



北京和利时电机技术有限公司
BEIJING HOLLYSYS ELECTRIC TECHNOLOGY CO.,LTD.



蜂鸟系列 低压无刷伺服驱动器

操作手册

www.hollysys-electric.com

目 录

第一章 产品概要

1.1.	产品特性	3
1.2.	型号命名	4
1.3.	评估套件	4
1.4.	技术特性	5
1.5.	外形尺寸图	5

第二章 安装和接线

2.1.	使用和储运环境	6
2.2.	安装注意事项	6
2.3.	接线说明	8
2.4.	上位通讯接线	9

第三章 试运行和操作

3.1.	空载试运行	11
3.2.	通电前检查	11
3.3.	通电试运行	11

第四章 集成用户界面

4.1.	伺服参数及操作说明	13
4.2.	伺服控制模式设定	15
4.3.	内置高级运动控制功能	16
4.4.	其他参数配置	18

第五章 RS232 通讯协议

5.1.	概述	20
5.2.	通讯范例	23
5.3.	高级代码编程	24
5.4.	RS485 组网运行	27

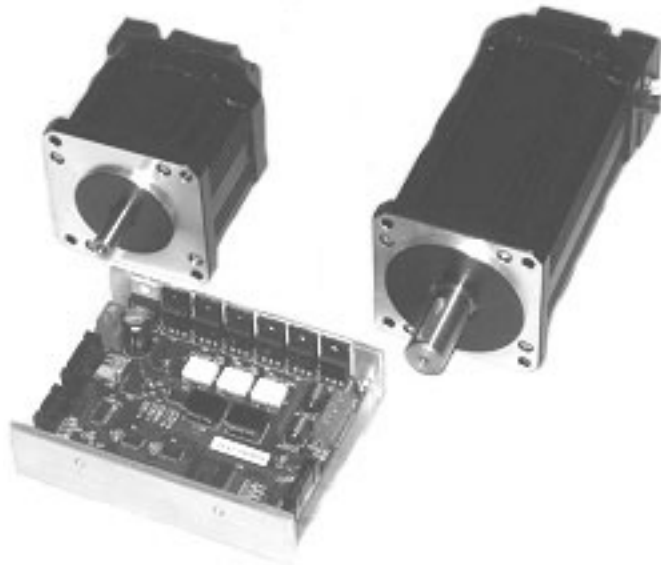
第六章 配套电机

6.1	型号说明	28
6.2	技术数据	29
6.3	外形尺寸图	30

附录1		31
-----	-------	----

附录2		33
-----	-------	----

第一章 产品概要



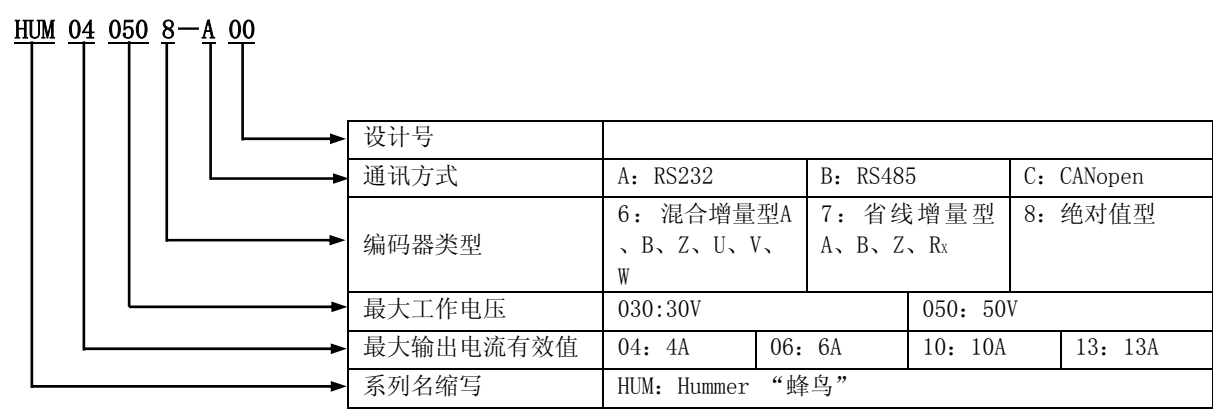
“蜂鸟（Hummer）”系列低压无刷伺服驱动器是北京和利时电机公司最新推出的适合低压直流供电的、小体积、高性能全数字伺服驱动器。硬件上采用32位高速RISC专用控制芯片，高效功率变换技术，以及创新编码器反馈技术；软件上采用最先进的电机控制策略，完全以软件方式实现了电流环、速度环、位置环的闭环伺服控制；驱动器嵌入了高级运动控制功能，通过通讯接口即可完成如多段点到点、直线插补、圆弧插补等功能。配合海豚（DOL）系列高性能的

永磁低压无刷伺服电机，整套伺服系统具有精巧的体积、精美的外观、高工艺设计及优异的性能，在车载系统、医疗设备、航空航天、机器人、电子半导体设备、纺织机械等很多对体积、输入电源、性能要求苛刻的应用领域无疑是客户最佳的选择。

1.1 产品特性

- ◇ 32位高速RISC专用芯片和实时多任务操作系统；
- ◇ 全闭环数字伺服电机控制，位置/速度/转矩 控制模式
- ◇ 多种控制模式：脉冲串（ $\leq 500\text{kpps}$ ）、模拟电压（0~5V）、通讯控制等；
- ◇ 快速响应特性，90度位置调整仅需 8ms；
- ◇ 简单实用的 Dynamic Motion Magus 软件，PC 图形化界面进行参数调整和运行控制；
- ◇ 驱动器内置点到点功能，S 曲线加速，直线/圆弧插补运动，以及电子齿轮功能；
- ◇ 利用 G 代码编程可方便实现数控加工控制；
- ◇ 利用 RS232/485 可以对最多 127 轴进行组网和参数设定；
- ◇ 非接触式、高可靠性、360 度磁性绝对位置传感器，14 位分辨率（16384 计数值/转）；
- ◇ 可从上电位置或者客户指定位置（0~360 度）启动进行位置伺服控制；
- ◇ 过压、过流、过载、位置反馈超差报警等；
- ◇ 单一+24V~+48VDC 供电；
- ◇ 驱动器体积小巧：100 x 75 x 25 mm；
- ◇ 配套高性能、低惯量、永磁 AC 伺服电机；容量 50/100/250 瓦；

1.2 型号命名



1.3 评估套件

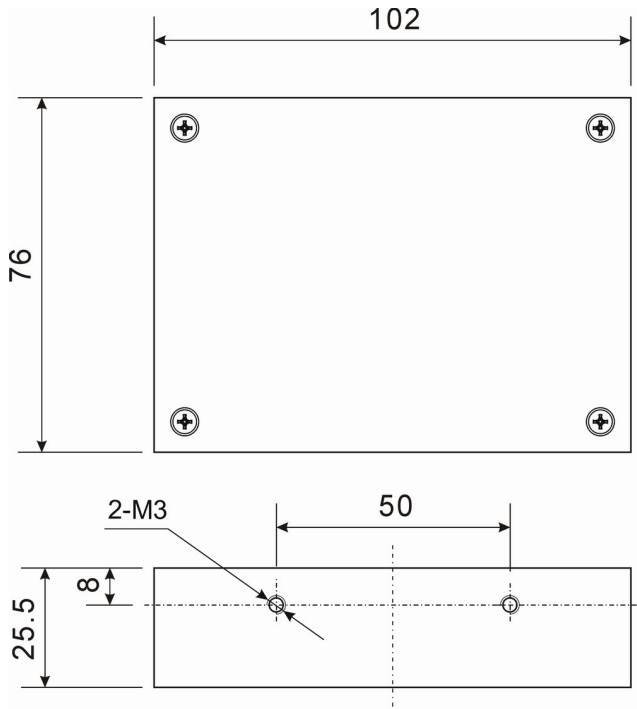
为了方便客户感受低压无刷伺服系统的特性，我们提供特别的“低压伺服入门评估套件”（LV Servo Easy Starter Kit），将完整的伺服系统（驱动器和电机）、调整软件、扩展电缆、操作手册等纳入一个完整的包装，用户可以直接向我们联系订购。该套件的组成见下表：

名称	组 成
低压伺服入门评估套件	* HUM系列伺服驱动器一台； * DOL系列伺服电机一台； * 上位通讯控制软件Dynamic Motion Magus(CD)一张； * 中文操作手册一本；（本书） * 接插件一套，含5、6、7芯各1件； * RS232连接电缆及其驱动电缆：一套； * SPI通讯接口及通讯协议； * 电机延长电缆：2M，一根； * 编码器延长电缆：2M，一根；

1.4 技术特性

驱动器型号	HUM-060508A	HUM-130508A
货物编码	030421	030433
电源电压	24~48VDC	
额定功率（W）	100	250
最大输出电流有效值（A）	6	13
正反转脉冲输入	Max. 500KHZ，+5V逻辑电平	
SPI输入		
模拟输入	0~+5V	
RS232	38400bps，+5V逻辑电平，外部缓冲	
电子齿轮	4096/齿数，500≤齿数≤16384	
反馈传感器	绝对位置传感器，16384脉冲/转	
90° 旋转定位阶跃响应时间	12ms	
位置控制精度	<0.088°	
速度控制精度	<0.1%	
转矩控制精度	<5%	
再生功率吸收	板上电路或者外接电容	
运行/环境温度	- 10~+60°C	
储存温度	- 40~+85°C	
运行/储存湿度	95 RH%	
震动/冲击极限	1.0G（ 9.8m/s² ）	
尺寸（mm）	100x76x25	
结构	基板安装	
重量（ Kg ）	0.175	
适配电机功率（W）	50/100	50/100/250

1.5 外形尺寸图



第二章 安装和接线

2.1 使用和储运环境

	项 目	条 件
伺 服 驱 动 器	安装环境	◆ 没有腐蚀、易燃气体、油雾、金属粉尘侵袭的地方； ◆ 室内，没有雨淋和太阳直射； ◆ 无潮无尘，通风良好；
	运行环境温度	-10℃ ~ 60℃
	运行环境湿度	<95%（没有凝露）
	储运温度	-40℃ ~ 85℃
	储运湿度	<95%（没有凝露）
	储运条件	在堆放和转运时，必须使用产品原包装，以防止受压损坏和跌落。
	振 动	<1G (9.8m/s ²)
伺 服 电 机	安装环境	◆ 没有腐蚀、易燃气体、油雾、金属粉尘侵袭的地方； ◆ 室内，没有雨淋和太阳直射； ◆ 无潮无尘，通风良好； ◆ 没有强烈振动；
	运行环境温度	0℃ ~ 40℃
	运行环境湿度	<95%（没有露滴）
	储运温度	-10℃ ~ 60℃，
	储运湿度	<95%（没有露滴）
	储运条件	在堆放和转运时，必须使用产品原包装，以防止跌落和电机轴碰坏。
	振动	运转状态小于0.5G(4.9m/s ²)，静态时小于2.5G(24.5m/s ²)

2.2 安装注意事项

2.2.1 拆封开箱

- a. 打开包装后，应确认驱动器和电机在运输过程中没有损坏。
- b. 轻轻晃动驱动器，确保其内部没有零件脱落，如：螺丝等。

- c. 检查驱动器和电机铭牌，确保是您所选的机型。
- d. 检查配件和所选附件是否齐全。

2.2.2 驱动器安装

- a. 确保驱动器的安装环境满足2.1项的要求。
- b. 驱动器必须安装在保护良好的电气柜中，垂直放置，结构上应该保证上位机控制器与驱动器之间的距离不大于3米。
- c. 驱动器与机柜和其他部件的安装间距应至少大于10mm，以留出足够通风的空间。
- d. 合理安装风扇，使控制柜内部温度分布均匀。因为驱动器的使用寿命与环境温度有很大关系，所以安装好后请确保驱动器四周10mm处的温度最高不得超过+60℃。
- e. 确保驱动器安装在不可燃的材料如金属上。

2.2.3 电机安装

- a. 确保电机的安装环境满足2.1项的要求。
- b. 电机水平安装时，应让电机的电缆出口朝下，以免油和水进入。电机垂直安装或与减速箱连接时，也应确保油和水不能进入电机。
- c. 确保电机电缆没有受到外部拉力和尽可能小的弯曲力，特别是在电缆出口处或连接处。另外电缆不能浸没在油或水中。
- d. 拆装连接在电机轴上的齿轮或带轮时，严禁敲击电机和电机轴，防止电机轴损伤或者编码器损坏，应采用专用压拔器拆装。
- e. 电机安装必须牢固且具备防止松动措施，同时应确保电机轴不承受超负荷的轴向和径向负载，否则可能损坏电机。
- f. 电机轴和负载的连接最好采用柔性联轴器。

