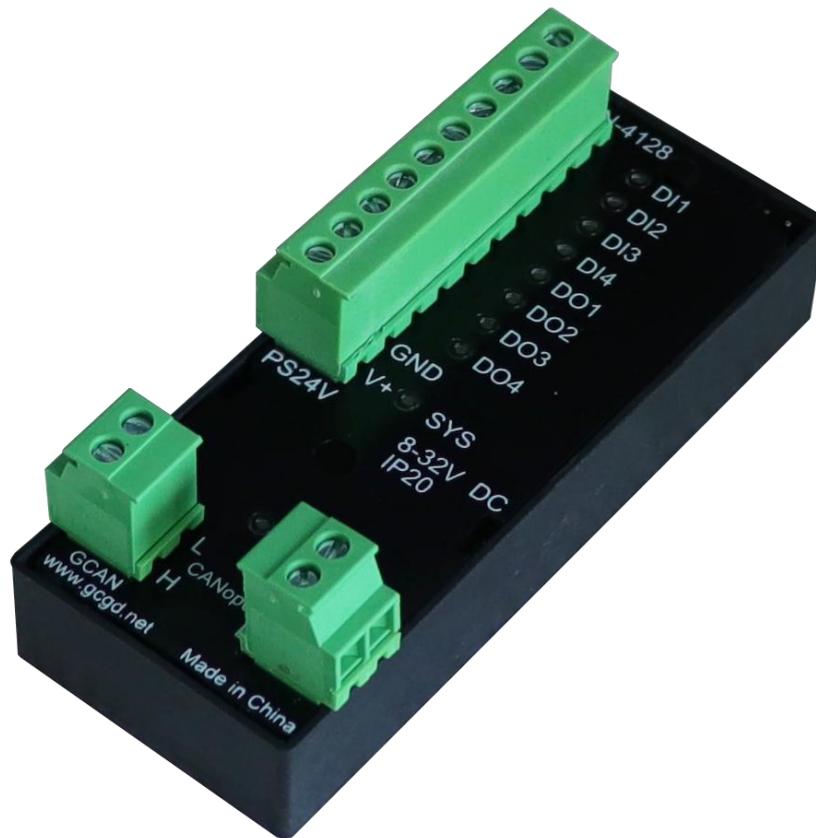


GCAN-4128

CAN/CANopen远程IO-4DI/4DO

用户手册



修订历史

| 版本 | 日期 | 原因 |
|-------|------------|--------|
| V1.00 | 2022/03/31 | 创建文档 |
| V1.10 | 2022/04/29 | 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 典型应用..... | 4 |
| 2 设备安装..... | 5 |
| 2.1 设备尺寸..... | 5 |
| 2.2 接口定义及功能..... | 5 |
| 3 设备使用..... | 7 |
| 3.1 电源连接..... | 7 |
| 3.4 系统状态指示灯..... | 7 |
| 4 快速上手..... | 10 |
| 5 标准 CANopen 模式..... | 11 |
| 5.1 上电启动报文..... | 11 |
| 5.2 NMT 状态切换..... | 11 |
| 5.3 PDO 命令..... | 11 |
| 5.4 SDO 命令..... | 12 |
| 5.4.1 SDO 写入对象字典..... | 12 |
| 5.4.2 修改使用模式..... | 13 |
| 5.4.3 修改节点号 Node ID..... | 14 |
| 5.4.4 修改波特率..... | 14 |
| 5.4.5 修改每一路 DI 的显示位置..... | 14 |
| 5.4.6 修改每一路 DO 的写入位置..... | 15 |
| 5.5 配置操作举例..... | 15 |
| 5.6 使用操作举例..... | 16 |
| 6 CAN 模式..... | 18 |
| 6.1 修改 CAN 模式下 DI 输入数据长度、循环时间..... | 18 |
| 6.2 修改每一路 DI 的显示位置..... | 18 |
| 6.3 修改 CAN 模式下 DO 数据长度..... | 19 |
| 6.4 修改每一路 DO 的写入位置..... | 19 |
| 6.5 修改输入数据帧 ID..... | 19 |
| 6.6 修改输出数据帧 ID..... | 20 |
| 6.7 操作举例..... | 20 |
| 7. GCAN-4128 对象字典..... | 22 |
| 8 技术规格..... | 25 |
| 9. 免责声明..... | 26 |
| 附录 A: CAN2.0A 协议帧格式..... | 27 |
| 附录 B: 设备使用..... | 28 |
| 与 CAN-bus 连接..... | 28 |
| CAN 总线终端电阻..... | 28 |
| 附录 C: CANopen 协议简介..... | 30 |
| 销售与服务..... | 37 |

1 功能简介

1.1 功能概述

GCAN-4128 模块是集成 1 路 CAN 总线，2 个 CAN 接口、4 路数字量输入通道、4 路数字量输出通道的工业级 CAN/CANopen 模式数字量输入输出模块。采用 GCAN-4128 模块，用户可用于采集数字量输入信号，并输出数字量信号，控制开关，实现远程开关设备和 CAN 网络之间的连接，构成 CAN 网络中数字量信号采集、输出的控制节点。

GCAN-4128 模块是数字量信号采集和输出的关键性工具，同时该模块具有体积小、接线便捷等特点，在机柜里甚至不需要固定，只要有个小空间，甩线进去即可。GCAN-4128 模块的总线接口集成隔离保护模块，使其避免由于瞬间的过压过流而对模块造成损坏。

1.2 性能特点

- CAN-bus 支持 CAN2.0A 帧格式，符合 ISO/DIS 11898 规范；
- CAN-bus 通讯波特率在 10Kbps~1Mbps 之间任意可编程；
- CAN-bus 接口采用电气隔离，隔离模块绝缘电压：DC 1500V；
- 使用 8~32V DC 供电（推荐标准电压 24V）；
- 数字量输入通道数：4 路；
- 数字量输出通道数：4 路；
- 数字量输入高电平信号（数字 1）：5V~32V；
- 数字量输入低电平信号（数字 0）：0~2V；
- 数字量输出信号：有输出时，供电电压-0.5V，额定输出电流 100mA；
- 数字量输出信号：无输出时，0~0.5V，不输出电流；
- 工作温度范围：-40℃~+85℃；
- 工作湿度：≤95%RH，无凝露；
- 防护等级：IP20；
- EMC 测试：EN50295。

1.3 典型应用

- 工业现场网络数据监控；
- 煤矿、油井远程通讯；
- CAN 教学应用远程通讯；
- CAN 工业自动化控制系统；
- 智能楼宇控制数据广播系统等 CAN-bus 应用系统。

2 设备安装

2.1 设备尺寸

设备外形尺寸：(长)78mm * (宽)31.5mm * (高，含接线端子)26.5mm，其示意图如图 2.1 所示。

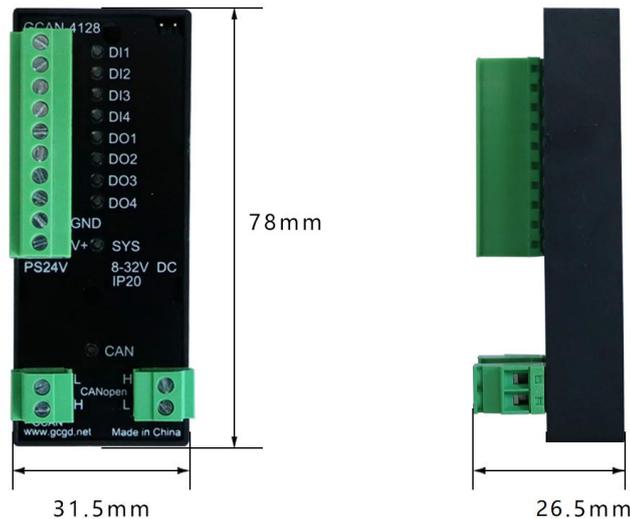


图 2.1 GCAN-4128 外形尺寸

2.2 接口定义及功能

GCAN-4128模块集成4路数字量输入接口、4路数字量输出接口、1路DC8~32V电源接口、1路CAN总线，2个CAN接口、1路系统复位接口。GCAN-4128模块数字量接口、电源接口、系统复位接口位置如图2.2所示，接口定义如表2.2所示。



图 2.2 4 路数字量输入输出、电源及恢复出厂接口位置

| 引脚 (由上至下) | 端口 | 名称 | 功能 |
|--------------|------|---------|--------------|
| 1-4 | 数字量 | DI1~DI4 | 数字量输入通道 |
| 5-8 | | DO1~DO4 | 数字量输出通道 |
| 9 | 电源 | GND | 8~32V 直流电源负极 |
| 10 | | V+ | 8~32V 直流电源正极 |
| 右上 | 系统复位 | reset | 上电短接后恢复出厂默认值 |

