

Logix5000 控制器运动控制 指令集



参考手册

产品目录号 1756-L1M1、1756-L1M2、1756-L1M3、1756-L55M12、1756-L55M13、1756-L55M14、1756-L55M16、1756-L55M22、1756-L55M23、1756-L55M24、1756-L60M03SE、1756-L61、1756-L62、1756-L63、1756-L64、1768-L43、1789-L60、1789-20D

重要用户须知

固态设备具有不同于机电设备的运行特性。《固态控制设备的应用、安装与维护安全指南》（出版号 [SGI-1.1](#)，本资料可从当地的罗克韦尔自动化销售处或通过 <http://www.rockwellautomation.com> 在线获得）说明了固态设备和硬接线电动机械设备之间的重要差别。由于这些差别的存在以及固态设备应用的多样性，因此所有技术人员有责任确保这些固态设备的每项应用是可行的。





对于因使用该设备而引起的间接或由此引发的损失，罗克韦尔自动化有限公司不承担任何责任或义务。

本手册所包含的示例和图表仅仅用于说明。因为任何特定安装有着特定的变化因素和要求，所以罗克韦尔自动化有限公司不承担用于基于示例和图表中实际应用的任何责任或义务。

关于本手册中所说明的信息、电路、设备或软件，罗克韦尔自动化有限公司不承担任何专利责任。

如果没有得到罗克韦尔自动化有限公司书面允许，严禁任何团体、公司、个人对本手册的内容进行整体或部分复制。

在本手册中，我们通过说明提醒用户有关安全注意事项。

警告 	是指有关在危险环境中引起爆炸而导致人身伤害、财产损失或经济损失的实际应用或环境情况的信息。
重要提示	是指用户对有关产品正确应用和理解所需掌握的重要信息。
注意 	是指有关导致人员伤亡、财产损失或经济损失的实际应用或环境情况的信息。注意事项的标志可帮助人们识别危险，避免危险，并且让人认识到危险所带来的后果
有电危险 	变频器或电机的表面或内部必须贴有相关的标签，警告人们可能存在危险电压。
高温危险 	变频器或电机的表面或内部必须贴有相关的标签，警告人们设备表面可能具有高温危险。

艾伦-布拉德利（Allen-Bradley）、罗克韦尔自动化和 TechConnect 为罗克韦尔自动化有限公司的标志。

不属于罗克韦尔自动化有限公司的标志为其各自公司的财产。

何处查找指令

使用这个定位器可查找关于 Logix 指令的参考详细信息（灰显的指令在其他手册中提供）。这个定位器同时还列出哪些编程语言可用于该指令。

如果定位器列出	指令位于
页码	本手册
概述	Logix5000 控制器通用指令集参考手册， 1756-RM003
过程控制	Logix5000 控制器过程控制和驱动器指令集参考手册， 1756-RM006
阶段	Logix5000 控制器 PhaseManager 用户手册， LOGIX-UM001

指令	位置	语言
ABL 缓冲区行的 ASCII 测试	概述	梯形图 结构文本
ABS 绝对值	概述	梯形图 结构文本 功能块
ACB 缓冲区中的 ASCII 字符	概述	梯形图 结构文本
ACL ASCII 清除缓冲区	概述	梯形图 结构文本
ACOS 反余弦	概述	结构文本
ACS 反余弦	概述	梯形图 功能块
ADD 加法	概述	梯形图 结构文本 功能块
AFI 恒假指令	概述	梯形图
AHL ASCII 握手行	概述	梯形图 结构文本
ALMA 模拟报警	概述	梯形图 结构文本 功能块
ALM 报警	过程控制	结构文本 功能块
ALMD 数字报警	概述	梯形图 结构文本 功能块
AND 按位与	概述	梯形图 结构文本 功能块
ARD ASCII 读取	概述	梯形图 结构文本
ARL ASCII 读取行	概述	梯形图 结构文本

指令	位置	语言
ASIN 反正弦	概述	结构文本
ASN 反正弦	概述	梯形图 功能块
ATAN 反正切	概述	结构文本
ATN 反正切	概述	梯形图 功能块
AVE 文件平均	概述	梯形图
AWA ASCII 写入附加	概述	梯形图 结构文本
AWT ASCII 写入	概述	梯形图 结构文本
BAND 布尔与	概述	结构文本 功能块
BNOT 布尔非	概述	结构文本 功能块
BOR 布尔或	概述	结构文本 功能块
BRK 中断	概述	梯形图
BSL 位左移	概述	梯形图
BSR 位右移	概述	梯形图
BTD 位域分配	概述	梯形图
BTDT 位域分配及目标	概述	结构文本 功能块
BTR 消息	概述	梯形图 结构文本

指令	位置	语言
BTW 消息	概述	梯形图 结构文本
BXOR 布尔异或	概述	结构文本 功能块
CC 协调控制	过程控制	结构文本 功能块
CLR 清零	概述	梯形图 结构文本
CMP 比较	概述	梯形图
CONCAT 字符串连接	概述	梯形图 结构文本
COP 复制文件	概述	梯形图 结构文本
COS 余弦	概述	梯形图 结构文本 功能块
CPS 同步复制文件	概述	梯形图 结构文本
CPT 计算	概述	梯形图
CTD 递减	概述	梯形图
CTU 递增	概述	梯形图
CTUD 递增 / 递减	概述	结构文本 功能块
D2SD 离散型_2 态设备	过程控制	结构文本 功能块
D3SD 离散型_3 态设备	过程控制	结构文本 功能块
DDT 诊断检测	概述	梯形图
DEDT 死区时间	过程控制	结构文本 功能块
DEG 度	概述	梯形图 结构文本 功能块
DELETE 字符串删除	概述	梯形图 结构文本
DERV 微分	过程控制	结构文本 功能块
DFF D 触发器	过程控制	结构文本 功能块
DIV 除法	概述	梯形图 结构文本 功能块

指令	位置	语言
DTOS DINT 转换为 String	概述	梯形图 结构文本
DTR 数据转换	概述	梯形图
EOT 转换结束	概述	梯形图 结构文本
EQU 等于	概述	梯形图 结构文本 功能块
ESEL 增强型选择	过程控制	结构文本 功能块
EVENT 触发事件任务	概述	梯形图 结构文本
FAL 文件算术和逻辑	概述	梯形图
FBC 文件位比较	概述	梯形图
FFL FIFO 加载	概述	梯形图
FFU FIFO 卸载	概述	梯形图
FGEN 函数发生器	过程控制	结构文本 功能块
FIND 查找字符串	概述	梯形图 结构文本
FLL 文件填充	概述	梯形图
FOR 有关	概述	梯形图
FRD 转换为整数	概述	梯形图 功能块
FSC 文件搜索和比较	概述	梯形图
GEQ 大于或等于	概述	梯形图 结构文本 功能块
GRT 大于	概述	梯形图 结构文本 功能块
GSV 获取系统值	概述	梯形图 结构文本
HLL 上限 / 下限	过程控制	结构文本 功能块
HPF 高通滤波器	过程控制	结构文本 功能块
ICON 输入线连接器	过程控制	功能块

指令	位置	语言
IMC 内部型号控制	过程控制	结构文本 功能块
INSERT 插入字符串	概述	梯形图 结构文本
INTG 积分器	过程控制	结构文本 功能块
IOT 立即输出	概述	梯形图 结构文本
IREF 输入参考	过程控制	功能块
JKFF JK 触发器	过程控制	结构文本 功能块
JMP 跳转到标签	概述	梯形图
JSR 跳转到子例程	概述	梯形图 结构文本 功能块
JXR 跳转到外部例程	概述	梯形图
LBL 标签	概述	梯形图
LDL2 二阶超前滞后	过程控制	结构文本 功能块
LDLG 超前 / 滞后	过程控制	结构文本 功能块
LEQ 小于或等于	概述	梯形图 结构文本 功能块
LES 小于	概述	梯形图 结构文本 功能块
LFL LIFO 加载	概述	梯形图
LFU LIFO 卸载	概述	梯形图
LIM 限制	概述	梯形图 功能块
LN 自然对数	概述	梯形图 结构文本 功能块
LOG 以 10 为底的对数	概述	梯形图 结构文本 功能块
LOWER 小写字母	概述	梯形图 结构文本
LPF 低通滤波器	过程控制	结构文本 功能块

指令	位置	语言
MAAT 应用轴整定参数	228	梯形图 结构文本
MAFR 运动轴故障复位	55	梯形图 结构文本
MAG 电子齿轮	100	梯形图 结构文本
MAHD 应用轴连接诊断	239	梯形图 结构文本
MAH 运动轴寻参	72	梯形图 结构文本
MAJ 轴速度	76	梯形图 结构文本
MAM 轴位移	88	梯形图 结构文本
MAOC 开启运动凸轮输出	196	梯形图 结构文本
MAPC 运动轴位置凸轮	128	梯形图 结构文本
MAR 开启轴位置抓拍	186	梯形图 结构文本
MASD 关闭运动轴	45	梯形图 结构文本
MAS 运动轴停止	58	梯形图 结构文本
MASR 运动轴关闭复位	48	梯形图 结构文本
MATC 运动轴时间凸轮	150	梯形图 结构文本
MAVE 移动平均值	过程控制	结构文本 功能块
MAW 开启运动监视	180	梯形图 结构文本
MAXC 捕捉最大值	过程控制	结构文本 功能块
MCD 运动动态调整	111	梯形图 结构文本
MCCD 运动插补动态调整	338	梯形图 结构文本
MCCM 运动圆弧插补	292	梯形图 结构文本
MCLM 运动直线插补	266	梯形图 结构文本
MCCP 计算凸轮运动曲线	123	梯形图 结构文本

指令	位置	语言
MCR 主控复位	概述	梯形图
MCT 运动插补坐标转换	362	梯形图 结构文本
MCTP 计算坐标转换位置	375	梯形图 结构文本
MDF 直接关运动使能	53	梯形图 结构文本
MDOC 关闭运动凸轮输出	223	梯形图 结构文本
MDO 运动直接使能	50	梯形图 结构文本
MDR 关闭轴位置抓拍	193	梯形图 结构文本
MDW 关闭运动监视	184	梯形图 结构文本
MEQ 掩码等于	概述	梯形图 结构文本 功能块
MGSD 关闭运动组	171	梯形图 结构文本
MGS 运动组停止	166	梯形图 结构文本
MGSP 抓拍运动组位置	176	梯形图 结构文本
MGSR 运动组关闭复位	174	梯形图 结构文本
MID 中间字符串	概述	梯形图 结构文本
MINC 捕捉最小值	过程控制	结构文本 功能块
MMC 模块多变量控制	过程控制	结构文本 功能块
MOD 模数	概述	梯形图 结构文本 功能块
MOV 位移	概述	梯形图
MRAT 运行轴整定	233	梯形图 结构文本
MRHD 运行轴连接诊断	243	梯形图 结构文本

指令	位置	语言
MRP 运动位置重设	118	梯形图 结构文本
MSF 关运动伺服使能	42	梯形图 结构文本
MSG 消息	概述	梯形图 结构文本
MSO 开运动伺服使能	39	梯形图 结构文本
MSTD 移动标准偏差	过程控制	结构文本 功能块
MUL 乘	概述	梯形图 结构文本 功能块
MUX 多路切换器	过程控制	功能块
MVM 屏蔽码位移	概述	梯形图
MVMT 屏蔽码位移带目标	概述	结构文本 功能块
NEG 求反	概述	梯形图 结构文本 功能块
NEQ 不等于	概述	梯形图 结构文本 功能块
NOP 无动作	概述	梯形图
NOT 按位非	概述	梯形图 结构文本 功能块
NTCH 陷波滤波器	过程控制	结构文本 功能块
OCON 输出线连接器	过程控制	功能块
ONS 单触发	概述	梯形图
OR 按位或	概述	梯形图 结构文本 功能块
OREF 输出参考	过程控制	功能块
OSFI 带输入的下降沿单触发	概述	结构文本 功能块
OSF 下降沿单触发	概述	梯形图
OSRI 带输入的上升沿单触发	概述	结构文本 功能块
OSR 上升沿单触发	概述	梯形图

指令	位置	语言
OTE 输出激励	概述	梯形图
OTL 输出锁存	概述	梯形图
OTU 输出解锁存	概述	梯形图
PATT 连接到设备阶段	阶段	梯形图 结构文本
PCLF 设备阶段清零失败	阶段	梯形图 结构文本
PCMD 设备阶段命令	阶段	梯形图 结构文本
PDET 从设备阶段断开	阶段	梯形图 结构文本
PFL 设备阶段失败	阶段	梯形图 结构文本
PIDE 增强型 PID	过程控制	结构文本 功能块
PID 比例 / 积分 / 微分	概述	梯形图 结构文本
PI 比例 + 积分	过程控制	结构文本 功能块
PMUL 脉冲乘法器	过程控制	结构文本 功能块
POSP 位置比例	过程控制	结构文本 功能块
POVR 设备阶段覆盖命令	阶段	梯形图 结构文本
PPD 设备阶段已暂停	阶段	梯形图 结构文本
PRNP 设备阶段新参数	阶段	梯形图 结构文本
PSC 阶段状态完成	阶段	梯形图 结构文本
PXRQ 设备阶段外部请求	阶段	梯形图 结构文本
RAD 弧度	概述	梯形图 结构文本 功能块
RESD 复位优先	过程控制	结构文本 功能块
RES 复位	概述	梯形图

指令	位置	语言
RET 返回	概述	梯形图 结构文本 功能块
RLIM 变化率限制器	过程控制	结构文本 功能块
RMPS 斜坡 / 渗透	过程控制	结构文本 功能块
RTO 保持计时器打开	概述	梯形图
RTOR 带有复位的保持计时器 打开	概述	结构文本 功能块
RTOS 实数转换为字符串	概述	梯形图 结构文本
SBR 子例程	概述	梯形图 结构文本 功能块
SCL 数据整定	过程控制	结构文本 功能块
SCRV S 曲线形	过程控制	结构文本 功能块
SEL 选择	过程控制	功能块
SETD 置位优先	过程控制	结构文本 功能块
SFP SFC 暂停	概述	梯形图 结构文本
SFR SFC 复位	概述	梯形图 结构文本
SIN 正弦	概述	梯形图 结构文本 功能块
SIZE 元素大小	概述	梯形图 结构文本
SNEG 取反选择	过程控制	结构文本 功能块
SOC 二阶控制器	过程控制	结构文本 功能块
SQI 定序程序输入	概述	梯形图
SQL 定序程序加载	概述	梯形图
SQO 定序程序输出	概述	梯形图
SQR 平方根	概述	梯形图 功能块
SQRT 平方根	概述	结构文本

指令	位置	语言
SRT 文件排序	概述	梯形图 结构文本
SRTP 脉宽调制	过程控制	结构文本 功能块
SSUM 求和选择	过程控制	结构文本 功能块
SSV 设置系统值	概述	梯形图 结构文本
STD 文件标准偏差	概述	梯形图
STOD String 转换为 DINT	概述	梯形图 结构文本
STOR 字符串转换为实数	概述	梯形图 结构文本
SUB 减法	概述	梯形图 结构文本 功能块
SWPB 字节交换	概述	梯形图 结构文本
TAN 正切	概述	梯形图 结构文本 功能块
TND 临时结束	概述	梯形图
TOD 转换为 BCD	概述	梯形图 功能块
TOFR 带复位的延时断开计时器	概述	结构文本 功能块
TOF 延时断开计时器	概述	梯形图
TONR 带复位的延时导通计时器	概述	结构文本 功能块
TON 延时导通计时器	概述	梯形图
TOT 累加器	过程控制	结构文本 功能块
TRN 截断	概述	梯形图 功能块
TRUNC 截断	概述	结构文本
UID 禁止用户中断	概述	梯形图 结构文本
UIE 允许用户中断	概述	梯形图 结构文本
UPDN 增 / 减累加器	过程控制	结构文本 功能块

指令	位置	语言
UPPER 大写字母	概述	梯形图 结构文本
XIC 检查是否已关闭	概述	梯形图
XIO 检查是否已打开	概述	梯形图
XOR 按位异或	概述	梯形图 结构文本 功能块
XPY X 的 Y 次方	概述	梯形图 结构文本 功能块


指令定位器	何处查找指令	3
目录		
前言	简介	13
	本手册适用对象	14
	本手册的目的	14
	顺序功能图 (SFC)	15
	惯例和相关术语	16
	其他资源	17
	章节 1	
运动概念	简介	19
	指令计时	19
	编写一个速度模式程序	24
	选择命令	33
	章节 2	
运动控制状态指令	简介	37
(MSO、MSF、MASD、	开运动伺服使能 (MSO)	39
MASR、MDO、MDF、MAFR)	关运动伺服使能 (MSF)	42
	关闭运动轴 (MASD)	45
	运动轴关闭复位 (MASR)	48
	运动直接使能 (MDO)	50
	直接关运动使能 (MDF)	53
	运动轴故障复位 (MAFR)	55
	章节 3	
运动位移指令	简介	57
(MAS、MAH、MAJ、MAM、	运动轴停止 (MAS)	58
MAG、MCD、MRP、MCCP、	运动轴寻参 (MAH)	72
MAPC、MATC、MCSV)	轴速度 (MAJ)	76
	轴位移 (MAM)	88
	电子齿轮 (MAG)	100
	运动动态调整 (MCD)	111
	运动位置重设 (MRP)	118
	计算凸轮运动曲线 (MCCP)	123
	运动轴位置凸轮 (MAPC)	128
	运动轴时间凸轮 (MATC)	150
	计算从轴运动值 (MCSV)	163
	章节 4	
运动组指令	简介	165
(MGS、MGSD、MGSR、	运动组停止 (MGS)	166
MGSP)	关闭运动组 (MGSD)	171
	运动组关闭复位 (MGSR)	174
	抓拍运动组位置 (MGSP)	176

	章节 5	
运动事件指令 (MAW、MDW、MAR、MDR、	简介	179
MAOC、MDOC)	开启运动监视 (MAW)	180
	关闭运动监视 (MDW)	184
	开启轴位置抓拍 (MAR)	186
	关闭轴位置抓拍 (MDR)	193
	开启运动凸轮输出 (MAOC)	196
	规划输出模块	216
	关闭运动凸轮输出 (MDOC)	223
	章节 6	
运动配置指令 (MAAT、MRAT、MAHD、	简介	227
MRHD)	应用轴整定参数 (MAAT)	228
	运行轴整定 (MRAT)	233
	应用轴连接诊断 (MAHD)	239
	运行轴连接诊断 (MRHD)	243
	章节 7	
运动协调指令 (MCLM、MCCM、MCCD、	简介	249
MCS、MCSD、MCT、	在混合指令时使用不同的终止类型	251
MCTP、MCSR)	选择终止类型	257
	共线运动的速度模式	260
	对称模式	262
	如何得到切线速度模式	264
	以不同速度混合运动	265
	运动直线插补 (MCLM)	266
	运动圆弧插补 (MCCM)	292
	运动插补动态调整 (MCCD)	338
	运动插补停止 (MCS)	349
	运动插补关闭 (MCSD)	358
	运动插补坐标转换 (MCT)	362
	计算坐标转换位置 (MCTP)	375
	章节 8	
整定 S 曲线模式	简介	387
	执行时机	387
	准备事宜	388
	程序	388
	其他资源	389
	附录 A	
运动控制指令错误代码 (ERR)	附加错误代码信息	399
	附录 B	

运动相关的数据类型（结构）	简介	401
	CAM 结构	401
	CAM_PROFILE 结构	402
	MOTION_GROUP 结构	403
	MOTION_INSTRUCTION 数据类型	404
	OUTPUT_CAM 结构	405
	OUTPUT_COMPENSATION 结构	407
	附录 C	
结构文本编程	简介	409
	结构文本语法	409
	赋值	411
	表达式	413
	指令	420
	构造	421
	IF...THEN	422
	CASE...OF	425
	FOR...DO	428
	WHILE...DO	431
	REPEAT...UNTIL	434
	注释	437
索引	罗克韦尔自动化支持	12

简介

本手册是关于 Logix5000 的指令手册之一。

任务 / 目标	文档
为序列应用进行控制器编程	Logix5000 控制器通用指令集参考手册, 出版号 1756-RM003
为过程或变频器应用进行控制器编程	《Logix5000 控制器过程控制和驱动器指令集参考手册》, 出版号 1756-RM006
为运动控制应用进行控制器编程	《Logix5000 控制器运动控制指令集》, 出版号 1756-RM007
您的当前位置 	
为使用设备阶段进行控制器编程	《PhaseManager 用户手册》, 出版号 LOGIX-UM001
将文本文件或标记导入项目	Logix5000 控制器导入 / 导出参考手册, 出版号 1756-RM084
将项目或标记导出到文本文件	
将 PLC-5 或 SLC 500 应用转换为 Logix5000 应用	Logix5000 控制器 PLC-5 或 SLC 500 逻辑到 Logix5550 逻辑转换参考手册, 出版号 1756-6.8.5
为运动轴、坐标系和运动控制模块配置控制器	《Logix5000 控制系统用户手册》, 出版号 LOGIX-UM002。

您可以使用以下 Logix5000 控制器进行运动控制：

- 1756 ControlLogix 控制器
- 1756 GuardLogix 控制器 (版本 16 及更高版本)
- 1768 CompactLogix 控制器 (版本 15 及更高版本)
- 1789 SoftLogix5800 控制器
- 20D PowerFlex 700S 带 DriveLogix 控制器

如果您拥有 PowerFlex 700S 变频器带 DriveLogix 控制器

您不能在 DriveLogix 控制器上使用以下指令：

- 运动直接使能 (MDO)
- 直接关运动使能 (MDF)
- 应用轴整定参数 (MAAT)
- 运行轴整定 (MRAT)
- 应用轴连接诊断 (MAHD)
- 运行轴连接诊断 (MRHD)

本手册适用对象

本手册为程序员提供可用于 Logix5000 控制器的运动控制指令的相关详细信息。您应该已经熟悉 Logix5000 控制器如何存储和处理数据。

新手程序员在开始使用指令之前，应阅读每条指令的所有内容。有经验的程序员可以参考指令信息以验证详情。

本手册的目的

本手册提供每条运动控制指令的相关信息。

这个部分	提供这个类型的信息
指令名称	识别指令。 定义指定是输入指令还是输出指令。
操作数	列出指令的所有操作数。
结构文本	介绍采用结构文本格式的操作数的使用。
运动控制指令结构	列出指令的控制状态位和值（如果有）。
描述	介绍指令的使用。 定义指令被使能和被禁止（如果适用）之间的区别。
算术状态标志	定义指令是否会影响算术状态标志。
故障条件	定义指令是否会生成轻微或严重故障。 如果是，定义故障类型和代码。
错误代码	列出并定义适用的错误代码。
状态位	列出受影响的状态位、其状态和定义。
示例	提供至少一个编程示例。 包括解释每个示例的介绍。

顺序功能图 (SFC)

顺序功能图是控制机器或过程的流程图。SFC 使用步序和过渡来指定指定的操作或动作。您可以使用 SFC 来：

- 组织您的系统的功能技术参数。
- 使用一系列步序和过渡来编程并控制您的系统。

通过使用顺序功能图 (SFC) 可获得以下优势。

- 过程图形化分割成大的逻辑片段。
- 加快各个逻辑片段的重复执行。
- 简化屏幕显示。
- 缩短设计和调试程序的时间。
- 故障处理更快、更简单。
- 直接访问机器出现故障的逻辑点。
- 增强和更新更加容易。

有关如何编程和使用 SFC 的详细信息，请参阅《Logix5000 控制器通用程序编程手册》，出版号 1756-PM001。

惯例和相关术语

设置和清除

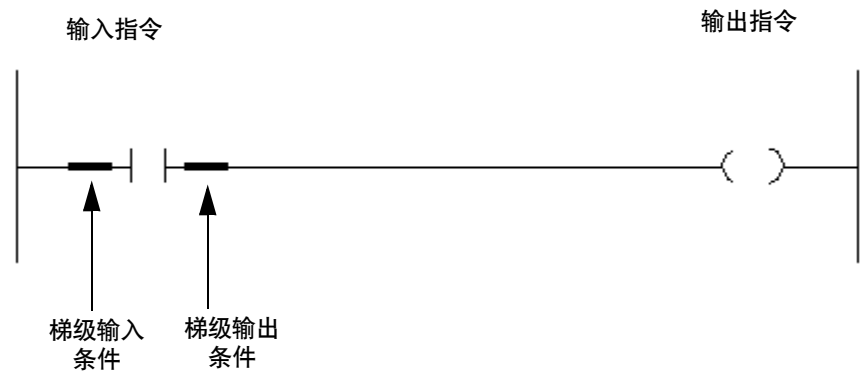
本手册使用设置和清除来定义位（布尔值）和值（非布尔值）的状态：

这个术语	含义是
设置	位被设置为 1(开) 值被设置为任何非零值
清除	位被清除为 0(关) 值内的所有位被清除为 0

如果指令的所有操作数都是用相同的最佳数据类型（通常为 DINT 或 REAL），则指令的执行会更快，需要内存更少。

梯级条件

控制器基于指令前的梯级条件（梯级输入条件）来评估梯形指令。根据梯级输入条件和指令，控制器设置指令后的梯级条件（梯级输出条件），进而影响任何后续指令。



如果某个输入指令的梯级输入条件为真，控制器将评估指令，并根据指令的结果设置梯级输出条件。如果指令执行结果为真，则梯级输出条件为真；如果指令执行结果为假，则梯级输出条件为假。

其他资源

请参阅这些手册和文档，了解关于 Logix5000 控制系统中使用运动控制模块的更多信息。

出版号	出版号
Logix5000 控制器快速入门	1756-QS001
Logix5000 控制器通用程序编程手册	1756-PM001
Logix5000 控制器运动控制指令参考手册	1756-RM007
Logix5000 控制器通用指令集参考手册	1756-RM003
Logix5000 控制器过程控制和驱动器指令集参考手册	1756-RM006
Logix5000 控制系统中的运动控制模块	LOGIX-UM002
运动轨迹规划应用解决方案	RA-AP031
PhaseManager 用户手册	LOGIX-UM001
ControlLogix 控制器用户手册	1756-UM001
CompactLogix 控制器用户手册	1768-UM001
模拟量编码器 (AE) 伺服模块安装说明	1756-IN047
ControlLogix SERCOS 接口模块安装指南	1756-IN572
CompactLogix SERCOS 接口模块安装指南	1768-IN005
1394 SERCOS 接口多轴运动控制系统安装手册	1394-IN002
1394 SERCOS 集成手册	1394-IN024
Ultra3000 数字式伺服驱动器安装手册	2098-IN003
Ultra3000 数字伺服驱动器集成手册	2098-IN005
Kinetix 6000 安装手册	2094-IN001
Kinetix 6000 集成手册	2094-IN002
8720MC 高性能变频器安装手册	8720MC-IN001
8720MC 高性能变频器集成手册	8720MC-IN002

注意：

运动概念

简介

本章介绍对于所有运动控制指令都通用的概念。

欲了解相关信息	参见页码
指令计时	19
编写一个速度模式程序	24
选择命令	33

指令计时

运动控制指令使用三种类型的计时顺序。

计时类型	描述
立即	指令在一次扫描内完成。
消息	指令在几次扫描内完成，因为指令向伺服模块发送消息。
过程	指令的完成可能需要不定长的时间。

立即类型的指令

立即类型的运动控制指令在一次扫描中执行至完成。如果控制器在执行这些指令的过程中检测到错误，则设置状态位，操作结束。

立即类型的指令示例包括：

- 运动动态调整 (MCD) 指令
- 抓拍运动组位置 (MGSP) 指令

立即指令的工作方式如下所述：

1. 当包含运动控制指令的梯级变为真时，控制器将：
 - 设置启用 (.EN) 位。
 - 清除完成 (.DN) 位。
 - 清除错误 (.ER) 位。

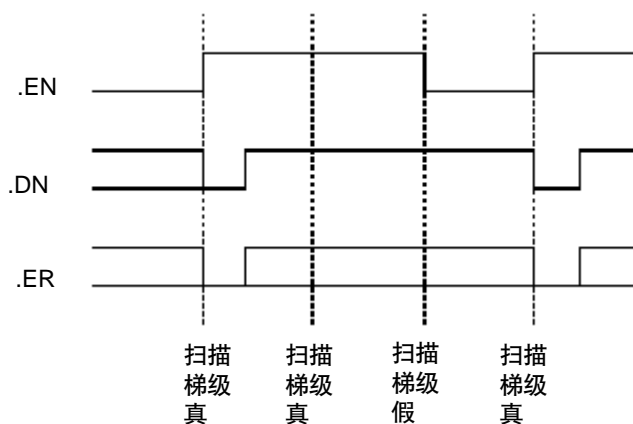
控制器完全执行该指令。

2.

如果控制器	那么
在指令执行时没有检测到错误	控制器设置 .DN 位。
在指令执行时检测到错误	控制器设置 .ER 位，并在控制结构中存储一个错误代码。

3. 在 .DN 或 .ER 位设置以后，下次该梯级变为假时，控制器将清除 .EN 位。

4. 当该梯级变成真时，控制器可以再次执行该指令。



立即类型的指令 – 梯级条件

消息类型的指令

消息类型的运动控制指令向伺服模块发送一条或多条消息。

消息类型的指令示例包括：

- 运动直接使能 (MDO) 指令
- 运动位置重设 (MRP) 指令

消息类型的指令工作方式如下所述：

1. 当包含运动控制指令的梯级变为真时，控制器将：
 - 设置启用 (.EN) 位。
 - 清除完成 (.DN) 位。
 - 清除错误 (.ER) 位。

2. 控制器通过建立发往伺服模块的消息请求，开始执行指令。

指令的剩余部分与程序扫描同时执行。

3. 控制器检查伺服模块是否准备可以接收新消息。
4. 控制器将检查结果放置在控制结构的消息状态字内。
5. 当模块就绪后，控制器构造消息并将其发送到模块。

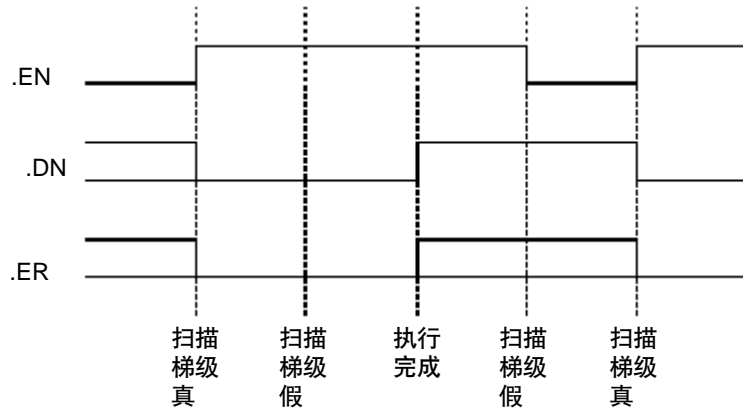
如果该指令需要多个消息，这个过程可能重复多次。

- 6.

如果控制器	那么
在指令执行时没有检测到错误	如果发往模块的所有消息都已完成，则控制器设置 .DN 位。
在指令执行时检测到错误	控制器设置 .ER 位，并在控制结构中存储一个错误代码。

7. 在 .DN 或 .ER 位设置以后，下次该梯级变为假时，控制器将清除 .EN 位。

8. 当梯级变为真时，控制器可以再次执行该指令。



消息类型的指令 – 梯级条件

过程类型的指令

过程类型的运动控制指令发起的运动过程可能需要不定长的时间来完成。

过程类型的指令示例包括：

- 开启运动位置监视 (MAW) 指令
- 轴位移 (MAM) 指令

过程类型的指令工作方式如下所述：

1. 当包含运动控制指令的梯级变为真时，控制器将：
 - 设置启用 (.EN) 位。
 - 清除完成 (.DN) 位。
 - 清除错误 (.ER) 位。
 - 清除过程完成 (.PC) 位。
 - 设置正在进行 (.IP) 位。
2. 控制器发起运动过程。

3.

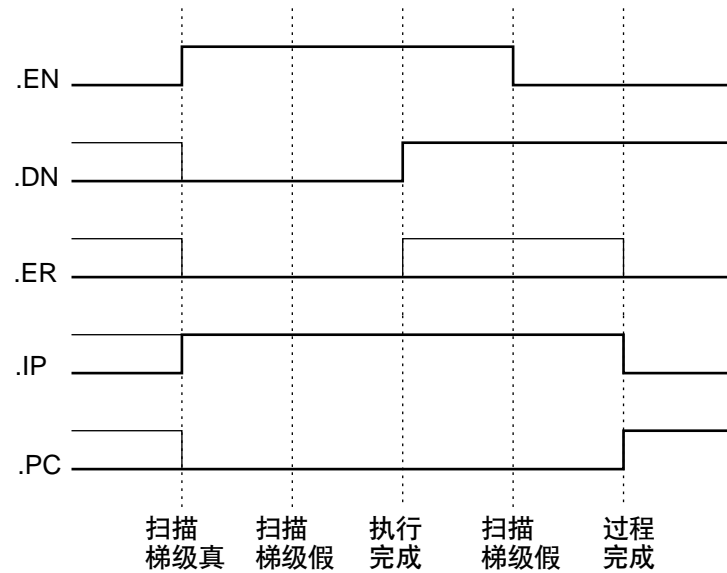
如果	则控制器
控制器在指令执行时没有检测到错误	<ul style="list-style-type: none"> • 设置 .DN 位。 • 设置正在进行 (.IP) 位。
控制器在指令执行时检测到错误	<ul style="list-style-type: none"> • 设置 .ER 位。 • 在控制结构中存储一个错误代码。
控制器检测到另一个运动控制指令的实例	清除该实例的 .IP 位。
运动过程到达可再次执行该指令的点	设置 .DN 位。 对于有些过程类型的指令，例如 MAM，这会在第一次扫描时发生。而对于其他指令，例如 MAH，.DN 位将直到整个归零过程完成后才会被设置。
在运动过程中发生下面一种情况： <ul style="list-style-type: none"> • 运动过程完成 • 另外一个指令的实例执行 • 另外一个指令停止该运动过程 • 运动故障停止该运动过程 	清除 .IP 位。

4. 在运动过程的发起之后，程序扫描可以继续。

指令的剩余部分和控制过程与程序扫描同时继续进行。

5. 在 .DN 位或 .ER 位设置以后，下次该梯级变为假时，控制器将清除 .EN 位。

6. 当梯级变为真时，指令可再次执行。



过程类型的指令 – 梯级条件

编写一个速度模式程序

您可以为各种指令使用这些运动模式中任何一种：

- 梯形模式，用于线性加速和减速
- S 曲线模式，用于控制跃度

有关	参见页码
跃度的定义	24
选择模式	25
在最初的跃度编程中使用时间 %	26
速度模式效果	27
跃度值计算	28

跃度的定义

跃度是指加速度或减速度的变化率。

跃度参数仅适用于使用以下指令的 S 曲线模式位移：

- MAJ
- MAM
- MAS
- MCD
- MCS
- MCCD
- MCCM
- MCLM

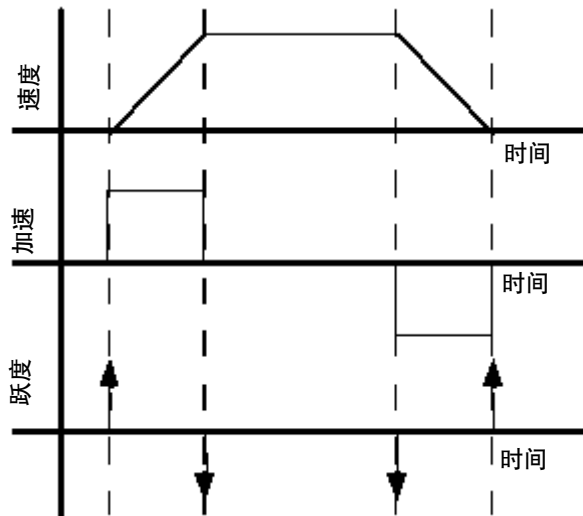
示例：如果加速度在 0.2 秒内从 0 变成 40 mm/s^2 ，则跃度为：

$$(40 \text{ mm/s}^2 - 0 \text{ mm/s}^2) / 0.2 \text{ s} = 200 \text{ mm/s}^3$$

选择模式

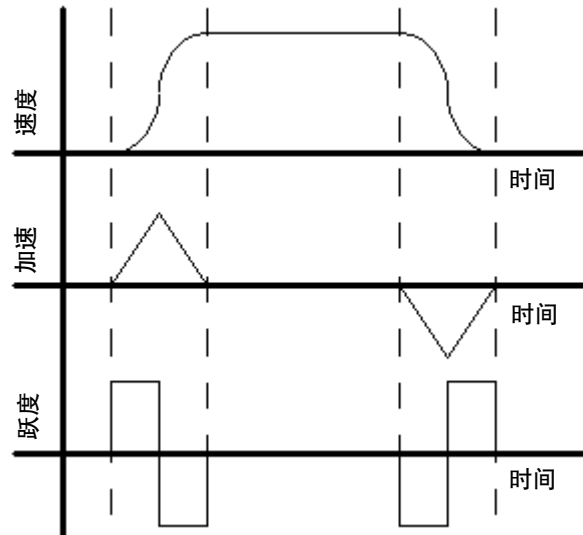
当您选择一种模式时，请考虑循环时间和平滑度。

如果您需要	请选择这种模式	注意事项
<ul style="list-style-type: none"> • 最快的加速和减速时间 • 在编写后续运动程序中有更大的灵活度 	梯形	<p>跃度不会限制加速和减速时间：</p> <ul style="list-style-type: none"> • 加速和减速速度控制着速度的最大变化。 • 您的设备和负荷上的压力会比 S 区域模式要高。 • 跃度被认为是无限的，显示为一条垂直线。



平滑的加速和减速可降低对设备和负荷上的压力

S 曲线



跃度限制加速和减速时间：

- 其加速和减速要比梯形模式时间要长。
- 如果指令使用S曲线模式，控制器在您启动指令时计算加速、减速和跃度。
- 控制器计算三角形加速和减速模式。

在最初的跃度编程中使用时间 %

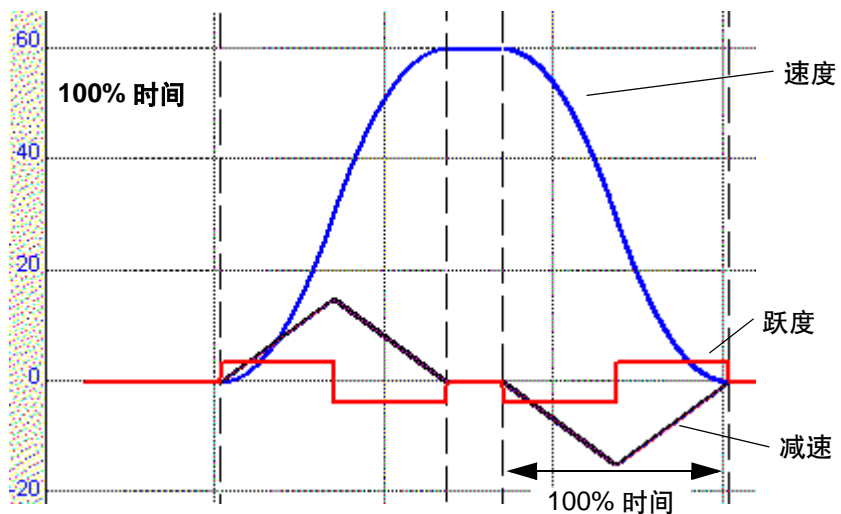
时间使用 % 用来指定有多少加速或减速时间具有跃度。您不需要计算实际的跃度值。

示例

模式

100% 时间

在 100% 时间处，加速或减速将改变轴加快或减慢的整个时间。

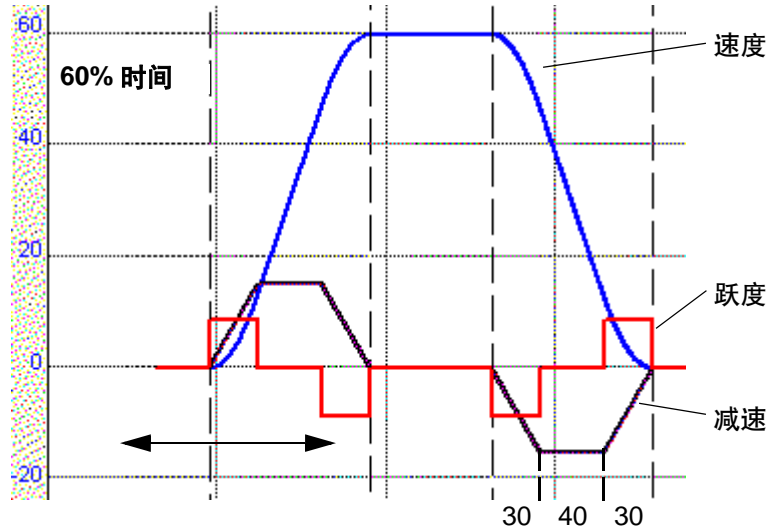


示例

模式

60% 时间

在 60% 时间处，加速或减速将改变轴加快或减慢的 60% 时间。对于其他 40% 时间，加速或减速是恒定的。



速度模式效果

这个表格总结了各种模式之间的差异：

模式 类型	加速 / 减速 时间	电机 压力	控制的优先级 最高到最低			
			加速 / 减速	速度	位置	
梯形	最快	最差	加速 / 减速	速度	位置	
S 曲线	2X 稍慢	最佳	跃度	加速 / 减速	速度	位置

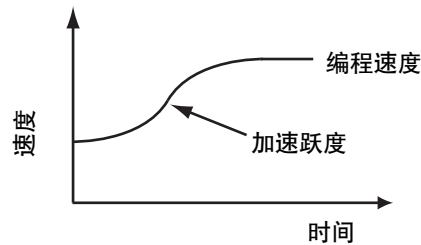
跃度值计算

如果指令使用或改变 S 曲线模式，控制器在您启动指令时计算加速、减速和跃度。

MAJ 的跃度参数以 % 时间为单位进行编程，按以下方式换算成工程单位：

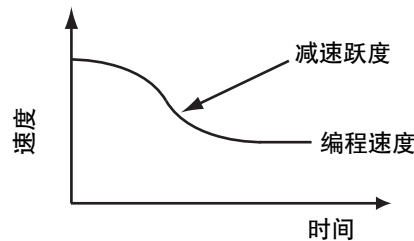
如果开始速度 < MAJ 编程速度

$$\text{加速度跃度 (单位 / 秒}^3\text{)} = \frac{\text{编程加速度}^2}{\text{编程速度}} * \left(\frac{200}{\% \text{ 时间}} - 1 \right)$$



如果开始速度 > MAJ 编程速度

$$\text{减速度跃度 (单位 / 秒}^3\text{)} = \frac{\text{编程减速速度}^2}{\text{Max(编程速度, [开始速度 - 编程速度])}} * \left(\frac{200}{\% \text{ 时间}} - 1 \right)$$

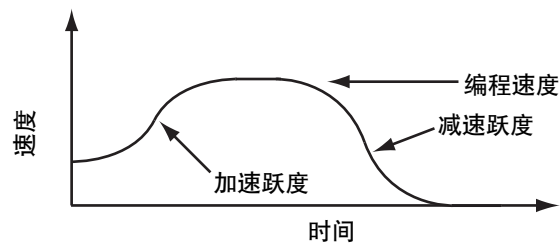


采用 % 时间为单位的编程位移的跃度，例如 MAM 或 MCLM 指令，按照以下方式换算成工程单位：

如果开始速度 < 编程速度

$$\text{加速度跃度 (单位 / 秒}^3\text{)} = \frac{\text{编程加速度}^2}{\text{编程速度}} * \left(\frac{200}{\% \text{ 时间}} - 1 \right)$$

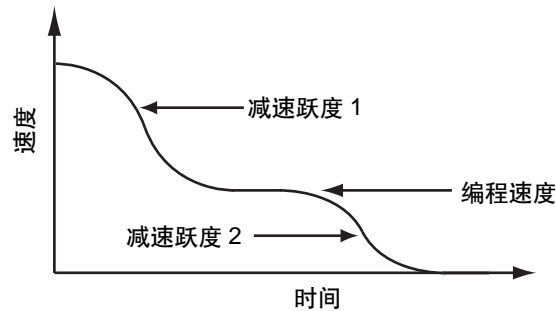
$$\text{减速度跃度 (单位 / 秒}^3\text{)} = \frac{\text{编程减速度}^2}{\text{Max(编程速度, [开始速度 - 编程速度])}} * \left(\frac{200}{\% \text{ 时间}} - 1 \right)$$



如果开始速度 > 编程速度

$$\text{减速度跃度 1} = \frac{\text{编程减速度}^2}{\text{Max(编程速度, [开始速度 - 编程速度])}} * \left(\frac{200}{\% \text{ 时间}} - 1 \right)$$

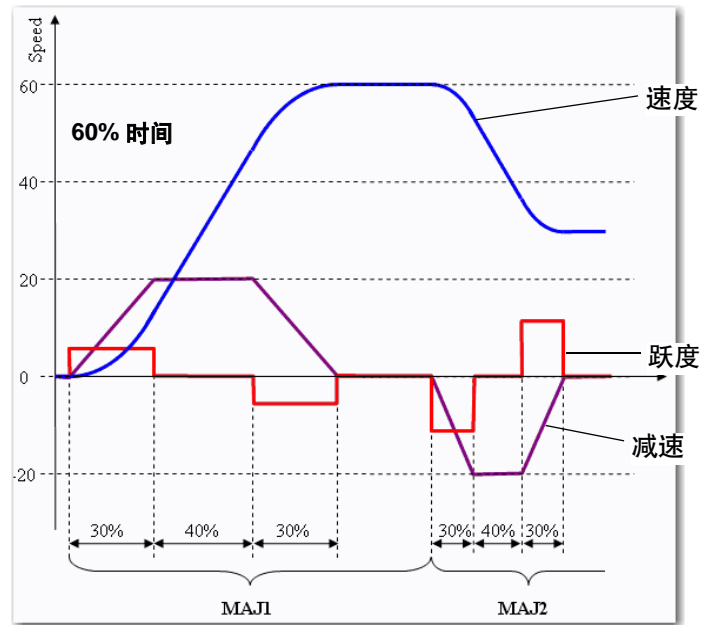
$$\text{减速度跃度 2} = \frac{\text{编程减速度}^2}{\text{编程速度}} * \left(\frac{200}{\% \text{ 时间}} - 1 \right)$$



当前速度 > 编程速度时，使用减速度跃度 1

当前速度 < 编程速度时，使用减速度跃度 2

根据指令的速度参数的不同，相同的“% 时间”跃度对加速模式和
对减速模式可能产生不同的斜率。



运动轨迹规划算法调整实际的跃度，使加速模式和减速模式包含最小的“% 时间”斜坡时间。如果开始速度接近编程速度参数，则实际的斜坡时间百分比可能会高于编程值。

在大多数情况下，条件如下：

如果：(开始速度 == 0.0) OR (开始速度为 > 2 * 最大速度)

那么：您取编程的斜坡时间百分比

否则：您取高于编程的斜坡时间百分比

从 % 时间换算成工程单位

如果您要将 % 时间换算成工程单位，请使用以下公式。

对于加速跃度：

$$j_a [\% \text{ 时间}] = \frac{2}{1 + \frac{j_a [\text{EU/s}^3] v_{\max}}{a_{\max}}} 100$$

对于减速跃度：

$$j_d [\% \text{ 时间}] = \frac{2}{1 + \frac{j_d [\text{EU/s}^3] v_{\max}}{d_{\max}}} 100$$

跃度编程单位是单位 / 秒³

如果要采用“单位 / 秒³”代替“% 时间”来指定跃度，请按照以下方式调整跃度值，使您能得到编程的值：

$$\text{临时速度} = \frac{\text{编程减速速度}^2}{\text{期望的跃度值, 单位是单位 / 秒}^3}$$

$$k = \frac{\text{开始速度} - \text{编程速度}}{\text{Max}(\text{编程速度}, \text{临时速度})}$$

如果 ($k < 1$)

- 指令面板减速跃度（采用单位 / 秒³）= 期望的减速跃度（采用单位 / 秒³）
否则
- 指令面板减速跃度（采用单位 / 秒³）= 期望的减速跃度（采用单位 / 秒³）* k

示例

示例 #1

开始速度 = 8.0 英寸 / 秒

期望速度 = 5.0 英寸 / 秒

期望减速度 = 2.0 英寸 / 秒²

期望减速跃度 = 1.0 英寸 / 秒³

$$\begin{aligned} \text{临时速度} &= (\text{期望减速度})^2 / \text{期望跃度值 (采用单位 / 秒}^3) \\ &= 2.0^2 / 1.0 = \\ &= 4.0 \text{ 英寸 / 秒} \end{aligned}$$

$$\begin{aligned} k &= (8.0 - 5.0) / \max(5.0, 4.0) = 3.0 / 5.0 = \\ &= 0.6 \end{aligned}$$

由于 $k < 1$ ，我们可以将期望的减速跃度直接输入面板

指令面板减速跃度（采用单位 / 秒³）= 1.0 英寸 / 秒³

示例

示例 #2

开始速度 = 13.0 英寸 / 秒

期望速度 = 5.0 英寸 / 秒

期望减速度 = 2.0 英寸 / 秒²

期望减速跃度 = 1.0 英寸 / 秒³

临时速度 = (期望减速度)² / 期望跃度值 (采用单位 / 秒³)
= 2.0² / 1.0 =
= 4.0 英寸 / 秒

$k = (13.0 - 5.0) / \max(5.0, 4.0) = 8.0 / 5.0 =$
 $= 1.6$

由于 $k > 1$ ，我们必须按以下方式计算减速跃度才能用在指令面板上：

指令面板减速跃度 (采用单位 / 秒³) =
= 1.0 英寸 / 秒³ * 1.6 =
= 1.6 英寸 / 秒³

您进行哪些改动？

- 15 或更早 - % 时间固定为 100。
- 16 或更晚 - % 时间默认为从较早版本上换算出来的项目上的 100% 时间。对于新的项目，您必须输入跃度值。

选择命令

使用这个表格来选择一个指令，查看其是否可用作运动直接命令：

如果您需要	而且	使用这个指令	运动直接命令
更改轴的状态	使能伺服驱动，并激活轴伺服环。	MSO 开运动伺服使能	是
	禁止伺服驱动，并去激活轴伺服环。	MSF 关运动伺服使能	是
	强制轴进入关闭状态，并阻止任何发起轴运动的指令。	MASD 关闭运动轴	是
	将轴过渡到就绪状态。如果伺服模块的所有轴都因为这条指令而离开关闭模式，则模块的 OK 继电器触点关闭。	MASR 运动轴关闭复位	是
	使能伺服驱动，并设置某个轴的伺服输出电压。	MDO 运动直接使能	是
	禁止伺服驱动，并将伺服输出电压设置为输出偏移电压。	MDF 直接关运动使能	是
	清除某个轴的所有运动故障。	MAFR 运动轴故障复位	是
控制轴的位置	停止某个轴上的任何运动过程。	MAS 运动轴停止	是
	使轴归零。	MAH 运动轴寻参	是
	使轴点动。	MAJ 轴速度	是
	将轴移到特定位置。	MAM 轴位移	是
	启动两个轴之间的电子齿轮	MAG 电子齿轮	是
	更改正在进行的位移或点动的速度、加速或减速。	MCD 运动动态调整	是
	更改轴的命令位置或者实际位置。	MRP 运动位置重设	是
	基于一系列凸轮点来计算凸轮曲线。	MCCP 计算凸轮运动曲线	否
	启动两个轴之间的电子凸轮	MAPC 运动轴位置凸轮	否
	启动电子凸轮作为时间的函数。	MATC 运动轴时间凸轮	否
计算从轴值、斜率和斜率导数以获得凸轮曲线和主轴值。	MCSV 计算从轴运动值	否	

如果您需要	而且	使用这个指令	运动直接命令
发起在所有轴上的操作。	停止所有轴的运动。	MGS 运动组停止	是
	强制所有轴进入关闭状态。	MGSD 关闭运动组	是
	将所有轴过渡到就绪状态。	MGSR 运动组关闭复位	是
	锁存当前命令和所有轴的实际位置。	MGSP 抓拍运动组位置	是
准备和解除特殊的事件检查功能，例如记录和观察位置	准备某个轴的观察位置事件检查。	MAW 开启运动位置监视	是
	解除某个轴的观察位置事件检查。	MDW 关闭运动位置监视	是
	准备某个轴的伺服模块记录事件检查。	MAR 开启轴位置抓拍	是
	解除某个轴的伺服模块记录事件检查。	MDR 关闭轴位置抓拍	是
	为某个轴和输出准备某个凸轮输出。	MAOC 开启运动凸轮输出	否
	解除连接到某个轴的一个或全部凸轮输出。	MDOC 关闭运动凸轮输出	否
调整某个轴并为您的控制系统运行诊断测试。这些测试包括： <ul style="list-style-type: none"> • 电机 / 编码器连接测试 • 编码器连接测试 • 标志测试 	使用 MAAT 指令的结果来计算和更新某个轴的私服增益和动态门限。	MAAT 应用轴整定参数	否
	运行某个轴的整定运动模式	MRAT 运行轴整定	否
	使用 MRHD 指令的结果来设置编码器和伺服极性。	MAHD 应用轴连接诊断	否
	在某个轴上运行诊断测试之一。	MRHD 运行轴连接诊断	否

如果您需要	而且	使用这个指令	运动直接命令
控制多轴协调运动	为协调系统的轴启动一个线性协调位移。	MCLM 运动直线插补	否
	为协调系统的轴启动一个圆形位移。	MCCM 运动圆弧插补	否
	更改坐标系上的有效运动的路径动态数据。	MCCD 运动插补动态调整	否
	停止坐标系的轴，或者取消转换。	MCS 运动插补停止	否
	关闭坐标系的轴。	MCSD 运动插补关闭	否
	启动一个将两个坐标系连起来的转换。这类似于双向齿轮。	MCT 运动插补坐标转换 ⁽¹⁾	否
	计算一个坐标系相对于另一个坐标系的位置。	MCTP 计算坐标转换位置 ⁽¹⁾	否
	将坐标系的轴过渡到就绪状态，并清除轴的故障。	MCSR 运动插补关闭复位	否

⁽¹⁾ 您只能在 1756-L6x 控制器上使用这条指令。

注意：

运动控制状态指令

(MSO、MSF、MASD、MASR、MDO、MDF、MAFR)

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

简介

运动控制状态指令直接控制或更改轴的工作状态。运动控制状态指令为：

如果您需要	使用这个指令	在以下语言中可用
使能伺服驱动，并激活轴伺服环。	MSO	梯形图 结构文本
禁止伺服驱动，并去激活轴伺服环。	MSF	梯形图 结构文本
强制所有轴进入关闭工作状态。轴一旦进入关闭工作状态，控制器将阻止任何发起轴运动的指令。	MASD	梯形图 结构文本
将轴从当前的关闭操作状态更改到轴就绪操作状态。如果伺服模块的所有轴都因为这条指令而离开关闭模式，则模块的 OK 继电器触点关闭。	MASR	梯形图 结构文本
使能伺服驱动，并设置某个轴的伺服输出电压。	MDO	梯形图 结构文本
去激活伺服驱动，并将伺服输出电压设置为输出偏移电压。	MDF	梯形图 结构文本
清除某个轴的所有运动故障。	MAFR	梯形图 结构文本

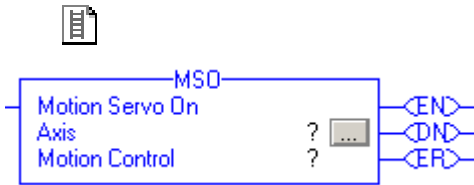
轴的五个工作状态为：

工作状态	描述
轴就绪	<p>这是轴的正常上电状态。在这种状态下：</p> <ul style="list-style-type: none"> • 伺服模块变频器使能输出不活跃。 • 伺服操作被禁止。 • 不存在伺服故障。
直接变频器控制	<p>这种操作状态允许伺服模块 DAC 直接控制外部变频器。在这种状态下：</p> <ul style="list-style-type: none"> • 伺服模块变频器使能输出活跃。 • 位置伺服操作被禁止。
伺服控制	<p>这种操作状态允许伺服模块执行关闭的循环运动。在这种状态下：</p> <ul style="list-style-type: none"> • 伺服模块变频器使能输出活跃。 • 伺服操作被使能。 • 轴被强制保持命令的伺服位置。
轴故障	<p>在这种工作状态下，存在某个伺服故障，变频器使能输出的状态、伺服的操作以及 OK 触点的条件取决于所存在的故障以及故障动作。</p>
关闭	<p>这种工作状态允许 OK 继电器触点打开变频器电源急停环内的一系列触点。在这种状态下：</p> <ul style="list-style-type: none"> • 伺服模块变频器使能输出不活跃。 • 伺服操作被禁止。 • OK 触点打开。

开运动伺服使能 (MSO)

使用 MSO 指令可激活指定轴的变频器放大器以及激活轴的伺服环。

操作数： 梯形图



操作数	类型	格式	描述
轴	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



MSO(Axis,MotionControl);

结构文本

操作数与梯形图 MSO 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴的伺服操作已经成功被使能而且伺服活跃状态位被设置时, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。

说明： 开运动伺服使能 (MSO) 指令直接激活变频器, 并使能已配置的与物理伺服轴关联的伺服环。它可以用在程序中的任何位置, 但轴正在运动时不应使用。如果尝试使用, 则 MSO 指令生成“轴在运动中”错误。

MSO 指令通过激活变频器和激活关联的伺服环, 自动使能指定的轴。轴的结果状态是指伺服控制状态。

这条指令的最常用的地方是在预备命令运动时, 对指定的轴在其当前位置激活伺服环。

要成功执行 MSO 指令，目标轴必须配置为伺服轴。如果不满足这个条件，则指令出错。

重要提示

MSO 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息，并需要时间让变频器输出稳定下来以及激活伺服环。完成 (.DN) 位并不立即设置，而只是在轴处于伺服控制状态以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MSO 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

扩展错误代码 (十进制)	相关联的错误代码 (十进制)	含义
对象模式冲突 (12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态。
权限被拒绝 (15)	SERVO_MESSAGE_FAILURE (12)	使能输入开关错误。(SERCOS)
设备处于错误状态 (16)	SERVO_MESSAGE_FAILURE (12)	设备状态不能用于操作。(SERCOS)

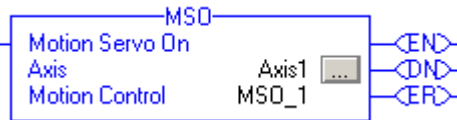
MSO 对状态位的更改 *轴状态位*

位名称	状态	含义
ServoActStatus	真	轴处于伺服控制状态，伺服环活跃。
DriveEnableStatus	真	轴变频器使能输出活跃。

运动状态位

无

示例： 当输入条件为真时，控制器使能伺服变频器并激活轴 1 所配置的轴伺服环。

梯形图**MSO 梯形图示例***结构文本*

```
MSO(Axis0,MSO_1);
```

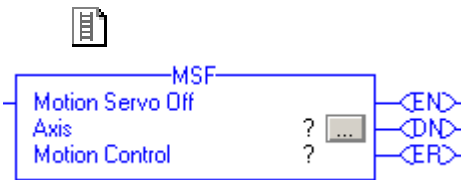
关运动伺服使能 (MSF)

使用 MSF 指令可去激活指定轴的变频器输出以及去激活轴的伺服环。

重要提示

如果当轴正在运动时执行 MSF 指令，轴将惯性运转至非控制停止。

操作数: *梯形图*



MSF(Axis,MotionControl);

操作数	类型	格式	描述
轴	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MSF 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴的伺服操作已经成功被禁止而且变频器使能和伺服活跃状态位都被设置时, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。

说明:

关运动伺服使能 (MSF) 指令直接并立即关闭变频器输出, 并禁止任何物理伺服轴上的伺服环。这可将轴置于轴就绪状态。MSF 指令还会禁止可能会在执行时被激活的任何运动计划器。MSF 指令不需要参数 – 只需简单地输入或选择所需的轴即可。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

当您手动移动轴时, 可使用 MSF 指令来将伺服操作关闭。原因是即使伺服操作关闭时, 位置继续被跟踪。当伺服环被 MSO 指令再次打开时, 轴在新位置将再次处于闭环控制之下。

轴的停止行为对于不同的变频器类型会各不相同。在某些情况下轴惯性运转至停止，而在另一些情况下轴会使用变频器提供的停止转矩减速停止。

要成功执行 MSF 指令，目标轴必须配置为伺服轴。如果不满足这个条件，则指令出错。如果您的轴类型为虚拟，则指令出错，因为对于虚轴，伺服操作和变频器使能状态被始终强制为真。已使用轴数据类型也会出错，因为生产控制器会改变已使用轴的状态。

重要提示

MSF 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息，并需要时间让变频器输出和伺服环被完全去激活。完成 (.DN) 位直到这条消息被成功发送、轴过渡到轴就绪状态后才会设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

MSF 对状态位的更改： *轴状态位*

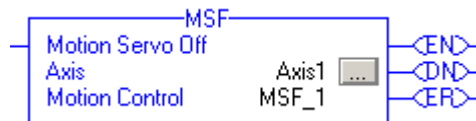
位名称	状态	含义
ServoActionStatus	假	轴处于伺服开启状态，伺服环活跃。
DecelStatus	假	轴变频器使能输出活跃。

运动状态位

位名称	状态	含义
AccelStatus	假	轴没有在加速。
DecelStatus	假	轴没有在减速。
MoveStatus	假	轴没有在移动。
JogStatus	假	轴没有在点动。
GearingStatus	假	轴没有在齿轮运动中。
HomingStatus	假	轴没有在归零。
StoppingStatus	假	轴没有在停止。
PositionCamStatus	假	轴没有在位置凸轮运动中。
TimeCamStatus	假	轴没有在时间凸轮运动中。
PositionCamPendingStatus	假	轴没有挂起的位置凸轮。
TimeCamPendingStatus	假	轴没有挂起的时间凸轮。
GearingLockStatus	假	轴没有处于齿轮锁定状态。
PositionCamLockStatus	假	轴没有处于凸轮锁定状态。

示例： 当输入条件为真时，控制器禁止伺服变频器以及轴 1 所配置的轴伺服环。

梯形图



MSF 梯形图示例

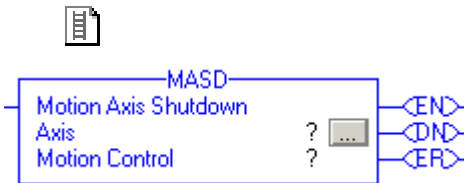
结构文本

```
MSF(Axis0,MSF_1);
```

关闭运动轴 (MASD)

使用 MASD 指令可强制指定的所有轴进入关闭状态。轴的关闭状态包括这些条件：变频器输出被禁止、伺服环被去激活、任何可用或关联的 OK 固态继电器触点打开。轴将保持关闭状态，直到执行轴或组关闭复位为止。

操作数：梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



MASD(Axis,MotionControl);

结构文本

操作数与梯形图 MASD 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴被成功置于关闭状态时，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。

说明： 关闭运动轴 (MASD) 指令直接并立即禁止变频器输出，禁止伺服环，并打开任何关联的 OK 触点。此操作将轴置于关闭状态。

MASD 指令发起的另一个操作是清除所有正在进行的运动过程并清除所有运动状态位。与此操作相关，命令还会清除当前为目标轴设置的所有运动控制指令的 IP 位。

MASD 指令强制目标轴进入关闭状态。关闭状态的独特特点之一就是，如果可用，则运动控制模块或变频器的 OK 固态继电器触点打开。这种功能可用于打开控制变频器系统主电源的急停环。注意，通常每个运动控制模块只有一个 OK 触点，这意味着对于给定的模块，在任一关联的轴上执行 MASD 指令都将打开 OK 触点。

关机状态的另一个特点是，任何发起运动的指令都被阻止执行。尝试这样做会导致执行错误。只有通过执行其中一个关闭复位指令，才能成功地发起运动。

要成功执行 MASD 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

轴将保持关闭状态，直到运动轴关闭复位 (MASR) 或运动组关闭复位 (MGSR) 指令执行为止。

重要提示

MASD 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。因此，完成 (.DN) 位直到这条消息被成功发送、轴过渡到关闭状态后才会设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

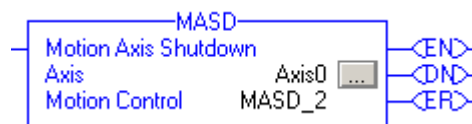
MASD 对状态位的更改： *轴状态位*

位名称	状态	含义
ServoActStatus	假	<ul style="list-style-type: none"> 轴处于轴就绪状态。 伺服环不活跃。
DriveEnableStatus	假	变频器使能输出不活跃。
ShutdownStatus	真	轴处于关闭状态。

运动状态位

位名称	状态	含义
AccelStatus	假	轴没有在加速。
DecelStatus	假	轴没有在减速。
MoveStatus	假	轴没有在移动。
JogStatus	假	轴没有在点动。
GearingStatus	假	轴没有在齿轮运动中。
HomingStatus	假	轴没有在归零。
StoppingStatus	假	轴没有在停止。
PositionCamStatus	假	轴没有在位置凸轮运动中。
TimeCamStatus	假	轴没有在时间凸轮运动中。
PositionCamPendingStatus	假	轴没有挂起的位置凸轮。
TimeCamPendingStatus	假	轴没有挂起的时间凸轮。
GearingLockStatus	假	轴没有处于齿轮锁定状态。
PositionCamLockStatus	假	轴没有处于凸轮锁定状态。

示例： 当输入条件为真时，控制器强制轴 1 进入关闭工作状态。

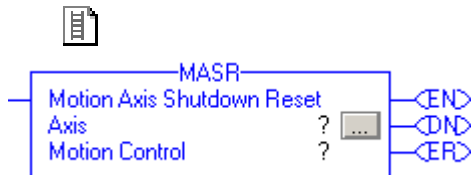
梯形图**MASD 梯形图示例***结构文本*

```
MASD(Axis0,MASD_1);
```

运动轴关闭复位 (MASR)

使用 MASR 指令可将轴从当前的关闭操作状态过渡到轴就绪操作状态。所有与指定的轴关联的故障都被自动清除。如果因这条指令的执行而使相关运动控制模块的所有轴都不再处于关闭状态，则模块的 OK 继电器触点关闭。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



结构文本

MASR(Axis,MotionControl);

操作数与梯形图 MASR 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴被成功从关闭状态中复位时，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。

说明: 运动轴关闭复位 (MASR) 指令清除所有的轴故障，并将指定的轴带离关闭状态。如果运动控制模块支持 OK 触点，而且没有其他模块轴处于关闭状态，MASR 指令将导致该模块的 OK 固态继电器触点关闭。不管 OK 触点的条件如何，执行 MASR 会将轴置于轴就绪状态。

就像 MASD 指令强制目标轴进入关闭状态一样，MASR 指令将轴带离关闭状态，进入轴就绪状态。关闭状态的独特特点之一就是，该运动控制模块任何关联的 OK 固态继电器触点打开。如果因执行 MASR 指令而使给定的运动控制模块中关联的轴都离开了关闭状态，则 OK 继电器触点会在 MASR 执行后关闭。这种功能可用于关闭控制变频器系统主电源的急停环，因此允许客户向变频器重新施加电源。注意，通常每个运动控制模块只有一个 OK 触点，这意味着可能需要对给定模块的所有相关轴都要执行 MASR 指令才会使 OK 触点关闭。

要成功执行 MASR 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示

MASR 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。因此，完成 (.DN) 位并不立即设置，而只是在消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

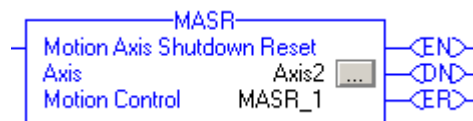
错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

状态位：

位名称	状态	含义
ShutdownStatus	假	轴不是处于关闭状态。

示例： 当输入条件为真时，控制器将 *轴 1* 从关闭工作状态过渡到轴就绪工作状态。

梯形图



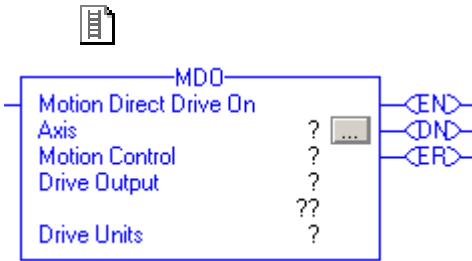
结构文本

```
MASR(Axis0,MASR_1);
```

运动直接使能 (MDO)

MDO 指令要配合使用支持外部模拟量伺服变频器接口（例如 1756-M02AE 或 1784-PM02AE 伺服模块）的运动控制模块。这条指令去激活模块的变频器使能，使能外部伺服变频器，同时将伺服模块的变频器输出电压设置为指定的电压。变频器输出值可采用伏特或者最大轴输出限制的 %。

操作数: 梯形图



MDO(Axis,MotionControl, DriveOutput,DriveUnits);

操作数	数据类型	描述
轴	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_INSTRUCTION 标签	用于访问指令状态参数的结构。
变频器输出	REAL	要输出的电压，采用伺服输出限制的 % 或者伏特
变频器单位	布尔型	变频器输出值被解释时所采用的单位。

结构文本

操作数与梯形图 MDO 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
DriveUnits	volts	0
	百分比	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴的变频器启用位被激活，而且指定的模拟量输出被成功应用时，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您输入的变频器输出值太大。

说明： 对于具有外部伺服变频器接口的运动控制模块，例如 1756-M02AE 或 1784-PM02AE，运动直接使能 (MDO) 指令可用来直接使能轴的变频器使能输出，并将模拟量输出设置为由变频器输出参数确定的指定电平。变频器输出参数可表示为电压，或者为输出限制属性所提供的最大配置输出电压值的百分比。

MDO 指令只能用在轴类型被配置为伺服的物理轴上。只有当轴不处于轴就绪状态时（例如伺服操作被关闭），这条指令才能执行。轴的结果状态是指变频器控制状态。

MDO 指令在将伺服模块的模拟量输出设置为指定的电压值之前先激活相应的变频器使能输出，从而自动使能指定的轴。通常，在变频器使能输出的激活与将模拟量输出设置为指定电平之间有 500 毫秒的延迟，让变频器的电源结构能够稳定下来。为了将这个变频器使能延迟期间的漂移降至最低，到变频器的输出电压被设置为输出偏移属性值（默认值为零）。其后，输出电压由 MDO 指令的指定变频器输出值来提供，并通过伺服输出状态属性值来指示。

与各种 Logix 伺服模块关联的 16 位 DAC 硬件将直接变频器运动控制的有效分辨率限制为 305 V 或 0.003%。在直接变频器操作的情况下，模块的伺服环不活跃并且被忽略。运动直接使能指令仅受到该轴的伺服输出极性配置位、输出偏移和输出限制属性的影响。如果输出限制配置值降至当前输出电压值以下，则伺服输出值自动箝位到输出限制值。

这条指令的最常用的地方是提供独立的可编程模拟输出，作为外部变频器的开环速度基准值。或者用于测试外部伺服变频器的闭环操作。

要成功执行 MDO 指令，目标轴必须配置为伺服轴并且处于轴就绪状态，而且伺服动作关闭。如果不满足这些条件，则指令出错。

重要提示

MDO 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息，并需要时间让变频器输出稳定下来。完成 (.DN) 位要到在轴处于变频器控制状态以后才被设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MDO 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

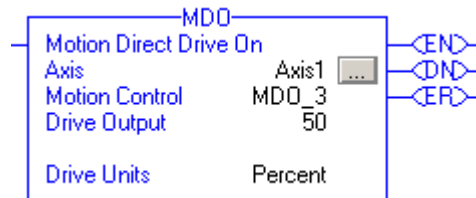
扩展错误代码 (十进制)	相关联的错误代码 (十进制)	含义
对象模式冲突 (12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态。

状态位： *MDO 对状态位的更改*

位名称	状态	含义
DriveEnableStatus	真	轴处于变频器控制状态，变频器使能输出激活。

示例： 当输入条件为真时，控制器激活轴 1 的伺服变频器，并设置轴 1 的伺服输出电压。在这个例子中，输出等于输出值的 2%。

梯形图



MDO 梯形图示例

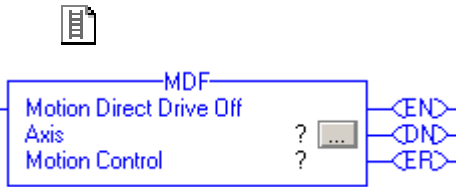
结构文本

```
MDO(Axis0,MDO_1,4,percent);
```

直接关运动使能 (MDF)

使用 MDF 指令可去激活伺服驱动，并将伺服输出电压设置为输出偏移电压。该输出偏移电压是产生零或最小变频器运动的输出电压。您可以在轴配置过程中指定这个值。

操作数: 梯形图



MDF(Axis,MotionControl);

操作数	数据类型	描述
轴	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_ INSTRUCTION 标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MDF 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴的变频器信号被成功禁止，而且变频器使能状态位被清除后，完成后被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。

说明: 对于具有外部伺服变频器接口的运动控制模块，例如 1756 第 02AE，直接关运动使能 (MDF) 指令直接禁止指定物理轴的运动控制模块变频器使能输出，同时还通过应用所配置的输出偏移值，将模块的伺服输出“归零”到外部变频器。

MDF 指令用于停止由前一条 MDO 指令发起的运动，将轴从直接变频器控制状态过渡回到轴就绪状态。

要成功执行 MDF 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示

MDF 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

MDF 对状态位的更改： *轴状态位*

位名称	状态	含义
DriveEnableStatus	假	轴正处于轴就绪状态，而且变频器使能输出现在活跃。

示例： 当输入条件为真时，控制器去激活 *轴 1* 的伺服变频器，并将 *轴_* 的伺服输出电压设置为输出偏移值。

梯形图



MDF 梯形图示例

结构文本

```
MDF(Axis0,MDF_1);
```

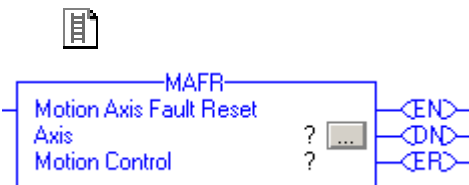
运动轴故障复位 (MAFR)

使用 MAFR 指令可清除某个轴的所有运动故障。这是清除轴运动故障的唯一方法。

重要提示

MAFR 指令删除故障状态，但并不执行任何恢复工作，例如使能伺服操作。此外，控制器删除故障状态后，产生该故障的条件可能仍然存在。如果在使用 MAFR 指令之前没有纠正条件，轴会立即再次出现故障。

操作数：梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



MAFR(Axis,MotionControl);

结构文本

操作数与梯形图 MAFR 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴的故障被成功清除时，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。

说明： 运动轴故障复位 (MAFR) 指令直接清除指定轴上的指定故障状态。它不纠正导致该错误的条件。如果在执行 MAFR 指令之前没有纠正条件，轴会立即再次出现故障，表现为故障状态并没有复位。

这条指令最常用于作为故障处理器程序的一部分，该程序提供特定于应用的故障操作来响应各种潜在的运动控制故障。一旦采取相应的故障操作后，MAFR 指令可用于清除所有活跃的故障状态位。

要成功执行 MAFR 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示

MAFR 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在消息被成功发送以后。不能保证此指令可以清除所有故障，因为一个或多个故障可能是某种持续条件的结果。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

MAFR 对状态位的更改： 无

MAFR 示例： 当输入条件为真时，控制器清除轴 1 的所有运动故障。

梯形图



MAFR 梯形图示例

结构文本

```
MAFR(Axis0,MAFR_1);
```

运动位移指令

(MAS、MAH、MAJ、MAM、MAG、MCD、MRP、MCCP、MAPC、MATC、MCSV)

简介

使用运动位移指令来控制轴的位置。

如果您需要	使用这个指令	在以下语言中可用
停止某个轴上的任何运动过程	运动轴停止 (MAS)	梯形图 结构文本
使轴归零	运动轴寻参 (MAH)	梯形图 结构文本
使轴点动	轴速度 (MAJ)	梯形图 结构文本
将轴移到特定位置	轴位移 (MAM)	梯形图 结构文本
启动两个轴之间的电子齿轮	电子齿轮 (MAG)	梯形图 结构文本
更改正在进行的移动或点动的速度、加速或减速	运动动态调整 (MCD)	梯形图 结构文本
更改轴的命令位置或者实际位置	运动位置重设 (MRP)	梯形图 结构文本
基于一系列凸轮点来计算凸轮曲线	计算凸轮运动曲线 (MCCP)	梯形图 结构文本
启动两个轴之间的电子凸轮	运动轴位置凸轮 (MAPC)	梯形图 结构文本
启动电子凸轮作为时间的函数	运动轴时间凸轮 (MATC)	梯形图 结构文本
计算从轴值、斜率和斜率导数以获得凸轮曲线和主轴值	计算从轴运动值 (MCSV)	梯形图 结构文本

运动轴停止 (MAS)

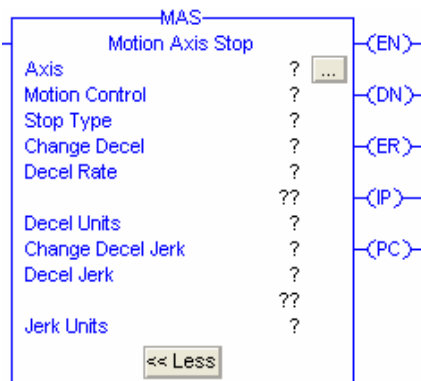
使用 MAS 指令来停止某个轴上的特定运动过程或者完全停止该轴。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数 梯形图



操作数	类型	格式	描述						
轴	AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	轴的名称						
运动控制	MOTION_INSTRUCTION	标签	指令的控制标签						
停止类型	DINT	立即	<table border="1"> <thead> <tr> <th>要停止</th> <th>选择这种停止类型</th> </tr> </thead> <tbody> <tr> <td>这个轴所有正在进行的运动</td> <td>全部 (0)</td> </tr> <tr> <td>仅某种类型的运动，而使其他运动过程继续运行</td> <td> 选择您要停止的运动类型： <ul style="list-style-type: none"> • 点动 (1) • 移动 (2) • 齿轮 (3) • 归零 (4) • 整定 (5) • 测试 (6) • 位置凸轮 (7) • 时间凸轮 (8) • 主偏移移动 (9) 当 MAS 指令完成后，轴仍然会在移动。 </td> </tr> </tbody> </table>	要停止	选择这种停止类型	这个轴所有正在进行的运动	全部 (0)	仅某种类型的运动，而使其他运动过程继续运行	选择您要停止的运动类型： <ul style="list-style-type: none"> • 点动 (1) • 移动 (2) • 齿轮 (3) • 归零 (4) • 整定 (5) • 测试 (6) • 位置凸轮 (7) • 时间凸轮 (8) • 主偏移移动 (9) 当 MAS 指令完成后，轴仍然会在移动。
要停止	选择这种停止类型								
这个轴所有正在进行的运动	全部 (0)								
仅某种类型的运动，而使其他运动过程继续运行	选择您要停止的运动类型： <ul style="list-style-type: none"> • 点动 (1) • 移动 (2) • 齿轮 (3) • 归零 (4) • 整定 (5) • 测试 (6) • 位置凸轮 (7) • 时间凸轮 (8) • 主偏移移动 (9) 当 MAS 指令完成后，轴仍然会在移动。								

操作数	类型	格式	描述						
变更减速度	DINT	立即	<table border="1"> <thead> <tr> <th>如果您需要</th> <th>那么选择</th> </tr> </thead> <tbody> <tr> <td>使用轴的最大减速度</td> <td>否 (0)</td> </tr> <tr> <td>指定减速度</td> <td>是 (1)</td> </tr> </tbody> </table>	如果您需要	那么选择	使用轴的最大减速度	否 (0)	指定减速度	是 (1)
如果您需要	那么选择								
使用轴的最大减速度	否 (0)								
指定减速度	是 (1)								
减速度	REAL	立即或标签	<p>重要事项: 如果运动正在进行之中而您降低减速度, 则轴可能会过冲其目标位置。</p> <p>轴的减速度。只有当变更减速为是时, 指令才会使用这个值。</p>						
减速度单位	DINT	立即	<p>您要为减速度使用什么单位?</p> <ul style="list-style-type: none"> • 单位每秒² (0) • 最大值 %(1) 						
变更减速度跃度	DINT	立即	<table border="1"> <thead> <tr> <th>如果您需要</th> <th>那么选择</th> </tr> </thead> <tbody> <tr> <td>使用轴的最大减速度跃度</td> <td>否 (0)</td> </tr> <tr> <td>编程减速度跃度速率</td> <td>是 (1)</td> </tr> </tbody> </table>	如果您需要	那么选择	使用轴的最大减速度跃度	否 (0)	编程减速度跃度速率	是 (1)
如果您需要	那么选择								
使用轴的最大减速度跃度	否 (0)								
编程减速度跃度速率	是 (1)								
减速跃度	REAL	立即或标签	<p>重要事项: 如果运动正在进行之中而您降低减速度跃度, 则轴可能会过冲其目标位置。</p> <p>您必须始终为减速度跃度操作数输入值。如果模式配置为 S 曲线, 这个指令仅使用该值。</p> <ul style="list-style-type: none"> • 减速跃度是轴的减速跃度值。 <p>使用这些值起步。</p> <p>减速度跃度 = 100 (时间 %)</p> <p>0 = 单位每秒³</p> <p>1 = 最大值 %</p> <p>2 = 时间 %(使用这个值起步)</p>						
跃度单位	DINT	立即							



MAS(Axis,MotionControl, StopType,ChangeDecel, DecelRate,DecelUnits, ChangeDecelJerk, DecelJerk,JerkUnits);

结构文本

结构文本操作数与梯形图操作数相同。

此操作数	拥有这些选项可让您	
	输入为文本	或者输入为
停止类型	all jog move gear home tune test timecam positioncam masteroffsetmove	0 1 2 3 4 5 6 7 8 9
变更减速度	无 是	0 1
减速度单位	单位每秒 ² %ofmaximum	0 1
变更减速度跃度	无 是	0 1
减速度跃度	无枚举	您 必须 始终为减速度跃度操作数输入值。如果模式配置为 S 曲线, 这个指令仅使用该值。 • 减速度跃度是轴的减速度值。 使用这个值起步。 减速度跃度 = 时间 100% (2)
跃度单位	unitspersec3 最大值 % % 时间	0 1 2

MOTION_INSTRUCTION

数据类型

要查看是否	检查这个位是否被设置	数据类型	注意:
从假到真的过渡将导致这个指令执行	EN	BOOL	EN 位保持设置状态, 直到过程完成、梯级变为假为止。
停止被成功发起	DN	BOOL	
出现错误	ER	BOOL	
轴正在停止	IP	BOOL	这些操作的任何一个都会结束 MAS 指令, 并清除 IP 位: <ul style="list-style-type: none"> • 轴被停止 • 另外一条 MAS 指令取代这条 MAS 指令 • 关闭命令 • 故障操作
轴被停止	PC	BOOL	PC 位保持设置状态, 直到梯级进行一次从假到真的过渡。

描述 当您想要对轴的任何正在进行的控制运动进行减速停止时, 即可使用运动轴停止 (MAS) 指令。该指令停止运动, 而不会禁止伺服环。对于 MAS 设置停止类型 = 全部时, 这种减速始终使用梯形模式, 不管编程的模式类型如何。使用该指令来:

- 停止特定的运动过程, 例如点动、移动或齿轮
- 完全停止该轴
- 中止由 MRHD 或 MRAT 指令发起的测试或整定过程

MAS 指令使用哪种类型的模式?

