

MOTEC 智能步进驱动器 CANOPEN 使用手册

Version 2.1

MOTEC(中国)营业体系

2017-10-26

(本手册适用于 MOTEC 智能步进驱动器)

版本说明:

2015 年1 月11 日发行, Version 1.0;

2017 年10月26 日发行, Version 2.1;

版权信息:

本手册为MOTEC(中国)营业体系(以下简称“MOTEC(中国)”)版权所有。

MOTEC(中国)对本手册拥有版权, 未经书面授权, 不可将本文的全部或部分内容进行复制、翻印、收录、再加工或任何形式的转让。

本文的编著几经审校。但MOTEC(中国)不对其内容和推论中可能存在的错误担责。因用户原因使用不当而对产品或用户造成的直接或间接损失, MOTEC(中国)同样免责。使用本产品时务必遵照使用说明, 以免造成设备或人身伤害。

本文中的内容的表述力图精确、可靠, 但错误和疏忽之处在所难免, MOTEC(中国)保留随时修改和完善本文档的权利。

最新版本的使用说明书可在www.motec365.com 下载。

联系方式:

MOTEC(中国)营业体系

北京诺信泰伺服科技有限公司

地址: 北京市通州区环科中路 17 号 11B (联东 U 谷西区)

电话: 010-56298855-666

传真: 010-65546721

邮编: 100027

网址: <http://www.motec365.com>

eMail: motecSupport@sina.com

目 录

1. 概述	6
1.1 CANopen 主要文档	6
1.2 本手册使用的术语和缩写	6
1.3 CAN 概述	7
1.4 CANopen 概述	7
2. 驱动器 CAN 接口和信号	8
2.1 接线和连接	8
2.2 使用 CAN 组成网络	8
2.3 通讯硬件设定	9
2.3 CAN 通信帧结构	10
2.4 节点配置	10
2.4.1 对象 0x1000: Device type(设备类型)	10
2.4.2 对象 0x100B: Node ID(节点号)	11
2.5 设备信息	11
2.5.1 对象 0x1008: Manufacturer device name(设备名称)	11
2.5.2 对象 0x1009: Manufacturer hardware version(硬件版本)	11
2.5.3 对象 0x100A: Manufacturer software version(软件版本)	12
2.5.4 对象 0x1018: Manufacturer software version(软件版本)	12
3. CANopen 通讯基础	14
3.1 CANopen 标识符分配表	14
3.2 配置 SDO	15
3.2.1 SDO 报文传送机制	15
3.2.2 对象 0x1200: Server SDO Parameter(驱动器 SDO 参数)	15
3.2.3 SDO 报文对参数的读写格式	16
3.2.4 SDO 应用举例	16
3.2.5 SDO 错误报文格式和错误代码	17
3.3 PDO 信息	18
3.3.1 PDO 接收配置	18
3.3.2 PDO 发送配置	20
3.3.3 PDO 映射举例	22
3.4 SYNC 信息	23
3.4.1 SYNC COB-ID 配置	23
3.5 EMERGENCY 信息	24
3.5.1 EMERGENCY COB-ID 配置	24
3.6 NMT 服务管理	25
3.7 Heartbeat 信息	26
3.7.1 对象 0x1016: Producer Heartbeat Time(生产者心跳报文时间)	26
3.7.2 对象 0x1017: Producer Heartbeat Time(生产者心跳报文时间)	27
3.8 life guard 功能	27
3.8.1 对象 0x100C: Guard Time(保护时间)	27

3.8.2 对象 0x100D: Life Time Factor (生命因子)	28
4. 参数配置	29
4.1 保存参数	29
4.1.1 对象 0x1010: Save_parameters (存储参数)	29
4.2 恢复默认参数	30
4.2.1 对象 0x1011: Restore_parameters (回复默认参数)	30
5. 设备控制	32
5.1 机器状态	32
5.1.1 简述	32
5.1.2 驱动器的状态及状态转换图	32
5.2 Controlword(控制字)	33
5.2.1 控制字简述	33
5.2.2 对象 0x6040: Controlword(控制字)	33
5.3 Statusword(状态字)	35
5.3.1 状态字简述	35
5.3.2 对象 0x6041: Statusword(状态字)	35
6. 运动控制	38
6.1 描述	38
6.1.1 功能描述	38
6.1.2 对象描述	38
6.2 轮廓位置控制模式	39
6.2.1 综述	40
6.2.2 对象描述	40
6.2.3 位置运动控制举例	45
6.3 轮廓速度运动模式	46
6.3.1 综述	47
6.3.2 对象描述	47
6.4 回原点操作	48
6.4.1 综述	48
7. MOTEC 专用通道	49
7.1 对象 0x2000: OPERATING_MODE (驱动器操作模式)	49
7.2 对象 0x2001: HALF_CURRENT_VALID (半流功能相关参数)	49
7.3 对象 0x2004: POSITION_RESET_FLAG (当前位置清零)	50
7.4 对象 0x2005: POWERSUPPLY_VOLTAGE (当前电源电压)	51
7.5 对象 0x2007: DIGITAL_IN_STATUS (当前数字输入状态)	51
7.6 对象 0x2008: DIGITAL_OUT_STATUS (当前数字输出状态)	52
7.7 对象 0x2009: ANINPUT_VALUE (当前模拟量输入状态)	52
7.8 对象 0x200A: ERROR_CODE1 (驱动器报警信息)	52
7.9 对象 0x200C: CONTROL_MODE (控制模式)	53
7.10 对象 0x200D: MOTION_MODE (位置控制模式)	53
7.11 对象 0x200E: STARTMOTION_FLAG (位置控制模式启动)	53
7.12 对象 0x200F: MOTION_DONE_FLAG (位置模式定位完成)	54
7.13 对象 0x2010: STATUS_FLAG (电机运动标志)	54
7.14 对象 0x2011: JOG_POSITIVE (正向点动)	55

7.15 对象 0x2012: JOG_NEGATIVE (负向点动)	55
7.16 对象 0x2013: HOMMING_START (启动回零)	56
7.17 对象 0x2014: HOMMING_STOP (停止回零)	56
7.18 对象 0x2015: QUICK_SOFT_STOP (急停)	56
7.19 对象 0x2019: encoder_value (编码器计数)	57
7.20 对象 0x201A: ENABLE_FLAG (使能/释放)	57
7.21 对象 0x2021 - 0x2038: CANopen_OBJ_RESERVED (保留对象字典)	57
8. 简化 DS402 状态机控制方式	59
8.1 关于 MOTEC 自定义的控制字状态字等的介绍	59
8.1.1 简化 DS402 状态机控制字	59
8.1.2 简化 DS402 状态机状态字	60
8.1.3 简化 DS402 状态机位置字, 速度字	62
8.2 简化 DS402 状态机位置字运动控制相关参数和流程描述	63
8.2.1 位置控制模式	63
8.2.2 速度模式	64
8.2.3 回零模式	64
8.2.4 点动控制	64
9. 联系方式	65

1. 概述

1.1 CANopen 主要文档

文档名称	文档定义
CiA Draft Standard 301 (Version 4.02)	Application Layer and Communication Profile
CiA Draft Standard 402 (Ver. 1.2)	CANopen Device Profile

1.2 本手册使用的术语和缩写

CAN 控制器局域网

CiA 在自动化国际用户和制造商协会中的 CAN。

COB 通讯对象在 CAN 网络上的一个传输单元。数据在 COB 内部沿着整个网络传输。COB 本身是 CAN 消息帧的一部分。

LMT 层管理，CAN 给定模型中的 CAN 应用层服务元素之一。它用来配置 CAN 给定模型中每层的参数。

NMT 网络管理，CAN 给定模型中的 CAN 应用层服务元素之一。它负责 CAN 网络上的初始化、配置和故障处理。

OD 在本地存储某个设备所识别的所有通讯对象（COB）。

参数 参数是驱动器的一个操作指令。可以使用驱动器操作面板或者通过 CAN 来读取和修改参数。

PDO 进程数据对象，一种 COB。用来传输时间关键数据，比如控制命令、状态字和实际值。

RO 表示只读访问。

RW 表示读/写访问。

WO 表示只写访问。

SDO 服务数据对象，一种 COB。用来传输非时间关键数据，比如运动的参数，细分。

1.3 CAN 概述

CAN 总线是一种串行通信协议，具有较高的通信速率的和较强的抗干扰能力，可以作为现场总线应用于电磁噪声较大的场合。由于 CAN 总线本身只定义 ISO/OSI 模型中的第一层（物理层）和第二层（数据链路层），通常情况下 CAN 总线网络都是独立的网络，所以没有网络层。在实际使用中，用户还需要自己定义应用层的协议，因此在 CAN 总线的发展过程中出现了各种版本的 CAN 应用层协议，现阶段最流行的 CAN 应用层协议主要 CANopen、DeviceNet 和 J1939 等协议。

1.4 CANopen 概述

CANopen 是一种架构在 CAN（控制局域网路）上的高层通讯协定，包括通讯子协定及设备子协定常在嵌入式系统中使用，也是工业控制常用到的一种现场总线。使用 CANopen 相连设备的硬件带有一个符合 ISO 11898 标准的 CAN 收发器和一个 CAN 控制器。

MOTEC 智能步进驱动器集成了 CANopen 协议，方便用户通过各种上位机（PLC、HMI 或其它运动控制器）通讯。MOTEC 智能步进驱动器采用集成 CAN 芯片进行 CANopen 通讯，使用此功能可驱动 MOTEC 智能步进驱动器、变更参数以及监视 MOTEC 智能步进驱动器状态等多项功能。

MOTEC 智能步进驱动器符合 CiA DS-301 协议。包括周期和事件驱动型通讯，不仅能够将总线负载减少到最低限度，而且还能确保极短的反应时间。它可以在较低的波特率下实现较高的通讯性能，从而减少了电磁兼容性问题，并降低了电缆成本。

MOTEC 智能步进驱动器定义了直接访问变频器参数机制以及时间关键进程数据通讯，满足 CiA DS-402（变频器和运动控制）。

2. 驱动器 CAN 接口和信号

2.1 接线和连接

MOTEC 智能步进驱动器提供三种上位机连接方式 RS232、RS485 和 CAN，使用 CANopen 协议的时候需要使用 CAN 接口, 接口 CN3 和 CN4 完全相同。CAN3 和 CAN4 的接口图见图 2. 1，对应的引脚列表见表 2. 1.

