



# EasySmc 2200 步进电机控制器

## 使 用 说 明

厦门华菱工控技术有限公司

Xiamen Hualing Industry Control Technology Co.,Ltd

一、概述	1
二、性能指标	1
三、运动控制系统组成框图	1
3.1、PLC作为上位机，如松下的FP0-C10CR，带RS-232C串口	1
3.2、工业计算机作为上位机，使用RS232 or RS485 or USB与 EasySmc 2200 构成主从系统	2
3.3、EasyHmi 2200 作为上位机，可最大限度的提高性能价格比，需要用户订制	2
四、接线说明和跳线定义：	2
4.1、外形尺寸说明	3
4.2、接线端说明	3
4.2.1、电源输入信号	3
4.2.2、通信口	3
4.2.3、步进电机驱动信号	3
4.2.4、下限输入，原点输入，上限输入，外部联锁输入信号	4
4.2.5、4路普通输出端	5
4.2.6、紧急故障停止输入信号	5
4.2.7、手动信号输入	5
4.3、手动/联机跳线说明	5
五、串行口控制命令说明	6
5.1、初始化命令	7
5.1.1、设置X轴加速度，起跳频率，最大速度（命令码 1……7）	7
5.1.2、设置Y轴加速度，起跳频率，最大速度（命令码 8……12）	8
5.1.3、设置Z轴加速度，起跳频率，最大速度（命令码 13……15）	9
5.1.4、设置各轴手动参数和反向复位距离S（命令码 16+DAT1）	9
5.1.5、恢复默认的控制参数，保存当前的控制参数（命令码 17+DAT1）	10
5.1.6、紧急制动、平滑制动命令（命令码 18+DAT1）	10
5.1.7、控制命令（命令码 19+DAT1）	10
5.2、轴位置参数设置命令	11
5.2.1、设置X轴软件限位寄存器（命令码 21+DAT1）	12
5.2.2、设置Y轴软件限位寄存器（命令码 22+DAT1）	12
5.2.3、设置Z轴软件限位寄存器（命令码 23+DAT1）	12
5.2.4、设置X轴当前位置寄存器（命令码 24+DAT1）	12
5.2.5、设置Y轴当前位置寄存器（命令码 25+DAT1）	12
5.2.6、设置Z轴当前位置寄存器（命令码 26+DAT1）	12
5.3、轴状态和运动参数查询命令	12
5.3.1、查询X轴状态（命令码 31+DAT1）	12
5.3.2、查询Y轴状态（命令码 32+DAT1）	13
5.3.3、查询Z轴状态（命令码 33+DAT1）	13
5.3.4、查询X轴运动参数（命令码 34+DAT1）	13
5.3.5、查询Y轴运动参数（命令码 35+DAT1）	14
5.3.6、查询Z轴运动参数（命令码 36+DAT1）	14
5.3.7、读出控制器当前手动、联机状态（命令码 37+DAT1）	15
5.3.8、命令执行情况的查询（命令码 38+DAT1）	16
5.4、复位运动命令和独立运动命令	16
5.4.1、X轴复位运动（命令码 41+DAT1）	16
5.4.2、Y轴复位运动（命令码 42+DAT1）	17
5.4.3、Z轴复位运动（命令码 43+DAT1）	17

5.4.4、X轴独立运动（命令码 44+DAT1） .....	18
5.5、直线插补命令 .....	18
5.5.1、XY直线插补运动（命令码 51+DAT1+DAT2） .....	18
5.5.2、XZ直线插补运动（命令码 52+DAT1+DAT2） .....	18
5.5.3、YZ直线插补运动（命令码 53+DAT1+DAT2） .....	18
5.5.4、XYZ直线插补运动（命令码 57+DAT1+DAT2+DAT3） .....	18
5.6、圆弧插补命令 .....	18
5.6.1、XY圆弧插补运动（命令码 61+DAT1+DAT2+DAT3） .....	19
5.6.2、XZ圆弧插补运动（命令码 62+DAT1+DAT2+DAT3） .....	19
5.6.3、YZ圆弧插补运动（命令码 63+DAT1+DAT2+DAT3） .....	19
5.7、样条插补命令 .....	19
5.7.1、命令格式 .....	19
5.7.2、样条同步输出功能 .....	21
5.8、其它 .....	21
5.8.1、计数器（2个） .....	21
5.8.2、程序下载、上载、脱机执行 .....	22
六、USB驱动程序安装和EM2200.DLL调用 .....	23
6.1、USB驱动程序安装 .....	23
6.2、EM2200.DLL调用 .....	23
6.2.1、Delphi用法如下 .....	23
6.2.2、Visual C++ 6.0用法如下 .....	24
6.2.3、Visual Basic 6.0用法如下 .....	25
6.2.4、写数据和读数据时 psend 和 prec 的数据格式 .....	25
七、USB写命令清单 .....	26
7.1、初始化命令 .....	26
7.1.1、设置X轴速度参数（命令码\$01……\$07） .....	26
7.1.2、设置Y轴速度参数（命令码\$08……\$12） .....	26
7.1.3、设置Z轴速度参数（命令码\$13……\$15） .....	27
7.1.4、设置手动参数和反向复位距离（命令码\$16+DAT1） .....	27
7.1.5、恢复默认的控制参数，保存当前的控制参数（命令码\$17+DAT1） .....	27
7.1.6、紧急制动、平滑制动命令（命令码\$18+DAT1） .....	27
7.1.7、控制命令（命令码\$19+DAT1） .....	28
7.2、轴位置参数设置命令 .....	29
7.2.1、设置X轴软件限位寄存器（命令码\$21+DAT1） .....	29
7.2.2、设置Y轴软件限位寄存器（命令码\$22+DAT1） .....	29
7.2.3、设置Z轴软件限位寄存器（命令码\$23+DAT1） .....	29
7.2.4、设置X轴当前位置寄存器（命令码\$24+DAT1） .....	29
7.2.5、设置Y轴当前位置寄存器（命令码\$25+DAT1） .....	30
7.2.6、设置Z轴当前位置寄存器（命令码\$26+DAT1） .....	30
7.3、复位运动和独立运动命令 .....	30
7.3.1、X轴复位运动（命令码\$41+DAT1） .....	30
7.3.2、Y轴复位运动（命令码\$42+DAT1） .....	30
7.3.3、Z轴复位运动（命令码\$43+DAT1） .....	31
7.3.4、X轴独立运动（命令码\$44+DAT1） .....	31
7.3.5、Y轴独立运动（命令码\$45+DAT1） .....	31
7.3.6、Z轴独立运动（命令码\$46+DAT1） .....	31

7.4、直线插补命令 .....	31
7.4.1、XY直线插补运动（命令码\$51+DAT1+DAT2） .....	31
7.4.2、XZ直线插补运动（命令码\$52+DAT1+DAT2） .....	32
7.4.3、YZ直线插补运动（命令码\$53+DAT1+DAT2） .....	32
7.4.4、XYZ直线插补运动（命令码\$57+DAT1+DAT2） .....	32
7.5、圆弧插补命令 .....	32
7.5.1、XY圆弧插补运动（命令码\$61+DAT1+DAT2+DAT3） .....	32
7.5.2、XZ圆弧插补运动（命令码\$62+DAT1+DAT2+DAT3） .....	32
7.5.3、YZ圆弧插补运动（命令码\$63+DAT1+DAT2+DAT3） .....	32
7.6、样条插补命令 .....	32
7.6.1、样条插补运动参数（命令码\$70+DAT1+DAT2+DAT3） .....	32
7.6.2、样条插补节点向量+第一个控制顶点坐标（命令码\$75+DAT1+DAT2+DAT3） .....	33
7.6.3、样条插补节点向量+第二个控制顶点坐标（命令码\$76+DAT1+DAT2+DAT3） .....	33
7.6.4、样条插补拐角加速度+倒数第二个控制顶点坐标（命令码\$77+DAT1+DAT2+DAT3） .....	33
7.6.5、样条插补弓高误差+倒数第一个控制顶点坐标（命令码\$78+DAT1+DAT2+DAT3） .....	33
7.6.6、样条同步输出功能（命令码\$19+DAT1） .....	33
7.7、其它 .....	33
7.7.1、计数器（2个） .....	33
7.7.2、程序下载 .....	33
八、USB读命令清单 .....	34
8.1、查询X轴状态（命令码\$31, rdat1: \$01.....\$06, rdat2 不使用） .....	34
8.2、查询Y轴状态（命令码\$32, 其它同X轴） .....	35
8.3、查询Z轴状态（命令码\$33, 其它同X轴） .....	35
8.4、查询X轴运动参数（命令码\$34, rdat1: \$01.....\$07, rdat2 不使用） .....	35
8.5、查询Y轴运动参数（命令码\$35, rdat1: \$01.....\$05, rdat2 不使用） .....	35
8.6、查询Z轴运动参数（命令码\$36, rdat1: \$01.....\$03, rdat2 不使用） .....	36
8.7、读出控制器手动、联机状态（命令码\$37, rdat1: \$00.....\$0F, rdat2 不使用） .....	36
8.8、命令执行情况查询 .....	37
九、故障检查、维修保养、售后服务 .....	38

# EasySmc 2200 步进电机控制器 使用说明



## 一、概述

EasySmc 2200 采用 60MHz 高速 ARM 32 位 处理器，实现 3 轴的步进电机控制。

控制器使用异步串行接口 (RS232、RS422、RS485) 或者 USB 与外部通信, 可以和 PLC, IPC, EasyHmi2100 组成运动控制系统。控制器执行运动控制的所有细节, 包括脉冲和方向信号的产生, 原点的复位, 上下限位开关的检测, 自动加减速, 插补运算等等, 能很方便的替代 PLC 的位置控制模块, IPC 的运动控制卡, 极大的提高性能价格比。由于采用集散控制, 上位机 (PLC、IPC、EasyHmi2100) 能从实时的运动控制任务中摆脱出来, 更好的执行人机交互和其它控制任务。

控制器功能齐全, 使用方便, 价格实惠。能实现任意两轴的直线, 圆弧插补, 输出脉冲达到 100KHz。

控制器全部程序由本公司自主开发, 可以快速, 准确的为用户订制特殊曲线的插补控制命令。

如果只需要增加普通的 I/O, 直线和圆弧插补, 请选用 EasySmc 2100。

如果需要使用具有速度前瞻控制的 B 样条插补功能 (495 个控制顶点), 请选用 EasySmc 2130。

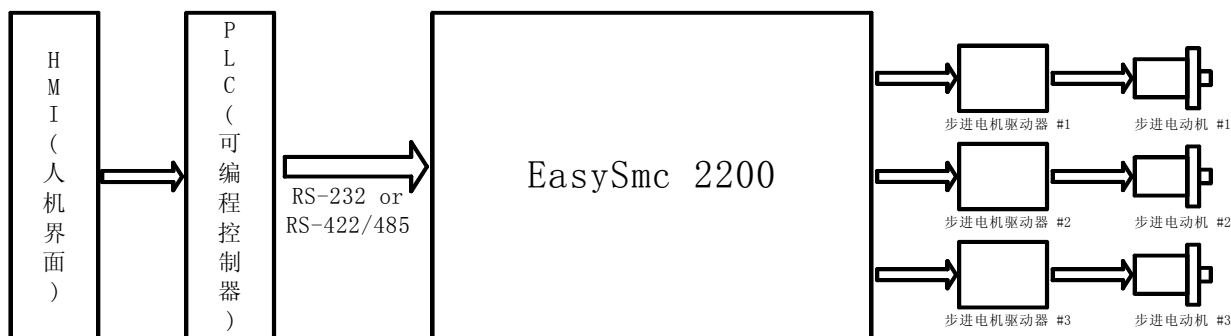
如果需要更高速的脉冲输出 (2MHz), 请选用 EasySmc 2300。

## 二、性能指标

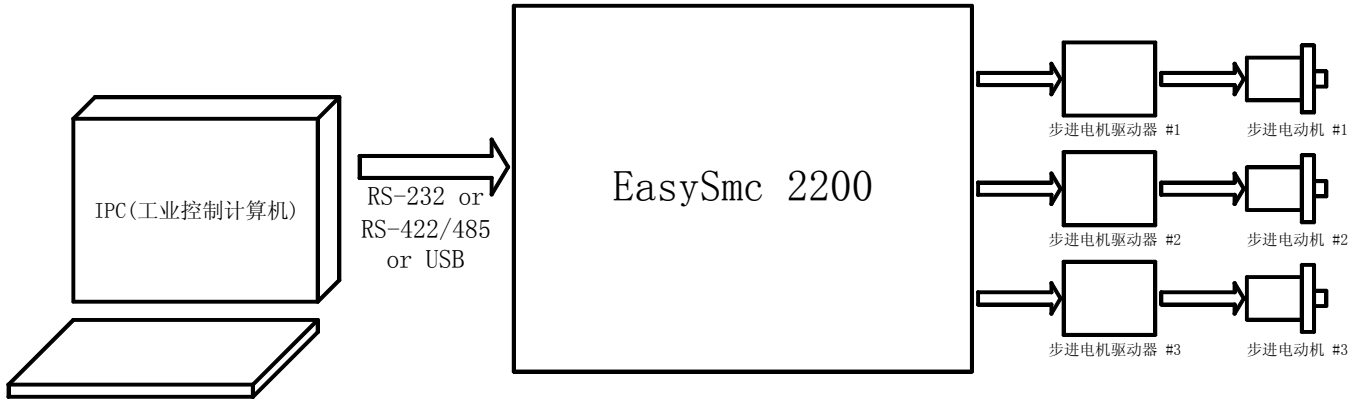
- 1、使用工业电源+24V 直流供电, 带反向输入保护, 最大功耗 10W;;
- 2、X, Y, Z 三轴独立、直线插补、圆弧插补步进控制;
- 3、脉冲输出速度 40KHz, 可以满足步进电机控制的需要 (1.8 度, 20 细分的进给速度 600 转/分);
- 4、梯形加减速速度, 防止运动过程中的失步和过冲;
- 5、单轴最大步进步数 - 16, 777, 215 — + 16, 777, 215;
- 6、每轴各带一上限, 下限的软件限位寄存器;
- 7、每轴带一原点信号, 上限、下限信号输入, 外部启动联锁信号; 系统带一紧急故障输入信号;
- 8、控制器具有手/自动切换功能, 方便调试;
- 9、可选择 RS232, RS422, RS485 通信接口, 波特率 2400bps, 4800bps, 9600bps, 19200bps;  
USB 通信, 每秒钟可发送 100 条命令, 即通信周期为 10ms;
- 10、所有 I/O 使用光电隔, 提高抗干扰性能; 4 路通用输出、4 路通用输入; 2 路 10 位 A/D 输入;
- 11、控制器命令可接受用户订制, 完成特殊曲线的插补控制; 采用压扣式接线端, 安装接线方便可靠。

## 三、运动控制系统组成框图

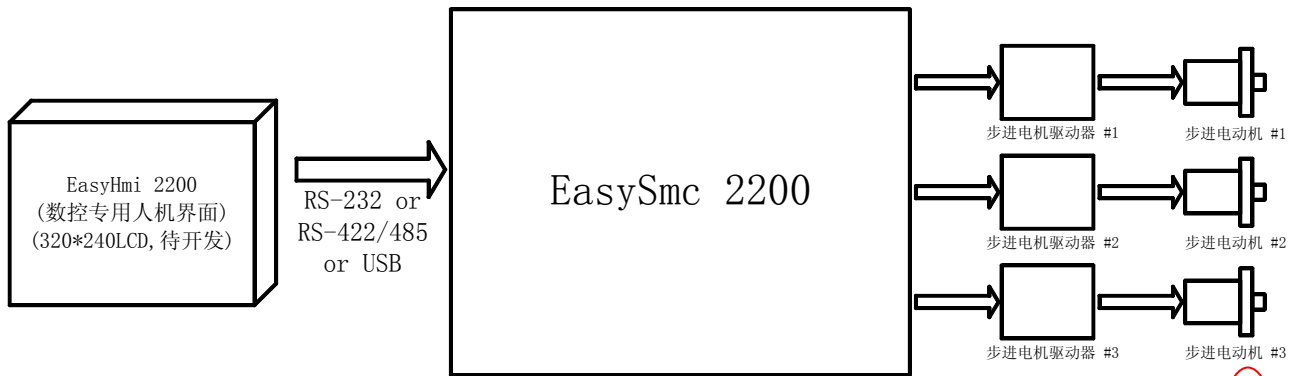
### 3.1、PLC作为上位机, 如松下的FP0-C10CR, 带RS-232C串口



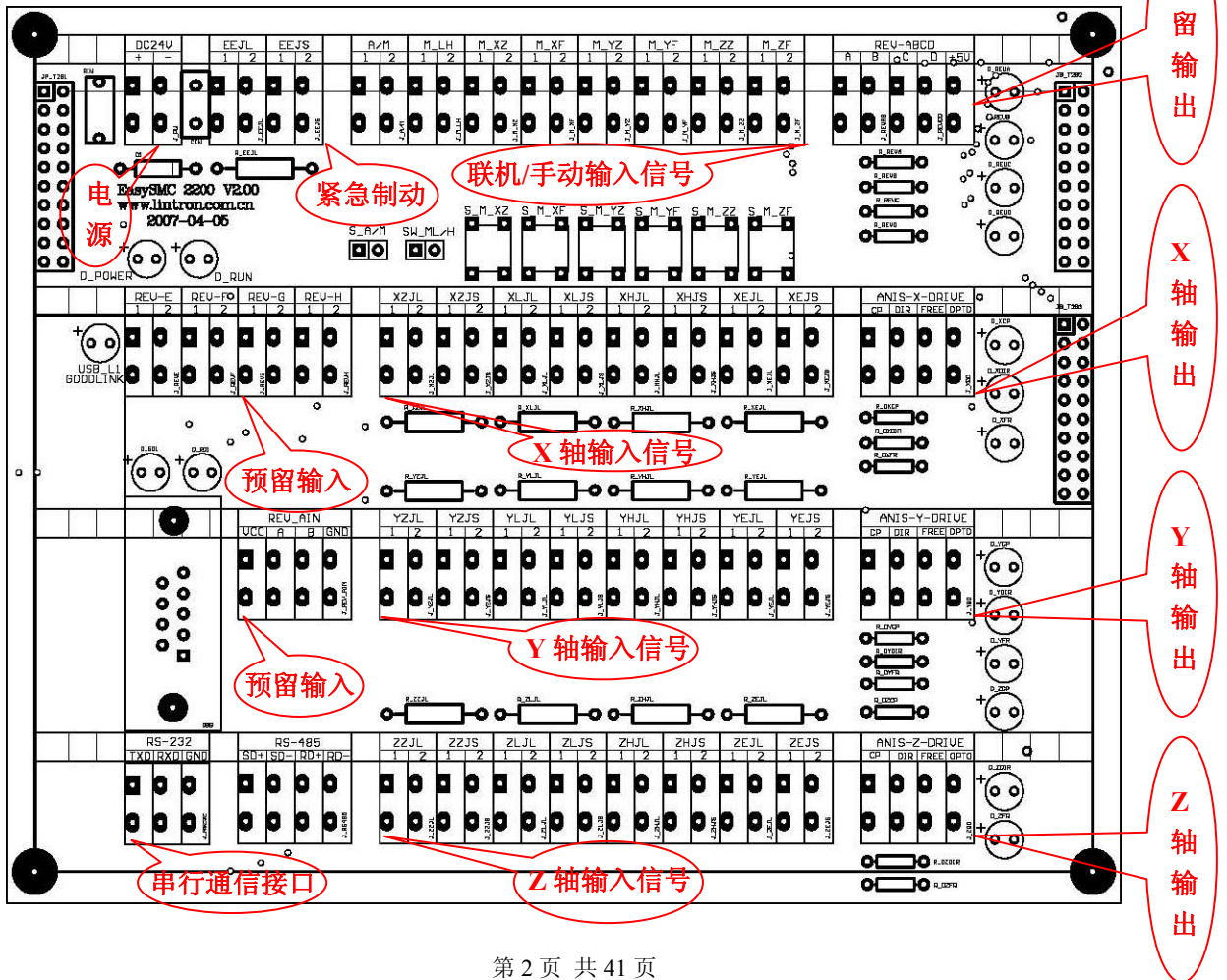
3.2、工业计算机作为上位机，使用RS232 or RS485 or USB与 EasySmc 2200 构成主从系统



