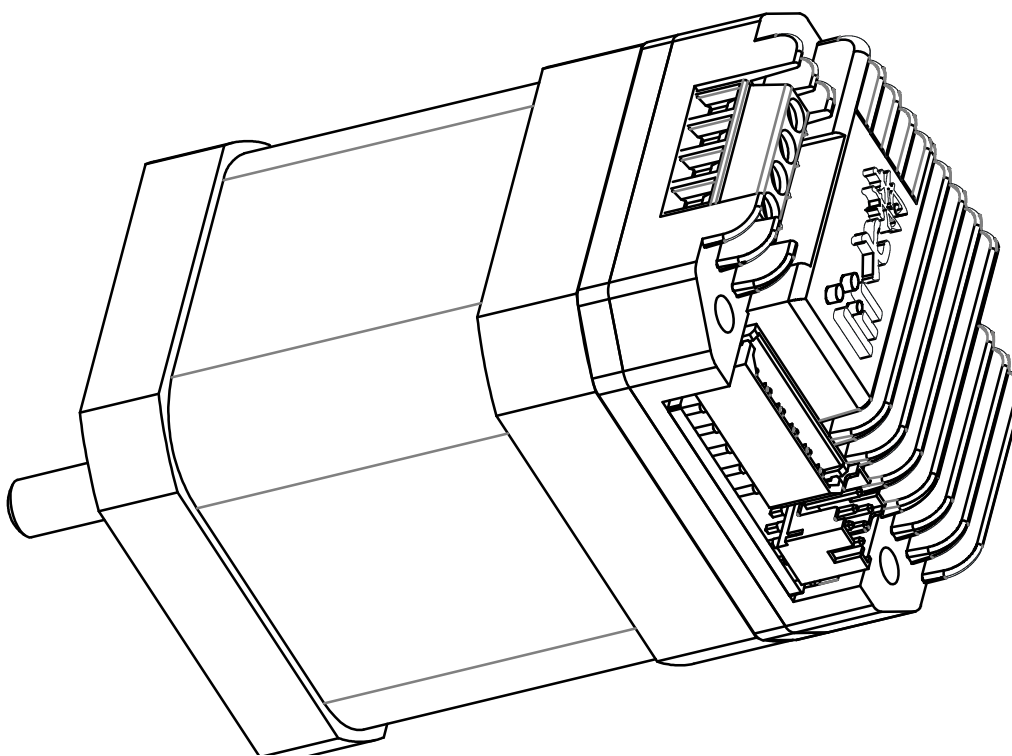


# CAN 总线接口步进闭环

## 使用说明书

(42 型：7TCSM42EXX)

| 版本      | 说明   |
|---------|------|
| Ver1.00 | 建立文档 |
|         |      |
|         |      |



## 1. 产品特点

- ☆ 专利产品，电机的速度和扭矩与电机参数匹配
- ☆ 微型设计，电机、编码器、驱动控制一体，电机不同速度对应明确扭矩
- ☆ 闭环控制，不丢步，伺服的闭环特性，防负载干扰，
- ☆ 闭环控制，自动加减速，堵转报警
- ☆ 矢量驱动，低速低噪音、低发热，高速大扭矩
- ☆ 网络集散控制，CAN2.0 组网，通信口均电气隔离
- ☆ 支持定位模式、速度模式
- ☆ 零位准确，有复位时的零位脱落动作
- ☆ 限位功能，碰到限位信号自动停止
- ☆ 到位功能，边沿触发和电平触发
- ☆ 提供计算机调试软件、DLL 和嵌入式源代码，方便调试和二次开发

## 2. 产品参数

表 1: 产品参数

| 产品参数   | 产品可更改运行参数       |                             |
|--------|-----------------|-----------------------------|
| 机身尺寸   | 42mm×42mm×66mm  | 可设置 电机运行扭矩                  |
| 工作电压   | DC24V           | 可设置 电机空闲电流比(锁机扭矩)           |
| 保持扭矩   | 0.35Nm          | 可设置 CAN-ID                  |
| 电机速度   | 1Rpm-1000Rpm    | 可设置 细分 1/2/4/8/16/32/64/128 |
| 编码位数   | 32768           | 可设置 限位开关信号极性                |
| CAN 接口 | 7TCAN           | 可设置 零位开关信号极性                |
| 限位电气   | PNP 和推挽 (0~24V) | 可设置 找零速度和运行速度               |
| 零位电气   | PNP 和推挽 (0~24V) | 可设置 找零光电开关脱落步数              |
| 操作温度   | -20°~60°        | 可设置 找零最大步数                  |
| 通信接口   | 电气隔离、TVS、防雷     | 可设置 电机空闲脱机、电机方向             |

### 3. 电气接口

电气接口包含用户接口、传感器接口和电机线。用户接口是型号 KF2EDGV-3.81/4 插座，包含电源线、CAN 通信线，具体说明见表 2。传感器接口是型号 PHD-2×3 插座，包括输出电源、零位传感器、限位(到位)传感器线，具体说明见表 3。