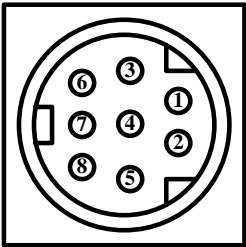


图 2.1 MOTEC 智能步进智能步进驱动器 CANopen 通讯硬件接口定义

表 2.1 CN3&CN4 网络插头接口表

CN3 和 CN4 网络插头(母头)			
Pin	Signal	Pin	Signal
1	GND	5	RS232R
2	RS485A	6	CANH
3	RS485B	7	RS232T
4	CANL	8	+5V

MOTEC 智能步进的 CANopen 所用的物理介质是符合 ISO 11898 标准，采用分驱动机制和公共反馈的双线总线。总线的最大长度取决于通讯速度，具体规定如下：

表 2.2 CAN 总线波特率与通信距离关系

通信波特率	最大总线长度
500k bit/s	100m
250k bit/s	250m
125k bit/s	500m
50k bit/s	1000m

2.2 使用 CAN 组成网络

驱动器 CN3 和 CN4 都可以作为通讯电缆输入端子，CN3 和 CN4 也都可以作为通讯电缆输出端子（如果还需连接从站，电缆从该端子连接到下一从站设备；如果不需连接其他从站，可以在该端子加终端电阻）。多台 MOTEC 智能步进驱动器连接时，CN3 和 CN4 一个作为通信输入口，一个作为通信输出口，如图 2. 2。

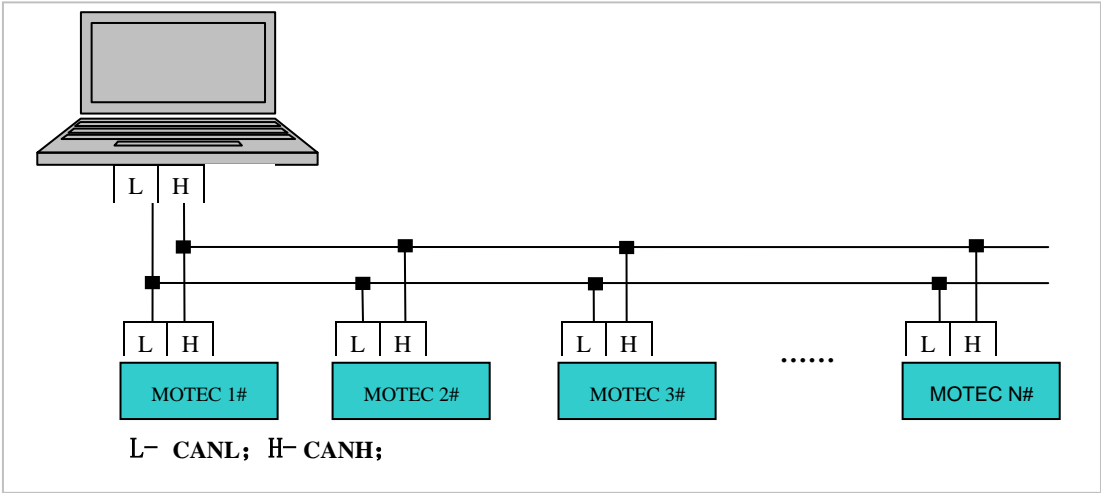


图 2.2 MOTEC 智能步进使用 CAN 组成网络

举例，网络由一 PLC 和 A、B、C 三台 MOTEC 智能步进驱动器组成，电缆接线如下：PLC→驱动器 MOTEC 1 的 CN3，MOTEC 1 的 CN4 →驱动器 MOTEC 2 的 CN3，MOTEC 2 的 CN4 →驱动器 MOTEC 3 的 CN3，MOTEC 3 的 CN4 → 120 欧姆终端电阻。CAN 总线线路必须用在 CAN_L 和 CAN_H 线之间每端连接的 120 欧姆（1%，1/4W）电阻来终接，驱动器内不包含 120Ω 的终端电阻，用户需要根据需要外接。如图 2.3 下所示。

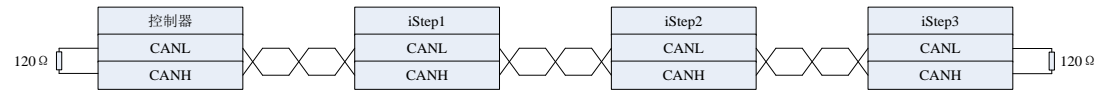


图 2.3 MOTEC 智能步进使用 CAN 组成网络示例

- 使用 CAN 通讯时，MOTEC 智能步进支持最大通讯速率为 500k bit/s，通讯速率可使用拨码开关调整；
- CAN 通讯口最多可以支持连接 110 台 MOTEC 智能步进驱动器，并可扩展；
- 实际通讯速率和通讯节点数量受使用环境与连接电缆长度等因素影响；

2.3 通讯硬件设定

为了方便用户使用，我们特别提供一组拨码开关来快速设定驱动器地址和通信波特率，其中 P1~P6 表示驱动器的地址，P7~P8 表示网络模式下的通信速率，具体如图 2.4 和表 2.3 所示。

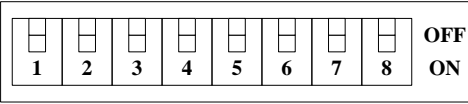



图 2.4 MOTEC 智能步进拨码开关示意


表 2.3 MOTEC 智能步进拨码开关设置

Switch : ON=1,OFF=0					
P6, 5, 4, 3, 2, 1	ID No.	Baud Rate			
000000	NULL	P8, 7	RS232&RS485	CAN	
000001	1	00	9600	50K	
000010	2	01	19200	125k	
...	10	38400	250k	
111111	63	11	115200	500k	

当拨码开关 P6~P1 都处于 OFF 的状态时，驱动器进入自测模式。这样无需连接外部任何控制信号，驱动器会控制电机以一定的速度匀速转动，可以用来诊断驱动器驱动部分电路工作是否正常。但驱动器进入正式工作阶段，不能将拨码开关 P6~P1 都处于 OFF 的状态。



注意：当拨码开关 P6~P1 都处于 OFF 的状态时，驱动器进入自测模式。因此，驱动器在调试和工作的时候拨码开关 P6~P1 不能都处于 OFF 状态，也就是说驱动器的地址不能为 0。



注意：驱动器与主站通讯的通讯速率和地址是通过驱动器上的拨码开关设置的，拨码开关的设置情况只在驱动器上电初始化时做检测，驱动器上电后进行拨码开关的设置将不影响驱动器的通讯速率和地址。通讯速率和地址的改变只有在驱动器重新上电后生效。

2.4 CAN 通信帧结构

CAN 通过数据帧在主机（控制器）和总线节点之间传输数据。表 2.4 说明了数据帧的结构。

表 2.4 CAN 通信数据帧的结构

帧头	仲裁域		控制域	数据域	校验域	应答域	帧尾
	COB-ID (通讯对象标识符)	RTR (远程请求)					
1 位	11 或 29 位	1 位	6 位	0~8 字节	16 位	2 位	7 位

2.5 节点配置

节点号由驱动器拨码开关配置(该值需要重新上电才生效)，节点号也可通过读取对象 0x100B 获得。

2.5.1 对象 0x1000：Device type(设备类型)

0x1000 对象定义了设备的设备类型。

对象描述：

Index	0x1000
Name	Manufacturer device name
Object Code	VAR
Data Type	UNIT32
Access	RO
PDO Mapping	NO
Units	--
Value Range	--
Default Value	402

数据描述：

数值	含义
00420192h	DS402

2.5.2 对象 0x100B: Node ID(节点号)

0x100B 对象反映了当前驱动器的节点号。

对象描述:

Index	0x100B
Name	Node ID
Object Code	VAR
Data Type	UINT8
Access	RO
PDO Mapping	NO
Units	--
Value Range	0~255
Default Value	1

2.6 设备信息

设备信息描述了本设备的硬件和软件的版本编号, 以及设备名称, 由下面几个对象定义。

索引	名称	备注
0x1008	Manufacturer device name	设备名称
0x1009	Manufacturer hardware version	硬件版本信息
0x100A	Manufacturer software version	软件版本信息
0x1018	Identity_Object	设备信息

2.6.1 对象 0x1008: Manufacturer device name(设备名称)

0x1008 对象描述了设备名称。

对象描述:

Index	0x1008
Name	MODEL_NUMBER
Object Code	VAR
Data Type	UINT16
Access	RO
PDO Mapping	NO
Units	--
Value Range	0~65535
Default Value	--
Description	101XX: 2 相智能步进驱动器 10100-SD266B 201XX: 3 相智能步进驱动器 20100 - SD388B;20101 - SD3228B 301XX: 空心杯驱动器 401XX: 低压伺服驱动器 501XX: 交流伺服 Beta 版驱动器

2.6.2 对象 0x1009: Manufacturer hardware version(硬件版本)

0x1009 对象描述了设备硬件版本。驱动器硬件版本, e. g. 30201, Version3.02.01

对象描述:

Index	0x1009
Name	Manufacturer hardware version
Object Code	VAR
Data Type	UINT16
Access	R0
PDO Mapping	NO
Units	--
Value Range	0~65535
Default Value	--
Description	e.g. 30201, Version3.02.01

2.6.3 对象 0x100A: Manufacturer software version(软件版本)

该对象描述了设备软件版本。驱动器软件版本

对象描述:

Index	0x100A
Name	Manufacturer software version
Object Code	VAR
Data Type	VISIBLE_STRING
Access	R0
PDO Mapping	NO
Units	--
Value Range	0~65535
Default Value	--
Description	e.g. 30201, Version3.02.01

2.6.4 对象 0x1018: Manufacturer software version(软件版本)

对象描述:

Index	0x1018
Name	Identity_Object
Object Code	VAR

目录描述:

Sub-Index	0
Description	Number of entries(条目数量)
Access	R0
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	3
Default	3

Sub-Index	1
Name	Vendor_ID
Data Type	UINT32
Category	Mandatory
Access	RW
PDO Mapping	NO
Units	--
Value Range	0~4294967295
Default Value	0

Sub-Index	2
Name	Productcode
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO
Units	--
Value Range	0~4294967295
Default Value	0

Sub-Index	3
Name	Revisionnumber
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO
Units	--
Value Range	0~4294967295
Default Value	0

3. CANopen 通讯基础

本章将介绍 CANopen 通信的基本协议 DS301 规定的部分内容

3.1 CANopen 标识符分配表

为了简化配置工作，CANopen 定义了“预定义主/从连接集”。这种默认的 CAN 标识符分配方案，对于最多具有 127 个 CANopen 设备的简单网络而言，在不需要配置 COB-ID 参数的情况下，CANopen 的设备能够进入正常工作状态。

预定义主/从连接集的 11 位 CAN 标识符包含一个 4 位功能代码和一个 7 位节点 ID（设备标识符）。功能代码用来设定服务类型，如 PDO、SDO、NMT 等以及设置消息的优先级；节点 ID 将消息明确地与设备对应起来。其中 COB-ID（通讯对象标识符）分配如表 2.5，CANopen 的功能代码如表 2.6。

表 2.5 COB 通讯对象标识符

功能码				NODE ID（节点地址）						
10	9	8	7	6	5	4	3	2	1	0

表 2.6 CAN 标识符分配表

通信对象	功能代码	COB-ID 范围	相应的对象字典
广播或组播消息			
网络管理 NMT	0000b	000h	
同步报文 SYNC	0001b	080h	1005h、1006h、1007h、 1028h
点对点消息			
紧急事件 Emergency	0001b	081h~0FFh	1014h、1015h
TPDO1	0011b	181h~1FFh	1800h
RPDO1	0100b	201h~27Fh	1400h
TPDO2	0101b	281h~2FFh	1801h
RPDO2	0110b	301h~37Fh	1401h
TPDO3	0111b	381h~3FFh	1802h
RPDO3	1000b	401h~47Fh	1402h
默认 SDO（tx）	1011b	581h~5FFh	1200h
默认 SDO（rx）	1100b	601h~67Fh	1200h
网络管理错误控制（Heartbeat）	1110b	701h~77Fh	1016h、1017h

注意：

- 1、PDO 和 SDO 的发送/接收是由（slave）CANopen 从站节点方观察的。
- 2、本驱动器的 CANopen 暂支持 3 个发送 PDO，3 个接收 PDO。

3.2 配置 SDO

3.2.1 SDO 报文传送机制

SDO 用来访问一个设备的对象字典。访问者被称作客户(client)，对象字典被访问且提供所请求服务的 CANopen 设备被称作服务器(server)。客户的 CAN 报文和服务器的应答 CAN 报文总是包含 8 字节数据（尽管不是所有的数据字节都一定有意义）。一个客户的请求一定有来自服务器的应答。

SDO 基本结构如下：

Client Server/Server Client

Byte0	Byte1~2	Byte3	Byte4~7
SDO 命令字	对象索引	对象子索引	数据

3.2.2 对象 0x1200: Server SDO Parameter(驱动器 SDO 参数)