如果停止类型为	则 MAS 指令使用这种模式
点动	与启动该点动的轴速度 (MAJ) 指令相同的模式类型
移动	与启动该位移的轴位移 (MAM) 指令相同的模式类型
以上都不是	梯形

示例

假设您使用 MAJ 指令以及 S 曲线模式来开始一个点动。然后您用 MAS 指令以及停止类型为点动来停止该点动。在这种情况下, MAS 指令使用 S 曲线模式来停止点动。

编程原则

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或者在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。

有关更多信息，请参见 [Troubleshoot Axis Motion](#) 上的 [page 9](#)。

指导原则	详细说明						
<ul style="list-style-type: none"> • 在梯形图中，每当您希望执行指令时，请切换梯级条件。 	<p>这是一条转换指令：</p> <ul style="list-style-type: none"> • 在梯形图中，每当您希望执行指令时，将梯级输入条件从清除切换到设置即可。 						
<ul style="list-style-type: none"> • 在结构文本中，限制指令只对转换执行。 	<p>在结构文本中，指令每次被扫描时就会执行。</p> <ul style="list-style-type: none"> • 限制指令只对转换执行。使用下面的方法之一： <ul style="list-style-type: none"> • SFC 操作限定符 • 结构文本构建 <p>有关更多信息，请参阅附件 C。</p>						
<ul style="list-style-type: none"> • 选择是停止全部运动还是特定类型的运动。 	<table border="1"> <thead> <tr> <th>如果您想要停止</th> <th>那么选择这种停止类型</th> </tr> </thead> <tbody> <tr> <td>这个轴所有正在进行的运动</td> <td>全部 指令使用梯形模式，停止该轴。</td> </tr> <tr> <td>仅停止某种类型的运动，而使其他运动过程继续运行</td> <td>您要停止的运动类型 当 MAS 指令完成后，轴仍然会在移动。 指令使用 S 曲线模式来停止轴，仅当： <ul style="list-style-type: none"> • 停止类型为点动或位移，并且 • 点动或位移使用 S 曲线模式。 </td> </tr> </tbody> </table>	如果您想要停止	那么选择这种停止类型	这个轴所有正在进行的运动	全部 指令使用梯形模式，停止该轴。	仅停止某种类型的运动，而使其他运动过程继续运行	您要停止的运动类型 当 MAS 指令完成后，轴仍然会在移动。 指令使用 S 曲线模式来停止轴，仅当： <ul style="list-style-type: none"> • 停止类型为点动或位移，并且 • 点动或位移使用 S 曲线模式。
如果您想要停止	那么选择这种停止类型						
这个轴所有正在进行的运动	全部 指令使用梯形模式，停止该轴。						
仅停止某种类型的运动，而使其他运动过程继续运行	您要停止的运动类型 当 MAS 指令完成后，轴仍然会在移动。 指令使用 S 曲线模式来停止轴，仅当： <ul style="list-style-type: none"> • 停止类型为点动或位移，并且 • 点动或位移使用 S 曲线模式。 						
<p>示例：假设您的轴正在同时执行点动和位移。再假设您想要仅停止点动，但让位移继续运行。在这种情况下，选择停止类型为点动。</p>							

指导原则	详细说明
<ul style="list-style-type: none"> 要停止齿轮或凸轮，请选择从轴。 	要停止齿轮或位置凸轮过程，请输入从轴来关闭特定的过程并停止该轴。如果主轴是一个伺服轴，您可以停止主轴，从而停止从轴，而不会禁止齿轮或位置凸轮。
<ul style="list-style-type: none"> 要停止主轴偏移位移，请输入从轴，但使用主轴单位。 	要停止绝对值或增量主轴偏移位移： <ul style="list-style-type: none"> 对于轴，请输入从轴。 对于减速度和跃度，请输入用于主轴的值和单位。
<ul style="list-style-type: none"> 请小心注意指令是否更改运动参数。 	当您执行 MAS 指令时，轴使用新的减速度和跃度值用于已经正在进行中的运动。这可能导致轴的速度过冲、结束位置过冲或方向相反。S 曲线模式对参数变更更加敏感。 有关更多信息，请参见 Troubleshoot Axis Motion 上的 page 9 。
<ul style="list-style-type: none"> 使用跃度操作数用于 S 曲线模式。 	使用跃度操作数的条件 <ul style="list-style-type: none"> 停止类型为点动或位移。 点动或位移使用 S 曲线模式。 在这些条件下，指令使用 S 曲线模式来停止轴。指令使用为所有其他的停止类型使用恒定减速度。您必须填写跃度操作数，不管使用哪种停止类型。
<ul style="list-style-type: none"> 在最初的跃度编程和整定中使用时间 %。 	为了使跃度的编程和整定更加方便，请将其输入为加速或减速时间的 %。 有关更多信息，请参阅： <ul style="list-style-type: none"> 编写一个速度模式程序 在 页码 27 整定 S 曲线模式，在 页码 387 上。

算术状态标志 不受影响

故障条件 无

错误代码 参见 [运动控制指令错误代码 \(ERR\)](#) (在 [页码 391](#) 上)。

扩展错误代码 使用扩展错误代码 (EXERR) 可得到关于错误的更多信息。

如果 ERR 为	并且 EXERR 是	那么												
		原因 处理措施												
13	各种	<p>某个操作数超出其范围。</p> <p>EXERR 是超出范围的操作数的编号。第一个操作数是 0。</p> <p>例如，如果 EXERR = 4，那么就检查减速度。</p> <table border="1"> <thead> <tr> <th>EXERR</th> <th>MAS 操作数</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>轴</td> </tr> <tr> <td>1</td> <td>运动控制</td> </tr> <tr> <td>2</td> <td>停止类型</td> </tr> <tr> <td>3</td> <td>变更减速度</td> </tr> <tr> <td>4</td> <td>减速度</td> </tr> </tbody> </table>	EXERR	MAS 操作数	0	轴	1	运动控制	2	停止类型	3	变更减速度	4	减速度
EXERR	MAS 操作数													
0	轴													
1	运动控制													
2	停止类型													
3	变更减速度													
4	减速度													

对状态位的更改 运动状态位

如果停止类型为 那么

非全部 指令为您停止的运动过程清除运动状态位。

全部 指令清除所有运动状态位。

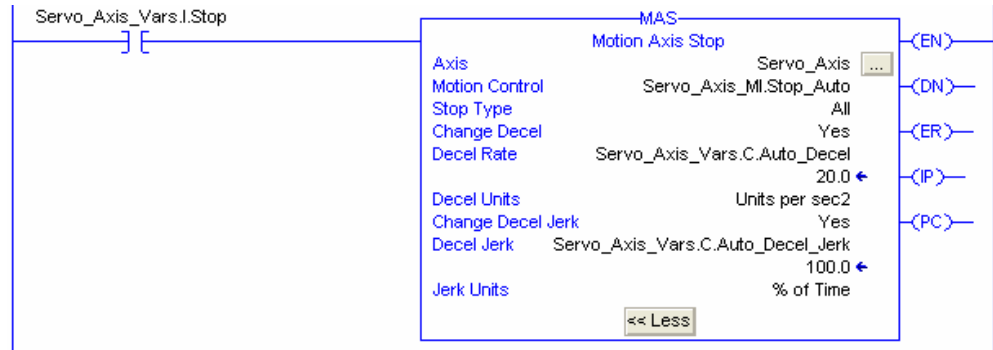
位	状态	含义
MoveStatus	假	轴没有在平移。
JogStatus	假	轴没有在点动。
GearingStatus	假	轴没有在齿轮运动中。
HomingStatus	假	轴没有在归零。
StoppingStatus	真	轴正在停止。
PositionCamStatus	假	轴没有在位置凸轮运动中。
TimeCamStatus	假	轴没有在时间凸轮运动中。
PositionCamPendingStatus	假	轴没有挂起的位置凸轮。
TimeCamPendingStatus	假	轴没有挂起的时间凸轮。
GearingLockStatus	假	轴没有处于齿轮锁定状态。
PositionCamLockStatus	假	轴没有处于凸轮锁定状态。

示例 1 当 *Servo_Axis_Vars.I.Stop* 打开时

停止 *Servo_Axis* 上的所有运动。

以 20.0 单位每秒 2 的减速度减速。

指令不使用减速度跃度值。由于停止类型为全部，指令使用梯形模式来停止轴。

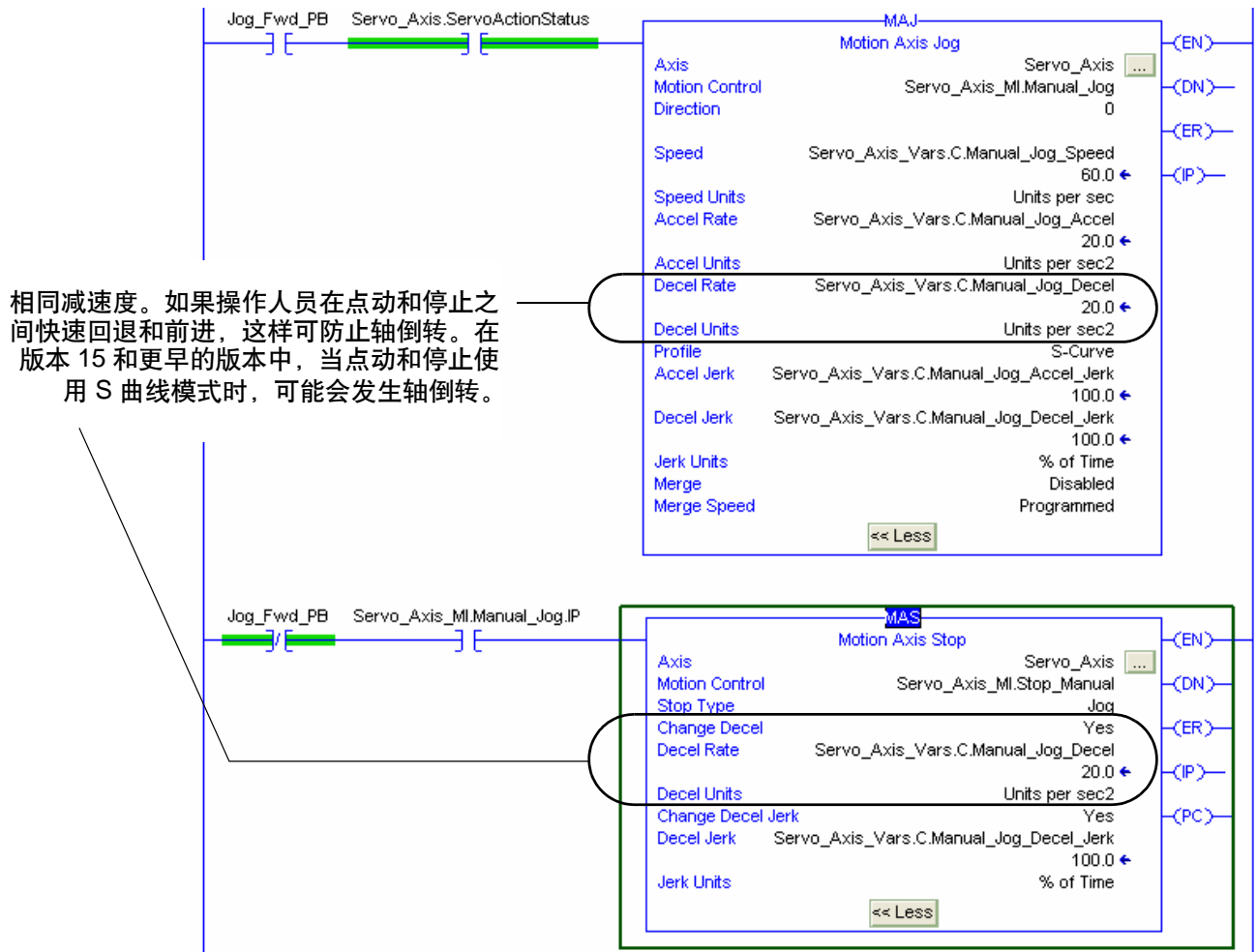
梯形图**结构文本**

示例 2 操作人员使用按钮来点动某个轴。该按钮打开和关闭 *Jog_Fwd_PB* 标签。当操作人员释放按钮时，MAS 指令停止轴。

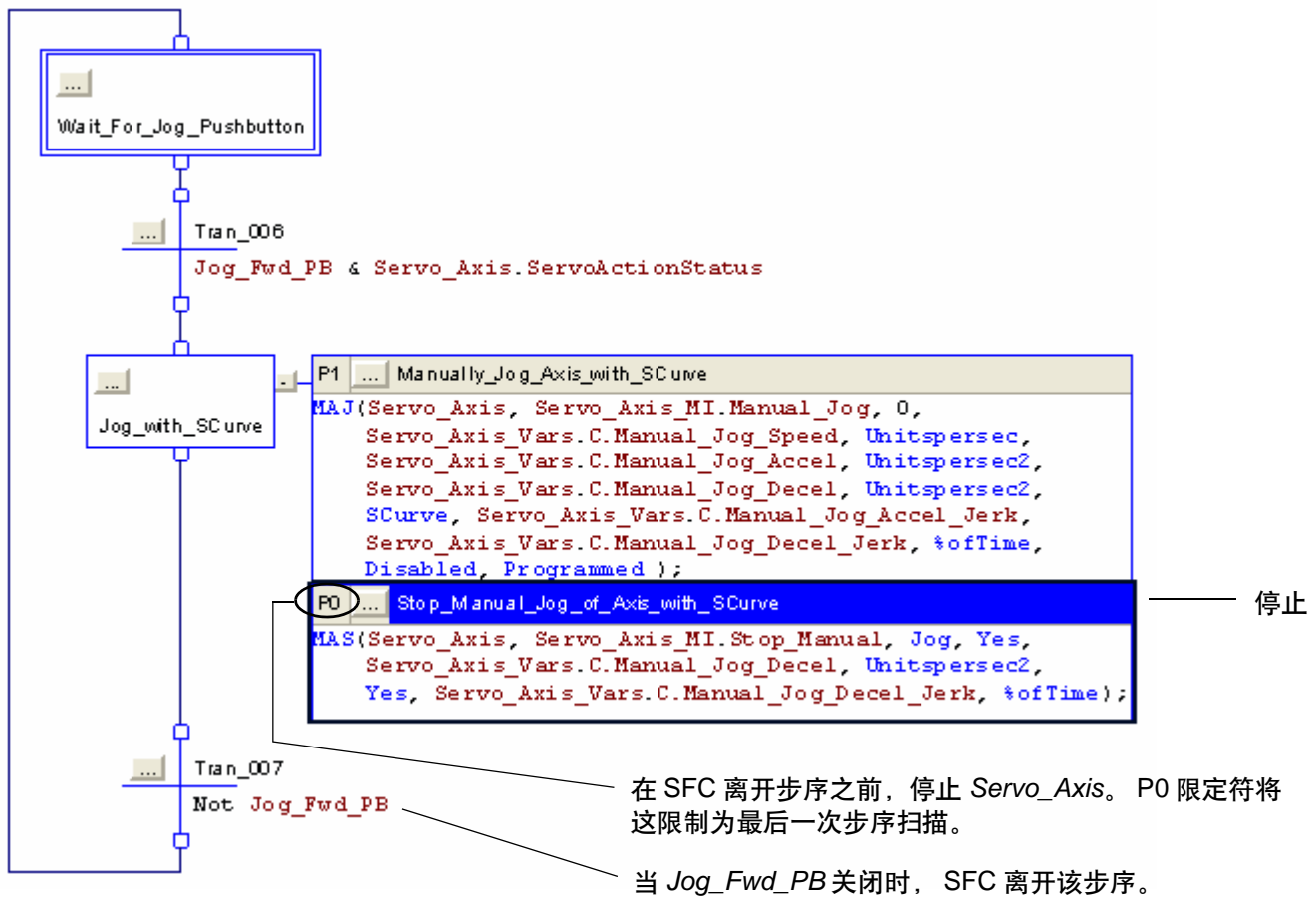
MAS 指令使用 S 曲线模式来停止轴，因为：

- MAJ 指令使用 S 曲线模式。
- 停止类型对于 MAS 指令为点动。

梯形图



结构文本



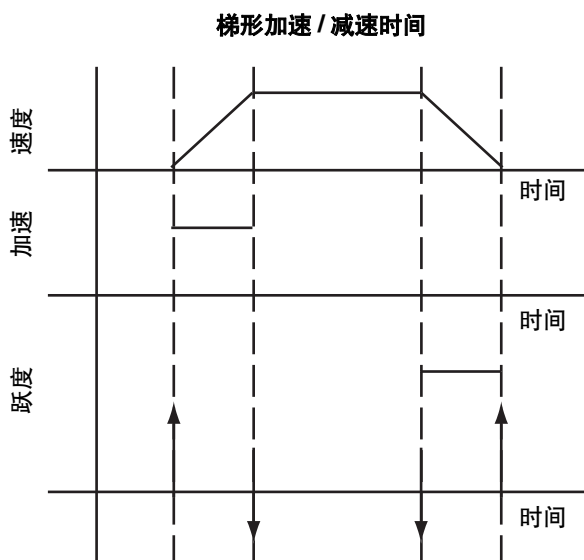
模式操作数

这个操作数有两种模式类型：

- [梯形速度模式](#)
- [S 曲线速度模式](#)

梯形速度模式

梯形速度模式是最常用的模式，因为它对于编程后续运动提供最大的灵活性，并提供最快的加速和减速时间。每单位时间的速度的变化是由加速度和减速度所指定的。跃度不是梯形模式中的一个要素。因此，跃度被认为是无限的，在下图中显示为一条垂直线。



S 曲线速度模式

当需要将对机械系统的压力和负荷降至最低时，S 曲线速度模式最为常用。使用两个附加的参数即加速度跃度和减速度跃度，根据机器的压力来平衡加速和减速时间。

根据跃度设置，加速度模式可以设置为几乎纯长方形（请参阅[梯形加速 / 减速时间](#)，在[页码 68](#)上，（最快，压力最大）），或者设置为三角形（请参阅[可编程 S 曲线加速 / 减速时间，加速度跃度 = 时间 60%](#)，在[页码 71](#)上，（最慢，压力最小））。

典型的加速模式为压力盒速度之间的折中，如[S 曲线加速 / 减速时间，向后兼容性设置：加速度跃度 = 时间 100%](#)所示（在[页码 71](#)上）。

跃度由用户指定（为单位 / 秒³ 或者最大值的百分比），或者由时间百分比计算而出。（在加速或减速模式中，时间百分比等于斜坡时间按的百分比）：

$$j_a [\text{EU/s}^3] = \frac{a_{\max}^2 [\text{EU/s}^2]}{v_{\max} [\text{EU/s}]} \left(\frac{200}{j_a [\text{时间 \%}]} - 1 \right)$$

$$j_a [\text{EU/s}^3] = \frac{d_{\max}^2 [\text{EU/s}^2]}{v_{\max} [\text{EU/s}]} \left(\frac{200}{j_a [\text{时间 \%}]} - 1 \right)$$

要牢记的一点是关于向后兼容性。跃度为时间 100% 将产生三角形的加速和减速模式。这些模式为上面[S 曲线加速 / 减速时间，向后兼容性设置：加速度跃度 = 时间 100%](#)（在[页码 71](#)上）所示的模式。

非常小的跃度值，即小于时间的 5%，将产生接近于长方形的加速和减速模式，如[梯形加速 / 减速时间](#)（在[页码 68](#)上）中所示。

重要提示

时间 % 值越高，所产生的跃度值限制越低，因此模式也就更慢。参见下表作为参考。

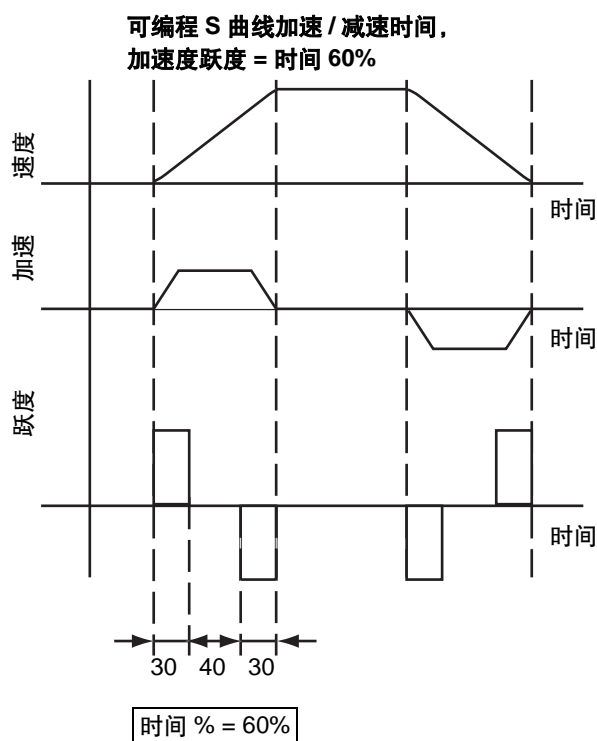
	(1) 梯形速度模式	S 形速度模式，且 1 <= 跃度 < 时间 100% ⁽²⁾	S 形速度模式， 且跃度 = 时间 100% ⁽³⁾
加速度 / 减速度跃度，单位采用单位 / 秒 ³	∞	$\frac{\text{最大加速度}^2}{\text{最大速度}}$ 至 ∞	$\frac{\text{最大加速度}^2}{\text{最大速度}}$
加速度 / 减速度跃度，单位采用最大值 %	不适用	0 - 100%	不适用
加速度 / 减速度跃度，单位采用时间 %	0%	1 - 100%	100%

(1) [页码 68](#) 上的示例（标签为梯形加速 / 减速时间）使用长方形的加速模式。

(2) [页码 71](#) 上的示例（标签为可编程 S 曲线加速 / 减速时间，加速度跃度 = 时间 60%）使用梯形的加速度模式。

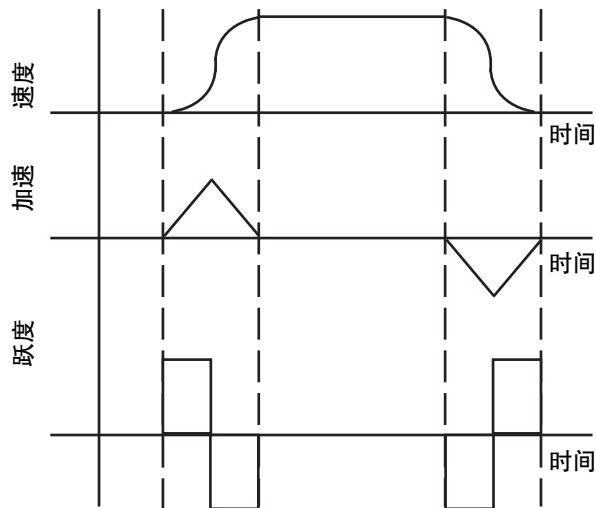
(3) [页码 71](#) 上的示例（标签为 S 曲线加速 / 减速时间，向后兼容性设置：加速度跃度 = 时间 100%）使用三角形的加速度模式。

当发起轴位移、更改动态参数或停止类型为位移或点动的 MCS 停止时，即进行计算。



S 曲线加速 / 减速时间，向后兼容性


设置：加速度跃度 = 时间 100%



运动轴寻参 (MAH)

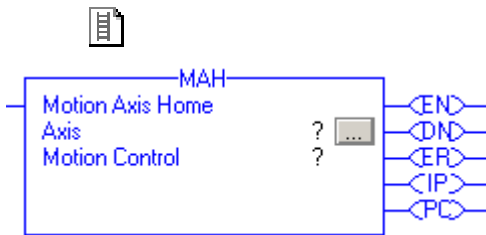
使用 MAH 指令来使某个轴归零。在进行轴配置中，可选择两种不同的归零模式：主动或被动。如果选择主动归零序列，则轴执行所配置的归零序列类型，并建立一个绝对值轴位置。但是，如果选择被动归零，则不会执行特定的归零序列，轴将保持等待下一个标记脉冲来建立归零位置。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数： 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_ 指令	标签	用于访问指令状态参数的结构。



MAH(Axis,MotionControl);

结构文本

操作数与梯形图 MAH 指令的操作数相同。

MOTION_INSTRUCTION 数据类型

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴归零已经成功完成后，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。
.IP(正在进行) 位 27	这个位在正的梯级转换时设置，在运动轴寻参完成后或者被停止命令、关闭或伺服故障终止时清除。
.PC(过程完成) 位 26	当轴归零已经成功完成后，过程完成位被设置。

说明： 运动轴寻参 (MAH) 指令用于校准指定轴的绝对位置。对于配置为伺服类型的轴来说，可使用主动、被动或绝对归零模式配置来进行归零。对于仅反馈型的轴，仅提供被动和绝对归零模式。绝对归零模式要求轴安装有绝对值反馈装置。

主动归零

当轴的归零模式被配置为主动时，物理轴首先被激活进行伺服操作。作为这个过程的一部分，所有其他正在进行的运动都被取消，相应的状态位被清除。然后使用所配置的归零序列将轴归零，可能是立即、开关、标记或开关 - 标记。后面三种归零序列的结果都是使轴按配置的归零方向点动，然后根据归零事件的监测重新定义位置以后，轴自动移动到所配置的归零位置。

重要提示

当在旋转轴上执行单向主动归零而且归零偏移值小于减速距离时，检测到归零事件后，控制器将轴移动到零退绕位置。这样可以保证所产生的向归零位置的运动是单向的。

被动归零

当轴归零模式被配置为被动时，MAH 在下一次出现编码器标记时重新定义物理轴的实际位置。被动归零最常用于将仅反馈轴校准到它们的标记，但也可以用于伺服轴。被动归零到达编码器标记时与主动归零相同，但运动控制器不会命令任何轴运动。

在发起被动归零后，轴必须移动通过编码器标记，才能让归零序列成功完成。对于闭环伺服轴，这可能通过 MAM 或 MAJ 指令来完成。对于物理的仅反馈轴，运动不能直接由运动控制器来命令，而必须通过其他方式来实现。

绝对归零

如果运动轴支持绝对反馈装置，则可以使用绝对归零模式。对于绝对归零模式，唯一有效的归零序列是“立即”。在这种情况下，绝对归零过程通过将配置的归零位置应用到绝对反馈装置所报告的位置，建立真正的轴的绝对位置。在通过 MAH 指令执行绝对归零过程之前，轴必须处于轴就绪状态，而且伺服环被禁止。

要在配置为主动归零模式的轴上成功执行 MAH 指令，目标轴必须配置为伺服轴类型。要成功执行 MAH 指令，目标轴必须配置为伺服或者反馈唯一轴。如果不满足任何条件，则指令出错。

重要提示

当 MAH 指令初次执行时，正在进行 (.IP) 位被设置，过程完成 (.PC) 位被清除。MAH 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输多条消息。完成 (.DN) 位并不立即设置，而只是在这些消息被成功发送以后。在完成 (.DN) 位被设置的同时，正在进行 (.IP) 位被清除，过程完成 (.PC) 位被设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#) 上）。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAH 指令收到伺服消息失败 (12) 错误消息或者非法归零配置 (41) 时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	过程根据请求终止 (1)	跟随某个关闭 / 禁止变频器的指令或者运动停止指令的归零执行，或者处理器变更请求了一次归零取消。
SERVO_MESSAGE_FAILURE (12)	无资源 (2)	没有足够的内存资源来完成请求。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	对象模式冲突 (12)	轴处于关闭状态。
SERVO_MESSAGE_FAILURE (12)	权限被拒绝 (15)	使能输入开关错误。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)	重新定义位置、归零和记录 2 是相互排斥的 (SERCOS)，设备状态不能用于动作。(SERCOS)

相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
ILLEGAL_HOMING_CONFIG (41)	归零序列 (4)	当归零序列不是立即时，您使用一条绝对值归零指令。
ILLEGAL_HOMING_CONFIG (41)	归零速度为零 (6)	归零速度不能为零。
ILLEGAL_HOMING_CONFIG (41)	归零返回速度为零 (7)	归零返回速度不能为零。

对于错误代码 54 – 最大减速度值为零，如果扩展错误代码返回一个正数 (0-n)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的轴最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

状态位：MAH 对状态位的更改

位名称	状态	含义
HomingStatus	真	轴正在归零
JogStatus	假	轴不再点动 *
MoveStatus	假	轴不再位移 *
GearingStatus	假	轴不再传动
StoppingStatus	假	轴不再停止

在执行主动归零序列的各个部分时，以下位可能会被设置或清除。在归零序列中，MAH 指令使用位移和点动运动模式来移动轴。这同时意味着因其他运动控制指令对位移或点动运动模式的任何中断，都可能影响到 MAH 发起的归零序列的成功完成。

如果在被动归零模式，MAH 指令只是设置归零状态位。

轴速度 (MAJ)

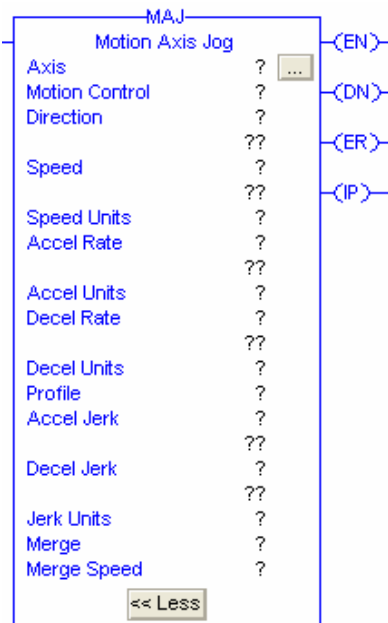
使用 MAJ 指令可以恒定速度移动轴，直到您告诉它停止为止。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数： 梯形图



操作数	类型	格式	描述						
轴	AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要点动的轴的名称						
运动控制	MOTION_INSTRUCTION	标签	指令的控制标签						
方向	DINT	立即数标签	<table border="1"> <thead> <tr> <th>对于这个点动方向</th> <th>输入</th> </tr> </thead> <tbody> <tr> <td>正向</td> <td>0</td> </tr> <tr> <td>反向</td> <td>1</td> </tr> </tbody> </table>	对于这个点动方向	输入	正向	0	反向	1
对于这个点动方向	输入								
正向	0								
反向	1								
速度	REAL	立即数标签	移动轴的速度，采用速度单位。						
速度单位	DINT	立即	您要为速度使用什么单位？ <ul style="list-style-type: none"> • 单位每秒 (0) • 最大值 %(1) 						

操作数	类型	格式	描述
加速度	REAL	立即数标签	轴的加速度，采用加速度单位
加速度单位	DINT	立即	您要为加速度使用什么单位？ <ul style="list-style-type: none"> • 单位每秒² (0) • 最大值 % (1)
减速度	REAL	立即数标签	轴的减速度，采用减速度单位
减速度单位	DINT	立即	您要为减速度使用什么单位？ <ul style="list-style-type: none"> • 单位每秒² (0) • 最大值 % (1)
模式	DINT	立即	选择运行点动的速度模式： <ul style="list-style-type: none"> • 梯形 (0) • S 曲线 (1) <p>有关更多信息，请参见编写一个速度模式程序上的页码 24。</p>
加速跃度	REAL	立即或标签	您 必须 始终为加速度和减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 <ul style="list-style-type: none"> • 加速跃度是轴的加速跃度值。 • 减速跃度是轴的减速跃度值。 <p>使用这些值起步。</p> <ul style="list-style-type: none"> • 加速度跃度 = 100 (时间 %) • 减速度跃度 = 100 (时间 %) • 跃度单位 = 2 <p>用以下跃度单位输入跃度速率。</p> <p>0 = 单位每秒³</p> <p>1 = 最大值 %</p> <p>2 = 时间 % (使用这个值起步)</p>
减速跃度	REAL	立即或标签	
跃度单位	DINT	立即	
合并	DINT	立即	您是否要将所有当前轴运动变成由这个指令控制的纯点动，而不管当前正在进行的运动控制指令如何？ <ul style="list-style-type: none"> • 否 – 选择禁止 (0) • 是 – 选择使能 (1)
合并速度	DINT	立即	如果合并被使能，您需要以什么速度点动？ <ul style="list-style-type: none"> • 此指令的速度 – 选择编程 (0) • 轴的当前速度 – 选择当前 (1)



MAJ(Axis,MotionControl, Direction,Speed, SpeedUnits,AccelRate, AccelUnits,DecelRate, DecelUnits,Profile, AccelJerk,DecelJerk, JerkUnits,Merge, MergeSpeed);

结构文本

结构文本操作数与梯形图操作数相同。

此操作数	拥有这些选项可让您	
	输入为文本	或者输入为数字
轴	无枚举	标签
MotionControl	无枚举	标签
方向	无枚举	立即或标签
速度	无枚举	立即或标签
SpeedUnits	单位每秒 最大值 %	0 1
AccelRate		
AccelUnits	单位每秒 ² 最大值 %	0 1
DecelRate		
DecelUnits	单位每秒 ² 最大值 %	0 1
模式	梯形 曲线	0 1
AccelJerk	无枚举	立即或标签
DecelJerk	无枚举	您 必须 始终为加速度和减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这些值起步。 <ul style="list-style-type: none"> • 加速度跃度 = 100 (时间 %) • 减速度跃度 = 100 (时间 %)
跃度单位	单位每秒 ³ 最大值 % % 时间	0 1 2(使用这个值起步)
合并	已禁止 已使能	0 1
MergeSpeed	编程 当前	0 1

MOTION_INSTRUCTION

数据类型

要查看是否	检查这个位是否被设置	数据类型	注意:
从假到真的过渡将导致这个指令执行	EN	BOOL	EN 位保持设置状态, 直到过程完成、梯级变为假为止。
点动被成功发起	DN	BOOL	
出现错误	ER	BOOL	
轴正在点动	IP	BOOL	<p>这些操作的任何一个都会停止此点动, 并清除 IP 位:</p> <ul style="list-style-type: none"> • 另外一条 MAJ 指令取代这条 MAJ 指令 • MAS 指令 • 来自另一条指令的合并 • 关闭命令 • 故障操作

说明： 使用 MAJ 指令可以恒定速度移动轴，与位置无关。

编程原则

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或者在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。

有关更多信息，请参见 [Troubleshoot Axis Motion](#) 上的 [page 9](#)。

指导原则	详细说明
<ul style="list-style-type: none"> • 在梯形图中，每当您希望执行指令时，请切换梯级条件。 	<p>这是一条转换指令：</p> <ul style="list-style-type: none"> • 在梯形图中，每当您希望执行指令时，将梯级输入条件从清除切换到设置即可。
<ul style="list-style-type: none"> • 在结构文本中，限制指令只对转换执行。 	<p>在结构文本中，指令每次被扫描时就会执行。</p> <ul style="list-style-type: none"> • 限制指令只对转换执行。使用下面的方法之一： <ul style="list-style-type: none"> • SFC 操作限定符 • 结构文本构建 <p>有关更多信息，请参阅附件 C。</p>
<ul style="list-style-type: none"> • 使用跃度操作数用于 S 曲线模式。 	<p>当 MAJ 指令使用 S 曲线模式时，使用跃度操作数。</p> <p>您必须填写跃度操作数，不管使用哪种模式。</p>
<ul style="list-style-type: none"> • 在最初的跃度编程和整定中使用时间 %。 	<p>为了使跃度的编程和整定更加方便，请将其输入为加速或减速时间的 %。</p> <p>有关更多信息，请参阅：</p> <ul style="list-style-type: none"> • 编写一个速度模式程序 在 页码 24 • 整定 S 曲线模式，在 页码 387 上。

指导原则	详细说明												
<ul style="list-style-type: none"> 使用合并来取消其他指令的运动。 	<p>您希望如何处理已经正在进行的任何运动？</p> <table border="1"> <thead> <tr> <th>如果您需要</th> <th>而且您需要</th> <th>那么设置</th> </tr> </thead> <tbody> <tr> <td>将点动加入任何正在进行的运动</td> <td>—————▶</td> <td>合并 = 禁止 合并速度 = 编程</td> </tr> <tr> <td>从其他指令中结束该运动，而只是点动</td> <td>按您在这个指令中设置的速度点动</td> <td>合并 = 使能 合并速度 = 编程</td> </tr> <tr> <td></td> <td>按轴当前正在运动的速度点动</td> <td>合并 = 使能 合并速度 = 当前 该指令忽略您在速度操作数中放置的值。</td> </tr> </tbody> </table>	如果您需要	而且您需要	那么设置	将点动加入任何正在进行的运动	—————▶	合并 = 禁止 合并速度 = 编程	从其他指令中结束该运动，而只是点动	按您在这个指令中设置的速度点动	合并 = 使能 合并速度 = 编程		按轴当前正在运动的速度点动	合并 = 使能 合并速度 = 当前 该指令忽略您在速度操作数中放置的值。
如果您需要	而且您需要	那么设置											
将点动加入任何正在进行的运动	—————▶	合并 = 禁止 合并速度 = 编程											
从其他指令中结束该运动，而只是点动	按您在这个指令中设置的速度点动	合并 = 使能 合并速度 = 编程											
	按轴当前正在运动的速度点动	合并 = 使能 合并速度 = 当前 该指令忽略您在速度操作数中放置的值。											
<ul style="list-style-type: none"> 小心注意在轴已经正在点动时您是否开始另一个点动。 	<p>如果已经有点动正在进行中而您又启动一个新的 MAJ 指令，则可能导致：</p> <ul style="list-style-type: none"> 轴加速从而过冲其速度 轴减速至反转 (版本 15 和更早) <p>如果 MAJ 指令使用 S 曲线模式，则会发生这种情况。</p> <p>原因：新的 MAJ 指令取消老的 MAJ 指令。轴使用新指令中的速度、加速度、减速度和跃度。</p> <p>有关更多信息，请参见 Troubleshoot Axis Motion 上的 page 9。</p>												
<ul style="list-style-type: none"> 使用 MAS 指令来停止点动。 	参见示例。												
<ul style="list-style-type: none"> 使用 MCD 指令更改点动时的速度。 	参见 示例 1 (在 页码 84 上)。												

算术状态标志： 不受影响

故障条件： 无

错误代码 参见 [运动控制指令错误代码 \(ERR\)](#) (在 [页码 391](#) 上)。

扩展错误代码 使用扩展错误代码 (EXERR) 可得到关于错误的更多信息。

如果 ERR 为	并且 EXERR 是	那么	原因	处理措施										
13	各种		某个操作数超出其范围。	<p>EXERR 是超出范围的操作数的编号。第一个操作数是 0。</p> <p>例如，如果 EXERR = 3，那么就检查速度。</p> <table border="1"> <thead> <tr> <th>EXERR</th> <th>操作数</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>轴</td> </tr> <tr> <td>1</td> <td>运动控制</td> </tr> <tr> <td>2</td> <td>方向</td> </tr> <tr> <td>3</td> <td>速度</td> </tr> </tbody> </table>	EXERR	操作数	0	轴	1	运动控制	2	方向	3	速度
EXERR	操作数													
0	轴													
1	运动控制													
2	方向													
3	速度													
54	-1		坐标系的最大减速度为 0。	转到坐标系的属性中，设置最大减速度。										
	0 或更多		坐标系中的某个轴的最大减速度为 0。	<ol style="list-style-type: none"> 1. 打开该轴的属性。 2. 使用 EXERR 值查看哪个轴的最大减速度为 0。 3. 您正在点动的轴的减速度为 0。 										

对状态位的更改： *运动状态位*

如果合并为	那么指令更改这些位		
	位名称	状态	含义
禁止	JogStatus	真	轴正在点动。
使能	JogStatus	真	轴正在点动。
	MoveStatus	假	轴不再运动。
	GearingStatus	假	轴不再传动。

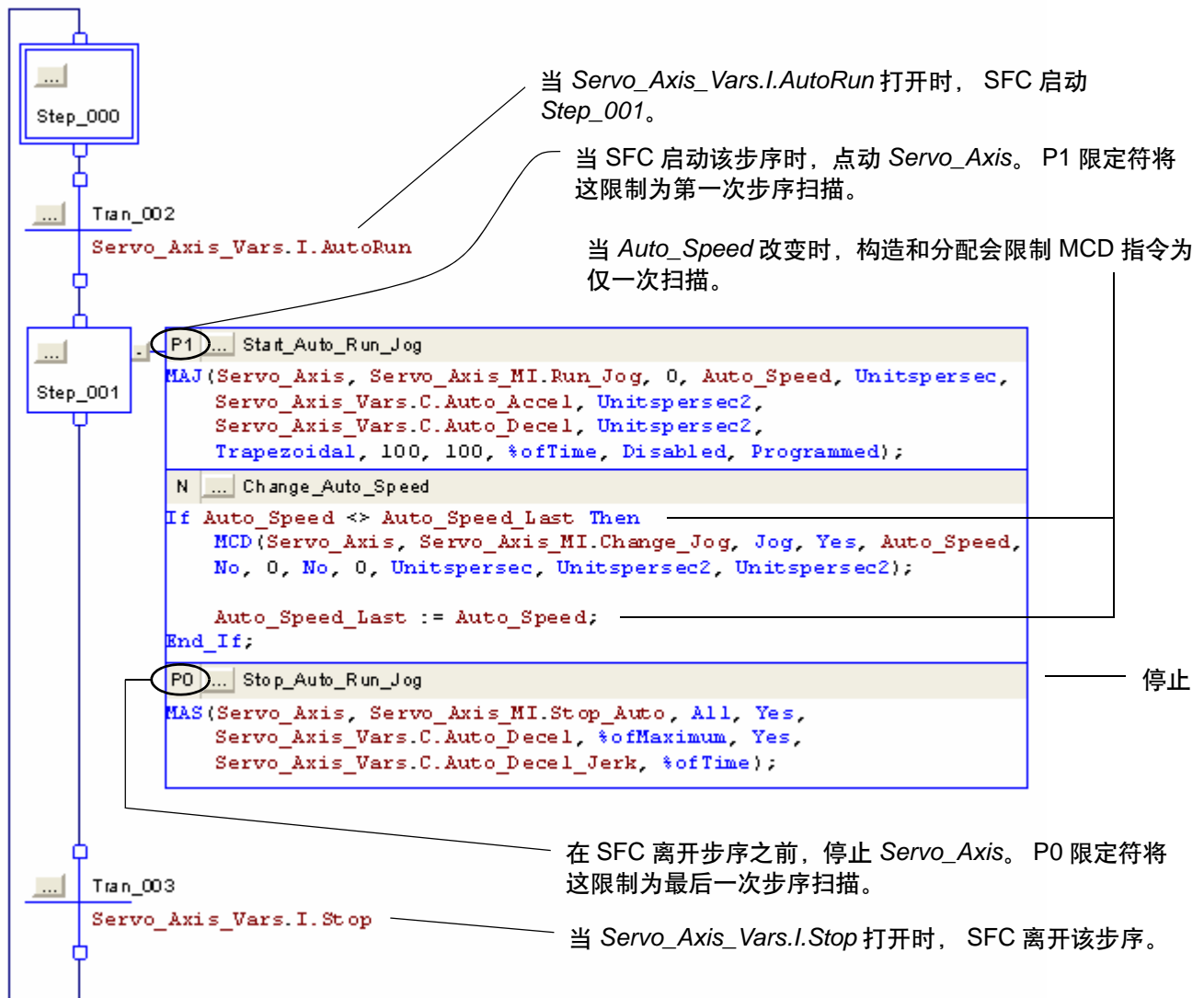
点动时的速度更改 - 结构文本

当 `Servo_Axis_Vars.I.AutoRun` 打开时

运行 `Servo_Axis` at `Auto_Speed`。

如果 `Auto_Speed` 变化，则将点动速度更改为 `Auto_Speed` 的新值。

当 `Servo_Axis_Vars.I.Stop` 打开时，停止 `Servo_Axis`。



示例 2 使用 S 曲线正向和反向点动 – 梯形图

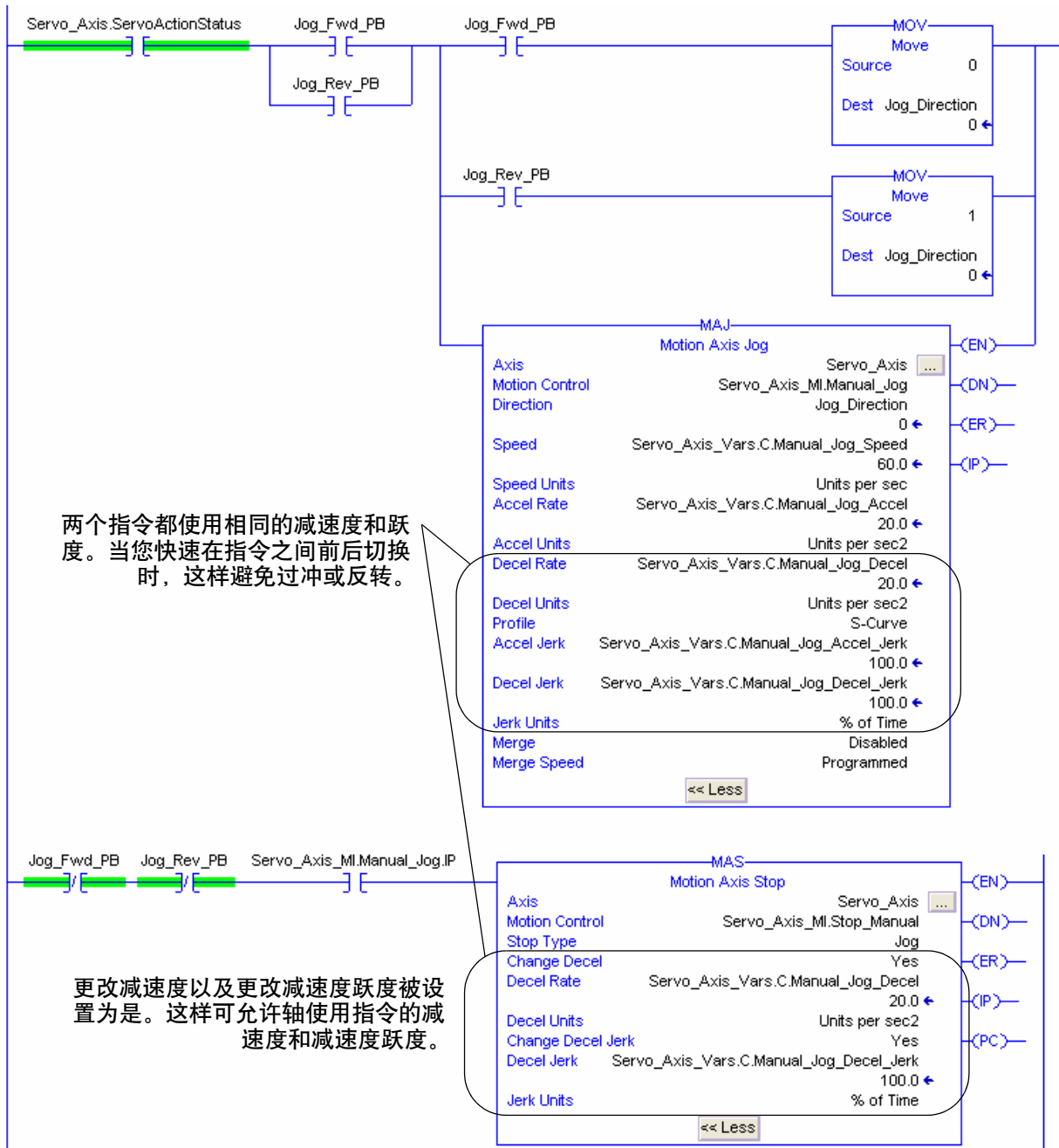
当伺服环被使能后

并且 Jog_Fwd_PB 或 Jog_Rev_PB 打开

设置 Jog_Direction。

运行 Servo_Axis, 按 Servo_Axis_Vars.C.Manual_Jog_Speed 速度。

当 Jog_Fwd_PB 和 Jog_Rev_PB 关闭时, 停止 Servo_Axis。



使用 S 曲线正向和反向点动 – 结构文本

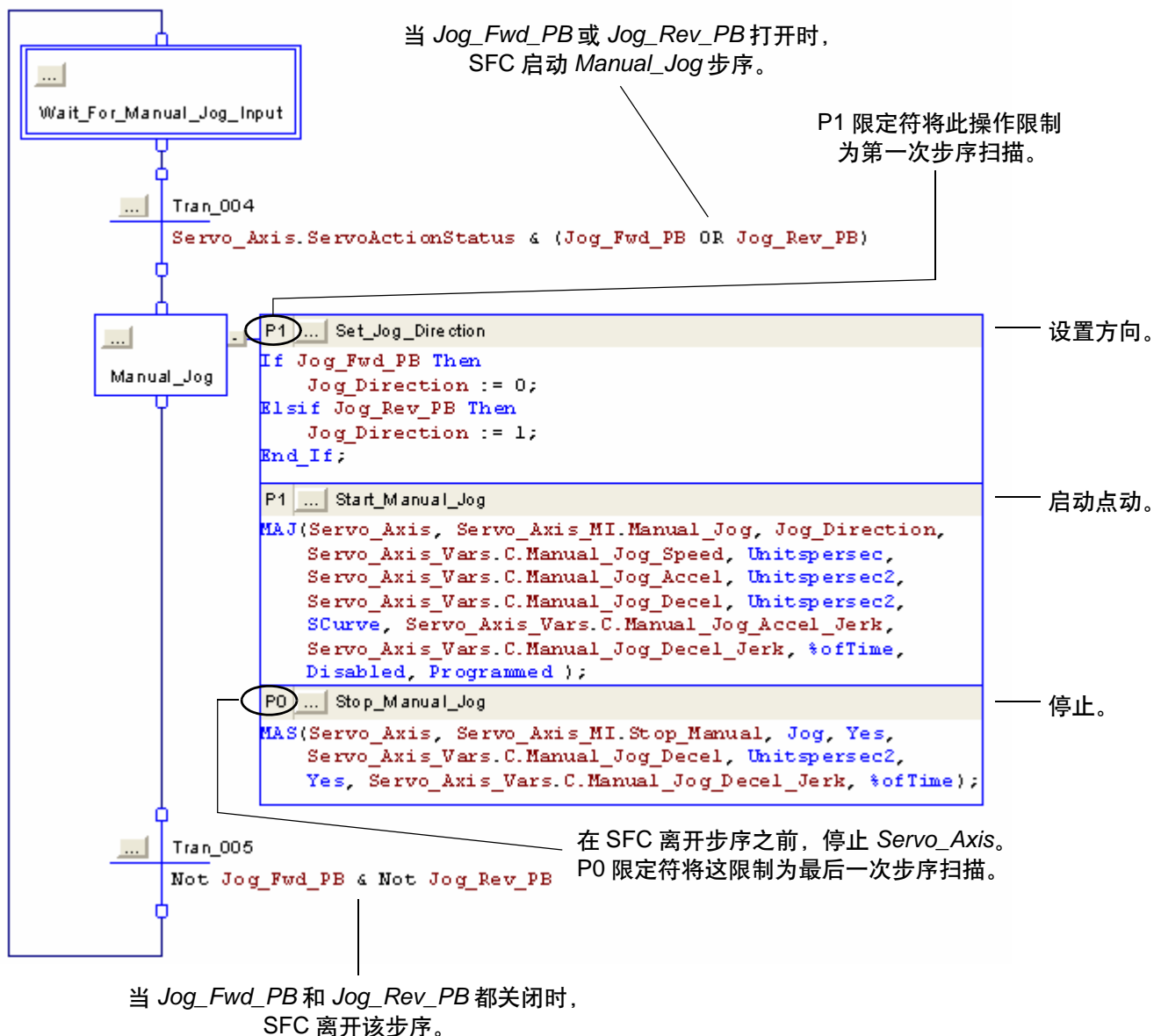
当伺服环被使能后

并且 *Jog_Fwd_PB* 或 *Jog_Rev_PB* 打开

设置 *Jog_Direction*。

运行 *Servo_Axis*，按 *Servo_Axis_Vars.C.Manual_Jog_Speed* 速度。

当 *Jog_Fwd_PB* 和 *Jog_Rev_PB* 关闭时，停止 *Servo_Axis*。



模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

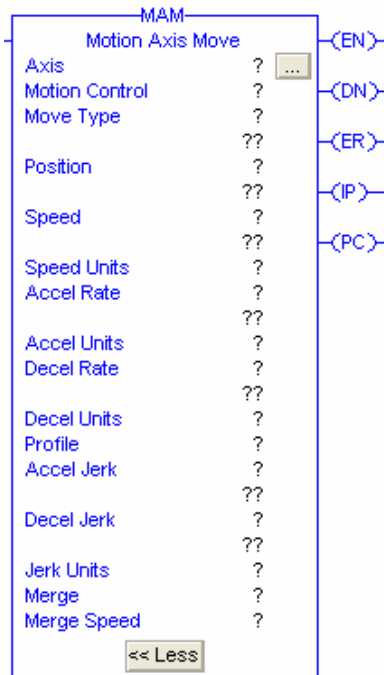
轴位移 (MAM)

使用轴位移 (MAM) 指令可将轴移动到指定的位置。



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数： *梯形图*



操作数	类型	格式	描述
轴	AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	轴的名称 对于绝对值或增量主轴偏移位移，请输入从轴。

操作数	类型	格式	描述																								
运动控制	MOTION_INSTRUCTION	标签	指令的控制标签																								
运动类型	DINT	立即数标签	<table border="1"> <thead> <tr> <th>要</th> <th>使用这种位移类型</th> <th>并且输入</th> </tr> </thead> <tbody> <tr> <td>将轴移到一个绝对位置</td> <td>绝对值</td> <td>0</td> </tr> <tr> <td>将轴移到距当前位置的指令距离</td> <td>增量</td> <td>1</td> </tr> <tr> <td>将旋转轴以最短方向移到某个绝对位置，不管其当前位置如何</td> <td>旋转最短路径</td> <td>2</td> </tr> <tr> <td>将旋转轴以正的方向移到某个绝对位置，不管其当前位置如何</td> <td>旋转正向</td> <td>3</td> </tr> <tr> <td>将旋转轴以负的方向移到某个绝对位置，不管其当前位置如何</td> <td>旋转负向</td> <td>4</td> </tr> <tr> <td>将位置凸轮主轴值偏移到一个绝对位置</td> <td>绝对主轴偏移</td> <td>5</td> </tr> <tr> <td>将位置凸轮主轴值偏移一个增量距离</td> <td>增量主轴偏移</td> <td>6</td> </tr> </tbody> </table> <p>请参阅为旋转轴选择一种位移类型（在页码 95上），了解关于旋转运动的更多信息。</p>	要	使用这种位移类型	并且输入	将轴移到一个绝对位置	绝对值	0	将轴移到距当前位置的指令距离	增量	1	将旋转轴以 最短 方向移到某个绝对位置，不管其当前位置如何	旋转最短路径	2	将旋转轴以 正 的方向移到某个绝对位置，不管其当前位置如何	旋转正向	3	将旋转轴以 负 的方向移到某个绝对位置，不管其当前位置如何	旋转负向	4	将位置凸轮主轴值偏移到一个绝对位置	绝对主轴偏移	5	将位置凸轮主轴值偏移一个增量距离	增量主轴偏移	6
要	使用这种位移类型	并且输入																									
将轴移到一个绝对位置	绝对值	0																									
将轴移到距当前位置的指令距离	增量	1																									
将旋转轴以 最短 方向移到某个绝对位置，不管其当前位置如何	旋转最短路径	2																									
将旋转轴以 正 的方向移到某个绝对位置，不管其当前位置如何	旋转正向	3																									
将旋转轴以 负 的方向移到某个绝对位置，不管其当前位置如何	旋转负向	4																									
将位置凸轮主轴值偏移到一个绝对位置	绝对主轴偏移	5																									
将位置凸轮主轴值偏移一个增量距离	增量主轴偏移	6																									
位置	REAL	立即标签	<table border="1"> <thead> <tr> <th>位移的绝对位置或增量距离</th> <th>对于这种位移类型</th> <th>输入这个位置值</th> </tr> </thead> <tbody> <tr> <td>绝对值</td> <td></td> <td>要位移到的位置</td> </tr> <tr> <td>增量</td> <td></td> <td>要位移的距离</td> </tr> <tr> <td>旋转最短路径</td> <td></td> <td></td> </tr> <tr> <td>旋转正向</td> <td></td> <td>要位移到的位置。输入一个小于位置退绕值的正值。</td> </tr> <tr> <td>旋转负向</td> <td></td> <td></td> </tr> <tr> <td>绝对主轴偏移</td> <td></td> <td>绝对偏移位置</td> </tr> <tr> <td>增量主轴偏移</td> <td></td> <td>增量偏移距离</td> </tr> </tbody> </table>	位移的绝对位置或增量距离	对于这种位移类型	输入这个位置值	绝对值		要位移到的位置	增量		要位移的距离	旋转最短路径			旋转正向		要位移到的位置。输入一个小于位置退绕值的正值。	旋转负向			绝对主轴偏移		绝对偏移位置	增量主轴偏移		增量偏移距离
位移的绝对位置或增量距离	对于这种位移类型	输入这个位置值																									
绝对值		要位移到的位置																									
增量		要位移的距离																									
旋转最短路径																											
旋转正向		要位移到的位置。输入一个小于位置退绕值的正值。																									
旋转负向																											
绝对主轴偏移		绝对偏移位置																									
增量主轴偏移		增量偏移距离																									
速度	REAL	立即数标签	移动轴的速度，采用速度单位。																								
速度单位	DINT	立即	您要为速度使用什么单位？ <ul style="list-style-type: none"> • 单位每秒 (0) • 最大值 %(1) 																								
加速度	REAL	立即数标签	轴的加速度，采用加速度单位																								
加速度单位	DINT	立即	您要为加速度使用什么单位？ <ul style="list-style-type: none"> • 单位每秒² (0) • 最大值 %(1) 																								
减速度	REAL	立即数标签	轴的减速度，采用减速度单位																								
减速度单位	DINT	立即	您要为减速度使用什么单位？ <ul style="list-style-type: none"> • 单位每秒² (0) • 最大值 %(1) 																								

操作数	类型	格式	描述
模式	DINT	立即	选择运行 位移的速度模式： <ul style="list-style-type: none"> • 梯形 (0) • S 曲线 (1) 有关更多信息，请参见 编写一个速度模式程序上的页码 24 。
加速跃度	REAL	立即数标签	只有当模式为 S 曲线时，指令才会使用跃度操作数。但是您始终都要填写跃度值。 <ul style="list-style-type: none"> • 加速跃度是轴的加速跃度值。 • 减速跃度是轴的减速跃度值。 使用这些值起步。 <ul style="list-style-type: none"> • 加速度跃度 = 100 • 减速度跃度 = 100 • 跃度单位 = 2 (时间 %) 您也可以使用以下跃度单位输入跃度值。 <ul style="list-style-type: none"> • 单位每秒³ (0) • 最大值 % (1)
减速跃度	REAL	立即数标签	
跃度单位	DINT	立即	
合并	DINT	立即	您是否要将所有当前轴运动变成由这个指令控制的纯位移，而不管当前正在进行的运动控制指令如何？ <ul style="list-style-type: none"> • 否 – 选择禁止 (0) • 是 – 选择使能 (1)
合并速度	DINT	立即	如果合并被使能，您需要以什么速度位移？ <ul style="list-style-type: none"> • 此指令的速度 – 选择编程 (0) • 轴的当前速度 – 选择当前 (1)



MAM(Axis, MotionControl, MoveType, Position, Speed, SpeedUnits, AccelRate, AccelUnits, DecelRate, DecelUnits, Profile, AccelJerk, DecelJerk, JerkUnits, Merge, MergeSpeed);

结构文本

操作数与梯形图指令的操作数相同。

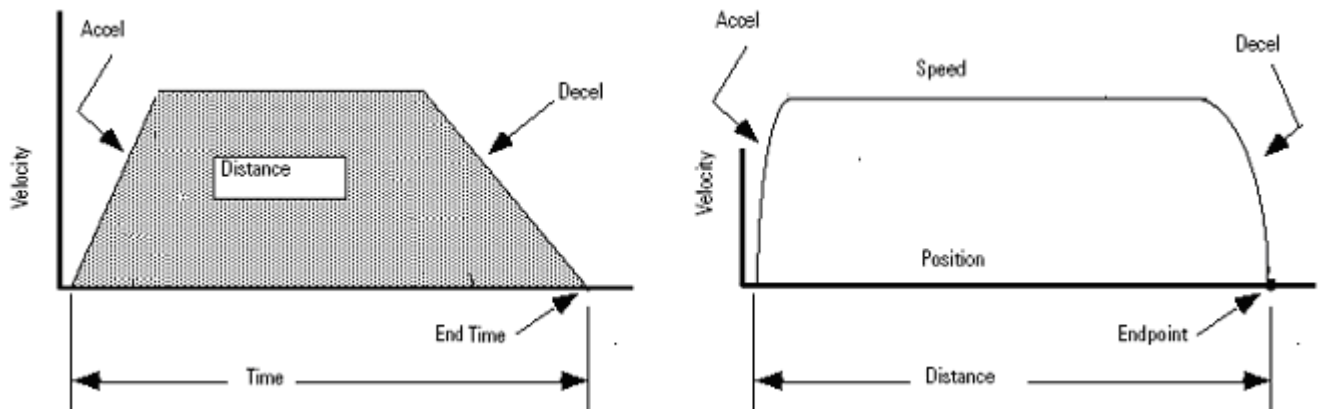
此操作数	拥有这些选项可让您	
	输入为文本	或者输入为数字
SpeedUnits	unitspersec	0
	%ofmaximum	1
AccelUnits	unitspersec2	0
	%ofmaximum	1
DecelUnits	unitspersec2	0
	%ofmaximum	1
模式	梯形	0
	曲线	1
JerkUnits	unitspersec3	0
	%ofmaximum	1
	%oftime	2
合并	已禁止	0
	已使能	1
MergeSpeed	programmed	0
	current	1

MOTION_INSTRUCTION 数据类型

要查看是否	检查这个位是否被设置	数据类型	注意:
从假到真的过渡将导致这个指令执行	EN	BOOL	EN 位保持设置状态, 直到过程完成、梯级变为假为止。
位移被成功发起	DN	BOOL	
出现错误	ER	BOOL	
轴正在向结束位置位移	IP	BOOL	这些操作的任何一个都会停止此位移, 并清除 IP 位: <ul style="list-style-type: none"> • 轴到达结束位置 • 另外一条 MAM 指令取代这条 MAM 指令 • MAS 指令 • 来自另一条指令的合并 • 关闭命令 • 故障操作
轴处于结束位置	PC	BOOL	<ul style="list-style-type: none"> • PC 位保持设置状态, 直到梯级进行一次从假到真的过渡。 • 在轴到达结束位置之前, 如果其他某个操作停止该位移, 则 PC 位保持清除状态。

说明: 轴位移 (MAM) 指令将轴移动到指定的绝对位置或者移动指定的增量距离。MAM 指令还能产生其他特殊类型的位移。

示例: 从静止开始的梯形位移



编程原则

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或者在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

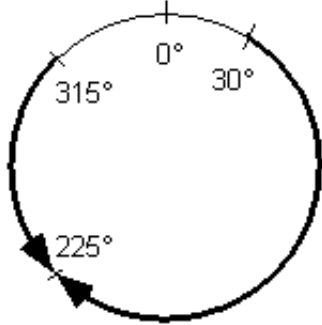
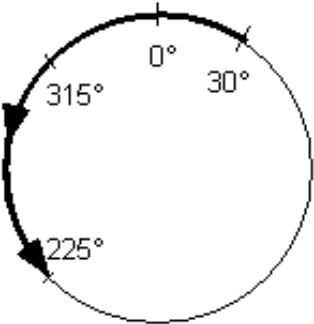
- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。

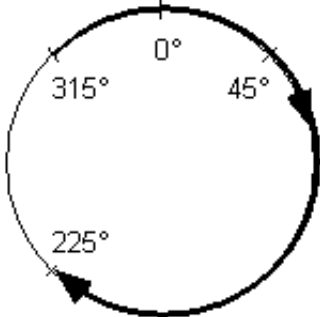
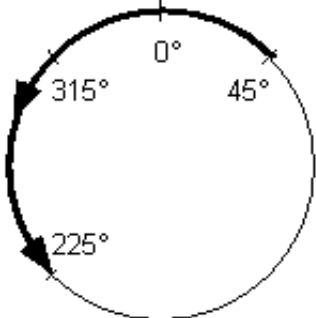
有关更多信息，请参见 [Troubleshoot Axis Motion](#) 上的 [page 9](#)。

指导原则	详细说明
<ul style="list-style-type: none"> • 在梯形图中，每当您希望执行指令时，请切换梯级条件。 	<p>这是一条转换指令：</p> <ul style="list-style-type: none"> • 在梯形图中，每当您希望执行指令时，将梯级输入条件从清除切换到设置即可。
<ul style="list-style-type: none"> • 在结构文本中，限制指令只对转换执行。 	<p>在结构文本中，指令每次被扫描时就会执行。</p> <ul style="list-style-type: none"> • 限制指令只对转换执行。使用下面的方法之一： <ul style="list-style-type: none"> • SFC 操作限定符 • 结构文本构建 <p>有关更多信息，请参阅附件 C。</p>
<ul style="list-style-type: none"> • 对于主轴偏移位移，请输入从轴，但使用主轴单位。 	<p>使用绝对值或增量主轴偏移位移来偏移位置凸轮的主轴值，而实际上不改变主轴的位置。这将沿着主轴平移位置凸轮曲线。</p> <ul style="list-style-type: none"> • 对于轴，请输入从轴。 • 对于位置，请输入绝对偏移位置或者增量偏移距离 • 对于速度、加速度、减速度和跃度，请为主轴输入这些值。 <p>指令加入速度、加速度、减速度和跃度值处的偏移。</p>
<ul style="list-style-type: none"> • 在最初的跃度编程和整定中使用时间 %。 	<p>为了使跃度的编程和整定更加方便，请将其输入为加速或减速时间的 %。</p> <p>有关更多信息，请参阅：</p> <ul style="list-style-type: none"> • 编写一个速度模式程序 在 页码 24 • 整定 S 曲线模式，在 页码 387 上。

指导原则	详细说明												
<ul style="list-style-type: none"> 使用合并来取消其他指令的运动。 	<p>您希望如何处理已经正在进行的任何运动？</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">如果您需要</th> <th style="text-align: left;">而且您需要</th> <th style="text-align: left;">那么设置</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">将位移加入任何正在进行的运动</td> <td style="text-align: center; vertical-align: middle;">—————▶</td> <td style="vertical-align: top;"> 合并 = 禁止 合并速度 = 编程 该指令忽略合并速度，但您还是要填写合并速度。 </td> </tr> <tr> <td style="vertical-align: top;">从其他指令中结束该运动，而只是位移</td> <td style="vertical-align: top;">按您在这个指令中设置的速度位移</td> <td style="vertical-align: top;">合并 = 使能 合并速度 = 编程</td> </tr> <tr> <td></td> <td style="vertical-align: top;">按轴当前正在运动的速度位移</td> <td style="vertical-align: top;"> 合并 = 使能 合并速度 = 当前 该指令忽略您在速度操作数中放置的值。 </td> </tr> </tbody> </table> <p>这是绝对值位移或增量主轴偏移位移吗？</p> <p>如果这是绝对值或增量主轴偏移位移，并且合并被使能，则</p> <ul style="list-style-type: none"> 位移仅结束已经正在进行的绝对值或增量主轴偏移位移。 位移不影响所有其他正在进行的运动。 	如果您需要	而且您需要	那么设置	将位移加入任何正在进行的运动	—————▶	合并 = 禁止 合并速度 = 编程 该指令忽略合并速度，但您还是要填写合并速度。	从其他指令中结束该运动，而只是位移	按您在这个指令中设置的速度位移	合并 = 使能 合并速度 = 编程		按轴当前正在运动的速度位移	合并 = 使能 合并速度 = 当前 该指令忽略您在速度操作数中放置的值。
如果您需要	而且您需要	那么设置											
将位移加入任何正在进行的运动	—————▶	合并 = 禁止 合并速度 = 编程 该指令忽略合并速度，但您还是要填写合并速度。											
从其他指令中结束该运动，而只是位移	按您在这个指令中设置的速度位移	合并 = 使能 合并速度 = 编程											
	按轴当前正在运动的速度位移	合并 = 使能 合并速度 = 当前 该指令忽略您在速度操作数中放置的值。											
<ul style="list-style-type: none"> 使用第二条 MAM 指令来更改正在进行的运动。 	<p>您可以更改位置、速度、加速度或减速度。更改将立即生效。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">要更改位置</th> <th style="text-align: left;">像这样设置第二条 MAM 指令。</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">绝对值位移</td> <td style="vertical-align: top;"> 选择任一： <ul style="list-style-type: none"> 将位移类型设置为绝对值，将位置设置为新位置。 将位移类型设置为增量，将位置设置为要更改结束位置的距离。新的结束位置是老的结束位置加上新的增量距离。 在任一情况下，轴移到新的位置而不会再老位置停止 – 包括任何要求的方向改变。 </td> </tr> <tr> <td style="vertical-align: top;">增量位移</td> <td style="vertical-align: top;"> 选择任一： <ul style="list-style-type: none"> 将位移类型设置为绝对值，将位置设置为新位置。轴直接进入新位置，而不完成增量位移。 将位移类型设置为增量，将位置设置为附加距离。轴将位移两个增量位移之和。 </td> </tr> </tbody> </table>	要更改位置	像这样设置第二条 MAM 指令。	绝对值位移	选择任一： <ul style="list-style-type: none"> 将位移类型设置为绝对值，将位置设置为新位置。 将位移类型设置为增量，将位置设置为要更改结束位置的距离。新的结束位置是老的结束位置加上新的增量距离。 在任一情况下，轴移到新的位置而不会再老位置停止 – 包括任何要求的方向改变。	增量位移	选择任一： <ul style="list-style-type: none"> 将位移类型设置为绝对值，将位置设置为新位置。轴直接进入新位置，而不完成增量位移。 将位移类型设置为增量，将位置设置为附加距离。轴将位移两个增量位移之和。 						
要更改位置	像这样设置第二条 MAM 指令。												
绝对值位移	选择任一： <ul style="list-style-type: none"> 将位移类型设置为绝对值，将位置设置为新位置。 将位移类型设置为增量，将位置设置为要更改结束位置的距离。新的结束位置是老的结束位置加上新的增量距离。 在任一情况下，轴移到新的位置而不会再老位置停止 – 包括任何要求的方向改变。												
增量位移	选择任一： <ul style="list-style-type: none"> 将位移类型设置为绝对值，将位置设置为新位置。轴直接进入新位置，而不完成增量位移。 将位移类型设置为增量，将位置设置为附加距离。轴将位移两个增量位移之和。 												
<ul style="list-style-type: none"> 将位移与齿轮合并可产生复杂模式和同步。 	<p>您可以将电子齿轮 (MAG) 指令与 MAM 指令一起使用。这将在位移上叠加齿轮，或者在齿轮上叠加位移。</p> <p>示例：在电子齿轮上叠加一个增量位移，可用于相位提前和延迟控制。</p>												

为旋转轴选择一种位移类型

运动类型	示例	描述
绝对值	<p>绝对位移至 225°。方向取决于轴的开始位置。</p> 	<p>对于绝对位移，行进的方向取决于轴的当前位置，不一定是到达结束位置的最短路径。如果开始位置小于结束位置，则会以正的方向运动；如果开始位置大于结束位置，则会以负的方向运动。</p> <p>指定的位置用三角法方式进行解释，可以为正或者为负。也可能大于位置退绕值。负的位置值等效于其相应的正值，当旋转轴通过 0 时非常有用。例如 -90° 与 +270° 相同。当位置大于位置退绕值时，轴移动超过一转后停止在绝对位置。</p>
增量		<p>指定的距离用三角法方式进行解释，可以为正或者为负。也可能大于位置退绕值。当距离大于位置退绕值时，轴移动超过一转后停止。</p>
旋转最短路径	<p>旋转最短路径从 30° 移动到 225°。</p> 	<p>重要事项：只有当轴的定位模式为旋转时（旋转轴），才能使用旋转最短路径位移。</p> <p>旋转最短路径位移是旋转轴的一种特殊的绝对值位移类型。轴将：</p> <ul style="list-style-type: none"> 以最短方向移动到指定的位置，不管其当前位置如何 如果需要将移动通过 0°。 <p>使用旋转最短路径位移，您：</p> <ul style="list-style-type: none"> 可以当轴正在位移或处于静止状态时开始位移 通过单次位移不能移动轴超过一转。

运动类型	示例	描述
旋转正向	旋转正向从 315° 移动到 225°。 	<p>重要事项：只有当轴处于静止状态不在位移时，再使用旋转正向位移。否则轴可能会以错误方向位移。</p> <p>旋转正向位移是旋转轴的一种特殊的绝对值位移类型。轴将：</p> <ul style="list-style-type: none"> • 以正向方向移动到指定的位置，不管其当前位置如何 • 如果需要将移动通过 0° <p>通过单次旋转最短路径位移不能移动轴超过一转。</p>
旋转负向	旋转负向从 45° 移动到 225°。 	<p>重要事项：只有在以下情况下使用旋转最短路径位移</p> <ul style="list-style-type: none"> • 当轴的定位模式为旋转时（旋转轴）。 • 当轴处于静止状态不在位移时。否则轴可能会以错误方向位移。 <p>旋转负向位移是旋转轴的一种特殊的绝对值位移类型。轴将：</p> <ul style="list-style-type: none"> • 以负向方向移动到指定的位置，不管其当前位置如何 • 如果需要将移动通过 0° <p>通过单次旋转最短路径位移不能移动轴超过一转。</p>

算术状态标志： 不受影响

故障条件： 无

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#)上）。

扩展错误代码： 使用扩展错误代码 (EXERR) 可得到关于错误的更多信息。

如果 ERR 为	并且 EXERR 是	那么	原因	处理措施												
13	各种		某个操作数超出其范围。	EXERR 是超出范围的操作数的编号。第一个操作数是 0。 例如，如果 EXERR = 4，那么就检查速度。												
				<table border="1"> <thead> <tr> <th>EXERR</th> <th>操作数</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>轴</td> </tr> <tr> <td>1</td> <td>运动控制</td> </tr> <tr> <td>2</td> <td>运动类型</td> </tr> <tr> <td>3</td> <td>位置</td> </tr> <tr> <td>4</td> <td>速度</td> </tr> </tbody> </table>	EXERR	操作数	0	轴	1	运动控制	2	运动类型	3	位置	4	速度
EXERR	操作数															
0	轴															
1	运动控制															
2	运动类型															
3	位置															
4	速度															
54	-1 0 或更多		坐标系的最大减速度为 0。 坐标系中的某个轴的最大减速度为 0。	转到坐标系轴的属性中，设置最大减速度。 1. 打开该轴的属性。 2. 使用 EXERR 值查看哪个轴的最大减速度为 0。 3. 您通过 MAM 指令正在位移的轴的减速度为 0。												

对状态位的更改： *运动状态位*

如果位移类型为	并且合并为	那么指令更改这些位	位名称	状态	含义
非绝对值主轴偏移或 增量主轴偏移	禁止		MoveStatus	真	轴正在位移
	使能		MoveStatus	真	轴正在位移
			JogStatus	假	轴不再点动
			GearingStatus	假	轴不再传动。
绝对值主轴偏移或增 量主轴偏移	—————▶	MasterOffsetMoveStat us	真	轴被偏移	

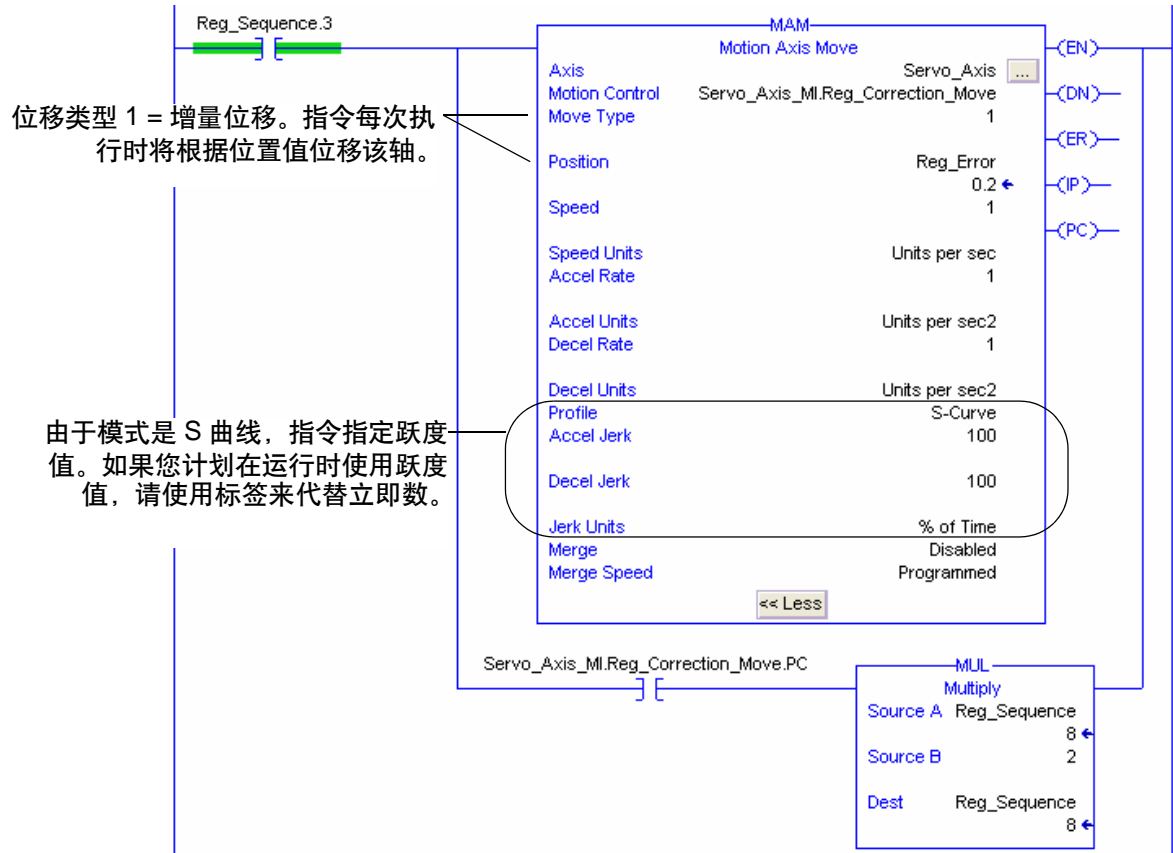
示例 位移-梯形图

这个示例使用 *Reg_Sequence* 的位模式来逐步完成逻辑。

当 *Reg_Sequence.3* 打开时

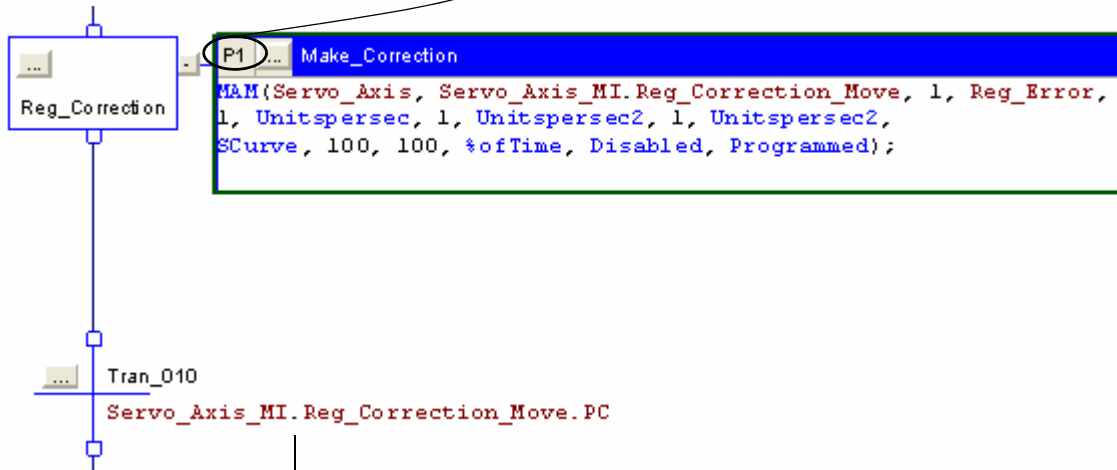
将 *Servo_Axis* 移动距离 *Reg_Error*。

当位移完成后，将 *Reg_Sequence* 乘以 2。这将关闭 *Reg_Sequence* 的位 3，打开位 4。



位移-结构文本

当 SFC 启动 *Reg_Correction* 步序时，将 *Servo_Axis* 位移距离 *Reg_Error*。P1 限定符将这限制为第一次步序扫描。



当位移完成后，位移的 PC 位打开。当 PC 位打开时，SFC 离开该步序。

模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

电子齿轮 (MAG)

电子齿轮 (MAG) 指令提供了任何两个轴之间按指定方向和指定比率进行电子齿轮传动。当被调用时，指定的从轴按照指定的比率（例如 1.345）或者按照从轴计数比主轴计数（例如 1:3）与主轴联动。MAG 指令支持以两种不同格式来指定齿轮速比：实数或分数，由比率格式输入选择来确定。从轴相对于主轴的方向是由一个非常灵活的方向输入参数来定义的。齿轮方向可以显性地设置为相同或相反，或者相对于当前齿轮方向设置为相反或不变。还要注意的，比率的值是带符号的。主轴参考选择可让齿轮输入从主轴的实际或命令位置中得到。当指令的离合器功能被激活时，在从轴锁定到主轴之前，齿轮指令命令从轴按照控制速度加速或减速并使用指令的加速度值，非常类似于汽车的离合器。

注意

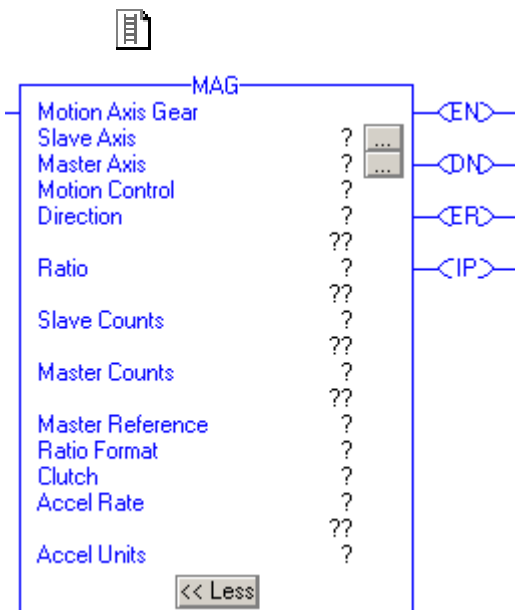


只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数：

梯形图

操作数	类型	格式	描述
从轴	AXIS_VIRTUAL	标签	要在其上执行操作的轴的名称。
	AXIS_GENERIC		
	AXIS_SERVO		
	AXIS_SERVO_DRIVE		
主轴	AXIS_FEEDBACK	标签	从轴跟随的轴。
	AXIS_CONSUMED		
	AXIS_VIRTUAL		
	AXIS_GENERIC		
	AXIS_SERVO		
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



操作数	类型	格式	描述
方向	UINT32	立即或标签	<p>从轴跟踪主轴的相对方向。</p> <p>选择下面一种：</p> <p>0 = 从轴按照与主轴相同的方向移动</p> <p>1 = 从轴按照与其当前方向相反的方向移动</p> <p>2 = 从轴从当前或以前的方向反向</p> <p>3 = 从轴继续其当前或以前的方向</p>
比率	REAL	立即或标签	带符号的实数值，建立每主用户单位的从用户单位齿轮速比。
从轴计数	UINT32	立即或标签	整数值，表示在指定分数齿轮速比中使用的从轴计数。
主轴计数	UINT32	立即或标签	整数值，表示在指定分数齿轮速比中使用的主轴计数。
主轴参考	布尔型	立即	<p>将主轴位置参考设置为命令位置或实际位置。</p> <p>0 = 实际 – 从轴运动将按照其编码器或其他反馈设备的测量的主轴的当前位置来生成。</p> <p>1 = 命令 – 从轴运动将从所需要的或命名的主轴位置来生成。</p>
比率格式	布尔型	立即	<p>所需的比率指定格式。选择任一：</p> <p>0 = 实数齿轮速比</p> <p>1 = 整数部分，为从轴编码器计数比主轴编码器计数</p>

操作数	类型	格式	描述
离合器	布尔型	立即	当离合器被使能时，运动控制按照指令定义的加速度值斜坡加速从轴达到齿轮速度。如果未使能，从轴将立即锁定到主轴。如果主轴当前正在移动，这种情况将导致从轴突然性非控制加速事件，这可能会造成轴出现故障。选择任一： 0 = 已使能 1 = 已禁止
加速度	布尔型	立即或标签	从轴的加速度，采用 % 或加速度单位。当离合器功能被使能时，它被应用。
加速度单位	DINT	立即	用于显示加速度值的单位。选择任一： 0 = 单位每秒 ² 1 = 最大加速度 %



MAG(SlaveAxis,MasterAxis, MotionControl,Direction, Ratio,SlaveCounts, MasterCounts, MasterReference,RatioFormat,Clutch, AccelRate, AccelUnits);

结构文本

操作数与梯形图 MAG 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
MasterReference	实际	0
	命令	1
RatioFormat	实数型	0
	fraction_slave_master_counts	1
离合器	已使能	0
	已禁止	1
AccelUnits	unitspersec2	0
	%ofmaximum	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴齿轮成功发起时, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	这个位在正的梯级转换时设置, 在被另一条电子齿轮命令所取代, 或者被停止命令、合并、关闭或伺服故障终止时清除。

说明: 电子齿轮 (MAG) 指令允许两个轴之间按指定比率进行电子齿轮传动。电子齿轮允许任何物理轴按照精确的比率同步到另一个物理轴的实际位置或命令位置。它在两个轴之间提供直接的边到边锁定 – 不使用最大速度、加速度或减速度限制。从轴的速度、加速度和减速度完全由主轴的运动以及指定的齿轮速比确定。

注意



在轴配置过程中所建立的最大速度、加速度或减速度限制不应用于电子齿轮。

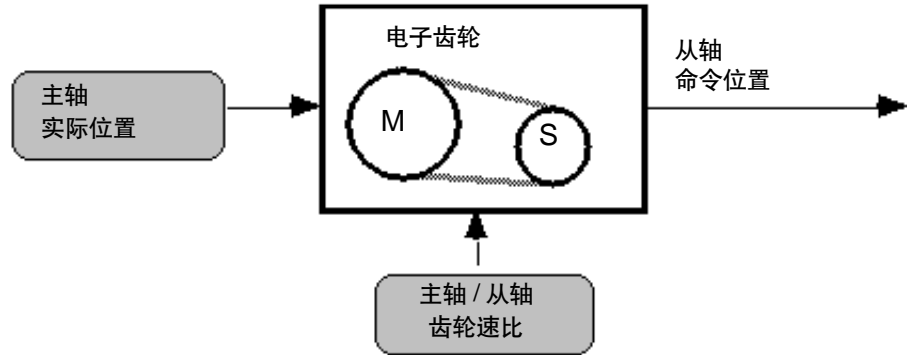
选择或输入所需的主轴、从轴和方向, 并为所需比率输入一个值或标签变量。如果轴被暗显 (灰掉) 或者未在从轴弹出菜单中显示, 则该物理轴没有定义为伺服操作。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于伺服操作。请使用标签编辑器来创建和配置新的轴。

电子齿轮在通过从轴后续的点动执行或位移过程后仍保持活跃。这样允许电子齿轮被点动或位移模式所叠加, 从而可创建复杂的运动和同步。

从动至实际位置

当实际位置被输入或选择为主轴参考源时，从轴运动是从主轴的实际位置生成的，如下所示。

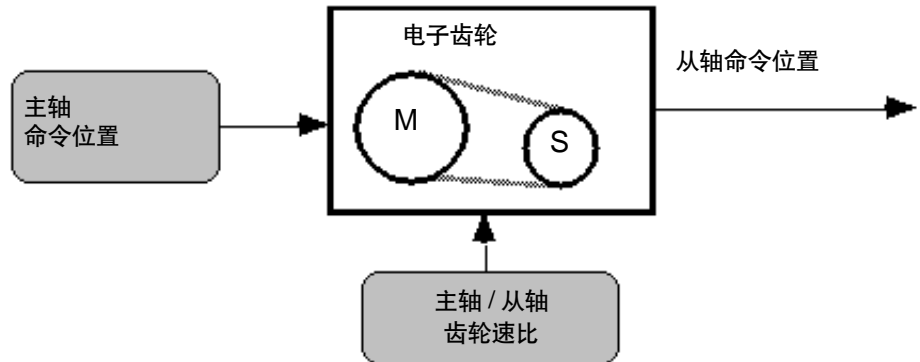


从动至实际位置

实际位置是由轴编码器测得的物理轴的当前位置。当主轴的轴类型被配置为反馈时，这是唯一的有效选择。

从动至命令位置

当命令位置被输入或选择为主轴参考源时，从轴运动是从主轴的命令位置生成的，如下所示。



从动至命令位置

命令位置（仅当主轴的轴类型被配置为伺服时才有效）是主轴当前期望的或命令的位置。

由于命令位置不包含任何相关的跟随错误、外部位置扰动或量化噪声，它对于齿轮来说是更加准确和稳定的参考。当齿轮连接至主轴的命令位置时，主轴必须被命令位移，从而在从轴上产生任何运动。请参阅运动轴对象技术参数，了解有关命令位置和实际位置轴参考的更多信息。

同方向齿轮

如果将方向选择或输入相同，当主轴在其*正向*方向移动时，从轴按其*正向*方向以指定的齿轮速比移动，反之亦然。

反方向齿轮

如果将方向选择或输入相反，当主轴在其*正向*方向移动时，从轴按其*负向*方向以指定的齿轮速比移动，反之亦然。

更改齿轮速比

如果将方向选择或输入不变，在保持当前齿轮方向（相同或相反）的同时可以更改齿轮速比。如果当前方向未知或者并不重要，这会非常有用。

反转齿轮方向

如果将方向选择或输入为反向，则电子齿轮的当前方向从相同更改为相反，或者从相反更改为相同。在缠绕应用中如果在每次缠绕结束时需要反转齿轮速比，这就非常有用。

实数齿轮速比

当比率格式被选择或输入为实数时，齿轮速比被指定为实数或者值在范围 0.00001 到 9.99999(含) 之间的标签变量，表示所需的从轴位置单位与主轴位置单位的比率。用这种方式表达的齿轮速比非常容易解读，因为它用轴所配置的位置单位定义的。

分数齿轮速比

当比率格式被选择或输入为分数时，齿轮速比被指定为一对整数或标签变量，表示从轴反馈计数与主轴反馈计数之间的比率。对从轴计数最多可使用五位数字 (99999)，对主轴计数最多可使用九位数字 (999999999)。请参阅本手册前面的标签变量生成器，了解有关标签变量的信息。

重要提示

当 MAG 指令的比率格式被指定为分数时，在轴配置步骤中所输入的转换常数**不会**使用。

如果您的齿轮速比无法准确地表示为小数点右边最多五位数字的实数，可使用分数作为比率格式。

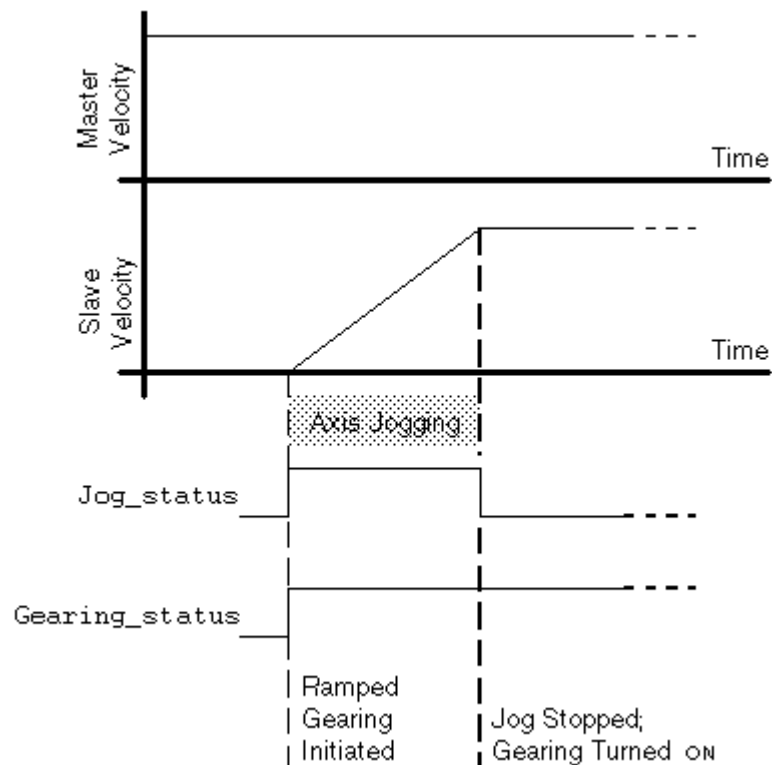
将齿轮速比指定为分数可直接实施无理数齿轮速比 (比如 $1/3$)，而不会有累积定位误差或舍入。由于主轴和从轴计数值不使用轴转换常数，而且因为它们都是整数，因此从轴和主轴之间的实际齿轮速比关系准确地符合指定的比率。

例如，无理数齿轮速比 $1/3$ 可等效指定为 1 次从轴计数比 3 次主轴计数、10 次从轴计数比 30 次主轴计数、3 次从轴计数比 9 次主轴计数，等等。

离合器

当离合器复选框被选中时，从轴将使用梯形速度模式（线性加速或减速）加速或减速，达到以指定的齿轮速比和方向传动到选定主轴当前所需的运动速度。从轴一旦到达齿轮速度，电子齿轮根据其他选择自动激活。输入所需的加速度作为当前配置的最大加速度值的百分比，或者直接使用为加速度配置的用户单位。

这种离合器功能的工作原理非常类似于汽车中的离合器，可从轴平稳地与主轴啮合，如下所示。



离合器功能

当主轴正在运动时如果电子齿轮被使能，使用离合器功能可避免所造成的非控制加速或减速。离合器功能还可以用于在过程中合并齿轮速比变更，甚至是方向改变。运动控制器自动将从轴斜坡加速至按照新的比率和 / 或方向主轴所暗指的速度。

离合器斜坡发生器的操作对从轴上可能正在进行的点动或位移过程没有影响。

更改主轴

电子齿轮的主轴可随时更改，甚至是齿轮当前已经被使能的情况下。但是，由于同时可以在多个轴上使能电子齿轮，如果伺服主轴和从轴被颠倒，轴可能会被交叉耦合，造成意外的运动。

例如，如果您正在将轴 0 啮合到轴 1（定义为伺服轴）上，然后又希望更改为轴 1 啮合到轴 0 上，则必须先禁止轴 0 上的齿轮（请参阅本节后面的禁止齿轮）。这是因为指定轴 1 为从轴、轴 0 为主轴，不会自动禁止轴 0 成为从轴同时轴 1 成为主轴。

齿轮兼位移

在电子齿轮被使能时，增量 MAM 指令可用于从轴（或者主轴，如果轴类型被配置为伺服）。这特别可用于实现相位提前 / 延迟控制。增量位移距离可用于消息主轴与从轴之间的任何相位误差，或者用于创建准确的非零相位关系。增量 MAM 指令还可以结合电子齿轮来补偿物料滑移。

通常相位调整中使用齿轮速比 1。在执行位移来进行纠正之前，1:1 的比率可保证计算出的相位误差不会改变。电子齿轮通常不用于绝对位移，因为最终的结束点无法预测。

要成功执行电子齿轮指令，目标轴必须配置为伺服轴类型并且处于伺服启动状态。如果不满足任何条件，则指令出错。

重要提示

MAG 指令执行在一次扫描中完成，因此完成 (.DN) 位和正在进行 (.IP) 位被立即设置，过程完成 (.PC) 位被立即清除。正在进行 (.IP) 位保持设置状态，直到发起的齿轮过程被另一条 MAG 指令取代，或者被运动轴停止命令、合并操作或伺服故障操作终止为止。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#)上）。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。

轴未配置 (11) 错误代码的扩展错误代码如下：

- 扩展错误代码 1 表示从轴未配置。
- 扩展错误代码 2 表示主轴未配置。

参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MAG 指令，扩展错误代码 4 指的是比率操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

对于错误代码 54—最大减速度值为零，如果扩展错误代码返回一个正数 (0-n)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的轴最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

状态位: MAG 对状态位的更改

如果离合器复选框未选中，则执行 MAG 指令只是将齿轮状态位设置为真。

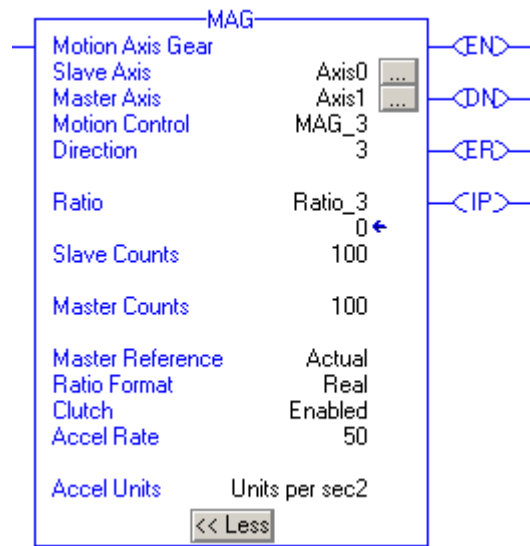
位名称	状态	含义
GearingStatus	真	轴正在齿轮运动中

如果离合器复选框被选中，则当离合过程完成后，执行 MAG 指令将把齿轮锁定状态位设置为真。

位名称	状态	含义
齿轮锁定状态	真	轴已经完成离合过程并锁定。
GearingStatus	真	轴正在齿轮运动中

示例: 当输入条件为真时，控制器在 *轴 2* 和 *轴 1* 之间提供电子齿轮。

梯形图



MAG 梯形图示例

结构文本

```
MAG(Axis0,Axis1,MAG_3,3,Ratio_3,0,100,100,Actual,Real,
Enabled,50,Unitspersec2);
```

运动动态调整 (MCD)

使用 MCD 指令可选择性地更改正在进行的位移模式或点动模式的速度、加速度或减速度。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数: 梯形图



MCD		
Motion Change Dynamics		(EN)
Axis	? ...	
Motion Control	?	(DN)
Motion Type	?	
Change Speed	?	(ER)
Speed	??	
Change Accel	?	
Accel Rate	?	
	??	
Change Decel	?	
Decel Rate	?	
	??	
Change Accel Jerk	?	
Accel Jerk	?	
	??	
Change Decel Jerk	?	
Decel Jerk	?	
	??	
Speed Units	?	
Accel Units	?	
Decel Units	?	
Jerk Units	?	
<< Less		

操作数	类型	格式	描述
轴	AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_ 指令	标签	用于访问指令状态参数的结构。
运动类型	UDINT	立即	要更改的模式（点动或位移）。选择任一： 0 = 点动 1 = 位移
变更速度	布尔型	立即	设置以使能速度变更。选择任一： 0 = 否 1 = 是
速度	REAL	立即或标签	移动轴的新速度，采用 % 或速度单位。
变更加速度	布尔型	立即	设置以使能加速度变更。选择任一： 0 = 否 1 = 是
加速度	REAL	立即或标签	轴的加速度，采用 % 或加速度单位。

操作数	类型	格式	描述
变更减速度	布尔型	立即	设置以能使减速度变更。选择任一： 0 = 否 1 = 是
减速度	REAL	立即或标签	轴的减速度，采用 % 或减速度单位。 如果运动正在进行之中而您降低减速度，则轴可能会过冲其目标位置。
变更加速度跃度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
加速跃度	SINT、INT、DINT 或 REAL	立即或标签	您 必须 始终为加速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 <ul style="list-style-type: none"> 加速跃度是轴的加速跃度值。 使用这个值起步。 加速度跃度 = 100 (时间 %) 跃度单位 = 2
变更减速度跃度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
减速跃度	SINT、INT、DINT 或 REAL	立即或标签	您 必须 始终为减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 <ul style="list-style-type: none"> 减速跃度是坐标系的减速跃度值。 使用这些值起步。 减速度跃度 = 100 (时间 %) 跃度单位 = 2
SpeedUnits	布尔型	立即	用于显示速度值的单位。选择任一： 0 = 单位每秒 1 = 最大速度 %

操作数	类型	格式	描述
加速度 单位	布尔型	立即	用于显示加速度值的单位。 选择任一： 0 = 单位每秒 ² 1 = 最大加速度 %
减速度 单位	布尔型	立即	用于显示减速度值的单位。 选择任一： 0 = 单位每秒 ² 1 = 最大减速度 %
跃度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 ³ 1 = 最大值 % 2 = 时间 %

结构文本

操作数与梯形图 MCD 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：



MCD(Axis, MotionControl, MotionType, ChangeSpeed, Speed, ChangeAccel, AccelRate, ChangeAccelJerk, AccelJerk, ChangeDecelJerk, ChangeDecel, DecelRate, SpeedUnits, AccelUnits, DecelUnits, JerkUnits);

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为
轴	无枚举	标签
MotionControl	无枚举	标签
MotionType	jog move	0 1
ChangeSpeed	无 是	0 1
速度	无枚举	立即或标签
ChangeAccel	无 是	0 1
AccelRate	无枚举	立即或标签
ChangeDecel	无 是	0 1
减速度	无枚举	立即或标签

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为
ChangeAccelJerk	无枚举	0 = 否 1 = 是
AccelJerk	无枚举	立即或标签 您 必须 始终为加速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这个值起步。 加速度跃度 = 100
ChangeDecelJerk	无枚举	0 = 否 1 = 是
DecelJerk	无枚举	立即或标签 您 必须 始终为减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这个值起步。 减速度跃度 = 100
SpeedUnits	unitspersec	0
	%ofmaximum	1
AccelUnits	unitspersec ²	0
	%ofmaximum	1
DecelUnits	unitspersec ²	0
	%ofmaximum	1
JerkUnits	Unitspersec ³	0
	%ofmaximum	1
	%oftime	2

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴动态变更成功发起时, 完成位被设置。该指令执行在一次扫描中完成, 因此完成 (.DN) 位被立即设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。

说明: MCD 指令可在进行中更改梯形模式位移的速度, 更改正在进行的梯形模式点动的速度、加速度和减速度。选择所需的物理轴和运动类型, 输入速度、加速度和减速度的值或标签变量。速度、加速度和减速度值可输入为当前最大配置值的百分比, 或者直接采用为轴配置的速度或减速度单位。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于伺服操作。请使用标签编辑器来创建和配置新的轴。

注意

**速度和 / 或终点位置过冲风险**

如果您通过任何方法动态地更改运动参数, 即通过运动动态指令 (MCD 或 MCCD) 进行更改, 或者在最后一个指令完成之前开始一条新的指令, 要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时, 如果最大减速度降低, 则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲:

- 当运动正在减速或者接近减速点时最大减速度降低; 或者
- 最大加速度跃度降低而轴正在加速。但是请牢记, 如果用时间 % 的单位指定, 跃度可能会间接被改变。

有关更多信息, 请参见 [Troubleshoot Axis Motion](#) 上的 [page 9](#)。

这是一条转换指令:

- 在梯形图中, 当每次应执行指令时, 将梯级输入条件从清除切换到设置。
- 在结构文本中, 限制指令只对转换执行。请参阅附录 C。

更改位移动态参数

但选择或输入的运动类型为位移时，正在进行的位移的速度、加速度和 / 或减速度可以更改为指定的值。速度更改在指定的加速度处进行（如果新速度高于当前速度）或者在指定的减速度处进行（如果新速度低于当前速度）。

