**CAN总线接口步进电机驱动器**

**使用说明书**

**(42型：)**

|  |  |
| --- | --- |
| 版本 | 说明 |
| Ver1.00 | 建立文档 |
| Ver1.01 | 传感器接口改变 |
| Ver2.00 | 电源和总线接口定义改变 |
| Ver2.01 | 增加变速通信协议 |
| Ver2.02 | 增加软件设置相电流功能 |
|  |  |

7TCSM421x.wmf

# 1.产品特点

☆ S加减速曲线，运行平稳，用户可更S曲线改参数

☆ 微型设计，安装便利，可与42步进电机一体化

☆ 网络集散控制，CAN2.0组网

☆ 支持定位模式、正反转模式、速度模式、到位模式

☆ 停止运行时自动减电流

☆ 电气接口简洁，且接线方便

☆ 零位准确，有复位时的零位脱落动作

☆ 限位，碰到限位信号自动停止,电平触发和边沿触发到位

☆ 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

表一:产品参数

|  |  |
| --- | --- |
| 产品参数 | 产品可更改运行参数 |
| 外观尺寸 42.2mm×42.2mm×14.5mm | 可设置 电机相电流 |
| 相电流 0.3A-1.7A连续可调 | 可设置 细分1、2、4、8、16、32 |
| 工作电压 DC12V-32V | 可设置 运动模式 |
| 步进细分 1、2、4、8、16、32 | 可设置 加速系数、启动速度和最大速度 |
| 步进频率 20Hz-20KHz可调 | 可设置 电机空闲脱机 |
| 零位电气 PNP和推挽（0-24V） | 可设置 复位光电开关脱落步数 |
| 限位电气 TTL/NPN | 可设置 电机方向 |
| CAN接口 CAN2.0A | 可设置 复位、限位触发有效电平 |
| 存储温度 -20°～85° | 可设置 最大步数(找零位开关的最多步数) |
| 保护电路 过热、过流、过载、电源反接  通信保护 电气隔离、TVS、防雷 | 可设置 CAN.ID |

# 3.电气接口

can 42电气接口正面.wmf

图1：步进一体机电气接口图

表2:用户接口定义说明

|  |  |  |
| --- | --- | --- |
| 插座:  KF2EDG-3.81 | V+ | 电源输入DC11V-DC32V |
| GND | 电源GND |
| SH | 总线级联信号CAN-H |
| SL | 总线级联信号CAN-L |

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 |
| Home | 零位开关输入端，PNP型，支持0-24V。推荐接光电开关 |
| UP | TTL电平(0-5V)、NPN信号、机械开关  定位、正反转、速度模式下：正反向限位信号  到位模式下：正反向或者反方向的到位信号 |
| DW | TTL电平(0-5V)、NPN信号、机械开关  定位、正反转、速度模式下：反反向限位信号  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 |

表4：电机线接口

|  |  |  |
| --- | --- | --- |
| 插座:  kf128-3.81 | A+ | 接步进电机的A相线圈的A+。实际也可接A-，电机方向会反置 |
| A- | 接步进电机的A相线圈的A-。实际也可接A+，电机方向会反置 |
| B+ | 接步进电机的B相线圈的B+。实际也可接B-，电机方向会反置 |
| B- | 接步进电机的B相线圈的B-。实际也可接B+，电机方向会反置 |

# 4.典型使用及接线图

4.1 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

C:\Users\Administrator\Documents\Tencent Files\2418938403\Image\C2C\0$E{DC5F6)Q73AE3WM0BG_C.png

图2：CAN总线接线图

4.2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3、4。

典型应用.wmf 典型接线图.wmf

图3：传感器安装位置参考 图4：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。

# 5.运行及工作模式说明

5.1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。不同的工作模式不需设置，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。

5.2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):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，零位传感器信号是无效状态，见图5所示。步进一体机反方向运行找零最大步数，在运行小于找零最大步数的过程中，检测到零位传感器信号，完成复位。如果步进一体机反转找零最大步数后，没有检测到零位传感器信号则停止运行，给出找零错误的报警信号。CAN接口可读取，同时CAN接口会主动返回报警信号。

执行复位时，零位传感器信号是有效状态，见图6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。如果零位传感器信号无效状态，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。CAN接口可读取，同时CAN接口会主动返回报警信号。因此，通过调试后，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。

机械行程

传感器

传感器

机械行程

正向最大位置

零位

步进一

体机

开关脱

落步数

开关脱

落步数

正向最大位置

零位

步进一

体机

图5：复位时，零位信号无效

图6：复位时，零位信号有效

每次重新上电后，都需要一次复位操作。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，则将新指令的目标位置作为最终的目标位置，不再关心上一次指令的目标位置。