这个对象是定义步进驱动器的 SDO 参数，其中包括接收 ID: COB-ID client-server (Rx) 和发送 ID: COB-ID client-server (Tx)。

对象描述：

Index	0x1200
Name	Server SDO Parameter
Object Code	ARRAY

目录描述：

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	2
Default	2

Sub-Index	1
Description	COB-ID client-server (Rx)
Access	RO
Data Type	UINT32
PDO Mapping	NO
Units	--
Value Range	--
Default	0x600 + Node ID

Sub-Index	2
Description	COB-ID client-server (Tx)
Access	RO
Data Type	UINT32
PDO Mapping	NO

Units	---
Value Range	---
Default	0x580 + Node ID

3.2.3 SDO 报文对参数的读写格式

SDO 报文对参数的操作格式分为读和写，对于不同长度的数据，使用的命令字不同

读取对象字典

读取 8 位数据，数据格式 UINT8/INT8

	Byte0	Byte1	Byte2	Byte3	Byte4
命令	40h	对象索引低位	对象索引高位	子索引	
回答	4Fh	对象索引低位	对象索引高位	子索引	数据

读取 16 位数据，数据格式 UINT16/INT16

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
命令	40h	对象索引低位	对象索引高位	子索引		
回答	4Bh	对象索引低位	对象索引高位	子索引	数据	数据

读取 32 位数据，数据格式 UINT32/INT32

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
命令	40h	对象索引低位	对象索引高位	子索引				
回答	43h	对象索引低位	对象索引高位	子索引	数据	数据	数据	数据

写入对象字典

写入 8 位数据，数据格式 UINT8/INT8

	Byte0	Byte1	Byte2	Byte3	Byte4
命令	2Fh	对象索引低位	对象索引高位	子索引	数据
回答	60h	对象索引低位	对象索引高位	子索引	

写入 16 位数据，数据格式 UINT16/INT16

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
命令	2Bh	对象索引低位	对象索引高位	子索引	数据	数据
回答	60h	对象索引低位	对象索引高位	子索引		

写入 32 位数据，数据格式 UINT32/INT32

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
命令	23h	对象索引低位	对象索引高位	子索引	数据	数据	数据	数据
回答	60h	对象索引低位	对象索引高位	子索引				

3.2.4 SDO 应用举例

读取对象字典

读取索引 6061h 子索引 00h 中的数据 (INT8)，假设为 12h

	Byte0	Byte1	Byte2	Byte3	Byte4
命令	40h	61h	60h	00h	
回答	4Fh	61h	60h	00h	12h

读取索引 6074h 子索引 00h 中的数据 (INT16)，假设为 1234h

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
命令	40h	74h	60h	00h		
回答	4Bh	74h	60h	00h	34h	12h

读取索引 6093h 子索引 02h 中的数据 (UINT32), 假设为 12345678h

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte65	Byte7
命令	40h	93h	60h	02h				
回答	43h	93h	60h	02h	78h	56h	34h	12h

写入对象字典

向索引 6060h 子索引 00h (INT8) 中写入数据 12h

	Byte0	Byte1	Byte2	Byte3	Byte4
命令	2Fh	60h	60h	00h	12h
回答	60h	60h	60h	00h	

向索引 6073h 子索引 00h (INT16) 中写入数据 1234h

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5
命令	2Bh	73h	60h	00h	34h	12h
回答	60h	73h	60h	00h		

向索引 6093h 子索引 02h (UINT32) 中写入数据 12345678h

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte65	Byte7
命令	23h	93h	60h	02h	78h	56h	34h	12h
回答	60h	93h	60h	02h				

3.2.5 SDO 错误报文格式和错误代码

SDO 错误报文格式:

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte65	Byte7
命令	XXh	索引低位	索引高位	子索引	数据	数据	数据	数据
回答	80h	索引低位	索引高位	子索引	F0	F1	F2	F3

80h 是 SDO 错误报文的命令码, F0、F1、F2、F3 是 SDO 错误报文的错误代码, 具体意义见表 3.1

表 3.1 SDO 错误代码表

F3	F2	F1	F0	代码功能描述
05	03	00	00h	触发位没有交替改变
05	04	00	01h	非法或未知的 Client/Server 命令字
06	01	00	00h	对象不支持访问
06	01	00	01h	试图读只写对象
06	01	00	02h	试图写只读对象
06	02	00	00h	对象字典中对象不存在
06	04	00	41h	对象不能够映射到 PDO
06	04	00	42h	映射的对象的数目和长度超出 PDO 长度
06	04	00	43h	一般性参数不兼容
06	04	00	47h	一般性设备内部不兼容
06	06	00	00h	硬件错误导致对象访问失败
06	07	00	10h	数据类型不匹配, 服务参数长度不匹配
06	07	00	12h	数据类型不匹配, 服务参数长度太大
06	07	00	13h	数据类型不匹配, 服务参数长度太短
06	09	00	11h	子索引不存在
06	09	00	30h	超出参数的值范围(写访问时)

06 09 00 31h	写入参数数值太大
06 09 00 32h	写入参数数值太小
06 09 00 36h	最大值小于最小值
08 00 00 00h	一般性错误
08 00 00 20h	数据不能传送或保存到应用
08 00 00 21h	由于本地控制导致数据不能传送或保存到应用

3.3 PDO 信息

本设备最多可允许 3 通道发送 PDO 和 3 通道的接收 PDO，在使用 PDO 前需要进行简单的配置，COB-ID 和接收、发送对象地址映射。

3.3.1 PDO 接收配置

3.3.1.1 对象 0x1400-0x1402: Receive PDO(1-3) Parameter (接收 PDO(1-3) 参数)

0x1400-0x1402 对象定义了步进驱动器的接收 PDO(1-3) 的参数，其中包括接收 ID: COB_ID_ PDO(1-3) (Rx) 和传送类型 Transmission_Type。

对象描述:

Index	0x1400 -0x1402
Name	Receive PDO Parameter
Object Code	ARRAY

目录描述:

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	2
Default	2

Sub-Index	1
Description	COB_ID_ PDO(RX)
Access	RW
Data Type	UINT32
PDO Mapping	NO
Units	---
Value Range	Index 1400h: 0x201 - 0x27F Index 1401h: 0x301 - 0x37F Index 1402h: 0x401 - 0x47F
Default	Index 1400h: 0x200+node_ID Index 1401h: 0x300+node_ID Index 1402h: 0x400+node_ID

Sub-Index	2
Description	Transmission_Type
Access	RW
Data Type	UINT8
PDO Mapping	NO
Units	---
Value Range	0~255
Default	254

发送类型说明:

值	说明
0	异步、同步传输，当 PDO 触发条件成立后，PDO 在收到，同步 SYNC 后，传送 PDO 讯号。
1-240	同步传输：当收到的 SYNC 讯号等于设定值后，PDO 同步发送出。
254, 255	异步传输：该传输的间隔由“Event timer”决定，每“Event timer”的时间就发送一帧 PDO 数据。

3.3.1.2 对象 0x1600-0x1602: Receive PDO(1-3) Mapping(接收 PDO(1-3)映射)

0x1600-0x1602 对象是定义步进驱动器的接收 PDO(1-3)的映射参数，其中 Sub_Index0 表示有多少个对象需要映射，在 Sub_Index(1-8)中记载了映射的对象及子对象，还有该子对象的数据长度，它的格式是：0xMMMSSL，其中 MMM (Index)，SS(Sub_Index)，LL(Lenght)。

例如：要接收映射一个对象为 0x6040(ControlWord)，它没有子对象故 SS=0x0，它的长度是 16Bit，所以 LL=0x10，最后 Sub_Index0 = 1；Sub_Index1 = 0x60400010。如此类推，一通道的接收 PDO 最多可以容纳总共 64Bit 长度的数据。

对象描述:

Index	0x1600 -0x1602
Name	Receive PDO(1-3) Mapping
Object Code	ARRAY

目录描述:

Sub-Index	0
Description	Number of entries(条目数量)
Access	RW
Data Type	UINT8
PDO Mapping	NO
Units	---
Value Range	0~4
Default	0

Sub-Index	1-4
Description	Mapping Parameter
Access	RW
Data Type	UINT32
PDO Mapping	NO
Units	---
Value Range	0~0xFFFFFFFF
Default	---

3.3.2 PDO 发送配置

3.3.2.1 对象 0x1800-0x1802: Transmit PDO(1-3) Parameter(发送 PDO(1-3) 参数)

0x1800-0x1802 对象是定义步进驱动器的发送 PDO(1-3) 的参数。

对象描述:

Index	0x1800 -0x1802
Name	Transmit PDO (1-3) Parameter
Object Code	ARRAY

目录描述:

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	---
Value Range	5
Default	5

Sub-Index	1
Description	COB_ID_used_by_PDO(发送 PDO 使用的 ID 号)
Access	RW
Data Type	UINT32
PDO Mapping	NO
Units	---
Value Range	Index 1800h: 0x181 - 0x200 Index 1801h: 0x281 - 0x300 Index 1802h: 0x381 - 0x400
Default	Index 1800h: 0x180+node_ID Index 1801h: 0x280+node_ID Index 1802h: 0x380+node_ID

Sub-Index	2
-----------	---

Description	Transmission_Type(传送类型)
Access	RW
Data Type	UINT8
PDO Mapping	NO
Units	---
Value Range	0xff
Default	255

值说明:

值	说明
0	异步、同步传输，当 PDO 触发条件成立后，PDO 在收到，同步 SYNC 后，传送 PDO 讯号。
1-240	同步传输：当收到的 SYNC 讯号等于设定值后，PDO 同步发送出。
254, 255	异步传输：该传输的间隔由“Event timer”决定，每“Event timer”的时间就发送一帧 PDO 数据。

Sub-Index	3
Description	Inhibit_Time (抑制时间)
Access	RW
Data Type	UINT16
PDO Mapping	NO
Units	100uS
Value Range	0~65535
Default	0

注：“Inhibit_Time”是指，在收到同步信号后，PDO 报文需要延时 $\text{Inhibit_Time} \times 100\mu\text{s}$ 的时间才发出，以此避免在多 PDO 报文同时发出时，所引起的时间冲突。

Sub-Index	4
Description	保留
Access	---
Data Type	---
PDO Mapping	---
Units	---
Value Range	---
Default	---

注：该子对象保留。

Sub-Index	5
Description	Event_Timer (事件时间)
Access	RW
Data Type	UINT16

PDO Mapping	NO
Units	mS
Value Range	0~65535
Default	0

注：在“Transmission_Type”的值设为 255 时，“Event_Timer”是 PDO 报文发送的间隔值。

3.3.2.2 对象 0x1A00-0x1A02: Transmit PDO(1-3) Mapping(发送 PDO(1-3)映射)

这个对象是定义步进驱动器的发送 PDO(1-3)的映射参数，其中 Sub_Index0 表示有多少个对象需要映射，在 Sub_Index(1-4)中记载了映射的对象及子对象，还有该子对象的数据长度，它的格式是：0xMMMSLL，其中 MMM (Index)，SS(Sub_Index)，LL(Lenght)。

例如：要发送映射一个对象为 0x6041(StatusWord)，它没有子对象故 SS=0x0，它的长度是 16Bit，所以 LL=0x10，最后 Sub_Index0 = 1；Sub_Index1 = 0x60410010。如此类推，一通道的发送 PDO 最多可以容纳总共 64Bit 长度的数据。

对象描述：

Index	0x1A00 -0x1A02
Name	Transmit PDO(1-3) Mapping
Object Code	ARRAY

目录描述：

Sub-Index	0
Description	Number of entries(条目数量)
Access	RW
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	0-4
Default	0

