

回零启动时减速点信号无效，遇到正向限位开关。



开始回零时 $HW=0$ ，以正向高速开始回零，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，继续反向低速运行，遇到 HW 下降沿停机。

回零启动时减速点信号有效。



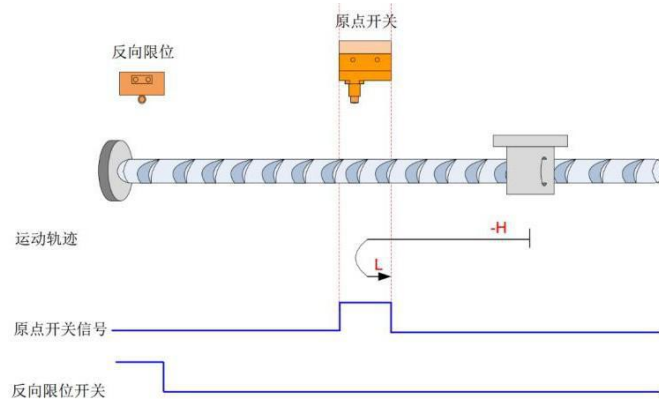
回零启动时 $HW=1$ ，则直接反向低速开始回零，遇到 HW 下降沿停机。

4) 6098h=27

原点：原点开关

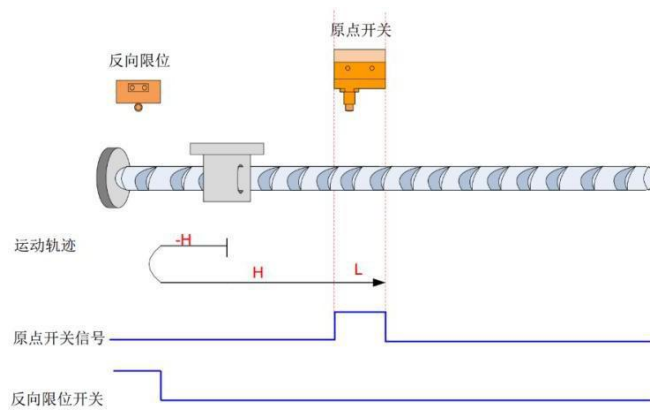
减速点：原点开关

回零启动时减速点信号无效，未遇到反向限位开关。



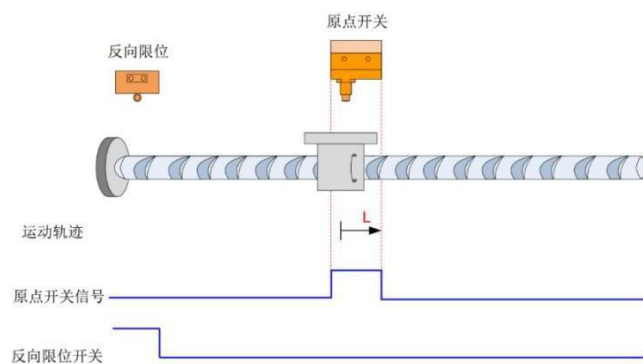
开始回零时 $HW=0$ ，以反向高速开始回零，未遇到限位开关，遇到 HW 上升沿后，减速，反向，正向低速运行，遇到 HW 下降沿停机。

回零启动时减速点信号无效，遇到反向限位开关。



开始回零时 $HW=0$ ，以反向高速开始回零，遇到限位开关，自动反向，正向高速运行，遇到 HW 上升沿后，减速，继续正向低速运行，遇到 HW 下降沿停机。

回零启动时减速点信号有效。



回零启动时 $HW=1$ ，则直接正向低速开始回零，遇到 HW 下降沿停机。

5) 6098h=35

当前位置清 0。

3.8 插补模式

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，根据实际应用需要，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。