表 2.2 4 路数字量输入输出、电源及系统复位接口定义

GSCAN-4128模块的CAN-bus通道，由1路2 Pin插拔式接线端子引出，左右2个CAN接口，可以用于连接1个CAN-bus网络或者CAN-bus接口的设备，支持CANopen协议。GSCAN-4128模块CAN-bus接口接口位置如图2.3所示，接口定义如表2.3所示。



图 2.3 CAN-bus 接口位置

| 引脚 (由左至右) | 端口 | 名称 | 功能 |
|--------------|-------|----|-------------------|
| CAN_L | CAN 左 | L | CAN_L 信号线 (CAN低) |
| CAN_H | | H | CAN_H 信号线 (CAN高) |
| CAN_H | CAN 右 | H | CAN_H 信号线 (CAN 高) |
| CAN_L | | L | CAN_L 信号线 (CAN低) |

表 2.3 CAN-bus 接口定义

3 设备使用

3.1 电源连接

GCAN-4128 模块支持工业现场常见的+8~32V DC 直流电源。为保证模块可靠工作，建议使用+12V 或+24V 的 DC 直流稳压电源。

3.4 系统状态指示灯

GCAN-4128模块具有1个SYS指示灯、8个IO指示灯、1个CAN指示灯来指示设备的运行状态。这3种指示灯的具体指示功能及状态如表3.2所示。

| 指示灯 | 状态 | 指示状态 | 图示 |
|-----|------|--------------------|-------|
| SYS | 绿色快闪 | 设备初始化通过， 进入待机状态 | 图 3.1 |
| | 绿色慢闪 | CANopen 模式 | |
| | 绿色常亮 | CAN 模式 | |
| | 不亮 | 电源供电故障 | |
| CAN | 绿色慢闪 | 数据量小、速率慢 | 图 3.2 |
| | 绿色快闪 | 数据量大、速率快 | |
| | 红绿闪烁 | CAN接口数据 传输错误 | |
| DI | 绿色常亮 | 对应DI有输入 | 图 3.3 |
| | 不亮 | 对应DI无输入 | |
| DO | 橘色常亮 | 对应DO有输出 | 图 3.4 |
| | 不亮 | 对应DO无输出 | |

表 3.2 GCAN-4128 模块指示灯状态

- GCAN-4128模块上电，SYS 指示灯绿色快闪后，若是默认的CANopen 模式，SYS 指示灯绿色慢闪；若是CAN模式，SYS 指示灯绿色常亮。
- GCAN-4128模块上电，数据量小、速率慢时，CAN 指示灯绿色慢闪，数据量大、速率快时，CAN 指示灯绿色极快闪。
- 当CAN接口数据传输错误时，CAN 指示灯绿色闪烁，待CAN接口数据恢复正常，CAN 指示灯绿色闪烁。
- DI有输入，对应DI灯绿色常亮；无输入，对应DI灯不亮。
- DO有输出，对应DO灯橘色常亮；无输出，对应DO灯不亮

