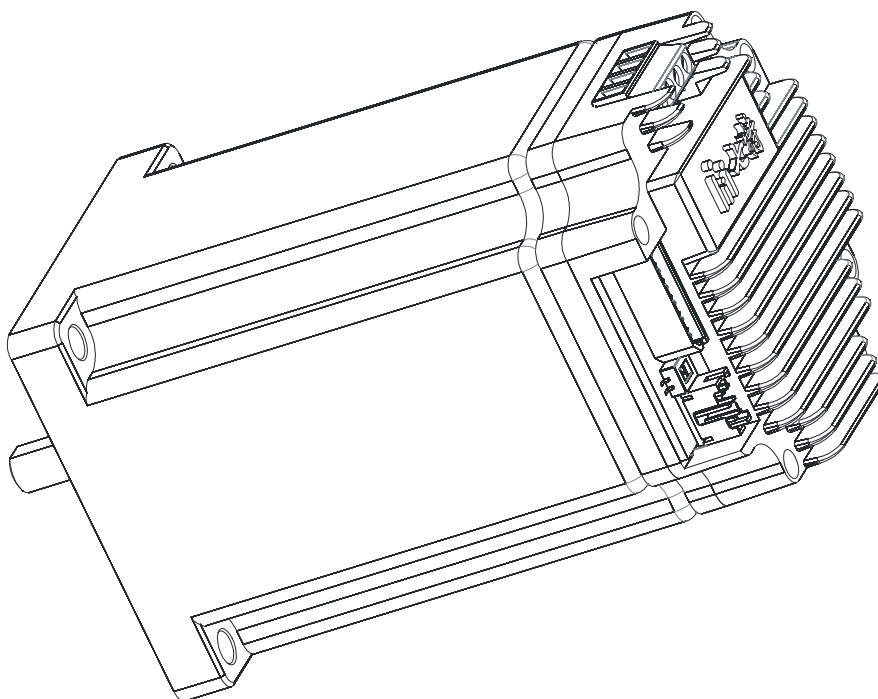


CAN、RS485 总线接口步进闭环

使用说明书

(57 型：7TCRSM57E76)

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、RS485 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	57mm×57mm×96mm	可设置 CAN-ID
工作电压	DC24V	可设置 485-ID
保持扭矩	1.2Nm	可设置 细分 1/2/4/8/16/32/64/128
电机速度	1Rpm-1000Rpm	可设置 限位开关信号极性
编码位数	32768	可设置 零位开关信号极性
CAN 接口	CAN2.0A	可设置 找零速度和运行速度
RS485 接口	ModBus-RTU	可设置 找零光电开关脱落步数
限位电气	NPN 或者 TTL(0~5V)	可设置 找零最大步数
零位电气	PNP 和推挽 (0~24V)	可设置 电机空闲脱机
操作温度	0°~85°	可设置 电机方向
通信接口	电气隔离、TVS、防雷	可设置 电机空闲电流比(锁机扭矩)

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 5267-6P 插座，包含电源线、CAN 通信线和 RS485 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。