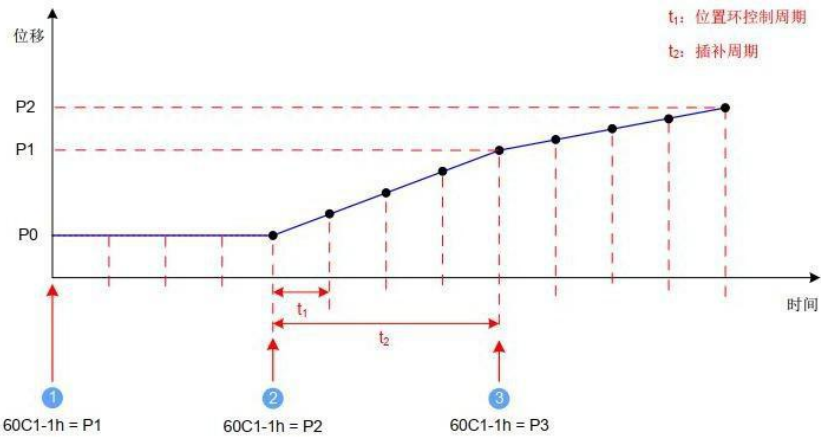


图 3.6 单轴线性插补电机位移曲线

注 1： 1 - 伺服电机当前绝对位置为 P0，接收到第一个绝对位置指令 P1，开始规划第一段位移曲线。

注 2： 2 - 伺服电机当前绝对位置为 P0，开始向第一个绝对位置 P1 运动，同时接收到第二个绝对位置指令 P2，规划第二段位移曲线。

注 3： 3 - 伺服电机到达第一个绝对位置 P1，开始向第二个绝对位置 P2 运动，同时接收到第三个绝对位置指令 P3，规划第三段位移曲线。

注 4： t1- 位置环控制周期，由伺服驱动器内部决定。

注 5： t2- 插补周期，可通过对象字典 60C2h 设定。伺服驱动器支持的同步周期： 1-20ms，当设置了在此范围之外的同步周期时，同步周期将被设定在限定值。

注 6： P0/P1/P2- 绝对位置，绝对位置指令通过对象字典 60C1-1h 发送，插补模式只支持绝对位置指令。

注 7： 每个同步周期的位移指令增量分别为： P1-P0， P2-P1。

第 4 章 故障处理

4.1 常见驱动器故障代码

故障符号	故障名称	故障内容
--	正常	
1	超速	伺服电机速度超过设定值
2	主电路过压	主电路电源电压过高
3	主电路欠压	主电路电源电压过低
4	位置超差	位置偏差计数器的数值超过设定值
5	驱动器过热	驱动器温度过高
6	速度放大器饱和故障	速度调节长时间饱和
7	驱动禁止异常	CCW/CW 驱动禁止输入都 OFF
8	位置偏差计数器溢出	位置偏差计数值的绝对值超过 2^{30}
11	IPM 模块故障	IPM 智能模块故障
13	驱动器过负载	伺服驱动器及电机过负载(瞬时过热)
14	制动故障	制动电路故障
18	继电器开关故障	继电器实际状态跟控制状态不一致
19	抱闸延时错误	抱闸未打开时有脉冲输入
20	EEPROM 错误	EEPROM 错误
21	FPGA 模块故障	FPGA 模块功能异常
23	电流采集电路故障	电流采集电路故障
29	用户转矩过载报警	电机负载超过用户设定的数值和持续时间
39	数据 CRC 校验错误	电机编码器尚未写入数据，都是 0
40	不支持机型	驱动器不支持此电机型号
41	需要切换电机型号	当前电机跟驱动已选的型号不一致。
42	AC 输入电压过低	AC 输入电压过低
47	上电时主电路电压过高	上电时主电路电压过高

50	编码器通信故障	驱动器跟编码器未建立通信连接
51	编码器通信异常	编码器建立通信后，出现中断，断开连接
52	编码器电池电压不足报警	编码器电池电压不足报警，信息未丢但需尽快更换
53	编码器电池电压错误报警	编码器电池电压错误报警，储存信息已发生错误，需复位编码器
54	编码器错误报警	编码器非电池类报警，但需要重新复位编码器
55	CRC 校验连续 3 次出错	编码器通信收到的数据CRC 效验连续 3 次出现错误
56	MODBUS 帧过长错误	接收到的MODBUS 帧数据过长
57	MODBUS 通讯格式异常	通讯参数设定不当或地址或数值不正确
58	单圈位置值错误	驱动器存储的单圈位置偏移值超出编码器分辨率
59	编码器上报 CF 错误	编码器连续上报CF 域错误，需要复位编码器
61	使能状态下 CAN 状态异常	使能通信中出现异常
62	同步帧丢失异常	启用同步信号之后，出现同步帧丢失