2.2.4 驱动器接线

- a. 电机绕组及电源输入导线请尽可能采用大小合适、带有绝缘套的冷压端子牢固压接后与驱动器接线端子连接。
- b. 当采用多股导线连接时，请优先采用冷压端子压接，其次为将线头充分烫上焊锡后再连接至驱动器接线端子上。禁止将多股导线直接连在驱动器接线端子上，以防止断脱的细股导线掉入驱动器内部而引起故障。
- c. 驱动器控制线的走线应保证距离电源和电机动力线的走线至少300mm，禁止将控制线和电源电机动力线铺设于同一个线槽或捆扎一起。
- d. 如果现场对电源的噪声抑制要求很高，在驱动器电源输入端，请务必安装一个参数合适的噪声滤波器。

2.3 接线说明

2.3.1 电源及电机功率端子接线

输入电源端子为JP1：“+”为直流电源正输入，“-”为直流电源地输入。

注！由于驱动器内部支撑电容容量有限，建议经常需要急加减速的应用场合，在输入直流电源母线上外接支撑电解电容；电容耐压不小于50V，容量在条件允许情况下越大越好。推荐负载在200W以内时选择1000U/63V。

电机接线端子为JP5：A接电机绕组U相，电机线为红色；B接电机绕组V相，电机线为兰色；C接电机W相，电机线为黄色。

下表为电源和电机接线要求：

符号	说 明	配 线 要 求
+	输入直流电源正	◆ 连接导线截面积应不小于2mm ² （功率200W之内）
-	输入直流电源地	◆ 连接导线截面积应不小于2mm ² （功率200W之内）
A	电机绕组U相（红）	◆ 连接导线截面积应不小于2mm ² （功率200W之内） U、V、W端子必须与电机绕组一对一连接！ ◆ 严禁接地、短路，或与电源端子连接。
B	电机绕组V相（兰）	
C	电机绕组W相（黄）	

注意事项

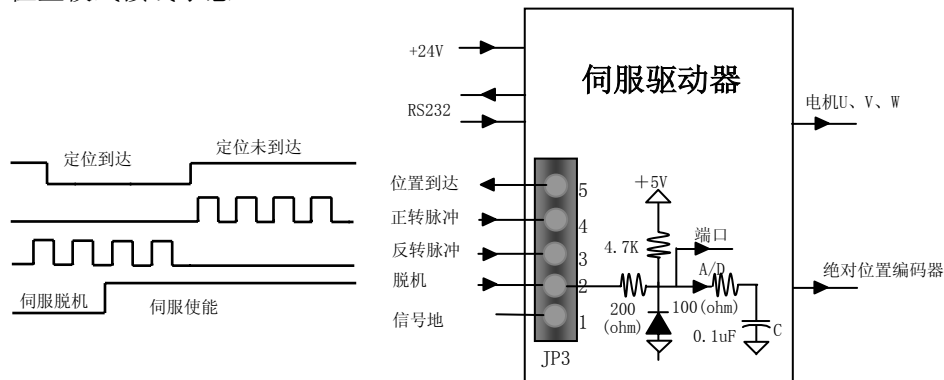
- 接线前请确保是由专业人员操作，且已经详细阅读过电机及驱动器接线说明。
- 所有的接线工作应确保是在驱动器完全断电的情况下进行。
- 请严格按照驱动器铭牌规定的电压和极性接线，否则会造成产品损坏或人身伤害。
- 驱动器的电机接线端子必须与电机绕组定义一对一连接，不可接错！
- 电机及电源的接线端子宜采用根部绝缘的冷轧式端子连接，严禁采用裸露多股导线，以防导线断屑掉入驱动器内引起短路而造成损坏。
- 在电源噪声较强的情况下，务必采用电源噪声滤波器。

2.3.2 控制信号输入接口

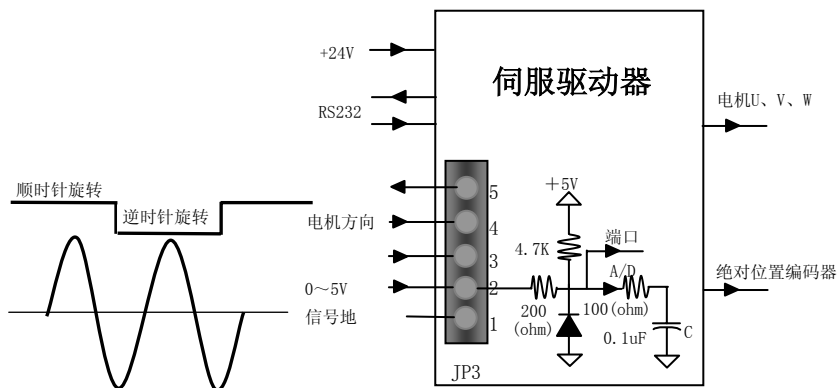
驱动器的控制信号输入接口为JP3，在不同的控制模式下各管脚功能定义如下：

管脚序号	位置控制模式	速度/转矩控制模式	SPI通讯模式
1	信号地	信号地	信号地
2	脱机；0 = 伺服脱机。	模拟量输入，0~5V	
3	反转脉冲输入，上升沿有效。		串行数据输入
4	正转脉冲输入，上升沿有效。	电机转向控制，1= 顺时针旋转	串行时钟输入
5	定位到达；1 = 定位已到达。		串行数据输出

位置模式接线示意：



速度/转矩模式接线示意：



2.3.3 编码器反馈连接

HUM系列伺服驱动器采用了专利技术的绝对位置编码器反馈技术，连接插头为JP4，管脚分别定义为：

管脚	定义
1脚	信号地
2脚	编码器输出的PWM格式的电机转子绝对位置信号
3脚	驱动器输出给编码器的串行时钟信号
4脚	驱动器输出给编码器的串行数据信号
5脚	编码器输出给驱动器的串行数据信号

2.4 上位通讯连线

本系列伺服驱动器支持两种通讯模式：RS232或SPI，实现上位控制系统（通用计算机或其它微控制器）对伺服驱动器的参数设置和运行控制；

2.4.1 用于RS232C通信

通过本公司提供的专用通讯电缆及上位通讯软件，将驱动器的JP2端口与PC机的串口相连，

可以方便的实现驱动器各种参数的设定及运行控制。JP2插头管脚定义为：

管脚	定义
1脚	信号地
5脚	RS232信号输入脚（RXD），TTL/CMOS电平兼容
6脚	RS232信号输出脚（TXD），TTL/CMOS电平兼容
7脚	驱动器提供的+5V电源
2、3、4脚	空脚，可悬空

注！ 由于本系列驱动器的RS232通讯口没有采用标准的RS232电平，因此若要与通用计算机的串口相连，必须使用本公司提供的电平转换专用电缆。

2.4.2 用于SPI串行外围接口通讯

此项功能为用户可选功能。当上位控制器需要使用SPI通讯模式时，我公司可以提供相应的软件通讯协议，以实现上位控制器对驱动器工作参数的设定及运行控制。该方式与位置、速度、转矩模式共用JP3插头。管脚定义为：

管脚	定义
1脚	信号地
2脚	位置模式下为脱机信号输入，速度/转矩模式下为模拟量输入
3脚	位置模式下为反转脉冲输入，SPI模式下为串行数据输入
4脚	位置模式下为正转脉冲输入，SPI模式下为串行时钟输入
5脚	位置模式下为到位信号输出，SPI模式下为串行数据输出

2.4.3 集成软件开发功能

在本公司提供的上位控制软件包Dynamic Motion Magus（DMM）里，用户不仅能够实时修改伺服驱动器的运行参数、调整伺服系统运行特性；同时，本软件包还提供了详细的串行通讯协议指令说明，用户通过串行指令即可轻易实现单台伺服驱动器的点到点、速度控制，或多至127台伺服驱动器进行RS232/RS485组网运行，实现3轴直线插补或2轴圆弧插补；另外，本软件包还提供了G代码开发功能，在Dynamic Motion Magus环境下，利用本软件包提供的32条指令，用户即可开发编写实际应用程序，控制一台或多台伺服驱动器实现位置、速度、直线插补、圆弧插补、找原点等数控加工程序。如果用户需要在其它软件环境下使用这些指令，则需要本公司提供的动态连接库函数的支持。目前该库函数正在开发完善中，用户如有需求可与我公司协商。详细内容参见第四章“DMM集成软件”使用说明。

第三章 试运行和操作

3.1 空载试运行

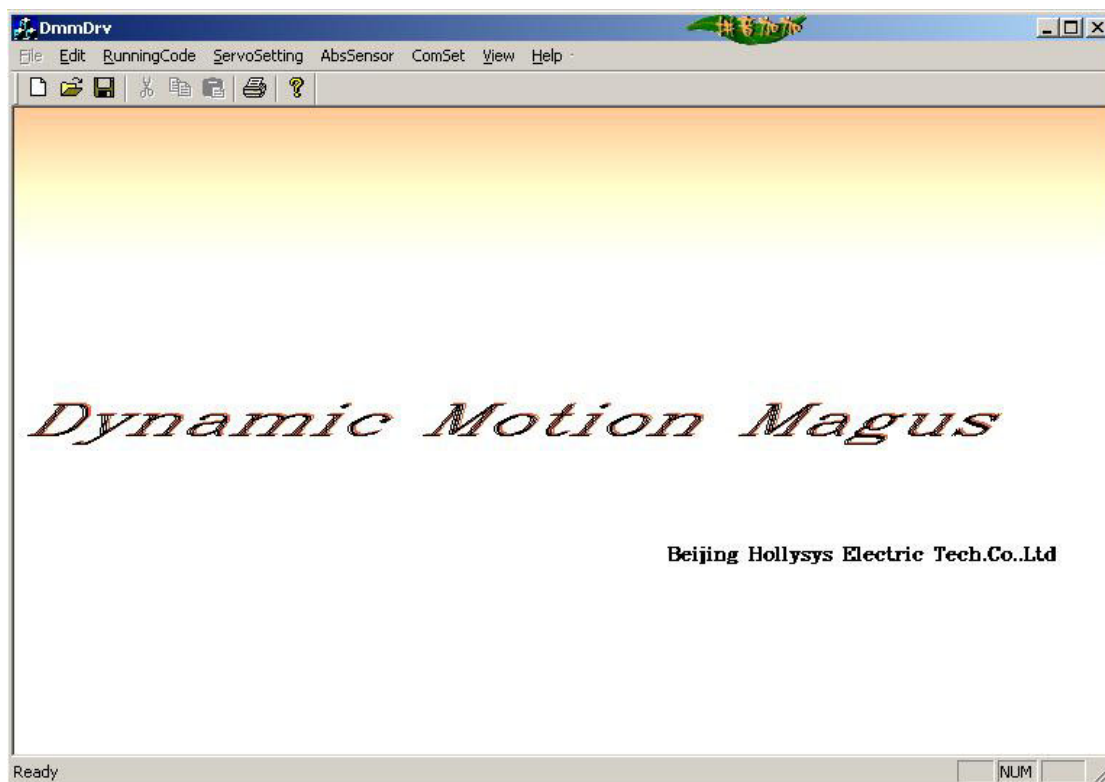
在正式使用本系列伺服驱动器前，需通过本公司提供的Dynamic Motion Magus上位通讯软件选择伺服驱动器的运行模式、命令输入模式、以及设定驱动器的各个工作参数，如主增益、速度增益等等。

