

# GCAN-305

嵌入式CANopen（从站）转UART

## 用户手册



文档版本：V3.20 （2018/7/18）

## 修订历史

版本	日期	原因
V1.00	2013/6/16	创建文档
V2.01	2013/12/20	修正设备工作参数
V3.01	2014/10/22	添加部分参数
V3.02	2015/08/11	修正部分参数
V3.20	2018/07/18	调整文档结构

# 目 录

1. 功能简介.....	4
1.1 功能概述.....	4
1.2 性能特点.....	4
1.3 静态参数.....	5
1.4 典型应用.....	5
2. 设备安装.....	6
2.1 设备尺寸.....	6
2.2 系统连接结构.....	6
2.3 接口定义及功能.....	7
3. 设备使用.....	8
3.1 GCAN-305 配套评估板.....	8
3.3.1 接口定义及功能.....	8
3.2 模块状态转换.....	9
3.3 系统状态指示灯.....	10
3.4 GCAN-305 节点 ID 与 CAN 总线波特率.....	11
4. GCAN-305 中使用 CANopen 协议 (DS301).....	13
4.1 GCAN-305 预定义连接.....	13
4.2 GCAN-305 操作□.....	13
5. GCAN-305 串口操作.....	24
5.1 串口通信协议.....	24
5.2 GCAN-305 操作命令□.....	25
5.3 GCAN-305 串口操作错误响应□.....	34
附录 A: CANopen 协议简介.....	36
附录 B: 串口可操作对象字典列表.....	40
附录 C: GCAN-305 对象字典.....	41
销售与服务.....	44

# 1. 功能简介

## 1.1 功能概述

GCAN-305 是广成科技有限公司开发的嵌入式 CANopen 从站通信转换模块。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-305 模块提供一路 CAN 接口，通过该接口可以与 CANopen 网络进行连接。另外 GCAN-305 提供两个 UART 接口——一个通信 UART 接口（通信波特率为 1200~115200bps），一个调试 UART 接口（固定波特率为 115200bps）。

GCAN-305 模块适用于任何具有串口通信能力的系统，通过搭载该模块，用户现有的串口通信设备可以以最快的速度拥有 CANopen 通信能力，抢占市场先机。用户仅需使用简单的串口通信协议即可实现与 GCAN-305 的通信，完成模块的配置和与 CANopen 总线的数据交换。

模块可以支持多达 12 个 RPDO 和 TPDO 过程数据传输，适用于各种干扰强、实时性要求高的场合，极其小巧的体积、高速实时的处理能力和灵活多样的应用，使 GCAN-305 模块非常易于嵌入到用户设备中，让用户的设备即刻拥有强大的 CANopen 通信功能。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-305 模块可远程升级内核程序，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;
- 串口波特率 1200~115200bps, 可定制 230400bps;
- 串口通信能力(Uart, TTL 电平);
- 各 96 字节的输入输出数据缓冲 (I/O);
- 支持拨码开关设置从站号 (1~127) 及 CAN 波特率 (20kbps、50kbps、100kbps、125kbps、250kbps、500kbps、800kbps、1000kbps);
- CAN->UART 转换最大时间 1ms;
- UART->CAN 转换最大时间 2ms;
- 工作电压/电流: +5V/80mA;
- IO 口电压: 3.3V;
- 体积 32mm×20.4mm×11mm (DIP24 封装);

- 工作温度和存储温度：-40℃~85℃。

### 1.3 静态参数

GCAN-305 模块的静态参数如表 1.1 所示，各项参数均在室温下测得。

符号	定义	测试条件	最小	最大	单位
<b>电源</b>					
V <sub>DD</sub>	供电电压		4.5	6	V
I <sub>DD</sub>	供电电流	所有 I/O 悬空,输入电压 5V	65	85	mA
<b>I/O 端口</b>					
V <sub>IL</sub>	低电平输入电压	V <sub>DD</sub> = 5V	-	0.8	V
V <sub>IH</sub>	高电平输入电压	V <sub>DD</sub> = 5V	2.0	5.5	V
V <sub>OL</sub>	低电平输出电压	I <sub>OL</sub> = -4mA	-	0.4	V
V <sub>OH</sub>	高电平输出电压	I <sub>OH</sub> = -4mA	2.6	3.3	V
I <sub>OL</sub>	低电平输出电流	V <sub>OL</sub> = 0.4V	-	4	mA
I <sub>OH</sub>	高电平输出电流	2.6V ≤ V <sub>OH</sub> ≤ V <sub>DD</sub>	-	-4	mA
I <sub>OL</sub>	低电平短路电流	2.6V ≤ V <sub>OL</sub> ≤ 3.3V	-	50	mA
I <sub>OH</sub>	高电平短路电流	V <sub>OH</sub> = 0V	-	-45	mA

表 1.1 GCAN-305 模块静态参数

### 1.4 典型应用

- 现有 RS-232 设备连接 CANopen 网络；
- PLC 设备总线改造连接 CANopen 网络通讯；
- CAN-bus 与串行总线之间的网关网桥；
- CANopen 从站开发、学习；
- CANopen 工业自动化控制系统；
- 大量设备使用 CANopen 统一控制。

## 2. 设备安装

本章介绍了当用户使用 GCAN-305 模块嵌入到自己的设备时的连接方法及注意事项。

### 2.1 设备尺寸

设备外形尺寸：(长) 32mm \* (宽) 20.4mm \* (高) 11mm，相邻引脚间距为 25.4mm 其顶视图如图 2.1 所示。

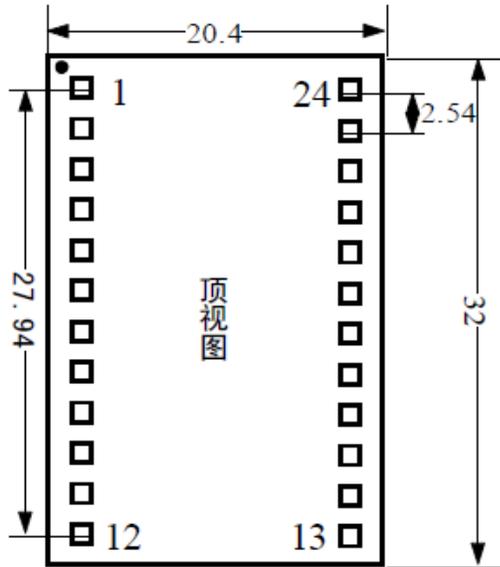


图2.1 GCAN-305模块顶视图

### 2.2 系统连接结构

GCAN-305 系统结构图如图 2.2 所示，用户仅需将 CPU 的串口（TTL 电平）与模块串口连接，将模块的 CAN 总线接口通过 CAN 收发器接入 CANopen 网络，即可建立用户 CPU 与 CANopen 网络的桥接。用户 CPU 串口发送过来的数据将最终以 PDO 报文方式发送到 CAN 总线，以及读取来自 CAN 总线的 RPDO 数据。当然用户也可以通过串口配置模块相关参数。

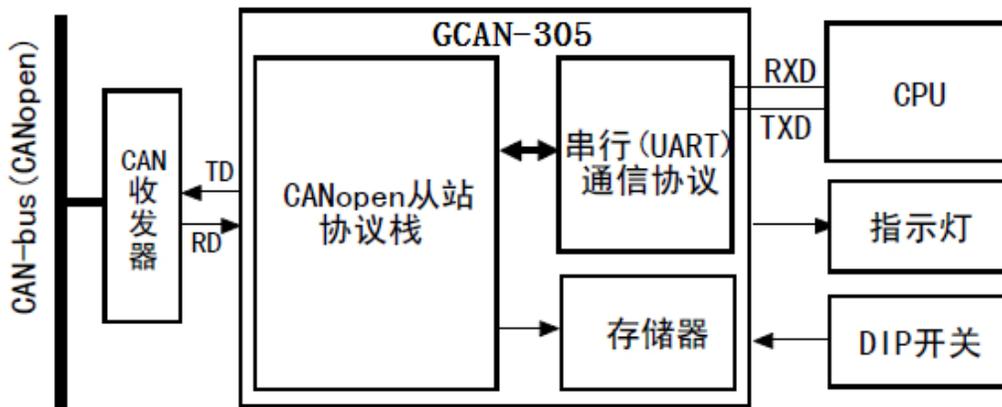


图 2.2 GCAN-305 系统连接结构

GCAN-305 模块可使用 DIP 开关来设置设备的节点号 (Node ID) 和 CAN 通信波特率, 在特殊情况下也可以不使用 DIP 开关来设置, 可通过用户 UART 接口或 CANopen 层设置功能 (LSS) 来设置该模块的节点号和 CAN 通信波特率。