3.3、EasyHmi 2200 作为上位机，可最大限度的提高性能价格比，需要用户订制



四、接线说明和跳线定义：



4.1、外形尺寸说明

控制器外形为：150mm\*120mm；安装孔为 Φ3mm，孔中心距边沿 3.80mm。

4.2、接线端说明

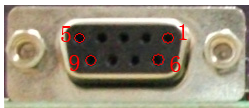
4.2.1、电源输入信号

接线端+24V，“+”接+24VDC的正端，“-”接+24VDC的负端；

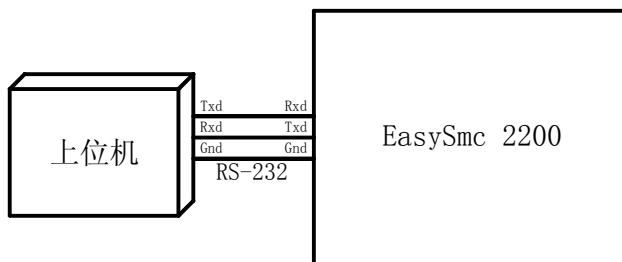
4.2.2、通信口

4.2.2.1、串行口信号

出厂默认使用 RS232 信号。如果碰到强烈干扰，通信电缆请使用屏蔽线。

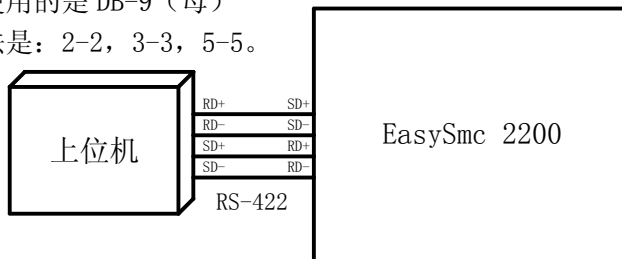


(1) RS232: TXD 发送端,  
RXD 接收端,  
GND 信号地;



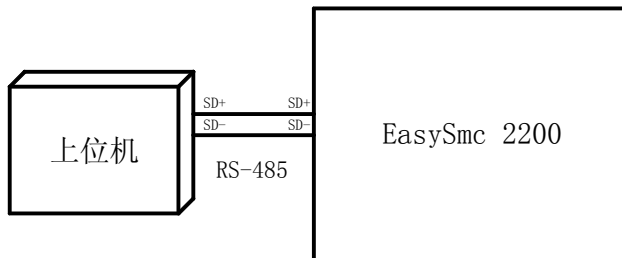
1 2 3  
J\_COM1 PC机使用的是 DB-9 (公)  
J\_COM2 控制器使用的是 DB-9 (母)  
1 2 3 接线方法是: 2-2, 3-3, 5-5。

(2) RS422: 发送端 SD+, SD-,  
接收端 RD-, RD+;



1 2 3  
J\_COM1   
J\_COM2   
1 2 3

(3) RS485: 同 RS422,  
跳线 J\_485A, J\_485Y 闭合;



1 2 3  
J\_COM1   
J\_COM2   
1 2 3

4.2.2.2、USB



4.2.3、步进电机驱动信号

步进电机驱动信号采用共阳极，负脉冲信号有效；

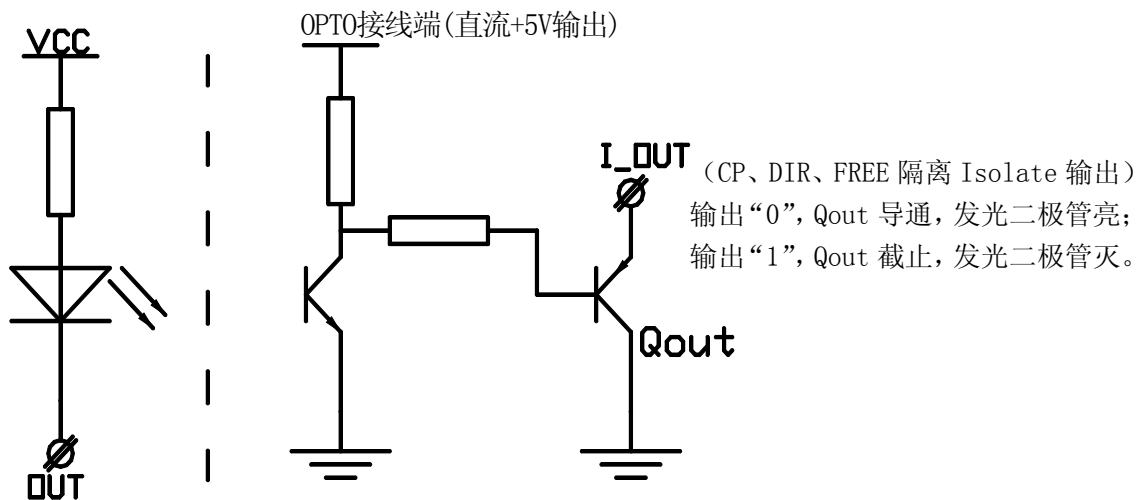
**CP:** 步进脉冲信号，控制器每发出一个脉冲，驱动器驱动步进电机旋转一步进角；

**DIR:** 方向电平信号，此信号决定电机的旋转方向。比如，此信号为高电平时，步进电机顺时针方向旋转；此信号为低电平时，步进电机逆时针方向旋转；

**FREE:** 脱机信号。此信号为低电平时，电机处于无力矩状态；此信号为高电平或者悬空不接时，脱机功能无效。脱机功能需要驱动器支持。不使用脱机时，FREE 做通用输出；

OPTO: 阳极公共端 (+5V DC 输出, 此电源由控制器提供, 切勿外接电源)。

四路信号的输出电路图如下: 三极管 Qout 的吸收电流在 100mA 以下。驱动器需要设置限流电阻。如果没有限流, 可能导致三极管烧毁, DC/DC 电源模块烧毁。



#### 4.2.4、下限输入, 原点输入, 上限输入, 外部连锁输入信号

考虑到大部分情况使用对射光电开关来检测外部输入信号, 控制器布置了光电开关的发射二极管的接线端, +24V DC 供电, 1.5KΩ 限流电阻。

如果不使用光电开关, 使用机械触点的行程开关, 接发射二极管的接线端闲置不使用。

下限, 原点, 上限输入在复位运动中使用。当控制器做独立, 直线, 圆弧运动时, 如果遇到下限有效或者上限有效, 视为“故障”, 控制器将立即停止输出, 等待清除命令缓冲区 (5.1.7.4)。

##### 4.2.4.1、X 轴外部输入信号说明

XZJL: 接 X 轴原点检测发光二极管, XZJL1 接阳极 (A), XZJL2 接阴极 (K);

XZJS: 接 X 轴原点检测三极管, XZJS1 接集电极 (C), XZJS2 接发射极 (E);

XLJL: 接 X 轴下限检测发光二极管, XLJL1 接阳极 (A), XLJL2 接阴极 (K);

XLJS: 接 X 轴下限检测三极管, XLJS1 接集电极 (C), XLJS2 接发射极 (E);

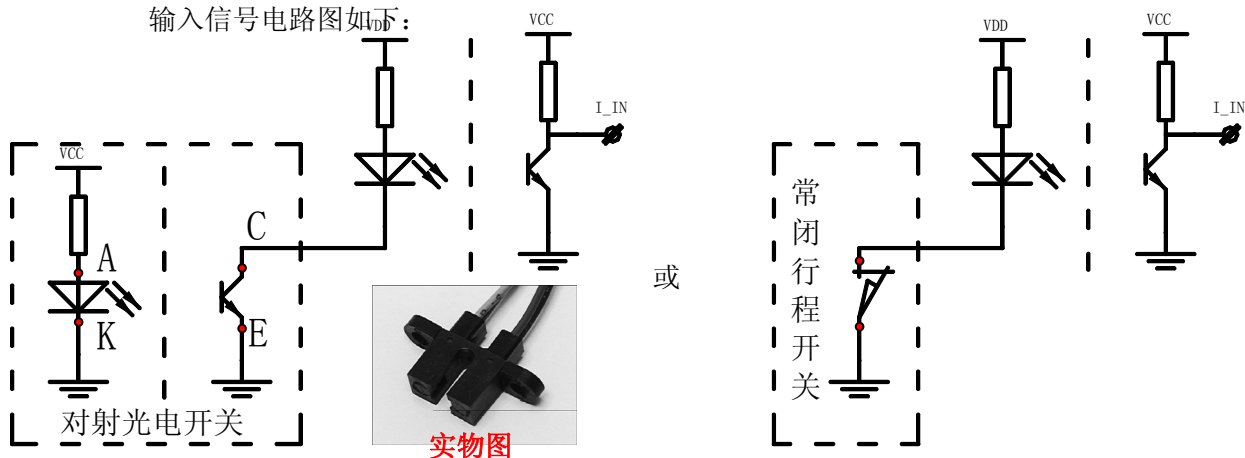
XHJL: 接 X 轴上限检测发光二极管, XHJL1 接阳极 (A), XHJL2 接阴极 (K);

XHJS: 接 X 轴上限检测三极管, XHJS1 接集电极 (C), XHJS2 接发射极 (E);

XEJL: 接 X 轴外部启动检测发光二极管, XEJL1 接阳极 (A), XEJL2 接阴极 (K);

XEJS: 接 X 轴外部启动检测三极管, XEJS1 接集电极 (C), XEJS2 接发射极 (E);

输入信号电路图如下:



注意: 调试过程中, 应该将 XLJS-1 和 2 短接, XHJS-1 和 2 短接, 否则控制器会视其为“上、下限故障”而无法输出脉冲。Y 轴, Z 轴的做法类同。

##### 4.2.4.2、Y 轴外部输入信号说明

YZJL: 接 Y 轴原点检测发光二极管, YZJL1 接阳极 (A), YZJL2 接阴极 (K);



YZJS: 接 Y 轴原点检测三极管, YZJS1 接集电极 (C), YZJS2 接发射极 (E);  
 YLJL: 接 Y 轴下限检测发光二极管, YLJL1 接阳极 (A), YLJL2 接阴极 (K);  
 YLJS: 接 Y 轴下限检测三极管, YLJS1 接集电极 (C), YLJS2 接发射极 (E);  
 YHJL: 接 Y 轴上限检测发光二极管, YHJL1 接阳极 (A), YHJL2 接阴极 (K);  
 YHJS: 接 Y 轴上限检测三极管, YHJS1 接集电极 (C), YHJS2 接发射极 (E);  
 YEJL: 接 Y 轴外部启动检测发光二极管, YEJL1 接阳极 (A), YEJL2 接阴极 (K);  
 YEJS: 接 Y 轴外部启动检测三极管, YEJS1 接集电极 (C), YEJS2 接发射极 (E);

#### 4.2.4.3、Z 轴外部输入信号说明

ZZJL: 接 Z 轴原点检测发光二极管, ZZJL1 接阳极 (A), ZZJL2 接阴极 (K);  
 ZZJS: 接 Z 轴原点检测三极管, ZZJS1 接集电极 (C), ZZJS2 接发射极 (E);  
 ZLJL: 接 Z 轴下限检测发光二极管, ZLJL1 接阳极 (A), ZLJL2 接阴极 (K);  
 ZLJS: 接 Z 轴下限检测三极管, ZLJS1 接集电极 (C), ZLJS2 接发射极 (E);  
 ZHJL: 接 Z 轴上限检测发光二极管, ZHJL1 接阳极 (A), ZHJL2 接阴极 (K);  
 ZHJS: 接 Z 轴上限检测三极管, ZHJS1 接集电极 (C), ZHJS2 接发射极 (E);  
 ZEJL: 接 Z 轴外部启动检测发光二极管, ZEJL1 接阳极 (A), ZEJL2 接阴极 (K);  
 ZEJS: 接 Z 轴外部启动检测三极管, ZEJS1 接集电极 (C), ZEJS2 接发射极 (E);

#### 4.2.4.4、四路普通输入端, 两路非隔离 10 位 A/D 输入端

4 路普通输入端为 REVE, REVF, REVG, REVH;  
 2 路 A/D 输入端为 REV-AINA, REV-AINB。

#### 4.2.5、4 路普通输出端

4 路普通输出端为 REVA, REVB, REVC, REVD;

#### 4.2.6、紧急故障停止输入信号

默认为常开信号, 闭合时禁止脉冲输出。

EEJL: 接紧急故障发光二极管, EEJL1 接阳极 (A), EEJL2 接阴极 (K);  
 EEJS: 接紧急故障检测三极管, EEJS1 接集电极 (C), EEJS2 接发射极 (E)

#### 4.2.7、手动信号输入

为了方便调试, 设置手动输入信号。

A/M: 手动/联机选择。断开: 联机; 闭合: 手动。

M\_L/H: 手动方式选择。断开: 电平触发; 闭合: 脉冲触发。

M\_XZ: X 轴正方向手动。接外部常开开关。

M\_XF: X 轴负方向手动。接外部常开开关。

M\_YZ: Y 轴正方向手动。接外部常开开关。

M\_YF: Y 轴负方向手动。接外部常开开关。

M\_ZZ: Z 轴正方向手动。接外部常开开关。

M\_ZF: Z 轴负方向手动。接外部常开开关。

### 4.3、手动/联机跳线说明

#### 4.3.1、手动状态

当跳线 S\_A/M 闭合时, 控制器处于手动状态, 手动输入有效, 串行口的命令无效。

当从联机状态切换到手动状态时, 必须等到命令缓冲区的命令执行完毕, 否则手动无效。

(1) 当 SW\_ML/H 断开时, 电平触发, 如果手动信号输入, 相应轴按默认速度做匀速运动;

(2) 当 SW\_ML/H 闭合时, 脉冲触发, 当手动信号输入一个脉冲, 相应轴运动一个点动距离。

#### 4.3.2、联机状态

当跳线 S\_A/M 断开时, 控制器处于联机状态。

(1) 当 SW\_ML/H 断开时, 控制器处于“在线运行”状态, 接收、执行来自上位机的命令。

(2) 当 SW\_ML/H 闭合时, 控制器处于“脱机运行”状态, 上位机在此状态下编辑修改控制器 ROM 中的程序; 修改完成后, 控制器可以脱离上位机执行控制器 ROM 中的程序。

4.3.3、通信地址码选择

SW\_COM (位置在 CPU PACK 右) 上方, ON=1, OFF=0。



(1) 通信地址码选择: SEL1、SEL2, 00, 01, 02, 03; 出厂默认 00。

拨码位 \ 地址码	00	01	02	03
SEL1, SEL2	0、0	0、1	1、0	1、1

(2) 通信波特率选择: SEL3, SEL4, 2400, 4800, 9600, 19200bps。出厂默认 19200bps。

拨码位 \ 波特率	2400bps	4800bps	9600bps	19200bps
SEL3, SEL4	0、0	0、1	1、0	1、1

五、串行口控制命令说明

通信协议: “1” 位起始位, “8” 数据位, “无” 校验位, “1” 位停止位。

控制器作为下位机, 接收、执行上位机发送的命令, 命令缓冲区长度为 63 条, 具体格式如下:

命令格式: STX (1BYTE) + 地址码 (2BYTE) + 命令码 (2BYTE)

+ 数据 (7 or 14 or 21 BYTE) + 启动条件 (1BYTE) + ETX (1BYTE) + 校验码 (2BYTE)

备注: (1) 除去 STX (02H) 和 ETX (03H) 使用十六进制数外, 其他部分都采用 ASC-II 码。

(2) “数据” 部分根据不同的命令码有不同的长度, 详细见以下的命令说明。

(3) 启动条件: 查看“启动条件”详细说明 (说明书第 6 页)。

(4) 校验码: 从地址码到 ETX 的所有字节的和。(详见 5.1.11 例子)

响应格式: (1) 写命令, 含设置命令和任务命令, 返回格式为:

STX + 地址码 + 命令码 + 启动条件 + ETX + 校验码

(2) 读命令, 状态查询命令, 返回格式为:

STX + 地址码 + 命令码 + 数据 + 启动条件 + 校验码

(3) 错误响应, 返回命令码为 99H, 返回格式为:

STX + 地址码 + 命令码 + 错误代码 + 启动条件 + 校验码

错误代码: I: (30H, 30H, 30H, 30H, 30H, 30H, 31H): 无此命令;

II: (30H, 30H, 30H, 30H, 30H, 30H, 32H): 控制器处于手动状态;

III: (30H, 30H, 30H, 30H, 30H, 30H, 33H): 通信缓冲区满;

IV: (30H, 30H, 30H, 30H, 30H, 30H, 34H): 通信过程受干扰,

接收到的数据校验和错误;

ASCII码表

D6D5D4 \ D3D2D1D0	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	,	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

备注：部分 ASC\_II 码 ‘0’：30H；‘1’：31H；‘3’：33H；‘4’：34H；‘5’：35H；‘6’：36H；‘7’：37H；‘8’：38H；‘9’：39H；‘A’：41H；‘B’：42H；‘C’：43H；‘D’：44H；‘E’：45H；‘F’：46H。

### “启动条件”详细说明：

“启动条件”由一个字节的 ASC-II 码构成，其中包含的信息有“命令的执行方式”，“外部启动联锁开关”的触发方式等。下面详细解释：

#### A、命令的执行方式：

##### 1、立即执行：

当接收第 N 条命令的时候，第 N-1 条命令还未执行完毕……

如果第 N 条命令的相关运动轴的运动已经停止，执行该命令。

##### 2、顺序执行：

按顺序从缓冲区中取出命令，当执行完一条命令时才能执行下一条命令。

即使第 N 条命令的相关运动轴已经停止，也必须等到 N-1 条命令执行完毕才能执行。

举例说明：

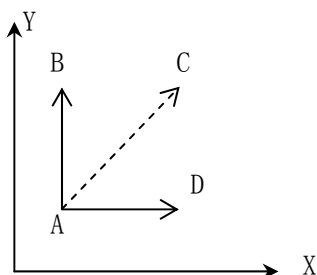
例（1）：第 N-1 条命令为 X 轴从 A 运动到 D，第 N 条命令为 Y 轴从 A 运动到 B。