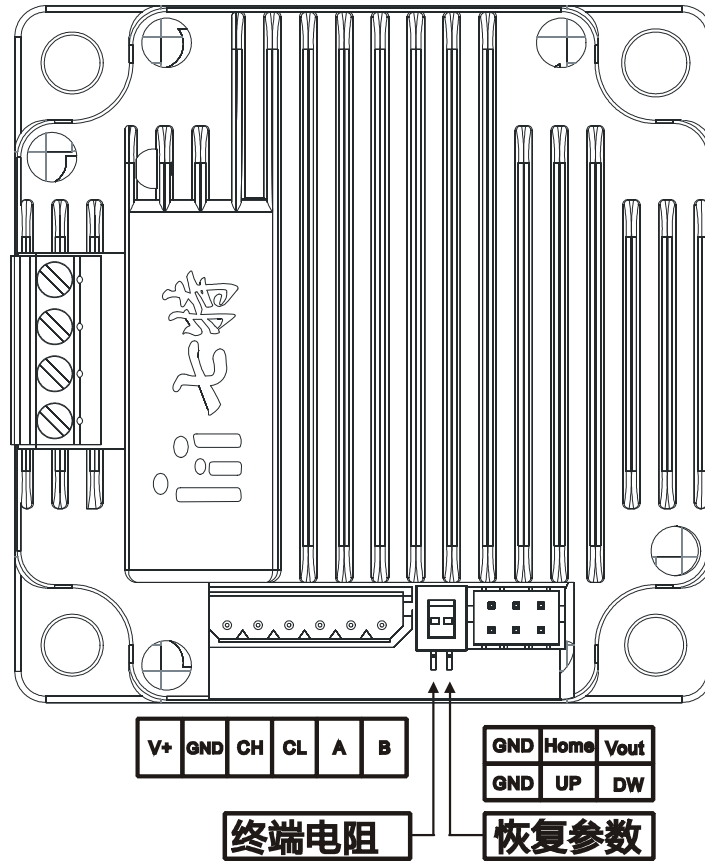


图 1: 步进一体机电气接口图

恢复参数：将拨码开关拨到步进一体机中心方向，上电后，步进一体机的参数恢复到出厂设置。此时，再将拨码开关拨到步进一体机外侧，正常使用。

表 2: 用户接口定义说明

插座: 5267-6P	V+	输入电源正极, 推荐 DC24V
	GND	电源 GND
	CH	CAN 总线级联信号 CANH, 与输入电源电气隔离
	CL	CAN 总线级联信号 CANL, 与输入电源电气隔离
	A	485 总线级联信号 A, 与输入电源电气隔离
	B	485 总线级联信号 B, 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
	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

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
<=200	1.0
300	0.7
400	0.5
500	0.4
600	0.3
700	0.25

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，速度越大，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。

5.3 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)：及找机械零点位置。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 2 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。通过 RS485 接口可读取报警信号，通过 CAN 接口可读取的同时 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 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。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。通过 RS485 接口可读取报警信号，通过 CAN 接口可读取的同时 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。

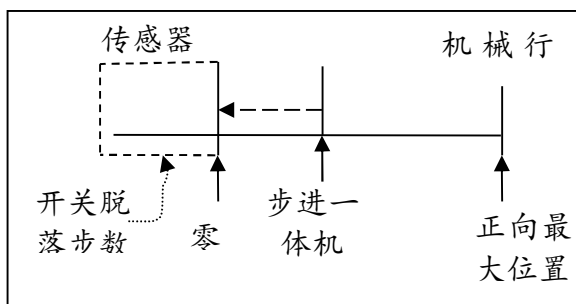


图 2: 复位时，零位信号无效

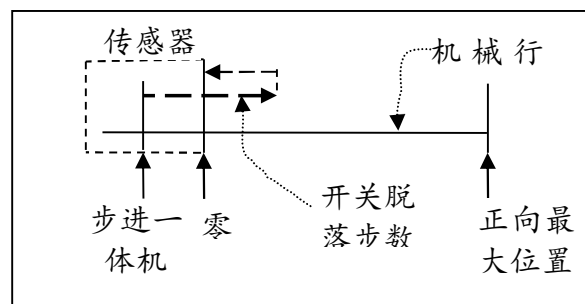


图 3: 复位时，零位信号有效

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。通过 RS485 接口可读取报警信号，通过 CAN 接口可读取的同时 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。

5.4 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。通过 RS485 接口可读取报警信号，通过 CAN 接口可读取的同时 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。

5.5 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，当正传步数为 0 时，步进一体机一直正传下去，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。通过 RS485 接口可读取报警信号，通过 CAN 接口可读取的同时 CAN 接口会主动返回报警信号。

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。

5.7 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、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。

5.8 步进一体机参数说明

CAN-ID、485-ID：步进一体机可同时或者用其中之一的通信接口，CAN 和 RS485 通信接口均为总线并联系统，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，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 $200 \times \text{细分}$ 。

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。

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。

找零速度：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 $\text{rpm} = 300000 / (\text{周期值} \times \text{细分})$ 。

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 $\text{rpm} = 300000 / (\text{周期值} \times \text{细分})$ 。已值 Rpm，对应的周期值 = $300000 / (\text{Rpm} \times \text{细分})$ 。

