

# Mis Programmer 软件使用手册



适用于 MS1240MI, MS3540MI, MS7080MI





目录
----

开始	3
安装编程软件	4
驱动器与 PC 的连接	5
如果 PC 机没有串口怎么办?	5
编程	6
编写程序	7
复制指今	q
反的用 <	10
田之中刷你住厅少森	10
用厂日走入半世	10
Idle Current Reduction(待机电流缩减)	12
细分(仅限于步进电机)	12
JOGGING(点动)	13
配置输入对话框	14
中断	15
快速制动	16
限位开关	17
触动限制开关时会发生什么?	17
面板错误代码	18
庙田코迁的人和奥西(MAAI)	10
这小头拍的八秒的声面(MMM)	13
MMI Promot	25
Feed To Length	
Feed & Set Output	29
Feed & Return	30
Feed To Sensor	31
Feed To Sensor & Return	33
Feed To Position	34
Set Abs Posistion	34
Save Abs Position	35
Wait Time	
Wait Input	38
Hand Wheel	39
Go To	
	40
Set Output	42
If Input Go To	43
Call and Return	
Change Current	47
Comment	48
COMMAND BUTTONS	49
Download, Upload & Execute	49
Save, Open, Print & Quit	50
ENCODER FEEDBACK: 带编码器选项权的 MS3540MI	51

### 开始

感谢您购买鸣志公司的产品。我们期待我们的产品在性能、价格以 及易于使用等方面为您的应用系统带来最佳的选择。

Mis Programmer 软件应用于鸣志电机产品,包括: MS1240MI, MS3540MI 以及 MS7080MI。本使用手册讲解怎样安装并使用 Mis Programmer Windows 软件。

要查看关于 Mis 系列驱动器的详细信息,请阅读产品附带的硬件使 用手册。在随机附带的 CD 及我们的网站(http://www.moons.com.cn) 上也有相应的硬件手册下载。

Mis Programmer 有如下特性:

- 功能强大,使用灵活。
- •程序可保存在 PC 中,供用户调用。
- •专用电缆与 PC 连接对驱动器进行编程。
- •程序下载后可永久保存在驱动器中。
- 可编程的输入/输出用于实现与其它设备的连接。
- •软件支持输入信号触发中断,行程信号触发跳转。
- •指令中包含电机运转、触发、跳转、循环、子程序调用、延时等。
- •用户自定义单位,例如毫米、度、升等。
- 灵活的人机界面(MMI)允许操作员输入距离、速度、循环次数等 参数。

运行 Mis 产品有如下几个步骤:

- 安装编程软件。
- 连接电机
- 连接输入/输出端及编程电缆。
- 编写程序。
- 接通电源, 下载并调试程序
- •关闭电源、断开编程电缆
- 重新上电,程序运行

如果您对我们的产品有任何疑问或改进意见,请致电(86)-21-52 634688或者给我们发电子邮件:info@moons.com.cn。

注意:本手册适用于Mis Programmer软件版本V2.5以上及最新版本的驱动器。如果您使用的驱动器低于这个版本,那么本使用手册上的某些特性可能不适用。Mis Programmer 软件会警告您如果要尝试下载软件,那么您的驱动器可能无法运行。

如果您的 PC 上已经安装旧版本的 Mis Programmer 软件,您可以 安装新版本的软件。您不必卸载旧版本的软件,最新版本的 Mis Programmer 软件可以和所有旧版软件同时存在。



打开电源: 程序自动运行

### 安装编程软件

Mis Programmer软件随同相关使用手册和其它软件在一张CD上。使用前必须将它安装在您的硬盘上。

Mis Programmer 软件要求计算机的最低配置如下:

- IBM 386,486 兼容机,奔腾及以上 CPU。推荐使用奔腾系列。
- Microsoft Windows 2000, ME, NT 或者 XP 操作系统。
- 最少 32M 内存。
- 20M 可用硬盘空间。
- CD-ROM .
- 需要带有 9 针的串口,一般来讲是 COM1。如果计算机没有串口,可以购买一个 USB 串口适配器。

就象其它 Windows 程序一样, Mis 软件的安装基本上是自动进行的, 安装过程非常简单:

- •将光盘放入 CD-ROM。
- 等待弹出一个 CD 浏览器。
- 选择 "Install Software"。
- 在下一个界面中,选择"Mis Programmer"。

您也可以从鸣志公司的网站上(www.moons.com.cn)下载软件和使用手册。

#### 编程提示:

- 1. 在 Mis Programmer 软件运行的时候请不要关闭 Mis 硬件的电源
- 在下载程序到驱动器的时候,请确认驱动器上的 JOG 输入端处于非活动状态。如果无法确认,请断 开驱动器上 JOG CW 和 JOG CCW 与其它设备的连接。

编程软件上有几个示例程序。加载几个示例并看看是一个不错的主意,这会对你自己的系统有所帮助。

### 驱动器与 PC 的连接

驱动器与 PC 的连接必须采用鸣志公司提供的专用编程电缆,否则可能会导致驱动器损坏。如需定制 编程电缆,请与本公司联系。

你还需要设置 Windows 软件的 COM 口。软件运行后会自动查找可用的 COM 口,但并不一定是你接上的那个端口。

你可以点击 "COM port"选项中的其中一个单选框来改变端口。 如果选中的端口存在而且未被占用的话,编程软件会与 Mis 进行通讯。

### 如果 PC 机没有串口怎么办?

如果您的 PC 机上没有串口,请购买"USB 串口转换器"。



### 编程

Mis Programmer 软件允许您设置电机的电流、细分、点动参数、限位开关极性。也允许您编写复杂的电机控制和交互程序。

Mis Programmer 程序的最大容量是 100 行。这样的容量你可以设计一个多个电机和机器控制的程序。 Mis Programmer 的命令或指令超过 20 个。

六个电机运转指令中 Feed To Length、Feed & Set Output、Feed & Return 为固定距离运转。Feed To Position 指令为绝对位置运转。Feed To Sensor 指令和 Feed To Sensor & Return 指令根据已连接某个输入端的传感器信号运转。Seek Home 为查找定位原点传感器指令。

两个定时指令: Wait Time 指令使程序等待一段时间。Wait Input 指令等待外部触发输入信号。

五个程序流程控制指令中 Go To 使程序跳转到特定行。If Input 在某个输入满足一定的条件时跳转到 某行,否则继续执行下面的行。Repeat 和 End Repeat 在满足条件时循环执行多次。如果在循环完成之前 (使用 If Input 指令)终止循环,你可以用 Reset Repeat Loop 指令重设循环次数。

Set Output 指令用于向其它设备发送信号。

MMI(人机界面)中的 MMI Prompt 指令,允许操作员输入距离,速度和循环次数。驱动器也可以显示提示信息,中断程序直到按 ENTER 键,或者询问用户如何决定或者响应,按 YES 键或者 NO 键来完成。

Comment 指令允许您输入一些注释以增加代码的可读性。

Set Absolute Position 指令用于设置电机当前的绝对位置。

Change Current 指令用于更改驱动器的电流。

Hand Wheel 指令用于外接编码器手轮手动进行电机定位。

以不同的方式组合这些指令,你可以编写出无穷无尽的应用程序。在编写程序之前,你应该先思考怎 么完成你的目标。当你有了一个清晰的思路时你就可以输入指令和参数了。

### 编写程序

如果您参照前面的章节"安装程序软件"你就可以开始编程了。如果还没有,请回到前面的章节重新 阅读一下。

Line		Description	Clear
1	↓		
a   2	+		
3	₩		
.    4	+		
5	₩		
6	+		
_ 7	₩		
_ 8	_ ↓		
9	♦		
its 10	. ↓		
<b>1</b> 1	↓		
12	↓		
13	+		
14	₩		
15	+		
<b>2</b> 16	+		
17	♦		
	Line Line Line Line Line Line Line Line	Line Line Line Line Line Line $1  \downarrow$ $2  \downarrow$ $4  \downarrow$ $5  \downarrow$ $6  \downarrow$ $7  \downarrow$ $8  \downarrow$ $9  \downarrow$ $10  \downarrow$ $11  \downarrow$ $12  \downarrow$ $13  \downarrow$ $14  \downarrow$ $15  \downarrow$ $16  \downarrow$ $17  \downarrow$	Line Description $ \begin{array}{c ccccccccccccccccccccccccccccccccccc$

按以下顺序打开软件:开始菜单→Moons'→Mis Programmer。主编程窗体很快就会出现,如前页中显示的图所示。标题栏会显示 Mis Programmer 软件的版本。

如果**驱动器**已经和 PC 连接,请接通驱动器的电源。接通电源后计算机会"滴"一下。"Driver"下拉框会显示**驱动器**的固件版本。

如果**驱动器**还没连接到计算机,你也可以编写程序。编程前你需要在"Driver"后的列表框中选择适当的驱动器型号。这样你就可以进入你打算使用的硬件的特性。

我们来输入一个简单的程序。我们通过在首行输入 Wait Time 指令启动一个简单的延时。

在程序窗口中"1"后面的是一个 图标的按钮。表示那一行还没有指令。在任何时候驱动器在遇 到 程序行的时候就会移动到下一行,就象这个图标的下箭头那样。当驱动器执行第 100 行指令后会自 动跳转到行 1,除非第 100 行为跳转指令。

进入第1行输入指令时点击程序图标。你会看见"指令界面",如下图所示。

点击"Wait Time"按钮会出现 Wait Time 对话框。在文件框中输入 1 秒然后按确定。

这样程序的第一行会显示等待图标,显示为"Wait 1 seconds"。



Program Line 1			
Feed to Length	Feed & Return	Feed to Position	Feed/Set Output
Feed to Sensor	Feed Sensor Rtn	◆1234 Set Abs Position	Seek Home
Wait Time	Wait Input	Save Abs Positio	n MMI Prompt
لے Go To	<b>₹</b> If Input Go To	Set Output	Comment
C Repeat	End Repeat	The set Loop or In	た. Change Servo
) Hand Wheel	Call	<b>₹</b> Return	None
Insert	De	lete	Cancel

点击第二行的图标按钮。这次你会看见像"Program Line ...."的对话点击Feed To Length。在Feed To Length 对话框,输入距离为20000,然后调整度条到10转/秒,按确定。