### 2.3 接口定义及功能

引脚号	名称	详细功能	引脚号	名称	详细功能
1	GND	电源地	24	V <sub>CC</sub>	电源输入 (5V)
2	/Rst	复位	23	CAN-R	CAN 接收
3	UART1-T	串口通信发送	22	CAN-T	CAN 发送
4	UART1-R	串口通信接收	21	ID0	节点号输入 0
5	UART0-T	调试/升级串口发送	20	ID1	节点号输入 1
6	UART0-R	调试/升级串口接收	19	ID2	节点号输入 2
7	/INT	中断引脚	18	ID3	节点号输入 3
8	Brt0	波特率设置 0	17	ID4	节点号输入 4
9	Brt0	波特率设置 1	16	ID5	节点号输入 5
10	Brt0	波特率设置 2	15	ID6	节点号输入 6
11	Brt0	波特率设置 3	14	E-Led	错误指示灯 (红)
12	/ISP-EN	升级使能	13	R-Led	运行指示灯 (绿)

表2.1模块引脚定义

由于GCAN-305模块中未集成有CAN收发器, 因此需要外接CAN收发器。UART0串口为GCAN-305模块的调试输出和程序升级接口, 在产品调试阶段建议引出该串口, 方便用户调试自己的程序; 在产品稳定性得到保证的情况下可以不用引出该串口。

典型原理图与PCB设计, 详细请查看GCAN-305技术文档。

## 3. 设备使用

### 3.1 GCAN-305 配套评估板

我公司可为用户提供 GCAN-305 配套的评估、开发、调试、测试底板，如图 3.1 所示。底板已涵盖了所有 GCAN-305 功能且使用简单方便，便于用户对模块进行开发和调试。

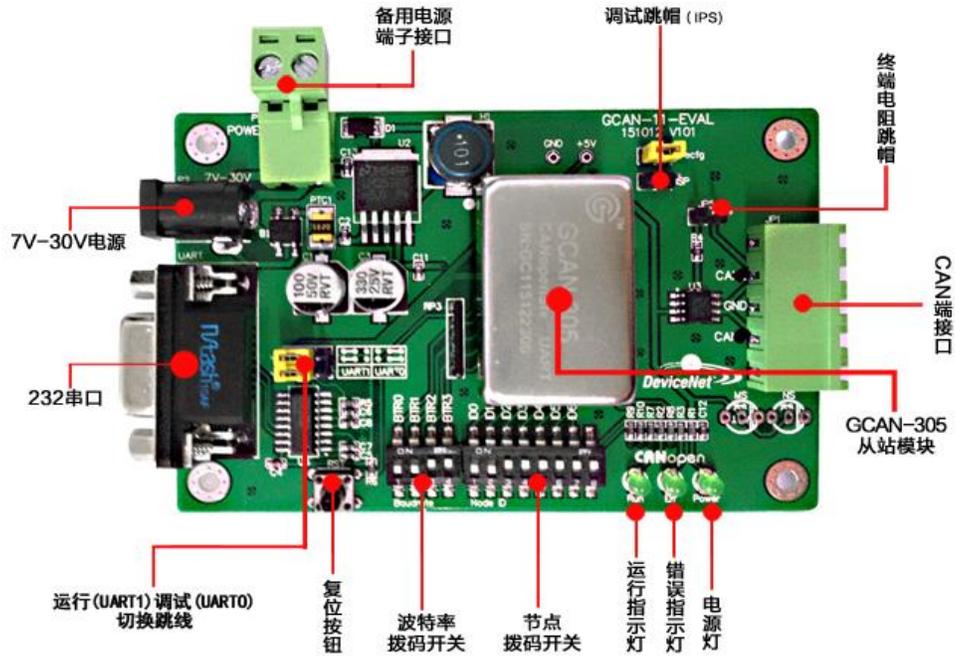


图3.1 GCAN-305模块评估板

#### 3.3.1 接口定义及功能

GCAN-305 模块评估板集成 1 路 DC 7-30V 电源接口、1 路备用电源接口、1 路标准 CAN-Bus 接口、1 路标准 RS232 接口。GCAN-305 模块评估板电源接口、RS232 接口位置下图 3.2 所示：

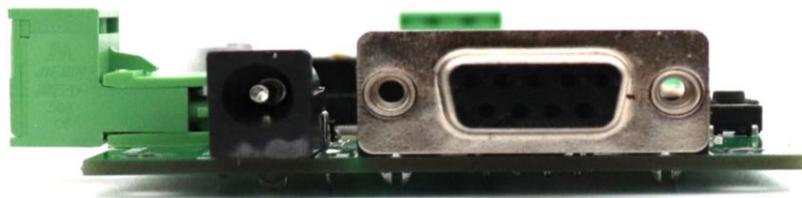


图3.2 GCAN-305模块评估板电源接口、RS232接口位置

GCAN-305 模块评估板 CAN-bus 接口由 1 个 3 Pin 接线端子引出，可以用于连接 1 个 CAN-bus 网络或者 CAN-bus 接口的设备，其接口位置及接口定义如图 3.3、表 3.1 所示。



图 3.3 GCAN-305 模块评估板 CAN-bus 接口

引脚 (由左至右)	端口	名称	功能
1	CAN	CAN-L	CAN_L 信号线 (CAN 低)
2		GND	CAN_GND 接地
3		CAN-H	CAN_H 信号线 (CAN 高)

表 3.1 GCAN-305 模块评估板 CAN-bus 信号分配

### 3.2 模块状态转换

GCAN-305各状态转换图如图3.4所示，图中各字母所表示的各种状态下可进行的的操作，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

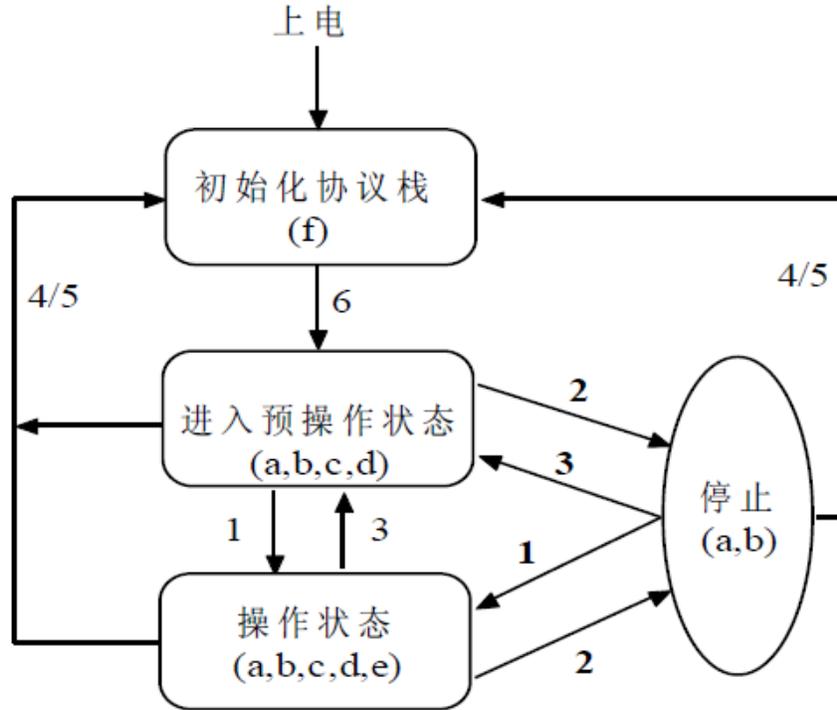


图3.4 GCAN-305各状态转换图

### 3.3 系统状态指示灯

按照CANopen协议规范文档DS303-3的定义，在GCAN-305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3.2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3.2 系统状态指示灯

其状态指示灯所指的各种状态含义如表3.3、表3.4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3.3 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3.4 运行状态指示灯 (RUN) 状态意义

### 3.4 GCAN-305 节点 ID 与 CAN 总线波特率

GCAN-305 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 3.5 和图 3.6 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。