Sub-Index	1-4
Description	Mapping Parameter
Access	RW
Data Type	UINT32
PDO Mapping	NO
Units	--
Value Range	0~0xFFFFFFFF
Default	0

3.3.3 PDO 映射举例

将下表中 2 个对象映射到 TPD02，TPD02 为异步周期性类型，周期时间 100ms，禁止时间 20ms。

对象	索引—>子索引	对象数据类型	对象意义
Position_actual_value	6063h—>0h	int32_t	实际位置
Velocity_actual_value	606Ch—>0h	int32_t	实际转速

1)、清除 TPD02 的映射个数:

1A01h—>0h = 0 ; 对应的 SD0 报文: 2F 01 1A 00 00h;

2)、设置映射的对象字典索引和子索引:

设置 TPD02 第一个映射为 6063—>0, int32_t 的数据长度为 20h,

1A01h—>1h = 6063 00 20h, 对应的 SD0 报文: 23 01 1A 01 20 00 63 60h;

设置 TPD02 第二个映射为 606C—>0, int32_t 的数据长度为 20h;

1A01h—>2h = 606C 00 20h, 对应的 SD0 报文: 23 01 1A 02 20 00 6C 60h;

3)、设置 TPD02 的映射个数:

1A01h—>0h = 2h, 对应的 SD0 报文: 2F 01 1A 00 02h;

4)、设置 TPD0 的通信参数:

设置 TPD02 为异步周期发送,

1801h—>02h = FFh, 对应的 SD0 报文: 2F 01 18 02 FFh;

设置发送周期为 100ms,

1801h—>05h = 64h, 对应的 SD0 报文: 2B 01 18 05 64 00h;

设置禁止时间为 20ms,

1801h—>03h = 14h, 对应的 SD0 报文: 2B 01 18 03 14 00h;

5)、映射完成。

3.4 SYNC 信息

3.4.1 SYNC COB-ID 配置

在网络范围内同步: 作为在整个网络范围内当前输入值基准, 同时保存或者随后传送(如果需要), 根据前一个 SYNC 后接收到的报文更新输出值。

MOTEC 智能驱动器应用于主从模式: SYNC 主节点定时发送 SYNC 对象, MOTEC 智能驱动器作为从节点收到 SYNC 报文后同步执行任务。

CANopen 建议用一个最高优先级的 COB-ID 以保证同步信号正常传送。SYNC 报文可以不传送数据以使报文尽可能短。SYNC 报文的 COB-ID 固定为 80h, COB-ID 可以从对象字典的 1005h 读出。

3.4.1.1 对象 0x1005: SYNC COB_ID(同步信息 ID 号)

0x1005 对象定义了同步信息接收邮箱的 ID 号。

对象描述:

Index	0x1005
Name	SYNC COB_ID
Object Code	VAR
Data Type	UINT32

Access	RO
PDO Mapping	NO
Units	--
Value Range	0x80

3.5 EMERGENCY 信息

当本机发生故障时，本机主动发出 EMERGENCY 信息。

3.5.1 EMERGENCY COB-ID 配置

EMERGENCY COB-ID 由本机自动根据 Node_ID 配置，其值是 0x80+Node_ID。

3.5.1.1 对象 0x1014: Emergency COB ID(紧急情况 ID 号)

0x1014 对象定义了设备在紧急情况发送邮箱 ID 号, 该值是 0x80 + Node ID。

对象描述:

Index	0x1014
Name	Emergency COB ID
Object Code	VAR
Data Type	UINT32
Access	RO
PDO Mapping	NO
Units	--
Value Range	0x80+Node_ID

3.5.1.2 Emergency 数据说明

当驱动器发生一个错误后，就会发送一帧 Emergency 信息，当有新的其它错误发生或者报警被清除后，才发送下一帧 Emergency 信息。Emergency 信息的数据结构为：

数据长度为 8 字节，

0, 1	2	3	4	5	6	7
Emergency Error Code	Errorregister(Object1001H)	Errorregister(Object1001H)	Manufacturer specific Error Field			

Manufacturer specific Error Field 为驱动器对象 0x200A，详细的故障信息请参考下表也可以参考对象 0x200A 的说明。

MOTEC 智能步进驱动器报警一览表

编号	LED 代码	故障信息	驱动器动作
1	01	系统错误	发布报警，驱动器不能正常工作
2	02	IPM 故障	发布报警，驱动器不能正常工作
4	04	驱动器内无参数	发布报警，电机释放
5	05	编码器计数方向与电机旋转方向相反	发布报警，电机释放
6	06	位置超差报警	发布报警，电机释放
7	07	供电电压过低	发布报警，电机释放

8	08	供电电压过高	发布报警，电机释放
9	09	参数设置超出允许范围	

3.5.1.3 对象 0x1001: Error Register (故障寄存器)

这个对象描述设备故障状态下的故障。

对象描述:

Index	0x1001
Name	Error register
Object Code	VAR
Data Type	UINT8

Access	RO
PDO Mapping	NO
Units	--
Value Range	0~255
Default Value	0

Error register 中数值说代表的内容如下表所示:

Bit	内容
0	Generic error
1	Current error
2	Voltage error
3	Temperature error
4	Communication error
5	Device profile specific
6	Reserved (always 0)
7	Manufacturer specific

其中如果相应的位置 1，则表示有故障发生。

3.6 NMT 服务管理

NMT 服务是在 CANOpen 网络中，主机控制从机节点的一种服务。其帧 ID 号是 0，利用这个 ID 发送命令给一个或者多个 CANOpen 从机。每个命令包含两个字节，第一个字节是命令代码，第二个字节是要控制的节点代码；当节点代码为 0 时，所有节点同时响应命令。

NMT 命令包含以下几种。

命令代码	含义	目标状态
0x01	开启远程节点: (开启 PDO 和 SDO 服务)	Operational
0x02	关闭远程节点: (关闭所有该 CANOPEN 节点的服务, 只保留 heartbeat 和 NMT)	Stopped
0x80	进入预操作状态: (开启 SDO 服务)	Pre-Operational
0x81	应用复位: (恢复应用数据, 将使用中的变量, 恢复为初始值)	Reset Application
0x82	连接复位: (复位硬件连接)	Reset Communication

3.7 Heartbeat 信息

Heartbeat 信息是主机监视从机的信号，从机会周期性地发送信息，主机可以检查接收到的心跳信息，进行相应的处理。Heartbeat 信息的帧 ID 号是 0x700+Node_ID，数据长度是 1 字节，其内容含义如下表：

Value	状态
0	Boot-up: (节点首次启动)
4	Stopped: (节点停止)
5	Operational: (节点处于操作状态)
127	Pre-operational: (节点处于预操作状态)

3.7.1 对象 0x1016: Producer Heartbeat Time(生产者心跳报文时间)

这个对象定义了设备心跳报文发出的时间间隔，将根据“消费者心跳时间间隔”参数（索引 1016h）中所设置的周期和 ID 来监控网络中其余站点产生的心跳报文，一般是主站产生的心跳报文，用来确定主站的状态，该周期以 ms 为单位，如果超过该周期，驱动器将会产生通信系统故障报警。

Bits	31-24	23-16	15-0
Value	reserved (value: 00h)	Node-ID	heartbeat time

对象描述：

Index	0x1016
Name	consumer_heartbeat
Object Code	VAR

目录描述：

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	1
Default	1

Sub-Index	1
Name	consumer_heartbeat
Data Type	UINT32
Category	Mandatory
Access	RW
PDO Mapping	NO

Units	--
Value Range	0~4294967295
Default Value	0

3.7.2 对象 0x1017: Producer Heartbeat Time(生产者心跳报文时间)

这个对象定义了设备心跳报文发出的时间间隔，当该值设为 0 时，不发出心跳报文。

对象描述：

Index	0x1017
Name	Producer Heartbeat Time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	ms
Value Range	0

3.8 life guard 功能

通过使用 life guard 功能，主节点可以检查每个节点当前的状态，当这些节点没有数据传送时尤其有意义。

Master 主节点发送远程帧（无数据）如下：

Master→Slave

COB-ID
0x700+Node_ID（主站 ID）

Slave 从节点发送如下报文应答

Slave→Master

COB-ID	Byte
0x700+Node_ID（从站 ID）	Bit7: toggle; bit6-0: 状态

数据部分包括一个触发位（bit7），触发位必须在每次节点保护应答中交替清“0”或者置“1”。触发位在第一次节点保护请求时设置为“0”。位 0 到位 6（bit0~bit6）表示节点状态，可为下表中数值。

Value	状态
0	Boot-up: (节点首次启动)
4	Stopped: (节点停止)
5	Operational: (节点处于操作状态)
127	Pre-operational: (节点处于预操作状态)

3.8.1 对象 0x100C: Guard Time(保护时间)

这个对象描述了主节点发送远程帧（无数据）的周期值。

Index	0x100C
Name	Guard Time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	ms
Value Range	0

3.8.2 对象 0x100D: Life Time Factor (生命因子)

这个对象描述了从节点应答主节点发送远程帧的最大时间，如果在 Guard Time*Life Time Factor 这段时间内，从节点没有应答，则认为设备掉线。

Index	0x100D
Name	Life Time Factor
Object Code	VAR
Data Type	UINT8
Access	RW
PDO Mapping	NO
Units	——
Value Range	0

4. 参数配置

4.1 保存参数

4.1.1 对象 0x1010: Save_parameters(存储参数)

0x1010 对象是用来处理参数存储命令的,往 0x1010 对象中的 subindex01,subindex02,subindex03 写入“save”的 ASCII 码“65766173h”,驱动器即执行将参数保存到驱动器 Flash 中的命令,执行存储命令后对象参数自动回零。

对象描述:

Index	0x1010
Name	Save parameters
Object Code	VAR

目录描述:

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	3
Default	3

Sub-Index	1
Name	Save_all_parameters
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO
Units	--
Value Range	0 or 65766173h
Default Value	0

Sub-Index	2
Name	Save_com_parameters
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO

Units	--
Value Range	0 or 65766173h
Default Value	0

Sub-Index	3
Name	Save_use_parameters
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO
Units	---
Value Range	0 or 65766173h
Default Value	0

4.2 恢复默认参数

4.2.1 对象 0x1011: Restore_parameters (回复默认参数)

0x1010 对象是用来处理参数存储命令的,往 0x1010 对象中的 subindex01,subindex02,subindex03 写入“load”的 ASCII 码“64616F6Ch”,驱动器即执行将恢复驱动器的默认参数表,执行恢复命令后对象参数自动回零。

对象描述:

Index	0x1011
Name	Restore_parameters
Object Code	VAR

目录描述:

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	3
Default	3

Sub-Index	1
Name	Restore_all_parameters
Data Type	UINT32

Category	Optional
Access	RW
PDO Mapping	NO
Units	--
Value Range	0 or 64616F6Ch
Default Value	0

Sub-Index	2
Name	Restore_com_parameters
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO
Units	--
Value Range	0 or 64616F6Ch
Default Value	0

Sub-Index	3
Name	Restore_use_parameters
Data Type	UINT32
Category	Optional
Access	RW
PDO Mapping	NO
Units	--
Value Range	0 or 64616F6Ch
Default Value	0

5. 设备控制

驱动器有多种状态需要控制器来控制，可以通过修改控制字来实现；驱动器的控制状态可以通过状态字反映，用户可以通过读状态字的各个标志位得知驱动器的控制状态。驱动器的操作可以通过写控制字来完成，同时在 MOTEC 驱动器中，我们还预留了 MOTEC 专用通道，用于对驱动器和电机的操作和控制。