**注意：电机线不可动。**

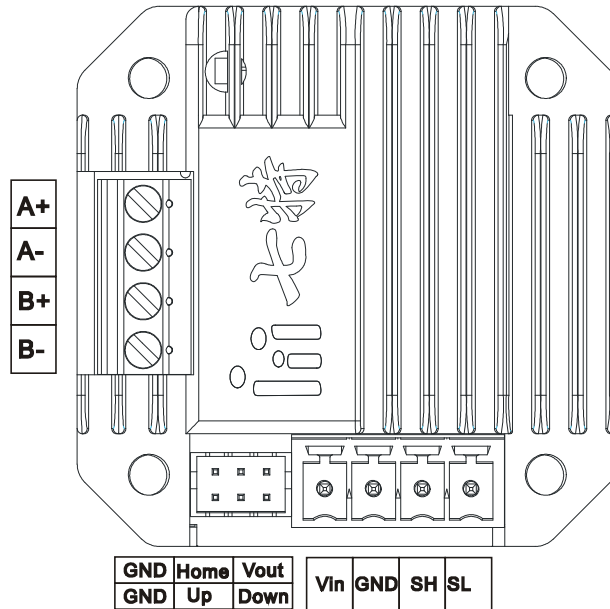


图 1: 步进一体机电气接口图

表 2: 用户接口定义说明

|             |     |                            |
|-------------|-----|----------------------------|
| KF2EDGV-3.8 | V+  | 输入电源正极，推荐 DC24V            |
|             | GND | 电源 GND                     |
|             | SH  | CAN 总线级联信号 CAN-H，与输入电源电气隔离 |
|             | SL  | CAN 总线级联信号 CAN-L，与输入电源电气隔离 |

表 3: 传感器接口定义说明

|                |      |   |
|----------------|------|---|
| 传感器插座: PHD-2×3 | Vout | 默认 Vout=V+; 可定制选择 Vout= 5V@30mA. 为传感器供电。                                |
|                | Home | 零位开关输入端 PNP 型，支持 0-24V。推荐接光电开关  |
|                | UP   | 正限位开关输入端 PNP 型，支持 0-24V。<br>定位、正反转、速度模式下：正反向限位信号<br>到位模式下：正反向或者反方向的到位信号 |
|                | DW   | 正限位开关输入端 PNP 型，支持 0-24V。<br>定位、正反转、速度模式下：反反向限位信号<br>到位模式下：正反向或者反方向的到位信号 |
|                | GND  | 电源 GND，与输入电源同一个 GND   |

## 4. 速度扭矩

常规步进电机驱动器在设定的某个电流下运行。在设定电流下，电机速度不同输出的扭矩不同。本专利产品，根据电机运行的不同速度，匹配不同的电流，因此实现了低速低噪音，低发热。步进一体机(步进电机、驱动、控制三合一)带动恒定阻尼负载运行起来，不同速度下提供的扭矩大小如表 4。

表 4：速度扭矩

| 速度(单位 rpm) | 扭矩 (Nm), 24V 供电下测试 |
|------------|--------------------|
| $\leq 200$ | 0.35               |
| 300        | 0.25               |
| 400        | 0.2                |
| 500        | 0.15               |
| 600        | 0.1                |
| 700        | 0.05               |

## 5. 运行及工作模式说明

### 5.1 闭环步进电机驱动控制介绍

步进电机特性：步进电机转速越大，所能提供的扭矩变小。

一般的机械机构中，步进电机运行，在初始时(机械机构静止时)，为克服摩擦力而需要步进电机提供较大的扭矩。机械负载动起来后因机械惯性，需要步进电机提供的扭矩变小。因此，步进电机的启动需要加速过程。

在步进电机加速运行起来后，如果碰到外力干扰，则会丢步。步进闭环不因某刻丢步而出现整个结果的误差。也可以在高速的时候提供较大扭矩输出。

### 5.2 步进一体机(步进电机、驱动、控制三合一)工作模式

工作模式：定位模式、正反转模式、速度模式、到位模式。不同的工作模式无需手动设置，步进一体机根据用户发送不同指令，自动切换到对应的工作模式。

步进一体机运行位置数据值变大方向，定义为步进一体机的正方向；步进一体机运行位置数据值变小方向定义为步进一体机的反方向。步进一体机运行方向可软件设定，而应对零位传感器安装在不同位置。

### 5.3 定位模式

定位模式：定位模式包含复位和定位。用户的机械行程，映射成一个单轴的坐标。通过标度变换，可以将用户的机械行程单位映射为一个有起点和终点的步数坐标。

复位(找零，找 home)：即找机械零点位置。复位需要一个外部零位传感器安装在机械机构上，配合步进一体机共同完成。

执行复位时，零位传感器信号为无效状态时：见图 2 所示。步进一体机反方向运行找零最大步数，在小于**找零最大步数**的运行过程中，若检测到零位传感器信号，则完成复位。如果步进一体机反转找零最大步数运行完成后，无法检测到零位传感器信号，则停止运行，并给出找零错误的报警信号，通过 CAN 接口可读取报警信号。因此，通过调试，可将找零最大步数设置成稍大于机械机构行程步数值即可。

执行复位时，零位传感器信号为有效状态时：见图 3 所示。步进一体机正方向运行**开关脱落步数**，然后读取零位传感器信号。若零位传感器信号无效状态，则步进一体机则反转复位；如果零位传感器信号还是有效状态，则给出找零错误的报警信号。通过 CAN 接口可读取报警信号。因此，通过调试，可将开关脱落步数设置成稍大于零位传感器有效信号的行程步数值。

