伺服驱动器

EtherCAT 通讯

技 术 指 南

发行日期: 2020 年 03 月 03 日
当前版本: V1.0.1
修改日期: 2020 年 03 月 05 日

深圳市正元电机有限公司..
深圳市龙岗区布吉街道甘坑社区同福路 9 号永平工业园 3 号厂房 6-8 楼
电话：+ 86-755-83688818
电话：+ 86-755-89585358
传真：（86）755 89585354
电子邮件：info@assunmotor.com.cn

简介

该协议说明将对 EtherCAT 网络通信的构建方法、物理参数的调整方法、激活各种功能的方法的技术规格进行说明。
本说明书的使用对象需要具备一定程度的伺服驱动器、运动控制、网络及 EtherCAT CoE (CANopen over EtherCAT)相关知识。

您可通过下方的 ETG 官网(EtherCAT Technology Group),获取 EtherCAT 的详细资料。
http://www.ethercat.org/

■ Trademark

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff AutomationGmbH, Germany.

EtherCAT 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 EtherCAT 标准及应用层协议 DS402 协议。

注意：EtherCAT 通讯技术适用于有相关了解的技术人员开发。

一、EtherCAT 协议介绍

1.1、EtherCAT 概要

EtherCAT 是 Ethernet for Control Automation Technology 的略称,是由德国倍福自动化有限公司(BECKHOFF)研发的通过实时以太网进行主站与从站间的开放网络通信,由 EtherCAT 技术协会(EtherCAT Technology Group)进行管理。

EtherCAT 可使用双绞线或光纤电缆进行连接。通过 EtherCAT 可以构建多种拓扑结构,比如总线拓扑结构、树形拓扑结构、菊花链式拓扑结构、蜂窝拓扑结构等。从主站发出的数据帧在经过从站时将读取 Output 数据,并以同样的方式插入 Input 数据。同时,由于通信协议遵守基于 IEEE802.3 的标准以太网协议,因此无需构建新的子总线用于 EtherCAT 连接。

协议通过各个向以太网帧直接传送、多个子报文构成,各个报文及 4GB 过程映像,实现了广播和组播通信的功能。使用符合 100BASE-TX 标准的以太网时,缆线的最大长度可达到 100m,系统最多可容纳 65535 台子轴,因此整个网络规模几乎是无限制的。

1.2、EtherCAT 行规

■ IEC61158 Section12
· IEC61158-2-12 (EtherCAT Physical Layer Specification and service definition)
·IEC61158-3-12 (EtherCAT Data-link service definition)
·IEC61158-5-12 (EtherCAT layer service definition)
·IEC61158-6-12 (EtherCAT layer protocol specification)

IEC61158 中规定了基于 Ethernet 的网络的基本通信结构,其中,Section12 中对 EtherCAT 通信行规的 EtherCAT 状态机(ESM)、利用 FMMU 特性的 Process Data 通信方式、基于 EtherCAT 邮箱的 CoE 服务通道、使用同步管理(SM)及分布式时钟(DC:Distributed Clock)方式的同步结构进行了说明。

■ IEC61800 Part7 (Adjustable speed electrical power drive systems)
·IEC61800-7-1 (Generic interface and use of profiles for power drive systems - Interface definition)
·IEC61800-7-200 (Generic interface and use of profiles for power drive systems - Profile specifications)
·IEC61800-7-300 (Generic interface and use of profiles for power drive systems - Mapping of profiles to network technologies)

IEC61800 第 7 部分可调速动力传动系统行规中对伺服驱动器的功能型动作进行了规定，Section1 中规定了 PDS 的通用接口及行规使用方法。

Section200 对行规类型的规格进行了规定。Profile type1(-201)中对数据协议 CiA402 的对象字典、迁移状态 FSA 及动作模式功能进行了详细的说明。Profile type4(-204)中主要对 SERCOS 的 IDN 及相位进行了详细的说明。

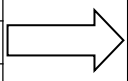
Section300 对网络技术中的映射进行了规定。Mapping of profile type1(-301)中对 CANopen、CANopen over EtherCAT 进行了的说明,Mapping of profile type4(-304)中对 SERCOS 及 Servo drive over EtherCAT 等通信协议进行了说明。

1.3、模型(Reference Model)

1.3.1 OSI 参考模型

OSI 参考模型(Open Systems Interconnection 七层结构)与 EtherCAT 通信模型的对比如下所示。在 EtherCAT 模型中没有第 3 层到第 6 层。

OSI 参考模型与 EtherCAT(CoE)模型的比较

<table>
<tr><td>层</td><td>OSI reference model</td><td></td><td>EtherCAT model</td></tr>
<tr><td>第 7 层</td><td>Application （应用层)</td><td></td><td>SDO (Service Data Object : Mailbox)
PDO (Process Data Object)
ESM (EtherCAT State Machine)
ESI (Slave Information Interface)</td></tr>
<tr><td>第 6 层</td><td>Presentation(表示层)</td><td rowspan="4">  </td><td rowspan="4">Empty</td></tr>
<tr><td>第 5 层</td><td>Session(会话层)</td></tr>
<tr><td>第 4 层</td><td>Transport(传 输层)</td></tr>
<tr><td>第 3 层</td><td>Network(网络层)</td></tr>
<tr><td>第 2 层</td><td>Data link (数据链路层)</td><td></td><td>SM (Sync Manager)
FMMU (Field Memory Management Unit)
PDI (Process Data Interface)
DC (Distributed Clock)</td></tr>
<tr><td>第 1 层</td><td>Physical (物理层)</td><td></td><td>100BASE-TX
E-BUS (LVDS for backplane)</td></tr>
</table>

■ 第 1 层(物理层)

进行电气交换及机械作业,以便将数据输出到通信电路。该层中规定了引脚

形状及电缆特性。

■ 第 2 层(数据链路层)

确保通信双方的物理通信路径,并检测通过路径的数据错误等。

■ 第 3 层(网络层)

进行通信路径的选择、通信路径内的地址管理等,以便将数据从发送方路由到接收方。

■ 第 4 层(传输层)

为保证发送方发送的数据可以正确且快速的发送到接收方,进行数据压缩,错误修正,重新发送数据等操作。

■ 第 5 层(会话层)

为进行程序间的数据收送进行建立/断开虚拟连接操作。

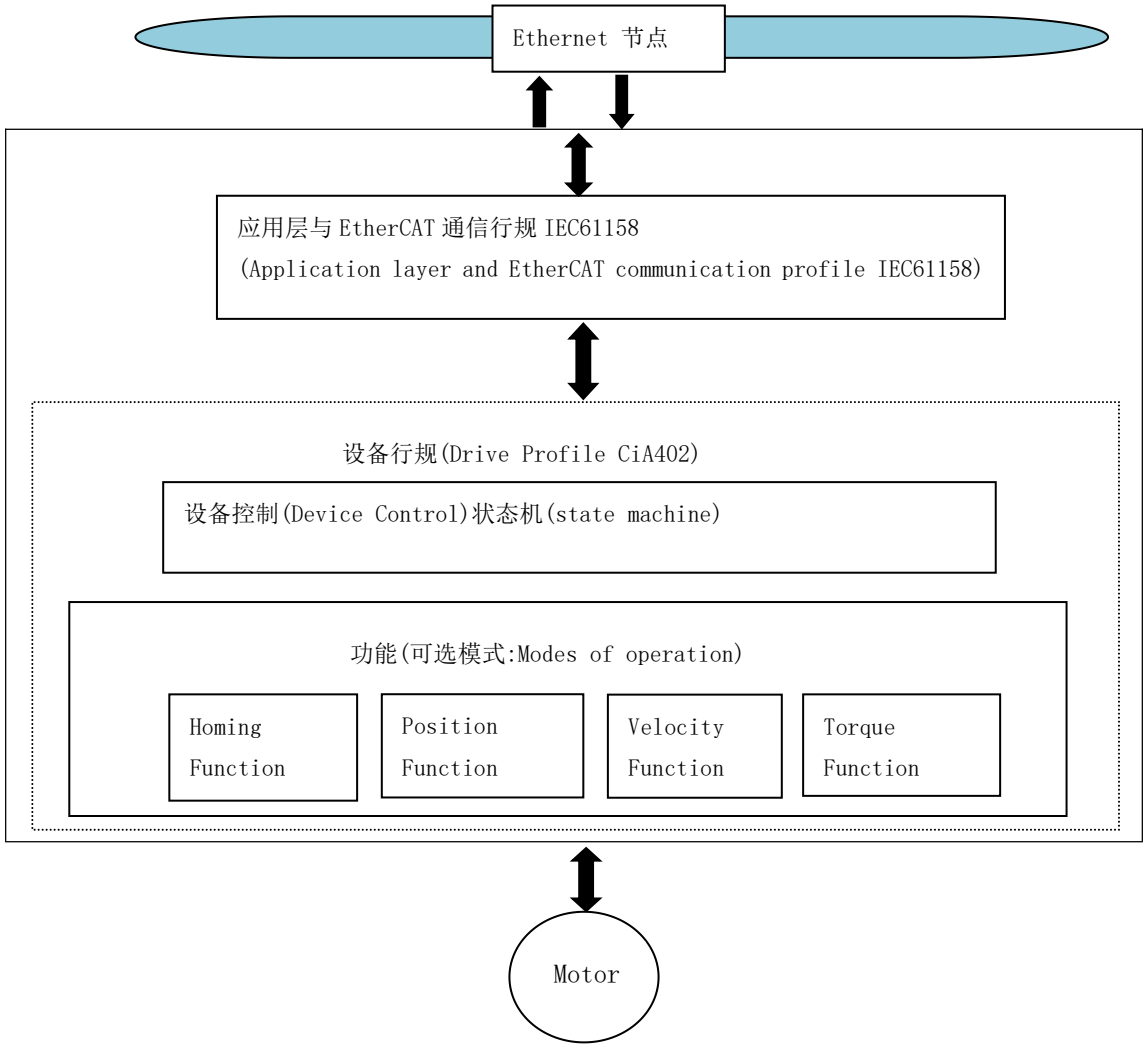
■ 第 6 层(表示层)

将从会话层中接收到的数据转化为适合用户使用的形式,将从应用层中接收到的数据转化为适用于通信的形式。

■ 第 7 层(应用层)

向用户及其他程序提供使用数据通信的各种服务。

1.4、设备构造



通信体系结构(Communication architecture)