3.2 通电前检查

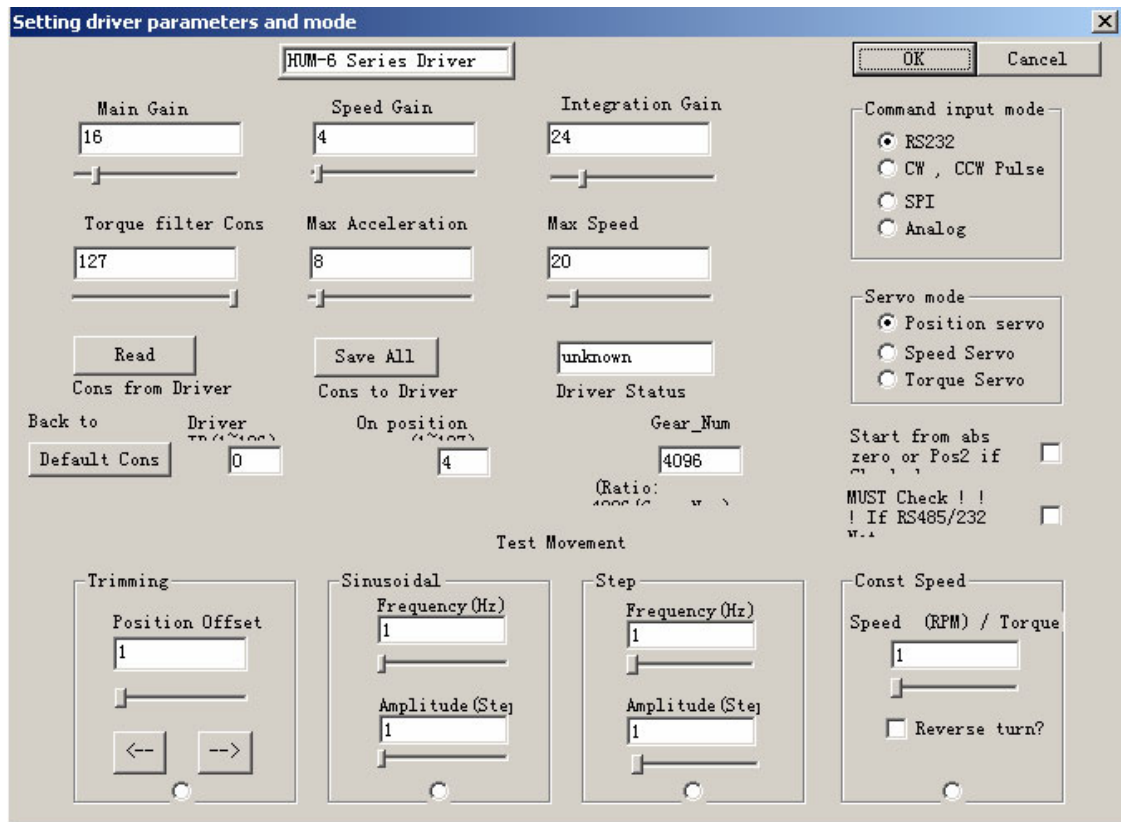
- 1) 确保所有接插件和连线连接正确，电源线和电机接线没有短路或接地。
- 2) 确保电源输入电压在规定范围内。
- 3) 确保接口电源的电压大小和极性正确。
- 4) 确保伺服电机已经固定好，电机轴没有连接负载。
- 5) 如果装有机械制动器，请确认制动器处于释放状态。

3.3 通电试运行

确认伺服驱动器已与PC机串口正确连接。运行 **DmmDrv** 应用程序



打开ServoSetting 下拉菜单，选中，进入伺服驱动器参数设定及试运行界面。



本软件提供了四种测试运转模式：

- 1) **正/反转点动 (Trimming) 功能：** 用户输入每次点动的脉冲数 (Position Offset)，点击下方的单选按钮“使能”点动功能，以后每点击左箭头一次电机即反转指定的脉冲数；同理，每点击右箭头一次电机即正转指定的脉冲数。
- 2) **S曲线往复运动 (Sinusoidal)：** 输入S曲线往复运行的幅值（即脉冲数）、频率，点击下方的单选按钮，电机即从静止以S曲线加速，维持一段时间恒速，再以S曲线减速至停止，同时保证电机位移量即为用户输入的幅值；频率即为每秒完成S曲线往复运行的次数。频率设定值越高意味着S曲线的加速度越大，而所有这一切都是由伺服驱动器内部自动完成。
- 3) **单步运动 (Step)：** 输入单步运转的脉冲数及往复频率，点击下方的单选按钮即可“使能”本项功能。
- 4) **恒速运转 (Const Speed)：** 输入电机运行速度，点击下方的单选按钮即可“使能”本项功能；同时还可通过方向选择按钮更改电机旋转方向。

通过以上四种试运行模式即可测试驱动器电机接线是否正确，功能是否正常。

第四章 集成用户界面

本系列伺服驱动系统提供了一个功能强大，用户界面友好的集成通讯软件包 **Dynamic Motion Magus**，可安装在WINDOWS98/2000/NT/XP等操作系统下；利用本公司提供的专用电缆将伺服驱动器与PC机的RS232口连接，运行DmmDrv. exe应用程序即可。

无论驱动器处于何种命令模式如位置模式、或速度模式、或转矩模式等，都可以通过RS232口来进行伺服系统配置及参数调整；用户也可以让电机进行诸如点到点、步进、正弦波往复等试运行。

运行DmmDrv 应用程序以后，打开ServoSetting 下拉菜单选择HUM-6，即可弹出伺服设定对话框。点击SAVE按钮即可将伺服驱动器的系统配置及运行参数存入到伺服驱动器的EEPROM里；用户也可以通过SPI（串行外围接口）总线实现同样功能。

注！：除非需要进行伺服系统重新配置或修改参数，否则千万不要通过SPI总线以固定频率频繁对伺服驱动器进行系统配置或参数更新，频繁对伺服驱动器进行写操作将大大缩短EEPROM寿命。EEPROM的寿命大约为100万次写操作；Max Acceleration(最大加速度)和Max Speed(最大速度)这两个参数不是存储在伺服驱动器的EEPROM里，因此对这两个参数的修改次数没有限制。

4.1 伺服参数及操作说明

4.1.1 读取伺服驱动器参数

本系列伺服驱动器在出厂时已经配置好缺省参数，缺省参数能保证在无惯量或小惯量负载下系统运行在最佳状态。

伺服驱动器缺省配置

位置模式	MainGain = 16	MaxSpd = 20
RS232 输入模式	SpeedGain = 4	MaxAcl = 8
相对位置模式	IntGain = 24	OnRange = 4
Driver_ID = 0	TrqCons = 127	Gear_Num = 4096
Config = 0x00		

强力推荐在第一次使用本伺服系统时先读出驱动器里的缺省参数，修改完后再保存回去。

警告！：如果多台伺服驱动器通过RS485/RS232串行网络连接在了一起，那么请选中RS485/RS232单选框，否则参数将会写入到网络上的所有伺服驱动器里。

如果只有一台伺服驱动器连接到RS485/RS232网络上并且RS485/RS232单选框未选中，那么就可以设定伺服驱动器的ID号，或者也可以读出伺服驱动器的全部参数包括以前设定过的ID号。在每台伺服驱动器都配置了唯一的ID号后，即可通过RS485/RS232网络将他们进行组网。

4.1.2 伺服参数设定

随着负载的不同，伺服运行性能也会有所变化，如单步响应出现过冲等。如果负载过重的话，甚至会出现伺服运行不稳定，因此，需要针对不同负载情况进行参数调整。本系列伺服驱动器提供了如下参数供用户调节以适应不同负载的需要：

主增益 (MainGain)，设定范围1~127:

此参数为伺服环主增益，通常随着负载加重需相应增加。值越大意味着伺服环的响应频带越宽。

速度增益 (SpeedGain)，设定范围1~127:

此参数为伺服环速度增益，通常随着负载加重需相应增加。值越大意味着伺服环的响应频带越窄。实际应用中，较重的负载通常对动态特性要求不是很苛刻，因此应该增加速度增益值，减小伺服环频宽。

重载时如果速度增益值偏高，转矩指令平滑度将下降，由此会引起电机运行噪音加大。此时可以通过此时可通过适当减小转矩常数（参见转矩常数）来消除噪音。

转矩滤波常数 (Trqcons)，设定范围1~127:

转矩滤波常数是一个一阶滤波常数，值越大表示滤波器的频带越宽；滤波器的表达式如下：

$a / (S + a)$ ，这里 $a = 26 \times \text{Trqcons}$ ；如果 $\text{Trqcons} = 100$ ，则 $a = 2600$ 。转矩滤波器的目的在于平滑转矩指令，特别是在负载较重且速度环增益设定值又比较大的情况下，应适当减小转矩常数的设定值；如果负载很轻，并且需要伺服系统具备较快的响应性，则应该把转矩常数的值设定得尽可能大甚至为127。

积分增益 (Int Gain)，设定范围1~127:

积分增益在伺服环里的作用是用来消除位置指令与电机实际位置的静态误差，同时能抑制伺服系统的转矩脉动。综合增益的设定值越大，抑制转矩脉动的能力就越强。

4.1.3 伺服驱动器ID

每台伺服驱动器应有自己唯一的ID号，如果只有一台伺服驱动器连接到RS232口上且RS485NET框未选中，用户就可以在ServoSetting对话框中读出或设置驱动器的ID号。每台伺服驱动器出厂默认ID值都是0。ID号用于多台伺服驱动器通过RS485组网时的驱动器标识。ID = 0 ~ 126。

当多台伺服驱动器连接到RS485/232网且RS485NET框被选中，在ServoSetting对话框里进行的参数读写操作只对相应ID号的伺服驱动器有效，其它伺服驱动器将不会响应所有操作。

4.1.4 定位范围 (OnRange)

定位范围值是用来判定电机位置是否已到达给定位置。此设定值可依据用户对定位精度的要求设定。如果用 Pset 代表位置指令，用 Pmotor 代表电机转子实际位置，则当 $|Pset - Pmotor| \leq \text{OnRange}$ 时，伺服驱动器就认为电机已经达到指令位置。OnRange的设定范围为1~127。OnRange值代表的实际定位精度为 $\text{OnRange} * 360 (\text{deg}) / 16384$ 。

4.1.5 从绝对零点启动或从当前位置启动

由于本系列伺服驱动器采用了360度范围的绝对位置编码器作为电机转子位置反馈，因此既可以选择伺服电机以增量的方式从转子当前位置开始运动，也可以以绝对位置方式运动；用户甚至可以任意指定电机转子的绝对位置作为零点并保存到编码器的存储单元（参见编码器说明）。

本系列伺服驱动器的缺省运行模式为增量式。

如果绝对零点单选框被选中，电机将自动运转到缺省的绝对零点或用户设定的绝对零点。电机自动找绝对零点的运动轨迹遵循路径最短的原则，通常运行角度小于180度。

如何让电机运动到绝对零点？

- a. 给伺服驱动器通入+24V 电源。
- b. 启动DmmDrv应用程序。
- c. 读出驱动器全部参数。
- d. 不要让电机驱动做任何试运行。
- e. 选中**从绝对位置启动**单选框。
- f. 点击SAVE（保存）按钮，电机将自动运行到绝对零点。

4.2 伺服控制模式设定

4.2.1 输入命令模式选择

有四种输入命令模式可以选择，RS232，正/反转脉冲，SPI，模拟量。即使在正/反转脉冲、SPI或模拟量输入模式下，仍然可以通过RS232口来设定系统配置及参数调整。

SPI接口为用户定制选件。如果用户需要，我公司可提供通讯协议。

在正/反转脉冲输入模式下，如果 $|Pset - Pmotor| \leq 0nRange$ ，5脚输出为低电平，表示定位到达。如果2脚输入为低电平，电机处于自由脱机状态。如果2脚为高电平或悬空，则电机伺服使能。对正/反转脉冲输入，信号的上升沿有效，电机旋转的正方向定义为从电机前端盖看过去的顺时针方向。

位置模式下，5V信号输入对应电机在顺时针方向旋转 $90 \times 4096 / Gear_Num$ (度)。

模拟量输入信号也可用于速度/转矩控制，甚至可以在JP3的2、1脚接入一只电位器进行位置、速度、转矩控制。

4.2.2 伺服运行模式

本系列伺服驱动器有三种运行模式：位置、速度和转矩模式。

位置伺服模式

位置模式下伺服驱动器响应上位控制器的位置命令。

速度伺服模式

速度模式下伺服驱动器响应上位控制器的速度命令。MainGain 和 SpeedGain 用来调节速度环的运行性能。输入模式可以是 RS232，或模拟量甚至外接电位器。

转矩伺服模式

转矩伺服模式下，根据输入转矩指令，伺服电机输出恒转矩。输入模式可以是 RS232，或模拟量甚至外接电位器。

4.2.3 电机试运行

如果选择了 RS232 命令模式，可以方便的实现伺服电机的多种试运转模式：围绕原点的小范围运动，点到点运动、正弦波往复运动，恒速运转等。

4.3 内置高级运动控制功能

本系列伺服驱动器具备3种内置的高级功能，通过 RS232 口可以很容易的控制伺服驱动器实现点到点、直线插补、圆弧插补等运动控制功能。

4.3.1 点到点运动

在RS232命令模式下，用户可以在DmmDrv 组态软件环境下编辑指令代码，控制电机实现点到点定位运行。由于点到点的距离有可能会比较长，因此伺服驱动器内部的S曲线加速功能自动启用。对S曲线运动来说，最大加速度和最大速度是两个非常重要的参数。

以下是驱动器内部的最大速度和最大加速度计算公式：

$$i = (\text{MaxSpd}+3) * (\text{MaxSpd}+3) / 16 \text{ 的整数部分}$$

$$\text{最大速度} = i * 3333.3 * \text{GearRatio}, \text{ 单位为PPS (脉冲/秒)}$$

或可以表示为：

$$\text{最大速度} = i * 12.21 * \text{GearRatio}, \text{ 单位为RPM (转/分)}$$

$$\text{这里 } \text{GearRatio} = 4096 / \text{Gear_Num}.$$

由于MaxSpd 的设定范围为 1 到 127，因此如果设定 GearRatio = 1，也即16384脉冲/转，则对应电机的最大速度为 12.21(rpm) 到 12893(rpm) 。

$$\text{最大加速度} = \text{MaxAcl} * 1388888.9 * \text{GearRatio}, \text{ 单位为脉冲/秒平方}$$

如果 MaxAcl = 10 且 GearRatio = 1，电机将在 40ms 内加速到34转/秒。

因此，对于点到点运动，电机将以设定的最大加速度加速到最大速度，然后保持一段恒速，再减速到零。整个运行过程非常平滑。

如果用 Amax 表示 最大加速度，单位：脉冲/秒平方(ppss)

用 Vmax 表示 最大速度，单位：脉冲/秒(pps)

用 S 表示 电机行程，单位：脉冲(p)

则电机走完全部行程需要的时间计算公式如下：

$$T_{\text{total}} = V/A_{\text{max}} + S/V \text{ (Second)}$$

这里 V = 取 Vmax 和 (sqrt(Amax*S)) 的最小值。

例如：

GearRatio = 1 或 Gear_Num = 4096，即16384脉冲/转。

MaxSpd = 68, MaxAcl = 12

Amax = 16666666(ppss)

Vmax = 1050000(pps), 或 3845(RPM)

S = 884736(p), 或 54 转

sqrt(Amax*S) = 3839999.9 > Vmax

因此 V = Vmax = 1050000(pps)

Ttotal = V/Amax + S/V = 1050000/16666666 + 884736/1050000

= 0.9056(秒) or 905.6(毫秒).