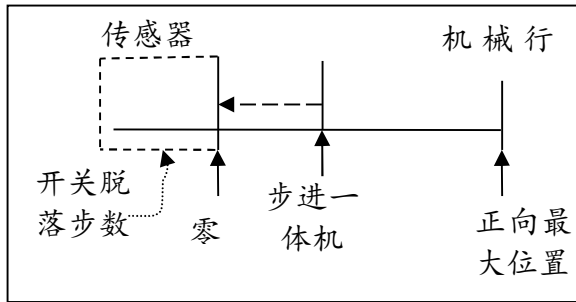


图 2: 复位时, 零位信号无效

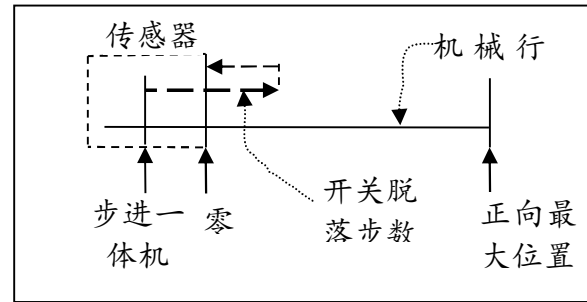


图 3: 复位时, 零位信号有效

每次重新上电后, 都需要进行一次复位操作。零位传感器的步进一体机反方向末端位置, 需要覆盖用户有效的机械结构行程

定位: 复位正确后才可定位。复位完成后, 机械机构处于机械零位的位置。通过通信接口发送步进一体机需要去的步数坐标(目标位置), 步进一体机会根据当前位置自动计算运行方向和步数。步进一体机接收到新的定位指令, 如果此时还没有完成上一次的定位指令, 则会将新指令的目标位置作为最终的目标位置, 不再执行上一指令的目标位置。

步进一体机在定位的过程中, 还没有达到目标位置时, 正方向运行碰到了 UP 限位开关或者反方向运行碰到负 DW 限位开关, 步进一体机则减速停止并给出报警信号。通过 CAN 接口可读取报警信号。此时, 系统硬件可能出错, 需要用户的系统报警并检修处理, 步进一体机不响应定位指令。

步进一体机工作在定位模式时, 可以执行变速命令。接收到变速指令后, 如果此时没有运行到目标位置, 则自动通过加减速调整到最新速度; 如果此时是在目标位置, 在新的目标位置指令后, 则按设置的最新速度运行。

## 5.4 正反转模式

正反转模式下, 步进一体机不需要零位的传感器, 限位传感器可选。正反转模式下, 步进一体机的位置值仅能作为参考。步进一体机在接受到正转指令, 则在当前位置正方向运行指定的步数。步进一体机在接受到反转指令, 则在当前位置正方向运行指定的步数。步进一体机, 在正反转模式下, 正方向运行碰到了 UP 限位开关或者反方向运行碰到负 DW 限位开关, 步进一体机则减速停止并给出报警信号。通过 CAN 接口可读取报警信号。

正反转模式下, 步进一体机碰到限位信号停止后, 可以继续再执行正反转指令。在继续执行新的指令时, 如果对应的限位信号还存在, 虽执行指令, 步进一体机因限位信号的存在而不会动作。

举例, 步进一体机正方向运行碰到 UP 信号后, 给出报警信号。当再接收到正方向运行指令, 此时如果限位信号存在, 步进一体机不动作, 此时如果限位信号不存在, 则可继续执行正

方向运行指令；当再接收到反方向运行指令，则直接反方向运行。在定位模式下，这两种情况，步进一体机都不动作。

## 5.5 速度模式

速度模式下，步进一体机不需要零位的传感器，限位传感器可选。速度模式和正反转模式基本一样。正反转模式下，当正转步数为 0 时，步进一体机会一直正转下去，自动切换成为速度模式下的正转；当反转步数为 0 时，步进一体机会一直反转下去，自动切换成为速度模式下的反转。速度模式下，正转碰到 UP 信号和反转碰到 DW 信号，步进一体机则减速停止并给出报警信号。通过 CAN 接口可读取报警信号。

## 5.6 到位模式

在定位模式、正反转模式、速度模式下，步进一体机的 UP 为正转的限位信号，DW 信号为反转的限位信号。发送定位到位、正反转到位指令后，UP 信号和 DW 信号自动切换成到位信号，不做限位信号用。

## 5.7 调试步骤(供参考)

第一步：接好电源和通信线后，使用调试软件或者自发指令，发送给步进一体机(不带机械负载)正转一个步数，查看电机是否转动，转动为正常状态。

第二步：调试电机方向。步进一体机接入机械机构，发送速度模式的正转步数，步数值先为一个小值，避免机械结构碰撞。查看零位开关是否在反转方向末端，方向相反则设置步进一体机的方向反置。查看机械结构是否能动起来，如果动不起来，则将一体机的速度设小(小于等于 200Rpm)避免扭矩不够。