1.5、设置

1.5.1 节点 ID

以太网中的各从站设备都有各自固定的节点 ID,一般来说,固定的节点 ID 可在位置寻址模式中进行设置。除节点 ID 以外,还可通过驱动器的调试上位机的参数表地址设置，范围为 1～65535。设置值将在控制电源上电后写入地址空间的站别名设置寄存器(0x0012)中。在控制电源 ON 的状态下修改轴的
地址时,要使新修改的地址生效需重启电源。

1.5.2 物理通信规格

物理通信规格		
名称	规格	备注
拓扑结构	总线	
数据流向	路径:由主站向第一台从站,再从最后一台从站折回	
通信媒介	双绞线	
通信速度	100 Mbit/s	
通信参数设置	基于 ISO/IEC 8802-3 标准的自动协商功能,自动跨接功能	
周期时间	由具体应用决定	
设备地址	所选择的地址	
同步	数据交换用(DC)的特殊协议	
从站报文	EtherCAT CoE 标准下 Mailbox 中的 SDO 报文	
主站报文	EtherCAT CoE 标准下 Mailbox 中的 SDO 报文	
初始化	上电后的初始化过程为 Init→Pre-Operational,→Safe-Operational →Operational 模式。	
电缆长度	Max 100m	节点间
节点	Max65,535	单个网段

1.6、通信规格

1.6.1 设备模型

■ Communication

此功能单元包括通过基础网络结构进行数据传输的各项功能。

■ Object Dictionary

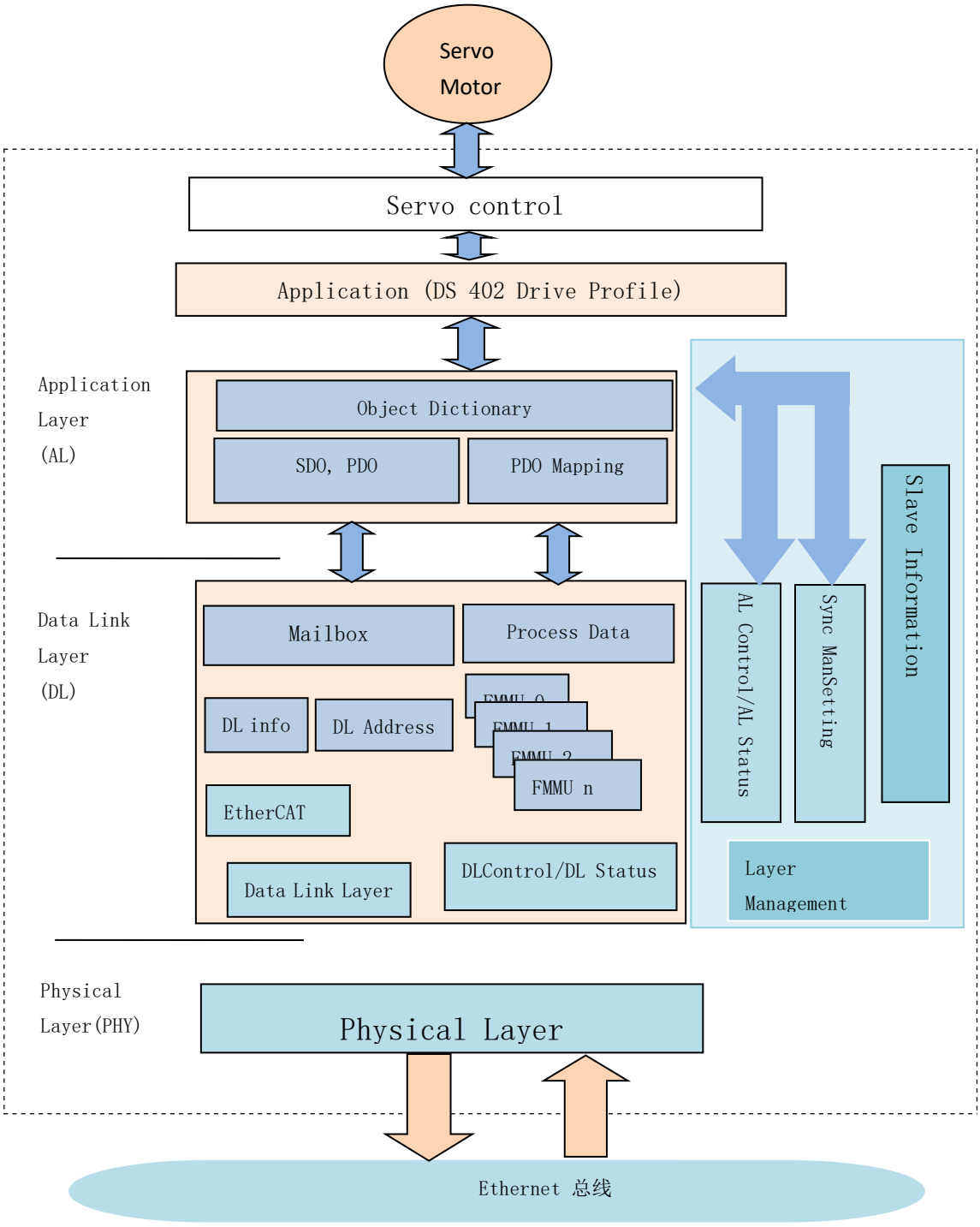
使用对象字典将对应用对象、通信对象及使用该设备的状态机产生影响。

■ Application

应用包括与动作环境相适应的进行数据交换的通信设备功能。

对象字典具有作为通信与应用间的接口的功能。

对设备应用的说明称为“设备行规”，设备行规中把对象字典的各个数据作为说明的条目。



对象字典与设备模型

■ 对象索引(Object Index)

所有的对象都使用 4 位 16 进制的 16bit 索引进行定位。
对象被按组配置在对象字典中。
CoE 中规定的对象字典概要如下所示。

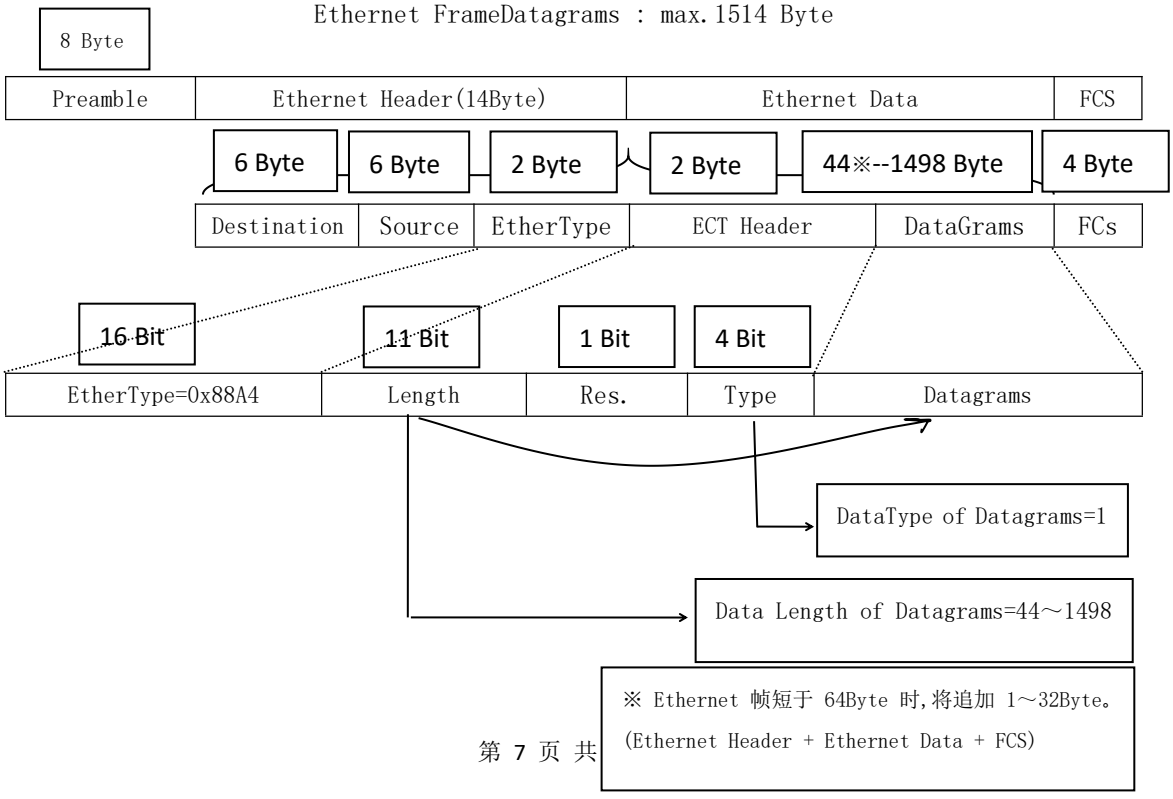
对象索引配置

Index(Hex)	对象
0x1000～0x1FFF	Communication Profile Area
0x2000～0x5FFF	Manufacturer Specific Profile Area
0x6000～0x9FFF	Standardized Device Profile Area
0xA000～0xFFFF	Reserved