第二行程序现在会显示电机图标,描述为"CW 20000 steps, 10 rps。"

点击第三行的图标。选择 "Go To"。当出现 Go To 对话框时,行数已经设成 1。点确定。程序现在就 象下面那样:



如果电机已经连接到驱动器,你现在就可以测试 了。

将电流设成电机的额定电流。然后点击屏幕中间 附近的下载按钮,如果驱动器已经接通电源并且连接

正确,下载条会显示进度,大概需要 1-3 秒种时间。(转换时间取决于驱动器内部重写的速度,固定内存,以及程序的大小)。程序下载完毕后你就可以执行程序了。



按执行按钮,你会看见中间的执行框。

电机会一分种转一圈。(这在你将细分设成默认的 20000 步/转时就定好了)。

复杂的程序也是照这样的方法输入的,在你的系统中 输入更多的程序行你会更多的关注这些确切参数和它们 的重要性。如果您对我们忽略的东西有一些想法,请给我 们打电话或者发传真。

注意:如果你需要修改程序中的指令,并且希望直接转到相关的对话框,按住 Shift 不放然后点击指 令图标。

#### 复制指令

有时候你会想复制一段指令,或者是复制只有一两个参数不同的指令,最快的方法就是用鼠标按住你 想复制那一行前面的图标,并将图标拖到程序中另外一行。这叫"拖拽"。

举例如下:假设你看见程序中的第5行是 Set Output 指令,你想在第11行也是同样的指令。将鼠标指向第5行的位置,点中按钮不放。移动鼠标到第11行的图标才松开鼠标,这个时候指令和参数都会复制到第11行。

#### 插入和删除程序步骤

#### Insert

#### 插入步骤

有时会出现你已经输入很多程序行,但没想到在中间需要加入指令。这样的话点击你想插入新的指令 位置的程序图标,系统照常会弹出 Program Line...对话框。在选择 20 种指令的前面点击 "Insert." 按钮。 所有的指令会重新排序,就会出现你加入的新行了。

你可以点击刚才加入的行来选择一种指令。

#### Delete

#### 删除步骤

除了在程序中删除一行外,你删除之后还可以将下面的行移上来。点击你想删除的程序行,当出现 Program Line...对话框时选择 "Delete." 就会删除你选择的那行,在这行后面的所有行都会向上面移动一 行,在程序的结束位置会留一个空行。

### 用户自定义单位

在 Mis Programmer 软件中你可以定义自己的单位。

首先要在 name 框中输入一个名称。右图中的例子中为 "inch"。输入的名称不能 超过 4 个字符。

接下来你需要输入自定义单位的电机步进的数值。例如,20,000步/转的螺距2转/英寸就是

#### (2/转英寸)(20,000步/转)=40,000步/英寸

有时螺距需要简化成英寸/转(比如 0.2),这样的话您定义的步/英寸会转换成步/转,以便和英寸/转分开。

最后确认选中"User Units"选择框。你会注意到面板上的单位由"rev/sec"变成了"inch/sec"。所 有你打开的程序指令对话框也会处理为英寸和英寸/秒。

在自定义单位的时候选择其它的 steps/rev 时要小心。如果是英制的螺距,想转换成毫米,比如将 steps/rev 设置成 25400 是很有帮助的。那么用户自定义单位对于 2 转的螺距就是

(2/转英寸)(254,000步/转) \_\_\_\_\_\_= 2000步/mm

25.4mm/英寸



**MOONS'** 

### 步进电机电流设置

驱动器的电流必须设置成和电机匹配。首先,驱动器的电流取决于电机的额定电流。如果使用的是用 户手册上推荐的驱动器,用户手册上会列出电机的额定电流。否则你就要看制造商提供的额定电流,通常 电机的额定电流会打印在电机的标签上。你只能在低于额定电流的情况下操作电机。这样会比在额定电流 的情况下力矩更小,但电机不会发热,而且噪音很小。

根据电机和驱动器的连接方法,驱动器的电流设置可以和电机的 额定电流不一样。例如,如果你使用的是鸣志电机,遵循下面的规则:

四线制电机: 使用额定电流。电机只能以一种方式连接。

**六线制电机:**标牌上的电流是终端连接的中心电流。如果您想将 电机串联起来,将电流除以 **1.4**。

**八线制电机:**标牌上的电流是终端的中心电流。串连时把这个电流乘以 1.4,并联时除以 1.4。

在 Mis Programmer 软件中这个电流由屏幕左上边的主窗体控制。要调整电流,左右滑动滚动条。点 击滚动条两边的箭头可以更精确地调节。

Current: 2.50 A/Phase					
Idle Current					
O 0%	0 25%				
• 50%	O 100%				

#### Idle Current Reduction(待机电流缩减)

驱动器具有在电机不动时自动减小电机电流的特性。这样以减少电机和驱动器的发热。例如,设置空置电流减小 50%会减少驱动器发热 50%,减少电机发热 75%。这个特性可以设置成以下四种中的任意一种: 0%、25%、50%和 100%。100%用在需要高度保持力矩的场合,驱动器就不会减小电流。0%用在不需要保持力矩的要求下。

要尽可能地减小电机和驱动器的发热,我们强烈推荐您使用待机电流缩减特性。

### 细分(仅限于步进电机)

大多数非细分型驱动器只提供整步和半步两种工作模式。整步模 式下电机的所有相位任何时间处于使用状态。半步模式下每步在全相 使用和单相使用这两个小步下交替使用。

> 细分驱动器可以精确控制每一相位每一步进位置的电流,这就意味着 更进一步地实现了电气细分。所有驱动器具有 13 种细分模式,细分模式 从 200 步/转开始。最大值将一整步分成 254 个微步,此时 1.8°的电机 细分为 50,800 步/转。

除了提供精确定位和平滑运转之外,细分驱动器还可以用于在不同的 单位之间转换。如果您正在使用 MMI(人机界面),并打算测量用户输入的 距离或速度,选择细分型驱动器就十分必要。

如果不是很确定建议您选择 20000 步/转。电机在 20000 步/转时比 更低的细分时(例如 2000)时运行更平滑,声音更小。

如果程序中有运转指令而您又想改变步进模式,软件会提醒您距离可 能需要改变(因为距离是根据步数而定的)。软件会自动按细分比例调整步 数,使绝对距离保持不变。







### Jogging(点动)

驱动器有两个输入端支持电机点动。

如果驱动器连接到运行 Mis Programmer 的 PC 上,点动输入 会在两种条件下起作用:

- 用户程序没有执行(但与 PC 相连)
- 用户程序正在执行 Wait Input 指令

驱动器在脱机模式下(比如,不和计算机相连),当用户程序正在 执行 Wait Input 指令时,点动输入才会起效。

调节如右图窗体中的滚动条,设置点动速度和点动加速度/减速度。

我们推荐加速度/减速度设置的适当值(25 转/s<sup>2</sup>),除非有很大惯量的负载,这样你需要设成更低。点动加速度的范围是从 1 到 3000 转/ s<sup>2</sup>。

点动速度的范围是从 0.025 到 50 转/秒。速度的大小根据系统的实际应用而定。

如果不需要点动,你可以把点动输入端当成通用5号和6号输入端,以使用Feed To Sensor、Wait Input 和 If Input 指令。CW JOG 输入端可以通过将"Make IN5/Jog CW a general purpose programmable input" 勾中分配成通用的输入端。In6/Jog CCW 输入也可以用作通用输入端。

Jog	Param	eters
Speed 🔽	4.000	Rev/sec
		▶
Accel	25	Rev/s/s
		▶
Config	gure Ing	puts

### 配置输入对话框

在主窗体左下方有一个"Configure Inputs"(配置输入)按钮。点击这个按钮会弹出下面的对话框。

S. Configure Inputs	×
Interrupt	Motion Output (Y2)
Input Condition Condition I C 5(cwjog) C High (open) C Low (closed)	Closed when position error is less than 10 counts
C 2 C 6(cowjog) C 3 C 7 C 4 C 8 C 1 gnore interrupt	Brake Output (Y1)
Quick Decel Rate     Branch Line       Image: State of the st	closing the Brake output     opening the Brake output     Release time 200 msec     Engage time 200 msec
End of Travel Limits         At End of Travel, Limit Switches will be Closed         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit sensor:         If a move other than Seek Home or Jog reaches a limit	Serve Fault If the serve faults: Step motion, hak pregam step motion, branch to line close fault output (Y3) open fault output (Y3) nether

#### 中断

你可以分配某一通用输入端作为中断程序跳转,不管程序在执行什么命令。例如 Mis 驱动器驱动的设备不能继续处理产品,你可能会想让 Mis 驱动器停止工作并等待。

中断在上图所示的对话框中进行设置。

第一步:选择输入端及中断条件。第二步:选择 "Branch on interrupt(中断跳转)"及程序行号。这个 行号就是中断后要转到的程序行。

#### 中断的两种类型

Mis 驱动器支持两种不同的中断: Call 中断和 Go To 中断。

#### Call 中断

Call 中断使用 "呼叫/返回" 机制, 使程序返回到 中断发生前的位置。下面是工作原理:

- 当中断输入满足定义的中断条件时驱动器会停止 当前任何指令,包括任何运动。
- 2. 驱动器将返回地址推入呼叫堆栈(Call Stack)。如 果您不清楚堆栈,请阅读 "Call And Return"。



- 程序进入中断地址。此时不能再进行中断,否则 程序会无休止地进入同一地址。如果驱动器连接到正在运行 Mis Programmer 软件的 PC,你会看见中 断条件由"启用"变成了"禁用"。
- 4. 等待中断条件清除。如果你使用 MMI, 你可能需要提醒操作员。
- 5. 一旦中断条件清除,使用 Return 指令重新开始主程序。请确认选中"Return from interrupt",而不是 "return from Subroutine"。Return 指令会重新启用中断并返回中断发生时正在运行的指令。如果这 条指令是 Feed To Sensor, Feed to PoMistion 或者 Seek Home 动作,运转会到达预定的目标。Feed To Length 运转通常会超过预定位置,因为中断后运动是重新开始的。例如,如果 6 英寸的 Feed To Length 移动在走到 3 英寸的时候发生中断,那么最终总的移动距离会是 9 英寸。
- 6. 对于 Call 中断,不要在中断处理中使用 Reset 指令,,否则会产生堆栈溢出。而且,在回到主程序之前您不能启用中断否则会在中断处理中产生中断,由此带来大型堆栈混乱(BSM)。