第三步：调试零位传感器信号。如果需要一体机工作在定位模式下，则接上零位传感器，保证复位速度小于等于 200rpm，发送步进一体机复位指令，此时注意随时可断驱动器电源，以免出现参数不对而碰撞机械机构。发送复位指令后，人为遮挡或者模拟零位信号有效的动作(比如槽型光电开关可以用纸片挡住槽)，看步进一体机是否停下。通过通信接口读取电机状态为复位正确的状态，即空闲状态。如果不能停下，检查线路、光电开关的信号匹配和逻辑状态是否需要反置。人为遮挡或者模拟零位信号有效后，让机械机构自动完成复位功能。此处可能因为找零最大步数步进一体机的参数不合理，需要多次执行复位指令。

第四步：初步调试运行速度。发送速度模式下的正反转指令，正反转的步数由小变大，调试其速度对应是否能将机械机构运行起来，大概运行顺畅即可。可初步调试出，复位所需要的找零最大步数、开关脱落步数。

第五步：调试限位信号。在有限位开关信号情况下，发送速度模式下的正反转指令，人为遮挡或者模拟限位信号有效的动作，看步进一体机是否停下。通过通信接口读取电机

状态是否为警报信号。如果不能停下，检查线路、光电开关的信号匹配和逻辑状态是否需要反置。

第六步，重新多次调整步进一体机参数，包括细分、静态电流百分比、找零速度、运行速度、找零最大步数、光电开关脱落步数、运行电流比等。静态电流百分比范围 10%-100%，在保证步进一体机静止空闲时，可以锁住机械结构负载的力满足使用的情况下，尽量调小。减小静态电流百分比可让步进一体机静止空闲时，少发热甚至不发热。运行电流比百分比范围 30%-100%，保证运行时不卡顿情况下，尽量调小，少发热、低振动和低噪音。

## 5.8 步进一体机参数说明

**CAN-ID:** CAN5 通信接口为总线并联系统，并联总线的 ID 需要不同

**细分:** 因矢量控制步进电机的线圈 A 和 B，将 A 和 B 线圈分别给不同电流，则能将步进电机输出轴矢量合力在步进电机步距角的一个夹角，最终实现一个步距角分几步走完，多少步走完一个步距角则为多少细分。理论上细分可以是任意数值，实际有限制。约定，一个脉冲走一步。如果步进电机步距角是 1.8 度，如果不细分下，步进电机转动一圈为 200 步；如果细分，步进电机转动一圈，需要走的步数为  $200 \times \text{细分}$ 。产品默认为 32 细分，细分参数只是在电机转一圈的步数时用到，它不影响电机输出的速度能力和扭矩能力，尽量保持不变。

**限位开关信号极性:** 限位传感器漏极有效下为高电平，无效为悬空状态。限位开关常规高电平触发有效，特殊情况可设置高电平为无效。

**零位开关信号极性:** 零位传感器漏极有效下为高电平，无效为悬空状态。零位开关常规高电平触发有效，特殊情况可设置高电平为无效。

**找零速度:** 电机在找零的时候使用的速度，单位 rpm。在满足需求情况下，找零速度尽量小，速度越小，在零点开关触发的瞬间晃动小，对应的零点位置更精确。

**运行速度:** 对应运行速度单位 rpm，根据需求设置。

**电机空闲脱机:** 电机停止后，是否锁机。锁机，步进一体机保持一定的力矩输出；不锁机，步进一体机输出轴和不接电源一样的机械状态。

**电机空闲电流比:** 在某些机械结构中，步进一体机静止时，不需要最大的保持力矩输出，降低静止时步进一体机的电流，减小发热和噪音，提高寿命。

**找零最多步数:** 详见 5.3 找零介绍。

**零位开关脱落步数:** 详见 5.3 找零介绍。

**降电流 Rpm:** 表 4 为速度扭矩曲线。在低速扭矩输出大，在不需要这么大扭矩输出的情况下，可以在 0-降电流 Rpm 范围降低电流运行。

**运行降电流比:** 表 4 为速度扭矩曲线，对应降电流比为 1。在低速扭矩输出大，在不需



要这么大扭矩输出的情况下，可以在 0-降电流 Rpm 范围降低电流运行 0.3-1 之间，值越小降电流降得越多。

**电机状态、硬件配置：** 详见表 M01，M02

表 M01：电机状态

| 数值       | 说明  |
|----------|---|
| 0xff     | 开机未定义状态。当不是 0xff 时，各 bit 位表示不同意思  |
| Bit[2:0] | 0x0: 空闲状态，不属于警报状态<br>0x1-7: 运行状态，不属于警报状态  |
| Bit[3]   | 1:在复位中; 0: 不在复位中  |
| Bit[7:4] | 0x1: 复位状态出错，零位开关未检测，警报状态<br>0x2: 复位状态出错，复位时触发 UP 信号，警报状态<br>0x3: 复位状态出错，复位时触发 DW 信号，警报状态<br>0x6: 正转时触发 Up 开关<br>0x7: 反转时触发 Down 开关<br>0x8: 堵转，负载扭矩超出电机设定速度的能输出的扭矩 |