1.6.2 通信

■ Ethernet 协议

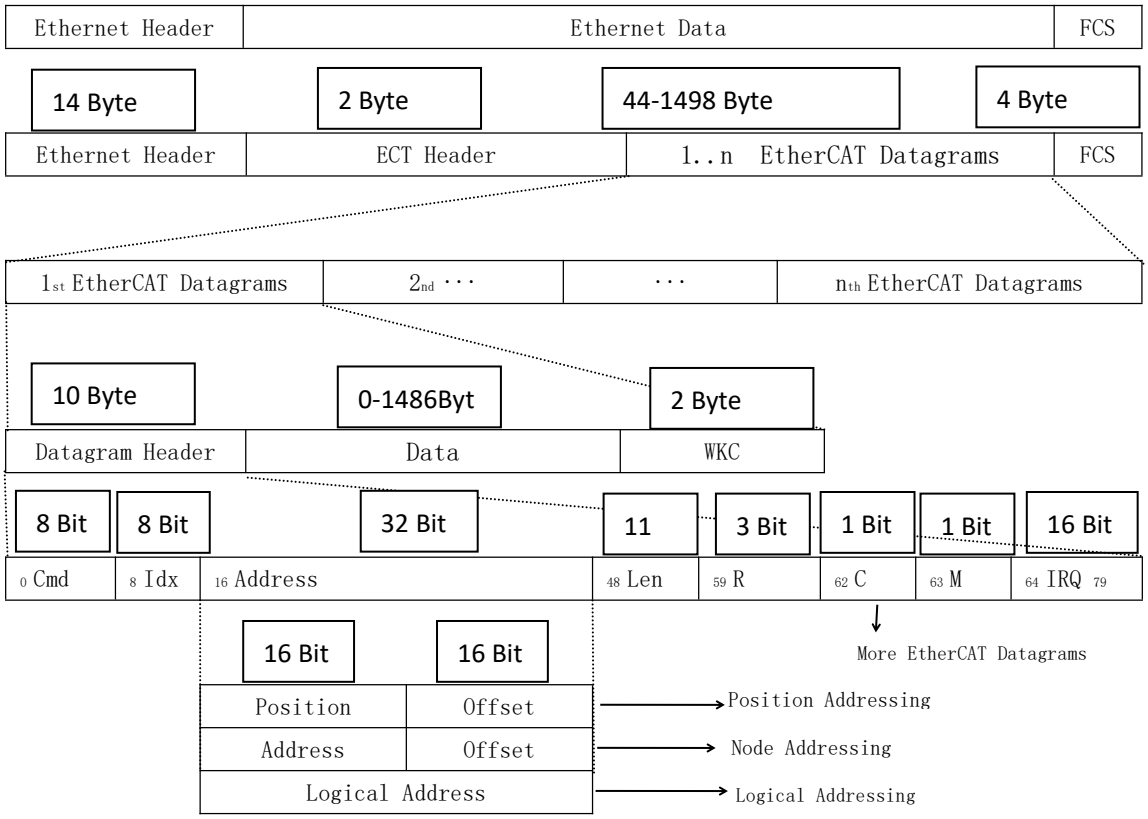
由于 EtherCAT 采用了 IEEE 802.3 标准的以太网帧,因此可以使用标准的网络控制器,所以主站侧无需另外搭
设硬件即可构建系统。
EtherCAT 的预设 Ether Type 为“ 0x88A4” ,以区别于其他以太网帧。
此外,EtherCAT 不需要使用 IP 协议。
数据帧对 EtherCAT 数据报进行定义,并根据 EtherCAT 帧头进行更详细的分类。
只有 EtherCAT 帧头为 Type“ 1” 的 EtherCAT 帧将被从站设备处理。



以太类型与以太网数据的报头

1.6.3 EtherCAT 协议

为了更便捷地构建网络,IEC61158EtherCAT 通信行规中对作为默认值的指令进行了标准化的规定。网络中的各个节点都被分配给一个地址,因此每个节点都可以使用同个以太网中的 EtherCAT 数据报。帧最终以 EtherCAT 数据报结束。



EtherCAT 数据报

1.6.4 数据报头

数据报前端的 10 Byte 数据报头决定了后续数据将被如何处理。

EtherCAT 数据报头

域	数据类型	值/说明
Cmd	BYTE	EtherCAT 指令类型
idx	BYTE	Idx 是用于复制/识别数据报的数据标识符,idx 在主站中进行处理,无法在从站中进行变更。

Address	BYTE[4]	指通过 32 bit 地址访问从站的访问方式。 ·自动递增地址 (16bit 设备地址+16bit 偏移地址) ·节点地址 (16bit 设备地址+16bit 偏移地址) ·逻辑地址 (32bit 逻辑地址)
Len	11bit	此数据报后的数据长度
R	3bit	Reserved,0
C	1bit	帧循环 0 : 无循环 1 : 已循环过
M	1bit	相邻的 EtherCAT 数据报 0 : 最后一个 EtherCAT 数据报 (n_{th} EtherCAT Datagrams) 1 : 后接 EtherCAT 数据报 (例:1_{st} EtherCAT Datagrams 后连接的数据报是 2_{st} EtherCAT Datagrams)
IRQ	WORD	执行逻辑 OR 后所有相关从站的 EtherCAT 中断请求寄存器
Data	BYTE[n]	读/写数据
WKC	WORD	工作计数器

1.6.5 指令类型

EtherCAT 数据报前端的 8 bit 指令类型决定了后续“ Address” 及访问方式。EtherCAT 指令类型列表如下所示。读写操作、读取操作将在写入操作前执行。

EtherCAT 指令类型

CMD	缩略语	名 称	说 明
0 (0x00)	NOP	No Operation	空指令
1 (0x01)	APRD	Auto Increment Read	递增地址, 收信地址为零时,向数据报设置读取数据
2 (0x02)	APWR	Auto Increment Write	递增地址, 收信地址为零时,将数据写入内存区。
3 (0x03)	APRW	Auto Increment ReadWrite	递增地址, 收信地址为零时,向数据报设置读取数据,并在同一内存区中写入数据。
4 (0x04)	FPRD	Configured Address Read	地址一致时,向数据报设置读取数据。
5 (0x05)	FPWR	Configured Address Write	地址一致时,向内存区写入数据。
6 (0x06)	FPRW	Configured Address Read Writ	址一致时, 向 EtherCAT 数据报设置读取数据,在同一内存区中写入数据。
7	BRD	Broadcast Read	设置所有从站的内存区数据及数据报数据的逻辑 OR。

(0x07)			
8 (0x08)	BWR	Broadcast Write	所有从站向内存区写入数据。
9 (0x09)	BRW	Broadcast Read Write	设置所有从站的内存区数据及数据报数据的逻辑 OR,并将数据写入内存区 （一般不使用 BWR。）
10 (0x0A)	LRD	Logical Memory Read	收信地址与读取设置 FMMU 一致时， 向数据报设置写入数据。
11 (0x0B)	LWR	Logical Memory Write	收信地址与写入设置 FMMU 一致时， 向内存区写入数据。
12 (0x0C)	LRW	Logical Memory Read Write	收信地址与界定设置 FMMU 一致时,向数据报设置读取数据 收信地址与写入设置 FMMU 一致时,向内存区写入数据。
13 (0x0D)	ARMW	Auto Increment Read Multiple Write	递增地址,收信地址为零时面向数据报中插入读取数据其他从站向内存区写入数据
14 (0x0E)	FRMW	Configured Read Multiple Write	地址一致时,向数据报设置读取数据 其他的从站向内存区写入数据。
15～255(0x0F～0xFF)			Reserved

关于 EtherCAT 数据报 32 位地址的 EtherCAT 寻址模式请参考下表

EtherCAT 寻址模式			
模式	域	数据类型	值/说明
Auto Increment Address	Position	WORD	各从站位置递增,Position = 0 的从站将被寻址。
	Offset	WORD	ESC 的本地寄存器或内存地址。
Configured Station Address	Address	WORD	设置的轴地址与设置的站地址(无效时)一致时,从站将被寻址。
	Offset	WORD	ESC 的本地寄存器或内存地址。
Logical Address	Address	DWORD	逻辑地址(通过 FMMU 设置)FMMU 配置与地址一致时,从站将被寻址。

1.6.6 WKC(工作计数器)

每个 EtherCAT 数据报都以 16bit 的工作计数器(WKC)结束。工作计数器计算被 EtherCAT 数据报正常访问的设备台数。
同时,从站驱动器中搭载的 ESC(硬件)也将递增工作计数器的数值。
主站通过对比向各从站发出的指令与 WKC 的目标值和实际计算结果,对 EtherCAT 数据报是否执行了处理进行确认。
各指令与工作计数器的递增过程如下表所示。

工作计数器的增加数		
指令	数据类型	增加数

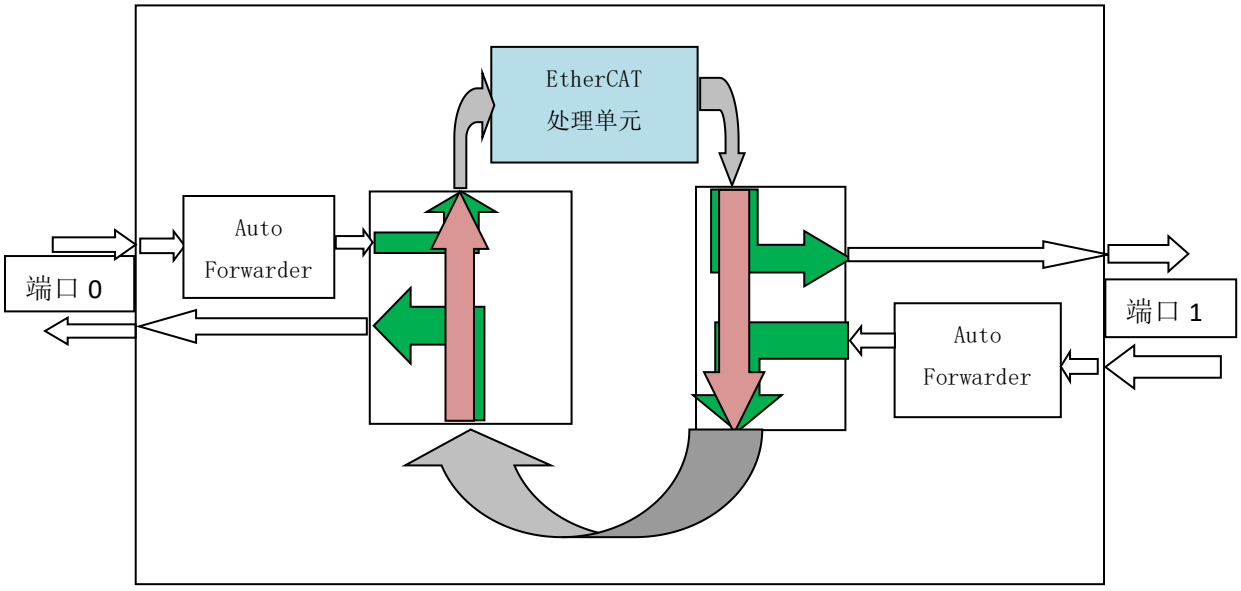
读取指令	读取失败	不变
	读取成功	+1
写入指令	写入失败	不变
	写入成功	+1
读/写指令	失败	不变
	读取成功	+1
	写入成功	+2
	读/写成功	+3