上位控制器(PC 或 微控制器或 DSP)使用绝对或相对运动模式发出位置指令，输入范围为-134217728 到 134217727。伺服驱动器接收到指令后立即自动开始S曲线运动，同时把驱动器内部状态寄存器的BIT5置1，表示驱动器正在做S曲线运动。当S曲线运动完毕后，状态寄存器的BIT5自动清

零；通过读取伺服驱动器状态寄存器可以知道S曲线运动是否已完毕。如果当前S曲线运动还没有结束，伺服驱动器又接收到新的指令，当前的运动将立即截止；同时电机从当前位置开始下一次S曲线运动。在这种情况下，如果位置命令使用绝对位置模式，那么电机最终将停在给定的命令位置；如果位置命令使用相对位置模式，电机最终停止的位置将不可知。

4.3.2 X、Y、Z 3 轴直线插补运动

X 轴驱动器 ID 必须设定为 0.

Y 轴驱动器 ID 必须设定为 1.

Z 轴驱动器 ID 必须设定为 2.

3 轴直线插补运动只能在 ID 号为0、1、2的伺服驱动器间执行；假设电机当前位置为 (X0, Y0, Z0)，目标位置为 (X1, Y1, Z1)。

第一数据包内容：X1 - X0

第二数据包内容：Y1 - Y0

第三数据包内容：Z1 - Z0

第四数据包内容：进给速度 (FeedRate)，设定值范围 1~127

如果Gear_Number 设定为4096，电机速度为 $1.526 * \text{FeedRate} (\text{Rpm})$ 。

四个数据包缺一不可，即使 $Z1 - Z0 = 0$ ，也要给出数据0。

伺服驱动器 0, 1, 2 接收到上述3组数据后，将从 (X0, Y0, Z0) 插补运动到 (X1, Y1, Z1)。

在插补运动期间，驱动器内部状态寄存器的 BIT5 = 1，运动完成后 bit5 = 0。

接收到一条相对或绝对位置命令后，伺服驱动器自动结束当前的插补运动。

4.3.3 X Y Z 任意2轴圆弧插补运动

X 轴驱动器 ID 必须设定为 0.

Y 轴驱动器 ID 必须设定为 1.

Z 轴驱动器 ID 必须设定为 2.

3 轴直线插补运动只能在 ID 号为0、1、2的伺服驱动器间执行；使用圆弧命令发送包含3组数据的5个数据包，使电机从当前位置 (X0, Y0) 运行到目标位置 (X1, Y1)，圆心为 (Xc, Yc)。

第一数据包内容：X0 - Xc

第二数据包内容：Y0 - Yc

第三数据包内容：X1 - Xc

第四数据包内容：Y1 - Yc

第五数据包内容：包含两字节：PlaneNumber | FeedRate

PlaneNumber = 0, 1, 2. 0 代表 X-Y平面，1 代表 Z-X 平面。2 代表 Y-Z平面。FeedRate = -127~127，没有 0。如果 FeedRate 为正，电机执行顺时针圆弧运动；如果 FeedRate 为负，电机执行逆时针圆弧运动；数据顺序 (PlaneNumber<<8) | FeedRate，高字节为 Plane number，低字节为 FeedRate. 圆或圆弧运动可以在 X-Y, Z-X, 或 Y-Z 平面运动，取决于 PlaneNumber的设定。

在插补运动期间，驱动器内部状态寄存器的 BIT5 = 1，运动完成后 bit5 = 0。

接收到一条相对或绝对位置命令后，伺服驱动器自动结束当前的插补运动。

假定 $R0 = \sqrt{(X0-Xc)*(X0-Xc) + (Y0-Yc)*(Y0-Yc)}$

$R1 = \sqrt{(X1-Xc)*(X1-Xc) + (Y1-Yc)*(Y1-Yc)}$

请确保 $|R0-R1| \leq 1$ ，否则驱动器可能会运行不正确。

请确保 $\sqrt{(X0-X1)*(X0-X1) + (Y0-Y1)*(Y0-Y1)} > 4$ ，也即本函数不能完成一个360度圆周运动；为了实现完整圆周运动，必须调用本函数两次。 $-134217728 \leq X0, Y0, X1, Y1, Xc, Yc \leq 134217727$ ，或可表示为半径 < 134217727 。

如果两台驱动器的Gear_number设定值不一样而又具备同样的丝杠螺距，那么就会加工出椭圆。

注：对通讯函数、数据包的详细说明请参见第五章RS232通讯协议说明。

4.4 其它参数配置

4.4.1 电子齿轮（Gear_Number）

Gear_Number 设定范围：500 to 16384，出厂设置为 4096。

Gear_Number 提供了可选的电子齿轮比：4096 / Gear_Num，范围 0.25 ~ 8.192。

例如，如果设定 Gear_Num = 4096，则每 16384 个输入脉冲对应电机旋转1周。如果设定

Gear_Number = 500，则 2000 输入脉冲对应电机旋转1周。

对位置伺服模式下的模拟量输入，输入信号范围 0 ~ 5(V)，对应电机旋转角度为：

$0 \sim 90 * 4096 / \text{Gear_Num}$ (度)。

电子齿轮比对于RS232模式同样有效。另，电子齿轮比只对位置模式有效。

4.4.2 驱动器配置寄存器和状态寄存器

本系列伺服驱动器的系统配置，如 RS232 指令模式或正/反转脉冲模式等，存储在驱动器内部EEPROM内的Config寄存器里；而驱动器当前状态如报警信息等则存储在驱动器内部EEPROM内的Status寄存器里。

Status 寄存器为1字节数据，低7位用于指示驱动器的状态，如伺服使能、报警、定位到达、脱机状态等。

Status = x b6 b5 b4 b3 b2 b1 b0

b0 = 0 : 定位到达，或 $|Pset - Pmotor| \leq \text{OnRange}$

b0 = 1 : 电机忙，或 $|Pset - Pmotor| > \text{OnRange}$

b1 = 0 : 电机处于伺服使能状态

b1 = 1 : 电机处于脱机状态

b4 b3 b2 = 0 : 无报警

b4 b3 b2 = 1 : 电机位置超差报警，或 $|Pset - Pmotor| > 8192$ 脉冲，180度

= 2 : 电机过流报警

= 3 : 电机过热、过载报警

= 4~7: 功能保留

b5 = 0 : 等候S曲线运动、直线插补、圆弧插补命令输入。

b5 = 1 : 电机忙于S曲线运动、直线插补或圆弧插补

b6 : JP3 第2脚状态指示

Config = x b6 b5 b4 b3 b2 b1 b0

b1 b0 = 0 : RS232 模式

1 : 正/反转脉冲模式

2 : SPI 模式

3 : 模拟量输入模式

b2 = 0 : 相对位置模式

b2 = 1 : 绝对位置模式，上电复位后电机将自动回绝对零点

b4 b3 = 0 : 位置伺服模式

1 : 速度伺服模式

2 : 转矩伺服模式

3 : 保留

b5 = 0 : 使能伺服

b5 = 1 : 关伺服使能

b6 : 保留

第五章 RS232通讯协议

RS232口在伺服驱动器上电以后自动处于激活状态。如果选择了RS232命令输入模式，那么不仅可以通过RS232口读写伺服驱动器的参数和状态，还能用RS232口向伺服驱动器发送点到点的位置命令；

如果命令输入模式为正/反转脉冲、SPI或模拟量输入模式，那么RS232口就只能用来读写伺服参数和状态。

使用DmmDrv 应用程序来实现RS232通讯控制功能是最方便的，当然，用户也可以根据我公司提供的以下协议实现其他微控制器或DSP对本系列伺服驱动器的控制。此时，本系列伺服驱动器为从机，其它控制器为主机。

以下为详细的通讯协议说明，多台伺服驱动器可以组成串行网络。

5.1 概述

5.1.1 通讯数据帧说明

通讯数据帧	说明
字节	包含8位，从高到低依次为 b7b6b5b4b3b2b1b0 或表示为 b[7:0]。 b7 为最高位，b0 为最低位
数据包P	由多字节组成，表示为 $P = B_n B_{n-1} B_{n-2} \dots B_1 B_0$ ，字节长度 $n+1$ ， B_n 为起始字节， B_0 为结束字节；字节长度随命令不同而不同，数据帧定义如下： $P = ID + \text{帧长度} + \text{功能码} + \text{数据} + \text{校验和}$
ID	1字节长度
帧长 + 功能码	共用1字节长度
数据	长度为1到3字节
校验和	1字节长度

注！ 最小数据包长度为 4 字节，数据为 1 字节；最大数据包长度为 7 字节，数据为 4 字节。

数据字节最少为 1，对有些功能码，数据字节是无意义的，可以任意赋值而不会影响功能码的正确执行。

5.1.2 数据字节格式

每帧数据的第一个字节的格式为：0xxxxxxx，最高位总是零；其它字节的格式为：1xxxxxxx，最高位总为1，x 表示可以为 0 或 1。因此通过判断数据字节的最高位是否为零即可确认一帧数据的开始。

5.1.3 帧数据字节详细说明

1、起始字节 B_n

起始字节的最高位总是0，低7位代表伺服驱动器的ID号，范围为0~126；ID号127被用来作为广播地址。驱动器的ID号可以通过DmmDrv 用户界面设定。

2、 B_{n-1} 字节

该字节用来表示本帧数据的字节长度及功能代码。

$B_{n-1} = 1 \ b_6 \ b_5 \ b_4 \ b_3 \ b_2 \ b_1 \ b_0$

$b_6 \ b_5$	代表数据帧的长度:
0 0	数据帧长度为4字节
0 1	数据帧长度为5字节
1 0	数据帧长度为6字节
1 1	数据帧长度为7字节

$b_4 \sim b_0$ 代表功能码:

以下数据为主控制器发送，伺服驱动器接收

功能描述	$b[4:0]$	后接数据字节	； 注释
设绝对零点	0x00	1(无意义)	； 设转子当前位置为绝对零点
走绝对位置	0x01	$1 \sim 4$	； 将电机轴移动至相对于零点的某一位置
走直线插补	0x02	$1 \sim 4$	； 电机执行直线插补运动
走相对位置	0x03	$1 \sim 4$	； 以电机轴当前位置为基准，移动一段距离
走圆弧插补	0x04	$1 \sim 4$	； 电机执行圆弧插补运动
设置驱动器ID	0x05	1	； 设置驱动器ID号
读取驱动器ID	0x06	1(无意义)	； 读取驱动器ID
设置配置寄存器	0x07	1	； 设置配置寄存器
读取配置寄存器	0x08	1(无意义)	； 读取配置寄存器
读取状态寄存器	0x09	1(无意义)	； 读取状态寄存器
恒速运动	0x0a	$1 \sim 3$	； 电机执行恒速运动
方波运动	0x0b	$1 \sim 3$	； 电机执行方波运动
正弦波运动	0x0c	$1 \sim 3$	； 电机执行正弦波运动
正弦波往复频率	0x0d	$1 \sim 3$	； 电机正弦波往复频率
通用读取命令	0x0e	$1 \sim 4$	
		后接数据0x1b	； 读电机绝对位置功能
保留	0x0f		；
设置主增益	0x10	1	； 设置主增益
设置速度增益	0x11	1	； 设置速度增益
设置积分增益	0x12	1	； 设置积分增益
设置转矩滤波常数	0x13	1	； 设置转矩滤波常数
设置最高速度	0x14	1	； 设置最高速度 $MaxSpd, 1 \sim 127$
设置最大加速度	0x15	1	； 设置最大加速度 $MaxAcc, 1 \sim 127$
设置定位范围	0x16	1	； 设置 $OnRange, 1 \sim 127$
设置电子齿轮	0x17	2	； 设置电子齿轮, $500 \sim 16384$

