M Servo Suite

软件手册

V1.1

©版权所有 上海安浦鸣志自动化设备有限公司

1 版本历史

版本	作者	参与者	日期	更改内容
1.0	雷友兵		2014-12-10	首次发布
1.1	唐康盛	雷友兵	2015-05-05	增加 M Servo Suite 的新特性

目录

1 版本历史 2
2 M Servo Suite 软件介绍 7
2.1 M Servo Suite 软件总览 7
2.2 M Servo Suite 软件安装 8
2.3 安装驱动程序 10
3 用 M Servo Suite 软件连接驱动器 11
3.1 连接至驱动器 11
3.1.1 使用串口通讯调试 11
3.1.2 使用以太网通讯调试 11
3.2 主界面 12
3.3 菜单栏 14
3.3.1 工程 14
3.3.2 配置 15
3.3.3 工具 16
3.3.4 Q 程序 19
3.3.5 驱动器 20
3.3.6 帮助 24
3.3.7 语言 25
3.4 工具栏 26
3.4.1 驱动器型号 26
3.4.2 通讯端口 26
3.4.3 伺服状态、清除报警 26
3.4.4 上传与下载 26
3.4.5 停止 27
4 使用 M Servo Suite 调试驱动器 28
4.1 第 1 步：配置 28
4.1.1 电机信息 28
4.1.2 控制模式选择 30
4.1.3 控制模式设置 31
4.1.4 速度模式 Velocity (I/O Controlled) 35
4.1.5 SCL/Q（Stream Command/Stand Alone）模式 37
4.1.6 Modbus 模式 38
4.1.7 力矩模式 Torque 39
4.1.8 CANopen 模式 41
4.1.9 位置误差报警阀值 41

4.2 I/O 配置 42
4.2.1 数字输入输出配置 42
4.2.2 输入输出功能 43
4.2.3 模拟输入配置 45
5 第 2 步：参数整定 46
5.1 伺服增益整定 46
5.1.1 增益参数介绍 46
5.2 使用 M Servo Suite 进行在线参数自动整定 47
5.2.1 第一步：选择电机 47
5.2.2 第二步：软限位功能 48
5.2.3 如何设定软件限位 49
5.2.4 第三步：执行自动整定 50
5.3 手动整定 52
5.3.1 位置环比例增益（KF） 52
5.3.2 积分增益(KI) 53
5.3.3 阻尼增益（KV） 55
5.3.4 微分增益(KD) 56
5.3.5 前馈增益(KK) 58
5.3.6 跟随因子（KL） 59
5.4 使用自动触发观测 60
6 第 3 步：Q 编程 60
6.1 Q 编程界面 61
6.2 当前程序段 62
6.3 参数编辑 62
7 运动仿真 64
7.1 初始化参数 64
7.2 点对点运动 64
7.3 点动 64
7.4 回原点 65
7.4.1 回原点模式 65
7.4.2 指令预览 68
8 SCL 终端 69
9 状态监测 70
10 SCL 指令集 72
10.1 指令 73
10.1.1 缓存指令 73

10.1.2 立即指令 73
10.2 使用指令 75
10.2.1 Q 系列驱动里的 Q 程序 76
10.2.2 SCL Utility 软件 76
10.3 Command Summary 指令总结 78
10.3.1 运动指令 78
10.3.2 伺服指令 80
10.3.3 配置指令 81
10.3.4 I/O 指令 82
10.3.5 通讯指令 83
10.3.6 Q 程序指令 83
10.3.7 寄存器指令 84
10.4 Host Command Reference 84
11 Q 编程参考 85
11.1 参考例程 85
11.1.1 点到点相对运动（Feed to Length） 85
11.1.2 点到点绝对运动（Feed to Position） 86
11.1.3 运动到传感器位置（Feed to Sensor） 87
11.1.4 循环（Looping） 88
11.1.5 跳转（Jumping） 88
11.1.6 程序调用（Calling） 89
11.1.7 多任务处理（Multi-tasking） 90
12 CANopen 参考 92
12.1 CANopen 通讯 92
12.2 为什么选用 CANopen？ 92
12.3 CANopen 参考例程 93
12.3.1 位置模式（Profile Position Mode） 93
12.3.2 速度模式（Profile Velocity Mode） 94
12.3.3 回原点模式（Homing Mode） 94
12.3.4 通用 Q 模式（Normal Q Mode） 95
12.3.5 同步 Q 模式（Sync Q Mode） 95
12.3.6 PDO 映射（PDO Mapping） 96
12.4 CANopen 相关下载 96
13 Modbus/RTU 参考 97
13.1 通讯地址 97
13.2 数据编码 97

13.3 通讯速率及通讯协议 98
13.4 Modbus/RTU 的消息帧 98
13.5 功能码 98
13.5.1 功能码 0x03，读多个保存寄存器 99
13.5.2 功能码 0x06，写单个寄存器。 100
13.5.3 功能码 0x10，写多个寄存器 101
13.6 Modbus 寄存器表 102
13.7 Command Opcode 命令操作码 104
13.8 MOONS 产品 MODBUS RTU 案例 107
13.8.1 位置控制例程 107
13.8.2 速度控制例程 109
13.8.3 内部 Q 编程控制例程 110
13.9 附件Ⅰ：功能码报文格式 114
13.9.1 功能 03 读取保持寄存器： 114
13.9.2 功能码 04 读取输入寄存器 114
13.9.3 功能码 06 预置单个寄存器： 115
13.9.4 功能码 16（10 HEX）预置多个寄存器 115
13.10 附件Ⅱ：MODBUS RTU 不正常响应及代码 117
14 Position Table 位置表模式 119
14.1 直线运动 119
14.1.1 直线运动软件设置流程 119
14.1.2 仿真操作 122
14.1.3 直线运动输入引脚定义 123
14.2 圆周运动 123
14.2.1 圆周运动的软件设置界面 124
14.2.2 圆周运动输入引脚定义 125

2 M Servo Suite 软件介绍

感谢您购买上海安浦鸣志自动化设备有限公司的 M 系列伺服产品。我们坚信凭借产品的卓越性能，良好的易用性和具有竞争力的价格会为您的应用带来超出预期的价值。

M2 系列交流伺服产品是由美国 AMP 及上海安浦鸣志研发技术团队共同打造的新一代交流伺服系统产品。M2 交流伺服驱动搭配安浦鸣志新一代 SM 系列伺服电机，提供 40/60/80mm 电机机座，功率范围覆盖 50/100/200/400/750W，特别适合高转速、高转矩、高精度、高安全性、长寿命的应用场合。

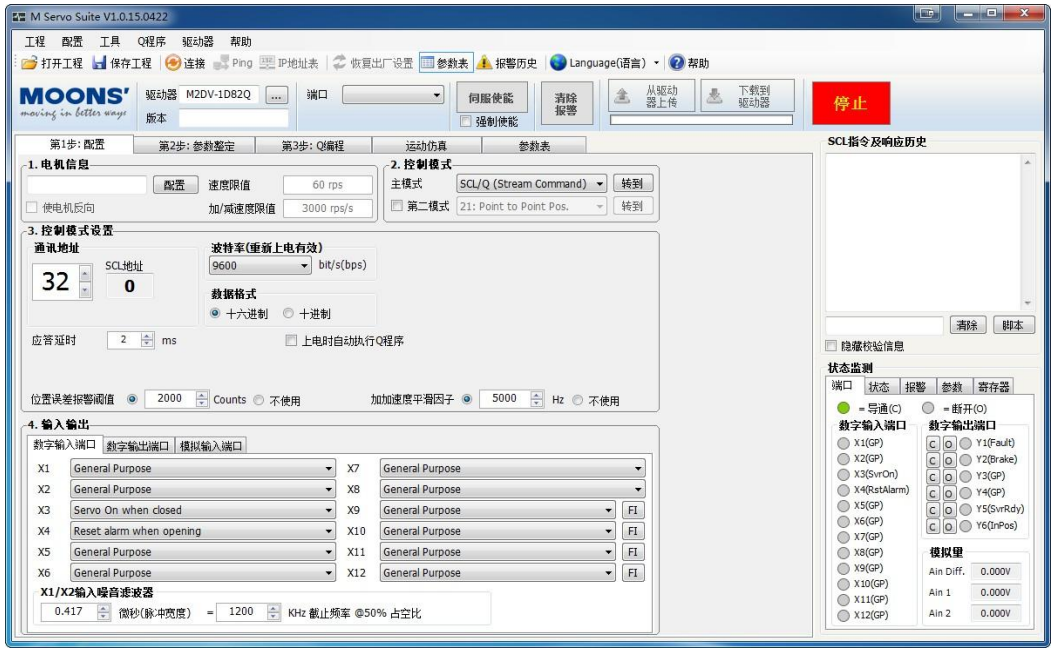
M2 交流伺服产品具备优秀的响应及整定时间，支持在线自动整定、共振抑制及灵活的内部位置表功能，支持独立 Q 编程，支持 Modbus/RTU，CANopen，Ethernet 等多种工业现场总线，具有安全可靠的技术保障。

可以广泛应用于 FA 工厂自动化设备、半导体制造设备、贴片机、PCB 制造设备、LED 制造设备、LED 分光检测及包装设备、包装机械、食品加工机械、医疗器械、传送机械、搬运机械及各种非标设备。

2.1 M Servo Suite 软件总览

M Servo Suite 软件用于配置、调试、整定 M 系列伺服产品。集成以下所有功能于一体：

- 控制模式及 I/O 配置
- 伺服参数自动整定
- 伺服参数手动微调
- 运动控制测试
- 多种状态监控
- Q 编程及调试



M Servo Suite 软件具有以下特性：

- 软件功能高度集成化
- 友好的图形用户界面
- 内置示波器功能
- 自动参数整定

如果您在使用电机或者软件的过程中遇到任何问题，或者您对我们的产品或者此用户手册有任何意见和建议，欢迎致电 (86)400-820-9661 或者传真至(8621)6296-8682. 另外您还可以发送电子邮件给我们：ama-support@moons.com.cn.

软件运行环境：

• Microsoft Windows XP (Service Pack 3)，Windows 7/8, Vista（32 位或 64 位）

• Microsoft .Net Framework 2.0

系统最低分辨率：

M Servo Suite 对显示器的分辨率要求最低为 1024 x 768。如果低于此分辨率将会有部分界面超出屏幕之外。

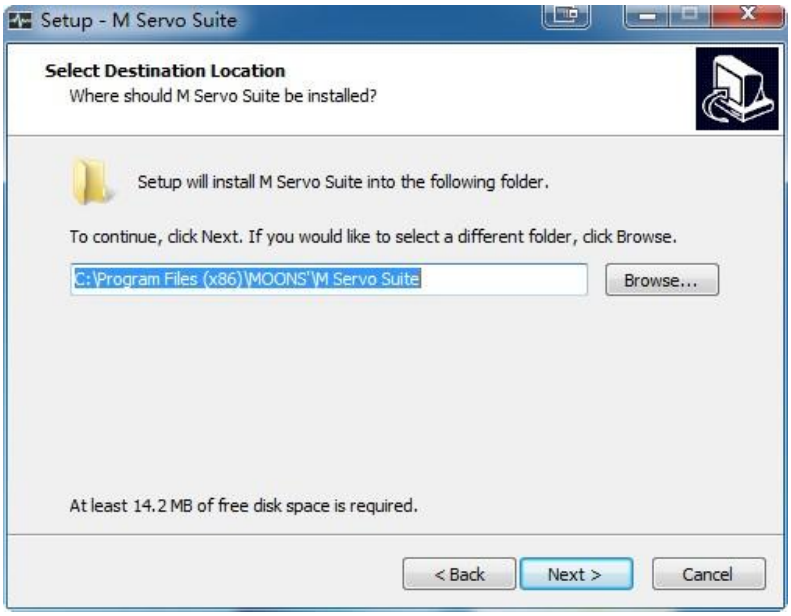
2.2 M Servo Suite 软件安装

M Servo Suite 可以从网站 www.moons.com.cn 上下载。

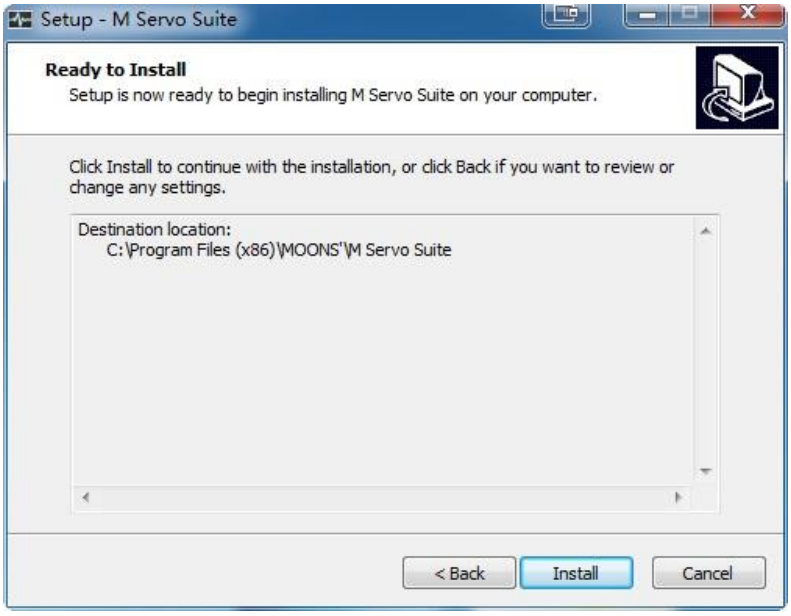
第一步. 打开 M Servo Suite 的安装文件后会出现安装向导。



第二步. 选择安装路径。



第三步. 准备安装。



第五步. 安装完成。



2.3 安装驱动程序

M2 驱动器使用 USB Mini 线进行通讯。在 M Servo Suite 软件安装时，会自动的安装驱动程序。

在与驱动器连接后，如果您的电脑提醒你需要安装驱动程序。您可以在软件的安装目录下找到 Driver Installation Tool 的文件夹。

在默认安装的情况，你可以在以下路径找到，

32 位操作系统：C:\Program Files\MOONS'\M Servo Suite\Driver Installation Tool

64 位操作系统：C:\Program Files (x86)\MOONS'\M Servo Suite\Driver Installation Tool

在此文件夹内，你可以选择：

“x86”文件件对应 32 位操作系统的驱动程序

“x64”文件件对应 64 位操作系统的驱动程序

根据你的操作系统，选择对应的文件夹，双击 MCP2200DriverInstallationTool，即可安装驱动程序。

3 用 M Servo Suite 软件连接驱动器

M Servo Suite 提供两种通讯类型，串口以及以太网。

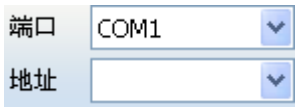
3.1 连接至驱动器

3.1.1 使用串口通讯调试

对于串口通讯的驱动器，连接主要包括以下几个步骤：

- 使用 USB mini 线连接您的 PC 与驱动器；
- 打开运行 M Servo Suite 软件；
- 选择正确的通讯端口；
- 给驱动器上电；
- M Servo Suite 自动识别驱动器型号与版本。

当运行 M Servo Suite 软件后，软件会将当前所有可用的串口列出.



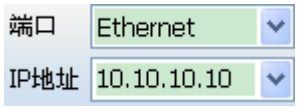
当驱动器与软件连接后，不管之前驱动器的通信波特率如何，软件都会将驱动器的通信波特率切换到 115200bps.

3.1.2 使用以太网通讯调试

对于以太网通信的驱动器来说，连接步骤如下：

- 使用 RJ45 网线连接驱动器和电脑或者路由器
- 运行 M Servo Suite 软件
- 选择通信方式为以太网，输入驱动器的 IP 地址
- 给驱动器上电

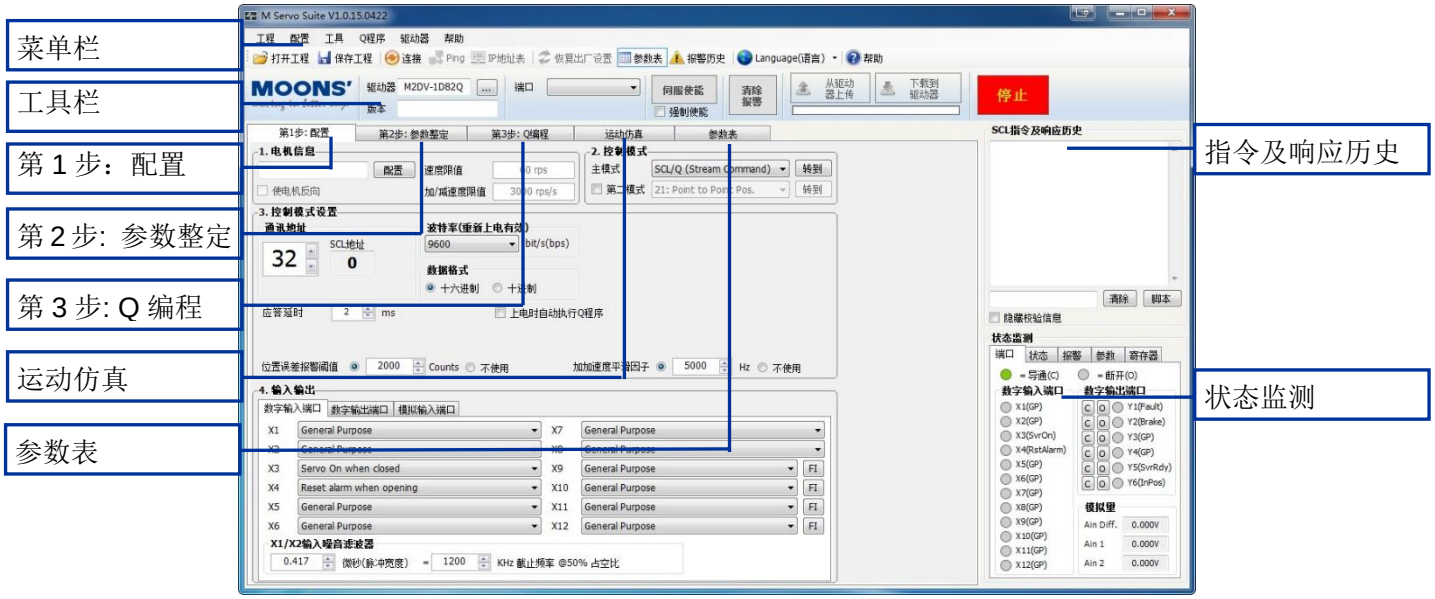
M Servo Suite 软件不能自动检测驱动器信息，您必须点击“连接”按钮，获取驱动器型号以及版本。



3.2 主界面

启动 M Servo Suite，点击开始菜单：开始 → 程序 → MOONS' → M Servo Suite → M Servo Suite.

主界面主要包括菜单栏，工具栏，主功能区（第 1 步：配置、第 2 步：参数整定、第 3 步：Q 编程、运动仿真、参数表），SCL 指令及响应历史和状态监测五个部分，如下图所示：



菜单栏

菜单栏提供了**工程、配置、工具、Q 程序、驱动器**帮助等功能。

工具栏

用于**打开工程、保存工程、进行通信设置、连接、PING、IP 地址表、恢复出厂设置、参数表、报警历史、驱动器使能或者非使能、驱动器信息显示、配置上传下载、紧急停止**等。

第 1 步: 配置

此选项卡中提供了驱动器相关基本功能配置功能，比如**电机信息、控制模式、控制模式设置、输入输出**等。

第 2 步: 参数整定

此选项卡提供**限位、参数自动整定、参数手动整定**及**陷波器**功能，并以图形显示参数整定的结果。

第 3 步: Q 编程

此选项卡提供了 Q 语言的编程环境，可以**编写，保存，调试运行，上传/下载 Q 程序**，此选项卡仅限-Q 型和-C 型驱动产品连接时有效。

运动仿真

此选项卡提供一些驱动器运动测试，例如**运动参数初始化、点对点运动、点动、回原点**等功能。

SCL 指令及响应历史

SCL 终端允许您发送 SCL 指令给驱动器。

状态监测

可以监控 I/O 状态、驱动器状态、报警信息、驱动器参数、及寄存器监视。

3.3 菜单栏



一级菜单	二级菜单	快捷键	功能
工程	打开工程	Ctrl+O	从磁盘打开工程文件（.mdvprj 格式）
	保存工程	Ctrl+S	将当前工程保存到磁盘（.mdvprj 格式）
	从驱动器上传	Ctrl+U	从驱动器上传工程
	下载到驱动器	Ctrl+D	将当前工程下载到驱动器
	打印	Ctrl+P	打印当前工程
	退出		退出 M Servo Suite 软件
配置	打开配置	Ctrl+Shift+O	从磁盘打开配置文件（.mdvcfg 格式）
	保存配置	Ctrl+Shift+S	将当前配置保存到磁盘（.mdvcfg 格式）
	从驱动器上传	Ctrl+Shift+U	从驱动器上传配置
	下载到驱动器	Ctrl+Shift+D	将当前配置下载到驱动器
	打印	Ctrl+Shift+P	打印配置
工具	固件下载		升级驱动器的固件
	标定		使用第三方电机时标定电机参数信息
	运动轨迹计算器		根据目标距离、加减速等参数模拟运动曲线
	导出 CANopen 参数		将 CANopen 参数导出到磁盘文件中
	CANopen Test Tool		打开 CANopen Test Tool 软件 (如果已安装)
Q 程序	打开 Q 程序		从磁盘打开 Q 程序文件（.qpr 格式）
	保存 Q 程序		将当前 Q 程序保存到磁盘（. qpr 格式）
	打开程序段		从磁盘打开程序段文件（.qsg 格式）
	保存程序段		将当前 Q 程序保存到磁盘（. qsg 格式）
	从驱动器上传		从驱动器上传 Q 程序
	下载到驱动器		将当前 Q 程序下载到驱动器
	清除 Q 程序		清除 Q 程序内容
	设置密码		设置 Q 程序的上传密码
	打印 Q 程序		打印参数表
驱动器	连接	Ctrl+R	连接驱动器
	停止	Atl+F5	紧急停止
	Ping		Ping 以太网通讯型的驱动器
	编辑 IP 地址表		编辑以太网通讯型的驱动器的 IP 地址表
	参数表		显示驱动器的参数表
	脚本		运行脚本文件
	恢复出厂设置	Ctrl+Shift+D	将当前参数恢复到工厂默认配置
	恢复整定参数默认值		
	报警历史	Ctrl+Shift+A	查看报警历史记录
	杂项		设置报警屏蔽、反电势、通讯及其它参数
帮助	关于		获取软件版本信息
	帮助内容	F1	点击打开软件帮助文件

3.3.1 工程

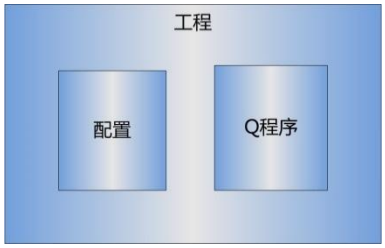
M Servo Suite 软件提供将您当前连接的驱动器中的配置及 Q 程序等作为一个工程文件保存到本地磁盘，也可以将原来保存在电脑中的工程文件显示到软件上，或者将工程文件直接打印出

来等功能。

用户也可以将驱动器内的配置及 Q 程序上传到电脑中，或者将已经保存好的工程文件下载到驱动器中。



对于支持 Q 编程的驱动器，工程的内容包含配置和 Q 程序，如下图所示：



保存工程时会同时保存配置和 Q 程序。如果想单独保存配置或 Q 程序，请使用配置菜单中的保存配置菜单或 Q 程序中的保存 Q 程序菜单。

对于不支持 Q 编程的驱动器，工程的内容与配置相同。

3.3.2 配置

M Servo Suite 软件也可以提供将您当前连接的驱动器中参数的配置作为一个配置文件保存到本地，也可以将原来保存在电脑中的配置文件显示到软件上，或者将配置文件直接打印出来等功能。

用户也可以将驱动器内的配置上传到电脑中，或者将已经保存好的配置文件下载到驱动器中。



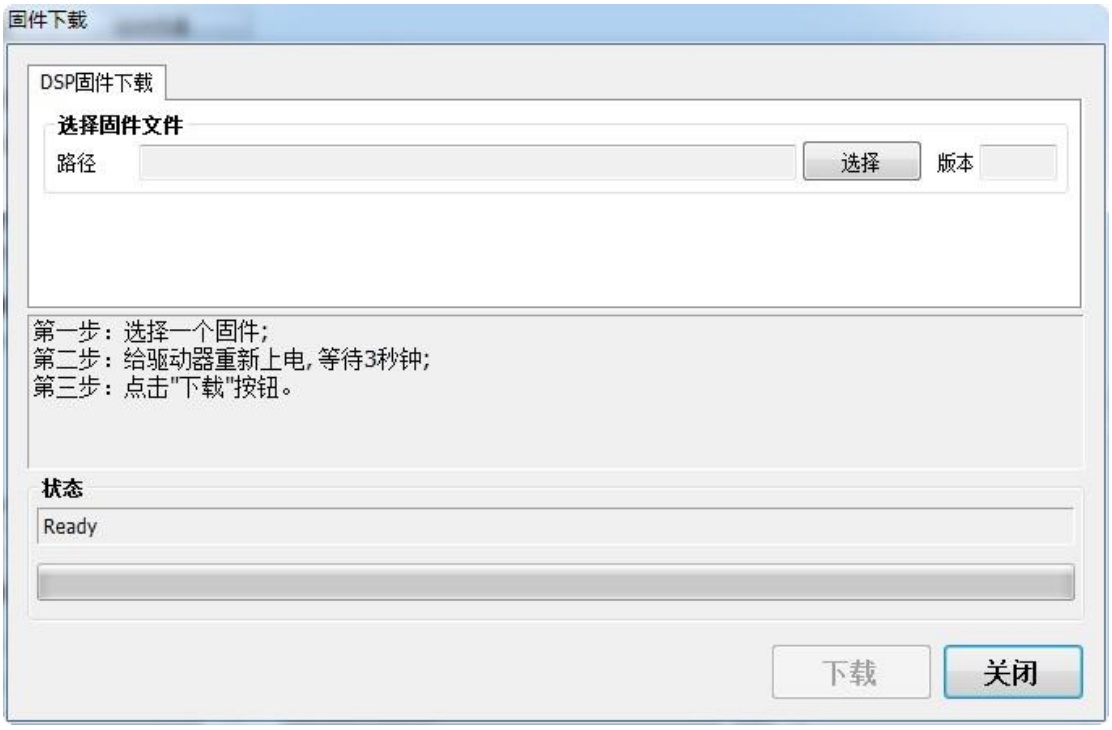
3.3.3 工具

M Servo Suite 提供了固件下载工具、使用第三方电机时的标定工具、运动轨迹模拟器以及针对与 CANopen 驱动器的参数备份工具及集成 CANopen Test Tool。



3.3.3.1 固件下载

固件下载用于升级驱动器的固件，升级前您最好先与鸣志的工程师确认固件是否与您使用的驱动器匹配。



固件下载步骤如下：

第一步：选择一个固件（固件的文件类型为.elf.e）；

第二步：给驱动器重新上电，等待 3 秒钟；

第三步：点击“下载”按钮。

注意：目前鸣志的驱动器不支持 RS485 组网时多台驱动器同时处于上电状态时的固件更新。RS485 组网模式下更新固件时您只能给需要更新固件的驱动器上电，其它驱动器的电源必须处于关断状态，否则会下载不成功。

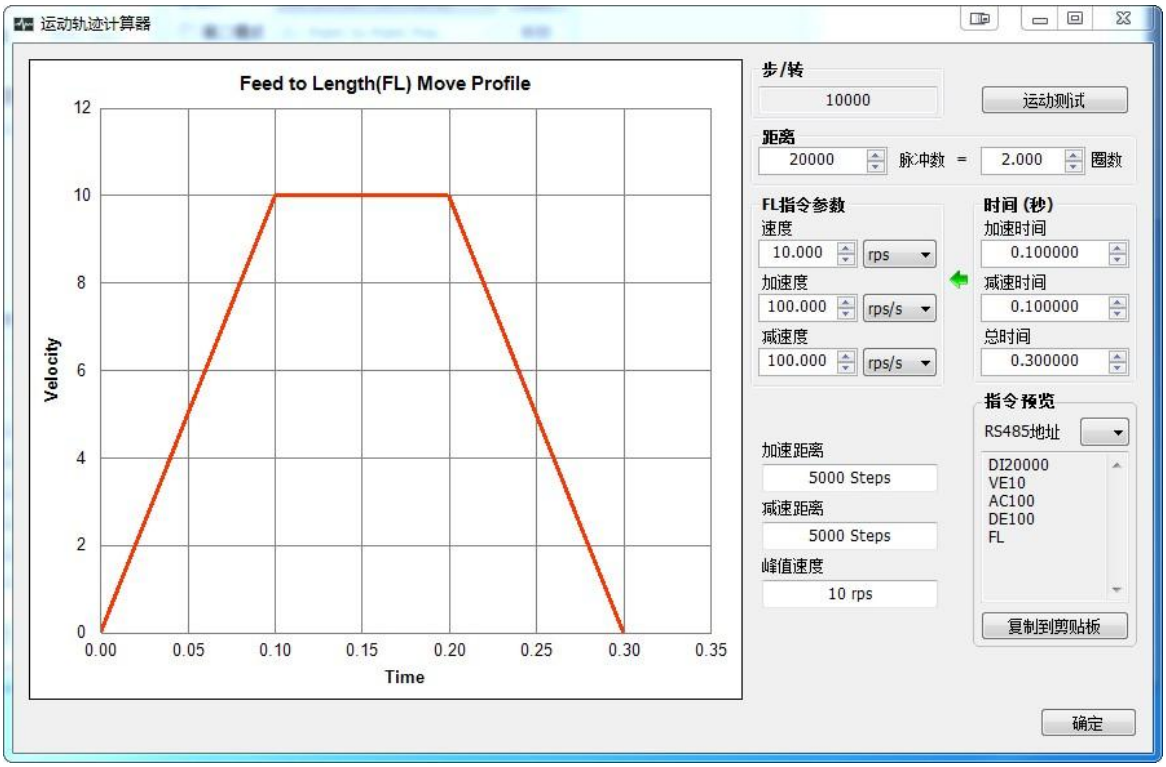
3.3.3.2 标定

当使用非鸣志公司生产的伺服电机时，需使用本工具进行标定。请与鸣志公司工程师确认你所用的伺服电机是否满足控制需求。



3.3.3.3 运动轨迹计算器

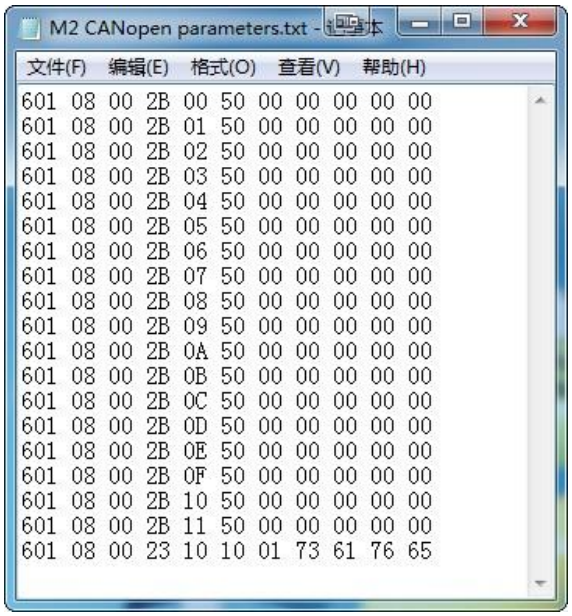
如果您事先需要对运动轨迹进行规划，例如在固定的距离下了解加速过程和匀速过程，运动轨迹计算器是一个实用的工具。



1. 在 ***FL 指令参数***中调整距离、速度、加速度等参数，左边的图形控件会自动画出相应的运动轨迹。
2. 同时如果您事先确定了加速时间、减速时间和总时间，您也可以点击“***时间换算成圈数***”，此时软件会自动计算相应的速度、加速度、减速度等运动参数。
3. ***指令预览窗口***可以预览上述操作生成的 SCL 运动参数指令
4. 点击***运动测试***按钮可以控制驱动器按上述操作的轨迹运行