暂停位移

MCD 指令可通过将速度更改为零，用于临时暂停正在进行的运动。使用另一条 MCD 指令并采用非零速度值即可按最初指定的方式完成位移。

更改点动动态参数

但选择或输入的运动类型为点动时，正在进行的点动的速度、加速度和 / 或减速度可以更改为指定的值。速度更改在指定的加速度处进行（如果新速度高于当前速度）或者在指定的减速度处进行（如果新速度低于当前速度）。

算术状态标志： 不受影响

故障条件： 无

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#)上）。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。

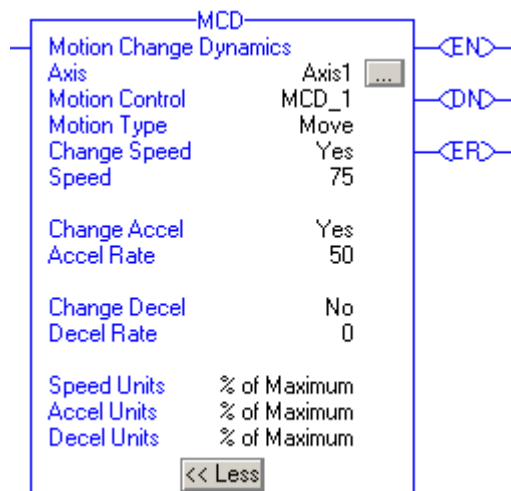
参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MCD 指令，扩展错误代码 4 指的是速度操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

对于错误代码 54 – 最大减速度值为零，如果扩展错误代码返回一个正数 (0-*n*)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

MCD 对状态位的更改: 无

示例: 当输入条件为真时，控制器更改轴 1 正在进行的位移模式或点动模式的速度、加速度或减速度。

梯形图



MCD 梯形图示例

结构文本

```
MCD(Axis1,MCD_1,Move,Yes,75,Yes,50,No,0,%ofmaximum,%ofmaximum,%ofmaximum);
```

模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

运动位置重设 (MRP)

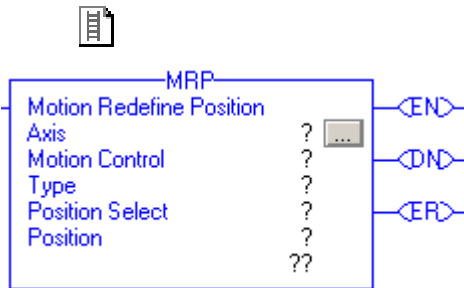
使用 MRP 指令可更改轴的命令位置或实际位置。由位置所指定的值用于更新轴的实际位置或命令位置。位置的重新计算可按绝对值或相对值的方式进行。如果选择绝对值，则位置的值被赋值到当前的实际位置或命令位置。如果选择相对值，则位置的值作为位移被加到当前的实际位置或命令位置上。重新定义当前轴位置的过程对于正在进行的运动没有影响，因为指令保留重新定义过程中的当前伺服跟随错误。因此，轴位置可在过程中进行重新定义，而不会干扰轴的运动。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤亡。

操作数：梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令参数的结构。
类型	布尔型	立即	您希望如何进行重新定义的方式。选择任一： 0 = 绝对值 1 = 相对值
位置选择	布尔型	立即	选择要执行重新定义操作的位置。选择任一： 0 = 实际位置 1 = 命令位置
位置	REAL	立即或标签	用于更改轴位置或当前位置偏移的值。



MRP(Axis,MotionControl,Type,PositionSelect,Position);

结构文本

操作数与梯形图 MRP 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
类型	absolute	0
	relative	1
PositionSelect	实际	0
	命令	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴的位置操作成功重新定义时，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。

说明： 运动位置重设 (MRP) 指令直接将指定的轴的命令位置或实际位置设置为指定的绝对或相对位置。这个指令不产生运动 – 只是当前轴位置被重新定义。选择或输入所需的轴、类型、位置选择，并为所需新位置输入一个值或标签变量。

如果目标轴没有出现在可用轴列表中，则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

当轴正在移动或者静止时，MRP 指令都可以使用。MRP 用于“在进行中”重新定义位置，可用于某些记录、滑差补偿和重新校准应用。

绝对值模式

当为 MRP 类型选择或输入绝对值时，新位置指定的就是轴的绝对位置。不会产生运动 – 只是当前的轴位置（实际的或命令的）被重新定义为指定的新位置。

如果使用了软件超行程限制（请参阅运动轴对象技术参数。了解软件超行程配置的有关信息），新位置必须处于最大正向和最大负向行程的配置值之间。否则，当指令被执行时，会生成一个软件超行程故障。

注意



如果软件超行程限制检查正在生效，以绝对值模式执行 MRP 可能在绝对值意义上废除当前的最大正向和最大负向行程限制。在对具有行程限制的轴进行绝对位置重新定义时，一定要小心。

当轴不在运动时，绝对值和相对值模式下的 MRP 指令具有相同的结果。但是当轴在运动时，在执行 MRP 指令和负值新位置所花的时间过程中，绝对值模式引入一个位置误差，等于轴在这个期间的运动。相对模式不会引入这种误差，保证与轴速度或位置无关的准确校正。

相对值模式

当为 MRP 类型选择或输入相对值时，新位置的值用于 *偏移* 轴的当前位置。不会产生运动 – 只是当前的轴位置（实际的或命令的）被重新定义为当前位置 *加上* 指定的新位置。

在相对值模式下，轴位置的重新定义使轴在运动中也不会引入位置误差。它特别适用于程序控制下的退绕轴位置，而不是使用内置的旋转轴功能。

当轴不在运动时，绝对值和相对值模式下的 MRP 指令具有相同的结果。但是当轴在运动时，在执行 MRP 指令和负值新位置所花的时间过程中，绝对值模式引入一个位置误差，等于轴在这个期间的运动。相对模式不会引入这种误差，保证与轴速度或位置无关的准确校正。

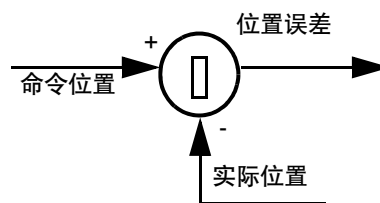
实际位置

当为 MRP 位置选择输入或选择实际值时，新位置直接应用到物理轴的实际位置。轴的命令位置也使用新的实际位置进行调整，以保留任何存在的位置误差。这样可保证当位置被重新定义后，不会出现意外的轴运动。请参阅运动轴对象技术参数，了解关于命令位置、实际位置和位置误差的更多讨论。

命令位置

当为 MRP 位置选择输入或选择命令值时，新位置直接应用到伺服轴或虚轴的命令位置。由于仅反馈轴没有命令位置，请从位置菜单中始终为仅主轴类的轴选择实际值。伺服轴的实际位置也使用新的命令位置进行调整，以保留任何存在的位置误差。这样可保证当位置被重新定义后，不会出现意外的轴运动。

命令位置是伺服的期望或命令位置，由任何以前的运动控制指令生成。实际位置是由轴编码器或其他反馈装置测得的物理轴或虚轴的当前位置。位置误差是这两者之间的差异，用于驱动电机来使实际位置等于命令位置。下图显示了这三个位置之间的关系。



位置关系

请参阅运动轴对象技术参数，详细了解 ControlLogix 运动控制器所使用的嵌套数字伺服环的概要信息。

要成功执行 MRP 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示

MRP 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输多条消息。因此，完成 (.DN) 位并不立即设置，而只是在这些消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件: 无

错误代码: 参见[运动控制指令错误代码 \(ERR\)](#) (在[页码 391](#) 上)。

扩展错误代码: 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MRP 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

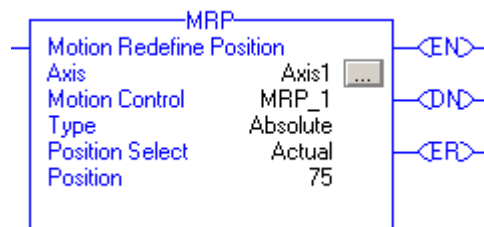
相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)	重新定义位置、归零和记录 2 是相互排斥的。(SERCOS)。

参数超出范围 (13) 错误的扩展错误代码工作方式有点不同。不是采用标准的枚举，扩展错误代码所显示的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MRP 指令，扩展错误代码 4 指的是位置操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

MRP 对状态位的更改: 无

示例: 当输入条件为真时，控制器更改轴 1 的位置。

梯形图



MRP 梯形图示例

结构文本

```
MRP(Axis1,MRP_1,Absolute,Actual,75);
```

计算凸轮运动曲线 (MCCP)

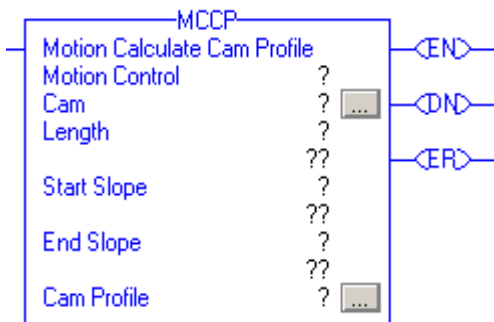
计算凸轮运动曲线 (MCCP) 基于一系列凸轮点来计算凸轮曲线。凸轮点数组可通过程序或者使用 RSLogix 5000 凸轮曲线编辑器来建立。凸轮数组中的每个凸轮点包括从位置值、主轴位置（位置凸轮）或时间（时间凸轮）值以及内插类型（线性或三次方）。所产生的凸轮曲线可供运动轴位置凸轮 (MAPC) 或者运动轴时间凸轮 (MATC) 指令使用，来按照主轴位置或时间来控制从轴的运动。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数：梯形图



操作数	类型	格式	描述
运动控制	MOTION_指令	标签	用于访问块状态参数的结构。
凸轮	CAM	数组	用于计算凸轮曲线的凸轮数组标签名称。数字数组索引表示在凸轮曲线计算中使用的数组中的开始凸轮元素。省略号可启动凸轮曲线编辑器。
长度	UINT	立即或标签	确定在凸轮曲线计算中使用的数组中的凸轮元素的个数。
起始斜率	REAL	立即或标签	这是曲线初始斜率的边界条件。它仅对于三次方的第一段有效，用于指定通过第一个点的斜率。
终止斜率	REAL	立即或标签	这是曲线终止斜率的边界条件。它仅对于三次方的最后一段有效，用于指定通过最后一个点的斜率。
凸轮曲线	CAM_PROFILE	数组	计算的凸轮曲线数组的标签名称，用作 MAPC 和 MATC 指令的输入。对于凸轮曲线数组，仅允许零数组元素 ([0])。省略号可启动凸轮曲线编辑器。



MCCP(MotionControl,Cam,Length,StartSlope,EndSlope,CamProfile);

结构文本

操作数与梯形图 MCCP 指令的操作数相同。对于数组操作数，您不需要包含数组索引。如果您不包含数组索引，则指令从数组 ([0]) 中的第一个元素开始。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时, 启用位被设置, 直到完成位被设置并且梯级变成假之前将保持设置状态。
.DN (完成) 位 29	当计算凸轮指令被成功执行并且计算出凸轮曲线数字以后, 完成位被设置。
.ER (错误) 位 28	错误位指示指令检测到错误, 例如凸轮数组的长度非法。

说明: 计算凸轮运动曲线 (MCCP) 基于指定凸轮数组内给定的点集来计算凸轮曲线。这个指令所产生的凸轮曲线可被后续的 MAPC 或 MATC 凸轮指令使用, 为从轴提供与主轴位置或主轴时间相关的复杂运动。

由于凸轮曲线可直接由 RSLogix 5000 凸轮曲线编辑器计算出来, MCCP 指令的主要目的是提供一种方法来基于对相应凸轮数组的程序变化实时地计算凸轮曲线。

指定凸轮数组

为了执行 MCCP 指令, 必须使用 RSLogix 标签编辑器或凸轮曲线编辑器来创建凸轮数组标签。下图显示如何建立凸轮数组标签并用作 MCCP 指令的输入。

凸轮数组元素包含从 (yp) 和主 (xp) 点对, 还有一个内插类型。由于与特定的轴位置或时间没有关联, x 和 y 点值没有单位。内插类型可为每个点指定为“线性”或“三次方”。

指定凸轮曲线标签

要执行 MAPC 指令, 还必须创建一个凸轮曲线数组标签。凸轮曲线数组标签可以通过 RSLogix 5000 标签编辑器来创建, 或者通过 MAPC/MATC 指令使用内置的凸轮曲线编辑器来创建。

凸轮曲线数组内部的数据可在编译时使用凸轮曲线编辑器进行修改, 或者在运行时使用计算凸轮运动曲线 (MCCP) 指令进行修改。在运动时更改, 必须创建一个凸轮数组才能使用 MCCP 指令。

状态参数用于指示凸轮曲线数组元素已经计算出来。如果凸轮指令的执行尝试使用凸轮曲线中任何未计算的元素, 则 MAPC 或 MATC 指令就会出错。类型参数决定了在这个凸轮数组元素与下一个凸轮元素之间应用的内插类型。

凸轮曲线数组状态成员

凸轮曲线数组中第一个元素的状态成员比较特别，用于数据完整性检查。因此，MCCP 指定凸轮曲线时必须使用将开始索引设置为 0。这第一个凸轮曲线元素状态成员可以取以下值：

状态变量	描述
0	凸轮曲线元素尚未计算出来
1	凸轮曲线元素正在进行计算
2	凸轮曲线元素已经计算出来
n	凸轮曲线元素已经计算出来，目前正被 (n-2) 条 MAPC 或 MATC 指令使用。

线性和三次方样条内插

结果计算出的凸轮曲线是完全通过内插得出的。这意味着如果当前的主轴位置或时间并不准确地对应于用于生成凸轮曲线的凸轮数组内的点，将在相邻的点之间使用线性和三次方内插来确定从轴的位置。这样，即提供最大可能的平滑从轴运动。MCCP 指令通过计算多项式方程的系数，确定从轴作为主轴位置或时间的函数，从而实现这一点。

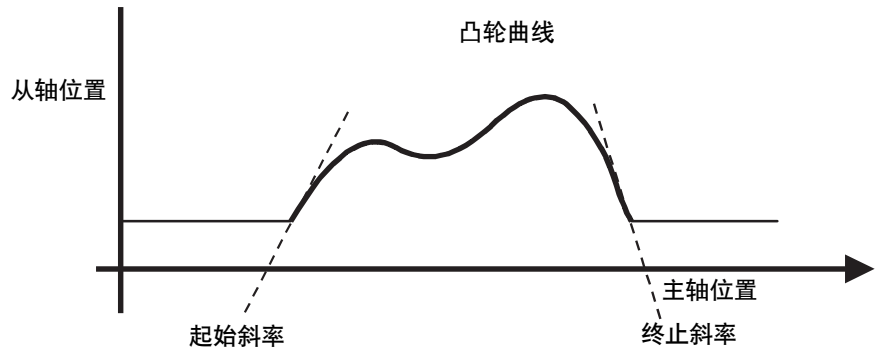
计算凸轮曲线

在指定的轴上计算凸轮曲线之前，MCCP 指令首先通过检查凸轮曲线元素的状态成员的值，检查凸轮曲线数组是否已经计算出来。如果状态值为 0 或 2，则 MCCP 继续进行凸轮曲线的计算。当凸轮曲线数组被完全计算出来后，MCCP 指令将第一个凸轮曲线元素的状态值设置为“正在进行计算”或者 1，然后将所有其他凸轮曲线元素的状态值设置为“正在进行计算”。在计算进行过程中，各个凸轮曲线成员的状态值被设置为“已经计算”或 2。当凸轮曲线数组内的所有元素都已计算出来后，第一个凸轮曲线元素的状态值被设置为“已经计算”。

但是，如果 MCCP 指令被执行时初始的凸轮曲线状态值为 1，则凸轮曲线正在被另一条 MCCP 指令计算，MCCP 指令出错。如果状态值为 >2，则凸轮曲线正在有效地被 MAPC 或 MATC 指令过程使用，MCCP 指令出错。

起始斜率和终止斜率

为了方便平滑地进入或离开三次方凸轮曲线，提供了斜率控制。起始斜率和终止斜率参数确定了从轴相对于主轴的初始变化速率。这些值用于在凸轮数组上执行的三次方样条计算中。下图显示了主从斜率关系。



起始斜率和终止斜率

起始斜率和终止斜率的默认值为 0，以方便从停止状态平滑地开始和结束凸轮曲线。但是，如果轴已经在做凸轮运动，则可以指定一个相应的非零起始斜率以匹配当前正在执行的凸轮的终止斜率，从而将两个凸轮曲线无缝地融合在一起。

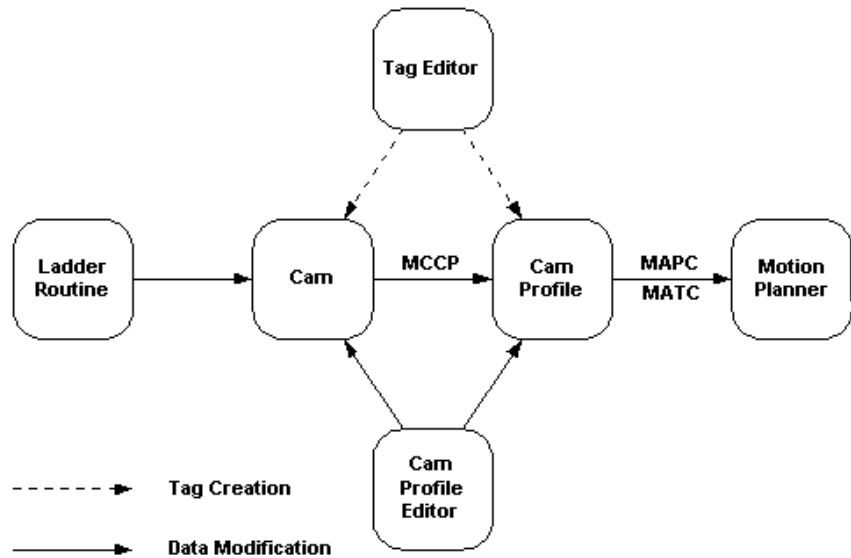
起始斜率和终止斜率不适用于采用线性内插时的凸轮曲线的开始和结束。

重要提示

MCCP 指令执行在一次扫描中完成。因此，这条指令应放置在单独的任务中，以避免影响用户程序的扫描周期。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。



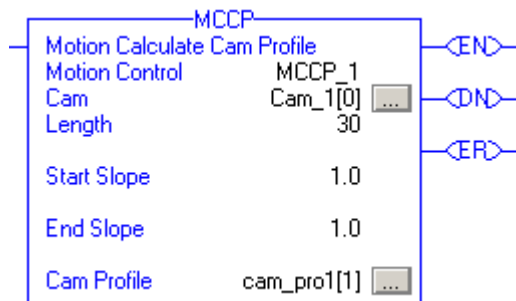
凸轮操作图

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#)上）。

扩展错误代码： 很多错误代码对于帮助定位问题不够具体，而扩展错误代码则提供更多的指令特定的信息。当 MCCP 指令收到非法凸轮长度 (26) 错误消息，让它知道长度输入参数并不对应于指令所期望的值时，相应的扩展错误代码提供了该凸轮在向指令提供的凸轮标签中的编号。当 MCCP 指令收到非法凸轮长度 (27) 错误消息，让它知道长度输入参数并不对应于指令所期望的值时，相应的扩展错误代码提供了指令正在尝试生成的凸轮点的个数。

MCCP 对状态位的更改： 无

示例： *梯形图*



MCCP 梯形图示例

结构文本

```
MCCP(MCCP_1,Cam_1[0],30,1.0,1.0,cam_pro1[1]);
```

运动轴位置凸轮 (MAPC)

运动轴位置凸轮 (MAPC) 指令按照指定的凸轮曲线在任何两个轴之间提供电子凸轮。

当执行时，指定的从轴与指定的主轴使用由 RSLogix 5000 凸轮曲线编辑器或者通过以前执行的计算凸轮运动曲线 (MCCP) 指令所建立的位置凸轮曲线来同步。从轴相对于主轴的方向是由一个灵活的方向输入参数来定义的。应用到从轴上的凸轮方向可以显性地设置为相同或相反，或者相对于当前凸轮方向设置为相反或不变。

为了将从轴位置准确地与主轴位置同步，可为主轴提供一个执行计划设置和相关的主轴锁定位置。当主轴行程按照由执行计划参数指定的方向通过主轴锁定位置时，从轴将按照从凸轮锁定位置开始的指定凸轮曲线锁定到主轴位置。

凸轮曲线还可以通过执行计划参数进行配置，以立即或挂起执行当前正在执行的位置凸轮曲线的完成。凸轮曲线也可以执行一次、连续执行或通过指定所需的执行模式来执行。

主轴参考选择可让来自主轴的凸轮输入从主轴的实际或位置命令位置中得到。为支持要求单向运动的应用，提供可一种“滑动离合器”功能，防止当主轴反转方向时从轴“倒车”。这种功能是由主轴方向参数控制的。

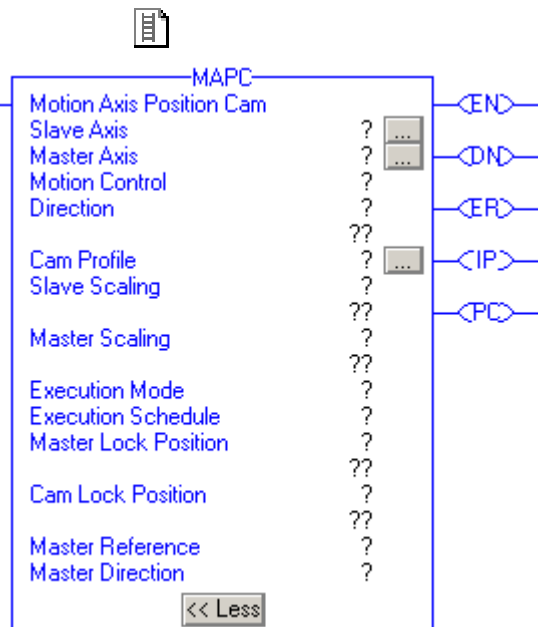
主轴和从轴标定功能可用于按照标准凸轮曲线来设定从轴运动的标定。而不用再创建新的凸轮表和计算新的凸轮曲线。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数：梯形图



操作数	类型	格式	描述
从轴	AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	凸轮曲线所应用到的轴的名称。省略号将启动轴属性对话框。
主轴	AXIS_FEEDBACK AXIS_CONSUMED AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	从轴按照凸轮曲线所跟随的轴。省略号将启动轴属性对话框。如果执行计划中选择了挂起，则主轴被忽略。
运动控制	MOTION_指令	标签	用于访问块状态参数的结构。
方向	UINT32	立即或标签	从轴相对于主轴的方向： <ul style="list-style-type: none"> • 相同 – 从轴位置值与主轴位置具有相同方向。 • 相反 – 从轴位置值与主轴位置具有相反方向。 或者相对于当前或以前的凸轮方向： <ul style="list-style-type: none"> • 反转 – 位置凸轮的当前或以前方向在执行时反转。选择反转后在第一次执行时，控制器将方向默认为相反。 • 不变 – 允许其他凸轮参数被更改，而不改变当前或以前的凸轮方向。选择不不变后在第一次执行时，控制器将方向默认为相同。

操作数	类型	格式	描述
凸轮曲线	CAM_PROFILE	数组	用于建立主 / 从位置关系的计算出的凸轮曲线数组的标签名称。对于凸轮曲线数组，仅允许零数组元素 ([0])。省略号可启动凸轮曲线编辑器。
从轴标定	REAL	立即或标签	设定从轴通过凸轮曲线所覆盖的总距离的标定。
主轴标定	REAL	立即或标签	设定主轴通过凸轮曲线所覆盖的总距离的标定。
执行模式	UINT32	立即	<p>确定凸轮曲线为执行一次还是重复执行：</p> <p>0 = 一次 – 仅在主轴移动到由凸轮曲线开始点和结束点定义的范围内时，从轴的凸轮运动才会开始。当主轴移动超出所定义的范围时，从轴上的运动停止，过程完成位被设置。如果主轴从新回到凸轮曲线范围内，从轴运动不会恢复。</p> <p>1 = 连续 – 一旦开始后，凸轮曲线将无限地执行下去。在需要凸轮位置以旋转或往复方式连续运动的旋转应用中，这种功能比较常用。</p> <p>2 = 持续 – 当主轴移动超出所定义的范围时，从轴上的运动停止，PositionCamLockStatus 位被清除。如果主轴从新回到凸轮曲线范围内，PositionCamLockStatus 位被设置，从轴运动将会恢复。</p>

操作数	类型	格式	描述
执行计划	UINT32	立即	<p>选择执行凸轮曲线时所用的方法。选项如下：</p> <p>0 = 立即 – 从轴立即锁定到主轴，位置凸轮过程开始。</p> <p>1 = 挂起 – 允许您在一个正在进行的位置凸轮完成后混合一个新的位置凸轮。当选择挂起时，以下参数将被忽略：主轴、主轴锁定位置和主轴参考。</p> <p>2 = 仅正向 – 当主轴位置从正的方向穿过主轴锁定位置时，凸轮曲线开始。</p> <p>3 = 仅反向 – 当主轴位置从反的方向穿过主轴锁定位置时，凸轮曲线开始。</p> <p>4 = 双向 – 当主轴位置从任一方向穿过主轴锁定位置时，凸轮曲线开始。</p>
主轴锁定位置	REAL	立即或标签	<p>从轴锁定主轴时主轴的绝对位置。如果执行计划值中选择了挂起，则主轴锁定位置被忽略。</p>

操作数	类型	格式	描述
凸轮锁定位置	REAL	立即或标签	这确定了凸轮曲线的开始位置。
主轴参考	UINT32	立即	<p>将主轴位置参考设置为命令位置或实际位置。如果执行计划值中选择了挂起，则主轴参考位置被忽略。</p> <p>0 = 实际 – 从轴运动将按照其编码器或其他反馈设备的测量的主轴的当前位置来生成。</p> <p>1 = 命令 – 从轴运动将从所需要的或命名的主轴位置来生成。</p>
主轴方向	UINT32	立即	<p>这按照凸轮曲线确定生成从轴运动的主轴的方向。</p> <p>选项如下：</p> <p>0 = 双向 – 从轴可在任一方向跟踪主轴。</p> <p>1 = 仅正向 – 从轴在主轴正的方向跟踪主轴。</p> <p>2 = 仅反向 – 从轴在主轴反的方向跟踪主轴。</p>



MAPC(SlaveAxis,MasterAxis, MotionControl,Direction, CamProfile,SlaveScaling, MasterScaling,ExecutionMode, ExecutionSchedule, MasterLockPosition, CamLockPosition, MasterReference, MasterDirection);

结构文本

操作数与梯形图 MAPC 指令的操作数相同。对于数组操作数，您不需要包含数组索引。如果您不包含数组索引，则指令从数组 ([0]) 中的第一个元素开始。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
ExecutionMode	一次	0
	连续	1
	持续	2
ExecutionSchedule	立即	0
	pending	1
	forwardonly	2
	reverseonly	3
	bidirectional	4
MasterReference	实际	0
	命令	1
MasterDirection	bidirectional	0
	forwardonly	1
	reverseonly	2

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到梯级变为假为止。
.DN (完成) 位 29	当轴位置凸轮成功发起时，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	这个位在正的梯级转换时设置，在被另一条运动轴位置凸轮命令所取代，或者被停止命令、合并、关闭或伺服故障终止时清除。
.PC(过程完成) 位 27	过程完整位在正的梯级过渡时被清除，在“一次”执行模式下当主轴位置离开由当前活跃的凸轮曲线所定义的主轴位置范围时，这个位被设置。

说明： 运动轴位置凸轮 (MAPC) 指令执行由前一个计算凸轮运动曲线 (MCCP) 指令，或者由 RSLogix 5000 凸轮曲线编辑器所建立的位置凸轮曲线。实际上，位置凸轮提供了一种在两个轴之间实现非线性的“电子齿轮”的功能。不使用最大速度、加速度或减速度限制。从轴的速度、加速度和减速度完全由主轴的运动以及从相关的凸轮表中得到的指定凸轮曲线来确定。

注意



在轴配置过程中所建立的最大速度、加速度或减速度限制不应用于电子凸轮。

凸轮方向

凸轮可以配置为向从轴的命令位置加减其增量成分。对这种行为的控制是通过方向参数来实现的。

同方向凸轮

当为 MAPC 指令的方向选择或输入相同时，从凸轮曲线中计算出的从轴位置值将*添加到*从轴的命令位置中。这是最常用的操作，因为曲线位置值的使用就像在原始凸轮表内输入一样。也就是说，连续增加曲线值会导致轴在*正*的方向上运动，反之亦然。

反方向凸轮

当为 MAPC 指令的方向选择或输入相反时，从凸轮曲线中计算出的从轴位置值将从从轴的命令位置中*减去*。因此，轴的运动与原始凸轮表内隐含的方向*相反*。也就是说，连续增加曲线值会导致轴在*负*的方向上运动，反之亦然。

保留当前凸轮方向

当为方向选择或输入不变时，其他位置凸轮参数可以更改，但保留当前或以前的凸轮方向（相同或相反）。如果当前方向未知或者并不重要，这会非常有用。选择不变更后在第一次凸轮执行时，控制器将方向默认为相同。

反转当前凸轮方向

如果选择反转时，位置凸轮的当前或以前的方向从相同更改为相反，或者从相反更改为相同。选择反转后在第一次凸轮执行时，控制器将方向默认为相反。

指定凸轮曲线

要执行 MAPC 指令，必须指定一个计算的凸轮曲线数组标签。凸轮曲线数组标签可以通过 RSLogix 5000 标签编辑器来创建，或者通过 MAPC 指令使用内置的凸轮曲线编辑器来创建，或者通过在现有的凸轮数组上执行计算凸轮运动曲线 (MCCP) 指令来创建。

凸轮曲线数组内部的数据可在编译时使用凸轮曲线编辑器进行修改，或者在运行时使用计算凸轮运动曲线 (MCCP) 指令进行修改。在运动时更改，必须创建一个凸轮数组才能使用 MCCP 指令。请参阅 MCCP 指令指导规范，了解有关转换凸轮数组的更多信息。

这个凸轮曲线数组结构元素除了状态元素以外都被“隐藏”而无法从 RSLogix 5000 标签编辑器进行编辑。这些元素对于用户没有什么用途。状态成员用于指示相应的凸轮曲线数组元素已经计算出来。如果凸轮指令的执行尝试使用凸轮曲线中任何未计算的元素，则指令就会出错。类型参数决定了在这个凸轮数组元素与下一个凸轮元素之间应用的内插类型（例如线性或三次方）。

凸轮曲线数组检查

凸轮曲线数组中第一个元素的状态成员比较特别，用于数据完整性检查。因此，MAPC 指定凸轮曲线时必须使用将开始索引设置为 0。

这第一个凸轮曲线元素状态成员可以取以下值。

状态值	描述
0	凸轮曲线元素尚未计算出来
1	凸轮曲线元素正在进行计算
2	凸轮曲线元素已经计算出来
n	凸轮曲线元素已经计算出来，目前正被 (n-2) 条 MAPC 或 MATC 指令使用。

在指定的轴上开始凸轮之前，MAPC 指令首先通过检查凸轮曲线元素的状态成员的值，检查凸轮曲线数组是否已经计算出来。如果状态为 0 或 1，则凸轮曲线尚未计算出来，MAPC 指令出错。如果凸轮曲线数组已经完全计算出来 (状态 >1)，则指令递增状态数字，表示已经被这个轴使用。

当凸轮完成或终止后，第一个凸轮曲线数组元素递减，以保持跟踪有效使用相关凸轮曲线的凸轮个数。

线性和三次方内插

位置凸轮是完全通过内插得出的。这意味着如果当前的主轴位置并不准确地对应于与凸轮曲线相关的凸轮表内的点，将在相邻的点之间使用线性和三次方内插来确定从轴的位置。这样，即提供最大可能的平滑从轴运动。

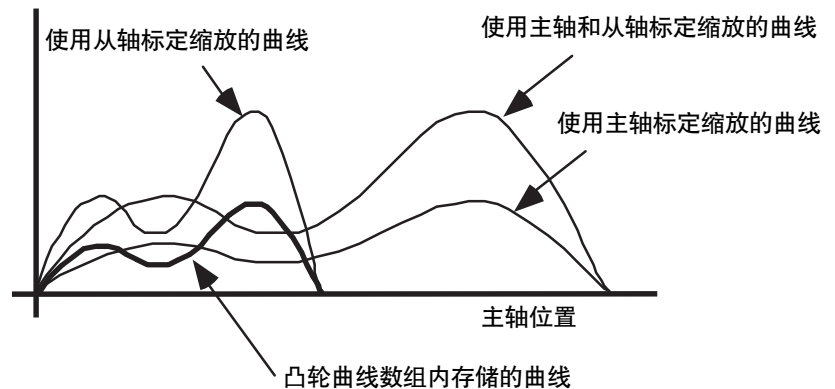
用于生成凸轮曲线的凸轮数组内的每一个点都可以配置用于线性或三次方内插。

电子凸轮在通过从轴后续的点动执行或位移过程后仍保持活跃。这样允许电子凸轮被点动或位移模式所叠加，从而可创建复杂的运动和同步。

位置凸轮标定

在执行时，位置凸轮曲线可以在主轴维度和从轴维度上设置标定。这种标定功能可用于使用存储的凸轮曲线用来确定运动模式的一般形式。然后使用标定参数来定义要执行该模式的总的主轴或从轴行程，如下图所示。这样，一个标准的凸轮曲线可用于生成一系列特定的凸轮曲线。

当凸轮曲线数组被 MAPC 指令指定时，该凸轮曲线数组所定义的主轴和从轴的值将分别采用主轴和从轴的位置单位。相对而言，主轴和从轴标定参数是无单位的，只是用作凸轮曲线的倍数。



凸轮曲线数组

默认情况下，主轴标定和从轴标定参数都被设为 1。要缩放某个位置凸轮曲线，请输入除了 1 以外的主轴标定或从轴标定值。

注意，增加凸轮曲线的主轴标定值将降低该曲线的速度和加速度，而增加凸轮曲线的从轴标定值将增加该曲线的速度和加速度。要保持标定曲线的速度和加速度大致等于无标定曲线的相应值，主轴标定和从轴标定的值应该相等。例如，如果某个曲线的从轴标定值为 2，主轴标定值也应该为 2，才能在执行标定位置凸轮时保持大致相等的速度和加速度。

注意



对于位置凸轮来说，降低主轴标定值或增加从轴标定值会增加曲线的所需速度和减速度。如果超过驱动系统的容量，这可能会造成运动故障。

凸轮曲线执行模式

可选择执行模式为一次或连续，来确定当主轴位置移动超过原始凸轮表所定义的曲线的开始点和结束点范围时，凸轮运动将如何行为。

如果选择一次（默认值），那么仅当主轴移动到由凸轮曲线开始点和结束点定义的范围内时，从轴的凸轮运动才会开始。当主轴移动超出曲线范围时，从轴上的运动停止，MAPC 指令上的过程完成位被设置。注意，与当前的 S 类别的做法相反，当主轴重新回到由开始点和结束点定义的曲线范围时，从轴运动不会恢复。

当选择连续模式时，指定的凸轮曲线一旦开始就将无限地执行下去。在连续操作中，当主轴位置移动超过曲线范围时，曲线的主轴和从轴位置不会“退绕”，从而形成凸轮曲线的重复。在需要位置凸轮以旋转或往复方式连续运动的旋转应用中，这种功能尤其常用。但是，要使用这种方法产生平滑的连续运动，在设计凸轮表的凸轮点时要小心，以保证在计算出的凸轮曲线的开始点和结束点之间没有位置、速度或加速度的不连续。

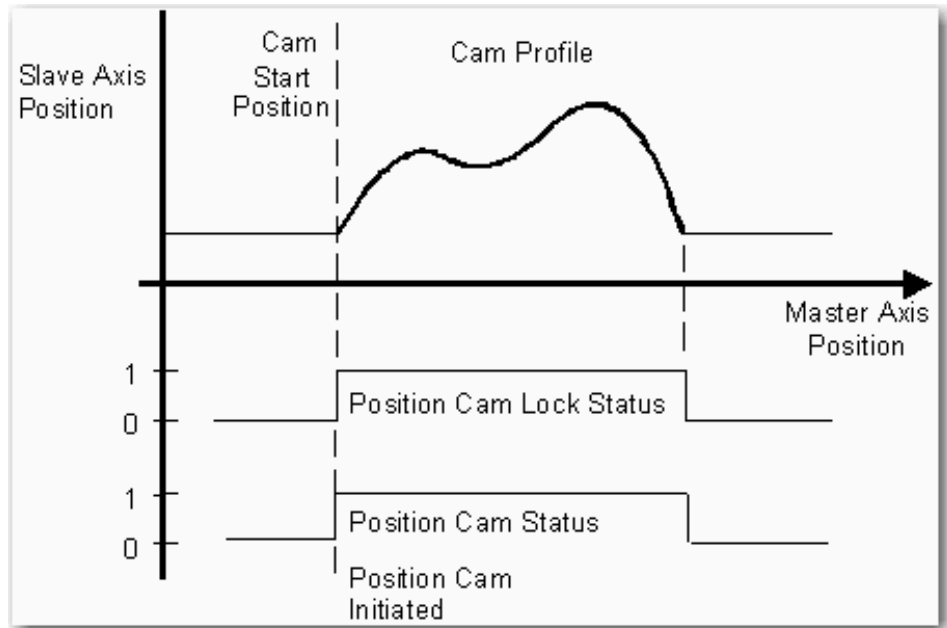
执行计划

对 MAPC 指令执行的控制是通过执行计划参数来实现的。

立即执行

默认情况下，MAPC 指令被计划为立即执行。在这种情况下，在使能位置凸轮过程时没有延迟，主轴锁定位置参数不起作用。从特定凸轮曲线的凸轮锁定位置开始，从轴立即锁定到主轴。

如下图所示，当 MAPC 指令被执行时，凸轮过程即在指定的从轴上发起，从轴的运动状态字内的位置凸轮状态被设置。如果执行计划参数设置为立即，则从轴按照指定的凸轮曲线立即锁定到主轴。指定的从轴上的位置凸轮锁定状态位也被设置，指示这种锁定状态。

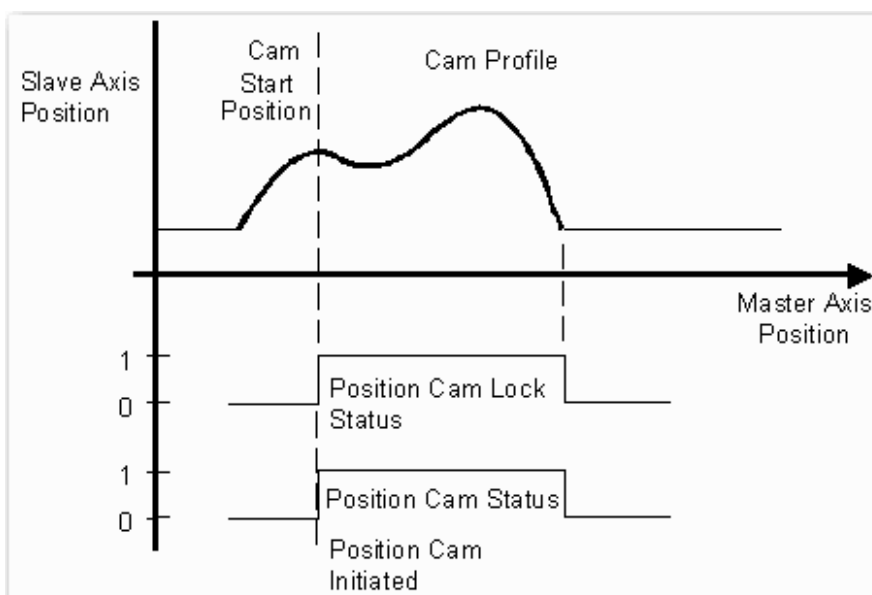


立即执行

更改凸轮锁定位置

MAPC 指令的凸轮锁定位置参数确定了当从轴锁定到主轴时凸轮曲线中的开始位置。通常，凸轮锁定位置设置为凸轮曲线的开始处，如上图所示。由于大多数凸轮表的起点都是 0，凸轮锁定位置通常设置为 0。同时，凸轮锁定位置可以设置为凸轮曲线主轴范围内的任何位置。如果所指定的凸轮锁定位置超出范围，则 MAPC 指令出错。

下图显示了将凸轮锁定位置值指定为凸轮表中除起点以外的其他点时的效果（这里，指定了凸轮曲线本身内的某个位置）。如果主轴正在移动中，在定义凸轮起点时要小心，不要对从轴造成速度或加速度的不连续。



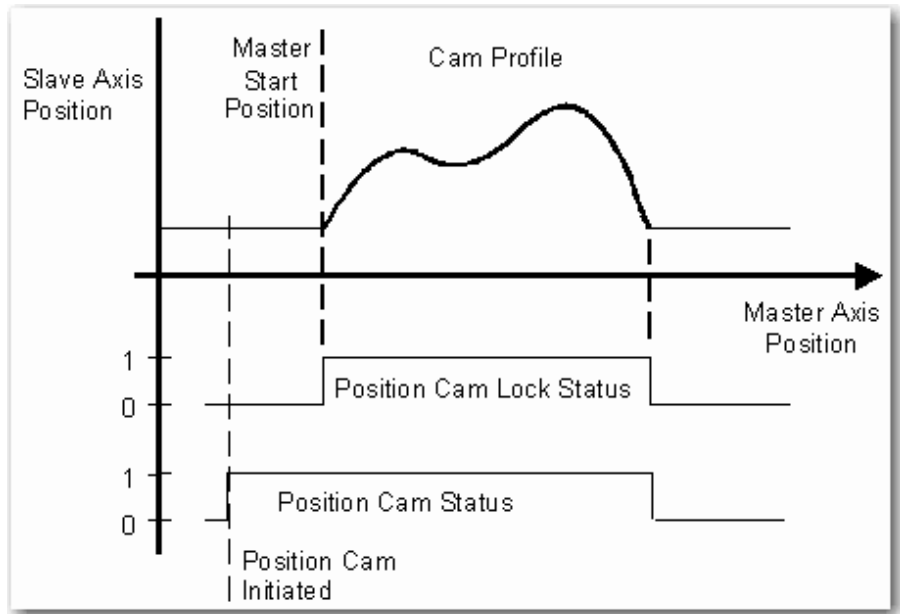
更改凸轮锁定位置

仅正向、仅反向或双向执行

当指令的执行计划参数设置为仅正向或双向时，从轴要等到主轴满足指定的条件后才会锁定的主轴。在这种情况下，主轴受到凸轮过程的监视，来确定主轴何时以指定的方向通过指定的主轴锁定位置。在旋转轴配置中，这种锁定标准仍然有效，与转数无关。

重要提示

如果在 MAPC 指令执行后但满足锁定条件之前，主轴的位置参考被重新定义（例如 MRP 指令），则凸轮曲线生成器按照重新定义位置操作之前有效的绝对位置参考系统来监视主轴。



仅正向、仅反向或双向执行

当主轴的绝对位置按指定的方向（下图中未‘仅正向’方向）通过指定的主轴锁定位置，则指定的从轴的运动状态字的位置凸轮状态位被设置。然后将按照指定的凸轮曲线，在凸轮曲线中指定的凸轮锁定位置发起从轴运动。从这个点开始，只使用主轴位置上的增量变化来从所定义的凸轮曲线中确定相应的从轴位置。对于主轴为旋转轴的应用来说，这是非常重要的，因为位置凸轮以后不受位置退绕过程的影响。

当主轴移出由凸轮曲线所定义的范围时（假设执行模式被配置为一次），运动状态字的位置凸轮锁定位和位置凸轮状态位都被清除。这种运动状态位条件表示凸轮过程已经完成。所关联的 MAPC 指令的位分支行为，即 PC 位被设置、IP 位被清除，也反映了这一点。

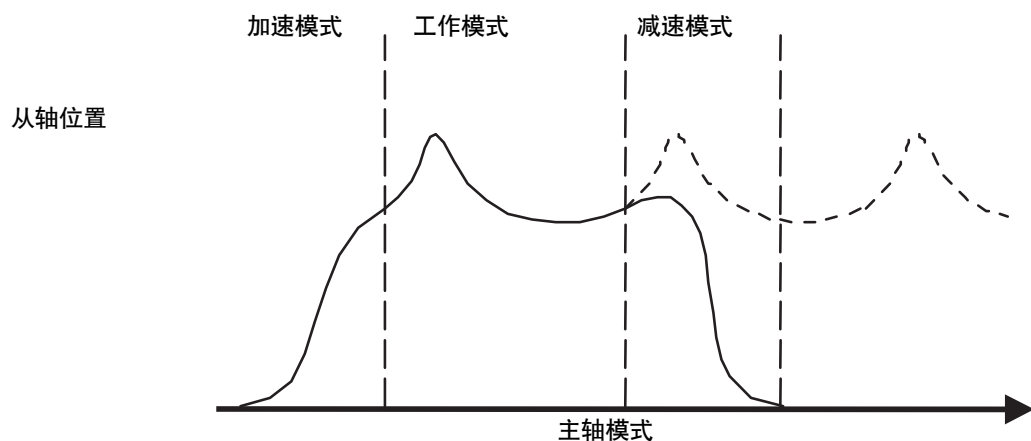
当主轴以仅正向或仅反向方向通过指定的主轴锁定位置时，在位置凸轮运动开始以后，主轴可以改变方向，从轴将相应地反转。

如果 MAPC 指令执行所在的从轴已经在做有效的位置凸轮运动，则生成非法动态参数变更的错误（错误代码 23）。其中唯一的例外是当执行计划被指定为“挂起”的情况。

挂起凸轮执行

作为一种选择，MAPC 指令的执行可以是当前正在执行的位置凸轮完成的延迟挂起。因此选择执行计划为挂起可用于在停止运动的情况下，无缝地混合两个位置凸轮曲线。

挂起执行功能尤其适用于类似于高速包装之类的应用中，其中从轴必须锁定到正在运行的主轴并且使用特定的模式加速到合适的速度。当这种加速模式完成后，它必须平滑地混合到工作模式中，通常为连续执行。要停止从轴，工作模式平滑地混合到减速模式中，从而使轴停止在已知的位置上，如下所示。



挂起凸轮执行

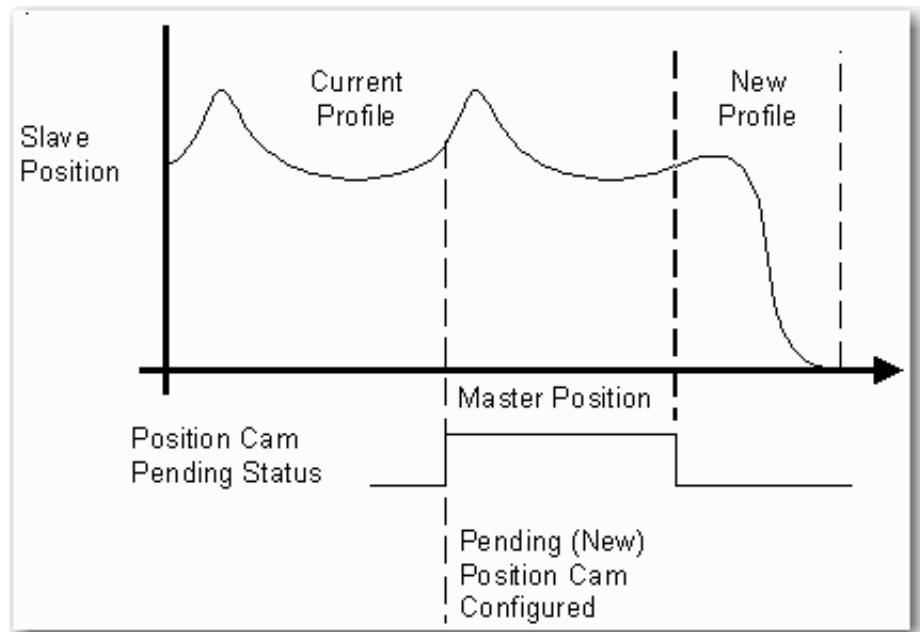
在当前的曲线仍在执行时，通过按挂起的凸轮曲线来执行位置凸轮曲线，可提前设置合适的凸轮曲线参数。这可让您从当前曲线到挂起曲线之间的过渡成为无缝的过渡，主轴与从轴之间的同步得到保持。但是，为保证过渡中运动的平滑，曲线的设计必须使当前曲线的结束与新曲线的开始之间不存在位置、速度或加速度的不连续。这是通过 RSLogix 5000 凸轮曲线编辑器来完成的。

挂起的位置凸轮指令一旦被执行，当主轴通过当前曲线的开始点或者结束点时，新的凸轮曲线自动生效（并变成当前曲线）。如果当前凸轮被配置为执行一次，新的曲线将在通过当前凸轮曲线完成时被发起，当前活跃 MAPC 指令的 PC 位被设置。如果当前凸轮被配置为连续执行，新的曲线将在通过当前凸轮曲线完成时被发起，当前活跃 MAPC 指令的 IP 位被清除。运动控制器根据主轴和从轴相对于变化时第一个曲线的位置，并使用这个信息来保持曲线之间的同步。

如果 MAPC 指令的执行计划被设置为立即，而且位置凸轮曲线目前正在执行，则 MAPC 指令出错。即使轴正在等待锁定到主轴时，也会发生这种情况。

如果执行计划被选择为挂起，但没有相应的正在进行的位置凸轮，MAPC 指令会执行，但直到另一个执行计划为非挂起的 MAPC 指令被发起时才会发生凸轮运动。这样可允许在执行最初的凸轮之前预先加载挂起的位置凸轮曲线。这种方法解决了以下情况：在挂起的位置凸轮被可靠地加载之前立即凸轮可能已经结束。

在挂起的位置凸轮被配置以后，指定的从轴的运动状态字的位置凸轮挂起状态位被设置为 1（真）。当挂起（新的）曲线被发起并变成当前曲线时，位置凸轮挂起状态位被立即清除，如下所示。



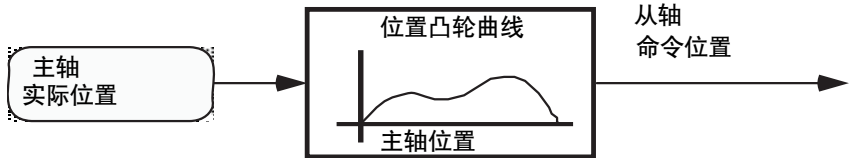
挂起位置凸轮

主轴参考

主轴参考参数确定了要链接到凸轮生成器的主轴位置源。这个源可以是主轴的实际位置或者命令位置。从命令位置可以得到更平滑的运动，但在有些情况下，例如当物理轴不被 ControlLogix 运动模块控制时，实际位置将是唯一可行的选项。

从动至实际位置

当实际位置被输入或选择为主轴参考源时，从轴运动是从主轴的实际位置生成的，如下所示。

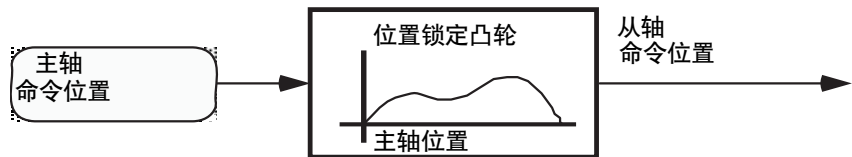


从动至实际位置

实际位置为按照其编码器或其他反馈设备的测量的主轴的当前位置。这是默认选择，当主轴类型被配置为仅反馈时这也是唯一的选择，因为通常需要同步两个轴的实际位置。

从动至命令位置

当命令位置被输入或选择为主轴参考源时，从轴运动是从主轴的命令位置生成的，如下所示。



从动至命令位置

命令位置（仅当主轴的轴类型被配置为伺服或虚轴时才可用）是主轴当前期望的或命令的位置。

由于命令位置不包含任何相关的跟随错误或外部位置扰动，它对于凸轮来说是更加准确和稳定的参考。当凸轮连接至主轴的命令位置时，主轴必须被命令位移，从而在从轴上产生任何运动。请参阅运动轴对象技术参数，了解有关命令位置和实际位置轴参考的更多信息。

主轴方向

正常情况下，主轴方向参数被设置为双向（默认）。但是，当为主轴方向选择为仅正向时，从轴在主轴正的方向跟踪主轴。当选择仅反向时，从轴在主轴反方向跟踪主轴。如果主轴更改方向，从轴不会反转方向，但会停留在主轴反向的位置。这种位置凸轮的单向功能用于提供一种电子滑动离合器，避免当主轴反转方向时凸轮运动生成器通过凸轮曲线向后移动。

当主轴再次反向时，当主轴达到其最初反向的位置时，从轴按所需的方向恢复运动而被重新“拾起”。这样，从轴保持与主轴的同步，同时又禁止了错误方向的运动。当某个方向的运动可能造成对设备或产品的物理损坏时，这种功能就特别有用。

凸轮兼位移

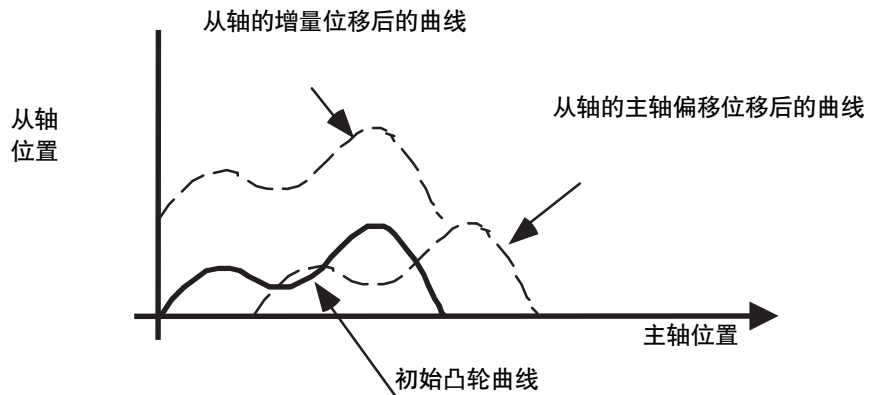
在凸轮传动的同时可以执行运动轴位移，从而在从轴正在运行时提供高级的相位和偏执控制。

增量位移

当位置凸轮正在运行时，增量轴位移 (MAM) 指令可用在从轴上（或者主轴上，如果主轴被配置为伺服操作）。这特别可用于实现相位提前/延迟控制。增量位移距离可用于消息主轴与从轴之间的任何相位误差，或者用于创建准确的相位关系。

主轴偏移位移

当位置凸轮正在运行中也可以使用 MAM 指令，从而在过程中改变凸轮的主轴参考位置。而在从轴上进行的增量位移不同，在从轴上进行的主轴偏移位移是相对于主轴来改变凸轮曲线，如下所示。



主轴偏移位移

当 MAPC 指令（除挂起以外）被发起后，相应的活跃主轴偏移位移被禁止，相应的主轴偏移、抓拍偏移和开始主轴偏移被复位为零。为实现主轴参考位置变化，MAM 指令必须在 MAPC 被发起后再发起。

请参阅轴位移 (MAM) 指令，了解有关主轴偏移位移的更多信息。

停止凸轮

与其他运动生成器（点动、位移、齿轮）一样，活跃的凸轮必须通过各种停止指令、MAS 或 MGS 来停止。当 ControlLogix 处理器改变 OS 模式时也必须停止凸轮运动。特别是，MAS 指令必须能够专门停止凸轮过程。这种行为应与专门用于停止齿轮过程的 MAS 功能相同。

凸轮合并

与其他运动生成器（点动、位移、齿轮）一样，活跃的凸轮也必须与运动合并功能兼容。尤其是位移和点动，必须能够与活跃的凸轮运动合并。这种行为应与应用到齿轮过程的合并功能相同。

故障恢复

有时候需要响应轴的故障条件，而不丢失以凸轮关系锁定的主轴与从轴之间的同步。对于活跃的凸轮，有一些途径来处理轴故障。

创建一个虚轴并将凸轮的一切都连接到它上，并根据需要将这个虚轴齿轮连接到机器上的实际主轴上。将所有轴的各种故障操作都设置为状态唯一。当出现轴故障时（例如变频器故障），监视轴故障状态的应用程序检测到故障，通过停止虚拟主轴对所有活跃的轴执行一次停止控制。在分析器级别，所有部分都仍然完全同步。使用故障轴上的以下错误来确定离开位置有多远。复位故障轴上的故障，使用 MAM 指令以及计算的跟随误差以控制速度回归位置。最后，开始运动虚拟主轴。

与上面的配置相同，但在这种情况下，当从轴故障时，轴的故障操作禁止变频器。这当然会终止从轴上的活跃凸轮过程。在这个点，应用程序应通过虚轴停止所有其他的轴。接下来，通过确定主轴的位置来重新定位故障轴，然后计算如果未发生故障那么从轴应该在的位置。

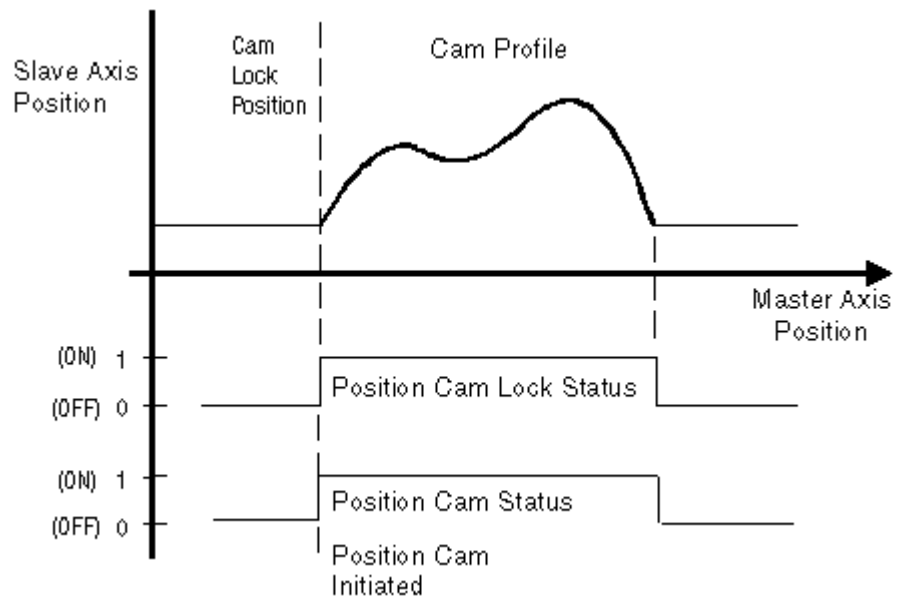
最后，进行一次立即锁定 MAPC 与设置为计算值的凸轮锁定位置重新同步。

重要提示

MAPC 指令执行在一次扫描中完成，因此完成 (.DN) 位和正在进行 (.IP) 位被立即设置。正在进行 (.IP) 位保持设置状态，直到发起的 PCAM 过程完成、被另一条 MAPC 指令取代，或者被运动轴停止命令、合并操作或伺服故障操作终止为止。当 MAPC 执行时，过程完成位被立即清除，当凸轮过程完成（配置为“一次”执行模式时）时被设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。



位置凸轮时间表

算术状态标志： 不受影响

故障条件： 无

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#)上）。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。

轴未配置 (11) 错误代码的扩展错误代码如下：

- 扩展错误代码 1 表示从轴未配置。
- 扩展错误代码 2 表示主轴未配置。

参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MAPC 指令，扩展错误代码 5 指的是从轴标定操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

对于错误代码 54—最大减速度值为零，如果扩展错误代码返回一个正数 (0- n)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

状态位： *MAPC 对状态位的更改*

如果执行计划被设置为立即，则 MAPC 指令执行仅仅将位置凸轮状态和位置凸轮锁定状态位同时设置为真。

位名称	状态	含义
位置凸轮状态	真	位置凸轮被使能
位置凸轮锁定状态	真	从轴按照凸轮曲线锁定到主轴。
位置凸轮挂起状态	假	无挂起位置凸轮

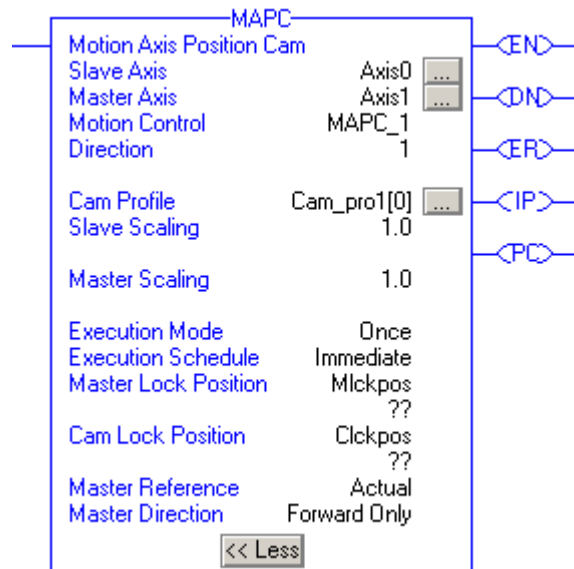
如果执行计划被设置为正向或反向，则 MAPC 指令执行开始将位置凸轮状态设置为真，将位置凸轮锁定状态位设置为假。当执行计划条件满足后，位置凸轮锁定状态位过渡为真。

位名称	状态	含义
位置凸轮状态	真	位置凸轮被使能
位置凸轮锁定状态	假	正等待主轴到达锁定位置的从轴。
位置凸轮挂起状态	假	无挂起位置凸轮

如果执行计划被设置为挂起，则 MAPC 指令执行不会影响位置凸轮状态和位置凸轮锁定状态位的当前状态。位置凸轮挂起状态位被立即设置为真，当挂起凸轮变成活跃凸轮后过渡为假。

位名称	状态	含义
位置凸轮状态	不适用	位置凸轮被使能
位置凸轮锁定状态	不适用	正等待主轴到达锁定位置的从轴。
位置凸轮挂起状态	真	挂起位置凸轮

示例： 梯形图



MAPC 梯形图示例


结构文本

```
MAPC(Axis0,Axis1,MAPC_1,1,Cam_pro1[0],1.0,1.0,Once,  
immediate,Mlckpos,Clckpos,Actual,Forwardonly);
```

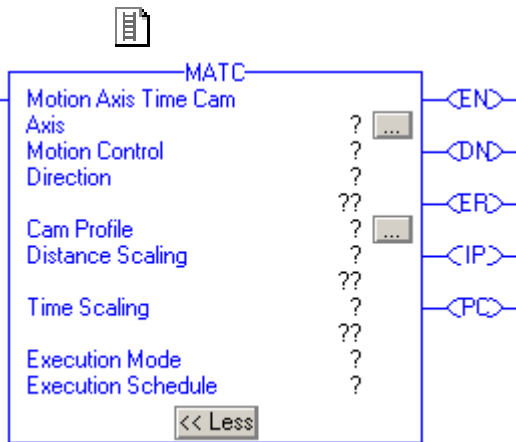
运动轴时间凸轮 (MATC)

运动轴时间凸轮 (MATC) 指令按照指定的凸轮曲线为轴提供作为时间函数的电子凸轮。时间凸轮允许执行除了内置的梯形或 S 曲线位移模式以外的复杂运动模式。当执行时，指定的轴使用由 RSLogix 5000 凸轮曲线编辑器或者通过以前执行的计算凸轮运动曲线 (MCCP) 指令所建立的时间凸轮曲线来与时间同步。轴运动相对于凸轮曲线的方向是由一个非常灵活的方向输入参数来定义的。凸轮方向可以显性地设置为相同或相反，或者相对于当前凸轮方向设置为相反或不变。凸轮曲线可以通过执行计划参数进行配置，以立即或挂起执行当前正在执行的时间凸轮曲线的完成。凸轮曲线也可以执行一次、连续执行或通过指定所需的执行模式来执行。距离和时间标定功能可用于按照标准凸轮曲线来设定轴运动的标定，而不用再创建新的凸轮表和计算新的凸轮曲线。

注意 只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。



操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	应用凸轮曲线的轴的名称。省略号将启动轴属性对话框。
运动控制	MOTION_指令	标签	用于访问块状态参数的结构。

操作数	类型	格式	描述
方向	UINT32	立即或标签	<p>从轴相对于主轴的方向：</p> <ul style="list-style-type: none"> • 相同 – 凸轮曲线中的轴位置值将加到轴的命令位置中。 • 相反 – 凸轮曲线中的轴位置值将从轴的命令位置中减去，生成与原始凸轮表暗指方向相反的方向的轴运动。 <p>或者相对于当前或以前的凸轮方向：</p> <ul style="list-style-type: none"> • 反转 – 位置凸轮的当前或以前的方向从相同更改为相反，反之亦然。选择反转后在第一次执行时，控制器将方向默认为相反。 • 不变 – 允许其他凸轮参数被更改，而不改变当前或以前的凸轮方向。选择不变更后在第一次执行时，控制器将方向默认为相同。
凸轮曲线	CAM_PROFILE	数组	<p>计算出的凸轮曲线数组的标签名称。对于凸轮曲线数组，仅允许零数组元素 ([0])。省略号可启动凸轮曲线编辑器。</p>
距离标定	REAL	立即或标签	<p>设定轴通过凸轮曲线所覆盖的总距离的标定。</p>
时间标定	REAL	立即或标签	<p>设置凸轮曲线覆盖的时间间隔的标定。</p>

操作数	类型	格式	描述
执行模式	UINT32	立即	<p>确定当时间移动超过凸轮曲线结束点时凸轮运动的行为方式。选项如下：</p> <p>0 = 一次 – 当时间凸轮的执行时间超过凸轮曲线内的时间范围时，MATC 指令完成，轴运动停止，时间凸轮状态位被清除。</p> <p>1 = 连续 – 凸轮曲线运动无限地执行下去。</p>
执行计划	UINT32	立即	<p>选择执行凸轮曲线时所用的方法。选项如下：</p> <p>0 = 立即 – 指令被计划为立即执行，对使能时间凸轮过程没有延迟。</p> <p>1 = 挂起 – 延迟时间凸轮的执行，直到当前或下一个立即执行的时间凸轮完成为止。这可用于将新的时间凸轮曲线与正在进行的过程进行混合，以实现无缝过渡。</p>



MATC(Axis,MotionControl, Direction,CamProfile, DistanceScaling,TimeScaling, ExecutionMode, ExecutionSchedule);

结构文本

操作数与梯形图 MATC 指令的操作数相同。对于数组操作数，您不需要包含数组索引。如果您不包含数组索引，则指令从数组 ([0]) 中的第一个元素开始。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
ExecutionMode	一次	0
	连续	1
ExecutionSchedule	立即	0
	pending	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时, 启用位被设置, 直到梯级变成假之前将保持设置状态。
.DN (完成) 位 29	当轴的时间凸轮指令被成功发起后, 完成位被设置。
.ER (错误) 位 28	错误位指示指令检测到错误, 例如轴是否未被配置。
.IP(正在进行) 位 26	正在进行位在正的梯级转换时设置, 在被停止命令、合并、关闭或伺服故障终止时清除。
.PC(过程完成) 位 27	过程完整位在正的梯级过渡时被清除, 在“一次”执行模式下当时间离开由当前活跃的凸轮曲线所定义的时间范围时, 这个位被设置。

说明: 运动轴时间凸轮 (MATC) 指令执行由前一个计算凸轮运动曲线 (MCCP) 指令, 或者由 RSLogix 5000 凸轮曲线编辑器所建立的时间凸轮曲线。时间凸轮提供实现除了内置的梯形或 S 曲线位移模式以外的复杂运动模式的能力。这个指令中不使用最大速度、加速度或减速度限制。从轴的速度、加速度和减速度完全由从相关的凸轮表中得到的指定凸轮曲线来确定。

注意



在轴配置过程中所建立的最大速度、加速度或减速度限制不应用于电子凸轮。

凸轮方向

凸轮可以配置为向轴的命令位置加减其增量成分。对这种行为的控制是通过方向参数来实现的。

同方向凸轮

当为 MATC 指令的方向选择或输入相同时, 从凸轮曲线中计算出的轴位置值将 *添加到* 轴的命令位置中。这是最常用的操作, 因为曲线位置值的使用就像在原始凸轮表内输入一样。也就是说, 连续增加曲线值会导致轴在 *正* 的方向上运动, 反之亦然。

反方向凸轮

当为 MAPC 指令的方向选择或输入相反时，从凸轮曲线中计算出的轴位置值将从轴的命令位置中减去。因此，轴的运动与原始凸轮表内隐含的方向相反。也就是说，连续增加曲线值会导致轴在负的方向上运动，反之亦然。

更改凸轮曲线

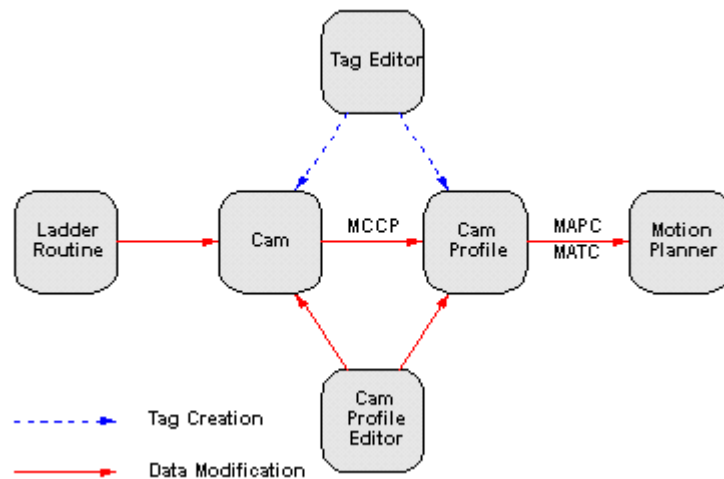
当为方向选择或输入不变时，其他时间凸轮参数可以更改，但保留当前或以前的凸轮方向（相同或相反）。如果当前方向未知或者并不重要，这会非常有用。选择不变更后在第一次凸轮执行时，控制器将方向默认为相同。

更改凸轮方向

如果选择反转时，时间凸轮的当前或以前的方向从相同更改为相反，或者从相反更改为相同。选择反转后在第一次凸轮执行时，控制器将方向默认为相反。

指定凸轮曲线

要执行 MATC 指令，必须指定一个计算的凸轮曲线数组标签。凸轮曲线数组标签可以通过 RSLogix 5000 标签编辑器来创建，或者通过 MATC 指令使用内置的凸轮曲线编辑器来创建，或者通过在现有的凸轮数组上执行计算凸轮运动曲线 (MCCP) 指令来创建。参见下图：



MATC 过程

凸轮曲线数组内部的数据可在编译时使用凸轮曲线编辑器进行修改，或者在运行时使用计算凸轮运动曲线 (MCCP) 指令进行修改。在运动时更改，必须创建一个凸轮数组才能使用 MCCP 指令。请参阅 MCCP 指令指导规范，了解有关转换凸轮数组的更多信息。

这个凸轮曲线数组元素结构除了状态和类型元素以外都被“隐藏”而无法从 RSLogix 5000 标签编辑器进行编辑。这些隐藏元素没有什么用途。状态参数用于指示凸轮曲线数组元素已经计算出来。如果凸轮指令的执行尝试使用凸轮曲线中任何未计算的元素，则指令就会出错。类型参数决定了在这个凸轮数组元素与下一个凸轮元素之间应用的内插类型。

凸轮曲线数组检查

凸轮曲线数组中第一个元素的状态成员比较特别，用于数据完整性检查。因此，MATC 指定凸轮曲线时必须始终将开始索引设置为 0。这第一个凸轮曲线元素状态成员可以取以下值：

状态值	描述
0	凸轮曲线元素尚未计算出来
1	凸轮曲线元素正在进行计算
2	凸轮曲线元素已经计算出来
n	凸轮曲线元素已经计算出来，目前正被 (n-2) 条 MAPC 或 MATC 指令使用。

在指定的轴上开始凸轮之前，MATC 指令首先通过检查凸轮曲线元素的状态成员的值，检查凸轮曲线数组是否已经计算出来。如果状态为 0 或 1，则凸轮曲线尚未计算出来，MATC 指令出错。如果凸轮曲线数组已经完全计算出来 (状态 >1)，则指令递增状态数字，表示已经被这个轴使用。

当凸轮完成或终止后，第一个凸轮曲线数组元素递减，以保持跟踪有效使用相关凸轮曲线的凸轮个数。

线性和三次方内插

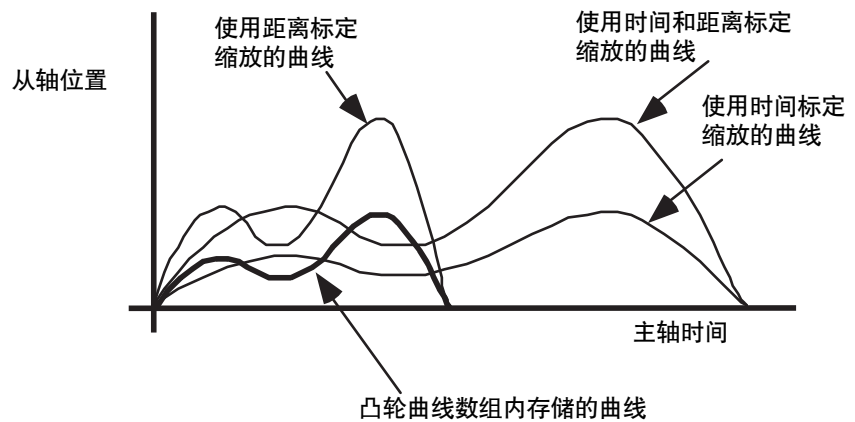
时间凸轮是完全通过内插得出的。这意味着如果当前的主轴时间值并不准确地对应于与凸轮曲线相关的凸轮表内的点，将在相邻的点之间使用线性和三次方内插来确定从轴的位置。这样，即提供最大可能的平滑从轴运动。

用于生成凸轮曲线的凸轮数组内的每一个点都可以配置用于线性或三次方内插。

电子凸轮在通过从轴后续的点动执行或位移过程后仍保持活跃。这样允许电子凸轮被点动或位移模式所叠加，从而可创建复杂的运动和同步。

时间凸轮标定

在执行时，时间凸轮曲线可以在时间和距离上设置标定。通过这种标定功能可将存储的曲线仅用作运动的形式，而使用标定来定义曲线被执行的时间或距离，如下所示。



时间凸轮标定

当凸轮曲线数组被 **MATC** 指令指定时，该凸轮曲线数组所定义的主协调值可采用时间单位（秒），从值采用从轴的单位。相对而言，时间和距离标定参数是“无单位”的值，只是用作凸轮曲线的倍数。

默认情况下，时间和距离标定参数都被设为 1。要缩放某个时间凸轮曲线，请输入除了 1 以外的时间标定或距离标定值。

增加凸轮曲线的时间标定值将降低该曲线的速度和加速度，而增加凸轮曲线的距离标定值将增加该曲线的速度和加速度。要保持标定曲线的速度和加速度大致等于无标定曲线的相应值，时间标定和距离标定的值应该相等。例如，如果某个曲线的距离标定值为 2，时间标定值也应该为 2，才能在执行标定时间凸轮时保持大致相等的速度和加速度。

注意



对于时间凸轮来说，降低时间标定值或增加距离标定值会增加曲线的所需速度和减速度。如果超过驱动系统的容量，这可能会造成运动故障。

凸轮曲线执行模式

可选择执行模式为一次或连续，来确定当时间移动超过原始凸轮表所定义的曲线的结束点范围时，凸轮运动将如何行为。

如果选择一次（默认值），那么轴的凸轮曲线运动将立即开始。当时间凸轮的执行时间超过凸轮曲线内的时间范围时，MATC 指令完成，轴运动停止，从轴运动状态字内的时间凸轮状态位被清除。

当选择连续模式时，指定的凸轮曲线立即开始，并将无限地执行下去。对于连续操作，当时间移动超过凸轮曲线的结束时，时间将“退绕”到凸轮曲线的开始处，从而使凸轮曲线无限地重复。在需要时间凸轮以旋转或往复方式连续运动的旋转应用中，这种功能尤其常用。但是，要使用这种方法产生平滑的连续运动，在设计凸轮表的凸轮点时要小心，以保证在计算出的凸轮曲线的开始点和结束点之间没有位置、速度或加速度的不连续。

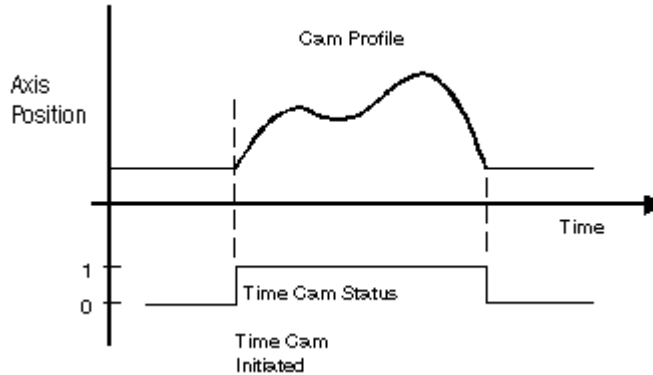
执行计划

对 MATC 指令执行计划的控制是通过执行计划参数来实现的。

立即执行

默认情况下，MATC 指令被计划为立即执行，其优点是执行计划参数的默认设置是立即。在这种情况下，在使能时间凸轮过程时没有延迟。

如下图所示，当 MATC 指令被执行时，凸轮过程即在指定的轴上发起，轴的运动状态字内的时间凸轮状态位被设置。如果执行计划参数设置为立即，则轴按照指定的凸轮曲线立即锁定到时间主坐标。



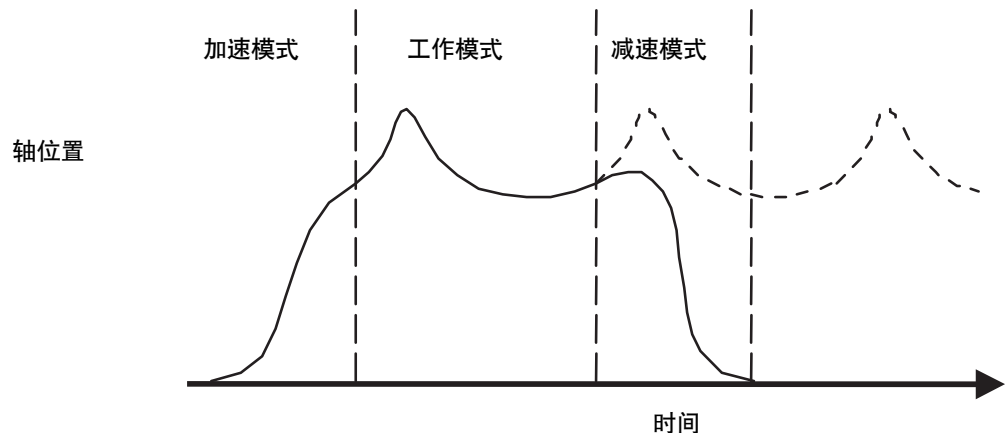
立即执行

如果 MATC 指令执行所在的轴已经在做有效的时间凸轮运动，则生成非法动态参数变更的错误 (错误代码 23)。其中唯一的例外是当执行计划被指定为“挂起”的情况。

挂起凸轮执行

作为一种选择， MATC 指令的执行事实上可以是当前正在执行的时间凸轮完成的延迟挂起。因此选择执行计划为挂起可用于在停止运动的情况下，无缝地混合两个时间凸轮曲线。

挂起执行功能尤其适用于当轴使用特定速度模式而必须要加速到某个速度的时候。当这种加速模式完成后，它必须平滑地混合到凸轮曲线中，通常为连续执行。要停止轴，工作模式可以平滑地混合到减速模式中，从而使轴停止在已知的位置上，如下所示。



挂起凸轮执行

在当前的曲线仍在执行时，通过按挂起的凸轮曲线来执行时间凸轮曲线，可提前设置合适的凸轮曲线参数。这可让您从当前曲线到挂起曲线之间的过渡成为无缝的过渡，时间主轴与位置从轴之间的同步得到保持。但是，为保证过渡中运动的平滑，曲线的设计必须使当前曲线

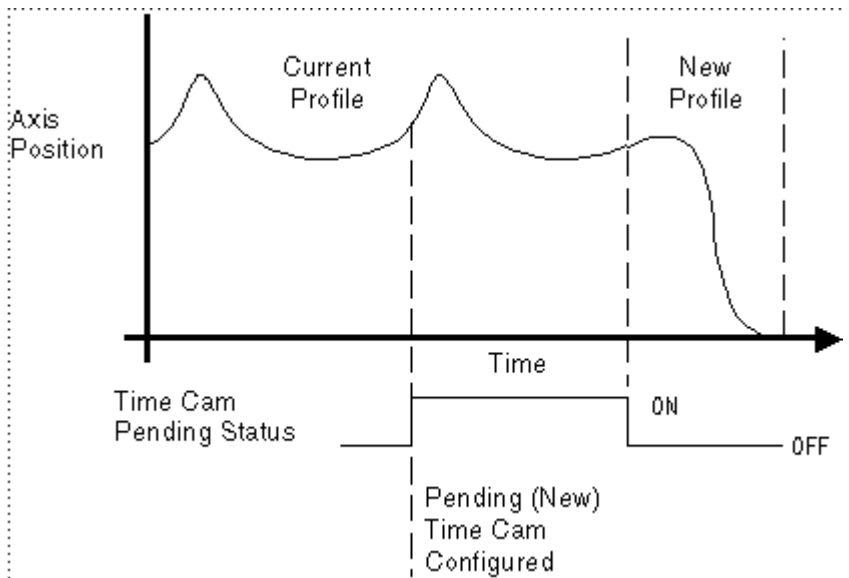
的结束与新曲线的开始之间不存在位置、速度或加速度的不连续。这是通过 RSLogix 5000 凸轮曲线编辑器来完成的。

挂起的时间凸轮指令一旦被执行，当凸轮时间通过当前曲线的结束点时，新的凸轮曲线自动生效（并变成当前曲线）。如果当前凸轮被配置为执行一次，新的曲线将在通过当前凸轮曲线完成时被发起，当前活跃 MATC 指令的 PC 位被设置。如果当前凸轮被配置为连续执行，新的曲线将在通过当前凸轮曲线完成时被发起，当前活跃 MATC 指令的 IP 位被清除。运动控制器根据时间和轴相对于变化时第一个曲线的位置，并使用这个信息来保持曲线之间的同步。

如果 MATC 指令的执行计划被设置为立即，而且时间凸轮曲线目前正在进行，则 MATC 指令产生非法动态更改错误。

如果执行计划被选择为挂起，但没有相应的正在进行的时间凸轮，MATC 指令会执行，但直到另一个执行计划为非挂起的 MATC 指令被发起时才会发生凸轮运动。这样可允许在执行最初的凸轮之前预先加载挂起的凸轮曲线。这种方法解决了以下情况：在挂起的凸轮被可靠地加载之前立即凸轮可能已经结束。

在挂起的时间凸轮被配置以后，指定的轴的运动状态字的时间凸轮挂起状态位被设置为 1（真）。当挂起（新的）曲线被发起并变成当前曲线时，时间凸轮挂起状态位被立即清除，如下所示。



时间凸轮挂起

停止凸轮

与其他运动生成器（点动、位移、齿轮）一样，活跃的凸轮必须通过各种停止指令、MAS 或 MGS 来停止。当 ControlLogix 处理器改变 OS 模式时也必须停止凸轮运动。特别是，MAS 指令必须能够专门停止凸轮过程。这种行为应与专门用于停止齿轮过程的 MAS 功能相同。

凸轮合并

与其他运动生成器（点动、位移、齿轮）一样，活跃的凸轮也必须与运动合并功能兼容。尤其是位移和点动，必须能够与活跃的凸轮运动合并。这种行为应与应用到齿轮过程的合并功能相同。

重要提示

MATC 指令执行在一次扫描中完成，因此完成 (.DN) 位和正在进行 (.IP) 位被立即设置。正在进行 (.IP) 位保持设置状态，直到发起的时间凸轮过程被另一条 MATC 指令取代，或者被运动轴停止命令、合并操作或伺服故障操作终止为止。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。

- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码： 参见[运动控制指令错误代码 \(ERR\)](#)（在[页码 391](#)上）。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MATC 指令，扩展错误代码 5 指的是时间标定操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

对于错误代码 54 – 最大减速度值为零，如果扩展错误代码返回一个正数 (0-*n*)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

MATC 对状态位的更改： *状态位*

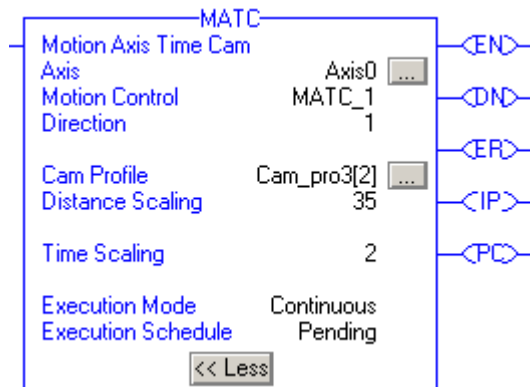
如果执行计划被设置为立即，则 MATC 指令执行仅仅将时间凸轮状态位设置为真。

位名称	状态	含义
TimeCamStatus	真	时间凸轮被使能
TimeCamPendingStatus	假	无挂起时间凸轮

如果执行计划被设置为挂起，则 MATC 指令执行不会影响时间凸轮状态位的当前状态。时间凸轮挂起状态位被立即设置为真，当挂起凸轮变成活跃凸轮后过渡为假。

位名称	状态	含义
TimeCamStatus	不适用	时间凸轮被使能
TimeCamPendingStatus	真	挂起时间凸轮

示例： 梯形图



MATC 梯形图示例

结构文本

```
MATC(Axis0,MATC_1,1,Cam_pro3[2],35,2,Continuous,  
Pending);
```

计算从轴运动值 (MCSV)

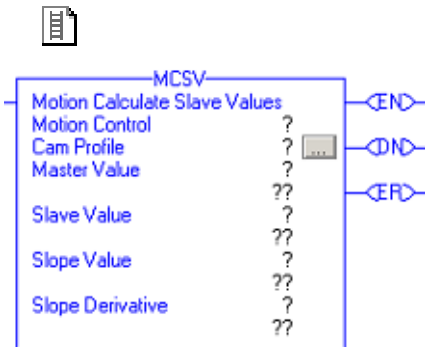
使用计算从轴运动值 (MCSV) 指令可为给定的凸轮曲线和主轴值来计算从轴值、斜率值和斜率的导数。

注意



只能使用一次运动控制标签。不要在其他指令中重复使用它。否则，您可能导致意外的设备运动，造成人员伤害。

操作数：梯形图



操作数	类型	格式	描述
运动控制	MOTION_INSTRUCTION	标签	用于访问指令状态参数的结构。
凸轮曲线	CAM_PROFILE	数组标签	数组索引被设置为 0 的元素数组。它定义了计算从轴值中使用的凸轮曲线。
主轴值	SINT、INT、DINT 或 REAL	立即或标签	在计算从轴值中使用的凸轮曲线上主轴上的准确值。
从轴值	REAL	标签	当主轴在指定的主轴值上时，凸轮曲线上从轴的值。
斜率值	REAL	标签	当主轴在指定的主轴值上时，凸轮曲线上从轴的值的一阶导数。
斜率导数	REAL	标签	当主轴在指定的主轴值上时，凸轮曲线上从轴的值值的二阶导数。

结构文本

操作数与梯形图 MCSV 指令的操作数相同。

MCSV(MotionControl,CamProfile, MasterValue,SlaveValue, SlopeValue,SlopeDerivative)

说明： 计算从轴运动值 (MCSV) 指令可为给定的凸轮曲线和主轴值来确定从轴值、斜率值和斜率的导数。作为对位置和时间凸轮功能的扩展，它为凸轮操作过程中从故障中恢复提供必需的值。

运动控制

以下控制位将受到 MCSV 指令的影响。

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时, 启用位被设置。当梯级从真过渡到假时, 启用位被复位。
.DN (完成) 位 29	当从轴值的计算成功完成后, 完成位被设置。当梯级从假过渡到真时, 完成位被复位。
.ER (错误) 位 28	当从轴值的计算尚未成功完成后, 错误位被设置。当梯级从假过渡到真时, 完成位被复位。

这是一条转换指令:

- 在梯形图中, 当每次应执行指令时, 将梯级输入条件从清除切换到设置。
- 在结构文本中, 限制指令只对转换执行。请参阅附录 C。

算术状态标志: 不受影响

故障条件: 无

错误代码: 参见[运动控制指令错误代码 \(ERR\)](#) (在[页码 391](#)上)。

扩展错误代码: 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号, 在面板中从上到下列出, 第一个操作数被计为零。因此对于 MCSV 指令, 扩展错误代码 2 指的是主轴值操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

MCSV 对状态位的更改: 无

运动组指令 (MGS、MGSD、MGSR、MGSP)

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

简介

组控制指令包括所有在指定组内所有轴上操作的运动控制指令。可应用到组上的指令包括位置抓拍、关闭控制和停止指令。注意，目前每个 Logix 控制器仅支持一个组。

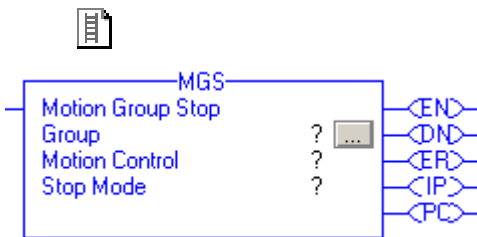
运动组指令指令为：

如果您需要	使用这个指令	在以下语言中可用
发起在轴组上的运动停止。	MGS	梯形图 结构文本
强制组内所有轴进入关闭操作状态。	MGSD	梯形图 结构文本
将轴组从关闭操作状态过渡到轴就绪操作状态。	MGSR	梯形图 结构文本
锁存当前命令和组内所有轴的实际位置。	MGSP	梯形图 结构文本

运动组停止 (MGS)

MGS 指令发起停止指定组内所有轴上所有正在进行的运动，所采用的方法是通过 MGS 指令的停止模式为每个组单独配置或按组配置。如果 MGS 停止模式被指定为编程，则组内的每个轴根据所配置的编程停止模式轴属性而被停止。这与 Logix 操作系统在 Logix 控制器状态变化时所采用的停止机制是相同的。这种编程停止模式属性目前提供五种停止轴的方法：快速停止、快速禁止、硬禁止、快速关闭和硬关闭。或者，可通过使用 MGS 指令选择一个显性的停止模式。如果选择的停止模式为快速禁止，则组内所有轴都通过快速禁止行为来停止。当组内所有轴的运动都进入停止后，在控制结构体内的过程完成 (PC) 位被设置。

操作数： 梯形图



操作数	类型	格式	描述
组	MOTION_GROUP	标签	要在其上执行操作的轴组的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
停止模式 (Stop Mode)	UDINT	立即	控制组内的轴如何被停止。选择下面一种方法： 0 = 编程 – 每个轴的停止都按照各个轴的配置来完成。 1 = 快速停止 – 组内每个轴都按最大减速度进行减速，停止后的轴保持在伺服活跃模式。 2 = 快速禁止 – 组内每个轴都按最大减速度进行减速，停止后的轴被置于轴就绪模式。



MGS(Group,MotionControl, StopMode);

结构文本

操作数与梯形图 MGS 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
StopMode	programmed	0
	faststop	1
	fastdisable	2

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当组编程停止已经对组内所有轴发起后，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的组。
.IP(正在进行) 位 26	在正的梯级过渡时被设置，当运动组编程停止完成后被清除。
.PC(过程完成) 位 27	当组内所有轴都按照每个轴的编程停止模式配置已经进入成功停止时，过程完成位被设置。

说明： 当停止模式参数被设置为编程时，运动组停止 (MGS) 指令将指定组内所有轴的运动按照为每个轴所配置的编程停止模式进入停止状态。如果轴同时有单轴运动和协调运动，则 MGS 使用轴的最大减速度停止单轴运动，使用坐标系的最大减速度停止协调轴运动。对于这种减速始终使用梯形模式，不管编程的模式类型如何。

MGS 指令发起相同的编程停止操作，在处理器的操作系统变化操作模式时（例如从运行模式变为程序模式）自动应用。这在设计定制的运动故障处理程序时尤其有用。

如果 MGS 停止模式被指定为快速停止，则组内的每个轴都被强制执行快速停止过程，不管所配置的编程停止模式如何。组内每个轴都按最大减速度进行减速，而且一旦停止后，轴保持在伺服活跃模式。

如果 MGS 停止模式被指定为快速禁止，则组内的每个轴都被强制执行快速禁止过程，不管所配置的编程停止模式如何。组内每个轴都按最大减速度进行减速，而且一旦停止后，轴被置于轴就绪状态（伺服不活跃，变频器被禁止）。

目前 MGS 指令支持五种编程停止模式：快速停止、快速禁止、硬禁止、快速关闭和硬关闭。每个轴都可以配置为使用这五种停止模式中的一种。下面介绍五种停止模式中的每一种应用到指定组的单个轴上时会有什么效果。

快速停止

对于配置为快速停止的轴，MGS 指令发起停止控制，非常类似于 MAS 指令发起的停止。在这种情况下，运动组编程停止 (MGS) 指令将轴运动进入停止控制，而不禁止轴伺服环。当希望保持伺服控制的情况下需要最快的减速停止时，可使用这种方法。

MGS 指令将在这种停止模式下为轴配置的最大减速度作为应用到轴上的减速斜坡的基础。

快速禁止

对于配置为快速禁止的轴，MGS 指令发起停止控制，非常类似于 MAS 指令发起的停止，区别是当轴进入停止状态后变频器被禁止。当在变频器被禁止之前需要最快的减速停止时，可使用 MGS。

