

# GCAN-402

CANopen 数据存储模块

用户手册



文档版本：V1.02 （2015/10/10）

### 修订历史

版本	日期	原因
V1.00	2015/05/22	创建文档
V1.01	2015/08/20	修正设备工作参数
V1.02	2015/10/10	添加部分参数

# 目 录

目 录.....	3
1. 功能简介.....	4
1.1 功能概述.....	4
1.2 性能特点.....	4
1.3 典型应用.....	4
2. 设备使用.....	5
2.1 模块状态装换.....	5
2.2 系统状态指示灯.....	5
2.3 GCAN-402 节点 ID 与 CAN 总线波特率.....	6
3. GCAN-402 中使用 CANopen 协议 (DS301) .....	8
3.1 GCAN-402 预定义连接.....	8
3.2 GCAN-402 操作 .....	8
附录 A: CANopen 协议简介.....	21
附录 B: GCAN-402 对象字典.....	25
销售与服务.....	29

# 1. 功能简介

## 1.1 功能概述

GCAN-402 是广成科技有限公司开发的 CANopen 从站数据存储模块。该模块内部已经集成了 CANopen 从站协议栈代码，不需要用户进行二次开发。协议栈遵循 CANopen 协议描述文档 DS301、DS302、DS303 以及 DS305 标准。在默认情况下，CANopen 从站启用预定义连接，并支持参数存储。

GCAN-402 提供两路 CAN 接口，通过该接口可以与 CANopen 网络进行连接。模块可以支持多达 64k 的数据存储，适用于各种干扰强、实时性要求高的工业场合，极其小巧的体积使 GCAN-402 模块非常易于嵌入到用户设备中，高速实时的处理能力和灵活多样的应用使该模块被广泛应用于 CANopen 通信网络中。我公司还可为用户提供个性化定制服务，可以单独为用户进行 EDS 文件和程序固件定制。

我公司 GCAN-402 可升级内核程序，我公司长期帮助用户维护产品的 CANopen 功能，时刻满足客户的需求。

## 1.2 性能特点

- 网络管理服务对象(NMT: Boot up, Node Guarding/Life guarding, Heartbeat Producer);
- 过程数据对象(12 个 TPDO 与 12 个 RPDO);
- 支持服务数据对象(SDO 服务器);
- 支持紧急报文对象(Emergency);
- 支持同步报文对象(Sync);
- 支持网络配置对象(LSS 从站);
- 支持支持网络时间消费;
- 工作电压+24V; 额定电压 24V, 可在 10V 到 30V 范围工作;
- 工作温度和存储温度: -40℃~85℃;
- 外壳防护 IP67。

## 1.3 典型应用

- PLC 设备访问连接 CANopen 网络;
- CANopen 从站开发;
- CANopen 工业自动化控制系统;
- 大量设备使用 CANopen 统一控制。

## 2. 设备使用

### 2.1 模块状态转换

GCAN-402各状态转换图如图2.1所示，图中各字母所表示的各种状态下可进行的操作，字母表达的操作为：

- a. NMT
- b. Node Guard
- c. SDO
- d. Emergency
- e. PDO
- f. Boot-up。

图中箭头所表示各个状态之间的转换关系，数字表示这种转换所需要进行的操作，数字表达的操作为：

- 1: Start\_Remote\_node (0x01)
- 2: Stop\_Remote\_Node (0x02)
- 3: Enter\_Pre-Operational\_State (0x80)
- 4: Reset\_Node (0x81)
- 5: Reset\_Communication (0x82)
- 6: 设备初始化结束，自动进入Pre\_Operational状态，发送Boot-up消息

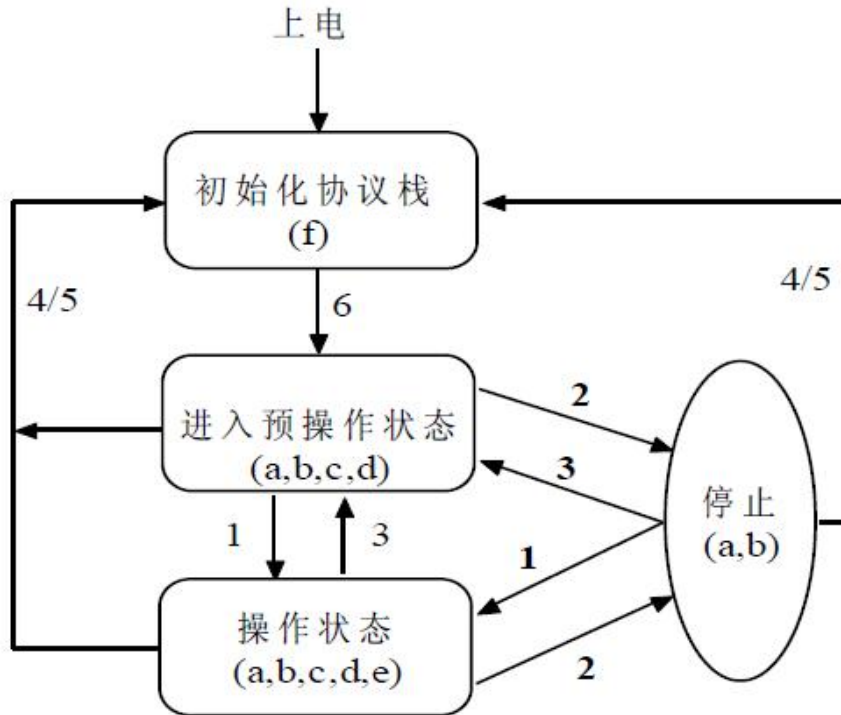


图2.1 GCAN-402各状态转换图

### 2.2 系统状态指示灯

按照CANopen协议规范文档DS303-3的定义，在GCAN-402模块中使用两个LED指示灯来指示当前模块所处的状态，如表2.1所示。

指示灯名称	颜色	管脚
运行指示灯 (RUN)	绿色	13
错误指示灯 (ERR)	红色	14

表2.1 系统状态指示灯

其状态指示灯所指的各种状态含义如表2.2、表2.3所示。

编号	ERR LED	状态	描述
1	暗	没有错误	器件处于工作状态
2	闪一下	到达警戒值	CAN 控制器的至少一个错误计数器到达或超出了警戒值 (错误帧太多)
3	亮	总线关闭	CAN 控制器总线关闭

表2.2 错误状态指示灯 (ERR) 状态意义

编号	RUN LED	状态	描述
1	暗	故障	请检查模块复位引脚及电源是否连接正确
2	闪	预操作	模块处于预操作状态
3	亮	工作	模块处于工作状态

表2.3 运行状态指示灯 (RNU) 状态意义

### 2.3 GCAN-402 节点 ID 与 CAN 总线波特率

GCAN-402 模块提供三种方式设置节点 ID 和节点波特率，其设置顺序如图 2.2 和图 2.3 所示。如果用户所使用的拨码开关值有效，则上电时启用该值，即使在存储器中存储有合法的 ID 值都不会使用。如果模块在运行期间，主站对模块进行了 LSS 设置，则使用 LSS 设置后的值，但是模块重新上电或重启之后，依然使用拨码开关的值。