第 5 章 对象字典

5.1 对象分类说明

索引”：指定各个对象在对象字典中的位置，以十六进制表示。

“数据类型”：具体请参见表 5.1。

数据类型	数值范围	数据长度	DS301 值
Int8	-128~+127	1 字节	0002
Int16	-32768~+32767	2 字节	0003
Int32	-2147483648~+ 2147483647	4 字节	0004
UInt8	0~255	1 字节	0005
UInt16	0~65535	2 字节	0006
UInt32	0~4294967295	4 字节	0007
String	ASCII	-	0009

表 5.1 数据类型说明

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

表 5.2 读写类型说明

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

表 5.3 对象分类说明

5.2 对象 1000h 组分类说明

索引	子索引	名称	可访问性	能否映射	数据类型	单位	数据范围	出厂设定
1000h	-	设备类型	RO	NO	UINT32	VAR	UIINT32	0
1001h	-	错误寄存器	RO	NO	UINT8	VAR	UINT8	0
1005h	-	同 步 报 文	RW	NO	UINT32	VAR	UINT32	0x80

		COB-ID																		
1006h	-	同步循环周期	RW	NO	UINT32	VAR	UINT32	0												
1008h	-	制造商设备名称	RO	NO	STRING	VAR	STRING	DVS_CANOPEN												
1009h	-	硬件版本	RO	NO	STRING	VAR	STRING	V1.0												
100Ah	-	软件版本	RO	NO	STRING	VAR	STRING	V1.01												
100Ch	-	节点守护时间（单位：ms）	RW	NO	UINT16	VAR	UINT16	0												
100Dh	-	寿命因子（须大于 1）	RW	NO	UINT8	VAR	UINT8	0												
1014h	-	紧急报文 COB-ID	RW	NO	UINT32	VAR	UINT32	0x80+Node_ID												
		最高位表明是否需要关闭设备的紧急报文，只可以该位写入数据为“0x80+Node_ID” 打开设备紧急报文；写入数据为“0x80000080+Node_ID”，关闭紧急报文。紧急报文生效时，其 COB-ID 必须与此对象保持一致。																		
1016h	-	消费者心跳时间（单位：ms）	RW	NO	-	ARR	-	-												
	00h	最大子索引	RO	NO	UINT8	VAR	1	1												
	01h	消费者心跳时间	RW	NO	UINT32	VAR	UINT32	0												
		参数包括监视的节点地址以及实际消费者时间，且该时间必须大于对应节点的心跳生产者时间（单位： ms）。不可以对同一个节点设置两个消费者时间。 <table><tr><td>31</td><td>24</td><td>23</td><td>16</td><td>15</td><td>0</td></tr><tr><td colspan="2">保留 (0)</td><td colspan="2">被监视地址</td><td colspan="2">监视时间</td></tr></table> MSBLSB							31	24	23	16	15	0	保留 (0)		被监视地址		监视时间	
31	24	23	16	15	0															
保留 (0)		被监视地址		监视时间																
1017h	-	生产者心跳时间	RW	NO	UINT16	VAR	UINT16	0												
1018h	-	设备对象描述	RO	NO	-	REC	-	-												