读主增益	0x18	1(无意义)	;
读速度增益	0x19	1(无意义)	;
读积分增益	0x1a	1(无意义)	;
读转矩滤波常数	0x1b	1(无意义)	;
读取最高速度	0x1c	1(无意义)	;
读取最大加速度	0x1d	1(无意义)	;
读取定位范围	0x1e	1(无意义)	;
读取电子齿轮	0x1f	1(无意义)	;

以下数据为伺服驱动器发送，主控制器接收

保留	0x00	~ 0x0a	;
是主增益	0x10	1	;
是速度增益	0x11	1	;
是积分增益	0x12	1	;
是转矩滤波常数	0x13	1	;
是最高速度	0x14	1	;
是最大加速度	0x15	1	;
是驱动器ID	0x16	1	;
是定位范围	0x17	1	;
是电子齿轮	0x18	2	;
是状态寄存器	0x19	1	;
是配置寄存器	0x1a	1	;
是转子位置	0x1b	1~4	;
功能保留	0x1c	~0x1f	;

由于功能码的不同，每帧数据的长度也不同；

3、Bn-2 ~ B1 字节

Bn-2~B1 (n>2) 为数据包中的数据. Bn-2 为最高字节，B1 为最低字节，每个数据字节的最高位恒为1，只有低7位有效.

如果数据只有一字节，其最高位恒为 1，而次高位为符号位，只有低6位代表数据大小，因此表示范围为-64 ~ 63。如果数据有两字节，则最高字节的最高位为1，次高位为符号位，低6位为有效数据；低字节的最高位恒为1，低7位代表了有效数据，与高字节的6位数据位组合起来，构成了13位的数据+1位符号位，因此表示范围为-8192~8191；其它依此类推。

注！本协议的数据表示方法与常规略有不同！！

数据帧长	数据范围	;
4	-64 ~ 63	; 数据为 B1
5	-8192 ~ 8191	; 数据为 B2 B1
6	-1048576 ~ 1048575	; 数据为 B3 B2 B1
7	-134217728 ~ 134217727	; 数据为 B4 B3 B2 B1

4、B0 字节

B0 字节为校验和，计算方法如下：

$$S = B_n + B_{n-1} + B_{n-2} + \dots + B_1$$

$$B0 = 0x80 + \text{Mod}(S, 128), B0 = 0x80 + S - 128 * [S/128],$$

$$B0 = 128 \sim 255$$

伺服驱动器接收到主控制器的数据包以后，计算校验和；如果校验和与B0相同，说明数据通讯正确，否则数据传输有错。

5.1.4 RS232 通讯格式

波特率： 38400

停止位： 1

奇/偶校验：无

5.1.5 RS232 信号电平及中间电平转换电缆

通过本系列伺服驱动器的 JP2 口输出的RxD 和 TxD 信号为TTL/CMOS 电平, 因此，为了把PC机的串口与 JP2 相连，需要本公司提供的具备电平转换功能的转接电缆。

5.2 通讯范例

例1：设 ID = 6 的电机当前位置为原点

数据包帧长为 4，包含1字节无意义数据。

$$B3 = 0x00 + ID = 0x06;$$

$$B2 = 0x80 + (\text{帧长} - 4) * 32 + \text{设当前位置为原点功能码} = 0x80;$$

$$B1 = 0x80; \text{有效数据部分无意义，可以全部填0}$$

$$S = B3 + B2 + B1 = 0x06; \text{只取低8位有效，高位舍弃}$$

$$B0 = 0x80 + \text{Mod}(S, 128) = 0x86;$$

例2：命令ID = 3 的电机以相对位置模式从当前转子位置开始顺时针旋转120步

功能码为 0x03;

120 = 0x78 = 0x000 0000 0111 1000 > 63, 需2字节数据：低7位 111 1000=0x78无符号位；

高7位为000 0000 = 0x00含符号位；

数据帧长为5；

$$B4 = 0x03;$$

$$B3 = 0x80 + (\text{数据帧长}-4) * 32 + 0x03 = 0x80 + 0x20 + 0x03 = 0xa3.$$

$$B2 = 0x80 + 0x00 = 0x80;$$

$$B1 = 0x80 + 0x78 = 0xf8;$$

$$S = B4 + B3 + B2 + B1 = 0x03 + 0xa3 + 0x80 + 0xf8 = 0x21e$$

$$B0 = 0x80 + \text{Mod}(S, 128) = 0x9e$$

例3：命令ID = 3 的电机以相对位置模式从当前转子位置开始逆时针旋转120步

功能码为0x03;

$-120 = 0xff88 = 0x1111\ 1111\ 1000\ 1000 < -64$, 需2字节数据: 低7位 $000\ 1000 = 0x08$, 高7位 $111\ 1111 = 0x7f$;

数据帧长5;

$B4 = 0x03$;

$B3 = 0x80 + (\text{数据帧长}-4)*32 + \text{功能码} = 0x80 + 0x20 + 0x03 = 0xa3$.

$B2 = 0x80 + 0x7f = 0xff$

$B1 = 0x80 + 0x08 = 0x88$

$S = B4 + B3 + B2 + B1 = 0x03 + 0xa3 + 0xff + 0x88 = 0x22d$

$B0 = 0x80 + \text{Mod}(S, 128) = 0x80 + 0x2d = 0xad$

例4：命令ID = 2 的电机以 60转/分的恒速顺时针运转.

速度值为 60, 1字节数据, $60 = 0x3c$; 数据帧长4;

$B3 = 0x02$;

$B2 = 0x80 + (\text{数据帧长}-4)*32 + \text{功能码} = 0x80 + 0x0a = 0x8a$;

$B1 = 0x80 + 0x3c = 0xbc$;

$S = B3 + B2 + B1 = 0x02 + 0x8a + 0xbc = 0x48$;

$B0 = 0x80 + \text{Mod}(S, 128) = 0xc8$

例5：命令ID = 2 的电机以 60转/分的恒速逆时针运转.

功能码为 $0x0a$;

$-60 = 0xc4 = 0x1100\ 0100 > -63$, 1 字节数据, 低7位为 $100\ 0100 = 0x44$;

数据帧长4

$B3 = 0x02$;

$B2 = 0x80 + (\text{数据帧长}-4)*32 + \text{功能码} = 0x80 + 0x0a = 0x8a$;

$B1 = 0x80 + 0x44 = 0xc4$;

$S = B3 + B2 + B1 = 0x02 + 0x8a + 0xc4 = 0x150$;

$B0 = 0x80 + \text{Mod}(S, 128) = 0x80 = 0x50 = 0xd0$;

例6：在X-Y-Z平面实现直线插补运动

假设X、Y、Z三轴电机的当前位置为 $(X0, Y0, Z0) = (0, 0, 0)$, 直线插补运动结束点的位置为 $(X1, Y1, Z1) = (100, 200, 100)$

X、Y、Z三轴使用公共ID = $0x7f$; 进给速率 = 3, 可设定范围 $1 \sim 127$; 功能码 $0x02$

发送的四帧数据依次为:

Send_Package (ID, $X1 - X0$), 也即 Send_Package ($0x7f, 100$)

Send_Package (ID, $Y1 - Y0$), 也即 Send_Package ($0x7f, 200$)

Send_Package (ID, $Z1 - Z0$), 也即 Send_Package ($0x7f, 100$)

Send_Package (ID, 进给速率), 也即 Send_Package ($0x7f, 3$)

在X、Y、Z三轴电机接收到所有四帧数据后，开始直线插补运动，最后在相同时刻到达终点。

在直线/圆弧插补过程中，上位机可以通过查询伺服驱动器状态寄存器的第5位 b5 来判断插补运动是否已经结束。

Send_Package(ID, Y1 - Y0)为自动生成一帧数据的函数，附录1为其详细C++源代码说明。

例7：在X-Y平面实现圆弧插补运动

假设当前电机位置为 $(X0, Y0) = (0, 0)$ ，圆弧终点位置为 $(X1, Y1) = (200, 0)$ ；顺时针方向画圆，进给速率 = 1；圆心位置为 $(Xc, Yc) = (100, 0)$ 。

X、Y、Z三轴使用公共ID = 0x7f；进给速率 = 1，可设定范围 1~127；功能码 0x04

首先必须发送以下四帧数据：

Send_Package(ID, X0 - Xc), 即 Send_Package(0x7f, -100)

Send_Package(ID, Y0 - Yc), 即 Send_Package(0x7f, 0)

Send_Package(ID, X1 - Xc), 即 Send_Package(0x7f, 100)

Send_Package(ID, Y1 - Yc), 即 Send_Package(0x7f, 0)

第五帧数据格式为：

Send_Package(ID, (PlaneNumber<<8) | FeedRate))

由于进给速率 = 1, planeNumber = 0, 因此

Send_Package(0x7f, 0x01) = Send_Package(0x7f, 1);

如果进给速率 = -1, planeNumber = 0, 则

Send_Package(0x7f, 0xff) = Send_Package(0x7f, 255);

如果进给速率 = 1, planeNumber = 1, 则

Send_Package(0x7f, 0x101) = Send_Package(0x7f, 257);

如果进给速率 = -1, planeNumber = 1, 则

Send_Package(0x7f, 0x1ff) = Send_Package(0x7f, 511);

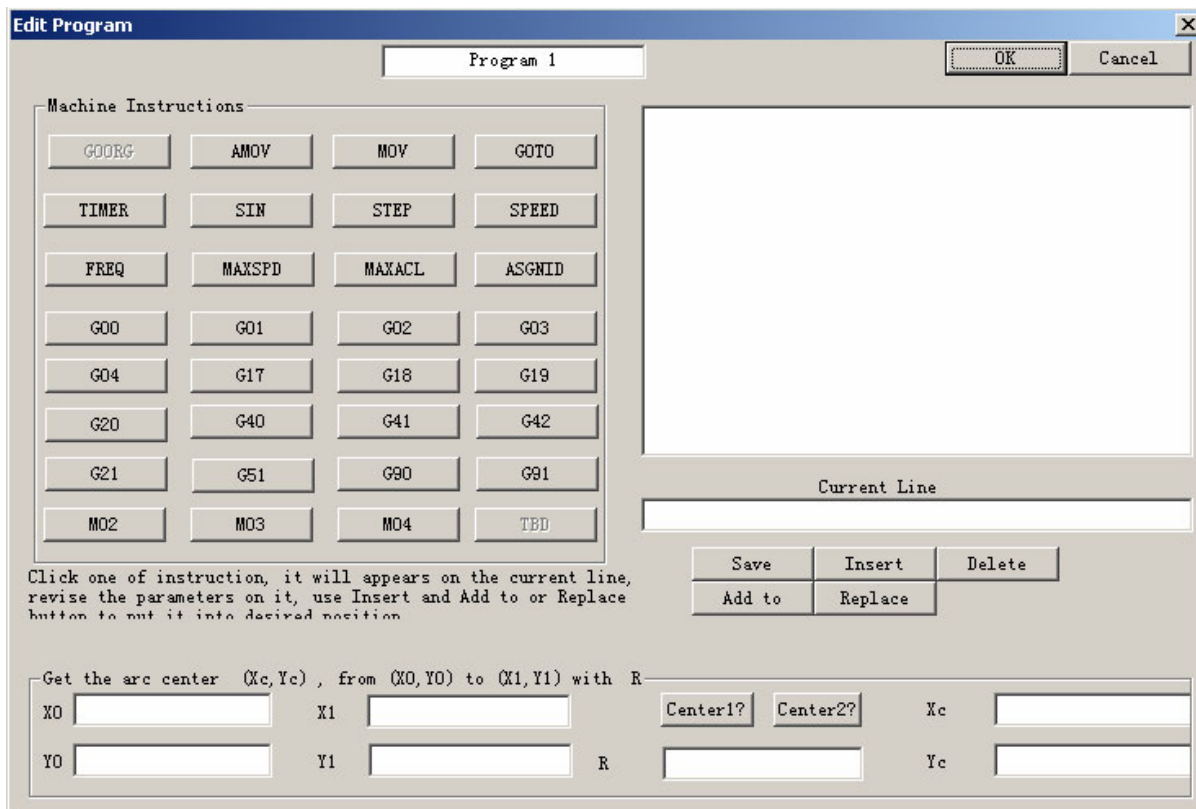
当X-Y-Z三轴驱动器接收到全部五帧数据以后，指定平面的两轴电机完成圆弧插补运动后同时到达圆弧终点位置。在圆弧插补运动期间，上位机可以通过查询驱动器状态寄存器来检查插补运动是否已经完成。