MGS 指令将在这种停止模式下为轴配置的最大减速度作为应用到轴上的减速斜坡的基础。

硬禁止

对于配置为硬禁止的轴，MGS 指令发起等效于 MSF 指令对轴的操作。这个操作立即将相应的变频器输出关闭，并禁止伺服环。根据变频器配置，这可能会造成轴惯性停止，但却以最快速度断开变频器输出电源。

快速关闭

对于配置为快速关闭的轴，MGS 指令发起快速停止，然后将等效于关闭运动轴 (MASD) 指令的操作应用到轴上。这个操作将相应的轴驱动输出关闭，禁止伺服环，打开任何相关运动控制模块的 OK 触点，然后将轴置于关闭状态。

硬关闭

对于配置为硬关闭的轴，MGS 指令发起等效于关闭运动轴 (MASD) 指令对轴的操作。这个操作将相应的轴变频器输出关闭，禁止伺服环，打开任何相关运动控制模块的 OK 触点，然后将轴置于关闭状态。根据变频器配置，这可能会造成轴惯性停止，但却通过 OK 触点以最快速度断开变频器电源。

要成功执行 MGS 指令，必须配置目标组。

重要提示

MGS 指令的执行可能需要多次扫描才能完成，因为消息可能需要组内一个或多个轴运动控制模块。因此完成 (DN) 位可能不立即被设置。但是，正在进行 (.IP) 位被立即设置，过程完成 (.PC) 位被立即清除。正在进行 (.IP) 位保持设置状态，直到组内所有轴的编程停止过程完成，或者停止指令被另一条 MGS 指令取代，或者被伺服故障操作终止为止。只有在任何其他上述终止停止过程以及清除正在进行 (.IP) 位的事件发生之前，组内每个轴所发起的减速模式已经完成，过程完成 (.PC) 位才会被设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

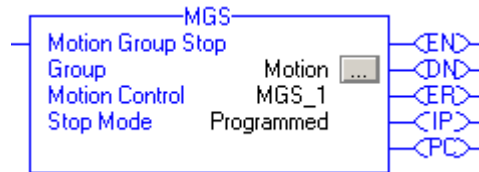
错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

状态位 MGS 对状态位的更改

位名称	状态	定义
StoppingStatus	真	轴正在停止（取决于轴的编程停止模式）。
JogStatus	假	轴不再点动。
MoveStatus	假	轴不再运动。
GearingStatus	假	轴不再传动。
HomingStatus	假	轴不再归零。

示例： 当输入条件为真时，控制器停止组 1 内的所有轴上的运动。在控制器停止所有运动以后，轴被禁止。

梯形图



MGS 梯形图示例

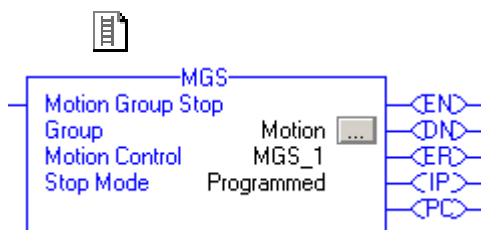
结构文本

```
MGS(Motion,MSG_1,Programmed);
```

关闭运动组 (MGSD)

使用 MGSD 指令可强制指定组内的所有轴进入关闭状态。轴的关闭状态为伺服关闭，输出被去激活，运动控制模块的 OK 固态继电器触点（如果适用）被打开。轴组保持关闭状态，直到组关闭复位被执行，或者每个轴通过关闭运动轴 (MASD) 指令被单独复位为止。

操作数：梯形图



MGSD(Group,MotionControl);

操作数	类型	格式	描述
组	MOTION_GROUP	标签	要在其上执行操作的轴组的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MGSD 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	启用位指示指令被启用。它保持为设置状态，直至伺服消息完成而且梯级输入条件变为假为止。
.DN (完成) 位 29	完成位表示指令何时将轴组置于关闭操作状态。
.ER (错误) 位 28	错误位指示指令检测到错误，例如发往伺服模块的消息失败。

说明： 关闭运动组 (MGSD) 指令将变频器输出关闭，禁止指定组内所有轴的伺服环，为组内所有适用运动控制模块打开任何相关联的 OK 触点。这个操作将组的所有轴置于关闭状态。MGSD 指令仅带一个参数；只需选择或输入要关闭的组即可。

MGSD 指令发起的另一个操作是清除所有正在进行的运动过程并清除所有运动状态位。与此操作相关，命令还会清除当前可能为组内每个轴设置的所有运动控制指令的 .IP 位。

MGSD 指令强制目标轴组进入关闭状态。关闭状态的独特特点之一就是，所有组运动控制模块的 OK 固态继电器触点打开。这种功能可用于打开控制各种变频器系统主电源的急停环。

关机状态的另一个特点是，任何发起组内轴的运动的指令都被阻止执行。尝试这样做会导致执行错误。只有通过执行其中一个关闭复位指令，才能成功地发起运动。

要成功执行 MGSD 指令，必须创建和配置目标组。

重要提示

MGSD 指令执行可能需要多次扫描来执行，因为它需要向一个或多个运动控制模块传输一条消息。因此，完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

状态位： *MGSD 对状态位的更改*

位名称	状态	定义
ServoActionStatus	假	轴正处于轴就绪状态，而且伺服环不活跃。
DriveEnableStatus	假	轴变频器使能输出不活跃。
ShutdownStatus	真	轴处于关闭状态。
AccelStatus	假	轴没有在加速。
DecelStatus	假	轴没有在减速。
GearingLockStatus	假	轴没有被锁定。
JogStatus	假	轴没有在点动。
MoveStatus	假	轴没有在平移。
GearingStatus	假	轴没有在齿轮运动中。
HomingStatus	假	轴没有在归零。

示例： 当输入条件为真时，控制器强制组 1 内的所有轴进入关闭操作状态。

梯形图



MGSD 梯形图示例

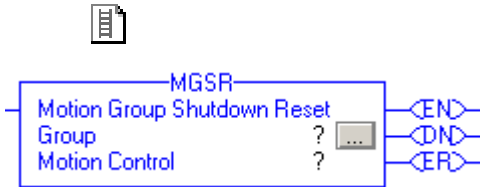
结构文本

```
MGSD(Motion, MGSD_2);
```

运动组关闭复位 (MGSR)

使用 MGSR 指令可将轴组从关闭操作状态过渡到轴就绪操作状态。这个命令的结果是，所有与组内的这个轴相关的故障将被清除，与指定的组相关的运动控制模块的 OK 继电器触点被关闭。

操作数：梯形图



MGSR(Group,MotionControl);

操作数	类型	格式	描述
组	MOTION_GROUP	标签	要在其上执行操作的轴组的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MGSR 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	启用位指示指令被启用。它保持为设置状态，直至伺服消息完成而且梯级输入条件变为假为止。
.DN (完成) 位 29	完成位表示指令何时将轴组从关闭操作状态中复位。
.ER (错误) 位 28	错误位指示指令检测到错误，例如发往伺服模块的消息失败。

说明： 运动组关闭复位 (MGSR) 指令通过清除所有轴故障，关闭组内运动控制模块的任何相关 OK 固态继电器触点，将指定组内的所有组脱离关闭状态。这个操作将运动组内的所有轴置于轴就绪状态。

就像 MGSD 指令一样，强制目标组内的所有轴进入关闭状态。MGSR 指令将指定组内的所有轴离开关闭状态，进入轴就绪状态。关闭状态的独特特点之一就是，如果支持，则每个组运动控制模块的 OK 固态继电器触点打开。因此，应用到运动控制模块组的 MGSR 指令的结果是，所有运动控制模块的 OK 继电器触点关闭。这种功能可用于关闭控制各种变频器系统主电源的急停环，并允许客户向变频器重新施加电源。

要成功执行 MGSR 指令，必须配置目标组。

重要提示

MGSR 指令执行可能需要多次扫描来执行，因为它需要向一个或多个运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

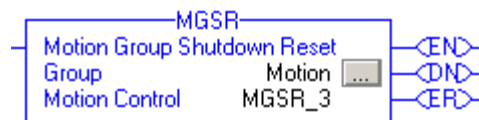
错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

状态位： *MGSR 对状态位的更改*

位名称	状态	定义
ServoActionStatus	假	轴正处于轴就绪状态，而且伺服环不活跃。
DriveEnableStatus	假	轴变频器使能输出不活跃。
ShutdownStatus	假	轴不在关闭状态。

示例： 当输入条件为真时，控制器将组 1 内的所有轴从关闭操作状态过渡到轴就绪操作状态。

梯形图



MGSR 梯形图示例

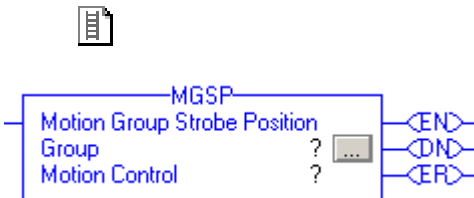
结构文本

```
MGSR(Motion, MGSR_3);
```

抓拍运动组位置 (MGSP)

使用 MGSP 指令可在单个时间点锁存指定组内所有轴的当前命令位置 and 实际位置。锁存的位置将在运动轴对象中的 StrobeActualPosition 和 StrobeCommandPosition 参数内提供给组内配置的每一个轴。

操作数: 梯形图



MGSP(Group,MotionControl);

操作数	类型	格式	描述
组	MOTION_GROUP	标签	要在其上执行操作的轴组的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MGSP 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴组被成功置于关闭状态时, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的组。

说明: 抓拍运动组位置 (MGSP) 指令在执行时以同步方式锁存指定组内所有轴的所有命令位置和实际位置值。MGSP 指令仅带一个参数; 只需选择或输入要抓拍的轴即可。

如果目标组没有出现在可用组列表中, 则该组尚未配置用于操作。请使用标签编辑器来创建和配置新的组。

MGSP 指令可随时用来捕捉指定组内所有轴的命令位置和实际位置信息。通常在涉及到组内不同轴的位置值的计算之前会需要这个操作。

要成功执行 MGSP 指令，必须配置目标组。

重要提示

MGSP 指令执行在一次扫描中完成，因此完成 (.DN) 位被立即设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

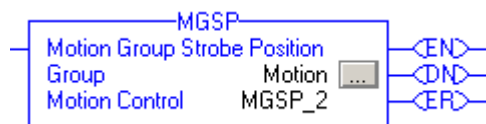
错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

状态位： *MGSP 对状态位的更改*

无

示例： 当输入条件为真时，控制器锁存组 1 内的所有轴的当前命令位置和实际位置。

梯形图



MGSP 梯形图示例

结构文本

```
MGSP(Motion, MGSP_2);
```

注意：

运动事件指令

(MAW、MDW、MAR、MDR、MAOC、MDOC)

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

简介

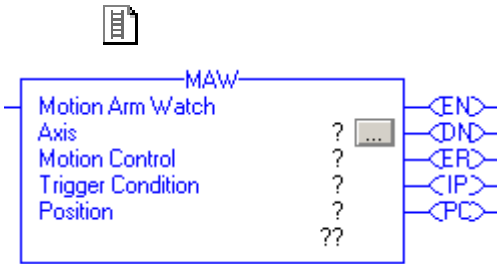
运动事件指令控制着特殊事件检查功能的准备和禁止，例如记录和位置监视。运动事件指令指令为：

如果您需要	使用这个指令	在以下语言中可用
准备某个轴的位置监视事件检查。	MAW	梯形图 结构文本
禁止某个轴的位置监视事件检查。	MDW	梯形图 结构文本
准备某个轴的伺服模块记录事件检查。	MAR	梯形图 结构文本
禁止某个轴的伺服模块记录事件检查。	MDR	梯形图 结构文本
准备凸轮输出	MAOC	梯形图 结构文本
禁止凸轮输出	MDOC	梯形图 结构文本

开启运动监视 (MAW)

使用 MAW 指令来准备运动控制模块为指定的轴进行位置监视事件检查。当这条指令被调用时，将使用轴的监视位置以及指定的正向或反向事件条件使能一个位置监视事件。准备完成后，当满足指定的监视事件条件、事件 (PC) 位被设置以及轴数据结构中的监视事件状态位被设置时，轴的实际位置将按照监视位置而被监视。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
触发条件	布尔型	立即	选件监视事件触发条件: 0 = 正向 – 伺服模块寻找实际位置从小于监视位置改变为大于监视位置。 1 = 反向 – 伺服模块寻找实际位置从大于监视位置改变为小于监视位置。
位置	REAL	立即或标签	位置监视的新值。



MAW(Axis,MotionControl,TriggerCondition,Position);

结构文本

操作数与梯形图 MAW 指令的操作数相同。

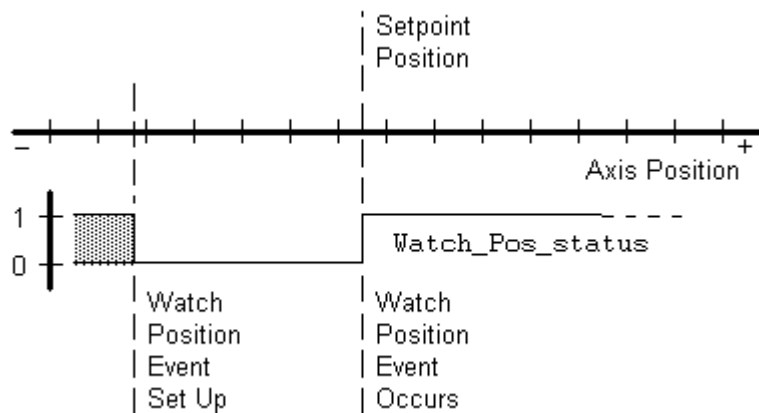
对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
TriggerCondition	正向	0
	反向	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴监视事件插件已经成功准备后, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	在正的梯级过渡时被设置, 在监视事件发生后, 或者被其他关闭运动监视所取代或者被运动禁止监视命令所终止时被清除。
.PC(过程完成) 位 27	当监视事件发生时被设置。

说明: 开启运动监视 (MAW) 指令设置在指定的物理轴到达指定的设置点位置时发生位置监视事件, 如下所示。

**设置点位置**

位置监视事件适用于当轴正在运行时将某个操作与指定的轴位置同步, 例如在某个轴位置激活某个电磁阀来将纸箱推入传送带。选择或输入所需的物理轴、所需的触发条件, 并为所需的监视位置输入一个值或标签变量。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

当准备位置监视指令被执行时， WatchEventStatus 位被设置为 0(假)，物理轴的实际位置被监视（按照伺服环的刷新速率），直至它达到指定的监视位置。当位置监视事件发生后，该轴的 WatchEventStatus 位被设置为 1(真)。

多个位置监视事件在某个给定的时间可能激活，但是对于任何给定的物理轴一次只能有一个激活。每个事件被独立监视，可能会使用相应的 WatchEventStatus 位进行检查。

重要提示 在大的 I/O 连接中，力的值可能会减慢控制器处理重复位置监视的速度。

要成功执行 MAW 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示 MAW 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAW 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

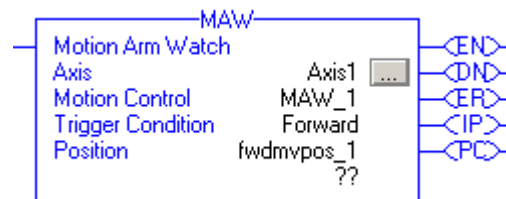
相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	无资源 (2)	没有足够的内存资源来完成请求。(SERCOS)

状态位： MAW 对状态位的更改

位名称	状态	含义
WatchEventArmedStatus	真	轴正在寻找位置监视事件。
WatchEventStatus	假	上一个监视事件被清除。

示例： 当输入条件为真时，控制器准备轴 1 上的位置监视事件检查。

梯形图



MAW 梯形图示例

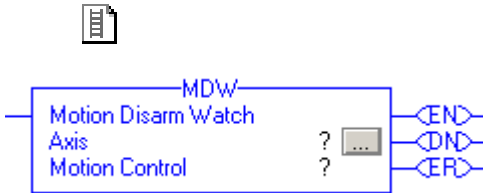
结构文本

```
MAW(Axis1,MAW_1,Forward,fwdmvpos_1);
```

关闭运动监视 (MDW)

使用 MDW 指令来禁止某个轴的位置监视事件检查。这个指令的效果是同时清除轴数据结构中的事件监视状态和监视准备状态位。执行这条指令还会清除与开启运动监视 (MAW) 指令相关的正在进行位。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



MDW(Axis,MotionControl);

结构文本

操作数与梯形图 MDW 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴监视事件检查已经成功禁止后, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。

说明: 关闭运动监视 (MDW) 指令取消由上一个 MAW 指令所建立的位置监视事件检查。关闭运动位置监视指令不需要参数; 只需输入或选择所需的物理轴即可。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

要成功执行 MDW 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示

MDW 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

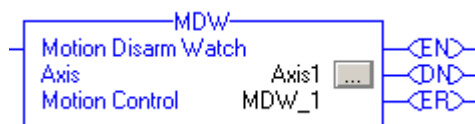
错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

状态位： MDW 对状态位的更改

位名称	状态	含义
WatchEventArmedStatus	假	轴没有正在寻找位置监视事件。
WatchEventStatus	假	上一个监视事件被清除。

示例： 当输入条件为真时，控制器禁止轴 1 上的位置监视事件检查。

梯形图



MDW 梯形图示例

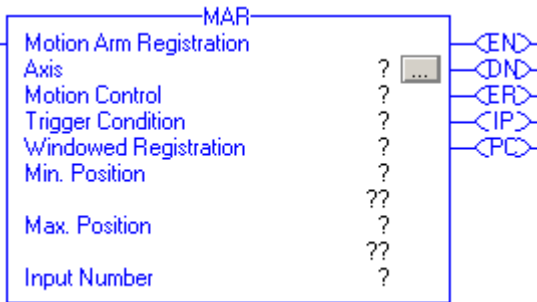
结构文本

```
MDW(Aaxis1,MDW_1);
```

开启轴位置抓拍 (MAR)

使用 MAR 指令来准备伺服模块为指定的轴进行记录事件检查。当这条指令被调用时，将基于选定的记录输入和指定的触发条件来准备一个记录事件。当指定的记录输入过渡满足触发条件时，运动控制模块在事件发生时根据硬件锁存的编码器计数来计算轴位置，并将其存储在轴的数据结构内的相应的记录位置变量中。同时，指令的事件 (PC) 位被同时设置，以及轴数据结构中的记录事件状态位。选择了窗口登录，则仅接受那些计算出的记录位置位于最大和最小位置窗口内的记录事件。如果记录位置位于此窗口以外，则自动重寻准备记录事件检查。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
触发条件	布尔型	立即	定义用来定义记录事件的记录输入过渡。选择任一： 0 = 在正边沿触发 1 = 在负边沿触发
窗口登录	布尔型	立即	如果记录是窗口型的，即计算出的记录位置位于要接受为有效记录事件的指定最大和最小位置限制以内，则设置 (1)。选择任一： 0 = 已禁止 1 = 已使能

操作数	类型	格式	描述
最小位置	REAL	立即或标签	当使能窗口登录时使用。记录位置必须大于最小位置限制，才会接受记录事件。
最大位置	REAL	立即或标签	当使能窗口登录时使用。记录位置必须小于最大位置限制，才会接受记录事件。
输入编号	UINT32	1 或 2	指定要选择的记录输入。 1 = 记录 1 位置 2 = 记录 2 位置



MAR(Axis,MotionControl, TriggerCondition, WindowedRegistration, MinimumPosition, MaximumPosition, InputNumber);

结构文本

操作数与梯形图 MAR 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

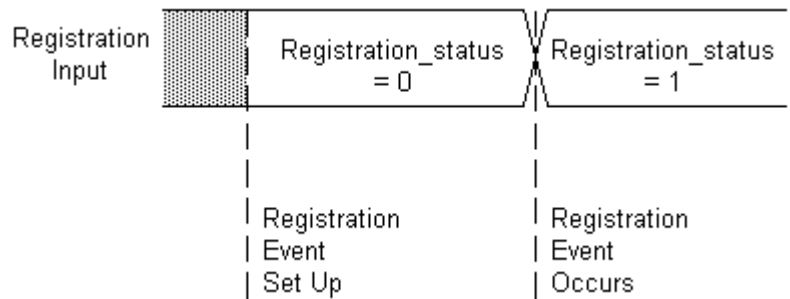
此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字：
TriggerCondition	positive_edge	0
	negative_edge	1
WindowedRegistration	已禁止	0
	已使能	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴记录事件插件已经成功准备后，完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	在正的梯级过渡时被设置，在记录事件发生后，或者被其他运动记录准备所取代或者被运动禁止记录命令所终止时被清除。
.PC(过程完成) 位 27	当记录事件发生时被设置。

说明： 开启轴位置抓拍 (MAR) 指令建立一个记录事件，在为该轴选择的指定高速记录输入的指定边沿上存储指定物理轴的实际位置。

当 MAR 指令被执行时，RegEventStatus 位设置为 0(假)，为指定轴选择的记录输入受到运动控制模块的监视，直到发生选定类型的记录输入过渡 (记录事件) 为止。当记录事件发生后，轴的 RegEventStatus 位被设置为 1(真)，轴的实际位置被存储在与记录输入相对应的记录位置变量中 (例如，记录 1 位置 1 或记录 2 位置)。

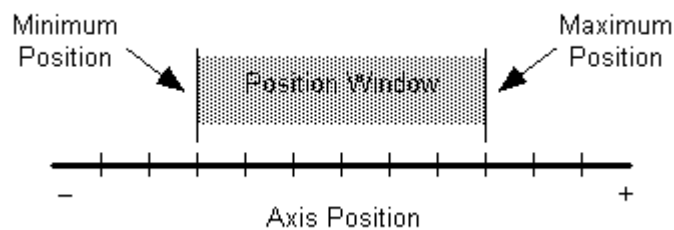


记录

对于某个给定的轴可随时激活多个记录事件，但每个记录输入只能有一个事件是激活的。每个事件被独立监视，可能会使用相应的 RegEventStatus 位进行检查。

窗口登录

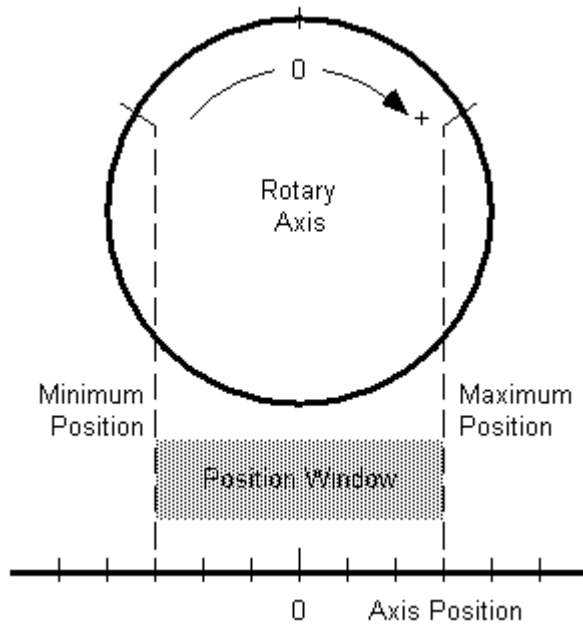
当“窗口记录”复选框被选中时，只有当轴位于最小和最大位置所定义的窗口以内时，选定的行程状态才会在发生时导致记录事件，如下所示。



窗口登录

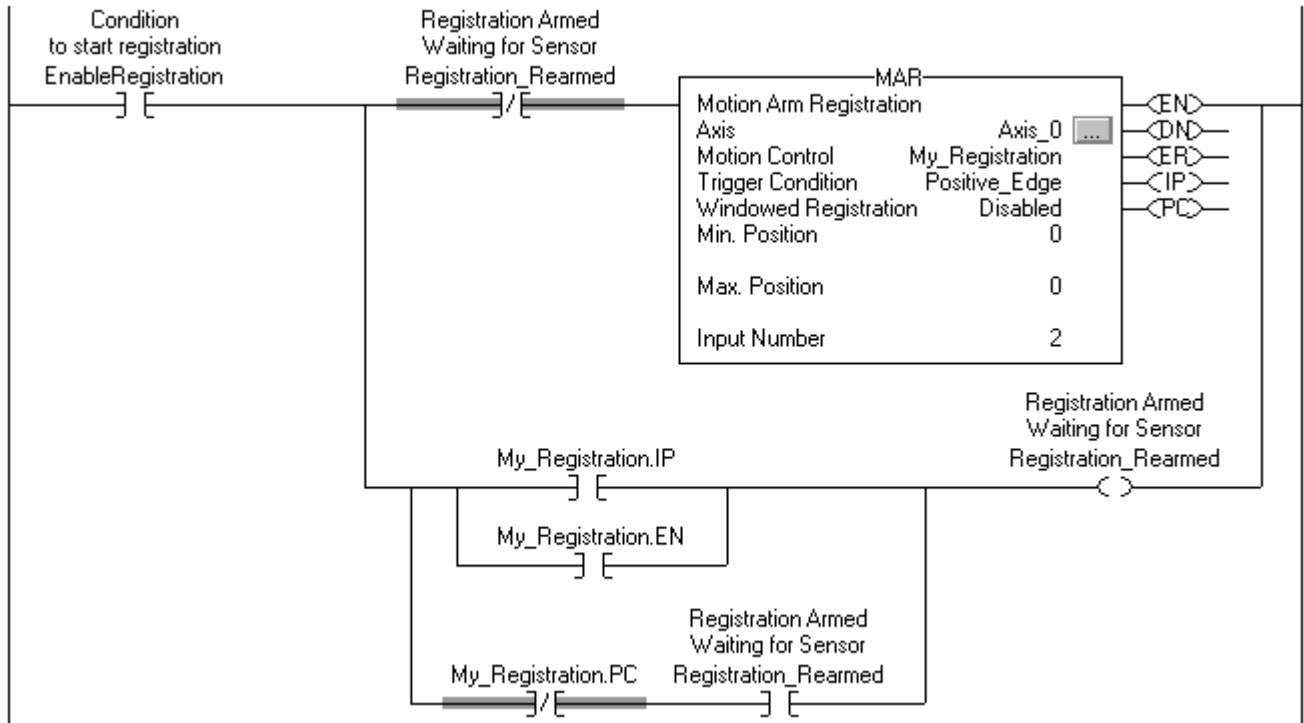
为所需的绝对值位置输入值或标签变量，用这些位置来定义记录输入的选定行程状态为有效的位置窗口。窗口登录适用于提供一种忽略记录传感器的杂波或随机过渡的机制，从而提高高速记录输入的噪声抗扰性。

对于线性轴，这些值可以为正、为负或者正负数组合。但是，最小位置值必须小于最大位置值，才会发生记录事件。对于旋转轴，两个值都必须小于在运动控制器的机器设置菜单中所设定的退绕值。最小位置值可以大于最大位置值，适用于穿过轴的退绕点的记录窗口，如下所示。



旋转轴的位置窗口

重新准备 MAR 指令 如果您的应用需要快速连续地探测记录传感器，我们建议您使用以下逻辑：



连续记录探测的梯形图逻辑

要重新准备 MAR 指令，梯级必须从假变为真。这个逻辑工作的速度取决于以下方面：

- 程序扫描周期
- 运动任务过程的刷新速率

重要提示 在大的 I/O 连接中，力的值可能会减慢控制器处理重复运动记录的速度。

要成功执行 MAR 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示 MAR 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAR 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	无资源 (2)	没有足够的内存资源来完成请求。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	无效值 (3)	提供的记录输入超出了范围。
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)。	重新定义位置、归零和记录 2 是相互排斥的。(SERCOS)

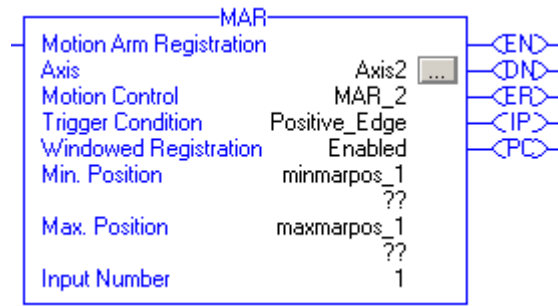
参数超出范围 (13) 错误的扩展错误代码工作方式有点不同。不是采用标准的枚举，扩展错误代码所显示的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MAR 指令，扩展错误代码 4 指的是最小位置值。然后必须将您的值与指令可以接受的值范围进行对比。

状态位： *MAR 对状态位的更改*

位名称	状态	含义
RegEventArmedStatus	真	轴正在寻找记录事件。
RegEventStatus	假	上一个记录事件被清除。

示例： 当输入条件为真时，控制器准备 *axis_0* 上的伺服模块记录事件检查。

梯形图



MAR 梯形图示例

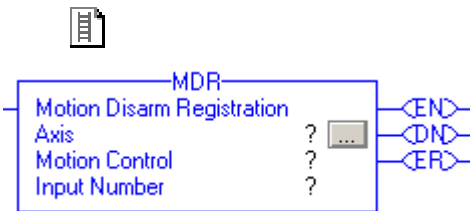
结构文本

```
MAR(Axis2,MAR_2,positive_edge,enabled,minmarpos_1,maxmarpos_1,1);
```

关闭轴位置抓拍 (MDR)

使用 MDR 指令来禁止指定的运动控制模块为指定的轴进行的记录输入事件检查。这条指令的结果是同时清除 RegEventStatus 和 RegArmedEventStatus 位。控制开启轴位置抓拍指令的正在进行位 (如果有的话)，将在执行 MDR 指令后被清除。

操作数: *梯形图*



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
输入编号	UINT32	1 或 2	指定要选择的记录输入。 1 = 记录 1 位置 2 = 记录 2 位置



MDR(Axis, MotionControl, InputNumber);

结构文本

操作数与梯形图 MDR 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当轴监视事件检查已经成功禁止后, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。

说明: 关闭轴位置抓拍 (MDR) 指令取消由上一条开启轴位置抓拍指令所建立的记录事件检查。只有与指定的记录输入相关的记录检查会被禁止。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

要成功执行 MDR 指令，目标轴必须配置为伺服或者反馈唯一轴。否则，该指令出错。

重要提示

MDR 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MDR 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	无效值 (3)	提供的记录输入超出了范围。

参数超出范围 (13) 错误的扩展错误代码工作方式有点不同。不是采用标准的枚举，扩展错误代码所显示的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MDR 指令，扩展错误代码 2 指的是输入编号操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

状态位： *MDR 对状态位的更改*

位名称	状态	含义
RegEventArmedStatus	假	轴没有正在寻找记录事件。
RegEventStatus	假	上一个记录事件被清除。

示例： 当输入条件为真时，控制器禁止 *axis_0* 上的记录事件检查。

梯形图



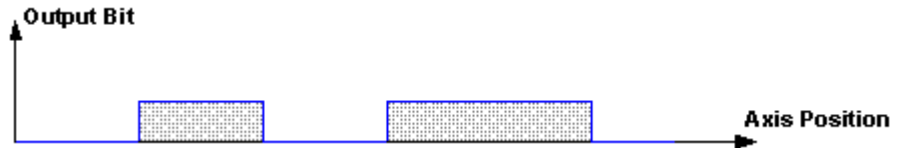
MDR 梯形图示例

结构文本

```
MRD(Axis2,MDR_1,2);
```

开启运动凸轮输出 (MAOC)

运动轨迹规划凸轮输出功能提供了按轴位置对输出位进行设置和复位的功能。



运动轨迹规划功能

在内部，凸轮输出对象处理运动轨迹规划凸轮输出功能。每个凸轮输出对象负责一个输出，其中包含 32 个输出位。每一个输出位都可以使用凸轮输出曲线进行单独编程，可对位置偏移和时间延迟进行补偿。

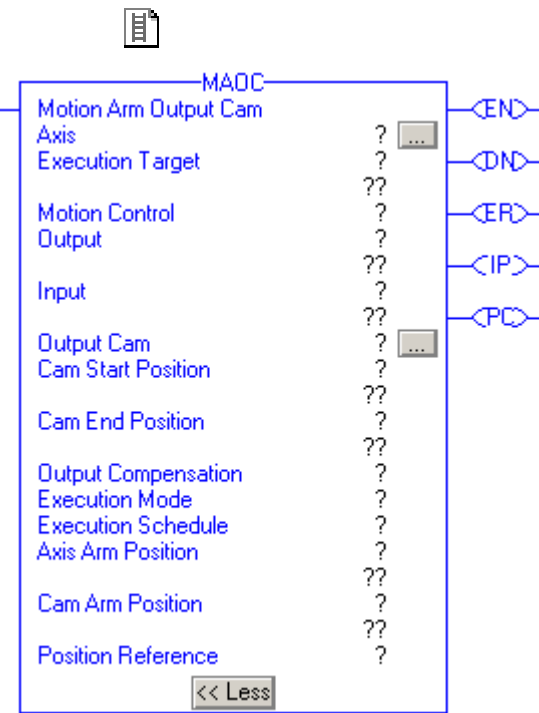
开启运动凸轮输出 (MAOC) 指令在指定的轴和输出之间发起特定凸轮输出的准备。当执行时，使用由 RSLogix 5000 凸轮输出编辑器建立的凸轮输出曲线将指定的凸轮输出位与指定的轴同步。这个关系可看做主/从关系，其中轴代表主角色，输出位代表从角色。因此，凸轮输出功能与位置凸轮功能相关，该功能提供主轴与从轴之间的某种关系。要准确地将凸轮输出与指定的轴同步，指定了一个执行计划以及相关的轴和凸轮准备位置。当轴按照由执行计划参数指定的方向行进通过轴准备位置时，凸轮位置即锁定轴位置，从指定的凸轮准备位置参数开始。这时，凸轮输出被准备，凸轮输出准备状态被设置。凸轮输出还可以通过执行计划参数进行配置，以立即或挂起执行当前正在执行的凸轮输出的完成。凸轮输出也可以执行一次、连续执行或持续执行，通过指定所需的执行模式来设置。持续行为可让凸轮输出在当凸轮位置超过凸轮输出范围时被禁止，当凸轮返回范围之内时重新准备。凸轮输出范围的定义是通过输入参数 **CamStartPosition** 和 **CamEndPosition** 来实现的。主参考选择可让轴输入从指定轴的实际或命令位置中得到。

注意



凸轮输出增加了超过大刷新速率的可能性。如果运动任务执行时间超过所配置的组的近似刷新周期，这可能会导致错误行为。对这种情况进行检查的唯一方式是从运动控制组属性页面上监视最大执行时间。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_CONSUMED AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	将位置输入提供给凸轮输出的轴的名称。省略号将启动轴属性对话框。
执行目标	UINT32	立即或标签	执行目标定义了连接到命名轴的集中的特定凸轮输出。行为是由以下方面确定的： <ul style="list-style-type: none"> ² 0...8 ñ 在 Logix 控制器中执行的凸轮输出。 ² 9...31 ñ 预留为将来使用。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
输出	DINT	标签	32 个输出位的集合，根据指定的凸轮输出而被复位或设置。它可以是一个内存位置或者物理输出。如果执行计划中选择了挂起，则输出被忽略。
输入	DINT	标签	32 个输出位的集合，可根据指定的凸轮输出被用作启用位。它可以是一个内存位置或者物理输入。如果执行计划中选择了挂起，则输入被忽略。
凸轮输出	OUTPUT_CAM	数组标签	OUTPUT_CAM 元素数组。元素不需要进行排序，数组大小是由指定的凸轮元素个数确定的。 数组大小受到 Logix 控制器的可用内存的限制。

操作数	类型	格式	描述
凸轮起始位置	SINT、INT、DINT 或 REAL	立即或标签	凸轮起始位置与凸轮终止位置一起定义了凸轮输出范围的左右边界。
凸轮终止位置	SINT、INT、DINT 或 REAL	立即或标签	凸轮终止位置与凸轮起始位置一起定义了凸轮输出范围的左右边界。
输出补偿	OUTPUT_COMPENSATION	数组标签	1 到 32 个 OUTPUT_COMPENSATION 元素的数组。该数组索引对应于输出位编号。数组的最小大小是由补偿的最高输出位决定的。
执行模式	UINT32	立即	共有三种 (3) 可能的执行模式。行为是由所选择的模式决定的。选项如下： 0 = 一次 – 当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出被禁止，运动控制指令的过程完成位被设置。 1 = 连续 – 当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出在凸轮输出范围的相对一侧继续。 2 = 持续 – 当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出被禁止。当凸轮位置重新回到输入凸轮范围内时，凸轮输出被重新准备。

操作数	类型	格式	描述
执行计划	UINT32	立即	<p>选择何时准备凸轮输出。 选项如下：</p> <p>0 = 立即 – 凸轮输出立即被准备。</p> <p>1 = 挂起 – 当正在执行的凸轮输出的凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出被准备。当选择挂起时，以下参数将被忽略：输出、输入、轴准备位置和参考。</p> <p>2 = 仅正向 – 当轴从正的方向接近或穿过指定的轴准备位置时，凸轮输出被准备。</p> <p>3 = 仅反向 – 当轴从反的方向接近或穿过指定的轴准备位置时，凸轮输出被准备。</p> <p>4 = 双向 – 当轴从任一方向接近或穿过指定的轴准备位置时，凸轮输出被准备。</p>

操作数	类型	格式	描述
轴准备位置	SINT、INT、DINT 或 REAL	立即或标签	当执行计划被设置为仅正向、仅反向或双向而且轴按照指定的方向运动时，它定义了凸轮输出被准备处的轴位置。如果执行计划中选择了挂起，则轴准备位置被忽略。
凸轮准备位置	SINT、INT、DINT 或 REAL	立即或标签	它定义了当凸轮输出被准备时与轴准备位置相关的凸轮位置。
参考	UINT32	立即	<p>它设置了凸轮输出是连接到轴的命令位置还是连接到轴的实际位置。如果执行计划中选择了挂起，则参考被忽略。</p> <p>0 = 实际 – 按照其编码器或其他反馈设备的测量的轴的当前位置。</p> <p>1 = 命令 – 所需要的或命名的主轴位置。</p>



MAOC(Axis,ExecutionTarget, MotionControl,Output,Input, OutputCam,CamStartPosition, CamEndPosition, OutputCompensation, ExecutionMode, ExecutionSchedule, AxisArmPosition, CamArmPosition,Reference);

结构文本

操作数与梯形图 MAOC 指令的操作数相同。对于数组操作数，您不需要包含数组索引。如果您不包含数组索引，则指令从数组 ([0]) 中的第一个元素开始。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
ExecutionMode	一次	0
	连续	1
	持续	2
ExecutionSchedule	立即	0
	pending	1
	forwardonly	2
	reverseonly	3
	bidirectional	4
参考	实际	0
	命令	1

MAOC 指令

有效的凸轮准备位置可以是在凸轮起始位置与凸轮终止位置之间（包含）的任何位置。如果凸轮准备位置被设置为等于（或非常接近于）凸轮起始位置或凸轮终止位置的值，补偿可能会将凸轮位置超出凸轮起始位置与凸轮终止位置之间的范围。补偿受到为位置偏移、锁存延迟和解锁存延迟指定的输出补偿值的影响，还包括根据 MAOC 指令的参考和输出参数所应用的内部补偿值。

如果 MAOC 指令配置了执行模式“连续”或“持续”，不会产生什么负面作用，而且当凸轮输出被准备以及轴运动时，不会存在挂起的 MAOC 指令。

如果 MAOC 指令配置了执行模式“仅一次”，则会产生以下负面作用，而且当凸轮输出被准备以及轴运动时，会存在挂起的 MAOC 指令。

- 一个或多个输出可能永远不会改变状态。
- MAOC 指令可能立即完成。

当凸轮输出被准备以及轴运动时存在挂起的 MAOC 指令的可能负面作用是，一个或多个输出可能会根据挂起的 MAOC 指令的配置而开始执行。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到梯级变为假为止。
.DN (完成) 位 29	当凸轮输出成功发起时, 完成位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	当凸轮输出被成功发起时, 这个位被设置; 如果被另一个开启运动凸轮输出命令所取代、被关闭运动凸轮输出命令所终止, 或者当执行模式被设置为“一次”时凸轮位置移动超出定义的凸轮输出范围, 则这个位被清除。
.PC(过程完成) 位 27	这个位在正的梯级过渡时被清除, 在“一次”执行模式下当凸轮位置移动超出定义的凸轮输出范围时, 这个位被设置。
.SEGMENT	它被设置到与错误 36 (非法凸轮输出) 或错误 37 (非法输出补偿) 相关联的数组索引。多个错误中只有第一个被储存。

说明: 开启运动凸轮输出 (MAOC) 指令执行通过人工设置、程序程序或通过 RSLogix 5000 凸轮输出编辑器所建立的凸轮输出曲线。在内部, 凸轮输出对象处理运动轨迹规划凸轮功能。每个凸轮输出对象负责一个输出, 其中包含 32 个输出位。每一个输出位都可以单独进行编程。目前, 凸轮输出功能在 Logix 控制器内为每个过程刷新周期执行一次 (目前可在 1 到 32 毫秒之间配置)。

轴

该轴将位置输入提供给凸轮输出。这个轴可以是虚拟的、实际的或用过的。

执行目标

执行目标定义了连接到指定轴的集中的特定凸轮输出。目前, 只能指定八个凸轮输出。

指定凸轮输出曲线

要执行 MAOC 指令，必须指定一个计算的凸轮输出数据数组标签。凸轮输出数组标签可以通过 RSLogix 5000 标签编辑器来创建，或者通过 MAOC 指令使用内置的凸轮输出编辑器来创建。该数据定义了每个凸轮输出元素的具体参数。凸轮输出元素的个数受到可用内存数量的限制。对于每个输出位可定义零个或多个凸轮。对于这些元素在凸轮输出数组内如何排列并无限制。

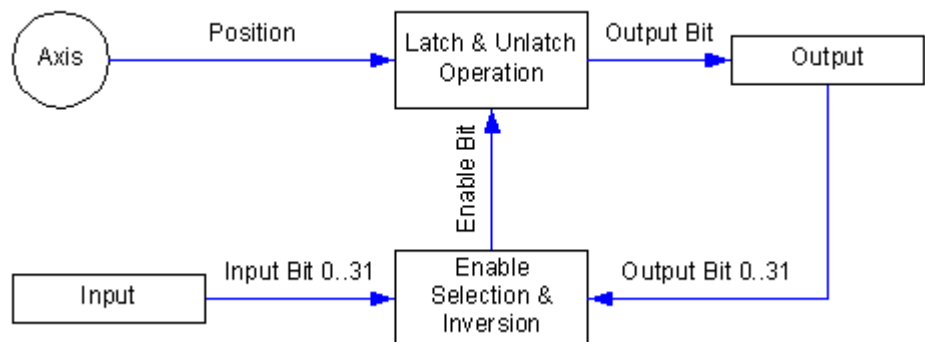
请参阅 OUTPUT_CAM 结构的描述，了解有关数据类型和编程单位的信息。

重要提示

当凸轮输出打开窗口位置被重新定义而被凸轮输出元素控制的输出正在活动时，会出现异常。在某些情况下，运动轨迹规划可能探测不到窗口被穿越，被凸轮输出元素控制的输出仍保持开启状态。

这个问题适用于受到 MAOC 指令控制的任何输出点或虚拟输出。

下图显示了由凸轮输出元素定义的轴、输入和输出之间的关系。



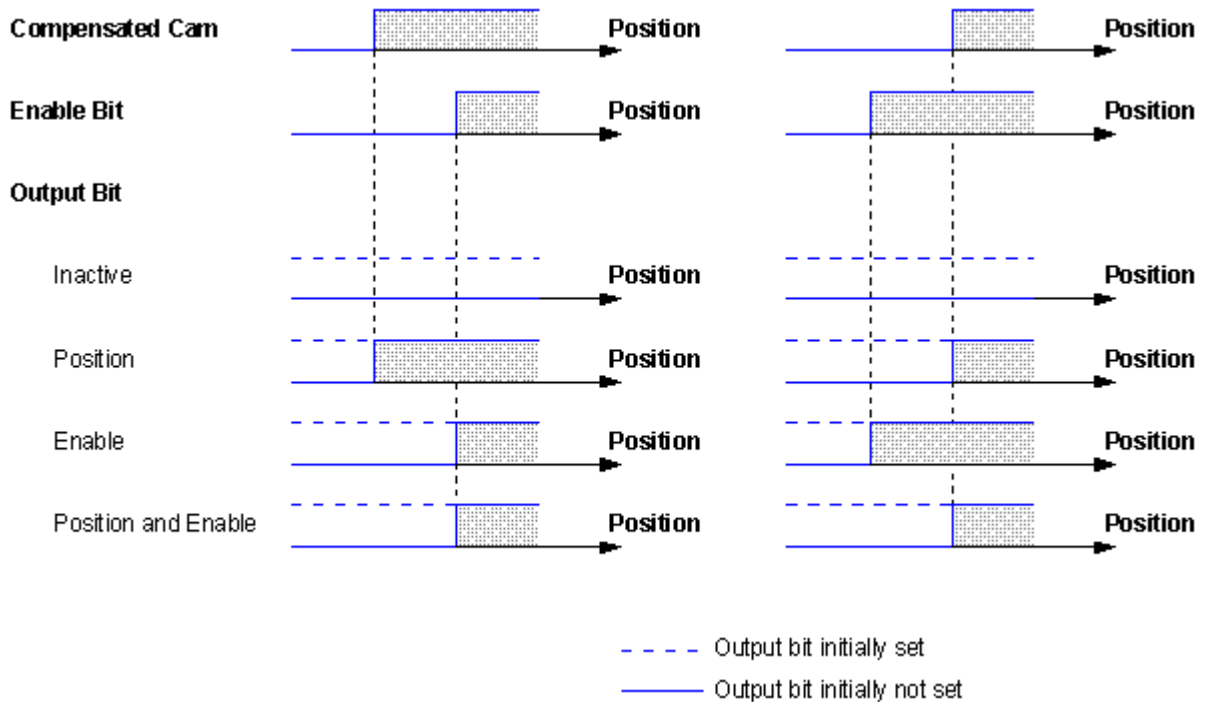
凸轮输出元素关系

锁存类型

根据所选择的锁存类型，相应的输出位将按照下表进行设置。

锁存类型	行为
无效	输出位不被更改。
位置	当轴进入被补偿的凸轮范围时，输出位被设置。
使能	当启用位变为活跃时，输出位被设置。
位置和使能	当轴进入被补偿的凸轮范围而且启用位变为活跃时，输出位被设置。

下图显示对于不同的补偿凸轮和启用位组合作为位置的函数，所选定的锁存类型对输出位的影响。



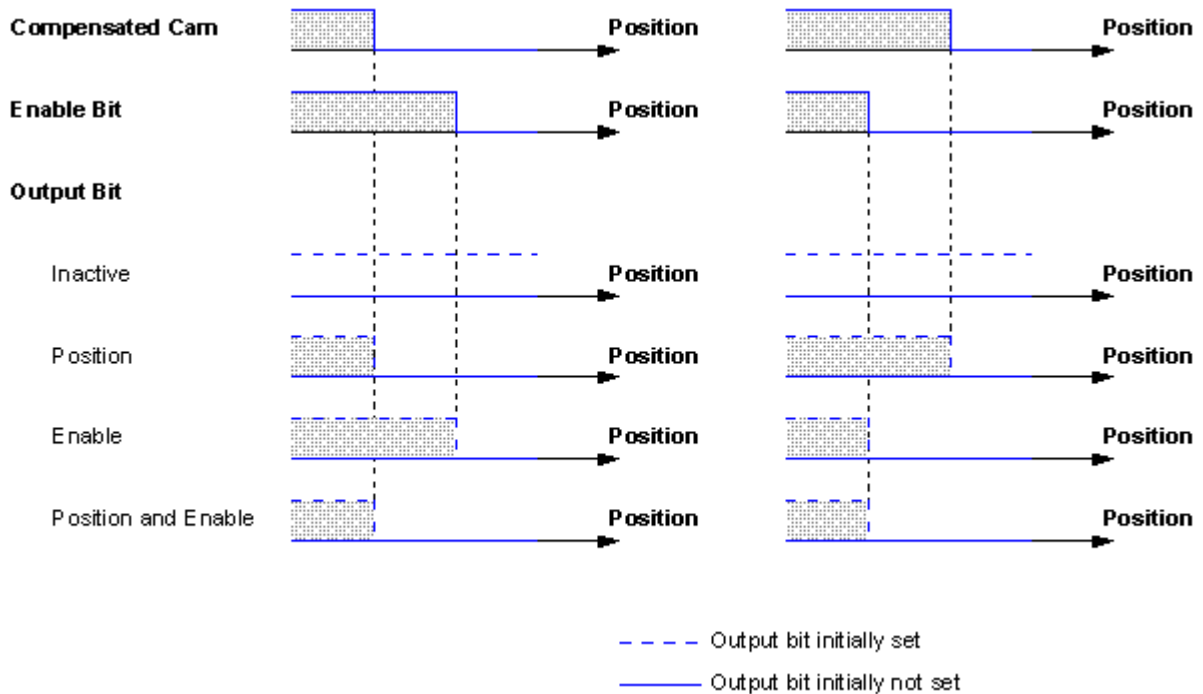
锁存位置

解锁存类型

根据所选择的解锁存类型，相应的输出位将按照下表进行复位。

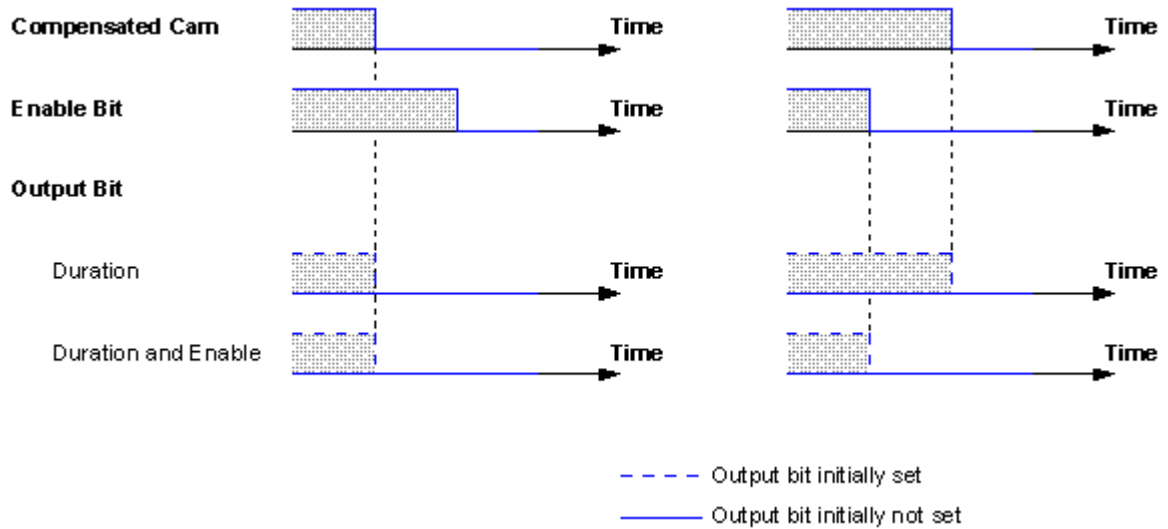
解锁存类型	行为
无效	输出位不被更改。
位置	当轴离开被补偿的凸轮范围时，输出位被复位。
持续时间	当持续时间结束时，输出位被复位。
使能	当启用位变为不活跃时，输出位被复位。
位置和使能	当轴离开被补偿的凸轮范围或者启用位变为不活跃时，输出位被复位。
持续时间和使能	当持续时间结束或者启用位变为不活跃时，输出位被复位。

下图显示对于不同的补偿凸轮和启用位组合作为位置的函数，所选定的解锁存类型对输出位的影响。



解锁存作为位置的函数

以及时间的函数。



解锁存作为时间的函数

左右凸轮位置

左右凸轮位置定义了凸轮输出元素的范围。如果锁存或解锁存类型被设置为“位置”或“位置和使能”而且启用位为活跃状态，则左右凸轮位置指定了输出位的锁存或解锁存位置。

持续时间

如果解锁存类型被设置为“持续时间”或“持续时间和使能”而且启用位为活跃状态，则凸轮持续时间指定了输出位的锁存和解锁存之间的时间。

使能类型

根据所选择的使能类型，启用位是输入、反转输入、输出或反转输出的一个元素。

凸轮输出数组检查

如果您选择的输出位小于 0 或大于 31，凸轮输出元素不被考虑，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的锁存类型小于 0 或大于 3，则使用值“不活跃”，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的解锁存类型小于 0 或大于 5，则使用值“不活跃”，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的左凸轮位置大于或等于右凸轮位置，而且锁存或解锁存类型被设置为“位置”或“位置和使能”，则凸轮输出元素不被考虑，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的左凸轮位置小于凸轮起始位置，而且锁存类型被设置为“位置”或“位置和使能”，则使用凸轮起始位置，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的右凸轮位置大于凸轮终止位置，而且解锁存类型被设置为“位置”或“位置和使能”，则使用凸轮终止位置，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的持续时间小于或等于 0，而且解锁存类型被设置为“持续时间”或“持续时间和使能”，则凸轮输出元素不被考虑，并警告用户出现指令错误“非法凸轮输出”。

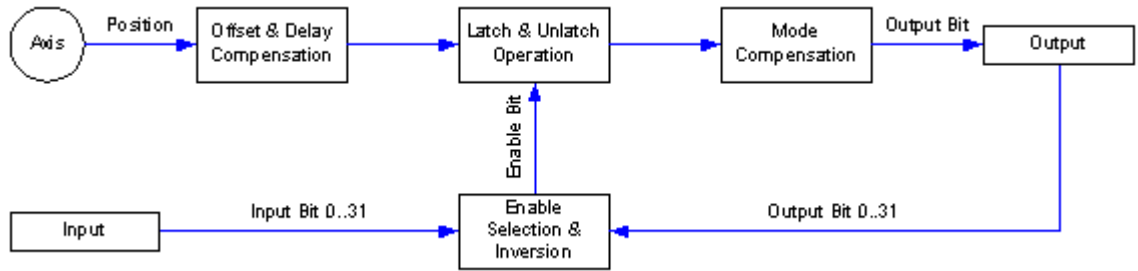
如果您选择的使能类型小于 0 或大于 3，而且锁存或解锁存类型被设置为“使能”、“位置和使能”或“持续时间和使能”，则凸轮输出元素不被考虑，并警告用户出现指令错误“非法凸轮输出”。

如果您选择的启用位小于 0 或大于 31，而且锁存或解锁存类型被设置为“使能”、“位置和使能”或“持续时间和使能”，则凸轮输出元素不被考虑，并警告用户出现指令错误“非法凸轮输出”。

指定输出补偿

输出补偿数据数组标签可以通过 RSLogix 5000 标签编辑器来指定。数据类型通过指定每个执行机构的特征，来定义每个输出位的具体参数。该数组索引对应于输出位编号。被补偿的最高输出位的数字定义的是此数组的最小大小。对输出补偿的更改将立即生效。

下图显示了输出补偿对轴、输入和输出之间关系的影响。

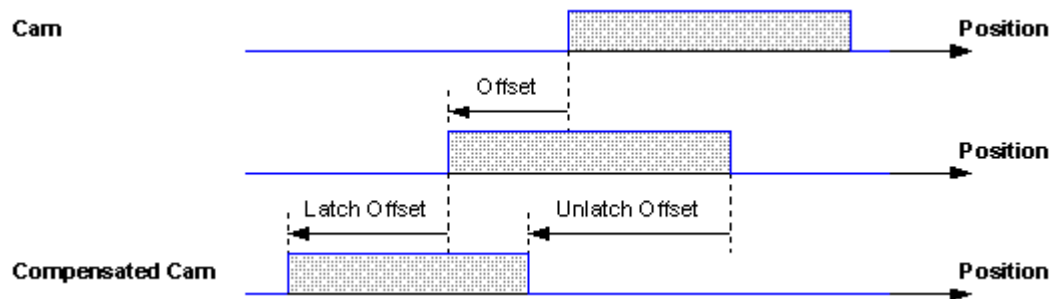


输出补偿

请参阅 OUTPUT_COMPENSATION 结构的描述，了解有关数据类型和编程单位的信息。

偏移和延迟补偿

偏移提供位置补偿，而锁存和解锁延迟则为锁存和解锁操作提供时间延迟补偿。下图显示了补偿值对凸轮输出元素的影响。



偏移和延迟补偿

凸轮范围是由凸轮输出元素的左右凸轮位置定义的。补偿凸轮范围是由凸轮范围、偏移和锁存及解锁存偏移定义的。锁存和解锁存偏移是由当前速度 v 所定义的：

$$\begin{aligned} \text{锁存偏移} &= v * \text{锁存延迟} \\ \text{解锁存偏移} &= v * \text{解锁存延迟} \end{aligned}$$

所得到的补偿偏移可能实际上大于凸轮起始位置和凸轮终止位置之差。

下面的公式显示了补偿值对凸轮输出元素持续时间的影响。

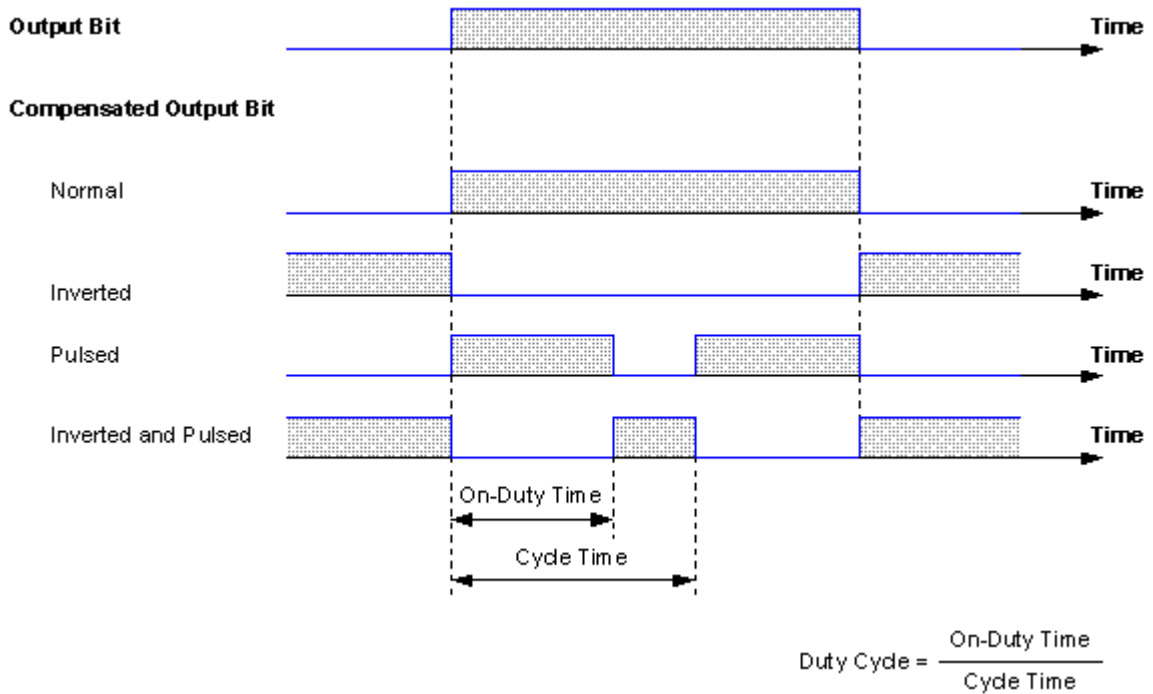
$$\text{补偿的持续时间} = \text{持续时间} + \text{锁存延迟} - \text{解锁存延迟}$$

模式补偿

根据所选择的模式，补偿的输出位将按照下表进行设置。

模式	行为
正常	<p>当锁存和解锁存操作的输出变得活跃时，输出位被设置。</p> <p>当锁存和解锁存操作的输出变得不活跃时，输出位被复位。</p>
取反	<p>当锁存和解锁存操作的输出变得不活跃时，输出位被设置。</p> <p>当锁存和解锁存操作的输出变得活跃时，输出位被复位。</p>
脉冲调制	<p>当锁存和解锁存操作的输出为活跃时，输出位被脉冲调制。脉冲的占有状态对应于输出位的活跃状态。</p> <p>当锁存和解锁存操作的输出变得不活跃时，输出位被复位。</p>
取反和脉冲调制	<p>当锁存和解锁存操作的输出为活跃时，输出位被脉冲调制。脉冲的占有状态对应于输出位的不活跃状态。</p> <p>当锁存和解锁存操作的输出变得不活跃时，输出位被设置。</p>

下图显示了模式、循环时间和占空比对输出位的影响。



模式补偿

输出补偿数组检查

如果您选择的锁存和解锁延迟补偿导致的补偿凸轮小于最小宽度，则补偿凸轮的宽度被设置为最小值。

如果您选择的模式小于 0 或大于 3，则考虑“正常”模式，并警告用户出现指令错误“非法输出补偿”。

如果您选择的占空比小于 0 或大于 100，而且模式被设置为“脉冲调制”或“取反和脉冲调制”，则考虑 0 或 100 占空比，并警告用户出现指令错误“非法输出补偿”。

如果您选择的循环时间小于或等于 0，而且模式被设置为“脉冲调制”或“取反和脉冲调制”，则输出位不被脉冲调制，并警告用户出现指令错误“非法输出补偿”。

输出

输出是 32 个输出位的集合，可根据指定的凸轮输出被设置和复位。输出可以是一个内存位置或者物理输出（例如 Local.O.O.Data）。

输入

输入是 32 个输入位的集合，可根据指定的凸轮输出被用作启用位。输入可以是一个内存位置或者物理输入（例如 Local.0.I.Data）。

凸轮起始位置和凸轮终止位置

凸轮起始位置与凸轮终止位置一起定义了凸轮输出范围的左右边界。当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出的行为由执行模式和执行计划来定义。对凸轮起始位置和凸轮终止位置的更改直到当前的 MAOC 指令完成以后才会生效。

执行模式

根据所选择的执行模式，当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出的行为可能有所不同。

执行模式	行为
一次	当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出被禁止，运动控制指令的过程完成位被设置。
持续	当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出被禁止。但是，当凸轮位置重新回到输入凸轮范围内时，凸轮输出被重新准备。
连续	当凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出在凸轮输出范围的相对一侧继续。

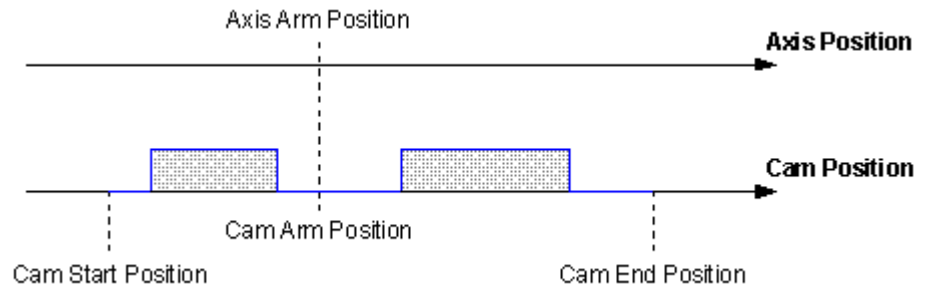
执行计划

根据所选择的执行计划，凸轮输出按照下表所示被准备。

执行计划	行为
立即	凸轮输出被立即准备。
挂起	当已准备的凸轮输出的凸轮位置移动超过凸轮起始位置或凸轮终止位置时，凸轮输出被准备。
仅正向	当轴从正的方向接近或穿过指定的轴准备位置时，凸轮输出被准备。
仅反向	当轴从反的方向接近或穿过指定的轴准备位置时，凸轮输出被准备。
双向	当轴从正的或反的方向接近或穿过指定的轴准备位置时，凸轮输出被准备。

轴准备和凸轮准备位置

当执行计划被设置为仅正向、仅反向或双向而且轴按照指定的方向运动时，轴准备位置定义了凸轮输出被准备处的轴位置。凸轮准备位置定义了当凸轮输出被准备时，与轴准备位置相关的凸轮准备位置。对轴准备位置或凸轮准备位置的更改直到当前的 MAOC 指令执行以后才会生效。



轴准备和凸轮准备位置

参考

根据所选择的参考，凸轮输出被连接到轴实际位置或命令位置。

重要提示

MAOC 指令执行在一次扫描中完成，因此完成 (.DN) 位和正在进行 (.IP) 位被立即设置。正在进行 (.IP) 位将保持设置状态，直到凸轮位置移动超过凸轮起始位置或凸轮终止位置（在“一次”执行模式中）、被另一条 MAOC 指令所取代或者被 MDOC 指令禁止为止。当 MAOC 执行时，过程完成位被立即清除，当凸轮位置移动超过凸轮起始位置或凸轮终止位置（在“一次”执行模式中）时被设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码： MAOC 错误代码 (.ERR)

错误消息	代码	描述
执行冲突	3	尝试与另一个当前正在进行的凸轮输出一起执行。
关闭状态错误	7	尝试执行处于关闭状态的轴。
轴未配置	11	传递的轴值引用一个未配置的轴，即轴尚未被指定给某个物理运动模块通道。
值超出范围	13	尝试使用超出范围的输入参数来执行。 1. 凸轮起始位置凸轮终止位置。 2. 凸轮准备位置在凸轮输出范围之外。 参见扩展错误部分，了解有关错误原因的更多信息。
归零过程中错误	16	尝试使用一个正在进行的归零过程来执行。
轴组未同步	19	尝试执行的轴所在的轴组内目前尚未同步。
轴为故障状态	20	尝试执行处于故障状态的轴。
组为故障状态	21	尝试执行的轴所在的某个组正处于故障状态。
非法动态更改	23	尝试对动态参数进行非法更改，例如在 S 曲线上合并、在进行中将模式从捕俘更改为 S 曲线、将 S 曲线更改为非零速度或者更改 S 曲线的加速度。
非法控制器模式操作	24	当处理器处于测试模式时尝试执行
非法执行目标	35	尝试执行的指定凸轮输出不被 Logix 控制器支持。

错误消息	代码	描述
非法凸轮输出	36	<p>尝试执行的凸轮输出数组包含至少一个成员超出范围：</p> <ol style="list-style-type: none"> 1. 输出位小于 0 或者大于 31。 2. 无效的锁存类型值。 3. 无效的解锁存类型值。 4. $左 \geq 右$，当锁存类型被设置为“位置”或“位置和使能”时。 5. $左 < 凸轮起始位置$，当锁存类型被设置为“位置”或“位置和使能”时。 6. $右 > 凸轮终止位置$，当解锁存类型被设置为“位置”或“位置和使能”时。 7. $持续时间 \leq 0$，当解锁存类型被设置为“持续时间”或“持续时间和使能”时。 8. 使能类型值无效，当锁存类型或解锁存类型被设置为“使能”或“持续时间和使能”时。 9. 启用位值无效，当锁存类型或解锁存类型被设置为“使能”、“位置和使能”或“持续时间和使能”时。
非法输出补偿	37	<p>尝试执行的凸轮输出数组包含至少一个成员超出范围：</p> <ol style="list-style-type: none"> 1. 无效的模式值。 2. $循环时间 \leq 0$，当模式被设置为“脉冲调制”或“取反和脉冲调制”时。 3. 占空比小于 0 或大于 100，当模式被设置为“脉冲调制”或“取反和脉冲调制”时。
MAOC 无效输出操作数	80	<p>ExErr#1: 无效的数据标签引用 - 输出操作数没有指向模块输出数据标签的 O.Data 元素。</p> <p>ExErr#2: 无效的 OB16IS 模块通讯格式 - OB16IS 通讯格式已从默认的“按点计划输出数据”更改为别的格式。</p>

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MAOC 指令，扩展错误代码 4 指的是输出操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

状态位： *MAOC 对状态位的影响*

状态位可以用于确定是否可以发起 MAOC 指令。MAOC 指令影响到运动轴结构中的以下状态字：

- OutputCamStatus
- OutputCamPendingStatus
- OutputCamLockStatus
- OutputCamTransitionStatus

如果执行计划被设置为仅正向、仅反向或者双向，则当以下任何两种条件成立时即可发起 MAOC 指令：

- OutputCamStatus 位 = FALSE

或

- OutputCamStatus 位 = TRUE
OutputCamLockStatus 位 = FALSE
OutputCamTransitionStatus 位 = FALSE

如果执行计划被设置为挂起，则当以下任何两种条件成立时即可发起 MAOC 指令：

- OutputCamStatus 位 = FALSE

或

- OutputCamStatus 位 = TRUE
OutputCamTransitionStatus 位 = FALSE

轴和模块故障条件禁止凸轮输出

当控制器监测到下面一种故障时，将会禁止凸轮输出：

- 对于 Axis_Servo 和 Axis_Servo_Drive：轴反馈丢失故障
- 对于 Axis_Servo 和 Axis_Servo_Drive：模块故障
- 对于 Axis_Consumed：物理轴故障

这些故障产生不可靠的反馈数据。

同时，当 MAOC 指令被发起时如果存在轴故障，则指令出错。

规划输出模块

1756-OB16IS 规划输出模块的设计与运动轴凸轮输出 (MAOC) 运动指令一起使用，来提供基于位置的输出控制（也称为 PLS）。MAOC 指令自身允许基于位置的输出控制，使用 ControlLogix 中的任何运动轴作为位置参考以及任何输出或布尔值作为输出。MAOC 根据运动轴位置按照运动组大的刷新速率（通常为 2ms-10ms）来更新输出。虽然这对于部分应用来说是足够了，但对于通常在转换和包装领域内出现的很多高速应用来说是太慢了。1756-OB16IS 模块支持对其 16 位输出中的 8 个输出打开 / 关闭时间（输出 0-7）进行计划（使用 100 μ s 为增量），从而提高了性能。输出计划的方法是，将数据输入一个或多个由输出连接数据存储单元所提供的 16 个计划中。

操作 此处定义的计划输出不应与较早的计划输出实现相混淆。以前的实现以模块为单位计划输出，所有输出点都由单个时间戳进行控制。现在的实现以点为单位计划输出，每一个输出点由其自己的时间戳进行控制。

各个计划在控制器内创建，存储在该模块的输出映像表中，并通过背板发送到规划输出模块。计划指定了一个序列计数、与计划关联的输出点、输出值应该应用到物理输出点的时间、以及在计划的时间要应用的值。I/O 模块接收并存储计划。每个计划的 CST 时间戳受到模块的监视。当某个计划到期后，即当前时间符合计划的时间戳，则输出值被应用到相应的输出位。ASIC 中的定时器硬件用于优化计划算法。这个硬件还减少了时延和抖动性能。每个计划的状态将在输出回波连接中报告，反映在模块的输入映像中。

计划输出功能依赖于 CST(协调系统时间) 时间戳。机架内至少要有一个控制器必须为 CST 时间主机。

未用的输出可用作正常输出，将立即应用而不需要等待 CST 时间戳到期。一个屏蔽码被发送到模块，用来指示哪些输出将作为正常输出。

规划输出模块支持最多八个可以单独进行计划的输出。计划输出必须在输出点 0 到 7 之间。1756-OB16IS 支持最多 16 个计划，其中每个输出两个计划。未“计划”的输出将用作正常输出点。使用屏蔽码来指示哪些点是计划的，哪些点是非计划的。抖动和时延性能小于 100 微秒。所有计划配置都通过 MAOC 指令来完成。

在当前计划尚未过期之前，如果 I/O 模块收到由序列计数的变化所指示的某个新计划，则当前计划将被覆盖。这种机制可用于取消正在活跃的计划。输出回波连接中返回的状态位可用于确定每个计划的当前状态以及触发相应的事件任务。

如果控制器发送一个新计划而 CST 时间戳已成过去，输出将被维持，直到 CST 时间完全回绕。模块不检查过期的 CST 时间戳。

警告



如果两个计划之间的时间小于最小计划间隔（例如 100 μ s），则出现抖动。这意味着即使两个输出在不同时间计划（例如时间 90 和时间 110），他们将会同时激活（例如时间 90）。（最小计划间隔设置不应快于 100 μ s。）

远程操作 使用 1756-OB16IS 模块的计划输出不适用于远程机架。

MAOC 指令的用法 当使用运动和 MAOC 指令时，输出映像中的值被控制器内的运动轨迹规划固件所控制。运动轨迹规划触发数据发送到模块。然而，正常的程序 / 任务扫描也会触发数据发送到模块。数据完整性由固件维护，固件始终为最后一个给定的计划设置序列计数。

凸轮输出指令处理计划输出的凸轮事件，要比非计划的输出快一个大的刷新周期。当检测到编程打开或关闭事件时，将向输出模块发送一个计划，以在下一个近似刷新周期内的合适时间打开 / 关闭输出。凸轮输出指令将近似刷新周期分成十六个时隙。

例如，如果近似刷新周期为 2 毫秒，则产生十六个 125 微秒的时隙。凸轮开 / 关事件将会按照它们在近似刷新周期中的位置被分配到各个时隙中。如果某个凸轮元素的锁存和解锁存事件被分配到同一个时隙，则会彼此抵消。这暗示着凸轮元素的最小脉冲宽度大于一个时隙。

一个凸轮元素的最小脉冲宽度应大于 100 微秒的 OB16IS 最小脉冲宽度，或者大更新最小脉冲宽度的 1/16，以较大者为准。

重要提示

1756-OB16IS 规划输出模块只能与一个 (1)MAOC 轴 / 执行目标关联。

MAOC 探测一个近似刷新周期前的锁存和解锁存事件，并计划该事件在下一个大的更新内发生。其实现方式是向每个计划的输出锁存和解锁存位置应用一个大的更新内部延迟。当检测到锁存或解锁存事件时，将计算从大更新开始到该事件的时间变化，并在下一个近似刷新周期所对应的协调系统时间计划输出。为方便这一点，凸轮输出功能可以访问当发生当前近似刷新周期时所捕获到的 CST。

MAO 能够处理 1756-OB16IS 的计划和计划输出位。

MAOC 分配所有八个计划输出供运动轨迹规划凸轮输出唯一使用。MAOC 在每个近似刷新周期将屏蔽码字段设置为 0xff，以防用户尝试更改它。这其中含义是，用户无法直接影响输出位 0-7，但的确可以修改输出位 8-15。

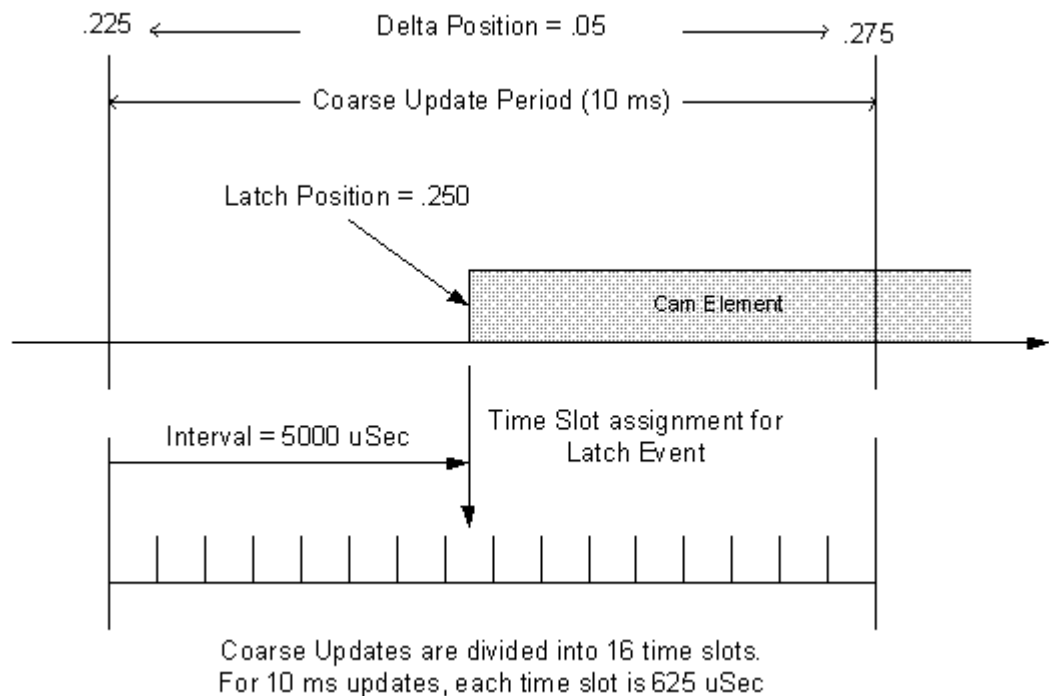
重要提示

通过将数据位强制设置为 0 或 1、将 ScheduleMask 中的相应位强制设置为 0 来强制设置输出 0-7。对于输出 8-15，只需要强制设置数据位即可。

由于 1756-OB16IS 仅支持十六个计划的限制，对每个近似刷新周期可以处理的事件个数也有限制。

每个近似刷新周期内仅有八个计划可用。这允许两个连续的大更新，其中每个更新包含八个输出事件。当前正被 1756-OB16IS 处理的八个计划作为一个组，第二个八个计划组可同时为下一个大更新设置。

下图显示了近似刷新周期、凸轮锁存事件和时隙之间的关系。



近似刷新周期、凸轮锁存和时隙之间的相互关系

每个时隙存储以下信息：

锁存事件屏蔽码 – 当检测到锁存事件后，它所属的时隙被计算出来，与锁存输出位相对应的锁存事件屏蔽码中的位被设置。

解锁存事件屏蔽码 – 当检测到解锁存事件后，它所属的时隙被计算出来，与解锁存输出位相对应的解锁存事件屏蔽码中的位被设置。

间隔 – 从发生锁存或解锁存事件的大周期的开始以来的时间，单位是微秒。

脉冲开启屏蔽码 – 对于脉冲调制输出，事件脉冲所在的时隙被计算出来，与脉冲事件的输出位相对应的脉冲开启屏蔽码中的位被设置。

脉冲关闭屏蔽码 – 对于脉冲调制输出，事件脉冲所在的时隙被计算出来，与脉冲事件的输出位相对应的脉冲关闭屏蔽码中的位被设置。

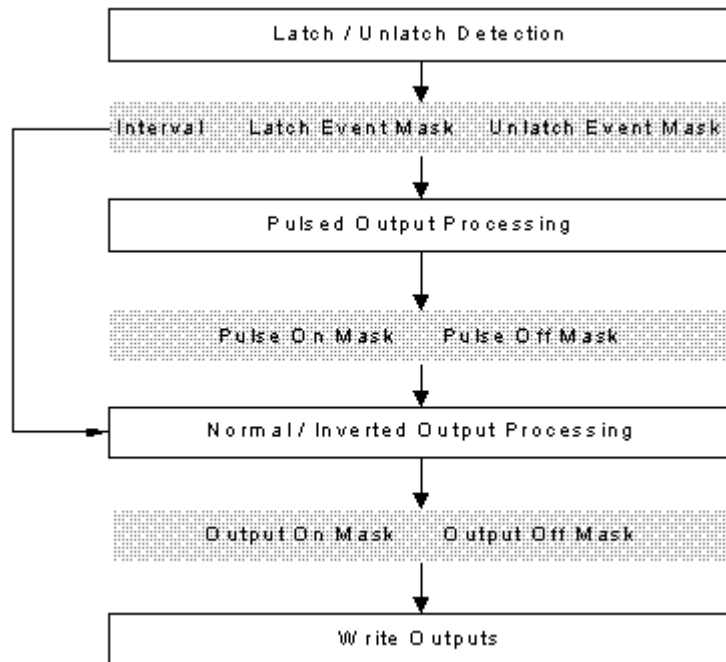
输出开启屏蔽码 – 对于正常输出，与锁存或脉冲开启时间的输出位相对应的位被设置，表示输出将为该事件打开。

对于取反输出，与解锁存或脉冲关闭时间的输出位相对应的位被设置，表示输出将为该事件打开。

输出开启屏蔽码 – 对于正常输出，与锁存或脉冲开启时间的输出位相对应的位被设置，表示输出将为该事件打开。

对于取反输出，与锁存或脉冲开启时间的输出位相对应的位被设置，表示输出将为该事件关闭。

下面是如何使用时隙的简要说明。



时隙数据使用概述

时隙也用于处理重叠的凸轮元素。将维护一个旗语来指示每个输出位的当前活跃状态。此外，如果编程凸轮元素的锁存和解锁存事件在同一时隙内发生，则彼此抵消。

I/O 子系统

用户可将 MAOC 指令的输出参数指定为内存标签或者输出模块的数据标签。标签的指针将传递到 MAOC 指令中。同时传递到 MAOC 的还有 IO_MAP 类型的内部参数。如果输出参数引用控制器内存，则 IO_MAP 参数为 NULL。如果输出参数引用某个输出模块，则 IO_MAP 参数指向模块的映射结构。然后，MAOC 指令通过检查变频器表格中存储的模块类型，可以确定输出参数是否关联某个 1756-OB16IS 模块。

输出数据结构

区	规格	描述
值	4 字节	非计划输出位的数据值。 0 = 关闭 1 = 打开
屏蔽码	4 字节	选择哪些位将被计划。 只有前八位 (0-7) 可以被计划。 0 = 非计划 1 = 计划

16 个计划结构的数组

区	规格	描述
计划 ID	1 字节	有效的 ID 为 1-16。任何其他值将表示计划不会被考虑。
序列号	1 字节	OB16IS 将维护计划的副本。序列号更改将会告诉 OB16IS 来处理这个计划中的数据。
点 ID	1 字节	表示与此计划关联的输出位。输入为值 00-07。
点值	1 字节	点 ID 中指定的输出位的下一个状态。 0 = 关闭 1 = 打开
时间戳 (Time Stamp)	4 字节	CST 的低 32 位。表示何时更改指定输出位的状态。

计划处理 值和屏蔽码字段被处理，所有非排定型数据位被移动到模块输出数据存储单元内。当计划被处理以后，这个数据被写入输出端子。

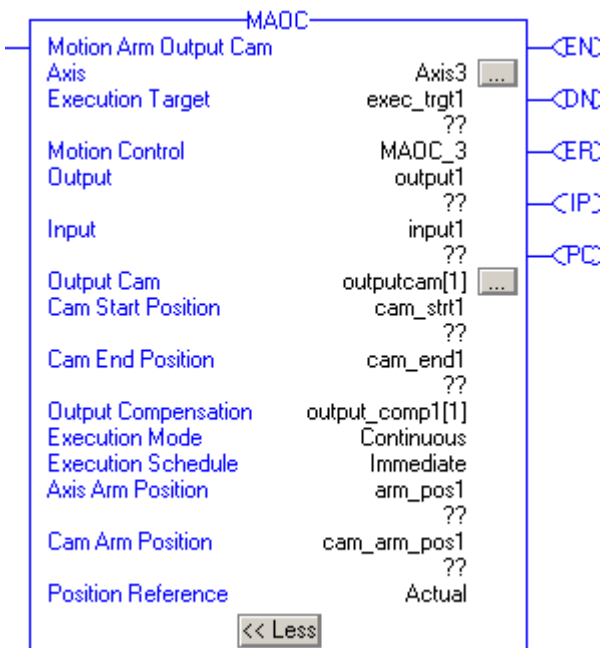
每个计划都被处理。在下列情况下计划将不被考虑：

- 计划 ID 不在 1 - 16 范围之内
- 点 ID 不在 0 - 7 范围之内
- 序列号没有变化

如果计划将被考虑，将会标记为“活跃”。

所有“活跃”的计划将会每 200 微秒检查一次。计划的时间戳与当前 CST 进行比较。如果当前 CST 大于或等于计划的时间戳，计划中的点值被移动到模块输出数据存储单元中用于指定的输出位。

M 示例： 梯形图



MAOC 梯形图示例

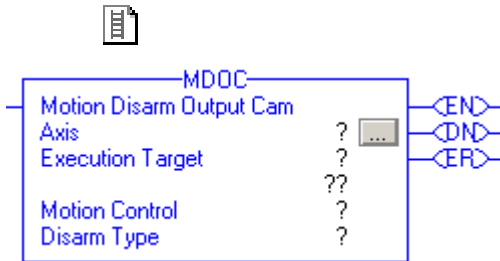
结构文本

```
MAOC(Axis3,exec_trgt1,MAOC_3,output1,
input1,outputcam1[1],cam_strt1,cam_end1,
output_comp1[1],continuous,immediate,arm_pos1,
cam_arm_pos1,actual);
```

关闭运动凸轮输出 (MDOC)

关闭运动凸轮输出 (MDOC) 指令向指定的轴上连接的一个或多个凸轮输出发起禁止。根据禁止类型，MDOC 将禁止所有凸轮输出或者仅某个特定的凸轮输出。在禁止以后，相应的输出将维护最后的状态。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_FEEDBACK AXIS_CONSUMED AXIS_VIRTUAL AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	标签	将位置输入提供给凸轮输出的轴的名称。省略号将启动轴属性对话框。
执行目标	SINT、INT 或 DINT	立即或标签	执行目标定义了连接到命名轴的集中的特定凸轮输出。行为是由以下方面确定的： ² 0...8 ñ 在 Logix 控制器中执行的凸轮输出。 ² 9...31 ñ 预留为将来使用。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
禁止类型	UINT32	立即	为指定的轴选择一个或全部要禁止的凸轮输出。选择任一： 0 = 全部 – 禁止指定的轴上连接的全部凸轮输出。 1 = 特定 – 禁止指定的轴上连接的由执行目标所定义的凸轮输出。

结构文本

操作数与梯形图 MDOC 指令的操作数相同。



MDOC(Axis,ExecutionTarget,
MotionControl,DisarmType);

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
禁止类型	all	0
	类型有关	1

MOTION_INSTRUCTION 结构

记忆单元：	说明：
.EN（启用）位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到梯级变为假为止。
.DN（完成）位 29	当凸轮输出成功禁止时，完成位被设置。
.ER（错误）位 28	这个位被设置时表示指令检测到错误。

说明： 关闭运动凸轮输出 (MDOC) 指令根据所选的禁止类型，禁止指定的轴上某个特定的或全部的凸轮输出。该轴将位置输入提供给凸轮输出。执行目标定义了连接到指定轴的集中的特定凸轮输出。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

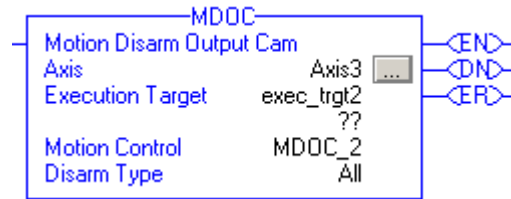
错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。参数超出范围 (13) 错误代码的扩展错误代码列出的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MDOC 指令，扩展错误代码 4 指的是禁止类型操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

状态位: MDOC 对状态位的更改

无

示例: 梯形图



MDOC 梯形图示例

结构文本

```
MDOC(Axis3,exec_trgt2,MDOC_2,all);
```

注意：

运动配置指令 (MAAT、MRAT、MAHD、MRHD)

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

简介

配置指令包括用于建立伺服配置参数并应用到轴上的所有运动控制指令。这个指令组包括连接测试诊断指令和整定指令。

使用运动配置指令可以整定轴，并可以为伺服系统运行诊断测试。这些测试包括：

- 电机编码器连接测试。
- 编码器连接测试。
- 标志测试

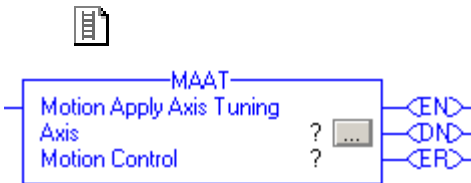
运动配置指令为：

如果您需要	使用这个指令	在以下语言中可用
<p>根据以前执行的 MRAT 指令来计算伺服增益和动态门限的完整集合。</p> <p>MAAT 指令还是用新的增益参数更新伺服模块。</p>	MAAT	<p>梯形图</p> <p>结构文本</p>
<p>伺服模块运行某个轴的整定运动模式的命令。</p>	MRAT	<p>梯形图</p> <p>结构文本</p>
<p>应用以前执行的 MRHD 指令的结果。</p> <p>MAHD 根据 MRHD 指令过程中观察到运动方向，生成一套新的编码器和伺服极性。</p>	MAHD	<p>梯形图</p> <p>结构文本</p>
<p>伺服模块在某个轴上运行三个诊断测试之一的命令。</p>	MRHD	<p>梯形图</p> <p>结构文本</p>

应用轴整定参数 (MAAT)

应用轴整定参数 (MAAT) 指令用于根据以前运行的运行轴整定 (MRAT) 指令的结果来计算一套伺服增益和动态门限，并使用新的增益参数更新运动控制模块。当这个指令不带显性参数时，输入源自于轴整定配置参数，如运动轴对象技术参数中所述。在执行 MAAT 指令后，相应的轴应准备就绪可以进行伺服激活。

操作数: 梯形图



MAAT(Axis,MotionControl);

操作数	类型	格式	描述
轴	AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MAAT 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	启用位指示指令被启用。它保持为设置状态，直至伺服消息完成而且梯级输入条件变为假为止。
.DN (完成) 位 29	完成位指示指令完成一个应用轴整定过程。
.ER (错误) 位 28	错误位指示指令检测到错误，例如轴是否未被配置。

说明: 应用轴整定参数 (MAAT) 指令用于执行一系列计算，为指定轴上的增益和动态配置参数生成值。作为 MAAT 所执行工作的一部分，这些结果配置参数将被应用，从而使轴准备就绪可以进行完整的伺服操作。这个指令设计用于跟随 MRAT 指令的执行，后者为 MAAT 指令生成轴输入配置值。有关更多信息，请参见 MRAT 指令的叙述。MAAT 不需要显性输入参数，只需输入或选择所需的物理轴即可。

如果目标轴没有出现在可用轴列表中，则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

MAAT 指令使用轴配置参数作为输入和输出。MRAT 使用的输入配置参数如下表所示。请参阅运动轴对象技术参数，了解有关这些参数的详细叙述。

MAAT 用作输入的轴配置参数取决于外部变频器的配置。如果外部速度伺服变频器配置位参数为 TRUE，表示接口至外部速度伺服变频器，则需要以下输入参数。

轴参数	数据类型	单位	含义
整定速度	实数型	位置单位 / 秒	整定模式的最高速度。
整定加速	实数型	位置单位 / 秒 ²	计算出的整定模式加速时间。
整定减速	实数型	位置单位 / 秒 ²	计算出的整定模式减速时间。
整定速度标定	实数型	mV/KCPS	轴变频器 / 电机 / 编码器系统测量的速度标定系数。
整定速度带宽	实数型	赫兹	外部速度伺服变频器的带宽

如果外部速度伺服变频器配置位参数为 FALSE，表示接口至外部转矩伺服变频器，则需要以下输入参数。

轴参数	数据类型	单位	含义
阻尼系数	实数型	-	计算增益所用的阻尼系数。
整定速度	实数型	位置单位 / 秒	整定模式的最高速度。
整定加速	实数型	位置单位 / 秒 ²	计算出的整定模式加速时间。
整定减速	实数型	位置单位 / 秒 ²	计算出的整定模式减速时间。
有效惯量	实数型	mV/ KCPS ²	变频器 / 电机系统计算出的有效惯量。
位置伺服带宽	实数型	赫兹	位置伺服环的最大带宽。

MAAT 作为输出生成的轴配置参数取决于外部变频器的配置。如果外部速度伺服变频器配置位参数为 TRUE，表示接口至外部速度伺服变频器，则生成以下输出参数。

轴参数	数据类型	单位	含义
位置比例增益	实数型	1/ 毫秒	位置伺服环比例增益
位置积分增益	实数型	1/ 毫秒 ²	位置伺服环积分增益 – 设置为零
速度前馈	实数型	-	位置伺服环比例增益
加速度前馈	实数型	-	速度命令前馈 – 设置为零
最大速度	实数型	位置单位 / 秒	运动模式的最大速度 – 设置为整定速度
最大加速度	实数型	位置单位 / 秒 ²	运动模式的最大加速度
最大减速度	实数型	位置单位 / 秒 ²	运动模式的最大加速度
输出滤波器带宽	实数型	赫兹	低通伺服输出滤波器的带宽
输出标定	实数型	mV/ KCPS	应用到位置伺服环至 DAC 的输出的标定系数。
位置误差容限	实数型	位置单位	允许而不会出现故障的最大伺服环位置误差。

如果外部速度伺服变频器配置位参数为 FALSE，表示接口至外部转矩伺服变频器，则生成以下输出参数。

轴参数	数据类型	单位	含义
位置比例增益	实数型	1/ 毫秒	位置伺服环比例增益
位置积分增益	实数型	1/ 毫秒 ²	位置伺服环积分增益
速度比例增益	实数型	1/ 毫秒	速度伺服环比例增益
速度积分增益	实数型	1/ 毫秒 ²	速度伺服环积分增益
速度前馈	实数型	-	位置伺服环比例增益
加速度前馈	实数型	-	速度命令前馈
最大速度	实数型	位置单位 / 秒	运动模式的最大速度 – 设置为整定速度

轴参数	数据类型	单位	含义
最大加速度	实数型	位置单位 / 秒 ²	运动模式的最大加速度
最大减速度	实数型	位置单位 / 秒 ²	运动模式的最大加速度
输出滤波器带宽	实数型	赫兹	低通伺服输出滤波器的带宽
输出标定	实数型	mV/ KCPS ²	应用到速度伺服环至 DAC 的输出的标定系数。
位置误差容限	实数型	位置单位	允许而不会出现故障的最大伺服环位置误差。

MAAT 指令所生成的上述输出参数将立即应用到指定的轴上，从而可以后续的运动。

有关整定配置参数的更多信息，请参阅运动轴对象技术参数。

要成功执行 MAAT 指令，目标轴必须配置为伺服轴并且处于轴就绪状态，而且伺服动作关闭。如果不满足这些条件，则指令出错。

重要提示

MAAT 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAAT 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

