



## CANopen 应用层和通信协议

应用层和通信协议

(本译文仅供个人技术交流，尊重原文版权方所有权利，禁止用于商业目的)

版本：4.2.0

2011年2月21日

## 历史记录

日期	更改
1994-11-01	发布 1.0 版本作为标准草案建议
1995-01-01	发布 1.1 版本作为标准草案建议
1995-09-22	发布 2.0 版本作为标准草案建议
1996-10-30	发布 3.0 版本作为标准决议草案
1999-06-16	发布 4.0 版本作为标准决议草案
2000-06-01	发布 4.0.1 版本作为标准决议草案
2002-02-13	发布 4.0.2 版本作为标准决议草案
2006-08-15	发布 4.1 版本作为标准草案建议
2007-12-07	发布 4.2 版本作为标准草案建议 <ul style="list-style-type: none"> <li>- 编辑上的更正和澄清</li> <li>- 定义供应商 ID 0000 0000<sub>h</sub></li> <li>- 澄清访问类型的定义</li> <li>- 附加其他允许的 CAN 物理层</li> </ul>
2010-08-11	发表更正 1
2011-02-21	发布 4.2 版本 作为公共规范
2011-02 21	发表更正 2

## 版权和专利许可信息

CAN in AUTOMATION (CIA) 提请注意该 CIA 协议文本有关的某些元素可能涉及专利权利。CiA 不负责确定任何或所有该等专利的权利。

因为该技术规范为免费许可, 保障该协议符合相关法律规定。除非另有书面声明, 版权持有者和/或其它各方提供这一技术规范方不提供任何类型的担保, 不论是明确的或隐含的, 包括但不限于暗示的适销性和适用于某特定用途的保证。全部规范的正确性和完整性取决于用户。即使该标准被证明失败, 也由用户来承担所有必要的维修、修理或更正费用。

## 商标

CANopen®和 CiA®是 CAN in Automation 注册的共同体商标。仅限 CiA 成员或 CANopen 供应商 ID 持有者使用。更多详细的条款可从 CiA 获取。

### ● CiA 2011

保留所有权利。除非另有规定, 禁止将本出版物的任何部分复制或用于任何形式或通过任何电子或机械手段, 包括复印和缩微胶卷, 除非从 CiA 以下地址获得书面许可。

CAN in Automatiene.V.

Kontumazgarten 3

DE-90429 德国纽伦堡

电话: +49-911-928819-0

传真: +49-911-928819-79

Url: [www.can-cia.org](http://www.can-cia.org)

电子邮件: [headquarters@can-cia.org](mailto:headquarters@can-cia.org)

## 目录

1	适用范围.....	9
2	参考资料.....	10
2.1	规范性引用.....	10
2.2	信息参考.....	10
3	缩写和定义.....	10
3.1	缩写.....	10
3.2	定义.....	11
	<b>标准帧</b> .....	11
	<b>扩展帧</b> .....	11
	<b>CAN-ID</b> .....	11
	<b>COB-ID</b> .....	11
	<b>实体</b> .....	11
	<b>FSA</b> .....	11
	<b>设备域</b> .....	11
	<b>逻辑设备</b> .....	11
	<b>Node-ID</b> .....	11
	<b>对象</b> .....	11
	<b>虚拟设备</b> .....	11
4	建模.....	12
4.1	设备域模型.....	12
4.2	通信参考模型.....	13
4.2.1	<b>简述</b> .....	13
4.2.2	<b>CANopen 应用层</b> .....	13
4.2.2.1	<b>概述</b> .....	13
4.2.2.2	<b>服务原语</b> .....	13
4.2.2.3	<b>应用层服务</b> .....	14
4.3	<b>CANopen 设备模型</b> .....	14
4.3.1	<b>简述</b> .....	14
4.4	通信协议分类.....	15
4.4.1	<b>简述</b> .....	15
4.4.2	<b>主/从协议</b> .....	15
4.4.3	<b>客户端/服务器协议</b> .....	16
4.4.4	<b>生产者/消费者协议-推/拉模式</b> .....	16
4.4.5	<b>对象字典</b> .....	17
4.5	网络系统模型.....	17
4.5.1	<b>设备协议</b> .....	17
5	物理层.....	18
5.1	<b>OSI 参考模型</b> .....	18
5.2	<b>介质相关接口</b> .....	18
5.3	<b>物理介质连接</b> .....	18
5.4	<b>物理信令</b> .....	18
6	数据链路层.....	20
6.1	<b>概述</b> .....	20
6.2	<b>CAN 帧类型</b> .....	20
7	应用层.....	21
7.1	<b>数据类型和编码规则</b> .....	21
7.1.1	<b>数据类型和编码规则简述</b> .....	21
7.1.2	<b>数据类型定义</b> .....	21

7.1.3	位序列 .....	22
7.1.3.1	位序列定义.....	22
7.1.3.2	位序列的传输语法.....	22
7.1.4	基本的数据类型。 .....	23
7.1.4.1	概述.....	23
7.1.4.2	NIL.....	23
7.1.4.3	Boolean.....	23
7.1.4.4	Void.....	23
7.1.4.5	Unsigned Interger.....	23
7.1.4.6	Signed Integer.....	24
7.1.4.7	Floating-Point 数 .....	25
7.1.5	复合数据类型。 .....	26
7.1.6	扩展数据类型 .....	26
7.1.6.1	概述.....	26
7.1.6.2	八位字节 字符串.....	26
7.1.6.3	可显示字符串.....	26
7.1.6.4	Unicode 字符串。 .....	27
7.1.6.5	时间.....	27
7.1.6.6	时间差.....	27
7.1.6.7	域.....	27
7.2	通信对象 .....	27
7.2.1	概述 .....	27
7.2.2	过程数据对象 ( PDO ).....	28
7.2.2.1	概述.....	28
7.2.2.2	传输 模式.....	29
7.2.2.3	触发模式.....	29
7.2.2.4	PDO 服务 .....	30
7.2.2.4.1	概述.....	30
7.2.2.4.2	服务 PDO 写.....	30
7.2.2.4.3	服务 PDO 读 .....	30
7.2.2.5	PDO 协议 .....	31
7.2.2.5.1	协议 PDO 写.....	31
7.2.2.5.2	协议 PDO 读.....	31
7.2.3	多路复用 PDO ( MPDO ) .....	31
7.2.3.1	简述.....	31
7.2.3.2	MPDO 寻址模式。 .....	32
7.2.3.2.1	目的地址模式(DAM).....	32
7.2.3.2.2	源地址模式 (SAM).....	32
7.2.3.3	MPDO 服务.....	32
7.2.3.3.1	简述.....	32
7.2.3.3.2	服务 MPDO 写 .....	32
7.2.3.4	MPDO 协议.....	33
7.2.3.4.1	协议 MPDO 写 .....	33
7.2.4	服务数据对象 ( SDO ) .....	34
7.2.4.1	简述.....	34
7.2.4.2	SDO 服务。 .....	35
7.2.4.2.1	简述.....	35
7.2.4.2.2	服务 SDO 下载.....	35
7.2.4.2.3	服务 SDO 下载启动.....	36
7.2.4.2.4	服务 SDO 分段下载.....	37
7.2.4.2.5	服务 SDO 上传.....	37
7.2.4.2.6	服务 SDO 上传启动.....	38

7.2.4.2.7	服务 SDO 分段上传.....	39
7.2.4.2.8	服务 SDO 块下载.....	39
7.2.4.2.9	服务 SDO 块下载启动.....	40
7.2.4.2.10	服务 SDO 块下载子块.....	41
7.2.4.2.11	服务 SDO 块下载 结束.....	42
7.2.4.2.12	服务 SDO 块 上传.....	42
7.2.4.2.13	服务 SDO 块上传 启动.....	43
7.2.4.2.14	服务 SDO 块上传 子块.....	44
7.2.4.2.15	服务 SDO 块上传 结束.....	45
7.2.4.2.16	服务 SDO 中止传输.....	45
7.2.4.3	SDO 协议.....	46
7.2.4.3.1	简述.....	46
7.2.4.3.2	协议 SDO 下载.....	46
7.2.4.3.3	协议 SDO 下载启动.....	47
7.2.4.3.4	协议 SDO 分段下载.....	48
7.2.4.3.5	协议 SDO 上传.....	49
7.2.4.3.6	协议 SDO 上传 启动.....	50
7.2.4.3.7	协议 SDO 分段上传.....	51
7.2.4.3.8	协议 SDO 块下载.....	52
7.2.4.3.9	协议 SDO 块下载启动.....	53
7.2.4.3.10	协议 SDO 块下载子块.....	54
7.2.4.3.11	协议 SDO 块下载结束.....	55
7.2.4.3.12	协议 PDO 块 上传.....	56
7.2.4.3.13	协议 SDO 块上传 启动.....	57
7.2.4.3.14	协议的 SDO 块上传子块.....	58
7.2.4.3.15	协议 SDO 块上传 结束.....	59
7.2.4.3.16	验证 SDO 块传输的 CRC 算法.....	59
7.2.4.3.17	协议 SDO 中止 传输.....	60
7.2.5	同步对象 (SYNC).....	62
7.2.5.1	简述.....	62
7.2.5.2	SYNC 服务.....	62
7.2.5.2.1	简述.....	62
7.2.5.2.2	服务同步写.....	62
7.2.5.3	同步协议.....	63
7.2.5.3.1	协议同步写.....	63
7.2.6	时间戳对象 (TIME).....	63
7.2.6.1	简述.....	63
7.2.6.2	TIME 服务.....	63
7.2.6.2.1	简述.....	63
7.2.6.2.2	服务 TIME 写.....	63
7.2.6.3	TIME 协议.....	64
7.2.6.3.1	协议 TIME 写.....	64
7.2.7	紧急对象 (EMCY).....	64
7.2.7.1	紧急对象的使用.....	64
7.2.7.2	紧急的对象服务.....	67
7.2.7.2.1	简述.....	67
7.2.7.2.2	服务 EMCY 写.....	67
7.2.7.3	紧急对象协议.....	67
7.2.7.3.1	协议 EMCY 写.....	67
7.2.8	网络管理.....	68
7.2.8.1	简述.....	68

7.2.8.2	NMT 服务 .....	68
7.2.8.2.1	节点控制服务 .....	68
7.2.8.2.1.1	简述 .....	68
7.2.8.2.1.2	启动远程节点服务 .....	68
7.2.8.2.1.3	停止远程节点服务 .....	68
7.2.8.2.1.4	进入预操作服务 .....	69
7.2.8.2.1.5	复位节点服务 .....	69
7.2.8.2.1.6	复位通信服务 .....	69
7.2.8.2.2	错误控制 服务 .....	70
7.2.8.2.2.1	节点监护事件服务 .....	70
7.2.8.2.2.2	节点监护事件服务 .....	71
7.2.8.2.2.3	心跳事件服务 .....	71
7.2.8.2.3	Boot-up 服务 .....	72
7.2.8.2.3.1	Boot-up 事件服务 .....	72
7.2.8.3	NMT 协议 .....	72
7.2.8.3.1	节点控制协议 .....	72
7.2.8.3.1.1	启动远程 节点协议 .....	72
7.2.8.3.1.2	停止远程节点协议 .....	72
7.2.8.3.1.3	进入预操作态协议 .....	73
7.2.8.3.1.4	复位节点协议 .....	73
7.2.8.3.1.5	复位通信协议 .....	73
7.2.8.3.2	错误控制协议 .....	74
7.2.8.3.2.1	节点监护协议 .....	74
7.2.8.3.2.2	心跳协议 .....	75
7.2.8.3.3	boot-up 协议 .....	76
7.3	网络初始化和系统 boot-up .....	77
7.3.1	简化的 NMT 启动 .....	77
7.3.2	NMT 状态机 .....	78
7.3.2.1	概述 .....	78
7.3.2.2	NMT 状态 .....	79
7.3.2.2.1	NMT 初始化态 .....	79
7.3.2.2.2	NMT 预操作态 .....	79
7.3.2.2.3	NMT 操作态 .....	80
7.3.2.2.4	NMT 停止态 .....	80
7.3.2.2.5	NMT 状态和通信对象关系 .....	80
7.3.2.3	NMT 状态转换 .....	80
7.3.3	通用预定义连接集 .....	80
7.3.4	特定预定义连接集 .....	81
7.3.5	受限 CAN-IDs .....	82
7.4	对象字典 .....	82
7.4.1	常规结构 .....	82
7.4.2	索引和子索引 的使用 .....	83
7.4.3	对象类型码的使用 .....	84
7.4.4	数据类型的使用 .....	84
7.4.5	访问权限的使用 .....	84
7.4.6	类别和条目类的使用 .....	85
7.4.7	数据类型入口的使用 .....	85
7.4.7.1	简述 .....	85
7.4.7.2	对象字典条目的组织结构 .....	87
7.4.8	预定义复合数据类型规范 .....	87
7.4.8.1	PDO 通讯参数记录规格 .....	87

7.4.8.2	PDO 映射参数记录规格 .....	88
7.4.8.3	SDO 参数记录规格 .....	88
7.4.8.4	身份记录规格.....	88
7.4.8.5	OS 调试记录规格.....	88
7.4.8.6	OS 命令记录规格.....	89
7.5	通信协议规范 .....	89
7.5.1	对象及其条目描述规范 .....	89
7.5.2	通信协议对象的详细规范 .....	90
7.5.2.1	对象 1000h: 设备类型.....	90
7.5.2.2	对象 1001h: 错误寄存器.....	91
7.5.2.3	对象 1002h: 制造商状态寄存器.....	92
7.5.2.4	对象 1003h: 预定义错误域.....	92
7.5.2.5	对象 1005h: COB - ID 同步消息 .....	94
7.5.2.6	对象 1006h: 通信循环周期 .....	95
7.5.2.7	对象 1007h: 同步窗长度.....	95
7.5.2.8	对象 1008h: 制造商设备名称.....	96
7.5.2.9	对象 1009h: 制造商的硬件版本.....	96
7.5.2.10	对象 100Ah: 制造商软件版本 .....	97
7.5.2.11	对象 100Ch: 监护周期 .....	97
7.5.2.12	对象 100Dh: 生存周期因子 .....	98
7.5.2.13	对象 1010h: 保存参数 .....	98
7.5.2.14	对象 1011h: 恢复缺省参数 .....	100
7.5.2.15	对象 1012h: 时间戳对象 COB-ID .....	103
7.5.2.16	对象 1013h: 高分辨率时间戳.....	104
7.5.2.17	对象 1014h: EMCY COB-ID.....	104
7.5.2.18	对象 1015h: EMCY 禁止时间 .....	105
7.5.2.19	对象 1016h: 消费者心跳超时.....	106
7.5.2.20	对象 1017h: 生产者心跳超时.....	107
7.5.2.21	对象 1018h: 对象身份.....	108
7.5.2.22	对象 1019h: 同步计数器溢出值.....	109
7.5.2.23	对象 1020h: 验证配置 .....	110
7.5.2.24	对象 1021h: 存储 EDS .....	111
7.5.2.25	对象 1022h: 存储格式 .....	112
7.5.2.26	对象 1023h: OS 命令 .....	113
7.5.2.27	对象 1024h: OS 命令模式 .....	114
7.5.2.28	对象 1025h: OS 调试接口 .....	115
7.5.2.29	对象 1026h: OS 提示符命令接口 .....	116
7.5.2.30	对象 1027h: 模块列表.....	117
7.5.2.31	对象 1028h: 紧急消费对象 .....	118
7.5.2.32	对象 1029h: 错误行为对象 .....	120
7.5.2.33	对象 1200h~127Fh: SDO 服务器参数 .....	121
7.5.2.34	对象 1280h~12FFh: SDO 客户端参数.....	124
7.5.2.35	对象 1400h~15FFh: RPDO 通信参数 .....	126
7.5.2.36	对象 1600h ~17FFh: RPDO 映射参数 .....	129
7.5.2.37	对象 1800h ~19FFh: TPDO 通信参数.....	132
7.5.2.38	对象 1A00h ~1BFFh: TPDO 映射参数 .....	137
7.5.2.39	对象 1FA0H~1FCFh: 对象扫描仪列表 .....	139
7.5.2.40	对象 1FD0H~1FFFh: 对象分配列表 .....	141
附件 A (更多信息).....		143

---

注译版本记录 ..... 145



## 1 适用范围

本规范定义了 CANopen 应用层。包括数据类型、编码规则、对象字典以及 CANopen 通信服务协议。此外，本规范也定义了 CANopen 网络管理服务协议。

本规范规定了 CANopen 通信协议，例如物理层、通信对象标识符预定义连接集、紧急对象(Emergency)、时间戳和同步通信对象。

## 2 参考资料

### 2.1 规范性引用

/EN61131-3/	EN 61131-3, 可编程控制器——第 3 部分: 编程语言
/ISO7498-1/	ISO 7498-1, 信息技术-开放系统互连-基本参考模型: 基本模型
/ISO8859/	ISO 8859, 信息技术——8 位单字节编码图形字符集
/ISO11898-1/	ISO 11898-1, 道路车辆——控制器区域网络(CAN)——第 1 部分: 数据链路层和物理信令
/ISO11898-2/	ISO 11898-2, 道路车辆——控制器区域网络(CAN)——第 2 部分: 高速介质访问单元
/ISO11898-3/	ISO 11898-3, 道路车辆——控制器区域网络(CAN)——第 3 部分: 低速度、容错的介质相关接口。
/ISO10646/	ISO 10646, 信息技术-通用多八位编码字符集(UCS)

### 2.2 信息参考

/IEEE754/	IEEE 754, 标准的二进制的浮点运算
/IEC62390/	IEC TR 62390, 常规的自动化设备 - 协议的准则

## 3 缩写和定义

### 3.1 缩写

ARQ	自动重复请求
CAN	控制局域网
CAN ID	CAN 标识符
COB	通信对象
COB-ID	COB 标识符
CRC	循环冗余校验
CSDO	Client-SDO
DAM	目的地址模式
FSA	有限状态机
LLC	逻辑链路控制
LSB	最低位/字节
MAC	介质访问控制
MDI	介质相关接口
MPDO	多路复用 PDO
MSB	最高位/字节
NMT	网络管理
NODE-ID	节点标识符
OSI	开放系统互连
PDO	过程数据对象
PLS	物理层的信令

PMA	物理介质连接
RPDO	接收 PDO
RTR	远程传输请求
SAM	源地址模式
SDO	服务数据对象
SSDO	Server-SDO
SYNC	同步对象
TPDO	发送 PDO

## 3.2 定义

### 标准帧

/ISO11898-1/中定义的最多可包含 8 个字节数据和 11 位标识符的消息

### 扩展帧

/ISO11898-1/中定义最多可包含 8 个字节数据和 29 位标识符的消息

### CAN-ID

/ISO11898-1/中定义的 CAN 数据和远程帧标识符

### COB-ID

包含 CAN-ID 和附加控制位的标识符

### 实体

特指事物如人、地点、过程、概念、组织或事件

### FSA

若干计算行为组成的模式作为一个状态，一个启动状态，输入一个字母，映射输入符号和当前状态到下一状态的转换功能；以一个字符串输入作为启动状态的计算起始；依赖传递函数变换到新状态（译者注：无法按字面翻译，译者定义为表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型，即有限状态机）。

### 设备域

1. 自动化系统中独立联网的物理实体，在特定的上下文和分隔符中使用自身接口执行指定的功能。
2. 在自动化系统中向其他实体执行控制、操纵和/或传感功能和相关接口的实体。

### 逻辑设备

根据设备域模型所组织的一系列对象和行为，描述了该设备的数据和行为使网络所理解。

### Node-ID

网络范围内每个 CANopen 设备的独特标识

### 对象

封装了状态和行为有明确界限和身份的实体。

### 虚拟设备

能够像域设备一样完成所属功能事件的软件实体。

## 4 建模

### 4.1 设备域模型

如图 1 所示，设备域至少包含一个 CANopen 设备。其中每个 CANopen 设备至少带有一个包含数据链路层(见本章第 6 节)和物理层(见本章第 5 节)的网络接口、一个 node-ID、至少一种通信状态机 (FSA)。通信状态机不仅带有 NMT 从状态机(见 7.3.2)，还包括紧急状态机(见 7.2.7)等其他附加状态机。这些附加状态机定义于所谓的框架协议内，不属于本协议范围。一个 CANopen 设备至少包含一个多至 8 个逻辑设备，且不可分割于多个设备域。每个逻辑设备可包含一个逻辑设备状态机(可选)和多个虚拟设备。逻辑设备不可分割于多个 CANopen 设备。逻辑设备定义于所谓的设备协议中(见 4.5.1)，不在本协议的范围内。每个虚拟设备包含一个虚拟设备状态机且不可分割于多个逻辑设备。虚拟设备定义于所谓的应用协议(请参阅子句 4.5.2)，不在本协议范围内。设备域的最小化结构如图 2 所示。

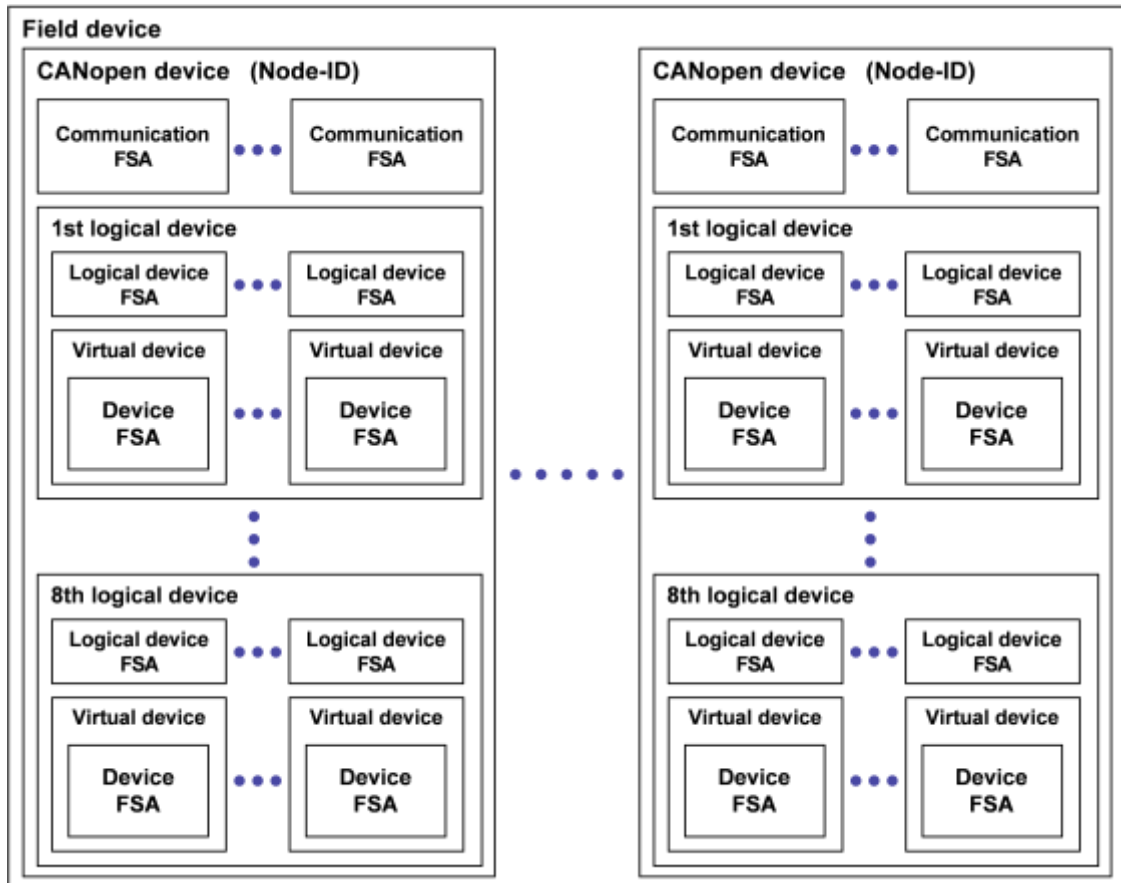


图 1: 设备域模型



图 2: 最小化设备域

## 4.2 通信参考模型

### 4.2.1 简述



图 3: 通信参考模型

通信概念符合 ISO-OSI 参考模型(图 3 左侧; 见/ISO7498-1/)。

### 4.2.2 CANopen 应用层

#### 4.2.2.1 概述

应用层用来描述 CANopen 设备间配置和同步实时数据的机制。该功能在应用层中被分割为若干不同的逻辑服务对象。服务对象提供特定的功能和所有相关的服务。在本协议的相关服务对象规范中进行具体描述。

通过调用应用层的服务对象接口来完成应用交互。协议提供的服务经由数据链路层实现交换数据。在本协议的服务对象相关协议规范中进行具体描述。

#### 4.2.2.2 服务原语

服务原语意指应用程序与应用程序之间交互。有四种不同的原语:

- 应用程序发起到应用层的服务请求。
- 应用层向应用程序发送的通知, 汇报一个应用层检测到的内部事件或者表明其收到一个服务请求。

- 应用程序为接收到的一个汇报向应用层做出应答。
- 应用层为收到应用程序的服务请求所作的响应确认。

### 4.2.2.3 应用层服务

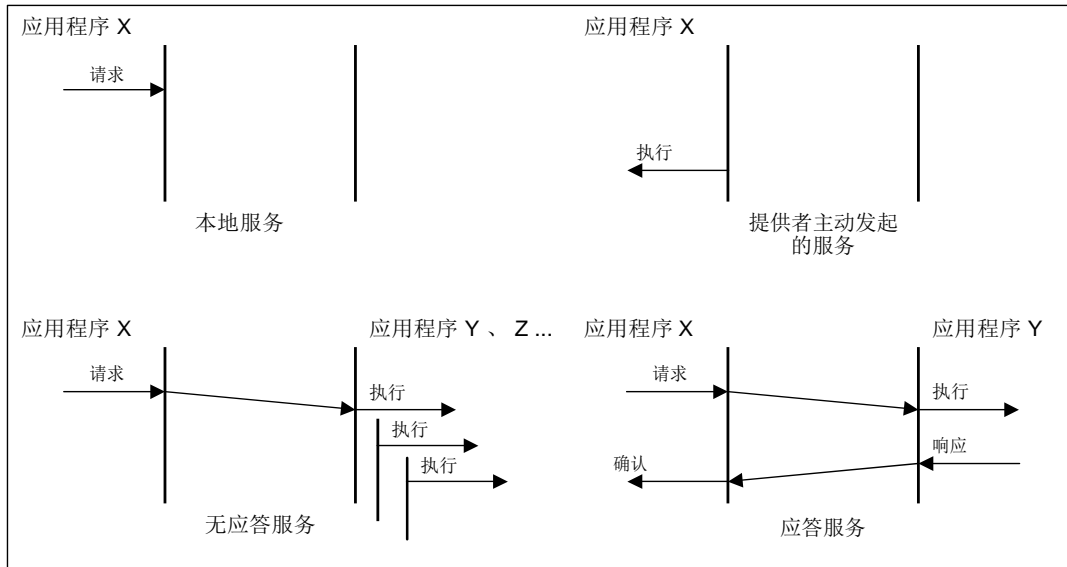


图 4：应用层服务

应用层服务定义若干在和协同操作的应用程序间交换原语的特定服务对象。CANopen 支持的应用层服务见图 4 所示。

- 本地服务只涉及本地的服务对象。应用程序向本地服务对象发出请求，执行请求的服务对象在本地，而并不与远程对应的服务对象产生通信行为。
- 发起的服务只涉及本地的服务对象。服务对象(即服务提供者)检测到一个事件但该事件并非一个请求服务。事件被通报给应用程序。
- 一次无应答服务涉及一个或多个对等服务对象。应用程序向本地服务对象发出一个请求。此请求将被传送到远程对应的服务对象，每个对应服务对象都会将请求作为一个操作行为传递给所属应用程序。处理结果不返回。
- 一次带应答服务只涉及远程对应的服务对象。应用程序向本地服务对象发出一个请求。此请求将传送到对应的远程服务对象，对应的服务对象将其作为一个操作行为传递给所属应用程序。其产生应答回传给请求方应用程序。

无应答服务和应答服务统称为远程服务。

## 4.3 CANopen 设备模型

### 4.3.1 简述

CANopen 设备的结构如下(图 5 所示)：

- 通信 - 此功能单元提供通信对象和与之相应的通过底层网络结构传输数据的能力。
- 对象字典 - 对象字典集合了设备上所有影响应用程序对象、通信对象和状态机行为的数据项。
- 应用程序 - 应用程序包括与过程环境能够产生相互作用的设备功能。(译者注：这里的应用程序仅只参与通信过程，会对网络其他设备造成影响的相关设备功能实现)

因此，对象字典是通信对象和应用程序对象之间的接口。(译者注：理解这一点很重要)

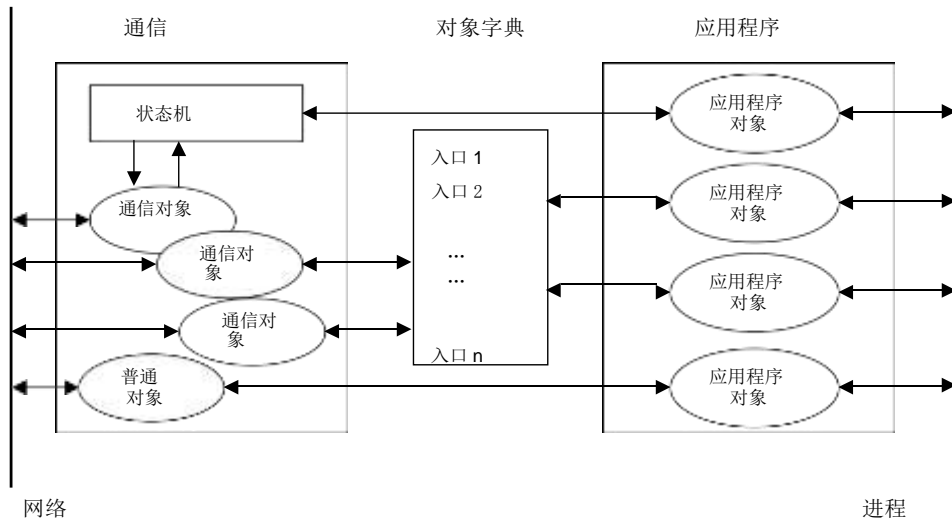


图 5: CANopen 设备模型

## 4.4 通信协议分类

### 4.4.1 简述

通信协议分类描述不同通信协议的原则和消息传输触发的可用模式。

CANopen 通信协议支持同步和事件驱动的消息传输。通过同步传输使得网络范围内的协同数据采集和动作输出成为可能。同步传输依赖于预定义通信对象（译者注：主要是指支持预定义连接集的 PDO）。同步传输遵循预定义的同步消息机制、事件驱动的消息传输可以在任何时间发起。

事件性质通信机制的基本属性要求定义通信的禁止时间。为保证网络上的低优先级通信对象不被饿死，需要为通信对象设计禁止时间。即将通信对象定义两次传输服务间的最小时间间隔作为禁止时间。

就其功能，归纳出三种类型的通信协议模型

- 主/从协议 (见 4.4.2)
- 客户端/服务器协议 (见 4.4.3)
- 生产者/消费者协议 (见 4.4.4)

### 4.4.2 主/从协议

在任何时候 CANopen 网络中都恰好存在一设备以主设备身份执行特定功能。CANopen 网络中所有其他的设备会被视为从设备。按照协议规定主设备会向被寻址从设备（们）发送请求，而从设备（们）响应。图 6 定义了无应答的主/从协议模式。图 7 定义了带应答的主/从协议模式。



图 6: 无应答的主/从通信协议

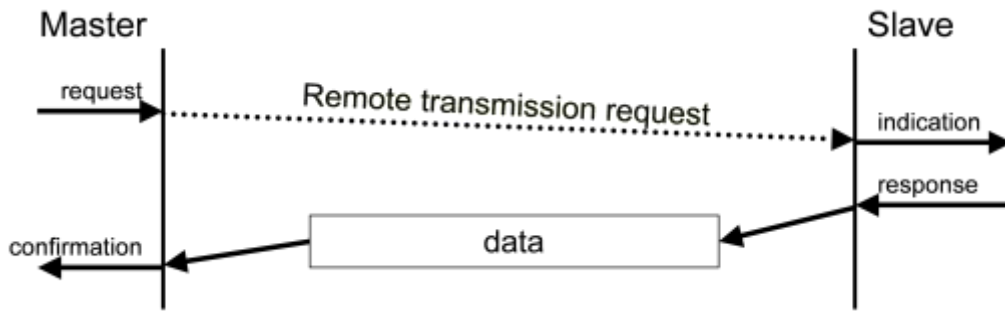


图 7：带应答的主/从通信协议

#### 4.4.3 客户端/服务器协议

单一客户和单一服务器之间的一种通信协议。客户端发出请求(上传/下载)触发服务器执行某项任务。在完成的任务后服务器应答请求。图 8 定义了客户端 / 服务器协议模式。

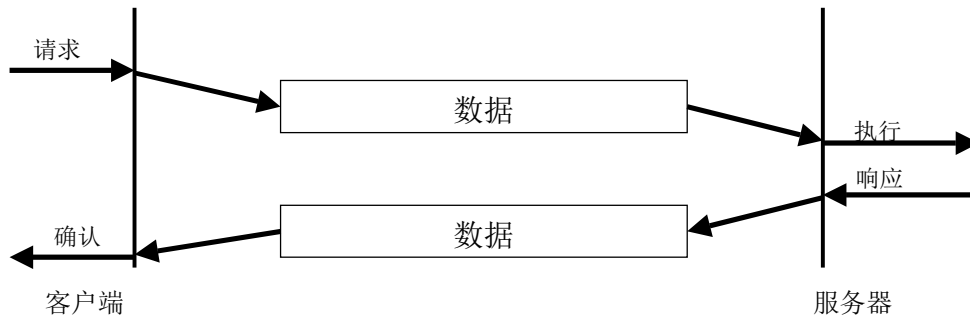


图 8：客户端/服务器通信协议

#### 4.4.4 生产者/消费者协议-推/拉模式

生产者 / 消费者协议涉及一个生产者和若干消费者(没有也可以)。“推”模式定义见图 9，其特点是由生产者发起一个无确认的协议请求。拉模式的定义见图 10，其特点是由生产者发起需要确认的协议请求。



图 9：推模式



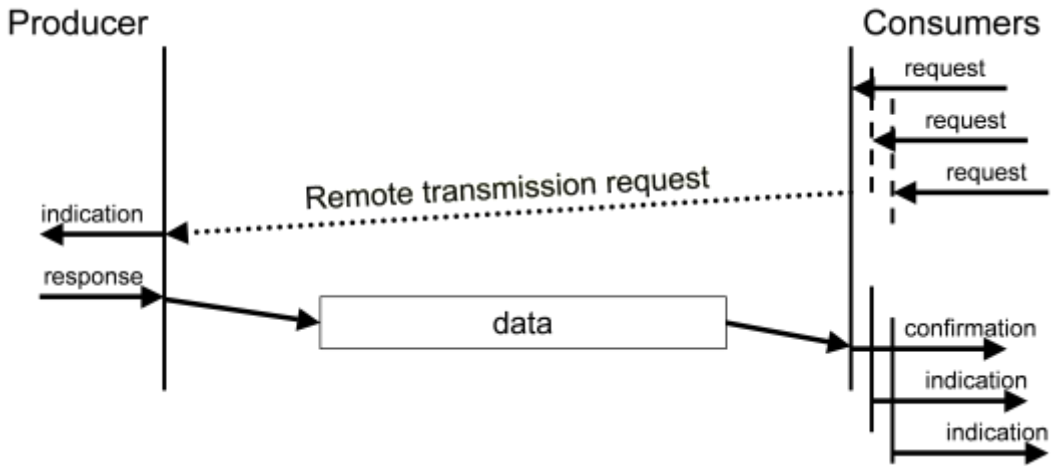


图 10：拉模式

#### 4.4.5 对象字典

对象字典中基本上是一组有序的、预定义范式的、经由网络可访问的接口。每个接口对象通过一个 16 位的索引和 8 位的子索引寻址。

### 4.5 网络系统模型

#### 4.5.1 设备协议

设备协议是对对象字典内对象的描述，对象字典属于包含一台虚拟设备的逻辑设备。描述包含了对对象的功能描述和规格描述。功能描述定义了对象在对象字典内的行为。规格描述定义了对象应执行或可执行的 CANopen 网络访问，以及依赖于什么样的访问方法。

#### 4.5.1 应用协议

应用协议是一种虚拟设备对象字典及其对象的描述，适配网络中所有 CANopen 设备（译注：接入 CANopen 网络的所有设备都要支持）。功能描述定义了对象在对象字典内的行为。规格描述定义了对象应执行或可执行的 CANopen 网络访问，以及索引和子索引是怎么样被访问的。