🐂 Reset Repeat Loop

#### Go To 中断

在很多情况下,不需要中断返回到中断前的 位置,你可能希望从程序开始的位置或者其它位 置启动。方法如下:

- 当中断输入端满足定义的中断条件时驱动 器会停止当前正在执行的任何指令,包括任 何移动。
- 2. 驱动器将返回地址推入呼叫堆栈(Call Stack)。这个堆栈我们实际上并没有用到。
- Reset Repeat Loop or Interrupt
   Reset Repeat Loop on that starts on line
   Note: you only need to use this instruction if
   your program branches out of a loop using an
   If Input or MMI Branch instruction.
   Reset Interrupt
   Cancel
   OK
   Also reset call stack
- 程序进入中断地址。此时不能再进行中断, 否则程序会无休止地进入同一地址。如果驱 动器连接到正在运行 Mis Programmer 软件的 PC,你会看见中断条件由"启用"变成了"禁用"。
- 4. 等待中断条件清除。如果你使用 MMI, 你可能需要提醒操作员。

X

5. 一旦中断条件清除,你需要用 Reset 指令重新启用中断。确认选中"Reset Interrupt",否则不会再次 产生中断。确认选中"also reset cal stack"。这样会从 Call Stack 移除中断地址而且可以防止堆栈溢 出。

**MOONS'** 

6. 使用 Go To 指令来完成中断处理。

#### 组合使用子程序和中断

Call 堆栈深度为5级,中断通常使用1级,所以在程序中可以同时使用子程序和中断,要确认在子程序中不能超过4级。这样即使4级深度的子程序调用,你仍然还有为中断留出的1级堆栈。

#### 快速制动

当发生意料之外的情况时通常需要迅速停止运转。快速制动用于当电机转动被行程开关产生中断的情形。也用于点击 *Mis Programmer* 软件执行状态对话框上的 STOP 按钮时。如果您将快速制动速率设置得很高,步进电机可能会"break lose(制动失败)"并在惯性的作用下运转到停止。这样的情况通常不会发生,除非负载的惯量很大。

#### 限位开关

Mis Programmer 的输入配置对话框有一个面板,可以选择限位开关或者传感器的类型。

如果开关会在电机到达一个限度时关闭,选择"Closed"选项。这常常叫做"常开"开关。如果开关 在电机没有超过一个限度时关闭,在电机到达一个限度时打开,选择"Open"选项。这种开关常常叫做 "常闭"开关。如果您没有在系统中使用开关,你可以选择"not used",把 cw 和 ccw 限位输入端作为7 号和8号通用输入端来使用 Wait Input 和 Feed To Sensor 指令。如果使用 If Input 指令,这个限位输入端 一直是有效的。

#### 触动限制开关时会发生什么?

如果您正在点动(使用 JOG CW 或者 JOG CCW 输入或者 MMI 上的箭头按键)并触动限位时,电机会 在正在行进的方向上停止。然后你可以从这个限位位置反向点动电机。

在 Seek Home 指令期间,电机会在到达限位之前反转,然后继续查找原点。 如果在 Feed To Length、Feed & Set Output、Feed To Position、Feed To Sensor、Feed & Return 或者 Feed To Sensor & Return 指令运动时遇到限位,驱动器会立即停止电机。

#### 中断和限位跳转

如果在对话框中选择"branch on interrupt"及"branch on limit",两种事件都会转到同一程序行。 你需要区分这两种事件。如果使用的是 Call/Return 这样的中断,你必须在中断处理中含有 Return 指令。 但在限位跳转中不能使用 Return 语句,否则会导致堆栈溢出。你可以用 If Input 指令来检测限位并跳转到 不同的指令段中。如果限制条件为"低"时,那么使用"If Input 7 or 8 low"。如果限制条件为"高"时, 那么使用"If Input 7 or 8 high"限位。

### 面板错误代码

Mis 驱动器面板上有一个双色(红/绿)LED 显示状态和警报代码。闪烁的绿色 LED 表示驱动器正常操作 电机。LED 一直是绿色表示电机无效。

在产生错误时,绿色的 LED 会闪烁一到三次,然后红灯闪烁若干次,这种闪烁模式会一直重复,直 到警报清除。

代码	错误描述

1 red, 0 green 超限 1 red, 3 green 子程序堆栈溢出

2 red, 3 green 子程序堆栈下溢

z ieu, 5 gieen J 柱/产堆伐下価

**4 red**, **3 green** 错误的 Mis 程序指令(内存或者软件错误)

**7 red, 1 green** 串口通信错误

### 使用灵活的人机界面(MMI)

Mis 驱动器具有使用灵活的人机界面(MMI),有时叫做操作面板。MMI 连接到驱动器的 RS232 端口,也就是连接到 PC 机的端口,它们使用同样的电缆。MMI 有显示四行文字的液晶显示屏(LCD)及 20 个按键用于输入数据。使用 MMI 可以完成以下 7 种功能:

- 1) 在 LCD 上显示一段信息。你可以显示机器标识或者显示状态信息。
- 2) 你可以暂停程序,直到用户按下 ENTER 键。
- 3) MMI 可以要求用户做出决定。例如,你希望提供用户一种选择,按 yes 或者 no 键来完成响应。
- 4)可以要求用户输入移动距离。如果您希望机器输出一段材料然后切断,操作员可以指定切断材料的长度。
- 5)可以要求用户输入移动的速度。这个选项允许操作员调节进给速率,流速或者其它电机相关的速度。
- 6)可以要求用户输入重复次数。你可以让用户设置处理的零件数。你也可以组合使用重复循环和 Wait Time 指令来调整等待时间。
- 7)可以显示菜单,等待用户按数字键来转到相应的程序行。可以使用任何一个或者全部 1-8 的数字键, 每一个按键都有它的跳转地址。

此时你可能会问"如果 MMI 和 PC 都是连接到驱动器上的同一端口,那么我怎样使用 MMI 在 PC 上运行程序呢?"。如果按"Execute(执行)"按钮而驱动器的程序包含有 MMI 指令的话你会看见 MMI emulator

对话框。MMI执行对话框外观和功能同 真实的 MMI 一样:它会显示信息,你可 以点击上面的按钮输入数据。上面也显 示输入/输出的状态,也显示允许中断的 控制面板,在任何时候单步执行或者重 新启动程序。

我们在驱动器中提供 MMI 的仿真 界面,以减少购买 RS-232 端口的费用。 也允许您在购买之前自己体验一下 MMI。





#### 怎样在 MMI 上显示一段信息

🐃 MMI Prompt	×	1					
Use the Man Machine Interface	ОК		Mach Statu	ine R s OK	unnin	9	
<ul> <li>Display text only</li> <li>Display text &amp; wait for enter</li> </ul>	Cancel						
C Display text, wait for yes/no & branch on yes		[		2	0		
C Display text & get speed	status OK		H	4	3	Ť	*
🔿 Display text & get repeat count			4	5	6	ł	$\rightarrow$
C MMI Menu - branch on number keys			7	8	9	YES	NO
Display a variable Dist1 C on line	3			0	SPACE	BKSP	ENTER
© on line	4	▎▝					

- 点击某程序行的图标
- 选择 MMI Prompt 指令
- 在文本框中输入 "Machine Running Status OK"
- •选择"Display Text Only"选项按钮
- 点击 OK 按钮
- 在使用其它 MMI 指令之前信息会一直显示在 LCD 上。

你也可以显示变量和当前重复循环的次数。例如:你想显示程序的总的循环次数和当前的次数。为此,选中 "Display a Variable"框,选择变量 Count1 并选中 "on line 3"。然后在程序中使用 Repeat 指令的时候也选择变量(Count1),然后选择 "display loop count on MMI line 4"。

#### 如何实现用户按 ENTER 键继续

<ul> <li>MMI Prompt</li> <li>Use the Man Machine Interface</li> <li>Display text only</li> </ul>	OK Cancel Help	Pleas labels ENTE	se reic s, the ER	ad n pres	s	
<ul> <li>Display text &amp; wait for enter</li> <li>Display text, wait for yes/no &amp; branch on yes</li> <li>Display text &amp; get distance</li> <li>Display text &amp; get speed</li> <li>Display text &amp; get repeat count</li> <li>MMI Menu - branch on number keys</li> </ul>	Text to display on MMI Please reload labels, then press ENTER	1 4 7	2 5 8	3 6 9	↑ ← YES	↓ → NO
Display a variable Dist1     On line 3     On line 4     MMI Jogging     Allow jogging on MMI arrow keys     CCW CW	3 4 •		U	OFACE	BKar	ENTER

- 选择 MMI Prompt 指令
- 在文本框中输入 "Please reload labels, then press ENTER"
- 选中"Display text & wait for enter"选项按钮
- 如果您想让用户使用 MMI 方向键点动控制电机,选中 "Allow jogging on MMI arrow keys"复选框。
- 点击 OK 按钮

#### 多个点动速度

驱动器允许从多个点动速度中选择。你仍然可以使用主窗口中指定的"全局"点动速度。你也可以选 择对应每条 Wait Input 指令唯一固定的"局部"点动速度。这样生成的程序允许既有快速点动速度又有慢 速点动速度。

你可以选择 MMI 变量作为点动速度。这和 MMI 速度指令组合起来就允许系统的操作员输入一种或多 种点动速度。

#### 显示变量

你也可以显示在第3行或第4行显示 MMI 变量。只要选择变量一由 Mis Programmer 软件来决定要显示的是什么类型的数据(速度、距离或者数量)以及应用适当的比例。



#### 怎样让用户作出决定(MMI 跳转)

MMI Prompt     X       Use the Man Machine Interface     OK       Display text only     Ine #       Display text & wait for enter     Text to display on MMI       C Display text & wait for yes/no & branch on yes     Change set up	Chang param YES o
C Display text & get speed     parameters? (press       C Display text & get repeat count     YES or NO)	1
Scale MMI variable Save data as	4
Limits Upper limit 50 Lower limit 1 Dist1	7