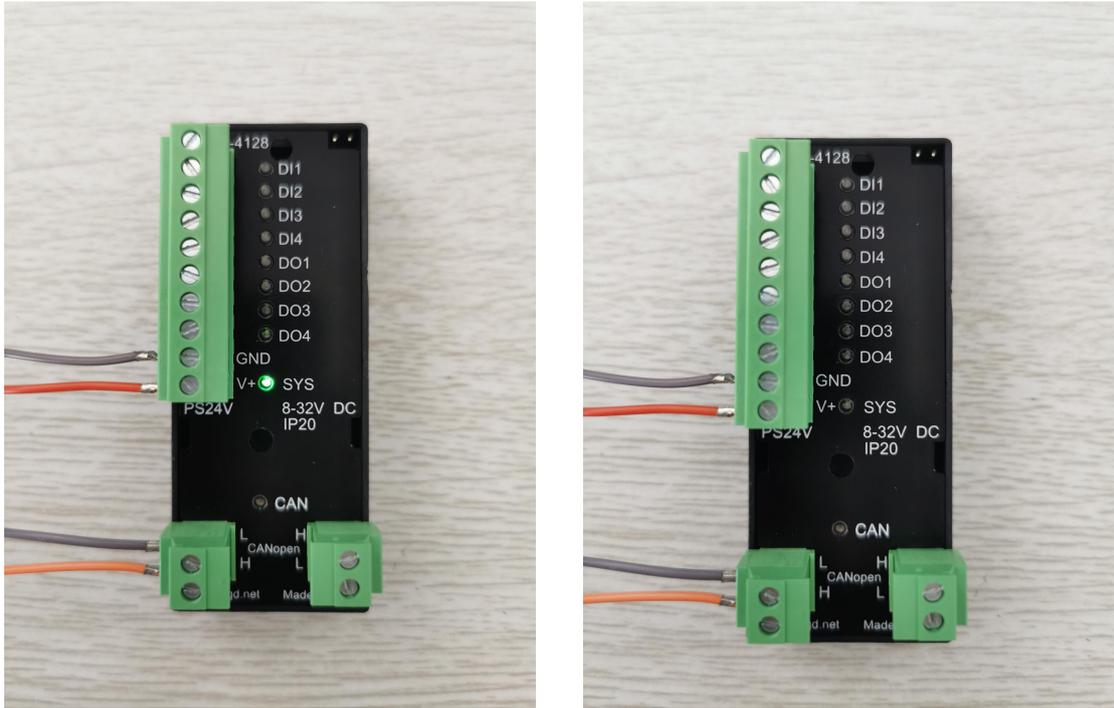


图 3.1 GCAN-4128 SYS 指示灯状态

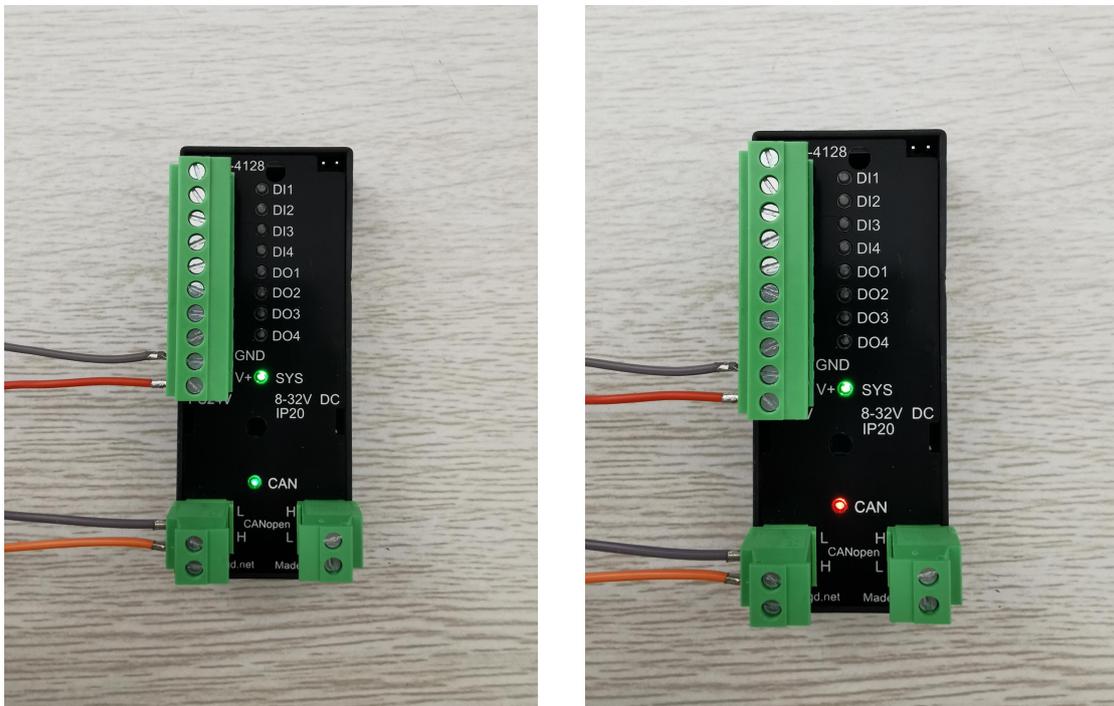


图 3.2 GCAN-4128 CAN 指示灯状态

GCAN-4128 模块 DI1 有输入，DI2-DI4 无输入，如图 3.3 所示；DO1 有输出，DO2-4 无输出，如图 3.4 所示。



图 3.3 GCAN-4128 DI 指示灯状态

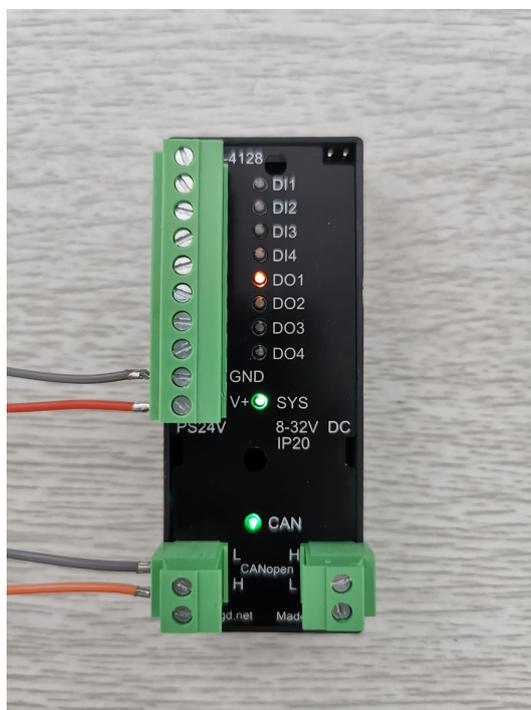


图 3.4 GCAN-4128 DO 指示灯状态

4 快速上手

GCAN-4128 模块支持 CANopen 和 CAN 两种模式，默认标准 CANopen 模式，波特率 500Kbps，使用说明如下：

GCAN-4128 模块默认上电后切换至 **CANopen 模式** 的 **05 启动状态**，默认模块上电后立即对数字量信号进行采集并可控制（写入）输出。

例如：默认 GCAN-4128 的 Node ID 为 1（以下节点号均为 1），则主站设备可接收到一条节点状态数据，帧 ID 为 0x701，标准帧，数据为 0x05。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|----|-----|------|----------|-----|-------------------------|-----|
| 00000000 | 000.020.013 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 63 |
| 00000001 | 000.000.000 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 1 |

当前状态，**TPDO1** 用于表示数字量 **DI** 状态，如图所示，帧 ID 为 **0x180+Node ID**，默认帧数据第一个字节为数字量输入状态。**DI4** 在高位，**DI1** 在低位。

RPDO1 用于表示控制（写入）数字量输出的状态，帧 ID 为 **0x200+Node ID**，数字量输出默认由一个字节来控制（写入），**DO4** 在高位，**DO1** 在低位。

具体请参照 5.3 章节。

5 标准 CANopen 模式

5.1 上电启动报文

GCAN-4128 模块满足标准 CANopen CiA 301 协议，是标准的 CANopen **从站** 设备。**GCAN-4128 启动后将主动发出一帧数据给主站，帧 ID 为 0x700+Node ID。**

例如：默认 GCAN-4128 的 Node ID 为 1（以下节点号均为 1），则**主站**设备可接收到一条节点状态数据，帧 ID 为 0x701，帧数据为 0x7F。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|----|-----|------|----------|-----|----|-----|
| 00000000 | 000.000.000 | 接收 | 701 | DATA | STANDARD | 1 | 7F | 1 |

5.2 NMT 状态切换

GCAN-4128 模块接收由主站发出的操作指令，帧 ID 为 0x000，DLC 为 2，帧数据第一个字节为命令符，第二个字节为节点号（若为 00 则控制全部节点）。

例如：GCAN-4128 的 Node ID 为 1，命令 GCAN-4128 为进入操作状态 (01)，则 NMT 命令帧 ID 为 0x000，帧数据为 0x01,0x01。

主站发送数据 01 01，GCAN-4128 模块进入 05 启动状态。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|------|-----|------|----------|-----|-------|-----|
| 00000000 | 001.000.031 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 384 |
| 00000001 | 000.000.000 | 发送成功 | 000 | DATA | STANDARD | 2 | 01 01 | 1 |

5.3 PDO 命令

GCAN-4128 模块采用 PDO（Process Data Object，过程数据对象）对数字量信号进行采集或写入输出。**DI、DO 每一路的具体位置均可自由配置**，详见 5.4 章节。

其中，**TPDO1 用于表示数字量 DI 状态，帧 ID 为 0x180+Node ID，默认帧数据第一个字节为数字量输入状态。**

TPDO1:

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x180 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | DI 状态 | | | | | | | |
| | 数据 | NC |

DI4 在高位，DI1 在低位。例如，GCAN-4128 模块节点号设为 1。此时 DI1 状态为 1，其余 DI 状态均为 0。则 CAN 总线**主站**一端接收的 DI 状态数据为 0x01。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|----|-----|------|----------|-----|-------------------------|-------|
| 00000000 | 000.020.015 | 接收 | 181 | DATA | STANDARD | 8 | 01 00 00 00 00 00 00 00 | 44048 |
| 00000001 | 001.000.063 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 880 |

如需要控制（写入）GCAN-4128 模块数字量输出的状态，需接收 RPDO1 命令（由主站发出的），帧 ID 为 0x200+Node ID，数据长度为 1，帧数据第一个字节为需要设置的数字量输出状态。

RPDO1 用于表示改变（写入）数字量输出的状态，帧 ID 为 0x200+Node ID，数字量输出默认由一个字节来控制（写入），DO4 在高位，DO1 在低位。

RPDO1:

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x200 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | DO 状态 | | | | | | | |
| | 数据 | NC |

DO4 在高位，DO1 在低位。例如，GCAN-4128 模块节点号设为 1。若需要设置 DO4 状态为 1，其余 DO 状态均为 0，则 CAN 总线主站一端发送的帧 ID 为 0x201，帧数据为 0x08。

| 序号 | 帧间隔时间us | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|-------------|------|-----|------|----------|-----|-------------------------|------|
| 00000000 | 001.000.056 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 64 |
| 00000001 | 000.020.015 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 2996 |
| 00000002 | 000.000.000 | 发送成功 | 201 | DATA | STANDARD | 1 | 08 | 1 |

回到 GCAN-4128 模块，DO4 口对应橘灯亮，代表 DO4 口有输出。

5.4 SDO 命令

服务数据对象 SDO（ServiceDataObjects）主要用来访问节点的对象字典，可以直接对 GCAN-4128 的参数进行读写配置。**配置后重新上电生效。**

5.4.1 SDO 写入对象字典

SDO 写入对象字典中数据类型的长度为 32 位参数命令：

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-----------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令符 | 索引 | | 子索引 | 数据 | | | |
| | 0x23 | 低位 | 高位 | | 低位-----高位 | | | |

SDO 写入对象字典中数据类型的长度为 16 位参数命令：

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 | 索引 | | 子索 | 数据 | | | |

| | | | | | | | | |
|--|------|----|----|--|----|----|------|------|
| | 符 | | 引 | | | | | |
| | 0x2B | 低位 | 高位 | | 低位 | 高位 | 0x00 | 0x00 |

SDO 写入对象字典中数据类型的长度为 8 位参数命令：

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x600 + NODE-ID | 指令符 | 索引 | | 子索引 | 数据 | 0x00 | 0x00 | 0x00 |
| | 0x2F | 低位 | 高位 | | | | | |

从机应答写入对象字典的值写入成功：

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x580 + NODE-ID | 指令符 | 索引 | | 子索引 | 预留 | | | |
| | 0x60 | 低位 | 高位 | | | | | |

从机应答写入对象字典的值写入失败：

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x580 + NODE-ID | 指令符 | 索引 | | 子索引 | 中止代码 | | | |
| | 0x80 | 低位 | 高位 | | | | | |

5.4.2 修改使用模式

GCAN-4128 模块可通过发送 SDO 命令写参数，修改模块的使用模式。该参数位于对象字典索引 0x2400、子索引 0x00，数据类型 Unsigned8。

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x600 + NODE-ID | 指令符 | 索引 | | 子索引 | 数据 | 0x00 | 0x00 | 0x00 |
| | 0x2F | 00 | 24 | | | | | |

| 数据 | 模式 |
|------|--------|
| 0x01 | CAN 模式 |
| 0x02 | 默认模式 |

| | |
|------|---------------|
| 0x03 | 标准 CANopen 模式 |
|------|---------------|

5.4.3 修改节点号 Node ID

GCAN-4128 模块可通过发送 SDO 命令写参数，修改 Node ID。该传输类型的数据位于对象字典索引 0x2407、子索引 0x00，数据类型 Unsigned8。**Node ID 范围为 1~127，对十六进制为 0x01~0x7F。**

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 数据 | 0x00 | 0x00 | 0x00 |
| | 0x2F | 07 | 24 | 00 | | | | |

5.4.4 修改波特率

GCAN-4128 模块可通过发送 SDO 命令写参数，修改模块的波特率。该传输类型的数据位于对象字典索引 0x2408、子索引 0x00，数据类型 Unsigned32。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 数据 | 0x00 | 0x00 | 0x00 |
| | 0x23 | 08 | 24 | 00 | | | | |