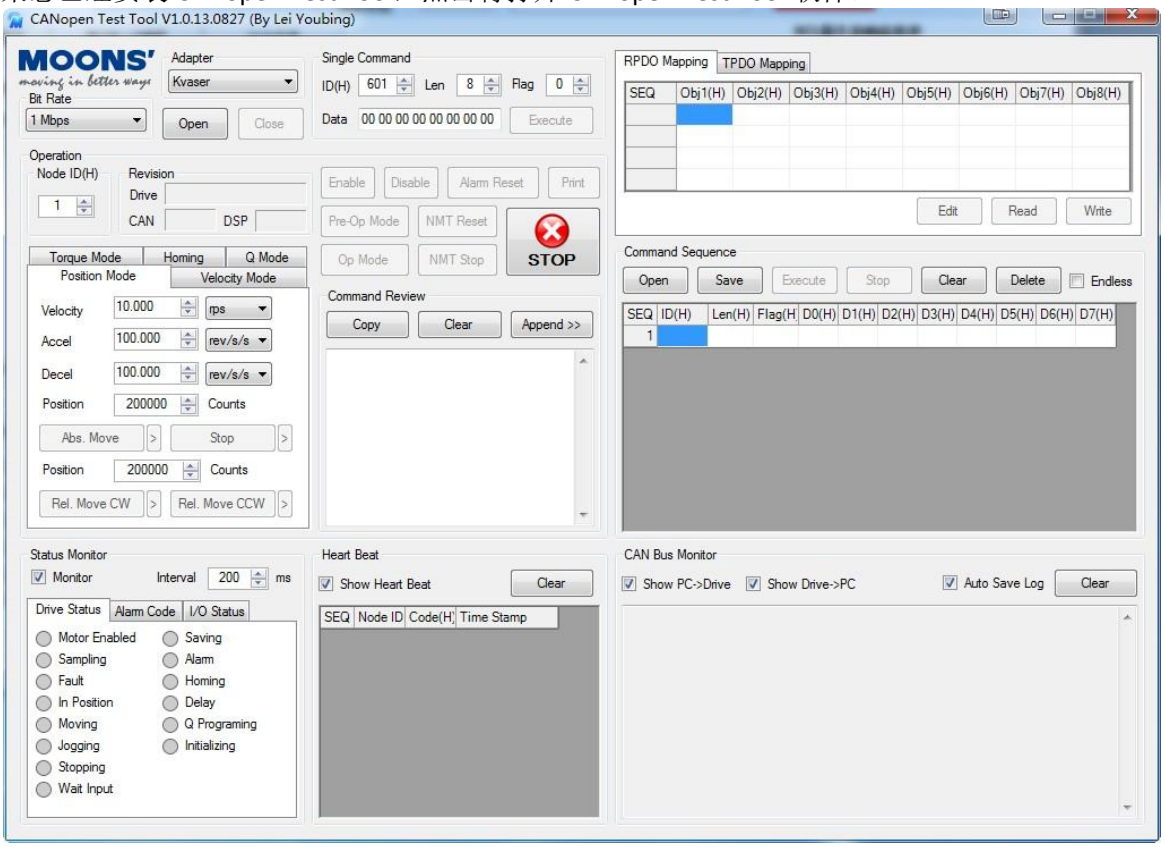
3.3.3.4 导出 CANopen 参数

如果您的驱动器是 CANopen 型的，那么您可以将 Tuning 参数导出到磁盘文件，然后用 CANopen Test Tool 打开后通过 CAN 总线下载到驱动器中。



3.3.3.5 CANopen Test Tool

如果您已经安装 CANopen Test Tool，点击将打开 CANopen Test Tool 软件。



3.3.4 Q 程序

如果 M Servo Suite 软件所连接的驱动器为支持 Q 程序的型号，用户可以使用本软件从驱动器中上传、下载到驱动器或者清除驱动器中现有的 Q 程序。也可以将编辑好的 Q 程序保存到电脑磁盘中，或者打开到电脑磁盘中的 Q 程序、Q 程序段等。



3.3.5 驱动器

驱动器菜单提供了如下功能：

菜单	名称	快捷键	功能描述
驱动器	连接	Ctrl+R	连接驱动器
	停止	Atl+F5	紧急停止
	Ping		Ping 以太网通讯型的驱动器
	编辑 IP 地址表		编辑以太网通讯型的驱动器的 IP 地址表
	参数表		显示驱动器的参数表
	脚本		运行脚本文件
	恢复出厂设置	Ctrl+Shift+D	将当前参数恢复到工厂默认配置
	恢复整定参数默认值		
	报警历史	Ctrl+Shift+A	查看报警历史记录
	杂项		设置报警屏蔽、反电势、通讯及其它参数

3.3.5.1 连接

连接功能使得软件与驱动器之间重新建立连接。



3.3.5.2 Ping

Ping 功能只用于以太网通信类型驱动器，用来确认软件与驱动器间的通信是否已经建立，如果通信已经建立，会返回驱动器的 ARM 版本号，以及驱动器 MAC 地址。



3.3.5.3 IP 地址表

IP 地址表提供用户自己编辑 IP 地址旋转开关 1~E 位置对应的 IP 地址。

注意：IP 地址表保存到驱动器后需要给驱动器重新上电才能生效。



3.3.5.4 参数表

参数表可以预览驱动器中所有已保存的参数。



序号	类别	命令	单位	软件	驱动器	默认值	范围	描述(双击查看更多信息)
000	PID	KP		10000	10000	8000	0 - 32767	全局增益1
001	PID	KG		12000	12000	10000	0 - 32767	全局增益2
002	PID	KF		16650	16650	6000	0 - 32767	比例增益
003	PID	KD		16200	16200	2500	0 - 32767	微分系数
004	PID	KV		18500	18500	8000	0 - 32767	阻尼系数
005	PID	KI		90	90	200	0 - 32767	积分增益
006	PID	KK		22563	22563	0	0 - 32767	加速度前馈增益
007	PID	KJ		5000	5000	5000	0, 10 - 5000	加加速度滤波频率
008	PID	VP		15000	15000	15000	0 - 32767	速度环比例系数
009	PID	VI		16076	16076	1000	0 - 32767	速度环积分系数
010	PID	KE		15000	15000	15000	0 - 32767	微分滤波因子
011	PID	KC		20000	20000	25000	0 - 32767	PID滤波因子
012	Control Mode	CM		7	7	21	1-8,11,12,15-18,21,22,25	主控制模式
013	Control Mode	CN		1	1	21	1-6,8,11,12,15-18,21	第二控制模式
014	Control Mode	PM		2	2	2	2, 5, 7, 8, 9	上电工作模式
015	Control Mode	JM		1	1	1	1 - 2	点动模式
016	Current Config	GC	0.01A	0	0	0	-180 - 180	力矩模式下的指令电流
017	Current Config	CC	A	1.800	1.800	0.500	0.000 - 1.800	最大电流
018	Current Config	CP	A	5.400	5.400	1.500	0.000 - 5.400	峰值电流
019	Current Config	HC	A	1.800	1.800	1.500	0.000 - 1.800	Hard Stop寻原点电流
020	Trajectory	VM	rps	60.000	60.000	60.000	0.025 - 100	最大速度
021	Trajectory	AM	rps/s	3000.000	3000.000	3000.000	0.167 - 5000	最大加速度
022	Trajectory	JS	rps	5.000	5.000	10.000	0.025 - 100	点动速度

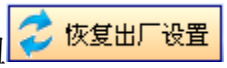
3.3.5.5 脚本

在脚本中可以编写一系列 SCL 指令并运行。如果勾中“循环执行”，软件在执行脚本时会循环执行脚本，直到点击“暂停”按钮。



如果勾中“执行时停止监控”，软件在执行脚本时会停止后台监测（例如状态监测），执行将会更加有效、准确（软件延时大大降低）。

3.3.5.6 恢复出厂设置

同快捷按钮

用来将驱动器中的参数恢复到出厂默认的参数。

注意：此操作将擦除所有您修改过的参数，在做此操作前，建议您先保存驱动器中的配置。

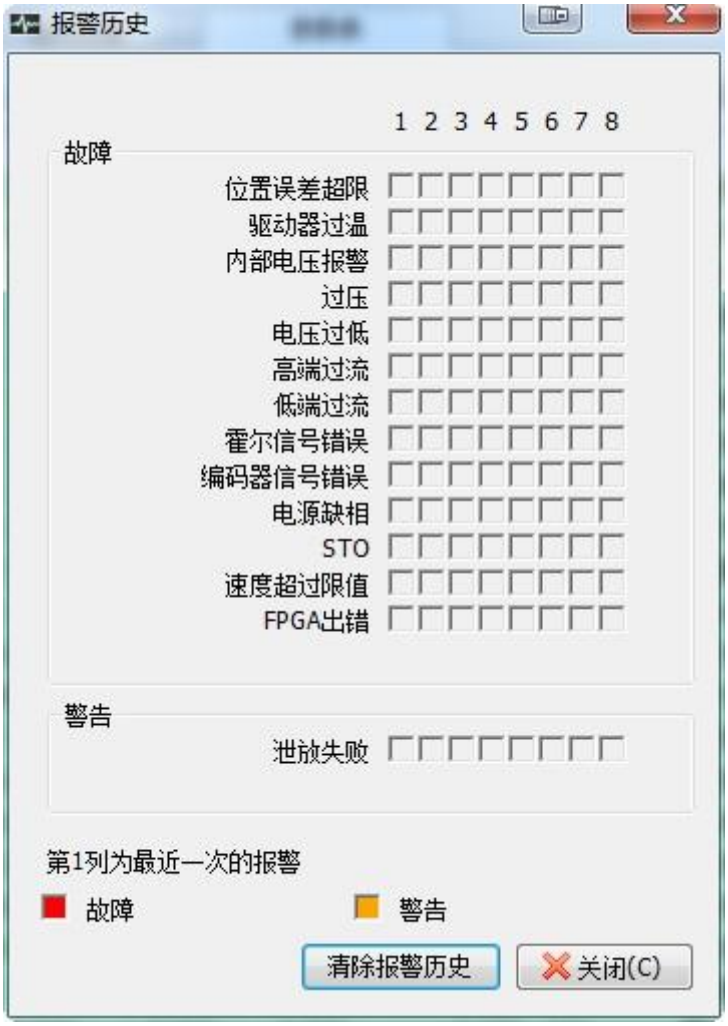
3.3.5.7 恢复整定参数默认值

用来将***第二步：参数整定***中的参数恢复成默认值。

3.3.5.8 报警历史

同快捷按钮

M2 系列交流伺服驱动器内部可以将一定时间内的所有报警记录保存下来，此信息可以帮助您分析解决相关问题。驱动器可以保存同种类型多达 8 次的警告历史记录。

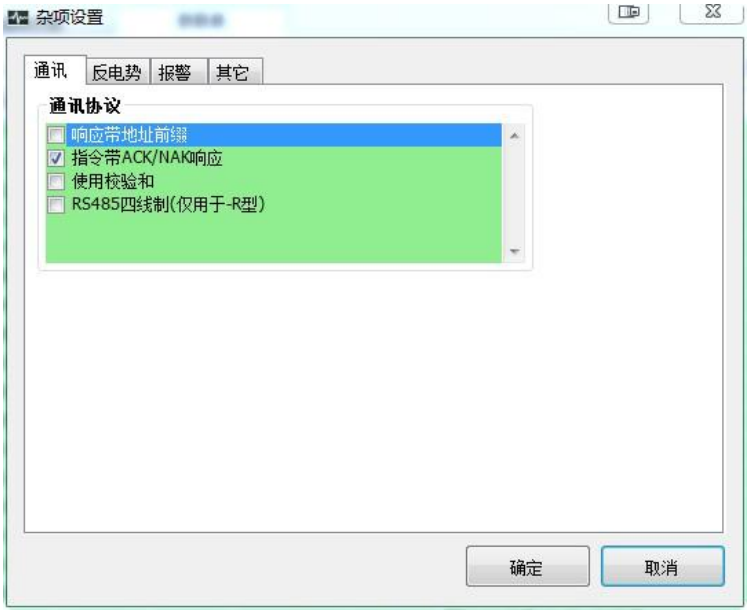


3.3.5.9 杂项

选项菜单提供设置报警屏蔽、反电势、通讯及其它参数的功能。

A. 通讯

用于设定上位机控制器与伺服驱动器之间的通讯协议。



响应带地址前馈：驱动器接收到 SCL 指令后，返回的字符串前带地址前缀
指令带 Ack/Nack 响应：以 Ack/Nack 响应接收到的指令
使用校验和：指令在通讯过程中使用校验和
全双工： 在 RS-485 通讯使用四线制接法时使用全双工通讯。

B. 反电势

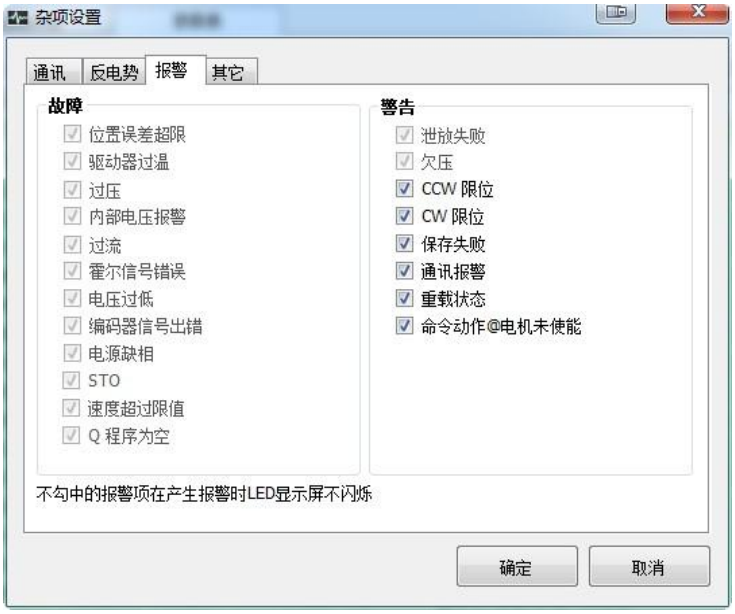
M2 驱动器中内置反电动势吸收模块，此选项用以设定驱动器使用外置吸收电阻的参数。



C. 报警菜单

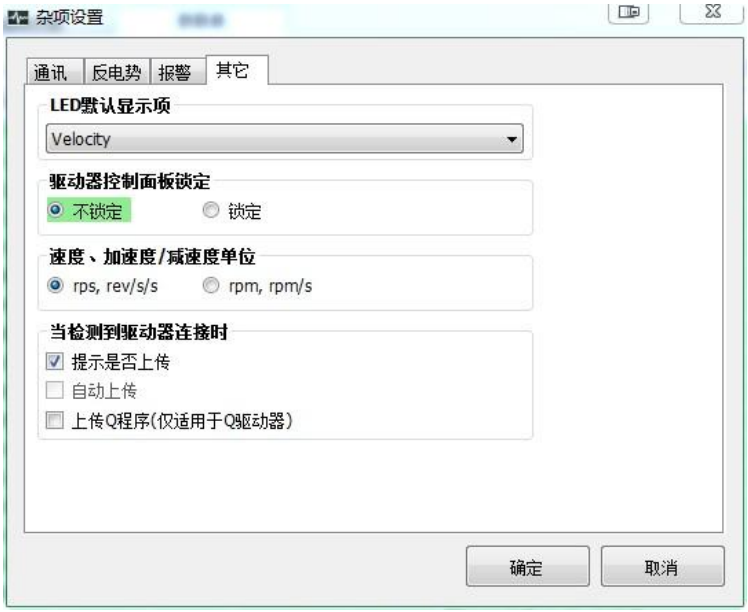
一般情况下，驱动器如果存在相关错误或者警告信息，驱动器的 LED 灯就会有相应的闪烁

方式来告知，如果出现某种警告后，您不想让驱动器 LED 灯闪烁的话，您可以使用此功能。



不勾选的警告信息在产生报警时 LED 灯将不再闪烁，但是此警告在报警历史记录中还是会记录。

D. 其它



LED 默认显示项：设定驱动器默认上电后的 LED 显示的内容。

驱动器面板锁定：设定是否锁定驱动器操作面板

速度、加速度/减速度单位：设定软件中显示的单位为 rps，rev/s/s 或者为 rpm，rpm/s/s。

当驱动器连接时：设定当软件与驱动器建立连接后的设置。

3.3.6 帮助

A. 关于

点击关于获取软件的版本信息。

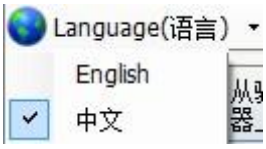


B. 帮助内容

点击获取帮助文档。

3.3.7 语言

M Servo Suite 支持中文和英语两种语言的显示，语言按钮下拉框提供快速切换显示语言的功能。



3.4 工具栏

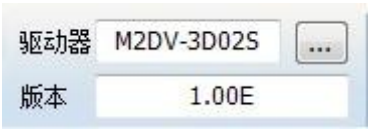
工具栏包含了 MOONS'的 Logo，驱动器型号、固件版本、通讯设置、伺服状态、报警状态、工程的上传和下载以及停止功能。



3.4.1 驱动器型号

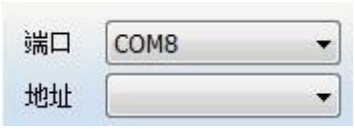
驱动器型号下拉列表中列出了当前通信类型下（串口、以太网）所有软件支持的 M2 系列伺服驱动器产品。

版本信息栏中用来显示驱动器的固件版本号。

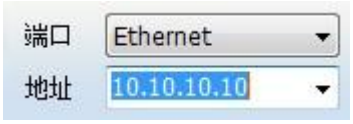


3.4.2 通讯端口

在连接驱动器之前需要设置驱动器的通讯方式。如果是 RS-485 的版本，则可以设定驱动器对应的地址以保证连接。



如果驱动器是 Ethernet 类型的驱动器，还需要配置驱动器的 IP 地址。



3.4.3 伺服状态、清除报警

伺服开关用来控制伺服状态. 即使能驱动器，或者非使能驱动器。如下图：



强制使能功能允许您在使能输入端口无论处于何种状态下均能将电机使能。

清除报警提供在遇到报警时清除报警。

注意：只有在驱动器的警告或者故障已经解除的情况下才能清除，否则依然会提示有报警。

3.4.4 上传与下载

你可以通过上传来复制您驱动器中设置和运行的参数到 PC 中。这样方便您改变一个正在运行的系统中的参数。

下载是用来从 M Servo Suite 中复制设置到驱动器中。如果您想改变驱动器设置，并传输到驱动器中，使用下载功能。如下图：



工具栏中“从驱动器上传”和“下载到驱动器”均指的是工程。

数据上传或者下载后，配置项的背景会变为绿色，意味着软件和驱动器的配置一致。如下图所示：



此时如果改变了配置，则背景会变为黄色，意味着软件的配置发生了变化。如下图所示：



当改变配置并下载后背景又会重新变为绿色，意味着下载成功。如下图所示：



如果驱动器没有连接或连接后没有上传下载，则所有配置项的背景会变为透明色或者白色，意味着软件和驱动器的配置没有进行过数据同步（上传或下载）。

3.4.5 停止

停止按钮提供立即停止电机的动作并将电机处于非使能状态。



4 使用 M Servo Suite 调试驱动器

使用 M Servo Suite 调试驱动器包括以下步骤：

第 1 步：配置

主要用来设定电机信息、控制模式选择及设定、输入输出引脚的功能设定。

第 2 步：参数整定

使用自动整定功能快速的设定增益参数，满足控制要求。

第 3 步：Q 编程

对于支持 Q 编程的驱动器，用本画面来编辑 Q 程序。

第 4 步：运动仿真

使用本界面可以显示点到点运动、电动、及实验各种回原点模式。

4.1 第 1 步：配置

配置包含电机信息、控制模式选择、控制模式设置和输入输出设置 4 步。如下图：



4.1.1 电机信息



电机信息出显示当前驱动器中所设定的伺服电机的信号，点击“***配置***”将弹出电机选择对话框，

在此界面中可以更改电机型号。



注意：请确保选择的电机型号与驱动器实际连接的电机型号是一致的。

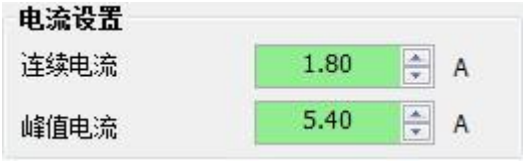
规格：此界面显示电机的连续电流及峰值电流（请勿修改）。

加减速限值：设定电机的最大速度及最大加速度。

电机方向：勾选“使电机反向”选框，可以使得电机转动方向与默认方向相反(此功能需重新上电才有效)。

4.1.1.1 电流设置

驱动器的电流设置必须要和相应的电机匹配，首先需要从电机的规格书中知道电机运行的额定电流。



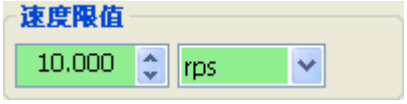
驱动器可以提供瞬时电流峰值。确保电机正常动作，驱动器在一秒后会自动减少电流来保证平均电流不会超过电机额定电流。禁止电机长期运行在高于额定电流的状态下。

不同的型号对应不同的峰值电流，请在使用之前查看您产品的规格书。

4.1.1.2 速度限值

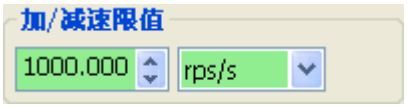
设置电机最大速度值。

提示：最大速度值在所有模式下有效，超出本值设定驱动器将产生失速报警。



4.1.1.3 加速度限值

设置电机最大加速度值。



4.1.2 控制模式选择

-S、-Q、-R 型驱动器具有两种控制模式：主模式和第二模式，您可以根据应用需求从控制模式下拉列表中选择控制模式。如下图。



主模式包含：Position(I/O Controlled)、Velocity(I/O Controlled)、SCL(Stream Command)、Torque、Position Table(仅-S 支持此模式)、SCL/Q(Stream Command)(仅-Q/-R 型支持)、Modbus(仅-R 型支持)。

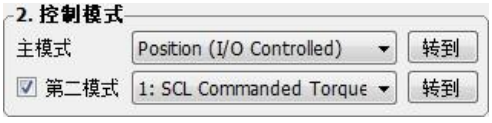
第二模式包括：SCL Commanded Torque、Analog Torque、Analog Torque+Dir、Analog Torque+R/S、Analog Torque+R/S+Dir、Analog Velocity、Analog Velocity+R/S、Fixed Velocity、Fixed Velocity+R/S、Fixed Velocity+CS、Fixed Velocity+R/S+CS、Point to Point Position。

控制模式的切换

当勾选第二模式时，输出口 X8 可以用来切换两个控制模式。点击“***转到***”按钮选择当前模式，并进行详细设置。

例如：

1） 选择主模式为 Position(I/O Controlled)，勾选第二模式，并选择为 Analog Velocity。即主模式为脉冲位置模式，第二模式为模拟量速度模式。



2） 点击***主模式***旁的“***转到***”按钮，界面 **3. 控制模式设置**转换为设置主模式的控制模式设定。



3） 点击***第二模式***旁的“***转到***”按钮，界面 **3. 控制模式设置**转换为设置第二模式的控制模式设定。



4.1.3 控制模式设置

4.1.3.1 位置模式 Position (I/O Controlled)

位置模式下有五种信号控制方式： 脉冲/方向、CW/CCW 脉冲、A/B 正交脉冲、差分模拟信号、单端模拟输入。



4.1.3.2 脉冲信号

信号控制是指用脉冲数字信号控制电机的位置。

脉冲信号类型

脉冲/方向：

接收电机控制器发出的信号。在这种模式下，输入脉冲频率决定速度，旋转方向由另外一个信号输入决定，您可以配置信号开或者闭来表示正转或反转。



CW/CCW 脉冲：

电机正转或反转由哪一路输入脉冲决定。驱动器有两个输入口，脉冲输入到其中一个会使电机顺时针转，脉冲输入另外一个会使电机逆时针旋转。



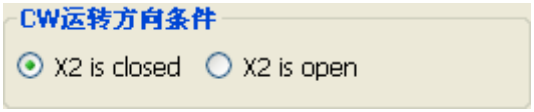
A/B 正交相位脉冲：

接受 A&B 正交脉冲，控制电机转动。



运转方向条件

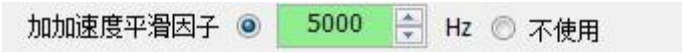
每种脉冲输入类型下都需要选择 CW 正转方向的条件。



加加速度平滑因子

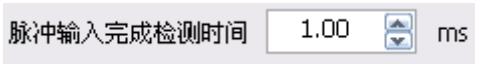
对速度和方向信号的动态滤波可以减少电机及机械系统的运动瞬变，使电机运行更加平滑，同时也可以减小机械磨损。

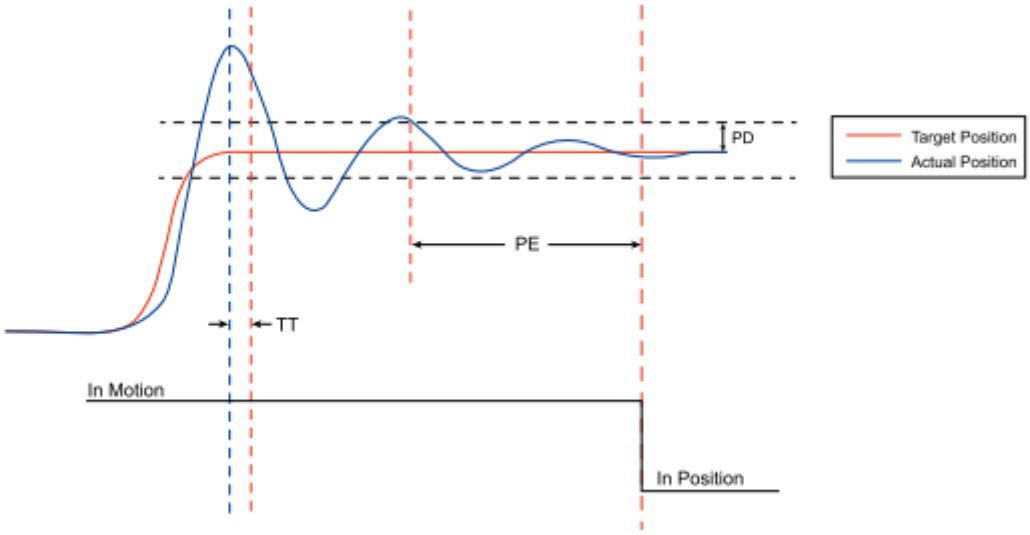
1) 数值越小，平滑效果越明显。
2) 加加速度平滑因子会对指令脉冲产生一定的延时 T，但不会影响到最终的定位精度



脉冲输入完成检测时间

用于设定一段时间。如果驱动器或电机在此设定时间段内没有收到脉冲信号，则认为上位控制器已停止发脉冲（即目标位置已经明确）。该参数将用于参与判断电机运动是否到位。详细情况请见 Host Command Reference 文档中的 TT 指令。





4.1.3.3 模拟信号位置模式

模拟量输入信号也可以作为位置控制的信号。



两种控制方式：差分模拟信号、单端模拟输入端口

两路输入信号可以被用作“差分”模式或一路输入被用作“单端”模式。差分信号可以降低噪声，尤其是在复杂电子环境下增强驱动能力。

范围： ±10V.

位置-建立一个输出对输入相对比例的增益值。

例如在位置模式下，如果范围被设置在+-10v 并且位置设置为 8000，那么当输入 10v 时电机会正转到 8000 的位置。-10V 时电机会有反向转动到-8000 位置。

偏置量-设置一个针对输入的偏移值来将电压偏移值置零或在需要时改变输入电压值。

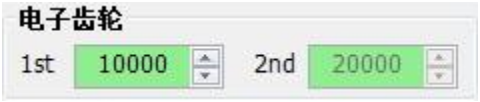
通常在模拟量输入系统中很难获得 0 电压值。使用偏置量可以调整外部控制器的输入电压到 0 值。“自动调整”功能可以自动检测并且纠正输入电压偏移。

死区-添加一个视输入电压为 0 的电压区域。

因为模拟量输入有不确定的特性，所以需要创造一个“死区”，在区域内输入电压视作为 0，不会影响到输出。例如，当使用操作杆控制电机时，用户可能想当操作杆在“零”位时没有任何的力矩输出，但是大多数操作杆没有这么精确而且依旧输出一部份电压，这时设置死区就可以消除这一部分电压的影响。

4.1.3.4 电子齿轮

电子齿轮允许你调整驱动器对脉冲输入信号对应的圈数。 M2 交流伺服系统具有两组电子齿轮，可以通过输入点 X9 切换。



例如：设定值为 10000，即驱动器每接受到 10000 个脉冲时，电机选择一圈。

4.1.3.5 电子齿轮比

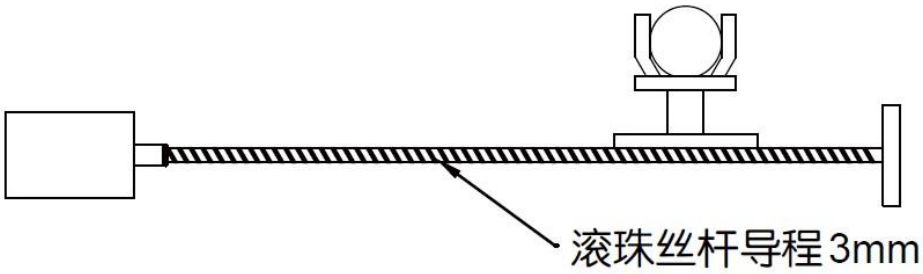
电子齿轮比是将从上位控制器输入的脉冲指令乘以所设定的电子齿轮比的值，作为对位置控制部的位置指令。



通过使用本功能，可任意设定单位输入指令脉冲的电机旋转、移动量如果电机每转所需脉冲

数为 10000 脉冲时，当电子齿轮比等于 1 时，上位机发送 10000 个脉冲，电机转动一圈。当电子齿轮比等于 1/2 时，则上位机每 2 个脉冲所对到电机转动脉冲为 1 个脉冲，即上位机发送 20000 个脉冲，电机转动一圈。

再例如：设定合理的电子齿轮比，将简化上位机发送脉冲数的计算。如下图：



丝杆导程为 3mm，当需要移动 4mm。

如果不使用电子齿轮比，计算所需要的脉冲数：

1) **计算圈数**

由于导程为 3mm，即电机每转一圈，工作台移动 3mm，那么移动 4mm 则需要转：“4/3 圈”

2) **计算所需脉冲数**

如果电机每转所需脉冲数为 10000 个脉冲，那么

$10000 \times \frac{4}{3}$=13333.33333…个脉冲

3) **输入脉冲数位 13333 个，且会产生累计误差。**

如果使用电子齿轮比

假设 1 脉冲对应 1um，电机每转所需脉冲数为 10000 个脉冲，电子齿轮分子为 a，分母为 b，那么电子齿轮比计算为：

$$\frac{3000}{10000} \times \frac{a}{b} = 1um$$

即当电子齿轮比为 $\frac{a}{b} = \frac{10}{3}$时，上位机发送 1 个脉冲，工件行走 1um。

4.1.4 速度模式 Velocity (I/O Controlled)

速度模式下可以设置以下参数：

速度控制类型：实时检测位置误差和仅速度控制

实时检测位置误差是指具有位置环，在运动过程中始终检测是否有存在位置误差超限。

仅速度控制是指只使用速度环的参数进行速度控制。注意在该模式下，为保证良好的控制效果，需要设定速度环比例增益 VP，及速度环积分参数 VI

速度控制方式：设定速度控制方式为固定速度、速度表还是与模拟量信号成正比。

加速度：设定速度模式下加速度的大小。

减速度：设定速度模式下减速度的大小。

速度模式的设置界面如下图所示。



速度控制方式有四种：

1) 固定速度
2) 根据 X10~X12 状态改变速度
3) 差分模拟量信号
4) 单端模拟量信号

4.1.4.1 固定速度

电机转速固定在设定值，启动或者停止由外部数字 I/O 设定。

加速度：设定速度模式下加速度的大小。

减速度：设定速度模式下减速度的大小。

速度固定在：设置需要的目标速度



4.1.4.2 根据 X10~X12 状态改变速度

根据 X10~X12 状态改变速度又称为多段速模式。



在该模式下，可以设定 8 段速度，并通过 X10\X11\X12 输入引脚的选择不同的速度。点击 ***“速度设置”*** 按钮可以设置详细的速度。



4.1.4.3 速度与模拟量输入成正比

速度模式有两种控制方式：差分模拟信号、单端模拟输入端口



1) 差分模拟量输入

两路输入信号可以被用作“差分”模式或一路输入被用作“单端”模式。差分信号可以降低噪声，尤其是在电子环境下增强驱动能力。



范围： ±10V.