表 M02：硬件配置

| 数值     | 说明(默认值:0x000)                     |
|--------|-----------------------------------|
| Bit0   | Up 开关逻辑配置。0: 低电平触发; 1: 高电平触发      |
| Bit1   | Dw 开关逻辑配置。0: 低电平触发; 1: 高电平触发      |
| Bit2-3 | 备用                                |
| Bit4   | 空闲锁机。0:电机状态空闲锁机; 1: 电机状态空闲不锁机     |
| Bit5   | 备用                                |
| Bit6   | 零位信号逻辑配置。0: 高电平触发; 1: 低电平触发       |
| Bit7   | 电机方向反置                            |
| Bit8   | 找零过程，碰触 DW 信号，报警停止运行。0: 不起用; 1 起用 |
| Bit9   | 找零过程，碰触 Up 信号，报警停止运行。0: 不起用; 1 起用 |

## 6. 典型使用及接线图

### 6.1 典型组网

☆ CAN 总线与计算机网关（以太网、USB、RS232 等转 CAN）连接使用

☆ CAN 总线与嵌入式控制器连接使用

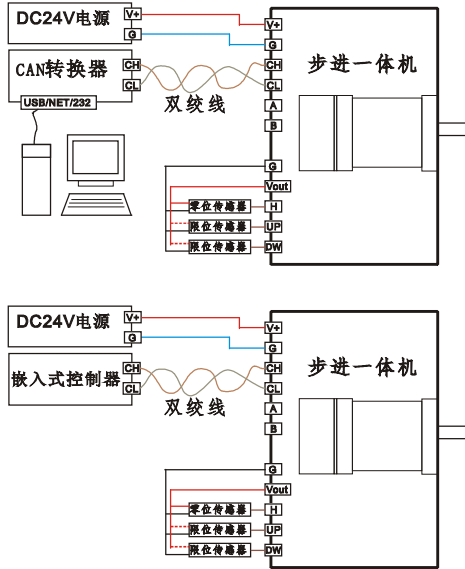


图 4：CAN 总线接线图

### 6.2 传感器接口典型接线图

推荐使用一个光电开关（输出 PNP 型、或者推挽输出）用于做精准的参考零位，两个限位用限位开关（行程开关、微动开关）等机械开关。如图 6、7 所示。

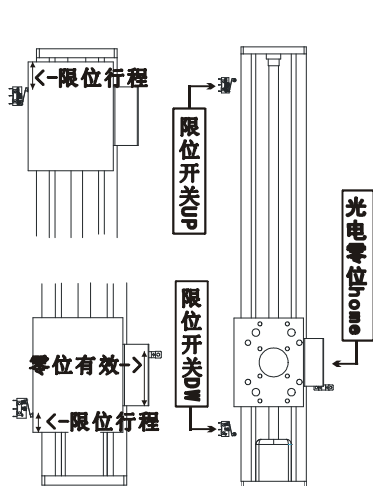


图 6：传感器安装位置参考

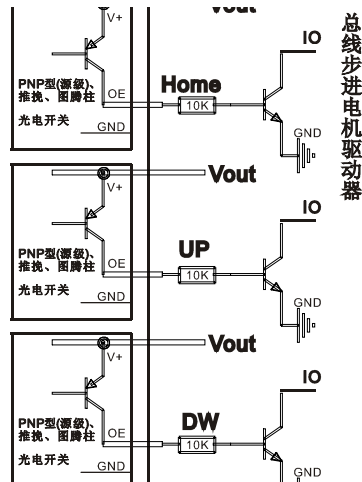


图 7：传感器接口参考接线图

在定位模式下，两个限位开关为可选，零位光电开关是必选项。

## 7. CAN 总线控制命令说明

### 7.1 CAN 总线数据帧说明

CAN 总线协议为 CAN2.0A, 波特率 125K。驱动器的地址为:0x0C1~0x0ff, 共 63 个地址, 默认为 0xC1。推荐个数小于等于 20 个, 通过通信修改地址。

表 5: CAN 总线硬件帧结构(辅助信息忽略)

|                   |     |             |
|-------------------|-----|-------------|
| CAN. ID-OBJ (地址帧) | 辅助帧 | 8 字节数据(数据帧) |
|-------------------|-----|-------------|

CAN 总线上挂着的设备, 都有一个自己的设备地址, 称为“本机 CAN. ID”。下述用 CAN. ID 表示。表 5 中的 CAN. ID-OBJ: 目标 CAN 设备地址, 这是硬件帧结构, 在 CAN2.0A 中是 11bits 位。在 CAN 总线上挂着的设备, 某个设备(CAN. ID)发送一帧硬件 CAN 信息, 发给谁呢? 就是发给总线上挂着的设备中的地址为 CAN. ID= CAN. ID-OBJ 的 CAN 设备。

### 7.2 数据帧(8 字节)说明

8 字节数据(数据帧)分为四部分, 表 6 所示的“CAN. ID-OBJ ”同表 5。8 字节数据(数据帧), 分段重新自定义, 为软件协议帧。

表 6: CAN 总线数据帧结构

|            |                      |       |     |     |
|------------|----------------------|-------|-----|-----|
| CAN.ID-OBJ | 8 字节数据(数据帧) [64bits] |       |     |     |
|            | CAN.ID               | 功能码类型 | 功能码 | 数据码 |

**CAN. ID:** 发送数据的 CAN 设备本机的 CAN 地址。举例, CAN 主控制器或者 CAN 网关(假设 CAN. ID=0x01)发送 CAN 帧给 CAN 总线步进一体机(地址为 0xC1), 则 CAN. ID-OBJ=0xC1, CAN. ID=0x01; 步进一体机(地址为 0xC1)发送 CAN 帧给 CAN 主控制器或者 CAN 网关(假设 CAN. ID=0x01), 则 CAN. ID-OBJ=0x01, CAN. ID=0xC1。