	Change the setup parameters? (press YES or NO)							
Γ	1	2	3	1	¥			
	4	5	6	Ť	→			
	7	8	9	YES	NO			
	. O SPACE BKSP ENTER							
1								

- 在第1行加入 MMI Prompt 指令
- 在文本框中输入 "Change setup parameters?(press yes or no)"。
- 选中"Display text, wait for yes/no & branck on yes"选项按钮
- 在 "Line #" 文本框中输入 12
- 点击 OK 按钮
- •从程序中的第12行开始加入设置参数的指令
- 在设置参数的指令结束的地方加入 Go To line 2 指令。

#### 怎样要求用户输入移动距离

MMI Prompt     X       Use the Man Machine Interface     0K       C Display text only     1       C Display text & wait for enter     Cancel       C Display text, wait for yes/no & branch on yes     Text to display on MMI	E Ix 8	Enter engti :.9	the p n, in ii	oart nches		
C Display text & get distance     Display text & get speed     Display text & get repeat count     Scaling     MMI variable	ŀ	1	2	3 6	↑	<b>↓</b> →
C multiply MMI entries by     1     to get steps       © user defined units - MMI entries will be in inch     Save data as       Limits     Dist1       Upper limit     12	ŀ	7	8 0	9 SPACE	YES BKSP	NO ENTER
Lower limit 0.5 inch						

- 选择 MMI Prompt 指令
- 在文本框中输入 "Enter part length, in inches"。
- 选中"Display text and get distance"选项按钮
- 输入一个比例因数,如果你要使用用户自定义单位,选择"user defined units MMI entries will be in inch"以自动缩放比例。
- 输入上限和下限(在本例中,我们想让操作员输入距离在 0.5 到 12 英寸之间)。
- •选择 MMI 变量来保存距离(我们这次选择 Dist1,但是在 MMI8 个变量中的任何一个都可以存储任何 类型的数据)
- 在程序的后面,你需要加入使用距离变量 Dist1 的 Feed Instruction(Feed To Length、Feed & Return 等等)指令。

#### 怎样获取用户输入的速度



- 选择一个 MMI Prompt 指令
- 在文本框中输入 "Enter the flow rate, in gallons/minute"。
- 选中"Display text and get speed"选项按钮
- 输入比例因数(在本例中, 1 加仑/分钟相当于电机的 10 转/秒)
- 输入上限和下限(在本例中,我们想让操作员输入流速在1到5加仑/分钟之间)。
- •选择 MMI 变量来保存速度(我们选择 Speed1)
- 在程序的后面,你需要加入使用速度变量 Speed1 的 Feed Instruction(Feed To Length、Feed & Return 等等)指令。

#### 怎样获取用户输入的重复次数

MI Prompt Use the Man Machine Interface C Display text only C Display text & wait for enter C Display text, wait for yes/no & branch on yes C Display text & get distance C Display text & get speed C Display text & get speed	X OK Iz Cancel Text to display on MMI How many parts should we run?
Scale Multiply input data by 10 Limits Upper limit 500 Lower limit 1	MMI variable Save data as Count1 ▼

	How many parts should we run? 150					
Γ	1	2	3	1	¥	
l	4	5	6	Ť	+	
l	7	8	9	YES	NO	
		0	SPACE	BKSP	ENTER	

- 选择 MMI Prompt 指令
- 在文本框中输入 "How many parts should we run?"。
- 选中"Display text and get repeat count"选项按钮
- 输入上限和下限(在本例中,我们允许操作员输入的数字在1和500之间)。
- •选择 MMI 变量来保存速度(我们选择 Count1)
- 在程序的后面,你需要加入使用重复次数变量 Count1 的 Repeat Instruction 指令。

#### 怎样创建 MMI 菜单

MMI Prompt							
<ul> <li>Use the Man Machine Interface</li> <li>Display text only</li> <li>Display text &amp; wait for enter</li> <li>Display text, wait for yes/no &amp; branch on yes</li> <li>Display text &amp; get distance</li> </ul>	OK Cancel Help Text to display on MMI 1=12 oz bottle		1=12 2=16 3=24 4=cus	oz bo oz bo oz bo stom	ottle ottle ottle size		
<ul> <li>Display text &amp; get speed</li> <li>Display text &amp; get repeat count</li> <li>MMI Menu - branch on number keys</li> </ul>	2=16 oz bottle 3=24 oz bottle 4=custom size	Í	1	2	3	1	¥
Branch Table			4	5	6	+	$\rightarrow$
If user presses 1, branch to line     If     If user presses 2, branch to line     20     □     If	user presses 5, branch to line 50		7	8	9	YES	NO
✓     If user presses 3, branch to line     30     □     If       ✓     If user presses 4, branch to line     40     □     If	user presses 7, branch to line 70	ļ		0	SPACE	BKSP	ENTER

- 选择一个 MMI Prompt 指令
- 选中"MMI Menu..."选项。
- 在文本框中输入菜单文本(你最多可以输入4行)
- •检查每条菜单对应哪一个键(本例我们使用 1, 2, 3 和 4 这四个键)
- •为每一行的按键分配程序行号(我们使用 10, 20, 30 和 40)
- 在程序的后面需要在这些行号中加入相应的指令。当用户按 MMI 上对应的键时转到相应的指令。例如,当用户按"1"时,程序进入第10行。

注意:如果加入了较多的空格以便使文本看起来正好,警告:有时 Mis Programmer 软件会在重新打开指令对话框时移除这些空格。更安全的做法是使用格式化字符,比如 "."(点) "\_"(下划线)。

### 产生动作



#### **MMI Prompt**

MMI Prompt 指令用于 MMI(人机界面)。MMI prompts 允许用户在屏幕上显示信息,并从操作员那里 搜集信息用于其它指令。MMI 也可以暂停程序,直到用户按下 ENTER 按钮。它允许用户作出决定,按 YES 或者 NO 按钮。如果用户按 YES,程序跳转到另外的行,如果用户按 NO,程序继续向下执行。

如果你只是想显示一段信息,比如 "Machine Running –Status OK",在程序中你想显示信息的地方 加入 MMI Prompt 指令。选中 "Display Text Only"并输入要显示的信息。当执行到这条 MMI Prompt 指 令时要显示的信息就会一直显示在屏幕上直到其它指令改变了 MMI 上显示的信息。

MMI Prompt	×
Use the Man Machine Interface	ОК
<ul> <li>C Display text only</li> <li>C Display text &amp; wait for enter</li> <li>C Display text, wait for yes/no &amp; branch on yes</li> <li>C Display text &amp; get distance</li> <li>C Display text &amp; get speed</li> <li>C Display text &amp; get repeat count</li> </ul>	Cancel Text to display on MMI enter part length in inches (0.5 - 6.0)
MMI Menu - branch on number keys         Scaling         multiply MMI entries by         user defined units - MMI entries will be in inch         Limits         Upper limit         Lower limit         0.5	eps Save data as Dist2

如果您想让机器 操作员可以改变象距 离、速度或者重复次数 这样的参数,你需要 MMP Prompt 指令询 问用户输入数据并保 存在非易失性的内存 中。在这种情况下,点 击想显示的数据类型 (距离、速度或者重复 次数)前面的单选框。

你需要设置上下 限。MMI Prompt 指令 会检查用户输入的数 据是否超过了您定义 的限度,如果输入的数 据超出范围会告诉用 户。例如,使用 MMI Prompt 指令取得重复

次数,将上下限设成100到1,指令不会接收比100大和比1小的数。

你必须告诉 MMI Prompt 指令将数据存储在非易失性内存中的什么地方。有 8 个位置可供选择。他们 是 Dist1, Dist2, Dist3, Speed1, Speed2, Count1, Count2 和 Count3。要记得你将数据保存在什么地 方。当你设置某条指令使用 MMI 变量的时候,你必须指定指令使用哪一个变量(Dist1, Dist2 等等)。

例如,你想让操作员能够设置机器处理的次数,那么在程序中加入 MMI Prompt 指令来询问重复次数 并保存为 Count1。在你程序中的其它地方你可以插入 Repeat loop 指令来处理某些零件。循环会在 Repeat 指令处开始,循环次数从已经配置的 MMI 变量 Count1 中获取。你甚至可以在 MMI 上显示程序运行的循环次数。

#### 比例缩放

驱动器是按每秒种多少步数和细分来工作的。MMI Prompt 也以接收其它单位的用户数据(例如英寸或 者英寸/秒)并会自动按比例缩放后转化成内部单位。有如下两种方法: 最简单的方法就是在主窗体中设置用户单位。这样你整个程序都可以输入自己的单位。参见第 10 页 中自定义单位的详细解释。

**MOONS'** 

另一个方法就是在 MMI Prompt 对话框中直接输入一个比例因子。这样你就可以在不同的 MMI Prompts 中使用不同的比例因子。

比例缩放只在采集距离和速度数据的时候有效。

#### 其它 MMI Prompts 指令的使用

如果您想让程序暂停,等用户按MMI上的ENTER键时才继续执行,选中"Display text & wait for enter" 选项按钮。根据需要,你也可以允许操作员使用 MMI 箭头键来点动控制。

要让用户作出决定,选择 "Display Text, wait for yes/no & branch on yes."确认在 Line #文本框中输入行号。程序会在用户按 YES 键的时候跳转到这个程序行。如果用户按 NO 键,程序会执行 MMI Prompt 指令后的程序行。

MMI 菜单选项让你为最多 8 个数字键分配行号并在 MMI 上显示 4 行文本。当操作员按其中一个数字 键时,程序转到相应的行。这是一个设置菜单驱动系统的简便方法。

如果使用 MMI Text only 或者 MMI Wait Enter 选项的话你也可以在第3行或者第4行显示 MMI 变量。 只要选择变量--Mis Programmer 软件会自己处理它是什么类型的数据(速度、距离或者数量)并使用正确 的比例。

#### 保护机器设置

MMI Prompts 通常用来对机器进行设置。在某些场合下要防止操作员改变这些设置。Mis Programmer 不提供密码保护功能,但有一个机械的按键开关可以达到这个目的。这个按键开关与 Mis 驱动器上可编程 的输入端相连。(请参考驱动器的硬件使用手册查看连线细节)在程序中加入一条 If Input 指令,当按键开关 处于保护位置的时候跳过 MMI Prompt。