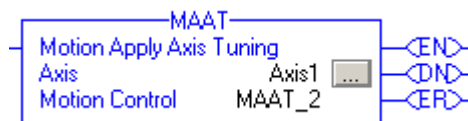
相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	无资源 (2)	没有足够的内存资源来完成请求。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	对象模式冲突 (12)	轴处于关闭状态。
SERVO_MESSAGE_FAILURE (12)	权限被拒绝 (15)	使能输入开关错误。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)	重新定义位置、归零和记录 2 是相互排斥的 (SERCOS)，设备状态不能用于动作。(SERCOS)

状态位： MAAT 更改至状态位

无

示例： 当输入条件为真时，控制器基于以前执行的运行轴整定 (MRAT) 指令的结果，为轴 1 计算一整套伺服增益和动态门限。

梯形图



MAAT 梯形图示例

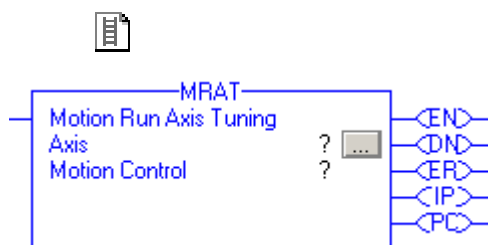
结构文本

```
MAAT(Axis1,MAAT_2);
```

运行轴整定 (MRAT)

使用 MRAT 来命令运动控制模块为指定的轴运行一次整定运动模式。整定运动模块包括一次或多次加速和减速斜坡，通过向伺服的变频器输出施加固定电压来实现。注意，这个指令在任何时候都不会关闭伺服环。当这个指令不带显性参数时，它不从轴整定配置参数中获取输入。执行 MRAT 指令的结果是一套测量数据，储存在轴对象内供后续在应用轴整定参数 (MAAT) 指令中使用。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。



MRAT(Axis,MotionControl);

结构文本

操作数与梯形图 MRAT 指令的操作数相同。

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置，并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当整定过程成功完成后这个位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误，例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	这个位在正的梯级转换时设置，在整定过程完成后或者被停止命令、关闭或伺服故障清除。
.PC(过程完成) 位 27	当整定过程成功完成后这个位被设置。

说明: 运行轴整定 (MRAT) 指令用于在指定的轴上执行一次整定运动模式。在这个简短的整定运动模式中，运动控制模块进行计时和速度测量，用作后续 MAAT(应用轴整定参数) 指令的输入。MRAT 不需要显性输入参数，只需输入或选择所需的物理轴即可。

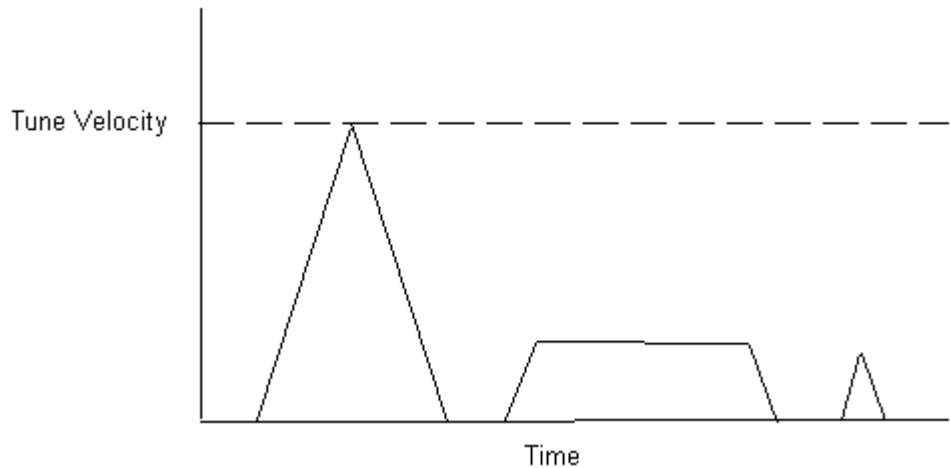
如果目标轴没有出现在可用轴列表中，则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

MRAT 指令使用轴配置参数作为输入和输出。MRAT 使用的输入配置参数如下表所示。

轴参数	数据类型	单位	含义
整定方向	布尔型	-	整定运动的方向 (0- 正向、1- 反向)
整定行程限制	实数型	位置单位	轴允许的最大冲程
整定速度	实数型	位置单位 / 秒	整定模式的最高速度。
阻尼系数	实数型	-	计算最大位置伺服带宽所用的阻尼系数。

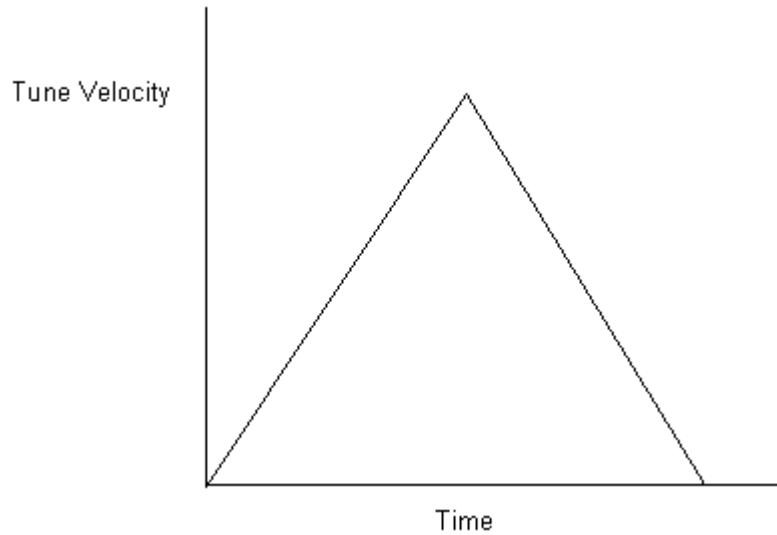
根据上述配置参数，MRAT 执行将生成指定轴上的一个运动事件，包括单个三角形速度模式或者一系列三个这种模式。整定速度必须位于变频器和电机的最大速度容量范围以内。为整定速度所配置的值应设置为所期望的轴的最大运行转速，使产生的整定参数能够基于该速度上的系统动态范围上。

如果外部速度伺服变频器配置位参数为 TRUE，表示接口至外部速度伺服变频器，则向轴上应用三个脉冲。这种情况的整定速度模式如下图所示。



当为真时的整定速度模式

如果外部速度伺服变频器配置位参数为 FALSE，表示接口至外部转矩伺服变频器，则向轴上仅应用一个脉冲。下面显示了整定速度模式。



当为假时的整定速度模式

MRAT 作为输出生成的轴配置参数取决于外部变频器的配置。如果外部速度伺服变频器配置位参数为 TRUE，表示接口至外部速度伺服变频器，则生成以下输出参数。

轴参数	数据类型	单位	含义
整定状态	实数型	-	整定过程的状态报告
整定加速时间	实数型	秒	测量出的整定模式加速时间。
整定减速时间	实数型	秒	测量出的整定模式减速时间。
整定加速	实数型	位置单位 / 秒 ²	计算出的整定模式加速时间。
整定减速	实数型	位置单位 / 秒 ²	计算出的整定模式减速时间。
整定速度标定	实数型	mV/KCPS	轴变频器 / 电机 / 编码器系统测量的速度标定系数。
整定上升时间	实数型	mV/KCPS	测量出的整定阶跃响应模式的上升时间。
整定速度带宽	实数型	赫兹	计算出的外部速度伺服变频器的带宽

如果外部速度伺服变频器配置位参数为 FALSE，表示接口至外部转矩伺服变频器，则生成以下输出参数。

轴参数	数据类型	单位	含义
整定状态	实数型	-	整定过程的状态报告
整定加速时间	实数型	秒	测量出的整定模式加速时间。
整定减速时间	实数型	秒	测量出的整定模式减速时间。
整定加速	实数型	位置单位 / 秒 ²	计算出的整定模式加速时间。
整定减速	实数型	位置单位 / 秒 ²	计算出的整定模式减速时间。
有效惯量	实数型	mV/ KCPS ²	变频器 / 电机系统计算出的有效惯量。
位置伺服带宽	实数型	赫兹	计算出的位置伺服环的最大带宽。

由 MRAT 指令生成的上述输出参数用作后续 MAAT 指令的输入，从而执行进一步的整定计算并将结果应用到各个轴的伺服和动态配置参数中。

整定状态参数

可能发生情况使控制器无法正确执行整定操作。在这种情况下，整定过程自动中止，并报告一个整定故障，储存在整定状态输出参数 (GSVable) 内。也可以使用 MAS 指令手动中止整定过程，从而产生由整定状态参数所报告的整定故障。整定状态的可能值如下表所示。

状态代码	代码	含义
整定成功	0	整定过程已经成功。
整定正在进行	1	整定正在进行之中。
整定中止	2	整定过程被用户中止。
整定超时	3	整定过程已经超时。
整定伺服故障	4	整定过程失败，因出现伺服故障
整定行程故障	5	轴达到整定的行程限制
整定极性故障	6	轴运动朝向错误方向，因为电机 / 编码器极性配置不正确。
整定速度故障	7	轴的整定速度太低，不能获得最低测量准确度。

重要提示

整定状态参数不会被 MRAT 指令的 .STATUS 子标签弄错。

要成功在某个轴上执行 MRAT 指令，目标轴必须配置为伺服轴类型并且处于轴就绪状态。如果不满足任何条件，则指令出错。

重要提示

当 MRAT 指令初次执行时，正在进行 (.IP) 位被设置，过程完成 (.PC) 位被清除。MRAT 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输多条消息。完成 (.DN) 位并不立即设置，而只是在这些消息被成功发送以后。在完成 (.DN) 位被设置的同时，正在进行 (.IP) 位被清除，过程完成 (.PC) 位被设置。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MRAT 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

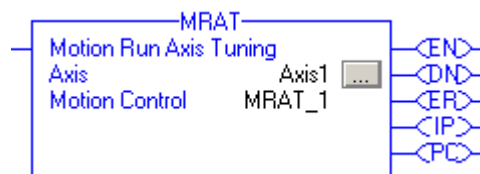
相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	过程根据请求终止 (1)	跟随某个关闭 / 禁止变频器的指令或者运动停止指令的整定执行，或者处理器变更请求了一次整定取消。
SERVO_MESSAGE_FAILURE (12)	对象模式冲突 (12)	轴处于关闭状态。
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)	整定过程顺序不正确。(SERCOS)

状态位： MRAT 更改至状态位

位名称：	状态：	含义：
DriveEnableStatus	真	当整定模式正在运行时，轴处于变频器控制状态，变频器使能输出激活。
TuneStatus	真	轴正在运行一次整定过程。

示例： 当输入条件为真时，控制器命令伺服模块为 #1 运行一次整定运动模式。

梯形图



MRAT 梯形图示例

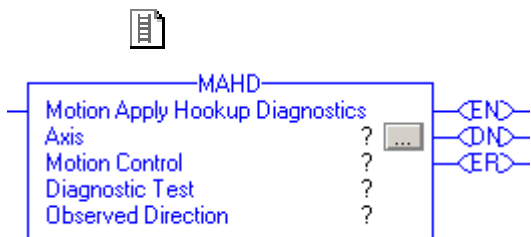
结构文本

```
MAR(Axis1,MRAT_1);
```

应用轴连接诊断 (MAHD)

应用轴连接诊断 (MAHD) 指令用于应用以前运行的运行轴连接诊断 (MRHD) 指令的结果，以基于在测试过程中观察到的运动方向生成一套新的编码器和伺服极性。作为应用过程的一部分，该指令使用这些极性设置更新运动控制模块。在执行 MAHD 指令后，假设已经建立一套稳定的增益，相应的轴应准备就绪可以进行伺服激活。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
诊断测试	UDINT	立即	选择特定的测试让运动控制模块来运行： 0 = 电机 / 编码器连接测试 1 = 编码器连接测试 2 = 编码器标志测试
观测方向	布尔型	立即	设置测试运动的方向。选择任一： 0 = 正向 1 = 反向



MAHD(Axis, MotionControl, DiagnosticTest, ObservedDirection);

结构文本

操作数与梯形图 MAHD 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您 ...	
	输入为文本	或者输入为数字
DiagnosticTest	motor_encoder	0
	编码器	1
	标记	2
ObservedDirection	正向	0
	反向	1

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当连接测试过程成功执行后这个位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。

说明: 应用轴连接诊断 (MAHD) 指令用于执行一系列计算, 为指定轴上的编码器极性和伺服极性配置位参数生成值。作为 MAHD 所执行工作的一部分, 这些结果配置位参数将被应用到运动控制模块, 从而使轴准备就绪可以进行完整的伺服操作。这个指令设计用于跟随 MRHD 指令的执行, 后者为 MAHD 指令生成轴输入配置值。有关更多信息, 请参见 MRHD 指令的叙述。MAHD 需要指定要应用的诊断测试, 以及在以前 MRHD 测试过程中观察到的运动方向。请输入或选择诊断测试和观测方向以及所需要的物理轴。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

MAHD 指令使用轴配置参数作为输入和输出。MAHD 使用的输入配置参数如下表所示。测试方向正向位将自动确定, 作为 MRHD 指令的输出。请参阅运动轴对象技术参数, 了解有关这些及其他参数的详细叙述。

轴参数	数据类型	单位	定义
测试方向前向	布尔型	-	在连接测试过程中由运动控制模块所看到的轴的运行方向。

电机编码器连接测试

如果选择了电机编码器连接测试, 控制器将基于观测方向指令参数以及由 MRHD 指令的输出所确定的测试方向前向位的状态, 同时为编码器极性以及变频器极性计算正确设置。编码器极性以及变频器极性一旦计算出来, MAHD 将这些值应用到相应的轴配置参数位上, 如下表所述:

轴参数	数据类型	单位	定义
编码器极性负	布尔型	-	反转进入运动控制模块的编码器反馈输入的方向。
变频器极性负	布尔型	-	反转从运动控制模块出来 DAC 模拟量输出的方向。

编码器连接测试

如果选择了编码器连接测试，控制器将基于观测方向指令参数以及由 MRHD 指令的输出所确定的测试方向前向位的状态，仅为编码器极性计算正确设置。编码器极性以及变频器极性一旦计算出来，MAHD 将这些值应用到相应的轴配置参数位上，如下表所述：

轴参数	数据类型	单位	定义
编码器极性负	布尔型	-	反转进入运动控制模块的编码器反馈输入的方向。

要成功执行 MAHD 指令运行电机编码器测试，目标轴必须配置为伺服或者反馈唯一轴类型。如果不满足任何条件，则指令出错。

重要提示

MAHD 指令执行可能需要多次扫描来执行，因为它需要向运动控制模块传输一条消息。完成 (.DN) 位并不立即设置，而只是在这条消息被成功发送以后。

这是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志： 不受影响

故障条件： 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAHD 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

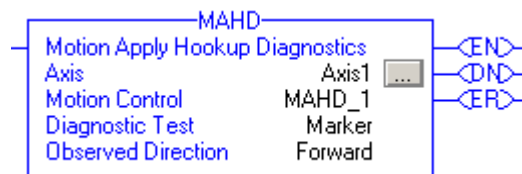
相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	无资源 (2)	没有足够的内存资源来完成请求。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	对象模式冲突 (12)	轴处于关闭状态。
SERVO_MESSAGE_FAILURE (12)	权限被拒绝 (15)	使能输入开关错误。(SERCOS)
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)	重新定义位置、归零和记录 2 是相互排斥的 (SERCOS)，设备状态不能用于动作。(SERCOS)

状态位： MAHD 更改至状态位

无

示例： 当输入条件为真时，控制器将以前执行的运行轴连接诊断 (MRHD) 指令的结果应用到轴 1。

梯形图



MAHD 梯形图示例

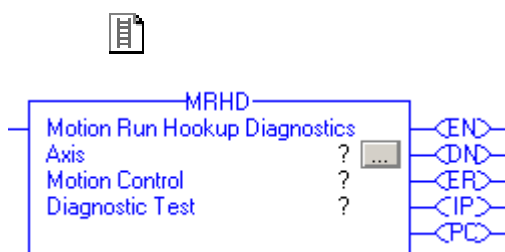
结构文本

```
MAHD(axis1,axis1_MAHD,marker,forward);
```

运行轴连接诊断 (MRHD)

使用 MRHD 指令可命令运动控制模块来在指定的轴上运行三种不同诊断中的任意一种，根据测试 ID 进行选择。目前的诊断可用于为伺服轴、仅编码器连接以及编码器标志连接来测试电机 / 编码器连接。只有电机 / 编码器诊断会发起轴上的运动。这个动作包含一个用户电机编码器测试增量的短暂移动。这种位移由伺服变频器输出的大约每秒 1 伏特的斜坡电平所发起。执行 MRHD 指令的结果是参数、状态位和测试方向前向都将被更新。

操作数: 梯形图



操作数	类型	格式	描述
轴	AXIS_SERVO AXIS_SERVO_DRIVE	标签	要在其上执行操作的轴的名称。
运动控制	MOTION_指令	标签	用于访问指令状态参数的结构。
诊断测试	DINT	立即	选择特定的测试让运动控制模块来运行： 0 = 电机 / 编码器连接测试 1 = 编码器连接测试 2 = 编码器标志连接测试 3 = 看门狗 OK 测试



MRHD(Axis, MotionControl, DiagnosticTest);

结构文本

操作数与梯形图 MRHD 指令的操作数相同。

对于需要您从可用选项中进行选择的操作数，请按照以下方式输入您的选择：

此操作数:	拥有这些选项可让您 ...	
	输入为文本:	或者输入为数字:
DiagnosticTest	motor_encoder	0
	编码器 (encoder)	1
	标记	2

MOTION_INSTRUCTION 结构

记忆单元	描述
.EN (启用) 位 31	当梯级进行一次从假到真的转换时这个位被设置, 并保持设置状态直到伺服消息转换完成、梯级变为假为止。
.DN (完成) 位 29	当连接测试应用过程成功执行后这个位被设置。
.ER (错误) 位 28	这个位被设置表示指令检测到错误, 例如您指定了一个未配置的轴。
.IP(正在进行) 位 26	这个位在正的梯级转换时设置, 在诊断测试过程完成后或者被停止命令、关闭或伺服故障清除。
.PC(过程完成) 位 27	当诊断测试应用过程成功完成后这个位被设置。

说明: 运行轴连接诊断 (MRHD) 指令用于在指定的轴上执行各种测试诊断, 以测试完整性以及在某些情况下的伺服现场连接极性。目前的测试诊断支持变频器连接、编码器连接、标志连接和运动控制模块 OK 触点连接。在部分这些测试过程中, 运动控制模块生成到外部变频器的输出, 以产生少量的运动。在部分这些连接诊断测试中所做的测量保存为输出配置参数, 同时作为后续 MAHD(应用轴连接诊断) 指令的输入数据。MRHD 仅需要一个显性输入参数, 即诊断测试。请输入或选择要运行的诊断测试以及要测试的轴。

如果目标轴没有出现在可用轴列表中, 则该轴尚未配置用于操作。请使用标签编辑器来创建和配置新的轴。

MRHD 指令使用轴配置参数作为输入和输出。MRHD 使用的输入配置参数如下表所示。

轴参数	数据类型	单位	定义
电机编码器测试增量	实数型	-	轴为了满足连接诊断测试而必须进行进的距离

MRHD 作为输出生成的轴配置参数取决于指定连接诊断的配置。

电机编码器连接测试

如果选择了电机编码器测试, 运动控制模块使能外部变频器, 在监视编码器反馈的同时生成一个每秒 1 伏特的输出斜坡发送至变频器。当轴位移的距离大于或等于所配置的电机编码器测试增量时, 测试电压

被重新设置回到零，变频器被禁止。运动控制模块然后报告行进方向，该方向储存在下面一个输出参数内。

轴参数	数据类型	单位	定义
测试状态	整数型	-	连接诊断测试过程的状态报告
测试方向前向	布尔型	-	在连接测试过程中由运动控制模块所看到的轴的运行方向。

如果因为连接不正确或者系统的其他问题，轴反馈未能在 2 秒钟内检测到轴到达所配置的电机编码器测试增量，则伺服将测试电压重新设置回到零，并禁止变频器。控制器通过测试状态轴输出参数来反映这个情况。这通常表示变频器的连线或者编码器的连线不正确。在选择编码器连接测试的情况下运行 MRHD 是找出是编码器问题还是变频器问题的一种有效方法。

编码器连接测试

如果选择了编码器测试，运动控制模块不生成任何轴运动，而只是监视轴编码器的反馈。然后轴可用手移动，或者用其他某种独立的驱动执行机构来产生运动。当轴移动的距离大于或等于所配置的电机编码器测试增量时，本测试完成。运动控制模块然后报告行进方向，作为下面一个 MRHD 输出参数。

轴参数	数据类型	单位	定义
测试状态	整数型	-	连接诊断测试过程的状态报告
测试方向前向	布尔型	-	在连接测试过程中由运动控制模块所看到的轴的运行方向。

如果因为连接不正确或者系统的其他问题，在将轴至少移动该距离后轴反馈未能检测到轴到达所配置的电机编码器测试增量，使用 MAS 指令中止测试，检查编码器的连线。

标志连接测试

如果选择了标志测试，运动控制模块不生成任何轴运动，而只是监视轴编码器的反馈。然后轴可用手移动，或者用其他某种独立的驱动执行机构来产生运动。当运动控制模块检测到某个标志（通道 Z）脉冲时，测试即完成。运动控制模块然后通过测试状态报告成功

轴参数	数据类型	单位	定义
测试状态	整数型	-	连接诊断测试过程的状态报告
测试方向前向	布尔型	-	在连接测试过程中由运动控制模块所看到的轴的运行方向。

如果因为连接不正确或者系统的其他问题，在将轴至少移动该距离后轴反馈未能检测到轴到达所配置的电机编码器测试增量，使用 MAS 指令中止测试，检查编码器的连线。

看门狗 OK 测试

如果选择了看门狗 OK 测试，运动控制模块不生成任何轴运动，而只是模拟 CPU 看门狗失效，即打开 OK 触点。OK 触点应在 2 秒钟后保持关闭。这个测试用于检查进入变频器系统紧急停机串的 OK 触点连线。如果运动控制模块 DSP 失效，这种机构可用于关闭通往变频器的电源。当两秒钟看门狗 OK 测试完成后，运动控制模块会通过测试状态报告成功，如下所示：

轴参数	数据类型	单位	定义
测试状态	整数型	-	连接诊断测试过程的状态报告

测试状态

可能发生情况使控制器无法正确执行测试操作。在这种情况下，测试过程自动中止，并报告一个测试故障，储存在测试状态输出参数内。也可以使用 MAS 指令手动中止测试过程，从而产生由测试状态参数所报告的测试故障。测试状态的可能值如下表所示：

错误消息	代码	定义
测试成功	0	测试过程已经成功。
测试正在进行	1	测试正在进行之中。
测试中止	2	测试过程被用户中止。

错误消息	代码	定义
测试超时	3	测试过程已经超时 (2 秒钟)
测试伺服故障	4	测试过程失败, 因出现伺服故障
测试增量故障	5	测试过程失败, 因测试增量距离不足以进行可靠的测量。

要成功执行 MRHD 指令来运行电机编码器测试, 目标轴必须配置为伺服轴类型并且处于轴就绪状态。对于其他测试, 这条指令在伺服或者反馈唯一轴类型上可正确执行。如果不满足任何条件, 则指令出错。

重要提示

当 MRHD 指令初次执行时, 正在进行 (.IP) 位被设置, 过程完成 (.PC) 位被清除。MRHD 指令执行可能需要多次扫描来执行, 因为它需要向运动控制模块传输多条消息。完成 (.DN) 位并不立即设置, 而是在这些消息被成功发送以后。在完成 (.DN) 位被设置的同时, 正在进行 (.IP) 位被清除, 过程完成 (.PC) 位被设置。

这是一条转换指令:

- 在梯形图中, 当每次应执行指令时, 将梯级输入条件从清除切换到设置。
- 在结构文本中, 限制指令只对转换执行。请参阅附录 C。

算术状态标志: 不受影响

故障条件: 无

错误代码 参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码： 扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MRHD 指令收到伺服消息失败 (12) 错误消息时，下面的扩展错误代码可帮助确定问题位置。

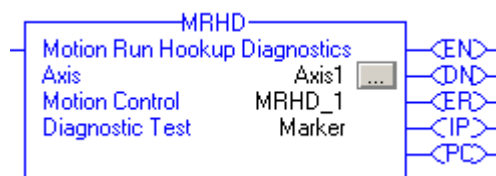
相关联的错误代码 (十进制)	扩展错误代码 (十进制)	含义
SERVO_MESSAGE_FAILURE (12)	过程根据请求终止 (1)	跟随某个关闭 / 禁止变频器的指令或者运动停止指令的测试执行，或者处理器变更请求了一次测试取消。
SERVO_MESSAGE_FAILURE (12)	对象模式冲突 (12)	轴处于关闭状态。
SERVO_MESSAGE_FAILURE (12)	设备处于错误状态 (16)	整定过程顺序不正确。(SERCOS)

状态位： MRHD 更改至状态位

位名称	状态	含义
DriveEnableStatus	真	2 轴处于变频器控制状态。 2 当整定模式正在运行时，变频器使能输出激活。
TestStatus	真	轴正在运行一次测试过程。

示例： 当输入条件为真时，控制器在轴 1 上运行编码器诊断测试。

梯形图



MRHD 梯形图示例

结构文本

```
MRHD(Axis1,MRHD_1,Marker);
```

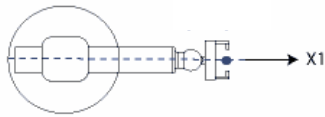
运动协调指令

(MCLM、MCCM、MCCD、MCS、MCSD、MCT、MCTP、MCSR)

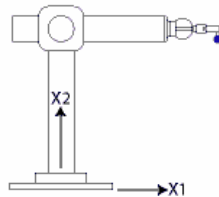
简介

使用运动协调指令可移动坐标系中的最多三个轴。

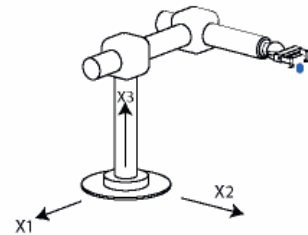
带正交轴的坐标系



笛卡尔坐标系

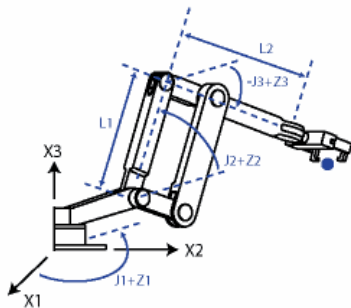


二维笛卡尔坐标系

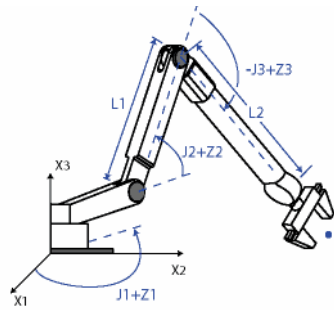


三维笛卡尔坐标系

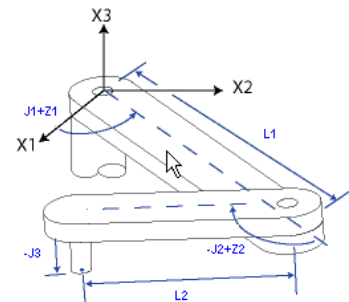
带非正交轴的坐标系



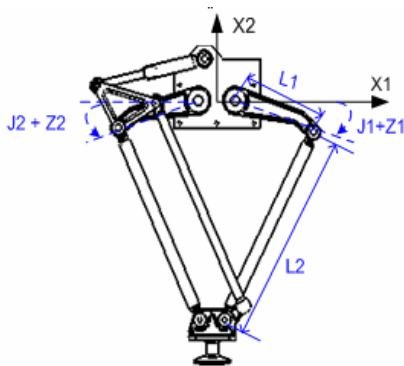
关节非独立坐标系



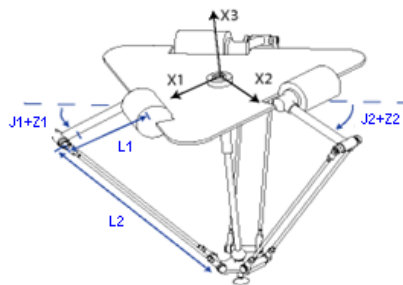
关节独立坐标系



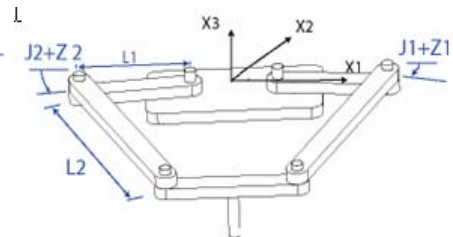
SCARA 独立坐标系



增量二维坐标系



增量三维坐标系



SCARA 增量坐标系

使用这个表格来选择运动协调指令。

选择运动协调指令

如果您需要	使用这个指令	在以下语言中可用
在笛卡尔坐标系内的指定轴上发起一维或多维线性协调运动。	运动直线插补 (MCLM)	<ul style="list-style-type: none"> • 梯形图 • 结构文本
在笛卡尔坐标系内的指定轴上发起二维或三维循环协调运动。	运动圆弧插补 (MCCM)	<ul style="list-style-type: none"> • 梯形图 • 结构文本
对指定的坐标系内的有效的协调运动发起路径动态参数变更。	运动插补动态调整 (MCCD)	<ul style="list-style-type: none"> • 梯形图 • 结构文本
停止坐标系的轴，或者取消转换。	运动插补停止 (MCS)	<ul style="list-style-type: none"> • 梯形图 • 结构文本
对指定坐标系中的所有轴发起受控停机。	运动插补关闭 (MCSD)	<ul style="list-style-type: none"> • 梯形图 • 结构文本
启动一个将两个坐标系连起来的坐标转换。	运动插补坐标转换 (MCT) ⁽¹⁾	<ul style="list-style-type: none"> • 梯形图 • 结构文本
计算一个坐标系相对于另一个坐标系的位置。	计算坐标转换位置 (MCTP) ⁽¹⁾	<ul style="list-style-type: none"> • 梯形图 • 结构文本
发起指定的坐标系上所有轴从关闭状态到轴就绪状态的复位，并清除轴故障。	运动插补关闭复位 (MCSR)	<ul style="list-style-type: none"> • 梯形图 • 结构文本

⁽¹⁾ 您只能在 1756-L6x 控制器上使用这条指令。

在混合指令时使用不同的终止类型

要混合两个 MCLM 或 MCCM 指令，请启动第一个，排队第二个。坐标系的标签为您提供两个位于排队指令。

- MovePendingStatus
- MovePendingQueueFullStatus

例如，下面的梯形图使用坐标系 cs1 来将 Move1 混合到 Move2 中。

示例混合指令梯形图

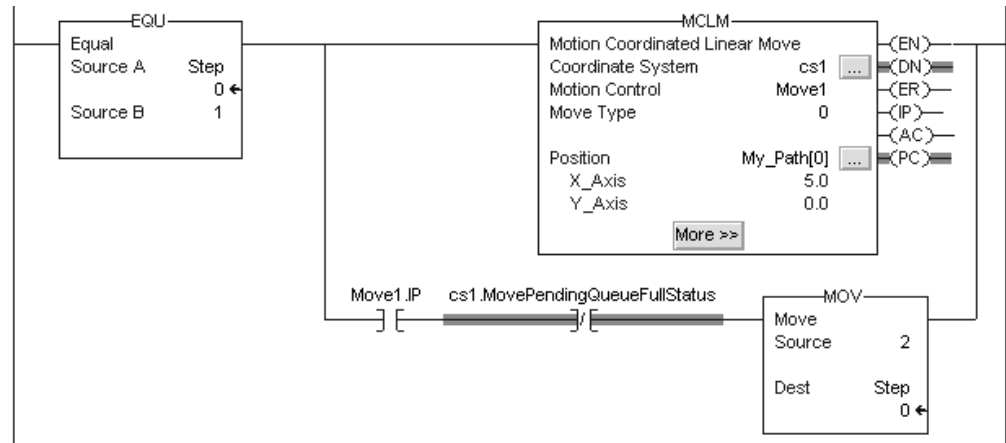
如果步序 = 1 则

Move1 启动，将轴移到位置 5, 0

而且一旦 Move1 正在进行

而且有空排队另一个运动

步序 = 2



下页待续

如果步序 = 2 则

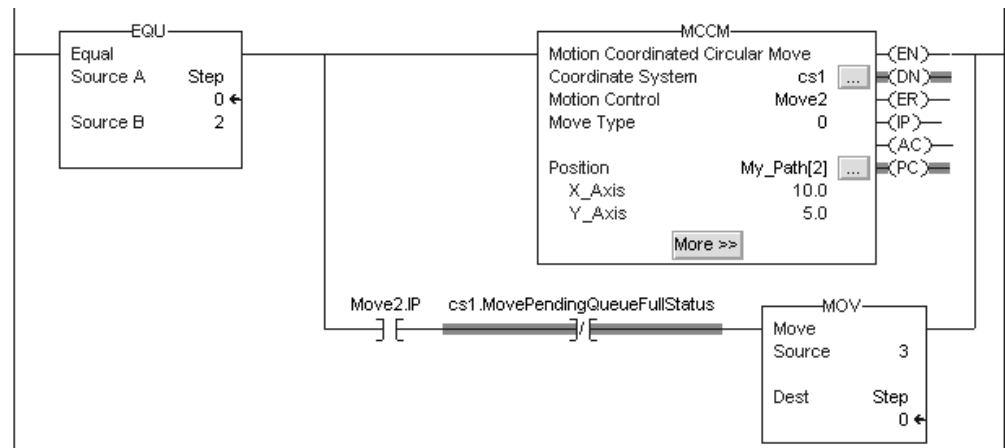
Move1 已经发生。

Move2 进入队列，等待 Move1 完成。

当 Move1 完成后，Move2 将轴移到位置 10, 5。

而且一旦 Move2 正在进行，且队列内有空，

步序 = 3



当一条指令完成后，它将从队列中删除，腾出空来为另一条指令进入队列。两个位的值始终都相同，因为您一次只能排队一条挂起的指令。如果应用需要几条指令同时按次序执行，则这些位使用这些参数进行设置。

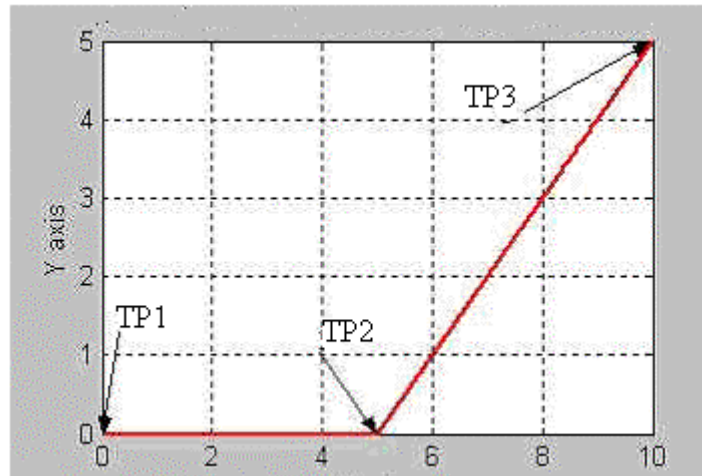
位参数

场合	那么
一条指令正在工作，第二条指令在队列内挂起	<ul style="list-style-type: none"> • MovePendingStatus 位 = 1 • MovePendingQueueFullStatus 位 = 1 • 您不能排队另一条指令
一条正在工作的指令完成并离开队列	<ul style="list-style-type: none"> • MovePendingStatus 位 = 0 • MovePendingQueueFullStatus 位 = 0 • 您可以排队另一条指令

MCLM 或 MCCM 指令的终止类型操作数规定了当前正在执行的运动如何终止。下图显示在各个转接点 (TP) 处受到影响的指令位和坐标系位的状态。

使用实际容差或无安排的混合运动转接点处的位状态

线性？线性运动



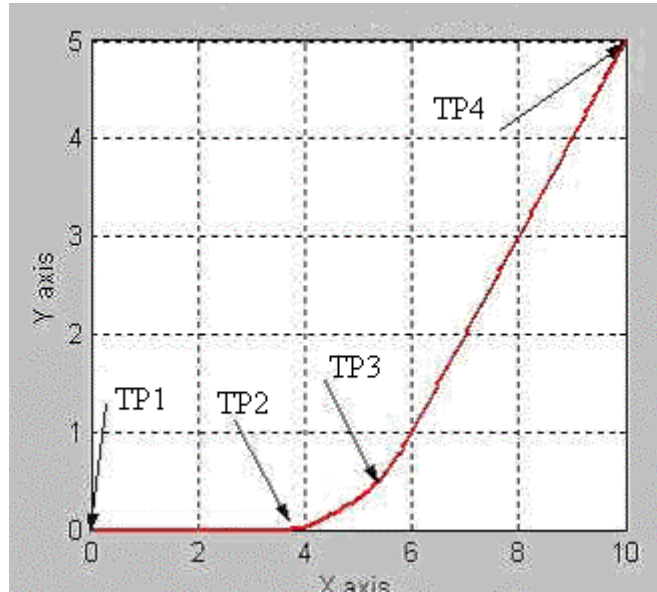
下表显示上图中终止类型为实际容差或无安排时各个转接点处的位状态。

实际容差或无安排终止类型时转接点处的位状态

位	TP1	TP2	TP3
Move1.DN	T	T	T
Move1.IP	T	F	F
Move1.AC	T	F	F
Move1.PC	F	T	T
Move2.DN	T	T	T
Move2.IP	T	T	F
Move2.AC	F	T	F
Move2.PC	F	F	T
cs1.MoveTransitionStatus	F	F	F
cs1.MovePendingStatus	T	F	F
cs1.MovePendingQueueFullStatus	T	F	F

使用无减速的混合运动转接点处的位状态

线性 → 线性运动



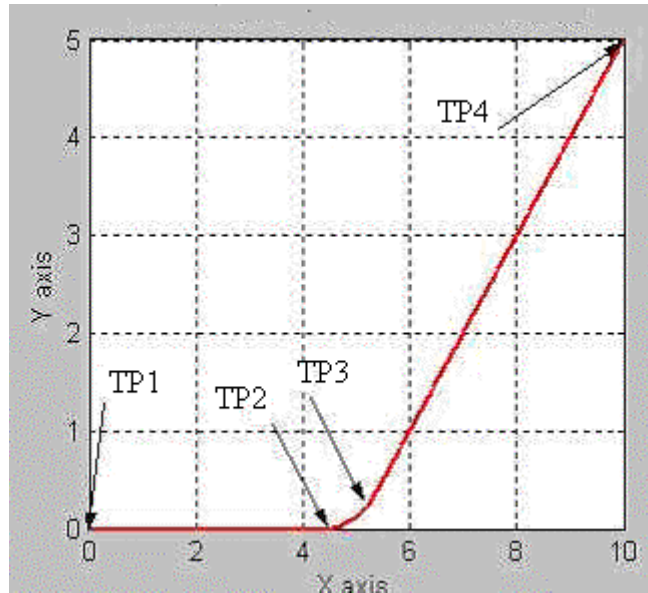
下表显示上图中终止类型为无减速时各个转接点处的位状态。对于无减速终止类型，转接点 TP2 的剩余移动量等于 Move1 指令的减速距离。如果 Move1 和 Move2 具有共线性，则 Move1.PC 将在 TP3（第一个运动的编程终点）处为真。

使用无减速终止类型的位状态

位	TP1	TP2	TP3	TP4
Move1.DN	T	T	T	T
Move1.IP	T	F	F	F
Move1.AC	T	F	F	F
Move1.PC	F	T	T	T
Move2.DN	T	T	T	T
Move2.IP	T	T	T	F
Move2.AC	F	T	T	F
Move2.PC	F	F	F	T
cs1.MoveTransitionStatus	F	T	F	F
cs1.MovePendingStatus	T	F	F	F
cs1.MovePendingQueueFullStatus	T	F	F	F

使用命令容差的混合运动转接点处的位状态

线性 → 线性运动



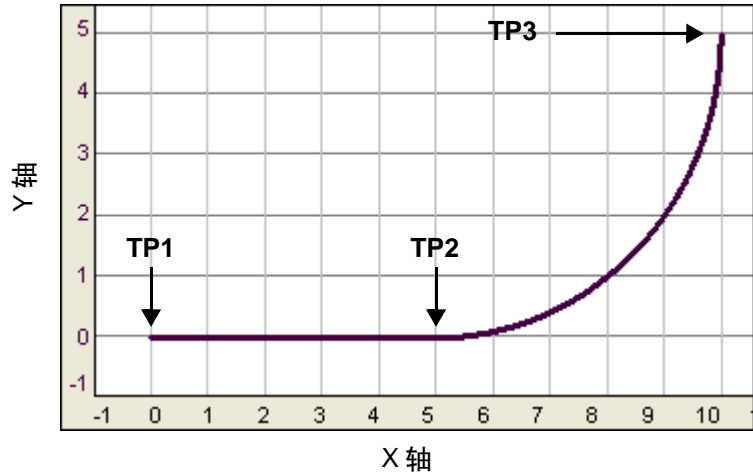
下表显示上图中终止类型为命令容差时各个转接点处的位状态。对于命令容差终止类型，转接点 TP2 的剩余移动量等于坐标系 cs1 的命令容差。

使用命令容差终止类型的位状态

位	TP1	TP2	TP3	TP4
Move1.DN	T	T	T	T
Move1.IP	T	F	F	F
Move1.AC	T	F	F	F
Move1.PC	F	T	T	T
Move2.DN	T	T	T	T
Move2.IP	T	T	T	F
Move2.AC	F	T	T	F
Move2.PC	F	F	F	T
cs1.MoveTransitionStatus	F	T	F	F
cs1.MovePendingStatus	T	F	F	F
cs1.MovePendingQueueFullStatus	T	F	F	F

使用轮廓跟踪速度限制或无限制的混合运动转接点处的位状态

线性 → 圆形运动



下表显示转接点处的位状态。

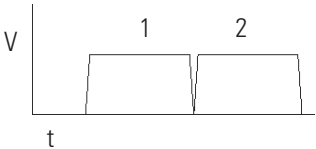
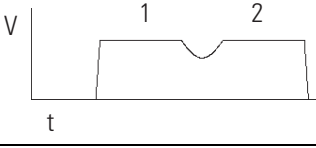
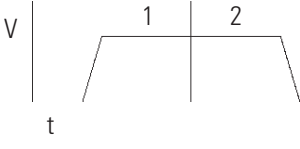
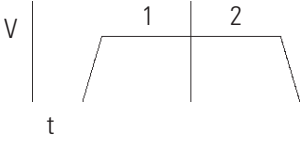
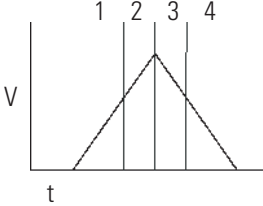
采用轮廓跟踪速度限制或无限制终止类型的位状态

位	TP1	TP2	TP3
Move1.DN	T	T	T
Move1.IP	T	F	F
Move1.AC	T	F	F
Move1.PC	F	T	T
Move2.DN	T	T	T
Move2.IP	T	T	F
Move2.AC	F	T	F
Move2.PC	F	F	T
cs1.MoveTransitionStatus	F	F	F
cs1.MovePendingStatus	T	F	F
cs1.MovePendingQueueFullStatus	T	F	F


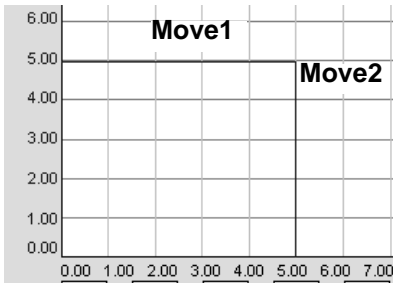
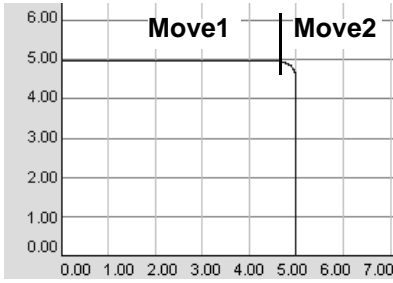
选择终止类型

终止类型决定了指令完成的时机。它同时决定了指令如何将其路径混合到排队的 MCLM 或 MCCM 指令（如果有）中。

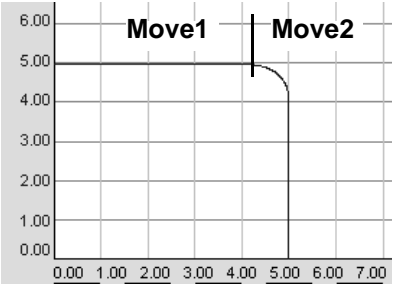


1. 选择终止类型。

如果您需要轴 (矢量速度)	您希望指令完成的时机是	则使用这种终止类型
在运动之间停止 	二者同时发生： <ul style="list-style-type: none"> 命令位置等于目标位置。 目标和实际位置之间的矢量距离小于或等于坐标系的实际位置容差。 	0 - 实际容差
保持速度恒定，除非在运动之间 	命令位置等于目标位置。	1 - 无安排
	命令位置进入坐标系的命令位置容差以内。	2 - 命令容差
过渡进入或离开某个圆而不停止 	轴到达了它必须以减速度进行减速的那个点。	3 - 无减速
过渡进入或离开某个圆而不停止 		4 - 轮廓跟踪速度限制
在多个运动之间加速或减速 		5 - 轮廓跟踪速度无限制

2. 确定这是对您正确的选择。

终止类型	示例路径	描述
0 - 实际容差		<p>指令保持活跃直至二者同时发生：</p> <ul style="list-style-type: none"> 命令位置等于目标位置。 目标和实际位置之间的矢量距离小于或等于坐标系的实际位置容差。 <p>在该点处指令完成，排队的 MCLM 或 MCCM 指令可以开始。</p> <p>重要事项： 确定要将实际容差设置为您的轴可以到达的值。否则该指令将一直停留在正在进行状态。</p>
1 - 无安排		<p>指令将保持活跃状态，直到命令位置等于目标位置。在该点处指令完成，排队的 MCLM 或 MCCM 指令可以开始。</p>
2 - 命令容差		<p>指令将保持活跃状态，直到命令位置进入坐标系的命令容差以内。在该点处指令完成，排队的 MCLM 或 MCCM 指令可以开始。</p> <p>如果没有排队的 MCLM 或 MCCM 指令，轴将在目标位置停止。</p>

RSLogix 5000 软件比较	到达	使用	用于
100 使用命令容差终止类型的第一条指令的配置长度 %	为坐标系配置的命令容差	两个长度的较短者	用于 第一条 指令的命令容差
100 使用命令容差终止类型的最后一条指令的配置长度 %	为坐标系配置的命令容差	两个长度的较短者	用于 倒数第二条 指令的命令容差
50 所有其他运动控制指令的各长度 %	为坐标系配置的命令容差	两个长度的较短者	用于 每一条 指令的命令容差

终止类型	示例路径	描述
3 - 无减速	 <p>The graph shows two moves: Move1 from x=0.00 to x=4.00 at y=5.00, and Move2 from x=4.00 to x=5.00 at y=5.00. At x=4.00, there is a sharp vertical drop to y=0.00, indicating a stop at the end of the first move.</p>	<p>指令将保持活跃状态，直到轴到达减速点。在该点处指令完成，排队的 MCLM 或 MCCM 指令可以开始。</p> <ul style="list-style-type: none"> • 减速点取决于您使用的是梯形还是 S 曲线模式。 • 如果没有排队的 MCLM 或 MCCM 指令，轴将在目标位置停止。
4 - 轮廓跟踪速度限制	 <p>The graph shows three moves: Move1 from x=0.00 to x=3.00 at y=5.00, Move2 from x=3.00 to x=5.00 at y=5.00, and Move3 from x=5.00 to x=6.00 at y=2.50. A smooth curve transition occurs between Move1 and Move2, starting at x=3.00 and ending at x=5.00.</p>	<p>指令将保持活跃状态，直到轴到达目标位置。在该点处指令完成，排队的 MCLM 或 MCCM 指令可以开始。</p> <ul style="list-style-type: none"> • 这种中止类型最适合于切线过渡。例如，可用它来从直线到圆、从圆到直线或者从圆到圆。 • 轴跟随该路径。 • 运动的长度决定了轴的最大速度。如果运动足够长，轴将在运动之间不会减速。如果运动太短，轴将在运动之间减速。
5 - 轮廓跟踪速度无限制	 <p>The graph shows three moves: Move1 from x=0.00 to x=3.00 at y=5.00, Move2 from x=3.00 to x=5.00 at y=5.00, and Move3 from x=5.00 to x=6.00 at y=2.50. A smooth curve transition occurs between Move1 and Move2, starting at x=3.00 and ending at x=5.00.</p>	<p>这种终止类型类似于轮廓速度限制。其不同点在于：</p> <ul style="list-style-type: none"> • 使用这种终止类型来获得几个运动之间的切线速度模式。这样减少跃度。 • 为避免在最后一次运动结束时位置过冲，您必须计算在模式减速一半过程中每个转接点处的减速速度。 • 您还必须计算模式减速一半中每个运动的开始速度。

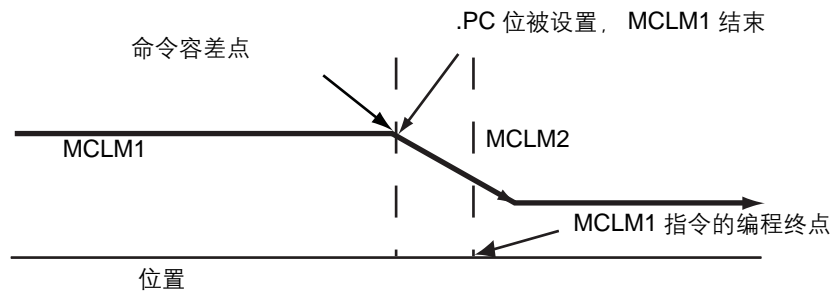
共线运动的速度模式

共线运动是在空间内同一条直线上的运动。它们的方向可以相同也可以相反。共线运动的速度模式可能十分复杂。本部分为您介绍几个例子和插图，帮助您理解使用 MCLM 指令编程的共线运动的速度模式。

采用终止类型 2 的共线运动速度模式

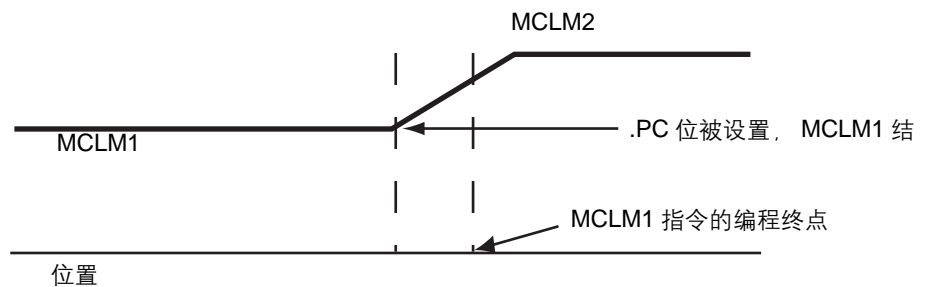
下图显示了使用命令容差 (2) 终止类型的两个共线运动的速度模式。第二条 MCLM 指令要比第一条 MCLM 指令的速度慢。当第一条 MCLM 指令到达其命令容差点时，运动结束，.PC 位被设置。

第二个运动比第一个运动速度慢而且采用终止类型 2 时的两个共线运动的速度模式



下图显示了使用命令容差 (2) 终止类型的两个共线运动的速度模式。第二条 MCLM 指令要比第一条 MCLM 指令的速度快。当第一条 MCLM 指令到达其命令容差点时，运动结束，.PC 位被设置。

第二个运动比第一个运动速度快而且采用终止类型 2 时的两个共线运动的速度模式



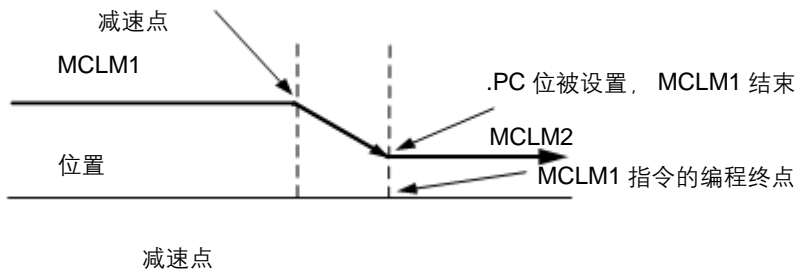
采用终止类型 3、4 或 5 的共线运动速度模式

此图显示两个共线运动的速度模式。第二条 MCLM 指令要比第一条 MCLM 指令的速度慢，使用下面一种终止类型：

- 无减速 (3)
- 轮廓跟踪速度限制 (4)
- 轮廓跟踪速度无限制 (5)

当第一条 MCLM 指令到达减速点时，它减速至第二个运动的编程速度。第一个运动结束，.PC 位被设置。

第二个运动比第一个运动速度慢而且采用终止类型 3、4 或 5 时的两个共线运动的速度模式

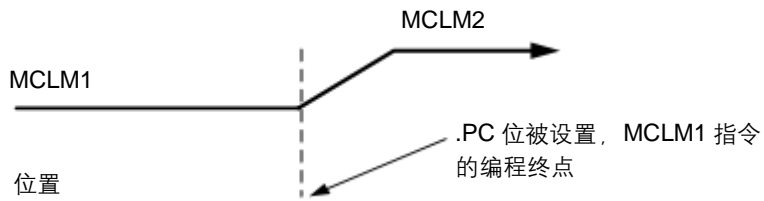


此图显示两个共线运动的速度模式。第二条 MCLM 指令要比第一条 MCLM 指令的速度快，使用下面一种终止类型：

- 无减速 (3)
- 轮廓跟踪速度限制 (4)
- 轮廓跟踪速度无限制 (5)

当第一个运动到达其编程终点时，.PC 位被设置。

第二个运动比第一个运动速度快而且采用终止类型 3、4 或 5 时的两个共线运动的速度模式



对称模式

模式路径对于所有运动模式都是对称的。

在正向和反向方向上以对称的方式编程速度、加速度和减速度值，可在正向方向点 A 到点 C 与反向方向上点 C 到点 A 产生相同的路径。

虽然这个概念可通过两个指令序列最简单地显示出来，但只要以对称方式进行编程，可应用到任何长度的指令序列上。

请参阅对称模式示例了解更多信息。

对称模式示例

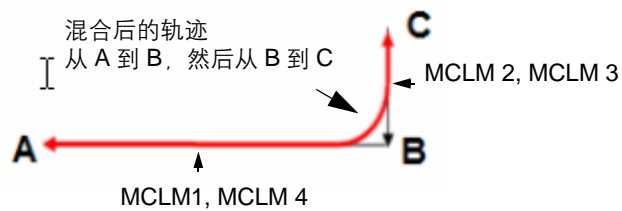
- MCLM 1(点 A 到点 B) 后跟 MCLM 2(点 B 到点 C)。
- MCLM 3(点 C 到点 B) 后跟 MCLM 4(点 B 到点 A)。
- MCLM 1 的加速度必须等于 MCLM 4 的减速度。
- MCLM 1 的减速度必须等于 MCLM 4 的加速度。
- MCLM 2 的加速度必须等于 MCLM 3 的减速度。
- MCLM 2 的减速度必须等于 MCLM 3 的加速度。

MCLM 1 (Pos = [2,0], Accel = 1, Decel = 2)

MCLM 2 (Pos = [2,1], Accel = 3, Decel = 4)

MCLM 3 (Pos = [2,0], Accel = 4, Decel = 3)

MCLM 4 (Pos = [0,0], Accel = 2, Decel = 1)

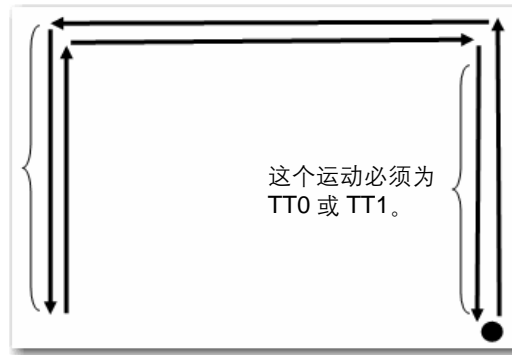


重要提示

我们建议您通过终止类型 0 或 1（即 TT0 或 TT1）来终止任何运动序列。

为保证您的轨迹是对称的，您必须通过终止类型 0 或 1 来终止任何运动序列。同时在自己回退运动的模式上的转向点也应使用终止类型 0 或 1。

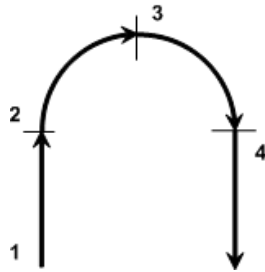
这个运动必须为 TT0 或 TT1。



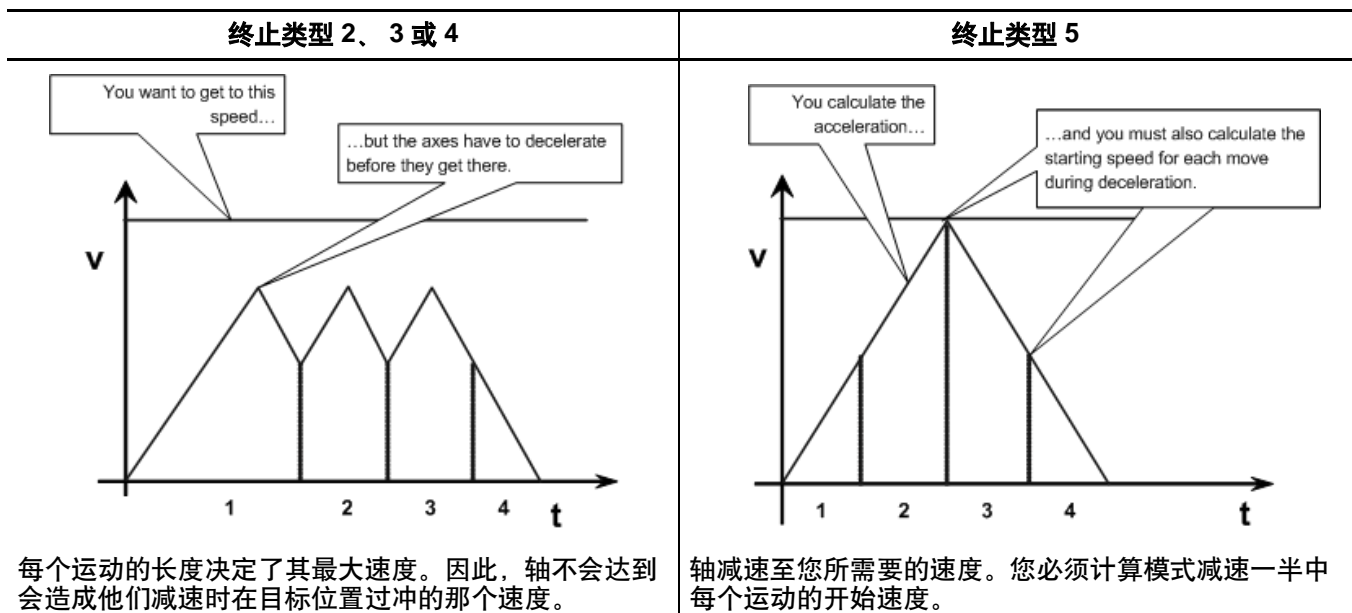
转向点

使用 TT2、TT3、TT4 或 TT5 作为模式（或转向点）的最后一个运动是安全的。但是，所产生的从 A 到 B 的轨迹并不一定始终与从 B 到 A 的轨迹相同。运动序列的显性终止可帮助控制器优化速度模式，降低 CPU 负荷，并保证模式的对称性。

如何得到切线速度模式 如果您想要通过四个运动编写一个拾取和放置操作，将跃度降至最低，使用切线速度模式。



然后，使用终止类型 5。其他的终止类型可能无法让您得到所需的速度。



以不同速度混合运动

您可以混合 MCLM 和 MCCM 指令，其中第二条指令的矢量速度不同于第一条指令的矢量速度。

如果下一个运动	而且第一个运动的终止类型为	那么
更慢	2 - 命令容差 3 - 无减速 4 - 轮廓速度限制 5 - 轮廓速度无限制	
更快	2 - 命令容差 3 - 无减速	
	4 - 轮廓速度限制 5 - 轮廓速度无限制	

运动直线插补 (MCLM)

使用 MCLM 指令可在笛卡尔坐标系内的指定轴上发起一维或多维线性协调运动。您可以将新位置定义为绝对位置或者增量位置。

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或者在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。



MCLM		
Motion Coordinated Linear Move		
Coordinate System	?	...
Motion Control	?	
Move Type	?	
	??	
Position	?	...
Speed	?	
	??	
Speed Units	?	
Accel Rate	?	
	??	
Accel Units	?	
Decel Rate	?	
	??	
Decel Units	?	
Profile	?	
Accel Jerk	?	
	??	
Decel Jerk	?	
	??	
Jerk Units	?	
Termination Type	?	
	??	
Merge	?	
Merge Speed	?	
<< Less		

运动直线插补 (MCLM) 指令使用笛卡尔坐标系中最多三个 (3) 静态耦合为主轴的轴执行线性运动。您可以指定使用绝对目标位置或者增量目标位置、期望的速度、最大加速度、最大减速度、加速度跃度、减速度跃度以及各自的单位。实际的速度是速度编程单位（单位每秒，或者最大值的 %，根据坐标系的配置）以及接受运动命令的主轴组合的函数。每个轴接受命令以某个速度运动，让所有轴同时到达编程终点（目标位置）。

操作数 - 梯形图

操作数	类型	格式	描述
坐标系	COORDINATE_SY STEM	标签	轴的协调组。
运动控制	MOTION_ 指令	标签	用于访问指令状态参数的结构。
运动类型	SINT、INT 或 DINT	立即或标 签	选择运动类型： 0 = 绝对值 1 = 增量
位置	REAL	数组标签 []	[协调单位]
速度	SINT、INT、DINT 或 REAL	立即或标 签	[协调单位]
速度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 1 = 最大值 %
加速度	SINT、INT、DINT 或 REAL	立即或标 签	[协调单位]
加速度单 位	SINT、INT 或 DINT	立即	0 = 单位每秒 ² 1 = 最大值 %
减速度	SINT、INT、DINT 或 REAL	立即或标 签	[协调单位]
减速度单 位	SINT、INT 或 DINT	立即	0 = 单位每秒 ² 1 = 最大值 %
模式	SINT、INT 或 DINT	立即	0 = 梯形 1 = S 曲线

操作数 - 梯形图

操作数	类型	格式	描述
加速跃度	SINT、INT、DINT 或 REAL	立即或标签	您 必须 始终为加速度和减速度跃度操作数输入值。如果模式操作数配置为 S 曲线，这个指令仅使用该值。 用以下跃度单位输入跃度值。 0 = 单位每秒 ³ 1 = 最大值 % 2 = 时间 % 使用这些值起步。 <ul style="list-style-type: none"> • 加速度跃度 = 100 (时间 %) • 减速度跃度 = 100 (时间 %) • 跃度单位 = 2
减速度跃度	SINT、INT、DINT 或 REAL	立即或标签	
跃度单位	SINT、INT 或 DINT	立即或标签	
终止类型	SINT、INT 或 DINT	立即或标签	0 = 实际容差 1 = 无安排 2 = 命令容差 3 = 无减速 4 = 轮廓跟踪速度限制 5 = 轮廓跟踪速度无限制 参见 选择终止类型 （在 页码 257 上）。
合并	SINT、INT 或 DINT	立即	0 = 已禁止 1 = 协调运动 2 = 全部运动
合并速度	SINT、INT 或 DINT	立即	0 = 编程值 1 = 当前值



MCLM(CoordinateSystem,
MotionControl,MoveType,
Position,Speed,Speedunits,
Accelrate,Accelunits,
Decelrate,Decelunits,Profile,
Acceljerk,Deceljerk, Jerkunits,
TerminationType,Merge,
Mergespeed);

结构文本

结构文本的操作数与梯形图 MCLM 指令的操作数相同。

当您输入枚举作为结构文本中的操作数值时，多个字枚举的输入**必须**没有空格。举例如下：当输入减速度单位时，值应该输入为 unitspersec² 而不是梯形图逻辑中显示的 Units per Sec²。

在输入结构文本操作数时，请将这个表格中的输入作为指南。

结构文本操作数的输入

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
坐标系	无枚举	标签
运动控制	无枚举	标签
运动类型	无枚举	0(绝对值) 1(增量)
位置	无枚举	数组标签
速度	无枚举	立即或标签
速度单位	单位每秒 最大值 %	0 1
加速度	无枚举	立即或标签
加速度单位	单位每秒 ² 最大值 %	0 1
减速度	无枚举	立即或标签
减速度单位	单位每秒 ² 最大值 %	0 1
模式	梯形 S 曲线	0 1

结构文本操作数的输入

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
加速跃度	无枚举	立即或标签
减速跃度	无枚举	您必须始终为加速度和减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这些值起步。 <ul style="list-style-type: none"> • 加速度跃度 = 100 (时间 %) • 减速度跃度 = 100 (时间 %) • 跃度单位 = 2
跃度单位	Unitspersec ³	0
	%ofmaximum	1
	%oftime	2(使用这个值起步)
终止类型	无枚举	0 = 实际容差 1 = 无安排 2 = 命令容差 3 = 无减速 4 = 轮廓跟踪速度限制 5 = 轮廓跟踪速度无限制 参见 选择终止类型 (在 页码 257 上)。
合并	禁止	0
	Coordinatedmotion	1
	Allmotion	2
合并速度	编程值	0
	当前值	1

坐标系

坐标系操作数指定了定义笛卡尔坐标系维度的运动轴系。对于这个版本，坐标系支持最大三个 (3) 主轴。只有配置为主轴的那些轴将包含在协调速度的计算中。

运动控制

以下控制位将受到 MCLM 指令的影响。

受到 MCLM 指令影响的控制位

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时, 启用位被设置, 当梯级从真过渡到假时, 启用位被复位。
.DN (完成) 位 29	当协调指令被成功验证并排队后, 完成位被设置。由于它是在排队后被设置, 因此在它离开队列后校对操作时, 可能会因为运行时间错误而显示为被设置。当梯级从假过渡到真时, 完成位被复位。
.ER (错误) 位 28	当梯级从假过渡到真时, 错误位被复位。当协调运动未成功发起时, 错误位被设置。当排队的指令遇到运行时间错误时, 错误位也可能与完成位一起被设置。
.IP(正在进行) 位 26	当协调运动被成功发起后, 正在进行位被设置。当没有后续运动而且协调运动到达新位置, 或者有后续运动而且协调运动到达终止类型的指定, 或者协调运动被其他合并类型为协调运动的 MCLM 或 MCCM 指令所取代, 或者被 MCS 指令终止时, 正在进行位将被复位。
.AC(活动) 位 23	当您的协调运动指令已经排队时, 活动位可让您了解哪个指令在控制运动。当协调运动变为活跃时, 活动位被设置。当过程完成位被设置, 或者当指令被停止时, 活动位被复位。
.PC(过程完成) 位 27	当梯级从假过渡到真时, 过程完成位被复位。当没有后续运动而且协调运动到达新位置, 或者有后续运动而且协调运动到达终止类型的指定时, 过程完成位将被复位。
.ACCEL(加速) 位 01	当协调运动处于加速阶段时, 加速位被设置。当协调运动处于恒定速度或减速阶段, 或者协调运动结束时, 加速位被复位。
.DECEL(减速) 位 02	当协调运动处于减速阶段时, 减速位被设置。当协调运动处于恒定速度或加速阶段, 或者协调运动结束时, 减速位被复位。

运动类型

运动类型操作数指定了用于指示协调运动路径的方法。运动类型可以为绝对值或者增量。

- 绝对值 - 轴通过一条直线路径按照由操作数指定的速度、加速度和减速度运动到由位置数组定义的位置。

当轴被配置为旋转工作时，绝对值运动类型的行为模式与线性轴相同。当轴位置超过退绕参数时，它将退绕。这样，轴的位置永远不会大于退绕值，也不会小于零。

指定位置的符号由内插器解释，可以为正或者为负。负的位置值将指使内插器向负方向运动旋转轴以获得所期望的绝对位置，而正值则表示需要正向运动来达到目标位置。当位置值大于退绕值时，将产生错误。轴永远不会穿过一个以上退绕循环，而会停止在一个绝对位置处。

- 增量 - 坐标系沿直线路径以指定的速度，使用由各自操作数确定的加速度和减速度，移动由位置数组所定义的距离。

指定的距离由内插器解释，可以为正或者为负。负的位置值将指使内插器向负方向运动轴，而正值则表示需要正向运动来达到目标位置。在增量模式下，允许超过一个退绕循环的运动。

MCLM 运动类型示例

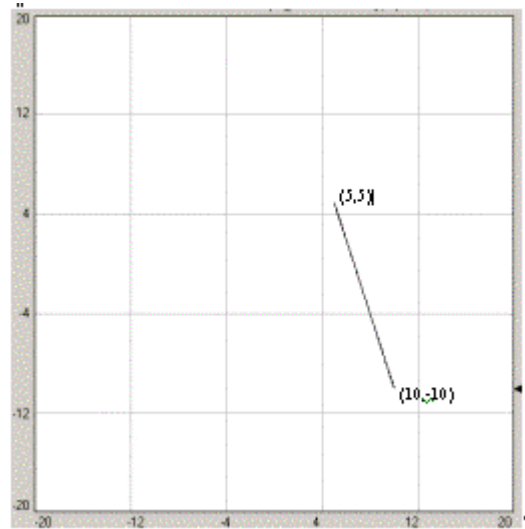
下面的示例介绍使用 MCLM 以及绝对值（第一个例子）和增量（第二个例子）运动类型来达到相同的结果。基本假设如下：

- 两个轴 Axis0 和 Axis1，都属于坐标系 coordinate_sys。
- Axis0 和 Axis1 彼此正交。
- Coordinated_sys 最初位于 (5,5) 单位。

Coordinated_sys 直线运动到 (10,-10) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。

下图是上面的假设所生成的路径。

路径的结果图



沿着矢量路径行进的总距离为：

$$D_{Axis0} = 10 - 5 = 5$$

$$D_{Axis1} = -10 - 5 = -15$$

$$TotalDist = \sqrt{(D_{Axis0})^2 + (D_{Axis1})^2} = 15.811388$$

选定轴的矢量速度等于指定的速度，单位是位置单位每秒。每个轴的速度与该轴行进的距离除以所有轴运动距离平方之和的平方根成正比。Axis0 的实际速度为运动矢量速度的以下百分比。

$$\%Axis0 \text{ 速度} = |Axis0 \text{ 距离} / \text{总距离}| = |5 / 15.811388| = .3162 = 31.62\%$$

$$\%Axis1 \text{ 速度} = |Axis1 \text{ 距离} / \text{总距离}| = |-15 / 15.811388| = .9487 = 94.87\%$$

对于本例,

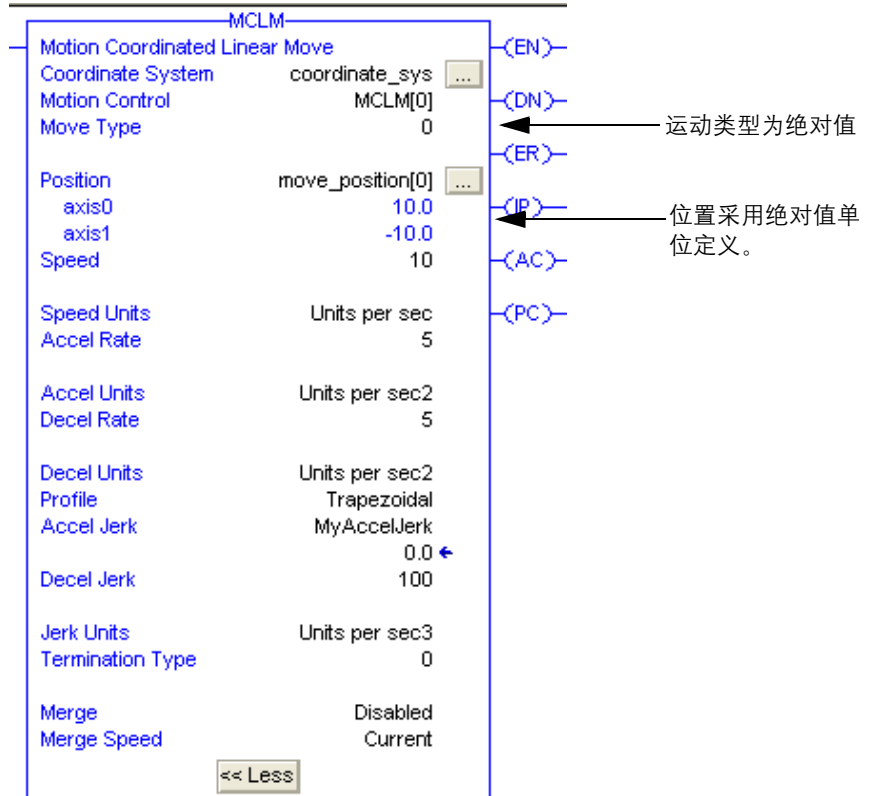
Axis0 速度 = $.3162 * 10.0 = 3.162$ 单位 / 秒

Axis1 速度 = $.9487 * 10.0 = 9.487$ 单位 / 秒

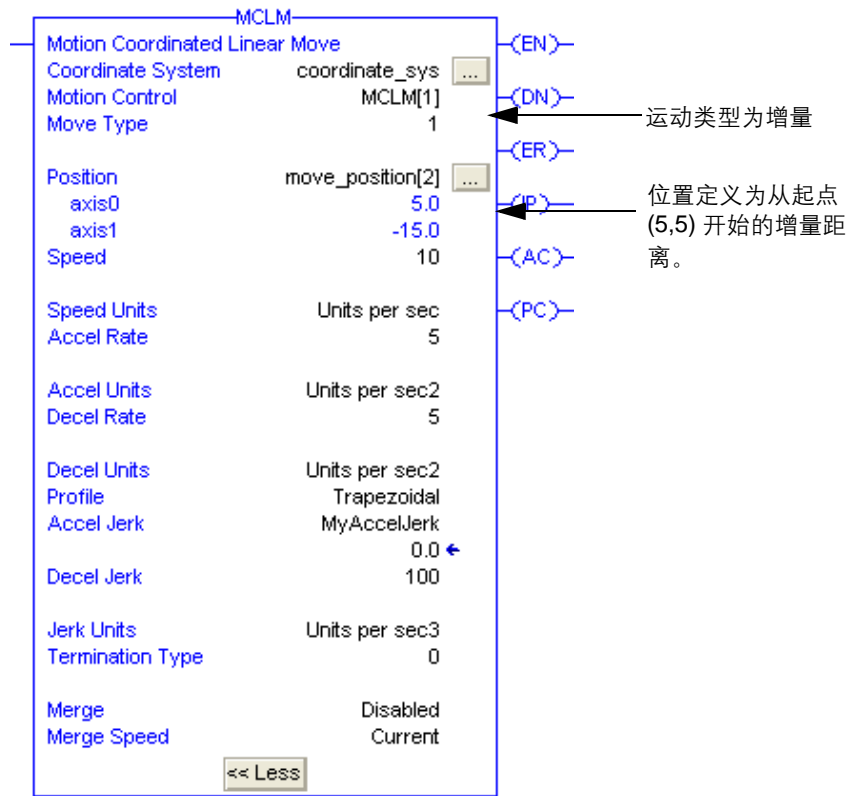
每个轴的加速度和减速度与速度的百分比相同。

以下梯形图指令显示分别使用运动类型 = 绝对值和运动类型 = 增量的情况下, 要实现这个路径所需的梯形图逻辑。

运动类型为绝对值时的 **MCLM** 梯形图指令



运动类型为增量时的 MCLM 梯形图指令



MCLM 指令与旋转轴示例

下面的例子介绍对定义为旋转轴的 MCLM 指令的路径图。

使用一个旋转轴和绝对值运动类型的 MCLM

第一个例子使用单轴坐标系和绝对值运动类型。路径图基于以下假设：

- 单轴坐标系，名为 coord_syst1
- Axis0 为旋转轴，退绕为 5 转。
- 开始位置为 4。
- 结束位置为 -2。

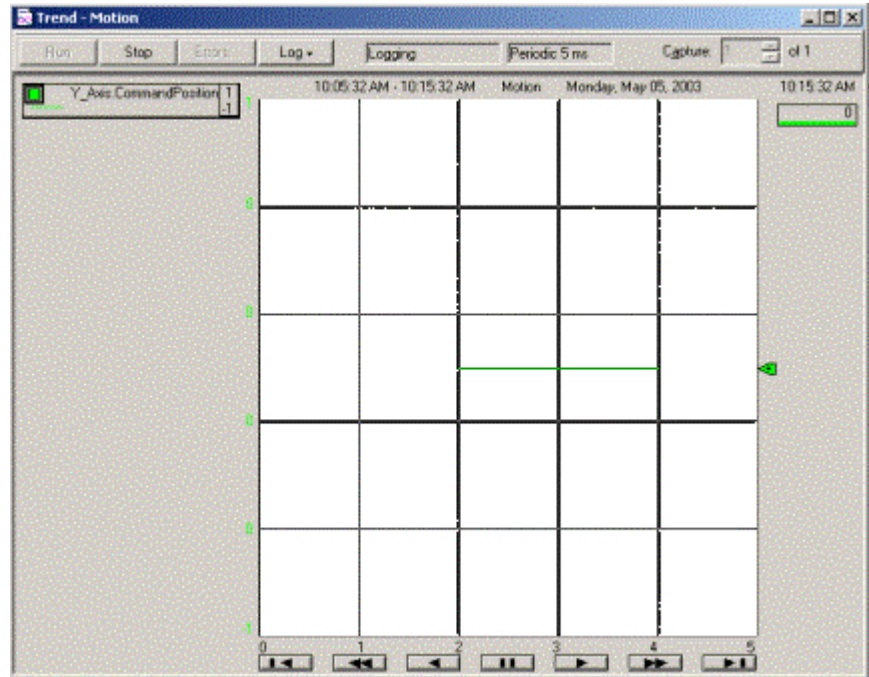
运动类型为绝对值时的 MCLM 梯形图指令

Parameter	Value
Motion Coordinated Linear Move	MCLM
Coordinate System	coord_syst1
Motion Control	MCLM[1]
Move Type	0
Position axis0	-2.0
Speed	1
Speed Units	Units per sec
Accel Rate	100
Accel Units	% of Maximum
Decel Rate	100
Decel Units	% of Maximum
Profile	S-Curve
Accel Jerk	MyAccelJerk
Decel Jerk	0.0
Jerk Units	Units per sec ³
Termination Type	3
Merge	Disabled
Merge Speed	Programmed

要记住对于绝对值运动类型 (0)，负号代表运动的方向。在这个例子中，轴向负方向运动到绝对值位置 +2.0。要向负方向移动到位置 0.0，您必须编写为 -360.0，因为 -0.0 在内部储存为 0.0。

运动路径的结果图如下图所示。

使用一个旋转轴和绝对值运动类型的 MCLM 路径图



终点为负值，因此轴按负方向从 4 运动到 2，不通过退绕方式行进。对于这个运动，终点要符合由轴的旋转退绕所定义的绝对值位置以内。因此，退绕值 6 或 -6 将是无效的。

使用两个旋转轴和增量运动类型的 MCLM

第二个 MCLM 的旋转轴示例为采用两个旋转轴和增量运动类型。路径图的假设如下：

- 二轴坐标系，名为 `coordinate_sys`
- Axis0 为旋转轴，退绕为 1 转。
- Axis1 为旋转轴，退绕为 2 转。
- 开始位置为 0,0。
- 到结束位置的增量为 -5,5。

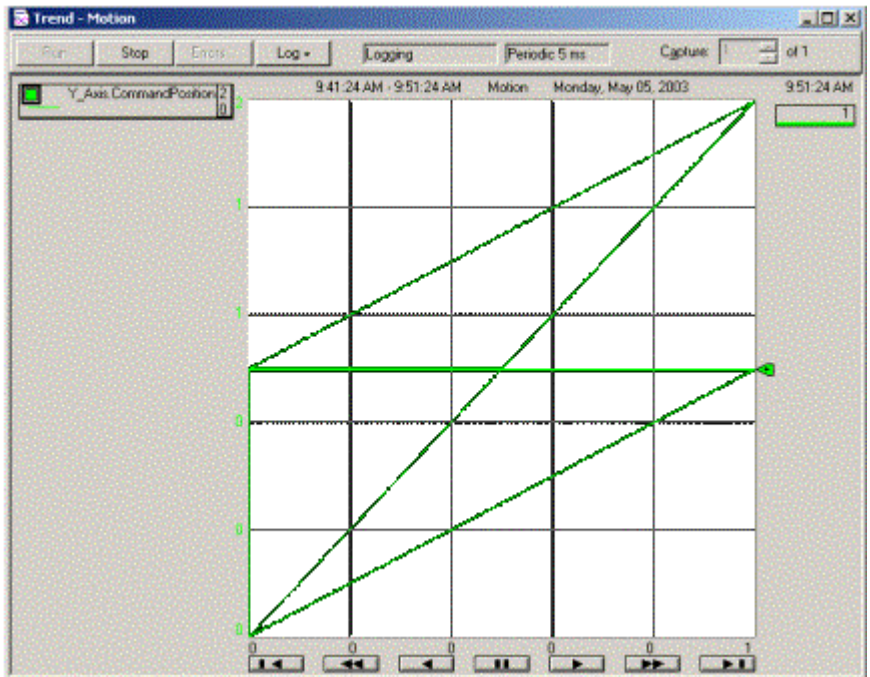
运动类型为增量时的 MCLM 梯形图指令

MCLM		(EN)
Motion Coordinated Linear Move		
Coordinate System	coordinate_sys [...]	(DN)
Motion Control	MCLM[3]	
Move Type	1	(ER)
Position	move_position[6] [...]	(IP)
axis0	5.0	
axis1	5.0	(AC)
Speed	1	(PC)
Speed Units	Units per sec	
Accel Rate	100	
Accel Units	% of Maximum	
Decel Rate	100	
Decel Units	% of Maximum	
Profile	S-Curve	
Accel Jerk	MyAccelJerk	
	0.0	
Decel Jerk	100	
Jerk Units	Units per sec ³	
Termination Type	3	
Merge	Disabled	
Merge Speed	Programmed	
<< Less		

运动类型为增量。

这条 MCLM 指令将产生下面的运动路径图。

使用两个旋转轴和增量运动类型的 MCLM 路径图



在上图中，轴沿着倒“z”字形运动了两圈半，停止的实际位置为 0,1。这等于 Axis0 的 5 转 / 退绕和 Axis1 的 2.5 转 / 退绕。这个运动的位置增量为正值，因此轴按正方向运动，其中 Axis0 从 0 运动到 1，Axis1 从 0 运动到 2。在这个例子中，终点需要符合由轴的旋转退绕所定义的绝对位置以内。协调运动的路径是在直线空间内确定的，但轴的位置则由旋转配置所限制的。

位置

一维数组，其维度定义为至少等于坐标系内指定的轴个数。位置数组定义了新的绝对值或者增量位置。

速度

速度操作数定义了协调运动路径上的最大矢量速度。

速度单位

速度单位操作数定义了应用到速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

加速度

加速度操作数定义了协调运动路径上的最大加速度。

加速度单位

加速度单位操作数定义了应用到加速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

减速度

减速度操作数定义了协调运动路径上的最大减速度。

减速度单位

减速度单位操作数定义了应用到减速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

模式

模式操作数确定了协调运动是使用梯形还是 S 曲线速度模式。

ControlLogix 运动控制器提供了梯形（线性加速和减速）以及 S 曲线（控制跃度）速度模式。下面给出了这些运动模式对各种应用要求的效果指南。

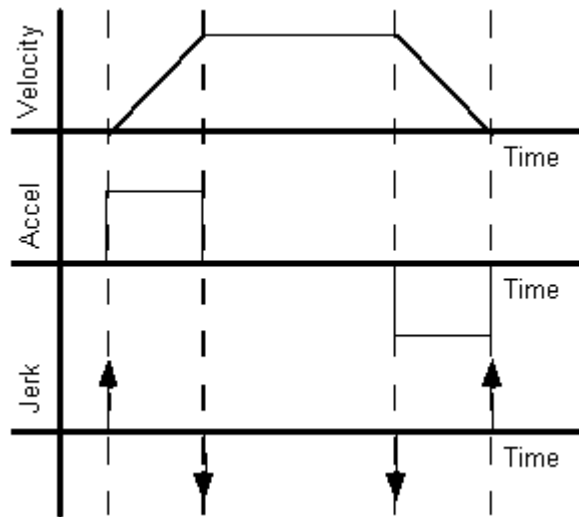
速度模式效果

模式 类型	加速 / 减速 时间	电机 压力	控制的优先级 最高到最低			
			加速 / 减速	速度	位置	位置
梯形	最快	最差	加速 / 减速	速度	位置	
S 曲线	2X 稍慢	最佳	跃度	加速 / 减速	速度	位置

- 梯形

梯形速度模式是最常用的模式，因为它对于编程后续运动提供最大的灵活性，并提供最快的加速和减速时间。速度的最大变化是由加速度和减速度所指定的。由于跃度不是梯形模式中的一种因素，因此被看做是无限大，在下图中显示为一系列的垂直线。

梯形加速 / 减速时间



- S 曲线

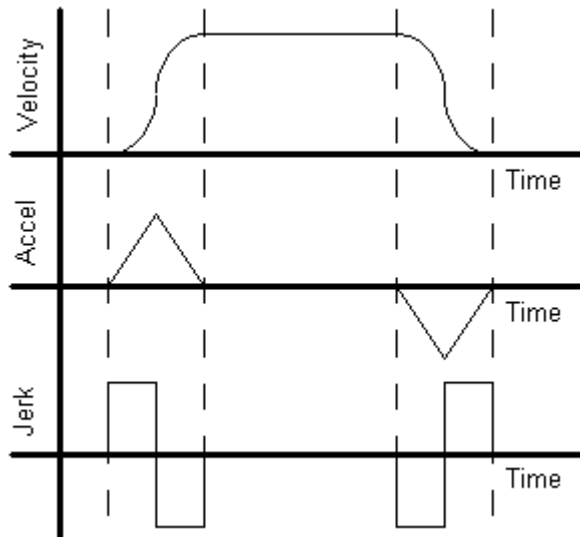
当需要将对机械系统的压力和负荷降至最低时，S 曲线速度模式最为常用。但是与梯形模式相比，S 曲线模式牺牲了加速和减速时间。速度可以加快或减慢的最大速率还进一步受到跃度的限制。

协调运动的加速和减速跃度值计算将在这些指令开始时进行。

- MAJ
- MCS
- MAM
- MCCD
- MAS
- MCCM
- MCD
- MCLM

计算出的跃度值生成三角形加速和减速模式，如下图所示。

S 曲线加速 / 减速时间



对于 S 曲线运动，跃度值是基于编程速度、加速度和减速度值所确定的，**而不是运动的长度**。当混合具有相同加速度和减速度值的运动时，即使运动可能不够长到能达到编程速度（速度限制运动），RSLogix 5000 软件也尝试保持稳定的跃度值。

如果 S 曲线运动被配置为	则增加速度
非速度限制	缩短运动的执行时间
速度限制	延长运动的执行时间

对于用零速度编程的 S 曲线运动，跃度值是由以前采用非零速度的指令所编程的速度值所决定的。

请参阅 MCCD 指令，了解关于 MCCD 指令所做更改的影响。

加速跃度

加速跃度定义了编程运动的最大加速跃度。有关计算加速跃度的更多信息，请参阅本章的“跃度单位”部分。

减速跃度

减速跃度定义了编程运动的最大减速跃度。有关计算减速跃度的更多信息，请参阅本章的“跃度单位”部分。

跃度单位

跃度单位定义了向加速跃度和减速跃度操作数所输入的值上应用的单位。所输入的值直接以指定坐标系的位置单位为单位，或者为百分比。当配置使用最大值 % 时，跃度被应用为坐标系属性中指定的最大加速跃度和最大减速跃度操作数的百分比。当配置为使用时间 % 时，该值为基于指令中指定的速度、加速度和减速度的一个百分比。

如果要将工程单位换算成时间 % 或者将时间 % 换算成工程单位，请使用[页码 28](#) 开头处介绍的公式。

终止类型

有关选择终止类型的信息，请参阅[页码 257](#)。

合并

合并操作数确定了是否要将所有指定轴的运动变成单纯的协调运动。合并选项包括：禁止合并、协调运动或全部运动。

- 禁止合并

任何当前正在执行的单轴运动指令如果涉及到指定坐标系中定义的任何轴，则不受此指令激活的影响，并且在受到影响的轴上产生叠加运动。同时，同一个指定坐标系中涉及的任何协调运动指令都基于其终止类型运行至完成。

- 协调运动

同一个指定的坐标系中涉及到的任何当前正在执行的协调运动指令都被终止。激活的运动混合进入当前的运动，并且按照合并速度参数中定义的速度。任何挂起的协调运动指令都被取消。任何当前正在执行的系统单轴运动指令如果涉及到指定坐标系中定义的任何轴，则不受此指令激活的影响，并且在受到影响的轴上产生叠加运动。

- 全部运动

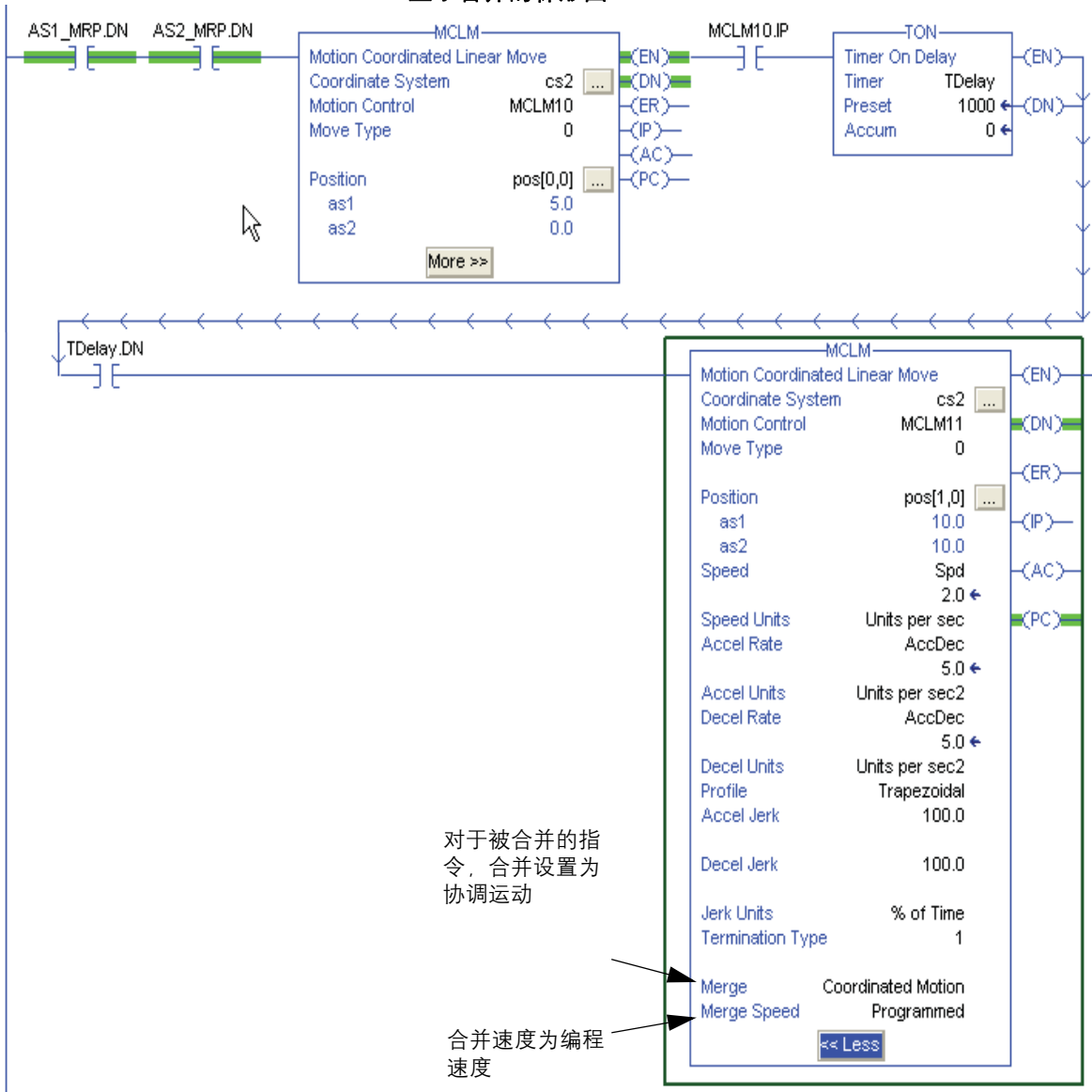
任何涉及到指定坐标系中定义的任何轴的当前正在执行的单轴运动指令，以及任何当前正在执行的协调运动指令，都将被终止。前面的运动混合进入当前的运动，并且按照合并速度参数中定义的速度。任何挂起的协调运动指令都被取消。

合并速度

合并速度操作数定义了当合并被使能时，在协调运动的路径上是当前速度还是将编程速度用作最大速度。

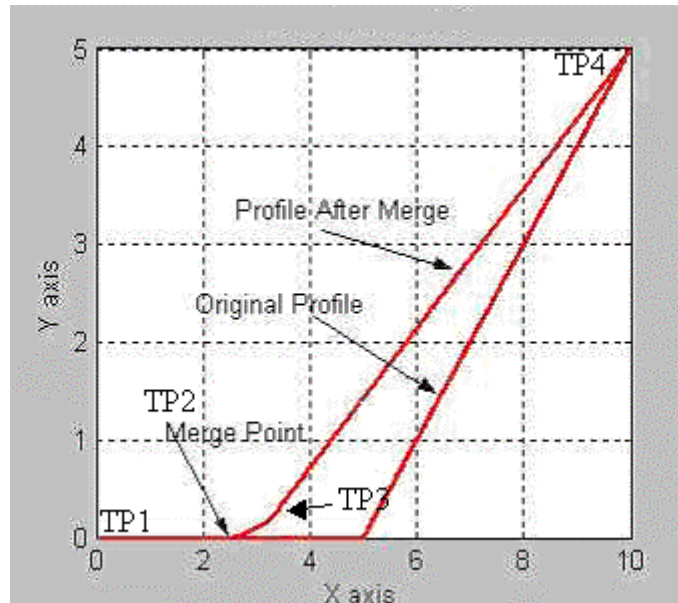
合并示例

MCLM 梯形图使用坐标系 cs2 将一条采用目标绝对值位置 (5,0) 的 mclm10 指令合并到一条目标位置为 (10,5) 的 mclm11 指令。
显示合并的梯形图



如果轴彼此正交，而且坐标系 cs2 最初位于 (0,0) 单位，则此图所产生的运动将取决于第二条指令执行所需的时间。第二个运动一旦开始就会立即进行混合，而且第一个运动立即终止。在此例的梯形图中，当定时器 Tdelay 期满后就开始过渡。

显示合并结果的图形



合并运动各个转接点处的位状态

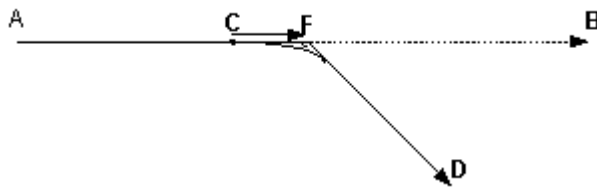
位	TP1	TP2	TP3	TP4
Move1.DN	T	T	T	T
Move1.IP	T	F	F	F
Move1.AC	T	F	F	F
mclm10.PC	F	T	T	T
Move2.DN	T	T	T	T
Move2.IP	T	T	T	F
Move2.AC	F	T	T	F
Move2.PC	F	F	F	T
cs2.MoveTransitionStatus	F	T	F	F
cs2.MovePendingStatus	T	F	F	F
cs2.MovePendingQueueFullStatus	T	F	F	F

目前，协调运动仅支持一个协调运动指令的排队。因此，MovePendingStatus 位和 MovePendingQueueFullStatus 位始终是相同的。

有关合并指令的附加信息

从点 A 到点 B 的运动被发起，如下图所示。当轴位于点 C 时，发起向点 D 的增量合并。结果，当前指令在点 C 处被终止。控制器计算沿着矢量 AB 从当前速度到零速度在点 C 处需要的减速距离。这个距离显示为矢量 CF。然后，通过将矢量 CF 加到点 C 上计算出假想点 F。所产生的从 C 到 D 的合并运动如下图所示。运动所沿的曲线从 C 开始，然后加入从 F 到 D 的直线。点 D 是使用合并指令中增量数据从合并起点（点 C）计算出来的。这个路径等同于原始编程运动从点 A 到 F，然后从 F 到 D，并且采用无减速的终止类型。

合并示例



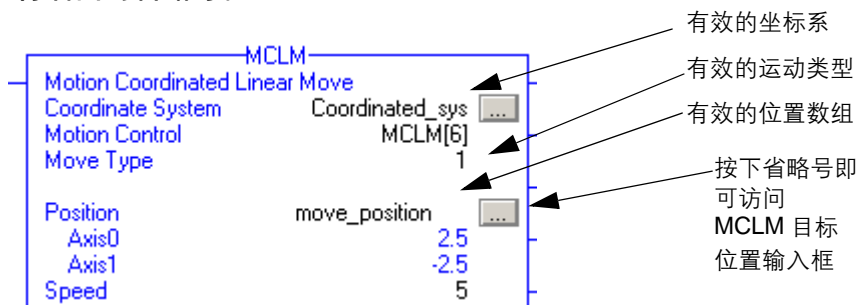
本示例应用于直线合并。

尝试合并一个环形运动如果结果路径并不定义一个圆，则可能导致编程错误。增量模式中的圆心是从点 C（合并点）计算出来的。但是，从点 F（计算出的减速终点）到合并运动的终点必须存在一个圆。

MCLM 目标位置输入对话框

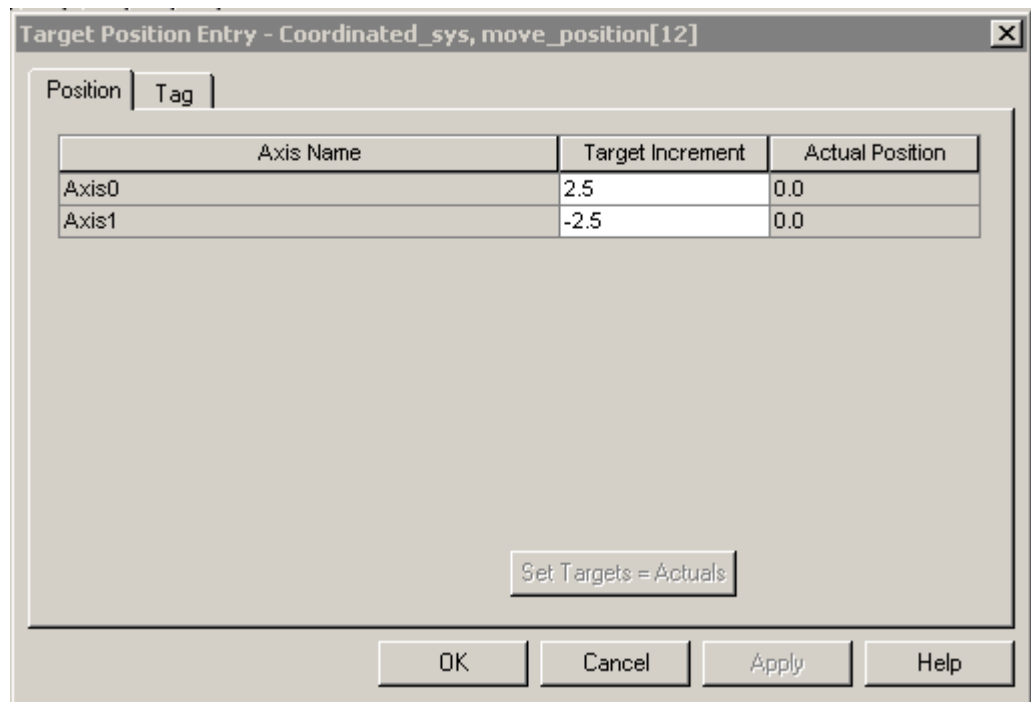
MCLM 指令的目标位置输入对话框提供一种简单的编辑位置的格式。要访问目标位置输入对话框，您必须已经将坐标系的名称插入指令，必须在位置字段中输入有效的标签名称而且有足够的元素来处理轴的个数，而且要选择有效的运动类型。

要访问 MCLM 指令的目标位置输入对话框，请在指令面板上按下位置行后面的省略号。



用于访问目标位置输入框的 MCLM 梯形图有效值

按下梯形图指令面板位置行上的省略号按钮，将调用下面的目标位置输入框，用于编辑位置值。



MCLM 指令目标位置输入对话框 - 位置选项卡

对话框标题显示指令的坐标系和标签名称。

目标位置输入对话框字段说明

功能	描述
轴名称 (Axis Name)	这些字段列出坐标系中所包含的各个轴的名称。在这个对话框中您不能更改轴的名称。
目标位置 / 目标增量 (Target Position/Target Increment)	这个字段包含在指令面板中指令的协调运动的终点或增量。该字段为数字。
实际位置 (Actual Position)	这些是坐标系中的轴的当前实际位置。当联机和坐标系自动标签更新被使能后，这些位置将动态更新。
设置目标 = 实际值 (Set Targets = Actuals) 按钮	这个按钮自动将实际位置值复制到目标位置列内。

所选择的运动类型控制着“设置目标 = 实际值”按钮的外观以及是否可用。

当运动类型为绝对值时，目标列的名称为“目标位置”，当运动类型为增量时，目标列的名称为“目标增量”，而且“设置目标 = 实际值”按钮被禁止（灰掉）。

MCLM 是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志

不受影响

故障条件

无

错误代码

参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码

扩展错误代码可帮助进一步定义为这条特殊指令所提供的错误消息。其行为取决于它们关联什么错误代码。

伺服关闭状态 (5)、关闭状态 (7)、轴类型非伺服 (8)、轴未配置 (11)、归零正在进行错误 (16) 以及非法轴数据类型 (38) 错误的扩展错误代码功能都是非常类似的。对于扩展错误代码将显示 $0 \dots n$ 的数字。这个数字是坐标系的索引，表示正处于错误状况中的轴。

对于错误代码轴未配置 (11)，还会有一个附加的值 -1，表示坐标系无法为协调运动设置该轴。

对于 MCLM 指令，错误代码 13 - 参数超出范围，扩展错误将返回一个数字，表示有问题的参数编号，这些参数在面板上以数字顺序从上到下（以零开始）列出。例如，2 表示运动类型的参数值有错误。

引用的错误代码和编号	扩展错误数字指示符	指令参数	描述
参数超出范围 (13)	2	运动类型	运动类型为小于 0 或者大于 1。
参数超出范围 (13)	3	位置	位置数组的长度不足以为坐标系中的所有轴提供位置。
参数超出范围 (13)	4	速度	速度小于 0。
参数超出范围 (13)	6	加速度	加速度小于或等于 0。
参数超出范围 (13)	8	减速度	减速度小于或等于 0。
参数超出范围 (13)	11	终止类型	终止类型为小于 0 或者大于 3。

对于错误代码 54 - 最大减速度值为零，如果扩展错误代码返回一个正数 ($0 \dots n$)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

MCLM 对状态位的更改

状态位提供一种监视运动控制指令进度的途径。有三种类型的状态位提供有关信息。

- 轴状态位
- 坐标系状态位
- 协调运动状态位

当 MCLM 指令发起后，状态位将经历以下变化。

轴状态位

位名称	含义
CoordinatedMotionStatus	当指令开始时设置。当指令结束时清除。

坐标系状态位

位名称	含义
MotionStatus	当 MCLM 指令正在工作而且坐标系连接到其关联轴时设置。

协调运动状态位

位名称	含义
AccelStatus	当矢量正在加速时设置。当混合正在进行或者矢量运动正在减速时清除。
DecelStatus	当矢量正在减速时设置。当混合正在进行或者矢量运动正在加速时清除。
ActualPosToleranceStatus	仅为实际容差终止类型设置。当满足以下两个条件时，这个位被设置。1) 内插已经完成。2) 到编程终点的实际距离小于所配置的坐标系实际容差值。当指令完成后，这个位保持设置状态。当新的指令开始后，这个位被复位。
CommandPosToleranceStatus	每当到编程终点的实际距离小于所配置的坐标系命令容差值时，对于所有终止类型这个位都会被设置。当指令完成后，这个位保持设置状态。当新的指令开始后，这个位被复位。
StoppingStatus	当 MCLM 指令发起时，停止状态位被清除。
MoveStatus	当 MCLM 开始轴运动时设置。当最后一个运动控制指令的 .PC 位设置时或者某个运动控制指令执行导致停止时清除。

协调运动状态位

位名称	含义
MoveTransitionStatus	当满足无减速或命令容差终止类型时设置。当混合共线运动时这个位不被设置，因为机器始终在路径上。当混合完成时、某个挂起指令的运动开始时或者某个运动控制指令的执行导致停止时被清除。表示不在路径上。
MovePendingStatus	当一个挂起协调运动指令处于指令队列时设置。当指令队列为空时清除。
MovePendingQueueFullStatus	当指令队列满时设置。当队列中有空容纳新的协调运动指令时清除。

目前，协调运动仅支持一个协调运动指令的排队。因此，MovePendingStatus 位和 MovePendingQueueFullStatus 位始终是相同的。

模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

运动圆弧插补(MCCM)

使用 MCCM 指令可在笛卡尔坐标系内的指定轴上发起二维或三维圆形协调运动。新的位置定义为绝对值或者增量位置，并且使用期望的速度。MCCM 的实际速度是运动模式的函数（给定速度或最大速度的百分比）。运动的速度是根据使用编程轴完成圆形运动所需的时间。每个轴接受命令以某个速度运动，让所有轴同时到达终点（目标位置）。

圆形的维度是由坐标系中包含的轴数定义的。例如，如果您的坐标系中包含三个轴，而某个 MCCM 指令的运动仅是二维的，则产生的运动仍被看做是一个三维弧形或圆形。

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。

MCCM

- Motion Coordinated Circular Move ? ... (EN)
- Coordinate System ? (DN)
- Motion Control ? (ER)
- Move Type ? (IP)
- Position ? ... (AC)
- Circle Type ? (PC)
- Via/Center/Radius ?
- Direction ?
- Speed ?
- Speed Units ?
- Accel Rate ?
- Accel Units ?
- Decel Rate ?
- Decel Units ?
- Profile ?
- Accel Jerk ?
- Decel Jerk ?
- Jerk Units ?
- Termination Type ?
- Merge ?
- Merge Speed ?