如果是立即执行，最终的运动轨迹是 A → C；

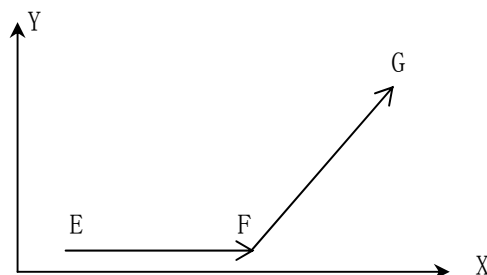
如果是顺序执行，最终的运动轨迹是 A → D → C。

例（2）：第 N-1 条命令为 X 轴从 E 运动到 F，第 N 条命令为直线 XY 插补从 F 运动到 G。

如果在 E 还没运动到 F 时，就执行了 F 到 G 的插补运动，第 N-1 条命令将没有完整执行，所以不管是立即执行还是顺序执行，最终的运动轨迹都只能是 E → F → G。



图（1）



图（2）

#### B、“外部启动联锁开关”的触发方式：

1、电平触发方式：当相应轴如 XEJS（常闭）为断开信号时，执行命令。

2、脉冲触发方式：当相应轴如 XEJS（常闭）瞬间断开后闭合，执行命令。

最后，“启动条件”的值如下：

30H：立即执行，外部启动联锁开关无效。

31H：立即执行，外部启动联锁开关无效。

32H：立即执行，外部电平触发方式。

33H：立即执行，外部脉冲触发方式。

34H：顺序执行，外部启动联锁开关无效。

35H：顺序执行，外部启动联锁开关无效。

36H：顺序执行，外部电平触发方式。

37H：顺序执行，外部脉冲触发方式。

38H：异常处理，仅适用于紧急制动命令和清除命令缓冲区命令。使用此启动条件时，紧急制动命令不进入通信缓冲区，接收到命令后马上执行制动。

#### 5.1、初始化命令

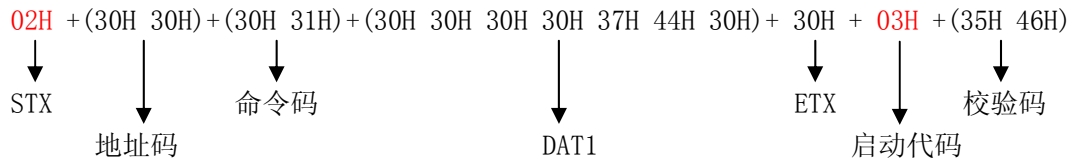
说明：所有“初始化命令”的响应均为“写”响应。

##### 5.1.1、设置X轴加速度，起跳频率，最大速度（命令码 1……7）

###### 5.1.1.1、X轴复位速度（命令码 01+DAT1）

DAT1：X轴复位速度（单位：Hz）。

举例：以下例子认为本控制器地址为 00H，本例，设置复位脉冲速度：2000Hz。



校验码计算：从“地址码”到“ETX”的所有字节的和。

$$(30H+30H) + (30H+31H) + (30H+30H+30H+30H+37H+44H+30H) + 30H+03H=25FH$$

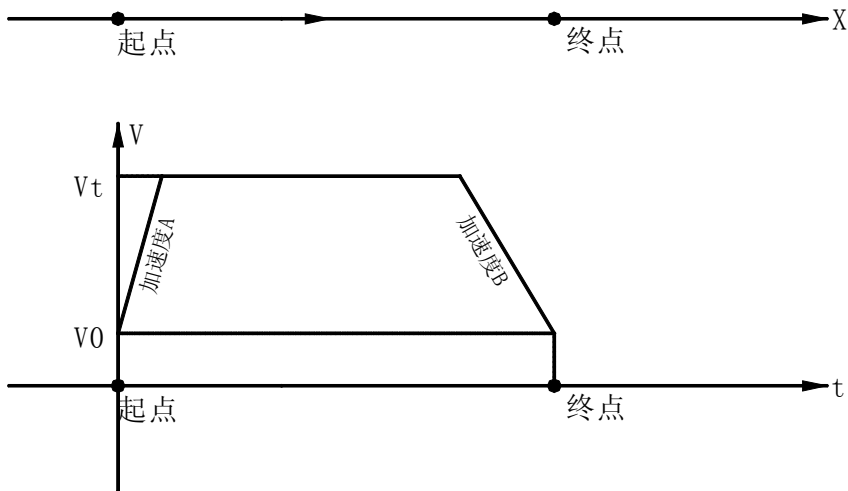
保留最低两位，为 5F，转换成 ASC\_II 为 35H，46H。

**5.1.1.2、X 轴独立运动，直线插补，圆弧插补的起跳频率（命令码 02+DAT1+DAT2+DAT3）**

- DAT1: X 轴独立运动的起跳频率；
- DAT2: X 轴直线插补运动的起跳频率；
- DAT3: X 轴圆弧插补运动的起跳频率。

**5.1.1.3、X 轴独立直线运动的加速度 A，匀速，加速度 B（命令码 03+DAT1+DAT2+DAT3）**

- DAT1: X 轴直线运动的加速度 A，即加速过程的加速度；（单位：Hz/s）
- DAT2: X 轴直线运动匀速频率
- DAT3: X 轴直线运动的加速度 B，即减速过程的加速度。



**5.1.1.4、XY 直线插补运动的加速度 A，匀速，加速度 B（命令码 04+DAT1+DAT2+DAT3）**

- DAT1: XY 轴直线插补运动的加速度 A，即加速过程的加速度；
- DAT2: XY 轴直线插补运动匀速频率
- DAT3: XY 轴直线插补运动的加速度 B，即减速过程的加速度。

**5.1.1.5、XZ 直线插补运动的加速度 A，匀速，加速度 B（命令码 05+DAT1+DAT2+DAT3）**

- DAT1: XZ 轴直线插补运动的加速度 A，即加速过程的加速度；
- DAT2: XZ 轴直线插补运动匀速频率
- DAT3: XZ 轴直线插补运动的加速度 B，即减速过程的加速度。

**5.1.1.6、XY 圆弧插补运动的加速度 A，匀速，加速度 B（命令码 06+DAT1+DAT2+DAT3）**

- DAT1: XY 轴圆弧插补运动的加速度 A，即加速过程的加速度；
- DAT2: XY 轴圆弧插补运动匀速频率
- DAT3: XY 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**5.1.1.7、XZ 圆弧插补运动的加速度 A，匀速，加速度 B（命令码 07+DAT1+DAT2+DAT3）**

- DAT1: XZ 轴圆弧插补运动的加速度 A，即加速过程的加速度；
- DAT2: XZ 轴圆弧插补运动匀速频率
- DAT3: XZ 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**5.1.2、设置Y轴加速度，起跳频率，最大速度（命令码 8……12）**

**5.1.2.1、Y 轴复位速度（命令码 08+DAT1）**

- DAT1: Y 轴复位速度;
- 5.1.2.2、Y 轴独立运动, 直线插补, 圆弧插补的起跳频率 (命令码 09+DAT1+DAT2+DAT3)
- DAT1: Y 轴独立运动的起跳频率;
- DAT2: Y 轴直线插补运动的起跳频率;
- DAT3: Y 轴脉冲插补运动的起跳频率。
- 5.1.2.3、Y 轴独立直线运动的加速度 A, 匀速, 加速度 B (命令码 10+DAT1+DAT2+DAT3)
- DAT1: Y 轴直线运动的加速度 A, 即加速过程的加速度;
- DAT2: Y 轴直线运动匀速频率
- DAT3: Y 轴直线运动的加速度 B, 即减速过程的加速度。
- 5.1.2.4、YZ 直线插补运动的加速度 A, 匀速, 加速度 B (命令码 11+DAT1+DAT2+DAT3)
- DAT1: YZ 轴直线插补运动的加速度 A, 即加速过程的加速度;
- DAT2: YZ 轴直线插补运动匀速频率
- DAT3: YZ 轴直线插补运动的加速度 B, 即减速过程的加速度。
- 5.1.2.5、YZ 圆弧插补运动的加速度 A, 匀速, 加速度 B (命令码 12+DAT1+DAT2+DAT3)
- DAT1: YZ 轴圆弧插补运动的加速度 A, 即加速过程的加速度;
- DAT2: YZ 轴圆弧插补运动匀速频率
- DAT3: YZ 轴圆弧插补运动的加速度 B, 即减速过程的加速度。
- 5.1.3、设置 Z 轴加速度, 起跳频率, 最大速度 (命令码 13……15)
- 5.1.3.1、Z 轴复位速度 (命令码 13+DAT1)
- DAT1: Z 轴复位速度。
- 5.1.3.2、Z 轴独立运动的起跳频率 (命令码 14+DAT1)
- DAT1: Z 轴独立运动的起跳频率。
- 5.1.3.3、Z 轴独立直线运动的加速度 A, 匀速, 加速度 B (命令码 15+DAT1+DAT2+DAT3)
- DAT1: Z 轴直线运动的加速度 A, 即加速过程的加速度;
- DAT2: Z 轴直线运动匀速频率
- DAT3: Z 轴直线运动的加速度 B, 即减速过程的加速度。
- 5.1.4、设置各轴手动参数和反向复位距离 S (命令码 16+DAT1)
- 5.1.4.1、设置 X 轴手动为电平方式的脉冲速度;
- DAT1 的第 1 字节为 31H, 后 6 字节为脉冲速度。
- 5.1.4.2、设置 Y 轴手动为电平方式的脉冲速度;
- DAT1 的第 1 字节为 32H, 后 6 字节为脉冲速度。
- 5.1.4.3、设置 Z 轴手动为电平方式的脉冲速度;
- DAT1 的第 1 字节为 33H, 后 6 字节为脉冲速度。
- 5.1.4.4、设置 X 轴手动为脉冲方式的点动距离;
- DAT1 的第 1 字节为 34H, 后 6 字节为点动距离。
- 5.1.4.5、设置 Y 轴手动为脉冲方式的点动距离;
- DAT1 的第 1 字节为 35H, 后 6 字节为点动距离。
- 5.1.4.6、设置 Z 轴手动为脉冲方式的点动距离;
- DAT1 的第 1 字节为 36H, 后 6 字节为点动距离。
- 5.1.4.7、设置 X 轴反向复位距离 S<sub>x</sub>;
- DAT1 的第 1 字节为 37H, 后 6 字节为反向复位距离 S<sub>x</sub>。
- X 轴反向复位距离 S<sub>x</sub>: 参见 5.4.1 X 轴复位运动。
- 5.1.4.8、设置 Y 轴反向复位距离 S<sub>y</sub>;
- DAT1 的第 1 字节为 38H, 后 6 字节为反向复位距离 S<sub>y</sub>。
- Y 轴反向复位距离 S<sub>y</sub>: 参见 5.4.2 Y 轴复位运动。
- 5.1.4.9、设置 Z 轴反向复位距离 S<sub>z</sub>;

DAT1 的第 1 字节为 39H, 后 6 字节为反向复位距离 Sz。

Z 轴反向复位距离 Sz: 参见 5.4.3 Z 轴复位运动。

#### 5.1.5、恢复默认的控制参数, 保存当前的控制参数 (命令码 17+DAT1)

##### 5.1.5.1、恢复默认的控制参数, 控制器默认参数见附录

发送: DAT1= (30H 30H 30H 30H 30H 30H 31H);

##### 5.1.5.2、保存当前的控制参数到 EEPROM

发送: DAT1= (30H 30H 30H 30H 30H 30H 32H)

#### 5.1.6、紧急制动、平滑制动命令 (命令码 18+DAT1)

##### “紧急制动”:

!!! 一般不使用, 要使用紧急制动建议使用外部 IO 线输入制动信号。

使用 DAT1 作为制动代码, 外部启动条件仅使用 38H:

DAT1= (30H 30H 30H 30H 30H 30H 30H), 三轴全部紧急制动止。

DAT1= (30H 30H 30H 30H 30H 30H 31H), 复位三轴紧急制动。

DAT1= (30H 30H 30H 30H 30H 30H 32H), X 轴紧急制动。

DAT1= (30H 30H 30H 30H 30H 30H 33H), 复位 X 轴紧急制动。

DAT1= (30H 30H 30H 30H 30H 30H 34H), Y 轴紧急制动。

DAT1= (30H 30H 30H 30H 30H 30H 35H), 复位 Y 轴紧急制动。

DAT1= (30H 30H 30H 30H 30H 30H 36H), Z 轴紧急制动。

DAT1= (30H 30H 30H 30H 30H 30H 37H), 复位 Y 轴紧急制动。

当接收到“紧急制动命令”后, 马上执行制动;

当接收到制动复位命令时, 继续执行通信缓冲区中的命令。

接收到的“紧急制动命令”和“复位制动命令”不进入通信缓冲区。

##### “平滑制动”:

接收到“平滑制动命令”后, 命令不进入通信缓冲区, 马上执行平滑制动命令。

DAT1= (30H 30H 30H 30H 30H 30H 38H), X, Y, Z 轴从当前速度以各自轴的加速度 B 减速到起跳频率后停止。

DAT1= (30H 30H 30H 30H 30H 30H 39H), X 轴从当前速度以加速度 B 减速到起跳频率后停止。

DAT1= (30H 30H 30H 30H 30H 30H 41H), Y 轴从当前速度以加速度 B 减速到起跳频率后停止。

DAT1= (30H 30H 30H 30H 30H 30H 42H), Z 轴从当前速度以加速度 B 减速到起跳频率后停止。

DAT1= (30H 30H 30H 30H 30H 30H 43H), 复位 X, Y, Z 轴平滑制动。

DAT1= (30H 30H 30H 30H 30H 30H 44H), 复位 X 轴平滑制动。

DAT1= (30H 30H 30H 30H 30H 30H 45H), 复位 Y 轴平滑制动。

DAT1= (30H 30H 30H 30H 30H 30H 46H), 复位 Z 轴平滑制动。

##### 其他说明:

1、当执行直线 XY, XZ 插补时, 如果需要制动, 必须对 X 轴进行制动。

2、当执行直线 YZ 插补时, 如果需要制动, 必须对 Y 轴进行制动。

3、当执行圆弧 XY, XZ 插补时, 如果需要制动, 必须对 X 轴进行制动。

4、当执行圆弧 YZ 插补时, 如果需要制动, 必须对 Y 轴进行制动。

#### 5.1.7、控制命令 (命令码 19+DAT1)

##### 5.1.7.1、等待命令

等待一段时间, 以毫秒为单位。

DAT1 的第 1 个字节为 31H, 后 6 个字节为等待时间。

举例: 等待 1 秒, 1000 毫秒; DAT1: (31H, 30H, 30H, 30H, 33H, 45H, 38H)

##### 5.1.7.2、(保留备用)

##### 5.1.7.3、复位命令的序号

DAT1 的第 1 字节为 33H, 执行本条命令后, 本条命令的序号为 01, 控制器接收的下一

条命令的序号为 02，配合 5.3.8，可以查询命令执行到哪一条。

#### 5.1.7.4、清除命令缓冲区命令

DAT1 的第 1 字节为 34H，执行本条命令后，结束所有命令的执行，控制器停止脉冲输出。其功能相当于“紧急制动”命令，只不过它执行完毕后会清除命令缓冲区中的命令。

**本命令的“启动条件”仅以为 38H，不进入通信缓冲区。**

#### 5.1.7.5、通信环路测试命令

DAT1 的第 1 字节为 35H，执行本条命令后，控制器没有任何操作，只是响应命令，以测试通信电路的硬件是否正确连接。

**本命令的“启动条件”仅以为 38H，不进入通信缓冲区。**

#### 5.1.7.6、电机脱机命令

DAT1 的第 1 字节为 36H，第 7 字节为脱机电机的序号，如下：

- (1) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 30H): Xfree、Yfree、Zfree 强置为低；
- (2) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 31H): Xfree 强置为低；
- (3) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 32H): Yfree 强置为低；
- (4) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 33H): Zfree 强置为低；
- (5) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 34H): Xfree、Yfree、Zfree 强置为高；
- (6) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 35H): Xfree 强置为高；
- (7) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 36H): Yfree 强置为高；
- (8) DAT1=(36H, 30H, 30H, 30H, 30H, 30H, 37H): Zfree 强置为高；

#### 5.1.7.7、信号 CP 强置命令

DAT1 的第 1 字节为 37H，如下：

- (1) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 30H): Xcp、Ycp、Zcp 强置为低；
- (2) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 31H): Xcp 强置为低；
- (3) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 32H): Ycp 强置为低；
- (4) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 33H): Zcp 强置为低；
- (5) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 34H): Xcp、Ycp、Zcp 强置为高；
- (6) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 35H): Xcp 强置为高；
- (7) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 36H): Ycp 强置为高；
- (8) DAT1=(37H, 30H, 30H, 30H, 30H, 30H, 37H): Zcp 强置为高；

#### 5.1.7.8、信号 DIR 强置命令

DAT1 的第 1 字节为 38H，如下：

- (1) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 30H): Xdir、Ydir、Zdir 强置为低；
- (2) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 31H): Xdir 强置为低；
- (3) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 32H): Ydir 强置为低；
- (4) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 33H): Zdir 强置为低；
- (5) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 34H): Xdir、Ydir、Zdir 强置为高；
- (6) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 35H): Xdir 强置为高；
- (7) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 36H): Ydir 强置为高；
- (8) DAT1=(38H, 30H, 30H, 30H, 30H, 30H, 37H): Zdir 强置为高；

#### 5.1.7.9、信号 REVA, REVB, REVC, REVD 强置命令（命令码 19+DAT1）

- (1) 置低电平命令，DAT1=第 1 字节为 43H，第 2-6 字节为 30H  
第 7 字节：REVA 对应 30H，REVB 对应 31H，REVC 对应 32H，REVD 对应 33H。
- (2) 置低电平命令，DAT1=第 1 字节为 44H，第 2-6 字节为 30H  
第 7 字节：REVA 对应 30H，REVB 对应 31H，REVC 对应 32H，REVD 对应 33H。

## 5.2、轴位置参数设置命令

**说明：所有“轴位置参数设置命令”的响应为“写”响应。**



**5.2.1、设置X轴软件限位寄存器（命令码 21+DAT1）**

DAT1: 设置正向软件限位寄存器, 第 1 字节 30H, 后 6 字节为正方向限位值;  
设置负向软件限位寄存器, 第 1 字节 31H, 后 6 字节为负方向限位值;  
当后 6 字节为“(30H, 30H, 30H, 30H, 30H, 30H)”时, 软件限位无效。

**5.2.2、设置Y轴软件限位寄存器（命令码 22+DAT1）**

同 5.2.1。

**5.2.3、设置Z轴软件限位寄存器（命令码 23+DAT1）**

同 5.2.1。

**5.2.4、设置X轴当前位置寄存器（命令码 24+DAT1）**

DAT1: 第 1 字节为方向信息, 后 6 字节为位置信息;  
当后 6 字节为(30H, 30H, 30H, 30H, 30H, 30H)时, 设置当前坐标位置为“0”, 经常配合复位命令使用;

**5.2.5、设置Y轴当前位置寄存器（命令码 25+DAT1）**

同 5.2.4。

**5.2.6、设置Z轴当前位置寄存器（命令码 26+DAT1）**

同 5.2.4。

**5.3、轴状态和运动参数查询命令**

**说明:** 所有“轴状态和运动参数查询命令”的响应为“读”响应。“启动条件”保留, 功能无效。  
所有查询命令不进入“通信缓冲区”。

**5.3.1、查询X轴状态（命令码 31+DAT1）****5.3.1.1、查询 X 轴位置**

发送: DAT1=(30H, 30H, 30H, 30H, 30H, 30H, 31H)  
响应: 02H + 地址码 + 33H 31H + DAT1 + 启动条件 + 03H + BCC  
DAT1 第 1 字节为方向信息, 30H 表示正向, 31H 表示负向, 后 6 字节为位置信息。