1.6.7 帧处理

EtherCAT 驱动器有 2 个端口,帧处理顺序(Processing)由逻辑端口编号决定。

帧处理顺序			
使用端口	帧处理顺序		
1 Port	端口 0 → 处理 → 端口 0	------	
	端口 1 → 处理 → 端口 1	------	
2 Ports	端口 0 → 处理 → 端口 1	⇒	端口 1 → → → 端口 0
	端口 1 → → → 端口 0	⇒	端口 0 → 处理 → 端口 1

经过 EtherCAT 处理单元的方向称为“处理(Processing)”，不经过处理单元时称为“转发(Forwarding)”。

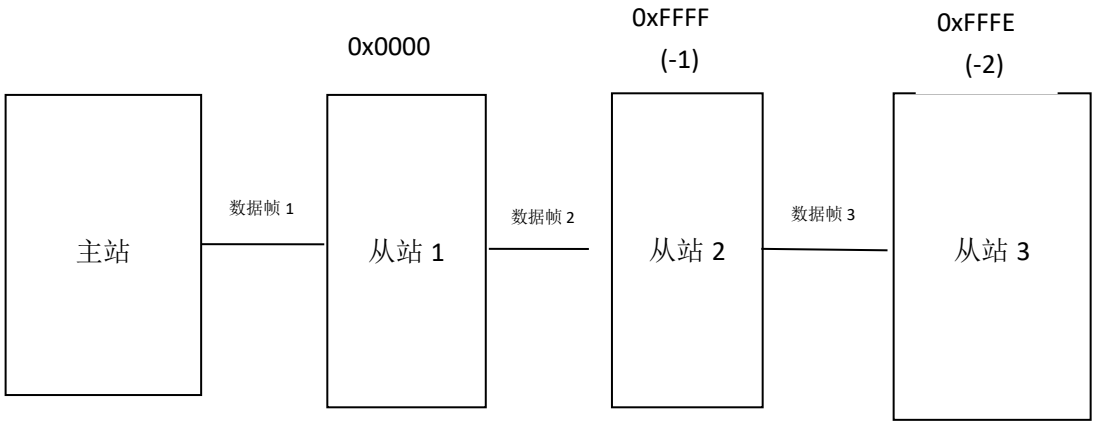


EtherCAT 驱动器的帧处理顺序

1.6.8 寻址示意图

1）、位置寻址(自动递增寻址)

在网段内顺序寻址时，从站根据各个从站的连接位置来确定从站地址。从站地址用负数来表示位置。当数据组从主站发出，经过从站时，每个从站的顺序地址增 1；从站在接收报文时，当顺序地址增加到 0 时，那么这个报文就是该从站的报文。由于这种机制在报文经过时更新设备地址，所以又被称为“自动增量寻址”。



顺序设备寻址时的从站地址

2）、节点寻址(固定位置寻址)

节点寻址操作中,与在一般位置寻址中由主站在站地址寄存器(0x0010)中设置的地址一致的从站将被寻址。此功能保障了在添加设备、网段拓扑结构发生变更或是卸载从站时仍然能够正常访问各站。

从站固有的节点地址为使用驱动器参数表地址 0x1145 的值为站别名(0x0012))中设置的地址,因此连接顺序发生改变也可以正确识别。此外,还可以通过 DL 控制器的设置对该种形式的地址进行访问。

（3）、逻辑寻址和 FMMU

逻辑寻址时,将网段内的 32bit 地址字段作为 1 个地址值使用。

逻辑寻址不是对各站进行寻址,而是对网段中 4GB 字段宽度的逻辑地址空间进行寻址。

该网段可以使用任何一个从站编号,通过现场总线内存管理单元 (FMMU)的内部地址映射方式可以将 32bit 的逻辑地址转换为物理地址。

各 FMMU 通道对与从站中的一个连续的物理空间相邻的逻辑空间进行映射。

进行逻辑寻址时,FMMU(Fieldbus Memory Management Unit、现场总线内存管理单元)将转变 ESC 的物理地址与 4GByte(32bit 地址)的主站逻辑地址。各 FMMU 通道通过将主站中连续的逻辑地址空间分配到从站的连续物理地址空间中,可以对主站管理的逻辑地址与跨各个从站的物理地址进行统一的管理。此外,FMMU 中支持“Read”,“Write”,“Read/Write”三种访问方式。

（4）、SM(SyncManager)同步管理器

ESC 内存可以无限制的用于主站与从站微控制器间的数据交换。但由于 SM 是使用通信内存对 ESC 内部进行寻址,因此
会有以下几个缺点:

■ 无法保证数据的整合性。
为保证数据交换的协调性,需要使用相关软件进行信号处理。

■ 无法保障数据安全。
构建数据安全机制需要通过使用软件来实现。

■ EtherCAT 主站或从站确认访问终止通知前,主站与从站都必须一直轮询内存。

允许 SM 操作与正常的数据通信在 EtherCAT 主站与从站间转换,同时向两方插入变更通知。SM 在主站中进行设置,数据交换时将使用内存区中设置的缓存区。

通信方向与缓存模式及邮箱模式相同。

对缓存区的访问通过 SM 的硬件进行控制,访问时需要首先对起始地址进行访问,否则访问将被拒绝。

访问起始地址后,可以对整个缓存区进行访问。访问终止地址后访问结束且缓存状态发生改变。同一帧只能访问一次终止地址。

SM 支持以下两种通信模式。

■ 缓存模式
使用缓存模式时,随时都可对 EtherCAT 主站与从站两方的通信缓存区进行访问。

收信方可以不断读取送信方中写入的最新缓存数据,发信方可以不断更新缓存值。但是在写入速度快于读取
速度时,旧的数据将被覆盖。
一般在 TxPDO·RxPDO 的 PDO 通信中使用缓存模式。

■ 邮箱模式
邮箱模式中使用握手机制而非数据交换机制,因此不会丢失数据。但 EtherCAT 主站和从站只有在另一方访问终止后才能访问缓存区。

首先发信方将数据写入缓存区,然后缓存区被锁定为只读,直至收信方读走缓存区的数据。

邮箱模式通常用于应用层非周期性数据的交换

1.6.9 缓存模式(三区缓存模式)

主站与从站能够同时进行数据的读取与写入的缓存模式叫做三区缓存模式。在三区缓存模式下,内存中将分配出 3 个相同大小的缓存区。开始地址及第一个缓存区大小将设置在 0x0800～的 SM0～7 配置寄存器中。
此缓存地址是为了在主站及从站中读取/写入数据而设置的。访问第一个缓存区(0)地址宽度时,由于 SM 会自动切换 3 个缓存区中 1 个 SM 自动切换访问。因此主站与从站只需要访问对方缓存区(0)的地址即可。

此外,由于缓存区(1)与(2)使用的内存将被自动保留,因此该部分内存无法使用,在设置其他 SM 时,请务必考虑到该部分内存的使用情况。

1.6.10 邮箱模式

邮箱模式通过握手机制的方式进行写入数据与读取数据的交换,保证收信方能够完整地接受到发信方发出的所有数据。

邮箱模式使用被设置的 1 个单位大小的缓存区,完成 SM 的初始设置及启动后,即可作为邮箱缓存区使用。

对缓存区写入数据完成后,写入访问将被锁定,此时收信方可以开始读取数据。数据正常读取完成后将解除锁定,缓存区重新被允许写入数据。
此模式下读取/写入数据所花费的时间不重要。

1.7 访问对象字典

EtherCAT 驱动器支持 CoE(CANopen over EtherCAT),并可通过以下两种方式中的任意一种访问设备对象
字典。

- 服务数据对象 (SDO)
- 过程数据对象 (PDO)

1）、服务数据对象(SDO)

主站通过使用 SDO 传输对象字典的项目进行读取/写入操作,可以控制设备设置、监视等从站驱动器的各项参数。
支持 EtherCAT CoE 的主站将对各从站设备进行 SDO 传输。
通过 SDO 主站发送的 T_SDO 进行数据变更、读取 R_SDO 的请求等操作。

2）、邮箱协议

邮箱是在主站到从站、从站到主站方向中使用的与逻辑寻址不同的独立的通

信,邮箱支持多种 DL 用户协议。
从站到从站的数据传输必须通过主站处理。
邮箱数据头(MailBox Header)中含有可以用于实现主站的重定向服务的地址字段。

邮箱使用两 SyncManager 通道(SM),一个用于主站到从站方向,另一个用于从站到主站方向。
(例:SM0 为主站 → 从站方向,SM1 为从站 → 主站方向)

由于邮箱通信是对非周期性的单一从站进行寻址,因此不使用逻辑寻址方式而使用不需要 FMMU 的物理寻址方式。

（3）、CANopen 数据头协议

“ CANopen 数据头” 由 2Byte 的识别码构成,包括“ 序号(Num)” 与“ 类型(Type)” 两部分。

“ CANopen 数据头” 的构成如下所示。

CoE 指令结构		
标记(略称)	数据长度	说 明
Number(Num)	9 bit	PDO 序号 (只在 PDO 传输时使用) 0x000～0x1FF
Type(Type)	4 bit	信息类型 0 : 保留 1 : 紧急事件信息 2 : SDO 请求 3 : SDO 响应 4 : 保留(TxPDO) 5 : 保留(RxPDO) 6 : 保留(远程 TxPDO 发送请求) 7 : 保留(远程 RxPDO 发送请求) 8 : SDO 信息 9～15 : 保留

（4）、SDO 消息

SDO 消息由“ CANopen 数据头” 及“ SDO 数据帧” 两部分构成。一般可以传输最大 4Byte 的数据,此外,通过使用“ 可选扩展数据” 区,可以将传输的数据扩展到 1470Byte。
由于 EtherCAT CoE 驱动器的传输数据大部分都小于 4Byte,因此可以进行快速的 SDO 数据传输。

（5）、过程数据对象(PDO)

■ 概要

EtherCAT 的实时数据传输需要使用“ 过程数据对象(PDO)” 进行。
PDO 传输不需要协议传输处理的系统开销(overhead)。
使用的 PDO 包括用于主站向从站传输的 RxPDO(收信 PDO)及从站向主站传输的