<< Less

MCCM 指令操作数 - 梯形图

操作数	类型	格式	描述
坐标系	COORDINATE_ SYSTEM	标签	轴的协调组。
运动控制	MOTION_ 指令	标签	用于访问指令状态参数的结构。
运动类型	SINT、 INT 或 DINT	立即或标签	0 = 绝对值 1 = 增量
位置	REAL	数组标签 []	[协调单位]
圆形类型	SINT、 INT 或 DINT	立即或标签	0 = 经由 1 = 中心 2 = 半径 3 = 中心增量
经由 / 中心 / 半径	REAL	数组标签 [] (经由 / 中心) 立即或标签 (半径)	[协调单位]
方向	SINT、 INT 或 DINT	立即或标签	2D 0 = CW 1 = CCW 2 = CW 完整 3 = CCW 完整 3D 最短 最长 最短完整 最长完整
速度	SINT、 INT、 DINT 或 REAL	立即或标签	[协调单位]
速度单位	SINT、 INT 或 DINT	立即	0 = 单位每秒 1 = 最大值 %
加速度	SINT、 INT、 DINT 或 REAL	立即或标签	[协调单位]
加速度单位	SINT、 INT 或 DINT	立即	0 = 单位每秒 ² 1 = 最大值 %
减速度	SINT、 INT、 DINT 或 REAL	立即或标签	[协调单位]

MCCM 指令操作数 - 梯形图

操作数	类型	格式	描述
减速度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 ² 1 = 最大值 %
模式	SINT、INT 或 DINT	立即	0 = 梯形 1 = S 曲线
加速跃度	SINT、INT、DINT 或 REAL	立即或标签	<p>您必须始终为加速度和减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。</p> <ul style="list-style-type: none"> 加速跃度是坐标系的加速跃度值。 减速度跃度是坐标系的减速度跃度值。 <p>用以下跃度单位输入跃度值。</p> <p>0 = 单位每秒³ 1 = 最大值 % 2 = 时间 %</p> <p>使用这些值起步。</p> <ul style="list-style-type: none"> 加速度跃度 = 100 (时间%) 减速度跃度 = 100 (时间%) 跃度单位 = 2
减速度跃度	SINT、INT、DINT 或 REAL	立即或标签	
跃度单位	SINT、INT 或 DINT	立即或标签	
终止类型	SINT、INT 或 DINT	立即或标签	0 = 实际容差 1 = 无安排 2 = 命令容差 3 = 无减速 4 = 轮廓跟踪速度限制 5 = 轮廓跟踪速度无限制 参见 选择终止类型 (在 页码 257 上)。
合并	SINT、INT 或 DINT	立即	0 = 已禁止 1 = 协调运动 2 = 全部运动
合并速度	SINT、INT 或 DINT	立即	0 = 编程值 1 = 当前值



MCCM(CoordinateSystem, MotionControl, MoveType, Position, CircleType, Via/Center/Radius, Direction, Speed, Speedunits, Accelrate, Accelunits, Decelrate, Decelunits, Profile, Accel Jerk, Deceljerk, Jerkunits, Terminationtype, Merge, Mergespeed);

结构文本

结构文本操作数与梯形图 MCCM 指令的操作数相同。

当输入枚举作为结构文本中的操作数值时，多个字枚举的输入必须没有空格。举例如下：当输入减速度单位时，值应该输入为 unitspersec^2 而不是梯形图逻辑中显示的 Units per Sec^2 。

在输入结构文本操作数时，请将这个表格中的输入作为指南。

结构文本操作数的输入

此操作数	拥有这些选项可让您输入为 ...																
	文本	或者为															
坐标系	无枚举	标签															
运动控制	无枚举	标签															
运动类型	无枚举	标签 0 = 绝对值 1 = 增量															
位置	无枚举	数组标签															
圆形类型	无枚举	标签 0 = 经由 1 = 中心 2 = 半径 3 = 中心增量															
经由 / 中心 / 半径	无枚举	数组标签 (经由 / 中心) 立即或标签 (半径)															
方向	无枚举	<table border="1"> <thead> <tr> <th></th> <th>2D</th> <th>3D</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>顺时针</td> <td>最短</td> </tr> <tr> <td>1</td> <td>逆时针</td> <td>最长</td> </tr> <tr> <td>2</td> <td>顺时针完整</td> <td>最短完整</td> </tr> <tr> <td>3</td> <td>逆时针完整</td> <td>最长完整</td> </tr> </tbody> </table>		2D	3D	0	顺时针	最短	1	逆时针	最长	2	顺时针完整	最短完整	3	逆时针完整	最长完整
	2D	3D															
0	顺时针	最短															
1	逆时针	最长															
2	顺时针完整	最短完整															
3	逆时针完整	最长完整															
速度	无枚举	立即或标签															
速度单位	Unitspersec	0															
	%ofmaximum	1															
加速度	无枚举	立即或标签															

结构文本操作数的输入

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
加速度单位	Unitspersec ²	0
	%ofmaximum	1
减速度	无枚举	立即或标签
减速度单位	Unitspersec ²	0
	%ofmaximum	1
模式	梯形	0
	S 曲线	1
加速跃度	无枚举	立即或标签
减速跃度	无枚举	您必须始终为加速度和减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这些值起步。 <ul style="list-style-type: none"> • 加速度跃度 = 100 (时间 %) • 减速度跃度 = 100 (时间 %) • 跃度单位 = 2
跃度单位	Unitspersec ³	0
	%ofmaximum	1
	%oftime	2(使用这个值起步)

结构文本操作数的输入

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
终止类型	无枚举	0 = 实际容差 1 = 无安排 2 = 命令容差 3 = 无减速 4 = 轮廓跟踪速度限制 5 = 轮廓跟踪速度无限制 参见 选择终止类型 （在 页码 257 上）。
合并	禁止 Coordinatedmotion Allmotion	0 1 2
合并速度	编程值 当前值	0 1

坐标系

坐标系操作数指定了定义笛卡尔坐标系维度的运动轴系统。对于这个版本，坐标系支持最大三个 (3) 主轴。只有配置为主轴（最多三个）的那些轴将包含在速度的计算中。只有主轴将参与实际的圆形运动。

运动控制

以下控制位将受到 MCCM 指令的影响。

受到 MCCM 指令影响的控制位

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时, 启用位被设置。当梯级从真过渡到假时, 启用位被复位。
.DN (完成) 位 29	当协调指令被成功验证并排队后, 完成位被设置。由于它是在排队后被设置, 因此在它离开队列后校对操作时, 可能会因为运行时间错误而显示为被设置。当梯级从假过渡到真时, 完成位被复位。
.ER (错误) 位 28	当梯级从假过渡到真时, 错误位被复位。当协调运动未能成功发起时, 错误位被设置。当排队的指令遇到运行时间错误时, 错误位也可能与完成位一起被设置。
.IP(正在进行) 位 26	当协调运动被成功发起后, 正在进行位被设置。当没有后续运动而且协调运动到达新位置, 或者有后续运动而且协调运动到达终止类型的指定, 或者协调运动被其他合并类型为协调运动的 MCCM 或 MCLM 指令所取代, 或者被 MCS 或 MCSD 指令终止时, 正在进行位将被复位。
.AC(活动) 位 23	当您的协调运动指令已经排队时, 活动位可让您了解哪个指令在控制运动。当协调运动变为活跃时, 活动位被设置。当过程完成位被设置, 或者当指令被停止时, 活动位被复位。
.PC(过程完成) 位 27	当梯级从假过渡到真时, 过程完成位被复位。当没有后续运动而且协调运动到达新位置, 或者有后续运动而且协调运动到达指定的终止类型时, 过程完成位将被复位。
.ACCEL(加速) 位 01	当协调运动处于加速阶段时, 加速位被设置。当协调运动处于恒定速度或减速阶段, 或者协调运动结束时, 加速位被复位。
.DECEL(减速) 位 02	当协调运动处于减速阶段时, 减速位被设置。当协调运动处于恒定速度或加速阶段, 或者协调运动结束时, 减速位被复位。

运动类型

运动类型操作数确定了位置数组用来表示协调运动路径的方法，以及经由 / 中心 / 半径参数用来表示经由和中心圆位置的方法。选项如下：绝对值或增量。

- 绝对值 - 坐标系按定义的速度，使用由各自操作数指定的加速度和减速度，沿着圆形路径运动到指定的位置。

当轴被配置为旋转工作时，绝对值运动与直线轴的处理方式相同。当轴位置超过退绕参数时，将生成一个错误。

指定位置数组的符号由控制器解释为运动的方向。负的位置值将指使内插器向负方向运动旋转轴以获得所期望的绝对位置。正值则表示需要正向运动来达到目标位置。要向负方向移动到退绕位置，必须使用负的退绕位置值，因为 -0 和 0 都作为 0 处理。当位置大于退绕值时，将产生错误。轴可以穿过退绕位置，但永远不会以增量方式超过一个退绕值。

- 增量 - 坐标系沿圆形路径以指定的速度，使用由各自操作数确定的加速度和减速度，移动由位置数组所定义的距离。

指定的距离由内插器解释，可以为正或者为负。负的位置值将指使内插器向负方向运动旋转轴，而正值则表示需要正向运动来达到目标位置。

位置

位置操作数是一维数组，其维度至少等于坐标系内指定的轴个数。是由位置数组来定义新的绝对值或者增量位置。

圆形类型

圆形类型操作数指定了标记为经由 / 中心 / 半径的数组如何被解释。选项如下：经由、圆形、半径、中心增量。

- 经由表示经由/中心/半径的数组成员指定了起点和终点之间的经由点。
- 中心表示经由 / 中心 / 半径的数组成员包含圆心。
- 半径表示经由 / 中心 / 半径的数组成员包含半径。其他成员被忽略。半径仅在二维坐标系中有效。
- 中心增量表示经由/中心/半径的数组成员定义的位置始终以增量方式定义圆形的中心，而不管运动类型操作数如何。增量值的符号为从起点到中心进行测量。

二维弧形示例

下面的例子介绍在各种圆形类型中如何使用绝对值和增量运动类型。

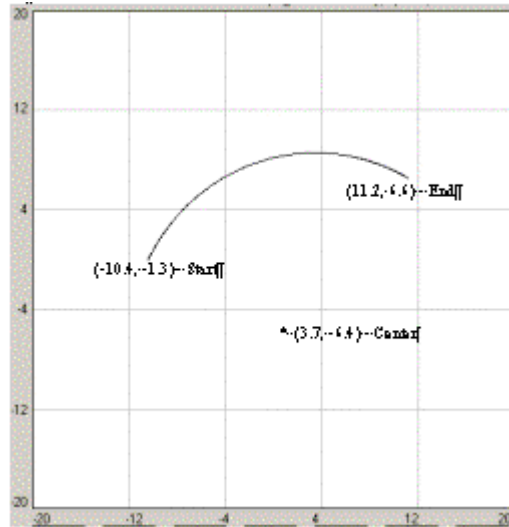
MCCM 使用中心圆形类型

下面的示例介绍使用 MCCM 指令圆形类型为中心以及绝对值（第一个例子）和增量（第二个例子）运动类型来达到相同的结果。基本假设如下：

- 两个轴 Axis0 和 Axis1，都属于坐标系 Coordinated_sys。
- Axis0 和 Axis1 彼此正交。
- Coordinated_sys 最初位于 (-10.4,-1.3) 单位。

Coordinated_sys 沿弧形运动到 (11.2,6.6) 单位，中心为 (3.7,-6.4) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图是上面的信息所生成的路径

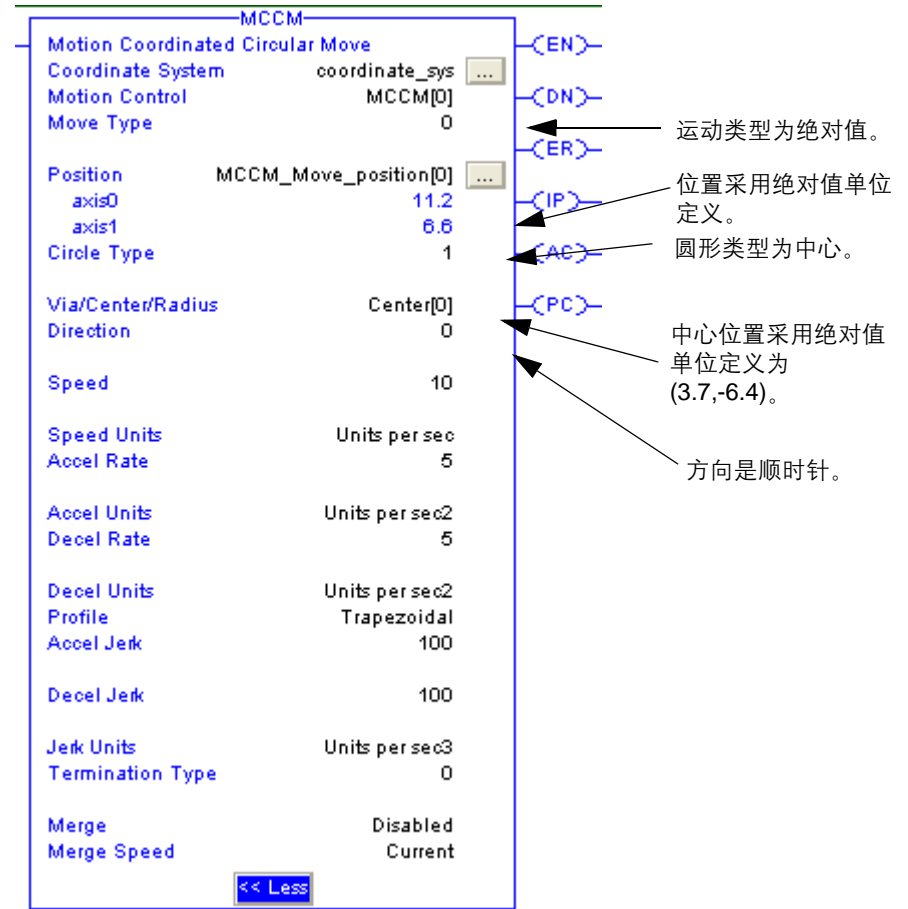
圆形类型为中心时的 MCCM 指令图



选定轴的矢量速度等于指定的速度，单位是单位每秒或者坐标系最大速度的百分比。同样，矢量加速度和减速度等于指定的加速度 / 减速度，单位采用单位每秒²或者坐标系最大加速度的百分比。

通过使用 MCCM 指令并选择顺时针方向、运动类型 = 绝对值或者运动类型 = 增量，即可实现这个路径。当选择圆形类型为中心时，经由 / 中心 / 半径位置定义的是弧形的圆心。

运动类型为绝对值时的 MCCM 梯形图指令



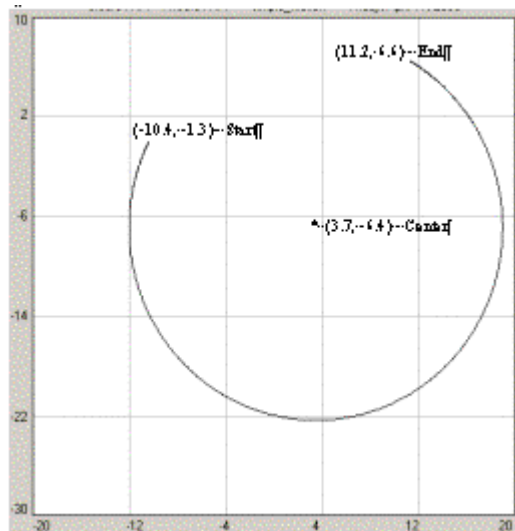
运动类型为增量时的 MCCM 梯形图指令

MCCM		
Motion Coordinated Circular Move		(EN)
Coordinate System	coordinate_sys ...	(DN)
Motion Control	MCCM[1]	(ER)
Move Type	1	(IP)
Position	MCCM_Move_position[2] ...	(AC)
axis0	21.6	(PC)
axis1	7.9	
Circle Type	1	
Via/Center/Radius	Center[1]	
Direction	0	
Speed	10	
Speed Units	Units per sec	
Accel Rate	5	
Accel Units	Units per sec ²	
Decel Rate	5	
Decel Units	Units per sec ²	
Profile	S-Curve	
Accel Jerk	100	
Decel Jerk	100	
JerK Units	Units per sec ³	
Termination Type	0	
Merge	Disabled	
Merge Speed	Current	

<< Less

运动类型为增量。
位置定义为从起点 (-10.4,-1.3) 开始的增量距离。
圆形类型为中心。
中心被定义为从起点 (-10.4,-1.3) 开始的增量距离 (14.1,-5.1)。
方向是顺时针。

如果选择逆时针方向 (方向 = 1)，则轴沿着下图所示的曲线运动。
方向为逆时针时的路径图



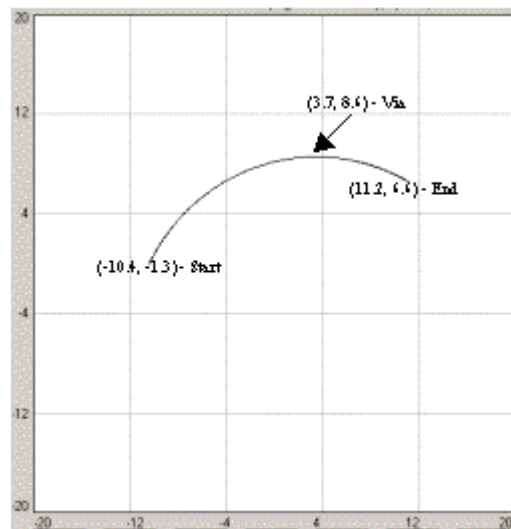
MCCM 指令使用经由圆形类型

下面的示例介绍使用 MCCM 指令圆形类型为经由以及绝对值（第一个例子）和增量（第二个例子）运动类型来达到相同的结果。基本假设如下：

- 两个轴 Axis0 和 Axis1，都属于坐标系 coordinate_sys。
- Axis0 和 Axis1 彼此正交。
- Coordinated_sys 最初位于 (-10.4,-1.3) 单位。

Coordinated_sys 沿弧形运动到 (11.2,6.6) 单位，通过点 (3.7,8.6) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图是上面的信息所生成的路径。

MCCM 指令采用操作数经由和绝对值时的路径图



选定轴的矢量速度等于指定的速度，单位是单位每秒或者坐标系最大速度的百分比。同样，矢量加速度和减速度等于指定的加速度 / 减速度，单位采用单位每秒² 或者坐标系最大加速度的百分比。

通过使用 MCCM 指令并选择顺时针方向、运动类型 = 绝对值或者运动类型 = 增量，即可实现这个路径。当选择圆形类型为经由时，经由 / 中心 / 半径位置定义的是弧形必须通过的点。

操作数值为经由和绝对值时的 MCCM 梯形图指令

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys ...
Motion Control	MCCM[2]
Move Type	0
Position	MCCM_Move_position[0] ...
axis0	11.2
axis1	6.6
Circle Type	0
Via/Center/Radius	Via[0]
Direction	0
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
Jerk Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current
<< Less	

运动类型为绝对值。

位置采用绝对值单位定义。

圆形类型为经由。

经由位置采用绝对值单位定义为 (3.7,8.6)。

方向是顺时针。

操作数值为经由和增量时的 MCCM 梯形图指令

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys ...
Motion Control	MCCM[2]
Move Type	0
Position	MCCM_Move_position[0] ...
axis0	11.2
axis1	6.6
Circle Type	0
Via/Center/Radius	Via[0]
Direction	0
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
Jerk Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current
<< Less	

运动类型为增量。

位置定义为从起点 (-10.4,-1.3) 开始的增量距离。

圆形类型为经由。

经由位置被定义为从起点 (-10.4,-1.3) 开始的增量距离 (14.1,9.9)。

方向是顺时针。

由于有三个点（轴的当前位置、指定的终点、指定的经由点），因此很难编程一个差的弧形。虽然仍可以编程一个弧形但并非想要的弧形，如果三个点共线（三个点都在一条直线上）或者不唯一（两个或多个点是同一个点），则会发生圆形编程错误的运行时间故障。此外，经由点暗示着弧形的方向，因此不需要再指定方向（将被忽略）。

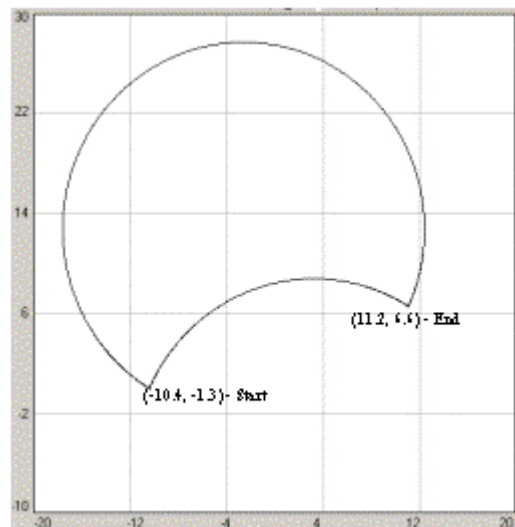
MCCM 指令使用半径圆形类型

下面的示例介绍使用 MCCM 指令圆形类型为半径以及绝对值（第一个例子）和增量（第二个例子）运动类型来达到相同的结果。基本假设如下：

- 两个轴 Axis0 和 Axis1，都属于坐标系 coordinate_sys。
- 坐标系的维度值匹配为 2。只有当为坐标系配置二维时，才能配置半径圆形类型。
- Axis0 和 Axis1 彼此正交。
- coordinate_sys 最初位于 (-10.4,-1.3) 单位。

Coordinated_sys 沿弧形运动到 (11.2,6.6) 单位，半径为 15 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图显示了通过上述信息使用半径值为 15 单位或 -15 单位时所生成的路径。

圆形类型为半径时的路径图



通过使用 MCCM 指令并选择顺时针方向、运动类型 = 绝对值或者运动类型 = 增量，即可实现这个路径。当选择圆形类型为半径时，经由 / 中心 / 半径位置定义的是弧形的半径。

MCCM 指令运动类型为绝对值；圆形类型为半径

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys <input type="button" value="..."/>
Motion Control	MCCM[0]
Move Type	0
Position	MCCM_Move_position[B] <input type="button" value="..."/>
axis0	11.2
axis1	6.6
Circle Type	2
Via/Center/Radius	Radius[2]
Direction	0
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	Trapezoidal
Accel Jerk	100
Decel Jerk	100
Jerk Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current
<input type="button" value=" << Less"/>	

- ← 运动类型为绝对值
- ← 位置采用绝对值单位定义。
- ← 圆形类型为半径
- ← 半径被定义为 15 单位，并且储存在半径 [2] 标签内。
- ← 方向是顺时针。

MCCM 指令运动类型为增量；圆形类型为半径

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys <input type="button" value="..."/>
Motion Control	MCCM[4]
Move Type	1
Position	MCCM_Move_position[0] <input type="button" value="..."/>
axis0	21.6
axis1	7.9
Circle Type	2
Via/Center/Radius	Radius[1]
Direction	0
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
JerK Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current

运动类型为增量

位置定义为从起点 (-10.4,-1.3) 开始的增量距离。

圆形类型为半径。

半径被定义为 15 单位，并且储存在半径 [1] 标签内。

方向是顺时针。

运动类型对半径值的指定没有影响。正的半径始终会产生更短 (<180°) 的弧形，负的半径产生更长 (>180°) 的弧形（参见路径图）。如果是 180°，则半径的符号不相干。圆形类型为半径仅在二维坐标系中是有效的。

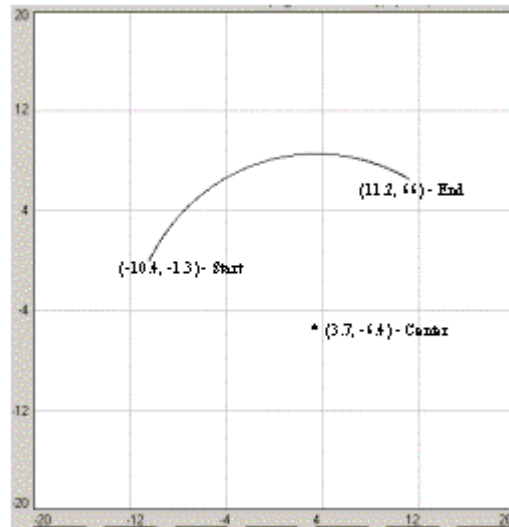
MCCM 使用中心增量圆形类型

下面的示例介绍使用 MCCM 指令圆形类型为中心增量以及绝对值（第一个例子）和增量（第二个例子）运动类型来达到相同的结果。基本假设如下：

- 两个轴 Axis0 和 Axis1，都属于坐标系 coordinate_sys。
- Axis0 和 Axis1 彼此正交。
- coordinate_sys 最初位于 (-10.4,-1.3) 单位。

coordinate_sys 沿弧形运动到 (11.2,6.6) 单位，中心为从起点开始的增量 (14.1,-5.1) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图是上面的信息所生成的路径。

圆形类型为中心增量时的路径图



通过使用 MCCM 指令并选择顺时针方向、运动类型 = 绝对值或者运动类型 = 增量，即可实现这个路径。当选择圆形类型为中心增量时，经由 / 中心 / 半径位置定义的是弧形的圆心。

MCCM 指令运动类型为绝对值；圆形类型为中心增量

MCCM		
Motion Coordinated Circular Move		
Coordinate System	coordinate_sys	...
Motion Control	MCCM[5]	
Move Type	0	← 运动类型为绝对值
Position	MCCM_Move_position[10]	...
axis0	11.2	← 位置采用绝对值单位定义。
axis1	6.6	← 圆形类型为中心增量。
Circle Type	3	
Via/Center/Radius	Via[2]	
Direction	0	← 中心被定义为从起点 (-10.4, -1.3) 开始的增量距离 (14.1, -5.1)。
Speed	10	← 方向是顺时针。
Speed Units	Units per sec	
Accel Rate	5	
Accel Units	Units per sec ²	
Decel Rate	5	
Decel Units	Units per sec ²	
Profile	S-Curve	
Accel Jerk	100	
Decel Jerk	100	
Jerk Units	Units per sec ³	
Termination Type	0	
Merge	Disabled	
Merge Speed	Current	
		<< Less

MCCM 指令采用运动类型为增量以及中心类型为中心增量时，与采用运动类型为增量以及圆形类型为中心时的 MCCM 指令是相同的。

二维整圆示例

创建一个完整的圆形是圆弧的一个特例。下图是二维整圆的一个例子。

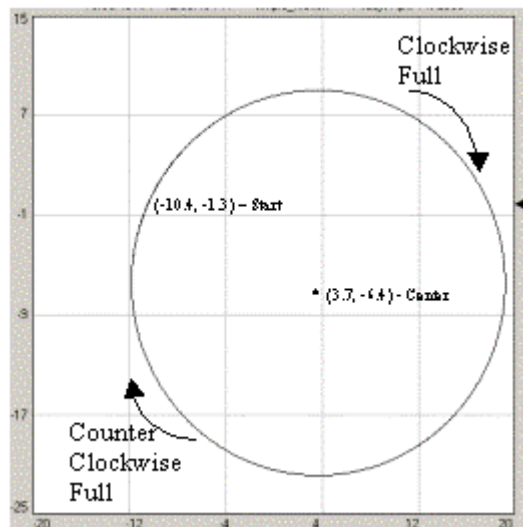
MCCM 整圆

下面的示例介绍使用 MCCM 指令圆形类型为中心以及绝对值（第一个例子）和增量（第二个例子）运动类型来创建一个完整的圆。基本假设如下：

- 两个轴 Axis0 和 Axis1，都属于坐标系 Coordinated_sys。
- Axis0 和 Axis1 彼此正交。
- Coordinated_sys 最初位于 (-10.4,-1.3) 单位。

Coordinated_sys 沿弧形运动到 (-10.4,-1.3) 单位，中心为距离起点的 (3.7,-6.4) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图是上面的信息所生成的圆形。

MCCM 指令整圆路径图



通过使用 MCCM 指令并选择顺时针方向、运动类型 = 绝对值或者运动类型 = 增量，即可实现这个路径。当选择圆形类型为中心时，经由 / 中心 / 半径位置定义的是弧形的圆心。

MCCM 指令运动类型为绝对值；圆形类型为中心

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys ...
Motion Control	MCCM[7]
Move Type	0
Position	MCCM_Move_position[14] ...
axis0	-10.4
axis1	-1.3
Circle Type	1
Via/Center/Radius	Center[8]
Direction	2
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
JerK Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current
<< Less	

运动类型为绝对值
 位置采用绝对值单位定义。
 圆形类型为中心。
 中心位置采用绝对值单位定义为 (3.7,-6.4)。
 方向是顺时针完整。

MCCM 采用运动类型为增量，中心类型为中心。

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys ...
Motion Control	MCCM[8]
Move Type	1
Position	MCCM_Move_position[16] ...
axis0	0.0
axis1	0.0
Circle Type	1
Via/Center/Radius	Center[4]
Direction	2
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
JerK Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current
<< Less	

运动类型为增量。
 圆形类型为中心。
 中心被定义为从起点 (-10.4,-1.3) 开始的增量距离 (14.1,-5.1)。
 方向是顺时针完整。

要使用半径作为圆形类型来绘制完整的圆形：

- 起点不能等于终点。
- 方向必须为顺时针完整或逆时针完整。
- 半径的符号不相干。

MCCM 旋转轴示例

下面的示例介绍使用 MCCM 指令并采用旋转轴以及运动类型分别为绝对值和增量。

使用三个轴、一个旋转轴以及绝对值运动类型的 MCCM 指令

第一个例子使用三个轴及一个旋转轴的坐标系和绝对值运动类型。路径图基于以下假设：

- 三轴坐标系名为 `coordinate_sys` (Axis2 即 Z 轴，在图中被忽略以减少混淆，更好地演示旋转轴 (Axis0) 的动作)。
- Axis0 为旋转轴，退绕为 5 转。
- 开始位置为 0, 0, 0。
- 结束位置为 5, 5, 5。
- 经由位置为 5, 3.5, 3.5

运动类型为绝对值时的 MCCM 梯形图指令

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys ...
Motion Control	MCCM[7]
Move Type	0
Position	MCCM_Move_position[0] ...
axis0	5.0
axis1	5.0
axis2	5.0
Circle Type	0
Via/Center/Radius	Via[1]
Direction	0
Speed	1
Speed Units	Units per sec
Accel Rate	100
Accel Units	% of Maximum
Decel Rate	100
Decel Units	% of Maximum
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
Jerk Units	Units per sec ³
Termination Type	3
Merge	Disabled
Merge Speed	Programmed
<< Less	

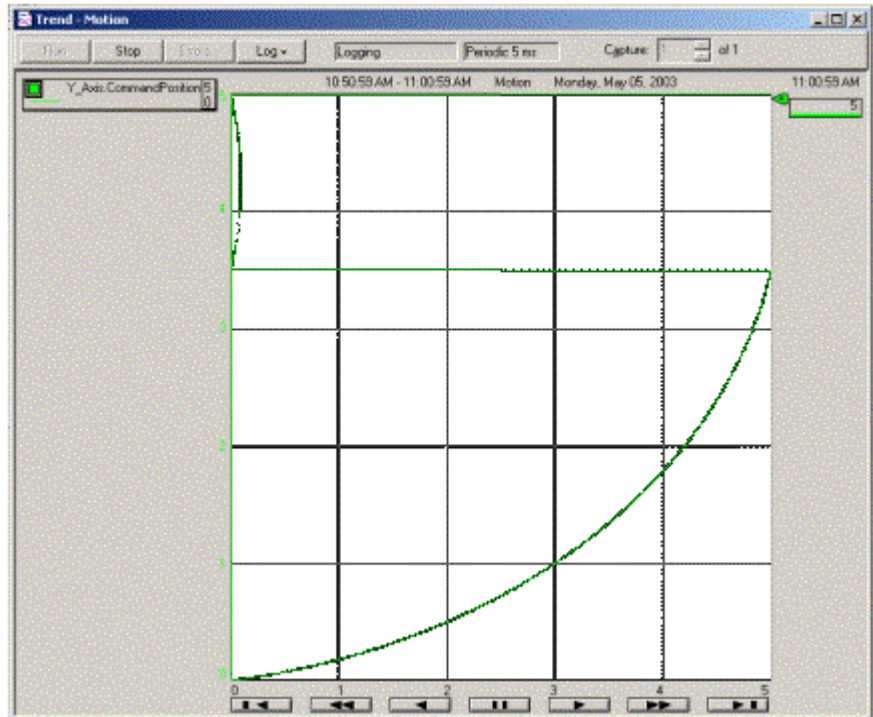
← 运动类型为绝对值。

← 圆形类型为经

← 方向为最短。

上面的 MCCM 指令产生下面的图形。

使用三个轴、一个旋转轴和绝对值运动类型的 MCCM 路径图



轴实际上沿着一个弧形按逆时针方向行进，从 (0,0,0) 到 (5,5,5) 并经由 (5,3.5,3.5) 位置。方向被指定为顺时针，但由于为圆形类型指定了经由，因此方向操作数被忽略。在生成一个 90 度的弧形时，运动停止。对于 Axis0 来说，即使其运动类型为绝对值，仍有一次行进通过退绕。应注意，协调运动的路径是在直线空间内确定的，但轴的位置则由旋转配置所限制的。终点和经由点要符合由 Axis0 的旋转退绕所定义的绝对值位置以内。但是，从这些选择中所产生的运动可能会穿过旋转轴的退绕。

使用两个旋转轴和增量运动类型的 MCCM 指令

这个例子使用两个旋转轴的坐标系和增量运动类型。路径图基于以下假设。

- 二轴坐标系，名为 coordinate_sys。
- Axis0 为旋转轴，退绕为 1 转。
- Axis1 为旋转轴，退绕为 2 转。
- 开始位置为 0,0。
- 到结束位置的增量为 0.5, -0.5。
- 到中心位置的增量为 0.5, 0。

运动类型为绝对值时的 MCCM 梯形图指令

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys <input type="button" value="..."/>
Motion Control	MCCM[8]
Move Type	1
Position	MCCM_Move_position[16] <input type="button" value="..."/>
axis0	0.0
axis1	0.0
Circle Type	1
Via/Center/Radius	Center[4]
Direction	0
Speed	1
Speed Units	Units per sec
Accel Rate	100
Accel Units	% of Maximum
Decel Rate	100
Decel Units	% of Maximum
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
JerK Units	Units per sec ³
Termination Type	1
Merge	Disabled
Merge Speed	Programmed
<input type="button" value=" << Less"/>	

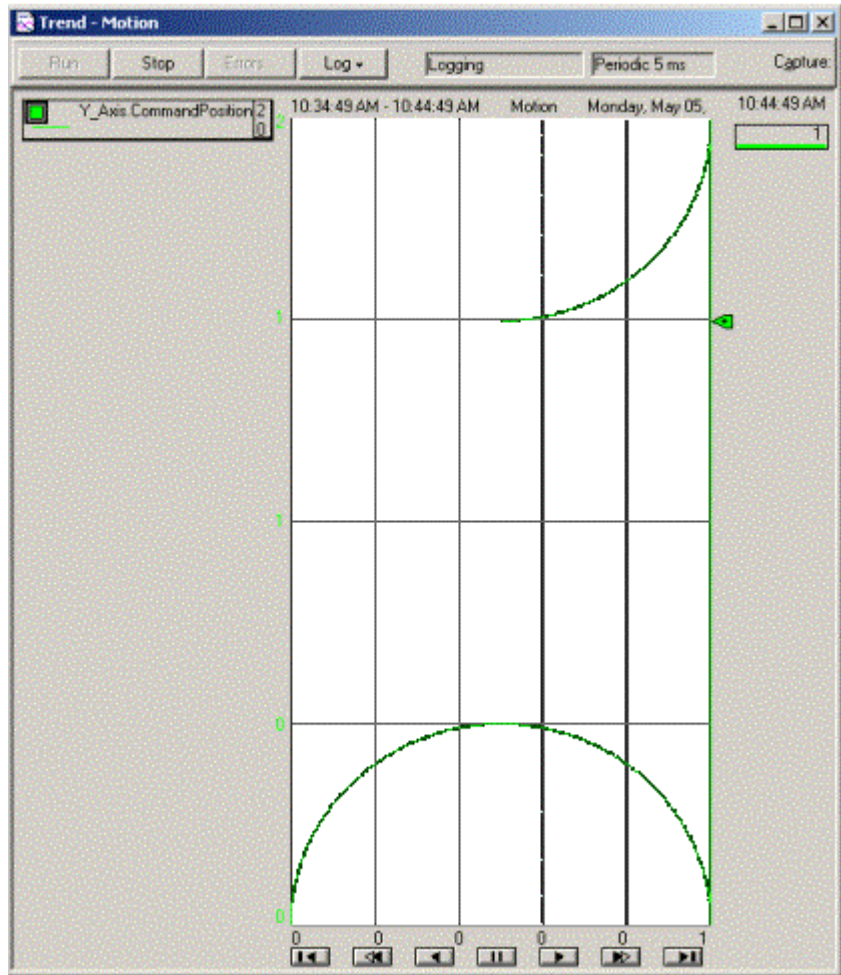
运动类型为增量。

圆形类型为中心。

方向是顺时针。

上面的 MCCM 指令产生下面的图形。

使用两个旋转轴和增量运动类型的 MCCM 路径图



轴按照顺时针方向在一个圆内行进，从 (0,0) 到 (0.5,1.5)。在生成一个 270 度的弧形时，运动停止。对于 Axis1 来说，有一次行进穿过退绕。应注意，协调运动的路径是在直线空间内确定的，但轴的位置则由旋转配置所限制的。对于圆形计算来说，终点是 (0.5,-0.5)，但运动的实际终点是 (0.5,1.5)。指令被指定后，即使一个轴采用负的增量目标位置，我们得到的仍然是顺时针方向的运动。终点不需要符合由轴的旋转退绕所定义的绝对值位置以内。

三维弧形

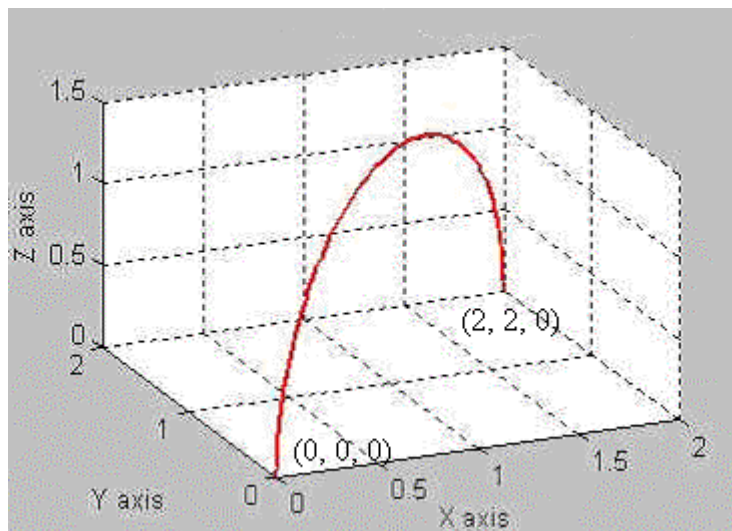
对于有三个主轴与其关联的坐标系，可以创建一个三维弧形。

使用 MCCM 以及圆形类型为经由的三维弧形

下面的示例介绍使用 MCCM 以及圆形类型为经由以及运动类型为绝对值来创建一个三维弧形。基本假设如下：

- 三个轴 Axis0、Axis1 和 Axis2，都属于坐标系 coordinate_sys。
- coordinate_sys 是一个三维坐标系。
- Axis0、Axis1 和 Axis2 彼此正交。
- coordinate_sys 最初位于 (0.0, 0.0, 0.0) 单位。

Coordinated_sys1 沿弧形运动到 (2.0, 2.0, 0.0) 单位，通过 (1.0, 1.0, 1.414) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图是上面的信息所生成的三维弧形。
使用圆形类型为经由的三维弧形



通过使用 MCCM 指令并选择运动类型为绝对值、圆形类型为经由，即可实现这个路径。当选择经由时，经由 / 中心 / 半径位置定义的是弧形必须通过的点。

圆形类型为经由时三维弧形的 MCCM 梯形图指令

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys <input type="button" value="..."/>
Motion Control	MCCM[B]
Move Type	0
Position	MCCM_Move_position[16] <input type="button" value="..."/>
axis0	2.0
axis1	2.0
axis2	0.0
Circle Type	0
Via/Center/Radius	Via[4]
Direction	0
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
Jerk Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Programmed

三维坐标系。

位置采用绝对值单位定义。

圆形类型为经由。

经由位置定义为绝对值单位 (1.0, 1.0, 1.414)。

对于经由圆形类型，方向被忽略。

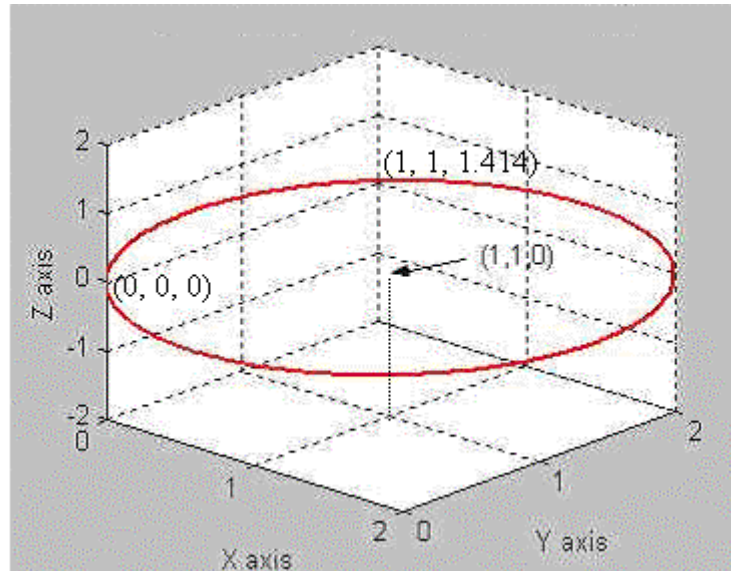
使用 MCCM 以及圆形类型为中心的三维弧形

下面的示例介绍使用 MCCM 以及圆形类型为中心以及运动类型为绝对值来创建一个三维弧形。基本假设如下：

- 三个轴 Axis0、Axis1 和 Axis2，都属于坐标系 coordinate_sys。
- coordinate_sys 是一个三维坐标系。
- Axis0、Axis1 和 Axis2 彼此正交。
- coordinate_sys 最初设定于 (0.0, 0.0, 0.0) 单位。

Coordinated_sys1 沿弧形运动到 (1.0, 1.0, 1.414 单位，中心位于 (1.0, 1.0, 1.0) 单位，矢量速度为 10.0 单位每秒，加速度和减速度值为 5.0 单位每秒²。下图是上面的信息所生成的三维弧形。

使用最短完整作为方向操作数时的三维路径



通过使用 MCCM 指令并选择运动类型为绝对值、圆形类型为中心，即可实现这个路径。当选择经由时，经由 / 中心 / 半径位置定义的是弧形必须通过的点。

圆形类型为中心时三维弧形的 MCCM 梯形图指令

MCCM	
Motion Coordinated Circular Move	
Coordinate System	coordinate_sys ...
Motion Control	MCCM[8]
Move Type	0
Position	MCCM_Move_position[0] ...
axis0	1.0
axis1	1.0
axis2	1.414
Circle Type	1
Via/Center/Radius	Center[8]
Direction	2
Speed	10
Speed Units	Units per sec
Accel Rate	5
Accel Units	Units per sec ²
Decel Rate	5
Decel Units	Units per sec ²
Profile	S-Curve
Accel Jerk	100
Decel Jerk	100
JerK Units	Units per sec ³
Termination Type	0
Merge	Disabled
Merge Speed	Current

<< Less

三维坐标系。

位置采用绝对值单位定义。

圆形类型为中心。

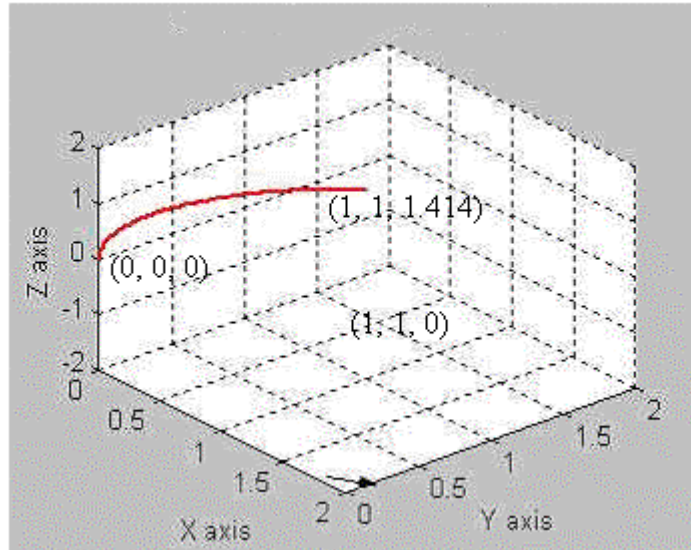
中心位置定义为绝对值单位 (1.0, 1.0, 0.0)。

方向是最短完整。

对于完整圆形，将为之操作数设置为除了起点之外的任何点，并使用其中一种完整方向类型。终点被认为是起点。这是因为在三维空间里，您需要三个点来为一个圆指定平面。

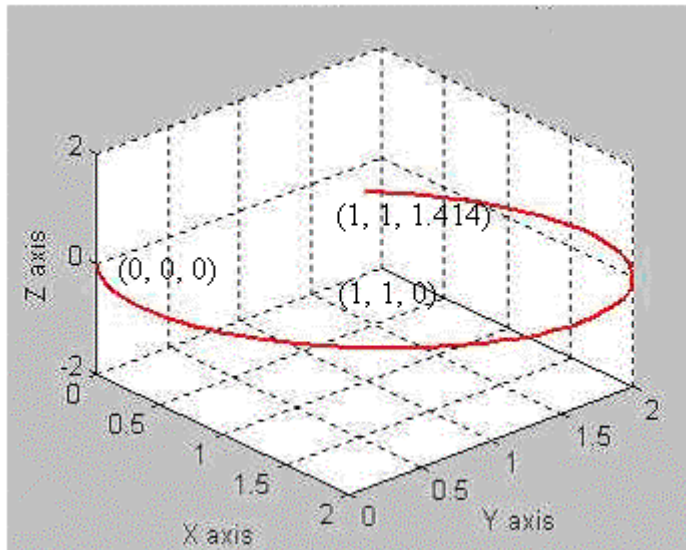
通过在上面的 MCCM 指令中将方向操作数更改为最短，即可生成以下路径。方向操作数的最短选项将从起点取最短路径到达由 MCCM 指令的位置操作数所定义的点。

使用最短作为方向操作数时的三维路径



在上面的 MCCM 指令中将方向操作数更改为最长，则所沿的路径为从起点开始到达由 MCCM 指令的位置操作数所定义的点之间最长的路径。参见下图查看一个最长路径的示例。

使用最长作为方向操作数时的三维路径



经由 / 中心 / 半径

根据所选的运动类型和圆形类型，经由 / 中心 / 半径位置参数定义了沿着圆形的位置、圆形的中心或者圆形半径的绝对值或增量值，如下表中定义。如果圆形类型为经由或中心，则经由 / 中心 / 半径位置参数是一维数组，其维度定义为至少等于坐标系内指定的轴个数。如果圆形类型是半径，则经由 / 中心 / 半径位置参数是单个值。

运动类型	圆形类型	行为
绝对值	经由	经由 / 中心 / 半径位置数组定义了一个沿着圆形的位罝。对于非完整圆形的情况，位置参数数组定义了弧形的终点。对于完整圆形的情况，位置参数数组定义了沿着圆形除了终点之外的任意第二个点。
增量	经由	经由 / 中心 / 半径位置数组与旧位置之和定义了沿着圆形的位罝。对于非完整圆形的情况，位置参数数组与旧位置之和定义了弧形的终点。对于完整圆形的情况，位置参数数组与旧位置之和定义了沿着圆形除了终点之外的任意第二个点。
绝对值	中心	经由 / 中心 / 半径位置数组定义了圆形的中心。对于非完整圆形的情况，位置参数数组定义了弧形的终点。对于完整圆形的情况，位置参数数组定义了沿着圆形除了终点之外的任意第二个点。

运动类型	圆形类型	行为
增量	中心	经由 / 中心 / 半径位置数组与旧位置之和定义了圆形的中心。对于非完整圆形的情况，位置参数数组与旧位置之和定义了弧形的终点。对于完整圆形的情况，位置参数数组与旧位置之和定义了沿着圆形除了终点之外的任意第二个点。
绝对值或增量	半径	经由 / 中心 / 半径位置单值定义了弧形的半径。这个值的符号用于确定中心点以区分可能出现的两个弧形。正值表示生成小于 180 弧形的中心点。负值表示生成大于 180 弧形的中心点。这种圆形类型仅对于二维圆形是有效的。位置参数数组按照运动类型来定义弧形的终点。
绝对值	中心增量	经由 / 中心 / 半径位置数组与旧位置之和定义了圆形的中心位置。对于非完整圆形的情况，位置参数数组定义了弧形的终点。对于完整圆形的情况，位置参数数组定义了沿着圆形除了终点之外的任意第二个点。
增量	中心增量	经由 / 中心 / 半径位置数组与旧位置之和定义了圆形的中心位置。对于非完整圆形的情况，位置参数数组与旧位置之和定义了弧形的终点。对于完整圆形的情况，位置参数数组与旧位置之和定义了沿着圆形除了终点之外的任意第二个点。

方向

方向操作数按照右手螺旋法则定义了二维圆形运动的旋转方向为顺时针或逆时针。对于三维圆形运动，方向为最短或最长。在二维和三维中，它都可以指示圆形运动是否要成为完整圆形。

速度

速度操作数定义了协调运动路径上的最大矢量速度。

速度单位

速度单位操作数定义了应用到速度操作数上的单位，可以直接采用协调单位，或者作为坐标系中定义的最大值的百分比。

加速度

加速度操作数定义了协调运动路径上的最大加速度。

加速度单位

加速度单位操作数定义了应用到加速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

减速度

减速度操作数定义了协调运动路径上的最大减速度。

减速度单位

减速度单位操作数定义了应用到减速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

模式

模式操作数确定了协调运动是使用梯形还是 S 曲线速度模式。请参阅 [页码 280](#) 上 MCLM 指令的模式部分，详细了解关于梯形和 S 曲线模式的信息。

加速跃度

加速跃度定义了编程运动的最大加速跃度。有关计算加速跃度的更多信息，请参阅本章的“跃度单位”部分。

减速跃度

减速跃度定义了编程运动的最大减速跃度。有关计算减速跃度的更多信息，请参阅本章的“跃度单位”部分。

跃度单位

跃度单位定义了向加速跃度和减速跃度操作数所输入的值上应用的单位。所输入的值直接以指定坐标系的位置单位为单位，或者为百分比。当配置使用最大值 % 时，跃度被应用为坐标系属性中指定的最大加速跃度和最大减速跃度操作数的百分比。当配置为使用时间 % 时，该值为基于指令中指定的速度、加速度和减速度的一个百分比。

如果您要将工程单位换算成 % 时间，请使用以下公式。

对于加速跃度：

$$j_a [\text{EU/s}^3] = \frac{a_{\text{max}}^2 [\text{EU/s}^2]}{v_{\text{max}} [\text{EU/s}]} \left(\frac{200}{j_a [\% \text{ of time}]} - 1 \right)$$

对于减速跃度：

$$j_d [\text{EU/s}^3] = \frac{a_{\text{max}}^2 [\text{EU/s}^2]}{v_{\text{max}} [\text{EU/s}]} \left(\frac{200}{j_d [\% \text{ of time}]} - 1 \right)$$

对于加速跃度：

$$j_a [\% \text{ of time}] = \frac{2}{1 + \frac{j_a [\text{EU/s}^3] v_{\text{max}} [\text{EU/s}]}{a_{\text{max}}^2 [\text{EU/s}^2]}} 100$$

对于减速跃度：

$$j_d [\% \text{ of time}] = \frac{2}{1 + \frac{j_d [\text{EU/s}^3] v_{\text{max}} [\text{EU/s}]}{a_{\text{max}}^2 [\text{EU/s}^2]}} 100$$

终止类型

有关选择终止类型的信息，请参阅本手册中的[页码 257](#)。

合并

合并定义了是否要将所有指定轴的运动变成单纯的协调运动。选项如下：禁止合并、协调运动或全部运动。

- 禁止合并 - 任何当前正在执行的单轴运动指令如果涉及到指定坐标系中定义的任何轴，则不受此指令激活的影响，并且在受到影响的轴上产生叠加运动。如果在同一个坐标系中或者在包含与当前活动坐标系共用的任何轴的其他坐标系中发起了第二条指令，则会标记一个错误。
- 协调运动 - 任何涉及到指定坐标系的当前正在执行的协调运动指令都将被终止，活跃的运动被混合到当前运动中，并按照合并速度参数中定义的速度。指定坐标系中任何挂起的协调运动指令都被取消。任何当前正在执行的系统单轴运动指令如果涉及到指定坐标系中定义的任何轴，则不受此指令激活的影响，并且在受到影响的轴上产生叠加运动。
- 全部运动 - 任何涉及到指定坐标系中定义的任何轴的当前正在执行的单轴运动指令，以及任何当前正在执行的协调运动指令，都将被终止。前面的运动混合进入当前的运动，并且按照合并速度参数中定义的速度。任何挂起的协调运动指令都被取消。

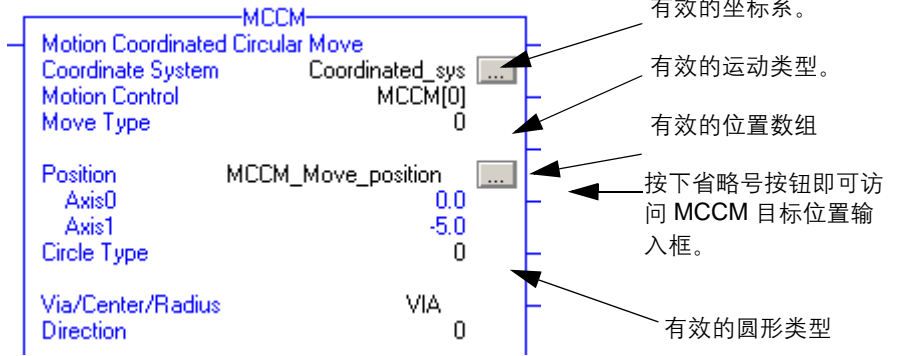
合并速度

合并速度操作数定义了当合并被使能时，在协调运动的路径上是当前速度还是将编程速度用作最大速度。当前速度是当前坐标系中定义的所有轴的全部运动（例如点动、MAM 和齿轮传动运动）的矢量和。

MCCM 目标位置输入对话框

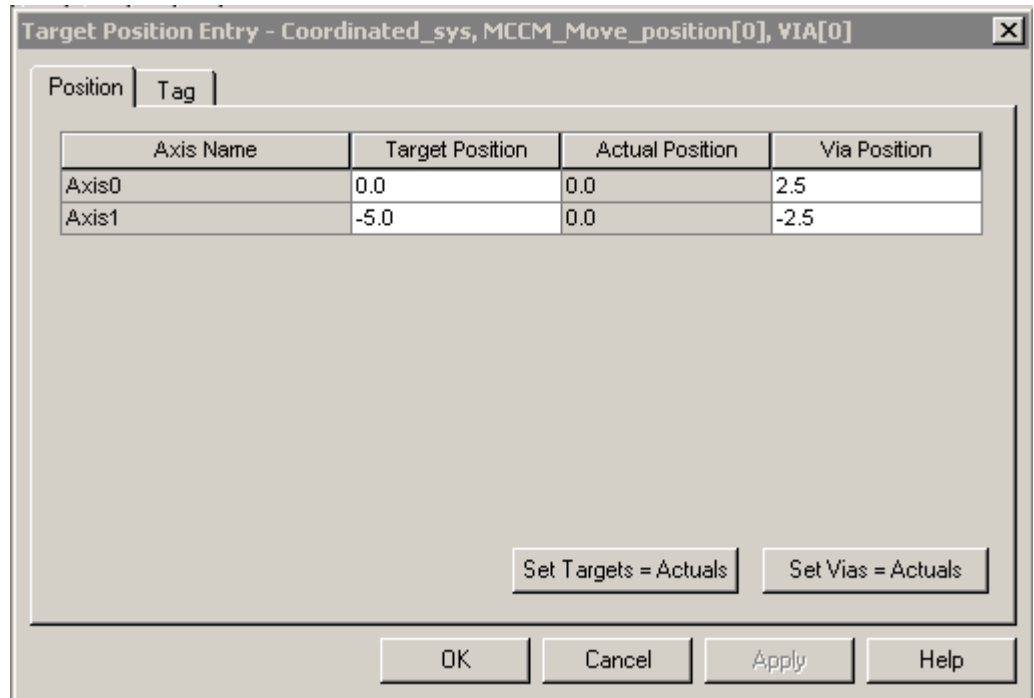
MCCM 目标位置输入对话框的访问方法是，单击梯形图指令面板位置操作数右边的省略号按钮。只有当指令的坐标系已经命名、对位置操作数提供的有效标签名称包含足够的元素来容纳轴的个数，而且选择了有效的运动类型和有效地圆形类型时，才可以访问目标位置输入对话框。如果这些标准没有满足，则状态条上会显示错误消息

用于访问目标位置输入框的 MCCM 梯形图有效值



按省略号按钮，则显示下面的对话框。

MCCM 指令目标位置输入对话框 - 位置选项卡



目标位置输入对话框字段

功能	描述
轴名称 (Axis Name)	这个列内为在梯形面板中命名的坐标系中各个轴的名称。这些名称不可编辑。
目标位置 / 目标增量 (Target Position/Target Increment)	这个列内的值都是数字。它们显示运动的终点或者增量距离，取决于有效的运动类型。列的标题表示显示的是哪个运动类型。
实际位置 (Actual Position)	这个列包含坐标系中的轴的当前实际位置。当联机和坐标系自动标签更新被使能后，这些位置将动态更新。
经由位置 / 经由增量中 心位置 / 中心增量半径	根据所选的圆形类型，这个列包含经由点位置或增量、中心位置或增量。
设置目标 = 实际值 (Set Targets = Actuals)	当运动类型为绝对值时这个按钮被使能，用于将实际位置字段内的值复制到目标位置字段中。
设置经由 = 实际值 (Set Vias = Actuals)	只有当运动类型为绝对值时，这个按钮才会使能。它用于将实际位置字段中的值复制到经由字段中。

所选择的运动类型和圆形类型决定着这个对话框的外观。下表演示所选择的运动类型和圆形类型组合如何影响到这个屏幕。

目标位置输入对话框变更

运动类型	圆形类型	行为
绝对值	经由	目标列的标题为目标位置。 经由列的标题为经由位置。 设置目标 = 实际值 (Set Targets = Actuals) 按钮被激活。 设置经由 = 实际值 (Set Vias = Actuals) 按钮被激活。
增量	经由	目标列的标题为目标增量。 经由列的标题为经由增量。 设置目标 = 实际值 (Set Targets = Actuals) 按钮未激活 (灰掉)。 设置经由 = 实际值 (Set Vias = Actuals) 按钮未激活 (灰掉)。
绝对值	中心	目标列的标题为目标位置。 中心列的标题为中心位置。 设置目标 = 实际值 (Set Targets = Actuals) 按钮被激活。 设置经由 = 实际值 (Set Vias = Actuals) 按钮被激活。
增量	中心	目标列的标题为目标增量。 中心列的标题为中心增量。 设置目标 = 实际值 (Set Targets = Actuals) 按钮未激活 (灰掉)。 设置经由 = 实际值 (Set Vias = Actuals) 按钮未激活 (灰掉)。
绝对值	半径	目标列的标题为目标位置。 半径列标题为半径。 设置目标 = 实际值 (Set Targets = Actuals) 按钮被激活。 设置经由 = 实际值 (Set Vias = Actuals) 按钮未激活 (灰掉)。

目标位置输入对话框变更

运动类型	圆形类型	行为
增量	半径	<p>目标列的标题为目标增量。</p> <p>半径列标题为半径。</p> <p>设置目标 = 实际值 (Set Targets = Actuals) 按钮未激活 (灰掉)。</p> <p>设置经由 = 实际值 (Set Vias = Actuals) 按钮未激活 (灰掉)。</p>
绝对值	中心增量	<p>目标列的标题为目标位置。</p> <p>中心增量列的标题为中心增量。</p> <p>设置目标 = 实际值 (Set Targets = Actuals) 按钮被激活。</p> <p>设置经由 = 实际值 (Set Vias = Actuals) 按钮未激活 (灰掉)。</p>
增量	中心增量	<p>目标列的标题为目标增量。</p> <p>中心增量列的标题为中心增量。</p> <p>设置目标 = 实际值 (Set Targets = Actuals) 按钮未激活 (灰掉)。</p> <p>设置经由 = 实际值 (Set Vias = Actuals) 按钮未激活 (灰掉)。</p>

MCCM 是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志：

不受影响

故障条件：

无

错误代码：

参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码：

扩展错误代码可帮助进一步定义为这条特殊指令所提供的错误消息。其行为取决于它们关联什么错误代码。

伺服关闭状态 (5)、关闭状态 (7)、轴类型非伺服 (8)、轴未配置 (11)、归零正在进行错误 (16) 以及非法轴数据类型 (38) 错误的扩展错误代码功能都是非常类似的。对于扩展错误代码将显示 0...n 的数字。这个数字是坐标系的索引，表示正处于错误状况中的轴。

对于错误代码轴未配置 (11)，还会有一个附加的值 -1，表示坐标系无法为协调运动设置该轴。

对于 MCCM 指令，错误代码 13 - 参数超出范围，扩展错误将返回一个数字，表示有问题的参数编号，这些参数在面板上以数字顺序从上到下（以零开始）列出。例如，2 表示运动类型的参数值有错误。

错误代码和（编号）	扩展错误数字指示符	指令参数	描述
参数超出范围 (13)	0	坐标系	主轴的个数不等于 2 或 3。
参数超出范围 (13)	2	运动类型	运动类型为小于 0 或者大于 1。
参数超出范围 (13)	3	位置	位置数组的长度不足以为坐标系中的所有轴提供位置。
参数超出范围 (13)	4	圆形类型	圆形类型为小于 0 或者大于 4。
参数超出范围 (13)	5	经由 / 中心 / 半径	经由 / 中心 / 半径数组的长度不足以为定义经由 / 中心点中的所有轴提供位置。
参数超出范围 (13)	6	方向	方向为小于 0 或者大于 3。
参数超出范围 (13)	7	速度	速度小于 0。
参数超出范围 (13)	9	加速度	加速度小于或等于 0。
参数超出范围 (13)	11	减速度	减速度小于或等于 0。
参数超出范围 (13)	14	终止类型	终止类型为小于 0 或者大于 3。

对于错误代码 54 - 最大减速度值为零，如果扩展错误代码返回一个正数 (0-n)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项

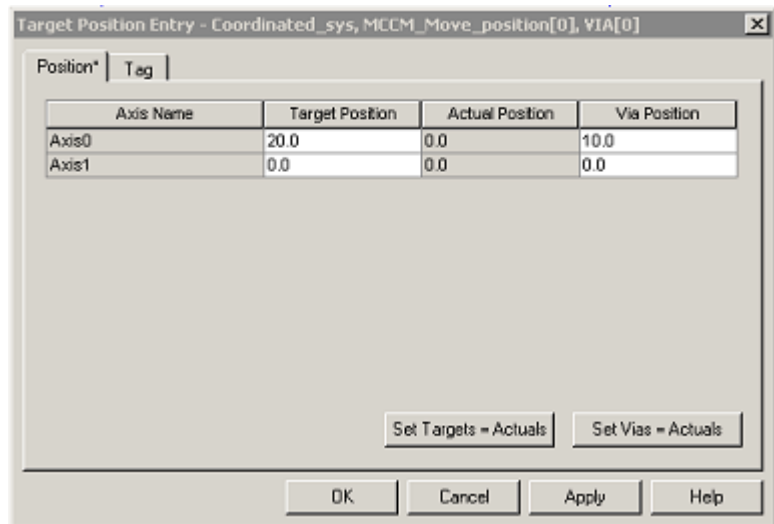
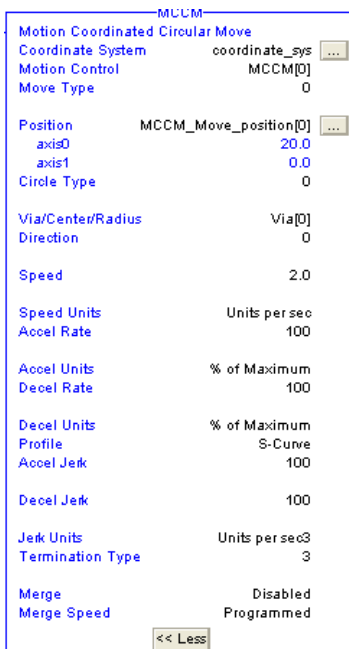
卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

圆形错误示例

由于 MCCM 指令的复杂性以及它可以产生的错误代码，提供下面简单示例来帮助您了解 MCCM 指令。

CIRCULAR_COLLINEARITY_ERROR (44) 示例

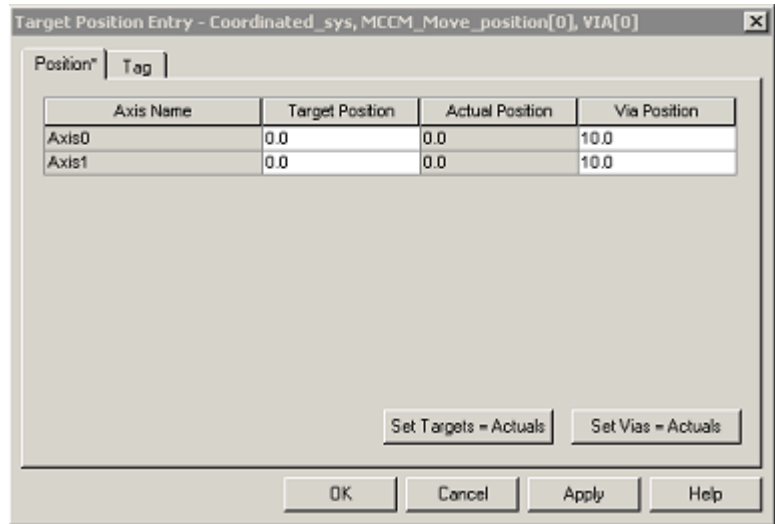
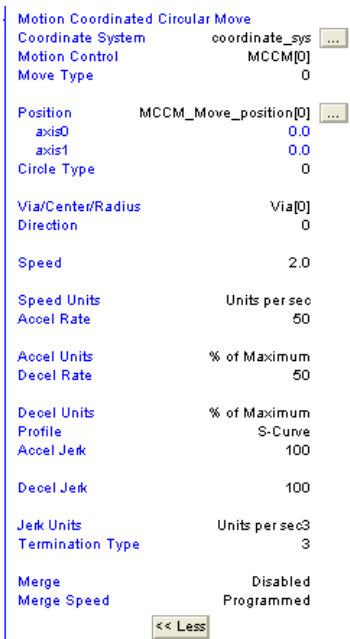
下面用于错误 #44 的例子将介绍一种起点、经由点和终点都位于一条直线上的情况。程序尝试从 0,0(当前位置)到 20,0 经由位置 10,0 生成一条二维弧形。由于这些点位于同一条直线上，因此无法计算出圆的中心点。如果程序用于一个三维中心类型的圆形而使用的起点、中心点和终点位于同一条直线上，则会产生这个错误。在这里，在无限多个平面内的编程点可有无限多个圆。梯形图程序和目标输入屏幕生成错误 #44。



CIRCULAR_START_END_ERROR (45) 示例

下面用于错误 #45 的例子将介绍一种起点和经由点是同一个点的情况。程序尝试从 0,0(当前位置) 回到 0,0 并通过位置 10,10 来生成一个二维的整圆。由于起点和经由点是同一个点，因此无法找到圆的中心点。

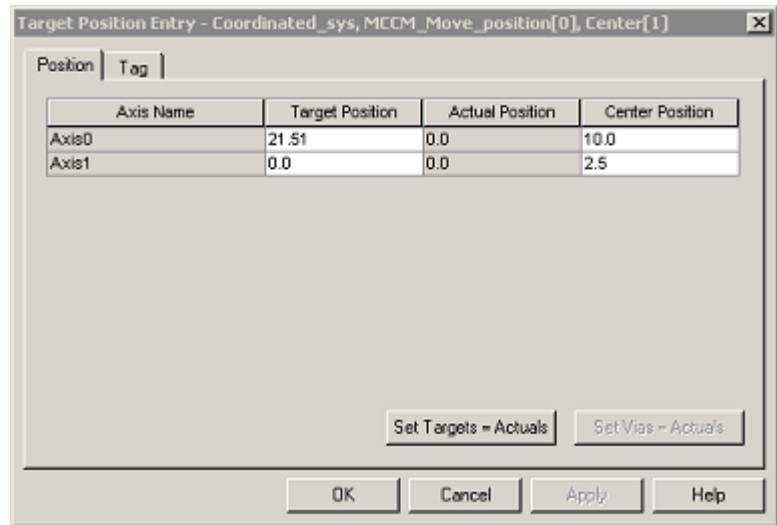
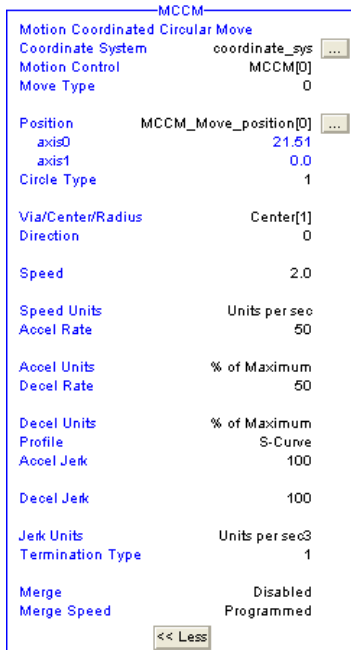
梯形图程序和目标输入屏幕生成错误 #45



CIRCULAR_R1_R2_MISMATCH_ERROR (46) 示例

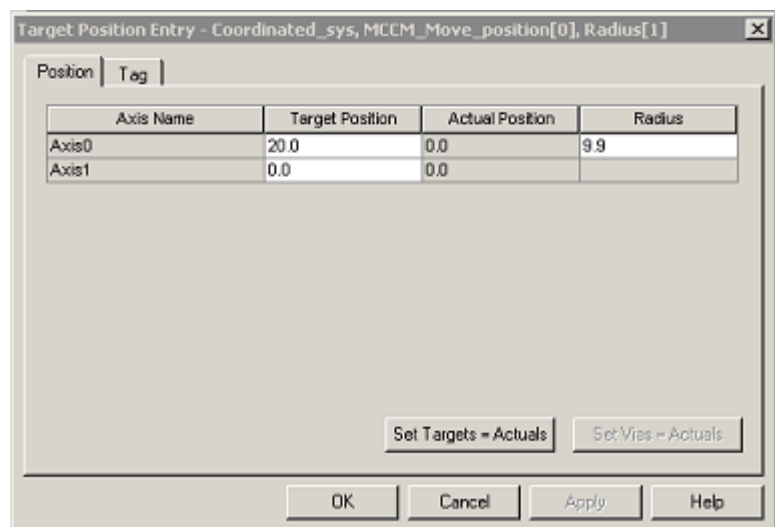
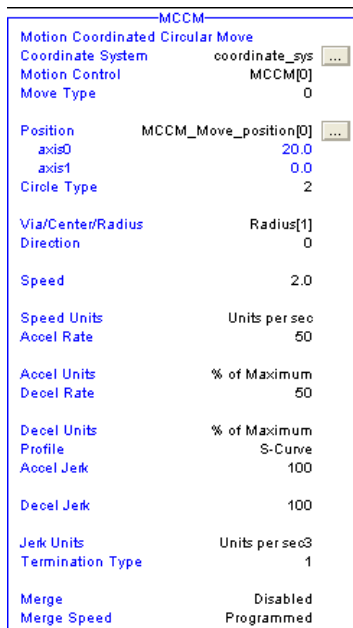
下面用于错误 #45 的例子介绍的情况是，半径的开始 / 结束长度差超过半径开始长度的 15%。程序尝试从 0,0(当前位置) 到 21.51,0 并使用中心点 10,10 生成一条二维弧形。半径的开始 / 结束长度差 (21.51 - 10 = 1.51) 超过半径开始长度 (.15 * 10 = 1.5) 的 15%。如果终点为 21.5，则这个例子让能够工作，中心点将重新计算以准确地落在起点和终点的中间。

梯形图程序和目标输入屏幕生成错误 #46



CIRCULAR_SMALL_R_ERROR (49) 示例

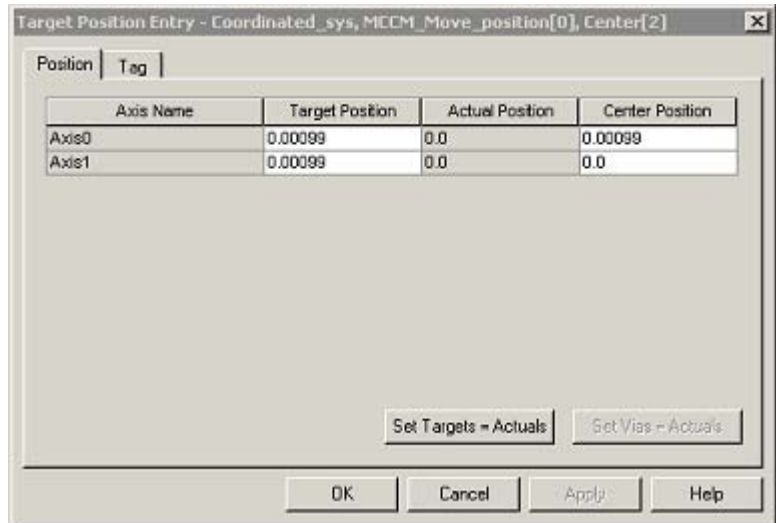
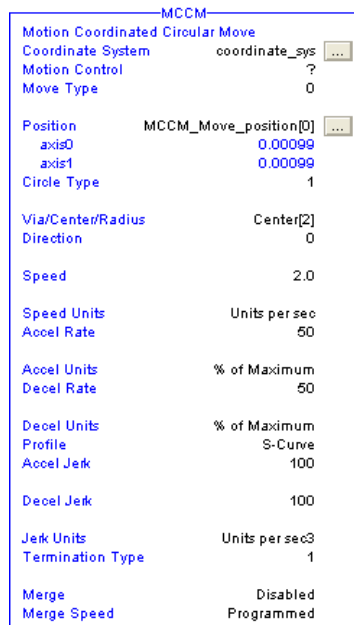
第一个关于错误 #49 的例子介绍的情况是，半径类型的圆使用的半径太短，无法跨过起点到终点之间的距离。程序尝试从 0,0(当前位置) 到 20,0 生成一条二维弧形。但是，用户尝试编程的半径类型的圆所使用的半径太短，无法跨过起点到终点之间的距离。
梯形图程序和目标输入屏幕生成错误 #49



CIRCULAR_SMALL_R_ERROR (49) 示例

第二个关于错误 #49 的例子介绍的情况是，半径类型的圆使用的半径幅值小于 0.001。程序尝试从 0,0(当前位置) 到 0.00099,0.00099 生成一个二维弧形。这个错误发生的原因是，用户尝试编程的半径类型的圆所使用的半径幅值小于 0.001 单位。

梯形图程序和目标输入屏幕生成错误 #49



MCCM 对状态位的更改：

状态位提供一种监视运动控制指令进度的途径。有三种类型的状态位提供有关信息。

- 轴
- 坐标系
- 协调运动

当 MCCM 指令发起后，状态位将经历以下变化。

轴状态位

位名称	含义
CoordinatedMotionStatus	当 MCCM 指令执行时设置，当指令完成时清除。

坐标系状态位

位名称	含义
MotionStatus	当 MCCM 指令正在工作而且坐标系连接到其关联轴时设置。

协调运动状态位

位名称	含义
AccelStatus	当矢量正在加速时设置。当混合正在进行或者矢量运动正在恒速或减速时清除。
DecelStatus	当矢量正在减速时设置。当混合正在进行或者矢量运动正在加速或者运动完成时清除。
ActualPosToleranceStatus	仅为实际容差终止类型设置。当满足以下两个条件时这个位被设置。1) 内插已经完成。2) 到编程终点的实际距离小于所配置的坐标系实际容差值。当指令完成后，这个位保持设置状态。当新的指令开始后，这个位被复位。
CommandPosToleranceStatus	每当到编程终点的实际距离小于所配置的坐标系命令容差值时，对于所有终止类型这个位都会被设置，并将在指令完成后保持设置状态。当新的指令开始后，这个位被复位。
StoppingStatus	当 MCCM 指令执行时，停止状态位被清除。
MoveStatus	当 MCCM 开始轴运动时设置。当最后一个运动控制指令的 .PC 位设置时或者某个运动控制指令执行导致停止时清除。
MoveTransitionStatus	当满足无减速或命令容差终止类型时设置。当混合共线运动时这个位不被设置，因为机器始终在路径上。当混合完成时、某个挂起指令的运动开始时或者某个运动控制指令的执行导致停止时被清除。表示不在路径上。

协调运动状态位

位名称	含义
MovePendingStatus	当一个挂起协调运动指令处于指令队列时设置。当指令队列为空时清除。
MovePendingQueueFullStatus	当指令队列满时设置。当队列中有空容纳新的协调运动指令时清除。

目前，协调运动仅支持一个协调运动指令的排队。因此，MovePendingStatus 位和 MovePendingQueueFullStatus 位始终是相同的。

圆形编程参考指南

圆形类型	用于二维 / 三维 / 两者	验证错误	方向 - 二维	方向 - 三维	注释
半径	2D	错误 25；无效指令 错误 45；终点 = 起点 错误 49；R 太小 ($ R < .001$) 或者 R 的长度不能跨过编程的点。	‘+’ 垂直于圆形平面观看为顺时针 / 逆时针	不适用	‘+’ 的半径强制弧长为 $\leq 180^\circ$ (最短弧形)。 “-” 的半径强制弧长为 $\geq 180^\circ$ (最长弧形)。 可以编程完整圆形。 对于完整圆形：将位置设置为除了起点以外圆上的任意点，并使用完整方向类型之一。
中心点	两者	错误 44；共线性 (仅三维) 错误 45；终点 = 起点 (仅三维) 错误 46；开始 / 结束半径不匹配 ($ R1 - R2 > .15 * R1$)。	‘+’ 垂直于圆形平面观看为顺时针 / 逆时针	最短 / 最长弧形。在完整圆形中，终点的位置定义了由方向参数所指的最短 / 最长路径。	1. 可以编程完整圆形。 2. 仅在二维中，终点 = 起点是有效的。因此，可以生成完整圆形： a. 通过设置终点 = 起点，在这种情况下，所有方向类型都会产生完整圆形。 b. 通过设置终点不等于起点而且使用完整方向类型。 3. 对于三维完整圆形：将位置设置为除了起点以外圆上的任意点，并使用完整方向类型之一。位置同时定义了弧形和最短方向类型。
经由点	两者	错误 44；共线性 错误 45；终点 = 起点	经由点始终决定着方向。	经由点始终决定着方向。方向操作数仅用于确定圆是完整的还是不完整的。	1. 可以编程完整圆形。 2. 对于完整圆形：将位置设置为除了起点以外圆上的任意点，并使用完整方向类型之一。

模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

运动插补动态调整 (MCCD)

运动插补动态调整 (MCCD) 指令将启动指定坐标系路径动态参数的更改。根据运动类型，MCCD 更改系统中当前活跃的协调运动模式。

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或者在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。



MCCD

Motion Coordinated Change Dynamics	?	(EN)
Coordinate System	?	...
Motion Control	?	(DN)
Motion Type	?	
Change Speed	?	(ER)
Speed	??	
Speed Units	?	
Change Accel	?	
Accel Rate	?	
Accel Units	??	
Change Decel	?	
Decel Rate	?	
Decel Units	??	
Change Accel Jerk	?	
Accel Jerk	?	
Change Decel Jerk	??	
Decel Jerk	?	
Jerk Units	??	
Scope	?	