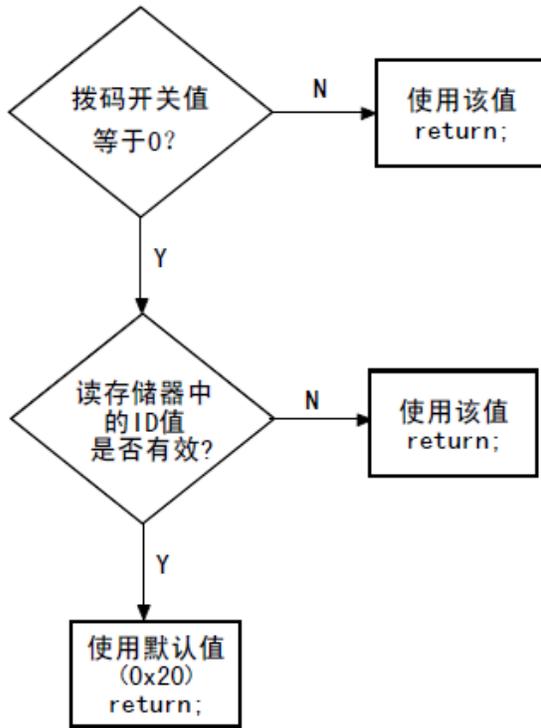


图3.5 节点ID设置顺序

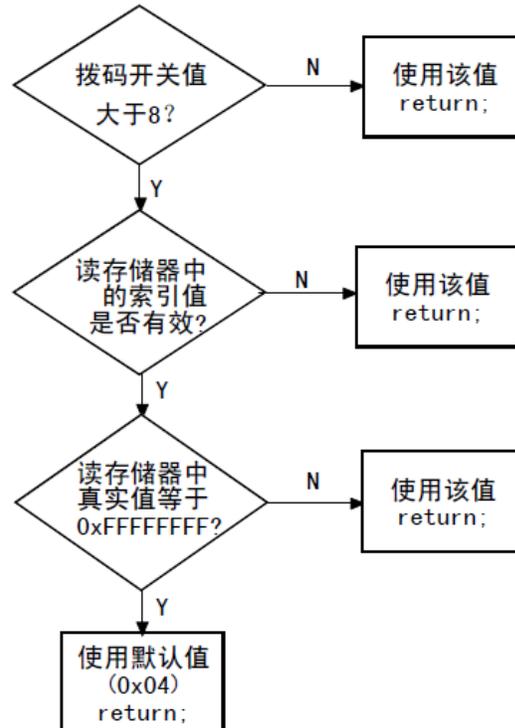
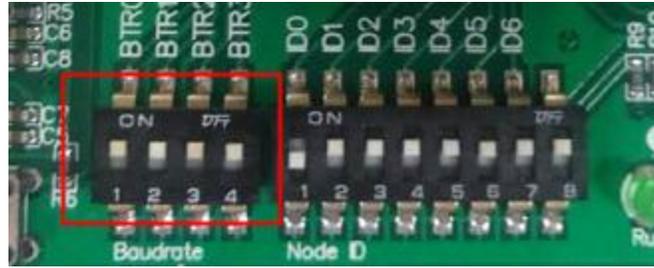


图3.6 波特率设置顺序

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，向上拨动为 0，向下拨动为 1。当波特率拨码开关 4 为 1（即拨码开关 4 向下），此时波特率由软件决定。当波特率拨码开关 4 为 0（即拨码开关 4 向上），波特率值由拨码开关决定，详情参见表 3.5。



图示	波特率
	1000k
	800k
	500k
	250k
	125k
	100k
	50k
	20k

表3.5 波特率索引值与实际对照关系

## 4. GCAN-305 中使用 CANopen 协议 (DS301)

### 4.1 GCAN-305 预定义连接

在 GCAN-305 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(UINT8, UINT16, UINT32 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 4.1 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4.1 CANopen预定义连接集

### 4.2 GCAN-305 操作

#### 4.2.1 网络管理服务 (NMT) □

##### 1. 网络控制 (NMT Module Control) □

GCAN-305 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 4.2 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 4.3 所示。

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

表4.2 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4.3 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 4.4 所示的命令。

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

表4.4 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 4.5 所示。

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

表4.5 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 4.6 所示，从节点应答相应格式如图 4.7 所示。

主节点→从节点(命令):

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

表4.6 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 + Node ID	1	Bit7:触发位, Bit0~Bit6 状态

表4.7 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 4.8 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4.8 节点保护状态值

例：假设主节点需要对节点号为 0x03 从节点进行节点保护，其命令如表 4.9 所示，从节点应答帧如表 4.10 所示。

主节点→从节点：

COB-ID(CAN-ID)	DLC
0x703	1

表4.9 保护节点(远程帧)

从节点→主节点：

COB-ID(CAN-ID)	DLC	BYTE0
0x703	1	0x85

表4.10 从节点(0x03)应答帧

其中 BYTE0 的 Bit7=1，状态=0x05，表示节点号为 0x03 的从站正处于操作状态。

### 3. 寿命保护 (NMT Life Guarding) □

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 4.6），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 4.7 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-305 初始化完成(Boot-up)，就会发送一个标识报文，其报文格式如表 4.11 所示。

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

表4.11 初始化完成标识报文格式

例：假设GCAN-305的节点号为0x03，则发送的启动报文如表4.12所示。

COB-ID(CAN-ID)	DLC	BYTE0
0x703	1	0x00

表4.12 0x703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-305 模块中只支持心跳报文生产，即 GCAN-305 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 4.7 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 0x03，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 4.13 所示的报文。

COB-ID(CAN-ID)	DLC	BYTE0
0x703	1	0x05

表4.13 从节点(0x03)心跳报文

**请注意：**在同一个 GCAN-305 模块中同一时间寿命保护和心跳报文不能同时使用。

### 4.2.2 同步报文对象(SYNC)

同步报文分为消费和生产，在 GCAN-305 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 4.14 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4.14 同步报文帧格式（远程帧）

### 4.2.3 紧急报文对象(EMCY)

在 GCAN-305 中支持紧急报文，即在 GCAN-305 内部出现错误时发送该报文，其报文格式如表 4.15 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 4.16 所示。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1	BYTE2	BYTE3—BYTE7
0x80 + Node ID	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4.15 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4.16 错误寄存器

紧急错误代码含义如表 4.17 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 16temperature
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 exceed
90xx	External error
F0xx	Additional functions
FFxx	Device specific

表4.17 紧急错误代码

例：假设节点地址为0x03，CAN总线错误超过警戒值则出现“ Error Passive”（8120）的警告，则GCAN-305发送紧急报文如表4.18所示。

COB-ID(CAN-ID)	DLC	BYTE0	BYTE1	BYTE2	BYTE3—BYTE7
0x83	8	0x20	0x81	0x11	0x00000000

表4.18 紧急报文（总线错误）

**请注意：**GCAN-305模块发生紧急情况时，这些错误主动发出。

#### 4.2.4 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-305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 + Node ID(发送)和0x600 + Node ID(接收)。SDO分为加速传输、段传输、和块传输。因为在GCAN-305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，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□

加速传输一帧最多只能传输4个字节的数据，报文基本结构如表4.19和表4.20所示，通过SDO的命令字来区分该帧数据的类型。

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x600 + Node ID	8	CMD(SDO 命令字)	对象索引	对象子索引	**

表4.19 SDO报文格式（客户端→服务器）

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
0x580 + Node ID	8	CMD(SDO 命令字)	对象索引	对象子索引	**

表4.20 SDO应答格式（服务器□客户端）

##### 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4.21和表4.22所示。具体的中止错误代码含义可参照表4.23。

COB-ID(CAN-ID)	DLC	Byte 0	Byte 1-2	Byte 3	Byte 4-7
----------------	-----	--------	----------	--------	----------

0x600 + Node ID /0x580+Node ID	8	CMD(SDO 命令字)	索引	子索引	**
-----------------------------------	---	--------------	----	-----	----

表4.21 中止报文格式

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

表4.22 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4.23 中止错误代码表

### 4.2.5 过程数据对象 (PDO) □

过程数据对象 (PDO) 用作传输实时数据, PDO的接收者可以是主节点也可以是其它的从节点, 且不需要应答。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-305出厂时已经为每个PDO预定义了映射对象, 如图4.1所示, 全部RPDO的数据映射项都默认连接到GCAN-305模块的8 bit 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-305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。