	01h	厂商 ID	RO	NO	UINT32	VAR	UINT32	0x445653								
	02h	设备代码	RO	NO	UINT32	VAR	UINT32	1								
	03h	修订版本号	RO	NO	UINT32	VAR	UINT32	1								
	04h	序列号	RO	NO	UINT32	VAR	UINT32	1								
1200h	-	SDO 服务器参数	RO	NO	-	REC	-	-								
	00h	最大子索引	RO	NO	UINT8	VAR	2	2								
	01h	客户端到服务器 COB-ID	RO	NO	UINT32	VAR	UINT32	0x600+ Node_ID								
	02h	服务器到客户端 COB-ID	RO	NO	UINT32	VAR	UINT32	0x580+ Node_ID								
1400h-1403h	-	RPDO 通信参数	RW	NO	-	REC	-	-								
	00h	最大子索引	RO	NO	UINT8	VAR	6	6								
	01h	RPDO 的 COB-ID	RW	NO	UINT32	VAR	UINT32	见下文								
		只可以改变最高位，最高位为“0”表明该 PDO 有效，最高位为“1”表明该 PDO 无效。出厂设定如下： 1400h:0x00000200+Node_ID 1401h:0x80000300+Node_ID 1402h:0x80000400+Node_ID 1403h:0x80000500+Node_ID														
	02h	RPDO 的 传输类型	RW	NO	UINT8	VAR	UINT8	255								
		只可以在 PDO 无效的状态下才可以修改此数值。不同的数值代表不同的 PDO 传输类型，如下表： <table><tr><th>数值</th><th>含义</th></tr><tr><td>0</td><td>同步非循环</td></tr><tr><td>1~240</td><td>同步循环</td></tr><tr><td>254, 255</td><td>异步非循环</td></tr></table>							数值	含义	0	同步非循环	1~240	同步循环	254, 255	异步非循环
数值	含义															
0	同步非循环															
1~240	同步循环															
254, 255	异步非循环															
	03h	生产禁止约	RW	NO	UINT16	VAR	UINT16	0								

		束时间（单位：1/10ms）																		
	05h	定时器触发时间（单位：ms）	RW	NO	UINT16	VAR	UINT16	0												
	06h	同步起始值	RW	NO	UINT8	VAR	UINT8	0												
1600h-1603h	-	RPDO 映射参数	RW	NO	-	REC	-	-												
		只可以在 PDO 无效的状态下才可以修改此对象。映射对象的总位长不得超过 64 位，只支持按字节映射，不支持按位映射。																		
	00h	PDO 有效映射对象个数	RW	NO	UINT8	VAR	0-8	-												
	1-8h	PDO 的各个映射对象	RW	NO	UINT32	VAR	UINT32	-												
		映射对象内容索引和子索引必须存在对象字典列表中，属性为可写状态，且为可映射。按以下格式写入对应子索引： <table><tr><td>31</td><td>16</td><td>15</td><td>8</td><td>7</td><td>0</td></tr><tr><td colspan="2">索引</td><td colspan="2">子索引</td><td colspan="2">对象长度</td></tr></table> MLBLSB							31	16	15	8	7	0	索引		子索引		对象长度	
31	16	15	8	7	0															
索引		子索引		对象长度																
1800h-1803h	-	TPDO 通信参数	RW	NO	-	REC	-	-												
	00h	最大子索引	RO	NO	UINT8	VAR	6	6												
	01h	TPDO 的 COB-ID	RW	NO	UINT32	VAR	UINT32	见下文												
		只可改变最高位和次高位。 最高位为“0”表明该 PDO 有效，最高位为“1”表明该 PDO 无效； 次高位表示是否支持远程帧触发该 PDO，由于驱动器不支持该功能，故此位没有意义，建议写“1”表示不允许远程帧触发该 PDO。 出厂设定如下： 1800h ： 0x40000180 + Node_ID 1801h ： 0xC0000280 + Node_ID 1802h ： 0xC0000380 + Node_ID																		

		1803h： 0xC0000480 + Node_ID																		
	02h	TPDO 的 传 输类型	RW	NO	UINT8	VAR	UINT8	255												
		只可以在 PDO 无效的状态下才可以修改此数值。不同的数值代表不同的 PDO 传输类型，如下表： <table><tr><th>数值</th><th>含义</th></tr><tr><td>0</td><td>同步非循环</td></tr><tr><td>1~240</td><td>同步循环</td></tr><tr><td>254、255</td><td>异步非循环</td></tr><tr><td>其它</td><td>保留</td></tr></table>							数值	含义	0	同步非循环	1~240	同步循环	254、255	异步非循环	其它	保留		
数值	含义																			
0	同步非循环																			
1~240	同步循环																			
254、255	异步非循环																			
其它	保留																			
	03h	禁止事件	RW	NO	UINT16	VAR	UINT16	0												
		只可以在 PDO 无效的状态下才可以修改此数值。 单位 100us，为“0”时无效禁止时间																		
	05h	事件计时器	RW	NO	UINT16	VAR	UINT16	0												
		只可以在 PDO 无效的状态下才可以修改此数值。 单位 1ms，为“0”时无效事件计时器。																		
1A00h- 1A03h	-	TPDO 映 射 参数	RW	NO	-	REC	-	-												
		只可以在 PDO 无效的状态下才可以修改此对象。映射对象的总位长不得超过 64 位，只支持按字节映射，不支持按位映射。																		
	00h	PDO 有效映 射对象个数	RW	NO	UINT8	VAR	0-8	0												
	1-8	TPDO 的 各 个映射对象	RW	NO	UINT32	VAR	UINT32	-												
		映射对象内容索引和子索引必须存在对象字典列表中，属性为可写状态，且为可映射。 按 以下格式写入对应子索引： <table><tr><td>31</td><td>16</td><td>15</td><td>8</td><td>7</td><td>0</td></tr><tr><td colspan="2">索引</td><td colspan="2">子索引</td><td colspan="2">对象长度</td></tr></table> MLB LSB							31	16	15	8	7	0	索引		子索引		对象长度	
31	16	15	8	7	0															
索引		子索引		对象长度																