索引	名称	备注
0x6040	controlword	控制字
0x6041	statusword	状态字

5.1 机器状态

5.1.1 简述

在 CANopen 中 DS402 规定中，驱动器的所有动作和状态都通过状态机映射。设备状态可通过控制字控制，而设备的当前状态则由状态字表达。

5.1.2 驱动器的状态及状态转换图

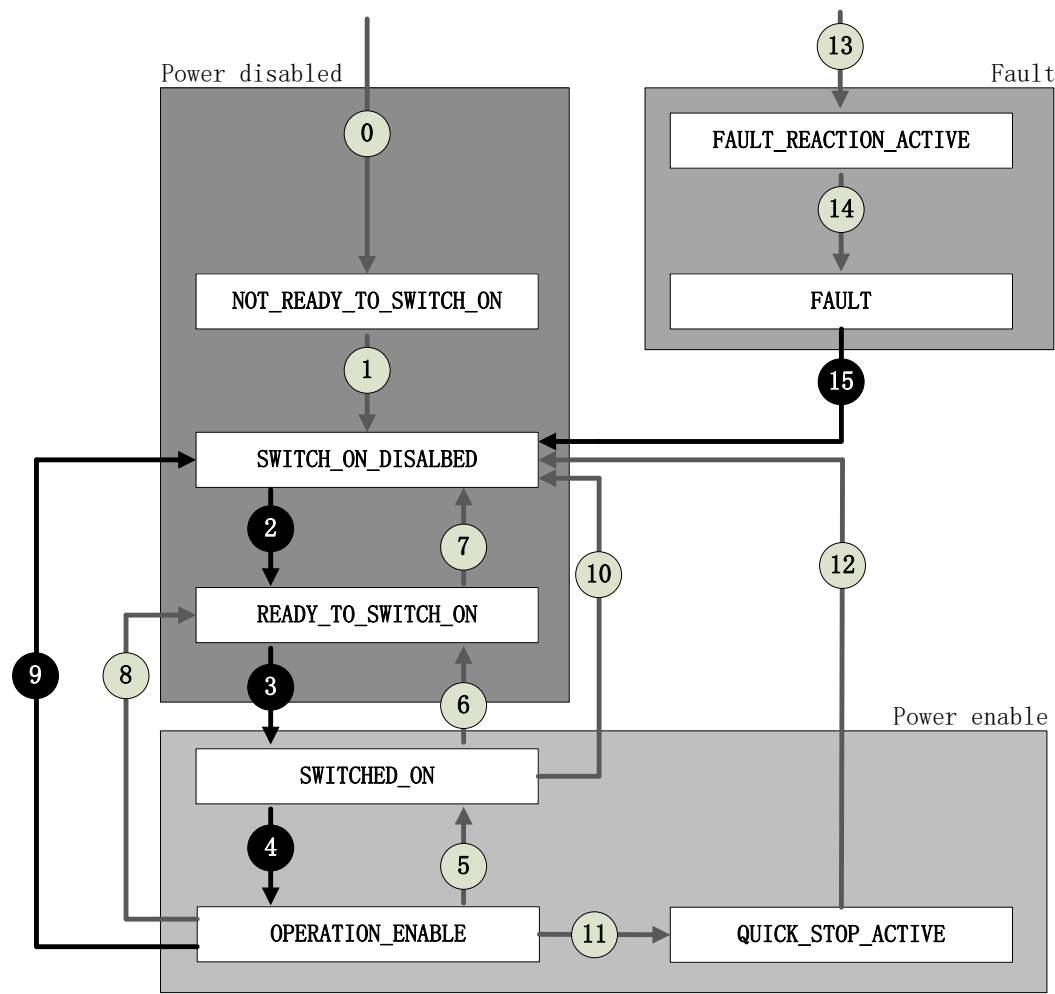


图 5.1 驱动器的状态及状态转换图

图 5.1 中，可以看到状态被划分为三部分，”Power Disable”（伺服关闭）、“Power enable”（伺服允许）、“Fault”（故障）。其中各个状态之间的转换，由主机来控制。

5.1.2.1 状态描述

以下是功能的描述：

■ NOT READY TO SWITCH ON

当驱动器控制电源已经上电，驱动器就已经初始化了，功能会被禁止。驱动器在这个状态之下，只有通讯是可以使用，用户可以一直监控状态字；

■ SWITCH ON DISABLED

驱动已经初始化，驱动器参数已经建立好，驱动器参数可能已经被修改，伺服使能禁止；

■ READY TO SWITCH ON

高压可能已经建立，参数可能已经被改变，驱动器功能依然处于禁止状态；

■ SWITCHED ON

伺服已经使能，但控制功能依然被禁止；

■ OPERATION ENABLE

在没有故障发生情况下，伺服使能，所有功能全部允许；

■ QUICK STOP ACTIVE

快速停止功能被执行，使用怎样的停止方式，要看相应的快速停止选项设置；

■ FAULT REACTION ACTIVE

驱动器发生了一个故障，一个故障反应功能正在被执行，伺服使能被禁止；

■ FAULT

驱动器发生了一个故障，，伺服使能被禁止。

5.1.2.2 状态转换

参考下面对 Controlword 的说明，其中由位 0，1，2，3，7 控制状态转换。

5.2 Controlword(控制字)

5.2.1 控制字简述

主机通过设置驱动器的 Controlword 来切换驱动器的状态。

5.2.2 对象 0x6040: Controlword(控制字)

这个对象定义了步进驱动器的控制字，控制字控制的包括有：

- 改变控制状态
- 控制操作模式
- 厂家指定的某些选项

对象描述：

Index	0x6040
Name	控制字

Object Code	VAR
Data Type	UINT16
Category	Mandatory
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	No

值说明:

15-11	10-9	8	7	6-4	3	2	1	0
Manufacturer Specific(厂家指定)	保留	Halt (命令停止)	Fault Reset (故障恢复)	Operation mode specific(操作模式指定)	Enable Operation(允许操作)	Quick Stop (快速停止)	Enable Voltage (允许通电)	Switch on(伺服使能)

(1) 位 0-3 和位 7

下表是产生不同的控制指令的位组合。

命令	控制字的位					转换
	7	3	2	1	0	
	故障复位	允许操作	快速停止	允许通电	伺服使能	
(Shutdown) 伺服禁止	0	X	1	1	0	2, 6, 8
(Switch ON) 伺服使能	0	0	1	1	1	3*
(Switch ON) 伺服使能	0	1	1	1	1	3**
(Disable Voltage) 禁止通电	0	X	X	0	X	7, 9, 10, 12
(Quick Stop) 快速停止	0	X	0	1	X	7, 10, 11
(Disable Operation) 禁止操作	0	0	1	1	1	5
(Enable Operation) 允许操作	0	1	1	1	1	4, 16
(Fault Reset) 故障复位	上升沿	X	X	X	X	15

设备控制命令触发:

- 标志了 X 的位表示不用理会;
- 有*标志的转换, 表示会执行伺服使能;
- 有**标志的跟有*标志的执行一样的动作。

(2) 位 4, 5, 6 和 8

这些控制位根据不同的控制模式, 有不同的功能, 下面是位的描述:

BIT	操作模式
-----	------

	轮廓位置模式	轮廓速度模式	轮廓力矩模式	
4	(New set-point) 设置新位置点	保留	保留	
5	保留	保留	保留	
6	(Abs/Rel) 绝对/相对	保留	保留	
8	(Halt) 停止	(Halt) 停止	(Halt) 停止	

(3) 位 9 和位 10

这两个位保留作将来用，在一般情况下设为零。

(4) 位 11, 12, 13, 14 和 15

这些位由厂商定义。

5.3 Statusword(状态字)

5.3.1 状态字简述

状态字是反映驱动器当前状态的一个重要对象,用户可以通过查询状态字各个标志位的状态确定当前驱动器和电机的状态。

5.3.2 对象 0x6041: Statusword(状态字)

状态字指示了驱动器当前的状态包括有:

- 驱动器功能状态
- 各种模式下的操作状态
- 厂家指定的其他选项

对象描述:

Index	0x6041
Name	状态字
Object Code	VAR
Data Type	UINT16
Category	Mandatory
Access	RO
PDO Mapping	YES
Units	--
Value Range	--
Default Value	0

值描述:

位	说明
0	Ready to switch on (准备锁轴)
1	Switched on (锁轴)
2	Operation enabled (操作允许)
3	Fault (故障)
4	Voltage enabled (通电允许)
5	Quick stop (快速停止)
6	Switch on disabled (禁止锁轴)
7	Warning (警告)

8	Manufacturer specific (厂家指定)
9	Remote (遥控)
10	Target reached (目标到达)
11	Internal limit active (内部限制生效)
12-13	Operation mode specific (操作模式指定)
14-15	Manufacturer specific (厂家指定)

(1) 位 0-3, 5 和 6

下表描述了位所指示的设备状态

值 (二进制)	状态
x0xx 0000	Not ready to switch on (没有准备锁轴)
x1xx 0000	Switch on disabled (禁止锁轴)
x01x 0001	Ready to switch on (准备锁轴)
x01x 0011	Switch on (锁轴)
x01x 0111	Operation enabled (操作允许)
x00x 0111	Quick stop active (快速停止激活)
x0xx 1111	Fault reaction active (故障反应激活)
x0xx 1000	Fault (故障)

(2) 位 4: 上电允许

当这个位置 1, 表示驱动器已经被通上高压。

(3) 位 5: 快速停止

快速停止控制完成后, 该位被清零。

(4) 位 7: 警告

一个驱动器警告发生时, 位 7 就会置 1, 直到报警被清除。当驱动器收到一个非法控制字, 同样会被置 1。

(5) 位 8: 保留

(6) 位 9: 遥控(不支持)

(7) 位 10: 目标到达

根据不同的模式有不同的含义:

轮廓位置模式: 当实际位置到达设定的位置窗中, 该位就会置 1, 直到有新的位置点设置才清零或位置实际值超出位置窗口。

轮廓速度模式: 当实际速度到达设定的速度窗口时, 该位就会置 1, 直到速度设定值被改变才清零或实际速度值超出速度窗口。

轮廓电流模式: 当实际电流到达设定的电流窗口时, 该位就会置 1, 直到电流设定值被改变才清零或实际电流值超出电流窗口。

(8) 位 11: 限位触发标志

当限位(无论是软件限位还是硬件开关限位, 或者是高限位和低限位)被触发时, 该位就会被置 1。

(9) 位 12:

根据不同的模式有不同的含义；

- 轮廓位置模式：设置新点回应。当新点设置命令被接受后，该位被置 1，直到控制字的（新位置点设置）置零，该位才被清零。
- 轮廓速度模式：当电机实际速度小于零速阈值时，该位置 1，否则该位置零。

(10) 位 13：错误标志

根据不同的模式有不同的含义；

- 轮廓位置模式：位置跟踪超差标志位。当位置跟踪误差超过位置误差阈值而且超过的时间大于设定的超时值时，该位置 1。
- 轮廓速度模式：速度跟踪超差标志位。当速度超过速度跟踪误差阈值而且超过的时间大于设定的超时值时，该位置 1。

6. 运动控制

6.1 描述

MOTEC 智能步进驱动器在 CANopen 模式下可以工作在位置、速度和回零模式, 分别如下:

- Profile position mode (轮廓位置模式)
- Profile Velocity mode (轮廓速度模式)
- Homming mode (回零模式)

6.1.1 功能描述

驱动器的动作取决于所选用的操作模式。因此, 用户可以根据自己的需要选择哪种操作模式。驱动器默认的操作模式为位置模式 (参数值为 1), 可以选择任意的模式修改, 速度控制模式 (参数值为 3)、电流控制模式 (参数为 4)。