**5.3.1.2、查询 X 轴输入信号**

发送: DAT1=(30H, 30H, 30H, 30H, 30H, 30H, 32H)  
响应: 02H + 地址码 + 33H 31H + DAT1 + 启动条件 + 03H + BCC  
DAT1 第 1 字节为 XZJS 的状态, 30H 表示 XZJS 闭合, 31H 表示 XZJS 断开;  
DAT1 第 2 字节为 XLJS 的状态, 30H 表示 XLJS 闭合, 31H 表示 XLJS 断开;  
DAT1 第 3 字节为 XHJS 的状态, 30H 表示 XHJS 闭合, 31H 表示 XHJS 断开;  
DAT1 第 4 字节为 XEJS 的状态, 30H 表示 XEJS 闭合, 31H 表示 XEJS 断开;  
DAT1 第 5 字节为 Xcp 的状态, 30H 表示 Xcp 为低电平, 31H 表示 Xcp 为高电平;  
DAT1 第 6 字节为 Xdir 的状态, 30H 表示 Xdir 为低电平, 31H 表示 Xdir 为高电平;  
DAT1 第 7 字节为 Xfree 的状态, 30H 表示 Xfree 为低电平, 31H 表示 Xfree 为高电平。

**5.3.1.3、查询 X、Y、Z 轴是否运动中**

发送: DAT1=(30H, 30H, 30H, 30H, 30H, 30H, 33H)  
响应: 02H + 地址码 + 33H 31H + DAT1 + 启动条件 + 03H + BCC  
DAT1 第 5 字节为 X 轴状态, 30H 表示 X 轴停止状态, 31H 表示 X 轴运动中;  
DAT1 第 6 字节为 Y 轴状态, 30H 表示 Y 轴停止状态, 31H 表示 Y 轴运动中;  
DAT1 第 7 字节为 Z 轴状态, 30H 表示 Z 轴停止状态, 31H 表示 Z 轴运动中。

**5.3.1.4、查询 X, Y, Z 当前坐标**

发送: DAT1=(30H, 30H, 30H, 30H, 30H, 30H, 34H)  
响应: 02H+ 地址码 + 33H 31H +DAT1+DAT2+DAT3+ 启动条件 +03H+ BCC  
DAT1: X 轴当前坐标, 第 1 字节为方向信息, 后 6 字节为位置信息;  
DAT2: Y 轴当前坐标, 第 1 字节为方向信息, 后 6 字节为位置信息;  
DAT3: Z 轴当前坐标, 第 1 字节为方向信息, 后 6 字节为位置信息。



**5.3.1.5、查询 REVE, REVF, REVG, REGH 输入状态**

发送: DAT1= (30H, 30H, 30H, 30H, 30H, 30H, 35H)

响应: 格式同 5.3.1.1。

DAT1 第 4 字节为 REVE 的状态, 30H 表示 REVE 闭合, 31H 表示 REVE 断开;

DAT1 第 5 字节为 REVF 的状态, 30H 表示 REVF 为低电平, 31H 表示 REVF 为高电平;

DAT1 第 6 字节为 REVG 的状态, 30H 表示 REVG 为低电平, 31H 表示 REVG 为高电平;

DAT1 第 7 字节为 REVG 的状态, 30H 表示 REVG 为低电平, 31H 表示 REVG 为高电平。

**5.3.1.6、查询 REV\_AIN1、AIN2 模拟量转换结果 (10 位 A/D 转换)**

发送: DAT1= (30H, 30H, 30H, 30H, 30H, 30H, 36H)

响应: 格式同 5.3.1.4。

DAT1: REV\_AIN1 模拟量转换结果;

DAT2: REV\_AIN2 模拟量转换结果;

DAT3: 保留。

**5.3.2、查询Y轴状态 (命令码 32+DAT1)**

同 5.3.1。

**5.3.3、查询Z轴状态 (命令码 33+DAT1)**

同 5.3.1。

**5.3.4、查询X轴运动参数 (命令码 34+DAT1)****5.3.4.1、读出 X 轴复位速度**

发送: DAT1= (30H 30H 30H 30H 30H 30H 31H)

响应: 02H+(地址码) + (命令码 33H+34H) +DAT1+ (启动条件) +03H+ (校验码)

DAT1 为 X 轴的复位速度;

**5.3.4.2、读出 X 轴独立运动, 直线插补, 圆弧插补的起跳频率**

发送: DAT1= (30H 30H 30H 30H 30H 30H 32H)

响应: 02H+(地址码)+(命令码 33H+34H)+DAT1+DAT2+DAT3+(启动条件)+03H+(校验码)

DAT1: X 轴独立运动的起跳频率;

DAT2: X 轴直线插补运动的起跳频率;

DAT3: X 轴圆弧插补运动的起跳频率。

**5.3.4.3、读出 X 轴独立直线运动的加速度 A, 匀速, 加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 33H)

响应: 格式同 5.3.4.2。

DAT1: X 轴直线运动的加速度 A, 即加速过程的加速度;

DAT2: X 轴直线运动匀速频率

DAT3: X 轴直线运动的加速度 B, 即减速过程的加速度。

**5.3.4.4、读出 XY 直线插补运动的加速度 A, 匀速, 加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 34H)

响应: 格式同 5.4.4.2。

DAT1: XY 轴直线插补运动的加速度 A, 即加速过程的加速度;

DAT2: XY 轴直线插补运动匀速频率

DAT3: XY 轴直线插补运动的加速度 B, 即减速过程的加速度。

**5.3.4.5、读出 XZ 直线插补运动的加速度 A, 匀速, 加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 35H)

响应: 格式同 5.3.4.2。

DAT1: XZ 轴直线插补运动的加速度 A, 即加速过程的加速度;

DAT2: XZ 轴直线插补运动匀速频率

DAT3: XZ 轴直线插补运动的加速度 B, 即减速过程的加速度。

**5.3.4.6、读出 XY 圆弧插补运动的加速度 A，匀速，加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 36H)

响应: 格式同 5.3.4.2。

DAT1: XY 轴圆弧插补运动的加速度 A，即加速过程的加速度；

DAT2: XY 轴圆弧插补运动匀速频率；

DAT3: XY 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**5.3.4.7、读出 XZ 圆弧插补运动的加速度 A，匀速，加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 37H)

响应: 格式同 5.3.4.2。

DAT1: XZ 轴圆弧插补运动的加速度 A，即加速过程的加速度；

DAT2: XZ 轴圆弧插补运动匀速频率

DAT3: XZ 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**5.3.5、查询Y轴运动参数（命令码 35+DAT1）**

**5.3.5.1、读出 Y 轴复位速度**

发送: DAT1= (30H 30H 30H 30H 30H 30H 31H)

响应: 格式同 5.3.4.1。

DAT1 为 Y 轴的复位速度；

**5.3.5.2、读出 Y 轴独立运动，直线插补，圆弧插补的起跳频率**

发送: DAT1= (30H 30H 30H 30H 30H 30H 32H)

响应: 格式同 5.3.4.2。

DAT1: Y 轴独立运动的起跳频率；

DAT2: Y 轴直线插补运动的起跳频率；

DAT3: Y 轴圆弧插补运动的起跳频率。

**5.3.5.3、读出 Y 轴独立直线运动的加速度 A，匀速，加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 33H)

响应: 格式同 5.3.4.2。

DAT1: Y 轴直线运动的加速度 A，即加速过程的加速度；

DAT2: Y 轴直线运动匀速频率

DAT3: Y 轴直线运动的加速度 B，即减速过程的加速度。

**5.3.5.4、读出 YZ 直线插补运动的加速度 A，匀速，加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 34H)

响应: 格式同 5.3.4.2。

DAT1: YZ 轴直线插补运动的加速度 A，即加速过程的加速度；

DAT2: YZ 轴直线插补运动匀速频率

DAT3: YZ 轴直线插补运动的加速度 B，即减速过程的加速度。

**5.3.5.5、读出 YZ 圆弧插补运动的加速度 A，匀速，加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 35H)

响应: 格式同 5.3.4.2。

DAT1: YZ 轴圆弧插补运动的加速度 A，即加速过程的加速度；

DAT2: YZ 轴圆弧插补运动匀速频率

DAT3: YZ 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**5.3.6、查询Z轴运动参数（命令码 36+DAT1）**

**5.3.6.1、读出 Z 轴复位速度**

发送: DAT1= (30H 30H 30H 30H 30H 30H 31H)

响应: 格式同 5.3.4.1。

DAT1 为 Z 轴的复位速度；

**5.3.6.2、读出 Z 轴独立运动的起跳频率**

发送: DAT1= (30H 30H 30H 30H 30H 30H 32H)

响应: 格式同 5.3.4.2。

DAT1: Z 轴独立运动的起跳频率;

**5.3.6.3、读出 Z 轴独立直线运动的加速度 A, 匀速, 加速度 B**

发送: DAT1= (30H 30H 30H 30H 30H 30H 33H)

响应: 格式同 5.3.4.3。

DAT1: Z 轴直线运动的加速度 A, 即加速过程的加速度;

DAT2: Z 轴直线运动匀速频率

DAT3: Z 轴直线运动的加速度 B, 即减速过程的加速度。

**5.3.7、读出控制器当前手动、联机状态 (命令码 37+DAT1)****5.3.7.0、读出控制器手动、联机状态**

发送: DAT1= (30H 30H 30H 30H 30H 30H 30H), 读出控制器手动/联机状态。

响应: DAT1= (30H 30H 30H 30H 30H 30H): 手动状态;

DAT1= (30H 30H 30H 30H 30H 31H): 联机状态。

**5.3.7.1、读出控制器 X 轴手动为电平方式的脉冲速度**

发送: DAT1= (30H 30H 30H 30H 30H 30H 31H)。

响应: DAT1 为 X 轴手动为电平方式的脉冲速度。

**5.3.7.2、读出控制器 Y 轴手动为电平方式的脉冲速度**

发送: DAT1= (30H 30H 30H 30H 30H 30H 32H)。

响应: DAT1 为 Y 轴手动为电平方式的脉冲速度。

**5.3.7.3、读出控制器 Z 轴手动为电平方式的脉冲速度**

发送: DAT1= (30H 30H 30H 30H 30H 30H 33H)。

响应: DAT1 为 Z 轴手动为电平方式的脉冲速度。

**5.3.7.4、读出控制器 X 轴手动为脉冲方式的点动距离**

发送: DAT1= (30H 30H 30H 30H 30H 30H 34H)。

响应: DAT1 为 X 轴手动为脉冲方式的点动距离。

**5.3.7.5、读出控制器 Y 轴手动为脉冲方式的点动距离**

发送: DAT1= (30H 30H 30H 30H 30H 30H 35H)。

响应: DAT1 为 Y 轴手动为脉冲方式的点动距离。

**5.3.7.6、读出控制器 Z 轴手动为脉冲方式的点动距离**

发送: DAT1= (30H 30H 30H 30H 30H 30H 36H)。

响应: DAT1 为 Z 轴手动为脉冲方式的点动距离。

**5.3.7.7、读出控制器 X 轴反向复位距离 Sx**

发送: DAT1= (30H 30H 30H 30H 30H 30H 37H)。

响应: DAT1 为 X 轴反向复位距离 Sx。

**5.3.7.8、读出控制器 Y 轴反向复位距离 Sy**

发送: DAT1= (30H 30H 30H 30H 30H 30H 38H)。

响应: DAT1 为 Y 轴反向复位距离 Sy。

**5.3.7.9、读出控制器 Z 轴反向复位距离 Sz**

发送: DAT1= (30H 30H 30H 30H 30H 30H 39H)。

响应: DAT1 为 Z 轴反向复位距离 Sz。

**5.3.7.A、读出控制器 X 轴正向软件限位寄存器 XZMAX**

发送: DAT1= (30H 30H 30H 30H 30H 30H 41H)。

响应: DAT1 为 X 轴正向软件限位寄存器 XZMAX。

**5.3.7.B、读出控制器 X 轴负向软件限位寄存器 XFMAX**

发送: DAT1= (30H 30H 30H 30H 30H 30H 42H)。

响应: DAT1 为 X 轴负向软件限位寄存器 XFMAX。

**5.3.7.C、读出控制器 Y 轴正向软件限位寄存器 YZMAX**

发送: DAT1= (30H 30H 30H 30H 30H 30H 43H)。

响应: DAT1 为 Y 轴正向软件限位寄存器 YZMAX。

**5.3.7.D、读出控制器 Y 轴负向软件限位寄存器 YFMAX**

发送: DAT1= (30H 30H 30H 30H 30H 30H 44H)。

响应: DAT1 为 Y 轴负向软件限位寄存器 YFMAX。

**5.3.7.E、读出控制器 Z 轴正向软件限位寄存器 ZZMAX**

发送: DAT1= (30H 30H 30H 30H 30H 30H 45H)。

响应: DAT1 为 Z 轴正向软件限位寄存器 ZZMAX。

**5.3.7.F、读出控制器 Z 轴负向软件限位寄存器 ZFMAX**

发送: DAT1= (30H 30H 30H 30H 30H 30H 46H)。

响应: DAT1 为 Z 轴负向软件限位寄存器 ZFMAX。

**5.3.8、命令执行情况的查询 (命令码 38+DAT1)**

**5.3.8.0、查询当前正在执行的命令的序号**

发送: DAT1= (30H 30H 30H 30H 30H 30H 30H)

响应: DAT1 为正在执行的命令的序号。

(1) DAT1 的第 1 字节为 30H 时, 控制器处于正常执行状态。

第 2, 3, 4, 5, 6, 7 字节为正在执行的命令序号。

(2) DAT1 的第 1 字节为 31H-37H 时, 表示控制器处于故障状态。

为 31H 时, 表示 X 轴下限位故障;

为 32H 时, 表示 X 轴上限位故障;

为 33H 时, 表示 Y 轴下限位故障;

为 34H 时, 表示 Y 轴上限位故障;

为 35H 时, 表示 Z 轴下限位故障;

为 36H 时, 表示 Z 轴上限位故障;

为 37H 时, 表示控制器紧急停止故障。

第 2, 3, 4, 5, 6, 7 字节为当前正在执行的命令序号。

**5.3.8.1、查询控制器接收到的有效命令数量**

发送: DAT1= (31H 30H 30H 30H 30H 30H 30H)

响应: DAT1 为控制器接收到的有效命令数量。

**5.3.8.2、(保留备用)**

**5.4、复位运动命令和独立运动命令**

说明: 所有“复位运动和独立运动命令”的响应为“写“响应”。

**5.4.1、X轴复位运动 (命令码 41+DAT1)**

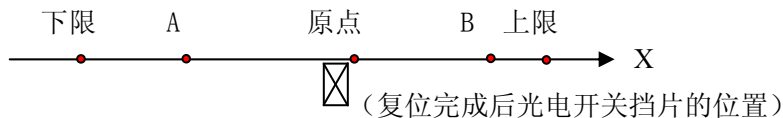
**5.4.1.1、X轴向上原点复位运动**

DAT1= (30H 30H 30H 30H 30H 30H 31H)

当 X 在原点的下方 (A) 时, 执行本命令, X 向原点运动, 碰到原点停止。

当 X 在原点的上方 (B) 时, 执行本命令, X 向上运动, 碰到上限点停止, 反向运动到下

图示位置, 原点复位完成。



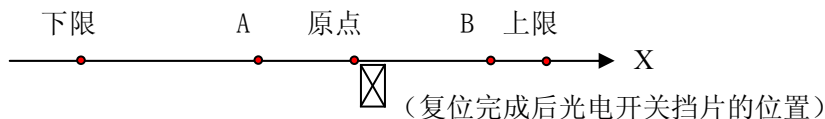
提示: X 轴复位起点在 B 时, 如果没有设置上限信号, 会导致故障, 即冲出上限。

**5.4.1.2、X轴向下原点复位运动**

DAT1= (30H 30H 30H 30H 30H 30H 32H);

当 X 在原点的上方 (B) 时, 执行本命令, X 向原点运动, 碰到原点停止。

当 X 在原点的下方 (A) 时, 执行本命令, X 向下运动, 碰到下限点停止, 反向运动直到到下图示位置, 原点复位完成。

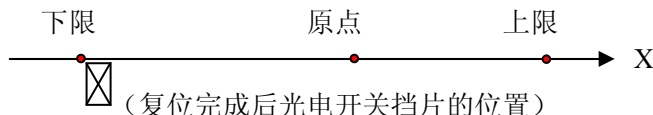


**提示:** X 轴复位起点在 A 时, 如果没有设置下限信号, 会导致故障, 即冲出下限。

#### 5.4.1.3、X 轴下限位复位运动

DAT1= (30H 30H 30H 30H 30H 30H 33H);

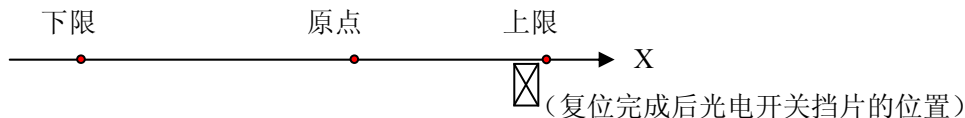
X 向下运动, 直到碰到下限点停止。



#### 5.4.1.4、X 轴上限位复位运动

DAT1= (30H 30H 30H 30H 30H 30H 34H)。

X 向上运动, 直到碰到上限点停止。



**反向复位距离 S:** 如果复位刚开始时, 复位开关已经动作, 则反向运动直至离开复位开关, 继续运动距离 S (默认 50 步) 后, 按复位运动命令的方向复位。本参数为保证复位的准确性。

建议用户在一上电的时候用起跳频率找到原点, 并设置为原点坐标。

此后的复位运动可以根据当前坐标位置, 先快速运动 (使用独立运动命令) 到原点附近, 然后慢速复位 (使用复位命令), 以保证复位的准确性。

**复位运动过程中不会改变“轴当前位置寄存器”的值。也不受“轴软件位置限位寄存器”值的限制。**

### 5.4.2、Y轴复位运动 (命令码 42+DAT1)

#### 5.4.2.1、Y 轴向上原点复位运动

DAT1= (30H 30H 30H 30H 30H 30H 31H);

#### 5.4.2.2、Y 轴向下原点复位运动

DAT1= (30H 30H 30H 30H 30H 30H 32H);

#### 5.4.2.3、Y 轴下限位复位运动

DAT1= (30H 30H 30H 30H 30H 30H 33H);

#### 5.4.2.4、Y 轴上限位复位运动

DAT1= (30H 30H 30H 30H 30H 30H 34H);

参考 X 轴复位。

### 5.4.3、Z轴复位运动 (命令码 43+DAT1)

#### 5.4.3.1、Z 轴向上原点复位运动

DAT1= (30H 30H 30H 30H 30H 30H 31H);

#### 5.4.3.2、Z 轴向下原点复位运动

DAT1= (30H 30H 30H 30H 30H 30H 32H);

#### 5.4.3.3、Z 轴下限位复位运动

DAT1= (30H 30H 30H 30H 30H 30H 33H);