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。CAN接口可读取，同时CAN接口会主动返回报警信号。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。

5.3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传指令，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传指令，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。CAN接口可读取，同时CAN接口会主动返回报警信号。

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。再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。   
**5.4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，当正传步数为0时，步进一体机会一直正传下去，自动切换成为速度模式下的正传；当反传步数为0时，步进一体机会一直反传下去，自动切换成为速度模式下的反传。速度模式下，正传碰到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。CAN接口可读取，同时CAN接口会主动返回报警信号。

5.5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为正传的限位信号，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。

5.6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)正传一个步数，查看电机是否转动，转动为正常。

第二步：调试电机方向。步进一体机接入机械机构，发送速度模式的正传步数，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转反命令，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、加速步数、加速系数、找零速度、运行速度、找零最大步数、光电开关脱落步数等。

5.7步进一体机参数说明

CAN-ID：步进一体机可同时或者用其中之一的通信接口，CAN通信接口为总线并联系统，并联总线的ID需要不同

细分：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，200步步进电机转动一圈；如果细分，步进电机转动一圈，需要走的步数为200×细分。

限位开关信号极性:限位传感器漏极有效下为低电平，无效为悬空状态。限位开关常规低电平触发有效，特殊情况可设置为低电平为无效。设置方法详细见表M02。

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，特殊情况可设置为高电平为无效。设置方法详细见表M02。

找零速度：同时也是启动速度，对应启动速度周期(单位us)，周期值越大，速度越慢，周期值越小，速度越快。对应的rpm = 300000/(周期值\*细分)。

运行速度:对应运行速度周期(单位us)，周期值越大，速度越慢，周期值越小，速度越快。对应的rpm = 300000/(周期值\*细分)。已值Rpm，对应的周期值= 300000/( Rpm\*细分)。

加速步数、加速系数：

电机运行过程（除找零时）的启动和停止有S曲线的加速和减速。加速曲线如图7

加速步数

图7：S加速曲线

S曲线参数：

启动速度脉冲周期(PlusStartTime) [单位us]

运行速度对应脉冲周期(PlusConstantTime) [单位us]

加速步数(AccSteps)

加速系数(AccCof)

根据实际负载，初次使用时，调整S曲线参数的启动速度脉冲周期，运行速度对应脉冲周期，加速系数。

※注意※：

S曲线参数需要满足一定条件，否则启动后速度和设置速度会不一致。(exp：自然对数e的x次方， log：e为低的对数)

PlusStartTime- PlusConstantTime < (exp(AccSteps\*AccCof/2)+1)\*0.95 或者

AccSteps > log(（PlusStartTime- PlusConstantTime）/0.95 -1) \*2/AccCof 或者

AccCof > log(（PlusStartTime- PlusConstantTime）/0.95 -1)\*2/AccSteps

加速时间: (PlusStartTime + PlusConstantTime)\* 加速步数\*0.5[us]

加速系数：数值越小启动越缓慢，能启动更重负载或者能让负载加速到更快速度；同时启动时间变长。

加速步数：数值越大启动越缓慢，能启动更重负载或者能让负载加速到更快速度；同时启动时间变长。

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  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  0x6: 正转时触发Up开关  0x7: 反转时触发Down开关 |

表M02：硬件配置

|  |  |
| --- | --- |
| 数值 | 说明(默认值:0x00) |
| 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 |

# 6.CAN总线控制命令说明

6.1 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。

6.2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 | | | | |
| 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  0x00 广播命令(不需要返回)  0x01请求命令(需要返回)  0x02请求命令的正确返回  0x03 请求命令的不正确返回  0x05 无此命令码  0x06 命令参数错误 | | | | | 功能码类型 |
| 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：  在传输不同参数时，用的数据类型不一致。 | 数据码 |
| 3 | char[0] | ShortDate[0]  小端模式 | IntDate  小端模式 | FloatDate  小端模式 |
| 4 | char[1] |
| 5 | char[2] | ShortDate[1]  小端模式 |
| 6 | char[3] |
| 7 | char[4] | 字节：ByteData | | |

6.3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；  ●命令码：0x00  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码：任意值  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x00  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回为“字型”。IntDate表示当前步数，ByteData表示电机状态。电机状态详细见表M01 |

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：  ●命令码：0x01  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码ByteData = 1，IntDate忽略  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x01  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回为“字型”。IntDate表示当前步数，ByteData表示电机状态。 |