TxPDO(发信 PDO)两种。
EtherCAT CoE 驱动器的 PDO 映射,通过在设置设备阶段使用 SDO 服务,可以对项目字典的该项目分
配需要的 PDO 数及 PDO 对象。
■ PDO 设置
通过设置 PDO,用户可以优化“ PDO 映射的信息区”、“ 收信/发信数据的形态(传输数据类型)及触发条件”。
■ PDO 映射
EtherCAT 驱动器的 PDO 映射无法变更。
因此 EtherCAT CoE 网络管理器可以自由地变更在操作中 PDO 传输的数据的映射。
RXPDO 的同步参数由 0x1c12 配置，TXPDO 的同步参数由 0x1c13 配置.
使用 “ 收信 PDO 映射参数(0x1600～0x1603)”变更 RxPDO 映射,使用 “ 发信 PDO 映射参数(0x1A00～0x1A03)” 变更 TxPDO 映射参数。
映射时,每个将被传输的 PDO 都需要设置索引、子索引及数据长度。
数据长度必须与对象字典中的数据长度一致。
PDO 映射可以通关上位机调试软件更改，也可以通关主站控制器更改。
举例说明对于 RxPDO 的更改：
用主站控制器更改步骤：

Sync Manager Assign Object

Index	Sub-index	Object contents	Note
1C12	00	4	个数
1C12	01	0x1600	
1C12	02	0x1601	
1C12	03	0x1602	
1C12	04	0x1603	

Index	Sub-index	Object contents	Note
1600	00	6	个数
1600	01	0x60400010	
1600	02	0x60400010	

1600	03	0x60400010	
1600	04	0x60400010	
1600	05	0x60400010	
1600	06	0x60400010	

1）、对 0x1c12 的子索引的值清零
2）、假如需要改映射 0x1600 所对应的映射参数内容，则需要先把 0x1600 的子索引的值清零。
3）、然后就可以改下值了，仔细检查后 再把 0x1600 和 0x1c12 的子索引的值 改成正确的数值。
4）、相同方式 改写其他的都是一样的方法

1.8、分布式时钟(DC)

在 EtherCAT 中,通过使用从站控制器中的分布式时钟(DC)单元进行从站间及从站与主站间的同步。EtherCAT 驱动器支持的 DC 功能如下所示。
■ 从站间或从站与主站间的时钟同步。
■ 为输入事件产生精确的时间标记。
■ 通过 DC 设置产生中断,从而进行准确的同步处理。
■ 同步采样数字输入。

1）、时钟同步
DC 时钟同步操作的基础是全部主站与从站的所有 EtherCAT 设备共有一个相同的 EtherCAT 系统时间。由于 EtherCAT 设备间互相同步,因此本地应用也相应地能够同步。在系统同步中,所有的从站将与标准时钟同步。
一般情况下,主站后连接的 1 个网段内的第一个从站的时间为“ 系统时间(System Time)” ,这个系统时间将作为同步
其他从站 DC 本地时钟“System Time”与主站时间的标准时钟使用。
通过计算可得出传播延迟、各从站的时钟漂移量及时钟偏量。
从站搭载的 ESC 可以对 EtherCAT System Time 产生同步信号

2）、系统时间(System Time)
EtherCAT 驱动器的系统时间(0x0910～0x0918)长 8 Byte,单位是 1ns/Lsb,最大可以计时 500 年。数据“ 0x0”表示 2000 年 1 月 1 日的 0:00Hour 0sec 000ms 000ns 000ns。以下将对同步中使用的专业术语进行说明。
■ 标准时钟(Reference Clock)定义

将一台 EtherCAT 设备的时钟作为标准时钟使用。

一般情况下,标准时钟是进行同步的主站与所有从站间具有 DC 功能的第一个从站。

标准时钟提供系统时间。

■ 本地时钟(Local Clock)定义

最初,各从站使用独立于标准时钟的本地时钟。

本地时钟与标准时钟间的时间差(偏移)与时钟漂移量相同,都可以通过补偿进行调整。通过测定本地时钟的速度与追加调节时钟偏移量可以补偿本地时钟。此外,各 DC 从站都保存有一个通过计算本地时钟与本地时钟偏移量得出的标准时钟的副本。

■ 传播延迟(Propagation Delay)

标准时钟与从站时钟间的传播延迟必须在系统时间传输到从站时获得。

■ 偏移(Offset)

本地时钟与标准时钟间存在偏移有两个原因。

一是从标准时钟的从站到本地时钟设备传输过程中的传播延迟;

二是由于上电时间差导致的本地时间初始值的时间差。偏移将通过对各从站进行补偿消除。

标准时钟的从站通过追加本地偏移,从本地时间中导出系统时间。这个偏移量显示了本地时间(上电后开始计算)与主站时间的时间差。

■ 漂移(Drift)

由于标准时钟与 DC 从站时钟不总是能一直使用相同的时钟源,因此它们的时钟源将受时钟间的偏差的影响,

导致时钟源总会比其他的时钟快一些,因此每个本地时钟将存在各自不同的漂移量。

3)、时钟同步过程

时钟同步过程分为以下三步。

1) 测量传播延迟

主站开始测量从站间个方向上的传播延迟。

各从站计算测试帧的收信时间。

之后主站读取时间戳,并计算所有从站间的传播延迟。

2) 补偿标准时钟(系统时间)偏移

将个从站时钟的本地时间与系统时间进行比较。

通过在各个从站中写入数值补偿每个从站的时间差。

所有的从站得到同一个时间值。

3) 标准时钟的漂移补偿

标准时钟与本地时钟间的漂移补偿必须通过计算时间差与调整本地时钟定期进行补偿。

4)、时钟同步的初始化步骤(例)

时钟同步(包括传播延迟测量的偏移补偿、漂移补偿)的初始化步骤如下所述。初始化后,所有的 DC 从站与标准时钟同步。

1) 主站通过读取从站的 DL 状态寄存器,把握网络构成。

2) 为读取端口 0 寄存器的收信时间,主站将通过广播写入发送最小 1 字节的数据。所有从站将通过各端口及 ECAT 处理单元匹配本地时间。

3) 主站将等待,直到广播写入发送的数据帧返回。

4) 主站根据网络构成读取所有从站的收信时间端口 0/1 及 ECAT 处理单元收信时间寄存器(0x0918:0x091F)。

5) 主站计算各传播延迟时间,将数值写入从站的系统时间延迟寄存器中。

6) 主站将在标准时钟(第 1 轴从站)的系统时间偏移寄存器中进行设置,以使标准时钟与主站时间相同。标准时钟的偏移值为主站时间减去标准时钟的 ECAT 处理单元收信时间(本地时间)所得的数值。

7) 主站计算所有 DC 从站的系统时间偏移值,并写入系统时间偏移寄存器。各从站的偏移值(第 2 轴及以后各从站)。为标准时钟的 ECAT 处理单元收信时间减去各 DC 从站 ECAT 处理单元收信时间所得的数值。

8) 静态漂移补偿时,主站将在一开始将“ ARMW” 或“ FRMW” 指令分几次发送各 DC 从站。(例:15,000 帧)

9) 动态漂移补偿时,主站将周期性的将“ ARMW” 或“ FRMW”。

5)、SYNC0/1 信号输出初始化步骤(例)

同步信号输出按以下步骤进行初始化。

1) 启用 PDI 控制寄存器(0x0140.10=1)DC SYNC Out Unit 的位。

2) 为使输出驱动程序设置适合从站内部的电路结构,在 SYNC/Latch PDI Configuration 寄存器中设置 SYNC0/1 输出。

※在初始化 EEPROM 数值时设置 0x0151。

3) 在 Pulse Length 寄存器中设置 SYNC 信号的脉冲宽度。(请务必设置 SYNC0 Cycle Time>0)

※初始化时,EEPROM 将对 0x0982:0x0983 进行设置。

4) 对 ECAT 或 0x0980 的设备描述部分的 PDI 分配同步单元。

5) 设置 SYNC0 信号周期(0x09A0:0x09A3)及 SYNC1 信号周期(0x09A4:0x09A7)。

6) 设置开始周期时间操作(0x0990:0x0997)为比容许输出周期更往后的时间。

(例:设置为读取系统时间后,加上开始时间及容许写入时间的数值)

7) 允许有效周期操作位(0x0981.0=1)作为 SYNC0 及 SYNC1 有效位(0x0981[2:1]=0x3)的同步信号。

同步单元将等待直到到输出第一个 SYNC0 脉冲的时间。

1.9、通信时序

EtherCAT 同步处理通过主站及从站内部的 EtherCAT 设备独立进行。同步模式的标准通信方式有以下三种。

1) 自由运行模式
从站应用与 EtherCAT 同步信号不同步。(非同步模式)
2) SM 事件同步模式
传输周期输出时从站应用与 SM2 事件同期,只传输周期输入时,从站应用与 SM3 ※事件同期。
※本驱动器不支持与 SM3 事件同步。
3) SYNC 事件同步模式
从站应用与 SYNC0 或 SYNC1 事件同步。

1.10、EtherCAT 状态机(ESM)

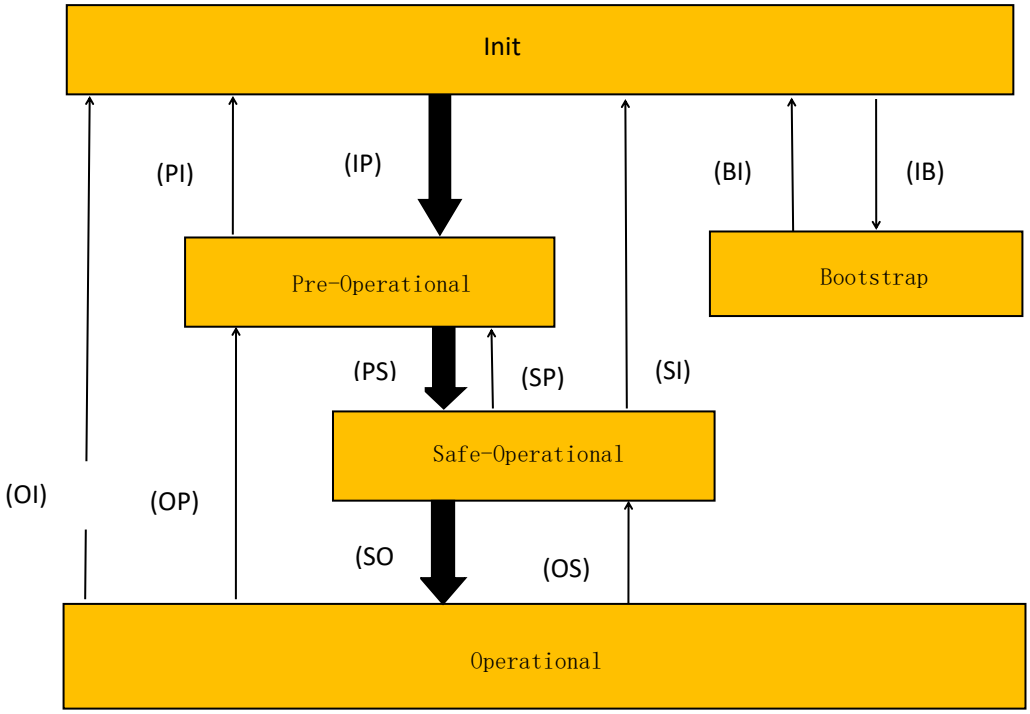
ESM 中包含 EtherCAT 中规定的各种状态

- Init (初始化)
- Pre-Operational (预运行)
- Safe-Operational (安全运行)
- Operational (运行)
- Bootstrap (引导状态)