速度-建立一个输出对输入相对比例的增益值。

例如在速度模式下，如果范围被设置在+-10v 并且速度设置为 10rps，那么当输入 10v 时电机会以 10rps 正方向转动。-10V 时电机会有反向 10rps 转动。

偏置量-设置一个针对输入的偏移值来将电压偏移值置零或在需要时改变输入电压值。

通常在模拟系统中很难获得 0 电压值。使用偏置量可以调整外部控制器的输入电压到 0 值。“自动调整”功能可以自动检测并且纠正输入电压偏移。

死区-添加一个视输入电压为 0 的电压区域。

因为模拟量输入有不确定的特性，就需要创造一个“死区”，在区域内输入电压视作为 0，不会影响到输出。例如，当使用操作杆控制电机时，用户可能想当操作杆在“零”位时没有任何的力矩输出，但是大多数操作杆没有这么精确而且依旧输出一部份电压，这时设置死区就可以消除这一部分电压的影响。

2) 单端模拟量输入口 1

在模拟量速度模式下，如果选择使用单端模拟量输入，那么模拟量输入端口 1 将作为速度限定，模拟量输入端口 2 可以选择作为力矩限定或者不使用。



选择单端模拟输入端口 1



设定模拟量输入端口 1

4.1.5 SCL/Q（Stream Command/Stand Alone）模式

SCL 给用户一个简单的通过串行端口控制电机驱动器的方法。这样就不需要独立的运动控制器来给驱动器提供脉冲/方向信号，同时也提供了一个简单的面向其它各种工业设备的应用途径，例如：具有有标准串行通讯端口的工业设备 PLC，工业 PC 或者 HMI 等。

Q 编程是一种单轴运动控制的方式，适用于鸣志可编程式的步进及伺服驱动器。Q 编程允许用户使用 Host Command 语句为-Q 驱动器创建功能复杂的可独立运行的程序，支持的指令包括运行控制、I/O、驱动器配置与状态、数学运算、寄存器操作和多任务处理。具有如下特点：

- 单轴运动控制
- 执行驻留程序

- 多任务处理
- 条件判断
- 数学运算
- 寄存器操作
- 运动轨迹模拟



通讯地址：RS-485 型产品需要设置通讯地址

应答延时：

驱动回复上位机指令时候的应答延时。通常在使用 2 线式接法的 RS485 通信时很有必要。因为同一组线用来接收和发送数据，在接收和发送数据见就必须加上应答延时以确保正常通信。

波特率

串行通信中的上电后生效波特率。该值被配置后将会立即被保存但不会立即生效，直到下次上电才生效，所以上位机软件可以随时配置该值。

数据格式

用以设定在通讯过程中数据传输的格式是 16 进制还是 10 进制。

上电时自动执行 Q 程序

如果驱动器支持 Q 编程，勾中时驱动器在每次上电时会自动从 Q 程序的第 1 段开始运行。

4.1.6 Modbus 模式

在支持 Modbus 的产品系列中，网络中的每个从设备都必须分配给一个唯一的地址，只有符合地址要求的从设备才会响应主设备发出的命令。在 Modbus 地址“0”是广播地址 ，不能作为从站地址。Modbus/RTU 下从站地址的范围是 1 到 32。



通讯地址

Modbus 协议下的从站地址，与 SCL 地址（与 M Servo Suite 软件通讯时的地址）之间的对应关系如下表：

节点 ID	1	2	3	4	5	6	7	8
SCL 地址	1	2	3	4	5	6	7	8
节点 ID	9	10	11	12	13	14	15	16
SCL 地址	9	:	;	<	=	>	?	@
节点 ID	17	18	19	20	21	22	23	24
SCL 地址	!	"	#	$	%	&	'	(
节点 ID	25	26	27	28	29	30	31	32
SCL 地址	)	*	+	,	-	.	/	0

波特率

串行通信中的上电后生效波特率。该值被配置后将会立即被保存但不会立即生效，直到下次上电才生效。

上电时自动执行 Q 程序

勾中时驱动器在每次上电时会自动从 Q 程序的第 1 段开始运行。

Modbus 模式下可以通过指令（地址：0x40125）调用 Q 程序。

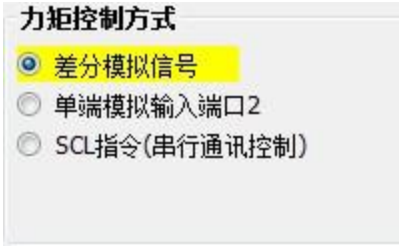
32 位 Word 顺序

在 Modbus/RTU 模式下，传送 32 位数据的格式是 Big-Endian(高位优先)还是 Little-Endian(低位优先)。

4.1.7 力矩模式 Torque

当驱动器设置在力矩模式，您可以定义输出的力矩大小也就是电机输出电流及旋转方向。在这个模式下电机速度取决于电机带动的负载。

注意：如果电机没有连接负载或空载，下载这个模式可能会使电机加速到很高的速度。



力矩模式分为两种控制方式：模拟信号和 SCL 指令。

4.1.7.1 模拟信号控制方式

力矩模式下的模拟信号类型有两种：差分模拟信号、单端模拟输入端口 2

1) 差分模拟信号

两路输入信号可以被用作“差分”模式或一路输入被用作“单端”模式。差分信号可以降低噪声，尤其是在电子环境下增强驱动能力。

范围：±10V

电流-建立与输入模拟量成正比的增益值。

例如在力矩模式中，如果范围被设置在±10v 并且电流设置为 2A，那么当输入 10v 时电机会有 2A 电流通过。-10V 时电流会有反向 2A 通过。

偏置量-设置一个针对输入的偏移值来将电压偏移值置零或在需要时改变输入电压值。

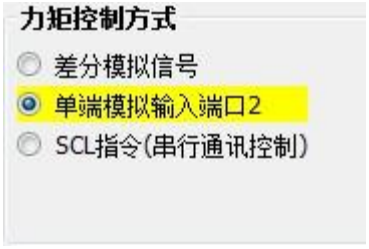
通常在模拟系统中很难获得 0 电压值。使用偏置量可以调整外部控制器的输入电压到 0 值。“自动调整”功能可以自动检测并且纠正输入电压偏移。

死区-添加一个视输入电压为 0 的电压区域。

因为模拟量输入有不确定的特性，就需要创造一个“死区”，在区域内输入电压视作为 0，不会影响到输出。例如，当使用操作杆控制电机时，用户可能想当操作杆在“零”位时没有任何的力矩输出，但是大多数操作杆没有这么精确而且依旧输出一部份电压，这时设置死区就可以消除这一部分电压的影响。

2) 单端模拟输入端口 2

在模拟量力矩模式下，如果选择使用单端模拟量输入，那么模拟量输入端口 2 将作为力矩限定输入，模拟量输入端口 1 可以选择作为限定力矩模拟下的电机转速。



选择单端模拟输入端口 2



设定模拟量输入端口 2

4.1.7.2 SCL 指令

SCL 指令控制方式需要主机发送 SCL 指令 GC 来控制电机的输出力矩。



4.1.8 CANopen 模式

用于设置 CANopen 控制型驱动器的节点 ID 以及 I/O 功能，配置完成后，在 CAN 总线上通信参照国际标准的 CANopen 协议通信。



Node ID

CANopen 总线上的每个节点必须要有独一无二的节点地址。CANopen 节点地址是用 7 位二进制码表示，范围是 1~127，即 16 进制 0x01~0x7F。

CAN 波特率

CANopen 总线通讯需要设定一致的通讯速率。CANopen 通信型驱动器支持 8 种通信波特率。

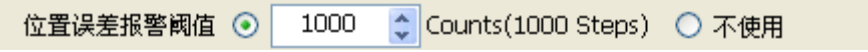
通讯速率
1Mbps
800Kbps
500Kbps
250Kbps
125Kbps
50Kbps
25Kbps
12.5Kbps

4.1.9 位置误差报警阀值

位置误差是编码器返回的实际位置和要求位置之间的差距。对于伺服系统来说微小的位置误差是正常的。但有时线圈可能损坏，传感器工作失败或电机堵转，您需要在发生任何损害之间尽快知道所有这些情况发生的原因，所以通过对位置误差的限制，当这些情况发生时驱动器会立即

停止给电机供电。

误差限制最小可以设置为 10 编码器计数，最大设置为 32000。当你首次使用时应该将这个值设置大些或者勾选不使用，这样驱动器不会在您试验运行参数的时候报警停机。当驱动器设置合适后，你可以通过正常运行时误差的大小来设置一个合适的值。例如：设置 M Servo Suite 示波器来检测位置误差，然后进行一些极限的测试，比如在最大速度或最大加速度下运行，如果位置误差显示 50，那么应该将“位置误差报警阀值”设置为大于 50 的值，比如 100。

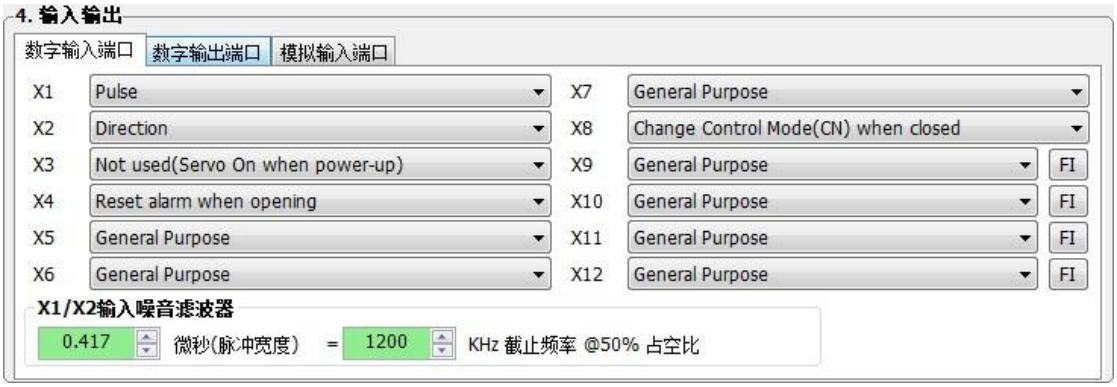


4.2 I/O 配置

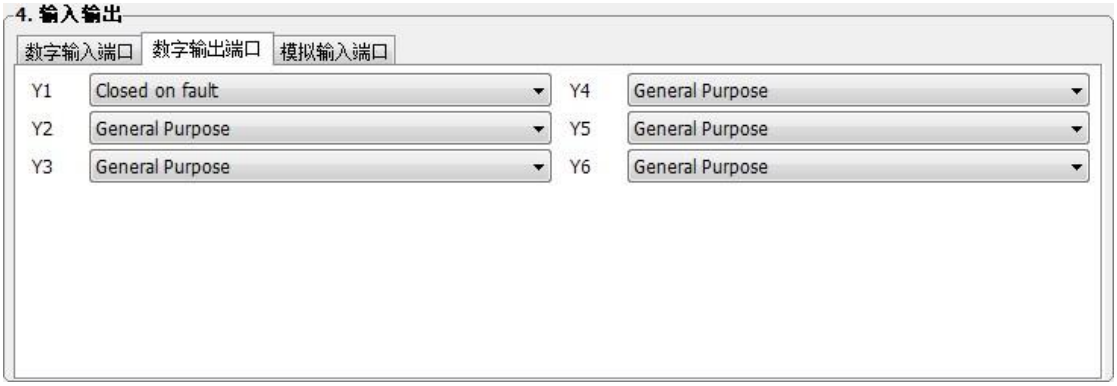
I/O 配置分为数字输入输出配置和模拟输入配置。

4.2.1 数字输入输出配置

数字输入输出配置包含数字输入（X）和数字输出（Y）的配置。详细的数字输入输出功能介绍请参考《用户手册》。

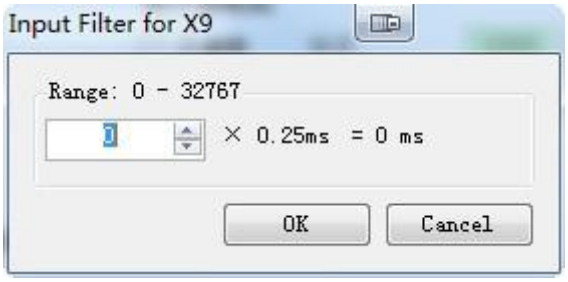


数字输入(X)



数字输入(Y)

4.2.1.1 FI 输入滤波器

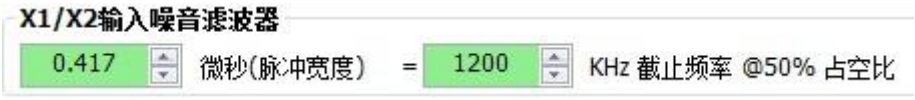


如果在输入端口配置后面带有“FI”按钮表示此 Input 支持 FI 输入滤波器。

FI 输入滤波器的范围是 0 – 32767。0 表示禁用，非 0 值表示 FI 输入滤波器起效。滤波器设置值表示数字输入端口电平保持的指令周期数。对于鸣志的伺服产品，指令周期为 0.25ms。例如 X3 的 FI 设置为 100，表示在 X3 上的电平变化必须维持 100×0.25 = 25ms 以上驱动器才认为 X3 的电平发生变化。

4.2.1.2 X1/X2 输入噪音滤波器

输入噪音滤波器是一种低通滤波器，高于指定频率的脉冲将被忽略。设置脉冲宽度后，软件自动计算频率值，如下图：



设置输入噪音滤波器时请保持适当的余量，推荐保持 50%的余量。

注意：软件的计算是以 50%的占空比进行的，如果脉冲信号的占空比不是 50%，请按照较窄的电平信号进行计算。

4.2.2 输入输出功能

4.2.2.1 输入功能

通用	信号名称	脚位	功能
X1	X1+	3	此引脚有 3 种功能 ● 在位置模式下作为 STEP 脉冲信号输入，可以接受脉冲信号、正发脉冲信号、正交脉冲信号中的一路 ● 当工作于转矩和速度模式时，该引脚可接受启动/停止信号。 ● 通用输入信号。
	X1-	4	
X2	X2+	5	此引脚有 3 种功能 ● 在位置模式下作为 DIR 脉冲信号输入，可以接受脉冲方向信号、正发脉冲信号、正交脉冲信号中的一路 ● 当工作于转矩和速度模式时，该引脚可接受方向信号用于切换转矩或速度方向 ● 通用输入信号
	X2-	6	
X3	X3+	29	● 使能信号，伺服使能/非使能信号 ● 通用输入信号
	X3-	31	

X4	X4+	35	● 报警清除信号，清除当前驱动器的报警内容 ● 通用输入信号
	X4-	34	
X5	X5+	8	● 正反转限位信号 ● 通用输入信号
	X5-	2	
X6	X6+	9	● 正反转限位信号 ● 通用输入信号
	X6-	1	
X7	X7+	39	● 伺服增益选择，在所有模式下使用 ● 通用输入信号
	X7-	38	
X8	X8+	12	● 控制模式切换，在 CM 和 CN 配置的两种模式间切换 ● 通用输入信号
	X8-	32	
X9	X9	26	● 切换脉冲输入电子齿轮比，切换脉冲输入电子电子齿轮比 ● 通用输入信号
X10	X9	27	● 脉冲输入禁止，使用此功能，可以禁止接受外部脉冲信号 ● 8 段速度选择，Input1 ● 通用输入信号
X11	X11	28	● 8 段速度选择，Input2 ● 通用输入信号
X12	X12	30	● 8 段速度选择，Input3 ● 通用输入信号

4.2.2.2 输出功能

通用	信号名称	脚位	功能
Y1	OUT1+	37	此引脚有 2 种功能 ● 报警输出 ●通用输出信号
	OUT1-	36	
Y2	OUT2+	11	此引脚有 2 种功能 ●电机刹车控制信号 ●通用输出信号
	OUT2-	10	
Y3	OUT3+	42	● 标示电机输出转矩是否达到限制值 ● 通用输出信号
	OUT-	33	
Y4	OUT4+	43	● Moving 信号输出，当动态位置误差小于设定值时，输出信号 ● 速度到达输出，电机转速达到设定速度时输出信号 ● 通用输出信号
	OUT-	33	
Y5	OUT5+	40	● Servo-Ready 伺服初始化准备完成输出 ● 通用输出信号
	OUT5-	41	
Y6	OUT6+	14	● 位置到达信号输出，当静态位置误差小于设定值时，输出信号 ● Tach Out 输出，可配置为一圈输出多少个脉冲功能 ● 通用输出信号
	OUT6-	13	

4.2.3 模拟输入配置



4.2.3.1 模拟信号输入低通滤波器

模拟信号滤波器用于减少在使用模拟量控制驱动器运动时外部干扰对驱动器的影响。



模拟输入信号滤波器

4.2.3.2 模拟量输入端口设置

范围：±10V

偏置量-设置一个针对输入的偏移值来将电压偏移值置零或在需要时改变输入电压值。

通常在模拟系统中很难获得 0 电压值。使用偏置量可以调整外部控制器的输入电压到 0 值。“自动调整”功能可以自动检测并且纠正输入电压偏移。

死区-添加一个视输入电压为 0 的电压区域。

5 第 2 步：参数整定

如同市面上大多数伺服驱动器一样，我们使用复杂的算法和电路来控制力矩，速度和电机与负载的位置。编码器可以将电机的位置反馈给驱动器，这样，驱动器可以不断的改变供给电机的电压和电流直到电机按要求动作。这叫做“闭环控制”。其中一个环路控制电机电流的大小。这一环路只需指定电机可以正常运行且不过热的最大电流。PID 环路将编码器反馈回来的电机实际位置和目标位置作比较。实际位置和目标位置的差值叫做位置误差，PID 环路通过三种方式来纠正这个位置误差：比例，积分，微分。

M2 伺服驱动器还增加了一些参数来达到更好的控制——阻尼增益和前馈增益。

5.1 伺服增益整定

伺服增益整定是优化伺服单元响应性的功能。

对于来自上位系统的指令，驱动器需尽可能无延迟且精准的按指令要求驱动电机。为使电机动作更接近指令、最大限度地发挥机械性能，从而需要进行增益调整。（图 6-1）

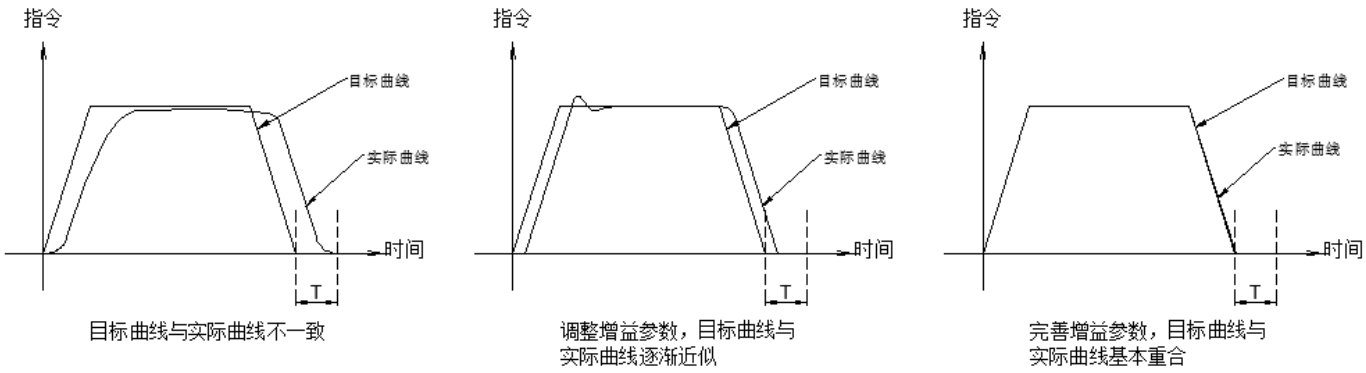


图 6-1

伺服增益通过多个参数控制，例如：全局比例系数（KP）、位置环比例系数（KF）、微分环节（KD）、阻尼系数（KV）、积分环节（KI）、加速度前馈系数（KK）、跟随因子（KL）、微分滤波系数（KE），以及控制器输出滤波系数（KC）。

所谓的 PID 参数整定也就是通过对这些参数的试取以满足运动系统的性能要求。一般情况下，刚性高的机械可通过提高伺服增益来提高响应性。但对于刚性低的机械，当提高伺服增益时，可能会产生振动，从而无法提高响应性

5.1.1 增益参数介绍

全局比例系数 KP：该参数是整个控制器的一个比例系数，它直接决定了系统的刚性，该参数对控制器后面所有的参数（除了前馈 KK）均进行了一次比例增益。通常推荐 6000~16000，可按照 6000~8000~10000~12000~14000~16000 试取，一般选取默认参数，不推荐改动。

位置环比例系数 KF：该参数对位置误差再进行一次比例系数增益，可提升系统的刚性及减小系统误差。

微分系数 KD：该参数是位置误差的微分增益，可抑制抖动，使系统运行平稳。

积分系数 KI：该参数是速度误差的积分增益，可消除系统的稳态误差。

阻尼系数 KV：该参数是速度误差的比例增益，可提升系统的刚性及抑制系统的振动。

加速度前馈系数 KK：该参数是加速度前馈的比例增益，可显著提升系统跟随性从而有效抑制加减速结束时的超调。

微分滤波系数 KE：通常微分项都带有一个低通滤波器，该参数就是决定该滤波器截止频率的因子。

控制器滤波系数 KC： PID 控制器的输出都往往带有一个低通滤波器，该参数就是决定该滤波器截止频率的因子。理论上截止频率可取系统要求最大带宽的 2 倍即可。

跟随因子 KL: 较高的值会降低系统的噪音，消除过冲，但同时会降低系统的动态跟随性。较低的值会提高系统的刚性，但可能会带来系统噪音。

以上提到的 8 个参数中 KP、KE、KC 在根据系统需求选定后都不需要再调整。因此主要整定的参数就是 KF、KD、KI、KV、KL 以及 KK。

5.2 使用 M Servo Suite 进行在线参数自动整定

通过上位机软件的简单操作，仅仅需要几分钟时间，M2 伺服系统即可完成对负载的动态反馈做出即时响应，并在线自动优化增益整定参数，实现系统的自动整定功能，从而大大节约了设备的调试时间，简化了调试过程。

注意：可以带负载进行在线自动整定。

5.2.1 第一步：选择电机

在进行在线参数自动整定之前，需确保驱动器中设定的电机是当前正在使用的电机。

操作 A：在 M Servo Suite 的“**配置**”页面----“**电机信息**”栏点击“**配置**”按钮（图 6-2）

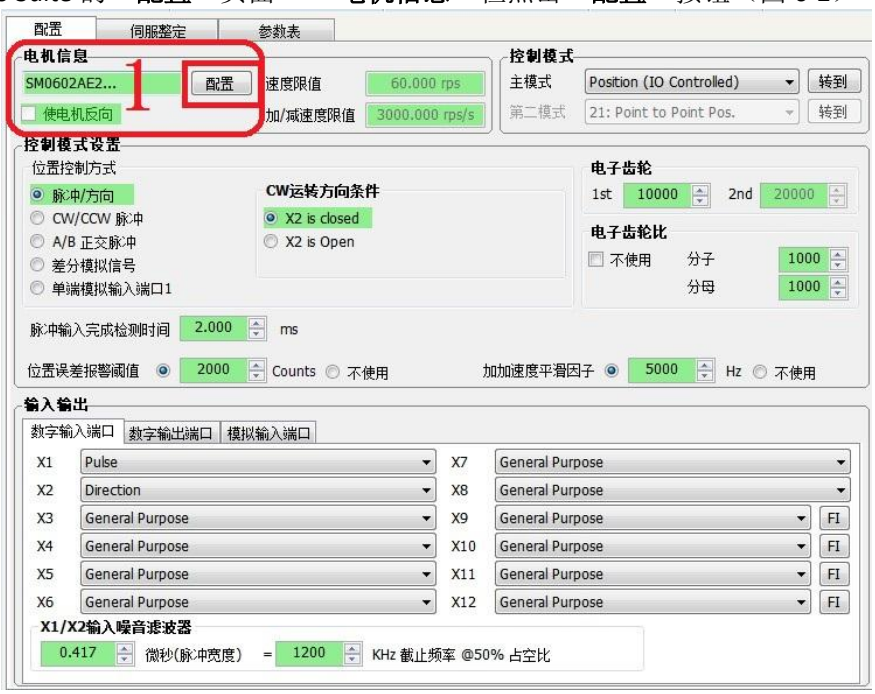


图 6-2

操作 B：在弹出的窗口中的电机列表下拉选项里面选择当前所使用的电机型号（图 6-3），然后点击确定。

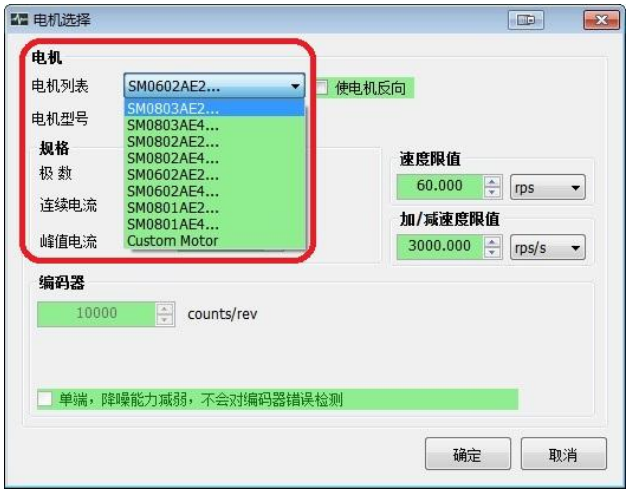


图 6-3

注意：参考 M2 用户手册中的 2.3 电机型号介绍章节，可帮助你确定所使用的型号。

5.2.2 第二步：软限位功能

软限位功能是指通过软件的方式设定虚拟极限开关的位置，可以保证在执行自动整定时，电机的运转范围在设定的软限位之间。从而保护机械设备。

注意：软限位功能重新上电后将失效。

用户可以选择在 M Servo Suite 的“**伺服整定**”页面----“**限位**”选项设定软限位功能，如不需要设定软件限位功能，则直接点击“**清除限位**”按钮，再进入“**自动整定**”页面开始自动整定（图 6-4）。

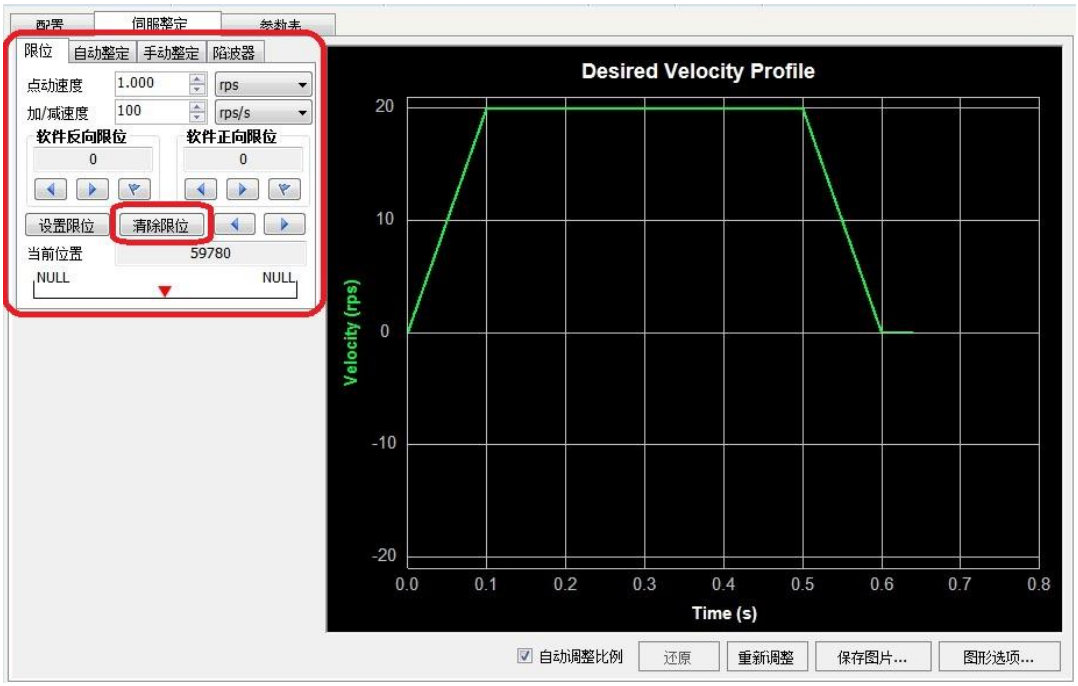


图 6-4

5.2.3 如何设定软件限位

如图 6-5 所示：

A. 在设定软限位时，设定点动的速度和点动的加/减速度

B. 为设定电机反向限位（电机逆时针旋转）

C. 为设定电机正向限位（电机顺时针旋转）

D. 为确定或者清除 B 步和 C 步所设定的限位。



图 6-5

设定软件限位的步骤

步骤	操作	图示
1	在电机使能后，设定软件反向限位 点击◀，电机反转；点击▶，电机正转。 到了所希望的方向限位后，点击▼，确定限位。	
2	设定软件正向限位 操作同上步	
3	确定所设定的限位 点击“设置限位”，确定软限位。 **注意：“软件正向限位”的设定值需大于“软件反向限位”设定值。否则软件将显示错误对话框。**	
4	设定好后。	

5.2.4 第三步：执行自动整定

在自动整定界面（图 6-6），按照如下步骤设定之后，即可执行自动整定。

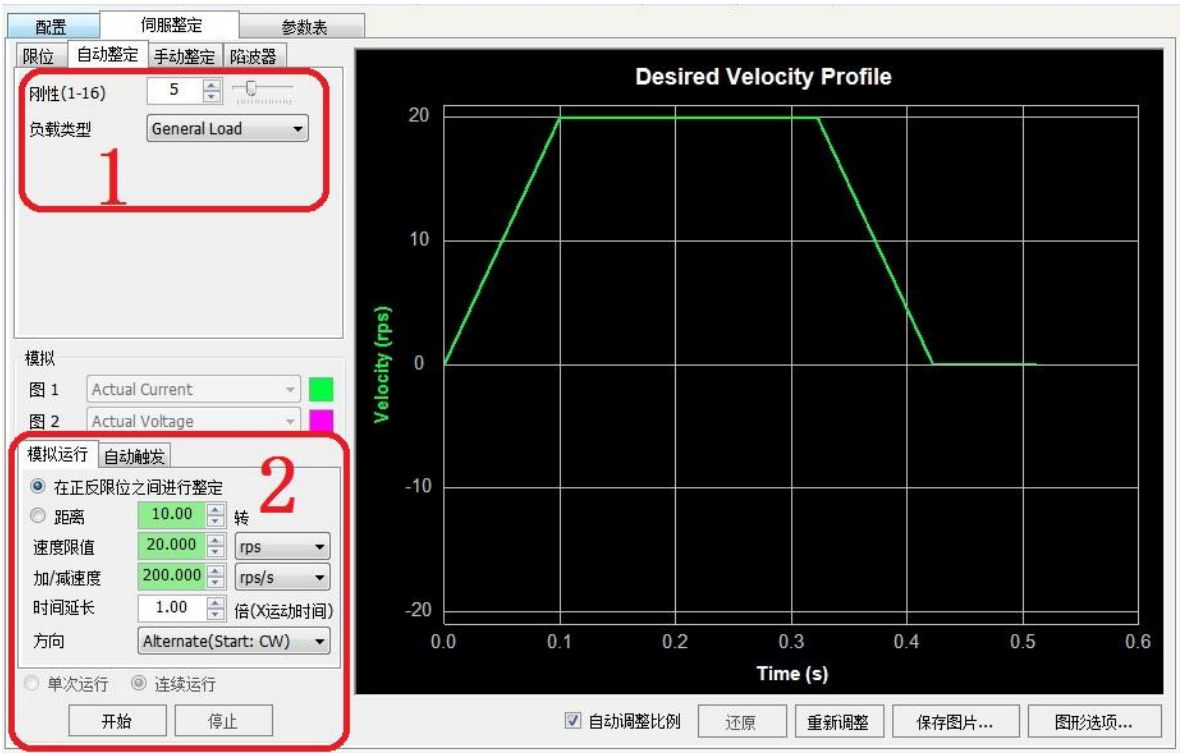
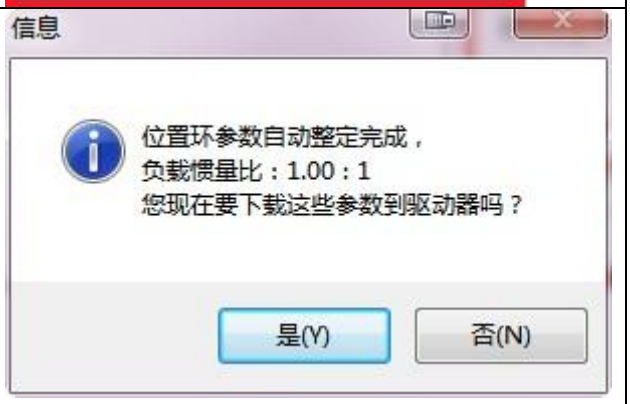


图 6-6 自动整定

操作步骤：

1	**设定刚性及负载类型** 设定需要的刚性及负载类型	
2	设定自动整定的距离、速度、以及加减速。 **注意：** **1) 如果已经设定了软限位，则勾选“在正反限位之间进行整定”** 2) **如果不需设定软限位，勾选“距离”，然后输入电机旋转的圈数（请到“限位”界面，确保清除了软限位的设定。）**	
3	点击开始，自动整定将开始执行	
4	自动整定结束，提示是否下载参数到驱动器中。	

注意：在自动整定过程中，可能由于部分参数过大，电机及所连接的机械负载会抖动并发出响声。请不必担心，参数会自动修正。

到此参数自动整定已经完成，客户可以根据实际情况，进行手动整定，完善增益参数，得到最佳的控制效果。

5.3 手动整定

用户根据机械的状态，手动修改下列参数来调整伺服增益，以进一步提高响应性：

- 全局比例系数（KP）
- 位置环比例系数（KF）
- 微分环节（KD）
- 阻尼系数（KV）
- 积分环节（KI）
- 跟随因子（KL）
- 加速度前馈系数（KK）
- 微分滤波系数（KE）
- 控制器输出滤波系数（KC）等

以上提到的 8 个参数中全局比例系数（KP）、微分滤波系数（KE）、控制器输出滤波系数（KC）在根据系统需求选定后都不需要再调整。因此主要整定的参数就是位置环比例系数（KF）、微分环节（KD）、阻尼系数（KV）、积分环节（KI）、跟随因子（KL）以及加速度前馈系数（KK）。本章节主要介绍上述各参数的作用及调整方法。

5.3.1 位置环比例增益（KF）

该参数对位置误差再进行一次比例系数增益，可提升系统的刚性及减小系统误差。PID 控制中，最简单的一相就是比例增益，也称为 P 增益。驱动器作用与电机上的电流与位置误差成正比。例如，如果电机没有运转，电机轴被手或是其他力转动，驱动器将增大电机电流直至电机返回到指定的目标位置。电机偏离指定位置越远，输出力矩越大。比例项（也称比例增益）根据输入的位置误差决定输出多大的力矩。通常来说，越大的惯量负载或是摩擦负载需要越大的力矩，因此需要一个越大的位置环比例增益。