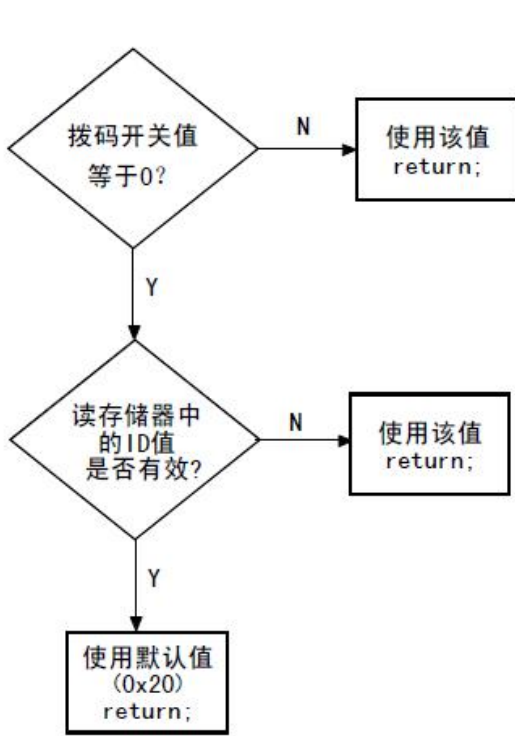


图2.2 节点ID设置顺序

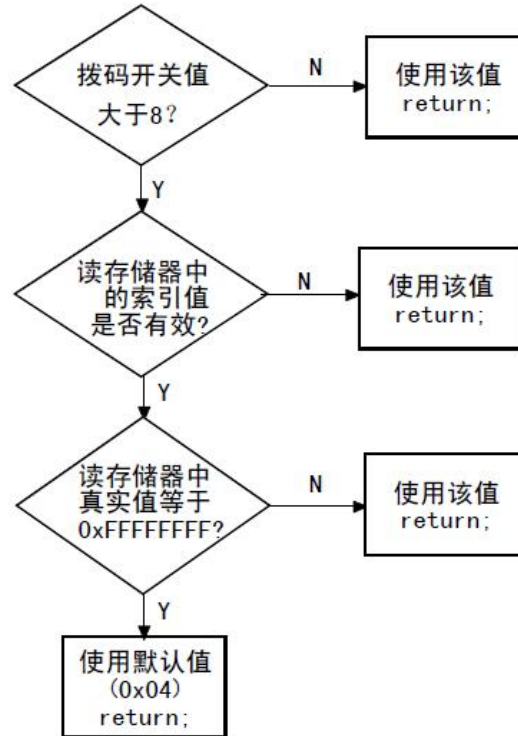


图2.3 波特率设置顺序

其中设置节点 ID 的拨码开关使用 7 位，当其值为 0 时，表示拨码开关无效，取值范围为 1~127 之间。波特率索引值使用 4 位拨码，当该拨码开关值大于 8 时无效，其有效的取值范围为 0~8，波特率索引值与标准波特率的对应关系如表 2.4 所示。如果该表格中的标准波特率无法满足需求，则用户可以自己计算波特率值（32 位长度的波特率定时参数）来设置模块，当拨码开关和波特率索引值都无效时，则模块会自动检测该值是否有效（值为 0xFFFFFFFF 和 0x00000000 时无效），如果已经设定波特率索引值则优先使用该值初始 CAN 通信波特率。其中 CAN 控制器的工作频率为 24MHz。

波特率索引值	标准波特率
0	1Mbps
1	800Kbps
2	500Kbps
3	250Kbps
4	125Kbps
5	100Kbps
6	50Kbps
7	20Kbps
8	10Kbps

表2.4 波特率索引值与实际对照关系

### 3. GCAN-402 中使用 CANopen 协议 (DS301)

#### 3.1 GCAN-402 预定义连接

在 GCAN-402 模块中使用了对象字典的 0x1000~0x1FFF 和厂商自定义区 0x2000~0x5FFF 的区域。这些对象字典负责着 CANopen 与 CAN 网络上的其它应用数据的通信和数据交换，对象字典使用了索引与子索引来定义，每个对象字典项都有自己的数据长度 (UIN8, UIN16, UIN32 等) 和属性 (RO、WO、RW、CONST、MAPPALE)。这些对象字典的数据可以通过 SDO 服务来修改，当然也只有这些项的属性必须是 WO 或 RW 才能修改。

预定义连接是指与通信相关的 COB-ID 与节点 ID 相关联。具体的预定义连接集如表 3.1 所示。

对象	功能码	节点地址	COB-ID	对象字典索引
广播报文				
NMT	0000	-	0	-
SYNC	0001	-	0x80	0x1005, 0x1006, 0x1007
TIME STAMP	0010	-	0x100	0x1012, 0x1013
点对点报文				
紧急报文	0001	1-127	0x81-0xFF	0x1014, 0x1015
TPDO1	0011	1-127	0x181-0x1FF	0x1800
RPDO1	0100	1-127	0x201-0x27F	0x1400
TPDO2	0101	1-127	0x281-0x2FF	0x1801
RPDO2	0110	1-127	0x301-0x37F	0x1401
TPDO3	0111	1-127	0x381-0x3FF	0x1802
RPDO3	1000	1-127	0x401-0x47F	0x1402
TPDO4	1001	1-127	0x481-0x4FF	0x1803
RPDO4	1010	1-127	0x501-0x57F	0x1403
默认 SDO (tx)	1011	1-127	0x581-0x5FF	0x1200
默认 SDO (rx)	1100	1-127	0x601-0x67F	0x1200
NMT 错误控制	1110	1-127	0x701-0x77F	0x1016, 0x1017

表3.1 CANopen预定义连接集

#### 3.2 GCAN-402 操作

##### 3.2.1 网络管理服务 (NMT)

###### 1. 网络控制 (NMT Module Control)

GCAN-402 支持 DS301 所定义的网络管理命令，这些网络管理命令可以是 CANopen 主站发出也可以是其它的从节点发出。其操作命令如表 3.2 所示，其中当 Node\_ID=0 时，则所有的从站设备被控制(广播方式)，CS 为命令字对应着不同的控制动作如表 3.3 所示。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1
0x000	2	CS(命令字)	NodeID(节点号)

表3.2 NMT操作指令



CS (命令字)	NMT 服务 (控制动作)
0x01	启动从站设备
0x02	停止从节点设备
0x80	使从站进入预操作
0x81	复位从节点
0x82	复位节点通信

表3.3 NMT命令字及相应功能服务

例：需要启动 CANopen 网络中所有的节点，可使用如下表 3.4 所示的命令。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1
0x000	2	0x01	0x00

表3.4 NMT启动从节点

如果需要控制网络中某个具体设备，使其进入到预操作状态，假设节点地址为 0x03，则命令如表 3.5 所示。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1
0x000	2	0x80	0x03