6.1.2 对象描述

6.1.2.1 对象 0x6060: Modes of operation(操作模式)

0x6060 对象决定步进驱动器所使用的控制模式。

Index	0x6060
Name	操作模式
Object Code	VAR
Data Type	INT16
Category	Mandatory
Access	RW
PDO Mapping	YES
Units	--
Value Range	1、3、4
Default Value	1

值描述:

Value	Description
-128.....-0	Reserved
1	Profile position mode
2	Reserved
3	Profiled velocity mode
4	Reserved
5	Reserved
6	Homming mode
7...127	Reserved

数值	描述
1	Profile position mode(轮廓位置模式)
3	Profile velocity mode (轮廓速度模式)
6	Homming mode (回零模式)

6.1.2.2 对象 0x6061: Modes of operation display(操作模式显示)

这个对象反映了现在的控制模式。

对象描述:

Index	0x6061
Name	操作模式显示
Object Code	VAR
Data Type	INT16
Category	Mandatory
Access	RO
PDO Mapping	YES
Units	--
Value Range	1、3、6
Default Value	1

值描述:

值	描述
1	Profile position mode(轮廓位置模式)
3	Profile velocity mode (轮廓速度模式)
6	Homming mode (回零模式)

6.2 轮廓位置控制模式

本节主要介绍了驱动器工作在位置控制模式时的相关参数的对象字典。

相关的对象字典有:

索引	名称	描述
0x6062	Position demand value	位置设定值
0x6063	Position actual value	位置实际值
0x6067	Position window	位置到达窗口
0x6068	Position window time	位置到达窗口时间
0x607a	Target position	目标位置
0x607d	Software position limit	位置软件限位
0x607f	Max profile velocity	轨迹规划最大速度值 (S/T 曲线)

0x6082	T-curve acceleration	T 曲线加速度值
0x6083	T-curve deceleration	T 曲线减速度值
0x6084	T-curve max reverse velocity	T 曲线最大反转速度值
0x6069	Halt deceleration	停止运动减速度

6.2.1 综述

在轮廓位置控制模式下，MOTEC 智能步进驱动器提供了 T 曲线轨迹规划用于满足使用者对不同场合位置控制的需求。

在位置控制模式下，如果需要在电机运动的过程中不断更新位置的设定值，可以使用 T 曲线模式下的连续运动模式实现相应的功能。在 T 曲线运动模式下，不但运动位置和目的位置可以随意变动，轨迹参数及最大速度和最大加（减）速度也可以随时改变。

6.2.2 对象描述

6.2.2.1 对象 0x6062: Position demand value(位置设定值)

0x6062 对象是位置设定值，单位为脉冲，位置控制模式下由轨迹规划所得的位置设定值。

对象描述：

Index	0x6062
Name	位置设定值
Object Code	VAR
Data Type	INT32
Category	Optional
Access	R0
PDO Mapping	YES
Units	Pulse
Value Range	-2147483648~2147483647
Default Value	0

6.2.2.2 对象 0x6063: Position actual value(位置实际值)

0x6063 对象定义了实时更新的电机位置实际值。

对象描述：

Index	0x6063
Name	位置实际值
Object Code	VAR
Data Type	INT32
Category	Optional
Access	R0
PDO Mapping	YES

Units	Pulse
Value Range	-2147483648~2147483647
Default Value	0

6.2.2.3 对象 0x6067: Position windows(位置窗口)

0x6067 对象定义了位置到达窗口，用来判断位置是否到达。位置到达窗口为位置设定值左右设定的区域，如果位置设定值为 P_{set} ，窗口值设为 P_{win} ，则位置窗口为 $[P_{set}-P_{win}, P_{set}+P_{win}]$ 。当实际位置处于位置窗口内时，位置到达标志位置位。

对象描述：

Index	0x6067
Name	位置到达窗口
Object Code	VAR
Data Type	UINT16
Category	Optional
Access	RW
PDO Mapping	NO
Units	Pulse
Value Range	0~65535
Default Value	1000

6.2.2.4 对象 0x6068: Position Window Time(位置窗口时间)

0x6068 对象定义了位置窗口时间值，当实际位置到达了位置窗口，并且持续了“位置窗口时间”设定的时间，位置到达标志置位，表示目标位置到达。

对象描述：

Index	0x6068
Name	位置窗口时间
Object Code	VAR
Data Type	UINT16
Category	Optional
Access	RW
PDO Mapping	NO
Units	ms
Value Range	0~65535
Default Value	1000

6.2.2.5 对象 0x607A: Profile target position(轮廓目标位置)

0x607A 对象定义了轮廓目标位置值，当控制字 ABS/REL 位(bit6)设置为 0 时，该值为相对位置值；当 ABS/REL 位设置为 1 时，该值为绝对位置值。而当使用 MOTEC 专用通道实施驱动器控制时，需要使用对象 0x2006 来设置运动方式是相对运动还是绝对运动。

对象描述:

Index	0x607A
Name	轮廓目标位置
Object Code	VAR
Data Type	INT32
Category	Mandatory
Access	RW
PDO Mapping	YES
Units	Pulse
Value Range	-2147483648~2147483647
Default Value	0

6.2.2.6 对象 0x607D: Software position limit(软件位置限制)

0x607D 对象定义了位置控制中为了防止电机位置超出安全范围，所设置的软件位置限制参数。软件位置限制设置并生效以后，电机只能运动在此范围之内。

对象描述:

Index	0x607D
Name	软件位置限制
Object Code	ARRAY
Data Type	INT32
Category	Mandatory

目录描述:

Sub-Index	0
Description	Number of entries (条目数量)
Entry category	Mandatory
Access	RO
PDO Mapping	NO
Units	--
Value Range	2
Default	2

Sub-Index	1
Description	Min Position Limit (最小位置限制)
Entry category	Mandatory
Access	RW
Data Type	INT32
PDO Mapping	NO

Units	Pulse
Value Range	-2147483648~2147483647
Default	--

Sub-Index	2
Description	Max Position Limit (最大位置限制)
Entry category	Mandatory
Access	RW
Data Type	INT32
PDO Mapping	NO
Units	Pulse
Value Range	-2147483648~2147483647
Default	--

6.2.2.7 对象 0x607F: Max profile velocity (轨迹规划最大速度值)

0x607F 对象定义了位置控制的时候，S/T 曲线轨迹规划中速度曲线的最大速度值。

对象描述：

Index	0x607F
Name	轨迹规划最大速度值
Object Code	VAR
Data Type	UINT16
Category	Optional
Access	RW
PDO Mapping	YES
Units	RPM
Value Range	0~5000
Default Value	3000

6.2.2.8 对象 0x6082: T-curve acceleration (T 曲线轨迹规划加速度)

0x6082 对象定义了 T 曲线轨迹规划中的加速度值。

对象描述：

Index	0x6082
Name	T 曲线轨迹规划加速度
Object Code	VAR
Data Type	UINT16
Access	RW

PDO Mapping	NO
Units	RPS2
Value Range	1~1000
Default Value	100

6.2.2.9 对象 0x6083: T-curve deceleration (T 曲线轨迹规划减速度)

0x6083 对象定义了 T 曲线轨迹规划中的减速度值。

对象描述:

Index	0x6083
Name	T 曲线轨迹规划减速度
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	RPS2
Value Range	1~1000
Default Value	100

6.2.2.10 对象 0x6084: T-curve max reverse velocity(T 曲线模式最大反转速度)

0x6084 对象定义了 T 曲线轨迹规划模式下, 当电机的运动目标位置从一个方向改变到另一个方向时, 电机需要反转, 而 0x6084 即定义了电机反转时的最大速度。这个值越大, 电机需要改变运动方向时反应越灵敏, 但改变方向时电机振动大。值越小, 反应越慢但是电机改变方向时运动更平稳。

对象描述:

Index	0x6084
Name	T 曲线模式下最大反转速度
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	RPM
Value Range	1~1000
Default Value	100

6.2.2.11 对象 0x6069: Halt deceleration(停止运动减速度值)

0x6069 这个对象定义了电机在停止运动过程中的减速度

对象描述:

Index	0x6069
Name	停止运动减速度值
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	RPS2
Value Range	1~1000
Default Value	100

6.2.3 位置运动控制举例

6.2.3.1 单段位置运动

在轮廓位置控制模式下，MOTEC 步进驱动器提供了 T 曲线轨迹规划用于满足使用者对不同场合位置控制的需求。单段位置运动即及起始速度和终止速度都为零的运动。

单段位置运动操作过程如下。首先控制器把位置控制数据（Profile target position, Max profile velocity, profile acceleration, profile deceleration）发送到步进驱动器，这样就算完成了一次位置设置，然后将 Controlword 的 new set point 位置 1，当新位置数据被驱动器接受后 statusword 的 set point acknowledge(bit12)会置 1，接着，主控制器需要对 Controlword 的 new set point 位清 0，驱动器也同时对 statusword 的 set point acknowledge 位清 0，在 Controlword 的 new set point 位置 1 后，电机就会运行。图 6.1 描绘了 T 曲线点到点运动的速度轨迹。



图 6.1 单段位置控制 T 曲线速度轨迹

6.2.3.2 多段位置运动组合

在位置控制模式下，如果需要在电机运动的过程中不断更新位置的设定值，可以使用 T 曲线模式下的连续运动模式实现相应的功能。在 T 曲线运动模式下，不但运动位置和目的位置可以随意变动，轨迹参数及最大速度和最大加（减）速度也可以随时改变。

MOTEC 步进驱动器的 T 曲线轨迹规划既支持起点和终点速度为 0 的运动,也支持连续更新位置设定值的连续运动模式。更新位置设定值的时间间隔除了受通讯时间影响之外,没有时间间隔的限制。除了在运动过程中能更新新的目标位置之外,也能实时更新 T 曲线轨迹的最大速度以及加速度和减速度。

多段 T 曲线连续运动的操作方式如下。首先控制器把位置控制数据 (Profile target position, Max profile velocity, T curve acceleration, T curve deceleration, T curve max reverse velocity).接着将 Controlword 的 new set point 位置 1, 当新位置数据被驱动器接受后 statusword 的 set point acknowledge 会置 1, 电机开始执行动作, 主控制器需要对 Controlword 的 new set point 位清 0, 并写入第二段位置控制数据, 驱动器也同时对 statusword 的 set point acknowledge 位清 0, 仅跟着主控制器对 Controlword 的 new set point 位置 1, 新位置数据被驱动器接受后 statusword 的 set point acknowledge 会置 1, 同时启动第二段位置运动。图 6.2 描绘了多段位置控制时的情况。

多段位置控制模式也可以分为绝对运动和相对运动,如果是在相对运动模式下,设置新的运动距离时,原来未完成运动距离将不计入新的运动距离。而在绝对运动模式下,电机始终以最新的位置为最终位置。