数据对应波特率的关系见下表，其他数据不生效。

| 数据 | 波特率 |
|------|----------|
| 0x00 | 1000Kbps |
| 0x01 | 800Kbps |
| 0x02 | 500Kbps |
| 0x03 | 250Kbps |
| 0x04 | 125Kbps |
| 0x05 | 100Kbps |
| 0x06 | 50Kbps |
| 0x07 | 20Kbps |
| 0x08 | 10Kbps |

5.4.5 修改每一路 DI 的显示位置

GCAN-4128 模块可通过发送 SDO 命令写参数，修改每一路 DI 的显示位置。该传输类型的数据位于对象字典索引 0x2404、子索引 0x00，数据类型 Unsigned32。

最多设置在第 64 位显示。换算成 16 进制，数据最低 0x01，最高 0x40，0x00 或者 0x40 以上不生效。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 第 1 路 | 第 2 路 | 第 3 路 | 第 4 路 |
| | 0x23 | 04 | 24 | 00 | | | | |

5.4.6 修改每一路 DO 的写入位置

GCAN-4128 模块可通过发送 SDO 命令写参数,修改每一路 DO 的写入位置。该传输类型的数据位于对象字典索引 0x2403、子索引 0x00,数据类型 Unsigned32。

最多设置在第 64 位。换算成 16 进制,数据最低 0x01,最高 0x40,0x00 或者 0x40 以上不生效。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 第 1 路 | 第 2 路 | 第 3 路 | 第 4 路 |
| | 0x23 | 03 | 24 | 00 | | | | |

5.5 配置操作举例

CANopen 模式配置操作举例:(以节点号为 1 举例) **默认模式上电后:**

1、模块配置为标准 CANopen 模式

报文: ID: 0x601 数据: 2F 00 24 00 03 00 00 00, 设置模块为 CAN 模式。

2、修改节点号 Node ID

报文: ID: 0x601 数据: 2F 07 24 00 03 00 00 00, 设置模块节点号 Node ID 为 3。

3、修改波特率

报文: ID: 0x601 数据: 23 08 24 00 01 00 00 00, 设置模块波特率为 800Kbps。

4、修改每一路 DI 的显示位置

报文: ID: 0x601 数据: 23 04 24 00 02 14 28 3C, 设置模块第 4 路在 0x3C 即十进制第 60 位显示, 模块第 3 路在 0x28 即十进制第 40 位显示, 模块第 2 路在 0x14 即十进制第 20 位显示, 模块第 4 路在 0x02 即十进制第 2 位显示。

5、修改每一路 DO 的写入位置

报文: ID: 0x601 数据: 23 03 24 00 02 14 28 3C, 设置模块第 4 路在 0x3C 即十进制第 60 位可被写入, 模块第 3 路在 0x28 即十进制第 40 位可被写入, 模块第 2 路在 0x14 即十进制第 20 位可被写入, 模块第 4 路在 0x02 即十进制第 2 位可被写入。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|------|-----|------|----------|-----|-------------------------|-----|
| 00000377 | 000.020.012 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000378 | 000.019.988 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000379 | 000.020.011 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000380 | 000.020.003 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000381 | 000.019.993 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000382 | 000.020.011 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000383 | 000.019.993 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000384 | 000.020.017 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000385 | 000.020.015 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000386 | 000.367.845 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 1 |
| 00000387 | 027.816.677 | 发送成功 | 601 | DATA | STANDARD | 8 | 2F 00 24 00 03 00 00 00 | 1 |
| 00000388 | 009.740.982 | 接收 | 581 | DATA | STANDARD | 8 | 60 00 24 00 02 00 00 00 | 1 |
| 00000389 | 026.087.995 | 发送成功 | 601 | DATA | STANDARD | 8 | 2F 07 24 00 03 00 00 00 | 1 |
| 00000390 | 026.087.547 | 接收 | 581 | DATA | STANDARD | 8 | 60 07 24 00 01 00 00 00 | 1 |
| 00000391 | 011.699.125 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 08 24 00 01 00 00 00 | 1 |
| 00000392 | 011.698.345 | 接收 | 581 | DATA | STANDARD | 8 | 60 08 24 00 02 00 00 00 | 1 |
| 00000393 | 013.783.920 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 04 24 00 02 14 28 3C | 1 |
| 00000394 | 013.782.845 | 接收 | 581 | DATA | STANDARD | 8 | 60 04 24 00 01 02 03 04 | 1 |
| 00000395 | 006.878.284 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 03 24 00 02 14 28 3C | 1 |
| 00000396 | 006.879.999 | 接收 | 581 | DATA | STANDARD | 8 | 60 03 24 00 01 02 03 04 | 1 |

重新上电后，配置生效。当前标准 CANopen 模式可以直接配置。

5.6 使用操作举例

CANopen 模式操作举例：（以节点号为 1 举例）配置完重新上电后：

1、NMT 状态切换

报文：ID: 0x000 数据：01 01，设置模块为 05 启动状态。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|------|-----|------|----------|-----|-------|-----|
| 00000000 | 001.000.031 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 384 |
| 00000001 | 000.000.000 | 发送成功 | 000 | DATA | STANDARD | 2 | 01 01 | 1 |

当前状态，即可读取 DI 数据，或对 DO 数据进行写入。

2、第 4 路在第 60 位显示，第 3 路在第 40 位显示，第 2 路在第 20 位显示，第 4 路在第 2 位显示。

TPDO1：帧 ID 为 0x180+Node ID，用以读取 DI 数据

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x180 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | DI1 | | DI2 | | DI3 | | | DI4 |
| | 0x02 | 00 | 08 | 00 | 0x80 | 00 | 00 | 0x08 |

若第一路、第二路有输入，第三路、第四路无输入，数据如图所示：

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|----|-----|------|----------|-----|-------------------------|-----|
| 00000000 | 000.020.018 | 接收 | 181 | DATA | STANDARD | 8 | 02 00 08 00 00 00 00 00 | 133 |
| 00000001 | 001.000.031 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 2 |

3、第 4 路在第 60 位可被写入（设置），第 3 路在第 40 位可被写入（设置），第 2 路在第 20 位可被写入（设置），第 4 路在第 2 位可被写入（设置）。

RPDO1：帧 ID 为 0x200+Node ID，用以写入（设置）DO 数据

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0x200 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | D01 | | D02 | | D03 | | | D04 |
| | 0x02 | 00 | 08 | 00 | 0x80 | 00 | 00 | 0x08 |

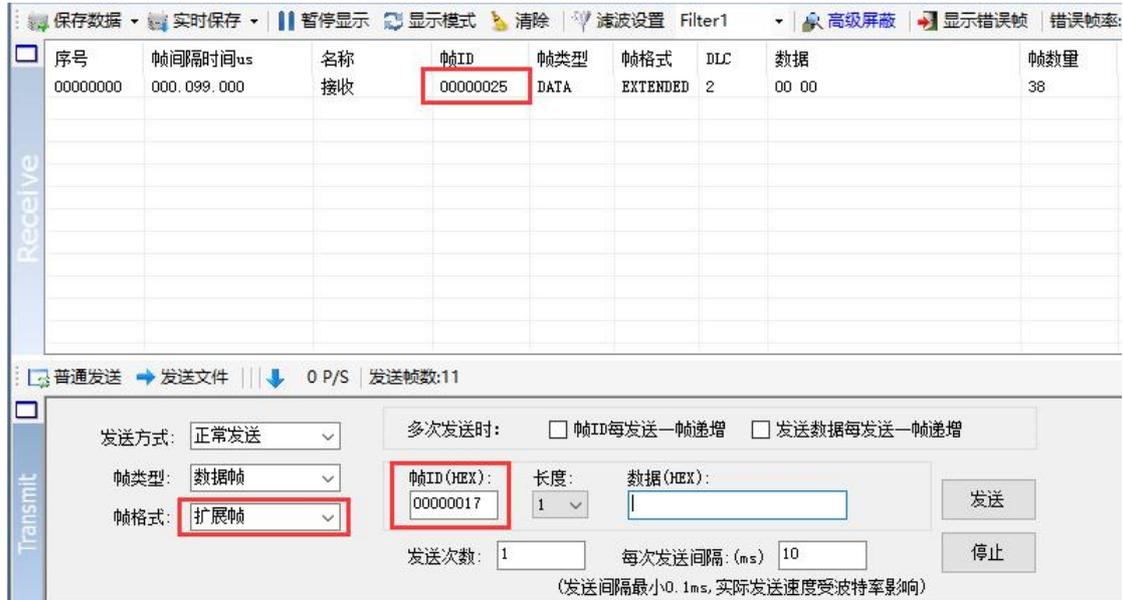
若写入（设置）第一路、第二路无输出，第三路、第四路有输出，数据如图所示：

| 序号 | 帧间隔时间us | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|-------------|------|-----|------|----------|-----|-------------------------|-----|
| 00000000 | 000.020.017 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 324 |
| 00000001 | 001.000.038 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 7 |
| 00000002 | 000.000.000 | 发送成功 | 201 | DATA | STANDARD | 8 | 00 00 00 00 80 00 00 08 | 1 |