表3.5 NMT启动指定从节点

## 2. 节点保护 (NMT Node Guarding)

通过节点保护服务，NMT 主节点可以检查每个节点的当前状态，当这些节点没有数据传送时这种服务尤其有意义。主节点通过发送远程帧来触发相应从节点的节点保护，其命令格式如表 3.6 所示，从节点应答相应格式如图 3.7 所示。

主节点→从节点(命令):

COB-ID(CAN-ID)	DLC
0x700 + NodeID	1

表3.6 NMT主节点保护命令帧(远程帧)

从节点→主节点(响应):

COB-ID(CAN-ID)	DLC	BYTE0
0x700 + NodeID	1	Bit7:触发位, Bit0~Bit6 状态

表3.7 NMT从节点应答帧

其中 Byte0 中的最高位(bit7)为触发位，即从站每发一帧应答就会交替变化(0、1)，以示帧与帧之间的区别，其中 Bit0~Bit6 为从节点的状态，该值所表达从站所处的状态如表 3.8 所示。

值 (Value)	所处状态
0x00	初始化 (Initialising)
0x04	停止状态 (Stopped)
0x05	操作状态 (Operational)
0x7F	预操作状态 (Pre-operational)

表3.8 节点保护状态值

例：假设主节点需要对节点号为 0x20 从节点进行节点保护，其命令如表 3.9 所示，从节点应答帧如表 3.10 所示。

主节点→从节点：

COB-ID(CAN-ID)	DLC
0x720	1

表3.9 保护节点(远程帧)

从节点→主节点：

COB-ID(CAN-ID)	DLC	BYTE0
0x720	1	0x85

表3.10 从节点(0x20)应答帧

其中 BYTE0 的 Bit7=1，状态=0x05，表示节点号为 0x20 的从站正处于操作状态。

### 3. 寿命保护 (NMT Life Gardting)

其中节点保护主要针对的是 NMT 主节点获取从节点的状态，而寿命保护主要是节点对另一节点的监控。寿命保护包括两个参数，即保护时间和生命因子，启用寿命保护的节点接收来自另一节点的远程帧（远程帧格式与节点保护帧格式相同如表 3.6），启用寿命保护的节点接收到该远程帧则应答该节点的状态（应答帧格式如表 3.7 所示）。

寿命保护的两个参数：保护时间和生命因子（分别位于对象字典的 0x100C 和 0x100D）构成了节点的寿命时间（即寿命时间=保护时间 x 生命因子），保护时间的单位为毫秒，如果两个参数中有一个为 0 则表示寿命保护未启用。如果在保护时间内未接收到远程帧则会出现“Message Lost”的提示信息，在寿命时间内未接收到远程帧则会出现“Connection Lost”信息，这些信息均在调试串口中打印出来，同时错误指示灯出现“闪烁两下”以示当前的寿命保护丢失。

### 4. 启动报文(NMT Boot-up)

当 GCAN-402 初始化完成(Boot-up)，就会发送一个标识报文，其报文格式如表 3.11 所示。

COB-ID(CAN-ID)	DLC	BYTE0
0x700 +NodeID	1	0x00

表3.11 初始化完成标识报文格式

例：假设GCAN-402的节点号为0x20，则发送的启动文如表3.12所示。

COB-ID(CAN-ID)	DLC	BYTE0
0x720	1	0x00

表3.12 0x20节点初始化完成标识报文格式

### 5. 心跳报文 (Heartbeat Producer)

心跳报文分为生产者和消费者，在 GCAN-402 模块中只支持心跳报文生产，即 GCAN-402 可以生产心跳报文。该参数在对象字典 0x1017 中定义（数据长度 16 位，单位：毫秒），其心跳报文如表 3.7 所示，与节点保护和寿命保护的应答帧相同。

例：假设节点地址为 0x20，处于操作状态，0x1017 中的参数设置为 100，则该从节点每隔 100 毫秒发送一帧如表 3.13 所示的报文。

COB-ID(CAN-ID)	DLC	BYTE0
0x720	1	0x05

表3.13 从节点(0x20)心跳报文

**注意：**在同一个 GCAN-402 模块中同一时间寿命保护和心跳报文不能同时使用。

### 3.2.2 同步报文对象(SYNC)

同步报文分为消费和生产，在 GCAN-402 中只支持同步报文的消费，即接收来自主节点或其它节点的同步报文，同步报文的帧结构如表 3.14 所示。对象字典的 0x1005 定义了接收同步报文的 COB-ID，在 CANopen 预定义连接集里定义其值为 0x80，对象字典的 0x1007 定义了同步的时间窗口（在接收到同步报文后要求更新相应数据的最长时间间隔）。同步报文主要应用在 PDO 接收和发送的过程中，其使用方法在以下 PDO 数据发送和接收过程中详细介绍。

COB-ID(CAN-ID)	DLC
0x80	0

表3.14 同步报文帧格式（远程帧）

### 4.2.3 紧急报文对象(EMCY)

在 GCAN-402 中支持紧急报文，即在 GCAN-402 内部出现错误时发送该报文，其报文格式如表 3.15 所示。其中紧急错误码指定当前出现的错误的具体类型。错误寄存器存放当前错误类型，根据该值可以判断出当前紧急报文所代表的错误类型，其值定义如表 3.16 所示。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1	BYTE2	BYTE3—BYTE7
0x80 + NodeID	8	紧急错误码		错误寄存器	生产厂商指定的信息
		索引 0x1003		索引 0x1001	

表3.15 紧急报文帧格式

位(Bit)	错误类型
0	普通错误(Generic)
1	电流错误(Current)
2	电压错误(Voltage)
3	温度错误(Temperature)
4	通信错误(Communication)
5	设备描述错误(Device profile specific)
6	Reserved(=0)
7	生产厂商定义的错误(Manufacturer specific)

表3.16 错误寄存器

紧急错误代码含义如表 3.17 所示。

应急错误代码	代码功能描述
00xx	Error Reset 或 No Error
10xx	Generic Error
20xx	Current
21xx	Current, device input side
22xx	Current, inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains voltage
32xx	Voltage inside the device
33xx	Output voltage
40xx	Temperature
41xx	Ambient temperature
42xx	Device temperature
50xx	Device hardware
60xx	Device software
61xx	Internal software
62xx	User software
63xx	Data set
70xx	Additional modules
80xx	Monitoring
81xx	communication
8110	CAN overrun
8120	Error Passive
8130	Life Guard Error 或 Heartbeat Error
8140	Recovered from Bus-Off
82xx	Protocol Error
8210	PDO no processed Due to length error
8220	Length exceedd
90xx	External error
F0xx	Additional functions
FFxx	Device specific

表3.17 紧急错误代码

例：假设节点地址为0x20，CAN总线错误超过警戒值则出现“Error Passive”的警告，则GCAN-402发送紧急报文如表3.18所示。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1	BYTE2	BYTE3—BYTE7
0xA0	8	0x20	0x81	0x11	0x00000000