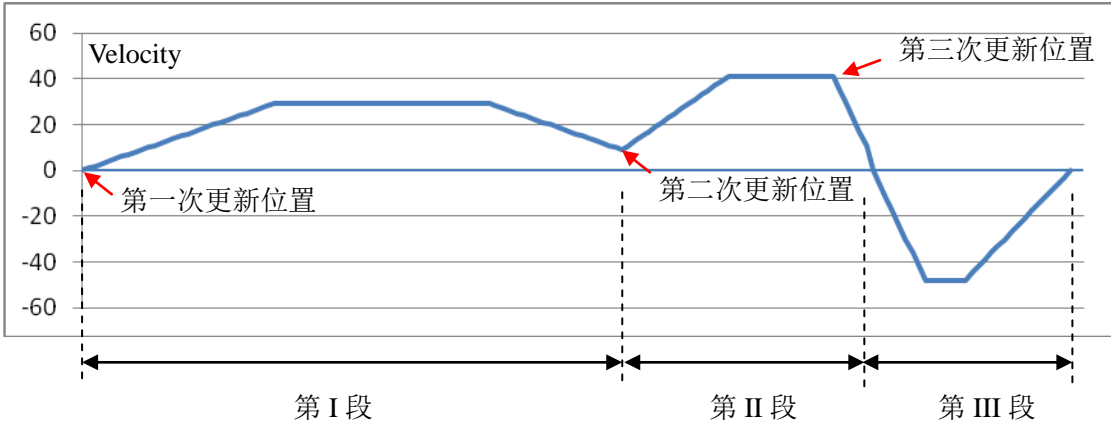


图 6.2 T 曲线连续运动速度曲线

6.3 轮廓速度运动模式

在轮廓速度运动模式下,MOTEC 智能步进驱动器在速度控制模式下采用 T 曲线的加减速控制模式。下面章节介绍与速度运动控制相关参数的对象字典。

主要有:

索引	名称	备注
0x606B	Velocity demand value	速度设定值
0x606C	Velocity actual value	速度实际值
0x606F	Velocity threshold	零速度阈值
0x60FF	Target velocity	目标速度值

6.3.1 综述

轮廓速度模式由用户设定的目标速度、加速度、减速度来进行的速度曲线控制。

6.3.2 对象描述

6.3.2.1 对象 0x606B: Velocity demand value(速度给定值)

0x606B 对象定义了速度给定值，当目标速度值改变后，速度给定值由当前实际速度、目标速度和加减速速度值根据 T 曲线轨迹规划算法得到。

对象描述：

Index	0x606B
Name	速度给定值
Object Code	VAR
Data Type	INT32
Category	Mandatory
Access	R0
PDO Mapping	YES
Units	RPM
Value Range	-32768~32767
Default Value	0

6.3.2.2 对象 0x606C: Velocity actual value(速度实际值)

0x606C 对象定义了电机在运动过程中的速度实际值。

对象描述：

Index	0x606C
Name	速度实际值
Object Code	VAR
Data Type	INT32
Category	Mandatory
Access	R0
PDO Mapping	YES
Units	RPM
Value Range	-32768~32767
Default Value	0

6.3.2.6 对象 0x60FF: Target velocity (目标速度)

0x60FF 对象定义了用户设定的目标速度值。

对象描述：

Index	0x60FF
-------	--------

Name	目标速度
Object Code	VAR
Data Type	INT32
Category	Mandatory
Access	RW
PDO Mapping	YES
Units	RPM
Value Range	-32768~32767
Default Value	NO

6.4 回原点操作

本节主要介绍了驱动器回原点操作时的动作。

6.4.1 综述

回原点操作是驱动在指定的位置范围内，以设定的速度、加速度和寻找原点的方法寻找工作零点。MOTEC 智能步进驱动器没有设置专门的回原点操作方式。

MOTEC 智能步进驱动器提供用于寻找工作零点的找原点操作，找原点操作可以是由网络指令触发，也可以通过 I/O 的状态来触发完成。MOTEC 步进驱动器的找原点过程包括两个个部分，分别是：1) 找原点开关；2) 脱离运动。通过设置参数表中的有关找原点操作的相关参数，这三个部分的功能可以相互组合从而形成适合用户使用的回原点方式。当然，根据用户的实际需求，可以只选择这两个个步骤中的两个或一个步骤来完成回原点操作。

需要说明的是找原点开关运动速度 16 位有符号数，其速度表示了电机的运动速度，而符号表示了运动方向。符号为正表示正向运动，符号位负表示负向运动。而第二阶段的脱离距离是有符号数，同时表示运动距离和运动方向。而停留稳定时间是一个阶段到下一个阶段的停留时间。

回原点操作的相关参数需要通过设置驱动器内部的参数表，具体详见驱动器的使用手册。

7. MOTEC 专用通道

以上章节的内容说明了如何使用 SDO 和 PDO 来修改驱动器参数或对驱动器进行操作。除了使用 CANOPEN 标准协议对电机进行控制之外。为了操作方便，MOTEC 步进驱动器还提供了 CANOPEN 简易操作模式，即 MOTEC 专用通道。MOTEC 专用通道只需使用 SDO 方式修改特定的寄存器对象就可以到达对电机和驱动器进行操作的目的。下表描述了 MOTEC 驱动器的一些特殊功能通道的对象字典。

7.1 对象 0x2000: OPERATING_MODE (驱动器操作模式)

0x2000 对象定义了驱动器的操作模式。

对象描述：

Index	0x2000
Name	驱动器操作模式
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~3
Default Value	0
Description	0- 网络操作模式； 1- 脉冲/方向操作模式； 2- 模拟量操作模式

7.2 对象 0x2001: HALF_CURRENT_VALID (半流功能相关参数)

0x2001 索引是用来设置驱动器的半流功能，Sub-Index1 是设置半流是否启用，Sub-Index2 是设置半流时降低电流为设定电流的百分比，Sub-Index3 是设置半流延时。

对象描述：

Index	0x2001
Name	HALF_CURRENT_VALID
Object Code	VAR

目录描述：

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--

Value Range	3
Default	3

Sub-Index	1
Name	HALF_CURRENT_MODE
Data Type	UINT16
Category	Optional
Access	RW
PDO Mapping	NO
Units	---
Value Range	0 or 1
Default Value	0

Sub-Index	2
Name	HALF_CURRENT_VALUE
Data Type	UINT16
Category	Optional
Access	RW
PDO Mapping	NO
Units	---
Value Range	1 - 99
Default Value	0

Sub-Index	3
Name	CURRENT_DELAY
Data Type	UINT16
Category	Optional
Access	RW
PDO Mapping	NO
Units	---
Value Range	0-65535
Default Value	0

7.3 对象 0x2004: POSITION_RESET_FLAG (当前位置清零)

0x2004 对象定义了驱动器的当前位置清零操作。

对象描述:

Index	0x2004
-------	--------

Name	当前位置清零
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	---
Value Range	0~1
Default Value	0
Description	1-当前位置清零

7.4 对象 0x2005: POWERSUPPLY_VOLTAGE (当前电源电压)

0x2005 对象定义了驱动器的当前电源电压。

对象描述:

Index	0x2005
Name	当前电源电压
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	---
Value Range	0~65535
Default Value	0
Description	0

7.5 对象 0x2007: DIGITAL_IN_STATUS (当前数字输入口状态)

0x2007 对象定义了驱动器的数字输入口的状态。

对象描述:

Index	0x2007
Name	当前数字输入口状态
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	---
Value Range	0~65535
Default Value	0
Description	0

7.6 对象 0x2008: DIGITAL_OUT_STATUS (当前数字输出口状态)

0x2008 对象定义了驱动器的数字输出口的状态。

对象描述:

Index	0x2008
Name	当前数字输出口状态
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	0
Description	0

7.7 对象 0x2009: ANINPUT_VALUE (当前模拟量输入口状态)

0x2009 对象定义了驱动器的模拟量输入口的状态。

对象描述:

Index	0x2009
Name	当前模拟量输入口状态
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	0
Description	0

7.8 对象 0x200A: ERROR_CODE1 (驱动器报警信息)

0x200A 对象定义了驱动器的报警信息。

对象描述:

Index	0x200A
Name	驱动器报警信息
Object Code	VAR
Data Type	UINT16
Access	RW

PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	0
Description	0

7.9 对象 0x200C: CONTROL_MODE (控制模式)

0x200C 对象定义了驱动器的控制模式。

对象描述:

Index	0x200C
Name	控制模式
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	0- 位置模式 1- 速度模式

7.10 对象 0x200D: MOTION_MODE (位置控制模式)

0x200D 对象定义了驱动器的位置控制模式。

对象描述:

Index	0x200D
Name	位置控制模式
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	0- 绝对位置控制 1- 相对位置控制

7.11 对象 0x200E: STARTMOTION_FLAG (位置控制模式启动)

0x200E 对象定义了驱动器的位置控制模式启动指令。

对象描述:

Index	0x200E
Name	位置控制模式启动指令
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	---
Value Range	0~1
Default Value	0
Description	1-位置控制启动

7.12 对象 0x200F: MOTION_DONE_FLAG (位置模式定位完成)

0x200F 对象定义了驱动器的位置模式定位完成。

对象描述:

Index	0x200F
Name	位置模式定位完成
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	---
Value Range	0~1
Default Value	0
Description	0-运行中 1-完成

7.13 对象 0x2010: STATUS_FLAG (电机运动标志)

0x2010 对象定义了驱动器的电机运动标志。

对象描述:

Index	0x2010
Name	电机运动标志
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES

Units	--
Value Range	0~1
Default Value	0
Description	0-电机运行 1-电机停止

7.14 对象 0x2011: JOG_POSITIVE (正向点动)

0x2011 对象定义了驱动器的正向点动功能。

对象描述:

Index	0x2011
Name	正向点动
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	0-停止正向点动 1-正向点动启动

7.15 对象 0x2012: JOG_NEGATIVE (负向点动)

0x2012 对象定义了驱动器的负向点动功能。

对象描述:

Index	0x2012
Name	负向点动
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	0-停止负向点动 1-负向点动启动

7.16 对象 0x2013: HOMMING_START (启动回零)

0x2013 对象定义了驱动器的启动回零功能。

对象描述:

Index	0x2013
Name	启动回零
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	1-启动回零

7.17 对象 0x2014: HOMMING_STOP (停止回零)

0x2014 对象定义了驱动器的停止回零功能。

对象描述:

Index	0x2014
Name	停止回零
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	1-停止回零

7.18 对象 0x2015: QUICK_SOFT_STOP (急停)

0x2015 对象定义了驱动器的急停功能。

对象描述:

Index	0x2015
Name	急停
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES

Units	--
Value Range	0~1
Default Value	0
Description	1-急停

7.19 对象 0x2019: encoder_value (编码器计数)

如果启用了编码器计数功能，0x2019 对象定义了驱动器的编码器计数数值。

对象描述:

Index	0x2019
Name	编码器计数数值
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	--
Value Range	-2147483648~2147483647
Default Value	0
Description	0

7.20 对象 0x201A: ENABLE_FLAG (使能/释放)

0x201A 对象定义了驱动器的使能/释放功能。

对象描述:

Index	0x201A
Name	使能/释放
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~1
Default Value	0
Description	0-释放 1-使能

7.21 对象 0x2021 - 0x2038: CANopen_OBJ_RESERVED (保留对象字典)

0x2021 - 0x2038 对象定义了一些保留的对象字典索引，全部为 16 位参数表，对应着驱动器参数表中的 Parameter301 - Parameter324。用户可以使用 CANopen 直接更改驱动器

的参数表，用来与驱动器内置 PLC 进行数据交互。

对象描述：

Index	0x2021 - 0x2038
Name	保留对象字典
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	0
Description	0

8. 简化 DS402 状态机控制方式

标准的 CANopen 驱动器类设备中, 使用 DS301 作为传输协议, 使用 DS402 作为步进运动控制协议。区别于标准的伺服, 步进系统中, 由于没有编码器反馈作为闭环控制, 并且 DS402 的复杂控制字和状态字的状态机对于简单的步进应用就显得有些繁琐, 使用起来也有诸多的不便。基于此种需要, MOTEC 的智能步进驱动器, 除了集成了标准的 DS301 和 DS402 协议之外, 基于 DS402, 自主设定了一套简单的控制字和状态字, 可以实现对于电机的各种操作。启用此模式, 需要对于驱动器的参数做设置。