命令码0x02：电机定位

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x02 | 电机定位，数据码为“字型”。CAN主控发送给CAN从机时：  ●命令码：0x02  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码（如下前提 :步进一体机正确复位情况下）  ->ByteData = 1, IntDate为定位的步数坐标,执行**定位模式**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。  ->ByteData = 6, IntDate为定位的步数坐标,执行**到位模式**。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。  ->ByteData = 7, IntDate为定位的步数坐标,执行**到位模式**。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。  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x02  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回为“字型”。IntDate表示当前步数，ByteData表示电机状态。 |

命令码0x03：电机正传

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x03 | 电机正传，数据码为“字型”。CAN主控发送给CAN从机时：  ●命令码：0x03  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码  ->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**正反转模式**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。  ->ByteData = 6, 步进一体机正确复位情况下, 在当前的位置正传IntDate，执行到位模式。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。  ->ByteData = 7, 步进一体机正确复位情况下, 在当前的位置正传IntDate，,执行到位模式。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。  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x03  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回为“字型”。IntDate表示当前步数，ByteData表示电机状态。 |

命令码0x04：电机正传

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x04 | 电机正传，数据码为“字型”。CAN主控发送给CAN从机时：  ●命令码：0x04  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码  ->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**正反转模式**的反转动传功能。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。  ->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**到位模式**。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。  ->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**到位模式**。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。  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x04  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回为“字型”。IntDate表示当前步数，ByteData表示电机状态。 |

命令码0x05：电机停止

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x05 | 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：  ●命令码：0x05  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码  ->ByteData= 1：减速停止  ->ByteData= 2：立即停止  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x05  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回为“字型”。IntDate表示当前步数，ByteData表示电机状态。 |

命令码0x06：改变步进一体机当前速度

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x06 | 改变当前的电机速度， CAN主控发送给CAN从机时：  ●命令码：0x06  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码  ->ByteData= 0：数据码为“浮点型”。 FloatDate表示目标Rpm(1.0-1000.0)  ->ByteData= 1：数据码为“字型”。 表示速度脉冲周期[us]  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x06  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回：  ByteData = 0, 数据码为“浮点型”，实际执行的Rpm。  ByteData = 1, 数据码为“字型”, 实际执行的脉冲周期值[us]。 |

命令码0x13：修改步进一体机参数

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x13 | 改变当前的电机速度， CAN主控发送给CAN从机时：  ●命令码：0x13  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码：详见表8  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x13  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码返回：详见表8 |

命令码0x14：保存参数

|  |  |
| --- | --- |
| 命令码 | CAN总线说明 |
| 0x14 | 改变当前的电机速度， CAN主控发送给CAN从机时：  ●命令码：0x14  ●命令命类型：0x01,详见表7的字节2(Bit[7:5])  ●数据码  ->ByteData=0：将0x13命令改变的参数存于内部FLASH。  CAN从机返回或者主动发送给CAN主控时，数据帧格式：  ●命令码：0x14  ●命令命类型：0x02,详见表7的字节2(Bit[7:5])  ●数据码：忽略值 |

表8：0x13的数据码

|  |  |
| --- | --- |
| ByteData | 说明(电机参数命令码0x13) |
| 0 | 字型；设置加速步数，IntDate为步数参数 [1~3000] |
| 1 | 字型；读取加速步数，IntDate忽略 |
| 4 | 字型；设置找零步数，IntDate为步数参数 [1~0x7fffffff] |
| 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 [1~0x7fffffff] |
| 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，详细见表M02 |
| 13 | 字型；读取进一体机的硬件参数，IntDate忽略 |
| 14 | 字型；设置运行速度周期，IntDate为周期us[20~5000] |
| 15 | 字型；读取运行速度冲周期，IntDate忽略 |
| 16 | 字型；设置找零速度脉冲周期，IntDate为周期us[200~5000] |
| 17 | 字型；读取找零速度脉冲周期，IntDate忽略 |
| 18 | 浮点型；设置加速系数 |
| 19 | 浮点型；读取加速系数 |
| 22 | 浮点型；设置运行电流0.3-1.7 |
| 23 | 浮点型；读取运行电流 |

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。

6.4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01、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] | | | | | |

# 7.包装

|  |  |  |
| --- | --- | --- |
| 序号 | 数量 | 部件 |
| 1 | 1 | CAN接口42型步进电机驱动器 [7TCSM4210Q] |
| 2 | 1 | PH2.0-2×3P带线 |
| 3 | 1 | KF128-3.81-4P |

# 附件一 控制器外形尺寸图

驱动器与42步进电机孔位、大小一致。

42电机驱动器1.10尺寸图_1.wmf

# 附件二 控制器安装示意图

控制器可以与42步进电机一体化安装，配有不同规格42步进电机(高度不一致)所需要的φ3螺钉2个。

与电机一体化安装时：建议保留步进电机2个螺丝。

CAN 42一体化安装图.wmf