当位置误差偏小时，将会导致系统响应不够快，位置误差减小趋势慢。如下图 6-7 所示，系统在加速过程、匀速过程、减速过程中，位置误差都很难恢复到 0 附近，但位置误差数值始终比较稳定。（紫色为实际速度曲线，绿色为位置误差曲线）

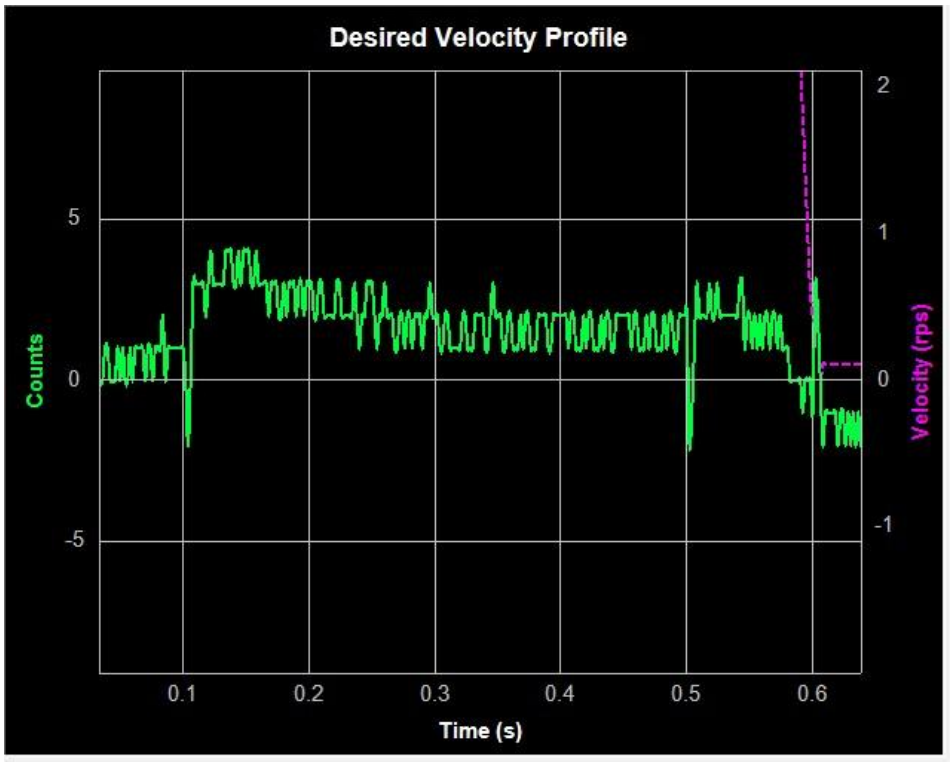


图 6-7 位置环比例增益（KF）偏小.KF = 2000

当位置环比例增益（KF）合适时，位置误差在加速过程、加速过程迅速趋近零，并在±1 脉冲之间振荡。如（图 6-8）所示

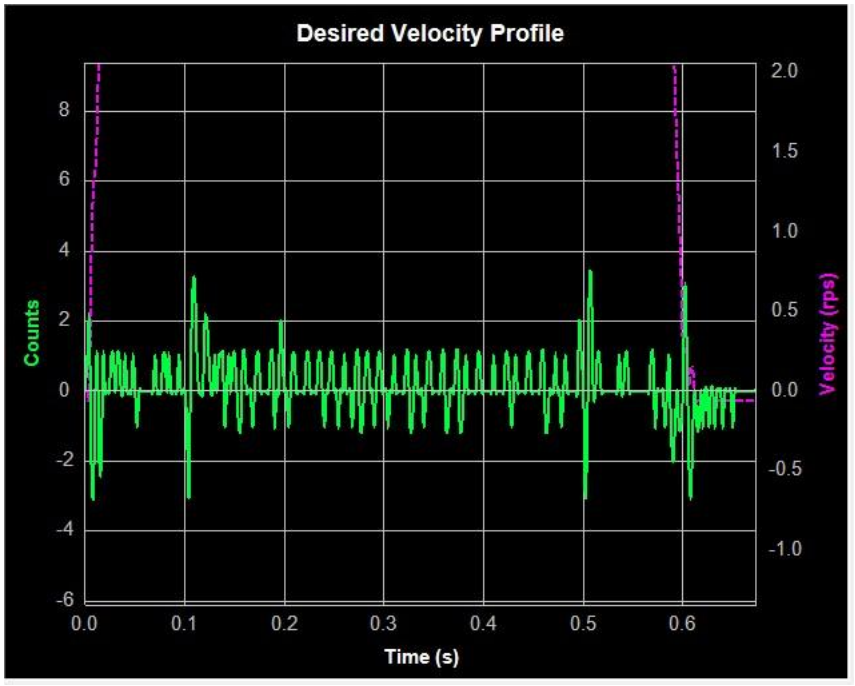


图 6-8 位置环比例增益（KF）合适,KF = 16000

5.3.2 积分增益(KI)

在比例增益控制下，位置误差可能无法恢复到零，或者需要很长时间恢复到零。积分增益将所有的误差累加并和比例增益一起作用，增大积分增益(KI)可以提高伺服系统的响应及应答性，并减小跟随误差。

当积分增益(KI)偏小时，系统响应会变慢，跟随性较差。如下图（图 6-9）所示，运动加减速过程中位置误差不能恢复到 0，但在匀速过程及停止能够较快恢复到零。

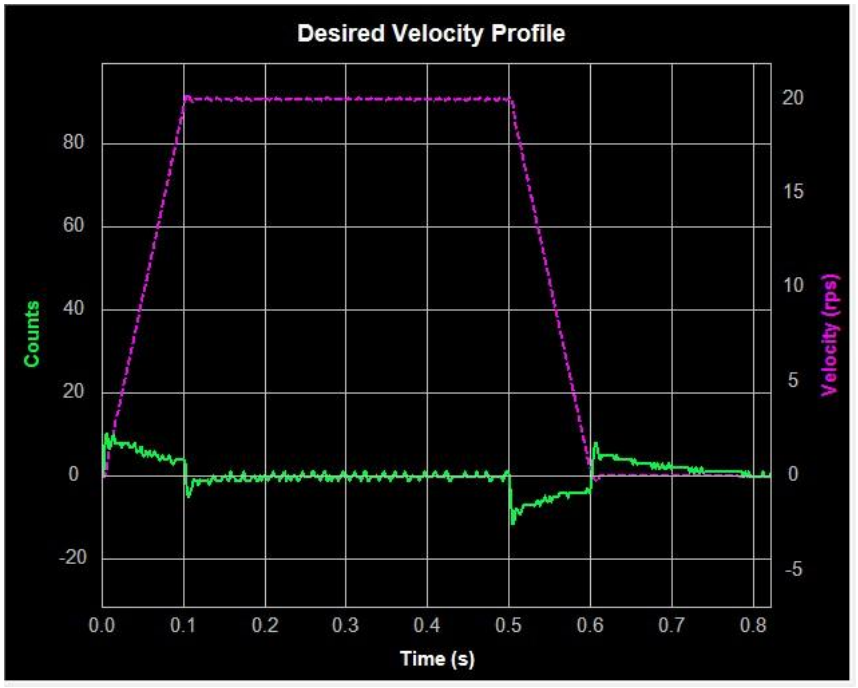


图 6-9 积分增益(KI) 偏小 KI = 50

增大积分增益(KI)可以提高伺服系统的响应及应答性，并减小跟随误差，如图 6-10、11-11 所示，可以看出，当积分增益(KI)逐渐增大，位置误差趋近于零的时间越短。

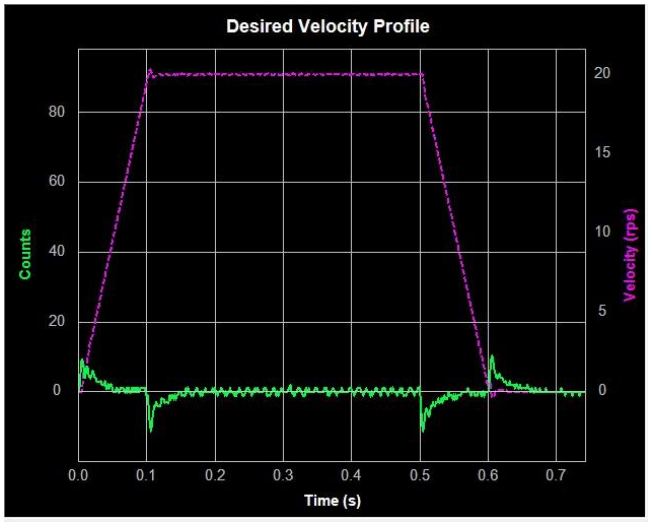


图 6-10 积分增益(KI)= 200

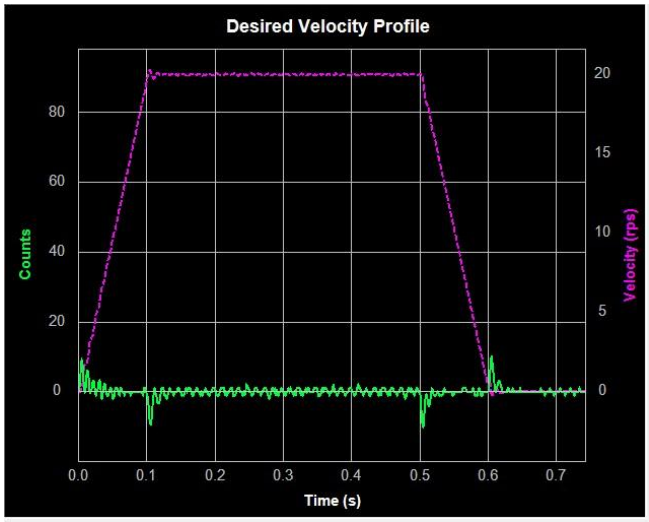


图 6-11 积分增益(KI) = 500

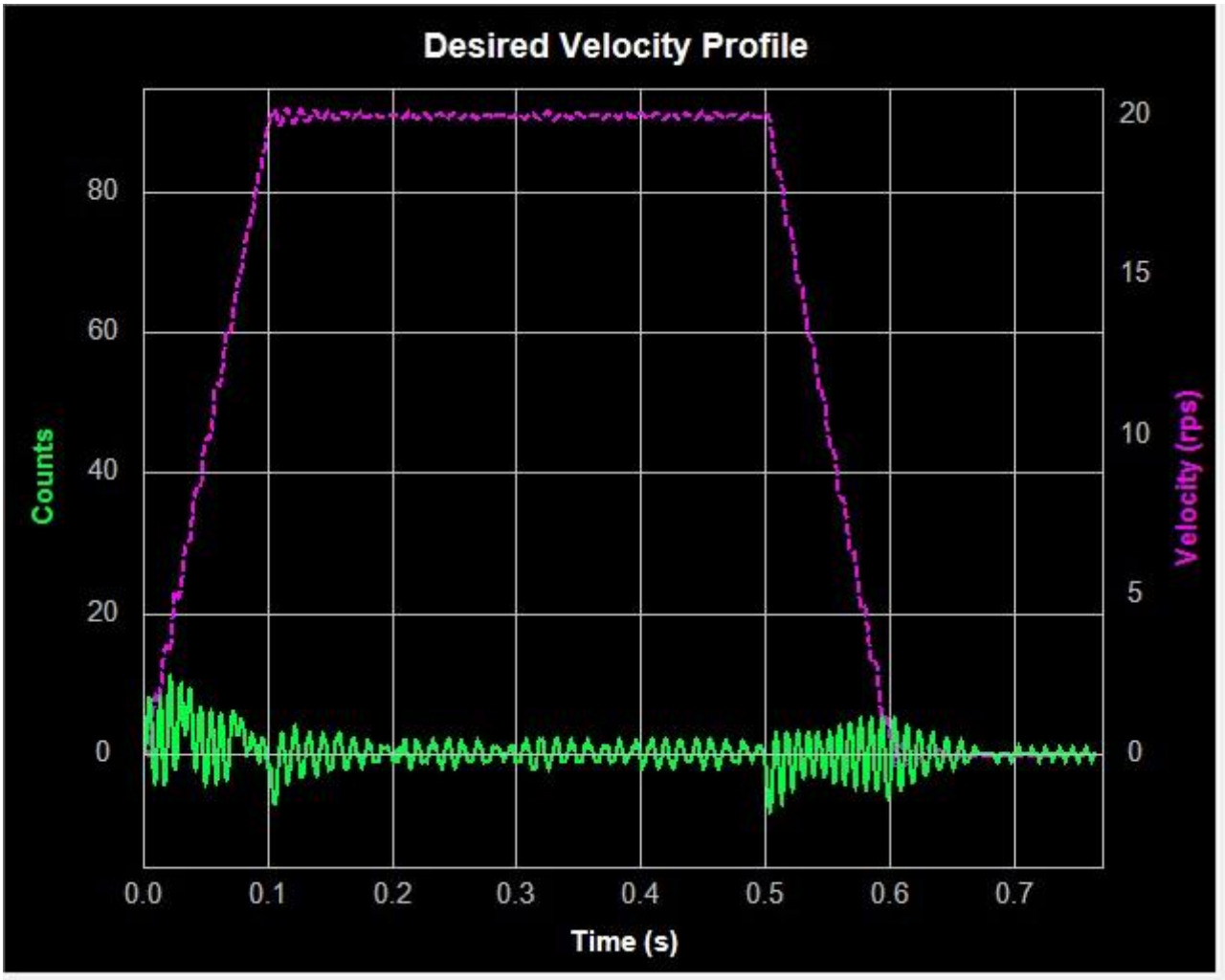


图 6-12 积分增益(KI) 过大 KI = 1000

5.3.3 阻尼增益（KV）

随着电机负载惯量的增加,伺服系统需要更多的阻尼来保证良好的运行效果。

当阻尼增益（KV）偏小时，将会导致系统在匀速运动过程或停止下来时阻尼不够，运动过程中的位置误差和速度误差都波动较大。如下图 6-13 所示，在运动过程中出现了位置误差和速度误差的较大波动，并呈现出振荡放大的效果。（紫色为实际速度曲线，绿色为位置误差曲线）

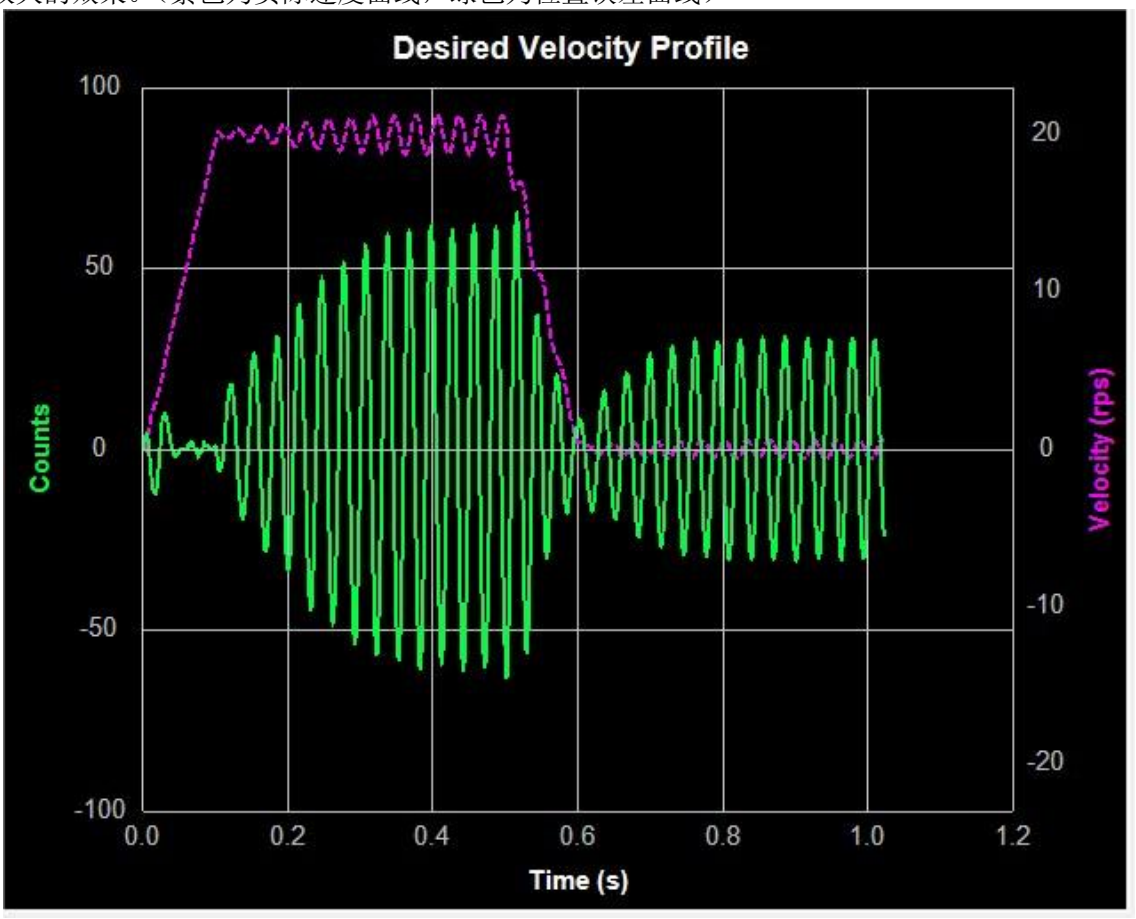


图 6-13 阻尼增益（KV）太小（KV = 5000）

逐渐增大阻尼增益（KV），匀速运动过程及运动停止时速度误差及位置误差都将减小，并趋于零（图 6-14）。

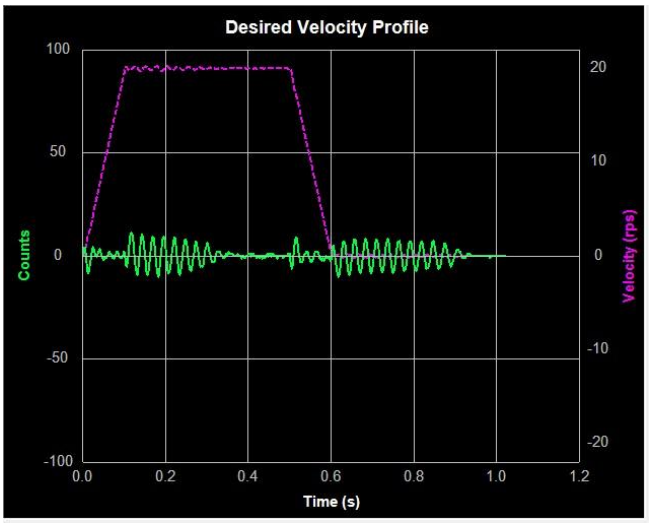


图 6-14 阻尼增益(KV)= 10000

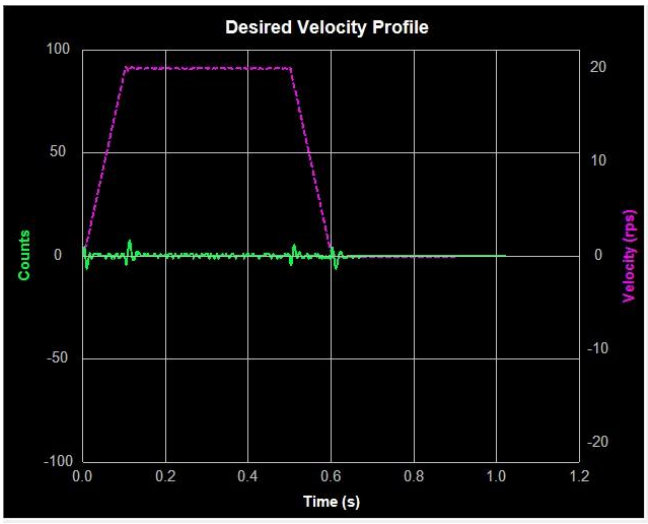


图 6-15 阻尼增益(KV) = 16000

当阻尼增益（KV）偏大时，系统阻尼效果过强，在加减速过程中容易引起系统振动和噪声。如下图 6-16 黄色标记所示：

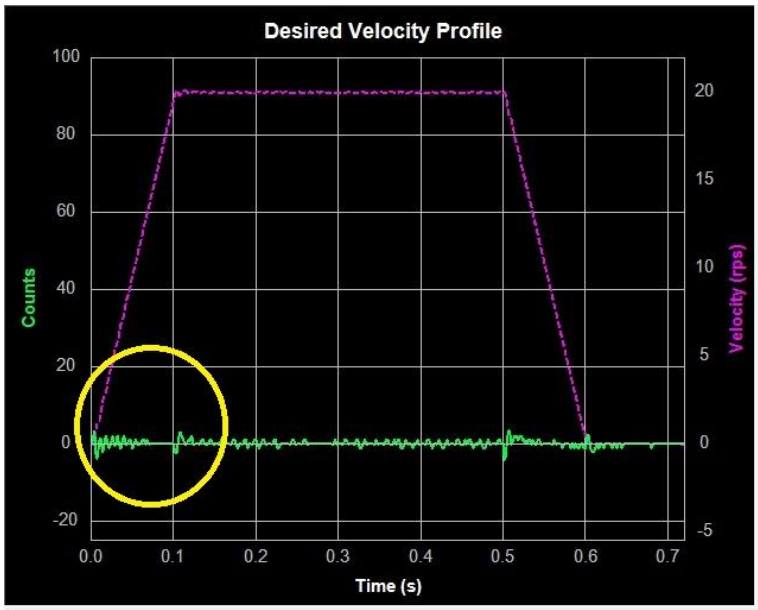


图 6-16 阻尼增益（KV）偏大 32000

5.3.4 微分增益(KD)

纯 PI 控制往往会对一个微小误差响应过度，从而产生一个更大的误差且变得不稳定。如果可以预先得知电机的动作，就避免这一情况。

比如说开车进入一个车库，人们不会完全进入车库之后才踩刹车，事实上，人们会更具目标距离不断减小而减慢车速。

驱动器可以分析位置误差的变化率来是实现超前输出。例如，如果电机有位置误差，但是该误差的变化率在降低，那么输出力矩也会相应的减小，从而避免响应过度。

当微分增益(KD)偏小时，系统抑制振动能力不足，将会在加/减速过程、匀速过程及停止后都产生明显的振荡，并且呈现一种振荡减小的趋势，并最终稳定下来，如下图 6-17 所示：

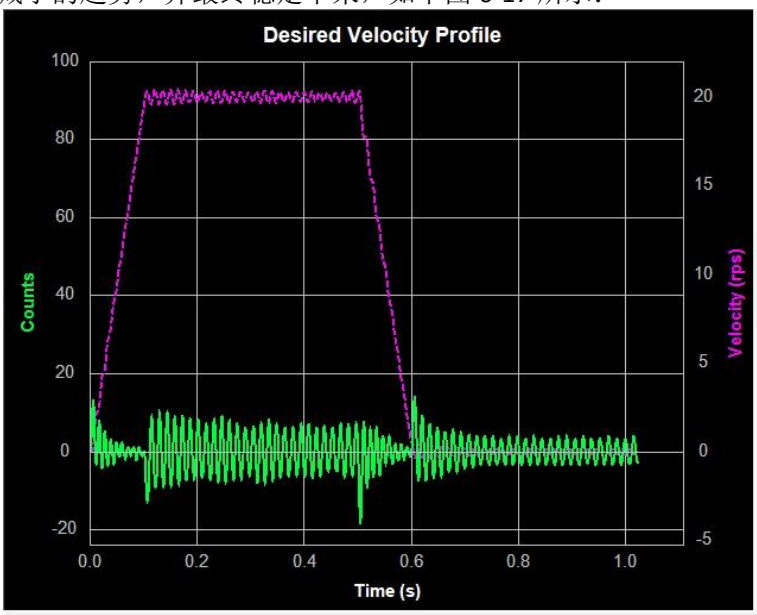


图 6-17 微分增益(KD)偏小，系统振荡，但幅度趋于减小（KD=3000）

当微分增益(KD)增大时，系统抑制振动能力明显加强，并快速趋于稳定，如图 6-18，图 6-19 所示。

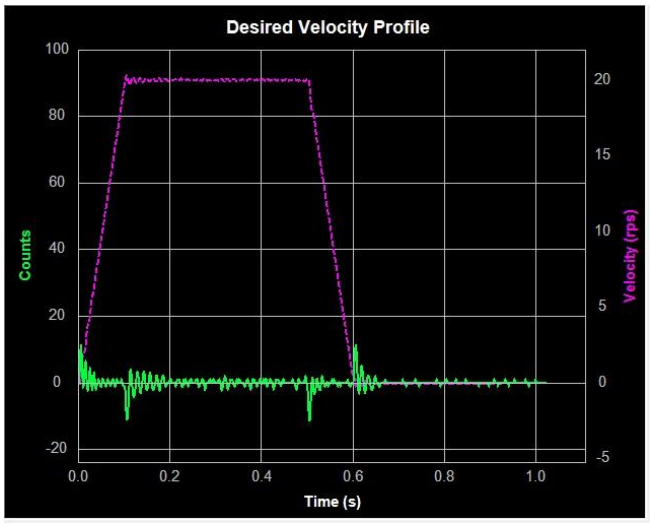


图 6-18 微分增益(KD)= 4000

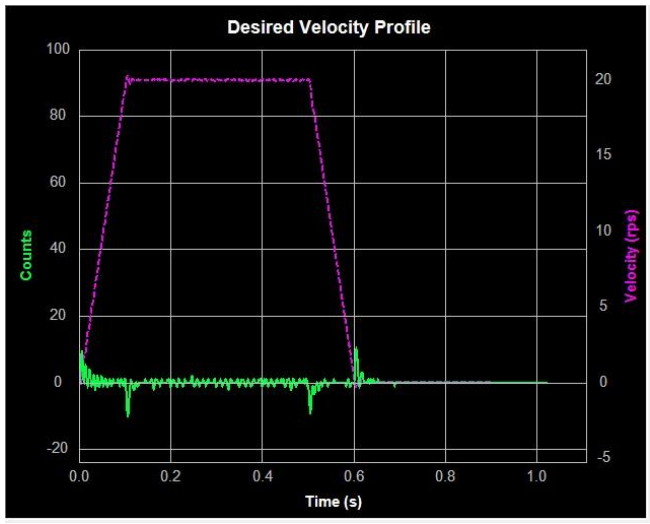


图 6-19 微分增益(KD) = 7000

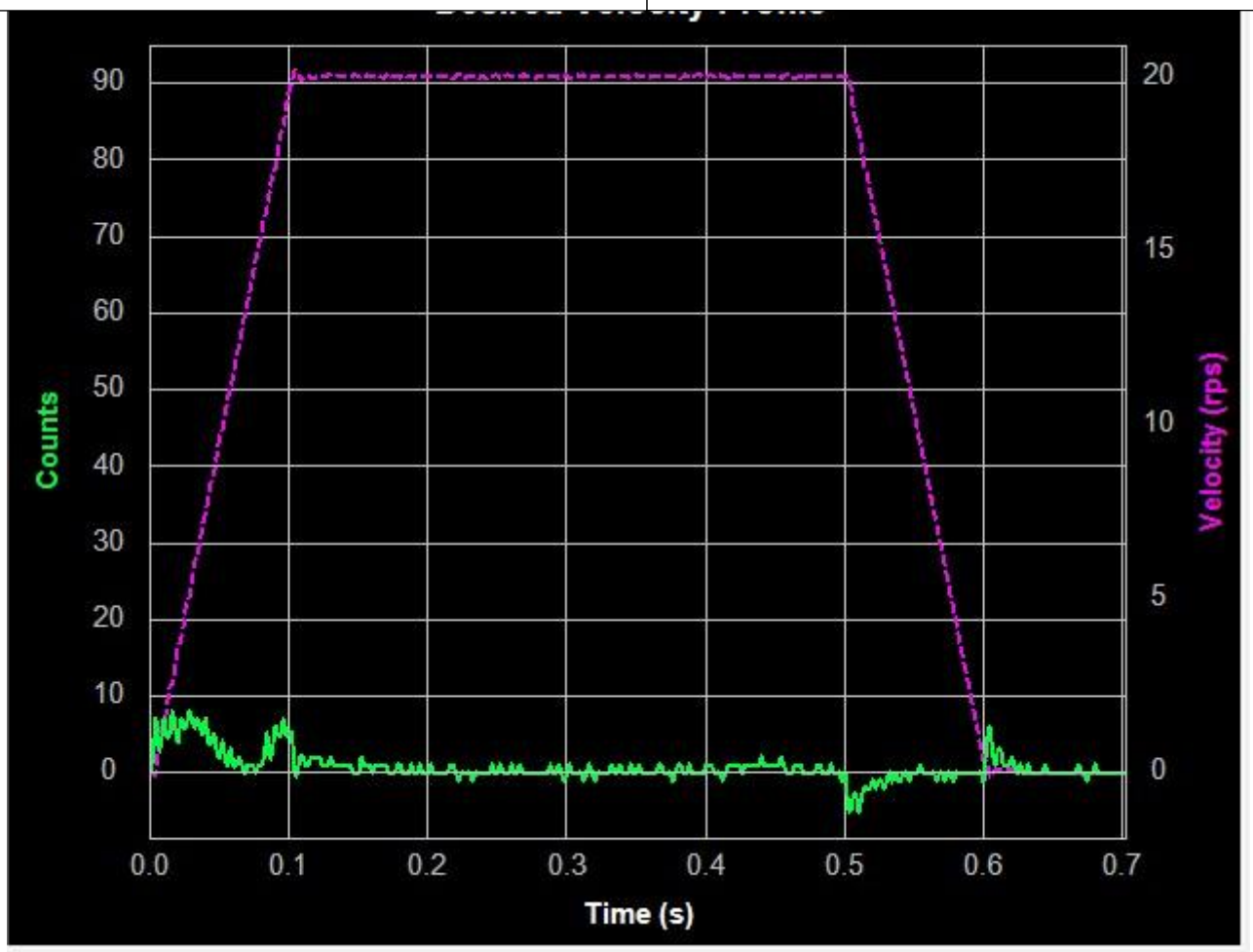


图 6-18 微分增益(KD)= 15000，系统极易受外部扰动，并引起振动，并发出噪音。

5.3.5 前馈增益(KK)

前馈增益(KK)为加速度前馈，较大的负载常具有较大的惯量，较大的惯量可通过预估系统所需力矩更为容易的得到控制。加速度前馈增益根据所需的力矩得出一个加速度值来实现前馈控制，消除跟随误差。

前馈增益(KK)偏小时，前馈输出不够，系统在加减速过程中的动态性能将会变差，表现在加减速过程中位置误差变大且不容易恢复到 0，系统稳定时间变长。如下图 6-21 所示：

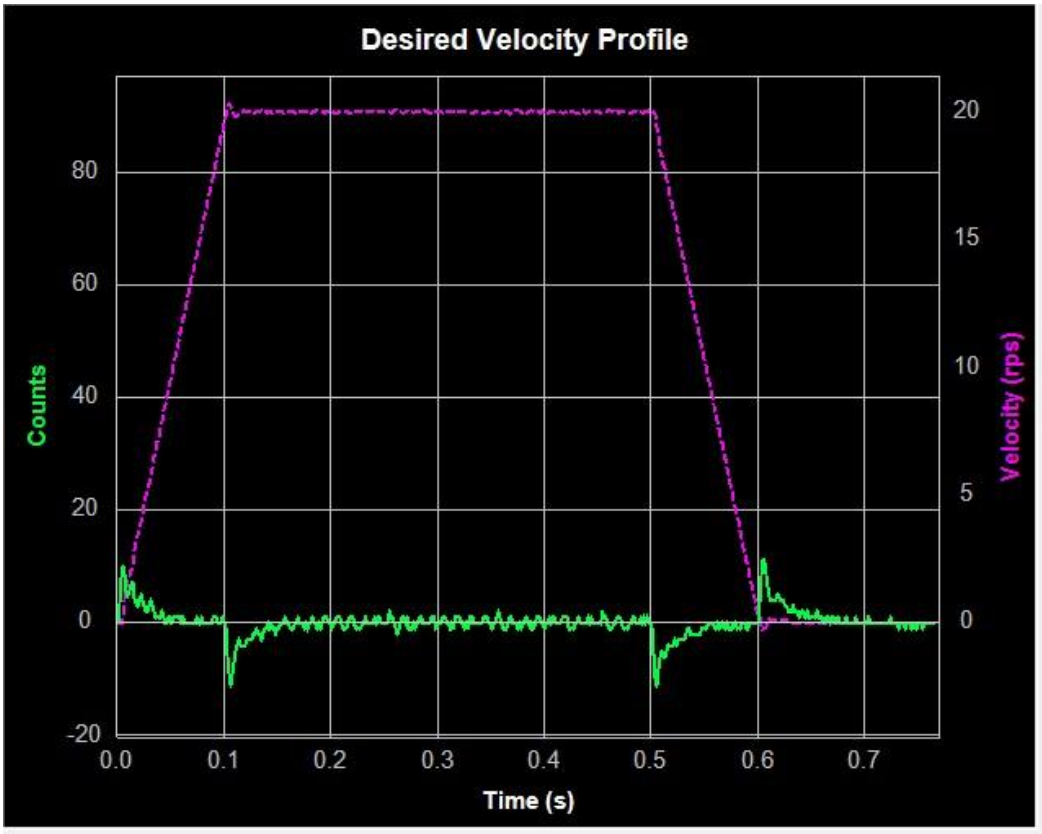


图 6-21 前馈增益(KK) 加速过程中跟随性较差（KK = 2000）

增大前馈增益(KK)，在负载惯量较大的机械系统下，系统在加减速过程中的动态性能将变好，加速过程中的位置误差也会快速的趋近与零，提高了系统追随性和响应性（图 6-22、图 6-23）。