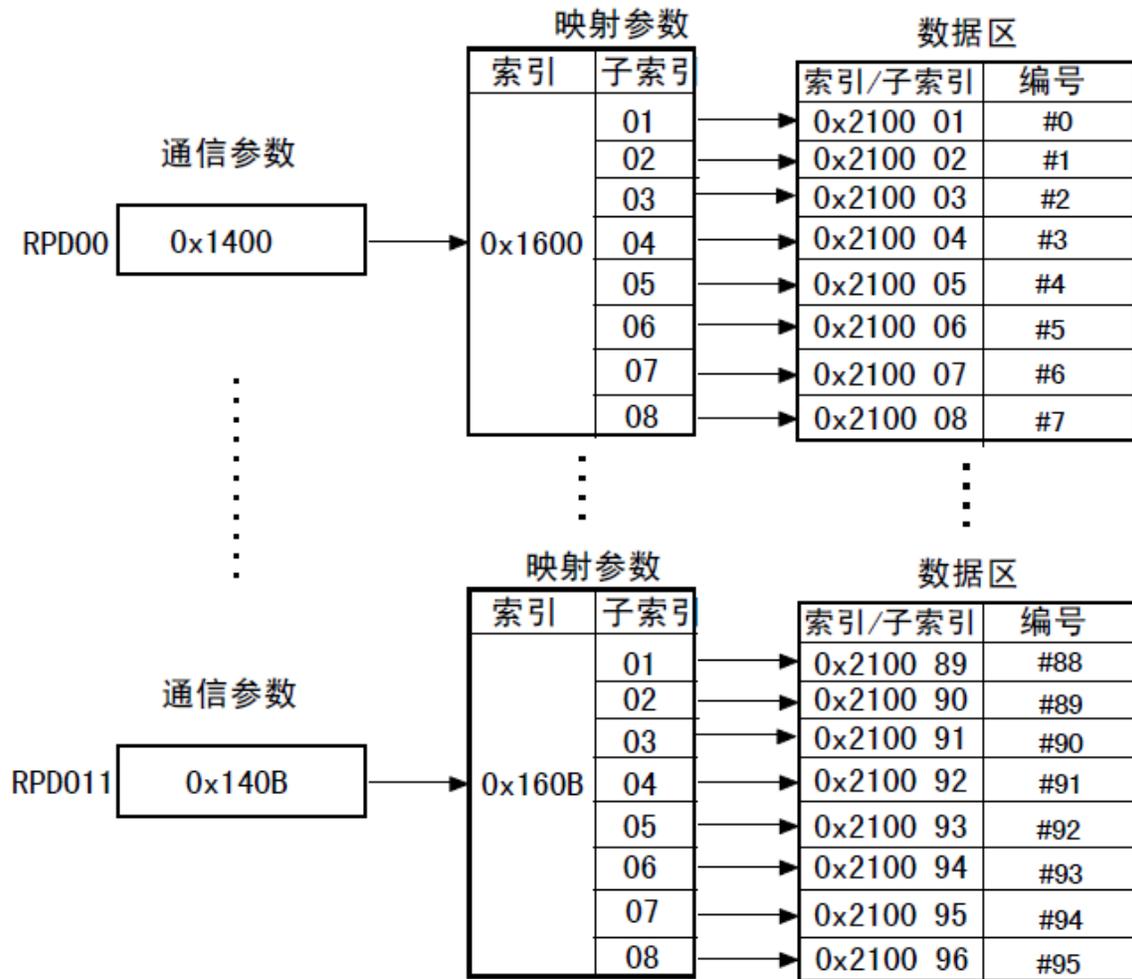


图4.1 RPDO映射关系

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4.24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4.24 RPDO的传输类型

**例：**假设GCAN-305节点为0x03，采用预定义连接，则RPDO0的COB-ID为0x203。则其接收的TPDO COB-ID也应为0x203如表4.25所示，该TPDO正好与节点Node ID为0x03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4.26所示。

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x203	8	0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88

表4.25 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4.26 数据区数据

## 2. 过程数据发送（TPDO） □

在GCAN-305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4.2所示。

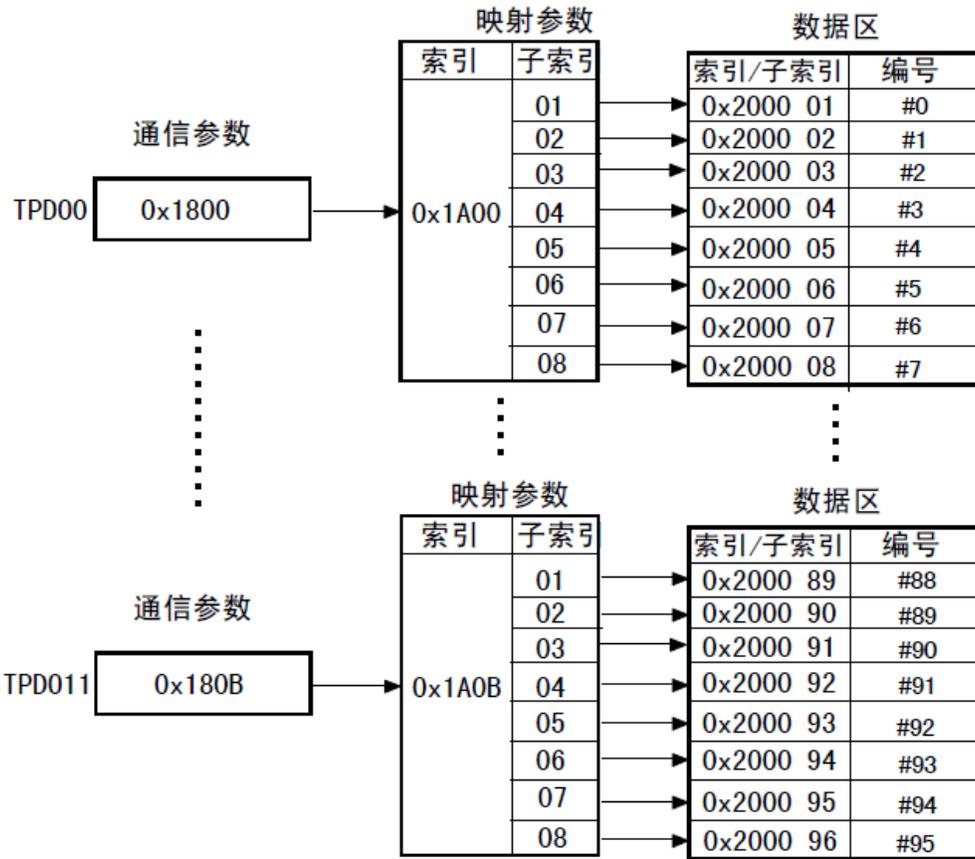


图4.2 TPDO 映射关系

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4.27 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Node ID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4.3所示。

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 4.3 TPDO 发送出的数据

## 5. GCAN-305 串口操作

### 5.1 串口通信协议

GCAN-305与用户通信采用异步串口进行通信，通信模式为半双工，通信信号为TTL电平，通信协议采用自定义串行通信协议。

#### 异步串口数据帧格式

每1个字节用10 bits传送，1个起始位、8个数据位、无奇偶校验位、1个停止位，波特率1200~115200 bps。应答方式：用户设备主动询问（主），GCAN-305被动回答（从）。主/从应答帧结构实现通常的数据通讯，数据的通讯由主机发起，称为命令帧，其帧格式如表5.1所示；从机接收到后进行应答，称为响应帧，如表5.2所示。

1Byte	1Byte	1Byte	1Byte	nByte	1Byte
起始字 0x7E	命令码 CMD	命令信息(长度) CMDinfo	特定参数 SpeByte	命令数据 DATA	校验码 CRC

表5.1 命令帧格式

1Byte	1Byte	1Byte	1Byte	nByte	1Byte
起始字 0x7E	响应码 ACK	响应信息(长度) ACKinfo	特定参数 SpeByte	响应数据 DATA	校验码 CRC

表5.2 应答帧格式

下面按各命令排序，详细介绍命令帧、响应帧的规则。命令帧、应答帧的总长度为：命令/响应信息的CMDinfo/ ACKinfo（数据长度）+ 5Byte，各字段说明如下：

- 帧起始字符SOF，固定为0x7E，长度为一个字节。
- 命令码CMD/响应码ACK，通常CMD=ACK，长度为一个字节。
- 命令信息CMDinfo/响应信息ACKinfo指出命令信息/响应信息的长度(字节)，不包括本身。CMDinfo/ACKinfo = 0 表示没有数据，CMDinfo/ACKinfo = 1 表示本帧含有1个字节数据。
- 特殊参数SpeByte包括Error、AllDataSegSize、DataSegNum信息，具体含义如表5.3所示。