翻到第25页来查看更多 MMI Prompts 的例子。



#### **Feed To Length**

Feed To Length 指令用于点对点的移动。如果你只是想让电机移动一个固定的步数就可以使用这个指令。你也可以使用前面通过 MMI Prompt 指令得到的速度或者距离数据。

当点击 Program Line...对话框中的 Feed To Length 按钮时,你会看见 Feed To Length 对话框。在这里输入移动的参数。

Feed to Length	×
Move the Motor a Fixed D	Distance Cancel OK
Parameters	Analysis
Distance 2000 Steps	Get distance from MMI Name Dist1 CCW CCW
30.00000 Rev/sec	Get speed from MMI Name Speed1 💌
Accel	
Reduce Speed During Move	Get speed change dist from MMI Dist2 💌 sec —— Get speed from MMI Speed2 💌

距离一这是你想让电机移动的步数。最大值是 16,000,000。如果选中"Get distance from MMI"复选框,你可以选择 8 个 MMI 变量中的一个作为距离。请注意选中"Get distance from MMI"并不会自动使驱动器停止并询问用户输入。你需要在程序中的其它地方加入 MMI Prompt 指令。

速度--这是你想让电机运转的最大速度,单位是转/秒。你可以将速度设置在 0.025 和 50rev/s 之间,以 0.025rev/s 为单位递增。如果你选中 "Get speed from MMI",你可以选择 8 个 MMI 变量中的一个作为速度。

你也可以通过选中"Reduce speed during move"减速运行。这种方法使用在刀具需要快速接近工件, 但又要在快接触到工件时减慢速度的场合。

加速度一电机不可能一开始就高速运转。驱动器需要逐渐的加快速度。加速度的大小取决于电机和负载的惯量,电机的可用力矩,以及你想让它运转多快。你需要做一些实验才会有更深的认识。加速度的范围是 1 到 3000 转/s<sup>2</sup>。

减速度-这是驱动器在电机移动即将结束时减速的速度。也是电机减速的速度。减速度的范围跟加速 度相同。由于磨擦会加快电机停止,所以你可以将减速度设成比加速度略大一些。

方向一你可以选择移动的方向为 cw(顺时针)或者 ccw(逆时针)。在前面的圆圈上打勾就行了。



分析一点击 Analysis 选项卡会出现移动的速度-时间曲线。上面也提供了一些关于移动的有用的统计, 比如移动过程和加速、减速花费的时间。你也可以选择速度一距离曲线。

Feed to Length			×
Move the Motor a Fixed Distar	nce	Cancel	ОК
Parameters	ſ	Analysis	
i 15.0 i 10.0 c 5.0 c 0.0 0 0.5 1 sec Graph © speed vs time © speed vs distance	1.5	time           Accel         0.100           at Speed         0.350           Decel         0.090           at Speed2         0.500           Final Decel         0.010           Total         1.050	inch 0.5 3.5 0.495 0.5 0.005 5.0 inch/sec



#### Feed & Set Output



Feed & Set Output 允许您在移动期间中的任何一点设置输出为"高"或者"低"。

Feed to Length & Set Output				×
$\textcircled{W}^{v}$ Move Motor Fixed	Distance & Set Ou	tput <u>ca</u>	ncel	OK
Move Parameters	Output Parameter	s]	Analy	sis
Set output at end of m Set output during mov 2.2 Get output distan Dist2	ove C e C inch G ce from MMI C C	ondition ' High (open) ' Low (closed) ' High pulse ' Low (closed) ' Low (close	- 2	msec



#### Feed & Return

Feed & Return 指令用在点对点的移动并回到起点的场合。

例如,如果电机正在驱动切割刀具,你会想让刀具切割之后返回起始位置。

Feed & Return		×
Move Motor Fixed Distance 15000 Steps Speed 7.50000 Rev/sec	nce, then Return Get distance from MMI Name Dist1 Get speed from MMI	OK Cancel Direction
T T	Name Speed1 💌	⊙ CW
		C CCW
	▶ 100 Rev/sec/sec	
Decel I	▶ 100 Rev/sec/sec	
Return	<b>20.00000</b> Rev/sec	
Return Delay	▶ 1.50 Seconds	

Feed & Return 需要设置的参数跟 Feed To Length 基本相同:距离、速度、加速度、减速度和方向。 要查看这方面的详细解释,请参阅手册上的 Feed To Length 一章。

你还需要设置返回速度。范围是 0.025 到 50 转/秒。在切割刀具的情况下,你希望缓慢进给,因为刀 具正在切割,然后迅速返回。这样,你可以将回退速度设成比前进速度大。

**Return delay** 决定了驱动器在进给结束到开始返回的时间。举个例子,这样可以让机器一定的时间在 回退之前来移除某个零件。由于电机和负载在移动之后需要时间去处理,你不能将返回延迟时间设成小于 0.2 秒除非你确信电机和负载比一般情况处理得更快。





#### Feed To Sensor

Feed To Sensor 指令允许移动电机直到外部事件改变了某个输入端的状态。

例如当运动距离是变化的。我们设想你正在使用电机分发来自于传送带的标签。你不能保证标签之间 的距离非常精确,所以你不希望每次都进给同样的步数。你可以在供给装置上安装一个传感器来"观察" 每一个标签的边缘然后向驱动器输入端发送信号来停止移动。

Feed To Sensor 会询问和其它 Feed 指令相同的参数:速度、加速度、减速度和方向。你还需要指定距离。这是因为在传感器出错的情况下驱动器必须有足够的空间来制动。速度越高,停止需要的时间越长。如果减速度增加了,那么电机会在更少的步数之内停下来。"Minimum Distance"框告诉你基于设置的速度和减速度所能容许的最小步数。你不能设成比这个最小值还小。

Feed to Sensor	×				
Move Motor a Fixed Distance Past Sensor					
Distance	ок				
inch beyond Get distance from MMI	Canad				
Sensor Name Dist1	Lancei				
Speed	Direction				
5.00000 inch/sec	C CW				
Name Speed1 ▼					
	⊙ CCW				
Accel  100.0 inch/sec/sec					
Decel  Image: Decel Image: Dece					
Sensor Input Condition					
C 1 C 3 C 5(cwjog) C 7(cwlim) C High C Rising Edge speed an	tance based on id decel rate				
O 2 ○ 4 ○ 6(ccwjog) ○ 8(ccwlim) ○ Low ○ Falling Edge 0.125					
Safety Distance					
☐ If distance exceeds safety limit before sensor is found, stop motor and goto line 1					
0.00005 inch					

你还需要告诉驱动器传感器连接到哪一个输入端以及输入条件。输入条件有如下四种:

高一移动到指定的输入到达一个高信号状态。这是输入端没有连接时的默认状态。

低一移动到指定输入到达一个低信号状态。

**上升沿**-移动到信号从低转到高时。这和高条件相似,但是有差别。例如你有一台连接到 Mis Index 的传感器,你想让传感器信号变"高"时移动停止下来。然而,传感器信号在移动完成后保持"高",一会儿回到"低"。这在滚动卷上的标签纸之间的空间比较小的打标应用中经常发生。如果你选择"高"作为输入条件,驱动器会完成移动,但不会再一次启动,因为输入信号仍然保持"高"。如果你选择上升沿条件,驱动器会在输入"高"时动作并且在传感器信号再次从低转到高时停止。

下降沿一和上升沿相反。驱动器等待输入电压升高再降低的条件。

如果你担心负载从没接触传感器(例如,如果传感器故障或者负载堵塞),选中"If distance exceeds safety limit..."。你就可以输入安全距离并指定当传感器没收到信号时程序转到什么地方。例如,你正在检测滚动卷上的标签信号,假设他们之间大概有1英寸的距离间隔,输入安全距离为3英寸,然后输入跳转行号为20。在第20行,你可以加上一些错误处理的指令,象MMI Prompt来告诉操作员检查标签进料情况。

注意: Mis Programmer 软件不会让你输入比最小减速距离还要小的距离,此距离是根据设定速度和 减速度而得到的。然而,如果速度和距离是由 MMI 变量来设置的话,并没有这样的错误检查。你有责任 来设置速度的下限和距离的下限以让操作员不能输入错误的值。不这样做可以会导致预料之外的长距离移 动。



#### Feed To Sensor & Return

Feed To Sensor & Return 指令就象 Feed to Sensor,但在移动完成之后再返回到起始点。 大部分的参数跟 Feed to Sensor 相同,但添加了两个新的参数:返回速度和返回延迟。

Feed to Sensor & Re	turn			$\times$		
C Move Motor Fixed Distance Past Sensor & Return						
Distance	Change have a d	E Got distance	o from MMI	0K		
6000	steps beyond sensor	Name Dist1		Cancel		
Speed 10.00000 Bev/se	ec.	Get speed from I	ммі	Direction		
•	•	Name Speed1	7	● CW		
				CCCW		
Accel		▶ 100 F	lev/sec/sec			
Decel		▶ 190 F	lev/sec/sec			
Return I		▶ 5.00000 F	lev/sec			
Return Delay		▶ <b>2.00</b> 9	econds			
Sensor Input	Con	dition	Minimum Dia	tance based on		
●1 ●3 ●5(cwjog	g) O 7(ewlim) O H	igh C Rising Edge	speed ar	nd decel rate		
C 2 C 4 C 6(cow)	og) O 8(ecwlim) 💽 L	ow C Falling Edge		5263		

Feed to Sensor & Return 很有用的应用就是距离变化的系统。如果我们生产的机器来切割不同尺寸的 布料,而且驱动器来驱动切割刀具,我们可能想在切割结束的地方加入传感器。这样我们就可以根据某一 天布料的宽度来调整传感器的位置而不用重新对驱动器进行编程。

每次触发时,驱动器会供给到刀具走到传感器,然后返回到起始点。

你应该注意到:任何程序设定的最大距离是 16,000,000 步。这是驱动器所能行走的最大距离。如果 你在碰到传感器时移动超过 16,000,000 步,驱动器不会返回到正确的位置。如果对你来讲这是个问题, 考虑选择低细分解决方案。在 50,000 步/转时, 320 转后你将超过 16,000,000 步的限制。在 2000 步/转时, 你将在超过这个限制之前运行 8000 转。