1)、ESM

ESM 状态的改变由主站发起。

主站通过向从站的 AL 控制寄存器中写入希望变更的新 ESM,要求变更状态。从站响应此指令,执行所请求的指令转换,并将状态转换是否成功的结果写入本地的 AL 状态中。若请求的状态转换失败,从站将给出错误标志(Error Flag)。



ESM 状态转化关系

状态迁移及本地管理服务		
迁移 标记	Direction ⇒	本地管理 服务
IP	INIT TO PREOP	开始邮箱通信
PI	PREOP TO INIT	停止邮箱通信
PS	PREOP TO SAFEOP	开始输入更新
SP	SAFEOP TO PREOP	停止输入更新
SO	SAFEOP TO OP	开始输出更新
OS	OP TO SAFEOP	停止输出更新
OP	OP TO PREOP	停止输出更新,停止输入更新
SI	SAFEOP TO INIT	停止输入更新,停止邮箱通信
OI	OP TO INIT	停止输出更新,停止输入更新。停止邮箱通信
IB	INIT TO BOOT	开始固件升级(FoE) 启动引导状态模式
BI	BOOT TO INIT	停止固件升级(FoE) 重启设备

2）、状态

■ Init 状态

初始化状态定义了主站与从站在应用层的初始通信关系。
在应用层中,主站与从站间不可以直接通信。
主站使用“Init”状态进行从站配置寄存器的初始化设置。
从站支持邮箱通信服务时,将按“Init”状态执行对应的 SM 设置。

■ Pre-Operational 状态

在预运行状态下,若从站支持可选邮箱,则可进行邮箱通信。
主站与从站可以使用邮箱来进行应用配置的初始化及参数的变更。此状态下不允许过程数据通信。

■ Safe-Operational 状态

在安全操作状态下,从站应用读入输入数据,但不产生输出信号。输出处在“安全状态” 。

■ Operational 状态

运行状态中,从站应用读入输入数据,主站应用发出输出数据。

■ Bootstrap 状态

二、过程数据对象(PDO)映射驱动器内部参数

EtherCAT PDO 映射需要控制器上电后修改。

三、EtherCAT 伺服驱动器启动流程介绍

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

首先，EtherCAT 伺服驱动器上电后，会自动进入 FREERUN 状态。

只有当从站进入 Safe-Op 或者 Op 状态后,才能向主站控制器发送过程数据；

只有当从站进入 Op 状态后,才能接收来自主站控制器的过程数据；

只有当从站进入 Safe-Op 或者 Op 状态后，主站可以选择 SM-Synchron 或者 DC-Synchron 同步模式，主站控制器才能控制 cia402 状态机控制驱动器电机，才进入过程数据操作状态，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 TWINCAT3 成功控制，可以为用户提供一份 xml 文件说明，在此不必详细介绍。

四、EtherCAT_Cia402 控制伺服驱动器介绍

4.1、启动方式(RunControl)

作为标准 CANOPEN 控制方式，首先需要对“启动方式”参数写 3。

索引(H)	驱动器内部参数地址(H)	参数名	属性	数据类型	备注
6116	1000	启动方式(RunControl)	读写	Uint16	3:标准 CANOPEN 方式

4.2、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述语：

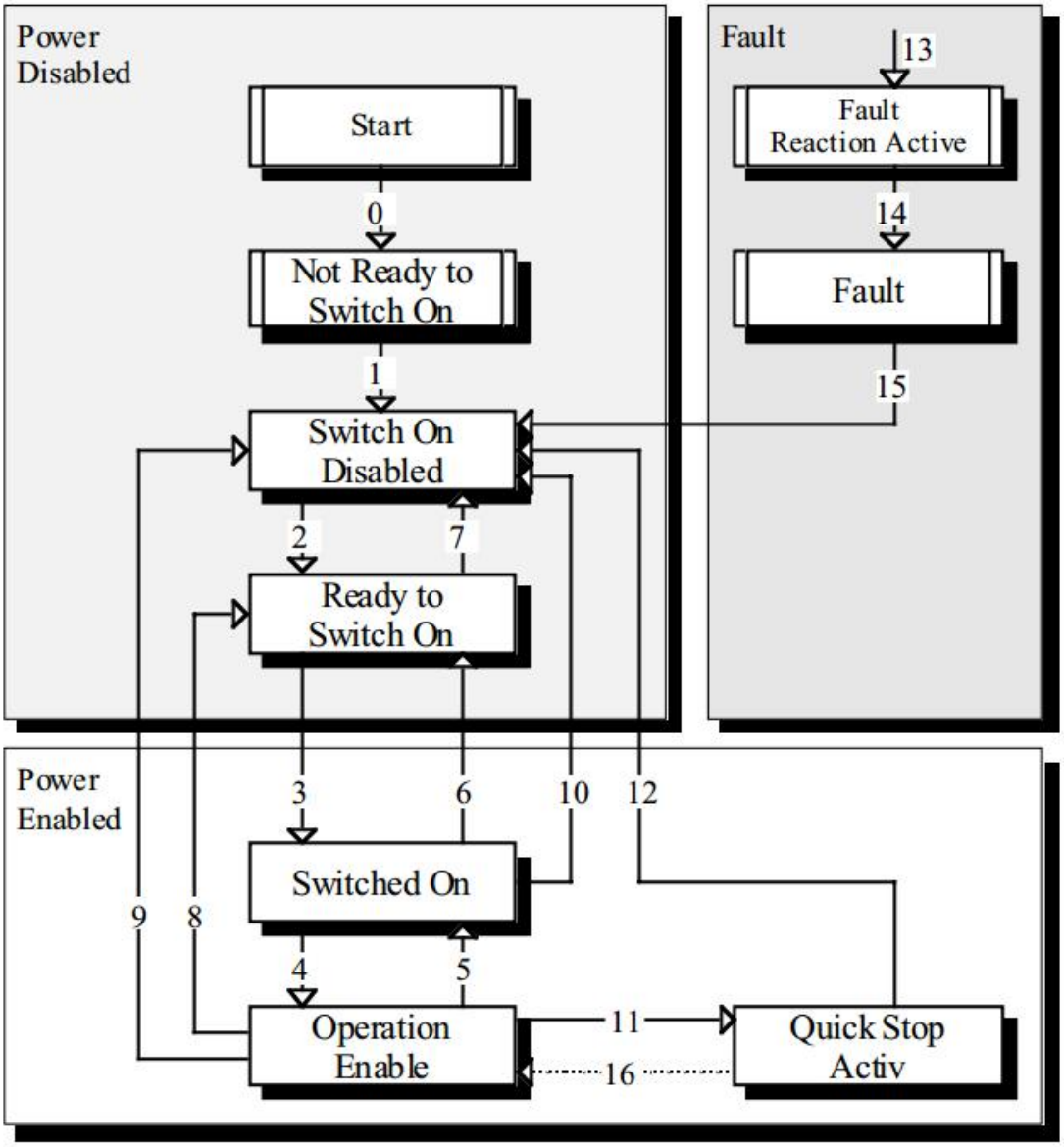
State(状态)： 如果主电激活或发生报警，伺服驱动都处在不同状态。 CANOPEN 总线控制下的状态机将在本节重点讲解。

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 contolword 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 contolword 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，可以可以进行 EtherCAT 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“ csP”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。

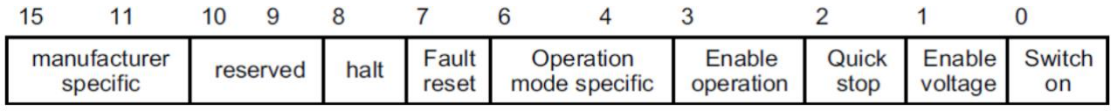
状态名	描述
Not Ready to Switch On	伺服驱动器正在初始化过程中，不能进行 CAN 通讯。
Switch On Disabled	伺服驱动器初始化完成，可以进行 CAN 通讯。

Ready to Switch On	伺服驱动器等待进入 Switch On 状态，电机没有被励磁。
Switched On	伺服驱动器伺服准备好状态，主电已上。
Operation Enable	伺服驱动器伺服给电机输入励磁信号，按照控制模式控制电机。
Quick Stop Active	伺服驱动器将根据设定的方式停机。
Fault Reaction Active	伺服驱动器检测到报警发生，按照设定的方式停机，电机仍然有励磁信号。
Fault	电机无励磁信号。

4.3、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：



4.4、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

4.5、控制方式介绍

伺服驱动器目前支持 CAN DSP402 中的 4 种控制模式：
回零模式（ HOMING MODE）
周期同步位置模式(CYCLIC SYNC POSITION MODE)
周期同步速度模式(CYCLIC SYNC VELOCITY MODE)
周期同步力矩模式(CYCLIC SYNC TORQUE MODE)

4.5.1、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

4.5.2、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、7、8、9、10
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)
9	周期同步速度模式(CYCLIC SYNC VELOCITY MODE)
10	周期同步力矩模式(CYCLIC SYNC TORQUE MODE)

4.5.3、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
Data Type	INT8
Access	RO
PDO Mapping	YES
Units	--
Value Range	1、3、4、6、8、9、10
Default Value	--

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 Z 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 Z 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	HomePosition	INT32	RW
6098 h	VAR	HomingMethod	INT8	RW
6099 h	ARRAY	SHOM_Speed	UINT32	RW

六、周期同步速度模式(CYCLIC SYNC 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: *Target velocity* not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: *Target velocity* reached Halt = 1: Axle has velocity 0
Speed	0	Speed is not equal 0
	1	Speed is equal 0
Max slippage error	0	Maximum slippage not reached
	1	Maximum slippage reached