Error为特定意义位。在命令信息中，Error为保留位，通常Error = 0；在响应帧中，Error为错误标识位，Error = 1，表明命令执行出错，DATA区跟随错误代码，Error = 0，表明请求成功，DATA区跟随应答数据。

AllDataSegSize、DataSegNum指出分段信息，当数据量超过255字节时，需要使用多帧来传送，此时使用分段传输方式，共可以分为7段，即7帧。

BIT.7	BIT.6	BIT.5	BIT.4	BIT.3	BIT.2	BIT.1	BIT.0
Error	AllDataSegSize			保留位	DataSegNum		

表5.3 特定参数（SpeByte）定义

AllDataSegSize表示本次信息数据共有几帧；DataSegNum表示本次发送的是第几帧该信息数据。当AllDataSegSize = DataSegNum时表示该信息数据传输完毕。AllDataSegSize和DataSegNum最小值为1，不能为0。

本协议将不使用分段，AllDataSegSize、DataSegNum 固定为“1”。

- 命令/响应数据DATA，此部分与命令/响应码相结合，描述数据的具体含义。长度在CMDInfo/ ACKInfo中说明，最大为255个字节/帧（仅数据区字节数）。
- 校验和CRC：命令/响应数据的校验和，长度为一个字节。校验和为前面所有数据的异或值。CRC的计算公式如下：

$$\text{CRC} = 0x7E \wedge \text{CMD} \wedge \text{CMDInfo} \wedge \text{SpeByte} \wedge \text{DATA}[0] \wedge \text{DATA}[1] \wedge \dots \wedge \text{DATA}[n-1]$$

或  $\text{CRC} = 0x7E \wedge \text{ACK} \wedge \text{ACKInfo} \wedge \text{SpeByte} \wedge \text{DATA}[0] \wedge \text{DATA}[1] \wedge \dots \wedge \text{DATA}[n-1]$

## 5.2 GCAN-305 操作命令

用户通过GCAN-305的通信串口对设备进行操作，操作命令如5.2.1节所示，本节所有的操作命令都认为成功执行，并得到正确执行的命。如果出现错误，其错误响应帧和错误代码将在5.3节中介绍。

### 5.2.1 读取设备信息(命令码：0x01)

通过串口用户可以读取用户设置的设备信息，该设备信息是用户设置的信息，位于对象字典的0x2404。其中设备类型位于对象字典0x1000 00，因为该设备是通用设备，没有使用标准设备描述，因此该参数按照CiA定义应该为0x00000000，建议用户不要更改此参数。操作命令及响应帧如表5.4和表5.5所示，n表示该命令帧数据长度。其中命令数据域的第一字节为操作模式，根据该字节判断该命令所读取的信息类别，以下将详细介绍操作命令。

**请注意：**通过串口读取的设备信息并不是GCAN-305设备信息，而是用户设备相关的设备信息。

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x01	n	0x11	DAT	校验码

表5.4 读取设备信息命令□

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x01	n	0x11/0x91	DAT	校验码

表5.5 读取设备信息响应□

#### 1. 读取设备类型(操作模式：0x01)

设备类型位于对象字典的0x1000 00，数据长度4字节，高2字节为GCAN-305设备类型，低2字节为用户设备类型代码（用户设置）。

**例：**如下命令读回设备类型为0x0A000011，即当前的的GCAN-305的设备类型为0x00A0，用户的设备类型为0x0000。

命令：0x7E 0x01 0x01 0x11 **0x01** 0x6E

响应：0x7E 0x01 0x05 0x11 **0x01 0x11 0x00 0x0A 0x00** 0x71

**请注意：**使用该模块建议不要更改设备类型，因为GCAN-305没有使用标准设备描述，因此设备类型为通常为0x00000000。

#### 2. 读硬件版本(操作模式:0x02)

硬件版本信息位于对象字典0x2404 01，数据长度为4字节，该值为用户设置信

息。

**例：**如下命令读取回来的硬件版本信息为0x00000100。

命令：0x7E 0x01 0x01 0x11 **0x02** 0x6D

响应：0x7E 0x01 0x05 0x11 **0x02 0x00 0x01 0x00 0x00** 0x68

### 3. 读软件版本（操作模式：**0x03**）□

软件版本信息位于对象字典的0x2404 02，数据长度4字节，该值为用户设置信息。

**例：**如下命令读取回来的软件版本信息为0x00000100。

命令：0x7E 0x01 0x01 0x11 **0x03** 0x6C

响应：0x7E 0x01 0x05 0x11 **0x03 0x00 0x01 0x00 0x00** 0x69

### 4. 读产品代码（操作模式：**0x04**）□

产品代码位于对象字典的0x2404 03，数据长度4字节，该值为用户设置信息。

**例：**如下命令读取回来的产品代码为0x00000001。

命令：0x7E 0x01 0x01 0x11 **0x04** 0x6B

响应：0x7E 0x01 0x05 0x11 **0x04 0x01 0x00 0x00 0x00** 0x6E

### 5. 读产品修订码（操作模式：**0x05**）□

产品代码位于对象字典的0x2404 04，数据长度4字节，该值为用户设置信息。

**例：**如下命令读取回来的产品修订码为0x00000001。

命令：0x7E 0x01 0x01 0x11 **0x05** 0x6A

响应：0x7E 0x01 0x05 0x11 **0x05 0x01 0x00 0x00 0x00** 0x6F

### 6. 读产品序列号（操作模式：**0x06**）□

产品代码位于对象字典的0x2404 05，数据长度4字节，该值为用户设置信息。

**例：**如下命令读取回来的产品序列号为0x00000001。

命令：0x7E 0x01 0x01 0x11 **0x06** 0x69

响应：0x7E 0x01 0x05 0x11 **0x06 0x01 0x00 0x00 0x00** 0x6C

### 7. 读产品名称（操作模式：**0x07**）□

产品代码位于对象字典的0x2404 06，数据长度12字节(字符串，前11字节有效)，该值为用户设置信息。

**例：**如下命令读取回来的产品名称为“DeviceName”。

命令：0x7E 0x01 0x01 0x11 0x07 0x68

响应：0x7E 0x01 0x0D 0x11 0x07 0x44 0x65 0x76 0x69 0x63 0x65 0x4E 0x61  
0x6D 0x65 0x00 0x84 0xFF

## 5.2.2 写设备信息（命令码：**0x02**）□

写设备信息与读设备信息相对应，相同的信息都是对应于相同的对象字典。其中设备类型位于对象字典0x1000 00，因为该设备是通用设备，没有使用标准设备描述，因此该参数按照CiA定义应该为0x00000000，建议用户不要更改此参数。其操作命令及响应帧如表5.6和表5.7所示，其n表示该帧命令数据长度。其中命令数据的第一字节为操作模式，根据该字节判断该命令所读取的信息类别，以下将详细介绍操作命令。

**请注意：**通过串口写入的设备信息并不能改变GCAN-305设备信息，而是更改

用户设备相关的设备信息。

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x02	n	0x11	DAT	校验码

表5.6 写入设备信息命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x02	n	0x11	DAT	校验码