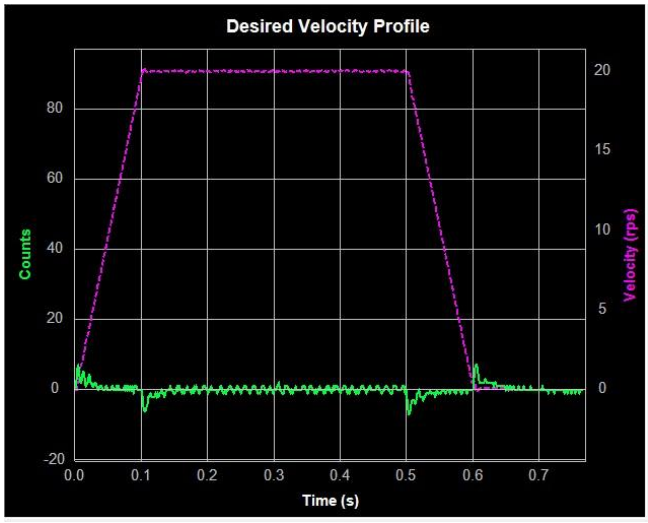


图 6-22 前馈增益(KK)= 4000

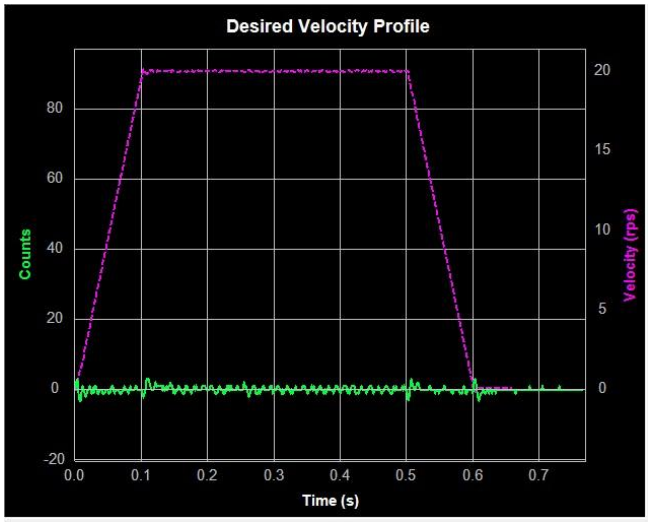


图 6-23 前馈增益(KK) = 11000

如下图 6-24 所示：

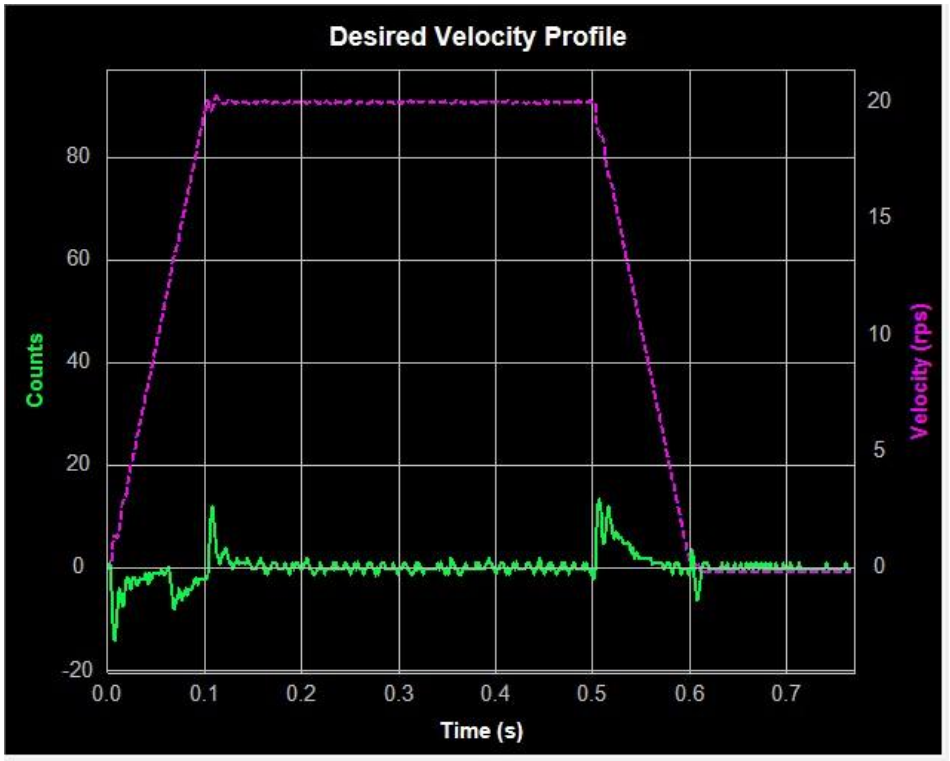


图 6-24 前馈增益(KK)，位置误差曲线反向（KK = 19000）

5.3.6 跟随因子（KL）

跟随因子为值为零时，伺服系统具有很好的刚性和动态追随性能，十分适合于轨迹控制，但可能会增加系统运行的噪音。对于点到点定位控制应用中，可以适当增加跟随因子，用以减小系统运行的噪音，消除过冲，但同时会降低运动过程中的动态跟随性。如图 6-25，图 6-26 所示（绿色为实际速度曲线，紫色为位置误差曲线）

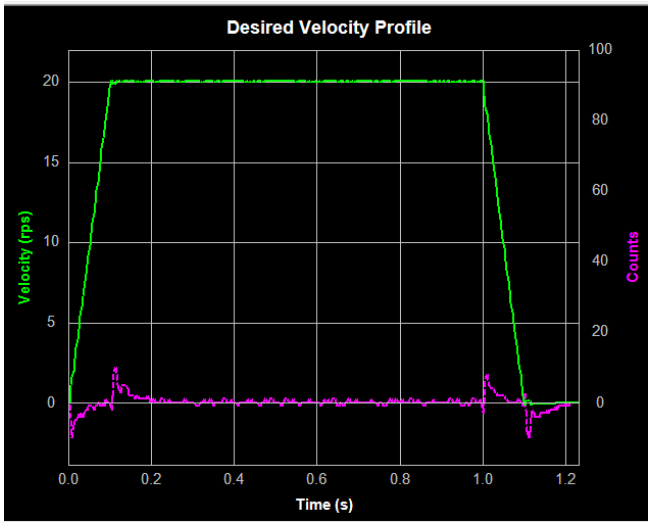


图 6-25 跟随因子(KL)= 0

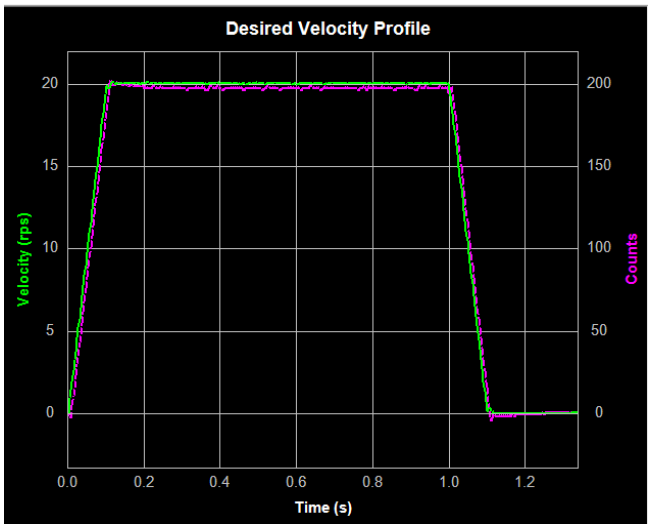
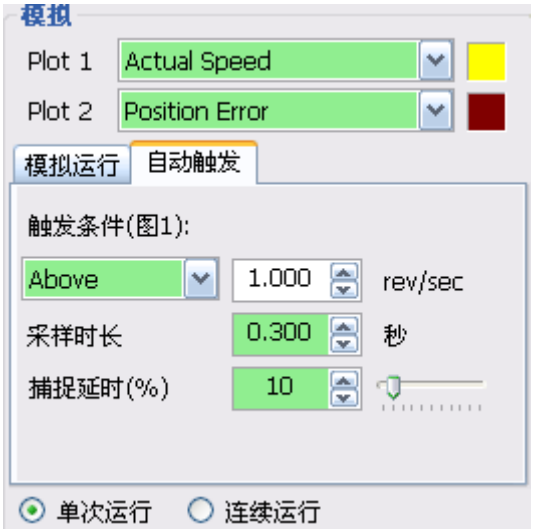


图 6-26 跟随因子(KL) = 15000

5.4 使用自动触发观测

在一些使用外部控制器来实现运动的场合，例如使用脉冲方向输入的位置控制模式，自动触发可用来获取观测数据并显示运动曲线。该采样观测技术不同于试运行的地方在于它并不是在运动一开始就触发观测，因为驱动器并不知道运动实际开始的时刻（使用的是外部控制器），自动触发等待预先设定好的触发条件来确定开始采样观测运动数据。

当使用自动触发观测时，首要的工作就是设置触发的条件。首先通过在 Plot 1 列出的下拉列表项中选择期望的观测触发项。自动触发器将监测该触发项，Plot 2 中的观测选项不被触发器监测。



在自动触发栏中可设置所选的观测触发项的触发条件和触发阈值。示例中，采样观测将在实际速度大于 1rps 的时候触发，采样观测时间持续 0.3 秒，同时也将采样触发开始前 10%时间段的数据。捕捉延时允许观测满足触发条件之前的数据，这样可观测到一个完整的运动曲线。

当改变观测触发项时，触发条件也会跟着改变。

例如，当选择 Position error 作为触发项时，触发条件就改为误差计数。

单次运行：点击开始按钮后，伺服驱动开始持续的收集数据，它将不停的检查数据是否满足触发条件，同时，M Servo Suite 开始监测伺服的状态以探测触发采样是否完成。一旦完成，观测窗口就会绘制观测得到的数据曲线。

连续运行：点击开始按钮后，会在每次采样捕捉条件触发后重复采样直到点击了停止按钮。在连续采样观测中，增益参数可以随时更改并自动更新到驱动器中，这将有利于更方便的调节增益参数并加快整定过程。

6 第 3 步：Q 编程

Q 编程是一款适用于鸣志可编程式步进、步进伺服及伺服产品的单轴运动控制软件。Q 编

程允许用户使用 SCL-Q 指令集为-Q 驱动器创建功能复杂可独立运行的程序，支持的指令包括运行控制、I/O、驱动器配置、与状态查询、数学运算、寄存器操作和多任务处理。具有如下特点：

- 运动控制（例如 FL、FP、SH 等）
- 执行驻留程序
- 多任务处理（请参阅 MT 指令）
- 条件判断（例如 OI、TI 等指令）
- 数学运算（例如 R+、R-、R*、R/、R|、R&等指令）
- 寄存器操作（例如 RX、RM 等指令）

1 个 Q 程序支持多达 10 个程序段，每个程序段最多可以编写 62 条指令。

如果勾中“上电时自动执行 Q 程序”，则驱动器在每次上电时会自动从 Q 程序的第 1 段开始运行，否则需要向驱动器发送通讯控制指令才能执行 Q 程序。

程序段之间的跳转是通过 QX 指令实现的。

第 10 程序段为中断处理程序段（例如通过外部 IO 触发的中断，请参阅 OI 指令），不推荐在第 10 程序段中编写非中断处理程序。

6.1 Q 编程界面

Q 编程的界面如下图所示：



打开 Q 程序：打开存储在磁盘上的 Q 程序文件。

保存 Q 程序：将 Q 程序保存到磁盘上。

打印：打印 Q 程序。

从驱动器上传：从驱动器上传 Q 程序。

下载到驱动器：将 Q 程序下载到驱动器。

清除 Q 程序：清除软件和驱动器中的 Q 程序。

执行：执行 Q 程序。

停止：停止执行 Q 程序。

设置密码：设置 Q 程序密码，存在密码的 Q 程序如果没有输入正确密码将无法上传。

注意：Q 程序密码一旦忘记便无法找回，所以千万要记得您所设置的 Q 程序密码。如果需要强制解锁，可输入原始密码“1234”，但驱动器内所有 Q 程序将被擦除为空。

上时自动执行 Q 程序：勾选时驱动器在每次上电时会自动从 Q 程序的第 1 段开始运行。

6.2 当前程序段



当前程序段包含以下按钮：

打开：打开存储在磁盘上的 Q 程序段文件。

保存：将当前 Q 程序段保存到磁盘上。

打印：打印当前 Q 程序段。

上传：从驱动器上传 Q 程序段。

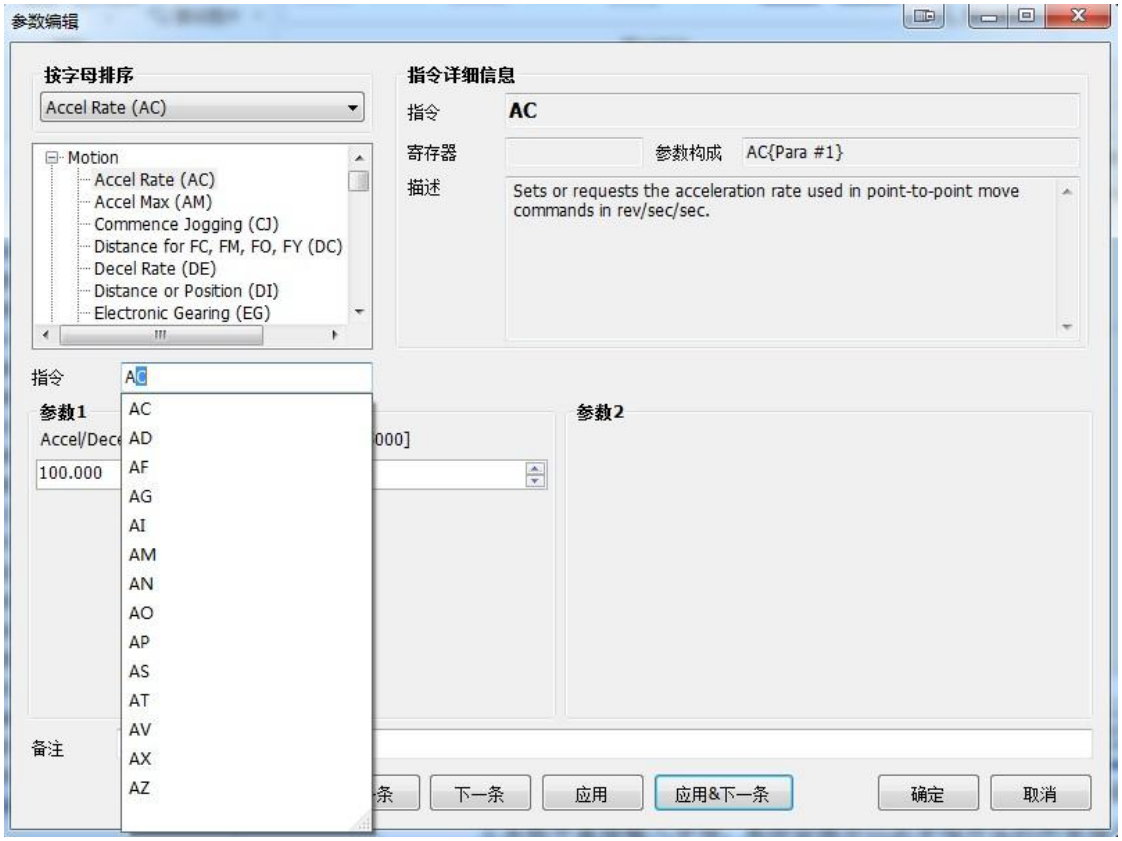
下载：将当前 Q 程序段下载到驱动器。

执行：执行当前 Q 程序段。

清除：清除当前 Q 程序段。

6.3 参数编辑

点击程序编辑界面中程序段 Cmd（命令）列的任意一个单元格时，单元格的右边会出现一个按钮，点击此按钮时会弹出参数编辑界面，如下图所示：



界面的左边有一棵显示所有指令的树，选择其中的节点会选中此指令。另外您也可以在指令

文本框中直接输入字母，系统会提示以此字母开头的所有指令。当指令选择完成后，系统会显示与此指令相关的参数 1 和参数 2。有的指令只需要 1 个参数，此时参数 2 中的内容就是空的。根据界面的提示就可以完成参数的编辑。您还可以对这条指令添加一段说明性的文本，即“备注”。

注意：备注内容可以保存到 Q 程序文件，但无法下载到驱动器。

以下是界面中各个按钮的功能说明：

插入：在 Q 编程界面的当前行中插入一条空指令。

上一条：将 Q 编程界面的当前行往前移动一行。

下一条：将 Q 编程界面的当前行往后移动一行。

应用：应用当前的参数设置。

应用&下一条：应用当前的参数设置并将 Q 编程界面的当前行往后移动一行。

确定：应用当前的参数设置并退出参数编辑界面。

取消：退出参数编辑界面而不应用当前的参数设置。

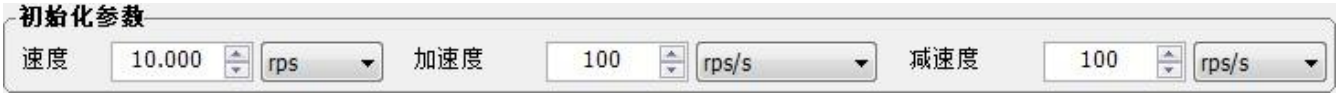
7 运动仿真

运动仿真页面提供了点到点运动、点动、回原点等实际运动的仿真模式，无需上位机发送控制信号。



7.1 初始化参数

设置运动仿真中的速度、加速度、减速度参数。



7.2 点对点运动

点对点运动

执行绝对位置运动及相对位置运动。绝对位置运动是以编码器 0 点位置为参考。

运动到传感器

设定想要运行到的传感器、方向及停止条件，点击“运动到”按钮开始执行。



7.3 点动

设置点动速度，加/减速度后，鼠标按住 CW 点动或者 CCW 点动后，电机将按照设定的运动速

度运动，直到松开鼠标。



7.4 回原点

回原点功能允许您设置回原点模式、传感器状态、找原点的速度以及加/减速度、回原点偏移量（步数）、Hard Stop 电流等参数。并可以在指令预览窗口看到上述所有动作生产的回原点指令。



7.4.1 回原点模式

回原点模式提供了三种方式：无传感器 Hard Stop 回原点、带传感器回原点、带传感器并寻找编码器 Index 信号的回原点。



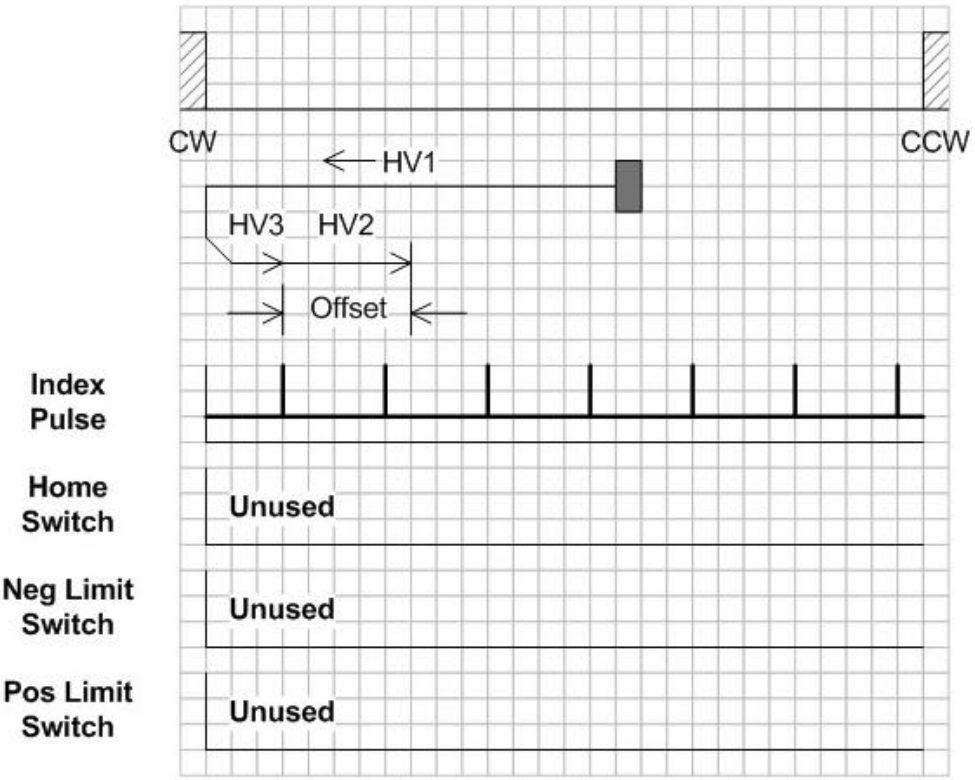
点击“*图示说明*”可以获取各回原点方式的详细说明。

7.4.1.1 无传感器 Hard Stop 回原点

无传感器回原点 即不需要机械机械部分安装原点传感器，采用撞硬限位的方式回原点。



需电机客户在设定***回原点的速度***、***回原点偏移量***、***HardStop 电流***、是***否搜寻 Index 信号***。回原点过程如下图所示，



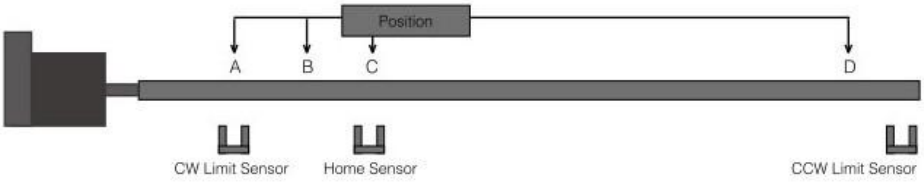
电机以 ***HV1 的速度***开始回原点，回原点的方向由***原点偏移量***决定（正数为电机顺时针旋转、负数为电机逆时针旋转），在碰到机械硬限位且电流大小达到 ***Hard Stop 电流***，电机停止运行。

A 如果设置***搜寻 Index 信号***，则反方向以 ***HV3 速度***运转寻找编码器的 Index 信号，找到 Index 信号后，以 ***HV2 速度***完成***回原点偏移量 Offset。***

B 如果无需***搜寻 Index 信号***，则反方向以 ***HV2 的速度***完成***回原点偏移量 Offset***。

7.4.1.2 带传感器的回原点

带传感器回原点过程如下图：



1) 当电机的负载位于 A 点时（CW 限位传感器已被触发）

- 电机以 ***HV1 速度***、***HA1/HL1 为加、减速度***寻找原点传感器 C。
- 一旦原点传感器 C 触发，电机继续向 CCW 方向以 ***HV2 速度***、***HA2/HL2 为加、减速度***运动脱离原点传感器 C，到达 ***HO*** 设定的距离。
- 其后，电机以 ***HV3 速度***、***HA3/HL3 为加、减速度***返回到原点传感器 C

2) 当电机的负载位于 B 点（位于 CW 限位和原点传感器 C 点之间）

- 电机以 ***HV1 速度***、***HA1/HL1 为加、减速度***向 CW 方向寻找 CW 限位传感器，
- CW 限位传感器触发后，电机停止
- 后面运动方式同 ***1) 电机负载位与 A 点***。

3) 当电机的负载处于 C 点（原点传感器已被触发）

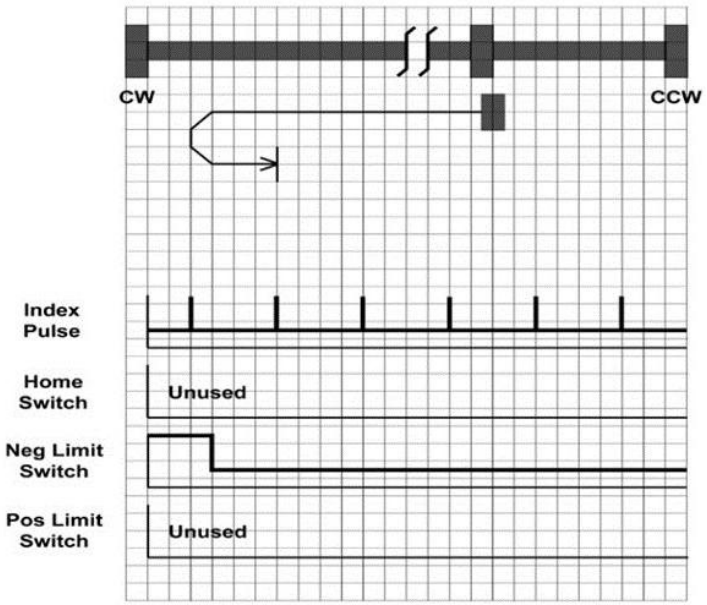
- 电机向 CCW 方向以 ***HV2 速度***、***HA2/HL2 为加、减速度***运动脱离原点传感器 C，到达 ***HO*** 设定的距离。
- 其后，电机以 ***HV3 速度***、***HA3/HL3 为加、减速度***返回到原点传感器 C

4) 当电机的负载位于 D 点（位于原点传感器 C 和 CCW 限位传感器之间）

- 电机以 ***HV1 速度***、***HA1/HL1 为加、减速度***沿 CW 方向寻找原点传感器 C
- 当原点传感器 C 触发后，后面运动同 3) 当电机的负载处于 C 点。

7.4.1.3 带传感器并寻找编码器 Index 信号的回原点

以寻找电机编码器的 Index 信号找原点。过程如下图



- 电机按照选择的 CW 运转
- 当碰到 CW 限位传感器停止运动
- 电机以 CCW 方向运转找第一个 Index 信号

或者是

- 电机按照选择的 CCW 运转
- 当碰到 CCW 限位传感器停止运动
- 电机以 CW 方向运转找第一个 Index 信号

7.4.2 指令预览

在回原点界面操作时，***“指令预览”***界面将显示出为实现当前设置的回原点模式所使用的 SCL 指令。如下图：



8 SCL 终端

SCL 终端也可以作为一个试运行工具，允许您使用 SCL 指令控制驱动器和电机。



在 SCL 终端中点击“脚本”按钮，软件会弹出“脚本”执行对话框。如下图所示：



在脚本中可以编写一系列 SCL 指令并运行。如果勾中“循环执行”，软件在执行脚本时会循环执行脚本，直到点击“暂停”按钮。

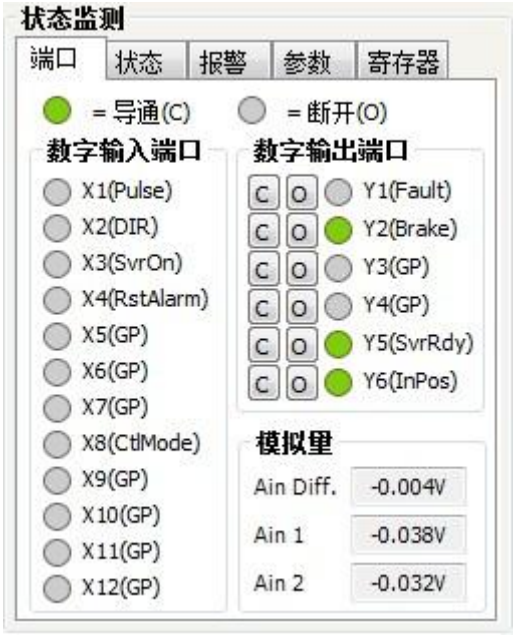


如果勾中“执行时停止监控”，软件在执行脚本时会停止后台监测（例如状态监测），执行将会更加有效、准确（软件延时大大降低）。

9 状态监测

可以监控 I/O 状态、驱动器状态、报警信息、驱动器参数、及寄存器监视。

I/O 状态



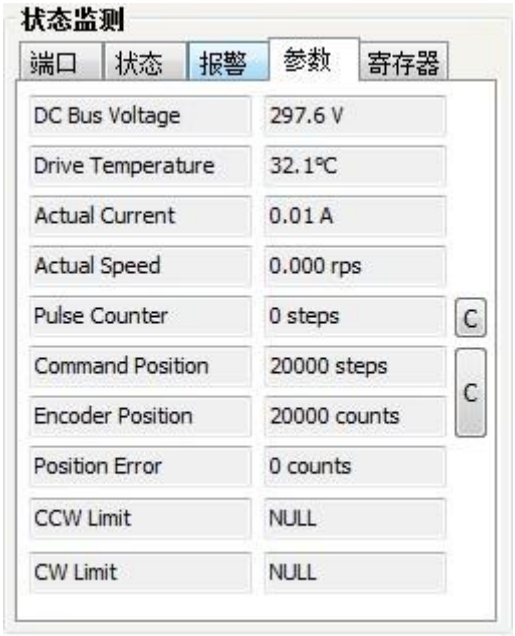
驱动器状态



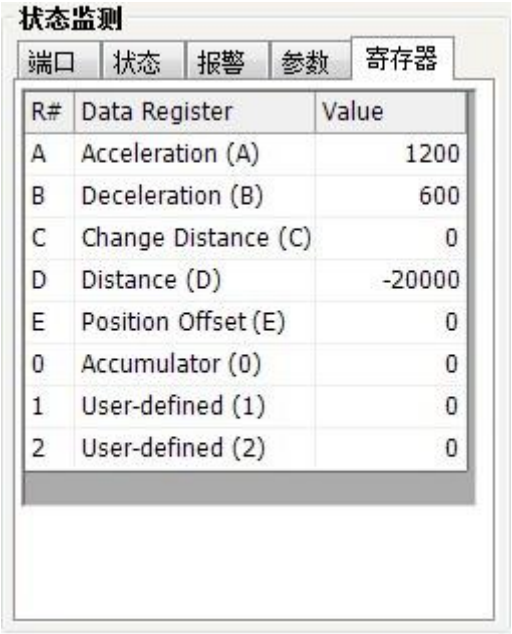
报警


状态监测
端口
报警
参数
寄存器
故障
位置误差超限
电压过低
驱动器过温
编码器信号错误
过压
Q 程序为空
内部电压报警
电源缺相
过流
STO
霍尔信号错误
速度超过限值
警告
CCW 限位
泄放失败
CW 限位
重载状态
保存失败
命令动作@未使能
通讯报警
电压警告

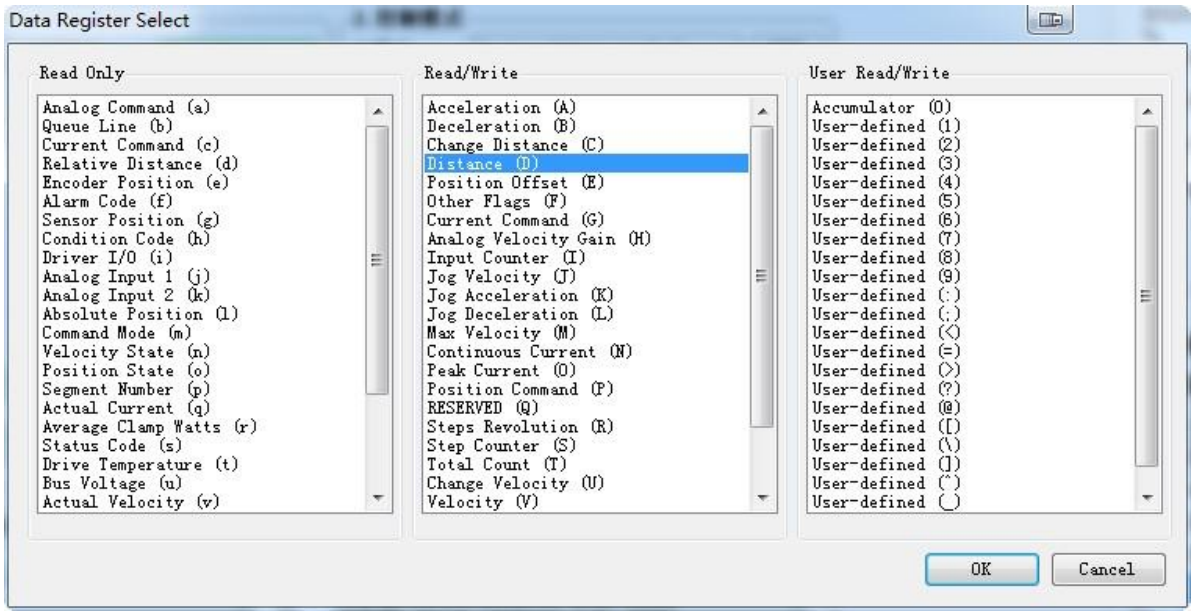
参数


状态监测
端口
状态
报警
参数
寄存器
DC Bus Voltage
297.6 V
Drive Temperature
32.1℃
Actual Current
0.01 A
Actual Speed
0.000 rps
Pulse Counter
0 steps
C
Command Position
20000 steps
C
Encoder Position
20000 counts
Position Error
0 counts
CCW Limit
NULL
CW Limit
NULL

寄存器



点击“Data Register”列中的行时会出现寄存器选择的按钮，点击此按钮软件弹出寄存器选择对话框框，如下图所示：



10 SCL 指令集

SCL 给用户一个简单的通过串行端口控制电机驱动器的方法。这样就不需要独立的运动控

制器来给驱动器提供脉冲/方向信号，同时也提供了一个简单的面向其它各种工业设备的应用途径，例如：具有有标准串行通讯端口的工业设备 PLC，工业 PC 或者 HMI 等。
有关详细的 SCL 指令及通讯协议，请点击查看。或者前往本公司网页下载 Host Command Reference 手册。