#### 5.4.3.4、Z 轴上限位复位运动

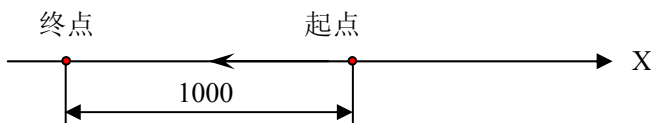
DAT1= (30H 30H 30H 30H 30H 30H 34H);

参考 X 轴复位。

5.4.4、X轴独立运动（命令码 44+DAT1）

DAT1: 第 1 字节为方向信息，30H 表示正向，31H 表示负向。  
后 6 字节为运动终点的相对坐标值。

举例：



DAT1= (31H, 30H, 30H, 30H, 33H, 45H, 38H)

5.4.5、Y轴独立运动（命令码 45+DAT1）

同 X 轴独立运动。

5.4.6、Z轴独立运动（命令码 46+DAT1）

同 X 轴独立运动。

5.5、直线插补命令

说明：所有“直线插补”命令的响应为“写”响应。

5.5.1、XY直线插补运动（命令码 51+DAT1+DAT2）

DAT1、DAT2 各由七个字节组成，构成直线运动插补终点与起点的相对坐标。

DAT1 存放 X 轴的相对坐标，DAT2 存放 Y 轴的相对坐标。

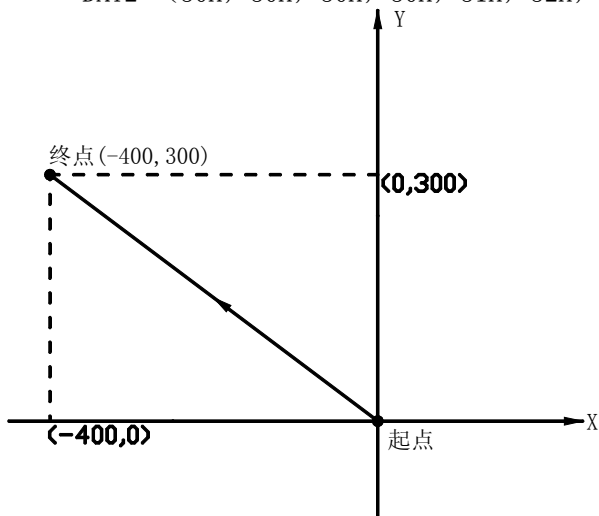
第 1 字节为方向信息，30H 表示正向，31H 表示负向。

后 6 字节为运动终点相对坐标值，最大 16777215 个脉冲当量。

举例：

DAT1= (31H, 30H, 30H, 30H, 31H, 39H, 30H)

DAT2= (30H, 30H, 30H, 30H, 31H, 32H, 43H)



5.5.2、XZ直线插补运动（命令码 52+DAT1+DAT2）

DAT1 存放 X 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

其它同 5.5.1。

5.5.3、YZ直线插补运动（命令码 53+DAT1+DAT2）

DAT1 存放 Y 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

其它同 5.5.1。

5.5.4、XYZ直线插补运动（命令码 57+DAT1+DAT2+DAT3）

DAT1 存放 X 轴的相对坐标，DAT2 存放 Y 轴的相对坐标。

DAT3 存放 Z 轴的相对坐标。

5.6、圆弧插补命令

说明：所有“圆弧插补命令”的响应为“写”响应。

5.6.1、XY圆弧插补运动（命令码 61+DAT1+DAT2+DAT3）

DAT1、DAT2：圆弧运动插补圆心与起点的相对坐标。

DAT1 存放 X 轴的相对坐标，DAT2 存放 Y 轴的相对坐标。

第 1 字节为方向信息，30H 表示正向，31H 表示负向。

后 6 字节为圆心的相对坐标值，最大 16777215 个脉冲当量。

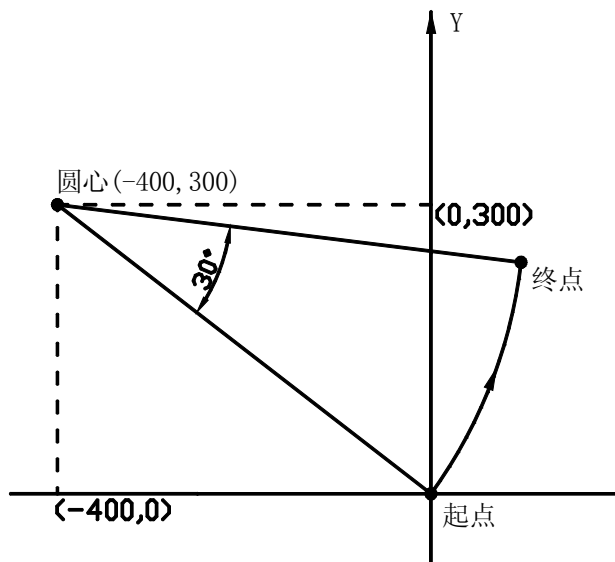
DAT3：圆弧运动的角度

第 1 字节为方向信息，30H 表示逆时针插补，31H 表示顺时针插补。

后 6 字节为角度值(放大 100 倍)，如 30.67 度，放大 100 倍，即 3067，十六进制为 BFBH，

ASC\_II 码为 42H, 46H, 42H。最大值为 167772.15 度。如果圆弧运动的角度为 (30H, 30H, 30H, 30H, 30H, 30H)，将做无限角度的圆弧插补运动，直到使用紧急命令停止。

举例：



DAT1= (31H, 30H, 30H, 30H, 30H, 31H, 39H, 30H)

DAT2= (30H, 30H, 30H, 30H, 30H, 31H, 32H, 43H)

DAT3= (30H, 30H, 30H, 30H, 30H, 42H, 42H, 38H)

5.6.2、XZ圆弧插补运动（命令码 62+DAT1+DAT2+DAT3）

DAT1 存放 X 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

其它同 5.6.1。

5.6.3、YZ圆弧插补运动（命令码 63+DAT1+DAT2+DAT3）

DAT1 存放 Y 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

同 5.6.1。

5.7、样条插补命令

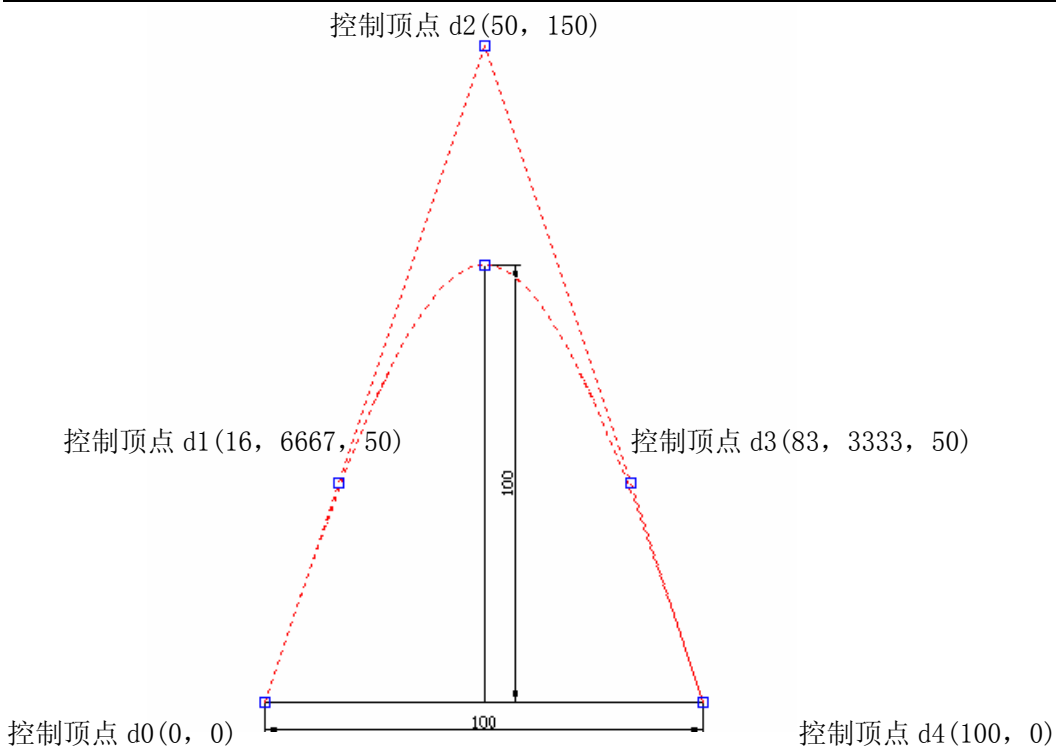
5.7.1、命令格式

此命令使控制器能够插补基于控制顶点（最多 500 个）的样条曲线，并且具有先进的基于“拐角加速度限制”的“速度前瞻控制”。

例一：

发送到 EASYSMC2200 的命令如下：

```
00, 70, 00007D0, 0003E80, 0004E20, 4 //样条插补运动参数
00, 75, 0015D63, 0003415, 0009C40, 4 //实体 1, 样条插补
00, 76, 002BAC5, 0009C40, 001D4C0, 4
00, 77, 0017700, 001046B, 0009C40, 4 //样条插补拐角加速度
00, 78, 00000A0, 0013880, 0000000, 4 //样条插补弓高误差
```



解释如下：

根据样条控制理论，上图的控制顶点 5 个，节点向量 9 个

(1) 控制顶点计算

采用相对坐标，控制顶点 0 的 X, Y 坐标为 0

假设脉冲当量为 0.00125mm

x0=0,  
y0=0, (不发送)

x1=16.6667/0.00125=13333,      转换 0003415 (Hex)  
y1=50.0000/0.00125=40000,      转换 0009C40 (Hex)

x2=50.0000/0.00125=40000,      转换 0009C40 (Hex)  
y2=150.0000/0.00125=120000,    转换 001D4C0 (Hex)

x3=83.3333/0.00125=66667,      转换 001046B (Hex)  
y3=50.0000/0.00125=40000,      转换 0009C40 (Hex)

x4=100.0000/0.00125=80000,      转换 0013880 (Hex)  
y4=0.0000/0.00125=0,            转换 0000000 (Hex)

(2) 节点向量计算

u0=0, u1=0, u2=0, u3=0; (不发送)  
u4=111.8034/0.00125=89443,      转换 0015D63 (Hex)  
u5=223.6068/0.00125=17885,      转换 002BAC5 (Hex)  
u6=u5, u7=u5, u8=u5 (不发送)

(1)00, 70, 00007D0, 0003E80, 0004E20, 4 //样条插补运动参数



dat1: 起跳频率  
 dat2: 加速度  
 dat3: 最高频率

(2)00, 75, 0015D63, 0003415, 0009C40, 4 //实体 1, 样条插补, 节点向量+第一个控制顶点坐标

dat1: 节点向量 u4  
 dat2: 控制顶点 1, 相对坐标 X1  
 dat3: 控制顶点 1, 相对坐标 Y1

(3)00, 76, 002BAC5, 0009C40, 001D4C0, 4, 节点向量+第二个控制顶点坐标

dat1: 节点向量 u5  
 dat2: 控制顶点 2, 相对坐标 X2  
 dat3: 控制顶点 2, 相对坐标 Y2

备注: 如果控制顶点坐标大于 5 个, 使用命令 76 继续插入节点向量和控制顶点坐标

(4)00, 77, 0017700, 001046B, 0009C40, 4 //样条插补拐角加速度+倒数第二个控制顶点坐标

dat1: 样条插补拐角加速度  
 dat2: 控制顶点 3, 相对坐标 X3  
 dat3: 控制顶点 3, 相对坐标 Y3

(5)00, 78, 00000A0, 0013880, 0000000, 4 //样条插补弓高误差+倒数第一个控制顶点坐标

dat1: 样条插补弓高误差  
 dat2: 控制顶点 4, 相对坐标 X4  
 dat3: 控制顶点 4, 相对坐标 Y4

### 5.7.2、样条同步输出功能

当正在插补一条样条曲线的时候, 需要在样条中的某一点打开一个 OUT 点, 运动到另一点, 关闭同一个 OUT 点, 这种情况下, 使用样条同步输出功能。

命令格式为: 19+DAT1, DAT1=X XX XX XX,

DAT1 的第 1 字节为 45H, 即字符 'E' 的 ASC\_II 码

DAT1 的第 2、3 字节: 定义开始输出位置, 样条正向算起第 XX 个型值点打开 OUT 点

DAT1 的第 4、5 字节: 定义停止输出位置, 样条反向算起第 XX 个型值点关闭 OUT 点

DAT1 的第 6、7 字节: 定义使用那个 OUT 点, 03 使用 XFREE, 06 使用 YFREE, 09 使用 ZFREE,

0A 使用 REVA, 0B 使用 REVB。

## 5.8、其它

### 5.8.1、计数器 (2 个)

#### 5.8.1.1、设置计数器: (命令码 19 + DAT1)

DAT1 的第 1 个字节为 39H, 即字符 '9' 的 ASC\_II 码

第 2-3 字节为计数器代号,

第 4-7 字节为计数器值 (0-65536)。

备注: EasySmc 上电自动将计数器清零。

举例: 设置计数器 1 为 100, 则 DAT1 = 9 01 0064。

ASC II 码为: 39H, 30H, 31H, 30H, 30H, 36H, 34H。

响应: 同“写”响应。

#### 5.8.1.2、计数器加减“1”操作 (命令号 19 + DAT1)

DAT1 的第 1 字节为 41H, 即字符 'A' 的 ASC\_II 码

第 2-3 字节为计数器代号，  
第 7 字节为“0”，对指定的计数器加“1”，  
为“1”，对指定的计数器减“1”。

响应：同“写”响应。

#### 5.8.1.3、读计数器值（命令号 38 + DAT1）

DAT1 的第 1 字节为 33H，  
第 2-3 字节为计数器代号。

响应：

DAT1 的第 1 个字节为 33H，  
第 2-3 字节为计数器代号，  
第 4-7 字节为计数器值（0-65536）。

### 5.8.2、程序下载、上载、脱机执行

准备工作，断开 A/M，闭合 M\_L/H，上电。

#### 5.8.2.1、进入下载状态（命令码 19 + DAT1）

DAT1：为 B 0 0 0 0 0 0，响应格式同写响应。

#### 5.8.2.2、发送“下载内容”到 EasySmc 2100，发送最多 62 条命令。

如果发送第 63 条命令，会响应“缓冲区”满错误。

#### 5.8.2.3、发送程序结束标志（命令码 19 + DAT1）

DAT1：为 B 0 0 0 0 F F，响应格式同写响应。

#### 5.8.2.4、选择扇区写入下载内容（命令码 19 + DAT1）

DAT1：为 B 0 0 0 0 0 1 —— B 0 0 0 0 0 8，  
当前控制器可以下载的命令数量为 62\*8=496 条。

#### 5.8.2.5、查询下载状态（命令码 38 + DAT1）

DAT1：第 1 字节为 34H，第 2 字节为 31H，其余字节为 30H。

返回 4 1 0 0 0 0 0 时，不处于下载状态，即 A/M 跳线没有闭合。

返回 4 1 0 0 0 0 1 时，控制器处于下载状态，可以发送下载内容。

#### 5.8.2.6、查询写入是否成功（命令码 38 + DAT1）

DAT1：第 1 字节为 34H，第 2 字节为 32H，其余字节为 30H。

返回 4 2 0 0 0 0 0 时，写入失败。

返回 4 2 0 0 0 0 1 时，写入成功。

#### 5.8.2.7、读取控制器 ROM 程序（命令码 38 + DAT1）

DAT1：第 1 字节为 35H，第 2-7 字节为命令序号。

响应：控制器中存储的程序。

#### 5.8.2.8、脱机执行控制器 ROM 中的程序

按下 M\_XZ，执行 ROM 程序过程中，不允许再往控制器发送写命令，只允许发送读命令。

按下 M\_ZZ，正在执行的程序紧急停止运行。

按下 M\_XF，执行 X 轴复位，复位方式需要定制。

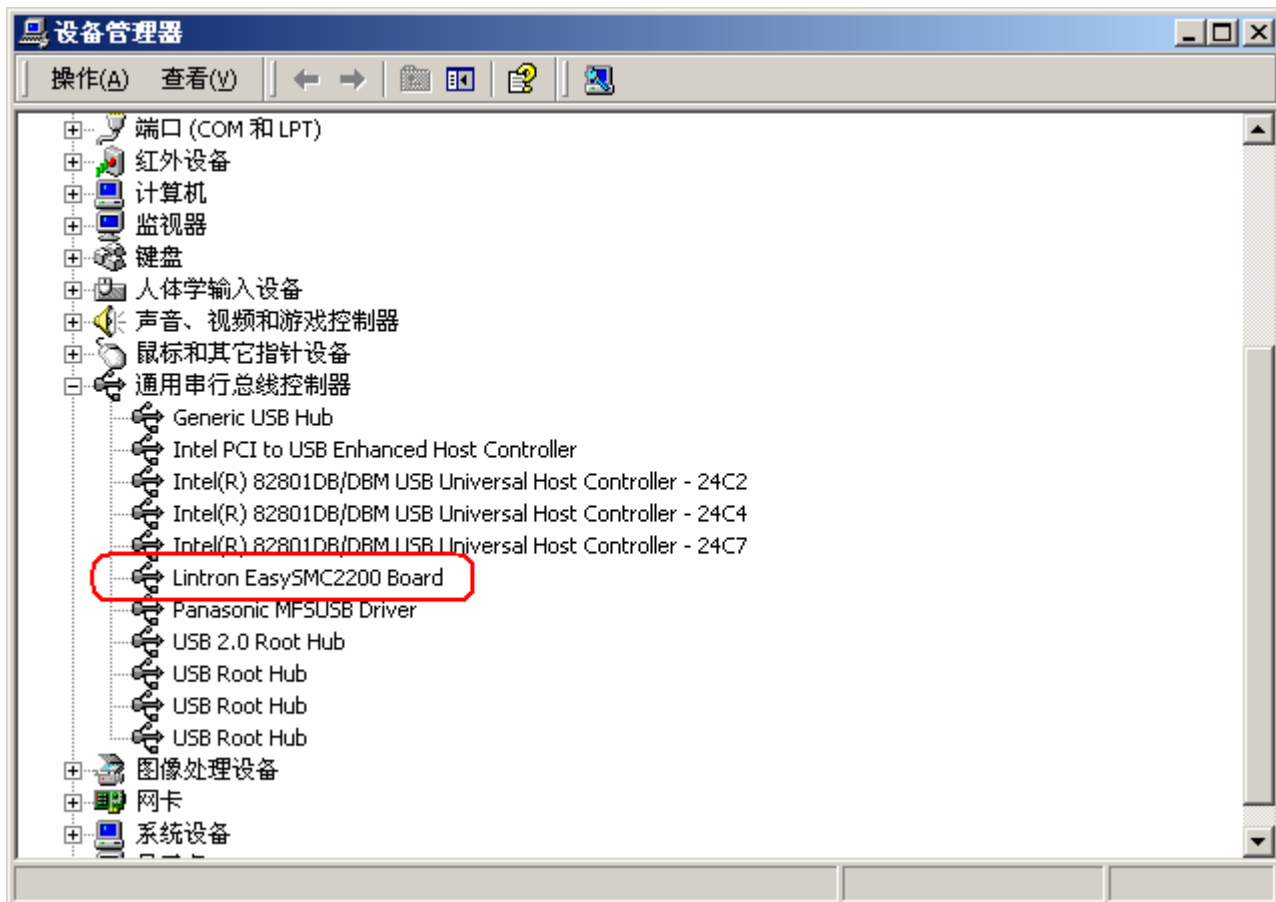
按下 M\_YF，执行 Y 轴复位，复位方式需要定制。

按下 M\_ZF，执行 Z 轴复位，复位方式需要定制。

## 六、USB驱动程序安装和EM2200. DLL调用

### 6.1、USB驱动程序安装

控制器上电以后，插入 USB 接线，会提示安装 USB 驱动程序，选择本公司提供的 D12test.inf 文件，操作系统任务栏会出现“Lintron EasySMC2200 Board”，没有正确的安装驱动，PC 和控制器无法正常通信。