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。

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 0x2: 复位状态出错，复位时触发 UP 信号，警报状态 0x3: 复位状态出错，复位时触发 DW 信号，警报状态 0x6: 正转时触发 Up 开关 0x7: 反转时触发 Down 开关 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
- ☆ RS485 总线与计算机网关（以太网、USB、RS232 等转 RS485）连接使用
- ☆ CAN 总线与嵌入式控制器连接使用
- ☆ RS485 总线与嵌入式控制器连接使用
- ☆ RS485 总线与 PLC 连接使用

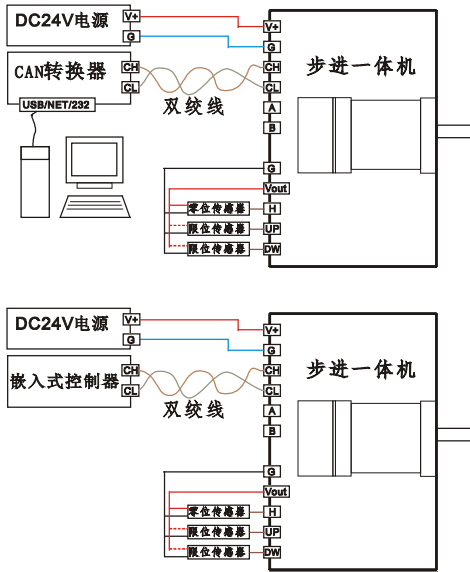


图 4: CAN 总线接线图

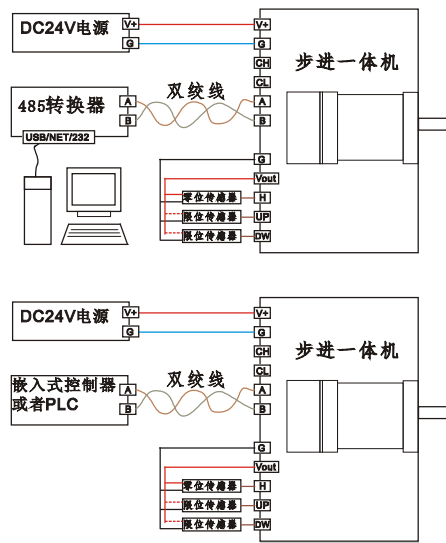


图 5: RS485 总线接线图

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。

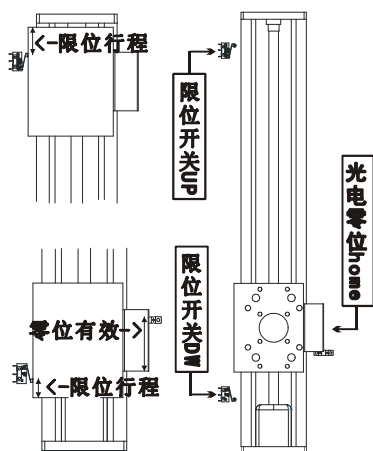


图 6: 传感器安装位置参考

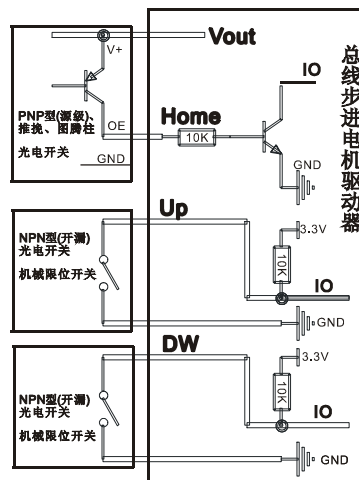


图 7: 传感器接口参考接线图

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 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。
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			