**功能码类型:** CAN 总线系统中, 每个 CAN 设备均可主动发送数据, 但是在使用中, 一般定义 CAN 主控制器或者 CAN 网关为主机(简称:CAN 主控), CAN 总线步进一体机和其它 CAN 设备为从机(简称:CAN 从机)。CAN 主控发送给 CAN 从机, 功能码类型一般取值 0x01, 详细值见表 7。CAN 从机发送给 CAN 主控, 无论是 CAN 从机应答 CAN 主机还是 CAN 从机主动发送数据给 CAN 主控, 功能码类型一般取值 0x02, 其它值详细见表 7 (比如 CAN 从机对应的步进一体机不支持 CAN 主机发送来的命令, 功能码类型值取 0x05)。

**功能码:** 也即指令码, 不同的功能码, 完成不同的任务。详细见 7.3, 命令码(功能码)及参数说明。

**数据码:** 数据码总共 5 个字节。根据不同的命令码, 对应不一样含义, 配合功能码完成一些参数的读取、动作的完成。

表 7: CAN 总线数据帧结构

| 字节 | 说明  |                      |                 |                   | 软件码                                   |
|----|---|----------------------|-----------------|-------------------|---------------------------------------|
| 0  | 本机 CAN 总线地址(11bits)的高 8bits   |                      |                 |                   | CAN.ID                                |
| 1  | Bit[7-5]: 本机 CAN 总线地址(11bits)的低 3bits<br>Bit[4-0]: 数据帧序列,当 Bit[4-0]=0 为一个命令最后一帧数据:一般设置为 0。                                    |                      |                 |                   |                                       |
| 2  | Bit[7:5]: 命令码(功能码)类型<br>0x00 广播命令(不需要返回)<br>0x01 请求命令(需要返回)<br>0x02 请求命令的正确返回<br>0x03 请求命令的不正确返回<br>0x05 无此命令码<br>0x06 命令参数错误 |                      |                 |                   | 功能码<br>类型                             |
|    | Bit[4:0]: 命令码(功能码), 详细见命令码(功能码)说明表  |                      |                 |                   | 功能码                                   |
|    | 字符型   | 半字型                  | 字型              | 浮点型               | 数据码说明:<br>在传输不同参数时,<br>用的数据类型不一<br>致。 |
| 3  | char[0]   | ShortDate[0]<br>小端模式 | IntDate<br>小端模式 | FloatDate<br>小端模式 |                                       |
| 4  | char[1]   |                      |                 |                   |                                       |
| 5  | char[2]   | ShortDate[1]<br>小端模式 |                 |                   |                                       |
| 6  | char[3]   |                      |                 |                   |                                       |
| 7  | char[4]   | 字节: ByteData         |                 |                   |                                       |

### 7.3 命令码(功能码)及参数说明

命令码 0x00: 通信测试命令

| 命令码  | CAN 总线说明  |
|------|---|
| 0x00 | 通信测试命令, CAN 主控发送给 CAN 从机时:<br><ul style="list-style-type: none"> <li>●命令码: 0x00</li> <li>●命令码类型: 0x01, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码: 任意值</li> </ul> CAN 从机返回或者主动发送给 CAN 主控时, 数据帧格式:<br><ul style="list-style-type: none"> <li>●命令码: 0x00</li> <li>●命令码类型: 0x02, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回为“字型”。IntDate 表示当前步数, ByteData 表示电机状态。</li> </ul> |

命令码 0x01: 电机复位

| 命令码  | CAN 总线说明  |
|------|---|
| 0x01 | 电机复位(电机定位模式), CAN 主控发送给 CAN 从机时:<br><ul style="list-style-type: none"> <li>●命令码: 0x01</li> <li>●命令码类型: 0x01, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码 ByteData = 1, IntDate 忽略</li> </ul> CAN 从机返回或者主动发送给 CAN 主控时, 数据帧格式:<br><ul style="list-style-type: none"> <li>●命令码: 0x01</li> <li>●命令码类型: 0x02, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回为“字型”。IntDate 表示当前步数, ByteData 表示电机状态。</li> </ul> |

## 命令码 0x02: 电机定位

| 命令码  | CAN 总线说明   |
|------|--|
| 0x02 | <p>电机定位，数据码为“字型”。CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x02</li> <li>●命令命类型：0x01, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码（如下前提：步进一体机正确复位情况下） <ul style="list-style-type: none"> <li>-&gt;ByteData = 1, IntDate 为定位的步数坐标, 执行<b>定位模式</b>的定位功能。到达目标后，主动发送命令给 CAN 主控。正向运行碰到 UP 开关，停下警报。反向运行碰到 DW 开关，停下警报。</li> <li>-&gt;ByteData = 6, IntDate 为定位的步数坐标, 执行<b>到位模式</b>。碰到 UP 有效边沿信号停止，碰不到信号则运行到目标坐标位置。到达目标或者碰到 UP 停止后，主动发送命令给 CAN 主控。</li> <li>-&gt;ByteData = 7, IntDate 为定位的步数坐标, 执行<b>到位模式</b>。碰到 DW 有效边沿信号停止，碰不到信号则运行到目标坐标位置。到达目标或者碰到 DW 停止后，主动发送命令给 CAN 主控。</li> </ul> </li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时，数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x02</li> <li>●命令命类型：0x02, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回为“字型”。IntDate 表示当前步数，ByteData 表示电机状态。</li> </ul> |