6 CAN 模式

GCAN-4128 模块配置为 CAN 模式后，重新上电，CAN 模式默认的数据收发如图所示。

输入帧 ID 指的是输入 DI 的 ID，默认 0x00000025；输出帧 ID 指的是可通过该 ID 控制 DO 的输出，默认为 0x00000017，扩展帧有效。



6.1 修改 CAN 模式下 DI 输入数据长度、循环时间

GCAN-4128 模块可通过发送 SDO 命令写参数，修改 DI 数据长度、循环时间。该传输类型的数据位于对象字典索引 0x2402、子索引 0x00，数据类型 Unsigned32。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|----------------|-----------|-------|-------|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x600 + NODE-ID | 指令 符 | 索引 | | 子索 引 | DI 数据 长度 | 循环时间 | | |
| | 0x23 | 02 | 24 | 00 | | 低位-----高位 | | |

6.2 修改每一路 DI 的显示位置

GCAN-4128 模块可通过发送 SDO 命令写参数，修改每一路 DI 的显示位置。该传输类型的数据位于对象字典索引 0x2404、子索引 0x00，数据类型 Unsigned32。

最多设置在第 64 位显示。换算成 16 进制，数据最低 0x01，最高 0x40，0x00 或者 0x40 以上不生效。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 第 1 路 | 第 2 路 | 第 3 路 | 第 4 路 |
| | 0x23 | 04 | 24 | 00 | | | | |

6.3 修改 CAN 模式下 DO 数据长度

GCAN-4128 模块可通过发送 SDO 命令写参数，修改 Node ID。该传输类型的数据位于对象字典索引 0x2401、子索引 0x00，数据类型 Unsigned8。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|----------------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | DO 数据 长度 | 0x00 | 0x00 | 0x00 |
| | 0x2F | 2401 | | 00 | | | | |

6.4 修改每一路 DO 的写入位置

GCAN-4128 模块可通过发送 SDO 命令写参数，修改每一路 DO 的写入位置。该传输类型的数据位于对象字典索引 0x2403、子索引 0x00，数据类型 Unsigned32。

最多设置在第 64 位。换算成 16 进制，数据最低 0x01，最高 0x40，0x00 或者 0x40 以上不生效。

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|-------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 第 1 路 | 第 2 路 | 第 3 路 | 第 4 路 |
| | 0x23 | 03 | 24 | 00 | | | | |

6.5 修改输入数据帧 ID

同理，GCAN-4128 模块可通过发送 SDO 命令写参数，修改模块的输入帧 ID。该传输类型的数据位于对象字典索引 0x2406、子索引 0x00，数据类型 Unsigned32。**输入数据帧 ID，指的是 CAN 模式下，输入 DI 数据的帧 ID。**

| COB-ID | 数据 | | | | | | | |
|--------------------|---------|-------|-------|---------|------------|-------|-------|---------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令 符 | 索引 | | 子索 引 | 输入 帧 ID | 0x00 | 0x00 | 帧格 式 |
| | 0x23 | 06 | 24 | 00 | | | | |

| 数据 | 帧格式 |
|------|-----|
| 0x80 | 扩展帧 |
| 0x00 | 标准帧 |

6.6 修改输出数据帧 ID

GCAN-4128 模块可通过发送 SDO 命令写参数，修改模块的输出数据帧 ID。该传输类型的数据位于对象字典索引 0x2405、子索引 0x00，数据类型 Unsigned32。**输出数据帧 ID，指的是 CAN 模式下，通过帧 ID 控制 DO 输出。**

| COB-ID | 数据 | | | | | | | |
|--------------------|-------|-------|-------|-------|--------|-------|-------|-------|
| 0x600 + NODE-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| | 指令符 | 索引 | | 子索引 | 输出帧 ID | 0x00 | 0x00 | 帧格式 |
| | 0x23 | 05 | 24 | 00 | | | | |

| 数据 | 帧格式 |
|------|-----|
| 0x80 | 扩展帧 |
| 0x00 | 标准帧 |

6.7 操作举例

CAN 模式操作举例：（以节点号为 1 举例）**默认模式上电后：**

1、模块配置为 CAN 模式

报文：ID: 0x601 数据：2F 00 24 00 01 00 00 00，设置模块为 CAN 模式。

2、修改 DI 输入长度、循环时间

报文：ID: 0x601 数据：23 02 24 00 08 E8 03 00，设置模块输入长度为 8 个字节，循环时间为 0x03E8 转换为 10 进制是 1000ms。

3、修改每一路 DI 的显示位置

报文：ID: 0x601 数据：23 04 24 00 02 14 28 3C，设置模块第 4 路在 0x3C 即十进制第 60 位显示，模块第 3 路在 0x28 即十进制第 40 位显示，模块第 2 路在 0x14 即十进制第 20 位显示，模块第 4 路在 0x02 即十进制第 2 位显示。

4、修改 CAN 模式下 DO 数据长度

报文：ID: 0x601 数据：2F 01 24 00 08 00 00 00，设置模块输出长度为 8 个字节。

5、修改每一路 DO 的写入位置

报文：ID: 0x601 数据：23 03 24 00 02 14 28 3C，设置模块第 4 路在 0x3C 即十进制第 60 位可被写入，模块第 3 路在 0x28 即十进制第 40 位可被写入，模块第 2 路在 0x14 即十进制第 20 位可被写入，模块第 4 路在 0x02 即十进制第 2 位可被写入。

6、修改输入数据帧 ID

报文：ID: 0x601 数据：23 06 24 00 11 00 00 00，设置模块输入数据帧 ID 为 0x11，帧格式为标准帧。

7、修改输出数据帧 ID

报文：ID：0x601 数据：23 05 24 00 33 00 00 00，设置模块输出数据帧 ID 为 0x33，帧格式为标准帧。

| 序号 | 帧间隔时间 μ s | 名称 | 帧ID | 帧类型 | 帧格式 | DLC | 数据 | 帧数量 |
|----------|---------------|------|----------|------|----------|-----|-------------------------|-----|
| 00000568 | 000.020.003 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000569 | 000.019.991 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000570 | 000.020.011 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000571 | 000.020.011 | 接收 | 181 | DATA | STANDARD | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000572 | 000.567.875 | 接收 | 701 | DATA | STANDARD | 1 | 05 | 1 |
| 00000573 | 095.588.102 | 发送成功 | 601 | DATA | STANDARD | 8 | 2F 00 24 00 01 00 00 00 | 1 |
| 00000574 | 025.432.569 | 接收 | 581 | DATA | STANDARD | 8 | 60 00 24 00 02 00 00 00 | 1 |
| 00000575 | 018.008.764 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 02 24 00 08 E8 03 00 | 1 |
| 00000576 | 018.011.577 | 接收 | 581 | DATA | STANDARD | 8 | 60 02 24 00 02 63 00 00 | 1 |
| 00000577 | 064.130.845 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 04 24 00 02 14 28 3C | 1 |
| 00000578 | 064.127.203 | 接收 | 581 | DATA | STANDARD | 8 | 60 04 24 00 01 02 03 04 | 1 |
| 00000579 | 000.000.324 | 接收 | 06040007 | DATA | EXTENDED | 8 | 00 00 00 00 00 00 00 00 | 1 |
| 00000580 | 026.344.979 | 发送成功 | 601 | DATA | STANDARD | 8 | 2F 01 24 00 08 00 00 00 | 1 |
| 00000581 | 026.345.583 | 接收 | 581 | DATA | STANDARD | 8 | 60 01 24 00 08 00 00 00 | 1 |
| 00000582 | 016.707.956 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 03 24 00 02 14 28 3C | 1 |
| 00000583 | 016.706.182 | 接收 | 581 | DATA | STANDARD | 8 | 60 03 24 00 01 02 03 04 | 1 |
| 00000584 | 013.983.989 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 06 24 00 11 00 00 00 | 1 |
| 00000585 | 013.985.390 | 接收 | 581 | DATA | STANDARD | 8 | 60 06 24 00 25 00 00 80 | 1 |
| 00000586 | 014.329.449 | 发送成功 | 601 | DATA | STANDARD | 8 | 23 05 24 00 33 00 00 00 | 1 |
| 00000587 | 014.327.440 | 接收 | 581 | DATA | STANDARD | 8 | 60 05 24 00 17 00 00 80 | 1 |