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: <ul style="list-style-type: none"> ●命令码: 0x00 ●命令码类型: 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: <ul style="list-style-type: none"> ●命令码: 0x00 ●命令码类型: 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 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: <ul style="list-style-type: none"> ●命令码: 0x01 ●命令码类型: 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: <ul style="list-style-type: none"> ●命令码: 0x01 ●命令码类型: 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	<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"> ●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) <ul style="list-style-type: none"> ->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。 <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"> ●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	<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"> ●命令码: 0x03 ●命令命类型: 0x01, 详见表 7 的字节 2(Bit[7:5]) ●数据码 <ul style="list-style-type: none"> ->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。 <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"> ●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	<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"> ●命令码：0x04 ●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5]) ●数据码 <ul style="list-style-type: none"> ->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。 <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"> ●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	<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"> ●命令码：0x05 ●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5]) ●数据码 <ul style="list-style-type: none"> ->ByteData= 1: 减速停止 ->ByteData= 2: 立即停止 <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"> ●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	<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"> ●命令码：0x06 ●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5]) ●数据码 <ul style="list-style-type: none"> ->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] <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"> ●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： <ul style="list-style-type: none"> 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	<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"> ●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 <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"> ●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	<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"> ●命令码：0x14 ●命令命类型：0x01,详见表 7 的字节 2(Bit[7:5]) ●数据码 <ul style="list-style-type: none"> ->ByteData= 254: 将参数至于默认参数存于内部 FLASH。 ->ByteData=0: 将 0x13 命令改变的参数存于内部 FLASH。 <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"> ●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)
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
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
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
26	字型; 设置堵转时间
27	字型; 读取堵转时间

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]			

8. RS485 总线控制命令说明

8.1 RS485 总线协议

硬件帧格式：1bit 起始位、位 8bit 数据位、1bit 停止位、无奇偶校验。默认波特率 9600，可设置 2400、4800、9600、19200、38400。

软件协议格式：标准的 **Modbus-RTU**。力控组态软件、ModBusPoll、sscom 测试。使用到的功能码有，读保持寄存器（功能码 03）、写单个寄存器（功能码 06）、写多个寄存器（功能码 16）。本设备在写多个寄存器时（功能码 16），一次最多只能写 2 个寄存器。本设备的变量用 1 个寄存器存数的，在写寄存器值时通过“功能码 06”候按 16bit 写；变量用 2 个寄存器存数的，在写寄存器值时通过“功能码 16”候按 32bit 写。

8.2 Modbus 寄存器值及功能码说明

读取当前位置、目标位置、电机状态

表 12：读取当前位置、目标位置、电机状态

功能码	寄存器地址	寄存器值说明	
03	1000	电机当前位置（高 16 位）	电机当前位置：读 1000 地址开始的 2 个寄存器
03	1001	电机当前位置（低 16 位）	
03	1002	电机目标位置（高 16 位）	电机目标位置：读 1002 地址开始的 2 个寄存器
03	1003	电机目标位置（低 16 位）	
03	1004	电机状态（详细见表 M01）	
03	1006-1007	当前运行速度脉冲周期	

当前位置、目标位置、电机状态、当前运行速度脉冲周期 4 个参数可以用 03 功能码读 1000 开始地址的 8 个寄存器。寄存器地址为 10 进制数据。

复位电机、电机停止

表 13：步进一体机运动指令(1)

功能码	寄存器地址	寄存器值说明
06	2000	复位电机(写任何值)，写地址 2000 寄存器
06	2001	电机停止，写地址 2001 寄存器 寄存器值=0：减速停止 寄存器值=251：立即停止

修改步进一体机的参数：485.ID、细分、硬件配置、波特率、找零速度、默认运行速度、空闲电流比、参数固化。寄存器地址为 10 进制数据。

表 14: 步进一体机参数

功能码	寄存器地址	寄存器值说明
06	3000	485.ID, 取低 8 位, 最大到 255
06	3001	细分, 取低 8 位, 最大到 255
06	3002	硬件参数配置, 详细见表 M02
06	3003	波特率
06	3008	找零速度, 对应的脉冲周期
06	3009	默认运行速度, 对应的脉冲周期; 找零之后的速度恢复到这个速度
06	3020	电机不动空闲时候电流比(10-100)
16	3010-3011	找零最多步数
16	3014-3015	开关脱落步数
06	4002	参数存储到内部 flash

步进一体机运动指令：定位、正反转等。寄存器地址为 10 进制数据，功能码数据为 10 进制数据。写寄存器值，其寄存器地址均为连续写 2 个寄存器值。

表 15: 步进一体机运动指令(3)