表5.7 写入设备信息响应

### 1. 写设备类型(操作模式: 0x01)

设备类型位于对象字典的0x1000 00, 数据长度4字节, 高2字节为GCAN-305设备类型, 低2字节为用户设备类型代码(用户设置)。

例: 写入设备类型数值为0x0011。

命令: 0x7E 0x02 0x05 0x11 **0x01 0x11 0x00 0x00 0x00 0x69**

响应: 0x7E 0x02 0x01 0x11 **0x01 0x6D**

### 2. 写硬件版本(操作模式:0x02)

硬件版本信息位于对象字典0x2404 01, 数据长度为4字节, 该值为用户设置信息。

例: 如下命令写入硬件版本为0x22222222。

命令: 0x7E 0x02 0x05 0x11 **0x02 0x22 0x22 0x22 0x22 0x6A**

响应: 0x7E 0x02 0x01 0x11 **0x02 0x6E**

### 3. 写软件版本(操作模式: 0x03) □

软件版本信息位于对象字典的0x2404 02, 数据长度4字节, 该值为用户设置信息。

例: 如下命令写入软件版本信息为0x33333333。

命令: 0x7E 0x02 0x05 0x11 **0x03 0x33 0x33 0x33 0x33 0x6B**

响应: 0x7E 0x02 0x01 0x11 **0x03 0x6F**

### 4. 写产品代码(操作模式: 0x04) □

产品代码位于对象字典的0x2404 03, 数据长度4字节, 该值为用户设置信息。

例: 如下命令写入产品代码为0x44444444。

命令: 0x7E 0x02 0x05 0x11 **0x04 0x44 0x44 0x44 0x44 0x6C**

响应: 0x7E 0x02 0x01 0x11 **0x04 0x68**

### 5. 写产品修订码(操作模式: 0x05) □

产品代码位于对象字典的0x2404 04, 数据长度4字节, 该值为用户设置信息。

例: 如下命令写入产品修订码为0x55555555。

命令: 0x7E 0x02 0x05 0x11 **0x05 0x55 0x55 0x55 0x55 0x6D**

响应: 0x7E 0x02 0x01 0x11 **0x05 0x69**

### 6. 写产品序列号(操作模式: 0x06) □

产品代码位于对象字典的0x2404 05, 数据长度4字节, 该值为用户设置信息。

例: 如下命令写入产品序列号为0x66666666。

命令: 0x7E 0x02 0x05 0x11 **0x06 0x66 0x66 0x66 0x66 0x6E**

响应: 0x7E 0x02 0x01 0x11 0x06 0x6A

## 7. 写产品名称 (操作模式: 0x07) □

产品代码位于对象字典的0x2404 06, 数据长度12字节(字符串, 前10字节有效), 该值为用户设置信息。

### 5.2.3 写 GCAN-305 输入缓冲区数据 (命令码: 0x10) □

GCAN-305的输入缓冲区从编号#0~#95宽度为8bit。对应输入缓冲区16bit的#0~#47和32bit的#0~#23, 它们占有相同的内存区域。如果需要对32bit区的#0号区域进行操作, 可操作8bit区的#0~#3以及对16bit区的#0~#1操作可得相同的结果, 以此类推, 因此在串口操作中只提供对8bit区域的操作。

GCAN-305输入数据缓冲区只是一个只写区域, 最多一次可写96字节的数据。操作命令和响应命令如表5.8和表5.9所示, 在命令数据的第1字节表示当前数据在输入数据区的偏移量, 所写的数据长度为n-1。

**请注意:** 偏移量+数据长度不应大于96。

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)		CRC
0x7E	0x10	n	0x11	地址偏移	DAT	校验码

表5.8 写缓冲区数据命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x10	n	0x11	DAT	校验码

表5.9 写缓冲区数据响应

**例:** 向8bit区编号#0开始的地址(即偏移量为0)写入8字节的数据, 数据为0x12 0x12 0x12 0x12 0x12 0x12 0x12 0x12。

命令: 0x7E 0x10 0x09 0x11 0x00 0x12 0x12 0x12 0x12 0x12 0x12 0x12 0x12 0x76

响应: 0x7E 0x10 0x01 0x11 0x00 0x7E

### 5.2.4 读取 GCAN-305 输出缓冲区数据 (命令码: 0x11)

GCAN-305的输出缓冲区从编号#0~#95宽度为8bit。对应输出缓冲区16bit的#0~#47和32bit的#0~#23, 它们占有相同的内存区域。如果需要对32bit区的#0号区域进行操作, 可操作8bit区的#0~#3以及对16bit区的#0号进行操作同样对#0~#1操作可得相同的结果, 以此类推, 因此在串口操作中只提供对8bit区域的操作。

GCAN-305输出数据缓冲区是一个只读区域, 用户最多一次可读96字节的数据。操作命令和响应命令如表5.10和表5.11所示, 在命令数据帧的第1字节表示当前数据在输出数据区的偏移量, 第2字节表示需要读出的数据的长度。在响应命令帧数据的第1字节表示在输出缓冲区的偏移量, 返回的数据长度为n-1。

**请注意:** 偏移量+数据长度不应大于96。

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)		CRC
0x7E	0x11	n	0x11	地址偏移	DAT	校验码

表5.10 读缓冲区数据命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)		CRC
0x7E	0x11	n	0x11	地址偏移	DAT	校验码

表5.11 读缓冲区数据响应

**例：** 读取输出缓冲区偏移量从#0开始的8字节数据，所读取的数据为0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

命令： 0x7E 0x11 0x02 0x11 **0x00 0x08** 0x74

响应： 0x7E 0x11 0x09 0x11 **0x00 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88** 0xFF

### 5.2.5 读写 GCAN-305 的 NodeID（命令码： 0x12）

在GCAN-305未使用DIP开关或其它方式对模块设置NodeID时，用户可以通过串口设定GCAN-305的NodeID，设置的NodeID被保存在存储器中，模块复位后生效（设置值必须在1~127之间）。

**请注意：** 仅在未使用DIP开关设定NodeID或设定无效（0x00）的情况下，通过串口设置的值才生效。

#### 1. 写GCAN-305的NodeID(操作模式： 0x00)

节点NodeID位于对象字典0x2403 01，表示当前模块设定的NodeID值，设定后复位或重启模块生效。写模块的NodeID命令格式与应答命令如表5.12和表5.13所示，命令数据的第1字节表示当前的操作模式，第2字节表示将要写入的NodeID值。写GCAN-305的NodeID如果成功则返回正确应答（n=0x01，命令字节第1字节为操作模式）。

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x12	2	0x11	操作模式(0x00)	DAT(NodeID)	校验码

表5.12 写节点NodeID命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x12	n	0x11	DAT	校验码

表5.13 写节点NodeID命令响应

**例：** 设置GCAN-305模块的NodeID为0x03，其命令与响应如下所示。

命令： 0x7E 0x12 0x02 0x11 **0x00 0x03** 0x5F

响应： 0x7E 0x12 0x01 0x11 **0x00** 0x7C

#### 2. 读GCAN-305的NodeID(操作模式： 0x01)

读节点NodeID所读取回来的NodeID是当前节点正使用的NodeID，与写NodeID位于不同的对象字典。读NodeID位于对象字典的0x2400 00，因此写入的NodeID与读回的NodeID可能会不同。其命令模式如表5.14和表5.15所示。

起始字节	命令码	命令信息	特定参数	命令数据(1 Byte)	CRC
0x7E	0x12	0x01	0x11	操作模式(0x01)	校验码

表5.14 读节点NodeID命令

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x12	2	0x11	操作模式(0x00)	DAT(NodeID)	校验码

表5.15 读节点NodeID命令响应

**请注意：**读回的NodeID值为当前模块正使用的NodeID，并不一定与写入的NodeID相同。

**例：**读取当前GCAN-305的NodeID，读回的NodeID值为0x03。

命令：0x7E 0x12 0x01 0x11 **0x01** 0x7D

响应：0x7E 0x12 0x02 0x11 **0x01 0x03** 0x5E

### 5.2.6 读写 GCAN-305 的波特率索引值(命令码：0x13)

与设置GCAN-305的NodeID相类似，在未使用DIP开关或其它方式设置CAN通信波特率时，用户可以通过串口来设置GCAN-305的波特率索引值，用户所设置的波特率索引值被保存在存储器中，只有GCAN-305上电或复位后才能生效，该值的范围为0~8，其它的值均无效。

#### 1. 写GCAN-305波特率索引值(操作模式：0x00)

波特率索引值位于对象字典的0x2403 02，设定值在0~8之间有效，否则返回错误代码。其命令格式如表5.16和表5.17所示，当设置命令执行正确时（n=0x01，命令字节第1字节为操作模式）。波特率索引值与真实值之间的对应关系如表3.4所示。

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x13	2	0x11	操作模式(0x00)	DAT(波特率索引)	校验码

表5.16 写节点波特率索引值命令

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x13	n	0x11	DAT	校验码

表5.17 写节点波特率索引值命令响应

**例：**设置GCAN-305模块的波特率索引值为0x04，根据表3.4可知，设置CAN波特率索引为0x04，则CAN通信波特率为125Kbps，命令与响应帧如下所示。

命令：0x7E 0x13 0x02 0x11 **0x00 0x04** 0x7A

响应：0x7E 0x13 0x01 0x11 **0x00** 0x7D

#### 2. 读GCAN-305波特率索引值(操作模式：0x01)

读取的波特率索引位于对象字典的0x2401 00。与读NodeID相类似，读回的波特率索引值并不一定与写入的值相同，因为读回的波特率索引值为当前模块正在使用的值，只有在写入的值被启用后才会相同（模块复位或重启）。其命令帧