<< Less

MCCD 指令操作数 - 梯形图

操作数	类型	格式	描述
坐标系	COORDINATE_SY STEM	标签	轴的协调组。
运动控制	MOTION_ 指令	标签	用于访问指令状态参数的结构。
运动类型	SINT、INT 或 DINT	立即	1 = 协调运动
变更速度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
速度	SINT、INT、DINT 或 REAL	立即或标签	[协调单位]
速度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 1 = 最大值 %

MCCD 指令操作数 - 梯形图

操作数	类型	格式	描述
变更加速度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
加速度	SINT、INT、DINT 或 REAL	立即或标 签	[协调单位]
加速度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 ² 1 = 最大值 %
变更减速度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
减速度	SINT、INT、DINT 或 REAL	立即或标 签	[协调单位]
减速度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 ² 1 = 最大值 %
变更加速度跃度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
加速跃度	SINT、INT、DINT 或 REAL	立即或标 签	您 必须 始终为加速度跃度 操作数输入值。如果模式 配置为 S 曲线，这个指令 仅使用该值。 • 加速跃度是坐标系的加 速跃度值。 使用这些值起步。 加速度跃度 = 100 (时间 %) 跃度单位 = 2
变更减速度跃度	SINT、INT 或 DINT	立即	0 = 否 1 = 是

MCCD 指令操作数 - 梯形图

操作数	类型	格式	描述
减速度跃度	SINT、INT、DINT 或 REAL	立即或标签	您 必须 始终为减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 <ul style="list-style-type: none"> 减速度跃度是坐标系的减速度跃度值。 使用这些值起步。 <ul style="list-style-type: none"> 减速度跃度 = 100 (时间%) 跃度单位 = 2
跃度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 ³ 1 = 最大值 % 2 = 时间 % (使用这个值起步)
范围	SINT、INT 或 DINT	立即	0 = 活跃的运动 1 = 活跃和挂起的运动



MCCD(CoordinateSystem, MotionControl, MotionType ChangeSpeed, Speed, SpeedUnits, ChangeAccel, AccelRate, AccelUnits, ChangeDecel, DecelRate, DecelUnits, ChangeAccelJerk, AccelJerk, ChangeDecelJerk, DecelJerk, JerkUnits, Scope);

结构文本

操作数与梯形图 MCCD 指令的操作数相同。

当输入枚举作为结构文本中的操作数值时，多个字枚举的输入必须没有空格。举例如下：当输入减速度单位时，值应该输入为 unitspersec² 而不是梯形图逻辑中显示的 Units per Sec²。

对于有枚举值的操作数，请按照以下方式输入您的选择：

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
坐标系	无枚举	标签
运动控制	无枚举	标签
运动类型	无枚举	标签 0 = 绝对值 1 = 增量
ChangeSpeed	否	0
	是	1

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
速度	无枚举	立即或标签
SpeedUnits	Unitspersec	0
	%ofmaximum	1
ChangeAccel	否	0
	是	1
加速度	无枚举	立即或标签
加速度单位	Unitspersec ²	0
	%ofmaximum	1
ChangeDecel	否	0
	是	1
减速度	无枚举	立即或标签
减速度单位	Unitspersec ²	0
	%ofmaximum	1
变更加速度跃度	无枚举	0 = 否 1 = 是
加速跃度	无枚举	立即或标签 您必须始终为加速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这个值起步。 加速度跃度 = 100 (时间 %)
变更减速度跃度	无枚举	0 = 否 1 = 是

此操作数	拥有这些选项可让您输入为 ...	
	文本	或者为
减速度跃度	无枚举	立即或标签 您 必须 始终为减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 使用这个值起步。 <ul style="list-style-type: none"> • 减速度跃度 = 100 (时间 %) • 跃度单位 = 2
跃度单位	Unitspersec ³	0
	%ofmaximum	1
	%oftime	2(使用这个值起步)
范围	无枚举	0 = 活跃的运动 1 = 活跃和挂起的运动

坐标系

坐标系操作数指定了定义坐标系维度的运动轴系。坐标系支持最大三个 (3) 主轴。

运动控制

以下控制位将受到 MCCD 指令的影响。

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时，启用位被设置。当梯级从真过渡到假时，启用位被复位。
.DN (完成) 位 29	当梯级从假过渡到真时，完成位被复位。当目标位置成功计算后，完成位被设置。
.ER (错误) 位 28	当梯级从假过渡到真时，错误位被复位。当目标位置未能成功计算后，错误位被设置。

运动类型

运动类型操作数决定要变更哪种运动模式。目前，协调运动是唯一可用的选项。

- 协调运动 - 当选中时，协调运动选项更改坐标系中目前活跃的运动。

变更速度

变更速度操作数确定是否要更改协调运动模式的速度。

- 否 - 对协调运动的速度不做变更。
- 是 - 通过速度和速度单位操作数中定义的值，更改协调运动的速度。

速度

速度操作数定义了协调运动路径上的最大速度。

速度单位

速度单位操作数定义了应用到速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

变更加速度

变更加速度操作数确定是否要更改协调运动模式的加速度。

- 否 - 对协调运动的加速度不做变更。
- 是 - 通过加速度和加速度单位操作数中定义的值，更改协调运动的加速度。

加速度

加速度操作数定义了协调运动路径上的最大加速度。

加速度单位

加速度单位操作数定义了应用到加速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

变更减速度

变更减速度操作数确定是否要更改协调运动模式的减速度。

- 否 - 对协调运动的减速度不做变更。
- 是 - 通过减速度和减速度单位操作数中定义的值，更改协调运动的减速度。

减速度

减速度操作数定义了协调运动路径上的最大减速度。

减速度单位

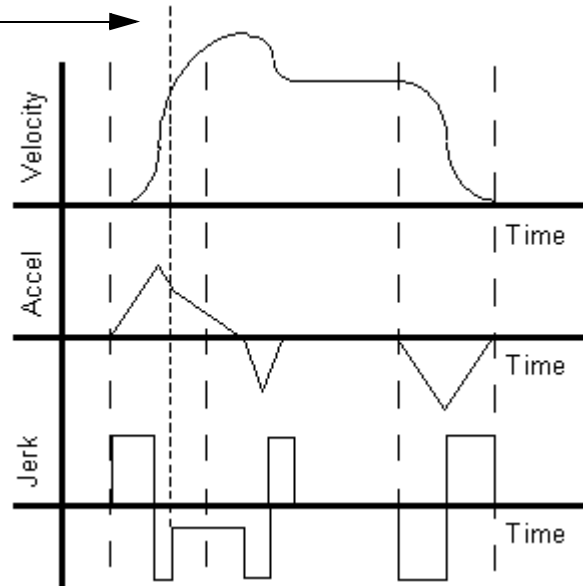
减速度单位操作数定义了应用到减速度操作数上的单位，可以直接采用指定坐标系的协调单位，或者作为坐标系中定义的最大值的百分比。

加速度和减速度值的变更对运动模式的影响

下图介绍当速度接近最大值时，如果使用 MCCD 指令来降低加速度会出现什么情况。新的加速度跃度变得更小，进一步限制了加速度的最大变化。发生速度过冲，因为加速度达到零所需的时间增加了。生成另一个模式将速度重新带到编程最大值。

对加速度变更的影响

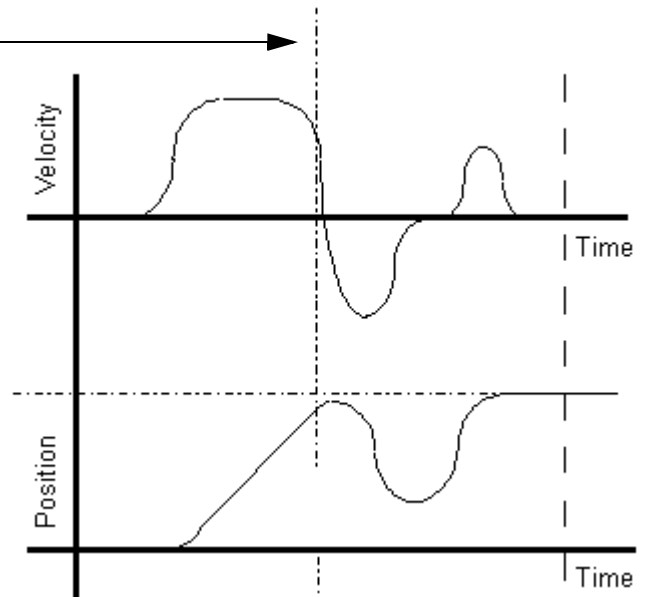
加速度被降低的点。 →



下图介绍当速度和位置接近其目标终点时，如果使用 MCCD 指令来降低减速度会出现什么情况。新的减速度跃度变得更小。减速到零所需的时间造成速度过冲，通过零点变为负值。轴运动也会反转方向，直到速度返回到零为止。生成一个附加的模式将位置重新带到编程目标。

对减速度变更的影响

减速度被降低的点。 →



变更加速度跃度

变更加速度跃度操作数确定是否要更改协调运动模式的加速度跃度。

- 否 - 对协调运动的加速度跃度不做变更。
- 是 - 通过加速度跃度和跃度单位操作数中定义的值，更改协调运动的加速度。

加速跃度

加速跃度定义了编程运动的最大加速跃度。有关计算加速跃度的更多信息，请参阅本章的“跃度单位”部分。

变更减速度跃度

变更减速度跃度操作数确定是否要更改协调运动模式的减速度跃度。

- 否 - 对协调运动的减速度跃度不做变更。
- 是 - 通过减速度跃度和跃度单位操作数中定义的值，更改协调运动的减速度。

减速跃度

减速跃度定义了编程运动的最大减速跃度。有关计算减速跃度的更多信息，请参阅本章的“跃度单位”部分。

跃度单位

跃度单位定义了向加速跃度和减速跃度操作数所输入的值上应用的单位。所输入的值直接以指定坐标系的位置单位为单位，或者为百分比。当配置使用最大值 % 时，跃度被应用为坐标系属性中指定的最大加速跃度和最大减速跃度操作数的百分比。当配置为使用时间 % 时，该值为基于指令中指定的速度、加速度和减速度的一个百分比。

如果您要将工程单位换算成 % 时间，请使用以下公式。

对于加速跃度：

$$j_a [\text{EU/s}^3] = \frac{\alpha_{\text{max}}^2 [\text{EU/s}^2]}{v_{\text{max}} [\text{EU/s}]} \left(\frac{200}{j_a [\% \text{ of time}]} - 1 \right)$$

对于减速跃度：

$$j_a [\text{EU/s}^3] = \frac{\alpha_{\text{max}}^2 [\text{EU/s}^2]}{v_{\text{max}} [\text{EU/s}]} \left(\frac{200}{j_a [\% \text{ of time}]} - 1 \right)$$

»Á¼° fZ“TM¾¼’ºp±ººaaÀ,,Š...¼§ŠÃµ•CEªŁ~“đp¼”D““æ¼¼’p¾¼°Ł

对于加速跃度：

$$j_a [\% \text{ of time}] = \frac{2}{1 + \frac{j_a [\text{EU/s}^3] v_{\text{max}} [\text{EU/s}]}{\alpha_{\text{max}}^2 [\text{EU/s}^2]}} 100$$

对于减速跃度：

$$j_a [\% \text{ of time}] = \frac{2}{1 + \frac{j_a [\text{EU/s}^3] v_{\text{max}} [\text{EU/s}]}{\alpha_{\text{max}}^2 [\text{EU/s}^2]}} 100$$

范围

为范围操作数选择活跃的运动将指定变更仅影响活跃的协调运动指令的运动动态参数。选择活跃和挂起的运动将指定变更影响活跃的协调运动指令以及队列中任何挂起的协调运动指令的运动动态参数。目前，队列大小被限制为活跃指令之后有一条指令。

MCCD 是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志

不受影响

故障条件

无

错误代码：

参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码：

扩展错误代码可帮助进一步定义为这条特殊指令所提供的错误消息。其行为取决于它们关联什么错误代码。

伺服关闭状态 (5)、关闭状态 (7)、轴类型非伺服 (8)、轴未配置 (11)、归零正在进行错误 (16) 以及非法轴数据类型 (38) 错误的扩展错误代码功能都是非常类似的。对于扩展错误代码将显示 $0...n$ 的数字。这个数字是坐标系的索引，表示正处于错误状况中的轴。

对于 MCCD 指令，错误代码 13 - 参数超出范围，扩展错误将返回一个数字，表示有问题的参数编号，这些参数在面板上以数字顺序从上到下（以零开始）列出。例如，2 表示运动类型的参数值有错误。

引用的错误代码和编号	扩展错误数字指示符	指令参数	描述
参数超出范围 (13)	2	运动类型	运动类型为小于 0 或者大于 1。
参数超出范围 (13)	4	速度	速度小于 0。
参数超出范围 (13)	7	加速度	加速度小于或等于 0。
参数超出范围 (13)	10	减速度	减速度小于或等于 0。

对于错误代码 54 - 最大减速度值为零，如果扩展错误代码返回一个正数 ($0-n$)，这表示坐标系中有问题的轴。转到坐标系属性的常规选项卡，查看轴网格的方括号 ([]) 列的下面，确定轴的最大减速度值是否为 0。单击有问题的轴旁边的省略号按钮即可访问轴属性屏幕。转到动态参数选项卡，对最大减速度值进行相应的更改。如果扩展代码编号为 -1，这表示坐标系的最大减速度值为 0。转到坐标系属性的动态参数选项卡，更正最大减速度值。

MCCD 对状态位的更改：

无影响。

模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

运动插补停止 (MCS)

运动插补停止 (MCS) 指令发起协调运动的控制停止。任何挂起的运动模式都被取消。

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

注意



速度和 / 或终点位置过冲风险

如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

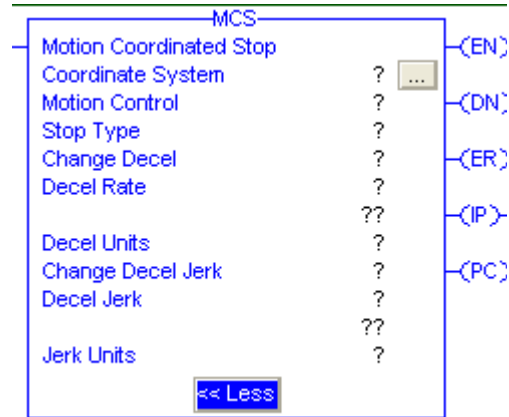
当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。

操作数

梯形图



操作数	类型	格式	描述	
坐标系	COORDINATE_SY STEM	标签	坐标系名称	
运动控制	MOTION_ 指令	标签	指令的控制标签	
停止类型	DINT	立即	如果您需要	选择这种停止类型
			停止坐标系轴的所有运动，停止坐标系参与的任何坐标转换	全部 (0)
			仅停止协调运动	协调运动 (2)
变更减速度	DINT	立即	如果您需要	那么选择
			使用坐标系的 ^{最大减速度}	否 (0)
			指定减速度	是 (1)
减速度	REAL	立即或标签	重要事项： 如果运动正在进行之中而您降低减速度，则轴可能会过冲其目标位置。 沿着协调运动路径的减速。指令使用这个值： <ul style="list-style-type: none"> • 只有当变更减速为是时。 • 仅适用于协调运动。 输入大于 0 的值。	
减速度单位	DINT	立即	0 = 单位每秒 ² 1 = 最大值 %	

操作数	类型	格式	描述
变更减速度跃度	SINT、INT 或 DINT	立即	0 = 否 1 = 是
减速度跃度	SINT、INT、DINT 或 REAL	立即或标签	您 必须 始终为减速度跃度操作数输入值。如果模式配置为 S 曲线，这个指令仅使用该值。 减速度跃度是坐标系的减速度跃度值。 使用这些值起步。 <ul style="list-style-type: none"> • 减速度跃度 = 100 (时间 %) • 跃度单位 = 2
跃度单位	SINT、INT 或 DINT	立即	0 = 单位每秒 ³ 1 = 最大值 % 2 = 时间 % (使用这个值起步)



结构文本

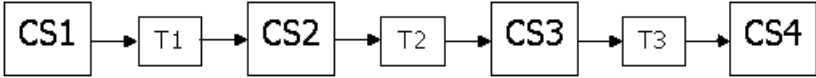
MCS(CoordinateSystem,
MotionControl,StopType,
ChangeDecel,DecelRate,DecelUnits,
ChangeDecelJerk,DecelJerk,
JerkUnits);

结构文本操作数与梯形图操作数相同。输入停止类型和减速单位，中间不留空格。

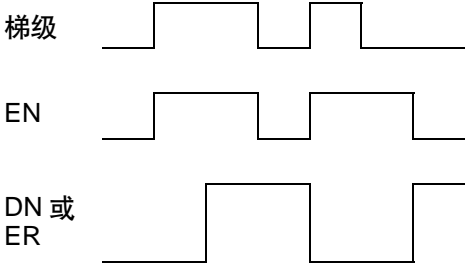
示例：输入坐标系操作数 CoordinateSystem。

停止类型如何影响坐标转换

下表介绍停止类型如何影响参与坐标转换的坐标系。

停止类型	描述
全部	<p>这种停止类型：</p> <ul style="list-style-type: none"> • 停止指定坐标系中的所有轴。它同时停止与此坐标系共轴的任何坐标系的轴。 • 取消坐标系参与的任何坐标转换。
协调运动	<p>这种停止类型仅停止协调运动。任何坐标转换仍保持活跃状态。</p>
协调坐标转换	<p>这种停止类型取消与指定坐标系相关的坐标转换。在所有相关的目标坐标系上，所有坐标转换相关的运动都会停止。但是，源坐标轴将继续按照指令的命令运动。</p> <p>示例</p> <p>如果四个坐标系通过三个坐标转换联动。第一个坐标系 (CS1) 为源，并且正在处理命令运动。</p>  <pre> graph LR CS1[CS1] --> T1[T1] T1 --> CS2[CS2] CS2 --> T2[T2] T2 --> CS3[CS3] CS3 --> T3[T3] T3 --> CS4[CS4] </pre> <p>在 CS2 上执行 MCS 指令并使用停止类型为协调坐标转换将会导致：</p> <ul style="list-style-type: none"> • 坐标转换 T1 和 T2 被取消。 • 坐标转换 T3 保持活跃。 • CS1 中的轴保持运动。 • 坐标系 CS2 和 CS3 中的轴通过 MCS 指令中选择的减速度或者最大协调减速度而停止。 • CS4 中的轴跟随各自的 CS3 轴。 <p>在运动轴停止 (MAS) 指令中，停止类型全部也会取消坐标转换。</p>

MOTION_INSTRUCTION 数据类型

要查看是否	检查这个位是否处于打开状态	数据类型	注意:
梯级状态为真。	EN	BOOL	<p>有时候即使梯级变成假，EN 位仍保持开的状态。如果梯级在指令完成或出错之前变成假，就会出现这种情况。</p>  <p>梯级</p> <p>EN</p> <p>DN 或 ER</p>
停止被成功发起。	DN	BOOL	
出现错误。	ER	BOOL	
轴正在停止。	IP	BOOL	<p>这些操作的任何一个都会结束 MCS 指令，并关闭 IP 位：</p> <ul style="list-style-type: none"> • 坐标系被停止。 • 另外一条 MCS 指令取代这条 MCS 指令。 • 关闭指令。 • 故障操作。
轴被停止。	PC	BOOL	PC 位保持开的状态，直到梯级进行一次从假到真的过渡。

算术状态标志:

不受影响

故障条件:

无

错误代码:

参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码：

参见[运动控制指令错误代码 \(ERR\)](#)。内有关于如何使用扩展错误代码的信息。

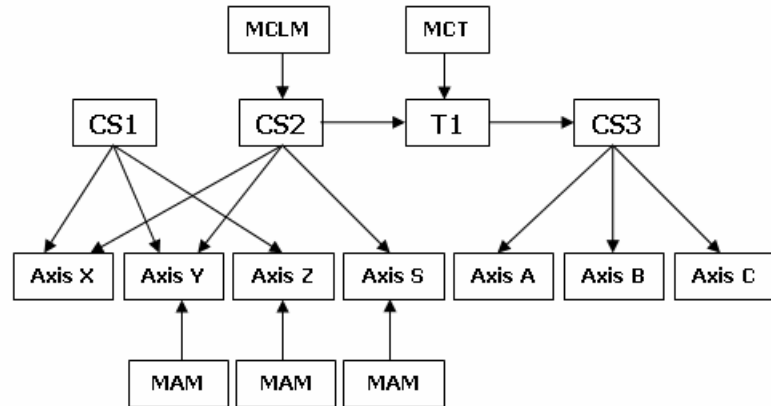
对状态位的更改

指令执行时会更改以下状态位。

在以下标签中	这个位	当停止类型为	变为
轴	CoordinatedMotionStatus	—————▶	关闭，当协调运动停止后
	TransformStateStatus	协调运动	不改变
		<ul style="list-style-type: none"> • 全部 • 协调坐标转换 	关闭
	ControlledByTransformStatus	协调运动	关闭，当轴停止而且 MCS 指令的 PC 位打开时
<ul style="list-style-type: none"> • 全部 • 协调坐标转换 		关闭	
坐标系	MotionStatus	—————▶	关闭，当协调运动停止后
	AccelStatus	—————▶	关闭
	DecelStatus	—————▶	打开（在停止过程中），然后关闭（当停止完成时）
	StoppingStatus	—————▶	打开（在停止过程中），然后关闭（当 PC 位打开时）
	MoveStatus	—————▶	关闭
	MoveTransitionStatus	—————▶	关闭
	MovePendingStatus	—————▶	关闭
	TransformSourceStatus	协调运动	不改变
		<ul style="list-style-type: none"> • 全部 • 协调坐标转换 	关闭
	TransformTargetStatus	协调运动	不改变
<ul style="list-style-type: none"> • 全部 • 协调坐标转换 		关闭	

停止类型如何影响坐标转换和轴运动示例

假设您处于这样的一种情况。



其中：

- 坐标系 1(CS1) 包含 X、Y 和 Z 轴。
- 坐标系 2(CS2) 包含 Y、Z 和 S 轴。
- 坐标系 3(CS3) 包含 A、B 和 C 轴。
- 坐标转换 (T1) 将源坐标系 CS2 与目标 CS3 联动。
- CS2(XYS) 轴映射到 CS3(ABC) 轴。
- MAM 指令在 Y、Z 和 S 轴上执行。
- MCLM 指令在 CS2 上执行。
- MCT 指令的执行以 CS2 作为源，CS3 作为目标。
- 在 CS2 或 CS3 上不执行协调指令。

这个表格显示使用不同停止类型执行各种 MCS 和 MAS 时的结果。

MCS 和 MACS 指令以及停止类型

指令	停止类型	结果
在 CS1 上的 MCS	全部	在 CS2 上的 MCLM 指令将停止。
		在 Y 上的 MAM 将停止。
		在 Z 上的 MAM 将停止。
		在 S 上的 MAM 将继续。
		T1 被取消。
		轴 ABC 将停止，因为取消了坐标转换。

MCS 和 MACS 指令以及停止类型

指令	停止类型	结果
在 CS2 上的 MCS	全部	在 CS2 上的 MCLM 指令将停止。
		在 Y 上的 MAM 将停止。
		在 S 上的 MAM 将停止。
		在 Z 上的 MAM 将继续。
		T1 被取消。
		轴 ABC 将停止，因为取消了坐标转换。
在 CS3 上的 MCS	全部	在 CS2 上的 MCLM 指令将继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 被取消。
		轴 ABC 将停止，因为取消了坐标转换。
在 CS1 上的 MCS	协调运动	在 CS2 上的 MCLM 指令将继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 CS2 上的 MCS	协调运动	在 CS2 上的 MCLM 指令将停止。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 CS3 上的 MCS	协调运动	在 CS2 上的 MCLM 指令将继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 Y 上的 MAS	全部	在 CS2 上的 MCLM 指令将停止。
		在 Y 上的 MAM 将停止。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 被取消。
		轴 ABC 将停止，因为取消了坐标转换。

MCS 和 MACS 指令以及停止类型

指令	停止类型	结果
在 Y 上的 MAS	运动	在 CS2 上的 MCLM 指令将继续。
		在 Y 上的 MAM 将停止。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 Z 上的 MAS	全部	在 CS2 上的 MCLM 指令将继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将停止。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 Z 上的 MAS	运动	在 CS2 上的 MCLM 指令将继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将停止。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 CS1 上的 MCS	协调坐标转换	在 CS2 上的 MCLM 指令继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		T1 保持活跃。
		轴 ABC 将跟随各自的 CS2 轴。
在 CS2 上的 MCS	协调坐标转换	T1 被取消。
		在 CS2 上的 MCLM 指令继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		轴 ABC 将停止，因为取消了坐标转换。
在 CS3 上的 MCS	协调坐标转换	T1 被取消。
		在 CS2 上的 MCLM 指令继续。
		在 Y 上的 MAM 将继续。
		在 S 上的 MAM 将继续。
		在 Z 上的 MAM 将继续。
		轴 ABC 将停止，因为取消了坐标转换。

模式操作数

当使用这个指令时，您应该考虑[模式操作数](#)。有关更多信息，请参阅[页码 68](#)。

运动插补关闭 (MCSD)

使用运动插补关闭 (MCSD) 指令可在命名的坐标系的所有轴上执行一次受控停机。

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

注意



速度和 / 或终点位置过冲风险

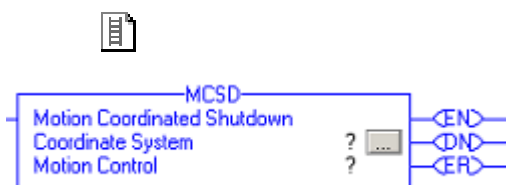
如果您通过任何方法动态地更改运动参数，即通过运动动态指令 (MCD 或 MCCD) 进行更改，或在最后一个指令完成之前开始一条新的指令，要小心速度和 / 或终点位置过冲的风险。

当运动正在减速或者接近减速点时，如果最大减速度降低，则梯形速度模式可能会过冲。

S 曲线速度模式在下面的情况下可能会过冲：

- 当运动正在减速或者接近减速点时最大减速度降低；或者
- 最大加速度跃度降低而轴正在加速。但是请牢记，如果用时间 % 的单位指定，跃度可能会间接被改变。

操作数梯形图



操作数	类型	格式	描述
坐标系	COORDINATE_SYSTEM	标签	轴的协调组。
运动控制	MOTION_INSTRUCTION	标签	用于访问指令状态参数的结构。



MCSD(CoordinateSystem, MotionControl);

结构文本

操作数与梯形图 MCSD 指令的操作数相同。

坐标系

坐标系操作数指定了定义笛卡尔坐标系维度的运动轴系。对于这个版本，坐标系支持最大三个 (3) 主轴。只有配置为主轴（最多三个）的那些轴将包含在协调速度的计算中。

运动控制

以下控制位将受到 MCSD 指令的影响。

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时，启用位被设置。当梯级从真过渡到假时，启用位被复位。
.DN (完成) 位 29	当协调关闭被成功发起后，完成位被设置。当梯级从假过渡到真时，完成位被复位。
.ER (错误) 位 28	当协调关闭未能成功发起时，错误位被设置。当梯级从假过渡到真时，错误位被复位。

MCSD 是一条转换指令：

- 在梯形图中，当每次应执行指令时，将梯级输入条件从清除切换到设置。
- 在结构文本中，限制指令只对转换执行。请参阅附录 C。

算术状态标志

不受影响

故障条件

无

错误代码

参见[运动控制指令错误代码 \(ERR\)](#)。

MCSD 对状态位的更改

状态位提供一种监视运动控制指令进度的途径。有三种类型的状态位提供有关信息。分别是：轴状态位、坐标系状态位和协调运动状态位。当 MCS 指令发起后，状态位将经历以下变化。

轴状态位

位名称	影响
CoordinatedMoveStatus	被清除

坐标系状态位

位名称	影响
ShutdownStatus	当 MCSD 被执行而且所有关联轴都被关闭后设置。
ReadyStatus	在 MCSD 执行后清除。

协调运动状态位

位名称	影响
AccelStatus	在 MCSD 执行后清除。
DecelStatus	在 MCSD 执行后清除。
ActualPosToleranceStatus	在 MCSD 执行后清除。
CommandPosToleranceStatus	在 MCSD 执行后清除。

协调运动状态位

位名称	影响
StoppingStatus	在 MCSD 执行后清除。
MoveStatus	在 MCSD 执行后清除。
MoveTransitionStatus	在 MCSD 执行后清除。
MovePendingStatus	在 MCSD 执行后清除。
MovePendingQueueFullStatus	在 MCSD 执行后清除。

运动插补坐标转换 (MCT)

使用 MCT 指令可启动一个将两个坐标系连起来的转换。这类似于双向齿轮。使用坐标转换的一种方法就是，将非笛卡尔机器人移动到笛卡尔位置。

注意

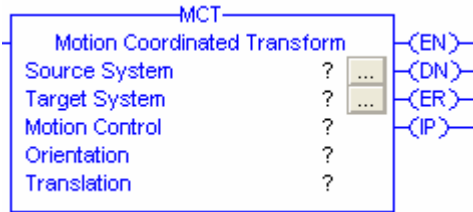


用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

重要提示

您只能在 1756-L6x 控制器上使用这条指令。

操作数：梯形图



操作数	类型	格式	描述	
源系统	COORDINATE_SYSTEM	标签	您用于编程该运动的坐标系。通常，这是笛卡尔坐标系。	
目标系统	COORDINATE_SYSTEM	标签	控制着实际设备的非笛卡尔坐标系	
运动控制	MOTION_INSTRUCTION	标签	指令的控制标签	
方位	REAL[3]	数组	希望绕着 X1、X2 或 X3 轴旋转目标位置吗？	
			如果	那么
			否	将数组值保持为零。
			是	向数组内输入旋转的度数。在数组的第一个元素中填写绕着 X1 旋转的度数，以此类推。
			使用三个 REAL 的数组，即使坐标系只有一个或两个轴。	
平移	REAL[3]	数组	希望沿着 X1、X2 或 X3 轴偏移目标位置吗？	
			如果	那么
			否	将数组值保持为零。
			是	向数组内输入偏移距离。向坐标单位内输入偏移距离。在数组的第一个元素中填写 X1 的偏移距离，以此类推。
			使用三个 REAL 的数组，即使坐标系只有一个或两个轴。	

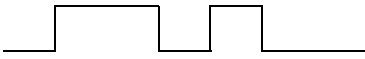
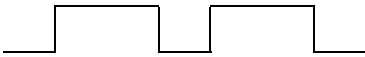
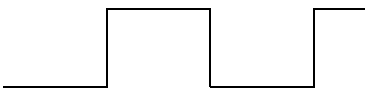


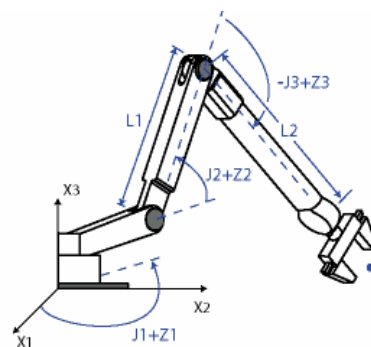
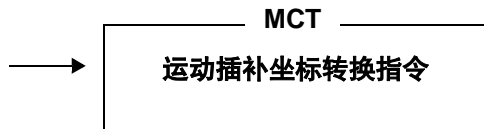
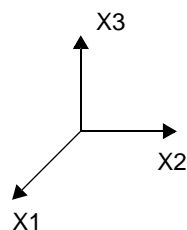
结构文本

MCT(Source System, Target System, Motion Control, Orientation, Translation);

结构文本操作数与梯形图操作数相同。

MOTION_INSTRUCTION 数据类型

要查看是否	检查这个位是否处于打开状态	数据类型	注意:
梯级状态为真。	EN	BOOL	<p>有时候即使梯级变成假，EN 位仍保持开的状态。如果梯级在指令完成或出错之前变成假，就会出现这种情况。</p> <p>梯级 </p> <p>EN </p> <p>DN 或 ER </p>
指令完成。	DN	BOOL	指令完成后坐标转换继续运行。
出现错误。	ER	BOOL	标识运动控制标签的错误代码字段内列出的错误编号，然后参阅本手册的 运动控制指令错误代码(ERR) （在 页码 391 上）。
坐标转换过程正在进行之中。	IP	BOOL	<p>这些操作的任何一个都会取消坐标转换，并关闭 IP 位：</p> <ul style="list-style-type: none"> • 适用的停止指令 • 关闭指令 • 故障操作



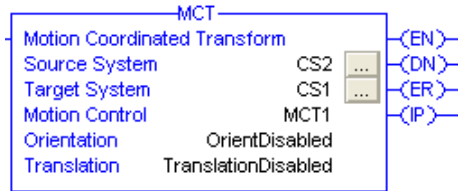
您将虚轴系移到笛卡尔位置 (X1, X2, X3)。

坐标转换将运动转换为关节角，并移动机器人。

坐标转换控制最多三个机器人关节：J1、J2 和 J3。
MCT 指令在两个坐标系之间的数据流动

下面的插图显示当 MCT 指令工作时数据是如何流动的。CS1 是一个笛卡尔坐标系，包含 X1、X2 和 X3 轴，作为 MCT 指令的源。CS2 是关节坐标系，包含 J1、J2 和 J3 轴，作为 MCT 指令的目标。

通过 MCT 指令执行运动时的数据流动 - 正向坐标转换



输入数据

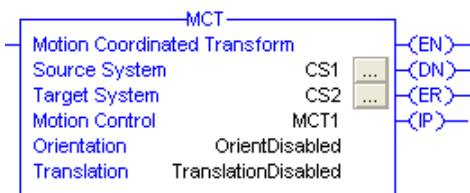
CS2: 数据	源
关节位置 (J1, J2, J3)	机器实体坐标系
连接长度 (L1, L2)	坐标系对话框
基本偏移 (X1b, X2b, X3b)	坐标系对话框
末端执行器偏移 (X1e, X2e, X3e)	坐标系对话框
零角度方位 (Z1, Z2, Z3)	坐标系对话框
方位 (数组 [3])	指令面板
平移 (数组 [3])	指令面板

活跃指令
 MCT

计算输出

CS1: 数据	目标
笛卡尔位置 (X1, X2, X3)	机器虚拟坐标系

通过 MCT 指令执行运动时的数据流动 - 反向坐标转换



输入数据

CS1: 数据	源
笛卡尔位置 (X1, X2, X3)	机器虚拟坐标系
连接长度 (L1, L2)	坐标系对话框
基本偏移 (X1b, X2b, X3b)	坐标系对话框
末端执行器偏移 (X1e, X2e, X3e)	坐标系对话框
零角度方位 (Z1, Z2, Z3)	坐标系对话框
方位 (数组 [3])	指令面板
平移 (数组 [3])	指令面板

活跃指令
 MCT

计算输出

CS2: 数据	目标
关节位置 (J1, J2, J3)	机器实体坐标系

编程原则

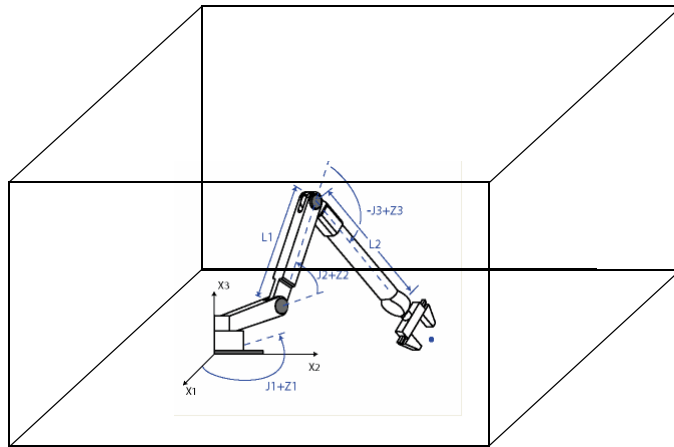
在使用 MCT 指令时请遵照以下指导原则。

注意



不要让机器人完全伸展或者完全缩回到自身上。否则它可能会以很高的速度开始运动。在那些位置，它失去自己作为左臂或右臂的配置。当出现这种情况时，它可能会以很高的速度开始运动。

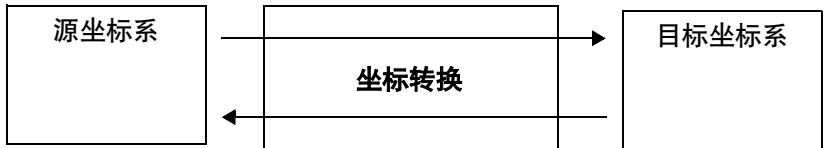
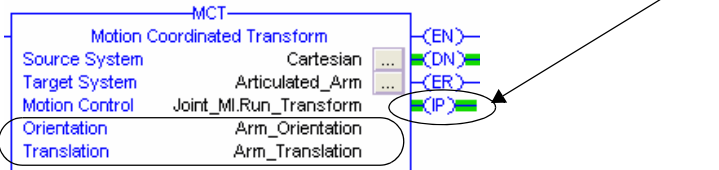
确定机器人的工作限制，将其保存在限制范围以内。



MCT 指令指导原则

指导原则	示例和注释
<p>为机器人的笛卡尔位置建立轴的坐标系。这些轴通常是虚拟的。</p>	 <p>坐标系中轴的个数。</p> <p>需要坐标转换的轴的个数。</p> <p>重要事项：在计算精度中您可能会看到截断错误。当以下两种条件都为真时会出现这种情况：</p> <ul style="list-style-type: none"> 在坐标转换中虚拟笛卡尔轴的转换常数非常小，例如 8000 次 / 位置单位。 非笛卡尔坐标系的连接长度非常小，例如 0.5 英寸。 <p>在坐标转换中最好给虚拟笛卡尔轴取较大的转换常数，例如 100,000 或 1,000,000 次 / 位置单位。机器人的最大行程限制为</p> $\frac{\pm 2^{31}}{\text{转换常数}} \text{ 协调单位}$
<p>为机器人的实际关节建立另一个坐标系。</p>	 <p>机器人几何类型</p> <p>坐标系中轴的个数。</p> <p>需要坐标转换的轴的个数。</p>

MCT 指令指导原则

指导原则	示例和注释
<p>在源和目标坐标系之间期望建立双向运动。</p> <p>使用 MCS 指令来取消坐标转换。</p>	<p>坐标转换是双向的。</p>  <p>当您开始坐标转换时，源坐标系的位置更改，以匹配目标坐标系的相应位置。其后，如果您移动任一系统，另一个系统随之运动。</p> <p>控制器继续控制轴，即使您停止扫描 MCT 指令或其梯级变为假。使用运动插补停止 (MCS) 指令可停止坐标系中的运动、取消坐标转换或者二者同时。</p>
<p>如果您改变了方位或平移，请再次执行 MCT 指令。</p>	<p>如果您希望坐标转换正在运行时改变方位或平移的值，</p>  <p>那么就再次执行该指令。要执行该指令，将梯级输入条件从假切换为真。</p> <p>同时，如果您改变了设备的几何属性，也要再次执行该指令。</p>

算术状态标志

不受影响

故障条件

无

错误代码

参见 [运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码：

扩展错误代码可帮助进一步定义为这条特殊指令所提供的错误消息。其行为取决于它们关联什么错误代码。

扩展错误代码

ERR	EXERR	处理措施	注意：
61	1	将两个坐标系都指定给运动组。	
	2	检查您正在使用的源和目标坐标系是否正确。	您不能使用同一个坐标系同时作为源和目标。
	3	将源坐标系的坐标转换维度设置为坐标系中轴的个数，最多为三。	
	4	将目标坐标系的坐标转换维度设置为要进行坐标转换的坐标系中轴的个数，最多为三。	
	5	使用不同的源坐标系。	您只能使用一个坐标系作为活跃的坐标转换的源。
	6	使用不同的目标坐标系。	您只能使用一个坐标系作为活跃的坐标转换的目标。
	7	寻找已经在其他坐标转换中使用的源或目标轴。使用坐标系中不同的轴。	您只能将一个轴用在一个源坐标系和一个目标坐标系中。
	8	使用一个不是这个坐标转换链源的目标坐标系。	您不能创建一个又重新回到原始源的圆形坐标转换链。
	9	检查您是否将正确的轴分配给每一个坐标系。	您不能将相同的轴用在源和目标坐标系中。
	10	停止两个坐标系中所有轴的运动过程（例如点动、移动和齿轮）。	如果任何运动过程正在控制源或目标的轴，则不能开始坐标转换。
	11	发起坐标转换连接所需资源不足。	
	12	设置连接长度。	您不能让连接长度为零。
	13	寻找源或目标的轴是否处于关闭状态。使用运动轴关闭复位 (MASR) 指令或直接命令来复位轴。	
	14	解除对所有源和目标的轴的禁止。	
	15	对于 Delta 或 SCARA Delta 机器人，检查为基本偏移和末端执行器偏移所配置的值。	对于 Delta 或 SCARA Delta 机器人，(X1b-X1e) 都不能小于 0.0。 对于 Delta 机器人，如果 L1 + (X1b-X1e) 的值大于 L2，也会出现这种错误。
	16	插件 SCARA 独立和 SCARA Delta 机器人配置，确保： <ul style="list-style-type: none"> 源坐标系的坐标转换维度配置为 2。 为源坐标系和目标坐标系配置的第三个轴是相同的。 	

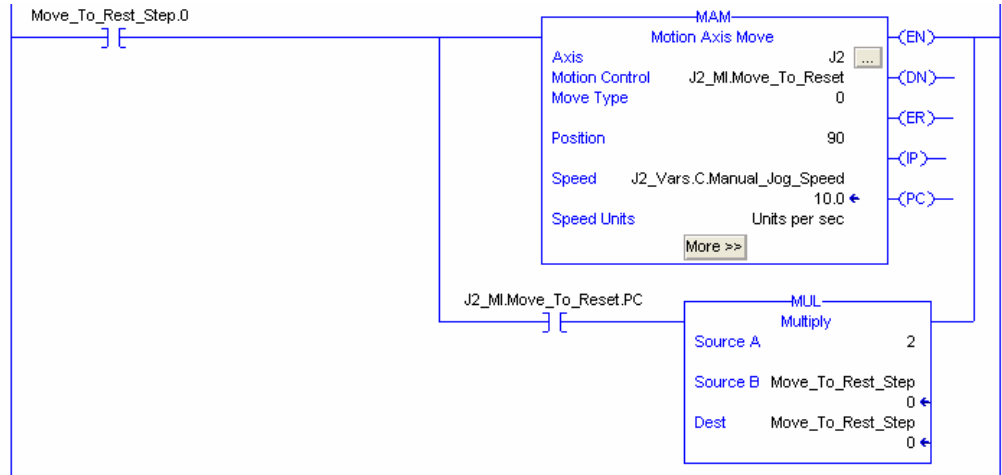
对状态位的更改

要查看是否	检查标签是否	而且这个位	有关
坐标系是否是活跃坐标转换的源。	坐标系	TransformSourceStatus	打开
坐标系是否是活跃坐标转换的目标。	坐标系	TransformTargetStatus	打开
轴是活跃坐标转换的一部分。	轴	TransformStateStatus	打开
轴因为某个坐标转换而正在运动。	轴	ControlledByTransformStatus	打开

例 1 - 拾取和放置梯形图

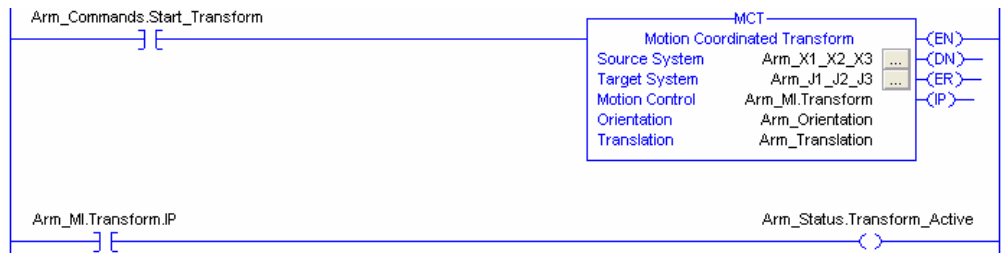
1. 移动到停靠例程

这个例程是一个运动序列，将关节独立的机器人放到左臂或右臂角的某个停靠位置。当 Move_To_Rest_Step.0 打开时，轴 J2 移动到 90YY。然后序列进入下一步。



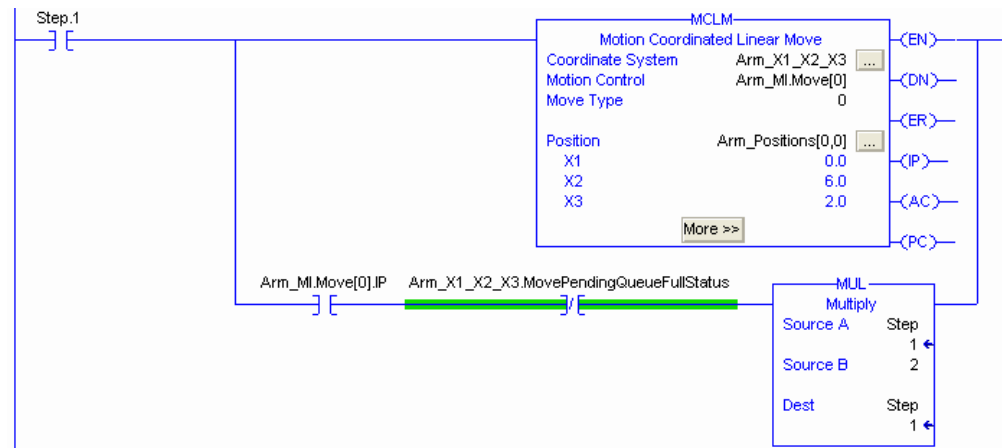
2. 开始坐标转换例程

当 Arm_Commands.Start_Transform 打开时，坐标转换开始。IP 位表明坐标转换正在运行。



3. 拾取和放置例程

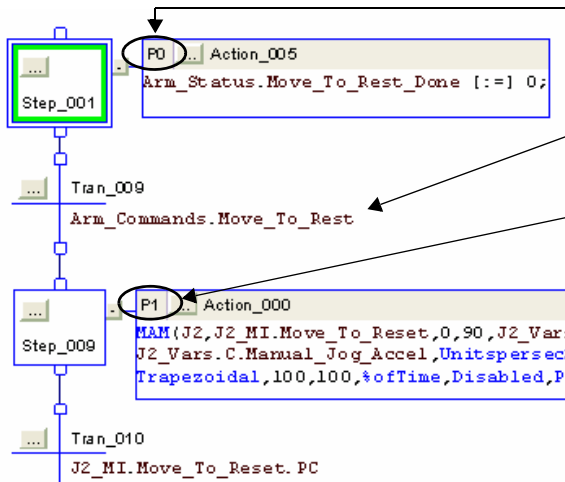
这个例程是移动笛卡尔坐标系的 MCLM 指令序列中的一个。机器人的关节跟随运动。当 Step.1 打开时，坐标系移到 0, 6, 2。然后运动处于正在进行中 (IP)，序列排队下一个运动。



拾取和放置 - 结构文本示例

1. 移动到停靠例程

这个例程是一个运动序列，将机器人放到左臂或右臂角的某个停靠位置。

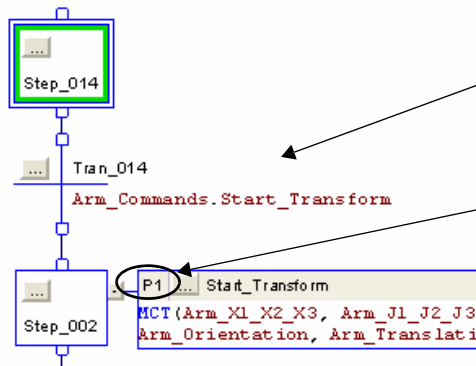


当 SFC 离开这个步序时，会关闭 Move_To_Reset_Done 位。

当 Move_To_Rest 位打开时，SFC 进入下一个步序。

这个步序将轴 J2 移到 90°。P1 限定符将这限制为第一次步序扫描。

2. 开始坐标转换例程

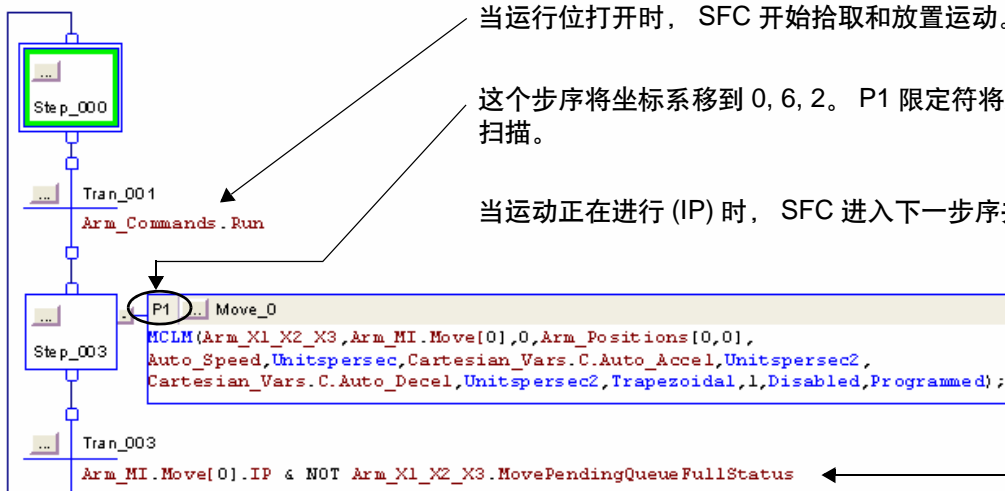


当 Move_To_Rest 位打开时，SFC 进入下一个步序。

这个步序开始坐标转换。P1 限定符将这限制为第一次步序扫描。

3. 拾取和放置例程

这个例程是移动笛卡尔坐标系的 MCLM 指令序列中的一个。机器人的关节跟随运动。



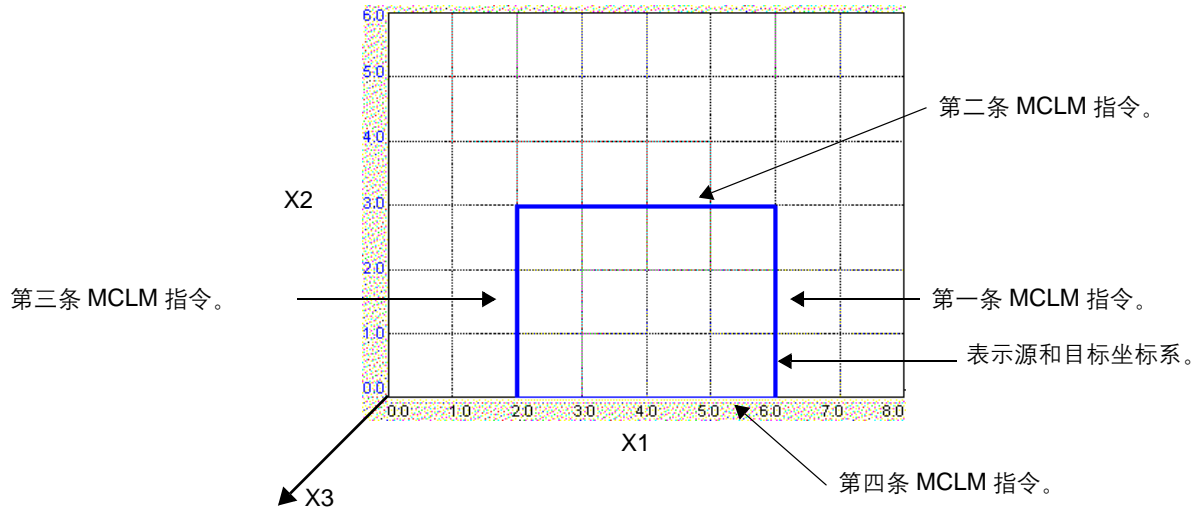
当运行位打开时，SFC 开始拾取和放置运动。

这个步序将坐标系移到 0, 6, 2。P1 限定符将这限制为第一次步序扫描。

当运动正在进行 (IP) 时，SFC 进入下一步序并排队下一个运动。

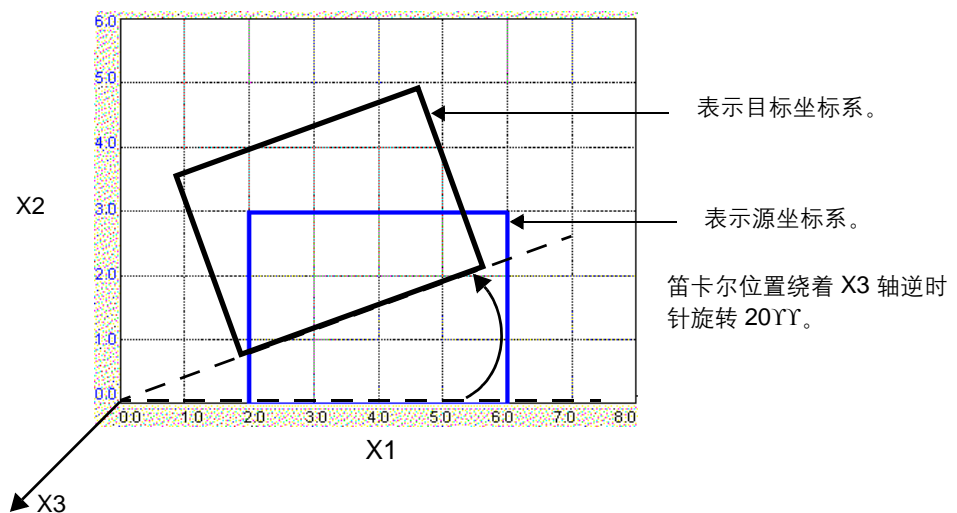
更改方位示例

如果您希望沿着长方形路径来移动坐标系。在这种情况下，执行 MCT 指令来开始坐标变换。然后，执行四个运动直线插补 (MCLM) 指令来产生长方形路径。



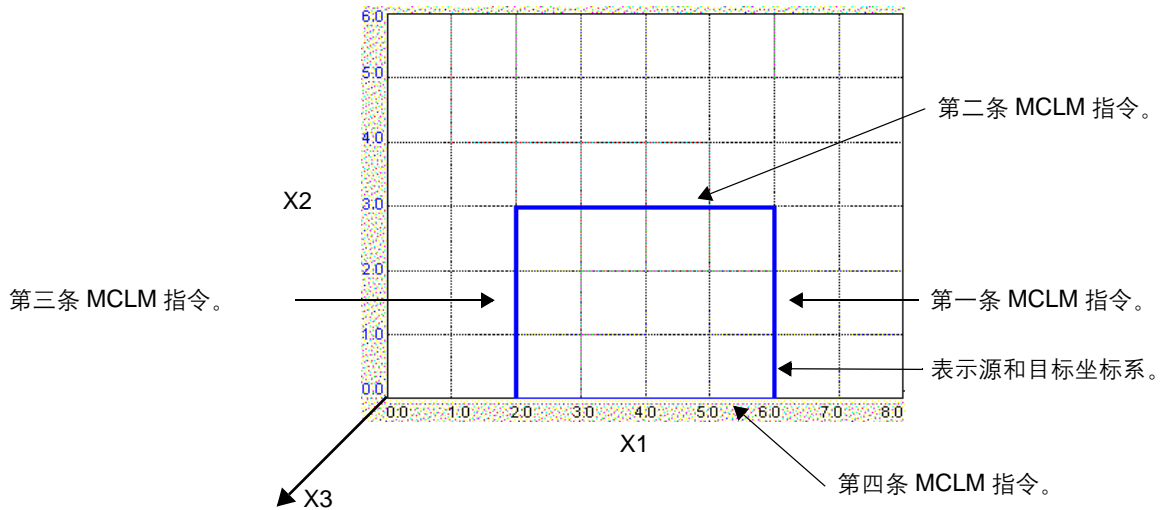
如果您希望将目标坐标系的笛卡尔位置绕着 X3 轴逆时针旋转 20°。

1. 向 MCT 指令中输入方位值 0°、0°、20°。
2. 再次执行 MCT 指令，即可将该方位应用到坐标转换中。
3. 再次执行相同的四条 MCLM 指令。



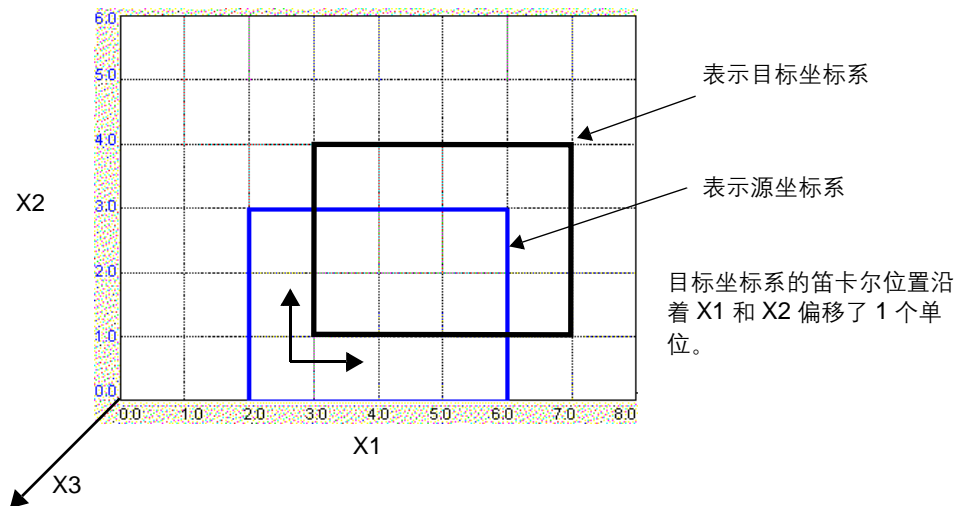
更改平移示例

如果您希望沿着长方形路径来移动坐标系。在这种情况下，执行 MCT 指令来开始坐标变换。然后，执行四个运动直线插补 (MCLM) 指令来产生长方形路径。



如果您希望将目标坐标系的笛卡尔位置沿着 X1 和 X2 轴偏移 1 个单位。

1. 向 MCT 指令中输入偏移值 1, 1, 0。
2. 再次执行 MCT 指令，即可将该平移应用到坐标转换中。
3. 再次执行相同的四条 MCLM 指令。



计算坐标转换位置 (MCTP)

使用 MCTP 指令可计算一个坐标系中某个点的位置在第二个坐标系中的等效点。

注意



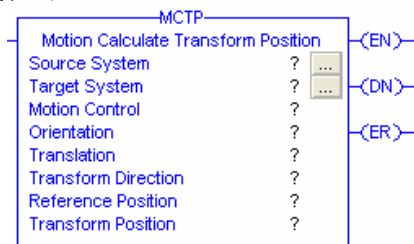
用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

重要提示

您只能在 1756-L6x 控制器上使用这条指令。

操作数

梯形图



操作数	类型	格式	描述	
源系统	COORDINATE_SYSTEM	标签	为机器人的笛卡尔位置建立的笛卡尔坐标系	
目标系统	COORDINATE_SYSTEM	标签	控制着实际设备的非笛卡尔坐标系	
运动控制	MOTION_INSTRUCTION	标签	指令的控制标签	
方位	REAL[3]	数组	希望绕着 X1、X2 或 X3 轴旋转目标位置吗?	
			如果	那么
			否	将数组值保持为零。
			是	向数组内输入旋转的度数。在数组的第一个元素中填写绕着 X1 旋转的度数，以此类推。
			使用三个 REAL 的数组，即使坐标系只有一个或两个轴。	

操作数	类型	格式	描述																												
平移	REAL[3]	数组	希望沿着 X1、X2 或 X3 轴偏移目标位置吗?																												
			如果	那么																											
			否	将数组值保持为零。																											
			是	向数组内输入偏移距离。向坐标单位内输入偏移距离。在数组的第一个元素中填写 X1 的偏移距离，以此类推。																											
使用三个 REAL 的数组，即使坐标系只有一个或两个轴。																															
坐标转换方向	DINT	立即	<table border="1"> <thead> <tr> <th>适用于机器人类型</th> <th>要计算</th> <th>基础转变为</th> <th>而且机器人</th> <th>选择</th> </tr> </thead> <tbody> <tr> <td>全部</td> <td>笛卡尔位置</td> <td>➔</td> <td>➔</td> <td>正向</td> </tr> <tr> <td>笛卡尔 Delta 2D Delta 3D SCARA Delta</td> <td>关节角</td> <td>➔</td> <td>➔</td> <td>反向</td> </tr> <tr> <td rowspan="2">关节独立 关节不独立</td> <td rowspan="4">关节角</td> <td rowspan="2">该点的同一象限</td> <td>右臂配置</td> <td>反向右臂</td> </tr> <tr> <td>左臂配置</td> <td>反向左臂</td> </tr> <tr> <td rowspan="2">SCARA 独立</td> <td rowspan="2">该点的相对象限</td> <td>右臂配置</td> <td>反向右臂镜像</td> </tr> <tr> <td>左臂配置</td> <td>反向左臂镜像</td> </tr> </tbody> </table>	适用于机器人类型	要计算	基础转变为	而且机器人	选择	全部	笛卡尔位置	➔	➔	正向	笛卡尔 Delta 2D Delta 3D SCARA Delta	关节角	➔	➔	反向	关节独立 关节不独立	关节角	该点的同一象限	右臂配置	反向右臂	左臂配置	反向左臂	SCARA 独立	该点的相对象限	右臂配置	反向右臂镜像	左臂配置	反向左臂镜像
			适用于机器人类型	要计算	基础转变为	而且机器人	选择																								
			全部	笛卡尔位置	➔	➔	正向																								
			笛卡尔 Delta 2D Delta 3D SCARA Delta	关节角	➔	➔	反向																								
			关节独立 关节不独立	关节角	该点的同一象限	右臂配置	反向右臂																								
						左臂配置	反向左臂																								
			SCARA 独立		该点的相对象限	右臂配置	反向右臂镜像																								
						左臂配置	反向左臂镜像																								
参考位置	REAL[3]	数组	<table border="1"> <thead> <tr> <th>如果坐标转换方向为</th> <th>则输入一个数组且含有</th> </tr> </thead> <tbody> <tr> <td>正向</td> <td>关节角</td> </tr> <tr> <td>反向</td> <td>笛卡尔位置</td> </tr> </tbody> </table>	如果坐标转换方向为	则输入一个数组且含有	正向	关节角	反向	笛卡尔位置																						
如果坐标转换方向为	则输入一个数组且含有																														
正向	关节角																														
反向	笛卡尔位置																														
坐标转换位置	REAL[3]	数组	储存计算位置的数组																												



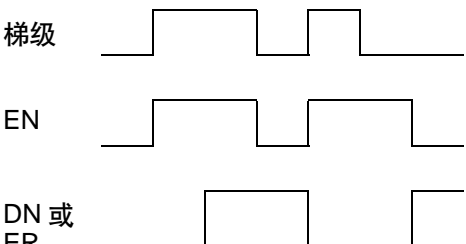
结构文本

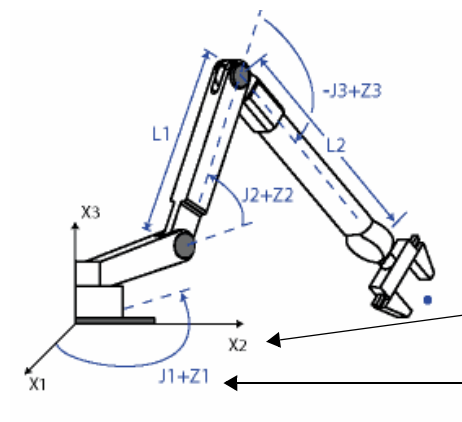
MCTP(Source System, Target System, Motion Control, Orientation, Translation, Transform Direction, Reference Position, Transform Position);

结构文本操作数与梯形图操作数相同。输入坐标转换方向，不带空格。

示例：将反向左臂的坐标转换方向输入为 InverseLeftArm。

运动控制指令数据类型

要查看是否	检查这个位是否处于打开状态	数据类型	注意:
梯级状态为真。	EN	BOOL	<p>有时候即使梯级变成假，EN 位仍保持开的状态。如果梯级在指令完成或出错之前变成假，就会出现这种情况。</p> 
指令完成。	DN	BOOL	
出现错误。	ER	BOOL	<p>标识运动控制标签的错误代码字段内列出的错误编号，然后参阅本手册的运动控制指令错误代码 (ERR)（在页码 391上）。</p>



您可以为指令提供 X1、X2 和 X3 位置，获得相应的 J1、J2 和 J3 角。

您可以为指令提供 J1、J2 和 J3 角，获得相应的 X1、X2 和 X3 位置。

MCTP 指令类似于 MCT 指令，除了 MCTP 指令不能开始坐标转换以外。您每次执行它时，就会计算一个位置。

编程原则

在使用 MCTP 指令时请遵照以下指导原则。

MCTP 指令指导原则

指导原则	示例和注释
将梯级从假切换到真以执行指令。	这是一条转换指令。在梯形图中，每当您希望执行指令时，将梯级输入条件从假切换到真即可。
在结构文本中，限制指令只对转换执行。	在结构文本中，指令每次被扫描时就会执行。限制指令只对转换执行。使用下面的方法之一： <ul style="list-style-type: none"> • SFC 操作限定符 • 结构文本构建

算术状态标志

不受影响

故障条件

无

错误代码

参见[运动控制指令错误代码 \(ERR\)](#)。

扩展错误代码

无

对状态位的更改

无

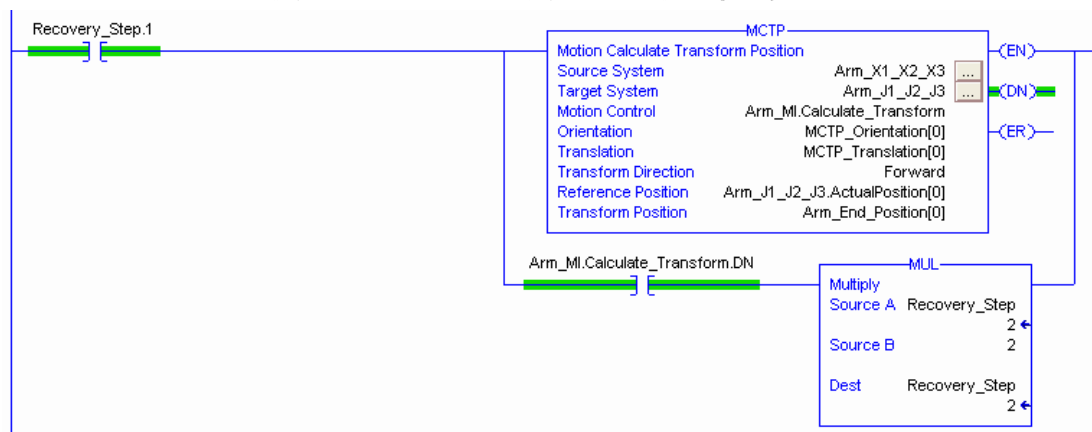
示例 1

如果您希望编写一个故障恢复序列。作为步序之一，您希望获取关节独立机器人的当前位置。在这种情况下，如果知道机器人的关节角，就可使用 MCTP 指令来计算机器人的笛卡尔位置。

计算位置 - 梯形图

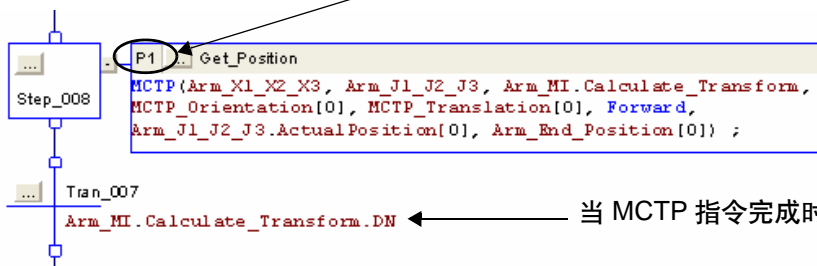
如果 Recovery_Step.1 打开，则根据其当前的关节角来计算机器人的 X1、X2 和 X3 位置。

当指令完成时，MUL 指令将序列带入下一步。



计算位置 - 结构文本

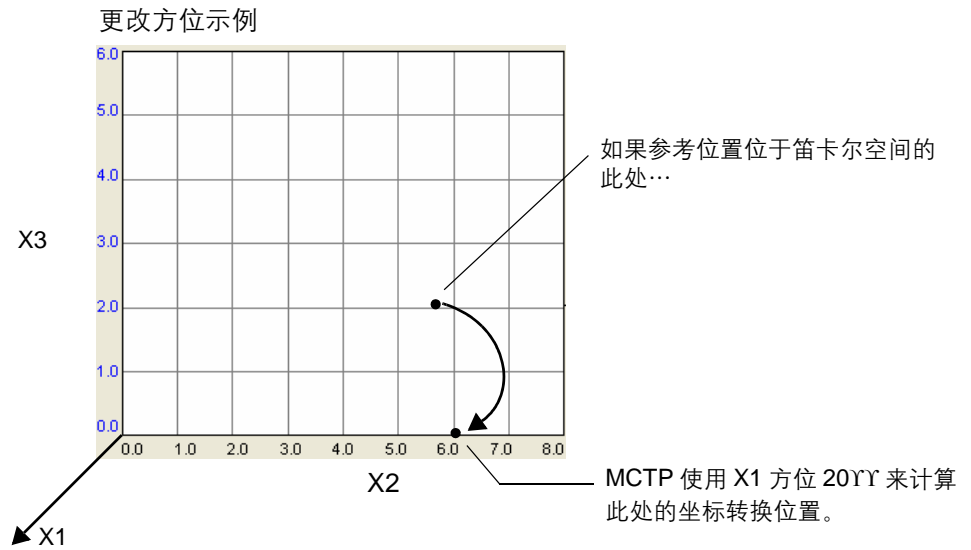
这个步序根据其当前的关节角来计算机器人的 X1、X2 和 X3 位置。P1 限定符将这限制为第一次步序扫描。



当 MCTP 指令完成时，SFC 进入下一个步序。

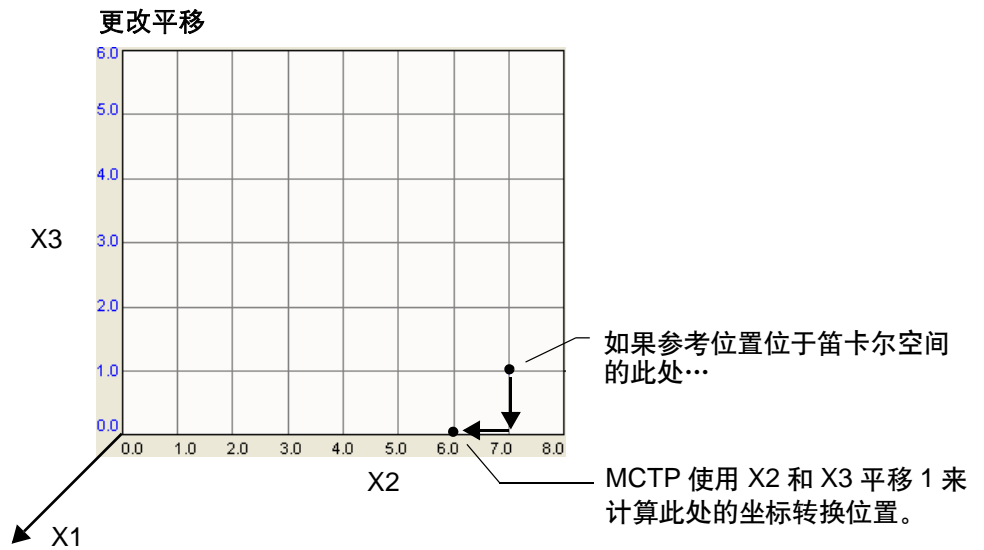
示例 2

如果您希望在示例 1 中输入方位值 20YY, 0YY, 0YY。在那个示例中，MCTP 指令进行一次正向坐标转换。



示例 3

如果您希望在示例 1 中输入平移值 0, 1, 1。在那个示例中，MCTP 指令进行一次正向坐标转换。



示例 4

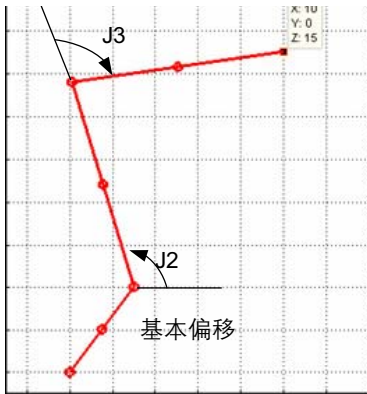
如果您的机器人有基本偏移，则可通过最多四种不同方式到达给定的点。如果您的机器人几何参数如下：

- L1 = 10
- L2 = 10
- X1b = 3.0

- X3b = 4.0

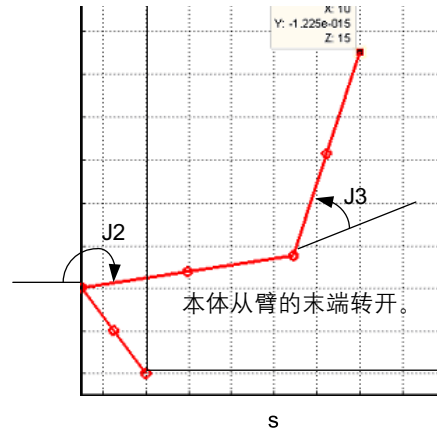
此例介绍如何到达位置 X1 = 10、X2 = 0 和 X3 = 15
坐标转换方向。

反向左臂



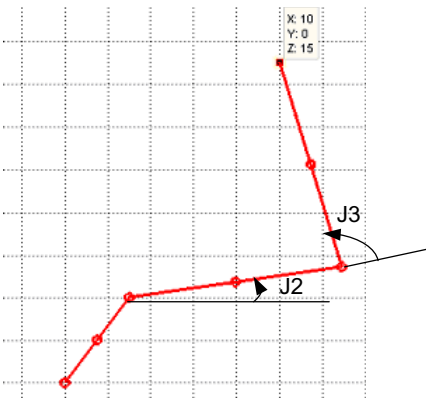
J1 = 0
J2 = 106.84
J3 = -98.63

反向左臂镜像



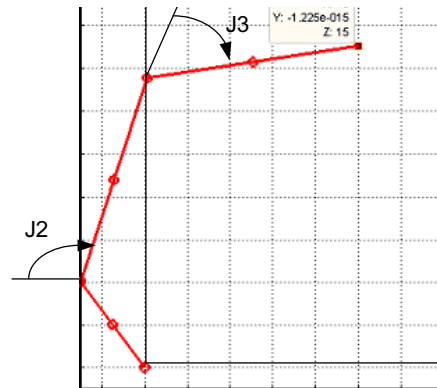
J1 = 180
J2 = 171.39
J3 = -63.26

反向右臂



J1 = 0
J2 = 8.22
J3 = 98.63

反向右臂镜像



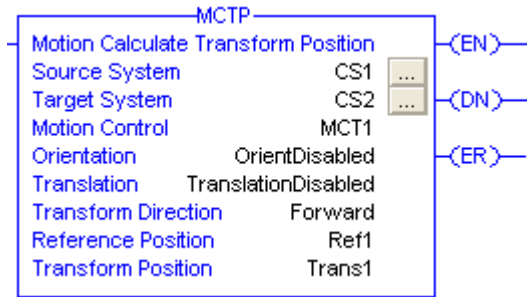
J1 = 180
J2 = 108.14
J3 = 63.26

MCTP 指令在两个坐标系之间的数据流动

下面的插图显示当执行 MCTP 指令来完成正向坐标转换和反向坐标转换时数据是如何流动的。CS1 标识符表示一个笛卡尔坐标系，包含

X1、X2 和 X3 轴，作为 MCTP 指令的源。CS2 标识符表示一个关节坐标系，包含 J1、J2 和 J3 轴，作为 MCTP 指令的目标。

通过 MCTP 指令执行运动时的数据流动 - 正向坐标转换



输入数据

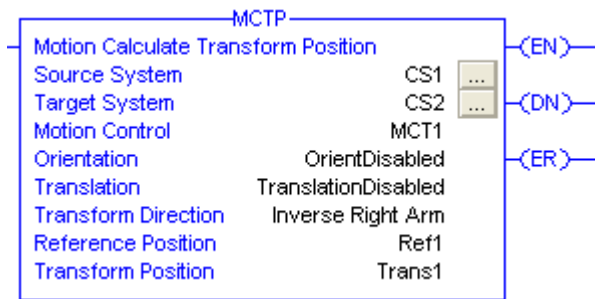
CS2: DATA	源
连接长度 (L1, L2)	坐标系对话框
基本偏移 (X1b, X2b, X3b)	坐标系对话框
末端执行器偏移 (X1e, X2e, X3e)	坐标系对话框
零角度方位 (Z1, Z2, Z3)	坐标系对话框
方位 (数组 [3])	指令面板
平移 (数组 [3])	指令面板
坐标转换方向	指令面板
参考位置	指令面板
通常为关节 - 目标	

执行的指令
MCTP

计算输出

CS1: 数据	目的地
笛卡尔位置 (X1, X2, X3)	指令面板
通常为笛卡尔	坐标转换位置

通过 MCTP 指令执行运动时的数据流动 - 反向坐标转换



输入数据

CS1: 数据	源
连接长度 (L1, L2)	坐标系对话框
基本偏移 (X1b, X2b, X3b)	坐标系对话框
末端执行器偏移 (X1e, X2e, X3e)	坐标系对话框
零角度方位 (Z1, Z2, Z3)	坐标系对话框
方位 (数组 [3])	指令面板
平移 (数组 [3])	指令面板
坐标转换方向	指令面板
参考位置 通常为笛卡尔 - 源	指令面板

执行的指令



MCTP

计算输出

CS2: 数据	目标
关节位置 (J1, J2, J3)	指令面板
通常为关节	坐标转换位置

运动插补关闭复位 (MCSR)

使用运动插补关闭复位 (MCSR) 指令可复位坐标系中的所有轴。MCSR 指令将轴从关闭状态复位成轴就绪状态。这条指令还清除任何轴故障

注意



用于指令运动控制属性的标签应仅使用一次。在其他指令中重复使用运动控制标签可能导致造成非预期的操作。这可能造成设备损坏或者人员伤害。

操作数:

梯形图

操作数	类型	格式	描述
坐标系	COORDINATE_SYSTEM	标签	将位置输入提供给凸轮输出的轴的名称。省略号将启动轴属性对话框。
运动控制	MOTION_INSTRUCTION	标签	用于访问指令状态参数的结构。

结构文本

操作数与梯形图 MCSR 指令的操作数相同。

坐标系

坐标系操作数指定了定义笛卡尔坐标系维度的运动轴系。对于这个版本，坐标系支持最大三个 (3) 主轴。只有配置为主轴（最多三个）的那些轴将包含在协调速度的计算中。



MCSR(CoordinateSystem,
MotionControl);

运动控制

以下控制位将受到 MCSR 指令的影响。

受到 MCSR 指令影响的控制位

记忆单元	描述
.EN (启用) 位 31	当梯级从假过渡到真时, 启用位被设置。当梯级从真过渡到假时, 启用位被复位。
.DN (完成) 位 29	当协调关闭复位被成功发起后, 完成位被设置。当梯级从真过渡到假时, 完成位被复位。
.ER (错误) 位 28	当协调关闭的复位未能发起时, 错误位被设置。当梯级从假过渡到真时, 错误位被复位。

这是一条转换指令:

- 在梯形图中, 当每次应执行指令时, 将梯级输入条件从清除切换到设置。
- 在结构文本中, 限制指令只对转换执行。请参阅附录 C。

算术状态标志:

不受影响

故障条件:

无

错误代码:

参见[运动控制指令错误代码 \(ERR\)](#)。

MCSR 对状态位的更改：

状态位提供一种监视运动控制指令进度的途径。有三种类型的状态位提供有关信息。分别是：轴状态位、坐标系状态位和协调运动状态位。当 MCS 指令发起后，状态位将经历以下变化。

轴状态位

位名称	影响
CoordinatedMoveStatus	无影响。

坐标系状态位

位名称	影响
ShutdownStatus	清除关闭状态位。

协调运动状态位

位名称	影响
MovePendingStatus	清空指令队列并清除状态位。
MovePendingQueueFullStatus	清空指令队列并清除状态位。

梯形图示例



结构文本

```
MCSR(Coordinated_sys,MCSR[3]);
```

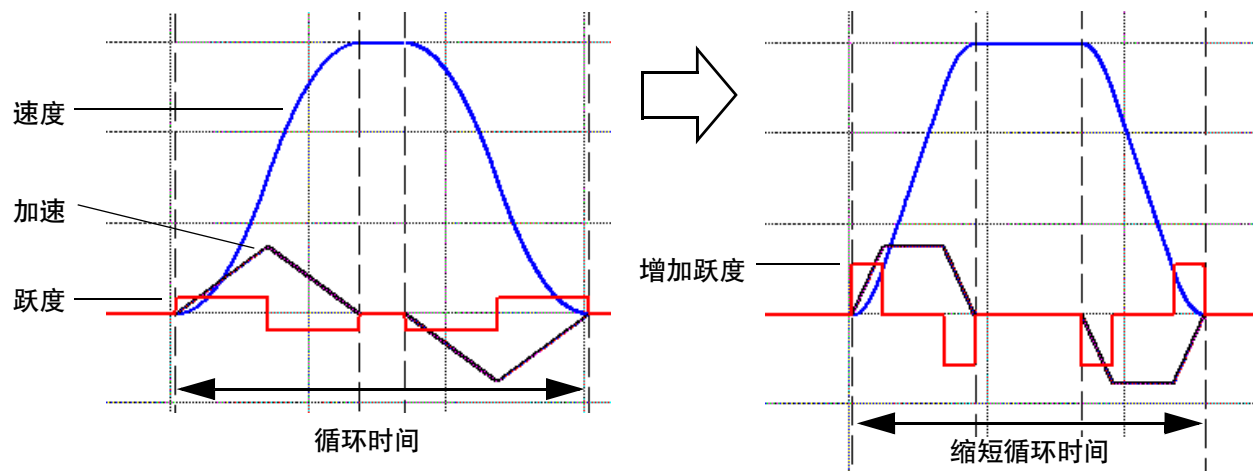
整定 S 曲线模式

简介

使用这个程序可对使用 S 曲线模式的运动在平滑度和循环时间之间进行平衡。

执行时机

当您想要缩短 S 曲线运动模式的循环时间，但又要保持模式的部分平滑度时，请执行这个程序。



要使用这个程序，您的应用必须满足以下要求：

- 控制器为版本 16 或更高。
- 以下一种指令在生成运动：
 - 轴位移 (MAM)
 - 轴速度 (MAJ)
 - 运动轴停止 (MAS)
- 指令使用 S 曲线模式





准备事宜

重要提示

在这个程序中，您增加了跃度。这将增加对设备和载荷的压力。确定您可以识别设备或负荷什么时候达到跃度限制。

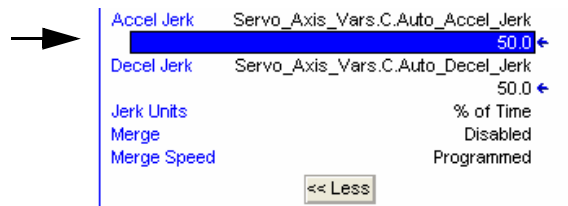
程序

1. 跃度单位是否设置为时间 %?