表3.18 紧急报文（总线错误）

注意：GCAN-402模块发生紧急情况时，这些错误主动发出。

### 3.2.4 服务数据对象(SDO)

对象字典充当应用层和通信层之间的主要数据交换媒介。一个CANopen设备的所有数据项可以在对象字典中被管理。每个对象字典项可以使用索引和子索引来定位。CANopen定义通常所说的服务数据对象（SDO）来访问这些项。GCAN-402支持1个SDO服务器，即可以提供SDO服务，且SDO使用预定义连接的发送和接收COB-ID，0x580 + NodeID(发送)和0x600 + NodeID(接收)。SDO分为加速传输、段传输、和块传输。因为在GCAN-402中SDO的加速传输经常会使用，所以本说明文档重点介绍加速传输和段传输，其它的传输类型可查阅CANopen DS301及相关的协议文档。

### 1. SDO 数据传输

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

SDO的基本结构如表3.19或表3.20所示。（ Client → Server / Server → Client）

Byte 0	Byte 1-2	Byte 3	Byte 4-7
SDO Command Specifier	对象索引	对象子索引	**

表3.19 SDO基本结构1

（\*\*最大4字节数据(expedited transfer)或4字节计数器（segmented transfer）或关于block transfer参数）

Byte 0	Byte 1-7
SDO 命令字	最大 7 字节数据（segmented transfer）

表3.20 SDO基本结构2

SDO 命令字包含如下信息：

- 下载/上传（Download / upload）
- 请求/应答（Request /response）
- 分段/加速传送（Segmented / expedited transfer）
- CAN帧数据字节长度
- 用于后续每个分段的交替清零和置位的触发位（toggle bit）

SDO有2种传送机制：

加速传送（Expedited transfer）：最多传输4字节数据。

分段传送（Segmented transfer）：传输数据长度大于4字节。

SDO中实现了5个请求/应答协议：启动域下载（Initiate Domain Download）、域分段下载（Download Domain Segment）、启动域上传（Initiate Domain Upload）、域分段上传（Upload Domain Segment）和域传送中止（Abort Domain Transfer）。

在CANopen通讯协议的最新版本中，引入了一种新的SDO传送机制——块传送（Block transfer）：当传送数据长度大于4字节时，多个分段只由1个确认报文应答（如果是下载，则由服务器启动传送，如果是上传，则由客户启动传送）以增加总线吞吐量。

相应的协议为：启动块下载（Initiate Block Download）、块分段下载（Download Block Segment）、块下载结束（End Block Download）、启动块上传（Initiate Block Upload）、块分段上传（Upload Block Segment）和块上传结束（End Block Upload）。

下载（Download）是指对对象字典进行写操作，上传（Upload）指对对象字典进行读操作。

这些协议的SDO命令字（SDO CAN报文的第一个字节）语法和细节在下面部分说明：（“-”表示不相关，应为0）。

启动域下载（Initiate Domain Download）								
Bit	7	6	5	4	3	2	1	0
Client	0	0	1	-	n		e	s
Server	0	1	1	-	-	-	-	-

说明：

n：如果e=1，且s=1，则有效，否则为0；表示数据部分中无意义数据的字节数（字节8-n到7数据无意义）。

e：0=正常传送，1=加速传送。

s：是否指明数据长度，0=数据长度未指明，1=数据长度指明。

e=0，s=0：由CiA保留。

e=0，s=1：数据字节为字节计数器，byte 4是数据低位部分（LSB），byte 7是数据高位部分（MSB）。

e=1：数据字节为将要下载（download）的数据。

域分段下载（Download Domain Segment）								
Bit	7	6	5	4	3	2	1	0
Client	0	0	0	t	n			c
Server	0	0	1	t	-	-	-	-

说明：

n：无意义的的数据字节数。如果没有指明段长度，则为0。

c：0=有后续分段需要download，1=最后一个段。

t：触发位，后续每个分段交替清零和置位（第一次传送为0，等效于request/response）。

启动域上传（Initiate Domain Upload）								
Bit	7	6	5	4	3	2	1	0
Client	0	1	0	-	-	-	-	-
Server	0	1	0	-	n		e	s

说明：

n，e，s：与启动域下载相同。

域分段上传（Upload Domain Segment）								
Bit	7	6	5	4	3	2	1	0
Client	0	1	1	t	-	-	-	-



Server	0	0	0	t	n	c
--------	---	---	---	---	---	---

说明:

n, c, t : 与域分段下载相同。

SDO客户或服务器通过发出如下格式的报文来中止SDO传送:

域传送中止 (Abort Domain Transfer)								
Bit	7	6	5	4	3	2	1	0
Client/Server	1	0	0	-	-	-	-	-

## 2. SDO中止服务

在SDO的传输过程中当出现错误，SDO的客户端和服务器都可以发送这个报文来通知对方中止当前的操作，中止报文的格式如表3.21和表3.22所示。具体的中止错误代码含义可参照表3.23。

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x600 + NodeID /0x58+NodeID	8	CMD(SDO 命令字)	索引	子索引	**

表3.21 终止报文格式

CMD (SDO 命令字节)									
Bit	7	6	5	4	3	2	1	0	
Client→Server/ Client→Server	1	0	0	-	-	-	-	-	

表3.22 中止报文命令字定义

中止代码	代码功能描述
0503 0000	触发位没有交替改变
0504 0000	SDO 协议超时
0504 0001	非法或未知的 Client/Server 命令字
0504 0002	无效的块大小 (仅 Block Transfer 模式)
0504 0003	无效的序号 (仅 Block Transfer 模式)
0503 0004	CRC 错误 (仅 Block Transfer 模式)
0503 0005	内存溢出
0601 0000	对象不支持访问
0601 0001	试图读只写对象
0601 0002	试图写只读对象
0602 0000	对象字典中对象不存在
0604 0041	对象不能够映射到 PDO
0604 0042	映射的对象的数目和长度超出 PDO 长度
0604 0043	一般性参数不兼容
0604 0047	一般性设备内部不兼容
0606 0000	硬件错误导致对象访问失败
0606 0010	数据类型不匹配, 服务参数长度不匹配
0606 0012	数据类型不匹配, 服务参数长度太大

0606 0013	数据类型不匹配，服务参数长度太短
0609 0011	子索引不存在
0609 0030	超出参数的值范围(写访问时)
0609 0031	写入参数数值太大
0609 0032	写入参数数值太小
0609 0036	最大值小于最小值
0800 0000	一般性错误
0800 0020	数据不能传送或保存到应用
0800 0021	由于本地控制导致数据不能传送或保存到应用
0800 0022	由于当前设备状态导致数据不能传送或保存到应用
0800 0023	对象字典动态产生错误或对象字典不存在（例如，通过文件生成对象字典，但由于文件损坏导致错误产生）

表3.23 终止错误代码表

### 3.2.5 过程数据对象（PDO）

过程数据对象（PDO）用作传输实时数据，在出厂时预定义连接集所定义的4个TPDO(索引范围0x1800~0x1803)和4个RPDO(0x1400~0x1403)可用。