## 5 物理层

### 5.1 OSI 参考模型

根据 OSI 参考模型，物理层 (如图 11 所示) 被分为三个子层：

- 介质相关接口
- 物理介质连接
- 物理信令



图 11: 物理层参考模型

### 5.2 介质相关接口

介质相关接口不属于本文范畴。

### 5.3 物理介质连接

支持 CANopen 设备连接的物理介质应满足通用高速两线制差分信号传输协议 / ISO11898-2 / 要求。

备注 其他物理介质接入技术如 / ISO11898-3 / 可能被使用。

根据 /ISO11898-2/ 规定使用的高速收发器最大等级为  $V_{\text{可以\_H}}$  和  $V_L$  应至 +16V。CANopen 设备之间的电隔离可选。这里建议使用的收发器的误连接耐压等级在 30V 以上。

### 5.4 物理信令

位编码/解码和同步应符合 / ISO11898-1 / 规范定义。

位定时应符合 / ISO11898-1 / 规范的要求，建议参照表 1 的定义 (总线长度范围估计见表 2)。表中所示速度至少支持一种，建议都支持。

表 1: 推荐位定时设置

位速率	标称位时间 $t_b$	有效的采样点位置范围	推荐的采样点位置
1 Mbit / s	1 $\mu$ s	75 %-90 %	87.5 %
800 kbit /	1.25 $\mu$ s	75 %- 90 %	87.5 %
500 kbit /	2 $\mu$ s	85 %-90 %	87.5 %
250 kbit /	4 $\mu$ s	85 %-90 %	87.5 %
125 kbit /	8 $\mu$ s	85 %-90 %	87.5 %
50 Kbit /	20 $\mu$ s	85 %-90 %	87.5 %
20 Kbit /	50 $\mu$ s	85 %-90 %	87.5 %
10 Kbit /	100 $\mu$ s	85 %-90 %	87.5 %

表 2: 估计的总线长度

位速率	总线长度 <sup>(1)</sup>
1 Mbit/s	25 m
800 kbit/s	50 m
500 kbit/s	100 m
250 kbit/s	250 m
125 kbit/s	500 m
50 Kbit/s	1000 m
20 Kbit/s	2500 m
10 Kbit/s	5000 m

注 1: 总线长度的估计是基于建议位置的采样点。

总线长度的估计是基于传播延迟时间为 5 ns/m。延迟时间要考虑到所使用控制器、CAN 收发器、以及可选的光耦合器。

## 6 数据链路层

### 6.1 概述

所述的网络应基于/ISO11898-1/的数据链路层及其子层。

### 6.2 CAN 帧类型

本规范基于 11 位 CAN-ID 帧。不要求支持 29 位标识符的 CAN 扩展帧。

注：然而，由于某些应用要求使用 29 位扩展帧，如果所有 CANopen 设备都支持扩展帧，网络可以运行在该模式下。

## 7 应用层

### 7.1 数据类型和编码规则

#### 7.1.1 数据类型和编码规则简述

为了在网络上交换有意义的数据，必须保证数据格式为生产者和消费者所识别。本规范的模型提供了此概念的数据类型。

编码规则定义了数据所表达值的类型和传输语义。值被表达比位序列。位特序列以字节为传输单元。各数据类型编码风格为小端模式。

应用程序通常请求的都是基本数据类型。要使用复合数据类型机制，可以扩展可用数据类型表。一些常用的数据类型被定义为如“Visible String”或“Time of Day”所示（参见于 7.1.6.3 和 7.1.6.5）。本文从技术角度对复合数据类型的定义一般取其应用含义，如“DEFTYPES”而不是“DEFSTRUCTS”。

#### 7.1.2 数据类型定义

数据类型定义由值和数据类型编码之间的关系决定。类型定义的命名中体现数据类型。数据语法和数据类型定义如下（见 / EN61131-3 /）。

data_definition	::= type_name data_name
type_definition	::= constructor type_name
constructor	::= compound_constructor   basic_constructor
compound_constructor	::= array_constructor   structure_constructor
array_constructor	::= 'ARRAY' '[' length ']' 'OF' type_name
structure_constructor	::= 'STRUCT' 'OF' component_list
component_list	::= component { ',' component }
component	::= type_name component_name
basic_constructor	::= 'BOOLEAN'   'VOID' bit_size   'INTEGER' bit_size   'UNSIGNED' bit_size   'REAL32'   'REAL64'   'NIL'
bit_size	::= '1'   '2'   <...>   '64'
length	::= positive_integer
data_name	::= symbolic_name
type_name	::= symbolic_name
component_name	::= symbolic_name
symbolic_name	::= letter { [ '_' ] ( letter   digit ) }
positive_integer	::= ( '1'   '2'   <...>   '9' ) { digit }
letter	::= 'A'   'B'   <...>   'Z'   'a'   'b'   <...>   'z'
digit	::= '0'   '1'   <...>   '9'

不要使用递归定义。

type\_definition 定义的数据类型称为 basic( res.~compound), basic\_constructor 定义的被称为 constructor( res.~compound\_constructor)。

### 7.1.3 位序列

#### 7.1.3.1 位序列定义

一位的值为 0 或 1。一个位序列  $b$  是一个包含 0 或更多位的有序组。如果位序列  $b$  包含多于 0 位, 可以被表示为  $b_j, j \geq 0. b_0, \dots, b_{n-1}$  是其中的位,  $n$  是正整数。

因而

$$b = b_0 b_1 \dots b_{n-1}$$

被称为一个长度  $|b| = n$  的位序列。长度为 0 的空序列用  $\varepsilon$  表示。

例:  $10110100_b, 1_b, 101_b$  等都是的位序列。

对下面位序列的翻转操作( $\neg$ )

$$b = b_0 b_1 \dots b_{n-1}$$

位序列翻转

$$\neg b = \neg b_0 \neg b_1 \dots \neg b_{n-1}$$

这里  $\neg 0 = 1, \neg 1 = 0$ 。

位序列的基本操作是连接。

$a = a_0 \dots a_{m-1}$  和  $b = b_0 \dots b_{n-1}$  的位序列连接  $a$  和  $b$  表示  $ab$  为

$$Ab = a_0 \dots a_{m-1} b_0 \dots b_{n-1}$$

例如:  $(10)(111) = 10111$  是 10 和 111 的连接。

以下为任意的位序列  $a$  和  $b$  的连接:

$$|ab| = |a| + |b|$$

和

$$\varepsilon a = a \varepsilon = a$$

#### 7.1.3.2 位序列的传输语法

为了跨网络传输, 位序列被标记为字节序列。在这里和下面的十六进制表示法用于字节表达。让  $b = b_0 \dots b_{n-1}$  的位序列的其中  $n \leq 64$ 。  $k$  表示非负整数,  $8(k-1) < n \leq 8k$ 。  $b$  在传输过程中被打包为表 12 的形式。  $b_i, i \geq n$  被忽略。

八位位组 1 首先传输而  $k$  最后。因此位序列以下列顺序方式传输:

$$b_7, b_6, \dots, b_0, b_{15}, \dots, b_8, \dots$$

八位码	1.	2.	k.
	b7..b0	b15..b8	b8k-1..b8k-8

图 12 : 位序列的传输语法

例如:

$$\begin{array}{r}
 \text{位 9} \quad \dots \quad \text{位 0} \\
 10_{\text{b}} \quad 0001_{\text{b}} \quad 1100_{\text{b}} \\
 2_{\text{h}} \quad 1_{\text{h}} \quad C_{\text{h}} \\
 = 21C_{\text{h}}
 \end{array}$$

位序列  $b = b_0 \dots b_9 = 0011\ 1000\ 01_{\text{b}}$  表示 `UNSIGNED10` 的值  $21C_{\text{h}}$ , 被转换为两个八位字节传输:

$1C_{\text{h}}$  和  $02_{\text{h}}$ 。

## 7.1.4 基本的数据类型。

### 7.1.4.1 概述

基本的数据类型 “type\_name” 与其创建字符相同( aka *Symbolic\_name*), 例如,

`BOOLEAN BOOLEAN`

是 `BOOLEAN` 数据类型的的类型定义。

### 7.1.4.2 NIL

基本数据类型 `NIL` 表示  $\epsilon$ 。

### 7.1.4.3 Boolean

基本数据类型 `BOOLEAN` 的值为 `TRUE` 或 `FALSE`。该值表示的位序列的长度为 1。值为 `TRUE` ( res. `FALSE` ) 代表的是位序列 1 ( res. 0 )。

### 7.1.4.4 Void

基本数据类型 `VOIDn` 表示长度为  $n$  的位序列。类型为 `VOIDn` 的数据值是未定义的。类型为 `VOIDn` 的位序列数据中的位要么显式指定, 要么标记为“随意”。

数据类型 `VOIDn` 是常用于保留字段和复合结构数据对齐的字节边界。

### 7.1.4.5 Unsigned Interger

基本数据类型 `UNSIGNEDn` 取值非负整数。范围是 0、...、 $2^n - 1$ 。该数据表示长度为  $n$  的位序列。

$$b = b_0 \dots b_{n-1}$$

分配的值

$$\text{UNSIGNEDn}(b) = b_{n-1}2^{n-1} + \dots + b_12^1 + b_02^0$$

请注意, 位顺序从左边最低字节开始。

例如: 值  $266 = 10A_{\text{h}}$  数据类型为 `UNSIGNED16` 在总线上以两个八位字节传输,  $0A_{\text{h}}$  然后是  $01_{\text{h}}$ 。

UNSIGNEDn 的数据类型传输定义在图 13 所示。

八位字节数据	1.	2.	3.	4.	5.	6.	7.	8.
UNSIGNED8	b <sub>7..b<sub>0</sub></sub>							
UNSIGNED16	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>						
UNSIGNED24	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>					
UNSIGNED32	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>				
UNSIGNED40	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>			
UNSIGNED48	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>	B <sub>47..b<sub>40</sub></sub>		
UNSIGNED56	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>	B <sub>47..b<sub>40</sub></sub>	B <sub>55..b<sub>48</sub></sub>	
UNSIGNED64	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>	B <sub>47..b<sub>40</sub></sub>	B <sub>55..b<sub>48</sub></sub>	B <sub>63..b<sub>56</sub></sub>

图 13 : 数据类型 UNSIGNEDn 的传输规则

### 7.1.4.6 Signed Integer

基本数据类型 INTEGERn 值为 整数。取值的范围是  $-2^{n-1} \sim 2^{n-1} - 1$ 。该数据表示长度为  $n$  的位序列。位序列

$$b = b_{n-1}..b_0$$

分配的值

$$\text{INTEGERn}(b) = b_{n-2}2^{n-2} + \dots + b_12^1 + b_02^0 \quad \text{如果 } b_{n-1} = 0$$

并执行双补运算

$$\text{INTEGERn}(b) = -\text{INTEGERn}(\bar{b}) - 1 \quad \text{如果 } b_{n-1} = 1$$

请注意，位顺序从左侧最低位开始。

例如：值  $-266 = \text{FEF6}_h$  数据类型 *INTERGER16* 在总线上以两个八位字节传输， $\text{F6}_h$  然后是  $\text{FE}_h$ 。

INTEGERn 的数据类型被定义见图 14。

八位字节数据	1.	2.	3.	4.	5.	6.	7.	8.
INTEGER8	b <sub>7..b<sub>0</sub></sub>							
INTEGER16	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>						
INTEGER24	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>					
INTEGER32	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>				
INTEGER40	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>			
INTEGER48	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>	B <sub>47..b<sub>40</sub></sub>		
INTEGER56	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>	B <sub>47..b<sub>40</sub></sub>	B <sub>55..b<sub>48</sub></sub>	
INTEGER64	b <sub>7..b<sub>0</sub></sub>	b <sub>15..b<sub>8</sub></sub>	B <sub>23..b<sub>16</sub></sub>	B <sub>31..b<sub>24</sub></sub>	B <sub>39..b<sub>32</sub></sub>	B <sub>47..b<sub>40</sub></sub>	B <sub>55..b<sub>48</sub></sub>	B <sub>63..b<sub>56</sub></sub>

图 14 : 数据类型 INTEGERn 传输规则



### 7.1.4.7 Floating-Point 数

基本数据类型 *REAL32* 和 *REAL64* 值为实数。

数据类型 *REAL32* 表示长度为 32 的位序列。编码的值遵循/ IEEE 754 /。传输语法见图 15。

数据类型 *REAL64* 表示长度为 64 的位序列。编码的值遵循/ IEEE 754 /。

长度为 32 的位序列有值 (有限非 0 实数、 $\pm 0$ 、 $\pm_{\infty}$ ) 或是 NaN (not-a-number)。位序列

$$b = b_0 \dots b_{31}$$

分配的值为 (有限的非零数)

$$REAL32(b) = (-1)^S 2^{E-127} (1+F)$$

这里

$S = b_{31}$  是符号。

$E = b_{30}2^7 + \dots + b_{23}2^0$ ,  $0 < E < 255$ 、为无偏 (un-biase) 指数。

$F = 2^{-23}(b_{22}2^{22} + \dots + b_12^1 + b_02^0)$  表示小数部分。

$E = 0$  是用于表达  $\pm 0$ 。  $E = 255$  是用于表示无穷大和 NaN's。请注意，位顺序从左侧最低位开始。

例如：

$$6.25 = 2^{E-127} (1 + F)$$

$$E = 129 = 2^7 + 2^0 \text{ 和}$$

$$F = 2^{-1} + 2^{-4} = 2^{-23} (2^{22} + 2^{19}) \text{ 项数表示为:}$$

S	E	F
$b_{31}$	$b_{30}..b_{23}$	$b_{22}..b_0$
0	100 0000 1 <sub>b</sub>	100 1000 0000 0000 0000 0000 <sub>b</sub>

$$6.25 = b_0..b_{31} = 0000 0000 0000 0000 0001 0011 0000 0010_b$$

转换顺序:

八位字节数据	1.	2.	3.	4.
REAL32	00h	00h	C8h	40h
	$b_7..b_0$	$b_{15}..b_8$	$B_{23}..b_{16}$	$B_{31}..b_{24}$

图 15 : 数据类型 *REAL32* 的传输规则

## 7.1.5 复合数据类型。

复合数据类型展开为包含基本数据类型的独立表单类型定义。相应地,复合数据类型‘*type\_name*’命名规则是由基本类型‘*basic\_type\_i*’组成的表化复合数据名‘*component\_name\_i*’确定。

复合数据类型由 ARRAY 和 STRUCT OF 构建。

```
STRUCT OF
    basic_type_1      component_name_1,
    basic_type_2      component_name_2,
    ...
    basic_type_N      component_name_N
type_name
```

ARRAY [长度] OF basic\_type type\_name

复合数据类的位序列由组成复合数据的各数据的位序列连接而成。

假设组件‘*component\_name\_i*’表达的位序列为

$$b(i), \text{ 其中 } i = 1, \dots, N$$

然后复合数据的位序列连接为

$$b_0(1) .. b_{n-1}(1) .. b_{n-1}(N).$$

例如:

如下数据类型

```
STRUCT OF
    INTEGER10      x,
    UNSIGNED5      u
NewData
```

假定  $x = -423 = 259_h$ ,  $u = 30 = 1E_h$ 。让  $b(x)$  和  $b(u)$  表示位序列  $x$  和  $u$ 。然后:

$$\begin{aligned} b(x) &= b_0(x) .. b_9(x) = 1001101001_b \\ b(u) &= b_0(u) .. b_4(u) = 01111_b \\ b(xu) &= b(x) b(u) = b_0(xu) .. b_{14}(xu) = 1001101001 01111_b \end{aligned}$$

该值的结构被转换为两个八位字节,  $59_h$  然后是  $7A_h$ 。

## 7.1.6 扩展数据类型

### 7.1.6.1 概述

扩展数据类型分为基本数据类型、复合数据类型在以下小节分别定义。

### 7.1.6.2 八位字节 字符串

数据类型 OCTET\_STRIN 长度定义如下, 长度 八位字节字符串的长度。

```
ARRAY [ length ] OF UNSIGNED8 OCTET_STRINGlength
```

### 7.1.6.3 可显示字符串

数据类型 VISIBLE\_STRING 长度定义如下, VISIBLE\_CHAR 类型数据取值  $0_h$  和  $20_h \sim 7E_h$ 。数据由 ISO 646-1973(E)7 位编码的字符解释。长度长度为可见字符串长度。

```
UNSIGNED8      VISIBLE_CHAR
ARRAY [ length ] OF VISIBLE_CHAR  VISIBLE_STRINGlength
```

无需  $0_h$  作为字符串结束标志。

### 7.1.6.4 Unicode 字符串。

数据类型 UNICODE\_STRING 长度定义如下；长度 unicode 字符串长度。

```
ARRAY [ length ] OF UNSIGNED16      UNICODE_STRING/length
```

### 7.1.6.5 时间

数据类型 TIME\_OF\_DAY 表示绝对时间。根据这一定义和编码规则, TIME\_OF\_DAY 由 48 位长度的位序列表达。

组件 ms 是午夜起算的毫秒计数,。组件 days 是自 1984 年 1 月 1 日以来的天数。

STRUCT OF

```
    UNSIGNED28 ms,
    VOID4      reserved,
    UNSIGNED16 days
```

TIME\_OF\_DAY

### 7.1.6.6 时间差

数据类型 TIME\_DIFFERENCE 表示时间差。根据这一定义和编码规则,时间差由 48 位长度的位序列表达。

ms 表示毫秒数。days 表示天数。

STRUCT OF

```
    UNSIGNED28 ms,
    VOID4      reserved,
    UNSIGNED16 days
```

TIME\_DIFFERENCE

### 7.1.6.7 域

域用于从客户端到服务器传输任意大的数据块,反之亦然。数据块内容由应用程序指定不在本文范围。

## 7.2 通信对象

### 7.2.1 概述

通信对象被描述为服务和协议。

所有服务以表格形式给出,包含为服务定义的所有原始参数。特定的服务类型包含不同的参数(如无应答、应答等等)。

所有服务都假定 CAN 的数据链路层和物理层未出错。 这些问题由应用程序解决因而不在本文范畴。

## 7.2.2 过程数据对象 (PDO)

### 7.2.2.1 概述

实时的数据传输通过 " 过程数据对象 (PDO )" 完成。传输 PDO 无协议开销。

由对象字典提供 PDO 数据和配置的接口。数据字典中对应的映射结构决定了一个 PDO 的数据类型和映射关系。如果 CANopen 设备支持可变映射的 PDO，则在配置过程中(见 7.3.1)，可通过 SDO 实现对 PDO 在数据字典中对应的配置进行修改。

CANopen 设备的 PDO 数量和长度可由应用规范、设备协议或应用协议指定。

PDO 分两种用法，发送和接收，归属 Transmit-PDO(TPDO) 和 Receive-PDO(RPDO)。支持 TPDO 的 CANopen 设备称为 PDO 生产者，支持 RPDO 的称为 PDO 消费者。PDO 由 PDO 通讯参数和 PDO 映射参数描述。其数据类型结构见 7.4.8。PDO 通讯参数描述了 PDO 的通信功能。PDO 映射参数包含了 PDO 传输内容信息。

每个 PDO 的通信和映射参数都是必须的。其对象介绍见 7.4。

设备协议中的 PDO 定义总会涉及 CANopen 设备中的 1<sup>st</sup> 逻辑设备。如果定义用到了 2<sup>nd</sup> 逻辑设备，设备协议定义的 PDO 编号增加 64(40<sub>h</sub>) 见表 3 中定义。

注意：每个逻辑设备不限制 PDO 数量，一个只包含单个逻辑设备的 CANopen 设备可以有多至 512 个 PDO。

表 3: PDO 编号计算举例。

CANopen 设备中的逻辑设备	CANopen 设备中的 PDO 编号	设备协议中的 PDO 编号
1 <sup>st</sup> 逻辑设备	PDO number + 0 (PDO1 to PDO64)	PDO number (PDO1 to PDO64)
2 <sup>nd</sup> 逻辑设备	PDO number + 64 (PDO65 to PDO128)	PDO number (PDO1 to PDO64)
3 <sup>rd</sup> 逻辑设备	PDO number + 128 (PDO129 to PDO192)	PDO number (PDO1 to PDO64)
4 <sup>th</sup> 逻辑设备	PDO number + 192 (PDO193 to PDO256)	PDO number (PDO1 to PDO64)
5 <sup>th</sup> 逻辑设备	PDO number + 256 (PDO257 to PDO320)	PDO number (PDO1 to PDO64)
6 <sup>th</sup> 逻辑设备	PDO number + 320 (PDO321 to PDO384)	PDO number (PDO1 to PDO64)
7 <sup>th</sup> 逻辑设备	PDO number + 384 (PDO385 to PDO448)	PDO number (PDO1 to PDO64)
8 <sup>th</sup> 逻辑设备	PDO number + 448 (PDO449 to PDO512)	PDO number (PDO1 to PDO64)

## 7.2.2.2 传输模式

PDO 传输模式分为:

- 同步传输
- 事件驱动的传输

CANopen 设备的同步由同步应用定期发送同步对象(SYNC 对象)来实现。SYNC 对象是一个预定义的通信对象(见 7.2.5)。图 16 说明了同步和事件驱动的传输原则过程。同步 PDO 在紧接着 SYNC 对象之后的同步窗中传输。

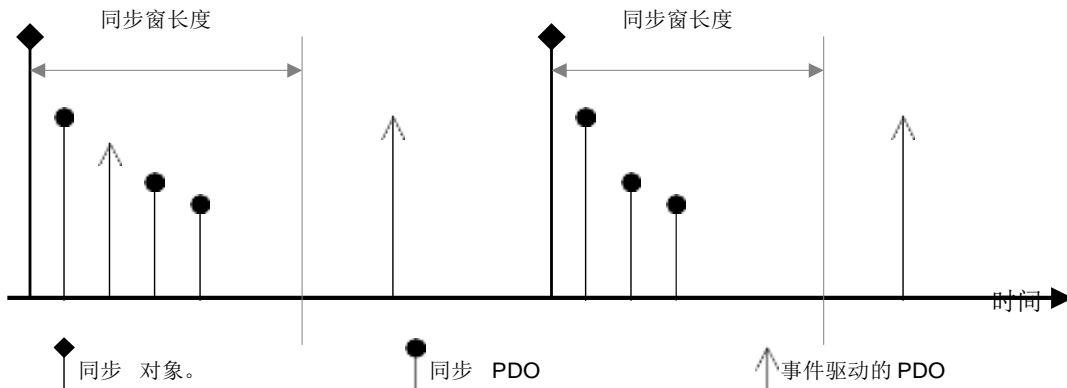


图 16：同步和事件驱动的传输

PDO 传输模式由传输类型参数指定，同样还有触发模式。

对于同步 TPDOs 传输类型还指定了传输率——同步周期倍数因数。传输类型 0 表示消息会在触发事件后再收到 SYNC（非周期）执行。1 表示每个 SYNC 触发一次消息。n 表示每隔 n 个 SYNC 触发一次消息。事件驱动的 TPDOs 与 SYNC 对象无关。

同步 RPDOs 收到数据后在随后的 SYNC 发生后传递给应用程序，传输类型同时指定了传输率。事件驱动 RPDOs 直接将数据传递给应用程序。

## 7.2.2.3 触发模式

有三种消息触发模式分类:

### - 事件和定时器驱动

由应用规范、设备协议、应用协议或定时(event-time)触发的消息传输。

### - 远程请求

PDO 的消费者发出一个 RTR 请求触发事件驱动 PDO 的发送。

### - 同步触发

由一定数量的 Sync 和内部事件(可选)触发。

## 7.2.2.4 PDO 服务

### 7.2.2.4.1 概述

PDO 传输遵循生产者 / 消费者关系见 4.4.4。

属性：

- PDO number: PDO 编号 [ 1..512 ]约束逻辑设备上的每个用户类型
- user type: 之一 { 消费者、生产者}
- data type: 据 PDO 映射
- inhibit-time:  $N * 100 \mu s$ 、 $N \geq 0$

### 7.2.2.4.2 服务 PDO 写

PDO 写是推模式。有零个或更多的 PDO 消费者。但只有一个 PDO 生产者。

通过这一服务的 PDO 生产者发送映射好的数据给消费者(们)。用于此服务的参数被定义在表 4 中。

表 4：PDO 写

参数	请求 / 执行
<b>要素</b>	<b>强制性</b>
PDO 编号	强制
数据	强制

### 7.2.2.4.3 服务 PDO 读

PDO 读是拉模式。有一位或更多的 PDO 消费者。但只有一个 PDO 生产者。

通过该服务, PDO 消费者请求生产者提供映射好的数据, 即远程响应。用于此服务的参数被定义在表 5 。

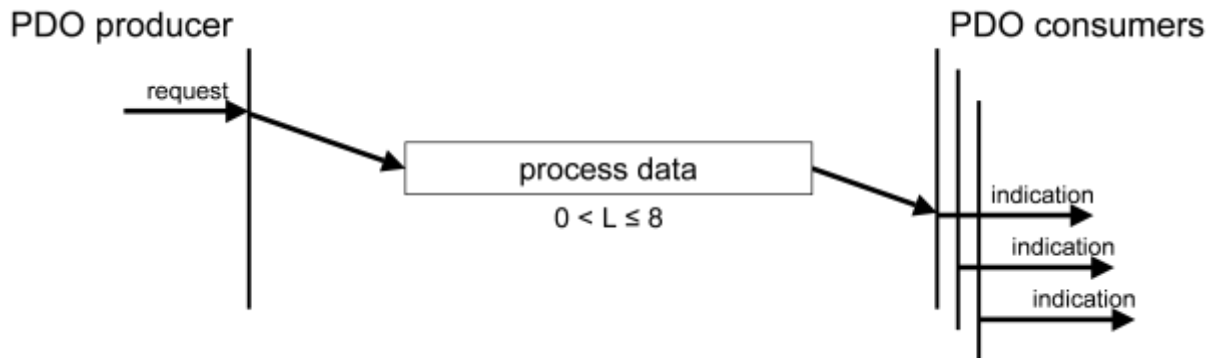
表 5：PDO 读

参数	请求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
PDO 编号	强制	
<b>远程结果</b>		<b>强制性</b>
数据		强制

## 7.2.2.5 PDO 协议

### 7.2.2.5.1 协议 PDO 写

请求无确认的服务 PDO 写。PDO 生产者将 PDO 过程数据上网。有可能有 0 到 n 个 PDO 消费者。在 PDO 消费者（们）接收到有效的 PDO 后执行相关操作。图 17 定义了 PDO 写协议。

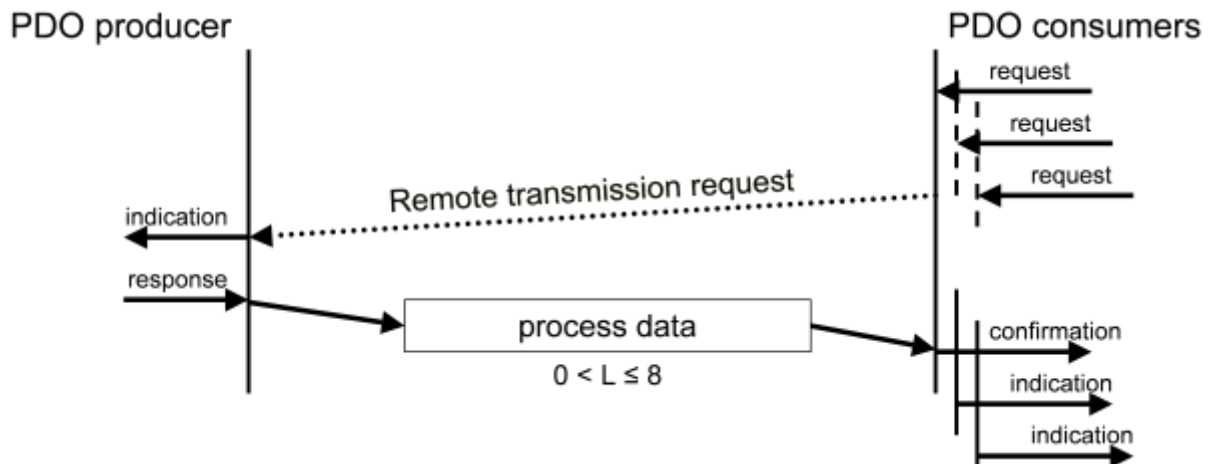


过程数据：L 字节的应用程序数据。

图 17：协议 PDO 写

### 7.2.2.5.2 协议 PDO 读

该服务 PDO 读须应答。一或多个 PDO 消费者发送 RTR 上网。PDO 生产者收到 RTR 后发送对应的 PDO。所有该 PDO 消费者关联执行。可能有 1 到 n 个 PDO 消费者。《阅读服务是可选的，取决于硬件的能力。图 18 指定 PDO 阅读协议。



Process 数据：L 字节的应用程序数据

图 18：协议 PDO 读

## 7.2.3 多路复用 PDO (MPDO)

### 7.2.3.1 简述

MPDO 提供了直接写 CANopen 设备对象字典的访问。数据的大小限制为最大 4 个字节。

有两种 MPDO 使用方法。第一是目的地址模式(DAM)MPDO，第二是源地址模式(SAM)MPDO。支持发送 MPDO 的 CANopen 设备称为 MPDO 生产者，支持接收 MPDO 的称为 MPDO 消费者。

由对象字典提供了 MPDO 数据和配置的接口。

“MPDOs 对应于对象中的对象字典和提供的接口的应用程序对象。”

## 7.2.3.2 MPDO 寻址模式。

### 7.2.3.2.1 目的地址模式(DAM)

标识该对象的 MPDO 消费者的对象字典对象提供了多路复用标识(见 7.2.3.4.1)。一个 DAM-MPDO 可被所有该 MPDO 消费者同时接收,也支持单一 MPDO 消费者。该写服务无应答,如果对象不存在产生 EMCY 帧。

发送 MPDO 由事件驱动,不支持定时、远程请求和同步触发模式。

### 7.2.3.2.2 源地址模式 (SAM)

多路复用 MPDO 依赖于 MPDO 生产者。每个 CANopen 设备只允许一个此类 MPDO 生产者存在。发送 MPDO 由事件驱动,不支持定时、远程请求和同步触发模式。MPDO 生产者使用一个扫描列表,以便知道发送哪些数据对象。MPDO 消费者使用分配列表,以便知道源到目的的关联。

## 7.2.3.3 MPDO 服务

### 7.2.3.3.1 简述

MPDO 传输遵循生产者 / 消费者关系, 见 4.4.4。

属性:

- PDO 编号: PDO 编号 [ 1..512 ]约束逻辑设备上的每个用户类型之一 { 消费者、生产者 }
- User type: 其中之一 { 消费者、生产者 }
- multiplexer: 其中包含索引和子索引, STRUCTURE OF UNSIGNED16, UNSIGNED8, 指 CANopen 设备对象字典的指定对象索引和子索引号。
- address type: 其中之一 { 源、目的 }
- Node-ID 消费者或生产者
- Inhibit-time:  $N * 100 \mu s$ 、 $N \geq 0$

### 7.2.3.3.2 服务 MPDO 写

服务 MPDO 写是推模式。有零个或更多的 MPDO 消费者。MPDO 生产者只有一个。用于此服务的参数被定义在表 6。

表 6 : 服务 MPDO 写

参数	要求 / 执行
<b>要素</b>	<b>强制性</b>
PDO 编号	强制
地址类型	强制
Node-ID	强制
Multiplexer	强制
数据	强制



## 7.2.3.4 MPDO 协议

### 7.2.3.4.1 协议 MPDO 写

请求服务 MPDO 写无应答。MPDO 生产者发送 MPDO 过程数据到网络。可能是有 0 到 n 个 MPDO 消费者，取决于给定的 node-ID。MPDO 消费者(们)接收到有效的 PDO 应执行。图 19 指定 MPDO 写入协议。

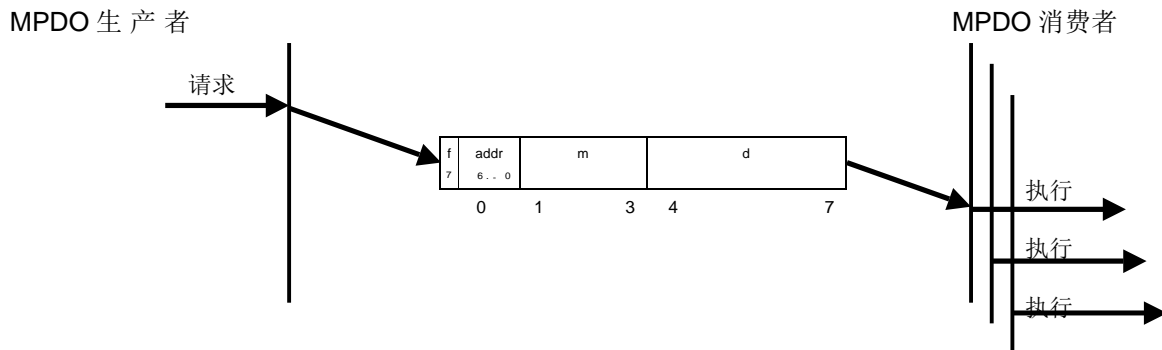


图 19 : 协议 MPDO 写

- **f**: 地址类型
  - 0: 源地址
  - 1: 目的地址
- **addr**: 目的寻址的 MPDO 消费者的或 MPDO 生产者源地址寻址的 node-ID。
  - 0: 在源寻址模式应保留。在目的寻址模式表示所有 CANopen 网络中接收 MPDO 配置为目的寻址模式的设备。
  - 1.. 127: 寻址 CANopen 网络中相同的 node-ID。
- **m**: multiplexer。表示 MPDO 携带过程数据的索引/子索引号，源寻址表示发送 CANopen 设备数据索引和子索引，目的寻址表示接收方数据存放的索引和子索引。
- **d**: 过程数据。数据长度低于 4 字节比较适合 32 位数据。

## 7.2.4 服务数据对象 (SDO)

### 7.2.4.1 简述

SDO 提供了直接访问 CANopen 设备对象字典的入口。这些入口条件可能包括数据的类型和大小。SDO 可用于在客户端和服务端之间传输多个数据集（每个包含任意大小数据块）。客户端通过多路器（对象字典的索引和子索引）来控制传输哪个数据集。数据集的内容定义在对象字典中。

基本上 SDO 作为一个分段序列来传输。先发送一个初始化分段让客户端和服务端各自准备好处理接下来的分段。SDO 也可以在初始化段中包含多至 4 字节的数据集，该机制称为快速 SDO 传输（译注：该机制更常用）。

作为可选功能，SDO 可支持块序列传输，其中每块可由多至 127 段（包含序号和数据）的段序列组成。先发送一个初始化块让客户端和服务端准备好处理接下来的块，并议定每块的分段数。传输完成阶段，客户端和服务端通过校验来验证之前接收到的数据集的正确性。上述方法称为 SDO 快传输，在大数据集传输上比段传输快。

SDO 块上传因为有协议开销使其传输块大于数据本身大小。因此需要合理利用 SDO 正常（分段式）或快速传输模式。SDO 块传输优于其它传输方式的最小数据集大小，取决于客户端在初始化阶段发给服务器的参数阈值。

Go-Back n ARQ 方案在 SDO 块传输中用于每个块的应答：

- SDO 块下载完成后，服务器向客户端承认成功收到最后一个 SDO 块传输分段。执行此操作的服务器也隐含的承认之前的分段接收正常。客户端应开始接下来的 SDO 块传输。此外服务器应在应答内容里明确下一 SDO 块的分段数。
- SDO 块上传完成后，客户端向服务器承认成功收到最后一个 SDO 块传输分段。执行此操作的客户端也隐含承认之前的分段接收正常。服务器应开始接下来的 SDO 块传输。此外客户端应在应答内容里明确下一 SDO 块的分段数。

任何类型的 SDO 传输都由客户端发起。数据字典持有者是服务器。客户端或服务端都可以主动中止传输。

通过在两个 CANopen 设备间建立点对点的通信信道。CANopen 设备可以支持多个 SDO。支持 Server-SDO 的一个作默认项（缺省 SDO）。

SDO 由 SDO 通信参数项描述。其数据类型结构说明见 7.4.8。SDO 通信参数描述了 SSSDO 和 CSSDO 的通信功能。

每个 SDO 通讯参数都是必不可少的。如果只有一个 SSSDO 存在，通信参数可以省略。上述对象描述见 7.4。

## 7.2.4.2 SDO 服务。

### 7.2.4.2.1 简述

SDO 通讯模型是 4.4.3 节所述的客户端 / 服务器模型。

属性：

- SDO 编号: 本地设备的每个用户类型都有一个 SDO 编号[ 1..128 ]
- 用户类型(user type) : 之一 { client , server }
- Mux data type: 其中包含索引和子索引， STRUCTURE OF UNSIGNED16, UNSIGNED8, 指 CANopen 设备对象字典的指定对象索引和子索引号。
- 传输类型(transfer type) :取决于数据传输长度: 快速、多于 4 字节的正常 (分段式) 或块；正常 (分段式) 或块超过 4 字节数据。
- 数据类型(data type) : 取决于索引和子索引对应的对象。

以下 SDO 服务由应用程序请求触发:

- SDO 下载, 其中分为
  - SDO 下载启动
  - SDO 下载分段
- SDO 上传, 其中分为
  - SDO 上传启动
  - SDO 上传分段
- SDO 中止传输

SDO 分段下载和上传的分段工作由通信软件 (译注: 协议栈) 负责。

SDO 快速传输应被支持。大于 4 字节的数据对象传输需要 SDO 分段传输支持。可选的 SDO 块传输是为了较高总线利用率传输大数据集:

- SDO 块下载, 其中分为
  - SDO 块下载启动
  - SDO 下载块
  - SDO 块下载结束
- SDO 块上传, 其中分为
  - SDO 块上传启动
  - SDO 上传块
  - SDO 块上传结束

当使用 SDO 块下载和块上传服务的通信软件负责传输数据分块工作。

SDO 块上传协议支持一个提高传输性能的开关。

使用 SDO 中止传输服务来中止 SDO 块传输。

### 7.2.4.2.2 服务 SDO 下载

客户端使用 SDO 载服务向服务器(持有数据字典)传输数据。数据的索引和子索引、数据集、大小都将通知服务器。该用于此服务的参数被定义在表 7 中。

该服务需要应答。远程返回的结果将会说明请求的成功或失败。一旦失败，原因是确定的。SDO 下载过程至少包括 SDO 下载启动，也许还会有 SDO 段下载服务（数据长度>4 字节）。

表 7：服务 SDO 下载

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
索引	强制	
大小	可选	
数据	强制	
<b>远程结果</b>		<b>强制性</b>
成功		选择
失败		选择
原因		可选的

### 7.2.4.2.3 服务 SDO 下载启动

客户端使用 SDO 下载启动来请求服务器准备数据下载。要下载的数据大小对于服务器来说是可选的（注释：根据索引信息就能清楚）。用于此服务参数被定义在表 8 中。

在多路复用器的数据和设置传输类型显示的服务器。在《性别歧视条例》加速下载、数据的数据集标识的多路复用器和大小显示在服务器上。

表 8：服务 SDO 下载启动

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
索引	强制	
传输类型	强制	
正常	选择	
快速	选择	
大小	可选	
数据	强制	<b>强制性</b>
<b>远程结果</b>		<b>强制</b>
成功		

该服务需要。远程结果将说明请求成功。一旦失败，SDO 中止传输请求应被发起。一旦一个索引 DOMAIN 成功下载，服务会认为要下载的数据集由该索引标识。

#### 7.2.4.2.4 服务 SDO 分段下载

客户端使用 SDO 下载服务传输分段数据给服务器。分段数据及可选的数据大小描述将发给服务器。接下来的参数表明是否还要下载更多分段或者本段已是最后一段。用于此服务的参数被定义在表 9 中。

表 9 : 服务 SDO 分段下载

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
持续性	强制	
更多	选择	
最末	选择	
大小	强制	
数据	强制	<b>强制性</b>
<b>远程结果</b>		强制
成功		

该服务带应答。远程结果参数将表明请求成功。一旦失败,应发起 SDO 中止传输请求。如果成功,服务器接收该分段数据并为下一段做准备。该 SDO 传输应至少包含一个 SDO 分段。

该服务须保证之前已成功进行了带有分段传输类型信息的 SDO 下载启动。

#### 7.2.4.2.5 服务 SDO 上传

客户端通过 SDO 上传服务从服务器(对象字典持有者)获取数据。数据集的索引信息会先发给服务器。用于此服务的参数被定义在表 10 中。

SDO 上传至少包含 SDO 上传启动服务,而 SDO 分段上传服务(数据长度 > 4 个字节)是可选的。

表 10: 服务《性别歧视条例》上传

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
索引	强制	
<b>远程结果</b>		<b>强制性</b>
成功		选择
大小		可选
数据		强制
失败		选择
原因		可选

该服务带应答。远程结果参数将会显示请求成功或失败。一旦失败,其原因作为可选项被回复。如果成功,数据及其大小作为确认返回。

#### 7.2.4.2.6 服务 SDO 上传启动

客户端通过 SDO 上传启动服务让服务器做好数据上传准备。数据集的索引信息会发给服务器,该服务所需参数在表 11 中定义。

该服务带应答。远程结果参数将表明请求成功。一旦失败,SDO 中止传输请求会被被执行。如果成功了,数据和大小作为应答返回。一旦 SDO 上传启动成功,该服务中标识所要上传数据集的索引及对应的应答数据将得到确认。

表 11: 服务 SDO 上传启动

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
索引	强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制
索引		强制
传输类型		强制
正常		选择
快速		选择
尺寸		可选
数据		强制

### 7.2.4.2.7 服务 SDO 分段上传

客户端通过 SDO 分段上传服务向服务器请求下一个数据分段。接下来的参数向客户端表明是否还有更多分段或此为最后一个上传分段 SDO 分段上传服务至少应包含一个分段。用于此服务的参数被定义在表 12 中。

该服务已得到确认。远程结果参数将指示成功的请求。在出现故障的情况下，SDO 中止传输请求应发起的。在个案的成功，这部分的数据和选择其大小已被确认。

成功的《性别歧视条例》上传启动服务与正常（分段）转移类型应为之前执行此项服务。

表 12：服务 SDO 分段上传

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制
继续		强制
更多		选择
最末		选择
大小		强制
数据		强制

### 7.2.4.2.8 服务 SDO 块下载

客户端通过块下载向服务器（对象字典持有者）发送数据。数据、数据集索引和可选的数据大小会被发送给服务器。用于此服务的参数被定义在表 13 中。

该服务带应答。远程结果参数将表明请求成功或失败。一旦失败，可选择在应答中包含原因。

表 13 : 服务 SDO 块下载

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
索引	强制	
大小	可选	
数据	强制	
<b>远程结果</b>		<b>强制性</b>
成功		选择
失败		选择
原因		可选

#### 7.2.4.2.9 服务 SDO 块下载启动

客户端通过 SDO 块下载启动服务通知服务器为下载做准备。用于此服务的参数被定义在表 14。数据集的索引和可选的数据大小信息会发给服务器。

SDO 块下载结束后，客户端和服务器都有能力和需求去校验数据完整性。

表 14 : 服务 SDO 块下载启动

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
CRC 能力	强制	
是	选择	
否	选择	
索引	强制	
大小	可选	
<b>远程结果</b>		<b>强制性</b>
成功		强制
CRC 能力		强制
是		选择
否		选择
blksize		强制

该服务带应答。远程结果参数将表明请求成功，服务器会收到每块的分段数和校验信息。一旦失败，SDO 中止传输服务将被发起。



## 7.2.4.2.10 服务 SDO 块下载子块

客户端通过子块下载服务向服务器发送下一数据块。数据块由一个段列组成。每段包含相应的分段数据和从 1 开始到 blksize 的分段序号。参数 blksize 由客户端和服务器之间通过 SDO 块下载启动服务协商好，并且可以由服务器在块应答里更改。接下来的参数说明服务器会继续接收子块还是结束。用于此服务的参数被定义在表 15。

表 15：服务 SDO 块下载子块

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
继续	强制	
更多	选择	
最末	选择	
数据	强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制
Ackseq		强制
Blksize		强制

该服务带应答。远程结果参数将表明请求成功。如果成功，ackseq 参数表达服务器成功接收的最后一段的段号。如果和客户端发送的最后一段的段号对应不上，则客户端会在下一次块传输时重传所有被服务器丢弃的段。如果出现致命错误，SDO 中断传输服务会被发起。如果服务器收到并确认所有数据段则进行下一块的接收准备。SDO 块下载应至少包含一个 SDO 块下载子块服务。

此服务之前需要一次成功的 SDO 块下载启动服务。

### 7.2.4.2.11 服务 SDO 块下载结束

客户端通过 SDO 块下载结束服务来结束与服务器的合作。最后传给服务器的分段字节里面并不包含有效数据。用于此服务的参数被定义在表 16 中。

客户端和服务器都有能力和需求对传输的数据进行校验。

表 16: 服务 SDO 块下载结束

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
Valid_data	强制	
校验和	条件的， 如果议定则强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制

该服务带应答。远程结果参数将指明请求成功 (客户端和服务器之间议定的话需校验和相匹配) 下载数据集有效。一旦失败, SDO 中止传输服务将被发起。

### 7.2.4.2.12 服务 SDO 块上传

客户端通过服务 SDO 块上传服务向服务器 (持有数据字典) 请求数据。请求的数据集索引会发给服务器。用于此服务的参数被定义在表 17 中。

该服务带应答。远程结果参数将会表明请求成功或失败。如果失败了, 应答信息里可包含失败原因。如果成功了, 将应答数据及其可选的尺寸。

表 17: 服务 SDO 块上传

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
索引	强制	
<b>远程结果</b>		<b>强制性</b>
成功		选择
尺寸		可选的
数据		强制
失败		选择
原因		可选的

## 7.2.4.2.13 服务 SDO 块上传 启动

客户端通过该服务通知服务器准备好数据上传。启动上传的索引和有能力的段数会报给服务器。用于此服务的参数被定义在表 18 中。

一项阈值切换协议会发给服务器。如果上传的字节数小于或等于该值，服务器会以 7.2.4.2.5 描述的 SDO 上传服务应答。客户端和服务器都有保证传输完整的校验能力和需求。可选的上传数据尺寸信息也会发给客户端。

表 18: 服务 SDO 块上传启动

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
Blksize	强制	
CRC 能力	强制	
是	选择	
否	选择	
索引	强制	
阈值	强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制
CRC 能力		强制
是		选择
否		选择
尺寸		可选的

该服务带应答。如果失败,SDO 传输中止服务应发起的。如果成功,可选的数据尺寸信息会发给客户端。

## 7.2.4.2.14 服务 SDO 块上传子块

这项服务是由之前的 SDO 上传启动服务或前一次的 SDO 块上传子块服务的客户端发起。服务器使用该服务将分块数据发给客户端。该块数据以分段序列的形式传输。每段包含相应的分段数据和从 1 开始到 blksize 的分段序号。参数 blksize 由客户端和服务端之间通过 SDO 块上传启动服务协商好，并且可以由客户端在块应答里更改。接下来的参数说明客户端会继续接收子块还是结束。用于此服务的参数被定义在表 19 中。

表 19: 服务 SDO 块上传子块

参数	要求 / 说明	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
继续	强制	
更多	选择	
最后	选择	
数据	强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制
Ackseq		强制
Blksize		强制

该服务带应答。远程结果参数将表明请求成功。如果成功，ackseq 参数表达客户端成功接收的最后一段的段号。如果和服务器发送的最后一段的段号对应不上，则服务器会在下一次块传输时重传所有被客户端丢弃的段。如果出现致命错误，SDO 中断传输服务会被发起。如果客户端收到并确认所有数据段则进行下一块的接收准备。SDO 块上传应至少包含一个 SDO 块上传子块服务。

此服务之前需要一次成功的 SDO 块上传启动服务。

### 7.2.4.2.15 服务 SDO 块上传 结束

服务器通过 SDO 块上传结束服务来结束与客户端的合作。最后传给客户端的分段字节里面并不包含有效数据。用于此服务的参数被定义在表 20 中。

客户端和服务器都有能力和需求对传输的数据进行校验。

表 20: 服务 SDO 块上传结束

参数	要求 / 执行	响应 / 确认
<b>要素</b>	<b>强制性</b>	
SDO 编号	强制	
Valid_data	强制	
校验和	议定则强制	
<b>远程结果</b>		<b>强制性</b>
成功		强制

该服务带确认。远程结果参数将指明请求成功 (客户端和服务端之间议定的话需校验和相匹配) 下载数据集有效。一旦失败, SDO 中止传输服务将被发起。

### 7.2.4.2.16 服务 SDO 中止传输

该服务能够中止 SDO 上传和下载服务, 并列原因。该服务无应答。客户端和服务端在任何时候都可以发起该服务。如果 SDO 客户端得到了一个明显确认, 就会向服务器发送中止服务的确认, 用于此服务的参数被定义在表 21 。

表 21: 服务 SDO 中止传输

参数	要求 / 说明
<b>要素</b>	<b>强制性</b>
SDO 编号	强制
索引	强制
原因	强制

## 7.2.4.3 SDO 协议

### 7.2.4.3.1 简述

SDO 一共定义了六种带应答的服务（SDO 下载，SDO 上传，SDO 上传启动，SDO 下载启动，SDO 分段下载和 SDO 分段上传）和一种不带应答的服务（SDO 中止传输），它们共同作用来进行 SDO 正常（分段）传输和加速传输。

SDO 一共定义了八种带应答的服务（SDO 块下载，SDO 块上传，SDO 块上传启动，SDO 块下载启动，SDO 块下载子块，SDO 块上传子块，SDO 块上传结束和 SDO 块下载结束）和一种不带应答的服务（SDO 中止传输），它们共同作用来进行 SDO 块传输。

### 7.2.4.3.2 协议 SDO 下载

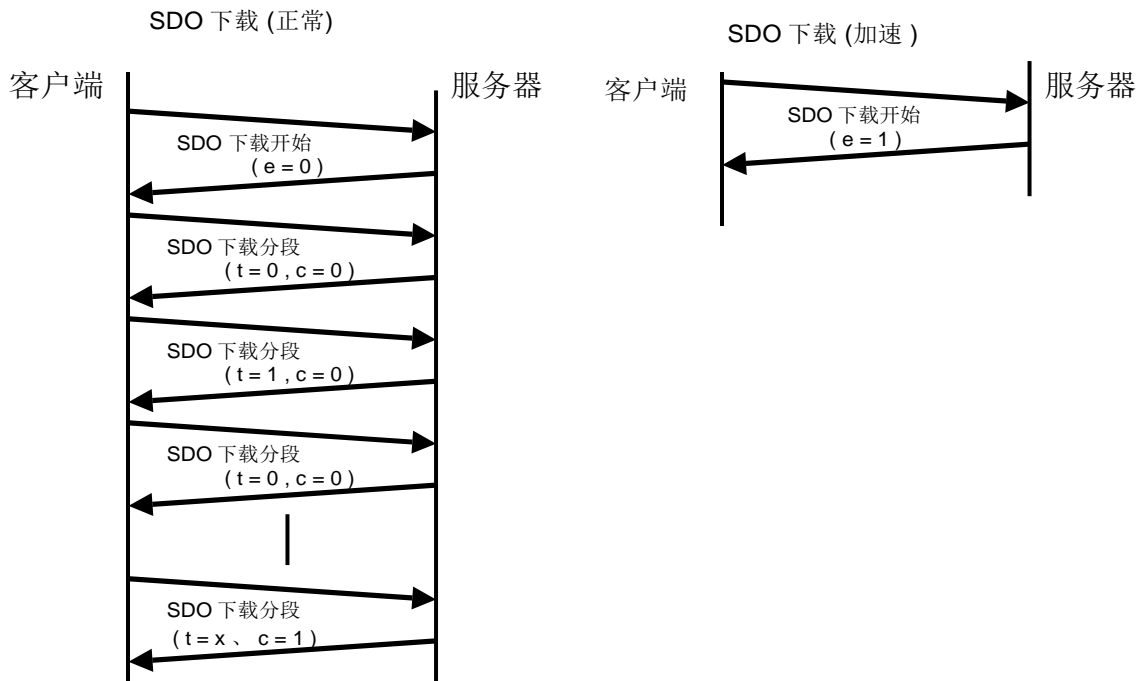


图 20：协议 SDO 下载

该协议（如图 20）为 SDO 下载服务，由 SDO 下载启动服务开始，接着是 0 或多个 SDO 分段下载服务组成的序列。该序列的终止条件如下：

- SDO 下载启动请求/执行对应返回一个应答/确认，其 e-bit 被置 1，表明成功完成一次加速下载。
- SDO 分段下载响应/确认帧 c-bit 被置 1，表示成功的完成了一次正常下载。
- SDO 中止传输请求/执行，表明下载过程未成功。
- 新的 SDO 下载启动请求/执行，说明下载序列还没完成就新开一个下载序列。

如果下载过程中，相邻两个分段的翻转位没有变化。后一段将被忽略。如果该种错误被回报到应用程序，应用程序可以决定是否需要中止下载。

## 7.2.4.3.3 协议 SDO 下载启动

SDO 下载启动协议执行过程如图 21 所示。

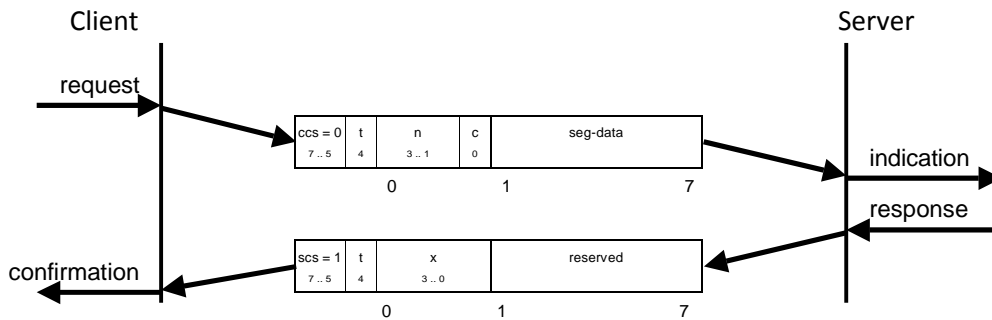


- **ccs** : 客户端命令说明符
  - 1: 下载启动请求
- **scs**: 服务器命令说明符
  - 3: 下载启动的响应
- **n**: 有效仅当  $e = 1, s = 1$ , 否则为 0。如果有效的话, 它表示 **d** 中不带数据的字节数。字节 [8-n, 7] 不包含数据。
- **e**: 传输类型
  - 0: 正常的传输
  - 1: 快速传输
- **S**: 尺寸指示
  - 0: 数据集的大小不指明
  - 1: 数据集的大小明确指出
    - **m**: 索引。表示传输数据的索引/子索引号。
- **d**: 数据
  - $e = 0, s = 0$ : **d** 预留以供将来使用。
  - $e = 0, s = 1$ : **d** 包含下载字节。
    - Byte 4 为 LSB, Byte 7 为 MSB。
  - $e = 1, s = 1$ : **d** 包含下载的长度  $4-n$  的数据, 编码取决于索引和子索引对应数据的类型。
  - $e = 1, s = 0$ : **d** 包含下载的未指定字节数的数据。
- **X**: 不使用、始终为 0
- **保留**: 为将来保留, 始终为 0

图 21: 协议 SDO 下载启动

## 7.2.4.3.4 协议 SDO 分段下载

该协议定义在图 22 中。



- **ccs** : 客户端命令说明  
0: 分段下载请求
- **scs**: 服务器命令说明  
1: 分段下载的响应
- **Seg-data**: 最多 7 字节的分段数据。编码取决于索引和子索引对应的数据类型。
- **n** : Seg-data 中不包含分段数据的字节数。[ 8-n, 7 ]为不包含分段数据的空字节。如果  $n = 0$  则 1 到 7 字节填满。  
注意: 如果该字节数即是启动服务中所明确的长度, 则该分段包含所有需要传输的数据。
- **c**: 表明是否还有要下载的分段。  
0 更多的分段需要被下载  
1: 没有更多的分段需要被下载
- **t**: 翻转位。每个下载分段中翻转一次。第一个分段翻转位置 0。该位应与对应的应答一致。
- **X**: 不使用、始终为 0
- **保留** : 为将来保留, 始终为 0

图 22 : 协议 SDO 分段下载



## 7.2.4.3.5 协议 SDO 上传

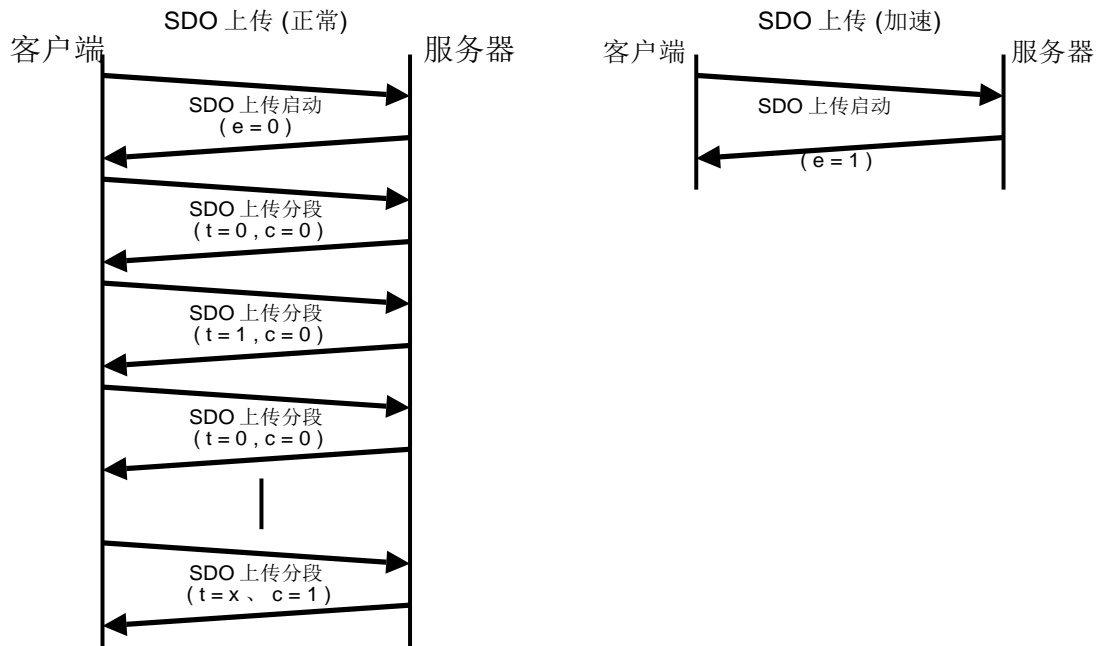


图 23: 协议 SDO 上传

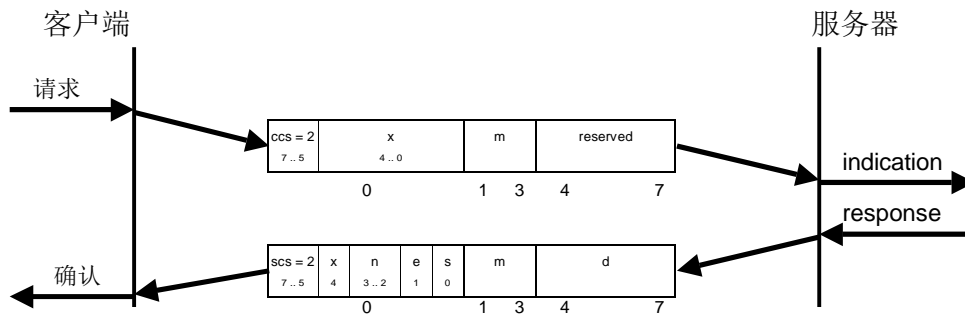
该协议 (如图 23) 为 SDO 上传服务, 由 SDO 上传启动服务开始, 接着是 0 或多个 SDO 分段上传服务组成的序列。该序列的终止条件如下:

- SDO 上传启动请求/执行对应返回一个应答/确认, 其 e-bit 被置 1, 表明成功完成一次加速上传。
- SDO 分段上传的响应 / 确认帧 c-bit 被置 1, 表示成功的完成了一次正常上传。
- SDO 中止传输请求 / 执行, 表明上传过程未成功。
- 新的 SDO 上传启动请求/执行, 说明上传序列还没完成就新开一个下载序列。

如果上传过程中, 相邻两个分段的翻转位没有变化。后一分段将被忽略。如果该种错误被回报到应用程序, 应用程序可以决定是否需要中止上传。

## 7.2.4.3.6 协议 SDO 上传启动

SDO 上传启动协议执行过程如图 24 所示。

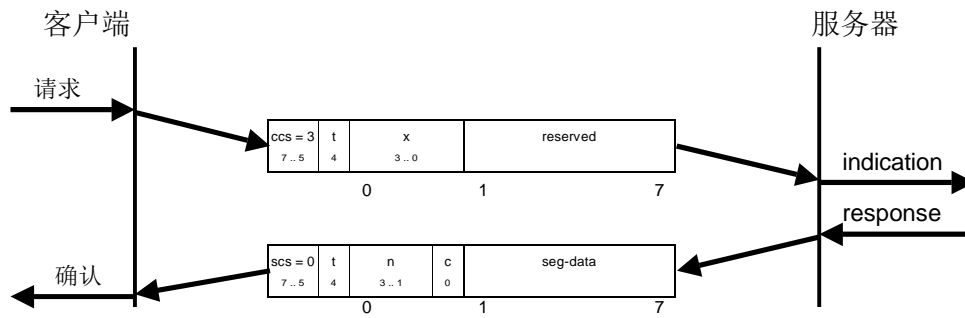


- **ccs** : 客户端命令说明
  - 2 : 上传开始请求
- **scs** : 服务器命令说明
  - 2 : 上传开始响应
- **n** : 仅当  $e = 1, s = 1$  时有效, 否则为 0。如果有效的话, 表示 **d** 中不包含数据的字节数。字节 [8-n, 7] 中不带数据。
- **e** : 传输类型
  - 0 : 常的传输
  - 1 : 加速传输
- **S** : 尺寸指示器
  - 0 : 数据集的大小不明确
  - 1 : 数据集的大小明确说明
- **m** : 索引。表示 SDO 传输数据对应的索引/子索引号。
- **d** : 数据
  - $e = 0, s = 0$  : 为将来预留。
  - $e = 0, s = 1$  : **d** 包含上传的字节数。  
Byte 4 为 lsb, Byte 7 为 msb。
  - $e = 1, s = 1$  : **d** 中包含长度 4-n 的上传数据,  
编码取决于索引和子索引引用数据的类型。
  - $e = 1, s = 0$  : **d** 包含未指定字节长度的数据上传。
- **X** : 不使用, 始终为 0
- **保留** : 为将来保留, 始终为 0

图 24 : 协议 SDO 上传启动

## 7.2.4.3.7 协议 SDO 分段上传

该协议定义在图 25 中。



- **ccs** : 客户端命令说明  
3:分段上传请求
- **scs**: 服务器命令说明  
0:分段上传响应
- **t:翻转位**。每个上传分段中翻转一次。第一个分段翻转位置 0。该位应与对应的应答一致。
- **C** : 指示是否还有更多的上传分段。  
0: 更多上传分段  
1: 最后一段
- **seg-data**: 最多 7 字节的上传分段数据。编码取决于索引和子索引指向的数据的类型。
- **n** : 不包含分段数据的字节数。字节[8-n, 7] 不包含分段数据。如果 n = 0, 字节 1 到 7 应填满分段数据。  
注意: 如果该字节数即是启动服务中所明确的长度, 则该分段包含所有需要传输的数据。
- **X** : 不使用, 始终为 0
- **保留** : 为将来保留, 始终为 0

图 25: 协议 SDO 分段上传

## 7.2.4.3.8 协议 SDO 块下载

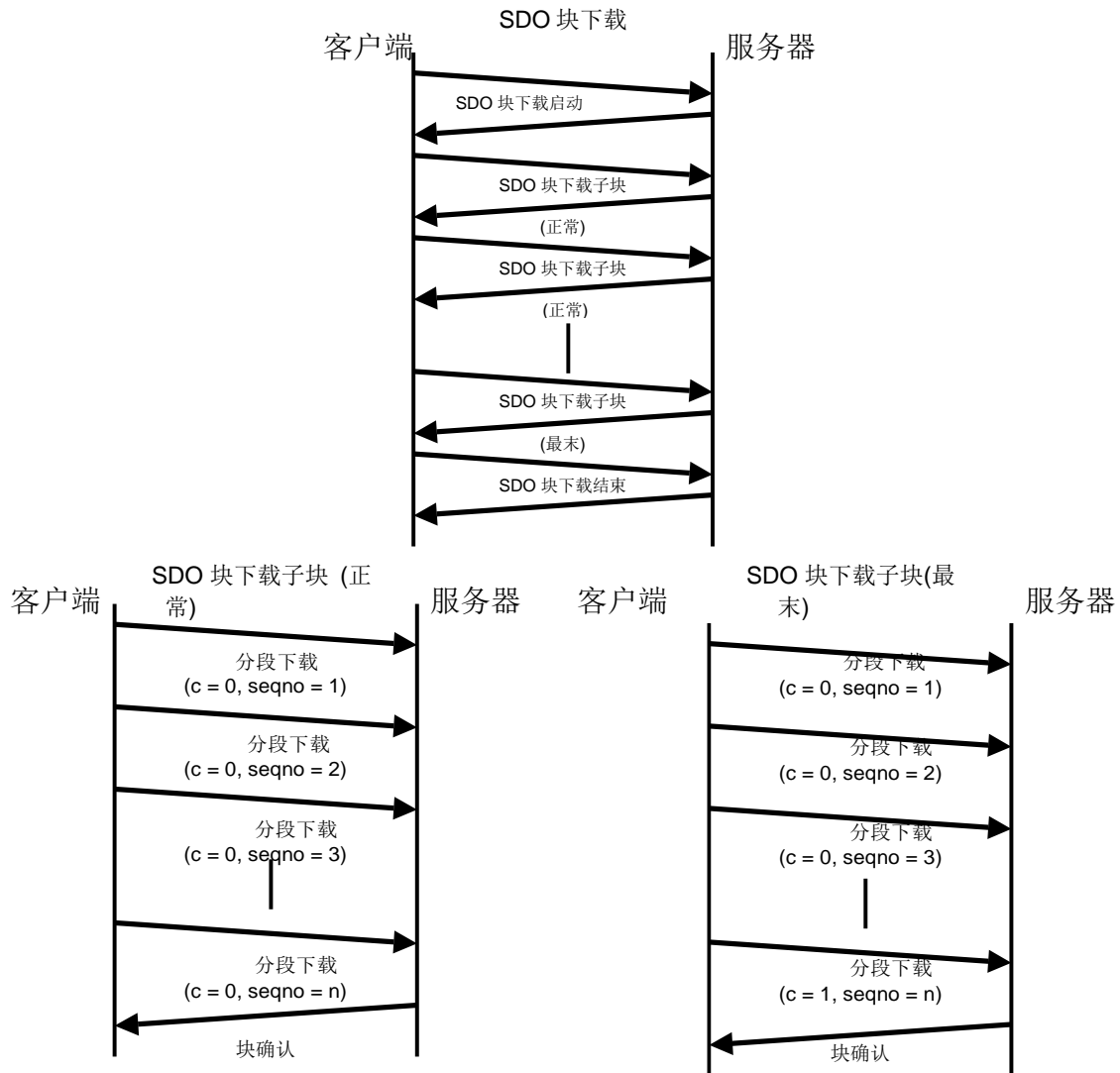


图 26:协议 SDO 块下载

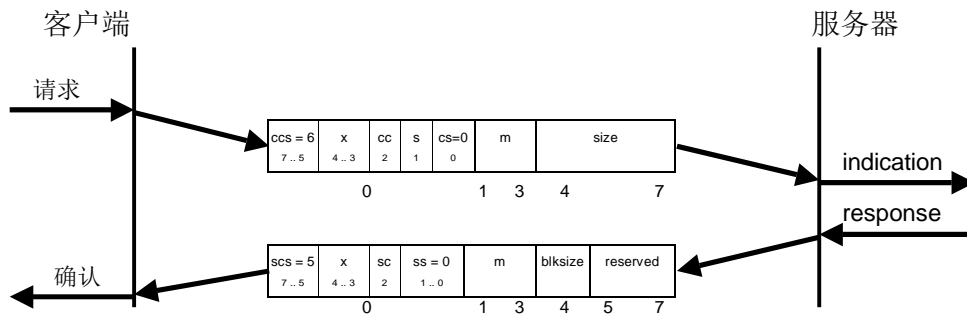
该协议定义如图 26，以块下载启动服务开始，然后是一个 SDO 子块下载服务序列。其序列终结条件如下：

- 块分段内的 c-bit 为 1 ,表明序列顺利完成。
- SDO 中止传输请求 / 指示，表明序列下载过程不成功。

块下载服务以块下载结束服务终止。客户端和服务端都有 CRC 校验能力，如果服务器校验接收数据与客户端的校验值不一致，则发起 SDO 中止传输。

## 7.2.4.3.9 协议 SDO 块下载启动

该协议定义于图 27。

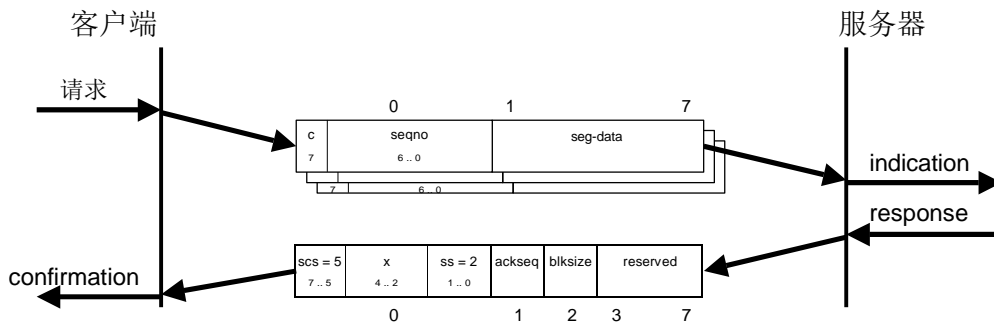


- **ccs:** 客户端命令说明
- 6: 块下载
- **scs:** 服务器命令说明
- 5: 块下载
- **s:** 尺寸说明
- 0: 不指明数据集大小
- 1: 明确数据集大小
- **cs:** 客户端子命令
- 0: 下载启动请求
- **ss:** 服务器子命令
- 0: 下载启动应答
- **cc:** 客户端 CRC 支持情况
- cc = 0: 不支持
- cc = 1: 支持
- **sc:** 服务器 CRC 支持情况
- sc = 0: 不支持
- sc = 1: 支持
- **m:** 索引. SDO 所传输数据索引/子索引.
- **size:** 下载字节数
- s = 0: 为将来保留,总是 0
- s = 1: size 表达要下载的字节数 Byte 4 为 LSB byte 7 为 MSB
- **blksize:** 每块的段数,  $0 < \text{blksize} < 128$
- **X:** 未使用, 总为 0
- **reserved:** 为将来预留, 总为 0

图 27 : 协议 SDO 块下载启动

## 7.2.4.3.10 协议 SDO 块下载子块

该协议定义于图 28。



**scs:** 服务器命令说明

5: 块下载

**ss:** 服务器子命令

2: 块下载应答

- **c:** 更多分段需要下载的标志

0: 更多

1: 没有了，进入块下载结束阶段。

**seqno:** 分段序号  $0 < \text{seqno} < 128$ .

**seg-data:** 至多 7 字节的分段数据。

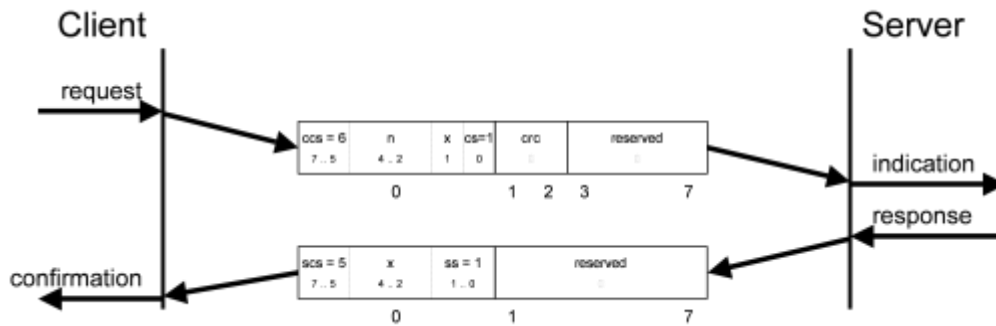
**ackseq:** 成功接收的最近一块的最近一段的序列号。如果被服务器置 0，表示客户端发送的序列号为 1 的分段未成功接收，所有分段需要重发。（译注：不确定）

- **blksize:** 每块的段数，  $0 < \text{blksize} < 128$
- **X:** 未使用，总为 0
- **reserved:** 预留，总为 0