### 6.2、EM2200. DLL调用

控制器的 USB 通信使用 EM2200. dll 文件提供的两个函数：

#### 6.2.1、Delphi用法如下

```

type
    blink=^byte;
    .....
    function senddata(psend:blink;addr:byte;len:integer):integer;
stdcall;external 'EM2200_DLL.dll';//PC 写数据命令
    function recdata (prec:blink;addr:byte;cmd:byte;rdat1:byte;rdat2:integer):integer;
stdcall;external 'EM2200_DLL.dll';//PC 读数据命令
  
```

senddata 入口参数：

- psend 是指向 byte 的指针
- addr 是保留地址，目前固定为\$00
- len 是写数据长度，目前固定为 16 字节；

recdata 入口参数说明：

- prec 是指向 byte 的指针
- addr 是保留地址，目前固定为\$00

cmd 是读数据的命令代号  
 rdat1 为读命令子数据 1  
 rdat2 为读命令子数据 2。

举例 1: 要让 EasySmc2200 的 X 轴走 1000 个脉冲

(备注: 在 DELPHI 中, \$00 表示十六进制数 00)

```
psend: array [0..63] of byte;//psend 填充格参考 6.2.4.1
psend[00]:=$01;//写命令代码
psend[01]:=$00;//地址
psend[02]:=$44;//命令号
psend [03]:=$00;//dat1 第 1 字节
psend [04]:=$00;//dat1 第 2 字节
psend [05]:=$03;//dat1 第 3 字节
psend [06]:=$E8;//dat1 第 4 字节
psend [15]:=$00;//启动代码
ret:=senddata(@psend,$00,16); //addr 固定为$00, len 固定为 16;
//响应的数据有 20 字节, 参考 6.2.4.2
```

举例 2: 读 X 轴复位速度

(备注: 1、在 DELPHI 中, \$00 表示十六进制数 00)

2、prec 需要是指向大小最少 16 的 byte 数组; addr 固定为 0;)

```
prec:array [0..15] of byte;//定义接收数据的数组
ret:integer;
ret:=recdata(@prec,$00,$34,$01,$00);
//命令码为$34, rdat1 为$01, rdat2 不使用, 为任意值;
//响应数据将存储在 readbuf 中, 格式参考 6.2.4.3
```

### 6.2.2、Visual C++ 6.0 用法如下

DLL 是用 Delphi 编写的, 所以 C++使用动态调用 DLL

```
typedef long (_stdcall * lpsenddata)(unsigned char *psend, unsigned char addr, long len);
typedef long (_stdcall * lprecdata)(unsigned char *prec, unsigned char addr, unsigned char
cmd, unsigned char rdat1, int rdat2);
.....
HINSTANCE hD11; //句柄
lpsenddata senddata;//函数指针类型
lprecdata recdata;//函数指针类型
.....
hD11 = LoadLibrary("EM2200_DLL.d11");
senddata = (lpSendFun)GetProcAddress(hD11, "senddata");
recdata = (lpRecFun)GetProcAddress(hD11, "recdata");
.....
unsigned char psend [64];//发送缓冲区
unsigned char prec [16];//接收缓冲区
long result;
.....
result = senddata (psend, 0x00, 16); //写数据命令
```

```
result = recdata (prec, 0x00, 0x31, 0x02, 16); //读数据命令
.....
```

```
FreeLibrary(hD11); //释放 hD11
```

**6.2.3、Visual Basic 6.0 用法如下**

```
Private Declare Function senddata Lib "EM2200_DLL.dll" _
(ByRef psend As Byte, ByVal addr As Byte, ByVal len As Long) As Long
```

```
Private Declare Function recdata Lib "EM2200_DLL.dll" _
(ByRef prec As Byte, ByVal addr As Byte, ByVal cmd As Byte, ByVal rdat1 As Byte, _
ByVal rdat2 As Long) As Long
```

**6.2.4、写数据和读数据时 psend 和 prec 的数据格式**

**6.2.4.1、写命令函数 senddata (psend:blink;addr:byte;len:integer) 补充说明**

psend 指向从 PC 发送到控制器的 16 字节数据，DAT2、DAT3 可选。

00	01	02	03、04、05、06	07、08、09、10	11、12、13、14	15
写代号 01	地址 00	命令码	DAT1	DAT2	DAT3	启动代码

- 说明：(1) 第 00 字节为写代号，固定为 01；  
 (2) 第 01 字节为地址，固定为 00；  
 (3) 第 02 字节为命令码，如设置 X 轴复位速度是 00；  
 (4) 如果没有 DAT2，和 DAT3，不要理会，无需填充；  
 (5) 启动代码固定放在第 15 字节。

**6.2.4.2、写响应，psend 格式**

psend 指向从控制器返回给 PC 的 20 字节数据。

00	01	02	03	04...07	08...11	12...15	16...19
地址 00	命令码	错误代码	忙状态	正执行行号	X 轴当前坐标	Y 轴当前坐标	Z 轴当前坐标

- 说明：
- (1) 正常响应的情况下，命令码应该和写的命令码相同，如果是\$99，则有两种情况
    - A, 第 02 字节为 02，控制器处于手动状态，需要检查跳线；
    - B, 第 02 字节为 03，控制器缓冲区满，需要重新发送刚发送的命令；
  - (2) 忙状态为 1 字节，最低位 1sb0 为 ZBUSY，0 表示 Z 轴不忙，1 表示 Z 轴忙；  
 低位 1sb1 为 YBUSY，0 表示 Y 轴不忙，1 表示 Y 轴忙；  
 低位 1sb2 为 XBUSY，0 表示 X 轴不忙，1 表示 X 轴忙；
  - (3) 第 04、05、06、07 字节标志控制器正在执行的命令序号  
 需要指出的是，当发送第一条命令的时候，控制器返回的是上一阶段的最后一条命令的序号，应该舍去，发送第二条命令的时候，控制器将返回正在执行的命令序号。  
 当所有命令发送完毕时，如果还需要检测各轴状态，可以使用 00, 19, 5000000, 0，本条命令是测试命令，不会影响控制器运行，而可以得到控制器的执行状态。
  - (4) 08...11: X 轴当前坐标；12...15: Y 轴当前坐标；16...19: Z 轴当前坐标；  
 4 个字节中，第 1 字节为符号代码，00 表示正，01 表示负；  
 第 2、3、4 字节为坐标值。

**6.2.4.3、读响应，prec 格式**

prec 指向从控制器返回给 PC 的 16 字节数据。

00	01	02、03、04、05	06、07、08、09	10、11、12、13	14	15
地址 00	命令码	DAT1	DAT2	DAT3	启动代码	DAT 个数

- 说明：(1) 第 00 字节固定为地址 00；  
 (2) 第 01 字节为命令码，和读命令 PC 发送的命令码相同；如果是\$99，表示读命令错误；  
 (3) 响应回来的数据 DAT1, DAT2, DAT3；

- (4) 原样返回的启动代码;
- (5) 第 15 字节为有效 DAT 个数, 为 01 时, 表示有效数据为 DAT1, (上载程序时有效)  
为 02 时, 表示有效数据为 DAT1, DAT2,  
为 03 时, 表示有效数据为 DAT1, DAT2, DAT3。

## 七、USB写命令清单

### 7.1、初始化命令

#### 7.1.1、设置X轴速度参数 (命令码\$01……\$07)

##### 7.1.1.1、X轴复位速度 (命令码\$01, DAT1)

DAT1: X轴复位速度; (单位: Hz)

##### 7.1.1.2、X轴独立运动, 直线插补, 圆弧插补的起跳频率 (命令码\$02, DAT1+DAT2+DAT3)

DAT1: X轴独立运动的起跳频率;

DAT2: X轴直线插补运动的起跳频率;

DAT3: X轴圆弧插补运动的起跳频率。

##### 7.1.1.3、X轴独立直线运动的加速度 A, 匀速, 加速度 B (命令码\$03+DAT1+DAT2+DAT3)

DAT1: X轴直线运动的加速度 A, 即加速过程的加速度; (单位: Hz/s)

DAT2: X轴直线运动匀速频率

DAT3: X轴直线运动的加速度 B, 即减速过程的加速度。

##### 7.1.1.4、XY 直线插补运动的加速度 A, 匀速, 加速度 B (命令码\$04+DAT1+DAT2+DAT3)

DAT1: XY轴直线插补运动的加速度 A, 即加速过程的加速度;

DAT2: XY轴直线插补运动匀速频率

DAT3: XY轴直线插补运动的加速度 B, 即减速过程的加速度。

##### 7.1.1.5、XZ 直线插补运动的加速度 A, 匀速, 加速度 B (命令码\$05+DAT1+DAT2+DAT3)

DAT1: XZ轴直线插补运动的加速度 A, 即加速过程的加速度;

DAT2: XZ轴直线插补运动匀速频率

DAT3: XZ轴直线插补运动的加速度 B, 即减速过程的加速度。

##### 7.1.1.6、XY 圆弧插补运动的加速度 A, 匀速, 加速度 B (命令码\$06+DAT1+DAT2+DAT3)

DAT1: XY轴圆弧插补运动的加速度 A, 即加速过程的加速度;

DAT2: XY轴圆弧插补运动匀速频率

DAT3: XY轴圆弧插补运动的加速度 B, 即减速过程的加速度。

##### 7.1.1.7、XZ 圆弧插补运动的加速度 A, 匀速, 加速度 B (命令码\$07+DAT1+DAT2+DAT3)

DAT1: XZ轴圆弧插补运动的加速度 A, 即加速过程的加速度;

DAT2: XZ轴圆弧插补运动匀速频率

DAT3: XZ轴圆弧插补运动的加速度 B, 即减速过程的加速度。

#### 7.1.2、设置Y轴速度参数 (命令码\$08……\$12)

##### 7.1.21、Y轴复位速度 (命令码\$08+DAT1)

DAT1: Y轴复位速度;

##### 7.1.22、Y轴独立运动, 直线插补, 圆弧插补的起跳频率 (命令码\$09+DAT1+DAT2+DAT3)

DAT1: Y轴独立运动的起跳频率;

DAT2: Y轴直线插补运动的起跳频率;

DAT3: Y轴脉冲插补运动的起跳频率。

##### 7.1.23、Y轴独立直线运动的加速度 A, 匀速, 加速度 B (命令码\$10+DAT1+DAT2+DAT3)

DAT1: Y轴直线运动的加速度 A, 即加速过程的加速度;

DAT2: Y轴直线运动匀速频率

DAT3: Y轴直线运动的加速度 B, 即减速过程的加速度。

##### 7.1.24、YZ 直线插补运动的加速度 A, 匀速, 加速度 B (命令码\$11+DAT1+DAT2+DAT3)

DAT1: YZ 轴直线插补运动的加速度 A, 即加速过程的加速度;

DAT2: YZ 轴直线插补运动匀速频率

DAT3: YZ 轴直线插补运动的加速度 B, 即减速过程的加速度。

#### 7.1.25、YZ 圆弧插补运动的加速度 A, 匀速, 加速度 B (命令码\$12+DAT1+DAT2+DAT3)

DAT1: YZ 轴圆弧插补运动的加速度 A, 即加速过程的加速度;

DAT2: YZ 轴圆弧插补运动匀速频率

DAT3: YZ 轴圆弧插补运动的加速度 B, 即减速过程的加速度。

### 7.1.3、设置Z轴速度参数 (命令码\$13……\$15)

#### 7.1.3.1、Z 轴复位速度 (命令码\$13+DAT1)

DAT1: Z 轴复位速度;

#### 7.1.3.2、Z 轴独立运动的起跳频率 (命令码\$14+DAT1)

DAT1: Z 轴独立运动的起跳频率;

#### 7.1.3.3、Z 轴独立直线运动的加速度 A, 匀速, 加速度 B (命令码\$15+DAT1+DAT2+DAT3)

DAT1: Z 轴直线运动的加速度 A, 即加速过程的加速度;

DAT2: Z 轴直线运动匀速频率

DAT3: Z 轴直线运动的加速度 B, 即减速过程的加速度。

### 7.1.4、设置手动参数和反向复位距离 (命令码\$16+DAT1)

#### 7.1.4.1、设置 X 轴手动为电平方式的脉冲速度

DAT1 的第 1 字节为\$01, 后 3 字节为脉冲速度。

#### 7.1.4.2、设置 Y 轴手动为电平方式的脉冲速度

DAT1 的第 1 字节为\$02, 后 3 字节为脉冲速度。

#### 7.1.4.3、设置 Z 轴手动为电平方式的脉冲速度

DAT1 的第 1 字节为\$03, 后 3 字节为脉冲速度。

#### 7.1.4.4、设置 X 轴手动为脉冲方式的点动距离

DAT1 的第 1 字节为\$04, 后 3 字节为点动距离。

#### 7.1.4.5、设置 Y 轴手动为脉冲方式的点动距离

DAT1 的第 1 字节为\$05, 后 3 字节为点动距离。

#### 7.1.4.6、设置 Z 轴手动为脉冲方式的点动距离

DAT1 的第 1 字节为\$06, 后 3 字节为点动距离。

#### 7.1.4.7、设置 X 轴反向复位距离 S<sub>x</sub>

DAT1 的第 1 字节为\$07, 后 3 字节为反向复位距离 S<sub>x</sub>。

#### 7.1.4.8、设置 Y 轴反向复位距离 S<sub>y</sub>

DAT1 的第 1 字节为\$08, 后 3 字节为反向复位距离 S<sub>y</sub>。

#### 7.1.4.9、设置 Z 轴反向复位距离 S<sub>z</sub>

DAT1 的第 1 字节为\$09, 后 3 字节为反向复位距离 S<sub>z</sub>。

### 7.1.5、恢复默认的控制参数, 保存当前的控制参数 (命令码\$17+DAT1)

#### 7.1.5.1、恢复默认的控制参数, 控制器默认参数见附录

发送: DAT1= 00 00 00 01 ;

#### 7.1.5.2、保存当前的控制参数到 EEPROM (命令码\$17+DAT1)

发送: DAT1= 00 00 00 02

### 7.1.6、紧急制动、平滑制动命令 (命令码\$18+DAT1)

“紧急制动”:

!!! 一般不使用, 要使用紧急制动建议使用外部 I/O 线输入制动信号。

使用 DAT1 作为制动代码, 外部启动条件仅使用 38H:

DAT1= (00 00 00 00), 三轴全部紧急制动止。

DAT1= (00 00 00 01), 复位三轴紧急制动。

DAT1= (00 00 00 02), X 轴紧急制动。  
 DAT1= (00 00 00 03), 复位 X 轴紧急制动。  
 DAT1= (00 00 00 04), Y 轴紧急制动。  
 DAT1= (00 00 00 05), 复位 Y 轴紧急制动。  
 DAT1= (00 00 00 06), Z 轴紧急制动。  
 DAT1= (00 00 00 07), 复位 Y 轴紧急制动。

当接收到“紧急制动命令”后，马上执行制动；

当接收到制动复位命令时，继续执行通信缓冲区中的命令。

接收到的“紧急制动命令”和“复位制动命令”不进入通信缓冲区。

#### “平滑制动”：

接收到“平滑制动命令”后，命令不进入通信缓冲区，马上执行平滑制动命令。

DAT1= (00 00 00 08), X, Y, Z 轴从当前速度以各自轴的加速度 B 减速到起跳频率后停止。  
 DAT1= (00 00 00 09), X 轴从当前速度以加速度 B 减速到起跳频率后停止。  
 DAT1= (00 00 00 0A), Y 轴从当前速度以加速度 B 减速到起跳频率后停止。  
 DAT1= (00 00 00 0B), Z 轴从当前速度以加速度 B 减速到起跳频率后停止。  
 DAT1= (00 00 00 0C), 复位 X, Y, Z 轴平滑制动。  
 DAT1= (00 00 00 0D), 复位 X 轴平滑制动。  
 DAT1= (00 00 00 0E), 复位 Y 轴平滑制动。  
 DAT1= (00 00 00 0F), 复位 Z 轴平滑制动。

#### 其他说明：

- 1、当执行直线 XY, XZ 插补时，如果需要制动，必须对 X 轴进行制动。
- 2、当执行直线 YZ 插补时，如果需要制动，必须对 Y 轴进行制动。
- 3、当执行圆弧 XY, XZ 插补时，如果需要制动，必须对 X 轴进行制动。
- 4、当执行圆弧 YZ 插补时，如果需要制动，必须对 Y 轴进行制动。

### 7.1.7、控制命令（命令码\$19+DAT1）

#### 7.1.7.1、等待命令

等待一段时间，以毫秒为单位。

DAT1 的第 1 个字节为 \$01，后 3 个字节为等待时间。

举例：等待 1 秒，1000 毫秒；DAT1: (01 00 03 E8)

#### 7.1.7.2、（保留备用）

#### 7.1.7.3、复位命令的序号

DAT1 的第 1 字节为 \$03，执行本条命令后，本条命令的序号为 01，控制器接收的下一条命令的序号为 02，配合 9.8.0，可以查询命令执行到哪一条。

#### 7.1.7.4、清除命令缓冲区命令

DAT1 的第 1 字节为 \$04，执行本条命令后，结束所有命令的执行，控制器停止脉冲输出。

其功能相当于“紧急制动”命令，只不过它执行完毕后会清除命令缓冲区中的命令。

**本命令的“启动条件”仅以为 38H，不进入通信缓冲区。**

#### 7.1.7.5、通信环路测试命令

DAT1 的第 1 字节为 \$05，执行本条命令后，控制器没有任何操作，只是响应命令，以测试通信电路的硬件是否正确连接。

**本命令的“启动条件”仅以为 38H，不进入通信缓冲区。**

#### 7.1.7.6、电机脱机命令

DAT1 的第 1 字节为 \$06，第 7 字节为脱机电机的序号，如下：

- (1) DAT1= (06 00 00 00): Xfree、Yfree、Zfree 强置为低；
- (2) DAT1= (06 00 00 01): Xfree 强置为低；
- (3) DAT1= (06 00 00 02): Yfree 强置为低；



- (4) DAT1= (06 00 00 03): Zfree 强置为低;
- (5) DAT1= (06 00 00 04): Xfree、Yfree、Zfree 强置为高;
- (6) DAT1= (06 00 00 05): Xfree 强置为高;
- (7) DAT1= (06 00 00 07): Yfree 强置为高;
- (8) DAT1= (06 00 00 08): Zfree 强置为高;

#### 7.1.7.7、信号 CP 强置命令

DAT1 的第 1 字节为 \$07, 如下:

- (1) DAT1= (07 00 00 00): Xcp、Ycp、Zcp 强置为低;
- (2) DAT1= (07 00 00 01): Xcp 强置为低;
- (3) DAT1= (07 00 00 02): Ycp 强置为低;
- (4) DAT1= (07 00 00 03): Zcp 强置为低;
- (5) DAT1= (07 00 00 04): Xcp、Ycp、Zcp 强置为高;
- (6) DAT1= (07 00 00 05): Xcp 强置为高;
- (7) DAT1= (07 00 00 06): Ycp 强置为高;
- (8) DAT1= (07 00 00 07): Zcp 强置为高;

#### 7.1.7.8、信号 DIR 强置命令

DAT1 的第 1 字节为 \$08, 如下:

- (1) DAT1= (08 00 00 00): Xdir、Ydir、Zdir 强置为低;
- (2) DAT1= (08 00 00 01): Xdir 强置为低;
- (3) DAT1= (08 00 00 02): Ydir 强置为低;
- (4) DAT1= (08 00 00 03): Zdir 强置为低;
- (5) DAT1= (08 00 00 04): Xdir、Ydir、Zdir 强置为高;
- (6) DAT1= (08 00 00 05): Xdir 强置为高;
- (7) DAT1= (08 00 00 06): Ydir 强置为高;
- (8) DAT1= (08 00 00 07): Zdir 强置为高;

#### 7.1.7.9、信号 REVA, REVB, REC, REVD 强置命令

- (1) DAT1= (0C 00 00 00): REVA 强置为低;
- (2) DAT1= (0C 00 00 01): REVB 强置为低;
- (3) DAT1= (0C 00 00 02): REVC 强置为低;
- (4) DAT1= (0C 00 00 03): REVD 强置为低;
- (5) DAT1= (0D 00 00 00): REVA 强置为高;
- (6) DAT1= (0D 00 00 01): REVB 强置为高;
- (7) DAT1= (0D 00 00 02): REVC 强置为高;
- (8) DAT1= (0D 00 00 03): REVD 强置为高;

### 7.2、轴位置参数设置命令

#### 7.2.1、设置X轴软件限位寄存器（命令码\$21+DAT1）

DAT1: 设置正向软件限位寄存器, 第 1 字节\$00, 后 3 字节为正方向限位值;  
设置负向软件限位寄存器, 第 1 字节\$01, 后 3 字节为负方向限位值;  
当后 3 字节为“(00 00 00)”时, 软件限位无效。

#### 7.2.2、设置Y轴软件限位寄存器（命令码\$22+DAT1）

同 7.2.1。

#### 7.2.3、设置Z轴软件限位寄存器（命令码\$23+DAT1）

同 7.2.1。

#### 7.2.4、设置X轴当前位置寄存器（命令码\$24+DAT1）

DAT1: 第 1 字节为方向信息, 后 3 字节为位置信息;  
当后 3 字节为(00 00 00)时, 设置当前坐标位置为“0”, 经常配合复位命令使用;

7.2.5、设置Y轴当前位置寄存器（命令码\$25+DAT1）

同 7.2.4。

7.2.6、设置Z轴当前位置寄存器（命令码\$26+DAT1）

同 7.2.4。

7.3、复位运动和独立运动命令

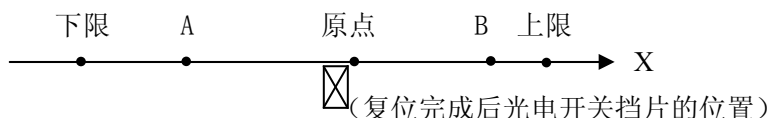
7.3.1、X轴复位运动（命令码\$41+DAT1）

7.3.1.1、X轴向上原点复位运动

DAT1= (00 00 00 01);

当 X 在原点的下方 (A) 时，执行本命令，X 向原点运动，碰到原点停止。

当 X 在原点的上方 (B) 时，执行本命令，X 向上运动，碰到上限点停止，反向运动到下图示位置，原点复位完成。



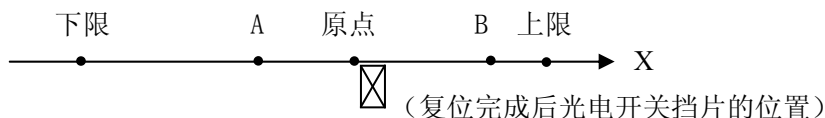
**提示：**X轴复位起点在 B 时，如果没有设置上限信号，会导致故障，即冲出上限。

7.3.1.2、X轴向下原点复位运动

DAT1= (00 00 00 02);

当 X 在原点的上方 (B) 时，执行本命令，X 向原点运动，碰到原点停止。

当 X 在原点的下方 (A) 时，执行本命令，X 向下运动，碰到下限点停止，反向运动直到下图示位置，原点复位完成。

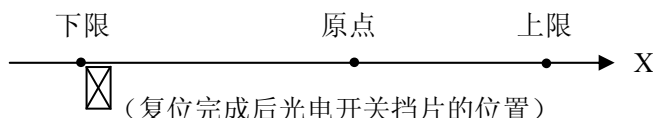


**提示：**X轴复位起点在 A 时，如果没有设置下限信号，会导致故障，即冲出下限。

7.3.1.3、X轴下限位复位运动

DAT1= (00 00 00 03);

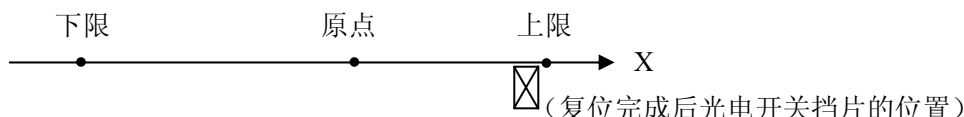
X 向下运动，直到碰到下限点停止。



7.3.1.4、X轴上限位复位运动

DAT1= (00 00 00 04);

X 向上运动，直到碰到上限点停止。



**反向复位距离 S：**如果复位刚开始时，复位开关已经动作，则反向运动直至离开复位开关，继续运动距离 S（默认 50 步）后，按运动复位命令的方向复位。本参数为保证复位的准确性。

建议用户在一上电的时候用起跳频率找到零点，并设置为原点坐标。

以后的复位运动可以根据坐标位置，先快速运动（使用独立运动命令），等到接近零点的时候慢速复位（使用复位命令），以保证复位的准确性。

**复位运动过程中不会改变“轴当前位置寄存器”的值。也不受“轴软件位置限位寄存器”值的限制。**

7.3.2、Y轴复位运动（命令码\$42+DAT1）

7.3.2.1、Y轴向上原点复位运动

DAT1= (00 00 00 01);

7.3.2.2、Y轴向下原点复位运动

DAT1= (00 00 00 02);

7.3.2.3、Y轴下限位复位运动

DAT1= (00 00 00 03);

7.3.2.4、Y轴上限位复位运动

DAT1= (00 00 00 04);

参考 X 轴复位。

7.3.3、Z轴复位运动 (命令码\$43+DAT1)

7.3.3.1、Z轴向上原点复位运动

DAT1= (00 00 00 01);

7.3.3.2、Z轴向下原点复位运动

DAT1= (00 00 00 02);

7.3.3.3、Z轴下限位复位运动

DAT1= (00 00 00 03);

7.3.3.4、Z轴上限位复位运动

DAT1= (00 00 00 04);

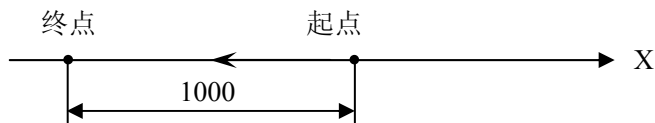
参考同 X 轴复位。

7.3.4、X轴独立运动 (命令码\$44+DAT1)

DAT1: 第 1 字节为方向信息, \$00 表示正向, \$01 表示负向。

后 3 字节为运动终点的相对坐标值。

举例:



DAT1= (01 00 03 E8)

7.3.5、Y轴独立运动 (命令码\$45+DAT1)

同 X 轴独立运动。

7.3.6、Z轴独立运动 (命令码\$46+DAT1)

同 X 轴独立运动。

7.4、直线插补命令

7.4.1、XY直线插补运动 (命令码\$51+DAT1+DAT2)

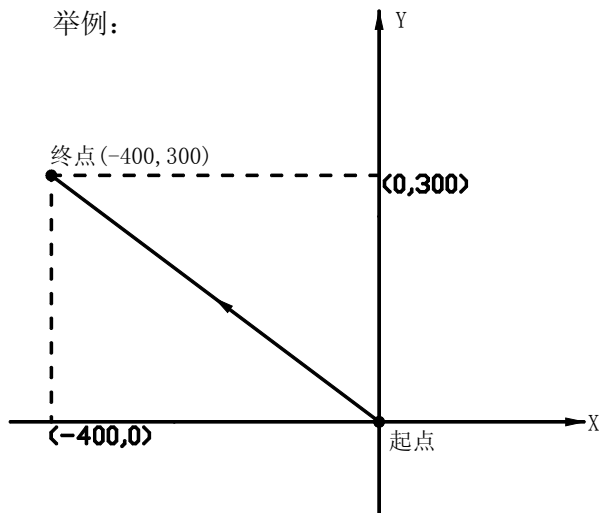
DAT1、DAT2 构成直线运动插补终点与起点的相对坐标。

DAT1 存放 X 轴的相对坐标, DAT2 存放 Y 轴的相对坐标。

第 1 字节为方向信息, \$00 表示正向, \$01 表示负向。

后 3 字节为运动终点相对坐标值, 最大 16777215 个脉冲当量。

举例:



DAT1= (01 00 01 90)

DAT2= (00 00 01 2C)

**7.4.2、XZ直线插补运动（命令码\$52+DAT1+DAT2）**

DAT1 存放 X 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

**7.4.3、YZ直线插补运动（命令码\$53+DAT1+DAT2）**

DAT1 存放 Y 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

**7.4.4、XYZ直线插补运动（命令码\$57+DAT1+DAT2）**

DAT1 存放 X 轴的相对坐标，DAT2 存放 Y 轴的相对坐标，DAT3 存放 Z 轴的相对坐标。

**7.5、圆弧插补命令**

**7.5.1、XY圆弧插补运动（命令码\$61+DAT1+DAT2+DAT3）**

DAT1、DAT2：圆弧运动插补圆心与起点的相对坐标。

DAT1 存放 X 轴的相对坐标，DAT2 存放 Y 轴的相对坐标。

第 1 字节为方向信息，\$00 表示正向，\$01 表示负向。

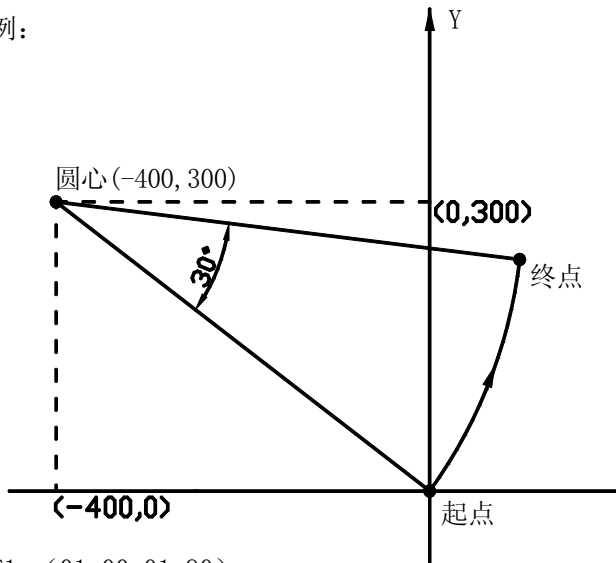
后 3 字节为圆心的相对坐标值，最大 16777215 个脉冲当量。

DAT3：圆弧运动的角度

第 1 字节为方向信息，\$00 表示逆时针插补，\$01 表示顺时针插补。

后 3 字节为角度值(放大 100 倍)，如 30.67 度，放大 100 倍，即 3067，十六进制为 BFBH。

举例：



DAT1= (01 00 01 90)

DAT2= (00 00 01 2C)

DAT3= (00 00 0B B8)

**7.5.2、XZ圆弧插补运动（命令码\$62+DAT1+DAT2+DAT3）**

DAT1 存放 X 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

**7.5.3、YZ圆弧插补运动（命令码\$63+DAT1+DAT2+DAT3）**

DAT1 存放 Y 轴的相对坐标，DAT2 存放 Z 轴的相对坐标。

**样条插补命令**

**7.6、样条插补命令**

**7.6.1、样条插补运动参数（命令码\$70+DAT1+DAT2+DAT3）**

DAT1：起跳频率；

DAT2：加速度；

DAT3：最高频率

**7.6.2、样条插补节点向量+第一个控制顶点坐标（命令码\$75+DAT1+DAT2+DAT3）**

DAT1: 节点向量 u4;

DAT2: 控制顶点 1, 相对坐标 X1;

DAT3: 控制顶点 1, 相对坐标 Y1

**7.6.3、样条插补节点向量+第二个控制顶点坐标（命令码\$76+DAT1+DAT2+DAT3）**

DAT1: 节点向量 u5;

DAT2: 控制顶点 2, 相对坐标 X2;

DAT3: 控制顶点 2, 相对坐标 Y2

备注: 如果控制顶点坐标大于 5 个, 使用命令 76 继续插入节点向量和控制顶点坐标

**7.6.4、样条插补拐角加速度+倒数第二个控制顶点坐标（命令码\$77+DAT1+DAT2+DAT3）**

DAT1: 样条插补拐角加速度;

DAT2: 控制顶点 3, 相对坐标 X3;

DAT3: 控制顶点 3, 相对坐标 Y3

**7.6.5、样条插补弓高误差+倒数第一个控制顶点坐标（命令码\$78+DAT1+DAT2+DAT3）**

DAT1: 样条插补弓高误差;

DAT2: 控制顶点 4, 相对坐标 X4;

DAT3: 控制顶点 4, 相对坐标 Y4

**7.6.6、样条同步输出功能（命令码\$19+DAT1）**

DAT1 的第 1 字节为 \$0E,

DAT1 的第 2 字节: 定义开始输出位置, 样条正向算起第 XX 个型值点打开 OUT 点

DAT1 的第 3 字节: 定义停止输出位置, 样条反向算起第 XX 个型值点关闭 OUT 点

DAT1 的第 4 字节: 定义使用那个 OUT 点, \$03 使用 XFREE, \$06 使用 YFREE, \$09 使用 ZFREE,

\$0A 使用 REVA, \$0B 使用 REVB。

**7.7、其它****7.7.1、计数器（2 个）****7.7.1.1、设置计数器（命令码\$19 + DAT1）**

DAT1 的第 1 个字节为 \$09,

第 2 字节为计数器代号,

第 3、4 字节为计数器值 (0-65536)。

**7.7.2.2、计数器加减“1”操作（命令号\$19 + DAT1）**

DAT1 的第 1 字节为 \$0A,

第 2 字节为计数器代号,

第 3、4 字节为 \$00, \$00, 对指定的计数器加“1”, 为 \$00, \$01 对指定的计数器减“1”。

**7.7.2、程序下载**

准备工作, 断开 A/M, 闭合 M\_L/H, 上电。

**7.7.2.1、进入下载状态（命令码\$19 + DAT1）**

DAT1: 为 \$0B, \$00, \$00, \$00, 响应格式同写响应。

**7.7.2.2、发送“下载内容”到 EasySmc 2200, 发送最多 62 条命令。**

如果发送第 63 条命令, 会响应“缓冲区”满错误。

**7.7.2.3、发送程序结束标志（命令码\$19 + DAT1）**

DAT1: 为 \$0B, \$00, \$00, \$FF, 响应格式同写响应。

**7.7.2.4、选择扇区写入下载内容（命令码\$19 + DAT1）**

DAT1: 为 \$0B, \$00, \$00, \$01 —— \$0B, \$00, \$ 00, \$08。

当前控制器可以下载的命令数量为 62\*8=496 条。

**7.7.2.5、脱机执行控制器 ROM 中的程序**

按下 M\_XZ, 执行 ROM 程序过程中, 不允许再往控制器发送写命令, 只允许发送读命令。

按下 M\_ZZ，正在执行的程序紧急停止运行。

按下 M\_XF，执行 X 轴复位，复位方式需要定制。

按下 M\_YF，执行 Y 轴复位，复位方式需要定制。

按下 M\_ZF，执行 Z 轴复位，复位方式需要定制。

## 八、USB读命令清单

### 8.1、查询X轴状态（命令码\$31，rdat1：\$01……\$06，rdat2 不使用）

#### 8.1.1、查询 X 轴位置信息（命令码\$31，rdat1：\$01，rdat2 不使用）

响应：

DAT1 (02、03、04、05)

第 02 字节为符号信息，\$00 表示符号为正，\$01 表示符号为负；

第 03、04、05 字节为 X 轴当前坐标值；

#### 8.1.2、查询 X 轴输入信号状态（命令码\$31，rdat1：\$02，rdat2 不使用）

响应：

DAT1 (02、03、04、05)

第 02 字节的低 4 位为 XZJS 的状态，\$0 表示 XZJS 闭合，\$1 表示 XZJS 断开；

第 03 字节的高 4 位为 XLJS 的状态，\$0 表示 XLJS 闭合，\$1 表示 XLJS 断开；

第 03 字节的低 4 位为 XHJS 的状态，\$0 表示 XHJS 闭合，\$1 表示 XHJS 断开；

第 04 字节的高 4 位为 XEJS 的状态，\$0 表示 XEJS 闭合，\$1 表示 XEJS 断开；

第 04 字节的低 4 位为 XCP 的状态，\$0 表示 XCP 闭合，\$1 表示 XCP 断开；

第 05 字节的高 4 位为 XDIR 的状态，\$0 表示 XDIR 闭合，\$1 表示 XDIR 断开；

第 05 字节的低 4 位为 XFREE 的状态，\$0 表示 XFREE 闭合，\$1 表示 XFREE 断开；

#### 8.1.3、查询 X 轴，Y 轴，Z 轴是否运动中（命令码\$31，rdat1：\$03，rdat2 不使用）

响应：

DAT1 (02、03、04、05)

第 04 字节的低 4 位为 X 轴运行状态，\$0 表示 X 轴停止状态，\$1 表示 X 轴运动中；

第 05 字节的高 4 位为 Y 轴运行状态，\$0 表示 Y 轴停止状态，\$1 表示 Y 轴运动中；

第 05 字节的低 4 位为 Z 轴运行状态，\$0 表示 Z 轴停止状态，\$1 表示 Z 轴运动中；

#### 8.1.4、查询 X，Y，Z 轴位置信息（命令码\$31，rdat1：\$04，rdat2 不使用）

响应：

DAT1 (02、03、04、05)

第 02 字节为符号信息，\$00 表示符号为正，\$01 表示符号为负；

第 03、04、05 字节为 X 轴当前坐标值；

DAT2 (06、07、08、09)

第 06 字节为符号信息，\$00 表示符号为正，\$01 表示符号为负；

第 07、08、09 字节为 Y 轴当前坐标值；

DAT3 (0A、0B、0C、0D)

第 0A 字节为符号信息，\$00 表示符号为正，\$01 表示符号为负；

第 0B、0C、0D 字节为 Z 轴当前坐标值；

#### 8.1.5、查询 REVA，REVB，REC，REV 轴输入信号状态（命令码\$31，rdat1：\$05，rdat2 不使用）

响应：

DAT1 (02、03、04、05)