功能码	寄存器地址	寄存器值说明
16	2002-2003	步进一体机正确复位情况下。电机定位：写地址 2002 开始的 2 个寄存器。寄存器值为定位的步数坐标,执行 定位模式 的定位功能。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。03 功能码读取 1004 寄存器值, 可以读取步进一体机的状态。
16	2004-2005	无论当前电机状态如何, 在当前的位置正转寄存器值(寄存器值!=0)步。步进一体机执行 正反转模式 的正转功能。寄存器值=0, 步进一体机执行 速度模式 下的正转。无论寄存器值何值, 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。03 功能码读取 1004 寄存器值, 可以读取步进一体机的状态。
16	2006-2007	无论当前电机状态如何, 在当前的位置反转寄存器值(寄存器值!=0)步。步进一体机执行 正反转模式 的反转功能。寄存器值=0, 步进一体机执行 速度模式 下的反转。无论寄存器值何值, 反向运行碰到 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。03 功能码读取 1004 寄存器值, 可以读取步进一体机的状态。

步进一体机运动指令：变速。寄存器地址为 10 进制数据，功能码数据为 10 进制数据。写寄存器值，2008 寄存器地址为 2 个寄存器值。2010-2011 寄存器地址均为连续写 2 个寄存器值。

表 16: 步进一体机运动指令(4)

功能码	寄存器地址	寄存器值说明
16	2008-2009	寄存器值为表示目标速度脉冲周期。
16	2010-2011	2 个寄存器值总共 32bits, 按单浮点格式读取。单浮点数表示目标 Rpm。

步进一体机运动指令：到位等。寄存器地址为 10 进制数据，功能码数据为 10 进制数据。到位功能均在步进一体机正确复位情况下。

表 17: 步进一体机运动指令(5)

功能码	寄存器地址	寄存器值说明
16	2202-2203	电机定位：写地址 2202 开始的 2 个寄存器。寄存器值为定位的步数坐标, 执行 到位模式 的定位功能。碰到 UP 边沿触发开关，减速停下。
16	2402-2403	电机定位：写地址 2402 开始的 2 个寄存器。寄存器值为定位的步数坐标, 执行 到位模式 的定位功能。碰到 DW 边沿触发开关，减速停下。
16	2204-2205	电机正转：写地址 2402 开始的 2 个寄存器。当前位置正转寄存器值步数, 执行 到位模式 的正转功能。碰到 UP 边沿触发开关，减速停下。
16	2405-2405	电机正转：写地址 2402 开始的 2 个寄存器。当前位置正转寄存器值步数, 执行 到位模式 的正转功能。碰到 DW 边沿触发开关，减速停下。
16	2206-2207	电机反转：写地址 2402 开始的 2 个寄存器。当前位置反传寄存器值步数, 执行 到位模式 的反转功能。碰到 UP 边沿触发开关，减速停下。
16	2406-2407	电机反转：写地址 2402 开始的 2 个寄存器。当前位置反转寄存器值步数, 执行 到位模式 的反转功能。碰到 DW 边沿触发开关，减速停下。

8.3 Modbus-RTU 协议介绍

参考<详解 modbus 通讯协议.pdf>的功能码描述。

8.3.1 读保持寄存器03 (0x03)

表 18: 03读保持寄存器主机发送请求

帧码	字节数	数值及说明
目标地址	1	步进一体机的地址，1-254，一般 0 为广播地址。
功能码	1	03(0x03)
起始地址	2	0x0000-0xffff
寄存器数量	2	1-125
CRC16	2	目标地址+功能码+起始地址+寄存器数量， 计算得出的值

表 19: 03 读保持寄存器从机响应

帧码	字节数	数值及说明
目标地址	1	步进一体机的地址。
功能码	1	03(0x03)
字节数	1	2×寄存器数量
寄存器值	2×寄存器数量	
CRC16	2	目标地址+功能码+字节数+寄存器值，计算得出的值

举例, 读取地址为 0x01 步进一体机电机状态:

主机发送: 01 03 03 EC 00 01 45 BB

数值	字节数	说明
01	1	目标地址, 0x01 的步进一体机
03	1	功能码
03 EC	2	起始地址 0x03EC=1004,和表 12 的寄存器地址值对应
00 01	2	寄存器数量 1 个
45 BB	2	CRC16

步进一体机返回: 01 03 02 00 00 B8 44

数值	字节数	及说明
01	1	目标地址, 0x01 的步进一体机
03	1	03(0x03)
02	1	寄存器个数 1 个
00 00	2×寄存器数量	寄存器值
B8 44	2	CRC16