图 28: 协议 SDO 块下载子块

## 7.2.4.3.11 协议 SDO 块下载结束

该协议定义于图 29。



**ccs:** 客户端命令说明

6: 块下载

**scs:** 服务器命令说明

5: 块下载

**cs:** 客户端子命令

1: 块下载结束请求

**ss:**服务器子命令

1: 块下载结束请求应答

**n:**最近一块的最近一段不包含数据的字节数. Bytes [8-n, 7] 不包含分段数据.

**crc:** 数据集的 16 位循环冗余校验(CRC)。CRC 算法见 7.2.4.3.16. 仅当 SDO 块下载启动中 cc 和 sc 被置 1 时 CRC 有效，否则 CRC 置 0.

**X:** 未使用, 总为 0

**reserved:** 预留, 总为 0

图 29:协议 SDO 块下载结束

## 7.2.4.3.12 协议 PDO 块上传

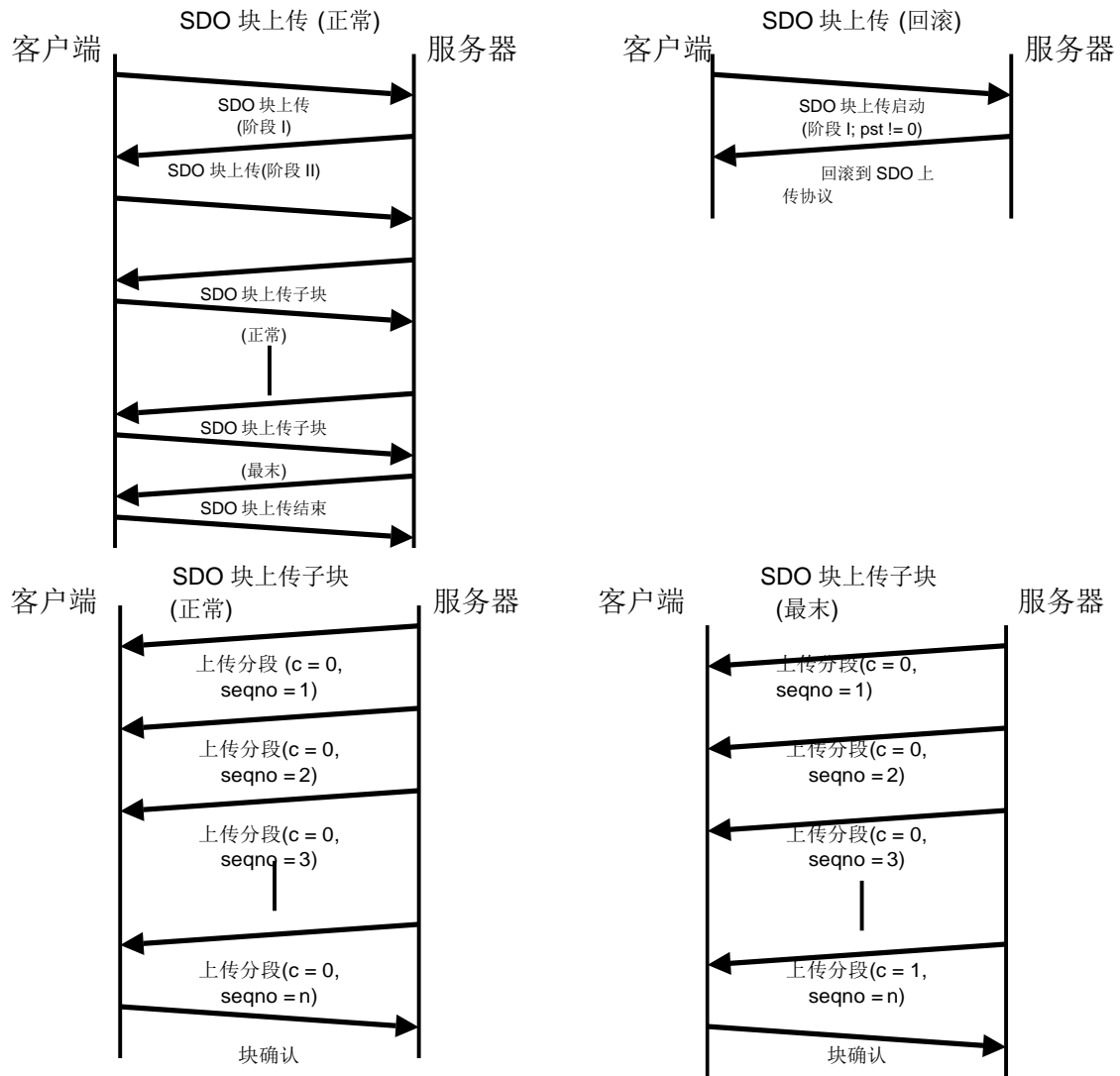


图 30: 协议 SDO 块上传

该协议定义如图 30，以块上传启动服务开始，客户端发给服务器使用 SDO 块上传服务代替 SDO 上传服务所需最小字节数阈值。如果数据集字节数小于等于该阈值则使用常规的 SDO 上传服务。

另一方面，块上传是以 SDO 块上传子块序列形式呈现，其序列终结条件如下：

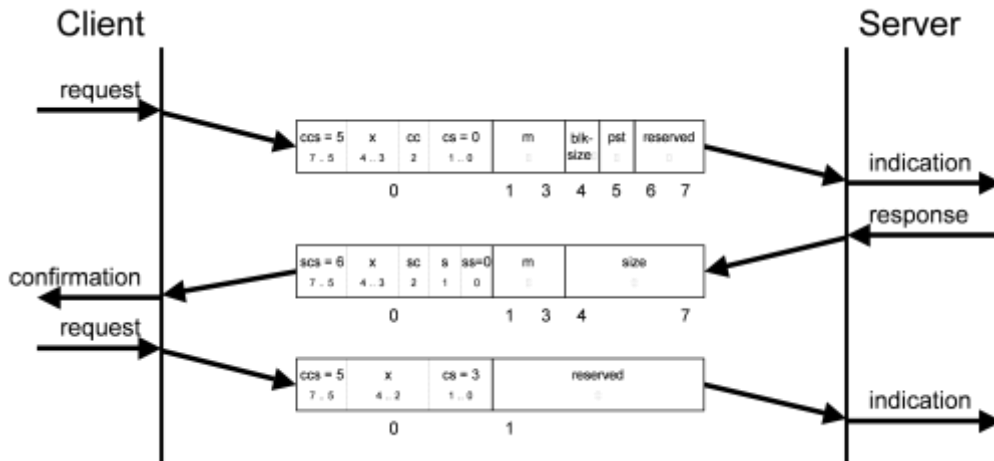
- 块上传分段内的 c-bit 为 1，表明序列顺利完成。
- SDO 中止传输请求 / 指示，表明序列下载过程不成功。

块下载服务以块下载结束服务终止。客户端和服务器都有 CRC 校验能力，如果客户端校验接收数据与服务器的校验值不一致，则发起 SDO 中止传输。



## 7.2.4.3.13 协议 SDO 块上传 启动

该协议定义于图 31。如果客户端请求的数据集字节数小于等于其发送的第一个请求中的协议切换阈值，则启用定义于 7.2.4.3.5 的常规 SDO 上传服务。



- **ccs:** 客户端命令说明

5: 块上传

- **scs:** 服务器命令说明

6: 块上传

- **cs:** 客户端子命令

0: 启动上传请求

3: 开始上传

- **ss:** 服务器子命令

0: 启动上传响应

- **m:** 索引。它代表传输的 SDO 数据对应的索引/子索引。

- **cc:** 客户端是否支持 CRC

`cc = 0`: 客户端不支持生成数据 CRC

`cc = 1`: 客户端支持生成数据 CRC

- **sc:** 服务器是否支持 CRC

`sc = 0`: 服务器不支持生成数据 CRC

`sc = 1`: 服务器支持生成数据 CRC

- **pst:** 根据切换阈值确定数据传输协议

`pst = 0`: 不允许切换传输协议。

`pst > 0`: 如果数据大小小于或等于 `pst`, 服务器将以 7.2.4.3.5 所述 SDO 上传服务来响应。

- **s:** 尺寸指示

0: 数据集的大小不指明

1: 数据集的大小明确。

- **size:** 以字节为单位的上传数据大小

`s = 0`: 保留, 始终为 0

`s = 1`: 上传 (译注: 原文有误, 写成了 downloaded) 的数据字节

数 Byte4 为 lsb, Byte7 为 msb

- **blksize:** 每块分段数  $0 < \text{blksize} < 128$ 。

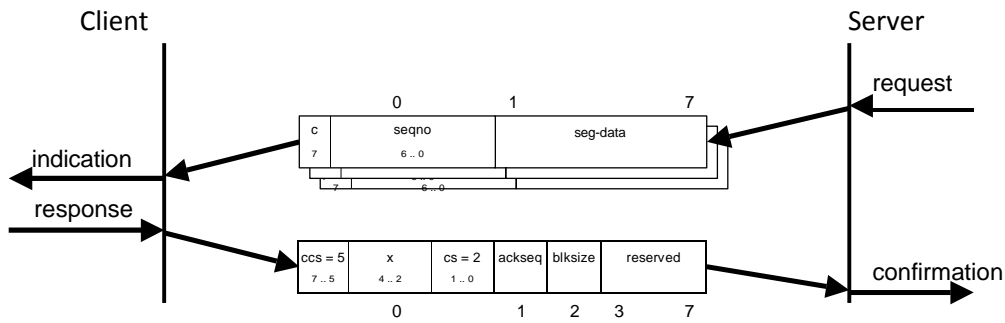
- **X:** 不使用, 始终为 0

- **reserved:** 为将来保留, 始终为 0

图 31: 协议 SDO 块上传启动

## 7.2.4.3.14 协议的 SDO 块上传子块

该协议定义于图 32。

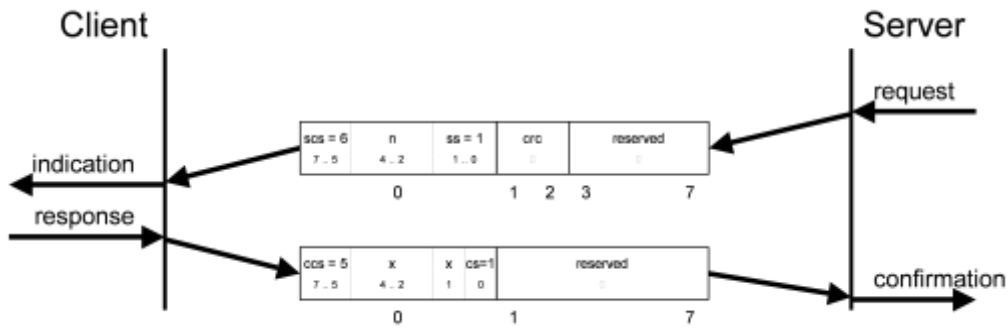


- **ccs** : 客户端命令说明  
5: 块上传
- **cs** : 客户端子命令  
2: 块上传响应
- **c** : 是否还有更多上传(译注: 原文为 downloaded)分段  
0: 更多上传分段  
1: 没有更多的上传分段, 进入“结束块上传”的阶段。
- **seqno**: 段序列编号  $0 < \text{seqno} < 128$  。
- **seg-data**: 多至 7 字节的上传分段数据。
- **ackseq**: 成功接收的最近一块的最近一段的序列号。如果被客户端置 0, 表示服务器发送的序列号为 1 的分段未成功接收, 所有分段需要重发。 (译注: 不确定)
- **blksize**: 每块的段数,  $0 < \text{blksize} < 128$
- **X**: 不使用、始终为 0
- **reserved**: 为将来保留, 始终为 0

图 32: 协议 SDO 块上传子块

### 7.2.4.3.15 协议 SDO 块上传 结束

该协议定义于图 33。



**ccs** : 客户端命令说明

5: 块上传

- **scs** : 服务器命令说明

6:块上传

- **cs** : 客户端子命令

1: 结束块上传请求

- **ss** : 服务器子命令

1: 结束块上传响应

- **n**:最近一块的最近一段不包含数据的字节数. Bytes [8-n, 7] 不包含分段数据.

- **crc**: 数据集的 16 位循环冗余校验(CRC)。CRC 算法见 7.2.4.3.16. 仅当 SDO 块上传启动中 cc 和 sc 被置 1 时 CRC 有效, 否则 CRC 置 0.

**X**: 未使用,总为 0

**reserved**: 预留, 总为 0

图 33: 协议 SDO 块上传结束

### 7.2.4.3.16 验证 SDO 块传输的 CRC 算法

要验证交换数据的正确性, 客户端和服务器须在 SDO 块上传和下载服务的结束协议中体现 CRC 计算。

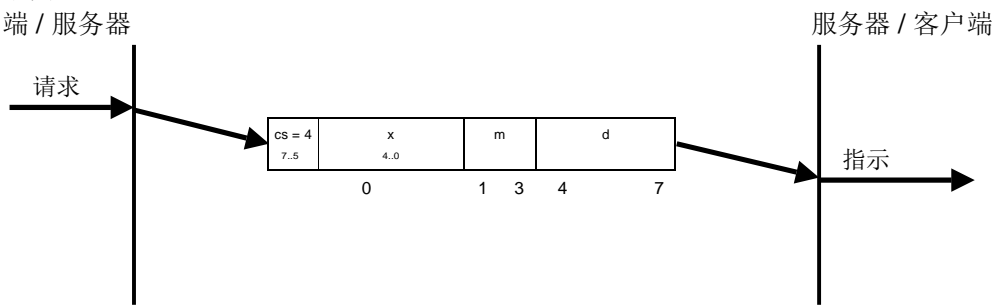
CRC 校验应具有以下参数:

- CRC 多项式:  $X^{16} + X^{12} + X^5 + 1$
- CRC 的宽度: 16 位
- 初始值: 0000<sub>h</sub>
- CRC 检查 (123456789 的 CRC 结果): 31C3<sub>H</sub>

## 7.2.4.3.17 协议 SDO 中止传输

该协议定义于图 34。

客户端 / 服务器



- **cs** : 命令说明符  
4: 中止传输请求
- **x** : 不使用、始终为 0
- **m** : 索引。SDO 索引和子索引。
- **d** : 包含 4 字节的中止原因代码。

图 34:协议 SDO 中止传输

中止代码定义于表 22 被编码为 UNSIGNED32 值。

表 22: SDO 中止代码

中止代码	描述
0503 0000 <sub>h</sub>	翻转位未变化
0504 0000 <sub>h</sub>	SDO 协议超时
0504 0001 <sub>h</sub>	客户端/服务器命令说明无效或未知
0504 0002 <sub>h</sub>	无效的块大小 (仅块模式)
0504 0003 <sub>h</sub>	无效的序列号 (仅块模式)
0504 <sub>h</sub> 0004 号	CRC 错误 (仅块模式)
0504 0005 <sub>h</sub>	内存不足
0601 0000 <sub>h</sub>	不支持的访问的对象
0601 0001 <sub>h</sub>	试图读取只写的对象
0601 0002 <sub>h</sub>	尝试写入只读对象
0602 0000 <sub>h</sub>	对象不存在于对象字典
0604 0041 <sub>h</sub>	对象不能被映射进 PDO。
0604 0042 <sub>h</sub>	对象长度和数量超出 PDO 的长度
0604 0043 <sub>h</sub>	常规参数不兼容的原因
0604 0047 <sub>h</sub>	设备内部不兼容
0606 0000 <sub>h</sub>	硬件错误导致的访问失败
0607 0010 <sub>h</sub>	数据类型不匹配, 服务长度参数不匹配。
0607 0012 <sub>h</sub>	数据类型不匹配, 服务长度参数太大
0607 0013 <sub>h</sub>	数据类型不匹配, 服务长度参数太小
0609 0011 <sub>h</sub>	子索引不存在
0609 0030 <sub>h</sub>	无效的参数值 (仅下载)

中止代码	描述
0609 0031 <sub>n</sub>	写入参数值太高 ( 仅下载 ) 。
0609 0032 <sub>n</sub>	写入参数值太低 ( 仅下载 ) 。
0609 0036 <sub>n</sub>	最大值小于最小值
060A 0023 <sub>n</sub>	资源不可用 : SDO 连接
0800 0000 <sub>n</sub>	常规错误
0800 0020 <sub>n</sub>	数据不能传输或保存到应用程序。
0800 0021 <sub>n</sub>	数据不能传输或保存应用程序, 由于本地的控制。
0800 0022 <sub>n</sub>	数据不能传输或保存到应用程序, 因为设备当前状态。
0800 0023 <sub>n</sub>	对象字典的动态生成失败或无对象字典的存在 ( 例如对象字典是从文件生成, 而由于文件的错误生成失败)
0800 0024 <sub>n</sub>	无可用的数据

未列出的中止代码应保留。

## 7.2.5 同步对象 (SYNC)

### 7.2.5.1 简述

同步生产者定期广播同步对象。SYNC 提供基本的网络同步机制。SYNC 间的时间周期由标准通信循环周期参数定义(见 7.5.2.6),可由配置工具在 CANopen 设备引导过程中写入。由于有优先级高于 SYNC 的通信, SYNC 生产者在发送报文时可能会产生延时抖动。SYNC 消费者可以使用通信循环周期的制造商规范。

使用可选的计数器参数,可以在当前 SYNC 循环和 PDO 通信之间建立显式关系(PDO 通信参数: SYNC 启动值, 详见 7.5.2.37)。

为了保证及时访问网络, SYNC 的 CAN-ID 优先级较高(见 7.5.2.5)。CANopen 设备通过同步操作来同步自身与同步对象生产者。详细内容需要具体应用来实现,不属于本文范围。

### 7.2.5.2 SYNC 服务

#### 7.2.5.2.1 简述

同步传输遵守生产者/消费者推送模型见 4.4.4。该服务无应答。

属性:

- 用户类型: 之一 { 消费者、生产者 }
- 数据类型: UNSIGNED8

#### 7.2.5.2.2 服务同步写

推送模型适用于同步写入服务。SYNC 消费者数量 可以是零或更多,但 SYNC 生产者只有一个。用于定义此服务的参数见表 23。

通过该服务, SYNC 生产者发送触发事件给消费者(们)。

表 23: 服务同步写

参数	要求 / 说明
要素 计数器	强制性 可选

该服务无确认。可选的参数计数器应每次传输增 1 的计数器参数是可选的。最大值是当前的同步计数溢出值(见 7.5.2.22)。一旦达到最大值,计数器在下一次传输时置 1。在 NMT boot-up 服务后计数器初值应置 1。该值的计数器将被重置为 1。如果 CANopen 设备的状态由 NMT 停止(stopped)进入 NMT 预操作(pre-operational),计数器置应复位置 1。

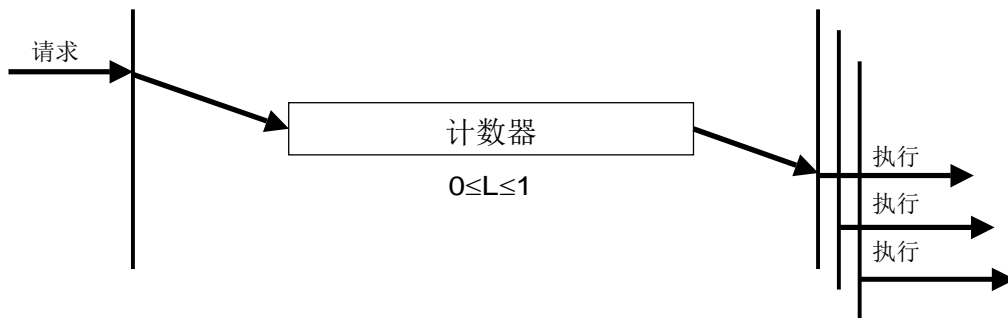
## 7.2.5.3 同步协议

### 7.2.5.3.1 协议同步写

此协议定义如图 35。

SYNC 生产者

SYNC 消费者



- 计数器: 1 字节计数器

图 35: 协议同步写

## 7.2.6 时间戳对象 (TIME)

### 7.2.6.1 简述

TIME 生产者广播时间戳对象。TIME 提供了简单的网络时钟。由于高于 TIME 的通信对象存在，TIME 的发送可能会产生延迟抖动。

为了保证及时地访问网络，TIME 的 CAN-ID 较高 (见 7.5.2.15)。CANopen 设备通过时间戳对象生产者的 TIME 对象校准本地时间。详细机制不属于本规范的范围。

### 7.2.6.2 TIME 服务

#### 7.2.6.2.1 简述

时间戳对象遵守生产者/消费者模型，见 4.4.4。该服务无应答。

属性：

- 用户类型：                之一 { 消费者、生产者 }
- 数据类型：                TIME\_OF\_DAY

#### 7.2.6.2.2 服务 TIME 写

推送模型适用于 TIME 写。有零或更多的消费者，但生产者只有一个。用于定义此服务的参数列于表 24。通过这一服务的生产者的时间发送当前的时间到消费者。

表 24: 服务 TIME 写

参数	要求 / 说明
要素 数据	强制性 强制

该服务无应答。

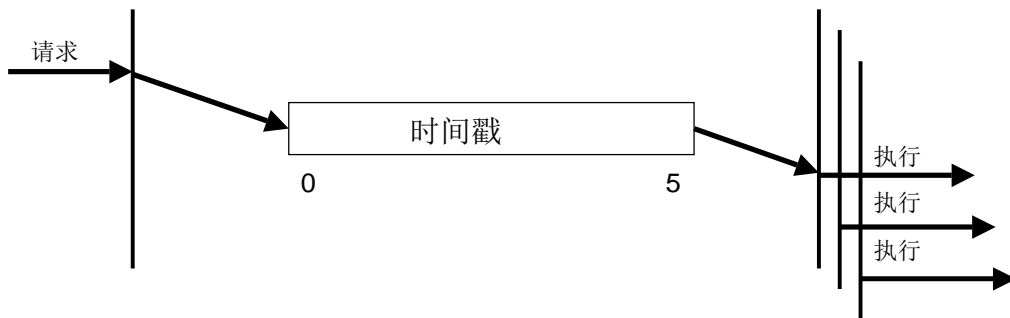
## 7.2.6.3 TIME 协议

### 7.2.6.3.1 协议 TIME 写

此协议定义于图 36。

TIME 生产者

TIME 消费者



- 时间戳: 6 时间戳对象

图 36: 协议 TIME 写

## 7.2.7 紧急对象 (EMCY)

### 7.2.7.1 紧急对象的使用

紧急对象由 CANopen 设备的内部错误触发并且由 CANopen 设备紧急生产者负责发起。紧急对象适合中断类型的错误报警。一个“错误事件”仅触发一次紧急对象通信。CANopen 设备无新错误不会再产生紧急对象。

零或多个紧急消费者可以接收紧急对象。紧急消费者的反应不在本文范围。

本规范定义了紧急错误码分类 (表 25)、紧急错误代码 (表 26) 和错误寄存器 (见 7.5.2.2)。由具体应用定义的附加错误信息 (低字节错误码) 和紧急条件不在本文范围。附加错误码由其他协议规范定义。

表 25: 紧急错误代码分类

错误码	描述
00 xx <sub>h</sub>	错误复位或没有错误。
10 xx <sub>h</sub>	常规错误
20 xx <sub>h</sub>	当前的
21 xx <sub>h</sub>	当前, CANopen 设备输入侧
22 xx <sub>h</sub>	当前 CANopen 设备内部
23 xx <sub>h</sub>	当前, CANopen 设备输出端
30 xx <sub>h</sub>	电压。
31 xx <sub>h</sub>	主供电电压
32 xx <sub>h</sub>	CANopen 设备内部电压
33 xx <sub>h</sub>	输出电压
40 xx <sub>h</sub>	温度
41 xx <sub>h</sub>	环境温度
42 xx <sub>h</sub>	CANopen 设备温度
50 xx <sub>h</sub>	CANopen 设备硬件
60 xx <sub>h</sub>	CANopen 设备软件



错误代码	描述
61 xx <sub>h</sub>	内部软件
62 xx <sub>h</sub>	用户软件
63 xx <sub>h</sub>	数据集
70 xx <sub>h</sub>	附加模块
80 xx <sub>h</sub>	监控
81 xx <sub>h</sub>	通讯
82 xx <sub>h</sub>	协议错误
90 xx <sub>h</sub>	外部错误
F0xx <sub>h</sub>	附加功能
FFxx <sub>h</sub>	CANopen 设备规范

表 26: 紧急错误代码

错误代码	描述
0000 <sub>h</sub>	错误复位或没有错误
1000 <sub>h</sub>	常规错误。
2000 <sub>h</sub>	当前 – 常规错误。
2100 <sub>h</sub>	当前, CANopen 设备输入端 – 常规
2200 <sub>h</sub>	当前 CANopen 设备内部 – 常规
2300 <sub>h</sub>	当前, CANopen 设备输出端 – 常规
3000 <sub>h</sub>	电压 – 常规错误
3100 <sub>h</sub>	主供电电压 – 常规
3200 <sub>h</sub>	CANopen 设备内部电压 – 常规
3300 <sub>h</sub>	输出电压 – 常规
4000 <sub>h</sub>	温度 – 常规错误。
4100 <sub>h</sub>	环境温度 – 常规
4200 <sub>h</sub>	设备温度 – 常规
5000 <sub>h</sub>	CANopen 设备硬件 – 常规错误
6000 <sub>h</sub>	CANopen 设备软件 – 常规错误
6100 <sub>h</sub>	内部软件 – 常规
6200 <sub>h</sub>	用户软件 – 常规
6300 <sub>h</sub>	数据集 – 常规
7000 <sub>h</sub>	附加模块 – 常规的错误。
8000 <sub>h</sub>	监测 - 常规错误
8100 <sub>h</sub>	通信 – 常规
8110 <sub>h</sub>	CAN 溢出 (对象丢失)
8120 <sub>h</sub>	CAN 被动模式错误
8130 <sub>h</sub>	节点保护错误或者心跳错误
8140 <sub>h</sub>	从总线关闭恢复
8150 <sub>h</sub>	CAN-ID 冲突

错误代码	描述
8200 <sub>h</sub>	协议错误 - 常规
8210 <sub>h</sub>	PDO 由于长度错误不能处理
8220 <sub>h</sub>	PDO 的长度超了
8230 <sub>h</sub>	DAM MPDO 不能被处理, 目标对象不可用
8240 <sub>h</sub>	意外的 SYNC 数据长度。
8250 <sub>h</sub>	RPDO 超时
9000 <sub>h</sub>	外部错误 - 常规错误。
F000 <sub>h</sub>	附加功能 - 常规错误。
FF 00 <sub>h</sub>	特定于设备 - 常规错误。

紧急对象是可选实现的。如 CANopen 设备支持紧急对象, 它应支持至少 2 个错误代码 0000<sub>h</sub> 和 1000<sub>h</sub>。其他所有错误代码都是可选的。

有 CANopen 设备应处于两个紧急状态之一 ( 图 37 )。其转换取决于紧急对象的传输。 错误状态与 NMT 状态机之间的关联由对象 1029<sub>h</sub> 定义( 见子句 7.5. 2.32 )。

1. 初始化后 CANopen 设备如果未检测到错误进入无错误状态。 没有错误消息被发送。
2. 在 CANopen 设备检测到内部错误并表达在紧急报文的前三个字节 ( 错误代码和错误寄存器 )。 CANopen 设备进入错误状态。 带有相应错误代码和错误寄存器的紧急对象被传输。 错误代码被填入本地对象 1003<sub>h</sub>( 预定义错误域)。
3. 一个, 而并非所有的错误原因消失了。 包含错误代码 0000<sub>h</sub>(错误复位)的紧急报文连同剩余的在设备规范错误域和错误寄存器中登记的错误被传输。
4. CANopen 设备产生了新的错误。 CANopen 设备仍然在错误状态, 传输带有对应错误码的紧急对象。 新的错误代码写入错误代码数组顶部 ( 1003<sub>h</sub>)。 保证错误码按时间排序(最早的错误 - 最高子索引, 见对象 1003<sub>h</sub>)。
5. 所有的错误被修复。 在 CANopen 设备进入无错误状态, 传输一个带有错误代码“错误复位 / 无错误”的紧急对象。
6. 重启或关闭电源。

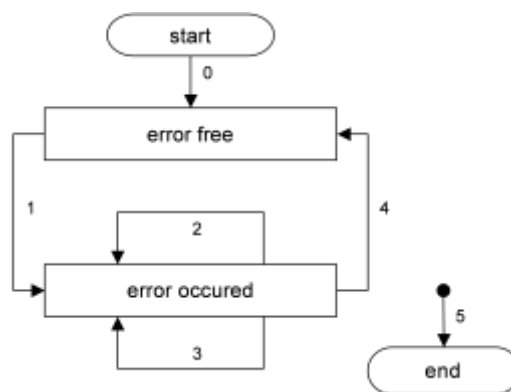


图 37: 紧急状态转换图

## 7.2.7.2 紧急的对象服务

### 7.2.7.2.1 简述

紧急对象传输遵守生产者/消费者模型详见 4.4.4。该服务无应答。

下面是对象属性说明：

- 用户类型：                  之一 { 消费者、生产者 }
- 数据类型：                  结构体
  - UNUNSIGNED16                  紧急错误代码
  - UNUNSIGNED8                  错误寄存器
  - ARRAY (5) of UNUNSIGNED8      制造商指定错误域
- 禁止的时间(inhibit-time):     $N * 100\mu\text{s}$ ,  $N \geq 0$

### 7.2.7.2.2 服务 EMCY 写

EMCY 写服务是推送模型。有零或更多的消费者 EMCY，而 EMCY 生产者只有一个。用于定义此服务的参数见表 27。

通过该服务 EMCY 生产者将当前的紧急数据发送给消费者（们）。

表 27: 服务 EMCY 写

参数	要求 / 说明
要素	强制性
数据	强制

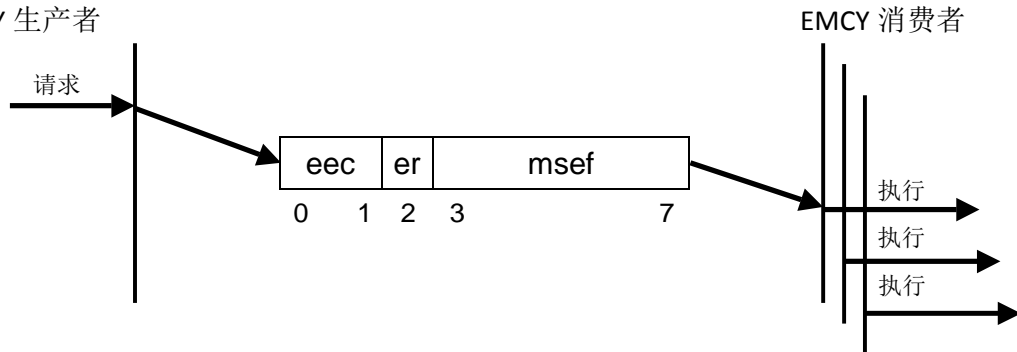
该服务无应答。

## 7.2.7.3 紧急对象协议

### 7.2.7.3.1 协议 EMCY 写

此协议定义于图 38。

EMCY 生产者



- **eec:** 紧急错误码 (见表 26)
- **er:** 错误寄存器(见对象 1001<sub>n</sub>)
- **msef:** 制造商定义错误码

图 38: 协议的 EMCY 写

不允许通过 RTR 触发紧急报文。收到 RTR 将不予应答。

## 7.2.8 网络管理

### 7.2.8.1 简述

CANopen 设备网络管理 ( NMT )为主从结构模型。NMT 对象用于执行 NMT 服务。通过 NMT 服务，CANopen 设备才能够执行初始化、启动、监控、复位或停止等行为。所有的 CANopen 设备被视为 NMT 从站。NMT 从站使用 node-ID 标识其身份，取值范围[1..127 ]。

网络中要有一个 CANopen 设备满足 NMT 主站功能需求。

### 7.2.8.2 NMT 服务

#### 7.2.8.2.1 节点控制服务

##### 7.2.8.2.1.1 简述

通过节点控制服务，NMT 主站控制 NMT 从站状态。NMT 状态值 {停止、预操作、操作、初始化}。节点控制服务可以操作单个 CANopen 设备或同时针对所有 CANopen 设备。NMT 主站状态由本地服务控制。节点控制服务可由本地应用程序初始化。

#### 7.2.8.2.1.2 启动远程节点服务

NMT 主站通过 NMT 服务启动远程节点，从而更改所选 NMT 从站状态。新状态应为操作态。用于此服务的参数被定义于表 28 。

表 28: 启动远程节点服务

参数	说明 / 要求
<b>要素</b>	<b>强制性</b>
Node-ID	选择
所有	选择

该服务无应答, 而且是强制支持的。

#### 7.2.8.2.1.3 停止远程节点服务

NMT 主站通过 NMT 服务停止远程节点，从而更改所选 NMT 从站状态。新状态应为停止态。用于此服务的参数被定义于表 29 。

表 29:停止远程节点服务

参数	说明 / 要求
<b>要素</b>	<b>强制性</b>
Node-ID	选择
所有	选择

该服务无应答, 而且是强制支持的。

### 7.2.8.2.1.4 进入预操作服务

NMT 主站通过 NMT 服务使远程节点进入预操作态，从而更改所选 NMT 从站状态。新状态应为预操作态。用于此服务的参数被定义于表 30。

表 30:进入预操作服务

参数	说明 / 要求
<b>要素</b>	<b>强制性</b>
Node-ID	选择
所有	选择

该服务无应答, 而且是强制支持的。

### 7.2.8.2.1.5 复位节点服务

NMT 主站通过 NMT 服务复位远程节点，从而更改所选 NMT 从站状态。新状态应为 NMT 子状态-复位应用。用于此服务的参数被定义于表 31。

表 31:复位远程节点服务

参数	说明 / 要求
<b>要素</b>	<b>强制性</b>
Node-ID	选择
所有	选择

该服务无应答, 而且是强制支持的。

### 7.2.8.2.1.6 复位通信服务

NMT 主站通过 NMT 服务复位远程节点通信部分，从而更改所选 NMT 从站状态。新状态应为 NMT 子状态-复位通信。用于此服务的参数被定义于表 32。

表 32:复位通信服务

参数	说明 / 要求
<b>要素</b>	<b>强制性</b>
Node-ID	选择
所有	选择

该服务无应答, 而且是强制支持的。

## 7.2.8.2.2 错误控制 服务

错误控制服务是用于检测 CAN 网络内出现的故障。

CANopen 设备的本地错误可能导致复位或状态变化。具体本地错误的定义不属于本规格的范围。

错误控制服务主要是 CANopen 设备通过定期发送消息实现的。有两种错误控制方法可用。

NMT 主站发送监护请求（节点监护协议：**node guarding protocol**），如果 NMT 从站没有在规定的时间内跨度（节点生存周期：**node life time**）回应或 NMT 从站的通讯状态发生变化，NMT 主站会向应用程序通报相关事件。

如果支持生存监护（NMT 从站保护主站），NMT 从站由数据字典中的监护周期（**guard time**）和生存周期因子（**life time factor**）决定自身节点生存周期（**lifetime**）。如果在生存周期内没收到监护请求，NMT 从站将此事件报告本地应用程序。如果监护周期和生存周期因子都是 0（默认值），NMT 从站将不会监护主站。

接收到第一帧带有监护 CAN-ID 的 RTR 意味着 NMT 从站的监护开始。此时可能是出在启动阶段或稍后。

心跳机制主要内容是 CANopen 设备的心跳（heartbeat）生产者周期性的发送心跳消息，一或多个 CANopen 网络中的设备知悉这一心跳消息。如果心跳生产者出现停跳，其消费者会上报该事件。