## 命令码 0x03: 电机正传

| 命令码  | CAN 总线说明  |
|------|---|
| 0x03 | <p>电机正传，数据码为“字型”。CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x03</li> <li>●命令命类型：0x01, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码 <ul style="list-style-type: none"> <li>-&gt;ByteData= 3, 无论当前电机状态如何，在当前的位置正转 IntDate(IntDate!=0)步。步进一体机执行<b>正反转模式</b>的正转功能。到达目标后，主动发送命令给 CAN 主控。IntDate=0, 步进一体机执行速度模式下的正转。无论 IntDate 何值，正向运行碰到 UP 开关，停下警报。执行此命令一次后，步进一体机状态强制变为“正确复位”，但此时的步数值和机械机构尺寸没有对应关系。</li> <li>-&gt;ByteData = 6, 步进一体机正确复位情况下，在当前的位置正传 IntDate，执行<b>到位模式</b>。碰到 UP 有效边沿信号停止，碰不到信号则运行到目标坐标位置。到达目标或者碰到 UP 停止后，主动发送命令给 CAN 主控。</li> <li>-&gt;ByteData = 7, 步进一体机正确复位情况下，在当前的位置正传 IntDate，，执行<b>到位模式</b>。碰到 DW 有效边沿信号停止，碰不到信号则运行到目标坐标位置。到达目标或者碰到 DW 停止后，主动发送命令给 CAN 主控。</li> </ul> </li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时，数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x03</li> <li>●命令命类型：0x02, 详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回为“字型”。IntDate 表示当前步数，ByteData 表示电机状态。</li> </ul> |

## 命令码 0x04: 电机正传

| 命令码  | CAN 总线说明   |
|------|--|
| 0x04 | <p>电机正传，数据码为“字型”。CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x04</li> <li>●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码 <ul style="list-style-type: none"> <li>-&gt;ByteData= 3, 无论当前电机状态如何，在当前的位置反转 IntDate(IntDate!=0)步。步进一体机执行<b>正反转模式</b>的反转动传功能。到达目标后，主动发送命令给 CAN 主控。</li> <li>IntDate=0, 步进一体机执行速度模式下的反转。无论 IntDate 何值，正向运行碰到 DW 开关，停下警报。执行此命令一次后，步进一体机状态强制变为“正确复位”，但此时的步数值和机械机构尺寸没有对应关系。</li> <li>-&gt;ByteData = 6, 步进一体机正确复位情况下，在当前的位置反转 IntDate, 执行<b>到位模式</b>。碰到 UP 有效边沿信号停止，碰不到信号则运行到目标坐标位置。到达目标或者碰到 UP 停止后，主动发送命令给 CAN 主控。</li> <li>-&gt;ByteData = 7, 步进一体机正确复位情况下，在当前的位置反转 IntDate, , 执行<b>到位模式</b>。碰到 DW 有效边沿信号停止，碰不到信号则运行到目标坐标位置。到达目标或者碰到 DW 停止后，主动发送命令给 CAN 主控。</li> </ul> </li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时，数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x04</li> <li>●命令命类型：0x02,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回为“字型”。IntDate 表示当前步数，ByteData 表示电机状态。</li> </ul> |

## 命令码 0x05: 电机停止

| 命令码  | CAN 总线说明   |
|------|--|
| 0x05 | <p>电机停止当前的运行(速度模式、定位模式、正反转模式、到位模式)，CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x05</li> <li>●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码 <ul style="list-style-type: none"> <li>-&gt;ByteData= 1: 减速停止</li> <li>-&gt;ByteData= 2: 立即停止</li> </ul> </li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时，数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x05</li> <li>●命令命类型：0x02,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回为“字型”。IntDate 表示当前步数，ByteData 表示电机状态。</li> </ul> |

## 命令码 0x06: 改变步进一体机当前速度

| 命令码  | CAN 总线说明  |
|------|---|
| 0x06 | <p>改变当前的电机速度， CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x06</li> <li>●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码 <ul style="list-style-type: none"> <li>-&gt;ByteData= 0: 数据码为“浮点型”。 FloatDate 表示目标 Rpm(1.0-1000.0)</li> <li>-&gt;ByteData= 1: 数据码为“字型”。 表示速度脉冲周期[us]</li> </ul> </li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时， 数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x06</li> <li>●命令命类型：0x02,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回： <ul style="list-style-type: none"> <li>ByteData = 0, 数据码为“浮点型”， 实际执行的 Rpm。</li> <li>ByteData = 1, 数据码为“字型”， 实际执行的脉冲周期值[us]。</li> </ul> </li> </ul> |

## 命令码 0x13: 修改步进一体机参数

| 命令码  | CAN 总线说明  |
|------|---|
| 0x13 | <p>改变当前的电机速度， CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x13</li> <li>●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码：详见表 8</li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时， 数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x13</li> <li>●命令命类型：0x02,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码返回：详见表 8</li> </ul> |