标识符	参数号	参数意义	参数范围
MOTEC_OPERATIONAL_CANOPEN	Parameter[16]	CANopen 中 MOTEC 快速通道选择	0-不启用 MOTEC 快速通道; 1-启用 MOTEC 快速通道

8.1 关于 MOTEC 自定义的控制字状态字等的介绍

8.1.1 简化 DS402 状态机控制字

控制字是一个 16 位无符号数, 在对象字典里面的索引和子索引为

索引	子索引	对象	Name	标识	长度
0x201B	0x00	控制字	MOTEC_CONTROL_WORD	CW	16 位

控制字分为两个部分, 低八位为运动控制部分

控制字低 8 位 (运动控制)							
Bit7	bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1-清除当前位置, 清除完成之后, 会自动清零	1-清除当前位置, 清除完成之后, 会自动清零	1-碰到内部负限位开关, 急停	1-碰到内部正限位开关, 急停	非点动模式下无效	非点动模式下无效	1-电机启动; 0-电机停止	1-使能; 0-释放
				1-启动负向点动	1-启动正向点动	点动模式下无效	
				0-停止负向点动	0-停止正向点动		

高八位为 IO 控制和运动模式控制

bit15	bit14	bit13	bit12	Bit11~bit8	
保留	1-OUTPUT1 导通; 0-OUTPUT1 关闭	1-OUTPUT2 导通; 0-OUTPUT2 关闭	1-OUTPUT1 导通; 0-OUTPUT1 关闭	0	绝对位置模式
				1	相对位置模式
				2	速度模式
				3	回零模式
				4	点动模式

对于控制字的说明。

控制字高八位中:

- 控制字的高八位中的低四位 bit8-bit11 组成了一个 4 位的无符号数, 可以设置 0-7, 其中 0-4 分别代表设定的控制模式;
- 控制字的高八位中的高四位中, bit12 用来操作 OUTPUT1 操作模式, bit13 用来操作 OUTPUT2 操作模式, bit14 用来操作 OUTPUT3 操作模式。

控制字低 8 位中

- BIT0 为使能位, BIT0 = 1, 同时使能开关触发, 电机使能;

- b、BIT1 为启动/停止位，在不同的模式下，有不同的操作方式，请查看 8.1.2 简化 DS402 状态机章节；
- c、BIT2 为正向点动控制位，只有在点动模式下，该位状态才有效；
- d、BIT3 与 BIT2 相同，为负向点动控制位；
- e、BIT4 为内部正限位开关，在运动过程中该位置 1 时，电机正向急停；
- f、BIT5 与 BIT4 相同，为内部负限位开关；
- g、BIT6 为位置清零开关，该位置 1 时候，会将当前电机的内部计数和编码器计数同时清零，并且在清零结束以后该位自动置 0；
- h、BIT7 为清除报警位，该位置 1 时候，会清除当前报警，在报警清除之后该位自动置 0。

8.1.1.1 对象 0x201B: MOTEC_CONTROL_WORD (MOTEC 控制字)

0x201B 对象定义了驱动器的 MOTEC 控制字。

对象描述：

Index	0x201B
Name	MOTEC 控制字
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	0
Description	0

8.1.2 简化 DS402 状态机状态字

状态字是一个 16 位无符号数，在对象字典里面的索引和子索引为

索引	子索引	对象	Name	标识	长度
0x201C	0x00	状态字	MOTEC_CONTROL_WORD	CW	16 位

状态字同样分为两个部分，低八位为运动控制部分

状态字低 8 位（驱动器状态位）							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
当前位置与编码器偏差过大	找到原点	定位完成	速度为 0	速度到达	有故障	启动	使能

高八位为数字输入口的状态

状态字高 8 位（输入口状态）							
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
1-INPUT 8 导通； 0-INPUT 8 关闭	1-INPUT 7 导通； 0-INPUT 7 关闭	1-INPUT 6 导通； 0-INPUT 6 关闭	1-INPUT 5 导通； 0-INPUT 5 关闭	1-INPUT 4 导通； 0-INPUT 4 关闭	1-INPUT 3 导通； 0-INPUT 3 关闭	1-INPUT 2 导通； 0-INPUT 2 关闭	1-INPUT 1 导通； 0-INPUT 1 关闭

对于状态字的说明：

状态字低八位用来表征当前的运动状态和电机状态：

- a、BIT0 表示当前电机是否使能，只有在控制字使能位和外部使能输入开关同时有效，该位才会被置 1，否则为 0；
- b、BIT1 表示当前运动是否是启动状态；
- c、BIT2 表示当前驱动器是否有故障，有故障的情况下，BIT2 = 1；
- d、BIT3 表示当前速度是否到达设定的最大速度
- e、BIT4 表示当前速度是否为 0
- f、BIT5 表示当前的位置控制是否定位完成；
- g、BIT6 表示是否找到原点，本标志位上电以后为 0，找到原点之后置 1，开始找原点的时候同时会被置 0；
- h、BIT7 表示在电机停止状态下，进入零速窗口下当前的电机内部值与编码器值误差是否大于最大允许误差。

状态字高八位用来表示当前的数字输入口的状态

- a、BIT8 表示 INPUT1 的状态，0 为 INPUT1 关闭，1 为 INPUT1 导通；
- b、BIT9 表示 INPUT2 的状态，0 为 INPUT2 关闭，1 为 INPUT2 导通；
- c、BIT10 表示 INPUT3 的状态，0 为 INPUT3 关闭，1 为 INPUT3 导通；
- d、BIT11 表示 INPUT4 的状态，0 为 INPUT4 关闭，1 为 INPUT4 导通；
- e、BIT12 表示 INPUT5 的状态，0 为 INPUT5 关闭，1 为 INPUT5 导通；
- f、BIT13 表示 INPUT6 的状态，0 为 INPUT6 关闭，1 为 INPUT6 导通；
- g、BIT14 表示 INPUT7 的状态，0 为 INPUT7 关闭，1 为 INPUT7 导通；
- h、BIT15 表示 INPUT8 的状态，0 为 INPUT8 关闭，1 为 INPUT8 导通；

8.1.2.1 对象 0x201C: MOTEC_STATUS_WORD (MOTEC 状态字)

0x201C 对象定义了驱动器的 MOTEC 状态字。

对象描述：

Index	0x201C
Name	MOTEC 状态字
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	0~65535
Default Value	0
Description	0

8.1.3 简化 DS402 状态机位置字，速度字

速度字和位置字分为设定速度，设定位置和实际速度，实际位置，在对象字典列表中如下

索引	子索引	对象	Name	标识	长度
0x201D	0x00	速度字	MOTEC_TARGET_VELOCITY	TV	16 位
0x201E	0x00	位置字	MOTEC_TARGET_POSTTION	TP	32 位
0x201F	0x00	当前速度	MOTEC_ACTUAL_VELOCITY	AV	16 位
0x2020	0x00	当前位置	MOTEC_ACTUAL_POSTTION	AP	32 位

其中，速度字和位置字在不同的模式下，拥有不同的作用，具体如下

操作模式	速度字（16 位有符号数）	位置字（32 位有符号数）
绝对位置模式	点到点运动的最大速度	绝对位置的目标位置
相对位置模式		相对位置的运动距离
速度模式	速度模式下的目标速度	无效
回零模式	无效	
点动模式	无效	

8.1.3.1 对象 0x201D: MOTEC_TARGET_VELOCITY (MOTEC 目标速度)

0x201D 对象定义了驱动器的 MOTEC 目标速度。

对象描述：

Index	0x201D
Name	MOTEC 目标速度
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	YES
Units	--
Value Range	-32768~32767
Default Value	0
Description	0

8.1.3.2 对象 0x201E: MOTEC_TARGET_POSTTION (MOTEC 目标位置)

0x201E 对象定义了驱动器的 MOTEC 目标位置。

对象描述：

Index	0x201D
Name	MOTEC 目标位置
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	--

Value Range	-2147483648~2147483647
Default Value	0
Description	0

8.1.3.3 对象 0x201F: MOTEC_ACTUAL_VELOCITY (MOTEC 当前速度)

0x201F 对象定义了驱动器的 MOTEC 当前速度。

对象描述:

Index	0x201F
Name	MOTEC 当前速度
Object Code	VAR
Data Type	INT16
Access	RW
PDO Mapping	YES
Units	---
Value Range	-32768~32767
Default Value	0
Description	0

8.1.3.4 对象 0x2020: MOTEC_ACTUAL_POSITION (MOTEC 当前位置)

0x2020 对象定义了驱动器的 MOTEC 当前位置。

对象描述:

Index	0x2020
Name	MOTEC 当前位置
Object Code	VAR
Data Type	INT32
Access	RW
PDO Mapping	YES
Units	---
Value Range	-2147483648~2147483647
Default Value	0
Description	0

8.2 简化 DS402 状态机位置字运动控制相关参数和流程描述

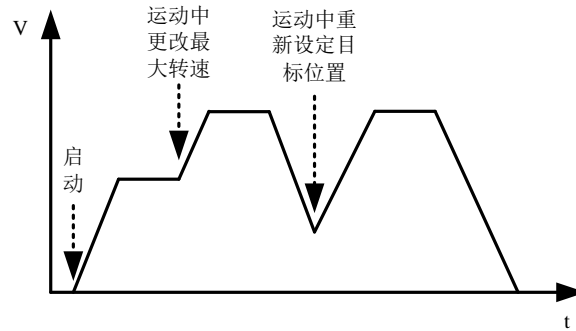
8.2.1 位置控制模式

位置模式下，有两种控制方式

a、绝对位置控制方式

如果要进入绝对位置控制方式，需要将控制字的运动模式设置为 0。

控制字高 8 位的低 4 位设置成 0，即为绝对位置模式，在绝对位置模式下，最大转速和目标位置都可以随时更改，绝对位置模式下启动电机运动需要将控制字第 BIT1 置 1。



绝对位置模式下的连续运动

b、相对位置模式

相对运动模式下控制字高 8 位的低四位设置成 1, 在相对运动模式下, 设定好目标位置以后, 需要将启动位, BIT1 置 1, 才能启动本段相对位置。如果需要启动另外一段相对运动, 则需要先将 BIT1 置 0, 再次置 1 的时候, 才会继续启动一段点到点运动。

8.2.2 速度模式

控制字高 8 位的低 4 位设置成 2, 进入速度模式, 将 BIT1 设置成 1, 则会启动速度运动, 设定的速度来自于速度字。电机运动的方向与设定速度的正负有关, 正数的速度电机正向运动, 负数的速度电机负向运动。

8.2.3 回零模式

回零流程

控制字高 8 位设置成 2, 当前位置处于回零模式, 将 BIT1 设置成 1, 则会启动回零运动, 回零速度为 P116 中设定的回零速度, 运动方向取决于回零速度的正负。

回零操作由几个部分的动作组成, 根据电机当前位置位于零点开关的某个部分或者回零方向, 最多会有如下几个操作:

- 回零触发以后, 电机以设定的回零转速朝设定的方向运动, 回零速度的正负决定电机的运动方向;
- 碰到限位开关或者零位开关以后, 电机停止运动, 并朝相反的方向运动一段回零偏移, 具体由 P119 和 P120 参数决定;
- 回零偏移结束以后, 回零启动位被置为 0, 回零完成位被置为 1, 回零完成。

8.2.4 点动控制

控制字高 8 位设置成 4, 进入点动模式, 点动是否运动取自控制字中的点动控制位, 速度来自于速度字。

9. 联系方式

Website: <http://www.motec365.com.cn>;

地址: 北京市通州区环科中路 17 号 11B;

服务热线: 010-56298855-666;

Email: motecSupport@sina.com;