重新上电后，配置生效。当前 CAN 模式无法配置，若想重新配置，请上电后短接复位键，即可重新配置。

7. GCAN-4128 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 | 0x000A0011 | 设备类型 |
| 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 | 0x00000080 | - |
| 0x1006 | | Communication Cycle Period | UINT16 | RW | 0x2710 | 通讯循环周期 |
| 0x1007 | | Sync Windows Length | UINT32 | RW | 0 | - |
| 0x1009 | | Manufacturer hardware version | STRING | CONST | 0 | 硬件版本 |
| 0x100A | | Manufacturer software version | STRING | CONST | 0 | 软件版本 |
| 0x100C | | Guard Time | UINT16 | RW | 0 | - |
| 0x100D | | Life Time Factor | UINT8 | RW | 3 | - |
| 0x1017 | | Producer Heartbeat Time | UINT16 | RW | 0x03E8 | - |
| 0x1018 | 0 | number of Entries | UINT8 | RO | 0x04 | - |
| | 1 | Vendor-ID | UINT32 | RO | 0x00000449 | 沈阳广成科技有限公司在CiA组织的厂商代码 |
| | 2 | Product code | UINT32 | RO | 0x00004128 | 产品代码 |
| | 3 | Revision number | UINT32 | RO | 0x00000000 | 修订码 |
| | 4 | Serial number | UINT32 | RO | 0x00000001 | 序列码 |

RPDO 通信参数

| | | | | | | |
|--------|---|-----------------------------|--------|-------|-----------------|----------------|
| 0x1400 | 0 | Highest sub-index supported | UINT8 | CONST | 5 | - |
| | 1 | COB-ID used by RPDO | UINT32 | RW | 0x200 + Node ID | RPDO所使用的COB-ID |
| 0x1401 | 0 | Highest sub-index supported | UINT8 | CONST | 5 | - |
| 0x1402 | 0 | Highest sub-index supported | UINT8 | CONST | 0 | - |
| 0x1403 | 0 | Highest sub-index supported | UINT8 | CONST | 0 | - |
| 0x1600 | 0 | number of mapped | UINT8 | RW | 8 | - |

| | | | | | | |
|--------|---|--------------------------|--------|----|------------|---|
| | | objects | | | | |
| | 1 | 1st application object | UINT32 | RW | 0x21000108 | |
| 0x1601 | 0 | number of mapped objects | UINT8 | RW | 8 | - |
| 0x1602 | 0 | number of mapped objects | UINT8 | RW | 8 | - |
| 0x1603 | 0 | number of mapped objects | UINT8 | RW | 8 | - |

TPDO 通信参数

| | | | | | | |
|--------|---|-----------------------------|--------|-------|---------------|--|
| | 0 | Highest sub-index supported | UINT8 | CONST | 5 | - |
| 0x1800 | 1 | COB-ID used by TPDO | UINT32 | RW | 0x180+ NODEID | TPDO所使用的COB-ID |
| | 2 | transmission type | UINT8 | RW | 0xFF (255) | 传输类型。0xFF为触发模式, 0xFE为循环发送, 循环发送的截止时间在索引0x2000中设置 |
| 0x1801 | 0 | Highest sub-index supported | UINT8 | CONST | 5 | - |
| 0x1802 | 0 | Highest sub-index supported | UINT8 | CONST | 5 | - |
| 0x1803 | 0 | Highest sub-index supported | UINT8 | CONST | 0 | - |
| 0x1A01 | 0 | number of mapped objects | UINT8 | RW | 4 | |
| 0x1A02 | 0 | number of mapped objects | UINT8 | RW | 8 | |
| 0x1A03 | 0 | number of mapped objects | UINT8 | RW | 8 | |

设备状态

| | | | | | | |
|--------|----|------------------------|--------|----|------------|-------------|
| 0x2400 | 00 | Can CANopen Type | UINT8 | RW | 0x02 | 模式切换 |
| 0x2401 | 00 | Outio Canlen | UINT8 | RW | 0x08 | 输出数据长度 |
| 0x2402 | 00 | Inputio Canlen Cantime | UINT32 | RW | 0x00006302 | 输入数据长度、循环时间 |
| 0x2403 | 00 | Out | UINT32 | RW | 0x04030201 | 输出位置 |
| 0x2404 | 00 | Input | UINT32 | RW | 0x04030201 | 输入位置 |
| 0x2405 | 00 | Out Canid | UINT32 | RW | 0x80000017 | 输出 CANID |
| 0x2406 | 00 | Input Canid | UINT32 | RW | 0x80000025 | 输入 CANID |
| 0x2407 | 00 | Nodeid | UINT8 | RW | 0x01 | 节点 ID |

| | | | | | | |
|--------|----|---------|--------|----|------------|---------|
| 0x2408 | 00 | Canbaud | UINT32 | RW | 0x00000002 | CAN 波特率 |
|--------|----|---------|--------|----|------------|---------|

8 技术规格

| 连接方式 | |
|---------|--|
| DI/DO接口 | 端子 |
| CAN接口 | 端子 |
| 接口特点 | |
| CAN接口 | 遵循ISO 11898标准，支持标准CANopen协议，支持CAN2.0A |
| CAN波特率 | 1000K、800K、500K、250K、125K、100K、50K、20K、10K |
| 电气隔离 | 1500V，DC-DC |
| CAN终端电阻 | 未集成，如有需要在CAN_H、CAN_L间添加 |
| 供电电源 | |
| 供电电压 | +8~32V DC |
| 供电电流 | 最大140mA（静默态电流：40mA） |
| 环境试验 | |
| 工作温度 | -40℃~+85℃ |
| 工作湿度 | 15%~90%RH，无凝露 |
| EMC测试 | EN 50295 |
| 防护等级 | IP 20 |
| 基本信息 | |
| 外形尺寸 | 78mm *31.5mm *26.5mm |
| 重量 | 100g |

9. 免责声明

感谢您购买广成科技的 GCAN 系列软硬件产品。GCAN 是沈阳广成科技有限公司的注册商标。本产品及手册为广成科技版权所有。未经许可，不得以任何形式复制翻印。在使用之前，请仔细阅读本声明，一旦使用，即被视为对本声明全部内容的认可和接受。请严格遵守手册、产品说明和相关的法律法规、政策、准则安装和使用该产品。在使用产品过程中，用户承诺对自己的行为及因此而产生的所有后果负责。因用户不当使用、安装、改装造成的任何损失，广成科技将不承担法律责任。

关于免责声明的最终解释权归广成科技所有。

附录 A：CAN2.0A 协议帧格式

1.CAN2.0A 标准帧

CAN 标准帧信息为11个字节，包括两部分：信息和数据部分。前3个字节为信息部分。

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------|-----------|-----|---|---|------------|---|---|---|--|
| 字节 1 | FF | RTR | × | × | DLC（数据长度） | | | | |
| 字节 2 | （报文识别码） | | | | ID.10—ID.3 | | | | |
| 字节 3 | ID.2—ID.0 | | | × | × | × | × | × | |
| 字节 4 | 数据 1 | | | | | | | | |
| 字节 5 | 数据 2 | | | | | | | | |
| 字节 6 | 数据 3 | | | | | | | | |
| 字节 7 | 数据 4 | | | | | | | | |
| 字节 8 | 数据 5 | | | | | | | | |
| 字节 9 | 数据 6 | | | | | | | | |
| 字节 10 | 数据 7 | | | | | | | | |
| 字节 11 | 数据 8 | | | | | | | | |

字节1为帧信息。第7位（FF）表示帧格式，在标准帧中，FF=0；第6位（RTR）表示帧的类型，RTR=0表示为数据帧，RTR=1表示为远程帧；DLC 表示在数据帧时实际的数据长度。

字节2、3 为报文识别码，11位有效。

字节4~11为数据帧的实际数据，远程帧时无效。

附录 B：设备使用

与 CAN-bus 连接

GCAN-4128模块接入CAN总线连接方式为将CAN_H连CAN_H，CAN_L连CAN_L即可建立通信。

CAN-bus网络采用直线拓扑结构，总线最远的2个终端需要安装120Ω的终端电阻；如果节点数目大于2，中间节点不需要安装120Ω的终端电阻。对于分支连接，其长度不应超过3米。CAN-bus总线的连接如图1所示。