#### **Feed To Position**

这条指令移动电机和负载到一个绝对的位置。例如,如果负载在 4 英寸的位置,程序执行 Feed To Poistion 6 inches,那么电机会顺时针转 2 英寸。如果负载处在 10 英寸的位置并且你执行 Feed To Position 6 inches,那么电机会逆时针移动 4 英寸。

Feed To Position 通常使用这几个参数:速度、加速度和减速度。象其它移动指令一样,速度可以调用 MMI 变量,允许操作员在 MMI 面板上输入。

位置可以是正值或负值,并且可以在 MMI 上输入。请注意 MMI 上是没有负号键的,所以不能在 MMI 上输入一个负数。你应该避免通过 Set Position 指令来使用负的绝对位置。

	X
Absolute Position	
Get position from MMI	OK
Name Dist1 🔽	Cancel
Get speed from MMI	
Name Speed1 🔻	
► 100.0 inch/s/s	
	Absolute Position Get position from MMI Name Dist1 Get speed from MMI Name Speed1 100.0 inch/s/s 100.0 inch/s/s



#### **Set Abs Posistion**

这条指令允许定义当前电机的位置作为任何你想要的绝对位置。Seek Home 指令执行完成后会自动 清除绝对位置计数器,将此时原点位置定义成 0。但可能你想让原点传感器位置作为 8 英寸的位置,或 90, 度或者什么的,在 Seek Home 指令之后简单的加入一条 Set position 指令,或者在程序中的任何其他地 方来定义绝对位置。





#### **Save Abs Position**

**MOONS'** 

这条指令允许保存当前的绝对位置到 MMI 变量。这在你想让操作员形象的定位负载的位置,并在稍候的程序中回到哪个位置时很有用。

在材料处理应用中,你可以创建程序,使用 Wait Input 或者 Hand Wheel 命令允许操作员移动负载到 某个位置。操作员按 ENTER 按钮退出 Wait Input 或者 Hand Wheel 指令。如果下一条指令就是 Save Abs Position,操作员谨慎地取得的负载位置就会记录在非易失性的内存中了。在程序中的别的地方你使用 Feed To Position 指令来返回到负载的位置。

因为 Mis 支持 8 个 MMI 变量,你可以保存 8 个不同的位置。

范例程序"LPdemo"给出了获得 2 个位置的示例。

即使驱动器的位置已经"获取"在下一天也会记住(因为是保存在非易失性内存中的),你需要在每次通电时将系统复位。要不然你保存的绝对位置将没有意义。

🐂 Save Abs Position	
Save current at an MM	osolute position to 11 variable
MMI Variable Dist1	Cancel



#### Seek Home

Seek Home 指令允许你移动电机直到找到原点。Home 传感器可以连到任何一个通用的输入端口。 某些应用要求电机每次通电时都从某个确定的位置开始,但不能保证上次断电时停留在什么地方。解 决方法就是连接一个传感器到驱动器的输入端并在程序的起始位置附近加入一条 Seek Home 命令。

Seek Ho	ome					×
	Find	the Home Se	ensor			ок
						Cancel
Spee	d 💽			▶ 2.000	Rev/sec	Direction -
Acce				100	Rev/sec/sec	⊙ CW
Dece	I I			→ 125	Rev/sec/sec	O CCW
Sens	or Inpu	Jt	- Condition	n		
C1	<b>C</b> 3	<b>C</b> 5 (CW Jog)	C High	C Rising Edge	Required cla limits i	& hard stop
C 2	• 4	C 6 (CCW Jog)	C Low	Falling Edge	320	Steps

Seek Home 会询 问很多跟其它 Feed 程 序基本一样的参数:速 度、加速度、减速度和 方向。

你还需要定义驱 动器连接到哪一个输 入端以及搜索条件。

高一移动到指定 的输入输入"高"信号 状态。这是输入端口没

#### 有任何连接时的缺省状态。

低一移动到指定输入"低"信号状态。

**上升沿**-移动到信号从低转到高时。这和高条件相似,但是有差别的。如果你执行 Seek Home 命令, 选择"高"输入,当负载已经在 Home 传感器时(导致输入为高位),负载不会移动。如果你换成"上升沿", 驱动器会移动负载到 Home 传感器的边缘。

如果你需要每次 Seek Home 命令之后让负载停在准确的同一位置,选择上升沿或者下降沿。

下降沿一和上升沿相反。驱动器等待输入电压升高再降低的条件。

驱动器是以指定的方向移动电机来执行 Seek Home 命令的。如果找到了 Home 传感器,电机减速到 停止,然后倒退到传感器处。如果在主传感器找到之前遇到上下限,驱动器反向移动并继续搜索 Home 传感器。

也许你已经注意到 Seek Home 对话框右下角有一个文本框。这告诉你驱动器减速到停止需要多少步。 "Required Clearance"框告诉你在限制传感器和任何硬物堵塞之间你必须允许多少距离,取决于你设置 的速度和减速度。如果你不允许足够的距离,负载可能会在搜索 Home 的时候超过限制并碰到什么东西。 速度越快,停止的时间越长。如果减速度在增加,那么电机会在较少的步数之内停止。



### Wait Time

这是最简单的指令。只要输入时间,驱动器就会在处理下一条指令之前暂停这么长的时间。范围是 0.01 到 300 秒。

Wait Time	×
🔄 Wait a Fixed Amount of Tir	ne
1.40 Seconds	OK
. I →	Cancel

你可以在周围放置一个循环让 Wait Time 指令持续时间更长。到现在还没介绍循环,所以我们在这里就省略了。第一个技巧就是把 30 分钟分成两部分。30 分钟不就是 1800 秒吗,对吧?我们在一次 Wait Time 指令中最多延迟 300 秒。好,那么我们把延迟 300 秒执行 6 次将怎样呢?

程序看起来象下面这样:



有限制吗?循环次数可以到达 65536 次,所以最大延迟时间是 65536 x 300=1966 万秒,5461 小时。 试试两重循环,一次嵌套在另一次之中,现在我们可以暂停 40000 年了。



#### Wait Input

很少有电机控制器能够在没有其它外部输入的情况下自己独立完成所有工作的。在大多数情况下,你 需要让驱动器在进入移动之前等待某些条件成立。Wait Input 命令就是用作这个用途。Wait Input 命令也 是唯一允许使用 JOG CW 和 JOG CCW 输入端来缓慢点动电机的指令。

Wait Input 指令有两种操作模式:单输入模式,用在只有一个输入端的地方,多输入模式,用在需要同时检测 8 个输入端的地方(BLU-Mis 驱动器为 16 个输入端)。

在单输入模式下,你必须指定哪一个输入端以及期待电压条件。有如下几个选择:

高一移动到指定的输入到达一个高信号状态。 这是传感器没有连接任何东西时的默认状态。所以 使用连接条件时要小心。如果连线断开,你可以结 束预料之外的移动。

低一移动到指定输入到达一个低信号状态。发 生在输入端正在传导电流时。如果使用瞬间接触开 关(常开型),这种条件会在按下按钮时发生。

**上升沿**一一直等待,直到信号从低转到高。这 和高条件相似,但差别是很重要的。设想信号进入 驱动器是你想让移动发生的一个条件。然而,信号 在移动完成后仍保持高位,一段时间之后才降低。 如果你选择"高"条件作为输入条件。驱动器会完 成移动后重新启动因为信号在首次移动时仍然保持 高位。如果你选择"上升沿",驱动器会等待输入电 压降低,然后升高时才移动。

💐 Wait Input		×
📲 Wait for a	n Input Co	ndition
Single input C Mu	ltiple Inputs	
- Input	Condition –	
• 1 • 2	O High (open)	Cancel
C 3 C 4		Jog Speed
C 5(cwipg) C 6(ccwipg)	C <sup>Rising</sup> Edge	Use global jog speed     O     O     O
O 7(cwlim) O 8(ccwlim)	⊖ Falling Edge	© Speed1 ▼

**MOONS'** 

下降沿一和上升沿相反。驱动器等待输入电压升高再降低的条件。

如果选择"Multiple Inputs",Wait Input 指令会同时扫描多个输入端来决定程序是否转到下一条指令。 输入可以通过二进制数(等待输入端1为低或输入端3为高位)或者乘积(等待输入端1为低与输入端3为高位)。

如果是 BLU-Mis 驱动器, 你会看见更多的选择, 包括 X0, 编码器索引。和鸣志电机一起, X0 在索引器上时读取高电位, 其余的读取低电位。

#### 多倍点动速度

驱动器允许选择多倍点动速度。你仍然可以使用"全局"点动速度,在主窗体中定义。你还可以选择 固定的在每个 Wait Input 指令中不同的"局部"速度。这样你可以生成快速和慢速点动速度。

你也可以选择 MMI 变量作为点动速度。这和 MMI Speed 组合使用可以让系统操作员输入一个或者多 个点动速度。

#### 注意:条件中的"低"(或者关闭)发生在电流流过输入终端时。高(打开)的意思是没有电流流过。



#### **Hand Wheel**

**MOONS'** 

Hand Wheel 指令类似于 Wait Input 指令。如果指定输入端和输入条件,比如"Input 4 low",指令会一直执行,直到上述条件成立。

Wait Input 指令允许用户使用 CW LOG 和 CCW JOG 输入端或者 MMI 上的方向键让电机运转。Hand Wheel 跟 Wait Input 的功能相同,只是 Hand Wheel 象旋转 CNC 手轮一样来定位电机位置。以通过告诉 驱动器每次手轮转动到下一个位置电机移动的距离来指定比例参数。

Hand wheel 在 100 个计数/转时运行最为良好。如果您想购买的话请联系制造厂商。 你可以把手轮定位看成"数字点动":它允许机器操作员精确的调整负载位置。

Follow En	coder	×
Follow	Hand Wheel Until Inp	out Condition
0	Connect Encoder Channel A Connect Encoder Channel B	x to Input 1 3 to Input 2
Distanc	e Per Encoder Count —	Condition
5	Steps	C High
- Input -		• Low
© 3 C	<b>)</b> 5 (cw jog) <b>(</b> 7 (cw lim)	C Rising Edge
C 4 C 6 (ccw jog) C 8 (ccw lim)		C Falling Edge
Text to	w until MMI ENTER display on MMI	OK Cancel