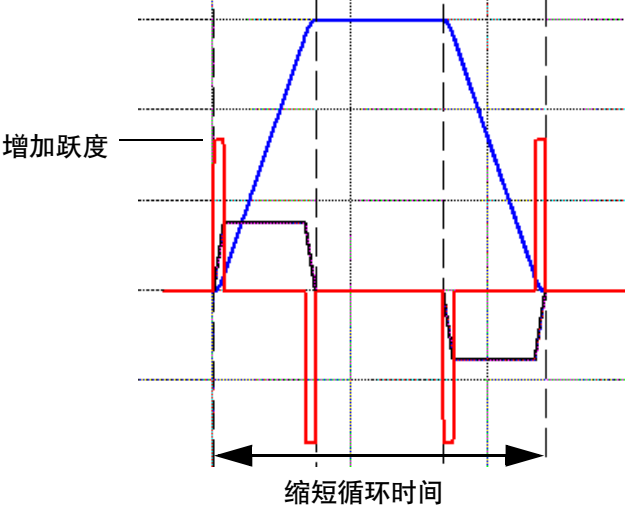
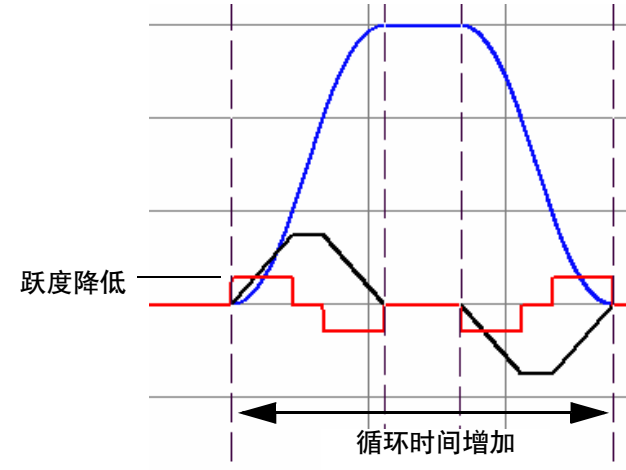
如果跃度单位是	那么
时间 % 	继续 步骤 2 。
最大值 % 	A. 将跃度单位更改为时间 %。 
单位每秒 3 	B. 继续 步骤 2 。

2. 将跃度值设置为时间 50%。

示例



3. 测试您的设备并观察其跃度。
4. 调整跃度值。

如果	那么	这会导致
跃度不太大	缩短时间 %。	
跃度太大	增加时间 %。	

5. 重复步骤 3 和 4，直到您在平滑度和循环时间之间找到需要的平滑。

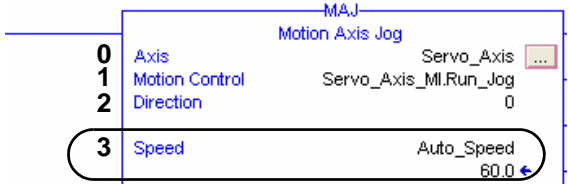
其他资源

- [编写一个速度模式程序](#) 在 [页码 24](#)
- [Troubleshoot Axis Motion](#) 在 [page 9](#)

注意：

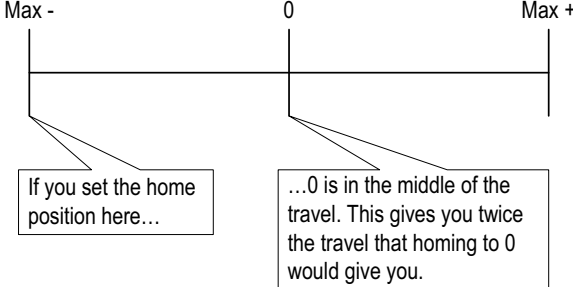
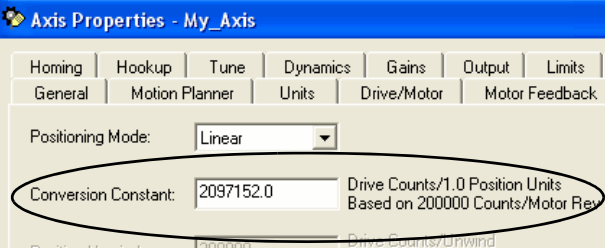
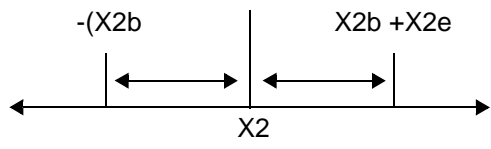
运动控制指令错误代码 (ERR)

错误	处理措施或原因	注意:
3	查找这类指令的其他实例。查看是否它的 EN 位为打开状态但是它的 DN 和 ER 位却是关闭状态 (已使能, 但是未完成或者出错)。等到它的 DN 或 ER 位打开。	<p>执行冲突</p> <p>如果同类型的指令已经使能但未完成或者出错, 则不能执行指令。</p>
4	在执行此指令之前, 先打开伺服环。	伺服启动状态错误
5	在执行此指令之前, 先关闭伺服环。	<p>伺服关闭状态错误</p> <p>对于运动协调指令, 请查看扩展错误代码 (EXERR)。它标识哪个轴导致了错误。</p> <p>示例: 如果 EXERR 为零, 请检查轴的维度是否为零。</p>
6	禁止轴的驱动。	驱动打开状态错误
7	执行运动轴关闭复位 (MASR) 指令或直接命令来复位轴。	<p>关闭状态错误</p> <p>对于运动协调指令, 请查看扩展错误代码 (EXERR)。它标识哪个轴导致了错误。</p> <p>示例: 如果 EXERR 为零, 请检查轴的维度是否为零。</p>
8	配置的轴类型不正确。	<p>错误轴类型</p> <p>对于运动协调指令, 请查看扩展错误代码 (EXERR)。它标识哪个轴导致了错误。</p> <p>示例: 如果 EXERR 为零, 请检查轴的维度是否为零。</p>
9	指令尝试执行的方向加重了当前的超行程情况。	超行程条件
10	主轴参考与从轴参考相同。	主轴冲突
11	至少一个轴未配置到物理运动模块, 或者尚未指定到某个运动组。	<p>轴未配置</p> <p>对于运动协调指令, 请查看扩展错误代码 (EXERR)。它标识哪个轴导致了错误。</p> <p>示例: 如果 EXERR 为零, 请检查轴的维度是否为零。</p>
12	发往伺服模块的消息失败。	伺服消息失败

错误	处理措施或原因	注意:
13	<p>查看扩展错误代码 (EXERR) 了解指令。它标识了超出其范围的操作数。</p> <p>示例: 假设 MAJ 指令出现 ERR = 13 和 EXERR = 3。在这种情况下, 更改速度使其位于范围以内。</p> 	<p>参数超出范围</p> <p>EXERR = 0 表示指令的第一个操作数超出其范围。</p>
14	指令无法应用整定参数, 因为在运行整定指令中出现错误。	整定过程错误
15	指令无法应用诊断参数, 因为在运行诊断测试指令中出现错误。	测试过程错误
16	等到归零过程完成。	归零过程中错误
17	指令尝试在轴上执行旋转运动, 但该轴未配置为旋转操作。	轴模式非旋转
18	轴的类型被配置为未使用。	轴类型未使用
19	运动组不在同步状态。这可能因为是丢失或错误配置伺服模块所致。	组未同步
20	轴处于故障状态。	轴为故障状态
21	组处于故障状态。	组为故障状态
22	在执行此指令之前, 先停止轴。	轴在运动中
23	指令尝试非法动态参数更改。	非法动态更改
24	将控制器离开测试模式。	无效 AC 模式操作
25	您尝试执行一个不正确的指令。	非法指令
26	凸轮系列长度非法。	非法凸轮长度
27	凸轮曲线系列长度非法。	非法凸轮曲线长度
28	您在凸轮元素中有非法的段类型。	非法凸轮类型
29	您的凸轮元素顺序非法。	非法凸轮顺序
30	您尝试执行一个正在进行计算的凸轮曲线。	凸轮曲线正在计算
31	您尝试执行的凸轮曲线系列正在使用中。	凸轮曲线正在使用
32	您尝试执行的凸轮曲线系列尚未计算。	凸轮曲线未计算
33	它尝试执行一条 MAH 指令单没有位置凸轮正在进行。	位置凸轮未使能
34	MAH 指令正尝试启动, 但有个记录已经在运行。	记录正在进行
35	控制器或凸轮输出模块不支持指定的输出凸轮、轴、输入或输出。	非法执行目标

错误	处理措施或原因	注意:
36	凸轮输出系列的大小不被支持, 或者它的一个成员的值超出范围。	<p>非法凸轮输出</p> <p>ExErr#1: 输出位小于 0 或者大于 31。</p> <p>ExErr#2: 锁存类型小于 0 或者大于 3。</p> <p>ExErr#3: 解锁存类型小于 0 或者大于 5。</p> <p>ExErr#4: 左侧或右侧位置超出凸轮范围, 锁存或解锁存被设置为“位置”或者“位置和使能”。</p> <p>ExErr#5: 持续时间小于或等于 0, 解锁存类型被设置为“持续时间”或“持续时间和使能”。</p> <p>ExErr#6: 使能类型小于 0 或大于 3, 锁存或解锁存被设置为“使能”、“位置和使能”或者“持续时间和使能”。</p> <p>ExErr#7: 启用位小于 0 或大于 31, 锁存或解锁存被设置为“使能”、“位置和使能”或者“持续时间和使能”。</p> <p>ExErr#8: 锁存被设置为“不活跃”, 解锁存类型被设置为“持续时间”或“持续时间和使能”。</p>
37	输出补偿系列的大小不被支持, 或者它的一个成员的值超出范围。	<p>非法输出补偿</p> <p>ExErr#1: 模式小于 0 或者大于 3。</p> <p>ExErr#2: 循环时间小于或等于 0, 模式被设置为“脉冲调制”或“取反和脉冲调制”。</p> <p>ExErr#3: 占空比小于 0 或大于 100, 模式被设置为“脉冲调制”或“取反和脉冲调制”。</p>
38	轴数据类型非法。它对于操作是不正确的。	<p>非法轴数据类型</p> <p>对于运动协调指令, 请查看扩展错误代码 (EXERR)。它标识哪个轴导致了错误。</p> <p>示例: 如果 EXERR 为零, 请检查轴的维度是否为零。</p>
39	您的过程中存在冲突。测试和整定不能同时运行。	过程冲突
40	在变频器被本地禁止时, 您在尝试运行 MSO 或 MAH 指令。	变频器本地禁止
41	归零配置是非法的。当归零序列不是立即时, 您使用一条绝对值归零指令。	非法归零配置
42	MASD 或 MGSD 指令已经超时, 因为它没有收到关闭状态位。通常当 MASD 或 MGSD 后跟复位指令时, 在关闭指令还没有收到关闭位时就发起该复位指令, 就会导致程序问题。	关闭状态超时
43	您尝试激活多个运动控制指令, 超过指令队列可以容纳的数量。	坐标系队列满

错误	处理措施或原因	注意:
44	您使用三个点画一条线，而无法去顶中心点经由点或者平面中心点。	圆形共线性错误
45	您指定了一个点半径或者“画一条线”中心点、经由点，而中心点半径或平面中心点、经由点无法确定。	圆形开始结束错误
46	编程的中心点距起点和终点不等距。	圆形 R1 R2 不匹配错误
47	请联系罗克韦尔自动化支持。	圆形无限解错误
48	请联系罗克韦尔自动化支持。	圆形无解错误
49	$ R < 0.01$ 。R 根本上太小而不能用于计算。	圆形小 R 错误
50	坐标系标签不关联一个运动组。	坐标系不在组内
51	您设置的终止类型为实际位置且值为 0，这个值不被支持。	无效实际容差
52	至少一个轴目前正在进行另一个坐标系的协调运动。	协调运动正在进行错误
53	解除对轴的禁止。	轴被禁止
54	1. 打开该轴的属性。 2. 在动态参数选项卡上，输入最大减速度的值。	零最大减速度 如果轴的最大减速度为零，则不能开始运动。
61	查看扩展错误代码 (EXERR) 了解指令。	连接冲突
62	取消控制此轴的坐标转换，或者当坐标转换正在工作时不使用此指令。	坐标转换正在进行 如果轴是活跃的坐标转换的一部分，您不能执行此指令。
63	取消控制此轴的坐标转换，或者等到坐标转换完成对该轴的运动。	轴在坐标转换运动中 如果某个坐标转换正在移动该轴，您不能执行此指令。
64	使用笛卡尔坐标系。	附件不被支持 您不能将这条指令用于非笛卡尔坐标系中。

错误	处理措施或原因	注意:
65	<p>轴移动过快，控制器无法存储位置。为防止出现该错误，请设置软行程限制将轴保持在位置范围以内。获得更多行程的方法之一就是使用最大负的或最大正的位置作为您的归零位置。</p> <p>示例</p> 	<p>轴位置溢出</p> <p>位置的范围取决于轴的转换常数。</p>  <ul style="list-style-type: none"> • 最大正的位置 = 2,147,483,647 / 轴的转换常数 • 最大负的位置 = -2,147,483,648 / 轴的转换常数 <p>假设您的转换常数为 2,097,152 次数 / 英寸。在这种情况下:</p> <ul style="list-style-type: none"> • 最大正的位置 = 2,147,483,647 / 2,097,152 次数 / 英寸 = 1023 英寸 • 最大负的位置 = -2,147,483,648 / 2,097,152 次数 / 英寸 = -1023 英寸 <p>对于运动协调指令，请查看扩展错误代码 (EXERR)。它标识哪个轴导致了错误。</p> <p>示例: 如果 EXERR 为零，请检查轴的维度是否为零。</p>
66	<p>确定将机器人保持在您在其中配置的准备解内。您可以在左准备或右准备解内配置机器人。</p>	<p>您正在尝试在象限边界处将关节独立或不独立的二轴机器人折回到自身上。</p>
67	<ul style="list-style-type: none"> • 将目标位置更改为机器人臂展范围以内的值。 • 如果 $X2b + X2e$ 不为零，保持在此区域以外: 	<p>无效坐标转换位置</p> <ul style="list-style-type: none"> • 您正在尝试移动到一个机器人无法够到的位置。 • MCT 在原点处尝试。 <p>为避免机器人折回自身上或者展开超过其范围，对于 Delta2D、Delta3D 和 SCARA Delta 机器人来说由固件内部计算关节限制。如果您尝试并配置一个违反这些限制的运动，则出现这个错误。</p>
68	<p>移动关节使机器人的末端不在坐标系的原点上。</p>	<p>坐标转换在原点处</p> <p>如果关节角结果产生 $X1 = 0$ 和 $X2 = 0$，您不能开始这个坐标转换。</p>

错误	处理措施或原因	注意:
69	<ul style="list-style-type: none"> 请检查关节的最大速度配置。 在坐标系的原点处，使用目标位置让机器人保持完全伸展或者折回到自身上。 以相对直的线移动通过位置 $X1 = 0$ 和 $X2 = 0$。 	<p>超出最大关节速度</p> <p>计算出的速度非常高。当机器人处于以下一种情况时会发生这种情况：</p> <ul style="list-style-type: none"> 完全伸展。 折回到自身上。 从 $X1 = 0$ 和 $X2 = 0$ 移开时使用与到达该位置不同的角度。 配置了错误的速度限制。 <p>示例： 这些运动产生此错误。</p>
70	寻找配置为旋转定位模式的源或目标。将其更改为线性定位模式。	<p>进行坐标转换的轴必须是线性的</p> <p>坐标转换只能用于线性的轴。</p>
71	等到您正在取消的坐标转换完全被取消。	坐标转换正在取消
72	检查目标位置。计算出的关节角超出 $\pm 360^\circ$ 。	超出最大关节角度
73	检查这个链中的每一条 MCT 指令是否产生无效位置。	<p>坐标系链接错误</p> <p>这条 MCT 指令是 MCT 指令链的一部分。链中的一条指令有问题。</p>
74	更改方位使角度处于 $\pm 360^\circ$ 范围内。	无效方位角
75	只能在 1756-L6x 控制器上使用这条指令。	<p>指令不被支持</p> <p>您只能在 1756-L6x 控制器上使用 MCT 或 MCTP 指令。</p>
76	<ol style="list-style-type: none"> 打开该轴的属性。 在动态参数选项卡上，输入最大减速度跃度的值。 	<p>零最大减速度跃度</p> <p>如果轴的最大减速度跃度为零，则不能开始使用 S 曲线模式的运动。</p>

错误	处理措施或原因	注意:
77	<p>您的坐标系中有多少个轴?</p> <ul style="list-style-type: none"> • 2 — 使用非镜像坐标转换方向。 • 3 — 使用非反转坐标转换方向。 	<p>坐标转换方向不被支持</p> <ol style="list-style-type: none"> 1. 您正在尝试在三轴坐标系中使用镜像方向以及非零的基本偏移 (X2b) 或执行器偏移 (X2e)。 2. 镜像方向在二轴坐标系中不被支持。 3. 您正在尝试使用二轴或三轴笛卡尔、Delta2D、Delta3D 或 SCARA Delta 目标坐标系以及除正向和反向以外的坐标转换方向。 <p>只有当下面两个条件都为真时，您才可以使用反转镜像方向:</p> <ul style="list-style-type: none"> • 您使用一个三轴坐标系。 • X2 维度的基本偏移 (X2b) 和末端执行器偏移 (X2e) 为零。
78	当运动停止指令正在进行时不要叠加运动控制指令	<p>在停止过程中不允许</p> <p>在停止过程中，您不能叠加某些运动控制指令。等待第一条指令完成后，再启动第二条指令。有关更多信息，请参阅第 398 和 399 页上的表格。</p>
79	再次将您的轴归零。	<p>内部归零序列错误</p> <p>如果您看到这个错误，请在您的应用程序中将您的轴重新归零。如果错误依然存在，请联系罗克韦尔自动化支持。</p>
80	纠正 MAOC 输出操作数或者 OB16IS 模块通讯格式。	<p>MAOC 无效输出操作数</p> <p>如果 MAOC 输出操作数引用了一个 OB16IS 规划输出模块，则当 MAOC 被发起时将执行两个附加的检查。</p> <ul style="list-style-type: none"> • 输出操作数必须引用模块输出数据标签的开头 'O.Data'。 • OB16IS 模块的通讯格式必须为默认的“按点的计划输出数据”。 <p>如果任何一个检查失败，您就会看到这个错误。</p> <p>ExErr#1: 无效的数据标签引用 - 输出操作数没有指向模块输出数据标签的 O.Data 元素。</p> <p>ExErr#2: 无效的 OB16IS 模块通讯格式 – OB16IS 通讯格式已从默认的“按点计划输出数据”更改为别的格式。</p>

错误	处理措施或原因	注意:
81	在活跃的 MGSR 指令上, 不要覆盖停止模式 = 编程值的 MASD 指令或 MGS 停止指令。	<p>部分组关闭复位。</p> <p>如果您的应用程序正在有效地执行一条 MGSR 指令, 而您尝试在其中一个受到活跃 MGSR 指令影响的轴上执行 MASD 指令或 MGS 停止指令 (停止模式 = 编程值), 则将会在 MGSR 指令上看到这个错误。</p>

当运动停止指令正在活跃时, 如果某些运动控制指令叠加, 则会得到错误。在这种情况下, 指令正在有效地停止, 第二条指令被发起覆盖活跃的指令。下面的表格列出了将会产生错误的一些叠加实例。

在这种情况下:

- 错误 # 7 = 关闭状态错误
- 错误 #61、ExErr #10 = 连接冲突, 坐标转换轴正在运动或被其他操作锁定
- 错误 #78 = 在停止过程中不允许

	活跃的停止指令								
	MGS			MGSD	MCS			MAS	
已发起第二条指令	停止模式 = 快速停止	停止模式 = 快速禁止	停止模式 = 编程值		停止类型 = 协调运动	停止类型 = 协调坐标转换	停止类型 = 全部	除了停止类型 = 全部以外的所有停止类型	停止类型 = 全部
MAAT	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78	错误 #78	错误 #78
MRAT	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78	错误 #78	错误 #78
MAHD	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78	错误 #78	错误 #78
MRHD	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78	错误 #78	错误 #78
MAH	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78	错误 #78	错误 #78
MAJ	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MAM	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MAG	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MCD	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78	错误 #78	错误 #78
MAPC	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MATC	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MDO	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MCT	错误 #78	错误 #78	错误 #78	错误 #7	错误 #61 ExErr #10	错误 #61 ExErr #10	错误 #61 ExErr #10	错误 #61 ExErr #10	错误 #61 ExErr #10
MCCD	错误 #78	错误 #78	错误 #78	错误 #7			错误 #78		错误 #78
MCLM/MCCM (合并 = 已禁止)	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #78		错误 #78
MCLM/MCCM (合并 = 已使能)	错误 #78	错误 #78	错误 #78	错误 #7		错误 #78	错误 #78		错误 #78

下面的表格列出了将会产生错误的更多叠加实例。

		活跃的停止指令						
		MGS			MGSD	MCS	MAS	MASD
已发起第二条指令	停止类型	停止模式 = 快速停止	停止模式 = 快速禁止	停止模式 = 编程值	无	停止类型 = 全部	停止类型 = 全部	无
MGS	停止模式 = 快速停止	错误 #78	错误 #78	错误 #78	错误 #7			
	停止模式 = 快速禁止	错误 #78	错误 #78	错误 #78	错误 #7			
	停止模式 = 编程值	错误 #78	错误 #78	错误 #78	错误 #7			
MGSR	无	错误 #78	错误 #78	错误 #78	错误 #7			错误 #7
MCS	停止类型 = 协调运动	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	
	停止类型 = 协调坐标转换	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	
	除了停止类型 = 全部以外的所有停止类型	错误 #78	错误 #78	错误 #78	错误 #7			
MAS	停止类型 != 全部	错误 #78	错误 #78	错误 #78	错误 #7	错误 #78	错误 #78	错误 #7
	停止类型 = 全部	错误 #78	错误 #78	错误 #78	错误 #7			错误 #7
MASR	无	错误 #78	错误 #78	错误 #78	错误 #7			错误 #7

附加错误代码信息

请参阅以下手册，了解在变频器和 / 或多轴运动控制系统上显示的错误代码的有关信息。

出版号	出版号
Kinetix 2000 多轴变频器用户手册	2093-UM001
Kinetix 6000 多轴变频器用户手册	2094-UM001
Kinetix 7000 高功率伺服变频器用户手册	2099-UM001
Ultra 3000 数字式伺服驱动器安装指南	2098-IN003
8720 高性能变频器安装指南	8720MC-IN001
1394 SERCOS 接口多轴运动控制系统安装手册	1394-IN002

注意：

运动相关的数据类型（结构）

简介

使用此附录可了解有关以下运动相关数据类型的信息：

数据类型	页码
CAM 结构	401
CAM_PROFILE 结构	402
MOTION_GROUP 结构	403
MOTION_INSTRUCTION 数据类型	404
OUTPUT_CAM 结构	405
OUTPUT_COMPENSATION 结构	407

其他资源

对于其他运动相关的数据类型，请参阅《Logix5000 控制系统用户手册》（出版号 LOGIX-UM002）中的“运动控制模块”。

CAM 结构

凸轮数据类型包括从轴和主轴点对以及内插类型。由于与特定的轴位置或时间没有关联，点值没有单位。内插类型可为每个段指定为线性或三次方。凸轮数组元素的格式如下表所示。

记忆单元	数据类型	描述	
主机	REAL	点的 x 值。	
SLAVE	REAL	点的 y 值。	
段类型	DINT	内插类型。	
		值	描述
		0	线性
		1	三次方

CAM_PROFILE 结构

CAM_PROFILE 数据类型是一个系数数组，表示可用作时间凸轮或位置凸轮指令输入的计算凸轮曲线。提供给您唯一元素是状态，如下表定义。

记忆单元	数据类型	描述	
状态	DINT	状态参数用于指示凸轮曲线数组元素已经计算出来。如果凸轮指令的执行尝试使用凸轮曲线中任何未计算的元素，则指令就会出错。	
		值	描述
		0	凸轮曲线元素尚未计算出来。
		1	凸轮曲线元素正在进行计算。
		2	凸轮曲线元素已经计算出来。
n	凸轮曲线元素已经计算出来，目前正被 (n-2) 条 MAPC 和 MATC 指令使用。		

MOTION_GROUP 结构

每个控制器有一个 MOTION_GROUP 结构。这个结构包含关于运动组的状态和配置信息。

记忆单元	数据类型	描述			
GroupStatus	DINT	组的状态位。			
		位	参数号	数据类型	描述
		InhibStatus	00	DINT	禁止状态。
		GroupSynced	01	DINT	同步状态。
		-no-tag	02	DINT	定时器事件启动。
保留	03-31				
MotionFault	DINT	组的运动故障位。			
		位	参数号	数据类型	描述
		ACAsyncConnFault	00	DINT	异步连接故障。
		ACSyncConnFault	01	DINT	同步连接故障。
		保留	02-31		
ServoFault	DINT	组的伺服模块故障位。			
		位	参数号	数据类型	描述
		POtrvIFault	00	DINT	正向超行程故障。
		NOtrvIFault	01	DINT	负向超行程故障。
		PosErrorFault	02	DINT	位置错误故障。
		EncCHALossFault	03	DINT	编码器通道 A 丢失故障。
		EncCHBLossFault	04	DINT	编码器通道 B 丢失故障。
		EncCHZLossFault	05	DINT	编码器通道 Z 丢失故障。
		EncNsFault	06	DINT	编码器噪声故障。
		DriveFault	07	DINT	变频器故障。
		保留	08-31		
		位	参数号	数据类型	描述
		SyncConnFault	00	DINT	同步连接故障。
		HardFault	01	DINT	伺服硬件故障。
		保留	02-31		
GroupFault	DINT	组的故障位。			
		位	参数号	数据类型	描述
		GroupOverlapFault	00	DINT	组任务重叠故障。
		CSTLossFault	01	DINT	控制器失去与 CST 主机的同步。
		GroupTaskLoading Fault	02	DINT	组的近似刷新周期太短，用户应用任务没有足够时间执行。
保留	03-31				

MOTION_INSTRUCTION 数据类型

您必须为所使用的每条运动控制指令定义一个运动控制标签。该标签使用 MOTION_INSTRUCTION 数据类型，储存关于该指令的状态信息。

记忆单元	数据类型	描述	
FLAGS	DINT	使用 DINT 可在一个 32 位值内访问该指令的所有状态位。	
		对于这个状态位	使用这个位编号
		EN	31
		DN	29
		ER	28
		PC	27
		IP	26
		AC	23
		DECEL	1
		ACCEL	0
EN	BOOL	启用位表示该指令被使能（梯级输入和梯级输出条件为真）。	
DN	BOOL	完成位表示所有计算和消息发送（如果有）都已完成。	
ER	BOOL	错误位指示指令被非法使用。	
PC	BOOL	过程完成位表示操作已经完成。 当指令完成执行时，.DN 位被设置。当发起的过程已经完成时，.PC 位被设置。	
IP	BOOL	正在进行位表示过程正在执行中。	
AC	BOOL	当您指令排队后，活动位可让您了解哪个指令在控制运动。当指令变为活跃时，活动位被设置。当过程完成位被设置，或者当指令被停止时，活动位被复位。	
ACCEL	BOOL	.ACCEL 位表示与点动、位移、齿轮等联系的单个指令的运行速度已经增加。	
DECEL	BOOL	.DECEL 位表示与点动、位移、齿轮等联系的单个指令的运行速度已经降低。	
ERR	INT	错误值包含于运动功能相关的错误代码。 参见 运动控制指令错误代码 (ERR) （在 页码 391 上）。	
STATUS	SINT	与运动功能相关的任何消息的状态。	
		消息状态	描述
		0x0	消息成功。
		0x1	模块正在处理另一条消息。
		0x2	模块正在等待前一条消息的响应。
		0x3	消息的响应失败。
		0x4	模块未准备好发送消息。

记忆单元	数据类型	描述
STATE	SINT	执行状态值跟踪某个功能的执行状态。很多运动功能都有多个步序，这个值将跟踪这些步序。当控制为运动控制指令设置 EN 位时，执行状态始终被设置为 0。其他执行状态取决于运动控制指令。
SEGMENT	DINT	段是一个点到下一个点（但不包括这个点）的距离。在凸轮执行时，SEGMENT 通过段号提供相对位置。
EXERR	SINT	扩展错误代码 – 用于得到关于错误的更多信息。

OUTPUT_CAM 结构

OUTPUT_CAM 数据类型是一个数组，定义了每个凸轮输出元素的具体参数。OUTPUT_CAM 包含以下成员。

记忆单元	数据类型	描述	
OutputBit	DINT	您必须在 0 到 31 的范围内选择输出位。小于 0 或大于 31 的选择将会导致非法凸轮输出错误，而且凸轮元素将不被考虑。	
LatchType	DINT	锁存类型决定着如何设置相应的输出位。值小于 0 或大于 3 将导致非法凸轮输出错误，并使用锁存类型为不活跃。	
		值	描述
		0 = 不活跃	输出位不被更改。
		1 = 位置	当轴进入被补偿的凸轮范围时，输出位被设置。
		2 = 使能	当启用位变为活跃时，输出位被设置。
UnlatchType	DINT	解锁存类型决定着如何复位输出位。选择值小于 0 或大于 5 将导致非法凸轮输出错误，并使用解锁存类型为不活跃。	
		值	描述
		0 = 不活跃	输出位不被更改。
		1 = 位置	当轴离开被补偿的凸轮范围时，输出位被复位。
		2 = 持续时间	当持续时间结束时，输出位被复位。
		3 = 使能	当启用位变为不活跃时，输出位被复位。
Left	REAL	左凸轮位置连同右凸轮位置一起定义了凸轮输出元素的凸轮范围。当锁存或解锁存类型被设置为位置或位置和使能并且启用位活跃时，左右凸轮位置指定了输出位的锁存和解锁存位置。如果左位置小于凸轮起始位置或者大于凸轮终止位置，则返回非法凸轮输出错误，该凸轮元素将不被考虑。	

记忆单元	数据类型	描述	
Right	REAL	右凸轮位置连同左凸轮位置一起定义了凸轮输出元素的凸轮范围。当锁存或解锁存类型被设置为 位置 或 位置和使能 并且启用位活跃时，左右凸轮位置指定了输出位的锁存和解锁存位置。如果右位置小于凸轮起始位置或者大于凸轮终止位置，则返回非法凸轮输出错误，该凸轮元素将不被考虑。	
持续时间	REAL	当锁存或解锁存类型被设置为 持续时间 或 持续时间和使能 并且启用位活跃时，持续时间指定了锁存和解锁存之间的时间（单位是秒）。值小于或等于 0 将导致非法凸轮输出错误，该凸轮元素将不被考虑。	
EnableType	DINT	当锁存或解锁存类型为 使能 、 位置和使能 或者 持续时间和使能 时，它定义了指定启用位的源和极性。值小于 0 或大于 31 将导致非法凸轮输出错误，该凸轮元素将不被考虑。	
		值	描述
		0 = 输入	启用位在输入参数中。
		1 = 反转输入	启用位在输入参数中，为低有效。
		2 = 输出	启用位在输出参数中。
3 = 反转输出	启用位在输出参数中，为低有效。		
EnableBit	DINT	当锁存或解锁存类型为 使能 、 位置和使能 或者 持续时间和使能 时，选择的启用位的值必须在 0 到 31 之间。值小于 0 或大于 31 将导致非法凸轮输出错误，该凸轮元素将不被考虑。	

OUTPUT_COMPENSATION 结构

OUTPUT_COMPENSATION 数据类型通过指定每个执行机构的特征，来定义每个输出位的具体参数。OUTPUT_COMPENSATION 包含以下成员：

记忆单元	数据类型	描述										
偏移量	REAL	偏移为锁存和解锁存操作提供位置补偿。										
LatchDelay	REAL	锁存延迟采用秒为单位，为锁存操作提供时间补偿。										
UnlatchDelay	REAL	解锁存延迟采用秒为单位，为解锁存操作提供时间补偿。										
Mode	DINT	<p>模式确定了输出位的行为。提供下面四个模式选项：值小于 0 或大于 3 将导致非法输出补偿错误。</p> <table border="1"> <thead> <tr> <th>值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0 = 正常</td> <td>输出位对于锁存操作被设置，对于解锁存操作被复位。</td> </tr> <tr> <td>1 = 反向</td> <td>输出位对于锁存操作被复位，对于解锁存操作被设置。</td> </tr> <tr> <td>2 = 脉冲调制</td> <td>输出位对于锁存操作以及脉冲的占有状态时被设置，对于解锁存操作以及脉冲的空置状态时被复位。</td> </tr> <tr> <td>3 = 反转和脉冲调制</td> <td>输出位对于锁存操作以及脉冲的占有状态时被复位，对于解锁存操作以及脉冲的空置状态时被设置。</td> </tr> </tbody> </table>	值	描述	0 = 正常	输出位对于锁存操作被设置，对于解锁存操作被复位。	1 = 反向	输出位对于锁存操作被复位，对于解锁存操作被设置。	2 = 脉冲调制	输出位对于锁存操作以及脉冲的占有状态时被设置，对于解锁存操作以及脉冲的空置状态时被复位。	3 = 反转和脉冲调制	输出位对于锁存操作以及脉冲的占有状态时被复位，对于解锁存操作以及脉冲的空置状态时被设置。
值	描述											
0 = 正常	输出位对于锁存操作被设置，对于解锁存操作被复位。											
1 = 反向	输出位对于锁存操作被复位，对于解锁存操作被设置。											
2 = 脉冲调制	输出位对于锁存操作以及脉冲的占有状态时被设置，对于解锁存操作以及脉冲的空置状态时被复位。											
3 = 反转和脉冲调制	输出位对于锁存操作以及脉冲的占有状态时被复位，对于解锁存操作以及脉冲的空置状态时被设置。											
CycleTime	REAL	脉冲时间，单位是秒。如果模式为 脉冲调制 或 反转和脉冲调制 ，若循环时间小于或等于 0，则产生非法输出补偿错误。										
DutyCycle	REAL	脉冲被打开（占有）时的循环时间的百分比。值 50 代表 50% 占有状态。值小于 0 或大于 100 将导致非法输出补偿错误。										

注意：

结构文本编程

简介

本附录介绍对于结构文本编程所独有的问题。请查阅本附录中的信息，确定您已了解结构文本编程将会如何执行。

欲了解相关信息	参见页码
结构文本语法	409
赋值	411
表达式	413
指令	420
构造	421
注释	437

结构文本语法

结构文本是一种文本型编程语言，它使用语句来定义要执行的内容。

- 结构文本不区分大小写。
- 使用制表符和回车符（分割行）可让您的结构文本更加容易阅读。它们对结构文本的执行没有影响。

结构文本不区分大小写。结构文本可包含以下组件：

项	定义	示例：
赋值	使用赋值语句来将值指定给标签。	标签 := 表达式；
(请参见页码 411)	:= 运算符就是赋值运算符。	
	使用分号 “;” 终止赋值。	

项	定义	示例:
表达式 (请参见页码 413)	<p>表达式是完整赋值或构造语句的一部分。表达式求值为某个数字（数值表达式）或者真假状态（布尔表达式）。</p> <p>表达式包括:</p> <p>标签 存储数据的带名称内存区域（BOOL、SINT、INT、DINT、REAL、字符串）。</p> <p>立即 常数值。</p> <p>运算符 指定表达式内部运算的符号或记忆单元。</p> <p>函数 函数在执行后将产生一个值。使用圆括号将函数的操作数括起来。</p> <p>即使语法类似，函数与指令也不相同，区别在于函数只能在表达式中使用。指令不能用于表达式中。</p>	<p>value1</p> <p>4</p> <p>tag1 + tag2</p> <p>tag1 >= value1</p> <p>function(tag1)</p>
指令 (请参见页码 420)	<p>指令就是单独的语句。</p> <p>指令使用圆括号将其操作数括起来。</p> <p>根据指令的类型，可能会有零个、一个或多个操作数。</p> <p>当执行时，指令产生数据结构中的一个或多个值。</p> <p>使用分号 “;” 终止指令。</p> <p>即使语法类似，指令与函数也不相同，区别在于指令不能在表达式中。函数只能用于表达式中。</p>	<p><i>instruction()</i>;</p> <p><i>instruction(operand)</i>;</p> <p><i>instruction(operand1, operand2, operand3)</i>;</p>
构造 (请参见页码 421)	<p>用于触发结构文本 代码的条件语句（例如其他语句）。</p> <p>使用分号 “;” 终止构造。</p>	<p>IF...THEN</p> <p>CASE</p> <p>FOR...DO</p> <p>WHILE...DO</p> <p>REPEAT...UNTIL</p> <p>EXIT</p>
注释 (请参见页码 437)	<p>解释或澄清某个结构文本部分的文字。</p> <ul style="list-style-type: none"> 使用注释可更容易地解释结构文本。 注释对结构文本的执行没有影响。 注释可以出现在结构文本中的任何位置。 	<p>// 注释</p> <p>(* 注释开始 ... 注释结束 *)</p> <p>/* 注释开始 ... 注释结束 */</p>

赋值

使用赋值可更改标签内存储的值。赋值的语法如下：

标签 := 表达式；

其中：

组件	描述
标签	表示正在获得新值的标签 标签必须为 BOOL、SINT、INT、DINT 或 REAL
:=	为赋值符号
表达式	表示将新值指定给标签 如果标签是这种数据类型 使用这种类型的表达式
	BOOL 布尔表达式
	SINT 数值表达式
	INT
	DINT
	REAL
；	结束赋值

标签保持所赋的值，直到另一个赋值更改该值为止。

表达式可能很简单，例如一个立即值或另一个标签名称，也可能很复杂，包含多个运算符和 / 或函数。请参阅[表达式](#)（在[页码 413](#)上）了解详情。

TIP

I/O 模块数据与逻辑的执行进行异步更新。如果您在逻辑中多次引用某个输入，则不同引用之间的输入可能会改变状态。如果您需要对每次引用的输入具有相同的状态，请缓冲输入值并引用该缓冲标签。

有关更多信息，请参阅《Logix5000 控制器通用程序编程手册》，出版号 1756-PM001。

请参阅索引了解更多信息。

指定非保持赋值

非保持赋值与上面所述的常规赋值不同，差别在于每次当控制器出现下面的情况时，非保持赋值将复位为零。

- 输入运行模式
- 离开 SFC 的步序，如果您将 SFC 配置为 *自动复位* (只有当您将赋值嵌入该步序的操作，或者使用该操作通过 JSR 指令来调用结构文本例程时，这才适用)。

非保持赋值的语法如下：

标签 [:=] 表达式；

其中：

组件	描述
标签	表示正在获得新值的标签 标签必须为 BOOL、SINT、INT、DINT 或 REAL
[:=]	为非保持赋值符号
表达式	表示将新值指定给标签 如果标签是这种数据类型 使用这种类型的表达式
	BOOL 布尔表达式
	SINT 数值表达式
	INT
	DINT
	REAL
;	结束赋值

将 ASCII 字符赋值给字符串

使用赋值运算符可将 ASCII 字符指定给某个字符串标签的 DATA 成员的某个元素。要赋值字符，请指定字符的值或者指定标签名称、DATA 成员以及字符元素。举例如下：

这样可以	这样不可以。
<code>string1.DATA[0]:= 65;</code>	<code>string1.DATA[0] := A;</code>
<code>string1.DATA[0]:= string2.DATA[0];</code>	<code>string1 := string2;</code>

要添加或插入字符串到字符串标签中，请使用下面的 ASCII 字符串指令：

要	使用这个指令
将字符添加到字符串的结尾	CONCAT
将字符插入字符串	INSERT

表达式

表达式上一个标签名称、等式或比较。要编写表达式，请使用下面任一元素：

- 存储值的标签名称（变量）
- 您直接输入到表达式的数字（立即值）
- 函数，例如：ABS、TRUNC
- 运算符，例如：+、-、<、>、And、Or

当您编写表达式时，请遵照下面的一般规则：

- 使用任何大写和小写字母组合。例如，“AND”的这三种变化都是可以接受的：AND、And、and。
- 对于更加复杂的要求，请使用圆括号将表达式内的表达式分组。这样可使整个表达式更容易跃度，保证表达按照所希望的顺序执行。请参阅“[确定执行的顺序](#)”（在[页码 419](#)上）。

在结构文本中，您将使用两种类型的表达式：

布尔表达式：生成布尔值 1(真)或 0(假)的表达式。

- 布尔表达式使用布尔标签、关系运算符和逻辑运算符来比较值或者检查条件是真还是假。例如，`tag1>65`。
- 简单的布尔表达式可以就是一个布尔标签。
- 通常，您使用布尔表达式来设定其他逻辑执行的条件。

数值表达式：计算整数或浮点值的表达式。

- 数值表达式使用算术运算符、算术函数和按位运算符。例如，`tag1+5`。
- 通常，您会将数值表达式嵌套在布尔表达式内部。例如，`(tag1+5)>65`。

使用下表来为您的表达式选择运算符：

如果您需要	那么
计算算术值	“ 使用算术运算符和算术函数 ”，在 页码 415 上。
比较两个值或字符串	“ 使用关系运算符 ”，在 页码 416 上。
检查条件是真还是假	“ 使用逻辑运算符 ”，在 页码 418 上。
比较值内的位	“ 使用按位运算符 ”，在 页码 419 上。

使用算术运算符和算术函数

您可以在算术表达式中组合多个运算符和函数。

算术运算符会计算新值。

要	使用这个运算符	最佳数据类型
相加	+	DINT、REAL
减法 / 取负	-	DINT、REAL
乘法	*	DINT、REAL
指数 (X 的 Y 次方)	**	DINT、REAL
相除	/	DINT、REAL
模数除	MOD	DINT、REAL

算术函数执行数学运算。为函数指定常数、非布尔标签或表达式。

要获得	使用这个函数	最佳数据类型
绝对值	ABS (numeric_expression)	DINT、REAL
反余弦	ACOS (numeric_expression)	REAL
反正弦	ASIN (numeric_expression)	REAL
反正切	ATAN (numeric_expression)	REAL
余弦	COS (numeric_expression)	REAL
弧度转换成度	DEG (numeric_expression)	DINT、REAL
自然对数	LN (numeric_expression)	REAL
以 10 为底的对数	LOG (numeric_expression)	REAL
度转换成弧度	RAD (numeric_expression)	DINT、REAL
正弦	SIN (numeric_expression)	REAL
平方根	SQRT (numeric_expression)	DINT、REAL
正切	TAN (numeric_expression)	REAL
截断	TRUNC (numeric_expression)	DINT、REAL

举例如下：

使用这种格式	示例	
	对于这种情况	您将编写
value1 运算符 value2	如果 gain_4 和 gain_4_adj 是 DINT 标签，而且您的指定内容为：“将 15 加上 gain_4，将结果储存在 gain_4_adj 中。”	gain_4_adj := gain_4+15;
运算符 value1	如果 alarm 和 high_alarm 是 DINT 标签，而且您的指定内容为：“对 high_alarm 取反，将结果储存在 alarm 中。”	alarm:= -high_alarm;
function(numeric_expression)	如果 overtravel 和 overtravel_POS 是 DINT 标签，而且您的指定内容为：“计算 overtravel 的绝对值，将结果储存在 overtravel_POS 中。”	overtravel_POS := ABS(overtravel);
value1 运算符 (function((value2+value3)/2))	如果 adjustment 和 position 是 DINT 标签，sensor1 和 sensor2 是 REAL 标签，而且您的指定内容为：“找出 sensor1 和 sensor2 平均的绝对值，加上 adjustment，将结果储存在 position 中。”	position := adjustment + ABS((sensor1 + sensor2)/2);

使用关系运算符

关系运算符比较两个值或字符串，并提供结果为真或假。关系运算的结果是布尔值：

如果比较是	结果是
真	1
假	0

使用这些关系运算符：

对于这种比较	使用这个运算符	最佳数据类型
等于	=	DINT、REAL、字符串
小于	<	DINT、REAL、字符串
小于或等于	<=	DINT、REAL、字符串
大于	>	DINT、REAL、字符串
大于或等于	>=	DINT、REAL、字符串
不等于	<>	DINT、REAL、字符串

举例如下：

使用这种格式	示例	
	对于这种情况	您将编写
value1 运算符 value2	如果 temp 是 DINT 标签，而且您的指定内容为：“如果 temp 小于 100Y，那么…”	IF temp<100 THEN...
stringtag1 运算符 stringtag2	如果 bar_code 和 dest 是字符串标签，而且您的指定内容为：“如果 bar_code 等于 dest，那么…”	IF bar_code=dest THEN...
char1 运算符 char2 要将 ASCII 字符直接输入表达式，请输入字符的十进制值。	如果 bar_code 是字符串标签，而且您的指定内容为：“如果 bar_code.DATA[0] 等于 ' A' ，那么…”	IF bar_code.DATA[0]=65 THEN...
bool_tag := bool_expressions	如果 count 和 length 是 DINT 标签，done 是 BOOL 标签，而且您的指定内容为：“如果 count 大于或等于 length，您将 done 计数。”	done := (count >= length);

字符串如何取值

ASCII 字符的十六进制值确定了一个字符串是否小于或大于另一个字符串。

- 当两个字符串在电话目录中排序时，字符串的顺序决定哪个更大。

ASCII 字符	十六进制编码
1ab	\$31\$61\$62
1b	\$31\$62
A	\$41
AB	\$41\$42
B	\$42
a	\$61
ab	\$61\$62

↑ 更小
↓ 更大

— AB < B
— a > B

- 字符串的字符如果相同，则字符串相等。
- 字符为区分大小写。大写的 “A” (\$41) 不等于小写的 “a” (\$61)。

使用逻辑运算符

逻辑运算符让您检查多个条件是真还是假。逻辑运算的结果是布尔值：

如果比较是	结果是
真	1
假	0

使用这些逻辑运算符：

有关	使用这个运算符	数据类型
逻辑与	&、AND	BOOL
逻辑或	或	BOOL
逻辑异或	XOR	BOOL
逻辑取反	NOT	BOOL

举例如下：

使用这种格式	示例	
	对于这种情况	您将编写
BOOLtag	如果 photoeye 是 BOOL 标签，而且您的指定内容为：“如果 photoeye_1 打开，那么…”	IF photoeye THEN...
NOT BOOLtag	如果 photoeye 是 BOOL 标签，而且您的指定内容为：“如果 photoeye 关闭，那么…”	IF NOT photoeye THEN...
expression1 & expression2	如果 photoeye 是 BOOL 标签，temp 是 DINT 标签，而且您的指定内容为：“如果 photoeye 打开而且 temp 小于 100Y，那么…”	IF photoeye & (temp<100) THEN...
expression1 OR expression2	如果 photoeye 是 BOOL 标签，temp 是 DINT 标签，而且您的指定内容为：“如果 photoeye 打开或者 temp 小于 100Y，那么…”	IF photoeye OR (temp<100) THEN...
expression1 XOR expression2	如果 photoeye1 和 photoeye2 是 BOOL 标签，而且您的指定内容为：“如果： <ul style="list-style-type: none"> • photoeye1 打开同时 photoeye2 关闭，或者 • photoeye1 关闭同时 photoeye2 打开， 那么…”	IF photoeye1 XOR photoeye2 THEN...
BOOLtag := expression1 & expression2	如果 photoeye1 和 photoeye2 是 BOOL 标签，open 是 BOOL 标签，而且您的指定内容为：“如果 photoeye1 和 photoeye2 同时打开，将 open 设置为真”。	open := photoeye1 & photoeye2;

使用按位运算符

按位运算符基于两个 值来处理某个值内的位。

有关	使用这个运算符	最佳数据类型
按位与	&、AND	DINT
按位或	或	DINT
按位异或	XOR	DINT
按位求补	NOT	DINT

举例如下：

使用这种格式	示例	
	对于这种情况	您将编写
value1 运算符 value2	如果 input1、input2 和 result1 是 DINT 标签，而且您的指定内容为：“计算 input1 和 input2 的按位结果，将结果储存在 result1 中。”	result1 := input1 AND input2;

确定执行的顺序

您写到表达式中的运算符按照规定的顺序执行，而不一定是从左到右。

- 相等顺序的运算为从左到右执行。
- 如果表达式包含多个运算符或函数，请将条件用圆括号“()”进行分组。这样保证执行顺序正确，并使表达式的跃度更加容易。

顺序	操作
1.	()
2.	函数 (...)
3.	**
4.	-(求反)
5.	NOT
6.	*, /, MOD
7.	+, - (减法)
8.	<, <=, >, >=
9.	=, <>
10.	&、AND
11.	XOR
12.	OR

指令

结构文本语句也可以是指令。请参阅本手册开头处的定位器表格，了解可用于结构文本的指令列表。结构文本指令在每次被扫描时执行。位于构造内的结构文本指令在每次构造条件为真时执行。如果构造的条件为假，则构造内部的语句不被扫描。没有梯级条件或状态转换来触发执行。

这与使用 `EnableIn` 来触发执行的功能块指令有所不同。如果 `EnableIn` 被始终设置，结构文本指令将会执行。

这与使用梯级输入条件来触发执行的梯形图指令也有所不同。某些梯形图指令只在梯级输入条件从假切换到真时执行。这些是过渡性梯形图指令。在结构文本中，指令将在每次被扫描到时执行，除非您预先设置结构文本指令的执行条件。

例如，`ABL` 指令是梯形图中的一条转换指令。在这个示例中，`ABL` 指令仅在 `tag_xic` 从被清除过渡到被设置时执行。`ABL` 指令在当 `tag_xic` 保持设置状态或者 `tag_xic` 被清除时不会执行。



在结构文本中，如果您编写下面的示例：

```
IF tag_xic THEN ABL(0,serial_control);
```

```
END_IF;
```

`ABL` 指令将在 `tag_xic` 被设置的每次扫描时执行，而不仅仅当 `tag_xic` 从被清除过渡到被设置时。

如果您希望 ABL 指令仅在 tag_xic 从被清除过渡到被设置时执行，则必须为结构文本指令设定条件。使用单次触发来触发执行。

```
osri_1.InputBit := tag_xic;
OSRI(osri_1);

IF (osri_1.OutputBit) THEN
    ABL(0,serial_control);
END_IF;
```

构造

构造可以单独编程，也可以嵌套于其他构造内部。structs.

如果您需要	使用这种构造	在以下语言中可用	参见页码
如果或者当特定条件 发生时做某事	IF...THEN	结构文本	422
根据某个数字值来选择要做的内容	CASE...OF	结构文本	425
做某个事情特定的次数，然后再做别的事情	FOR...DO	结构文本	428
保持一直做某事，只要某些条件为真	WHILE...DO	结构文本	431
保持一直做某事，直到某个条件为真	REPEAT...UNTIL	结构文本	434

部分关键字为预留为将来使用

这些构造不可用：

- GOTO
- REPEAT

RSLogix 5000 软件将不会让您使用它们。

IF...THEN

使用 IF...THEN 在如果或当特定条件发生时做某事。

操作数:



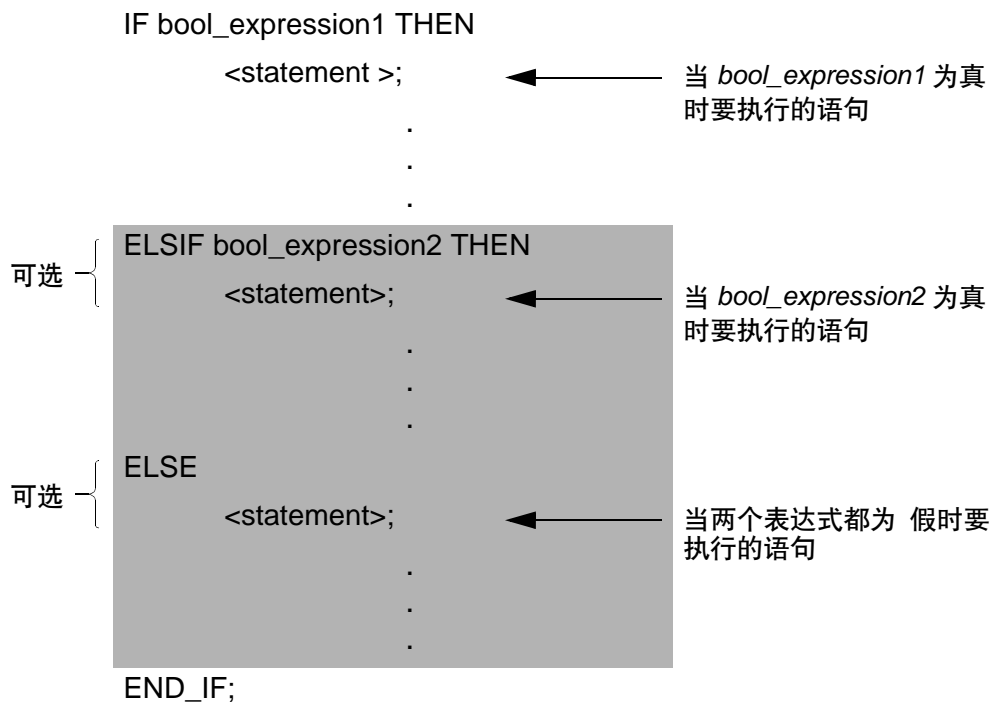
```
IF bool_expression THEN
<statement>;
END_IF;
```

结构文本

操作数	类型	格式	输入
bool_表达式	BOOL	标签表达式	对布尔值（布尔表达式）求值的布尔标签或表达式

说明:

语法如下:



要使用 ELSIF 或 ELSE，请遵照以下指导原则：

1. 要从几个可能的语句组中进行选择，请添加一个或多个 ELSIF 语句。
 - 每个 ELSIF 代表一个选择路径。
 - 根据需要指定多个 ELSIF 路径。
 - 控制器执行第一个为真的 IF 或 ELSIF，跳过其余的 ELSIF 以及 ELSE。
2. 要在所有的 IF 或 ELSIF 条件为假时做某事，请添加一个 ELSE 语句。

这个表格汇总了 IF、 THEN、 ELSIF 以及 ELSE 的组合。

如果您需要	而且	那么使用这种构造
如果或者当条件为真时做某事	如果条件为假时做某事	IF...THEN
	否则如果条件为假时做某事	IF...THEN...ESLE
根据输入条件从可选语句（或者语句组）中选择	如果条件为假时做某事	IF...THEN...ELSIF
	如果所有条件为假则赋值默认语句	IF...THEN...ELSIF...ELSE

算术状态标志： 不受影响

故障条件： 无

例 1： IF...THEN

如果您需要	输入这个结构文本
如果 rejects > 3, 那么 conveyor = off (0) alarm = on (1)	IF rejects > 3 THEN conveyor := 0; alarm := 1; END_IF;

例 2： IF...THEN...ELSE

如果您需要	输入这个结构文本
如果传送带方向触点 = 正向 (1), 那么 light = off 否则 light = on	IF conveyor_direction THEN light := 0; ELSE light [:=] 1; END_IF;

[:=] 告诉控制器每当控制器出现以下情况时 清除 light:

- 输入运行模式
- 离开 SFC 的步序, 如果您将 SFC 配置为自动复位 (只有当您将赋值嵌入该步序的操作, 或者使用该操作通过 JSR 指令来调用结构文本例程时, 这才适用)。

例 3: IF...THEN...ELSIF

如果您需要	输入这个结构文本
如果糖度低限开关 = 低 (开) 并且糖度高限开关 = 非高 (开), 那么 入口阀 = 打开 (开) 直到糖度高限开关 = 高 (关)	IF Sugar.Low & Sugar.High THEN Sugar.Inlet := 1; ELSIF NOT(Sugar.High) THEN Sugar.Inlet := 0; END_IF;

[:=] 告诉控制器每当控制器出现以下情况时 清除 Sugar.Inlet:

- 输入运行模式
- 离开 SFC 的步序, 如果您将 SFC 配置为自动复位 (只有当您赋值嵌入该步序的操作, 或者使用该操作通过 JSR 指令来调用结构文本例程时, 这才适用)。

例 4: IF...THEN...ELSIF...ELSE

如果您需要	输入这个结构文本
如果箱体温度 > 100 那么泵 = 慢	IF tank.temp > 200 THEN pump.fast :=1; pump.slow :=0; pump.off :=0;
如果箱体温度 > 200 那么泵 = 快	ELSIF tank.temp > 100 THEN pump.fast :=0; pump.slow :=1; pump.off :=0;
否则泵 = 关闭	ELSE pump.fast :=0; pump.slow :=0; pump.off :=1;
	END_IF;

CASE...OF

使用 CASE 可根据某个数字值来选择要做的内容。

操作数:



CASE *numeric_expression* OF

selector1: *statement*;

selectorN: *statement*;

ELSE

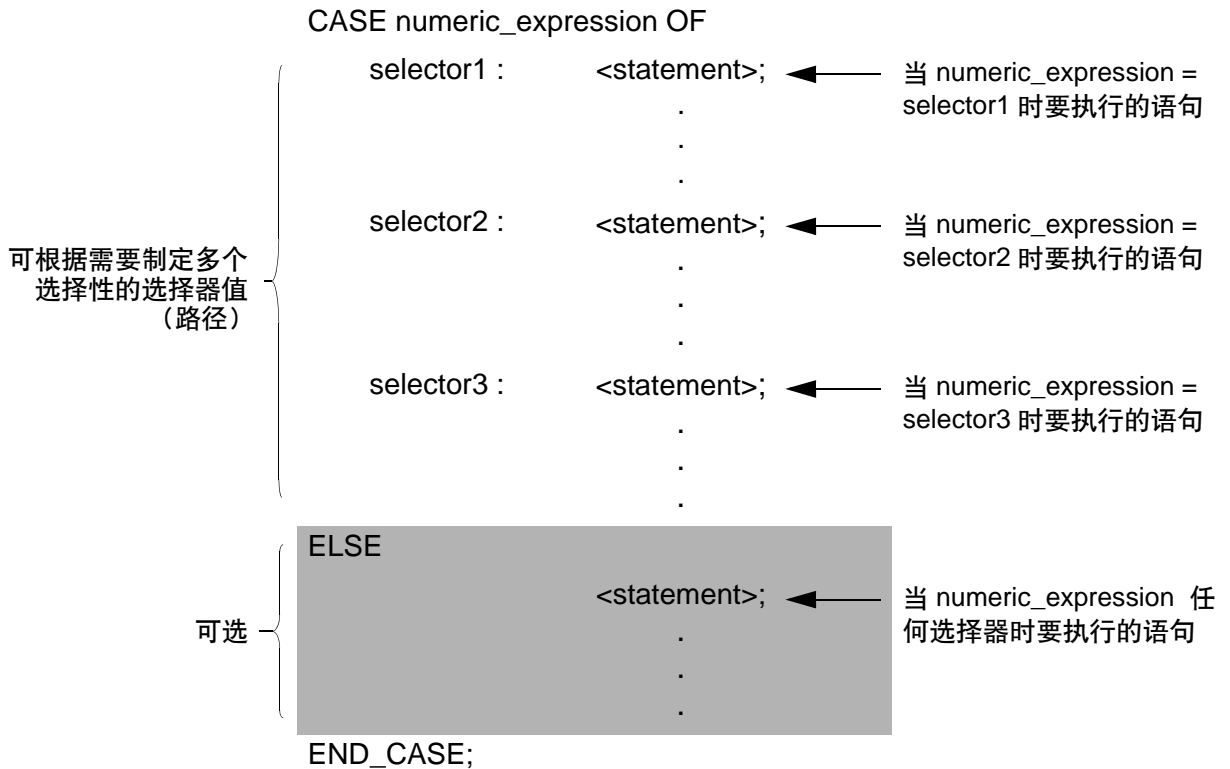
结构文本

操作数	类型	格式	输入
numeric_表达式	SINT INT DINT REAL	标签 表达式	为某个数字（数值表达式）取值的标签或表达式
选择器	SINT INT DINT REAL	立即	与 numeric_expression 类型相同

重要提示

如果您使用 REAL 值，则为选择器使用一个值范围，因为 REAL 值非常可能位于一个值范围内，而不是准确的某个特定值。

说明： 语法如下：



请参阅下页上的表格了解有效的选择器值。

输入选择器值的语法如下：

当选择器为	输入
一个值	value: statement
多个分离的值	value1, value2, valueN : <statement> 使用逗号 (,) 来分隔各个值。
值的范围	value1...valueN : <statement> 使用两个点号 (..) 来识别范围。
分离的值加上值的范围	valuea, valueb, value1...valueN : <statement>

CASE 构造非常类似于 C 或 C++ 编程语言中的开关语句 但是，使用 CASE 语句，控制器仅执行与第一个匹配的选择器值相关的语句。在该选择器的语句以后，执行始终会中断，并转到 END_CASE 语句。

算术状态标志： 不受影响

故障条件： 无

示例

如果您需要	输入这个结构文本
如果配方编号 = 1, 那么	CASE recipe_number OF
成分 A 出口 1 = 打开 (1)	1: Ingredient_A.Outlet_1 :=1;
成分 B 出口 4 = 打开 (1)	Ingredient_B.Outlet_4 :=1;
如果配方编号 = 2 或 3, 那么	2,3: Ingredient_A.Outlet_4 :=1;
成分 A 出口 4 = 打开 (1)	Ingredient_B.Outlet_2 :=1;
成分 B 出口 2 = 打开 (1)	
如果配方编号 = 4、5、6 或 7, 那么	4..7: Ingredient_A.Outlet_4 :=1;
成分 A 出口 4 = 打开 (1)	Ingredient_B.Outlet_2 :=1;
成分 B 出口 2 = 打开 (1)	
如果配方编号 = 8、11、12 或 13, 那么	8,11..13 Ingredient_A.Outlet_1 :=1;
成分 A 出口 1 = 打开 (1)	Ingredient_B.Outlet_4 :=1;
成分 B 出口 4 = 打开 (1)	
否则所有出口 = 关闭 (0)	ELSE
	Ingredient_A.Outlet_1 [:=]0;
	Ingredient_A.Outlet_4 [:=]0;
	Ingredient_B.Outlet_2 [:=]0;
	Ingredient_B.Outlet_4 [:=]0;
	END_CASE;

[:=] 告诉控制器每当控制器出现以下情况时 同时还清除出口标签:

- 输入运行模式
- 离开 SFC 的步序, 如果您将 SFC 配置为自动复位 (只有当您将赋值嵌入该步序的操作, 或者使用该操作通过 JSR 指令来调用结构文本例程时, 这才适用)。

FOR...DO

使用 FOR...DO 循环做某个事情特定的次数，然后再做别的事情。

操作数：



```
FOR count:= initial_value TO
final_value BY increment DO
```

```
<statement>;
```

```
END_FOR;
```

结构文本

操作数	类型	格式	描述
次数	SINT INT DINT	标签	储存 FOR...DO 执行过程中次数位置的标签
initial_value	SINT INT DINT	标签 表达式 立即	必须取值为数字 指定次数的初始值
final_value	SINT INT DINT	标签 表达式 立即	指定次数的最终值，这决定了何时退出循环
增量	SINT INT DINT	标签 表达式 立即	(可选) 每次通过循环时递增次数的量 如果您不指定增量，则次数按 1 递增。

重要提示 确定不要在一次扫描中在循环内部迭代过多次数。

- 控制器不会执行例程中的任何其他语句，直到它完成循环为止。
- 如果完成循环所需的时间大于看门狗对任务的定时器，则出现重要故障。
- 请考虑使用其他构造，例如 IF...THEN。

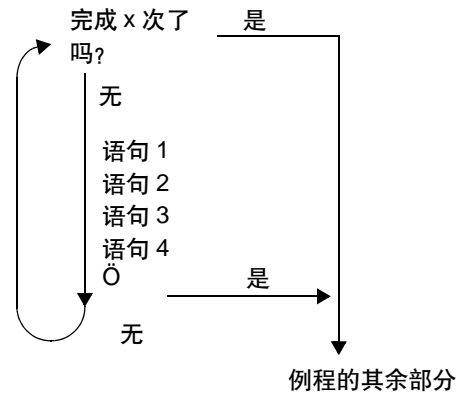
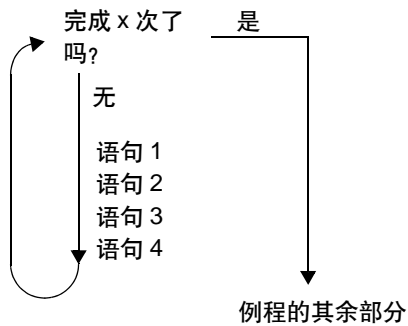
说明： 语法如下：

```
FOR count := initial_value
TO final_value
可选 { BY increment
DO
<statement>;
可选 { IF bool_expression THEN
EXIT;
END_IF;
END_FOR;
```

如果您不指定增量，则循环按 1 递增。

如果有您想要提前退出循环的条件，请使用其他语句，例如 IF...THEN 构造，来设定 EXIT 语句的条件。

这些图显示了 FOR...DO 循环的执行方式，以及 EXIT 语句如何提前离开循环。



FOR...DO 循环执行特定的次数。

要在次数达到最后的值之前终止循环，请使用 EXIT 语句。

算术状态标志： 不受影响

故障条件：

出现重要故障，如果	故障类型	故障代码
构造循环过长	6	1

例 1：

如果您需要	输入这个结构文本
清除任何布尔数组中的位 0 - 31： 1. 将下标标签初始化为 0。 2. 清除数组 [下标]。例如，当下标 = 5 时，清除数组 [5]。 3. 将下标加 1。 4. 如果下标 > 31，则重复 2 和 3。 否则，停止。	For subscript:=0 to 31 by 1 do array[subscript] := 0; End_for;

例 2：

如果您需要	输入这个结构文本
<p>用户自定义数据类型 (结构) 存储您的仓库内某件物品的相关信息:</p>	<pre>SIZE(Inventory,0,Inventory_Items); For position:=0 to Inventory_Items - 1 do If Barcode = Inventory[position].ID then Quantity := Inventory[position].Qty; Exit; End_if; End_for;</pre>
<ul style="list-style-type: none">• 该物品的条形码 ID (字符串数据类型)• 该物品的库存数量 (DINT 数据类型)	
<p>上述结构数组为您的仓库内每件不同物品包含一个元素。您希望从数组中搜索某特特定产品 (使用其条形码), 并确定其库存数量。</p>	
<p>1. 获取库存数组的大小 (物品件数), 并将结果存储在 Inventory_Items 内 (DINT 标签)。</p>	
<p>2. 将位置标签初始化为 0。</p>	
<p>3. 如果条形码匹配数组内的某件物品 ID, 那么:</p> <ul style="list-style-type: none">a. 设置数量标签 = Inventory[position].Qty。这将生成物品的库存数量。b. 停止。	
<p>条形码是一个字符串标签, 存储您正在搜索的物品的条形码。例如, 当位置 = 5 时, 将条形码与 Inventory[5].ID 进行比较。</p>	
<p>4. 将位置加 1。</p>	
<p>5. 如果位置 (Inventory_Items -1), 重复 3 和 4。由于元素编号从 0 开始, 最后一个元素要比数组内元素个数少 1。</p>	
<p>否则, 停止。</p>	

WHILE...DO

使用 WHILE...DO 循环保持一直做某事，只要某些条件为真。

操作数：



WHILE *bool_expression* DO

<*statement*>;

END_WHILE;

结构文本

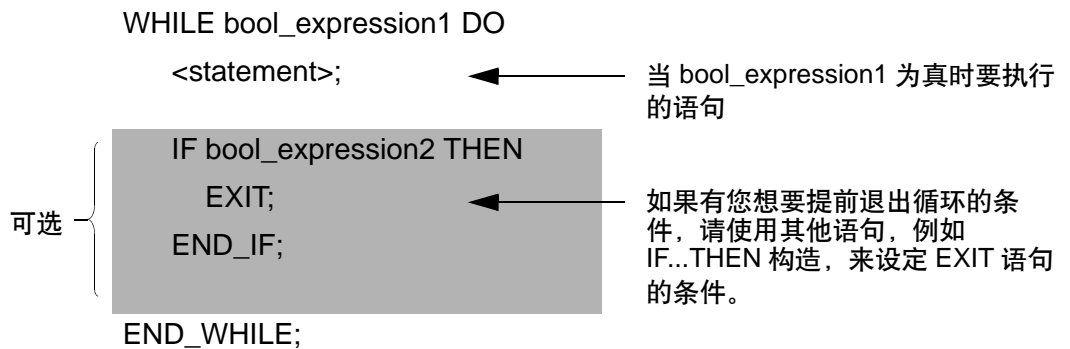
操作数	类型	格式	输入
bool_表达式	BOOL	标签表达式	对布尔值求值的布尔标签或表达式

重要提示

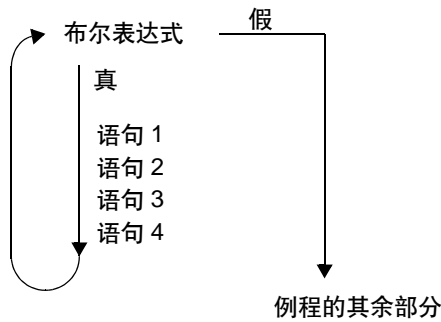
确定不要在一次扫描中在循环内部迭代过多次数。

- 控制器不会执行例程中的任何其他语句，直到它完成循环为止。
- 如果完成循环所需的时间大于看门狗对任务的定时器，则出现重要故障。
- 请考虑使用其他构造，例如 IF...THEN。

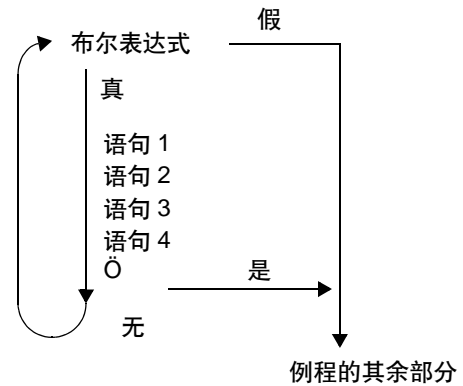
说明： 语法如下：



这些图显示了 WHILE...DO 循环的执行方式，以及 EXIT 语句如何提前离开循环。



当 bool_expression 为真时，控制器仅执行 WHILE...DO 循环内的语句。



要在条件为真之前终止循环，请使用 EXIT 语句。

算术状态标志： 不受影响

故障条件：

出现重要故障，如果	故障类型	故障代码
构造循环过长	6	1

例 1：

如果您需要	输入这个结构文本
<p>WHILE...DO 循环先评估其条件。如果条件为真，那么控制器就执行 WHILE...DO 循环内的语句。</p> <p>这不同于 REPEAT...UNTIL 循环，因为 REPEAT...UNTIL 循环执行构造中的语句，然后确定条件是否为真后再来执行这些语句。REPEAT...UNTIL 循环中的语句始终会被至少执行一次。WHILE...DO 循环中的语句可能会用不执行。</p>	<pre>pos := 0; While ((pos <= 100) & structarray[pos].value <> targetvalue)) do pos := pos + 2; String_tag.DATA[pos] := SINT_array[pos]; end_while;</pre>

例 2:

如果您需要	输入这个结构文本
<p>将 ASCII 字符从 SINT 数组移到字符串标签内。(在 SINT 数组中, 每个元素容纳一个字符。)当您到达回车符时停止。</p>	<pre>element_number := 0; SIZE(SINT_array, 0, SINT_array_size); While SINT_array[element_number] <> 13 do String_tag.DATA[element_number] := SINT_array[element_number]; element_number := element_number + 1; String_tag.LEN := element_number; If element_number = SINT_array_size then exit; end_if; end_while;</pre>
<ol style="list-style-type: none">1. 将 Element_number 初始化为 0。2. 数出 SINT_array(包含 ASCII 字符的数组)中元素的个数, 将结构存储在 SINT_array_size (DINT 标签) 内。3. 如果 SINT_array[element_number] 处的字符 = 13 (回车符的十进制值), 那么停止。4. 设置 String_tag[element_number] = SINT_array[element_number] 处的字符。5. 将 element_number 加 1。这样让控制器检查 SINT_array 中的下一个字符。6. 设置 String_tag 的长度成员 = element_number。(这记录了目前为止 String_tag 内的字符个数。)7. 如果 element_number = SINT_array_size, 那么停止。(您现在位于数组结尾, 此处不含回车符。)8. 转到 3。	

REPEAT...UNTIL

使用 REPEAT...UNTIL 循环保持一直做某事，直到条件为真为止。

操作数：



REPEAT

<statement>;

UNTIL *bool_expression*

END_REPEAT;

结构文本

操作数	类型	格式	输入
bool_表达式	BOOL	标签表达式	对布尔值（布尔表达式）求值的布尔标签或表达式

重要提示

确定不要在一次扫描中在循环内部迭代过多次数。

- 控制器不会执行例程中的任何其他语句，直到它完成循环为止。
- 如果完成循环所需的时间大于看门狗对任务的定时器，则出现重要故障。
- 请考虑使用其他构造，例如 IF...THEN。

说明： 语法如下：

```

REPEAT
  <statement>;
  IF bool_expression2 THEN
    EXIT;
  END_IF;
UNTIL bool_expression1
END_REPEAT;

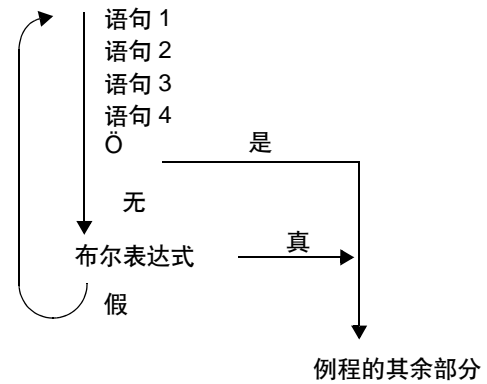
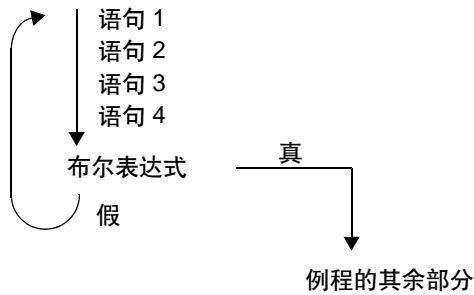
```

可选 {

← 当 *bool_expression1* 为假时要执行的语句

← 如果有您想要提前退出循环的条件，请使用其他语句，例如 IF...THEN 构造，来设定 EXIT 语句的条件。

这些图显示了 REPEAT...UNTIL 循环的执行方式，以及 EXIT 语句如何提前离开循环。



当 bool_expression 为假时，控制器仅执行 REPEAT...UNTIL 循环内的语句。

要在条件为假之前终止循环，请使用 EXIT 语句。

算术状态标志： 不受影响

故障条件：

出现重要故障，如果	故障类型	故障代码
构造循环过长	6	1

例 1：

如果您需要	输入这个结构文本
REPEAT...UNTIL 循环执行构造中的语句，然后确定条件是否为真后再来执行这些语句。	pos := -1; REPEAT
这不同于 WHILE...DO 循环，因为 WHILE...DO 循环会先评估其条件。如果条件为真，那么控制器就执行 WHILE...DO 循环内的语句。REPEAT...UNTIL 循环中的语句始终会被至少执行一次。WHILE...DO 循环中的语句可能会用不执行。	pos := pos + 2; UNTIL ((pos = 101) OR (structarray[pos].value = targetvalue)) end_repeat;

例 2:

如果您需要	输入这个结构文本
<p>将 ASCII 字符从 SINT 数组移到字符串标签内。(在 SINT 数组中, 每个元素容纳一个字符。)当您到达回车符时停止。</p>	<pre>element_number := 0; SIZE(SINT_array, 0, SINT_array_size); Repeat String_tag.DATA[element_number] := SINT_array[element_number]; element_number := element_number + 1; String_tag.LEN := element_number; If element_number = SINT_array_size then exit; end_if; Until SINT_array[element_number] = 13 end_repeat;</pre>
<ol style="list-style-type: none">1. 将 Element_number 初始化为 0。2. 数出 SINT_array(包含 ASCII 字符的数组) 中元素的个数, 将结构存储在 SINT_array_size (DINT 标签) 内。3. 设置 String_tag[element_number] = SINT_array[element_number] 处的字符。4. 将 element_number 加 1。这样让控制器检查 SINT_array 中的下一个字符。5. 设置 String_tag 的长度成员 = element_number。(这记录了目前为止 String_tag 内的字符个数。)6. 如果 element_number = SINT_array_size, 那么停止。(您现在位于数组结尾, 此处不含回车符。)7. 如果 SINT_array[element_number] 处的字符 = 13 (回车符的十进制值), 那么停止。	
否则, 转到 3 。	

注释

要让您的结构文本更容易解释，请向其中添加注释。

- 注释可让您使用普通语言来介绍您的结构文本如何工作。
- 注释对结构文本的执行没有影响。

结构文本注释被下载到控制器内存中，可用于上载。要向您的结构文本中添加注释：

要添加注释	使用下面一种格式
在一行内	// 注释
在结构文本行的结尾处	(* 注释 *) /* 注释 */
在结构文本同一行内	(* 注释 *) /* 注释 */
跨多行	(* 注释开始 ... 注释结束 *) /* 注释开始 ... 注释结束 */

举例如下：

格式	示例
// 注释	在行的开头处 // 检查传送带方向 IF conveyor_direction THEN... 在行的结尾处 ELSE // 如果传送带不在运动，则设置警报指示灯 light := 1; END_IF;
(* 注释 *)	Sugar.Inlet[:=]1;(* 打开进口 *) IF Sugar.Low (* 低级别 LS*)& Sugar.High (* 高级别 LS*)THEN... (* 控制再循环泵的速度。速度由容器内的温度决定。 *) IF tank.temp > 200 THEN...
/* 注释 */	Sugar.Inlet:=0;/* 关闭进口 */ IF bar_code=65 /*A*/ THEN... /* 获取库存数组内元素的个数，将值存储在 Inventory_Items 标签内 */ SIZE(Inventory,0,Inventory_Items);

数字

1756-OB16IS 216

87589

H1 一级标题

我如何才能得到切线速度模式? 264

A**ASCII**

结构文本赋值 413

按位运算符

结构文本 419

B**表达式**

布尔表达式

结构文本 413

结构文本

按位运算符 419

概述 413

关系运算符 416

函数 415

逻辑运算符 418

算术运算符 415

数值表达式

结构文本 413

执行的顺序

结构文本 419

布尔表达式

结构文本 413

C**CAM_PROFILE 数据类型** 402**CAM 数据类型** 401**CASE** 425**错误**

附加信息 399

运动控制指令 391

错误代码

变频器 399

运动控制指令 391

D**电子齿轮** 33, 100**对称模式**

路径 262

多轴协调运动

圆形编程参考指南 337

多轴协调运动指令 249

MCCD 338

MCCM 292

MCLM 266

操作数 266

MCS 349

MCSD 358

MCSR 384

MCT 362

简介 249

运动插补动态调整 (MCCD)

操作数

变更加速度 343

否 343, 346

是 343, 346

变更减速度 344

否 344

是 344

变更速度 343

否 343

是 343

范围 347

梯形图 338

加速度 344

加速度单位 344

减速度 344

减速度单位 344

结构文本 340

速度 343

速度单位 343

运动控制 342

运动类型 343

协调运动 343

坐标系 342

错误代码 348

对状态位的更改 349

故障条件 348

扩展错误代码 348

描述 338

算术状态标志 347

运动插补关闭 (MCSD)

操作数 359

梯形图 359

结构文本 359

运动控制 359

坐标系 359

错误代码 360

对状态位的更改 360

协调运动状态位 360

轴状态位 360

坐标系状态位 360

故障条件 360

- 描述 358
- 算术状态标志 360
- 运动插补关闭复位 384
- 运动插补关闭复位 (MCSR)
 - 操作数 384
 - 梯形图 384
 - 结构文本 384
 - 运动控制 385
 - 坐标系 384
- 错误代码 385
- 对状态位的更改 386
 - 协调运动状态位 386
 - 轴状态位 386
 - 坐标系状态位 386
- 故障条件 385
- 描述 163, 384
- 算术状态标志 385
- 运动插补停止 349 (MCS)
 - 操作数 350
 - 错误代码 353
 - 梯形图 350
 - 减速度 353
 - 结构文本 351
 - 运动控制 353
 - 对状态位的更改 354
 - 轴状态位 354
 - 故障条件 353
 - 扩展错误代码 354
 - 描述 349
 - 算术状态标志 353
- 运动插补坐标转换 (MCT) 362
- 运动圆弧插补 292
- 运动圆弧插补 (MCCM)
 - 操作数 293
 - 方向 322
 - 合并 324
 - 禁止合并 325
 - 全部运动 325
 - 协调运动 325
 - 合并速度 325
 - 梯形图 293
 - 加速度 322
 - 加速度单位 322
 - 减速度 323
 - 减速度单位 323
 - 结构文本 295
 - 经由 / 中心 / 半径 321
 - 模式 323
 - 速度 322
 - 速度单位 322
- 位置 299
- 圆形类型 300
 - 半径 300
 - 经由 300
 - 中心 300
 - 中心增量 300
- 运动控制 298
- 运动类型 299
 - 绝对值 299
 - 增量 299
- 终止类型 257, 260, 262
 - 轮廓跟踪速度无限制 257, 260, 262
 - 轮廓跟踪速度限制 257, 260, 262
- 命令容差 257, 260, 262
- 实际容差 257, 260, 262
- 无安排 257, 260, 262
- 无减速 257, 260, 262
- 坐标系 297
- 对状态位的更改 334
 - 轴状态位 335
 - 坐标系状态位 335
- 故障条件 329
- 扩展错误代码 330
- 描述 292
- 目标位置输入对话框 325
- 算术状态标志 329
- 运动直线插补 266
- 运动直线插补 (MCLM)
 - 操作数
 - 合并 283
 - 禁止合并 283
 - 全部运动 283
 - 协调运动 283
 - 合并速度 283
 - 梯形图 266
 - 加速度 279
 - 加速度单位 279
 - 减速度 279
 - 减速度单位 279
 - 结构文本 269
 - 模式 280
 - S 曲线形 280
 - 速度模式效果 280
 - 梯形 280
 - 速度 279
 - 速度单位 279

- 位置 279
 - 运动控制 271
 - 运动类型 272
 - 绝对值 272
 - 增量 272
 - 终止类型 251, 257, 260, 262
 - 轮廓跟踪速度无限制 257, 260, 262
 - 轮廓跟踪速度限制 257, 260, 262
 - 命令容差 257, 260, 262
 - 实际容差 257, 260, 262
 - 无安排 257, 260, 262
 - 无减速 257
 - 坐标系 270
 - 对称模式 262
 - 对状态位的更改 290
 - 协调运动状态位 290
 - 轴状态位 290
 - 坐标系状态位 290
 - 故障条件 288
 - 扩展错误代码 289
 - 描述 266
 - 目标位置输入对话框 286
 - 速度模式 260
 - 算术状态标志 288
- F**
- FOR** **O** 428
 - 赋值**
 - ASCII 字符 413
 - 保持 411
 - 非保持 412
- G**
- 共线运动**
 - 速度模式
 - 终止类型 260
 - 构造**
 - 结构文本 421
 - 故障处理**
 - 变频器错误 399
 - 指令错误 391
 - 关闭运动监视** 184
 - 关闭运动凸轮输出** 34, 223
 - 关闭运动位置监视** 34
 - 关闭运动轴** 33
- 关闭运动组** 34, 171
 - 关闭轴位置抓拍** 34, 193
 - 关系运算符**
 - 结构文本 416
 - 关运动伺服使能** 33
 - 规划输出模块** 216
 - 过程运动指令** 22
- H**
- 函数**
 - 结构文本 415
- I**
- IF...THEN** 422
- J**
- 计划输出**
 - 16 个计划结构的数组 221
 - I/O 子系统 220
 - MAOC 指令的用法 217
 - 操作 216
 - 计划处理 221
 - 输出数据结构 221
 - 远程操作 217
 - 计算从轴运动值** 33
 - 计算从轴运动值 (MCSV)** 163
 - 计算凸轮运动曲线** 33, 123
- 结构**
- 参见数据类型
- 结构文本**
- CASE 425
 - FOR **O** 428
 - IF...THEN 422
 - REPEAT **NTIL** 434
 - WHILE **O** 431
 - 按位运算符 419
 - 表达式 413
 - 非保持赋值 412
 - 赋值 411
 - 赋值 ASCII 字符 413
 - 构造 421
 - 关系运算符 416
 - 函数 415
 - 逻辑运算符 418
 - 数值表达式 413
 - 算术运算符 415
 - 注释 437
 - 字符串取值 417
 - 组件 409

K

开启运动监视 180
 开启运动凸轮输出 34, 196
 开启运动位置监视 34
 开启轴位置抓拍 34, 186
 开运动伺服使能 33, 39

L

立即运动指令 19
 逻辑运算符
 结构文本 418

M

MAAT 228
 MAFR 55
 MAG 100
 MAH 72
 MAHD 239
 MAJ 76
 MAM 88
 MAOC 196, 223
 指令 201
 MAPC 128
 MAR 186
 MAS 58
 MASD 45
 对状态位的更改 47
 运动状态位 47
 轴状态位 47
 MASR 48
 MATC 150
 MAW 180
 MCCD
 操作数
 梯形图 338
 结构文本 340
 示例
 梯形图 349
 加速度和减速度值的变更
 对运动模式的影响
 344

MCCM

示例

二维弧形 300
 使用半径圆形类型 306
 使用经由圆形类型 304
 使用中心圆形 300

使用中心增量圆形类型 308

二维整圆 310
 结构文本 337
 三维弧形 317
 圆形类型经由 317
 圆形类型中心 318
 旋转轴 312
 运动类型为绝对值 312
 运动类型为增量 314
 圆形错误 331
 CIRCULAR_COLLIN
 EARITY_ERR
 OR (44) 331
 CIRCULAR_R1_R2_
 MISMATCH_E
 RROR (46)
 332
 CIRCULAR_SMALL_
 R_ERROR
 (49) 333, 334
 CIRCULAR_START_
 END_ERROR
 (45) 332

MCCP 123**MCD 指令 111****MCLM**

示例

合并 284
 混合
 不同速度 265
 终止类型 251
 旋转轴 276
 运动类型为绝对值 276
 运动类型为增量 277
 有关合并指令的附加注释
 286
 运动类型 272

MCS

操作数
 梯形图 350
 示例

梯形图 355

MCSD

示例

梯形图 362

MCSR

示例

梯形图 386
结构文本 386

MCT 362
MCTP 375
MCT 指令 362
MDF 53
MDO 指令 50
MDR 193
MDW 184
MGS 166
MGSD 171
MGSP 176
MGSR 指令 174
MOTION_GROUP 数据类型 403
MOTION_INSTRUCTION 数据类型
404
MRAT 233
MRHD 243
MRP 118
MSF 42
MSO 39

O

OUTPUT_CAM 数据类型 405
OUTPUT_COMPENSATION 数据类型
407

P

postscan
结构文本 412

R

REPEAT NTIL 434

S

S 曲线模式
整定 387
数据类型
CAM 401
CAM_PROFILE 402
MOTION_GROUP 403
MOTION_INSTRUCTION 404
OUTPUT_CAM 405
OUTPUT_COMPENSATION 407
数学运算符
结构文本 415

数值表达式 413

说明

结构文本 437

速度模式

共线运动 260
切线 264

算术运算符

结构文本 415

W

WHILE O 431

文档

结构文本 437

X

消息运动指令 21

Y

应用轴连接诊断 34, 239

应用轴整定参数 34, 228

跃度

整定 387

运动

错误代码 391
过程类型的指令 22
立即类型的指令 19
数据类型 401
消息类型的指令 21

运动插补动态调整 35

运动插补关闭 35

运动插补关闭复位 35

运动插补停止 35

运动插补坐标转换 362

运动动态调整 33, 111

运动控制指令

错误代码 391
协调运动指令

运动插补动态调整 (MCCD) 35

运动插补关闭 (MCSD) 35

运动插补关闭复位 (MCSR) 35

运动插补停止 (MCS) 35

运动圆弧插补 (MCCM) 35

运动直线插补 (MCLM) 35

运动配置指令

应用轴连接诊断 (MAHD) 34

应用轴整定参数 (MAAT) 34

运行轴连接诊断 (MRHD) 34

运行轴整定 (MRAT) 34

运动事件指令

关闭运动凸轮输出 (MDOC)

- 34
- 关闭运动位置监视 (MDW) 34
- 关闭轴位置抓拍 (MDR) 34
- 开启运动凸轮输出 (MAOC) 34
- 开启运动位置监视 (MAW) 34
- 开启轴位置抓拍 (MAR) 34
- 运动位移指令
 - 电子齿轮 (MAG) 33
 - 计算从轴运动值 33
 - 计算凸轮运动曲线 (MCCP) 33
 - 运动动态调整 (MCD) 33
 - 运动位置重设 (MRP) 33
 - 运动轴时间凸轮 (MATC) 33
 - 运动轴停止 (MAS) 33
 - 运动轴位置凸轮 (MAPC) 33
 - 运动轴寻参 (MAH) 33
 - 轴速度 (MAJ) 33
 - 轴位移 (MAM) 33
- 运动控制状态指令
 - 关闭运动轴 (MASD) 33
 - 关运动伺服使能 (MSF) 33
 - 开运动伺服使能 (MSO) 33
 - 运动直接使能 (MDO) 33
 - 运动轴故障复位 (MAFR) 33
 - 运动轴关闭复位 (MASR) 33
 - 直接关运动使能 (MDF) 33
- 运动组指令
 - 关闭运动组 (MGSD) 34
 - 运动组关闭复位 (MGSR) 34
 - 运动组停止 (MGS) 34
 - 抓拍运动组位置 (MGSP) 34
- 运动配置指令 227
 - 简介 227
 - 应用轴连接诊断 (MAHD) 239
 - MOTION_INSTRUCTION 结构 240
 - 编码器连接测试 241
 - 操作数 239
 - 梯形图 239
 - 结构文本 239
 - 电机编码器连接测试 240
 - 描述 240
 - 状态位 242
 - 应用轴整定参数 (MAAT) 228
 - MOTION_INSTRUCTION 结构 228
 - 操作数 228
 - 梯形图 228
 - 结构文本 228
 - 描述 228
 - 状态位 232
 - 运行轴连接诊断 (MRHD) 243
 - MOTION_INSTRUCTION 结构 244
 - 编码器连接测试 245
 - 标志连接测试 246
 - 操作数 243
 - 梯形图 243
 - 结构文本 243
 - 测试状态 246
 - 电机编码器连接测试 244
 - 更改至状态位 248
 - 看门狗 OK 测试 246
 - 扩展错误代码 248
 - 描述 244
 - 运行轴整定 (MRAT) 233
 - MOTION_INSTRUCTION 结构 233
 - 操作数 233
 - 梯形图 233
 - 结构文本 233
 - 更改至状态位 238
 - 扩展错误代码 238
 - 描述 233
 - 整定状态参数 236
- 运动事件指令 179
 - 关闭运动监视 (MDW) 184
 - MOTION_INSTRUCTION 结构 184
 - 操作数 184
 - 梯形图 184
 - 结构文本 184
 - 对状态位的更改 185
 - 描述 184
 - 关闭运动凸轮输出 (MDOC) 223
 - MOTION_INSTRUCTION 结构 224
 - 操作数 223
 - 梯形图 223
 - 结构文本 223
 - 扩展错误代码 224
 - 描述 224
 - 关闭运动凸轮输出 (MDOC) 状态位 225
 - 关闭轴位置抓拍 (MDR) 193
 - MOTION_INSTRUCTION 结构 193
 - 操作数 193
 - 梯形图 193
 - 结构文本 193
 - 对状态位的更改 195
 - 扩展错误代码 194
 - 描述 193
 - 简介 179
 - 开启运动监视 (MAW) 180
 - MOTION_INSTRUCTION 结

- 构 181
 - 操作数 180
 - 梯形图 180
 - 结构文本 180
 - 对状态位的更改 183
 - 扩展错误代码 182
 - 描述 181
- 开启运动凸轮输出 (MAOC) 196
 - MOTION_INSTRUCTION 结构 202
 - 参考 212
 - 操作数 197
 - 梯形图 197
 - 结构文本 200
 - 持续时间 206
 - 错误代码 213
 - 对状态位的影响 215
 - 解锁存类型 205
 - 扩展错误代码 215
 - 描述 202
 - 模式补偿 209
 - 偏移和延迟补偿 208
 - 使能类型 206
 - 输出 210
 - 输出补偿数组检查 210
 - 输入 211
 - 锁存类型 204
 - 凸轮起始位置和凸轮终止位置 211
 - 凸轮输出数组检查 207
 - 执行计划 211
 - 执行模式 211
 - 执行目标 202
 - 指定输出补偿 208
 - 指定凸轮输出曲线 203
 - 轴 202
 - 轴准备和凸轮准备位置 212
 - 左右凸轮位置 206
- 开启轴位置抓拍 (MAR) 186
 - MOTION_INSTRUCTION 结构 187
 - 操作数 186
 - 梯形图 186
 - 结构文本 187
 - 窗口登录 188
 - 对状态位的更改 191
 - 扩展错误代码 191
 - 描述 188
- 运动位移**
 - MCD 111
- 运动位移指令**
 - MCSV 163
 - 操作数
 - 梯形图 58
- 电子齿轮 (MAG) 100
 - MOTION_INSTRUCTION 结构 103
 - 操作数 100
 - 梯形图 100
 - 结构文本 102
 - 齿轮兼位移 108
 - 从动至命令位置 104
 - 从动至实际位置 104
 - 对状态位的更改 109
 - 反方向齿轮 105
 - 反转齿轮方向 105
 - 分数齿轮速比 106
 - 更改齿轮速比 105
 - 更改主轴 107
 - 扩展错误代码 108
 - 离合器 106
 - 描述 103
 - 实数齿轮速比 105
 - 同方向齿轮 105
- 计算从轴运动值 (MCSV) 163
 - 操作数 163
 - 梯形图 163
 - 结构文本 163
 - 运动控制 164
 - 对状态位的更改 164
 - 故障条件 164
 - 结构文本 163
 - 算术状态标志 164
- 计算凸轮运动曲线 (MCCP) 123
 - MOTION_INSTRUCTION 结构 124
 - 操作数 123
 - 梯形图 123
 - 结构文本 123
 - 计算凸轮曲线 125
 - 扩展错误代码 127
 - 描述 124
 - 起始斜率和终止斜率 126
 - 凸轮曲线数组状态成员 125
 - 线性和三次方样条内插 125
 - 指定凸轮曲线标签 124
 - 指定凸轮数组 124
- 简介 57
- 运动动态调整 (MCD) 111
 - MOTION_INSTRUCTION 结构 115
 - 操作数 111
 - 梯形图 111
 - 结构文本 113
 - 更改点动动态参数 116
 - 更改位移动态参数 116

- 扩展错误代码 116
- 描述 115
- 暂停位移 116
- 运动位置重设 (MRP) 118
 - MOTION_INSTRUCTION 结构 119
 - 操作数 118
 - 梯形图 118
 - 结构文本 119
 - 绝对值模式 119
 - 扩展错误代码 122
 - 描述 119
 - 命令位置 121
 - 实际位置 120
 - 相对值模式 120
- 运动轴时间凸轮 (MATC) 150
 - MOTION_INSTRUCTION 结构 153
 - 操作数 150
 - 梯形图 150
 - 结构文本 152
 - 反方向凸轮 154
 - 更改凸轮方向 154
 - 更改凸轮曲线 154
 - 扩展错误代码 161, 164
 - 立即执行 157
 - 描述 153
 - 时间凸轮标定 156
 - 停止凸轮 160
 - 同方向凸轮 153
 - 凸轮方向 153
 - 凸轮合并 160
 - 凸轮曲线数组检查 155
 - 凸轮曲线执行模式 157
 - 线性和三次方内插 155
 - 执行计划 157
 - 指定凸轮曲线 154
- 运动轴停止 (MAS) 58
 - MOTION_INSTRUCTION 结构 61, 353
 - 操作数 58
 - 梯形图 58
 - 结构文本 60
 - 扩展错误代码 63, 81, 378
 - 描述 61
- 运动轴位置凸轮 (MAPC) 128
 - MOTION_INSTRUCTION 结构 133
 - 保留当前凸轮方向 134
 - 操作数 129
 - 梯形图 129
 - 结构文本 132
 - 从动至命令位置 143
 - 从动至实际位置 143
 - 对状态位的更改 147
 - 反方向凸轮 134
 - 反转当前凸轮方向 134
 - 更改凸轮锁定位置 138
 - 故障恢复 145
 - 挂起凸轮执行 141
 - 仅正向、仅反向或双向执行 139
 - 扩展错误代码 147
 - 描述 134
 - 停止凸轮 145
 - 同方向凸轮 134
 - 凸轮方向 134
 - 凸轮合并 145
 - 凸轮兼位移 144
 - 凸轮曲线数组检查 135
 - 凸轮曲线执行模式 137
 - 位置凸轮标定 136
 - 线性和三次方内插 136
 - 增量位移 144
 - 执行计划 137
 - 立即执行 138
 - 指定凸轮曲线 135
 - 主轴参考 142
 - 主轴方向 143
 - 主轴偏移位移 144
- 运动轴寻参 (MAH) 72
 - MOTION_INSTRUCTION 结构 72
 - 被动归零 73
 - 操作数 72
 - 梯形图 72
 - 结构文本 72
 - 对状态位的更改 75
 - 绝对归零 73
 - 扩展错误代码 74
 - 描述 73
 - 主动归零 73
- 轴速度 (MAJ) 76, 362
 - MOTION_INSTRUCTION 结构 79, 363, 377
 - 操作数 76, 362, 375
 - 梯形图 76
 - 结构文本 78, 363, 376
 - 描述 80, 365, 378
 - 速度模式效果 27
- 轴位移 (MAM) 88
 - MOTION_INSTRUCTION 结构 92
 - 操作数 88
 - 梯形图 88

- 结构文本 91
 - 绝对值位移 95
 - 扩展错误代码 97
 - 描述 92
 - 旋转最短路径位移 95
- 抓拍运动组位置 (MGSP)
 - 状态位 177
- 运动位置重设** 33, 118
- 运动协调指令**
 - 参见多轴协调运动指令
- 运动学**
 - 参见多轴协调运动指令
- 运动圆弧插补** 35
- 运动直接使能** 33, 50
- 运动直线插补** 35
- 运动轴故障复位** 33
- 运动轴关闭复位** 33
- 运动轴时间凸轮** 33, 150
- 运动轴停止** 33, 58
- 运动轴位置凸轮** 33, 128
- 运动轴寻参** 33, 72
- 运动控制状态**
 - MDO 50
- 运动控制状态指令** 37
 - 关闭运动轴 (MASD) 45
 - MOTION_INSTRUCTION 结构 45
 - 操作数 45
 - 梯形图 45
 - 结构文本 45
 - 描述 45
 - 关运动伺服使能 (MSF) 42
 - MOTION_INSTRUCTION 结构 42
 - 操作数 42
 - 梯形图 42
 - 结构文本 42
 - 描述 42
 - 简介 37
 - 开运动伺服使能 (MSO) 39
 - MOTION_INSTRUCTION 结构 39
 - 操作数 39
 - 梯形图 39
 - 结构文本 39
 - 描述 39
 - 运动直接使能 (MDO) 50
 - MOTION_INSTRUCTION 结构 50
 - 操作数 50
 - 梯形图 50
 - 结构文本 50
 - 对状态位的更改 52
 - 描述 51
- 运动轴故障复位 (MAFR)** 55
 - MOTION_INSTRUCTION 结构 55
 - 操作数 55
 - 梯形图 55
 - 结构文本 55
 - 描述 55
- 运动轴关闭复位 (MASR)** 48
 - MOTION_INSTRUCTION 结构 48
 - 操作数 48
 - 梯形图 48
 - 结构文本 48
 - 描述 48
- 直接关运动使能 (MDF)** 53
 - MOTION_INSTRUCTION 结构 53
 - 操作数 53
 - 梯形图 53
 - 结构文本 53
 - 描述 53
- 运动组**
 - MGSR 174
- 运动组关闭复位** 34, 174
- 运动组停止** 34, 166
- 运动组指令** 165
 - 关闭运动组 (MGSD) 171
 - MOTION_INSTRUCTION 结构 171
 - 操作数 171
 - 梯形图 171
 - 结构文本 171
 - 对状态位的更改 172
 - 描述 171
 - 简介 165
 - 运动组关闭复位 (MGSR) 174
 - MOTION_INSTRUCTION 结构 174
 - 操作数 174
 - 梯形图 174
 - 结构文本 174
 - 对状态位的更改 175
 - 描述 174
 - 运动组停止 (MGS) 166
 - MOTION_INSTRUCTION 结构 167
 - 操作数 166
 - 梯形图 166
 - 结构文本 166
 - 对状态位的更改 170

- 快速关闭 168
- 快速禁止 168
- 快速停止 168
- 描述 167
- 硬关闭 169
- 硬禁止 168
- 抓拍运动组位置 (MGSP) 176
 - MOTION_INSTRUCTION 结构 176
 - 操作数 176
 - 梯形图 176
 - 结构文本 176
 - 描述 176
- 运算符**
 - 执行的顺序
 - 结构文本 419
- 运行轴连接诊断** 34, 243
- 运行轴整定** 34, 233

Z

- 整定**
 - S 曲线模式 387
 - 跃度 387
- 直接关运动使能** 33
- 直接命令**
 - 支持的命令
 - 运动控制状态 33
- 执行的顺序**
 - 结构文本 419
- 轴速度** 33, 76, 362
- 轴位移** 33, 88
- 注释**
 - 结构文本 437
- 抓拍运动组位置** 34, 176
- 字符串**
 - 结构文本中的取值 417
- 坐标转换**
 - 开始坐标转换 362

罗克韦尔自动化支持

罗克韦尔自动化有限公司在网站上提供了技术信息以期帮助您使用其产品。在网站 <http://support.rockwellautomation.com> 上，您可以找到技术手册、常见问题 (FAQ) 知识库、技术与应用说明、示例代码和指向软件补丁包的链接，以及 MySupport 功能—您可以自定义此功能以便充分利用这些工具。

如果在安装、配置和故障处理方面需要进一步的电话技术支持，我们将提供 TechConnect 支持程序。如需更多信息，请联系本地的分销商或罗克韦尔自动化有限公司驻本地代表处，或者访问 <http://support.rockwellautomation.com>。

安装帮助

如果您在安装硬件模块的最初 24 小时内遇到了问题，请查阅本手册中的相关信息。另外，您也可以拨打特别用户支持号码以求得初步帮助，从而让您的模块正常运行起来：

美国	1.440.646.3223 星期一到星期五，上午 8:00 至下午 5:00(美国东部时间)
美国以外地区	请联系罗克韦尔自动化有限公司驻本地代表处以解决任何技术支持问题。

新产品退货

罗克韦尔对其所有产品进行了测试，以期确保产品在离厂时性能完全正常。但是，如果您的产品工作不正常而需要退货：

美国	联系分销商。您须向分销商提供用户支持代码 (使用上述电话号码可获得一个代码)，才可以完成退货程序。
美国以外地区	有关退货程序，请联系罗克韦尔自动化有限公司驻本地代表处。

www.rockwellautomation.com

动力、控制与信息解决方案总部

美洲罗克韦尔自动化有限公司：1201 South Second Street, Milwaukee, WI 53204-2496, 美国，电话：(1) 414.382.2000，传真：(1) 414.382.4444

欧洲/中东/非洲罗克韦尔自动化有限公司：Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, 比利时，电话：(32) 2 663 0600，传真：(32) 2 663 0640

亚太罗克韦尔自动化有限公司：香港数码港道100号数码港3座F区14楼，电话：(852) 2887 4788，传真：(852) 2508 1846

出版号 1756-RM007J-ZH-P - 2008 年 7 月

© 2008 罗克韦尔自动化有限公司版权所有。美国印刷