8.3.2 写单个寄存器 06 (0x06)

表 20: 06 写单个寄存器主机发送请求

帧码	字节数	数值及说明
目标地址	1	步进一体机的地址, 1-254, 一般 0 为广播地址。
功能码	1	06(0x06)
寄存器地址	2	0x0000-0xffff
寄存器值	2	0x0000-0xffff
CRC16	2	目标地址+功能码+起始地址+寄存器数量, 计算得出的值

表 21: 06 写单个寄存器从机响应

帧码	字节数	数值及说明
目标地址	1	步进一体机的地址。
功能码	1	06(0x06)
寄存器地址	2	2×寄存器数量
寄存器值	2	
CRC16	2	目标地址+功能码+字节数+寄存器值, 计算得出的值

举例, 写地址为 0x01 步进一体机减速停止:

主机发送: 01 06 07 D1 00 00 D8 87

数值	字节数	说明
01	1	目标地址, 0x01 的步进一体机
06	1	功能码
07 D1	2	起始地址 0x07D1=2001,和表 13 的寄存器地址值对应
00 00	2	寄存器值 0x0000, 和表 13 的寄存器地址值对应
D8 87	2	CRC16

步进一体机返回: 01 06 07 D1 00 00 D8 87

数值	字节数	说明
01	1	目标地址, 0x01 的步进一体机
06	1	功能码
07 D1	2	起始地址 0x07D1=2001,和表 13 的寄存器地址值对应
00 00	2	寄存器值 0x0000, 和表 13 的寄存器地址值对应
D8 87	2	CRC16

8.3.3 写多个寄存器 16 (0x10)

表20: 22写多个寄存器主机发送请求

帧码	字节数	数值及说明
目标地址	1	步进一体机的地址, 1-254, 一般 0 为广播地址。
功能码	1	16(0x16)
寄存器起始地址	2	0x0000-0xffff
寄存器数量	2	0x0000-0xffff
字节数	1	寄存器数量×2
寄存器值	寄存器数量×2	寄存器值
CRC16	2	目标地址+功能码+起始地址+寄存器数量, 计算得出的值

表 23: 16 写单个寄存器从机响应

帧码	字节数	数值及说明
目标地址	1	步进一体机的地址。
功能码	1	16(0x10)
寄存器起始地址	2	2×寄存器数量
寄存器数量	2	1-123
CRC16	2	目标地址+功能码+字节数+寄存器值, 计算得出的值

举例, 写地址为 0x01 步进一体机正传 1000 步, 地址 2004 连续写两个寄存器值。

主机发送: 01 10 07 D4 00 02 04 00 00 03 E8 D9 8E

数值	字节数	说明
01	1	目标地址, 0x01 的步进一体机
10	1	功能码 0x10=16
07 D4	2	寄存器起始地址 0x07D4=2004,和表 15 的寄存器地址值对应
00 02	2	寄存器数量 2 个
04	1	字节数量=2 个×2
00 00 03 E8	4	寄存器值 0x00 00 03 E8 = 1000
D9 87	2	CRC16

步进一体机返回: 01 10 07 D4 00 02 00 84

数值	字节数	说明
01	1	目标地址, 0x01 的步进一体机
10	1	功能码 0x10=16
07 D4	2	寄存器起始地址 0x07D4=2004,和表 15 的寄存器地址值对应
00 02	2	寄存器数量 0x0002 个
00 84	2	CRC16

8.4 校验码算法

```

unsigned short ModRtuCrcTx(unsigned char buf[], int len)
{
    unsigned short crc = 0xFFFF;
    int pos,i;
    for (pos = 0; pos < len; pos++)
    {
        crc ^= buf[pos];
        for (i = 8; i != 0; i--)
        {
            if ((crc & 0x0001) != 0)
            {
                crc >>= 1;
                crc ^= 0xA001;
            }
            else    crc >>= 1;
        }
    }
    return crc;
}

```

9. 包装

序号	数量	部件
1	1	CAN、RS485 总线接口步进闭环 [7TCRSM57E56/76]
2	1	PH2.0-2×3P 带线
3	1	5264-6P 带线
4	1	包装盒一个

10. 一体机外形尺寸图