监护或心跳都是强制性的。

备注：虽然心跳和保护默认都是禁用的，但仍然推荐使用错误控制机制。

### 7.2.8.2.2.1 节点监护事件服务

通过该服务，NMT 能够知悉 node-ID 标识的远程设备产生或正在解决错误。用于此服务的参数被定义于表 33。

事件得到解决会执行的操作

- 由于状态切换导致事件发生将收到期望的状态，或（译注：不确定）
- 由于超时引起的事件发生其后续远程执行将得到确认。（译注：不确定）

表 33: 服务节点监护事件

参数	执行
<b>要素</b>	<b>强制性</b>
Node-ID	强制
状态	强制
发生	选择
解决	选择
原因	可选的
超时	选择
状态更改	选择

该服务在初始化和操作态提供。

### 7.2.8.2.2.2 节点监护事件服务

通过该服务，NMT 从站能够获知远程错误的发生和解决。用于此服务的参数被定义于表 34。

事件的解决由事件发生后续的远程请求来表达。

表 34: 节点监护事件服务

参数	执行
<b>要素</b>	<b>强制性</b>
状态	强制
发生	选择
解决	选择

该服务由提供者发起并且是可选的功能。

### 7.2.8.2.2.3 心跳事件服务

通过该服务，心跳消费者能够获悉 node-ID 标识的 CANopen 设备发生错误、得到解决或 NMT 状态切换。用于此服务的参数被定义于表 35。

事件发生后续的的远程消息表明事件的解决。

表 35:心跳事件服务

参数	执行
<b>要素</b>	<b>强制性</b>
Node-ID	强制
状态	强制
发生	选择
解决	选择
原因	可选
超时	选择
状态切换	选择

该服务由消费者发起并且是可选的功能。

### 7.2.8.2.3 Boot-up 服务

#### 7.2.8.2.3.1 Boot-up 事件服务

通过该服务，NMT 从站上报本地状态从初始化切换到预操作态。用于定义此服务的参数在表 36 中。

表 36:Boot-up 事件服务

参数	处理
要素 Node-ID	强制性 强制

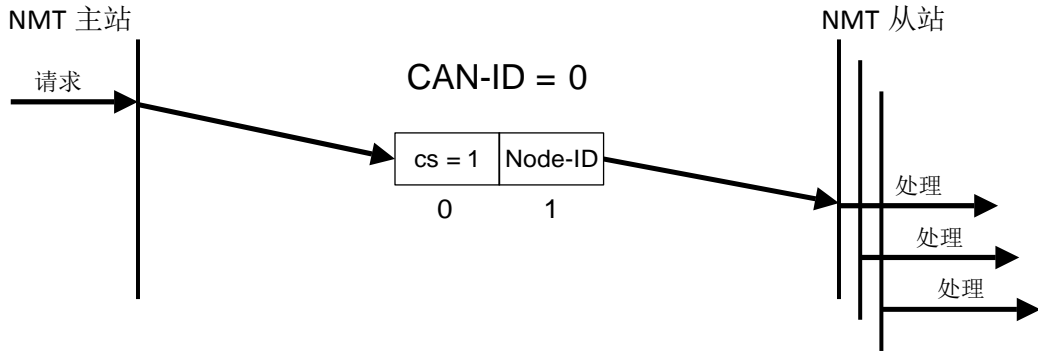
该服务由提供者发起并且是强制功能。

### 7.2.8.3 NMT 协议

#### 7.2.8.3.1 节点控制协议

##### 7.2.8.3.1.1 启动远程节点协议

该协议定义于图 39。



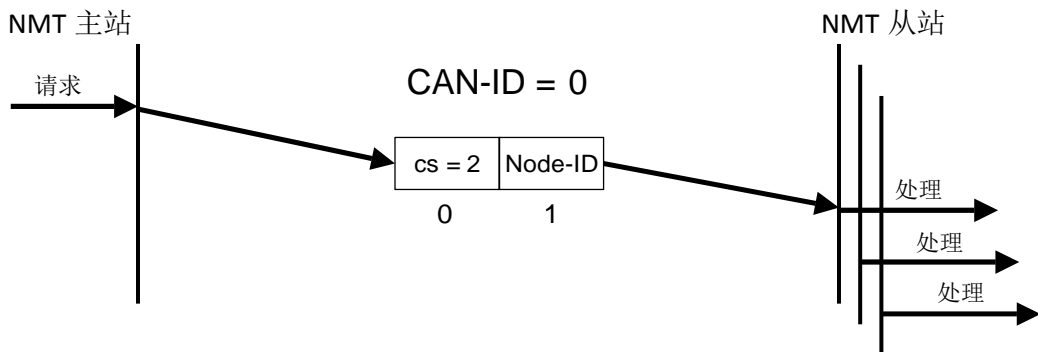
cs: NMT 命令说明

- 1: 启动

图 39: 启动远程节点协议

##### 7.2.8.3.1.2 停止远程节点协议

该协议定义于图 40。



cs: NMT 命令说明

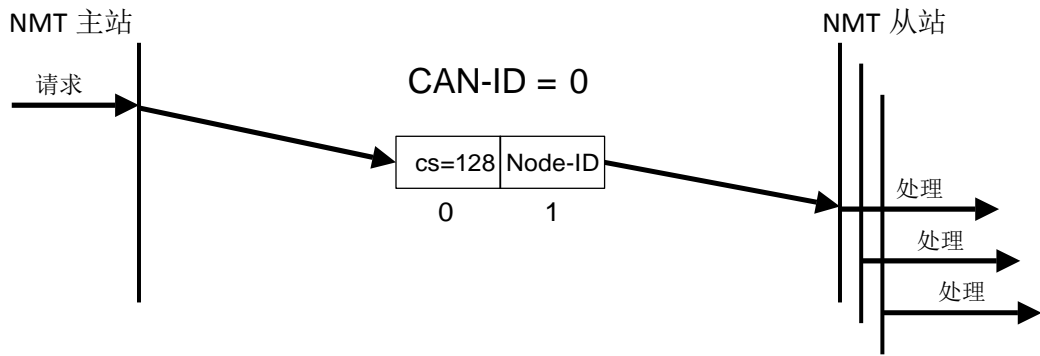
- 2: 停止

图 40: 停止远程节点协议



## 7.2.8.3.1.3 进入预操作态协议

该协议定义于图 41。



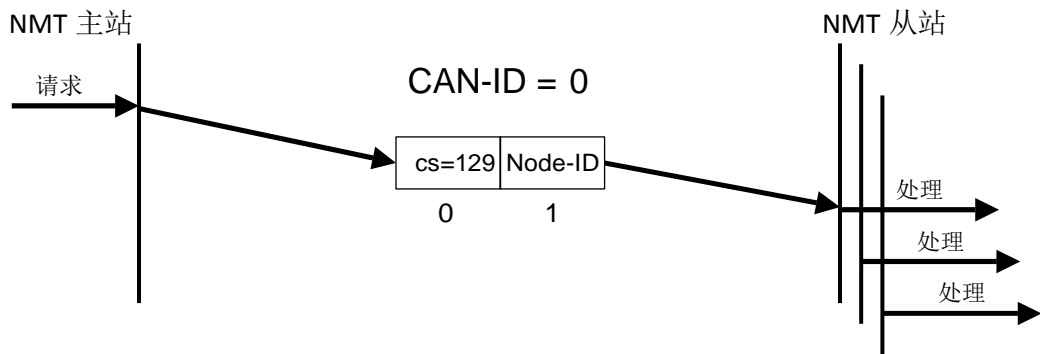
**cs:** NMT 命令说明

128: 进入预操作态

图 41: 进入预操作态协议

## 7.2.8.3.1.4 复位节点协议

该协议定义于图 42。



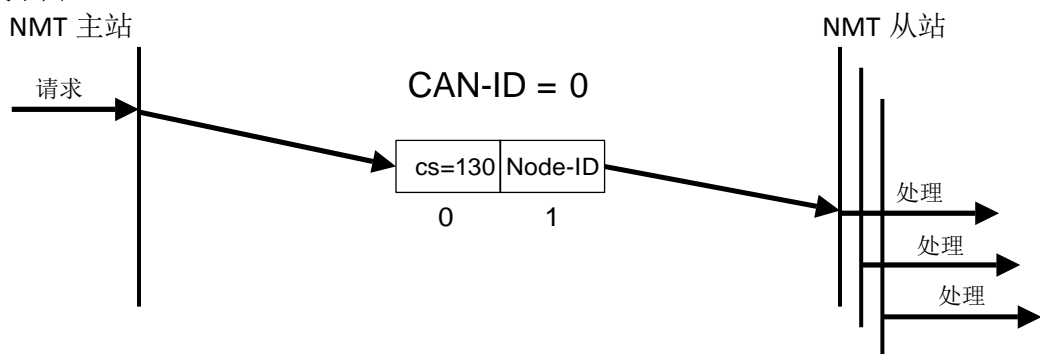
**cs:** NMT 命令说明

129: 复位节点

图 42: 复位节点协议

## 7.2.8.3.1.5 复位通信协议

该协议定义于图 43。



**cs:** NMT 命令说明

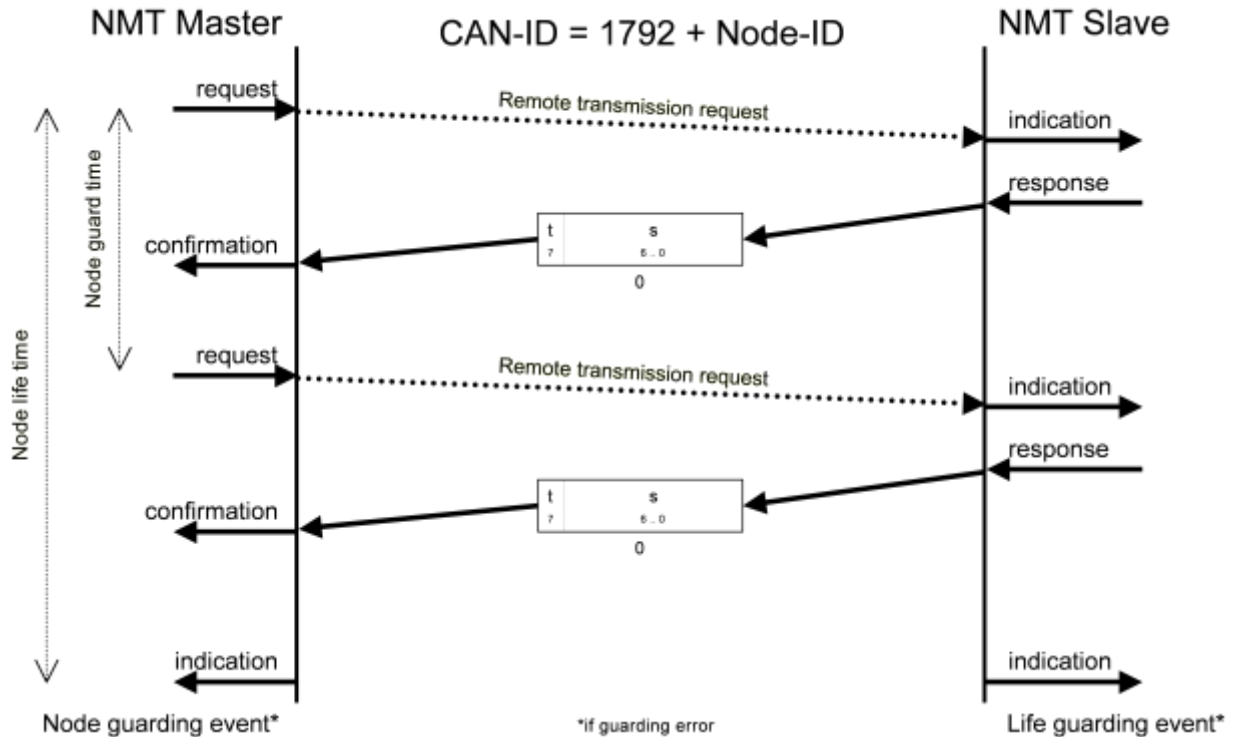
129: 复位通信

图 43: 复位通信协议

## 7.2.8.3.2 错误控制协议

## 7.2.8.3.2.1 节点监护协议

该协议定义于图 44。每个 NMT 服务一个节点监护协议的 RTR。该协议实现提供者启动的错误控制服务。



**s** : NMT 从站状态

- 4 : 停止态
- 5 : 操作态
- 127 : 预操作态

**t** : 翻转位。NMT 从站每次应答翻转一次。首次应答为 0。NMT 复位通信后置 0(无其他 NMT 状态会复位翻转位)。如果收到连续两个翻转位相同的响应，后一次将被忽略。

图 44:节点监护协议

NMT 主站常规时段内轮询各从站。该时段叫做监护周期，且每个从站可能不同。从站响应带有自身 NMT 状态。节点生存周期为其监护周期乘以生存周期因子。每个节点的生存周期可以不同。如果 NMT 从站在生存周期内没有被查询，远程节点错误事件将通过 NMT 生存监护事件服务上报。

远程节点的错误上报条件

- 节点生存周期内内没收到 RTR
- 报告的 NMT 从站状态与预期不匹配

远程节点报告过错误并且错误监护协议报文不再产生,说明错误通过 NMT 节点监护服务和生存监护服务得到解决。

默认的监护周期和生存周期因子值由对应的数据字典对象说明。

## 7.2.8.3.2.2 心跳协议

心跳协议定义于图 45，为不需要 RTRs 支持的节点控制服务。心跳生产者周期的发送心跳。一个或多个消费者接收之。生产者和消费者的关系由数据字典配置。心跳消费者在消费期内监控关联心跳。如果消费期内没收到心跳，则一个心跳事件将产生。

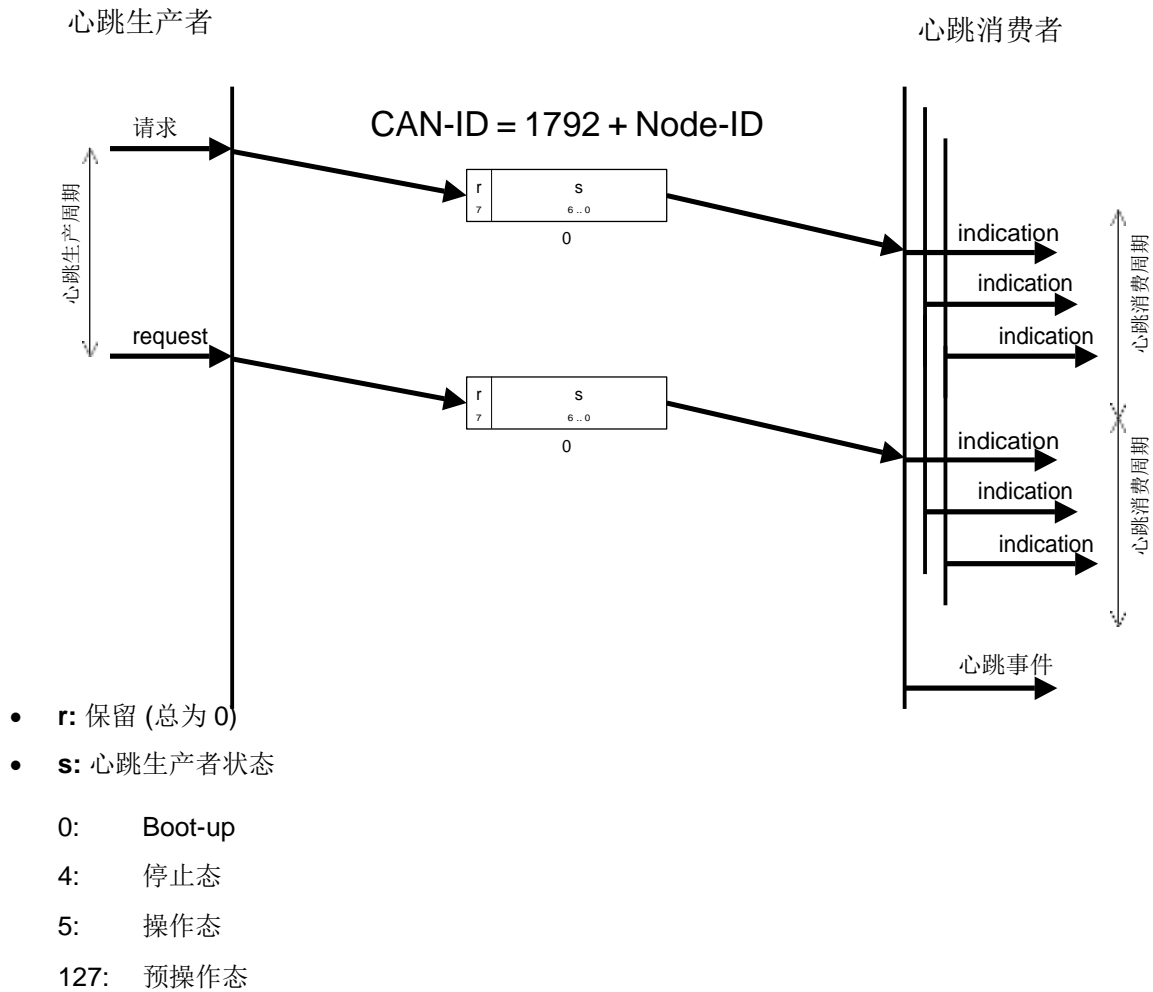


图 45:心跳协议

如果 CANopen 配置了心跳生产周期其心跳协议功能立即启动。如果心跳生产周期非 0，CANopen 设备心跳协议在 NMT 初始化和预操作态就开始运行。boot-up 消息被视为第一次心跳。不允许在 NMT 从站中同时使用保护协议和心跳协议这两种错误控制机制。如果心跳生产周期不等于 0 心跳协议生效。

### 7.2.8.3.3 boot-up 协议

该协议定义于图 46，用于 NMT 从站，通报节点从初始化态进入预操作态。该协议使用与错误控制协议相同的 CAN-ID。

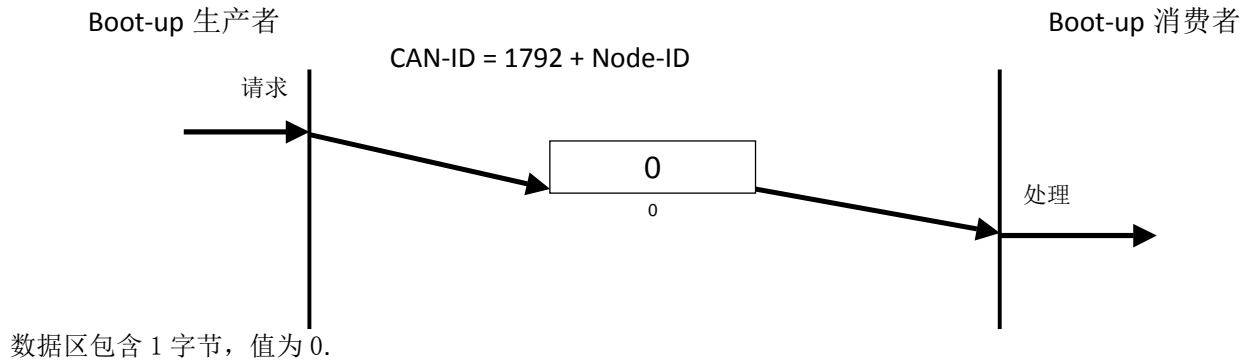


图 46: boot-up 协议

## 7.3 网络初始化和系统 boot-up

### 7.3.1 简化的 NMT 启动

例如，简化的 NMT 启动显示于图 47。该流程的定义不属于本规范的范围。

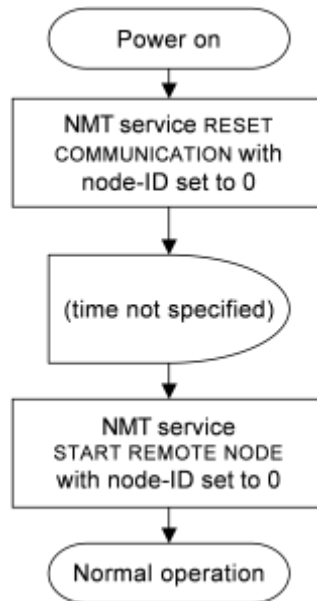


图 47: 简单 NMT 启动

## 7.3.2 NMT 状态机

### 7.3.2.1 概述

图 48 描述了 CANopen 设备的 NMT 状态图。CANopen 设备在初始化后直接进入与操作态。在这一状态可以对 CANopen 设备配置参数以及通过 SDO 服务进行 CAN-ID-allocation (例如使用配置工具)。然后 CANopen 设备就可以直接进入操作态运行。

NMT 状态机决定了通信功能单元的行为 (见 4.3)。CANopen 设备依赖的应用状态机和 NMT 状态机是设备协议和应用协议的范畴。

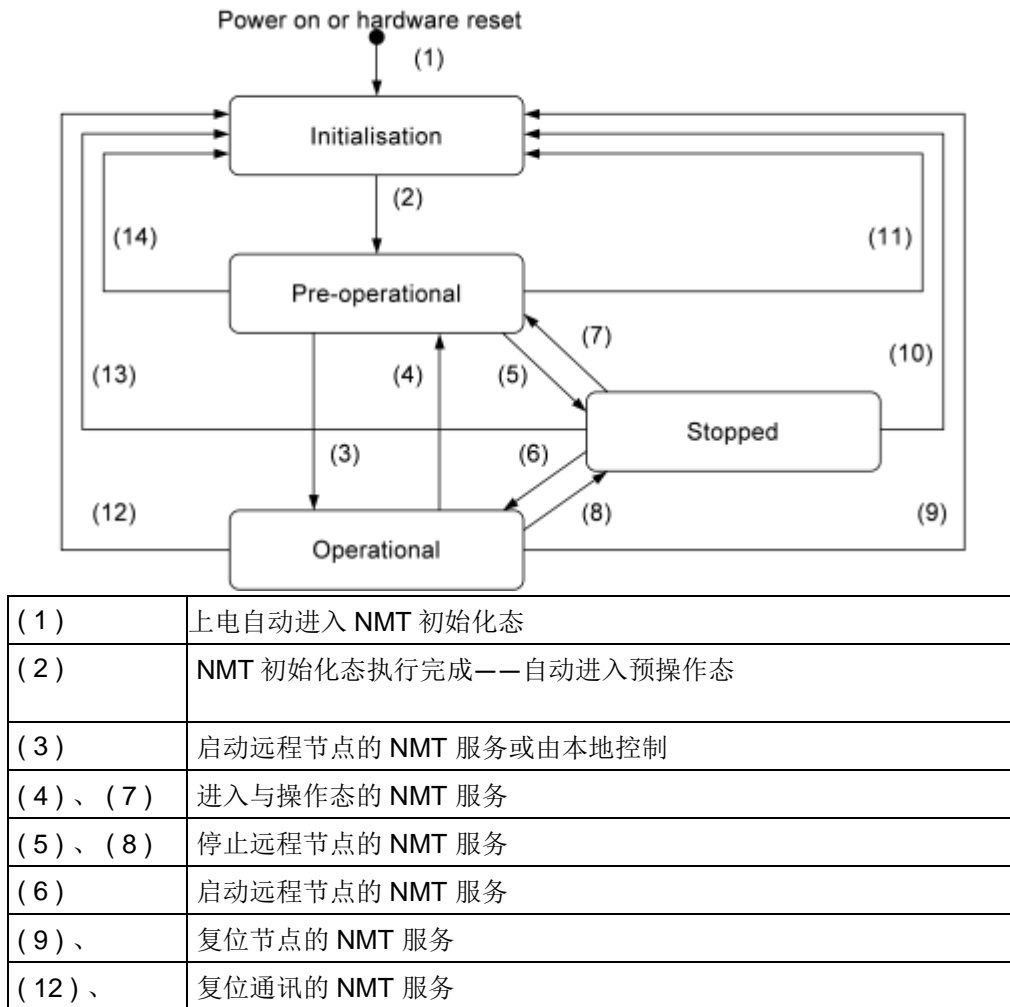


图 48: CANopen 设备 NMT 状态图

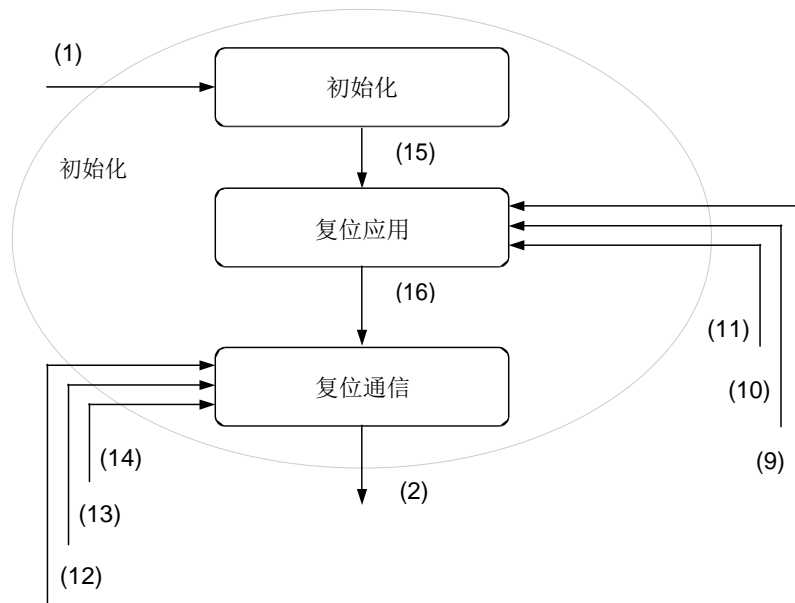
## 7.3.2.2 NMT 状态

### 7.3.2.2.1 NMT 初始化态

初始化态分为三个子状态(如图 49),以便能够完全或部分复位 CANopen 设备。

1. **初始化:** CANopen 上电或硬复位后的第一个 NMT 子状态。执行基本的 CANopen 设备初始化后自动进入复位应用子状态。
2. **复位应用:** 该子状态下, 制造商协议区和标准设备协议区参数被赋上电值。之后自动进入复位通信子状态。
3. **复位通信:** 该子状态下, 通信协议区参数被赋上电值。然后 CANopen 设备执行 boot-up 写服务并进入预操作态。

上电值为最近一次存储的参数。如果不支持保存、保存操作未执行、复位前执行了恢复默认指令(见 7.5.2.14), 上电值即为根据通信和设备协议定义的默认值。



(1)	上电自动进入 NMT 初始化
(2)	NMT 初始化完成——进预操作态
(12)、	NMT 复位通讯服务
(9)、	NMT 复位节点服务
(15)	NMT 初始化子状态完成 – 自动进入 NMT 复位应用子状态
(16)	NMT 复位应用子状态完成 – 自动进入 NMT 复位通信子状态

图 49: NMT 初始化状态结构

### 7.3.2.2.2 NMT 预操作态

在预操作态, 允许 SDO 通信, 不允许 PDO 通信, 此状态通常用于配置 PDO 的参数和映射对象 (PDO 映射) 等。

CANopen 设备可以由 NMT 启动远程节点服务或通过本地控制切换至操作态。

### 7.3.2.2.3 NMT 操作态

此状态允许所有通信服务。传输 PDOs，通过 SDO 访问数据字典，然而由于执行方面的问题或者是应用状态机可能要求限制对相关对象字典的访问，例如某对象可能在应用程序执行过程中不允许修改。

### 7.3.2.2.4 NMT 停止态

切换 CANopen 设备进入 NMT 停止态来停止所有通信服务(除节点保护和心跳, 如果被激活的话)。此外, 这种状态可用于实现特定的应用行为。这种行为属于设备协议和应用协议的范畴。

此状态下如果触发的 EMCY 消息将被挂起。在切换到其它状态后, 最近的 EMCY 将被激活。

注：错误的历史可通过访问只读对象 1003<sub>h</sub> 获取。

### 7.3.2.2.5 NMT 状态和通信对象关系

表 37 的规定之间的关系。表中的服务只在 CANopen 设备处在适当的状态下才能被执行。

表 37 : NMT 状态和通信对象关系

	预操作	操作	停止
PDO		X	
SDO	X	X	
SYNC	X	X	
TIME	X	X	
EMCY	X	X	
节点控制和错误控制	X	X	X

### 7.3.2.3 NMT 状态转换

NMT 状态转换条件

- 接到 NMT 节点控制服务
- 硬件复位, 或
- 节点控制本地服务启动的由设备和应用协议定义的应用事件。

### 7.3.3 通用预定义连接集

为了减少配置工作量而定义的简单的网络 CAN-ID 分配方案。这些 CAN-IDs 将在 NMT 初始化态完成进入预操作态后生效(如果未经修改)。对象 SYNC, TIME, EMCY 写和 PDO 在动态分配新的 CAN-IDs 后将被重新配置。CANopen 设备应只为受支持的通信对象提供相应的 CAN-IDs。

CAN-ID 分配方案 (定义中表 38 和表 39) 包括功能部分, 它决定了对象的优先级和 node-ID 部分, 它用于区分 CANopen 设备。这将允许在单一主站和多至 127 个从站间进行点对点通信。同时还支持无应答的 NMT, SYNC 和 TIME 广播消息。广播的 node-ID 为零。



预定义连接集支持一个紧急对象，一个 SDO,至多 4 RPDOs 4 TPDOs 和 NMT 对象。

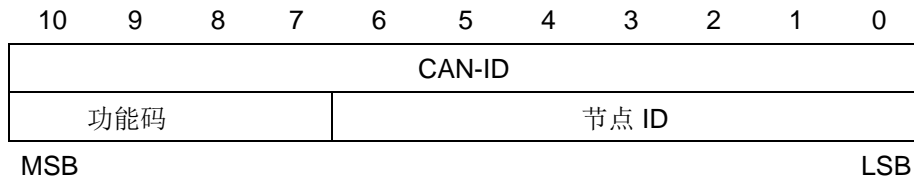


图 50: 通用预定义连接集 CAN-ID 分配方案设置

表 38 和表 39 列出了支持的对象和访 CAN-IDs。

表 38: 通用预定义连接集广播对象

COB	功能码	CAN-ID
NMT	0000 <sub>b</sub>	0 ( 000 <sub>H</sub> )
SYNC	0001 <sub>b</sub>	128 ( 080 <sub>H</sub> )
TIME	0010 <sub>b</sub>	256 ( 100 <sub>H</sub> )

表 39: 点对点对象的通用预定义的连接设置

COB	功能码	CAN-ID
EMCY	0001 <sub>b</sub>	129 (081 <sub>H</sub> ) – 255 (0FF <sub>H</sub> )
PDO 1 ( tx )	0011 <sub>b</sub>	385 (181 <sub>H</sub> ) – 511 (1FF <sub>H</sub> )
PDO 1 ( rx )	0100 <sub>b</sub>	513 (201 <sub>H</sub> ) – 639 (27F <sub>H</sub> )
PDO 2 ( tx )	0101 <sub>b</sub>	641 (281 <sub>H</sub> ) – 767 (2FF <sub>H</sub> )
PDO 2 ( rx )	0110 <sub>b</sub>	769 (301 <sub>H</sub> ) – 895 (37F <sub>H</sub> )
PDO 3 ( tx )	0111 <sub>b</sub>	897 (381 <sub>H</sub> ) – 1023 (3FF <sub>H</sub> )
PDO 3 ( rx )	1000 <sub>b</sub>	1025 (401 <sub>H</sub> ) – 1151 (47F <sub>H</sub> )
PDO 4 ( tx )	1001 <sub>b</sub>	1153 (481 <sub>H</sub> ) – 1279 (4FF <sub>H</sub> )
PDO 4 ( rx )	1010 <sub>b</sub>	1281 (501 <sub>H</sub> ) – 1407 (57F <sub>H</sub> )
SDO ( tx )	1011 <sub>b</sub>	1409 (581 <sub>H</sub> ) – 1535 (5FF <sub>H</sub> )
SDO ( rx )	1100 <sub>b</sub>	1537 (601 <sub>H</sub> ) – 1663 (67F <sub>H</sub> )
NMT 错误控制	1110 <sub>b</sub>	1793 (701 <sub>H</sub> ) – 1919 (77F <sub>H</sub> )

表 39 是 CANopen 设备视角。

通用预定义连接集仅使用 11 位可 CAN-ID 标准帧,即使网络支持扩展帧。

通用预定义连接集适用于所有的 CANopen 设备,并且都遵循特定的设备协议,不遵循任何应用协议。

### 7.3.4 特定预定义连接集

特定预定义连接集用于替换通用预定义连接集,其遵守某应用协议。特定预定义连接集不在本规范定义范围;而是由相应的应用协议定义。

### 7.3.5 受限 CAN-IDs

任何列在表 40 的 CAN-IDs 使用受限。受限 CAN-ID 不能用于任何可配置通信对象，SYNC, TIME, EMCY, PDO, 和 SDO 都不行。

表 40: 受限 CAN-ID

CAN-ID	COB 使用
0 (000 <sub>h</sub> )	NMT
1 (001 <sub>h</sub> ) – 127 (07F <sub>h</sub> )	保留
257 (101 <sub>h</sub> ) – 384 (180 <sub>h</sub> )	保留
1409 (581 <sub>h</sub> ) – 1535 (5FF <sub>h</sub> )	默认 SDO (tx)
1537 (601 <sub>h</sub> ) – 1663 (67F <sub>h</sub> )	默认 SDO (rx)
1760 (6E0 <sub>h</sub> ) – 1791 (6FF <sub>h</sub> )	保留
1793 (701 <sub>h</sub> ) – 1919 (77F <sub>h</sub> )	NMT 错误控制
2020 (780 <sub>h</sub> ) – 2047 (7FF <sub>h</sub> )	保留

## 7.4 对象字典

### 7.4.1 常规结构

标准对象字典的整体布局定义与表表 41

表 41: 对象字典结构。

索引	对象。
0000 <sub>h</sub>	未使用
0001 <sub>h</sub> – 001F <sub>h</sub>	静态数据类型
0020 <sub>h</sub> – 003F <sub>h</sub>	复合数据类型。
0040 <sub>h</sub> – 005F <sub>h</sub>	制造商指定复合数据类型。
0060 <sub>h</sub> – 025F <sub>h</sub>	设备协议指定数据类型。
0260 <sub>h</sub> – 03FF <sub>h</sub>	保留的
0400 <sub>h</sub> – 0FFF <sub>h</sub>	保留的
1000 <sub>h</sub> – 1FFF <sub>h</sub>	通信协议区
2000 <sub>h</sub> – 5FFF <sub>h</sub>	制造商指定协议区
6000 <sub>h</sub> – 67FF <sub>h</sub>	标准化协议区 1 <sup>st</sup> 逻辑设备
6800 <sub>h</sub> – 6FFF <sub>h</sub>	标准化协议区 2 <sup>nd</sup> 逻辑设备
7000 <sub>h</sub> – 77FF <sub>h</sub>	标准化协议区 3 <sup>rd</sup> 逻辑设备
7800 <sub>h</sub> – 7FFF <sub>h</sub>	标准化协议区 4 <sup>th</sup> 逻辑设备
8000 <sub>h</sub> – 87FF <sub>h</sub>	标准化协议区 5 <sup>th</sup> 逻辑设备
8800 <sub>h</sub> – 8FFF <sub>h</sub>	标准化协议区 6 <sup>th</sup> 逻辑设备
9000 <sub>h</sub> – 97FF <sub>h</sub>	标准化协议区 7 <sup>th</sup> 逻辑设备
9800 <sub>h</sub> – 9FFF <sub>h</sub>	标准化协议区 8 <sup>th</sup> 逻辑设备
A000 <sub>h</sub> – AFFF <sub>h</sub>	标准网络变量区
B000 <sub>h</sub> – BFFF <sub>h</sub>	标准系统变量区
C000 <sub>h</sub> – FFFF <sub>h</sub>	保留

对象字典容量 65536,通过 16 位索引寻址,每个对象可包含多至 256 个子项,通过 8 位子索引号寻址。

0001<sub>h</sub> 到 001F<sub>h</sub> 的静态数据类型包含标准的类型定义如 BOOLEAN、INTEGE、UNSIGNED、浮点、字符串等。

0020<sub>h</sub> 到 003F<sub>h</sub> 的复合数据类型是相对于标准数据类型而言的普适所有 CANopen 设备的预定义结构。

0040<sub>h</sub> 到 005F<sub>h</sub> 的制造商指定复合数据类型是相对于标准数据类型而言的特定 CANopen 设备所定义的结构。

设备协议可以定义附加数据类型作为其设备的类型。设备协议定义的静态数据类型和复合数据类型位列 0060<sub>h</sub> 到 025F<sub>h</sub>。

CANopen 设备可选择支持只读索引区的复合数据类型 ( 从 0020<sub>h</sub> 到 005F<sub>h</sub> 和 0060<sub>h</sub> 到 025F<sub>h</sub>) 结构。子索引 0 保存子索引数,后续子索引数据类型为表 44 规定的 UNSIGNED16 。

从 1000<sub>h</sub> 到 1FFF<sub>h</sub> 的通信协议区包含了通信相关参数。这些对象适用于所有的 CANopen 设备。

从 6000<sub>h</sub> 到 9FFF<sub>h</sub> 的标准协议区包含可由网络读写的某类 CANopen 设备的所有数据对象。设备协议使用从 6000<sub>h</sub> 到 9FFF<sub>h</sub> 区域来描述功能和参数。

对象字典在概念上迎合了可选功能,这意味着生产商在其 CANopen 设备提供某些扩展功能,可以使用预定义方式。从 2000<sub>h</sub> 到 5FFF<sub>h</sub> 的空间正是为 制造商的定制功能所留。

从 A0000<sub>h</sub> 到 AFFF<sub>h</sub> 的网络变量区应包含输入变量和输出 变量,这是可编程 CANopen 设备的一部分。这些网络变量的定义不在本规范的范围,而是将来的协议和框架的一部分。

B000<sub>h</sub> 到 BFFF<sub>h</sub> 的系统变量应包含输入变量和输出变量,其为有层次感的基础 CANopen 网络的一部分。这些系统变量的定义不在本规范的范围,而是将来的协议和框架的一部分。

## 7.4.2 索引和子索引 的使用

16 位的索引引用可寻址对象字典的所有对象。简单的变量可由索引直接引用。记录和数组可由索引寻址其数据整体机构。

定义子索引来访问数据结构的具体元素。对于单一的对象字典对象如 UNSIGNED8, BOOLEAN, INTEGER32 等值,其子索引始终是 00<sub>h</sub>。对于复合的对象字典对象如带有多个数据字段的数组或记录,子索引的引用域为主索引所指向的数据结构。子索引访问的字段可以是不同的数据类型。

### 7.4.3 对象类型码的使用

对象编码表达了对象字典特定索引处的对象类型。使用到以下定义：

表 42: 对象字典对象定义

对象名称	说明	对象类型码
NULL	无数据字段对象	00 <sub>h</sub>
DOMAIN	大体量数据，如可执行的程序代码	02 <sub>h</sub>
DEFTYPE	表示类型定义如 BOOLEAN, UNSIGNED16, FLOAT 等等	05 <sub>h</sub>
DEFSTRUCT	定义了一种新的记录类型如在 21 <sub>h</sub> 的 PDO 映射结构	06 <sub>h</sub>
VAR	单值如 UNSIGNED8, BOOLEAN, FLOAT, INTEGER16, VISIBLE STRING 等	07 <sub>h</sub>
ARRAY	多数据字段对象，每个数据字段都是一种简单相同的基本数据类型变量如 UNSIGNED16 的数组。子索引 0 类型为 UNSIGNED8，并非数组的一部分。	08 <sub>h</sub>
RECORD	多数据字段对象，数据字段可以是任意简单变量的组合。子索引 0 是 UNSIGNED8，子索引 255 是 UNSIGNED32,并非记录的一部分。	09 <sub>h</sub>