第 04 字节的高 4 位为 REVE 的状态，\$0 表示 REVE 闭合，\$1 表示 REVE 断开；

第 04 字节的低 4 位为 REVF 的状态，\$0 表示 REVF 闭合，\$1 表示 REVF 断开；

第 05 字节的高 4 位为 REVG 的状态，\$0 表示 REVG 闭合，\$1 表示 REVG 断开；

第 05 字节的低 4 位为 REVH 的状态，\$0 表示 REVH 闭合，\$1 表示 REVH 断开；

**8.1.6、查询 REV\_AIN1、AIN2 模拟量转换结果（命令码\$31，rdat1: \$06，rdat2 不使用）**

响应:

DAT1: REV\_AIN1 模拟量转换结果;  
DAT2: REV\_AIN2 模拟量转换结果;  
DAT3: 保留。

**8.2、查询Y轴状态（命令码\$32，其它同X轴）****8.3、查询Z轴状态（命令码\$33，其它同X轴）****8.4、查询X轴运动参数（命令码\$34，rdat1: \$01……\$07，rdat2 不使用）****8.4.1、读出 X 轴复位速度（rdat1: \$01）**

响应:

DAT1 (02、03、04、05) 存放 X 轴复位速度;

**8.4.2、读出 X 轴独立运动，直线插补，圆弧插补的起跳频率（rdat1: \$02）**

响应:

DAT1 (02、03、04、05) X 轴独立运动的起跳频率;  
DAT2 (06、07、08、09) X 轴直线插补的起跳频率;  
DAT3 (10、11、12、13) X 轴圆弧插补的起跳频率;

**8.4.3、读出 X 轴独立直线运动的加速度 A，匀速，加速度 B（rdat1: \$03）**

响应:

DAT1: X 轴直线运动的加速度 A，即加速过程的加速度;  
DAT2: X 轴直线运动匀速频率  
DAT3: X 轴直线运动的加速度 B，即减速过程的加速度。

**8.4.4、读出 XY 直线插补运动的加速度 A，匀速，加速度 B（rdat1: \$04）**

响应:

DAT1: XY 轴直线插补运动的加速度 A，即加速过程的加速度;  
DAT2: XY 轴直线插补运动匀速频率  
DAT3: XY 轴直线插补运动的加速度 B，即减速过程的加速度。

**8.4.5、读出 XZ 直线插补运动的加速度 A，匀速，加速度 B（rdat1: \$05）**

响应:

DAT1: XZ 轴直线插补运动的加速度 A，即加速过程的加速度;  
DAT2: XZ 轴直线插补运动匀速频率  
DAT3: XZ 轴直线插补运动的加速度 B，即减速过程的加速度。

**8.4.6、读出 XY 圆弧插补运动的加速度 A，匀速，加速度 B（rdat1: \$06）**

响应:

DAT1: XY 轴圆弧插补运动的加速度 A，即加速过程的加速度;  
DAT2: XY 轴圆弧插补运动匀速频率;  
DAT3: XY 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**8.4.7、读出 XZ 圆弧插补运动的加速度 A，匀速，加速度 B（rdat1: \$07）**

响应:

DAT1: XZ 轴圆弧插补运动的加速度 A，即加速过程的加速度;  
DAT2: XZ 轴圆弧插补运动匀速频率  
DAT3: XZ 轴圆弧插补运动的加速度 B，即减速过程的加速度。

**8.5、查询Y轴运动参数（命令码\$35，rdat1: \$01……\$05，rdat2 不使用）****8.5.1、读出 Y 轴复位速度（rdat1: \$01）**

响应:

DAT1 为 Y 轴的复位速度;

**8.5.2、读出 Y 轴独立运动，直线插补，圆弧插补的起跳频率（rdat1: \$02）**

响应:

- DAT1: Y 轴独立运动的起跳频率;
- DAT2: Y 轴直线插补运动的起跳频率;
- DAT3: Y 轴圆弧插补运动的起跳频率。

**8.5.3、读出 Y 轴独立直线运动的加速度 A, 匀速, 加速度 B (rdat1: \$03)**

响应:

- DAT1: Y 轴直线运动的加速度 A, 即加速过程的加速度;
- DAT2: Y 轴直线运动匀速频率
- DAT3: Y 轴直线运动的加速度 B, 即减速过程的加速度。

**8.5.4、读出 YZ 直线插补运动的加速度 A, 匀速, 加速度 B (rdat1: \$04)**

响应:

- DAT1: YZ 轴直线插补运动的加速度 A, 即加速过程的加速度;
- DAT2: YZ 轴直线插补运动匀速频率
- DAT3: YZ 轴直线插补运动的加速度 B, 即减速过程的加速度。

**8.5.5、读出 YZ 圆弧插补运动的加速度 A, 匀速, 加速度 B (rdat1: \$05)**

响应:

- DAT1: YZ 轴圆弧插补运动的加速度 A, 即加速过程的加速度;
- DAT2: YZ 轴圆弧插补运动匀速频率
- DAT3: YZ 轴圆弧插补运动的加速度 B, 即减速过程的加速度。

**8.6、查询 Z 轴运动参数 (命令码 \$36, rdat1: \$01……\$03, rdat2 不使用)**

**8.6.1、读出 Z 轴复位速度 (rdat1: \$01)**

响应:

- DAT1 为 Z 轴的复位速度;

**8.6.2、读出 Z 轴独立运动的起跳频率 (rdat1: \$02)**

响应:

- DAT1: Z 轴独立运动的起跳频率;

**8.6.3、读出 Z 轴独立直线运动的加速度 A, 匀速, 加速度 B (rdat1: \$03)**

响应:

- DAT1: Z 轴直线运动的加速度 A, 即加速过程的加速度;
- DAT2: Z 轴直线运动匀速频率
- DAT3: Z 轴直线运动的加速度 B, 即减速过程的加速度。

**8.7、读出控制器手动、联机状态 (命令码 \$37, rdat1: \$00……\$0F, rdat2 不使用)**

**8.7.0、读出控制器手动、联机状态 (rdat1: \$00)**

响应:

- DAT1: (02、03、04、05) = \$00 \$00 \$00 \$00, 手动状态
- DAT1: (02、03、04、05) = \$00 \$00 \$00 \$01, 联机状态

**8.7.1、读出控制器 X 轴手动为电平方式的脉冲速度 (rdat1: \$01)**

响应:

- DAT1 为 X 轴手动为电平方式的脉冲速度。

**8.7.2、读出控制器 Y 轴手动为电平方式的脉冲速度 (rdat1: \$02)**

响应:

- DAT1 为 Y 轴手动为电平方式的脉冲速度。

**8.7.3、读出控制器 Z 轴手动为电平方式的脉冲速度 (rdat1: \$03)**

响应:

- DAT1 为 Z 轴手动为电平方式的脉冲速度。



**8.7.4、读出控制器 X 轴手动为脉冲方式的点动距离 (rdat1: \$04)**

响应:

DAT1 为 X 轴手动为脉冲方式的点动距离。

**8.7.5、读出控制器 Y 轴手动为脉冲方式的点动距离 (rdat1: \$05)**

响应:

DAT1 为 Y 轴手动为脉冲方式的点动距离。

**8.7.6、读出控制器 Z 轴手动为脉冲方式的点动距离 (rdat1: \$06)**

响应:

DAT1 为 Z 轴手动为脉冲方式的点动距离。

**8.7.7、读出控制器 X 轴反向复位距离 Sx (rdat1: \$07)**

响应:

DAT1 为 X 轴反向复位距离 Sx。

**8.7.8、读出控制器 Y 轴反向复位距离 Sy (rdat1: \$08)**

响应:

DAT1 为 Y 轴反向复位距离 Sy。

**8.7.9、读出控制器 Z 轴反向复位距离 Sz (rdat1: \$09)**

响应:

DAT1 为 Z 轴反向复位距离 Sz。

**8.7.A、读出控制器 X 轴正向软件限位寄存器 XZMAX (rdat1: \$0A)**

响应:

DAT1 为 X 轴正向软件限位寄存器 XZMAX。

**8.7.B、读出控制器 X 轴负向软件限位寄存器 XFMAX (rdat1: \$0B)**

响应:

DAT1 为 X 轴负向软件限位寄存器 XFMAX。

**8.7.C、读出控制器 Y 轴正向软件限位寄存器 YZMAX (rdat1: \$0C)**

响应:

DAT1 为 Y 轴正向软件限位寄存器 YZMAX。

**8.7.D、读出控制器 Y 轴负向软件限位寄存器 YFMAX (rdat1: \$0D)**

响应:

DAT1 为 Y 轴负向软件限位寄存器 YFMAX。

**8.7.E、读出控制器 Z 轴正向软件限位寄存器 ZZMAX (rdat1: \$0E)**

响应:

DAT1 为 Z 轴正向软件限位寄存器 ZZMAX。

**8.7.F、读出控制器 Z 轴负向软件限位寄存器 ZFMAX (rdat1: \$0F)**

响应:

DAT1 为 Z 轴负向软件限位寄存器 ZFMAX。

**8.8、命令执行情况查询****8.8.0、查询当前正在执行的命令的序号 (命令码\$38, rdat1: \$00, rdat2 不使用)**

响应:

DAT1 (02 03 04 05) 为正在执行的命令的序号。

(1) DAT1 的“02”字节为\$00时, 控制器处于正常执行状态。

第3, 4, 5, 字节为正在执行的命令序号。

(2) DAT1 的“02”字节为\$00-\$07时, 表示控制器处于故障状态。

为 \$01H 时, 表示 X 轴下限位故障;

为 \$02H 时, 表示 X 轴上限位故障;

为 \$03H 时, 表示 Y 轴下限位故障;

为 \$04H 时，表示 Y 轴上限位故障；  
为 \$05H 时，表示 Z 轴下限位故障；  
为 \$06H 时，表示 Z 轴上限位故障；  
为 \$07H 时，表示控制器紧急停止故障。

“02 03 04” 字节为当前正在执行的命令序号。

#### 8.8.1、查询控制器接收到的有效命令数量（命令码\$38， rdat1: \$01， rdat2 不使用）

响应：DAT1 为控制器接收到的有效命令数量。

#### 8.8.2、保留备用（命令码\$38， rdat1: \$02， rdat2 不使用）

#### 8.8.3、读计数器值（命令码\$38， rdat1: \$03， rdat2 使用!）

发送：

rdat1 的第 1 字节为\$03

rdat2 为计数器代号。

响应：

DAT1 计数器值（0-65536）。

#### 8.8.4、查询下载状态、写入状态

##### 8.8.4.1、查询下载状态（命令码\$38， rdat1: \$04， rdat2 使用!）

发送：

rdat1: \$04

rdat2: \$01

响应：

DAT1=\$00, \$00, \$00, \$00 时，不处于下载状态，即 A/M 跳线没有闭合。

DAT1=\$00, \$00, \$00, \$01 时，控制器处于下载状态，可以发送下载内容。

##### 8.8.4.2、查询写入是否成功（命令码\$38， rdat1: \$04， rdat2 使用!）

发送：

rdat1: \$04

rdat2: \$02

响应：

DAT1=\$00, \$00, \$00, \$00 时，写入失败。

DAT1=\$00, \$00, \$00, \$01 时，写入成功。

#### 8.5、读取控制器 ROM 程序（命令码\$38， rdat1: \$05， rdat2 使用!）

发送：

rdat2: 为命令序号。

响应：

DAT1 为控制器中存储的程序，

DAT2 为控制器中存储的程序，

DAT3 为控制器中存储的程序。

第 15 字节为有效 DAT 个数，为 01 时，表示有效数据为 DAT1，

为 02 时，表示有效数据为 DAT1， DAT2，

为 03 时，表示有效数据为 DAT1， DAT2， DAT3。

## 九、故障检查、维修保养、售后服务

自出厂之日起 1 年内免费保修。

# 附录

## 1、默认参数值

序号	名称	参数大小 (DEC)	参数值 (HEX)
5.1.1.1	X 轴复位速度	2000	7D0
5.1.1.2	X 轴独立运动起跳频率	2000	7D0
	X 轴直线插补运动起跳频率	2000	7D0
	X 轴圆弧插补运动起跳频率	2000	7D0
5.1.1.3	X 轴独立直线运动的加速度 A	20,000	4E20
	X 轴独立直线运动的匀速脉冲频率	20,000	4E20
	X 轴独立直线运动的加速度 B	20,000	4E20
5.1.1.4	XY 直线插补运动的加速度 A	20,000	4E20
	XY 直线插补运动的匀速脉冲频率	20,000	4E20
	XY 直线插补运动的加速度 B	20,000	4E20
5.1.1.5	XZ 直线插补运动的加速度 A	20,000	4E20
	XZ 直线插补运动的匀速脉冲频率	20,000	4E20
	XZ 直线插补运动的加速度 B	20,000	4E20
5.1.1.6	XY 圆弧插补运动的加速度 A	20,000	4E20
	XY 圆弧插补运动的匀速脉冲频率	20,000	4E20
	XY 圆弧插补运动的加速度 B	20,000	4E20
5.1.1.7	XZ 圆弧插补运动的加速度 A	20,000	4E20
	XZ 圆弧插补运动的匀速脉冲频率	20,000	4E20
	XZ 圆弧插补运动的加速度 B	20,000	4E20
5.1.2.1	Y 轴复位速度	2000	7D0
5.1.2.2	Y 轴独立运动起跳频率	2000	7D0
	Y 轴直线插补运动起跳频率	2000	7D0
	Y 轴圆弧插补运动起跳频率	2000	7D0
5.1.2.3	Y 轴独立直线运动的加速度 A	20,000	4E20
	Y 轴独立直线运动的匀速脉冲频率	20,000	4E20
	Y 轴独立直线运动的加速度 B	20,000	4E20
5.1.2.4	YZ 直线插补运动的加速度 A	20,000	4E20
	YZ 直线插补运动的匀速脉冲频率	20,000	4E20
	YZ 直线插补运动的加速度 B	20,000	4E20
5.1.2.5	YZ 圆弧插补运动的加速度 A	20,000	4E20
	YZ 圆弧插补运动的匀速脉冲频率	20,000	4E20
	YZ 圆弧插补运动的加速度 B	20,000	4E20
5.1.3.1	Z 轴复位速度	2000	7D0
5.1.3.2	Z 轴独立运动起跳频率	2000	7D0
5.1.3.3	Z 轴独立直线运动的加速度 A	20,000	4E20
	Z 轴独立直线运动的匀速脉冲频率	20,000	4E20
	Z 轴独立直线运动的加速度 B	20,000	4E20

5.1.4.1	X轴手动为电平方式的脉冲速度	2000	7D0
5.1.4.2	Y轴手动为电平方式的脉冲速度	2000	7D0
5.1.4.3	Z轴手动为电平方式的脉冲速度	2000	7D0
5.1.4.4	X轴手动为脉冲方式的点动距离	1000	3E8
5.1.4.5	Y轴手动为脉冲方式的点动距离	1000	3E8
5.1.4.6	Z轴手动为脉冲方式的点动距离	1000	3E8
5.1.4.7	X轴反向复位距离 $S_x$	1000	3E8
5.1.4.8	Y轴反向复位距离 $S_y$	1000	3E8
5.1.4.9	Z轴反向复位距离 $S_z$	1000	3E8
5.2.1	X轴正向软件限位寄存器	16,777,215	FFFFFF
	X轴负向软件限位寄存器	16,777,215	FFFFFF
5.2.2	Y轴正向软件限位寄存器	16,777,215	FFFFFF
	Y轴负向软件限位寄存器	16,777,215	FFFFFF
5.2.3	Z轴正向软件限位寄存器	16,777,215	FFFFFF
	Z轴负向软件限位寄存器	16,777,215	FFFFFF
5.2.4	X轴当前位置寄存器	0	0
5.2.5	Y轴当前位置寄存器	0	0
5.2.6	Z轴当前位置寄存器	0	0

## 2、编程举例

- (1) STX, 00, 19, 3000000, 0, ETX, BCC ; 设置命令序号为 1
- (2) STX, 00, 41, 0000001, 0, ETX, BCC ; X轴原点复位
- (3) STX, 00, 42, 0000002, 0, ETX, BCC ; Y轴原点复位
- (4) STX, 00, 24, 0000000, 0, ETX, BCC ; 复位完成, 设置当前 X 坐标为 0
- (5) STX, 00, 25, 0000000, 0, ETX, BCC ; 复位完成, 设置当前 Y 坐标为 0
- (6) STX, 00, 02, 00007D0, 00007D0, 00007D0, 0, ETX, BCC ; 设置 X 轴起跳频率
- (7) STX, 00, 03, 0004E20, 0004E20, 0004E20, 0, ETX, BCC ; 设置 X 轴加速, 最高频率
- (8) STX, 00, 44, 0002710, 0, ETX, BCC ; X 轴正向运动 10000 个脉冲

可以使用本公司的 EASY\_MOTION 测试软件, 完成对控制器的参数修改, 单步运行, 全速运行, 等等。

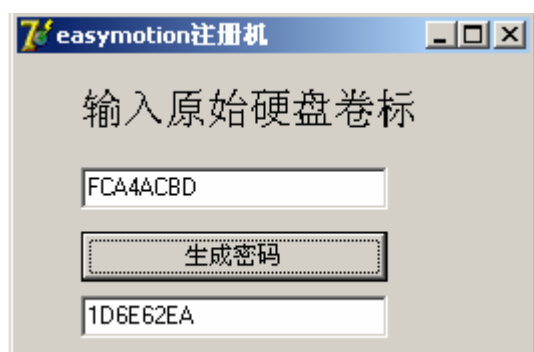
### 3、关于 EasyMotion 软件的安装

双击 setup.exe 安装，next...next 完成安装。

找到快捷键 Lintron→EasyMotion\_zxj→EasyMotion.exe 运行，会出现以下画面：

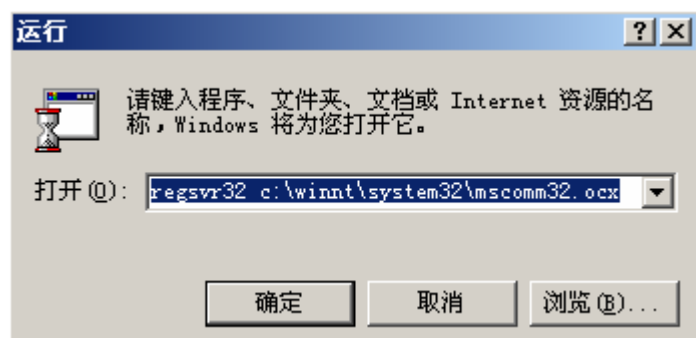


双击安装光盘的“EasyMotion 注册机.exe”软件运行，出现以下画面：



在第一个文本区输入第一个画面下的文字，如“FCA4ACBD”，点击“生成密码”，将生成密码下的字母和数字输入到第一个画面，即可进入控制软件界面。

特殊地，由于操作系统版本很多，如果有些系统报错，请在运行栏输入 **regsvr32 c:\winnt\system32\mscomm32.ocx**，如果你的操作系统不是安装在 c:\winnt，改成你的安装目录，比如 c:\windows。



单击“确定”后会出现以下画面



单击“确定”之后再运行软件。



厦门华菱工控技术有限公司

Xiamen Hualing Industry Control Technology Co.,Ltd

地址：福建省厦门市高科技产业开发区光厦楼北 5 楼

邮编：361006

联系电话：0592-6798422 传真：0592-5660695

技术支持：13799757073（林先生）

E-mail: [linrongfeng77@21.com](mailto:linrongfeng77@21.com) QQ:49579631