#### Go To

Go To 指令用于在程序中实现跳转。你需要在程序的最后加上 Go To 指令来跳转到起始行。



Go To 指令只有一个参数需要设置:目标程序行。点击上下按钮来增加或者减小行号。





#### **Repeat/End Repeat**

Repeat	×		
C Repeat a Fixed Num	ber of Times		
10 Times	OK		
Get repeat count from MMI Cancel			
Name Count1 💌			
C count up C count down			
1			

$\square$	<b>⇒1</b>	$\bigcirc$	Repeat 4 times
	2		Wait for Input 1 falling edge
	3		CW 100 steps, 10 rps
	4	$\bigcirc$	End Repeat Loop
	5		Wait for Input 1 falling edge
	6		CCW 400 steps, 10 rps
	_7	←	Go to line 1

→ <b>1</b>	$\odot$	Repeat 10 times
<b>⊃2</b>	G	Repeat 10000 times
3		CW 100 steps, 10 rps
4	$\odot$	End Repeat Loop
5	$\odot$	End Repeat Loop

有时你需要重复某一条指令,首先你应该知 道重复的次数。Repeat 循环指令循环的最大次数 可以达到 65,536。

例如,设想我们正在往一组瓶子中分装液体。 有 25 个瓶子,以 5 行 5 列排列,每装完一个瓶子 电机需要运转 100 步。每次我们收到一个触发命 令,电机就运转到下一个位置。在 5 个瓶子装满 后要回到起始位置,即返回 400 步。

#### 程序定义如下图所示。

我们在程序的第 1 行输入了 repeat loop 指 令,循环 4 次。接下来让驱动器等待输入端 1 的

电压下降,然后顺时针运转 100 步,将分装器移动到下一个瓶子。 如此循环 4 次之后驱动器退出循 环,转到第5行。这次当输入端 1 电压下降时电机逆时针转动 400 步,然后返回起点。

有时候你需要循环65,536次 以上。我们设想给切割刀具输送 材料,您希望输送100,000次。

最好的解决方法就是设置 2 重嵌套循环。总的循环次数等于 两个循环次数的乘积。

100,000=10,000 x 10,所以我们可以设置其中一个循环次数为 10次,另一个循环为 10,000次。

## MOONS' Mis Programmer 软件使用手册

Repeat 指令也可以使用通过 MMI Prompt 指令收集并存储用户输入的数据作为循环次数。只需选中 "Get repeat count from MMI"并从列表中选择一个变量即可。

例如,你在程序中加入了一条 MMI Prompt 指令来询问用户零件处理的次数并保存为 Count1。你就可以设置 Repeat 指令的次数为 MMI 变量 Count1。

你也可以在程序运行时在 MMI 上显示循环次数。你可以采用累计次数(例如显示已处理的零件数)或剩 余次数的方式。

Repeat 指令在 MMI 上显示循环次数时显示在第 4 行。你可以在 Repeat 指令之前加入一条 MMI Prompt 指令来告诉操作员循环的信息。如图所示:



注意:如果您使用 If Input 指令来退出 Repeat Loop 指令,循环的次数还会在下一次的循环中保留。 如果用 If Input 指令来退出循环,你需要使用 Reset Repeat Loop 指令来复位。否则循环次数会在 If Input 结束时保留。



#### **Reset Loop or Interrupt**

#### Reset Loop 选项

如果循环被 If Input 或者 Feed To Sensor 指令终止,可以使用 Repeat Loop 指令来强制重置循环计数器。

有时候有必要在循环完成之前终止循环。例如,你用一个循环往 100 个瓶子中灌装液体。如果蓄水池 干,你想终止循环,你可以在循环中加入 If Input 指令,由液体传感器来触发。If Input 会跳出循环,可以 用一条 MMI Prompt 指令来告诉操作工来重新填满蓄水池。

现在设想已经装满了 60 瓶,还剩 40 瓶。如果你想让循环返回到中止的地方,那么在操作员在填满蓄 水池之后回到循环,这样循环会自动填满剩下的 40 瓶。

🖹 Reset Repeat Loop	×
🍈 Reset Repeat	Loop or Interrupt
<ul> <li>Reset Repeat Loop on the Note: you only need to your program branches on If Input or MMI Bra</li> </ul>	at starts on line <b>1</b> use this instruction if out of a loop using an nch instruction.
C Reset Interrupt Also reset call stack.	Cancel OK

另一方面,如果你正在零件中打孔,每个零件需要 打5个孔。在每打一个孔后,电机向前移动零件1英寸。 这样,你需要循环5次。在打好第3个孔后,检测到该 零件已损坏,此时你需要退出循环,但循环次数还剩下 2次。

如果你简单的重新进入循环,下一个部件会只打 2 个孔。你实际上希望的是循环从 0 重新开始。为此,你 必须 Reset Repeat Loop 指令,如上图所示。

#### Reset Interrupt 选项

这个选项用来在退出中断处理程序后重新激活中断,只适用于 Go To 类型的中断。对于 Call/Return 中断,使用 Return from Interrupt 指令来重新激活中断。

	→∎	Ľ	JRepeat 5 times
	2	2	advance the part 1 inch
	3		CW 1.0 inch, 5.0 ips
	4	~	if part is bad, spit it out
_	5	₹	If Input 2 low, go to line 12
	6	2	signal drill
	7		Set Output 1 low pulse 500 msec
	8	X	wait for drill to finish
	9		Wait for Input 1 falling edge
	10	G	End Repeat Loop
	11	↓	
L	<b>→12</b>	8	advance part 10 inches
	13		CW 10.0 inch, 5.0 ips
	14	$\bigcirc$	Reset repeat loop on line 1
	15	◄	Go to line 1



#### Set Output



起初,我们讨论将 Wait Input 指令作为在处理之前让驱动器等待外部事件。有时你想要的恰恰相反:驱动器应该告诉其它设备继续处理。Set Output 指令让你在三个输出端口中选择一个并且输出一个电压信号。要获取电路和连接的详细信息,参阅硬件手册上的"Wire Inputs And Outputs"章节。

输出信号有四种情况:

High-打开光电晶体管。电路中在"-"端的输出端是地, "+"端上拉,这导致高电压出现在"+"端。

Low-关闭光电晶体管。电路中在"-"端的输出端是地, "+"端下拉,这导致低电压出现在"+"端。

High Pulse-打开光电晶体管一段时间(2 到 500 毫秒)。

Low Pulse-关闭光电晶体管一段时间(2 到 500 毫秒)。 在上电时驱动器将三个可编程输出端置高(开路)。



举个使用 Set Output 指令的例子,让 我们思考填充容器的例子。每次驱动器运 转到新的位置,它会告诉分装机。我们采 用低脉冲,但你的选择取决于分装机接收 的信号类型。

在每一条 Feed To Length 之后加入 一条 Set Output 指令,程序显示如左图所 示。

这里你可能需要一个较长时间的脉冲,这可以通过组合两条 Set Output 命令和一条 Wait Time 命令来完成。如下图所示,在3号输出端产生一个5秒的高脉冲。







#### If Input Go To

这条指令允许驱动器根据输入信号来决定如何执行。你可以选择"Single Input"或者"Multiple Inputs"。

	CW 1000 steps, 10 rps
$\odot$	Wait 0.5 seconds
	If Input 1 low, go to line 8
$\odot$	Wait 1 seconds
←	Go to line 4

你需要选择一个输入端。接着你还需要告诉驱动器 信号条件。最后,你必须设置当输入条件满足时转到的 程序行。

这里我们设计 If Input 指令基于以下三个考虑:

1)这条指令可以根据外部条件在程序中跳转。例如, 设想你正在构建一台机器,驱动器的任务是供给下一个 零件。通常驱动器在供给下一零件时只等待半秒时间, 因为这正是切割零件的时间。但有时候处理不同材料的 零件(比如铝、钢)时花费的时间更多,你希望通过触发 一个开关来改变两个零件之间的延迟时间为 1.5 秒。方 法如下:

程序在第4行运行期间输送零件,第5行等待半秒。如 果连接到输入端1的开关处于关闭状态(低电压信号状态),程序将会跳转到第8行,跳过额外的1秒延迟时间。 如果开关处于打开状态,将发生1秒的延迟。你可以在 开关开路的位置标记为"钢",在关闭的位置标记为"铝"。

2)第 2 个原因是这条指令可以根据某输入端口来改 变距离或者速度这样的参数。思考一下上一个例子,当 开关置为"钢"或"铝"时刀具怎样知道运行多快呢?

我们可以将控制转动的开关连接到某个通用输入端。这次我们的程序如下图所示:

÷	If Input 1 low, go to line 7
	CW 1000 steps, 1 rps
←	Go to line 8
	CW 1000 steps, 3 rps
Ŧ	

当驱动器运行到第 4 步时,程序会检测"钢/铝"开关。 如果信号低(铝),程序就跳转到第 7 行: Feed To Length, 此时刀具每秒种转动 3 转。如果信号高(钢),程序不跳转, 继续执行第 5 行的 Feed To Length,此时刀具每秒种转 动 1 转。然后,驱动器由于第 5 行为 Go To 指令跳过第 2 个 Feed 指令。(不要忘了加 Go To 指令否则刀具将停 止两次)。

3) 最后一个原因就是,这样你在100 行的程序空间内可

入做成两个完全不同的功能。假如每个任务需要以编写功能更多的程序。可能你想使您的系统根据某输 4条指令。方法如下:



根据输入端 2 的状态,程序执行第 4-9 行或者执行第 11-15 行。程序最终都返回到第 3 行来再次检测开关的输入条件。

如果你选择 "Multiple Inputs" 那么你可以组合多个输入端,根据"或"或者"与"的逻辑表达式。 例如: "If input 2 low or 3 low or 4 high goto line x"。

注意:如果使用 If Input 指令来退出某个循环,循环不会在下次自动重置,你应该使用 Reset Repeat Loop 指令来重置循环。否则,循环计数器将在循环退出的地方再继续。





#### Call and Return

Call and Return 指令用来创建并调用子程序,它有以下两个好处。首先,如果程序中的某部分出现不止一次,你可以通过创建一个子程序来节省程序行,这样在别的任何地方都可以来调用这个子程序。例如,你希望在每一次转动结束之后有一个"握手"动作。你可以在程序的结尾处加入一个"握手"程序,紧跟着一条 Return 指令。然后你在每一次转动之后都调用这个子程序。这样可以节省程序行以便程序可以完成更多的功能。