6.3、速度控制模式相关参数

6.3.1、电机相关参数

Index	Object	Name	Type	Attr.
6069h	VAR	VelocityActualValue	INT32	RO
60FF h	VAR	TargetVelocity	INT32	RW
606Bh	VAR	VelocityDemandValue	INT32	RO
60B1 h	VAR	Velocity offset	INT32	RW

主站可以读取 VelocityActualValue 参数确定电机当前转速；
主站可以通过读 TargetVelocity 参数设置驱动器速度大小；
主站可以通过读 velocity_demand_value 参数确定驱动器当前给定速度值；

七、周期同步力矩模式(CYCLIC SYNC 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: *Target velocity* not (yet) reached Halt = 1: Axle decelerates
	1	Halt = 0: *Target velocity* reached Halt = 1: Axle has velocity 0

7.2、电流控制模式相关参数

7.2.1 电机相关参数

Index	Object	Name	Type	Attr.
6077h	VAR	TorqueActualValue	INT16	RO
6071h	VAR	TargetTorque	INT16	RW
60B2h	VAR	Torque offset	INT16	RW

主站可以读取 TorqueActualValue 参数确定电机当前电流值；
主站可以通过读 TargetTorque 参数设置驱动器电流设定值；

八、周期同步位置模式(CYCLIC SYNC 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 *target position*
	1	Assume *target position*
Change set immediately	0	Finish the actual positioning and then start the next positioning
	1	Interrupt the actual positioning and start the next positioning
abs / rel	0	*Target position* is an absolute value
	1	*Target position* is a relative value
Halt	0	Execute positioning
	1	Stop axle with *profile deceleration* (if not supported with *profile acceleration*)

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

8.3、位置控制相关参数

8.3.1、电机相关参数

Index	Object	Name	Type	Attr.
6064h	VAR	Position actual value	INT32	RO
607Ah	VAR	TargetPosition	INT32	RW
6081 h	VAR	profile_velocity	UINT32	RW
60B0 h	VAR	Position offset	UINT32	RW
6083 h	VAR	profile_acceleration	UINT32	RW
6084 h	VAR	profile_deceleration	UINT32	RW
6085 h	VAR	quick_stop_deceleration	UINT32	RW
6086 h	VAR	motion_profile_type	INT16	RW
6071 h	VAR	TargetTorque	INT16	RW

Position actual value

Position actual value 为位置的实时反馈值。

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

target_position

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

Index	607Ah
Name	target_position
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES

Units	position units
Value Range	----
Default Value	0

profile_velocity

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

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

end_velocity

end_velocity 是到达给定位置（ target_ position）时的速度。通常为了在到达给定位置时停止驱动器，该参数设置为 0；但在连续多点位置时，该值可以设置成非零值，并且默认与 profile_velocity 大小相同。（单位：转/分钟）

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

profile_acceleration

profile_acceleration 是到达给定位置期间的加速度。

Index	**6083h**
Name	profile_acceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	RPM/s
Value Range	----
Default Value	20RPM/s

profile_deceleration

profile_deceleration 是到达给定位置期间的减速度。(该加速度暂时用 profile_acceleration 代替。所以该参数暂时不支持)

Index	6084h
Name	profile_deceleration
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	RPM/s
Value Range	----
Default Value	20RPM/s

motion_profile_type

motion_profile_type 被用来选择何种速度曲线。目前位置下只支持梯形速度曲线。写 0 表示选择梯形速度曲线。(目前只支持梯形速度曲线)

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

TargetTorque

TargetTorque 被用来作为 CANOPEN 控制电流大小。(单位：0.1A)

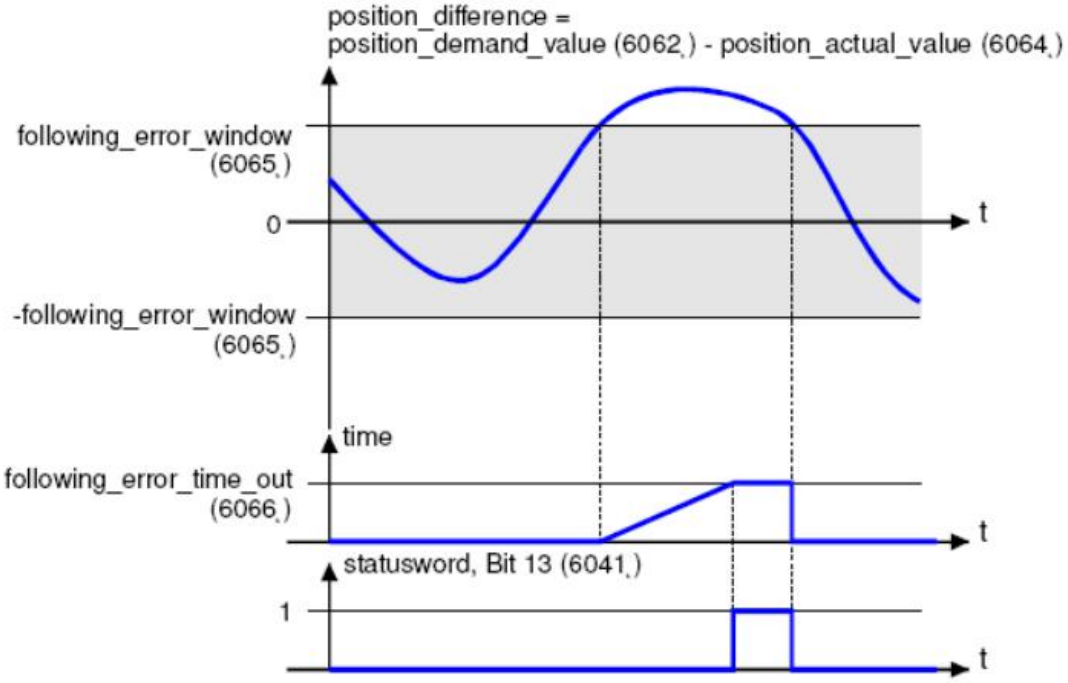
Index	6071h
Name	PositionRefMode
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	YES
Units	speed units
Value Range	----
Default Value	0

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)

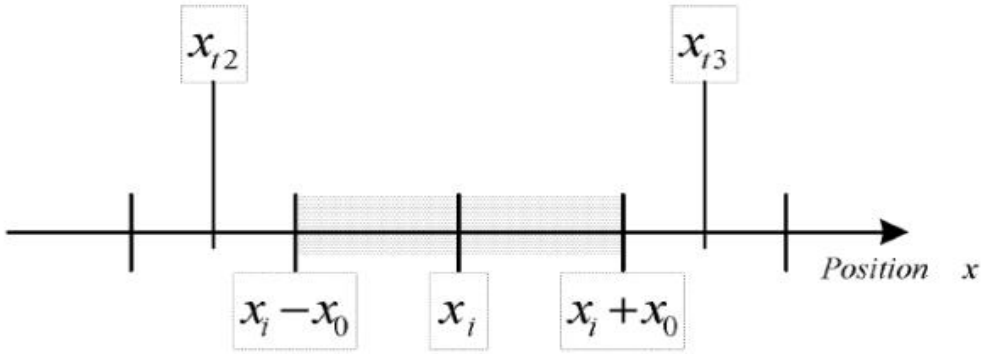
是通过电机编码器测量出。位置控制器的动作受参数设置影响。为了控制系统的稳定性，必须对位置环输出值(control_effort)限幅。该输出值作为速度环给定速度。在 Factor group 单元，所有的输入和输出值都转换成驱动器相应的内部单位。
下面介绍位置控制时的子功能：

8.4.1、跟随误差(following error)



跟随误差(following error) –功能描述

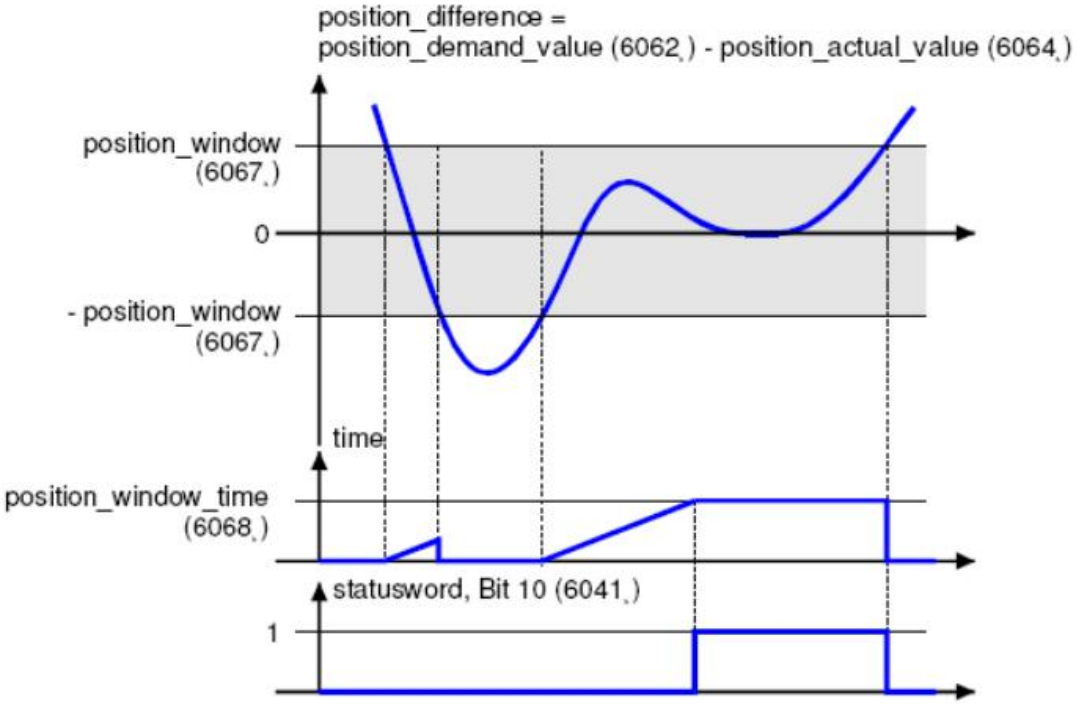
跟随误差指的是实际位置(position_actual_value)和期望位置(position_demand_value)的偏差。 上图所示，如果在设定时间(following_error_time_out)内，跟随误差值大于跟随误差窗口(following_error_windows)，那么状态字(statusword)的 bit13(following_error)将被置 1。