#### 1. 过程数据接收（RPDO）

在GCAN-402出厂时已经为每个PDO预定义了映射对象，如图3.1所示，全部RPDO的数据映射项都默认连接到GCAN-402模块的8bit输出区。即当RPDO接收到来自网络的数据之后，把数据更新到所对应的输出数据区，当数据更新完成之后，GCAN-402会给出一个中断信号（高电平→低电平），当数据未被读取时中断信号引脚将一址保持低电平，数据被读出之后中断引脚将保持在高电平。



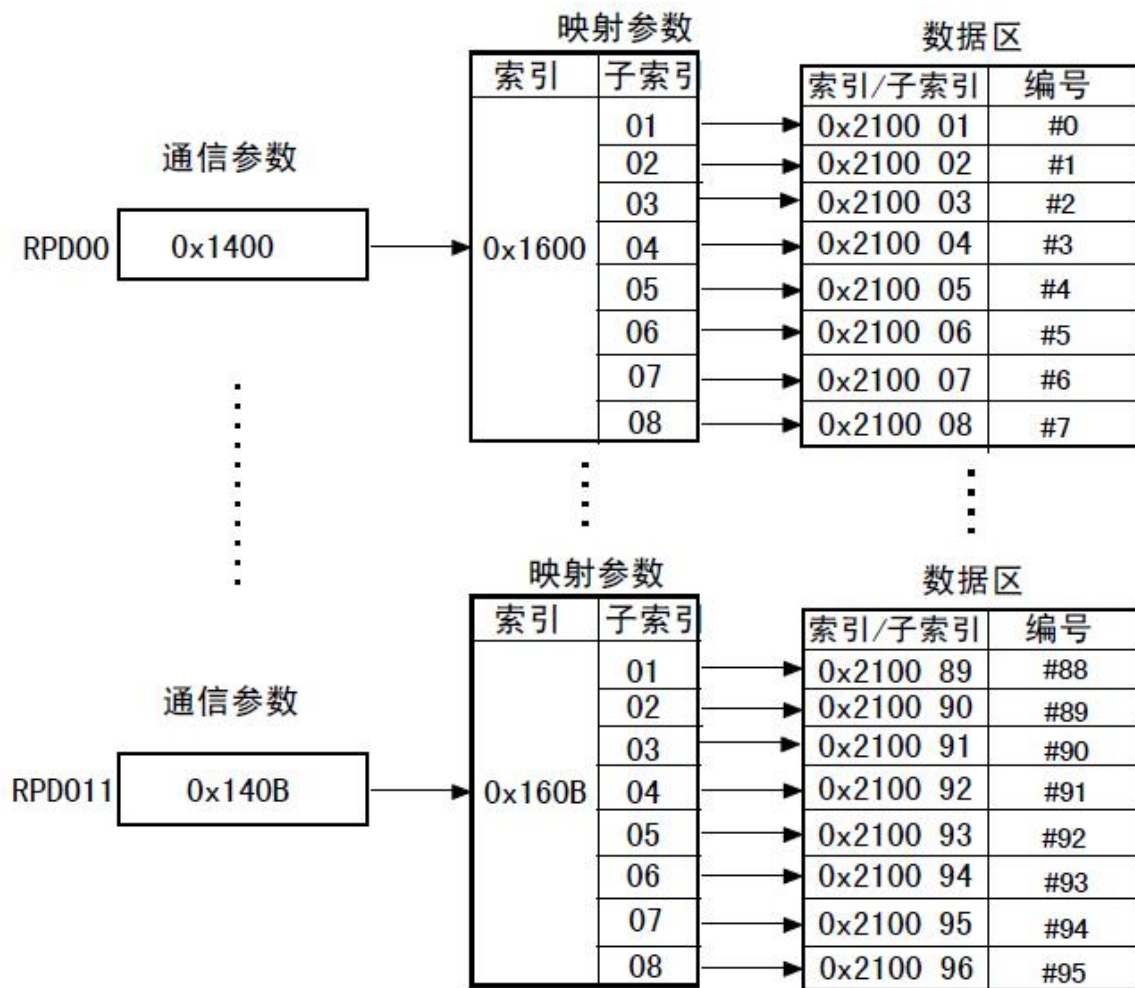


图3.1 RPDO映射关系

RPDO的通信参数相比TPDO来说比较少，RPDO只有传输类型（transmission type）一项对应通信。其值定义如表3.24所示。

传输类型	PDO 接收	数据更新
0	PDO 将一直被接收，分析。如果需要，在接收到下个有效的 SYNC 报文时对数据进行更新。	在接收到一个 SYNC 报文时对数据进行分析。如果与之前的 RPDO 相比数据已经更改了，那么数据将在输出上被更新。SYNC 报文的传输是非循环的。
1-240		在接收到第 n 个编号的 SYNC 报文时对数据进行分析。如果与之前的 RPDO 相比数据已经更改了，那么数据将在输出上被更新。传输类型与值 n 相对应。SYNC 报文的传输是循环的。
241-251		保留
252		保留
253		保留

254	PDO 将一直被接收到。	应用定义更新输出数据的事件。
255	PDO 将一直被接收到。	设备子协议定义更新输出数据的事件。

表3.24 RPDO的传输类型

例：假设GCAN-402节点为0x20，采用预定义连接，则RPDO0的COB-ID为0x220。则其接收的TPDO COB-D也应为0x220如表3.25所示，该TPDO正好与节点NodeID为0x20的RPDO1的COB-ID相同，则该RPDO接收这帧PDO数据，并且按照图4.1所示的映射关系图把数据更新到数据输出区，最后输出缓冲区对应的数据如表3.26所示。

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x220	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88

表3.25 RPDO0接收其它节点的TPDO

数据区编号	#0	#1	#2	#3	#4	#5	#6	#7
数据	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88

表3.26 数据区数据

## 2. 过程数据发送 (TPDO)

在GCAN-402中最多支持12个TPDO，在出厂时预定义的4个PDO可用，即TPDO0~TPDO3，其中TPDO4~TPDO11不可用。预定义的TPDO都已经在出厂时已经预定义了映射参数，分别映射到数据输入区0x2000 01~0x2000 96，如图3.2所示。