### 7.4.4 数据类型的使用

对象的数据类型信息包括下列预定义类型：BOOLEAN, FLOAT, UNSIGNED, INTEGER, VISIBLE/OCTET STRING, TIME\_OF\_DAY, TIME\_DIFFERENCE 和 DOMAIN (见 7.1)。它还包含预定义的复合数据类型 PDO 映射和制造商、协议规范或应用协议规范所定义的其他类型。禁止定义记录的记录、记录数组或包含数组的记录。数组或记录对象内的一个子索引表示一个数据字段。

### 7.4.5 访问权限的使用

该属性定义了对象的访问权限。其视点是对 CANopen 网络设备来说的。

它应是以下项之一：

表 43: 数据对象访问权限属性

属性	描述
rw	读写权限
wo	只写权限
ro	只读权限
const	只读并且值为常量 该值可以在 NMT 初始化态更改。其余状态不可变更。

## 7.4.6 类别和条目类的使用

类别和条目类别定义了对象是强制性的、可选的还是条件的。CANopen 设备必须支持强制对象。CANopen 设备可以支持可选对象。如果设备执行特定的功能才会支持对应的对象。在这种情况下，详细的对象规范会描述其中关系，并且该对象被定义为条件的。

## 7.4.7 数据类型入口的使用

### 7.4.7.1 简述

静态数据类型放在对象字典里只是出于定义的目的。范围从 0001<sub>h</sub> 到 0007<sub>h</sub>、0010<sub>h</sub>、从 0012<sub>h</sub> 到 0016<sub>h</sub>，并从 0018<sub>h</sub> 到 001B<sub>h</sub> 可以被映射，以便确定该 CANopen 设备(不关系)RPDO 中适当的未使用空间。DEFTYPE 和 DEFSTRUCT 不允许映射到 RPDOs 中。

数据类型如下所示：

表 44: 对象字典的数据类型

索引	对象	名称
0001 <sub>h</sub>	DEFTYPE	BOOLEAN
0002 <sub>h</sub>	DEFTYPE	INTEGER8
0003 <sub>h</sub>	DEFTYPE	INTEGER16
0004 <sub>h</sub>	DEFTYPE	INTEGER32
0005 <sub>h</sub>	DEFTYPE	UNSIGNED8
0006 <sub>h</sub>	DEFTYPE	UNSIGNED16
0007 <sub>h</sub>	DEFTYPE	UNSIGNED32
0008 <sub>h</sub>	DEFTYPE	REAL32
0009 <sub>h</sub>	DEFTYPE	VISIBLE_STRING
000A <sub>h</sub>	DEFTYPE	OCTET_STRING
000B <sub>h</sub>	DEFTYPE	UNICODE_STRING
000C <sub>h</sub>	DEFTYPE	TIME_OF_DAY
000D <sub>h</sub>	DEFTYPE	TIME_DIFFERENCE
000E <sub>h</sub>	保留	
000F <sub>h</sub>	DEFTYPE	DOMAIN
0010 <sub>h</sub>	DEFTYPE	INTEGER24
0011 <sub>h</sub>	DEFTYPE	REAL64

索引	对象	名称
0012 <sub>h</sub>	DEFTYPE	INTEGER40
0013 <sub>h</sub>	DEFTYPE	INTEGER48
0014 <sub>h</sub>	DEFTYPE	INTEGER56
0015 <sub>h</sub>	DEFTYPE	INTEGER64
0016 <sub>h</sub>	DEFTYPE	UNSIGNED24
0017 <sub>h</sub>	保留	
0018 <sub>h</sub>	DEFTYPE	UNSIGNED40
0019 <sub>h</sub>	DEFTYPE	UNSIGNED48
001A <sub>h</sub>	DEFTYPE	UNSIGNED56
001B <sub>h</sub>	DEFTYPE	UNSIGNED64
001C <sub>h</sub> – 001F <sub>h</sub>	保留	
0020 <sub>h</sub>	DEFSTRUCT	PDO_COMMUNICATION_PARAMETER
0021 <sub>h</sub>	DEFSTRUCT	PDO_MAPPING
0022 <sub>h</sub>	DEFSTRUCT	SDO_PARAMETER
0023 <sub>h</sub>	DEFSTRUCT	IDENTITY
0024 <sub>h</sub> – 003F <sub>h</sub>	保留	
0040 <sub>h</sub> – 005F <sub>h</sub>	DEFSTRUCT	制造商的复合数据类型
0060 <sub>h</sub> – 007F <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 1 <sup>st</sup> 的逻辑设备
0080 <sub>h</sub> – 009F <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 1 <sup>st</sup> 的逻辑设备
00A0 <sub>h</sub> – 00BF <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 2 <sup>nd</sup> 的逻辑设备
00C0 <sub>h</sub> – 00DF <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 2 <sup>nd</sup> 的逻辑设备
00E0 <sub>h</sub> – 00FF <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 3 <sup>rd</sup> 的逻辑设备
0100 <sub>h</sub> – 011F <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 3 <sup>rd</sup> 的逻辑设备
0120 <sub>h</sub> – 013F <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 4 <sup>th</sup> 的逻辑设备
0140 <sub>h</sub> – 015F <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 4 <sup>th</sup> 的逻辑设备
0160 <sub>h</sub> – 017F <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 5 <sup>th</sup> 的逻辑设备
0180 <sub>h</sub> – 019F <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 5 <sup>th</sup> 的逻辑设备
01A0 <sub>h</sub> – 01BF <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 6 <sup>th</sup> 的逻辑设备
01C0 <sub>h</sub> – 01DF <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 6 <sup>th</sup> 的逻辑设备
01E0 <sub>h</sub> – 01FF <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 7 <sup>th</sup> 的逻辑设备
0200 <sub>h</sub> – 021F <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 7 <sup>th</sup> 的逻辑设备
0220 <sub>h</sub> – 023F <sub>h</sub>	DEFTYPE	设备协议规范的标准数据类型 8 <sup>th</sup> 的逻辑设备
0240 <sub>h</sub> – 025F <sub>h</sub>	DEFSTRUCT	设备协议规范的标准数据类型 8 <sup>th</sup> 的逻辑设备

数据类型使用详见 7.1。每个 CANopen 设备不需要支持所有已定义的数据类型。CANopen 设备仅需支持它在 1000<sub>h</sub> 到 AFFF 用到的数据类型。

预定义的复合数据类型放在标准的数据类型之后。详见 7.4. 8。

CANopen 设备可选支持标准数据类型编码的数据长度 (UNSIGNED32)。例如索引 000C<sub>h</sub>

(TIME\_OF\_DAY) 包含值 0000 0030<sub>h</sub> = 48<sub>d</sub> 使用 48 位序列作为 TIME\_OF\_DAY 的数据类型编码。如果长度是可变的 (例如 000F<sub>h</sub> = 域), 对象的入口包含 0000 0000<sub>h</sub>。

为支持复合数据类型, CANopen 设备可选择提供该种数据类型结构。子索引 00<sub>h</sub> 为索引支持最大子索引数, 当然 00 和 FF<sub>h</sub> 以及根据表 44 编码为 UNSIGNED16 (UNSIGNED8 是老的实现) 的包含数据类型的子索引。对象索引 0020<sub>h</sub> 描述了 PDO 通讯参数结构如下所示 (见对象从 1400<sub>h</sub> 到 15FF<sub>h</sub>):

表 45: 复合数据类型示例

子索引	值	(说明)
00 <sub>h</sub>	04 <sub>h</sub>	(4 个子索引)
01 <sub>h</sub>	0007 <sub>h</sub>	(UNSIGNED32)
02 <sub>h</sub>	0005 <sub>h</sub>	(UNSIGNED8)
03 <sub>h</sub>	0006 <sub>h</sub>	(UNSIGNED16)
04 <sub>h</sub>	0005 <sub>h</sub>	(UNSIGNED8)

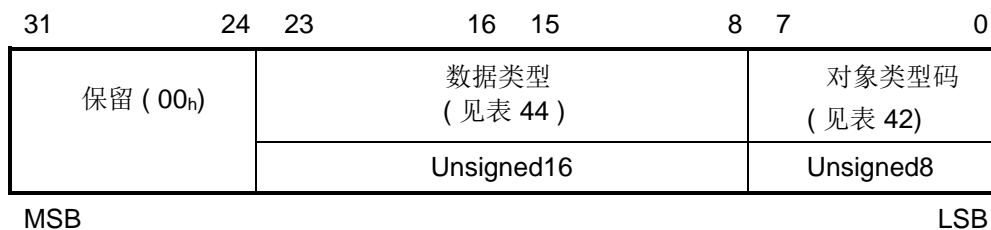
标准(简单)和复合的制造商特定数据类型可通过读取子索引 01<sub>h</sub> 来区分: 复合数据类型对象该子索引是有值的, 而标准数据类型对象该索引不存在, 因此会得到中止 SDO 传输的反馈。

请注意, 某些对象入口数据类型为 UNSIGNED32 是带有字符结构的 (例如 PDO COB-ID, 见图 67)。

#### 7.4.7.2 对象字典条目的组织结构

如果某个对象字典对象包含几个分项, 子索引 00<sub>h</sub> 描述了最大子索引, 不考虑到 FF<sub>h</sub>。其对象入口编码为 UNSIGNED8。

子索引 FF<sub>h</sub> 描述了对象类型及其结构。它被编码为 UNSIGNED32 且其组织如下:

图 51: 子索引 FF<sub>h</sub> 结构

可选支持子索引 FF<sub>h</sub>。如果支持的话就要连整个对象字典的复合数据类型结构也一并提供, 以使得可以上传整个对象字典条目结构。

#### 7.4.8 预定义复合数据类型规范

本节介绍用于通信的预定义复合数据类型结构。值的范围和含义的解释由使用这些类型的对象的详细描述给出。

##### 7.4.8.1 PDO 通讯参数记录规格

表 46 说明 PDO 通讯参数的记录

表 46: PDO 通讯参数记录

索引	子索引	名称	数据类型
0020 <sub>h</sub>	00 <sub>h</sub>	最大子索引号	UNSIGNED8
	01 <sub>h</sub>	COB - ID	UNSIGNED32
	02 <sub>h</sub>	传输类型	UNSIGNED8
	03 <sub>h</sub>	禁止时间	UNSIGNED16
	04 <sub>h</sub>	保留	UNSIGNED8
	05 <sub>h</sub>	事件定时器	UNSIGNED16
	06 <sub>h</sub>	同步启动值	UNSIGNED8

### 7.4.8.2 PDO 映射参数记录规格

表 47 表达 PDO 映射参数记录

表 47: PDO 映射参数记录

索引	子索引	名称	数据类型
0021 <sub>h</sub>	00 <sub>h</sub>	PDO 映射对象数	UNSIGNED8
	01 <sub>h</sub>	第 1 个映射的对象	UNSIGNED32
	02 <sub>h</sub>	第 2 个映射的对象	UNSIGNED32
	:	:::	:::::
	40 <sub>h</sub>	第 64 个映射的对象	UNSIGNED32

### 7.4.8.3 SDO 参数记录规格

表 48 指定 SDO 参数记录

表 48: SDO 参数记录

索引	子索引	名称	数据类型
0022 <sub>h</sub>	00 <sub>h</sub>	最大子索引支持	UNSIGNED8
	01 <sub>h</sub>	COB - ID 客户端->服务器	UNSIGNED32
	02 <sub>h</sub>	COB - ID 服务器->客户端	UNSIGNED32
	03 <sub>h</sub>	SDO 客户端应答服务器的 Node-ID	UNSIGNED8

### 7.4.8.4 身份记录规格

表 49 指定身份的记录。

表 49: 身份记录

索引	子索引	名称	数据类型
0023 <sub>h</sub>	00 <sub>h</sub>	最大子索引支持	UNSIGNED8
	01 <sub>h</sub>	Vendor-ID	UNSIGNED32
	02 <sub>h</sub>	产品码	UNSIGNED32
	03 <sub>h</sub>	修订版本号	UNSIGNED32
	04 <sub>h</sub>	序列号	UNSIGNED32

### 7.4.8.5 OS 调试记录规格

表 50 指定 OS 调试记录



表 50: OS 调试记录

索引	分指数	名称	数据类型
0024 <sub>h</sub>	00 <sub>h</sub>	最大子索引支持	UNSIGNED8
	01 <sub>h</sub>	命令	OCTET_STRING
	02 <sub>h</sub>	状态 00 - 命令已完成 - 没有错误 01 <sub>h</sub> - 命令完成 - 错误 02 <sub>h</sub> —— 预留 ::: FE <sub>h</sub> —— 预留 FF <sub>h</sub> - 命令执行	UNSIGNED8
	03 <sub>h</sub>	回复	OCTET_STRING

### 7.4.8.6 OS 命令记录规格

表 51 中指定 OS 命令的记录。

表 51: OS 命令的记录。

索引	子索引	名称	数据类型
00 时 25 <sub>h</sub>	00 <sub>h</sub>	最大子索引支持	UNSIGNED8
	01 <sub>h</sub>	命令	OCTET_STRING
	02 <sub>h</sub>	状态 00 - 命令已完成 - 没有错误 - 未回复 01 <sub>h</sub> - 命令已完成 - 没有错误 - 答复 02 <sub>h</sub> - 命令完成 - 错误 - 未回复 03 <sub>h</sub> - 命令完成 - 错误 - 回复 04 <sub>h</sub> - 预留 ::: FE <sub>h</sub> —— 预留 FF <sub>h</sub> - 命令执行	UNSIGNED8
	03 <sub>h</sub>	回复	OCTET_STRING

## 7.5 通信协议规范

### 7.5.1 对象及其条目描述规范

对象字典对象的条目结构描述遵循下面方式:所有设备协议、接口协议和应用协议都基于通信协议所使用的对象及其条目描述规则,见表 52 和表 53。

表 52: 格式化对象描述

#### 对象描述

索引	协议定义的索引号
名称	参数名称
对象类型码	变量的分类
数据类型	数据类型分类
类别	可选的或强制性的区分

对象类型码应遵循表 52 来定义。为了更好的可读性,对象描述要附带符号对象的名称。

表 53: 对象值的描述格式

## 入口描述

子索引	子项描述号
描述	描述子索引名称 ( 字段只用于数组、记录和结构)
数据类型	数据类型分类 ( 字段只用于记录和结构)
条目类别	表明对象条目是可选、强制还是条件的
访问权限	只读 ( ro ) 或读 / 写 ( rw ) 或只写 ( wo ) 或 const
PDO 映射	如果该对象可映射至 PDO 则被定义。说明： 可选：对象可以映射到 PDO 默认值：对象是默认映射的一部分 ( 请参见设备协议或应用协议 ) TPDO：对象可被映射到 TPDO 不应被映射到 RPDO RPDO：对象可被映射到 RPDO 不应该映射到 TPDO 无：对象不应映射到 PDO
值范围	可能的值的范围，或数据类型的全部取值范围
默认值	无：无默认值 协议指定：由协议指定缺省值 制造商指定：默认值由 CANopen 设备制造商指定 值：CANopen 设备初始化后的默认值

对于简单的变量其值的定义会出现一次。对于复合数据类型其值定义应包括每个元素 (子索引)。

## 7.5.2 通信协议对象的详细规范

### 7.5.2.1 对象 1000<sub>h</sub>: 设备类型

此对象提供有关设备类型信息。该对象描述了逻辑设备类型及其功能。它由两个 16 位域组成，一个描述所用设备协议或应用协议，另一个给出逻辑设备的附加功能信息。附加的信息参数为设备协议和应用协议所指定。其说明不属于本规范范围；它定义于相应的设备协议和应用协议。

#### 值定义

该值为 0000<sub>h</sub> 表示逻辑设备不遵循标准设备协议。在这种情况下附加的信息应为 0000<sub>h</sub> (如果没有更多的逻辑) 或 FFFF<sub>h</sub> (如果还有其它逻辑设备)。

多逻辑设备其附加信息应为 FFFF<sub>h</sub> 且其设备协议应为对象字典中第一逻辑设备。所有其他逻辑设备模块确定其协议于对象 67 FF<sub>h</sub> + x \* 800<sub>h</sub> 且 x = 逻辑设备内部编号 ( 从 1 到 8 ) 减去 1。这些对象将描述逻辑设备的设备类型，与对象 1000<sub>h</sub> 具有相同的值定义。

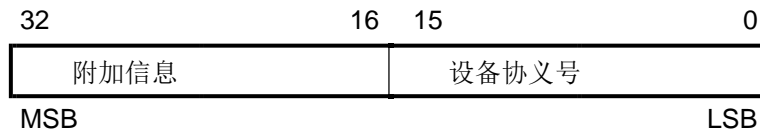


图 52: 设备类型参数结构

**对象描述**

索引	1000 <sub>h</sub>
名称	设备类型
对象类型码	VAR
数据类型	UNSIGNED32
类别	强制

**入口描述**

子索引	00 <sub>h</sub>
访问权限	ro
PDO 映射	否
值范围。	请参阅 <i>值定义</i>
默认值	协议或制造商指定

**7.5.2.2 对象 1001<sub>h</sub>: 错误寄存器**

此对象提供错误信息。CANopen 设备将内部错误记录映射到该对象。这是紧急对象的一部分。

**值定义**

表 54: 错误寄存器结构

位	M/O	意义
0	M	通用错误
1	O	电流
2	O	电压
3	O	温度
4	O	通信错误 (溢出、错误状态)
5	O	设备协议指定
6	O	保留 (始终为 0 <sub>b</sub> )
7	O	制造商指定

如果某一特定的错误发生相应位置 1<sub>b</sub>。通用错误位必须支持。其它可选。任何错误产生都将置位通用错误。

**对象描述**

索引	1001 <sub>h</sub>
名称	错误寄存器
对象类型码	VAR
数据类型	UNSIGNED8
类别	强制

## 入口描述

子索引	00 <sub>h</sub>
访问权限	ro
PDO 映射	可选
值范围。	请参阅 <i>值定义</i>
默认值	无

### 7.5.2.3 对象 1002<sub>h</sub>: 制造商状态寄存器

此对象为制造商提供一个常规状态寄存器。本规范仅定义此对象大小和位置。

## 对象描述

索引	1002 <sub>h</sub>
名称	制造商状态寄存器
对象类型码	VAR
数据类型	UNSIGNED32
类别	可选

## 入口描述

分指数	00 <sub>h</sub>
访问权限	ro
PDO 映射	可选
值范围。	UNSIGNED32
默认值	无

### 7.5.2.4 对象 1003<sub>h</sub>: 预定义错误域

此对象表达 CANopen 设备上产生的错误,并由紧急对象标示。且提供错误历史记录。

## 值定义

- 该对象子索引 00<sub>h</sub> 为从 01<sub>h</sub> 起始的实际的错误记录数量。  
备注：若无错误存在其子索引 00<sub>h</sub> 值为 00<sub>h</sub>，读子索引 01<sub>h</sub> 响应到 SDO 中止消息（中止代码：0800 0024 或 0800 0000）。
- 每个新的错误应储存在子索引 01<sub>h</sub>；较旧的错误应依次下移。
- 写 00<sub>h</sub> 到子索引 00<sub>h</sub> 将删除整个错误历史记录(清空数组)。其余非 00<sub>h</sub> 值均不允许,将导致中止信息(错误代码:0609 0030<sub>h</sub>)。
- 错误号码的类型为 UNSIGNED32(见表 26)，由 16 位错误代码和 16 位制造商指定的附加错误信息组成。错误代码位于低 2 字节(LSB)，附加信息位于高 2 字节(MSB)。支持此对象应至少包含两个条目。长度子索引 00<sub>h</sub> 和至少一个错误项 01<sub>h</sub>。

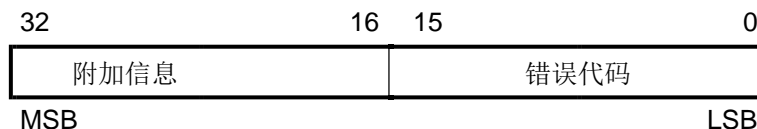


图 53: 预定义错误域结构

## 对象描述

索引	1003 <sub>h</sub>
名称	预定义错误域
对象类型码	数组
数据类型	UNSIGNED32
类别	可选

## 入口描述

子索引	00 <sub>h</sub>
描述	错误数
条目类别	强制
访问权限	rw
PDO 映射	否
值范围。	00 <sub>h</sub> 到 FE <sub>h</sub>
默认值	00 <sub>h</sub>

子索引	01 <sub>h</sub>
描述	标准错误域
条目类别	强制
访问权限	ro
PDO 映射	否
值范围。	UNSIGNED32
默认值	无

分指数	02 <sub>h</sub> 到 FE <sub>h</sub>
描述	标准错误域
条目类别	可选
访问权限	ro
PDO 映射	否
值范围。	UNSIGNED32
默认值	无

### 7.5.2.5 对象 1005<sub>n</sub>: COB - ID 同步消息

此对象为已配置的同步( SYNC )对象 COB-ID。此外,它定义了 CANopen 设备是否生成同步。对象结构定义于图 54 和表 55。

值定义

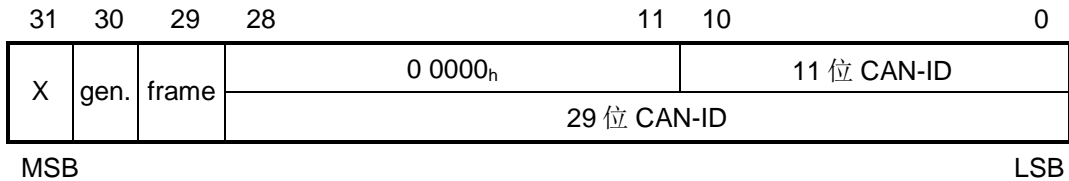


图 54: 同步 COB-ID 结构

表 55: 同步 COB-ID 描述

位 ( S )	值	描述
x	x	任意
gen.	0 <sub>b</sub>	所在 CANopen 设备不生成同步消息
	1 <sub>b</sub>	CANopen 设备生成同步消息
frame	0 <sub>b</sub>	11 位 CAN-ID 有效 (CAN 标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (CAN 扩展帧)
29 位 CAN-ID	x	29 位扩展帧 CAN-ID
11 位 CAN-ID	x	11 位标准帧 CAN-ID

位 29 (frame) 和位 30(gen.)可以是静态的(不可更改)。如果 CANopen 设备不生成同步消息,试图设置 30 位(gen.)为 1<sub>b</sub>将得到 SDO 中止传输的应答(中止代码:0609 0030<sub>n</sub>)。如果 CANopen 设备仅支持标准帧,则试图置位 29(frame)为 1<sub>b</sub>将得到中止传输的应答 (中止代码 : 0609 0030<sub>n</sub>)。设置位 30 至 1<sub>b</sub>并且同步计数器的溢出值大于 0, 第一次 SYNC 消息在计数器值复位为 1 时启动。对象有效时 ( 位 30 = 1<sub>b</sub>)不允许更改 0~29 位。

对象描述

索引	1005 <sub>n</sub>
名称	同步 COB-ID
对象类型码	VAR
数据类型	UNSIGNED32
类别	条件的; 强制的, 如果 PDO 通信支持基于同步方式

入口说明

子索引	00 <sub>n</sub>
访问权限	rw ; Const 如果 COB - ID 不可更改
PDO 映射	否
值范围。	请参阅 值定义
默认值	0000 0080 <sub>n</sub> 或 8000 0080 <sub>n</sub>

### 7.5.2.6 对象 1006<sub>h</sub>: 通信循环周期

此对象定义 SYNC 时间间隔。

#### 值定义

该值单位是  $\mu\text{s}$ 。如果将该值置 0000 0000<sub>h</sub> 则 SYNC 消息被禁用。通过更改值为非 0000 0000<sub>h</sub> 并且同步计数器的溢出值大于 0，第一次 SYNC 消息在计数器值复位为 1 时启动。SYNC 消息将在值更新后一个通信循环周期内启动传输。

#### 对象描述

索引	1006 <sub>h</sub>
名称	通信循环周期
对象类型码	VAR
数据类型	UNSIGNED32
类别	有条件的; 同步生产者为强制的

#### 入口描述

子索引	00 <sub>h</sub>
访问权限	rw
PDO 映射	否
值范围	UNSIGNED32
默认值	0000 0000 <sub>h</sub>

### 7.5.2.7 对象 1007<sub>h</sub>: 同步窗长度

此对象表达同步 PDOs 时窗长度配置。

如果同步窗过期，所有的 TPDOs 将被丢弃并且会产生 EMCY 消息;所有的 RPDOs 也被丢弃直到下次同步消息的接收。同步的 RPDO 处理在下一个同步消息到来后恢复。

#### 值定义

该值单位是  $\mu\text{s}$ 。如果将该值设置为 0000 0000<sub>h</sub>，同步窗口将被禁用。

#### 对象描述

索引	1007 <sub>h</sub>
名称	同步窗长度
对象类型码	VAR
数据类型	UNSIGNED32
类别	可选的

## 入口描述

子索引	00 <sub>h</sub>
访问权限	rw
PDO 映射	否
值范围	UNSIGNED32
默认值	0000 0000 <sub>h</sub>

**7.5.2.8 对象 1008<sub>h</sub>: 制造商设备名称**

此对象为制造商的设备命名。

## 对象描述

索引	1008 <sub>h</sub>
名称	制造商设备名称
对象类型码	VAR
数据类型	VISIBLE_STRING
类别	可选

## 入口描述

子索引	00 <sub>h</sub>
访问权限	const
PDO 映射	否
取值范围	VISIBLE_STRING
默认值	制造商指定

**7.5.2.9 对象 1009<sub>h</sub>: 制造商的硬件版本**

此对象为制造商提供的硬件版本

## 对象描述

索引号	1009 <sub>h</sub>
名称	制造商硬件版本
对象类型码	VAR
数据类型	VISIBLE_STRING
类别	可选

## 入口描述

子索引	00 <sub>h</sub>
访问权限	const
PDO 映射	No
取值范围	VISIBLE_STRING
默认值	制造商指定



### 7.5.2.10 对象 100A<sub>n</sub>: 制造商软件版本

此对象为制造商提供的软件版本

#### 对象描述

索引号	100A <sub>n</sub>
名称	制造商软件版本
对象类型码	VAR
数据类型	VISIBLE_STRING
类别	可选

#### 入口描述

子索引	00 <sub>n</sub>
访问权限	const
PDO 映射	No
取值范围	VISIBLE_STRING
默认值	制造商指定

### 7.5.2.11 对象 100C<sub>n</sub>: 监护周期

索引对象 100C<sub>n</sub> 和 100D<sub>n</sub> 是生存周期因子，前者配置了监护周期。其乘积给出了生存监护协议的生存周期。

#### 值定义

该值单位为 ms。该值为 0000<sub>n</sub> 时禁用生存监护。

#### 对象描述

索引	100C <sub>n</sub>
名称	监护周期
对象类型码	VAR
数据类型	UNSIGNED16
类别	有条件的； 如果支持节点保护则为强制的

#### 入口描述

分指数	00 <sub>n</sub>
访问权限	rw； ro，如果不支持寿命保护
PDO 映射	否
值范围	UNSIGNED16
默认值	0000 <sub>n</sub>

### 7.5.2.12 对象 100D<sub>h</sub>: 生存周期因子

生存周期因子和监护周期的乘积给出了生存监护协议的生存周期。

#### 值定义

该值为 00<sub>h</sub> 应禁用生存监护。

#### 对象描述

索引	100D <sub>h</sub>
名称	生存周期因子
对象类型码	VAR
数据类型	UNSIGNED8
类别	有条件的； 如果支持节点监护，是强制性的。

#### 入口描述

分指数	00 <sub>h</sub>
访问权限	rw ro, 如果不支持生存监护
PDO 映射	否
值范围。	UNSIGNED8
默认值	00 <sub>h</sub>

### 7.5.2.13 对象 1010<sub>h</sub>: 保存参数

此对象控制保存参数到非易失性存储器中。

#### 值定义

为了访问 CANopen 设备，应提供相关信息的保存能力。  
几个参数组的分类：

- 子索引 00<sub>h</sub> 包含子索引数。
- 子索引 01<sub>h</sub> 指的是所有可存储在 CANopen 设备上的参数。
- 子索引 02<sub>h</sub> 指的是通信相关的参数 (索引从 1000<sub>h</sub> 到 1 FFF<sub>h</sub>)。
- 子索引 03<sub>h</sub> 指的是应用相关的参数 (索引从 6000<sub>h</sub> 到 9 FFF<sub>h</sub>)。
- 子索引从 04<sub>h</sub> 到 7F<sub>h</sub> 制造商可选的单独的参数。
- 子索引从 80<sub>h</sub> 到 FE<sub>h</sub> 为将来的使用而保留。

为了避免存储的参数错误，存储应仅当特定暗号写到相应的子索引才被执行。暗号必须是“save”：

	暗号	MSB			LSB
/ISO8859/字符		e	v	a	s
hex		65 <sub>h</sub>	76 <sub>h</sub>	61 <sub>h</sub>	73 <sub>h</sub>

图 55: 存储权限暗号

在相应子索引接收到正确的暗号后，CANopen 设备储存参数然后通过 SDO 确认(SDO 下载启动响应)。如果存储失败，CANopen 设备响应 SDO 中止(中止代码：0606 0000<sub>h</sub>)。

如果写入暗号有误，CANopen 设备拒绝保存，并且响应 SDO 中止传输(中止代码：0800 002x<sub>h</sub>)。

通过读访问，CANopen 设备提供存储功能相关信息，格式如下：



图 56: 读结构

表 56: 读访问结构

位	值	描述
auto	0 <sub>b</sub>	CANopen 设备不自动保存参数
	1 <sub>b</sub>	CANopen 设备自动保存参数
cmd	0 <sub>b</sub>	CANopen 设备不接受保存参数命令
	1 <sub>b</sub>	CANopen 设备按命令执行保存参数动作

自动保存意味着 CANopen 设备参数的非易失存储无需用户请求。

#### 对象描述

索引	1010 <sub>h</sub>
名称	存储参数
对象类型码	ARRAY
数据类型	UNSIGNED32
类别	可选

#### 入口描述

子索引	00 <sub>h</sub>
描述	最高子索引数
条目类别	强制
访问权限	Const
PDO 映射	否
取值范围	01 <sub>h</sub> ~7F <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	保存所有参数。
条目类别	强制
访问权限	rw ro, 如果支持自动保存
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 55 为写入; 图 56 为只读)
默认值	协议或制造商指定

子索引	02 <sub>h</sub>
描述	保存的通信参数
条目类别	可选
访问权限	rw ro, 如果支持自动保存
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 55 为写入;图 56 为只读访问权限)
默认值	协议或制造商指定

子索引	03 <sub>h</sub>
描述	保存应用参数
条目类别	可选
访问权限	rw ro, 如果支持自动保存
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 55 为写入;图 56 为只读访问权限)
默认值	协议或制造商指定

分指数	04 <sub>h</sub> ~7F <sub>h</sub>
描述	保存制造商定义的参数
条目类别	可选
访问权限	rw ro, 如果支持自动保存
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 55 为写入;图 56 为只读访问权限)
默认值	协议或制造商指定

### 7.5.2.14 对象 1011<sub>h</sub>: 恢复缺省参数

通过该对象根据协议、设备协议和应用协议恢复参数默认值。

#### 值定义

通过读取访问, CANopen 设备提供有关恢复相关值的功能信息。其参数组的分别:

- 子索引 00<sub>h</sub> 包含支持的最大子索引数。
- 子索引 01<sub>h</sub> 指恢复所有参数。
- 子索引 02<sub>h</sub> 指恢复通信相关参数 (索引从 1000<sub>h</sub> 到 1FFF<sub>h</sub>)。
- 子索引 03<sub>h</sub> 恢复应用相关参数 (索引从 6000<sub>h</sub> 到 9FFF<sub>h</sub>)。
- 子索引从 04<sub>h</sub> 到 7F<sub>h</sub> 为制造商指定的恢复特定参数。
- 子索引从 80<sub>h</sub> 到 FE<sub>h</sub> 为将来的使用而保留。

为了避免误动作，恢复应仅当写入暗号到相应的子索引有效。暗号为“load”:

暗号	MSB			LSB
/ISO8859/ 字符	d	a	o	l
hex	64 <sub>h</sub>	61 <sub>h</sub>	6F <sub>h</sub>	6C <sub>h</sub>

图 57: 恢复默认值的写入暗号

CANopen 设备接收到正确的暗号后就会恢复相应参数值为默认值，然后通过 SDO 应答确认(SDO 下载启动应答)。如果恢复失败会中止传输(中止代码:0606 0000<sub>h</sub>)。如果暗号写入错误，CANopen 设备应拒绝恢复操作并应答 SDO 中止传输(中止代码: 0800 002x<sub>h</sub>)。

CANopen 设备复位(对于子索引 01<sub>h</sub> 到 7 F<sub>h</sub> 执行 NMT 复位节点服务,对于子索引 02<sub>h</sub> 需要执行 NMT 复位通信服务) 或掉电重启后默认值生效。

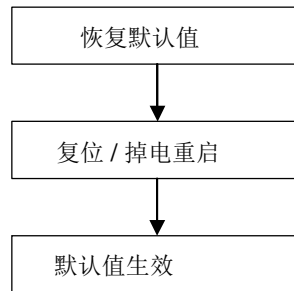


图 58: 恢复默认值过程

通过读访问，CANopen 设备提供恢复参数功能相关信息，格式如下:



图 59: 恢复默认的读访问结构

表 57:恢复默认的读访问结构

位	值	描述
cmd	0 <sub>b</sub>	CANopen 设备不支持恢复默认参数
	1 <sub>b</sub>	CANopen 设备支持恢复参数

对象描述

索引	1011 <sub>h</sub>
名称	恢复缺省参数
对象类型码	ARRAY
数据类型	UNSIGNED32
类别	可选

子索引	00 <sub>h</sub>
描述	最高子索引数
条目类别	强制
访问权限	Const
PDO 映射	否
取值范围	01 <sub>h</sub> ~7F <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	恢复所有默认的参数。
条目类别	强制
访问权限	rw
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 57 为写访问;图 59 为读访问)
默认值	协议或制造商指定

子索引	02 <sub>h</sub>
描述	恢复默认的通信参数
条目类别	可选
访问权限	rw
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 57 为写访问;图 59 为读访问)
默认值	协议或制造商指定

子索引	03 <sub>h</sub>
描述	恢复应用的默认参数
条目类别	可选
访问权限	rw
PDO 映射	否
取值范围	请参阅 <i>值定义</i> (图 57 为写访问;图 59 为读访问)
默认值	协议或制造商指定

子索引	04 <sub>h</sub> ~7F <sub>h</sub>
描述	还原制造商定义的默认参数。
条目类别	可选
PDO 映射	否
取值范围	请参阅 值定义 (图 57 为写访问;图 59 为读访问)
默认值	协议或制造商指定

### 7.5.2.15 对象 1012<sub>h</sub>: 时间戳对象 COB-ID

此对象为时间戳对象(TIME)的 COB-ID 配置。此外，它还定义了 CANopen 设备是否消费 TIME 或是否生成 TIME。对象结构见图 60 和表 58。

值定义

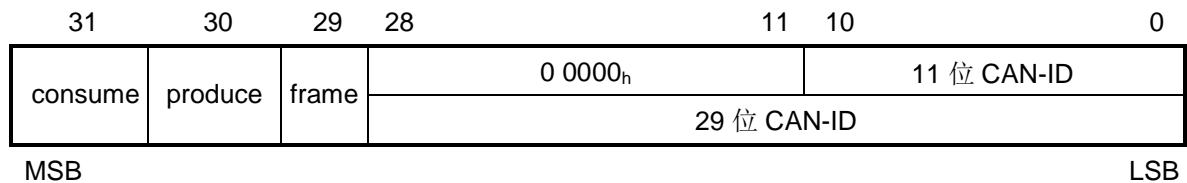


图 60: TIME COB-ID 结构

表 58:TIME COB-ID 说明

位(们)	值	描述
consume	0 <sub>b</sub>	CANopen 设备不消费 TIME 消息
	1 <sub>b</sub>	CANopen 设备消费 TIME 消息
produce	0 <sub>b</sub>	CANopen 设备不会产生 TIME 消息
	1 <sub>b</sub>	CANopen 设备产生 TIME 消息
frame	0 <sub>b</sub>	11 CAN-ID 有效(CAN 标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (CAN 扩展帧)
29 位 CAN-ID	x	29 位 CAN-ID 的扩展帧
11 位 CAN-ID	x	11 位 CAN-ID 的标准帧