注：由于圆弧的起点和终点坐标不允许完全相同，因此画完整圆的运动必须分解为执行2次。

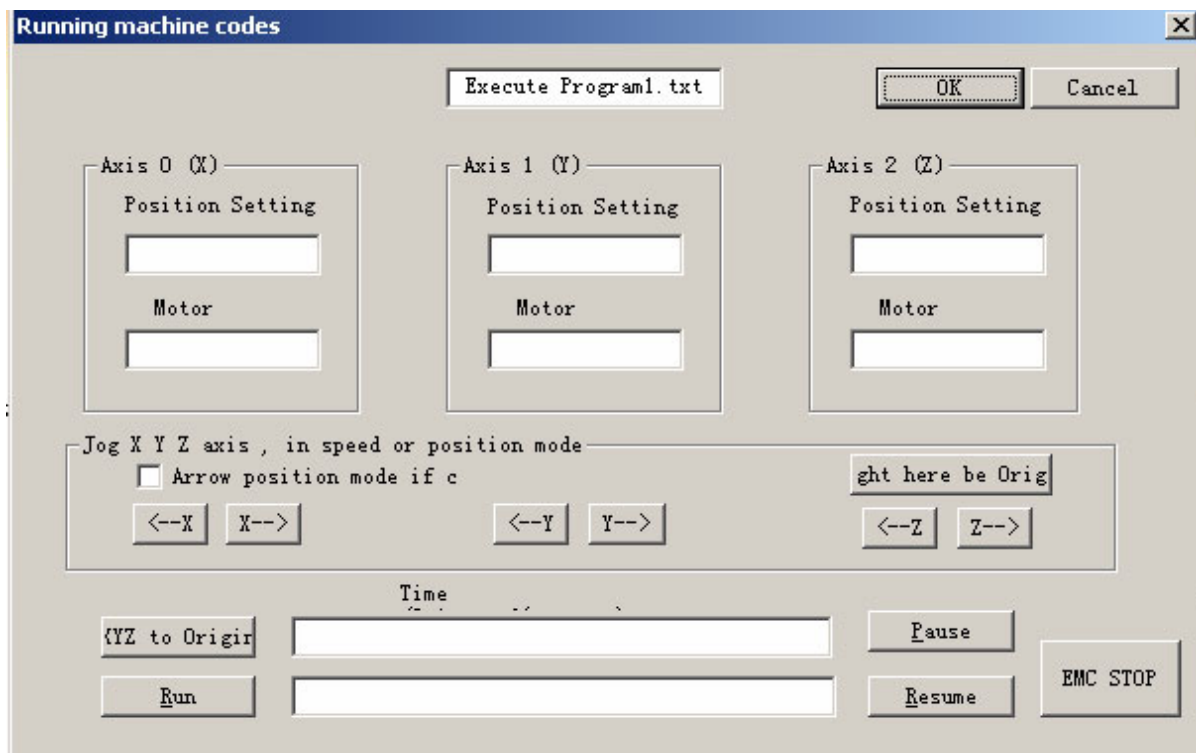
5.3 高级代码编程

在RS232命令输入模式下，通过 DmmDrv 应用程序的编辑和运行代码对话框，用户可以编辑并运行高级代码指令程序。我们提供了32条指令，其中包括20条类BASIC指令和20条G代码指令。为了方便用户，我们随机提供了5个例程，程序名字分别为Program1~Program5.txt；安装DmmDrv.exe 以及program1~5.txt，把所有文件拷贝到同一个目录。

打开编辑菜单，通过几个简单的按钮，用户可以编辑、修改、保存这些应用程序。当然其它文本编辑器也可以用来编辑用户程序，如微软wordpad或notepad。



打开运行代码菜单，用户可以运行编辑完毕的程序实现诸如点到点、圆弧、直线插补等功能。



以下是函数详细功能介绍：

注：所有输入的命令必须用大写字母。

xxx: AMOV , axis , Absolute position

功能：运动到指定的绝对位置；

axis : 伺服驱动器 ID 号
 Absolute position : 绝对目标位置, 范围 $0 \sim +134217728$
 行号 xxx 范围为 0 to 999, 也即最多可编辑999条指令。

xxx: MOV , axis , move distance

功能: 从当前位置运行到指定相对位置

axis : 伺服驱动器 ID 号
 move distance : 相对目标位置, $0 \sim +134217728$

注: 如果目标位置为正数, 电机顺时针旋转; 如果为负数, 电机逆时针旋转。

xxx: GOTO , Line number

功能: 跳转到指定行

Line number : 当前文件范围内的任意行号

xxx: TIMER , number

功能: 延时指定的时间

number : $1 \sim 999$, 延时时间 = number * 0.01(s)

xxx: SIN , axis , Amplitude

功能: 电机做正弦波运动

axis : 伺服驱动器 ID 号
 Amplitude : 正弦波幅值?

xxx: SQUARE , axis , Amplitude

功能: 电机单步运动

axis : 伺服驱动器 ID 号
 Amplitude : 单步运动幅值

xxx: FREQ , axis , Frequency

功能: 设置正弦波及单步往复运动的频率

axis : 伺服驱动器 ID 号
 Frequency : 频率值, $1 \sim 30$ (Hz)

xxx: SPEED , axis , Speed Value

功能: 控制电机恒速运转

axis : 伺服驱动器 ID 号
 Speed Value : 速度值, $-6000 \sim 6000$ (rpm)

xxx: MAXSPD , axis , MaxSpd value

功能: 设置 S 曲线运动的最大速度 MaxSpd

axis : 伺服驱动器 ID 号
 MaxSpd Value : MaxSpd , $1 \sim 127$, 参见4.9.1

xxx: MAXACL , axis , MaxAcl value

功能: 设置 S 曲线运动最大加速度 MaxAcl

axis : 伺服驱动器 ID 号
 MaxAcl Value : MaxAcl , $1 \sim 127$, 参见4.9.1

xxx: ASGNID , 127 , new ID

功能: 当RS232网络上只有一台伺服驱动器时, 给驱动器赋新的 ID 号。

new ID : 0 ~ 126

附录2为G代码的详细说明。

5.4 RS485 组网运行

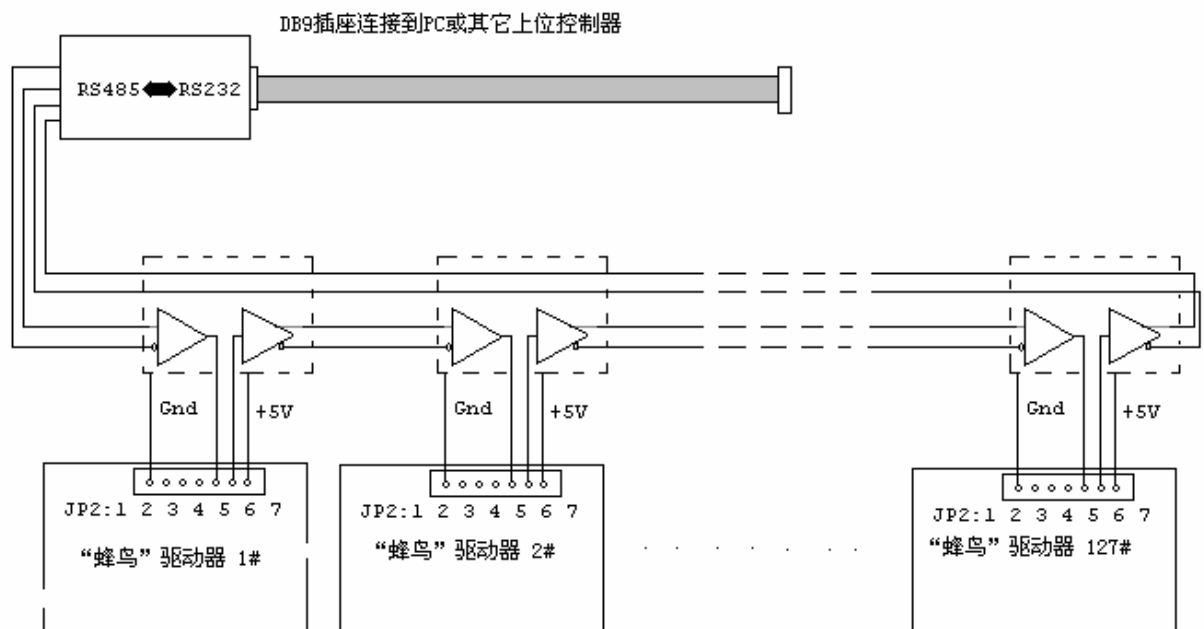
多台伺服驱动器在各自分配了唯一的ID号以后, 就可以通过RS485通讯网络连接在一起, 从而实现 RS485 联网运行; 联网运行时必须选中RS485单选框, 这样, 根据ID号就可以对指定伺服驱动器进行系统配置或参数调节。

在RS485联网状态下不能给伺服驱动器分配ID号; 要想给伺服驱动器分配ID号, 必须保证网络上只联了一台驱动器且RS485单选框未选中。

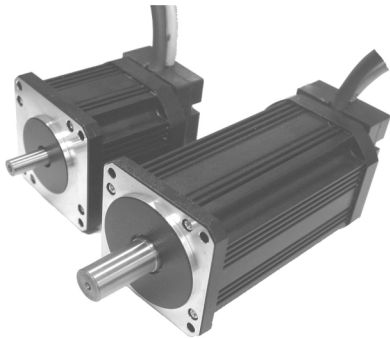
RS485网是一个串行网络, 上位机发来一个数据包后, 第一台驱动器先接收数据然后检查ID号是否与自己吻合。如果吻合, 就留下数据; 否则就把数据传递给下一台驱动器, 以此类推。

由于ID号包含在数据包中的第一个字节里, 一台伺服驱动器接收到第一个字节后立即进行ID校验决定是继续接收数据还是往下传输, 因此整个网络数据传输效率可以快速高效。

关于RS485组网的详细内容请与本公司联系; 下图为组网示意图:



第六章 配套电机



海豚（Dolphin）系列无刷伺服电机是专门为蜂鸟系列驱动器配套设计的低压伺服电机，并装置了绝对位置反馈编码器。该系列电机共有五款，50W/48V、100W/48V、250W/48V和50W/24V、100W/24V，采用DC24~48V供电，外径为57MM。电机设计采用了正弦波反电势，高性能钕铁硼磁钢，以及小惯量转子，使得这个系列的电机具有良好的动态性能和很高的位置分辨率。

6.1 型号说明

57	DOL	05	30	L	8	L	S	
								轴键形式
								K: 键槽
								F: 扁平轴
								S: 光轴
								G: 减速器适配
								电机出线形式
								L: 引线
								B: 螺纹连接器
								C: 拔插连接器
								传感器类型
								1: 开关霍尔
								2: 线性霍尔
								3: 光学编码器
								6: 混合增量型
								7: 省线增量型
								8: 绝对值型
								电机电压
								L: 48VDC
								H: 300VDC
								N: 24VDC
								P: 12VDC
								M: 36VDC
								I: 150DC
								电机额定转速
								1: 100rpm
								30: 3000
								50: 5000
								电机容量
								05: 50W
								25: 250W
								电机系列
								Dolphin : 海豚
								机座号