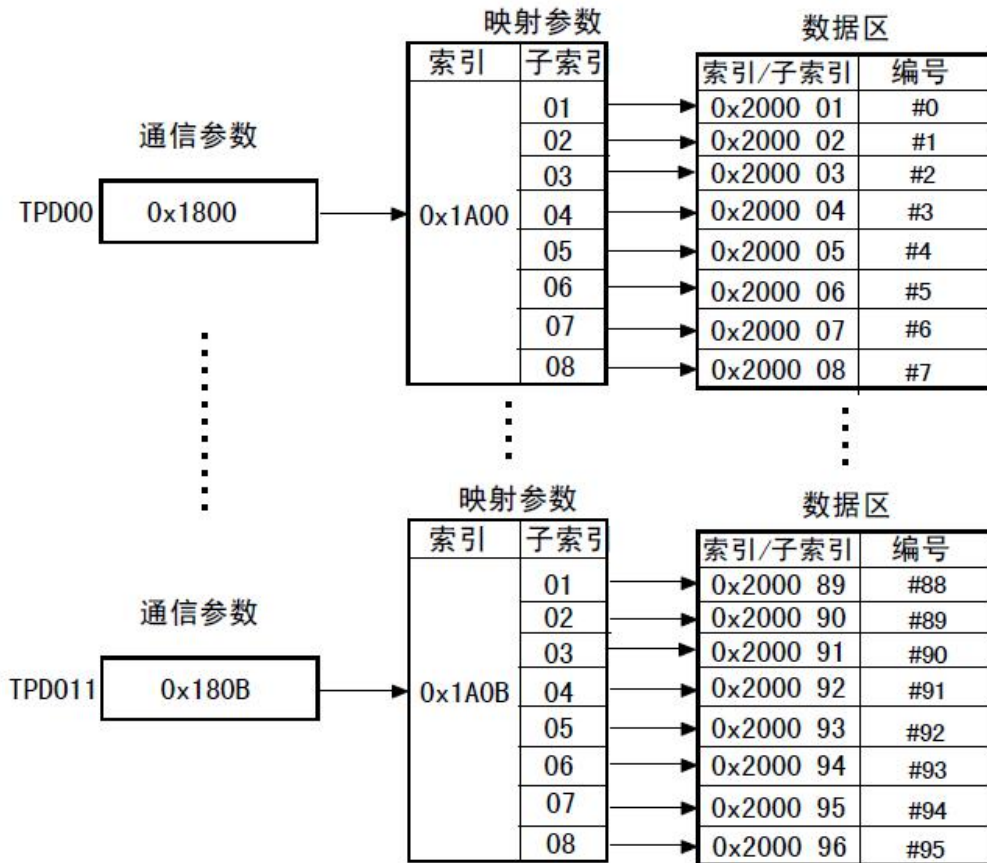


图3.2 TPDO 映射关系

每个TPDO都包含有相应的通信参数，这些通信参数决定着TPDO发送的类型以及发送的触发条件等。其中参数主要包含有三种，分别为传输类型、禁止时间以及事件定时。

● 传输类型 (Transmission type)

传输类型定义了该TPDO传输方式，通信参数的子索引2定义该对象，具体值定义如表4.27所示。

传输类型	数据请求	发送 PDO
0	数据（输入值）在接收到一个 SYNC 报文时被读取。	如果与之前的 PDO 内容相比 PDO 数据已经更改了，那么 PDO 将被发送。
1—240	数据接收第 n 个编号的 SYNC 报文时被收集和更新，然后在总线上发送。 传输类型对应值 n。	
241—251	保留	
252	数据（输入值）在接收到一个 SYNC 报文时被读取。	PDO 在请求时通过一个远程帧被发送。
253	应用持续收集和更新输入数据。	
254	应用定义引发数据请求和 PDO 传输的事件。造成 PDO 传输的事件可以是事件定时器的时间已到。事件定时器周期用子索引 5 来配置。PDO 传输（与事件和事件定时器是否被配置都无关）总是启动一个新的事件定时器周期。	
255	设备子协议定义引发数据请求和 PDO 传输的事件。造成 PDO 传输的事件可以是事件定时器的时间已到。事件定时器周期用子索引 5 来配置。PDO 传输（与事件和事件定时器是否被配置都无关）总是启动一个新的事件定时器周期。	

表3.27 TPDO传输类型

● 禁止时间(Inhibit Time)

禁止时间的定义是为了防止TPDO发送过于频繁而占用大量的总线带宽，从而影响到总线通信。因而定义了同一个TPDO发送PDO的最短时间间隔（单位为毫秒），当该参数为0时无效，在通信参数子索引3中定义。

● 定时时间(Event Time)

定时时间参数定义了该PDO的发送循环时间（单位为毫秒），需要PDO的传输类型设置为254或255，当该参数为0时无效，在通信参数子索引5中定义。

例：假设当前的节点NodeID为0x20，TPDO0定时时间（Event time）参数为1000，传输类型（Transmission type）为254，数据输入区#0~#7号的数据为0x18，则TPDO0发送数据如图3.3所示。

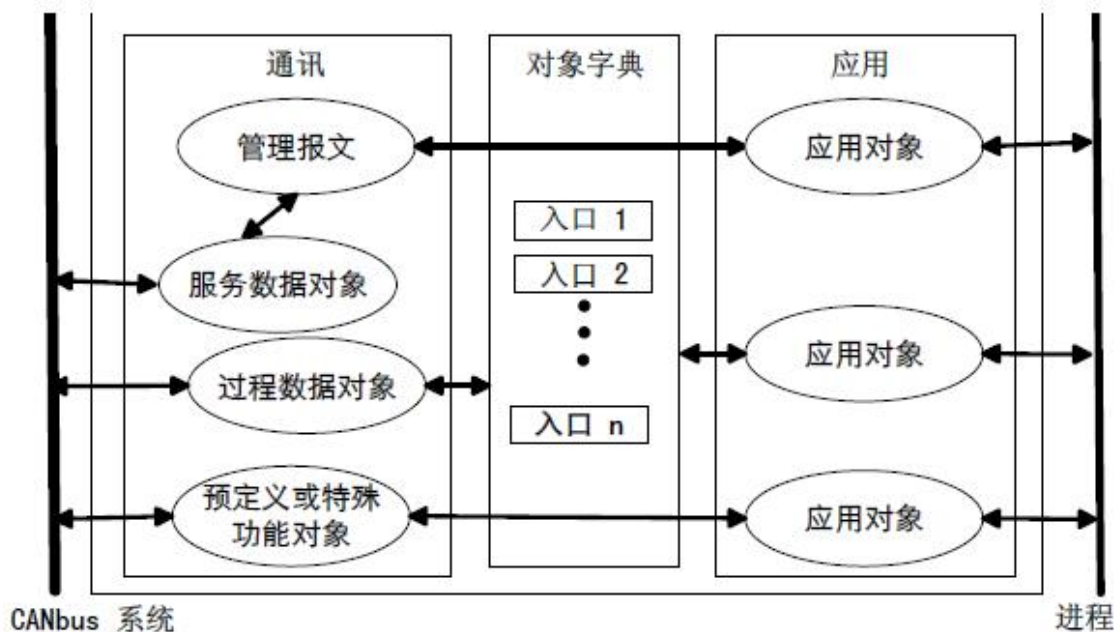
帧间隔时间为1000毫秒

序号	帧间隔时间us	名称	帧ID	帧类型	帧格式	DLC	数据	帧数量
00000023	1.000.782	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
00000024	1.000.781	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
00000025	1.000.781	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
00000026	1.000.781	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
00000027	1.000.781	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
00000028	1.000.782	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
00000029	1.000.781	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1
0000002A	1.000.781	接收	1A0	DATA	STANDARD	8	18 18 18 18 18 18 18 18	1