10.1 指令

上位机指令包括了两种类型：缓存指令和立即指令。缓存指令会以队列的形式缓存到启动器的指令缓存器中，然后以先进先出的顺序执行。立即指令则不会被缓存，当驱动器接收到以后会立即执行。

10.1.1 缓存指令

当上位机的指令成功发送到驱动器以后，缓存的指令会以先进先出的顺序在执行（特殊情况请参照 Q 驱动的多任务模式）。比如说当驱动器先接收到 FL（Feed to length）指令，然后再接收到了 SS（Send String）指令。SS 指令则会保存在驱动器的指令缓存器里，在驱动器执行完 FL 指令以后再执行 SS 指令。除了一些特殊指令像是 PS（Pause）和 CT(Continue), 这两条指令可以暂停执行缓存中的指令并等到合适时机再继续执行，所有驱动器装载缓存器里的指令后，他们会安照顺序执行而不需要任何的其他的指令输入。

Q 程序

Q 类型驱动器支持 Q 编程，Q 程序可以通过 Q programmer 软件来编辑。在 Q 程序里，只允许使用缓存指令。

Q 系类驱动器多任务模式

多任务模式只能在 Q 模式下使用，它可以改变同时只能执行一条缓存指令的规则。当多任务模式使用时，Q 驱动器可以先使用 FL，FP，CJ，FS，等指令让电机运动。而不用等待运动完成，还能接收和执行其他上位机的指令。

10.1.2 立即指令

立即指令会在驱动器收到以后立刻执行，在有需要的时候他可以与驱动器的缓存指令同时执行。比如说，BS 指令（Buffer Status）允许你在任何时候查看驱动器缓存器的存储状态，或者 IS 指令（input status）可以立即反馈驱动器输入口的状态。立即指令是允许驱动器在任何时候执行的。

我们建议客户使用为驱动器配置相对合理的 Ack/Nack 反馈方式。这样可能会对驱动器运行效率有微小的降低，但是他可以有效的保证驱动器能正确的接受和执行上位机发出的指令，减少通讯错误的概率。如果因为应用上的原因 Ack/Nack 不能使用的话，请确保驱动器的接收或发送

指令之接有大于 10ms 的延时来保证通讯质量。

立即指令主要应用于上位机接收驱动器的反馈信息，立即处理的方式使得状态监测变得更加高效。

10.2 使用指令

一个标准的指令是一串字符加回车符。而字符串的格式则是指令加指令相关参数的形式。回车代表着命令输入完毕。以下是一个基本的格式

YXXAB<cr>

在上述的格式中，“Y”代表着在 RS-485 通讯时的驱动器地址（只在 RS-485 通讯时使用）。”XX”代表上位机的指令，由两个大写字母来标准。“A”代表指令的第一个可用参数，“B”代表第二个可用参数。由于指令所需参数不同，A 和 B 不一定需要使用，而且他们的取值长度和范围也是根据具体情况来决定的。“<cr>”代表回车，预示着指令输入结束。而回车的输入方式则根据你的上位机控制方式来定。

一旦驱动器收到了回车，他会去判断上位机发出的指令是否正确，如果指令正确，他会立即被执行或者存储在驱动器的缓存寄存器内。如果使用 Ack/Nack 的应答方式，驱动器则会反馈一个应答到上位机，如果命令被立即执行，驱动器会反馈“%”，如果指令被缓存，驱动器会反馈“*”。

当 Ack/Nack 使用时，请务必在上位机收到反馈以后再发送下一条指令。如果 Ack/Nack 不使用时，请确保每条非运动指令间需要 10ms 时间间隔

如果驱动器在收到指令以后没有反应或者应答。请打开 Ack/Nack 通讯协议，在收到错误指令的时候，驱动器会反馈“？”。通常情况下 Nack 的回复也会带上错误编号。具体错误编号请参照附录错误表，或者 PR 指令的详解。

驱动器的给上位机的应答格式会与 SCL 指令相仿：

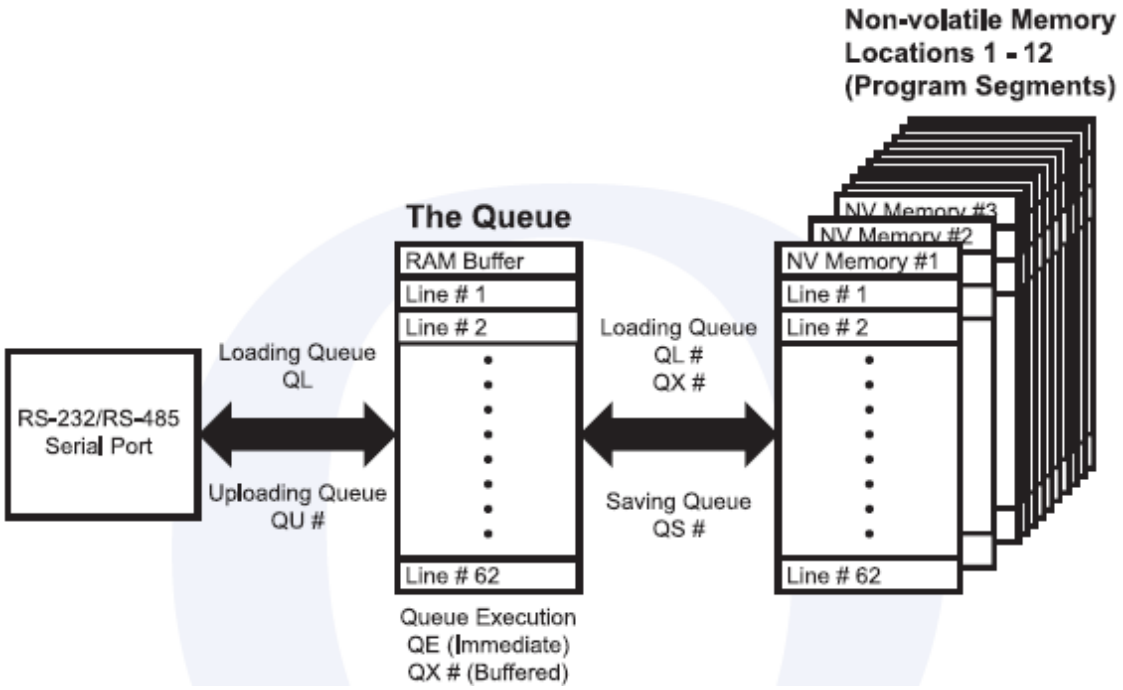
YXX=A<cr>

在上述的指令格式中，“Y”代表着在 RS-485 通讯时的驱动器地址（只在 RS-485 通讯时使用）。”XX”代表上位机的指令，由两个大写字母来标准。“A”代表指令的第一个可用参数是返回值，可能以 10 进制或者 16 进制的方式显示。 “<cr>”代表回车，预示着指令输入结束。而回车的输入方式则根据你的上位机控制方式来定。

10.2.1 Q 系列驱动里的 Q 程序

由于 Q 系列驱动器的指令储存功能，他可以使驱动器以程序驻留的模式运行。而这些指令的输入格式基本与直接命令输入格式相同“XXAB”。Q programmer 的软件是用来编辑以及上传或者下载 Q 程序到驱动器里的，他可以从鸣志官网 www.moons.com.cn 上免费下载。

下图介绍了上位机指令是如何发送指令对驱动器的指令缓存器队列，和 Q 程序存储器进行操作的。上传和下载 Q 程序缓存队列可以通过 QL 和 QU 的指令来执行。相似的存储在程序存储器中的 Q 程序也可以通过 QL 和 QX 来读取，而 QS 指令则是用于将 Q 程序保存到程序存储器。保存好的 Q 程序也可以通过 QE 和 QX 指令来执行。



Q programmer 软件可以用简化的方式来实现上图中的功能。

10.2.2 SCL Utility 软件

SCL Utility software 是一款非常简单易用软件，他可以帮助你快速的熟悉和了解这些 SCL 指令。SCL Utility software 可以在鸣志官网 www.moons.com.cn 免费下载。

在通过 SCL Utility software 给驱动器发送指令时，可以直接键盘输入指令，然后按回车键来发送指令到驱动器。（提示：所有的 SCL 指令都是以大写形式输入的，所以可以首先打开 CapsLock，方便大写字母输入）。按键盘的回车键会结束当前指令的输入，并把指令发送到对应的驱动器上。

请尝试以下指令输入。注意：<ENTER>代表回车输入，也代表指令输入结束并发送。

注意：我们强烈建议在驱动器不接负载时学习 SCL 指令。这样可以保证您设备的安全性。

AC25<ENTER>　设置加速度为 25 rps/s
DE25<ENTER>　设置减速度为 25 rps/s
VE5<ENTER>　设置马达转速为 5rps
FL20000<ENTER>　让驱动器在顺时针旋转 20000 步

如果您的驱动器在 FL20000 指令后仍然没有转动，请查看您的驱动器 LED 显示是否有报错。如果有的话，您可以通过发送 AR（AR<ENTER>）指令来清除。如果报警清除了，驱动器绿灯常亮，这代表驱动器处于非使能状态。请发送 ME 指令（ME<ENTER>）来使能驱动器。如果绿灯闪烁代表驱动器已经使能，如果没有请再次尝试。

例子 2：

JA10<ENTER> 设置点动加速度为 10 rps/s
JL10<ENTER> 设置点动减速度为 10 rps/s
JS1<ENTER>　设置顺时针点动转速为 1rps
CJ<ENTER>　开始点动运转
CS-1<ENTER> 将点动的方向改为逆时针速度为 1 rps

SJ<ENTER>　停止点动

在上述例子中，马达的转速通过 CS 指令进行了新的设定，而速度的改变方式会根据 JA 和 JL 的加减速设定来改变。您可以以同样的方式，设定不同的 JA，JL，JS，CS 值来尝试不同的效果。

10.3 Command Summary 指令总结

这个章节按功能罗列了一系列 SCL 指令的具体列表，每个列表中包含了各个指令的具体信息。

- “Command” 描述指令的字符
- “Description” 描述指令的名字
- “NV”代表这个表示指令参数可以存储到非易失性存储器，它可以通过 SA（Save）指令来保存到驱动器的非易失性程序存储器中。注意：有一些指令（PA，PB，PC，PI，和 PM），其参数会立即被保存到驱动器的非易失性程序存储器中，不需要 SA 指令。
- “Write only” or “Read only”代表着该指令只读或者只写
- “Immediate” 代表该指令是立即指令
- “Compatibility” 指示该指令支持的驱动器

以便快速查找，指令被分为“运动，伺服，配置，I/O，通讯，Q 程序，寄存器”

- “运动”指令是用来控制步进或伺服马达运转的
- “伺服”指示包含了伺服的整定参数，马达的使能或去使能，或者滤波器设定
- “配置”指令用于根据应用配置驱动器参数。其中包括：伺服整定参数配置，步进细分和谐振参数等
- “I/O” 指令用与配置、查询或者控制驱动器的输入/输出
- “通讯”指令用与配置驱动器的通讯方式
- “Q 程序”指令用与来控制和执行驱动器的 Q 程序
- “寄存器”指令用与处理数据寄存器，大多数指令只适合 Q 系列驱动器。

10.3.1 运动指令

Command	Description	NV	write only	read only	Immediate	Compatibility
AC	Accel Rate	·				All drives
AM	Accel Max	·				All drives
CJ	Commence Jogging		·			All drives
DC	Distance for FC, FM, FO, FY	·				All drives
DE	Decel Rate	·				All drives
DI	Distance or Position	·				All drives
ED	Encoder Direction	·				Servos and steppers with encoder feedback
EF	Encoder Function	·				Servos and steppers with encoder feedback
EG	Electronic Gearing	·				All drives
EH	Extended Homing		·			All Step-Servo drives and M2 Servo drives
EI	Input Noise Filter	·				All drives

EP	Encoder Position					Servos and steppers with encoder feedback
FC	Feed to Length with Speed Change		•			All drives
FD	Feed to Double Sensor		•			All drives
FE	Follow Encoder		•			All drives
FH	Find Home		•			All Step-Servo drives and M2 Servo drives
FL	Feed to Length		•			All drives
FM	Feed to Sensor with Mask Dist		•			All drives
FO	Feed to Length & Set Output		•			All drives
FP	Feed to Position		•			All drives
FS	Feed to Sensor		•			All drives
FY	Feed to Sensor with Safety Dist		•			All drives
HA	Homing Acceleration	•				All Step-Servo drives and M2 Servo drives
HC	Hard Stop Current	•				All Step-Servo drives
HL	Homing Deceleration	•				All Step-Servo drives and M2 Servo drives
HO	Homing Offset	•				All Step-Servo drives and M2 Servo drives
HS	Hard Stop Homing		•			All Step-Servo drives
HV	Homing Velocity	•				All Step-Servo drives and M2 Servo drives
HW	Hand Wheel		•			All drives
JA	Jog Accel/Decel rate	•				All drives
JC	Velocity mode second speed	•				All drives
JD	Jog Disable		•			All drives
JE	Jog Enable		•			All drives
JL	Jog Decel rate	•				All drives
JM	Jog Mode	•				AI drives (see JM command)
JS	Jog Speed	•				All drives
MD	Motor Disable		•			All drives
ME	Motor Enable		•			All drives
MR	Microstep Resolution	•				Stepper drives only
PA	Power-up Accel Current	•				STM stepper drives only
SD	Set Direction	•				STM stepper drives with Flex I/O only
SH	Seek Home		•			All drives
SJ	Stop Jogging		•		•	All drives
SM	Stop the Move		•			Q drives only
SP	Set Absolute Position					All drives
ST	Stop Motion		•		•	All drives
VC	Velocity for Speed Change (FC)	•				All drives
VE	Velocity Setting (For Feed Commands)	•				All drives
VM	Velocity Max	•				All drives
WM	Wait on Move		•			Q drives only
WP	Wait on Position		•			Q drives only

10.3.2 伺服指令

Command	Description	NV	write only	read only	Immediate	Compatibility
CN	Second Control Mode	·				M2 servo drives only
CO	Node ID/ IP Address Series Number	·				M2 servo drives only
CP	Change Peak Current	·				Servo drives only
DD	Default Display Item of LEDs	·				M2 servo drives only
DS	Switching Electronic Gearing	·				M2 servo drives only
EN	Numerator of Electronic Gearing Ratio	·				M2 servo drives only
EP	Encoder Position					Servo drives only
EU	Denominator of Electronic Gearing Ratio	·				M2 servo drives only
FA	Function of the Single-ended Analog Input	·				M2 servo drives only
GC	Current Command	·			·	Servo drives only
GG	Controller Global Gain Selection	·				M2 servo drives only
IC	Immediate Current Command			·	·	Servo drives only
IE	Immediate Encoder Position			·	·	Servo drives only
IQ	Immediate Actual Current			·	·	Servo drives only
IX	Immediate Position Error			·	·	Servo drives only
JC	Eight Jog Velocities	·				M2 servo drives only
KC	Overall Servo Filter	·				Servo drives only
KD	Differential Constant	·				Servo drives only
KE	Differential Filter	·				Servo drives only
KF	Velocity Feedforward Constant	·				Servo drives only
KI	Integrator Constant	·				Servo drives only
KJ	Jerk Filter Frequency	·				SV7 Servo drives only
KK	Inertia Feedforward Constant	·				Servo drives only
KP	Proportional Constant	·				Servo drives only
KV	Velocity Feedback Constant	·				Servo drives only
MS	Control Mode Selection	·				M2 servo drives only
PF	Position Fault	·				Servo drives, drives with encoder feedback
PH	Inhibition of the pulse command	·				M2 servo drives only
PK	Parameter Lock	·				M2 servo drives only
PL	Position Limit	·				Servo drives only
PP	Power-Up Peak Current	·				Servo drives only
PV	Second Electronic Gearing	·				M2 servo drives only
TV	Torque Ripple	·				M2 servo drives only
VI	Velocity Integrator Constant	·				Servo drives only
VP	Velocity Mode Proportional Constant	·				Servo drives only
VR	Velocity Ripple	·				M2 servo drives only

10.3.3 配置指令

Command	Description	NV	write only	read only	Immediate	Compatibility
AL	Alarm Code			·	·	All drives
AR	Alarm Reset		·		·	All drives
BD	Brake Disengage Delay time	·				All drives
BE	Brake Engage Delay time	·				All drives
BS	Buffer Status			·	·	All drives
CA	Change Acceleration Current	·				STM stepper drives only
CC	Change Current	·				All drives
CD	Idle Current Delay	·				Stepper drives only
CF	Anti-resonance Filter Frequency	·				Stepper drives only
CG	Anti-resonance Filter Gain	·				Stepper drives only
CI	Change Idle Current	·				Stepper drives only
CM	Control mode	·				All drives
CP	Change peak current	·				Servo drives only
DA	Define Address	·				All drives
DL	Define Limits	·				All drives
DP	Dumping Power	·				SS drives only
DR	Data Register for Capture		·			Q servo drives only
ED	Encoder Direction	·				Servo drives, drives with encoder feedback
ER	Encoder or Resolution	·				Servo drives, drives with encoder feedback
HG	4th Harmonic Filter Gain	·				Stepper drives only
HP	4th Harmonic Filter Phase	·				Stepper drives only
IA	Immediate Analog			·	·	All drives
ID	immediate Distance			·	·	All drives
IE	Immediate Encoder			·	·	Servo drives, drives with encoder feedback
IF	Immediate Format	·			·	All drives
IQ	Immediate Current			·	·	Servo drives only
IP	Immediate Position			·	·	All drives
IT	Immediate Temperature			·	·	All drives
IU	Immediate Voltage			·	·	All drives
IV	Immediate Velocity			·	·	All drives
LP	Software Limit CW					All Step-Servo drives and M2 Servo drives
LM	Software Limit CCW					All Step-Servo drives and M2 Servo drives
LV	Low Voltage Threshold	·				All drives
MD	Motor Disable				·	All drives
ME	Motor Enable				·	All drives
MN	Model Number			·	·	All drives
MO	Motion Output	·				All drives

MR	Microstep Resolution	•				All drives (deprecated - see EG command)
MV	Model & Revision			•	•	All drives except Blu servos
OF	On Fault		•			Q drives only
OI	On Input		•			Q drives only
OP	Option Board	•		•	•	All drives
PA	Power-up Acceleration Current	•				
PC	Power up Current	•				All drives
PD	In Position Counts	•				All Step-Servo drives and M2 Servo drives
PE	In Position Timing	•				All Step-Servo drives and M2 Servo drives
PF	Position Fault	•				Servo drives, drives with encoder feedback
PI	Power up Idle Current	•				Stepper drives only
PL	In Position Limit	•				Servo drives only
PM	Power up Mode	•				All drives
PP	Power up peak current	•				Servo drives only
PW	Pass Word		•			Q drives only
RE	Restart / Reset		•		•	All drives
RL	Register Load				•	All drives
RS	Request Status			•	•	All drives
RV	Revision Level			•	•	All drives
SA	Save all NV Parameters		•			All drives
SC	Status Code			•	•	
SD	Set Direction	•				STM stepper drives with Flex I/O only
SF	Step Filter Frequency	•				Stepper drives only
SI	Enable Input usage	•				All drives
SK	Stop & Kill		•		•	All drives
TT	Pulse Complete Timing	•				All Step-Servo drives and M2 Servo drives
ZC	Regen Resistor Continuous Wattage	•				BLuAC5 and STAC6 drives only
ZR	Regen Resistor Value	•				BLuAC5 and STAC6 drives only
ZT	Regen Resistor Peak Time	•				BLuAC5 and STAC6 drives only

10.3.4 I/O 指令

Command	Description	NV	write only	read only	Immediate	Compatibility
AD	Analog Deadband	•				All stepper drives and SV servo drives
AF	Analog Filter	•				All drives
AG	Analog Velocity Gain	•				All stepper drives and SV servo drives
AI	Alarm Input usage	•				All drives
AN	Analog Torque Gain	•				All Step-Servo drives and M2 Servo drives
AO	Alarm Output usage	•				All drives

AP	Analog Position Gain	•				All drives
AS	Analog Scaling	•				All stepper drives and SV servo drives
AT	Analog Threshold	•				All drives
AV	Analog Offset	•				All drives
AZ	Analog Zero (Auto Zero)		•			All drives
BD	Brake Disengage Delay time	•				All drives
BE	Brake Engage Delay time	•				All drives
BO	Brake Output usage	•				All drives
DL	Define Limits	•				All drives
EI	Input Noise Filter	•				All drives
FI	Filter Input	•				All drives (Note: not NV on Blu servos)
FX	Filter Selected Inputs					Blu, STAC5, STAC6, SVAC3
IH	Immediate High Output		•		•	All drives
IL	Immediate Low Output		•		•	All drives
IO	Output Status				•	All drives
IS	Input Status request			•	•	All drives
MO	Motion Output	•				All drives
OI	On Input		•			Q drives only
SI	Enable Input usage	•				All drives
SO	Set Output		•			All drives
TI	Test Input		•			Q drives only
TO	Tach Output	•				TSM drives only
WI	Wait on Input		•			All drives

10.3.5 通讯指令

Command	Description	NV	write only	read only	Immediate	Compatibility
BR	Baud Rate	•				All drives
BS	Buffer Status				•	All drives
CE	Communications Error				•	All drives
IF	Immediate Format	•			•	All drives
PB	Power up Baud Rate	•				All drives
PR	Protocol	•				All drives
TD	Transmit Delay	•				All drives

10.3.6 Q 程序指令

Command	Description	NV	write only	read only	Immediate	Compatibility
AX	Alarm Reset		•			All drives
MT	Multi-Tasking					Q drives only
NO	No Operation		•			Q drives only
OF	On Fault		•			Q drives only
OI	On Input		•			Q drives only
PS	Pause		•			All drives

QC	Queue Call		·			Q drives only
QD	Queue Delete		·			Q drives only
QE	Queue Execute		·		·	Q drives only
QG	Queue Goto		·			Q drives only
QJ	Queue Jump		·			Q drives only
QK	Queue Kill		·			Q drives only
QL	Queue Load		·		·	Q drives only
QR	Queue Repeat		·			Q drives only
QS	Queue Save		·		·	Q drives only
QU	Queue Upload			·	·	Q drives only
QX	Queue Load & Execute		·			Q drives only
SM	Stop Move		·			Q drives only
SS	Send String		·			All drives
TI	Test Input		·			Q drives only
WD	Wait Delay using Data Register		·			Q drives only
WI	Wait for Input		·			All drives
WM	Wait for Move to complete		·			Q drives only
WP	Wait for Position in complex move		·			Q drives only
WT	Wait Time		·			Q drives only

10.3.7 寄存器指令

Command	Description	NV	write only	read only	Immediate	Compatibility
CR	Compare Register		·			Q drives only
DR	Data Register for Capture		·			Q drives only
RC	Register Counter		·			Q drives only
RD	Register Decrement		·			Q drives only
RI	Register Increment		·			Q drives only
RL	Register Load				·	Q drives only
RM	Register Move		·			Q drives only
RR	Register Read		·			Q drives only
RU	Register Upload		·		·	
RW	Register Write		·			Q drives only
RX	Register Load					Q drives only
R+	Register Addition		·			Q drives only
R-	Register Subtraction		·			Q drives only
R*	Register Multiplication		·			Q drives only
R/	Register Division		·			Q drives only
R&	Register Logical AND		·			Q drives only
R|	Register Logical OR		·			Q drives only
TR	Test Register		·			Q drives only
TS	Time Stamp read		·			Q drives only

10.4 Host Command Reference

11 Q 编程参考

SCL 指令的使用已经有很多年的历史。命令与卫星的历史可以追溯到很多年的使用。Q 程序是建立在 SCL 指令基础之上的一个新的平台，扩展了 SCL 命令的使用，允许用户创建与存储 SCL 指令。这些程序可以保存在驱动器的非易失性存储器中，驱动可以脱离主机独立运行这些程序。Q 程序为系统设计提供了高度的灵活性和强大的功能。，主要特点如下：

- 运动控制（例如 FL、FP、SH 等）
- 执行驻留程序
- 多任务处理（请参阅 MT 指令）
- 条件判断（例如 OI、TI 等指令）
- 数学运算（例如 R+、R-、R*、R/、R|、R&等指令）
- 寄存器操作（例如 RX、RM 等指令）

1 个 Q 程序支持多达 12 个程序段，每个程序段最多可以编写 62 条指令。

11.1 参考例程

以下内容给出了 Q 编程的参考例程，并对这些指令逐条进行了解释，Q 编程中所有指令均为缓存指令（Buffered Commands）

11.1.1 点到点相对运动（Feed to Length）



Line	Label	Cmd	Param1	Param2	Comment
1		WI	X3F		Wait for falling Edge of Input #3
2		VE	20		Set Velocity to 20 Rev/Sec
3		DI	32000		Set Distance to 4 revs
4		FL			Do a Feed to Length
5					

FL（Feed to Length）指令用来完成点到点相对位置运动，当执行该指令时，电机将按照设定的加速度（AC），减速度（DE）及运行速度（VE），完成一段固定的相对位置距离（DI）。电机转动的方向由 DI 指令的正负极性决定。比如，DI32000 代表电机顺时针转动 32000 步（或微步），而 DI-32000 代表电机逆时针转动 32000 步（或微步）。上图中列出了一个参考例程，Q 程序通过 WI 指令首先等待输入口 3 接收一个下降沿触发信号，如果条件满足，电机将按照 20 转/秒的转速，转动 4 圈；如果条件不满足，程序将一直停留在 WI 指令处等待输入口 3 接收正确的触发信号为止。

11.1.2 点到点绝对运动（Feed to Position）



Line	Label	Cmd	Param1	Param2	Comment
1		WI	X3F		Wait for falling Edge of Input #3
2		VE	20		Set Velocity to 20 Rev/Sec
3		DI	32000		Set Distance to 4 revs
4		FL			Do a Feed to Length
5		WT	1		Wait 1 second
6		DI	0		Set feed position to ""0""
7		FP			Do a Feed to Position
8					

FP（Feed to Position）指令用来完成点到点绝对位置运动，当执行该指令时，电机将按照设定的加速度（AC），减速度（DE）及运行速度（VE），完成一段绝对位置距离（DI）。DI 指令代表目标绝对位置，电机转动方向不由其正负极性决定，而由电机当前绝对位置与目标绝对位置来决定。上图中列出了一个参考例程，Q 程序通过 WI 指令首先等待输入口 3 接收一个下降沿触发信号，如果条件满足，电机将按照 20 转/秒的转速，转动 4 圈，然后等待 1 秒（WT 指令），然后电机以 20 转/秒的转速回到绝对位置 0 点；如果条件不满足，程序将一直停留在 WI 指令处等待输入口 3 接收正确的触发信号为止。

SP（Set Position）指令用来设置电机当前的绝对位置，如发送“SP0”指令，将电机当前位置设为绝对位置零点。注意，SP 指令设置的参数是以编码器 Encoder Counts 作为基本单位的，比如一个装有 500 线编码器的电机，一圈有 2000 个 Encoder Counts，如发送“SP5000”指令，将电机当前位置设为以绝对位置零点顺时针方向 2.5 圈的位置。

11.1.3 运动到传感器位置（Feed to Sensor）



Line	Label	Cmd	Param1	Param2	Comment
1		WI	X3F		Wait for falling Edge of Input #3
2		DL	3		Turn OFF limit detection
3		VE	5		Set Velocity to 5 Rev/Sec
4		DI	8000		Set offset Distance to 1 rev
5		FS	X7H		Do a Feed to Sensor (#7 high)
6		WT	1		Wait 1 second
7		VE	20		Set Velocity to 20 Rev/Sec
8		DI	0		Set feed position to ""0""
9		FP			Do a Feed to Position
10		DL	2		Turn ON limit detection

FS（Feed to Sensor）指令使电机以一个固定速度转动，直到一个输入口的电平状态满足触发条件，电机减速，运动停止。运动参数由 AC，DE，VE 及 DI 指令决定，注意 DI 指令在 FS 指令执行时代表输入口电平状态满足触发条件后，电机减速到 0 的运动距离，注意 DI 设定的减速具体必须大于最小减速距离 Dm，最小减速距离由 DE，VE，EG 指令决定，计算公式如下，其中 V 代表 VE 设定值，R 代表 EG 设定值，D 代表 DE 设定值：

$$\text{minimum decel distance} = \frac{(V)^2(R)}{2(D)}$$

注意，当 DI 设定值大于最小减速距离 Dm 时，电机接收到输入口触发电平状态（找到传感器）后，会继续以当前速度向前运动（DI-Dm）的距离，然后以 DE 设定的减速度减速，直到完成 Dm 的减速距离。同时，DI 指令也决定了电机刚开始运动的方向，比如，DI8000 代表执行 FS 指令后，电机顺时针转动，而 DI-8000 代表执行 FS 指令后，电机逆时针转动。输入口电平状态分为 H（高电平），L（低电平），R（上升沿），F（下降沿）这几种状态。

上图中列出了一个参考例程，Q 程序通过 WI 指令首先等待输入口 3 接收一个下降沿触发信号，如果条件满足，取消限位功能，电机以 5 转/秒的转速顺时针方向转动，直到输入口 7 接收到一个高电平触发状态，驱动器找到了传感器，于是以 DE 设定的减速度，DI 设定的减速距离完成减速到 0。然后电机等待 1 秒钟，随后以 20 转/秒的速度运行到绝对位置零点处，打开限位功能。

11.1.4 循环（Looping）

用户可以通过两种方式来实现程序的循环。第一种使用 QG（Queue Goto）指令，设置 QG 参数使程序回到设定的行数。下图中列出了一个参考例程，在 FL 指令后等待时间 0.5 秒（WT0.5），然后使用 QG 指令，使程序回到第一行重新执行，实现不断循环。



Line	Label	Cmd	Param1	Param2
1	Label1	DI	40000	
2		AC	500	
3		DE	500	
4		VE	20	
5		FL		
6		WT	0.5	
7		QG	#Label1	

第二种方法是使用 QR（Queue Repeat）指令，QR 指令指明跳转到哪一行和循环次数。下图中列出了一个参考例程，QR 指令表示跳转到第二行，参数为 3，即循环次数由用户自定义寄存器 3 中的值决定，在这里 RX 指令写入数值 5 到自定义寄存器 3， 所以程序循环执行 5 次。



Line	Label	Cmd	Param1	Param2
1		RX	3	5
2	Label1	DI	40000	
3		AC	500	
4		DE	500	
5		VE	20	
6		FL		
7		WT	0.5	
8		QR	3	#Label1
9				

11.1.5 跳转（Jumping）

程序跳转由 QJ（Queue Jump）指令实现，跳转和循环不同，跳转主要判断条件是否满足，

即跳转指令通常和 TI（输入检测），TR（寄存器检测），CR（寄存器比较）指令配合使用。下图中列出了一个参考例程，有两种可能的运动，顺时针旋转，当输入信号 5 有效时（低电平有效）电机开始逆时针旋转，加速度 300，减速度 450，速度 18.5，两种运动间有 0.25 秒的等待时间。然后，检测 X5 信号的状态，当 X5 信号有效时（True），程序跳转到第 10 行，开始逆时针旋转，如果 X5 信号为高时程序直接到第 7 行进行顺时针旋转， 执行完后回到第一行进行循环。



Line	Label	Cmd	Param1	Param2
1	Label2	AC	300	
2		DE	450	
3		VE	18.5	
4		WT	0.25	
5		TI	5L	
6		QJ	T	#Label1
7		DI	50000	
8		FL		
9		QG	#Label2	
10	Label1	DI	-50000	
11		FL		
12		QG	#Label2	

11.1.6 程序调用（Calling）

程序调用是在不同的程序段（Segment）之间实现的。QC（Queue Call）指令允许用户结束一个当前程序段，调用另一个程序段，执行完后，回到执行调用指令的第一个程序段。这样，用户可以把需要多次循环的程序单独放到一个段中进行调用，以减少循环的次数和降低程序结构的复杂度。下图中列出了一个参考例程，由第 1 程序段（Segment 1）和第 2 程序段（Segment 2）组成：

第1程序段 第2程序段

当前程序段

打开 保存 打印

Segment 1

Line	Label	Cmd	Param1	Param2
1	Label1	AC	300	
2		DE	450	
3		VE	18.5	
4		DI	40000	
5		FL		
6		QC	2	
7		VE	1	
8		DI	4000	
9		FL		
10		QC	2	
11		QG	#Label1	

第1程序段 第2程序段

当前程序段

打开 保存 打印

Segment 2

Line	Label	Cmd	Param1	Param2
1		SO	1L	
2		WT	0.25	
3		SO	2L	
4		WT	0.25	
5		SO	2H	
6		WT	0.25	
7		SO	1H	
8		QC	1	

第 1 程序段在第 6、8 行调用第 2 程序段，第 2 程序段中设置 Y1 低电平输出，等待 0.25 秒，设置 Y2 低电平输出，等待 0.25 秒。然后设置 Y2 高电平输出，等待 0.25 秒，设置 Y1 高电平输出，回到第 1 程序段，继续执行第 1 程序段内后续指令。

11.1.7 多任务处理（Multi-tasking）

多任务处理模式（MT1）允许程序在执行运行指令（如 FL，FP，CJ，FS 等）时，同时执行其他指令，而不需要等待前一个运行指令的结束。在单任务处理模式（MT0）下，Q 程序是顺序执行的，即执行下一条指令会等待上一条指令的结束。例如，FL 指令后是 SO 指令，那么驱动器在 FL 指令结束后才会设置输出。当开启多任务处理模式（MT1），Q 程序执行运行指令的同时会执行后面的指令。例如，上面的 FL，SO 指令，执行多任务处理，驱动器开始运行并且

立即执行输出设置，不需要等待 FL 指令执行完毕。多任务处理由 MT 指令设置，MT1 为多任务处理开启，MT0 为多任务处理关闭（单任务处理模式）。



例如上图所示，当 MT=1，驱动器执行 FL 指令，等待 0.5 秒后设置输出口 Y1 为低电平，不需要等待 FL 指令结束后，再等待 0.5 秒才输出低电平。以上是一个最基本的例子，如果您尝试对您的驱动器进行编程，请尽量保证 DI 值足够大以观察不同的指令执行后的区别。注意，因为电机不能同时执行两种运动，即使多任务处理开启后，运动指令还是会有先后顺序。例如，多任务处理开启后，程序中连续有两种运行指令，那么驱动器还是会等待第一条指令完成后执行下一条指令。

12 CANopen 参考

12.1 CANopen 通讯

CANopen 是一种架构在控制器局域网路（Controllor Area Network, CAN）上的高层通讯协定，包括通讯子协定及设备子协定常在嵌入式系统中使用，也是工业控制常用到的一种现场总线。

CANopen 实作了 OSI 模型中的网络层以上（包括网络层）的协定。CANopen 标准包括寻址方案、数个小的通讯子协定及由设备子协定所定义的应用层。 CANopen 支援网络管理、设备监控及节点间的通讯，其中包括一个简易的传输层，可处理资料的分段传送及其组合。一般而言资料链结层及实体层会用 CAN 来实作。

基本的 CANopen 设备及通讯子协定定义在 CAN in Automation (CiA) 301。针对个别设备的子协定以 CiA 301 为基础再进行扩充。如针对 I/O 模组的 CiA 401 及针对运动控制的 CiA 402。

12.2 为什么选用 CANopen?