## 命令码 0x14: 保存参数

| 命令码  | CAN 总线说明  |
|------|---|
| 0x14 | <p>改变当前的电机速度， CAN 主控发送给 CAN 从机时：</p> <ul style="list-style-type: none"> <li>●命令码：0x14</li> <li>●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码 <ul style="list-style-type: none"> <li>-&gt;ByteData= 254: 将参数至于默认参数存于内部 FLASH。</li> <li>-&gt;ByteData=0: 将 0x13 命令改变的参数存于内部 FLASH。</li> </ul> </li> </ul> <p>CAN 从机返回或者主动发送给 CAN 主控时， 数据帧格式：</p> <ul style="list-style-type: none"> <li>●命令码：0x14</li> <li>●命令命类型：0x02,详见表 7 的字节 2(Bit[7:5])</li> <li>●数据码：忽略值</li> </ul> |

表 8: 0x13 的数据码

| ByteData | 说明(电机参数命令码 0x13)                          |
|----------|---|
| 4        | 字型; 设置找零步数, IntDate 为步数参数 [1~0x7ffffff]   |
| 5        | 字型; 读取找零步数, IntDate 忽略                    |
| 6        | 字型; 设置脱落步数, IntDate 为步数参数 [1~0x7ffffff]   |
| 7        | 字型; 读取脱落步数, IntDate 忽略                    |
| 8        | 字型; 设置模块 CAN.ID, IntDate 为 ID [0xC1~0xff] |
| 9        | 字型; 读取模块 ID, IntDate 忽略                   |
| 10       | 字型; 设置细分, IntDate 为[1 2 4 8 16 32 64 128] |
| 11       | 字型; 读取细分, IntDate 忽略                      |
| 12       | 字型; 设置步进一体机的硬件参数, 详见见表                    |
| 13       | 字型; 读取进一体机的硬件参数, IntDate 忽略               |
| 24       | 字型; 设置待机电流的百分比 10-100                     |
| 25       | 字型; 读取待机电流的百分比                            |
| 40       | 浮点型; 设置开机默认运行速度 Rpm[1~1000]               |
| 41       | 浮点型; 读取开机默认运行速度 Rpm[1~1000]               |
| 42       | 浮点型; 设置复位/找零速度 Rpm[1~1000]                |
| 43       | 浮点型; 读取复位/找零速度 Rpm[1~1000]                |
| 44       | 浮点型; 设置减电流比 0.3-1                         |
| 45       | 浮点型; 读取减电流比                               |
| 46       | 浮点型; 设置降电流截止速度 rpm                        |
| 47       | 浮点型; 读取降电流截止速度 rpm                        |

表 13 所述: 设置参数时, CAN 主控发送给 CAN 从机的 IntData/FloatData 为设置的值, CAN 从机返回给 CAN 主控的为步进一体机的实际设置值; 读取参数时, CAN 主控发送给 CAN 从机的 IntData/FloatData 值忽略, CAN 从机返回给 CAN 主控的为步进一体机的当前值。

## 7.4 举例说明

CAN 主控发送正传 1000 步命令给步进一体机: 功能码 03、功能码类型 01、数据码前 4 字节=1000、数据码第 5 字节=3

表 9: CAN 主控发送正传 1000 步协议分解

| CAN.ID-OBJ | 8字节数据(数据帧)     |       |      |         |
|------------|----------------|-------|------|---------|
|            | CAN.ID(11Bits) | 功能码类型 | 功能码  | 数据码     |
| 0xC1       | 0x001 (HEX)    | 0x01  | 0x03 | 1000 03 |



|      |                         |                  |                  |                  |    |
|------|-------------------------|------------------|------------------|------------------|----|
| 0xC1 | 00                      | 0x20[b0010 0000] | 0x23[b0010 0011] | 0x000003E8(1000) | 03 |
| 0xC1 | 00 20 23 E8 03 00 00 03 |                  |                  |                  |    |

步进一体机返回 CAN 主控：功能码 03、功能码类型 03、数据码前 4 字节=1000、数据码第 5 字节=3

表 10: CAN 从机运行 1000 后的数据返回

| CAN.ID-OBJ | 8字节数据(数据帧)                   |                  |                  |      |           |    |
|------------|------------------------------|------------------|------------------|------|-----------|----|
|            | CAN.ID(11Bits)               |                  | 类型               | 功能码  | 数据码       |    |
| 0x01       | 0x0C1[000 1100 0001]         |                  | 0x02             | 0x03 | xxxx xxxx | xx |
| 0x01       | 0x18[000 1100 0]             | 0x20[b0010 0000] | 0x43[b0100 0011] |      | xxxx xxxx | xx |
| 0x01       | 18 20 43 xx xx xx xx[电机当前位置] |                  | xx[电机状态]         |      |           |    |

## 9. 包装

| 序号 | 数量 | 部件                           |
|----|----|------------------------------|
| 1  | 1  | CAN 总线接口步进闭环 [7TCSM42E48/40] |
| 2  | 1  | PH2.0-2×3P 带线                |
| 3  | 1  | KF2EDGV-3.81/4 插件一个          |
| 4  | 1  | 包装盒一个                        |

### 10. 一体机外形尺寸图