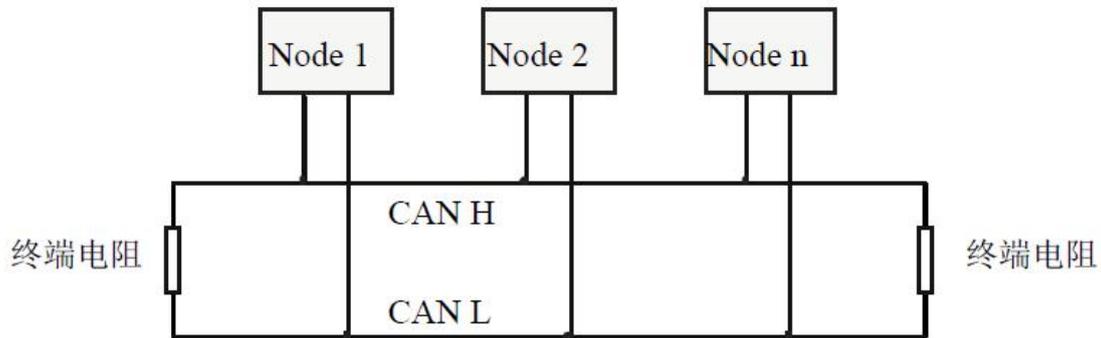


图 1 CAN-bus 网络的拓扑结构

请注意：CAN-bus 电缆可以使用普通双绞线、屏蔽双绞线。理论最大通信距离主要取决于总线波特率，最大总线长度和波特率关系详见表 1。若通讯距离超过 1km，应保证线的截面积大于Φ 1.0mm²，具体规格应根据距离而定，常规是随距离的加长而适当加大。

| 波特率 | 总线长度 |
|------------|-------|
| 1 Mbit/s | 25m |
| 800 kbit/s | 66m |
| 500 kbit/s | 100m |
| 250 kbit/s | 250m |
| 125 kbit/s | 500m |
| 50 kbit/s | 1.0km |
| 20 kbit/s | 2.5km |
| 10 kbit/s | 5.0km |

表 1 波特率与最大总线长度参照表

CAN 总线终端电阻

为了增强CAN通讯的可靠性，消除CAN总线终端信号反射干扰，CAN总线网络最远的两个端点通常要加入终端匹配电阻，如图2所示。终端匹配电阻的值

由传输电缆的特性阻抗所决定。例如双绞线的特性阻抗为 120Ω ，则总线上的两个端点也应集成 120Ω 终端电阻。如果网络上其他节点使用不同的收发器，则终端电阻须另外计算。

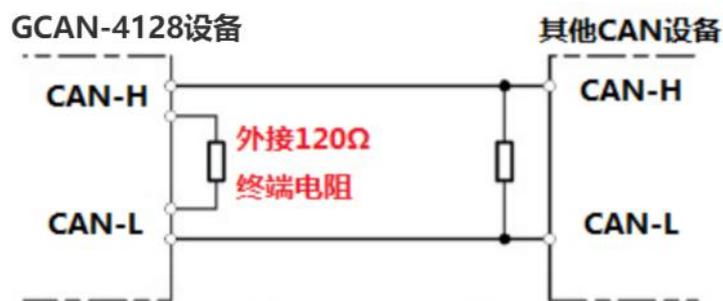


图2 GCAN-4128 与其他 CAN 节点设备连接

请注意：GCAN-4128模块内部未集成 120Ω 终端电阻。如果节点数目大于2，中间节点不需要安装 120Ω 的终端电阻。需要使用时，将电阻两端分别接入CAN_H、CAN_L即可，如图2所示。

附录 C: 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。

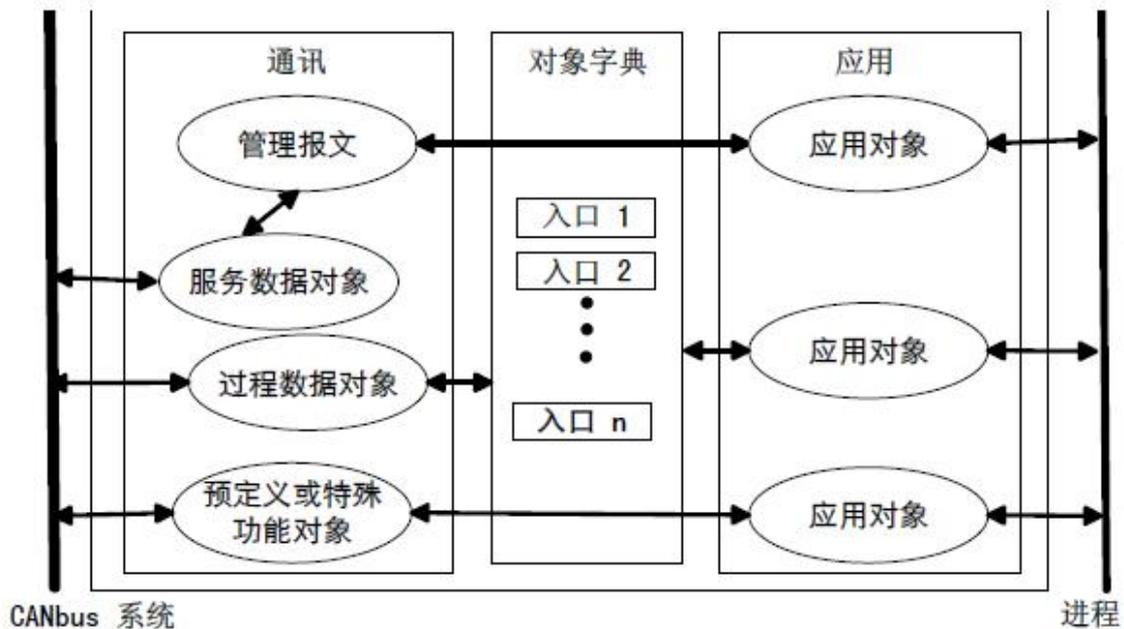


图 B.1 CANopen 设备结构

图B.1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。

B.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。

OD: Object Dictionary, 对象字典。

EDS: Electronic Data Sheet, 电子数据文档。

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B.2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。

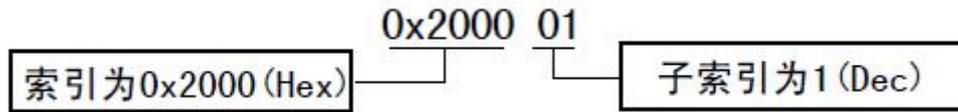


图 B.2 索引/子索引书写规则

B.2 预定义CAN标识符

| Object对象 | 功能代码 | CAN-ID范围 |
|---------------------------------|-------|-----------|
| NMT网络管理命令 | 0000b | 000h |
| Sync同步报文 | 0001b | 080h |
| Time Stamp时间戳报文 | 0010b | 100h |
| Emergency紧急报文 | 0001b | 081h-0FFh |
| TPDO1发送过程数据对象1 | 0011b | 181h-1FFh |
| RPDO1接收过程数据对象1 | 0100b | 201h-27Fh |
| TPDO2发送过程数据对象2 | 0101b | 281h-2FFh |
| RPDO2接收过程数据对象2 | 0110b | 301h-37Fh |
| TPDO3发送过程数据对象3 | 0111b | 381h-3FFh |
| RPDO3接收过程数据对象3 | 1000b | 401h-47Fh |
| TPDO4发送过程数据对象4 | 1001b | 481h-4FFh |
| RPDO4接收过程数据对象4 | 1010b | 501h-57Fh |
| SDO Server-to-Client 服务数据对象 (答) | 1011b | 581h-5FFh |
| SDO Client-to-Server 服务数据对象 (问) | 1100b | 601h-67Fh |
| NMT error control 网络管理错误控制 | 1110b | 701h-77Fh |

B.3 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协议包含了许多的子协议，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)

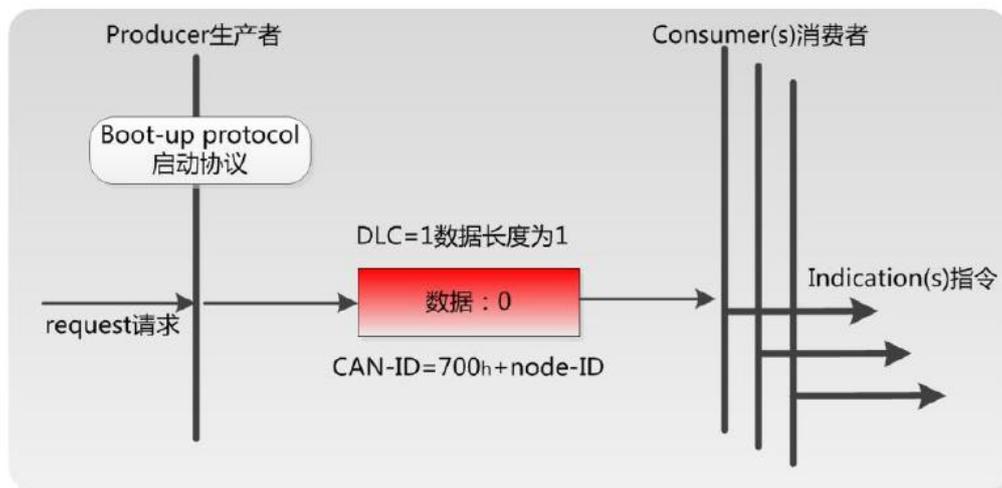
设备子协议，为各种不同类型的设备定义对象字典中的对象。目前已有十几种为不同类型的设备定义的子协议，例如DS401、DS402、DS406 等，其索引值范围为0x6000~0x9FFF。