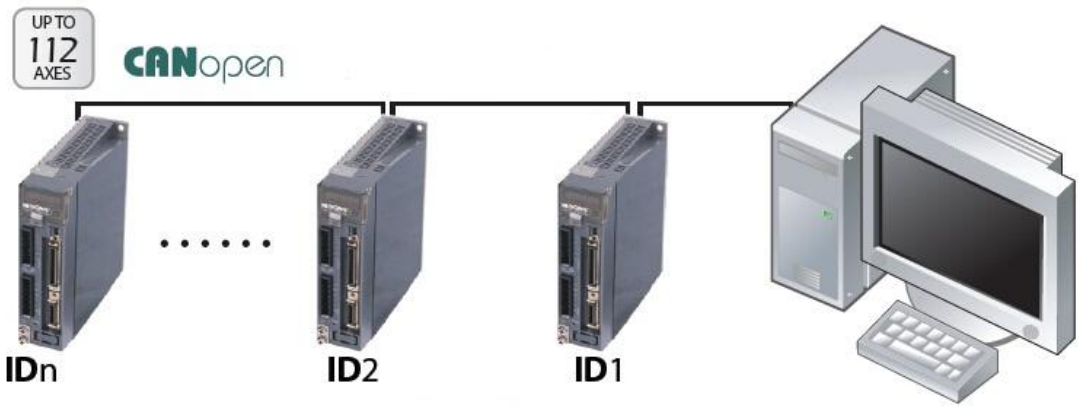
可支持多轴

CANopen 最多支持控制 127 个轴，且通讯速率可高达 1Mbps。

CANopen 总线对数据通信数据块进行编码的方式，可以**多主方式工作**，采用可靠的错误处理和检错机制，其协议包含了系统的，专业的工业用设备子协议，是专门为工业控制而发展的总线协议。符合工业控制所要求的高速，稳定可靠，高实时性要求

布线简单

传统脉冲或模拟量方式一个轴动辄需要连接 4 根线或者更多，CANopen 方案布线简单，所有轴并联接入 CAN_H 和 CAN_L 信号即可。减少布线，更少出错，减少布线成本，人工成本，后期电缆老化或故障的维护成本，以及由于复杂的线路故障带来的不可估量的损失。



常见的 CANopen 术语

- 节点 ID（NODE-ID）
- 对象字典 OD（Object Dictionary）

- 电子数据表 EDS（Electronic Data Sheet）
- 通讯对象标识符 COB-ID(Communication Object Identification)
- 网络管理 NMT（Network Management）
- 过程数据对象 PDO(Process Data Object)
- 服务数据对象 SDO(Service Data Object)
- 接收过程数据对象 RPDO(Receive PDO)
- 发送过程数据对象 TPDO(Transmit PDO)

12.3 CANopen 参考例程

12.3.1 位置模式（Profile Position Mode）

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 ‘Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 ‘Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Operation Enabled
**** Set to Profile Position Mode ****
$0603 $8 $2F $60 $60 $00 $01 $00 $00 $00 ‘Set to Profile Position Mode
**** Set Motion Parameters ****
$0603 $8 $23 $81 $60 $00 $F0 $00 $00 $00 ‘Set Profile Velocity to 1 rps
$0603 $8 $23 $83 $60 $00 $58 $02 $00 $00 ‘Set Acceleration to 100 rps/s
$0603 $8 $23 $84 $60 $00 $58 $02 $00 $00 ‘Set Deceleration to 100 rps/s
Single Move Absolute
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $1F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Clear New Set Point Bit
Single Move Relative
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $5F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $00 $00 $00 ‘Clear New Set Point Bit
Multiple Move, Stopping between Moves
$0603 $8 $23 $81 $60 $00 $B0 $04 $00 $00 ‘Set Profile Velocity to 5 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $5F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $00 $00 $00 ‘Clear New Set Point Bit
$0603 $8 $23 $81 $60 $00 $60 $09 $00 $00 ‘Set Profile Velocity to 10 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 600000 steps
$0603 $8 $2B $40 $60 $00 $5F $00 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $00 $00 $00 ‘Clear New Set Point Bit
Multiple Move, Continuous Motion

$0603 $8 $23 $81 $60 $00 $B0 $04 $00 $00 ‘Set Profile Velocity to 5 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $5F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $02 $00 $00 ‘Clear New Set Point Bit
$0603 $8 $23 $81 $60 $00 $60 $09 $00 $00 ‘Set Profile Velocity to 10 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 600000 steps
$0603 $8 $2B $40 $60 $00 $5F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $4F $02 $00 $00 ‘Clear New Set Point Bit
Multiple Move, Immediate Change in Motion
$0603 $8 $23 $81 $60 $00 $B0 $04 $00 $00 ‘Set Profile Velocity to 5 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 200000 steps
$0603 $8 $2B $40 $60 $00 $7F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $6F $02 $00 $00 ‘Clear New Set Point Bit
$0603 $8 $23 $81 $60 $00 $60 $09 $00 $00 ‘Set Profile Velocity to 10 rps
$0603 $8 $23 $7A $60 $00 $40 $0D $03 $00 ‘Set Target Position to 600000 steps
$0603 $8 $2B $40 $60 $00 $7F $02 $00 $00 ‘Set New Set Point Bit to 1
$0603 $8 $2B $40 $60 $00 $6F $02 $00 $00 ‘Clear New Set Point Bit

12.3.2 速度模式（Profile Velocity Mode）

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 ‘Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 ‘Switched on
$0603 $8 $2B $40 $60 $00 $0F $01 $00 $00 ‘Operation Enabled; Motion Halted
**** Set to Profile Velocity Mode ****
$0603 $8 $2F $60 $60 $00 $03 $00 $00 $00 ‘Set to Profile Velocity Mode
**** Set Motion Parameters ****
$0603 $8 $23 $FF $60 $00 $F0 $00 $00 $00 ‘Set Target Velocity to 1 rps
$0603 $8 $23 $83 $60 $00 $58 $02 $00 $00 ‘Set Acceleration to 100 rps/s
$0603 $8 $23 $84 $60 $00 $58 $02 $00 $00 ‘Set Deceleration to 100 rps/s
**** Start/Stop Motion ****
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Motion Starts
$0603 $8 $23 $FF $60 $00 $60 $09 $00 $00 ‘Change Target Velocity to 10 rps
$0603 $8 $2B $40 $60 $00 $0F $01 $00 $00 ‘Motion Halts

12.3.3 回原点模式（Homing Mode）

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 ‘Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 ‘Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Operation Enabled

**** Set to Homing Mode ****
$0603 $8 $2F $60 $60 $00 $06 $00 $00 $00 ‘Set to Homing Mode
$0603 $8 $2F $98 $60 $00 $13 $00 $00 $00 ‘Set Homing Method to 19
**** Set Motion Parameters ****
$0603 $8 $23 $9A $60 $00 $58 $02 $00 $00 ‘Set Homing Acceleration to 100rps/s
$0603 $8 $23 $99 $60 $01 $F0 $00 $00 $00 ‘Set Homing Velocity (Search for Switch) to 1rps
$0603 $8 $23 $99 $60 $02 $78 $00 $00 $00 ‘Set Index Velocity (Search for Index or Zero) to 0.5rps
$0603 $8 $23 $7C $60 $00 $40 $9C $00 $00 ‘Set Homing Offset to 40000 Steps
$0603 $8 $2F $01 $70 $00 $03 $00 $00 $00 ‘Set Homing Switch to Input 3
**** Start/Stop Homing ****
$0603 $8 $2B $40 $60 $00 $1F $00 $00 $00 ‘Homing Starts
$0603 $8 $2B $40 $60 $00 $1F $01 $00 $00 ‘Homing Stops

12.3.4 通用 Q 模式（Normal Q Mode）

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 ‘Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 ‘Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Operation Enabled
**** Set to Normal Q Mode ****
$0603 $8 $2F $60 $60 $00 $FF $00 $00 $00 ‘Set to Normal Q Mode
$0603 $8 $2F $07 $70 $00 $01 $00 $00 $00 ‘Set Q Segment Number to 1
**** Start/Stop Q Program ****
$0603 $8 $2B $40 $60 $00 $1F $00 $00 $00 ‘Q Program Starts
$0603 $8 $2B $40 $60 $00 $1F $01 $00 $00 ‘Q Program Halts

12.3.5 同步 Q 模式（Sync Q Mode）

**** Enable Motor Power - CiA 402 State Machine ****
ID DLC Data
$0603 $8 $2B $40 $60 $00 $06 $00 $00 $00 ‘Ready to Switch on
$0603 $8 $2B $40 $60 $00 $07 $00 $00 $00 ‘Switched on
$0603 $8 $2B $40 $60 $00 $0F $00 $00 $00 ‘Operation Enabled
**** Set to Sync Q Mode ****
$0603 $8 $2F $60 $60 $00 $FE $00 $00 $00 ‘Set to Sync Q Mode
$0603 $8 $2F $07 $70 $00 $01 $00 $00 $00 ‘Set Q Segment Number to 1
$0603 $8 $23 $05 $10 $00 $80 $00 $00 $00 ‘Set Sync Pulse to 0x80
**** Start/Stop Q Program ****
$80 $0 ‘Q Program Starts
$0603 $8 $2B $40 $60 $00 $0F $01 $00 $00 ‘Q Program Halts

12.3.6 PDO 映射（PDO Mapping）

****Mapping TPDO2 ****
$0000 $2 $80 $03 ‘Return back to “PreOperation” Mode
$0603 $8 $23 $01 $18 $01 $80 $02 $00 $80 ‘Turn off the TPDO2
$0603 $8 $2F $01 $1A $00 $00 $00 $00 $00 ‘Set Number of Mapped objects to zero
$0603 $8 $23 $01 $1A $01 $10 $00 $41 $61 ‘Map object1(0x6041) to TPDO2 subindex1.
$0603 $8 $23 $01 $1A $02 $20 $00 $0A $70 ‘Map object2(0x700A) to TPDO2 subindex2.
$0603 $8 $2F $01 $1A $00 $02 $00 $00 $00 ‘Set Number of total Mapped objects to Two
$0603 $8 $23 $01 $18 $01 $80 $02 $00 $00 ‘Turn on the TPDO2

12.4 CANopen 相关下载

Eds 文件	链接
CANopen 用户手册	链接

13 Modbus/RTU 参考

鸣志的 Modbus 主要是用基于串行总线的 Modbus RTU 模式.

MODBUS 通讯协议是一种工业现场总线通讯协议，MODBUS 是 OSI 模型第 7 层上的应用层报文传输协议。它定义的是一种设备控制器可以识别和使用的信息帧结构，独立于物理层介质，可以承载于多种网络类型中。

- 目前使用下列的物理层实现 MODBUS：
  a) 基于以太网上的 TCP/IP。
  b) 各种介质（有线：EIA/TIA-232-E、EIA-422、EIA/TIA-485-A；光纤、无线等等）上的异步串行传输。

Modbus 协议是一个 master/slave 架构的协议。有一个节点是 master 节点，其他使用 Modbus 协议参与通信的节点是 slave 节点。每一个 slave 设备都有一个唯一的地址。在串行网络中，只有被指定为主节点的节点可以启动一个命令（在以太网上，任何一个设备都能发送一个 Modbus 命令，但是通常也只有一个主节点设备启动指令）。

一个 ModBus 命令包含了准备执行的设备的 Modbus 地址。所有设备都会收到命令，但只有指定位置的设备会执行及回应指令（地址 0 例外，指定地址 0 的指令是广播指令，所有收到指令的设备都会运行，不过不回应指令）。所有的 Modbus 命令包含了校验码，以确定到达的命令没有被破坏。基本的 ModBus 命令能指令一个 RTU 改变它的寄存器的某个值，控制或者读取一个 I/O 端口，以及指挥设备回送一个或者多个其寄存器中的数据。

13.1 通讯地址

网络中的每个从设备都必须分配给一个唯一的地址，只有符合地址要求的从设备才会响应主设备发出的命令。在 Modbus 地址“0”是广播地址 ，不能作为从站地址。Modbus RTU/ASCII 下从站地址从 1 到 31。

13.2 数据编码

驱动器使用“Big-Endian”或者“Little-Endian”表示 32 位数据的存放。每个寄存器都为 16 位的 word，故在存放 32 位数据时，需要占用 2 个寄存器地址。

Big-Endian：就是高位字节排放在内存的低地址端，低位字节排放在内存的高地址端

例如：在寄存器地址 40031 和 40032 中存放一个 32 位的数据 0x12345678，其中 0x1234 是高位字节，0x5678 是低位字节。使用 Big-Endian 存放时：

则 40031 = 0x1234
40032 = 0x5678

那么：要传送 0x12345678 时，发送的第一个字节是 0x1234，然后是 0x5678.

Little-Endian：就是低位字节排放在内存的低地址端，高位字节排放在内存的高地址端

例如：在寄存器地址 40031 和 40032 中存放一个 32 位的数据 0x12345678，其中 0x1234 是高位字节，0x5678 是低位字节。使用 Big-Endian 存放时：

则 40031 = 0x5678

40032 = 0x1234

那么：要传送 0x12345678，发送的第一个字节是 0x5678，然后是 0x1234.

指令 PR 定义了 Modbus/RTU 的编码方式。

13.3 通讯速率及通讯协议

伺服的通讯格式固定为 8，N，1。即数据位：8，停止位：1，奇偶校验：None。

指令 BR 及 PB 定义了通讯的波特率。

串行通信中的上电后生效波特率。该值被配置后将会立即被保存但不会立即生效，直到下次上电才生效，所以上位机软件可以随时配置该值。

1 = 9600bps

2 = 19200bps

3 = 38400bps

4 = 57600bps

5 = 115200bps

13.4 Modbus/RTU 的消息帧

MODBUS RTU 是一种主从技术，且 CRC 校验范围为是从设备地址位到数据位。Modbus/RTU 的标准消息帧如下：

Address 地址	Function Code 功能码	Data 数据	CRC 循环冗余校验

根据数据结果，可以得到两种类型的响应报文：

正常的 Modbus 响应：

响应功能码 = 请求功能码

异常的 Modbus 响应：

响应功能码 = 请求功能码 + 0x80

提供一个异常码来指示差错原因

13.5 功能码

MOONS 驱动器目前支持如下的 MODBUS 功能码：

1) 0x03： 读保持寄存器；

2) 0x04： 读输入寄存器；

3) 0x06： 写单个寄存器；

4) 0x10：写多个寄存器。

13.5.1 功能码 0x03，读多个保存寄存器

读取从站地址为 1 的编码器实际位置，该寄存器地址为 40005，是 32 位数据，假设里面数值为 250000.

参数 P-75(PR) = 5 ，以 Big-Endian 传送

通讯过程如下表：

主站发送信息： Command Message（Master）				从站回传报文 Response Message（slave）		
功能	数据	字节数		功能	数据	字节数
Slave Address 从站地址	01H	1		Slave Address 从站地址	01H	1
Function Code 功能码	03H	1		Function Code 功能码	03H	1
Starting Data Address 数据起始地址 (寄存器 40005)	00H(High) 04H(Low)	2		Number of Data （In Byte） 数据数	04	1
Number of Data （In word） 数据个数	00(High) 02(Low)	2		Content of Starting Data Address 40005 起始地址 40005 的数据	00H(High) 26H(Low)	2
CRC Check Low CRC 校验低字节	85	1		Content of second Data Address 40006 第二个地址 40006 的数据	25H(High) A0(Low)	2
CRC Check High CRC 校验高字节	CA	1		CRC Check Low CRC 校验低字节	01H	1
				CRC Check High CRC 校验高字节	10H	1

即主机发送： 01 03 00 04 00 02 85 CA

驱动器返回： 01 03 04 00 26 25 A0 01 10

如果报错：则返回的数据格式为 01 83 XX CRC_L CRC_H

其中 XX = 01 ： 不支持 03 功能码

XX = 02 ： 读取驱动器地址及个数不正确

XX = 03 ： 读取寄存器地址超出范围

XX = 04 ： 读取失败

13.5.2 功能码 0x06，写单个寄存器。

写入数据给从站地址为 11，该寄存器为电机运行速度，该寄存器地址为 40030。

假设我们需要设定电机转速为 12.5rps，则写入数据为 12.5 x 240 = 3000，换算成 16 进制为：12CH。

通讯过程如下表：

主站发送信息： Command Message（Master）				从站回传报文 Response Message（slave）		
功能	数据	字节数		功能	数据	字节数
Slave Address 从站地址	0BH	1		Slave Address 从站地址	0BH	1
Function Code 功能码	06H	1		Function Code 功能码	06H	1
Starting Data Address 数据起始地址 (寄存器 40030)	00H(High) 1DH(Low)	2		Starting Data Address 数据起始地址 (寄存器 40030)	00H(High) 1DH(Low)	2
Content of Data 数据内容	01(High) 2C(Low)	2		Content of Data 数据内容	01(High) 2C(Low)	2
CRC Check Low CRC 校验低字节	19	1		CRC Check Low CRC 校验低字节	19	1
CRC Check High CRC 校验高字节	2B	1		CRC Check High CRC 校验高字节	2B	1

即主机发送： 0B 06 00 1D 01 2C 19 2B

驱动器返回： 0B 06 00 1D 01 2C 19 2B

如果报错：则返回的数据格式为 01 86 XX CRC_L CRC_H

其中 XX = 01 ： 不支持 06 功能码

XX = 02 ： 写入驱动器地址不正确

XX = 03 ： 写入寄存器地址超出范围

XX = 04 ： 写入失败

13.5.3 功能码 0x10，写多个寄存器

向从站地址为 10 的驱动器写入目标距离 30000，换算成 16 进制 7530h，目标距离对应对寄存器地址为 40031.

主站发送信息： Command Message（Master）				从站回传报文 Response Message（slave）		
功能	数据	字节数		功能	数据	字节数
Slave Address 从站地址	0AH	1		Slave Address 从站地址	0AH	1
Function Code 功能码	10H	1		Function Code 功能码	10H	1
Starting Data Address 数据起始地址 (寄存器 40031)	00H(High) 1EH(Low)	2		Starting Data Address 数据起始地址 (寄存器 40031)	00H(High) 1EH(Low)	2
Number of Data （In word） 数据数	00H(High) 02H(Low)	2		Number of Data （In word） 数据数	00H(High) 02H(Low)	2
Number of Data （In byte） 数据数	04H	1		CRC Check Low CRC 校验低字节	20	1
Content of first Data address 第一个地址的数据内容	00(High) 00(Low)	2		CRC Check High CRC 校验高字节	B5	1
Content of second Data address 第二个地址的数据内容	75H(High) 30H(Low)	2				
CRC Check Low CRC 校验低字节	70	1				
CRC Check High CRC 校验高字节	8F	1				

即主机发送 ： 0A 10 00 1E 00 02 04 00 75 30 70 8F

驱动器返回 ： 0A 10 00 1E 00 02 20 B5

如果报错：则返回的数据格式为 01 90 XX CRC_L CRC_H

其中 XX = 01 ： 不支持 10 功能码

XX = 02 ： 写入驱动器地址个个数不正确

XX = 03 ： 写入寄存器地址超出范围

XX = 04 ： 写入失败

13.6 Modbus 寄存器表

Modbus 寄存器表				
寄存器地址	属性	数据类型	对应SCL指令	对应SCL寄存器
40001	Read	SHORT	Alarm Code (AL)	f
40002	Read	SHORT	Status Code (SC)	s
40003	Read	SHORT	Immediate Expanded Inputs (IS)	y
40004	Read	SHORT	Driver Board Inputs (ISX)	i
40005..6	Read	LONG	Encoder Position (IE, EP)	e
40007..8	Read	LONG	Immediate Absolute Position	I
40009..10	Write	LONG	Absolute Position Command	P(大写)
40011	Read	SHORT	Immediate Actual Velocity (IV0)	v
40012	Read	SHORT	Immediate Target Velocity (IV1)	w
40013	Read	SHORT	Immediate Drive Temperature (IT)	t
40014	Read	SHORT	Immediate Bus Voltage (IU)	u
40015..16	Read	LONG	Immediate Position Error (IX)	x
40017	Read	SHORT	Immediate Analog Input Value (IA)	a
40018	Read	SHORT	Q Program Line Number	b
40019	Read	SHORT	Immediate Current Command (IC)	c
40020..21	Read	LONG	Relative Distance (ID)	d
40022..23	Read	LONG	Sensor Position	g
40024	Read	SHORT	Condition Code	h
40025	Read	SHORT	Analog Input 1 (IA1)	j
40026	Read	SHORT	Analog Input 2 (IA2)	k
40027	Read	SHORT	Command Mode (CM)	m
40028	R/W	SHORT	Point-to-Point Acceleration (AC)	A
40029	R/W	SHORT	Point-to-Point Deceleration (DE)	B
40030	R/W	SHORT	Velocity (VE)	V
40031..32	R/W	LONG	Point-to-Point Distance (DI)	D
40033..34	R/W	LONG	Change Distance (DC)	C
40035	R/W	SHORT	Change Velocity (VC)	U
40036	Read	SHORT	Velocity Move State	n

40037	Read	SHORT	Point-to-Point Move State	o
40038	Read	SHORT	Q Program Segment Number	p
40039	Read	SHORT	Average Clamp Power (regen)	r
40040	Read	SHORT	Phase Error	z
40041..42	R/W	LONG	Position Offset	E
40043	R/W	SHORT	Miscellaneous Flags	F
40044	R/W	SHORT	Current Command (GC)	G
40045..46	R/W	LONG	Input Counter	I
40047	R/W	SHORT	Jog Accel (JA)	
40048	R/W	SHORT	Jog Decel (JL)	
40049	R/W	SHORT	Jog Velocity (JS)	J
40050	R/W	SHORT	Accel/Decel Current (CA)	
40051	R/W	SHORT	Running Current (CC)	N
40052	R/W	SHORT	Idle Current (CI)	
40053	R/W	SHORT	Steps per Revolution	R
40054	R/W	SHORT	Pulse Counter	S
40055	R/W	SHORT	Time Stamp	W
40056	R/W	SHORT	Analog Position Gain (AP)	X
40057	R/W	SHORT	Analog Threshold (AT)	Y
40058	R/W	SHORT	Analog Offset (AV	Z
40059..60	R/W	LONG	Accumulator	0
40061..62	R/W	LONG	User Defined	1
40063..64	R/W	LONG	User Defined	2
40065..66	R/W	LONG	User Defined	3
40067..68	R/W	LONG	User Defined	4
40069..70	R/W	LONG	User Defined	5
40071..72	R/W	LONG	User Defined	6
40073..74	R/W	LONG	User Defined	7
40075..76	R/W	LONG	User Defined	8
40077..78	R/W	LONG	User Defined	9
40079..80	R/W	LONG	User Defined	:
40081..82	R/W	LONG	User Defined	;

40083..84	R/W	LONG	User Defined	<
40085..86	R/W	LONG	User Defined	=
40087..88	R/W	LONG	User Defined	>
40089..90	R/W	LONG	User Defined	?
40091..92	R/W	LONG	User Defined	@
40093..94	R/W	LONG	User Defined	[
40095..96	R/W	LONG	User Defined	\
40097..98	R/W	LONG	User Defined	]
40099..100	R/W	LONG	User Defined	^
40101..102	R/W	LONG	User Defined	_
400103..104	R/W	LONG	User Defined	`
40105	R/W	SHORT	Brake Release Delay	
40106	R/W	SHORT	Brake Engage Delay	
40107	R/W	SHORT	Idle Current Delay	
40108	R/W	SHORT	Hyperbolic Smoothing Gain	
40109	R/W	SHORT	Hyperbolic Smoothing Phase	
40110	R/W	SHORT	Analog Filter Gain	
40111..124			(reserved)	
40125	R/W	SHORT	Command Opcode	
40126	R/W	SHORT	Parameter 1	
40127	R/W	SHORT	Parameter 2	
40128	R/W	SHORT	Parameter 3	
40129	R/W	SHORT	Parameter 4	
40130	R/W	SHORT	Parameter 5	

13.7 Command Opcode 命令操作码

寄存器 40125 被定义为 Command Opcode，对 40125 写入下表所示的操作码，驱动器将执行对应的功能。

1） SCL 编码表 SCL Command Encoding Table

SCL 指令编码对应表							
Function 功能	SCL 指令	Opcode 操作码	Parameter1 参数 1	Parameter2 参数 2	Parameter3 参数 3	Parameter4 参数 4	Parameter5 参数 5

Alarm Reset	AX	0xBA	×	×	×	×	×
Start Jogging	CJ	0x96	×	×	×	×	×
Stop Jogging	SJ	0xD8	×	×	×	×	×
Encoder Function	EF	0xD6	0,1,2 or 6	×	×	×	×
Encoder Position	EP	0x98	Position	×	×	×	×
Feed to Double Sensor	FD	0x69	I/O Point 1	Condition 1	I/O Point 2	Condition 2	×
Follow Encoder	FE	0xCC	I/O Point	Condition	×	×	×
Feed to Length	FL	0x66	×	×	×	×	×
Feed to Sensor with Mask Distance	FM	0x6A	I/O Point	Condition	×	×	×
Feed and Set Output	FO	0x68	I/O Point	Condition	×	×	×
Feed to Position	FP	0x67	×	×	×	×	×
Feed to Sensor	FS	0x6B	I/O Point	Condition	×	×	×
Feed to Sensor with Safety Distance	FY	0x6C	I/O Point	Condition	×	×	×
Jog Disable	JD	0xA3	×	×	×	×	×
Jog Enable	JE	0xA2	×	×	×	×	×
Motor Disable	MD	0x9E	×	×	×	×	×
Motor Enable	ME	0x9F	×	×	×	×	×
Seek Home	SH	0x6E	I/O Point	Condition	×	×	×

Set Position	SP	0xA5	Position	×	×	×	×
Filter Input	FI	0xC0	I/O Point	Filter Time	×	×	×
Filter Select Inputs	FX	0xD3	×	×	×	×	×
Step Filter Freq	SF	0x06	Freq	×	×	×	×
Analog Deadband	AD	0xD2	0.001 V	×	×	×	×
Alarm Reset Input	AI	0x46	Function ('1'..'3')	I/O Point	×	×	×
Alarm Output	AO	0x47	Function ('1'..'3')	I/O Point	×	×	×
Analog Scaling	AS	0xD1	×	×	×	×	×
Define Limits	DL	0x42	1..3	×	×	×	×
Set Output	SO	0x8B	I/O Point	Condition	×	×	×
Wait for Input	WI	0x70	×	×	×	×	×
Queue Load & Execute	QX	0x78	1..12	×	×	×	×
Wait Time	WT	0x6F	0.01 sec	×	×	×	×
Stop Move, Kill Buffer	SK	0xE1	×	×	×	×	×
Stop Move, Kill Buffer, Normal Decel	SKD	0xE2	×	×	×	×	×

详细的指令所对应的功能，请参考 Host Command Reference 手册。

2) **数字输入/输出口的选择以及 I/0 口的状态**

字母	十六进制	含义

‘0’	0x30	编码器零点
‘1’	0x31	输入或输出口1
‘2’	0x32	输入或输出口2
‘3’	0x33	输入或输出口3
‘4’	0x34	输入或输出口4
‘L’	0x4C	低电平（光耦导通）
‘H’	0x48	高电平（光耦开路）
‘R’	0x52	上升沿
‘F’	0x46	下降沿

例如：在 MOONS 产品 SCL 指令中“FS1F”；在 MODBUS RTU 里，执行写入寄存器 40125=0x6B，40126=0x31,40127=0x46，即执行相应的动作。

13.8 MOONS 产品 MODBUS RTU 案例

13.8.1 位置控制例程

MOONS SCL 指令：

SCL 指令	数值	单位	RTU 寄存器地址	寄存器十六进制格式	说明
AC	100	Rps/sec	40028	00 1B	预设置加速度为 100，需要对寄存器 40028 写入 600（0x0258）.
DE	100	Rps/sec	40029	00 1C	预设置减速度为 100，需要对寄存器 40029 写入 600(0x0258).
VE	1	Rps	40030	00 1D	预设置速度为 1，需要对寄存器 40030 写入 240(0x00F0).
DI	200000	Counts	40031,40032	00 1E，00 1F	预设定目标位置为 200000，需要对 32 寄存器写入 200000（0x00030D40）。

注意：看如下报文前，请先阅读附件 I 的报文格式。

预写入规划曲线的加速度（40028）=600（0x0258），减速度（40029）=600（0x0258），速度（40030）=240(0x00F0)，目标位置（40032,40031）=200000（0x00030D40），注意 PR 模式不同，其 32 位寄存器输入数据的高低位区别，其报文如下：

若 PR=133 模式下：

写入报文：*01 10 00 1B 00 05 0A 02 58 02 58 00 F0 00 03 0D 40 CD 83*

反馈报文：*01 10 00 1B 00 05 70 0D*

若 PR=5 模式下：

写入报文：*01 10 00 1B 00 05 0A 02 58 02 58 00 F0 0D 40 00 03 7B 9A*

反馈报文： *01 10 00 1B 00 05 70 0D*

写入报文讲解如下：

报文：	01	10	00 1B	00 05	0A	01 F4	01 F4	00 F0	00 03	0D 40	BC 94
说明	地址	功能码	起始地址寄存器	寄存器的数量	总的BYTE位数	写入第一寄存器数据	写入第二寄存器数据	写入第三寄存器数据	写入第四寄存器数据	写入第五寄存器数据	CRC 校验（从地址到第 5 寄存器数据）

成功反馈报文讲解如下：

报文：	01	10	00 1B	00 05	70 0D
说明：	地址	功能码	起始寄存器地址	写入寄存器的数量	CRC 校验码

预写入命令操作码寄存器（40125）=0x0066，即执行相对位置控制，其报文如下：

写入报文：*01 06 00 7C 00 66 C8 38* //**预对寄存器 40125 操作，需要通过如下方式转换成十六进制，即将 40125-4000-1=124，十进制 124 的十六进制表示方式为 0x007C**//

写入报文讲解如下：

报文：	01	06	00 7C	00 66	C8 38
说明：	地址	功能码	寄存器地址	写入数据	CRC 校验码（从地址位到数值位）

预写入命令操作码寄存器（40125）=0x0067，即执行绝对位置控制，其报文如下：

写入报文：*01 06 00 7C 00 67 09 F8*

预写入命令操作码寄存器（40125）=0x00E1，即执行停止控制，其报文如下：

写入报文：01 06 00 7C 00 E1 88 5A // **以最大减速度停止，ACL 指令 SK**//

预读取目标位置寄存器（40032,40031），其报文如下：

读取报文：01 03 00 1E 00 02 A4 0D //**读取 RTU 寄存器 40031，40032**//

备注：

1) MODBUS RTU 报文读写时，注意寄存器地址间的转换关系，如寄存器 40125 转换后为 0x00 7C，即 40125-40000-1=124（0x7C）.
2) PR=133 模式和 PR=5 模式的区别：

为了将目标位置 DI=200000 写入目标位置寄存器（40032，40031），即向 32 位寄存器 地址写入 200000(0x030D40)。

- 在 PR=133 模式下，表示先写入 32 位寄存器为低端位地址优先：

写入报文：01 10 00 1E 00 02 04 0D 40 00 03 30 56 (DI=200000) (PR=133 模式下)

报文：	01	10	00 1E	00 02	04	0D 40	00 03	30 56
说明：	地址	功能码	起始地址	寄存器数量	BYTE 位总数	写入第一寄存器数据（即 40031）	写入第二寄存器数据（即 40032）	CRC 校验码