5.3 对象 6000h 组分类说明

索引	子索引	名称	可访问性	能否映射	数据类型	单位	数据范围	出厂设定
603Fh	-	错误代码	RO	Y	UINT16	-	0-65536	0
6040h	-	控制字	RW	Y	UINT16	-	0-65536	0
6041h	-	状态字	RO	Y	UINT16	-	0-65536	0
605Ah	-	快速停机方式选择	RW	Y	INT16	-	1-2 5-6	1
6060h	-	模式选择	RW	Y	INT8	-	0-7	0
6061h	-	模式显示	RO	Y	INT8	-	0-7	0
6063h	-	电机位置反馈	RO	Y	INT32	编码器单位	INT32	0
6064h	-	用户位置反馈	RO	Y	INT32	用户单位	INT32	0
606Ch	-	用户实际速度反馈	RO	Y	INT32	rpm	0-3000	0
6071h	-	目标转矩	RW	Y	INT16	0.1%	-1000-1000	0
6077h	-	实际转矩	RO	Y	INT16	0.1%	-1000-1000	0
607Ah	-	目标位置	RW	Y	INT32	用户单位	INT32	0
607Ch	-	原点偏置	RW	Y	INT32	用户单位	INT32	0
607Eh	-	指令极性	RW	Y	UINT8	-	0-1	0
6081h	-	轮廓速度	RW	Y	UINT32	Inc/s	0-3000	100000
6083h	-	轮廓加速度	RW	Y	UINT32	Inc/s/ms	UINT32	100000
6084h	-	轮廓减速度	RW	Y	UINT32	Inc/s/ms	UINT32	100000
6085h	-	快速停机减	RW	Y	UINT32	Inc/s/	UINT32	100000

		速度				ms		
6087h	-	转矩斜坡	RW	Y	UINT32	0.1% /ms	UINT32	1
6091h	-	齿轮比	-	-	-	-	-	-
	01h	电机单位	RW	Y	UINT32	inc	UINT32	131072
	02h	用户单位	RW	Y	UINT32	inc	UINT32	100000
6098h	-	回零模式	RW	Y	INT8	-	0-99	17
6099h	-	回零速度	-	-	-	-	-	-
	01h	搜索减速点 速度	RW	Y	UINT32	Inc/s	0-3000	100000
	02h	搜索原点速 度	RW	Y	UINT32	Inc/s	0-3000	20000
609Ah	-	回零加速度	RW	Y	UINT32	Inc/s/ ms	UINT32	1000000
60C1h	-	插补数据记 录	-	-	-	-	-	-
	01h	插补位移	RW	Y	INT32	用 户 单位	INT32	0
60C2h	-	插补时间	-	-	-	-	-	-
	01h	插补时间单 位	RW	Y	UINT8		1-20	1
	02h	插补时间索 引	RW	Y	INT8		-	-3 表示 ms
60F4h	-	用户位置偏 差	RO	Y	INT32	用 户 单位	UINT32	0
60FDh	-	DI 状态	RW	Y	UINT32	-	UINT32	0
60FEh	-	数字输出	-	-	-	-	-	-
	1h	DO 状态	RW	Y	UINT32	-	UINT32	0
60FFh	-	目标速度	RW	Y	INT32	Inc/s	INT32	0
6502h	-	支持伺服运 行模式	RO	Y	UINT32		-	0x6D