B.4 CANopen通讯

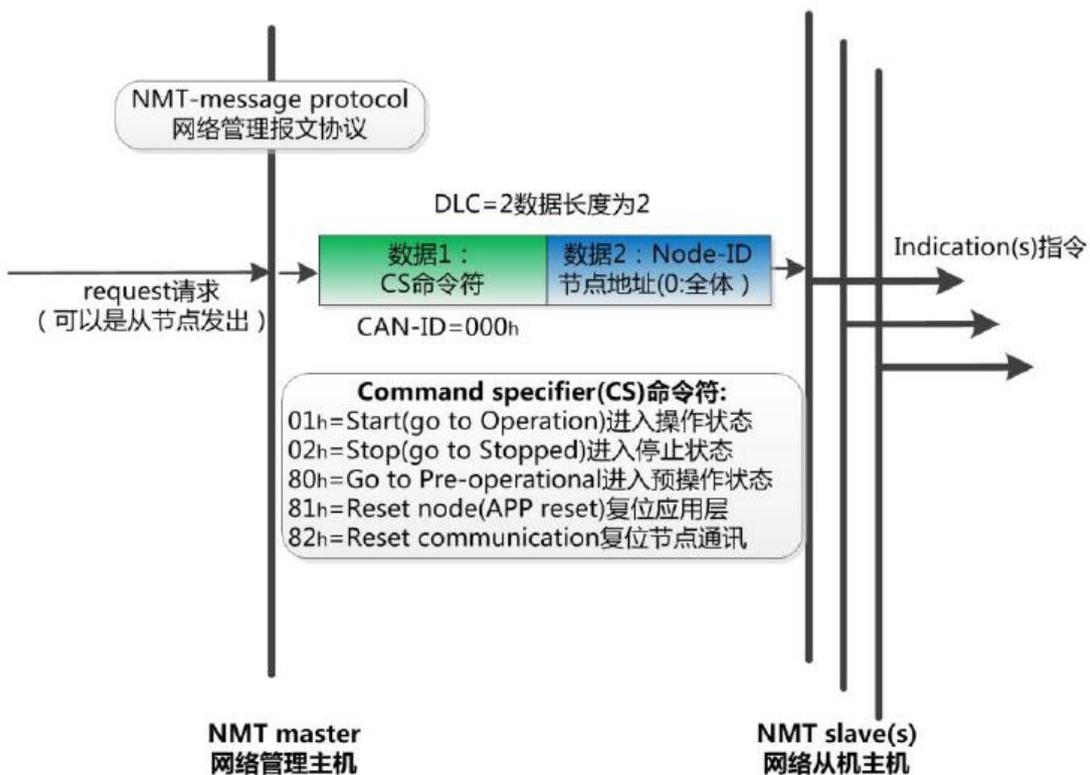
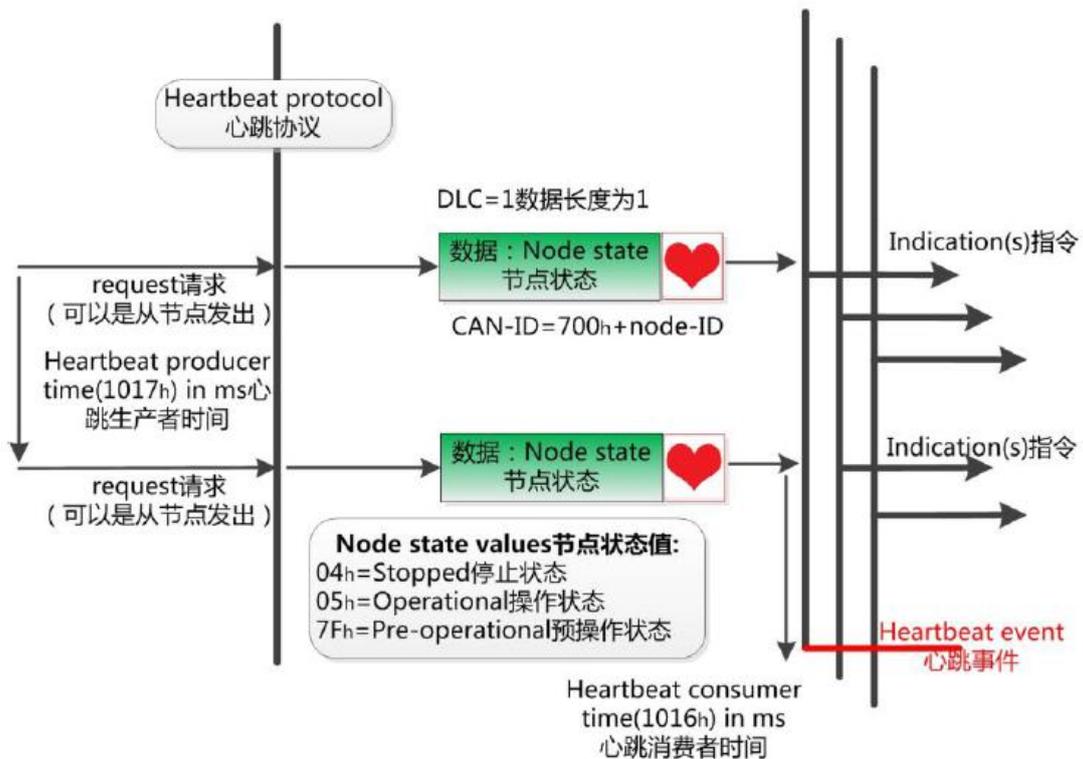
在CANopen协议中主要定义了管理报文对象NMT（Network Management）、服务数据对象SDO(Service Data Object)、过程数据对象PDO(Process Data Object)、预定义报文或特殊功能对象等四种对象。

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。



如上图所示，举个例子，某 CANopen 从站设备上电之后将发送一个帧 ID 为 0x702，数据为 0x00 的数据；说明该设备已启动，且节点号为 2。



如上图所示，举个例子，某 CANopen 主站向从站发送一帧数据，帧 ID 为 0x000，帧数据为 0x01、0x02，则该指令可使节点号为 2 的 CANopen 从站设备

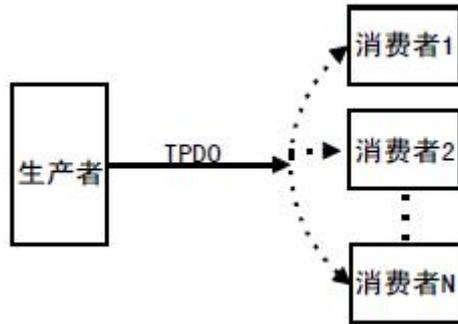
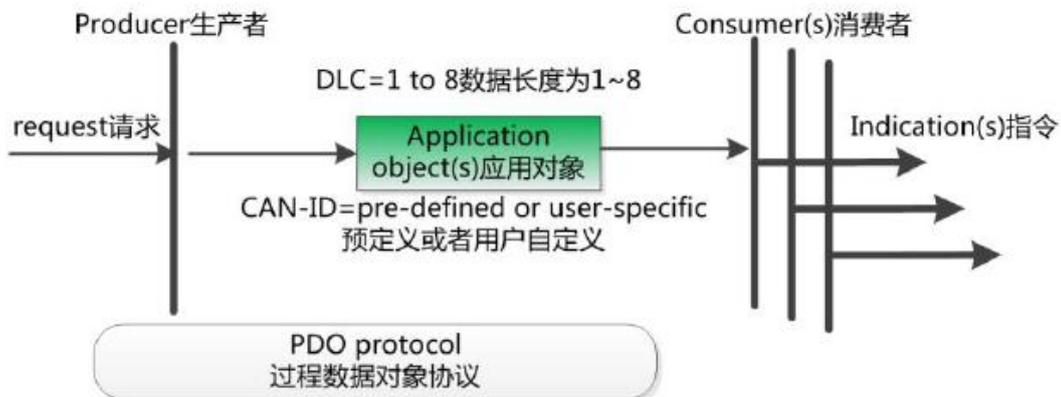


图 B.3 生产者消费者模型

- 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。

B.5 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）。

销售与服务

沈阳广成科技有限公司

地址：辽宁省沈阳市浑南区长青南街 135-21 号 5 楼

邮编：110000

网址：www.gcgd.net

淘宝官方店：<https://shop72369840.taobao.com/>

天猫官方店：<https://gcan.tmall.com/>

京东官方店：<https://mall.jd.com/index-684755.html>

全国销售与服务电话：400-6655-220

售前服务电话与微信号：13889110770

售前服务电话与微信号：18309815706

售后服务电话与微信号：18609820321

售后服务电话与微信号：18609810321