格式如表5.18和表5.19所示。

起始字节	命令码	命令信息	特定参数	命令数据(n Byte)	CRC
0x7E	0x13	1	0x11	操作模式(0x01)	校验码

表5.18 读节点波特率索引值命令

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x13	2	0x11	操作模式(0x01)	DAT(波特率索引)	校验码

表5.19 读节点波特率索引值命令响应

**例：**假设当前模块的波特率为125Kbps，则读回的索引值为0x04，其命令与响应如下所示。

命令：0x7E 0x13 0x01 0x11 **0x01** 0x7C

响应：0x7E 0x13 0x02 0x11 **0x01 0x04** 0x7B

### 5.2.7 读写 GCAN-305 CAN 控制器定时参数（命令码：0x14） □

真实波特率是一个32位的值，包括了CAN控制器位定时以及分频参数等。与波特率索引值不一样，在GCAN-305模块中已经固化了一个标准波特率表，因此可以通过索引的方式来设置CAN通信波特率，如果用户需要用到该表中没有提供的波特率，则用户可以通过命令码0x14来设置任意的波特率，具体特殊波特率的值请与我公司联系。

**请注意：**在DIP开关设置和波特率索引值都无效的情况下，用户所设置的波特率定时参数才有效。

#### 1. 写波特率值(操作模式：0x00)

真实波特率位于对象字典0x2403 03，只有在DIP开关设置和波特率索引值都无效的情况下才会被启用（0x00000000和0xFFFFFFFF无效），其命令格式如表5.20和表5.21所示，当命令执行成功则返回正确应答（n=0x01，命令字节第1字节为操作模式）。

起始字节	命令码	命令信息	特定参数	命令数据(5Byte)		CRC
0x7E	0x14	5	0x11	操作模式(0x00)	DAT(波特率索引)	校验码

表5.20 写节点波特率值命令

起始字节	命令码	命令信息	特定参数	命令数据(nByte)	CRC
0x7E	0x14	n	0x11	DAT	校验码

表5.21 写节点波特率命令响应

**例：**向GCAN-305模块中写入125Kbps的波特率值，其命令与响应如下所示。

命令：0x7E 0x14 0x05 0x11 **0x00 0x0B 0x00 0x2B 0x00** 0x5E

响应：0x7E 0x14 0x01 0x11 **0x00** 0x7A

#### 2. 读波特率值(操作模式：0x01)

读波特率值与写波特率值位相同的对象字典0x2403 03，其命令模式如表5.22和表5.23所示。

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x14	n(0x01)	0x11	DAT(操作模式)	校验码

表5.22 写节点波特率值命令

起始字节	命令码	命令信息	特定参数	命令数据(5Byte)		CRC
0x7E	0x14	5	0x11	操作模式(0x01)	DAT(波特率值)	校验码

表5.23 写节点波特率值命令响应

例：假设当前波特率值为0x2B000B，则读命令和响应命令如所示。

命令：0x7E 0x14 0x01 0x11 **0x01** 0x7B

响应：0x7E 0x14 0x05 0x11 **0x01 0x0B 0x00 0x2B 0x00** 0x5F

### 5.2.8 发送紧急代码（命令码：0x15） □

当用户设备出现某种错误之后，可通过CANopen发送到CAN总线上，通知CANopen主站设备当前设备发生了错误。错误代码由5个字节组成，由用户自定义。其格式如表5.24和表5.25所示，当执行正确则应答正确响应（n=0x01，命令字节第1字节为操作模式）。

起始字节	命令码	命令信息	特定参数	命令数据(6 Byte)		CRC
0x7E	0x15	5	0x11	操作模式（0x00）	DAT(5 字节)	校验码

表5.24 发送紧急错误代码命令

起始字节	命令码	命令信息	特定参数	命令数据(nByte)	CRC
0x7E	0x15	n	0x11	DAT	校验码

表5.25 发送紧急错误代码命令响应

例：假设当前设备发生错误，错误代码定义为0x9988776655，则其命令格式如下所示。

命令：0x7E 0x15 0x06 0x11 **0x00 0x55 0x66 0x77 0x88 0x99** 0x29

响应：0x7E 0x15 0x01 0x11 **0x00** 0x7B

### 5.2.9 读取当前模块状态（命令码：0x16） □

用户通过该命令可以读取当前模块所处的状态，所读取回来的值与当前状态的对应关系如表4.8所示。其命令格式如表5.26和表5.27所示。

起始字节	命令码	命令信息	特定参数	命令数据(1 Byte)	CRC
0x7E	0x15	1	0x11	操作码(0x01)	校验码

表5.26 获取当前GCAN-305状态

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x15	2	0x11	操作码(0x01)	DAT(状态值)	校验码

表5.27 获取当前GCAN-305状态响应

例：假设当前GCAN-305模块处于操作状态，则读取的状态值为0x05。

命令：0x7E 0x16 0x01 0x11 **0x01** 0x79

响应：0x7E 0x16 0x02 0x11 **0x01 0x05** 0x7F

### 5.2.10 启动节点进入操作状态（命令码：0x17）

用户可通过该条命令使CANopen网络中的所有从站设备进入到操作状态，其中包括模块本身也会进入到操作状态。其命令格式如表5.28和表5.29所示。

**请注意：**在网络中有主站管理的情况下，谨慎使用此条命令。

起始字节	命令码	命令信息	特定参数	命令数据(1 Byte)	CRC
0x7E	0x17	1	0x11	操作码(0x00)	校验码

表5.28 使能从站进入操作状态

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x17	1	0x11	操作码(0x00)	校验码

表5.29 使能从站进入操作状态响应

例：使能当前CANopen网络进入到操作状态，其命令与响应如下所示。

命令：0x7E 0x17 0x01 0x11 **0x00** 0x79

响应：0x7E 0x17 0x01 0x11 **0x00** 0x79

### 5.2.11 改变通信串口波特率（命令码：0x18） □

通信串口出厂默认波特率为115200bps，其设置范围如表5.30所示。根据需求更改串口通信波特率，同时波特率索引值被保存直到再次被更改（设定波特率值时需使用当前的波特率进行通信，更改完成之后延迟一段时间再使用新设定的波特率进行通信）。禁止在应用时频繁改变通信波特率，这样可能导致通信不成功或数据丢失。

索引值	波特率值(bps)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

表5.30 串口波特率值及索引

其命令格式如表5.31、表5.32所示。

起始字节	命令码	命令信息	特定参数	命令数据(2 Byte)		CRC
0x7E	0x18	2	0x11	操作码(0x00)	波特率索引值	校验码

表5.31 设置串口波特率命令

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x18	1	0x11	操作码(0x00)	校验码

表5.32 串口通信波特率响应

例：设置当前模块的波特率为115200bps，则索引值为7，则命令帧和响应帧如下所示。

命令：0x7E 0x18 0x02 0x11 0x00 0x07 0x72

响应：0x7E 0x18 0x01 0x11 0x00 0x76

**请注意：**建议更改完成波特率之后延时20ms左右再用新设定的波特率进行通信。

### 5.2.12 网络时间标识对象消费 Time\_Stamp (命令码：0x19)

在 CANopen 网络中为了同步所有节点的时间，CANopen 主机向网络中发送同步时间标识对象。GCAN-305 作为一个标准的 CANopen 从站模块，能够及时的获取到网络中的同步时间，使用者可以通过通信串口及时的获取到该网络时间，该时间格式遵循 DS301 V4.02 的定义。

**请注意：**只有在 GCAN-305 的网络时间被更新后才能读出正确的网络时间，否则返回错误代码（数据未更新的错误代码），并且该时间只能被读出一次，其后该值无效并返回错误代码。

读取时间的命令格式如表5.33和表5.34所示。

起始字节	命令码	命令信息	特定参数	命令数据(1Byte)	CRC
0x7E	0x19	1	0x11	操作码(0x01)	校验码

表5.33 读取当前网络时间命令

起始字节	命令码	命令信息	特定参数	命令数据(7Byte)		CRC
0x7E	0x19	7	0x11	操作码(0x01)	TimeOfDay	校验码

表5.34 读取当前网络时间响应

例：在读取数据时当前模块有网络时间已经被更新，该时间为以1984年1月1日为起点的相对时间。该示例返回的时间相对于1984年1月1日为0x2552天0x0237777C毫秒，可算出当前时间为2010年2月27日，10时19分49秒。