位 29 (frame)、30 (produce)可以是静态的(不能更改)。如果 CANopen 设备不支持生成 TIME，试图设置位 30(produce) 1<sub>b</sub> 应答 SDO 中止传输(中止代码:0609 0030<sub>h</sub>)。如果 CANopen 设备仅支持标准帧，试图设置位 29(frame)1<sub>b</sub> 应答 SDO 中止传输(中止代码:0609 0030<sub>h</sub>)。当此对象存在(位 30 = 1<sub>b</sub> 或位 31 = 1<sub>b</sub>)时位 0 至 29 不可更改。

对象描述

索引	1012 <sub>h</sub>
名称	时间戳 COB-ID
对象类型码	VAR
数据类型	UNSIGNED32
类别	可选的

子索引	00 <sub>h</sub>
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED32
默认值	CAN-ID : 100 <sub>h</sub> frame: 0 <sub>b</sub> 有效: 协议或制造商指定

### 7.5.2.16 对象 1013<sub>h</sub>: 高分辨率时间戳

此对象表达已配置的高分辨率时间戳。它可能会被映射到 PDO,以交换高分辨率时间戳的消息。鼓励更深入的应用。

#### 值定义

该值单位 $\mu\text{s}$ 。

#### 对象描述

索引	1013 <sub>h</sub>
名称	高分辨率时间戳
对象类型码	VAR
数据类型	UNSIGNED32
类别	可选

#### 入口说明

子索引	00 <sub>h</sub>
访问权限	rw; ro, 如果仅支持生成高分辨率时间戳 rw 或 wo, 如果仅支持高分辨率时间戳消费
PDO 映射	可选
取值范围	UNSIGNED32
默认值	0

### 7.5.2.17 对象 1014<sub>h</sub>: EMCY COB-ID

此对象表达已配置的 EMCY 写服务 COB-ID。

#### 值定义

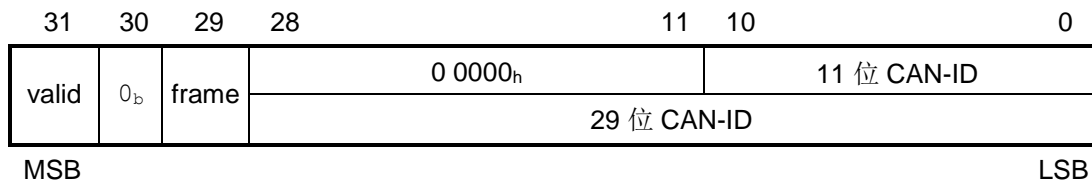


图 61: EMCY 标识结构



表 59: EMCY COB-ID 说明

位 (们)	值	描述
valid	0 <sub>b</sub>	EMCY 存在/有效
	1 <sub>b</sub>	EMCY 不存在/无效
30	0 <sub>b</sub>	保留 ( 始终为 0 <sub>b</sub> )
frame	0 <sub>b</sub>	11 CAN-ID 有效(CAN 标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (CAN 扩展帧 )
29 位 CAN-ID	x	29 位 CAN-ID 的扩展帧
11 位 CAN-ID	x	11 位 CAN-ID 的标准帧

仅支持标准帧的 CANopen 设备企图置位 29(frame)为 1<sub>b</sub>将得到 SDO 中止传输响应(中止代码:0609 0030<sub>h</sub>)。当紧急对象有效(位 31 = 0<sub>b</sub>)时, 位 0 至 29 禁止更改。

#### 对象描述

索引	1014 <sub>h</sub>
名称	紧急消息 COB-ID
对象类型码	VAR
数据类型	UNSIGNED32
类别	有条件的 ; 强制性的, 如果支持紧急对象

#### 入口说明

子索引	00 <sub>h</sub>
访问权限	rw ; const 如果 COB-ID 不可更改
PDO 映射	否
取值范围	UNSIGNED32
默认值	CAN-ID : 80 <sub>h</sub> + Node-ID frame: 0 <sub>b</sub> valid: 协议或制造商指定

### 7.5.2.18 对象 1015<sub>h</sub>: EMCY 禁止时间

此对象表达 EMCY 消息禁止时间的配置。

#### 值定义

该值单位 100 $\mu$ s。值为 0 将禁用禁止时间。

#### 对象描述

索引	1015 <sub>h</sub>
名称	EMCY 禁止时间
对象类型码	VAR
数据类型	UNSIGNED16
类别	可选的

分指数	00 <sub>h</sub>
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED16
默认值	0

### 7.5.2.19 对象 1016<sub>h</sub>: 消费者心跳超时

该对象表达期望的心跳周期。监测心跳生产者从接收第一次心跳开始。

注意：消费者心跳超时应大于相应的生产者的心跳周期。

注：在检测到生产者的第一次心跳前其状态未知。

#### 值定义

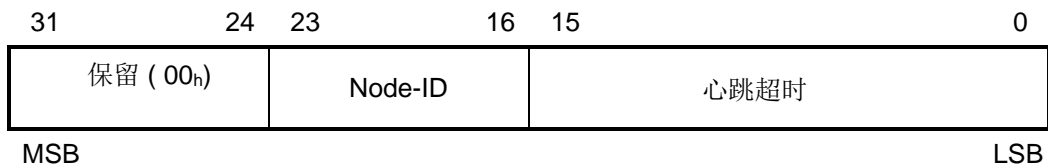


图 62: 消费者心跳时间结构

如果心跳时间为 0 或者 Node-ID 为 0 或大于 127，则对应的对象条目无效。心跳超时单位 ms。

为相同的 node-ID 尝试配置多个非 0 心跳时间，CANopen 设备将响应 SDO 中止传输(中止代码:0604 0043<sub>h</sub>)。

#### 对象描述

索引	1016 <sub>h</sub>
名称	消费者心跳超时
对象类型码	ARRAY
数据类型	UNSIGNED32
类别	可选的

#### 入口说明

子索引	00 <sub>h</sub>
描述	最大子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> ~7F <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	消费者心跳超时
条目类别	强制
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED32 ( 图 62 )
默认值	0000 0000 <sub>h</sub>

分指数	02 <sub>h</sub> ~7F <sub>h</sub>
描述	消费者心跳超时
条目类别	可选
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED32 ( 图 62 )
默认值	0000 0000 <sub>h</sub>

### 7.5.2.20 对象 1017<sub>h</sub>: 生产者心跳超时

该对象表达心跳周期的配置。

#### 值定义

该值单位 ms 。 0 值将禁用生产者心跳。

#### 对象描述

索引	1017 <sub>h</sub>
名称	生产者心跳超时
对象类型码	VAR
数据类型	UNSIGNED16
类别	有条件的； 强制性的，如果不支持监护

#### 入口说明

分指数	00 <sub>h</sub>
访问权限	rw const, 如果默认值为协议指定且不可更改
PDO 映射	否
取值范围	UNSIGNED16
默认值	0 或协议指定

### 7.5.2.21 对象 1018<sub>h</sub>: 对象身份

此对象应提供 CANopen 设备的身份识别信息。

值定义

子索引 01<sub>h</sub> 应包含有独特的值<sup>1</sup>为每个 CANopen 设备供应商所独有。值 0000 0000<sub>h</sub> 应表明无效的供应商标识。

子索引 02<sub>h</sub> 应包含有独特的值, 来识别特定类型的 CANopen 设备。值 0000 0000<sub>h</sub> 应被保留。

子索引 03<sub>h</sub> 应包含 CANopen 设备的主修订版本号 and 次修订版本号(参见图 63)。主修订版本号应指明特定的 CANopen 行为。这意味着如果 CANopen 功能发生变化, 主修订版本号应增加。次修订版本号应识别具有相同 CANopen 行为的不同版本的 CANopen 设备。值 0000 0000<sub>h</sub> 应被保留。

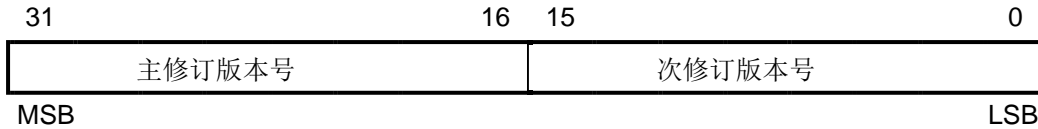


图 63: 修订版本号结构

子索引 04<sub>h</sub> 包含序列号, 为某一组产品和特定版本 CANopen 设备的唯一标识。值 0000 0000<sub>h</sub> 应被保留。

对象描述

索引	1018 <sub>h</sub>
名称	对象身份
对象类型码	RECORD
数据类型	Identity
类别	强制性

输入说明。

子索引	00 <sub>h</sub>
描述	最大子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> 到 04 <sub>h</sub>
默认值	协议或制造商指定

<sup>1</sup> 值由 CAN in Automation ( CiA ) 唯一分配。

子索引	01 <sub>h</sub>
描述	Vendor-ID
条目类别	强制性
访问权限	ro
PDO 映射	否
值范围。	UNSIGNED32
默认值	由 CiA 唯一分配给制造商

子索引	02 <sub>h</sub>
描述	产品码
条目类别	可选的
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED32
默认值	协议或制造商指定

子索引	03 <sub>h</sub>
描述	修订版本号
条目类别	可选的
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED32
默认值	协议或制造商指定

分指数	04 <sub>h</sub>
描述	序列号
条目类别	可选的
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED32
默认值	协议或制造商指定

### 7.5.2.22 对象 1019<sub>n</sub>: 同步计数器溢出值

此对象表达同步计数器支持的最大值配置。如果 CANopen 设备支持同步计数器，生产者和消费者应支持此对象。如果该值大于 1，同步信息应带有数据长度为 1 字节的数据。SYNC 消费者应忽略该值本身。一旦 SYNC 消费者发现接收的 SYNC 消息数据长度与配置的不同将发送 EMCY 消息( 错误代码：8240<sub>h</sub> - 意外的同步数据长度)。

## 值定义

值	描述
0	SYNC 消息数据长度为 0
1	保留
2~240	SYNC 消息数据长度为 1。第一个数据字节包含计数器
241~255	保留的

该值应为所有 TPDO 的传输类型 (  $1 < n \leq 240$  ) 的最小公倍数。从而可以确保 SYNC 事件总是发生于 SYNC 循环的同一计数值。

一旦 SYNC 循环周期非 0，对本索引值的更改将触发 SDO 中止(中止代码:0800 0022<sub>h</sub> 或 0800 0000<sub>h</sub>)。

## 对象描述

索引	1019 <sub>h</sub>
名称	同步计数器溢出值
对象的代码。	VAR
数据类型	UNSIGNED8
类别	可选的

## 入口说明

子索引	00 <sub>h</sub>
访问权限	rw ; const , 如果协议指定默认值不可更改
PDO 映射	否
取值范围	UNSIGNED8
默认值	0 或协议指定

### 7.5.2.23 对象 1020<sub>h</sub>: 验证配置

此对象保存下载配置的日期和时间。如果 CANopen 设备支持保存参数到非易失性存储器，网络配置工具或 CANopen 管理器使用该对象在设备复位后验证配置，检查是否需要重新配置。配置工具于该对象存储日期和时间同时存入 DCF 相同值。现在配置工具通过向 CANopen 设备索引 1010<sub>h</sub> 的子索引 01<sub>h</sub> 写入 “save” 来保存配置。CANopen 设备复位后应能自动或由请求来恢复最近一次的配置和识别标记。如果有任何其他命令更改了启动配置值，CANopen 设备均应重置验证配置对象为 0。

配置管理器将比较识别标记和 DCF 中的配置值，决定是否需要重置。

注意：使用此对象需允许超前于启动过程的识别标识。如果使用该对象，系统集成商能够识别出用户更改了配置，并且随后激活了 1010<sub>h</sub> 的保存命令，但并未更改 1020<sub>h</sub>。因而系统集成商通过使用该功能可以 100 % 确认上述情况。

## 值定义

子索引 01<sub>h</sub> (配置日期) 应包含自 1984 年 1 月 1 日的天数。

子索引 02<sub>h</sub> (配置时间) 为自午夜 的 ms 数。

## 对象描述

索引	1020 <sub>h</sub>
名称	验证配置
对象类型码	ARRAY
数据类型	UNSIGNED32
类别	可选的

## 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	02 <sub>h</sub>
默认值	02 <sub>h</sub>

子索引	01 <sub>h</sub>
描述	配置日期
条目类别	强制性
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED32
默认值	制造商指定

分指数	02 <sub>h</sub>
描述	配置时间
条目类别	强制性
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED32
默认值	制造商指定

### 7.5.2.24 对象 1021<sub>h</sub>: 存储 EDS

此对象应描述装载的 EDS。EDS 文件存储于 CANopen 设备上具有以下优势：

- 制造商不存在通过磁盘分发 EDS 的问题。
- 管理不同软件版本的不同 EDS 版本不容易出错，如果它们是存放在一起。
- 完整的网络配置保存在网络上。这使得分析或重新配置网络更方便并且对用户更透明。

#### 值定义

文件名不需要存储，因为每个 EDS 文件包含其自身文件名。

## 对象描述

索引	1021 <sub>h</sub>
名称	EDS 的存储
对象的代码。	VAR
数据类型	域
类别	可选的

## 入口说明

子索引	00 <sub>h</sub>
访问权限	ro
PDO 映射	否
值范围。	制造商指定
默认值	无

7.5.2.25 对象 1022<sub>h</sub>: 存储格式

该对象应指明存储格式。允许使用的压缩格式。对象仅描述外部行为。

## 值定义

表 60: EDS 存储格式

值	描述
00 <sub>h</sub>	/ ISO 10646 / 不压缩
01 <sub>h</sub>	保留的
::: ::	::: ::
7F <sub>h</sub>	保留的
80 <sub>h</sub>	制造商指定
::: ::	::: ::
FF <sub>h</sub>	制造商指定

## 对象描述

索引	1022 <sub>h</sub>
名称	存储格式
对象类型码	VAR
数据类型	UNSIGNED16
类别	有条件的； 如果支持存储 EDS 则强制

## 入口说明

子索引	00 <sub>h</sub>
访问权限	ro
PDO 映射	否
值范围。	UNSIGNED8
默认值	无



### 7.5.2.26 对象 1023<sub>h</sub>: OS 命令

该对象将用作可编程设备的命令驱动接口。命令内容为 / ISO8859 / 字符或由制造商完全指定的二进制。由主系统将 OS 命令传给对象。

#### 值定义

如果 CANopen 设备实现此功能，必须支持所有子索引，附加的对象条目由制造商指定。当状态范围从 0 到 3 表示有新的命令输入：命令和所有参数应作为一整块传入子索引 01<sub>h</sub>。命令传输完成应立即执行。主机轮询子索引 02<sub>h</sub>，直到其值从 0 到 3。如果状态值是 1 或 3，设备会应答。如果需要多次应答，或状态从 1 到 0 或 3 到 2，或不能缓存应答，CANopen 设备应返回相同的回复。

#### 对象描述

索引	1023 <sub>h</sub>
名称	OS 命令
对象的代码。	RECORD
数据类型	OS command record
类别	可选的

#### 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	03 <sub>h</sub>
默认值	03 <sub>h</sub>

子索引	01 <sub>h</sub>
描述	命令
条目类别	强制性
访问权限	rw; wo、如果只支持写
PDO 映射	否
取值范围	无
默认值	制造商指定

子索引	02 <sub>h</sub>
描述	状态
条目类别	强制性
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED8
默认值	无

子索引	03 <sub>h</sub>
描述	回复
条目类别	强制性
访问权限	ro
PDO 映射	否
取值范围	无
默认值	无

### 7.5.2.27 对象 1024<sub>h</sub>: OS 命令模式

此对象应控制应用指定队列的命令执行。其目的是希望通过该对象指导应用指定队列中最近的命令。

#### 值定义

表 61: OS 命令模式值

值	描述
00 <sub>h</sub>	立即执行下一条命令
01 <sub>h</sub>	缓冲下一条命令
02 <sub>h</sub>	执行缓冲区中的命令
03 <sub>h</sub>	中止当前的命令和缓冲区中所有命令
04 <sub>h</sub>	制造商指定
.....	.....
FF <sub>h</sub>	制造商指定

#### 对象描述

索引	1024 <sub>h</sub>
名称	OS 命令模式
对象类型码	VAR
数据类型	UNSIGNED8
类别	可选的

## 入口说明

分指数	00 <sub>h</sub>
访问权限	wo
PDO 映射	否
取值范围	UNSIGNED8
默认值	制造商指定

7.5.2.28 对象 1025<sub>h</sub>: OS 调试接口

此对象对应 OS 调试接口，是可编程 CANopen 设备调试器的二进制命令接口。命令内容由制造商指定。该对象允许用户连接到远程的调试器。

## 值定义

参阅 OS 命令

## 对象描述

索引	1025 <sub>h</sub>
名称	OS 调试接口
对象类型码	RECORD
数据类型	OS debug record
类别	可选

## 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	03 <sub>h</sub>
默认值	03 <sub>h</sub>

子索引	01 <sub>h</sub>
描述	命令
条目类别	强制性
访问权限	rw; wo, 如果仅支持只写
PDO 映射	否
取值范围	无
默认值	制造商指定

子索引	02 <sub>h</sub>
描述	状态
条目类别	强制性
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED8
默认值	无

子索引	03 <sub>h</sub>
描述	回复
条目类别	强制性
访问权限	ro
PDO 映射	否
取值范围	无
默认值	无

### 7.5.2.29 对象 1026<sub>h</sub>: OS 提示符命令接口

该对象是可编程 CANopen 设备的字符驱动命令接口。命令内容由制造商指定。这个对象允许用户实现远程键盘控制。

值定义

子索引 01<sub>h</sub> StdIn 是通过 SDO 或 PDO 用来向 CANopen 设备传输单个字符。每个新字符均被加入内部输入队列。CANopen 设备应答通过子索引 02<sub>h</sub> Stdout 输出。该对象通常被映射到一个事件驱动的 PDO 或轮询 SDO 中。子索引 03<sub>h</sub> StdErr 用于错误输出。该对象通常被映射到一个事件驱动的 PDO 或轮询 SDO 中。

对象描述

索引	1026 <sub>h</sub>
名称	OS 提示符命令接口
对象类型码	ARRAY
数据类型	UNSIGNED8
类别	可选

入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	02 <sub>h</sub> ~03 <sub>h</sub>
默认值	无

子索引	01 <sub>h</sub>
描述	StdIn
条目类别	强制
访问权限	wo
PDO 映射	可选
取值范围	UNSIGNED8
默认值	制造商指定

子索引	02 <sub>h</sub>
描述	StdOut。
条目类别	强制
访问权限	ro
PDO 映射	可选
取值范围	UNSIGNED8
默认值	无

子索引	03 <sub>h</sub>
描述	StdErr。
条目类别	可选
访问权限	ro
PDO 映射	可选
取值范围	UNSIGNED8
默认值	无

### 7.5.2.30 对象 1027<sub>h</sub>: 模块列表

总线耦合器允许连接若干模块的组合，该对象为其提供描述模块化 CANopen 设备的通用方法。提供当前已连接模块的信息列表。

值定义

连续的子索引 (1 " N " 254)描述了相应模块的顺序连接。每个对象的条目包含了模块的识别号。此识别号能够唯一区分连到总线耦合器上的所有设备类型。

该对象的条目子索引 00<sub>h</sub>为实际连接到总线耦合器上的模块数量。

注：如果没有模块存子索引 00<sub>h</sub>的值为 00<sub>h</sub>，读取子索引 01<sub>h</sub>响应 SDO 中止消息 (中止代码: 0800 0024<sub>h</sub>或 0800 0000<sub>h</sub>)。

对象描述

索引	1027 <sub>h</sub>
名称	模块列表
对象类型码	ARRAY
数据类型	UNSIGNED16
类别	有条件的; 强制,如果支持模块化的设备

## 入口说明

子索引	00 <sub>h</sub>
描述	连接模块数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> ~FE <sub>h</sub>
默认值	无

子索引	01 <sub>h</sub>
描述	模块 1
条目类别	强制
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED16
默认值	无

子索引	02 <sub>h</sub> ~FE <sub>h</sub>
描述	模块 2 至模块 254
条目类别	可选
访问权限	ro
PDO 映射	否
取值范围	UNSIGNED16
默认值	无

**7.5.2.31 对象 1028<sub>n</sub>: 紧急消费对象**

此对象应表明该 CANopen 设备消费的 EMCY 对象的 COB-ID 配置。

## 值定义

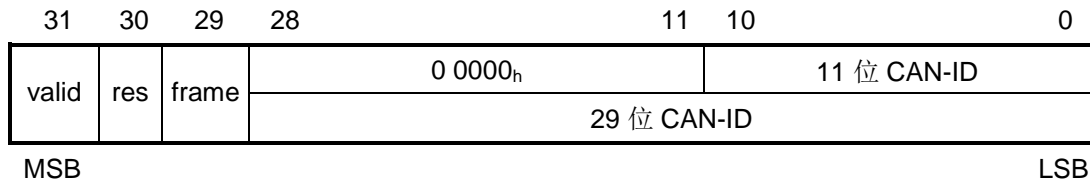


图 64: EMCY COB-ID 结构

表 62: EMCY COB-ID 说明

位 (们)	值	描述
valid	0 <sub>b</sub>	EMCY 消费者存在 / 有效
	1 <sub>b</sub>	EMCY 消费者不存在/无效
res	0 <sub>b</sub>	保留 ( 始终为 0 <sub>b</sub> )
frame	0 <sub>b</sub>	11 位 CAN-ID 有效 (标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (扩展帧)
29 位 CAN-ID	X	29 位 CAN-ID 的扩展帧
11 位 CAN-ID	X	11 位 CAN-ID 的标准帧

仅支持标准帧的 CANopen 设备企图置位 29(frame)为 1<sub>b</sub>将得到 SDO 中止传输响应(中止代码:0609 0030<sub>h</sub>)。当紧急对象有效(位 31 = 0<sub>b</sub>)时, 位 0 至 29 禁止更改。

子索引号关联 node-ID 。

### 对象描述

索引	1028 <sub>h</sub>
名称	紧急消费者
对象类型码	ARRAY
数据类型	UNSIGNED32
类别	可选

### 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> ~7F <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	紧急消费者 1
条目类别	强制
访问权限	rw; const, 如果紧急消费值不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	02 <sub>h</sub> ~7F <sub>h</sub>
描述	紧急消费者 2~127
条目类别	可选
访问权限	rw; const, 如果紧急消费值不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

### 7.5.2.32 对象 1029<sub>n</sub>: 错误行为对象

如果在操作态检测到严重的 CANopen 设备故障，CANopen 设备默认应自动进入 NMT 预操作态。如果支持该对象，CANopen 设备将配置为可选择进入 NMT 停止态或继续留在当前 NMT 状态。CANopen 设备故障包括以下通信错误：

- CAN 接口总线关闭
- 生存监护事件状态为“occurred”且其原因为“time out”。
- 心跳事件状态为“occurred”且其原因为“time out”。

某些 CANopen 设备错误也可能导致 CANopen 设备的内部故障。

#### 值定义

表 63: 错误类别值

值	描述
00 <sub>h</sub>	更改为 NMT 预操作(仅当当前设备状态为 NMT 操作态)
01 <sub>h</sub>	NMT 状态不更改
02 <sub>h</sub>	更改为 NMT 停止态
03 <sub>h</sub>	保留
:::::	:::::
7 F <sub>h</sub>	保留
80 <sub>h</sub>	制造商指定
:::::	:::::
FF <sub>h</sub>	制造商指定

#### 对象描述

索引	1029 <sub>n</sub>
名称	错误行为
对象类型码	ARRAY
数据类型	UNSIGNED8
类别	可选



## 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> ~FE <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	通信错误
条目类别	强制
访问权限	rw; const, 如果通信错误的错误行为不可更改
PDO 映射	否
取值范围	UNSIGNED8
默认值	00 <sub>h</sub>

子索引	02 <sub>h</sub> ~FE <sub>h</sub>
描述	协议或制造商指定的错误
条目类别	可选
访问权限	rw; const, 如果错误的行为不可更改
PDO 映射	否
取值范围	请参阅值定义。
默认值	协议或制造商指定

### 7.5.2.33 对象 1200<sub>h</sub>~127F<sub>h</sub>:SDO 服务器参数

为了描述 CANopen 设备使用的 SDO，前文介绍了 SDO 参数的数据类型。该数据类型在对象字典索引 22<sub>h</sub>处。该结构定义见 7.4.8。

#### 值定义

子索引 00<sub>h</sub>是 SDO 支持的参数数量。子索引 01<sub>h</sub>和 02<sub>h</sub>指定 SDO 的 COB-ID。子索引 03<sub>h</sub>是该 CANopen 设备的客户端 node-ID。

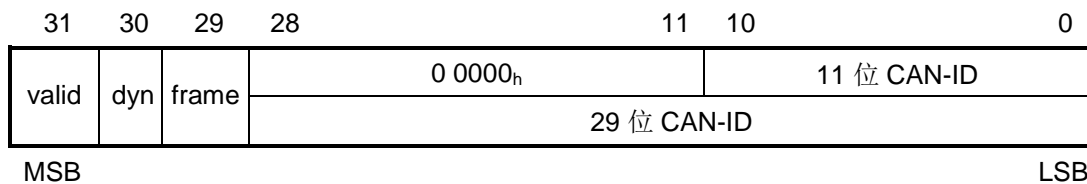


图 65: SDO 服务器 COB-ID 结构

表 64: SDO 服务器 COB-ID 说明

位 (们)	值	描述
valid	0 <sub>b</sub>	SDO 存在 / 有效
	1 <sub>b</sub>	SDO 不存在/无效
dyn	0 <sub>b</sub>	值静态分配
	1 <sub>b</sub>	值动态分配
frame	0 <sub>b</sub>	11 位 CAN-ID 有效 (标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (扩展帧)
29 位 CAN-ID	x	29 位 CAN-ID 的扩展帧
11 位 CAN-ID	x	11 位 CAN-ID 的标准帧

SDO 仅在子索引 01<sub>h</sub> 和 02<sub>h</sub> 的最高位置 0<sub>b</sub> 时存在。仅支持标准帧的 CANopen 设备企图置位 29(frame)为 1<sub>b</sub> 将得到 SDO 中止传输响应(中止代码:0609 0030<sub>h</sub>)。当该对象存在并有效(位 31 = 0<sub>b</sub>) 时, 位 0 至 29 禁止更改。

如果子索引 01<sub>h</sub> 或 02<sub>h</sub> 位 dyn(位 30)设为 1<sub>b</sub>, 该对象所有参数不能保存在非易失性存储器中, 该位用于标记支持动态 SDO 连接的 CANopen 设备。动态 SDO 连接是使用前临时配置。静态 SDO 连接配置是长效的, 因而要保存在非易失性存储器中。CANopen 管理器可以通过 dyn 位检测 SDO 连接的动态性。

#### 对象描述

索引	1200 <sub>h</sub> ~127F <sub>h</sub>
名称	SDO 服务器参数
对象类型码	RECORD
数据类型	SDO parameter record
类别	条件 索引 1200 <sub>h</sub> : 可选 索引 1201 <sub>h</sub> ~127F <sub>h</sub> : 只要支持了其参数就是强制性的

#### 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	索引 1200 <sub>h</sub> : 02 <sub>h</sub> 索引 1201 <sub>h</sub> ~127F <sub>h</sub> : 02 <sub>h</sub> ~03 <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	COB-ID 客户端->服务器(rx)
条目类别	强制性
访问权限	索引 1200 <sub>h</sub> :               const 索引 1201 <sub>h</sub> ~127F <sub>h</sub> :       rw; const,如果由应用协议定义
PDO 映射	可选
取值范围	请参阅值定义
默认值	索引 1200 <sub>h</sub> :               CAN-ID: 600 <sub>h</sub> +Node-ID frame: 0 <sub>b</sub> dyn: 0 <sub>b</sub> valid: 0 <sub>b</sub>  索引 1201 <sub>h</sub> ~127F <sub>h</sub> :       CAN-ID:制造商制定(参见 7.3.5) frame: 制造商指定 dyn: 0 <sub>b</sub> valid: 1 <sub>b</sub> 或由应用协议定义

子索引	02 <sub>h</sub>
描述	COB-ID 服务器->客户端(tx)
条目类别	强制性
访问权限	索引 1200 <sub>h</sub> :               ro 索引 1201 <sub>h</sub> ~127F <sub>h</sub> :       rw; const,如果由应用协议定义
PDO 映射	可选
取值范围	请参阅值定义
默认值	索引 1200 <sub>h</sub> :               CAN-ID: 580 <sub>h</sub> +Node-ID frame: 0 <sub>b</sub> dyn: 0 <sub>b</sub> valid: 0 <sub>b</sub>  索引 1201 <sub>h</sub> ~127F <sub>h</sub> :       CAN-ID:制造商制定(参见 7.3.5) frame: 制造商指定 dyn: 0 <sub>b</sub> valid: 1 <sub>b</sub> 或由应用协议定义

子索引	03 <sub>h</sub>
描述	SDO 客户端 Node-ID
条目类别	可选
访问权限	rw
PDO 映射	否
取值范围	01 <sub>h</sub> ~7F <sub>h</sub>
默认值	制造商指定



## 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制性
访问权限	const
PDO 映射	否
取值范围	03 <sub>h</sub>
默认值	03 <sub>h</sub>

子索引	01 <sub>h</sub>
描述	COB-ID 客户端->服务器(tx)
条目类别	强制
访问权限	rw; const,如果由应用协议定义
PDO 映射	可选
取值范围	请参阅值定义
默认值	CAN-ID: 制造商指定 (参见 7.3.5) frame: 制造商指定 dyn: 0 <sub>b</sub> valid: 1 <sub>b</sub> 或由应用协议定义

子索引	02 <sub>h</sub>
描述	COB-ID 服务器->客户端(rx)
条目类别	强制
访问权限	rw; const, 如果由应用程序定义
PDO 映射	可选
取值范围	请参阅值定义。
默认值	CAN-ID: 制造商指定 (参见 7.3.5) frame: 制造商指定 dyn: 0 <sub>b</sub> valid: 1 <sub>b</sub> 或由应用协议定义

子索引	03 <sub>h</sub>
描述	SDO 服务器 Node-ID
条目类别	强制
访问权限	rw; const, 如果值不可更改的
PDO 映射	否
取值范围	01 <sub>h</sub> ~7F <sub>h</sub>
默认值	制造商指定

### 7.5.2.35 对象 1400<sub>h</sub>~15FF<sub>h</sub>:RPDO 通信参数

此对象包含 CANopen 设备接收 PDOs 的通信参数。

#### 值定义

子索引 00<sub>h</sub> 包含有效的对象条目录录。其值至少是 02<sub>h</sub>。如果支持禁止时间其值为 03<sub>h</sub>，如果支持事件定时器其值为 05<sub>h</sub>。

子索引 01<sub>h</sub> 包含 RPDO 的 COB-ID (见图 67 和表 66)。

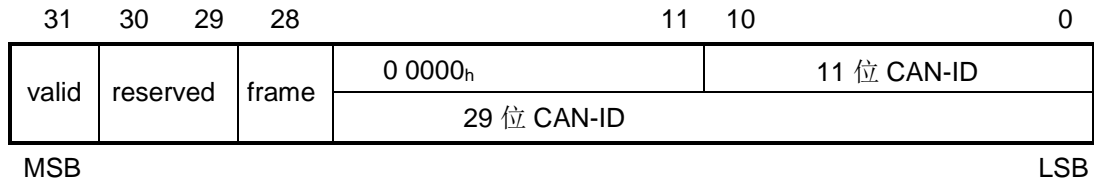


图 67:RPDO 的 COB-ID 结构

表 66: RPDO COB-ID 说明

位(们)	值	描述
valid	0 <sub>b</sub>	PDO 存在/有效
	1 <sub>b</sub>	PDO 不存/无效
reserved	x	任意
frame	0 <sub>b</sub>	11 位 CAN-ID 有效 (标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (扩展帧)
29 位 CAN-ID	x	29 位 CAN-ID 扩展帧
11 位 CAN-ID	x	11 位 CAN-ID 标准帧

valid(位 31)允许设定在 NMT 操作态哪些 RPDO 可用。PDOs 可以配置但不使用，清除该位即可。对于支持 4 个 RPDOs 的 CANopen 设备，该功能是必要的，因为每个 CANopen 设备前四个 RPDO 是带有默认 CAN-ID 的通用预定义连接集。仅支持标准帧的 CANopen 设备企图置位 29(frame)为 1<sub>b</sub>将得到 SDO 中止传输响应(中止代码:0609 0030<sub>h</sub>)。当该对象存在并有效(位 31 = 0<sub>b</sub>)时，位 0 至 29 禁止更改。如果设备支持使能(位 31=0<sub>b</sub>)和禁止(bit 31 = 1<sub>b</sub>)RPDO 操作，对位 0 至 30 更改将返回 SDO 中止传输报文(中止代码: 0609 0030<sub>h</sub> 或 0800 000<sub>h</sub>)。

如果 CANopen 设备使用设备协议，一定要支持通用预定义连接集 (见表 67)。

表 67: RPDO 的通用预定义连接集

索引	默认值
1400 <sub>h</sub>	CAN-ID: 200 <sub>h</sub> + Node-ID frame: 0 <sub>b</sub> reserved: 制造商指定 valid: 协议或制造商指定
1401 <sub>h</sub>	CAN-ID : 300 <sub>h</sub> + Node-ID frame: 0 <sub>b</sub> reserved: 制造商指定 有效: 协议或制造商指定

索引	默认值
1402 <sub>h</sub>	CAN-ID : 400 <sub>h</sub> + Node-ID frame: 0 <sub>b</sub> reserved: 制造商指定 有效: 协议或制造商指定
1403 <sub>h</sub>	CAN-ID : 500 <sub>h</sub> + Node-ID frame: 0 <sub>b</sub> reserved: 制造商指定 有效: 协议或制造商指定
1404 <sub>h</sub> 15 FF <sub>h</sub>	CAN-ID : 协议或制造商指定(见 7.3.5) frame: 协议或制造商指定 reserved: 制造商指定 有效: 1 <sub>b</sub> 或由应用协议定义

如果 CANopen 设备使用应用协议, 则应遵守应用协议的预定义连接集。

子索引 02<sub>h</sub> 定义了 RPDO 的接收性质(见表 68)。试图更改其值为任何不支持的传输类型将应答 SDO 中止传输服务(中止代码:0609 0030<sub>h</sub>)。

表 68: RPDO 传输类型说明

值	描述
00 <sub>h</sub>	同步
:::::	:::::
F 0 <sub>h</sub>	同步
F 1 <sub>h</sub>	保留
:::::	:::::
FD <sub>h</sub>	保留
FE <sub>h</sub>	事件驱动(制造商指定)
FF <sub>h</sub>	事件驱动(设备协议和应用协议指定)

- 同步意味着该 CANopen 设备在收到紧接着的一帧 SYNC(见图 68)后生效接收的数据。
- 事件驱动意味着任何时候都可以接收 PDO。且数据立即生效。

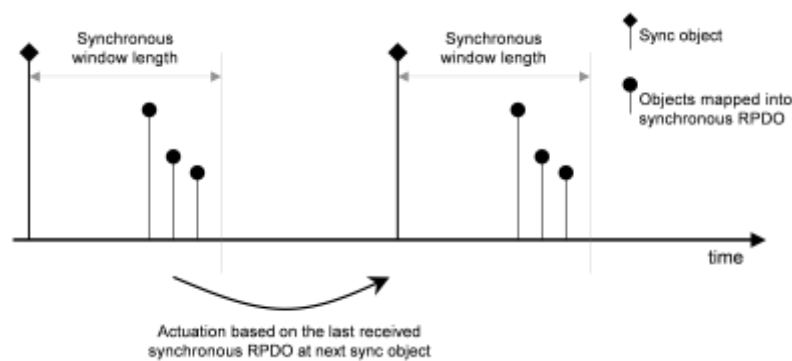


图 68: 总线同步和生效

子索引 03<sub>h</sub> 包含禁止时间。该值单位为 100 $\mu$ s。值为 0 表示禁用。该值在 PDO 生效 (子索引 01<sub>h</sub> 位 31 设为 0<sub>b</sub>) 时不可更改。RPDO 使用该时间**执行特定功能**。

子索引 04<sub>h</sub> 保留。它不应被执行;取或写入将导致 SDO 中止传输 (中止代码: 0609 0011<sub>h</sub>)。

子索引 05<sub>h</sub> 包含事件定时器。单位 ms。值为 0 将禁用事件定时器。RPDO 可使用该时间进行期限监控。配置事件定时器后接收到一帧 RPDO 激活期限监控。超时结果通知本地应用程序。

子索引 06<sub>h</sub> 包含同步起始值。RPDOs 不使用该值;因此对其读写将应答 SDO 中止传输(中止代码:0609 0011<sub>h</sub>)。

#### 对象描述

索引	1400 <sub>h</sub> ~15FF <sub>h</sub>
名称	RPDO 通信参数
对象类型码	RECORD
数据类型	PDO 通信参数记录
类别	有条件的; 强制, 如果支持对应 RPDO