图 3.3 TPDO 发送出的数据

## 附录 A: CANopen 协议简介

CANopen协议是在20世纪90年代末，由CiA组织（CAN-in-Automation）在CAL（CAN Application Layer）的基础上发展而来，一经推出便在欧洲得到了广泛的认可与应用。经过对CANopen协议规范文本的多次修改，使得CANopen协议的稳定性、实时性、抗干扰性都得到了进一步的提高。并且CiA在各个行业不断推出设备子协议，使CANopen协议在各个行业得到更快的发展与推广。目前CANopen协议已经在运动控制、车辆工业、电机驱动、工程机械、船舶海运等行业得到广泛的应用。



图A1 CANopen设备结构

图A1所示为CANopen设备结构，CANopen协议通常分为用户应用层、对象字典、以及通讯三个部分。

### A.1 相关名词解释和书写规则

#### 1. 名词解释：

PDO: Process Data Object, 过程数据对象。

TPDO: Transmit Process Data Object, 发送过程数据对象。

RPDO: Receive Process Data Object, 接收过程数据对象。

SDO: Service Data Object, 服务数据对象。

NMT: Network Management, 网络管理。

SYNC: Synchronization Objects, 同步报文对象。

EMCY: Emergency Objects, 紧急对象报文。

CAN-ID: Controller Area Network-Identify, 控制器局域网标识符。

COB-ID: Communication Object-Identify, 通信对象标识符。

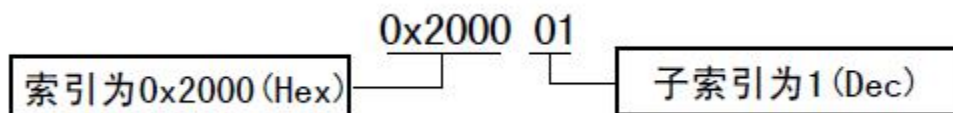
SSDO: Servers Service Data Object, 服务数据服务器。

DS: Draft Standard, 标准草案。



## 2. 书写规则

本手册中，对象字典索引与子索引的书写遵循如下图A2所示的规则，其中索引为16进制表示，子索引为10进制表示，索引与子索引中间用空格隔开。



图A2 索引/子索引书写规则

### A.2 CANopen对象字典

CANopen对象字典(OD: Object Dictionary)是CANopen协议最为核心的概念。所谓的对象字典就是一个有序的对象组，每个对象采用一个16位的索引值来寻址，这个索引值通常被称为索引，其有效范围在0x1000到0x9FFF之间。为了允许访问数据结构中的单个元素，同时也定义了一个8位的索引值，这个索引值通常被称为子索引。每个CANopen设备都有一个对象字典，对象字典包含了描述这个设备和它的网络行为的所有参数，对象字典通常用电子数据文档（EDS: Electronic Data Sheet）来记录这些参数，而不需要把这些参数记录在纸上。对于CANopen网络中的主节点来说，不需要对CANopen从节点的每个对象字典项都访问。CANopen对象字典中的项由一系列子协议来描述。子协议为对象字典中的每个对象都描述了它的功能、名字、索引、子索引、数据类型，以及这个对象是否必需、读写属性等等，这样可保证不同厂商的同类型设备兼容。CANopen协议的核心描述子协议是DS301，其包括了CANopen协议应用层及通信结构描述，其它的子协议都是对DS301协议描述文本的补充与扩展。CANopen协议包含了许多子协议，其主要划分为以下类型。

#### 1. 通讯子协议（Communication Profile）

通讯子协议，描述对象字典的主要形式和对象字典中的通讯对象以及参数。这个子协议适用所有的CANopen设备，其索引值范围从0x1000~0x1FFF。

#### 2. 制造商自定义子协议（Manufacturer-specific Profile）

制造商自定义子协议，对于在设备子协议中未定义的特殊功能，制造商可以在此区域根据需求定义对象字典对象。因此这个区域对于不同的厂商来说，相同的索引的对象字典项定义不一定相同，其索引值范围为0x2000~0x5FFF。

### A.3 CANopen通讯

在CANopen协议中主要定义了管理报文对象NMT（Network Management）、服务数据对象SDO(Service Data Object)、过程数据对象PDO(Process Data Object)、预定义报文或特殊功能对象等四种对象。

#### 1. 网络管理NMT（Network Management）

管理报文负责层管理、网络管理和ID分配服务，例如，初始化、配置和网络管理（其中包括节点保护）。网络管理中，同一个网络中只允许有一个主节点、一个或多个从节点，并遵循主从模式。通过NMT 服务，我们可以对节点进行初始化、运行、监控、复位和停止。

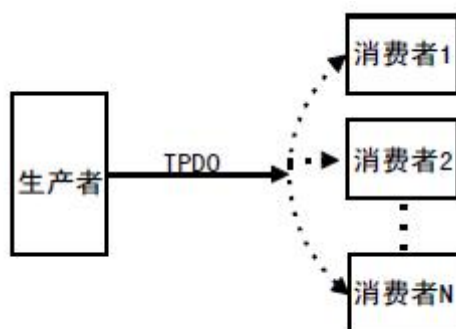
所有节点都被认为是NMT 从站。

## 2. 服务数据对象SDO (Service Data Object)

SDO主要用于主节点对从节点的参数配置。服务确认是SDO的最大的特点，为每个消息都生成一个应答，确保数据传输的准确性。在一个CANopen系统中，通常CANopen从节点作为SDO服务器，CANopen主节点作为客户端。客户端通过索引和子索引，能够访问数据服务器上的对象字典。这样CANopen主节点可以访问从节点的任意对象字典项的参数，并且SDO也可以传输任何长度的数据（当数据长度超过4个字节时就拆分成多个报文来传输）。

## 3. 过程数据对象PDO (Process Data Object)

PDO用来传输实时数据，其传输模型为生产者消费者模型如图A3所示。数据长度被限制为1~8字节。PDO通信对象具有如下的特点：



图A3 生产者消费者模型

- PDO通讯没有协议规定，PDO数据内容由它的CAN-ID（也可称为COB-ID）定义；
- 每个PDO在对象字典中用2个对象描述：
  - >PDO通讯参数，该通讯参数定义了设备所使用的COB-ID、传输类型、定时周期；
  - >PDO映射参数，映射参数包含了一个对象字典中的对象列表，这些对象映射到相应的PDO，其中包括数据的长度（单位：位），对于生产者和消费者都必须要知道这个映射参数，才能够正确的解释PDO内容。
- PDO消息内容是预定义的，如果PDO支持可变PDO映射，那么该PDO是可以通过SDO进行配置；
- PDO可以有多种的传输方式：
  - >同步传输（通过接收同步对象实现同步），同步传输又可分为非周期和周期传输。非周期传输是由远程帧预触发或者由设备子协议中规定的对象特定事件预触发传送。周期传输则是通过接收同步对象（SYNC）来实现，可以设置1~240个同步对象触发；
  - >异步传输（由特定事件触发），其触发方式可有两种，第一种是通过发送与PDO的COB-ID相同的远程帧来触发PDO的发送，第二种是由设备子协议中规定的对象特定事件来触发（例如，定时传输，数据状态变化传输等）。