跟随误差(following error) –举例

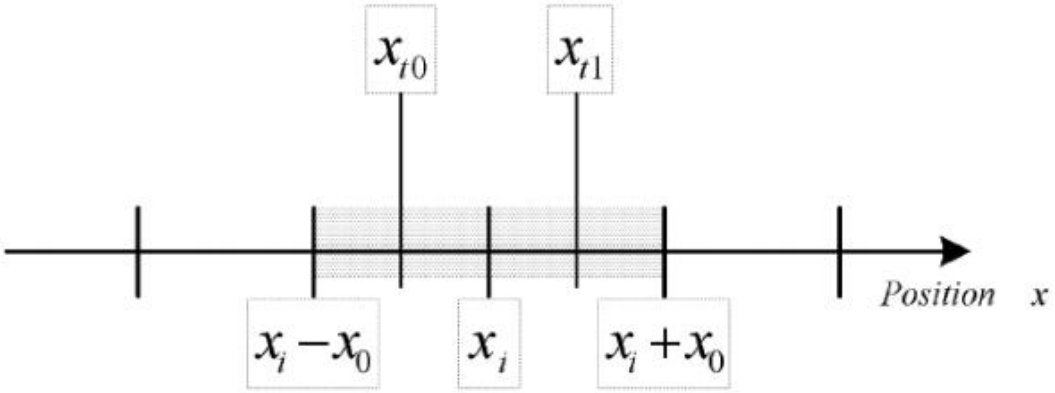
8.4.2、位置到达

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 xi-x0 到 xi+x0 区间范围。举例，位置 xt0 和 xt1 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。



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
606Ah	VAR	control_effort	INT32	RO

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

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

Index	6065h
Name	following_error_window
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES

Units	position units
Value Range	0 – 7FFFFFFF h
Default Value	256

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

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

Index	6067h
Name	position_window
Object Code	VAR
Data Type	UINT32
Access	RW
PDO Mapping	YES
Units	position units
Value Range	----
Default Value	400

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

Default Value	0

九、警报配置

关于一些报警配置需要在驱动器调试软件上操作。

十、应用举例

10.1、控制速度环

操作步骤：
（1）、对 OperationMode（索引 0x6060）写 9，表示选择速度环(CSV)，并且可以通过 DisplayMode（索引 0x6061）观察到当前工作模式；

（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。
（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）观察到控制的转速大小。

10.2、控制位置环

CSP 模式一般适用于绝对位置模式。

（1）、对工作模式 OperationMode（索引 0x6060）写 8，表示选择位置环(CSP)，并且可以通过 DisplayMode（索引 0x6061）观察到当前工作模式；
（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，在写完位置后，把 bit4 位置 1 表示把位置指令让驱动器执行。
（3）、对该参数 max_motor_speed（索引 0x6080）控制电机运行最大速度，表示位置启动后，完成加速后最终到达的速度，对 TargetPosition（索引 0x607A）写需要转到的目标位置，单位：脉冲个数，可以通过 PositionFdb（索引 0x6064）观察当前电机的位置反馈值，
（4）、主站下发的位置指令、速度指令、加速度指令、减速度指令等参数，驱动器接收到指令后会立即生效。

10.3、控制电流环

（1）、对 OperationMode（索引 0x6060）写 10，表示选择电流环(CST)，并且可以通过 DisplayMode（索引 0x6061）观察到当前工作模式；
（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。
（3）、对 TargetTorque（索引 0x6071）写需要控制的电流值，单位：默认 0.1A。可以通过 TorqueActualValue（索引 0x6077）观察当前电机的电流反馈值，也可以通过 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，目前力矩参数单位均为 0.1A，直接以电流形式表示。

10.4、保存参数

将当前驱动器参数保存到非易失性存储器中，主站需要向该映射的子索引写0x65766173，驱动器收到命令后，会把驱动器的当前参数保存到非易失性存储器中，如果保存失败，驱动器会在故障信号的 EEPROM 故障位置 1；在完成保存后，该映射的值会被清零。

名称	索引（H）	属性	数据类型
保存全部参数	1010/01	RW	Uint32
保存 CIA301 参数	1010/02	RW	Uint32
保存 DS402 参数	1010/03	RW	Uint32
保存制造商参数	1010/04	RW	Uint32

10.5、电子齿轮比

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。

驱动器计算齿轮比是通过这些对象索引： 608F,、6091,、6092

$$\text{电子齿轮比} = \frac{\dfrac{\text{Encoder_increments(608F_01)}}{\text{Motor_revolutions(608F_02)}} \times \dfrac{\text{Gear_ratio(6091_01)}}{\text{Gear_ratio(6091_02)}}}{\dfrac{\text{Feed_constant(6092_01)}}{\text{Feed_constant(6092_02)}}}$$

电子齿轮比影响到的参数：Position actual value(6064_00),
position_actual_internal_value(6063_00),
TargetPosition(607A_00),
Velocity sensor actual value(6069_00), VelocityDemandValue(606B_00),
VelocityActualValue(606C_00),
ProfileVelocity(6081_00),TargetVelocity(60FF_00) 等参数。

十一、附录

Index/SubIndex	Name	DataType	Attribute	Mappable?	Comment
1000/00	Device Type	UINT32	RO	NO	
1001/00	Error Register	UINT8	RO	NO	
1008/00	Device Name	UINT32	RO	NO	
1009/00	HardwareVersion	UINT32	RO	NO	
100A/00	SoftwareVersion	UINT32	RO	NO	
1010/01	Save Para	UINT32	Sub0:RO Sub1:RW	NO	
1018/04	Identity Object	UINT32	RW	NO	

CiA402 device profile

Axis	Name	DataType	Attribute	Mappable?	Comment
603F	ErrorCode	UINT16	RO	TxMap	
6040	ControlWord	UINT16	RW	RxMap	
6041	StatusWord	UINT16	RO	TxMap	
6060	OperationMode	INT8	RW	RxMap	
6061	DisplayMode	INT8	RO	TxMap	
605A	Quick_stop_option_code	INT16	RW	NO	
605B	Shutdown_option_code	INT16	RW	NO	
605C	Disable_operation_option_code	INT16	RW	NO	
605D	Halt_mode_code	INT16	RW	NO	
605E	Fault_reaction_option_code	INT16	RW	NO	
6062	position_demand_value	INT32	RO	TxMap	
6063	position_actual_internal_value	INT32	RO	TxMap	
6064	Position actual value	INT32	RO	TxMap	
6067	ReachPosition	UINT32	RW	NO	
607A	Position Target value	INT32	RW	RxMap	

607E	Polarity	UINT8	RW	NO	
60B0	position_offset	INT32	RW	RxMap	
60B1	velocity_offset	INT32	RW	RxMap	
60B2	torque_offset	INT16	RW	RxMap	
60FC	position_demand_internal_value	INT32	RO	TxMap	
60FA	Control_effort	INT32	RO	TxMap	
607F	max_profile_velocity	UINT32	RW	NO	
6080	max_motor_speed	UINT32	RW	NO	
6081	ProfileVelocity	UINT32	RW	RxMap	
6082	EndVelocity	UINT32	RW	RxMap	
6083	Profileacceleration	UINT32	RW	RxMap	
6084	profile_deceleration	UINT32	RW	RxMap	
6085	quick_stop_deceleration	UINT32	RW	RxMap	
6086	motion_profile_type	IINT16	RW	NO	
60C5	Max_acceleration	UINT32	RW	NO	
60C6	Max_deceleration	UINT32	RW	NO	
60FF	TargetVelocity	INT32	RW	RxMap	
6069	Velocity Sensor ActualValue	INT32	RO	TxMap	
606A	SensorSelect	INT16	RW	NO	
606B	VelocityDemandValue	INT32	RO	TxMap	
606C	VelocityActualValue	INT32	RO	TxMap	
606D	Velocity window	UINT16	RW	NO	
606F	Velocity threshold	UINT16	RW	NO	
6071	TargetTorque	INT16	RW	RxMap	
6072	Max Torque	INT16	RW	NO	
6073	Max Currentt	INT16	RW	NO	
6074	torque_demand	INT16	RO	TxMap	
6077	TorqueActualValue	INT16	RO	TxMap	
6078	Currentt ActualValue	INT16	RO	TxMap	
607C	Home Offset	INT32	RW	NO	
607E	Polarity	UINT8	RW	NO	
608F/01	Encoder_increments	UINT32	RW	NO	
608F/02	Motor_revolutions	UINT32	RW	NO	
6091/01	Gear_ratio_Motor_revolutions	UINT32	RW	NO	
6091/02	Gear_ratio_Shaft_revolutions	UINT32	RW	NO	
6092/01	Feed_constant_Feed	UINT32	RW	NO	
6092/02	Feed_constant_Shaft_revolutions	UINT32	RW	NO	

6098	Homing Method	INT8	RW	NO	
6099/01	High Homing Speed	INT32	RW	NO	
6099/02	Low Homing Speed	INT32	RW	NO	
60B0	Position offset	INT32	RW	RxMap	
60B1	Velocity offset	INT32	RW	RxMap	
60B2	Torque offset	INT16	RW	RxMap	
6502	Cia402_Driving_Mode	UINT32	RO	NO	

驱动器内部参数

Axis	Name	DataType	Attribute	Mappable?	Comment

通信配置参数

RxPDOassign（0x1C12)

SubIndex(H)	Name	DataType	Attribute	Mappable?	Comment
00	PDO 映射个数	UINT8	RW	NO	最大可设为 16
01-10	1600—1607 1610-1617	UINT16	RW	NO	

1600-1610 / 1610-1607

SubIndex(H)	Name	DataType	Attribute	Mappable?	Comment
00	PDO 映射个数	UINT8	RW	NO	最大可设为 10
01-0A	-------	UINT32	RW	NO	按照需求做映射

TxPDOassign（0x1C13)

SubIndex(H)	Name	DataType	Attribute	Mappable?	Comment
00	PDO 映射个数	UINT8	RW	NO	最大可设为 16
01-10	1A00—1A07 1A10-1A17	UINT16	RW	NO	

1A00-1A10 / 1A10-1A07

SubIndex(H)	Name	DataType	Attribute	Mappable?	Comment
00	PDO 映射个数	UINT8	RW	NO	最大可设为 10
01-0A	-------	UINT32	RW	NO	按照需求做映射