成功反馈报文： 01 10 00 1E 00 02 21 CE

➢ 在 PR=5 模式下，表示先写入 32 位寄存器为高端位地址优先：

写入报文：01 10 00 1E 00 02 04 00 03 0D 40 86 4F （DI=200000）（PR=5 模式下）

报文：	01	10	00 1E	00 02	04	00 03	0D 40	30 56
说明：	地址	功能码	起始地址	寄存器数量	BYTE 位总数	写入第一寄存器数据（即 40032）	写入第二寄存器数据（即 40031）	CRC 校验码

成功反馈报文： 01 10 00 1E 00 02 21 CE

13.8.2 速度控制例程

MOONS SCL 指令：

SCL 指令	设定数值	单位	RTU 寄存器地址	十六进制寄存器地址报文	写入寄存器数值	说明
JA	100	Rps/sec	40047	00 2E	600	预设置加速度为 100，需要对寄存器 40047 写入 600（0x0258）
JL	100	Rps/sec	40048	00 2F	600	预设置减速度为 100，需要对寄存器 40048 写入 600(0x0258)。
JS	10	Rps	40049	00 30	2400	预设置速度为 10，需要对寄存器 40030 写入 2400(0x0960)。

预写入规划曲线的加速度（40047）=0x0258，减速度（40048）=0x0258，速度（40030）=0x0960，其报文如下：

写入报文：01 10 00 2E 00 03 06 02 58 02 58 09 60 20 23
成功反馈报文：01 10 00 2E 00 03 E0 01

预执行启动动作，写入命令操作码寄存器（40125）=0X0096，其报文如下：
写入报文:01 06 00 7C 00 96 C8 7C //对寄存器 40125 写入 00 96//
成功反馈报文：01 06 00 7C 00 96 C8 7C

预执行停止动作，写入命令操作码寄存器（40125）=0X00D8，其报文如下：
写入报文：01 06 00 7C 00 D8 48 48 //对寄存器 40125 写入 00 D8//
成功反馈报文：01 06 00 7C 00 D8 48 48

13.8.3 内部 Q 编程控制例程

MOONS 驱动器内部 Q 程序例程，

```
        WT  2.00          //**延时 2 秒**//
        RX  2   11        //**将 11 写入用户寄存器 2.设定判断条件 1**//
        RX  3   12        //**将 12 写入用户寄存器 3.设定判断条件 2**//
        EP  0             //**编码器的为零**//
        SP  0             //**清零操作**//
LABEL3  CR  1   2         //**比较用户寄存器 1 和用户寄存器 2 的数值**//
        QJ  E   #LABEL1     //**判断两寄存器内数值是否相等,若相等,则跳转到 Lable
11**//
        CR  1   3         //**比较用户寄存器 1 和用户寄存器 3 的数值**//
        QJ  E   #LABEL2     //**判断两寄存器内数值是否相等,若相等,则跳转到 Lable
16**//
        QG  #LABEL3       //**跳转到 Lable 6**//
LABEL1  RM 4 A            //**将用户寄存器 4 的内容写入寄存器 A，寄存器 A 为位置环
                            加速度寄存器，且数值=实际加速度乘以 6**//
        RM 5 B            //**将用户寄存器 5 的内容写入寄存器 B，寄存器 B 为位置
                            环减速度寄存器，且数值=实际减速度乘以 6**//
        RM 6 V            //**将用户寄存器 6 的内容写入寄存器 V，寄存器 V 为位置
                                环速度寄存器，且数值=实际速度乘以 240**//
        RM 7 D            //**将用户寄存器 7 的内容写入寄存器 D，寄存器 D 为目标
                          位置寄存器，且 1:1 关系**//
        FP                //**执行绝对定位**//
        QG  #LABEL3       //**跳转到 Lable 6**//
```

```
LABEL2      RM 8 D          //**将用户寄存器 8 的内容写入寄存器 D，寄存器 D 为目标
                            位置寄存器，且 1:1 关系**//
            FP              //**执行绝对定位**//
            QG #LABEL3      //**跳转到 Lable 6**//
```

程序变量对照表

功能	映射用户寄存器	RTU 寄存器地址	预设曲线参数	单位	用户寄存器写入数值	说明
加速度	4	40067,40068	100	Rps/sec	600	设定值=加速度*6
减速度	5	40069，40070	100	Rps/sec	600	设定值=减速度*6
速度	6	40071，40072	1	Rps	240	设定值=速度*240
第一目标位置	7	40073，40074	200000	Counts	200000	1:1 关系
第二目标位置	8	40075，40076	-200000	Counts	-200000	/
操作码	1	40061，40062				判断，执行

MOONS 兼容 MODBUS RTU 驱动器设置

其他设置如上所述，但需要应用驱动器内部 Q 编程功能，需要设置 PM=9，即驱动器上电，自动运行 Q 程序。

MODBUS 主站报文写入操作：

1） 若 PR=133 情况下:

预读取用户寄存器 1（40062,40061），2（40064,40063）和 3(40066,40065)中的数据，其报文如下：

读取报文：01 03 00 3C 00 06 05 C4

反馈报文：01 03 0C 00 00 00 00 00 0B 00 00 00 0C 00 00 E9 B3

预写入用户寄存器 4(40068,40067)=600，用户寄存器 5(40070,40069)=600（0x00000258），用户寄存器 6(40072,40071)=240(0x000000F0)，用户寄存器 7(40074,40073)=200000(0x00030D40)，用户寄存器 8=-200000(0xFFFCF2C0)，其报文如下：

写入报文：01 10 00 42 00 0A 14 02 58 00 00 02 58 00 00 00 F0 00 00 0D 40 00 03 F2 C0 FF FC DC FC

预写入用户寄存器 1(40062,40061)=11（0x000B）,驱动器内部 Q 程序判断，若寄存器 1 内数据=寄存器 2 内数据，则运行到目标位置为 200000，其报文如下：

写入报文： 01 10 00 3C 00 02 04 00 0B 00 00 81 2C

或者： 01 06 00 3C 00 0B 08 01 //**写入 40061**//

预写入用户寄存器 1(40062,40061)=12（0x0000C），驱动器内部 Q 程序判断，若寄存器 1 内数据=寄存器 3 内数据，则运行到目标位置为-200000，其报文如下：

写入报文：01 06 00 3C 00 0C 49 C3 //**写入 40061**//

预读取用户寄存器 4(40068,40067)的数据，Q 程序可将用户寄存器 4 的内容写入到加速度寄存器 A 中，其报文如下：

读取报文：01 03 00 42 00 02 64 1F

预读取用户寄存器 6(40072,40071)的数据，Q 程序可将用户寄存器 6 的内容写入到速度寄存器 V 中，其报文如下：

读取报文：01 03 00 46 00 02 25 DE

预读取用户寄存器 7（40074,40073）的数据，Q 程序可将用户寄存器 7 的内容写入到位置寄存器 D 中，其报文如下：

读取报文：01 03 00 48 00 02 44 1D

预读取用户寄存器 8（40076,40075）的数据，Q 程序可将用户寄存器 8 的内容写入到位置寄存器 D 中，其报文如下：

读取报文：01 03 00 4A 00 02 E5 DD

读取状态编码寄存器（40002），可显示驱动器的运行状态，例如若反馈信息为 0x4001，表示 Q 程序在运行且驱动器处于使能状态，详情请见“Host Command Refrence 手册”，其报文如下：

读取报文:01 03 00 01 00 01 D5 CA

反馈报文：01 03 02 40 01 48 44

读取 Q 程序行运行位置状态（40018），其报文如下：

读取报文：01 03 00 11 00 01 D4 0F

读取规划曲线的加速度（40028），减速度（40029），速度（40030），目标位置（40032,40031），其报文如下：

读取报文：01 03 00 1B 00 05 F5 CE

2） 若 PR=5 情况下：

预读取用户寄存器 1（40062,40061），2（40064,40063）和 3(40066,40065)中的数据，其报文如下：

读取报文：01 03 00 3C 00 06 05 C4

反馈报文：01 03 0C 00 00 00 00 00 00 00 0B 00 00 00 0C 36 B4

预写入用户寄存器 4(40068,40067)=600，用户寄存器 5(40070,40069)=600（0x00000258），用户寄存器 6(40072,40071)=240(0x000000F0)，用户寄存器 7(40074,40073)=200000(0x00030D40)，用户寄存器 8=-200000(0xFFFCF2C0)，其报文如下：

写入报文：01 10 00 42 00 0A 14 00 00 02 58 00 00 02 58 00 00 00 F0 00 03 0D 40 FF FC F2 C0 BF 30

预写入用户寄存器 1(40062,40061)=11（0x000B），驱动器内部 Q 程序判断，若寄存器 1 内数据=寄存器 2 内数据，则运行到目标位置为 200000，其报文如下：

写入报文： 01 10 00 3C 00 02 04 00 00 00 0B B1 29

预写入用户寄存器 1(40062,40061)=12（0x0000C），驱动器内部 Q 程序判断，若寄存器 1 内数据=寄存器 3 内数据，则运行到目标位置为-200000，其报文如下：

写入报文：01 10 00 3C 00 02 04 00 00 00 0C F0 EB

或者：01 06 00 3D 00 0C 18 03 //**写入 40062**//

13.9 附件Ⅰ：功能码报文格式

13.9.1 功能 03 读取保持寄存器：

查询报文：

QUERY	
Field Name	Example (Hex)
Slave Address	11
Function	03
Starting Address Hi	00
Starting Address Lo	6B
No. of Points Hi	00
No. of Roints Lo	03
Error Check (LRC or CRC)	—

响应报文：

RESPONSE	
Field Name	Example (Hex)
Slave Address	11
Function	03
Byte Count	06
Data Hi (Register 40108)	02
Data Lo(Register 40108)	2B
Data Hi(Register 40109)	00
Data Lo(Register 40109)	00
Data Hi(Register 40110)	00
Data Lo(Register 40110)	64
Error Check (LRC or CRC)	—

13.9.2 功能码 04 读取输入寄存器

查询报文：

QUERY	
Field Name	Example (Hex)
Slave Address	11
Function	04
Starting Address Hi	00
Starting Address Lo	08
No. of Points Hi	00
No. of Roints Lo	01
Error Check (LRC or CRC)	—

响应报文：

RESPONSE	
Field Name	Example (Hex)
Slave Address	11
Function	04
Byte Count	02
Data Hi(Register 30009)	00
Data Lo(Register 30009)	0A
Error Check (LRC or CRC)	—

13.9.3 功能码 06 预置单个寄存器：

查询报文：

QUERY	
Field Name	Example (Hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

反馈报文：

RESPONSE	
Field Name	Example (Hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

13.9.4 功能码 16（10 HEX）预置多个寄存器

查询报文：

QUERY

Field Name	Example (Hex)
Slave Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Byte Count	04
Data Hi	00
Data Lo	0A
Data Hi	01
Data Lo	02
Error Check (LRC or CRC)	—

反馈报文：

PESPONSE

Field Name	Example (Hex)
Slave Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Error Check (LRC or CRC)	—

13.10 附件Ⅱ：MODBUS RTU 不正常响应及代码

主机发出查询，从机不正常响应。（为十六进制）。

QUERY

Byte	Contents	Example
1	Slave Address	0A
2	Function	01
3	Starting Address Hi	04
4	Starting Address Lo	A1
5	No. of Coils Hi	00
6	No. of Coils Lo	01
7	LRC	4F

EXCEPTION RESPONSE

Byte	Contents	Example
1	Slave Address	0A
2	Function	81
3	Exception Code	02
4	LRC	73

上例中，从站设备地址 10（0A HEX），读线圈状态的功能代码（01），主机请求线圈的地址 1245(04A1 HEX)，读取线圈数量为一个（0001 HEX）.

若从机中不存在此线圈地址，即以不正常代码（02），向主机返回一个不正常的响应。说明不合法地址。

不正常代码表：

代码	名称	含义
01	不合法功能代码	从机接收的是一种不能执行功能代码。发出查询命令后，该代码指示无程序功能。
02	不合法数据地址	接收的数据地址，是从机不允许的地址。
03	不合法数据	查询数据区的值是从机不允许的值。
04	从机设备故障	从机执行主机请求的动作时出现不可恢复的错误。

05	确认	从机已接收请求处理数据，但需要较长的处理时间，为避免主机出现超时错误而发送该确认响应。主机以此再发送一个“查询程序完成”未决定从机是否已完成处理。
06	从机设备忙碌	从机正忙于处理一个长时程序命令，请求主机在从机空闲时发送信息。
07	否定	从机不能执行查询要求的程序功能时，该代码使用十进制 13 或 14 代码，向主机返回一个“不成功的编程请求”信息。主机应请求诊断从机的错误信息。
08	内存奇偶校验错误	从机读扩展内存中的数据时，发现有奇偶校验错误，主机按从机的要求重新发送数据请求。

14 Position Table 位置表模式

位置表模式使用外部输入信号 X7~X12 选择多位置中的一组作为位置指令，并使用位置触发信号 X4 触发该位置。无需上位机发送脉冲即可实现点到点的直线运动或者圆周运动。图 13.10.1 位置表模式

注意：只有-S 型的 M2 系列交流伺服才支持位置表模式。

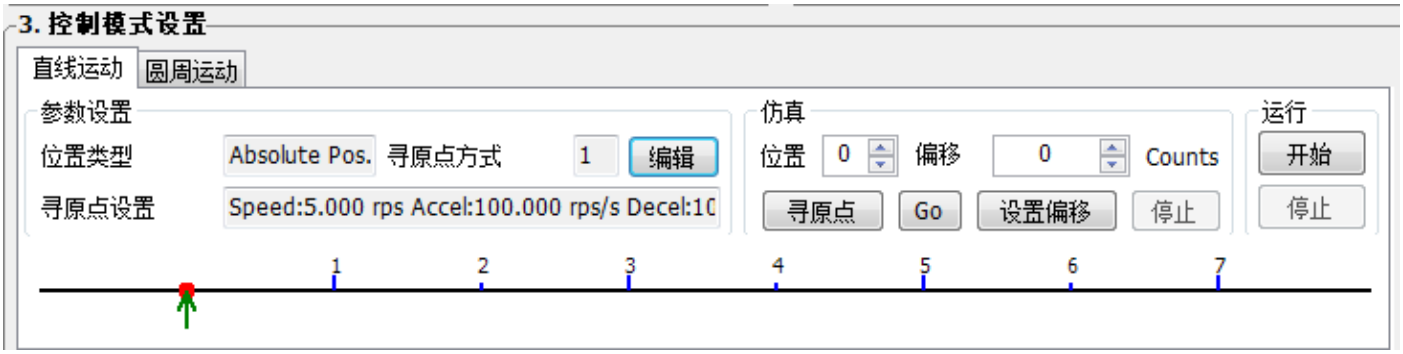


图 13.10.1 位置表模式

14.1 直线运动

位置表模式下的直线运动可设定多达 63 点位置（不包括原点）。其“软件设定流程”如下。

14.1.1 直线运动软件设置流程

1) 打开 M Servo Suite，与驱动器建立通讯连接（通讯连接步骤请参考软件使用手册），在***第 1 步：配置***界面，将“***2.控制模式***”中的主模式选择为“***Position Table***”。如图 14.1 选择 Position Table

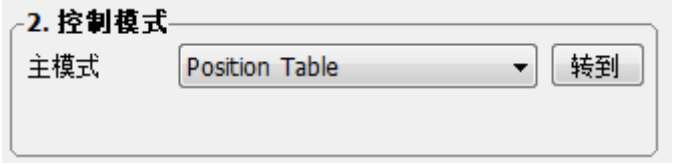


图 14.1 选择 Position Table

2) 在 3.控制模式设置中选择直线运动，如图 14.2 直线运动设置

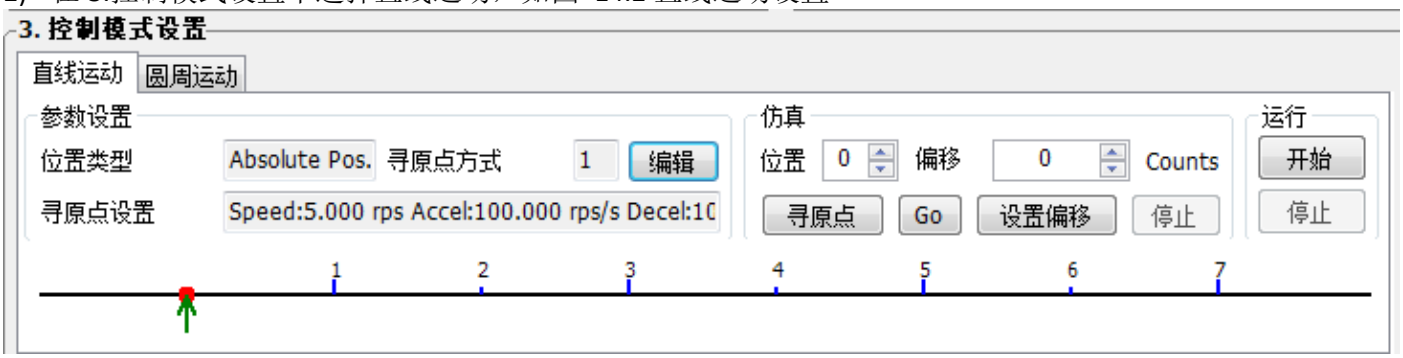


图 14.2 直线运动设置

3) 点击“编辑”按钮进行详细设置。如下图 14.3 直线运动详细设置。

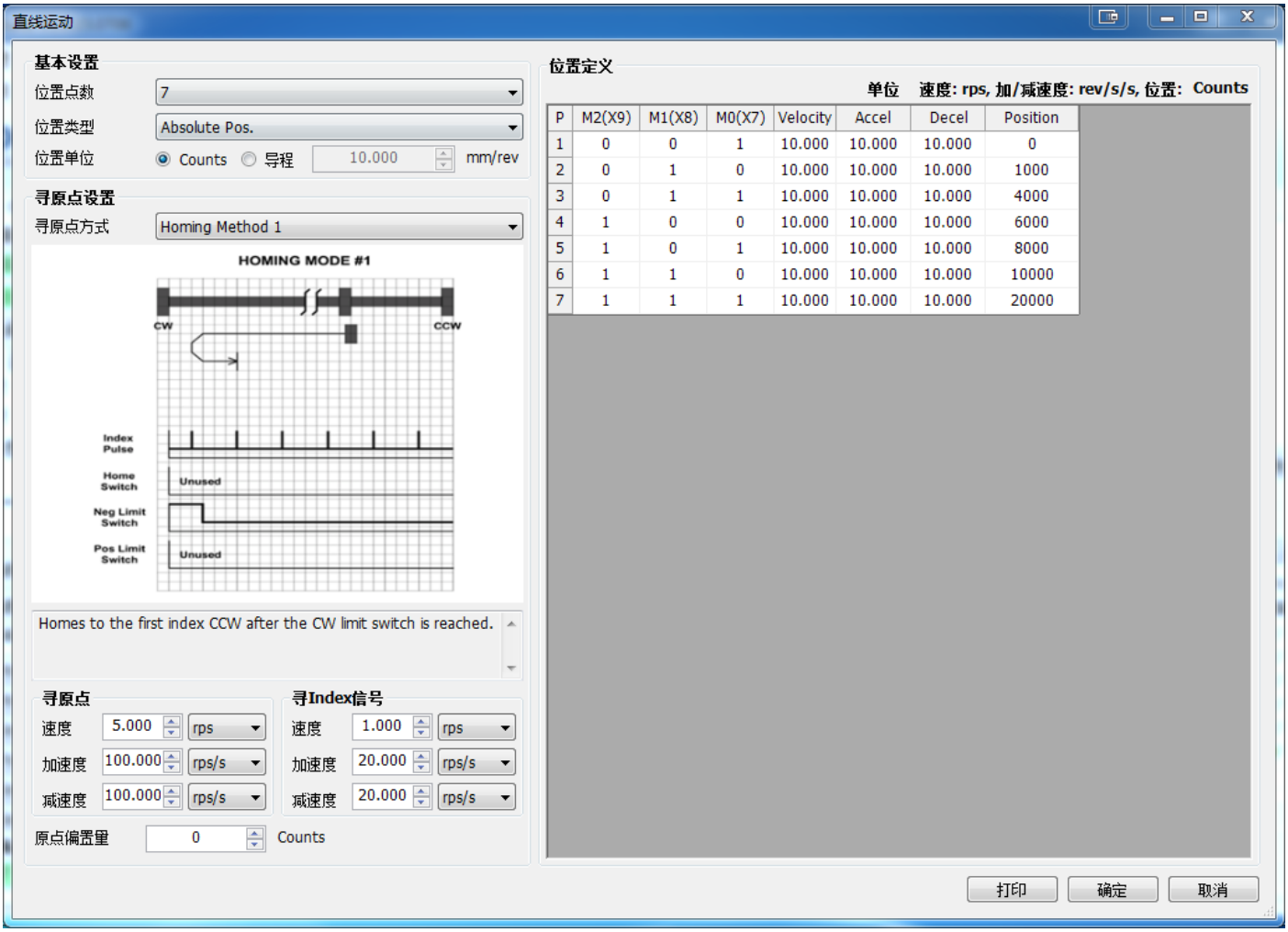


图 14.3 直线运动详细设置

基本设置

位置点数：选择需要的定位点数。软件提供了 7、15、31、63 个位置点数供选择。

位置类型：选择点到点定位方式为相对运动（Relative Pos.），绝对运动（Absolute Pos.）

假设 P1 位置是 5 圈，P2 位置是 10 圈，相对运动和绝对运动的区别如下：

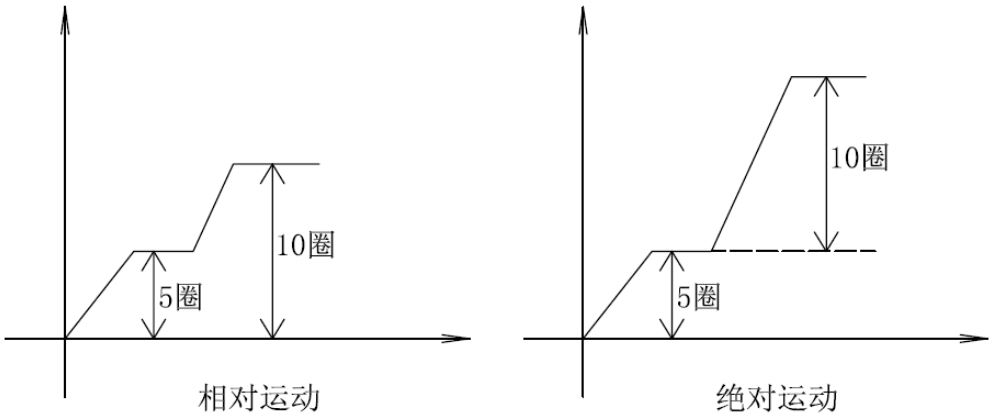


图 14.4 相对运动、绝对运动对比

位置单位：选择点位位置单位。

Counts：代表电机编码器脉冲数，电机每旋转一圈脉冲数为 10000counts。

导程： 设定电机每转一圈机构前进距离，单位为毫米每圈，mm/rev

寻原点设置

寻原点方式：目前提供了 12 种回原点方式。回原点方式详细介绍请参考 7.5.3 回原点方式章节。

寻原点：

此设置界面设定寻原点的第一段速度、加速度及减速度。

寻 Index 信号

此设置界面设定当碰到原点开关后，寻找电机编码器 Index 信号的速度、加速度及减速度。

原点偏移量：设定完成寻原点动作后，原点的偏移量。

打印

点击“打印”按钮可以将上述的配置参数打印出来，方便编程使用。如下图所示

Position Table Configuration. ©Shanghai AMP & MOONS' Automation Co., Ltd.
直线运动
位置点数: 7
位置类型: Absolute Pos.
寻原点方式: Homing Method 1
寻原点
速度: 5 rev/s 加速度: 100 rev/s/s 减速度: 100 rev/s/s
寻Index信号
速度: 5 rev/s 加速度: 100 rev/s/s 减速度: 100 rev/s/s
原点偏置量: 0

位置定义

				单位	速度: rps, 加/减速度: rev/s/s, 位置:		
P	M2(X9)	M1(X8)	M0(X7)	Velocity	Accel	Decel	Position
1	0	0	1	10.000	10.000	10.000	10000
2	0	1	0	10.000	10.000	10.000	20000
3	0	1	1	10.000	10.000	10.000	40000
4	1	0	0	10.000	10.000	10.000	60000
5	1	0	1	10.000	10.000	10.000	80000
6	1	1	0	10.000	10.000	10.000	100000
7	1	1	1	10.000	10.000	10.000	200000

图 14.5 打印 Position Table 参数表

位置定义

位置定义列表中可以分别设定每个位置点所对应的距离、速度、加速度、减速度。同时也显示了每个位置点对应的位置选择引脚 X7~X12 电平状态。

单位 速度: rps, 加/减速度: rev/s/s, 位置: Counts

P	M4(X11)	M3(X10)	M2(X9)	M1(X8)	M0(X7)	Velocity	Accel	Decel	Position
1	0	0	0	0	1	10.000	10.000	10.000	10000
2	0	0	0	1	0	10.000	10.000	10.000	20000
3	0	0	0	1	1	10.000	10.000	10.000	40000
4	0	0	1	0	0	10.000	10.000	10.000	60000
5	0	0	1	0	1	10.000	10.000	10.000	80000
6	0	0	1	1	0	10.000	10.000	10.000	100000
7	0	0	1	1	1	10.000	10.000	10.000	200000
8	0	1	0	0	0	10.000	100.000	100.000	14000
9	0	1	0	0	1	10.000	100.000	100.000	16000
10	0	1	0	1	0	10.000	100.000	100.000	18000
11	0	1	0	1	1	10.000	100.000	100.000	20000
12	0	1	1	0	0	10.000	100.000	100.000	22000
13	0	1	1	0	1	10.000	100.000	100.000	24000
14	0	1	1	1	0	10.000	100.000	100.000	26000
15	0	1	1	1	1	10.000	100.000	100.000	28000
16	1	0	0	0	0	10.000	100.000	100.000	30000

图 14.6 位置定义表

M0(X7) ~ M5(X12)的状态：0 代表该输入点导通（closed）；1 代表该输入点开路（Open）。

在完成回原点动作后，当 X4 Position Triger 由 Open 变成 Closed 时，驱动器将运行到由 M0(X7) ~ M5(X12)所选择的位置点。

4) 点击“确认”按钮完成直线运动的详细设置。
5) 点击“下载到驱动器”按钮，确认配置。
6) 点击“开始”按钮或关闭软件，重新给驱动器上电，驱动器进入位置表工作模式。

14.1.2 仿真操作

当完成上述设置，可以使用仿真功能模式确认电机运行到对应的点位。



图 14.7 直线运动仿真

寻原点：点击“寻原点“按钮开始寻原点。

Go：在“位置”选择框选择位置点，点击“Go”按钮，即可运动到该点位。

图 14.7 直线运动仿真中标识②中的绿色箭头将同时移动到对应的点位。

设置偏移：确认“偏移”的位置量。点击此按钮将修改位置定义表中的位置信息。

停止：点击本按钮，电机停止运行。

14.1.3 直线运动输入引脚定义

直线运动输入引脚定义如下：

输入引脚	功能定义	描述
X1	Homing Sensor	原点开关
X2	Homing Triger	回原点动作触发输入信号
X3	General Purpose	通用输入，无功能
	Servo On When Closed	当输入信号 Closed 导通时，伺服使能
	Servo On When Open	当输入信号为 Open 开路时，伺服使能
X4	Position Triger	位选触发信号。 当该信号从 Open 变成 Closed 时，电机将运行到由 M0(X7) ~ M5(X12)所选择的位置点。
X5	General Purpose	通用输入，无功能
	CW Limit Senser	正转限位开关，具体功能设定请参考 M2 用户手册 7.1.3 正反转限位章节。
X6	General Purpoose	通用输入，无功能
	CCW Limit Sensor	反转限位开关，具体功能设定请参考 M2 用户手册 7.1.3 正反转限位章节。
X7~X12	M Input	位置点选择输入。

14.2 圆周运动

圆周运动特别适用于分度盘的应用，只需要输入外部机械减速比及等分分度比，即可轻松完成设定。

驱动器将根据触发信号 X4，按照设定好的方向，驱动电机带动转盘实现点到点的等分运转。

14.2.1 圆周运动的软件设置界面

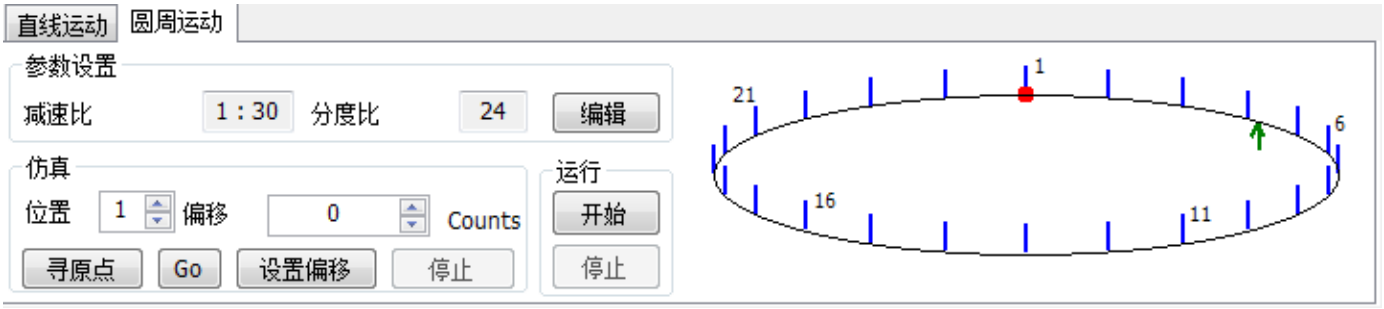


图 14.1 圆周运动

减速比：设定外部机械总的减速比。

分度比：设定所需要的等分。

编辑：点击“编辑”按钮可以进入详细设置界面（图 14.2 圆周运动详细设置）。

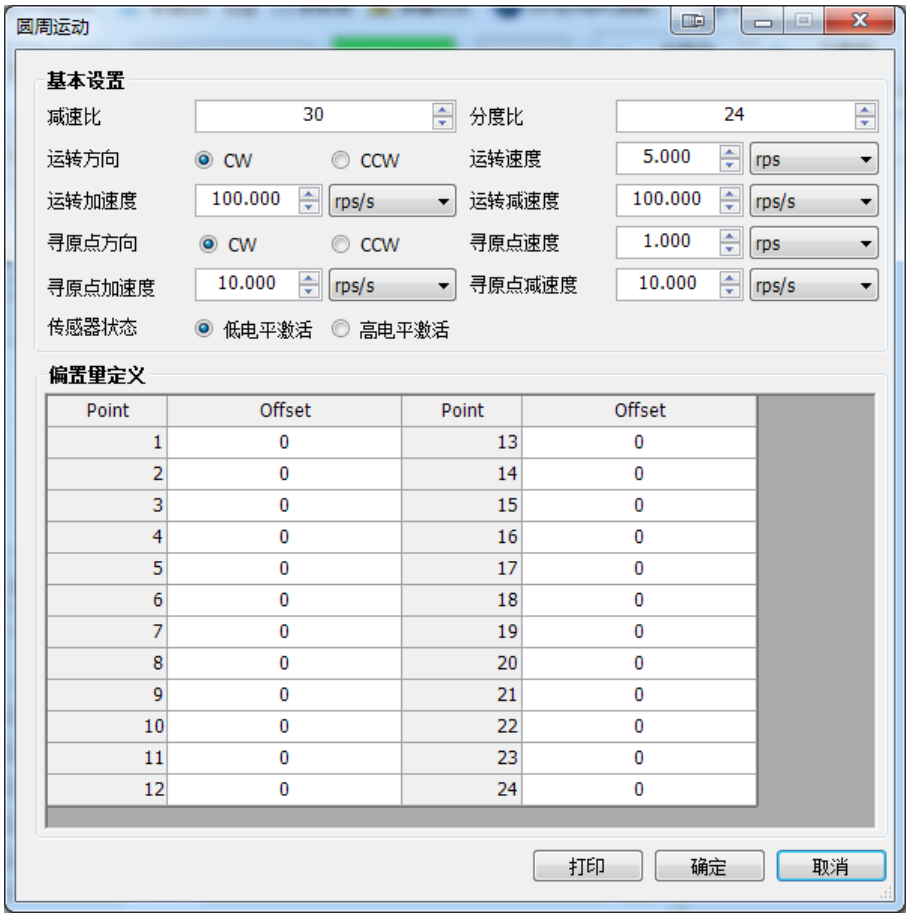


图 14.2 圆周运动详细设置

运转方向：选择转盘运转方向。

运转速度、运转加速度、运转减速度：设定电机运转的速度、加速度、减速度。

寻原点方向：设点转盘寻原点传感器的方向。

寻原点速度、寻原点加速度、寻原点减速度：设定寻原点动作时，电机的速度、加速度、减速度。

传感器状态：设点原点传感器的类型：低电平激活，高电平激活

偏置量定义： 设定每一个等分点的偏移量，显现位置微调的功能。

14.2.2 圆周运动输入引脚定义

圆周运动输入引脚定义如下：

输入引脚	功能定义	描述
X1	Homing Sensor	原点开关
X2	Homing Trigger	回原点动作触发输入信号
X3	General Purpose	通用输入，无功能
	Servo On When Closed	当输入信号 Closed 导通时，伺服使能
	Servo On When Open	当输入信号为 Open 开路时，伺服使能
X4	Position Trigger	位选触发信号。 当该信号从 Open 变成 Closed 时，电机将运行到下一个等分位。