## 4. 预定义报文或特殊功能对象

预定义报文或特殊功能对象为CANopen设备提供特定的功能，方便CANopen主站对从站管理。在CANopen协议中，已经为特殊的功能预定义了COB-ID，其主要有以下几种特殊报文：

- 同步（SYNC），该报文对象主要实现整个网络的同步传输，每个节点都以

该同步报文作为PDO同步触发参数，因此该同步报文的COB-ID具有较高的优先级以及最短的传输时间；

- 时间标记对象（Time Stamp），为各个节点提供公共的时间参考；
- 紧急事件对象（Emergency），当设备内部发生错误触发该对象，即发送设备内部错误代码；
- 节点/寿命保护（Node/Life Guarding），主节点可通过节点保护方式获取从节点的状态。从节点可通过寿命保护方式获取主节点的状态；
- 启动报文对象（Boot-up），从节点初始化完成后向网络中发送该对象，并进入到预操作状态。

#### A.4 CANopen网络配置

在CANopen协议描述文本DS305中定义了一种网络配置协议即网络配置服务 LSS (Layer Setting Service)，其通过CAN总线，用具有LSS 主机功能的CANOpen模块来查询或修改具有LSS 从机的CANOpen模块的某些参数。

通过使用LSS ，可以对下面的参数进行查询或修改：

- CANopen 从站的Node-ID；
- 物理层的位定时参数（波特率）；
- LSS地址（特征对象1018h）。



## 附录 B: GCAN-402 对象字典

索引 (Index)	子索引 (Subindex)	名称 (Name)	类型 (Type)	属性 (Attr.)	默认值 (Deaf.)	描述 (Desc.)
<b>通信参数区</b>						
0x1000	-	Device Type	UINT32	RO	0x00000000	设备类型，高 2 字节为致远电子专用，低 2 字节供用户使用
0x1001		Error Register	UINT8	RO	0	当前错误类型
0x1003	0	number of errors	UINT8	RO	0	-
	1~4	standard error field	UINT32	RO	0	历史紧急错误代码
0x1005	-	COB-ID SYNC	UINT32	RW	0x80	-
0x1007		Sync Windows Length	UINT32	RW	0	-
0x1008		GCAN-402 name	STRING	Const	GCAN-402	GCAN-402 设备名称
0x1009		GCAN-402 hardware version	STRING	Const	V1.01	GCAN-402 硬件版本
0x100A		GCAN-402 software version	STRING	Const	V1.00	GCAN-402 软件版本
0x100C		Guard Time	UINT16	RW	0	-
0x100D		Life Time Factor	UINT8	RW	0	-
0x1010	0	largest supported Sub-Index	UINT8	RO	1	-
	1	save all parameters	UINT32	RW	0	-
0x1011	0	largest supported Sub-Index	UINT8	RO	1	-
	1	restore all default para.	UINT32	RW	0	-
0x1014		COB-ID Emergency message	UINT32	RW	NodeID+0x80	-
0x1016	0	Number Of Entries	UINT8	RO	0x01	-
	1	Consumer Heartbeat Time #1	UINT32	RW	-	-
0x1017		Producer Heartbeat Time	UINT16	RW	0	-
0x1018	0	number of Entries	UINT8	RO	0x04	-
	2	Product code	UINT32	RO	-	GCAN-402产品代码
	3	Revision number	UINT32	RO	-	GCAN-402修订码
	4	Serial number	UINT32	RO	-	GCAN-402序列码

### RPDO 通信参数

0x1400	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x200	RPDO所使用的COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x1401	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x300	RPDO所使用的COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x1402	0	largest subindex supported	UINT8	RO	2	-
	1	COB-Id used	UINT32	RW	NodeID+0x400	RPDO所使用的COB-ID
	2	transmission type	UINT8	RW	0xFE	-
0x1403	0	largest subindex supported	UINT8	RO	2	RPDO所使用的COB-ID
	1	COB-Id used	UINT32	RW	NodeID+0x500	
	2	transmission type	UINT8	RW	0xFE	-

### TPDO 通信参数

0x1800	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x180	TPDO所使用的COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输PDO禁止时间
	5	event timer	UINT16	RW	0	传输PDO定时时间
0x1801	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x280	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1802	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x380	TPDO 所使用的 COB-ID

	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间
0x1803	0	largest subindex supported	UINT8	RO	0x05	-
	1	COB-ID used	UINT32	RW	NODEID+0x8000480	TPDO 所使用的 COB-ID
	2	transmission type	UINT8	RW	0xFE	传输类型
	3	inhibit time	UINT16	RW	0	传输 PDO 禁止时间
	5	event timer	UINT16	RW	0	传输 PDO 定时时间

### 自定义

0x2000	0	Device ID	UINT32	RW		
0x2001	0	data	STRING	Const		
0x2002	0	data	STRING	Const		
0x2003	0	data	STRING	Const		
0x2004	0	data	STRING	Const		
0x2005	0	data	STRING	Const		
0x2006	0	data	STRING	Const		
0x2007	0	data	STRING	Const		
0x2008	0	Baudrate	UINT32	RW		

### 设备状态

0x2400		Actual NodeId	UINT8		-	模块当前的 NodeID
0x2401		Actual BaudRate	UINT8		-	模块当前的波特率索引值
0x2402		ModleStatus	UINT8		-	模块当前的模块状态
0x2403	0	Entries Number	UINT8	CONST	3	-
	1	Set NodeId	UINT8	RO	-	用户通过 uart 口设置的 NodeID, 可能与当前 NodeID 不同
	2	Set Index of Baudrate	UINT8	RO	-	用户通过 uart 口设置的波特率索引值, 可能与当前波特率索引不同
	3	Set Baudrate Value	UINT32	RO	-	用户通过 uart 口设置的 CAN 定时参数, 当该值有效, 模块就会使用该值初始化 CAN 控制器
0x2404	0	Number Entries	UINT8	RO	6	-
	1	Device Hardware version	UINT32	RO	-	用户设备的硬件版本
	2	Device Software Version	UINT32	RO	-	用户设备软件版本
	3	Device Product	UINT32	RO	-	用户设备产品代码

		Code				
	4	Device Revision Num	UINT32	RO	-	用户设备修订码
	5	Device SN Num..	UINT32	RO	-	用设备序列号
	6	Device Name	STRING	RO	-	用户设备名称
0x2405	0	UartComm. Baudrate	UINT8	RO	0x07	通信串口波特率

## 销售与服务

沈阳广成科技有限公司

地址：辽宁省沈阳市皇姑区崇山中路 42 号工业设计中心

邮编：110000

电话：024-31230060

网址：[www.gcgd.net](http://www.gcgd.net)

全国销售与服务电话：400-6655-220

售后服务电话与微信号：13840170070



全国服务电话：400-6655-220