命令：0x7E 0x19 0x01 0x11 0x01 0x76

响应：0x7E 0x19 0x07 0x11 0x01 0x7C 0x77 0x37 0x02 0x52 0x25 0x39

当未接收到网络时间或没有更新网络时间时，其响应命令码如下所示。

错误响应：0x7E 0x19 0x02 0x91 0x01 0x08 0xFD（详细说明请见章节5.3）

## 5.3 GCAN-305 串口操作错误响应

在所有的串口操作命令中，通信过程中当命令的参数不正确或其它错误发生

时，GCAN-305都会返回错误代码（特定参数的最高位为1，表示当前为错误应答帧），错误代码帧格式如表5.35所示。其中ACK与操作的命令码相同，应答操作模式码与当前命令操作模式码相同。错误代码表示当前的操作所出现错误类别，错误代码表如表5.36所示。

起始字	响应码	响应信息(长度)	特定参数	响应数据（2字节）		校验码
0x7E	ACK=CMD	0x02	0x91	操作模式	错误代码	CRC

表5.35 GCAN-305命令执行错误响应

错误代码(数据)	说明	备注
0x01	命令错误	不支持该命令
0x02	数据长度错误	超出可写区域
0x03	地址错误	超出地址范围
0x04	在操作协议栈时出现错误	-
0x05	存储数据出错	-
0x06	数据值超出范围	-
0x07	操作模式不支持	-
0x08	网络时间未更新，当前不可用	-

表5.36 错误代码表

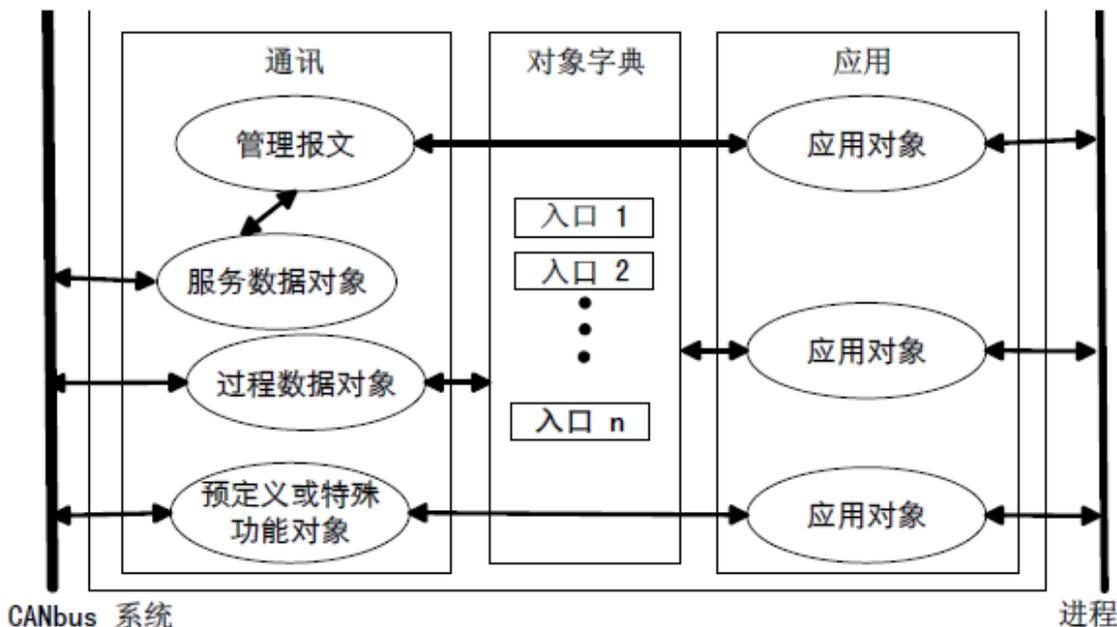
**例：**假设现在写GCAN-305的波特率索引值为10，因为波特率索引值的范围为0~8。所以该值超出范围无效，必然在执行过程中出错，错误代码为0x06(数据值超出范围)，命令帧与响应帧如下所示。

命令：0x7E 0x13 0x02 0x11 **0x00 0x10** 0x6E

响应：0x7E 0x13 0x02 0x91 **0x00 0x06** 0xF8

## 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。

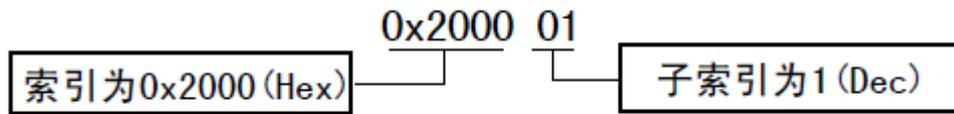
### A.1 相关名词解释和书写规则

#### 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。

#### 3. 设备子协议(Device Profile)

设备子协议，为各种不同类型的设备定义对象字典中的对象。目前已有十几种为不同类型的设备定义的子协议，例如DS401、DS402、DS406 等，其索引值范围为0x6000~0x9FFF。

### 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

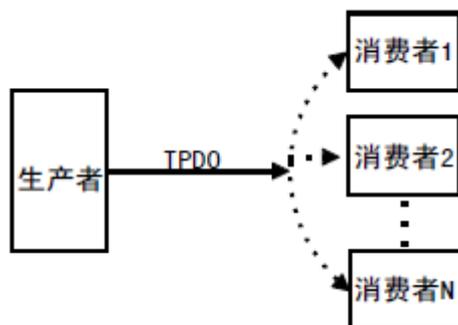
始化、运行、监控、复位和停止。所有节点都被认为是NMT从站。

## 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：串口可操作对象字典列表

操作名称	对应对象字典索引/子索引	备注
读写设备类型	0x1000 00	设备类型, (建议不用更改, 非标准设备)
硬件版本	0x2404 01	长度 4 字节
软件版本	0x2404 02	长度 4 字节
产品代码	0x2404 03	长度 4 字节
修订码	0x2404 04	长度 4 字节
产品代码	0x2404 05	长度 4 字节
设备名称	0x2404 06	GCAN-305/用户可更改(字符串, 11 字节长度),
数据输入缓冲区	0x2000 00~0x2000 60(8bit) 0x2010 00~0x2010 30(16bit) 0x2020 00~0x2020 18(32bit)	该区域对应于 CANopen 的输入缓冲区, 其中 8bit、16bit、32bit 占据相同的内存区
数据输出缓冲区	0x2100 00~0x2100 60(8bit) 0x2110 00~0x2110 30(16bit) 0x2120 00~0x2120 18(32bit)	该区域对应于 CANopen 的输出缓冲区, 其中 8bit、16bit、32bit 占有相同的内存区
当前 CANopen 从节点号	0x2400 00	当前值
当前 CANopen 波特率	0x2401 00	当前值
模块所处的状态	0x2402 00	当前模块所处的状态
设置的 CANopen 从节点 ID 值	0x2403 01	该值为节点 ID 的设置值
设置 CANopen 从节点 CAN 波特率	0x2403 02	该值为 0xFF 时无效(波特率索引值)
设置 CANopen 从节点 CAN 波特率真实值	0x2401 03	该值为 0xFFFFFFFF 时无效(波特率真实值)
通信串口波特率索引值	0x2405 00	当前通信串口的波特率

## 附录 C: GCAN-305 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.)
---------------	-------------------	--------------	--------------	---------------	----------------	---------------

### 通信参数区

0x1000	-	Device Type	UINT32	RO	0x00000000	设备类型，高 2 字节为广成科技专用，低二字节供用户使用
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-305 name	STRING	Const	GCAN-305	GCAN-305 设备名称
0x1009		GCAN-305 hardware version	STRING	Const	V1.01	GCAN-305 硬件版本
0x100A		GCAN-305 software version	STRING	Const	V1.00	GCAN-305 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-305产品代码
	3	Revision number	UINT32	RO	-	GCAN-305修订码
	4	Serial number	UINT32	RO	-	GCAN-305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	NOEID+ 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	NOEID+ 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	NOEID+ 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+0x480	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

### 设备状态

0x2400		Actual Node ID	UINT8		-	模块当前的 Node ID
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 Node ID	UINT8	RO	-	用户通过uart口设置的 Node ID, 可能与当前 Node ID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 Code	UINT32	RO	-	用户设备产品代码
	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

售前服务电话与微信号：18309815706

售后服务电话与微信号：13840170070



全国服务电话：400-6655-220