#### 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	02 <sub>h</sub> 至 06 <sub>h</sub>
默认值	无

子索引	01 <sub>h</sub>
描述	RPDO 使用的 COB-ID
条目类别	强制
访问权限	rw; const 如果 COB-ID 不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	请参阅值定义

子索引	02 <sub>h</sub>
描述	传输类型
条目类别	强制
访问权限	rw; const, 如果传输类型是不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定



子索引	03 <sub>h</sub>
描述	禁止时间
条目类别	可选
访问权限	rw; const, 如果禁止时间不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	04 <sub>h</sub>
描述	兼容性条目
条目类别	可选
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED8
默认值	制造商指定

子索引	05 <sub>h</sub>
描述	事件定时器
条目类别	可选
访问权限	rw; const, 如果事件的定时器不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	06 <sub>h</sub>
描述	同步起始值
条目类别	可选
访问权限	rw const, 如果同步起始值不可更改的
PDO 映射	否
值范围。	UNSIGNED8
默认值	协议或制造商指定

### 7.5.2.36 对象 1600<sub>h</sub>~17FF<sub>h</sub>:RPDO 映射参数

此对象包含 CANopen 设备发送 PDOs 的映射参数。

#### 值定义

子索引 00<sub>h</sub> 包含有效对象条目映射记录或指定值 (见表 69),例如,如果支持 MPDO。有效的对象条目数须为该应用程序对象相应 RPDO 应收到的数目。

表 69: RPDO 映射值

值	描述
00 <sub>h</sub>	映射已禁用
01 <sub>h</sub>	子索引 01 <sub>h</sub> 有效
02 <sub>h</sub>	子索引 01 <sub>h</sub> 和 02 <sub>h</sub> 有效
03 <sub>h</sub>	子索引从 01 <sub>h</sub> 到 03 <sub>h</sub> 有效
04 <sub>h</sub>	子索引从 01 <sub>h</sub> 到 04 <sub>h</sub> 有效
:::::	:::::
40 <sub>h</sub>	子索引从 01 <sub>h</sub> 到 40 <sub>h</sub> 有效
41 <sub>h</sub>	保留的
:::::	:::::
FD <sub>h</sub>	保留的
FE <sub>h</sub>	SAM-MPDO
FF <sub>h</sub>	DAM-MPDO

子索引从 01<sub>h</sub> 到 40<sub>h</sub> 包含应用对象映射信息。对象描述了 PDO 数据内容对应的索引、子索引和长度 (见图 69 和图 70)。长度为数据位长度, 用于验证映射。

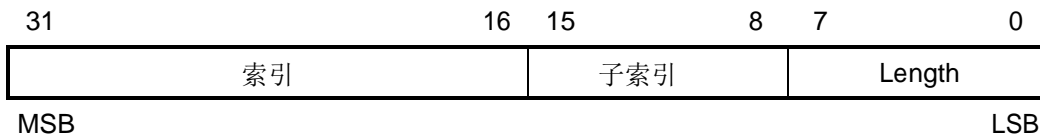


图 69: RPDO 映射结构

试图更改其值在对象字典中没有对应对象, 将返回 SDO 中止传输。其不对应情况包括映射(索引和子索引)不存在, 数据长度不一致, PDO 长度异常等。索引和子索引需引用简单的数据类型(参见表 44)所谓的伪映射。如果没有相应的对象可用, 则根据 TPDO 填充 RPDO, 以匹配其长度。

下面的动作将导致重映射, 可能发生于 NMT 预操作和操作态, 如果支持:

1. 通过设置 RPDO 通信参数子索引 01<sub>h</sub> 位 *valid* 为 1<sub>b</sub> 销毁 RPDO。
2. 通过设置子索引 00<sub>h</sub> 至 00<sub>h</sub> 禁用映射。
3. 通过相应子索引修改映射值。
4. 通过设置子索引 00<sub>h</sub> 为映射对象数来使能映射。
5. 通过设置 RPDO 通信参数子索引 01<sub>h</sub> 位 *valid* 为 0<sub>b</sub> 创建 RPDO。

如果在第 3 步的 CANopen 设备检测到索引和子索引不存在或不允许被映射将响应 SDO 中止传输(中止代码:0602 0000<sub>h</sub> 或 0604 0041<sub>h</sub>)。

如果在第 4 步的 CANopen 设备检测到 RPDO 映射不可用或无效, CANopen 设备响应 SDO 中止传输(中止代码:0602 0000<sub>h</sub> 或 0604 0042<sub>h</sub>)。

如果 CANopen 设备接收 PDO 数据字节数超出映射数据字节数(长度), CANopen 设备应使用前者的数据字节长度, 并且启动 EMCY 写服务, 如果支持的话。

如果 CANopen 设备接收 PDO 数据字节少于映射数据字节数(长度), 然后 CANopen 设备应启动 EMCY 写服务, 如果支持的话, 错误代码 8210<sub>h</sub>。

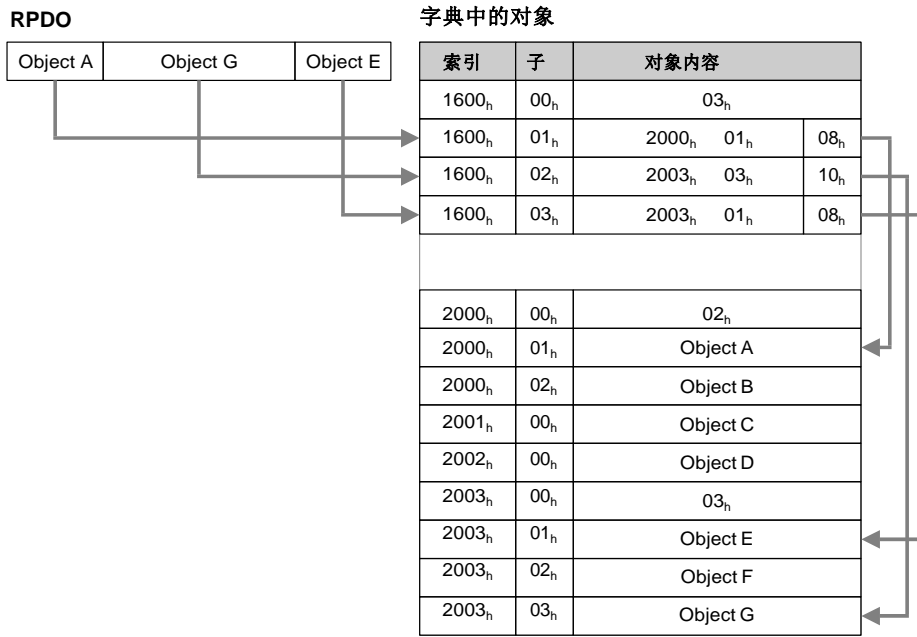


图 70: RPDO 映射规则

对象描述

索引	1600 <sub>h</sub> ~17FF <sub>h</sub>
名称	RPDO 映射参数
对象的代码。	RECORD
数据类型	PDO 映射参数记录
类别	条件; 强制, 对于每个支持的 PDO

入口说明

子索引	00 <sub>h</sub>
描述	PDO 映射数
条目类别	强制
访问权限	rw; const, 如果映射不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	应用对象 1 <sup>st</sup>
条目类别	强制
访问权限	rw ; const 如果映射条目不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	02 <sub>h</sub> ~40 <sub>h</sub>
描述	第 2 <sup>nd</sup> 到 64 <sup>th</sup> 应用程序对象
条目类别	可选
访问权限	rw; const 如果映射条目不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

### 7.5.2.37 对象 1800<sub>h</sub>~19FF<sub>h</sub>: TPDO 通信参数

此对象包含 CANopen 设备支持的 PDOs 通信参数。

#### 值定义

子索引 00<sub>h</sub> 包含有效的对象条目记录。其值至少是 02<sub>h</sub>。如果支持禁止时间其值为 03<sub>h</sub>，如果支持事件定时器其值为 05<sub>h</sub>。

子索引 01<sub>h</sub> 包含 TPDO 的 COB-ID(见图 71 和表 70)。

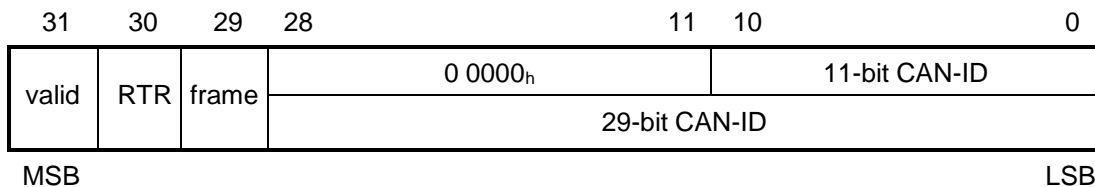


图 71: TPDO 的 COB-ID 结构

表 70: TPDO 的 COB-ID 描述

位 (S)	值	描述
valid	0 <sub>b</sub>	PDO 存在/有效
	1 <sub>b</sub>	PDO 不存/无效
RTR	0 <sub>b</sub>	此 PDO 允许 RTR
	1 <sub>b</sub>	此 PDO 不允许 RTR
frame	0 <sub>b</sub>	11 位 CAN-ID 有效 (标准帧)
	1 <sub>b</sub>	29 位 CAN-ID 有效 (扩展帧)
29 位 CAN-ID	x	29 位 CAN-ID 扩展帧
11 位 CAN-ID	x	11 位 CAN-ID 标准帧

valid(位 31)允许设定在 NMT 操作态哪些 TPDO 可用。PDOs 可以配置但不使用，清除该位即可。对于支持 4 个 TPDOs 的 CANopen 设备，该功能是必要的，因为每个 CANopen 设备前四个 RPDO 是带有默认 CAN-ID 的通用预定义连接集。仅支持标准帧的 CANopen 设备企图置位 29(frame)为 1<sub>b</sub>将得到 SDO 中止传输响应(中止代码:0609 0030<sub>h</sub>)。当该对象存在并有效(位 31 = 0<sub>b</sub>)时，位 0 至 29 禁止更改。如果设备支持使能(位 31=0<sub>b</sub>)和禁止(bit 31 = 1<sub>b</sub>)TPDO 操作，对位 0 至 30 更改将返回 SDO 中止传输报文(中止代码: 0609 0030<sub>h</sub> 或 0800 000<sub>h</sub>)。

如果 CANopen 设备使用设备协议，一定要支持通用预定义连接集 (见表 67)。

表 71: TPDO 通用预定义连接集

索引	默认值
1800 <sub>h</sub>	CAN-ID: 180 <sub>n</sub> + Node-ID frame: 0 <sub>b</sub> RTR: 协议或制造商指定 valid: 协议或制造商指定
1801 <sub>h</sub>	CAN-ID: 280 <sub>n</sub> + Node-ID frame: 0 <sub>b</sub> RTR: 协议或制造商指定 valid: 协议或制造商指定
1802 <sub>h</sub>	CAN-ID: 380 <sub>n</sub> + Node-ID frame: 0 <sub>b</sub> RTR: 协议或制造商指定 valid: 协议或制造商指定
1803 <sub>h</sub>	CAN-ID: 480 <sub>n</sub> + Node-ID frame: 0 <sub>b</sub> RTR: 协议或制造商指定 valid: 协议或制造商指定
1804 年的 <sub>h</sub> 到 19 FF <sub>h</sub>	CAN-ID: 协议或制造商指定(见 7.3.5) frame: 协议或制造商指定 RTR: 协议或制造商指定 valid: 1 <sub>b</sub> 或应用协议定义

如果 CANopen 设备使用应用协议，则应遵守应用协议的预定义连接集。

子索引 02<sub>h</sub> 定义了 TPDO 的发送性质(见表 72)。试图更改其值为任何不支持的传输类型将应答 SDO 中止传输服务(中止代码:0609 0030<sub>h</sub>)。

表 72: 描述的 TPDO 变速箱类型

值	描述
00 <sub>h</sub>	同步 (非周期)
01 <sub>h</sub>	同步 (每个 SYNC 一周期)
02 <sub>h</sub>	同步 (每 2 <sup>nd</sup> SYNC 一周期)
03 <sub>h</sub>	同步 (每 3 <sup>rd</sup> SYNC 一周期)
04 <sub>h</sub>	同步 (每 4 <sup>th</sup> SYNC 一周期)
.....	.....
F0 <sub>h</sub>	同步 (每 240 <sup>th</sup> SYNC 一周期)
F1 <sub>h</sub>	保留
.....	.....
FB <sub>h</sub>	保留
FC <sub>h</sub>	仅 RTR(同步)
FD <sub>h</sub>	仅 RTR(事件驱动)
FE <sub>h</sub>	事件驱动(制造商指定)
FF <sub>h</sub>	事件驱动(设备协议和应用协议指定)

- 同步意味着该 PDO 在 SYNC 后发送。CANopen 设备在收到 SYNC 后开始数据采样(参见图 72)。如果是非周期方式, CANopen 设备在内部事件给定并且紧接着的一次 SYNC 后开始采样, 随后发送 PDO。如果是周期方式, 每个 SYNC、2<sup>nd</sup> SYNC、3<sup>rd</sup> SYNC 等(由给定参数值决定)启动一次采样, 发送一次 PDO。
- 仅 RTR 指 PDO 由远程 RTR 触发发送。如果是同步方式, CANopen 设备每个 SYNC 接收后开始采样, 然后缓冲进 PDO(参见图 72)。如果是事件驱动方式, CANopen 设备在收到 RTR 请求后开始采样并直接生成 PDO 发送。
- 事件驱动的方式指 PDO 的发送由 CANopen 设备任意时刻发生的内部事件触发。事件的定义不在本文范围, 可由设备协议或应用协议规定。

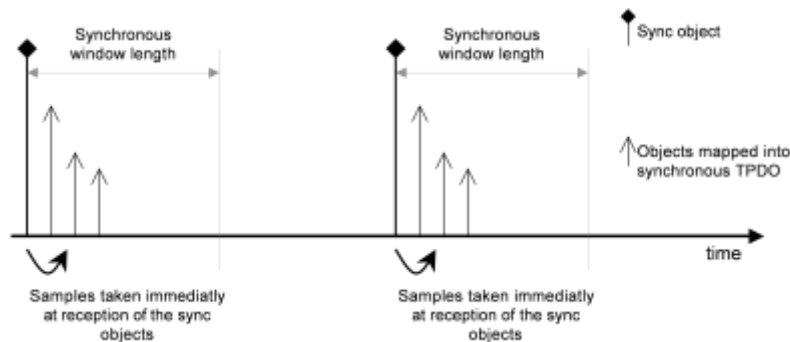


图 72: 总线同步和采样

子索引 03<sub>h</sub> 为禁止时间。如果传输类型设置为 FE<sub>h</sub> 和 FF<sub>h</sub>, 它就是最小的 PDO 传输间隔。该值单位 100 $\mu$ s。值为 0 将禁用禁止时间。PDO 存在(子索引 01<sub>h</sub> 位 31 设为 0<sub>b</sub>)该值不应被更改。

子索引 04<sub>h</sub> 保留。它不应被执行;在这种情况下读取或写入将导致 SDO 中止传输(中止代码:0609 0011<sub>h</sub>)。

子索引 05<sub>h</sub> 包含事件定时器。如果传输类型设置为 FE<sub>h</sub> 和 FF<sub>h</sub>，该值为最大 PDO 传输时间间隔。该值单位 ms。值为 0 将禁用事件定时器。

子索引 06<sub>h</sub> 为 SYNC 起始值。SYNC 起始值为 0 表示该 PDO 不处理 SYNC 帧的计数。SYNC 起始值为 1 至 240 表示该 PDO 处理 SYNC 帧的计数。如果 SYNC 消息计数器未使能(见 7.5.2.22)，子索引 06<sub>h</sub> 应被忽略。SYNC 帧的计数器值等于 SYNC 起始值时，视为收到第一帧 SYNC。当 PDO 存在(子索引 01<sub>h</sub> 位 31 设为 0<sub>b</sub>)该值不应更改。

**注意** 如果 CANopen 设备进入 NMT 操作态，而检测到 SYNC 计数器值大于 SYNC 起始值，CANopen 设备必须等待下一轮正确的同步计数器值。

#### 对象描述

索引	1800 <sub>h</sub> ~19FF <sub>h</sub>
名称	TPDO 的通信参数
对象类型码	RECORD
数据类型	PDO 通信参数记录
类别	有条件的; 强制, 如果对应 TPDO 被支持

#### 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	02 <sub>h</sub> 至 05 <sub>h</sub>

子索引	01 <sub>h</sub>
描述	TPDO 使用的 COB-ID
条目类别	强制
访问权限	rw; const, 如果 COB-ID 不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	请参阅值定义

子索引	02 <sub>h</sub>
描述	传输类型
条目类别	强制
访问权限	rw; const, 如果传输类型是不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	03 <sub>h</sub>
描述	禁止时间
条目类别	可选
访问权限	rw; const, 如果禁止时间不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	04 <sub>h</sub>
描述	保留
条目类别	可选
访问权限	rw
PDO 映射	否
取值范围	UNSIGNED8
默认值	制造商指定

子索引	05 <sub>h</sub>
描述	事件定时器
条目类别	可选
访问权限	rw; const, 如果事件定时器不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定



子索引	06 <sub>h</sub>
描述	同步起始值
条目类别	可选
访问权限	rw; const, 如果同步起始值不可更改
PDO 映射	否
取值范围	UNSIGNED8
默认值	协议或制造商指定

### 7.5.2.38 对象 1A00<sub>h</sub>~1BFF<sub>h</sub>: TPDO 映射参数

此对象包含 CANopen 设备接收 PDOs 的映射参数。

#### 值定义

子索引 00<sub>h</sub> 包含有效对象条目映射记录或指定值 (见表 73), 例如, 如果支持 MPDO。有效的对象条目数须为该应用程序对象相应 TPDO 应发送的数目。

表 73: TPDO 映射值

值	描述
00 <sub>h</sub>	映射已禁用
01 <sub>h</sub>	子索引 01 <sub>h</sub> 有效
02 <sub>h</sub>	子索引 01 <sub>h</sub> 和 02 <sub>h</sub> 有效
03 <sub>h</sub>	子索引从 01 <sub>h</sub> 到 03 <sub>h</sub> 有效
04 <sub>h</sub>	子索引从 01 <sub>h</sub> 到 04 <sub>h</sub> 有效
:::::	:::::
40 <sub>h</sub>	子索引从 01 <sub>h</sub> 到 40 <sub>h</sub> 有效
41 <sub>h</sub>	保留的
:::::	:::::
FD <sub>h</sub>	保留的
FE <sub>h</sub>	SAM-MPDO
FF <sub>h</sub>	DAM-MPDO

子索引从 01<sub>h</sub> 到 40<sub>h</sub> 包含应用对象映射信息。对象描述了 PDO 数据内容对应的索引、子索引和长度 (见图 73 和图 74)。长度为数据位长度, 用于验证映射。

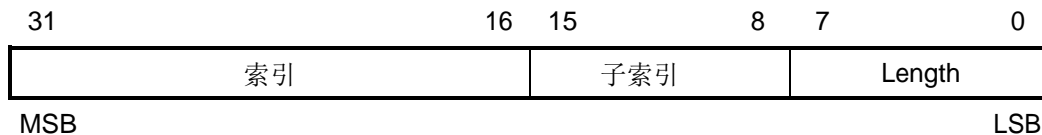


图 73: TPDO 映射结构

试图更改其值在对象字典中没有对应对象, 将返回 SDO 中止传输。其不对应情况包括映射(索引和子索引)不存在, 数据长度不一致, PDO 长度异常等。索引和子索引需引用简单的数据类型(参见表 44)所谓的伪映射。如果没有相应的对象可用, 则根据 TPDO 填充 RPDO, 以匹配其长度。

下面的动作将导致重映射, 可能发生于 NMT 预操作和操作态, 如果支持:

1. 通过设置 RPDO 通信参数子索引 01<sub>h</sub> 位 *valid* 为 1<sub>b</sub> 销毁 TPDO。
2. 通过设置子索引 00<sub>h</sub> 至 00<sub>h</sub> 禁用映射。
3. 通过相应子索引修改映射值。
4. 通过设置子索引 00<sub>h</sub> 为映射对象数来使能映射。
5. 通过设置 RPDO 通信参数子索引 01<sub>h</sub> 位 *valid* 为 0<sub>b</sub> 创建 TPDO。

如果在第 3 步的 CANopen 设备检测到索引和子索引不存在或不允许被映射将响应 SDO 中止传输(中止代码:0602 0000<sub>h</sub> 或 0604 0041<sub>h</sub>)。

如果在第 4 步的 CANopen 设备检测到 TPDO 映射不可用或无效, CANopen 设备响应 SDO 中止传输(中止代码:0602 0000<sub>h</sub> 或 0604 0042<sub>h</sub>)。

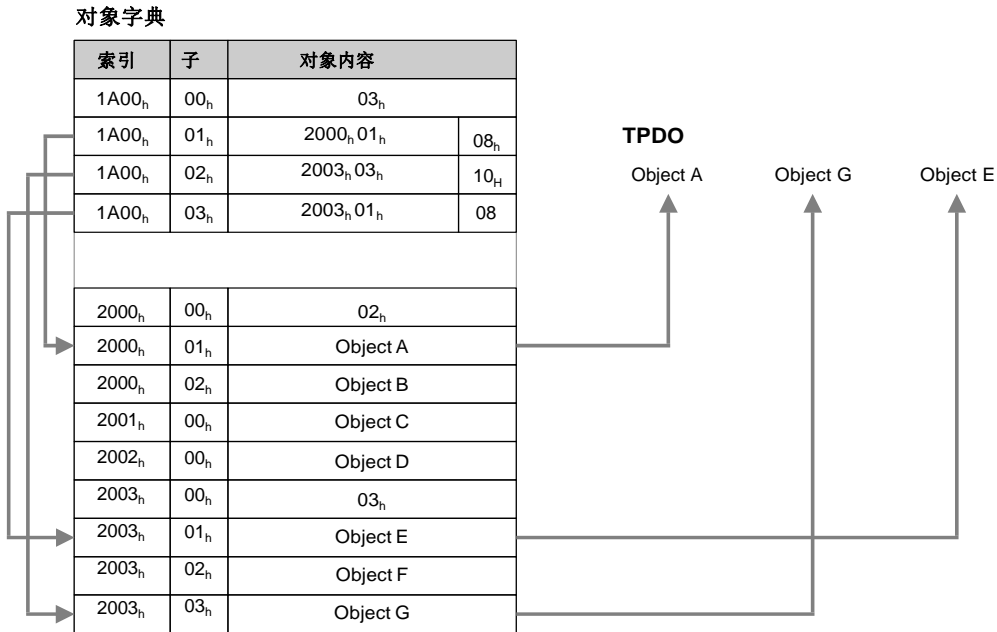


图 74: TPDO 映射规则

**对象描述**

索引	1A00 <sub>h</sub> ~1BFF <sub>h</sub>
名称	TPDO 映射
对象类型	RECORD
数据类型	PDO 映射参数记录
类别	条件的; 强制, 如果支持对应 PDO

## 入口说明

分指数	00 <sub>h</sub>
描述	TPDO 映射对象数
条目类别	强制
访问权限	rw; const, 如果 PDO 映射是不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	应用对象 1 <sup>st</sup>
条目类别	强制
访问权限	rw; const 如果映射条目不可更改
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

子索引	02 <sub>h</sub> ~40 <sub>h</sub>
描述	第 2 <sup>nd</sup> 到 64 <sup>th</sup> 应用程序对象
条目类别	可选
访问权限	rw; const 如果映射条目不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

### 7.5.2.39 对象 1FA0<sub>h</sub>~1FCF<sub>h</sub>: 对象扫描仪列表

SAM-MPDO 生产者使用该对象扫描列表配置,确定发送哪些对象。

值定义

每个对象条目描述了一个 MPDO 通信对象。通过块大小参数可以描述连续的子索引。

对象条目数为 0 表示为配置。

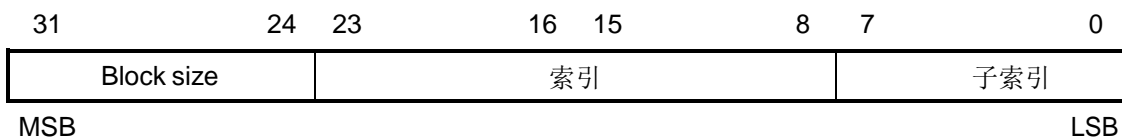


图 75: 对象扫描列表对象入口

## 对象描述

索引	1 FA 0 <sub>H</sub> ~1FCF <sub>H</sub>
名称	对像扫描列表
对象类型码	ARRAY
数据类型	UNSIGNED32
类别	可选

## 入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> to FE <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	扫描对像 1
条目类别	强制
访问权限	rw; ro 或 const , 如果扫描的条目是不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

分指数	02 <sub>h</sub> to FE <sub>h</sub>
描述	扫描 对像 2 到 254
条目类别	可选
访问权限	rw; ro 或 const , 如果扫描的条目是不可更改的
PDO 映射	否
取值范围	请参阅值定义
默认值	协议或制造商指定

### 7.5.2.40 对象 1FD0<sub>H</sub>~1FFF<sub>H</sub>: 对象分配列表

SAM-MPDO 消费者使用该对象作为远程生产者和本地对像字典之间的交叉引用。

值定义

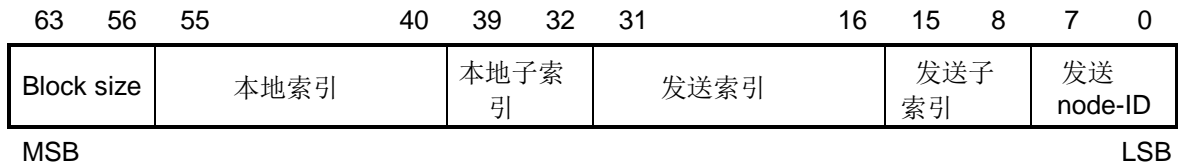


图 76: 对象分配表条目

每个对象条目描述了 MPDO 接收的数据如何转移到本地对象字典。如果标志域值为 0，生产者的 node-ID，生产者索引和子索引条目匹配,数据就会写入条目索引和子索引指向的本地字典。

参数块的大小描述了连续的子索引数。例如：如果发送方子索引 1-9 对应于接收方 11-19, 其定义范围为

发送方子索引 = 1

本地子索引 = 11

块大小 = 9

对象条目值为 0 表示未配置。

对象描述

索引	1 FD 0 <sub>H</sub> ~1FFF <sub>H</sub>
名称	对象分配列表
对象类型码	ARRAY
数据类型	UNSIGNED64
类别	可选

入口说明

子索引	00 <sub>h</sub>
描述	子索引数
条目类别	强制
访问权限	const
PDO 映射	否
取值范围	01 <sub>h</sub> to FE <sub>h</sub>
默认值	协议或制造商指定

子索引	01 <sub>h</sub>
描述	分配单元 1
条目类别	强制
访问权限	rw; ro 或 const , 如果分配项不可更改
PDO 映射	否
取值范围	UNSIGNED64
默认值	协议或制造商指定

分指数	02 <sub>h</sub> to FE <sub>h</sub>
描述	分配单元 2 到 254
条目类别	可选
访问权限	rw; ro 或 const , 如果分配项不可更改
PDO 映射	否
取值范围	UNSIGNED64
默认值	协议或制造商指定

## 附件 A (更多信息)

## 应用建议

应用该协议，需要遵循以下规则以保证互操作性。这些规则包括下列方面：

## 无效 COB's

如果一个 COB 的 COB-ID 遵守协议规范，但根据协议规范它包含无效的参数值，那么它就是无效的。这可能由于低层的错误或应用错误造成，无效 COB's 需由上层应用处理而不在本文范围，而无效 COB 应被协议栈忽略。

## Time-out's

由于 COB's 可能被忽略，应答确认可能永远也不能到达。要解决这种问题，就需要在一定的时间后通知用户(time-out)。超时并不算对服务的确认。超时表示该服务尚未完成。由应用程序应该处理这种情况。超时时间由具体应用指定，因而不属于本规范的范围。然而，推荐具体应用能够提供修改超时时间的工具以满足要求。

## PDO 传输类型 0、254、255

如果在相应协议中未特别规定，传输类型 254 和 255 的 PDOs 在设备进入操作态后直接发送，而类型 0 还要收到一条 Sync。

## 对象字典通信对象总览

表 74: 标准对象

索引	对象	名称	数据类型	Acc. <sup>2</sup>	M/O
1000 <sub>h</sub>	VAR	device type	UNSIGNED32	ro	M
1001 <sub>h</sub>	VAR	error register	UNSIGNED8	ro	M
1002 <sub>h</sub>	VAR	manufacturer status register	UNSIGNED32	ro	O
1003 <sub>h</sub>	ARRAY	pre-defined error field	UNSIGNED32	ro	O
1004 <sub>h</sub>	reserved for compatibility reasons				
1005 <sub>h</sub>	VAR	COB-ID SYNC	UNSIGNED32	rw	O
1006 <sub>h</sub>	VAR	communication cycle period	UNSIGNED32	rw	O
1007 <sub>h</sub>	VAR	synchronous window length	UNSIGNED32	rw	O
1008 <sub>h</sub>	VAR	manufacturer device name	VIS-STRING	const	O
1009 <sub>h</sub>	VAR	manufacturer hardware version	VIS-STRING	const	O
100A <sub>h</sub>	VAR	manufacturer software version	VIS-STRING	const	O
100B <sub>h</sub>	reserved for compatibility reasons				
100C <sub>h</sub>	VAR	guard time	UNSIGNED16	rw	O
100D <sub>h</sub>	VAR	life time factor	UNSIGNED8	rw	O
100E <sub>h</sub>	reserved for compatibility reasons				
100F <sub>h</sub>	reserved for compatibility reasons				
1010 <sub>h</sub>	ARRAY	store parameters	UNSIGNED32	rw	O
1011 <sub>h</sub>	ARRAY	restore default parameters	UNSIGNED32	rw	O

<sup>2</sup> Access type listed here may vary for certain sub-indices. See detailed object specification.

Index	Object	Name	Data type	Acc. <sup>2</sup>	M/O
1012 <sub>h</sub>	VAR	COB-ID TIME	UNSIGNED32	rw	O
1013 <sub>h</sub>	VAR	high resolution time stamp	UNSIGNED32	rw	O
1014 <sub>h</sub>	VAR	COB-ID EMCY	UNSIGNED32	rw	O
1015 <sub>h</sub>	VAR	Inhibit Time EMCY	UNSIGNED16	rw	O
1016 <sub>h</sub>	ARRAY	Consumer heartbeat time	UNSIGNED32	rw	O
1017 <sub>h</sub>	VAR	Producer heartbeat time	UNSIGNED16	rw	O
1018 <sub>h</sub>	RECORD	Identity Object	IDENTITY (23 <sub>h</sub> )	ro	M
1019 <sub>h</sub>	VAR	Synchronous counter overflow value	UNSIGNED8	rw	O
<b>Device configuration</b>					
1020 <sub>h</sub>	ARRAY	Verify configuration	UNSIGNED32	rw	O
<b>EDS storage</b>					
1021 <sub>h</sub>	VAR	Store EDS	DOMAIN	rw	O
1022 <sub>h</sub>	VAR	Storage format	UNSIGNED8	rw	O
<b>OS command and prompt</b>					
1023 <sub>h</sub>	RECORD	OS command	COMMANDPAR (25 <sub>h</sub> )	rw	O
1024 <sub>h</sub>	VAR	OS command mode	UNSIGNED8	wo	O
1025 <sub>h</sub>	RECORD	OS debugger interface	OS DEBUG (24 <sub>h</sub> )	rw	O
1026 <sub>h</sub>	ARRAY	OS prompt	UNSIGNED8	rw	O
<b>Modular devices</b>					
1027 <sub>h</sub>	ARRAY	Module list	UNSIGNED16	ro	M/O***
<b>Emergency and error behavior</b>					
1028 <sub>h</sub>	ARRAY	Emergency consumer	UNSIGNED32	rw	O
1029 <sub>h</sub>	ARRAY	Error behavior	UNSIGNED8	rw	O
102A <sub>h</sub>		reserved			
.....	.....	.....	.....	.....	.....
11FF <sub>h</sub>		reserved			
<b>Server SDO parameter</b>					
1200 <sub>h</sub>	RECORD	1 <sup>st</sup> SDO server parameter	SDO PARAM (22 <sub>h</sub> )	ro	O
1201 <sub>h</sub>	RECORD	2 <sup>nd</sup> SDO server parameter	SDO PARAM (22 <sub>h</sub> )	rw	M/O**
.....	.....	.....	.....	.....	.....
127F <sub>h</sub>	RECORD	128 <sup>th</sup> SDO server parameter	SDO PARAM (22 <sub>h</sub> )	rw	M/O**
<b>Client SDO parameter</b>					
1280 <sub>h</sub>	RECORD	1 <sup>st</sup> SDO client parameter	SDO PARAM (22 <sub>h</sub> )	rw	M/O**
1281 <sub>h</sub>	RECORD	2 <sup>nd</sup> SDO client parameter	SDO PARAM (22 <sub>h</sub> )	rw	M/O**
.....	.....	.....	.....	.....	.....
12FF <sub>h</sub>	RECORD	128 <sup>th</sup> SDO client parameter	SDO PARAM (22 <sub>h</sub> )	rw	M/O**
1300 <sub>h</sub>		reserved			
.....	.....	.....	.....	.....	.....
13FF <sub>h</sub>		reserved			
<b>RPDO communication parameter</b>					
1400 <sub>h</sub>	RECORD	1 <sup>st</sup> RPDO communication parameter	PDO COMMPAR (20 <sub>h</sub> )	rw	M/O*



Index	Object	Name	Data type	Acc. <sup>2</sup>	M/O
1401 <sub>h</sub>	RECORD	2 <sup>nd</sup> RPDO communication parameter	PDO COMMPAR (20 <sub>h</sub> )		M/O*
.....	.....	.....	.....	.....	.....
15FF <sub>h</sub>	RECORD	512 <sup>th</sup> RPDO communication parameter	PDO COMMPAR (20 <sub>h</sub> )	rw	M/O*
<b>RPDO mapping parameter</b>					
1600 <sub>h</sub>	RECORD	1 <sup>st</sup> RPDO mapping parameter	PDO MAPPING (21 <sub>h</sub> )	rw	M/O*
1601 <sub>h</sub>	RECORD	2 <sup>nd</sup> RPDO mapping parameter	PDO MAPPING (21 <sub>h</sub> )	rw	M/O*
.....	.....	.....	.....	.....	.....
17FF <sub>h</sub>	RECORD	512 <sup>th</sup> RPDO mapping parameter	PDO MAPPING (21 <sub>h</sub> )	rw	M/O*
<b>TPDO communication parameter</b>					
1800 <sub>h</sub>	RECORD	1 <sup>st</sup> TPDO communication parameter	PDO COMMPAR (20 <sub>h</sub> )	rw	M/O*
1801 <sub>h</sub>	RECORD	2 <sup>nd</sup> TPDO communication parameter	PDO COMMPAR (20 <sub>h</sub> )	rw	M/O*
.....	.....	.....	.....	.....	.....
19FF <sub>h</sub>	RECORD	512 <sup>th</sup> TPDO communication parameter	PDO COMMPAR (20 <sub>h</sub> )	rw	M/O*
<b>TPDO mapping parameter</b>					
1A00 <sub>h</sub>	RECORD	1 <sup>st</sup> TPDO mapping parameter	PDO MAPPING (21 <sub>h</sub> )	rw	M/O*
1A01 <sub>h</sub>	RECORD	2 <sup>nd</sup> TPDO mapping parameter	PDO MAPPING (21 <sub>h</sub> )	rw	M/O*
.....	.....	.....	.....	.....	.....
1BFF <sub>h</sub>	RECORD	512 <sup>th</sup> TPDO mapping parameter	PDO MAPPING (21 <sub>h</sub> )	rw	M/O*
<b>Multiplex PDO</b>					
1FA0 <sub>h</sub>	ARRAY	Object scanner list	UNSIGNED32	rw	O
.....	.....	.....	.....	.....	.....
1FCF <sub>h</sub>	ARRAY	Object scanner list	UNSIGNED32	rw	O
1FD0 <sub>h</sub>	ARRAY	Object dispatching list	UNSIGNED64	rw	O
.....	.....	.....	.....	.....	.....
1FFF <sub>h</sub>	ARRAY	Object dispatching list	UNSIGNED64	rw	O

\* 如果某个 CANopen 设备支持 PDOs，则对应的 PDO 通信参数和映射参数项必须支持。当然可以是 ro 的。

\*\* 如果某个 CANopen 设备支持 SDOs，则相应的 SDO 参数项必须支持。

\*\*\* 见对象定义。

### 注译版本记录

版本	译者	日期	备注
V0.01	winshton	2015/10/28	创建
V0.1	winshton	2015/12/9	初稿