子程序也可以帮助您避免程序错误。如果你在程序中已经有很多个地方都使用相同的程序段,如果你 想对某条指令进行修改,你必须修改程序中所有使用这条指令的地方。如果使用子程序,你只须修改一个 地方就行了。

例如:设想你正在输送零件然后进行切割。在每次转动之后你 必须给其它设备发送信号来完成切割,然后等待完成。一个典型的 程序次序如下:

Set Output(告诉其它设备即将进行切割) Wait Time(给其它设备一段响应Set Output 指令的时间) Wait Input(等待其它设备的完成信号)

要创建一个子程序,在程序末尾(比如第 90 行)加入上述三行指令,并加入一条 Return from Subroutine 指令。每次程序完成一个动作,然后调用第 90 行的子程序。

当您编写 Return 指令的程序时, 要确认选中"from Subroutine"选项。



#### Call Stack

每次你调用某个子程序时,驱动器必须记录从什么地方调用的,这样子程序的 Return 指令才会知道 回到什么地方。"返回地址"保存在堆栈中。如果你调用的子程序又调用了另一个程序,另一个子程序的 地址也会推入"栈"中。这样的处理不能永无休止的继续下去,因为堆栈调用只有5层深度。

子程序可以包含了其它的子程序。但不允许超过 5 次没有 Return 的调用。如果你使用中断,你必须预留一级堆栈,因为中断使用一级堆栈。

#### Animation

当处于开发模式(驱动器仍然由 PC 上的 Mis Programmer 应用程序控制),堆栈的调用过程会在状 态栏中显示。这样你可以体会堆栈是如何工作的以及 堆栈是否溢出。



46



#### **Change Current**

电机的电流通常在 Mis Programmer 的主窗体中设置并应用到所有的程序指令中。但倘若你想通过指 令改变电流,例如,在操作员手工调节机械装置或装载一个新的标签卷时,你想临时关掉电机电流。要手 工克服步进电机的保持力矩是很困难的,所以有时候最好完全切断电流。

另一方面,你可能会会想临时增加电机电流以获得更大的力矩,但不能长时间让它以大电流运转防止 过热。

Change Current 指令允许关闭电流,并能恢复到正常设置的电流,或者采用一个新的电流设置。

Change Current	×
🚛 Change Motor Current	
C Disable (no current - motor turns freely)	OK
C Resume normal current (2.24A, 50% idle)	Cancel
New current & idle current settings	
Current	_
3.00 A/phase	
	ו
Idle Current	
	\$





#### Comment

Comment 指令让你在程序中加入注释。这样别人以后要修改你的程序时他们就会明白你是怎么做的, 也可以帮助您自己更好地组织程序。

不管程序保存在磁盘或者下载到驱动器,注释都将保留。注释语句不会影响程序的运行: 当驱动器执 行程序时,它会跳转注释语句。

我们建议你在程序的第一行加入注释,这样以后的程序员就可以知道程序编写的作者,时间,目的。 你也可以在程序中别的地方加入注释。比如就在 Feed To Length 转动之前,你可以加入这样的注释:"转 动零件 1/4 圈"。

现在花少量的时间为程序加入注释可能会为你以后节省大量的时间。

由于程序中有注释数量的限制。驱动器具有一个 400 个字符的 "string pool(字符串池)"。所有 MMI Prompts 文本都保存到这个字符串池中,就象任何 Comment 指令的字符串不能超过 12 个字符一样。

Add Comment 🛛 🗙			
Ì	Add a comment to yo	ur program	
Part feeding program 1/3/97			
Enter	r up to 30 characters	OK	
		Cancel	

### **Command Buttons**

#### Download, Upload & Execute

驱动器的设计支持程序完成并通过测试后脱离主机就能运行。首先,你可能 需要在 PC 上花费大量的时间来运行程序。这样在程序遇到错误时可以快速的作出 修改。

Mis Programmer 软件为驱动器提供四个命令按钮。

Download 通过串口把程序下载到驱动器。传送大概需要3秒时间。执行程序之前必须先下载程序。

#### 注意: 在往驱动器下载程序时, 要确保 JOG 输入端没有被激活。 如果不能确定, 那么拔除 JOG CW 和 JOG CCW 连接件。

**Upload** 让你获取得驱动器的程序并显示到屏幕上。如果你想修改已经存在于驱动器中的程序,你可以使用上传命令将程序传给 PC 机。

**Execute** 让驱动器从第 1 行开始执行内存中的程序。在点击"Execute"按钮后,你会看见一个包含 状态显示以及 5 个程序控制按钮的对话框。

如果驱动器没有连接到 PC 机或者 Mis Programmer 软件没有运行,驱动器从第 1 行开始执行上一次 下载的程序。

按"stop"按钮中断正在执行的程序并关闭执行面板。

Pause 按钮将暂停程 序。但并不关闭执行对话框。 在程序暂停时,输入的状态 显示会不断刷新,这样你可 以实时调节传感器和开关并 观察结果。

Step 按钮执行下一行程 序,然后自动暂停。

按"Run"按钮从你暂 停的地方继续执行。"Reset" 按钮将程序转到第1行。









#### Save, Open, Print & Quit

除了和驱动器交换程序之外,编程软件还可以保存程序到硬盘及从硬盘上打 开已保存的程序,还可以在打印机上打印程序。

Save 按钮可以将程序保存到硬盘上。按下 Save 按钮后会出现一个文件对话框让您输入程序的文件名。

Open 按钮弹出一个对话框显示所有硬盘上的".MIS5"文件。选择其中一个文件,然后点击"OK"按钮加载这个文件。

Print 按钮让您在打印机上打印程序。

Quit 按钮将退出 Mis Programmer 软件。

Save
Open
Print
Quit

### Encoder Feedback: 带编码器选项板的 MS3540MI

大多数步进电机系统运行时都是"Open Loop"(开环)的,也就是说电机位置是由命令给定的,并且 驱动器认为电机精确跟踪。如果系统设计成合理数量的力矩余额,而且在电机的运行轨迹上没有障碍,这 种情况下电机运行良好。

然而,有的应用要求从编码器反馈信息来确认命令是否正确执行。例如,如果您正在加工非常昂贵的 零件,值得在电机控制硬件上多花一些钱,这样即使运行异常的话也不会破坏要处理的零件。另一方面, 使用解码器的反馈来补偿电机定位时的微小偏差可以获得更高精度的定位。

要支持这样的应用, MS3540MI 驱动器借助于 1000-175 编码器选择板可接受积分编码器输入。我们 也提供装有编码器的电机。

如果您选择 MS3540MI 驱动器并打开"steps/rev"对话框,你可以告诉 Mis Programmer 软件使用哪

Microstep Resolution Microstep Resolution OK Cancel O 200 steps/rev (Full) Encoder Feedback O 400 steps/rev (Half) Available with: O 2000 steps/rev (1/10) 3540i with Encoder Option Board 5000 steps/rev (1/25) counts/rev 10,000 steps/rev (1/50) 4000 (1000 lines) 12,800 steps/rev (1/64) Use encoder to verify all 18,000 steps/rev (.02 deg)
 [
 ] moves. If error exceeds 20,000 steps/rev (1/100)
 10 counts (50 steps): O 21,600 steps/rev (1 arc min) C lanore the error. O 25,000 steps/rev (1/125). Correct the error. After O 25,400steps/rev (1/127) 20 tries, ao to line 25 O 25,600 steps/rev (1/128). O 36,000 steps/rev (.01 deg) Correction Move 50,000 steps/rev (1/250) Speed 5.00000 Rev/sec 50,800 steps/rev (1/254) 100 Accel Rev/s/s O Custom: 8000 steps/rev

一种编码器以及在出现错误时执行什么动 作。

首先要做的事就是输入编码器的 counts/rev。很多编码器的分辨率是以"线" 为单位的。由于Mis驱动器使用X4积分解码, counts/rev为"线"的4倍。

你可以使用任何编码器,只要电机的步数/转能被 counts/rev 整除。在左边的对话框中,系统设置成 20,000 步/转, 4000counts/rev。20,000 / 4000 = 5,是一个 整数,所以这样的组合是可以的。你也可以 对此编码器选择 36,000 步/转,但不能是 12,800。

接下来,输入可以承受的最大位置误差。在这个例子中我们使用10个编码器输出单位,也就是 0.0025英寸。如果在Feed To Length, Feed and Return或者Feed To Position指令执行完之后,误差超过0.0025

英寸,控制会自动通过系列短距离的移动来尝试纠正错误。控制也会在Wait Input和MMI Prompt指令期间 完成静态位置维持。静态位置维持用于修正由于外界干扰而引起的电机移动。在障碍或者不可抗力的作用 下,可能无法正常修正误差。在这样的情况下,你可以让控制器放弃修正并转到程序中的其它部分。这样, 你就建立了 "error recovery(误差恢复)"程序。左图显示了一个简单的误差修正程序的例子。 我们在 steps/rev 对话框中输入 10 作为编码器错误跳转行。误差恢复行自动显示成淡蓝色,这样在阅读程序时就可以很容易看出来。如果你的使用手册是打印出来的,误差修正行可能显示为灰色的。



在这个例子中,当位置误差发 生时,控制器尝试最大到 20 次来修 正误差,然后放弃并跳转到第 10 行。在第 10 行,我们可以通过 MMI Prompt 指令告诉操作员纠正错误。

#### 修正时移动速度和加速度

**你可以通过在**microstep/ encoder 对话框中指定速度和加速 度来纠正误差。

一旦产生错误跳转时,自动纠正处 理会关闭,直到使用 Go To 或者 If Input 指令再次跳转。这样,控制器 会在误差恢复子程序中停止尝试纠 正误差,一旦问题解决并跳转回来

的时候会自动恢复误差纠正功能。

如果是 MS3540MI 的驱动器,你又没有使用编码器,选择"Ignore the error."选项。这样会关闭自动纠错及错误跳转功能。如果您使用编码器,选择"Correct the error"。

注意:如果使用 eocoder feedback,我们不推荐使用 "0% idle current reduction"。