6.2 外形尺寸图

图1：57DOL0530L8LS & 57DOL0530N8LS

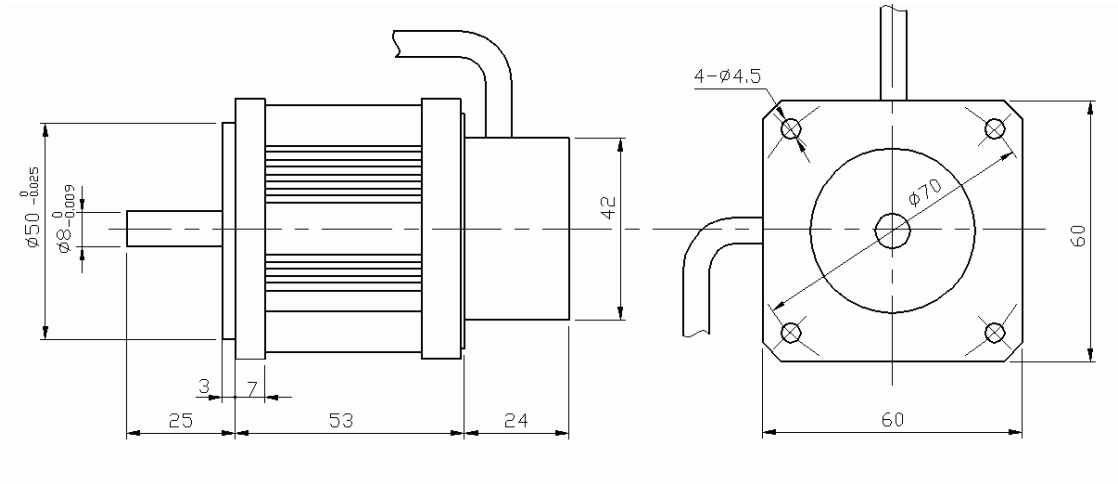


图2：57DOL1030L8LS & 57DOL1030N8LS

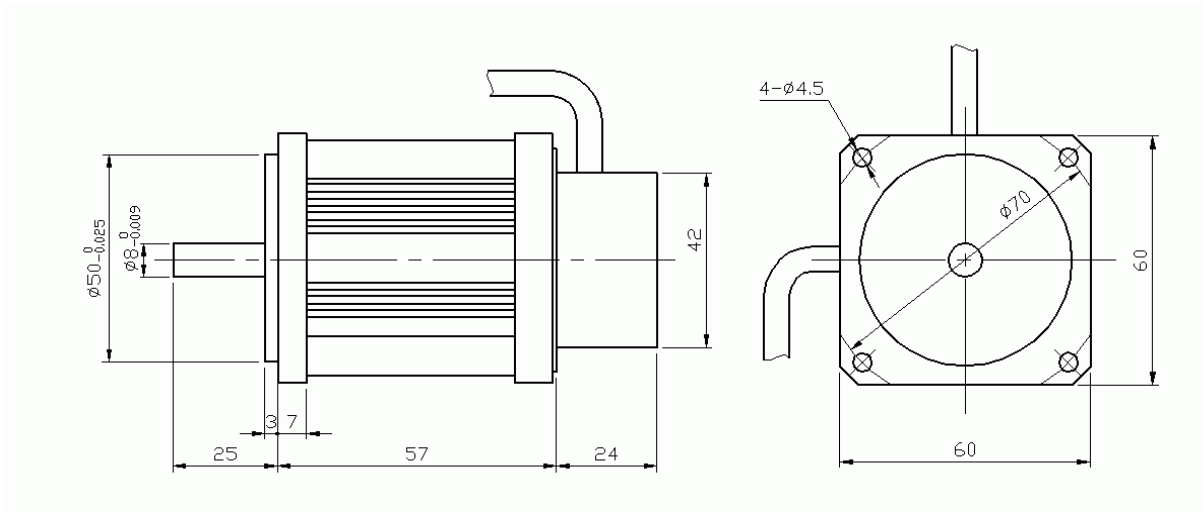
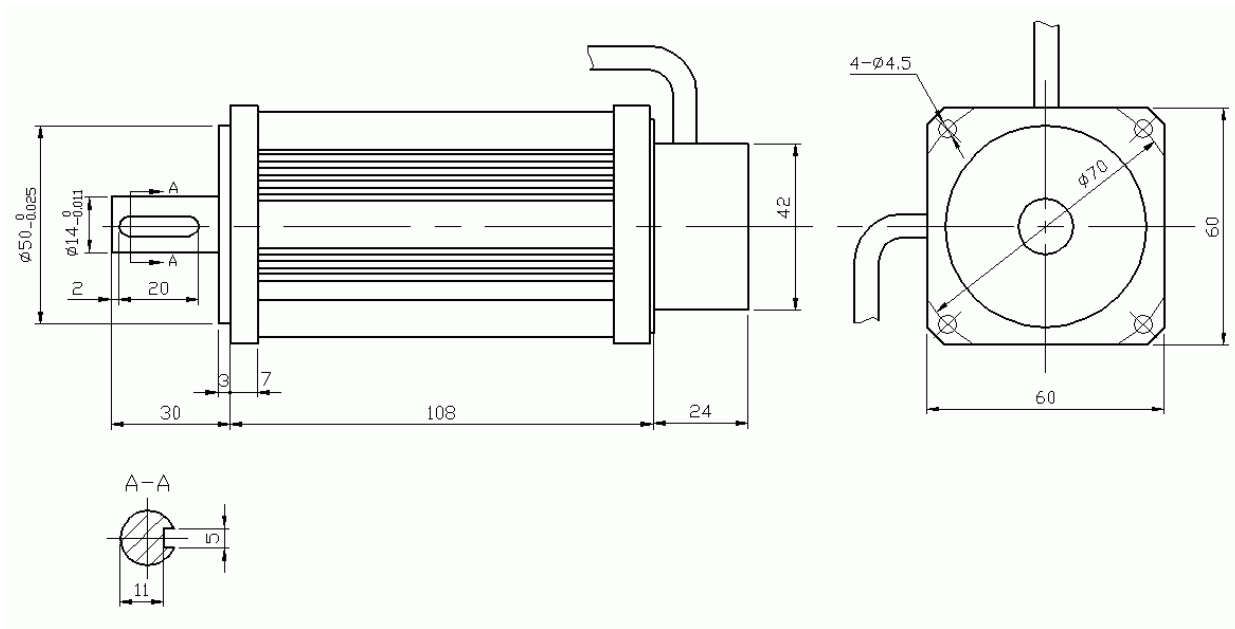


图3：57DOL2530L8LK



6.3 技术数据

技术参数	单位	伺服电机型号		
		57D0L0530L8LS	57D0L1030L8LS	57D0L2530L8LK
货物编码		033059	033057	033060
额定输出功率	W	50	100	250
额定电源电压	V	48V (DC)	48V (DC)	48V (DC)
额定转矩	Nm	0.159	0.318	0.796
瞬间最大转矩	Nm	0.477	0.955	2.388
额定转速	rpm	3000	3000	3000
最高转速	rpm	5000	3600	3600
电机转子惯量	kg·cm ²	0.0411	0.167	0.438
转矩系数	Nm/A	0.0392	0.0597	0.115
额定相电流	A	3.53	2.62	5.78
额定相电压	V	12.8	15.1	28.6
瞬间最大相电流	A	10.59	7.86	17.34
电枢绕组相电阻	Ω	0.432	0.178	0.277
电枢绕组相电感	mH	1.94	0.742	0.782
机械时间常数	ms	1.16	2.50	0.919
电气时间常数	ms	4.50	4.16	2.82
重量	kg	0.64	0.75	1.49
适配驱动器		HUM-060508A	HUM-060508A	HUM-130508A
速度、位置传感器		绝对值编码器 (16384脉冲/转)		
工作制		连续		
绝缘等级		B级		
绝缘耐压		AC1500V, 1分钟		
绝缘电阻		DC500V, 10M以上		
结构		全封闭, 自冷式		
振动		2.5G以下		
海拔		1000m以下		
使用温度、湿度		温度: 0~40℃; 湿度: 90%RH以下 (不结霜条件)		
保存温度、湿度		温度: -20~60℃; 湿度: 90%RH以下 (不结霜条件)		
安装方式		法兰安装		

技术参数	单位	伺服电机型号	
		57DOL0530N8LS	57DOL1030N8LS
货物编码		033058	033056
额定输出功率	W	50	100
额定电源电压	V	24V (DC)	24V (DC)
额定转矩	Nm	0.159	0.318
瞬间最大转矩	Nm	0.477	0.955
额定转速	rpm	3000	3000
最高转速	rpm	5000	3600
电机转子惯量	kg. cm ²	0.0211	0.167
转矩系数	Nm/A	0.0203	0.0404
额定相电流	A	6.47	7.02
额定相电压	V	7.01	11.98
瞬间最大相电流	A	19.4	21.1
电枢绕组相电阻	Ω	0.156	0.387
电枢绕组相电感	mH	0.601	0.305
机械时间常数	ms	0.87	2.90
电气时间常数	ms	3.85	0.788
重量	kg	0.64	0.75
适配驱动器		HUM-130508A	HUM-130508A
速度、位置传感器		绝对值编码器（16384脉冲/转）	
工作制		连续	
绝缘等级		B级	
绝缘耐压		AC1500V, 1分钟	
绝缘电阻		DC500V, 10M以上	
结构		全封闭, 自冷式	
振动		2.5G以下	
海拔		1000m以下	
使用温度、湿度		温度：0~40℃；湿度：90%RH以下（不结霜条件）	
保存温度、湿度		温度：-20~60℃；湿度：90%RH以下（不结霜条件）	
安装方式		法兰安装	

附录1：生成串行控制命令数据帧的 C++ 源代码

注：在前面描述RS232通讯协议的章节中，B0为数据帧中的最后1个字节；但在以下代码中，B0为代码运行后生成的首字节。

```
// Send a package with a function by Global_Func
// Displacement:  $-2^{27} \sim 2^{27} - 1$ 
void DlgRun::Send_Package(char ID , long Displacement)
{
    unsigned char B[8], Package_Length, Function_Code;
    long TempLong;
    B[1] = B[2] = B[3] = B[4] = B[5] = (unsigned char)0x80;
    B[0] = ID&0x7f;
    Function_Code = Global_Func & 0x1f;

    TempLong = Displacement & 0xffffffff;           //Max 28bits
    B[5] += (unsigned char)TempLong&0x0000007f;
    TempLong = TempLong>>7;
    B[4] += (unsigned char)TempLong&0x0000007f;
    TempLong = TempLong>>7;
    B[3] += (unsigned char)TempLong&0x0000007f;
    TempLong = TempLong>>7;
    B[2] += (unsigned char)TempLong&0x0000007f;
    Package_Length = 7;

    TempLong = Displacement;
    TempLong = TempLong >> 20;
    if(( TempLong == 0x00000000) || ( TempLong == 0xffffffff))
    {
        //Three byte data
        B[2] = B[3];
        B[3] = B[4];
        B[4] = B[5];
        Package_Length = 6;
    }

    TempLong = Displacement;
    TempLong = TempLong >> 13;
    if(( TempLong == 0x00000000) || ( TempLong == 0xffffffff))
    {
        //Two byte data
        B[2] = B[3];
        B[3] = B[4];
        Package_Length = 5;
    }

    TempLong = Displacement;
    TempLong = TempLong >> 6;
    if(( TempLong == 0x00000000) || ( TempLong == 0xffffffff))
    {
        //One byte data
        B[2] = B[3];
        Package_Length = 4;
    }

    B[1] += (Package_Length-4)*32 + Function_Code;

    //long x = Cal_SignValue(B);verify the packet,driver use this to get position command
    Make_CRC_Send(Package_Length, B);
}
```

```

void DlgRun::Make_CRC_Send(unsigned char Plength,unsigned char B[8])
{
    unsigned char Error_Check = 0;
    for(int i=0;i<Plength-1;i++)
    {
        OutputBuffer[OutBfTopPointer] = B[i];
        OutBfTopPointer++;
        Error_Check += B[i];
    }
    Error_Check = Error_Check|0x80;
    B[Plength-1] = Error_Check;          //Last byte of packet
    OutputBuffer[OutBfTopPointer] = Error_Check;
    OutBfTopPointer++;

    WritePackage();                      //Write to RS232 buffer
}

/*Get position with sign
*/
long DlgRun::Cal_SignValue(unsigned char One_Package[8])
{
    char Package_Length,OneChar,i;
    long Lcmd;
    OneChar = One_Package[1];
    OneChar = OneChar>>5;
    OneChar = OneChar&0x03;
    Package_Length = 4 + OneChar;

    OneChar = One_Package[2];/*First byte 0x7f, bit 6 reprents sign      */
    OneChar = OneChar << 1;
    Lcmd = (long)OneChar; /*Sign extended to 32bits
    */
    Lcmd = Lcmd >> 1;
    for(i=3;i<Package_Length-1;i++)
    {
        OneChar = One_Package[i];
        OneChar &= 0x7f;
        Lcmd = Lcmd<<7;
        Lcmd += OneChar;
    }
    return(Lcmd);                      /*Lcmd :  $-2^{27} \sim 2^{27} - 1$ 
    */
}

void main(void)
{
    /*Move motor 2 to the position of 321456*/
    char Axis_Num = 2;
    Global_Func = (char)Go_Absolute_Pos;
    long pos = 321456;
    Send_Package(Axis_Num,Pos);
}

```

附录2: DmmDrv 支持的 G 代码详细说明

G00 : 以设定的最大速度和最大加速度按相对位置模式快速移动指定位移量;
G01 : 直线插补运动;
G02 : 顺时针圆弧插补运动;
G03 : 逆时针圆弧插补运动;
G04 : 暂停;
G17 : 选择XY平面运动;
G18 : 选择ZX平面运动;
G19 : 选择YZ平面运动;
G20 : 选择英制输入, 单位英寸;
G21 : 选择米制输入, 单位毫米;
G40 : 取消切削半径补偿;
G41 : 切削半径左侧补偿;
G42 : 切削半径右侧补偿;
G51 : 输入X Y Z 三轴丝杠副的螺距;
G90 : 绝对位置模式选择;
G91 : 增量位置模式选择;
M02 : 结束程序;
M03 : 主轴顺时针旋转;
M04 : 主轴逆时针旋转;

Px : X 轴丝杠副螺距, 单位毫米或英寸;

Py : Y 轴丝杠副螺距, 单位毫米或英寸;

Pz : Z 轴丝杠副螺距, 单位毫米或英寸;

Pavg: X Y Z 三轴丝杠副平均螺距, 详细说明参见 **G51** 部分;

FeedRate 允许范围: $(1.526 \sim 193.78) * P_{avg}$ 单位: 毫米/分, 或英寸/分;

X Y Z 最大移动范围: $(-8192 \sim 8191.999939) * P_{avg}$, 单位: 毫米或英寸;

一行程序最多允许字长: 80 字符。

G00 X* Y*** Z*****

G00 可以让 X,Y,Z 电机按照 MaxSpd 和 MaxAcc 设定的最大速度和最大加速度移动到指定位置。

直到所有轴电机都达到指定位置后, G00 才算执行完毕; G00 不能确保三轴电机同时到达目标位置。

例如: 假定当前三轴电机位置为 $(X_0, Y_0, Z_0) = (0, 0, 0)$, 执行000: **G00 X23.45**, X 轴电机将移动 23.45 毫米或英寸。

G01 X* Y*** Z*** F***

X, Y, Z 三轴直线插补运动. 进给速度代表行程最长的轴的运行速度;

例如: 002: **G01 X12.34 Y43.21 Z5.678 F60**

进给速率 F60 代表 60 毫米/分, 或 60 英寸/分, 同时代表了Y轴速度。

G02 X* Y*** Z*** I*** J*** K*** F***

X, Y, Z 中的任意2轴在**顺时针**方向做圆弧插补, X, Y, Z 代表圆弧终点, I, J, K 代表圆弧中点坐标, F代表进给速率。

例如: 003: **G02 X12.0 Y0 I6.0 J0 F30.0**; X-Y 平面;

004: **G02 Z12.0 X0.0 K6.0 I0 F20.0**; Z-X 平面;

注: 由于圆弧的起点和终点坐标不能相同, 因此要画整圆必须调用 G02 两次;

另: 用 G02/G03 编程实现圆弧插补时, 为了获得精确的一组圆弧数据, 圆弧中点坐标必须

精确到小数点后五位，如 I2.54698、J34.56788。如果圆弧的起点、终点及半径已知，DmmDrv 的编辑界面里提供了精确的圆弧中点计算工具，使用非常方便。

G03 X* Y*** Z*** I*** J*** K*** F***

X, Y, Z 中的任意2轴在**逆时针**方向做圆弧插补，其它同G02

G04 P***

程序暂停；如 G04 P500 代表程序暂停 5 秒。

G17 : 选择 XY 平面。如: 000: G17 ; XY 平面

G18 : 选择 ZX 平面。如: 000: G18 ; ZX 平面

G19 : 选择 YZ 平面。如: 000: G19 ; YZ 平面

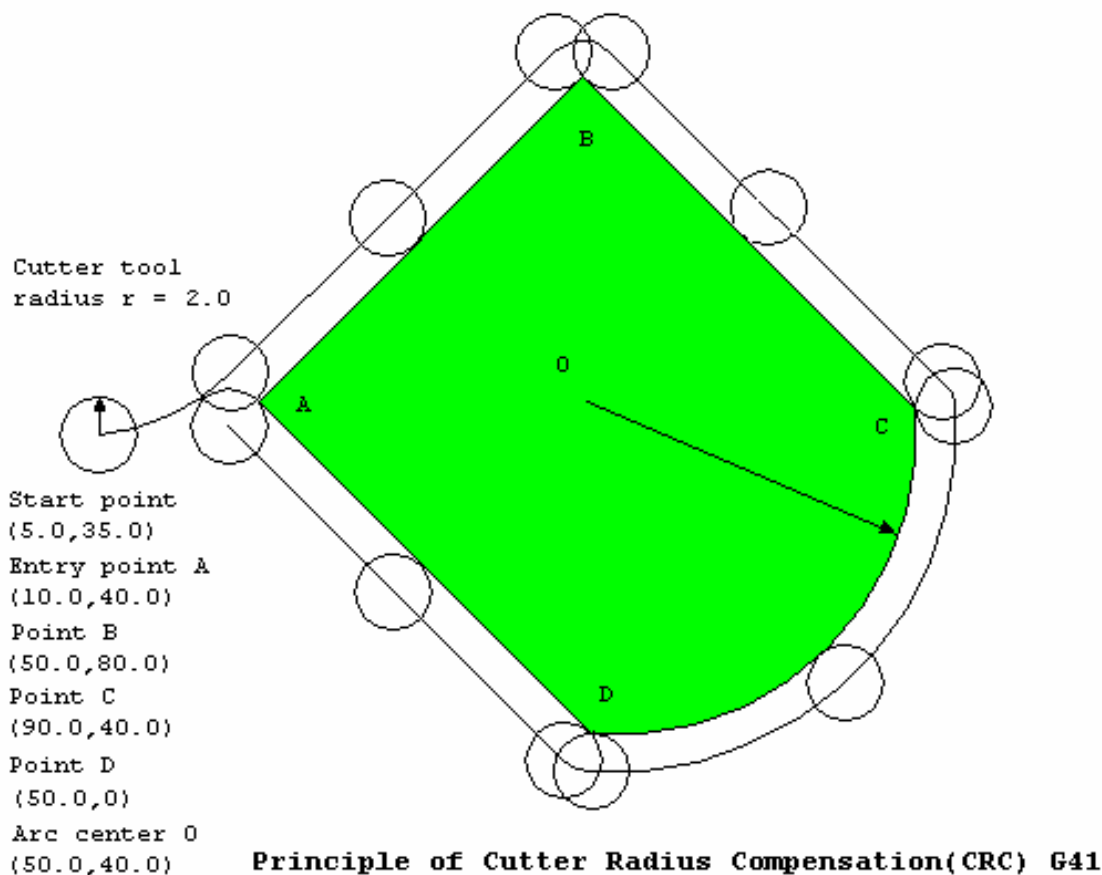
G20 : 设定英制输入，即当量单位为英寸；

G21 : 设定公制输入，即当量单位为毫米；

G40 : 取消切削刀具半径补偿。如: 000: G40 ; 取消补偿

G41 D* G01 X**** Y**** Z**** F***

切削刀具半径左侧补偿



G41 代表在XY, ZX, 或 YZ 平面的加工刀具半径补偿. 假设加工工件的外轮廓如上图绿色区域所示，刀具始终沿着工件左侧加工；刀具初次接触工件的 A 点称为切入点，离切入点较近的点称为开始点。首先通过 G00 或G01把刀具移动到开始点，然后通过 G41 命令，刀具将沿着一个弧线移动到与线段AB正切的位置，也即切入点A；当加工BC段时，驱动器将以B点为圆心自动走一段圆弧。

上述加工的完整 G 代码如下：

```

000: G17                ; 选择 X-Y 平面
001: G90                ; 设定绝对位置模式
002: G21                ; 设定 mm 输入
003: G51 I5.0 J5.0 K5.0 ; X Y Z 丝杠导程 5.0 mm
004: G01 X5.0 Y35.0 F20.0 ; 刀具移动到开始点
005: G41 D2.0 G01 X10.0 Y40.0 F20.0 ; 铣刀半径 2mm、左侧补偿, 移动到切入点
006: G01 X50.0 Y80.0 F20.0 ; 走 AB 线段
007: G01 X90.0 Y40.0 F20.0 ; 走 BC 线段
008: G02 X50.0 Y0.0 I50.0 J40.0 F20.0 ; 走圆弧 CD 插补
009: G01 X10.0 Y40.0 ; 走 DA 线段回切入点 A
010: G40                ; 取消切削刀具半径补偿
011: G01 X0.0 Y0.0 F20.0 ; 刀具回原点
012: M02                ; 程序结束

```

如果 AB 段是圆弧, 则 005 和 006 行的 G 码如下:

```

005: G41 D2.0 G01 X10.0 Y40.0 F20.0 ; 铣刀半径 2mm、左侧补偿, 移动到切入点
006: G02 X50.0 Y80.0 I50.0 J40.0 F20.0; 走 AB 圆弧

```

注: 使用了 G41 或 G42, 则刀具的半径就不能设置为 0 !

G42 D* G01 X**** Y**** Z**** F***

功能与 G42 类似, 只是刀具为右侧半径补偿;

G51 I* J*** K*****

输入 X Y Z 三轴丝杠副螺距; 如 G51 I5.0 J5.0 K5.0, 代表 X Y Z 三轴都需要, 并且螺距都为 5.0, 单位为毫米或英寸, 取决于设定。如果只用到 X-Y 轴, 则命令格式如下: G51 I*** J***; 如果只用到 X 轴, 命令格式如: G51 I***。

假设 P_x 代表 X 轴螺距, P_y 、 P_z 分别代表 Y、Z 轴的螺距; P_x 、 P_y 、 P_z 的设定可以互不相同, 如果三轴输入的行程相同而丝杠螺距不同, 驱动器会自动调整齿数 Gear_Number 来实现相同的行程。

如果 X Y Z 三轴都存在, 且丝杠螺距也已设定, 则:

$P_{avg} = (P_x + P_y + P_z) / 3.0$, 称为平均螺距, 驱动器要求:

$500 \leq \text{Gear_Number} = P_x * 4096 / P_{avg} \leq 16384$;

$500 \leq \text{Gear_Number} = P_y * 4096 / P_{avg} \leq 16384$;

$500 \leq \text{Gear_Number} = P_z * 4096 / P_{avg} \leq 16384$;

如果 $P_x = P_y = P_z$, 则 $P_{avg} = P_x = P_y = P_z$, 则三轴的 Gear_Number 都为 4096。

如果只有 X Y, G51 I*** J*** 定义了 X Y 的螺距, 则:

$P_{avg} = (P_x + P_y) / 2.0$, 称为平均螺距, 驱动器要求:

$500 \leq \text{Gear_Number} = P_x * 4096 / P_{avg} \leq 16384$;

$500 \leq \text{Gear_Number} = P_y * 4096 / P_{avg} \leq 16384$;

如果 $P_x = P_y$, 则 $P_{avg} = P_x = P_y$, 两轴的 Gear_Number 都为 4096。

如果只有 X 轴, G51 I*** 定义了 X 轴的螺距, 则:

$P_{avg} = P_x$.

$500 \leq \text{Gear_Number} = P_x * 4096 / P_{avg} \leq 16384$;

X Y Z 三轴的最大移动范围为: $(-8192 \sim 8191.999939) * P_{avg}$, 毫米或英寸。

当通过 G51 设定好 X Y Z 三轴丝杠的螺距后, 驱动器能自动调整自己的齿数设定, 保

证在输入相同的位置命令时，丝杠副移动相同的位移。

G90 : 绝对位置命令

G91 : 相对位置命令

M02 : 程序结束

M03 S***: 主轴顺时针旋转，如 045: M03 S1000，则主轴以1000转/分速度顺时针旋转。

M04 S***: 主轴逆时针旋转，如 046: M03 S1000，则主轴以1000转/分速度逆时针旋转

X, Y, Z 轴使用了 ID 号 0, 1, 2, 因此，在RS232/485联网运行时，主轴 ID 号必须设定为 3 。

最后: 每行代码字符总数不能超过 80 ！

制 造 商： 北京和利时电机技术有限公司
(原北京四通电机公司)

地 址： 北京市海淀区学清路11号

通讯地址： 北京2877信箱

电话总机： (010) 62932100

销售热线： (010) 62927938

传 真： (010) 62927946

网 址： www.hollysys-electric.com

深圳分公司： 0755-26581960/61/62/63

南京办事处： 025-84293632/52/53